

9.0

IBM Message Service Client for .NET

IBM

Poznámka

Než začnete používat tyto informace a produkt, který podporují, přečtěte si informace, které uvádí [“Poznámky” na stránce 241](#).

Toto vydání se vztahuje k verzi 9 vydání 0 produktu IBM® MQ a ke všem následujícím vydáním a modifikacím, dokud nebude v nových vydáních uvedeno jinak.

Když odešlete informace do IBM, udělíte společnosti IBM nevýlučné právo použít nebo distribuovat informace libovolným způsobem, který společnost považuje za odpovídající, bez vzniku jakýchkoliv závazků vůči vám.

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

Obsah

Message Service Client for .NET.....	5
Úvod do produktu IBM Message Service Client for .NET.....	5
Styl systému zpráv.....	6
Model objektu XMS.....	7
Atributy a vlastnosti objektů.....	8
Spravované objekty.....	8
Model zpráv produktu XMS.....	9
Zabránění aplikacím od použití novější verze XMS.....	12
Nastavení prostředí serveru systému zpráv.....	12
Konfigurace správce front a zprostředkovatele pro aplikaci, která se připojuje ke správci front produktu IBM MQ.....	13
Konfigurace zprostředkovatele pro aplikaci, která používá připojení v reálném čase ke zprostředkovateli.....	14
Konfigurace sběrnice pro integraci služeb pro aplikaci, která se připojuje k produktu WebSphere Application Server.....	16
Použití ukázkových aplikací produktu XMS.....	17
Ukázkové aplikace.....	17
Spuštění ukázkových aplikací.....	18
Sestavení ukázkových aplikací produktu .NET.....	19
Vývoj aplikací produktu XMS.....	19
Zápis aplikací XMS.....	20
Zápis aplikací produktu XMS .NET.....	42
Práce se spravovanými objekty.....	48
Zabezpečení komunikací pro aplikace XMS.....	63
Zprávy produktu XMS.....	66
Odstraňování problémů.....	80
Konfigurace trasování pro aplikace produktu .NET.....	80
Konfigurace FFDC pro aplikace .NET.....	84
Tipy pro odstraňování problémů.....	84
Klienti Message Service Clients for .NET.....	86
.NETRozhraní.....	86
Vlastnosti objektů XMS.....	173
Poznámky.....	241
Informace o programovacím rozhraní.....	242
Ochranné známky.....	242

Úvod do produktu IBM Message Service Client for .NET

Produkt IBM Message Service Client for .NET poskytuje rozhraní API s názvem XMS , které má stejnou sadu rozhraní jako Java Message Service (JMS) API. Produkt IBM Message Service Client for .NET obsahuje plně spravovanou implementaci produktu XMS, kterou lze používat v libovolném jazyku vyhovujícím produktu .NET .

XMS podporuje:

- dvoubodový Systém zpráv
- Publikování/odběr Systém zpráv
- Doručení synchronní zprávy
- Asynchronní doručení zprávy

Aplikace produktu XMS si může vyměňovat zprávy s následujícími typy aplikací:

- Aplikace XMS .
- Aplikace IBM MQ classes for JMS .
- Nativní aplikace IBM MQ
- Aplikace JMS, která používá výchozího poskytovatele systému zpráv produktu IBM MQ .

Aplikace produktu XMS se může připojit k libovolnému z následujících serverů systému zpráv a využívat je těchto prostředků:

IBM MQ správce front

Aplikace se může připojit buď v rámci vazeb, nebo v režimu klienta.

WebSphere Application Server Služba Integration Bus

Aplikace může použít přímé připojení TCP/IP, nebo může použít HTTP přes TCP/IP.

IBM Integration Bus

Zprávy jsou transportovány mezi aplikací a zprostředkovatelem pomocí produktu WebSphere MQ Real-time Transport. Zprávy mohou být doručeny do aplikace pomocí produktu Přenos výběrového vysílání WebSphere MQ.

Připojením ke správci front produktu IBM MQ může aplikace XMS použít ke komunikaci s IBM Integration Busprodukt WebSphere MQ Enterprise Transport . Alternativně může aplikace XMS publikovat a odebírat připojení k produktu IBM MQ.

Související pojmy

[“Styly systému zpráv” na stránce 6](#)

[“Model objektu XMS” na stránce 7](#)

Rozhraní API produktu XMS je objektově orientované rozhraní. Model objektu XMS je založen na modelu objektu JMS 1.1 .

[“Model zpráv produktu XMS” na stránce 9](#)

Model zprávy produktu XMS je stejný jako model zpráv produktu IBM MQ classes for JMS .

Úvod do produktu IBM Message Service Client for .NET

Produkt IBM Message Service Client for .NET poskytuje rozhraní API s názvem XMS , které má stejnou sadu rozhraní jako Java Message Service (JMS) API. Produkt IBM Message Service Client for .NET obsahuje plně spravovanou implementaci produktu XMS, kterou lze používat v libovolném jazyku vyhovujícím produktu .NET .

XMS podporuje:

- dvoubodový Systém zpráv
- Publikování/odběr Systém zpráv

- Doručení synchronní zprávy
- Asynchronní doručení zprávy

Aplikace produktu XMS si může vyměňovat zprávy s následujícími typy aplikací:

- Aplikace XMS .
- Aplikace IBM MQ classes for JMS .
- Nativní aplikace IBM MQ
- Aplikace JMS, která používá výchozího poskytovatele systému zpráv produktu IBM MQ .

Aplikace produktu XMS se může připojit k libovolnému z následujících serverů systému zpráv a využívat je těchto prostředků:

IBM MQ správce front

Aplikace se může připojit buď v rámci vazeb, nebo v režimu klienta.

WebSphere Application Server Služba Integration Bus

Aplikace může použít přímé připojení TCP/IP, nebo může použít HTTP přes TCP/IP.

IBM Integration Bus

Zprávy jsou transportovány mezi aplikací a zprostředkovatelem pomocí produktu WebSphere MQ Real-time Transport. Zprávy mohou být doručeny do aplikace pomocí produktu Přenos výběrového vysílání WebSphere MQ.

Připojením ke správci front produktu IBM MQ může aplikace XMS použít ke komunikaci s IBM Integration Busprodukt WebSphere MQ Enterprise Transport . Alternativně může aplikace XMS publikovat a odebírat připojení k produktu IBM MQ.

Související pojmy

[“Styly systému zpráv” na stránce 6](#)

[“Model objektu XMS” na stránce 7](#)

Rozhraní API produktu XMS je objektově orientované rozhraní. Model objektu XMS je založen na modelu objektu JMS 1.1 .

[“Model zpráv produktu XMS” na stránce 9](#)

Model zprávy produktu XMS je stejný jako model zpráv produktu IBM MQ classes for JMS .

Styly systému zpráv

Produkt XMS podporuje styly systému zpráv dvoubodový a Publikování/odběr .

Styly systému zpráv se také nazývají domény systému zpráv.

dvoubodový Systém zpráv

Společný formulář systému zpráv produktu dvoubodový používá řazení do fronty. V nejjednodušším případě aplikace odešle zprávu do jiné aplikace určením, implicitně nebo explicitně, cílovou frontou. Základní systém zpráv a systém front přijímá zprávu z odesílající aplikace a směřuje zprávu do její cílové fronty. Přijímající aplikace pak může načíst zprávu z fronty.

Pokud podkladový systém zpráv a systém front obsahuje IBM Integration Bus, může produkt IBM Integration Bus replikovat zprávu a směřovat kopie zprávy do jiných front. V důsledku toho může zprávu obdržet více než jedna aplikace. Produkt IBM Integration Bus může také transformovat zprávu a přidat do ní data.

Klíčovou charakteristikou systému zpráv produktu dvoubodový je to, že aplikace umístí zprávu do lokální fronty, jakmile odešle zprávu. Základní systém zpráv a systém front zpráv určuje cílovou frontu, do níž je zpráva odeslána. Přijímající aplikace načte zprávu z cílové fronty.

Publikování/odběr Systém zpráv

V systému zpráv produktu Publikování/odběr jsou k dispozici dva typy aplikací: vydavatel a odběratel.

Vydavatel poskytuje informace ve formě zpráv o publikování. Když vydavatel publikuje zprávu, určuje téma, které identifikuje předmět informací uvnitř zprávy.

Odběratel je spotřebitel informací, které jsou publikovány. Odběratel určuje témata, která se zajímají o vytváření odběrů.

Systém publikování/odběru přijímá publikování od vydavatelů a odběrů od odběratelů. Publikuje publikování na odběratele. Odběratel přijímá publikování pouze v těch tématech, k jejichž odběru se přihlásili.

Klíčovou charakteristikou systému zpráv produktu Publikování/odběr je to, že vydavatel identifikuje téma, když publikuje zprávu. Neoznačuje odběratele. Je-li zpráva publikována na tématu, pro které nejsou žádní odběratelé, žádná aplikace neobdrží zprávu.

Aplikace může být jak vydavatel, tak i odběratel.

Model objektu XMS

Rozhraní API produktu XMS je objektově orientované rozhraní. Model objektu XMS je založen na modelu objektu JMS 1.1 .

Následující seznam shrnuje hlavní třídy XMS nebo typy objektů:

ConnectionFactory

Objekt `ConnectionFactory` zapouzdřuje sadu parametrů pro připojení. Aplikace používá k vytvoření připojení aplikační server `ConnectionFactory` . Aplikace může poskytovat parametry za běhu programu a vytvořit objekt `ConnectionFactory` . Parametry připojení lze případně uložit do úložiště spravovaných objektů. Aplikace může načíst objekt z úložiště a vytvořit z něj objekt `ConnectionFactory` .

Connection

Objekt `Connection` zapouzdřuje aktivní připojení z aplikace na server systému zpráv. Aplikace používá připojení k vytvoření relací.

Destination

Aplikace odesílá zprávy nebo přijímá zprávy pomocí objektu `Destination` . V doméně Publikování/odběr objekt `Destination` zapouzdřuje téma a v doméně dvoubodový zapouzdřuje objekt `Destination` frontu. Aplikace může poskytnout parametry za účelem vytvoření objektu `Destination` za běhu. Eventuálně může vytvořit objekt `Destination` z definice objektu, která je uložena v úložišti spravovaných objektů.

Session

Objekt `Session` je jediný vláknový kontext pro odesílání a příjem zpráv. Aplikace používá objekt `Session` k vytváření objektů `Message`, `MessageProducer` a `MessageConsumer` .

Message

Objekt `Message` zapouzdřuje objekt `Message` , který aplikace odesílá pomocí objektu `MessageProducer` , nebo přijímá pomocí objektu `MessageConsumer` .

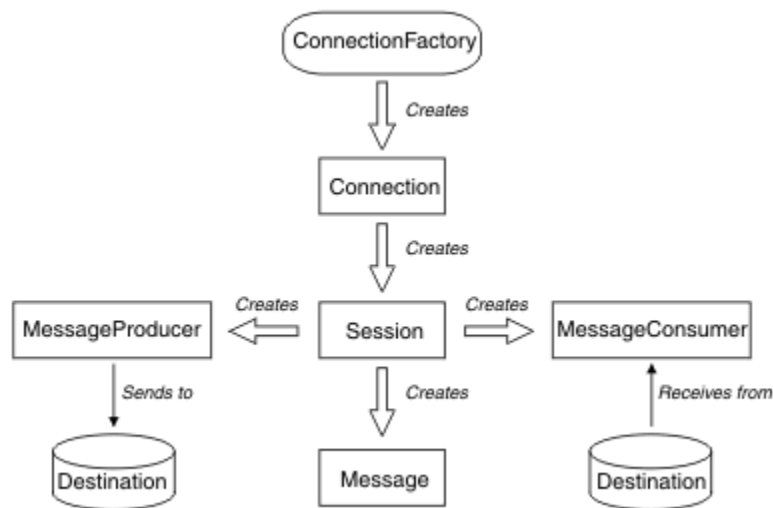
MessageProducer

Objekt `MessageProducer` je používán aplikací k odesílání zpráv do místa určení.

MessageConsumer

Objekt produktu `MessageConsumer` je používán aplikací pro příjem zpráv odeslaných do místa určení.

Produkt [Obrázek 1](#) na stránce 8 zobrazuje tyto objekty a jejich vztahy. Tento diagram zobrazuje hlavní typy objektu XMS : `ConnectionFactory`, `Connection`, `Session`, `MessageProducer`, `MessageConsumer`, `Message` a `Destination`. Aplikace používá továrnu připojení k vytvoření připojení a používá připojení k vytváření relací. Aplikace pak může použít relaci k vytváření zpráv, producentů zpráv a spotřebitelů zpráv. Aplikace používá producenta zpráv k odesílání zpráv do místa určení a k příjmu zpráv odeslaných na místo určení používá spotřebitele zpráv.



Obrázek 1. Objekty produktu XMS a jejich vztahy

V produktu .NET jsou třídy XMS definovány jako sada rozhraní .NET . Když kódující aplikace produktu XMS .NET , potřebujete pouze deklarovaná rozhraní.

Model objektu XMS je založen na rozhraních nezávislých na doméně, která jsou popsána ve specifikaci produktu Java Message Service , verze 1.1. Nejsou poskytnuty žádné třídy specifické pro doménu, jako např. Topic, TopicPublisher a TopicSubscriber.

Atributy a vlastnosti objektů

Objekt XMS může mít atributy a vlastnosti, které jsou charakteristikami objektu, které jsou implementovány různými způsoby.

Atributy

Vlastnost objektu, která je vždy přítomna a zabírá úložiště, i když tento atribut nemá hodnotu. V tomto ohledu je atribut podobný poli v datové struktuře pevné délky. Rozlišovací funkce atributů je, že každý atribut má své vlastní metody pro nastavení a získání jeho hodnoty.

Vlastnosti

Vlastnost objektu je přítomna a zabírá úložiště až po nastavení jeho hodnoty. Vlastnost nelze odstranit nebo se její úložiště obnoví po nastavení jeho hodnoty. Jeho hodnotu můžete změnit. XMS poskytuje sadu generických metod pro nastavení a získání hodnot vlastností.

Související pojmy

XMS primitivní typy

XMS poskytuje ekvivalenty osmi primitivních typů Java (byte, short, int, long, float, double, char a boolean). To umožňuje výměnu zpráv mezi XMS a JMS , aniž by došlo ke ztrátě nebo poškození dat.

Implicitní převod hodnoty vlastnosti z jednoho datového typu na jiný.

Když aplikace získá hodnotu vlastnosti, může ji převést hodnota XMS na jiný datový typ. Mnoho pravidel určuje, které konverze jsou podporovány a jak produkt XMS provádí převody.

Související odkazy

Datové typy pro prvky dat aplikace

Aby se zajistilo, že aplikace XMS si může vyměňovat zprávy s aplikací IBM MQ classes for JMS , musí být obě aplikace schopny interpretovat data aplikace v těle zpráv stejným způsobem.

Spravované objekty

Pomocí spravovaných objektů můžete spravovat nastavení připojení používaná klientskými aplikacemi, které mají být spravovány z centrálního úložiště. Aplikace načítá definice objektů z centrálního úložiště a používá je k vytváření objektů ConnectionFactory a Destination . Pomocí spravovaných objektů můžete odstranit několik aplikací z prostředků, které používají za běhu.

Například aplikace produktu XMS lze zapisovat a testovat pomocí spravovaných objektů, které odkazují na sadu připojení a cílů v testovacím prostředí. Po implementaci aplikací lze administrované objekty změnit tak, aby se aplikace mohly odkazovat na připojení a cíle v produkčním prostředí.

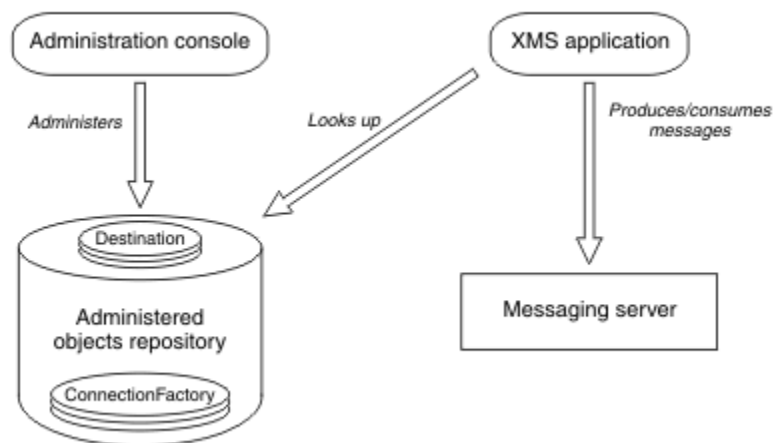
Produkt XMS podporuje dva typy spravovaných objektů:

- Objekt `ConnectionFactory`, který je používán aplikacemi k vytvoření počátečního připojení k serveru.
- Objekt `Destination`, který je používán aplikacemi k určení cíle pro odesílané zprávy a zdroj zpráv, které jsou přijímány. Místo určení je buď téma, nebo fronta na serveru, ke kterému se aplikace připojuje.

Administrativní nástroj **JMSAdmin** je dodáván s IBM MQ. Používá se k vytváření a správě spravovaných objektů v centrálním úložišti spravovaných objektů.

Spravované objekty v úložišti lze používat v aplikacích IBM MQ classes for JMS a XMS. Aplikace produktu XMS mohou použít objekty `ConnectionFactory` a `Destination` k připojení ke správci front IBM MQ. Administrátor může měnit definice objektů uložené v úložišti, aniž by to mělo vliv na kód aplikace.

Následující diagram ukazuje, jak aplikace XMS obvykle používá spravované objekty. Na levé straně diagramu je zobrazeno úložiště obsahující objekt `ConnectionFactory` a definice cílových objektů, které jsou spravovány pomocí administrativní konzoly. Na pravé straně diagramu je zobrazena aplikace XMS, která hledá definice objektů v úložišti, a poté používá tyto definice objektů při připojování k serveru systému zpráv.



Obrázek 2. Typické použití spravovaných objektů aplikací produktu XMS

Související pojmy

Práce se spravovanými objekty

Témata v této sekci obsahují informace o spravovaných objektech. Aplikace produktu XMS mohou načítat definice objektů z úložiště centrálního administrovaných objektů a používat je k vytváření továren připojení a míst určení.

Podporované typy úložiště spravovaných objektů

Objekty systému souborů a LDAP spravované objekty lze použít pro připojení k produktu IBM MQ a WebSphere Application Server, zatímco pojmenování COS lze použít k připojení pouze k WebSphere Application Server.

Související úlohy

Vytvoření spravovaných objektů

Definice objektu `ConnectionFactory` a Cílový objekt, které aplikace produktu XMS vyžadují k vytvoření připojení k serveru systému zpráv, musí být vytvořeny pomocí příslušných nástrojů pro správu.

Model zpráv produktu XMS

Model zprávy produktu XMS je stejný jako model zpráv produktu IBM MQ classes for JMS.

Produkt XMS zejména implementuje stejná pole záhlaví zprávy a vlastnosti zpráv, které produkt IBM MQ classes for JMS implementuje:

- Pole záhlaví JMS . Tato pole mají názvy, které začínají předponou JMS.
- Definované vlastnosti produktu JMS . Tato pole mají vlastnosti, jejichž názvy začínají předponou JMSX.
- Definované vlastnosti produktu IBM . Tato pole mají vlastnosti, jejichž názvy začínají předponou JMS_IBM_.

V důsledku toho mohou aplikace produktu XMS vyměňovat zprávy s aplikacemi produktu IBM MQ classes for JMS . V každé zprávě jsou některé z polí záhlaví a vlastností nastaveny aplikací a ostatní jsou nastaveny pomocí XMS nebo IBM MQ classes for JMS. Některé z polí nastavených pomocí XMS nebo IBM MQ classes for JMS jsou nastaveny při odeslání zprávy a ostatní při jejím přijetí. Pole záhlaví a vlastnosti jsou v případě potřeby šířeny se zprávou prostřednictvím serveru zpráv. Jsou k dispozici pro každou aplikaci, která přijímá zprávu.

Instalace produktu Message Service Client for .NET pomocí průvodce instalací

Instalace používá instalační program produktu InstallShield X/Windows MSI. K dispozici jsou dvě volby nastavení, takže můžete zvolit buď úplnou, nebo vlastní instalaci.

Informace o této úloze

Chcete-li instalovat produkt Message Service Client for .NET v systému Windows, postupujte podle této procedury.

Postup

1. Provádíte-li instalaci z balíku SupportPac , postupujte podle následujících kroků, jinak pokračujte přímo krokem “2” na stránce 10.
 - a) V systému Windows se přihlaste jako administrátor.
 - b) Spustíte instalační program dotNETClientsetup.exe.
2. Počkejte, až se průvodce instalací otevře a zobrazí se následující zpráva:

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

Klepněte na tlačítko **Další**.

Průvodce vás může požádat, abyste si přečetli licenční smlouvu.

3. Pokud jste požádáni o přečtení licenční smlouvy a souhlasíte s podmínkami licenční smlouvy, klepněte na volbu **Přijímám podmínky v licenční smlouvě** a poté klepněte na tlačítko **Další**.

Průvodce instalací požádá, abyste zvolili typ nastavení, který nejlépe vyhovuje vašim potřebám.

4. Vyberte typ nastavení, které požadujete:

- Chcete-li instalovat všechny funkce programu a instalovat je ve výchozím instalačním adresáři, klepněte na tlačítko **Dokončit**.
- Chcete-li vybrat funkce, které chcete instalovat, a určete, kam mají být instalovány, klepněte na volbu **Vlastní**.

5. Klepněte na tlačítko **Další**.

Vyberete-li úplnou volbu instalace, průvodce instalací zobrazí zprávu, že je připravena zahájit instalaci, jak je popsáno v kroku “8” na stránce 11. Pokud vyberete volbu vlastní instalace, průvodce instalací vás požádá, abyste vybrali funkce, které chcete instalovat, a musíte dokončit krok “6” na stránce 11 a krok “7” na stránce 11 , než se přesunete na krok “8” na stránce 11.

6. Pouze pro vlastní instalaci klepněte na ikonu v seznamu funkcí, abyste uvedli změny, jak chcete, aby byly funkce produktu Message Service Client for .NET nainstalovány. Nechcete-li instalovat produkt Message Service Client for .NET do navrženého adresáře, zvolte jiný adresář.

Pokud se rozhodnete instalovat produkt Message Service Client for .NET do adresáře, který momentálně neexistuje, průvodce instalací vytvoří adresář pro vás.

Chcete-li vyvíjet aplikace produktu XMS , ujistěte se, že je vybrána funkce **Vývojové nástroje a ukázky** . Tato funkce poskytuje ukázkové aplikace a knihovny a všechny další soubory potřebné ke kompilaci aplikací produktu .NET . Pokud tuto funkci nevyberete, budou nainstalovány pouze soubory nezbytné ke spuštění aplikací produktu XMS .

7. Používáte-li volbu vlastní instalace, klepněte na tlačítko **Další** po výběru voleb, které požadujete, jak je popsáno v kroku “6” na stránce [11](#).

Průvodce instalací zobrazí zprávu, že je připravena zahájit instalaci.

8. Klepnutím na tlačítko **Instalovat** spusťte instalaci.

Průvodce instalací zobrazí panel ukazující průběh instalace. Počkejte, až se ukazatel průběhu dokončí. Po úspěšném dokončení instalace se v okně zobrazí následující zpráva:

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

9. Klepnutím na tlačítko **Dokončit** zavřete průvodce instalací.

Výsledky

Úspěšně jste nainstalovali produkt Message Service Client for .NET, který je připraven k použití.

Jak pokračovat dále

Před spuštěním aplikací produktu XMS , včetně ukázkových aplikací dodávaných s produktem XMS, je třeba nastavit prostředí serveru systému zpráv, aby se zobrazily podrobnosti viz: [“Nastavení prostředí serveru systému zpráv”](#) na stránce [12](#).

Související pojmy

Webová služba vyhledávání JNDI

Chcete-li přistupovat k adresáři pojmenování COS z produktu XMS, musí být na serveru WebSphere Application Server service integration bus implementována webová služba vyhledávání produktu JNDI . Tato webová služba přeloží informace o produktu Java ze služby názvů COS do formuláře, který lze číst v aplikacích produktu XMS .

Nastavení prostředí serveru systému zpráv

Témata v této sekci popisují, jak nastavit prostředí serveru systému zpráv, aby bylo aplikacím produktu XMS umožněno připojit se k serveru.

Použití ukázkových aplikací produktu XMS

Použijte ukázkové aplikace dodávané s produktem XMS k ověření instalace a nastavení serveru systému zpráv a k usnadnění sestavování vlastních aplikací. Ukázky poskytují přehled běžných funkcí jednotlivých rozhraní API.

Nezbytné předpoklady pro aplikace XMS připojující se k WebSphere MQ

Některé předpoklady se použijí, pokud se aplikace XMS připojí k produktu WebSphere MQ.

U aplikací, které se připojují ke správci front produktu WebSphere MQ , je třeba nainstalovat příslušné knihovny klienta produktu WebSphere MQ na počítač, který používáte ke spuštění aplikace XMS . Tyto knihovny jsou předinstalovány na počítačích s lokálním správcem front.

Pro produkt KlientXMS pro .NET použijte knihovny klienta dodané s produktem IBM WebSphere MQ 7.0.1.0 nebo novější. Jedná se o třídy *WebSphere MQ pro .NET*. Povolí připojení režimu klienta k IBM

WebSphere MQ 7.0, 6.0a 5.3 správci front a připojení režimu vázání k lokálnímu správci front, pokud je také IBM WebSphere MQ 7.0.1.0 nebo pozdější.

Balík Microsoft .NET Framework 2.0 Redistributable Package musí být nainstalován na počítači, na kterém se má instalovat XMS . Pokud tento balík není k dispozici, instalace XMS se nezdaří. Poté je třeba ukončit instalační proceduru, instalovat balík Microsoft .NET Framework 2.0 Redistributable Package na počítač a znovu spustit instalační proceduru.

Na webu pro stahování Microsoft musíte vyhledat soubor dotnetfx.exe pro Microsoft .NET Framework 2.0 Redistributable Package (x86) a NetFx64.exe pro Microsoft .NET Framework 2.0 Redistributable Package (x64), podle toho, co je použitelné.

Související pojmy

Nastavení prostředí serveru systému zpráv

Témata v této sekci popisují, jak nastavit prostředí serveru systému zpráv, aby bylo aplikacím produktu XMS umožněno připojit se k serveru.

Zabránění aplikacím od použití novější verze XMS

Když je nainstalována novější verze produktu XMS , automaticky se aplikace používající předchozí verzi automaticky přepnou na novější verzi, aniž by bylo nutné je znovu zkompilovat.

Informace o této úloze

Funkce koexistence více verzí zajišťuje, že instalace novější verze produktu XMS nepřepíše předchozí verzi produktu XMS . Místo toho v mezipaměti Global Assembly Cache (GAC) existuje více instancí podobných montážních celků XMS .NET , ale mají odlišná čísla verzí. Interně GAC používá soubor zásad pro směrování volání aplikací do nejnovější verze produktu XMS. Aplikace běží bez nutnosti překompilace a mohou využívat nové funkce dostupné v novější verzi produktu XMS .NET .

Je-li však požadována aplikace pro použití starší verze produktu XMS , nastavte atribut publisherpolicy na hodnotu no v konfiguračním souboru aplikace.

Poznámka: Konfigurační soubor aplikace je soubor s názvem, který se skládá z názvu spustitelného programu, do kterého se soubor vztahuje, spolu s příponou .config. Například konfigurační soubor aplikace pro text.exe by měl mít název text.exe.config.

Kdykoli však všechny aplikace systému používají stejnou verzi produktu XMS .NET.

Nastavení prostředí serveru systému zpráv

Témata v této sekci popisují, jak nastavit prostředí serveru systému zpráv, aby bylo aplikacím produktu XMS umožněno připojit se k serveru.

U aplikací, které se připojují ke správci front produktu IBM MQ , je vyžadován klient IBM MQ (nebo správce front pro režim vazeb).

V současné době neexistují žádné předpoklady pro aplikace, které používají připojení v reálném čase ke zprostředkovateli.

Před spuštěním libovolných aplikací produktu XMS , včetně ukázkových aplikací dodávaných s produktem XMS, je třeba nastavit prostředí serveru systému zpráv.

Tento oddíl obsahuje následující témata:

- [“Konfigurace správce front a zprostředkovatele pro aplikaci, která se připojuje ke správci front produktu IBM MQ” na stránce 13](#)
- [“Konfigurace zprostředkovatele pro aplikaci, která používá připojení v reálném čase ke zprostředkovateli” na stránce 14](#)
- [“Konfigurace sběrnice pro integraci služeb pro aplikaci, která se připojuje k produktu WebSphere Application Server” na stránce 16](#)

Související pojmy

Webová služba vyhledávání JNDI

Chcete-li přistupovat k adresáři pojmenování COS z produktu XMS, musí být na serveru WebSphere Application Server service integration bus implementována webová služba vyhledávání produktu JNDI . Tato webová služba přeloží informace o produktu Java ze služby názvů COS do formuláře, který lze číst v aplikacích produktu XMS .

Použití ukázkových aplikací produktu XMS

Použijte ukázkové aplikace dodávané s produktem XMS k ověření instalace a nastavení serveru systému zpráv a k usnadnění sestavování vlastních aplikací. Ukázky poskytují přehled běžných funkcí jednotlivých rozhraní API.

Související úlohy

Instalace produktu Message Service Client for .NET pomocí průvodce instalací

Instalace používá instalační program produktu InstallShield X/Windows MSI. K dispozici jsou dvě volby nastavení, takže můžete zvolit buď úplnou, nebo vlastní instalaci.

Související odkazy

Nezbytné předpoklady pro aplikace XMS připojující se k WebSphere MQ

Některé předpoklady se použijí, pokud se aplikace XMS připojí k produktu WebSphere MQ.

Konfigurace správce front a zprostředkovatele pro aplikaci, která se připojuje ke správci front produktu IBM MQ

Tento oddíl předpokládá, že používáte produkt IBM WebSphere MQ 7.0.1 nebo novější. Před spuštěním aplikace, která se připojuje ke správci front produktu IBM MQ , je třeba nakonfigurovat správce front. V případě aplikace typu publikování/odběr je vyžadována některá další konfigurace, pokud používáte rozhraní publikování/odběru ve frontě.

Než začnete

XMS pracuje s IBM Integration Bus nebo WebSphere Message Broker 6.1 nebo novějším

Před spuštěním této úlohy proveďte následující kroky:

- Ujistěte se, že aplikace má přístup ke správci front, který je spuštěn.
- Je-li vaše aplikace aplikací publikování/odběru a používá rozhraní publikování/odběru ve frontě, ujistěte se, že je atribut "PSMODE" nastaven na hodnotu "ENABLED" ve správci front.
- Přesvědčte se, že vaše aplikace používá továrnu připojení, jejíž vlastnosti jsou správně nastaveny pro připojení ke správci front. Je-li vaše aplikace aplikací publikování/odběru, zkontrolujte, zda jsou pro použití zprostředkovatele nastaveny příslušné vlastnosti továrny připojení. Další informace o vlastnostech továrny připojení, "Vlastnosti objektu ConnectionFactory" na stránce 175.

Informace o této úloze

Správce front a zprostředkovatele můžete nakonfigurovat tak, aby spouštěly aplikace produktu XMS stejným způsobem jako při konfiguraci správce front a rozhraní publikování/odběru ve frontě za účelem spuštění aplikací produktu IBM MQ JMS . Následující kroky shrnují to, co je třeba provést.

Postup

1. Ve správci front vytvořte fronty, které vaše aplikace potřebuje.

Přehled způsobu vytváření front naleznete v tématu Definování front.

Je-li vaše aplikace aplikací publikování/odběru a používá rozhraní publikování/odběru ve frontě, které potřebuje přístup do systémových front produktu IBM MQ classes for JMS , před vytvořením front vyčkejte až 4a .

2. Udělte jménu uživatele přidruženému k aplikaci oprávnění pro připojení ke správci front a příslušné oprávnění pro přístup k frontám.

Přehled autorizace najdete v tématu [Zabezpečení](#). Pokud se vaše aplikace připojuje ke správci front v režimu klienta, prohlédněte si také [Klienti a servery](#).

3. Pokud se vaše aplikace připojí ke správci front v režimu klienta, zkontrolujte, zda je ve správci front definován kanál připojení k serveru a zda je modul listener spuštěn.

Tento krok nemusíte provádět pro každou aplikaci, která se připojuje ke správci front. Jedna definice kanálu pro připojení k serveru a jeden modul listener mohou podporovat všechny aplikace, které se připojují v režimu klienta.

4. Je-li vaše aplikace aplikací typu publikování-odběr a používá rozhraní publikování/odběru ve frontě, proveďte následující kroky.

- a) Ve správci front vytvořte systémové fronty produktu IBM MQ classes for JMS spuštěním skriptu příkazů MQSC, které jsou dodávány s produktem IBM MQ. Ujistěte se, že má ID uživatele přidružené k IBM Integration Bus nebo WebSphere Message Broker oprávnění pro přístup k frontám.

Informace o tom, kde najít skript a jak jej spustit, najdete v tématu [Použití modulu IBM MQ classes for Java](#).

Tento krok proveďte pouze jednou pro správce front. Stejnou sadu systémových front produktu IBM MQ classes for JMS může podporovat všechny aplikace XMS a IBM MQ classes for JMS, které se připojují ke správci front.

- b) Udělte jménu uživatele přidruženému k aplikaci oprávnění pro přístup k systémovým frontám produktu IBM MQ classes for JMS.

Informace o tom, jaké oprávnění ID uživatele potřebuje, najdete v tématu [Použití modulu IBM MQ classes for JMS](#).

- c) Pro zprostředkovatele produktu IBM Integration Bus nebo WebSphere Message Broker vytvořte a implementujte tok zpráv pro obsluhu fronty, kde aplikace odesílají zprávy, které publikují.

Základní tok zpráv se skládá z uzlu zpracování zpráv MQInput pro čtení publikovaných zpráv a uzlu zpracování zpráv publikování, aby bylo možné publikovat zprávy.

Informace o tom, jak vytvořit a implementovat tok zpráv, naleznete v dokumentaci produktu IBM Integration Bus nebo WebSphere Message Broker, která je k dispozici na webu [Webová stránka knihovny dokumentace produktu IBM Integration Bus](#).

Tento krok není třeba provést, pokud je na zprostředkovateli již implementován vhodný tok zpráv.

Výsledky

Nyní můžete svou aplikaci spustit.

Související úlohy

[Konfigurace zprostředkovatele pro aplikaci, která používá připojení v reálném čase ke zprostředkovateli](#)
Než budete moci spustit aplikaci, která používá v reálném čase připojení ke zprostředkovateli, musíte tento zprostředkovatel nakonfigurovat.

[Konfigurace sběrnice pro integraci služeb pro aplikaci, která se připojuje k produktu WebSphere Application Server](#)

Než budete moci spustit aplikaci, která se připojuje ke sběrnici pro integraci služeb produktu WebSphere Application Server service integration technologies, musíte nakonfigurovat integraci služeb stejným způsobem, jako nakonfigurujete sběrnici pro integraci služeb ke spuštění aplikací produktu JMS, které používají výchozího poskytovatele systému zpráv.

Konfigurace zprostředkovatele pro aplikaci, která používá připojení v reálném čase ke zprostředkovateli

Než budete moci spustit aplikaci, která používá v reálném čase připojení ke zprostředkovateli, musíte tento zprostředkovatel nakonfigurovat.

Než začnete

Před spuštěním této úlohy proveďte následující kroky:

- Ujistěte se, že vaše aplikace má přístup ke zprostředkovateli, který je spuštěn.
- Přesvědčte se, že vaše aplikace používá továrnu připojení, jejíž vlastnosti jsou správně nastaveny pro připojení v reálném čase ke zprostředkovateli. Další informace o vlastnostech továrny připojení viz [“Vlastnosti objektu ConnectionFactory” na stránce 175](#).

Informace o této úloze

Zprostředkovatele lze konfigurovat tak, aby spouštěl aplikace produktu XMS stejným způsobem, jako jste konfigurovali zprostředkovatele tak, aby spouštěl aplikace produktu IBM MQ classes for JMS . Následující kroky shrnují to, co je třeba provést:

Postup

1. Vytvořte a implementujte tok zpráv pro čtení zpráv z portu TCP/IP, na kterém zprostředkovatel naslouchá a publikuje zprávy.

Tuto možnost můžete provést jedním z následujících způsobů:

- Vytvořte tok zpráv, který obsahuje uzel zpracování zpráv produktu **Real-timeOptimizedFlow** .
- Vytvořte tok zpráv, který obsahuje uzel zpracování zpráv produktu **Real-timeInput** a uzel zpracování zpráv publikování.

Chcete-li naslouchat na portu použitým pro připojení v reálném čase, musíte nakonfigurovat uzel **Real-timeOptimizedFlow** nebo **Real-timeInput** . V produktu XMS je výchozí číslo portu pro připojení v reálném čase 1506.

Tento krok není třeba provést, pokud je na zprostředkovateli již implementován vhodný tok zpráv.

2. Pokud vyžadujete doručení zpráv do vaší aplikace pomocí produktu IBM MQ classes for JMS, nakonfigurujte zprostředkovatele tak, aby povolovaly výběrové vysílání. Konfigurujte témata, která musí být povolena multicast, a zadejte spolehlivou kvalitu služeb pro tato témata vyžadující spolehlivé výběrové vysílání.
3. Pokud vaše aplikace dodá ID uživatele a heslo, když se připojuje ke zprostředkovateli, a chcete, aby zprostředkovatel ověřoval vaši aplikaci pomocí těchto informací, nakonfigurujte server jmen uživatelů a zprostředkovatele pro jednoduché ověření pomocí hesla telnet.

Výsledky

Nyní můžete svou aplikaci spustit.

Související úlohy

[Konfigurace správce front a zprostředkovatele pro aplikaci, která se připojuje ke správci front produktu IBM MQ](#)

Tento oddíl předpokládá, že používáte produkt IBM WebSphere MQ 7.0.1 nebo novější. Před spuštěním aplikace, která se připojuje ke správci front produktu IBM MQ , je třeba nakonfigurovat správce front. V případě aplikace typu publikování/odběr je vyžadována některá další konfigurace, pokud používáte rozhraní publikování/odběru ve frontě.

[Konfigurace sběrnice pro integraci služeb pro aplikaci, která se připojuje k produktu WebSphere Application Server](#)

Než budete moci spustit aplikaci, která se připojuje ke sběrnici pro integraci služeb produktu WebSphere Application Server service integration technologies , musíte nakonfigurovat integraci služeb stejným způsobem, jako nakonfigurujete sběrnici pro integraci služeb ke spuštění aplikací produktu JMS , které používají výchozího poskytovatele systému zpráv.

Konfigurace sběrnice pro integraci služeb pro aplikaci, která se připojuje k produktu WebSphere Application Server

Než budete moci spustit aplikaci, která se připojuje ke sběrnici pro integraci služeb produktu WebSphere Application Server service integration technologies , musíte nakonfigurovat integraci služeb stejným způsobem, jako nakonfigurujete sběrnici pro integraci služeb ke spuštění aplikací produktu JMS , které používají výchozího poskytovatele systému zpráv.

Než začnete

Před spuštěním této úlohy musíte provést následující kroky:

- Ujistěte se, že je vytvořena sběrnice systému zpráv a že je váš server přidán do sběrnice jako člen sběrnice.
- Ujistěte se, že aplikace má přístup ke sběrnici pro integraci služeb, která obsahuje alespoň jeden stroj systému zpráv, který je spuštěný.
- Je-li vyžadována operace HTTP, musí být definován příchozí transportní kanál stroje systému zpráv HTTP. Při výchozím nastavení jsou kanály pro SSL a TCP definovány během instalace serveru.
- Přesvědčte se, že vaše aplikace používá továrnu připojení, jejíž vlastnosti jsou správně nastaveny pro připojení ke sběrnici pro integraci služeb s použitím serveru samozavedení. Minimální požadované informace jsou:
 - Koncový bod poskytovatele, který popisuje umístění a protokol, který má být použit při vyjednávání připojení k serveru systému zpráv (tj. prostřednictvím samozaváděcího serveru). Ve svém nejjednodušším tvaru pro server instalovaný s výchozími nastaveními může být koncový bod nastaven na název hostitele serveru.
 - Název sběrnice, přes kterou se odesílají zprávy.

Další informace o vlastnostech továrny připojení viz [“Vlastnosti objektu ConnectionFactory” na stránce 175](#).

Informace o této úloze

Je třeba definovat všechny prostory front nebo témat, které požadujete. Při výchozím nastavení je prostor tématu s názvem Default.Topic.Space definován během instalace serveru, ale pokud vyžadujete další prostory tématu, musíte tyto prostory témat vytvořit sami. Jednotlivá témata v rámci prostoru tématu není třeba předdefinovat, protože server podle potřeby vytváří instanci těchto jednotlivých témat.

Následující kroky shrnují to, co je třeba provést.

Postup

1. Vytvořte fronty, které vaše aplikace potřebuje pro systém zpráv typu point-to-point.
2. Vytvořte všechny další prostory tématu, které vaše aplikace potřebuje pro systém zpráv publikování/ odběru.

Výsledky

Nyní můžete svou aplikaci spustit.

Související úlohy

[Konfigurace správce front a zprostředkovatele pro aplikaci, která se připojuje ke správci front produktu IBM MQ](#)

Tento oddíl předpokládá, že používáte produkt IBM WebSphere MQ 7.0.1 nebo novější. Před spuštěním aplikace, která se připojuje ke správci front produktu IBM MQ , je třeba nakonfigurovat správce front. V případě aplikace typu publikování/odběr je vyžadována některá další konfigurace, pokud používáte rozhraní publikování/odběru ve frontě.

[Konfigurace zprostředkovatele pro aplikaci, která používá připojení v reálném čase ke zprostředkovateli](#)

Než budete moci spustit aplikaci, která používá v reálném čase připojení ke zprostředkovateli, musíte tento zprostředkovatel nakonfigurovat.

Použití ukázkových aplikací produktu XMS

Použijte ukázkové aplikace dodávané s produktem XMS k ověření instalace a nastavení serveru systému zpráv a k usnadnění sestavování vlastních aplikací. Ukázky poskytují přehled běžných funkcí jednotlivých rozhraní API.

Související pojmy

Webová služba vyhledávání JNDI

Chcete-li přistupovat k adresáři pojmenování COS z produktu XMS, musí být na serveru WebSphere Application Server service integration bus implementována webová služba vyhledávání produktu JNDI . Tato webová služba přeloží informace o produktu Java ze služby názvů COS do formuláře, který lze číst v aplikacích produktu XMS .

Nastavení prostředí serveru systému zpráv

Témata v této sekci popisují, jak nastavit prostředí serveru systému zpráv, aby bylo aplikacím produktu XMS umožněno připojit se k serveru.

Související úlohy

Instalace produktu Message Service Client for .NET pomocí průvodce instalací

Instalace používá instalační program produktu InstallShield X/Windows MSI. K dispozici jsou dvě volby nastavení, takže můžete zvolit buď úplnou, nebo vlastní instalaci.

Ukázkové aplikace

Ukázkové aplikace poskytují přehled běžných funkcí jednotlivých rozhraní API. Můžete je použít k ověření instalace a nastavení serveru systému zpráv a k usnadnění sestavení vašich vlastních aplikací.

Potřebujete-li pomoci při vytváření vlastních aplikací, můžete ukázkové aplikace použít jako výchozí bod. Pro každou aplikaci je k dispozici jak zdrojový, tak i kompilovaná verze. Přezkoumejte vzorový zdrojový kód a identifikujte klíčové kroky pro vytvoření každého požadovaného objektu pro vaši aplikaci (ConnectionFactory, Connection, Session, Destination, and a Producer, or a Consumer, nebo obojí) a pro nastavení libovolných specifických vlastností, které jsou potřeba k uvedení, jak chcete, aby vaše aplikace fungovala. Další informace naleznete v části [“Zápis aplikací XMS”](#) na stránce 20. Ukázky se mohou měnit v budoucích verzích produktu XMS.

V následující tabulce jsou uvedeny tři sady ukázkových aplikací (jeden pro každé rozhraní API), které jsou dodávány s produktem XMS.

Název vzorku	Popis
SampleConsumerCS	Aplikace spotřebitele zpráv, která přijímá zprávy z fronty nebo se přihlašuje k odběru tématu.
SampleProducerCS	Aplikace producenta zpráv, která produkuje zprávy do fronty nebo na téma.
SampleConfigCS	Konfigurační aplikace, kterou lze použít k vytvoření spravovaného úložiště objektů, které je založené na souborech. Aplikace obsahuje továrnu připojení a místo určení pro vaše konkrétní nastavení připojení. Toto spravované úložiště objektů lze poté použít spolu s každým vzorovým odběratelským a producentem aplikací.

Ukázky, které podporují stejné funkce v různých rozhraních API, mají syntaktické rozdíly.

- Ukázkový spotřebitel zpráv a aplikace producenta podporují následující funkce:
 - Připojení k produktu IBM MQ, IBM Integration Bus (pomocí připojení v reálném čase ke zprostředkovateli) a WebSphere Application Server service integration bus
 - Vyhledávání v úložišti spravovaných objektů pomocí počátečního rozhraní kontextu

- Připojení k frontám (IBM MQ a WebSphere Application Server service integration bus) a témata (IBM MQ, připojení v reálném čase ke zprostředkovateli a WebSphere Application Server service integration bus)
- Základní, bajt, mapa, objekt, proud a textové zprávy
- Ukázková aplikace pro spotřebitele zpráv podporuje synchronní a asynchronní režimy příjmu a příkazy selektoru SQL.
- Ukázková aplikace Producent zpráv podporuje trvalé a dočasné režimy doručení.

Provozní režimy

Ukázky mohou pracovat v jednom ze dvou režimů:

Jednoduchý režim

Ukázky můžete spustit s minimálním uživatelským vstupem.

rozšířený režim

Můžete přizpůsobit jemněji způsob, jakým ukázky fungují.

Všechny vzorky jsou kompatibilní a mohou tedy fungovat v různých jazycích.

Související pojmy

Vytváření vlastních aplikací

Sestavujete své vlastní aplikace, jako např. sestavení ukázkových aplikací.

Související úlohy

Spuštění ukázkových aplikací

Ukázkové aplikace produktu .NET můžete spustit interaktivně v jednoduchém nebo rozšířeném režimu nebo neinteraktivně pomocí automaticky generovaných nebo přizpůsobených souborů odpovědí.

Sestavení ukázkových aplikací produktu .NET

Při sestavování ukázkové aplikace produktu .NET se vytvoří spustitelná verze vybrané ukázky.

Spuštění ukázkových aplikací

Ukázkové aplikace produktu .NET můžete spustit interaktivně v jednoduchém nebo rozšířeném režimu nebo neinteraktivně pomocí automaticky generovaných nebo přizpůsobených souborů odpovědí.

Než začnete

Před spuštěním některé z dodaných ukázkových aplikací musíte nejprve nastavit prostředí serveru systému zpráv, aby se aplikace mohly připojit k serveru. Viz [“Nastavení prostředí serveru systému zpráv” na stránce 12.](#)

Postup

Chcete-li spustit ukázkovou aplikaci .NET, proveďte následující kroky:

Tip: Když spouštíte ukázkovou aplikaci, zadejte? v každém čase získat pomoc s tím, co dělat dál.

1. Vyberte režim, ve kterém chcete spustit ukázkovou aplikaci.

Napište buď `Advanced`, nebo `Simple`.

2. Odpovězte na otázky.

Chcete-li vybrat výchozí hodnotu, která se zobrazí v hranatých závorkách na konci otázky, stiskněte klávesu `Enter`. Chcete-li vybrat jinou hodnotu, zadejte příslušnou hodnotu a stiskněte klávesu `Enter`.

Zde je příklad otázky:

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

V tomto případě je výchozí hodnota wpm (připojení k WebSphere Application Server service integration bus).

Výsledky

Když spustíte ukázkové aplikace, soubory odpovědí se generují automaticky v aktuálním pracovním adresáři. Názvy souborů odpovědí jsou ve formátu *connection_type-sample_type.rsp*; například *wpm-producer.rsp*. Je-li to nutné, můžete vygenerovaný soubor odpovědí použít k opětovnému spuštění ukázkové aplikace se stejnými volbami, takže je nemusíte znovu zadávat.

Související pojmy

Ukázkové aplikace

Ukázkové aplikace poskytují přehled běžných funkcí jednotlivých rozhraní API. Můžete je použít k ověření instalace a nastavení serveru systému zpráv a k usnadnění sestavení vašich vlastních aplikací.

Související úlohy

Sestavení ukázkových aplikací produktu .NET

Při sestavování ukázkové aplikace produktu .NET se vytvoří spustitelná verze vybrané ukázky.

Sestavení ukázkových aplikací produktu .NET

Při sestavování ukázkové aplikace produktu .NET se vytvoří spustitelná verze vybrané ukázky.

Než začnete

Nainstalujte příslušný kompilátor. Tato úloha předpokládá, že jste nainstalovali produkt Microsoft Visual Studio 2012 a že jste obeznámeni s jeho používáním.

Postup

Chcete-li sestavit ukázkovou aplikaci .NET, proveďte následující kroky:

1. Klepněte na soubor řešení *Samples.sln* poskytnutý s ukázkami produktu .NET.
2. Klepněte pravým tlačítkem myši na řešení Ukázky v okně Průzkumník řešení a vyberte volbu **Sestavit řešení**.

Výsledky

Spustitelný program je vytvořen v příslušné podsložce ukázky, buď *bin/Debug*, nebo *bin/Release*, v závislosti na konfiguraci, kterou jste zvolili. Tento program má stejný název jako složka, s příponou *CS*. Například, pokud sestavujete verzi *C#* ukázkové aplikace producenta zpráv, produkt *SampleProducerCS.exe* se vytvoří ve složce *SampleProducer*.

Související pojmy

Ukázkové aplikace

Ukázkové aplikace poskytují přehled běžných funkcí jednotlivých rozhraní API. Můžete je použít k ověření instalace a nastavení serveru systému zpráv a k usnadnění sestavení vašich vlastních aplikací.

[“Vytváření vlastních aplikací” na stránce 41](#)

Sestavujete své vlastní aplikace, jako např. sestavení ukázkových aplikací.

Související úlohy

Spuštění ukázkových aplikací

Ukázkové aplikace produktu .NET můžete spustit interaktivně v jednoduchém nebo rozšířeném režimu nebo neinteraktivně pomocí automaticky generovaných nebo přizpůsobených souborů odpovědí.

Vývoj aplikací produktu XMS

Témata v této sekci obsahují informace, které můžete najít užitečné při zápisu aplikací XMS.

Informace o zápisu aplikací XMS naleznete v následujících tématech:

Zápis aplikací XMS

Témata v této sekci obsahují informace, které vám pomohou při psaní aplikací produktu XMS .

Tento oddíl obsahuje obecné koncepty pro zápis aplikací produktu XMS . Viz také [“Zápis aplikací produktu XMS .NET”](#) na stránce 42 pro informace specifické pro vytváření aplikací .NET .

Tento oddíl obsahuje následující témata:

- [“Model vytváření podprocesů”](#) na stránce 20
- [“ConnectionFactories a objekty připojení”](#) na stránce 21
- [“Relace”](#) na stránce 23
- [“Cíle”](#) na stránce 27
- [“Producenti zpráv”](#) na stránce 32
- [“Spotřebitelé zpráv”](#) na stránce 32
- [“Prohlížeče front”](#) na stránce 36
- [“Žadatelé”](#) na stránce 36
- [“Odstranění objektu”](#) na stránce 36
- [“XMS primitivní typy”](#) na stránce 37
- [“Implicitní převod hodnoty vlastnosti z jednoho datového typu na jiný.”](#) na stránce 38
- [“iterátory”](#) na stránce 41
- [“Identifikátory kódované znakové sady”](#) na stránce 41
- [“Chybové kódy a kódy výjimek produktu XMS”](#) na stránce 41
- [“Vytváření vlastních aplikací”](#) na stránce 41

Související pojmy

Zápis aplikací produktu XMS .NET

Témata v této sekci obsahují informace, které vám pomohou při psaní aplikací produktu XMS .NET .

Související odkazy

[.NETRozhraní](#)

Tato sekce dokumentuje rozhraní třídy .NET a jejich vlastnosti a metody.

Model vytváření podprocesů

Obecná pravidla řídí, jak vícevláknová aplikace může používat objekty XMS .

- Souběžně s různými podprocesy mohou být použity pouze objekty následujících typů:
 - ConnectionFactory
 - Připojení
 - Data ConnectionMeta
 - Místo určení
- Objekt relace může být v jednom okamžiku použit pouze na jednom podprocesu.

Výjimky z těchto pravidel jsou označeny položkami opatřeným popiskem "Kontext podprocesu" v definicích rozhraní metod v produktu [“Klienti Message Service Clients for .NET”](#) na stránce 86.

Související pojmy

[Chybové stavy, které lze zpracovat za běhu](#)

Návratové kódy z volání API jsou chybové stavy, které lze zpracovat za běhu. Způsob, jakým se zabývá tímto typem chyby, závisí na tom, zda používáte rozhraní API jazyka C nebo Jazyk C++ .

ConnectionFactoryy a objekty připojení

Objekt `ConnectionFactory` poskytuje šablonu, kterou aplikace používá k vytvoření objektu připojení. Aplikace používá objekt `Connection` k vytvoření objektu relace.

Pro produkt .NET používá aplikace XMS nejprve objekt `XMSFactoryFactory` k získání odkazu na objekt `ConnectionFactory`, který odpovídá požadovanému typu protokolu. Tento objekt `ConnectionFactory` pak může vytvářet připojení pouze pro daný typ protokolu.

Aplikace produktu XMS může vytvořit více připojení a aplikace s podporou podprocesů může souběžně používat jeden objekt `Connection` ve více podprocesech. Objekt `Connection` zapouzdřuje komunikační spojení mezi aplikací a serverem systému zpráv.

Spojení slouží několika účelům:

- Když aplikace vytvoří připojení, může být ověřena její identita.
- Aplikace může přidružovat jedinečný identifikátor klienta k připojení. Identifikátor klienta se používá k podpoře trvalých odběrů v doméně publikování/odběru. Identifikátor klienta lze nastavit dvěma způsoby:

Preferovaný způsob přiřazení identifikátoru klienta připojení je konfigurovat v objektu `ConnectionFactory` specifického pro klienta pomocí vlastností a transparentně přiřadit k připojení, které vytvoří.

Alternativním způsobem přiřazení identifikátoru klienta je použití hodnoty specifické pro poskytovatele, která je nastavena na objektu připojení. Tato hodnota nepřepíše identifikátor, který byl administrativně konfigurován. Je k dispozici pro případ, kdy žádný administrativně uvedený identifikátor neexistuje. Pokud existuje administrativně zadaný identifikátor, pokus o jeho přepsání na základě hodnoty specifické pro poskytovatele způsobí vyvolání výjimky. Pokud aplikace explicitně nastavuje identifikátor, musí to provést okamžitě po vytvoření připojení a před tím, než je provedena nějaká jiná akce na připojení; jinak je vyvolána výjimka.

Aplikace XMS typicky vytváří připojení, jednu nebo více relací a několik výrobců zpráv a spotřebitelů zpráv.

Vytvoření připojení je relativně nákladné z hlediska systémových prostředků, protože zahrnuje vytvoření komunikačního připojení a může také zahrnovat ověření aplikace.

Související úlohy

Vytvoření spravovaných objektů

Definice objektu `ConnectionFactory` a cílový objekt, které aplikace produktu XMS vyžadují k vytvoření připojení k serveru systému zpráv, musí být vytvořeny pomocí příslušných nástrojů pro správu.

Související odkazy

[IConnectionFactory](#) (pro rozhraní .NET)

Aplikace používá továrnu připojení k vytvoření připojení.

Vlastnosti objektu `ConnectionFactory`

Přehled vlastností objektu `ConnectionFactory` s odkazy na podrobnější referenční informace.

[IDestination](#) (pro rozhraní .NET)

Cíl je místo, kam aplikace odesílá zprávy, nebo je to zdroj, ze kterého aplikace přijímá zprávy, nebo obojí.

Vlastnosti místa určení

Přehled vlastností objektu `Destination`, s odkazy na podrobnější referenční informace.

Spuštěný a zastavený režim připojení

Připojení může fungovat buď v režimu spuštění, nebo v zastaveném režimu.

Když aplikace vytvoří připojení, je připojení v zastaveném režimu. Když se připojení nachází v zastaveném režimu, může aplikace inicializovat relace a může odesílat zprávy, ale nemůže je přijímat, a to synchronně nebo asynchronně.

Aplikace může spustit připojení voláním metody `Start Connection`. Je-li připojení spuštěno v režimu spuštění, může aplikace odesílat a přijímat zprávy. Aplikace pak může zastavit a znovu spustit připojení voláním metody `Stop Connection` a `Start Connection`.

Související pojmy

Uzavření připojení

Aplikace zavře připojení voláním metody Close Connection.

Zpracování výjimek

Pokud aplikace používá připojení pouze ke spotřebování zpráv asynchronně, zjišťuje se pouze o problému s připojením pouze pomocí modulu listener pro výjimky.

Připojení ke sběrnici pro integraci služeb

Aplikace XMS se může připojit ke sběrnici pro integraci služeb produktu WebSphere Application Server buď pomocí přímého připojení TCP/IP, nebo pomocí protokolu HTTP přes TCP/IP.

Uzavření připojení

Aplikace zavře připojení voláním metody Close Connection.

Když aplikace uzavře připojení, XMS provede následující akce:

- Zavře všechny relace přidružené k tomuto připojení a odstraní určité objekty přidružené k těmto relacím. Další informace o tom, které objekty se odstraní, viz [“Odstranění objektu” na stránce 36](#). Produkt XMS současně odvolá všechny probíhající transakce v rámci relací.
- Končí komunikační spojení se serverem systému zpráv.
- Uvolní paměť a další vnitřní prostředky použité připojením.

Produkt XMS nepotvrdí příjem žádných zpráv, které během relace nepotvrdil, a to před uzavřením připojení. Další informace o potvrzení příjmu zpráv viz [“Potvrzení zprávy” na stránce 24](#).

Související pojmy

Spuštěný a zastavený režim připojení

Připojení může fungovat buď v režimu spuštění, nebo v zastaveném režimu.

Zpracování výjimek

Pokud aplikace používá připojení pouze ke spotřebování zpráv asynchronně, zjišťuje se pouze o problému s připojením pouze pomocí modulu listener pro výjimky.

Připojení ke sběrnici pro integraci služeb

Aplikace XMS se může připojit ke sběrnici pro integraci služeb produktu WebSphere Application Server buď pomocí přímého připojení TCP/IP, nebo pomocí protokolu HTTP přes TCP/IP.

Zpracování výjimek

Pokud aplikace používá připojení pouze ke spotřebování zpráv asynchronně, zjišťuje se pouze o problému s připojením pouze pomocí modulu listener pro výjimky.

Výjimky produktu XMS .NET jsou odvozeny od výjimky System.Exception. Další informace naleznete v části [“Ošetření chyb v produktu .NET” na stránce 46](#).

Související pojmy

Spuštěný a zastavený režim připojení

Připojení může fungovat buď v režimu spuštění, nebo v zastaveném režimu.

Uzavření připojení

Aplikace zavře připojení voláním metody Close Connection.

Připojení ke sběrnici pro integraci služeb

Aplikace XMS se může připojit ke sběrnici pro integraci služeb produktu WebSphere Application Server buď pomocí přímého připojení TCP/IP, nebo pomocí protokolu HTTP přes TCP/IP.

Připojení ke sběrnici pro integraci služeb

Aplikace XMS se může připojit ke sběrnici pro integraci služeb produktu WebSphere Application Server buď pomocí přímého připojení TCP/IP, nebo pomocí protokolu HTTP přes TCP/IP.

Protokol HTTP může být použit v situacích, kdy přímé připojení TCP/IP není možné. Jedna běžná situace je při komunikaci přes ochrannou bariéru (firewall), jako například když si dva podniky vyměňují zprávy. Použití HTTP ke komunikaci přes ochrannou bariéru se často označuje jako *tunelové propojení HTTP*.

Tunelování HTTP je však přirozeně pomalejší než použití přímého připojení TCP/IP, protože se hlavičky HTTP výrazně přidávají k množství dat, která jsou přenášena, a protože protokol HTTP vyžaduje více komunikačních toků než TCP/IP.

Chcete-li vytvořit připojení TCP/IP, může aplikace použít továrnu připojení, jejíž vlastnost `XMSCS_WPM_TARGET_TRANSPORT_CHAIN` je nastavena na hodnotu `XMSCS_WPM_TARGET_TRANSPORT_CHAIN_BASIC`. Jedná se o výchozí hodnotu vlastnosti. Je-li připojení úspěšně vytvořeno, vlastnost `XMSCS_WPM_CONNECTION_PROTOCOL` připojení je nastavena na hodnotu `XMSCS_WPM_CP_TCP`.

Chcete-li vytvořit připojení používající protokol HTTP, musí aplikace používat továrnu připojení, jejíž vlastnost `XMSCS_WPM_TARGET_TRANSPORT_CHAIN` je nastavena na název příchozího transportního řetězu, který je konfigurován pro použití transportního kanálu protokolu HTTP. Je-li připojení úspěšně vytvořeno, vlastnost `XMSCS_WPM_CONNECTION_PROTOCOL` připojení je nastavena na hodnotu `XMSCS_WPM_CP_HTTP`. Informace o způsobu konfigurace řetězů přenosu naleznete v tématu [Konfigurace řetězů přenosu](#) v dokumentaci produktu WebSphere Application Server .

Aplikace má při připojování k samozaváděcímu serveru podobnou volbu komunikačních protokolů. Vlastnost `XMSCS_WPM_PROVIDER_ENDPOINTS` továrny připojení je posloupností jedné nebo více adres koncových bodů serverů samozavedení. Komponenta transportního řetězu samozavedení každé adresy koncového bodu může být buď `XMSCS_WPM_BOOTSTRAP_TCP`, pro připojení TCP/IP k serveru samozavedení nebo `XMSCS_WPM_BOOTSTRAP_HTTP` pro připojení, které používá protokol HTTP.

Související pojmy

Spuštěný a zastavený režim připojení

Připojení může fungovat buď v režimu spuštění, nebo v zastaveném režimu.

Uzavření připojení

Aplikace zavře připojení voláním metody `Close Connection`.

Zpracování výjimek

Pokud aplikace používá připojení pouze ke spotřebování zpráv asynchronně, zjišťuje se pouze o problému s připojením pouze pomocí modulu listener pro výjimky.

Související úlohy

Vytvoření spravovaných objektů

Definice objektu `ConnectionFactory` a cílový objekt, které aplikace produktu XMS vyžadují k vytvoření připojení k serveru systému zpráv, musí být vytvořeny pomocí příslušných nástrojů pro správu.

Související odkazy

`ICConnectionFactory` (pro rozhraní .NET)

Aplikace používá továrnu připojení k vytvoření připojení.

Vlastnosti objektu `ConnectionFactory`

Přehled vlastností objektu `ConnectionFactory` s odkazy na podrobnější referenční informace.

`IDestination` (pro rozhraní .NET)

Cíl je místo, kam aplikace odesílá zprávy, nebo je to zdroj, ze kterého aplikace přijímá zprávy, nebo obojí.

Vlastnosti místa určení

Přehled vlastností objektu `Destination`, s odkazy na podrobnější referenční informace.

Relace

Relace je jednovláknový kontext pro odesílání a příjem zpráv.

Aplikace může použít relaci k vytváření zpráv, producentů zpráv, spotřebitelů zpráv, prohlížečů fronty a dočasných míst určení. Aplikace může také použít relaci ke spuštění lokálních transakcí.

Aplikace může vytvořit více relací, kde každá relace produkuje a spotřebovává zprávy nezávisle na ostatních relacích. Pokud se dva spotřebitelé zpráv v samostatných relacích (nebo dokonce ve stejné relaci) přihlásí k odběru stejného tématu, obdrží každá z nich kopii libovolné zprávy publikované k tomuto tématu.

Na rozdíl od objektu připojení nemůže být objekt relace použit souběžně na různých podprocesech. Z jiného podprocesu lze volat pouze metodu Close objektu relace z jiného podprocesu, než je podproces, který objekt Session používá. Metoda Close Session ukončí relaci a uvolní všechny systémové prostředky alokované pro relaci.

Pokud musí aplikace souběžně zpracovávat zprávy ve více než jednom podprocesu, aplikace musí vytvořit relaci na každém podprocesu a poté tuto relaci použít pro každou operaci odeslání nebo přijetí v rámci tohoto podprocesu.

Překonání relace

Aplikace produktu XMS mohou spouštět lokální transakce. *Lokální transakce* je transakce, která zahrnuje změny pouze na prostředky správce front nebo sběrnice pro integraci služeb, ke které je aplikace připojena.

Informace v tomto tématu jsou relevantní pouze v případě, že se aplikace připojuje ke správci front produktu IBM MQ nebo ke sběrnici SIBus produktu WebSphere Application Server . Informace nejsou relevantní pro připojení v reálném čase ke zprostředkovateli.

Chcete-li spustit lokální transakce, musí aplikace nejprve vytvořit relaci s transakcemi voláním metody Vytvořit relaci objektu připojení a uvést jako parametr, že relace je zpracovávána. Následně jsou všechny zprávy odeslané a přijaté v rámci relace seskupeny do posloupnosti transakcí. Transakce je ukončena, když aplikace potvrdí nebo odvolá zprávy, které odeslala a obdržela od začátku transakce.

Pro potvrzení transakce volá aplikace metodu Commit objektu Session. Když je transakce potvrzena, všechny zprávy odeslané v rámci transakce se stanou dostupnými pro doručení do jiných aplikací a všechny zprávy přijaté v rámci transakce jsou potvrzeny, aby se server systému zpráv nepokoušel o jejich doručení do aplikace znovu. V doméně typu point-to-point server zpráv také odebírá přijaté zprávy z jejich front.

Chcete-li odvolat transakci, aplikace volá metodu Rollback objektu Session. Když je transakce odvolána, všechny zprávy odeslané v rámci transakce jsou vyřazeny serverem systému zpráv a všechny zprávy přijaté v rámci transakce budou znovu k dispozici pro doručení. V dvoubodové doméně se zprávy, které byly přijaty, vrátí zpět do front a znovu se stanou viditelnými pro jiné aplikace.

Nová transakce se spustí automaticky, když aplikace vytvoří transakci s transakcemi nebo zavolá metodu Commit nebo Rollback. Proto má relace, která obsahuje transakci, vždy aktivní transakci.

Když aplikace zavře relaci transakce, dojde k implicitnímu odvolání transakce. Když aplikace uzavře připojení, dojde k implicitnímu odvolání transakce pro všechny relace, které se v relaci nachází.

Transakce je zcela obsažena v rámci relace transakce. Transakce nemůže přesahovat relace. To znamená, že aplikace nemůže odesílat a přijímat zprávy ve dvou nebo více transakčních relacích, a poté potvrdit nebo odvolat všechny tyto akce jako jedinou transakci.

Související pojmy

Potvrzení zprávy

Každá relace, která nemá transakci, má režim potvrzení, který určuje, jak jsou přijímány zprávy přijaté aplikací. K dispozici jsou tři režimy potvrzení a volba režimu potvrzení má vliv na návrh aplikace.

Asynchronní doručení zprávy

Produkt XMS používá jeden podproces k obsluze všech asynchronních doručení zpráv pro danou relaci. To znamená, že v daném okamžiku může běžet pouze jedna funkce modulu listener pro zprávy nebo jedna metoda `onMessage()` .

Doručení synchronní zprávy

Zprávy se doručí synchronně aplikaci, pokud aplikace používá metody příjmu objektů `MessageConsumer` .

Režim doručení zprávy

Produkt XMS podporuje dva režimy doručování zpráv.

Potvrzení zprávy

Každá relace, která nemá transakci, má režim potvrzení, který určuje, jak jsou přijímány zprávy přijaté aplikací. K dispozici jsou tři režimy potvrzení a volba režimu potvrzení má vliv na návrh aplikace.

Informace v tomto tématu jsou relevantní pouze v případě, že se aplikace připojuje ke správci front produktu IBM MQ nebo ke sběrnici SIBus produktu WebSphere Application Server . Informace nejsou relevantní pro připojení v reálném čase ke zprostředkovateli.

Produkt XMS používá stejný mechanismus pro potvrzení příjmu zpráv, které platforma JMS používá.

Pokud relace není součástí transakce, je způsob, jakým zprávy přijaté aplikací potvrdí, je určen režimem potvrzení relace. Tři režimy potvrzení jsou popsány v následujících odstavcích:

XMSC_AUTO_ACKNOWLEDGE

Relace automaticky potvrdí každou zprávu přijatou aplikací.

Pokud jsou zprávy doručovány synchronně do aplikace, relace potvrdí příjem zprávy pokaždé, když je úspěšně dokončeno přijetí volání.

Pokud aplikace obdrží zprávu úspěšně, ale selhání zabránilo provedení potvrzení, bude zpráva znovu k dispozici pro doručení. Aplikace musí být proto schopna zpracovat zprávu, která je znovu doručena.

XMSC_DUPPS_OK_ACKNOWLEDGE

Relace bere na vědomí zprávy přijaté aplikací v době, kdy je vybrána.

Použití tohoto režimu potvrzení snižuje množství práce, které musí relace dělat, ale selhání zabraňující potvrzení zprávy může vést k tomu, že bude znovu k dispozici více než jedna zpráva k doručení. Aplikace musí být proto schopna zpracovávat zprávy, které jsou znovu doručeny.

XMSC_KLIENTA_

Aplikace potvrdí zprávy, které obdrží, voláním metody Acknowledge třídy Message.

Aplikace může potvrdit příjem každé zprávy jednotlivě, nebo může obdržet dávku zpráv a zavolat metodu Potvrdit pouze pro poslední zprávu, kterou obdrží. Když se metoda Acknowledge nazývá všechny zprávy přijaté od poslední doby, kdy byla metoda volána, jsou potvrzeny.

Ve spojení s jakýmkoli z těchto režimů potvrzení může aplikace zastavit a znovu spustit doručování zpráv v relaci voláním metody Recover třídy Session. Zprávy, jejichž příjem byl dříve nepotvrzený, jsou znovu doručeny. Je však možné, že nebudou doručeny ve stejné posloupnosti, v jaké byly dříve doručeny. Do té doby mohla dorazit zpráva s vyšší prioritou a některé původní zprávy mohly vypršet. V doméně dvoubodového spojení mohly být některé z původních zpráv spotřebovány jinou aplikací.

Aplikace může určit, zda je zpráva znovu doručena, tím, že zkontrolujete obsah pole záhlaví JMSRedelivered zprávy. Aplikace to provede voláním metody Get JMSRedelivered třídy Message.

Související pojmy

Překonat relace

Aplikace produktu XMS mohou spouštět lokální transakce. *Lokální transakce* je transakce, která zahrnuje změny pouze na prostředky správce front nebo sběrnice pro integraci služeb, ke které je aplikace připojena.

Asynchronní doručení zprávy

Produkt XMS používá jeden podproces k obsluze všech asynchronních doručení zpráv pro danou relaci. To znamená, že v daném okamžiku může běžet pouze jedna funkce modulu listener pro zprávy nebo jedna metoda onMessage() .

Doručení synchronní zprávy

Zprávy se doručí synchronně aplikaci, pokud aplikace používá metody příjmu objektů MessageConsumer .

Režim doručení zprávy

Produkt XMS podporuje dva režimy doručování zpráv.

Asynchronní doručení zprávy

Produkt XMS používá jeden podproces k obsluze všech asynchronních doručení zpráv pro danou relaci. To znamená, že v daném okamžiku může běžet pouze jedna funkce modulu listener pro zprávy nebo jedna metoda onMessage() .

Pokud více než jeden spotřebitel zpráv v relaci přijímá zprávy asynchronně a funkce modulu listener pro zprávy nebo metoda onMessage() doručí zprávu spotřebiteli zpráv, musí všichni ostatní spotřebitelé

zpráv, kteří čekají na stejnou zprávu, pokračovat v čekání. Další zprávy, které čekají na doručení do relace, musí také pokračovat v čekání.

Pokud aplikace vyžaduje souběžné doručování zpráv, vytvořte více než jednu relaci, aby produkt XMS používal více než jeden podproces k obsluze asynchronního doručování zpráv. Tímto způsobem lze souběžně spustit více než jednu funkci modulu listener pro zprávy nebo metodu produktu `onMessage()`.

Relace není prováděna asynchronním přiřazením modulu listener pro zprávy ke spotřebiteli. Relace se stane asynchronně pouze tehdy, když je volána metoda `Connection.Start`. Všechna synchronní volání jsou povolena, dokud není volána metoda `Connection.Start`. Doručení zpráv pro spotřebitele se spustí, když se zavolá `Connection.Start`.

Pokud se musí na asynchronní relaci provést synchronní volání, jako např. vytvoření spotřebitele nebo producenta, musí být volána funkce `Connection.Stop`. Relaci lze obnovit voláním metody `Connection.Start` ke spuštění doručování zpráv. Jedinou výjimkou je podproces doručení zprávy relace, který je podproces, který doručuje zprávy do funkce zpětného volání. Tento podproces může provést jakékoli volání na relaci (kromě volání `Close`) v rámci funkce zpětného volání zprávy.

Poznámka: V režimu nespravovaného režimu není volání MQDISC v rámci funkce call-back podporováno klientem produktu IBM MQ .NET. Klientská aplikace tedy nemůže vytvářet nebo zavírat relace v rámci zpětného volání `MessageListener` v režimu asynchronního příjmu. Vytvořte a vyhodte relaci mimo metodu produktu `MessageListener`.

Související pojmy

Překonat relace

Aplikace produktu XMS mohou spouštět lokální transakce. *Lokální transakce* je transakce, která zahrnuje změny pouze na prostředky správce front nebo sběrnice pro integraci služeb, ke které je aplikace připojena.

Potvrzení zprávy

Každá relace, která nemá transakci, má režim potvrzení, který určuje, jak jsou přijímány zprávy přijaté aplikací. K dispozici jsou tři režimy potvrzení a volba režimu potvrzení má vliv na návrh aplikace.

Doručení synchronní zprávy

Zprávy se doručí synchronně aplikaci, pokud aplikace používá metody příjmu objektů `MessageConsumer`.

Režim doručení zprávy

Produkt XMS podporuje dva režimy doručování zpráv.

Doručení synchronní zprávy

Zprávy se doručí synchronně aplikaci, pokud aplikace používá metody příjmu objektů `MessageConsumer`.

Pomocí metod příjmu může aplikace čekat zadané časové období pro zprávu nebo může čekat nekonečně dlouhou dobu. Případně, pokud aplikace nechce čekat na zprávu, může použít metodu `Receive with No Wait`.

Související pojmy

Překonat relace

Aplikace produktu XMS mohou spouštět lokální transakce. *Lokální transakce* je transakce, která zahrnuje změny pouze na prostředky správce front nebo sběrnice pro integraci služeb, ke které je aplikace připojena.

Potvrzení zprávy

Každá relace, která nemá transakci, má režim potvrzení, který určuje, jak jsou přijímány zprávy přijaté aplikací. K dispozici jsou tři režimy potvrzení a volba režimu potvrzení má vliv na návrh aplikace.

Asynchronní doručení zprávy

Produkt XMS používá jeden podproces k obsluze všech asynchronních doručení zpráv pro danou relaci. To znamená, že v daném okamžiku může běžet pouze jedna funkce modulu listener pro zprávy nebo jedna metoda `onMessage()`.

Režim doručení zprávy

Produkt XMS podporuje dva režimy doručování zpráv.

Režim doručení zpráv

Produkt XMS podporuje dva režimy doručování zpráv.

- *Trvalé* zprávy jsou doručeny jednou. Server systému zpráv přijímá zvláštní opatření, jako je například protokolování zpráv, aby se zajistilo, že trvalé zprávy nebudou při přenosu ztraceny, a to i v případě selhání.
- *Netrvalé* zprávy jsou doručeny ne více než jednou. Netrvalé zprávy jsou méně spolehlivé než trvalé zprávy, protože mohou být při přenosu ztraceny v případě selhání.

Volba režimu doručení je kompromisem mezi spolehlivostí a výkonem. Přechnodné zprávy se obvykle přepravují rychleji než trvalé zprávy.

Související pojmy

Překonat relace

Aplikace produktu XMS mohou spouštět lokální transakce. *Lokální transakce* je transakce, která zahrnuje změny pouze na prostředky správce front nebo sběrnice pro integraci služeb, ke které je aplikace připojena.

Potvrzení zprávy

Každá relace, která nemá transakci, má režim potvrzení, který určuje, jak jsou přijímány zprávy přijaté aplikací. K dispozici jsou tři režimy potvrzení a volba režimu potvrzení má vliv na návrh aplikace.

Asynchronní doručení zprávy

Produkt XMS používá jeden podproces k obsluze všech asynchronních doručení zpráv pro danou relaci. To znamená, že v daném okamžiku může běžet pouze jedna funkce modulu listener pro zprávy nebo jedna metoda `onMessage()`.

Doručení synchronní zprávy

Zprávy se doručí synchronně aplikaci, pokud aplikace používá metody příjmu objektů `MessageConsumer`.

Cíle

Aplikace XMS používá objekt `Destination` k určení místa určení odesílaných zpráv a zdroje přijímaných zpráv.

Aplikace XMS může buď za běhu vytvořit cílový objekt, nebo může získat předdefinovaný cíl z úložiště spravovaných objektů.

Stejně jako u `ConnectionFactory` je nejflexibilnějším způsobem, jak aplikace XMS určit místo určení, definovat ji jako spravovaný objekt. Při použití tohoto přístupu mohou aplikace napsané v jazycích C, C++ a .NET a v produktu Javasdílet definice místa určení. Vlastnosti administrovaných cílových objektů lze změnit beze změny kódu.

Pro aplikace produktu .NET lze vytvořit místo určení pomocí metody `CreateTopic` nebo `CreateQueue`. Tyto dvě metody jsou k dispozici v objektech `ISession` a `XMSFactoryFactory` v rozhraní API produktu .NET. Další informace naleznete v tématech "[Cíle v .NET](#)" na stránce 44 a "[Instinace](#)" na stránce 104.

Související odkazy

IDestination (pro rozhraní .NET)

Cíl je místo, kam aplikace odesílá zprávy, nebo je to zdroj, ze kterého aplikace přijímá zprávy, nebo obojí.

Vlastnosti místa určení

Přehled vlastností objektu `Destination`, s odkazy na podrobnější referenční informace.

Jednotné identifikátory prostředků tématu

Identifikátor URI (Uniform Resource Identifier) tématu určuje název tématu; pro něj lze také zadat jednu či více vlastností.

Identifikátor URI pro téma začíná tématem posloupnosti: //, za nímž následuje jméno tématu a (volitelné) seznam dvojic název-hodnota, které nastavují zbývající vlastnosti tématu. Název tématu nemůže být prázdný.

Zde je příklad ve fragmentu kódu .NET :

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

Další informace o vlastnostech tématu, včetně názvu a platných hodnot, které můžete použít v identifikátoru URI, najdete v tématu [“Vlastnosti místa určení”](#) na stránce 181.

Při určování identifikátoru URI tématu pro použití v odběru lze použít zástupné znaky. Syntaxe těchto zástupných znaků závisí na typu připojení a verzi zprostředkovatele. K dispozici jsou následující volby:

- IBM WebSphere MQ 7.0 správce front se zástupným znakem pro úroveň znaků
- IBM WebSphere MQ 7.0 správce front s použitím formátu zástupných znaků na úrovni tématu
- WebSphere Application Server sběrnice SIBus

IBM WebSphere MQ 7.0 správce front se zástupným znakem pro úroveň znaků

IBM WebSphere MQ 7.0 správce front se zástupným znakem pro úroveň znaků používá následující zástupné znaky:

- * pro 0 nebo více znaků
- CO? pro 1 znak
- % pro řídicí znak

V části [Tabulka 1](#) na stránce 28 jsou uvedeny příklady použití tohoto schématu zástupných znaků.

Tabulka 1. Ukázkové identifikátory URI používající schéma zástupného znaku na úrovni znaků pro správce front produktu IBM WebSphere MQ 7.0

Jednotný identifikátor prostředku	Vyhovuje	Příklady
"topic://Sport *Výsledky"	Všechna témata začínající slovy "Sport" a končící slovy "Výsledky"	"topic://SportsResults" a "topic://Sport/Hockey/National/Div3/Results"
" topic://Sport?Výsledky "	Všechna témata začínající na "Sport" následovaná jedním znakem a za nimi následují "Výsledky"	"topics://SportsResults" a "topic://SportXResults"
"topic://Sport/ * Ball*/Div? / Results*/???"	Témata	"topic://Sport/Football/Div1/Results/2002/Nov" a "topic://Sport/Netball/National/Div3/Results/02/Jan"

IBM WebSphere MQ 7.0 správce front s použitím formátu zástupných znaků na úrovni tématu

Správce front IBM WebSphere MQ 7.0 s použitím formátu zástupných znaků na úrovni tématu používá následující zástupné znaky:

- # pro shodu s více úrovněmi
- + pro shodu s jednou úrovní

[Tabulka 2](#) na stránce 29 uvádí některé příklady použití tohoto schématu zástupného znaku.

Tabulka 2. Příklad identifikátorů URI používajících zástupný schéma na úrovni tématu pro správce front produktu IBM WebSphere MQ 7.0

Jednotný identifikátor prostředku	Vyhovuje	Příklady
"topic://Sport/ + / Results"	Všechna témata s jednoduchým hierarchickým názvem mezi sportem a výsledky	"topic://Sport/Football/Results" a "topic://Sport/Ju-Jitsu/Results"
"topic://Sport/#/ Výsledky"	Všechna témata začínající řetězcem "Sport/" a končící řetězcem "/Results"	"topic://Sport/Football/Results" a "topic://Sport/Hockey/National/Div3/Results"
" topic://Sport/ Football/#"	Všechna témata začínající řetězcem "Sport/Football/"	"topic://Sport/Football/Results" a "topic://Sport/Footballu/TeamNews/Signings/Manaserial"

WebSphere Application Server sběrnice SIBus

Sběrnice pro integraci služeb produktu WebSphere Application Server používá následující zástupné znaky:

- * aby se shodovaly se všemi znaky na jedné úrovni v hierarchii
- // odpovídá 0 nebo více úrovním
- //. aby odpovídala 0 nebo více úrovním (na konci výrazu tématu)

Tabulka 3 na stránce 29 uvádí některé příklady použití tohoto schématu zástupného znaku.

Tabulka 3. Příklady identifikátorů URI použití schématu zástupného znaku pro sběrnici pro integraci služeb produktu WebSphere Application Server

Jednotný identifikátor prostředku	Vyhovuje	Příklady
"topic://Sport/ * ball/ Výsledky"	Všechna témata s jednoduchým hierarchickým názvem končícím "míč" mezi sportem a výsledky	"topic://Sport/Football/Results" a "topic://Sport/Netball/Results"
"topic://Sport// Results"	Všechna témata začínající řetězcem "Sport/" a končící řetězcem "/Results"	"topic://Sport/Football/Results" a "topic://Sport/Hockey/National/Div3/Results"
"topic://Sport/ Football//."	Všechna témata začínající řetězcem "Sport/Football/"	"topic://Sport/Football/Results" a "topic://Sport/Footballu/TeamNews/Signings/Manaserial"
"topic://Sport/ * ball// Results//."	Témata	"topic://Sport/Football/Results" a "topic://Sport/Netball/National/Div3/Results/2002/November"

Související pojmy

Jednotné identifikátory prostředků fronty

Identifikátor URI pro frontu určuje název fronty. Může také určovat jednu či více vlastností fronty.

Dočasná místa určení

Aplikace produktu XMS mohou vytvářet a používat dočasná místa určení.

Cílové zástupné znaky

Produkt XMS poskytuje podporu pro cílové zástupné znaky a zajišťuje, aby zástupné znaky mohly být předávány do místa, kde jsou potřeba pro porovnávání. Pro každý typ serveru, se kterým může pracovat produkt XMS, existuje jiný zástupný schéma.

Související odkazy

[IDestination \(pro rozhraní .NET\)](#)

Cíl je místo, kam aplikace odesílá zprávy, nebo je to zdroj, ze kterého aplikace přijímá zprávy, nebo obojí.

[Vlastnosti místa určení](#)

Přehled vlastností objektu Destination, s odkazy na podrobnější referenční informace.

Jednotné identifikátory prostředků fronty

Identifikátor URI pro frontu určuje název fronty. Může také určovat jednu či více vlastností fronty.

Identifikátor URI pro frontu začíná frontou posloupnosti: // následovanou názvem fronty; může také obsahovat seznam dvojic název-hodnota, které nastavují zbývající vlastnosti fronty.

Pro fronty produktu IBM MQ (avšak nikoli pro výchozí fronty poskytovatele systému zpráv produktu WebSphere Application Server) může být správce front, ve kterém je fronta umístěna, zadán před frontou, přičemž název správce front je oddělen od názvu fronty.

Je-li zadán správce front, musí se jednat o ten, ke kterému je produkt XMS přímo připojen pro připojení pomocí této fronty, nebo musí být přístupný z této fronty. Vzdálení správci front jsou podporovány pouze pro načítání zpráv z front, nikoli pro vkládání zpráv do front. Podrobné informace naleznete v dokumentaci správce front produktu IBM MQ .

Není-li zadán správce front, je oddělovač navíc/je volitelný a jeho přítomnost nebo nepřítomnost nečiní žádný rozdíl v definici fronty.

Následující definice front jsou ekvivalentem pro frontu IBM MQ s názvem QB ve správci front s názvem QM_A, ke kterému je produkt XMS přímo připojen:

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

Související pojmy

[Jednotné identifikátory prostředků tématu](#)

Identifikátor URI (Uniform Resource Identifier) tématu určuje název tématu; pro něj lze také zadat jednu či více vlastností.

[Dočasná místa určení](#)

Aplikace produktu XMS mohou vytvářet a používat dočasná místa určení.

[Cílové zástupné znaky](#)

Produkt XMS poskytuje podporu pro cílové zástupné znaky a zajišťuje, aby zástupné znaky mohly být předávány do místa, kde jsou potřeba pro porovnávání. Pro každý typ serveru, se kterým může pracovat produkt XMS , existuje jiný zástupný schéma.

Související odkazy

[IDestination \(pro rozhraní .NET\)](#)

Cíl je místo, kam aplikace odesílá zprávy, nebo je to zdroj, ze kterého aplikace přijímá zprávy, nebo obojí.

[Vlastnosti místa určení](#)

Přehled vlastností objektu Destination, s odkazy na podrobnější referenční informace.

Dočasná místa určení

Aplikace produktu XMS mohou vytvářet a používat dočasná místa určení.

Aplikace obvykle používá dočasné místo určení pro příjem odpovědí na zprávy požadavků. Chcete-li určit místo určení, kam má být odeslána odpověď na zprávu požadavku, aplikace volá metodu Set JMSReplyTo objektu Message reprezentujícího zprávu požadavku. Místo určení uvedené ve výzvě může být dočasné místo určení.

Ačkoli se relace používá k vytvoření dočasného cíle, rozsah dočasného cíle je ve skutečnosti připojení, které bylo použito k vytvoření relace. Kterákoli z relací připojení mohou vytvořit generátory zpráv a spotřebitele zpráv pro dočasné místo určení. Dočasné místo určení zůstane, dokud nebude explicitně odstraněno, nebo se ukončí připojení, podle toho, co nastane dříve.

Když aplikace vytvoří dočasnou frontu, vytvoří se fronta na serveru systému zpráv, ke kterému je aplikace připojena. Je-li aplikace připojena ke správci front, vytvoří se dynamická fronta z modelové fronty, jejíž název je určen vlastností `XMSCS_WMQ_TEMPORARY_MODEL`, a předpona použitá k vytvoření názvu dynamické fronty je určena vlastností `XMSCS_WMQ_TEMP_Q_PREFIX`. Je-li aplikace připojena ke sběrnici pro integraci služeb, vytvoří se dočasná fronta v rámci sběrnice a předpona použitá k vytvoření názvu dočasné fronty je určena vlastností `XMSCS_WPM_TEMP_Q_PREFIX`.

Když aplikace, která je připojena ke sběrnici pro integraci služeb, vytvoří dočasné téma, je předpona použitá k vytvoření názvu dočasného tématu určena vlastností `XMSCS_WPM_TEMP_OPTIC_PREFIX`.

Související pojmy

Jednotné identifikátory prostředků tématu

Identifikátor URI (Uniform Resource Identifier) tématu určuje název tématu; pro něj lze také zadat jednu či více vlastností.

Jednotné identifikátory prostředků fronty

Identifikátor URI pro frontu určuje název fronty. Může také určovat jednu či více vlastností fronty.

Cílové zástupné znaky

Produkt XMS poskytuje podporu pro cílové zástupné znaky a zajišťuje, aby zástupné znaky mohly být předávány do místa, kde jsou potřeba pro porovnávání. Pro každý typ serveru, se kterým může pracovat produkt XMS, existuje jiný zástupný schéma.

Související odkazy

IDestination (pro rozhraní .NET)

Cíl je místo, kam aplikace odesílá zprávy, nebo je to zdroj, ze kterého aplikace přijímá zprávy, nebo obojí.

Vlastnosti místa určení

Přehled vlastností objektu Destination, s odkazy na podrobnější referenční informace.

Cílové zástupné znaky

Produkt XMS poskytuje podporu pro cílové zástupné znaky a zajišťuje, aby zástupné znaky mohly být předávány do místa, kde jsou potřeba pro porovnávání. Pro každý typ serveru, se kterým může pracovat produkt XMS, existuje jiný zástupný schéma.

Systémy jsou:

Typ připojení	Schéma zástupných znaků	Popis
WebSphere MQ správce front	* ? %	0 nebo více znaků 1 znak Řídící znak
Připojení v reálném čase ke zprostředkovateli	# +	Porovnat více úrovní Shoda s jednou úrovní
WebSphere Service Integration Bus	* // //.	Shoda s libovolnými znaky na jedné úrovni v hierarchii Shoda s 0 nebo více úrovněmi Shoda 0 nebo více úrovní (na konci výrazu tématu)

Související pojmy

Jednotné identifikátory prostředků tématu

Identifikátor URI (Uniform Resource Identifier) tématu určuje název tématu; pro něj lze také zadat jednu či více vlastností.

Jednotné identifikátory prostředků fronty

Identifikátor URI pro frontu určuje název fronty. Může také určovat jednu či více vlastností fronty.

Dočasná místa určení

Aplikace produktu XMS mohou vytvářet a používat dočasná místa určení.

Související odkazy

IDestination (pro rozhraní .NET)

Cíl je místo, kam aplikace odesílá zprávy, nebo je to zdroj, ze kterého aplikace přijímá zprávy, nebo obojí.

Vlastnosti místa určení

Přehled vlastností objektu Destination, s odkazy na podrobnější referenční informace.

Producenti zpráv

V produktu XMS lze producenta zpráv vytvořit buď s platným místem určení, nebo s žádným přidruženým místem určení. Při vytváření producenta zpráv s místem určení s hodnotou Null je třeba při odesílání zprávy určit platné místo určení.

Producenti zpráv bez přidruženého cíle

V produktu XMS .NET lze vytvořit producenta zpráv s místem určení s hodnotou Null.

Chcete-li vytvořit producenta zpráv bez přidruženého cíle při použití rozhraní API produktu .NET, musí být jako parametr předán parametr `CreateProducer()` objektu `ISession` (například `session.CreateProducer(null)`). Platné místo určení však musí být uvedeno, když je zpráva odeslána.

Producenti zpráv s přidruženým místem určení

V tomto scénáři je producent zpráv vytvořen s použitím platného místa určení. Během operace odeslání není třeba určit místo určení.

Spotřebitelé zpráv

Spotřebitelé zpráv lze klasifikovat jako trvalé a netrvalé odběratele a spotřebitele synchronní a asynchronní zprávy.

Trvalí odběratele

Trvalý odběratel je spotřebitel zpráv, který přijímá všechny zprávy publikované v rámci tématu, včetně zpráv publikovaných v době, kdy je odběratel neaktivní.

Informace v tomto tématu jsou relevantní pouze v případě, že se aplikace připojuje ke správci front produktu IBM MQ nebo ke sběrnici SIBus produktu WebSphere Application Server. Informace nejsou relevantní pro připojení v reálném čase ke zprostředkovateli.

Chcete-li vytvořit trvalého odběratele pro určité téma, aplikace volá metodu `Vytvořit trvalý odběratel` objektu `relace` a určuje jako parametry název, který identifikuje trvalý odběr, a cílový objekt reprezentující dané téma. Aplikace může vytvořit trvalého odběratele s selektorem zpráv nebo bez něj a může určit, zda má trvalý odběratel přijímat zprávy publikované svým vlastním připojením.

Relace použitá k vytvoření trvalého odběratele musí mít přidružený identifikátor klienta. Identifikátor klienta je stejný jako identifikátor přidružený k připojení, které se používá k vytvoření relace; je určeno způsobem popsaným v tématu [“ConnectionFactoryies a objekty připojení”](#) na stránce 21.

Název, který identifikuje trvalý odběr, musí být jedinečný v rámci identifikátoru klienta, a proto je identifikátor klienta součástí úplného, jedinečného identifikátoru trvalého odběru. Server systému zpráv uchovává záznam trvalého odběru a zajišťuje, aby všechny zprávy publikované v rámci daného tématu byly zachovány, dokud je nepotvrdí trvalý odběratel nebo dokud nevyprší jejich platnost.

Server zasílání zpráv pokračuje v údržbě záznamu trvalého odběru i po zavření trvalého odběratele. Chcete-li znovu použít trvalý odběr, který byl vytvořen dříve, musí aplikace vytvořit trvalého odběratele s uvedením stejného názvu odběru a s použitím relace se stejným identifikátorem klienta, jako jsou přidružené k trvalému odběru. Pouze jedna relace v daném okamžiku může mít trvalého odběratele pro konkrétní trvalý odběr.

Rozsah trvalého odběru je server systému zpráv, který udržuje záznam odběru. Pokud dva aplikace připojené k různým serverům systému zpráv vytvářejí trvalého odběratele s použitím stejného názvu odběru a identifikátoru klienta, vytvoří se dva zcela nezávislé trvalé odběry.

Chcete-li odstranit trvalý odběr, aplikace volá metodu zrušení odběru objektu relace a určuje jako parametr název, který identifikuje trvalý odběr. Identifikátor klienta přidružený k relaci musí být stejný jako identifikátor přidružený k trvalému odběru. Server systému zpráv odstraní záznam o trvalém odběru, který spravuje, a neodešle další zprávy trvalému odběrateli.

Chcete-li změnit existující odběr, může aplikace vytvořit trvalého odběratele s použitím stejného názvu odběru a identifikátoru klienta, avšak s určením jiného tématu nebo selektoru zpráv (nebo obou těchto voleb). Změna trvalého odběru je ekvivalentní odstranění odběru a vytvoření nového odběru.

Pro aplikaci, která se připojuje ke správci front IBM WebSphere MQ 7.0, spravuje produkt XMS fronty odběratelů. Proto není vyžadována aplikace pro zadání fronty odběratele. XMS bude ignorovat frontu odběratele, je-li zadána.

Nicméně pro aplikaci, která se připojuje ke správci front produktu IBM WebSphere MQ 6.0, musí mít každý trvalý odběratel frontu odběratele. Chcete-li zadat název fronty odběratele pro dané téma, nastavte vlastnost `XMSCS_WMQ_DUR_SUBQ` objektu Destination, která představuje téma. Výchozí fronta odběratele je `SYSTEM.JMS.D.SUBSCRIBER.QUEUE`.

Trvalý odběratel, kteří se připojují ke správci front produktu IBM WebSphere MQ 6.0, mohou sdílet jednu frontu odběratele nebo může každý trvalý odběratel načíst své zprávy ze své vlastní nevýlučné fronty odběratele. Informace o tom, jaký přístup byste měli používat pro vaši aplikaci, naleznete v tématu *XMS IBM WebSphere MQ Použití produktu Java*.

Nezapomeňte, že frontu odběratele nelze změnit pro trvalý odběr. Jediný způsob, jak změnit frontu odběratele, je odstranit odběr a vytvořit nový.

U aplikace, která se připojuje ke sběrnici pro integraci služeb, musí mít každý trvalý odběratel trvalý domovský adresář odběru. Chcete-li určit domovský adresář trvalého odběru pro všechny trvalé odběratele, kteří využívají stejné připojení, nastavte vlastnost `XMSCS_WPM_DUR_SUB_HOME` objektu ConnectionFactory, který se používá k vytvoření připojení. Chcete-li určit domovskou stránku trvalého odběru pro konkrétní téma, nastavte vlastnost `XMSCS_WPM_DUR_SUB_HOME` objektu Destination reprezentujícího dané téma. Před vytvořením trvalého odběratele, který připojení používá, musí být zadán domovský adresář odběru pro připojení. Jakákoli hodnota uvedená pro cíl přepíše hodnotu uvedenou pro připojení.

Netrvalí odběratelé

Netrvalý odběratel je spotřebitel zpráv, který přijímá pouze zprávy publikované v době, kdy je odběratel aktivní. Zprávy doručené v době, kdy je odběratel neaktivní, se ztratí.

Informace v tomto tématu jsou relevantní pouze v případě, že používáte systém zpráv publikování/odběru ve správci front produktu IBM WebSphere MQ 6.0.

Pokud se objekty spotřebitele neodstraňují před zavřením připojení nebo během jeho zavírání, mohou zprávy zůstat ve frontách zprostředkovatele pro odběratele, kteří již nejsou aktivní.

V této situaci mohou být fronty vymazány z těchto zpráv pomocí obslužného programu pro vyčištění, který je poskytnut s IBM WebSphere MQ classes for JMS Classes for JMS. Podrobnosti o způsobu použití tohoto obslužného programu jsou k dispozici v části *IBM WebSphere MQ Použití produktu Java*. Může být také třeba zvýšit hloubku fronty odběratele, pokud v této frontě existuje velký počet zpráv, které zbývají.

Synchronní spotřebitelé zpráv

Spotřebitel synchronní zprávy přijímá zprávy z fronty synchronně.

Synchronní spotřebitel zpráv přijímá jednu zprávu v daném okamžiku. Když se použije metoda `Receive(wait interval)`, volání čeká pouze zadané časové období v milisekundách pro zprávu, nebo dokud není spotřebitel zpráv uzavřen.

Je-li použita metoda `Wait() ReceiveNo()`, přijímá synchronní spotřebitel zpráv zprávy bez prodlevy; pokud je k dispozici další zpráva, je přijata okamžitě, jinak se vrátí ukazatel na objekt zprávy s hodnotou `Null`.

Spotřebitelé asynchronní zprávy

Asynchronně asynchronní spotřebitel zpráv přijímá zprávu z fronty asynchronně. Modul listener zpráv registrovaný aplikací je vyvolán vždy, když je ve frontě k dispozici nová zpráva.

Nezpracovatelné zprávy v produktu XMS

Pozastavené zprávy jsou takové, které nemohou být zpracovány přijímající aplikací MDB. Je-li zjištěna nezpracovatelná zpráva, objekt XMS MessageConsumer ji může znovu zařadit podle dvou vlastností fronty, BOQUEUE a BOTHRESH.

Za určitých okolností může být zpráva doručená do objektu MDB vrácena do fronty produktu IBM MQ . K tomu může dojít například v případě, že je doručena zpráva v rámci pracovní jednotky, která je následně odvolána. Zpráva, která je odvolána, je obvykle doručována znovu, ale špatně formátovaná zpráva může opakovaně způsobit selhání objektu typu message-driven bean, a proto ji nelze doručit. Takováto zpráva se nazývá nezpracovatelná zpráva. Produkt IBM MQ můžete nakonfigurovat tak, aby nezpracovaná zpráva byla automaticky přenesena do jiné fronty pro další zkoumání nebo byla vyřazena. Informace o tom, jak nakonfigurovat produkt IBM MQ tímto způsobem, najdete v tématu [Zpracování nezpracovatelných zpráv v aplikaci ASF](#).

Někdy je do fronty doručena špatně formátovaná zpráva. V tomto kontextu nesprávně naformátovaný znamená, že přijímající aplikace nemůže správně zpracovat zprávu. Taková zpráva může způsobit selhání přijímající aplikace a vrácení této špatně formátované zprávy. Zprávu lze poté opakovaně doručit do vstupní fronty a opakovaně ji podporovat aplikací. Tyto zprávy jsou známy jako nezpracovatelné zprávy. Objekt XMS MessageConsumer detekuje nezpracovatelné zprávy a přesměrovává je na alternativní místo určení.

Správce front produktu IBM MQ uchovává záznamy o počtu případů, kdy byla každá zpráva vrácena zpět. Když tento počet dosáhne konfigurovatelné prahové hodnoty, spotřebitel zpráv znovu odešle zprávu do pojmenované výstupní fronty. Pokud z nějakého důvodu tento požadavek selže, zpráva bude odebrána ze vstupní fronty a buď znovu zařazena do fronty nedoručených zpráv, nebo je vyřazena.

Objekty produktu XMS ConnectionConsumer zpracovávají nezpracovatelné zprávy stejným způsobem a s použitím stejných vlastností fronty. Pokud více odběratelů připojení monitoruje stejnou frontu, je možné, že nezpracovatelné zprávy mohou být doručeny aplikaci vícekrát, než je prahová hodnota před tím, než dojde k opětovnému zařazení zpráv do fronty. Toto chování je způsobeno tím, jak jednotliví spotřebitelé připojení monitorují fronty a přeřadí nezpracovatelné zprávy do fronty.

Prahová hodnota a název back-výstupní fronty jsou atributy fronty IBM MQ . Názvy atributů jsou BackoutThreshold a BackoutRequeueQName. Fronta, na kterou se vztahují, je následující:

- Pro systém zpráv typu point-to-point je to základní lokální fronta. To je důležité, pokud spotřebitelé zpráv a spotřebitelé připojení používají aliasy fronty.
- Pro systém zpráv typu publikování/odběr v normálním režimu poskytovatele systému zpráv produktu IBM MQ je to modelová fronta, z níž je vytvořena spravovaná fronta tématu.
- V případě systému zpráv typu publikování/odběr v režimu migrace poskytovatele systému zpráv produktu IBM MQ se jedná o frontu CCSUB definovanou v objektu továrny TopicConnectionnebo ve frontě CCDSUB, která je definována v objektu Topic.

Chcete-li nastavit atributy QName BackoutThreshold a BackoutRequeue, zadejte následující příkaz MQSC:

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

V případě systému zpráv publikování/odběru, pokud váš systém vytváří pro každý odběr dynamickou frontu, jsou tyto hodnoty atributů získány z tříd WebSphere MQ pro modelovou frontu JMS, SYSTEM.JMS.MODEL.QUEUE. Chcete-li tato nastavení změnit, použijte:

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

Je-li prahová hodnota vrácení nula, zpracování nezpracovatelných zpráv je zakázáno a nezpracovatelné zprávy zůstanou ve vstupní frontě. Jinak, když se počet vrácení dosáhne prahové hodnoty, zpráva se odešle do pojmenované fronty vrácení.

Pokud počet vrácení dosáhne prahové hodnoty, ale zpráva nemůže jít do fronty vyřazených zpráv, odešle se zpráva do fronty nedoručených zpráv, nebo je-li zpráva dočasná, bude vyřazena.

Tato situace nastane, pokud není definována fronta vrácení, nebo pokud objekt MessageConsumer nemůže odeslat zprávu do fronty vrácení.

Konfigurace systému pro zpracování nezpracovatelných zpráv

Fronta, kterou produkt XMS .NET používá při zjišťování atributů **BOTHRESH** a **BOQNAME**, závisí na stylu prováděného systému zpráv:

- Pro systém zpráv typu point-to-point je to základní lokální fronta. To je důležité, pokud aplikace XMS .NET spotřebovává zprávy buď z front aliasů, nebo z front klastru.
- Pro systém zpráv typu publikování/odběr se vytvoří spravovaná fronta, která bude uchovávat zprávy pro aplikaci. XMS .NET se dotazuje spravované fronty na určení hodnot pro atributy **BOTHRESH** a **BOQNAME**. Spravovaná fronta je vytvořena z modelové fronty přidružené k objektu Topic, k jehož odběru se aplikace přihlásal, a dědí hodnoty atributů **BOTHRESH** a **BOQNAME** z modelové fronty. Použitá modelová fronta závisí na tom, zda přijímající aplikace odebrala trvalý nebo netrvalý odběr:
 - Modelová fronta použitá pro trvalé odběry je určena atributem **MDURMDL** objektu Topic. Výchozí hodnota tohoto atributu je **SYSTEM.DURABLE.MODEL.QUEUE**.
 - U netrvalých odběrů je použita modelová fronta určena atributem **MNDURMDL**. Výchozí hodnota atributu **MNDURMDL** je **SYSTEM.NDURABLE.MODEL.QUEUE**.

Při zjišťování atributů **BOTHRESH** a **BOQNAME**, XMS .NET:

- Otevře lokální frontu nebo cílovou frontu pro alias frontu.
- Inquires atributy **BOTHRESH** a **BOQNAME**.
- Uzavře lokální frontu nebo cílovou frontu pro alias frontu.

Volby otevření, které se používají při otevírání lokální fronty nebo cílové fronty pro alias frontu, závisí na použité verzi produktu IBM MQ :

- Pro IBM MQ 9.0.0 Fix Pack 8a starší, pokud je lokální fronta nebo cílová fronta pro alias frontu, je fronta klastru, pak XMS .NET otevře frontu s volbami **MQOO_INPUT_AS_Q_DEF**, **MQOO_INQUIRE** a **MQOO_FAIL_IF QUIESCING**. To znamená, že uživatel, který spouští přijímající aplikaci, musí mít dotazům a získat přístup k lokální instanci fronty klastru.

Produkt XMS .NET otevře všechny ostatní typy lokální fronty s volbami otevření **MQOO_INQUIRE** a **MQOO_FAIL_IF QUIESCING**. Má-li produkt XMS .NET zadávat dotazy na hodnoty atributů, musí mít uživatel, který spouští přijímající aplikaci, přístup k dotazům na lokální frontě.

- **V 9.0.0.9** Při použití produktu XMS .NET z produktu IBM MQ 9.0.0 Fix Pack 9 musí uživatel, který spouští přijímající aplikaci, dotazovat přístup k lokální frontě bez ohledu na typ fronty.

Chcete-li přesunout nezpracovatelné zprávy do fronty vrácených zpráv do fronty nebo do fronty nedoručených zpráv správce front, musíte uživateli udělit oprávnění `put` a `passall` oprávnění aplikace.

Zpracování nezpracovatelných zpráv v ASF

Když používáte funkce ASF (Application Server Facilities), `ConnectionConsumer`, nikoli `MessageConsumer`, zpracuje nezpracovatelné zprávy. Funkce `ConnectionConsumer` znovu řadí zprávy podle vlastností `QName` fronty `BackoutThreshold` a `QName BackoutRequeue`.

Když aplikace používá `ConnectionConsumers`, okolnosti, za kterých je zpráva zálohována, závisí na relaci, kterou poskytuje aplikační server:

- Je-li relace netranksakční, s parametrem `AUTO_ACKNOWLEDGE` nebo `DUPS_OK_ACKNOWLEDGE`, je zpráva vrácena pouze po chybě systému nebo v případě neočekávaného ukončení činnosti aplikace.

- Pokud relace není součástí transakce s CLIENT_ACKNOWLEDGE, nepotvrzené zprávy mohou být vráceny aplikačním serverem voláním `Session.recover()`.

Zpravidla je to implementace klienta `MessageListener` nebo volání aplikačního serveru `Message.acknowledge()`. `Message.acknowledge()` bere na vědomí všechny zprávy doručené v relaci tak daleko.

- Když relace obsahuje transakci, mohou být nepotvrzené zprávy podporovány aplikačním serverem, který volá `Session.rollback()`.

Prohlížeče front

Aplikace používá prohlížeč front k procházení zpráv ve frontě, aniž by je odebíráte.

Při vytváření prohlížeče fronty volá aplikace metodu `Vytvořit prohlížeč front` objektu `ISession` a určuje jako parametr cílový objekt, který identifikuje frontu, která má být procházena. Aplikace může vytvořit prohlížeč front se selektorem zpráv nebo bez něj.

Po vytvoření prohlížeče fronty může aplikace volat metodu `GetEnumerator` objektu `IQueueBrowser` a získat tak seznam zpráv ve frontě. Tato metoda vrací výčtový nástroj, který zapouzdří seznam objektů zpráv. Pořadí objektů zpráv v seznamu je stejné jako pořadí, ve kterém budou zprávy načítány z fronty. Aplikace pak může použít výčtový nástroj k procházení každé zprávy postupně.

výčtový nástroj se dynamicky aktualizuje, protože zprávy jsou vloženy do fronty a odebrány z fronty. Pokaždé, když aplikace zavolá `IEnumerator.MoveNext()` k procházení další zprávy ve frontě, zpráva bude odrážet aktuální obsah fronty.

Aplikace může pro daný prohlížeč front volat metodu `GetEnumerator` více než jednou. Každé volání vrátí nový výčtový nástroj. Aplikace proto může používat více než jeden výčtový nástroj k procházení zpráv ve frontě a k udržování více pozic ve frontě.

Aplikace může použít prohlížeč front k vyhledání vhodné zprávy k odebrání z fronty a poté k odebrání zprávy použít spotřebitele zpráv se selektorem zpráv. Selektor zpráv může vybrat zprávu na základě hodnoty pole záhlaví `JMSMessageID`. Další informace o tomto a dalších polích záhlaví zprávy rozhraní JMS naleznete v tématu [“Pole záhlaví ve zprávě XMS”](#) na stránce 67.

Žadatelé

Aplikace používá žadatele k odeslání zprávy požadavku a pak k čekání na přijetí odpovědi a pro přijetí odpovědi.

Mnoho aplikací systému zpráv je založeno na algoritmech, které odesílají zprávy požadavků a poté čekají na odpověď. Produkt XMS poskytuje třídu s názvem `Requestor`, která pomáhá při vývoji tohoto stylu aplikace.

Chcete-li vytvořit žadatele, aplikace volá konstruktor `Vytvoření žadatele` třídy `Requestor`, který uvádí jako parametry objekt relace a objekt `Destination`, který označuje, kam se mají zprávy požadavku odeslat. Relace nesmí být zpracovávána, ani mít režim potvrzení `XMSCS_CLIENT_ACKNOWLEDGE`. Konstruktor automaticky vytvoří dočasnou frontu nebo téma, kam se mají odeslat zprávy odpovědi.

Po vytvoření žadatele může aplikace volat metodu požadavku objektu `Requestor` k odeslání zprávy požadavku a pak čekat na odpověď z aplikace, která přijme zprávu požadavku, a poté na něj přijmout odpověď. Volání čeká, dokud nebude obdržena odpověď, nebo dokud relace neskončí, podle toho, co nastane dříve. Pro každou zprávu požadavku je požadována pouze jedna odpověď žadatele.

V případě, že aplikace zavře žadatele, odstraní se dočasná fronta nebo téma. Přidružená relace se však nezavře.

Odstranění objektu

Když aplikace odstraní objekt XMS, který vytvořil, XMS uvolní vnitřní prostředky, které byly k objektu přiděleny.

Když aplikace vytvoří objekt XMS, XMS alokuje paměť a další vnitřní prostředky k objektu. Produkt XMS uchovává tyto interní prostředky, dokud aplikace explicitně neodstraní objekt voláním metody zavření

nebo odstranění objektu, při které XMS uvolní vnitřní prostředky. Pokud se aplikace pokusí odstranit objekt, který je již odstraněný, volání se ignoruje.

Když aplikace odstraní objekt `Connection` nebo `Session`, produkt XMS automaticky odstraní určité přidružené objekty a uvolní své vnitřní prostředky. Jedná se o objekty, které byly vytvořeny objektem `Connection` nebo `Session`, a nemají žádnou funkci nezávislou na objektu. Tyto objekty jsou zobrazeny v Tabulka 4 na stránce 37.

Poznámka: Pokud aplikace uzavře spojení se závislou relací, všechny objekty závislé na těchto relacích se také odstraní. Pouze objekt `Connection` nebo Objekt relace může mít závislé objekty.

Tabulka 4. Objekty, které jsou automaticky odstraněny		
Odstraněný objekt	Metoda	Závislé objekty, které se odstraňují automaticky
Připojení	Zavřít připojení	Data a objekty relací <code>ConnectionMeta</code>
Relace	Zavřít relaci	Objekty <code>MessageConsumer</code> , <code>MessageProducer</code> , <code>QueueBrowser</code> a <code>Requestor</code> .

Spravované transakce IBM MQ XA prostřednictvím XMS

Spravované transakce IBM MQ XA lze použít prostřednictvím rozhraní XMS.

Chcete-li používat transakce XA prostřednictvím rozhraní XMS, musí být vytvořena relace transakce. Je-li transakce XA používána, je řízení transakcí prostřednictvím globálních transakcí DTC (Distributed Transaction Coordinator) a nejedná se o relace XMS. Při použití transakcí XA nelze v relaci XMS zadat `Session.commit` nebo `Session.rollback`. Místo toho použijte k potvrzení nebo odvolávání transakcí metody `Transscope.Commit` nebo `Transscope.Rollback` DTC. Je-li relace použita pro transakci XA, musí být producent nebo odběratel vytvořený s použitím relace součástí transakce XA. Nelze je použít pro žádné operace mimo rozsah transakce XA. Nelze je použít pro operace jako `Producer.send` nebo `Consumer.receive` mimo transakci XA.

Objekt výjimky `IllegalStateException` je vyhozen, pokud

- Relace s transakcemi XA se používá pro `Session.commit` nebo `Session.rollback`.
- Objekty producenta nebo spotřebitele, které se jednou používají v relaci s transakcemi XA, jsou použity vedle rozsahu transakce XA.

Transakce XA nejsou podporovány v asynchronních odběratelích.

Poznámka:

1. Zavření může být vydáno na objektu `Producer`, `Consumer`, `Session` nebo `Connection` před potvrzením transakce XA. Ve kterých případech jsou zprávy v transakci odvolány. Podobně, je-li připojení poškozeno před potvrzením transakce XA, všechny zprávy v transakci jsou odvolány. Pro objekt `Producer` operace `rollback` znamená, že zprávy nejsou vloženy do fronty. Pro objekt `Consumer` znamená odvolání transakce, že zprávy zůstanou ve frontě.
2. Pokud objekt `Producer` vloží zprávu s hodnotou `TimeToLive` v `TransactionScope` a příkaz `commit` je vydán po uplynutí této doby, platnost zprávy může vypršet před vydáním příkazu `commit`. V takovém případě nebude zpráva zpřístupněna pro objekty produktu `Consumer`.
3. Objekty `Session` nejsou podporovány mezi podprocesy. Použití transakcí s objekty produktu `Session`, které jsou sdíleny mezi podprocesy, není podporováno.

XMS primitivní typy

XMS poskytuje ekvivalenty osmi primitivních typů Java (`byte`, `short`, `int`, `long`, `float`, `double`, `char` a `boolean`). To umožňuje výměnu zpráv mezi XMS a JMS, aniž by došlo ke ztrátě nebo poškození dat.

Tabulka 5 na stránce 38 uvádí ekvivalentní datový typ Java, velikost a minimální a maximální hodnotu každého primitivního typu XMS.

Tabulka 5. Datové typy produktu XMS a jejich ekvivalenty produktu Java

Datový typ XMS	Kompatibilní datový typ Java	Velikost	Minimální hodnota	Maximální hodnota
System.Boolean	typ boolean	32 bitů	ne	ano
System.SBYTE	bajt	8 bitů	-2^7 (-128)	2^7-1 (127)
System.BYTE	bajt	8 bitů	-2^7 (-128)	2^7-1 (127)
System.CHAR	bajt	8 bitů	-2^7 (-128)	2^7-1 (127)
System.Int16	short	16 bitů	-2^{15} (-32768)	$2^{15}-1$ (32767)
System.Int32	celé číslo	32 bitů	-2^{31} (-2147483648)	$2^{31}-1$ (2147483647)
System.Int64	long	64 bitů	-2^{63} (-9223372036858	$2^{63}-1$ (92233720368547
System.Single	float	32 bitů	-3.402823E-38 (s přesností na 7 číslic)	3.402823E+38 (až 7-číslice přesnost)
System.Double	dvojitý	64 bitů	-1.79769313486231E-308 (s přesností na 15 číslic)	1.79769313486231E+308 (až 15 číslic přesnosti)

Související pojmy

Atributy a vlastnosti objektů

Objekt XMS může mít atributy a vlastnosti, které jsou charakteristikami objektu, které jsou implementovány různými způsoby.

Implicitní převod hodnoty vlastnosti z jednoho datového typu na jiný.

Když aplikace získá hodnotu vlastnosti, může ji převést hodnota XMS na jiný datový typ. Mnoho pravidel určuje, které konverze jsou podporovány a jak produkt XMS provádí převody.

Související odkazy

Datové typy pro prvky dat aplikace

Aby se zajistilo, že aplikace XMS si může vyměňovat zprávy s aplikací IBM MQ classes for JMS, musí být obě aplikace schopny interpretovat data aplikace v těle zprávy stejným způsobem.

Implicitní převod hodnoty vlastnosti z jednoho datového typu na jiný.

Když aplikace získá hodnotu vlastnosti, může ji převést hodnota XMS na jiný datový typ. Mnoho pravidel určuje, které konverze jsou podporovány a jak produkt XMS provádí převody.

Vlastnost objektu má název a hodnotu; hodnota má přidružený datový typ, kde hodnota vlastnosti je také označována jako *typ vlastnosti*.

Aplikace používá metody třídy PropertyContext k získání a nastavení vlastností objektů. Chcete-li získat hodnotu vlastnosti, aplikace volá metodu, která je vhodná pro daný typ vlastnosti. Chcete-li například získat hodnotu celočíselné vlastnosti, aplikace obvykle volá metodu vlastnosti GetInt.

Když však aplikace získá hodnotu vlastnosti, lze ji převést pomocí produktu XMS na jiný datový typ. Chcete-li například získat hodnotu celočíselné vlastnosti, může aplikace volat metodu vlastnosti GetString, která vrací hodnotu vlastnosti ve formě řetězce. Převody podporované produktem XMS jsou zobrazeny v [Tabulka 6 na stránce 38](#).

Tabulka 6. Podporované konverze z typu vlastnosti na jiné datové typy

Typ vlastnosti	Podporované cílové datové typy
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

<i>Tabulka 6. Podporované konverze z typu vlastnosti na jiné datové typy (pokračování)</i>	
Typ vlastnosti	Podporované cílové datové typy
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
Pole System.SByte	System.String
System.Int16	System.String, System.Int32, System.Int64

Podporovaná konverze se řídí následujícími obecnými pravidly:

- Hodnoty číselných vlastností lze převádět z jednoho datového typu do jiného za předpokladu, že během převodu nebudou ztracena žádná data. Například hodnota vlastnosti s datovým typem System.Int32 může být převedena na hodnotu s datovým typem System.Int64, ale nemůže být převedena na hodnotu s datovým typem System.Int16.
- Hodnotu vlastnosti libovolného datového typu lze převést na řetězec.
- Hodnotu vlastnosti řetězce lze převést na jakýkoli jiný typ dat za předpokladu, že je řetězec správně formátován pro převod. Pokud se aplikace pokusí převést hodnotu vlastnosti řetězce, která není správně naformátována, XMS může vrátit chybu.
- Pokud se aplikace pokusí o převod, který není podporován, produkt XMS může vrátit chybu.

Když je hodnota vlastnosti převedena z jednoho datového typu do jiného, platí následující pravidla:

- Při převodu hodnoty logické vlastnosti na řetězec se hodnota true převede na řetězec "true" a hodnota false se převede na řetězec "false".
- Při převodu hodnoty logické vlastnosti na číselný datový typ, včetně System.SByte, hodnota true se převede na 1 a hodnota false je převedena na 0.
- Při převodu hodnoty vlastnosti řetězce na logickou hodnotu bude řetězec "true" (nerozlišovat velká a malá písmena) nebo "1" převeden na hodnotu true a řetězec "false" (bez rozlišování velkých a malých písmen) nebo "0" bude převeden na hodnotu false. Všechny ostatní řetězce nelze konvertovat.
- Při převodu hodnoty vlastnosti řetězce na hodnotu s datovým typem System.Int32, System.Int64, System.SByte nebo System.Int16 musí mít řetězec následující formát:

[mezery] [znak]číslice

Řetězcové komponenty jsou definovány takto:

mezery

Volitelné úvodní prázdné znaky.

SIGN

Volitelné znaménko plus (+) nebo znak minus (-).

číslice

Souvislá posloupnost číselných znaků (0-9). Musí být přítomen alespoň jeden číselný znak.

Po pořadí znaků číslic může řetězec obsahovat i jiné znaky, které nejsou číslice, ale konverze se zastaví, jakmile je dosaženo začátku těchto znaků. Předpokládá se, že řetězec představuje desítkové celé číslo.

XMS může vrátit chybu, pokud není řetězec správně naformátován.

- Při převodu hodnoty vlastnosti řetězce na hodnotu s datovým typem System.Double nebo System.Float musí mít řetězec následující formát:

[mezery] [znak] [číslice] [point[d_cifer]] [e_char[_znak]_číslice]

Řetězcové komponenty jsou definovány takto:

mezery

(Volitelné) Úvodní prázdné znaky.

SIGN

(Volitelné) Znaménko plus (+) nebo znak minus (-).

číslice

Souvislá posloupnost číselných znaků (0-9). Alespoň jeden znak číslice musí být přítomen buď v *číslicích* , nebo v *d_digits*.

bod

(Nepovinné) Desetinný oddělovač (.

číslice

Souvislá posloupnost číselných znaků (0-9). Alespoň jeden znak číslice musí být přítomen buď v *číslicích* , nebo v *d_digits*.

_char

Znak exponent, který je buď *E* , nebo *e*.

_sign

(Volitelné) Znaménko plus (+) nebo minus (-) pro exponent.

_číslice

Souvislá posloupnost znaků číslic (0-9) pro exponent. Pokud řetězec obsahuje exponent, musí být přítomen alespoň jeden znak číslice.

Po pořadí znaků číslic nebo volitelných znaků představujících exponent může řetězec obsahovat jiné znaky, které nejsou číslice, ale konverze se zastaví, jakmile se dosáhne první z těchto znaků. Předpokládá se, že řetězec představuje desetinné číslo s plovoucí řádovou čárkou s exponentem, který je mocninou 10.

XMS může vrátit chybu, pokud není řetězec správně naformátován.

- Při převodu číselné hodnoty vlastnosti na řetězec, včetně hodnoty vlastnosti s datovým typem System.SByte, se hodnota převede na řetězcovou reprezentaci hodnoty jako dekadické číslo, nikoli řetězec obsahující znak ASCII pro tuto hodnotu. Například celé číslo 65 se převede na řetězec "65", nikoli na řetězec "A".
- Při převodu hodnoty vlastnosti bajtového pole na řetězec se každý bajt převede na 2 hexadecimální znaky, které představují bajt. Příklad: Bajtové pole {0xF1, 0x12, 0x00, 0xFF} je převedeno na řetězec "F11200FF".

Převody z typu vlastnosti na jiné datové typy jsou podporovány metodami jak vlastností Property, tak i třídy PropertyContext .

Související pojmy

Atributy a vlastnosti objektů

Objekt XMS může mít atributy a vlastnosti, které jsou charakteristikami objektu, které jsou implementovány různými způsoby.

XMS primitivní typy

XMS poskytuje ekvivalenty osmi primitivních typů Java (byte, short, int, long, float, double, char a boolean). To umožňuje výměnu zpráv mezi XMS a JMS , aniž by došlo ke ztrátě nebo poškození dat.

Související odkazy

Mapové zprávy

Tělo zprávy mapy obsahuje sadu dvojic název-hodnota, kde má každá hodnota přidružený datový typ.

Zprávy proudů

Tělo zprávy s proudem hodnot obsahuje proud hodnot, kde má každá hodnota přidružený datový typ.

Datové typy pro prvky dat aplikace

Aby se zajistilo, že aplikace XMS si může vyměňovat zprávy s aplikací IBM MQ classes for JMS , musí být obě aplikace schopny interpretovat data aplikace v těle zprávy stejným způsobem.

iterátory

Iterátor zapouzdří seznam objektů a kurzor, který udržuje aktuální pozici v seznamu. Koncept iterátoru, který je k dispozici v produktu IBM Message Service Client for C/C++ , je implementován s použitím rozhraní IEnumerator v produktu IBM Message Service Client for .NET.

Když se vytvoří iterátor, pozice kurzoru je před prvním objektem. Aplikace používá iterátor k načtení každého objektu postupně.

Třída iterátoru IBM Message Service Client for C/C++ je ekvivalentem třídy Enumerator v Java. XMS .NET je podobný jako Java a používá rozhraní IEnumerator.

Aplikace může použít IEnumerator k provedení následujících úloh:

- Chcete-li získat vlastnosti zprávy
- Chcete-li získat dvojici název-hodnota v textu zprávy mapy,
- Procházení zpráv ve frontě
- Chcete-li získat názvy vlastností JMS definovaných zpráv podporovaných připojením,

Identifikátory kódované znakové sady

V produktu XMS .NET jsou všechny řetězce předány s použitím nativního řetězce .NET . Vzhledem k tomu, že má pevné kódování, není třeba k jeho interpretaci žádné další informace. Proto není vlastnost XMSCS_CLIENT_CCSDID vyžadována pro aplikace XMS .NET .

Chybové kódy a kódy výjimek produktu XMS

XMS používá rozsah kódů chyb k označení selhání. Tyto chybové kódy nejsou v této dokumentaci výslovně uvedeny, protože se mohou lišit od vydání k vydání. Pouze XMS kódy výjimek (ve formátu XMS_X_ ...) jsou dokumentovány, protože zůstávají stejné ve všech verzích produktu XMS.

Vytváření vlastních aplikací

Sestavujete své vlastní aplikace, jako např. sestavení ukázkových aplikací.

Sestavte aplikaci produktu .NET , jak je popsáno v tématu [“Sestavení ukázkových aplikací produktu .NET” na stránce 19](#), které také uvádí nezbytné předpoklady pro sestavení vašich vlastních aplikací produktu .NET . Další informace o tom, jak sestavit vlastní aplikace, použijte soubory Makefile poskytnuté pro každou ukázkovou aplikaci.

Tip: Chcete-li pomoci s diagnózou problému v případě selhání, můžete zjistit, že je užitečné kompilovat aplikace se symboly, které jsou zahrnuty.

Související pojmy

Ukázkové aplikace

Ukázkové aplikace poskytují přehled běžných funkcí jednotlivých rozhraní API. Můžete je použít k ověření instalace a nastavení serveru systému zpráv a k usnadnění sestavení vašich vlastních aplikací.

Související odkazy

[.NET Rozhraní](#)

Tato sekce dokumentuje rozhraní třídy .NET a jejich vlastnosti a metody.

[Vlastnosti objektů XMS](#)

Tento kapitola dokumentuje vlastnosti objektu definované parametrem XMS.

Automatické opětovné připojení klienta IBM MQ prostřednictvím XMS

Konfigurujte klienta produktu XMS tak, aby se automaticky znovu připojil po síti, správci front nebo selhání serveru při použití klienta produktu IBM WebSphere MQ 7.1 a pozdějších, jako poskytovatele zpráv.

Pomocí vlastností WMQ_CONNECTION_NAME_LIST a WMQ_CLIENT_RECONNECT_OPTIONS třídy MQConnectionFactory lze konfigurovat připojení klienta pro automatické nové připojení. Automatické

opětovné připojení klienta znovu připojí klienta po selhání připojení nebo jako volba po zastavení správce front. Návrh některých klientských aplikací je z nich činí nevhodným pro automatické opětovné připojení.

Automaticky znovu připojitelná připojení klienta se stanou znovu připojitelné, jakmile se ustanovené spojení.

Poznámka: Vlastnosti Volby opětovného připojení klienta, časový limit opětovného připojení klienta a Seznam názvů připojení lze také nastavit prostřednictvím tabulky CCDT (Client Channel Definitions Table) nebo povolením opětovného připojení klienta prostřednictvím souboru mqclient.ini.

Poznámka: Pokud jsou v objektu ConnectionFactory a v tabulce CCDT nastaveny vlastnosti opětovného připojení, bude mít pravidlo přednosti následující pravidla. Je-li výchozí hodnota vlastnosti seznamu názvů připojení nastavena v objektu ConnectionFactory, má přednost tabulka CCDT. Není-li seznam názvů připojení nastaven na výchozí hodnotu, mají přednost hodnoty vlastností nastavené v objektu ConnectionFactory. Výchozí hodnota seznamu názvů připojení je localhost(1414).

Zápis aplikací produktu XMS .NET

Témata v této sekci obsahují informace, které vám pomohou při psaní aplikací produktu XMS .NET .

Tato sekce poskytuje informace, které jsou specifické pro zápis aplikací XMS .NET . Obecné informace o zápisu aplikací XMS naleznete v tématu [“Zápis aplikací XMS”](#) na stránce 20.

Tento oddíl obsahuje následující témata:

- [“Datové typy pro .NET”](#) na stránce 42
- [“Spravované a nespravované operace v produktu .NET”](#) na stránce 43
- [“Cíle v .NET”](#) na stránce 44
- [“Vlastnosti v produktu .NET”](#) na stránce 44
- [“Neexistující zpracování vlastností v produktu .NET”](#) na stránce 45
- [“Ošetření chyb v produktu .NET”](#) na stránce 46
- [“Listenery zpráv a výjimek v produktu .NET”](#) na stránce 46

Související pojmy

[Zápis aplikací XMS](#)

Témata v této sekci obsahují informace, které vám pomohou při psaní aplikací produktu XMS .

Související odkazy

[.NETRozhraní](#)

Tato sekce dokumentuje rozhraní třídy .NET a jejich vlastnosti a metody.

Datové typy pro .NET

Produkt XMS .NET podporuje soubory 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, a System.Object. Datové typy pro XMS .NET se liší od datových typů pro XMS C/C++. Toto téma můžete použít k identifikaci odpovídajících datových typů.

Následující tabulka zobrazuje odpovídající datové typy XMS .NET a XMS C/C++ a stručně je popisuje.

Tabulka 7. Datové typy pro XMS .NET a XMS C/C++		
Typ XMS .NET	XMS Typ C/C++	Popis
System.SByte	xmsSBYTE xmsINT8	Podepsaná 8bitová hodnota

Tabulka 7. Datové typy pro XMS .NET a XMS C/C++ (pokračování)

Typ XMS .NET	XMS Typ C/C++	Popis
System.Byte	xmsBYTE xmsUINT8	Nepodepsaná 8bitová hodnota
System.Int16	xmsINT16 xmsSHORT	Byla podepsána 16bitová hodnota
System.UInt16	xmsUINT16 xmsUSHORT	Nepodepsaná 16bitová hodnota
System.Int32	xmsINT32 xmsINT	Hodnota 32bitového systému
System.UInt32	xmsUINT32 xmsUINT	Nepodepsaná 32bitová hodnota
System.Int64	xmsLONG xmsINT64	Podepsaná 64bitová hodnota
System.UInt64	xmsULONG xmsUINT64	Nepodepsaná 64bitová hodnota
System.Char	xmsCHAR16	Nepodepsaný 16bitový znak (Unicode pro .NET)
System.Single	xmsFLOAT	32bitový systém IEEE
System.Double	xmsDOUBLE	64bitový standard IEEE
System.Boolean	xmsBOOL	Hodnota True/False
Nelze použít	xmsCHAR	Podepsaná nebo nepodepsaná 8bitová hodnota (podepsaná nebo nepodepsaná závisí na platformě)
System.Decimal	Nelze použít	96 bitů s celočíselným násobkem 10^0 až 10^{28}
System.Object	Nelze použít	Základ všech typů
System.String	Nelze použít	Řetězcový typ

Spravované a nespravované operace v produktu .NET

Spravovaný kód se provádí výhradně v rámci běhového prostředí obecného jazyka .NET a je zcela závislý na službách poskytovaných tímto běhovým prostředím. Aplikace je klasifikována jako nespravovaná, pokud libovolná část aplikace běží nebo volá služby mimo prostředí JRE (common language runtime) produktu .NET .

Některé rozšířené funkce v současné době v rámci spravovaného prostředí .NET nemohou být podporovány.

Pokud vaše aplikace vyžaduje některé funkce, které nejsou aktuálně podporovány v plně spravovaném prostředí, můžete aplikaci změnit tak, aby používala nespravované prostředí, aniž byste potřebovali podstatnou změnu ve vaší aplikaci. Při výběru této volby je však třeba poznamenat, že sada XMS používá při výběru tohoto výběru nespravovaný kód.

Připojení ke správci front produktu IBM MQ

Spravovaná připojení k WMQ_CM_CLIENT nebudou podporovat komunikace bez protokolu TCP a kompresi kanálu. Tato připojení však mohou být podporována pomocí nespravovaného připojení (WMQ_CM_CLIENT_UNMANAGED). Další informace viz [Vyvíjení aplikací .NET](#).

Pokud vytváříte továrnu na připojení ze spravovaného objektu v nespravovaném prostředí, musíte ručně změnit hodnotu pro režim připojení na hodnotu XMSCS_WMQ_CM_CLIENT_UNMANAGED.

Připojení ke stroji systému zpráv sběrnice pro integraci služeb produktu WebSphere Application Server

Připojení ke stroji systému zpráv sběrnice pro integraci služeb produktu WebSphere Application Server , který vyžaduje použití protokolu SSL (včetně HTTPS), nejsou v současné době podporovány jako spravovaný kód.

Související odkazy

[XMSC_WMQ_CONNECTION_MODE](#)

Cíle v .NET

V produktu .NET se cíle vytvářejí podle typu protokolu a lze je použít pouze pro typ protokolu, pro který jsou vytvořeny.

Jsou poskytnuty dvě funkce pro vytváření cílů, jedno pro témata a jedna pro fronty:

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

Tyto funkce jsou k dispozici na následujících dvou objektech v rozhraní API:

- `ISession`
- `XMSFactoryFactory`

V obou případech mohou tyto metody přijmout řetězec stylu identifikátoru URI, který může obsahovat parametry, v následujícím formátu:

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

Alternativně mohou tyto metody přijmout pouze cílový název, tj. název bez tématu: `//` nebo `queue://` předpona a bez parametrů.

To znamená, že následující řetězec stylu identifikátoru URI:

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

bude mít stejný výsledek jako následující název místa určení:

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

Co se týče WebSphere Application Server sběrnice integrace služeb JMS, témata mohou být také uvedena v zkrácené formě, která zahrnuje jak *topicname* , tak i *topicspace* , ale nemohou obsahovat parametry:

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

Vlastnosti v produktu .NET

Aplikace .NET používá metody v rozhraní `PropertyContext` k získání a nastavení vlastností objektů.

Rozhraní `PropertyContext` zapouzdřuje metody, které získají a nastavují vlastnosti. Tyto metody jsou zděděny, přímo nebo nepřímo, následujícími třídami:

- [BytesMessage](#)
- [Připojení](#)
- [ConnectionFactory](#)
- [ConnectionMetaData](#)
- [Místo určení](#)
- [MapMessage](#)
- [Zpráva](#)
- [MessageConsumer](#)
- [MessageProducer](#)
- [ObjectMessage](#)
- [QueueBrowser](#)
- [Relace](#)
- [StreamMessage](#)
- [TextMessage](#)

Pokud aplikace nastavuje hodnotu vlastnosti, nová hodnota nahradí všechny předchozí hodnoty vlastnosti.

Další informace o vlastnostech produktu XMS naleznete v tématu [“Vlastnosti objektů XMS”](#) na stránce 173.

Pro snadné použití jsou názvy vlastností a hodnoty produktu XMS v produktu XMS předdefinované jako veřejné konstanty ve struktuře s názvem `XMSC`. Názvy těchto konstant se nacházejí ve formátu `XMSC.konstanta`; například `XMSC.USERID` (konstanta názvu vlastnosti) a `XMSC.DELIVERY_AS_APP` (konstantní hodnota).

Kromě toho můžete přistupovat k konstantám IBM MQ pomocí `IBM.XMS.MQC`. Pokud je `IBM.XMS` již naimportován, můžete přistupovat k hodnotám pro tyto vlastnosti ve tvaru `MQC.konstanta`. Například `MQC.MQRO_COA_WITH_FULL_DATA`.

Dále platí, že pokud máte hybridní aplikaci, která používá jak třídy XMS .NET, tak třídy IBM MQ pro .NET a že importuje oba `IBM.XMS` a `IBM.WMQ`, poté je třeba plně kvalifikovat obor názvů struktury `MQC`, abyste se ujistili, že každý výskyt je jedinečný.

Některé rozšířené funkce nejsou v současné době ve spravovaném prostředí .NET podporovány. Další podrobnosti lze najít v tématu [“Spravované a nespravované operace v produktu .NET”](#) na stránce 43.

Neexistující zpracování vlastností v produktu .NET

Zpracování neexistujících vlastností v XMS .NET je v podstatě konzistentní se specifikací JMS a také s implementacemi C a C++ produktu XMS.

V případě služby JMS může při přístupu k neexistující vlastnosti dojít k výjimce systému Java, když se metoda pokusí převést neexistující (null) hodnotu na požadovaný typ. Pokud vlastnost neexistuje, dojde k následujícím výjimkám:

- Vlastnost `getString` vlastnost `getObject` vrací hodnotu null
- `getBooleanVlastnost` vrací false, protože hodnota `Boolean.valueOf(null)` vrátí hodnotu false
- `getIntProperty` etc. throw `java.lang.NumberFormatException`, protože `Integer.valueOf(null)` vyvolá výjimku

Pokud vlastnost v produktu XMS .NET neexistuje, dojde k následujícím výjimkám:

- `GetStringVlastnost` a `GetObjectProperty` (a `GetBytesProperty`) vracejí hodnotu null (což je stejné jako Java)

- `GetBooleanVlastnost` throws `System.NullReferenceException`
- Vlastnost `GetIntatd` hodí `System.NullReferenceException`

Tato implementace se liší od Java, ale je v podstatě konzistentní se specifikací JMS a s rozhraními XMS C a C + +. Podobně jako implementace produktu Java šíří produkt XMS .NET všechny výjimky z volání `System.Convert` volajícím. Na rozdíl od produktu Java však XMS explicitně vyhodí výjimky `NullReference` než pouze použití nativního chování rámce produktu .NET přes předání hodnoty null do rutin pro převod systému. Pokud vaše aplikace nastaví vlastnost na String like "abc" a vyvolá `GetIntProperty`, `System.FormatException` vyvolá `Convert.ToInt32("abc")` je šířen na volající modul, který je konzistentní s Java. Výjimka `MessageFormat` je generována pouze v případě, že typy použité pro vlastnosti `setProperty` a `getProperty` jsou nekompatibilní. Toto chování je také konzistentní s Java.

Ošetření chyb v produktu .NET

Výjimky produktu XMS .NET jsou odvozeny od výjimky `System.Exception`. Volání metody XMS mohou vyvolat specifické výjimky produktu XMS, jako je výjimka `MessageFormat`, obecná výjimka `XMSEExceptions` nebo výjimky systému, jako je výjimka `NullReference`.

Zapište aplikace, abyste zachytili jakoukoli z těchto chyb, buď ve specifických blocích `catch`, nebo v obecných blocích `catch System.Exception`, podle požadavků na požadavky aplikací.

Listenery zpráv a výjimek v produktu .NET

Aplikace produktu .NET používá modul `listener` pro zprávy k asynchronnímu příjmu zpráv a modul `listener` pro výjimky používá k asynchronnímu oznámení problému s připojením.

Funkčnost obou `listenerů` zpráv i výjimek je stejná pro produkt .NET a pro jazyk C + +. Existují však i malé rozdíly v implementaci.

Listenery zpráv v produktu .NET

Chcete-li asynchronně přijímat zprávy, je třeba provést následující kroky:

1. Definujte metodu, která odpovídá podpisu delegáta posluchače zpráv. Vámi definovaná metoda může být statická nebo instanční metoda a může být definována v libovolné přístupné třídě. Podpis delegáta je následující:

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

a tak byste mohli definovat metodu jako:

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

2. Vytvořte instanci této metody jako delegáta pomocí něčeho podobného následujícímu:

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

3. Zaregistrujte delegáta s jedním nebo více spotřebiteli tím, že ji nastavíte na vlastnost `MessageListener` spotřebitele:

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

Delegovanou osobu můžete odebrat nastavením hodnoty `MessageListener` zpět na hodnotu `null`:

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

Moduly listener výjimek v prostředí .NET

Modul listener pro výjimky pracuje stejně jako modul listener pro zprávy, ale má jinou definici delegáta a je přiřazen k připojení, spíše než spotřebitel zpráv. To je stejné jako pro C + +.

1. Definujte metodu. Podpis delegáta je následující:

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

a tak může být definovaná metoda:

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

2. Vytvořit instanci této metody jako delegáta pomocí něčeho podobného:

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

3. Zaregistrujte delegáta s připojením tak, že nastavíte jeho vlastnost ExceptionListener :

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

Delegát můžete odebrat resetováním modulu ExceptionListener na:

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

Pokud žádné odkazy na ně nezůstanou, výjimky nebo zprávy se automaticky odstraní pomocí programu pro uvolnění paměti systému.

Následuje ukázka kódu:

```
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);
}
}

```

Práce se spravovanými objekty

Témata v této sekci obsahují informace o spravovaných objektech. Aplikace produktu XMS mohou načítat definice objektů z úložiště centrálního administrovaných objektů a používat je k vytváření továren připojení a míst určení.

Tato sekce poskytuje informace, které vám pomohou s vytvářením a správou spravovaných objektů s popisem typů spravovaných úložišť objektů, které produkt XMS podporuje. Tento oddíl také vysvětluje, jak aplikace XMS vytvoří připojení k úložišti administrovaných objektů, aby načetl požadované administrované objekty.

Tento oddíl obsahuje následující témata:

- [“Podporované typy úložiště spravovaných objektů” na stránce 48](#)
- [“Mapování vlastností pro spravované objekty” na stránce 49](#)
- [“Povinné vlastnosti pro spravované objekty ConnectionFactory” na stránce 52](#)
- [“Povinné vlastnosti pro spravované objekty typu Destination” na stránce 54](#)
- [“Vytvoření spravovaných objektů” na stránce 55](#)
- [“Objekty InitialContext” na stránce 57](#)
- [“Vlastnosti InitialContext” na stránce 58](#)
- [“Formát identifikátoru URI pro počáteční kontexty produktu XMS” na stránce 59](#)
- [“Webová služba vyhledávání JNDI” na stránce 61](#)
- [“Načítání spravovaných objektů” na stránce 62](#)

Související pojmy

Spravované objekty

Pomocí spravovaných objektů můžete spravovat nastavení připojení používaná klientskými aplikacemi, které mají být spravovány z centrálního úložiště. Aplikace načítá definice objektů z centrálního úložiště a používá je k vytváření objektů `ConnectionFactory` a `Destination`. Pomocí spravovaných objektů můžete odstranit několik aplikací z prostředků, které používají za běhu.

Podporované typy úložiště spravovaných objektů

Objekty systému souborů a LDAP spravované objekty lze použít pro připojení k produktu IBM MQ a WebSphere Application Server, zatímco pojmenování COS lze použít k připojení pouze k WebSphere Application Server.

Adresáře objektů systému souborů mají formu serializovaných objektů Java Naming Directory Interface (JNDI). Adresáře objektů LDAP jsou adresáře, které obsahují objekty JNDI. Adresáře systému souborů a objektu LDAP lze spravovat buď pomocí nástroje JMSAdmin, který je poskytován s produktem IBM WebSphere MQ 6.0, nebo pomocí nástroje IBM MQ Explorer, který je dodáván s produktem IBM WebSphere MQ 7.0 a novějším. Jak systém souborů, tak adresáře objektu LDAP mohou být použity ke správě připojení klientů centralizací továren a cílů připojení produktu IBM WebSphere MQ. Administrátor sítě může implementovat více aplikací, které odkazují na stejné centrální úložiště, a které jsou automaticky aktualizovány, aby odrážely změny nastavení připojení provedených v centrálním úložišti.

Adresář pojmenování COS obsahuje továrny připojení a cíle produktu WebSphere Application Server service integration bus a lze jej spravovat pomocí administrativní konzoly serveru WebSphere Application

Server . Aby aplikace XMS načítala objekty z adresáře pojmenování COS, musí být nasazena webová služba vyhledávání JNDI . Tato webová služba není k dispozici na všech WebSphere Application Server service integration technologies. Podrobnosti naleznete v dokumentaci produktu.

Poznámka: Restartujte aplikační připojení, aby se změny v adresáři objektů projevily.

Související pojmy

Mapování vlastností pro spravované objekty

Chcete-li aplikacím povolit použití továrny připojení a definic cílových objektů produktu IBM MQ JMS a WebSphere Application Server , musí být vlastnosti načtené z těchto definic mapovány na odpovídající vlastnosti produktu XMS , které lze nastavit v továrnách a cílech připojení XMS .

Vlastnosti InitialContext

Parametry konstrukturu InitialContext zahrnují umístění úložiště spravovaných objektů, které jsou poskytnuty jako identifikátor URI (Uniform Resource Indicator). Aby mohla aplikace vytvořit připojení k úložišti, může být nezbytné poskytnout více informací než informace obsažené v identifikátoru URI.

Formát identifikátoru URI pro počáteční kontexty produktu XMS

Umístění úložiště spravovaných objektů je poskytováno jako jednotný indikátor prostředku (URI). Formát identifikátoru URI závisí na typu kontextu.

Webová služba vyhledávání JNDI

Chcete-li přistupovat k adresáři pojmenování COS z produktu XMS, musí být na serveru WebSphere Application Server service integration bus implementována webová služba vyhledávání produktu JNDI . Tato webová služba přeloží informace o produktu Java ze služby názvů COS do formuláře, který lze číst v aplikacích produktu XMS .

Načítání spravovaných objektů

Produkt XMS načítá administrovaný objekt z úložiště s použitím adresy zadané při vytvoření objektu InitialContext nebo ve vlastnostech InitialContext .

Spravované objekty

Pomocí spravovaných objektů můžete spravovat nastavení připojení používaná klientskými aplikacemi, které mají být spravovány z centrálního úložiště. Aplikace načítá definice objektů z centrálního úložiště a používá je k vytváření objektů ConnectionFactory a Destination . Pomocí spravovaných objektů můžete odstranit několik aplikací z prostředků, které používají za běhu.

Související úlohy

Vytvoření spravovaných objektů

Definice objektu ConnectionFactory a Cílový objekt, které aplikace produktu XMS vyžadují k vytvoření připojení k serveru systému zpráv, musí být vytvořeny pomocí příslušných nástrojů pro správu.

Objekty InitialContext

Aplikace musí vytvořit počáteční kontext, který má být použit k vytvoření připojení k úložišti spravovaných objektů, aby bylo možné načíst požadované spravované objekty.

Související odkazy

Povinné vlastnosti pro spravované objekty ConnectionFactory

Když aplikace vytvoří továrnu připojení, musí být definován počet vlastností pro vytvoření připojení k serveru systému zpráv.

Povinné vlastnosti pro spravované objekty typu Destination

Aplikace, která vytváří cíl, musí nastavit několik vlastností, které aplikace na administrovaného cílovém objektu.

Mapování vlastností pro spravované objekty

Chcete-li aplikacím povolit použití továrny připojení a definic cílových objektů produktu IBM MQ JMS a WebSphere Application Server , musí být vlastnosti načtené z těchto definic mapovány na odpovídající vlastnosti produktu XMS , které lze nastavit v továrnách a cílech připojení XMS .

Chcete-li například vytvořit továrnu připojení produktu XMS s vlastnostmi načtenými z továrny na připojení IBM MQ JMS, musí být vlastnosti mapovány mezi těmito dvěma objekty.

Všechna mapování vlastností se provádějí automaticky.

Následující tabulka ukazuje mapování mezi některými z nejběžnějších vlastností továren připojení a míst určení. Vlastnosti zobrazené v této tabulce jsou pouze malou sadou příkladů, a ne všechny zobrazené vlastnosti jsou relevantní pro všechny typy připojení a servery.

<i>Tabulka 8. Příklady mapování názvu pro továrnu připojení a vlastnosti místa určení</i>		
Název vlastnosti produktu IBM MQ JMS	XMS Název vlastnosti	WebSphere Application Server service integration bus Název vlastnosti
PERZISTENCE (ZA)	<u>REŽIM XMSCS_DELIVERY_MODE</u>	
VYPRŠENÍ (EXP)	<u>XMSCS_TIME_TO_LIVE</u>	
PRIORITA (PRI)	<u>PRIORITA XMSCSC</u>	
	<u>NÁZEV_HOSTITEL_XMSC</u>	serverName
	<u>XMSCS_WPM_BUS_NAME</u>	busName
	<u>XMSCS_WPM_TOPIC_SPACE</u>	topicName

<i>Tabulka 9. Vlastnosti produktu XMS .NET</i>					
Vlastnost	Typ objektu				
	CF	QCF	TCF	Fronta	Téma
<u>APPLICATIONNAME</u>	Y	Y	Y	Není k dispozici	Není k dispozici
<u>ASYNCEXCEPTION</u>	Y	Y	Y	Není k dispozici	Není k dispozici
<u>CCDTURL</u>	Y	Y	Y	Není k dispozici	Není k dispozici
<u>CHANNEL</u>	Y	Y	Y	Není k dispozici	Není k dispozici
<u>ConnectionNameList</u>	Y	Y	Y	Není k dispozici	Není k dispozici
<u>CLIENTRECONNECTOPTIONS</u>	Y	Y	Y	Není k dispozici	Není k dispozici
<u>CLIENTRECONNECTTIMEOUT</u>	Y	Y	Y	Není k dispozici	Není k dispozici
<u>CLIENTID</u>	Není k dispozici	Y	Není k dispozici	Není k dispozici	Není k dispozici
<u>COMPHDR</u> "1" na stránce 51	Y	Není k dispozici	Y	Není k dispozici	Není k dispozici
<u>COMPMSG</u> "1" na stránce 51	Y	Y	Y	Není k dispozici	Není k dispozici
<u>CONNOPT</u> "1" na stránce 51	Y	Y	Y	Není k dispozici	Není k dispozici
<u>CONNTAG</u> "1" na stránce 51	Y	Y	Y	Není k dispozici	Není k dispozici
<u>Popis</u> "1" na stránce 51	Není k dispozici	Y	Není k dispozici	Y	Y
<u>EXPIRY</u> "1" na stránce 51	Není k dispozici	Není k dispozici	Není k dispozici	Y	Y

Tabulka 9. Vlastnosti produktu XMS .NET (pokračování)

Vlastnost	Typ objektu				
	CF	QCF	TCF	Fronta	Téma
<u>FAILIFQUIESCE</u>	Y	Y	Y	Y	Y
<u>HOSTNAME</u>	Není k dispozici	Y	Není k dispozici	Není k dispozici	Není k dispozici
<u>LOCALADDRESS</u>	Není k dispozici	Y	Není k dispozici	Není k dispozici	Není k dispozici
<u>PERSISTENCE</u>	Není k dispozici	Není k dispozici	Není k dispozici	Y	Y
<u>PORT</u>	Není k dispozici	Y	Není k dispozici	Není k dispozici	Není k dispozici
<u>Priorita "1" na stránce 51</u>	Není k dispozici	Není k dispozici	Není k dispozici	Y	Y
<u>PROVIDERVISION "1" na stránce 51</u>	Není k dispozici	Y	Není k dispozici	Není k dispozici	Není k dispozici
<u>QMANAGER</u>	Y	Y	Y	Y	Není k dispozici
<u>QUEUE "1" na stránce 51</u>	Není k dispozici	Není k dispozici	Není k dispozici	Y	Není k dispozici
<u>SHARECONVALLOWED</u>	Y	Y	Y	Není k dispozici	Není k dispozici
<u>Téma "1" na stránce 51</u>	Není k dispozici	Není k dispozici	Není k dispozici	Není k dispozici	Y
<u>Přenos "1" na stránce 51</u>	Není k dispozici	Y	Není k dispozici	Není k dispozici	Není k dispozici

Poznámka:

1. Tyto vlastnosti nemají vlastnosti na úrovni aplikace, ale mohou být volitelně nastaveny pomocí spravovaných vlastností.

Související pojmy

Podporované typy úložiště spravovaných objektů

Objekty systému souborů a LDAP spravované objekty lze použít pro připojení k produktu IBM MQ a WebSphere Application Server, zatímco pojmenování COS lze použít k připojení pouze k WebSphere Application Server.

Vlastnosti InitialContext

Parametry konstruktoru InitialContext zahrnují umístění úložiště spravovaných objektů, které jsou poskytnuty jako identifikátor URI (Uniform Resource Indicator). Aby mohla aplikace vytvořit připojení k úložišti, může být nezbytné poskytnout více informací než informace obsažené v identifikátoru URI.

Formát identifikátoru URI pro počáteční kontexty produktu XMS

Umístění úložiště spravovaných objektů je poskytováno jako jednotný indikátor prostředku (URI). Formát identifikátoru URI závisí na typu kontextu.

Webová služba vyhledávání JNDI

Chcete-li přistupovat k adresáři pojmenování COS z produktu XMS, musí být na serveru WebSphere Application Server service integration bus implementována webová služba vyhledávání produktu JNDI . Tato webová služba přeloží informace o produktu Java ze služby názvů COS do formuláře, který lze číst v aplikacích produktu XMS .

Načítání spravovaných objektů

Produkt XMS načítá administrovaný objekt z úložiště s použitím adresy zadané při vytvoření objektu InitialContext nebo ve vlastnostech InitialContext .

Související úlohy

Vytvoření spravovaných objektů

Definice objektu ConnectionFactory a Cílový objekt, které aplikace produktu XMS vyžadují k vytvoření připojení k serveru systému zpráv, musí být vytvořeny pomocí příslušných nástrojů pro správu.

Objekty InitialContext

Aplikace musí vytvořit počáteční kontext, který má být použit k vytvoření připojení k úložišti spravovaných objektů, aby bylo možné načíst požadované spravované objekty.

Související odkazy

Povinné vlastnosti pro spravované objekty ConnectionFactory

Když aplikace vytvoří továrnu připojení, musí být definován počet vlastností pro vytvoření připojení k serveru systému zpráv.

Povinné vlastnosti pro spravované objekty typu Destination

Aplikace, která vytváří cíl, musí nastavit několik vlastností, které aplikace na administrovaného cílovém objektu.

IDestination (pro rozhraní .NET)

Cíl je místo, kam aplikace odesílá zprávy, nebo je to zdroj, ze kterého aplikace přijímá zprávy, nebo obojí.

Vlastnosti místa určení

Přehled vlastností objektu Destination, s odkazy na podrobnější referenční informace.

ConnectionFactory (pro rozhraní .NET)

Aplikace používá továrnu připojení k vytvoření připojení.

Vlastnosti objektu ConnectionFactory

Přehled vlastností objektu ConnectionFactory s odkazy na podrobnější referenční informace.

Povinné vlastnosti pro spravované objekty ConnectionFactory

Když aplikace vytvoří továrnu připojení, musí být definován počet vlastností pro vytvoření připojení k serveru systému zpráv.

Vlastnosti uvedené v následujících tabulkách jsou minimální požadavky, které má aplikace nastavit k vytvoření připojení k serveru systému zpráv. Chcete-li upravit způsob vytvoření připojení, může vaše aplikace podle potřeby nastavit případně další vlastnosti objektu ConnectionFactory . Další informace naleznete v části "Vlastnosti objektu ConnectionFactory" na stránce 175. Součástí je úplný seznam dostupných vlastností.

Připojení ke správci front produktu IBM MQ

Tabulka 10. Nastavení vlastností pro spravované objekty ConnectionFactory pro připojení ke správci front produktu IBM MQ

PovinnéXMS	Je vyžadována ekvivalentní vlastnost produktu IBM MQ JMS
<u>XMSC_CONNECTION_TYPE</u>	Produkt XMS to používá z názvu třídy faktorie připojení a vlastnosti TRANSPORT (TRAN).
<u>NÁZEV_HOSTITEL_XMSC</u>	NÁZEV HOSTITELE (HOSTITELE)
<u>PORT_XMSCS_WMQ_PORT</u>	PORT
<u>XMSCS_WMQ_QUEUE_MANAGER</u>	Název správce front

Připojení v reálném čase ke zprostředkovateli

Tabulka 11. Nastavení vlastností pro spravované objekty *ConnectionFactory* pro připojení v reálném čase ke zprostředkovateli

PovinnéXMS	Je vyžadována ekvivalentní vlastnost produktu IBM MQ JMS
<u>XMSC_CONNECTION_TYPE</u>	Produkt XMS to používá z názvu třídy faktorie připojení a vlastnosti TRANSPORT (TRAN).
<u>NÁZEV_HOSTITELE XMSCS_RTT_NAME</u>	NÁZEV HOSTITELE (HOSTITELE)
<u>VÝPORT XMSCS_RTT_PORT</u>	PORT

Připojení k WebSphere Application Server service integration bus

Tabulka 12. Nastavení vlastností pro spravované objekty *ConnectionFactory* pro připojení k serveru *WebSphere Application Server service integration bus*

XMS vlastnost	Popis
<u>XMSC_CONNECTION_TYPE</u>	Typ serveru systému zpráv, ke kterému se aplikace připojuje. Tato hodnota je určena z názvu třídy továrny připojení.
<u>XMSCS_WPM_BUS_NAME</u>	U továrny připojení se jedná o název sběrnice SIBus, ke které se aplikace připojuje, nebo pro cíl, název sběrnice SIBus, ve které cíl existuje.

Související pojmy

Podporované typy úložiště spravovaných objektů

Objekty systému souborů a LDAP spravované objekty lze použít pro připojení k produktu IBM MQ a WebSphere Application Server, zatímco pojmenování COS lze použít k připojení pouze k WebSphere Application Server.

Mapování vlastností pro spravované objekty

Chcete-li aplikacím povolit použití továrny připojení a definic cílových objektů produktu IBM MQ JMS a WebSphere Application Server, musí být vlastnosti načtené z těchto definic mapovány na odpovídající vlastnosti produktu XMS, které lze nastavit v továrnách a cílech připojení XMS.

Vlastnosti InitialContext

Parametry konstruktoru *InitialContext* zahrnují umístění úložiště spravovaných objektů, které jsou poskytnuty jako identifikátor URI (Uniform Resource Indicator). Aby mohla aplikace vytvořit připojení k úložišti, může být nezbytné poskytnout více informací než informace obsažené v identifikátoru URI.

Formát identifikátoru URI pro počáteční kontexty produktu XMS

Umístění úložiště spravovaných objektů je poskytováno jako jednotný indikátor prostředku (URI). Formát identifikátoru URI závisí na typu kontextu.

Webová služba vyhledávání JNDI

Chcete-li přistupovat k adresáři pojmenování COS z produktu XMS, musí být na serveru WebSphere Application Server service integration bus implementována webová služba vyhledávání produktu JNDI. Tato webová služba přeloží informace o produktu Java ze služby názvů COS do formuláře, který lze číst v aplikacích produktu XMS.

Načítání spravovaných objektů

Produkt XMS načítá administrovaný objekt z úložiště s použitím adresy zadané při vytvoření objektu *InitialContext* nebo ve vlastnostech *InitialContext*.

Zabezpečená připojení ke správci front produktu IBM MQ

Chcete-li povolit aplikaci XMS .NET zabezpečené připojení ke správci front produktu IBM MQ, musí být příslušné vlastnosti definovány v objektu *ConnectionFactory*.

Zabezpečená připojení ke stroji systému zpráv produktu WebSphere Application Server service integration bus

Chcete-li povolit aplikaci XMS .NET zabezpečené připojení ke stroji systému zpráv produktu WebSphere Application Server service integration bus , musí být příslušné vlastnosti definovány v objektu ConnectionFactory .

Související úlohy

Vytvoření spravovaných objektů

Definice objektu ConnectionFactory a Cílový objekt, které aplikace produktu XMS vyžadují k vytvoření připojení k serveru systému zpráv, musí být vytvořeny pomocí příslušných nástrojů pro správu.

Objekty InitialContext

Aplikace musí vytvořit počáteční kontext, který má být použit k vytvoření připojení k úložišti spravovaných objektů, aby bylo možné načíst požadované spravované objekty.

Související odkazy

Povinné vlastnosti pro spravované objekty typu Destination

Aplikace, která vytváří cíl, musí nastavit několik vlastností, které aplikace na administrovaného cílovém objektu.

[IConnectionFactory \(pro rozhraní .NET\)](#)

Aplikace používá továrnu připojení k vytvoření připojení.

[Vlastnosti objektu ConnectionFactory](#)

Přehled vlastností objektu ConnectionFactory s odkazy na podrobnější referenční informace.

Povinné vlastnosti pro spravované objekty typu Destination

Aplikace, která vytváří cíl, musí nastavit několik vlastností, které aplikace na administrovaného cílovém objektu.

Tabulka 13. Nastavení vlastností pro spravované cílové objekty		
Typ připojení	Vlastnost	Popis
IBM MQ správce front	FRONTA (QU)	Fronta, ke které se chcete připojit
	TÉMA (NAHOŘE)	Téma, které aplikace používá jako místo určení
Připojení v reálném čase ke zprostředkovateli	TÉMA (NAHOŘE)	Téma, které aplikace používá jako místo určení
WebSphere Application Server service integration bus	topicName	Pokud se vaše aplikace připojuje k tématu
	queueName	Pokud se vaše aplikace připojuje ke frontě

Související pojmy

Podporované typy úložiště spravovaných objektů

Objekty systému souborů a LDAP spravované objekty lze použít pro připojení k produktu IBM MQ a WebSphere Application Server, zatímco pojmenování COS lze použít k připojení pouze k WebSphere Application Server.

[Mapování vlastností pro spravované objekty](#)

Chcete-li aplikacím povolit použití továrny připojení a definic cílových objektů produktu IBM MQ JMS a WebSphere Application Server , musí být vlastnosti načtené z těchto definic mapovány na odpovídající vlastnosti produktu XMS , které lze nastavit v továrnách a cílech připojení XMS .

[Vlastnosti InitialContext](#)

Parametry konstrukturu InitialContext zahrnují umístění úložiště spravovaných objektů, které jsou poskytnuty jako identifikátor URI (Uniform Resource Indicator). Aby mohla aplikace vytvořit připojení k úložišti, může být nezbytné poskytnout více informací než informace obsažené v identifikátoru URI.

Formát identifikátoru URI pro počáteční kontexty produktu XMS

Umístění úložiště spravovaných objektů je poskytováno jako jednotný indikátor prostředku (URI). Formát identifikátoru URI závisí na typu kontextu.

Webová služba vyhledávání JNDI

Chcete-li přistupovat k adresáři pojmenování COS z produktu XMS, musí být na serveru WebSphere Application Server service integration bus implementována webová služba vyhledávání produktu JNDI . Tato webová služba přeloží informace o produktu Java ze služby názvů COS do formuláře, který lze číst v aplikacích produktu XMS .

Načítání spravovaných objektů

Produkt XMS načítá administrovaný objekt z úložiště s použitím adresy zadané při vytvoření objektu InitialContext nebo ve vlastnostech InitialContext .

Související úlohy

Vytvoření spravovaných objektů

Definice objektu ConnectionFactory a Cílový objekt, které aplikace produktu XMS vyžadují k vytvoření připojení k serveru systému zpráv, musí být vytvořeny pomocí příslušných nástrojů pro správu.

Objekty InitialContext

Aplikace musí vytvořit počáteční kontext, který má být použit k vytvoření připojení k úložišti spravovaných objektů, aby bylo možné načíst požadované spravované objekty.

Související odkazy

Povinné vlastnosti pro spravované objekty ConnectionFactory

Když aplikace vytvoří továrnu připojení, musí být definován počet vlastností pro vytvoření připojení k serveru systému zpráv.

IDestination (pro rozhraní .NET)

Cíl je místo, kam aplikace odesílá zprávy, nebo je to zdroj, ze kterého aplikace přijímá zprávy, nebo obojí.

Vlastnosti místa určení

Přehled vlastností objektu Destination, s odkazy na podrobnější referenční informace.

Vytvoření spravovaných objektů

Definice objektu ConnectionFactory a Cílový objekt, které aplikace produktu XMS vyžadují k vytvoření připojení k serveru systému zpráv, musí být vytvořeny pomocí příslušných nástrojů pro správu.

Než začnete

Další podrobnosti o různých typech spravovaných úložišť objektů, které produkt XMS podporuje, naleznete v tématu [“Podporované typy úložiště spravovaných objektů”](#) na stránce 48.

Informace o této úloze

Chcete-li vytvořit administrované objekty pro produkt IBM MQ , použijte nástroj administrace platformy JMS IBM MQ Explorer nebo IBM MQ (JMSAdmin).

Chcete-li vytvořit administrované objekty pro produkt IBM MQ nebo IBM Integration Bus, použijte nástroj pro administraci JMS (JMSAdmin) produktu IBM MQ .

Chcete-li vytvořit spravované objekty pro produkt WebSphere Application Server service integration bus, použijte administrativní konzolu serveru WebSphere Application Server .

Následující kroky shrnují, co děláte pro vytvoření spravovaných objektů.

Postup

1. Vytvořte továrnu na připojení a definujte potřebné vlastnosti pro vytvoření připojení z vaší aplikace k vybranému serveru.

Minimální vlastnosti, které produkt XMS vyžaduje k vytvoření připojení, jsou definovány v produktu [“Povinné vlastnosti pro spravované objekty ConnectionFactory”](#) na stránce 52.

2. Vytvořte požadovaný cíl na serveru zasílání zpráv, ke kterému se aplikace připojuje:

- Pro připojení ke správci front produktu IBM MQ vytvořte frontu nebo téma.
- V případě připojení v reálném čase ke zprostředkovateli vytvořte téma.
- Pro připojení k produktu WebSphere Application Server service integration bus vytvořte frontu nebo téma.

Minimální vlastnosti, které produkt XMS vyžaduje k vytvoření připojení, jsou definovány v produktu [“Povinné vlastnosti pro spravované objekty typu Destination”](#) na stránce 54.

Související pojmy

Podporované typy úložiště spravovaných objektů

Objekty systému souborů a LDAP spravované objekty lze použít pro připojení k produktu IBM MQ a WebSphere Application Server, zatímco pojmenování COS lze použít k připojení pouze k WebSphere Application Server.

Mapování vlastností pro spravované objekty

Chcete-li aplikacím povolit použití továrny připojení a definic cílových objektů produktu IBM MQ JMS a WebSphere Application Server, musí být vlastnosti načtené z těchto definic mapovány na odpovídající vlastnosti produktu XMS, které lze nastavit v továrnách a cílech připojení XMS.

Vlastnosti InitialContext

Parametry konstruktoru InitialContext zahrnují umístění úložiště spravovaných objektů, které jsou poskytnuty jako identifikátor URI (Uniform Resource Indicator). Aby mohla aplikace vytvořit připojení k úložišti, může být nezbytné poskytnout více informací než informace obsažené v identifikátoru URI.

Formát identifikátoru URI pro počáteční kontexty produktu XMS

Umístění úložiště spravovaných objektů je poskytováno jako jednotný indikátor prostředku (URI). Formát identifikátoru URI závisí na typu kontextu.

Webová služba vyhledávání JNDI

Chcete-li přistupovat k adresáři pojmenování COS z produktu XMS, musí být na serveru WebSphere Application Server service integration bus implementována webová služba vyhledávání produktu JNDI. Tato webová služba přeloží informace o produktu Java ze služby názvů COS do formuláře, který lze číst v aplikacích produktu XMS.

Načítání spravovaných objektů

Produkt XMS načítá administrovaný objekt z úložiště s použitím adresy zadané při vytvoření objektu InitialContext nebo ve vlastnostech InitialContext.

Spravované objekty

Pomocí spravovaných objektů můžete spravovat nastavení připojení používaná klientskými aplikacemi, které mají být spravovány z centrálního úložiště. Aplikace načítá definice objektů z centrálního úložiště a používá je k vytváření objektů ConnectionFactory a Destination. Pomocí spravovaných objektů můžete odstranit několik aplikací z prostředků, které používají za běhu.

ConnectionFactoryies a objekty připojení

Objekt ConnectionFactory poskytuje šablonu, kterou aplikace používá k vytvoření objektu připojení. Aplikace používá objekt Connection k vytvoření objektu relace.

Připojení ke sběrnici pro integraci služeb

Aplikace XMS se může připojit ke sběrnici pro integraci služeb produktu WebSphere Application Server buď pomocí přímého připojení TCP/IP, nebo pomocí protokolu HTTP přes TCP/IP.

Související úlohy

Objekty InitialContext

Aplikace musí vytvořit počáteční kontext, který má být použit k vytvoření připojení k úložišti spravovaných objektů, aby bylo možné načíst požadované spravované objekty.

Související odkazy

Povinné vlastnosti pro spravované objekty ConnectionFactory

Když aplikace vytvoří továrnu připojení, musí být definován počet vlastností pro vytvoření připojení k serveru systému zpráv.

Povinné vlastnosti pro spravované objekty typu Destination

Aplikace, která vytváří cíl, musí nastavit několik vlastností, které aplikace na administrovaného cílovém objektu.

IConnectionFactory (pro rozhraní .NET)

Aplikace používá továrnu připojení k vytvoření připojení.

Vlastnosti objektu ConnectionFactory

Přehled vlastností objektu ConnectionFactory s odkazy na podrobnější referenční informace.

IDestination (pro rozhraní .NET)

Cíl je místo, kam aplikace odesílá zprávy, nebo je to zdroj, ze kterého aplikace přijímá zprávy, nebo obojí.

Vlastnosti místa určení

Přehled vlastností objektu Destination, s odkazy na podrobnější referenční informace.

Objekty InitialContext

Aplikace musí vytvořit počáteční kontext, který má být použit k vytvoření připojení k úložišti spravovaných objektů, aby bylo možné načíst požadované spravované objekty.

Informace o této úloze

Objekt InitialContext zapouzdřuje připojení do úložiště. Rozhraní API produktu XMS poskytuje metody k provádění následujících úloh:

- Vytvoření objektu InitialContext
- Vyhledejte administrovaný objekt v úložišti spravovaných objektů.

Další podrobnosti o vytvoření objektu InitialContext viz [“InitialContext” na stránce 106](#) pro .NET a [“Vlastnosti objektu InitialContext” na stránce 183](#).

Související pojmy

Podporované typy úložiště spravovaných objektů

Objekty systému souborů a LDAP spravované objekty lze použít pro připojení k produktu IBM MQ a WebSphere Application Server, zatímco pojmenování COS lze použít k připojení pouze k WebSphere Application Server.

Mapování vlastností pro spravované objekty

Chcete-li aplikacím povolit použití továrny připojení a definic cílových objektů produktu IBM MQ JMS a WebSphere Application Server, musí být vlastnosti načtené z těchto definic mapovány na odpovídající vlastnosti produktu XMS, které lze nastavit v továrnách a cílech připojení XMS.

Vlastnosti InitialContext

Parametry konstruktoru InitialContext zahrnují umístění úložiště spravovaných objektů, které jsou poskytnuty jako identifikátor URI (Uniform Resource Indicator). Aby mohla aplikace vytvořit připojení k úložišti, může být nezbytné poskytnout více informací než informace obsažené v identifikátoru URI.

Formát identifikátoru URI pro počáteční kontexty produktu XMS

Umístění úložiště spravovaných objektů je poskytováno jako jednotný indikátor prostředku (URI). Formát identifikátoru URI závisí na typu kontextu.

Webová služba vyhledávání JNDI

Chcete-li přistupovat k adresáři pojmenování COS z produktu XMS, musí být na serveru WebSphere Application Server service integration bus implementována webová služba vyhledávání produktu JNDI. Tato webová služba přeloží informace o produktu Java ze služby názvů COS do formuláře, který lze číst v aplikacích produktu XMS.

Načítání spravovaných objektů

Produkt XMS načítá administrovaný objekt z úložiště s použitím adresy zadané při vytvoření objektu InitialContext nebo ve vlastnostech InitialContext.

Související úlohy

Vytvoření spravovaných objektů

Definice objektu `ConnectionFactory` a cílový objekt, které aplikace produktu XMS vyžadují k vytvoření připojení k serveru systému zpráv, musí být vytvořeny pomocí příslušných nástrojů pro správu.

Související odkazy

Povinné vlastnosti pro spravované objekty `ConnectionFactory`

Když aplikace vytvoří továrnu připojení, musí být definován počet vlastností pro vytvoření připojení k serveru systému zpráv.

Povinné vlastnosti pro spravované objekty typu `Destination`

Aplikace, která vytváří cíl, musí nastavit několik vlastností, které aplikace na administrovaného cílovém objektu.

InitialContext (pro rozhraní .NET)

Aplikace používá objekt `InitialContext` k vytvoření objektů z definic objektů, které jsou načteny z úložiště spravovaných objektů.

Vlastnosti objektu `InitialContext`

Přehled vlastností objektu `InitialContext` s odkazy na podrobnější referenční informace.

Vlastnosti `InitialContext`

Parametry konstruktoru `InitialContext` zahrnují umístění úložiště spravovaných objektů, které jsou poskytnuty jako identifikátor URI (Uniform Resource Indicator). Aby mohla aplikace vytvořit připojení k úložišti, může být nezbytné poskytnout více informací než informace obsažené v identifikátoru URI.

V rozhraní JNDI a v implementaci produktu .NET v produktu XMS jsou další informace poskytnuty v prostředí `Hashtable` prostředí konstrukturu.

Umístění spravovaného úložiště objektů je definováno ve vlastnosti `XMSCS_IC_URL`. Tato vlastnost se obvykle předává při volání `Create`, ale lze ji upravit tak, aby se připojovala k jinému adresáři pojmenování před vyhledáním. Pro kontexty `FileSystem` nebo `LDAP` tato vlastnost definuje adresu adresáře. Pro pojmenování `COS` se jedná o adresu webové služby, která tyto vlastnosti používá pro připojení k adresáři rozhraní JNDI.

Následující vlastnosti jsou předány nezměněné webové službě, která je použije pro připojení k adresáři rozhraní JNDI.

- Adresa URL `XMSCS_IC_PROVIDER_URL`
- `XMSC_IC_SECURITY_CREDENTIALS`
- `XMSC_IC_SECURITY_AUTHENTICATION`
- `XMSC_IC_SECURITY_PRINCIPAL`
- `XMSC_IC_SECURITY_PROTOCOL`

Související pojmy

Podporované typy úložiště spravovaných objektů

Objekty systému souborů a `LDAP` spravované objekty lze použít pro připojení k produktu `IBM MQ` a `WebSphere Application Server`, zatímco pojmenování `COS` lze použít k připojení pouze k `WebSphere Application Server`.

Mapování vlastností pro spravované objekty

Chcete-li aplikacím povolit použití továrny připojení a definic cílových objektů produktu `IBM MQ JMS` a `WebSphere Application Server`, musí být vlastnosti načtené z těchto definic mapovány na odpovídající vlastnosti produktu XMS, které lze nastavit v továrnách a cílech připojení XMS.

Formát identifikátoru URI pro počáteční kontexty produktu XMS

Umístění úložiště spravovaných objektů je poskytováno jako jednotný indikátor prostředku (URI). Formát identifikátoru URI závisí na typu kontextu.

Webová služba vyhledávání JNDI

Chcete-li přistupovat k adresáři pojmenování `COS` z produktu XMS, musí být na serveru `WebSphere Application Server service integration bus` implementována webová služba vyhledávání produktu JNDI. Tato webová služba přeloží informace o produktu Java ze služby názvů `COS` do formuláře, který lze číst v aplikacích produktu XMS.

Načítání spravovaných objektů

Produkt XMS načítá administrovaný objekt z úložiště s použitím adresy zadané při vytvoření objektu InitialContext nebo ve vlastnostech InitialContext .

Související úlohy

Vytvoření spravovaných objektů

Definice objektu ConnectionFactory a Cílový objekt, které aplikace produktu XMS vyžadují k vytvoření připojení k serveru systému zpráv, musí být vytvořeny pomocí příslušných nástrojů pro správu.

Objekty InitialContext

Aplikace musí vytvořit počáteční kontext, který má být použit k vytvoření připojení k úložišti spravovaných objektů, aby bylo možné načíst požadované spravované objekty.

Související odkazy

Povinné vlastnosti pro spravované objekty ConnectionFactory

Když aplikace vytvoří továrnu připojení, musí být definován počet vlastností pro vytvoření připojení k serveru systému zpráv.

Povinné vlastnosti pro spravované objekty typu Destination

Aplikace, která vytváří cíl, musí nastavit několik vlastností, které aplikace na administrovaného cílovém objektu.

InitialContext (pro rozhraní .NET)

Aplikace používá objekt InitialContext k vytvoření objektů z definic objektů, které jsou načteny z úložiště spravovaných objektů.

Vlastnosti objektu InitialContext

Přehled vlastností objektu InitialContext s odkazy na podrobnější referenční informace.

Formát identifikátoru URI pro počáteční kontexty produktu XMS

Umístění úložiště spravovaných objektů je poskytováno jako jednotný indikátor prostředku (URI). Formát identifikátoru URI závisí na typu kontextu.

Kontext FileSystem

Pro kontext FileSystem je adresa URL umístění adresáře založeného na systému souborů. Struktura adresy URL je dle definice RFC 1738, *Uniform Resource Locators (URL)*: adresa URL má předponu file:// a syntaxe následující za touto předponou je platná definice souboru, který lze otevřít na systému, na kterém je spuštěn produkt XMS .

Tato syntaxe může být specifická pro platformu a může být použita buď '/separators, nebo' \ '. Použijete-li '\', pak každý oddělovač musí být změněn pomocí přídavného '\'. Tím zabráníte tomu, aby se rámec .NET pokusil interpretovat oddělovač jako řídicí znak pro to, co následuje.

Následující příklady ilustrují tuto syntaxi:

```
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
```

Kontext protokolu LDAP

Pro kontext protokolu LDAP je základní struktura adresy URL definována pomocí RFC 2255, *Formát adresy URL LDAPs* předponou ldap:// bez rozlišení malých a velkých písmen.

Přesná syntaxe je ilustrována v následujícím příkladu:

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

Tato syntaxe je definována ve specifikaci RFC, ale bez podpory pro jakékoli atributy, rozsah, filtry nebo rozšíření.

Mezi příklady této syntaxe patří:

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

Kontext WSS

Pro kontext WSS se adresa URL nachází ve formě koncového bodu webových služeb s předponou `http://`.

Případně můžete použít předponu `cosnaming://` nebo `wsvc://`,

Tyto dvě předpony jsou interpretovány jako význam, že používáte kontext WSS s adresou URL, ke které přistupujete prostřednictvím protokolu `http`, který umožňuje snadno odvodit počáteční typ kontextu přímo z adresy URL.

Mezi příklady této syntaxe patří následující:

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

Související pojmy

Podporované typy úložiště spravovaných objektů

Objekty systému souborů a LDAP spravované objekty lze použít pro připojení k produktu IBM MQ a WebSphere Application Server, zatímco pojmenování COS lze použít k připojení pouze k WebSphere Application Server.

Mapování vlastností pro spravované objekty

Chcete-li aplikacím povolit použití továrny připojení a definic cílových objektů produktu IBM MQ JMS a WebSphere Application Server, musí být vlastnosti načtené z těchto definic mapovány na odpovídající vlastnosti produktu XMS, které lze nastavit v továrnách a cílech připojení XMS.

Vlastnosti InitialContext

Parametry konstruktoru `InitialContext` zahrnují umístění úložiště spravovaných objektů, které jsou poskytnuty jako identifikátor URI (Uniform Resource Indicator). Aby mohla aplikace vytvořit připojení k úložišti, může být nezbytné poskytnout více informací než informace obsažené v identifikátoru URI.

Webová služba vyhledávání JNDI

Chcete-li přistupovat k adresáři pojmenování COS z produktu XMS, musí být na serveru WebSphere Application Server service integration bus implementována webová služba vyhledávání produktu JNDI. Tato webová služba přeloží informace o produktu Java ze služby názvů COS do formuláře, který lze číst v aplikacích produktu XMS.

Načítání spravovaných objektů

Produkt XMS načítá administrovaný objekt z úložiště s použitím adresy zadané při vytvoření objektu `InitialContext` nebo ve vlastnostech `InitialContext`.

Související úlohy

Vytvoření spravovaných objektů

Definice objektu `ConnectionFactory` a cílový objekt, které aplikace produktu XMS vyžadují k vytvoření připojení k serveru systému zpráv, musí být vytvořeny pomocí příslušných nástrojů pro správu.

Objekty InitialContext

Aplikace musí vytvořit počáteční kontext, který má být použit k vytvoření připojení k úložišti spravovaných objektů, aby bylo možné načíst požadované spravované objekty.

Související odkazy

Povinné vlastnosti pro spravované objekty `ConnectionFactory`

Když aplikace vytvoří továrnu připojení, musí být definován počet vlastností pro vytvoření připojení k serveru systému zpráv.

Povinné vlastnosti pro spravované objekty typu `Destination`

Aplikace, která vytváří cíl, musí nastavit několik vlastností, které aplikace na administrovaného cílovém objektu.

InitialContext (pro rozhraní .NET)

Aplikace používá objekt InitialContext k vytvoření objektů z definic objektů, které jsou načteny z úložiště spravovaných objektů.

Vlastnosti objektu InitialContext

Přehled vlastností objektu InitialContext s odkazy na podrobnější referenční informace.

Webová služba vyhledávání JNDI

Chcete-li přistupovat k adresáři pojmenování COS z produktu XMS, musí být na serveru WebSphere Application Server service integration bus implementována webová služba vyhledávání produktu JNDI . Tato webová služba přeloží informace o produktu Java ze služby názvů COS do formuláře, který lze číst v aplikacích produktu XMS .

Webová služba je k dispozici v souboru podnikového archivu SIBXJndiLookupEAR.ear, který je umístěn v instalačním adresáři. V případě aktuálního vydání produktu IBM Message Service Client for .NET lze v adresáři `install_dir\java\lib` nalézt soubor SIBXJndiLookupEAR.ear . Lze jej instalovat v rámci serveru WebSphere Application Server service integration bus pomocí administrativní konzoly nebo skriptovacího nástroje `wsadmin`. Další informace o implementaci aplikací webových služeb najdete v dokumentaci k produktu.

Chcete-li definovat webovou službu v rámci aplikací produktu XMS , stačí jednoduše nastavit vlastnost `XMSCS_IC_URL` objektu InitialContext na adresu URL koncového bodu webové služby. Je-li například webová služba implementována na hostiteli serveru s názvem `MyServer`, je příkladem adresy URL koncového bodu webové služby:

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

Nastavení vlastnosti `XMSCS_IC_URL` umožňuje voláním vyhledávání InitialContext vyvolat webovou službu na definovaném koncovém bodu, který zase vyhledá požadovaný spravovaný objekt ze služby názvů COS.

Aplikace produktu .NET mohou používat webovou službu. Implementace na straně serveru je stejná pro produkty XMS C, /C++ a XMS .NET. XMS .NET vyvolá webovou službu přímo prostřednictvím rámce Microsoft .NET .

Související pojmy

Podporované typy úložiště spravovaných objektů

Objekty systému souborů a LDAP spravované objekty lze použít pro připojení k produktu IBM MQ a WebSphere Application Server, zatímco pojmenování COS lze použít k připojení pouze k WebSphere Application Server.

Mapování vlastností pro spravované objekty

Chcete-li aplikacím povolit použití továrny připojení a definic cílových objektů produktu IBM MQ JMS a WebSphere Application Server , musí být vlastnosti načtené z těchto definic mapovány na odpovídající vlastnosti produktu XMS , které lze nastavit v továrnách a cílech připojení XMS .

Vlastnosti InitialContext

Parametry konstruktora InitialContext zahrnují umístění úložiště spravovaných objektů, které jsou poskytnuty jako identifikátor URI (Uniform Resource Indicator). Aby mohla aplikace vytvořit připojení k úložišti, může být nezbytné poskytnout více informací než informace obsažené v identifikátoru URI.

Formát identifikátoru URI pro počáteční kontexty produktu XMS

Umístění úložiště spravovaných objektů je poskytováno jako jednotný indikátor prostředku (URI). Formát identifikátoru URI závisí na typu kontextu.

Načítání spravovaných objektů

Produkt XMS načítá administrovaný objekt z úložiště s použitím adresy zadané při vytvoření objektu InitialContext nebo ve vlastnostech InitialContext .

Nastavení prostředí serveru systému zpráv

Témata v této sekci popisují, jak nastavit prostředí serveru systému zpráv, aby bylo aplikacím produktu XMS umožněno připojit se k serveru.

Použití ukázkových aplikací produktu XMS

Použijte ukázkové aplikace dodávané s produktem XMS k ověření instalace a nastavení serveru systému zpráv a k usnadnění sestavování vlastních aplikací. Ukázky poskytují přehled běžných funkcí jednotlivých rozhraní API.

Související úlohy

Vytvoření spravovaných objektů

Definice objektu ConnectionFactory a Cílový objekt, které aplikace produktu XMS vyžadují k vytvoření připojení k serveru systému zpráv, musí být vytvořeny pomocí příslušných nástrojů pro správu.

Objekty InitialContext

Aplikace musí vytvořit počáteční kontext, který má být použit k vytvoření připojení k úložišti spravovaných objektů, aby bylo možné načíst požadované spravované objekty.

Instalace produktu Message Service Client for .NET pomocí průvodce instalací

Instalace používá instalační program produktu InstallShield X/Windows MSI. K dispozici jsou dvě volby nastavení, takže můžete zvolit buď úplnou, nebo vlastní instalaci.

Související odkazy

Povinné vlastnosti pro spravované objekty ConnectionFactory

Když aplikace vytvoří továrnu připojení, musí být definován počet vlastností pro vytvoření připojení k serveru systému zpráv.

Povinné vlastnosti pro spravované objekty typu Destination

Aplikace, která vytváří cíl, musí nastavit několik vlastností, které aplikace na administrovaného cílovém objektu.

Načítání spravovaných objektů

Produkt XMS načítá administrovaný objekt z úložiště s použitím adresy zadané při vytvoření objektu InitialContext nebo ve vlastnostech InitialContext .

Objekty, které se mají načíst, mohou mít následující typy názvů:

- Prostý název popisující cílový objekt, například místo určení fronty s názvem SalesOrders .
- Složený název, který může být vytvořen z SubContexts, oddělených znakem '/' a musí končit názvem objektu. Příkladem složeného názvu je "Warehouse/PickLists/DispatchQueue2", kde Warehouse a Picklists jsou SubContexts v adresáři pojmenování a DispatchQueue2 je název objektu Destination.

Související pojmy

Podporované typy úložiště spravovaných objektů

Objekty systému souborů a LDAP spravované objekty lze použít pro připojení k produktu IBM MQ a WebSphere Application Server, zatímco pojmenování COS lze použít k připojení pouze k WebSphere Application Server.

Mapování vlastností pro spravované objekty

Chcete-li aplikacím povolit použití továrny připojení a definic cílových objektů produktu IBM MQ JMS a WebSphere Application Server , musí být vlastnosti načtené z těchto definic mapovány na odpovídající vlastnosti produktu XMS , které lze nastavit v továrnách a cílech připojení XMS .

Vlastnosti InitialContext

Parametry konstruktora InitialContext zahrnují umístění úložiště spravovaných objektů, které jsou poskytnuty jako identifikátor URI (Uniform Resource Indicator). Aby mohla aplikace vytvořit připojení k úložišti, může být nezbytné poskytnout více informací než informace obsažené v identifikátoru URI.

Formát identifikátoru URI pro počáteční kontexty produktu XMS

Umístění úložiště spravovaných objektů je poskytováno jako jednotný indikátor prostředku (URI). Formát identifikátoru URI závisí na typu kontextu.

Webová služba vyhledávání JNDI

Chcete-li přistupovat k adresáři pojmenování COS z produktu XMS, musí být na serveru WebSphere Application Server service integration bus implementována webová služba vyhledávání produktu JNDI . Tato webová služba přeloží informace o produktu Java ze služby názvů COS do formuláře, který lze číst v aplikacích produktu XMS .

Související úlohy

Vytvoření spravovaných objektů

Definice objektu ConnectionFactory a Cílový objekt, které aplikace produktu XMS vyžadují k vytvoření připojení k serveru systému zpráv, musí být vytvořeny pomocí příslušných nástrojů pro správu.

Objekty InitialContext

Aplikace musí vytvořit počáteční kontext, který má být použit k vytvoření připojení k úložišti spravovaných objektů, aby bylo možné načíst požadované spravované objekty.

Související odkazy

Povinné vlastnosti pro spravované objekty ConnectionFactory

Když aplikace vytvoří továrnu připojení, musí být definován počet vlastností pro vytvoření připojení k serveru systému zpráv.

Povinné vlastnosti pro spravované objekty typu Destination

Aplikace, která vytváří cíl, musí nastavit několik vlastností, které aplikace na administrovaného cílovém objektu.

InitialContext (pro rozhraní .NET)

Aplikace používá objekt InitialContext k vytvoření objektů z definic objektů, které jsou načteny z úložiště spravovaných objektů.

Vlastnosti objektu InitialContext

Přehled vlastností objektu InitialContext s odkazy na podrobnější referenční informace.

Zabezpečení komunikací pro aplikace XMS

Tento oddíl poskytuje informace o nastavení zabezpečených komunikací, které umožňují aplikacím produktu XMS připojit se prostřednictvím protokolu SSL (Secure Sockets Layer) ke stroji systému zpráv produktu WebSphere Application Server service integration bus nebo ke správci front IBM MQ .

Sekce obsahuje následující témata:

- [“Zabezpečená připojení ke správci front produktu IBM MQ” na stránce 63](#)
- [“Mapování názvů CipherSuite a CipherSpec pro připojení ke správci front produktu IBM MQ” na stránce 65](#)
- [“Zabezpečená připojení ke stroji systému zpráv produktu WebSphere Application Server service integration bus” na stránce 65](#)
- [“Mapování názvů CipherSuite a CipherSpec pro připojení k produktu WebSphere Application Server service integration bus” na stránce 66](#)

Zabezpečená připojení ke správci front produktu IBM MQ

Chcete-li povolit aplikaci XMS .NET zabezpečené připojení ke správci front produktu IBM MQ , musí být příslušné vlastnosti definovány v objektu ConnectionFactory .

Protokol použitý v dohadování šifrování může být buď SSL (Secure Sockets Layer), nebo TLS (Transport Layer Security), podle toho, kterou sadu CipherSuite zadáte v objektu ConnectionFactory .

Používáte-li knihovny klienta IBM WebSphere MQ 7.0.0 Fix Pack 1 a novější a připojujete se ke správci front produktu IBM WebSphere MQ 7.0 , můžete v aplikaci XMS vytvořit více připojení ke stejnému správci front. Připojení k jinému správci front však není povoleno. Pokusíte-li se o tento pokus, obdržíte chybu MQRC_SSL_ALREADY_INITIALIZED .

Používáte-li knihovny klienta IBM WebSphere MQ 6.0 a novější, můžete vytvořit připojení SSL pouze tehdy, když nejprve uzavřete předchozí připojení SSL. Několik souběžných připojení SSL ze stejného procesu ke stejným nebo různým správcům front není povoleno. Pokusíte-li se více než jeden požadavek, obdržíte varování MQRC_SSL_ALREADY_INITIALIZED, což může znamenat, že některé požadované parametry pro připojení SSL byly ignorovány.

Vlastnosti ConnectionFactory pro připojení přes SSL ke správci front IBM MQ se stručným popisem jsou zobrazeny v následující tabulce:

<i>Tabulka 14. Vlastnosti objektu ConnectionFactory pro připojení ke správci front produktu IBM MQ prostřednictvím protokolu SSL</i>	
Název vlastnosti	Popis
<u>XMSCS_WMQ_SSL_CERT_STORES</u>	Umístění serverů obsahující seznamy odvolaných certifikátů (CRL), které mají být použity v připojení SSL ke správci front.
<u>XMSC_WMQ_SSL_CIPHER_SPEC</u>	Název specifikace CipherSpec, která má být použita v zabezpečeném připojení ke správci front.
<u>XMSCS_WMQ_SSL_CIPHER_SUITE</u>	Název sady CipherSuite, která má být použita v rámci připojení TLS ke správci front. Protokol použitý při vyjednávání zabezpečeného připojení závisí na určené sadě CipherSuite.
<u>XMSCS_WMQ_CRYPT_HW</u>	Podrobnosti konfigurace šifrovacího hardwaru připojeného k systému klienta.
<u>POŽAD_ZÁSOBNÍ_CHYBA_XMSC</u>	Hodnota této vlastnosti určuje, zda aplikace může nebo nemůže používat šifrovací sady, které nevyhovují standardu FIPS. Je-li tato vlastnost nastavena na hodnotu true, pro připojení klientského serveru se používají pouze algoritmy FIPS.
<u>XMSCS_WMQ_SSL_KEY_REPOSITORY</u>	Umístění souboru databáze klíčů, ve kterém jsou uloženy klíče a certifikáty.
<u>XMSCS_WMQ_SSL_KEY_RESETCOUNT</u>	Hodnota KeyResetCount představuje celkový počet nešifrovaných bajtů odeslaných a přijatých v rámci konverzace SSL, než je znovu vyjednána tajný klíč.
<u>XMSCS_WMQ_SSL_PEER_NAME</u>	Název typu peer, které se použije při připojení SSL ke správci front.

Související odkazy

IConnectionFactory (pro rozhraní .NET)

Aplikace používá továrnu připojení k vytvoření připojení.

Vlastnosti objektu ConnectionFactory

Přehled vlastností objektu ConnectionFactory s odkazy na podrobnější referenční informace.

Povinné vlastnosti pro spravované objekty ConnectionFactory

Když aplikace vytvoří továrnu připojení, musí být definován počet vlastností pro vytvoření připojení k serveru systému zpráv.

Mapování názvů CipherSuite a CipherSpec pro připojení ke správci front produktu IBM MQ

Položka InitialContext přeloží mezi vlastností továrny připojení JMSAdmin a hodnotou SSLCIPHERSUITE a XMS téměř ekvivalentní hodnotou XMSCS_WMQ_SSL_CIPHER_SPEC produktu XMSC. Podobný překlad je nezbytný, pokud zadáte hodnotu pro vlastnost XMSCS_WMQ_SSL_CIPHER_SUITE, ale vynechte hodnotu pro vlastnost XMSCS_WMQ_SSL_CIPHER_SPEC.

Produkt [Tabulka 15 na stránce 65](#) uvádí seznam dostupných ekvivalentů CipherSpecs a jejich JSSE CipherSuite .

CipherSpec	Ekvivalentní JSSE CipherSuite
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

Poznámka:

- TLS_RSA_WITH_3DES_EDE_CBC_SHA je zamítnutý. Nicméně lze ji přesto použít k přenosu až 32 GB dat před ukončením připojení s chybou AMQ9288. Chcete-li se této chybě vyhnout, je třeba při použití této CipherSpecbuď zabránit použití trojitého DES, nebo povolit resetování tajného klíče.

Zabezpečená připojení ke stroji systému zpráv produktu WebSphere Application Server service integration bus

Chcete-li povolit aplikaci XMS .NET zabezpečené připojení ke stroji systému zpráv produktu WebSphere Application Server service integration bus , musí být příslušné vlastnosti definovány v objektu ConnectionFactory .

Produkt XMS poskytuje podporu zabezpečení SSL a HTTPS pro připojení k serveru WebSphere Service Integration Bus. SSL a HTTPS poskytují zabezpečené připojení pro ověření a utajení.

Podobně jako zabezpečení produktu WebSphere je zabezpečení produktu XMS konfigurováno s ohledem na standardy zabezpečení JSSE a konvence pojmenování, které zahrnují použití seznamu CipherSuites k určení algoritmů používaných při vyjednávání zabezpečeného připojení. Protokol použitý v dohadování šifrování může být buď SSL, nebo TLS, v závislosti na tom, kterou CipherSuite uvedete v objektu ConnectionFactory .

[Tabulka 16 na stránce 65](#) vypíše vlastnosti, které musí být definovány v objektu ConnectionFactory .

Název vlastnosti	Popis
<u>XMSEWPMSSL_CIPHER_SUITE</u>	Název sady CipherSuite , který má být použit pro připojení TLS k stroji systému zpráv produktu WebSphere Service Integration Bus . Protokol použitý při vyjednávání zabezpečeného připojení závisí na určené sadě CipherSuite.
<u>XMSCS_WPM_SSL_KEYRING_LABEL</u>	Certifikát, který má být použit při ověřování na serveru.

Dále je uveden příklad vlastností ConnectionFactory pro zabezpečená připojení ke stroji systému zpráv produktu 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);

```

Kde řetězový název by měl být nastaven na hodnotu `BootstrapTunneledSecureMessaging` nebo `BootstrapSecureMessaging` a číslo portu je číslo portu, na kterém server samozavedení naslouchá příchozím požadavkům.

Dále je uveden příklad vlastností `ConnectionFactory` pro zabezpečená připojení k stroji systému zpráv produktu `WebSphere Application Server service integration bus` s vloženými ukázkovými hodnotami:

```

/* 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");

```

Související odkazy

[IConnectionFactory](#) (pro rozhraní .NET)

Aplikace používá továrnu připojení k vytvoření připojení.

[Vlastnosti objektu ConnectionFactory](#)

Přehled vlastností objektu `ConnectionFactory` s odkazy na podrobnější referenční informace.

[Povinné vlastnosti pro spravované objekty ConnectionFactory](#)

Když aplikace vytvoří továrnu připojení, musí být definován počet vlastností pro vytvoření připojení k serveru systému zpráv.

Mapování názvů CipherSuite a CipherSpec pro připojení k produktu WebSphere Application Server service integration bus

Protože sada `GSKit` používá `CipherSpecs` spíše než `CipherSuites`, musí být názvy `CipherSuite` ve stylu `JSSE` uvedené ve vlastnosti `XMSCS_WPM_CIPHER_SUITE` mapovány na názvy `CipherSpec` stylu `GSKit`.

Tabulka 17 na stránce 66 uvádí ekvivalentní `CipherSpec` pro každou rozpoznanou sadu `CipherSuite`.

Tabulka 17. Dostupná sada CipherSuites a jejich ekvivalentní specifikace CipherSpecs	
CipherSuite	Ekvivalentní CipherSpec
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

Poznámka:

- `TLS_RSA_WITH_3DES_EDE_CBC_SHA` je zamítnutý. Nicméně lze ji přesto použít k přenosu až 32 GB dat před ukončením připojení s chybou `AMQ9288`. Chcete-li se této chybě vyhnout, je třeba při použití této `CipherSpec`buď zabránit použití trojitého DES, nebo povolit resetování tajného klíče.

Zprávy produktu XMS

Tento oddíl popisuje strukturu a obsah zpráv produktu XMS a vysvětluje, jak aplikace zpracovává zprávy produktu XMS .

Tento oddíl obsahuje následující témata:

- [“Části zprávy produktu XMS” na stránce 67](#)
- [“Pole záhlaví ve zprávě XMS” na stránce 67](#)

- [“Vlastnosti zprávy XMS” na stránce 68](#)
- [“Tělo zprávy produktu XMS” na stránce 72](#)
- [“Voliče zpráv” na stránce 77](#)
- [“Mapování zpráv produktu XMS na zprávy produktu IBM MQ” na stránce 78](#)

Související odkazy

[IMessage \(pro rozhraní .NET\)](#)

Objekt zpráv představuje zprávu, kterou aplikace odesílá nebo přijímá. IMessage je nadtřída pro třídy zpráv, jako např. IMapMessage.

Části zprávy produktu XMS

Zpráva XMS se skládá ze záhlaví, sady vlastností a těla.

Header

Záhlaví zprávy obsahuje pole a všechny zprávy obsahují stejnou sadu polí záhlaví. XMS a aplikace používají hodnoty polí záhlaví k identifikaci a směrování zpráv. Další informace o polích záhlaví viz [“Pole záhlaví ve zprávě XMS” na stránce 67](#).

Sada vlastností

Vlastnosti zprávy určují další informace o zprávě. Ačkoli všechny zprávy mají stejnou sadu polí záhlaví, každá zpráva může mít jinou sadu vlastností. Další informace naleznete v části [“Vlastnosti zprávy XMS” na stránce 68](#).

Tělo

Tělo zprávy obsahuje data aplikace. Další informace naleznete v části [“Tělo zprávy produktu XMS” na stránce 72](#).

Aplikace může vybrat zprávy, které chce přijímat. Pomocí selektorů zpráv, které určují kritéria výběru. Kritéria mohou být založena na hodnotách určitých polí záhlaví a na hodnotách libovolné vlastnosti zprávy. Další informace o selektorech zpráv viz [“Voliče zpráv” na stránce 77](#).

Související odkazy

[Pole záhlaví ve zprávě XMS](#)

Chcete-li povolit aplikaci XMS výměnu zpráv s aplikací WebSphere JMS , záhlaví zprávy XMS obsahuje pole záhlaví zprávy JMS .

[Vlastnosti zprávy XMS](#)

Produkt XMS podporuje tři druhy vlastností zprávy: JMS definované vlastnosti, IBM definované vlastnosti a vlastnosti definované aplikací.

[Tělo zprávy produktu XMS](#)

Tělo zprávy obsahuje data aplikace. Zpráva však může mít žádný text a skládá se pouze z polí záhlaví a vlastností.

[Voliče zpráv](#)

Aplikace XMS používá selektory zpráv k výběru zpráv, které chce přijímat.

[Mapování zpráv produktu XMS na zprávy produktu IBM MQ](#)

Pole záhlaví JMS a vlastnosti zprávy produktu XMS jsou mapovány na pole ve strukturách záhlaví zprávy IBM MQ .

Pole záhlaví ve zprávě XMS

Chcete-li povolit aplikaci XMS výměnu zpráv s aplikací WebSphere JMS , záhlaví zprávy XMS obsahuje pole záhlaví zprávy JMS .

Názvy těchto polí záhlaví začínají předponou JMS. Popis polí záhlaví zprávy JMS naleznete v příručce *Java Message Service Specification*.

XMS implementuje pole záhlaví zprávy JMS jako atributy objektu Message Object. Každé pole záhlaví má své vlastní metody pro nastavení a získání jeho hodnoty. Popis těchto metod viz [“IMessage” na stránce 119](#). Pole záhlaví je vždy čitelné a zapisovatelné.

Tabulka 18 na stránce 68 vypíše pole záhlaví zprávy JMS a uvádí, jak je hodnota každého pole nastavena pro vyslanou zprávu. Některá pole jsou nastavena automaticky produktem XMS , když aplikace odešle zprávu nebo, v případě JMSRedelivered, když aplikace obdrží zprávu.

<i>Tabulka 18. Pole záhlaví zprávy produktu JMS</i>	
Název pole záhlaví zprávy produktu JMS	Jak je hodnota nastavena pro přenesená zpráva (ve formátu metoda [třída])
JMSCorrelationID	Nastavte JMSCorrelationID [Zpráva]
JMSDeliveryMode	Odeslat [MessageProducer]
JMSDestination	Odeslat [MessageProducer]
JMSExpiration	Odeslat [MessageProducer]
JMSMessageID	Odeslat [MessageProducer]
JMSPriority.	Odeslat [MessageProducer]
JMSRedelivered	Přijmout [MessageConsumer]
JMSReplyTo	Nastavte JMSReplyTo [Zpráva]
JMSTimestamp	Odeslat [MessageProducer]
JMSType.	Nastavit JMSType [Zpráva]

Související odkazy

Části zprávy produktu XMS

Zpráva XMS se skládá ze záhlaví, sady vlastností a těla.

Vlastnosti zprávy XMS

Produkt XMS podporuje tři druhy vlastností zprávy: JMS definované vlastnosti, IBM definované vlastnosti a vlastnosti definované aplikací.

Tělo zprávy produktu XMS

Tělo zprávy obsahuje data aplikace. Zpráva však může mít žádný text a skládá se pouze z polí záhlaví a vlastností.

Voliče zpráv

Aplikace XMS používá selektory zpráv k výběru zpráv, které chce přijímat.

Mapování zpráv produktu XMS na zprávy produktu IBM MQ

Pole záhlaví JMS a vlastnosti zprávy produktu XMS jsou mapovány na pole ve strukturách záhlaví zprávy IBM MQ .

Vlastnosti zprávy XMS

Produkt XMS podporuje tři druhy vlastností zprávy: JMS definované vlastnosti, IBM definované vlastnosti a vlastnosti definované aplikací.

Aplikace XMS si může vyměňovat zprávy s aplikací WebSphere JMS , protože produkt XMS podporuje následující předdefinované vlastnosti objektu zprávy:

- Tytéž vlastnosti definované JMS, které produkt WebSphere JMS podporuje. Názvy těchto vlastností začínají předponou JMSX.
- Tytéž vlastnosti definované IBM, které produkt WebSphere JMS podporuje. Názvy těchto vlastností začínají předponou JMS_IBM_.

Každá předdefinovaná vlastnost má dva názvy:

- Název JMS , pro vlastnost definované uživatelem JMSnebo název WebSphere JMS pro vlastnost definované IBM.

Jedná se o název, pod kterým je vlastnost známá v produktu JMS nebo WebSphere JMS, a je to také název, který je přenášen se zprávou, která má tuto vlastnost. Aplikace XMS používá tento název k identifikaci vlastnosti ve výrazu selektoru zpráv.

- Název produktu XMS , který identifikuje vlastnost ve všech situacích s výjimkou výrazu selektoru zpráv. Každý název XMS je definován jako pojmenovaná konstanta ve třídě IBM .XMS .XMSC . Hodnota uvedené konstanty je odpovídajícím názvem JMS nebo WebSphere JMS .

Kromě předdefinovaných vlastností může aplikace XMS vytvářet a používat svou vlastní sadu vlastností zpráv. Tyto vlastnosti se nazývají *definované vlastnosti aplikace*.

Poté, co aplikace vytvoří zprávu, jsou vlastnosti zprávy přístupné pro čtení a zápis. Po odeslání zprávy zůstanou vlastnosti čitelné a přístupné pro zápis. Když aplikace přijme zprávu, vlastnosti této zprávy jsou jen pro čtení. Pokud aplikace volá metodu `Clear Properties` třídy `Message`, když jsou vlastnosti zprávy jen pro čtení, vlastnosti budou přístupné pro čtení a zápis. Metoda také vymaže vlastnosti.

Obdržená zpráva se při předání po vymazání vlastností zprávy chová způsobem, který je konzistentní s chováním při předávání standardního rozhraní WMQ XMS for .NET `BytesMessage` s vyčištěním vlastností zprávy.

To se však nedoporučuje, protože budou ztraceny následující vlastnosti:

- Hodnota vlastnosti `JMS_IBM_Encoding`, z čehož vyplývá, že data zprávy nelze smysluplně dekodovat.
- Hodnota vlastnosti `JMS_Format`, z čehož vyplývá, že hlavička záhlaví mezi záhlavím (MQMD nebo novým záhlavím MQRFH2) a existujícími záhlavími by byla porušena.

Chcete-li určit hodnoty všech vlastností zprávy, aplikace může volat metodu `Get Properties` třídy `Message`. Metoda vytvoří iterátor, který zapouzdří seznam objektů vlastností, kde každý objekt vlastností představuje vlastnost zprávy. Aplikace pak může použít metody třídy `Iterator` k načtení každého objektu `Vlastnost` a může použít metody třídy `Vlastnosti` k načtení názvu, datového typu a hodnoty jednotlivých vlastností.

Související odkazy

Části zprávy produktu XMS

Zpráva XMS se skládá ze záhlaví, sady vlastností a těla.

Pole záhlaví ve zprávě XMS

Chcete-li povolit aplikaci XMS výměnu zpráv s aplikací WebSphere JMS , záhlaví zprávy XMS obsahuje pole záhlaví zprávy JMS .

Tělo zprávy produktu XMS

Tělo zprávy obsahuje data aplikace. Zpráva však může mít žádný text a skládá se pouze z polí záhlaví a vlastností.

Voliče zpráv

Aplikace XMS používá selektory zpráv k výběru zpráv, které chce přijímat.

Mapování zpráv produktu XMS na zprávy produktu IBM MQ

Pole záhlaví JMS a vlastnosti zprávy produktu XMS jsou mapovány na pole ve strukturách záhlaví zprávy IBM MQ .

JMSdefinované vlastnosti zprávy

Několik vlastností zpráv definovaných službou JMS jsou podporovány jak XMS , tak WebSphere JMS.

Produkt [Tabulka 19](#) na stránce 70 obsahuje seznam vlastností zpráv definovaných službou JMS, které jsou podporovány jak XMS , tak WebSphere JMS. Popis vlastností definovaných produktem JMS naleznete v tématu *Specifikace Java Message Service*. Vlastnosti definované JMS nejsou platné pro připojení v reálném čase ke zprostředkovateli.

Tabulka uvádí datový typ každé vlastnosti a uvádí, jak je hodnota vlastnosti nastavena pro odeslanou zprávu. Některé vlastnosti jsou nastaveny automaticky produktem XMS , když aplikace odešle zprávu, nebo v případě `JMSXDeliveryCount`, když aplikace přijme zprávu.

Tabulka 19. JMSdefinované vlastnosti zprávy

Název XMS definované vlastnosti JMS	Název JMS	Datový typ	Jak je hodnota nastavena pro přenesená zpráva (ve formátu metoda [třída])
JMSX_APPID	JMSXAppID	System.String	Odeslat [MessageProducer]
POČET DORUČENÍ JMSX_DELIVERY_COUNT	JMSXDeliveryCount	System.Int32	Přijmout [MessageConsumer]
JMSX_GROUPID	JMSXGroupID	System.String	Nastavte vlastnost řetězce [PropertyContext]
JMSX_GROUPSEQ	JMSXGroupSeq	System.Int32	Nastavit celočíselné vlastnosti [PropertyContext]
ID UŽIVATELE JMSX_ID	JMSXUserID	System.String	Odeslat [MessageProducer]

IBMdefinované vlastnosti zprávy

Několik IBMdefinovaných vlastností zprávy je podporováno produkty XMS a WebSphere JMS.

Příkaz Tabulka 20 na stránce 70 uvádí IBM definované vlastnosti zprávy, které jsou podporovány jak XMS , tak WebSphere JMS. Další informace o definovaných vlastnostech produktu IBMnaleznete v dokumentaci produktu IBM MQ nebo v dokumentaci produktu WebSphere Application Server .

Tabulka uvádí datový typ každé vlastnosti a uvádí, jak je hodnota vlastnosti nastavena pro odeslanou zprávu. Některé vlastnosti jsou nastaveny automaticky produktem XMS , když aplikace odesílá zprávu.

Tabulka 20. Vlastnosti definované uživatelem IBMv rámci zprávy

Název XMS definované vlastnosti IBM	Název produktu WebSphere JMS	Datový typ	Jak je hodnota nastavena pro přenesená zpráva (ve formátu metoda [třída])
JMS_IBM_CHARACTER_SET	JMS_IBM_Character_Set.	System.Int32	Nastavit celočíselné vlastnosti [PropertyContext]
JMS_IBM_Encoding	JMS_IBM_Encoding	System.Int32	Nastavit celočíselné vlastnosti [PropertyContext]
JMS_IBM_EXCEPTIONMESSAGE	JMS_IBM_ExceptionMessage .	System.String	Přijmout [MessageConsumer]
JMS_IBM_EXCEPTIONREASON	JMS_IBM_ExceptionReason	System.Int32	Přijmout [MessageConsumer]
JMS_IBM_EXCEPTIONTIMESTAMP	JMS_IBM_ExceptionTimestamp	System.Int64	Přijmout [MessageConsumer]
JMS_IBM_EXCEPTIONPROBLEMDESTINATION	Cíl JMS_IBM_ExceptionProblem	System.String	Přijmout [MessageConsumer]

Tabulka 20. Vlastnosti definované uživatelem IBMv rámci zprávy (pokračování)

Název XMS definované vlastnosti IBM	Název produktu WebSphere JMS	Datový typ	Jak je hodnota nastavena pro přenesená zpráva (ve formátu metoda [třída])
JMS_IBM_FEEDBACK	JMS_IBM_Feedback.	System.Int32	Nastavit celočíselné vlastnosti [PropertyContext]
FORMÁT JMS_IBM_FORMAT	JMS_IBM_Format.	System.String	Nastavte vlastnost řetězce [PropertyContext]
JMS_IBM_LAST_MSG_IN_GROUP	JMS_IBM_Last_Msg_In_Group.	System.Boolean	Nastavit celočíselné vlastnosti [PropertyContext]
JMS_IBM_MSGTYPE	JMS_IBM_MsgType .	System.Int32	Nastavit celočíselné vlastnosti [PropertyContext]
JMS_IBM_PUTAPPLTYPE	Typ JMS_IBM_PutAppl	System.Int32	Odeslat [MessageProducer]
JMS_IBM_PUTDATE	JMS_IBM_PutDate .	System.String	Odeslat [MessageProducer]
ČAS SOUBORU JMS_IBM_PUTTIME	JMS_IBM_PutTime .	System.String	Odeslat [MessageProducer]
JMS_IBM_REPORT_COA	JMS_IBM_Report_COA	System.Int32	Nastavit celočíselné vlastnosti [PropertyContext]
JMS_IBM_REPORT_COD	JMS_IBM_Report_COD.	System.Int32	Nastavit celočíselné vlastnosti [PropertyContext]
SKÓDOVÁ_ZPRÁVA JMS_IBM_REPORT_DISCARD_MSG	JMS_IBM_Report_Discard_Msg.	System.Int32	Nastavit celočíselné vlastnosti [PropertyContext]
VÝJIMKA JMS_IBM_REPORT_EXCEPTION	Výjimka JMS_IBM_Report_Exception	System.Int32	Nastavit celočíselné vlastnosti [PropertyContext]
JMS_IBM_REPORT_EXPIRATION	JMS_IBM_Report_Expiration.	System.Int32	Nastavit celočíselné vlastnosti [PropertyContext]
JMS_IBM_REPORT_NAN	JMS_IBM_Report_NAN.	System.Int32	Nastavit celočíselné vlastnosti [PropertyContext]
JMS_IBM_REPORT_PAN	JMS_IBM_Report_PAN.	System.Int32	Nastavit celočíselné vlastnosti [PropertyContext]
JMS_IBM_REPORT_PASS_CORREL_ID	JMS_IBM_Report_Pass_Correl_ID.	System.Int32	Nastavit celočíselné vlastnosti [PropertyContext]

Tabulka 20. Vlastnosti definované uživatelem IBMv rámci zprávy (pokračování)

Název XMS definované vlastnosti IBM	Název produktu WebSphere JMS	Datový typ	Jak je hodnota nastavena pro přenesená zpráva (ve formátu metoda [třída])
JMS_IBM_REPORT_PASS_MSG_ID	JMS_ID_zprávy_IBM_Report_ID	System.Int32	Nastavit celočíselné vlastnosti [PropertyContext]
JMS_IBM_SYSTEM_MESS_AGEID	JMS_IBM_System_MessageID	System.String	Odeslat [MessageProducer]

Vlastnosti zprávy definované uživatelem

Aplikace produktu XMS může vytvořit a používat vlastní sadu vlastností zprávy. Když aplikace odešle zprávu, tyto vlastnosti jsou také přeneseny spolu se zprávou. Přijímající aplikace používající selektory zpráv může poté vybrat zprávy, které chce přijímat, na základě hodnot těchto vlastností.

Chcete-li povolit aplikaci WebSphere JMS vybrat a zpracovat zprávy odeslané aplikací XMS, název vlastnosti definované aplikací musí odpovídat pravidlům pro vytváření identifikátorů ve výrazech selektorů zpráv, jak je dokumentováno v dokumentaci produktu IBM MQ. Hodnota vlastnosti definované aplikací musí mít jeden z následujících datových typů: System.Boolean, System.SByte, System.Int16, System.Int32, System.Int64, System.Float, System.Double, nebo System.String.

Tělo zprávy produktu XMS

Tělo zprávy obsahuje data aplikace. Zpráva však může mít žádný text a skládá se pouze z polí záhlaví a vlastností.

Produkt XMS podporuje pět typů těla zprávy:

Bajtů

Tělo obsahuje proud bajtů. Zpráva s tímto typem těla se nazývá *bajtová zpráva*. Rozhraní `IBytesMessage` obsahuje metody pro zpracování textu bajtové zprávy. Další informace viz téma [“Bajtové zprávy”](#) na stránce 74.

Mapa

Tělo obsahuje sadu dvojic název-hodnota, přičemž každá hodnota má přidružený datový typ. Zpráva s tímto typem těla se nazývá *mapová zpráva*. Rozhraní `IMapMessage` obsahuje metody pro zpracování textu zprávy mapy. Další informace viz téma [“Mapové zprávy”](#) na stránce 74.

Objekt

Tělo obsahuje serializovaný objekt Java nebo .NET. Zpráva s tímto typem těla se nazývá *zpráva objektu*. Rozhraní `IObjectMessage` obsahuje metody pro zpracování textu zprávy objektu. Další informace viz téma [“Zprávy objektu”](#) na stránce 75.

Proud

Tělo obsahuje proud hodnot, kde má každá hodnota přidružený datový typ. Zpráva s tímto typem těla se nazývá *proudová zpráva*. Rozhraní `IStreamMessage` obsahuje metody pro zpracování textu zprávy proudu. Další informace viz téma [“Zprávy proudů”](#) na stránce 76.

Text

Tělo obsahuje řetězec. Zpráva s tímto typem těla se nazývá *textová zpráva*. Rozhraní `ITextMessage` obsahuje metody pro zpracování textu zprávy. Další informace viz téma [“Textové zprávy”](#) na stránce 77.

Rozhraní `IMessage` je nadřazeným prvkem všech objektů zpráv a lze jej použít ve funkcích systému zpráv ke znázornění libovolného z typů zpráv produktu XMS.

Další informace o velikosti a maximální a minimální hodnotě každého z těchto datových typů viz [Tabulka 5](#) na stránce 38.

Související odkazy

Části zprávy produktu XMS

Zpráva XMS se skládá ze záhlaví, sady vlastností a těla.

Pole záhlaví ve zprávě XMS

Chcete-li povolit aplikaci XMS výměnu zpráv s aplikací WebSphere JMS , záhlaví zprávy XMS obsahuje pole záhlaví zprávy JMS .

Vlastnosti zprávy XMS

Produkt XMS podporuje tři druhy vlastností zprávy: JMS definované vlastnosti, IBM definované vlastnosti a vlastnosti definované aplikací.

Voliče zpráv

Aplikace XMS používá selektory zpráv k výběru zpráv, které chce přijímat.

Mapování zpráv produktu XMS na zprávy produktu IBM MQ

Pole záhlaví JMS a vlastnosti zprávy produktu XMS jsou mapovány na pole ve strukturách záhlaví zprávy IBM MQ .

Datové typy pro prvky dat aplikace

Aby se zajistilo, že aplikace XMS si může vyměňovat zprávy s aplikací IBM MQ classes for JMS , musí být obě aplikace schopny interpretovat data aplikace v těle zprávy stejným způsobem.

Z tohoto důvodu musí mít každý prvek dat aplikace zapsaný v těle zprávy pomocí aplikace XMS jeden z datových typů uvedených v Tabulka 21 na stránce 73. Pro každý datový typ tabulka zobrazuje kompatibilní datový typ Java . XMS poskytuje metody pro zápis prvků dat aplikací pouze s těmito datovými typy.

XMS Datový typ	Představuje	Kompatibilní datový typ Java
System.Boolean	Logická hodnota true nebo false	typ boolean
System.Char16	dvoubajtový znak	ZNAK
System.SByte	Podepsané 8bitové celé číslo	bajt
System.Int16	Signováno 16bitové celé číslo	short
System.Int32	Podepsané 32bitové celé číslo	celé číslo
System.Int64	Signed 64-bit integer	long
System.Float	Číslo s plovoucí řádovou čárkou	float
System.Double	Číslo s plovoucí řádovou čárkou a dvojitou přesností	dvojitý
System.String	Řetězec znaků	Řetězec

Informace o velikosti, maximální hodnotě a minimální hodnotě každého z těchto datových typů viz “XMS primitivní typy” na stránce 37.

Související pojmy

Atributy a vlastnosti objektů

Objekt XMS může mít atributy a vlastnosti, které jsou charakteristikami objektu, které jsou implementovány různými způsoby.

XMS primitivní typy

XMS poskytuje ekvivalenty osmi primitivních typů Java (byte, short, int, long, float, double, char a boolean). To umožňuje výměnu zpráv mezi XMS a JMS , aniž by došlo ke ztrátě nebo poškození dat.

Implicitní převod hodnoty vlastnosti z jednoho datového typu na jiný.

Když aplikace získá hodnotu vlastnosti, může ji převést hodnota XMS na jiný datový typ. Mnoho pravidel určuje, které konverze jsou podporovány a jak produkt XMS provádí převody.

Související odkazy

Bajtové zprávy

Tělo bajtové zprávy obsahuje proud bajtů. Tělo obsahuje pouze aktuální data a je odpovědností odesílajícího a přijímacího programu interpretovat tato data.

Mapové zprávy

Tělo zprávy mapy obsahuje sadu dvojic název-hodnota, kde má každá hodnota přidružený datový typ.

Zprávy objektu

Tělo zprávy objektu obsahuje serializovaný objektJava nebo .NET .

Zprávy proudů

Tělo zprávy s proudem hodnot obsahuje proud hodnot, kde má každá hodnota přidružený datový typ.

Textové zprávy

Tělo textové zprávy obsahuje řetězec.

Bajtové zprávy

Tělo bajtové zprávy obsahuje proud bajtů. Tělo obsahuje pouze aktuální data a je odpovědností odesílajícího a přijímacího programu interpretovat tato data.

Bajtové zprávy jsou užitečné v případě, že aplikace XMS potřebuje vyměňovat zprávy s aplikacemi, které nepoužívají rozhraní API XMS nebo JMS .

Poté, co aplikace vytvoří bajtovou zprávu, tělo zprávy bude pouze pro zápis. Aplikace sestaví aplikační data do těla voláním vhodných metod zápisu pro rozhraní IMessage pro produkt .NET. Pokaždé, když aplikace zapíše hodnotu do bajtového proudu zpráv, hodnota se sestaví okamžitě za předchozí hodnotou napsanou aplikací. Produkt XMS udržuje interní kurzor, aby si zapamatoval pozici posledního bajtu, který byl sestaven.

Když aplikace odešle zprávu, tělo zprávy se stane jen pro čtení. V tomto režimu může aplikace odeslat zprávu opakovaně.

Když aplikace přijme bajtovou zprávu, tělo zprávy je určeno pouze pro čtení. Aplikace může použít příslušné metody čtení rozhraní IMessage ke čtení obsahu proudu bajtů zpráv. Aplikace čte bajty v pořadí a XMS udržuje interní kurzor, aby si pamatoval pozici posledního bajtu, který byl přečten.

Pokud aplikace volá metodu Reset rozhraní IMessage , když je tělo bajtové zprávy zapisovatelné, tělo se stane jen pro čtení. Metoda také znovu umístí kurzor na začátek bajtového proudu zpráv.

Pokud aplikace volá metodu ClearBody rozhraní IMessage pro .NET , když je tělo bajtové zprávy jen pro čtení, tělo bude zapisovatelné. Tato metoda také vymaže tělo.

Související odkazy

Datové typy pro prvky dat aplikace

Aby se zajistilo, že aplikace XMS si může vyměňovat zprávy s aplikací IBM MQ classes for JMS , musí být obě aplikace schopny interpretovat data aplikace v těle zprávy stejným způsobem.

Mapové zprávy

Tělo zprávy mapy obsahuje sadu dvojic název-hodnota, kde má každá hodnota přidružený datový typ.

Zprávy objektu

Tělo zprávy objektu obsahuje serializovaný objektJava nebo .NET .

Zprávy proudů

Tělo zprávy s proudem hodnot obsahuje proud hodnot, kde má každá hodnota přidružený datový typ.

Textové zprávy

Tělo textové zprávy obsahuje řetězec.

IMessage (pro rozhraní .NET)

Bajtová zpráva je zpráva, jejíž tělo tvoří proud bajtů.

Mapové zprávy

Tělo zprávy mapy obsahuje sadu dvojic název-hodnota, kde má každá hodnota přidružený datový typ.

V každé dvojici název-hodnota je název řetězec, který identifikuje hodnotu, a hodnota je prvek dat aplikace, který má jeden z datových typů produktu XMS, které jsou uvedeny v seznamu [Tabulka 21 na stránce 73](#). Pořadí dvojic název-hodnota není definováno. Třída `MapMessage` obsahuje metody pro nastavení a získání dvojic název-hodnota.

Aplikace může přistupovat k páru název-hodnota náhodně zadáním jeho názvu.

Aplikace `.NET` může použít vlastnost `MapNames` k získání výčtu názvů v těle zprávy mapování.

Když aplikace získá hodnotu páru název-hodnota, lze ji převést pomocí XMS na jiný datový typ. Chcete-li například získat celé číslo z těla zprávy mapy, aplikace může volat metodu `GetString` třídy `MapMessage`, která vrací celé číslo jako řetězec. Podporované konverze jsou stejné jako ty, které jsou podporovány, když XMS převádí hodnotu vlastnosti z jednoho datového typu na jiný. Další informace o podporovaných převodech viz [“Implicitní převod hodnoty vlastnosti z jednoho datového typu na jiný.” na stránce 38](#).

Poté, co aplikace vytvoří zprávu mapy, tělo zprávy lze číst a zapisovat do něj. Tělo zůstane po odeslání zprávy čitelné a přístupné pro zápis. Když aplikace přijme mapovou zprávu, tělo zprávy je jen pro čtení. Pokud aplikace volá metodu `Vymazat` tělo třídy zpráv, když je tělo zprávy mapy jen pro čtení, tělo bude čitelné a zapisovatelné. Tato metoda také vymaže tělo.

Související pojmy

[Implicitní převod hodnoty vlastnosti z jednoho datového typu na jiný.](#)

Když aplikace získá hodnotu vlastnosti, může ji převést hodnota XMS na jiný datový typ. Mnoho pravidel určuje, které konverze jsou podporovány a jak produkt XMS provádí převody.

Související odkazy

[Datové typy pro prvky dat aplikace](#)

Aby se zajistilo, že aplikace XMS si může vyměňovat zprávy s aplikací `IBM MQ classes for JMS`, musí být obě aplikace schopny interpretovat data aplikace v těle zprávy stejným způsobem.

[Bajtové zprávy](#)

Tělo bajtové zprávy obsahuje proud bajtů. Tělo obsahuje pouze aktuální data a je odpovědností odesílajícího a přijímajícího programu interpretovat tato data.

[Zprávy objektu](#)

Tělo zprávy objektu obsahuje serializovaný objekt `Java` nebo `.NET`.

[Zprávy proudů](#)

Tělo zprávy s proudem hodnot obsahuje proud hodnot, kde má každá hodnota přidružený datový typ.

[Textové zprávy](#)

Tělo textové zprávy obsahuje řetězec.

[IMapMessage \(pro rozhraní .NET\)](#)

Mapová zpráva je zpráva, jejíž tělo tvoří sadu dvojic název-hodnota, kde má každá hodnota přidružený datový typ.

Zprávy objektu

Tělo zprávy objektu obsahuje serializovaný objekt `Java` nebo `.NET`.

Aplikace produktu XMS může přijmout zprávu objektu, změnit její pole záhlaví a vlastnosti a poté ji odeslat do jiného cíle. Aplikace může také zkopírovat tělo zprávy objektu a použít ji k vytvoření další zprávy objektu. Produkt XMS zachází s tělem zprávy objektu jako s polem bajtů.

Poté, co aplikace vytvoří zprávu objektu, je tělo zprávy přístupné pro čtení a zápis. Tělo zůstane po odeslání zprávy čitelné a přístupné pro zápis. Když aplikace obdrží zprávu objektu, tělo zprávy je jen pro čtení. Pokud aplikace volá metodu `Clear` Body rozhraní `IMessage` pro `.NET`, je-li tělo zprávy objektu jen pro čtení, tělo bude čitelné a schopné zápisu. Tato metoda také vymaže tělo.

Související odkazy

[Datové typy pro prvky dat aplikace](#)

Aby se zajistilo, že aplikace XMS si může vyměňovat zprávy s aplikací `IBM MQ classes for JMS`, musí být obě aplikace schopny interpretovat data aplikace v těle zprávy stejným způsobem.

[Bajtové zprávy](#)

Tělo bajtové zprávy obsahuje proud bajtů. Tělo obsahuje pouze aktuální data a je odpovědností odesílajícího a přijímajícího programu interpretovat tato data.

Mapové zprávy

Tělo zprávy mapy obsahuje sadu dvojic název-hodnota, kde má každá hodnota přidružený datový typ.

Zprávy proudů

Tělo zprávy s proudem hodnot obsahuje proud hodnot, kde má každá hodnota přidružený datový typ.

Textové zprávy

Tělo textové zprávy obsahuje řetězec.

IObjectMessage (pro rozhraní .NET)

Zpráva objektu je zpráva, jejíž tělo tvoří serializovaný objekt Java nebo .NET.

Zprávy proudů

Tělo zprávy s proudem hodnot obsahuje proud hodnot, kde má každá hodnota přidružený datový typ.

Datový typ hodnoty je jeden z datových typů produktu XMS , které jsou uvedeny v části [Tabulka 21](#) na stránce 73.

Poté, co aplikace vytvoří zprávu s proudem hodnot, je tělo zprávy schopné zápisu. Aplikace sestaví aplikační data do těla voláním vhodných metod zápisu pro rozhraní `IStreamMessage` pro produkt .NET. Pokaždé, když aplikace zapíše hodnotu do proudu zpráv, sestaví se hodnota a její datový typ bezprostředně za předchozí hodnotou napsanou aplikací. Produkt XMS udržuje interní kurzor, aby si zapamatoval pozici poslední hodnoty, která byla sestavena.

Když aplikace odešle zprávu, tělo zprávy se stane jen pro čtení. V tomto režimu může aplikace odeslat zprávu vícekrát.

Když aplikace přijme proudovou zprávu, tělo zprávy je určeno pouze pro čtení. Aplikace může pro čtení obsahu proudu zpráv použít vhodné metody čtení rozhraní `IStreamMessage` pro produkt .NET . Aplikace čte hodnoty v posloupnosti a XMS udržuje interní kurzor, aby si zapamatoval pozici poslední hodnoty, která byla přečtena.

Když aplikace načte hodnotu z proudu zpráv, může ji převést hodnota XMS do jiného datového typu. Chcete-li například číst celé číslo z proudu zpráv, může aplikace volat metodu `ReadString` , která vrací celé číslo jako řetězec. Podporované konverze jsou stejné jako ty, které jsou podporovány, když XMS převádí hodnotu vlastnosti z jednoho datového typu na jiný. Další informace o podporovaných převodech viz [“Implicitní převod hodnoty vlastnosti z jednoho datového typu na jiný.”](#) na stránce 38.

Dojde-li k chybě při pokusu aplikace o načtení hodnoty z proudu zpráv, kurzor není pokročilý. Aplikace se může zotavit z chyby tím, že se pokusí načíst hodnotu jako jiný datový typ.

Pokud aplikace volá metodu `Reset` rozhraní `IStreamMessage` pro XMS , když je tělo zprávy typu `message-only`, tělo bude jen pro čtení. Metoda také znovu umístí kurzor na začátek toku zpráv.

Pokud aplikace volá metodu `ClearBody` rozhraní `IMessage` pro XMS , když tělo zprávy proudu je jen pro čtení, tělo se stane pouze pro zápis. Tato metoda také vymaže tělo.

Související pojmy

[Implicitní převod hodnoty vlastnosti z jednoho datového typu na jiný.](#)

Když aplikace získá hodnotu vlastnosti, může ji převést hodnota XMS na jiný datový typ. Mnoho pravidel určuje, které konverze jsou podporovány a jak produkt XMS provádí převody.

Související odkazy

[Datové typy pro prvky dat aplikace](#)

Aby se zajistilo, že aplikace XMS si může vyměňovat zprávy s aplikací IBM MQ classes for JMS , musí být obě aplikace schopny interpretovat data aplikace v těle zprávy stejným způsobem.

Bajtové zprávy

Tělo bajtové zprávy obsahuje proud bajtů. Tělo obsahuje pouze aktuální data a je odpovědností odesílajícího a přijímajícího programu interpretovat tato data.

Mapové zprávy

Tělo zprávy mapy obsahuje sadu dvojic název-hodnota, kde má každá hodnota přidružený datový typ.

Zprávy objektu

Tělo zprávy objektu obsahuje serializovaný objekt Java nebo .NET .

Textové zprávy

Tělo textové zprávy obsahuje řetězec.

IStreamMessage (pro rozhraní .NET)

Proudová zpráva je zpráva, jejíž tělo tvoří proud hodnot, kde má každá hodnota přidružený datový typ. Obsah těla se zapisuje a čte postupně.

Textové zprávy

Tělo textové zprávy obsahuje řetězec.

Poté, co aplikace vytvoří textovou zprávu, tělo zprávy lze číst a zapisovat do něj. Tělo zůstane po odeslání zprávy čitelné a přístupné pro zápis. Když aplikace přijme textovou zprávu, tělo zprávy je určeno pouze pro čtení. Pokud aplikace volá metodu Vyčistit tělo rozhraní IMessage pro .NET , když je tělo textové zprávy určeno pouze pro čtení, tělo bude čitelné a schopné zápisu. Tato metoda také vymaže tělo.

Související odkazy

Datové typy pro prvky dat aplikace

Aby se zajistilo, že aplikace XMS si může vyměňovat zprávy s aplikací IBM MQ classes for JMS , musí být obě aplikace schopny interpretovat data aplikace v těle zprávy stejným způsobem.

Bajtové zprávy

Tělo bajtové zprávy obsahuje proud bajtů. Tělo obsahuje pouze aktuální data a je odpovědností odesílajícího a přijímajícího programu interpretovat tato data.

Mapové zprávy

Tělo zprávy mapy obsahuje sadu dvojic název-hodnota, kde má každá hodnota přidružený datový typ.

Zprávy objektu

Tělo zprávy objektu obsahuje serializovaný objekt Java nebo .NET .

Zprávy proudů

Tělo zprávy s proudem hodnot obsahuje proud hodnot, kde má každá hodnota přidružený datový typ.

ITextMessage (pro rozhraní .NET)

Textová zpráva je zpráva, jejíž tělo tvoří řetězec.

Voliče zpráv

Aplikace XMS používá selektory zpráv k výběru zpráv, které chce přijímat.

Když aplikace vytvoří spotřebitele zpráv, může přidružovat výraz selektoru zpráv k odběrateli. Výraz výběru zprávy určuje kritéria výběru.

Když se aplikace připojuje ke správci front produktu IBM WebSphere MQ 7.0 , je výběr zprávy proveden na straně správce front. Produkt XMS neprovede žádný výběr a jednoduše doručí zprávu, kterou obdrží od správce front, a poskytuje tak lepší výkon.

Aplikace může vytvořit více než jednoho spotřebitele zpráv, každý se svým vlastním výrazem selektoru zpráv. Pokud přichází zpráva splňuje kritéria výběru více než jednoho spotřebitele zpráv, produkt XMS doručí zprávu každému z těchto spotřebitelů.

Výraz selektoru zpráv může odkazovat na následující vlastnosti zprávy:

- Vlastnosti definované službou JMS
- Vlastnosti definované uživatelem IBM
- Vlastnosti definované aplikací

Může také odkazovat na následující pole záhlaví zprávy:

- JMSCorrelationID
- JMSDeliveryMode
- JMSMessageID

- JMSPriority.
- JMSTimestamp
- JMSType.

Výraz selektoru zpráv však nemůže odkazovat na data v těle zprávy.

Zde je příklad výrazu selektoru zpráv:

```
JMSPriority > 3 AND manufacturer = 'Jaguar' AND model in ('xj6','xj12')
```

Produkt XMS doručuje zprávu spotřebiteli zpráv s tímto výrazem selektoru zpráv pouze v případě, že má zpráva prioritu větší než 3; vlastnost definované aplikací, výrobce, s hodnotou Jaguar; a jinou definovanou aplikací, modelem, s hodnotou xj6 nebo xj12.

Pravidla syntaxe pro vytvoření výrazu selektoru zpráv v produktu XMS jsou stejná jako pravidla uvedená v části IBM MQ classes for JMS. Informace o tom, jak sestavit výraz selektoru zpráv, viz dokumentace produktu IBM MQ, že ve výrazu selektoru zpráv musí být názvy JMSdefinovaných vlastností názvy JMS a názvy definovaných vlastností IBMmusí být názvy IBM MQ classes for JMS. Názvy XMS nelze použít v výrazu selektoru zpráv.

Související odkazy

[Části zprávy produktu XMS](#)

Zpráva XMS se skládá ze záhlaví, sady vlastností a těla.

[Pole záhlaví ve zprávě XMS](#)

Chcete-li povolit aplikaci XMS výměnu zpráv s aplikací WebSphere JMS, záhlaví zprávy XMS obsahuje pole záhlaví zprávy JMS.

[Vlastnosti zprávy XMS](#)

Produkt XMS podporuje tři druhy vlastností zprávy: JMS definované vlastnosti, IBM definované vlastnosti a vlastnosti definované aplikací.

[Tělo zprávy produktu XMS](#)

Tělo zprávy obsahuje data aplikace. Zpráva však může mít žádný text a skládá se pouze z polí záhlaví a vlastností.

[Mapování zpráv produktu XMS na zprávy produktu IBM MQ](#)

Pole záhlaví JMS a vlastnosti zprávy produktu XMS jsou mapovány na pole ve strukturách záhlaví zprávy IBM MQ.

Mapování zpráv produktu XMS na zprávy produktu IBM MQ

Pole záhlaví JMS a vlastnosti zprávy produktu XMS jsou mapovány na pole ve strukturách záhlaví zprávy IBM MQ.

Je-li aplikace XMS připojena ke správci front produktu IBM MQ, jsou zprávy odeslané do správce front mapovány do zpráv produktu IBM MQ stejným způsobem, jakým jsou zprávy produktu IBM MQ classes for JMS mapovány na zprávy produktu IBM MQ za podobných okolností.

Je-li vlastnost XMSCS_WMQ_TARGET_CLIENT objektu Destination nastavena na hodnotu XMSCS_WMQ_TARGET_DEST_JMS, jsou pole záhlaví JMS a vlastnosti zprávy odeslané do cíle mapovány na pole ve struktuře záhlaví MQMD a MQRFH2 zprávy produktu IBM MQ. Nastavení vlastnosti XMSCS_WMQ_TARGET_CLIENT tímto způsobem předpokládá, že aplikace přijímající zprávu může zpracovat záhlaví MQRFH2. Přijímající aplikace může být tedy další aplikací produktu XMS, aplikací produktu IBM MQ classes for JMS nebo nativní aplikací produktu IBM MQ, která byla navržena pro zpracování záhlaví MQRFH2.

Je-li vlastnost XMSCS_WMQ_TARGET_CLIENT objektu Destination nastavena na hodnotu XMSCS_WMQ_TARGET_DEST_MQ, jsou pole záhlaví JMS a vlastnosti zprávy odeslané do cíle mapovány na pole ve struktuře záhlaví MQMD ve zprávě produktu IBM MQ. Zpráva neobsahuje záhlaví MQRFH2 a všechna pole záhlaví JMS a vlastnosti, které nelze mapovat na pole ve struktuře záhlaví MQMD, budou

ignorovány. Aplikace, která přijme zprávu, může být proto nativním IBM MQ , který není navržen pro zpracování záhlaví MQRFH2 .

Zprávy produktu IBM MQ přijaté od správce front se mapují na zprávy produktu XMS stejným způsobem, že se zprávy produktu IBM MQ mapují na zprávy produktu IBM MQ classes for JMS za podobných okolností.

Má-li přichodící zpráva produktu IBM MQ záhlaví MQRFH2 , má výsledná zpráva XMS tělo, jehož typ je určen hodnotou vlastnosti **Msd** obsažené ve složce mcd záhlaví MQRFH2 . Není-li vlastnost **Msd** přítomna v záhlaví MQRFH2 , nebo pokud zpráva IBM MQ neobsahuje záhlaví MQRFH2 , má výsledná zpráva XMS tělo, jehož typ je určen hodnotou pole *Format* v záhlaví MQMD. Je-li pole *Format* nastaveno na hodnotu MQFMT_STRING, bude zpráva XMS textovou zprávou. Jinak se jedná o bajtovou zprávu ve zprávě XMS . Pokud zpráva IBM MQ neobsahuje žádné záhlaví MQRFH2 , jsou nastavena pouze tato pole záhlaví JMS a vlastnosti, které lze odvodit z polí v záhlaví MQMD.

Další informace o mapování zpráv produktu IBM MQ classes for JMS na zprávy produktu IBM MQ naleznete v dokumentaci produktu IBM MQ .

Související odkazy

Části zprávy produktu XMS

Zpráva XMS se skládá ze záhlaví, sady vlastností a těla.

Pole záhlaví ve zprávě XMS

Chcete-li povolit aplikaci XMS výměnu zpráv s aplikací WebSphere JMS , záhlaví zprávy XMS obsahuje pole záhlaví zprávy JMS .

Vlastnosti zprávy XMS

Produkt XMS podporuje tři druhy vlastností zprávy: JMS definované vlastnosti, IBM definované vlastnosti a vlastnosti definované aplikací.

Tělo zprávy produktu XMS

Tělo zprávy obsahuje data aplikace. Zpráva však může mít žádný text a skládá se pouze z polí záhlaví a vlastností.

Voliče zpráv

Aplikace XMS používá selektory zpráv k výběru zpráv, které chce přijímat.

Čtení a zápis deskriptoru zpráv z aplikace IBM Message Service Client for .NET

Můžete mít přístup ke všem polím deskriptoru zpráv (MQMD) zprávy IBM MQ s výjimkou StrucId a Verze; BackoutCount lze číst, ale ne zapisovat. Tato funkce je k dispozici pouze při připojování k produktu IBM WebSphere MQ 6.0 nebo novějším správci front a je řízena vlastnostmi místa určení popsaným později.

Atributy zpráv poskytované aplikací IBM Message Service Client for .NET usnadňují aplikacím XMS nastavit pole MQMD a také řídit aplikaci IBM WebSphere MQ .

Některá omezení se používají při používání systému zpráv publikování/odběru. Například pole MQMD jako MsgID a CorrelId, jsou-li nastavena, jsou ignorována.

Funkce popsaná v tomto tématu není k dispozici pro systém zpráv publikování/odběru, když se připojujete ke správci front produktu IBM WebSphere MQ 6.0 . Je také nedostupný, když je vlastnost **PROVIDERVERSION** nastavena na 6.

Přístup k datům zprávy produktu IBM MQ z aplikace produktu IBM Message Service Client for .NET

K kompletním datům zprávy produktu IBM MQ můžete přistupovat včetně záhlaví MQRFH2 (je-li přítomno) a všech dalších záhlaví IBM MQ (jsou-li k dispozici) v aplikaci IBM Message Service Client for .NET jako tělo zprávy JMSBytesMessage.

Funkce popsaná v tomto tématu je k dispozici pouze při připojování k správci front produktu IBM WebSphere MQ 7.0 nebo novějším a poskytovatel systému zpráv produktu WebSphere MQ je v normálním režimu.

Vlastnosti objektu místa určení určují, jak aplikace XMS přistupuje k celé zprávě produktu IBM MQ (včetně záhlaví MQRFH2 , je-li přítomno) jako těla zprávy JMSBytesMessage.

Odstraňování problémů

Tato část obsahuje informace, které vám pomohou zjistit a řešit problémy při používání produktu IBM Message Service Client for .NET.

Tento oddíl obsahuje následující témata:

- [“Konfigurace trasování pro aplikace produktu .NET” na stránce 80](#)
- [“Konfigurace FFDC pro aplikace .NET” na stránce 84](#)
- [“Tipy pro odstraňování problémů” na stránce 84](#)

Konfigurace trasování pro aplikace produktu .NET

Pro aplikace produktu XMS .NET můžete nakonfigurovat trasování z konfiguračního souboru aplikace stejně jako z proměnných prostředí XMS . Můžete vybrat komponenty, které chcete trasovat. Trasování se obvykle používá pod vedením podpory produktu IBM .

Trasování pro XMS .NET je založeno na standardní infrastruktuře trasování .NET .

Všechny trasování kromě trasování chyb jsou standardně zakázány. Můžete zapnout trasování a konfigurovat nastavení trasování jedním z následujících způsobů:

- Použitím konfiguračního souboru aplikace s názvem, který se skládá z názvu spustitelného programu, do kterého se soubor vztahuje, s příponou `.config`. Například konfigurační soubor aplikace pro `text.exe` by měl mít název `text.exe.config`. Použití konfiguračního souboru aplikace je upřednostňovaným způsobem, jak povolit trasování pro aplikace produktu XMS .NET . Další podrobnosti viz [“Konfigurace trasování s použitím konfiguračního souboru aplikace” na stránce 81](#).
- Při použití proměnných prostředí produktu XMS pro aplikace XMS C nebo C + + . Další podrobnosti viz [“Konfigurace trasování pomocí proměnných prostředí XMS” na stránce 83](#).

Aktivní trasovací soubor má název ve formátu `xms_tracePID.log`, kde *PID* představuje ID procesu aplikace. Velikost aktivního trasovacího souboru je standardně omezena na 20 MB. Je-li tento limit dosažen, soubor se přejmenuje a archivuje. Archivní soubory mají názvy ve formátu `xms_tracePID_YY.MM.DD_HH.MM.SS.log`

Standardně je počet souborů trasování, které se uchovávají, čtyři, tj. jeden aktivní soubor a tři archivované soubory. Tyto čtyři soubory se používají jako posuvná vyrovnávací paměť, dokud se aplikace nezastaví, přičemž nejstarší soubor bude odstraněn a nahrazen nejnovějším souborem. Počet trasovacích souborů můžete změnit zadáním jiného čísla v konfiguračním souboru aplikace. Musí však existovat alespoň dva soubory (jeden aktivní soubor a jeden archivovaný soubor).

K dispozici jsou dva formáty souborů trasování:

- Soubory trasování základního formátu jsou čitelné pro člověka ve formátu WebSphere Application Server . Tento formát je výchozím formátem trasovacího souboru. Základní formát není kompatibilní s nástroji pro analýzu trasování.
- Rozšířené trasovací soubory formátu jsou kompatibilní s nástroji pro analýzu trasování. Musíte uvést, že chcete vytvořit trasovací soubory v rozšířeném formátu v konfiguračním souboru aplikace.

Položky trasování obsahují následující informace:

- Datum a čas, kdy bylo trasování zaprotokolováno
- Název třídy
- Typ trasování
- Trasovací zpráva

Následující příklad ukazuje výpis z nějakého trasování:

```
[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

V předchozím příkladu je formát následující:

[Date Time:Microsecs] or Exit	Thread-id	Classname	Trace-type	Methodname	Entry
----------------------------------	-----------	-----------	------------	------------	-------

kde Trace-type je:

- > pro položku
- < pro ukončení
- d pro ladicí informace

Konfigurace trasování s použitím konfiguračního souboru aplikace

Preferovaný způsob konfigurace trasování pro aplikace produktu XMS .NET je s konfiguračním souborem aplikace. Sekce trasování tohoto souboru zahrnuje parametry, které definují umístění, které má být trasováno, umístění trasovacího souboru a maximální povolenou velikost, počet použitých trasovacích souborů a formát trasovacího souboru.

Chcete-li zapnout trasování pomocí konfiguračního souboru aplikace, stačí umístit soubor do stejného adresáře jako spustitelný soubor pro vaši aplikaci.

Trasování může být povoleno jak komponentou, tak i typem trasování. Je také možné zapnout trasování pro celou trasovací skupinu. Můžete zapnout trasování pro komponenty v hierarchii buď jednotlivě, nebo hromadně. K dispozici jsou tyto typy trasování:

- Trasování ladění
- Trasování výjimky
- Varování, informační zprávy a chybové zprávy
- Záznam metody a trasování ukončení

Následující příklad ukazuje nastavení trasování definovaná v sekci Trasování konfiguračního souboru aplikace:

```
<?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>
```

Část [Tabulka 22 na stránce 82](#) popisuje nastavení parametrů podrobněji.

Tabulka 22. Nastavení parametrů trasování konfiguračního souboru aplikace

Parametr	Popis
<code>traceSpecification=ComponentName=type=state</code>	<p><i>ComponentName</i> je název třídy, kterou chcete trasovat. V tomto názvu můžete použít zástupný znak *. Například <code>*=all=enabled</code> určuje, že chcete trasovat všechny třídy, a <code>IBM.XMS.impl.*=all=enabled</code> uvádí, že požadujete pouze trasování rozhraní API.</p> <p><i>type</i> může být libovolný z následujících typů trasování:</p> <ul style="list-style-type: none"> • vše • ladění • událost • EntryExit <p>Produkt <i>state</i> může být povolen nebo zakázán.</p> <p>Můžete zadat více prvků trasování společně s použitím oddělovače ':' (dvojtečka).</p>
<code>traceFilePath="filename"</code>	<p>Pokud nezádáte cestu <code>traceFile</code>, nebo pokud je přítomna cesta <code>traceFile</code>, ale obsahuje prázdný řetězec, bude trasovací soubor umístěn do aktuálního adresáře. Chcete-li uložit trasovací soubor do uvedeného adresáře, zadejte do cesty <code>traceFile</code> název adresáře, například:</p> <pre>traceFilePath="c:\somepath"</pre>
<code>traceFileSize="size"</code>	<p>Maximální povolená velikost trasovacího souboru. Když soubor dosáhne této velikosti, je archivován a přejmenován. Výchozí maximum je 20 KB, které je určeno jako <code>traceFileSize="20000000"</code>.</p>
<code>traceFileNumber="number"</code>	<p>Počet trasovacích souborů, které mají být uchovány. Předvolba je 4 (jeden aktivní soubor a 3 archivní soubory). Minimální povolený počet je 2.</p>
<code>traceFormat="format"</code>	<p>Výchozí formát trasování je základní. Trasovací soubory jsou vytvářeny v tomto formátu, pokud zadáte <code>traceFormat="basic"</code>, nebo pokud nezádáte <code>traceFormat</code>, nebo pokud je <code>traceFormat</code> přítomen, ale obsahuje prázdný řetězec.</p> <p>Pokud vyžadujete trasování kompatibilní s nástroji pro analýzu trasování, musíte zadat <code>traceFormat="advanced"</code>.</p>

Nastavení trasování v konfiguračním souboru aplikace je dynamické a znovu se znovu načte při každém uložení nebo nahrazení souboru. Pokud jsou chyby nalezeny v souboru po jeho úpravě, vrátí se nastavení trasovacího souboru k jejich výchozím hodnotám.

Související pojmy

Konfigurace trasování pomocí proměnných prostředí XMS

Jako alternativu k použití konfiguračního souboru aplikace můžete zapnout trasování pomocí proměnných prostředí produktu XMS. Tyto proměnné prostředí se používají pouze v případě, že v konfiguračním souboru aplikace není žádná specifikace trasování.

Konfigurace trasování pomocí proměnných prostředí XMS

Jako alternativu k použití konfiguračního souboru aplikace můžete zapnout trasování pomocí proměnných prostředí produktu XMS . Tyto proměnné prostředí se používají pouze v případě, že v konfiguračním souboru aplikace není žádná specifikace trasování.

Chcete-li konfigurovat trasování pro aplikaci XMS .NET , nastavte následující proměnné prostředí před spuštěním aplikace:

<i>Tabulka 23. Nastavení proměnných prostředí pro trasování .NET</i>			
Proměnné prostředí	Výchozí	Nastavení	Význam
XMS_TRACE_ON	Nelze použít	Nepoužije se: hodnota této proměnné je ignorována.	Je-li nastavena hodnota XMS_TRACE_ON, je ve výchozím nastavení povoleno trasování.
CESTA_K_SOUBORU_XMS_TRASOVÁNÍ	Aktuální pracovní adresář	/dirpath/	Cesta k adresáři, do kterého jsou zapisovány trasování a záznamy FFDC. Produkt XMS vytvoří soubory FFDC a trasovací soubory v aktuálním pracovním adresáři, pokud nezádáte alternativní umístění. Alternativní umístění můžete určit nastavením proměnné prostředí XMS_TRACE_FILE_PATH na úplnou cestu k adresáři, do kterého má produkt XMS vytvořit soubory FFDC a trasovací soubory. Dříve než spustíte aplikaci, kterou chcete trasovat, musíte tuto proměnnou prostředí nastavit. Musíte se ujistit, že identifikátor uživatele, pod kterým aplikace běží, má oprávnění k zápisu do adresáře, kde XMS vytváří soubory FFDC a trasovací soubory.
FORMÁT XMS_TRACE_FORMAT	ZÁKLADNÍ	ZÁKLADNÍ, ROZŠÍŘENÉ	Uvádí požadovaný formát trasování, který může být buď BASIC, nebo ADVANCED. Výchozí formát je BASIC. Formát ADVANCED je kompatibilní s nástroji pro analýzu trasování.

Tabulka 23. Nastavení proměnných prostředí pro trasování .NET (pokračování)

Proměnné prostředí	Výchozí	Nastavení	Význam
SPECIFIKACE XMS_TRACE_SPECIFICAT ION	Nelze použít	Viz “Konfigurace trasování s použitím konfiguračního souboru aplikace” na stránce 81.	Přepíše specifikaci trasování, která odpovídá formátu určenému v parametru “Konfigurace trasování s použitím konfiguračního souboru aplikace” na stránce 81.

Související pojmy

Konfigurace trasování s použitím konfiguračního souboru aplikace

Preferovaný způsob konfigurace trasování pro aplikace produktu XMS .NET je s konfiguračním souborem aplikace. Sekce trasování tohoto souboru zahrnuje parametry, které definují umístění, které má být trasováno, umístění trasovacího souboru a maximální povolenou velikost, počet použitých trasovacích souborů a formát trasovacího souboru.

Konfigurace FFDC pro aplikace .NET

Pro implementaci produktu .NET v produktu XMS je pro každé FFDC vytvořen jeden soubor FFDC.

Soubory FFDC (First Failure Data Capture) jsou uloženy v čitelných textových souborech. Tyto soubory mají názvy ve tvaru `xmsffdcprocessID_DateTimestamp.txt`. Příklad názvu souboru je `xmsffdc264_2006.01.06T13.18.52.990955.txt`. Časové razítko obsahuje rozlišení mikrosekund.

Soubory začínají datem a časem, kdy došlo k výjimce, následovaná typem výjimky. Mezi tyto soubory patří jedinečné krátké pole `probeId`, které lze použít k vyhledání místa, kde došlo k tomuto FFDC.

Nemusíte provádět žádnou konfiguraci, abyste mohli zapnout funkci FFDC. Ve výchozím nastavení jsou všechny soubory FFDC zapsány do aktuálního adresáře. V případě potřeby však můžete zadat jiný adresář tak, že změníte `ffdcDirectory` v sekci Trasování konfiguračního souboru aplikace. V následujícím příkladu jsou všechny trasovací soubory protokolovány do adresáře `c:\client\ffdc`:

```
<IBM.XMS>
  <Trace ffdc=true ffdcDirectory="c:\client\ffdc"/>
</IBM.XMS>
```

Trasování lze zakázat nastavením hodnoty FFDC na hodnotu `false` v části Trasovací údaje konfiguračního souboru aplikace.

Pokud nepoužíváte konfigurační soubor aplikace, funkce FFDC je zapnuta a trasování je vypnuto.

Tipy pro odstraňování problémů

Tyto rady vám pomohou při odstraňování problémů s používáním produktu XMS.

Aplikace XMS se nemůže připojit ke správci front (MQRC_NOT_AUTHORIZED).

Klient produktu XMS .NET může mít odlišné chování od chování klienta IBM MQ JMS . Proto můžete zjistit, že se aplikace XMS nemůže připojit ke správci front, ačkoli vaše aplikace JMS může.

- Jednoduchým řešením tohoto problému je pokus o použití ID uživatele, které není delší než 12 znaků a je plně autorizováno v seznamu oprávnění správce front. Pokud toto řešení není ideální, bude pro použití uživatelských procedur zabezpečení použít jiný, ale složitější přístup. Potřebujete-li další náповědu k tomuto problému, obraťte se na podporu IBM .
- Nastavíte-li vlastnost `XMSCS_USERID` továrny připojení, musí odpovídat ID uživatele a heslu přihlášeného uživatele. Pokud tuto vlastnost nenastavíte, bude správce front používat při výchozím nastavení ID uživatele přihlášeného uživatele.

- Ověřování uživatele pro produkt IBM MQ se provádí pomocí podrobností o aktuálně přihlášeném uživateli a ne z informací poskytnutých v XMSC.USERID a XMSC.PASSWORD pole. Je navržen tak, aby udržoval konzistenci s produktem IBM MQ. Další informace o ověření naleznete v tématu *Ověřovací informace* v online dokumentaci produktu IBM MQ .

Připojení přesměřováno do stroje systému zpráv

Když se připojíte ke sběrnici pro integraci služeb produktu WebSphere Application Server 6.0.2 , mohou být všechna připojení přesměřována od původního koncového bodu poskytovatele ke stroji systému zpráv, který sběrnice zvolí pro dané připojení klienta. V takovém případě bude vždy přesměřovávat připojení k hostitelskému serveru určenému názvem hostitele spíše než adresou IP. Proto se můžete setkat s problémy s připojením, pokud nelze vyřešit název hostitele.

Chcete-li se úspěšně připojit ke sběrnici pro integraci služeb produktu WebSphere Application Server 6.0.2 , budete možná muset poskytnout mapování mezi názvy hostitelů a adresami IP na hostitelském počítači klienta. Můžete například určit mapování v lokální hostitelské tabulce na hostitelském počítači klienta.

Podpora pro ověření pomocí protokolu telnet

Protokol přenosu XMS .NET Real Time Transport podporuje pouze jednoduché ověřování pomocí protokolu telnet. Protokol přenosu XMS .NET Real Time Transport nepodporuje kvalitu ochrany.

Nastavení hodnot pro typ vlastnosti double

Na 64bitové platformě produktu Windows nemusí metody vlastností SetDoubleProperty () nebo GetDoubleProperty () pracovat správně při nastavení nebo získání hodnot pro typ vlastnosti double, pokud jsou hodnoty menší než hodnoty Double.Epsilon.

Například, pokud se pokusíte nastavit hodnotu 4.9E-324 pro vlastnost typu double, 64bitové platformy Windows s ní budou pracovat jako 0.0. Pokud tedy v distribuovaném prostředí systému zpráv platforma JMS nebo jiná aplikace nastaví hodnotu pro vlastnost typu double jako 4.9E-324 na libovolném 32bitovém počítači UNIX nebo Windows a produkt XMS .NET běží na 64bitovém počítači, hodnota vrácená funkcí GetDoubleProperty () je 0.0. Jedná se o známý problém produktu Microsoft .NET Framework 2.0 Framework.

Chybové stavy, které lze zpracovat za běhu

Návratové kódy z volání API jsou chybové stavy, které lze zpracovat za běhu. Způsob, jakým se zabývá tímto typem chyby, závisí na tom, zda používáte rozhraní API jazyka C nebo Jazyk C++ .

Jak zjistit chyby za běhu

Pokud aplikace volá funkci rozhraní API jazyka C a volání selže, vrátí se odezva s návratovým kódem jiným než XMS_OK s chybovým blokem XMS obsahujícím další informace o příčině selhání.

Rozhraní API produktu Jazyk C++ vyvolá výjimku, je-li použita metoda.

Aplikace používá modul listener pro výjimky k asynchronnímu upozornění na problém s připojením. Modul listener pro výjimky je dodáván a je inicializován pomocí rozhraní API XMS C nebo Jazyk C++ .

Jak zpracovat chyby za běhu

Některé chybové stavy jsou známkou toho, že některý prostředek není k dispozici a akce, kterou může aplikace provést, závisí na funkci produktu XMS , kterou volá aplikace. Pokud se například nepodaří připojit k serveru připojení, může se aplikace pokusit o opakované pokusy o opakované pokusy, dokud nebude vytvořeno připojení. Blok chyb XMS nebo výjimka nemusí obsahovat dostatek informací k určení akce, která má být provedena, a v těchto situacích často dochází k propojenému bloku chyb nebo výjimce, která obsahuje další specifické diagnostické informace.

V rozhraní C API vždy otestujte odezvu s návratovým kódem jiným než XMS_OK a vždy předejte blok chyb na volání rozhraní API. provedená akce obvykle závisí na tom, jaká funkce rozhraní API je aplikace používající.

V rozhraní API produktu Jazyk C++ vždy zahrňte volání metod do bloků try a zachytávat všechny typy výjimky XMS , určete třídu výjimky v konstruktu catch.

Listener výjimek je asynchronní cesta k chybovému stavu, která může být spuštěna kdykoli. Když je ve vlastním podprocesu spuštěna funkce modulu listener pro výjimky, obvykle se jedná o indikaci závažnějšího selhání, než je normální chybový stav rozhraní API produktu XMS . Může být provedena nějaká vhodná akce, ale musíte být opatrní, abyste dodržovali pravidla pro model podprocesů produktu XMS , jak je popsáno v tématu [“Model vytváření podprocesů”](#) na stránce 20.

Související pojmy

Model vytváření podprocesů

Obecná pravidla řídí, jak vícevláknová aplikace může používat objekty XMS .

Klienti Message Service Clients for .NET

Tento referenční oddíl obsahuje informace, které vám pomohou s používáním produktu Message Service Client for .NET. Tyto informace vám pomohou provést úlohy zahrnuté do programování s produktem XMS.

.NETRozhraní

Tato sekce dokumentuje rozhraní třídy .NET a jejich vlastnosti a metody.

Následující tabulka shrnuje všechna rozhraní, která jsou definována v rámci oboru názvů IBM . XMS .

<i>Tabulka 24. Souhrn rozhraní třídy .NET</i>	
Rozhraní	Popis
“IBytesMessage” na stránce 88	Bajtová zpráva je zpráva, jejíž tělo tvoří proud bajtů.
“IConnection” na stránce 98	Objekt připojení představuje aktivní připojení aplikace k serveru systému zpráv.
“IConnectionFactory” na stránce 101	Aplikace používá továrnu připojení k vytvoření připojení.
“Data IConnectionMetaData” na stránce 103	Objekt datového objektu ConnectionMetaposkytuje informace o připojení.
“Instinace” na stránce 104	Cíl je místo, kam aplikace odesílá zprávy, nebo je to zdroj, ze kterého aplikace přijímá zprávy, nebo obojí.
“ExceptionListener” na stránce 106	Aplikace používá modul listener pro výjimky k asynchronnímu upozornění na problém s připojením.
“Výjimka IllegalState” na stránce 106	XMS vyvolá tuto výjimku, pokud aplikace volá metodu v nesprávném nebo nevhodném čase, nebo pokud XMS není v odpovídajícím stavu pro požadovanou operaci.
“InitialContext” na stránce 106	Aplikace používá objekt InitialContext k vytvoření objektů z definic objektů, které jsou načteny z úložiště spravovaných objektů.
“Výjimka IDException InvalidClient” na stránce 109	XMS vyvolá tuto výjimku, pokud se aplikace pokusí nastavit identifikátor klienta pro připojení, ale identifikátor klienta je neplatný nebo je již používán.

Tabulka 24. Souhrn rozhraní třídy .NET (pokračování)

Rozhraní	Popis
“ Výjimka InvalidDestination ” na stránce 109	XMS vyvolá tuto výjimku, pokud aplikace určuje, že místo určení není platné.
“ Výjimka InvalidSelector ” na stránce 109	XMS vyvolá tuto výjimku, pokud aplikace poskytuje výraz selektoru zpráv, jehož syntaxe není platná.
“ IMapMessage ” na stránce 110	Mapová zpráva je zpráva, jejíž tělo tvoří sadu dvojic název-hodnota, kde má každá hodnota přidružený datový typ.
“ IMessage ” na stránce 119	Objekt zpráv představuje zprávu, kterou aplikace odesílá nebo přijímá. IMessage je nadtřída pro třídy zpráv, jako např. IMapMessage.
“ IMessageConsumer ” na stránce 125	Aplikace používá spotřebitele zpráv k přijetí zpráv odeslaných do místa určení.
“ MessageEOFException ” na stránce 128	XMS vyvolá tuto výjimku, pokud XMS narazí na konec proudu bajtů zpráv, když aplikace čte tělo bajtové zprávy.
“ Výjimka MessageFormat ” na stránce 128	XMS vyvolá tuto výjimku, pokud XMS narazí na zprávu s formátem, který není platný.
“ IMessageListener (delegát) ” na stránce 128	Aplikace používá modul listener pro zprávy k asynchronnímu příjmu zpráv.
“ MessageNotReadableException ” na stránce 129	XMS vyvolá tuto výjimku, pokud se aplikace pokusí číst tělo zprávy, která je pouze pro zápis.
“ MessageNotWritableException ” na stránce 129	XMS vyvolá tuto výjimku, pokud se aplikace pokusí o zápis do těla zprávy, která je jen pro čtení.
“ IMessageProducer ” na stránce 129	Aplikace používá producenta zpráv k odesílání zpráv do místa určení.
“ IObjectMessage ” na stránce 135	Zpráva objektu je zpráva, jejíž tělo tvoří serializovaný objekt Java nebo .NET.
“ IPropertyContext ” na stránce 136	IPropertyContext je abstraktní nadtřída, která obsahuje metody pro získání a nastavení vlastností. Tyto metody jsou děděny jinými třídami.
“ IQueueBrowser ” na stránce 145	Aplikace používá prohlížeč front k procházení zpráv ve frontě, aniž by je odebíráte.
“ Žadatel ” na stránce 147	Aplikace používá žadatele k odeslání zprávy požadavku a následné čekání na odpověď a přijetí odpovědi.
“ Výjimka ResourceAllocation ” na stránce 148	XMS vyvolá tuto výjimku, pokud XMS nemůže přidělit prostředky požadované metodou.
“ SecurityException ” na stránce 149	XMS vyvolá tuto výjimku, je-li identifikátor uživatele a heslo poskytnuté pro ověření aplikace odmítnuty. XMS také vyvolá tuto výjimku, pokud kontrola oprávnění selže a zabrání dokončení metody.
“ ISession ” na stránce 149	Relace je jednovláknový kontext pro odesílání a příjem zpráv.

Tabulka 24. Souhrn rozhraní třídy .NET (pokračování)

Rozhraní	Popis
“IStreamMessage” na stránce 160	Proudová zpráva je zpráva, jejíž tělo tvoří proud hodnot, kde má každá hodnota přidružený datový typ.
“ITextMessage” na stránce 169	Textová zpráva je zpráva, jejíž tělo tvoří řetězec.
“TransactionInProgressException” na stránce 170	XMS vyvolá tuto výjimku, pokud aplikace požádá o operaci, která není platná, protože probíhá transakce.
“TransactionRolledBackException” na stránce 170	XMS vyvolá tuto výjimku, pokud aplikace volá Session.commit() k potvrzení aktuální transakce, ale transakce se pak vrátí zpět.
XMSC	Pro produkt .NET jsou názvy vlastností a hodnoty produktu XMS definovány v této třídě jako veřejné konstanty. Další podrobnosti viz “Vlastnosti objektů XMS” na stránce 173 .
“Výjimka XMSException” na stránce 171	Pokud XMS zjistí chybu během zpracování volání metody .NET, příkaz XMS vyvolá výjimku. Výjimkou je objekt, který zapouzdřuje informace o chybě. Existují různé typy výjimek XMS a objekt XMSException je pouze jedním typem výjimky. Třída XMSException je však nadtřídou jiných tříd výjimek XMS. XMS generuje objekt XMSException v situacích, kdy není vhodný žádný z jiných typů výjimek.
“XMSFactoryFactory” na stránce 172	Pokud aplikace nepoužívá spravované objekty, použijte tuto třídu k vytvoření továren připojení, front a témat.

Definice každé metody uvádí kódy výjimek, které XMS může vrátit, pokud zjistí chybu během zpracování volání metody. Každý kód výjimky je představován svou pojmenovanou konstantou, která má odpovídající výjimku.

Související pojmy

[Vytváření vlastních aplikací](#)

Sestavujete své vlastní aplikace, jako např. sestavení ukázkových aplikací.

[Zápis aplikací XMS](#)

Témata v této sekci obsahují informace, které vám pomohou při psaní aplikací produktu XMS.

[Zápis aplikací produktu XMS .NET](#)

Témata v této sekci obsahují informace, které vám pomohou při psaní aplikací produktu XMS .NET.

Související odkazy

[Vlastnosti objektů XMS](#)

Tento kapitola dokumentuje vlastnosti objektu definované parametrem XMS.

IBytesMessage

Bajtová zpráva je zpráva, jejíž tělo tvoří proud bajtů.

Hierarchie dědičnosti:

IBM.XMS.IPropertyContext


```
|
+----IBM.XMS.IMessage
      |
      +----IBM.XMS.IBytesMessage
```

Související odkazy

Bajtové zprávy

Tělo bajtové zprávy obsahuje proud bajtů. Tělo obsahuje pouze aktuální data a je odpovědností odesílajícího a přijímacího programu interpretovat tato data.

Vlastnosti produktu .NET

BodyLength - Získat délku těla

Rozhraní:

```
Int64 BodyLength
{
    get;
}
```

Získejte délku těla zprávy v bajtech, je-li tělo zprávy jen pro čtení.

Vrácená hodnota je délka celého těla bez ohledu na to, kde je kurzor pro čtení zprávy momentálně umístěn.

Výjimky:

- Výjimka `XMSEException`
- `MessageNotReadableException`

Metody

ReadBoolean - Čtení hodnoty typu `Boolean`

Rozhraní:

```
Boolean ReadBoolean();
```

Přečtěte si logickou hodnotu z proudu zpráv v bajtech.

Parametry:

Není

vrátí:

Logická hodnota, která je přečtená.

Výjimky:

- Výjimka `XMSEException`
- `MessageNotReadableException`
- `MessageEOFException`

ReadSignedByte - Read Byte

Rozhraní:

```
Int16 ReadSignedByte();
```

Přečíst další bajt z proudu zpráv bajtů jako podepsané 8bitové celé číslo.

Parametry:

Není

vrátí:

Bajt, který je čten.

Výjimky:

- Výjimka XMSEException
- MessageNotReadableException
- MessageEOFException

ReadBytes -Čtení bajtů

Rozhraní:

```
Int32 ReadBytes(Byte[] array);  
Int32 ReadBytes(Byte[] array, Int32 length);
```

Čtení pole bajtů z proudu bajtů zpráv, které začíná od aktuální pozice kurzoru.

Parametry:

array (výstup)

Vyrovňovací paměť, která má obsahovat pole bajtů, které se čtou. Pokud je počet bajtů zbývajících k přečtení z proudu před voláním větší nebo roven délce vyrovňovací paměti, vyrovňovací paměť je zaplněna. Jinak bude vyrovňovací paměť částečně vyplněna všemi zbývajícími bajty.

Určíte-li na vstupu ukazatel null, metoda přeskóčí bajty, aniž by je četla. Pokud je počet bajtů zbývajících k přečtení z proudu před voláním větší nebo rovnou délce vyrovňovací paměti, počet vynechaných bajtů je roven délce vyrovňovací paměti. Jinak budou všechny zbývajcí bajty vynechány. Kurzor zůstává na další pozici, aby bylo možné číst v proudu bajtů zpráv.

délka (vstup)

Délka vyrovňovací paměti v bajtech

vrátí:

Počet bajtů, které jsou načteny do vyrovňovací paměti. Je-li vyrovňovací paměť částečně vyplněna, hodnota je menší než délka vyrovňovací paměti, což znamená, že zde již nejsou žádné další bajty, které mají být čteny. Pokud zbývají před voláním žádné bajty, které mají být přečteny z proudu, hodnota je XMSC_END_OF_STREAM.

Uvedete-li na vstupu ukazatel null, metoda nevrátí žádnou hodnotu.

Výjimky:

- Výjimka XMSEException
- MessageNotReadableException

ReadChar -Čtení znaků

Rozhraní:

```
Char ReadChar();
```

Přečtete si další 2 bajty z bajtového proudu zpráv jako znak.

Parametry:

Není

vrátí:

Znak, který se čte.

Výjimky:

- Výjimka XMSEException
- MessageNotReadableException
- MessageEOFException

ReadDouble -Přečíst číslo s pohyblivou řádovou čárkou s dvojitou přesností

Rozhraní:

```
Double ReadDouble();
```

Přečtěte si dalších 8 bajtů z proudu zpráv v bajtech jako číslo s pohyblivou řádovou čárkou a dvojitou přesností.

Parametry:

Není

vrátí:

Číslo s plovoucí řádovou čárkou a dvojitou přesností, které je přečteno.

Výjimky:

- Výjimka XMSEException
- MessageNotReadableException
- MessageEOFException

ReadFloat -Čtení čísla s pohyblivou řádovou čárkou

Rozhraní:

```
Single ReadFloat();
```

Přečtěte si následující 4 bajty z bajtového proudu zpráv jako číslo s pohyblivou řádovou čárkou.

Parametry:

Není

vrátí:

Číslo v plovoucí řádové čárce, které se čte.

Výjimky:

- Výjimka XMSEException
- MessageNotReadableException
- MessageEOFException

ReadInt -Přečíst celé číslo

Rozhraní:

```
Int32 ReadInt();
```

Přečtěte si další 4 bajty z bajtového proudu zpráv jako 32bitové celé číslo se znaménkem.

Parametry:

Není

vrátí:

Celé číslo, které je přečteno.

Výjimky:

- Výjimka XMSEException
- MessageNotReadableException
- MessageEOFException

ReadLong - Číst dlouhé celé číslo

Rozhraní:

```
Int64 ReadLong();
```

Přečtěte si dalších 8 bajtů z bajtového proudu zpráv jako podepsané 64bitové celé číslo.

Parametry:

Není

vrátí:

Dlouhé celé číslo, které je přečteno.

Výjimky:

- Výjimka XMSEException
- MessageNotReadableException
- MessageEOFException

ReadShort -Přečíst krátké celé číslo

Rozhraní:

```
Int16 ReadShort();
```

Přečtěte si další 2 bajty z bajtového proudu zpráv jako 16bitové celé číslo se znaménkem.

Parametry:

Není

vrátí:

Krátké celé číslo, které je přečteno.

Výjimky:

- Výjimka XMSEException
- MessageNotReadableException
- MessageEOFException

ReadByte -Čtení nepodepsaných bajtů

Rozhraní:

```
Byte ReadByte();
```

Přečíst další bajt z proudu bajtů zpráv jako 8bitové celé číslo typu unsigned.

Parametry:

Není

vrátí:

Bajt, který je čten.

Výjimky:

- Výjimka XMSEException
- MessageNotReadableException
- MessageEOFException

ReadUnsignedShort-Read Unsigned Short Integer

Rozhraní:

```
Int32 ReadUnsignedShort();
```

Přečtěte si další 2 bajty z proudu zpráv o bajtech jako 16bitové celé číslo bez znaménka.

Parametry:

Není

vrátí:

Krátké celé číslo bez znaménka, které je přečteno.

Výjimky:

- Výjimka XMSEException
- MessageNotReadableException
- MessageEOFException

ReadUTF -Čtení řetězce UTF

Rozhraní:

```
String ReadUTF();
```

Přečíst řetězec, zakódovaný v UTF-8, z bajtového proudu zpráv.

Poznámka: Před voláním funkce ReadUTF() se ujistěte, že kurzor ve vyrovnávací paměti ukazuje na začátek toku bajtů zpráv.

Parametry:

Není

vrátí:

Řetězcový objekt zapouzdřující řetězec, který se čte.

Výjimky:

- Výjimka XMSEException
- MessageNotReadableException
- MessageEOFException

Reset-Reset

Rozhraní:

```
void Reset();
```

Umístěte tělo zprávy do režimu jen pro čtení a přesuňte kurzor na začátek bajtového proudu zpráv.

Parametry:

Není

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- MessageNotReadableException

WriteBoolean -Zapsat logickou hodnotu

Rozhraní:

```
void WriteBoolean(Boolean value);
```

Zapište logickou hodnotu do bajtového proudu zpráv.

Parametry:

hodnota (vstup)

Logická hodnota, která má být zapsána.

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

WriteByte -Zápis bajtu

Rozhraní:

```
void WriteByte(Byte value);  
void WriteSignedByte(Int16 value);
```

Napište bajt do bajtového proudu zpráv.

Parametry:

hodnota (vstup)

Bajt, který se má zapsat.

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

WriteBytes -Zápis bajtů

Rozhraní:

```
void WriteBytes(Byte[] value);
```

Zapisovat pole bajtů do proudu bajtů zprávy.

Parametry:

hodnota (vstup)

Pole bajtů, které mají být zapsány.

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

WriteBytes -Zápis částečných bajtů do pole

Rozhraní:

```
void WriteBytes(Byte[] value, int offset, int length);
```

Napište dílčí pole bajtů do proudu zpráv o bajtech, jak je definováno uvedenou délkou.

Parametry:

hodnota (vstup)

Pole bajtů, které mají být zapsány.

offset (vstup)

Počáteční bod pro pole bajtů, které mají být zapsány.

délka (vstup)

Počet bajtů, které se mají zapsat.

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

WriteChar -Zápis znaku

Rozhraní:

```
void WriteChar(Char value);
```

Napište znak do toku bajtů zpráv jako 2 bajty, bajt s velkým pořadovým číslem jako první.

Parametry:

hodnota (vstup)

Znak, který má být zapsán.

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

WriteDouble -Write Double Precision Floating Point Number

Rozhraní:

```
void WriteDouble(Double value);
```

Převeďte číslo s pohyblivou řádovou čárkou a dvojitou přesností na dlouhé celé číslo a запиšte dlouhé celé číslo na proud bajtů zpráv jako 8 bajtů, bajt s velkým pořadovým číslem jako první.

Parametry:**hodnota (vstup)**

Číslo s plovoucí řádovou čárkou s dvojitou přesností, které se má zapsat.

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

WriteFloat -Zápis čísla s pohyblivou řádovou čárkou

Rozhraní:

```
void WriteFloat(Single value);
```

Převeďte číslo s pohyblivou řádovou čárkou na celé číslo a запиšte celé číslo do proudu bajtů zpráv jako 4 bajty, bajt s velkým pořadovým číslem jako první a první.

Parametry:**hodnota (vstup)**

Číslo plovoucí řádové čárky, které se má zapsat.

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

WriteInt -Zapsat celé číslo

Rozhraní:

```
void WriteInt(Int32 value);
```

Zapište celé číslo do proudu bajtů zpráv jako 4 bajty, bajt s velkým pořadovým číslem jako první.

Parametry:**hodnota (vstup)**

Celé číslo, které má být zapsáno.

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

WriteLong -zapište dlouhé celé číslo

Rozhraní:

```
void WriteLong(Int64 value);
```

Zapište dlouhé celé číslo na proud bajtů zpráv jako 8 bajtů, bajt s velkým pořadovým číslem jako první.

Parametry:**hodnota (vstup)**

Dlouhé celé číslo, které má být zapsáno.

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

WriteObject -Zapsat objekt

Rozhraní:

```
void WriteObject(Object value);
```

Napište uvedený objekt do bajtového proudu zpráv.

Parametry:**hodnota (vstup)**

Objekt, který má být zapsán, což musí být odkaz na primitivní typ.

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

WriteShort -zapište krátké celé číslo

Rozhraní:

```
void WriteShort(Int16 value);
```

Napište krátké celé číslo do toku bajtů zpráv jako 2 bajty, bajt s velkým pořadovým číslem jako první.

Parametry:

hodnota (vstup)

Krátké celé číslo, které má být zapsáno.

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

WriteUTF - Zápis řetězce UTF

Rozhraní:

```
void WriteUTF(String value);
```

Napište řetězec do proudu zpráv v bajtech kódovaný v UTF-8.

Parametry:

hodnota (vstup)

Řetězcový objekt zapouzdřující řetězec, který má být zapsán.

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

Zděděné vlastnosti a metody

Níže uvedené vlastnosti jsou zděděny z rozhraní IMessage:

JMSCorrelationID, JMSDeliveryMode, JMSDestination, JMSExpiration, JMSMessageID, JMSPriority, JMSRedelivered, JMSReplyTo, JMSTimestamp, JMSType, Properties

Níže uvedené metody jsou zděděny z rozhraní IMessage:

clearBody, clearProperties, PropertyExists

Níže uvedené metody jsou zděděny z rozhraní IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

IConnection

Objekt připojení představuje aktivní připojení aplikace k serveru systému zpráv.

Hierarchie dědičnosti:

```
IBM.XMS.IPropertyContext
|
+---- IBM.XMS.IConnection
```

For a list of the XMS defined properties of a Connection object, see [“Vlastnosti připojení”](#) na stránce 174.

Vlastnosti produktu .NET

ClientID - Získání a nastavení ID klienta

Rozhraní:

```
String ClientID
{
    get;
    set;
}
```

Získejte a nastavte identifikátor klienta pro připojení.

Identifikátor klienta může být buď předkonfigurován administrátorem v rámci ConnectionFactory, nebo přiřazen nastavením ClientID.

Identifikátor klienta se používá pouze k podpoře trvalých odběrů v doméně publikování/odběru a je ignorován v dvoubodové doméně.

Pokud aplikace nastavuje identifikátor klienta pro připojení, musí to aplikace provést okamžitě po vytvoření připojení a před provedením jakékoli jiné operace na připojení. Pokud se aplikace pokusí o nastavení identifikátoru klienta po tomto bodu, volání vyvolá výjimku `IllegalStateException`.

Tato vlastnost není platná pro připojení v reálném čase ke zprostředkovateli.

Výjimky:

- Výjimka `XMSEException`
- Výjimka `IllegalState`
- Výjimka `IDException InvalidClient`

ExceptionHandler - Získat a nastavit modul listener pro výjimky

Rozhraní:

```
ExceptionHandler ExceptionListener
{
    get;
    set;
}
```

Získejte modul listener pro výjimky, který je registrován u připojení, a zaregistrujte modul listener pro výjimky s připojením.

Není-li modul listener pro výjimky registrován s připojením, metoda vrátí hodnotu null. Je-li již modul listener pro výjimky registrován s připojením, můžete zrušit registraci zadáním hodnoty null namísto modulu listener pro výjimky.

Další informace o použití listenerů výjimek viz [“Listenery zpráv a výjimek v produktu .NET”](#) na stránce 46.

Výjimky:

- Výjimka `XMSEException`

Metadata-získat metadata

Rozhraní:

```
ICollectionMetaData MetaData
{
    get;
}
```

Získejte metadata pro připojení.

Výjimky:

- Výjimka XMSEException

Metody

Zavřít-Zavřít připojení

Rozhraní:

```
void Close();
```

Zavřete připojení.

Pokud se aplikace pokusí zavřít připojení, které je již uzavřeno, volání se ignoruje.

Parametry:

Není

vrátí:

Void

Výjimky:

- Výjimka XMSEException

CreateSession -Vytvořit relaci

Rozhraní:

```
ISession CreateSession(Boolean transacted,
                        AcknowledgeMode acknowledgeMode);
```

Vytvořte relaci.

Parametry:

transkovaný (vstup)

Hodnota `True` znamená, že relace je zpracovávána. Hodnota `False` znamená, že relace se neobchoduje.

V případě připojení v reálném čase ke zprostředkovateli musí být hodnota `False`.

acknowledgeMode (vstup)

Indikuje, jak jsou potvrzovány zprávy přijaté aplikací. Hodnota musí být jednou z následujících hodnot z výčtového nástroje `AcknowledgeMode` :

```
AcknowledgeMode.AutoAcknowledge
AcknowledgeMode.ClientAcknowledge
AcknowledgeMode.DupsOkAcknowledge
```

Pro připojení v reálném čase ke zprostředkovateli musí být tato hodnota `AcknowledgeMode.AutoAcknowledge` nebo `AcknowledgeMode.DupsOkAcknowledge`

Tento parametr je ignorován, pokud relace obsahuje transakci. Další informace o režimech potvrzení naleznete v tématu [“Potvrzení zprávy”](#) na stránce 24.

vrátí:

Objekt `Session`

Výjimky:

- Výjimka `XMSEException`

Start-Spustit připojení

Rozhraní:

```
void Start();
```

Spusťte nebo restartujte doručení příchozích zpráv pro připojení. Volání je ignorováno, je-li připojení již spuštěno.

Parametry:

Není

vrátí:

`Void`

Výjimky:

- Výjimka `XMSEException`

Zastavit-Zastavit připojení

Rozhraní:

```
void Stop();
```

Zastavte doručování příchozích zpráv pro připojení. Volání je ignorováno, pokud je připojení již zastaveno.

Parametry:

Není

vrátí:

`Void`

Výjimky:

- Výjimka `XMSEException`

Zděděné vlastnosti a metody

Níže uvedené metody jsou zděděny z rozhraní `IPropertyContext`:

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

IConnectionFactory

Aplikace používá továrnu připojení k vytvoření připojení.

Hierarchie dědičnosti:

```
IBM.XMS.IPropertyContext
|
+---- IBM.XMS.IConnectionFactory
```

Seznam vlastností definovaných XMS pro objekt ConnectionFactory naleznete v tématu [“Vlastnosti objektu ConnectionFactory”](#) na stránce 175.

Související pojmy

ConnectionFactoryy a objekty připojení

Objekt ConnectionFactory poskytuje šablonu, kterou aplikace používá k vytvoření objektu připojení. Aplikace používá objekt Connection k vytvoření objektu relace.

Připojení ke sběrnici pro integraci služeb

Aplikace XMS se může připojit ke sběrnici pro integraci služeb produktu WebSphere Application Server buď pomocí přímého připojení TCP/IP, nebo pomocí protokolu HTTP přes TCP/IP.

Zabezpečená připojení ke správci front produktu IBM MQ

Chcete-li povolit aplikaci XMS .NET zabezpečené připojení ke správci front produktu IBM MQ , musí být příslušné vlastnosti definovány v objektu ConnectionFactory .

Zabezpečená připojení ke stroji systému zpráv produktu WebSphere Application Server service integration bus

Chcete-li povolit aplikaci XMS .NET zabezpečené připojení ke stroji systému zpráv produktu WebSphere Application Server service integration bus , musí být příslušné vlastnosti definovány v objektu ConnectionFactory .

Mapování vlastností pro spravované objekty

Chcete-li aplikacím povolit použití továrny připojení a definic cílových objektů produktu IBM MQ JMS a WebSphere Application Server , musí být vlastnosti načtené z těchto definic mapovány na odpovídající vlastnosti produktu XMS , které lze nastavit v továrnách a cílech připojení XMS .

Související úlohy

Vytvoření spravovaných objektů

Definice objektu ConnectionFactory a Cílový objekt, které aplikace produktu XMS vyžadují k vytvoření připojení k serveru systému zpráv, musí být vytvořeny pomocí příslušných nástrojů pro správu.

Související odkazy

Povinné vlastnosti pro spravované objekty ConnectionFactory

Když aplikace vytvoří továrnu připojení, musí být definován počet vlastností pro vytvoření připojení k serveru systému zpráv.

Metody

CreateConnection -Vytvoření továrny připojení (s použitím výchozí identity uživatele)

Rozhraní:

```
IConnection CreateConnection();
```

Vytvořte továrnu připojení s výchozími vlastnostmi.

Pokud se připojujete k WebSphere MQ a není nastaven parametr XMSCS_USERID, pak správce front standardně používá userID přihlášeného uživatele. Požadujete-li další ověření na úrovni připojení pro jednotlivé uživatele, můžete napsat uživatelskou proceduru pro ověření klienta, která je konfigurována v produktu WebSphere MQ.

Parametry:

Není

Výjimky:

- Výjimka XMSEException

CreateConnection - Vytvořit připojení (pomocí zadané identity uživatele)

Rozhraní:

```
IConnection CreateConnection(String userId, String password);
```

Vytvořte připojení s použitím zadané identity uživatele.

Pokud se připojujete k WebSphere MQ a není nastaven parametr XMSCS_USERID, pak správce front standardně používá userID přihlášeného uživatele. Požadujete-li další ověření na úrovni připojení pro jednotlivé uživatele, můžete napsat uživatelskou proceduru pro ověření klienta, která je konfigurována v produktu WebSphere MQ.

Připojení se vytvoří v zastaveném režimu. Žádné zprávy se doručují až do volání aplikace

Connection.start().

Parametry:

userID (vstup)

Řetězcový objekt zapouzdřující identifikátor uživatele, který má být použit pro ověření aplikace. Zadáte-li hodnotu null, dojde k pokusu o vytvoření připojení bez ověření.

password (vstup)

Řetězcový objekt zapouzdřující heslo, které má být použito pro ověření aplikace. Zadáte-li hodnotu null, dojde k pokusu o vytvoření připojení bez ověření.

vrátí:

Objekt připojení.

Výjimky:

- Výjimka XMSEException
- VÝJIMKA XMS_X_SECURITY_EXCEPTION

Zděděné vlastnosti a metody

Níže uvedené metody jsou zděděny z rozhraní [IPropertyContext](#):

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

Data IConnectionMetaData

Objekt datového objektu ConnectionMetaposkytuje informace o připojení.

Hierarchie dědičnosti:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IConnectionMetaData
```

Seznam vlastností definovaných XMS pro objekt dat ConnectionMetanaleznete v tématu [“Vlastnosti dat ConnectionMeta”](#) na stránce 181.

Vlastnosti produktu .NET

Rozhraní:

```
System.Collections.IEnumerator JMSXPropertyNames
{
    get;
}
```

Vrátí výčet názvů vlastností zpráv definovaných produktem JMS , které jsou podporovány připojením. JMS vlastností definovaných zpráv není podporováno v reálném čase připojení ke zprostředkovateli.

Výjimky:

- Výjimka XMSException

Zděděné vlastnosti a metody

Níže uvedené metody jsou zděděny z rozhraní IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

Instinace

Cíl je místo, kam aplikace odesílá zprávy, nebo je to zdroj, ze kterého aplikace přijímá zprávy, nebo obojí.

Hierarchie dědičnosti:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IDestination
```

Seznam definovaných vlastností XMS pro objekt Destination lze najít v tématu “Vlastnosti místa určení” na stránce 181.

Související pojmy

ConnectionFactory a objekty připojení

Objekt ConnectionFactory poskytuje šablonu, kterou aplikace používá k vytvoření objektu připojení. Aplikace používá objekt Connection k vytvoření objektu relace.

Připojení ke sběrnici pro integraci služeb

Aplikace XMS se může připojit ke sběrnici pro integraci služeb produktu WebSphere Application Server buď pomocí přímého připojení TCP/IP, nebo pomocí protokolu HTTP přes TCP/IP.

Cíle

Aplikace XMS používá objekt Destination k určení místa určení odesílaných zpráv a zdroje přijímaných zpráv.

Cílové zástupné znaky

Produkt XMS poskytuje podporu pro cílové zástupné znaky a zajišťuje, aby zástupné znaky mohly být předávány do místa, kde jsou potřeba pro porovnávání. Pro každý typ serveru, se kterým může pracovat produkt XMS , existuje jiný zástupný schéma.

Jednotné identifikátory prostředků tématu

Identifikátor URI (Uniform Resource Identifier) tématu určuje název tématu; pro něj lze také zadat jednu či více vlastností.

Jednotné identifikátory prostředků fronty

Identifikátor URI pro frontu určuje název fronty. Může také určovat jednu či více vlastností fronty.

Dočasná místa určení

Aplikace produktu XMS mohou vytvářet a používat dočasná místa určení.

Mapování vlastností pro spravované objekty

Chcete-li aplikacím povolit použití továrny připojení a definic cílových objektů produktu IBM MQ JMS a WebSphere Application Server , musí být vlastnosti načtené z těchto definic mapovány na odpovídající vlastnosti produktu XMS , které lze nastavit v továrnách a cílech připojení XMS .

Související úlohy

Vytvoření spravovaných objektů

Definice objektu ConnectionFactory a Cílový objekt, které aplikace produktu XMS vyžadují k vytvoření připojení k serveru systému zpráv, musí být vytvořeny pomocí příslušných nástrojů pro správu.

Související odkazy

Povinné vlastnosti pro spravované objekty typu Destination

Aplikace, která vytváří cíl, musí nastavit několik vlastností, které aplikace na administrovaného cílovém objektu.

Vlastnosti produktu .NET

Název-Získat název cíle

Rozhraní:

```
String Name
{
    get;
}
```

Získejte název místa určení. Název je řetězec zapouzdřující buď název fronty, nebo název tématu.

Výjimky:

- Výjimka XMSEException

TypeId -Získat typ cíle

Rozhraní:

```
DestinationType TypeId
{
    get;
}
```

Získejte typ cíle. Typ cíle je jedna z následujících hodnot:

```
DestinationType.Queue
DestinationType.Topic
```

Výjimky:

- Výjimka XMSEException

Zděděné vlastnosti a metody

Níže uvedené metody jsou zděděny z rozhraní IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

ExceptionHandler

Hierarchie dědičnosti:

Není

Aplikace používá modul listener pro výjimky k asynchronnímu upozornění na problém s připojením.

Pokud aplikace používá připojení pouze k asynchronnímu příjmu zpráv a k žádnému jinému účelu, pak jedinou možností, jak se aplikace může naučit o problému s připojením, je použití modulu listener pro výjimky. V jiných situacích může modul listener pro výjimky poskytnout bezprostřednější způsob, jak se naučit o problému s připojením, než čekat na další synchronní volání na XMS.

Delegát

ExceptionHandler -Listener výjimek

Rozhraní:

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

Oznámit aplikaci problému s připojením.

Metody, které implementují tohoto delegáta, mohou být zaregistrovány spolu s připojením.

Další informace o použití listenerů výjimek viz [“Listenery zpráv a výjimek v produktu .NET”](#) na stránce 46.

Parametry:

exception (vstup)

Ukazatel na výjimku vytvořenou příkazem XMS.

vrátí:

Void

Výjimka IllegalStateException

Hierarchie dědičnosti:

```
IBM.XMS.XMSException
|
+----IBM.XMS.Exception
|
+----IBM.XMS.IllegalStateException
```

XMS vyvolá tuto výjimku, pokud aplikace volá metodu v nesprávném nebo nevhodném čase, nebo pokud XMS není v odpovídajícím stavu pro požadovanou operaci.

Zděděné vlastnosti a metody

Níže uvedené metody jsou zděděny z rozhraní [XMSException](#):

[GetErrorCode](#), [GetLinkedException](#)

InitialContext

Aplikace používá objekt InitialContext k vytvoření objektů z definic objektů, které jsou načteny z úložiště spravovaných objektů.

Hierarchie dědičnosti:

Není

Související pojmy

[Vlastnosti InitialContext](#)

Parametry konstruktora InitialContext zahrnují umístění úložiště spravovaných objektů, které jsou poskytnuty jako identifikátor URI (Uniform Resource Indicator). Aby mohla aplikace vytvořit připojení k úložišti, může být nezbytné poskytnout více informací než informace obsažené v identifikátoru URI.

Formát identifikátoru URI pro počáteční kontexty produktu XMS

Umístění úložiště spravovaných objektů je poskytováno jako jednotný indikátor prostředku (URI). Formát identifikátoru URI závisí na typu kontextu.

Načítání spravovaných objektů

Produkt XMS načítá administrovaný objekt z úložiště s použitím adresy zadané při vytvoření objektu InitialContext nebo ve vlastnostech InitialContext .

Související úlohy

Objekty InitialContext

Aplikace musí vytvořit počáteční kontext, který má být použit k vytvoření připojení k úložišti spravovaných objektů, aby bylo možné načíst požadované spravované objekty.

Vlastnosti produktu .NET

Prostředí-Získat prostředí

Rozhraní:

```
Hashtable Environment
{
    get;
}
```

Získejte prostředí.

Výjimky:

- Výjimky jsou specifické pro používanou adresářovou službu.

Konstruktory

InitialContext -vytvořit počáteční kontext

Rozhraní:

```
InitialContext(Hashtable env);
```

Vytvořte objekt InitialContext .

Parametry:

Informace potřebné k vytvoření připojení k úložišti spravovaných objektů jsou poskytovány konstruktoru v tabulce Hashtable prostředí.

Výjimky:

- Výjimka XMSEException

Metody

Prostředí AddTo-přidání nové vlastnosti do prostředí

Rozhraní:

```
Object AddToEnvironment(String propName, Object propVal);
```

Přidejte novou vlastnost do prostředí.

Parametry:**propName (vstup)**

Řetězcový objekt zapouzdřující název vlastnosti, která má být přidána.

propVal (vstup)

Hodnota vlastnosti, která má být přidána.

vrátí:

Původní hodnota vlastnosti.

Výjimky:

- Výjimky jsou specifické pro používanou adresářovou službu.

Zavřít-Zavřít tento kontext

Rozhraní:

```
void Close()
```

Zavřete tento kontext.

Parametry:

Není

vrátí:

Není

Výjimky:

- Výjimky jsou specifické pro používanou adresářovou službu.

Vyhledat-Vyhledat objekt v počátečním kontextu

Rozhraní:

```
Object Lookup(String name);
```

Vytvoření objektu z definice objektu, která se načítá z úložiště spravovaných objektů.

Parametry:**name (vstup)**

Řetězcový objekt zapouzdřující název administrovaného objektu, který má být načten. Název může být buď jednoduchý název, nebo komplexní název. Další podrobnosti viz [“Načítání spravovaných objektů”](#) na stránce 62.

vrátí:

Buďto objekt IConnectionFactory, nebo hodnota IDestination, v závislosti na typu načítaného objektu. Pokud má funkce přístup k adresáři, ale nemůže najít požadovaný objekt, je vrácena hodnota null.

Výjimky:

- Výjimky jsou specifické pro používanou adresářovou službu.

RemoveFromEnvironment-Odebrat vlastnost z prostředí

Rozhraní:

```
Object RemoveFromEnvironment(String propName);
```

Odeberte vlastnost z prostředí.

Parametry:**propName (vstup)**

Řetězcový objekt zapouzdřující název vlastnosti, která má být odebrána.

vrátí:

Objekt, který byl odebrán.

Výjimky:

- Výjimky jsou specifické pro používanou adresářovou službu.

Výjimka `IXMLException InvalidClient`

Hierarchie dědičnosti:

```
IBM.XMS.XMLException
|
+----IXMLException
|
+----IXML.InvalidClientIXMLException
```

XMS vyvolá tuto výjimku, pokud se aplikace pokusí nastavit identifikátor klienta pro připojení, ale identifikátor klienta je neplatný nebo je již používán.

Zděděné vlastnosti a metody

Níže uvedené metody jsou zděděny z rozhraní `IXMLException`:

[GetErrorCode](#), [GetLinkedException](#)

Výjimka `IXMLException InvalidDestination`

Hierarchie dědičnosti:

```
IBM.XMS.XMLException
|
+----IXMLException
|
+----IXML.InvalidDestinationIXMLException
```

XMS vyvolá tuto výjimku, pokud aplikace určuje, že místo určení není platné.

Zděděné vlastnosti a metody

Níže uvedené metody jsou zděděny z rozhraní `IXMLException`:

[GetErrorCode](#), [GetLinkedException](#)

Výjimka `IXMLException InvalidSelector`

Hierarchie dědičnosti:

```
IBM.XMS.XMLException
|
+----IXMLException
|
+----IXML.InvalidSelectorIXMLException
```

XMS vyvolá tuto výjimku, pokud aplikace poskytuje výraz selektoru zpráv, jehož syntaxe není platná.

Zděděné vlastnosti a metody

Níže uvedené metody jsou zděděny z rozhraní `IXMLException`:

[GetErrorCode](#), [GetLinkedException](#)

IMapMessage

Mapová zpráva je zpráva, jejíž tělo tvoří sadu dvojic název-hodnota, kde má každá hodnota přidružený datový typ.

Hierarchie dědičnosti:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IMessage
|
+----IBM.XMS.IMapMessage
```

Když aplikace získá hodnotu páru název-hodnota, lze hodnotu převést pomocí XMS na jiný datový typ. Další informace o této formě implicitní konverze viz [“Mapové zprávy” na stránce 74](#).

Související odkazy

[Mapové zprávy](#)

Tělo zprávy mapy obsahuje sadu dvojic název-hodnota, kde má každá hodnota přidružený datový typ.

Vlastnosti produktu .NET

MapNames -Získat názvy map

Rozhraní:

```
System.Collections.IEnumerator MapNames
{
    get;
}
```

Získejte výčet názvů v těle zprávy mapování.

Výjimky:

- Výjimka XMSEException

Metody

GetBoolean -Získat logickou hodnotu

Rozhraní:

```
Boolean GetBoolean(String name);
```

Získejte logickou hodnotu identifikovanou názvem z těla zprávy mapování.

Parametry:

name (vstup)

Řetězcový objekt zapouzdřující název, který identifikuje logickou hodnotu.

vrátí:

Logická hodnota načtená z těla zprávy mapování.

Výjimky:

- Výjimka XMSEException

GetByte - Získat bajt

Rozhraní:

```
Byte    GetByte(String name);  
Int16   GetSignedByte(String name);
```

Získejte bajt identifikovaný názvem z těla zprávy mapování.

Parametry:

name (vstup)

Řetězcový objekt zapouzdřující název, který identifikuje bajt.

vrátí:

Bajt načtený z těla zprávy mapy. Na bajtu se neprovádí žádná konverze dat.

Výjimky:

- Výjimka XMSEException

GetBytes - získání bajtů

Rozhraní:

```
Byte[]  GetBytes(String name);
```

Získejte pole bajtů identifikovaných názvem z těla zprávy mapování.

Parametry:

name (vstup)

Řetězcový objekt zapouzdřující název, který identifikuje pole bajtů.

vrátí:

Počet bajtů v poli.

Výjimky:

- Výjimka XMSEException

GetChar - Získání znaku

Rozhraní:

```
Char    GetChar(String name);
```

Získejte znak identifikovaný názvem z těla zprávy mapování.

Parametry:

name (vstup)

Řetězcový objekt zapouzdřující název, který identifikuje znak.

vrátí:

Znak načtený z těla zprávy mapy.

Výjimky:

- Výjimka XMSEException

GetDouble - Získat číslo s pohyblivou řádovou čárkou dvojitě přesnosti

Rozhraní:

```
Double GetDouble(String name);
```

Získejte číslo s plovoucí řádovou čárkou a dvojitou přesnost, které jsou identifikovány názvem z textu zprávy mapy.

Parametry:

name (vstup)

Řetězcový objekt zapouzdřující název, který identifikuje číslo s plovoucí řádovou čárkou a dvojitou přesností.

vrátí:

Číslo s plovoucí řádovou čárkou a dvojitou přesností načtené z těla zprávy mapy.

Výjimky:

- Výjimka XMSEException

GetFloat - získat číslo s pohyblivou řádovou čárkou

Rozhraní:

```
Single GetFloat(String name);
```

Získejte číslo s pohyblivou řádovou čárkou identifikované názvem z textu zprávy mapy.

Parametry:

name (vstup)

Řetězcový objekt zapouzdřující název, který identifikuje číslo s pohyblivou řádovou čárkou.

vrátí:

Číslo s plovoucí řádovou čárkou načtené z textu zprávy mapy.

Výjimky:

- Výjimka XMSEException

GetInt - Získat celé číslo

Rozhraní:

```
Int32 GetInt(String name);
```

Získejte celé číslo identifikované názvem z těla zprávy mapování.

Parametry:

name (vstup)

Řetězcový objekt zapouzdřující název, který identifikuje celé číslo.

vrátí:

Celé číslo načtené z těla zprávy mapování.

Výjimky:

- Výjimka XMSEException

GetLong -Získat dlouhé celé číslo

Rozhraní:

```
Int64 GetLong(String name);
```

Získejte dlouhé celé číslo identifikované názvem z textu zprávy mapy.

Parametry:

name (vstup)

Řetězový objekt zapouzdřující název, který identifikuje dlouhé celé číslo.

vrátí:

Dlouhé celé číslo načtené z těla zprávy mapování.

Výjimky:

- Výjimka XMSEException

GetObject -Získat objekt

Rozhraní:

```
Object GetObject(String name);
```

Získejte odkaz na hodnotu páru název-hodnota z textu zprávy mapy. Dvojice název-hodnota je identifikována názvem.

Parametry:

name (vstup)

Řetězový objekt zapouzdřující název dvojice název-hodnota.

vrátí:

Hodnota, která je jedním z následujících typů objektů:

Boolean
Byte
Byte[]
Char
Double
Single
Int32
Int64
Int16
String

Výjimky:

- Výjimka XMSEException

GetShort -Získat krátké celé číslo

Rozhraní:

```
Int16 GetShort(String name);
```

Získejte krátké celé číslo identifikované názvem z textu zprávy mapy.

Parametry:**name (vstup)**

Řetězcový objekt zapouzdřující název, který identifikuje krátké celé číslo.

vrátí:

Krátké celé číslo načtené z těla zprávy mapování.

Výjimky:

- Výjimka XMSEException

GetString -Získat řetězec

Rozhraní:

```
String GetString(String name);
```

Získejte řetězec identifikovaný názvem z těla zprávy mapování.

Parametry:**name (vstup)**

Řetězcový objekt zapouzdřující název, který identifikuje řetězec v těle zprávy mapování.

vrátí:

Řetězcový objekt zapouzdřující řetězec načtený z textu zprávy mapy. Je-li vyžadována konverze dat, bude tato hodnota řetězcem po převodu.

Výjimky:

- Výjimka XMSEException

ItemExists -Kontrola názvu-dvojice hodnot existuje.

Rozhraní:

```
Boolean ItemExists(String name);
```

Zkontrolujte, zda tělo zprávy mapování obsahuje dvojici název-hodnota s určeným názvem.

Parametry:**name (vstup)**

Řetězcový objekt zapouzdřující název dvojice název-hodnota.

vrátí:

- True, pokud text zprávy mapy obsahuje dvojici název-hodnota s uvedeným názvem.
- False, pokud tělo zprávy mapování neobsahuje dvojici název-hodnota s uvedeným názvem.

Výjimky:

- Výjimka XMSEException

SetBoolean -Nastavit logickou hodnotu

Rozhraní:

```
void SetBoolean(String name, Boolean value);
```

Nastavte logickou hodnotu v těle mapy zpráv.

Parametry:**name (vstup)**

Řetězcový objekt zapouzdřující název, který identifikuje logickou hodnotu v těle zprávy mapování.

hodnota (vstup)

Logická hodnota, která má být nastavena.

vrátí:

Void

Výjimky:

- Výjimka XMSEException

*SetByte -Nastavit bajt***Rozhraní:**

```
void SetByte(String name, Byte value);  
void SetSignedByte(String name, Int16 value);
```

Nastavte bajt v textu zprávy mapy.

Parametry:**name (vstup)**

Řetězcový objekt zapouzdřující název pro identifikaci bajtu v těle zprávy mapování.

hodnota (vstup)

Bajt, který má být nastaven.

vrátí:

Void

Výjimky:

- Výjimka XMSEException

*SetBytes -Nastavit bajty***Rozhraní:**

```
void SetBytes(String name, Byte[] value);
```

Nastavení pole bajtů v těle mapy zpráv.

Parametry:**name (vstup)**

Řetězcový objekt zapouzdřující název, aby identifikoval pole bajtů v těle zprávy mapování.

hodnota (vstup)

Pole bajtů, které mají být nastaveny.

vrátí:

Void

Výjimky:

- Výjimka XMSEException

SetChar -Nastavit znak

Rozhraní:

```
void SetChar(String name, Char value);
```

Nastavení 2-bajtového znaku v těle zprávy mapy.

Parametry:

name (vstup)

Řetězcový objekt zapouzdřující jméno, aby identifikoval znak v textu zprávy mapy.

hodnota (vstup)

Znak, který má být nastaven.

vrátí:

Void

Výjimky:

- Výjimka XMSEException

SetDouble -Nastavit číslo s pohyblivou řádovou čárkou dvojitě přesnosti

Rozhraní:

```
void SetDouble(String name, Double value);
```

Nastavte číslo s plovoucí řádovou čárkou a dvojitou přesností v textu zprávy mapy.

Parametry:

name (vstup)

Řetězcový objekt zapouzdřující název pro identifikaci čísla s plovoucí řádovou čárkou a dvojitou přesností v textu zprávy mapy.

hodnota (vstup)

Je třeba nastavit číslo s plovoucí řádovou čárkou a dvojitou přesností.

vrátí:

Void

Výjimky:

- Výjimka XMSEException

SetFloat -Nastavit počet čísel s pohyblivou řádovou čárkou

Rozhraní:

```
void SetFloat(String name, Single value);
```

Nastavte číslo s pohyblivou řádovou čárkou v textu zprávy mapy.

Parametry:

name (vstup)

Řetězcový objekt zapouzdřující název, který identifikuje číslo plovoucí řádové čárky v těle zprávy mapování.

hodnota (vstup)

Číslo plovoucí řádové čárky, které má být nastaveno.

vrátí:

Void

Výjimky:

- Výjimka XMSEException

*SetInt -Nastavit celé číslo***Rozhraní:**

```
void SetInt(String name, Int32 value);
```

Nastavit celé číslo v těle mapy zpráv.

Parametry:**name (vstup)**

Řetězcový objekt zapouzdřující název, který identifikuje celé číslo v těle zprávy mapování.

hodnota (vstup)

Celé číslo, které má být nastaveno.

vrátí:

Void

Výjimky:

- Výjimka XMSEException

*SetLong -Nastavit dlouhé celé číslo***Rozhraní:**

```
void SetLong(String name, Int64 value);
```

Nastavte dlouhé celé číslo v těle mapy zpráv.

Parametry:**name (vstup)**

Řetězcový objekt zapouzdřující název, aby identifikoval dlouhé celé číslo v těle zprávy mapování.

hodnota (vstup)

Dlouhé celé číslo, které má být nastaveno.

vrátí:

Void

Výjimky:

- Výjimka XMSEException

*SetObject -Nastavit objekt***Rozhraní:**

```
void SetObject(String name, Object value);
```

Nastavte hodnotu, která musí být primitivním typem XMS , v textu zprávy mapy.

Parametry:**name (vstup)**

Řetězcový objekt zapouzdřující jméno, aby identifikoval hodnotu v textu zprávy mapy.

hodnota (vstup)

Pole bajtů obsahující hodnotu, která má být nastavena.

vrátí:

Void

Výjimky:

- Výjimka XMSEException

SetShort -Nastavit krátké celé číslo

Rozhraní:

```
void SetShort(String name, Int16 value);
```

Nastavte krátké celé číslo v textu zprávy mapy.

Parametry:**name (vstup)**

Řetězcový objekt zapouzdřující název, aby identifikoval krátké celé číslo v těle zprávy mapování.

hodnota (vstup)

Krátké celé číslo, které má být nastaveno.

vrátí:

Void

Výjimky:

- Výjimka XMSEException

SetString -Nastavit řetězec

Rozhraní:

```
void SetString(String name, String value);
```

Nastavte řetězec v textu zprávy mapy.

Parametry:**name (vstup)**

Řetězcový objekt zapouzdřující název, aby identifikoval řetězec v těle zprávy mapování.

hodnota (vstup)

Řetězcový objekt zapouzdřující řetězec, který má být nastaven.

vrátí:

Void

Výjimky:

- Výjimka XMSEException

Zděděné vlastnosti a metody

Níže uvedené vlastnosti jsou zděděny z rozhraní [IMessage](#):

[JMSCorrelationID](#), [JMSDeliveryMode](#), [JMSDestination](#), [JMSExpiration](#), [JMSMessageID](#), [JMSPriority](#), [JMSRedelivered](#), [JMSReplyTo](#), [JMSTimestamp](#), [JMSType](#), [Properties](#)

Níže uvedené metody jsou zděděny z rozhraní [IMessage](#):

[clearBody](#), [clearProperties](#), [PropertyExists](#)

Níže uvedené metody jsou zděděny z rozhraní [IPropertyContext](#):

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

IMessage

Objekt zpráv představuje zprávu, kterou aplikace odesílá nebo přijímá. IMessage je nadtrída pro třídy zpráv, jako např. [IMapMessage](#).

Hierarchie dědičnosti:

```
IBM.XMS.IPropertyContext
|
+---- IBM.XMS.IMessage
```

Seznam polí záhlaví zpráv rozhraní JMS v objektu Zpráva najdete v tématu [“Pole záhlaví ve zprávě XMS”](#) na stránce 67. Seznam definovaných vlastností objektu Message ve službě JMS naleznete v tématu [“JMSdefinované vlastnosti zprávy”](#) na stránce 69. Seznam definovaných vlastností objektu Message (IBM) naleznete v tématu [“IBMdefinované vlastnosti zprávy”](#) na stránce 70. Seznam vlastností JMS_IBM_MQMD* pro objekt Message Object naleznete v tématu [“Vlastnosti JMS_IBM_MQMD*”](#) na stránce 186 .

Zprávy jsou odstraněny programem pro uvolnění paměti. Když je zpráva odstraněna, uvolní prostředky, které použil.

Související odkazy

[Zprávy produktu XMS](#)

Tento oddíl popisuje strukturu a obsah zpráv produktu XMS a vysvětluje, jak aplikace zpracovává zprávy produktu XMS .

Vlastnosti produktu .NET

ID GetJMSCorrelation-Získat a nastavit JMSCorrelationID

Rozhraní:

```
String JMSCorrelationID
{
    get;
    set;
}
```

Získejte a nastavte identifikátor korelace zprávy jako objekt typu String.

Výjimky:

- Výjimka [XMSException](#)

JMSDeliveryMode - Získat a nastavit JMSDeliveryMode

Rozhraní:

```
DeliveryMode JMSDeliveryMode
{
    get;
    set;
}
```

Získejte a nastavte režim doručení zprávy.

Režim doručení zprávy je jedna z následujících hodnot:

```
DeliveryMode.Persistent
DeliveryMode.NonPersistent
```

V případě nově vytvořené zprávy, která nebyla odeslána, je režim doručení `DeliveryMode.Trvalý`, s výjimkou připojení v reálném čase k zprostředkovateli, pro který je režim doručení `DeliveryMode.NonPersistent`. Pro přijatou zprávu metoda vrací režim doručení, který byl nastaven voláním `IMessageProducer.send()`, když byla zpráva odeslána, pokud přijímající aplikace nezmění režim doručení nastavením `JMSDeliveryMode`.

Výjimky:

- Výjimka `XMSEException`

JMSDestination - Získání a nastavení JMSDestination

Rozhraní:

```
IDestination JMSDestination
{
    get;
    set;
}
```

Získejte a nastavte cíl zprávy.

Místo určení je nastaveno voláním `IMessageProducer.send()`, když je zpráva odeslána. Hodnota `JMSDestination` je ignorována. Místo určení `JMSDestination` však můžete použít ke změně místa určení přijaté zprávy.

Pro nově vytvořenou zprávu, která nebyla odeslána, vrací metoda objekt místa určení s hodnotou `Null`, pokud odesílající aplikace nenastavuje místo určení nastavením `JMSDestination`. Pro přijatou zprávu metoda vrací objekt `Destination` pro cíl, který byl nastaven voláním funkce `IMessageProducer.send()`, když byla zpráva odeslána, pokud přijímající aplikace nezmění cíl nastavením `JMSDestination`.

Výjimky:

- Výjimka `XMSEException`

JMSExpiration - Získat a nastavit JMSExpiration

Rozhraní:

```
Int64 JMSExpiration
{
    get;
    set;
}
```

Získejte a nastavte čas vypršení platnosti zprávy.

Doba vypršení platnosti je nastavena voláním funkce `IMessageProducer.send ()` při odeslání zprávy. Jeho hodnota se vypočítá tak, že se přidá čas k životu, jak je uvedeno odesílající aplikací, do času odeslání zprávy. Čas vypršení platnosti je vyjádřen v milisekundách od 00:00:00 GMT k 1. lednu 1970.

Pro nově vytvořenou zprávu, která nebyla odeslána, je doba vypršení platnosti 0, pokud odesílající aplikace nenastavuje jinou dobu vypršení platnosti nastavením `JMSExpiration`. Pro přijatou zprávu metoda vrátí čas vypršení platnosti, který byl nastaven voláním funkce `IMessageProducer.send ()`, když byla zpráva odeslána, pokud přijímající aplikace nezmění čas vypršení platnosti nastavením `JMSExpiration`.

Je-li doba života rovna 0, volání `IMessageProducer.send ()` nastaví čas vypršení platnosti na 0, aby označoval, že zpráva nevyprší.

Produkt XMS vyřadí zprávy s vypršenou platností a nedoručuje je do aplikací.

Výjimky:

- Výjimka `XMSException`

JMSMessageID - Získat a nastavit JMSMessageID

Rozhraní:

```
String JMSMessageID
{
    get;
    set;
}
```

Získejte a nastavte identifikátor zprávy jako řetězcový objekt zapouzdřující identifikátor zprávy.

Identifikátor zprávy je nastaven voláním funkce `IMessageProducer.send ()`, když je zpráva odeslána. Pro přijatou zprávu vrací metoda identifikátor zprávy, který byl nastaven voláním `IMessageProducer.send ()`, když byla zpráva odeslána, pokud přijímající aplikace nezmění identifikátor zprávy nastavením `JMSMessageID`.

Pokud zpráva nemá žádný identifikátor zprávy, metoda vrátí hodnotu `null`.

Výjimky:

- Výjimka `XMSException`

JMSPriority - Získat a nastavit JMSPriority

Rozhraní:

```
Int32 JMSPriority
{
    get;
    set;
}
```

Získejte a nastavte prioritu zprávy.

Priorita je nastavena voláním funkce `IMessageProducer.send ()`, když je zpráva odeslána. Hodnota je celé číslo v rozsahu 0, nejnižší priorita, k 9, nejvyšší priorita.

Pro nově vytvořenou zprávu, která nebyla odeslána, je priorita 4, pokud odesílající aplikace nenastaví jinou prioritu nastavením parametru `JMSPriority`. Pro přijatou zprávu metoda vrací prioritu, která byla nastavena voláním funkce `IMessageProducer.send ()`, když byla zpráva odeslána, pokud přijímající aplikace nezmění prioritu nastavením parametru `JMSPriority`.

Výjimky:

- Výjimka `XMSException`

JMSRedelivered-Get a Set JMSRedelivered

Rozhraní:

```
Boolean JMSRedelivered
{
    get;
    set;
}
```

Získejte informace o tom, zda je zpráva znovu doručena, a uveďte, zda je zpráva znovu doručena. Indikace je nastavena voláním `IMessageConsumer.receive ()`, když je přijata zpráva.

Tato vlastnost má následující hodnoty:

- `True`, pokud je zpráva znovu doručena.
- `False`, pokud zpráva nebyla znovu doručena.

Pro připojení v reálném čase ke zprostředkovateli je hodnota vždy `False`.

Indikace opětovného doručení sady `JMSRedelivered` před odesláním zprávy je ignorována voláním funkce `IMessageProducer.send ()` při odeslání zprávy a je ignorována a nahrazena voláním `IMessageConsumer.receive ()`, když je zpráva přijata. Můžete však použít `JMSReverse`, abyste změnili indikaci pro přijatou zprávu.

Výjimky:

- Výjimka `XMSEException`

JMSReplyTo -Získat a nastavit JMSReplyTo

Rozhraní:

```
IDestination JMSReplyTo
{
    get;
    set;
}
```

Získejte a nastavte místo určení, kam se má odeslat odpověď na zprávu.

Hodnota této vlastnosti je cílovým objektem pro místo určení, kam má být odeslána odpověď na zprávu. Cílový objekt s hodnotou `Null` znamená, že se neočekává žádná odpověď.

Výjimky:

- Výjimka `XMSEException`

JMSTimestamp-Získat a nastavit JMSTimestamp

Rozhraní:

```
Int64 JMSTimestamp
{
    get;
    set;
}
```

Získejte a nastavte čas, kdy byla zpráva odeslána.

Časové razítko nastavuje volání `IMessageProducer.send ()`, když je zpráva odeslána a je vyjádřena v milisekundách od 00:00:00 GMT k 1. lednu 1970.

Pro nově vytvořenou zprávu, která nebyla odeslána, je časové razítko 0, pokud odesílající aplikace nenastavuje jiné časové razítko nastavením `JMSTimestamp`. Pro přijatou zprávu vrací metoda časové razítko, které bylo nastaveno voláním `IMessageProducer.send ()`, když byla zpráva odeslána, pokud přijímající aplikace nemění časové razítko nastavením `JMSTimestamp`.

Výjimky:

- Výjimka `XMSEException`

Notes:

1. Je-li časová značka nedefinovaná, metoda vrátí hodnotu 0, ale nevyvolá žádnou výjimku.

JMSType-Získat a nastavit JMSType

Rozhraní:

```
String JMSType
{
    get;
    set;
}
```

Získejte a nastavte typ zprávy.

Hodnota `JMSType` je řetězec zapouzdřující typ zprávy. Je-li požadována konverze dat, je tato hodnota typ po převodu.

Výjimky:

- Výjimka `XMSEException`

PropertyNames -Získat vlastnosti

Rozhraní:

```
System.Collections.IEnumerator PropertyNames
{
    get;
}
```

Získejte výčet názvů vlastností pro zprávu.

Výjimky:

- Výjimka `XMSEException`

Metody

Potvrdit-Potvrdit

Rozhraní:

```
void Acknowledge();
```

Potvrdit tuto zprávu a všechny dříve nepotvrzené zprávy přijaté relací.

Aplikace může volat tuto metodu, je-li režim potvrzení relace `AcknowledgeMode.ClientAcknowledge`. Volání metody jsou ignorována, pokud má relace nějaký jiný režim potvrzení nebo se jedná o transakci.

Zprávy, které byly přijaty, ale nebyly potvrzeny, mohou být znovu doručeny.

Další informace o potvrzení zpráv viz [“Potvrzení zprávy”](#) na stránce 24.

Parametry:

Není

vrátí:

Void

Výjimky:

- Výjimka `XMSEException`
- Výjimka `IllegalState`

ClearBody -Vymazat tělo

Rozhraní:

```
void ClearBody();
```

Vyčistěte tělo zprávy. Pole záhlaví a vlastnosti zprávy nejsou vymazány.

Pokud aplikace vymaže tělo zprávy, tělo zůstane ve stejném stavu jako prázdné tělo v nově vytvořené zprávě. Stav prázdného těla v nově vytvořené zprávě závisí na typu těla zprávy. Další informace naleznete v části [“Tělo zprávy produktu XMS”](#) na stránce 72.

Aplikace může kdykoli vymazat tělo zprávy, bez ohledu na to, v jakém stavu se tělo nachází. Je-li tělo zprávy jen pro čtení, jediný způsob, jak může aplikace zapisovat do těla, je aplikace, aby bylo tělo první.

Parametry:

Není

vrátí:

Void

Výjimky:

- Výjimka `XMSEException`

ClearProperties -Vymazat vlastnosti

Rozhraní:

```
void ClearProperties();
```

Vymažte vlastnosti zprávy. Pole záhlaví a tělo zprávy nejsou vymazány.

Pokud aplikace vymaže vlastnosti zprávy, vlastnosti budou přístupné pro čtení a zápis.

Aplikace může kdykoli vymazat vlastnosti zprávy, bez ohledu na to, ve kterém stavu jsou vlastnosti uvedeny. Jsou-li vlastnosti zprávy určeny pouze ke čtení, lze do aplikace zapisovat pouze vlastnosti, aby bylo možné vlastnosti nejprve vymazat.

Parametry:

Není

vrátí:

Void

Výjimky:

- Výjimka XMSEException

PropertyExists -Kontrola vlastnosti existuje

Rozhraní:

```
Boolean PropertyExists(String propertyName);
```

Zkontrolujte, zda má zpráva vlastnost s určeným názvem.

Parametry:

propertyName (vstup)

Řetězcový objekt zapouzdřující název vlastnosti.

vrátí:

- True, pokud má zpráva vlastnost s uvedeným názvem.
- False, pokud zpráva nemá vlastnost s uvedeným názvem.

Výjimky:

- Výjimka XMSEException

Zděděné vlastnosti a metody

Níže uvedené metody jsou zděděny z rozhraní [IPropertyContext](#):

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

IMessageConsumer

Aplikace používá spotřebitele zpráv k přijetí zpráv odeslaných do místa určení.

Hierarchie dědičnosti:

```
IBM.XMS.IPropertyContext
|
+---- IBM.XMS.IMessageConsumer
```

Seznam definovaných vlastností XMS objektu MessageConsumer naleznete v tématu [“Vlastnosti objektu MessageConsumer”](#) na stránce 189.

Vlastnosti produktu .NET

MessageListener -Získat a nastavit modul listener pro zprávy

Rozhraní:

```
MessageListener MessageListener
{
    get;
    set;
}
```

Získejte modul listener pro zprávy, který je registrován u spotřebitele zpráv, a zaregistrujte modul listener zpráv se spotřebitelem zpráv.

Není-li u spotřebitele zpráv zaregistrován žádný modul listener pro zprávy, parametr `MessageListener` má hodnotu `Null`. Pokud je modul listener pro zprávy již registrován se spotřebitelem zpráv, můžete registraci zrušit tím, že místo toho uvedete hodnotu `null`.

Další informace o použití listenerů zpráv viz [“Listenery zpráv a výjimek v produktu .NET”](#) na stránce 46.

Výjimky:

- Výjimka `XMSEException`

MessageSelector -Získat selektor zpráv

Rozhraní:

```
String MessageSelector
{
    get;
}
```

Získejte selektor zpráv pro spotřebitele zpráv. Návrátová hodnota je řetězcový objekt zapouzdřující výraz selektoru zpráv. Je-li požadována konverze dat, tato hodnota je výraz selektoru zpráv po převodu. Pokud spotřebitel zpráv nemá selektor zpráv, hodnota `MessageSelector` je objekt typu `String` s hodnotou `null`.

Výjimky:

- Výjimka `XMSEException`

Metody

Zavřít-Zavřít spotřebitele zpráv

Rozhraní:

```
void Close();
```

Zavřete spotřebitele zpráv.

Pokud se aplikace pokusí zavřít spotřebitele zpráv, který je již zavřen, volání se ignoruje.

Parametry:

Není

vrátí:

`Void`

Výjimky:

- Výjimka `XMSEException`

Přijmout-Přijmout

Rozhraní:

```
IMessage Receive();
```

Přijmout další zprávu pro spotřebitele zpráv. Volání čeká po neomezenou dobu na zprávu, nebo dokud nebude spotřebitel zpráv uzavřen.

Parametry:

Není

vrátí:

Ukazatel na objekt zprávy. Je-li spotřebitel zpráv zavřen, zatímco volání čeká na zprávu, metoda vrátí ukazatel na objekt Message Object s hodnotou Null.

Výjimky:

- Výjimka XMSEException

Příjem-příjem (s intervalem čekání)

Rozhraní:

```
IMessage Receive(Int64 delay);
```

Přijmout další zprávu pro spotřebitele zpráv. Volání čeká pouze určené období pro zprávu, nebo dokud není spotřebitel zpráv uzavřen.

Parametry:**delay (vstup)**

Doba (v milisekundách), po kterou volání čeká na zprávu. Uvedete-li interval čekání 0, volání čeká nekonečně dlouhou dobu na zprávu.

vrátí:

Ukazatel na objekt zprávy. Pokud během intervalu čekání nepřijde žádná zpráva, nebo pokud je spotřebitel zpráv zavřen, zatímco volání čeká na zprávu, metoda vrátí ukazatel na objekt Message Object s hodnotou null, ale nevyvolá žádnou výjimku.

Výjimky:

- Výjimka XMSEException

ReceiveNoČekání-Přijmout s nečekanou dobou čekání

Rozhraní:

```
IMessage ReceiveNoWait();
```

Přijmout další zprávu pro spotřebitele zpráv, je-li k dispozici okamžitě.

Parametry:

Není

vrátí:

Ukazatel na objekt zprávy. Není-li okamžitě k dispozici žádná zpráva, metoda vrátí ukazatel na objekt Message Object s hodnotou Null.

Výjimky:

- Výjimka XMSEException

Zděděné vlastnosti a metody

Níže uvedené metody jsou zděděny z rozhraní [IPropertyContext](#):

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#),

[SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

MessageEOFException

Hierarchie dědičnosti:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.MessageEOFException
```

XMS vyvolá tuto výjimku, pokud XMS narazí na konec proudu bajtů zpráv, když aplikace čte tělo bajtové zprávy.

Zděděné vlastnosti a metody

Níže uvedené metody jsou zděděny z rozhraní [XMSEException](#):

[GetErrorCode](#), [GetLinkedException](#)

Výjimka MessageFormat

Hierarchie dědičnosti:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.MessageFormatException
```

XMS vyvolá tuto výjimku, pokud XMS narazí na zprávu s formátem, který není platný.

Zděděné vlastnosti a metody

Níže uvedené metody jsou zděděny z rozhraní [XMSEException](#):

[GetErrorCode](#), [GetLinkedException](#)

IMessageListener (delegát)

Hierarchie dědičnosti:

Není

Aplikace používá modul listener pro zprávy k asynchronnímu příjmu zpráv.

Delegát

MessageListener - modul listener pro zprávy

Rozhraní:

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

asynchronně doručte zprávu spotřebiteli zpráv.

Metody, které implementují tohoto delegáta, mohou být zaregistrovány spolu s připojením.

Další informace o použití listenerů zpráv viz [“Listenery zpráv a výjimek v produktu .NET”](#) na stránce 46.

Parametry:

mesg (vstup)

Objekt Zpráva.

vrátí:

Void

MessageNotReadableException

Hierarchie dědičnosti:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.MessageNotReadableException
```

XMS vyvolá tuto výjimku, pokud se aplikace pokusí číst tělo zprávy, která je pouze pro zápis.

Zděděné vlastnosti a metody

Níže uvedené metody jsou zděděny z rozhraní [XMSEException](#):

[GetErrorCode](#), [GetLinkedException](#)

MessageNotWritableException

Hierarchie dědičnosti:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.MessageNotWritableException
```

XMS vyvolá tuto výjimku, pokud se aplikace pokusí o zápis do těla zprávy, která je jen pro čtení.

Zděděné vlastnosti a metody

Níže uvedené metody jsou zděděny z rozhraní [XMSEException](#):

[GetErrorCode](#), [GetLinkedException](#)

IMessageProducer

Aplikace používá producenta zpráv k odesílání zpráv do místa určení.

Hierarchie dědičnosti:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IMessageProducer
```

Seznam definovaných vlastností XMS objektu MessageProducer viz "[Vlastnosti objektu MessageProducer](#)" na stránce 189.

Vlastnosti produktu .NET

DeliveryMode - Získat a nastavit výchozí režim doručení

Rozhraní:

```
DeliveryMode DeliveryMode
```

```
{
    get;
    set;
}
```

Získejte a nastavte výchozí režim doručení pro zprávy odeslané producentem zpráv.

Výchozí režim doručení má jednu z následujících hodnot:

```
DeliveryMode.Persistent
DeliveryMode.NonPersistent
```

V případě připojení v reálném čase ke zprostředkovateli musí být hodnota `DeliveryMode.NonPersistent`.

Výchozí hodnota je `DeliveryMode.Persistent`, kromě připojení v reálném čase pro zprostředkovatele, jehož výchozí hodnota je `DeliveryMode.NonPersistent`.

Výjimky:

- Výjimka `XMSEException`

Cíl-Získat cíl

Rozhraní:

```
IDestination Destination
{
    get;
}
```

Získejte místo určení pro producenta zpráv.

Parametry:

Není

vrátí:

Cílový objekt. Pokud producent zpráv nemá místo určení, metoda vrátí objekt místa určení s hodnotou `Null`.

Výjimky:

- Výjimka `XMSEException`

ID DisableMsg-Získání a nastavení příznaku Zakázat ID zprávy

Rozhraní:

```
Boolean DisableMessageID
{
    get;
    set;
}
```

Získejte informace o tom, zda přijímající aplikace vyžaduje zahrnutí identifikátorů zpráv do zpráv odeslaných producentem zpráv, a indikuje, zda přijímající aplikace vyžaduje zahrnutí identifikátorů zpráv do zpráv odeslaných producentem zpráv.

V případě připojení ke správci front nebo v reálném čase připojení k zprostředkovateli je tento příznak ignorován. Při připojení ke sběrnici pro integraci služeb je tento parametr dodržován.

`ID DisabledMsg` má následující hodnoty:

- `True`, pokud přijímající aplikace nevyžaduje zahrnutí identifikátorů zpráv do zpráv odeslaných producentem zpráv.

- `False`, pokud přijímající aplikace vyžaduje, aby byly do zpráv odeslaných producentem zpráv zahrnuty identifikátory zpráv.

Výjimky:

- Výjimka `XMSEException`

DisableMsgTS-Získat a nastavit příznak zablokování časového razítka

Rozhraní:

```
Boolean DisableMessageTimestamp
{
    get;
    set;
}
```

Získejte informace o tom, zda přijímající aplikace vyžaduje zahrnutí časových razítek do zpráv odeslaných producentem zpráv, a indikuje, zda přijímající aplikace vyžaduje zahrnutí časových razítek do zpráv odeslaných producentem zpráv.

V reálném čase připojení k zprostředkovateli je tento příznak ignorován. V případě připojení ke správci front nebo připojení ke sběrnici pro integraci služeb je tento příznak uznán.

`DisableMsgTS` má následující hodnoty:

- `True`, pokud přijímající aplikace nevyžaduje, aby byla časová razítka zahrnuta do zpráv odeslaných producentem zpráv.
- `False`, pokud přijímající aplikace vyžaduje, aby byla časová razítka zahrnuta do zpráv odeslaných producentem zpráv.

vrátí:

Výjimky:

- Výjimka `XMSEException`

Priorita-Získat a nastavit výchozí prioritu

Rozhraní:

```
Int32 Priority
{
    get;
    set;
}
```

Získejte a nastavte výchozí prioritu pro zprávy odeslané producentem zpráv.

Hodnota výchozí priority zpráv je celé číslo v rozsahu 0, nejnižší priorita, až 9, nejvyšší priorita.

V reálném čase připojení k zprostředkovateli je priorita zprávy ignorována.

Výjimky:

- Výjimka `XMSEException`

TimeToLive-Získat a nastavit výchozí dobu na život

Rozhraní:

```
Int64 TimeToLive
```

```
{
  get;
  set;
}
```

Získejte a nastavte výchozí dobu, po kterou existuje zpráva, než vyprší platnost.

Čas je měřen od doby, kdy producent zpráv odešle zprávu a je výchozím nastavením času života v milisekundách. Hodnota 0 znamená, že platnost zprávy nikdy nevyprší.

Pro připojení v reálném čase ke zprostředkovateli je tato hodnota vždy 0.

Výjimky:

- Výjimka XMSEException

Metody

Zavřít-Producent zpráv o zavření

Rozhraní:

```
void Close();
```

Zavřete výrobce zpráv.

Pokud se aplikace pokusí zavřít producenta zpráv, který je již zavřen, volání se ignoruje.

Parametry:

Není

vrátí:

Void

Výjimky:

- Výjimka XMSEException

Odeslat-Odeslat

Rozhraní:

```
void Send(IMessage msg) ;
```

Odešle zprávu na místo určení, které bylo určeno při vytvoření producenta zprávy. Odeslat zprávu s použitím výchozího režimu doručení producenta zpráv, priority a času života.

Parametry:

msg (vstup)

Objekt Zpráva.

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- Výjimka MessageFormat
- Výjimka InvalidDestination

Odeslat-Odeslat (určení režimu doručení, priority a doby životnosti)

Rozhraní:

```
void Send(IMessage msg,  
         DeliveryMode deliveryMode,  
         Int32 priority,  
         Int64 timeToLive);
```

Odešle zprávu na místo určení, které bylo určeno při vytvoření producenta zpráv. Odeslat zprávu s použitím určeného režimu doručení, priority a času k životu.

Parametry:

msg (vstup)

Objekt Zpráva.

deliveryMode (vstup)

Režim doručení pro zprávu, který musí mít jednu z následujících hodnot:

`DeliveryMode.Persistent`

`DeliveryMode.NonPersistent`

V případě připojení v reálném čase ke zprostředkovateli musí být hodnota `DeliveryMode.NonPersistent`.

priority (vstup)

Priorita zprávy. Hodnota může být celé číslo v rozsahu 0, pro nejnižší prioritu, do 9, pro nejvyšší prioritu. V reálném čase připojení k zprostředkovateli je tato hodnota ignorována.

timeToLive (vstup)

Doba životnosti zprávy v milisekundách. Hodnota 0 znamená, že platnost zprávy nikdy nevyprší. V případě připojení v reálném čase ke zprostředkovateli musí být hodnota 0.

vrátí:

Void

Výjimky:

- Výjimka `XMSEException`
- Výjimka `MessageFormat`
- Výjimka `InvalidDestination`
- Výjimka `IllegalState`

Odeslat-Odeslat (na určené místo určení)

Rozhraní:

```
void Send(IDestination dest, IMessage msg) ;
```

Odešlete zprávu do zadaného cíle, pokud používáte producenta zpráv, pro který nebylo určeno místo určení, když byl vytvořen producent zpráv. Odeslat zprávu s použitím výchozího režimu doručení producenta zpráv, priority a času života.

Obvykle určujete cíl při vytváření producenta zpráv, ale pokud ne, musíte zadat cíl pokaždé, když odešlete zprávu.

Parametry:

dest (vstup)

Cílový objekt.

msg (vstup)

Objekt Zpráva.

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- Výjimka MessageFormat
- Výjimka InvalidDestination

Odeslat-Odeslat (na určené místo určení, určení režimu doručení, priority a času životnosti).

Rozhraní:

```
void Send(IDestination dest,  
         IMessage msg,  
         DeliveryMode deliveryMode,  
         Int32 priority,  
         Int64 timeToLive) ;
```

Odešlete zprávu do zadaného cíle, pokud používáte producenta zpráv, pro který nebylo určeno místo určení, když byl vytvořen producent zpráv. Odeslat zprávu s použitím určeného režimu doručení, priority a času k životu.

Obvykle určujete cíl při vytváření producenta zpráv, ale pokud ne, musíte zadat cíl pokaždé, když odešlete zprávu.

Parametry:**dest (vstup)**

Cílový objekt.

msg (vstup)

Objekt Zpráva.

deliveryMode (vstup)

Režim doručení pro zprávu, který musí mít jednu z následujících hodnot:

DeliveryMode.Persistent

DeliveryMode.NonPersistent

V případě připojení v reálném čase ke zprostředkovateli musí být hodnota DeliveryMode.NonPersistent.

priority (vstup)

Priorita zprávy. Hodnota může být celé číslo v rozsahu 0, pro nejnižší prioritu, do 9, pro nejvyšší prioritu. V reálném čase připojení k zprostředkovateli je tato hodnota ignorována.

timeToLive (vstup)

Doba životnosti zprávy v milisekundách. Hodnota 0 znamená, že platnost zprávy nikdy nevyprší. V případě připojení v reálném čase ke zprostředkovateli musí být hodnota 0.

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- Výjimka MessageFormat
- Výjimka InvalidDestination
- Výjimka IllegalState

Zděděné vlastnosti a metody

Níže uvedené metody jsou zděděny z rozhraní [IPropertyContext](#):

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

IOBJECTMESSAGE

Zpráva objektu je zpráva, jejíž tělo tvoří serializovaný objekt Java nebo .NET.

Hierarchie dědičnosti:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IMessage
|
+----IBM.XMS.IObjectMessage
```

Související odkazy

[Zprávy objektu](#)

Tělo zprávy objektu obsahuje serializovaný objekt Java nebo .NET .

Vlastnosti produktu .NET

Objekt-Získat a nastavit objekt jako bajty

Rozhraní:

```
System.Object Object
{
    get;
    set;
}

Byte[] GetObject();
```

Získejte a nastavte objekt, který tvoří tělo zprávy objektu.

Výjimky:

- Výjimka `XMSEException`
- `MessageNotReadableException`
- `MessageEOFException`
- `MessageNotWritableException`

Zděděné vlastnosti a metody

Níže uvedené vlastnosti jsou zděděny z rozhraní [IMessage](#):

[JMSCorrelationID](#), [JMSDeliveryMode](#), [JMSTDestination](#), [JMSEExpiration](#), [JMSPriority](#), [JMSPriority](#), [JMSRedelivered](#), [JMSReplyTo](#), [JMSTimestamp](#), [JMSType](#), [Properties](#)

Níže uvedené metody jsou zděděny z rozhraní [IMessage](#):

[clearBody](#), [clearProperties](#), [PropertyExists](#)

Níže uvedené metody jsou zděděny z rozhraní [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 je abstraktní nadtřída, která obsahuje metody pro získání a nastavení vlastností. Tyto metody jsou děděny jinými třídami.

Hierarchie dědičnosti:

Není

Metody

Vlastnost GetBoolean-Získat logickou vlastnost

Rozhraní:

```
Boolean GetBooleanProperty(String property_name);
```

Získejte hodnotu logické vlastnosti s určeným názvem.

Parametry:

název_vlastnosti (vstup)

Řetězový objekt zapouzdřující název vlastnosti.

vrátí:

Hodnota vlastnosti.

Kontext podprocesu:

Určeno podtřídou

Výjimky:

- Výjimka XMSEException

Vlastnost GetByte-Získat bajtovou vlastnost

Rozhraní:

```
Byte    GetByteProperty(String property_name) ;  
Int16   GetSignedByteProperty(String property_name) ;
```

Získejte hodnotu vlastnosti byte identifikovanou pomocí názvu.

Parametry:

název_vlastnosti (vstup)

Řetězový objekt zapouzdřující název vlastnosti.

vrátí:

Hodnota vlastnosti.

Kontext podprocesu:

Určeno podtřídou

Výjimky:

- Výjimka XMSEException

Vlastnost GetBytesProperty-Get Byte Array Property

Rozhraní:

```
Byte[] GetBytesProperty(String property_name) ;
```

Získejte hodnotu vlastnosti pole bajtů identifikované pomocí názvu.

Parametry:

název_vlastnosti (vstup)

Řetězcový objekt zapouzdřující název vlastnosti.

vrátí:

Počet bajtů v poli.

Kontext podprocesu:

Určeno podtřídou

Výjimky:

- Výjimka XMSEException

Vlastnost GetChar-získání vlastnosti znaku

Rozhraní:

```
Char GetCharProperty(String property_name) ;
```

Získejte hodnotu dvoubajtové znakové vlastnosti identifikované názvem.

Parametry:

název_vlastnosti (vstup)

Řetězcový objekt zapouzdřující název vlastnosti.

vrátí:

Hodnota vlastnosti.

Kontext podprocesu:

Určeno podtřídou

Výjimky:

- Výjimka XMSEException

Vlastnost GetDouble-získání vlastnosti Double Precision Floating Point

Rozhraní:

```
Double GetDoubleProperty(String property_name) ;
```

Získejte hodnotu vlastnosti s pohyblivou řádovou čárkou s dvojitou přesností určenou názvem.

Parametry:

název_vlastnosti (vstup)

Řetězcový objekt zapouzdřující název vlastnosti.

vrátí:

Hodnota vlastnosti.

Kontext podprocesu:

Určeno podtřídou

Výjimky:

- Výjimka XMSEException

Vlastnost GetFloat-získání vlastnosti s pohyblivou řádovou čárkou

Rozhraní:

```
Single GetFloatProperty(String property_name) ;
```

Získejte hodnotu vlastnosti s pohyblivou řádovou čárkou identifikovanou pomocí názvu.

Parametry:**název_vlastnosti (vstup)**

Řetězcový objekt zapouzdřující název vlastnosti.

vrátí:

Hodnota vlastnosti.

Kontext podprocesu:

Určeno podtřídou

Výjimky:

- Výjimka XMSEException

Vlastnost GetInt-vlastnost GetInt

Rozhraní:

```
Int32 GetIntProperty(String property_name) ;
```

Získejte hodnotu celočíselné vlastnosti identifikované pomocí názvu.

Parametry:**název_vlastnosti (vstup)**

Řetězcový objekt zapouzdřující název vlastnosti.

vrátí:

Hodnota vlastnosti.

Kontext podprocesu:

Určeno podtřídou

Výjimky:

- Výjimka XMSEException

Vlastnost GetLong-Získat vlastnost long integer

Rozhraní:

```
Int64 GetLongProperty(String property_name) ;
```

Získejte hodnotu vlastnosti typu long integer identifikovanou pomocí názvu.

Parametry:**název_vlastnosti (vstup)**

Řetězcový objekt zapouzdřující název vlastnosti.

vrátí:

Hodnota vlastnosti.

Kontext podprocesu:

Určeno podtřídou

Výjimky:

- Výjimka XMSEException

Vlastnost GetObject-Získat vlastnost objektu

Rozhraní:

```
Object GetObjectProperty( String property_name) ;
```

Získejte hodnotu a datový typ vlastnosti identifikované pomocí názvu.

Parametry:**název_vlastnosti (vstup)**

Řetězcový objekt zapouzdřující název vlastnosti.

vrátí:

Hodnota vlastnosti, která je jedním z následujících typů objektů:

Boolean
Byte
Byte[]
Char
Double
Single
Int32
Int64
Int16
String

Kontext podprocesu:

Určeno podtřídou

Výjimky:

- Výjimka XMSEException

Vlastnost GetShort-získání vlastnosti krátkých celých čísel

Rozhraní:

```
Int16 GetShortProperty(String property_name) ;
```

Získejte hodnotu vlastnosti krátkého celého čísla identifikovanou pomocí názvu.

Parametry:**název_vlastnosti (vstup)**

Řetězcový objekt zapouzdřující název vlastnosti.

vrátí:

Hodnota vlastnosti.

Kontext podprocesu:

Určeno podtřídou

Výjimky:

- Výjimka XMSEException

Vlastnost GetString-vlastnost GetString

Rozhraní:

```
String GetStringProperty(String property_name) ;
```

Získejte hodnotu řetězcové vlastnosti identifikované názvem.

Parametry:**název_vlastnosti (vstup)**

Řetězcový objekt zapouzdřující název vlastnosti.

vrátí:

Řetězcový objekt zapouzdřující řetězec, který je hodnotou vlastnosti. Je-li vyžadována konverze dat, bude tato hodnota řetězcem po převodu.

Kontext podprocesu:

Určeno podtřídou

Výjimky:

- Výjimka XMSEException

Vlastnost SetBoolean-Nastavit logickou vlastnost

Rozhraní:

```
void SetBooleanProperty( String property_name, Boolean value) ;
```

Nastavte hodnotu logické vlastnosti identifikované pomocí názvu.

Parametry:**název_vlastnosti (vstup)**

Řetězcový objekt zapouzdřující název vlastnosti.

hodnota (vstup)

Hodnota vlastnosti.

vrátí:

Void

Kontext podprocesu:

Určeno podtřídou

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

Vlastnost SetByte-Nastavit vlastnost Byte

Rozhraní:

```
void SetByteProperty( String property_name, Byte value) ;  
void SetSignedByteProperty( String property_name, Int16 value) ;
```

Nastavte hodnotu vlastnosti byte identifikovanou pomocí názvu.

Parametry:

název_vlastnosti (vstup)

Řetězcový objekt zapouzdřující název vlastnosti.

hodnota (vstup)

Hodnota vlastnosti.

vrátí:

Void

Kontext podprocesu:

Určeno podtřídou

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

SetBytesVlastnost-Nastavit vlastnost bajtového pole

Rozhraní:

```
void SetBytesProperty( String property_name, Byte[] value ) ;
```

Nastavte hodnotu vlastnosti bajtového pole identifikovaného pomocí názvu.

Parametry:

název_vlastnosti (vstup)

Řetězcový objekt zapouzdřující název vlastnosti.

hodnota (vstup)

Hodnota vlastnosti, která je polem bajtů.

vrátí:

Void

Kontext podprocesu:

Určeno podtřídou

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

Vlastnost SetChar-Nastavit vlastnost znaku

Rozhraní:

```
void SetCharProperty( String property_name, Char value) ;
```

Nastavte hodnotu dvoubajtové znakové vlastnosti identifikované názvem.

Parametry:

název_vlastnosti (vstup)

Řetězcový objekt zapouzdřující název vlastnosti.

hodnota (vstup)

Hodnota vlastnosti.

vrátí:

Void

Kontext podprocesu:

Určeno podtřídou

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

Vlastnost SetDouble-Set Double Precision Floating Point Property

Rozhraní:

```
void SetDoubleProperty( String property_name, Double value) ;
```

Nastavte hodnotu vlastnosti s plovoucí řádovou čárkou a dvojitou přesností určenou názvem.

Parametry:**název_vlastnosti (vstup)**

Řetězcový objekt zapouzdřující název vlastnosti.

hodnota (vstup)

Hodnota vlastnosti.

vrátí:

Void

Kontext podprocesu:

Určeno podtřídou

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

Vlastnost SetFloat-nastavení vlastnosti s pohyblivou řádovou čárkou

Rozhraní:

```
void SetFloatProperty( String property_name, Single value) ;
```

Nastavte hodnotu vlastnosti s plovoucí desetinnou čárkou určenou pomocí názvu.

Parametry:**název_vlastnosti (vstup)**

Řetězcový objekt zapouzdřující název vlastnosti.

hodnota (vstup)

Hodnota vlastnosti.

vrátí:

Void

Kontext podprocesu:

Určeno podtřídou

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

SetIntVlastnost-Nastavit celočíselnou vlastnost

Rozhraní:

```
void SetIntProperty( String property_name, Int32 value) ;
```

Nastavte hodnotu celočíselné vlastnosti identifikované pomocí názvu.

Parametry:

název_vlastnosti (vstup)

Řetězcový objekt zapouzdřující název vlastnosti.

hodnota (vstup)

Hodnota vlastnosti.

vrátí:

Void

Kontext podprocesu:

Určeno podtřídou

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

Vlastnost SetLong-Nastavit vlastnost long integer

Rozhraní:

```
void SetLongProperty( String property_name, Int64 value) ;
```

Nastavte hodnotu vlastnosti typu long integer identifikovanou pomocí názvu.

Parametry:

název_vlastnosti (vstup)

Řetězcový objekt zapouzdřující název vlastnosti.

hodnota (vstup)

Hodnota vlastnosti.

vrátí:

Void

Kontext podprocesu:

Určeno podtřídou

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

Vlastnost objektu SetObject-Nastavit vlastnost objektu

Rozhraní:

```
void SetObjectProperty( String property_name, Object value) ;
```

Nastavte hodnotu a datový typ vlastnosti identifikované pomocí názvu.

Parametry:**název_vlastnosti (vstup)**

Řetězcový objekt zapouzdřující název vlastnosti.

objectType (vstup)

Hodnota vlastnosti, která musí být jedna z následujících typů objektů:

Boolean
Byte
Byte[]
Char
Double
Single
Int32
Int64
Int16
String

hodnota (vstup)

Hodnota vlastnosti jako pole bajtů.

délka (vstup)

Počet bajtů v poli.

vrátí:

Void

Kontext podprocesu:

Určeno podtřídou

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

Vlastnost SetShort-Nastavit krátký celočíselný majetek

Rozhraní:

```
void SetShortProperty( String property_name, Int16 value) ;
```

Nastavte hodnotu vlastnosti krátkého celého čísla identifikovanou pomocí názvu.

Parametry:**název_vlastnosti (vstup)**

Řetězcový objekt zapouzdřující název vlastnosti.

hodnota (vstup)

Hodnota vlastnosti.

vrátí:

Void

Kontext podprocesu:

Určeno podtřídou

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

Vlastnost SetString-Nastavit vlastnost řetězce

Rozhraní:

```
void SetStringProperty( String property_name, String value);
```

Nastavte hodnotu vlastnosti řetězce identifikovanou pomocí názvu.

Parametry:

název_vlastnosti (vstup)

Řetězcový objekt zapouzdřující název vlastnosti.

hodnota (vstup)

Řetězcový objekt zapouzdřující řetězec, který je hodnotou vlastnosti.

vrátí:

Void

Kontext podprocesu:

Určeno podtřídou

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

IQueueBrowser

Aplikace používá prohlížeč front k procházení zpráv ve frontě, aniž by je odebíráte.

Hierarchie dědičnosti:

```
IBM.XMS.IPropertyContext
System.Collections.IEnumerable
|
+---- IBM.XMS.IQueueBrowser
```

Vlastnosti produktu .NET

MessageSelector -Získat selektor zpráv

Rozhraní:

```
String MessageSelector
{
    get;
}
```

Získejte selektor zpráv pro prohlížeč front.

Selektor zpráv je řetězcový objekt zapouzdřující výraz selektoru zpráv. Je-li požadována konverze dat, tato hodnota je výraz selektoru zpráv po převodu. Pokud prohlížeč front nemá selektor zpráv, metoda vrátí řetězcový objekt s hodnotou null.

Výjimky:

- Výjimka XMSEException

Fronta-Získat frontu

Rozhraní:

```
IDestination Queue
{
    get;
}
```

Získejte frontu přidruženou k prohlížeči fronty jako cílový objekt reprezentující frontu.

Výjimky:

- Výjimka XMSEException

Metody

Zavření-Zavření prohlížeče fronty

Rozhraní:

```
void Close();
```

Zavřete prohlížeč front.

Pokud se aplikace pokusí zavřít prohlížeč front, který je již zavřen, volání se ignoruje.

Parametry:

Není

vrátí:

Void

Výjimky:

- Výjimka XMSEException

GetEnumerator -Získání zpráv

Rozhraní:

```
IEnumerator GetEnumerator();
```

Získejte seznam zpráv ve frontě.

Tato metoda vrací výčtový nástroj, který zapouzdří seznam objektů zpráv. Pořadí objektů zpráv je stejné jako pořadí, ve kterém budou zprávy načítány z fronty. Aplikace pak může použít výčtový nástroj k procházení každé zprávy postupně.

výčtový nástroj se dynamicky aktualizuje, protože zprávy jsou vloženy do fronty a odebrány z fronty. Pokaždé, když aplikace zavolá `IEnumerator.MoveNext()` k procházení další zprávy ve frontě, zpráva bude odrážet aktuální obsah fronty.

Pokud aplikace volá tuto metodu více než jednou pro prohlížeč front, každý volání vrátí nový výčtový nástroj. Aplikace proto může používat více než jeden výčtový nástroj k procházení zpráv ve frontě a k udržování více pozic ve frontě.

Parametry:

Není

vrátí:

Objekt iterátoru.

Výjimky:

- Výjimka XMSEException

Zděděné vlastnosti a metody

Níže uvedené metody jsou zděděny z rozhraní [IPropertyContext](#):

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

Žadatel

Aplikace používá žadatele k odeslání zprávy požadavku a následné čekání na odpověď a přijetí odpovědi.

Hierarchie dědičnosti:

Není

Konstruktory

Žadatel-Vytvořit žadatele

Rozhraní:

```
Requestor(ISession sess, IDestination dest);
```

Vytvořte žadatele.

Parametry:

sess (vstup)

Objekt relace. Relace nesmí být zpracovávána a musí mít jeden z následujících režimů potvrzení:

`AcknowledgeMode.AutoAcknowledge`

`AcknowledgeMode.DupsOkAcknowledge`

dest (vstup)

Cílový objekt reprezentující místo určení, kam může aplikace odesílat zprávy požadavků.

Kontext podprocesu:

Relace přidružená k žadateli

Výjimky:

- Výjimka XMSEException

Metody

Zavřít-Zavřený žadatel

Rozhraní:

```
void Close();
```

Zavřete žadatele.

Pokud se aplikace pokusí zavřít žadatele, který je již zavřen, volání se ignoruje.

Poznámka: Když aplikace uzavře žadatele, přidružená relace se nezavře také. V tomto ohledu se produkt XMS chová odlišně v porovnání s platformou JMS.

Parametry:

Není

vrátí:

Void

Kontext podprocesu:

Libovolný

Výjimky:

- Výjimka XMSEException

Požadavek-odezva požadavku

Rozhraní:

```
IMessage Request(IMessage requestMessage);
```

Odešlete zprávu s požadavkem a poté vyčkejte a obdržíte odpověď od aplikace, která přijímá zprávu požadavku.

Volání této metody bloků, dokud není přijata odpověď nebo dokud relace neskončí, podle toho, co nastane dříve.

Parametry:

requestMessage (vstup)

Objekt Message shrnující zprávu požadavku.

vrátí:

Ukazatel na objekt Message shrnující zprávu odpovědi.

Kontext podprocesu:

Relace přidružená k žadateli

Výjimky:

- Výjimka XMSEException

Výjimka ResourceAllocation

Hierarchie dědičnosti:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.ResourceAllocationException
```

XMS vyvolá tuto výjimku, pokud XMS nemůže přidělit prostředky požadované metodou.

Zděděné vlastnosti a metody

Níže uvedené metody jsou zděděny z rozhraní XMSEException:

GetErrorCode, GetLinkedException

SecurityException

Hierarchie dědičnosti:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
      |
      +----IBM.XMS.SecurityException
```

XMS vyvolá tuto výjimku, je-li identifikátor uživatele a heslo poskytnuté pro ověření aplikace odmítnuty. XMS také vyvolá tuto výjimku, pokud kontrola oprávnění selže a zabrání dokončení metody.

Zděděné vlastnosti a metody

Níže uvedené metody jsou zděděny z rozhraní [XMSEException](#):

[GetErrorCode](#), [GetLinkedException](#)

ISession

Relace je jednovláknový kontext pro odesílání a příjem zpráv.

Hierarchie dědičnosti:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.ISession
```

Seznam definovaných vlastností objektu relace XMS naleznete v tématu "[Vlastnosti relace](#)" na stránce 189.

Vlastnosti .NET

AcknowledgeMode - Získat režim potvrzení

Rozhraní:

```
AcknowledgeMode AcknowledgeMode
{
    get;
}
```

Získejte režim potvrzení pro relaci.

Režim potvrzení je zadán při vytvoření relace.

V případě, že relace není prováděna, je režim potvrzení jedním z následujících hodnot:

```
AcknowledgeMode.AutoAcknowledge
AcknowledgeMode.ClientAcknowledge
AcknowledgeMode.DupsOkAcknowledge
```

Další informace o režimech potvrzení naleznete v tématu "[Potvrzení zprávy](#)" na stránce 24.

Relace, která transakci obsahuje, nemá žádný režim potvrzení. Pokud relace obsahuje transakci, metoda vrací místo toho `AcknowledgeMode.SessionTransacted`.

Výjimky:

- Výjimka `XMSEException`

Transacted-Určit, zda se transakce provádí

Rozhraní:

```
Boolean Transacted
{
    get;
}
```

Určete, zda relace obsahuje transakci.

Uzpracovávána transakce je:

- `True`, je-li relace zpracovávána.
- `False`, pokud relace není zpracovávána.

V případě připojení v reálném čase ke zprostředkovateli metoda vždy vrátí hodnotu `False`.

Výjimky:

- Výjimka `XMSEException`

Metody

Zavřít-Zavřít relaci

Rozhraní:

```
void Close();
```

Zavřete relaci. Je-li relace zpracovávána, transakce s probíhání probíhají bude odvolána.

Pokud se aplikace pokusí zavřít relaci, která je již zavřena, volání se ignoruje.

Parametry:

Není

vrátí:

Void

Kontext podprocesu:

Libovolný

Výjimky:

- Výjimka `XMSEException`

Potvrdit-Potvrdit

Rozhraní:

```
void Commit();
```

Potvrdit všechny zprávy zpracované v aktuální transakci.

Relace musí být relací transakcí.

Parametry:

Není

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- Výjimka IllegalState
- TransactionRolledBackException

CreateBrowser -Vytvoření prohlížeče front

Rozhraní:

```
IQueueBrowser CreateBrowser(IDestination queue) ;
```

Vytvořte prohlížeč front pro uvedenou frontu.

Parametry:

queue (vstup)

Cílový objekt reprezentující frontu.

vrátí:

Objekt QueueBrowser .

Výjimky:

- Výjimka XMSEException
- Výjimka InvalidDestination

CreateBrowser -Vytvoření prohlížeče front (se selektorem zpráv)

Rozhraní:

```
IQueueBrowser CreateBrowser(IDestination queue, String selector) ;
```

Vytvořte prohlížeč front pro uvedenou frontu pomocí selektoru zpráv.

Parametry:

queue (vstup)

Cílový objekt reprezentující frontu.

selektor (vstup)

Řetězcový objekt zapouzdřující výraz selektoru zpráv. Do prohlížeče fronty se doručí pouze zprávy s vlastnostmi, které se shodují s výrazem selektoru zpráv.

Objekt s hodnotou null String znamená, že pro prohlížeč front neexistuje žádný selektor zpráv.

vrátí:

Objekt QueueBrowser .

Výjimky:

- Výjimka XMSEException
- Výjimka InvalidDestination
- Výjimka InvalidSelector

Zpráva CreateBytesZpráva-Vytvořit bajtovou zprávu

Rozhraní:

```
IBytesMessage CreateBytesMessage();
```

Vytvořte bajtovou zprávu.

Parametry:

Není

vrátí:

Objekt BytesMessage .

Výjimky:

- Výjimka XMSEException
- Výjimka IllegalState(relace je zavřena)

CreateConsumer -Vytvořit odběratele

Rozhraní:

```
IMessageConsumer CreateConsumer(IDestination dest) ;
```

Vytvořte spotřebitele zpráv pro určené místo určení.

Parametry:

dest (vstup)

Cílový objekt.

vrátí:

Objekt MessageConsumer .

Výjimky:

- Výjimka XMSEException
- Výjimka InvalidDestination

CreateConsumer -Vytvoření odběratele (s selektorem zpráv)

Rozhraní:

```
IMessageConsumer CreateConsumer(IDestination dest,  
                                String selector) ;
```

Vytvořte spotřebitele zpráv pro určené místo určení pomocí selektoru zpráv.

Parametry:

dest (vstup)

Cílový objekt.

selektor (vstup)

Řetězcový objekt zapouzdřující výraz selektoru zpráv. Do konzumentu zpráv jsou doručeny pouze zprávy s vlastnostmi, které se shodují s výrazem selektoru zpráv.

Objekt s hodnotou null String znamená, že pro spotřebitele zpráv neexistuje žádný selektor zpráv.

vrátí:

Objekt MessageConsumer .

Výjimky:

- Výjimka XMSEException
- Výjimka InvalidDestination
- Výjimka InvalidSelector

CreateConsumer -Vytvoření odběratele (s selektorem zpráv a příznakem lokální zprávy)

Rozhraní:

```
IMessageConsumer CreateConsumer(IDestination dest,  
                                String selector,  
                                Boolean noLocal) ;
```

Vytvořte spotřebitele zpráv pro určené místo určení pomocí selektoru zpráv a v případě, že cílem je téma, určete, zda spotřebitel zpráv přijímá zprávy publikované pomocí vlastního připojení.

Parametry:

dest (vstup)

Cílový objekt.

selektor (vstup)

Řetězcový objekt zapouzdřující výraz selektoru zpráv. Do konzumentu zpráv jsou doručeny pouze zprávy s vlastnostmi, které se shodují s výrazem selektoru zpráv.

Objekt s hodnotou null String znamená, že pro spotřebitele zpráv neexistuje žádný selektor zpráv.

noLocal (vstup)

Hodnota True znamená, že spotřebitel zpráv nepřijímá zprávy publikované vlastním připojením. Hodnota False znamená, že spotřebitel zpráv přijímá zprávy publikované svým vlastním připojením. Výchozí hodnota je False.

vrátí:

Objekt MessageConsumer .

Výjimky:

- Výjimka XMSEException
- Výjimka InvalidDestination
- Výjimka InvalidSelector

CreateDurableOdběratel-Vytvořit trvalý odběratel

Rozhraní:

```
IMessageConsumer CreateDurableSubscriber(IDestination dest,  
                                         String subscription) ;
```

Vytvořte trvalého odběratele pro určené téma.

Tato metoda není platná pro připojení v reálném čase ke zprostředkovateli.

Další informace o trvalých odběratelích naleznete v tématu [“Trvalí odběratelé”](#) na stránce 32.

Parametry:

dest (vstup)

Cílový objekt reprezentující téma. Téma nesmí být dočasné téma.

odběr (vstup)

Řetězcový objekt zapouzdřující název, který identifikuje trvalý odběr. Název musí být jedinečný v rámci identifikátoru klienta pro připojení.

vrátí:

Objekt MessageConsumer reprezentující trvalého odběratele.

Výjimky:

- Výjimka XMSEException
- Výjimka InvalidDestination

CreateDurableOdběratel-Vytvořit trvalý odběratel (se selektorem zpráv a lokálním příznakem zprávy)

Rozhraní:

```
IMessageConsumer CreateDurableSubscriber(IDestination dest,  
                                         String subscription,  
                                         String selector,  
                                         Boolean noLocal) ;
```

Vytvořte trvalého odběratele pro určené téma pomocí selektoru zpráv a určením, zda trvalý odběratel přijímá zprávy publikované pomocí vlastního připojení.

Tato metoda není platná pro připojení v reálném čase ke zprostředkovateli.

Další informace o trvalých odběratelích naleznete v tématu [“Trvalí odběratelé”](#) na stránce 32.

Parametry:**dest (vstup)**

Cílový objekt reprezentující téma. Téma nesmí být dočasné téma.

odběr (vstup)

Řetězcový objekt zapouzdřující název, který identifikuje trvalý odběr. Název musí být jedinečný v rámci identifikátoru klienta pro připojení.

selektor (vstup)

Řetězcový objekt zapouzdřující výraz selektoru zpráv. Do trvalého odběratele jsou doručovány pouze zprávy s vlastnostmi, které se shodují s výrazem selektoru zpráv.

Řetězcový objekt s hodnotou Null znamená, že pro trvalého odběratele neexistuje žádný selektor zpráv.

noLocal (vstup)

Hodnota True znamená, že trvalý odběratel nepřijímá zprávy publikované svým vlastním připojením. Hodnota False znamená, že trvalý odběratel přijímá zprávy publikované svým vlastním připojením. Výchozí hodnota je False.

vrátí:

Objekt MessageConsumer reprezentující trvalého odběratele.

Výjimky:

- Výjimka XMSEException
- Výjimka InvalidDestination
- Výjimka InvalidSelector

Zpráva *CreateMap*-Vytvořit zprávu mapování

Rozhraní:

```
IMapMessage CreateMapMessage();
```

Vytvořte zprávu mapování.

Parametry:

Není

vrátí:

Objekt MapMessage .

Výjimky:

- Výjimka XMSEException
- Výjimka IllegalState(relace je zavřena)

CreateMessage -Vytvořit zprávu

Rozhraní:

```
IMessage CreateMessage();
```

Vytvořte zprávu, která nemá tělo.

Parametry:

Není

vrátí:

Objekt Zpráva.

Výjimky:

- Výjimka XMSEException
- Výjimka IllegalState(relace je zavřena)

Zpráva *CreateObject*-Vytvoření zprávy objektu

Rozhraní:

```
IObjectMessage CreateObjectMessage();
```

Vytvořte zprávu objektu.

Parametry:

Není

vrátí:

Objekt ObjectMessage .

Výjimky:

- Výjimka XMSEException
- Výjimka IllegalState(relace je zavřena)

CreateProducer -Vytvoření producenta

Rozhraní:

```
IMessageProducer CreateProducer(IDestination dest) ;
```

Vytvořte producenta zpráv pro odesílání zpráv do zadaného místa určení.

Parametry:

dest (vstup)

Cílový objekt.

Určíte-li objekt místa určení s hodnotou Null, bude producent zpráv vytvořen bez místa určení. V takovém případě musí aplikace určit místo určení vždy, když k odeslání zprávy použije producenta zpráv.

vrátí:

Objekt MessageProducer .

Výjimky:

- Výjimka XMSEException
- Výjimka InvalidDestination

CreateQueue -Vytvořit frontu

Rozhraní:

```
IDestination CreateQueue(String queue) ;
```

Vytvořte objekt místa určení, který bude reprezentovat frontu na serveru systému zpráv.

Tato metoda nevytvoří frontu na serveru systému zpráv. Frontu musíte vytvořit dříve, než bude moci aplikace volat tuto metodu.

Parametry:

queue (vstup)

Řetězcový objekt zapouzdřující název fronty nebo zapouzdření identifikátoru URI (Uniform Resource Identifier), který identifikuje frontu.

vrátí:

Cílový objekt reprezentující frontu.

Výjimky:

- Výjimka XMSEException

Zpráva CreateStream-Vytvořit proudovou zprávu

Rozhraní:

```
IStreamMessage CreateStreamMessage();
```

Vytvořte zprávu proudu.

Parametry:

Není

vrátí:

Objekt StreamMessage .

Výjimky:

- Výjimka XMSEException
- VÝJIMKA XMS_ILLEGAL_STATE_EXCEPTION

*CreateTemporaryQueue-Vytvoření dočasné fronty***Rozhraní:**

```
IDestination CreateTemporaryQueue() ;
```

Vytvořte dočasnou frontu.

Obor dočasné fronty je připojení. Pouze relace vytvořené připojením mohou použít dočasnou frontu.

Dočasná fronta zůstává, dokud není explicitně odstraněna, nebo dokud se připojení neukončí, podle toho, co nastane dříve.

Další informace o dočasných frontách viz [“Dočasná místa určení”](#) na stránce 30.

Parametry:

Není

vrátí:

Cílový objekt reprezentující dočasnou frontu.

Výjimky:

- Výjimka XMSEException

*Téma CreateTemporary-Vytvořit dočasné téma***Rozhraní:**

```
IDestination CreateTemporaryTopic() ;
```

Vytvořte dočasné téma.

Rozsah dočasného tématu je připojení. Pouze relace vytvořené připojením mohou používat dočasné téma.

Dočasné téma zůstává do okamžiku, kdy je explicitně odstraněno, nebo pokud připojení skončí, podle toho, co nastane dříve.

Další informace o dočasných tématech naleznete v tématu [“Dočasná místa určení”](#) na stránce 30.

Parametry:

Není

vrátí:

Cílový objekt reprezentující dočasné téma.

Výjimky:

- Výjimka XMSEException

*CreateTextZpráva-Vytvořit textovou zprávu***Rozhraní:**

```
ITextMessage CreateTextMessage();
```

Vytvořte textovou zprávu s prázdným tělem.

Parametry:

Není

vrátí:

Objekt TextMessage .

Výjimky:

- Výjimka XMSEException

CreateTextZpráva -Vytvořit textovou zprávu (inicializováno)

Rozhraní:

```
ITextMessage CreateTextMessage(String initialValue);
```

Vytvořte textovou zprávu, jejíž tělo je inicializováno s uvedeným textem.

Parametry:

initialValue (vstup)

Objekt typu String, který zapouzdřuje text, aby inicializoval tělo textové zprávy.

Není

vrátí:

Objekt TextMessage .

Výjimky:

- Výjimka XMSEException

CreateTopic -Vytvořit téma

Rozhraní:

```
IDestination CreateTopic(String topic) ;
```

Vytvoření cílového objektu pro znázornění tématu.

Parametry:

topic (vstup)

Řetězcový objekt zapouzdřující název tématu nebo zapouzdření identifikátoru URI (Uniform Resource Identifier), který identifikuje dané téma.

vrátí:

Cílový objekt reprezentující téma.

Výjimky:

- Výjimka XMSEException

Obnova-obnova

Rozhraní:

```
void Recover();
```

Obnovte relaci. Doručování zpráv je zastaveno a pak restartováno s nejstarší nepotvrzenou zprávou.

Relace nesmí být relací s transakcemi.

Další informace o obnově relace najdete v tématu [“Potvrzení zprávy”](#) na stránce 24.

Parametry:

Není

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- Výjimka IllegalState

Odvolání-odvolání

Rozhraní:

```
void Rollback();
```

Odvolejte všechny zprávy zpracované v aktuální transakci.

Relace musí být relací transakcí.

Parametry:

Není

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- Výjimka IllegalState

Zrušit odběr-Zrušit odběr

Rozhraní:

```
void Unsubscribe(String subscription);
```

Odstraňte trvalý odběr. Server systému zpráv odstraní záznam o trvalém odběru, který spravuje, a neodešle další zprávy trvalému odběrateli.

Aplikace nemůže odstranit trvalý odběr za následujících okolností:

- Existuje aktivní spotřebitel zpráv pro trvalý odběr
- Spotřebovaná zpráva je součástí nevyřízené transakce.
- Spotřebovaná zpráva nebyla potvrzena

Tato metoda není platná pro připojení v reálném čase ke zprostředkovateli.

Parametry:

odběr (vstup)

Řetězcový objekt zapouzdřující název, který identifikuje trvalý odběr.

vrátí:

Void

Výjimky:

- Výjimka XMSEException

- Výjimka `InvalidDestination`
- Výjimka `IllegalState`

Zděděné vlastnosti a metody

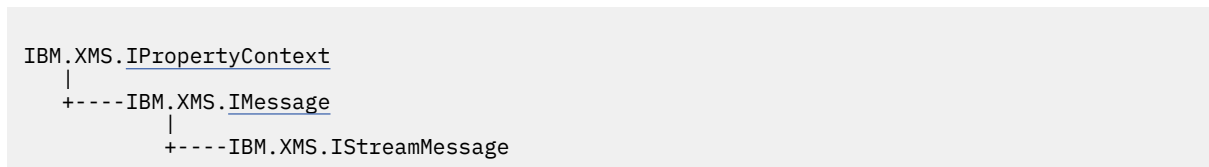
Níže uvedené metody jsou zděděny z rozhraní `IPropertyContext`:

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

IStreamMessage

Proudová zpráva je zpráva, jejíž tělo tvoří proud hodnot, kde má každá hodnota přidružený datový typ. Obsah těla se zapisuje a čte postupně.

Hierarchie dědičnosti:



Když aplikace načte hodnotu z proudu zpráv, může ji převést hodnota XMS do jiného datového typu. Další informace o této formě implicitní konverze viz [“Zprávy proudů”](#) na stránce 76.

Související odkazy

[Zprávy proudů](#)

Tělo zprávy s proudem hodnot obsahuje proud hodnot, kde má každá hodnota přidružený datový typ.

Metody

ReadBoolean - Čtení hodnoty typu `Boolean`

Rozhraní:

```
Boolean ReadBoolean();
```

Přečtete si logickou hodnotu z proudu zpráv.

Parametry:

Není

vrátí:

Logická hodnota, která je přečtená.

Výjimky:

- Výjimka `XMSEException`
- `MessageNotReadableException`
- `MessageEOFException`

ReadByte - Čtení bajtů

Rozhraní:

```
Int16  ReadSignedByte();  
Byte   ReadByte();
```

Přečtete si 8bitové celé číslo z proudu zpráv.

Parametry:

Není

vrátí:

Bajt, který je čten.

Výjimky:

- Výjimka `XMSEException`
- `MessageNotReadableException`
- `MessageEOFException`

ReadBytes - Čtení bajtů

Rozhraní:

```
Int32  ReadBytes(Byte[] array);
```

Přečtete si pole bajtů z proudu zpráv.

Parametry:

array (vstup)

Vyrovňovací paměť obsahující pole bajtů, které se čtou, a délku vyrovnávací paměti v bajtech.

Je-li počet bajtů v poli menší nebo roven délce vyrovnávací paměti, celé pole se načte do vyrovnávací paměti. Je-li počet bajtů v poli větší než délka vyrovnávací paměti, je vyrovnávací paměť zaplněna částí pole a vnitřní kurzor označuje pozici dalšího bajtu, který se má přečíst. Následující volání do `readBytes()` čte bajty z pole od aktuální pozice kurzoru.

Uvedete-li na vstupu ukazatel `null`, přeskočí volání přes pole bajtů, aniž by jej četl.

vrátí:

Počet bajtů, které jsou načteny do vyrovnávací paměti. Je-li vyrovnávací paměť částečně vyplněna, hodnota je menší než délka vyrovnávací paměti, což znamená, že v poli, které zbývá přečíst, nejsou žádné další bajty. Pokud před voláním nezbývají žádné bajty, které by bylo možné přečíst z pole, hodnota je `XMSE_END_OF_BYTEARRAY`.

Uvedete-li na vstupu ukazatel `null`, metoda nevrátí žádnou hodnotu.

Výjimky:

- Výjimka `XMSEException`
- `MessageNotReadableException`
- `MessageEOFException`

ReadChar - Čtení znaků

Rozhraní:

```
Char   ReadChar();
```

Čtení 2-bajtového znaku z proudu zpráv.

Parametry:

Není

vrátí:

Znak, který se čte.

Výjimky:

- Výjimka XMSEException
- MessageNotReadableException
- MessageEOFException

ReadDouble -Přečíst číslo s pohyblivou řádovou čárkou s dvojitou přesností

Rozhraní:

```
Double ReadDouble();
```

Přečtěte si 8bajtové číslo s pohyblivou řádovou čárkou o dvojnásobné přesnosti z proudu zpráv.

Parametry:

Není

vrátí:

Číslo s plovoucí řádovou čárkou a dvojitou přesností, které je přečteno.

Výjimky:

- Výjimka XMSEException
- MessageNotReadableException
- MessageEOFException

ReadFloat -Čtení čísla s pohyblivou řádovou čárkou

Rozhraní:

```
Single ReadFloat();
```

Přečtěte si 4bajtové číslo s pohyblivou řádovou čárkou z proudu zpráv.

Parametry:

Není

vrátí:

Číslo v plovoucí řádové čárce, které se čte.

Výjimky:

- Výjimka XMSEException
- MessageNotReadableException
- MessageEOFException

ReadInt -Přečíst celé číslo

Rozhraní:

```
Int32 ReadInt();
```

Čtení 32bitového celého čísla z proudu zpráv.

Parametry:

Není

vrátí:

Celé číslo, které je přečteno.

Výjimky:

- Výjimka XMSEException
- MessageNotReadableException
- MessageEOFException

ReadLong -Číst dlouhé celé číslo

Rozhraní:

```
Int64 ReadLong();
```

Přečtěte si 64bitové celé číslo se znaménkem, které je ze zprávy proudu.

Parametry:

Není

vrátí:

Dlouhé celé číslo, které je přečteno.

Výjimky:

- Výjimka XMSEException
- MessageNotReadableException
- MessageEOFException

ReadObject -Číst objekt

Rozhraní:

```
Object ReadObject();
```

Přečtěte si hodnotu z proudu zpráv a vraťte její datový typ.

Parametry:

Není

vrátí:

Hodnota, která je jedním z následujících typů objektů:

- Boolean
- Byte
- Byte[]
- Char
- Double

Single
Int32
Int64
Int16
String

Výjimky:

Výjimka XMSEException

ReadShort -Přečíst krátké celé číslo

Rozhraní:

```
Int16 ReadShort();
```

Přečíst podepsané 16bitové celé číslo z proudu zpráv.

Parametry:

Není

vrátí:

Krátké celé číslo, které je přečteno.

Výjimky:

- Výjimka XMSEException
- MessageNotReadableException
- MessageEOFException

ReadString -Číst řetězec

Rozhraní:

```
String ReadString();
```

Číst řetězec z proudu zpráv. Je-li to nutné, produkt XMS převede znaky v řetězci na lokální kódovou stránku.

Parametry:

Není

vrátí:

Řetězcový objekt zapouzdřující řetězec, který se čte. Je-li požadován převod dat, je tento řetězec po převodu.

Výjimky:

- Výjimka XMSEException
- MessageNotReadableException
- MessageEOFException

Reset-Reset

Rozhraní:

```
void Reset();
```

Umístěte tělo zprávy do režimu jen pro čtení a přesuňte kurzor na začátek toku zpráv.

Parametry:

Není

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- MessageNotReadableException
- MessageEOFException

WriteBoolean -Zapsat logickou hodnotu

Rozhraní:

```
void WriteBoolean(Boolean value);
```

Napište logickou hodnotu do proudu zpráv.

Parametry:

hodnota (vstup)

Logická hodnota, která má být zapsána.

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

WriteByte -Zápis bajtu

Rozhraní:

```
void WriteByte(Byte value);  
void WriteSignedByte(Int16 value);
```

Napište bajt do proudu zpráv.

Parametry:

hodnota (vstup)

Bajt, který se má zapsat.

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

WriteBytes -Zápis bajtů

Rozhraní:

```
void WriteBytes(Byte[] value);
```

Zapisovat pole bajtů do proudu zpráv.

Parametry:

hodnota (vstup)

Pole bajtů, které mají být zapsány.

délka (vstup)

Počet bajtů v poli.

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

WriteChar -Zápis znaku

Rozhraní:

```
void WriteChar(Char value);
```

Napište znak do proudu zpráv jako 2 bajty, bajt s velkým pořadovým číslem jako první.

Parametry:

hodnota (vstup)

Znak, který má být zapsán.

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

WriteDouble -Write Double Precision Floating Point Number

Rozhraní:

```
void WriteDouble(Double value);
```

Převeďte číslo s pohyblivou řádovou čárkou a dvojitou přesností na dlouhé celé číslo a zapište dlouhé celé číslo do toku zpráv jako 8 bajtů, bajt s velkým pořadovým číslem jako první.

Parametry:

hodnota (vstup)

Číslo s plovoucí řádovou čárkou s dvojitou přesností, které se má zapsat.

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

WriteFloat -Zápis čísla s plovoucí řádovou čárkou

Rozhraní:

```
void WriteFloat(Single value);
```

Převeďte číslo s plovoucí řádovou čárkou na celé číslo a zapište celé číslo do proudu zpráv jako 4 bajty, bajt s velkým pořadovým číslem jako první.

Parametry:**hodnota (vstup)**

Číslo plovoucí řádové čárky, které se má zapsat.

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

WriteInt -Zapsat celé číslo

Rozhraní:

```
void WriteInt(Int32 value);
```

Zapište celé číslo do proudu zpráv jako 4 bajty, bajt s velkým pořadovým číslem jako první.

Parametry:**hodnota (vstup)**

Celé číslo, které má být zapsáno.

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

WriteLong -zapište dlouhé celé číslo

Rozhraní:

```
void WriteLong(Int64 value);
```

Zapište dlouhé celé číslo do toku zpráv jako 8 bajtů, bajt s velkým pořadovým číslem jako první.

Parametry:**hodnota (vstup)**

Dlouhé celé číslo, které má být zapsáno.

vrátí:

Void

Výjimky:

- Výjimka XMSEException
- MessageNotWritableException

*WriteObject -Zapsat objekt***Rozhraní:**

```
void WriteObject(Object value);
```

Zadejte hodnotu do proudu zpráv se zadaným datovým typem.

Parametry:**objectType (vstup)**

Hodnota, která musí být jedním z následujících typů objektů:

- Boolean
- Byte
- Byte[]
- Char
- Double
- Single
- Int32
- Int64
- Int16
- String

hodnota (vstup)

Pole bajtů obsahující hodnotu, která má být zapsána.

délka (vstup)

Počet bajtů v poli.

vrátí:

Void

Výjimky:

- Výjimka XMSEException

*WriteShort -zapište krátké celé číslo***Rozhraní:**

```
void WriteShort(Int16 value);
```

Napište krátké celé číslo do toku zpráv jako 2 bajty, bajt s velkým pořadovým číslem jako první.

Parametry:**hodnota (vstup)**

Krátké celé číslo, které má být zapsáno.

vrátí:

Void

Výjimky:

- Výjimka `XMSEException`
- `MessageNotWritableException`

`WriteString` -řetězec zápisu.

Rozhraní:

```
void WriteString(String value);
```

Napište řetězec do proudu zpráv.

Parametry:

hodnota (vstup)

Řetězcový objekt zapouzdřující řetězec, který má být zapsán.

vrátí:

Void

Výjimky:

- Výjimka `XMSEException`
- `MessageNotWritableException`

Zděděné vlastnosti a metody

Níže uvedené vlastnosti jsou zděděny z rozhraní `IMessage`:

[JMSCorrelationID](#), [JMSDeliveryMode](#), [JMSDestination](#), [JMSExpiration](#), [JMSMessageID](#), [JMSPriority](#), [JMSRedelivered](#), [JMSReplyTo](#), [JMSTimestamp](#), [JMSType,Properties](#)

Níže uvedené metody jsou zděděny z rozhraní `IMessage`:

[clearBody](#), [clearProperties](#), [PropertyExists](#)

Níže uvedené metody jsou zděděny z rozhraní `IPropertyContext`:

[GetBooleanProperty](#), [GetByteProperty](#), [GetBytesProperty](#), [GetCharProperty](#), [GetDoubleProperty](#), [GetFloatProperty](#), [GetIntProperty](#), [GetLongProperty](#), [GetObjectProperty](#), [GetShortProperty](#), [GetStringProperty](#), [SetBooleanProperty](#), [SetByteProperty](#), [SetBytesProperty](#), [SetCharProperty](#), [SetDoubleProperty](#), [SetFloatProperty](#), [SetIntProperty](#), [SetLongProperty](#), [SetObjectProperty](#), [SetShortProperty](#), [SetStringProperty](#)

ITextMessage

Textová zpráva je zpráva, jejíž tělo tvoří řetězec.

Hierarchie dědičnosti:

```
IBM.XMS.IPropertyContext
|
+---- IBM.XMS.IMessage
|
+---- IBM.XMS.ITextMessage
```

Související odkazy

[Textové zprávy](#)

Tělo textové zprávy obsahuje řetězec.

Vlastnosti produktu .NET

Text-Získat a nastavit text

Rozhraní:

```
String Text
{
    get;
    set;
}
```

Získejte a nastavte řetězec, který tvoří tělo textové zprávy.

Je-li to nutné, produkt XMS převede znaky v řetězci na lokální kódovou stránku.

Výjimky:

- Výjimka XMSEException
- MessageNotReadableException
- MessageNotWritableException
- MessageEOFException

Zděděné vlastnosti a metody

Níže uvedené vlastnosti jsou zděděny z rozhraní IMessage:

JMSCorrelationID, JMSDeliveryMode, JMSDestination, JMSExpiration, JMSMessageID, JMSPriority, JMSRedelivered, JMSReplyTo, JMSTimestamp, JMSType, Properties

Níže uvedené metody jsou zděděny z rozhraní IMessage:

clearBody, clearProperties, PropertyExists

Níže uvedené metody jsou zděděny z rozhraní IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

TransactionInProgressException

Hierarchie dědičnosti:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.TransactionInProgressException
```

XMS vyvolá tuto výjimku, pokud aplikace požádá o operaci, která není platná, protože probíhá transakce.

Zděděné vlastnosti a metody

Níže uvedené metody jsou zděděny z rozhraní XMSEException:

GetErrorCode, GetLinkedException

TransactionRolledBackException

Hierarchie dědičnosti:

```
IBM.XMS.XMSEException
```

```

|
+----IBM.XMS.XMSEException
      |
      +----IBM.XMS.TransactionRolledBackException

```

XMS vyvolá tuto výjimku, pokud aplikace volá `Session.commit()` k potvrzení aktuální transakce, ale transakce se pak vrátí zpět.

Zděděné vlastnosti a metody

Níže uvedené metody jsou zděděny z rozhraní `XMSEException`:

`GetErrorCode`, `GetLinkedException`

Výjimka XMSEException

Pokud XMS zjistí chybu během zpracování volání metody `.NET`, příkaz XMS vyvolá výjimku. Výjimkou je objekt, který zapouzdřuje informace o chybě.

Hierarchie dědičnosti:

```

System.Exception
|
+----IBM.XMS.XMSEException

```

Existují různé typy výjimek XMS a objekt `XMSEException` je pouze jedním typem výjimky. Třída `XMSEException` je však nadtřídou jiných tříd výjimek XMS. XMS generuje objekt `XMSEException` v situacích, kdy není vhodný žádný z jiných typů výjimek.

Vlastnosti produktu .NET

ErrorCode - Získat kód chyby

Rozhraní:

```

public String ErrorCode
{
    get {return errorCode_;}
}

```

Získejte kód chyby.

Výjimky:

- Výjimka `XMSEException`

LinkedException - Získat propojené výjimky

Rozhraní:

```

public Exception LinkedException
{
    get { return linkedException_;}
    set { linkedException_ = value;}
}

```

Získejte další výjimku v řetězci výjimek.

Pokud v řetězci nejsou žádné další výjimky, vrací metoda hodnotu `null`.

Výjimky:

- Výjimka `XMSEException`

XMSFactoryFactory

Pokud aplikace nepoužívá spravované objekty, použijte tuto třídu k vytvoření továren připojení, front a témat.

Hierarchie dědičnosti:

Není

Vlastnosti produktu .NET

Metadata-Načíst metadata

Rozhraní:

```
IConnectionMetaData MetaData
```

Získejte metadata, která odpovídají typu připojení objektu XMSFactoryFactory .

Výjimky:

Není

Metody

CreateConnectionTovárna-Vytvořit továrnu připojení

Rozhraní:

```
IConnectionFactory CreateConnectionFactory();
```

Vytvořte objekt ConnectionFactory deklarovaného typu.

Parametry:

Není

vrátí:

Objekt ConnectionFactory .

Výjimky:

- Výjimka XMSException

CreateQueue -Vytvořit frontu

Rozhraní:

```
IDestination CreateQueue(String name);
```

Vytvořte objekt místa určení, který bude reprezentovat frontu na serveru systému zpráv.

Tato metoda nevytvoří frontu na serveru systému zpráv. Frontu musíte vytvořit dříve, než bude moci aplikace volat tuto metodu.

Parametry:

name (vstup)

Řetězcový objekt zapouzdřující název fronty nebo zapouzdření identifikátoru URI (Uniform Resource Identifier), který identifikuje frontu.

vrátí:

Cílový objekt reprezentující frontu.

Výjimky:

- Výjimka XMSEException

CreateTopic -Vytvořit téma

Rozhraní:

```
IDestination CreateTopic(String name);
```

Vytvoření cílového objektu pro znázornění tématu.

Parametry:**name (vstup)**

Řetězcový objekt zapouzdřující název tématu nebo zapouzdření identifikátoru URI (Uniform Resource Identifier), který identifikuje dané téma.

vrátí:

Cílový objekt reprezentující téma.

Výjimky:

- Výjimka XMSEException

GetInstance -Získání instance třídy XMSFactoryFactory

Rozhraní:

```
static XMSFactoryFactory GetInstance(int connectionType);
```

Vytvořte instanci XMSFactoryFactory. Aplikace XMS používá objekt XMSFactoryFactory k získání odkazu na objekt ConnectionFactory, který odpovídá požadovanému typu protokolu. Tento objekt ConnectionFactory pak může vytvářet připojení pouze pro tento typ protokolu.

Parametry:**connectionType (vstup)**

Typ připojení, pro které objekt ConnectionFactory produkuje připojení:

- XMSC.CT_WPM
- XMSC.CT_RTT
- XMSC.CT_WMQ

vrátí:

Objekt XMSFactoryFactory vyhrazený pro deklarovaný typ připojení.

Výjimky:

- Výjimka NotSupported

Vlastnosti objektů XMS

Tento kapitola dokumentuje vlastnosti objektu definované parametrem XMS.

kapitola obsahuje následující sekce:

- [“Vlastnosti připojení” na stránce 174](#)
- [“Vlastnosti objektu ConnectionFactory” na stránce 175](#)

- [“Vlastnosti dat ConnectionMeta” na stránce 181](#)
- [“Vlastnosti místa určení” na stránce 181](#)
- [“Vlastnosti objektu InitialContext” na stránce 183](#)
- [“Vlastnosti zprávy” na stránce 184](#)
- [“Vlastnosti objektu MessageConsumer” na stránce 189](#)
- [“Vlastnosti objektu MessageProducer” na stránce 189](#)
- [“Vlastnosti relace” na stránce 189](#)

Každý sekce vypisuje vlastnosti objektu uvedeného typu a poskytuje krátký popis každé vlastnosti.

Tato kapitola také obsahuje [“Definice vlastností” na stránce 189](#) sekce, která poskytuje definici každé vlastnosti.

Pokud aplikace definuje své vlastní vlastnosti objektů popsaných v tomto kapitola, nezpůsobí chybu, ale může dojít k nepředvídatelným výsledkům.

Poznámka: Názvy a hodnoty vlastností v této sekci jsou zobrazeny ve tvaru `XMSC.NAME`, což je formulář používaný pro jazyky C a C++. Avšak v produktu .NET může být název vlastnosti `XMSC.NAME` nebo `XMSC_NAME`, v závislosti na tom, jak ji používáte:

- Pokud určujete vlastnost, název vlastnosti musí být ve tvaru `XMSC.NAME`, jak je uvedeno v následujícím příkladu:

```
cf.SetStringProperty(XMSC.WMQ_CHANNEL, "DOTNET.SVRCONN");
```

- Pokud zadáváte řetězec, musí být název vlastnosti ve tvaru `XMSC_NAME` tak, jak je uvedeno v následujícím příkladu:

```
cf.SetStringProperty("XMSC_WMQ_CHANNEL", "DOTNET.SVRCONN");
```

V produktu .NET jsou názvy vlastností a hodnoty zadány jako konstanty ve třídě `XMSC`. Tyto konstanty identifikují řetězce a budou použity libovolnou aplikací produktu `XMS.NET`. Používáte-li tyto předdefinované konstanty, názvy a hodnoty vlastností jsou ve formátu `XMSC.NAME`, takže byste například mohli použít `XMSC.USERID`, spíše než `XMSCS_USERID`.

Datové typy jsou také ve formuláři použitím pro C/C++. Odpovídající hodnoty pro .NET lze nalézt v [“Datové typy pro .NET” na stránce 42](#).

Související pojmy

Vytváření vlastních aplikací

Sestavujete své vlastní aplikace, jako např. sestavení ukázkových aplikací.

Související odkazy

[.NETRozhraní](#)

Tato sekce dokumentuje rozhraní třídy .NET a jejich vlastnosti a metody.

Vlastnosti připojení

Přehled vlastností objektu připojení s odkazy na podrobnější referenční informace.

Tabulka 25. Vlastnosti připojení	
Název vlastnosti	Popis
“XMSC_WMQ_RESOLVE_QUEUE_MANAGER” na stránce 222	Tato vlastnost se používá k získání názvu správce front, ke kterému je připojen.
“XMSC_WMQ_RESOLVE_QUEUE_MANAGER_ID” na stránce 223	Tato vlastnost je naplněna identifikátorem ID správce front po připojení.
<code>XMSC_WPM_CONNECTION_PROTOCOL</code>	Komunikační protokol použitý pro připojení k jádru systému zpráv. Tato vlastnost je jen pro čtení.

<i>Tabulka 25. Vlastnosti připojení (pokračování)</i>	
Název vlastnosti	Popis
<u>NÁZEV_HOSTITEL_XMSC</u>	Název hostitele nebo adresa IP systému, který obsahuje jádro systému zpráv, ke kterému je aplikace připojena. Tato vlastnost je jen pro čtení.
<u>XMSCS_WPM_ME_NAME</u>	Název jádra systému zpráv, ke kterému je aplikace připojena. Tato vlastnost je jen pro čtení.
<u>XCSMS_WPM_PORT</u>	Číslo portu naslouchaného v jádru systému zpráv, ke kterému je aplikace připojena. Tato vlastnost je jen pro čtení.

Objekt připojení má také vlastnosti určené jen pro čtení, které jsou odvozeny od vlastností továrny připojení, která byla použita k vytvoření připojení. Tyto vlastnosti se odvozují nejen od vlastností továrny připojení, které byly nastaveny v době vytvoření připojení, ale také z výchozích hodnot vlastností, které nebyly nastaveny. Vlastnosti zahrnují pouze ty vlastnosti, které jsou relevantní pro typ serveru systému zpráv, ke kterému je aplikace připojena. Názvy vlastností jsou stejné jako názvy vlastností továrny připojení.

Vlastnosti objektu ConnectionFactory

Přehled vlastností objektu ConnectionFactory s odkazy na podrobnější referenční informace.

<i>Tabulka 26. Vlastnosti objektu ConnectionFactory</i>	
Název vlastnosti	Popis
<u>“XMSC_ASYNC_EXCEPTIONS” na stránce 199</u>	Tato vlastnost určuje, zda rozhraní XMS informuje modul ExceptionListener pouze v případě, že dojde k přerušení připojení nebo když dojde k asynchronnímu výskytu jakékoli výjimky pro volání rozhraní XMS API. Tato vlastnost platí pro všechna připojení vytvořená z této továrny připojení, která mají registrován modul ExceptionListener.
<u>ID XMSCS_CLIENT_ID</u>	Identifikátor klienta pro účely připojení.
<u>XMSC_CONNECTION_TYPE</u>	Typ serveru systému zpráv, ke kterému se aplikace připojuje.
<u>HESLO_HESLO</u>	Heslo, které lze použít k ověření aplikace při pokusu o připojení k serveru systému zpráv.
<u>“XMSC_RTC_BROKER_PING_INTERVAL” na stránce 204</u>	Časový interval v milisekundách, po který XMS .NET kontroluje připojení k serveru systému zpráv v reálném čase ke zjištění jakékoli aktivity.
<u>XMSCS_TRT_CONNECTION_PROTOCOL</u>	Komunikační protokol použitý pro připojení v reálném čase ke zprostředkovateli.
<u>NÁZEV_HOSTITELE XMSCS_RTT_NAME</u>	Název nebo adresa IP hostitele systému, na kterém je spuštěn zprostředkovatel.
<u>ADRESA_XCST_RT_LOCAL_ADDRESS</u>	Název hostitele nebo adresa IP lokálního síťového rozhraní, které má být použito pro připojení v reálném čase ke zprostředkovateli.
<u>XMSCS_RTT_MULTICAST</u>	Nastavení výběrového vysílání pro továrnu připojení nebo cíl.
<u>VÝPORT XMSCS_RTT_PORT</u>	Číslo portu, na kterém zprostředkovatel naslouchá příchozím požadavkům.

<i>Tabulka 26. Vlastnosti objektu ConnectionFactory (pokračování)</i>	
Název vlastnosti	Popis
<u>ID UŽIVATELE XMSC</u>	Identifikátor uživatele, který lze použít k ověření aplikace při pokusu o připojení k serveru systému zpráv.
<u>XMSCS_WMQ_BROKER_CONTROLQ</u>	Název řídicí fronty použité zprostředkovatelem. Poznámka: Tuto vlastnost lze použít s verzí 2.0 produktu IBM Message Service Client for .NET , ale nemá žádný vliv na aplikaci připojenou ke správci front produktu IBM WebSphere MQ 7.0 , pokud vlastnost XMSCS_WMQ_PROVIDER_VERSION dané továrny připojení není nastavena na číslo verze nižší než 7.
<u>XMSCS_WMQ_BROKER_PUBQ</u>	Název fronty monitorované zprostředkovatelem, kde aplikace odesílají zprávy, které publikují. Poznámka: Tuto vlastnost lze použít s verzí 2.0 produktu IBM Message Service Client for .NET , ale nemá žádný vliv na aplikaci připojenou ke správci front produktu IBM WebSphere MQ 7.0 , pokud vlastnost XMSCS_WMQ_PROVIDER_VERSION dané továrny připojení není nastavena na číslo verze nižší než 7.
<u>XMSCS_WMQ_BROKER_QMGR</u>	Název správce front, ke kterému je zprostředkovatel připojen. Poznámka: Tuto vlastnost lze použít s verzí 2.0 produktu IBM Message Service Client for .NET , ale nemá žádný vliv na aplikaci připojenou ke správci front produktu IBM WebSphere MQ 7.0 , pokud vlastnost XMSCS_WMQ_PROVIDER_VERSION dané továrny připojení není nastavena na číslo verze nižší než 7.
<u>XMSCS_WMQ_BROKER_SUBQ</u>	Název fronty odběratele pro spotřebitele netrvalých zpráv. Poznámka: Tuto vlastnost lze použít s verzí 2.0 produktu IBM Message Service Client for .NET , ale nemá žádný vliv na aplikaci připojenou ke správci front produktu IBM WebSphere MQ 7.0 , pokud vlastnost XMSCS_WMQ_PROVIDER_VERSION dané továrny připojení není nastavena na číslo verze nižší než 7.
<u>XMSCS_WMQ_BROKER_VERSION</u>	Typ zprostředkovatele použitý aplikací pro připojení nebo pro cíl. Poznámka: Tuto vlastnost lze použít s verzí 2.0 produktu IBM Message Service Client for .NET , ale nemá žádný vliv na aplikaci připojenou ke správci front produktu IBM WebSphere MQ 7.0 , pokud vlastnost XMSCS_WMQ_PROVIDER_VERSION dané továrny připojení není nastavena na číslo verze nižší než 7.
<u>“XMSC_WMQ_CCDTURL” na stránce 209</u>	Adresa URL (Uniform Resource Locator) identifikující název a umístění souboru, který obsahuje tabulku definic kanálů klienta, a také určuje, jakým způsobem lze k souboru přistupovat.
<u>XMSCS_WMQ_CHANNEL</u>	Název kanálu, který se má použít pro připojení.

<i>Tabulka 26. Vlastnosti objektu ConnectionFactory (pokračování)</i>	
Název vlastnosti	Popis
<u>“XMSC_WMQ_CLIENT_RECONNECT_OPTIONS”</u> na stránce 210	Tato vlastnost uvádí volby nového připojení klienta pro nová připojení vytvořená touto továrnou
<u>“XMSC_WMQ_CLIENT_RECONNECT_TIMEOUT”</u> na stránce 210	Tato vlastnost určuje dobu trvání (v sekundách), po kterou se připojení klienta pokouší znovu navázat spojení.
<u>XMSC_WMQ_CONNECTION_MODE</u>	Režim, pomocí kterého se aplikace připojuje ke správci front.
<u>“XMSC_WMQ_CONNECTION_NAME_LIST”</u> na stránce 211	Tato vlastnost určuje hostitele, ke kterému se klient pokusí znovu připojit po přerušení připojení.
<u>XMSCS_WMQ_FAIL_IF_QUIESCE</u>	Zda se volání konkrétních metod nezdaří, pokud je správce front, k němuž je aplikace připojen, ve stavu uvedení do klidového stavu.
<u>NÁZEV_HOSTITEL_XMSC</u>	Název nebo adresa IP hostitele systému, na kterém je spuštěn správce front.
<u>XMSC_WMQ_LOCAL_ADDRESS</u>	Pro připojení ke správci front tato vlastnost určuje rozhraní lokální sítě, lokálního portu nebo rozsahu lokálních portů, nebo obojí.
<u>XMSC_WMQ_MESSAGE_SELECTION</u>	Určuje, zda má být výběr zpráv proveden klientem XMS nebo zprostředkovatelem. Poznámka: Tuto vlastnost lze použít s verzí 2.0 produktu IBM Message Service Client for .NET , ale nemá žádný vliv na aplikaci připojenou ke správci front produktu IBM WebSphere MQ 7.0 , pokud vlastnost XMSCS_WMQ_PROVIDER_VERSION dané továrny připojení není nastavena na číslo verze nižší než 7.
<u>XMSCS_WMQ_MSG_BATCH_SIZE</u>	Maximální počet zpráv, které mají být načteny z fronty v jedné dávce, v případě asynchronního doručování zpráv. Poznámka: Tuto vlastnost lze použít s verzí 2.0 produktu IBM Message Service Client for .NET , ale nemá žádný vliv na aplikaci připojenou ke správci front produktu IBM WebSphere MQ 7.0 , pokud vlastnost XMSCS_WMQ_PROVIDER_VERSION dané továrny připojení není nastavena na číslo verze nižší než 7.
<u>XMSCS_WMQ_POLLING_INTERVAL</u>	Pokud každý modul listener pro zprávy v rámci relace nemá ve frontě žádnou odpovídající zprávu, jedná se o maximální interval v milisekundách, který uplyne předtím, než se každý modul listener pro zprávy znovu pokusí získat zprávu z příslušné fronty. Poznámka: Tuto vlastnost lze použít s verzí 2.0 produktu IBM Message Service Client for .NET , ale nemá žádný vliv na aplikaci připojenou ke správci front produktu IBM WebSphere MQ 7.0 , pokud vlastnost XMSCS_WMQ_PROVIDER_VERSION dané továrny připojení není nastavena na číslo verze nižší než 7.
<u>“XMSC_WMQ_PROVIDER_VERSION”</u> na stránce 219	Verze, vydání, úroveň modifikace a opravný balík správce front, ke kterému se tato aplikace hodlá připojovat.

<i>Tabulka 26. Vlastnosti objektu ConnectionFactory (pokračování)</i>	
Název vlastnosti	Popis
PORT_XMSCS_WMQ_PORT	Číslo portu, na kterém správce front naslouchá příchozím požadavkům.
INTER_INTERVALOVÝ_ČASOVÝ_ADRESA_XMSC	Počet zpráv publikovaných vydavatelem před tím, než klient produktu XMS vyžádá potvrzení od zprostředkovatele. Poznámka: Tuto vlastnost lze použít s verzí 2.0 produktu IBM Message Service Client for .NET , ale nemá žádný vliv na aplikaci připojenou ke správci front produktu IBM WebSphere MQ 7.0 , pokud vlastnost XMSCS_WMQ_PROVIDER_VERSION dané továrny připojení není nastavena na číslo verze nižší než 7.
“XMSC_WMQ_PUT_ASYNC_ALLOWED” na stránce 215	Tato vlastnost určuje, zda producenti zpráv mohou používat k odesílání zpráv do tohoto místa určení asynchronní operace put.
XMSCS_WMQ_QMGR_CCSID	Identifikátor (CCSID) kódové znakové sady nebo kódové stránky, v níž se mezi klientem XMS a klientem WebSphere MQ vyměňují pole znakových dat definovaná v rozhraní MQI (Message Queue Interface).
XMSCS_WMQ_QUEUE_MANAGER	Název správce front, s nímž má být navázáno připojení.
XMSC_WMQ_RECEIVE_EXIT	Identifikuje uživatelskou proceduru pro přijetí zprávy kanálu, která má být spuštěna.
XMSCS_WMQ_RECEIVE_EXIT_INIT	Uživatelská data, která jsou předána uživatelské proceduře pro přijetí zprávy kanálu při volání.
XMSCS_WMQ_SECURITY_EXIT	Identifikuje uživatelskou proceduru pro zabezpečení zprávy kanálu.
XMSCS_WMQ_SECURITY_EXIT_INIT	Uživatelská data, která jsou předána uživatelské proceduře pro zabezpečení zprávy kanálu při volání.
“XMSC_WMQ_SEND_CHECK_COUNT” na stránce 224	Počet povolených odeslaných volání mezi kontrolou asynchronních chyb vložení během jedné netrasmakční relace XMS.
XCS_WMQ_SEND_EXIT	Identifikuje uživatelskou proceduru pro odeslání zprávy kanálu.
XMSCS_WMQ_SEND_EXIT_INIT	Uživatelská data, která jsou předána uživatelským procedurám pro odeslání zprávy kanálu při volání.
“XMSC_WMQ_SHARE_CONV_ALLOWED” na stránce 224	Zda může připojení klienta sdílet svůj soket s dalšími propojeními XMS nejvyšší úrovně z téhož procesu do stejného správce front, pokud se definice kanálu shodují. Tato vlastnost slouží k povolení úplné izolace produktu Connections v samostatných soketech, pokud je to vyžadováno pro vývoj aplikací, údržbu nebo provozní důvody.
XMSCS_WMQ_SSL_CERT_STORES	Umístění serverů obsahující seznamy odvolaných certifikátů (CRL), které mají být použity v připojení SSL ke správci front.

<i>Tabulka 26. Vlastnosti objektu ConnectionFactory (pokračování)</i>	
Název vlastnosti	Popis
<u>XMSC_WMQ_SSL_CIPHER_SPEC</u>	Název specifikace CipherSpec, která má být použita v zabezpečeném připojení ke správci front.
<u>XMSCS_WMQ_SSL_CIPHER_SUITE</u>	Název sady CipherSuite, která má být použita v rámci připojení TLS ke správci front. Protokol použitý při vyjednávání zabezpečeného připojení závisí na určené sadě CipherSuite.
<u>XMSCS_WMQ_CRYPTO_HW</u>	Podrobnosti konfigurace šifrovacího hardwaru připojeného k systému klienta.
<u>POŽAD_ZÁSOBNÍ_CHYBA_XMSC</u>	Hodnota této vlastnosti určuje, zda aplikace může nebo nemůže používat šifrovací sady, které nevyhovují standardu FIPS. Je-li tato vlastnost nastavena na hodnotu true, pro připojení klientského serveru se používají pouze algoritmy FIPS.
<u>XMSCS_WMQ_SSL_KEY_REPOSITORY</u>	Umístění souboru databáze klíčů, ve kterém jsou uloženy klíče a certifikáty.
<u>XMSCS_WMQ_SSL_KEY_RESETCOUNT</u>	Hodnota KeyResetCount představuje celkový počet nešifrovaných bajtů odeslaných a přijatých v rámci konverzace SSL, než je znovu vyjednán tajný klíč.
<u>XMSCS_WMQ_SSL_PEER_NAME</u>	Název typu peer, které se použije při připojení SSL ke správci front.
<u>XMSCS_WMQ_SYNCINT_ALL_GETS</u>	Určuje, zda musí být všechny zprávy načteny z front v rámci řízení synchronizačního bodu.
“XMSC_WMQ_TARGET_CLIENT” na stránce 231	
<u>XMSCS_WMQ_TEMP_Q_PREFIX</u>	Předpona používaná k vytvoření názvu dynamické fronty produktu WebSphere MQ , která se vytvoří, když aplikace vytvoří XMS dočasnou frontu.
<u>TEPLOT_TEPLOTA_XMSCS_WMQ__TOPIC_PREFIX</u>	Při vytváření dočasných témat vygeneruje XMS řetězec tématu ve tvaru "TEMP/TEMPTOPICPREFIX/unique_id", nebo pokud tato vlastnost obsahuje výchozí hodnotu, vygeneruje se tento řetězec "TEMP/unique_id". Určení neprázdné hodnoty umožní definování specifických modelových front za účelem vytvoření spravovaných front pro odběratele dočasných témat vytvořených v rámci tohoto připojení.
<u>Model XMSCS_WMQ_TEMPORARY_MODEL</u>	Název modelové fronty produktu WebSphere MQ , ze které se vytvoří dynamická fronta, když aplikace vytvoří XMS dočasnou frontu.
<u>XMSCS_WPM_BUS_NAME</u>	U továrny připojení se jedná o název sběrnice SIBus, ke které se aplikace připojuje, nebo pro cíl, název sběrnice SIBus, ve které cíl existuje.
<u>XMSC_WPM_CONNECTION_PROXIMITY</u>	Nastavení blízkosti připojení pro připojení.
<u>ADRESA_DOP_SUB_V_ADRESÁŘ_XMSC</u>	Název jádra systému zpráv, v nichž jsou spravovány všechny trvalé odběry pro připojení nebo cíl.

<i>Tabulka 26. Vlastnosti objektu ConnectionFactory (pokračování)</i>	
Název vlastnosti	Popis
<u>XMSCS_WPM_LOCAL_ADDRESS</u>	Pro připojení ke sběrnici SIBus tato vlastnost určuje rozhraní lokální sítě, lokálního portu nebo rozsahu lokálních portů, nebo obojí.
<u>XMSCS_WPM_NON_PERSISTENT_MAP</u>	Úroveň spolehlivosti přechodných zpráv, které se odesílají pomocí připojení.
<u>XMSCS_WPM_PERSISTENT_MAP</u>	Úroveň spolehlivosti trvalých zpráv, které se odesílají pomocí připojení.
<u>XMSCS_WPM_PROVIDER_ENDPOINTS</u>	Sekvence jednoho nebo více adres koncového bodu zaváděcích serverů.
<u>XMSCS_WPM_TARGET_GROUP</u>	Název cílové skupiny jader systému zpráv.
<u>XMSC_WPM_TARGET_TARGET_SIGNIFICANCE</u>	Významnost cílové skupiny jader systému zpráv.
<u>XMSC_WPM_TARGET_TRANSPORT_CHAIN</u>	Název transportu příchozích požadavků, který musí aplikace používat pro připojení k jádru systému zpráv.
<u>XMSCS_WPM_TARGET_TYPE</u>	Typ cílové skupiny jader systému zpráv.
<u>XMSCS_WPM_TEMP_Q_PREFIX</u>	Předpona použitá k vytvoření názvu dočasné fronty, která je vytvořena ve sběrnici pro integraci služeb, když aplikace vytvoří XMS dočasnou frontu.
<u>XMSCS_WPM_TEMP_TOPIC_PREFIX</u>	Předpona používaná k vytvoření názvu dočasného tématu vytvořeného aplikací.

Související pojmy

ConnectionFactoryies a objekty připojení

Objekt ConnectionFactory poskytuje šablonu, kterou aplikace používá k vytvoření objektu připojení. Aplikace používá objekt Connection k vytvoření objektu relace.

Připojení ke sběrnici pro integraci služeb

Aplikace XMS se může připojit ke sběrnici pro integraci služeb produktu WebSphere Application Server buď pomocí přímého připojení TCP/IP, nebo pomocí protokolu HTTP přes TCP/IP.

Zabezpečená připojení ke správci front produktu IBM MQ

Chcete-li povolit aplikaci XMS .NET zabezpečené připojení ke správci front produktu IBM MQ , musí být příslušné vlastnosti definovány v objektu ConnectionFactory .

Zabezpečená připojení ke stroji systému zpráv produktu WebSphere Application Server service integration bus

Chcete-li povolit aplikaci XMS .NET zabezpečené připojení ke stroji systému zpráv produktu WebSphere Application Server service integration bus , musí být příslušné vlastnosti definovány v objektu ConnectionFactory .

Mapování vlastností pro spravované objekty

Chcete-li aplikacím povolit použití továrny připojení a definic cílových objektů produktu IBM MQ JMS a WebSphere Application Server , musí být vlastnosti načtené z těchto definic mapovány na odpovídající vlastnosti produktu XMS , které lze nastavit v továrnách a cílech připojení XMS .

Související úlohy

Vytvoření spravovaných objektů

Definice objektu ConnectionFactory a Cílový objekt, které aplikace produktu XMS vyžadují k vytvoření připojení k serveru systému zpráv, musí být vytvořeny pomocí příslušných nástrojů pro správu.

Související odkazy

Povinné vlastnosti pro spravované objekty ConnectionFactory

Když aplikace vytvoří továrnu připojení, musí být definován počet vlastností pro vytvoření připojení k serveru systému zpráv.

Vlastnosti dat ConnectionMeta

Přehled vlastností datového objektu ConnectionMetas odkazy na podrobnější referenční informace.

<i>Tabulka 27. Vlastnosti dat ConnectionMeta</i>	
Název vlastnosti	Popis
XMSCS_JMS_MAJOR_VERZE	Hlavní číslo verze specifikace JMS , na které je založena XMS . Tato vlastnost je jen pro čtení.
XMSCS_JMS_MINOR_VERSION	Vedlejší číslo verze specifikace JMS , na které je založena XMS . Tato vlastnost je jen pro čtení.
XMSCS_JMS_VERSION	Identifikátor verze specifikace JMS , na které je založena XMS . Tato vlastnost je jen pro čtení.
VÝZNAM_VELIKO_XMSC	Číslo verze klienta XMS . Tato vlastnost je jen pro čtení.
XMSCS_MINOR_VERSION	Číslo verze klienta XMS . Tato vlastnost je jen pro čtení.
NÁZEV_ADRESÁŘ_XMSC	Poskytovatel klienta XMS . Tato vlastnost je jen pro čtení.
VERZE_XMSCS_VERSION	Identifikátor verze klienta XMS . Tato vlastnost je jen pro čtení.

Vlastnosti místa určení

Přehled vlastností objektu Destination, s odkazy na podrobnější referenční informace.

<i>Tabulka 28. Vlastnosti místa určení</i>	
Název vlastnosti	Popis
REŽIM_XMSCS_DELIVERY_MODE	Režim doručení zpráv odeslaných do cíle.
PRIORITA_XMSCSC	Priorita zpráv odeslaných do cíle.
XMSCS_RTT_MULTICAST	Nastavení výběrového vysílání pro továrnu připojení nebo cíl.
XMSCS_TIME_TO_LIVE	Doba životnosti zpráv odeslaných do cíle.
XMSCS_WMQ_BROKER_VERSION	Typ zprostředkovatele použitý aplikací pro připojení nebo pro cíl.
CCSID_XMSCS_WMQ_CCSID	Identifikátor (CCSID) kódované znakové sady nebo kódové stránky, že řetězce znakových dat v těle zprávy jsou v době, kdy klient XMS předá zprávu do cíle.
DÍL_UPO_WMQTMQ_XMSC	Název fronty odběratele pro trvalého odběratele, který přijímá zprávy z cíle. Poznámka: Tuto vlastnost lze použít s verzí 2.0 produktu IBM Message Service Client for .NET , ale nemá žádný vliv na aplikaci připojenou ke správci front produktu IBM WebSphere MQ 7.0 , pokud vlastnost XMSCS_WMQ_PROVIDER_VERSION dané továrny připojení není nastavena na číslo verze nižší než 7.
XMSCS_WMQ_ENCODING	Jak číselná data v těle zprávy jsou představována, když klient produktu XMS předává zprávu do cíle.

<i>Tabulka 28. Vlastnosti místa určení (pokračování)</i>	
Název vlastnosti	Popis
<u>XMSCS_WMQ_FAIL_IF QUIESCE</u>	Zda se volání konkrétních metod nezdaří, pokud je správce front, k němuž je aplikace připojen, ve stavu uvedení do klidového stavu.
<u>"XMSC_WMQ_MESSAGE_BODY" na stránce 213</u>	Tato vlastnost určuje, zda aplikace XMS zpracovává strukturu MQRFH2 zprávy produktu IBM WebSphere MQ jako součást informačního obsahu zprávy (tj. v rámci těla zprávy).
<u>"XMSC_WMQ_MQMD_MESSAGE_CONTEXT" na stránce 214</u>	určuje, jaká úroveň kontextu zprávy má být nastavena aplikací XMS . Aby tato vlastnost mohla nabýt účinnosti, musí aplikace běžet s příslušným oprávněním kontextu.
<u>"XMSC_WMQ_MQMD_READ_ENABLED" na stránce 215</u>	Tato vlastnost určuje, zda aplikace XMS může extrahovat hodnoty z polí MQMD, či nikoli.
<u>"XMSC_WMQ_MQMD_WRITE_ENABLED" na stránce 215</u>	Tato vlastnost určuje, zda může aplikace XMS určovat hodnoty polí MQMD či nikoli.
<u>"XMSC_WMQ_READ_AHEAD_ALLOWED" na stránce 216</u>	Tato vlastnost určuje, zda mohou příjemci zpráv a zprostředkovatelé front používat dopředné čtení k načtení netrvalých netraskčních zpráv z tohoto cíle do interní vyrovnávací paměti před jejich přijetím.
<u>"XMSC_WMQ_READ_AHEAD_CLOSE_POLICY" na stránce 216</u>	Tato vlastnost určuje, zda budou zprávy doručovány do asynchronního modulu listener zpráv, co se stane se zprávami v interní vyrovnávací paměti pro čtení při zavření spotřebitele zpráv.
<u>"XMSC_WMQ_RECEIVE_CCSD" na stránce 221</u>	Vlastnost místa určení, která nastavuje cílové CCSID pro převod zpráv správce front. Hodnota je ignorována, pokud není hodnota XMSCS_WMQ_RECEIVE_CONVERSION nastavena na hodnotu WMQ_RECEIVE_CONVERSION_QMGR.
<u>"XMSC_WMQ_RECEIVE_CONVERSION" na stránce 222</u>	Cílová vlastnost, která určuje, zda bude převod dat prováděn správcem front.
<u>CLIENT_WMQ_TARGET_CLIENT</u>	Určuje, zda zprávy odeslané do cíle obsahují záhlaví MQRFH2.
<u>TEPLOT_TEPLOTA XMSCS_WMQ__TOPIC_PREFIX</u>	Při vytváření dočasných témat vygeneruje XMS řetězec tématu ve tvaru "TEMP/TEMPTOPICPREFIX/unique_id", nebo pokud tato vlastnost obsahuje výchozí hodnotu, vygeneruje se tento řetězec "TEMP/unique_id". Určení neprázdné hodnoty umožní definování specifických modelových front za účelem vytvoření spravovaných front pro odběratele dočasných témat vytvořených v rámci tohoto připojení.
<u>XMSCS_WPM_BUS_NAME</u>	U továrny připojení se jedná o název sběrnice SIBus, ke které se aplikace připojuje, nebo pro cíl, název sběrnice SIBus, ve které cíl existuje.
<u>XMSCS_WPM_TOPIC_SPACE</u>	Název prostoru témat, který obsahuje dané téma.

Související pojmy

ConnectionFactories a objekty připojení

Objekt ConnectionFactory poskytuje šablonu, kterou aplikace používá k vytvoření objektu připojení. Aplikace používá objekt Connection k vytvoření objektu relace.

Připojení ke sběrnici pro integraci služeb

Aplikace XMS se může připojit ke sběrnici pro integraci služeb produktu WebSphere Application Server buď pomocí přímého připojení TCP/IP, nebo pomocí protokolu HTTP přes TCP/IP.

Cíle

Aplikace XMS používá objekt Destination k určení místa určení odesílaných zpráv a zdroje přijímaných zpráv.

Cílové zástupné znaky

Produkt XMS poskytuje podporu pro cílové zástupné znaky a zajišťuje, aby zástupné znaky mohly být předávány do místa, kde jsou potřeba pro porovnávání. Pro každý typ serveru, se kterým může pracovat produkt XMS, existuje jiný zástupný schéma.

Jednotné identifikátory prostředků tématu

Identifikátor URI (Uniform Resource Identifier) tématu určuje název tématu; pro něj lze také zadat jednu či více vlastností.

Jednotné identifikátory prostředků fronty

Identifikátor URI pro frontu určuje název fronty. Může také určovat jednu či více vlastností fronty.

Dočasná místa určení

Aplikace produktu XMS mohou vytvářet a používat dočasná místa určení.

Mapování vlastností pro spravované objekty

Chcete-li aplikacím povolit použití továrny připojení a definic cílových objektů produktu IBM MQ JMS a WebSphere Application Server, musí být vlastnosti načtené z těchto definic mapovány na odpovídající vlastnosti produktu XMS, které lze nastavit v továrnách a cílech připojení XMS.

Související úlohy

Vytvoření spravovaných objektů

Definice objektu ConnectionFactory a Cílový objekt, které aplikace produktu XMS vyžadují k vytvoření připojení k serveru systému zpráv, musí být vytvořeny pomocí příslušných nástrojů pro správu.

Související odkazy

Povinné vlastnosti pro spravované objekty typu Destination

Aplikace, která vytváří cíl, musí nastavit několik vlastností, které aplikace na administrovaného cílovém objektu.

Vlastnosti objektu InitialContext

Přehled vlastností objektu InitialContext s odkazy na podrobnější referenční informace.

Název vlastnosti	Popis
<u>Adresa URL XMSCS_IC_PROVIDER_URL</u>	Použije se k vyhledání adresáře pojmenování rozhraní JNDI tak, aby služba pojmenování COS nevyžadovala být na stejném serveru jako webová služba.
<u>XMSC_IC_SECURITY_AUTHENTICATION</u>	Na základě kontextu Java Kontextové rozhraní SECURITY_AUTHENTICATION. Tato vlastnost je použitelná pouze pro kontext pojmenování COS.
<u>XMSC_IC_SECURITY_CREDENTIALS</u>	Založeno na kontextovém rozhraní Java SECURITY_CREDENTIALS. Tato vlastnost je použitelná pouze pro kontext pojmenování COS.
<u>XMSC_IC_SECURITY_PRINCIPAL</u>	Založeno na kontextu Java kontextu SECURITY_PRINCIPAL. Tato vlastnost je použitelná pouze pro kontext pojmenování COS.

<i>Tabulka 29. Vlastnosti objektu InitialContext (pokračování)</i>	
Název vlastnosti	Popis
<u>XMSC_IC_SECURITY_PROTOCOL</u>	Založeno na kontextu Java kontextu SECURITY_PROTOCOL Tato vlastnost je použitelná pouze pro kontext pojmenování COS.
<u>Adresa XMSCS_IC_URL</u>	Pro kontexty LDAP a FileSystem, adresa úložiště obsahující administrované objekty. Pro kontexty pojmenování COS, adresa webové služby, která vyhledává objekty v adresáři.

Související pojmy

Vlastnosti InitialContext

Parametry konstruktoru InitialContext zahrnují umístění úložiště spravovaných objektů, které jsou poskytnuty jako identifikátor URI (Uniform Resource Indicator). Aby mohla aplikace vytvořit připojení k úložišti, může být nezbytné poskytnout více informací než informace obsažené v identifikátoru URI.

Formát identifikátoru URI pro počáteční kontexty produktu XMS

Umístění úložiště spravovaných objektů je poskytováno jako jednotný indikátor prostředku (URI). Formát identifikátoru URI závisí na typu kontextu.

Načítání spravovaných objektů

Produkt XMS načítá administrovaný objekt z úložiště s použitím adresy zadané při vytvoření objektu InitialContext nebo ve vlastnostech InitialContext .

Související úlohy

Objekty InitialContext

Aplikace musí vytvořit počáteční kontext, který má být použit k vytvoření připojení k úložišti spravovaných objektů, aby bylo možné načíst požadované spravované objekty.

Vlastnosti zprávy

Přehled vlastností objektu Zpráva s odkazy na podrobnější referenční informace.

<i>Tabulka 30. Vlastnosti zprávy</i>	
Název vlastnosti	Popis
<u>JMS_IBM_CHARACTER_SET</u>	Identifikátor (CCSID) kódované znakové sady nebo kódové stránky, že řetězce znakových dat v těle zprávy jsou v době, kdy klient XMS předává zprávu do zamýšleného místa určení. V prostředí XMS má tato vlastnost číselnou hodnotu a mapuje se na CCSID. Tato vlastnost je však založena na vlastnosti JMS, takže má řetězcovou hodnotu typu a mapuje se na znakovou sadu Java , která představuje tento numerický identifikátor CCSID.
<u>JMS_IBM_Encoding</u>	Jak číselná data v těle zprávy jsou představována, když klient produktu XMS předává zprávu do zamýšleného cíle.
<u>JMS_IBM_EXCEPTION_MESSAGE</u>	Text, který popisuje, proč byla zpráva odeslána do cíle výjimek. Tato vlastnost je jen pro čtení.
<u>JMS_IBM_EXCEPTIONPROBLEMDESTINATION</u>	Název cíle, ve kterém byla zpráva před odesláním zprávy do cíle výjimek.
<u>JMS_IBM_EXCEPTIONREASON</u>	Kód příčiny označující příčinu, proč byla zpráva odeslána do cíle výjimek.
<u>JMS_IBM_EXCEPTIONTIMESTAMP</u>	Čas, kdy byla zpráva odeslána do cíle výjimek.
<u>JMS_IBM_FEEDBACK</u>	Kód, který označuje chování zprávy sestavy.

<i>Tabulka 30. Vlastnosti zprávy (pokračování)</i>	
Název vlastnosti	Popis
<u>JMS_IBM_FORMAT</u>	Povahu dat aplikace ve zprávě.
<u>JMS_IBM_LAST_MSG_IN_GROUP</u>	Označuje, zda je zpráva poslední zprávou ve skupině zpráv.
<u>JMS_IBM_MSGTYPE</u>	Typ zprávy.
<u>JMS_IBM_PUTAPPLTYPE</u>	Typ aplikace, která odeslala zprávu.
<u>JMS_IBM_PUTDATE</u>	Datum, kdy byla zpráva odeslána.
<u>ČAS SOUBORU JMS_IBM_PUTTIME</u>	Čas, kdy byla zpráva odeslána.
<u>JMS_IBM_REPORT_COA</u>	Požadavek na zprávy sestavy 'potvrdit při příjmu' určující, kolik dat aplikace z původní zprávy musí být zahrnuto do zprávy sestavy.
<u>JMS_IBM_REPORT_COD</u>	Požadavek na zprávy sestavy 'potvrdit při doručení' určující, kolik dat aplikace z původní zprávy musí být zahrnuto do zprávy sestavy.
<u>JMS_IBM_REPORT_DISCARD_MSG</u>	Požadavek, aby byla zpráva vyřazena, pokud ji nelze doručit do zamýšleného cíle.
<u>JMS_IBM_REPORT_EXCEPTION</u>	Požadavek na zprávy sestavy výjimky určující, kolik dat aplikace z původní zprávy musí být zahrnuto do zprávy sestavy.
<u>JMS_IBM_REPORT_EXPIRATION</u>	Požadavek na zprávy sestavy vypršení platnosti určující, kolik dat aplikace z původní zprávy musí být zahrnuto do zprávy sestavy.
<u>JMS_IBM_REPORT_NAN</u>	Požadavek na zprávy sestavy negativního oznámení na akci.
<u>JMS_IBM_REPORT_PAN</u>	Požadavek na zprávy sestavy pozitivního oznámení na akci.
<u>JMS_IBM_REPORT_PASS_CORREL_ID</u>	Požadavek, aby identifikátor korelace jakékoli sestavy nebo zprávy odpovědi byl stejný jako identifikátor korelace původní zprávy.
<u>JMS_IBM_REPORT_PASS_MSG_ID</u>	Požadavek, aby identifikátor zprávy jakékoli sestavy nebo zprávy odpovědi byl stejný jako identifikátor zprávy původní zprávy.
<u>JMS_IBM_RETAIN</u>	Nastavení této vlastnosti označuje, správce front bude zprávu považovat za Zachované publikování.
<u>JMS_IBM_SYSTEM_MESSAGEID</u>	Identifikátor, který jedinečně identifikuje zprávu v rámci sběrnice SIBus. Tato vlastnost je jen pro čtení.
<u>JMSX_APPID</u>	Název aplikace, která odeslala zprávu.
<u>JMS_DELIVERY_COUNT</u>	Počet pokusů o doručení zprávy.
<u>JMSX_GROUPID</u>	Identifikátor skupiny zpráv, do které zpráva patří.
<u>JMSX_GROUPSEQ</u>	Pořadové číslo zprávy v rámci skupiny zpráv.
<u>JMS_USERID</u>	Identifikátor uživatele přidružený k aplikaci, která odeslala zprávu.

Vlastnosti JMS_IBM_MQMD*

IBM Message Service Client for .NET umožňuje klientským aplikacím číst/zapisovat pole MQMD pomocí rozhraní API. Umožňuje také přístup k datům zprávy produktu MQ. Standardně je přístup k produktu MQMD zakázán a musí být explicitně povolen aplikací s použitím vlastností cíle XMSCS_WMQ_MQMD_WRITE_ENABLED a XMSCS_WMQ_MQMD_READ_ENABLED. Tyto dvě vlastnosti jsou na sobě nezávislé.

Všechna pole MQMD s výjimkou StrucId a verze jsou odkryty jako další vlastnosti objektu zprávy a mají předponu JMS_IBM_MQMD.

Vlastnosti JMS_IBM_MQMD* mají vyšší prioritu než jiné vlastnosti, jako např. JMS_IBM* popsané v předchozí tabulce.

Odesílání zpráv

Všechna pole MQMD s výjimkou StrucId a verze jsou reprezentovány. Tyto vlastnosti se odkazují pouze na pole MQMD, kde se vlastnost vyskytuje jak v deskriptoru MQMD, tak v záhlaví MQRFH2, verze v MQRFH2 není nastavena ani extrahována. Je možné nastavit libovolnou z těchto vlastností, kromě JMS_IBM_MQMD_BackoutCount. Jakákoli hodnota nastavená pro parametr JMS_IBM_MQMD_BackoutCount je ignorována.

Má-li vlastnost maximální délku a zadáte příliš dlouhou hodnotu, bude tato hodnota zkrácena.

Pro určité vlastnosti musíte také nastavit vlastnost XMSCS_WMQ_MQMD_MESSAGE_CONTEXT na objektu Destination. Aby tato vlastnost mohla nabýt účinnosti, musí aplikace běžet s příslušným oprávněním kontextu. Pokud nenastavíte hodnotu XMSCS_WMQ_MQMD_MESSAGE_CONTEXT na příslušnou hodnotu, bude hodnota vlastnosti ignorována. Pokud jste nastavili XMSCS_WMQ_MQMD_MESSAGE_CONTEXT na odpovídající hodnotu, ale nemáte dostatečné oprávnění ke kontextu pro správce front, bude vydána výjimka. Vlastnosti vyžadující specifické hodnoty parametru XMSCS_WMQ_MQMD_MESSAGE_CONTEXT jsou následující.

Následující vlastnosti vyžadují, aby byl objekt XMSCS_WMQ_MQMD_MESSAGE_CONTEXT nastaven na hodnotu XMSCS_WMQ_MDCTX_SET_IDENTITY_CONTEXT nebo XMSCS_WMQ_MDCTX_SET_ALL_CONTEXT:

- JMS_IBM_MQMD_UserIdentifier .
- JMS_IBM_MQMD_AccountingToken .
- JMS_IBM_MQMD_ApplIdentityData

Následující vlastnosti vyžadují, aby byl objekt XMSCS_WMQ_MQMD_MESSAGE_CONTEXT nastaven na hodnotu XMSCS_WMQ_MDCTX_SET_ALL_CONTEXT:

- JMS_IBM_MQMD_PutApplTyp
- Název JMS_IBM_MQMD_PutAppl
- JMS_IBM_MQMD_PutDate .
- JMS_IBM_MQMD_PutTime .
- JMS_IBM_MQMD_ApplOriginData

Příjem zpráv

Všechny tyto vlastnosti jsou k dispozici v přijaté zprávě, je-li vlastnost XMSCS_WMQ_MQMD_READ_ENABLED nastavena na hodnotu true, a to bez ohledu na skutečné vlastnosti, které vytváří daná sada aplikací. Aplikace nemůže upravit vlastnosti přijaté zprávy, pokud nejsou nejprve vymazány všechny vlastnosti, podle specifikace JMS. Obdrženou zprávu lze předat bez úpravy vlastností.

Poznámka: Obdrží-li vaše aplikace zprávu z vlastnosti XMSCS_WMQ_MQMD_READ_ENABLED nastavenou na hodnotu true a předá ji cíli s hodnotou XMSCS_WMQ_MQMD_WRITE_ENABLED nastavenou na hodnotu

true, budou mít tyto výsledky ve všech hodnotách polí MQMD přijaté zprávy do předané zprávy. Tabulka vlastností

<i>Tabulka 31. Vlastnosti objektu zprávy reprezentující pole MQMD</i>		
Vlastnost	Popis	Typ
ZPRÁVA JMS_IBM_MQMD_REPORT	Volby pro zprávy sestav	System.Int32
JMS_IBM_MQMD_MSGTYPE	Typ zprávy	System.Int32
VYPRŠENÍ PLATNOSTI JMS_IBM_MQMD_EXPIRY	doba trvání zprávy	System.Int32
ZPĚTNÁ VAZBA JMS_IBM_MQMD_FEEDBACK	Zpětná vazba nebo kód příčiny	System.Int32
KÓDOVÁNÍ JMS_IBM_MQMD_ENCODING	Číselný kódování dat zprávy	System.Int32
JMS_IBM_MQMD_CODEDCHARSETID	Identifikátor znakové sady dat zprávy	System.Int32
FORMÁT JMS_IBM_MQMD_FORMAT	Název formátu dat zprávy	System.String
PRIORITA JMS_IBM_MQMD_PRIORITY Poznámka: Přiřadíte-li hodnotu JMS_IBM_MQMD_PRIORITY, která není v rozsahu 0-9, tato hodnota porušuje specifikaci JMS.	Priorita zprávy	System.Int32
PERZISTENTNÍ PERZISTENCE JMS_IBM_MQMD_	Trvalost zpráv	System.Int32
MSGID JMS_IBM_MQMD_MSGID Poznámka: Ve specifikaci platformy JMS je uvedeno, že ID zprávy musí být nastaveno poskytovatelem rozhraní JMS a že musí být buď jedinečné, nebo musí mít hodnotu null. Přiřadíte-li hodnotu JMS_IBM_MQMD_MSGID, bude tato hodnota zkopírována do hodnoty JMSMessageID. Takže není nastaven poskytovatelem JMS a nemusí být jedinečná: tato hodnota porušuje specifikaci JMS.	Identifikátor zprávy	Bajtové pole Poznámka: Použití vlastností bajtového pole ve zprávě porušuje specifikaci platformy JMS.
JMS_IBM_MQMD_CORRELID Poznámka: Přiřadíte-li hodnotu JMS_IBM_MQMD_CORRELID, která začíná řetězcem 'ID:', tato hodnota porušuje specifikaci JMS.	Identifikátor korelace	Bajtové pole Poznámka: Použití vlastností bajtového pole ve zprávě porušuje specifikaci platformy JMS.
JMS_IBM_MQMD_BACKOUTCOUNT	Počítadlo odvolaných	System.Int32
JMS_IBM_MQMD_REPLYTOQ	Název fronty odpovědí	System.String
JMS_IBM_MQMD_REPLYTOQMGR	Název správce front odpovědí	System.String
JMS_IBM_MQMD_USERIDENTIFIER	Identifikátor uživatele	System.String

Tabulka 31. Vlastnosti objektu zprávy reprezentující pole MQMD (pokračování)		
Vlastnost	Popis	Typ
ROZLIŠUJÍCÍ TOKEN JMS_IBM_MQMD_ACCOUNTINGTOKEN	Token evidence	Bajtové pole Poznámka: Použití vlastností bajtového pole ve zprávě porušuje specifikaci platformy JMS.
JMS_IBM_MQMD_APPLIDENTITYDATA	Údaje o žádosti vztahující se k totožnosti	System.String
JMS_IBM_MQMD_PUTAPPLTYPE	Typ aplikace, která vložila zprávu	System.Int32
JMS_IBM_MQMD_PUTAPPLNAME	Název aplikace, která vložila zprávu	System.String
JMS_IBM_MQMD_PUTDATE	Datum, kdy byla zpráva vložena	System.String
ČAS PUTTIME JMS_IBM_MQMD_PUTTIME	Čas, kdy byla zpráva vložena	System.String
JMS_IBM_MQMD_APPLORIGINDATA	Údaje o žádosti vztahující se k původu	System.String
JMS_IBM_MQMD_GROUPID	Identifikátor skupiny	Bajtové pole Poznámka: Použití vlastností bajtového pole ve zprávě porušuje specifikaci platformy JMS.
JMS_IBM_MQMD_MSGSEQNUMBER	Pořadové číslo lokální zprávy v rámci skupiny	System.Int32
POSUNUTÍ JMS_IBM_MQMD_OFFSET	Posunutí dat ve fyzické zprávě od začátku logické zprávy	System.Int32
PARAMETRY JMS_IBM_MQMD_MSGFLAGS	Příznaky zprávy	System.Int32
JMS_IBM_MQMD_ORIGINALLENGTH	Délka původní zprávy	System.Int32

Další podrobnosti viz [MQMD](#).

Příklady

Tento příklad vede k vložení zprávy do fronty nebo tématu s MQMD.UserIdentifier nastaven na "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
// ...
```

Před nastavením parametru JMS_IBM_MQMD_USERIDENTIFIER je třeba nastavit hodnotu XMSCS_WMQ_MQMD_MESSAGE_CONTEXT. Další informace o použití příkazu XMSCS_WMQ_MQM_MESSAGE_CONTEXT viz Vlastnosti objektu zprávy.

Podobně můžete extrahovat obsah polí MQMD nastavením hodnoty XMSCS_WMQ_MQMD_READ_ENABLED na hodnotu true před přijetím zprávy a poté použitím metod get zprávy, jako je například vlastnost getString. Všechny přijaté vlastnosti jsou jen pro čtení.

Tento příklad vede k hodnotě pole MQMD.ApplIdentityData pole zprávy bylo získáno z fronty nebo tématu.

```
// 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);
```

Vlastnosti objektu MessageConsumer

Přehled vlastností objektu MessageConsumer s odkazy na podrobnější referenční informace.

Tabulka 32. Vlastnosti objektu MessageConsumer	
Název vlastnosti	Popis
<u>XMSCS_IS_SUBSCRIPTION_MULTICAST</u>	Označuje, zda jsou zprávy doručovány spotřebiteli zpráv pomocí produktu Přenos výběrového vysílání WebSphere MQ. Tato vlastnost je jen pro čtení.
<u>VÝJIM_PODSÍTY_XMSCS_ISLEABLE_MULTICAST</u>	Označuje, zda jsou zprávy doručovány spotřebiteli zpráv pomocí produktu Přenos výběrového vysílání WebSphere MQ se spolehlivou kvalitou služeb. Tato vlastnost je jen pro čtení.

Postupujte podle části [.Vlastnosti prostředí NET pro IMessageConsumer](#) jsou uvedeny v dalších podrobnostech.

Vlastnosti objektu MessageProducer

Přehled vlastností objektu MessageProducer s odkazy na podrobnější referenční informace.

Viz [.Vlastnosti prostředí NET IMessageProducer](#) pro více podrobností.

Vlastnosti relace

Přehled vlastností objektu Relace s odkazy na podrobnější referenční informace.

Viz [.Další podrobnosti lze najít v tématu NET properties of ISession](#) .

Definice vlastností

Tato sekce poskytuje definici každé vlastnosti objektu.

Každá definice vlastnosti obsahuje následující informace:

- Datový typ vlastnosti.
- Typy objektů, které mají vlastnost
- Pro vlastnost Destination zadejte název, který lze použít v identifikátoru URI (Uniform Resource Identifier).
- Podrobnější popis vlastnosti
- Platné hodnoty vlastnosti
- Výchozí hodnota vlastnosti.

Vlastnosti, jejichž názvy začínají na jednu z následujících předpon, jsou relevantní pouze pro daný typ připojení:

XMSC_RTT

Vlastnosti jsou relevantní pouze pro připojení v reálném čase ke zprostředkovateli. Názvy vlastností jsou definovány jako pojmenované konstanty v hlavičkovém souboru `xmsc_rtt.h`.

XMSC_WMQ

Vlastnosti jsou relevantní pouze v případě, že se aplikace připojuje ke správci front produktu WebSphere MQ. Názvy vlastností jsou definovány jako pojmenované konstanty v hlavičkovém souboru `xmsc_wmq.h`.

XMSC_WPM

Vlastnosti jsou relevantní pouze v případě, že se aplikace připojuje ke sběrnici pro integraci služeb produktu WebSphere. Názvy vlastností jsou definovány jako pojmenované konstanty v hlavičkovém souboru `xmsc_wpm.h`.

Nejsou-li ve svých definicích uvedeny jinak, jsou zbývající vlastnosti relevantní pro všechny typy připojení. Názvy vlastností jsou definovány jako pojmenované konstanty v hlavičkovém souboru `xmsc.h`. Vlastnosti, jejichž názvy začínají předponou JMSX jsou JMS definované vlastnosti zprávy, a vlastnosti, jejichž názvy začínají předponou JMS_IBM jsou IBM definované vlastnosti zprávy. Další informace o vlastnostech zpráv viz [“Vlastnosti zprávy XMS” na stránce 68](#).

Není-li ve své definici uvedeno jinak, je každá vlastnost relevantní jak v doménách dvoubodový, tak v doméně Publikování/odběr.

Aplikace může získat a nastavit hodnotu libovolné vlastnosti, pokud tato vlastnost není označena jako jen pro čtení.

JMS_IBM_CHARACTER_SET

Datový typ:

System.Int32

Vlastnost:

Zpráva

Identifikátor (CCSID) kódované znakové sady nebo kódové stránky, že řetězce znakových dat v těle zprávy jsou v době, kdy klient XMS předává zprávu do zamýšleného místa určení. V prostředí XMS má tato vlastnost číselnou hodnotu a mapuje se na CCSID. Tato vlastnost je však založena na vlastnosti JMS, takže má řetězcovou hodnotu typu a mapuje se na znakovou sadu Java, která představuje tento numerický identifikátor CCSID. Tato vlastnost přepíše jakékoli CCSID uvedené pro cíl vlastností `XMSCS_WMQ_CCSD`.

Ve výchozím nastavení není vlastnost nastavena.

Tato vlastnost není relevantní, když se aplikace připojuje ke sběrnici pro integraci služeb.

JMS_IBM_Encoding

Datový typ:

System.Int32

Vlastnost:

Zpráva

Jak číselná data v těle zprávy jsou představována, když klient produktu XMS předává zprávu do zamýšleného cíle. Tato vlastnost přepíše jakékoli kódování určené pro místo určení pomocí vlastnosti `XMSC_WMQ_ENCODING`. Vlastnost uvádí znázornění binárních celých čísel, pakovaných dekadických celých čísel a čísel s pohyblivou řádovou čárkou.

Platné hodnoty vlastnosti jsou stejné jako hodnoty, které lze zadat v poli **Encoding** deskriptoru zpráv.

Aplikace může pro nastavení vlastnosti použít následující pojmenované konstanty:

pojmenovaná konstanta	Význam
<code>MQENC_INTEGER_NORMAL</code>	Normální celočíselné kódování
<code>MQENC_INTEGER_REVERSED</code>	Obrácené celočíselné kódování
<code>MQENC_DECIMAL_NORMAL</code>	Normální pakované kódování desetinných čísel
<code>MQENC_DECIMAL_REVERSED</code>	Obrácené pakované kódování desetinných čísel
<code>MQENC_FLOAT_IEEEEE_NORMAL</code>	Normální kódování čísel s plovoucí desetinnou čárkou
<code>MQENC_FLOAT_IEEEE_OBRÁCENÝ</code>	Obrácené kódování čísel s pohyblivou řádovou čárkou IEEE
<code>MQENC_FLOAT_S390</code>	Kódování čísel s pohyblivou řádovou čárkou z/OS
<code>MQENC_NATIVE</code>	Kódování nativního počítače

Chcete-li vytvořit hodnotu vlastnosti, může aplikace přidat tři z těchto konstant takto:

- Konstanta, jejíž název začíná řetězcem `MQENC_INTEGER`, určuje reprezentaci binárních celých čísel.
- Konstanta, jejíž název začíná řetězcem `MQENC_DECIMAL`, určuje reprezentaci pakovaných dekadických celých čísel.
- Konstanta, jejíž název začíná řetězcem `MQENC_FLOAT`, určuje reprezentaci čísel s pohyblivou řádovou čárkou.

Alternativně může aplikace nastavit vlastnost na hodnotu `MQENC_NATIVE`, jejíž hodnota závisí na prostředí.

Ve výchozím nastavení není vlastnost nastavena.

Tato vlastnost není relevantní, když se aplikace připojuje ke sběrnici pro integraci služeb.

JMS_IBM_EXCEPTIONMESSAGE

Datový typ:

Řetězec

Vlastnost:

Zpráva

Text, který popisuje, proč byla zpráva odeslána do cíle výjimek. Tato vlastnost je jen pro čtení.

Tato vlastnost je relevantní pouze v případě, že se aplikace připojuje ke sběrnici pro integraci služeb a přijímá zprávu z místa určení výjimek.

NÁZEV SOUBORU JMS_IBM_EXCEPTION

Datový typ:

Řetězec

Vlastnost:

Zpráva

Název cíle, ve kterém byla zpráva před odesláním zprávy do cíle výjimek.

Tato vlastnost je relevantní pouze v případě, že se aplikace připojuje ke sběrnici pro integraci služeb a přijímá zprávu z místa určení výjimek.

JMS_IBM_EXCEPTIONREASON

Datový typ:

System.Int32

Vlastnost:

Zpráva

Kód příčiny označující příčinu, proč byla zpráva odeslána do cíle výjimek.

Tato vlastnost je relevantní pouze v případě, že se aplikace připojuje ke sběrnici pro integraci služeb a přijímá zprávu z místa určení výjimek.

JMS_IBM_EXCEPTIONTIMESTAMP

Datový typ:

System.Int64

Vlastnost:

Zpráva

Čas, kdy byla zpráva odeslána do cíle výjimek.

Čas je vyjádřen v milisekundách od 00:00:00 GMT k 1. lednu 1970.

Tato vlastnost je relevantní pouze v případě, že se aplikace připojuje ke sběrnici pro integraci služeb a přijímá zprávu z místa určení výjimek.

JMS_IBM_FEEDBACK

Datový typ:

System.Int32

Vlastnost:

Zpráva

Kód, který označuje chování zprávy sestavy.

Platné hodnoty této vlastnosti jsou kódy zpětné vazby a kódy příčiny, které lze zadat v poli **Feedback** deskriptoru zpráv.

Ve výchozím nastavení není vlastnost nastavena.

FORMÁT JMS_IBM_FORMAT

Datový typ:

Řetězec

Vlastnost:

Zpráva

Povahu dat aplikace ve zprávě.

Platné hodnoty vlastnosti jsou stejné jako hodnoty, které lze zadat v poli **Format** deskriptoru zpráv.

Ve výchozím nastavení není vlastnost nastavena.

Tato vlastnost není relevantní, když se aplikace připojuje ke sběrnici pro integraci služeb.

JMS_IBM_LAST_MSG_IN_GROUP

Datový typ:

System.Boolean

Vlastnost:

Zpráva

Označuje, zda je zpráva poslední zprávou ve skupině zpráv.

Nastavte vlastnost na hodnotu true, pokud se jedná o poslední zprávu ve skupině zpráv. Jinak nastavte vlastnost na hodnotu false, nebo vlastnost nenastavujte. Ve výchozím nastavení není vlastnost nastavena.

Hodnota true odpovídá příznaku stavu MQMF_LAST_MSG_IN_GROUP, který lze zadat v poli **MsgFlags** deskriptoru zpráv.

Tato vlastnost se ignoruje v doméně Publikování/odběr a není relevantní, když se aplikace připojuje ke sběrnici pro integraci služeb.

JMS_IBM_MSGTYPE

Datový typ:

System.Int32

Vlastnost:

Zpráva

Typ zprávy.

Platné hodnoty vlastnosti jsou následující:

Platná hodnota	Význam
MQM_DATAGRAM	Zpráva je taková, která nevyžaduje odpověď.
POŽADAVEK MQMT_REQUEST	Zpráva je taková, která vyžaduje odpověď.
MQMT_REPLY	Zpráva je zpráva odpovědi.
SESTAVA MQMT_REPORT	Zpráva je zpráva hlášení.

Tyto hodnoty odpovídají typům zpráv, které lze zadat v poli **MsgType** deskriptoru zpráv.

Ve výchozím nastavení není vlastnost nastavena.

Tato vlastnost není relevantní, když se aplikace připojuje ke sběrnici pro integraci služeb.

JMS_IBM_PUTAPPLTYPE

Datový typ:

System.Int32

Vlastnost:

Zpráva

Typ aplikace, která odeslala zprávu.

Platnými hodnotami vlastnosti jsou typy aplikací, které lze zadat v poli **PutAppType** deskriptoru zpráv.

Ve výchozím nastavení není vlastnost nastavena.

Tato vlastnost není relevantní, když se aplikace připojuje ke sběrnici pro integraci služeb.

JMS_IBM_PUTDATE

Datový typ:

Řetězec

Vlastnost:

Zpráva

Datum, kdy byla zpráva odeslána.

Platné hodnoty vlastnosti jsou stejné jako hodnoty, které lze zadat v poli **PutDate** deskriptoru zpráv.

Ve výchozím nastavení není vlastnost nastavena.

Tato vlastnost není relevantní, když se aplikace připojuje ke sběrnici pro integraci služeb.

ČAS SOUBORU JMS_IBM_PUTTIME

Datový typ:

Řetězec

Vlastnost:

Zpráva

Čas, kdy byla zpráva odeslána.

Platné hodnoty vlastnosti jsou stejné jako hodnoty, které lze zadat v poli **PutTime** deskriptoru zpráv.

Ve výchozím nastavení není vlastnost nastavena.

Tato vlastnost není relevantní, když se aplikace připojuje ke sběrnici pro integraci služeb.

JMS_IBM_REPORT_COA

Datový typ:

System.Int32

Vlastnost:

Zpráva

Požadavek na zprávy sestavy 'potvrdit při příjmu' určující, kolik dat aplikace z původní zprávy musí být zahrnuto do zprávy sestavy.

Platné hodnoty vlastnosti jsou následující:

Platná hodnota	Význam
MQRO_COA	Požadavek 'potvrdit při příchodu' zprávy hlášení, bez dat aplikace z původní zprávy obsažené ve zprávě sestavy.
MQRO_COA_WITH_DATA	Požadavek 'potvrdit při příchodu' zprávy hlášení, s prvními 100 bajty dat aplikace z původní zprávy zahrnuté ve zprávě sestavy.
MQRO_COA_WITH_FULL_DATA	Požadavek 'potvrdit při příchodu' zprávy hlášení se všemi daty aplikace z původní zprávy, která je zahrnuta ve zprávě sestavy.

Tyto hodnoty odpovídají volbám sestavy, které lze zadat v poli **Report** deskriptoru zpráv. Další informace o těchto volbách viz [Sestava \(MQLONG\)](#).

Ve výchozím nastavení není vlastnost nastavena.

JMS_IBM_REPORT_COD

Datový typ:

System.Int32

Vlastnost:

Zpráva

Požadavek na zprávy sestavy 'potvrdit při doručení' určující, kolik dat aplikace z původní zprávy musí být zahrnuto do zprávy sestavy.

Platné hodnoty vlastnosti jsou následující:

Platná hodnota	Význam
MQRO_COD	Požadavek 'potvrdit na doručení' zprávy hlášení, bez dat aplikace z původní zprávy zahrnuté ve zprávě sestavy.
MQRO_COD_WITH_DATA	Požadavek 'potvrdit na doručení' zprávy hlášení, s prvními 100 bajty dat aplikace z původní zprávy zahrnuté ve zprávě sestavy.

Platná hodnota

MQRO_COD_WITH_FULL_DATA

Význam

Požadavek 'potvrdit na doručovací' zprávy hlášení se všemi daty aplikace z původní zprávy zahrnuté ve zprávě sestavy.

Tyto hodnoty odpovídají volbám sestavy, které lze zadat v poli **Report** deskriptoru zpráv.

Ve výchozím nastavení není vlastnost nastavena.

SKÓDOVÁ_ZPRÁVA_JMS_IBM_REPORT_DISCARD_MSG**Datový typ:**

System.Int32

Vlastnost:

Zpráva

Požadavek, aby byla zpráva vyřazena, pokud ji nelze doručit do zamýšleného cíle.

Nastavte vlastnost na hodnotu MQRO_DISCARD_MSG na žádost, aby byla zpráva vyřazena, pokud ji nelze doručit do zamýšleného místa určení. Pokud vyžadujete, aby byla zpráva vložena do fronty nedoručených zpráv nebo byla odeslána do místa určení výjimek, nenastavujte vlastnost. Ve výchozím nastavení není vlastnost nastavena.

Hodnota MQRO_DISCARD_MSG odpovídá volbě sestavy, která může být zadána v poli **Report** deskriptoru zpráv.

VÝJIMKA_JMS_IBM_REPORT_EXCEPTION**Datový typ:**

System.Int32

Vlastnost:

Zpráva

Požadavek na zprávy sestavy výjimky určující, kolik dat aplikace z původní zprávy musí být zahrnuto do zprávy sestavy.

Platné hodnoty vlastnosti jsou následující:

Platná hodnota

VÝJIMKA MQRO_EXCEPTION

Význam

Vyžádejte zprávy o výjimce, a to bez dat aplikace z původní zprávy obsažené ve zprávě sestavy.

MQRO_EXCEPTION_WIT_DATA

Vyžádejte si zprávy o výjimce požadavku s prvními 100 bajty dat aplikace z původní zprávy obsažené ve zprávě sestavy.

MQRO_EXCEPTION_WITH_FULL_DATA

Vyžádat zprávy o zprávě výjimky se všemi daty aplikace z původní zprávy obsažené ve zprávě sestavy.

Tyto hodnoty odpovídají volbám sestavy, které lze zadat v poli **Report** deskriptoru zpráv.

Ve výchozím nastavení není vlastnost nastavena.

JMS_IBM_REPORT_EXPIRATION**Datový typ:**

System.Int32

Vlastnost:

Zpráva

Požadavek na zprávy sestavy vypršení platnosti určující, kolik dat aplikace z původní zprávy musí být zahrnuto do zprávy sestavy.

Platné hodnoty vlastnosti jsou následující:

Platná hodnota	Význam
MQRO_EXPIRATION	Vyžádejte zprávy o vypršení platnosti požadavku, bez dat aplikace z původní zprávy obsažené ve zprávě sestavy.
MQRO_EXPIRATION_WITH_DATA	Vyžádat zprávy o vypršení platnosti požadavku, s prvními 100 bajty dat aplikace z původní zprávy obsažené ve zprávě sestavy.
MQRO_EXPIRATION_WITH_FULL_DATA	Vyžádat zprávy o vypršení platnosti požadavku se všemi daty aplikace z původní zprávy obsažené ve zprávě sestavy.

Tyto hodnoty odpovídají volbám sestavy, které lze zadat v poli **Report** deskriptoru zpráv.

Ve výchozím nastavení není vlastnost nastavena.

JMS_IBM_REPORT_NAN

Datový typ:

System.Int32

Vlastnost:

Zpráva

Požadavek na zprávy sestavy negativního oznámení na akci.

Nastavte vlastnost na hodnotu MQRO_NAN tak, aby bylo možné požadovat zprávy s oznámením o negativních akcích. Pokud nevyžadujete zprávy sestavy upozornění na negativní akci, nenastavujte tuto vlastnost. Ve výchozím nastavení není vlastnost nastavena.

Hodnota MQRO_NAN odpovídá volbě sestavy, která může být zadána v poli **Report** deskriptoru zpráv.

JMS_IBM_REPORT_PAN

Datový typ:

System.Int32

Vlastnost:

Zpráva

Požadavek na zprávy sestavy pozitivního oznámení na akci.

Nastavte vlastnost na hodnotu MQRO_PAN a vyžádejte si zprávy s oznámením o kladných akcích. Pokud nevyžadujete zprávy sestav s kladným oznámením, nenastavujte vlastnost. Ve výchozím nastavení není vlastnost nastavena.

Hodnota MQRO_PAN odpovídá volbě sestavy, která může být zadána v poli **Report** deskriptoru zpráv.

JMS_IBM_REPORT_PASS_CORREL_ID

Datový typ:

System.Int32

Vlastnost:

Zpráva

Požadavek, aby identifikátor korelace jakékoli sestavy nebo zprávy odpovědi byl stejný jako identifikátor korelace původní zprávy.

Platné hodnoty vlastnosti jsou následující:

Platná hodnota

ID_KOLEKCE_MQRO_PASS_RELACE_

MQRO_COPY_MSG_ID_TO_CORREL_ID

Význam

Požadavek, aby identifikátor korelace jakékoli sestavy nebo zprávy odpovědi byl stejný jako identifikátor korelace původní zprávy.

Požadujte, aby korelační identifikátor jakékoli sestavy nebo zprávy odpovědi byl stejný jako identifikátor zprávy původní zprávy.

Tyto hodnoty odpovídají volbám sestavy, které lze zadat v poli **Report** deskriptoru zpráv.

Výchozí hodnota vlastnosti je MQRO_COPY_MSG_ID_TO_CORREL_ID.

JMS_IBM_REPORT_PASS_MSG_ID**Datový typ:**

System.Int32

Vlastnost:

Zpráva

Požadavek, aby identifikátor zprávy jakékoli sestavy nebo zprávy odpovědi byl stejný jako identifikátor zprávy původní zprávy.

Platné hodnoty vlastnosti jsou následující:

Platná hodnota

MQRO_PASS_MSG_ID

MQRO_NEW_MSG_ID

Význam

Požadavek, aby identifikátor zprávy jakékoli sestavy nebo zprávy odpovědi byl stejný jako identifikátor zprávy původní zprávy.

Požadavek na vygenerování nového identifikátoru zprávy pro každou zprávu nebo zprávu s odpovědí.

Tyto hodnoty odpovídají volbám sestavy, které lze zadat v poli Sestava v deskriptoru zpráv.

Výchozí hodnota vlastnosti je MQRO_NEW_MSG_ID.

JMS_IBM_RETAIN**Datový typ:**

System.Int32

Vlastnost:

Zpráva

Nastavení této vlastnosti označuje, správce front bude zprávu považovat za Zachované publikování. Když odběratel obdrží zprávy z témat, může okamžitě po odběru obdržet další zprávy, kromě zpráv obdržených v předchozích verzích. Tyto zprávy jsou nepovinnými zachovaným publikováním pro témata přihlášená k odběru. Pro každé téma, které odpovídá odběru, existuje zachované publikování, které je zpřístupněno k doručení odběratelnému odběrateli zpráv.

Hodnota RETAIN_PUBLIKACE je jedinou platnou hodnotou této vlastnosti. Tato vlastnost standardně není nastavena.

Poznámka: Tato vlastnost je relevantní pouze v doméně publikování/odběr.

JMS_IBM_SYSTEM_MESSAGEID**Datový typ:**

Řetězec

Vlastnost:

Zpráva

Identifikátor, který jedinečně identifikuje zprávu v rámci sběrnice SIBus. Tato vlastnost je jen pro čtení.

Tato vlastnost je relevantní pouze v případě, že se aplikace připojuje ke sběrnici pro integraci služeb.

JMSX_APPID

Datový typ:
Řetězec

Vlastnost:
Zpráva

Název aplikace, která odeslala zprávu.

Tato vlastnost je definovaná JMS vlastností s názvem JMS JMSXAppID. Další informace o vlastnosti naleznete v příručce *Java Message Service Specification, verze 1.1*.

Ve výchozím nastavení není vlastnost nastavena.

Tato vlastnost není platná pro připojení v reálném čase ke zprostředkovateli.

POČET DORUČENÍ JMSX_DELIVERY_COUNT

Datový typ:
System.Int32

Vlastnost:
Zpráva

Počet pokusů o doručení zprávy.

Tato vlastnost je definovaná JMS vlastností s názvem JMS JMSXDeliveryCount. Další informace o vlastnosti naleznete v příručce *Java Message Service Specification, verze 1.1*.

Ve výchozím nastavení není vlastnost nastavena.

Tato vlastnost není platná pro připojení v reálném čase ke zprostředkovateli.

JMSX_GROUPID

Datový typ:
Řetězec

Vlastnost:
Zpráva

Identifikátor skupiny zpráv, do které zpráva patří.

Tato vlastnost je definovaná JMS vlastností s názvem JMS JMSXGroupID. Další informace o vlastnosti naleznete v příručce *Java Message Service Specification, verze 1.1*.

Ve výchozím nastavení není vlastnost nastavena.

Tato vlastnost není platná pro připojení v reálném čase ke zprostředkovateli.

JMSX_GROUPSEQ

Datový typ:
System.Int32

Vlastnost:
Zpráva

Pořadové číslo zprávy v rámci skupiny zpráv.

Tato vlastnost je definovaná JMS vlastností s názvem JMS JMSXGroupSeq. Další informace o vlastnosti naleznete v příručce *Java Message Service Specification, verze 1.1*.

Ve výchozím nastavení není vlastnost nastavena.

Tato vlastnost není platná pro připojení v reálném čase ke zprostředkovateli.

ID UŽIVATELE JMSX_ID

Datový typ:

Řetězec

Vlastnost:

Zpráva

Identifikátor uživatele přidružený k aplikaci, která odeslala zprávu.

Tato vlastnost je definovaná JMS vlastností s názvem JMS JMSXUserID. Další informace o vlastnosti naleznete v příručce *Java Message Service Specification, verze 1.1*.

Ve výchozím nastavení není vlastnost nastavena.

Tato vlastnost není platná pro připojení v reálném čase ke zprostředkovateli.

XMSC_ASYNC_EXCEPTIONS

Datový typ:

System.Int32

Vlastnost:

ConnectionFactory

Použitelné objekty:

Dlouhý název nástroje pro administraci JMS: ASYNCEXCEPTION

Krátký název nástroje pro administraci JMS: AEX

Tato vlastnost určuje, zda rozhraní XMS informuje modul ExceptionListener pouze v případě, že dojde k přerušení připojení nebo když dojde k asynchronnímu výskytu jakékoli výjimky pro volání rozhraní XMS API. Tato vlastnost platí pro všechna připojení vytvořená z této továrny připojení, která mají registrován modul ExceptionListener.

Platné hodnoty pro tuto vlastnost jsou:

XMSC_ASYNC_EXCEPTIONS_ALL

Jakákoli výjimka byla zjištěna asynchronně, mimo rozsah volání synchronního volání rozhraní API a všechny přerušené výjimky připojení jsou odeslány do ExceptionListener.

XMSC_ASYNC_EXCEPTIONS_CONNECTIONBROKEN

Do modulu ExceptionListense odesílají pouze výjimky označující přerušené připojení. Jakékoli další výjimky, které se vyskytly během asynchronního zpracování, se do modulu ExceptionListenerneohlašují, a proto není aplikace informována o těchto výjimkách.

Při výchozím nastavení je tato vlastnost nastavena na hodnotu XMSCS_ASYNC_EXCEPTIONS_ALL.

XMSC_CLIENT_ID

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Použitelné objekty:

Dlouhý název nástroje pro administraci JMS: CLIENTID

Krátký název nástroje pro administraci JMS: CID

Identifikátor klienta pro účely připojení.

Identifikátor klienta se používá pouze k podpoře trvalých odběrů v doméně Publikování/odběr a je ignorován v doméně dvoubodový . Další informace o nastavení identifikátorů klienta viz [“ConnectionFactoryes a objekty připojení” na stránce 21](#).

Tato vlastnost není relevantní pro připojení v reálném čase ke zprostředkovateli.

XMSC_CONNECTION_TYPE

Datový typ:

System.Int32

Vlastnost:

ConnectionFactory

Typ serveru systému zpráv, ke kterému se aplikace připojuje.

Platné hodnoty vlastnosti jsou následující:

Platná hodnota	Význam
XMSC_CT_RTT	Připojení v reálném čase ke zprostředkovateli.
XMSC_CT_WMQ	Připojení ke správci front produktu WebSphere MQ .
XMSC_CT_WPM	Připojení ke sběrnici pro integraci služeb produktu WebSphere .

Ve výchozím nastavení není vlastnost nastavena.

XMSC_DELIVERY_MODE

Datový typ:

System.Int32

Vlastnost:

Místo určení

Název použitý v identifikátoru URI:

persistence (pro místo určení WebSphere MQ)

deliveryMode (pro výchozí místo určení poskytovatele systému zpráv produktu WebSphere)

Použitelné objekty:

Dlouhý název nástroje pro administraci JMS: PERSISTENCE

Krátký název nástroje pro administraci JMS: PER

Režim doručení zpráv odeslaných do cíle.

Platné hodnoty vlastnosti jsou následující:

Platná hodnota	Význam
XMSC_DELIVERY_NOT_PERSISTENT	Zpráva odeslaná do místa určení je přechodná. Výchozí režim doručení producenta zpráv nebo libovolný režim doručení zadaný v rámci volání Odeslat je ignorován. Je-li cílem fronty WebSphere MQ , je hodnota atributu fronty <i>DefPersistence</i> také ignorována.
XMSC_DELIVERY_PERSISTENT	Zpráva odeslaná do místa určení je trvalá. Výchozí režim doručení producenta zpráv nebo libovolný režim doručení zadaný v rámci volání Odeslat je ignorován. Je-li cílem fronty WebSphere MQ , je hodnota atributu fronty <i>DefPersistence</i> také ignorována.
XMSC_DELIVERY_AS_APP	Zpráva odeslaná do místa určení má režim doručení uvedený v odeslaném volání. Pokud volba Odeslat neuvádí žádný režim doručení, použije se výchozí režim doručení producenta zpráv. Je-li cílem fronty WebSphere MQ , hodnota atributu fronty <i>DefPersistence</i> se ignoruje.

Platná hodnota

XMSC_DELIVERY_AS_DEST

Význam

Je-li cílem fronty produktu WebSphere MQ , má zpráva vkládaná do fronty režim doručení určený hodnotou atributu fronty *DefPersistence*. Výchozí režim doručení producenta zpráv nebo libovolný režim doručení zadaný v rámci volání Odeslat je ignorován.

Pokud cíl není frontou WebSphere MQ , význam je stejný jako ve formátu XMSCS_DELIVERY_AS_APP.

Výchozí hodnota je XMSCS_DELIVERY_AS_APP.

XMSC_IC_CASH_**Datový typ:**

Řetězec

Vlastnost:

InitialContext

Použije se k vyhledání adresáře pojmenování rozhraní JNDI tak, aby služba pojmenování COS nevyžadovala být na stejném serveru jako webová služba.

XMSC_IC_SECURITY_AUTHENTICATION**Datový typ:**

Řetězec

Vlastnost:

InitialContext

Na základě kontextu Java Kontextové rozhraní SECURITY_AUTHENTICATION. Tato vlastnost je použitelná pouze pro kontext pojmenování COS.

XMSC_IC_SECURITY_CREDENTIALS**Datový typ:**

Řetězec

Vlastnost:

InitialContext

Založeno na kontextovém rozhraní Java SECURITY_CREDENTIALS. Tato vlastnost je použitelná pouze pro kontext pojmenování COS.

XMSC_IC_SECURITY_PRINCIPAL**Datový typ:**

Řetězec

Vlastnost:

InitialContext

Založeno na kontextu Java kontextu SECURITY_PRINCIPAL. Tato vlastnost je použitelná pouze pro kontext pojmenování COS.

XMSC_IC_SECURITY_PROTOCOL**Datový typ:**

Řetězec

Vlastnost:

InitialContext

Založeno na kontextu Java kontextu SECURITY_PROTOCOL Tato vlastnost je použitelná pouze pro kontext pojmenování COS.

XMSC_ICURL

Datový typ:

Řetězec

Vlastnost:

InitialContext

Pro kontexty LDAP a FileSystem, adresa úložiště obsahující administrované objekty.

Pro kontexty pojmenování COS, adresa webové služby, která vyhledává objekty v adresáři.

XMSC_IS_SUBSCRIPTION_MULTICAST

Datový typ:

System.Boolean

Vlastnost:

MessageConsumer

Označuje, zda jsou zprávy doručovány spotřebiteli zpráv pomocí produktu Přenos výběrového vysílání WebSphere MQ. Tato vlastnost je jen pro čtení.

Hodnota vlastnosti je true, pokud jsou zprávy doručovány spotřebiteli zpráv pomocí produktu Přenos výběrového vysílání WebSphere MQ. Jinak je hodnota false.

Tato vlastnost je relevantní pouze pro připojení v reálném čase ke zprostředkovateli.

XMSC_IS_SUBSCRIPTION_RELIABLE_MULTICAST

Datový typ:

System.Boolean

Vlastnost:

MessageConsumer

Označuje, zda jsou zprávy doručovány spotřebiteli zpráv pomocí produktu Přenos výběrového vysílání WebSphere MQ se spolehlivou kvalitou služeb. Tato vlastnost je jen pro čtení.

Hodnota vlastnosti je true, pokud jsou zprávy doručovány spotřebiteli zpráv pomocí produktu Přenos výběrového vysílání WebSphere MQ se spolehlivou kvalitou služeb. Jinak je hodnota false.

Tato vlastnost je relevantní pouze pro připojení v reálném čase ke zprostředkovateli.

XMSC_JMS_MAJOR_VERSION

Datový typ:

System.Int32

Vlastnost:

Data ConnectionMeta

Hlavní číslo verze specifikace JMS , na které je založena XMS . Tato vlastnost je jen pro čtení.

XMSC_JMS_MINOR_VERSION

Datový typ:

System.Int32

Vlastnost:

Data ConnectionMeta

Vedlejší číslo verze specifikace JMS , na které je založena XMS . Tato vlastnost je jen pro čtení.

XMSC_JMS_

Datový typ:

Řetězec

Vlastnost:

Data ConnectionMeta

Identifikátor verze specifikace JMS , na které je založena XMS . Tato vlastnost je jen pro čtení.

XMSC_HLAVNÍ_VERZE

Datový typ:

System.Int32

Vlastnost:

Data ConnectionMeta

Číslo verze klienta XMS . Tato vlastnost je jen pro čtení.

XMSC_MINOR_VERSION

Datový typ:

System.Int32

Vlastnost:

Data ConnectionMeta

Číslo verze klienta XMS . Tato vlastnost je jen pro čtení.

XMSC_PASSWORD

Datový typ:


Bajtové pole

Vlastnost:

ConnectionFactory

Heslo, které lze použít k ověření aplikace při pokusu o připojení k serveru systému zpráv. Heslo se používá s vlastností XMSCS_USERID .

Ve výchozím nastavení není vlastnost nastavena.

 Připojete-li se k produktu WebSphere MQ v systému Multiplatformy nastavíte vlastnost XMSCS_USERID továrny připojení, musí se shodovat s hodnotou **userid** přihlášeného uživatele. Pokud tyto vlastnosti nenastavíte, správce front použije při výchozím nastavení **userid** přihlášeného uživatele. Požadujete-li další ověření na úrovni připojení pro jednotlivé uživatele, můžete napsat uživatelskou proceduru pro ověření klienta, která je konfigurována v produktu WebSphere MQ. Další informace o vytvoření uživatelské procedury ověření klienta můžete zjistit v tématu Ověření v manuálu Klienti produktu WebSphere MQ .

Chcete-li ověřit uživatele při připojení k produktu WebSphere MQ v systému z/OS , je třeba použít uživatelskou proceduru pro zabezpečení zprávy.

XMSC_PRIORITY

Datový typ:

System.Int32

Vlastnost:

Místo určení

Název použitý v identifikátoru URI:

priority (priorita)

Priorita zpráv odeslaných do cíle.

Platné hodnoty vlastnosti jsou následující:

Platná hodnota

Celé číslo v rozsahu 0, nejnižší priorita, až 9, nejvyšší priorita

Význam

Zpráva odeslaná do místa určení má určenou prioritu. Výchozí priorita producentu zpráv nebo libovolná priorita zadaná v rámci výzvy k odeslání je ignorována. Je-li cílem fronta WebSphere MQ , je hodnota atributu fronty **DefPriority** také ignorována.

XMSC_PRIORITY_AS_APP

Zpráva odeslaná do místa určení má prioritu určenou v rámci volání Odeslat. Pokud volání typu odeslání určuje žádnou prioritu, použije se výchozí priorita producenta zpráv. Je-li cílem fronta WebSphere MQ , hodnota atributu fronty **DefPriority** se ignoruje.

XMSC_PRIORITY_AS_DEST

Je-li cílem fronta produktu WebSphere MQ , má zpráva uvedená ve frontě prioritu uvedenou v hodnotě atributu fronty **DefPriority**. Výchozí priorita producentu zpráv nebo libovolná priorita zadaná v rámci výzvy k odeslání je ignorována.

Pokud cíl není frontou WebSphere MQ , význam je stejný jako ve formátu XMSCS_PRIORITY_AS_APP.

Výchozí hodnota je XMSCS_PRIORITY_AS_APP.

WebSphere MQ Real-Time Transport a Přenos výběrového vysílání WebSphere MQ nepřijímají žádné akce založené na prioritě zprávy.

XMSC_NÁZEV_POSKYTOVATELE**Datový typ:**

Řetězec

Vlastnost:

Data ConnectionMeta

Poskytovatel klienta XMS . Tato vlastnost je jen pro čtení.

XMSC_RTC_BROKER_PING_INTERVAL**Datový typ:**

System.Int32

Vlastnost:

ConnectionFactory

Časový interval v milisekundách, po který XMS .NET kontroluje připojení k serveru systému zpráv v reálném čase ke zjištění jakékoli aktivity. Není-li zjištěna žádná aktivita, klient iniciuje příkaz ping; připojení se zavře, pokud není zjištěna žádná odezva na příkaz ping.

Výchozí hodnota vlastnosti je 30000.

XMSC_RTT_CONNECTION_PROTOCOL**Datový typ:**

System.Int32

Vlastnost:

ConnectionFactory

Komunikační protokol použitý pro připojení v reálném čase ke zprostředkovateli.

Hodnota vlastnosti musí být XMSCS_RTT_CP_TCP, což znamená v reálném čase připojení k zprostředkovateli prostřednictvím protokolu TCP/IP. Výchozí hodnota je XMSCS_RTT_CP_TCP.

XMSC_RTT_HOST_NAME

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Název nebo adresa IP hostitele systému, na kterém je spuštěn zprostředkovatel.

Tato vlastnost se používá spolu s vlastností XMSCS_RTT_PORT k identifikaci zprostředkovatele.

Ve výchozím nastavení není vlastnost nastavena.

XMSC_RTT_LOKÁLNÍ_ADRESA

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Název hostitele nebo adresa IP lokálního síťového rozhraní, které má být použito pro připojení v reálném čase ke zprostředkovateli.

Tato vlastnost je užitečná pouze v případě, že systém, na kterém je aplikace spuštěna, má dvě nebo více síťových rozhraní a vy potřebujete mít možnost určit, které rozhraní musí být použito pro připojení v reálném čase. Má-li systém pouze jedno síťové rozhraní, lze použít pouze toto rozhraní. Pokud má systém dvě nebo více síťových rozhraní a tato vlastnost není nastavena, je rozhraní vybráno náhodně.

Ve výchozím nastavení není vlastnost nastavena.

XMSC_RTT_MULTICAST

Datový typ:

System.Int32

Vlastnost:

ConnectionFactory a Destination

Název použitý v identifikátoru URI:

musikasa

Nastavení výběrového vysílání pro továrnu připojení nebo cíl. Pouze místo určení, které je téma, může mít tuto vlastnost.

Aplikace používá tuto vlastnost k povolení výběrového vysílání ve spojení se zprostředkovatelem v reálném čase a, je-li povoleno výběrové vysílání, určujete přesný způsob, jakým se výběrové vysílání používá k doručování zpráv od zprostředkovatele ke spotřebiteli zpráv. Vlastnost nemá žádný vliv na způsob, jakým Producent zpráv odesílá zprávy do zprostředkovatele.

Platné hodnoty vlastnosti jsou následující:

Platná hodnota

XMSC_RTT_MULTICAST_DISABLED

Význam

Zprávy nejsou doručovány spotřebiteli zpráv pomocí produktu Přenos výběrového vysílání WebSphere MQ. Tato hodnota je výchozí hodnotou pro objekt ConnectionFactory .

Platná hodnota

XMSC_RTT_MULTICAST_ASCF

Význam

Zprávy jsou doručovány spotřebiteli zpráv v souladu s nastavením výběrového vysílání pro továrnu připojení přidruženou k odběrateli zpráv. Výběrové nastavení multicast pro továrnu připojení je zaznamenáno v době vytvoření připojení. Tato hodnota je platná pouze pro cílový objekt a je výchozí hodnotou pro cílový objekt.

XMSC_RTT_MULTICAST_ENABLED

Je-li téma konfigurováno pro výběrové vysílání ve zprostředkovateli, jsou zprávy doručovány spotřebiteli zpráv pomocí produktu Přenos výběrového vysílání WebSphere MQ. Spolehlivá kvalita služeb se používá, je-li téma nakonfigurováno pro spolehlivé výběrové vysílání.

XMSC_RTT_MULTICAST_RELIABLE

Je-li téma nakonfigurováno pro spolehlivé výběrové vysílání ve zprostředkovateli, jsou zprávy doručovány spotřebiteli zpráv pomocí produktu Přenos výběrového vysílání WebSphere MQ se spolehlivou kvalitou služeb. Není-li téma konfigurováno pro spolehlivé výběrové vysílání, nemůžete pro dané téma vytvořit spotřebitele zpráv.

XMSC_RTT_MULTICAST_NOT_RELIABLE

Je-li téma konfigurováno pro výběrové vysílání ve zprostředkovateli, jsou zprávy doručovány spotřebiteli zpráv pomocí produktu Přenos výběrového vysílání WebSphere MQ. Spolehlivá kvalita služeb se nepoužívá ani v případě, že je téma nakonfigurováno pro spolehlivé výběrové vysílání.

XMSC_RTT_PORT**Datový typ:**

System.Int32

Vlastnost:

ConnectionFactory

Číslo portu, na kterém zprostředkovatel naslouchá příchozím požadavkům. Na zprostředkovateli je třeba konfigurovat uzel zpracování zpráv typu Real-timeInput nebo Real-timeOptimizedMessage Flow k naslouchání na tomto portu.

Tato vlastnost se používá s vlastností [XMSCS_RTT_HOST_NAME](#) k identifikaci zprostředkovatele.

Výchozí hodnota vlastnosti je [XMSCS_RTT_DEFAULT_PORT](#) nebo 1506.

XMSC_TIME_TO_LIVE**Datový typ:**

System.Int32

Vlastnost:

Místo určení

Název použitý v identifikátoru URI:

vypršení platnosti (pro místo určení WebSphere MQ)

timeToLive (pro výchozí místo určení poskytovatele systému zpráv produktu WebSphere)

Doba životnosti zpráv odeslaných do cíle.

Platné hodnoty vlastnosti jsou následující:

Platná hodnota	Význam
0	Zpráva odeslaná do místa určení nikdy nevyprší platnost.
Kladné celé číslo	Zpráva odeslaná do místa určení má zadaný čas k životu v milisekundách. Výchozí doba života producenta zpráv nebo jakákoli doba, po kterou má být zadán požadavek na odeslání, je ignorována.
XMSC_TIME_TO_LIVE_AS_APP	Zpráva odeslaná do místa určení má čas, který má být zadán při volání pro odeslání. Pokud při odesílání není určen žádný čas k životu, použije se výchozí doba života producenta zpráv.

Výchozí hodnota je XMSCS_TIME_TO_LIVE_AS_APP.

XMSC_ID_UŽIVATELE

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Identifikátor uživatele, který lze použít k ověření aplikace při pokusu o připojení k serveru systému zpráv. Identifikátor uživatele se používá s vlastností XMSC_PASSWORD.

Ve výchozím nastavení není vlastnost nastavena.

Multi Připojujete-li se k produktu IBM MQ for Multiplatformsa nastavíte vlastnost XMSCS_USERID továrny připojení, musí se shodovat s hodnotou **userid** přihlášeného uživatele. Pokud tyto vlastnosti nenastavíte, správce front použije při výchozím nastavení **userid** přihlášeného uživatele. Požadujete-li další ověření na úrovni připojení jednotlivých uživatelů, můžete napsat uživatelskou proceduru pro ověření klienta, která je konfigurována v produktu IBM MQ.

z/OS Chcete-li ověřit uživatele při připojování k produktu IBM MQ for z/OS, je třeba použít uživatelskou proceduru pro zabezpečení zprávy.

XMSC_

Datový typ:

Řetězec

Vlastnost:

Data ConnectionMeta

Identifikátor verze klienta XMS. Tato vlastnost je jen pro čtení.

XMSC_WMQ_BROKER_CONTROLQ

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Název řídicí fronty použité zprostředkovatelem.

Výchozí hodnota vlastnosti je SYSTEM.BROKER.CONTROL.QUEUE.

Tato vlastnost je relevantní pouze v doméně Publikování/odběr.

XMSC_WMQ_BROKER_PUBQ

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Název fronty monitorované zprostředkovatelem, kde aplikace odesílají zprávy, které publikují.

Výchozí hodnota vlastnosti je SYSTEM.BROKER.DEFAULT.STREAM.

Tato vlastnost je relevantní pouze v doméně Publikování/odběr .

XMSC_WMQ_BROKER_QMGR

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Název správce front, ke kterému je zprostředkovatel připojen.

Ve výchozím nastavení není vlastnost nastavena.

Tato vlastnost je relevantní pouze v doméně Publikování/odběr .

XMSC_WMQ_BROKER_SUBQ

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Název fronty odběratele pro spotřebitele netrvalých zpráv.

Název fronty odběratele musí začínat následujícími znaky:

SYSTEM.JMS.ND.

Chcete-li, aby všichni odběratelé netrvalých zpráv sdíleli frontu odběratele, zadejte úplný název sdílené fronty. Fronta s uvedeným názvem musí existovat předtím, než aplikace může vytvořit netrvalý spotřebitel zpráv.

Chcete-li, aby každý spotřebitel zpráv netrvalých zpráv načítal zprávy ze své vlastní výlučné fronty odběratele, zadejte název fronty, který končí hvězdičkou (*). Poté, když aplikace vytvoří netrvalý spotřebitel zpráv, klient XMS vytvoří dynamickou frontu pro výhradní použití odběratelem zprávy. Klient XMS používá hodnotu vlastnosti k nastavení obsahu pole **DynamicQName** v deskriptoru objektu, který se používá k vytvoření dynamické fronty.

Výchozí hodnota vlastnosti je SYSTEM.JMS.ND.SUBSCRIBER.QUEUE, což znamená, že produkt XMS při výchozím nastavení používá přístup prostřednictvím sdílené fronty.

Tato vlastnost je relevantní pouze v doméně Publikování/odběr .

XMSC_WMQ_BROKER_VERSION

Datový typ:

System.Int32

Vlastnost:

ConnectionFactory a Destination

Název použitý v identifikátoru URI:

brokerVersion

Typ zprostředkovatele použitý aplikací pro připojení nebo pro cíl. Pouze místo určení, které je téma, může mít tuto vlastnost.

Platné hodnoty vlastnosti jsou následující:

Platná hodnota	Význam
XMSC_WMQ_BROKER_V1	Aplikace používá zprostředkovatele WebSphere MQ Publikování/ odběr . Aplikace může tuto hodnotu také použít, pokud migrujete z produktu WebSphere MQ Publikování/odběr do produktu WebSphere Message Broker , ale neměnila aplikaci.
XMSC_WMQ_BROKER_V2	Aplikace používá zprostředkovatele produktu IBM Integration Bus.
XMSC_WMQ_BROKER_UNSPECIFIED	Po migraci zprostředkovatele nastavte tuto vlastnost tak, aby se záhlaví RFH2 již nadále nepoužívala. Po migraci tato vlastnost již není relevantní.

Výchozí hodnota pro továrnu připojení je XMSCS_WMQ_BROKER_UNSPECIFIED, ale při výchozím nastavení není vlastnost nastavena pro místo určení. Nastavení této vlastnosti pro cíl přepíše jakoukoli hodnotu zadanou vlastností továrny připojení.

XMSC_WMQ_CCDTURL

Datový typ:

System.String

Vlastnost:

ConnectionFactory

Použitelné objekty:

Dlouhý název nástroje pro administraci JMS: CCDTURL

Krátký název nástroje pro administraci JMS: CCDT

Adresa URL (Uniform Resource Locator) identifikující název a umístění souboru, který obsahuje tabulku definic kanálů klienta, a také určuje, jakým způsobem lze k souboru přistupovat.

Při výchozím nastavení tato vlastnost není nastavena.

XMSC_WMQ_CC SID

Datový typ:

System.Int32

Vlastnost:

Místo určení

Název použitý v identifikátoru URI:

CCSID

Identifikátor (CCSID) kódované znakové sady nebo kódové stránky, že řetězce znakových dat v těle zprávy jsou v době, kdy klient XMS předá zprávu do cíle. Je-li nastaveno pro individuální zprávu, vlastnost JMS_IBM_CHARACTER_SET přepíše CCSID uvedený pro místo určení touto vlastností.

Výchozí hodnota vlastnosti je 1208.

Tato vlastnost je relevantní pouze pro zprávy odeslané do místa určení, nikoli na zprávy přijaté z cíle.

XMSC_WMQ_CHANNEL

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Použitelné objekty:

Dlouhý název nástroje pro administraci JMS: CHANNEL

Krátký název nástroje pro administraci JMS: CHAN

Název kanálu, který se má použít pro připojení.

Ve výchozím nastavení není vlastnost nastavena.

Tato vlastnost je relevantní pouze v případě, že se aplikace připojuje ke správci front v režimu klienta.

XMSC_WMQ_CLIENT_RECONNECT_OPTIONS**Datový typ:**

Řetězec

Vlastnost:

ConnectionFactory

Použitelné objekty:

Dlouhý název nástroje pro administraci JMS: CLIENTRECONECOPTIONS

Krátký název nástroje pro administraci JMS: CROPT

Tato vlastnost uvádí volby nového připojení klienta pro nová připojení vytvořená touto továrnou. Je nalezen v XMSC a je jedním z:

- WMQ_CLIENT_RECONNECT_AS_DEF (výchozí). Použijte hodnotu zadanou v souboru `mqclient.ini`. Nastavte tuto hodnotu pomocí vlastnosti **DefRecon** v sekci Kanály. Může být nastavena na jednu z následujících možností:
 1. Ano. Behaves jako volba WMQ_CLIENT_RECONNECT
 2. -NE. Předvolba. Neuvádí žádné volby opětovného připojení
 3. QMGR. Behaves jako volba WMQ_CLIENT_RECONNECT_Q_MGR
 4. Zakázáno. Behaves jako volba WMQ_CLIENT_RECONNECT_DISABLED
- WMQ_CLIENT_RECONNECT. Připojte se k libovolnému správci front uvedenému v seznamu názvů připojení.
- WMQ_CLIENT_RECONNECT_Q_MGR. Znovu naváže spojení se stejným správcem front, ke kterému je původně připojen. Vrací MQRC_RECONNECT_QMID_MISMATCH, pokud se správce front, ke kterému se pokouší připojit (uvedený v seznamu názvů připojení), liší od QMID ke správci front, k němuž se původně připojil.
- WMQ_CLIENT_RECONNECT_DISABLED. Opětovné připojení je zakázáno.

XMSC_WMQ_CLIENT_RECONNECT_TIMEOUT**Datový typ:**

Řetězec

Vlastnost:

ConnectionFactory

Použitelné objekty:

Dlouhý název nástroje pro administraci JMS: CLIENTRECONNECTSIMEOUT

Krátký název nástroje pro administraci JMS: CRT

Vlastnost XMSCS_WMQ_CLIENT_RECONT_TIMEOUT je platná pouze pro spravovaného klienta XMS .NET .

Tato vlastnost určuje dobu trvání (v sekundách), po kterou se připojení klienta pokouší znovu navázat spojení.

Po pokusu o nové připojení po tuto dobu dojde k selhání klienta pomocí funkce MQRC_RECONIPRI_FAILED. Výchozí nastavení této vlastnosti je XMSC.WMQ_CLIENT_RECONNECT_TIMEOUT_DEFAULT.

Výchozí hodnota této vlastnosti je 1800.

XMSC_WMQ_CONNECTION_MODE

Datový typ:

System.Int32

Vlastnost:

ConnectionFactory

Režim, pomocí kterého se aplikace připojuje ke správci front.

Platné hodnoty vlastnosti jsou následující:

Platná hodnota	Význam
XMSC_WMQ_CM_VAZEB	Pro optimální výkon se používá připojení ke správci front v režimu vazeb. Tato hodnota je výchozí hodnotou pro C/C + .
XMSC_WMQ_CM_KLIENT	Připojení ke správci front v režimu klienta za účelem zajištění plně spravovaného zásobníku. Tato hodnota je výchozí hodnotou pro .NET.
XMSCS_WMQ_CM_CLIENT_UNMANAGED (pouze pro .NET)	Připojení ke správci front, který vynutí vytvoření nespravovaného zásobníku klienta.

Související pojmy

Spravované a nespravované operace v produktu .NET

Spravovaný kód se provádí výhradně v rámci běhového prostředí obecného jazyka .NET a je zcela závislý na službách poskytovaných tímto běhovým prostředím. Aplikace je klasifikována jako nespravovaná, pokud libovolná část aplikace běží nebo volá služby mimo prostředí JRE (common language runtime) produktu .NET .

XMSC_WMQ_CONNECTION_NAME_LIST

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Použitelné objekty:

Dlouhý název nástroje pro administraci JMS: CONNECTIONNAMELIST

Krátký název nástroje pro administraci JMS: CNLIST

Tato vlastnost určuje hostitele, ke kterému se klient pokusí znovu připojit po přerušení připojení.

Seznam názvů připojení je čárkami oddělený seznam dvojic hostitel/ip port. Výchozí nastavení této vlastnosti je WMQ_CONNECTION_NAME_LIST_DEFAULT.

Příklad:127.0.0.1 (1414) ,host2.example.com(1400)

Výchozí nastavení této vlastnosti je localhost (1414).

XMSC_WMQ_DUR_SUBQ

Datový typ:

Řetězec

Vlastnost:

Místo určení

Název fronty odběratele pro trvalého odběratele, který přijímá zprávy z cíle. Pouze místo určení, které je téma, může mít tuto vlastnost.

Název fronty odběratele musí začínat následujícími znaky:

SYSTEM.JMS.D.

Chcete-li, aby všechny trvalé odběratele sdílely frontu odběratele, zadejte úplný název sdílené fronty. Než může aplikace vytvořit trvalého odběratele, musí existovat fronta s určeným názvem.

Chcete-li, aby každý trvalý odběratel načítal zprávy ze své vlastní výlučné fronty odběratele, zadejte název fronty, který končí hvězdičkou (*). Poté, když aplikace vytvoří trvalého odběratele, vytvoří klient produktu XMS dynamickou frontu pro výhradní použití trvalým odběratelem. Klient XMS používá hodnotu vlastnosti k nastavení obsahu pole **DynamicQName** v deskriptoru objektu, který se používá k vytvoření dynamické fronty.

Výchozí hodnota vlastnosti je SYSTEM.JMS.D.SUBSCRIBER.QUEUE, což znamená, že produkt XMS při výchozím nastavení používá přístup prostřednictvím sdílené fronty.

Tato vlastnost je relevantní pouze v doméně Publikování/odběr .

XMSC_WMQ_ENCODING

Datový typ:

System.Int32

Vlastnost:

Místo určení

Jak číselná data v těle zprávy jsou představována, když klient produktu XMS předává zprávu do cíle. Je-li pro jednotlivou zprávu nastavena vlastnost JMS_IBM_ENCODING , přepisuje hodnotu kódování určené pro cíl touto vlastností. Vlastnost uvádí znázornění binárních celých čísel, pakovaných dekadických celých čísel a čísel s pohyblivou řádovou čárkou.

Platné hodnoty vlastnosti jsou stejné jako hodnoty, které mohou být zadány v poli **Encoding** deskriptoru zpráv.

Aplikace může pro nastavení vlastnosti použít následující pojmenované konstanty:

pojmenovaná konstanta	Význam
MQENC_INTEGER_NORMAL	Normální celočíselné kódování
MQENC_INTEGER_REVERSED	Obrácené celočíselné kódování
MQENC_DECIMAL_NORMAL	Normální pakované kódování desetinných čísel
MQENC_DECIMAL_REVERSED	Obrácené pakované kódování desetinných čísel
MQENC_FLOAT_IEEEE_NORMAL	Normální kódování čísel s plovoucí desetinnou čárkou
MQENC_FLOAT_IEEE_OBRÁCENÝ	Obrácené kódování čísel s pohyblivou řádovou čárkou IEEE
MQENC_FLOAT_S390	Kódování čísel s pohyblivou řádovou čárkou z/OS
MQENC_NATIVE	Kódování nativního počítače

Chcete-li vytvořit hodnotu vlastnosti, může aplikace přidat tři z těchto konstant takto:

- Konstanta, jejíž název začíná řetězcem MQENC_INTEGER, určuje reprezentaci binárních celých čísel.
- Konstanta, jejíž název začíná řetězcem MQENC_DECIMAL, určuje reprezentaci pakovaných dekadických celých čísel.
- Konstanta, jejíž název začíná řetězcem MQENC_FLOAT, určuje reprezentaci čísel s pohyblivou řádovou čárkou.

Alternativně může aplikace nastavit vlastnost na hodnotu MQENC_NATIVE, jejíž hodnota závisí na prostředí.

Výchozí hodnota vlastnosti je MQENC_NATIVE.

Tato vlastnost je relevantní pouze pro zprávy odeslané do místa určení, nikoli na zprávy přijaté z cíle.

XMSC_WMQ_FAIL_IF_QUIESCE

Datový typ:

System.Int32

Vlastnost:

ConnectionFactory a Destination

Název použitý v identifikátoru URI:

FAILIFQUIESCE

Použitelné objekty:

Dlouhý název nástroje pro administraci JMS: FAILIFQUIESCE

Krátký název nástroje pro administraci JMS: FIQ

Zda se volání konkrétních metod nezdaří, pokud je správce front, k němuž je aplikace připojen, ve stavu uvedení do klidového stavu.

Platné hodnoty vlastnosti jsou následující:

Platná hodnota	Význam
XMSC_WMQ_FIQ_YES	Volání určitých metod selže, pokud se správce front nachází ve stavu uvedení do klidového stavu. Když aplikace zjistí, že správce front je uváděn do klidového stavu, může aplikace dokončit její okamžitou úlohu a zavřít připojení, což umožňuje zastavení správce front.
XMSC_WMQ_FIQ_NO	Žádná volání metody se nezdařila, protože správce front je ve stavu uvedení do klidového stavu. Zadáte-li tuto hodnotu, nebude moci aplikace zjistit, zda je správce front uváděn do klidového stavu. Aplikace může pokračovat v provádění operací se správcem front, a zabránit tak zastavení správce front.

Výchozí hodnota továrny připojení je XMSC_WMQ_FIQ_YES, ale při výchozím nastavení není vlastnost nastavena pro místo určení. Nastavení této vlastnosti pro cíl přepíše jakoukoli hodnotu zadanou vlastností továrny připojení.

XMSC_WMQ_MESSAGE_BODY

Datový typ:

System.Int32

Vlastnost:

Místo určení

Tato vlastnost určuje, zda aplikace XMS zpracovává strukturu MQRFH2 zprávy produktu IBM WebSphere MQ jako součást informačního obsahu zprávy (tj. v rámci těla zprávy).

Poznámka: Při odesílání zpráv do místa určení nahradí vlastnost XMSC_WMQ_MESSAGE_BODY existující vlastnost XMS Destination property XMSC_TARGET_CLIENT_CLIENT.

Platné hodnoty pro tuto vlastnost jsou:

XMSC_WMQ_MESSAGE_BODY_JMS

Přijmout: Příchozí typ zprávy a tělo zprávy XMS se určuje podle obsahu MQRFH2 (pokud existuje) nebo MQMD (pokud v přijaté zprávě IBM WebSphere MQ není MQRFH2).

Odeslat: Tělo odchozí zprávy XMS obsahuje předřazené a automaticky generované záhlaví MQRFH2 založené na vlastnostech zprávy XMS a v polích záhlaví.

XMSC_WMQ_MESSAGE_BODY_MQ

Přijmout: Příchozí typ zprávy XMS je vždy ByteMessage, bez ohledu na obsah přijaté zprávy IBM WebSphere MQ nebo pole formátu pro přijatý MQMD. Tělo zprávy XMS je nezměněna data zprávy vrácená voláním rozhraním API poskytovatele systému zpráv. Znaková sada a kódování dat v těle zprávy je určeno poli CodedCharSetId a zakódováním v rámci polí MQMD. Formát dat v těle zprávy je určen polem Formát deskriptoru MQMD.

Odeslat: Tělo odchozí zprávy XMS obsahuje informační obsah aplikace tak, jak je, a do těla se nepřidá automaticky generované záhlaví IBM WebSphere MQ .

XMSC_WMQ_MESSAGE_BODY_UNSPECIFIED

Přijmout: Klient XMS určuje vhodnou hodnotu pro tuto vlastnost. V případě cesty příjmu je tato hodnota hodnotou vlastnosti WMQ_MESSAGE_BODY_JMS.

Odeslat: Klient XMS určuje vhodnou hodnotu pro tuto vlastnost. Na cestě k odeslání je tato hodnota hodnotou vlastnosti XMSCS_WMQ_TARGET_CLIENT.

Při výchozím nastavení je tato vlastnost nastavena na hodnotu XMSCS_WMQ_MESSAGE_BODY_UNSPECIFIED.

XMSC_WMQ_MQMD_MESSAGE_CONTEXT

Datový typ:

System.Int32

Vlastnost:

Místo určení

určuje, jaká úroveň kontextu zprávy má být nastavena aplikací XMS . Aby tato vlastnost mohla nabýt účinnosti, musí aplikace běžet s příslušným oprávněním kontextu.

Platné hodnoty pro tuto vlastnost jsou:

XMSC_WMQ_MDCEX_DEFAULT

Pro odchozí zprávy volání MQOPEN rozhraní API a struktura MQPMO neurčují žádné explicitní volby kontextu zprávy.

XMSC_WMQ_MDCTX_SET_IDENTITY_CONTEXT

Volání rozhraní MQOPEN API určuje volbu kontextu zprávy MQOO_SET_IDENTITY_CONTEXT a struktura MQPMO určuje hodnotu MQPMO_SET_IDENTITY_CONTEXT.

XMSC_WMQ_MDCTX_SET_ALL_CONTEXT

Volání rozhraní MQOPEN API uvádí volbu kontextu zprávy MQOO_SET_ALL_CONTEXT a struktura MQPMO určuje MQPMO_SET_ALL_CONTEXT.

Při výchozím nastavení je tato vlastnost nastavena na hodnotu XMSCS_WMQ_MDCTX_DEFAULT.

Poznámka: Tato vlastnost není relevantní, když se aplikace připojuje k systému Integration Bus.

Následující vlastnosti vyžadují vlastnost XMSCS_WMQ_MQMD_MESSAGE_CONTEXT, která má být nastavena na hodnotu vlastnosti XMSCS_WMQ_MDCTX_SET_IDENTITY_CONTEXT nebo vlastnost XMSCS_WMQ_MDCTX_SET_ALL_CONTEXT při odesílání zprávy za účelem dosažení požadovaného účinku:

- JMS_IBM_MQMD_USERIDENTIFIER
- ROZLIŠUJÍCÍ TOKEN JMS_IBM_MQMD_ACCOUNTINGTOKEN
- JMS_IBM_MQMD_APPLIDENTITYDATA

Následující vlastnosti vyžadují, aby vlastnost XMSCS_WMQ_MQMD_MESSAGE_CONTEXT byla nastavena na hodnotu vlastnosti XMSCS_WMQ_MDCTX_SET_ALL_CONTEXT při odesílání zprávy za účelem dosažení požadovaného efektu:

- JMS_IBM_MQMD_PUTAPPLTYPE
- JMS_IBM_MQMD_PUTAPPLNAME
- JMS_IBM_MQMD_PUTDATE
- ČAS PUTTIME JMS_IBM_MQMD_PUTTIME
- JMS_IBM_MQMD_APPLORIGINDATA

XMSC_WMQ_MQMD_READ_ENABLED

Datový typ:

System.Int32

Vlastnost:

Místo určení

Tato vlastnost určuje, zda aplikace XMS může extrahovat hodnoty z polí MQMD, či nikoli.

Platné hodnoty pro tuto vlastnost jsou:

XMSC_WMQ_READ_ENABLED_NO

Při odesílání zpráv nejsou vlastnosti JMS_IBM_MQMD* v odeslané zprávě aktualizovány tak, aby odražely aktualizované hodnoty polí v produktu MQMD.

Při příjmu zpráv nejsou žádné z vlastností JMS_IBM_MQMD* k dispozici v přijaté zprávě, i když některé nebo všechny z nich jsou nastaveny odesilatelem.

XMSC_WMQ_READ_ENABLED_YES

Při odesílání zpráv jsou všechny vlastnosti JMS_IBM_MQMD* v odeslané zprávě aktualizovány tak, aby odražely aktualizované hodnoty polí v MQMD, včetně vlastností, které odesílatel explicitně nenastavil.

Při příjmu zpráv jsou všechny vlastnosti JMS_IBM_MQMD* k dispozici v přijaté zprávě, včetně vlastností, které odesílatel explicitně nenastavil.

Při výchozím nastavení je tato vlastnost nastavena na hodnotu XMSCS_WMQ_READ_ENABLED_NO.

XMSC_WMQ_MQMD_WRITE_ENABLED

Datový typ:

System.Int32

Vlastnost:

Místo určení

Tato vlastnost určuje, zda může aplikace XMS určovat hodnoty polí MQMD či nikoli.

Platné hodnoty pro tuto vlastnost jsou:

XMSC_WMQ_WRITE_ENABLED_NO

Všechny vlastnosti JMS_IBM_MQMD* jsou ignorovány a jejich hodnoty se nekopírují do základní struktury MQMD.

XMSC_WMQ_WRITE_ENABLED_YES

Vlastnosti JMS_IBM_MQMD* jsou zpracovány. Jejich hodnoty jsou zkopírovány do podkladové struktury MQMD.

Při výchozím nastavení je tato vlastnost nastavena na hodnotu XMSCS_WMQ_WRITE_ENABLED_NO.

XMSC_WMQ_PUT_ASYNC_ALLOWED

Datový typ:

System.Int32

Vlastnost:

Místo určení

Tato vlastnost určuje, zda producenti zpráv mohou používat k odesílání zpráv do tohoto místa určení asynchronní operace put.

Platné hodnoty pro tuto vlastnost jsou:

XMSC_WMQ_PUT_ASYNC_ALLOWED_AS_DEST

Určete, zda jsou asynchronní operace vložení povoleny odkazem na definici fronty nebo tématu.

XMSC_WMQ_PUT_ASYNC_ALLOWED_AS_Q_DEF

Určete, zda jsou asynchronní operace vložení povoleny odkazem na definici fronty.

XMSC_WMQ_PUT_ASYNC_ALLOWED_AS_TOPIC_DEF

Určete, zda jsou asynchronní operace vložení povoleny odkazem na definici tématu.

XMSC_WMQ_PUT_ASYNC_ALLOWED_DISABLED

Asynchronní operace put nejsou povoleny.

XMSC_WMQ_PUT_ASYNC_ALLOWED_ENABLED

Asynchronní operace put jsou povoleny.

Při výchozím nastavení je tato vlastnost nastavena na hodnotu XMSCS_WMQ_PUT_ASYNC_ALLOWED_AS_DEST.

Poznámka: Tato vlastnost není relevantní, když se aplikace připojuje k systému Integration Bus.

XMSC_WMQ_READ_AHEAD_ALLOWED**Datový typ:**

System.Int32

Vlastnost:

Místo určení

Tato vlastnost určuje, zda mohou příjemci zpráv a zprostředkovatelé front používat dopředné čtení k načtení netrvalých netraskčních zpráv z tohoto cíle do interní vyrovnávací paměti před jejich přijetím.

Platné hodnoty pro tuto vlastnost jsou:

XMSC_WMQ_READ_AHEAD_ALLOWED_AS_Q_DEF

Určete, zda je dopředné čtení povoleno s odkazem na definici fronty.

XMSC_WMQ_READ_AHEAD_ALLOWED_AS_TOPIC_DEF

Určete, zda je dopředné čtení povoleno s odkazem na definici tématu.

XMSC_WMQ_READ_AHEAD_ALLOWED_AS_DEST

Určete, zda je dopředné čtení povoleno s odkazem na definici fronty nebo tématu.

XMSC_WMQ_READ_AHEAD_ALLOWED_DISABLED

Čtení napřed není povoleno při přijímání nebo procházení zpráv.

XMSC_WMQ_READ_AHEAD_ALLOWED_ENABLED

Čtení napřed je povoleno.

Při výchozím nastavení je tato vlastnost nastavena na hodnotu XMSCS_WMQ_READ_AHEAD_ALLOWED_AS_DEST.

XMSC_WMQ_READ_AHEAD_CLOSE_POLICY**Datový typ:**

System.Int32

Vlastnost:

Místo určení

Tato vlastnost určuje, zda budou zprávy doručovány do asynchronního modulu listener zpráv, co se stane se zprávami v interní vyrovnávací paměti pro čtení při zavření spotřebitele zpráv.

Tato vlastnost je použitelná při zadávání uzavírajících voleb fronty při přijímání zpráv z místa určení a nepoužitelná při odesílání zpráv do místa určení.

Tato vlastnost je ignorována pro prohlížeče front, protože během procházení jsou zprávy stále k dispozici ve frontách.

Platné hodnoty pro tuto vlastnost jsou:

XMSC_WMQ_READ_AHEAD_CLOSE_POLICY_DELIVER_CURRENT

Před návratem bude dokončeno pouze aktuální vyvolání modulu listener pro zprávy, případně zanechání zpráv v interní vyrovnávací paměti dopředného čtení, které se pak vyřadí.

XMSC_WMQ_READ_AHEAD_CLOSE_POLICY_DELIVER_ALL

Všechny zprávy v interní vyrovnávací paměti dopředného čtení jsou před návratem doručeny do modulu listener pro zprávy aplikace.

Při výchozím nastavení je tato vlastnost nastavena na hodnotu XMSCS_WMQ_READ_AHEAD_CLOSE_POLICY_DELIVER_CURRENT.

Poznámka:

- **Abnormální ukončení aplikace**

Při náhlém ukončení aplikace XMS budou všechny zprávy v vyrovnávací paměti dopředného čtení ztraceny.

- **Důsledky pro transakce**

Čtení napřed je vypnuto, když aplikace používají transakci. Aplikace tak nevidí žádný rozdíl v chování, když používají relace zahrnující transakci.

- **Implikace režimů potvrzení relace**

Čtení napřed je povoleno na netransakčních relacích, když jsou režimy potvrzení buď XMSCS_AUTO_ACKNOWLEDGE nebo XMSCS_DUPS_OK_ACKNOWLEDGE. Čtení napřed je vypnuto, pokud je režim potvrzení relace XMSCS_CLIENT_ACKNOWLEDGE, bez ohledu na relace s transakcemi nebo bez transakcí.

- **Implikace pro selektory fronty a selektory prohlížeče front**

Výběry fronty a selektory prohlížeče front používané v aplikacích XMS získají výhodu výkonu při čtení napřed. Zavření Prohlížeče front nesníží výkon, protože zpráva je stále k dispozici ve frontě pro další operace. Nejsou žádné jiné důsledky pro prohlížeče front a selektory prohlížeče front, kromě výhod výkonu čtení napřed.

XMSC_WMQ_NÁZEV_HOSTITELE

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Použitelné objekty:

Dlouhý název nástroje pro administraci JMS: HOSTNAME

Krátký název nástroje pro administraci JMS: HOST

Název nebo adresa IP hostitele systému, na kterém je spuštěn správce front.

Tato vlastnost se používá pouze v případě, že se aplikace připojuje ke správci front v režimu klienta.

Vlastnost se používá spolu s vlastností [XMSCS_WMQ_PORT](#) k identifikaci správce front.

Výchozí hodnota vlastnosti je localhost.

XMSC_WMQ_LOCAL_ADDRESS

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Použitelné objekty:

Dlouhý název nástroje pro administraci JMS: LOCALADDRESS

Krátký název nástroje pro administraci JMS: LA

Pro připojení ke správci front tato vlastnost určuje rozhraní lokální sítě, lokálního portu nebo rozsahu lokálních portů, nebo obojí.

Hodnota vlastnosti je řetězec s následujícím formátem:

[*název_hostitele*] [(*low_port*) [,*high_port*]]

Význam proměnných je následující:

název_hostitele

Název hostitele nebo adresa IP lokálního síťového rozhraní, které má být použito pro připojení.

Poskytnutí těchto informací je nezbytné pouze v případě, že systém, na kterém je aplikace spuštěna, má dvě nebo více síťových rozhraní a vy potřebujete mít možnost určit, které rozhraní musí být pro připojení použito. Má-li systém pouze jedno síťové rozhraní, lze použít pouze toto rozhraní. Má-li systém dvě nebo více síťových rozhraní a nespecifikujete, které rozhraní musí být použito, je rozhraní vybráno náhodně.

nízká_port

Číslo lokálního portu, které má být použito pro připojení.

Je-li zadána také hodnota *high_port*, hodnota *low_port* je interpretována jako nejnižší číslo portu v rozsahu čísel portů.

vysoká_port

Nejvyšší číslo portu v rozsahu čísel portů. Jeden z portů v uvedeném rozsahu musí být použit pro připojení.

Maximální délka řetězce je 48 znaků.

Zde je několik příkladů platných hodnot vlastnosti:

PLANET NAME
9.20.4.98
JUPER (1000)
9.20.4.98(1000,2000)
(1000)
(1000,2000)

Ve výchozím nastavení není vlastnost nastavena.

Tato vlastnost je relevantní pouze v případě, že se aplikace připojuje ke správci front v režimu klienta.

XMSC_WMQ_MESSAGE_SELECTION

Datový typ:

System.Int32

Vlastnost:

ConnectionFactory

Určuje, zda má být výběr zpráv proveden klientem XMS nebo zprostředkovatelem.

Platné hodnoty vlastnosti jsou následující:

Platná hodnota	Význam
XMSC_WMQ_MSEL_CLIENT	Výběr zpráv provádí klient XMS .
XMSC_WMQ_MSEL_BROKER	Výběr zpráv provádí zprostředkovatel.

Výchozí hodnota je XMSCS_WMQ_MSEL_CLIENT.

Tato vlastnost je relevantní pouze v doméně Publikování/odběr . Výběr zpráv zprostředkovatelem není podporován, je-li vlastnost XMSC_WMQ_BROKER_VERSION nastavena na hodnotu XMSC_WMQ_BROKER_V1.

XMSC_WMQ_MSG_BATCH_SIZE

Datový typ:

System.Int32

Vlastnost:

ConnectionFactory

Maximální počet zpráv, které mají být načteny z fronty v jedné dávce, v případě asynchronního doručování zpráv.

Když aplikace používá asynchronní doručování zpráv za určitých podmínek, klient XMS načte dávku zpráv z fronty před odesláním každé zprávy jednotlivě do aplikace. Tato vlastnost určuje maximální počet zpráv, které mohou být v dávce.

Hodnota vlastnosti je kladné celé číslo a výchozí hodnota je 10. Zvažte nastavení této vlastnosti na jinou hodnotu pouze v případě, že máte specifický problém s výkonem, který je třeba adresovat.

Je-li aplikace připojena ke správci front v síti, může zvýšení hodnoty této vlastnosti snížit režii sítě a doby odezvy, ale zvýšit množství paměti potřebné k ukládání zpráv v systému klienta. Naopak snížení hodnoty této vlastnosti může zvýšit režii sítě a doby odezvy, ale sníží množství paměti potřebné k uložení zpráv.

XMSC_WMQ_POLLING_INTERVAL

Datový typ:

System.Int32

Vlastnost:

ConnectionFactory

Pokud každý modul listener pro zprávy v rámci relace nemá ve frontě žádnou odpovídající zprávu, jedná se o maximální interval v milisekundách, který uplyne předtím, než se každý modul listener pro zprávy znovu pokusí získat zprávu z příslušné fronty.

Pokud se často stává, že pro žádný z listenerů zpráv v rámci relace není k dispozici žádná vhodná zpráva, zvažte zvýšení hodnoty této vlastnosti.

Hodnota vlastnosti je kladné celé číslo. Výchozí hodnota je 5000.

XMSC_WMQ_PORT

Datový typ:

System.Int32

Vlastnost:

ConnectionFactory

Použitelné objekty:

Dlouhý název nástroje pro administraci JMS: PORT

Krátký název nástroje pro administraci JMS: PORT

Číslo portu, na kterém správce front naslouchá příchozím požadavkům.

Tato vlastnost se používá pouze v případě, že se aplikace připojuje ke správci front v režimu klienta. Vlastnost se používá s vlastností `XMSCS_WMQ_HOST_NAME` k identifikaci správce front.

Výchozí hodnota vlastnosti je `XMSCS_WMQ_DEFAULT_CLIENT_PORT` nebo 1414.

XMSC_WMQ_PROVIDER_VERSION

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Verze, vydání, úroveň modifikace a opravný balík správce front, ke kterému se tato aplikace hodlá připojovat. Platné hodnoty pro tuto vlastnost jsou:

- Nespecifikováno

Nebo řetězec v jednom z následujících formátů

- V.R.M.F
- V.R.M
- V.R
- V

kde V, R, M a F jsou celá čísla větší nebo rovná nule.

Hodnota 7 nebo vyšší označuje, že tato verze je určena jako IBM WebSphere MQ 7.0 ConnectionFactory pro připojení ke správci front IBM WebSphere MQ 7.0. Hodnota, která je starší než 7 (například "6.0.2.0"), označuje, že je určen pro použití se správcem front dříve než verze 7.0. Výchozí hodnota, nespecifikovaná, povoluje připojení k jakékoli úrovni správce front, určení použitelných vlastností a funkcí dostupných na základě schopností správce front.

Při výchozím nastavení je tato vlastnost nastavena na hodnotu "unspecified".

Poznámka:

- Pokud je hodnota XMSCS_WMQ_PROVIDER_VERSION nastavena na hodnotu 6, nedojde k žádné sdílení soketu. 2. PRO
- Selhání připojení, je-li parametr XMSCS_WMQ_PROVIDER_VERSION nastaven na hodnotu 7 a na serveru SHARECNV pro kanál je nastavena hodnota 0.
- Specifické funkce produktu IBM WebSphere MQ 7.0 jsou zakázány, je-li hodnota XMSCS_WMQ_PROVIDER_VERSION nastavena na hodnotu UNSPECIFIED a hodnota SHARECNV je nastavena na hodnotu 0.

Verze klienta produktu IBM WebSphere MQ také hraje hlavní roli v tom, zda klientská aplikace XMS může používat specifické funkce produktu IBM WebSphere MQ 7.0. V následující tabulce je popsáno chování.

Poznámka: Vlastnost systému XMSCS_WMQ_OVERRIDEPROVIDERVERSION potlačuje vlastnost XMSCS_WMQ_PROVIDER_VERSION. Tuto vlastnost lze použít v případě, že nelze změnit nastavení faktorie připojení.

Tabulka 33. Klient XMS -Schopnost použít specifické funkce produktu IBM WebSphere MQ 7.0 .

#	XMSC_WMQ_PROVIDER_VERSION	IBM WebSphere MQ Verze klienta	IBM WebSphere MQ 7.0 funkce
1	nespecifikováno	7	ZAP
2	nespecifikováno	6	VYP
3	7	7	ZAP
4	7	6	Výjimka
5	6	6	VYP
6	6	7	VYP

XMSC_WMQ_PUB_ACK_INTERVAL

Datový typ:

System.Int32

Vlastnost:

ConnectionFactory

Počet zpráv publikovaných vydavatelem před tím, než klient produktu XMS vyžádá potvrzení od zprostředkovatele.

Snížíte-li hodnotu této vlastnosti, bude klient vyžadovat potvrzení příjmu častěji, a proto se výkon vydavatele sníží. Zvýšíte-li tuto hodnotu, bude klientovi trvat déle, než ohlásí výjimku při selhání zprostředkovatele.

Hodnota vlastnosti je kladné celé číslo. Výchozí hodnota je 25.

XMSC_WMQ_QMGR_CCSID

Datový typ:

System.Int32

Vlastnost:

ConnectionFactory

Identifikátor (CCSID) kódové znakové sady nebo kódové stránky, v níž se mezi klientem XMS a klientem WebSphere MQ vyměňují pole znakových dat definovaná v rozhraní MQI (Message Queue Interface). Tato vlastnost se nevztahuje na řetězce znakových dat v těle zpráv.

Když se aplikace XMS připojí ke správci front v režimu klienta, klient XMS se spojí s klientem WebSphere MQ. Informace vyměňované mezi dvěma klienty obsahují pole znakových dat, která jsou definována v rámci MQI. Za normálních okolností klient produktu WebSphere MQ předpokládá, že tato pole jsou v kódové stránce systému, na kterém jsou klienti spuštěni. Pokud klient XMS poskytuje a očekává přijetí těchto polí v jiné kódové stránce, musíte nastavit tuto vlastnost, aby bylo možné informace o klientovi WebSphere MQ informovat.

Pokud klient produktu WebSphere MQ předává tato pole znakových dat správci front, musí být data v nich převedena, je-li to nutné, do kódové stránky použité správcem front. Podobně platí, že pokud klient produktu WebSphere MQ přijme tato pole ze správce front, data v nich musí být převedena, pokud je to nutné, do kódové stránky, ve které klient XMS očekává přijetí dat. Klient produktu WebSphere MQ používá tuto vlastnost k provedení těchto převodů dat.

Ve výchozím nastavení není vlastnost nastavena.

Nastavení této vlastnosti je ekvivalentní nastavení proměnné prostředí MQCCSID pro klienta produktu WebSphere MQ, který podporuje nativní klientské aplikace produktu WebSphere MQ. Další informace o této proměnné prostředí naleznete v tématu *Klienti produktu WebSphere MQ*.

XMSC_WMQ_QUEUE_MANAGER

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Použitelné objekty:

Dlouhý název nástroje pro administraci JMS: QMANAGER

Krátký název nástroje pro administraci JMS: QMGR

Název správce front, s nímž má být navázáno připojení.

Ve výchozím nastavení není vlastnost nastavena.

XMSC_WMQ_RECEIVE_CCSID

Vlastnost místa určení, která nastavuje cílové CCSID pro převod zpráv správce front. Hodnota je ignorována, pokud není hodnota XMSCS_WMQ_RECEIVE_CONVERSION nastavena na hodnotu WMQ_RECEIVE_CONVERSION_QMGR.

Datový typ:

Celé číslo

Hodnota:

Libovolné kladné celé číslo.

Výchozí hodnota je 1208.

Zadání hodnoty `GMO_CONVERT` ve zprávě je volitelné. Je-li zadána hodnota `GMO_CONVERT`, bude provedena konverze podle zadané hodnoty.

XMSC_WMQ_RECEIVE_CONVERSION

Cílová vlastnost, která určuje, zda bude převod dat prováděn správcem front.

Datový typ:

Celé číslo

Hodnoty:

`XMSCS_WMQ_RECEIVE_CONVERSION_CLIENT_MSG` (DEFAULT): Provést konverzi dat pouze na klientovi XMS. Převod se vždy provádí s použitím kódové stránky 1208.

`XMSCS_WMQ_RECEIVE_CONVERSION_QMGR`: Provedení převodu dat na správci front před odesláním zprávy klientovi XMS.

XMSC_WMQ_RECEIVE_EXIT

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Identifikuje uživatelskou proceduru pro přijetí zprávy kanálu, která má být spuštěna.

Hodnota vlastnosti je řetězec, který identifikuje uživatelskou proceduru pro přijetí zprávy kanálu a má následující formát:

libraryName(entryPointNázev)

kde:

- **libraryName** je úplná cesta ke spravovanému ukončení .dll
- **entryPointName** je název třídy kvalifikované oborem názvů

Například: `C:\MyReceiveExit.dll(MyReceiveExitNameSpace.MyReceiveExitClassName)`

Ve výchozím nastavení není vlastnost nastavena.

Tato vlastnost je relevantní pouze v případě, že se aplikace připojuje ke správci front ve spravovaném režimu klienta. Podporovány jsou také pouze spravované ukončení.

XMSC_WMQ_RECEIVE_EXIT_INIT

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Uživatelská data, která jsou předána uživatelské proceduře pro přijetí zprávy kanálu při volání.

Hodnota vlastnosti je řetězec. Ve výchozím nastavení není vlastnost nastavena.

Tato vlastnost je relevantní pouze v případě, že se aplikace připojuje ke správci front ve spravovaném klientském režimu a je nastavena vlastnost [“XMSC_WMQ_RECEIVE_EXIT”](#) na stránce 222.

XMSC_WMQ_RESOLVE_QUEUE_MANAGER

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Tato vlastnost se používá k získání názvu správce front, ke kterému je připojen.

Je-li použit s tabulkou CCDT (Client Channel Definition Table), může se tento název lišit od názvu správce front určeného v Továrně připojení.

XMSC_WMQ_RESOLVE_QUEUE_MANAGER_ID

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Tato vlastnost je naplněna identifikátorem ID správce front po připojení.

XMSC_WMQ_SECURITY_EXIT

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Identifikuje uživatelskou proceduru pro zabezpečení zprávy kanálu.

Hodnota vlastnosti je řetězec, který identifikuje uživatelskou proceduru zabezpečení kanálu a má následující formát:

libraryName(entryPointNázev)

kde:

- **libraryName** je úplná cesta ke spravované uživatelské knihovně DLL.
- **entryPointName** je název třídy kvalifikované oborem názvů

Příklad: C:\MySecurityExit.dll(MySecurityExitNameSpace.MySecurityExitClassName)

Maximální délka řetězce je 128 znaků.

Ve výchozím nastavení není vlastnost nastavena.

Tato vlastnost je relevantní pouze v případě, že se aplikace připojuje ke správci front ve spravovaném režimu klienta. Podporovány jsou také pouze spravované ukončení.

XMSC_WMQ_SECURITY_EXIT_INIT

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Uživatelská data, která jsou předána uživatelské proceduře pro zabezpečení zprávy kanálu při volání.

Maximální délka řetězce uživatelských dat je 32 znaků.

Ve výchozím nastavení není vlastnost nastavena.

Tato vlastnost je relevantní pouze v případě, že se aplikace připojuje ke správci front ve spravovaném klientském režimu a je nastavena vlastnost [“XMSC_WMQ_SECURITY_EXIT”](#) na stránce 223 .

XMSC_WMQ_SEND_EXIT

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Identifikuje uživatelskou proceduru pro odeslání zprávy kanálu.

Hodnota vlastnosti je řetězec. Uživatelská procedura odeslání kanálu má následující formát:

libraryName(entryPointNázev)

kde:

- **libraryName** je úplná cesta ke spravované uživatelské knihovně DLL.
- **entryPointName** je název třídy kvalifikované oborem názvů

Například C:\MySendExit.dll(MySendExitNameSpace.MySendExitClassName)

Ve výchozím nastavení není vlastnost nastavena.

Tato vlastnost je relevantní pouze v případě, že se aplikace připojuje ke správci front ve spravovaném režimu klienta. Podporovány jsou také pouze spravované ukončení.

XMSC_WMQ_SEND_EXIT_INIT

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Uživatelská data, která jsou předána uživatelským procedurám pro odeslání zprávy kanálu při volání.

Hodnota vlastnosti je řetězec jedné nebo více položek dat uživatele oddělených čárkami. Ve výchozím nastavení není vlastnost nastavena.

Pravidla pro uvedení uživatelských dat, která jsou předávána posloupnosti uživatelských procedur odeslání kanálu, jsou stejná jako pravidla pro uvedení uživatelských dat, která jsou předána posloupnosti uživatelských procedur příjmu kanálu. Z tohoto důvodu viz [“XMSC_WMQ_RECEIVE_EXIT_INIT”](#) na stránce 222.

Tato vlastnost je relevantní pouze v případě, že se aplikace připojuje ke správci front ve spravovaném klientském režimu a je nastavena vlastnost [“XMSC_WMQ_SEND_EXIT”](#) na stránce 223 .

XMSC_WMQ_SEND_CHECK_COUNT

Datový typ:

System.Int32

Vlastnost:

ConnectionFactory

Počet povolených odeslaných volání mezi kontrolou asynchronních chyb vložení během jedné netransakční relace XMS.

Při výchozím nastavení je tato vlastnost nastavena na hodnotu 0.

XMSC_WMQ_SHARE_CONV_ALLOWED

Datový typ:

System.Int32

Vlastnost:

ConnectionFactory

Použitelné objekty:

Dlouhý název nástroje pro administraci JMS: SHARECONVALLOWED

Krátký název nástroje pro administraci JMS: SCALD

Zda může připojení klienta sdílet svůj soket s dalšími propojeními XMS nejvyšší úrovně z téhož procesu do stejného správce front, pokud se definice kanálu shodují. Tato vlastnost slouží k povolení úplné izolace produktu Connections v samostatných soketech, pokud je to vyžadováno pro vývoj aplikací, údržbu nebo provozní důvody. Nastavení této vlastnosti pouze označuje rozhraní XMS , aby se základní sdílený soket sdílel. Neoznačuje, kolik připojení sdílí jeden soket. Počet připojení sdílejících soket je určen hodnotou SHARECNV vyjednanou mezi klientem WebSphere MQ Client a serverem WebSphere MQ .

Aplikace může nastavit následující pojmenované konstanty pro nastavení vlastnosti:

- XMSC_WMQ_SHARE_CONV_ALLOWED_FALSE-Připojení nesdílí socket.
- XMSCS_WMQ_SHARE_CONV_ALLOWED_TRUE-Připojení sdílí socket.

Při výchozím nastavení je vlastnost nastavena na hodnotu XMSCS_WMQ_SHARE_CONV_ALLOWED_ENABLED.

Tato vlastnost je relevantní pouze v případě, že se aplikace připojuje ke správci front v režimu klienta.

XMSC_WMQ_SSL_CERT_STORES

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Umístění serverů obsahující seznamy odvolaných certifikátů (CRL), které mají být použity v připojení SSL ke správci front.

Hodnota vlastnosti je seznam jedné nebo více adres URL oddělených čárkami. Každá adresa URL má následující formát:

```
[user[/password]@]ldap://[serveraddress][:portnum][, ...]
```

Tento formát je kompatibilní s základním formátem MQJMS, ale je tento formát rozšířen.

Je platné mít prázdný `serveraddress`. V tomto případě XMS předpokládá, že hodnota je řetězec "localhost".

Příklad seznamu je:

```
myuser/mypassword@ldap://server1.mycom.com:389  
ldap://server1.mycom.com  
ldap://  
ldap://:389
```

Pouze pro produkt .NET : Z produktu IBM MQ 8.0podporují spravovaná připojení k produktu IBM MQ (WMQ_CM_CLIENT) a nespravovaná připojení k produktu IBM MQ (WMQ_CM_CLIENT_UNMANAGED) obě podpory připojení TLS/SSL.

Ve výchozím nastavení není vlastnost nastavena.

Související informace

[Podpora zabezpečení SSL a TLS pro nespravovaného klienta .NET](#)

[Podpora zabezpečení SSL a TLS pro spravovaného klienta .NET](#)

XMSC_WMQ_SSL_CIPHER_SPEC

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Název specifikace CipherSpec, která má být použita v zabezpečeném připojení ke správci front.

Specifikace šifer, které můžete použít s podporou TLS IBM WebSphere MQ , jsou vypsány v následující tabulce. Požadujete-li osobní certifikát, určíte velikost klíče pro dvojici veřejný a soukromý klíč. Velikost klíče, která se použije během navázání komunikace přes zabezpečení SSL, je velikost uložená v certifikátu, pokud není určena CipherSpec, jak je uvedeno v tabulce. Při výchozím nastavení tato vlastnost není nastavena.

Název specifikace šifrování	Použitý protokol	Algoritmus přepočtu klíče	Šifrovací algoritmus	Šifrování bitů	FIPS ¹	Suite B 128 bitů	Suite B 192 bitů
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	SHA-1	AES	128	Ano	Ne	Ne
TLS_RSA_WITH_AES_256_CBC_SHA ²	TLS 1.0	SHA-1	AES	256	Ano	Ne	Ne
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	SHA-1	DES	56	Ne	Ne	Ne
TLS_RSA_WITH_3DES_EDE_CBC_SHA ⁴	TLS 1.0	SHA-1	3DES	168	Ano	Ne	Ne
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	SHA-256	AES	128	Ano	Ne	Ne
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	SHA-384	AES	256	Ano	Ne	Ne
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	SHA-256	AES	128	Ano	Ne	Ne
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	SHA-256	AES	256	Ano	Ne	Ne
ECDHE_ECDSA_RC4_128_SHA256	TLS 1.2	SHA-256	RC4	128	Ne	Ne	Ne
ECDHE_ECDSA_3DES_EDE_CBC_SHA256	TLS 1.2	SHA-256	3DES	168	Ano	Ne	Ne
ECDHE_RSA_RC4_128_SHA256	TLS 1.2	SHA-256	RC4	128	Ne	Ne	Ne
ECDHE_RSA_3DES_EDE_CBC_SHA256	TLS 1.2	SHA-256	3DES	168	Ano	Ne	Ne
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	SHA-256	AES	128	Ano	Ne	Ne
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	SHA-384	AES	256	Ano	Ne	Ne
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	SHA-256	AES	128	Ano	Ne	Ne
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	SHA-384	AES	256	Ano	Ne	Ne
ECDHE_ECDSA_AES_128_GCM_SHA256	TLS 1.2	SHA-256	AES	128	Ano	Ano	Ne
ECDHE_ECDSA_AES_256_GCM_SHA384	TLS 1.2	SHA-384	AES	256	Ano	Ne	Ano
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	SHA-256	AES	128	Ano	Ne	Ne
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	SHA-384	AES	256	Ano	Ne	Ne
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	SHA-256	Není	0	Ne	Ne	Ne
ECDHE_RSA_NULL_SHA256	TLS 1.2	SHA-256	Není	0	Ne	Ne	Ne

Název specifikace šifrování	Použitý protokol	Algoritmus přepočtu klíče	Šifrovací algoritmus	Šifrování bitů	FIPS ¹	Suite B 128 bitů	Suite B 192 bitů
ECDHE_ECDSA_NULL_SHA256	TLS 1.2	SHA-256	Není	0	Ne	Ne	Ne
TLS_RSA_WITH_NULL_NULL	TLS 1.2	Není	Není	0	Ne	Ne	Ne
TLS_RSA_WITH_RC4_128_SHA256	TLS 1.2	SHA-256	RC4	128	Ne	Ne	Ne

Notes:

1. Uvádí, zda CipherSpec vyhovuje standardu FIPS (Federal Information Processing Standards) 140-2. Informace o standardu FIPS a informace o tom, jak nakonfigurovat produkt WebSphere MQ for FIPS 140-2, najdete v tématu *Federal Information Processing Standards (FIPS)* v online dokumentaci produktu IBM WebSphere MQ .
2. Tuto volbu CipherSpec nelze použít k zabezpečení připojení z programu Průzkumník produktu WebSphere MQ ke správci front, pokud nejsou použity příslušné soubory neomezených zásad pro prostředí JRE používané průzkumníkem.
3. Tato specifikace šifrování byla certifikována FIPS 140-2 před 19. květnem 2007.
4. Je-li produkt WebSphere MQ konfigurován pro kompatibilní operaci FIPS 140-2, lze tuto volbu CipherSpec použít k přenosu až 32 GB dat před ukončením připojení s chybou AMQ9288. Chcete-li se této chybě vyhnout, buď se vyhnete použití trojitého DES (který je zamítnut), nebo povolte reset tajného klíče při použití této CipherSpec v konfiguraci FIPS 140-2.

Související informace

[Zabezpečení](#)

[Integrita dat zpráv](#)

[Určení specifikace CipherSpecs](#)

XMSC_WMQ_SSL_CIPHER_SUITE

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Název sady CipherSuite, která má být použita v rámci připojení TLS ke správci front. Protokol použitý při vyjednávání zabezpečeného připojení závisí na určené sadě CipherSuite.

Tato vlastnost má následující kanonické hodnoty:

- 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

Tato hodnota může být dodána jako alternativa k `XMSCS_WMQ_SSL_CIPHER_SPEC`.

Je-li zadána neprázdná hodnota pro proměnnou `XMSCS_WMQ_SSL_CIPHER_SPEC`, tato hodnota přepíše nastavení hodnoty `XCOMS_WMQ_SSL_CIPHER_SUITE`. Pokud `XMSCS_WMQ_SSL_CIPHER_SPEC` nemá hodnotu, použije se jako šifrovací sada, která má být poskytnuta sadě GSKit, hodnota `XMSCS_WMQ_SSL_CIPHER_SUITE`. V tomto případě je hodnota mapována na ekvivalentní hodnotu `CipherSpec`, jak je popsáno v tématu “[Mapování názvů CipherSuite a CipherSpec pro připojení ke správci front produktu IBM MQ](#)” na stránce 65.

Pokud jsou parametry `XMSCS_WMQ_SSL_CIPHER_SPEC` a `XMSCS_WMQ_SSL_CIPHER_SUITE` prázdné, bude pole `pChDef->SSLCipherSpec` vyplněno mezerami.

Pouze pro produkt .NET : Z produktu IBM MQ 8.0podporují spravovaná připojení k produktu IBM MQ (`WMQ_CM_CLIENT`) a nespravovaná připojení k produktu IBM MQ (`WMQ_CM_CLIENT_UNMANAGED`) obě podpory připojení TLS/SSL.

Ve výchozím nastavení není vlastnost nastavena.

Související informace

[Podpora zabezpečení SSL a TLS pro nespravovaného klienta .NET](#)

[Podpora zabezpečení SSL a TLS pro spravovaného klienta .NET](#)

XMSC_WMQ_SSL_CRYPTO_HW

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Podrobnosti konfigurace šifrovacího hardwaru připojeného k systému klienta.

Tato vlastnost má následující kanonické hodnoty:

- `GSK_ACCELERATOR_RAINBOW_CS_OFF`
- `GSK_ACCELERATOR_RAINBOW_CS_ON`
- `GSK_ACCELERATOR_NCIPHER_NF_OFF`
- `GSK_ACCELERATOR_NCIPHER_NF_ON`

Pro kryptografický hardware PKCS11 existuje speciální formát pro kryptografický hardware (kde `DriverPath`, `TokenLabel` a `TokenPassword` jsou uživatelem zadané řetězce):

```
GSK_PKCS11=PKCS#11 DriverPath; PKCS#11 TokenLabel;PKCS#11 TokenPassword
```

Příkaz XMS neinterpretuje ani nemění obsah řetězce. Zkopíruje dodanou hodnotu až do limitu 256 jednobajtových znaků do struktury `MQSCO.CryptoHardware`.

Pouze pro produkt .NET : Z produktu IBM MQ 8.0podporují spravovaná připojení k produktu IBM MQ (`WMQ_CM_CLIENT`) a nespravovaná připojení k produktu IBM MQ (`WMQ_CM_CLIENT_UNMANAGED`) obě podpory připojení TLS/SSL.

Ve výchozím nastavení není vlastnost nastavena.

Související informace

[Podpora zabezpečení SSL a TLS pro nespravovaného klienta .NET](#)

[Podpora zabezpečení SSL a TLS pro spravovaného klienta .NET](#)

XMSC_WMQ_SSL_FIPS_REQUIRED

Datový typ:

Logická hodnota

Vlastnost:

ConnectionFactory

Hodnota této vlastnosti určuje, zda aplikace může nebo nemůže používat šifrovací sady, které nevyhovují standardu FIPS. Je-li tato vlastnost nastavena na hodnotu true, pro připojení klientského serveru se používají pouze algoritmy FIPS.

Tato vlastnost může mít následující hodnoty, které se převádějí na dvě kanonické hodnoty pro MQSCO.FipsRequired:

<i>Tabulka 34. Tabulka hodnot pro MQSCO.FlipsRequired</i>		
Hodnota	Popis	Odpovídající hodnota MQSCO.FipsRequired
ne	Lze použít libovolnou CipherSpec .	MQSSL_FIPS_NO (výchozí)
ano	Ve specifikaci CipherSpec pro toto připojení klienta lze použít pouze šifrovací algoritmy s certifikací FIPS.	MQSSL_FIPS_YES

XMS zkopíruje příslušnou hodnotu do MQSCO.FipsRequired před voláním MQCONNX.

Parametr MQSCO.FipsRequired je k dispozici pouze z produktu WebSphere MQ verze 6. Je-li tato vlastnost nastavena na WebSphere MQ verze 5.3, produkt XMS se nepokusí navázat připojení ke správci front a místo toho vydá příslušnou výjimku.

Pouze pro produkt .NET : Z produktu IBM MQ 8.0podporují spravovaná připojení k produktu IBM MQ (WMQ_CM_CLIENT) a nespravovaná připojení k produktu IBM MQ (WMQ_CM_CLIENT_UNMANAGED) obě podpory připojení TLS/SSL.

Související informace

[Podpora zabezpečení SSL a TLS pro nespravovaného klienta .NET](#)

[Podpora zabezpečení SSL a TLS pro spravovaného klienta .NET](#)

XMSC_WMQ_SSL_KEY_REPOSITORY

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Umístění souboru databáze klíčů, ve kterém jsou uloženy klíče a certifikáty.

XMS zkopíruje řetězec až do limitu 256 jednobajtových znaků do struktury MQSCO.KeyRepository . Produkt WebSphere MQ interpretuje tento řetězec jako název souboru včetně úplné cesty.

Pouze pro produkt .NET : Z produktu IBM MQ 8.0podporují spravovaná připojení k produktu IBM MQ (WMQ_CM_CLIENT) a nespravovaná připojení k produktu IBM MQ (WMQ_CM_CLIENT_UNMANAGED) obě podpory připojení TLS/SSL.

Ve výchozím nastavení není vlastnost nastavena.

Související informace

[Podpora zabezpečení SSL a TLS pro nespravovaného klienta .NET](#)

[Podpora zabezpečení SSL a TLS pro spravovaného klienta .NET](#)

XMSC_WMQ_SSL_KEY_RESETCOUNT

Datový typ:

System.Int32

Vlastnost:

ConnectionFactory

Hodnota KeyResetCount představuje celkový počet nešifrovaných bajtů odeslaných a přijatých v rámci konverzace SSL, než je znovu vyjednán tajný klíč. Počet bajtů zahrnuje řídicí informace odeslané agentem MCA.

Produkt XMS zkopíruje hodnotu, kterou zadáte pro tuto vlastnost, do pole MQSCO.KeyResetCount před voláním MQCONN.

Parametr MQSCO.KeyRestCount je k dispozici pouze z produktu WebSphere MQ verze 6. Je-li tato vlastnost nastavena na WebSphere MQ verze 5.3, produkt XMS se nepokusí navázat připojení ke správci front a místo toho vydá příslušnou výjimku.

Pouze pro produkt .NET : Z produktu IBM MQ 8.0podporují spravovaná připojení k produktu IBM MQ (WMQ_CM_CLIENT) a nespravovaná připojení k produktu IBM MQ (WMQ_CM_CLIENT_UNMANAGED) obě podpory připojení TLS/SSL.

Výchozí hodnota této vlastnosti je nula, což znamená, že tajné klíče nejsou nikdy znovu vyjednávány.

Související informace

[Podpora zabezpečení SSL a TLS pro nespravovaného klienta .NET](#)

[Podpora zabezpečení SSL a TLS pro spravovaného klienta .NET](#)

XMSC_WMQ_SSL_PEER_NAME

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Název typu peer, které se použije při připojení SSL ke správci front.

Pro tuto vlastnost není k dispozici žádný seznam kanonických hodnot. Namísto toho musíte sestavit tento řetězec podle pravidel pro SSLPEER.

Příklad názvu partnera:

```
"CN=John Smith, O=IBM ,OU=Test , C=GB"
```

XMS zkopíruje řetězec do správné jednobajtové kódové stránky a umístí správné hodnoty do MQCD.SSLPeerNamePtr a MQCD.SSLPeerNameLength před voláním MQCONN.

Tato vlastnost je relevantní pouze v případě, že se aplikace připojuje ke správci front v režimu klienta.

Pouze pro produkt .NET : Z produktu IBM MQ 8.0podporují spravovaná připojení k produktu IBM MQ (WMQ_CM_CLIENT) a nespravovaná připojení k produktu IBM MQ (WMQ_CM_CLIENT_UNMANAGED) obě podpory připojení TLS/SSL.

Ve výchozím nastavení není vlastnost nastavena.

Související informace

[Podpora zabezpečení SSL a TLS pro nespravovaného klienta .NET](#)

[Podpora zabezpečení SSL a TLS pro spravovaného klienta .NET](#)

[SSLPEERNAME](#)

XMSC_WMQ_SYNCPOINT_ALL_GETS

Datový typ:

System.Boolean

Vlastnost:

ConnectionFactory

Určuje, zda musí být všechny zprávy načteny z front v rámci řízení synchronizačního bodu.

Platné hodnoty vlastnosti jsou následující:

Platná hodnota

ne

ano

Význam

Jsou-li okolnosti vhodné, může klient produktu XMS načítat zprávy z front mimo ovládací prvek bodu synchronizace.

Klient produktu XMS musí načíst všechny zprávy z front v rámci řízení synchronizačního bodu.

Výchozí hodnota je false.

XMSC_WMQ_TARGET_CLIENT**Datový typ:**

System.Int32

Vlastnost:

Místo určení

Název použitý v identifikátoru URI:

targetClient

Určuje, zda zprávy odeslané do cíle obsahují záhlaví MQRFH2.

Pokud aplikace odešle zprávu obsahující záhlaví MQRFH2 , přijímající aplikace musí být schopna zpracovat záhlaví.

Platné hodnoty vlastnosti jsou následující:

Platná hodnota

XMSC_WMQ_TARGET_DEST_JMS

Význam

Zprávy odeslané do místa určení obsahují záhlaví MQRFH2 . Tuto hodnotu zadejte v případě, že aplikace odesílá zprávy do jiné aplikace produktu XMS , aplikace WebSphere JMS. nebo nativní aplikaci produktu WebSphere MQ , která je navržena tak, aby zpracoval záhlaví MQRFH2 .

XMSC_WMQ_TARGET_DEST_MQ

Zprávy odeslané do místa určení neobsahují záhlaví MQRFH2 . Tuto hodnotu zadejte v případě, že aplikace odesílá zprávy nativní aplikaci produktu WebSphere MQ , která není navržena pro zpracování záhlaví MQRFH2 .

Výchozí hodnota je XMSCS_WMQ_TARGET_DEST_JMS.

XMSC_WMQ_TEMP_Q_PREFIX**Datový typ:**

Řetězec

Vlastnost:

ConnectionFactory

Předpona používaná k vytvoření názvu dynamické fronty produktu WebSphere MQ , která se vytvoří, když aplikace vytvoří XMS dočasnou frontu.

Pravidla pro vytvoření předpony jsou stejná jako pravidla pro vytvoření obsahu pole **DynamicQName** v deskriptoru objektu, ale poslední neprázdný znak musí být hvězdička (*). Pokud tato vlastnost není nastavena, je použita hodnota CSQ . * v systémech z/OS a AMQ . * na ostatních platformách. Ve výchozím nastavení není vlastnost nastavena.

Tato vlastnost je relevantní pouze v doméně dvoubodový .

XMSC_WMQ_TEMP_TOPIC_PREFIX**Datový typ:**

Řetězec

Vlastnost:

ConnectionFactory, cíl

Při vytváření dočasných témat vygeneruje XMS řetězec tématu ve tvaru "TEMP/TEMPTOPICPREFIX/unique_id", nebo pokud tato vlastnost obsahuje výchozí hodnotu, vygeneruje se tento řetězec "TEMP/unique_id". Určení neprázdné hodnoty umožní definování specifických modelových front za účelem vytvoření spravovaných front pro odběratele dočasných témat vytvořených v rámci tohoto připojení.

Jakýkoli řetězec, který není typu null, sestávající pouze z platných znaků pro řetězec tématu IBM WebSphere MQ je platnou hodnotou pro tuto vlastnost.

Při výchozím nastavení je tato vlastnost nastavena na hodnotu "" (prázdný řetězec).

Poznámka: Tato vlastnost je relevantní pouze v doméně publikování/odběru.

XMSC_WMQ_TEMPORARY_MODEL**Datový typ:**

Řetězec

Vlastnost:

ConnectionFactory

Název modelové fronty produktu WebSphere MQ , ze které se vytvoří dynamická fronta, když aplikace vytvoří XMS dočasnou frontu.

Výchozí hodnota vlastnosti je SYSTEM.DEFAULT.MODEL.QUEUE.

Tato vlastnost je relevantní pouze v doméně dvoubodový .

XMSC_WMQ_WILDCARD_FORMAT**Datový typ:**

System.Int32

Vlastnost:

ConnectionFactory, cíl

Tato vlastnost určuje, která verze syntaxe zástupných znaků má být použita.

Při použití publikování/odběru s IBM WebSphere MQ '*' a '?' jsou považovány za zástupné znaky. Whereas '#' and '+' are treated as wildcards when using publish subscribe with IBM Integration Bus. Tato vlastnost nahrazuje vlastnost XMSCS_WMQ_BROKER_VERSION.

Platné hodnoty pro tuto vlastnost jsou:

XMSC_WMQ_WILDCARD_TOPIC_ONLY

Rozlišuje pouze zástupné znaky na úrovni tématu, tj. '#' a '+' jsou považovány za zástupné znaky. Tato hodnota je stejná jako XMSC_WMQ_BROKER_V2.

XMSC_WMQ_WILDCARD_CHAR_ONLY

Uvědomí pouze zástupné znaky znaků, např. "*" a '?' jsou považovány za zástupné znaky. Tato hodnota je stejná jako XMSC_WMQ_BROKER_V1.

Při výchozím nastavení je tato vlastnost nastavena na hodnotu XMSCS_WMQ_WILDCARD_TOPIC_ONLY.

XMSC_WPM_BUS_NAME**Datový typ:**

Řetězec

Vlastnost:

ConnectionFactory a Destination

Název použitý v identifikátoru URI:

busName

U továrny připojení se jedná o název sběrnice SIBus, ke které se aplikace připojuje, nebo pro cíl, název sběrnice SIBus, ve které cíl existuje.

U místa určení, které je tématem, je tato vlastnost názvem sběrnice pro integraci služeb, ve které existuje přidružený prostor tématu. Tento prostor tématu je určen vlastností `XMSCS_WPM_TOPIC_SPACE`.

Není-li vlastnost nastavena pro cíl, předpokládá se, že fronta nebo přidružený prostor tématu existují ve sběrnici pro integraci služeb, ke které se aplikace připojuje.

Ve výchozím nastavení není vlastnost nastavena.

XMSC_WPM_CONNECTION_PROTOCOL

Datový typ:

System.Int32

Vlastnost:

Připojení

Komunikační protokol použitý pro připojení k jádru systému zpráv. Tato vlastnost je jen pro čtení.

Možné hodnoty vlastnosti jsou následující:

Hodnota	Význam
XMSC_WPM_CP_HTTP	Připojení používá protokol HTTP přes TCP/IP.
XMSC_WPM_CPP_TCP	Připojení používá protokol TCP/IP.

XMSC_WPM_CONNECTION_PROXAS

Datový typ:

System.Int32

Vlastnost:

ConnectionFactory

Nastavení blízkosti připojení pro připojení. Tato vlastnost určuje způsob zavření stroje systému zpráv, ke kterému se aplikace připojuje, musí být na server samozavedení.

Platné hodnoty vlastnosti jsou následující:

Platná hodnota	Nastavení blízkosti připojení
XMSC_WPM_CONNECTION_PROXIMITY_BUS	Sběrnice
XMSC_WPM_CONNECTION_PROXIMITY_CLUSTER	Klastr
XMSC_WPM_CONNECTION_PROXIMITY_HOST	Hostitel
XMSC_WPM_CONNECTION_PROXIMITY_SERVER	Server

Výchozí hodnota je `XMSCS_WPM_CONNECTION_PROXIMITY_BUS`.

XMSC_WPM_DUR_SUB_HOME

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Název použitý v identifikátoru URI:

durableSubscriptionHome

Název jádra systému zpráv, v nichž jsou spravovány všechny trvalé odběry pro připojení nebo cíl. Zprávy, které mají být doručovány na trvalé odběratele, jsou uloženy v bodu publikování se stejným strojem systému zpráv.

Před vytvořením trvalého odběratele, který připojení používá, musí být zadán domovský adresář odběru pro připojení. Jakákoli hodnota uvedená pro cíl přepíše hodnotu uvedenou pro připojení.

Ve výchozím nastavení není vlastnost nastavena.

Tato vlastnost je relevantní pouze v doméně Publikování/odběr .

XMSC_WPM_HOST_NAME

Datový typ:

Řetězec

Vlastnost:

Připojení

Název hostitele nebo adresa IP systému, který obsahuje jádro systému zpráv, ke kterému je aplikace připojena. Tato vlastnost je jen pro čtení.

XMSC_WPM_LOCAL_ADDRESS

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Pro připojení ke sběrnici SIBus tato vlastnost určuje rozhraní lokální sítě, lokálního portu nebo rozsahu lokálních portů, nebo obojí.

Hodnota vlastnosti je řetězec s následujícím formátem:

[*název_hostitele*] [(*low_port*) [,*high_port*]]

Význam proměnných je následující:

název_hostitele

Název hostitele nebo adresa IP lokálního síťového rozhraní, které má být použito pro připojení.

Poskytnutí těchto informací je nezbytné pouze v případě, že systém, na kterém je aplikace spuštěna, má dvě nebo více síťových rozhraní a vy potřebujete mít možnost určit, které rozhraní musí být pro připojení použito. Má-li systém pouze jedno síťové rozhraní, lze použít pouze toto rozhraní. Má-li systém dvě nebo více síťových rozhraní a nespecifikujete, které rozhraní musí být použito, je rozhraní vybráno náhodně.

nizká_port

Číslo lokálního portu, které má být použito pro připojení.

Je-li zadána také hodnota *high_port* , hodnota *low_port* je interpretována jako nejnižší číslo portu v rozsahu čísel portů.

vysoká_port

Nejvyšší číslo portu v rozsahu čísel portů. Jeden z portů v uvedeném rozsahu musí být použit pro připojení.

Zde je několik příkladů platných hodnot vlastnosti:

PLANET NAME
9.20.4.98
JUPER (1000)
9.20.4.98(1000,2000)
(1000)
(1000,2000)

Ve výchozím nastavení není vlastnost nastavena.

XMSC_WPM_ME_NAME

Datový typ:

Řetězec

Vlastnost:

Připojení

Název jádra systému zpráv, ke kterému je aplikace připojena. Tato vlastnost je jen pro čtení.

XMSC_WPM_NON_PERSISTENT_MAP**Datový typ:**

System.Int32

Vlastnost:

ConnectionFactory

Úroveň spolehlivosti přechodných zpráv, které se odesílají pomocí připojení.

Platné hodnoty vlastnosti jsou následující:

Platná hodnota

XMSC_WPM_MAPPING_AS_DESTINATION

XMSC_WPM_MAPPING_BEST_EFFORT_NON_
TrvalýXMSC_WPM_MAPPING_EXPRESS_NON_
TrvalýXMSC_WPM_MAPPING_RELIABLE_NON_
Trvalý

XMSC_WPM_MAPPING_RELIABLE_PERSISTENT

XMSC_WPM_MAPPING_ASSURED_PERSISTENT

Úroveň spolehlivosti

Určeno výchozí úrovní spolehlivosti zadanou pro frontu nebo prostor tématu v rámci sběrnice pro integraci služeb.

Nejlepší snaha, přechodné

Expresní přechodné

Spolehlivé přechodné

Spolehlivá trvalá

Zajištěné,

Výchozí hodnota je XMSCS_WPM_MAPPING_EXPRESS_NON_PERSISTENT.

XMSC_WPM_PERSISTENT_MAP**Datový typ:**

System.Int32

Vlastnost:

ConnectionFactory

Úroveň spolehlivosti trvalých zpráv, které se odesílají pomocí připojení.

Platné hodnoty vlastnosti jsou následující:

Platná hodnota

XMSC_WPM_MAPPING_AS_DESTINATION

XMSC_WPM_MAPPING_BEST_EFFORT_NON_
TrvalýXMSC_WPM_MAPPING_EXPRESS_NON_
Trvalý**Úroveň spolehlivosti**

Určeno výchozí úrovní spolehlivosti zadanou pro frontu nebo prostor tématu v rámci sběrnice pro integraci služeb.

Nejlepší snaha, přechodné

Expresní přechodné

Platná hodnota

XMSC_WPM_MAPPING_RELIABLE_NON_
Trvalý

XMSC_WPM_MAPPING_RELIABLE_PERSISTENT

XMSC_WPM_MAPPING_ASSURED_PERSISTENT

Úroveň spolehlivosti

Spolehlivé přechodné

Spolehlivá trvalá

Zajištěné,

Výchozí hodnota je XMSCS_WPM_MAPPING_RELIABLE_PERSISTENT.

XMSC_WPM_PORT**Datový typ:**

System.Int32

Vlastnost:

Připojení

Číslo portu naslouchaného v jádru systému zpráv, ke kterému je aplikace připojena. Tato vlastnost je jen pro čtení.

XMSC_WPM_PROVIDER_ENDPOINTS**Datový typ:**

Řetězec

Vlastnost:

ConnectionFactory

Sekvence jednoho nebo více adres koncového bodu zaváděcích serverů. Adresy koncových bodů jsou odděleny čárkami.

Server samozavedení je aplikační server, který je zodpovědný za výběr stroje systému zpráv, ke kterému se aplikace připojuje. Adresa koncového bodu serveru samozavedení má následující formát:

název_hostitele:číslo_portu:název_řetězu

Významy součástí adresy koncového bodu jsou následující:

název_hostitele

Název hostitele nebo adresa IP systému, na kterém je umístěn server samozavedení. Není-li zadán žádný název hostitele nebo adresa IP, je standardní hodnota localhost.

číslo_portu

Číslo portu, na kterém server samozavedení naslouchá příchozím požadavkům. Není-li zadáno žádné číslo portu, je výchozí hodnota 7276.

název_řetězu

Název transportního řetězu samozavedení používaného serverem samozavedení. Platné hodnoty jsou:

Platná hodnota

XMSC_WPM_BOOTSTRAP_HTTP

XMSC_WPM_BOOTSTRAP_HTTPS

XMSC_WPM_BOOTSTRAP_SSL

XMSC_WPM_BOOTSTRAP_TCP

Název transportního řetězu samozavedení

System zpráv BootstrapTunneled

BootstrapTunneledSecureMessaging

System zpráv BootstrapSecure

System zpráv BootstrapBasic

Není-li uveden žádný název, výchozí hodnota je XMSCS_WPM_BOOTSTRAP_TCP.

Není-li zadána žádná adresa koncového bodu, bude použita výchozí hodnota localhost:7276:BootstrapBasicMessaging.

XMSC_WPM_SSL_CIPHER_SUITE

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Název sady CipherSuite , který má být použit pro připojení TLS k stroji systému zpráv produktu WebSphere Service Integration Bus . Protokol použitý při vyjednávání zabezpečeného připojení závisí na určené sadě CipherSuite.

<i>Tabulka 35. Volby CipherSuite pro připojení ke stroji systému zpráv produktu WebSphere Service Integration Bus</i>	
Šifrovací sada	Použitý protokol
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

Notes:

1. TLS_RSA_WITH_AES_128_CBC_SHA a TLS_RSA_WITH_AES_256_CBC_SHA CipherSuites jsou podporovány pouze v produktu Windows nebo Solaris . (To je diktováno sadou GSKit.)
2. TLS_RSA_WITH_3DES_EDE_CBC_SHA je zamítnutý. Nicméně lze ji přesto použít k přenosu až 32 GB dat před ukončením připojení s chybou AMQ9288. Chcete-li se této chybě vyhnout, je třeba při použití této CipherSuite zabránit použití trojitého DES, nebo povolit resetování tajného klíče.

Pro tuto vlastnost není žádná výchozí hodnota. Chcete-li použít zabezpečení SSL nebo TLS, musíte zadat hodnotu této vlastnosti, jinak se aplikace nebude moci úspěšně připojit k serveru.

XMSC_WPM_SSL_FIPS_REQUIRED

Datový typ:

Logická hodnota

Vlastnost:

ConnectionFactory

Hodnota této vlastnosti určuje, zda aplikace může nebo nemůže používat šifrovací sady, které nevyhovují standardu FIPS. Je-li tato vlastnost nastavena na hodnotu true, pro připojení typu klient-server se používají pouze algoritmy FIPS. Nastavení hodnoty této vlastnosti na TRUE zabrání aplikaci používat šifrovací sady, které vyhovují standardu FIPS.

Při výchozím nastavení je vlastnost nastavena na hodnotu FALSE (to znamená, režim FIPS je vypnutý).

XMSC_WPM_SSL_KEY_REPOSITORY

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Cesta k souboru, který je souborem svazku klíčů obsahujícím veřejné nebo soukromé klíče, který má být použit v zabezpečeném připojení.

Nastavení vlastnosti souboru svazku klíčů na speciální hodnotu XMSCS_WPM_SSL_MS_CERTIFICATE_STORE určuje použití databáze klíčů produktu Microsoft Windows . Pomocí databáze klíčů produktu Microsoft Windows , která se nachází pod **Ovládacím panelem** >

Možnosti Internetu > Obsah > Certifikáty, je třeba odebrat potřebu samostatné databáze souborů s klíči. Použití této konstanty v systému Windows x64 a jiných platformách není povoleno.

Ve výchozím nastavení není vlastnost nastavena.

XMSC_WPM_SSL_KEYRING_LABEL

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Certifikát, který má být použit při ověřování na serveru. Není-li zadána žádná hodnota, bude použit výchozí certifikát.

Ve výchozím nastavení není vlastnost nastavena.

XMSC_WPM_SSL_KEYRING_PW

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Heslo pro soubor svazku klíčů.

Tuto vlastnost lze použít jako alternativu k použití souboru XMSCS_WPM_SSL_KEYRING_STASH_FILE ke konfiguraci hesla pro soubor svazku klíčů.

Ve výchozím nastavení není vlastnost nastavena.

XMSC_WPM_SSL_KEYRING_STASH_FILE

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Název binárního souboru obsahujícího heslo souboru úložiště klíčů.

Tuto vlastnost lze použít jako alternativu k použití souboru XMSCS_WPM_SSL_KEYRING_PW ke konfiguraci hesla pro soubor svazku klíčů.

Ve výchozím nastavení není vlastnost nastavena.

XMSC_WPM_TARGET_GROUP

Datový typ:

Řetězec

Vlastnost:

ConnectionFactory

Název cílové skupiny jader systému zpráv. Povaha cílové skupiny je určena vlastností XMSCS_WPM_TARGET_TYPE.

Tuto vlastnost nastavte v případě, že chcete omezit hledání stroje systému zpráv na podskupinu strojů systému zpráv v rámci sběrnice pro integraci služeb. Chcete-li, aby vaše aplikace byla schopna připojit se k libovolnému stroji systému zpráv v rámci sběrnice pro integraci služeb, nenastavujte tuto vlastnost.

Ve výchozím nastavení není vlastnost nastavena.

XMSC_WPM_TARGET_VÝZNAMNOST

Datový typ:

System.Int32

Vlastnost:

ConnectionFactory

Významnost cílové skupiny jader systému zpráv.

Platné hodnoty vlastnosti jsou následující:

Platná hodnotaXMSC_WPM_TARGET_SIGNIFICANCE_
PreferovanýXMSC_WPM_TARGET_SIGNIFICANCE_
POVINNÉ**Význam**

Je-li k dispozici stroj systému zpráv v cílové skupině, je vybrán. Jinak je vybrán stroj systému zpráv mimo cílovou skupinu za předpokladu, že se nachází ve stejné sběrnici pro integraci služeb.

Vybraný stroj systému zpráv musí být v cílové skupině. Není-li stroj systému zpráv v cílové skupině k dispozici, dojde k selhání procesu připojení.

Výchozí hodnota vlastnosti je XMSCS_WPM_TARGET_SIGNIFICANCE_PREFERRED.

XMSC_WPM_TARGET_TRANSPORT_CHAIN**Datový typ:**

Řetězec

Vlastnost:

ConnectionFactory

Název transportu příchozích požadavků, který musí aplikace používat pro připojení k jádru systému zpráv.

Hodnotou této vlastnosti může být název libovolného příchozího transportního řetězu, který je k dispozici na aplikačním serveru, který je hostitelem stroje systému zpráv. Pro jeden z předdefinovaných příchozích transportních řetězů je poskytnuta následující pojmenovaná konstanta:

pojmenovaná konstanta

XMSC_WPM_TARGET_TRANSPORT_CHAIN_BASIC

Název dopravního řetězce

Systém zpráv InboundBasic

Výchozí hodnota vlastnosti je XMSCS_WPM_TARGET_TRANSPORT_CHAIN_BASIC.

XMSC_WPM_TARGET_TYPE**Datový typ:**

System.Int32

Vlastnost:

ConnectionFactory

Typ cílové skupiny jader systému zpráv. Tato vlastnost určuje charakter cílové skupiny identifikované vlastností XMSCS_WPM_TARGET_GROUP.

Platné hodnoty vlastnosti jsou následující:

Platná hodnota

XMSC_WPM_TARGET_TYPE_BUSMEMBER

XMSC_WPM_TARGET_TYPE_CUSTOM

Význam

Název cílové skupiny je název člena sběrnice. Cílová skupina je všechny stroje systému zpráv v rámci člena sběrnice.

Název cílové skupiny je název skupiny strojů systému zpráv definovaných uživatelem. Cílová skupina je všechny stroje systému zpráv, které jsou registrovány ve skupině definované uživatelem.

Platná hodnota

XMSC_WPM_TARGET_TYPE_ME

Význam

Název cílové skupiny je název stroje systému zpráv. Cílová skupina je určený stroj systému zpráv.

Ve výchozím nastavení není vlastnost nastavena.

XMSC_WPM_TEMP_Q_PREFIX**Datový typ:**

Řetězec

Vlastnost:

ConnectionFactory

Předpona použitá k vytvoření názvu dočasné fronty, která je vytvořena ve sběrnici pro integraci služeb, když aplikace vytvoří XMS dočasnou frontu. Předpona může obsahovat až 12 znaků.

Název dočasné fronty začíná znaky "_Q" následovaným předponou. Zbytek názvu se skládá ze systémem generovaných znaků.

Ve výchozím nastavení není vlastnost nastavena, což znamená, že název dočasné fronty nemá předponu.

Tato vlastnost je relevantní pouze v doméně dvoubodový .

XMSC_WPM_TEMP_TOPIC_PREFIX**Datový typ:**

Řetězec

Vlastnost:

ConnectionFactory

Předpona používaná k vytvoření názvu dočasného tématu vytvořeného aplikací. Předpona může obsahovat až 12 znaků.

Název dočasného tématu začíná znaky "_T", za nimiž následuje předpona. Zbytek názvu se skládá ze systémem generovaných znaků.

Ve výchozím nastavení není vlastnost nastavena, což znamená, že název dočasného tématu nemá předponu.

Tato vlastnost je relevantní pouze v doméně Publikování/odběr .

XMSC_WPM_TOPIC_SPACE**Datový typ:**

Řetězec

Vlastnost:

Místo určení

Název použitý v identifikátoru URI:

topicSpace

Název prostoru témat, který obsahuje dané téma. Pouze místo určení, které je téma, může mít tuto vlastnost.

Ve výchozím nastavení není vlastnost nastavena, což znamená, že se předpokládá výchozí prostor tématu.

Tato vlastnost je relevantní pouze v doméně Publikování/odběr .

Poznámky

Tyto informace byly vyvinuty pro produkty a služby poskytované v USA.

Společnost IBM nemusí nabízet produkty, služby nebo funkce uvedené v tomto dokumentu v jiných zemích. Informace o produktech a službách, které jsou ve vaší oblasti aktuálně dostupné, získáte od místního zástupce společnosti IBM. Odkazy na produkty, programy nebo služby společnosti IBM v této publikaci nejsou míněny jako vyjádření nutnosti použití pouze uvedených produktů, programů či služeb společnosti IBM. Místo toho lze použít jakýkoli funkčně ekvivalentní produkt, program nebo službu, které neporušují žádná práva k duševnímu vlastnictví IBM. Ověření funkčnosti produktu, programu nebo služby pocházející od jiného výrobce je však povinností uživatele.

Společnost IBM může vlastnit patenty nebo nevyřízené žádosti o patenty zahrnující předměty popsané v tomto dokumentu. Vlastnictví tohoto dokumentu neposkytuje licenci k těmto patentům. Dotazy týkající se licencí můžete posílat písemně na adresu:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Odpovědi na dotazy týkající se licencí pro dvoubajtové znakové sady (DBCS) získáte od oddělení IBM Intellectual Property Department ve vaší zemi, nebo tyto dotazy můžete zasílat písemně na adresu:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

Následující odstavec se netýká Velké Británie nebo kterékoliv jiné země, kde taková opatření odporují místním zákonům: SPOLEČNOST INTERNATIONAL BUSINESS MACHINES CORPORATION TUTO PUBLIKACI POSKYTUJE TAKOVOU, "JAKÁ JE", BEZ JAKÝCHKOLIV ZÁRUK, VYJÁDŘENÝCH VÝSLOVNĚ NEBO VYPLÝVAJÍCÍCH Z OKOLNOSTÍ, VČETNĚ, A TO ZEJMÉNA, ZÁRUK NEPORUŠENÍ PRÁV TŘETÍCH STRAN, PRODEJNOSTI NEBO VHODNOSTI PRO URČITÝ ÚČEL VYPLÝVAJÍCÍCH Z OKOLNOSTÍ. Některé právní řády u určitých transakcí nepřipouštějí vyloučení záruk výslovně vyjádřených nebo vyplývajících z okolností, a proto se na vás toto omezení nemusí vztahovat.

Uvedené údaje mohou obsahovat technické nepřesnosti nebo typografické chyby. Údaje zde uvedené jsou pravidelně upravovány a tyto změny budou zahrnuty v nových vydáních této publikace. Společnost IBM může kdykoli bez upozornění provádět vylepšení nebo změny v produktech či programech popsaných v této publikaci.

Veškeré uvedené odkazy na webové stránky, které nespravuje společnost IBM, jsou uváděny pouze pro referenci a v žádném případě neslouží jako záruka funkčnosti těchto webů. Materiály uvedené na tomto webu nejsou součástí materiálů pro tento produkt IBM a použití uvedených stránek je pouze na vlastní nebezpečí.

Společnost IBM může použít nebo distribuovat jakékoli informace, které jí sdělíte, libovolným způsobem, který společnost považuje za odpovídající, bez vyžádání vašeho svolení.

Vlastníci licence k tomuto programu, kteří chtějí získat informace o možnostech (i) výměny informací s nezávisle vytvořenými programy a jinými programy (včetně tohoto) a (ii) oboustranného využití vyměňovaných informací, mohou kontaktovat informační středisko na adrese:

IBM Corporation
Koordinátor spolupráce softwaru, oddělení 49XA
148 00 Praha 4-Chodby

148 00 Praha 4-Chodov
U.S.A.

Poskytnutí takových informací může být podmíněno dodržením určitých podmínek a požadavků zahrnujících v některých případech uhrazení stanoveného poplatku.

IBM poskytuje licencovaný program popsany v těchto informacích a veškeré dostupné licencované materiály na základě podmínek smlouvy IBM Customer Agreement, IBM International Program License Agreement nebo jiné ekvivalentní smlouvy mezi námi.

Jakékoli údaje o výkonnosti obsažené v této publikaci byly zjištěny v řízeném prostředí. Výsledky získané v jakémkoli jiném operačním prostředí se proto mohou výrazně lišit. Některá měření mohla být prováděna na vývojových verzích systémů a není zaručeno, že tato měření budou stejná i na běžně dostupných systémech. Některá měření mohla být navíc odhadnuta pomocí extrapolace. Skutečné výsledky mohou být jiné. Čtenáři tohoto dokumentu by měli zjistit použitelné údaje pro své specifické prostředí.

Informace týkající se produktů jiných výrobců pocházejí od dodavatelů těchto produktů, z jejich veřejných oznámení nebo z jiných veřejně dostupných zdrojů. Společnost IBM tyto produkty netestovala a nemůže potvrdit správný výkon, kompatibilitu ani žádné jiné výroky týkající se produktů jiných výrobců než IBM. Otázky týkající se kompatibility produktů jiných výrobců by měly být směřovány dodavatelům těchto produktů.

Veškerá tvrzení týkající se budoucího směru vývoje nebo záměrů společnosti IBM se mohou bez upozornění změnit nebo mohou být zrušena a reprezentují pouze cíle a plány společnosti.

Tyto údaje obsahují příklady dat a sestav používaných v běžných obchodních operacích. Aby byla představa úplná, používají se v příkladech jména osob a názvy společností, značek a produktů. Všechna tato jména a názvy jsou fiktivní a jejich podobnost se jmény, názvy a adresami používanými ve skutečnosti je zcela náhodná.

LICENČNÍ INFORMACE:

Tyto informace obsahují ukázkové aplikační programy ve zdrojovém jazyce ilustrující programovací techniky na různých operačních platformách. Tyto ukázkové programy můžete bez závazků vůči společnosti IBM jakýmkoli způsobem kopírovat, měnit a distribuovat za účelem vývoje, používání, odbytu či distribuce aplikačních programů odpovídajících rozhraní API pro operační platformu, pro kterou byly ukázkové programy napsány. Tyto příklady nebyly plně testovány za všech podmínek. Společnost IBM proto nemůže zaručit spolehlivost, upotřebitelnost nebo funkčnost těchto programů.

Při prohlížení těchto dokumentů v elektronické podobě se nemusí zobrazit všechny fotografie a barevné ilustrace.

Informace o programovacím rozhraní

Informace programátorských rozhraní, je-li poskytnuta, vám pomohou vytvořit aplikační software pro použití s tímto programem.

Tato příručka obsahuje informace o zamýšlených programovacích rozhraních, které umožňují zákazníkům psát programy za účelem získání služeb produktu WebSphere MQ.

Tyto informace však mohou obsahovat i diagnostické údaje a informace o úpravách a ladění. Informace o diagnostice, úpravách a vyladění jsou poskytovány jako podpora ladění softwarových aplikací.

Důležité: Nepoužívejte tyto informace o diagnostice, úpravách a ladění jako programátorské rozhraní, protože se mohou měnit.

Ochranné známky

IBM, logo IBM, ibm.com jsou ochranné známky společnosti IBM Corporation, registrované v mnoha jurisdikcích po celém světě. Aktuální seznam ochranných známek IBM je k dispozici na webu na stránce "Copyright and trademark information" www.ibm.com/legal/copytrade.shtml. Ostatní názvy produktů a služeb mohou být ochrannými známkami společnosti IBM nebo jiných společností.

Microsoft a Windows jsou ochranné známky společnosti Microsoft Corporation ve Spojených státech a případně v dalších jiných zemích.

UNIX je registrovaná ochranná známka skupiny The Open Group ve Spojených státech a případně v dalších jiných zemích.

Linux[®] je registrovaná ochranná známka Linuse Torvaldse ve Spojených státech a případně v dalších jiných zemích.

Tento produkt obsahuje software vyvinutý v rámci projektu Eclipse Project (<http://www.eclipse.org/>).

Java a všechny ochranné známky a loga založené na termínu Java jsou ochranné známky nebo registrované ochranné známky společnosti Oracle anebo příbuzných společností.



Číslo položky:

(1P) P/N: