

9.0

IBM Message Service Client for .NET

IBM

Nota

Prima di utilizzare queste informazioni e il prodotto che supportano, leggere le informazioni in [“Informazioni particolari” a pagina 251](#).

Questa edizione si applica alla release 0 della versione 9 di IBM® MQ e a tutte le release e modifiche successive, se non diversamente indicato nelle nuove edizioni.

Quando si inviano informazioni a IBM, si concede a IBM un diritto non esclusivo di utilizzare o distribuire le informazioni in qualsiasi modo ritenga appropriato senza incorrere in alcun obbligo verso l'utente.

© **Copyright International Business Machines Corporation 2007, 2023.**

Indice

Message Service Client for .NET.....	5
Introduzione a IBM Message Service Client for .NET.....	5
Stili di messaggistica.....	6
Modello oggetto XMS.....	7
Attributi e proprietà degli oggetti.....	8
oggetti gestiti.....	9
Il modello di messaggio XMS.....	10
Come impedire alle applicazioni di utilizzare la versione più recente di XMS.....	12
Impostazione dell'ambiente del server di messaggistica.....	13
Configurazione del gestore code e del broker per un'applicazione che si connette a un gestore code IBM MQ.....	13
Configurazione di un broker per un'applicazione che utilizza una connessione in tempo reale a un broker.....	15
Configurazione del service integration bus per un'applicazione che si connette a WebSphere Application Server.....	16
Utilizzo di applicazioni di esempio XMS.....	17
Le applicazioni di esempio.....	17
Esecuzione delle applicazioni di esempio.....	19
Creazione delle applicazioni di esempio .NET.....	19
Sviluppo di applicazioni XMS.....	20
Scrittura di applicazioni XMS.....	20
Scrittura di applicazioni XMS .NET.....	44
Utilizzo degli oggetti gestiti.....	50
Protezione delle comunicazioni per applicazioni XMS.....	66
XMS messaggi.....	69
Risoluzione dei problemi.....	83
Configurazione della traccia per applicazioni .NET.....	83
Configurazione FFDC per applicazioni .NET.....	88
Suggerimenti per la risoluzione dei problemi.....	88
Riferimento per Message Service Client for .NET.....	90
.NETInterfacce.....	90
Proprietà degli oggetti XMS.....	180
Informazioni particolari.....	251
Informazioni sull'interfaccia di programmazione.....	252
Marchi.....	252

Introduzione a IBM Message Service Client for .NET

IBM Message Service Client for .NET fornisce un'API (application programming interface) denominata XMS che ha la stessa serie di interfacce di Java Message Service (JMS) API. IBM Message Service Client for .NET contiene un'implementazione completamente gestita di XMS, che può essere utilizzata da qualsiasi linguaggio conforme a .NET .

XMS supporta:

- Point-to-point messaggistica
- Pubblicazione/sottoscrizione messaggistica
- Consegna sincrona dei messaggi
- Consegna di messaggi asincroni

Un'applicazione XMS può scambiare messaggi con i seguenti tipi di applicazione:

- Un'applicazione XMS
- Un'applicazione IBM MQ classes for JMS
- Un'applicazione IBM MQ nativa
- Un'applicazione JMS che utilizza il provider di messaggistica predefinito IBM MQ

Un'applicazione XMS può connettersi e utilizzare le risorse di uno dei seguenti server di messaggistica:

IBM MQ Gestore code

L'applicazione può connettersi in modalità bind o client.

WebSphere Application Server Servizio Integration Bus

L'applicazione può utilizzare una connessione TCP/IP diretta oppure può utilizzare HTTP su TCP/IP.

IBM Integration Bus

I messaggi vengono trasportati tra l'applicazione e il broker utilizzando WebSphere MQ Real-time Transport. I messaggi possono essere consegnati all'applicazione utilizzando WebSphere MQ Multicast Transport.

Connettendosi a un gestore code IBM MQ , un'applicazione XMS può utilizzare WebSphere MQ Enterprise Transport per comunicare con IBM Integration Bus. In alternativa, un'applicazione XMS può pubblicare e sottoscrivere collegandosi a IBM MQ.

Concetti correlati

[“Stili di messaggistica” a pagina 6](#)

[“Modello oggetto XMS” a pagina 7](#)

L'API XMS è un'interfaccia orientata agli oggetti. Il modello oggetto XMS è basato sul modello oggetto JMS 1.1 .

[“Il modello di messaggio XMS” a pagina 10](#)

Il modello di messaggio XMS è uguale al modello di messaggio IBM MQ classes for JMS .

Introduzione a IBM Message Service Client for .NET

IBM Message Service Client for .NET fornisce un'API (application programming interface) denominata XMS che ha la stessa serie di interfacce di Java Message Service (JMS) API. IBM Message Service Client for .NET contiene un'implementazione completamente gestita di XMS, che può essere utilizzata da qualsiasi linguaggio conforme a .NET .

XMS supporta:

- Point-to-point messaggistica
- Pubblicazione/sottoscrizione messaggistica
- Consegna sincrona dei messaggi

- Consegna di messaggi asincroni

Un'applicazione XMS può scambiare messaggi con i seguenti tipi di applicazione:

- Un'applicazione XMS
- Un'applicazione IBM MQ classes for JMS
- Un'applicazione IBM MQ nativa
- Un'applicazione JMS che utilizza il provider di messaggistica predefinito IBM MQ

Un'applicazione XMS può connettersi e utilizzare le risorse di uno dei seguenti server di messaggistica:

IBM MQ Gestore code

L'applicazione può connettersi in modalità bind o client.

WebSphere Application Server Servizio Integration Bus

L'applicazione può utilizzare una connessione TCP/IP diretta oppure può utilizzare HTTP su TCP/IP.

IBM Integration Bus

I messaggi vengono trasportati tra l'applicazione e il broker utilizzando WebSphere MQ Real-time Transport. I messaggi possono essere consegnati all'applicazione utilizzando WebSphere MQ Multicast Transport.

Connettendosi a un gestore code IBM MQ, un'applicazione XMS può utilizzare WebSphere MQ Enterprise Transport per comunicare con IBM Integration Bus. In alternativa, un'applicazione XMS può pubblicare e sottoscrivere collegandosi a IBM MQ.

Concetti correlati

[“Stili di messaggistica” a pagina 6](#)

[“Modello oggetto XMS” a pagina 7](#)

L'API XMS è un'interfaccia orientata agli oggetti. Il modello oggetto XMS è basato sul modello oggetto JMS 1.1.

[“Il modello di messaggio XMS” a pagina 10](#)

Il modello di messaggio XMS è uguale al modello di messaggio IBM MQ classes for JMS.

Stili di messaggistica

XMS supporta gli stili Point-to-point e Pubblicazione/sottoscrizione di messaggistica.

Gli stili di messaggistica sono anche chiamati domini di messaggistica.

Point-to-point messaggistica

Una forma comune di messaggistica Point-to-point utilizza l'accodamento. Nel caso più semplice, un'applicazione invia un messaggio a un'altra applicazione identificando, implicitamente o esplicitamente, una coda di destinazione. Il sistema di messaggistica e accodamento sottostante riceve il messaggio dall'applicazione mittente e instrada il messaggio alla coda di destinazione. L'applicazione ricevente può quindi richiamare il messaggio dalla coda.

Se il sistema di messaggistica e accodamento sottostante contiene IBM Integration Bus, IBM Integration Bus potrebbe replicare un messaggio e instradare le copie del messaggio a code differenti. Di conseguenza, più di un'applicazione può ricevere il messaggio. IBM Integration Bus potrebbe anche trasformare un messaggio e aggiungervi dati.

Una caratteristica chiave della messaggistica Point-to-point è che un'applicazione inserisce un messaggio in una coda locale quando invia un messaggio. Il sistema di messaggistica e accodamento sottostante determina a quale coda di destinazione viene inviato il messaggio. L'applicazione ricevente richiama il messaggio dalla coda di destinazione.

Pubblicazione / sottoscrizione messaggistica

Nella messaggistica Pubblicazione/sottoscrizione esistono due tipi di applicazione: publisher e sottoscrittore.

Un *publisher* fornisce informazioni sotto forma di messaggi di pubblicazione. Quando un publisher pubblica un messaggio, specifica un argomento che identifica l'oggetto delle informazioni all'interno del messaggio.

Un *sottoscrittore* è un consumatore delle informazioni pubblicate. Un sottoscrittore specifica gli argomenti a cui è interessato creando sottoscrizioni.

Il sistema di pubblicazione / sottoscrizione riceve le pubblicazioni dai publisher e le sottoscrizioni dai sottoscrittori. Instrada le pubblicazioni ai sottoscrittori. Un sottoscrittore riceve le pubblicazioni solo sugli argomenti sottoscritti.

Una caratteristica chiave della messaggistica Pubblicazione/sottoscrizione è che un publisher identifica un argomento quando pubblica un messaggio. Non identifica i sottoscrittori. Se un messaggio viene pubblicato su un argomento per cui non sono presenti sottoscrittori, nessuna applicazione riceve il messaggio.

Un'applicazione può essere sia un autore (publisher) che un sottoscrittore (subscriber).

Modello oggetto XMS

L'API XMS è un'interfaccia orientata agli oggetti. Il modello oggetto XMS è basato sul modello oggetto JMS 1.1 .

Il seguente elenco riassume le classi XMS principali o i tipi di oggetto:

ConnectionFactory

Un oggetto `ConnectionFactory` contiene una serie di parametri per una connessione. Un'applicazione utilizza `ConnectionFactory` per creare una connessione. Un'applicazione può fornire i parametri al runtime e creare un oggetto `ConnectionFactory` . In alternativa, i parametri di connessione possono essere memorizzati in un repository di oggetti gestiti. Un'applicazione può richiamare un oggetto dal repository e creare un oggetto `ConnectionFactory` da esso.

Connection

Un oggetto `Connection` incapsula una connessione attiva da un'applicazione a un server di messaggistica. Un'applicazione utilizza una connessione per creare sessioni.

Destination

Un'applicazione invia o riceve messaggi utilizzando un oggetto `Destination` . Nel dominio Pubblicazione/sottoscrizione , un oggetto `Destination` incapsula un argomento e, nel dominio Point-to-point , un oggetto `Destination` incapsula una coda. Un'applicazione può fornire i parametri per creare un oggetto `Destination` al runtime. In alternativa, è possibile creare un oggetto `Destination` da una definizione di oggetto memorizzata nel repository di oggetti gestiti.

Session

Un oggetto `Session` è un contesto a thread singolo per l'invio e la ricezione di messaggi. Un'applicazione utilizza un oggetto `Session` per creare oggetti `Message`, `MessageProducer` e `MessageConsumer` .

Message

Un oggetto `Message` incapsula l'oggetto `Message` che un'applicazione invia utilizzando un oggetto `MessageProducer` o che riceve utilizzando un oggetto `MessageConsumer` .

MessageProducer

Un oggetto `MessageProducer` viene utilizzato da un'applicazione per inviare messaggi a una destinazione.

MessageConsumer

Un oggetto `MessageConsumer` viene utilizzato da un'applicazione per ricevere i messaggi inviati ad una destinazione.

Figura 1 a pagina 8 mostra questi oggetti e le relazioni. Questo diagramma mostra i principali tipi di oggetto XMS : `ConnectionFactory`, `Connection`, `Session`, `MessageProducer`, `MessageConsumer`, `Message` e `Destination`. Un'applicazione utilizza una produzione connessioni per creare una connessione e una connessione per creare sessioni. L'applicazione può quindi utilizzare una sessione per creare messaggi, produttori di messaggi e utenti di messaggi. L'applicazione utilizza un producer di messaggi per inviare

messaggi a una destinazione e utilizza un consumer di messaggi per ricevere i messaggi inviati a una destinazione.

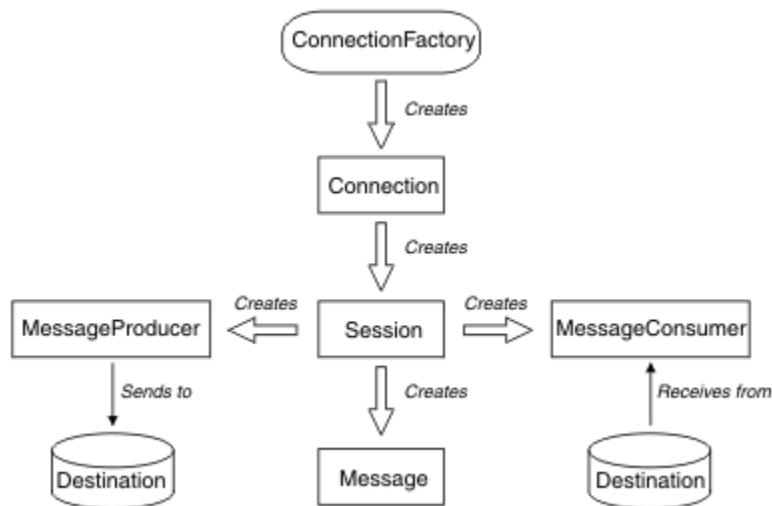


Figura 1. Oggetti XMS e relative relazioni

In .NET, le classi XMS sono definite come una serie di interfacce .NET . Quando si codificano applicazioni XMS .NET , sono necessarie solo le interfacce dichiarate.

Il modello di oggetto XMS si basa sulle interfacce indipendenti dal dominio descritte in Java Message Service Specifica, Versione 1.1. Le classi specifiche del dominio, come Topic, TopicPublisher e TopicSubscriber, non vengono fornite.

Attributi e proprietà degli oggetti

Un oggetto XMS può avere attributi e proprietà, che sono caratteristiche dell'oggetto, che sono implementati in modi diversi.

Attributi

Una caratteristica dell'oggetto che è sempre presente e occupa la memoria, anche se l'attributo non ha un valore. In questo senso, un attributo è simile a un campo in una struttura dati a lunghezza fissa. Una caratteristica distintiva degli attributi è che ogni attributo ha i suoi metodi per impostare e ottenere il suo valore.

Proprietà

Una proprietà di un oggetto è presente e occupa memoria solo dopo che è stato impostato il relativo valore. Una proprietà non può essere eliminata o la sua memoria non può essere ripristinata dopo che il suo valore è stato impostato. È possibile modificarne il valore. XMS fornisce una serie di metodi generici per l'impostazione e il richiamo dei valori delle proprietà.

Concetti correlati

XMS Tipi primitivi

XMS fornisce equivalenti degli otto tipi primitivi Java (byte, short, int, long, float, double, char e boolean). Ciò consente lo scambio di messaggi tra XMS e JMS senza che i dati vadano persi o danneggiati.

Conversione implicita di un valore di proprietà da un tipo di dati a un altro

Quando un'applicazione acquisisce il valore di una proprietà, il valore può essere convertito da XMS in un altro tipo di dati. Molte regole regolano quali conversioni sono supportate e come XMS esegue le conversioni.

Riferimenti correlati

Tipi di dati per elementi di dati dell'applicazione

Per essere certi che un'applicazione XMS possa scambiare messaggi con un'applicazione IBM MQ classes for JMS , entrambe le applicazioni devono essere in grado di interpretare i dati dell'applicazione nel corpo di un messaggio nello stesso modo.

oggetti gestiti

Utilizzando gli oggetti gestiti, è possibile gestire le impostazioni di connessione utilizzate dalle applicazioni client da gestire da un repository centrale. Un'applicazione richiama le definizioni degli oggetti dal repository centrale e le usa per creare oggetti `ConnectionFactory` e `Destination`. Utilizzando gli oggetti gestiti, è possibile disaccoppiare le applicazioni dalle risorse che utilizzano al runtime.

Ad esempio, le applicazioni XMS possono essere scritte e verificate con oggetti gestiti che fanno riferimento a un insieme di connessioni e destinazioni in un ambiente di test. Quando le applicazioni vengono distribuite, gli oggetti gestiti possono essere modificati per configurare le applicazioni in modo che facciano riferimento a connessioni e destinazioni nell'ambiente di produzione.

XMS supporta due tipi di oggetti gestiti:

- Un oggetto `ConnectionFactory`, utilizzato dalle applicazioni per stabilire la connessione iniziale al server.
- Un oggetto `Destination`, utilizzato dalle applicazioni per specificare la destinazione per i messaggi che vengono inviati e l'origine dei messaggi che vengono ricevuti. Una destinazione è un argomento o una coda sul server a cui si connette un'applicazione.

Lo strumento di amministrazione **JMSAdmin** viene fornito con IBM MQ. Viene utilizzato per creare e gestire oggetti gestiti in un repository centrale di oggetti gestiti.

Gli oggetti gestiti nel repository possono essere utilizzati dalle applicazioni IBM MQ classes for JMS e XMS. Le applicazioni XMS possono utilizzare gli oggetti `ConnectionFactory` e `Destination` per connettersi a un IBM MQ gestore code. Un amministratore può modificare le definizioni di oggetto contenute nel repository senza influenzare il codice dell'applicazione.

Il seguente diagramma mostra in che modo un'applicazione XMS generalmente utilizza gli oggetti gestiti. Il lato sinistro del diagramma mostra un repository contenente le definizioni di `ConnectionFactory` e degli oggetti di destinazione gestiti mediante una console di amministrazione. La parte destra del diagramma mostra un'applicazione XMS che ricerca le definizioni degli oggetti nel repository e utilizza queste definizioni di oggetti durante la connessione a un server di messaggistica.

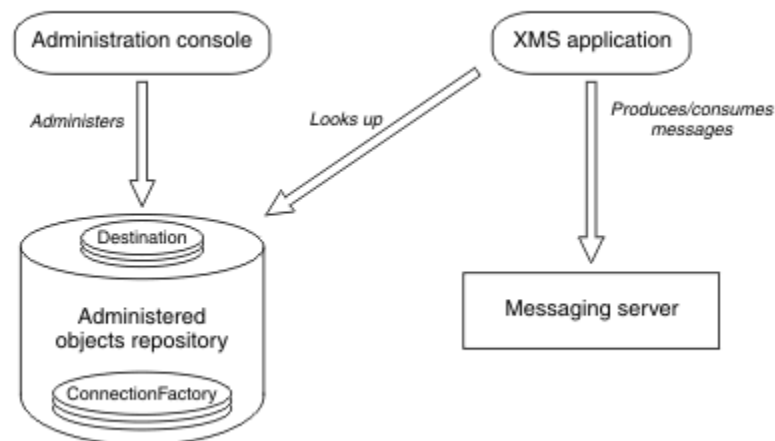


Figura 2. Utilizzo tipico di oggetti gestiti da un'applicazione XMS

Concetti correlati

Utilizzo degli oggetti gestiti

Gli argomenti in questa sezione forniscono informazioni sugli oggetti gestiti. Le applicazioni XMS possono recuperare le definizioni degli oggetti da un repository di oggetti amministrato centrale e utilizzarli per creare factory di connessione e destinazioni.

Tipi supportati di repository di oggetti gestiti

Gli oggetti gestiti del file system e LDAP possono essere utilizzati per connettersi a IBM MQ e WebSphere Application Server, mentre la denominazione COS può essere utilizzata per connettersi solo a WebSphere Application Server.

Attività correlate

Creazione di oggetti gestiti

Le definizioni dell'oggetto ConnectionFactory e Destination richieste dalle applicazioni XMS per stabilire una connessione a un server di messaggistica devono essere create utilizzando gli strumenti di gestione appropriati.

Il modello di messaggio XMS

Il modello di messaggio XMS è uguale al modello di messaggio IBM MQ classes for JMS .

In particolare XMS implementa gli stessi campi di intestazione del messaggio e proprietà del messaggio che IBM MQ classes for JMS implementa:

- JMS campi di intestazione. Questi campi hanno nomi che iniziano con il prefisso JMS.
- JMS proprietà definite. Questi campi hanno proprietà i cui nomi iniziano con il prefisso JMSX.
- IBM proprietà definite. Questi campi hanno proprietà i cui nomi iniziano con il prefisso JMS_IBM_.

Di conseguenza, le applicazioni XMS possono scambiare messaggi con le applicazioni IBM MQ classes for JMS . In ogni messaggio, alcuni dei campi di intestazione e delle proprietà vengono impostati dall'applicazione e altri vengono impostati da XMS o IBM MQ classes for JMS. Alcuni dei campi impostati da XMS o IBM MQ classes for JMS vengono impostati quando il messaggio viene inviato e altri quando viene ricevuto. I campi di intestazione e le proprietà vengono propagati con un messaggio tramite un server di messaggistica, dove appropriato. Vengono resi disponibili a tutte le applicazioni che ricevono il messaggio.

Installazione di Message Service Client for .NET mediante la procedura guidata di installazione

L'installazione utilizza un programma di installazione InstallShield X/Windows MSI. Sono disponibili due opzioni di configurazione, in modo che sia possibile scegliere un'installazione completa o personalizzata.

Informazioni su questa attività

Per installare Message Service Client for .NET su Windows, seguire questa procedura.

Procedura

1. Se si sta eseguendo l'installazione da un SupportPac completare la seguente procedura, altrimenti procedere direttamente al passo [“2” a pagina 10](#).
 - a) Su Windows, accedere come amministratore.
 - b) Eseguire il programma di installazione dotNETClientsetup.exe.
2. Attendere l'apertura della procedura guidata di installazione e visualizzare il seguente messaggio:

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

Fare clic su **Avanti**.

La procedura guidata potrebbe richiedere di leggere l'accordo di licenza.

3. Se viene richiesto di leggere l'accordo di licenze e si accettano i termini dell'accordo di licenze, fare clic su **Accetto i termini nell'accordo di licenze**, quindi fare clic su **Avanti**.

La procedura guidata di installazione richiede di selezionare il tipo di installazione più adatto alle proprie esigenze.

4. Selezionare il tipo di configurazione richiesto:

- Per installare tutte le funzioni del programma e installarle nella directory di installazione predefinita, fare clic su **Completa**.
- Per scegliere quali funzioni si desidera installare e specificare dove sono installate, fare clic su **Personalizzato**.

5. Fare clic su **Avanti**.

Se si seleziona l'opzione di installazione completa, la procedura guidata di installazione visualizza un messaggio che indica che è pronta per iniziare l'installazione, come descritto nel passo [“8” a pagina 11](#). Se si seleziona l'opzione di installazione personalizzata, la procedura guidata di installazione richiede di selezionare le funzioni che si desidera installare ed è necessario completare i passi [“6” a pagina 11](#) e [“7” a pagina 11](#) prima di passare al passo [“8” a pagina 11](#).

6. Solo per un'installazione personalizzata, fare clic su un'icona nell'elenco di funzionalità per specificare eventuali modifiche al modo in cui si desidera installare le funzionalità Message Service Client for .NET . Se non si desidera installare Message Service Client for .NET nella directory consigliata, scegliere un'altra directory.

Se si sceglie di installare Message Service Client for .NET in una directory che non esiste attualmente, la procedura guidata di installazione crea la directory automaticamente.

Se si desidera sviluppare applicazioni XMS , verificare che la funzione **Strumenti di sviluppo ed esempi** sia selezionata. Questa funzione fornisce le applicazioni di esempio e le librerie e qualsiasi altro file richiesto per compilare le applicazioni .NET . Se non si seleziona questa funzione, vengono installati solo i file richiesti per eseguire le applicazioni XMS .

7. Se si sta utilizzando l'opzione di installazione personalizzata, fare clic su **Avanti** dopo aver selezionato le opzioni richieste come descritto nel passo [“6” a pagina 11](#).

L'installazione guidata visualizza un messaggio che indica che è pronta per iniziare l'installazione.

8. Fare clic su **Installa** per avviare l'installazione.

La procedura guidata di installazione visualizza una barra che mostra l'avanzamento dell'installazione. Attendere il completamento della barra di avanzamento. Quando l'installazione viene completata correttamente, la finestra visualizza il messaggio seguente:

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

9. Fare clic su **Fine** per chiudere la procedura guidata di installazione.

Risultati

È stato installato correttamente Message Service Client for .NET, che è pronto per l'uso.

Operazioni successive

Prima di eseguire qualsiasi applicazione XMS , incluse le applicazioni di esempio fornite con XMS, è necessario impostare l'ambiente del server di messaggistica, per i dettagli consultare: [“Impostazione dell'ambiente del server di messaggistica” a pagina 13](#).

Concetti correlati

Servizio Web Ricerca JNDI

Per accedere a una directory di denominazione COS da XMS, un JNDI servizio Web di ricerca deve essere distribuito su un server WebSphere Application Server service integration bus . Questo servizio web converte le informazioni Java dal servizio di denominazione COS in un modulo che le applicazioni XMS possono leggere.

Impostazione dell'ambiente del server di messaggistica

Gli argomenti in questa sezione descrivono come impostare l'ambiente del server di messaggistica per consentire alle applicazioni di XMS di connettersi a un server.

Utilizzo di applicazioni di esempio XMS

Utilizzare le applicazioni di esempio fornite con XMS per verificare l'installazione e la configurazione del server di messaggistica e per creare le proprie applicazioni. Gli esempi forniscono una panoramica delle funzioni comuni di ciascuna API.

Prerequisiti per le applicazioni XMS che si collegano a WebSphere MQ

Alcuni prerequisiti si applicano se la tua applicazione XMS si connette a WebSphere MQ.

Per le applicazioni che si connettono a un gestore code WebSphere MQ , è necessario installare le librerie client WebSphere MQ appropriate sulla macchina utilizzata per eseguire l'applicazione XMS . Queste librerie sono preinstallate sulle macchine con un gestore code locale.

Per Client XMS per .NET, utilizzare le librerie client fornite con IBM WebSphere MQ 7.0.1.0 o successive. Queste sono le classi *WebSphere MQ per .NET*. Abilitano le connessioni in modalità client ai gestori code IBM WebSphere MQ 7.0, 6.0e 5.3 e le connessioni in modalità bind a un gestore code locale, se è anche IBM WebSphere MQ 7.0.1.0 o successivo.

Microsoft .NET Framework 2.0 Redistributable Package deve essere installato sul computer su cui deve essere installato XMS . Se questo pacchetto non è disponibile, l'installazione di XMS avrà esito negativo. È quindi necessario uscire dalla procedura di installazione, installare Microsoft .NET Framework 2.0 Redistributable Package sul computer ed eseguire nuovamente la procedura di installazione.

Sul sito di download Microsoft , è necessario cercare dotnetfx.exe per Microsoft .NET Framework 2.0 Redistributable Package (x86) e NetFx64.exe per Microsoft .NET Framework 2.0 Redistributable Package (x64), a seconda dei casi.

Concetti correlati

[Impostazione dell'ambiente del server di messaggistica](#)

Gli argomenti in questa sezione descrivono come impostare l'ambiente del server di messaggistica per consentire alle applicazioni di XMS di connettersi a un server.

Come impedire alle applicazioni di utilizzare la versione più recente di XMS

Per impostazione predefinita, quando è installata una versione più recente di XMS , le applicazioni che utilizzano la versione precedente passano automaticamente alla nuova versione senza dover ricompilare.

Informazioni su questa attività

La funzione di coesistenza di più versioni garantisce che l'installazione di una versione XMS più recente non sovrascriva la versione XMS precedente. Invece, più istanze di assembly XMS .NET simili coesistono in GAC (Global Assembly Cache), ma hanno numeri di versione diversi. Internamente, il GAC usa un file della politica per instradare le chiamate dell'applicazione all'ultima versione di XMS. Le applicazioni vengono eseguite senza necessità di ricompilazione e possono utilizzare nuove funzioni disponibili nella nuova versione di XMS .NET .

Tuttavia, se un'applicazione è richiesta per utilizzare la versione precedente di XMS , impostare l'attributo `publisherpolicy` su `no` nel file di configurazione dell'applicazione.

Nota: Un file di configurazione dell'applicazione è un file con un nome composto dal nome del programma eseguibile a cui il file si riferisce, con il suffisso `.config`. Ad esempio, il file di configurazione dell'applicazione per `text.exe` avrà il nome `text.exe.config`.

In qualsiasi momento, tuttavia, tutte le applicazioni di un sistema utilizzano la stessa versione di XMS .NET.

Impostazione dell'ambiente del server di messaggistica

Gli argomenti in questa sezione descrivono come impostare l'ambiente del server di messaggistica per consentire alle applicazioni di XMS di connettersi a un server.

Per le applicazioni che si connettono a un gestore code IBM MQ , è necessario il client IBM MQ (o il gestore code per la modalità di bind).

Attualmente non esistono prerequisiti per le applicazioni che utilizzano una connessione in tempo reale a un broker.

È necessario configurare l'ambiente del server di messaggistica prima di eseguire qualsiasi applicazione XMS , incluse le applicazioni di esempio fornite con XMS.

Questa sezione contiene i seguenti argomenti:

- [“Configurazione del gestore code e del broker per un'applicazione che si connette a un gestore code IBM MQ” a pagina 13](#)
- [“Configurazione di un broker per un'applicazione che utilizza una connessione in tempo reale a un broker” a pagina 15](#)
- [“Configurazione del service integration bus per un'applicazione che si connette a WebSphere Application Server” a pagina 16](#)

Concetti correlati

[Servizio Web Ricerca JNDI](#)

Per accedere a una directory di denominazione COS da XMS, un JNDI servizio Web di ricerca deve essere distribuito su un server WebSphere Application Server service integration bus . Questo servizio web converte le informazioni Java dal servizio di denominazione COS in un modulo che le applicazioni XMS possono leggere.

[Utilizzo di applicazioni di esempio XMS](#)

Utilizzare le applicazioni di esempio fornite con XMS per verificare l'installazione e la configurazione del server di messaggistica e per creare le proprie applicazioni. Gli esempi forniscono una panoramica delle funzioni comuni di ciascuna API.

Attività correlate

[Installazione di Message Service Client for .NET mediante la procedura guidata di installazione](#)

L'installazione utilizza un programma di installazione InstallShield X/Windows MSI. Sono disponibili due opzioni di configurazione, in modo che sia possibile scegliere un'installazione completa o personalizzata.

Riferimenti correlati

[Prerequisiti per le applicazioni XMS che si collegano a WebSphere MQ](#)

Alcuni prerequisiti si applicano se la tua applicazione XMS si connette a WebSphere MQ.

Configurazione del gestore code e del broker per un'applicazione che si connette a un gestore code IBM MQ

Questa sezione presuppone che si stia utilizzando IBM WebSphere MQ 7.0.1o versioni successive. Prima di poter eseguire un'applicazione che si connette ad un gestore code IBM MQ , è necessario configurarlo. Per un'applicazione di pubblicazione / sottoscrizione, è richiesta una configurazione aggiuntiva se si utilizza l'interfaccia di pubblicazione / sottoscrizione accodata.

Prima di iniziare

XMS funziona con IBM Integration Bus o WebSphere Message Broker 6.1 o versioni successive

Prima di avviare questa attività, effettuare le seguenti operazioni:

- Assicurarsi che l'applicazione abbia accesso a un gestore code in esecuzione.

- Se l'applicazione è un'applicazione di pubblicazione / sottoscrizione e utilizza l'interfaccia di pubblicazione / sottoscrizione accodata, assicurarsi che l'attributo "PSMODE" sia impostato su "ENABLED" sul gestore code.
- Verificare che l'applicazione utilizzi una factory di connessione le cui proprietà siano impostate in maniera appropriata per connettersi al gestore code. Se l'applicazione è un'applicazione di pubblicazione / sottoscrizione, assicurarsi che le proprietà di factory di connessione appropriate siano impostate per l'utilizzo del broker. Per ulteriori informazioni sulle proprietà di un factory di connessione, "[Proprietà di ConnectionFactory](#)" a pagina 181.

Informazioni su questa attività

Configurare il gestore code e il Broker per eseguire le applicazioni XMS nello stesso modo in cui si configura il Gestore code e l'interfaccia di pubblicazione / sottoscrizione accodata per eseguire le applicazioni IBM MQ JMS . La seguente procedura riepiloga le operazioni da effettuare.

Procedura

1. Sul Gestore code, creare le code necessarie all'applicazione.

Per una panoramica su come creare le code, consultare [Definizione delle code](#).

Se l'applicazione è un'applicazione di pubblicazione / sottoscrizione e utilizza l'interfaccia di pubblicazione / sottoscrizione accodata che necessita dell'accesso alle code di sistema IBM MQ classes for JMS , attendere il passo [4a](#) prima di creare le code.

2. Concedere all'ID utente associato alla propria applicazione l'autorità di connettersi al gestore code e l'autorizzazione appropriata per accedere alle code.

Per una panoramica sull'autorizzazione, consultare [Protezione](#). Se l'applicazione si connette al gestore code in modalità client, consultare anche [Client e server](#).

3. Se l'applicazione si connette al gestore code in modalità client, assicurarsi che un canale di connessione server sia definito nel gestore code e che un listener sia avviato.

Non è necessario eseguire questa procedura per ciascuna applicazione che si connette al gestore code. Una definizione di canale di connessione server e un listener possono supportare tutte le applicazioni che si collegano in modalità client.

4. Se l'applicazione è un'applicazione di pubblicazione / sottoscrizione e utilizza l'interfaccia di pubblicazione / sottoscrizione accodata, attenersi alla seguente procedura.

- a) Sul gestore code, creare le code del sistema IBM MQ classes for JMS eseguendo lo script dei comandi MQSC forniti con IBM MQ. Assicurarsi che l'ID utente associato a IBM Integration Bus o WebSphere Message Broker disponga dell'autorizzazione per accedere alle code.

Per informazioni su dove trovare lo script e su come eseguirlo, consultare [Utilizzo di IBM MQ classes for Java](#).

Eseguire questa operazione solo una volta per il gestore code. La stessa serie di code di sistema IBM MQ classes for JMS può supportare tutte le applicazioni XMS e IBM MQ classes for JMS che si connettono al gestore code.

- b) Concedere all'ID utente associato all'applicazione l'autorità di accedere alle code di sistema IBM MQ classes for JMS .

Per informazioni sulle autorizzazioni necessarie all'ID utente, consultare [Utilizzo di IBM MQ classes for JMS](#).

- c) Per un broker IBM Integration Bus o WebSphere Message Broker, creare e distribuire un flusso di messaggi per servire la coda in cui le applicazioni inviano i messaggi che pubblicano.

Il flusso di messaggi di base comprende un nodo di elaborazione messaggi MQInput per leggere i messaggi pubblicati e un nodo di elaborazione messaggi Publication per pubblicare i messaggi.

Per informazioni su come creare e distribuire un flusso di messaggi, consultare la documentazione del prodotto IBM Integration Bus o WebSphere Message Broker disponibile in [Pagina Web della libreria della documentazione del prodotto IBM Integration Bus](#).

Non è necessario eseguire questa operazione se un flusso di messaggi adatto è già distribuito nel broker.

Risultati

È ora possibile avviare l'applicazione.

Attività correlate

[Configurazione di un broker per un'applicazione che utilizza una connessione in tempo reale a un broker](#)
Prima di poter eseguire un'applicazione che utilizza una connessione in tempo reale a un broker, è necessario configurare tale broker.

[Configurazione del service integration bus per un'applicazione che si connette a WebSphere Application Server](#)

Prima di poter eseguire un'applicazione che si connette a un SIB (Service Integration Bus) WebSphere Application Server service integration technologies , è necessario configurare l'integrazione servizi nello stesso modo in cui si configura il SIB (Service Integration Bus) per eseguire le applicazioni JMS che utilizzano il provider di messaggistica predefinito.

Configurazione di un broker per un'applicazione che utilizza una connessione in tempo reale a un broker

Prima di poter eseguire un'applicazione che utilizza una connessione in tempo reale a un broker, è necessario configurare tale broker.

Prima di iniziare

Prima di avviare questa attività, effettuare le seguenti operazioni:

- Assicurarsi che l'applicazione abbia accesso a un broker in esecuzione.
- Assicurarsi che l'applicazione utilizzi una factory di connessione le cui proprietà siano impostate in maniera appropriata per una connessione in tempo reale a un broker. Per ulteriori informazioni sulle proprietà di una factory di connessione, consultare [“Proprietà di ConnectionFactory” a pagina 181](#).

Informazioni su questa attività

Configurare un broker per eseguire applicazioni XMS nello stesso modo in cui si configura un broker per eseguire applicazioni IBM MQ classes for JMS . La seguente procedura riepiloga le operazioni da effettuare:

Procedura

1. Creare e distribuire un flusso di messaggi per leggere i messaggi dalla porta TCP/IP su cui un broker è in ascolto e pubblicare i messaggi.

È possibile effettuare questa operazione in uno dei seguenti modi:

- Creare un flusso di messaggi che contiene un nodo di elaborazione messaggi **Real-timeOptimizedFlow** .
- Creare un flusso di messaggi che contenga un nodo di elaborazione messaggi **Real-timeInput** e un nodo di elaborazione messaggi Publication.

È necessario configurare il nodo **Real-timeOptimizedFlow** o **Real-timeInput** per essere in ascolto sulla porta utilizzata per le connessioni in tempo reale. In XMS, il numero di porta predefinito per le connessioni in tempo reale è 1506.

Non è necessario eseguire questa operazione se un flusso di messaggi adatto è già distribuito nel broker.

2. Se si desidera che i messaggi vengano consegnati all'applicazione utilizzando IBM MQ classes for JMS, configurare il broker per abilitare il multicast. Configurare gli argomenti che devono essere abilitati al

multicast, specificando una QoS (quality of service) affidabile per quegli argomenti che richiedono il multicast affidabile.

3. Se l'applicazione fornisce un ID utente e una password quando si connette al broker e si desidera che il broker autentichi l'applicazione utilizzando queste informazioni, configurare il server dei nomi utente e il broker per l'autenticazione della password di tipo telnet semplice.

Risultati

È ora possibile avviare l'applicazione.

Attività correlate

[Configurazione del gestore code e del broker per un'applicazione che si connette a un gestore code IBM MQ](#)

Questa sezione presuppone che si stia utilizzando IBM WebSphere MQ 7.0.1o versioni successive. Prima di poter eseguire un'applicazione che si connette ad un gestore code IBM MQ, è necessario configurarlo. Per un'applicazione di pubblicazione / sottoscrizione, è richiesta una configurazione aggiuntiva se si utilizza l'interfaccia di pubblicazione / sottoscrizione accodata.

[Configurazione del service integration bus per un'applicazione che si connette a WebSphere Application Server](#)

Prima di poter eseguire un'applicazione che si connette a un SIB (Service Integration Bus) WebSphere Application Server service integration technologies, è necessario configurare l'integrazione servizi nello stesso modo in cui si configura il SIB (Service Integration Bus) per eseguire le applicazioni JMS che utilizzano il provider di messaggistica predefinito.

Configurazione del service integration bus per un'applicazione che si connette a WebSphere Application Server

Prima di poter eseguire un'applicazione che si connette a un SIB (Service Integration Bus) WebSphere Application Server service integration technologies, è necessario configurare l'integrazione servizi nello stesso modo in cui si configura il SIB (Service Integration Bus) per eseguire le applicazioni JMS che utilizzano il provider di messaggistica predefinito.

Prima di iniziare

Prima di avviare questa attività, è necessario effettuare le seguenti operazioni:

- Verificare che sia creato un bus di messaggistica e che il proprio server sia stato aggiunto al bus come membro del bus.
- Assicurarsi che l'applicazione abbia accesso a un SIB (Service Integration Bus) che contiene almeno un motore di messaggistica in esecuzione.
- Se è richiesta l'operazione HTTP, è necessario definire un canale di trasporto in entrata del motore di messaggistica HTTP. Per impostazione predefinita, i canali per SSL e TCP vengono definiti durante l'installazione del server.
- Verificare che l'applicazione utilizzi una factory di connessione le cui proprietà siano impostate in maniera appropriata per connettersi al SIB (Service Integration Bus) utilizzando un server di avvio. Le informazioni minime richieste sono:
 - L'endpoint del provider, che descrive l'ubicazione e il protocollo da utilizzare quando si negozia una connessione al server di messaggistica (ossia, tramite il server di avvio). Nella sua forma più semplice, per un server installato con le impostazioni predefinite, l'endpoint di fornitura può essere impostato sul nome host del server.
 - Il nome del bus attraverso cui vengono inviati i messaggi.

Per ulteriori informazioni sulle proprietà di una factory di connessione, consultare [“Proprietà di ConnectionFactory” a pagina 181](#).

Informazioni su questa attività

Qualsiasi coda o spazio argomento richiesto deve essere definito. Per impostazione predefinita, uno spazio argomento denominato Default.Topic.Space viene definito durante l'installazione del server ma, se sono necessari ulteriori spazi argomento, è necessario creare tali spazi. Non è necessario predefinire singoli argomenti all'interno di uno spazio argomento, poiché il server crea un'istanza di questi singoli argomenti in modo dinamico come richiesto.

La seguente procedura riepiloga le operazioni da effettuare.

Procedura

1. Creare le code necessarie all'applicazione per la messaggistica point - to - point.
2. Creare eventuali spazi argomenti aggiuntivi necessari all'applicazione per la messaggistica di pubblicazione / sottoscrizione.

Risultati

È ora possibile avviare l'applicazione.

Attività correlate

Configurazione del gestore code e del broker per un'applicazione che si connette a un gestore code IBM MQ

Questa sezione presuppone che si stia utilizzando IBM WebSphere MQ 7.0.1o versioni successive. Prima di poter eseguire un'applicazione che si connette ad un gestore code IBM MQ , è necessario configurarlo. Per un'applicazione di pubblicazione / sottoscrizione, è richiesta una configurazione aggiuntiva se si utilizza l'interfaccia di pubblicazione / sottoscrizione accodata.

Configurazione di un broker per un'applicazione che utilizza una connessione in tempo reale a un broker
Prima di poter eseguire un'applicazione che utilizza una connessione in tempo reale a un broker, è necessario configurare tale broker.

Utilizzo di applicazioni di esempio XMS

Utilizzare le applicazioni di esempio fornite con XMS per verificare l'installazione e la configurazione del server di messaggistica e per creare le proprie applicazioni. Gli esempi forniscono una panoramica delle funzioni comuni di ciascuna API.

Concetti correlati

Servizio Web Ricerca JNDI

Per accedere a una directory di denominazione COS da XMS, un JNDI servizio Web di ricerca deve essere distribuito su un server WebSphere Application Server service integration bus . Questo servizio web converte le informazioni Java dal servizio di denominazione COS in un modulo che le applicazioni XMS possono leggere.

Impostazione dell'ambiente del server di messaggistica

Gli argomenti in questa sezione descrivono come impostare l'ambiente del server di messaggistica per consentire alle applicazioni di XMS di connettersi a un server.

Attività correlate

Installazione di Message Service Client for .NET mediante la procedura guidata di installazione

L'installazione utilizza un programma di installazione InstallShield X/Windows MSI. Sono disponibili due opzioni di configurazione, in modo che sia possibile scegliere un'installazione completa o personalizzata.

Le applicazioni di esempio

Le applicazioni di esempio forniscono una panoramica delle funzioni comuni di ciascuna API. È possibile utilizzarli per verificare l'installazione e l'impostazione del server di messaggistica e per creare le proprie applicazioni.

Se si ha bisogno di aiuto per creare le proprie applicazioni, è possibile utilizzare le applicazioni di esempio come punto di partenza. Sia l'origine che una versione compilata vengono fornite per ciascuna

applicazione. Esaminare il codice di origine di esempio e identificare i passi chiave per creare ciascun oggetto richiesto per l'applicazione (ConnectionFactory, Connection, Session, Destination e un Producer o un Consumer o entrambi) e per impostare le proprietà specifiche necessarie per specificare il funzionamento dell'applicazione. Per ulteriori informazioni, fare riferimento a [“Scrittura di applicazioni XMS”](#) a pagina 20. Gli esempi sono soggetti a modifiche nelle release future di XMS.

La seguente tabella mostra le tre serie di applicazioni di esempio (una per ciascuna API) fornite con XMS.

Nome del campione	Descrizione
SampleConsumerCS	Un'applicazione del consumatore di messaggi che prende i messaggi da una coda o sottoscrive un argomento.
SampleProducerCS	Un'applicazione del produttore di messaggi che produce messaggi su una coda o su un argomento.
SampleConfigCS	Un'applicazione di configurazione che è possibile utilizzare per creare un repository di oggetti gestiti basato su file. L'applicazione contiene una produzione connessioni e una destinazione per le specifiche impostazioni di connessione. Questo repository di oggetti gestiti può quindi essere utilizzato con ciascuna delle applicazioni consumer e producer di esempio.

Gli esempi che supportano le stesse funzioni nelle varie API presentano differenze sintattiche.

- Le applicazioni consumer e producer di messaggi di esempio supportano entrambe le seguenti funzioni:
 - Le connessioni a IBM MQ, IBM Integration Bus (utilizzando una connessione in tempo reale a un broker) e un WebSphere Application Server service integration bus
 - Ricerche di repository di oggetti amministrati utilizzando l'interfaccia di contesto iniziale
 - Connessioni a code (IBM MQ e WebSphere Application Server service integration bus) e argomenti (IBM MQ, connessione in tempo reale a broker e WebSphere Application Server service integration bus)
 - Messaggi di base, byte, mappa, oggetto, flusso e testo
- L'applicazione consumatore di messaggi di esempio supporta le modalità di ricezione sincrone e asincrona e le istruzioni SQL Selector.
- L'applicazione produttore di messaggi di esempio supporta le modalità di consegna persistenti e non persistenti.

Modalità operative

Gli esempi possono funzionare in uno dei due modi:

Modalità semplice

È possibile eseguire gli esempi con l'input utente minimo.

Modalità avanzata

È possibile personalizzare più finemente il modo in cui operano gli esempi.

Tutti gli esempi sono compatibili e possono quindi operare in più lingue.

Concetti correlati

[Creazione di applicazioni personalizzate](#)

Si creano le proprie applicazioni come si creano le applicazioni di esempio.

Attività correlate

[Esecuzione delle applicazioni di esempio](#)

È possibile eseguire le applicazioni di esempio .NET in modo interattivo in modalità semplice o avanzata oppure in modo non interattivo utilizzando file di risposta personalizzati o generati automaticamente.

[Creazione delle applicazioni di esempio .NET](#)

Quando si crea un'applicazione .NET di esempio, viene creata una versione eseguibile dell'esempio scelto.

Esecuzione delle applicazioni di esempio

È possibile eseguire le applicazioni di esempio .NET in modo interattivo in modalità semplice o avanzata oppure in modo non interattivo utilizzando file di risposta personalizzati o generati automaticamente.

Prima di iniziare

Prima di eseguire una delle applicazioni di esempio fornite, è necessario impostare l'ambiente del server di messaggistica in modo che le applicazioni possano connettersi a un server. Vedere [“Impostazione dell'ambiente del server di messaggistica”](#) a pagina 13.

Procedura

Per eseguire un'applicazione di esempio .NET, completare la seguente procedura:

Suggerimento: Quando si sta eseguendo un'applicazione di esempio, immettere? in qualsiasi momento per ottenere aiuto su cosa fare in seguito.

1. Selezionare la modalità in cui si desidera eseguire l'applicazione di esempio.

Immettere `Advanced` o `Simple`.

2. Rispondere alle domande.

Per selezionare il valore predefinito, visualizzato tra parentesi alla fine della domanda, premere `Invio`. Per selezionare un valore diverso, immettere il valore appropriato e premere `Invio`.

Ecco una domanda di esempio:

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

In questo caso, il valore predefinito è `wpm` (connessione a un WebSphere Application Server service integration bus).

Risultati

Quando si eseguono le applicazioni di esempio, i file di risposta vengono generati automaticamente nella directory di lavoro corrente. I nomi dei file di risposta sono nel formato `connection_type-sample_type.rsp`; ad esempio, `wpm-producer.rsp`. Se necessario, è possibile utilizzare il file di risposta generato per rieseguire l'applicazione di esempio con le stesse opzioni, in modo da non dover immettere nuovamente le opzioni.

Concetti correlati

[Le applicazioni di esempio](#)

Le applicazioni di esempio forniscono una panoramica delle funzioni comuni di ciascuna API. È possibile utilizzarli per verificare l'installazione e l'impostazione del server di messaggistica e per creare le proprie applicazioni.

Attività correlate

[Creazione delle applicazioni di esempio .NET](#)

Quando si crea un'applicazione .NET di esempio, viene creata una versione eseguibile dell'esempio scelto.

Creazione delle applicazioni di esempio .NET

Quando si crea un'applicazione .NET di esempio, viene creata una versione eseguibile dell'esempio scelto.

Prima di iniziare

Installare il compilatore appropriato. Questa attività presuppone che Microsoft Visual Studio 2012 sia installato e che si abbia familiarità con l'utilizzo.

Procedura

Per creare un'applicazione di esempio .NET , completare la seguente procedura:

1. Fare clic sul file della soluzione `Samples.sln` fornito con gli esempi .NET .
2. Fare clic con il tasto destro del mouse sulla soluzione `Esempi` nella finestra `Esplora soluzioni` e selezionare **Crea soluzione**.

Risultati

Un programma eseguibile viene creato nella sottocartella appropriata dell'esempio, `bin/Debug` o `bin/Release`, a seconda della configurazione scelta. Questo programma ha lo stesso nome della cartella, con suffisso `CS`. Ad esempio, se si sta creando la versione C# dell'applicazione di esempio del produttore del messaggio, `SampleProducerCS.exe` viene creato nella cartella `SampleProducer` .

Concetti correlati

[Le applicazioni di esempio](#)

Le applicazioni di esempio forniscono una panoramica delle funzioni comuni di ciascuna API. È possibile utilizzarli per verificare l'installazione e l'impostazione del server di messaggistica e per creare le proprie applicazioni.

[“Creazione di applicazioni personalizzate” a pagina 43](#)

Si creano le proprie applicazioni come si creano le applicazioni di esempio.

Attività correlate

[Esecuzione delle applicazioni di esempio](#)

È possibile eseguire le applicazioni di esempio .NET in modo interattivo in modalità semplice o avanzata oppure in modo non interattivo utilizzando file di risposta personalizzati o generati automaticamente.

Sviluppo di applicazioni XMS

Gli argomenti in questa sezione forniscono informazioni utili per la scrittura di applicazioni XMS .

Per informazioni sulla scrittura di applicazioni XMS , fare riferimento ai seguenti argomenti:

Scrittura di applicazioni XMS

Gli argomenti in questa sezione forniscono informazioni utili per la scrittura di applicazioni XMS .

Questa sezione contiene concetti generali per la scrittura di applicazioni XMS . Consultare anche [“Scrittura di applicazioni XMS .NET” a pagina 44](#) per informazioni specifiche sulla creazione di applicazioni .NET .

Questa sezione contiene i seguenti argomenti:

- [“Il modello di threading” a pagina 21](#)
- [“ConnectionFactoryes e oggetti Connection” a pagina 21](#)
- [“Sessioni” a pagina 24](#)
- [“Destinazioni” a pagina 28](#)
- [“Produttori di messaggi” a pagina 33](#)
- [“Utenti del messaggio” a pagina 34](#)
- [“Browser coda” a pagina 38](#)
- [“Richiedenti” a pagina 38](#)
- [“Eliminazione oggetto” a pagina 38](#)
- [“XMS Tipi primitivi” a pagina 39](#)
- [“Conversione implicita di un valore di proprietà da un tipo di dati a un altro” a pagina 40](#)
- [“Iteratori” a pagina 43](#)
- [“CCSID \(Coded character set identifier\)” a pagina 43](#)

- [“Codici di errore e di eccezione XMS” a pagina 43](#)
- [“Creazione di applicazioni personalizzate” a pagina 43](#)

Concetti correlati

[Scrittura di applicazioni XMS .NET](#)

Gli argomenti contenuti in questa sezione forniscono informazioni utili per la scrittura di applicazioni XMS .NET .

Riferimenti correlati

[.NETInterfacce](#)

Questa sezione documenta le interfacce della classe .NET e le relative proprietà e metodi.

Il modello di threading

Le regole generali gestiscono il modo in cui un'applicazione a più thread può utilizzare gli oggetti XMS .

- Solo gli oggetti dei seguenti tipi possono essere utilizzati contemporaneamente su thread differenti:
 - ConnectionFactory
 - Connessione
 - Dati ConnectionMeta
 - Destination
- Un oggetto Session può essere utilizzato solo su un singolo thread alla volta.

Le eccezioni a tali regole sono indicate da voci etichettate "Contesto thread" nelle definizioni di interfaccia dei metodi in [“Riferimento per Message Service Client for .NET” a pagina 90](#).

Concetti correlati

[Condizioni di errore che possono essere gestite al runtime](#)

I codici di ritorno dalle chiamate API sono condizioni di errore che possono essere gestite al runtime. Il modo in cui si affronta questo tipo di errore dipende dal fatto che si stia utilizzando l'API C o C++ .

ConnectionFactoryies e oggetti Connection

Un oggetto ConnectionFactory fornisce un template che un'applicazione utilizza per creare un oggetto Connection. L'applicazione utilizza l'oggetto Connection per creare un oggetto Session.

Per .NET, l'applicazione XMS utilizza innanzitutto un oggetto XMSFactoryFactory per ottenere un riferimento a un oggetto ConnectionFactory appropriato al tipo di protocollo richiesto. Questo ConnectionFactory può quindi produrre connessioni solo per quel tipo di protocollo.

Un'applicazione XMS può creare più connessioni e un'applicazione a più thread può utilizzare un singolo oggetto Connection contemporaneamente su più thread. Un oggetto Connection incapsula una connessione di comunicazione tra un'applicazione e un server di messaggistica.

Una connessione ha diversi scopi:

- Quando un'applicazione crea una connessione, è possibile autenticarla.
- Un'applicazione può associare un identificativo client univoco ad una connessione. L'identificativo client viene utilizzato per supportare sottoscrizioni durevoli nel dominio di pubblicazione / sottoscrizione. L'identificativo client può essere impostato in due modi:

Il modo preferito per assegnare un identificatore client di connessioni è quello di configurare un oggetto ConnectionFactory specifico del client utilizzando le proprietà e assegnarlo in modo trasparente alla connessione che crea.

Un modo alternativo di assegnare un identificatore client consiste nell'utilizzare un valore specifico del provider impostato sull'oggetto Connection. Questo valore non sovrascrive l'identificativo che è stato configurato amministrativamente. Viene fornito per il caso in cui non esiste alcun identificativo specificato amministrativamente. Se un identificativo specificato amministrativamente esiste, un tentativo di sovrascriverlo con un valore specifico del provider causa l'emissione di un'eccezione. Se un'applicazione imposta esplicitamente un identificativo, deve farlo immediatamente dopo la creazione

della connessione e prima che venga eseguita qualsiasi altra operazione sulla connessione; altrimenti, viene generata un'eccezione.

Un'applicazione XMS in genere crea una connessione, una o più sessioni e un numero di produttori di messaggi e di utenti di messaggi.

La creazione di una connessione è relativamente dispendiosa in termini di risorse di sistema poiché implica la creazione di una connessione di comunicazione e potrebbe anche comportare l'autenticazione dell'applicazione.

Attività correlate

Creazione di oggetti gestiti

Le definizioni dell'oggetto `ConnectionFactory` e `Destination` richieste dalle applicazioni XMS per stabilire una connessione a un server di messaggistica devono essere create utilizzando gli strumenti di gestione appropriati.

Riferimenti correlati

`IConnectionFactory` (per l'interfaccia .NET)

Un'applicazione utilizza una produzione connessioni per creare una connessione.

Proprietà di `ConnectionFactory`

Una panoramica delle proprietà dell'oggetto `ConnectionFactory`, con link a informazioni di riferimento più dettagliate.

`IDestination` (per l'interfaccia .NET)

Una destinazione è il punto in cui un'applicazione invia messaggi o è un'origine da cui un'applicazione riceve messaggi o entrambi.

Proprietà della destinazione

Una panoramica delle proprietà dell'oggetto `Destinazione`, con link a informazioni di riferimento più dettagliate.

Modalità di connessione avviata e arrestata

Una connessione può operare in modalità avviata o arrestata.

Quando un'applicazione crea una connessione, la connessione è in modalità arrestata. Quando la connessione è in modalità arrestata, l'applicazione può inizializzare le sessioni e può inviare messaggi ma non può riceverli, in modo sincrono o asincrono.

Un'applicazione può avviare una connessione richiamando il metodo `Start Connection`. Quando la connessione è in modalità avviata, l'applicazione può inviare e ricevere messaggi. L'applicazione può quindi arrestare e riavviare la connessione richiamando i metodi `Arresta connessione` e `Start Connection`.

Concetti correlati

Chiusura connessione

Un'applicazione chiude una connessione richiamando il metodo `Chiudi connessione`.

Gestione eccezione

Se un'applicazione utilizza una connessione solo per utilizzare i messaggi in modo asincrono, apprende un problema con la connessione solo utilizzando un listener di eccezioni.

Connessione a un SIB (Service Integration Bus)

Un'applicazione XMS può connettersi a un SIB (Service Integration Bus) WebSphere Application Server utilizzando una connessione TCP/IP diretta o utilizzando HTTP su TCP/IP.

Chiusura connessione

Un'applicazione chiude una connessione richiamando il metodo `Chiudi connessione`.

Quando un'applicazione chiude una connessione, XMS esegue le seguenti operazioni:

- Chiude tutte le sessioni associate alla connessione ed elimina alcuni oggetti associati a queste sessioni. Per ulteriori informazioni su quali oggetti vengono eliminati, consultare [“Eliminazione oggetto”](#) a pagina

38. Allo stesso tempo, XMS esegue il rollback di tutte le transazioni attualmente in corso all'interno delle sessioni.

- Termina la connessione di comunicazione con il server di messaggi.
- Rilascia la memoria e le altre risorse interne utilizzate dalla connessione.

XMS non riconosce la ricezione di alcun messaggio che non è stato possibile confermare durante una sessione, prima di chiudere la connessione. Per ulteriori informazioni sul riconoscimento della ricezione dei messaggi, consultare [“Riconoscimento messaggio”](#) a pagina 25.

Concetti correlati

Modalità di connessione avviata e arrestata

Una connessione può operare in modalità avviata o arrestata.

Gestione eccezione

Se un'applicazione utilizza una connessione solo per utilizzare i messaggi in modo asincrono, apprende un problema con la connessione solo utilizzando un listener di eccezioni.

Connessione a un SIB (Service Integration Bus)

Un'applicazione XMS può connettersi a un SIB (Service Integration Bus) WebSphere Application Server utilizzando una connessione TCP/IP diretta o utilizzando HTTP su TCP/IP.

Gestione eccezione

Se un'applicazione utilizza una connessione solo per utilizzare i messaggi in modo asincrono, apprende un problema con la connessione solo utilizzando un listener di eccezioni.

Le eccezioni XMS .NET sono tutte derivate da System.Exception. Per ulteriori informazioni, fare riferimento a [“Gestione degli errori in .NET”](#) a pagina 48.

Concetti correlati

Modalità di connessione avviata e arrestata

Una connessione può operare in modalità avviata o arrestata.

Chiusura connessione

Un'applicazione chiude una connessione richiamando il metodo Chiudi connessione.

Connessione a un SIB (Service Integration Bus)

Un'applicazione XMS può connettersi a un SIB (Service Integration Bus) WebSphere Application Server utilizzando una connessione TCP/IP diretta o utilizzando HTTP su TCP/IP.

Connessione a un SIB (Service Integration Bus)

Un'applicazione XMS può connettersi a un SIB (Service Integration Bus) WebSphere Application Server utilizzando una connessione TCP/IP diretta o utilizzando HTTP su TCP/IP.

Il protocollo HTTP può essere utilizzato in situazioni in cui non è possibile una connessione TCP/IP diretta. Una situazione comune è quando si comunica attraverso un firewall, ad esempio quando due aziende si scambiano messaggi. L'utilizzo di HTTP per comunicare tramite firewall viene spesso definito *tunneling HTTP*. Il tunneling HTTP, tuttavia, è intrinsecamente più lento rispetto all'utilizzo di una connessione TCP/IP diretta, poiché le intestazioni HTTP si aggiungono in modo significativo alla quantità di dati che vengono trasferiti e poiché il protocollo HTTP richiede più flussi di comunicazione rispetto a TCP/IP.

Per creare una connessione TCP/IP, è possibile che un'applicazione utilizzi un factory di connessione la cui proprietà `XMSC_WPM_TARGET_TRANSPORT_CHAIN` è impostata su `XMSC_WPM_TARGET_TRANSPORT_CHAIN_BASIC`. Questo è il valore predefinito della proprietà. Se la connessione viene creata correttamente, la proprietà `XMSC_WPM_CONNECTION_PROTOCOL` della connessione è impostata su `XMSC_WPM_CP_TCP`.

Per creare una connessione che utilizzi HTTP, è necessario che un'applicazione utilizzi un factory di connessione la cui proprietà `XMSC_WPM_TARGET_TRANSPORT_CHAIN` è impostata sul nome di una catena di trasporto in entrata, configurata per utilizzare un canale di trasporto HTTP. Se la connessione è stata creata correttamente, la proprietà `XMSC_WPM_CONNECTION_PROTOCOL` della connessione è impostata su `XMSC_WPM_CP_HTTP`. Per informazioni su come configurare le catene di trasporto,

consultare [Configurazione delle catene di trasporto](#) nella documentazione di WebSphere Application Server .

Un'applicazione ha una scelta simile di protocolli di comunicazione quando si connette a un server di avvio. La proprietà `XMSC_WPM_PROVIDER_ENDPOINTS` di una factory di connessione è una sequenza di uno o più indirizzi endpoint dei server bootstrap. Il componente della catena di trasporto bootstrap di ciascun indirizzo endpoint può essere `XMSC_WPM_BOOTSTRAP_TCP`, per una connessione TCP/IP a un server bootstrap o `XMSC_WPM_BOOTSTRAP_HTTP`, per una connessione che utilizza HTTP.

Concetti correlati

[Modalità di connessione avviata e arrestata](#)

Una connessione può operare in modalità avviata o arrestata.

[Chiusura connessione](#)

Un'applicazione chiude una connessione richiamando il metodo `Chiudi connessione`.

[Gestione eccezione](#)

Se un'applicazione utilizza una connessione solo per utilizzare i messaggi in modo asincrono, apprende un problema con la connessione solo utilizzando un listener di eccezioni.

Attività correlate

[Creazione di oggetti gestiti](#)

Le definizioni dell'oggetto `ConnectionFactory` e `Destination` richieste dalle applicazioni XMS per stabilire una connessione a un server di messaggistica devono essere create utilizzando gli strumenti di gestione appropriati.

Riferimenti correlati

[IConnectionFactory \(per l'interfaccia .NET\)](#)

Un'applicazione utilizza una produzione connessioni per creare una connessione.

[Proprietà di ConnectionFactory](#)

Una panoramica delle proprietà dell'oggetto `ConnectionFactory` , con link a informazioni di riferimento più dettagliate.

[IDestination \(per l'interfaccia .NET\)](#)

Una destinazione è il punto in cui un'applicazione invia messaggi o è un'origine da cui un'applicazione riceve messaggi o entrambi.

[Proprietà della destinazione](#)

Una panoramica delle proprietà dell'oggetto `Destinazione`, con link a informazioni di riferimento più dettagliate.

Sessioni

Una sessione è un contesto a thread singolo per inviare e ricevere messaggi.

Un'applicazione può utilizzare una sessione per creare messaggi, produttori di messaggi, utenti di messaggi, browser di code e destinazioni temporanee. Un'applicazione può anche utilizzare una sessione per eseguire transazioni locali.

Un'applicazione può creare più sessioni, in cui ogni sessione produce e consuma messaggi indipendentemente dalle altre sessioni. Se due utenti di messaggi in sessioni separate (o anche nella stessa sessione) sottoscrivono lo stesso argomento, ciascuno riceve una copia di qualsiasi messaggio pubblicato su tale argomento.

A differenza di un oggetto `Connection`, un oggetto `Session` non può essere utilizzato contemporaneamente su thread differenti. Solo il metodo di chiusura sessione di un oggetto `Session` può essere richiamato da un thread diverso da quello che l'oggetto `Session` sta utilizzando al momento. Il metodo `Chiudi sessione` termina una sessione e rilascia tutte le risorse di sistema assegnate alla sessione.

Se un'applicazione deve elaborare i messaggi contemporaneamente su più di un thread, l'applicazione deve creare una sessione su ciascun thread e quindi utilizzare tale sessione per qualsiasi operazione di invio o ricezione all'interno di tale thread.

Sessioni transazionali

Le applicazioni XMS possono eseguire transazioni locali. Una *transazione locale* è una transazione che implica modifiche solo alle risorse del gestore code o del service integration bus a cui è connessa l'applicazione.

Le informazioni contenute in questo argomento sono rilevanti solo se un'applicazione si connette a un gestore code IBM MQ o a un SIB (Service Integration Bus) WebSphere Application Server . Le informazioni non sono rilevanti per una connessione in tempo reale a un broker.

Per eseguire transazioni locali, un'applicazione deve prima creare una sessione sottoposta a transazione richiamando il metodo di creazione sessione di un oggetto Connection, specificando come parametro che la sessione viene sottoposta a transazione. Successivamente, tutti i messaggi inviati e ricevuti all'interno della sessione vengono raggruppati in una sequenza di transazioni. Una transazione termina quando l'applicazione esegue il commit o il rollback dei messaggi inviati e ricevuti dall'inizio della transazione.

Per eseguire il commit di una transazione, un'applicazione richiama il metodo Commit dell'oggetto Session. Quando viene eseguito il commit di una transazione, tutti i messaggi inviati all'interno della transazione diventano disponibili per il recapito ad altre applicazioni e tutti i messaggi ricevuti all'interno della transazione vengono riconosciuti in modo che il server di messaggistica non tenti di consegnarli nuovamente all'applicazione. Nel dominio point - to - point, il server di messaggistica rimuove anche i messaggi ricevuti dalle code.

Per eseguire il rollback di una transazione, l'applicazione richiama il metodo di rollback dell'oggetto Session. Quando viene eseguito il rollback di una transazione, tutti i messaggi inviati all'interno della transazione vengono eliminati dal server di messaggistica e tutti i messaggi ricevuti all'interno della transazione diventano nuovamente disponibili per la consegna. Nel dominio point - to - point, i messaggi ricevuti vengono reinseriti nelle loro code e diventano nuovamente visibili ad altre applicazioni.

Una nuova transazione viene avviata automaticamente quando un'applicazione crea una sessione sottoposta a transazione o richiama il metodo Commit o Rollback. Pertanto, una sessione transazionale ha sempre una transazione attiva.

Quando un'applicazione chiude una sessione transata, si verifica un rollback implicito. Quando un'applicazione chiude una connessione, si verifica un rollback implicito per tutte le sessioni transazionali della connessione.

Una transazione è interamente contenuta all'interno di una sessione transata. Una transazione non può estendere le sessioni. Ciò significa che non è possibile per un'applicazione inviare e ricevere messaggi in due o più sessioni transazionali e quindi eseguire il commit o il rollback di tutte queste azioni come una singola transazione.

Concetti correlati

Riconoscimento messaggio

Ogni sessione non sottoposta a transazione dispone di una modalità di riconoscimento che determina il modo in cui vengono riconosciuti i messaggi ricevuti dall'applicazione. Sono disponibili tre modalità di riconoscimento e la scelta della modalità di riconoscimento influisce sulla progettazione dell'applicazione.

Consegna di messaggi asincroni

XMS utilizza un thread per gestire tutte le consegne di messaggi asincroni per una sessione. Ciò significa che è possibile eseguire solo una funzione listener dei messaggi o un metodo onMessage () alla volta.

Consegna sincrona dei messaggi

I messaggi vengono consegnati in modo sincrono a un'applicazione se l'applicazione utilizza i metodi di ricezione degli oggetti MessageConsumer .

Modalità consegna messaggi

XMS supporta due modalità di consegna dei messaggi.

Riconoscimento messaggio

Ogni sessione non sottoposta a transazione dispone di una modalità di riconoscimento che determina il modo in cui vengono riconosciuti i messaggi ricevuti dall'applicazione. Sono disponibili tre modalità di riconoscimento e la scelta della modalità di riconoscimento influisce sulla progettazione dell'applicazione.

Le informazioni contenute in questo argomento sono rilevanti solo se un'applicazione si connette a un gestore code IBM MQ o a un SIB (Service Integration Bus) WebSphere Application Server . Le informazioni non sono rilevanti per una connessione in tempo reale a un broker.

XMS utilizza lo stesso meccanismo per confermare la ricezione dei messaggi utilizzati da JMS.

Se non viene eseguita la transazione di una sessione, il modo in cui vengono riconosciuti i messaggi ricevuti dall'applicazione è determinato dalla modalità di riconoscimento della sessione. Le tre modalità di riconoscimento sono descritte nei seguenti paragrafi:

RICONOSCIUTO_AUTO_XMSC_

La sessione riconosce automaticamente ogni messaggio ricevuto dall'applicazione.

Se i messaggi vengono consegnati in modo sincrono all'applicazione, la sessione conferma la ricezione di un messaggio ogni volta che una chiamata di ricezione viene completata correttamente.

Se l'applicazione riceve correttamente un messaggio, ma un errore impedisce il verificarsi del riconoscimento, il messaggio diventa nuovamente disponibile per la consegna. L'applicazione deve quindi essere in grado di gestire un messaggio che viene riconsegnato.

XMSC_DUPS_OK_RICONOSCIMENTO

La sessione riconosce i messaggi ricevuti dall'applicazione nel momento in cui viene selezionata.

L'utilizzo di questa modalità di riconoscimento riduce la quantità di lavoro che la sessione deve eseguire, ma un malfunzionamento che impedisce la conferma di ricezione del messaggio potrebbe causare la ridisponibilità di più di un messaggio per la consegna. L'applicazione deve quindi essere in grado di gestire i messaggi riconsegnati.

CONFERMA_CLIENT_XMSC_XX_ENCODE_CASE_CAPS_LOCK_OFF

L'applicazione riconosce i messaggi ricevuti richiamando il metodo Acknowledge della classe Message.

L'applicazione può confermare la ricezione di ciascun messaggio singolarmente oppure può ricevere un batch di messaggi e richiamare il metodo Acknowledge solo per l'ultimo messaggio ricevuto. Quando il metodo di riconoscimento viene richiamato, vengono riconosciuti tutti i messaggi ricevuti dall'ultima volta che il metodo è stato richiamato.

Insieme a una di queste modalità di riconoscimento, un'applicazione può arrestare e riavviare la consegna dei messaggi in una sessione richiamando il metodo Recover della classe Session. I messaggi la cui ricezione non era stata precedentemente confermata vengono riconsegnati. Tuttavia, potrebbero non essere consegnati nella stessa sequenza in cui sono stati precedentemente consegnati. Nel frattempo, potrebbero essere arrivati messaggi con priorità più elevata e alcuni dei messaggi originali potrebbero essere scaduti. Nel dominio point - to - point, alcuni dei messaggi originali potrebbero essere stati utilizzati da un'altra applicazione.

Un'applicazione può determinare se un messaggio viene riconsegnato esaminando il contenuto del campo di intestazione JMSRefornta del messaggio. L'applicazione esegue questa operazione richiamando il metodo Get JMSRenei della classe Message.

Concetti correlati

Sessioni transazionali

Le applicazioni XMS possono eseguire transazioni locali. Una *transazione locale* è una transazione che implica modifiche solo alle risorse del gestore code o del service integration bus a cui è connessa l'applicazione.

Consegna di messaggi asincroni

XMS utilizza un thread per gestire tutte le consegne di messaggi asincroni per una sessione. Ciò significa che è possibile eseguire solo una funzione listener dei messaggi o un metodo onMessage () alla volta.

Consegna sincrona dei messaggi

I messaggi vengono consegnati in modo sincrono a un'applicazione se l'applicazione utilizza i metodi di ricezione degli oggetti MessageConsumer .

Modalità consegna messaggi

XMS supporta due modalità di consegna dei messaggi.

Consegna di messaggi asincroni

XMS utilizza un thread per gestire tutte le consegne di messaggi asincroni per una sessione. Ciò significa che è possibile eseguire solo una funzione listener dei messaggi o un metodo `onMessage()` alla volta.

Se più di un consumatore di messaggi in una sessione sta ricevendo messaggi in modo asincrono e una funzione del listener di messaggi o un metodo `onMessage()` sta consegnando un messaggio a un consumatore di messaggi, qualsiasi altro consumatore di messaggi che è in attesa dello stesso messaggio deve continuare ad attendere. Anche altri messaggi in attesa di essere consegnati alla sessione devono continuare ad attendere.

Se un'applicazione richiede la consegna simultanea dei messaggi, creare più di una sessione in modo che XMS utilizzi più di un thread per gestire la consegna asincrona dei messaggi. In questo modo, è possibile eseguire contemporaneamente più di una funzione listener di messaggi o di un metodo `onMessage()`.

Una sessione non viene resa asincrona assegnando un listener di messaggi a un consumer. Una sessione diventa asincrona solo quando viene chiamato il metodo `Connection.Start`. Tutte le chiamate sincrone sono consentite fino a quando non viene richiamato il metodo `Connection.Start`. La consegna del messaggio ai consumer inizia quando viene richiamato il `Connection.Start`.

Se le chiamate sincrone, come la creazione di un consumer o di un producer, devono essere effettuate su una sessione asincrona, è necessario richiamare `Connection.Stop`. È possibile riprendere una sessione richiamando il metodo `Connection.Start` per avviare la consegna dei messaggi. L'unica eccezione è il thread di consegna dei messaggi di sessione, che è il thread che consegna i messaggi alla funzione di callback. Questo thread può effettuare qualsiasi chiamata sulla sessione (tranne una chiamata di chiusura) nella funzione di callback del messaggio.

Nota: In modalità non gestita, la chiamata MQDISC all'interno di una funzione di call-back non è supportata dal client IBM MQ .NET. Quindi, l'applicazione client non può creare o chiudere le sessioni all'interno del callback `MessageListener` in modalità di ricezione asincrona. Creare ed eliminare la sessione all'esterno del metodo `MessageListener`.

Concetti correlati

Sessioni transazionali

Le applicazioni XMS possono eseguire transazioni locali. Una *transazione locale* è una transazione che implica modifiche solo alle risorse del gestore code o del service integration bus a cui è connessa l'applicazione.

Riconoscimento messaggio

Ogni sessione non sottoposta a transazione dispone di una modalità di riconoscimento che determina il modo in cui vengono riconosciuti i messaggi ricevuti dall'applicazione. Sono disponibili tre modalità di riconoscimento e la scelta della modalità di riconoscimento influisce sulla progettazione dell'applicazione.

Consegna sincrona dei messaggi

I messaggi vengono consegnati in modo sincrono a un'applicazione se l'applicazione utilizza i metodi di ricezione degli oggetti `MessageConsumer`.

Modalità consegna messaggi

XMS supporta due modalità di consegna dei messaggi.

Consegna sincrona dei messaggi

I messaggi vengono consegnati in modo sincrono a un'applicazione se l'applicazione utilizza i metodi di ricezione degli oggetti `MessageConsumer`.

Utilizzando i metodi di ricezione, un'applicazione può attendere un periodo di tempo specificato per un messaggio oppure può attendere indefinitamente. In alternativa, se un'applicazione non desidera attendere un messaggio, può utilizzare il metodo `Ricevi senza attesa`.

Concetti correlati

Sessioni transazionali

Le applicazioni XMS possono eseguire transazioni locali. Una *transazione locale* è una transazione che implica modifiche solo alle risorse del gestore code o del service integration bus a cui è connessa l'applicazione.

Riconoscimento messaggio

Ogni sessione non sottoposta a transazione dispone di una modalità di riconoscimento che determina il modo in cui vengono riconosciuti i messaggi ricevuti dall'applicazione. Sono disponibili tre modalità di riconoscimento e la scelta della modalità di riconoscimento influisce sulla progettazione dell'applicazione.

Consegna di messaggi asincroni

XMS utilizza un thread per gestire tutte le consegne di messaggi asincroni per una sessione. Ciò significa che è possibile eseguire solo una funzione listener dei messaggi o un metodo `onMessage()` alla volta.

Modalità consegna messaggi

XMS supporta due modalità di consegna dei messaggi.

Modalità consegna messaggi

XMS supporta due modalità di consegna dei messaggi.

- I messaggi *persistenti* vengono consegnati una sola volta. Un server di messaggistica prende precauzioni speciali, come la registrazione dei messaggi, per garantire che i messaggi persistenti non vadano persi in transito, anche in caso di errore.
- I messaggi *Non persistenti* non vengono consegnati più di una volta. I messaggi non persistenti sono meno affidabili dei messaggi persistenti perché possono essere persi in transito in caso di errore.

La scelta della modalità di consegna è un compromesso tra affidabilità e prestazioni. I messaggi non persistenti vengono generalmente trasportati più rapidamente rispetto ai messaggi persistenti.

Concetti correlati

Sessioni transazionali

Le applicazioni XMS possono eseguire transazioni locali. Una *transazione locale* è una transazione che implica modifiche solo alle risorse del gestore code o del service integration bus a cui è connessa l'applicazione.

Riconoscimento messaggio

Ogni sessione non sottoposta a transazione dispone di una modalità di riconoscimento che determina il modo in cui vengono riconosciuti i messaggi ricevuti dall'applicazione. Sono disponibili tre modalità di riconoscimento e la scelta della modalità di riconoscimento influisce sulla progettazione dell'applicazione.

Consegna di messaggi asincroni

XMS utilizza un thread per gestire tutte le consegne di messaggi asincroni per una sessione. Ciò significa che è possibile eseguire solo una funzione listener dei messaggi o un metodo `onMessage()` alla volta.

Consegna sincrona dei messaggi

I messaggi vengono consegnati in modo sincrono a un'applicazione se l'applicazione utilizza i metodi di ricezione degli oggetti `MessageConsumer`.

Destinazioni

Un'applicazione di XMS utilizza un oggetto Destinazione per specificare la destinazione dei messaggi inviati e l'origine dei messaggi ricevuti.

Un'applicazione XMS può creare un oggetto Destinazione al runtime o ottenere una destinazione predefinita dal repository di oggetti gestiti.

Come con un `ConnectionFactory`, il modo più flessibile per un'applicazione XMS per specificare una destinazione è definirlo come un oggetto gestito. Utilizzando questo approccio, le applicazioni scritte in linguaggio C, C++ e .NET e Java, possono condividere le definizioni della destinazione. Le proprietà degli oggetti di destinazione gestiti possono essere modificate senza modificare alcun codice.

Per le applicazioni .NET, creare una destinazione utilizzando il metodo `CreateTopic` o `CreateQueue`. Questi due metodi sono disponibili in entrambi gli oggetti `ISession` e `XMSFactoryFactory` nell'API .NET. Per ulteriori informazioni, consultare [“Destinazioni in .NET”](#) a pagina 46 e [“IDestinazione”](#) a pagina 109.

Riferimenti correlati

IDestination (per l'interfaccia .NET)

Una destinazione è il punto in cui un'applicazione invia messaggi o è un'origine da cui un'applicazione riceve messaggi o entrambi.

Proprietà della destinazione

Una panoramica delle proprietà dell'oggetto Destinazione, con link a informazioni di riferimento più dettagliate.

Identificativi risorsa uniforme argomento

L'URI (uniform resource identifier) dell'argomento specifica il nome dell'argomento; può anche specificare una o più proprietà per esso.

L'URI per un argomento inizia con l'argomento di sequenza: //, seguito dal nome dell'argomento e (facoltativo) un elenco di coppie nome - valore che impostano le restanti proprietà dell'argomento. Il nome di un argomento non può essere vuoto.

Di seguito è riportato un esempio in un frammenti di codice .NET :

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

Per ulteriori informazioni sulle proprietà di un argomento, inclusi il nome e i valori validi che è possibile utilizzare in un URI, consultare ["Proprietà della destinazione"](#) a pagina 188.

Quando si specifica un URI di argomento da utilizzare in una sottoscrizione, è possibile utilizzare i caratteri jolly. La sintassi per questi caratteri jolly dipende dal tipo di connessione e dalla versione broker; sono disponibili le seguenti opzioni:

- Gestore code IBM WebSphere MQ 7.0 con formato carattere jolly a livello di carattere
- Gestore code IBM WebSphere MQ 7.0 con formato carattere jolly di livello argomento
- SIB (Service Integration Bus) WebSphere Application Server

Gestore code IBM WebSphere MQ 7.0 con formato carattere jolly a livello di carattere

Il gestore code IBM WebSphere MQ 7.0 con formato carattere jolly a livello di carattere utilizza i seguenti caratteri jolly:

- * per 0 o più caratteri
- ? per 1 carattere
- % per un carattere escape

[Tabella 1 a pagina 29](#) fornisce alcuni esempi su come utilizzare questo schema di caratteri jolly.

<i>Tabella 1. URI di esempio che utilizzano lo schema di caratteri jolly per il gestore code IBM WebSphere MQ 7.0</i>		
URI (Uniform Resource Identifier)	Corrisponde a	Esempi
"topic://Sport *Risultati"	Tutti gli argomenti che iniziano con "Sport" e terminano con "Risultati"	"topic://SportsResults"topic://Sport/Hockey/National/Div3/Results"
" topic://Sport?Risultati "	Tutti gli argomenti che iniziano con "Sport" seguiti da un unico carattere, seguito da "Risultati"	"topic:/SportsResults" e "topic://SportXResults"
"topic://Sport/ * ball*/Div? / Results*/??"	Argomenti	"topic://Sport/Football/Div1/Results/2002/Nov" e "topic://Sport/Netball/National/Div3/Results/02/Jan"

Gestore code IBM WebSphere MQ 7.0 con formato carattere jolly del livello di argomento

Il gestore code IBM WebSphere MQ 7.0 con il formato del carattere jolly del livello argomento utilizza i seguenti caratteri jolly:

- # per corrispondere a più livelli
- + per abbinare un singolo livello

Tabella 2 a pagina 30 fornisce alcuni esempi su come utilizzare questo schema di caratteri jolly.

<i>Tabella 2. URI di esempio che utilizzano uno schema di caratteri jolly a livello di argomento per il gestore code IBM WebSphere MQ 7.0</i>		
URI (Uniform Resource Identifier)	Corrisponde a	Esempi
"argomento://Sport/ + / Risultati"	Tutti gli argomenti con un unico nome di livello gerarchico tra Sport e Risultati	"topic://Sport/Football/Results" e "topic://Sport/Ju-Jitsu/Results"
"topic: //Sport/#/ Risultati"	Tutti gli argomenti che iniziano con "Sport/" e finiscono in "/Risultati"	"topic://Sport/Football/Results" e "topic://Sport/Hockey/National/Div3/Results"
" argomento: //Sport/ Calcio/#"	Tutti gli argomenti che iniziano con "Sport/Football/"	"topic://Sport/Football/Results" e "topic://Sport/Football/TeamNews/Signings/Managerial"

SIB (Service Integration Bus) WebSphere Application Server

Il SIB (service integration bus) WebSphere Application Server utilizza i seguenti caratteri jolly:

- * per far corrispondere qualsiasi carattere ad un livello nella gerarchia
- // per mettere in corrispondenza 0 o più livelli
- //. per far corrispondere 0 o più livelli (alla fine di un'espressione Argomento)

Tabella 3 a pagina 30 fornisce alcuni esempi su come utilizzare questo schema di caratteri jolly.

<i>Tabella 3. URI di esempio che utilizzano lo schema jolly per il SIB (Service Integration Bus) WebSphere Application Server</i>		
URI (Uniform Resource Identifier)	Corrisponde a	Esempi
"topic://Sport/ * ball/ Results"	Tutti gli argomenti con un unico nome di livello gerarchico che termina in "palla" tra Sport e Risultati	"topic://Sport/Football/Results" e "topic://Sport/Netball/Results"
"argomento://Sport// Risultati"	Tutti gli argomenti che iniziano con "Sport/" e finiscono in "/Risultati"	"topic://Sport/Football/Results" e "topic://Sport/Hockey/National/Div3/Results"
"topic://Sport/ Football//."	Tutti gli argomenti che iniziano con "Sport/Football/"	"topic://Sport/Football/Results" e "topic://Sport/Football/TeamNews/Signings/Managerial"
"topic://Sport/ * ball// Results//."	Argomenti	"topic://Sport/Football/Results" e "topic://Sport/Netball/National/Div3/Results/2002/November"

Concetti correlati

Identificativi risorsa uniforme coda

L'URI di una coda specifica il nome della coda; può anche specificare una o più proprietà della coda.

Destinazioni temporanee

Le applicazioni XMS possono creare e utilizzare destinazioni temporanee.

Caratteri jolly di destinazione

XMS fornisce supporto per i caratteri jolly di destinazione, garantendo che i caratteri jolly possano essere trasmessi al luogo in cui sono necessari per la corrispondenza. Esiste un diverso schema di caratteri jolly per ciascun tipo di server che XMS può utilizzare.

Riferimenti correlati

IDestination (per l'interfaccia .NET)

Una destinazione è il punto in cui un'applicazione invia messaggi o è un'origine da cui un'applicazione riceve messaggi o entrambi.

Proprietà della destinazione

Una panoramica delle proprietà dell'oggetto Destinazione, con link a informazioni di riferimento più dettagliate.

Identificativi risorsa uniforme coda

L'URI di una coda specifica il nome della coda; può anche specificare una o più proprietà della coda.

L'URI per una coda inizia con la coda di sequenza: //, seguito dal nome della coda; potrebbe includere anche un elenco di coppie nome - valore che impostano le rimanenti proprietà della coda.

Per le code IBM MQ (ma non per le code del provider di messaggistica predefinito WebSphere Application Server), il gestore code su cui si trova la coda può essere specificato prima della coda, con una / che separa il nome del gestore code dal nome della coda.

Se viene specificato un gestore code, questo deve essere quello a cui XMS è direttamente connesso per la connessione che utilizza questa coda oppure deve essere accessibile da questa coda. I gestori code remoti sono supportati solo per richiamare i messaggi dalle code e non per inserire i messaggi nelle code. Per i dettagli completi, fare riferimento alla documentazione del gestore code IBM MQ.

Se non viene specificato alcun gestore code, il separatore / extra è facoltativo e la sua presenza o assenza non fa alcuna differenza nella definizione della coda.

Le definizioni di coda riportate di seguito sono tutte equivalenti per una coda IBM MQ denominata QB su un gestore code denominato QM_A, a cui XMS è direttamente connesso:

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

Concetti correlati

Identificativi risorsa uniforme argomento

L'URI (uniform resource identifier) dell'argomento specifica il nome dell'argomento; può anche specificare una o più proprietà per esso.

Destinazioni temporanee

Le applicazioni XMS possono creare e utilizzare destinazioni temporanee.

Caratteri jolly di destinazione

XMS fornisce supporto per i caratteri jolly di destinazione, garantendo che i caratteri jolly possano essere trasmessi al luogo in cui sono necessari per la corrispondenza. Esiste un diverso schema di caratteri jolly per ciascun tipo di server che XMS può utilizzare.

Riferimenti correlati

IDestination (per l'interfaccia .NET)

Una destinazione è il punto in cui un'applicazione invia messaggi o è un'origine da cui un'applicazione riceve messaggi o entrambi.

Proprietà della destinazione

Una panoramica delle proprietà dell'oggetto Destinazione, con link a informazioni di riferimento più dettagliate.

Destinazioni temporanee

Le applicazioni XMS possono creare e utilizzare destinazioni temporanee.

Un'applicazione generalmente utilizza una destinazione temporanea per ricevere risposte ai messaggi di richiesta. Per specificare la destinazione in cui deve essere inviata una risposta a un messaggio di richiesta, un'applicazione richiama il metodo `Set JMSReplyTo` dell'oggetto `Message` che rappresenta il messaggio di richiesta. La destinazione specificata nella chiamata può essere una destinazione temporanea.

Anche se una sessione viene utilizzata per creare una destinazione temporanea, l'ambito di una destinazione temporanea è in realtà la connessione utilizzata per creare la sessione. Qualsiasi sessione della connessione può creare produttori di messaggi e consumer di messaggi per la destinazione temporanea. La destinazione temporanea rimane fino a quando non viene esplicitamente eliminata o fino al termine della connessione, a seconda di quale delle due si verifica per prima.

Quando un'applicazione crea una coda temporanea, viene creata una coda nel server di messaggistica a cui è connessa l'applicazione. Se l'applicazione è connessa ad un gestore code, viene creata una coda dinamica dalla coda modello il cui nome è specificato dalla proprietà `XMSC_WMQ_TEMPORARY_MODEL` e il prefisso utilizzato per formare il nome della coda dinamica è specificato dalla proprietà `XMSC_WMQ_TEMP_Q_PREFIX`. Se l'applicazione è connessa ad un SIB (Service Integration Bus), viene creata una coda temporanea nel bus e il prefisso utilizzato per formare il nome della coda temporanea viene specificato dalla proprietà `XMSC_WPM_TEMP_Q_PREFIX`.

Quando un'applicazione connessa a un service integration bus crea un argomento temporaneo, il prefisso utilizzato per formare il nome dell'argomento temporaneo viene specificato dalla proprietà `XMSC_WPM_TEMP_TOPIC_PREFIX`.

Concetti correlati

Identificativi risorsa uniforme argomento

L'URI (uniform resource identifier) dell'argomento specifica il nome dell'argomento; può anche specificare una o più proprietà per esso.

Identificativi risorsa uniforme coda

L'URI di una coda specifica il nome della coda; può anche specificare una o più proprietà della coda.

Caratteri jolly di destinazione

XMS fornisce supporto per i caratteri jolly di destinazione, garantendo che i caratteri jolly possano essere trasmessi al luogo in cui sono necessari per la corrispondenza. Esiste un diverso schema di caratteri jolly per ciascun tipo di server che XMS può utilizzare.

Riferimenti correlati

IDestination (per l'interfaccia .NET)

Una destinazione è il punto in cui un'applicazione invia messaggi o è un'origine da cui un'applicazione riceve messaggi o entrambi.

Proprietà della destinazione

Una panoramica delle proprietà dell'oggetto Destinazione, con link a informazioni di riferimento più dettagliate.

Caratteri jolly di destinazione

XMS fornisce supporto per i caratteri jolly di destinazione, garantendo che i caratteri jolly possano essere trasmessi al luogo in cui sono necessari per la corrispondenza. Esiste un diverso schema di caratteri jolly per ciascun tipo di server che XMS può utilizzare.

I programmi sono:

Tipo di connessione	Schema di caratteri jolly	Descrizione
WebSphere MQ Gestore code	* ? %	0 o più caratteri 1 carattere Carattere Escape
Connessione in tempo reale a un broker	# +	Metti in corrispondenza più livelli Corrispondenza con un singolo livello
WebSphere Integration Bus	* // //.	Corrisponde a qualsiasi carattere ad un livello nella gerarchia Corrisponde a 0 o più livelli Corrispondenza di 0 o più livelli (alla fine di un'espressione Argomento)

Concetti correlati

Identificativi risorsa uniforme argomento

L'URI (uniform resource identifier) dell'argomento specifica il nome dell'argomento; può anche specificare una o più proprietà per esso.

Identificativi risorsa uniforme coda

L'URI di una coda specifica il nome della coda; può anche specificare una o più proprietà della coda.

Destinazioni temporanee

Le applicazioni XMS possono creare e utilizzare destinazioni temporanee.

Riferimenti correlati

IDestination (per l'interfaccia .NET)

Una destinazione è il punto in cui un'applicazione invia messaggi o è un'origine da cui un'applicazione riceve messaggi o entrambi.

Proprietà della destinazione

Una panoramica delle proprietà dell'oggetto Destinazione, con link a informazioni di riferimento più dettagliate.

Produttori di messaggi

In XMS, è possibile creare un produttore di messaggi con una destinazione valida o senza alcuna destinazione associata. Quando si crea un produttore di messaggi con una destinazione null, è necessario specificare una destinazione valida quando si invia un messaggio.

Produttori di messaggi senza destinazione associata

In XMS .NET, è possibile creare un produttore di messaggi con destinazione null.

Per creare un produttore di messaggi senza destinazione associata quando si utilizza l'API .NET, è necessario passare NULL come parametro nel metodo `CreateProducer()` dell'oggetto `ISession` (ad esempio, `session.CreateProducer(null)`). Tuttavia, è necessario specificare una destinazione valida quando viene inviato il messaggio.

Produttori di messaggi con destinazione associata

In questo scenario, il produttore del messaggio viene creato utilizzando una destinazione valida. Durante l'operazione di invio, non è necessario specificare la destinazione.

Utenti del messaggio

I consumer di messaggi possono essere classificati come sottoscrittori durevoli e non durevoli e consumer di messaggi sincroni e asincroni.

Sottoscrittori permanenti

Un sottoscrittore durevole è un utente di messaggi che riceve tutti i messaggi pubblicati su un argomento, inclusi i messaggi pubblicati mentre il sottoscrittore non è attivo.

Le informazioni contenute in questo argomento sono rilevanti solo se un'applicazione si connette a un gestore code IBM MQ o a un SIB (Service Integration Bus) WebSphere Application Server . Le informazioni non sono rilevanti per una connessione in tempo reale a un broker.

Per creare un sottoscrittore durevole per un argomento, un'applicazione richiama il metodo `CreaSottoscrittoreDurevole` di un oggetto `Session`, specificando come parametri un nome che identifica la sottoscrizione durevole e un oggetto `Destinazione` che rappresenta l'argomento. L'applicazione può creare un sottoscrittore durevole con o senza un selettore di messaggi e può specificare se il sottoscrittore durevole deve ricevere i messaggi pubblicati dalla propria connessione.

La sessione utilizzata per creare un sottoscrittore durevole deve avere un identificativo client associato. L'identificativo client è lo stesso associato alla connessione utilizzata per creare la sessione; è specificato come descritto in [“ConnectionFactories e oggetti Connection”](#) a pagina 21.

Il nome che identifica la sottoscrizione durevole deve essere univoco all'interno dell'identificativo client e quindi l'identificativo client fa parte dell'identificativo univoco completo della sottoscrizione durevole. Il server di messaggistica conserva un record della sottoscrizione durevole e garantisce che tutti i messaggi pubblicati sull'argomento vengano conservati fino a quando non vengono riconosciuti dal sottoscrittore durevole o fino a quando non scadono.

Il server di messaggistica continua a conservare il record della sottoscrizione durevole anche dopo la chiusura del sottoscrittore durevole. Per riutilizzare una sottoscrizione durevole creata in precedenza, un'applicazione deve creare un sottoscrittore durevole specificando lo stesso nome di sottoscrizione e utilizzando una sessione con lo stesso identificativo client, come quelli associati alla sottoscrizione durevole. Solo una sessione alla volta può avere un sottoscrittore durevole per una particolare sottoscrizione durevole.

L'ambito di una sottoscrizione durevole è il server di messaggistica che conserva un record della sottoscrizione. Se due applicazioni connesse a server di messaggistica differenti creano un sottoscrittore durevole utilizzando lo stesso nome sottoscrizione e lo stesso identificativo client, vengono create due sottoscrizioni durevoli completamente indipendenti.

Per eliminare una sottoscrizione durevole, un'applicazione richiama il metodo `Unsubscribe` di un oggetto `Session`, specificando come parametro il nome che identifica la sottoscrizione durevole. L'identificativo client associato alla sessione deve essere uguale a quello associato alla sottoscrizione durevole. Il server di messaggistica elimina il record della sottoscrizione durevole che sta gestendo e non invia ulteriori messaggi al sottoscrittore durevole.

Per modificare una sottoscrizione esistente, un'applicazione può creare un sottoscrittore durevole utilizzando lo stesso nome sottoscrizione e lo stesso identificativo client, ma specificando un argomento o un selettore di messaggi differenti (o entrambi). La modifica di una sottoscrizione durevole equivale ad eliminare la sottoscrizione e a crearne una nuova.

Per un'applicazione che si connette a un gestore code IBM WebSphere MQ 7.0 , XMS gestisce le code del sottoscrittore. Pertanto, l'applicazione non è richiesta per specificare una coda del sottoscrittore. XMS ignorerà la coda del sottoscrittore, se specificata.

Tuttavia, per un'applicazione che si connette al gestore code IBM WebSphere MQ 6.0 , ogni sottoscrittore durevole deve avere una coda sottoscrittore designata. Per specificare il nome della coda del sottoscrittore per un argomento, impostare la proprietà `XMSC_WMQ_DUR_SUBQ` dell'oggetto `Destinazione` che rappresenta l'argomento. La coda del sottoscrittore predefinita è `SYSTEM.JMS.D.SUBSCRIBER.QUEUE`.

I sottoscrittori durevoli che si collegano a gestori code IBM WebSphere MQ 6.0 possono condividere una singola coda di sottoscrittori oppure ciascun sottoscrittore durevole può recuperare i propri messaggi dalla propria coda di sottoscrittori esclusiva. Per una discussione su quale approccio adottare per la tua applicazione, vedi *XMSIBM WebSphere MQ Utilizzo di Java*.

Non è possibile modificare la coda del sottoscrittore per una sottoscrizione durevole. L'unico modo per modificare la coda del sottoscrittore è eliminare la sottoscrizione e crearne una nuova.

Per un'applicazione che si connette a un SIB (Service Integration Bus), ogni sottoscrittore durevole deve avere una home di sottoscrizione durevole designata. Per specificare la home della sottoscrizione durevole per tutti i sottoscrittori durevoli che utilizzano la stessa connessione, impostare la proprietà `XMSC_WPM_DUR_SUB_HOME` dell'oggetto `ConnectionFactory` utilizzato per creare la connessione. Per specificare la home di sottoscrizione durevole per un singolo argomento, impostare la proprietà `XMSC_WPM_DUR_SUB_HOME` dell'oggetto `Destinazione` che rappresenta l'argomento. È necessario specificare una home di sottoscrizione durevole per una connessione prima che un'applicazione possa creare un sottoscrittore durevole che utilizza la connessione. Qualsiasi valore specificato per una destinazione sovrascrive il valore specificato per la connessione.

Sottoscrittori non durevoli

Un sottoscrittore non durevole è un utente di messaggi che riceve solo i messaggi pubblicati mentre il sottoscrittore è attivo. I messaggi consegnati mentre il sottoscrittore è inattivo vengono persi.

Le informazioni in questo argomento sono rilevanti solo quando si utilizza la messaggistica di pubblicazione / sottoscrizione su un gestore code IBM WebSphere MQ 6.0 .

Se gli oggetti consumer non vengono eliminati prima o durante la chiusura della connessione, i messaggi possono rimanere nelle code del broker per i sottoscrittori non più attivi.

In questa situazione, le code possono essere eliminate da questi messaggi utilizzando il programma di utilità `Cleanup` fornito con `IBM WebSphere MQ classes for JMS`. I dettagli su come utilizzare questo programma di utilità sono forniti in *IBM WebSphere MQ Utilizzo di Java* . Potrebbe anche essere necessario aumentare la profondità della coda della coda del sottoscrittore se su questa coda rimane un numero elevato di messaggi.

Utenti di messaggi sincroni

Il destinatario del messaggio sincrono riceve i messaggi da una coda in modo sincrono.

Un utente di messaggi sincrono riceve un messaggio alla volta. Quando viene utilizzato il metodo `Receive(wait interval)` ; la chiamata attende solo un periodo di tempo specificato, in millisecondi, per un messaggio o fino a quando il destinatario del messaggio non viene chiuso.

Se viene utilizzato il metodo `ReceiveNoWait()` , il destinatario del messaggio sincrono riceve i messaggi senza alcun ritardo; se il messaggio successivo è disponibile, viene ricevuto immediatamente, altrimenti viene restituito un puntatore a un oggetto `Message` null.

Consumer messaggi asincroni

Il destinatario del messaggio asincrono riceve il messaggio da una coda in modo asincrono. Il listener dei messaggi registrato dall'applicazione viene richiamato ogni volta che un nuovo messaggio è disponibile sulla coda.

Messaggi non elaborabili in XMS

Un messaggio non elaborabile è quello che non può essere elaborato da un'applicazione MDB di ricezione. Se viene rilevato un messaggio non elaborabile, l'oggetto `MessageConsumer XMS` può riaccodare il messaggio in base a due proprietà della coda, `BOQUEUE` e `BOTHRESH`.

In alcune circostanze, un messaggio consegnato a un MDB potrebbe essere sottoposto a rollback su una coda IBM MQ . Ciò può verificarsi, ad esempio, se un messaggio viene consegnato all'interno di un'unità di lavoro di cui viene successivamente eseguito il rollback. Un messaggio di cui viene eseguito il rollback viene generalmente consegnato di nuovo, ma un messaggio formattato in modo errato potrebbe causare ripetutamente un errore MDB e quindi non può essere consegnato. Tale messaggio è chiamato un messaggio di veleno. È possibile configurare IBM MQ in modo che il messaggio non elaborabile venga

automaticamente trasferito a un'altra coda per ulteriori indagini o venga eliminato. Per informazioni su come configurare IBM MQ in questo modo, consultare [Gestione dei messaggi non elaborabili in ASF](#).

A volte, un messaggio formattato in modo errato arriva su una coda. In questo contesto, un formato non corretto indica che l'applicazione ricevente non può elaborare correttamente il messaggio. Un messaggio di questo tipo può causare il malfunzionamento dell'applicazione ricevente e il backout di questo messaggio formattato in modo errato. Il messaggio può essere inviato ripetutamente alla coda di input e ripetutamente ripristinato dall'applicazione. Questi messaggi sono noti come messaggi dannosi. L'oggetto XMS MessageConsumer rileva i messaggi non elaborabili e li reinvia a una destinazione alternativa.

Il gestore code IBM MQ conserva un record del numero di volte in cui è stato eseguito il backout di ciascun messaggio. Quando questo valore raggiunge un valore di soglia configurabile, l'utente del messaggio riaccoda il messaggio ad una coda di backout denominata. Se questa riaccodamento non riesce per qualsiasi motivo, il messaggio viene rimosso dalla coda di input e riaccodato alla coda di messaggi non recapitabili o eliminato.

Gli oggetti XMS ConnectionConsumer gestiscono i messaggi non elaborabili nello stesso modo e utilizzando le stesse proprietà della coda. Se più utenti della connessione stanno monitorando la stessa coda, è possibile che il messaggio non elaborabile possa essere consegnato a un'applicazione più volte del valore di soglia prima che si verifichi la riaccodamento. Questo comportamento è dovuto al modo in cui i singoli utenti della connessione monitorano le code e riaccodano i messaggi dannosi.

Il valore di soglia ed il nome della coda di backout sono attributi di una coda IBM MQ. I nomi degli attributi sono BackoutThreshold e BackoutRequeueQName. La coda a cui si applicano è la seguente:

- Per la messaggistica point - to - point, questa è la coda locale sottostante. Ciò è importante quando gli utenti dei messaggi e dei collegamenti utilizzano gli alias della coda.
- Per la messaggistica di pubblicazione / sottoscrizione in modalità normale del fornitore di messaggistica di IBM MQ, è la coda modello da cui viene creata la coda gestita dell'argomento.
- Per la messaggistica di pubblicazione / sottoscrizione in modalità di migrazione del provider di messaggistica IBM MQ, è la coda CCSUB definita nell'oggetto factory TopicConnection o la coda CCDSUB definita nell'oggetto argomento.

Per impostare gli attributi QName BackoutThreshold e BackoutRequeue, immettere il seguente comando MQSC:

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

Per la messaggistica di pubblicazione / sottoscrizione, se il sistema crea una coda dinamica per ciascuna sottoscrizione, questi valori di attributo vengono ottenuti dalla coda di modello WebSphere MQ per la coda di modello JMS, SYSTEM.JMS.MODEL.QUEUE. Per modificare queste impostazioni, utilizzare:

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

Se il valore della soglia di backout è zero, la gestione dei messaggi non elaborabili è disabilitata e i messaggi non elaborabili rimangono sulla coda di input. Altrimenti, quando il conteggio di backout raggiunge il valore di soglia, il messaggio viene inviato alla coda di backout indicata.

Se il conteggio di backout raggiunge il valore di soglia, ma il messaggio non può andare alla coda di backout, il messaggio viene inviato alla coda di messaggi non recapitabili o, se il messaggio è non persistente, viene eliminato.

Questa situazione si verifica se la coda di backout non è definita o se l'oggetto MessageConsumer non può inviare il messaggio alla coda di backout.

Configurazione del sistema per eseguire la gestione dei messaggi non elaborabili

La coda utilizzata da XMS .NET durante l'interrogazione degli attributi **BOTHRESH** e **BOQNAME** dipende dallo stile di messaggistica eseguito:

- Per la messaggistica point - to - point, questa è la coda locale sottostante. Ciò è importante quando un'applicazione XMS .NET utilizza i messaggi dalle code alias o dalle code cluster.
- Per la messaggistica di pubblicazione / sottoscrizione, viene creata una coda gestita per contenere i messaggi per una applicazione. XMS .NET interroga la coda gestita per determinare i valori per gli attributi **BOTHRESH** e **BOQNAME** .

La coda gestita viene creata da una coda modello associata all'oggetto Argomento a cui l'applicazione ha eseguito la sottoscrizione ed eredita i valori degli attributi **BOTHRESH** e **BOQNAME** dalla coda modello. La coda modello utilizzata dipende dal fatto che l'applicazione ricevente abbia eliminato una sottoscrizione durevole o non durevole:

- La coda modello utilizzata per le sottoscrizioni durevoli è specificata dall'attributo **MDURMDL** dell'argomento. Il valore predefinito di questo attributo è SYSTEM . DURABLE . MODEL . QUEUE.
- Per sottoscrizioni non durevoli, la coda modello utilizzata è specificata dall'attributo **MNDURMDL** . Il valore predefinito dell'attributo **MNDURMDL** è SYSTEM . NDURABLE . MODEL . QUEUE.

Quando si interrogano gli attributi **BOTHRESH** e **BOQNAME** , XMS .NET:

- Apre la coda locale o la coda di destinazione per una coda alias.
- Interroga gli attributi **BOTHRESH** e **BOQNAME** .
- Chiude la coda locale o la coda di destinazione per una coda alias.

Le opzioni di apertura utilizzate quando si apre una coda locale o la coda di destinazione per una coda alias, dipendono dalla versione di IBM MQ utilizzata:

- Per IBM MQ 9.0.0 Fix Pack 8e versioni precedenti, se la coda locale o la coda di destinazione per una coda alias è una coda cluster, XMS .NET apre la coda con le opzioni MQOO_INPUT_AS_Q_DEF, MQOO_INQUIRE e MQOO_FAIL_IF QUIESCING . Ciò significa che l'utente che esegue l'applicazione ricevente deve disporre dell'interrogazione e dell'accesso all'istanza locale della coda cluster.

XMS .NET apre tutti gli altri tipi di coda locale con le opzioni di apertura MQOO_INQUIRE e MQOO_FAIL_IF QUIESCING. Affinché XMS .NET esegua una query dei valori degli attributi, l'utente che esegue l'applicazione ricevente deve avere accesso alla coda locale.

- **V 9.0.0.9** Quando si utilizza XMS .NET da IBM MQ 9.0.0 Fix Pack 9, l'utente che esegue l'applicazione ricevente deve avere accesso alla coda locale, indipendentemente dal tipo di coda.

Per spostare i messaggi non elaborabili in una coda di riaccodamento di backout o in una coda di messaggi non recapitabili del gestore code, è necessario concedere all'utente che esegue l'applicazione le autorizzazioni put e passall .

Gestione dei messaggi non elaborabili in ASF

Quando si utilizzano le ASF (Application Server Facilities), ConnectionConsumer, invece di MessageConsumer, elabora messaggi non elaborabili. Il ConnectionConsumer riaccoda i messaggi in base alle proprietà QName BackoutThreshold e BackoutQueuedella coda.

Quando un'applicazione utilizza ConnectionConsumers, le circostanze in cui viene eseguito il backout di un messaggio dipendono dalla sessione fornita dal server delle applicazioni:

- Quando la sessione non è sottoposta a transazione, con AUTO_RICONOSCIUTO_AUTOMATICO o DUPS_OK_RICONOSCIMENTO, viene eseguito il backout di un messaggio solo dopo un errore di sistema o se l'applicazione termina in modo imprevisto.
- Quando la sessione non è sottoposta a transazioni con CLIENT_ACKNOWLEDGED, i messaggi non riconosciuti possono essere ripristinati dal server delle applicazioni che richiama Session.recover().

Di solito, l'implementazione client di MessageListener o del server delle applicazioni richiama Message.acknowledge(). Message.acknowledge() riconosce tutti i messaggi consegnati nella sessione fino ad ora.

- Quando la sessione viene eseguita, i messaggi non riconosciuti possono essere ripristinati dal server delle applicazioni che richiama Session.rollback().

Browser coda

Un'applicazione utilizza un browser della coda per esaminare i messaggi su una coda senza rimuoverli.

Per creare un browser delle code, un'applicazione richiama il metodo `Crea browser delle code` di un oggetto `ISession`, specificando come parametro un oggetto `Destinazione` che identifica la coda da esaminare. L'applicazione può creare un browser della coda con o senza un selettore di messaggi.

Dopo aver creato un browser di code, l'applicazione può chiamare il metodo `GetEnumerator` dell'oggetto `IQueueBrowser` per ottenere un elenco dei messaggi sulla coda. Il metodo restituisce un enumeratore che incapsula un elenco di oggetti `Messaggio`. L'ordine degli oggetti `Messaggio` nell'elenco è lo stesso dell'ordine in cui i messaggi vengono richiamati dalla coda. L'applicazione può quindi utilizzare l'enumeratore per esaminare ogni messaggio a turno.

L'enumeratore viene aggiornato dinamicamente quando i messaggi vengono inseriti nella coda e rimossi dalla coda. Ogni volta che l'applicazione richiama `IEnumerator.MoveNext()` per sfogliare il messaggio successivo nella coda, il messaggio riflette il contenuto corrente della coda.

Un'applicazione può richiamare il metodo `GetEnumerator` più di una volta per un determinato browser della coda. Ciascuna chiamata restituisce un nuovo enumeratore. L'applicazione può quindi utilizzare più di un enumeratore per esaminare i messaggi su una coda e mantenere più posizioni all'interno della coda.

Un'applicazione può utilizzare un browser di code per cercare un messaggio adatto da eliminare da una coda e quindi utilizzare un consumatore di messaggi con un selettore di messaggi per rimuovere il messaggio. Il selettore del messaggio può selezionare il messaggio in base al valore del campo di intestazione `JMSMessageID`. Per informazioni su questo e altri campi di intestazione del messaggio JMS, consultare [“Campi di intestazione in un messaggio XMS” a pagina 70](#).

Richiedenti

Un'applicazione utilizza un richiedente per inviare un messaggio di richiesta e quindi attendere e ricevere la risposta.

Molte applicazioni di messaggistica si basano su algoritmi che inviano un messaggio di richiesta e quindi attendono una risposta. XMS fornisce una classe denominata `Requestor` per facilitare lo sviluppo di questo stile di applicazione.

Per creare un richiedente, un'applicazione richiama il costruttore `Create Requestor` della classe `Requestor`, specificando come parametri un oggetto `Session` e un oggetto `Destination` che identifica dove devono essere inviati i messaggi di richiesta. La sessione non deve essere sottoposta a transazione né deve avere una modalità di riconoscimento `XMSC_CLIENT_ACKNOWLEDGEMENT`. Il costruttore crea automaticamente una coda temporanea o un argomento in cui devono essere inviati i messaggi di risposta.

Una volta creato un richiedente, l'applicazione può richiamare il metodo di richiesta dell'oggetto `Requestor` per inviare un messaggio di richiesta e quindi attendere e ricevere una risposta dall'applicazione che riceve il messaggio di richiesta. La chiamata attende la ricezione della risposta o la fine della sessione, a seconda di quale si verifica per prima. Solo una risposta è richiesta dal richiedente per ogni messaggio di richiesta.

Quando l'applicazione chiude il richiedente, la coda temporanea o l'argomento vengono eliminati. La sessione associata, tuttavia, non viene chiusa.

Eliminazione oggetto

Quando un'applicazione elimina un oggetto XMS creato, XMS rilascia le risorse interne assegnate all'oggetto.

Quando un'applicazione crea un oggetto XMS, XMS assegna memoria e altre risorse interne all'oggetto. XMS conserva queste risorse interne finché l'applicazione non elimina esplicitamente l'oggetto richiamando il metodo `close` o `delete` dell'oggetto, a quel punto XMS rilascia le risorse interne. Se un'applicazione tenta di eliminare un oggetto già eliminato, la chiamata viene ignorata.

Quando un'applicazione elimina un oggetto `Connection` o `Session`, XMS elimina automaticamente alcuni oggetti associati e rilascia le relative risorse interne. Si tratta di oggetti creati dall'oggetto `Connection` o `Session` e che non hanno alcuna funzione indipendente dall'oggetto. Questi oggetti sono mostrati in [Tabella 4 a pagina 39](#).

Nota: se un'applicazione chiude una connessione con sessioni dipendenti, vengono eliminati anche tutti gli oggetti dipendenti da tali sessioni. Solo un oggetto `Connection` o `Session` può avere oggetti dipendenti.

<i>Tabella 4. Oggetti eliminati automaticamente</i>		
Oggetto eliminato	Metodo	Oggetti dipendenti che vengono eliminati automaticamente
Connessione	Chiudi connessione	Oggetti <code>ConnectionMetaData</code> e <code>Session</code>
Sessione	Chiudi sessione	<code>MessageConsumer</code> , <code>MessageProducer</code> , <code>QueueBrowse</code> e oggetti <code>Requestor</code>

Transazioni IBM MQ XA gestite tramite XMS

Le transazioni IBM MQ XA gestite possono essere utilizzate tramite XMS.

Per utilizzare le transazioni XA tramite XMS, è necessario creare una sessione transazionale. Quando la transazione XA è in uso, il controllo transazioni avviene tramite le transazioni globali DTC (Distributed Transaction Coordinator) e non avviene tramite le sessioni XMS. Quando si utilizzano transazioni XA, `Session.commit` o `Session.rollback` non possono essere emessi nella sessione XMS. Utilizzare invece i metodi `Transscope.Commit` o `Transscope.Rollback` DTC per eseguire il commit o il rollback delle transazioni. Se una sessione viene utilizzata per la transazione XA, il producer o il consumer creato utilizzando la sessione deve essere una parte della transazione XA. Non possono essere utilizzati per operazioni al di fuori dell'ambito della transazione XA. Non possono essere utilizzati per operazioni come `Producer.send` o `Consumer.receive` all'esterno della transazione XA.

Viene generato un oggetto eccezione `IllegalStateException` se

- La sessione di transazione XA viene utilizzata per `Session.commit` o `Session.rollback`.
- Gli oggetti producer o consumer una volta utilizzati nella sessione di transazione XA vengono utilizzati al di fuori dell'ambito della transazione XA.

Le transazioni XA non sono supportate nei consumer asincroni.

Nota:

1. Una chiusura potrebbe essere emessa sull'oggetto `Producer`, `Consumer`, `Session` o `Connection` prima del commit della transazione XA. In quali casi viene eseguito il rollback dei messaggi nella transazione. Allo stesso modo, se la connessione viene interrotta prima del commit della transazione XA, viene eseguito il rollback di tutti i messaggi nella transazione. Per un oggetto `Producer`, un rollback indica che i messaggi non vengono inseriti sulla coda. Per un oggetto `Consumer`, un rollback indica che i messaggi rimangono nella coda.
2. Se un oggetto `Producer` inserisce un messaggio con `TimeToLive` in `TransactionScope` e un commit viene emesso una volta trascorso il tempo, il messaggio può scadere prima dell'emissione di commit. In questo caso, il messaggio non viene reso disponibile agli oggetti `Consumer`.
3. Gli oggetti `Session` non sono supportati tra i thread. L'utilizzo di transazioni con oggetti `Session` condivisi tra thread non è supportato.

XMS Tipi primitivi

XMS fornisce equivalenti degli otto tipi primitivi Java (`byte`, `short`, `int`, `long`, `float`, `double`, `char` e `boolean`). Ciò consente lo scambio di messaggi tra XMS e JMS senza che i dati vadano persi o danneggiati.

[Tabella 5 a pagina 40](#) elenca il tipo di dati Java equivalente, la dimensione e il valore minimo e massimo di ciascun tipo primitivo XMS.

Tabella 5. Tipi di dati XMS e relativi Java equivalenti

Tipo di dati XMS	Tipo di datiJava compatibile	Dimensione	Valore minimo	Valore massimo
System.Boolean	booleano	32 bit	falso	vero
System.SBYTE	byte	8 bit	-2^7 (-128)	2^7-1 (127)
System.BYTE	byte	8 bit	-2^7 (-128)	2^7-1 (127)
System.CHAR	byte	8 bit	-2^7 (-128)	2^7-1 (127)
System.Int16	breve	16 bit	-2^{15} (-32768)	$2^{15}-1$ (32767)
System.Int32	int	32 bit	-2^{31} (-2147483648)	$2^{31}-1$ (2147483647)
System.Int64	completo	64 bit	-2^{63} (-9223372036854775808)	$2^{63}-1$ (9223372036854775807)
System.Single	mobile	32 bit	-3.402823E-38 (precisione a 7 cifre)	3.402823E+38 (con precisione a 7 cifre)
System.Double	doppio	64 bit	-1.79769313486231E-308 (precisione a 15 cifre)	1.79769313486231E+308 (precisione a 15 cifre)

Concetti correlati

Attributi e proprietà degli oggetti

Un oggetto XMS può avere attributi e proprietà, che sono caratteristiche dell'oggetto, che sono implementati in modi diversi.

Conversione implicita di un valore di proprietà da un tipo di dati a un altro

Quando un'applicazione acquisisce il valore di una proprietà, il valore può essere convertito da XMS in un altro tipo di dati. Molte regole regolano quali conversioni sono supportate e come XMS esegue le conversioni.

Riferimenti correlati

Tipi di dati per elementi di dati dell'applicazione

Per essere certi che un'applicazione XMS possa scambiare messaggi con un'applicazione IBM MQ classes for JMS, entrambe le applicazioni devono essere in grado di interpretare i dati dell'applicazione nel corpo di un messaggio nello stesso modo.

Conversione implicita di un valore di proprietà da un tipo di dati a un altro

Quando un'applicazione acquisisce il valore di una proprietà, il valore può essere convertito da XMS in un altro tipo di dati. Molte regole regolano quali conversioni sono supportate e come XMS esegue le conversioni.

Una proprietà di un oggetto ha un nome e un valore; il valore ha un tipo di dati associato, dove il valore di una proprietà è indicato anche come *tipo di proprietà*.

Un'applicazione utilizza i metodi della classe PropertyContext per richiamare e impostare le proprietà degli oggetti. Per ottenere il valore di una proprietà, un'applicazione richiama il metodo appropriato per il tipo di proprietà. Ad esempio, per ottenere il valore di una proprietà integer, un'applicazione in genere richiama il metodo di proprietà GetInt.

Tuttavia, quando un'applicazione ottiene il valore di una proprietà, il valore può essere convertito da XMS in un altro tipo di dati. Ad esempio, per ottenere il valore di una proprietà integer, un'applicazione può chiamare il metodo di proprietà GetString, che restituisce il valore della proprietà come stringa. Le conversioni supportate da XMS sono mostrate in [Tabella 6 a pagina 41](#).

<i>Tabella 6. Conversioni supportate da un tipo di proprietà ad altri tipi di dati</i>	
Tipo di proprietà	Tipi di dati di destinazione supportati
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
array System.SByte	System.String
System.Int16	System.String, System.Int32, System.Int64

Le seguenti regole generali regolano le conversioni supportate:

- I valori delle proprietà numeriche possono essere convertiti da un tipo di dati a un altro, purché non si perda alcun dato durante la conversione. Ad esempio, il valore di una proprietà con tipo dati System.Int32 può essere convertito in un valore con tipo dati System.Int64, ma non può essere convertito in un valore con tipo dati System.Int16.
- Un valore di proprietà di qualsiasi tipo di dati può essere convertito in una stringa.
- Un valore della proprietà stringa può essere convertito in qualsiasi altro tipo di dati, a condizione che la stringa sia formattata correttamente per la conversione. Se un'applicazione tenta di convertire un valore della proprietà stringa che non è formattato correttamente, XMS potrebbe restituire errori.
- Se un'applicazione tenta una conversione non supportata, XMS potrebbe restituire un errore.

Le seguenti regole si applicano quando un valore di proprietà viene convertito da un tipo di dati a un altro:

- Quando si converte un valore della proprietà booleana in una stringa, il valore true viene convertito nella stringa "true" e il valore false viene convertito nella stringa "false".
- Quando si converte un valore della proprietà booleana in un tipo di dati numerico, incluso System.SByte, il valore true viene convertito in 1 e il valore false viene convertito in 0.
- Quando si converte un valore di proprietà stringa in un valore booleano, la stringa "true" (non sensibile al maiuscolo / minuscolo) o "1" viene convertita in true e la stringa "false" (non sensibile al maiuscolo / minuscolo) o "0" viene convertita in false. Tutte le altre stringhe non possono essere convertite.
- Quando si converte un valore di proprietà stringa in un valore con tipo di dati System.Int32, System.Int64, System.SByte o System.Int16, la stringa deve avere il seguente formato:

[spazi] [segno] cifre

I componenti stringa sono definiti come segue:

spazi vuoti

Caratteri vuoti iniziali facoltativi.

segno

Un segno più (+) o segno meno (-) facoltativo.

cifre

Una sequenza contigua di caratteri cifra (0-9). Deve essere presente almeno un carattere cifra.

Dopo la sequenza di caratteri cifra, la stringa può contenere altri caratteri che non sono caratteri cifra, ma la conversione si interrompe non appena viene raggiunto il primo di questi caratteri. Si presuppone che la stringa rappresenti un numero intero decimale.

XMS potrebbe restituire un errore se la stringa non è formattata correttamente.

- Quando si converte un valore di proprietà della stringa in un valore con tipo di dati System.Double o System.Float, la stringa deve avere il seguente formato:

[*blank*] [*segno*] [*cifre*] [*punto*[*d_cifre*]] [*e_char*[*e_segno*]*e_cifre*]

I componenti stringa sono definiti come segue:

spazi vuoti

(Facoltativo) Caratteri vuoti iniziali.

segno

(Facoltativo) Il segno più (+) o il segno meno (-).

cifre

Una sequenza contigua di caratteri cifra (0-9). Almeno un carattere cifra deve essere presente in *cifre* o *d_digits*.

punto

(Facoltativo) Punto decimale (.).

d_cifre

Una sequenza contigua di caratteri cifra (0-9). Almeno un carattere cifra deve essere presente in *cifre* o *d_digits*.

e_car

Un carattere esponente, che è *E* o *e*.

e_firma

(Facoltativo) Segno più (+) o segno meno (-) per l'esponente.

e_cifre

Una sequenza contigua di caratteri cifra (0-9) per l'esponente. Deve essere presente almeno un carattere cifra se la stringa contiene un carattere esponente.

Dopo la sequenza di caratteri cifra, o i caratteri facoltativi che rappresentano un esponente, la stringa può contenere altri caratteri che non sono caratteri cifra, ma la conversione si interrompe non appena viene raggiunto il primo di questi caratteri. Si presuppone che la stringa rappresenti un numero a virgola mobile decimale con un esponente che sia una potenza di 10.

XMS potrebbe restituire un errore se la stringa non è formattata correttamente.

- Quando si converte un valore di proprietà numerico in una stringa, incluso un valore di proprietà con tipo di dati System.SByte, il valore viene convertito nella rappresentazione di stringa del valore come un numero decimale, non la stringa contenente il carattere ASCII per tale valore. Ad esempio, il numero intero 65 viene convertito nella stringa "65", non nella stringa "A".
- Quando si converte un valore della proprietà della schiera di byte in una stringa, ogni byte viene convertito in 2 caratteri esadecimali che rappresentano il byte. Ad esempio, l'array di byte {0xF1, 0x12, 0x00, 0xFF} viene convertito nella stringa "F11200FF".

Le conversioni da un tipo di proprietà ad altri tipi di dati sono supportate dai metodi delle classi Proprietà e PropertyContext .

Concetti correlati

Attributi e proprietà degli oggetti

Un oggetto XMS può avere attributi e proprietà, che sono caratteristiche dell'oggetto, che sono implementati in modi diversi.

XMS Tipi primitivi

XMS fornisce equivalenti degli otto tipi primitivi Java (byte, short, int, long, float, double, char e boolean). Ciò consente lo scambio di messaggi tra XMS e JMS senza che i dati vadano persi o danneggiati.

Riferimenti correlati

Associa messaggi

Il corpo di un messaggio di associazione contiene una serie di coppie nome - valore, in cui ogni valore ha un tipo di dati associato.

Il processo di flusso dei messaggi

Il corpo di un messaggio di flusso contiene un flusso di valori, in cui ogni valore ha un tipo di dati associato.

Tipi di dati per elementi di dati dell'applicazione

Per essere certi che un'applicazione XMS possa scambiare messaggi con un'applicazione IBM MQ classes for JMS , entrambe le applicazioni devono essere in grado di interpretare i dati dell'applicazione nel corpo di un messaggio nello stesso modo.

Iteratori

Un iteratore incapsula un elenco di oggetti e un cursore che mantiene la posizione corrente nell'elenco. Il concetto di un iteratore, come disponibile in IBM Message Service Client for C/C++ , viene implementato utilizzando l'interfaccia IEnumerator in IBM Message Service Client for .NET.

Quando viene creato un iteratore, la posizione del cursore si trova prima del primo oggetto. Un'applicazione utilizza un iteratore per richiamare ciascun oggetto.

La classe Iterator di IBM Message Service Client for C/C++ è equivalente alla classe Enumerator in Java. XMS .NET è simile a Java e utilizza un'interfaccia IEnumerator.

Un'applicazione può utilizzare un IEnumerator per eseguire le seguenti attività:

- Per ottenere le proprietà di un messaggio
- Per ottenere le coppie nome - valore nel corpo di un messaggio di associazione
- Per sfogliare i messaggi su una coda
- Per ottenere i nomi delle proprietà del messaggio definite da JMS supportate da un collegamento

CCSID (Coded character set identifier)

In XMS .NET, tutte le stringhe vengono trasmesse utilizzando la stringa .NET nativa. Poiché questa ha una codifica fissa, non sono richieste ulteriori informazioni per interpretarla. Quindi, la proprietà XMSC_CLIENT_CCSID non è richiesta per le applicazioni XMS .NET .

Codici di errore e di eccezione XMS

XMS utilizza un intervallo di codici di errore per indicare errori. Questi codici di errore non sono elencati esplicitamente in questa documentazione perché possono variare da release a release. Solo i codici di eccezione XMS (nel formato XMS_X_ ...) vengono documentati perché rimangono gli stessi tra le release di XMS.

Creazione di applicazioni personalizzate

Si creano le proprie applicazioni come si creano le applicazioni di esempio.

Crea la tua applicazione .NET , come descritto in [“Creazione delle applicazioni di esempio .NET” a pagina 19](#), che elenca anche i prerequisiti di cui hai bisogno per creare le tue applicazioni .NET . Per ulteriori informazioni su come creare le applicazioni, utilizzare i makefile forniti per ciascuna applicazione di esempio.

Suggerimento: Per facilitare la diagnosi dei problemi in caso di errore, potrebbe essere utile compilare le applicazioni con i simboli inclusi.

Concetti correlati

Le applicazioni di esempio

Le applicazioni di esempio forniscono una panoramica delle funzioni comuni di ciascuna API. È possibile utilizzarli per verificare l'installazione e l'impostazione del server di messaggistica e per creare le proprie applicazioni.

Riferimenti correlati

.NETInterfacce

Questa sezione documenta le interfacce della classe .NET e le relative proprietà e metodi.

Proprietà degli oggetti XMS

capitolo documenta le proprietà dell'oggetto definite da XMS.

Riconnessione automatica del client IBM MQ tramite XMS

Configurare il proprio client XMS per riconnettersi automaticamente in seguito a un errore di rete, gestore code o server durante l'utilizzo del client IBM WebSphere MQ 7.1 e versioni successive, come provider di messaggi.

Utilizzare le proprietà `WMQ_CONNECTION_NAME_LIST` e `WMQ_CLIENT_RECONNECT_OPTIONS` della classe `MQConnectionFactory` per configurare una connessione client per la riconnessione automatica. La riconnessione client automatica riconnette un client dopo un errore di connessione o come opzione dopo l'arresto del gestore code. La progettazione di alcune applicazioni client le rende inadatte per la riconnessione automatica.

Le connessioni client ricollegabili automaticamente diventano ricollegabili una volta stabilita la connessione.

Nota: Le proprietà Opzioni di riconnessione client, Timeout riconnessione cliente Elenco nomi connessione possono essere impostate anche mediante CCDT (Client Channel Definitions Table) oppure abilitando la riconnessione client tramite il file `mqclient.ini`.

Nota: Se le proprietà di riconnessione sono impostate sull'oggetto `ConnectionFactory` e anche in CCDT, la regola di precedenza è la seguente. Se il valore predefinito della proprietà dell'elenco dei nomi di connessione è impostato nell'oggetto `ConnectionFactory`, la CCDT ha la precedenza. Se l'elenco dei nomi di connessione non è impostato sul valore predefinito, i valori delle proprietà impostati nell'oggetto `ConnectionFactory` hanno la precedenza. Il valore predefinito dell'elenco nomi di connessione è `localhost(1414)`.

Scrittura di applicazioni XMS .NET

Gli argomenti contenuti in questa sezione forniscono informazioni utili per la scrittura di applicazioni XMS .NET.

Questa sezione fornisce informazioni specifiche per la scrittura di applicazioni XMS .NET. Per informazioni generali sulla scrittura di applicazioni XMS, consultare [“Scrittura di applicazioni XMS” a pagina 20](#).

Questa sezione contiene i seguenti argomenti:

- [“Tipi di dati per .NET” a pagina 44](#)
- [“Operazioni gestite e non gestite in .NET” a pagina 45](#)
- [“Destinazioni in .NET” a pagina 46](#)
- [“Proprietà in .NET” a pagina 47](#)
- [“Gestione di proprietà non esistenti in .NET” a pagina 47](#)
- [“Gestione degli errori in .NET” a pagina 48](#)
- [“Listener di messaggi ed eccezioni in .NET” a pagina 48](#)

Concetti correlati

Scrittura di applicazioni XMS

Gli argomenti in questa sezione forniscono informazioni utili per la scrittura di applicazioni XMS.

Riferimenti correlati

.NETInterfacce

Questo sezione documenta le interfacce della classe .NET e le relative proprietà e metodi.

Tipi di dati per .NET

XMS .NET supporta `System.Boolean`, `System.Byte`, `System.SByte`, `System.Char`, `System.String`, `System.Single`, `System.Double`, `System.Decimal`, `System.Int16`, `System.Int32`, `System.Int64`, `System.UInt16`, `System.UInt32`, `System.UInt64`, e `System.Object`. I tipi di dati per il XMS .NET sono

diversi dai tipi di dati per XMS C/C + . È possibile utilizzare questo argomento per identificare i corrispondenti tipi di dati.

La seguente tabella mostra i tipi di dati XMS .NET e XMS C/C++ corrispondenti e li descrive brevemente.

<i>Tabella 7. Tipi di dati per XMS .NET e XMS C/C++</i>		
Tipo XMS .NET	tipo C/C++ XMS	Descrizione
System.SByte	xmsSBYTE xmsINT8	Valore a 8 bit con segno
System.Byte	xmsBYTE xmsUINT8	Valore a 8 bit senza segno
System.Int16	xmsINT16 xmsSHORT	Valore a 16 bit con segno
System.UInt16	xmsUINT16 xmsUSHORT	Valore a 16 bit senza segno
System.Int32	xmsINT32 xmsINT	Valore a 32 bit con segno
System.UInt32	xmsUINT32 xmsUINT	Valore a 32 bit senza segno
System.Int64	xmsLONG xmsINT64	Valore a 64 bit con segno
System.UInt64	xmsULONG xmsUINT64	Valore a 64 bit senza segno
System.Char	xmsCHAR16	Carattere a 16 bit senza segno (Unicode per .NET)
System.Single	xmsFLOAT	Float a 32 bit IEEE
System.Double	xmsDOUBLE	Mobile IEEE a 64 bit
System.Boolean	xmsBOOL	Un valore True / False
Non applicabile	xmsCHAR	Valore a 8 bit con o senza segno (il valore con segno o senza segno dipende dalla piattaforma)
System.Decimal	Non applicabile	Numero intero con segno a 96 bit da 10^0 a 10^{28}
System.Object	Non applicabile	Base di tutti i tipi
System.String	Non applicabile	Tipo di stringa

Operazioni gestite e non gestite in .NET

Il codice gestito viene eseguito esclusivamente nell'ambiente di runtime del linguaggio comune .NET ed è interamente dipendente dai servizi forniti da tale runtime. Un'applicazione viene classificata come non gestita se una parte dell'applicazione viene eseguita o richiama i servizi all'esterno dell'ambiente di runtime del linguaggio comune .NET .

Alcune funzionalità avanzate non possono essere attualmente supportate nell'ambiente .NET gestito.

Se l'applicazione richiede alcune funzionalità che non sono attualmente supportate nell'ambiente completamente gestito, è possibile modificare l'applicazione per utilizzare l'ambiente non gestito senza richiedere modifiche sostanziali all'applicazione. Tuttavia, si noti che lo stack XMS utilizza il codice non gestito quando viene eseguita questa selezione.

Connessioni a un gestore code IBM MQ

Le connessioni gestite a WMQ_CM_CLIENT non supporteranno le comunicazioni non TCP e la compressione del canale. Tuttavia, queste connessioni potrebbero essere supportate utilizzando una connessione non gestita (WMQ_CM_CLIENT_UNMANAGED). Per ulteriori informazioni, consultare [Sviluppo di applicazioni .NET](#).

Se si crea una factory di connessione da un oggetto gestito in un ambiente non gestito, è necessario modificare manualmente il valore per la modalità di connessione in XMSC_WMQ_CM_CLIENT_UNMANAGED.

Connessioni a un motore di messaggistica SIB (service integration bus) WebSphere Application Server

Le connessioni a un motore di messaggistica SIB (Service Integration Bus) di WebSphere Application Server che richiedono l'utilizzo del protocollo SSL (incluso HTTPS) non sono attualmente supportate come codice gestito.

Riferimenti correlati

[XMSC_WMQ_CONNECTION_MODE](#)

Destinazioni in .NET

In .NET, le destinazioni vengono create in base al tipo di protocollo e possono essere utilizzate solo sul tipo di protocollo per cui sono create.

Sono fornite due funzioni per la creazione di destinazioni, una per gli argomenti e una per le code:

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

Queste funzioni sono disponibili sui seguenti due oggetti nell'API:

- `ISession`
- `XMSFactoryFactory`

In entrambi i casi, questi metodi possono accettare una stringa di stile URI, che può includere parametri, nel seguente formato:

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

In alternativa, questi metodi possono accettare solo un nome di destinazione, ovvero un nome senza un argomento: `//` o `code://` prefisso e senza parametri.

Ciò significa che la seguente stringa di stile URI:

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

produce lo stesso risultato del seguente nome destinazione:

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

Come per WebSphere Application Server service integration bus JMS, gli argomenti possono essere specificati anche in un formato abbreviato, che include sia *topicname* che *topicspace* ma non può includere parametri:

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

Proprietà in .NET

Un'applicazione .NET utilizza i metodi nell'interfaccia PropertyContext per richiamare e impostare le proprietà degli oggetti.

L'interfaccia PropertyContext incapsula i metodi che acquisisce e imposta le proprietà. Questi metodi vengono ereditati, direttamente o indirettamente, dalle seguenti classi:

- [BytesMessage](#)
- [Connessione](#)
- [ConnectionFactory](#)
- [ConnectionMetaDati](#)
- [Destinazione](#)
- [MapMessage](#)
- [Messaggio](#)
- [MessageConsumer](#)
- [MessageProducer](#)
- [ObjectMessage](#)
- [QueueBrowser](#)
- [sessione](#)
- [StreamMessage](#)
- [TextMessage](#)

Se un'applicazione imposta il valore di una proprietà, il nuovo valore sostituisce qualsiasi valore precedente della proprietà.

Per ulteriori informazioni sulle proprietà XMS, consultare [“Proprietà degli oggetti XMS”](#) a pagina 180.

Per facilità d'uso, i valori e i nomi delle proprietà XMS in XMS sono predefiniti come costanti pubbliche in una struttura denominata XMSC. I nomi di queste costanti sono nel formato XMSC.*costante*; ad esempio, XMSC.USERID (una costante del nome proprietà) e XMSC.DELIVERY_AS_APP (una costante valore).

Inoltre, è possibile accedere alle costanti IBM MQ utilizzando IBM.XMS.MQC. Se IBM.XMS è già importato, è possibile accedere ai valori per queste proprietà nel modulo MQC.*costante*. Ad esempio, MQC.MQRO_COA_WITH_FULL_DATA.

Inoltre, se si dispone di un'applicazione ibrida che utilizza sia le classi XMS .NET che IBM MQ per .NET e che importa entrambi IBM.XMS e IBM.WMQ WMQ, quindi, è necessario qualificare completamente lo spazio dei nomi struct MQC per garantire che ogni ricorrenza sia univoca.

Alcune funzionalità avanzate non sono attualmente supportate nell'ambiente .NET gestito. Fare riferimento a [“Operazioni gestite e non gestite in .NET”](#) a pagina 45 per ulteriori dettagli.

Gestione di proprietà non esistenti in .NET

La gestione delle proprietà non esistenti in XMS .NET è sostanzialmente coerente con le specifiche JMS e anche con le implementazioni C e C++ di XMS.

In JMS, l'accesso a una proprietà non esistente può generare un'eccezione di sistema Java quando un metodo tenta di convertire il valore non esistente (null) nel tipo richiesto. Se una proprietà non esiste, si verificano le seguenti eccezioni:

- La proprietà `getString` la proprietà `getObject` restituiscono un valore `null`
- La proprietà `getBoolean` restituisce `false` perché `Boolean.valueOf(null)` restituisce `false`
- `getIntProperty` etc. throw `java.lang.NumberFormatException` perché `Integer.valueOf(null)` genera l'eccezione

Se una proprietà non esiste in XMS .NET , si verificano le seguenti eccezioni:

- La proprietà `GetString` la proprietà `GetObject`(e la proprietà `GetBytes`) restituiscono un valore `null` (che è uguale a Java)
- `GetBoolean` la proprietà genera `System.NullReferenceException`
- `GetIntProperty` etc. throws `System.NullReferenceException`

Questa implementazione è diversa da quella di Java, ma è ampiamente coerente con le specifiche JMS e con le interfacce XMS C e C + +. Come l'implementazione di Java , XMS .NET propaga tutte le eccezioni dalla chiamata `System.Convert` al chiamante. A differenza di Java , tuttavia, XMS genera esplicitamente le eccezioni `NullReferenceExceptions` piuttosto che utilizzare semplicemente il comportamento nativo del framework .NET passando `null` alle routine di conversione del sistema. Se la tua applicazione imposta una proprietà su una stringa come "abc" e richiama la proprietà `GetInt`, la `System.FormatException` generata da `Convert.ToInt32("abc")` viene propagato al chiamante, che è congruente con Java. `MessageFormat` l'eccezione viene generata solo se i tipi utilizzati per `setProperty` e `getProperty` non sono compatibili. Questo comportamento è coerente anche con Java.

Gestione degli errori in .NET

Le eccezioni XMS .NET sono tutte derivate da `System.Exception`. Le chiamate al metodo XMS possono generare eccezioni XMS specifiche come l'eccezione `MessageFormat`, `XMSEExceptions` generali o le eccezioni di sistema come l'eccezione `NullReference`.

Scrivere le applicazioni per rilevare questi errori, in blocchi `catch` specifici o in blocchi `catch System.Exception` generali, in base ai requisiti delle applicazioni.

Listener di messaggi ed eccezioni in .NET

Un'applicazione .NET utilizza un listener di messaggi per ricevere messaggi in modo asincrono e utilizza un listener di eccezione per ricevere una notifica asincrona di un problema con una connessione.

La funzionalità dei listener di messaggi e di eccezioni è la stessa per .NET e per C + +. Tuttavia, esistono alcune piccole differenze di implementazione.

Listener di messaggi in .NET

Per ricevere i messaggi in modo asincrono, è necessario completare la seguente procedura:

1. Definire un metodo che corrisponda alla firma del delegato del listener messaggi. Il metodo definito può essere un metodo statico o di istanza e può essere definito in qualsiasi classe accessibile. La firma del delegato è la seguente:

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

e quindi è possibile definire il metodo come:

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

2. Creare un'istanza di questo metodo come un delegato utilizzando un metodo simile al seguente:

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

3. Registrare il delegato con uno o più consumer impostandolo sulla proprietà `MessageListener` del consumer:


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

È possibile rimuovere il delegato impostando nuovamente MessageListener su null:

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

Listener di eccezione in .NET

Il listener di eccezioni funziona in modo molto simile al listener di messaggi, ma ha una diversa definizione di delegato e viene assegnato alla connessione piuttosto che al consumer di messaggi. Questo è lo stesso di C + +.

1. Definire il metodo. La firma del delegato è la seguente:

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

e quindi il metodo definito potrebbe essere:

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

2. Istanziare questo metodo come un delegato utilizzando qualcosa di simile a:

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

3. Registrare il delegato con il collegamento impostandone la proprietà ExceptionListener :

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

È possibile rimuovere il delegato reimpostando ExceptionListener su:

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

Quando non rimangono riferimenti, le eccezioni o i messaggi vengono eliminati automaticamente dal raccogliatore dati inutilizzati dei sistemi.

Il seguente è un codice di esempio:

```
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);
}
}

```

Utilizzo degli oggetti gestiti

Gli argomenti in questa sezione forniscono informazioni sugli oggetti gestiti. Le applicazioni XMS possono recuperare le definizioni degli oggetti da un repository di oggetti amministrato centrale e utilizzarli per creare factory di connessione e destinazioni.

Questa sezione fornisce informazioni utili per la creazione e la gestione di oggetti gestiti, descrivendo i tipi di repository di oggetti gestiti supportati da XMS . La sezione spiega anche come un'applicazione XMS effettua una connessione a un repository di oggetti gestiti per richiamare gli oggetti gestiti richiesti.

Questa sezione contiene i seguenti argomenti:

- [“Tipi supportati di repository di oggetti gestiti” a pagina 50](#)
- [“Associazione di proprietà per gli oggetti gestiti” a pagina 52](#)
- [“Proprietà richieste per gli oggetti ConnectionFactory gestiti” a pagina 54](#)
- [“Proprietà richieste per gli oggetti Destinazione gestiti” a pagina 56](#)
- [“Creazione di oggetti gestiti” a pagina 58](#)
- [“Oggetti InitialContext” a pagina 60](#)
- [“Proprietà InitialContext” a pagina 61](#)
- [“Formato URI per contesti iniziali XMS” a pagina 62](#)
- [“Servizio Web Ricerca JNDI” a pagina 64](#)
- [“Richiamo di oggetti gestiti” a pagina 65](#)

Concetti correlati

oggetti gestiti

Utilizzando gli oggetti gestiti, è possibile gestire le impostazioni di connessione utilizzate dalle applicazioni client da gestire da un repository centrale. Un'applicazione richiama le definizioni degli oggetti dal repository centrale e le usa per creare oggetti `ConnectionFactory` e `Destination` . Utilizzando gli oggetti gestiti, è possibile disaccoppiare le applicazioni dalle risorse che utilizzano al runtime.

Tipi supportati di repository di oggetti gestiti

Gli oggetti gestiti del file system e LDAP possono essere utilizzati per connettersi a IBM MQ e WebSphere Application Server, mentre la denominazione COS può essere utilizzata per connettersi solo a WebSphere Application Server.

Le directory degli oggetti del file system assumono la forma di oggetti Java Naming Directory Interface (JNDI) serializzati. Gli indirizzari oggetto LDAP sono indirizzari che contengono oggetti JNDI . Le directory

di oggetti del file system e di LDAP possono essere gestite utilizzando lo strumento JMSAdmin, fornito con IBM WebSphere MQ 6.0, o IBM MQ Explorer, fornito con IBM WebSphere MQ 7.0 e versioni successive. Sia il file system che le directory di oggetto LDAP possono essere utilizzate per gestire connessioni client centralizzando le destinazioni e i factory di connessione IBM WebSphere MQ. L'amministratore di rete può distribuire più applicazioni che fanno riferimento allo stesso repository centrale e che vengono aggiornate automaticamente per riflettere le modifiche alle impostazioni di connessione apportate nel repository centrale.

Una directory di denominazione COS contiene destinazioni e factory di connessione WebSphere Application Server service integration bus e può essere gestita utilizzando la console di gestione WebSphere Application Server. Per un'applicazione XMS per richiamare gli oggetti dalla directory di denominazione COS, è necessario distribuire un servizio Web di ricerca JNDI. Questo servizio Web non è disponibile su tutti gli WebSphere Application Server service integration technologies. Fare riferimento alla documentazione del prodotto per maggiori dettagli.

Nota: Riavviare le connessioni dell'applicazione per rendere effettive le modifiche alla directory dell'oggetto.

Concetti correlati

Associazione di proprietà per gli oggetti gestiti

Per consentire alle applicazioni di utilizzare le definizioni oggetto di destinazione e factory di connessione IBM MQ JMS e WebSphere Application Server, le proprietà ... richiamate da tali definizioni devono essere associate alle proprietà ... XMS corrispondenti che possono essere impostate sulle destinazioni e sulle produzioni connessioni XMS.

Proprietà InitialContext

I parametri del costruttore InitialContext includono l'ubicazione del repository di oggetti gestiti, forniti come URI (uniform resource indicator). Affinché un'applicazione stabilisca una connessione al repository, potrebbe essere necessario fornire più informazioni rispetto a quelle contenute nell'URI.

Formato URI per contesti iniziali XMS

L'ubicazione del repository di oggetti gestiti viene fornita come URI (uniform resource indicator). Il formato dell'URI dipende dal tipo di contesto.

Servizio Web Ricerca JNDI

Per accedere a una directory di denominazione COS da XMS, un JNDI servizio Web di ricerca deve essere distribuito su un server WebSphere Application Server service integration bus. Questo servizio web converte le informazioni Java dal servizio di denominazione COS in un modulo che le applicazioni XMS possono leggere.

Richiamo di oggetti gestiti

XMS richiama un oggetto gestito dal repository utilizzando l'indirizzo fornito quando viene creato l'oggetto InitialContext o nelle proprietà InitialContext.

oggetti gestiti

Utilizzando gli oggetti gestiti, è possibile gestire le impostazioni di connessione utilizzate dalle applicazioni client da gestire da un repository centrale. Un'applicazione richiama le definizioni degli oggetti dal repository centrale e le usa per creare oggetti `ConnectionFactory` e `Destination`. Utilizzando gli oggetti gestiti, è possibile disaccoppiare le applicazioni dalle risorse che utilizzano al runtime.

Attività correlate

Creazione di oggetti gestiti

Le definizioni dell'oggetto `ConnectionFactory` e `Destination` richieste dalle applicazioni XMS per stabilire una connessione a un server di messaggistica devono essere create utilizzando gli strumenti di gestione appropriati.

Oggetti InitialContext

Un'applicazione deve creare un contesto iniziale da utilizzare per stabilire una connessione al repository di oggetti gestiti per richiamare gli oggetti gestiti richiesti.

Riferimenti correlati

Proprietà richieste per gli oggetti `ConnectionFactory` gestiti

Quando un'applicazione crea una factory di connessione, è necessario definire un numero di proprietà per creare una connessione a un server di messaggistica.

Proprietà richieste per gli oggetti Destinazione gestiti

Un'applicazione che sta creando una destinazione deve impostare diverse proprietà dell'applicazione su un oggetto Destinazione gestito.

Associazione di proprietà per gli oggetti gestiti

Per consentire alle applicazioni di utilizzare le definizioni oggetto di destinazione e factory di connessione IBM MQ JMS e WebSphere Application Server , le propriet ... richiamate da tali definizioni devono essere associate alle propriet ... XMS corrispondenti che possono essere impostate sulle destinazioni e sulle produzioni connessioni XMS .

Per creare, ad esempio, un factory di connessione XMS con proprietà richiamate da un factory di connessione JMS IBM MQ , le proprietà devono essere associate tra i due.

Tutte le associazioni di proprietà vengono eseguite automaticamente.

La tabella riportata di seguito illustra le associazioni tra alcune delle proprietà più comuni delle factory di connessione e delle destinazioni. Le proprietà mostrate in questa tabella sono solo una piccola serie di esempi e non tutte le proprietà mostrate sono rilevanti per tutti i tipi di connessione e server.

<i>Tabella 8. Esempi di mappatura dei nomi per la factory di connessione e le proprietà di destinazione</i>		
Nome proprietà IBM MQ JMS	XMS nome proprietà.	WebSphere Application Server service integration bus nome proprietà.
PERSISTENZA (PER)	<u>MODALITÀ_DISTRIBUZIONE_XMSCD</u>	
SCADENZA (SCAD)	<u>XMSC_TIME_TO_LIVE</u>	
PRIORITÀ (PRI)	<u>PRIORITÀ_XMSC_XX_ENCODE_CASE_CAPS_LOCK_OFF</u>	
	<u>Nome_HOST_WPM_XMSC</u>	serverName
	<u>XMSC_WPM_BUS_NAME</u>	busName
	<u>SPACE_XMSC_WPM_TOPIC_XX_ENCODE_CASE_ONE spazio</u>	topicName

<i>Tabella 9. Proprietà di XMS .NET</i>					
Proprietà	Tipo oggetto				
	CF	QCF	TCF	Coda	Argomento
<u>APPLICATIONNAME</u>	Y	Y	Y	N.d.	N.d.
<u>ECCEZIONE</u>	Y	Y	Y	N.d.	N.d.
<u>CCDTURL</u>	Y	Y	Y	N.d.	N.d.
<u>CHANNEL</u>	Y	Y	Y	N.d.	N.d.
<u>ConnectionNameList</u>	Y	Y	Y	N.d.	N.d.
<u>CLIENTRECONNECTOPTIONS</u>	Y	Y	Y	N.d.	N.d.

Tabella 9. Proprietà di XMS .NET (Continua)

Proprietà	Tipo oggetto				
	CF	QCF	TCF	Coda	Argomento
<u>CLIENTRECONNECTTIMEOUT</u>	Y	Y	Y	N.d.	N.d.
<u>CLIENTID</u>	N.d.	Y	N.d.	N.d.	N.d.
<u>COMPHDR</u> "1" a pagina 53	Y	N.d.	Y	N.d.	N.d.
<u>COMPMSG</u> "1" a pagina 53	Y	Y	Y	N.d.	N.d.
<u>CONNOPT</u> "1" a pagina 53	Y	Y	Y	N.d.	N.d.
<u>CONNTAG</u> "1" a pagina 53	Y	Y	Y	N.d.	N.d.
<u>DESCRIZIONE</u> "1" a pagina 53	N.d.	Y	N.d.	Y	Y
<u>SCADENZA</u> "1" a pagina 53	N.d.	N.d.	N.d.	Y	Y
<u>FAILIFQUIESCE</u>	Y	Y	Y	Y	Y
<u>Nome host</u>	N.d.	Y	N.d.	N.d.	N.d.
<u>LOCALADDRESS</u>	N.d.	Y	N.d.	N.d.	N.d.
<u>PERSISTENCE</u>	N.d.	N.d.	N.d.	Y	Y
<u>PORTA</u>	N.d.	Y	N.d.	N.d.	N.d.
<u>Priorità</u> "1" a pagina 53	N.d.	N.d.	N.d.	Y	Y
<u>VERSIONEPROVIDER</u> "1" a pagina 53	N.d.	Y	N.d.	N.d.	N.d.
<u>QMANAGER</u>	Y	Y	Y	Y	N.d.
<u>Coda</u> "1" a pagina 53	N.d.	N.d.	N.d.	Y	N.d.
<u>SHARECONVALLOWED</u>	Y	Y	Y	N.d.	N.d.
<u>ARGOMENTO</u> "1" a pagina 53	N.d.	N.d.	N.d.	N.d.	Y
<u>Transport</u> "1" a pagina 53	N.d.	Y	N.d.	N.d.	N.d.

Nota:

1. Queste proprietà non dispongono di proprietà a livello di applicazione ma possono essere facoltativamente impostate utilizzando le proprietà gestite.

Concetti correlati

Tipi supportati di repository di oggetti gestiti

Gli oggetti gestiti del file system e LDAP possono essere utilizzati per connettersi a IBM MQ e WebSphere Application Server, mentre la denominazione COS può essere utilizzata per connettersi solo a WebSphere Application Server.

Proprietà InitialContext

I parametri del costruttore InitialContext includono l'ubicazione del repository di oggetti gestiti, forniti come URI (uniform resource indicator). Affinché un'applicazione stabilisca una connessione al repository, potrebbe essere necessario fornire più informazioni rispetto a quelle contenute nell'URI.

Formato URI per contesti iniziali XMS

L'ubicazione del repository di oggetti gestiti viene fornita come URI (uniform resource indicator). Il formato dell'URI dipende dal tipo di contesto.

Servizio Web Ricerca JNDI

Per accedere a una directory di denominazione COS da XMS, un JNDI servizio Web di ricerca deve essere distribuito su un server WebSphere Application Server service integration bus . Questo servizio web converte le informazioni Java dal servizio di denominazione COS in un modulo che le applicazioni XMS possono leggere.

Richiamo di oggetti gestiti

XMS richiama un oggetto gestito dal repository utilizzando l'indirizzo fornito quando viene creato l'oggetto InitialContext o nelle proprietà InitialContext .

Attività correlate

Creazione di oggetti gestiti

Le definizioni dell'oggetto ConnectionFactory e Destination richieste dalle applicazioni XMS per stabilire una connessione a un server di messaggistica devono essere create utilizzando gli strumenti di gestione appropriati.

Oggetti InitialContext

Un'applicazione deve creare un contesto iniziale da utilizzare per stabilire una connessione al repository di oggetti gestiti per richiamare gli oggetti gestiti richiesti.

Riferimenti correlati

Proprietà richieste per gli oggetti ConnectionFactory gestiti

Quando un'applicazione crea una factory di connessione, è necessario definire un numero di proprietà per creare una connessione a un server di messaggistica.

Proprietà richieste per gli oggetti Destinazione gestiti

Un'applicazione che sta creando una destinazione deve impostare diverse proprietà dell'applicazione su un oggetto Destinazione gestito.

IDestination (per l'interfaccia .NET)

Una destinazione è il punto in cui un'applicazione invia messaggi o è un'origine da cui un'applicazione riceve messaggi o entrambi.

Proprietà della destinazione

Una panoramica delle proprietà dell'oggetto Destinazione, con link a informazioni di riferimento più dettagliate.

IConnectionFactory (per l'interfaccia .NET)

Un'applicazione utilizza una produzione connessioni per creare una connessione.

Proprietà di ConnectionFactory

Una panoramica delle proprietà dell'oggetto ConnectionFactory , con link a informazioni di riferimento più dettagliate.

Proprietà richieste per gli oggetti ConnectionFactory gestiti

Quando un'applicazione crea una factory di connessione, è necessario definire un numero di proprietà per creare una connessione a un server di messaggistica.

Le proprietà elencate nelle seguenti tabelle sono il minimo richiesto per un'applicazione da impostare per creare una connessione a un server di messaggistica. Se si desidera personalizzare il modo in cui viene creata una connessione, l'applicazione può impostare qualsiasi proprietà aggiuntiva dell'oggetto `ConnectionFactory` come necessario. Per ulteriori informazioni, fare riferimento a “[Proprietà di ConnectionFactory](#)” a pagina 181. Viene incluso un elenco completo delle proprietà disponibili.

Connessione a un gestore code IBM MQ

<i>Tabella 10. Impostazioni delle proprietà per gli oggetti ConnectionFactory gestiti per le connessioni a un gestore code IBM MQ</i>	
ObbligatorioXMS	Proprietà IBM MQ JMS equivalente obbligatoria
<code>XMSC_CONNECTION_TYPE</code>	XMS consente di eseguire questa operazione dal nome della classe factory di connessione e dalla proprietà <code>TRANSPORT</code> (<code>TRAN</code>).
<code>NOME_HOST_WMQ_XMSC</code>	NOME HOST (<code>HOST</code>)
<code>XMSC_WMQ_PORT</code>	PORTA
<code>XMSC_WMQ_QUEUE_MANAGER</code>	Il nome del gestore code

Connessione in tempo reale a un broker

<i>Tabella 11. Impostazioni di proprietà per oggetti ConnectionFactory gestiti per connessioni in tempo reale a un broker</i>	
ObbligatorioXMS	Proprietà IBM MQ JMS equivalente obbligatoria
<code>XMSC_CONNECTION_TYPE</code>	XMS consente di eseguire questa operazione dal nome della classe factory di connessione e dalla proprietà <code>TRANSPORT</code> (<code>TRAN</code>).
<code>XMSC_RTT_NOME_HOST</code>	NOME HOST (<code>HOST</code>)
<code>PORTA_RTT_XMSC</code>	PORTA

Connessione a un WebSphere Application Server service integration bus

<i>Tabella 12. Impostazioni delle proprietà ... per gli oggetti ConnectionFactory gestiti per le connessioni a un WebSphere Application Server service integration bus</i>	
XMS proprietà	Descrizione
<code>XMSC_CONNECTION_TYPE</code>	Il tipo di server di messaggistica a cui si connette un'applicazione.. Ciò è determinato dal nome della classe factory di connessione.
<code>XMSC_WPM_BUS_NAME</code>	Per una factory di connessione, il nome del bus di integrazione del servizio a cui si connette l'applicazione oppure, per una destinazione, il nome del bus di integrazione del servizio in cui esiste la destinazione.

Concetti correlati

[Tipi supportati di repository di oggetti gestiti](#)

Gli oggetti gestiti del file system e LDAP possono essere utilizzati per connettersi a IBM MQ e WebSphere Application Server, mentre la denominazione COS può essere utilizzata per connettersi solo a WebSphere Application Server.

[Associazione di proprietà per gli oggetti gestiti](#)

Per consentire alle applicazioni di utilizzare le definizioni oggetto di destinazione e factory di connessione IBM MQ JMS e WebSphere Application Server , le propriet ... richiamate da tali definizioni devono essere associate alle propriet ... XMS corrispondenti che possono essere impostate sulle destinazioni e sulle produzioni connessioni XMS .

Proprietà InitialContext

I parametri del costruttore InitialContext includono l'ubicazione del repository di oggetti gestiti, forniti come URI (uniform resource indicator). Affinché un'applicazione stabilisca una connessione al repository, potrebbe essere necessario fornire più informazioni rispetto a quelle contenute nell'URI.

Formato URI per contesti iniziali XMS

L'ubicazione del repository di oggetti gestiti viene fornita come URI (uniform resource indicator). Il formato dell'URI dipende dal tipo di contesto.

Servizio Web Ricerca JNDI

Per accedere a una directory di denominazione COS da XMS, un JNDI servizio Web di ricerca deve essere distribuito su un server WebSphere Application Server service integration bus . Questo servizio web converte le informazioni Java dal servizio di denominazione COS in un modulo che le applicazioni XMS possono leggere.

Richiamo di oggetti gestiti

XMS richiama un oggetto gestito dal repository utilizzando l'indirizzo fornito quando viene creato l'oggetto InitialContext o nelle proprietà InitialContext .

Connessioni protette a un gestore code IBM MQ

Per consentire a un'applicazione di XMS .NET di stabilire connessioni sicure a un gestore code IBM MQ , le proprietà pertinenti devono essere definite nell'oggetto ConnectionFactory .

Connessioni sicure a un motore di messaggistica WebSphere Application Server service integration bus

Per abilitare un'applicazione XMS .NET per stabilire connessioni sicure a un motore di messaggistica WebSphere Application Server service integration bus , le proprietà pertinenti devono essere definite nell'oggetto ConnectionFactory .

Attività correlate

Creazione di oggetti gestiti

Le definizioni dell'oggetto ConnectionFactory e Destination richieste dalle applicazioni XMS per stabilire una connessione a un server di messaggistica devono essere create utilizzando gli strumenti di gestione appropriati.

Oggetti InitialContext

Un'applicazione deve creare un contesto iniziale da utilizzare per stabilire una connessione al repository di oggetti gestiti per richiamare gli oggetti gestiti richiesti.

Riferimenti correlati

Proprietà richieste per gli oggetti Destinazione gestiti

Un'applicazione che sta creando una destinazione deve impostare diverse proprietà dell'applicazione su un oggetto Destinazione gestito.

IConnectionFactory (per l'interfaccia .NET)

Un'applicazione utilizza una produzione connessioni per creare una connessione.

Proprietà di ConnectionFactory

Una panoramica delle proprietà dell'oggetto ConnectionFactory , con link a informazioni di riferimento più dettagliate.

Proprietà richieste per gli oggetti Destinazione gestiti

Un'applicazione che sta creando una destinazione deve impostare diverse proprietà dell'applicazione su un oggetto Destinazione gestito.

Tabella 13. Impostazioni di proprietà per gli oggetti di destinazione gestiti

Tipo di connessione	Proprietà	Descrizione
IBM MQ Gestore code	CODA (QU) ARGOMENTO (SUPERIORE)	La coda a cui si desidera connettersi L'argomento utilizzato dall'applicazione come destinazione
Connessione in tempo reale a un broker	ARGOMENTO (SUPERIORE)	L'argomento utilizzato dall'applicazione come destinazione
WebSphere Application Server service integration bus	topicName queueName	Se l'applicazione si connette a un argomento Se l'applicazione si connette a una coda

Concetti correlati

Tipi supportati di repository di oggetti gestiti

Gli oggetti gestiti del file system e LDAP possono essere utilizzati per connettersi a IBM MQ e WebSphere Application Server, mentre la denominazione COS può essere utilizzata per connettersi solo a WebSphere Application Server.

Associazione di proprietà per gli oggetti gestiti

Per consentire alle applicazioni di utilizzare le definizioni oggetto di destinazione e factory di connessione IBM MQ JMS e WebSphere Application Server, le propriet ... richiamate da tali definizioni devono essere associate alle propriet ... XMS corrispondenti che possono essere impostate sulle destinazioni e sulle produzioni connessioni XMS.

Proprietà InitialContext

I parametri del costruttore InitialContext includono l'ubicazione del repository di oggetti gestiti, forniti come URI (uniform resource indicator). Affinché un'applicazione stabilisca una connessione al repository, potrebbe essere necessario fornire più informazioni rispetto a quelle contenute nell'URI.

Formato URI per contesti iniziali XMS

L'ubicazione del repository di oggetti gestiti viene fornita come URI (uniform resource indicator). Il formato dell'URI dipende dal tipo di contesto.

Servizio Web Ricerca JNDI

Per accedere a una directory di denominazione COS da XMS, un JNDI servizio Web di ricerca deve essere distribuito su un server WebSphere Application Server service integration bus. Questo servizio web converte le informazioni Java dal servizio di denominazione COS in un modulo che le applicazioni XMS possono leggere.

Richiamo di oggetti gestiti

XMS richiama un oggetto gestito dal repository utilizzando l'indirizzo fornito quando viene creato l'oggetto InitialContext o nelle proprietà InitialContext.

Attività correlate

Creazione di oggetti gestiti

Le definizioni dell'oggetto ConnectionFactory e Destination richieste dalle applicazioni XMS per stabilire una connessione a un server di messaggistica devono essere create utilizzando gli strumenti di gestione appropriati.

Oggetti InitialContext

Un'applicazione deve creare un contesto iniziale da utilizzare per stabilire una connessione al repository di oggetti gestiti per richiamare gli oggetti gestiti richiesti.

Riferimenti correlati

Proprietà richieste per gli oggetti ConnectionFactory gestiti

Quando un'applicazione crea una factory di connessione, è necessario definire un numero di proprietà per creare una connessione a un server di messaggistica.

IDestination (per l'interfaccia .NET)

Una destinazione è il punto in cui un'applicazione invia messaggi o è un'origine da cui un'applicazione riceve messaggi o entrambi.

Proprietà della destinazione

Una panoramica delle proprietà dell'oggetto Destinazione, con link a informazioni di riferimento più dettagliate.

Creazione di oggetti gestiti

Le definizioni dell'oggetto ConnectionFactory e Destination richieste dalle applicazioni XMS per stabilire una connessione a un server di messaggistica devono essere create utilizzando gli strumenti di gestione appropriati.

Prima di iniziare

Per ulteriori dettagli sui diversi tipi di repository di oggetti gestiti supportati da XMS , consultare [“Tipi supportati di repository di oggetti gestiti”](#) a pagina 50.

Informazioni su questa attività

Per creare gli oggetti amministrati per IBM MQ utilizzare lo strumento di gestione JMS (JMSAdmin) IBM MQ Explorer o IBM MQ .

Per creare gli oggetti gestiti per IBM MQ o IBM Integration Bus, utilizzare lo strumento di amministrazione JMS IBM MQ (JMSAdmin).

Per creare oggetti gestiti per WebSphere Application Server service integration bus, utilizzare la console di gestione WebSphere Application Server .

La seguente procedura riepiloga le operazioni da eseguire per creare gli oggetti gestiti.

Procedura

1. Creare una factory di connessione e definire le proprietà necessarie per creare una connessione dall'applicazione al server scelto.

Le proprietà minime richieste da XMS per stabilire una connessione sono definite in [“Proprietà richieste per gli oggetti ConnectionFactory gestiti”](#) a pagina 54.

2. Creare la destinazione richiesta sul server di messaggistica, a cui si connette l'applicazione:

- Per una connessione a un gestore code IBM MQ , creare una coda o un argomento.
- Per una connessione in tempo reale a un broker, creare un argomento.
- Per una connessione a un WebSphere Application Server service integration bus, creare una coda o un argomento.

Le proprietà minime richieste da XMS per stabilire una connessione sono definite in [“Proprietà richieste per gli oggetti Destinazione gestiti”](#) a pagina 56.

Concetti correlati

Tipi supportati di repository di oggetti gestiti

Gli oggetti gestiti del file system e LDAP possono essere utilizzati per connettersi a IBM MQ e WebSphere Application Server, mentre la denominazione COS può essere utilizzata per connettersi solo a WebSphere Application Server.

Associazione di proprietà per gli oggetti gestiti

Per consentire alle applicazioni di utilizzare le definizioni oggetto di destinazione e factory di connessione IBM MQ JMS e WebSphere Application Server , le propriet ... richiamate da tali definizioni devono essere associate alle propriet ... XMS corrispondenti che possono essere impostate sulle destinazioni e sulle produzioni connessioni XMS .

Proprietà InitialContext

I parametri del costruttore InitialContext includono l'ubicazione del repository di oggetti gestiti, forniti come URI (uniform resource indicator). Affinché un'applicazione stabilisca una connessione al repository, potrebbe essere necessario fornire più informazioni rispetto a quelle contenute nell'URI.

Formato URI per contesti iniziali XMS

L'ubicazione del repository di oggetti gestiti viene fornita come URI (uniform resource indicator). Il formato dell'URI dipende dal tipo di contesto.

Servizio Web Ricerca JNDI

Per accedere a una directory di denominazione COS da XMS, un JNDI servizio Web di ricerca deve essere distribuito su un server WebSphere Application Server service integration bus . Questo servizio web converte le informazioni Java dal servizio di denominazione COS in un modulo che le applicazioni XMS possono leggere.

Richiamo di oggetti gestiti

XMS richiama un oggetto gestito dal repository utilizzando l'indirizzo fornito quando viene creato l'oggetto InitialContext o nelle proprietà InitialContext .

oggetti gestiti

Utilizzando gli oggetti gestiti, è possibile gestire le impostazioni di connessione utilizzate dalle applicazioni client da gestire da un repository centrale. Un'applicazione richiama le definizioni degli oggetti dal repository centrale e le usa per creare oggetti ConnectionFactory e Destination . Utilizzando gli oggetti gestiti, è possibile disaccoppiare le applicazioni dalle risorse che utilizzano al runtime.

ConnectionFactories e oggetti Connection

Un oggetto ConnectionFactory fornisce un template che un'applicazione utilizza per creare un oggetto Connection. L'applicazione utilizza l'oggetto Connection per creare un oggetto Session.

Connessione a un SIB (Service Integration Bus)

Un'applicazione XMS può connettersi a un SIB (Service Integration Bus) WebSphere Application Server utilizzando una connessione TCP/IP diretta o utilizzando HTTP su TCP/IP.

Attività correlate

Oggetti InitialContext

Un'applicazione deve creare un contesto iniziale da utilizzare per stabilire una connessione al repository di oggetti gestiti per richiamare gli oggetti gestiti richiesti.

Riferimenti correlati

Proprietà richieste per gli oggetti ConnectionFactory gestiti

Quando un'applicazione crea una factory di connessione, è necessario definire un numero di proprietà per creare una connessione a un server di messaggistica.

Proprietà richieste per gli oggetti Destinazione gestiti

Un'applicazione che sta creando una destinazione deve impostare diverse proprietà dell'applicazione su un oggetto Destinazione gestito.

IConnectionFactory (per l'interfaccia .NET)

Un'applicazione utilizza una produzione connessioni per creare una connessione.

Proprietà di ConnectionFactory

Una panoramica delle proprietà dell'oggetto ConnectionFactory , con link a informazioni di riferimento più dettagliate.

IDestination (per l'interfaccia .NET)

Una destinazione è il punto in cui un'applicazione invia messaggi o è un'origine da cui un'applicazione riceve messaggi o entrambi.

Proprietà della destinazione

Una panoramica delle proprietà dell'oggetto Destinazione, con link a informazioni di riferimento più dettagliate.

Oggetti InitialContext

Un'applicazione deve creare un contesto iniziale da utilizzare per stabilire una connessione al repository di oggetti gestiti per richiamare gli oggetti gestiti richiesti.

Informazioni su questa attività

Un oggetto InitialContext incapsula una connessione al repository. L'API XMS fornisce i metodi per eseguire le seguenti attività:

- Crea un oggetto InitialContext
- Cercare un oggetto gestito nel repository di oggetti amministrato.

Per ulteriori dettagli relativi alla creazione di un oggetto InitialContext , consultare [“InitialContext”](#) a pagina 111 per .NET e [“Proprietà di InitialContext”](#) a pagina 190.

Concetti correlati

Tipi supportati di repository di oggetti gestiti

Gli oggetti gestiti del file system e LDAP possono essere utilizzati per connettersi a IBM MQ e WebSphere Application Server, mentre la denominazione COS può essere utilizzata per connettersi solo a WebSphere Application Server.

Associazione di proprietà per gli oggetti gestiti

Per consentire alle applicazioni di utilizzare le definizioni oggetto di destinazione e factory di connessione IBM MQ JMS e WebSphere Application Server , le propriet ... richiamate da tali definizioni devono essere associate alle propriet ... XMS corrispondenti che possono essere impostate sulle destinazioni e sulle produzioni connessioni XMS .

Proprietà InitialContext

I parametri del costruttore InitialContext includono l'ubicazione del repository di oggetti gestiti, forniti come URI (uniform resource indicator). Affinché un'applicazione stabilisca una connessione al repository, potrebbe essere necessario fornire più informazioni rispetto a quelle contenute nell'URI.

Formato URI per contesti iniziali XMS

L'ubicazione del repository di oggetti gestiti viene fornita come URI (uniform resource indicator). Il formato dell'URI dipende dal tipo di contesto.

Servizio Web Ricerca JNDI

Per accedere a una directory di denominazione COS da XMS, un JNDI servizio Web di ricerca deve essere distribuito su un server WebSphere Application Server service integration bus . Questo servizio web converte le informazioni Java dal servizio di denominazione COS in un modulo che le applicazioni XMS possono leggere.

Richiamo di oggetti gestiti

XMS richiama un oggetto gestito dal repository utilizzando l'indirizzo fornito quando viene creato l'oggetto InitialContext o nelle proprietà InitialContext .

Attività correlate

Creazione di oggetti gestiti

Le definizioni dell'oggetto ConnectionFactory e Destination richieste dalle applicazioni XMS per stabilire una connessione a un server di messaggistica devono essere create utilizzando gli strumenti di gestione appropriati.

Riferimenti correlati

Proprietà richieste per gli oggetti ConnectionFactory gestiti

Quando un'applicazione crea una factory di connessione, è necessario definire un numero di proprietà per creare una connessione a un server di messaggistica.

Proprietà richieste per gli oggetti Destinazione gestiti

Un'applicazione che sta creando una destinazione deve impostare diverse proprietà dell'applicazione su un oggetto Destinazione gestito.

InitialContext (per interfaccia .NET)

Un'applicazione utilizza un oggetto InitialContext per creare oggetti dalle definizioni di oggetto richiamati da un repository di oggetti gestiti.

Proprietà di InitialContext

Una panoramica delle proprietà dell'oggetto InitialContext , con link a informazioni di riferimento più dettagliate.

Proprietà InitialContext

I parametri del costruttore InitialContext includono l'ubicazione del repository di oggetti gestiti, forniti come URI (uniform resource indicator). Affinché un'applicazione stabilisca una connessione al repository, potrebbe essere necessario fornire più informazioni rispetto a quelle contenute nell'URI.

In JNDI e nell'implementazione .NET di XMS, le informazioni aggiuntive vengono fornite in un ambiente Hashtable al costruttore.

L'ubicazione del repository di oggetti amministrati è definita nella proprietà XMSC_IC_URL . Questa proprietà viene generalmente passata alla chiamata di creazione, ma può essere modificata per connettersi a una directory di denominazione diversa prima della ricerca. Per i contesti FileSystem o LDAP, questa proprietà definisce l'indirizzo della directory. Per la denominazione COS, questo è l'indirizzo del servizio Web che utilizza queste proprietà per connettersi alla directory JNDI.

Le seguenti proprietà vengono inoltrate senza modifiche al servizio Web che le utilizzerà per connettersi alla directory JNDI.

- URL_PROVIDER_IC_XMSC
- CREDENZIALI_XMSC_IC_SECURITY_CREDENTIALS
- XMSC_IC_SECURITY_AUTHENTICATION
- XMSC_IC_SECURITY_PRINCIPAL
- PROTOCOLLO_SICUREZZA_XMSC_ICA

Concetti correlati

Tipi supportati di repository di oggetti gestiti

Gli oggetti gestiti del file system e LDAP possono essere utilizzati per connettersi a IBM MQ e WebSphere Application Server, mentre la denominazione COS può essere utilizzata per connettersi solo a WebSphere Application Server.

Associazione di proprietà per gli oggetti gestiti

Per consentire alle applicazioni di utilizzare le definizioni oggetto di destinazione e factory di connessione IBM MQ JMS e WebSphere Application Server , le propriet ... richiamate da tali definizioni devono essere associate alle propriet ... XMS corrispondenti che possono essere impostate sulle destinazioni e sulle produzioni connessioni XMS .

Formato URI per contesti iniziali XMS

L'ubicazione del repository di oggetti gestiti viene fornita come URI (uniform resource indicator). Il formato dell'URI dipende dal tipo di contesto.

Servizio Web Ricerca JNDI

Per accedere a una directory di denominazione COS da XMS, un JNDI servizio Web di ricerca deve essere distribuito su un server WebSphere Application Server service integration bus . Questo servizio web converte le informazioni Java dal servizio di denominazione COS in un modulo che le applicazioni XMS possono leggere.

Richiamo di oggetti gestiti

XMS richiama un oggetto gestito dal repository utilizzando l'indirizzo fornito quando viene creato l'oggetto InitialContext o nelle proprietà InitialContext .

Attività correlate

Creazione di oggetti gestiti

Le definizioni dell'oggetto ConnectionFactory e Destination richieste dalle applicazioni XMS per stabilire una connessione a un server di messaggistica devono essere create utilizzando gli strumenti di gestione appropriati.

Oggetti InitialContext

Un'applicazione deve creare un contesto iniziale da utilizzare per stabilire una connessione al repository di oggetti gestiti per richiamare gli oggetti gestiti richiesti.

Riferimenti correlati

Proprietà richieste per gli oggetti ConnectionFactory gestiti

Quando un'applicazione crea una factory di connessione, è necessario definire un numero di proprietà per creare una connessione a un server di messaggistica.

Proprietà richieste per gli oggetti Destinazione gestiti

Un'applicazione che sta creando una destinazione deve impostare diverse proprietà dell'applicazione su un oggetto Destinazione gestito.

InitialContext (per interfaccia .NET)

Un'applicazione utilizza un oggetto InitialContext per creare oggetti dalle definizioni di oggetto richiamati da un repository di oggetti gestiti.

Proprietà di InitialContext

Una panoramica delle proprietà dell'oggetto InitialContext , con link a informazioni di riferimento più dettagliate.

Formato URI per contesti iniziali XMS

L'ubicazione del repository di oggetti gestiti viene fornita come URI (uniform resource indicator). Il formato dell'URI dipende dal tipo di contesto.

Contesto FileSystem

Per il contesto FileSystem , l'URL fornisce l'ubicazione della directory basata sul file system. La struttura dell'URL è quella definita da RFC 1738, *URL (Uniform Resource Locators)*: l'URL ha il prefisso `file://` e la sintassi che segue questo prefisso è una definizione valida di un file che può essere aperto sul sistema su cui è in esecuzione XMS .

Questa sintassi può essere specifica della piattaforma e può utilizzare i separatori `' / o' \'`. Se si utilizza `'\'`, è necessario eseguire l'escape di ciascun separatore utilizzando un ulteriore `'\'`. Ciò impedisce al framework .NET di tentare di interpretare il separatore come un carattere escape per quanto segue.

Questi esempi illustrano questa sintassi:

```
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
```

Contesto LDAP

Per il contesto LDAP, la struttura di base dell'URL è quella definita da RFC 2255, *Il formato dell'URL LDAP*, con il prefisso non sensibile al maiuscolo / minuscolo `ldap://`

La sintassi precisa viene illustrata nel seguente esempio:

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

Questa sintassi è quella definita in RFC ma non supporta attributi, ambiti, filtri o estensioni.

Esempi di questa sintassi sono:

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

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

Contesto WSS

Per il contesto WSS, l'URL è sotto forma di endpoint dei servizi Web, con prefisso `http://`.

In alternativa, è possibile utilizzare il prefisso `cosnaming://` o `wsvc://`,

Questi due prefissi vengono interpretati nel senso che si sta utilizzando un contesto WSS con l'URL a cui si accede tramite `http`, che consente al tipo di contesto iniziale di derivare facilmente direttamente dall'URL.

Esempi di questa sintassi sono:

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

Concetti correlati

Tipi supportati di repository di oggetti gestiti

Gli oggetti gestiti del file system e LDAP possono essere utilizzati per connettersi a IBM MQ e WebSphere Application Server, mentre la denominazione COS può essere utilizzata per connettersi solo a WebSphere Application Server.

Associazione di proprietà per gli oggetti gestiti

Per consentire alle applicazioni di utilizzare le definizioni oggetto di destinazione e factory di connessione IBM MQ JMS e WebSphere Application Server, le propriet ... richiamate da tali definizioni devono essere associate alle propriet ... XMS corrispondenti che possono essere impostate sulle destinazioni e sulle produzioni connessioni XMS.

Proprietà InitialContext

I parametri del costruttore InitialContext includono l'ubicazione del repository di oggetti gestiti, forniti come URI (uniform resource indicator). Affinché un'applicazione stabilisca una connessione al repository, potrebbe essere necessario fornire più informazioni rispetto a quelle contenute nell'URI.

Servizio Web Ricerca JNDI

Per accedere a una directory di denominazione COS da XMS, un JNDI servizio Web di ricerca deve essere distribuito su un server WebSphere Application Server service integration bus. Questo servizio web converte le informazioni Java dal servizio di denominazione COS in un modulo che le applicazioni XMS possono leggere.

Richiamo di oggetti gestiti

XMS richiama un oggetto gestito dal repository utilizzando l'indirizzo fornito quando viene creato l'oggetto InitialContext o nelle proprietà InitialContext.

Attività correlate

Creazione di oggetti gestiti

Le definizioni dell'oggetto ConnectionFactory e Destination richieste dalle applicazioni XMS per stabilire una connessione a un server di messaggistica devono essere create utilizzando gli strumenti di gestione appropriati.

Oggetti InitialContext

Un'applicazione deve creare un contesto iniziale da utilizzare per stabilire una connessione al repository di oggetti gestiti per richiamare gli oggetti gestiti richiesti.

Riferimenti correlati

Proprietà richieste per gli oggetti ConnectionFactory gestiti

Quando un'applicazione crea una factory di connessione, è necessario definire un numero di proprietà per creare una connessione a un server di messaggistica.

Proprietà richieste per gli oggetti Destinazione gestiti

Un'applicazione che sta creando una destinazione deve impostare diverse proprietà dell'applicazione su un oggetto Destinazione gestito.

InitialContext (per interfaccia .NET)

Un'applicazione utilizza un oggetto InitialContext per creare oggetti dalle definizioni di oggetto richiamati da un repository di oggetti gestiti.

Proprietà di InitialContext

Una panoramica delle proprietà dell'oggetto InitialContext , con link a informazioni di riferimento più dettagliate.

Servizio Web Ricerca JNDI

Per accedere a una directory di denominazione COS da XMS, un JNDI servizio Web di ricerca deve essere distribuito su un server WebSphere Application Server service integration bus . Questo servizio web converte le informazioni Java dal servizio di denominazione COS in un modulo che le applicazioni XMS possono leggere.

Il servizio Web viene fornito nel file EAR (Enterprise Archive) SIBXJndiLookupEAR.ear, presente nella directory di installazione. Per la release corrente di IBM Message Service Client for .NET, è possibile trovare SIBXJndiLookupEAR.ear nella directory *install_dir\java\lib* . Questo può essere installato all'interno di un server WebSphere Application Server service integration bus utilizzando la console di gestione o lo strumento di script wsaadmin. Per ulteriori informazioni sulla distribuzione delle applicazioni del servizio Web, fare riferimento alla documentazione del prodotto.

Per definire il servizio web all'interno delle applicazioni XMS , è sufficiente impostare la proprietà XMSC_IC_URL dell'oggetto InitialContext sull'URL dell'endpoint del servizio web. Ad esempio, se il servizio Web viene distribuito su un host del server denominato MyServer, un esempio di URL dell'endpoint del servizio Web:

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

L'impostazione della proprietà XMSC_IC_URL consente alle chiamate di ricerca InitialContext di richiamare il servizio Web sull'endpoint definito, che a sua volta ricerca l'oggetto gestito richiesto dal servizio di denominazione COS.

Le applicazioni .NET possono utilizzare il servizio Web. La distribuzione lato server è la stessa per XMS C, /C++ e XMS .NET. XMS .NET richiama il servizio Web direttamente tramite il framework Microsoft .NET .

Concetti correlati

Tipi supportati di repository di oggetti gestiti

Gli oggetti gestiti del file system e LDAP possono essere utilizzati per connettersi a IBM MQ e WebSphere Application Server, mentre la denominazione COS può essere utilizzata per connettersi solo a WebSphere Application Server.

Associazione di proprietà per gli oggetti gestiti

Per consentire alle applicazioni di utilizzare le definizioni oggetto di destinazione e factory di connessione IBM MQ JMS e WebSphere Application Server , le propriet ... richiamate da tali definizioni devono essere associate alle propriet ... XMS corrispondenti che possono essere impostate sulle destinazioni e sulle produzioni connessioni XMS .

Proprietà InitialContext

I parametri del costruttore InitialContext includono l'ubicazione del repository di oggetti gestiti, forniti come URI (uniform resource indicator). Affinché un'applicazione stabilisca una connessione al repository, potrebbe essere necessario fornire più informazioni rispetto a quelle contenute nell'URI.

Formato URI per contesti iniziali XMS

L'ubicazione del repository di oggetti gestiti viene fornita come URI (uniform resource indicator). Il formato dell'URI dipende dal tipo di contesto.

Richiamo di oggetti gestiti

XMS richiama un oggetto gestito dal repository utilizzando l'indirizzo fornito quando viene creato l'oggetto InitialContext o nelle proprietà InitialContext .

Impostazione dell'ambiente del server di messaggistica

Gli argomenti in questa sezione descrivono come impostare l'ambiente del server di messaggistica per consentire alle applicazioni di XMS di connettersi a un server.

Utilizzo di applicazioni di esempio XMS

Utilizzare le applicazioni di esempio fornite con XMS per verificare l'installazione e la configurazione del server di messaggistica e per creare le proprie applicazioni. Gli esempi forniscono una panoramica delle funzioni comuni di ciascuna API.

Attività correlate

Creazione di oggetti gestiti

Le definizioni dell'oggetto `ConnectionFactory` e `Destination` richieste dalle applicazioni XMS per stabilire una connessione a un server di messaggistica devono essere create utilizzando gli strumenti di gestione appropriati.

Oggetti `InitialContext`

Un'applicazione deve creare un contesto iniziale da utilizzare per stabilire una connessione al repository di oggetti gestiti per richiamare gli oggetti gestiti richiesti.

Installazione di `Message Service Client for .NET` mediante la procedura guidata di installazione

L'installazione utilizza un programma di installazione `InstallShield X/Windows MSI`. Sono disponibili due opzioni di configurazione, in modo che sia possibile scegliere un'installazione completa o personalizzata.

Riferimenti correlati

Proprietà richieste per gli oggetti `ConnectionFactory` gestiti

Quando un'applicazione crea una factory di connessione, è necessario definire un numero di proprietà per creare una connessione a un server di messaggistica.

Proprietà richieste per gli oggetti `Destinazione` gestiti

Un'applicazione che sta creando una destinazione deve impostare diverse proprietà dell'applicazione su un oggetto `Destinazione` gestito.

Richiamo di oggetti gestiti

XMS richiama un oggetto gestito dal repository utilizzando l'indirizzo fornito quando viene creato l'oggetto `InitialContext` o nelle proprietà `InitialContext`.

Gli oggetti da richiamare possono avere i seguenti tipi di nomi:

- Un nome semplice che descrive l'oggetto `Destinazione`, ad esempio, una destinazione coda denominata `SalesOrders`
- Un nome composto, che può essere costituito da `SubContexts`, separato da '/', e deve terminare con il nome oggetto. Un esempio di nome composto è `"Warehouse/PickLists/DispatchQueue2"` dove `Warehouse` e `Picklists` sono `SubContexts` nella directory di denominazione e `DispatchQueue2` è il nome di un oggetto di destinazione.

Concetti correlati

Tipi supportati di repository di oggetti gestiti

Gli oggetti gestiti del file system e LDAP possono essere utilizzati per connettersi a IBM MQ e WebSphere Application Server, mentre la denominazione COS può essere utilizzata per connettersi solo a WebSphere Application Server.

Associazione di proprietà per gli oggetti gestiti

Per consentire alle applicazioni di utilizzare le definizioni oggetto di destinazione e factory di connessione IBM MQ JMS e WebSphere Application Server, le propriet ... richiamate da tali definizioni devono essere associate alle propriet ... XMS corrispondenti che possono essere impostate sulle destinazioni e sulle produzioni connessioni XMS.

Proprietà `InitialContext`

I parametri del costruttore `InitialContext` includono l'ubicazione del repository di oggetti gestiti, forniti come URI (uniform resource indicator). Affinché un'applicazione stabilisca una connessione al repository, potrebbe essere necessario fornire più informazioni rispetto a quelle contenute nell'URI.

Formato URI per contesti iniziali XMS

L'ubicazione del repository di oggetti gestiti viene fornita come URI (uniform resource indicator). Il formato dell'URI dipende dal tipo di contesto.

Servizio Web Ricerca JNDI

Per accedere a una directory di denominazione COS da XMS, un JNDI servizio Web di ricerca deve essere distribuito su un server WebSphere Application Server service integration bus . Questo servizio web converte le informazioni Java dal servizio di denominazione COS in un modulo che le applicazioni XMS possono leggere.

Attività correlate

Creazione di oggetti gestiti

Le definizioni dell'oggetto ConnectionFactory e Destination richieste dalle applicazioni XMS per stabilire una connessione a un server di messaggistica devono essere create utilizzando gli strumenti di gestione appropriati.

Oggetti InitialContext

Un'applicazione deve creare un contesto iniziale da utilizzare per stabilire una connessione al repository di oggetti gestiti per richiamare gli oggetti gestiti richiesti.

Riferimenti correlati

Proprietà richieste per gli oggetti ConnectionFactory gestiti

Quando un'applicazione crea una factory di connessione, è necessario definire un numero di proprietà per creare una connessione a un server di messaggistica.

Proprietà richieste per gli oggetti Destinazione gestiti

Un'applicazione che sta creando una destinazione deve impostare diverse proprietà dell'applicazione su un oggetto Destinazione gestito.

InitialContext (per interfaccia .NET)

Un'applicazione utilizza un oggetto InitialContext per creare oggetti dalle definizioni di oggetto richiamati da un repository di oggetti gestiti.

Proprietà di InitialContext

Una panoramica delle proprietà dell'oggetto InitialContext , con link a informazioni di riferimento più dettagliate.

Protezione delle comunicazioni per applicazioni XMS

Questa sezione fornisce informazioni sull'impostazione delle comunicazioni protette per consentire alle applicazioni XMS di connettersi tramite SSL (Secure Sockets Layer) a un motore di messaggistica WebSphere Application Server service integration bus o a un gestore code IBM MQ .

La sezione contiene i seguenti argomenti:

- [“Connessioni protette a un gestore code IBM MQ” a pagina 66](#)
- [“Associazioni di nomi CipherSuite e CipherSpec per connessioni a un gestore code IBM MQ” a pagina 67](#)
- [“Connessioni sicure a un motore di messaggistica WebSphere Application Server service integration bus” a pagina 68](#)
- [“Associazioni di nomi CipherSuite e CipherSpec per le connessioni a WebSphere Application Server service integration bus” a pagina 69](#)

Connessioni protette a un gestore code IBM MQ

Per consentire a un'applicazione di XMS .NET di stabilire connessioni sicure a un gestore code IBM MQ , le proprietà pertinenti devono essere definite nell'oggetto ConnectionFactory .

Il protocollo utilizzato nella negoziazione di codifica può essere SSL (Secure Sockets Layer) o TLS (Transport Layer Security), a seconda della CipherSuite specificata nell'oggetto ConnectionFactory .

Se si utilizzano le librerie client IBM WebSphere MQ 7.0.0 Fix Pack 1 e successive e ci si connette a un gestore code IBM WebSphere MQ 7.0 , è possibile creare più connessioni allo stesso gestore code nell'applicazione XMS . Tuttavia, la connessione a un gestore code differente non è consentita. Se si tenta, si riceve l'errore MQRC_SSL_ALREADY_INITIALIZED .

Se si utilizzano le librerie client IBM WebSphere MQ 6.0 e successive, è possibile creare una connessione SSL solo se si chiude prima qualsiasi connessione SSL precedente. Non sono consentite più connessioni

SSL simultanee dallo stesso processo allo stesso gestore code o a gestori code differenti. Se si tenta più di una richiesta, si riceve l'avvertenza MQRC_SSL_ALREADY_INITIALIZED, che potrebbe indicare che alcuni parametri richiesti per la connessione SSL sono stati ignorati.

Le proprietà ConnectionFactory per le connessioni tramite SSL a un gestore code IBM MQ , con una breve descrizione, sono riportate nella seguente tabella:

<i>Tabella 14. Proprietà di ConnectionFactory per connessioni a un gestore code IBM MQ tramite SSL</i>	
Nome della proprietà	Descrizione
<u>XMSC_WMQ_SSL_CERT_STORES</u>	L'ubicazione dei server che contengono i CRL (cettificate revocation list) da utilizzare su una connessione SSL a un gestore code.
<u>XMSC_WMQ_SSL_CIPHER_SPEC</u>	Il nome della CipherSpec da utilizzare su una connessione protetta a un gestore code.
<u>XMSC_WMQ_SSL_CIPHER_SUITE</u>	Il nome della CipherSuite da utilizzare su una connessione TLS a un gestore code. Il protocollo utilizzato nella negoziazione della connessione protetta dipende dalla CipherSuite specificata.
<u>XMSC_WMQ_SSL_CRYPT_HW</u>	I dettagli di configurazione per l'hardware di crittografia connesso al sistema client.
<u>XMSC_WMQ_SSL_FIPS_REQUIRED</u>	Il valore della proprietà determina se un'applicazione può o meno utilizzare delle suite di crittografia non conformi a FIPS. Se questa proprietà è impostata su true, per la connessione client-server vengono utilizzati solo gli algoritmi FIPS.
<u>XMSC_WMQ_SSL_KEY_REPOSITORY</u>	L'ubicazione del file del database delle chiavi in cui sono memorizzati chiavi e certificati.
<u>XMSC_WMQ_SSL_KEY_RESETCOUNT</u>	Il KeyResetCount rappresenta il numero totale di byte non crittografati inviati e ricevuti in una conversazione SSL prima che venga rinegoziata la chiave segreta.
<u>XMSC_WMQ_SSL_PEER_NAME</u>	Il nome peer da utilizzare su una connessione SSL a un gestore code.

Riferimenti correlati

[IConnectionFactory](#) (per l'interfaccia .NET)

Un'applicazione utilizza una produzione connessioni per creare una connessione.

[Proprietà di ConnectionFactory](#)

Una panoramica delle proprietà dell'oggetto ConnectionFactory , con link a informazioni di riferimento più dettagliate.

[Proprietà richieste per gli oggetti ConnectionFactory gestiti](#)

Quando un'applicazione crea una factory di connessione, è necessario definire un numero di proprietà per creare una connessione a un server di messaggistica.

Associazioni di nomi CipherSuite e CipherSpec per connessioni a un gestore code IBM MQ

InitialContext converte tra la proprietà del factory di connessione JMSAdmin SSLCIPHERSUITE e il XMS quasi equivalente XMSC_WMQ_SSL_CIPHER_SPEC. Una traduzione simile è necessaria se si specifica un valore per XMSC_WMQ_SSL_CIPHER_SUITE ma si omette il valore per XMSC_WMQ_SSL_CIPHER_SPEC.

[Tabella 15 a pagina 68](#) elenca i CipherSpecs disponibili e i relativi equivalenti JSSE CipherSuite .

Tabella 15. CipherSpecs disponibili e i relativi equivalenti JSSE CipherSuite

CipherSpec	CipherSuite JSSE equivalente
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

Nota:

- TLS_RSA_WITH_3DES_EDE_CBC_SHA è obsoleto. Tuttavia, può essere ancora utilizzato per trasferire fino a 32 GB di dati prima che la connessione venga terminata con l'errore AMQ9288. Per evitare questo errore, è necessario evitare di utilizzare il triplo DES o abilitare la reimpostazione della chiave segreta quando si utilizza questo CipherSpec.

Connessioni sicure a un motore di messaggistica WebSphere Application Server service integration bus

Per abilitare un'applicazione XMS .NET per stabilire connessioni sicure a un motore di messaggistica WebSphere Application Server service integration bus , le proprietà pertinenti devono essere definite nell'oggetto ConnectionFactory .

XMS fornisce il supporto SSL e HTTPS per le connessioni a WebSphere Integration Bus. SSL e HTTPS forniscono connessioni sicure per l'autenticazione e la riservatezza.

Come la sicurezza WebSphere , la sicurezza XMS è configurata in base agli standard di sicurezza JSSE e alle convenzioni di denominazione, che includono l'utilizzo di CipherSuites per specificare gli algoritmi utilizzati quando si negozia una connessione sicura. Il protocollo utilizzato nella negoziazione della codifica può essere SSL o TLS, a seconda della CipherSuite specificata nell'oggetto ConnectionFactory .

Tabella 16 a pagina 68 elenca le proprietà che devono essere definite nell'oggetto ConnectionFactory .

Tabella 16. Proprietà di ConnectionFactory per le connessioni sicure a un motore di messaggistica WebSphere Application Server service integration bus

Nome della proprietà	Descrizione
<u>XMSC_WPM_SSL_CIPHER_SUITE</u>	Il nome della CipherSuite da utilizzare su una connessione TLS a un motore di messaggistica WebSphere Integration Bus . Il protocollo utilizzato nella negoziazione della connessione protetta dipende dalla CipherSuite specificata.
<u>XMSC_WPM_SSL_KEYRING_LABEL</u>	Il certificato da utilizzare quando si esegue l'autenticazione presso il server.

Di seguito viene riportato un esempio di proprietà ConnectionFactory per le connessioni sicure a un motore di messaggistica WebSphere Application Server service integration bus :

```
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);
```

Dove chain_name deve essere impostato su BootstrapTunneledSecureMessaging o BootstrapSecureMessaging e port_number è il numero della porta su cui il server bootstrap è in ascolto per le richieste in entrata.

Di seguito è riportato un esempio di proprietà ConnectionFactory per le connessioni sicure a un motore di messaggistica WebSphere Application Server service integration bus con valori di esempio inseriti:

```
/* 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");
```

Riferimenti correlati

[IConnectionFactory](#) (per l'interfaccia .NET)

Un'applicazione utilizza una produzione connessioni per creare una connessione.

[Proprietà di ConnectionFactory](#)

Una panoramica delle proprietà dell'oggetto ConnectionFactory, con link a informazioni di riferimento più dettagliate.

[Proprietà richieste per gli oggetti ConnectionFactory gestiti](#)

Quando un'applicazione crea una factory di connessione, è necessario definire un numero di proprietà per creare una connessione a un server di messaggistica.

Associazioni di nomi CipherSuite e CipherSpec per le connessioni a WebSphere Application Server service integration bus

Poiché GSKit utilizza CipherSpecs invece di CipherSuites, i nomi CipherSuite in stile JSSE specificati nella proprietà XMSC_WPM_SSL_CIPHER_SUITE devono essere associati ai nomi CipherSpec in stile GSKit.

[Tabella 17 a pagina 69](#) elenca la CipherSpec equivalente per ogni CipherSuite riconosciuto.

CipherSuite	CipherSpec equivalente
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
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 è obsoleto. Tuttavia, può essere ancora utilizzato per trasferire fino a 32 GB di dati prima che la connessione venga terminata con l'errore AMQ9288. Per evitare questo errore, è necessario evitare di utilizzare il triplo DES o abilitare la reimpostazione della chiave segreta quando si utilizza questo CipherSpec.

XMS messaggi

Questa sezione descrive la struttura e il contenuto dei messaggi XMS e spiega come le applicazioni elaborano i messaggi XMS.

Questa sezione contiene i seguenti argomenti:

- [“Parti di un messaggio XMS” a pagina 70](#)
- [“Campi di intestazione in un messaggio XMS” a pagina 70](#)
- [“Proprietà di un messaggio XMS” a pagina 71](#)
- [“Il contenuto di un messaggio XMS” a pagina 75](#)
- [“Selettori di messaggi” a pagina 80](#)
- [“Mappatura dei messaggi XMS sui messaggi IBM MQ” a pagina 82](#)

Riferimenti correlati

[IMessage \(per l'interfaccia .NET\)](#)

Un oggetto Message rappresenta un messaggio che un'applicazione invia o riceve. IMessage è una superclasse per le classi di messaggi come IMapMessage.

Parti di un messaggio XMS

Un messaggio XMS è composto da un'intestazione, una serie di proprietà e un corpo.

Intestazione

L'intestazione di un messaggio contiene campi e tutti i messaggi contengono la stessa serie di campi intestazione. XMS e le applicazioni utilizzano i valori dei campi di intestazione per identificare e instradare i messaggi. Per ulteriori informazioni sui campi di intestazione, consultare [“Campi di intestazione in un messaggio XMS”](#) a pagina 70.

Insieme di proprietà

Le proprietà di un messaggio specificano ulteriori informazioni sul messaggio. Sebbene tutti i messaggi abbiano la stessa serie di campi di intestazione, ogni messaggio può avere una serie diversa di proprietà. Per ulteriori informazioni, fare riferimento a [“Proprietà di un messaggio XMS”](#) a pagina 71.

Corpo

Il corpo di un messaggio contiene dati dell'applicazione. Per ulteriori informazioni, fare riferimento a [“Il contenuto di un messaggio XMS”](#) a pagina 75.

Un'applicazione può selezionare i messaggi che desidera ricevere. Utilizzando i selettori di messaggi, che specifica i criteri di selezione. I criteri possono essere basati sui valori di determinati campi di intestazione e sui valori di qualsiasi proprietà di un messaggio. Per ulteriori informazioni sui selettori di messaggi, consultare [“Selettori di messaggi”](#) a pagina 80.

Riferimenti correlati

[Campi di intestazione in un messaggio XMS](#)

Per consentire a una applicazione XMS di scambiare messaggi con un'applicazione WebSphere JMS , l'intestazione di un messaggio XMS contiene i campi di intestazione del messaggio JMS .

[Proprietà di un messaggio XMS](#)

XMS supporta tre tipi di proprietà del messaggio: JMS proprietà definite IBM e proprietà definite dall'applicazione.

[Il contenuto di un messaggio XMS](#)

Il corpo di un messaggio contiene dati dell'applicazione. Tuttavia, un messaggio non può avere alcun corpo e comprende solo i campi di intestazione e le proprietà.

[Selettori di messaggi](#)

Un'applicazione XMS utilizza selettori di messaggi per selezionare i messaggi che desidera ricevere.

[Mappatura dei messaggi XMS sui messaggi IBM MQ](#)

I campi di intestazione JMS e le proprietà di un messaggio XMS sono associati ai campi nelle strutture di intestazione di un messaggio IBM MQ .

Campi di intestazione in un messaggio XMS

Per consentire a una applicazione XMS di scambiare messaggi con un'applicazione WebSphere JMS , l'intestazione di un messaggio XMS contiene i campi di intestazione del messaggio JMS .

I nomi di questi campi di intestazione iniziano con il prefisso JMS. Per una descrizione dei campi dell'intestazione del messaggio JMS, consultare *Java Message Service Specifica*.

XMS implementa i campi di intestazione del messaggio JMS come attributi di un oggetto Message. Ogni campo di intestazione ha i propri metodi per impostare e ottenere il proprio valore. Per una descrizione di questi metodi, consultare [“IMessaggio”](#) a pagina 124. Un campo di intestazione è sempre leggibile e scrivibile.

Tabella 18 a pagina 71 elenca i campi di intestazione del messaggio JMS e indica in che modo il valore di ciascun campo è impostato per un messaggio trasmesso. Alcuni dei campi vengono impostati automaticamente da XMS quando un'applicazione invia un messaggio o, nel caso di JMSRenei, quando un'applicazione riceve un messaggio.

<i>Tabella 18. Campi di intestazione del messaggio JMS</i>	
Nome del campo di intestazione del messaggio JMS	Modalità di impostazione del valore per un messaggio trasmesso (nel formato <i>method [class]</i>)
JMSCorrelationID	Impostare JMSCorrelationID [Messaggio]
JMSDeliveryMode	Invia [MessageProducer]
JMSDestination	Invia [MessageProducer]
JMSExpiration	Invia [MessageProducer]
JMSMessageID	Invia [MessageProducer]
JMSPriority	Invia [MessageProducer]
JMSRedelivered	Ricevi [MessageConsumer]
JMSReplyTo	Impostare JMSReplyTo [Messaggio]
JMSTimestamp	Invia [MessageProducer]
JMSType	Imposta JMSType [Messaggio]

Riferimenti correlati

Parti di un messaggio XMS

Un messaggio XMS è composto da un'intestazione, una serie di proprietà e un corpo.

Proprietà di un messaggio XMS

XMS supporta tre tipi di proprietà del messaggio: JMS proprietà definite IBM e proprietà definite dall'applicazione.

Il contenuto di un messaggio XMS

Il corpo di un messaggio contiene dati dell'applicazione. Tuttavia, un messaggio non può avere alcun corpo e comprende solo i campi di intestazione e le proprietà.

Selettori di messaggi

Un'applicazione XMS utilizza selettori di messaggi per selezionare i messaggi che desidera ricevere.

Mappatura dei messaggi XMS sui messaggi IBM MQ

I campi di intestazione JMS e le proprietà di un messaggio XMS sono associati ai campi nelle strutture di intestazione di un messaggio IBM MQ .

Proprietà di un messaggio XMS

XMS supporta tre tipi di proprietà del messaggio: JMS proprietà definite IBM e proprietà definite dall'applicazione.

Un'applicazione XMS può scambiare messaggi con un'applicazione WebSphere JMS perché XMS supporta le seguenti proprietà predefinite di un oggetto Messaggio:

- Le stesse proprietà definite da JMSsupportate da WebSphere JMS . I nomi di queste proprietà iniziano con il prefisso JMSX.
- Le stesse proprietà definite da IBMSupportate da WebSphere JMS . I nomi di queste proprietà iniziano con il prefisso JMS_IBM_.

Ogni proprietà predefinita ha due nomi:

- Un nome JMS , per una proprietà definita da JMSo un nome WebSphere JMS , per una proprietà definita da IBM.

Questo è il nome con cui la proprietà è nota in JMS o WebSphere JMSed è anche il nome che viene trasmesso con un messaggio che ha questa proprietà. Un'applicazione XMS utilizza questo nome per identificare la proprietà in un'espressione del selettore messaggi.

- Un nome XMS per identificare la proprietà in tutte le situazioni tranne che in un'espressione del selettore messaggi. Ogni nome XMS è definito come costante con nome nella classe `IBM.XMS.XMSC`. Il valore della costante denominata è il nome JMS o WebSphere JMS corrispondente.

Oltre alle proprietà predefinite, un'applicazione XMS può creare e utilizzare la propria serie di proprietà del messaggio. Queste proprietà sono denominate *proprietà definite dall'applicazione*.

Dopo che un'applicazione crea un messaggio, le proprietà del messaggio sono leggibili e scrivibili. Le proprietà rimangono leggibili e scrivibili dopo che l'applicazione invia il messaggio. Quando un'applicazione riceve un messaggio, le proprietà del messaggio sono di sola lettura. Se un'applicazione richiama il metodo `Clear Properties` della classe `Message` quando le proprietà di un messaggio sono di sola lettura, le proprietà diventano leggibili e scrivibili. Il metodo cancella anche le proprietà.

Il messaggio ricevuto, quando inoltrato dopo aver cancellato le proprietà del messaggio, si comporterà in modo coerente con il funzionamento dell'inoltro di un `BytesMessage` standard WMQ XMS per .NET con le proprietà del messaggio ripulite.

Ciò, tuttavia, non è consigliato poiché le seguenti proprietà andranno perse:

- Valore della proprietà `JMS_IBM_Encoding` che indica che i dati del messaggio non possono essere decodificati in modo significativo.
- Valore della proprietà `JMS_IBM_Format`, che indica che il concatenamento dell'intestazione tra l'intestazione del messaggio (MQMD o la nuova MQRFH2) e le intestazioni esistenti verrebbe interrotto.

Per determinare i valori di tutte le proprietà di un messaggio, un'applicazione può chiamare il metodo `Get Properties` della classe `Message`. Il metodo crea un iteratore che incapsula un elenco di oggetti proprietà, dove ogni oggetto proprietà rappresenta una proprietà del messaggio. L'applicazione può quindi utilizzare i metodi della classe `Iterator` per richiamare ciascun oggetto Proprietà a turno e può utilizzare i metodi della classe `Proprietà` per recuperare il nome, il tipo di dati e il valore di ogni proprietà.

Riferimenti correlati

Parti di un messaggio XMS

Un messaggio XMS è composto da un'intestazione, una serie di proprietà e un corpo.

Campi di intestazione in un messaggio XMS

Per consentire a una applicazione XMS di scambiare messaggi con un'applicazione WebSphere JMS, l'intestazione di un messaggio XMS contiene i campi di intestazione del messaggio JMS.

Il contenuto di un messaggio XMS

Il corpo di un messaggio contiene dati dell'applicazione. Tuttavia, un messaggio non può avere alcun corpo e comprende solo i campi di intestazione e le proprietà.

Selettori di messaggi

Un'applicazione XMS utilizza selettori di messaggi per selezionare i messaggi che desidera ricevere.

Mappatura dei messaggi XMS sui messaggi IBM MQ

I campi di intestazione JMS e le proprietà di un messaggio XMS sono associati ai campi nelle strutture di intestazione di un messaggio IBM MQ.

Proprietà definite da JMS di un messaggio

Diverse proprietà definite da JMS di un messaggio sono supportate sia da XMS che WebSphere JMS.

Tabella 19 a pagina 73 elenca le proprietà definite JMS di un messaggio supportate sia da XMS che WebSphere JMS. Per una descrizione delle proprietà definite da JMS, vedere *Java Message Service Specifica*. Le proprietà definite da JMS non sono valide per una connessione in tempo reale a un broker.

La tabella specifica il tipo di dati di ciascuna proprietà e indica in che modo il valore della proprietà è impostato per un messaggio trasmesso. Alcune delle proprietà vengono impostate automaticamente da XMS quando un'applicazione invia un messaggio o, nel caso di `JMSXDeliveryCount`, quando un'applicazione riceve un messaggio.

Tabella 19. Proprietà definite da JMSdi un messaggio

Nome XMS della proprietà definita da JMS	Nome JMS	Tipo dati	Modalità di impostazione del valore per un messaggio trasmesso (nel formato <i>method [class]</i>)
IDAPP_JMSX	JMSXAppID	System.String	Invia [MessageProducer]
CONTEGGIO_DISTRIBUZIONE_JMSX	JMSXDeliveryCount	System.Int32	Ricevi [MessageConsumer]
IDGRUPPO_JMS	JMSXGroupID	System.String	Imposta proprietà stringa [PropertyContext]
GROUPSEQ JMS	JMSXGroupSeq	System.Int32	Imposta proprietà numero intero [PropertyContext]
IDUSER_JMS	JMSXUserID	System.String	Invia [MessageProducer]

Proprietà definite da IBMdi un messaggio

Diverse proprietà IBMdefinite di un messaggio sono supportate da XMS e WebSphere JMS.

Tabella 20 a pagina 73 elenca le IBM proprietà definite di un messaggio supportate sia da XMS che WebSphere JMS. Per ulteriori informazioni sulle proprietà definite da IBM, consultare IBM MQ o la documentazione del prodotto WebSphere Application Server .

La tabella specifica il tipo di dati di ciascuna proprietà e indica in che modo il valore della proprietà è impostato per un messaggio trasmesso. Alcune proprietà vengono impostate automaticamente da XMS quando un'applicazione invia un messaggio.

Tabella 20. IBMdi un messaggio

Nome XMS della proprietà definita da IBM	Nome WebSphere JMS	Tipo dati	Modalità di impostazione del valore per un messaggio trasmesso (nel formato <i>method [class]</i>)
JMS_IBM_CHARACTER_SET	Serie di caratteristiche IBM JMS	System.Int32	Imposta proprietà numero intero [PropertyContext]
JMS_IBM_Encoding	JMS_IBM_Encoding	System.Int32	Imposta proprietà numero intero [PropertyContext]
MESSAGGIO_IBM_JMS	JMS_IBM_ExceptionMessage	System.String	Ricevi [MessageConsumer]
JMS_IBM_EXCEPTIONREASON	JMS_IBM_ExceptionReason	System.Int32	Ricevi [MessageConsumer]
JMS_IBM_EXCEPTIONTIMESTAMP	JMS_IBM_ExceptionTimestamp	System.Int64	Ricevi [MessageConsumer]
JMS_IBM_EXCEPTIONPROBLEM_DESTINAZIONE	Destinazione JMS_IBM_ExceptionProblem	System.String	Ricevi [MessageConsumer]
JMS_IBM_FEEDBACK	Feedback IBM JMS	System.Int32	Imposta proprietà numero intero [PropertyContext]

Tabella 20. IBMdi un messaggio (Continua)

Nome XMS della proprietà definita da IBM	Nome WebSphere JMS	Tipo dati	Modalità di impostazione del valore per un messaggio trasmesso (nel formato <i>method [class]</i>)
JMS_IBM_FORMAT	Formato_IBM_JMS	System.String	Imposta proprietà stringa [PropertyContext]
JMS_IBM_LAST_MSG_IN_GROUP	JMS_IBM_Last_Msg_In_Group	System.Boolean	Imposta proprietà numero intero [PropertyContext]
TIPO_IBM_JMS	JMS_IBM_MsgType	System.Int32	Imposta proprietà numero intero [PropertyContext]
TIPO_PUTAPPL_IBM_JMS	JMS_IBM_PutApplTipo	System.Int32	Invia [MessageProducer]
DATA_IBM_JMS	JMS_IBM_PutDate	System.String	Invia [MessageProducer]
PUTORA_IBM_JMS	JMS_IBM_PutTime	System.String	Invia [MessageProducer]
COA IBM_REPORT_JMS	COA report IBM JMS	System.Int32	Imposta proprietà numero intero [PropertyContext]
COD IBM REPORT_JMS	COD report IBM JMS	System.Int32	Imposta proprietà numero intero [PropertyContext]
JMS_IBM_REPORT_DISCARD_MSG	JMS_IBM_Report_Discard_Msg	System.Int32	Imposta proprietà numero intero [PropertyContext]
JMS_IBM_REPORT_EXCEPTION	Eccezione JMS_IBM_Report_Exception	System.Int32	Imposta proprietà numero intero [PropertyContext]
SCADENZA_REPORT_IBM_JMS	Scadenza report IBM JMS	System.Int32	Imposta proprietà numero intero [PropertyContext]
NAN_IBM_REPORT_JMS	NAN report IBM JMS	System.Int32	Imposta proprietà numero intero [PropertyContext]
PAN IBM REPORT_JMS	PAN report IBM JMS	System.Int32	Imposta proprietà numero intero [PropertyContext]
JMS_IBM_REPORT_PASSWORD_CORRELATION_ID	ID correlazione password report IBM JMS	System.Int32	Imposta proprietà numero intero [PropertyContext]
ID_IBM_REPORT_PASSWORD_MESSAGE_JMS	ID_ms_ms_IBM_Report_Password_Msg_ID	System.Int32	Imposta proprietà numero intero [PropertyContext]
IDMESSAGGIO_IBM_SYSTEM_MESSAGE_ID	JMS_IBM_System_MessageID	System.String	Invia [MessageProducer]

Proprietà definite dall'applicazione di un messaggio

Un'applicazione XMS può creare e utilizzare la propria serie di proprietà del messaggio. Quando un'applicazione invia un messaggio, anche queste proprietà vengono trasmesse con il messaggio. Un'applicazione ricevente, utilizzando i selettori di messaggi, può quindi selezionare quali messaggi desidera ricevere in base ai valori di queste proprietà.

Per consentire a un'applicazione WebSphere JMS di scegliere ed elaborare i messaggi inviati da un'applicazione XMS, è necessario che il nome di una proprietà definita dall'applicazione sia conforme alle regole per la formazione di identificatori nelle espressioni del selettore messaggi, come documentato

nella documentazione del prodotto IBM MQ . Il valore di una proprietà definita dall'applicazione deve avere uno dei seguenti tipi di dati: System.Boolean, System.SByte, System.Int16, System.Int32, System.Int64, System.Float, System.Double o System.String.

Il contenuto di un messaggio XMS

Il corpo di un messaggio contiene dati dell'applicazione. Tuttavia, un messaggio non può avere alcun corpo e comprende solo i campi di intestazione e le proprietà.

XMS supporta cinque tipi di messaggio:

Byte

Il corpo contiene un flusso di byte. Un messaggio con questo tipo di corpo viene denominato *messaggio byte*. L'interfaccia `IBytesMessage` contiene i metodi per elaborare il contenuto di un messaggio di byte. Per ulteriori informazioni, consulta [“Messaggi byte” a pagina 77](#).

Associa

Il corpo contiene una serie di coppie nome - valore, in cui ogni valore ha un tipo di dati associato. Un messaggio con questo tipo di corpo viene denominato *messaggio di mappa*. L'interfaccia `IMapMessage` contiene i metodi per elaborare il contenuto di un messaggio di associazione. Per ulteriori informazioni, consulta [“Associa messaggi” a pagina 77](#).

Oggetto

Il corpo contiene un oggetto Java o .NET serializzato. Un messaggio con questo tipo di corpo viene denominato *messaggio oggetto*. L'interfaccia `IObjectMessage` contiene i metodi per elaborare il testo di un messaggio oggetto. Per ulteriori informazioni, consulta [“Messaggi oggetto” a pagina 78](#).

Flusso

Il corpo contiene un flusso di valori, in cui ciascun valore ha un tipo di dati associato. Un messaggio con questo tipo di corpo viene denominato *messaggio di flusso*. L'interfaccia `IStreamMessage` contiene i metodi per elaborare il contenuto di un messaggio di flusso. Per ulteriori informazioni, consulta [“Il processo di flusso dei messaggi” a pagina 79](#).

Testo

Il body contiene una stringa. Un messaggio con questo tipo di corpo viene denominato *messaggio di testo*. L'interfaccia `ITextMessage` contiene i metodi per elaborare il contenuto di un messaggio di testo. Per ulteriori informazioni, consulta [“Messaggi di testo” a pagina 80](#).

L'interfaccia `IMessage` è l'elemento principale di tutti gli oggetti messaggio e può essere utilizzata nelle funzioni di messaggistica per rappresentare qualsiasi tipo di messaggio XMS .

Per informazioni sulla dimensione e sui valori massimo e minimo di ciascuno di tali tipi di dati, consultare [Tabella 5 a pagina 40](#).

Riferimenti correlati

[Parti di un messaggio XMS](#)

Un messaggio XMS è composto da un'intestazione, una serie di proprietà e un corpo.

[Campi di intestazione in un messaggio XMS](#)

Per consentire a una applicazione XMS di scambiare messaggi con un'applicazione WebSphere JMS , l'intestazione di un messaggio XMS contiene i campi di intestazione del messaggio JMS .

[Proprietà di un messaggio XMS](#)

XMS supporta tre tipi di proprietà del messaggio: JMS proprietà definite IBM e proprietà definite dall'applicazione.

[Selettori di messaggi](#)

Un'applicazione XMS utilizza selettori di messaggi per selezionare i messaggi che desidera ricevere.

[Mappatura dei messaggi XMS sui messaggi IBM MQ](#)

I campi di intestazione JMS e le proprietà di un messaggio XMS sono associati ai campi nelle strutture di intestazione di un messaggio IBM MQ .

Tipi di dati per elementi di dati dell'applicazione

Per essere certi che un'applicazione XMS possa scambiare messaggi con un'applicazione IBM MQ classes for JMS, entrambe le applicazioni devono essere in grado di interpretare i dati dell'applicazione nel corpo di un messaggio nello stesso modo.

Per questo motivo, ogni elemento dei dati dell'applicazione scritto nel corpo di un messaggio da parte di un'applicazione XMS deve avere uno dei tipi di dati elencati in [Tabella 21 a pagina 76](#). Per ciascun tipo di dati, la tabella mostra il tipo di dati Java compatibile. XMS fornisce i metodi per scrivere elementi di dati dell'applicazione solo con questi tipi di dati.

XMS Tipo dati	Rappresenta	Tipo di dati Java compatibile
System.Boolean	Il valore booleano true o false	booleano
System.Char16	Carattere a doppio byte	char
System.SByte	Numero intero a 8 bit con segno	byte
System.Int16	Numero intero a 16 bit con segno	breve
System.Int32	Numero intero a 32 bit con segno	int
System.Int64	Numero intero a 64 bit con segno	completo
System.Float	Numero a virgola mobile con segno	mobile
System.Double	Numero a virgola mobile a doppia precisione con segno	doppio
System.String	Stringa di caratteri	Stringa

Per informazioni sulla dimensione, il valore massimo e il valore minimo di ognuno di questi tipi di dati, consultare [“XMS Tipi primitivi” a pagina 39](#).

Concetti correlati

Attributi e proprietà degli oggetti

Un oggetto XMS può avere attributi e proprietà, che sono caratteristiche dell'oggetto, che sono implementati in modi diversi.

XMS Tipi primitivi

XMS fornisce equivalenti degli otto tipi primitivi Java (byte, short, int, long, float, double, char e boolean). Ciò consente lo scambio di messaggi tra XMS e JMS senza che i dati vadano persi o danneggiati.

Conversione implicita di un valore di proprietà da un tipo di dati a un altro

Quando un'applicazione acquisisce il valore di una proprietà, il valore può essere convertito da XMS in un altro tipo di dati. Molte regole regolano quali conversioni sono supportate e come XMS esegue le conversioni.

Riferimenti correlati

Messaggi byte

Il corpo di un messaggio di byte contiene un flusso di byte. Il corpo contiene solo i dati effettivi ed è responsabilità delle applicazioni di invio e di ricezione interpretare tali dati.

Associa messaggi

Il corpo di un messaggio di associazione contiene una serie di coppie nome - valore, in cui ogni valore ha un tipo di dati associato.

Messaggi oggetto

Il corpo di un messaggio di oggetto contiene un oggettoJava o .NET serializzato.

Il processo di flusso dei messaggi

Il corpo di un messaggio di flusso contiene un flusso di valori, in cui ogni valore ha un tipo di dati associato.

Messaggi di testo

Il corpo di un messaggio di testo contiene una stringa.

Messaggi byte

Il corpo di un messaggio di byte contiene un flusso di byte. Il corpo contiene solo i dati effettivi ed è responsabilità delle applicazioni di invio e di ricezione interpretare tali dati.

I messaggi di byte sono utili se un'applicazione XMS deve scambiare messaggi con applicazioni che non utilizzano l'interfaccia di programmazione dell'applicazione XMS o JMS .

Dopo che un'applicazione crea un messaggio di byte, il corpo del messaggio è di sola scrittura. L'applicazione assembla i dati dell'applicazione nel corpo richiamando i metodi di scrittura appropriati dell'interfaccia `IBytesMessage` per .NET. Ogni volta che l'applicazione scrive un valore nel flusso di messaggi di byte, il valore viene assemblato immediatamente dopo il precedente valore scritto dall'applicazione. XMS mantiene un cursore interno per ricordare la posizione dell'ultimo byte assemblato.

Quando l'applicazione invia il messaggio, il corpo del messaggio diventa di sola lettura. In questa modalità, l'applicazione può inviare il messaggio ripetutamente.

Quando un'applicazione riceve un messaggio di byte, il corpo del messaggio è di sola lettura. L'applicazione può utilizzare i metodi di lettura appropriati dell'interfaccia `IBytesMessage` per leggere il contenuto del flusso di messaggi di byte. L'applicazione legge i byte in sequenza e XMS mantiene un cursore interno per ricordare la posizione dell'ultimo byte letto.

Se un'applicazione richiama il metodo `Reset` dell'interfaccia `IBytesMessage` quando il corpo di un messaggio di byte è scrivibile, il corpo diventa di sola lettura. Il metodo riposiziona anche il cursore all'inizio del flusso di messaggi di byte.

Se un'applicazione richiama il metodo `Clear Body` dell'interfaccia `IMessage` per .NET quando il corpo di un messaggio di byte è di sola lettura, il corpo diventa scrivibile. Il metodo cancella anche il corpo.

Riferimenti correlati

Tipi di dati per elementi di dati dell'applicazione

Per essere certi che un'applicazione XMS possa scambiare messaggi con un'applicazione IBM MQ classes for JMS , entrambe le applicazioni devono essere in grado di interpretare i dati dell'applicazione nel corpo di un messaggio nello stesso modo.

Associa messaggi

Il corpo di un messaggio di associazione contiene una serie di coppie nome - valore, in cui ogni valore ha un tipo di dati associato.

Messaggi oggetto

Il corpo di un messaggio di oggetto contiene un oggettoJava o .NET serializzato.

Il processo di flusso dei messaggi

Il corpo di un messaggio di flusso contiene un flusso di valori, in cui ogni valore ha un tipo di dati associato.

Messaggi di testo

Il corpo di un messaggio di testo contiene una stringa.

IBytesMessage (per interfaccia .NET)

Un messaggio di byte è un messaggio il cui contenuto comprende un flusso di byte.

Associa messaggi

Il corpo di un messaggio di associazione contiene una serie di coppie nome - valore, in cui ogni valore ha un tipo di dati associato.

In ogni coppia nome - valore, il nome è una stringa che identifica il valore e il valore è un elemento dei dati dell'applicazione che ha uno dei tipi di dati XMS elencati in [Tabella 21 a pagina 76](#). L'ordine delle coppie nome - valore non è definito. La classe `MapMessage` contiene i metodi per impostare e richiamare le coppie nome - valore.

Un'applicazione può accedere a una coppia nome - valore in modo casuale specificandone il nome.

Un'applicazione .NET può utilizzare la proprietà MapNames per ottenere un'enumerazione dei nomi nel contenuto del messaggio della mappa.

Quando un'applicazione ottiene il valore di una coppia nome - valore, il valore può essere convertito da XMS in un altro tipo di dati. Ad esempio, per ottenere un numero intero dal corpo di un messaggio di associazione, un'applicazione può richiamare il metodo GetString della classe MapMessage , che restituisce il valore intero come stringa. Le conversioni supportate sono uguali a quelle supportate quando XMS converte un valore di proprietà da un tipo di dati a un altro. Per ulteriori informazioni sulle conversioni supportate, consultare [“Conversione implicita di un valore di proprietà da un tipo di dati a un altro”](#) a pagina 40.

Dopo che un'applicazione crea un messaggio di mappa, il corpo del messaggio è leggibile e scrivibile. Il corpo rimane leggibile e scrivibile dopo che l'applicazione ha inviato il messaggio. Quando un'applicazione riceve un messaggio di mappa, il corpo del messaggio è di sola lettura. Se un'applicazione richiama il metodo Clear Body della classe Message quando il corpo di un messaggio della mappa è di sola lettura, il corpo diventa leggibile e scrivibile. Il metodo cancella anche il corpo.

Concetti correlati

[Conversione implicita di un valore di proprietà da un tipo di dati a un altro](#)

Quando un'applicazione acquisisce il valore di una proprietà, il valore può essere convertito da XMS in un altro tipo di dati. Molte regole regolano quali conversioni sono supportate e come XMS esegue le conversioni.

Riferimenti correlati

[Tipi di dati per elementi di dati dell'applicazione](#)

Per essere certi che un'applicazione XMS possa scambiare messaggi con un'applicazione IBM MQ classes for JMS , entrambe le applicazioni devono essere in grado di interpretare i dati dell'applicazione nel corpo di un messaggio nello stesso modo.

[Messaggi byte](#)

Il corpo di un messaggio di byte contiene un flusso di byte. Il corpo contiene solo i dati effettivi ed è responsabilità delle applicazioni di invio e di ricezione interpretare tali dati.

[Messaggi oggetto](#)

Il corpo di un messaggio di oggetto contiene un oggettoJava o .NET serializzato.

[Il processo di flusso dei messaggi](#)

Il corpo di un messaggio di flusso contiene un flusso di valori, in cui ogni valore ha un tipo di dati associato.

[Messaggi di testo](#)

Il corpo di un messaggio di testo contiene una stringa.

[IMapMessage \(per l'interfaccia .NET\)](#)

Un messaggio della mappa è un messaggio il cui corpo comprende una serie di coppie nome - valore, in cui ogni valore ha un tipo di dati associato.

Messaggi oggetto

Il corpo di un messaggio di oggetto contiene un oggettoJava o .NET serializzato.

Un'applicazione XMS può ricevere un messaggio oggetto, modificarne i campi di intestazione e le proprietà e inviarlo ad un'altra destinazione. Un'applicazione può anche copiare il corpo di un messaggio oggetto e utilizzarlo per formare un altro messaggio oggetto. XMS tratta il testo di un messaggio oggetto come un array di byte.

Dopo che un'applicazione crea un messaggio oggetto, il corpo del messaggio è leggibile e scrivibile. Il corpo rimane leggibile e scrivibile dopo che l'applicazione ha inviato il messaggio. Quando un'applicazione riceve un messaggio oggetto, il corpo del messaggio è di sola lettura. Se un'applicazione richiama il metodo Clear Body dell'interfaccia IMessage per .NET quando il contenuto di un messaggio oggetto è di sola lettura, il corpo diventa leggibile e scrivibile. Il metodo cancella anche il corpo.

Riferimenti correlati

[Tipi di dati per elementi di dati dell'applicazione](#)

Per essere certi che un'applicazione XMS possa scambiare messaggi con un'applicazione IBM MQ classes for JMS , entrambe le applicazioni devono essere in grado di interpretare i dati dell'applicazione nel corpo di un messaggio nello stesso modo.

Messaggi byte

Il corpo di un messaggio di byte contiene un flusso di byte. Il corpo contiene solo i dati effettivi ed è responsabilità delle applicazioni di invio e di ricezione interpretare tali dati.

Associa messaggi

Il corpo di un messaggio di associazione contiene una serie di coppie nome - valore, in cui ogni valore ha un tipo di dati associato.

Il processo di flusso dei messaggi

Il corpo di un messaggio di flusso contiene un flusso di valori, in cui ogni valore ha un tipo di dati associato.

Messaggi di testo

Il corpo di un messaggio di testo contiene una stringa.

IObjectMessage (per interfaccia .NET)

Un messaggio oggetto è un messaggio il cui corpo comprende un oggetto Java o .NET serializzato.

Il processo di flusso dei messaggi

Il corpo di un messaggio di flusso contiene un flusso di valori, in cui ogni valore ha un tipo di dati associato.

Il tipo di dati di un valore è uno dei tipi di dati XMS elencati in [Tabella 21 a pagina 76](#).

Dopo che un'applicazione ha creato un messaggio di flusso, il corpo del messaggio è scrivibile. L'applicazione assembla i dati dell'applicazione nel corpo richiamando i metodi di scrittura appropriati dell'interfaccia `IStreamMessage` per .NET. Ogni volta che l'applicazione scrive un valore nel flusso di messaggi, il valore e il tipo di dati vengono assemblati immediatamente dopo il precedente valore scritto dall'applicazione. XMS conserva un cursore interno per ricordare la posizione dell'ultimo valore assemblato.

Quando l'applicazione invia il messaggio, il corpo del messaggio diventa di sola lettura. In questa modalità, l'applicazione può inviare il messaggio più volte.

Quando un'applicazione riceve un messaggio di flusso, il corpo del messaggio è di sola lettura. L'applicazione può utilizzare i metodi di lettura appropriati dell'interfaccia `IStreamMessage` per .NET per leggere il contenuto del flusso di messaggi. L'applicazione legge i valori in sequenza e XMS mantiene un cursore interno per ricordare la posizione dell'ultimo valore letto.

Quando un'applicazione legge un valore dal flusso di messaggi, il valore può essere convertito da XMS in un altro tipo di dati. Ad esempio, per leggere un numero intero dal flusso di messaggi, un'applicazione può chiamare il metodo `ReadString` , che restituisce il numero intero come stringa. Le conversioni supportate sono uguali a quelle supportate quando XMS converte un valore di proprietà da un tipo di dati a un altro. Per ulteriori informazioni sulle conversioni supportate, consultare [“Conversione implicita di un valore di proprietà da un tipo di dati a un altro” a pagina 40](#).

Se si verifica un errore mentre un'applicazione sta tentando di leggere un valore dal flusso di messaggi, il cursore non è avanzato. L'applicazione può recuperare dall'errore tentando di leggere il valore come un altro tipo di dati.

Se un'applicazione richiama il metodo `Reset` dell'interfaccia `IStreamMessage` per XMS quando il contenuto di un messaggio di flusso è di sola scrittura, il corpo diventa di sola lettura. Il metodo riposiziona il cursore all'inizio del flusso di messaggi.

Se un'applicazione richiama il metodo `Clear Body` dell'interfaccia `IMessage` per XMS quando il contenuto di un messaggio di flusso è di sola lettura, il corpo diventa di sola scrittura. Il metodo cancella anche il corpo.

Concetti correlati

[Conversione implicita di un valore di proprietà da un tipo di dati a un altro](#)

Quando un'applicazione acquisisce il valore di una proprietà, il valore può essere convertito da XMS in un altro tipo di dati. Molte regole regolano quali conversioni sono supportate e come XMS esegue le conversioni.

Riferimenti correlati

Tipi di dati per elementi di dati dell'applicazione

Per essere certi che un'applicazione XMS possa scambiare messaggi con un'applicazione IBM MQ classes for JMS , entrambe le applicazioni devono essere in grado di interpretare i dati dell'applicazione nel corpo di un messaggio nello stesso modo.

Messaggi byte

Il corpo di un messaggio di byte contiene un flusso di byte. Il corpo contiene solo i dati effettivi ed è responsabilità delle applicazioni di invio e di ricezione interpretare tali dati.

Associa messaggi

Il corpo di un messaggio di associazione contiene una serie di coppie nome - valore, in cui ogni valore ha un tipo di dati associato.

Messaggi oggetto

Il corpo di un messaggio di oggetto contiene un oggettoJava o .NET serializzato.

Messaggi di testo

Il corpo di un messaggio di testo contiene una stringa.

IStreamMessage (per l'interfaccia .NET)

Un messaggio di flusso è un messaggio il cui contenuto comprende un flusso di valori, dove ogni valore ha un tipo di dati associato. Il contenuto del corpo viene scritto e letto in modo sequenziale.

Messaggi di testo

Il corpo di un messaggio di testo contiene una stringa.

Dopo che un'applicazione crea un messaggio di testo, il corpo del messaggio è leggibile e scrivibile. Il corpo rimane leggibile e scrivibile dopo che l'applicazione ha inviato il messaggio. Quando un'applicazione riceve un messaggio di testo, il corpo del messaggio è di sola lettura. Se un'applicazione richiama il metodo Clear Body dell'interfaccia IMessage per .NET quando il contenuto di un messaggio di testo è di sola lettura, il corpo diventa leggibile e scrivibile. Il metodo cancella anche il corpo.

Riferimenti correlati

Tipi di dati per elementi di dati dell'applicazione

Per essere certi che un'applicazione XMS possa scambiare messaggi con un'applicazione IBM MQ classes for JMS , entrambe le applicazioni devono essere in grado di interpretare i dati dell'applicazione nel corpo di un messaggio nello stesso modo.

Messaggi byte

Il corpo di un messaggio di byte contiene un flusso di byte. Il corpo contiene solo i dati effettivi ed è responsabilità delle applicazioni di invio e di ricezione interpretare tali dati.

Associa messaggi

Il corpo di un messaggio di associazione contiene una serie di coppie nome - valore, in cui ogni valore ha un tipo di dati associato.

Messaggi oggetto

Il corpo di un messaggio di oggetto contiene un oggettoJava o .NET serializzato.

Il processo di flusso dei messaggi

Il corpo di un messaggio di flusso contiene un flusso di valori, in cui ogni valore ha un tipo di dati associato.

ITextMessage (per l'interfaccia .NET)

Un messaggio di testo è un messaggio il cui contenuto comprende una stringa.

Selettori di messaggi

Un'applicazione XMS utilizza selettori di messaggi per selezionare i messaggi che desidera ricevere.

Quando un'applicazione crea un consumatore di messaggi, può associare un'espressione del selettore di messaggi al consumatore. L'espressione del selettore messaggi specifica i criteri di selezione.

Quando un'applicazione si connette al gestore code IBM WebSphere MQ 7.0 , la selezione del messaggio viene effettuata sul lato gestore code. XMS non effettua alcuna selezione e consegna semplicemente il messaggio ricevuto dal gestore code, fornendo così prestazioni migliori.

Un'applicazione può creare più di un consumatore di messaggi, ciascuno con la propria espressione del selettore di messaggi. Se un messaggio in entrata soddisfa i criteri di selezione di più di un consumer di messaggi, XMS consegna il messaggio a ognuno di questi consumer.

Un'espressione del selettore messaggi può fare riferimento alle seguenti proprietà di un messaggio:

- Proprietà definite da JMS
- IBM
- Proprietà definite dall'applicazione

Può anche fare riferimento ai seguenti campi di intestazione del messaggio:

- JMSCorrelationID
- JMSDeliveryMode
- JMSMessageID
- JMSPriority
- JMSTimestamp
- JMSType

Un'espressione del selettore messaggi, tuttavia, non può fare riferimento ai dati nel corpo di un messaggio.

Di seguito è riportato un esempio di espressione del selettore di messaggi:

```
JMSPriority > 3 AND manufacturer = 'Jaguar' AND model in ('xj6','xj12')
```

XMS consegna un messaggio a un consumatore di messaggi con questa espressione del selettore di messaggi solo se il messaggio ha una priorità maggiore di 3; una proprietà definita dall'applicazione, produttore, con un valore di Jaguar; e un'altra applicazione definita - proprietà, modello, con un valore di xj6 o xj12.

Le regole di sintassi per la formazione di un'espressione del selettore di messaggi in XMS sono le stesse di quelle in IBM MQ classes for JMS. Per informazioni su come creare un'espressione del selettore messaggi, consultare la documentazione del prodotto IBM MQ . Si noti che, in un'espressione del selettore messaggi, i nomi delle proprietà definite da JMSdevono essere i nomi JMS e i nomi delle proprietà definite da IBMdevono essere i nomi IBM MQ classes for JMS . Non è possibile utilizzare i nomi XMS in un'espressione del selettore messaggi.

Riferimenti correlati

Parti di un messaggio XMS

Un messaggio XMS è composto da un'intestazione, una serie di proprietà e un corpo.

Campi di intestazione in un messaggio XMS

Per consentire a una applicazione XMS di scambiare messaggi con un'applicazione WebSphere JMS , l'intestazione di un messaggio XMS contiene i campi di intestazione del messaggio JMS .

Proprietà di un messaggio XMS

XMS supporta tre tipi di proprietà del messaggio: JMS proprietà definite IBM e proprietà definite dall'applicazione.

Il contenuto di un messaggio XMS

Il corpo di un messaggio contiene dati dell'applicazione. Tuttavia, un messaggio non può avere alcun corpo e comprende solo i campi di intestazione e le proprietà.

Mappatura dei messaggi XMS sui messaggi IBM MQ

I campi di intestazione JMS e le proprietà di un messaggio XMS sono associati ai campi nelle strutture di intestazione di un messaggio IBM MQ .

Mappatura dei messaggi XMS sui messaggi IBM MQ

I campi di intestazione JMS e le proprietà di un messaggio XMS sono associati ai campi nelle strutture di intestazione di un messaggio IBM MQ .

Quando un'applicazione XMS è connessa a un gestore code IBM MQ , i messaggi inviati al gestore code vengono associati ai messaggi IBM MQ nello stesso modo in cui i messaggi IBM MQ classes for JMS vengono associati ai messaggi IBM MQ in circostanze simili.

Se la proprietà `XMSC_WMQ_TARGET_CLIENT` di un oggetto di destinazione è impostata su `XMSC_WMQ_TARGET_DEST_JMS`, i campi di intestazione JMS e le proprietà di un messaggio inviato alla destinazione vengono associati ai campi nelle strutture di intestazioni MQMD e MQRFH2 del messaggio IBM MQ . Se si imposta la proprietà `XMSC_WMQ_TARGET_CLIENT` in questo modo, si presuppone che l'applicazione che riceve il messaggio possa gestire un'intestazione MQRFH2 . L'applicazione ricevente potrebbe quindi essere un'altra applicazione XMS , un'applicazione IBM MQ classes for JMS o un'applicazione IBM MQ nativa progettata per gestire un'intestazione di MQRFH2 .

Se invece la proprietà `XMSC_WMQ_TARGET_CLIENT` di un oggetto di destinazione è impostata su `XMSC_WMQ_TARGET_DEST_MQ`, i campi di intestazione JMS e le proprietà di un messaggio inviato alla destinazione vengono associati ai campi nella struttura dell'intestazione MQMD del messaggio IBM MQ . Il messaggio non contiene un'intestazione MQRFH2 e tutti i campi di intestazione JMS e le proprietà che non possono essere associati ai campi nella struttura dell'intestazione MQMD vengono ignorati. L'applicazione che riceve il messaggio può quindi essere una IBM MQ nativa che non è progettata per gestire un'intestazione MQRFH2 .

I messaggi IBM MQ ricevuti da un gestore code vengono associati ai messaggi XMS nello stesso modo in cui i messaggi IBM MQ vengono associati ai messaggi IBM MQ classes for JMS in circostanze simili.

Se un messaggio IBM MQ in entrata dispone di un'intestazione MQRFH2 , il messaggio XMS risultante ha un corpo il cui tipo è determinato dal valore della proprietà **Msd** contenuto nella cartella mcd dell'intestazione MQRFH2 . Se la proprietà **Msd** non è presente nell'intestazione MQRFH2 o se il messaggio IBM MQ non ha un'intestazione MQRFH2 , il messaggio XMS risultante ha un corpo il cui tipo è determinato dal valore del campo *Format* nell'intestazione MQMD. Se il campo *Format* è impostato su MQFMT_STRING, il messaggio XMS è un messaggio di testo. Altrimenti, il messaggio XMS è un messaggio di byte. Se il messaggio IBM MQ non ha un'intestazione MQRFH2 , vengono impostati solo i campi di intestazione JMS e le proprietà che possono derivare dai campi nell'intestazione MQMD.

Per ulteriori informazioni sull'associazione dei messaggi IBM MQ classes for JMS sui messaggi IBM MQ , consultare la documentazione del prodotto IBM MQ .

Riferimenti correlati

Parti di un messaggio XMS

Un messaggio XMS è composto da un'intestazione, una serie di proprietà e un corpo.

Campi di intestazione in un messaggio XMS

Per consentire a una applicazione XMS di scambiare messaggi con un'applicazione WebSphere JMS , l'intestazione di un messaggio XMS contiene i campi di intestazione del messaggio JMS .

Proprietà di un messaggio XMS

XMS supporta tre tipi di proprietà del messaggio: JMS proprietà definite IBM e proprietà definite dall'applicazione.

Il contenuto di un messaggio XMS

Il corpo di un messaggio contiene dati dell'applicazione. Tuttavia, un messaggio non può avere alcun corpo e comprende solo i campi di intestazione e le proprietà.

Selettori di messaggi

Un'applicazione XMS utilizza selettori di messaggi per selezionare i messaggi che desidera ricevere.

Lettura e scrittura del descrittore del messaggio da una applicazione IBM Message Service Client for .NET

È possibile accedere a tutti i campi del descrittore del messaggio (MQMD) di un messaggio IBM MQ ad eccezione di StrucId e Version; BackoutCount può essere letto ma non scritto. Questa funzione è disponibile solo quando ci si connette a un gestore code IBM WebSphere MQ 6.0 o successivo ed è controllata dalle proprietà di destinazione descritte in seguito.

Gli attributi del messaggio forniti da IBM Message Service Client for .NET facilitano le applicazioni XMS per impostare i campi MQMD e anche per guidare applicazioni IBM WebSphere MQ .

Alcune limitazioni si applicano quando si utilizza la messaggistica di pubblicazione / sottoscrizione. Ad esempio, i campi MQMD come MsgID e CorrelId, se impostati, vengono ignorati.

La funzione descritta in questo argomento non è disponibile per la messaggistica di pubblicazione / sottoscrizione quando ci si connette a un gestore code IBM WebSphere MQ 6.0 . Non è disponibile anche quando la proprietà **PROVIDERVERSION** è impostata su 6.

Accesso ai dati del messaggio IBM MQ da una applicazione IBM Message Service Client for .NET

È possibile accedere ai dati completi del messaggio IBM MQ , inclusa l'intestazione MQRFH2 (se presente) e qualsiasi altra intestazione IBM MQ (se presente) all'interno di un'applicazione IBM Message Service Client for .NET come corpo di un JMSBytesMessage.

La funzione descritta in questo argomento è disponibile solo quando ci si connette a un gestore code IBM WebSphere MQ 7.0 o successivo e il provider di messaggistica WebSphere MQ è in modalità normale.

Le proprietà dell'oggetto di destinazione determinano in che modo l'applicazione XMS accede all'intero messaggio IBM MQ (inclusa l'intestazione MQRFH2 , se presente) come corpo di un JMSBytesMessage.

Risoluzione dei problemi

Questa sezione fornisce informazioni che consentono di rilevare e gestire i problemi quando si utilizza IBM Message Service Client for .NET.

Questa sezione contiene i seguenti argomenti:

- [“Configurazione della traccia per applicazioni .NET” a pagina 83](#)
- [“Configurazione FFDC per applicazioni .NET” a pagina 88](#)
- [“Suggerimenti per la risoluzione dei problemi” a pagina 88](#)

Configurazione della traccia per applicazioni .NET

Per le applicazioni di XMS .NET , è possibile configurare la traccia da un file di configurazione dell'applicazione e dalle variabili di ambiente XMS . È possibile selezionare i componenti che si desidera tracciare. La traccia viene normalmente utilizzata sotto la guida del supporto IBM .

La traccia per XMS .NET è basata sull'infrastruttura di traccia .NET standard.

Per impostazione predefinita, è disabilitata tutta la traccia ad eccezione della traccia degli errori. È possibile attivare la funzione di traccia e configurare le relative impostazioni in uno dei seguenti modi:

- Utilizzando un file di configurazione dell'applicazione con un nome costituito dal nome del programma eseguibile a cui si riferisce il file, con il suffisso `.config`. Ad esempio, il file di configurazione dell'applicazione per `text.exe` avrà il nome `text.exe.config`. L'utilizzo di un file di configurazione dell'applicazione è il metodo preferito per abilitare la traccia per applicazioni XMS .NET . Per ulteriori dettagli, fare riferimento a [“Traccia della configurazione utilizzando un file di configurazione dell'applicazione” a pagina 84](#).
- Utilizzando le variabili di ambiente XMS come per le applicazioni XMS C o C++. Per ulteriori dettagli, fare riferimento a [“Configurazione della traccia utilizzando le variabili di ambiente XMS” a pagina 86](#).

Il file di traccia attivo ha un nome nel formato `xms_tracePID.log` dove *PID* rappresenta l'ID processo dell'applicazione. La dimensione del file di traccia attivo è per impostazione predefinita limitata a 20 MB. Quando viene raggiunto questo limite, il file viene ridenominato e archiviato. I file archiviati hanno il formato `xms_tracePID_YY.MM.DD_HH.MM.SS.log`

Per impostazione predefinita, il numero di file di traccia conservati è quattro, ossia un file attivo e tre file archiviati. Questi quattro file vengono utilizzati come buffer di rolling fino a quando l'applicazione non si arresta, con il file meno recente rimosso e sostituito dal file più recente. È possibile modificare il numero di file di traccia specificando un altro numero nel file di configurazione dell'applicazione. Tuttavia, ci devono essere almeno due file (un file attivo e uno archiviato).

Sono disponibili due formati di file di traccia:

- I file di traccia del formato di base sono leggibili, in formato WebSphere Application Server . Questo formato è il formato file di traccia predefinito. Il formato di base non è compatibile con gli strumenti dell'analizzatore di traccia.
- I file di traccia di formato avanzato sono compatibili con gli strumenti di analisi della traccia. È necessario specificare che si desidera produrre i file di traccia in formato avanzato nel file di configurazione dell'applicazione.

Le voci di traccia contengono le seguenti informazioni:

- La data e l'ora in cui è stata registrata la traccia
- Il nome della classe
- Il tipo di traccia
- Il messaggio di traccia

Il seguente esempio mostra un estratto da una traccia:

```
[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
```

Nell'esempio precedente, il formato è:

[Date Time:Microsecs] or Exit	Thread-id	Classname	Trace-type	Methodname	Entry
----------------------------------	-----------	-----------	------------	------------	-------

dove Trace-type è:

- > per la voce
- < per Esci
- d per informazioni di debug

Traccia della configurazione utilizzando un file di configurazione dell'applicazione

Il modo preferito di configurare la traccia per le applicazioni XMS .NET è con un file di configurazione dell'applicazione. La sezione di traccia di questo file include i parametri che definiscono cosa deve essere tracciato, l'ubicazione del file di traccia e la dimensione massima consentita, il numero di file di traccia utilizzati e il formato del file di traccia.

Per attivare la traccia utilizzando il file di configurazione dell'applicazione, è sufficiente posizionare il file nella stessa directory del file eseguibile per l'applicazione.

La traccia può essere abilitata sia per componente che per tipo di traccia. È anche possibile attivare la traccia per un intero gruppo di traccia. È possibile attivare la traccia per i componenti in una gerarchia singolarmente o collettivamente. I tipi di traccia disponibili includono:

- Traccia di Debug

- Traccia eccezione
- Avvertenze, messaggi informativi e messaggi di errore
- Traccia di entrata e di uscita del metodo

Il seguente esempio mostra le impostazioni di traccia definite nella sezione Traccia di un file di configurazione dell'applicazione:

```
<?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>
```

Tabella 22 a pagina 85 descrive le impostazioni dei parametri in modo più dettagliato.

Tabella 22. Impostazioni del parametro di traccia del file di configurazione dell'applicazione	
Parametro	Descrizione
<code>traceSpecification=ComponentName=type=state</code>	<p><i>ComponentName</i> è il nome della classe che si desidera tracciare. È possibile utilizzare un carattere jolly * in questo nome. Ad esempio, <code>*=all=enabled</code> specifica che si desidera tracciare tutte le classi e <code>IBM.XMS.impl.*=all=enabled</code> specifica che si richiede solo la traccia API.</p> <p><i>type</i> può essere uno dei seguenti tipi di traccia:</p> <ul style="list-style-type: none"> • tutti • Debug • evento • EntryExit <p><i>state</i> può essere abilitato o disabilitato.</p> <p>È possibile raggruppare più elementi di traccia utilizzando un delimitatore ':' (due punti).</p>
<code>traceFilePath="filename"</code>	<p>Se non si specifica un percorso traceFile o se il percorso traceFile è presente ma contiene una stringa vuota, il file di traccia viene collocato nella directory corrente. Per memorizzare il file di traccia in una directory denominata, specificare il nome della directory nel percorso traceFile, ad esempio:</p> <pre>traceFilePath="c:\somepath"</pre>
<code>traceFileSize="size"</code>	<p>La dimensione massima consentita del file di traccia. Quando un file raggiunge questa dimensione, viene archiviato e ridenominato. Il valore massimo predefinito è 20 KB, specificato come <code>traceFileSize="20000000"</code>.</p>

Tabella 22. Impostazioni del parametro di traccia del file di configurazione dell'applicazione (Continua)	
Parametro	Descrizione
traceFileNumber="number"	Il numero di file di traccia da conservare. Il valore predefinito è 4 (un file attivo e 3 file di archiviazione). Il numero minimo consentito è 2.
traceFormat="format"	Il formato di traccia predefinito è di base. I file di traccia vengono prodotti in questo formato se si specifica traceFormat="basic", se non si specifica traceFormat se traceFormat è presente ma contiene una stringa vuota. Se si richiede una traccia compatibile con gli strumenti del programma di analisi della traccia, è necessario specificare traceFormat="advanced".

Le impostazioni di traccia nel file di configurazione dell'applicazione sono dinamiche e vengono rilette ogni volta che il file viene salvato o sostituito. Se vengono rilevati degli errori nel file dopo che è stato modificato, le impostazioni del file di traccia ritornano ai valori predefiniti.

Concetti correlati

Configurazione della traccia utilizzando le variabili di ambiente XMS

Come alternativa all'utilizzo di un file di configurazione dell'applicazione, è possibile attivare la traccia utilizzando le variabili d'ambiente XMS . Queste variabili di ambiente vengono utilizzate solo se nel file di configurazione dell'applicazione non è presente alcuna specifica di traccia.

Configurazione della traccia utilizzando le variabili di ambiente XMS

Come alternativa all'utilizzo di un file di configurazione dell'applicazione, è possibile attivare la traccia utilizzando le variabili d'ambiente XMS . Queste variabili di ambiente vengono utilizzate solo se nel file di configurazione dell'applicazione non è presente alcuna specifica di traccia.

Per configurare la traccia per una applicazione XMS .NET , impostare le seguenti variabili di ambiente prima di eseguire l'applicazione:

Tabella 23. Impostazioni della variabile di ambiente per la traccia .NET			
Variabili di ambiente	Valore predefinito	Impostazioni	Significato
TRACE_XMS ON	Non applicabile	Non applicabile: il valore di questa variabile viene ignorato	Se XMS_TRACE_ON è impostato, tutta la traccia è abilitata per impostazione predefinita.

Tabella 23. Impostazioni della variabile di ambiente per la traccia .NET (Continua)

Variabili di ambiente	Valore predefinito	Impostazioni	Significato
PERCORSO_FILE_TRACCI A_X	Directory di lavoro corrente	/percorsodir/	<p>Il percorso della directory in cui vengono scritti i record FFDC e di traccia.</p> <p>XMS crea file di traccia e FFDC nella directory di lavoro corrente, a meno che non si specifichi un'ubicazione alternativa. È possibile specificare un'ubicazione alternativa impostando la variabile di ambiente XMS_TRACE_FILE_PATH sul nome percorso completo della directory in cui si desidera che XMS crei i file FFDC e di traccia. È necessario impostare questa variabile di ambiente prima di iniziare l'applicazione che si desidera tracciare. È necessario assicurarsi che l'identificativo utente con cui viene eseguita l'applicazione disponga dell'autorizzazione per scrivere nella directory in cui XMS crea i file di traccia e FFDC.</p>
FORMATO_TRACCIA_XMS	DI BASE	BASE, AVANZATO	<p>Specifica il formato di traccia richiesto, che può essere BASIC o ADVANCED. Il formato predefinito è BASIC. Il formato ADVANCED è compatibile con gli strumenti dell'analizzatore di traccia.</p>
SPECIFICA_TRACCIA_XM S	Non applicabile	<p>Vedere “Traccia della configurazione utilizzando un file di configurazione dell'applicazione” a pagina 84</p>	<p>Sovrascrive la specifica di traccia, che segue il formato specificato in “Traccia della configurazione utilizzando un file di configurazione dell'applicazione” a pagina 84.</p>

Concetti correlati

[Traccia della configurazione utilizzando un file di configurazione dell'applicazione](#)

Il modo preferito di configurare la traccia per le applicazioni XMS .NET è con un file di configurazione dell'applicazione. La sezione di traccia di questo file include i parametri che definiscono cosa deve essere tracciato, l'ubicazione del file di traccia e la dimensione massima consentita, il numero di file di traccia utilizzati e il formato del file di traccia.

Configurazione FFDC per applicazioni .NET

Per l'implementazione .NET di XMS, viene prodotto un file FFDC per ogni FFDC.

I file FFDC (First Failure Data Capture) sono memorizzati in file di testo leggibili. Questi file hanno il formato `xmsffdcprocessID_DateTimestamp.txt`. Un esempio di nome file è `xmsffdc264_2006.01.06T13.18.52.990955.txt`. La data/ora contiene una risoluzione di microsecondi.

I file iniziano con la data e l'ora in cui si è verificata l'eccezione, seguita dal tipo di eccezione. I file includono un breve probeId univoco, che può essere utilizzato per individuare il punto in cui si è verificato questo FFDC.

Non è necessario eseguire alcuna configurazione per attivare FFDC. Per impostazione predefinita, tutti i file FFDC vengono scritti nella directory corrente. Tuttavia, se necessario, è possibile specificare una directory differente modificando `ffdcDirectory` nella sezione Traccia del file di configurazione dell'applicazione. Nel seguente esempio, tutti i file di traccia vengono registrati nella directory `c:\client\ffdc`:

```
<IBM.XMS>  
  <Trace ffdc=true ffdcDirectory="c:\client\ffdc"/>  
</IBM.XMS>
```

È possibile disabilitare la traccia impostando FFDC su false nella sezione Traccia del file di configurazione dell'applicazione.

Se non si utilizza un file di configurazione dell'applicazione, FFDC è attivo e la traccia è disattivata.

Suggerimenti per la risoluzione dei problemi

Utilizzare questi suggerimenti per risolvere i problemi relativi all'uso di XMS.

Un'applicazione XMS non riesce a collegarsi a un gestore code (MQRC_NOT_AUTHORIZED)

Il client XMS .NET potrebbe avere un comportamento diverso da quello del client IBM MQ JMS . Pertanto, potresti scoprire che la tua applicazione XMS non può connettersi al tuo gestore code, anche se la tua applicazione JMS può farlo.

- Una soluzione semplice a questo problema consiste nel tentare di utilizzare un ID utente che non sia più lungo di 12 caratteri e che sia autorizzato completamente nell'elenco di autorizzazioni del gestore code. Se questa soluzione non è ideale, un approccio diverso ma più complesso sarebbe quello di utilizzare le uscite di sicurezza. Se hai bisogno di ulteriore aiuto su questo problema, contatta il Supporto IBM per assistenza.
- Se si configura la proprietà `XMSC_USERID` del factory di connessione, deve corrispondere all'ID utente e alla password dell'utente collegato. Se non si imposta questa proprietà, il gestore code utilizza l'ID utente dell'utente collegato per impostazione predefinita.
- L'autenticazione utente per IBM MQ viene eseguita utilizzando i dettagli dell'utente attualmente collegato e non le informazioni fornite in `XMSC.USERID` e `XMSC.PASSWORD` . Questo è progettato per mantenere la coerenza con IBM MQ. Per ulteriori informazioni sull'autenticazione, fare riferimento a *Informazioni di autenticazione* nella documentazione del prodotto IBM MQ in linea.

Connessione reindirizzata al motore di messaggistica

Quando ci si connette a un SIB (service integration bus) WebSphere Application Server 6.0.2 , tutte le connessioni possono essere reindirizzate dall'endpoint del provider originale al motore di messaggistica scelto dal bus per tale connessione client. In questo caso, reindirizzerà sempre la connessione a un server host specificato dal nome host, piuttosto che da un indirizzo IP. Pertanto, è possibile che si verifichino problemi di connessione se non è possibile risolvere il nome host.

Per connettersi correttamente al SIB (Service Integration Bus) WebSphere Application Server 6.0.2 , potrebbe essere necessario fornire un'associazione tra i nomi host e gli indirizzi IP sulla macchina host del client. Ad esempio, è possibile specificare l'associazione in una tabella di host locali sulla macchina host client.

Supporto per l'autenticazione della password di tipo telnet-like

Il protocollo XMS .NET Real Time Transport supporta solo l'autenticazione della password di tipo telnet. Il protocollo XMS .NET Real Time Transport non supporta la qualità della protezione.

Impostazione dei valori per il tipo di proprietà double

Su una piattaforma Windows a 64 bit, i metodi SetDoubleProperty () o GetDoubleProperty () potrebbero non funzionare correttamente quando si impostano o si richiamano valori per il tipo di proprietà double, se i valori sono inferiori a Double.Epsilon.

Ad esempio, se si tenta di impostare un valore di 4.9E-324 per una proprietà con tipo double, le piattaforme Windows a 64 bit lo considerano come 0.0. Quindi, in un ambiente di messaggistica distribuito, se un JMS o un'altra applicazione imposta il valore per una proprietà doppia come 4.9E-324 su qualsiasi macchina UNIX o Windows a 32 bit e XMS .NET viene eseguito su una macchina a 64 bit, il valore restituito da GetDoubleProperty () è 0.0. Questo è un problema noto con Microsoft .NET Framework 2.0 Framework.

Condizioni di errore che possono essere gestite al runtime

I codici di ritorno dalle chiamate API sono condizioni di errore che possono essere gestite al runtime. Il modo in cui si affronta questo tipo di errore dipende dal fatto che si stia utilizzando l'API C o C++ .

Come rilevare gli errori al runtime

Se un'applicazione richiama una funzione API C e la chiamata ha esito negativo, viene restituita una risposta con un codice di ritorno diverso da XMS_OK con un blocco di errori XMS contenente ulteriori informazioni sul motivo dell'errore.

L'API C++ genera un'eccezione quando viene utilizzato un metodo.

Un'applicazione utilizza un listener di eccezioni per ricevere una notifica asincrona di un problema con una connessione. Il listener di eccezioni viene fornito e inizializzato utilizzando l'API XMS C o C++ .

Come gestire gli errori al runtime

Alcune condizioni di errore sono un'indicazione che alcune risorse non sono disponibili e l'azione che un'applicazione può intraprendere dipende dalla funzione XMS richiamata dall'applicazione. Ad esempio, se una connessione non riesce a collegarsi al server, l'applicazione potrebbe riprovare periodicamente fino a quando non viene effettuata una connessione. Un blocco di errori o un'eccezione XMS potrebbe non contenere informazioni sufficienti per determinare quale azione intraprendere e, in queste situazioni, spesso vi è un blocco di errori collegato o un'eccezione che contiene informazioni diagnostiche più specifiche.

Nell'API C, verifica sempre una risposta con un codice di ritorno diverso da XMS_OK e inoltra sempre un blocco di errore sulla chiamata API. L'azione eseguita di solito dipende dalla funzione API utilizzata dall'applicazione.

Nell'API C++ , includere sempre le chiamate ai metodi in un blocco try e, per rilevare tutti i tipi di eccezione XMS , specificare la classe Exception nel costrutto catch.

Il listener delle eccezioni è un percorso di condizione di errore asincrono che può essere avviato in qualsiasi momento. Quando la funzione del listener di eccezioni viene avviata, sul proprio thread, di solito è un'indicazione di un malfunzionamento più grave rispetto a una normale condizione di errore dell'API XMS . È possibile intraprendere qualsiasi azione appropriata, ma è necessario prestare attenzione a seguire le regole per il modello di thread XMS come descritto in [“Il modello di threading” a pagina 21.](#)

Concetti correlati

[Il modello di threading](#)

Le regole generali gestiscono il modo in cui un'applicazione a più thread può utilizzare gli oggetti XMS .

Riferimento per Message Service Client for .NET

Questa sezione di riferimento fornisce informazioni che consentono di utilizzare Message Service Client for .NET. Queste informazioni consentono di eseguire le attività coinvolte nella programmazione con XMS.

.NETInterfacce

Questa sezione documenta le interfacce della classe .NET e le relative proprietà e metodi.

La seguente tabella riepiloga tutte le interfacce definite nello spazio dei nomi IBM . XMS .

Interfaccia	Descrizione
“IBytesMessage” a pagina 93	Un messaggio di byte è un messaggio il cui contenuto comprende un flusso di byte.
“IConnessione” a pagina 103	Un oggetto Connection rappresenta la connessione attiva dell'applicazione ad un server di messaggi.
“IConnectionFactory” a pagina 106	Un'applicazione utilizza una produzione connessioni per creare una connessione.
“Dati IConnectionMeta” a pagina 108	Un oggetto ConnectionMetaData fornisce informazioni su una connessione.
“IDestinazione” a pagina 109	Una destinazione è il punto in cui un'applicazione invia messaggi o è un'origine da cui un'applicazione riceve messaggi o entrambi.
“ExceptionListener” a pagina 110	Un'applicazione utilizza un listener di eccezioni per ricevere una notifica asincrona di un problema con una connessione.
“Eccezione IllegalState” a pagina 111	XMS genera questa eccezione se un'applicazione richiama un metodo in un momento non corretto o inappropriato o se XMS non si trova in uno stato appropriato per l'operazione richiesta.
“InitialContext” a pagina 111	Un'applicazione utilizza un oggetto InitialContext per creare oggetti dalle definizioni di oggetto richiamati da un repository di oggetti gestiti.
“IDeception InvalidClient” a pagina 113	XMS genera questa eccezione se un'applicazione tenta di impostare un identificatore client per una connessione, ma l'identificativo client non è valido o è già in uso.

Tabella 24. Riepilogo delle interfacce della classe .NET (Continua)

Interfaccia	Descrizione
“Eccezione InvalidDestination” a pagina 114	XMS genera questa eccezione se un'applicazione specifica una destinazione non valida.
“Eccezione InvalidSelector” a pagina 114	XMS genera questa eccezione se un'applicazione fornisce un'espressione del selettore di messaggi la cui sintassi non è valida.
“IMapMessage” a pagina 114	Un messaggio della mappa è un messaggio il cui corpo comprende una serie di coppie nome - valore, in cui ogni valore ha un tipo di dati associato.
“IMessaggio” a pagina 124	Un oggetto Message rappresenta un messaggio che un'applicazione invia o riceve. IMessage è una superclasse per le classi di messaggi come IMapMessage.
“IMessageConsumer” a pagina 130	Un'applicazione utilizza un consumatore di messaggi per ricevere i messaggi inviati a una destinazione.
“MessageEOFException” a pagina 133	XMS genera questa eccezione se XMS rileva la fine di un flusso di messaggi di byte quando un'applicazione sta leggendo il contenuto di un messaggio di byte.
“Eccezione MessageFormat” a pagina 133	XMS genera questa eccezione se XMS rileva un messaggio con un formato non valido.
“IMessageListener (delegato)” a pagina 134	Un'applicazione utilizza un listener di messaggi per ricevere i messaggi asincrono.
“MessageNotReadableException” a pagina 134	XMS genera questa eccezione se un'applicazione tenta di leggere il corpo di un messaggio di sola scrittura.
“MessageNotWritableException” a pagina 134	XMS genera questa eccezione se un'applicazione tenta di scrivere nel corpo di un messaggio di sola lettura.
“IMessageProducer” a pagina 135	Un'applicazione utilizza un produttore di messaggi per inviare messaggi a una destinazione.
“IObjectMessage” a pagina 140	Un messaggio oggetto è un messaggio il cui corpo comprende un oggetto Java o .NET serializzato.
“IPropertyContext” a pagina 141	IPropertyContext è una superclasse astratta che contiene metodi che richiamano e impostano le proprietà. Questi metodi vengono ereditati dalle altre classi.
“IQueueBrowser” a pagina 151	Un'applicazione utilizza un browser della coda per ricercare i messaggi su una coda senza rimuoverli.
“Richiedente” a pagina 153	Un'applicazione utilizza un richiedente per inviare un messaggio di richiesta e attendere e ricevere la risposta.
“Eccezione ResourceAllocation” a pagina 154	XMS genera questa eccezione se XMS non può allocare le risorse richieste da un metodo.

Tabella 24. Riepilogo delle interfacce della classe .NET (Continua)

Interfaccia	Descrizione
“SecurityException” a pagina 154	XMS genera questa eccezione se l'identificativo utente e la parola d'ordine forniti per autenticare un'applicazione vengono rifiutati. XMS genera questa eccezione anche se un controllo di autorizzazione non riesce e impedisce il completamento di un metodo.
“ISessione” a pagina 155	Una sessione è un contesto a thread singolo per l'invio e la ricezione di messaggi.
“IStreamMessage” a pagina 166	Un messaggio di flusso è un messaggio il cui contenuto comprende un flusso di valori, dove ogni valore ha un tipo di dati associato.
“ITextMessage” a pagina 175	Un messaggio di testo è un messaggio il cui contenuto comprende una stringa.
“TransactionInProgressException” a pagina 176	XMS genera questa eccezione se un'applicazione richiede un'operazione non valida perché è in corso una transazione.
“TransactionRolledBackException” a pagina 177	XMS genera questa eccezione se un'applicazione richiama Session.commit() per eseguire il commit della transazione corrente, ma viene eseguito il rollback della transazione.
XMSC	Per .NET, i valori e i nomi proprietà XMS sono definiti in questa classe come costanti pubbliche. Per ulteriori dettagli, fare riferimento a “Proprietà degli oggetti XMS” a pagina 180 .
“Eccezione XMSEException” a pagina 177	Se XMS rileva un errore durante l'elaborazione di una chiamata a un metodo .NET, XMS genera un'eccezione. Un'eccezione è un oggetto che contiene informazioni sull'errore. Esistono diversi tipi di eccezione XMS e un oggetto XMSEException è solo un tipo di eccezione. Tuttavia, la classe XMSEException è una superclasse delle altre classi di eccezione XMS. XMS genera un oggetto XMSEException in situazioni in cui nessuno degli altri tipi di eccezione è appropriato.
“XMSFactoryFactory” a pagina 178	Se un'applicazione non utilizza oggetti gestiti, utilizzare questa classe per creare factory di connessione, code e argomenti.

La definizione di ciascun metodo elenca i codici di errore che XMS potrebbe restituire se rileva un errore durante l'elaborazione di una chiamata al metodo. Ogni codice di eccezione è rappresentato dalla sua costante denominata, che ha un'eccezione corrispondente.

Concetti correlati

[Creazione di applicazioni personalizzate](#)

Si creano le proprie applicazioni come si creano le applicazioni di esempio.

[Scrittura di applicazioni XMS](#)

Gli argomenti in questa sezione forniscono informazioni utili per la scrittura di applicazioni XMS.

[Scrittura di applicazioni XMS .NET](#)

Gli argomenti contenuti in questa sezione forniscono informazioni utili per la scrittura di applicazioni XMS .NET .

Riferimenti correlati

Proprietà degli oggetti XMS capitolo documenta le proprietà dell'oggetto definite da XMS.

IBytesMessage

Un messaggio di byte è un messaggio il cui contenuto comprende un flusso di byte.

Gerarchia di eredità:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IMessage
|
+----IBM.XMS.IBytesMessage
```

Riferimenti correlati

Messaggi byte

Il corpo di un messaggio di byte contiene un flusso di byte. Il corpo contiene solo i dati effettivi ed è responsabilità delle applicazioni di invio e di ricezione interpretare tali dati.

Proprietà di .NET

BodyLength - Richiama la lunghezza del body

Interfaccia:

```
Int64 BodyLength
{
    get;
}
```

Richiamare la lunghezza del corpo del messaggio in byte quando il corpo del messaggio è di sola lettura

Il valore restituito è la lunghezza dell'intero corpo indipendentemente da dove è attualmente posizionato il cursore per la lettura del messaggio.

Eccezioni:

- Eccezione XMSEException
- MessageNotReadableException

Metodi

ReadBoolean - Leggi valore booleano

Interfaccia:

```
Boolean ReadBoolean();
```

Leggere un valore booleano dal flusso di messaggi byte.

Parametri:

Nessuno

Restituisce:

Il valore booleano letto.

Eccezioni:

- Eccezione XMSEException
- MessageNotReadableException
- MessageEOFException

Byte ReadSigned- Byte di lettura

Interfaccia:

```
Int16 ReadSignedByte();
```

Leggere il byte successivo dal flusso di messaggi di byte come un numero intero a 8 bit con segno.

Parametri:

Nessuno

Restituisce:

Il byte letto.

Eccezioni:

- Eccezione XMSEException
- MessageNotReadableException
- MessageEOFException

ReadBytes - Byte letti

Interfaccia:

```
Int32 ReadBytes(Byte[] array);  
Int32 ReadBytes(Byte[] array, Int32 length);
```

Leggere un array di byte dal flusso di messaggi di byte a partire dalla posizione corrente del cursore.

Parametri:**array (output)**

Il buffer che contiene l'array di byte letti. Se il numero di byte rimanenti da leggere dal flusso prima della chiamata è maggiore o uguale alla lunghezza del buffer, il buffer viene riempito. Altrimenti, il buffer viene riempito parzialmente con tutti i restanti byte.

Se si specifica un puntatore null sull'input, il metodo ignora i byte senza leggerli. Se il numero di byte rimanenti da leggere dal flusso prima che la chiamata sia maggiore o uguale alla lunghezza del buffer, il numero di byte ignorati è uguale alla lunghezza del buffer. Altrimenti, tutti i byte rimanenti vengono ignorati. Il cursore rimane nella successiva posizione da leggere nel flusso di messaggi di byte.

lunghezza (input)

La lunghezza del buffer in byte

Restituisce:

Il numero di byte letti nel buffer. Se il buffer è parzialmente riempito, il valore è inferiore alla lunghezza del buffer, indicando che non ci sono altri byte rimanenti da leggere. Se non restano byte da leggere dal flusso prima della chiamata, il valore è XMSC_END_OF_STREAM.

Se si specifica un puntatore nullo sull'immissione, il metodo non restituisce alcun valore.

Eccezioni:

- Eccezione XMSEException

- MessageNotReadableException

ReadChar - Carattere di lettura

Interfaccia:

```
Char ReadChar();
```

Leggere i 2 byte successivi dal flusso di messaggi byte come carattere.

Parametri:

Nessuno

Restituisce:

Il carattere che viene letto.

Eccezioni:

- Eccezione XMSEException
- MessageNotReadableException
- MessageEOFException

ReadDouble - Leggi numero a virgola mobile di precisione doppia

Interfaccia:

```
Double ReadDouble();
```

Leggere i successivi 8 byte dal flusso di messaggi byte come un numero a virgola mobile a doppia precisione.

Parametri:

Nessuno

Restituisce:

Il numero a virgola mobile di precisione doppia letto.

Eccezioni:

- Eccezione XMSEException
- MessageNotReadableException
- MessageEOFException

ReadFloat - Leggi numero a virgola mobile

Interfaccia:

```
Single ReadFloat();
```

Leggere i successivi 4 byte dal flusso di messaggi byte come numero a virgola mobile.

Parametri:

Nessuno

Restituisce:

Il numero a virgola mobile letto.

Eccezioni:

- Eccezione XMSEException
- MessageNotReadableException
- MessageEOFException

ReadInt - Lettura numero intero

Interfaccia:

```
Int32 ReadInt();
```

Leggere i successivi 4 byte dal flusso di messaggi di byte come numero intero a 32 bit con segno.

Parametri:

Nessuno

Restituisce:

Il numero intero letto.

Eccezioni:

- Eccezione XMSEException
- MessageNotReadableException
- MessageEOFException

ReadLong - Lettura numero intero lungo

Interfaccia:

```
Int64 ReadLong();
```

Leggere i successivi 8 byte dal flusso di messaggi di byte come un numero intero a 64 bit con segno.

Parametri:

Nessuno

Restituisce:

Il numero intero lungo letto.

Eccezioni:

- Eccezione XMSEException
- MessageNotReadableException
- MessageEOFException

ReadShort - Lettura numero intero breve

Interfaccia:

```
Int16 ReadShort();
```

Leggere i 2 byte successivi dal flusso di messaggi di byte come un numero intero a 16 bit con segno.

Parametri:

Nessuno

Restituisce:

Il numero intero breve letto.

Eccezioni:

- Eccezione XMSEException
- MessageNotReadableException
- MessageEOFException

ReadByte - Lettura byte senza segno

Interfaccia:

```
Byte ReadByte();
```

Leggere il byte successivo dal flusso di messaggi di byte come un numero intero a 8 bit senza segno.

Parametri:

Nessuno

Restituisce:

Il byte letto.

Eccezioni:

- Eccezione XMSEException
- MessageNotReadableException
- MessageEOFException

ReadUnsignedShort - Read Unsigned Short Integer

Interfaccia:

```
Int32 ReadUnsignedShort();
```

Leggere i 2 byte successivi dal flusso di messaggi di byte come un numero intero a 16 bit senza segno.

Parametri:

Nessuno

Restituisce:

Il numero intero breve senza segno che viene letto.

Eccezioni:

- Eccezione XMSEException
- MessageNotReadableException
- MessageEOFException

ReadUTF - Leggi stringa UTF

Interfaccia:

```
String ReadUTF();
```

Leggere una stringa, codificata in UTF-8, dal flusso di messaggi di byte.

Nota: Prima di richiamare `ReadUTF()`, assicurarsi che il cursore del buffer punti all'inizio del flusso di messaggi di byte.

Parametri:

Nessuno

Restituisce:

Un oggetto stringa che incapsula la stringa letta.

Eccezioni:

- Eccezione `XMSEException`
- `MessageNotReadableException`
- `MessageEOFException`

Reimposta - Reimposta

Interfaccia:

```
void Reset();
```

Inserire il contenuto del messaggio in modalità di sola lettura e riposizionare il cursore all'inizio del flusso di messaggi byte.

Parametri:

Nessuno

Restituisce:

Nulla

Eccezioni:

- Eccezione `XMSEException`
- `MessageNotReadableException`

WriteBoolean - Scrivi valore booleano

Interfaccia:

```
void WriteBoolean(Boolean value);
```

Scrivere un valore booleano nel flusso di messaggi di byte.

Parametri:

valore (immissione)

Il valore booleano da scrivere.

Restituisce:

Nulla

Eccezioni:

- Eccezione `XMSEException`
- `MessageNotWritableException`

WriteByte - Byte di scrittura

Interfaccia:

```
void WriteByte(Byte value);  
void WriteSignedByte(Int16 value);
```

Scrivere un byte nel flusso di messaggi byte.

Parametri:

valore (immissione)

Il byte da scrivere.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

WriteBytes - Byte scritti

Interfaccia:

```
void WriteBytes(Byte[] value);
```

Scrivere un array di byte nel flusso di messaggi di byte.

Parametri:

valore (immissione)

La schiera di byte da scrivere.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

WriteBytes - Scrivi array di byte parziali

Interfaccia:

```
void WriteBytes(Byte[] value, int offset, int length);
```

Scrivere un array parziale di byte nel flusso di messaggi di byte, come definito dalla lunghezza specificata.

Parametri:

valore (immissione)

La schiera di byte da scrivere.

offset (input)

Il punto di partenza per la schiera di byte da scrivere.

lunghezza (input)

Il numero di byte da scrivere.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

WriteChar - Scrivi carattere

Interfaccia:

```
void WriteChar(Char value);
```

Scrivere un carattere nel flusso di messaggi byte come 2 byte, prima byte di ordine elevato.

Parametri:**valore (immissione)**

Il carattere da scrivere.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

WriteDouble - Scrivere il numero a virgola mobile di precisione doppia

Interfaccia:

```
void WriteDouble(Double value);
```

Convertire un numero a virgola mobile di precisione doppia in un numero intero lungo e scrivere il numero intero lungo nel flusso di messaggi di byte come 8 byte, primo byte di ordine elevato.

Parametri:**valore (immissione)**

Il numero a virgola mobile di precisione doppia da scrivere.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

WriteFloat - Numero virgola mobile di scrittura

Interfaccia:

```
void WriteFloat(Single value);
```

Converti un numero a virgola mobile in un intero e scrivi il numero intero nel flusso di messaggi di byte come 4 byte, primo byte di ordine elevato.

Parametri:**valore (immissione)**

Il numero a virgola mobile da scrivere.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

WriteInt - Scrittura numero intero

Interfaccia:

```
void WriteInt(Int32 value);
```

Scrivere un numero intero nel flusso di messaggi di byte come 4 byte, primo byte di ordine superiore.

Parametri:**valore (immissione)**

Il numero intero da scrivere.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

WriteLong - Scrittura numero intero lungo

Interfaccia:

```
void WriteLong(Int64 value);
```

Scrivere un numero intero lungo nel flusso di messaggi di byte come 8 byte, primo byte di ordine superiore.

Parametri:**valore (immissione)**

Il numero intero lungo da scrivere.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

WriteObject - Scrivi oggetto

Interfaccia:

```
void WriteObject(Object value);
```

Scrivere l'oggetto specificato nel flusso di messaggi di byte.

Parametri:

valore (immissione)

L'oggetto da scrivere, che deve essere un riferimento a un tipo primitivo.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

WriteShort - Scrivi numero intero breve

Interfaccia:

```
void WriteShort(Int16 value);
```

Scrivere un intero breve nel flusso di messaggi di byte come 2 byte, prima byte di ordine elevato.

Parametri:

valore (immissione)

Il numero intero breve da scrivere.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

WriteUTF - Scrivi stringa UTF

Interfaccia:

```
void WriteUTF(String value);
```

Scrivere una stringa, codificata in UTF-8, nel flusso di messaggi byte.

Parametri:

valore (immissione)

Un oggetto String che incapsula la stringa da scrivere.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

Proprietà e metodi ereditati

Le seguenti proprietà sono ereditate dall'interfaccia IMessage:

JMSCorrelationID, JMSDeliveryMode, JMSDestination, JMSExpiration, JMSMessageID, JMSPriority, JMSRedelivered, JMSReplyTo, JMSTimestamp, JMSType, Properties

I seguenti metodi sono ereditati dall'interfaccia IMessage:

clearBody, clearProperties, PropertyExists

I seguenti metodi sono ereditati dall'interfaccia IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

IConnesione

Un oggetto Connection rappresenta la connessione attiva dell'applicazione ad un server di messaggi.

Gerarchia di eredità:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IConnection
```

Per un elenco delle XMS proprietà definite di un oggetto Connection, consultare [“Proprietà della connessione”](#) a pagina 180.

Proprietà di .NET

ClientID - Acquisisci e imposta ID client

Interfaccia:

```
String ClientID
{
    get;
    set;
}
```

Richiamare e impostare l'identificativo client per la connessione

L'identificativo del client può essere preconfigurato dall'amministratore in un `ConnectionFactory` o assegnato impostando `ClientID`.

Un identificativo client viene utilizzato solo per supportare sottoscrizioni durevoli nel dominio di pubblicazione / sottoscrizione e viene ignorato nel dominio point - to - point.

Se un'applicazione imposta un identificativo client per una connessione, l'applicazione deve farlo immediatamente dopo la creazione della connessione e prima di eseguire qualsiasi altra operazione sulla connessione. Se l'applicazione tenta di impostare un identificativo client dopo questo punto, la chiamata genera l'eccezione `IllegalState`.

Questa proprietà non è valida per una connessione in tempo reale a un broker.

Eccezioni:

- Eccezione `XMSException`
- Eccezione `IllegalState`
- Eccezione `InvalidClient`

ExceptionListener - Ottieni e imposta listener di eccezioni

Interfaccia:

```
ExceptionListener ExceptionListener
{
    get;
    set;
}
```

Richiamare il listener di eccezione registrato con la connessione e registrare un listener di eccezione con la connessione

Se non viene registrato alcun listener di eccezione con la connessione, il metodo restituisce un valore null. Se un listener di eccezioni è già registrato con la connessione, è possibile annullare la registrazione specificando un valore null invece del listener di eccezioni.

Per ulteriori informazioni sull'utilizzo dei listener delle eccezioni, consultare [“Listener di messaggi ed eccezioni in .NET”](#) a pagina 48.

Eccezioni:

- Eccezione XMSEException

Metadati - ottieni metadati

Interfaccia:

```
IConnectionMetaData MetaData
{
    get;
}
```

Ottieni i metadati per la connessione

Eccezioni:

- Eccezione XMSEException

Metodi

Chiudi - Chiudi connessione

Interfaccia:

```
void Close();
```

Chiudere la connessione

Se un'applicazione tenta di chiudere una connessione già chiusa, la chiamata viene ignorata.

Parametri:

Nessuno

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException

CreateSession - Crea sessione

Interfaccia:

```
ISession CreateSession(Boolean transacted,  
                        AcknowledgeMode acknowledgeMode);
```

Creare una sessione

Parametri:

transazionali (input)

Il valore `True` indica che la sessione viene eseguita. Il valore `False` indica che la sessione non viene eseguita.

Per una connessione in tempo reale a un broker, il valore deve essere `False`.

acknowledgeMode (input)

Indica in che modo vengono riconosciuti i messaggi ricevuti da un'applicazione. Il valore deve essere uno dei seguenti dall'enumeratore `AcknowledgeMode` :

```
AcknowledgeMode.AutoAcknowledge  
AcknowledgeMode.ClientAcknowledge  
AcknowledgeMode.DupsOkAcknowledge
```

Per una connessione in tempo reale ad un broker, il valore deve essere `AcknowledgeMode.AutoAcknowledge` o `AcknowledgeMode.DupsOkAcknowledge`

Questo parametro viene ignorato se viene eseguita la transazione della sessione. Per ulteriori informazioni sulle modalità di riconoscimento, consultare [“Riconoscimento messaggio” a pagina 25](#).

Restituisce:

L'oggetto `Session`

Eccezioni:

- Eccezione `XMSEException`

Avvia - Avvia connessione

Interfaccia:

```
void Start();
```

Avviare o riavviare la consegna dei messaggi in arrivo per la connessione. La chiamata viene ignorata se la connessione è già avviata.

Parametri:

Nessuno

Restituisce:

Nulla

Eccezioni:

- Eccezione `XMSEException`

Arresta - Arresta connessione

Interfaccia:

```
void Stop();
```

Arrestare la consegna dei messaggi in entrata per la connessione. La chiamata viene ignorata se la connessione è già stata arrestata.

Parametri:

Nessuno

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException

Proprietà e metodi ereditati

I seguenti metodi sono ereditati dall'interfaccia `IPropertyContext`:

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

IConnectionFactory

Un'applicazione utilizza una produzione connessioni per creare una connessione.

Gerarchia di eredità:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IConnectionFactory
```

Per un elenco delle XMS proprietà definite di un oggetto `ConnectionFactory`, consultare [“Proprietà di ConnectionFactory”](#) a pagina 181.

Concetti correlati

[ConnectionFactory](#) e oggetti `Connection`

Un oggetto `ConnectionFactory` fornisce un template che un'applicazione utilizza per creare un oggetto `Connection`. L'applicazione utilizza l'oggetto `Connection` per creare un oggetto `Session`.

[Connessione a un SIB \(Service Integration Bus\)](#)

Un'applicazione XMS può connettersi a un SIB (Service Integration Bus) WebSphere Application Server utilizzando una connessione TCP/IP diretta o utilizzando HTTP su TCP/IP.

[Connessioni protette a un gestore code IBM MQ](#)

Per consentire a un'applicazione di XMS .NET di stabilire connessioni sicure a un gestore code IBM MQ, le proprietà pertinenti devono essere definite nell'oggetto `ConnectionFactory`.

[Connessioni sicure a un motore di messaggistica WebSphere Application Server service integration bus](#)

Per abilitare un'applicazione XMS .NET per stabilire connessioni sicure a un motore di messaggistica WebSphere Application Server service integration bus, le proprietà pertinenti devono essere definite nell'oggetto `ConnectionFactory`.

[Associazione di proprietà per gli oggetti gestiti](#)

Per consentire alle applicazioni di utilizzare le definizioni oggetto di destinazione e factory di connessione IBM MQ JMS e WebSphere Application Server, le propriet ... richiamate da tali definizioni devono essere associate alle propriet ... XMS corrispondenti che possono essere impostate sulle destinazioni e sulle produzioni connessioni XMS.

Attività correlate

Creazione di oggetti gestiti

Le definizioni dell'oggetto `ConnectionFactory` e `Destination` richieste dalle applicazioni XMS per stabilire una connessione a un server di messaggistica devono essere create utilizzando gli strumenti di gestione appropriati.

Riferimenti correlati

Proprietà richieste per gli oggetti ConnectionFactory gestiti

Quando un'applicazione crea una factory di connessione, è necessario definire un numero di proprietà per creare una connessione a un server di messaggistica.

Metodi

CreateConnection - Crea factory di connessione (utilizzando l'identità utente predefinita)

Interfaccia:

```
IConnection CreateConnection();
```

Creare una produzione connessioni con le proprietà predefinite.

Se ci si connette a WebSphere MQ e `XMSC_USERID` non è impostato, per impostazione predefinita il gestore code utilizza l' `userID` dell'utente collegato. Se si richiede un'ulteriore autenticazione a livello di connessione di singoli utenti, è possibile scrivere un'uscita di autenticazione client configurata in WebSphere MQ.

Parametri:

Nessuno

Eccezioni:

- Eccezione `XMSEException`

CreateConnection - Crea connessione (utilizzando un'identità utente specificata)

Interfaccia:

```
IConnection CreateConnection(String userId, String password);
```

Creare un collegamento utilizzando un'identità utente specificata.

Se ci si connette a WebSphere MQ e `XMSC_USERID` non è impostato, per impostazione predefinita il gestore code utilizza l' `userID` dell'utente collegato. Se si richiede un'ulteriore autenticazione a livello di connessione di singoli utenti, è possibile scrivere un'uscita di autenticazione client configurata in WebSphere MQ.

La connessione viene creata in modalità arrestata. Nessun messaggio viene consegnato fino a quando l'applicazione non richiama `Connection.start()`.

Parametri:

userID (input)

Un oggetto stringa che incapsula l'identificativo utente da utilizzare per autenticare l'applicazione. Se si fornisce un valore null, viene effettuato un tentativo di creare la connessione senza autenticazione.

password (input)

Un oggetto stringa che incapsula la parola d'ordine da utilizzare per autenticare l'applicazione. Se si fornisce un valore null, viene effettuato un tentativo di creare la connessione senza autenticazione.

Restituisce:

L'oggetto Connection.

Eccezioni:

- Eccezione XMSEException
- XMS_X_SECURITY_EXCEPTION

Proprietà e metodi ereditati

I seguenti metodi sono ereditati dall'interfaccia [IPropertyContext](#):

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

Dati IConnectionMeta

Un oggetto ConnectionMetaData fornisce informazioni su una connessione.

Gerarchia di eredità:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IConnectionMetaData
```

Per un elenco delle XMS proprietà definite di un oggetto dati ConnectionMeta, consultare [“Proprietà dei dati ConnectionMeta”](#) a pagina 187.

Proprietà di .NET

JMSXPropertyNames - Richiama proprietà messaggio definite JMS

Interfaccia:

```
System.Collections.IEnumerator JMSXPropertyNames
{
    get;
}
```

Restituisce un'enumerazione dei nomi delle JMS proprietà del messaggio definite supportate dalla connessione.

Le proprietà del messaggio definite da JMS non sono supportate da una connessione in tempo reale a un broker.

Eccezioni:

- Eccezione XMSEException

Proprietà e metodi ereditati

I seguenti metodi sono ereditati dall'interfaccia [IPropertyContext](#):

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

IDestinazione

Una destinazione è il punto in cui un'applicazione invia messaggi o è un'origine da cui un'applicazione riceve messaggi o entrambi.

Gerarchia di eredità:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IDestination
```

Per un elenco delle XMS proprietà definite di un oggetto Destinazione, vedere [“Proprietà della destinazione”](#) a pagina 188.

Concetti correlati

[ConnectionFactory](#) e oggetti [Connection](#)

Un oggetto [ConnectionFactory](#) fornisce un template che un'applicazione utilizza per creare un oggetto [Connection](#). L'applicazione utilizza l'oggetto [Connection](#) per creare un oggetto [Session](#).

[Connessione a un SIB \(Service Integration Bus\)](#)

Un'applicazione XMS può connettersi a un SIB (Service Integration Bus) WebSphere Application Server utilizzando una connessione TCP/IP diretta o utilizzando HTTP su TCP/IP.

[Destinazioni](#)

Un'applicazione di XMS utilizza un oggetto [Destinazione](#) per specificare la destinazione dei messaggi inviati e l'origine dei messaggi ricevuti.

[Caratteri jolly di destinazione](#)

XMS fornisce supporto per i caratteri jolly di destinazione, garantendo che i caratteri jolly possano essere trasmessi al luogo in cui sono necessari per la corrispondenza. Esiste un diverso schema di caratteri jolly per ciascun tipo di server che XMS può utilizzare.

[Identificativi risorsa uniforme argomento](#)

L'URI (uniform resource identifier) dell'argomento specifica il nome dell'argomento; può anche specificare una o più proprietà per esso.

[Identificativi risorsa uniforme coda](#)

L'URI di una coda specifica il nome della coda; può anche specificare una o più proprietà della coda.

[Destinazioni temporanee](#)

Le applicazioni XMS possono creare e utilizzare destinazioni temporanee.

[Associazione di proprietà per gli oggetti gestiti](#)

Per consentire alle applicazioni di utilizzare le definizioni oggetto di destinazione e factory di connessione IBM MQ JMS e WebSphere Application Server, le propriet ... richiamate da tali definizioni devono essere associate alle propriet ... XMS corrispondenti che possono essere impostate sulle destinazioni e sulle produzioni connessioni XMS.

Attività correlate

[Creazione di oggetti gestiti](#)

Le definizioni dell'oggetto [ConnectionFactory](#) e [Destination](#) richieste dalle applicazioni XMS per stabilire una connessione a un server di messaggistica devono essere create utilizzando gli strumenti di gestione appropriati.

Riferimenti correlati

[Proprietà richieste per gli oggetti Destinazione gestiti](#)

Un'applicazione che sta creando una destinazione deve impostare diverse proprietà dell'applicazione su un oggetto [Destinazione](#) gestito.

Proprietà di .NET

Nome - Ottieni nome destinazione

Interfaccia:

```
String Name
{
    get;
}
```

Ottieni il nome della destinazione Il nome è una stringa che incapsula il nome di una coda o il nome di un argomento.

Eccezioni:

- Eccezione XMSEException

TypeId - Acquisisci tipo di destinazione

Interfaccia:

```
DestinationType TypeId
{
    get;
}
```

Ottieni il tipo di destinazione. Il tipo di destinazione è uno dei seguenti valori:

DestinationType.Queue
DestinationType.Topic

Eccezioni:

- Eccezione XMSEException

Proprietà e metodi ereditati

I seguenti metodi sono ereditati dall'interfaccia [IPropertyContext](#):

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

ExceptionListener

Gerarchia di eredità:

Nessuno

Un'applicazione utilizza un listener di eccezioni per ricevere una notifica asincrona di un problema con una connessione.

Se un'applicazione utilizza una connessione solo per utilizzare i messaggi in modo asincrono e per nessun altro scopo, l'unico modo in cui l'applicazione può acquisire informazioni su un problema con la connessione è utilizzando un listener di eccezioni. In altre situazioni, un listener di eccezioni può fornire un modo più immediato di conoscere un problema con una connessione piuttosto che attendere la successiva chiamata sincrona a XMS.

Delega

Interfaccia:

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

Notifica l'applicazione di un problema con una connessione

I metodi che implementano questo delegato possono essere registrati con il collegamento.

Per ulteriori informazioni sull'utilizzo dei listener delle eccezioni, consultare [“Listener di messaggi ed eccezioni in .NET”](#) a pagina 48.

Parametri:

eccezione (input)

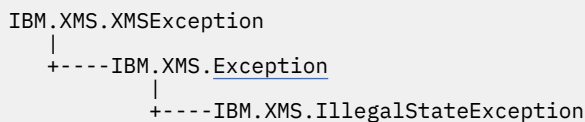
Un puntatore a un'eccezione creata da XMS.

Restituisce:

Nulla

Eccezione **IllegalState**

Gerarchia di eredità:



XMS genera questa eccezione se un'applicazione richiama un metodo in un momento non corretto o inappropriato o se XMS non si trova in uno stato appropriato per l'operazione richiesta.

Proprietà e metodi ereditati

I seguenti metodi sono ereditati dall'interfaccia [XMSEException](#):

[GetErrorCode](#), [GetLinkedException](#)

InitialContext

Un'applicazione utilizza un oggetto InitialContext per creare oggetti dalle definizioni di oggetto richiamati da un repository di oggetti gestiti.

Gerarchia di eredità:

Nessuno

Concetti correlati

[Proprietà InitialContext](#)

I parametri del costruttore InitialContext includono l'ubicazione del repository di oggetti gestiti, forniti come URI (uniform resource indicator). Affinché un'applicazione stabilisca una connessione al repository, potrebbe essere necessario fornire più informazioni rispetto a quelle contenute nell'URI.

[Formato URI per contesti iniziali XMS](#)

L'ubicazione del repository di oggetti gestiti viene fornita come URI (uniform resource indicator). Il formato dell'URI dipende dal tipo di contesto.

[Richiamo di oggetti gestiti](#)

XMS richiama un oggetto gestito dal repository utilizzando l'indirizzo fornito quando viene creato l'oggetto InitialContext o nelle proprietà InitialContext .

Attività correlate

[Oggetti InitialContext](#)

Un'applicazione deve creare un contesto iniziale da utilizzare per stabilire una connessione al repository di oggetti gestiti per richiamare gli oggetti gestiti richiesti.

Proprietà di .NET

Ambiente - Ottieni l'ambiente

Interfaccia:

```
Hashtable Environment
{
    get;
}
```

Ottieni l'ambiente.

Eccezioni:

- Le eccezioni sono specifiche per il servizio di directory utilizzato.

Costruttori

InitialContext - Crea contesto iniziale

Interfaccia:

```
InitialContext(Hashtable env);
```

Creare un oggetto InitialContext .

Parametri:

Le informazioni richieste per stabilire una connessione al repository di oggetti gestiti vengono fornite al costruttore in un ambiente Hashtable.

Eccezioni:

- Eccezione XMSEException

Metodi

Ambiente AddTo- Aggiungi una nuova proprietà all'ambiente

Interfaccia:

```
Object AddToEnvironment(String propName, Object propVal);
```

Aggiungere una nuova proprietà all'ambiente.

Parametri:

propName (input)

Un oggetto String che incapsula il nome della proprietà da aggiungere.

propVal (input)

Il valore della proprietà da aggiungere.

Restituisce:

Il vecchio valore della proprietà.

Eccezioni:

- Le eccezioni sono specifiche per il servizio di directory utilizzato.

Chiudi - Chiudi questo contesto

Interfaccia:

```
void Close()
```

Chiudi questo contesto.

Parametri:

Nessuno

Restituisce:

Nessuno

Eccezioni:

- Le eccezioni sono specifiche per il servizio di directory utilizzato.

Ricerca - Ricerca oggetto nel contesto iniziale

Interfaccia:

```
Object Lookup(String name);
```

Creare un oggetto da una definizione oggetto richiamata dal repository di oggetti gestiti.

Parametri:

nome (input)

Un oggetto stringa che incapsula il nome dell'oggetto gestito da richiamare. Il nome può essere un nome semplice o un nome complesso. Per ulteriori dettagli, fare riferimento a [“Richiamo di oggetti gestiti”](#) a pagina 65.

Restituisce:

Un IConnectionFactory o un IDestination, a seconda del tipo di oggetto richiamato. Se la funzione può accedere all'indirizzario, ma non è in grado di trovare l'oggetto richiesto, viene restituito un valore null.

Eccezioni:

- Le eccezioni sono specifiche per il servizio di directory utilizzato.

Ambiente RemoveFrom- Rimuove una proprietà dall'ambiente

Interfaccia:

```
Object RemoveFromEnvironment(String propName);
```

Rimuovere una proprietà dall'ambiente.

Parametri:

propName (input)

Un oggetto stringa che racchiude il nome della proprietà da eliminare.

Restituisce:

L'oggetto che è stato rimosso.

Eccezioni:

- Le eccezioni sono specifiche per il servizio di directory utilizzato.

IDException InvalidClient

Gerarchia di eredità:

```
IBM.XMS.XMSException
```

```
+----IBM.XMS.XMSEnception
      |
      +----IBM.XMS.InvalidClientIDException
```

XMS genera questa eccezione se un'applicazione tenta di impostare un identificatore client per una connessione, ma l'identificativo client non è valido o è già in uso.

Proprietà e metodi ereditati

I seguenti metodi sono ereditati dall'interfaccia [XMSEnception](#):

[GetErrorCode](#), [GetLinkedException](#)

Eccezione InvalidDestination

Gerarchia di eredità:

```
IBM.XMS.XMSEnception
|
+----IBM.XMS.XMSEnception
      |
      +----IBM.XMS.InvalidDestinationException
```

XMS genera questa eccezione se un'applicazione specifica una destinazione non valida.

Proprietà e metodi ereditati

I seguenti metodi sono ereditati dall'interfaccia [XMSEnception](#):

[GetErrorCode](#), [GetLinkedException](#)

Eccezione InvalidSelector

Gerarchia di eredità:

```
IBM.XMS.XMSEnception
|
+----IBM.XMS.XMSEnception
      |
      +----IBM.XMS.InvalidSelectorException
```

XMS genera questa eccezione se un'applicazione fornisce un'espressione del selettore di messaggi la cui sintassi non è valida.

Proprietà e metodi ereditati

I seguenti metodi sono ereditati dall'interfaccia [XMSEnception](#):

[GetErrorCode](#), [GetLinkedException](#)

IMapMessage

Un messaggio della mappa è un messaggio il cui corpo comprende una serie di coppie nome - valore, in cui ogni valore ha un tipo di dati associato.

Gerarchia di eredità:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IMessage
      |
      +----IBM.XMS.IMapMessage
```

Quando un'applicazione ottiene il valore della coppia nome - valore, il valore può essere convertito da XMS in un altro tipo di dati. Per ulteriori informazioni su questa forma di conversione implicita, consultare [“Associa messaggi” a pagina 77](#).

Riferimenti correlati

Associa messaggi

Il corpo di un messaggio di associazione contiene una serie di coppie nome - valore, in cui ogni valore ha un tipo di dati associato.

Proprietà di .NET

MapNames - Richiama nomi mappa

Interfaccia:

```
System.Collections.IEnumerator MapNames
{
    get;
}
```

Ottieni un'enumerazione dei nomi nel contenuto del messaggio della mappa.

Eccezioni:

- Eccezione XMSEException

Metodi

GetBoolean - Richiama valore booleano

Interfaccia:

```
Boolean GetBoolean(String name);
```

Richiamare il valore booleano identificato dal nome dal contenuto del messaggio della mappa.

Parametri:

nome (input)

Un oggetto String che incapsula il nome che identifica il valore booleano.

Restituisce:

Il valore booleano richiamato dal contenuto del messaggio di associazione.

Eccezioni:

- Eccezione XMSEException

GetByte - Acquisisci byte

Interfaccia:

```
Byte    GetByte(String name);
Int16   GetSignedByte(String name);
```

Richiamare il byte identificato dal nome dal corpo del messaggio della mappa.

Parametri:

nome (input)

Un oggetto stringa che incapsula il nome che identifica il byte.

Restituisce:

Il byte richiamato dal corpo del messaggio di mappa. Nessuna conversione dati viene eseguita sul byte.

Eccezioni:

- Eccezione XMSEException

GetBytes - Richiama byte

Interfaccia:

```
Byte[] GetBytes(String name);
```

Richiamare l'array di byte identificati dal nome dal contenuto del messaggio di associazione.

Parametri:**nome (input)**

Un oggetto stringa che incapsula il nome che identifica l'array di byte.

Restituisce:

Il numero di byte nell'array.

Eccezioni:

- Eccezione XMSEException

GetChar - Acquisisci carattere

Interfaccia:

```
Char GetChar(String name);
```

Richiamare il carattere identificato dal nome dal corpo del messaggio della mappa.

Parametri:**nome (input)**

Un oggetto String che incapsula il nome che identifica il carattere.

Restituisce:

Il carattere richiamato dal contenuto del messaggio di associazione.

Eccezioni:

- Eccezione XMSEException

GetDouble - Acquisisci numero a virgola mobile di precisione doppia

Interfaccia:

```
Double GetDouble(String name);
```

Richiamare il numero a virgola mobile a doppia precisione identificato dal nome dal contenuto del messaggio della mappa.

Parametri:**nome (input)**

Un oggetto String che incapsula il nome che identifica il numero a virgola mobile a doppia precisione.

Restituisce:

Il numero a virgola mobile di precisione doppia richiamato dal corpo del messaggio della mappa.

Eccezioni:

- Eccezione XMSEException

GetFloat - Acquisisci numero a virgola mobile

Interfaccia:

```
Single GetFloat(String name);
```

Richiamare il numero a virgola mobile identificato per nome dal corpo del messaggio della mappa.

Parametri:**nome (input)**

Un oggetto String che incapsula il nome che identifica il numero a virgola mobile.

Restituisce:

Il numero a virgola mobile richiamato dal corpo del messaggio della mappa.

Eccezioni:

- Eccezione XMSEException

GetInt - Richiamo numero intero

Interfaccia:

```
Int32 GetInt(String name);
```

Richiamare il valore intero identificato dal nome dal corpo del messaggio della mappa.

Parametri:**nome (input)**

Un oggetto stringa che incapsula il nome che identifica il numero intero.

Restituisce:

Il numero intero richiamato dal contenuto del messaggio di associazione.

Eccezioni:

- Eccezione XMSEException

GetLong - Richiamo numero intero lungo

Interfaccia:

```
Int64 GetLong(String name);
```

Richiamare il numero intero lungo identificato dal nome dal corpo del messaggio della mappa.

Parametri:**nome (input)**

Un oggetto stringa che incapsula il nome che identifica il numero intero lungo.

Restituisce:

Il numero intero lungo richiamato dal contenuto del messaggio della mappa.

Eccezioni:

- Eccezione XMSEException

GetObject - Acquisisci oggetto

Interfaccia:

```
Object GetObject(String name);
```

Richiamare un riferimento al valore di una coppia nome - valore, dal contenuto del messaggio della mappa. La coppia nome - valore è identificata dal nome.

Parametri:**nome (input)**

Un oggetto String che incapsula il nome della coppia nome - valore.

Restituisce:

Il valore, che è uno dei seguenti tipi di oggetto:

Boolean
Byte
Byte[]
Char
Double
Single
Int32
Int64
Int16
String

Eccezioni:

Eccezione XMSEException

GetShort - Richiama numero intero breve

Interfaccia:

```
Int16 GetShort(String name);
```

Richiamare il numero intero breve identificato per nome dal corpo del messaggio della mappa.

Parametri:**nome (input)**

Un oggetto String che incapsula il nome che identifica il numero intero breve.

Restituisce:

Il numero intero breve richiamato dal corpo del messaggio di associazione.

Eccezioni:

- Eccezione XMSEException

GetString - Richiama stringa

Interfaccia:

```
String GetString(String name);
```

Ottenere la stringa identificata dal nome dal corpo del messaggio della mappa.

Parametri:

nome (input)

Un oggetto stringa che racchiude il nome che identifica la stringa nel contenuto del messaggio della mappa.

Restituisce:

Un oggetto stringa che incapsula la stringa richiamata dal contenuto del messaggio della mappa. Se è richiesta la conversione dei dati, questo valore è la stringa dopo la conversione.

Eccezioni:

- Eccezione XMSEException

ItemExists - Verifica che la coppia nome - valore esista

Interfaccia:

```
Boolean ItemExists(String name);
```

Controllare se il corpo del messaggio della mappa contiene una coppia nome - valore con il nome specificato.

Parametri:

nome (input)

Un oggetto String che incapsula il nome della coppia nome - valore.

Restituisce:

- True, se il contenuto del messaggio della mappa contiene una coppia nome - valore con il nome specificato.
- False, se il corpo del messaggio della mappa non contiene una coppia nome - valore con il nome specificato.

Eccezioni:

- Eccezione XMSEException

SetBoolean - Imposta valore booleano

Interfaccia:

```
void SetBoolean(String name, Boolean value);
```

Impostare un valore booleano nel corpo del messaggio della mappa.

Parametri:

nome (input)

Un oggetto stringa che incapsula il nome per identificare il valore booleano nel contenuto del messaggio della mappa.

valore (immissione)

Il valore booleano da impostare.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException

SetByte - Imposta byte

Interfaccia:

```
void SetByte(String name, Byte value);  
void SetSignedByte(String name, Int16 value);
```

Impostare un byte nel contenuto del messaggio di mappa.

Parametri:**nome (input)**

Un oggetto String che incapsula il nome per identificare il byte nel contenuto del messaggio di mappa.

valore (immissione)

Il byte da impostare.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException

SetBytes - imposta byte

Interfaccia:

```
void SetBytes(String name, Byte[] value);
```

Impostare un array di byte nel contenuto del messaggio di mappa.

Parametri:**nome (input)**

Un oggetto stringa che incapsula il nome per identificare l'array di byte nel contenuto del messaggio di mappa.

valore (immissione)

L'array di byte da impostare.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException

SetChar - Imposta carattere

Interfaccia:

```
void SetChar(String name, Char value);
```

Impostare un carattere a 2 byte nel contenuto del messaggio di mappa.

Parametri:

nome (input)

Un oggetto stringa che incapsula il nome per identificare il carattere nel contenuto del messaggio della mappa.

valore (immissione)

Il carattere da impostare.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException

SetDouble - Imposta numero a virgola mobile di precisione doppia

Interfaccia:

```
void SetDouble(String name, Double value);
```

Impostare un numero a virgola mobile a precisione doppia nel corpo del messaggio della mappa.

Parametri:

nome (input)

Un oggetto String che incapsula il nome per identificare il numero a virgola mobile di precisione doppia nel contenuto del messaggio della mappa.

valore (immissione)

Il numero a virgola mobile di precisione doppia da impostare.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException

SetFloat - Imposta numero a virgola mobile

Interfaccia:

```
void SetFloat(String name, Single value);
```

Impostare un numero a virgola mobile nel corpo del messaggio della mappa.

Parametri:

nome (input)

Un oggetto String che incapsula il nome per identificare il numero a virgola mobile nel corpo del messaggio di mappa.

valore (immissione)

Il numero a virgola mobile da impostare.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException

SetInt - Imposta numero intero

Interfaccia:

```
void SetInt(String name, Int32 value);
```

Impostare un numero intero nel contenuto del messaggio di associazione.

Parametri:**nome (input)**

Un oggetto stringa che incapsula il nome per identificare il intero nel corpo del messaggio della mappa.

valore (immissione)

Il numero intero da impostare.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException

SetLong - Imposta numero intero lungo

Interfaccia:

```
void SetLong(String name, Int64 value);
```

Impostare un numero intero lungo nel corpo del messaggio di mappa.

Parametri:**nome (input)**

Un oggetto String che incapsula il nome per identificare il numero intero lungo nel contenuto del messaggio della mappa.

valore (immissione)

Il numero intero lungo da impostare.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException

SetObject - Imposta oggetto

Interfaccia:

```
void SetObject(String name, Object value);
```

Impostare un valore, che deve essere un tipo primitivo XMS , nel contenuto del messaggio della mappa.

Parametri:**nome (input)**

Un oggetto stringa che incapsula il nome per identificare il valore nel contenuto del messaggio della mappa.

valore (immissione)

Un array di byte contenente il valore da impostare.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException

SetShort - Imposta numero intero breve

Interfaccia:

```
void SetShort(String name, Int16 value);
```

Impostare un numero intero breve nel corpo del messaggio di mappa.

Parametri:**nome (input)**

Un oggetto String che incapsula il nome per identificare il numero intero breve nel contenuto del messaggio della mappa.

valore (immissione)

Il numero intero breve da impostare.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException

SetString - Imposta stringa

Interfaccia:

```
void SetString(String name, String value);
```

Impostare una stringa nel contenuto del messaggio della mappa.

Parametri:**nome (input)**

Un oggetto stringa che racchiude il nome per identificare la stringa nel contenuto del messaggio della mappa.

valore (immissione)

Un oggetto String che incapsula la stringa da impostare.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException

Proprietà e metodi ereditati

Le seguenti proprietà sono ereditate dall'interfaccia [IMessage](#):

[JMSCorrelationID](#), [JMSDeliveryMode](#), [JMSDestination](#), [JMSExpiration](#), [JMSMessageID](#), [JMSPriority](#), [JMSRedelivered](#), [JMSReplyTo](#), [JMSTimestamp](#), [JMSType](#), [Properties](#)

I seguenti metodi sono ereditati dall'interfaccia [IMessage](#):

[clearBody](#), [clearProperties](#), [PropertyExists](#)

I seguenti metodi sono ereditati dall'interfaccia [IPropertyContext](#):

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

IMessaggio

Un oggetto Message rappresenta un messaggio che un'applicazione invia o riceve. IMessage è una superclasse per le classi di messaggi come [IMapMessage](#).

Gerarchia di eredità:

```
IBM.XMS.IPropertyContext
|
+---- IBM.XMS.IMessage
```

Per un elenco dei campi di intestazione del messaggio JMS in un oggetto Messaggio, consultare [“Campi di intestazione in un messaggio XMS”](#) a pagina 70. Per un elenco delle proprietà definite JMS di un oggetto Messaggio, consultare [“Proprietà definite da JMS di un messaggio”](#) a pagina 72. Per un elenco delle IBM proprietà definite di un oggetto Message, consultare [“Proprietà definite da IBM di un messaggio”](#) a pagina 73. Per un elenco di proprietà JMS_IBM_MQMD* per l'oggetto Messaggio, consultare [“Proprietà JMS_IBM_MQMD*”](#) a pagina 193

I messaggi vengono eliminati dal raccoglitore dati inutilizzati. Quando un messaggio viene eliminato, questo libera le risorse che stava utilizzando.

Riferimenti correlati

[XMS messaggi](#)

Questa sezione descrive la struttura e il contenuto dei messaggi XMS e spiega come le applicazioni elaborano i messaggi XMS .

Proprietà di .NET

GetJMSCorrelationID - Acquisisci e imposta JMSCorrelationID

Interfaccia:

```
String JMSCorrelationID
{
    get;
    set;
}
```

Richiamare e impostare l'identificatore di correlazione del messaggio come oggetto stringa.

Eccezioni:

- Eccezione [XMSEException](#)

JMSDeliveryMode - Ottieni e imposta JMSDeliveryMode

Interfaccia:

```
DeliveryMode JMSDeliveryMode
{
    get;
    set;
}
```

Richiamare e impostare la modalità di consegna del messaggio.

La modalità di consegna del messaggio è uno dei seguenti valori:

```
DeliveryMode.Persistent
DeliveryMode.NonPersistent
```

Per un messaggio appena creato che non è stato inviato, la modalità di consegna è `DeliveryMode.Persistent`, tranne che per una connessione in tempo reale a un broker per cui la modalità di consegna è `DeliveryMode.NonPersistent`. Per un messaggio ricevuto, il metodo restituisce la modalità di consegna impostata dalla chiamata `IMessageProducer.send ()` quando il messaggio è stato inviato a meno che l'applicazione ricevente non modifichi la modalità di consegna impostando `JMSDeliveryMode`.

Eccezioni:

- Eccezione `XMSEException`

JMSDestination - Richiama e imposta JMSDestination

Interfaccia:

```
IDestination JMSDestination
{
    get;
    set;
}
```

Richiamare e impostare la destinazione del messaggio.

La destinazione viene impostata dalla chiamata `IMessageProducer.send ()` quando il messaggio viene inviato. Il valore di `JMSDestination` viene ignorato. Tuttavia, è possibile utilizzare `JMSDestination` per modificare la destinazione di un messaggio ricevuto.

Per un messaggio appena creato che non è stato inviato, il metodo restituisce un oggetto `Destinazione` null, a meno che l'applicazione di invio non imposti una destinazione impostando `JMSDestination`. Per un messaggio ricevuto, il metodo restituisce un oggetto `Destinazione` per la destinazione impostata dalla chiamata `IMessageProducer.send ()` quando il messaggio è stato inviato a meno che l'applicazione ricevente non modifichi la destinazione impostando `JMSDestination`.

Eccezioni:

- Eccezione `XMSEException`

JMSExpiration - Richiama e imposta JMSExpiration

Interfaccia:

```
Int64 JMSExpiration
{
    get;
```

```
    set;  
}
```

Richiamare e impostare l'ora di scadenza del messaggio.

L'ora di scadenza viene impostata dalla chiamata `IMessageProducer.send ()` quando il messaggio viene inviato. Il suo valore viene calcolato aggiungendo la durata, come specificato dall'applicazione mittente, all'ora in cui viene inviato il messaggio. L'ora di scadenza è espressa in millisecondi a partire dalle 00:00 GMT del 1 ° gennaio 1970.

Per un messaggio appena creato che non è stato inviato, l'ora di scadenza è 0 a meno che l'applicazione mittente non imposti un'ora di scadenza differente impostando `JMSEExpiration`. Per un messaggio ricevuto, il metodo restituisce l'ora di scadenza impostata dalla chiamata `IMessageProducer.send ()` quando il messaggio è stato inviato, a meno che l'applicazione ricevente non modifichi l'ora di scadenza impostando `JMSEExpiration`.

Se la durata è 0, la chiamata `IMessageProducer.send ()` imposta la scadenza su 0 per indicare che il messaggio non scade.

XMS elimina i messaggi scaduti e non li consegna alle applicazioni.

Eccezioni:

- Eccezione `XMSEException`

JMSMessageID - Richiama e imposta JMSMessageID

Interfaccia:

```
String JMSMessageID  
{  
    get;  
    set;  
}
```

Richiamare e impostare l'identificativo del messaggio come un oggetto stringa che incapsula l'identificativo del messaggio.

L'identificativo del messaggi viene impostato dalla chiamata `IMessageProducer.send ()` quando il messaggio viene inviato. Per un messaggio ricevuto, il metodo restituisce l'identificativo del messaggio impostato dalla chiamata `IMessageProducer.send ()` quando il messaggio è stato inviato, a meno che l'applicazione ricevente non modifichi l'identificativo del messaggio impostando `JMSMessageID`.

Se il messaggio non ha un identificativo di messaggio, il metodo restituisce un valore null.

Eccezioni:

- Eccezione `XMSEException`

JMSPriority - Richiama e imposta JMSPriority

Interfaccia:

```
Int32 JMSPriority  
{  
    get;  
    set;  
}
```

Richiamare e impostare la priorità del messaggio.

La priorità viene impostata dalla chiamata `IMessageProducer.send ()` quando il messaggio viene inviato. Il valore è un numero intero compreso nell'intervallo 0, la priorità più bassa, in 9, la priorità più alta.

Per un messaggio appena creato che non è stato inviato, la priorità è 4 a meno che l'applicazione mittente non imposti una priorità differente impostando `JMSPriority`. Per un messaggio ricevuto, il metodo restituisce la priorità impostata dalla chiamata `IMessageProducer.send ()` quando il messaggio è stato inviato, a meno che l'applicazione ricevente non modifichi la priorità impostando `JMSPriority`.

Eccezioni:

- Eccezione `XMSEException`

JMSReformita - Richiama e imposta JMSReconsegnata

Interfaccia:

```
Boolean JMSRedelivered
{
    get;
    set;
}
```

Ottenere un'indicazione se il messaggio è in fase di riconsegnamento e indicare se il messaggio è in fase di riconsegnamento. L'indicazione viene impostata dalla chiamata `IMessageConsumer.receive ()` quando viene ricevuto il messaggio.

Questa proprietà ha i seguenti valori:

- `True`, se il messaggio viene riconsegnato.
- `False`, se il messaggio non viene riconsegnato.

Per una connessione in tempo reale a un broker, il valore è sempre `False`.

Un'indicazione di riconsegna impostata da `JMSResent` prima dell'invio del messaggio viene ignorata dalla chiamata `IMessageProducer.send ()` quando il messaggio viene inviato e viene ignorata e sostituita dalla chiamata `IMessageConsumer.receive ()` quando il messaggio viene ricevuto. Tuttavia, è possibile utilizzare `JMSRenei` per modificare l'indicazione per un messaggio ricevuto.

Eccezioni:

- Eccezione `XMSEException`

JMSReplyTo - Get e Set JMSReplyTo

Interfaccia:

```
IDestination JMSReplyTo
{
    get;
    set;
}
```

Richiamare e impostare la destinazione in cui deve essere inviata una risposta al messaggio.

Il valore di questa proprietà è un oggetto Destinazione per la destinazione a cui deve essere inviata una risposta al messaggio. Un oggetto di destinazione null indica che non è prevista alcuna risposta.

Eccezioni:

- Eccezione `XMSEException`

JMSTimestamp - Richiamo e impostazione di JMSTimestamp

Interfaccia:

```
Int64 JMSTimestamp
{
    get;
    set;
}
```

Richiamare e impostare l'ora in cui è stato inviato il messaggio.

La data/ora viene impostata dalla chiamata `IMessageProducer.send ()` quando il messaggio viene inviato ed è espressa in millisecondi a partire dalle 00:00:00 GMT del 1 ° gennaio 1970.

Per un messaggio appena creato che non è stato inviato, la data / ora è 0 a meno che l'applicazione mittente non imposti una data / ora differente impostando `JMSTimestamp`. Per un messaggio che è stato ricevuto, il metodo restituisce la data / ora impostata dalla chiamata `IMessageProducer.send ()` quando il messaggio è stato inviato, a meno che l'applicazione ricevente non modifichi la data / ora impostando `JMSTimestamp`.

Eccezioni:

- Eccezione `XMSEException`

Note:

1. Se la data / ora non è definita, il metodo restituisce 0 ma non genera alcuna eccezione.

JMSType - Richiama e imposta JMSType

Interfaccia:

```
String JMSType
{
    get;
    set;
}
```

Richiamare e impostare il tipo di messaggio.

Il valore di `JMSType` è una stringa che incapsula il tipo del messaggio. Se è richiesta la conversione dei dati, questo valore è il tipo dopo la conversione.

Eccezioni:

- Eccezione `XMSEException`

PropertyNames - Richiama proprietà

Interfaccia:

```
System.Collections.IEnumerator PropertyNames
{
    get;
}
```

Richiamare un'enumerazione delle proprietà dei nomi del messaggio.

Eccezioni:

- Eccezione `XMSEException`

Metodi

Riconoscimento - Riconosci

Interfaccia:

```
void Acknowledge();
```

Riconoscere questo messaggio e tutti i messaggi precedentemente non riconosciuti ricevuti dalla sessione.

Un'applicazione può richiamare questo metodo se la modalità di riconoscimento della sessione è AcknowledgeMode.ClientAcknowledge. Le chiamate al metodo vengono ignorate se la sessione ha un'altra modalità di riconoscimento o se viene eseguita la transazione.

I messaggi ricevuti ma non riconosciuti potrebbero essere riconsegnati.

Per ulteriori informazioni sul riconoscimento dei messaggi, consultare [“Riconoscimento messaggio” a pagina 25](#).

Parametri:

Nessuno

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException
- Eccezione IllegalState

ClearBody - Cancella body

Interfaccia:

```
void ClearBody();
```

Cancella il corpo del messaggio. I campi di intestazione e le proprietà del messaggio non vengono cancellati.

Se un'applicazione cancella il contenuto di un messaggio, il corpo rimane nello stesso stato di un corpo vuoto in un messaggio appena creato. Lo stato di un corpo vuoto in un messaggio appena creato dipende dal tipo di corpo del messaggio. Per ulteriori informazioni, fare riferimento a [“Il contenuto di un messaggio XMS” a pagina 75](#).

Un'applicazione può cancellare il contenuto di un messaggio in qualsiasi momento, indipendentemente dallo stato in cui si trova il corpo. Se il corpo di un messaggio è di sola lettura, l'unico modo in cui un'applicazione può scrivere nel corpo è che l'applicazione cancella prima il corpo.

Parametri:

Nessuno

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException

ClearProperties - Cancella proprietà

Interfaccia:

```
void ClearProperties();
```

Cancella le proprietà del messaggio. I campi di intestazione e il corpo del messaggio non vengono cancellati.

Se un'applicazione cancella le proprietà di un messaggio, le proprietà diventano leggibili e scrivibili.

Un'applicazione può cancellare le proprietà di un messaggio in qualsiasi momento, indipendentemente dallo stato in cui si trovano le proprietà. Se le proprietà di un messaggio sono di sola lettura, l'unico modo in cui le proprietà possono diventare scrivibili è che l'applicazione cancella prima le proprietà.

Parametri:

Nessuno

Restituisce:

Nulla

Eccezioni:

- Eccezione `XMSEException`

PropertyExists - Verifica proprietà esistente

Interfaccia:

```
Boolean PropertyExists(String propertyName);
```

Controllare se il messaggio ha una proprietà con il nome specificato.

Parametri:

propertyName (input)

Un oggetto `String` che incapsula il nome della proprietà.

Restituisce:

- `True`, se il messaggio ha una proprietà con il nome specificato.
- `False`, se il messaggio non ha una proprietà con il nome specificato.

Eccezioni:

- Eccezione `XMSEException`

Proprietà e metodi ereditati

I seguenti metodi sono ereditati dall'interfaccia [IPropertyContext](#):

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

IMessageConsumer

Un'applicazione utilizza un consumatore di messaggi per ricevere i messaggi inviati a una destinazione.

Gerarchia di eredità:

```
IBM.XMS.IPropertyContext
|
+---- IBM.XMS.IMessageConsumer
```

Per un elenco delle proprietà XMS definite di un oggetto MessageConsumer , consultare [“Proprietà di MessageConsumer”](#) a pagina 197.

Proprietà di .NET

MessageListener - Richiama e imposta listener messaggi

Interfaccia:

```
MessageListener MessageListener
{
    get;
    set;
}
```

Richiamare il listener dei messaggi registrato con il consumer dei messaggi e registrare un listener dei messaggi con tale consumer.

Se nessun listener di messaggi è registrato con il consumatore di messaggi, MessageListener è null. Se un listener di messaggi è già registrato con il consumatore di messaggi, è possibile annullare la registrazione specificando invece un valore null.

Per ulteriori informazioni sull'utilizzo dei listener di messaggi, consultare [“Listener di messaggi ed eccezioni in .NET”](#) a pagina 48.

Eccezioni:

- Eccezione XMSEException

MessageSelector - Richiama selettore messaggi

Interfaccia:

```
String MessageSelector
{
    get;
}
```

Richiamare il selettore del messaggio per il destinatario del messaggio. Il valore di ritorno è un oggetto String che incapsula l'espressione del selettore messaggi. Se è richiesta la conversione dei dati, questo valore è l'espressione del selettore di messaggi dopo la conversione. Se il consumer del messaggio non dispone di un selettore di messaggi, il valore di MessageSelector è un oggetto String null.

Eccezioni:

- Eccezione XMSEException

Metodi

Chiudi - Chiudi consumer messaggi

Interfaccia:

```
void Close();
```

Chiudere il destinatario del messaggio

Se un'applicazione tenta di chiudere un consumatore di messaggi già chiuso, la chiamata viene ignorata.

Parametri:

Nessuno

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException

Ricezione - Ricezione

Interfaccia:

```
IMessage Receive();
```

Ricevere il messaggio successivo per il destinatario del messaggio. La chiamata attende indefinitamente un messaggio o fino a quando il destinatario del messaggio non viene chiuso.

Parametri:

Nessuno

Restituisce:

Un puntatore all'oggetto Messaggio. Se il destinatario del messaggio viene chiuso mentre la chiamata è in attesa di un messaggio, il metodo restituisce un puntatore a un oggetto Messaggio null.

Eccezioni:

- Eccezione XMSEException

Ricezione - Ricezione (con intervallo di attesa)

Interfaccia:

```
IMessage Receive(Int64 delay);
```

Ricevere il messaggio successivo per il destinatario del messaggio. La chiamata attende solo un periodo specificato per un messaggio o fino a quando il destinatario del messaggio non viene chiuso.

Parametri:

ritardo (input)

Il tempo, in millisecondi, in cui la chiamata attende un messaggio. Se si specifica un intervallo di attesa di 0, la chiamata attende indefinitamente un messaggio.

Restituisce:

Un puntatore all'oggetto Messaggio. Se non arriva alcun messaggio durante l'intervallo di attesa o se l'utente del messaggio è chiuso mentre la chiamata è in attesa di un messaggio, il metodo restituisce un puntatore a un oggetto Messaggio null ma non genera alcuna eccezione.

Eccezioni:

- Eccezione XMSEException

ReceiveNoAttesa - Ricezione senza attesa

Interfaccia:

```
IMessage ReceiveNoWait();
```

Ricevere il messaggio successivo per il destinatario del messaggio, se disponibile immediatamente.

Parametri:

Nessuno

Restituisce:

Un puntatore ad un oggetto `Message`. Se nessun messaggio è disponibile immediatamente, il metodo restituisce un puntatore a un oggetto `Message` null.

Eccezioni:

- Eccezione `XMSEException`

Proprietà e metodi ereditati

I seguenti metodi sono ereditati dall'interfaccia `IPropertyContext`:

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

MessageEOFException

Gerarchia di eredità:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.MessageEOFException
```

XMS genera questa eccezione se XMS rileva la fine di un flusso di messaggi di byte quando un'applicazione sta leggendo il contenuto di un messaggio di byte.

Proprietà e metodi ereditati

I seguenti metodi sono ereditati dall'interfaccia `XMSEException`:

[GetErrorCode](#), [GetLinkedException](#)

Eccezione MessageFormatException

Gerarchia di eredità:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.MessageFormatException
```

XMS genera questa eccezione se XMS rileva un messaggio con un formato non valido.

Proprietà e metodi ereditati

I seguenti metodi sono ereditati dall'interfaccia [XMSEException](#):

[GetErrorCode](#), [GetLinkedException](#)

IMessageListener (delegato)

Gerarchia di eredità:

Nessuno

Un'applicazione utilizza un listener di messaggi per ricevere i messaggi asincrono.

Delega

MessageListener - Listener messaggi

Interfaccia:

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

Consegnare un messaggio in modo asincrono al destinatario del messaggio

I metodi che implementano questo delegato possono essere registrati con il collegamento.

Per ulteriori informazioni sull'utilizzo dei listener di messaggi, consultare [“Listener di messaggi ed eccezioni in .NET”](#) a pagina 48.

Parametri:

mesg (input)

L'oggetto Messaggio.

Restituisce:

Null

MessageNotReadableException

Gerarchia di eredità:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.MessageNotReadableException
```

XMS genera questa eccezione se un'applicazione tenta di leggere il corpo di un messaggio di sola scrittura.

Proprietà e metodi ereditati

I seguenti metodi sono ereditati dall'interfaccia [XMSEException](#):

[GetErrorCode](#), [GetLinkedException](#)

MessageNotWritableException

Gerarchia di eredità:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.MessageNotWritableException
```

XMS genera questa eccezione se un'applicazione tenta di scrivere nel corpo di un messaggio di sola lettura.

Proprietà e metodi ereditati

I seguenti metodi sono ereditati dall'interfaccia [XMSException](#):

[GetErrorCode](#), [GetLinkedException](#)

IMessageProducer

Un'applicazione utilizza un produttore di messaggi per inviare messaggi a una destinazione.

Gerarchia di eredità:

```
IBM.XMS.IPropertyContext
|
+---- IBM.XMS.IMessageProducer
```

Per un elenco delle XMS proprietà definite di un oggetto MessageProducer , consultare [“Proprietà di MessageProducer”](#) a pagina 197.

Proprietà di .NET

DeliveryMode - Richiama e imposta la modalità di consegna predefinita

Interfaccia:

```
DeliveryMode DeliveryMode
{
    get;
    set;
}
```

Richiamare e impostare il modo di consegna predefinito per i messaggi inviati dal produttore del messaggio.

La modalità di consegna predefinita ha uno dei seguenti valori:

`DeliveryMode.Persistent`
`DeliveryMode.NonPersistent`

Per una connessione in tempo reale a un broker, il valore deve essere `DeliveryMode.NonPersistent`.

Il valore predefinito è `DeliveryMode.Persistent`, tranne per una connessione in tempo reale a un broker per cui il valore predefinito è `DeliveryMode.NonPersistent`.

Eccezioni:

- Eccezione `XMSException`

Destinazione - Ottieni destinazione

Interfaccia:

```
IDestination Destination
{
    get;
}
```

Ottieni la destinazione per il produttore del messaggio.

Parametri:

Nessuno

Restituisce:

L'oggetto Destinazione. Se il produttore del messaggio non dispone di una destinazione, il metodo restituisce un oggetto Destinazione null.

Eccezioni:

- Eccezione XMSEException

DisableMsgID - Richiama e imposta indicatore di disabilitazione ID messaggio

Interfaccia:

```
Boolean DisableMessageID
{
    get;
    set;
}
```

Indica se un'applicazione ricevente richiede che gli identificativi dei messaggi siano inclusi nei messaggi inviati dal produttore del messaggio e se un'applicazione ricevente richiede che gli identificativi dei messaggi siano inclusi nei messaggi inviati dal produttore del messaggio.

In una connessione a un gestore code o in una connessione in tempo reale a un broker, questo indicatore viene ignorato. Su una connessione a un bus di integrazione servizi, l'indicatore viene rispettato.

L'ID DisabledMsgha i valori seguenti:

- `True`, se un'applicazione di ricezione non richiede che gli identificativi di messaggio siano inclusi nei messaggi inviati dal produttore del messaggio.
- `False`, se un'applicazione ricevente richiede che gli identificativi dei messaggi siano inclusi nei messaggi inviati dal produttore del messaggio.

Eccezioni:

- Eccezione XMSEException

DisableMsgTS - Richiama e imposta indicatore di disabilitazione data/ora

Interfaccia:

```
Boolean DisableMessageTimestamp
{
    get;
    set;
}
```

Indica se un'applicazione di ricezione richiede che la data / ora sia inclusa nei messaggi inviati dal produttore del messaggio e indica se un'applicazione di ricezione richiede che la data / ora sia inclusa nei messaggi inviati dal produttore del messaggio.

In una connessione in tempo reale a un broker, questo indicatore viene ignorato. Su una connessione a un gestore code o su una connessione a un SIB (service integration bus), l'indicatore viene rispettato.

DisableMsgTS ha i valori seguenti:

- `True`, se un'applicazione ricevente non richiede che le date / ore siano incluse nei messaggi inviati dal produttore del messaggio.
- `False`, se un'applicazione ricevente richiede che le date / ore siano incluse nei messaggi inviati dal produttore del messaggio.

Restituisce:**Eccezioni:**

- Eccezione XMSEException

Priority - Ottieni e imposta priorità predefinita

Interfaccia:

```
Int32 Priority
{
    get;
    set;
}
```

Richiamare e impostare la priorità predefinita per i messaggi inviati dal produttore del messaggio.

Il valore della priorità predefinita del messaggio è un numero intero compreso tra 0, la priorità più bassa, e 9, la priorità più elevata.

In una connessione in tempo reale a un broker, la priorità di un messaggio viene ignorata.

Eccezioni:

- Eccezione XMSEException

TimeToLive - Ottieni e imposta il TTL (Time to Live) predefinito

Interfaccia:

```
Int64 TimeToLive
{
    get;
    set;
}
```

Richiamare e impostare il periodo di tempo predefinito per cui un messaggio esiste prima che scada.

Il tempo viene misurato dal momento in cui il mittente del messaggio invia il messaggio ed è il tempo di vita predefinito in millisecondi. Il valore 0 indica che un messaggio non scade mai.

Per una connessione in tempo reale a un broker, questo valore è sempre 0.

Eccezioni:

- Eccezione XMSEException

Metodi

Chiudi - Chiudi produttore messaggi

Interfaccia:

```
void Close();
```

Chiudere il mittente del messaggio.

Se un'applicazione tenta di chiudere un produttore di messaggi che è già chiuso, la chiamata viene ignorata.

Parametri:

Nessuno

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException

Invia - Invia

Interfaccia:

```
void Send(IMessage msg) ;
```

Inviare un messaggio alla destinazione specificata quando è stato creato il produttore del messaggio. Inviare il messaggio utilizzando la modalità di distribuzione predefinita del produttore del messaggio, la priorità e il time to live.

Parametri:**msg (input)**

L'oggetto Messaggio.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException
- Eccezione MessageFormat
- Eccezione InvalidDestination

Invia - Invia (specificando una modalità di consegna, una priorità e un TTL (time to live))

Interfaccia:

```
void Send(IMessage msg,  
          DeliveryMode deliveryMode,  
          Int32 priority,  
          Int64 timeToLive);
```

Inviare un messaggio alla destinazione specificata quando è stato creato il produttore del messaggio. Inviare il messaggio utilizzando la modalità di consegna, la priorità e la durata specificate.

Parametri:**msg (input)**

L'oggetto Messaggio.

deliveryMode (input)

La modalità di consegna per il messaggio, che deve essere uno dei seguenti valori:

DeliveryMode.Persistent

DeliveryMode.NonPersistent

Per una connessione in tempo reale a un broker, il valore deve essere DeliveryMode.NonPersistent.

priorità (input)

La priorità del messaggio. Il valore può essere un numero intero compreso nell'intervallo 0, per la priorità più bassa, per 9, per la priorità più alta. In una connessione in tempo reale a un broker, il valore viene ignorato.

timeToLive (input)

La durata del messaggio in millisecondi. Il valore 0 indica che il messaggio non scade mai. Per una connessione in tempo reale a un broker, il valore deve essere 0.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException
- Eccezione MessageFormat
- Eccezione InvalidDestination
- Eccezione IllegalState

Invia - Invia (a una destinazione specificata)

Interfaccia:

```
void Send(IDestination dest, IMessage msg) ;
```

Inviare un messaggio a una destinazione specificata se si sta utilizzando un produttore di messaggi per cui non è stata specificata alcuna destinazione quando è stato creato il produttore di messaggi. Inviare il messaggio utilizzando la modalità di distribuzione predefinita del produttore del messaggio, la priorità e il time to live.

Generalmente, si specifica una destinazione quando si crea un produttore di messaggi ma, in caso contrario, è necessario specificare una destinazione ogni volta che si invia un messaggio.

Parametri:**dest (input)**

L'oggetto Destinazione.

msg (input)

L'oggetto Messaggio.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException
- Eccezione MessageFormat
- Eccezione InvalidDestination

Invia - Invia (a una destinazione specificata, specificando una modalit ... di consegna, una priorit ... e la durata)

Interfaccia:

```
void Send(IDestination dest,  
          IMessage msg,  
          DeliveryMode deliveryMode,  
          Int32 priority,  
          Int64 timeToLive) ;
```

Inviare un messaggio a una destinazione specificata se si sta utilizzando un produttore di messaggi per cui non è stata specificata alcuna destinazione quando è stato creato il produttore di messaggi. Inviare il messaggio utilizzando la modalità di consegna, la priorità e la durata specificate.

Generalmente, si specifica una destinazione quando si crea un produttore di messaggi ma, in caso contrario, è necessario specificare una destinazione ogni volta che si invia un messaggio.

Parametri:

dest (input)

L'oggetto Destinazione.

msg (input)

L'oggetto Messaggio.

deliveryMode (input)

La modalità di consegna per il messaggio, che deve essere uno dei seguenti valori:

`DeliveryMode.Persistent`

`DeliveryMode.NonPersistent`

Per una connessione in tempo reale a un broker, il valore deve essere

`DeliveryMode.NonPersistent`.

priorità (input)

La priorità del messaggio. Il valore può essere un numero intero compreso nell'intervallo 0, per la priorità più bassa, per 9, per la priorità più alta. In una connessione in tempo reale a un broker, il valore viene ignorato.

timeToLive (input)

La durata del messaggio in millisecondi. Il valore 0 indica che il messaggio non scade mai. Per una connessione in tempo reale a un broker, il valore deve essere 0.

Restituisce:

Nulla

Eccezioni:

- Eccezione `XMSEException`
- Eccezione `MessageFormat`
- Eccezione `InvalidDestination`
- Eccezione `IllegalState`

Proprietà e metodi ereditati

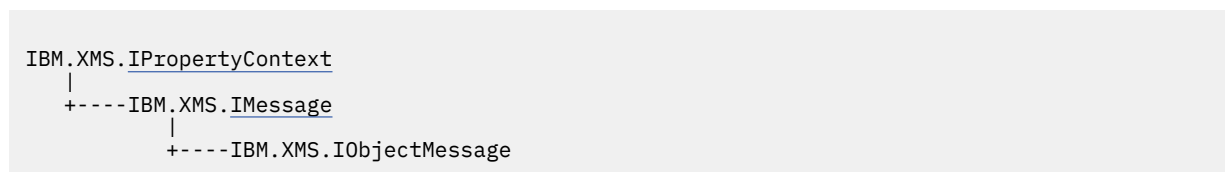
I seguenti metodi sono ereditati dall'interfaccia `IPropertyContext`:

`GetBooleanProperty`, `GetByteProperty`, `GetBytesProperty`, `GetCharProperty`, `GetDoubleProperty`, `GetFloatProperty`, `GetIntProperty`, `GetLongProperty`, `GetObjectProperty`, `GetShortProperty`, `GetStringProperty`, `SetBooleanProperty`, `SetByteProperty`, `SetBytesProperty`, `SetCharProperty`, `SetDoubleProperty`, `SetFloatProperty`, `SetIntProperty`, `SetLongProperty`, `SetObjectProperty`, `SetShortProperty`, `SetStringProperty`

IOBJECTMESSAGE

Un messaggio oggetto è un messaggio il cui corpo comprende un oggetto Java o .NET serializzato.

Gerarchia di eredità:



Riferimenti correlati

[Messaggi oggetto](#)

Il corpo di un messaggio di oggetto contiene un oggettoJava o .NET serializzato.

Proprietà di .NET

Oggetto - Ottieni e imposta oggetto come byte

Interfaccia:

```
System.Object Object
{
    get;
    set;
}

Byte[] GetObject();
```

Richiamare e impostare l'oggetto che costituisce il corpo del messaggio oggetto.

Eccezioni:

- Eccezione XMSEException
- MessageNotReadableException
- MessageEOFException
- MessageNotWritableException

Proprietà e metodi ereditati

Le seguenti proprietà sono ereditate dall'interfaccia [IMessage](#):

[JMSCorrelationID](#), [JMSDeliveryMode](#), [JMSDestination](#), [JMSEExpiration](#), [JMSMessageID](#), [JMSPriority](#), [JMSRedelivered](#), [JMSReplyTo](#), [JMSTimestamp](#), [JMSType](#), [Properties](#)

I seguenti metodi sono ereditati dall'interfaccia [IMessage](#):

[clearBody](#), [clearProperties](#), [PropertyExists](#)

I seguenti metodi sono ereditati dall'interfaccia [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 è una superclasse astratta che contiene metodi che richiamano e impostano le proprietà. Questi metodi vengono ereditati dalle altre classi.

Gerarchia di eredità:

Nessuno

Metodi

Proprietà GetBoolean- Richiama proprietà booleana

Interfaccia:

```
Boolean GetBooleanProperty(String property_name);
```

Richiama il valore della proprietà booleana con il nome specificato.

Parametri:

nome_proprietà (input)

Un oggetto String che incapsula il nome della proprietà.

Restituisce:

Il valore della proprietà.

Contesto thread:

Determinato dalla sottoclasse

Eccezioni:

- Eccezione XMSEException

Proprietà GetByte- Proprietà Get Byte

Interfaccia:

```
Byte GetByteProperty(String property_name) ;  
Int16 GetSignedByteProperty(String property_name) ;
```

Richiamare il valore della proprietà byte identificata dal nome.

Parametri:

nome_proprietà (input)

Un oggetto String che incapsula il nome della proprietà.

Restituisce:

Il valore della proprietà.

Contesto thread:

Determinato dalla sottoclasse

Eccezioni:

- Eccezione XMSEException

Proprietà GetBytes- proprietà Richiamo matrice di byte

Interfaccia:

```
Byte[] GetBytesProperty(String property_name) ;
```

Richiama il valore della proprietà dell'array di byte identificato dal nome.

Parametri:

nome_proprietà (input)

Un oggetto String che incapsula il nome della proprietà.

Restituisce:

Il numero di byte nell'array.

Contesto thread:

Determinato dalla sottoclasse

Eccezioni:

- Eccezione XMSEException

Proprietà GetChar- Acquisisci proprietà carattere

Interfaccia:

```
Char GetCharProperty(String property_name) ;
```

Richiamare il valore della proprietà di caratteri a 2 byte identificata dal nome.

Parametri:**nome_proprietà (input)**

Un oggetto String che incapsula il nome della proprietà.

Restituisce:

Il valore della proprietà.

Contesto thread:

Determinato dalla sottoclasse

Eccezioni:

- Eccezione XMSEException

Proprietà GetDouble- Richiamo proprietà a virgola mobile di precisione doppia

Interfaccia:

```
Double GetDoubleProperty(String property_name) ;
```

Richiama il valore della proprietà a virgola mobile a doppia precisione identificata per nome.

Parametri:**nome_proprietà (input)**

Un oggetto String che incapsula il nome della proprietà.

Restituisce:

Il valore della proprietà.

Contesto thread:

Determinato dalla sottoclasse

Eccezioni:

- Eccezione XMSEException

Proprietà GetFloat- Acquisisci proprietà a virgola mobile

Interfaccia:

```
Single GetFloatProperty(String property_name) ;
```

Ottieni il valore della proprietà a virgola mobile identificata per nome.

Parametri:**nome_proprietà (input)**

Un oggetto String che incapsula il nome della proprietà.

Restituisce:

Il valore della proprietà.

Contesto thread:

Determinato dalla sottoclasse

Eccezioni:

- Eccezione XMSEException

Proprietà GetInt- Proprietà GetInt

Interfaccia:

```
Int32  GetIntProperty(String property_name) ;
```

Richiamare il valore della proprietà integer identificata dal nome.

Parametri:**nome_proprietà (input)**

Un oggetto String che incapsula il nome della proprietà.

Restituisce:

Il valore della proprietà.

Contesto thread:

Determinato dalla sottoclasse

Eccezioni:

- Eccezione XMSEException

Proprietà GetLong- Richiamo proprietà numero intero lungo

Interfaccia:

```
Int64  GetLongProperty(String property_name) ;
```

Richiamare il valore della proprietà long integer identificata dal nome.

Parametri:**nome_proprietà (input)**

Un oggetto String che incapsula il nome della proprietà.

Restituisce:

Il valore della proprietà.

Contesto thread:

Determinato dalla sottoclasse

Eccezioni:

- Eccezione XMSEException

Proprietà GetObject- Acquisisci proprietà oggetto

Interfaccia:

```
Object  GetObjectProperty( String property_name) ;
```


Ottieni il valore e il tipo di dati della proprietà identificata per nome.

Parametri:

nome_proprietà (input)

Un oggetto String che incapsula il nome della proprietà.

Restituisce:

Il valore della proprietà, che è uno dei seguenti tipi di oggetto:

Boolean
Byte
Byte[]
Char
Double
Single
Int32
Int64
Int16
String

Contesto thread:

Determinato dalla sottoclasse

Eccezioni:

- Eccezione XMSEException

Proprietà GetShort- Richiamo proprietà numero intero breve

Interfaccia:

```
Int16 GetShortProperty(String property_name) ;
```

Richiama il valore della proprietà numero intero breve identificata dal nome.

Parametri:

nome_proprietà (input)

Un oggetto String che incapsula il nome della proprietà.

Restituisce:

Il valore della proprietà.

Contesto thread:

Determinato dalla sottoclasse

Eccezioni:

- Eccezione XMSEException

Proprietà GetString- GetString

Interfaccia:

```
String GetStringProperty(String property_name) ;
```

Richiamare il valore della proprietà della stringa identificata per nome.

Parametri:**nome_proprietà (input)**

Un oggetto String che incapsula il nome della proprietà.

Restituisce:

Un oggetto String che incapsula la stringa che è il valore della proprietà. Se è richiesta la conversione dei dati, questo valore è la stringa dopo la conversione.

Contesto thread:

Determinato dalla sottoclasse

Eccezioni:

- Eccezione XMSEException

Proprietà SetBoolean- Imposta proprietà booleana

Interfaccia:

```
void SetBooleanProperty( String property_name, Boolean value) ;
```

Impostare il valore della proprietà booleana identificata dal nome.

Parametri:**nome_proprietà (input)**

Un oggetto String che incapsula il nome della proprietà.

valore (immissione)

Il valore della proprietà.

Restituisce:

Nulla

Contesto thread:

Determinato dalla sottoclasse

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

Proprietà SetByte- Imposta proprietà byte

Interfaccia:

```
void SetByteProperty( String property_name, Byte value) ;  
void SetSignedByteProperty( String property_name, Int16 value) ;
```

Impostare il valore della proprietà byte identificata dal nome.

Parametri:**nome_proprietà (input)**

Un oggetto String che incapsula il nome della proprietà.

valore (immissione)

Il valore della proprietà.

Restituisce:

Nulla

Contesto thread:

Determinato dalla sottoclasse

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

Proprietà SetBytes- Imposta proprietà matrice di byte

Interfaccia:

```
void SetBytesProperty( String property_name, Byte[] value ) ;
```

Impostare il valore della proprietà della matrice di byte identificata dal nome.

Parametri:**nome_proprietà (input)**

Un oggetto String che incapsula il nome della proprietà.

valore (immissione)

Il valore della proprietà, che è un array di byte.

Restituisce:

Nulla

Contesto thread:

Determinato dalla sottoclasse

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

Proprietà SetChar- Imposta proprietà carattere

Interfaccia:

```
void SetCharProperty( String property_name, Char value) ;
```

Impostare il valore della proprietà di caratteri a 2 byte identificata dal nome.

Parametri:**nome_proprietà (input)**

Un oggetto String che incapsula il nome della proprietà.

valore (immissione)

Il valore della proprietà.

Restituisce:

Nulla

Contesto thread:

Determinato dalla sottoclasse

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

Proprietà SetDouble- Imposta proprietà a virgola mobile di precisione doppia

Interfaccia:

```
void SetDoubleProperty( String property_name, Double value) ;
```

Impostare il valore della proprietà a virgola mobile di precisione doppia identificata dal nome.

Parametri:

nome_proprietà (input)

Un oggetto String che incapsula il nome della proprietà.

valore (immissione)

Il valore della proprietà.

Restituisce:

Nulla

Contesto thread:

Determinato dalla sottoclasse

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

Proprietà SetFloat- Imposta proprietà a virgola mobile

Interfaccia:

```
void SetFloatProperty( String property_name, Single value) ;
```

Impostare il valore della proprietà a virgola mobile identificato dal nome.

Parametri:

nome_proprietà (input)

Un oggetto String che incapsula il nome della proprietà.

valore (immissione)

Il valore della proprietà.

Restituisce:

Nulla

Contesto thread:

Determinato dalla sottoclasse

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

Proprietà SetInt- Imposta proprietà numero intero

Interfaccia:

```
void SetIntProperty( String property_name, Int32 value) ;
```

Impostare il valore della proprietà integer identificata dal nome.

Parametri:**nome_proprietà (input)**

Un oggetto String che incapsula il nome della proprietà.

valore (immissione)

Il valore della proprietà.

Restituisce:

Nulla

Contesto thread:

Determinato dalla sottoclasse

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

Proprietà SetLong- Imposta proprietà numero intero lungo

Interfaccia:

```
void SetLongProperty( String property_name, Int64 value) ;
```

Impostare il valore della proprietà long integer identificata dal nome.

Parametri:**nome_proprietà (input)**

Un oggetto String che incapsula il nome della proprietà.

valore (immissione)

Il valore della proprietà.

Restituisce:

Nulla

Contesto thread:

Determinato dalla sottoclasse

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

Proprietà SetObject- Imposta proprietà oggetto

Interfaccia:

```
void SetObjectProperty( String property_name, Object value) ;
```

Impostare il valore e il tipo di dati di una proprietà identificata per nome.

Parametri:**nome_proprietà (input)**

Un oggetto String che incapsula il nome della proprietà.

objectType (input)

Il valore della proprietà, che deve essere uno dei seguenti tipi di oggetto:

Boolean
Byte

Byte[]
Char
Double
Single
Int32
Int64
Int16
String

valore (immissione)

Il valore della proprietà come un array di byte.

lunghezza (input)

Il numero di byte nell'array.

Restituisce:

Nulla

Contesto thread:

Determinato dalla sottoclasse

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

Proprietà SetShort- Imposta proprietà numero intero breve

Interfaccia:

```
void SetShortProperty( String property_name, Int16 value) ;
```

Impostare il valore della proprietà short integer identificata dal nome.

Parametri:

nome_proprietà (input)

Un oggetto String che incapsula il nome della proprietà.

valore (immissione)

Il valore della proprietà.

Restituisce:

Nulla

Contesto thread:

Determinato dalla sottoclasse

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

Proprietà SetString- Imposta proprietà stringa

Interfaccia:

```
void SetStringProperty( String property_name, String value);
```

Impostare il valore della proprietà stringa identificata dal nome.

Parametri:**nome_proprietà (input)**

Un oggetto String che incapsula il nome della proprietà.

valore (immissione)

Un oggetto String che incapsula la stringa che è il valore della proprietà.

Restituisce:

Null

Contesto thread:

Determinato dalla sottoclasse

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

IQueueBrowser

Un'applicazione utilizza un browser della coda per ricercare i messaggi su una coda senza rimuoverli.

Gerarchia di eredità:

```
IBM.XMS.IPropertyContext
System.Collections.IEnumerable
|
+---- IBM.XMS.IQueueBrowser
```

Proprietà di .NET

MessageSelector - Richiama selettore messaggi

Interfaccia:

```
String MessageSelector
{
    get;
}
```

Richiamare il selettore messaggi per il browser della coda.

Il selettore messaggi è un oggetto stringa che incapsula l'espressione del selettore messaggi. Se è richiesta la conversione dei dati, questo valore è l'espressione del selettore di messaggi dopo la conversione. Se il browser della coda non dispone di un selettore di messaggi, il metodo restituisce un oggetto String null.

Eccezioni:

- Eccezione XMSEException

Coda - Coda di ricezione

Interfaccia:

```
IDestination Queue
{
    get;
}
```

Richiamare la coda associata al browser della coda come oggetto di destinazione che rappresenta la coda.

Eccezioni:

- Eccezione XMSEException

Metodi

Chiudi - Chiudi browser coda

Interfaccia:

```
void Close();
```

Chiudere il browser della coda.

Se un'applicazione tenta di chiudere un browser della coda già chiuso, la chiamata viene ignorata.

Parametri:

Nessuno

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException

GetEnumerator - Acquisisci messaggi

Interfaccia:

```
IEnumerator GetEnumerator();
```

Richiamare un elenco dei messaggi sulla coda.

Il metodo restituisce un enumeratore che incapsula un elenco di oggetti `Message`. L'ordine degli oggetti `Message` è uguale a quello in cui i messaggi vengono richiamati dalla coda. L'applicazione può quindi utilizzare l'enumeratore per esaminare ogni messaggio a turno.

L'enumeratore viene aggiornato dinamicamente quando i messaggi vengono inseriti nella coda e rimossi dalla coda. Ogni volta che l'applicazione richiama `IEnumerator.MoveNext()` per sfogliare il messaggio successivo nella coda, il messaggio riflette il contenuto corrente della coda.

Se un'applicazione richiama questo metodo più di una volta per un browser della coda, ogni chiamata restituisce un nuovo enumeratore. L'applicazione può quindi utilizzare più di un enumeratore per esaminare i messaggi su una coda e mantenere più posizioni all'interno della coda.

Parametri:

Nessuno

Restituisce:

L'oggetto iteratore.

Eccezioni:

- Eccezione XMSEException

Proprietà e metodi ereditati

I seguenti metodi sono ereditati dall'interfaccia [IPropertyContext](#):

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

Richiedente

Un'applicazione utilizza un richiedente per inviare un messaggio di richiesta e attendere e ricevere la risposta.

Gerarchia di eredità:

Nessuno

Costruttori

Richiedente - Crea richiedente

Interfaccia:

```
Requestor(ISession sess, IDestination dest);
```

Creare un richiedente.

Parametri:

sess (input)

Un oggetto Session. La sessione non deve essere sottoposta a transazione e deve avere una delle seguenti modalità di riconoscimento:

`AcknowledgeMode.AutoAcknowledge`

`AcknowledgeMode.DupsOkAcknowledge`

dest (input)

Un oggetto Destinazione che rappresenta la destinazione in cui l'applicazione può inviare i messaggi di richiesta.

Contesto thread:

La sessione associata al richiedente

Eccezioni:

- Eccezione `XMSEException`

Metodi

Chiudi - Chiudi richiedente

Interfaccia:

```
void Close();
```

Chiudere il richiedente.

Se un'applicazione tenta di chiudere un richiedente già chiuso, la chiamata viene ignorata.

Nota: Quando un'applicazione chiude un richiedente, anche la sessione associata non viene chiusa. A questo proposito, XMS si comporta in modo diverso rispetto a JMS.

Parametri:

Nessuno

Restituisce:

Nulla

Contesto thread:

Qualsiasi

Eccezioni:

- Eccezione XMSEException

Richiesta - Richiesta risposta

Interfaccia:

```
IMessage Request(IMessage requestMessage);
```

Inviare un messaggio di richiesta, quindi attendere e ricevere una risposta dall'applicazione che riceve il messaggio di richiesta.

Una chiamata a questo metodo si blocca fino a quando non viene ricevuta una risposta o fino alla fine della sessione, a seconda di quale sia la prima.

Parametri:**requestMessage (input)**

L'oggetto Messaggio che incapsula il messaggio di richiesta.

Restituisce:

Un puntatore all'oggetto Messaggio che incapsula il messaggio di risposta.

Contesto thread:

La sessione associata al richiedente

Eccezioni:

- Eccezione XMSEException

Eccezione ResourceAllocation**Gerarchia di eredità:**

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.ResourceAllocationException
```

XMS genera questa eccezione se XMS non può allocare le risorse richieste da un metodo.

Proprietà e metodi ereditati

I seguenti metodi sono ereditati dall'interfaccia [XMSEException](#):

[GetErrorCode](#), [GetLinkedException](#)

SecurityException**Gerarchia di eredità:**

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.SecurityException
```

XMS genera questa eccezione se l'identificativo utente e la parola d'ordine forniti per autenticare un'applicazione vengono rifiutati. XMS genera questa eccezione anche se un controllo di autorizzazione non riesce e impedisce il completamento di un metodo.

Proprietà e metodi ereditati

I seguenti metodi sono ereditati dall'interfaccia XMSException:

GetErrorCode, GetLinkedException

ISessione

Una sessione è un contesto a thread singolo per l'invio e la ricezione di messaggi.

Gerarchia di eredità:

```
IBM.XMS.IPropertyContext
|
+---- IBM.XMS.ISession
```

Per un elenco delle proprietà definite XMS di un oggetto Session, vedere [“Proprietà della sessione”](#) a pagina 197.

Proprietà .NET

AcknowledgeMode - Modalità ricezione riconoscimento

Interfaccia:

```
AcknowledgeMode AcknowledgeMode
{
    get;
}
```

Richiamare la modalità di riconoscimento per la sessione

La modalità di riconoscimento viene specificata quando la sessione viene creata.

Se la sessione non è sottoposta a transazione, la modalità di riconoscimento è uno dei seguenti valori:

```
AcknowledgeMode.AutoAcknowledge
AcknowledgeMode.ClientAcknowledge
AcknowledgeMode.DupsOkAcknowledge
```

Per ulteriori informazioni sulle modalità di riconoscimento, consultare [“Riconoscimento messaggio”](#) a pagina 25.

Una sessione sottoposta a transazione non dispone di una modalità di riconoscimento. Se viene eseguita la transazione della sessione, il metodo restituisce `AcknowledgeMode.SessionTransacted`.

Eccezioni:

- Eccezione XMSException

Transazionali - Determinare se transazionali

Interfaccia:

```
Boolean Transacted
{
    get;
}
```

Determinare se la sessione è sottoposta a transazione.

La transazione indicata è:

- `True`, se la sessione è sottoposta a transazione.
- `False`, se la sessione non è sottoposta a transazione.

Per una connessione in tempo reale a un broker, il metodo restituisce sempre `False`.

Eccezioni:

- Eccezione `XMSEException`

Metodi

Chiudi - Chiudi sessione

Interfaccia:

```
void Close();
```

Chiudere la sessione. Se la sessione viene sottoposta a transazione, viene eseguito il rollback di qualsiasi transazione in corso.

Se un'applicazione tenta di chiudere una sessione già chiusa, la chiamata viene ignorata.

Parametri:

Nessuno

Restituisce:

Null

Contesto thread:

Qualsiasi

Eccezioni:

- Eccezione `XMSEException`

Commit - Commit

Interfaccia:

```
void Commit();
```

Sincronizzare tutti i messaggi elaborati nella transazione corrente

La sessione deve essere una sessione transattiva.

Parametri:

Nessuno

Restituisce:

Null

Eccezioni:

- Eccezione `XMSEException`
- Eccezione `IllegalState`
- Eccezione `TransactionRolledBackException`

CreateBrowser - Crea browser coda

Interfaccia:

```
IQueueBrowser CreateBrowser(IDestination queue) ;
```

Creare un browser di code per la coda specificata.

Parametri:

coda (input)

Un oggetto di destinazione che rappresenta la coda.

Restituisce:

L'oggetto QueueBrowser .

Eccezioni:

- Eccezione XMSEException
- Eccezione InvalidDestination

CreateBrowser - Crea browser della coda (con selettore messaggi)

Interfaccia:

```
IQueueBrowser CreateBrowser(IDestination queue, String selector) ;
```

Creare un browser di code per la coda specificata utilizzando un selettore di messaggi.

Parametri:

coda (input)

Un oggetto di destinazione che rappresenta la coda.

selettore (input)

Un oggetto stringa che incapsula un'espressione del selettore messaggi. Solo i messaggi con proprietà che corrispondono all'espressione del selettore messaggi vengono consegnati al browser della coda.

Un oggetto stringa null indica che non è presente alcun selettore di messaggi per il browser della coda.

Restituisce:

L'oggetto QueueBrowser .

Eccezioni:

- Eccezione XMSEException
- Eccezione InvalidDestination
- Eccezione InvalidSelector

Messaggio CreateBytes- Crea messaggio di byte

Interfaccia:

```
IBytesMessage CreateBytesMessage();
```

Creare un messaggio byte.

Parametri:

Nessuno

Restituisce:

L'oggetto `BytesMessage` .

Eccezioni:

- Eccezione `XMSEException`
- Eccezione `IllegalState`(la sessione è chiusa)

CreateConsumer - Crea consumer

Interfaccia:

```
IMessageConsumer CreateConsumer(IDestination dest) ;
```

Creare un utente del messaggio per la destinazione specificata

Parametri:**dest (input)**

L'oggetto Destinazione.

Restituisce:

L'oggetto `MessageConsumer` .

Eccezioni:

- Eccezione `XMSEException`
- Eccezione `InvalidDestination`

CreateConsumer - Crea consumer (con selettore messaggi)

Interfaccia:

```
IMessageConsumer CreateConsumer(IDestination dest,  
                                String selector) ;
```

Creare un consumatore di messaggi per la destinazione specificata utilizzando un selettore di messaggi.

Parametri:**dest (input)**

L'oggetto Destinazione.

selettore (input)

Un oggetto stringa che incapsula un'espressione del selettore messaggi. Solo i messaggi con proprietà che corrispondono all'espressione del selettore messaggi vengono consegnati al consumatore di messaggi.

Un oggetto `String` null indica che non esiste alcun selettore di messaggi per il consumer del messaggio.

Restituisce:

L'oggetto `MessageConsumer` .

Eccezioni:

- Eccezione `XMSEException`
- Eccezione `InvalidDestination`
- Eccezione `InvalidSelector`

CreateConsumer - Crea consumer (con selettore di messaggi e indicatore di messaggi locali)

Interfaccia:

```
IMessageConsumer CreateConsumer(IDestination dest,  
                                String selector,  
                                Boolean noLocal) ;
```

Creare un consumatore di messaggi per la destinazione specificata utilizzando un selettore di messaggi e, se la destinazione è un argomento, specificando se il consumatore di messaggi riceve i messaggi pubblicati dalla propria connessione.

Parametri:

dest (input)

L'oggetto Destinazione.

selettore (input)

Un oggetto stringa che incapsula un'espressione del selettore messaggi. Solo i messaggi con proprietà che corrispondono all'espressione del selettore messaggi vengono consegnati al consumatore di messaggi.

Un oggetto String null indica che non esiste alcun selettore di messaggi per il consumer del messaggio.

noLocal (input)

Il valore True indica che l'utente del messaggio non riceve i messaggi pubblicati dalla propria connessione. Il valore False indica che il destinatario del messaggio riceve i messaggi pubblicati dalla propria connessione. Il valore predefinito è False.

Restituisce:

L'oggetto MessageConsumer .

Eccezioni:

- Eccezione XMSEException
- Eccezione InvalidDestination
- Eccezione InvalidSelector

CreateDurableSottoscrittore - Crea sottoscrittore durevole

Interfaccia:

```
IMessageConsumer CreateDurableSubscriber(IDestination dest,  
                                         String subscription) ;
```

Creare un sottoscrittore durevole per l'argomento specificato.

Questo metodo non è valido per una connessione in tempo reale a un broker.

Per ulteriori informazioni sui sottoscrittori durevoli, consultare [“Sottoscrittori permanenti”](#) a pagina 34.

Parametri:

dest (input)

Un oggetto Destinazione che rappresenta l'argomento. L'argomento non deve essere un argomento temporaneo.

sottoscrizione (input)

Un oggetto stringa che incapsula un nome che identifica la sottoscrizione durevole. Il nome deve essere univoco all'interno dell'identificativo client per la connessione.

Restituisce:

L'oggetto MessageConsumer che rappresenta il sottoscrittore durevole.

Eccezioni:

- Eccezione XMSEException
- Eccezione InvalidDestination

CreateDurableSottoscrittore - Crea sottoscrittore durevole (con selettore di messaggi e indicatore di messaggio locale)

Interfaccia:

```
IMessageConsumer CreateDurableSubscriber(IDestination dest,  
                                         String subscription,  
                                         String selector,  
                                         Boolean noLocal) ;
```

Creare un sottoscrittore (subscriber) durevole per l'argomento specificato utilizzando un selettore messaggi e specificando se il sottoscrittore (subscriber) durevole riceve i messaggi pubblicati dalla propria connessione.

Questo metodo non è valido per una connessione in tempo reale a un broker.

Per ulteriori informazioni sui sottoscrittori durevoli, consultare [“Sottoscrittori permanenti”](#) a pagina 34.

Parametri:**dest (input)**

Un oggetto Destinazione che rappresenta l'argomento. L'argomento non deve essere un argomento temporaneo.

sottoscrizione (input)

Un oggetto stringa che incapsula un nome che identifica la sottoscrizione durevole. Il nome deve essere univoco all'interno dell'identificativo client per la connessione.

selettore (input)

Un oggetto stringa che incapsula un'espressione del selettore messaggi. Solo i messaggi con proprietà che corrispondono all'espressione del selettore messaggi vengono consegnati al sottoscrittore durevole.

Un oggetto String null indica che non esiste alcun selettore di messaggi per il sottoscrittore durevole.

noLocal (input)

Il valore True indica che il sottoscrittore durevole non riceve i messaggi pubblicati dalla propria connessione. Il valore False indica che il sottoscrittore durevole non riceve i messaggi pubblicati dalla propria connessione. Il valore predefinito è False.

Restituisce:

L'oggetto MessageConsumer che rappresenta il sottoscrittore durevole.

Eccezioni:

- Eccezione XMSEException
- Eccezione InvalidDestination
- Eccezione InvalidSelector

Messaggio CreateMap- Crea messaggio mappa

Interfaccia:

```
IMapMessage CreateMapMessage();
```


Creare un messaggio di associazione.

Parametri:

Nessuno

Restituisce:

L'oggetto MapMessage .

Eccezioni:

- Eccezione XMSEException
- Eccezione IllegalState(la sessione è chiusa)

CreateMessage - Crea messaggio

Interfaccia:

```
IMessage CreateMessage();
```

Creare un messaggio senza corpo.

Parametri:

Nessuno

Restituisce:

L'oggetto Messaggio.

Eccezioni:

- Eccezione XMSEException
- Eccezione IllegalState(la sessione è chiusa)

Messaggio CreateObject- Crea messaggio oggetto

Interfaccia:

```
IObjectMessage CreateObjectMessage();
```

Creare un messaggio oggetto.

Parametri:

Nessuno

Restituisce:

L'oggetto ObjectMessage .

Eccezioni:

- Eccezione XMSEException
- Eccezione IllegalState(la sessione è chiusa)

CreateProducer - Crea Producer

Interfaccia:

```
IMessageProducer CreateProducer(IDestination dest) ;
```

Creare un produttore di messaggi per inviare i messaggi alla destinazione specificata

Parametri:**dest (input)**

L'oggetto Destinazione.

Se si specifica un oggetto Destinazione null, il produttore del messaggio viene creato senza una destinazione. In questo caso, l'applicazione deve specificare una destinazione ogni volta che utilizza il mittente del messaggio per inviare un messaggio.

Restituisce:

L'oggetto MessageProducer .

Eccezioni:

- Eccezione XMSEException
- Eccezione InvalidDestination

CreateQueue - Crea coda

Interfaccia:

```
IDestination CreateQueue(String queue) ;
```

Creare un oggetto Destinazione per rappresentare una coda nel server di messaggistica.

Questo metodo non crea la coda nel server di messaggistica. È necessario creare la coda prima che un'applicazione possa richiamare questo metodo.

Parametri:**coda (input)**

Un oggetto stringa che incapsula il nome della coda o un URI (uniform resource identifier) che identifica la coda.

Restituisce:

L'oggetto Destinazione che rappresenta la coda.

Eccezioni:

- Eccezione XMSEException

Messaggio CreateStream- Crea messaggio di flusso

Interfaccia:

```
IStreamMessage CreateStreamMessage();
```

Creare un messaggio di flusso.

Parametri:

Nessuno

Restituisce:

L'oggetto StreamMessage .

Eccezioni:

- Eccezione XMSEException
- XMS_ILLEGAL_STATE_EXCEPTION

Coda CreateTemporary- Crea coda temporanea

Interfaccia:

```
IDestination CreateTemporaryQueue() ;
```

Creare una coda temporanea

L'ambito della coda temporanea è la connessione. Solo le sessioni create dalla connessione possono utilizzare la coda temporanea.

La coda temporanea rimane fino a quando non viene esplicitamente eliminata o fino al termine della connessione, a seconda di quale sia la prima.

Per ulteriori informazioni sulle code temporanee, consultare [“Destinazioni temporanee” a pagina 32.](#)

Parametri:

Nessuno

Restituisce:

L'oggetto Destinazione che rappresenta la coda temporanea.

Eccezioni:

- Eccezione XMSEException

Argomento CreateTemporary- Crea argomento temporaneo

Interfaccia:

```
IDestination CreateTemporaryTopic() ;
```

Creare un argomento temporaneo.

L'ambito dell'argomento temporaneo è la connessione. Solo le sessioni create dalla connessione possono utilizzare l'argomento temporaneo.

L'argomento temporaneo rimane fino a quando non viene eliminato esplicitamente o fino al termine della connessione, a seconda di quale sia la data più breve.

Per ulteriori informazioni sugli argomenti temporanei, consultare [“Destinazioni temporanee” a pagina 32.](#)

Parametri:

Nessuno

Restituisce:

L'oggetto Destinazione che rappresenta l'argomento temporaneo.

Eccezioni:

- Eccezione XMSEException

Messaggio CreateText- Crea messaggio di testo

Interfaccia:

```
ITextMessage CreateTextMessage();
```

Creare un messaggio di testo con un corpo vuoto.

Parametri:

Nessuno

Restituisce:

L'oggetto TextMessage .

Eccezioni:

- Eccezione XMSEException

CreateTextMessaggio - Crea messaggio di testo (inizializzato)

Interfaccia:

```
ITextMessage CreateTextMessage(String initialValue);
```

Creare un messaggio di testo il cui corpo sia inizializzato con il testo specificato.

Parametri:**initialValue (input)**

Un oggetto String che incapsula il testo per inizializzare il contenuto del messaggio di testo.

Nessuno

Restituisce:

L'oggetto TextMessage .

Eccezioni:

- Eccezione XMSEException

CreateTopic - Crea argomento

Interfaccia:

```
IDestination CreateTopic(String topic) ;
```

Creare un oggetto Destinazione per rappresentare un argomento.

Parametri:**argomento (input)**

Un oggetto Stringa che incapsula il nome dell'argomento o un URI (uniform resource identifier) che identifica l'argomento.

Restituisce:

L'oggetto Destinazione che rappresenta l'argomento.

Eccezioni:

- Eccezione XMSEException

Recupera - Ripristina

Interfaccia:

```
void Recover();
```

Ripristinare la sessione La consegna del messaggio viene arrestata e riavviata con il messaggio non riconosciuto più vecchio.

La sessione non deve essere una sessione con transazioni.

Per ulteriori informazioni sul ripristino di una sessione, consultare [“Riconoscimento messaggio”](#) a pagina 25.

Parametri:

Nessuno

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException
- Eccezione IllegalState

Rollback - Rollback

Interfaccia:

```
void Rollback();
```

Eseguire il rollback di tutti i messaggi elaborati nella transazione corrente

La sessione deve essere una sessione transattiva.

Parametri:

Nessuno

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException
- Eccezione IllegalState

Annulla sottoscrizione - Annulla sottoscrizione

Interfaccia:

```
void Unsubscribe(String subscription);
```

Eliminare una sottoscrizione durevole. Il server di messaggistica elimina il record della sottoscrizione durevole che sta gestendo e non invia ulteriori messaggi al sottoscrittore durevole.

Un'applicazione non può eliminare una sottoscrizione durevole in nessuna delle seguenti circostanze:

- Mentre è presente un consumer di messaggi attivo per la sottoscrizione durevole
- Mentre un messaggio utilizzato fa parte di una transazione in sospeso
- Mentre un messaggio utilizzato non è stato riconosciuto

Questo metodo non è valido per una connessione in tempo reale a un broker.

Parametri:

sottoscrizione (input)

Un oggetto stringa che incapsula il nome che identifica la sottoscrizione durevole.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException

- Eccezione InvalidDestination
- Eccezione IllegalState

Proprietà e metodi ereditati

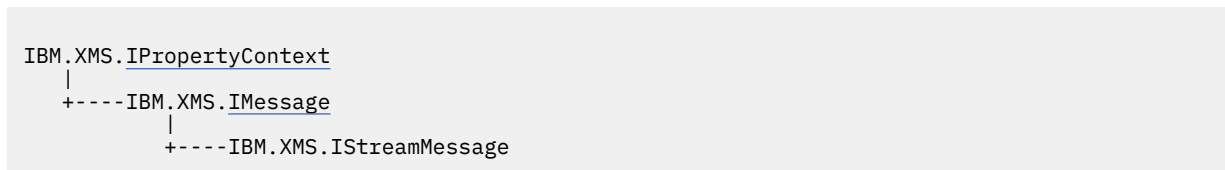
I seguenti metodi sono ereditati dall'interfaccia `IPropertyContext`:

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

IStreamMessage

Un messaggio di flusso è un messaggio il cui contenuto comprende un flusso di valori, dove ogni valore ha un tipo di dati associato. Il contenuto del corpo viene scritto e letto in modo sequenziale.

Gerarchia di eredità:



Quando un'applicazione legge un valore dal flusso di messaggi, il valore può essere convertito da XMS in un altro tipo di dati. Per ulteriori informazioni su questa forma di conversione implicita, consultare [“Il processo di flusso dei messaggi”](#) a pagina 79.

Riferimenti correlati

[Il processo di flusso dei messaggi](#)

Il corpo di un messaggio di flusso contiene un flusso di valori, in cui ogni valore ha un tipo di dati associato.

Metodi

ReadBoolean - Leggi valore booleano

Interfaccia:

```

Boolean ReadBoolean();
  
```

Leggere un valore booleano dal flusso di messaggi.

Parametri:

Nessuno

Restituisce:

Il valore booleano letto.

Eccezioni:

- Eccezione XMSEException
- MessageNotReadableException
- MessageEOFException

ReadByte - Byte di lettura

Interfaccia:

```
Int16  ReadSignedByte();  
Byte   ReadByte();
```

Leggere un numero intero a 8 bit dal flusso di messaggi.

Parametri:

Nessuno

Restituisce:

Il byte letto.

Eccezioni:

- Eccezione XMSEException
- MessageNotReadableException
- MessageEOFException

ReadBytes - Byte letti

Interfaccia:

```
Int32  ReadBytes(Byte[] array);
```

Leggere un array di byte dal flusso di messaggi.

Parametri:

array (immissione)

Il buffer contenente l'array di byte letti e la lunghezza del buffer in byte.

Se il numero di byte nella schiera è inferiore o uguale alla lunghezza del buffer, l'intera schiera viene letta nel buffer. Se il numero di byte nella schiera è maggiore della lunghezza del buffer, il buffer viene riempito con parte della schiera e un cursore interno contrassegna la posizione del byte successivo da leggere. Una chiamata successiva a `readBytes()` legge i byte dall'array iniziando dalla posizione corrente del cursore.

Se si specifica un puntatore null sull'input, la chiamata salta l'array di byte senza leggerlo.

Restituisce:

Il numero di byte letti nel buffer. Se il buffer è parzialmente riempito, il valore è inferiore alla lunghezza del buffer, indicando che non ci sono più byte nell'array rimanenti da leggere.

Se non ci sono byte rimanenti da leggere dall'array prima della chiamata, il valore è `XMSE_END_OF_BYTEARRAY`.

Se si specifica un puntatore nullo sull'immissione, il metodo non restituisce alcun valore.

Eccezioni:

- Eccezione XMSEException
- MessageNotReadableException
- MessageEOFException

ReadChar - Carattere di lettura

Interfaccia:

```
Char ReadChar();
```

Leggere un carattere a 2 byte dal flusso di messaggi.

Parametri:

Nessuno

Restituisce:

Il carattere che viene letto.

Eccezioni:

- Eccezione XMSEException
- MessageNotReadableException
- MessageEOFException

ReadDouble - Leggi numero a virgola mobile di precisione doppia

Interfaccia:

```
Double ReadDouble();
```

Leggere un numero a virgola mobile a doppia precisione a 8 byte dal flusso di messaggi.

Parametri:

Nessuno

Restituisce:

Il numero a virgola mobile di precisione doppia letto.

Eccezioni:

- Eccezione XMSEException
- MessageNotReadableException
- MessageEOFException

ReadFloat - Leggi numero a virgola mobile

Interfaccia:

```
Single ReadFloat();
```

Leggere un numero a virgola mobile a 4 byte dal flusso di messaggi.

Parametri:

Nessuno

Restituisce:

Il numero a virgola mobile letto.

Eccezioni:

- Eccezione XMSEException
- MessageNotReadableException
- MessageEOFException

ReadInt - Lettura numero intero

Interfaccia:

```
Int32 ReadInt();
```

Leggere un intero a 32 bit con segno dal flusso di messaggi.

Parametri:

Nessuno

Restituisce:

Il numero intero letto.

Eccezioni:

- Eccezione XMSEException
- MessageNotReadableException
- MessageEOFException

ReadLong - Lettura numero intero lungo

Interfaccia:

```
Int64 ReadLong();
```

Leggere un intero a 64 bit con segno dal flusso di messaggi.

Parametri:

Nessuno

Restituisce:

Il numero intero lungo letto.

Eccezioni:

- Eccezione XMSEException
- MessageNotReadableException
- MessageEOFException

ReadObject - Lettura oggetto

Interfaccia:

```
Object ReadObject();
```

Leggere un valore dal flusso di messaggi e restituirne il tipo di dati.

Parametri:

Nessuno

Restituisce:

Il valore, che è uno dei seguenti tipi di oggetto:

Boolean
Byte
Byte[]
Char

Double
Single
Int32
Int64
Int16
String

Eccezioni:

Eccezione XMSEException

ReadShort - Lettura numero intero breve

Interfaccia:

```
Int16 ReadShort();
```

Leggere un numero intero a 16 bit dal flusso di messaggi.

Parametri:

Nessuno

Restituisce:

Il numero intero breve letto.

Eccezioni:

- Eccezione XMSEException
- MessageNotReadableException
- MessageEOFException

ReadString - Lettura stringa

Interfaccia:

```
String ReadString();
```

Leggere una stringa dal flusso di messaggi. Se richiesto, XMS converte i caratteri nella stringa nella codepage locale.

Parametri:

Nessuno

Restituisce:

Un oggetto stringa che incapsula la stringa letta. Se è richiesta la conversione dei dati, questa è la stringa dopo la conversione.

Eccezioni:

- Eccezione XMSEException
- MessageNotReadableException
- MessageEOFException

Reimposta - Reimposta

Interfaccia:

```
void Reset();
```

Inserire il corpo del messaggio in modalità di sola lettura e riposizionare il cursore all'inizio del flusso di messaggi.

Parametri:

Nessuno

Restituisce:

Nulla

Eccezioni:

- Eccezione `XMSEException`
- `MessageNotReadableException`
- `MessageEOFException`

WriteBoolean - Scrivi valore booleano

Interfaccia:

```
void WriteBoolean(Boolean value);
```

Scrivere un valore booleano nel flusso di messaggi.

Parametri:

valore (immissione)

Il valore booleano da scrivere.

Restituisce:

Nulla

Eccezioni:

- Eccezione `XMSEException`
- `MessageNotWritableException`

WriteByte - Byte di scrittura

Interfaccia:

```
void WriteByte(Byte value);  
void WriteSignedByte(Int16 value);
```

Scrivere un byte nel flusso di messaggi.

Parametri:

valore (immissione)

Il byte da scrivere.

Restituisce:

Nulla

Eccezioni:

- Eccezione `XMSEException`

- MessageNotWritableException

WriteBytes - Byte scritti

Interfaccia:

```
void WriteBytes(Byte[] value);
```

Scrivere un array di byte nel flusso di messaggi

Parametri:

valore (immissione)

La schiera di byte da scrivere.

lunghezza (input)

Il numero di byte nell'array.

Restituisce:

Null

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

WriteChar - Scrivi carattere

Interfaccia:

```
void WriteChar(Char value);
```

Scrivere un carattere nel flusso di messaggi come 2 byte, prima byte di ordine superiore.

Parametri:

valore (immissione)

Il carattere da scrivere.

Restituisce:

Null

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

WriteDouble - Scrivere il numero a virgola mobile di precisione doppia

Interfaccia:

```
void WriteDouble(Double value);
```

Convertire un numero a virgola mobile di precisione doppia in numero intero lungo e scrivere il numero intero lungo nel flusso di messaggi come byte 8, byte di ordine superiore per primo.

Parametri:

valore (immissione)

Il numero a virgola mobile di precisione doppia da scrivere.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

WriteFloat - Numero virgola mobile di scrittura

Interfaccia:

```
void WriteFloat(Single value);
```

Converti un numero a virgola mobile in un numero intero e scrivi il numero intero nel flusso di messaggi come 4 byte, primo byte di ordine elevato.

Parametri:**valore (immissione)**

Il numero a virgola mobile da scrivere.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

WriteInt - Scrittura numero intero

Interfaccia:

```
void WriteInt(Int32 value);
```

Scrivere un numero intero nel flusso di messaggi come 4 byte, prima byte di ordine superiore.

Parametri:**valore (immissione)**

Il numero intero da scrivere.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

WriteLong - Scrittura numero intero lungo

Interfaccia:

```
void WriteLong(Int64 value);
```

Scrivere un numero intero lungo nel flusso di messaggi come 8 byte, prima byte di ordine superiore.

Parametri:**valore (immissione)**

Il numero intero lungo da scrivere.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

WriteObject - Scrivi oggetto

Interfaccia:

```
void WriteObject(Object value);
```

Scrivere un valore, con un tipo di dati specificato, nel flusso di messaggi.

Parametri:**objectType (input)**

Il valore, che deve essere uno dei seguenti tipi di oggetto:

Boolean
Byte
Byte[]
Char
Double
Single
Int32
Int64
Int16
String

valore (immissione)

Un array di byte contenente il valore da scrivere.

lunghezza (input)

Il numero di byte nell'array.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException

WriteShort - Scrivi numero intero breve

Interfaccia:

```
void WriteShort(Int16 value);
```

Scrivere un numero intero breve nel flusso di messaggi come 2 byte, prima byte di ordine superiore.

Parametri:**valore (immissione)**

Il numero intero breve da scrivere.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

WriteString - Scrivi stringa

Interfaccia:

```
void WriteString(String value);
```

Scrivere una stringa nel flusso di messaggi

Parametri:**valore (immissione)**

Un oggetto String che incapsula la stringa da scrivere.

Restituisce:

Nulla

Eccezioni:

- Eccezione XMSEException
- MessageNotWritableException

Proprietà e metodi ereditati

Le seguenti proprietà sono ereditate dall'interfaccia [IMessage](#):

[JMSCorrelationID](#), [JMSDeliveryMode](#), [JMSDestination](#), [JMSExpiration](#), [JMSMessageID](#), [JMSPriority](#), [JMSRedelivered](#), [JMSReplyTo](#), [JMSTimestamp](#), [JMSType](#), [Properties](#)

I seguenti metodi sono ereditati dall'interfaccia [IMessage](#):

[clearBody](#), [clearProperties](#), [PropertyExists](#)

I seguenti metodi sono ereditati dall'interfaccia [IPropertyContext](#):

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

ITextMessage

Un messaggio di testo è un messaggio il cui contenuto comprende una stringa.

Gerarchia di eredità:

```
IBM.XMS.IPropertyContext
|
+---- IBM.XMS.IMessage
      |
      +---- IBM.XMS.ITextMessage
```

Riferimenti correlati

[Messaggi di testo](#)

Il corpo di un messaggio di testo contiene una stringa.

Proprietà di .NET

Testo - Richiama e imposta testo

Interfaccia:

```
String Text
{
    get;
    set;
}
```

Richiamare e impostare la stringa che costituisce il corpo del messaggio di testo.

Se richiesto, XMS converte i caratteri nella stringa nella codepage locale.

Eccezioni:

- [Eccezione XMSEException](#)
- [MessageNotReadableException](#)
- [MessageNotWritableException](#)
- [MessageEOFException](#)

Proprietà e metodi ereditati

Le seguenti proprietà sono ereditate dall'interfaccia [IMessage](#):

[JMSCorrelationID](#), [JMSDeliveryMode](#), [JMSDestination](#), [JMSExpiration](#), [JMSMessageID](#), [JMSPriority](#), [JMSRedelivered](#), [JMSReplyTo](#), [JMSTimestamp](#), [JMSType](#), [Properties](#)

I seguenti metodi sono ereditati dall'interfaccia [IMessage](#):

[clearBody](#), [clearProperties](#), [PropertyExists](#)

I seguenti metodi sono ereditati dall'interfaccia [IPropertyContext](#):

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

TransactionInProgressException

Gerarchia di eredità:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
      |
      +----IBM.XMS.TransactionInProgressException
```

XMS genera questa eccezione se un'applicazione richiede un'operazione non valida perché è in corso una transazione.

Proprietà e metodi ereditati

I seguenti metodi sono ereditati dall'interfaccia [XMSEException](#):

[GetErrorCode](#), [GetLinkedException](#)

TransactionRolledBackException

Gerarchia di eredità:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
      |
      +----IBM.XMS.TransactionRolledBackException
```

XMS genera questa eccezione se un'applicazione richiama `Session.commit()` per eseguire il commit della transazione corrente, ma viene eseguito il rollback della transazione.

Proprietà e metodi ereditati

I seguenti metodi sono ereditati dall'interfaccia [XMSEException](#):

[GetErrorCode](#), [GetLinkedException](#)

Eccezione XMSEException

Se XMS rileva un errore durante l'elaborazione di una chiamata a un metodo .NET , XMS genera un'eccezione. Un'eccezione è un oggetto che contiene informazioni sull'errore.

Gerarchia di eredità:

```
System.Exception
|
+----IBM.XMS.XMSEException
```

Esistono diversi tipi di eccezione XMS e un oggetto XMSEException è solo un tipo di eccezione. Tuttavia, la classe XMSEException è una superclasse delle altre classi di eccezione XMS . XMS genera un oggetto XMSEException in situazioni in cui nessuno degli altri tipi di eccezione è appropriato.

Proprietà di .NET

ErrorCode - Richiama codice di errore

Interfaccia:

```
public String ErrorCode
{
    get {return errorCode_;}
}
```

Ottieni il codice di errore.

Eccezioni:

- Eccezione XMSEException

LinkedException - Acquisisci eccezione collegata

Interfaccia:

```
public Exception LinkedException
{
    get { return linkedException_;}
```

```
set { linkedException_ = value; }  
}
```

Richiama l'eccezione successiva nella catena di eccezioni

Il metodo restituisce un valore null se non ci sono più eccezioni nella catena.

Eccezioni:

- Eccezione XMSEException

XMSFactoryFactory

Se un'applicazione non utilizza oggetti gestiti, utilizzare questa classe per creare factory di connessione, code e argomenti.

Gerarchia di eredità:

Nessuno

Proprietà di .NET

Metadati - Richiama metadati

Interfaccia:

```
IConnectionMetaData MetaData
```

Richiamare i metadati appropriati per il tipo di connessione dell'oggetto XMSFactoryFactory .

Eccezioni:

Nessuno

Metodi

CreateConnectionFactory - Crea factory di connessione

Interfaccia:

```
IConnectionFactory CreateConnectionFactory();
```

Creare un oggetto ConnectionFactory del tipo dichiarato.

Parametri:

Nessuno

Restituisce:

L'oggetto ConnectionFactory .

Eccezioni:

- Eccezione XMSEException

CreateQueue - Crea coda

Interfaccia:

```
IDestination CreateQueue(String name);
```

Creare un oggetto Destinazione per rappresentare una coda nel server di messaggistica.

Questo metodo non crea la coda nel server di messaggistica. È necessario creare la coda prima che un'applicazione possa richiamare questo metodo.

Parametri:

nome (input)

Un oggetto stringa che incapsula il nome della coda o un URI (uniform resource identifier) che identifica la coda.

Restituisce:

L'oggetto Destinazione che rappresenta la coda.

Eccezioni:

- Eccezione XMSEException

CreateTopic - Crea argomento

Interfaccia:

```
IDestination CreateTopic(String name);
```

Creare un oggetto Destinazione per rappresentare un argomento.

Parametri:

nome (input)

Un oggetto Stringa che incapsula il nome dell'argomento o un URI (uniform resource identifier) che identifica l'argomento.

Restituisce:

L'oggetto Destinazione che rappresenta l'argomento.

Eccezioni:

- Eccezione XMSEException

GetInstance - Richiamare un'istanza di XMSFactoryFactory

Interfaccia:

```
static XMSFactoryFactory GetInstance(int connectionType);
```

Creare un'istanza di XMSFactoryFactory. Un'applicazione XMS utilizza un oggetto XMSFactoryFactory per ottenere un riferimento a un oggetto ConnectionFactory appropriato per il tipo di protocollo richiesto. Questo oggetto ConnectionFactory può quindi produrre connessioni solo per quel tipo di protocollo.

Parametri:

connectionType (input)

Il tipo di connessione per cui l'oggetto ConnectionFactory produce connessioni:

- XMSC.CT_WPM
- XMSC.CT_RTT
- XMSC.CT_WMQ

Restituisce:

L'oggetto XMSFactoryFactory dedicato al tipo di connessione dichiarato.

Eccezioni:

- Eccezione NotSupportedException

Proprietà degli oggetti XMS

capitolo documenta le proprietà dell'oggetto definite da XMS.

capitolo contiene i seguenti sezioni:

- [“Proprietà della connessione” a pagina 180](#)
- [“Proprietà di ConnectionFactory” a pagina 181](#)
- [“Proprietà dei dati ConnectionMeta” a pagina 187](#)
- [“Proprietà della destinazione” a pagina 188](#)
- [“Proprietà di InitialContext” a pagina 190](#)
- [“Proprietà del messaggio” a pagina 191](#)
- [“Proprietà di MessageConsumer” a pagina 197](#)
- [“Proprietà di MessageProducer” a pagina 197](#)
- [“Proprietà della sessione” a pagina 197](#)

Ogni sezione elenca le proprietà di un oggetto del tipo specificato e fornisce una breve descrizione di ciascuna proprietà.

Questo capitolo contiene anche [“Definizioni proprietà” a pagina 197](#) sezione, che fornisce una definizione di ciascuna proprietà.

Se un'applicazione definisce le proprie proprietà degli oggetti descritti in questo capitolo, non causa un errore, ma potrebbe causare risultati imprevedibili.

Nota: I nomi e i valori delle proprietà in questa sezione sono mostrati nel formato XMSC.*NAME*, che è il modulo utilizzato per C e C++. Tuttavia, in .NET, il formato del nome della proprietà può essere XMSC.*NAME* o XMSC_*NAME*, a seconda di come viene utilizzato:

- Se si sta specificando una proprietà, il nome della proprietà deve essere nel formato XMSC.*NAME* come mostrato nel seguente esempio:

```
cf.SetStringProperty(XMSC.WMQ_CHANNEL, "DOTNET.SVRCONN");
```

- Se si sta specificando una stringa, il nome della proprietà deve essere nel formato XMSC_*NAME* come mostrato nel seguente esempio:

```
cf.SetStringProperty("XMSC_WMQ_CHANNEL", "DOTNET.SVRCONN");
```

In .NET, i nomi e i valori delle proprietà vengono forniti come costanti nella classe XMSC. Queste costanti identificano le stringhe e vengono utilizzate da qualsiasi applicazione XMS.NET. Se si utilizzano queste costanti predefinite, i nomi e i valori delle proprietà sono nel formato XMSC.*NOME*, quindi, ad esempio, si utilizza XMSC.USERID, piuttosto che XMSC_USERID.

I tipi di dati sono anche nel formato utilizzato per C/C++. È possibile trovare i corrispondenti valori per .NET in [“Tipi di dati per .NET” a pagina 44](#).

Concetti correlati

Creazione di applicazioni personalizzate

Si creano le proprie applicazioni come si creano le applicazioni di esempio.

Riferimenti correlati

[.NETInterfacce](#)

Questa sezione documenta le interfacce della classe .NET e le relative proprietà e metodi.

Proprietà della connessione

Una panoramica delle proprietà dell'oggetto Connection, con collegamenti a informazioni di riferimento più dettagliate.

<i>Tabella 25. Proprietà della connessione</i>	
Nome della proprietà	Descrizione
“XMSC_WMQ_RESOLVED_QUEUE_MANAGER” a pagina 231	Questa proprietà viene utilizzata per ottenere il nome del gestore code a cui è connesso.
“ID_MANAGER_XMSC_WMQ_RESOLVED_QUEUE_” a pagina 231	Questa proprietà viene popolata con l'ID del gestore code dopo la connessione.
XMSC_WPM_CONNECTION_PROTOCOL	Il protocollo di comunicazione utilizzato per la connessione al motore di messaggistica. Questa proprietà è di sola lettura.
Nome_HOST_WPM_XMSC	Il nome host o l'indirizzo IP del sistema che contiene il motore di messaggistica a cui è connessa l'applicazione. Questa proprietà è di sola lettura.
NOME_ME_WPM_XMSC	Il nome del motore di messaggistica a cui è connessa l'applicazione. Questa proprietà è di sola lettura.
XMSC_WPM_PORT	Il numero della porta di ascolto da parte del motore di messaggistica a cui è connessa l'applicazione. Questa proprietà è di sola lettura.

Un oggetto Connection ha anche proprietà di sola lettura derivate dalle proprietà del factory di connessione utilizzato per creare la connessione. Queste proprietà derivano non solo dalle proprietà del factory di connessione impostate al momento della creazione della connessione, ma anche dai valori predefiniti delle proprietà non impostate. Le proprietà includono solo quelle rilevanti per il tipo di server di messaggistica a cui è connessa l'applicazione. I nomi delle proprietà sono uguali ai nomi delle proprietà del factory di connessione.

Proprietà di ConnectionFactory

Una panoramica delle proprietà dell'oggetto ConnectionFactory , con link a informazioni di riferimento più dettagliate.

<i>Tabella 26. Proprietà di ConnectionFactory</i>	
Nome della proprietà	Descrizione
“XMSC_ASYNC_EXCEPTIONS” a pagina 207	Questa proprietà determina se XMS informa un ExceptionListener solo quando una connessione viene interrotta o quando si verifica qualsiasi eccezione in modo asincrono per una chiamata API XMS. Questa proprietà si applica a tutte le connessioni create da questa ConnectionFactory che hanno un ExceptionListener registrato.
ID_CLI_XMSC	L'identificativo client per una connessione.
XMSC_CONNECTION_TYPE	Il tipo di server di messaggistica a cui si connette un'applicazione.
PASSWORD_XMSC_	Una password che può essere utilizzata per autenticare l'applicazione quando prova a connettersi a un server di messaggistica.
“XMSC_RTT_BROKER_PING_INTERVAL” a pagina 212	L'intervallo di tempo, in millisecondi, dopo il quale XMS .NET controlla la connessione a un server di messaggistica in tempo reale per rilevare eventuale attività.

Tabella 26. Proprietà di ConnectionFactory (Continua)

Nome della proprietà	Descrizione
<u>XMSC_RTT_CONNECTION_PROTOCOL</u>	Il protocollo di comunicazione utilizzato per una connessione in tempo reale a un broker.
<u>XMSC_RTT_NOME_HOST</u>	Il nome host o l'indirizzo IP del sistema su cui viene eseguito un broker.
<u>XMSC_RTT_LOCAL_ADDRESS</u>	Il nome host o l'indirizzo IP dell'interfaccia di rete locale da utilizzare per una connessione in tempo reale a un broker.
<u>XMSC_RTT_MULTICAST</u>	L'impostazione multicast per una destinazione o una factory di connessione.
<u>PORTA_RTT_XMSC</u>	Il numero della porta su cui un broker è in ascolto per le richieste in entrata.
<u>IDUSER_XMSC</u>	Un identificativo utente che può essere utilizzato per autenticare l'applicazione quando prova a connettersi a un server di messaggistica.
<u>XMSC_WMQ_BROKER_CONTROLQ</u>	<p>Il nome della coda di controllo utilizzata da un broker.</p> <p>Nota: Questa proprietà può essere utilizzata con la versione 2.0 di IBM Message Service Client per .NET , ma non ha alcun effetto per un'applicazione connessa a un gestore code IBM WebSphere MQ 7.0 a meno che la proprietà <code>XMSC_WMQ_PROVIDER_VERSION</code> della factory di connessione non sia impostata su un numero di versione inferiore a 7.</p>
<u>XMSC_WMQ_BROKER_PUBQ</u>	<p>Il nome della coda monitorata da un broker dove le applicazioni inviano i messaggi che pubblicano.</p> <p>Nota: Questa proprietà può essere utilizzata con la versione 2.0 di IBM Message Service Client per .NET , ma non ha alcun effetto per un'applicazione connessa a un gestore code IBM WebSphere MQ 7.0 a meno che la proprietà <code>XMSC_WMQ_PROVIDER_VERSION</code> della factory di connessione non sia impostata su un numero di versione inferiore a 7.</p>
<u>XMSC_WMQ_BROKER_QMGR</u>	<p>Il nome del gestore code a cui è connesso un broker.</p> <p>Nota: Questa proprietà può essere utilizzata con la versione 2.0 di IBM Message Service Client per .NET , ma non ha alcun effetto per un'applicazione connessa a un gestore code IBM WebSphere MQ 7.0 a meno che la proprietà <code>XMSC_WMQ_PROVIDER_VERSION</code> della factory di connessione non sia impostata su un numero di versione inferiore a 7.</p>

Tabella 26. Proprietà di ConnectionFactory (Continua)

Nome della proprietà	Descrizione
XMSC_WMQ_BROKER_SUBQ	<p>Il nome della coda sottoscrittore per un utilizzatore di messaggi non durevole.</p> <p>Nota: Questa proprietà può essere utilizzata con la versione 2.0 di IBM Message Service Client per .NET , ma non ha alcun effetto per un'applicazione connessa a un gestore code IBM WebSphere MQ 7.0 a meno che la proprietà XMSC_WMQ_PROVIDER_VERSION della factory di connessione non sia impostata su un numero di versione inferiore a 7.</p>
XMSC_WMQ_BROKER_VERSION	<p>Il tipo di broker utilizzato dall'applicazione per una connessione o per la destinazione.</p> <p>Nota: Questa proprietà può essere utilizzata con la versione 2.0 di IBM Message Service Client per .NET , ma non ha alcun effetto per un'applicazione connessa a un gestore code IBM WebSphere MQ 7.0 a meno che la proprietà XMSC_WMQ_PROVIDER_VERSION della factory di connessione non sia impostata su un numero di versione inferiore a 7.</p>
“ URL CCDT WMQ_XMSC ” a pagina 217	<p>Un URL (A Uniform Resource Locator) che identifica il nome e l'ubicazione del file che contiene la tabella di definizione del canale client e che specifica anche come è possibile accedere al file.</p>
XMSC_WMQ_CHALLENGATO	<p>Il nome del canale da utilizzare per una connessione.</p>
“ XMSC_WMQ_CLIENT_RECONNECT_OPTIONS ” a pagina 218	<p>Questa proprietà specifica le opzioni di riconnessione client per le nuove connessioni create da questo factory</p>
“ XMSC_WMQ_CLIENT_RECONNECT_TIMEOUT ” a pagina 219	<p>Questa proprietà specifica il periodo di tempo, in secondi, in cui una connessione client tenta di riconnettersi.</p>
XMSC_WMQ_CONNECTION_MODE	<p>La modalità in base alla quale un'applicazione si connette a un gestore code.</p>
“ ELENCO NOMI CONNESSIONI WMQ_XMSC ” a pagina 219	<p>Questa proprietà specifica gli host a cui il client tenta di riconnettersi dopo che la connessione è stata interrotta.</p>
XMSC_WMQ_FAIL_IF QUIESCE	<p>Indicazione della mancata riuscita delle chiamate a specifici metodi se il gestore code a cui è connessa l'applicazione è in uno stato di non attivo.</p>
NOME_HOST WMQ_XMSC	<p>Il nome host o l'indirizzo IP del sistema su cui viene eseguito un gestore code.</p>
XMSC_WMQ_LOCAL_ADDRESS	<p>Per una connessione a un gestore code, questa proprietà specifica l'interfaccia di rete locale da utilizzare oppure la porta locale o l'intervallo di porte locali da utilizzare oppure entrambe le cose.</p>

Tabella 26. Proprietà di ConnectionFactory (Continua)

Nome della proprietà	Descrizione
XMSC_WMQ_MESSAGE_SELECTION	<p>Determina se la selezione del messaggio viene effettuata dal client XMS o dal broker.</p> <p>Nota: Questa proprietà può essere utilizzata con la versione 2.0 di IBM Message Service Client per .NET , ma non ha alcun effetto per un'applicazione connessa a un gestore code IBM WebSphere MQ 7.0 a meno che la proprietà XMSC_WMQ_PROVIDER_VERSION della factory di connessione non sia impostata su un numero di versione inferiore a 7.</p>
XMSC_WMQ_MSG_BATCH_XX_ENCODE_CASE_ONE dimensione	<p>Il numero massimo di messaggi da richiamare da una coda in un singolo batch quando si utilizza la consegna di messaggi asincrona.</p> <p>Nota: Questa proprietà può essere utilizzata con la versione 2.0 di IBM Message Service Client per .NET , ma non ha alcun effetto per un'applicazione connessa a un gestore code IBM WebSphere MQ 7.0 a meno che la proprietà XMSC_WMQ_PROVIDER_VERSION della factory di connessione non sia impostata su un numero di versione inferiore a 7.</p>
XMSC_WMQ_POLLING_INTERVAL	<p>Se ciascun listener messaggi in una sessione non dispone di alcun messaggio adatto nella propria coda, questo valore è l'intervallo massimo, in millisecondi, che trascorre prima che ciascun listener messaggi provi di nuovo ad ottenere un messaggio dalla propria coda.</p> <p>Nota: Questa proprietà può essere utilizzata con la versione 2.0 di IBM Message Service Client per .NET , ma non ha alcun effetto per un'applicazione connessa a un gestore code IBM WebSphere MQ 7.0 a meno che la proprietà XMSC_WMQ_PROVIDER_VERSION della factory di connessione non sia impostata su un numero di versione inferiore a 7.</p>
"XMSC_WMQ_PROVIDER_VERSION" a pagina 228	<p>La versione, la release, il livello di modifica e il fix pack del gestore code al quale l'applicazione intende connettersi.</p>
XMSC_WMQ_PORT	<p>Il numero della porta su cui un gestore code è in ascolto per le richieste in entrata.</p>
XMSC_WMQ_PUB_ACK_INTERVAL	<p>Il numero di messaggi pubblicati da un publisher prima che il client XMS richieda un riconoscimento dal broker.</p> <p>Nota: Questa proprietà può essere utilizzata con la versione 2.0 di IBM Message Service Client per .NET , ma non ha alcun effetto per un'applicazione connessa a un gestore code IBM WebSphere MQ 7.0 a meno che la proprietà XMSC_WMQ_PROVIDER_VERSION della factory di connessione non sia impostata su un numero di versione inferiore a 7.</p>
"XMSC_WMQ_PUT_ASYNC_ALLOWED" a pagina 224	<p>Questa proprietà determina se ai mittenti del messaggio è consentito utilizzare i put asincroni per inviare messaggi a questa destinazione.</p>

Tabella 26. Proprietà di ConnectionFactory (Continua)

Nome della proprietà	Descrizione
<u>XMSC_WMQ_QMGR_CCSID</u>	L'identificativo (CCSID) della serie di caratteri codificati, o codepage, in cui i campi di dati carattere definiti in MQI (Message Queue Interface) vengono scambiati tra il client XMS e quello WebSphere MQ .
<u>XMSC_WMQ_QUEUE_MANAGER</u>	Il nome del gestore code a cui connettersi.
<u>XMSC_WMQ_RECEIVE_EXIT</u>	Identifica un'uscita di ricezione del canale da eseguire.
<u>XMSC_WMQ_RECEIVE_EXIT_INIT</u>	I dati utente passati a un'uscita di ricezione del canale quando viene richiamata.
<u>XMSC_WMQ_SECURITY_EXIT</u>	Identifica un'uscita di sicurezza del canale.
<u>XMSC_WMQ_SECURITY_EXIT_INIT</u>	I dati utente passati a un'uscita di sicurezza del canale quando viene richiamata.
“ <u>WMQ_XMSC_SEND_CHECK_COUNT</u> ” a pagina <u>233</u>	Il numero di chiamate di invio da consentire tra i controlli per rilevare eventuali errori di put asincrono all'interno di una singola sessione XML non sottoposta a transazione.
<u>XMSC_WMQ_SEND_EXIT</u>	Identifica un'uscita di invio del canale.
<u>XMSC_WMQ_SEND_EXIT_INIT</u>	I dati utente passati alle uscite di invio del canale quando vengono richiamate.
“ <u>XMSC_WMQ_SHARE_CONV_ALLOWED</u> ” a pagina <u>233</u>	Se una connessione client può condividere il proprio socket con altre connessioni XMS di livello superiore dallo stesso processo allo stesso gestore code, se le definizioni di canale corrispondono. Questa proprietà viene fornita per consentire un isolamento completo delle connessioni in socket separati se necessario per motivi correlati allo sviluppo, alla manutenzione e alla gestione delle applicazioni.
<u>XMSC_WMQ_SSL_CERT_STORES</u>	L'ubicazione dei server che contengono i CRL (certificate revocation list) da utilizzare su una connessione SSL a un gestore code.
<u>XMSC_WMQ_SSL_CIPHER_SPEC</u>	Il nome della CipherSpec da utilizzare su una connessione protetta a un gestore code.
<u>XMSC_WMQ_SSL_CIPHER_SUITE</u>	Il nome della CipherSuite da utilizzare su una connessione TLS a un gestore code. Il protocollo utilizzato nella negoziazione della connessione protetta dipende dalla CipherSuite specificata.
<u>XMSC_WMQ_SSL_CRYPTO_HW</u>	I dettagli di configurazione per l'hardware di crittografia connesso al sistema client.
<u>XMSC_WMQ_SSL_FIPS_REQUIRED</u>	Il valore della proprietà determina se un'applicazione può o meno utilizzare delle suite di crittografia non conformi a FIPS. Se questa proprietà è impostata su true, per la connessione client-server vengono utilizzati solo gli algoritmi FIPS.
<u>XMSC_WMQ_SSL_KEY_REPOSITORY</u>	L'ubicazione del file del database delle chiavi in cui sono memorizzati chiavi e certificati.

Tabella 26. Proprietà di ConnectionFactory (Continua)

Nome della proprietà	Descrizione
XMSC_WMQ_SSL_KEY_RESETCOUNT	Il KeyResetCount rappresenta il numero totale di byte non crittografati inviati e ricevuti in una conversazione SSL prima che venga rinegoziata la chiave segreta.
XMSC_WMQ_SSL_PEER_NAME	Il nome peer da utilizzare su una connessione SSL a un gestore code.
XMSC_WMQ_SYNCPOINT_ALL_GETS	Indica se tutti i messaggi devono essere richiamati dalle code entro il controllo del punto di sincronizzazione.
"XMSC_WMQ_TARGET_CLIENT" a pagina 240	
XMSC_WMQ_TEMP_Q_PREFIX	Il prefisso utilizzato per formare il nome della coda dinamica WebSphere MQ creata quando l'applicazione crea la coda temporanea un XMS .
XMSC_WMQ_TEMP_TOPIC_PREFIX	Quando si creano argomenti temporanei, XMS genera una stringa di argomenti nel formato "TEMP/TEMPTOPICPREFIX/unique_id" oppure se questa proprietà contiene il valore predefinito, viene generata questa stringa, "TEMP/unique_id". La specifica di un valore non vuoto consente la definizione di specifiche code modello per la creazione delle code gestite per i sottoscrittori di argomenti temporanei creati in questa connessione.
XMSC_WMQ_TEMPORARY_MODEL	Il nome della coda modello WebSphere MQ da cui viene creata una coda dinamica quando l'applicazione crea una un XMS coda temporanea.
XMSC_WPM_BUS_NAME	Per una factory di connessione, il nome del bus di integrazione del servizio a cui si connette l'applicazione oppure, per una destinazione, il nome del bus di integrazione del servizio in cui esiste la destinazione.
XMSC_WPM_CONNECTION_PROSSIMITÀ	L'impostazione di prossimità della connessione per la connessione.
XMSC_WPM_DUR_SUB_HOME	Il nome del motore di messaggistica dove vengono gestite tutte le sottoscrizioni durevoli per una connessione o una destinazione.
XMSC_WPM_LOCAL_ADDRESS	Per una connessione a un bus di integrazione del servizio, questa proprietà specifica l'interfaccia di rete locale da utilizzare oppure la porta locale o l'intervallo di porte locali da utilizzare, oppure entrambe le cose.
XMSC_WPM_NON_PERSISTENT_MAP	Il livello di affidabilità dei messaggi non persistenti inviati utilizzando la connessione.
XMSC_WPM_PERSISTENT_MAP	Il livello di affidabilità dei messaggi persistenti inviati utilizzando la connessione.
XMSC_WPM_PROVIDER_ENDPOINTS	Una sequenza di uno o più indirizzi endpoint di server di avvio.
XMSC_WPM_XX_ENCODE_CASE_ONE gruppo_destinazione	Il nome del gruppo di destinazione dei motori di messaggistica.

Tabella 26. Proprietà di ConnectionFactory (Continua)

Nome della proprietà	Descrizione
XMSC_WPM_TARGET_SIGNIFICATIVITÀ	La significatività del gruppo di destinazione dei motori di messaggistica.
XMSC_WPM_TARGET_TRANSPORT_CHAIN	Il nome della catena di trasporto in entrata che l'applicazione deve utilizzare per connettersi a un motore di messaggistica.
XMSC_WPM_XX_ENCODE_CASE_ONE tipo_destinazione	Il tipo di gruppo di destinazione dei motori di messaggistica.
XMSC_WPM_TEMP_Q_PREFIX	Il prefisso utilizzato per formare il nome della coda temporanea creata nel SIB (Service Integration Bus) quando l'applicazione crea la coda temporanea un XMS .
XMSC_WPM_TEMP_TOPIC_PREFIX	Il prefisso utilizzato per formare il nome di un argomento temporaneo creato dall'applicazione.

Concetti correlati

[ConnectionFactoryes e oggetti Connection](#)

Un oggetto ConnectionFactory fornisce un template che un'applicazione utilizza per creare un oggetto Connection. L'applicazione utilizza l'oggetto Connection per creare un oggetto Session.

[Connessione a un SIB \(Service Integration Bus\)](#)

Un'applicazione XMS può connettersi a un SIB (Service Integration Bus) WebSphere Application Server utilizzando una connessione TCP/IP diretta o utilizzando HTTP su TCP/IP.

[Connessioni protette a un gestore code IBM MQ](#)

Per consentire a un'applicazione di XMS .NET di stabilire connessioni sicure a un gestore code IBM MQ , le proprietà pertinenti devono essere definite nell'oggetto ConnectionFactory .

[Connessioni sicure a un motore di messaggistica WebSphere Application Server service integration bus](#)

Per abilitare un'applicazione XMS .NET per stabilire connessioni sicure a un motore di messaggistica WebSphere Application Server service integration bus , le proprietà pertinenti devono essere definite nell'oggetto ConnectionFactory .

[Associazione di proprietà per gli oggetti gestiti](#)

Per consentire alle applicazioni di utilizzare le definizioni oggetto di destinazione e factory di connessione IBM MQ JMS e WebSphere Application Server , le propriet ... richiamate da tali definizioni devono essere associate alle propriet ... XMS corrispondenti che possono essere impostate sulle destinazioni e sulle produzioni connessioni XMS .

Attività correlate

[Creazione di oggetti gestiti](#)

Le definizioni dell'oggetto ConnectionFactory e Destination richieste dalle applicazioni XMS per stabilire una connessione a un server di messaggistica devono essere create utilizzando gli strumenti di gestione appropriati.

Riferimenti correlati

[Proprietà richieste per gli oggetti ConnectionFactory gestiti](#)

Quando un'applicazione crea una factory di connessione, è necessario definire un numero di proprietà per creare una connessione a un server di messaggistica.

Proprietà dei dati ConnectionMeta

Una panoramica delle proprietà dell'oggetto dati ConnectionMeta, con link a informazioni di riferimento più dettagliate.

Tabella 27. Proprietà dei dati ConnectionMeta

Nome della proprietà	Descrizione
<u>XMSC_JMS_MAJOR_VERSION</u>	Il numero di versione principale della specifica JMS su cui si basa XMS . Questa proprietà è di sola lettura.
<u>XMSC_JMS_MINOR_VERSION</u>	Il numero di versione minore della specifica JMS su cui si basa XMS . Questa proprietà è di sola lettura.
<u>VERSIONE JMS XMSC</u>	L'identificativo della versione della specifica JMS su cui si basa XMS . Questa proprietà è di sola lettura.
<u>VERSIONE_MAGGIORE_XMSCR</u>	Il numero di versione del client XMS . Questa proprietà è di sola lettura.
<u>XMSC_VERSIONE_MINORE</u>	Il numero di release del client XMS . Questa proprietà è di sola lettura.
<u>NOME_PROVIDER_XMSCD</u>	Il fornitore del client XMS . Questa proprietà è di sola lettura.
<u>VERSIONE XMSC</u>	L'identificativo della versione del client XMS . Questa proprietà è di sola lettura.

Proprietà della destinazione

Una panoramica delle proprietà dell'oggetto Destinazione, con link a informazioni di riferimento più dettagliate.

Tabella 28. Proprietà della destinazione

Nome della proprietà	Descrizione
<u>MODALITÀ_DISTRIBUZIONE_XMSCD</u>	La modalità di consegna del messaggio inviato alla destinazione.
<u>PRIORITÀ_XMSC_XX_ENCODE_CASE_CAPS_LOCK_OFF</u>	La priorità dei messaggi inviati alla destinazione.
<u>XMSC_RTT_MULTICAST</u>	L'impostazione multicast per una destinazione o una factory di connessione.
<u>XMSC_TIME_TO_LIVE</u>	La durata (TTL, Time To Live) per i messaggi inviati alla destinazione.
<u>XMSC_WMQ_BROKER_VERSION</u>	Il tipo di broker utilizzato dall'applicazione per una connessione o per la destinazione.
<u>XMSC_WMQ_CCSD</u>	L'identificativo (CCSID) della serie di caratteri codificati, o codepage, in cui si trovano le stringhe di dati carattere nel corpo di un messaggio quando il client XMS inoltra il messaggio alla destinazione.
<u>XMSC_WMQ_DUR_SUBQ</u>	il nome della coda sottoscrittore per un sottoscrittore durevole che sta ricevendo messaggi dalla destinazione. Nota: Questa proprietà può essere utilizzata con la versione 2.0 di IBM Message Service Client per .NET , ma non ha alcun effetto per un'applicazione connessa a un gestore code IBM WebSphere MQ 7.0 a meno che la proprietà XMSC_WMQ_PROVIDER_VERSION della factory di connessione non sia impostata su un numero di versione inferiore a 7.

<i>Tabella 28. Proprietà della destinazione (Continua)</i>	
Nome della proprietà	Descrizione
<u>XMSC_WMQ_ENCODING</u>	Modalità di rappresentazione dei dati numerici nel corpo di un messaggio quando il client XMS inoltra il messaggio alla destinazione.
<u>XMSC_WMQ_FAIL_IF QUIESCE</u>	Indicazione della mancata riuscita delle chiamate a specifici metodi se il gestore code a cui è connessa l'applicazione è in uno stato di non attivo.
<u>"CORPO messaggio wmq_xmsc_" a pagina 222</u>	Questa proprietà determina se un'applicazione XMS elabora MQRFH2 di un messaggio IBM WebSphere MQ come parte del payload del messaggio (ovvero, come parte del corpo del messaggio).
<u>"XMSC_WMQ_MQMD_MESSAGE_CONTEXT" a pagina 222</u>	Determina quale livello di contesto del messaggio deve essere impostato dall'applicazione XMS . L'applicazione deve essere in esecuzione con l'autorizzazione di contesto appropriata perché questa proprietà diventi effettiva.
<u>"XMSC_WMQ_MQMD_READ_ENABLED" a pagina 223</u>	Questa proprietà determina se un'applicazione XMS può estrarre o meno i valori dei campi MQMD.
<u>"XMSC_WMQ_MQMD_WRITE_ENABLED" a pagina 223</u>	Questa proprietà determina se un'applicazione XMS può utilizzare o meno i valori dei campi MQMD.
<u>"XMSC_WMQ_READ_AHEAD_ALLOWED" a pagina 224</u>	Questa proprietà determina se ai destinatari dei messaggi e ai browser della coda è consentito utilizzare la lettura anticipata per ottenere i messaggi non persistenti e non transazionali da questa destinazione in un buffer interno prima di riceverli.
<u>"XMSC_WMQ_READ_AHEAD_CLOSE_POLICY" a pagina 225</u>	Questa proprietà determina, per i messaggi di cui si sta eseguendo la consegna a un listener di messaggi asincroni, cosa succede ai messaggi nel buffer di lettura anticipata interno quando viene chiuso il destinatario del messaggio.
<u>"XMSC_WMQ_RECEIVE_CC SID" a pagina 230</u>	Proprietà di destinazione che imposta il CCSID di destinazione per la conversione del messaggio del gestore code. Il valore viene ignorato a meno che XMSC_WMQ_RECEIVE_CONVERSION non sia impostato su WMQ_RECEIVE_CONVERSION_QMGR.
<u>"XMSC_WMQ_RECEIVE_CONVERSION" a pagina 230</u>	Proprietà di destinazione che determina se la conversione dati verrà eseguita dal gestore code.
<u>XMSC_WMQ_TARGET_CLIENT</u>	Indica se i messaggi inviati alla destinazione contengono un'intestazione MQRFH2.
<u>XMSC_WMQ_TEMP_TOPIC_PREFIX</u>	Quando si creano argomenti temporanei, XMS genera una stringa di argomenti nel formato "TEMP/TEMPTOPICPREFIX/unique_id" oppure se questa proprietà contiene il valore predefinito, viene generata questa stringa, "TEMP/unique_id". La specifica di un valore non vuoto consente la definizione di specifiche code modello per la creazione delle code gestite per i sottoscrittori di argomenti temporanei creati in questa connessione.

Tabella 28. Proprietà della destinazione (Continua)

Nome della proprietà	Descrizione
<u>XMSC_WPM_BUS_NAME</u>	Per una factory di connessione, il nome del bus di integrazione del servizio a cui si connette l'applicazione oppure, per una destinazione, il nome del bus di integrazione del servizio in cui esiste la destinazione.
<u>SPACE</u> <u>XMSC_WPM_TOPIC_XX_ENCODE_CASE_ONE</u> spazio	Il nome dello spazio argomento che contiene l'argomento.

Concetti correlati

ConnectionFactoryies e oggetti Connection

Un oggetto ConnectionFactory fornisce un template che un'applicazione utilizza per creare un oggetto Connection. L'applicazione utilizza l'oggetto Connection per creare un oggetto Session.

Connessione a un SIB (Service Integration Bus)

Un'applicazione XMS può connettersi a un SIB (Service Integration Bus) WebSphere Application Server utilizzando una connessione TCP/IP diretta o utilizzando HTTP su TCP/IP.

Destinazioni

Un'applicazione di XMS utilizza un oggetto Destinazione per specificare la destinazione dei messaggi inviati e l'origine dei messaggi ricevuti.

Caratteri jolly di destinazione

XMS fornisce supporto per i caratteri jolly di destinazione, garantendo che i caratteri jolly possano essere trasmessi al luogo in cui sono necessari per la corrispondenza. Esiste un diverso schema di caratteri jolly per ciascun tipo di server che XMS può utilizzare.

Identificativi risorsa uniforme argomento

L'URI (uniform resource identifier) dell'argomento specifica il nome dell'argomento; può anche specificare una o più proprietà per esso.

Identificativi risorsa uniforme coda

L'URI di una coda specifica il nome della coda; può anche specificare una o più proprietà della coda.

Destinazioni temporanee

Le applicazioni XMS possono creare e utilizzare destinazioni temporanee.

Associazione di proprietà per gli oggetti gestiti

Per consentire alle applicazioni di utilizzare le definizioni oggetto di destinazione e factory di connessione IBM MQ JMS e WebSphere Application Server , le propriet ... richiamate da tali definizioni devono essere associate alle propriet ... XMS corrispondenti che possono essere impostate sulle destinazioni e sulle produzioni connessioni XMS .

Attività correlate

Creazione di oggetti gestiti

Le definizioni dell'oggetto ConnectionFactory e Destination richieste dalle applicazioni XMS per stabilire una connessione a un server di messaggistica devono essere create utilizzando gli strumenti di gestione appropriati.

Riferimenti correlati

Proprietà richieste per gli oggetti Destinazione gestiti

Un'applicazione che sta creando una destinazione deve impostare diverse proprietà dell'applicazione su un oggetto Destinazione gestito.

Proprietà di InitialContext

Una panoramica delle proprietà dell'oggetto InitialContext , con link a informazioni di riferimento più dettagliate.

Tabella 29. Proprietà di InitialContext

Nome della proprietà	Descrizione
URL_PROVIDER_IC_XMSC	Utilizzata per individuare la directory di denominazione JNDI in modo che il servizio di denominazione COS non debba necessariamente trovarsi sullo stesso server del servizio web.
XMSC_IC_SECURITY_AUTHENTICATION	Basato su SECURITY_AUTHENTICATION dell'interfaccia di contesto Java . Questa proprietà è applicabile solo al contesto di denominazione COS.
CREDENZIALI XMSC_IC_SECURITY_CREDENTIALS	Basato sull'interfaccia di contesto Java SECURITY_CREDENTIALS. Questa proprietà è applicabile solo al contesto di denominazione COS.
XMSC_IC_SECURITY_PRINCIPAL	In base all'interfaccia di contesto Java SECURITY_PRINCIPAL. Questa proprietà è applicabile solo al contesto di denominazione COS.
PROTOCOLLO_SICUREZZA_XMSC_ICA	Basato su Java SECURITY_PROTOCOL dell'interfaccia del contesto Questa proprietà è applicabile solo al contesto di denominazione COS.
URL_IC_XMSC	Per i contesti LDAP e FileSystem, l'indirizzo del repository che contiene gli oggetti amministrati. Per i contesti di denominazione COS, l'indirizzo del servizio web che cerca gli oggetti nella directory.

Concetti correlati

Proprietà InitialContext

I parametri del costruttore InitialContext includono l'ubicazione del repository di oggetti gestiti, forniti come URI (uniform resource indicator). Affinché un'applicazione stabilisca una connessione al repository, potrebbe essere necessario fornire più informazioni rispetto a quelle contenute nell'URI.

Formato URI per contesti iniziali XMS

L'ubicazione del repository di oggetti gestiti viene fornita come URI (uniform resource indicator). Il formato dell'URI dipende dal tipo di contesto.

Richiamo di oggetti gestiti

XMS richiama un oggetto gestito dal repository utilizzando l'indirizzo fornito quando viene creato l'oggetto InitialContext o nelle proprietà InitialContext .

Attività correlate

Oggetti InitialContext

Un'applicazione deve creare un contesto iniziale da utilizzare per stabilire una connessione al repository di oggetti gestiti per richiamare gli oggetti gestiti richiesti.

Proprietà del messaggio

Una panoramica delle proprietà dell'oggetto Messaggio, con link a informazioni di riferimento più dettagliate.

Tabella 30. Proprietà del messaggio

Nome della proprietà	Descrizione
<u>IMPOSTAZIONE_IBM_JMS</u>	L'identificativo (CCSID) della serie di caratteri codificati, o codepage, in cui si trovano le stringhe di dati di caratteri nel contenuto del messaggio quando il client XMS inoltra il messaggio alla destinazione desiderata. In XMS questa proprietà ha un valore numerico e si associa a CCSID. Tuttavia, questa proprietà si basa su una proprietà JMS per cui dispone di un valore di tipo stringa ed esegue l'associazione al set di caratteri Java che rappresenta questo CCSID numerico.
<u>JMS_IBM_Encoding</u>	Modalità di rappresentazione dei dati numerici nel corpo del messaggio quando il client XMS inoltra il messaggio alla destinazione desiderata.
<u>MESSAGGIO_IBM_JMS</u>	Testo che descrive perché il messaggio è stato inviato alla destinazione eccezioni. Questa proprietà è di sola lettura.
<u>JMS_IBM_EXCEPTIONPROBLEMDESTINATION</u>	Il nome della destinazione in cui si trovava il messaggio prima che il messaggio venisse inviato alla destinazione eccezioni.
<u>JMS_IBM_EXCEPTIONREASON</u>	Un codice motivo che indica il motivo per cui il messaggio è stato inviato alla destinazione eccezioni.
<u>JMS_IBM_EXCEPTIONTIMESTAMP</u>	L'ora in cui il messaggio è stato inviato alla destinazione eccezioni.
<u>JMS_IBM_FEEDBACK</u>	Un codice che indica la natura di un messaggio di report.
<u>FORM_IBM_JMS</u>	La natura dei dati dell'applicazione nel messaggio
<u>JMS_IBM_LAST_MSG_IN_GROUP</u>	Indicare se il messaggio è l'ultimo messaggio in un gruppo di messaggi.
<u>TIPO_IBM_JMS</u>	Il tipo del messaggio.
<u>TIPO_IBM_JMS</u>	Il tipo di applicazione che ha inviato il messaggio.
<u>DATA_IBM_JMS</u>	La data in cui il messaggio è stato inviato.
<u>ORA_IBM_JMS</u>	L'ora in cui il messaggio è stato inviato.
<u>COA IBM REPORT_JMS</u>	Richiedere i messaggi di report di 'conferma all'arrivo', specificando quanti dati dell'applicazione dal messaggio originale devono essere inclusi in un messaggio di report.
<u>COD REPORT_IBM_JMS</u>	Richiedere i messaggi di report di 'conferma alla consegna', specificando quanti dati dell'applicazione dal messaggio originale devono essere inclusi in un messaggio di report.
<u>MSG_DISCARD_IBM_REPORT_JMS</u> JMS	Richiedere che il messaggio venga eliminato se non può essere consegnato alla sua destinazione prevista.
<u>JMS_IBM_REPORT_ECCEZIONE</u>	Richiedere dei messaggi di report di eccezione, specificando quanti dati dell'applicazione dal messaggio originale devono essere inclusi in un messaggio di report.
<u>JMS_IBM_REPORT_EXPIRATION</u>	Richiedere dei messaggi di report di scadenza, specificando quanti dati dell'applicazione dal messaggio originale devono essere inclusi in un messaggio di report.

<i>Tabella 30. Proprietà del messaggio (Continua)</i>	
Nome della proprietà	Descrizione
<u>NAN_IBM_REPORT_JMS</u>	Richiedere i messaggi di report di notifica di azione non eseguita con esito positivo.
<u>PAN_IBM_REPORT_JMS</u>	Richiedere i messaggi di report di notifica di azione eseguita con esito positivo.
<u>ID_CORREL_IBM_REPORT_JMS</u>	Richiedere che l'identificativo correlazioni di qualsiasi messaggio di report o di risposta sia uguale all'identificativo correlazioni del messaggio originale.
<u>ID_MSG_PASS_IBM_REPORT_JMS</u>	Richiedere che l'identificativo del messaggio di qualsiasi messaggio di report o di risposta sia uguale all'identificativo del messaggio originale.
<u>RETAIN_IBM_JMS</u>	L'impostazione di questa proprietà indica al gestore code di trattare un messaggio come Pubblicazione conservata.
<u>ID_MESSAGGIO_SISTEMA_JMS</u>	Un identificativo che identifica il messaggio in modo univoco all'interno del bus di integrazione del servizio. Questa proprietà è di sola lettura.
<u>IDAPP_JMS</u>	Il nome dell'applicazione che ha inviato il messaggio.
<u>CONTEGGIO_DISTRIBUZIONE_JMSX</u>	Il numero di tentativi di consegna del messaggio.
<u>IDGROUP_JMS</u>	L'identificativo del gruppo messaggi a cui appartiene il messaggio.
<u>GROUPSEQ_JMSX</u>	Il numero di sequenza del messaggio all'interno di un gruppo messaggi.
<u>IDUSER_JMS</u>	L'identificativo utente associato all'applicazione che ha inviato il messaggio.

Proprietà JMS_IBM_MQMD*

IBM Message Service Client for .NET consente alle applicazioni client di leggere / scrivere i campi MQMD utilizzando API. Consente inoltre l'accesso ai dati dei messaggi di MQ . Per impostazione predefinita, l'accesso a MQMD è disattivato e deve essere abilitato esplicitamente dall'applicazione utilizzando le proprietà di destinazione XMSC_WMQ_MQMD_WRITE_ENABLED e XMSC_WMQ_MQMD_READ_ENABLED. Queste due proprietà sono indipendenti l'una dall'altra.

Tutti i campi MQMD tranne StrucId e Version sono esposti come proprietà aggiuntive dell'oggetto Message e hanno come prefisso JMS_IBM_MQMD.

Le proprietà JMS_IBM_MQMD* hanno la precedenza su altre proprietà come JMS_IBM* descritte nella tabella precedente.

invio di messaggi

Tutti i campi MQMD tranne StrucId e Version sono rappresentati. Queste proprietà fanno riferimento solo ai campi MQMD; dove una proprietà si verifica sia nell'intestazione MQMD che MQRFH2 , la versione in MQRFH2 non è impostata o estratta. È possibile impostare una qualsiasi di queste proprietà, ad eccezione di JMS_IBM_MQMD_BackoutCount. Qualsiasi valore impostato per JMS_IBM_MQMD_BackoutCount viene ignorato.

Se una proprietà ha una lunghezza massima e si fornisce un valore troppo lungo, il valore viene troncato.

Per alcune proprietà, è necessario impostare anche la proprietà XMSC_WMQ_MQMD_MESSAGE_CONTEXT sull'oggetto Destinazione. L'applicazione deve essere in

esecuzione con l'autorizzazione di contesto appropriata perché questa proprietà diventi effettiva. Se non si imposta XMSC_WMQ_MQMD_MESSAGE_CONTEXT su un valore appropriato, il valore della proprietà viene ignorato. Se si imposta XMSC_WMQ_MQMD_MESSAGE_CONTEXT su un valore appropriato ma non si dispone dell'autorizzazione di contesto sufficiente per il gestore code, viene emessa un'eccezione. Le proprietà che richiedono valori specifici di XMSC_WMQ_MQMD_MESSAGE_CONTEXT sono le seguenti.

Le seguenti proprietà richiedono che XMSC_WMQ_MQMD_MESSAGE_CONTEXT sia impostato su XMSC_WMQ_MDCTX_SET_IDENTITY_CONTEXT o XMSC_WMQ_MDCTX_SET_ALL_CONTEXT:

- JMS_IBM_MQMD_UserIdentifier
- JMS_IBM_MQMD_AccountingToken
- Dati JMS_IBM_MQMD_ApplIdentity

Le seguenti proprietà richiedono che XMSC_WMQ_MQMD_MESSAGE_CONTEXT sia impostato su XMSC_WMQ_MDCTX_SET_ALL_CONTEXT:

- Tipo JMS_IBM_MQMD_PutAppl
- Nome JMS_IBM_MQMD_PutAppl
- JMS_IBM_MQMD_PutDate
- PutTime JMS_IBM_MQMD_
- JMS_IBM_MQMD_ApplOriginDati

ricezione di messaggi

Tutte queste proprietà sono disponibili su un messaggio ricevuto se la proprietà XMSC_WMQ_MQMD_READ_ENABLED è impostata su true, indipendentemente dalle proprietà effettive impostate dall'applicazione di produzione. Un'applicazione non può modificare le proprietà di un messaggio ricevuto a meno che tutte le proprietà non vengano prima cancellate, in base alla specifica JMS. Il messaggio ricevuto può essere inoltrato senza modificare le proprietà.

Nota: Se la propria applicazione riceve un messaggio da una destinazione con la proprietà XMSC_WMQ_MQMD_READ_ENABLED impostata su true e lo inoltra a una destinazione con XMSC_WMQ_MQMD_WRITE_ENABLED impostata su true, tutti i valori del campo MQMD del messaggio ricevuto vengono copiati nel messaggio inoltrato. Tabella delle proprietà

<i>Tabella 31. Proprietà dell'oggetto Message che rappresentano i campi MQMD</i>		
Proprietà	Descrizione	Tipo
REPORT IBM_MQMD_JMS	Opzioni per messaggi di report	System.Int32
JMS_IBM_MQMD_MSGTIPO	Tipo messaggio	System.Int32
EXPIRY IBM_MQMD_JMS	Durata messaggio	System.Int32
IBM_MQMD_FEEDBACK JMS	Feedback o codice motivo	System.Int32
JMS_IBM_MQMD_ENCODING	Codifica numerica dei dati di messaggio	System.Int32
ID_CODICE_IBM_MQMD_JMS	CSID (Character set identifier) dei dati del messaggio	System.Int32
JMS_IBM_MQMD_FORMATO	Nome formato dei dati di messaggio	System.String
JMS_IBM_MQMD_PRIORITY	Priorità messaggio	System.Int32
<p>Nota: Se si assegna un valore a JMS_IBM_MQMD_PRIORITY non compreso nell'intervallo 0-9, questo valore viola la specifica JMS.</p>		

Tabella 31. Proprietà dell'oggetto Message che rappresentano i campi MQMD (Continua)

Proprietà	Descrizione	Tipo
JMS_IBM_MQMD_PERSISTENZA	Persistenza messaggio	System.Int32
IDMSG_IBM_MQMD_JMS Nota: La specifica JMS indica che l'ID messaggio deve essere impostato dal provider JMS e che deve essere univoco o null. Se si assegna un valore a JMS_IBM_MQMD_MSGID, questo valore viene copiato in JMSMessageID. Quindi non è impostato dal provider JMS e potrebbe non essere univoco: questo valore viola la specifica JMS.	ID messaggio	Matrice byte Nota: L'utilizzo delle proprietà della matrice di byte su un messaggio viola le specifiche JMS.
IDCORREL_IBM_MQMD_JMS Nota: Se si assegna un valore a JMS_IBM_MQMD_CORRELID che inizia con la stringa 'ID:', questo valore viola la specifica JMS.	Identificativo di correlazione	Matrice byte Nota: L'utilizzo delle proprietà della matrice di byte su un messaggio viola le specifiche JMS.
JMS_IBM_MQMD_BACKOUTCOUNT	Contatore backout	System.Int32
QMS_IBM_MQMD_REPLYTOQ	Nome della coda di risposta	System.String
JMS_IBM_MQMD_REPLYTOQMGR	Nome del gestore code di risposta	System.String
JMS_IBM_MQMD_USERIDENTIFIER	Identificativo utente	System.String
JMS_IBM_MQMD_ACCOUNTINGTOKEN	Token account	Matrice byte Nota: L'utilizzo delle proprietà della matrice di byte su un messaggio viola le specifiche JMS.
JMS_IBM_MQMD_APPLIDENTITYDATA	Dati dell'applicazione correlati all'identità	System.String
PUTAPPLTYPE_IBM_MQMD_JMS	Tipo di applicazione che inserisce il messaggio	System.Int32
NomePUTAPPL_IBM_MQMD_JMS	Nome dell'applicazione che ha inserito il messaggio	System.String
PUTDATE_MQMD_IBM_JMS	Data in cui il messaggio è stato inserito	System.String
PUTORA_IBM_MQMD_JMS	Ora in cui il messaggio è stato inserito	System.String
JMS_IBM_MQMD_APPLORIGINDATA	Dati dell'applicazione correlati all'origine	System.String
IDGROUP_IBM_MQMD_JMS	ID gruppo	Matrice byte Nota: L'utilizzo delle proprietà della matrice di byte su un messaggio viola le specifiche JMS.

Tabella 31. Proprietà dell'oggetto Message che rappresentano i campi MQMD (Continua)

Proprietà	Descrizione	Tipo
JMS_IBM_MQMD_MSGSEQNUMBER	Numero di sequenza del messaggio locale all'interno del gruppo	System.Int32
OFFSET IBM_MQMD_JMS	Offset di dati nel messaggio fisico dall'inizio del messaggio logico	System.Int32
JMS_IBM_MQMD_MSGFLAGS	Indicatori di messaggio	System.Int32
JMS_IBM_MQMD_ORIGINALLENGTH	Lunghezza del messaggio originale	System.Int32

Per ulteriori informazioni, vedere [MQMD](#) .

Esempi

Questo esempio determina l'inserimento di un messaggio in una coda o in un argomento con MQMD.UserIdIdentifier impostato su 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
// ...
```

È necessario impostare XMSC_WMQ_MQMD_MESSAGE_CONTEXT prima di impostare JMS_IBM_MQMD_USERIDENTIFIER. Per ulteriori informazioni relative all'utilizzo di XMSC_WMQ_MQMD_MESSAGE_CONTEXT, consultare Proprietà dell'oggetto Messaggio.

Allo stesso modo, è possibile estrarre il contenuto dei campi MQMD impostando XMSC_WMQ_MQMD_READ_ENABLED su true prima di ricevere un messaggio e quindi utilizzando i metodi get del messaggio, ad esempio la proprietà getString. Tutte le proprietà ricevute sono di sola lettura.

Questo esempio risulta nel campo del valore che contiene il valore di MQMD.ApplIdentityData di un messaggio ricevuto da una coda o da un argomento.

```
// 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 required MQMD field value using a property
System.String value = rcvMsg.getStringProperty(JMS_IBM_MQMD_APPLIDENTITYDATA);
```

Proprietà di MessageConsumer

Una panoramica delle proprietà dell'oggetto MessageConsumer , con link a informazioni di riferimento più dettagliate.

Nome della proprietà	Descrizione
XMSC_IS_SUBSCRIPTION_MULTICAST	Indica se i messaggi vengono consegnati al consumatore di messaggi utilizzando WebSphere MQ Multicast Transport. Questa proprietà è di sola lettura.
XMSC_IS_SUBSCRIPTION_RELIABLE_MULTICAST	Indica se i messaggi vengono consegnati al consumatore di messaggi utilizzando WebSphere MQ Multicast Transport con una QoS (quality of service) affidabile. Questa proprietà è di sola lettura.

Fare riferimento a [.Proprietà NET di IMessageConsumer](#) per ulteriori dettagli.

Proprietà di MessageProducer

Una panoramica delle proprietà dell'oggetto MessageProducer , con collegamenti a informazioni di riferimento più dettagliate.

Consultare [.Proprietà NET di IMessageProducer](#) per ulteriori dettagli.

Proprietà della sessione

Una panoramica delle proprietà dell'oggetto Session, con link a informazioni di riferimento più dettagliate.

Consultare [.Proprietà NET di ISession](#) per ulteriori dettagli.

Definizioni proprietà

Questa sezione fornisce una definizione di ogni proprietà oggetto.

Ogni definizione di proprietà include le seguenti informazioni:

- Il tipo di dati della proprietà
- I tipi di oggetto che hanno la proprietà
- Per una proprietà di Destinazione, il nome che può essere utilizzato in un URI (uniform resource identifier)
- Una descrizione più dettagliata dell'immobile
- I valori validi della proprietà
- Il valore predefinito della proprietà

Le proprietà i cui nomi iniziano con uno dei seguenti prefissi sono rilevanti solo per il tipo di collegamento specificato:

RTT XMSC

Le proprietà sono rilevanti solo per una connessione in tempo reale a un broker. I nomi delle proprietà vengono definiti come costanti denominate nel file di intestazione `xmsc_rtt.h`.

WMQ XMSC

Le proprietà sono rilevanti solo quando un'applicazione si connette ad un gestore code WebSphere MQ . I nomi delle proprietà vengono definiti come costanti denominate nel file di intestazione `xmsc_wmq.h`.

WPM XMSC

Le proprietà sono pertinenti solo quando un'applicazione si connette a un SIB (service integration bus) WebSphere . I nomi delle proprietà vengono definiti come costanti denominate nel file di intestazione `xmsc_wpm.h`.

Se non diversamente specificato nelle loro definizioni, le restanti proprietà sono rilevanti per tutti i tipi di connessione. I nomi delle proprietà vengono definiti come costanti denominate nel file di intestazione `xmsc.h`. Le proprietà i cui nomi iniziano con il prefisso `JMSX` sono JMS proprietà definite di un messaggio e le proprietà i cui nomi iniziano con il prefisso `JMS_IBM` sono IBM proprietà definite di un messaggio. Per ulteriori informazioni sulle proprietà dei messaggi, consultare [“Proprietà di un messaggio XMS” a pagina 71](#).

Se non diversamente specificato nella relativa definizione, ogni proprietà è rilevante sia nei domini Point-to-point che Pubblicazione/sottoscrizione.

Un'applicazione può ottenere e impostare il valore di qualsiasi proprietà, a meno che la proprietà non sia di sola lettura.

JMS_IBM_CHARACTER_SET

Tipo di dati:

`System.Int32`

Proprietà di:

Messaggio

L'identificativo (CCSID) della serie di caratteri codificati, o codepage, in cui si trovano le stringhe di dati di caratteri nel contenuto del messaggio quando il client XMS inoltra il messaggio alla destinazione desiderata. In XMS questa proprietà ha un valore numerico e si associa a CCSID. Tuttavia, questa proprietà si basa su una proprietà JMS per cui dispone di un valore di tipo stringa ed esegue l'associazione al set di caratteri Java che rappresenta questo CCSID numerico. Questa proprietà sovrascrive qualsiasi CCSID specificato per la destinazione dalla proprietà [XMSC_WMQ_CCSD](#).

Per default, la proprietà non è impostata.

Questa proprietà non è rilevante quando un'applicazione si connette a un SIB (service integration bus).

JMS_IBM_Encoding

Tipo di dati:

`System.Int32`

Proprietà di:

Messaggio

Modalità di rappresentazione dei dati numerici nel corpo del messaggio quando il client XMS inoltra il messaggio alla destinazione desiderata. Questa proprietà sovrascrive qualsiasi codifica specificata per la destinazione dalla proprietà [XMSC_WMQ_ENCODING](#). La proprietà specifica la rappresentazione di numeri interi binari, interi decimali compressi e numeri a virgola mobile.

I valori validi della proprietà sono gli stessi valori che possono essere specificati nel campo **Encoding** di un descrittore di messaggi.

Un'applicazione può utilizzare le seguenti costanti denominate per impostare la proprietà:

Costante con nome	Significato
<code>MQENC_INTEGER_NORMAL</code>	Codifica numero intero normale
<code>MQENC_INTEGER_REVERSED</code>	Codifica numero intero inversa
<code>MQENC_DECIMAL_NORMAL</code>	Codifica decimale compresso normale
<code>MQENC_DECIMAL_REVERSED</code>	Codifica decimale compresso inversa
<code>MQENC_FLOAT_IEEE_NORMAL</code>	Codifica a virgola mobile IEEE normale
<code>MQENC_FLOAT_IEEE_REVERSED</code>	Codifica a virgola mobile IEEE inversa
<code>MQENC_FLOAT_S390</code>	Codifica a virgola mobile dell'architettura z/OS
<code>MQENC_NATIVE</code>	Codifica macchina nativa

Per formare un valore per la proprietà, l'applicazione può aggiungere tre di queste costanti come segue:

- Una costante il cui nome inizia con MQENC_INTEGER, per specificare la rappresentazione di numeri interi binari
- Una costante il cui nome inizia con MQENC_DECIMAL, per specificare la rappresentazione di numeri interi decimali compressi
- Una costante il cui nome inizia con MQENC_FLOAT, per specificare la rappresentazione dei numeri a virgola mobile

In alternativa, l'applicazione può impostare la proprietà su MQENC_NATIVE, il cui valore dipende dall'ambiente.

Per default, la proprietà non è impostata.

Questa proprietà non è rilevante quando un'applicazione si connette a un SIB (service integration bus).

MESSAGGIO_IBM_JMS

Tipo di dati:

Stringa

Proprietà di:

Messaggio

Testo che descrive perché il messaggio è stato inviato alla destinazione eccezioni. Questa proprietà è di sola lettura.

Questa proprietà è rilevante solo quando un'applicazione si connette a un SIB (service integration bus) e riceve un messaggio da una destinazione eccezioni.

JMS_IBM_EXCEPTIONPROBLEMDESTINATION

Tipo di dati:

Stringa

Proprietà di:

Messaggio

Il nome della destinazione in cui si trovava il messaggio prima che il messaggio venisse inviato alla destinazione eccezioni.

Questa proprietà è rilevante solo quando un'applicazione si connette a un SIB (service integration bus) e riceve un messaggio da una destinazione eccezioni.

JMS_IBM_EXCEPTIONREASON

Tipo di dati:

System.Int32

Proprietà di:

Messaggio

Un codice motivo che indica il motivo per cui il messaggio è stato inviato alla destinazione eccezioni.

Questa proprietà è rilevante solo quando un'applicazione si connette a un SIB (service integration bus) e riceve un messaggio da una destinazione eccezioni.

JMS_IBM_EXCEPTIONTIMESTAMP

Tipo di dati:

System.Int64

Proprietà di:

Messaggio

L'ora in cui il messaggio è stato inviato alla destinazione eccezioni.

L'ora è espressa in millisecondi a partire dalle 00:00:00 GMT del 1 ° gennaio 1970.

Questa proprietà è rilevante solo quando un'applicazione si connette a un SIB (service integration bus) e riceve un messaggio da una destinazione eccezioni.

JMS_IBM_FEEDBACK

Tipo di dati:

System.Int32

Proprietà di:

Messaggio

Un codice che indica la natura di un messaggio di report.

I valori validi della proprietà sono i codici di feedback e i codici motivo che possono essere specificati nel campo **Feedback** di un descrittore di messaggi.

Per default, la proprietà non è impostata.

JMS_IBM_FORMAT

Tipo di dati:

Stringa

Proprietà di:

Messaggio

La natura dei dati dell'applicazione nel messaggio

I valori validi della proprietà sono gli stessi valori che possono essere specificati nel campo **Format** di un descrittore di messaggi.

Per default, la proprietà non è impostata.

Questa proprietà non è rilevante quando un'applicazione si connette a un SIB (service integration bus).

JMS_IBM_LAST_MSG_IN_GROUP

Tipo di dati:

System.Boolean

Proprietà di:

Messaggio

Indicare se il messaggio è l'ultimo messaggio in un gruppo di messaggi.

Impostare la proprietà su true se il messaggio è l'ultimo messaggio in un gruppo di messaggi. Altrimenti, impostare la proprietà su false o non impostare la proprietà. Per default, la proprietà non è impostata.

Il valore true corrisponde all'indicatore di stato MQMF_LAST_MSG_IN_GROUP, che può essere specificato nel campo **MsgFlags** di un descrittore di messaggi.

Questa proprietà viene ignorata nel dominio Pubblicazione/sottoscrizione e non è rilevante quando un'applicazione si connette a un SIB (service integration bus).

TIPO_IBM_JMS

Tipo di dati:

System.Int32

Proprietà di:

Messaggio

Il tipo del messaggio.

Di seguito sono riportati i valori validi della proprietà:

Valore valido	Significato
MQMT_DATAGRAM	Il messaggio non richiede una risposta.
MQMT_REQUEST	Il messaggio richiede una risposta.
MQMT_REPLY	Il messaggio è un messaggio di risposta.
REPORT MQMT	Il messaggio è un messaggio di report.

Questi valori corrispondono ai tipi di messaggi che è possibile specificare nel campo **MsgType** di un descrittore di messaggi.

Per default, la proprietà non è impostata.

Questa proprietà non è rilevante quando un'applicazione si connette a un SIB (service integration bus).

TIPO_PUTAPPL_IBM_JMS

Tipo di dati:
System.Int32

Proprietà di:
Messaggio

Il tipo di applicazione che ha inviato il messaggio.

I valori validi della proprietà sono i tipi di applicazione che è possibile specificare nel campo **PutApp1Type** di un descrittore di messaggi.

Per default, la proprietà non è impostata.

Questa proprietà non è rilevante quando un'applicazione si connette a un SIB (service integration bus).

DATA_IBM_JMS

Tipo di dati:
Stringa

Proprietà di:
Messaggio

La data in cui il messaggio è stato inviato.

I valori validi della proprietà sono gli stessi valori che possono essere specificati nel campo **PutDate** di un descrittore di messaggi.

Per default, la proprietà non è impostata.

Questa proprietà non è rilevante quando un'applicazione si connette a un SIB (service integration bus).

PUTORA_IBM_JMS

Tipo di dati:
Stringa

Proprietà di:
Messaggio

L'ora in cui il messaggio è stato inviato.

I valori validi della proprietà sono gli stessi valori che possono essere specificati nel campo **PutTime** di un descrittore di messaggi.

Per default, la proprietà non è impostata.

Questa proprietà non è rilevante quando un'applicazione si connette a un SIB (service integration bus).

COA IBM_REPORT_JMS

Tipo di dati:

System.Int32

Proprietà di:

Messaggio

Richiedere i messaggi di report di 'conferma all'arrivo', specificando quanti dati dell'applicazione dal messaggio originale devono essere inclusi in un messaggio di report.

Di seguito sono riportati i valori validi della proprietà:

Valore valido	Significato
COA MQRO	Richiedere i messaggi di report 'conferma all'arrivo', senza dati dell'applicazione dal messaggio originale inclusi in un messaggio di report.
DATA_COA_WITH_MQRO	Richiedere i messaggi di report 'conferma all'arrivo', con i primi 100 byte di dati dell'applicazione dal messaggio originale inclusi in un messaggio di report.
DATI_COA_WITH_MQRO_FULL_DATA	Richiedere i messaggi di report 'conferma all'arrivo', con tutti i dati dell'applicazione del messaggio originale inclusi in un messaggio di report.

Questi valori corrispondono alle opzioni di report che possono essere specificate nel campo **Report** di un descrittore di messaggi. Per ulteriori informazioni su queste opzioni, consultare [Report \(MQLONG\)](#).

Per default, la proprietà non è impostata.

COD IBM REPORT_JMS

Tipo di dati:

System.Int32

Proprietà di:

Messaggio

Richiedere i messaggi di report di 'conferma alla consegna', specificando quanti dati dell'applicazione dal messaggio originale devono essere inclusi in un messaggio di report.

Di seguito sono riportati i valori validi della proprietà:

Valore valido	Significato
COD MQRO	Richiedere i messaggi di report 'conferma alla consegna', senza dati dell'applicazione dal messaggio originale inclusi in un messaggio di report.
DATI MQRO_COD_WITH_	Richiedere i messaggi di report 'conferma alla consegna', con i primi 100 byte di dati dell'applicazione dal messaggio originale inclusi in un messaggio di report.
DAD_COD MQRO_WITH_FULL_DATA	Richiedere i messaggi di report 'conferma alla consegna', con tutti i dati dell'applicazione dal messaggio originale inclusi in un messaggio di report.

Questi valori corrispondono alle opzioni di report che è possibile specificare nel campo **Report** di un descrittore di messaggi.

Per default, la proprietà non è impostata.

JMS_IBM_REPORT_DISCARD_MSG

Tipo di dati:

System.Int32

Proprietà di:

Messaggio

Richiedere che il messaggio venga eliminato se non può essere consegnato alla sua destinazione prevista.

Impostare la proprietà su MQRO_DISCARD_MSG per richiedere che il messaggio venga eliminato se non può essere consegnato alla destinazione prevista. Se invece si richiede che il messaggio venga inserito in una coda di messaggi non recapitabili o inviato a una destinazione eccezioni, non impostare la proprietà. Per default, la proprietà non è impostata.

Il valore MQRO_DISCARD_MSG corrisponde a un'opzione di report che può essere specificata nel campo **Report** di un descrittore di messaggi.

JMS_IBM_REPORT_EXCEPTION

Tipo di dati:

System.Int32

Proprietà di:

Messaggio

Richiedere dei messaggi di report di eccezione, specificando quanti dati dell'applicazione dal messaggio originale devono essere inclusi in un messaggio di report.

Di seguito sono riportati i valori validi della proprietà:

Valore valido

MQRO_ECCEZIONE

MQRO_EXCEPTION_WITH_DATA

MQRO_EXCEPTION_WITH_FULL_DATA

Significato

Richiedere i messaggi di report di eccezioni, senza dati dell'applicazione dal messaggio originale inclusi in un messaggio di report.

Richiedere i messaggi di report di eccezione, con i primi 100 byte di dati dell'applicazione dal messaggio originale inclusi in un messaggio di report.

Richiedere i messaggi di report di eccezione, con tutti i dati dell'applicazione dal messaggio originale inclusi in un messaggio di report.

Questi valori corrispondono alle opzioni di report che è possibile specificare nel campo **Report** di un descrittore di messaggi.

Per default, la proprietà non è impostata.

SCADENZA_REPORT_IBM_JMS

Tipo di dati:

System.Int32

Proprietà di:

Messaggio

Richiedere dei messaggi di report di scadenza, specificando quanti dati dell'applicazione dal messaggio originale devono essere inclusi in un messaggio di report.

Di seguito sono riportati i valori validi della proprietà:

Valore valido

SCADENZA_MQRO

Significato

Richiedere i messaggi di report di scadenza, senza dati dell'applicazione dal messaggio originale inclusi in un messaggio di report.

MQRO_EXPIRATION_WITH_DATA

Messaggi di report di scadenza della richiesta, con i primi 100 byte di dati dell'applicazione dal messaggio originale inclusi in un messaggio di report.

MQRO_EXPIRATION_WITH_FULL_DATA

Richiedere i messaggi di report di scadenza, con tutti i dati dell'applicazione del messaggio originale inclusi in un messaggio di report.

Questi valori corrispondono alle opzioni di report che è possibile specificare nel campo **Report** di un descrittore di messaggi.

Per default, la proprietà non è impostata.

NAN IBM REPORT_JMS**Tipo di dati:**

System.Int32

Proprietà di:

Messaggio

Richiedere i messaggi di report di notifica di azione non eseguita con esito positivo.

Impostare la proprietà su MQRO_NAN per richiedere i messaggi di report di notifica azione negativa. Se non si richiedono messaggi di report di notifica azione negativa, non impostare la proprietà. Per default, la proprietà non è impostata.

Il valore MQRO_NAN corrisponde ad un'opzione di report che può essere specificata nel campo **Report** di un descrittore di messaggi.

PAN IBM REPORT_JMS**Tipo di dati:**

System.Int32

Proprietà di:

Messaggio

Richiedere i messaggi di report di notifica di azione eseguita con esito positivo.

Impostare la proprietà su MQRO_PAN per richiedere i messaggi di report di notifica azione positiva. Se non si richiedono messaggi di report di notifica azione positiva, non impostare la proprietà. Per default, la proprietà non è impostata.

Il valore MQRO_PAN corrisponde ad un'opzione di report che può essere specificata nel campo **Report** di un descrittore di messaggi.

ID_CORREL_PASS IBM REPORT_JMS**Tipo di dati:**

System.Int32

Proprietà di:

Messaggio

Richiedere che l'identificativo correlazioni di qualsiasi messaggio di report o di risposta sia uguale all'identificativo correlazioni del messaggio originale.

Di seguito sono riportati i valori validi della proprietà:

Valore valido

ID CORREL_PASS_MQRO_

Significato

Richiedere che l'identificativo correlazioni di qualsiasi messaggio di report o di risposta sia uguale all'identificativo correlazioni del messaggio originale.

ID_COPY_MQRO_MSG_TO_CORREL_ID

Richiedere che l'identificativo di correlazione di qualsiasi report o messaggio di risposta sia uguale all'identificativo del messaggio originale.

Questi valori corrispondono alle opzioni di report che è possibile specificare nel campo **Report** di un descrittore di messaggi.

Il valore predefinito della proprietà è MQRO_COPY_MSG_ID_TO_CORREL_ID.

ID_IBM_REPORT_PASS_MSG_JMS**Tipo di dati:**

System.Int32

Proprietà di:

Messaggio

Richiedere che l'identificativo del messaggio di qualsiasi messaggio di report o di risposta sia uguale all'identificativo del messaggio originale.

Di seguito sono riportati i valori validi della proprietà:

Valore valido

MQRO_PASS_MSG_ID

Significato

Richiedere che l'identificativo del messaggio di qualsiasi messaggio di report o di risposta sia uguale all'identificativo del messaggio originale.

ID_MSG_NEW_MQRO

Richiedere che venga generato un nuovo identificativo di messaggio per ogni report o messaggio di risposta.

Tali valori corrispondono alle opzioni del prospetto che è possibile specificare nel campo **Prospetto** di un descrittore di messaggi.

Il valore predefinito della proprietà è MQRO_NEW_MSG_ID.

RETAIN IBM_JMS**Tipo di dati:**

System.Int32

Proprietà di:

Messaggio

L'impostazione di questa proprietà indica al gestore code di trattare un messaggio come Pubblicazione conservata. Quando un sottoscrittore riceve messaggi dagli argomenti, potrebbe ricevere ulteriori messaggi immediatamente dopo la sottoscrizione, oltre ai messaggi ricevuti nelle release precedenti. Questi messaggi sono le pubblicazioni facoltative conservate per gli argomenti sottoscritti. Per ogni argomento che corrisponde alla sottoscrizione, se è presente una pubblicazione conservata, la pubblicazione viene resa disponibile per la consegna al destinatario del messaggio di sottoscrizione.

RETAIN_PUBLICATION è l'unico valore valido per questa proprietà. Per default questa proprietà non è impostata.

Nota: Questa proprietà è rilevante solo nel dominio di pubblicazione / sottoscrizione

IDMESSAGGIO_IBM_SISTEMA_JMS

Tipo di dati:

Stringa

Proprietà di:

Messaggio

Un identificativo che identifica il messaggio in modo univoco all'interno del bus di integrazione del servizio. Questa proprietà è di sola lettura.

Questa proprietà è rilevante solo quando un'applicazione si connette a un SIB (service integration bus).

IDAPP_JMSX

Tipo di dati:

Stringa

Proprietà di:

Messaggio

Il nome dell'applicazione che ha inviato il messaggio.

Questa proprietà è la proprietà JMS definita con il JMS nome JMSXAppID. Per ulteriori informazioni sulla proprietà, consultare *Java Message Service Specification, Version 1.1*.

Per default, la proprietà non è impostata.

Questa proprietà non è valida per una connessione in tempo reale a un broker.

CONTEGGIO_DISTRIBUZIONE_JMSX

Tipo di dati:

System.Int32

Proprietà di:

Messaggio

Il numero di tentativi di consegna del messaggio.

Questa proprietà è la proprietà JMS definita con il JMS nome JMSXDeliveryCount. Per ulteriori informazioni sulla proprietà, consultare *Java Message Service Specification, Version 1.1*.

Per default, la proprietà non è impostata.

Questa proprietà non è valida per una connessione in tempo reale a un broker.

IDGRUPPO_JMS

Tipo di dati:

Stringa

Proprietà di:

Messaggio

L'identificativo del gruppo messaggi a cui appartiene il messaggio.

Questa proprietà è la proprietà JMS definita con il JMS nome JMSXGroupID. Per ulteriori informazioni sulla proprietà, consultare *Java Message Service Specification, Version 1.1*.

Per default, la proprietà non è impostata.

Questa proprietà non è valida per una connessione in tempo reale a un broker.

GROUPSEQ JMS

Tipo di dati:

System.Int32

Proprietà di:

Messaggio

Il numero di sequenza del messaggio all'interno di un gruppo messaggi.

Questa proprietà è la proprietà JMS definita con il JMS nome JMSXGroupSeq. Per ulteriori informazioni sulla proprietà, consultare *Java Message Service Specification, Version 1.1*.

Per default, la proprietà non è impostata.

Questa proprietà non è valida per una connessione in tempo reale a un broker.

IDUSER_JMS**Tipo di dati:**

Stringa

Proprietà di:

Messaggio

L'identificativo utente associato all'applicazione che ha inviato il messaggio.

Questa proprietà è la proprietà JMS definita con il JMS nome JMSXUserID. Per ulteriori informazioni sulla proprietà, consultare *Java Message Service Specification, Version 1.1*.

Per default, la proprietà non è impostata.

Questa proprietà non è valida per una connessione in tempo reale a un broker.

XMSC_ASYNC_EXCEPTIONS**Tipo di dati:**

System.Int32

Proprietà di:

ConnectionFactory

Oggetti applicabili:

Nome completo strumento di gestione JMS: ASYNCEXCEPTION

Nome breve dello strumento di amministrazione JMS: AEX

Questa proprietà determina se XMS informa un ExceptionListener solo quando una connessione viene interrotta o quando si verifica qualsiasi eccezione in modo asincrono per una chiamata API XMS.

Questa proprietà si applica a tutte le connessioni create da questa ConnectionFactory che hanno un ExceptionListener registrato.

I valori validi per questa proprietà sono:

XMSC_ASYNC_EXCEPTIONS_ALL

Qualsiasi eccezione rilevata in modo asincrono, al di fuori dell'ambito di una chiamata API sincrona e tutte le eccezioni di connessione interrotta vengono inviate a ExceptionListener.

XMSC_ASYNC_EXCEPTIONS_CONNECTIONBROKEN

Solo le eccezioni che indicano una connessione interrotta vengono inviate a ExceptionListener.

Tutte le altre eccezioni che si verificano durante l'elaborazione asincrona non vengono notificate a ExceptionListener quindi l'applicazione non viene informata di tali eccezioni.

Per impostazione predefinita, questa proprietà è impostata su XMSC_ASYNC_EXCEPTIONS_ALL.

ID_CLI_XMSC**Tipo di dati:**

Stringa

Proprietà di:

ConnectionFactory

Oggetti applicabili:

Nome esteso strumento di amministrazione JMS: CLIENTID

Nome breve strumento di amministrazione JMS: CID

L'identificativo client per una connessione.

Un ID client viene utilizzato solo per supportare sottoscrizioni durevoli nel dominio Pubblicazione/ sottoscrizione e viene ignorato nel dominio Point-to-point . Per ulteriori informazioni sull'impostazione degli ID client, consultare [“ConnectionFactoryes e oggetti Connection”](#) a pagina 21.

Questa proprietà non è rilevante per una connessione in tempo reale a un broker.

XMSC_CONNECTION_TYPE

Tipo di dati:

System.Int32

Proprietà di:

ConnectionFactory

Il tipo di server di messaggistica a cui si connette un'applicazione.

Di seguito sono riportati i valori validi della proprietà:

Valore valido	Significato
RTT CT_XMSC	Una connessione in tempo reale a un broker.
CT_XMSC - WMQ	Una connessione a un gestore code WebSphere MQ .
WPM CT_XMSC	Una connessione a un SIB (Service Integration Bus) WebSphere .

Per default, la proprietà non è impostata.

MODALITÀ_DISTRIBUZIONE_XMSC_

Tipo di dati:

System.Int32

Proprietà di:

Destination

Nome utilizzato in un URI:

persistenza (per una destinazione WebSphere MQ)

deliveryMode (per una destinazione del provider di messaggistica predefinita WebSphere)

Oggetti applicabili:

Nome lungo strumento di amministrazione JMS: PERSISTENCE

Nome breve strumento di amministrazione JMS: PER

La modalità di consegna del messaggio inviato alla destinazione.

Di seguito sono riportati i valori validi della proprietà:

Valore valido	Significato
XMSC_DELIVERY_NON_PERSISTENTE	Un messaggio inviato alla destinazione è provvisorio. La modalità di consegna predefinita del mittente del messaggio o qualsiasi modalità di consegna specificata nella chiamata di invio viene ignorata. Se la destinazione è una coda WebSphere MQ , anche il valore dell'attributo della coda <i>DefPersistence</i> viene ignorato.

Valore valido

XMSC_DELIVERY_PERSISTENT

Significato

Un messaggio inviato alla destinazione è permanente. La modalità di consegna predefinita del mittente del messaggio o qualsiasi modalità di consegna specificata nella chiamata di invio viene ignorata. Se la destinazione è una coda WebSphere MQ , anche il valore dell'attributo della coda *DefPersistence* viene ignorato.

XMSC_DELIVERY_AS_APP

Un messaggio inviato alla destinazione ha la modalità di consegna specificata nella chiamata di invio. Se la chiamata di invio non specifica alcuna modalità di consegna, viene utilizzata la modalità di distribuzione predefinita del produttore del messaggio. Se la destinazione è una coda WebSphere MQ , il valore dell'attributo della coda *DefPersistence* viene ignorato.

XMSC_DELIVERY_AS_DEST

Se la destinazione è una coda WebSphere MQ , un messaggio inserito nella coda ha la modalità di consegna specificata dal valore dell'attributo della coda *DefPersistence*. La modalità di consegna predefinita del mittente del messaggio o qualsiasi modalità di consegna specificata nella chiamata di invio viene ignorata.

Se la destinazione non è una coda WebSphere MQ , il significato è lo stesso di XMSC_DELIVERY_AS_APP.

Il valore predefinito è XMSC_DELIVERY_AS_APP.

URL PROVIDER_IC_XMSC_**Tipo di dati:**

Stringa

Proprietà di:

InitialContext

Utilizzata per individuare la directory di denominazione JNDI in modo che il servizio di denominazione COS non debba necessariamente trovarsi sullo stesso server del servizio web.

AUTENTICAZIONE_SICUREZZA_IC_XMSC_**Tipo di dati:**

Stringa

Proprietà di:

InitialContext

Basato su SECURITY_AUTHENTICATION dell'interfaccia di contesto Java . Questa proprietà è applicabile solo al contesto di denominazione COS.

CREDENZIALI_XMSC_IC_SECURITY_CREDENTIALS**Tipo di dati:**

Stringa

Proprietà di:

InitialContext

Basato sull'interfaccia di contesto Java SECURITY_CREDENTIALS. Questa proprietà è applicabile solo al contesto di denominazione COS.

PRINCIPALE_SICUREZZA_ICA_XMSC_

Tipo di dati:

Stringa

Proprietà di:

InitialContext

In base all'interfaccia di contesto Java SECURITY_PRINCIPAL. Questa proprietà è applicabile solo al contesto di denominazione COS.

PROTOCOLLO_SICUREZZA_IC_XMSC_

Tipo di dati:

Stringa

Proprietà di:

InitialContext

Basato su Java SECURITY_PROTOCOL dell'interfaccia del contesto Questa proprietà è applicabile solo al contesto di denominazione COS.

URL IC_XMSC

Tipo di dati:

Stringa

Proprietà di:

InitialContext

Per i contesti LDAP e FileSystem, l'indirizzo del repository che contiene gli oggetti amministrati.

Per i contesti di denominazione COS, l'indirizzo del servizio web che cerca gli oggetti nella directory.

XMSC_IS_SUBSCRIPTION_MULTICAST

Tipo di dati:

System.Boolean

Proprietà di:

MessageConsumer

Indica se i messaggi vengono consegnati al consumatore di messaggi utilizzando WebSphere MQ Multicast Transport. Questa proprietà è di sola lettura.

Il valore della proprietà è true se i messaggi vengono consegnati al consumatore di messaggi utilizzando WebSphere MQ Multicast Transport. Altrimenti, il valore è false.

Questa proprietà è rilevante solo per una connessione in tempo reale a un broker.

XMSC_IS_SUBSCRIPTION_RELIABLE_MULTICAST

Tipo di dati:

System.Boolean

Proprietà di:

MessageConsumer

Indica se i messaggi vengono consegnati al consumatore di messaggi utilizzando WebSphere MQ Multicast Transport con una QoS (quality of service) affidabile. Questa proprietà è di sola lettura.

Il valore della proprietà è true se i messaggi vengono consegnati al consumatore di messaggi utilizzando WebSphere MQ Multicast Transport con una qualità di servizio affidabile. Altrimenti, il valore è false.

Questa proprietà è rilevante solo per una connessione in tempo reale a un broker.

XMSC_JMS_MAJOR_VERSION

Tipo di dati:

System.Int32

Proprietà di:

Dati ConnectionMeta

Il numero di versione principale della specifica JMS su cui si basa XMS . Questa proprietà è di sola lettura.

VERSIONE_MINORE_JMS_XMSC_XX_ENCODE_CASE_CAPS_LOCK_OFF

Tipo di dati:

System.Int32

Proprietà di:

Dati ConnectionMeta

Il numero di versione minore della specifica JMS su cui si basa XMS . Questa proprietà è di sola lettura.

VERSIONE_JMS_XMSC

Tipo di dati:

Stringa

Proprietà di:

Dati ConnectionMeta

L'identificativo della versione della specifica JMS su cui si basa XMS . Questa proprietà è di sola lettura.

VERSIONE_MAGGIORE_XMSC_

Tipo di dati:

System.Int32

Proprietà di:

Dati ConnectionMeta

Il numero di versione del client XMS . Questa proprietà è di sola lettura.

XMSC_VERSIONE_MINORE

Tipo di dati:

System.Int32

Proprietà di:

Dati ConnectionMeta

Il numero di release del client XMS . Questa proprietà è di sola lettura.

PASSWORD_XMSC_

Tipo di dati:


Array di byte

Proprietà di:

ConnectionFactory

Una password che può essere utilizzata per autenticare l'applicazione quando prova a connettersi a un server di messaggistica. La password viene utilizzata con la proprietà XMSC_USERID .

Per default, la proprietà non è impostata.

 Se ci si connette a WebSphere MQ su Multiplatforme si imposta la proprietà XMSC_USERID della factory di connessione, deve corrispondere al **userid** dell'utente collegato. Se non si impostano queste proprietà, il gestore code utilizza il **userid** dell'utente collegato per impostazione predefinita. Se si richiede un'ulteriore autenticazione a livello di connessione di singoli utenti, è possibile scrivere un'uscita di autenticazione client configurata in WebSphere MQ. Per ulteriori informazioni

sulla creazione di un'uscita di autenticazione client, consultare l'argomento Autenticazione nel manuale WebSphere MQ Client.

Per autenticare l'utente durante la connessione a WebSphere MQ on z/OS , è necessario utilizzare un'uscita di sicurezza.

PRIORITÀ_XMSC_

Tipo di dati:

System.Int32

Proprietà di:

Destination

Nome utilizzato in un URI:

priorità

La priorità dei messaggi inviati alla destinazione.

Di seguito sono riportati i valori validi della proprietà:

Valore valido	Significato
Un numero intero compreso nell'intervallo 0, la priorità più bassa, a 9, la priorità più alta	Un messaggio inviato alla destinazione ha la priorità specificata. La priorità predefinita del mittente del messaggio o qualsiasi priorità specificata nella chiamata di invio viene ignorata. Se la destinazione è una coda WebSphere MQ , anche il valore dell'attributo della coda DefPriority viene ignorato.
APPLICAZIONE_PRIORITÀ_XMSC_	Un messaggio inviato alla destinazione ha la priorità specificata nella chiamata di invio. Se la chiamata di invio non specifica alcuna priorità, viene utilizzata la priorità predefinita del mittente del messaggio. Se la destinazione è una coda WebSphere MQ , il valore dell'attributo della coda DefPriority viene ignorato.
XMSC_PRIORITÀ_AS_DEST	Se la destinazione è una coda WebSphere MQ , un messaggio inserito nella coda ha la priorità specificata dal valore dell'attributo della coda DefPriority . La priorità predefinita del mittente del messaggio o qualsiasi priorità specificata nella chiamata di invio viene ignorata. Se la destinazione non è una coda WebSphere MQ , il significato è lo stesso di XMSC_PRIORITY_AS_APP.

Il valore predefinito è XMSC_PRIORITY_AS_APP.

WebSphere MQ Real-time Transport e WebSphere MQ Multicast Transport non intraprendono alcuna azione in base alla priorità di un messaggio.

NOME_PROVIDER_XMSCO

Tipo di dati:

Stringa

Proprietà di:

Dati ConnectionMeta

Il fornitore del client XMS . Questa proprietà è di sola lettura.

XMSC_RTT_BROKER_PING_INTERVAL

Tipo di dati:

System.Int32

Proprietà di:

ConnectionFactory

L'intervallo di tempo, in millisecondi, dopo il quale XMS .NET controlla la connessione a un server di messaggistica in tempo reale per rilevare eventuale attività. Se non viene rilevata alcuna attività, il client avvia un ping; la connessione viene chiusa se non viene rilevata alcuna risposta al ping.

Il valore predefinito della proprietà è 30000.

PROTOCOLLO_COLLEGAMENTO_RTT_XMSC_

Tipo di dati:

System.Int32

Proprietà di:

ConnectionFactory

Il protocollo di comunicazione utilizzato per una connessione in tempo reale a un broker.

Il valore della proprietà deve essere XMSC_RTT_CP_TCP, che indica una connessione in tempo reale a un broker su TCP/IP. Il valore predefinito è XMSC_RTT_CP_TCP.

NOME_HOST_RTT_XMSC_

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Il nome host o l'indirizzo IP del sistema su cui viene eseguito un broker.

Questa proprietà viene utilizzata con la proprietà XMSC_RTT_PORT per identificare il broker.

Per default, la proprietà non è impostata.

XMSC_RTT_LOCAL_ADDRESS

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Il nome host o l'indirizzo IP dell'interfaccia di rete locale da utilizzare per una connessione in tempo reale a un broker.

Questa proprietà è utile solo se il sistema su cui è in esecuzione l'applicazione ha due o più interfacce di rete ed è necessario poter specificare quale interfaccia deve essere utilizzata per una connessione in tempo reale. Se il sistema dispone di una sola interfaccia di rete, è possibile utilizzare solo tale interfaccia. Se il sistema ha due o più interfacce di rete e la proprietà non è impostata, l'interfaccia viene selezionata in modo casuale.

Per default, la proprietà non è impostata.

XMSC_RTT_MULTICAST

Tipo di dati:

System.Int32

Proprietà di:

ConnectionFactory e Destinazione

Nome utilizzato in un URI:

multicast

L'impostazione multicast per una destinazione o una factory di connessione. Solo una destinazione che è un argomento può avere questa proprietà.

Un'applicazione utilizza questa proprietà per abilitare il multicast in associazione con una connessione in tempo reale ad un broker e, se il multicast è abilitato, per specificare il modo preciso in cui il multicast

viene utilizzato per consegnare i messaggi dal broker ad un consumatore di messaggi. La proprietà non ha alcun effetto sul modo in cui un produttore di messaggi invia i messaggi al broker.

Di seguito sono riportati i valori validi della proprietà:

Valore valido

Significato

XMSC_RTT_MULTICAST_DISABLED

I messaggi non vengono consegnati a un consumatore di messaggi utilizzando WebSphere MQ Multicast Transport. Questo è il valore predefinito per un oggetto ConnectionFactory .

XMSC_RTT_MULTICAST_ASCF

I messaggi vengono consegnati ad un consumatore di messaggi in base all'impostazione multicast per la produzione connessioni associata al consumatore di messaggi. L'impostazione multicast per il factory di connessione viene annotato al momento della creazione della connessione. Questo valore è valido solo per un oggetto Destinazione ed è il valore predefinito per un oggetto Destinazione.

XMSC_RTT_MULTICAST_ENABLED

Se l'argomento è configurato per multicast nel broker, i messaggi vengono consegnati a un consumatore di messaggi utilizzando WebSphere MQ Multicast Transport. Viene utilizzata una QoS (quality of service) affidabile se l'argomento è configurato per il multicast affidabile.

XMSC_RTT_MULTICAST_AFFIDABILE

Se l'argomento è configurato per il multicast affidabile nel broker, i messaggi vengono consegnati a un consumatore di messaggi utilizzando WebSphere MQ Multicast Transport con una QoS (quality of service) affidabile. Se l'argomento non è configurato per il multicast affidabile, non è possibile creare un consumer di messaggi per l'argomento.

XMSC_RTT_MULTICAST_NON_AFFIDABILE

Se l'argomento è configurato per multicast nel broker, i messaggi vengono consegnati a un consumatore di messaggi utilizzando WebSphere MQ Multicast Transport. Una QoS (quality of service) affidabile non viene utilizzata anche se l'argomento è configurato per il multicast affidabile.

PORTA_RTT_XMSC

Tipo di dati:

System.Int32

Proprietà di:

ConnectionFactory

Il numero della porta su cui un broker è in ascolto per le richieste in entrata. Sul broker, è necessario configurare un nodo di elaborazione dei messaggi del flusso Real -timeInput o Real -timeOptimizedper l'ascolto su questa porta.

Questa proprietà viene utilizzata con la proprietà [XMSC_RTT_HOST_NAME](#) per identificare il broker.

Il valore predefinito della proprietà è XMSC_RTT_DEFAULT_PORT o 1506.

XMSC_TIME_TO_LIVE

Tipo di dati:

System.Int32

Proprietà di:

Destination

Nome utilizzato in un URI:

scadenza (per una destinazione WebSphere MQ)

timeToLive (per una destinazione del fornitore di messaggistica predefinito WebSphere)

La durata (TTL, Time To Live) per i messaggi inviati alla destinazione.

Di seguito sono riportati i valori validi della proprietà:

Valore valido	Significato
0	Un messaggio inviato alla destinazione non scade mai.
Un numero intero positivo	Un messaggio inviato alla destinazione ha la durata specificata in millisecondi. Il TTL (time to live) predefinito del produttore del messaggio o qualsiasi TTL (time to live) specificato nella chiamata di invio viene ignorato.
XMSC_TIME_TO_LIVE_AS_APP	Un messaggio inviato alla destinazione ha la durata specificata nella chiamata di invio. Se la chiamata di invio non specifica alcuna durata, viene utilizzata la durata predefinita del produttore del messaggio.

Il valore predefinito è XMSC_TIME_TO_LIVE_AS_APP.

IDUSER_XMSC

Tipo di dati:


Stringa


Proprietà di:

ConnectionFactory

Un identificativo utente che può essere utilizzato per autenticare l'applicazione quando prova a connettersi a un server di messaggistica. L'identificativo utente viene utilizzato con la propriet [XMSC_PASSWORD](#).

Per default, la proprietà non è impostata.

 Se ci si connette a IBM MQ for Multiplatformse si imposta la proprietà XMSC_USERID della factory di connessione, deve corrispondere al **userid** dell'utente collegato. Se non si impostano queste proprietà, il gestore code utilizza il **userid** dell'utente collegato per impostazione predefinita. Se si richiede un'ulteriore autenticazione a livello di connessione di singoli utenti, è possibile scrivere un'uscita di autenticazione del client configurata in IBM MQ.

 Per autenticare l'utente durante la connessione a IBM MQ for z/OS è necessario utilizzare un'uscita di sicurezza.

VERSIONE XMSC

Tipo di dati:

Stringa

Proprietà di:

Dati ConnectionMeta

L'identificativo della versione del client XMS . Questa proprietà è di sola lettura.

BROKER_CONTROLQ WMQ_XMSC

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Il nome della coda di controllo utilizzata da un broker.

Il valore predefinito della proprietà è SYSTEM.BROKER.CONTROL.QUEUE.

Questa proprietà è rilevante solo nel dominio Pubblicazione/sottoscrizione .

BROKER_PUBQ WMQ_XMSC

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Il nome della coda monitorata da un broker dove le applicazioni inviano i messaggi che pubblicano.

Il valore predefinito della proprietà è SYSTEM.BROKER.DEFAULT.STREAM.

Questa proprietà è rilevante solo nel dominio Pubblicazione/sottoscrizione .

Gestore code BROKER_WMQ_XMSC

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Il nome del gestore code a cui è connesso un broker.

Per default, la proprietà non è impostata.

Questa proprietà è rilevante solo nel dominio Pubblicazione/sottoscrizione .

BROKER_SUBQ WMQ_XMSC

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Il nome della coda sottoscrittore per un utilizzatore di messaggi non durevole.

Il nome della coda del sottoscrittore deve iniziare con i seguenti caratteri:

SYSTEM.JMS.ND.

Se si desidera che tutti gli utenti di messaggi non durevoli condividano una coda del sottoscrittore, specificare il nome completo della coda condivisa. Una coda con il nome specificato deve esistere prima che un'applicazione possa creare un consumer di messaggi non durevoli.

Se si desidera che ogni utente di messaggi non durevoli richiami i messaggi dalla propria coda di sottoscrittore esclusiva, specificare un nome coda che termini con un asterisco (*). Quindi, quando un'applicazione crea un consumer di messaggi non duraturo, il client XMS crea una coda dinamica per l'uso esclusivo da parte del consumer di messaggi. Il client XMS utilizza il valore della proprietà per impostare il contenuto del campo **DynamicQName** nel descrittore dell'oggetto utilizzato per creare la coda dinamica.

Il valore predefinito della proprietà è SYSTEM.JMS.ND.SUBSCRIBER.QUEUE, che significa che XMS utilizza l'approccio della coda condivisa per impostazione predefinita.

Questa proprietà è rilevante solo nel dominio Pubblicazione/sottoscrizione .

VERSIONE_BROKER_WMQ_XMSC_

Tipo di dati:

System.Int32

Proprietà di:

ConnectionFactory e Destinazione

Nome utilizzato in un URI:

brokerVersion

Il tipo di broker utilizzato dall'applicazione per una connessione o per la destinazione. Solo una destinazione che è un argomento può avere questa proprietà.

Di seguito sono riportati i valori validi della proprietà:

Valore valido

Significato

XMSC_WMQ_BROKER_V1

L'applicazione utilizza un broker WebSphere MQ Pubblicazione/sottoscrizione .

L'applicazione può utilizzare questo valore anche se si esegue la migrazione da WebSphere MQ Pubblicazione/sottoscrizione a WebSphere Message Broker ma non ha modificato l'applicazione.

XMSC_WMQ_BROKER_V2

L'applicazione utilizza un broker di IBM Integration Bus.

XMSC_WMQ_BROKER non specificato

Dopo la migrazione del broker, impostare questa proprietà in modo che le intestazioni RFH2 non vengano più utilizzate. Dopo la migrazione, questa proprietà non è più rilevante.

Il valore predefinito per una factory di connessione è XMSC_WMQ_BROKER_UNSPECIFIED ma, per impostazione predefinita, la proprietà non è impostata per una destinazione. L'impostazione della proprietà per una destinazione sovrascrive qualsiasi valore specificato dalla proprietà del factory di connessione.

URL CCDT WMQ_XMSC

Tipo di dati:

System.String

Proprietà di:

ConnectionFactory

Oggetti applicabili:

Nome esteso strumento di gestione JMS: CCDTURL

Nome breve dello strumento di amministrazione JMS: CCDT

Un URL (A Uniform Resource Locator) che identifica il nome e l'ubicazione del file che contiene la tabella di definizione del canale client e che specifica anche come è possibile accedere al file.

Per impostazione predefinita, questa proprietà non è impostata.

CCSID WMQ_XMSC

Tipo di dati:

System.Int32

Proprietà di:

Destination

Nome utilizzato in un URI:

CCSID

L'identificativo (CCSID) della serie di caratteri codificati, o codepage, in cui si trovano le stringhe di dati carattere nel corpo di un messaggio quando il client XMS inoltra il messaggio alla destinazione. Se è impostata per un singolo messaggio, la proprietà JMS_IBM_CHARACTER_SET sovrascrive il CCSID specificato per la destinazione da questa proprietà.

Il valore predefinito della proprietà è 1208.

Questa proprietà è relativa solo ai messaggi inviati alla destinazione, non ai messaggi ricevuti dalla destinazione.

XMSC_WMQ_CHALLENGATO**Tipo di dati:**

Stringa

Proprietà di:

ConnectionFactory

Oggetti applicabili:

Nome esteso dello strumento di amministrazione JMS: CHANNEL

Nome breve dello strumento di amministrazione JMS: CHAN

Il nome del canale da utilizzare per una connessione.

Per default, la proprietà non è impostata.

Questa proprietà è rilevante solo quando un'applicazione si connette a un gestore code in modalità client.

XMSC_WMQ_CLIENT_RECONNECT_OPTIONS**Tipo di dati:**

Stringa

Proprietà di:

ConnectionFactory

Oggetti applicabili:

Nome esteso dello strumento di amministrazione JMS: CLIENTRECONNECTOPTIONS

Nome breve strumento di amministrazione JMS: CROPT

Questa proprietà specifica le opzioni di riconnessione client per le nuove connessioni create da questo factory. Si trova in XMSC ed è uno dei seguenti:

- WMQ_CLIENT_RECONNECT_AS_DEF (predefinito). Utilizzare il valore specificato nel file `mqclient.ini`. Impostare il valore utilizzando la proprietà **DefRecon** nella sezione Canali. Può essere impostato su uno dei seguenti:
 1. Sì. Si comporta come l'opzione WMQ_CLIENT_RECONNECT
 2. No. Predefinito. Non specifica alcuna opzione di riconnessione
 3. QMGR. Si comporta come l'opzione WMQ_CLIENT_RECONNECT_Q_MGR
 4. DISABILITATO. Si comporta come l'opzione WMQ_CLIENT_RECONNECT_DISABLED
- WMQ_CLIENT_RECONNECT. Riconnettersi a uno dei gestori code specificati nell'elenco dei nomi di connessione.
- WMQ_CLIENT_RECONNECT_Q_MGR. Si riconnette allo stesso gestore code a cui è originariamente connesso. Restituisce MQRC_RECONNECT_QMID_MISMATCH se il gestore code a cui tenta di connettersi (specificato nell'elenco dei nomi di connessione) ha un QMID differente per il gestore code a cui è originariamente connesso.
- WMQ_CLIENT_RECONNECT_DISABLED. La riconnessione è disabilitata.

XMSC_WMQ_CLIENT_RECONNECT_TIMEOUT

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Oggetti applicabili:

Nome esteso dello strumento di amministrazione JMS: CLIENTRECONNECTTIMEOUT

Nome breve dello strumento di amministrazione JMS: CRT

La proprietà XMSC_WMQ_CLIENT_RECONNECT_TIMEOUT è valida solo per il client XMS .NET gestito.

Questa proprietà specifica il periodo di tempo, in secondi, in cui una connessione client tenta di riconnettersi.

Dopo aver tentato di riconnettersi per questo periodo di tempo, il client avrà esito negativo con MQRC_RECONNECT_FAILED. L'impostazione predefinita per questa proprietà è XMSC.WMQ_CLIENT_RECONNECT_TIMEOUT_DEFAULT.

Il valore predefinito di questa proprietà è 1800.

XMSC_WMQ_CONNECTION_MODE

Tipo di dati:

System.Int32

Proprietà di:

ConnectionFactory

La modalità in base alla quale un'applicazione si connette a un gestore code.

Di seguito sono riportati i valori validi della proprietà:

Valore valido	Significato
XMSC_WMQ_CM_BINDINGS	Una connessione a un gestore code in modalità bind, per prestazioni ottimali. Questo è il valore predefinito per C/C + +.
XMSC_WMQ_CM_CLIENT	Una connessione a un gestore code in modalità client, per garantire uno stack completamente gestito. Questo valore è il valore predefinito per .NET.
XMSC_WMQ_CM_CLIENT_UNMANAGED (solo per .NET)	Una connessione a un gestore code che forza uno stack client non gestito.

Concetti correlati

Operazioni gestite e non gestite in .NET

Il codice gestito viene eseguito esclusivamente nell'ambiente di runtime del linguaggio comune .NET ed è interamente dipendente dai servizi forniti da tale runtime. Un'applicazione viene classificata come non gestita se una parte dell'applicazione viene eseguita o richiama i servizi all'esterno dell'ambiente di runtime del linguaggio comune .NET .

ELENCO_NOMI_CONNESSIONI_WMQ_XMSC_

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Oggetti applicabili:

Nome esteso strumento di amministrazione JMS: CONNECTIONNAMELIST

Nome breve dello strumento di gestione JMS: CNLIST

Questa proprietà specifica gli host a cui il client tenta di riconnettersi dopo che la connessione è stata interrotta.

L'elenco dei nomi di connessione è un elenco separato da virgole di coppie host / porta ip. L'impostazione predefinita per questa proprietà è WMQ_CONNECTION_NAME_LIST_DEFAULT.

Ad esempio, 127.0.0.1 (1414), host2.example.com(1400)

L'impostazione predefinita di questa proprietà è localhost (1414).

XMSC_WMQ_DUR_SUBQ

Tipo di dati:

Stringa

Proprietà di:

Destination

il nome della coda sottoscrittore per un sottoscrittore durevole che sta ricevendo messaggi dalla destinazione. Solo una destinazione che è un argomento può avere questa proprietà.

Il nome della coda del sottoscrittore deve iniziare con i seguenti caratteri:

SYSTEM.JMS.D.

Se si desidera che tutti i sottoscrittori durevoli condividano una coda del sottoscrittore, specificare il nome completo della coda condivisa. Una coda con il nome specificato deve esistere prima che un'applicazione possa creare un sottoscrittore durevole.

Se si desidera che ogni sottoscrittore durevole richiami i messaggi dalla propria coda di sottoscrittori esclusiva, specificare un nome coda che termini con un asterisco (*). Quindi, quando un'applicazione crea un sottoscrittore duraturo, il client XMS crea una coda dinamica per l'uso esclusivo da parte del sottoscrittore durevole. Il client XMS utilizza il valore della proprietà per impostare il contenuto del campo **DynamicQName** nel descrittore dell'oggetto utilizzato per creare la coda dinamica.

Il valore predefinito della proprietà è SYSTEM.JMS.D.SUBSCRIBER.QUEUE, che significa che XMS utilizza l'approccio della coda condivisa per impostazione predefinita.

Questa proprietà è rilevante solo nel dominio Pubblicazione/sottoscrizione .

XMSC_WMQ_ENCODING

Tipo di dati:

System.Int32

Proprietà di:

Destination

Modalità di rappresentazione dei dati numerici nel corpo di un messaggio quando il client XMS inoltra il messaggio alla destinazione. Se è impostato per un singolo messaggio, la proprietà JMS_IBM_ENCODING sovrascrive la codifica specificata per la destinazione da questa proprietà. La proprietà specifica la rappresentazione di numeri interi binari, interi decimali compressi e numeri a virgola mobile.

I valori validi della proprietà sono uguali ai valori che possono essere specificati nel campo **Encoding** di un descrittore di messaggi.

Un'applicazione può utilizzare le seguenti costanti denominate per impostare la proprietà:

Costante con nome	Significato
MQENC_INTEGER_NORMAL	Codifica numero intero normale
MQENC_INTEGER_REVERSED	Codifica numero intero inversa
MQENC_DECIMAL_NORMAL	Codifica decimale compresso normale
MQENC_DECIMAL_REVERSED	Codifica decimale compresso inversa
MQENC_FLOAT_IEEE_NORMAL	Codifica a virgola mobile IEEE normale

Costante con nome	Significato
MQENC_FLOAT_IEEE_REVERSED	Codifica a virgola mobile IEEE inversa
MQENC_FLOAT_S390	Codifica a virgola mobile dell'architettura z/OS
MQENC_NATIVE	Codifica macchina nativa

Per formare un valore per la proprietà, l'applicazione può aggiungere tre di queste costanti come segue:

- Una costante il cui nome inizia con MQENC_INTEGER, per specificare la rappresentazione di numeri interi binari
- Una costante il cui nome inizia con MQENC_DECIMAL, per specificare la rappresentazione di numeri interi decimali compressi
- Una costante il cui nome inizia con MQENC_FLOAT, per specificare la rappresentazione dei numeri a virgola mobile

In alternativa, l'applicazione può impostare la proprietà su MQENC_NATIVE, il cui valore dipende dall'ambiente.

Il valore predefinito della proprietà è MQENC_NATIVE.

Questa proprietà è relativa solo ai messaggi inviati alla destinazione, non ai messaggi ricevuti dalla destinazione.

XMSC_WMQ_FAIL_IF QUIESCE

Tipo di dati:

System.Int32

Proprietà di:

ConnectionFactory e Destinazione

Nome utilizzato in un URI:

FAILIFQUIESCE

Oggetti applicabili:

Nome completo dello strumento di gestione JMS: FAILIFQUIESCE

Nome breve strumento di amministrazione JMS: FIQ

Indicazione della mancata riuscita delle chiamate a specifici metodi se il gestore code a cui è connessa l'applicazione è in uno stato di non attivo.

Di seguito sono riportati i valori validi della proprietà:

Valore valido	Significato
SÌ WMQ_XMSC	Le chiamate a determinati metodi hanno esito negativo se il gestore code si trova in uno stato di inattività. Quando l'applicazione rileva che il gestore code è in fase di sospensione, può completare l'attività immediata e chiudere la connessione, consentendo l'arresto del gestore code.
N. FIQ WMQ_XMSC	Nessuna chiamata al metodo non riesce perché il gestore code è in uno stato di inattività. Se si specifica questo valore, l'applicazione non è in grado di rilevare che il gestore code è in fase di sospensione. L'applicazione potrebbe continuare ad eseguire operazioni sul gestore code e quindi impedire l'arresto del gestore code.

Il valore predefinito per una factory di connessioni è XMSC_WMQ_FIQ_YES ma, per impostazione predefinita, la proprietà non è impostata per una destinazione. L'impostazione della proprietà per una destinazione sovrascrive qualsiasi valore specificato dalla proprietà del factory di connessione.

CORPO messaggio wmq_xmsc_

Tipo di dati:

System.Int32

Proprietà di:

Destination

Questa proprietà determina se un'applicazione XMS elabora MQRFH2 di un messaggio IBM WebSphere MQ come parte del payload del messaggio (ovvero, come parte del corpo del messaggio).

Nota: Quando si inviano messaggi a una destinazione, la proprietà XMSC_WMQ_MESSAGE_BODY sostituisce la proprietà di destinazione esistente XMS XMSC_WMQ_TARGET_CLIENT.

I valori validi per questa proprietà sono:

XMSC_WMQ_MESSAGE_BODY_JMS

Ricevi: il tipo e il corpo del messaggio XMS in entrata sono determinati dal contenuto di MQRFH2 (se presente) o MQMD (se non è presente alcun MQRFH2) nel messaggio IBM WebSphere MQ ricevuto.

Invia: il corpo del messaggio XMS in uscita contiene un'intestazione MQRFH2 premissa e generata automaticamente basata sulle proprietà e sui campi di intestazione del messaggio XMS .

MESSAGE_BODY_MQ XMSC_WMQ_MQ

Ricevi: il tipo di messaggio XMS in entrata è sempre ByteMessage, indipendentemente dal contenuto del messaggio IBM WebSphere MQ ricevuto o dal campo del formato dell'MQMD ricevuto. Il corpo del messaggio XMS è costituito dai dati del messaggio non modificati restituiti dalla chiamata API del provider di messaggistica sottostante. La serie di caratteri e la codifica dei dati nel corpo del messaggio sono determinati dai campi CodedCharSetId e Codifica di MQMD. Il formato dei dati nel corpo del messaggio è determinato dal campo Formato di MQMD.

Send: il corpo del messaggio XMS in uscita contiene il payload dell'applicazione; e nessuna intestazione IBM WebSphere MQ generata automaticamente viene aggiunta al corpo.

XMSC_WMQ_MESSAGE_BODY_UNSPECIFIED

Ricezione: il client XMS stabilisce un valore adatto per questa proprietà. Nel percorso di ricezione, questo valore è il valore della proprietà WMQ_MESSAGE_BODY_JMS.

Invia: il client XMS determina un valore appropriato per questa proprietà. Nel percorso di invio, questo valore è il valore della proprietà XMSC_WMQ.

Per impostazione predefinita, questa proprietà è impostata su XMSC_WMQ_MESSAGE_BODY_UNSPECIFIED.

XMSC_WMQ_MQMD_MESSAGE_CONTEXT

Tipo di dati:

System.Int32

Proprietà di:

Destination

Determina quale livello di contesto del messaggio deve essere impostato dall'applicazione XMS . L'applicazione deve essere in esecuzione con l'autorizzazione di contesto appropriata perché questa proprietà diventi effettiva.

I valori validi per questa proprietà sono:

XMSC_WMQ_MDCTX_DEFAULT

Per i messaggi in uscita, la chiamata API MQOPEN e la struttura MQPMO non specificano opzioni di contesto del messaggio esplicite.

XMSC_WMQ_MDCTX_SET_IDENTITY_CONTEXT

La chiamata API MQOPEN specifica l'opzione di contesto del messaggio MQOO_SET_IDENTITY_CONTEXT e la struttura di MQPMO specifica MQPMO_SET_IDENTITY_CONTEXT.

XMSC_WMQ_MDCTX_SET_ALL_CONTEXT

La chiamata API MQOPEN specifica l'opzione di contesto del messaggio MQOO_SET_ALL_CONTEXT e la struttura MQPMO specifica MQPMO_SET_ALL_CONTEXT.

Per impostazione predefinita questa proprietà è impostata su XMSC_WMQ_MDCTX_DEFAULT.

Nota: Questa proprietà non è rilevante quando un'applicazione si connette al sistema Integration Bus.

Le seguenti proprietà richiedono che la proprietà XMSC_WMQ_MQMD_MESSAGE_CONTEXT sia impostata sul valore della proprietà XMSC_WMQ_MDCTX_SET_IDENTITY_CONTEXT o sul valore della proprietà XMSC_WMQ_MDCTX_SET_ALL_CONTEXT quando si invia un messaggio per ottenere l'effetto desiderato:

- JMS_IBM_MQMD_USERIDENTIFIER
- JMS_IBM_MQMD_ACCOUNTINGTOKEN
- JMS_IBM_MQMD_APPLIDENTITYDATA

Le seguenti proprietà richiedono che la proprietà XMSC_WMQ_MQMD_MESSAGE_CONTEXT sia impostata sul valore della proprietà XMSC_WMQ_MDCTX_SET_ALL_CONTEXT quando si invia un messaggio per ottenere l'effetto desiderato:

- PUTAPPLTYPE IBM_MQMD_JMS
- NomePUTAPPL_IBM_MQMD_JMS
- PUTDATE MQMD_IBM_JMS
- PUTORA_IBM_MQMD_JMS
- JMS_IBM_MQMD_APPLORIGINDATA

XMSC_WMQ_MQMD_READ_ENABLED

Tipo di dati:

System.Int32

Proprietà di:

Destination

Questa proprietà determina se un'applicazione XMS può estrarre o meno i valori dei campi MQMD.

I valori validi per questa proprietà sono:

XMSC_WMQ_READ_ENABLED_NO

Quando si inviano messaggi, le proprietà JMS_IBM_MQMD* di un messaggio inviato non vengono aggiornate per riflettere i valori dei campi aggiornati in MQMD.

Quando si ricevono i messaggi, nessuna delle proprietà JMS_IBM_MQMD* è disponibile su un messaggio ricevuto, anche se alcune o tutte sono impostate dal mittente.

XMSC_WMQ_READ_ENABLED_SI

Quando si inviano messaggi, tutte le proprietà JMS_IBM_MQMD* di un messaggio inviato vengono aggiornate in modo da riflettere i valori del campo aggiornati in MQMD, incluse quelle proprietà che il mittente non ha impostato esplicitamente.

Quando si ricevono messaggi, tutte le proprietà JMS_IBM_MQMD* sono disponibili su un messaggio ricevuto, incluse quelle che il mittente non ha impostato esplicitamente.

Per impostazione predefinita, questa proprietà è impostata su XMSC_WMQ_READ_ENABLED_NO.

XMSC_WMQ_MQMD_WRITE_ENABLED

Tipo di dati:

System.Int32

Proprietà di:

Destination

Questa proprietà determina se un'applicazione XMS può utilizzare o meno i valori dei campi MQMD.

I valori validi per questa proprietà sono:

XMSC_WMQ_WRITE_ENABLED_NO

Tutte le proprietà JMS_IBM_MQMD* vengono ignorate e i relativi valori non vengono copiati nella struttura MQMD sottostante.

XMSC_WMQ_WRITE_ENABLED_SÌ

Le proprietà JMS_IBM_MQMD* vengono elaborate. I loro valori vengono copiati nella struttura MQMD sottostante.

Per impostazione predefinita, questa proprietà è impostata su XMSC_WMQ_WRITE_ENABLED_NO.

XMSC_WMQ_PUT_ASYNC_ALLOWED

Tipo di dati:

System.Int32

Proprietà di:

Destination

Questa proprietà determina se ai mittenti del messaggio è consentito utilizzare i put asincroni per inviare messaggi a questa destinazione.

I valori validi per questa proprietà sono:

XMSC_WMQ_PUT_ASYNC_ALLOWED_AS_DEST

Determinare se gli inserimenti asincroni sono consentiti facendo riferimento alla definizione dell'argomento o della coda.

XMSC_WMQ_PUT_ASYNC_ALLOWED_AS_Q_DEF

Determinare se gli inserimenti asincroni sono consentiti facendo riferimento alla definizione della coda.

XMSC_WMQ_PUT_ASYNC_ALLOWED_AS_TOPIC_DEF

Determinare se gli inserimenti asincroni sono consentiti facendo riferimento alla definizione dell'argomento.

XMSC_WMQ_PUT_ASYNC_ALLOWED_DISABLED

Le immissioni asincrone non sono consentite.

XMSC_WMQ_PUT_ASYNC_ALLOWED_ENABLED

Le immissioni asincrone sono consentite.

Per impostazione predefinita questa proprietà è impostata su XMSC_WMQ_PUT_ASYNC_ALLOWED_AS_DEST.

Nota: Questa proprietà non è rilevante quando un'applicazione si connette al sistema Integration Bus.

XMSC_WMQ_READ_AHEAD_ALLOWED

Tipo di dati:

System.Int32

Proprietà di:

Destination

Questa proprietà determina se ai destinatari dei messaggi e ai browser della coda è consentito utilizzare la lettura anticipata per ottenere i messaggi non persistenti e non transazionali da questa destinazione in un buffer interno prima di riceverli.

I valori validi per questa proprietà sono:

XMSC_WMQ_READ_AHEAD_ALLOWED_AS_Q_DEF

Determinare se la lettura anticipata è consentita facendo riferimento alla definizione della coda.

XMSC_WMQ_READ_AHEAD_ALLOWED_AS_ARGOMENTO_DEF

Determinare se la lettura anticipata è consentita facendo riferimento alla definizione dell'argomento.

XMSC_WMQ_READ_AHEAD_ALLOWED_AS_DEST

Determinare se la lettura anticipata è consentita facendo riferimento alla definizione dell'argomento o della coda.

XMSC_WMQ_READ_AHEAD_ALLOWED_DISABLED

La lettura anticipata non è consentita durante l'utilizzo o l'esplorazione dei messaggi.

XMSC_WMQ_READ_AHEAD_ALLOWED_ENABLED

La lettura anticipata è consentita.

Per impostazione predefinita, questa proprietà è impostata su XMSC_WMQ_READ_AHEAD_ALLOWED_AS_DEST.

XMSC_WMQ_READ_AHEAD_CLOSE_POLICY

Tipo di dati:

System.Int32

Proprietà di:

Destination

Questa proprietà determina, per i messaggi di cui si sta eseguendo la consegna a un listener di messaggi asincroni, cosa succede ai messaggi nel buffer di lettura anticipata interno quando viene chiuso il destinatario del messaggio.

Questa proprietà è applicabile nella specifica delle opzioni della coda di chiusura quando si utilizzano i messaggi da una destinazione e non è applicabile quando si inviano i messaggi a una destinazione.

Questa proprietà viene ignorata per i browser della coda poiché durante l'esplorazione i messaggi sono ancora disponibili nelle code.

I valori validi per questa proprietà sono:

XMSC_WMQ_READ_AHEAD_CLOSE_POLICY_DELIVER_CURRENT

Solo la chiamata del listener dei messaggi corrente viene completata prima della restituzione, lasciando potenzialmente i messaggi nel buffer di lettura anticipata interno, che vengono quindi eliminati.

XMSC_WMQ_READ_AHEAD_CLOSE_POLICY_DELIVER_ALL

Tutti i messaggi nel buffer di lettura anticipata interno vengono consegnati al listener dei messaggi dell'applicazione prima della restituzione.

Per impostazione predefinita, questa proprietà è impostata su XMSC_WMQ_READ_AHEAD_CLOSE_POLICY_DELIVER_CURRENT.

Nota:

• Terminazione anomala dell'applicazione

Tutti i messaggi nel buffer di lettura anticipata vengono persi quando un'applicazione XMS termina bruscamente.

• Implicazioni sulle transazioni

La lettura anticipata è disabilitata quando le applicazioni utilizzano la transazione. Quindi, l'applicazione non vede alcuna differenza nel comportamento quando utilizzano le sessioni transazionali.

• Implicazioni delle modalità di riconoscimento della sessione

La lettura anticipata è abilitata su una sessione non sottoposta a transazione quando le modalità di riconoscimento sono XMSC_AUTO_ACKNOWLEDGEMENT o XMSC_DUPS_OK_ACKNOWLEDGEMENT. La lettura anticipata è disabilitata se la modalità di riconoscimento della sessione è

XMSC_CLIENT_ACKNOWLEDGEMENT indipendentemente dalle sessioni transazionali o non transazionali.

- **Implicazioni sui browser della coda e sui selettori del browser della coda**

I browser della coda e i selettori del browser della coda, utilizzati in applicazioni XMS , ottengono il vantaggio delle prestazioni dalla lettura anticipata. La chiusura del browser della coda non degrada le prestazioni, poiché il messaggio è ancora disponibile nella coda per ulteriori operazioni. Non vi sono altre implicazioni sui browser delle code e sui selettori dei browser delle code, a parte i vantaggi delle prestazioni della lettura anticipata.

Nome_HOST_WMQ_XMSC

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Oggetti applicabili:

Nome esteso strumento di gestione JMS: HOSTNAME

Nome breve dello strumento di amministrazione JMS: HOST

Il nome host o l'indirizzo IP del sistema su cui viene eseguito un gestore code.

Questa proprietà viene utilizzata solo quando un'applicazione si connette a un gestore code in modalità client. La proprietà viene utilizzata con la proprietà XMSC_WMQ_PORT per identificare il gestore code.

Il valore predefinito della proprietà è localhost.

XMSC_WMQ_LOCAL_ADDRESS

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Oggetti applicabili:

Nome esteso dello strumento di gestione JMS: LOCALADDRESS

Nome breve strumento di amministrazione JMS: LA

Per una connessione a un gestore code, questa proprietà specifica l'interfaccia di rete locale da utilizzare oppure la porta locale o l'intervallo di porte locali da utilizzare oppure entrambe le cose.

Il valore della proprietà è una stringa con il seguente formato:

[*nome_host*] [(*porta_inferiore*) [,*porta_alto*]]

I significati delle variabili sono i seguenti:

nome_host

Il nome host o l'indirizzo IP dell'interfaccia di rete locale da utilizzare per la connessione.

Fornire queste informazioni è necessario solo se il sistema su cui è in esecuzione l'applicazione ha due o più interfacce di rete ed è necessario essere in grado di specificare quale interfaccia deve essere utilizzata per la connessione. Se il sistema dispone di una sola interfaccia di rete, è possibile utilizzare solo tale interfaccia. Se il sistema ha due o più interfacce di rete e non si specifica quale interfaccia deve essere utilizzata, l'interfaccia viene selezionata in modo casuale.

porta_inferiore

Il numero della porta locale da utilizzare per la connessione.

Se viene specificato anche *high_port* , *low_port* viene interpretato come il numero di porta più basso in un intervallo di numeri di porta.

porta_alto

Il numero di porta più alto in un intervallo di numeri di porta. Una delle porte nell'intervallo specificato deve essere utilizzata per la connessione.

La lunghezza massima della stringa è 48 caratteri.

Ecco alcuni esempi di valori validi della proprietà:

JUPITER
9.20.4.98
JUPITER (1000)
9.20.4.98(1000,2000)
(1000)
(1000,2000)

Per default, la proprietà non è impostata.

Questa proprietà è rilevante solo quando un'applicazione si connette a un gestore code in modalità client.

SELEZIONE_MESSAGGIO_WMQ_XMSC_

Tipo di dati:

System.Int32

Proprietà di:

ConnectionFactory

Determina se la selezione del messaggio viene effettuata dal client XMS o dal broker.

Di seguito sono riportati i valori validi della proprietà:

Valore valido	Significato
XMSC_WMQ_MSEL_CLIENT	La selezione del messaggio viene effettuata dal client XMS .
XMSC_WMQ_MSEL_BROKER	La selezione del messaggio viene effettuata dal broker.

Il valore predefinito è XMSC_WMQ_MSEL_CLIENT.

Questa proprietà è rilevante solo nel dominio Pubblicazione/sottoscrizione . La selezione dei messaggi da parte del Broker non è supportata se la proprietà XMSC_WMQ_BROKER_VERSION è impostata su XMSC_WMQ_BROKER_V1.

XMSC_WMQ_MSG_BATCH_SIZE

Tipo di dati:

System.Int32

Proprietà di:

ConnectionFactory

Il numero massimo di messaggi da richiamare da una coda in un singolo batch quando si utilizza la consegna di messaggi asincrona.

Quando un'applicazione utilizza la consegna asincrona dei messaggi, in determinate condizioni, il client XMS richiama un batch di messaggi da una coda prima di inoltrare ogni messaggio singolarmente all'applicazione. Questa proprietà specifica il numero massimo di messaggi che possono trovarsi nel batch.

Il valore della proprietà è un numero intero positivo e il valore predefinito è 10. Impostare la proprietà su un valore diverso solo se si ha un problema di prestazioni specifico che è necessario risolvere.

Se un'applicazione è connessa a un gestore code su una rete, l'aumento del valore di questa proprietà può ridurre i sovraccarichi di rete e i tempi di risposta, ma aumentare la quantità di memoria richiesta per memorizzare i messaggi sul sistema client. Al contrario, l'abbassamento del valore di questa proprietà

potrebbe aumentare i sovraccarichi di rete e i tempi di risposta, ma ridurre la quantità di memoria richiesta per memorizzare i messaggi.

INTERVALLO_POLLING_WMQ_XMSC_

Tipo di dati:

System.Int32

Proprietà di:

ConnectionFactory

Se ciascun listener messaggi in una sessione non dispone di alcun messaggio adatto nella propria coda, questo valore è l'intervallo massimo, in millisecondi, che trascorre prima che ciascun listener messaggi provi di nuovo ad ottenere un messaggio dalla propria coda.

Se si verifica spesso che non è disponibile alcun messaggio adatto per nessuno dei listener di messaggi in una sessione, aumentare il valore di questa proprietà.

Il valore della proprietà è un numero intero positivo. Il valore predefinito è 5000.

PORTA_WMQ_XMSC

Tipo di dati:

System.Int32

Proprietà di:

ConnectionFactory

Oggetti applicabili:

Nome esteso dello strumento di amministrazione JMS: PORT

Nome breve dello strumento di amministrazione JMS: PORT

Il numero della porta su cui un gestore code è in ascolto per le richieste in entrata.

Questa proprietà viene utilizzata solo quando un'applicazione si connette a un gestore code in modalità client. La proprietà viene utilizzata con la proprietà [XMSC](#) per identificare il gestore code.

Il valore predefinito della proprietà è XMSC_WMQ_DEFAULT_CLIENT_PORT o 1414.

XMSC_WMQ_PROVIDER_VERSION

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

La versione, la release, il livello di modifica e il fix pack del gestore code al quale l'applicazione intende connettersi. I valori validi per questa proprietà sono:

- Non specificato

Oppure una stringa in uno dei seguenti formati

- V.R.M.F
- V.R.M
- V.R
- V

dove V, R, M e F sono numeri interi maggiori o uguali a zero.

Un valore pari o superiore a 7 indica che questa versione è intesa come IBM WebSphere MQ 7.0 ConnectionFactory per le connessioni a un gestore code IBM WebSphere MQ 7.0 . Un valore precedente a 7 (ad esempio "6.0.2.0"), indica che è stato progettato per essere utilizzato con gestori code precedenti alla versione 7.0. Il valore predefinito, non specificato, consente le connessioni a qualsiasi livello del

gestore code, determinando le proprietà applicabili e la funzionalità disponibili in base alle capacità del gestore code.

Per impostazione predefinita, questa proprietà è impostata su "non specificato".

Nota:

- Non si verifica alcuna condivisione del socket se XMSC_WMQ_PROVIDER_VERSION è impostato su 6. 2.
- La connessione ha esito negativo se XMSC_WMQ_PROVIDER_VERSION è impostato su 7 e sul server SHARECNV per il canale è impostato su 0.
- Le funzioni specifiche di IBM WebSphere MQ 7.0 sono disabilitate se XMSC_WMQ_PROVIDER_VERSION è impostato su UNSPECIFIED e SHARECNV è impostato su 0.

La versione di IBM WebSphere MQ Client ha anche un ruolo importante nel determinare se un'applicazione client XMS può utilizzare funzioni specifiche di IBM WebSphere MQ 7.0 . La seguente tabella descrive il funzionamento.

Nota: Una proprietà di sistema XMSC_WMQ_OVERRIDEPROVIDERVERSION sovrascrive la proprietà XMSC_WMQ_PROVIDER_VERSION. Questa proprietà può essere utilizzata se non è possibile modificare l'impostazione del factory di connessione.

Tabella 33. client XMS - Possibilità di utilizzare funzioni specifiche di IBM WebSphere MQ 7.0 .

#	XMSC_WMQ_PROVIDER_VERSION	Versione del client IBM WebSphere MQ	IBM WebSphere MQ 7.0Funzioni
1	non specificato	7	SU
2	non specificato	6	NON ATTIVO
3	7	7	SU
4	7	6	Eccezione
5	6	6	NON ATTIVO
6	6	7	NON ATTIVO

XMSC_WMQ_PUB_ACK_INTERVAL

Tipo di dati:

System.Int32

Proprietà di:

ConnectionFactory

Il numero di messaggi pubblicati da un publisher prima che il client XMS richieda un riconoscimento dal broker.

Se si diminuisce il valore di questa proprietà, il client richiede i riconoscimenti più spesso e quindi le prestazioni del publisher diminuiscono. Se si aumenta il valore, il client attende un periodo più lungo prima di inviare un'eccezione in caso di errore del broker.

Il valore della proprietà è un numero intero positivo. Il valore predefinito è 25.

CCSID QMGR XMSC_WMQ_

Tipo di dati:

System.Int32

Proprietà di:

ConnectionFactory

L'identificativo (CCSID) della serie di caratteri codificati, o codepage, in cui i campi di dati carattere definiti in MQI (Message Queue Interface) vengono scambiati tra il client XMS e quello WebSphere MQ . Questa proprietà non si applica alle stringhe di dati carattere nei corpi dei messaggi.

Quando l'applicazione un XMS si connette a un gestore code in modalità client, il client XMS si collega al client WebSphere MQ . Le informazioni scambiate tra i due client contengono campi di dati carattere definiti in MQI. In circostanze normali, il client WebSphere MQ presuppone che questi campi si trovino nella codepage del sistema su cui sono in esecuzione i client. Se il client XMS fornisce e prevede di ricevere questi campi in una codepage differente, è necessario impostare questa proprietà per informare il client WebSphere MQ .

Quando il client WebSphere MQ inoltra questi campi di dati di caratteri al gestore code, i dati in essi contenuti devono essere convertiti, se necessario, nella codepage utilizzata dal gestore code. Allo stesso modo, quando il client WebSphere MQ riceve questi campi dal gestore code, i dati in essi contenuti devono essere convertiti, se necessario, nella codepage in cui il client XMS prevede di ricevere i dati. Il client WebSphere MQ utilizza questa proprietà per eseguire queste conversione dati.

Per default, la proprietà non è impostata.

L'impostazione di questa proprietà equivale all'impostazione della variabile di ambiente MQCCSID per un client WebSphere MQ che supporta le applicazioni client WebSphere MQ native. Per ulteriori informazioni su questa variabile di ambiente, consultare *WebSphere MQ Client*.

XMSC_WMQ_QUEUE_MANAGER

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Oggetti applicabili:

Nome esteso dello strumento di amministrazione JMS: QMANAGER

Nome breve strumento di amministrazione JMS: QMGR

Il nome del gestore code a cui connettersi.

Per default, la proprietà non è impostata.

XMSC_WMQ_RECEIVE_CCSID

Proprietà di destinazione che imposta il CCSID di destinazione per la conversione del messaggio del gestore code. Il valore viene ignorato a meno che XMSC_WMQ_RECEIVE_CONVERSION non sia impostato su WMQ_RECEIVE_CONVERSION_QMGR.

Tipo di dati:

Intero

Valore:

Qualsiasi numero intero positivo.

Il valore predefinito è 1208.

La specifica di un valore GMO_CONVERT in un messaggio è facoltativa. Se viene specificato un valore GMO_CONVERT , la conversione viene eseguita in base al valore specificato.

XMSC_WMQ_RECEIVE_CONVERSIONE

Proprietà di destinazione che determina se la conversione dati verrà eseguita dal gestore code.

Tipo di dati:

Intero

Valori:

XMSC_WMQ_RECEIVE_CONVERSION_CLIENT_MSG (VALORE PREDEFINITO): esegue la conversione dei dati solo sul client XMS . La conversione viene sempre eseguita utilizzando la codepage 1208.

XMSC: eseguire la conversione dei dati sul gestore code prima di inviare un messaggio al client XMS .

XMSC_WMQ_RECEIVE_EXIT

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Identifica un'uscita di ricezione del canale da eseguire.

Il valore della proprietà è una stringa che identifica un'uscita di ricezione del canale e ha il formato seguente:

libraryName(NomeentryPoint)

dove,

- **libraryName** è il percorso completo dell'uscita gestita .dll
- **entryPointName** è il nome della classe qualificato dallo spazio dei nomi

Ad esempio, C:\MyReceiveExit.dll(MyReceiveExitNameSpace.MyReceiveExitClassName)

Per default, la proprietà non è impostata.

Questa proprietà è rilevante solo quando un'applicazione si connette a un gestore code in modalità client gestito. Inoltre, sono supportate solo le uscite gestite.

XMSC_WMQ_RECEIVE_EXIT_INIT

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

I dati utente passati a un'uscita di ricezione del canale quando viene richiamata.

Il valore della proprietà è una stringa. Per default, la proprietà non è impostata.

Questa proprietà è rilevante solo quando un'applicazione si connette ad un gestore code in modalità client gestito e la proprietà "[XMSC_WMQ_RECEIVE_EXIT](#)" a pagina 231 è impostata.

XMSC_WMQ_RESOLVED_QUEUE_MANAGER

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Questa proprietà viene utilizzata per ottenere il nome del gestore code a cui è connesso.

Quando viene utilizzato con CCDT (Client Channel Definition Table), questo nome potrebbe essere diverso dal nome del gestore code specificato nel factory di connessione.

ID_MANAGER_XMSC_WMQ_RESOLVED_QUEUE_

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Questa proprietà viene popolata con l'ID del gestore code dopo la connessione.

XMSC_WMQ_SECURITY_EXIT

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Identifica un'uscita di sicurezza del canale.

Il valore della proprietà è una stringa che identifica un'uscita di sicurezza del canale e ha il formato seguente:

libraryName(NomeentryPoint)

dove,

- **libraryName** è il percorso completo dell'uscita gestita .dll
- **entryPointName** è il nome della classe qualificato dallo spazio dei nomi

Ad esempio, C:\MySecurityExit.dll(MySecurityExitNameSpace.MySecurityExitClassName)

La lunghezza massima della stringa è 128 caratteri.

Per default, la proprietà non è impostata.

Questa proprietà è rilevante solo quando un'applicazione si connette a un gestore code in modalità client gestito. Inoltre, sono supportate solo le uscite gestite.

XMSC_WMQ_SECURITY_EXIT_INIT**Tipo di dati:**

Stringa

Proprietà di:

ConnectionFactory

I dati utente passati a un'uscita di sicurezza del canale quando viene richiamata.

La lunghezza massima della stringa di dati utente è 32 caratteri.

Per default, la proprietà non è impostata.

Questa proprietà è rilevante solo quando un'applicazione si connette ad un gestore code in modalità client gestito e la proprietà "[XMSC_WMQ_SECURITY_EXIT](#)" a pagina 231 è impostata.

XMSC_WMQ_SEND_EXIT**Tipo di dati:**

Stringa

Proprietà di:

ConnectionFactory

Identifica un'uscita di invio del canale.

Il valore della proprietà è una stringa. Un'uscita di invio del canale ha il formato seguente:

libraryName(NomeentryPoint)

dove,

- **libraryName** è il percorso completo dell'uscita gestita .dll
- **entryPointName** è il nome della classe qualificato dallo spazio dei nomi

Ad esempio, C:\MySendExit.dll(MySendExitNameSpace.MySendExitClassName)

Per default, la proprietà non è impostata.

Questa proprietà è rilevante solo quando un'applicazione si connette a un gestore code in modalità client gestito. Inoltre, sono supportate solo le uscite gestite.

XMSC_WMQ_SEND_EXIT_INIT

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

I dati utente passati alle uscite di invio del canale quando vengono richiamate.

Il valore della proprietà è una stringa di uno o più elementi di dati utente separati da virgole. Per default, la proprietà non è impostata.

Le regole per specificare i dati utente che vengono passati a una sequenza di uscite di invio del canale sono le stesse regole per specificare i dati utente che vengono passati a una sequenza di uscite di ricezione del canale. Per le regole, consultare [“XMSC_WMQ_RECEIVE_EXIT_INIT” a pagina 231](#).

Questa proprietà è rilevante solo quando un'applicazione si connette ad un gestore code in modalità client gestito e la proprietà [“XMSC_WMQ_SEND_EXIT” a pagina 232](#) è impostata.

WMQ_XMSC_SEND_CHECK_COUNT

Tipo di dati:

System.Int32

Proprietà di:

ConnectionFactory

Il numero di chiamate di invio da consentire tra i controlli per rilevare eventuali errori di put asincrono all'interno di una singola sessione XML non sottoposta a transazione.

Per impostazione predefinita, questa proprietà è impostata su 0.

XMSC_WMQ_SHARE_CONV_ALLOWED

Tipo di dati:

System.Int32

Proprietà di:

ConnectionFactory

Oggetti applicabili:

Nome lungo strumento di gestione JMS: SHARECONVALLOWED

Nome breve dello strumento di amministrazione JMS: SCALD

Se una connessione client può condividere il proprio socket con altre connessioni XMS di livello superiore dallo stesso processo allo stesso gestore code, se le definizioni di canale corrispondono. Questa proprietà viene fornita per consentire un isolamento completo delle connessioni in socket separati se necessario per motivi correlati allo sviluppo, alla manutenzione e alla gestione delle applicazioni. L'impostazione di questa proprietà indica semplicemente a XMS di rendere condiviso il socket sottostante. Non indica quante connessioni condividono un singolo socket. Il numero di connessioni che condividono un socket viene determinato dal valore SHARECNV negoziato tra WebSphere MQ client e WebSphere MQ server.

Un'applicazione può impostare le seguenti costanti denominate per impostare la proprietà:

- XMSC_WMQ_SHARE_CONV_ALLOWED_FALSE - Le connessioni non condividono un socket.
- XMSC_WMQ_SHARE_CONV_ALLOWED_TRUE - Le connessioni condividono un socket.

Per impostazione predefinita, la proprietà è impostata su XMSC_WMQ_SHARE_CONV_ALLOWED_ENABLED.

Questa proprietà è rilevante solo quando un'applicazione si connette a un gestore code in modalità client.

XMSC_WMQ_SSL_CERT_STORES

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

L'ubicazione dei server che contengono i CRL (cettificate revocation list) da utilizzare su una connessione SSL a un gestore code.

Il valore della proprietà è un elenco di uno o più URL separati da virgole. Ciascun URL ha il formato seguente:

```
[user[/password]@]ldap://[serveraddress][:portnum][, ...]
```

Questo formato è compatibile con, ma esteso da, il formato MQJMS di base.

È valido avere un serveraddressvuoto. In questo caso, XMS presume che il valore sia la stringa "localhost".

Un elenco di esempio è:

```
myuser/mypassword@ldap://server1.mycom.com:389
ldap://server1.mycom.com
ldap://
ldap://:389
```

Solo per .NET : da IBM MQ 8.0, le connessioni gestite a IBM MQ (WMQ_CM_CLIENT) e le connessioni non gestite a IBM MQ (WMQ_CM_CLIENT_UNMANAGED) supportano entrambe le connessioni TLS/SSL.

Per default, la proprietà non è impostata.

Informazioni correlate

[Supporto SSL e TLS per il client .NET non gestito](#)

[Supporto SSL e TLS per il client .NET gestito](#)

XMSC_WMQ_SSL_CIPHER_SPEC**Tipo di dati:**

Stringa

Proprietà di:

ConnectionFactory

Il nome della CipherSpec da utilizzare su una connessione protetta a un gestore code.

Le specifiche di cifratura che puoi utilizzare con il supporto TLS IBM WebSphere MQ sono riportate nella seguente tabella. Quando si richiede un certificato personale, si specifica una dimensione di chiave per la coppia di chiavi pubblica e privata. La dimensione della chiave utilizzata durante l'handshake SSL è la dimensione memorizzata nel certificato a meno che non sia determinata da CipherSpec, come indicato nella tabella. Per impostazione predefinita, questa proprietà non è impostata.

Nome CipherSpec	Protocollo utilizzato	Algoritmo hash	Algoritmo di codifica	Bit di codifica	FIPS ¹	Suite B a 128 bit	Suite B 192 bit
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	SHA-1	AES	128	Sì	No	No
TLS_RSA_WITH_AES_256_CBC_SHA ²	TLS 1.0	SHA-1	AES	256	Sì	No	No
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	SHA-1	DES	56	No	No	No
TLS_RSA_WITH_3DES_EDE_CBC_SHA ⁴	TLS 1.0	SHA-1	3DES	168	Sì	No	No
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	SHA-256	AES	128	Sì	No	No

Nome CipherSpec	Protocollo utilizzato	Algoritmo hash	Algoritmo di codifica	Bit di codifica	FIPS ¹	Suite B a 128 bit	Suite B 192 bit
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	SHA-384	AES	256	Sì	No	No
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	SHA-256	AES	128	Sì	No	No
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	SHA-256	AES	256	Sì	No	No
ECDHE_ECDSA_RC4_128_SHA256	TLS 1.2	SHA-256	RC4	128	No	No	No
ECDHE_ECDSA_3DES_EDE_CBC_SHA256	TLS 1.2	SHA-256	3DES	168	Sì	No	No
ECDHE_RSA_RC4_128_SHA256	TLS 1.2	SHA-256	RC4	128	No	No	No
ECDHE_RSA_3DES_EDE_CBC_SHA256	TLS 1.2	SHA-256	3DES	168	Sì	No	No
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	SHA-256	AES	128	Sì	No	No
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	SHA-384	AES	256	Sì	No	No
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	SHA-256	AES	128	Sì	No	No
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	SHA-384	AES	256	Sì	No	No
ECDHE_ECDSA_AES_128_GCM_SHA256	TLS 1.2	SHA-256	AES	128	Sì	Sì	No
ECDHE_ECDSA_AES_256_GCM_SHA384	TLS 1.2	SHA-384	AES	256	Sì	No	Sì
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	SHA-256	AES	128	Sì	No	No
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	SHA-384	AES	256	Sì	No	No
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	SHA-256	Nessuno	0	No	No	No
ECDHE_RSA_NULL_SHA256	TLS 1.2	SHA-256	Nessuno	0	No	No	No
ECDHE_ECDSA_NULL_SHA256	TLS 1.2	SHA-256	Nessuno	0	No	No	No
TLS_RSA_WITH_NULL_NULL	TLS 1.2	Nessuno	Nessuno	0	No	No	No
TLS_RSA_WITH_RC4_128_SHA256	TLS 1.2	SHA-256	RC4	128	No	No	No

Nome CipherSpec	Protocollo utilizzato	Algoritmo hash	Algoritmo di codifica	Bit di codifica	FIPS ¹	Suite B a 128 bit	Suite B 192 bit
-----------------	-----------------------	----------------	-----------------------	-----------------	-------------------	-------------------	-----------------

Note:

1. Specifica se CipherSpec è conforme a FIPS (Federal Information Processing Standards) 140-2. Per una spiegazione di FIPS e per informazioni su come configurare WebSphere MQ per operazioni compatibili con FIPS 140-2, consultare *Federal Information Processing Standards (FIPS)* nella documentazione online del prodotto IBM WebSphere MQ .
2. Questa CipherSpec non può essere utilizzata per proteggere una connessione da WebSphere MQ Explorer a un gestore code a meno che i file di politica senza limitazioni appropriati non vengano applicati al JRE utilizzato da Explorer.
3. Questa CipherSpec era certificata FIPS 140-2 prima del 19 maggio 2007.
4. Quando WebSphere MQ è configurato per operazioni conformi a FIPS 140-2, questa CipherSpec può essere utilizzata per trasferire fino a 32 GB di dati prima che la connessione venga terminata con l'errore AMQ9288. Per evitare questo errore, evitare di utilizzare il triplo DES (obsoleto) o abilitare la reimpostazione della chiave segreta quando si utilizza questa CipherSpec in una configurazione FIPS 140-2.

Informazioni correlate

[Protezione](#)

[Integrità dei dati dei messaggi](#)

[Specifiche di CipherSpecs](#)

XMSC_WMQ_SSL_CIPHER_SUITE

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Il nome della CipherSuite da utilizzare su una connessione TLS a un gestore code. Il protocollo utilizzato nella negoziazione della connessione protetta dipende dalla CipherSuite specificata.

Questa proprietà ha i seguenti valori canonici:

- SSL_RSA_WITH_DES_CBC_SHA
- SSL_RSA_EXPORT1024_WITH_DES_CBC_SHA
- 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

Questo valore può essere fornito come un'alternativa a [XMSC_WMQ_SSL_CIPHER_SPEC](#).

Se viene specificato un valore non vuoto per [XMSC_WMQ_SSL_CIPHER_SPEC](#), questo valore sovrascrive l'impostazione per [XMSC_WMQ_SSL_CIPHER_SUITE](#). Se [XMSC_WMQ_SSL_CIPHER_SPEC](#) non ha un valore, il valore di [XMSC_WMQ_SSL_CIPHER_SUITE](#) viene utilizzato come suite di cifratura da fornire a GSKit. In tal caso, il valore viene associato sul valore CipherSpec equivalente, come descritto in

“Associazioni di nomi CipherSuite e CipherSpec per connessioni a un gestore code IBM MQ” a pagina 67.

Se sia XMSC_WMQ_SSL_CIPHER_SPEC che XMSC_WMQ_SSL_CIPHER_SUITE sono vuoti, il campo pChDef ->SSLCipherSpec viene riempito con spazi.

Solo per .NET : da IBM MQ 8.0, le connessioni gestite a IBM MQ (WMQ_CM_CLIENT) e le connessioni non gestite a IBM MQ (WMQ_CM_CLIENT_UNMANAGED) supportano entrambe le connessioni TLS/SSL.

Per default, la proprietà non è impostata.

Informazioni correlate

[Supporto SSL e TLS per il client .NET non gestito](#)

[Supporto SSL e TLS per il client .NET gestito](#)

XMSC_WMQ_SSL_CRYPTO_HW

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

I dettagli di configurazione per l'hardware di crittografia connesso al sistema client.

Questa proprietà ha i seguenti valori canonici:

- GSK_ACCELERATOR_RAINBOW_CS_OFF
- GSK_ACCELERATOR_RAINBOW_CS_ON
- GSK_ACCELERATOR_NCIPHER_NF_OFF
- GSK_ACCELERATOR_NCIPHER_NF_ON

Esiste un formato speciale per l'hardware crittografico PKCS11 (dove DriverPath, TokenLabel e TokenPassword sono stringhe specificate dall'utente):

```
GSK_PKCS11=PKCS#11 DriverPath; PKCS#11 TokenLabel;PKCS#11 TokenPassword
```

XMS non interpreta o modifica il contenuto della stringa. Copia il valore fornito, fino a un limite di 256 caratteri a byte singolo, in MQSCO.CryptoHardware .

Solo per .NET : da IBM MQ 8.0, le connessioni gestite a IBM MQ (WMQ_CM_CLIENT) e le connessioni non gestite a IBM MQ (WMQ_CM_CLIENT_UNMANAGED) supportano entrambe le connessioni TLS/SSL.

Per default, la proprietà non è impostata.

Informazioni correlate

[Supporto SSL e TLS per il client .NET non gestito](#)

[Supporto SSL e TLS per il client .NET gestito](#)

XMSC_WMQ_SSL_FIPS_REQUIRED

Tipo di dati:

Booleano

Proprietà di:

ConnectionFactory

Il valore della proprietà determina se un'applicazione può o meno utilizzare delle suite di crittografia non conformi a FIPS. Se questa proprietà è impostata su true, per la connessione client-server vengono utilizzati solo gli algoritmi FIPS.

Questa proprietà può avere i seguenti valori, che vengono convertiti in due valori canonici per MQSCO MQSCO.FipsRequired:

Tabella 34. Tabella dei valori per MQSCO.FlipsRequired FlipsRequired

Valore	Descrizione	Valore corrispondente di MQSCO.FlipsRequired
falso	È possibile utilizzare qualsiasi CipherSpec .	MQSSL_FIPS_NO (impostazione predefinita)
vero	Solo gli algoritmi di crittografia certificati FIPS possono essere utilizzati in CipherSpec che si applica a questa connessione client.	SÌ MQSSL_FIPS

XMS copia il relativo valore in MQSCO.FlipsRequired prima di richiamare MQCONN.

Il parametro MQSCO.FlipsRequired è disponibile solo da WebSphere MQ versione 6. Se WebSphere MQ versione 5.3, se questa proprietà è impostata, XMS non tenta di stabilire la connessione con il gestore code e genera invece un'eccezione appropriata.

Solo per .NET : da IBM MQ 8.0, le connessioni gestite a IBM MQ (WMQ_CM_CLIENT) e le connessioni non gestite a IBM MQ (WMQ_CM_CLIENT_UNMANAGED) supportano entrambe le connessioni TLS/SSL.

Informazioni correlate

[Supporto SSL e TLS per il client .NET non gestito](#)

[Supporto SSL e TLS per il client .NET gestito](#)

XMSC_WMQ_SSL_KEY_REPOSITORY

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

L'ubicazione del file del database delle chiavi in cui sono memorizzati chiavi e certificati.

XMS copia la stringa, fino a un limite di 256 caratteri a byte singolo, in MQSCO.KeyRepository . WebSphere MQ interpreta questa stringa come un nome file, incluso il percorso completo.

Solo per .NET : da IBM MQ 8.0, le connessioni gestite a IBM MQ (WMQ_CM_CLIENT) e le connessioni non gestite a IBM MQ (WMQ_CM_CLIENT_UNMANAGED) supportano entrambe le connessioni TLS/SSL.

Per default, la proprietà non è impostata.

Informazioni correlate

[Supporto SSL e TLS per il client .NET non gestito](#)

[Supporto SSL e TLS per il client .NET gestito](#)

XMSC_WMQ_SSL_KEY_RESETCOUNT

Tipo di dati:

System.Int32

Proprietà di:

ConnectionFactory

Il KeyResetCount rappresenta il numero totale di byte non crittografati inviati e ricevuti in una conversazione SSL prima che venga rinegoziata la chiave segreta. Il numero di byte include le informazioni di controllo inviate da MCA.

XMS copia il valore fornito per questa proprietà in MQSCO.KeyResetCount prima di richiamare MQCONN.

Il parametro MQSCO.KeyRestCount è disponibile solo da WebSphere MQ versione 6. Se WebSphere MQ versione 5.3, se questa proprietà è impostata, XMS non tenta di stabilire la connessione con il gestore code e genera invece un'eccezione appropriata.

Solo per .NET : da IBM MQ 8.0, le connessioni gestite a IBM MQ (WMQ_CM_CLIENT) e le connessioni non gestite a IBM MQ (WMQ_CM_CLIENT_UNMANAGED) supportano entrambe le connessioni TLS/SSL.

Il valore predefinito di questa proprietà è zero, il che significa che le chiavi segrete non vengono mai rinegoziate.

Informazioni correlate

[Supporto SSL e TLS per il client .NET non gestito](#)

[Supporto SSL e TLS per il client .NET gestito](#)

XMSC_WMQ_SSL_PEER_NAME

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Il nome peer da utilizzare su una connessione SSL a un gestore code.

Non esiste alcun elenco di valori canonici per questa proprietà. Invece, è necessario creare questa stringa in base alle regole per SSLPEER.

Un esempio di nome peer è:

```
"CN=John Smith, O=IBM ,OU=Test , C=GB"
```

XMS copia la stringa nella codepage a byte singolo corretta e inserisce i valori corretti in MQCD.SSLPeerNamePtr e MQCD.SSLPeerNameLength prima di chiamare MQCONN.

Questa proprietà è rilevante solo se l'applicazione si connette a un gestore code in modalità client.

Solo per .NET : da IBM MQ 8.0, le connessioni gestite a IBM MQ (WMQ_CM_CLIENT) e le connessioni non gestite a IBM MQ (WMQ_CM_CLIENT_UNMANAGED) supportano entrambe le connessioni TLS/SSL.

Per default, la proprietà non è impostata.

Informazioni correlate

[Supporto SSL e TLS per il client .NET non gestito](#)

[Supporto SSL e TLS per il client .NET gestito](#)

[SSLPEERNAME](#)

XMSC_WMQ_SYNCPOINT_ALL_GETS

Tipo di dati:

System.Boolean

Proprietà di:

ConnectionFactory

Indica se tutti i messaggi devono essere richiamati dalle code entro il controllo del punto di sincronizzazione.

Di seguito sono riportati i valori validi della proprietà:

Valore valido	Significato
falso	Quando le circostanze sono appropriate, il client XMS può richiamare i messaggi dalle code al di fuori del controllo del punto di sincronizzazione.
vero	Il client XMS deve recuperare tutti i messaggi dalle code all'interno del controllo del punto di sincronizzazione.

Il valore predefinito è false.

XMSC_WMQ_TARGET_CLIENT

Tipo di dati:

System.Int32

Proprietà di:

Destination

Nome utilizzato in un URI:

targetClient

Indica se i messaggi inviati alla destinazione contengono un'intestazione MQRFH2.

Se un'applicazione invia un messaggio contenente un'intestazione MQRFH2 , l'applicazione ricevente deve essere in grado di gestire l'intestazione.

Di seguito sono riportati i valori validi della proprietà:

Valore valido	Significato
XMSC_WMQ_TARGET_DEST_JMS	I messaggi inviati alla destinazione contengono un'intestazione MQRFH2 . Specificare questo valore se l'applicazione sta inviando i messaggi a un'altra applicazione XMS , un'applicazione WebSphere JMS o un'applicazione WebSphere MQ nativa progettata per gestire un'intestazione MQRFH2 .
XMSC_WMQ_TARGET_DEST_MQ	I messaggi inviati alla destinazione non contengono un'intestazione MQRFH2 . Specificare questo valore se l'applicazione sta inviando i messaggi a una applicazione WebSphere MQ nativa che non è progettata per gestire un'intestazione MQRFH2 .

Il valore predefinito è XMSC_WMQ_TARGET_DEST_JMS.

XMSC_WMQ_TEMP_Q_PREFIX

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Il prefisso utilizzato per formare il nome della coda dinamica WebSphere MQ creata quando l'applicazione crea la coda temporanea un XMS .

Le regole per formare il prefisso sono le stesse regole per formare il contenuto del campo **DynamicQName** in un descrittore oggetto, ma l'ultimo carattere non vuoto deve essere un asterisco (*). Se la proprietà non è impostata, il valore utilizzato è CSQ .* su z/OS e AMQ .* sulle altre piattaforme. Per default, la proprietà non è impostata.

Questa proprietà è rilevante solo nel dominio Point-to-point .

XMSC_WMQ_TEMP_TOPIC_PREFIX

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory, Destinazione

Quando si creano argomenti temporanei, XMS genera una stringa di argomenti nel formato "TEMP/TEMPTOPICPREFIX/unique_id" oppure se questa proprietà contiene il valore predefinito, viene generata questa stringa, "TEMP/unique_id". La specifica di un valore non vuoto consente la definizione di specifiche code modello per la creazione delle code gestite per i sottoscrittori di argomenti temporanei creati in questa connessione.

Qualsiasi stringa non null costituita solo da caratteri validi per una stringa di argomenti IBM WebSphere MQ è un valore valido per questa proprietà.

Per impostazione predefinita questa proprietà è impostata su "" (stringa vuota).

Nota: Questa proprietà è rilevante solo nel dominio di pubblicazione / sottoscrizione.

XMSC_WMQ_TEMPORA_MODEL

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Il nome della coda modello WebSphere MQ da cui viene creata una coda dinamica quando l'applicazione crea una un XMS coda temporanea.

Il valore predefinito della proprietà è SYSTEM.DEFAULT.MODEL.QUEUE.

Questa proprietà è rilevante solo nel dominio Point-to-point .

XMSC_WMQ_WILDCARD_FORMATO

Tipo di dati:

System.Int32

Proprietà di:

ConnectionFactory, Destinazione

Questa proprietà determina quale versione della sintassi dei caratteri jolly deve essere utilizzata.

Quando si utilizza la pubblicazione / sottoscrizione con IBM WebSphere MQ '*' e '?' sono considerati caratteri jolly. Mentre '#' e '+' vengono considerati come caratteri jolly quando si utilizza la sottoscrizione di pubblicazione con IBM Integration Bus. Questa proprietà sostituisce la proprietà XMSC_WMQ_BROKER_VERSION.

I valori validi per questa proprietà sono:

XMSC_WMQ_WILDCARD_SOLO argomento

Riconosce solo i caratteri jolly a livello di argomento, ad es. '#' e '+' sono trattati come caratteri jolly. Questo valore è uguale a XMSC_WMQ_BROKER_V2.

XMSC_WMQ_WILDCARD_CHAR_ONLY

Riconosce solo i caratteri jolly, ad esempio '*' e '?' sono considerati caratteri jolly. Questo valore è uguale a XMSC_WMQ_BROKER_V1.

Per impostazione predefinita, questa proprietà è impostata su XMSC_WMQ_WILDCARD_TOPIC_ONLY.

BUS_XMSC_WPM_NOME

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory e Destinazione

Nome utilizzato in un URI:

busName

Per una factory di connessione, il nome del bus di integrazione del servizio a cui si connette l'applicazione oppure, per una destinazione, il nome del bus di integrazione del servizio in cui esiste la destinazione.

Per una destinazione che è un argomento, questa proprietà è il nome del SIB (service integration bus) in cui esiste lo spazio argomento associato. Questo spazio argomento viene specificato dalla proprietà XMSC_WPM_TOPIC_SPACE .

Se la proprietà non è impostata per una destinazione, si presuppone che la coda o lo spazio argomento associato esista nel SIB (Service Integration Bus) a cui si connette l'applicazione.

Per default, la proprietà non è impostata.

PROTOCOLLO_COLLEGAMENTO_WPM_XMSC_

Tipo di dati:

System.Int32

Proprietà di:

Connessione

Il protocollo di comunicazione utilizzato per la connessione al motore di messaggistica. Questa proprietà è di sola lettura.

I valori possibili della proprietà sono i seguenti:

Valore	Significato
HTTP CP WPM XMSC	La connessione utilizza HTTP su TCP/IP.
CP_XMSC_WPM_TCP	La connessione utilizza TCP/IP.

XMSC_WPM_CONNECTION_PROSSIMITÀ

Tipo di dati:

System.Int32

Proprietà di:

ConnectionFactory

L'impostazione di prossimità della connessione per la connessione. Questa proprietà determina la vicinanza del motore di messaggistica a cui si connette l'applicazione al server di avvio.

Di seguito sono riportati i valori validi della proprietà:

Valore valido	Impostazione di prossimità della connessione
XMSC_WPM_CONNECTION_PROXIMITY_BUS	Autobus
XMSC_WPM_CONNECTION_PROXIMITY_CLUSTER	Cluster
HOST_PROSSIMITÀ_WPM_CONNECTION_XMSC_	Host
SERVER_PROSSIMITÀ_WPM_XMSC_CONNESSIONE	Server

Il valore predefinito è XMSC_WPM_CONNECTION_PROXIMITY_BUS.

XMSC_WPM_DUR_SUB_HOME

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Nome utilizzato in un URI:

Home di durableSubscription

Il nome del motore di messaggistica dove vengono gestite tutte le sottoscrizioni durevoli per una connessione o una destinazione. I messaggi da consegnare ai sottoscrittori durevoli vengono memorizzati nel punto di pubblicazione dello stesso motore di messaggistica.

È necessario specificare una home di sottoscrizione durevole per una connessione prima che un'applicazione possa creare un sottoscrittore durevole che utilizza la connessione. Qualsiasi valore specificato per una destinazione sovrascrive il valore specificato per la connessione.

Per default, la proprietà non è impostata.

Questa proprietà è rilevante solo nel dominio Pubblicazione/sottoscrizione .

Nome_HOST_WPM_XMSC

Tipo di dati:

Stringa

Proprietà di:

Connessione

Il nome host o l'indirizzo IP del sistema che contiene il motore di messaggistica a cui è connessa l'applicazione. Questa proprietà è di sola lettura.

XMSC_WPM_LOCAL_ADDRESS

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Per una connessione a un bus di integrazione del servizio, questa proprietà specifica l'interfaccia di rete locale da utilizzare oppure la porta locale o l'intervallo di porte locali da utilizzare, oppure entrambe le cose.

Il valore della proprietà è una stringa con il seguente formato:

[*nome_host*] [(*porta_inferiore*) [,*porta_alto*]]

I significati delle variabili sono i seguenti:

nome_host

Il nome host o l'indirizzo IP dell'interfaccia di rete locale da utilizzare per la connessione.

Fornire queste informazioni è necessario solo se il sistema su cui è in esecuzione l'applicazione ha due o più interfacce di rete ed è necessario essere in grado di specificare quale interfaccia deve essere utilizzata per la connessione. Se il sistema dispone di una sola interfaccia di rete, è possibile utilizzare solo tale interfaccia. Se il sistema ha due o più interfacce di rete e non si specifica quale interfaccia deve essere utilizzata, l'interfaccia viene selezionata in modo casuale.

porta_inferiore

Il numero della porta locale da utilizzare per la connessione.

Se viene specificato anche *high_port* , *low_port* viene interpretato come il numero di porta più basso in un intervallo di numeri di porta.

porta_alto

Il numero di porta più alto in un intervallo di numeri di porta. Una delle porte nell'intervallo specificato deve essere utilizzata per la connessione.

Ecco alcuni esempi di valori validi della proprietà:

JUPITER
9.20.4.98
JUPITER (1000)
9.20.4.98(1000,2000)
(1000)
(1000,2000)

Per default, la proprietà non è impostata.

Nome_ME_WPM_XMSC

Tipo di dati:

Stringa

Proprietà di:

Connessione

Il nome del motore di messaggistica a cui è connessa l'applicazione. Questa proprietà è di sola lettura.

MAP_PERSISTENT_WPM_NON_XMSC

Tipo di dati:
System.Int32

Proprietà di:
ConnectionFactory

Il livello di affidabilità dei messaggi non persistenti inviati utilizzando la connessione.

Di seguito sono riportati i valori validi della proprietà:

Valore valido

XMSC_WPM_MAPPING_AS_DESTINATION

XMSC_WPM_MAPPING_BEST_FACILE non_
Persistente

XMSC_WPM_MAPPING_EXPRESS_NON_
Persistente

XMSC_WPM_MAPPING_RELIABLE_NON_
Persistente

XMSC_WPM_MAPPING_RELIABLE_PERSISTENT

XMSC_WPM_MAPPING_ASSURED_PERSISTENT

Livello di affidabilità

Determinato dal livello di affidabilità predefinito specificato per la coda o lo spazio argomento nel SIB (Service Integration Bus)

Massimo sforzo non persistente

Espresso non persistente

Affidabile non persistente

Affidabile persistente

Assicurato persistente

Il valore predefinito è XMSC_WPM_MAPPING_EXPRESS_NON_PERSISTENT.

MAP_WPM_PERSISTENT_XMSC

Tipo di dati:
System.Int32

Proprietà di:
ConnectionFactory

Il livello di affidabilità dei messaggi persistenti inviati utilizzando la connessione.

Di seguito sono riportati i valori validi della proprietà:

Valore valido

XMSC_WPM_MAPPING_AS_DESTINATION

XMSC_WPM_MAPPING_BEST_FACILE non_
Persistente

XMSC_WPM_MAPPING_EXPRESS_NON_
Persistente

Livello di affidabilità

Determinato dal livello di affidabilità predefinito specificato per la coda o lo spazio argomento nel SIB (Service Integration Bus)

Massimo sforzo non persistente

Espresso non persistente

Valore valido

XMSC_WPM_MAPPING_RELIABLE_NON_PERSISTENTE

XMSC_WPM_MAPPING_RELIABLE_PERSISTENT

XMSC_WPM_MAPPING_ASSURED_PERSISTENT

Livello di affidabilità

Affidabile non persistente

Affidabile persistente

Assicurato persistente

Il valore predefinito è XMSC_WPM_MAPPING_RELIABLE_PERSISTENT.

PORTA_WPM_XMSC**Tipo di dati:**

System.Int32

Proprietà di:

Connessione

Il numero della porta di ascolto da parte del motore di messaggistica a cui è connessa l'applicazione. Questa proprietà è di sola lettura.

XMSC_WPM_PROVIDER_ENDPOINTS**Tipo di dati:**

Stringa

Proprietà di:

ConnectionFactory

Una sequenza di uno o più indirizzi endpoint di server di avvio. Gli indirizzi endpoint sono separati da virgole.

Un server bootstrap è un server delle applicazioni responsabile della selezione del motore di messaggistica a cui si connette l'applicazione. L'indirizzo endpoint di un server di avvio ha il seguente formato:

nome_host:numero_porta:nome_catena

I significati dei componenti di un indirizzo endpoint sono i seguenti:

nome_host

Il nome host o l'indirizzo IP del sistema su cui risiede il server di avvio. Se non viene specificato alcun nome host o indirizzo IP, il valore predefinito è localhost.

port_number

Il numero della porta su cui il server bootstrap è in attesa delle richieste in entrata. Se non viene specificato alcun numero di porta, il valore predefinito è 7276.

nome_catena

Il nome di una catena di trasporto di avvio utilizzata dal server di avvio. I valori validi sono i seguenti:

Valore valido

XMSC_WPM_BOOTSTRAP_HTTP

XMSC_WPM_BOOTSTRAP_HTTPS

XMSC_WPM_BOOTSTRAP_SSL

XMSC_WPM_BOOTSTRAP_TCP

Nome della catena di trasporto bootstrap

Messaggistica BootstrapTunneled

BootstrapTunneledSecureMessaging

Messaggistica BootstrapSecure

Messaggistica BootstrapBasic

Se non viene specificato alcun nome, il valore predefinito è XMSC_WPM_BOOTSTRAP_TCP.

Se non viene specificato alcun indirizzo endpoint, il valore predefinito è localhost:7276:BootstrapBasicMessaging.

XMSC_WPM_SSL_CIPHER_SUITE

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Il nome della CipherSuite da utilizzare su una connessione TLS a un motore di messaggistica WebSphere Integration Bus . Il protocollo utilizzato nella negoziazione della connessione protetta dipende dalla CipherSuite specificata.

Suite di cifratura	Protocollo utilizzato
TLS_RSA_WITH_DES_CBC_SHA	TLSv1
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLSv1
TLS_RSA_WITH_AES_128_CBC_SHA	TLSv1
TLS_RSA_WITH_AES_256_CBC_SHA	TLSv1

Note:

1. TLS_RSA_WITH_AES_128_CBC_SHA e TLS_RSA_WITH_AES_256_CBC_SHA CipherSuites sono supportati solo su Windows o Solaris . (Questo è dettato da GSKit.)
2. TLS_RSA_WITH_3DES_EDE_CBC_SHA è obsoleto. Tuttavia, può essere ancora utilizzato per trasferire fino a 32 GB di dati prima che la connessione venga terminata con l'errore AMQ9288. Per evitare questo errore, è necessario evitare di utilizzare il triplo DES o abilitare la reimpostazione della chiave segreta quando si utilizza questo CipherSpec.

Non esiste alcun valore predefinito per questa proprietà. Se si desidera utilizzare SSL o TLS, è necessario specificare un valore per questa proprietà, altrimenti l'applicazione non sarà in grado di connettersi correttamente al server.

XMSC_WPM_SSL_FIPS_REQUIRED

Tipo di dati:

Booleano

Proprietà di:

ConnectionFactory

Il valore della proprietà determina se un'applicazione può o meno utilizzare delle suite di crittografia non conformi a FIPS. Se questa proprietà è impostata su true, solo gli algoritmi FIPS vengono utilizzati per la connessione client - server.L'impostazione del valore di questa proprietà su TRUE impedisce all'applicazione di utilizzare suite di cifratura non conformi a FIPS.

Per impostazione predefinita, la proprietà è impostata su FALSE (ossia, modalità FIPS disattivata).

XMSC_WPM_SSL_KEY_REPOSITORY

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Un percorso al file che è il file keyring contenente le chiavi pubbliche o private da utilizzare nella connessione sicura.

L'impostazione della proprietà del file keyring sul valore speciale di XMSC_WPM_SSL_MS_CERTIFICATE_STORE specifica l'utilizzo del database delle chiavi Microsoft

Windows . L'utilizzo del database di chiavi Microsoft Windows , disponibile in **Pannello di controllo > Opzioni Internet > Contenuto > Certificati**, elimina la necessità di un database di file di chiavi separato. L'utilizzo di questa costante su Windows x64 e altre piattaforme non è consentito.

Per default, la proprietà non è impostata.

XMSC_WPM_SSL_KEYRING_LABEL

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Il certificato da utilizzare quando si esegue l'autenticazione presso il server. Se non viene specificato alcun valore, viene utilizzato il certificato predefinito.

Per default, la proprietà non è impostata.

PW ARCHIVIAZIONE CHIAVI XMSC_WPM_SSL_KEYRING_

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

La password per il file keyring.

Questa proprietà può essere utilizzata come un'alternativa all'utilizzo di XMSC_WPM_SSL_KEYRING_STASH_FILE per configurare la password per il file keyring.

Per default, la proprietà non è impostata.

XMSC_WPM_SSL_KEYRING_STASH_FILE

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Il nome di un file binario contenente la password del file del repository delle chiavi.

Questa proprietà può essere utilizzata come alternativa all'uso di XMSC_WPM_SSL_KEYRING_PW per configurare la password per il file keyring.

Per default, la proprietà non è impostata.

GRUPPO_SISTEMA_WPM_XMSC

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Il nome del gruppo di destinazione dei motori di messaggistica. La natura del gruppo di destinazione è determinata dalla proprietà XMSC_WPM_TARGET_TYPE .

Impostare questa proprietà se si desidera limitare la ricerca di un motore di messaggistica a un sottogruppo dei motori di messaggistica nel SIB (service integration bus). Se si desidera che l'applicazione sia in grado di collegarsi a qualsiasi motore di messaggistica nel SIB (Service Integration Bus), non impostare questa proprietà.

Per default, la proprietà non è impostata.

SIGNIFICATIVA_DESTINAZIONE_WPM_XMSC_

Tipo di dati:

System.Int32

Proprietà di:

ConnectionFactory

La significatività del gruppo di destinazione dei motori di messaggistica.

Di seguito sono riportati i valori validi della proprietà:

Valore valido	Significato
XMSC_WPM_TARGET_SIGNIFICANCE_ Preferito	Viene selezionato un motore di messaggistica nel gruppo di destinazione, se disponibile. In caso contrario, viene selezionato un motore di messaggistica esterno al gruppo di destinazione, purché si trovi nello stesso SIB (service integration bus).
XMSC_WPM_TARGET_SIGNIFICANCE_ Obbligatorio	Il motore di messaggistica selezionato deve essere nel gruppo di destinazione. Se un motore di messaggistica nel gruppo di destinazione non è disponibile, il processo di connessione non riesce.

Il valore predefinito della proprietà è XMSC_WPM_TARGET_SIGNIFICANCE_PREFERRED.

XMSC_WPM_TARGET_TRANSPORT_CHAIN

Tipo di dati:

Stringa

Proprietà di:

ConnectionFactory

Il nome della catena di trasporto in entrata che l'applicazione deve utilizzare per connettersi a un motore di messaggistica.

Il valore della proprietà può essere il nome di qualsiasi catena di trasporto in entrata disponibile nel server delle applicazioni che ospita il motore di messaggistica. La seguente costante denominata viene fornita per una delle catene di trasporto in entrata predefinite:

Costante con nome	Nome della catena di trasporto
XMSC_WPM_TARGET_TRANSPORT_CHAIN_BASIC	Messaggistica InboundBasic

Il valore predefinito della proprietà è XMSC_WPM_TARGET_TRANSPORT_CHAIN_BASIC.

TIPO_SISTEMA_DI_DESTINAZIONE_XMSC

Tipo di dati:

System.Int32

Proprietà di:

ConnectionFactory

Il tipo di gruppo di destinazione dei motori di messaggistica. Questa proprietà determina la natura del gruppo di destinazione identificato dalla proprietà XMSC_WPM_TARGET_GROUP.

Di seguito sono riportati i valori validi della proprietà:

Valore valido

XMSC_WPM_TARGET_TYPE_BUSMEMBER

Significato

Il nome del gruppo di destinazione è il nome di un membro bus. Il gruppo di destinazione è tutti i motori di messaggistica nel membro bus.

XMSC_WPM_TARGET_TYPE_CUSTOM

Il nome del gruppo di destinazioni è il nome di un gruppo definito dall'utente di motori di messaggistica. Il gruppo di destinazione è costituito da tutti i motori di messaggistica registrati con il gruppo definito dall'utente.

XMSC_WPM_TARGET_TYPE_ME

Il nome del gruppo di destinazione è il nome di un motore di messaggistica. Il gruppo di destinazioni è il motore di messaggistica specificato.

Per default, la proprietà non è impostata.

XMSC_WPM_TEMP_Q_PREFIX**Tipo di dati:**

Stringa

Proprietà di:

ConnectionFactory

Il prefisso utilizzato per formare il nome della coda temporanea creata nel SIB (Service Integration Bus) quando l'applicazione crea la coda temporanea un XMS . Il prefisso può contenere fino a 12 caratteri.

Il nome di una coda temporanea inizia con i caratteri "_Q" seguiti dal prefisso. Il resto del nome è costituito da caratteri generati dal sistema.

Per impostazione predefinita, la proprietà non è impostata, il che significa che il nome di una coda temporanea non ha un prefisso.

Questa proprietà è rilevante solo nel dominio Point-to-point .

XMSC_WPM_TEMP_TOPIC_PREFIX**Tipo di dati:**

Stringa

Proprietà di:

ConnectionFactory

Il prefisso utilizzato per formare il nome di un argomento temporaneo creato dall'applicazione. Il prefisso può contenere fino a 12 caratteri.

Il nome di un argomento temporaneo inizia con i caratteri "_T" seguiti dal prefisso. Il resto del nome è costituito da caratteri generati dal sistema.

Per impostazione predefinita, la proprietà non è impostata, il che significa che il nome di un argomento temporaneo non ha un prefisso.

Questa proprietà è rilevante solo nel dominio Pubblicazione/sottoscrizione .

SPACE XMSC_WPM_TOPIC_XX_ENCODE_CASE_ONE spazio**Tipo di dati:**

Stringa

Proprietà di:

Destination

Nome utilizzato in un URI:

topicSpace

Il nome dello spazio argomento che contiene l'argomento. Solo una destinazione che è un argomento può avere questa proprietà.

Per impostazione predefinita, la proprietà non viene impostata, il che significa che viene utilizzato lo spazio argomento predefinito.

Questa proprietà è rilevante solo nel dominio Pubblicazione/sottoscrizione .

Informazioni particolari

Queste informazioni sono state sviluppate per i prodotti ed i servizi offerti negli Stati Uniti.

IBM potrebbe non offrire i prodotti, i servizi o le funzioni descritti in questo documento in altri paesi. Consultare il rappresentante IBM locale per informazioni sui prodotti e sui servizi disponibili nel proprio paese. Ogni riferimento relativo a prodotti, programmi o servizi IBM non implica che solo quei prodotti, programmi o servizi IBM possano essere utilizzati. In sostituzione a quelli forniti da IBM possono essere usati prodotti, programmi o servizi funzionalmente equivalenti che non comportino la violazione dei diritti di proprietà intellettuale o di altri diritti dell'IBM. È comunque responsabilità dell'utente valutare e verificare la possibilità di utilizzare altri programmi e/o prodotti, fatta eccezione per quelli espressamente indicati dall'IBM.

IBM potrebbe disporre di applicazioni di brevetti o brevetti in corso relativi all'argomento descritto in questo documento. La fornitura di tale documento non concede alcuna licenza a tali brevetti. Chi desiderasse ricevere informazioni relative a licenze può rivolgersi per iscritto a:

Director of Commercial Relations
IBM Corporation
Schoenaicher Str. 220
D-7030 Boeblingen
U.S.A.

Per richieste di licenze relative ad informazioni double-byte (DBCS), contattare il Dipartimento di Proprietà Intellettuale IBM nel proprio paese o inviare richieste per iscritto a:

Intellectual Property Licensing
Legge sulla proprietà intellettuale e legale
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

Il seguente paragrafo non si applica al Regno Unito o a qualunque altro paese in cui tali dichiarazioni sono incompatibili con le norme locali: INTERNATIONAL BUSINESS MACHINES CORPORATION FORNISCE LA PRESENTE PUBBLICAZIONE "NELLO STATO IN CUI SI TROVA" SENZA GARANZIE DI ALCUN TIPO, ESPRESSE O IMPLICITE, IVI INCLUSE, A TITOLO DI ESEMPIO, GARANZIE IMPLICITE DI NON VIOLAZIONE, DI COMMERCIALIZZABILITÀ E DI IDONEITÀ PER UNO SCOPO PARTICOLARE. Alcuni stati non consentono la rinuncia a garanzie esplicite o implicite in determinate transazioni; quindi la presente dichiarazione potrebbe non essere applicabile.

Questa pubblicazione potrebbe contenere imprecisioni tecniche o errori tipografici. Le informazioni incluse in questo documento vengono modificate su base periodica; tali modifiche vengono incorporate nelle nuove edizioni della pubblicazione. IBM si riserva il diritto di apportare miglioramenti o modifiche al prodotto/i e/o al programma/i descritti nella pubblicazione in qualsiasi momento e senza preavviso.

Qualsiasi riferimento a siti Web non IBM contenuto nelle presenti informazioni è fornito per consultazione e non vuole in alcun modo promuovere i suddetti siti Web. I materiali presenti in tali siti Web non sono parte dei materiali per questo prodotto IBM e l'utilizzo di tali siti Web è a proprio rischio.

Tutti i commenti e i suggerimenti inviati potranno essere utilizzati liberamente da IBM e diventeranno esclusiva della stessa.

Coloro che detengono la licenza su questo programma e desiderano avere informazioni su di esso allo scopo di consentire (i) uno scambio di informazioni tra programmi indipendenti ed altri (compreso questo) e (ii) l'uso reciproco di tali informazioni, dovrebbero rivolgersi a:

IBM Corporation
Coordinatore interoperabilità software, Dipartimento 49XA
Autostrada 3605 52 N

Rochester, MN 55901
U.S.A.

Queste informazioni possono essere rese disponibili secondo condizioni contrattuali appropriate, compreso, in alcuni casi, il pagamento di un addebito.

Il programma su licenza descritto in queste informazioni e tutto il materiale su licenza disponibile per esso sono forniti da IBM in base ai termini dell' IBM Customer Agreement, IBM International Program License Agreement o qualsiasi altro accordo equivalente tra le parti.

Tutti i dati relativi alle prestazioni contenuti in questo documento sono stati determinati in un ambiente controllato. Pertanto, i risultati ottenuti in altri ambienti operativi possono variare in modo significativo. Alcune misurazioni potrebbero essere state fatte su sistemi a livello di sviluppo e non vi è alcuna garanzia che queste misurazioni saranno le stesse sui sistemi generalmente disponibili. Inoltre, alcune misurazioni potrebbero essere state stimate mediante estrapolazione. I risultati quindi possono variare. Gli utenti di questo documento dovrebbero verificare i dati applicabili per il loro ambiente specifico.

Le informazioni relative a prodotti non IBM provengono dai fornitori di tali prodotti, dagli annunci pubblicati o da altre fonti pubblicamente disponibili. IBM non ha verificato tali prodotti e, pertanto, non può garantirne l'accuratezza delle prestazioni. Eventuali commenti relativi alle prestazioni dei prodotti non IBM devono essere indirizzati ai fornitori di tali prodotti.

Tutte le dichiarazioni riguardanti la direzione o l'intento futuro di IBM sono soggette a modifica o ritiro senza preavviso e rappresentano solo scopi e obiettivi.

Questa pubblicazione contiene esempi di dati e prospetti utilizzati quotidianamente nelle operazioni aziendali, Per illustrarle nel modo più completo possibile, gli esempi includono i nomi di individui, società, marchi e prodotti. Tutti questi nomi sono fittizi e qualsiasi somiglianza con nomi ed indirizzi adoperati da imprese realmente esistenti sono una mera coincidenza.

LICENZA SUL COPYRIGHT:

Queste informazioni contengono programmi applicativi di esempio in lingua originale, che illustrano le tecniche di programmazione su diverse piattaforme operative. È possibile copiare, modificare e distribuire questi programmi di esempio sotto qualsiasi forma senza alcun pagamento alla IBM, allo scopo di sviluppare, utilizzare, commercializzare o distribuire i programmi applicativi in conformità alle API (application programming interface) a seconda della piattaforma operativa per cui i programmi di esempio sono stati scritti. Questi esempi non sono stati testati approfonditamente tenendo conto di tutte le condizioni possibili. IBM, quindi, non può garantire o sottintendere l'affidabilità, l'utilità o il funzionamento di questi programmi.

Se si sta visualizzando queste informazioni in formato elettronico, le fotografie e le illustrazioni a colori potrebbero non apparire.

Informazioni sull'interfaccia di programmazione

Le informazioni sull'interfaccia di programmazione, se fornite, consentono di creare software applicativo da utilizzare con questo programma.

Questo manuale contiene informazioni sulle interfacce di programmazione che consentono al cliente di scrivere programmi per ottenere i servizi di WebSphere MQ.

Queste informazioni, tuttavia, possono contenere diagnosi, modifica e regolazione delle informazioni. La diagnosi, la modifica e la regolazione delle informazioni vengono fornite per consentire il debug del software applicativo.

Importante: Non utilizzare queste informazioni di diagnosi, modifica e ottimizzazione come interfaccia di programmazione poiché sono soggette a modifica.

Marchi

IBM, il logo IBM , ibm.com, sono marchi di IBM Corporation, registrati in molte giurisdizioni nel mondo. Un elenco aggiornato dei marchi IBM è disponibile sul web in "Copyright and trademark

information"www.ibm.com/legal/copytrade.shtml. Altri nomi di prodotti e servizi potrebbero essere marchi di IBM o altre società.

Microsoft e Windows sono marchi di Microsoft Corporation negli Stati Uniti e/o in altri paesi.

UNIX è un marchio registrato di The Open Group negli Stati Uniti e/o in altri paesi.

Linux® è un marchio registrato di Linus Torvalds negli Stati Uniti e/o in altri paesi.

Questo prodotto include il software sviluppato da Eclipse Project (<http://www.eclipse.org/>).

Java e tutti i marchi e i logo Java sono marchi registrati di Oracle e/o di società affiliate.



Numero parte:

(1P) P/N: