9.0

IBM Message Service Client for .NET

Uwaga

Przed skorzystaniem z niniejszych informacji oraz produktu, którego one dotyczą, należy zapoznać się z informacjami zamieszczonymi w sekcji “Uwagi” na stronie 253.

To wydanie dotyczy wersji 9 wydania 0 produktu IBM® MQ oraz wszystkich kolejnych wydań i modyfikacji, o ile nie zostanie to określone inaczej w nowych edycjach.

Wysyłając informacje do IBM, użytkownik przyznaje IBM niewyłączne prawo do używania i rozpowszechniania informacji w dowolny sposób, jaki uzna za właściwy, bez żadnych zobowiązań wobec ich autora.

Spis treści

Wprowadzenie do IBM Message Service Client for .NET

Produkt IBM Message Service Client for .NET udostępnia aplikacyjny interfejs programistyczny (API) o nazwie XMS , który zawiera ten sam zestaw interfejsów co Java Message Service (JMS) API. Produkt IBM Message Service Client for .NET zawiera w pełni zarządzaną implementację produktu XMS, która może być używana przez dowolny język zgodny z produktem .NET .

Produkt XMS obsługuje:

- punkt z punktem przesyłanie komunikatów
- Publikowanie/subskrypcja przesyłanie komunikatów
- Synchroniczne dostarczanie komunikatów
- Dostarczanie komunikatów asynchronicznych

Aplikacja XMS może wymieniać komunikaty z następującymi typami aplikacji:

- Aplikacja XMS
- Aplikacja IBM MQ classes for JMS
- Rodzima aplikacja produktu IBM MQ
- Aplikacja JMS, która używa domyślnego dostawcy przesyłania komunikatów produktu IBM MQ .

Aplikacja XMS może łączyć się z dowolnym z następujących serwerów przesyłania komunikatów i korzystać z nich:

IBM MQ menedżer kolejek
Aplikacja może łączyć się z powiązaniami lub trybem klienta.

WebSphere Application Server Usługa Integration Bus
Aplikacja może korzystać z bezpośredniego połączenia TCP/IP lub może używać protokołu HTTP przez TCP/IP.

IBM Integration Bus
Komunikaty są transportowane między aplikacją a brokerem przy użyciu produktu WebSphere MQ Real-time Transport. Komunikaty mogą być dostarczane do aplikacji za pomocą programu WebSphere MQ Multicast Transport.

W celu nawiązania połączenia z menedżerem kolejek produktu IBM MQ aplikacja XMS może używać produktu WebSphere MQ Enterprise Transport do komunikowania się z produktem IBM Integration Bus. Alternatywnie aplikacja XMS może publikować i subskrybować połączenie z produktem IBM MQ.

Pojęcia pokrewne

“Style przesyłania komunikatów” na stronie 6

“Model obiektu XMS” na stronie 7
Interfejs API produktu XMS jest interfejsem obiektowym. Model obiektu XMS jest oparty na modelu obiektu JMS 1.1 .

“Model komunikatów produktu XMS” na stronie 10
Model komunikatów produktu XMS jest taki sam, jak model komunikatu produktu IBM MQ classes for JMS .

Wprowadzenie do IBM Message Service Client for .NET

Produkt IBM Message Service Client for .NET udostępnia aplikacyjny interfejs programistyczny (API) o nazwie XMS , który zawiera ten sam zestaw interfejsów co Java Message Service (JMS) API. Produkt IBM Message Service Client for .NET zawiera w pełni zarządzaną implementację produktu XMS, która może być używana przez dowolny język zgodny z produktem .NET .

Produkt XMS obsługuje:

- punkt z punktem przesyłanie komunikatów

- Publikowanie/subskrypcja przesyłanie komunikatów
- Synchroniczne dostarczanie komunikatów
- Dostarczanie komunikatów asynchronicznych

Aplikacja XMS może wymieniać komunikaty z następującymi typami aplikacji:

- Aplikacja XMS
- Aplikacja IBM MQ classes for JMS
- Rodzima aplikacja produktu IBM MQ
- Aplikacja JMS, która używa domyślnego dostawcy przesyłania komunikatów produktu IBM MQ .

Aplikacja XMS może łączyć się z dowolnym z następujących serwerów przesyłania komunikatów i korzystać z nich:

IBM MQ menedżer kolejek
Aplikacja może łączyć się z powiązaniami lub trybem klienta.

WebSphere Application Server Usługa Integration Bus
Aplikacja może korzystać z bezpośredniego połączenia TCP/IP lub może używać protokołu HTTP przez TCP/IP.

IBM Integration Bus
Komunikaty są transportowane między aplikacją a brokerem przy użyciu produktu WebSphere MQ Real-time Transport. Komunikaty mogą być dostarczane do aplikacji za pomocą programu WebSphere MQ Multicast Transport.

W celu nawiązania połączenia z menedżerem kolejek produktu IBM MQ aplikacja XMS może używać produktu WebSphere MQ Enterprise Transport do komunikowania się z produktem IBM Integration Bus. Alternatywnie aplikacja XMS może publikować i subskrybować połączenie z produktem IBM MQ.

Pojęcia pokrewne

“Style przesyłania komunikatów” na stronie 6

“Model obiektu XMS” na stronie 7
Interfejs API produktu XMS jest interfejsem obiektowym. Model obiektu XMS jest oparty na modelu obiektu JMS 1.1 .

“Model komunikatów produktu XMS” na stronie 10
Model komunikatów produktu XMS jest taki sam, jak model komunikatu produktu IBM MQ classes for JMS .

Style przesyłania komunikatów

Produkt XMS obsługuje style przesyłania komunikatów w produkcie punkt z punktem i Publikowanie/subskrypcja .

Style przesyłania komunikatów są również nazywane domenami przesyłania komunikatów.

Punkt-punkt przesyłanie komunikatów

Wspólna forma przesyłania komunikatów produktu punkt z punktem używa kolejkowania. W najprostszym przypadku aplikacja wysyła komunikat do innej aplikacji, identyfikując, niejawnie lub jawnie, kolejkę docelową. Bazowy system przesyłania komunikatów i kolejkowania odbiera komunikat od aplikacji wysyłającej i kieruje do niej komunikat do kolejki docelowej. Aplikacja odbierający może następnie pobrać komunikat z kolejki.

Jeśli bazowy system przesyłania komunikatów i kolejkowania zawiera produkt IBM Integration Bus, produkt IBM Integration Bus może replikować komunikaty i kopie trasy komunikatu do różnych kolejek. W wyniku tego komunikat może otrzymać więcej niż jedna aplikacja. Produkt IBM Integration Bus może także transformować komunikat i dodawać do niego dane.

Kluczowa cecha funkcji przesyłania komunikatów produktu punkt z punktem polega na tym, że aplikacja umieszcza komunikat w kolejce lokalnej, gdy wysyła komunikat. Bazowy system przesyłania komunikatów

i kolejkowania określa kolejkę docelową, do której wysyłany jest komunikat. Aplikacja odbierający pobiera komunikat z kolejki docelowej.

Publikowanie/subskrypcja przesyłanie komunikatów

W przesyłaniu komunikatów produktu Publikowanie/subskrypcja dostępne są dwa typy aplikacji: publikator i subskrybent.

Wydawca dostarcza informacje w postaci komunikatów publikacji. Gdy publikator publikuje komunikat, określa on temat, który identyfikuje temat informacji wewnątrz komunikatu.

subskrybent jest konsumentem publikowanych informacji. Subskrybent określa tematy, którymi interesuje się tworzenie subskrypcji.

System publikowania/subskrypcji otrzymuje od subskrybentów publikacje z wydawców i subskrypcje. Kieruje publikacjami do subskrybentów. Subskrybent otrzymuje publikacje dotyczące tylko tych tematów, do których zasubskrybował.

Kluczowa cecha mechanizmu przesyłania komunikatów produktu Publikowanie/subskrypcja polega na tym, że publikator identyfikuje temat, gdy publikuje komunikat. Nie identyfikuje on subskrybentów. Jeśli komunikat jest publikowany w temacie, dla którego nie ma subskrybentów, to komunikat nie otrzymuje żadnej aplikacji.

Aplikacja może być zarówno publikatorem, jak i subskrybentem.

Model obiektu XMS

Interfejs API produktu XMS jest interfejsem obiektowym. Model obiektu XMS jest oparty na modelu obiektu JMS 1.1 .

Poniższa lista zawiera podsumowanie głównych klas produktu XMS lub typów obiektów:

`ConnectionFactory`
Obiekt `ConnectionFactory` hermetyzuje zestaw parametrów dla połączenia. Aplikacja korzysta z `ConnectionFactory` w celu utworzenia połączenia. Aplikacja może udostępniać parametry w czasie wykonywania i tworzyć obiekt `ConnectionFactory` . Alternatywnie parametry połączenia mogą być przechowywane w repozytorium administrowanych obiektów. Aplikacja może pobrać obiekt z repozytorium i utworzyć z niego obiekt `ConnectionFactory` .

`Connection`
Obiekt `Connection` hermetyzuje aktywne połączenie z aplikacji na serwer przesyłania komunikatów. Aplikacja korzysta z połączenia w celu utworzenia sesji.

`Destination`
Aplikacja wysyła komunikaty lub odbiera komunikaty przy użyciu obiektu `Destination` . W domenie Publikowanie/subskrypcja obiekt `Destination` hermetykuje temat, a w domenie punkt z punktem obiekt `Destination` hermetykuje kolejkę. Aplikacja może udostępnić parametry w celu utworzenia obiektu `Destination` w czasie wykonywania. Alternatywnie może on utworzyć obiekt `Destination` na podstawie definicji obiektu przechowywanej w repozytorium obiektów administrowanych.

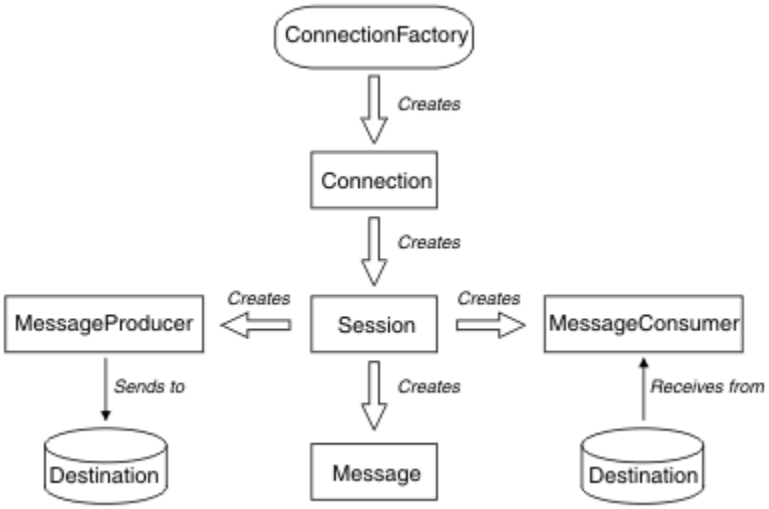
`Session`
Obiekt `Session` jest jednowątkowym kontekstem do wysyłania i odbierania komunikatów. Aplikacja korzysta z obiektu `Session` do tworzenia obiektów `Message`, `MessageProducer`i `MessageConsumer` .

`Message`
Obiekt `Message` hermetyzuje obiekt `Message` , który aplikacja wysyła za pomocą obiektu `MessageProducer` , lub otrzymuje za pomocą obiektu `MessageConsumer` .

`MessageProducer`
Obiekt `MessageProducer` jest używany przez aplikację do wysyłania komunikatów do miejsca docelowego.

`MessageConsumer`
Obiekt `MessageConsumer` jest używany przez aplikację do odbierania komunikatów wysyłanych do miejsca docelowego.

Rysunek 1 na stronie 8 przedstawia te obiekty i ich relacje. Na tym diagramie przedstawiono główne typy obiektów XMS : ConnectionFactory, Connection, Session, MessageProducer, MessageConsumer, Message i Destination. Aplikacja korzysta z fabryki połączeń w celu utworzenia połączenia i korzysta z połączenia w celu utworzenia sesji. Aplikacja może następnie użyć sesji w celu utworzenia komunikatów, producentów komunikatów i konsumentów komunikatów. Aplikacja używa producenta komunikatów do wysyłania komunikatów do miejsca docelowego i używa konsumenta komunikatów do odbierania komunikatów wysyłanych do miejsca docelowego.



Rysunek 1. Obiekty XMS i ich relacje

W programie .NETklasy XMS są definiowane jako zestaw interfejsów produktu .NET . Podczas kodowania aplikacji XMS .NET wymagane są tylko zadeklarowane interfejsy.

Model obiektów XMS jest oparty na interfejsach niezależnych od domeny, które zostały opisane w specyfikacji Java Message Service , wersja 1.1. Klasy specyficzne dla domeny, takie jak `Topic`, `TopicPublisher`i `TopicSubscriber`, nie są udostępniane.

Atrybuty i właściwości obiektów

Obiekt XMS może mieć atrybuty i właściwości, które są właściwościami obiektu, które są implementowane na różne sposoby.

Atrybuty
Cecha charakterystyczna, która jest zawsze obecna i zajmuje pamięć masową, nawet jeśli atrybut nie ma wartości. W tym zakresie atrybut jest podobny do pola w strukturze danych o stałej długości. Cechą wyróżniającą atrybuty jest to, że każdy atrybut ma własne metody ustawiania i pobierania wartości.

Właściwości
Właściwość obiektu jest prezentowana i zajmuje pamięć masową dopiero po ustawieniu jej wartości. Nie można usunąć właściwości lub jej pamięć odzyskana po ustawieniu jej wartości. Można zmienić jego wartość. Produkt XMS udostępnia zestaw metod ogólnych do ustawiania i pobierania wartości właściwości.

Pojęcia pokrewne

Typy podstawowe produktu XMS
Produkt XMS udostępnia odpowiedniki ośmiu typów podstawowych Java (byte, short, int, long, float, double, char i boolean). Pozwala to na wymianę komunikatów między XMS a JMS bez utraty lub uszkodzenia danych.

Niejawne przekształcenie wartości właściwości z jednego typu danych na inny.
Gdy aplikacja pobiera wartość właściwości, wartość może zostać przekształcona przez produkt XMS na inny typ danych. Wiele reguł decyduje o tym, które konwersje są obsługiwane i w jaki sposób program XMS wykonuje konwersje.

Odsyłacze pokrewne

Typy danych dla elementów danych aplikacji
Aby aplikacja XMS mogła wymieniać komunikaty z aplikacją IBM MQ classes for JMS , zarówno aplikacje, jak i aplikacje muszą być w stanie interpretować dane aplikacji w treści komunikatu w ten sam sposób.

Administrowane obiekty

Za pomocą administrowanych obiektów można administrować ustawieniami połączenia używalnymi przez aplikacje klienckie, które mają być administrowane z centralnego repozytorium. Aplikacja pobiera definicje obiektów z centralnego repozytorium i używa ich do tworzenia obiektów `ConnectionFactory` i `Destination` . Za pomocą administrowanych obiektów można usunąć kilka aplikacji z zasobów, które są używane w czasie wykonywania.

Na przykład aplikacje produktu XMS mogą być zapisywane i testowane przy użyciu obiektów administrowanych, które odwołują się do zestawu połączeń i miejsc docelowych w środowisku testowym. Po wdrożeniu aplikacji administrowane obiekty można zmienić w celu skonfigurowania aplikacji tak, aby odwoływały się do połączeń i miejsc docelowych w środowisku produkcyjnym.

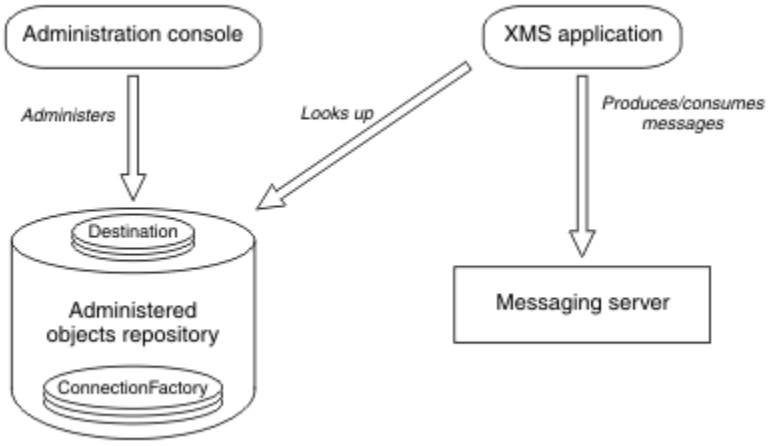
Produkt XMS obsługuje dwa typy administrowanych obiektów:

- Obiekt `ConnectionFactory` , który jest używany przez aplikacje do nawiązania początkowego połączenia z serwerem.
- Obiekt `Destination` , który jest używany przez aplikacje do określania miejsca docelowego dla wysyłanych komunikatów oraz do źródła komunikatów odbieranych. Miejsce docelowe jest tematem lub kolejką na serwerze, z którym łączy się aplikacja.

Narzędzie administracyjne **`JMSAdmin`** jest dostarczane z produktem IBM MQ. Jest on używany do tworzenia obiektów administrowanych i zarządzania nimi w centralnym repozytorium obiektów administrowanych.

Administrowane obiekty w repozytorium mogą być używane przez aplikacje IBM MQ classes for JMS i XMS . Aplikacje produktu XMS mogą używać obiektów `ConnectionFactory` i `Destination` do łączenia się z menedżerem kolejek IBM MQ. Administrator może zmienić definicje obiektów przechowywane w repozytorium bez wpływu na kod aplikacji.

Na poniższym diagramie przedstawiono, w jaki sposób aplikacja XMS zwykle używa administrowanych obiektów. Lewa strona diagramu zawiera repozytorium zawierające definicje obiektów ConnectionFactory i obiektów docelowych, które są administrowane przy użyciu konsoli administracyjnej. Po prawej stronie diagramu wyświetlana jest aplikacja XMS , która wyszukuje definicje obiektów w repozytorium, a następnie korzysta z tych definicji obiektów podczas nawiązywania połączenia z serwerem przesyłania komunikatów.



Rysunek 2. Typowe zastosowanie administrowanych obiektów przez aplikację XMS

Pojęcia pokrewne

Praca z administrowanymi obiektami
Tematy w tej sekcji zawierają informacje na temat administrowanych obiektów. Aplikacje produktu XMS mogą pobierać definicje obiektów z repozytorium administrowanych obiektów centralnych i używać ich do tworzenia fabryk połączeń i miejsc docelowych.

Obsługiwane typy administrowanego repozytorium obiektów
Obiekty administrowane przez system plików i LDAP mogą być używane do łączenia się z IBM MQ i WebSphere Application Server, podczas gdy nazwy COS mogą być używane tylko do łączenia się z serwerem WebSphere Application Server.

Zadania pokrewne

Tworzenie obiektów administrowanych
Definicje obiektów ConnectionFactory i Destination, które są wymagane przez aplikacje produktu XMS w celu nawiązania połączenia z serwerem przesyłania komunikatów, muszą zostać utworzone przy użyciu odpowiednich narzędzi administracyjnych.

Model komunikatów produktu XMS

Model komunikatów produktu XMS jest taki sam, jak model komunikatu produktu IBM MQ classes for JMS .

W szczególności produkt XMS implementuje te same pola nagłówka komunikatu i właściwości komunikatu, które są implementowane przez produkt IBM MQ classes for JMS :

- Pola nagłówka JMS . Te pola mają nazwy rozpoczynający się przedrostkiem JMS.
- JMS zdefiniowane właściwości. Te pola mają właściwości, których nazwy są rozpoczynanie od przedrostka JMSX.
- IBM zdefiniowane właściwości. Te pola mają właściwości, których nazwy są rozpoczynanie z przedrostkiem JMS_IBM_.

W rezultacie aplikacje produktu XMS mogą wymieniać komunikaty z aplikacjami produktu IBM MQ classes for JMS . W każdym komunikacie niektóre pola nagłówka i właściwości są ustawiane przez aplikację, a inne są ustawiane przez produkt XMS lub IBM MQ classes for JMS. Niektóre pola ustawione przez XMS lub IBM MQ classes for JMS są ustawiane podczas wysyłania komunikatu, a inne po jego odebraniu. Pola nagłówka i właściwości są propagowane z komunikatem za pośrednictwem serwera przesyłania komunikatów, gdzie jest to właściwe. Są one dostępne dla każdej aplikacji, która odbiera komunikat.

Instalowanie produktu Message Service Client for .NET przy użyciu kreatora instalacji

Instalacja korzysta z instalatora MSI InstallShield X/Windows. Dostępne są dwie opcje konfiguracji, dzięki czemu można wybrać instalację kompletną lub niestandardową.

O tym zadaniu

Aby zainstalować produkt Message Service Client for .NET w systemie Windows, należy wykonać następującą procedurę.

Procedura

1. Jeśli instalacja jest wykonywana z pakietu serwisowego SupportPac , wykonaj następujące kroki, w przeciwnym razie przejdź bezpośrednio do kroku “2” na stronie 10.
   a) W systemie Windowszaloguj się jako administrator.
   b) Uruchom instalator dotNETClientsetup.exe.
2. Poczekaj na otwarcie kreatora instalacji i wyświetlenie następującego komunikatu:

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

Kliknij przycisk **Dalej**.

Kreator może poprosić użytkownika o zapoznanie się z umową licencyjną.

3. Jeśli użytkownik jest proszony o zapoznanie się z umową licencyjną i zaakceptowanie warunków umowy licencyjnej, kliknij opcję **Akceptuję warunki umowy licencyjnej**, a następnie kliknij przycisk **Dalej**.

   Kreator instalacji poprosi o wybranie typu konfiguracji, który najlepiej odpowiada potrzebom użytkownika.

4. Wybierz wymagany typ konfiguracji:

   - Aby zainstalować wszystkie składniki programu, a następnie zainstalować je w domyślnym katalogu instalacyjnym, kliknij przycisk **Zakończone**.
   - Aby wybrać opcje, które mają zostać zainstalowane, i określić miejsce, w którym są zainstalowane, kliknij opcję **Niestandardowe**.

5. Kliknij przycisk **Dalej**.

   Jeśli zostanie wybrana opcja zakończenia instalacji, kreator instalacji wyświetli komunikat, że jest on gotowy do rozpoczęcia instalacji zgodnie z opisem w kroku “8” na stronie 11. Jeśli zostanie wybrana opcja instalacji niestandardowej, kreator instalacji poprosi o wybranie składników, które mają zostać zainstalowane, a następnie należy wykonać krok “6” na stronie 11 i krok “7” na stronie 11 przed przeniesieniem do kroku “8” na stronie 11.

6. W przypadku instalacji niestandardowej należy kliknąć ikonę na liście składników, aby określić zmiany dotyczące sposobu instalowania składników produktu Message Service Client for .NET . Jeśli produkt Message Service Client for .NET nie ma być instalowany w katalogu sugerowanym, należy wybrać inny katalog.

   Jeśli produkt Message Service Client for .NET zostanie zainstalowany w katalogu, który aktualnie nie istnieje, kreator instalacji utworzy ten katalog.

   Jeśli planowane jest tworzenie aplikacji produktu XMS , należy upewnić się, że wybrano opcję **Narzędzia programistyczne i przykłady** . Ten składnik udostępnia przykładowe aplikacje oraz biblioteki i wszystkie inne pliki wymagane do skompilowania aplikacji produktu .NET . Jeśli ta opcja nie zostanie wybrana, zostaną zainstalowane tylko pliki wymagane do uruchomienia aplikacji produktu XMS .

7. Jeśli używana jest opcja instalacji niestandardowej, kliknij przycisk **Dalej** po wybraniu opcji, które są wymagane zgodnie z opisem w kroku “6” na stronie 11.

   Kreator instalacji wyświetli komunikat, że jest on gotowy do rozpoczęcia instalacji.

8. Kliknij przycisk **Instaluj** , aby rozpocząć instalację.

   Kreator instalacji wyświetli pasek pokazujący postęp instalacji. Poczekaj, aż pasek postępu zostanie zakończony. Po pomyślnym zakończeniu instalacji w oknie zostanie wyświetlony następujący komunikat:

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

9. Kliknij przycisk **Zakończ**, aby zamknąć kreatora instalacji.

Wyniki

Produkt Message Service Client for .NETzostał pomyślnie zainstalowany, co jest gotowe do użycia.

Co dalej

Przed uruchomieniem jakichkolwiek aplikacji produktu XMS , w tym przykładowych aplikacji udostępnionych w produkcie XMS, należy skonfigurować środowisko serwera przesyłania komunikatów,

aby uzyskać szczegółowe informacje na ten temat: “Konfigurowanie środowiska serwera przesyłania komunikatów” na stronie 13.

Pojęcia pokrewne

Usługa Web Service wyszukiwania JNDI
Aby uzyskać dostęp do katalogu nazw COS z poziomu produktu XMS, usługa Web Service wyszukiwania produktu JNDI musi zostać wdrożona na serwerze WebSphere Application Server service integration bus . Ta usługa Web Service tłumaczy informacje o produkcie Java z usługi nazw COS na formularz, który może być odczytany przez aplikacje produktu XMS .

Konfigurowanie środowiska serwera przesyłania komunikatów
Tematy w tej sekcji opisują sposób konfigurowania środowiska serwera przesyłania komunikatów w celu umożliwienia aplikacjom produktu XMS nawiązywania połączenia z serwerem.

Korzystanie z przykładowych aplikacji produktu XMS
Użyj przykładowych aplikacji dostarczanych razem z produktem XMS w celu zweryfikowania instalacji i konfiguracji serwera przesyłania komunikatów, a także w celu ułatwienia budowania własnych aplikacji. Przykłady udostępniają przegląd wspólnych funkcji każdego interfejsu API.

Wymagania wstępne dla aplikacji XMS łączących się z produktem WebSphere MQ

Niektóre wymagania wstępne mają zastosowanie, jeśli aplikacja XMS łączy się z produktem WebSphere MQ.

W przypadku aplikacji, które łączą się z menedżerem kolejek produktu WebSphere MQ , należy zainstalować odpowiednie biblioteki klienta produktu WebSphere MQ na komputerze, który jest używany do uruchamiania aplikacji XMS . Te biblioteki są wstępnie zainstalowane na komputerach z lokalnym menedżerem kolejek.

W przypadku systemu KlientXMS dla platformy .NETnależy użyć bibliotek klienta dostarczanych z produktem IBM WebSphere MQ 7.0.1.0 lub nowszym. Są to klasy *WebSphere MQ dla produktu .NET*. Umożliwiają one nawiązanie połączeń w trybie klienta z menedżerami kolejek produktu IBM WebSphere MQ 7.0, 6.0i 5.3 oraz połączeniami w trybie powiązań z lokalnym menedżerem kolejek, jeśli jest on także IBM WebSphere MQ 7.0.1.0 lub nowszy.

Pakiet Microsoft .NET Framework 2.0 Redistributable Package musi być zainstalowany na komputerze, na którym ma zostać zainstalowany produkt XMS . Jeśli ten pakiet nie jest dostępny, instalacja produktu XMS nie powiedzie się. Następnie należy zakończyć procedurę instalacji, zainstalować na komputerze pakiet Microsoft .NET Framework 2.0 Redistributable Package, a następnie ponownie uruchomić procedurę instalacji.

W serwisie pobierania firmy Microsoft należy wyszukać plik dotnetfx.exe w przypadku pakietu Microsoft .NET Framework 2.0 Redistributable Package (x86) i NetFx64.exe dla pakietu Microsoft .NET Framework 2.0 Redistributable Package (x64), w zależności od tego, która z tych wartości jest odpowiednia.

Pojęcia pokrewne

Konfigurowanie środowiska serwera przesyłania komunikatów
Tematy w tej sekcji opisują sposób konfigurowania środowiska serwera przesyłania komunikatów w celu umożliwienia aplikacjom produktu XMS nawiązywania połączenia z serwerem.

Blokowanie aplikacji przy użyciu nowszej wersji XMS

Domyślnie, gdy zainstalowana jest nowsza wersja produktu XMS , aplikacje korzystające z poprzedniej wersji automatycznie przełączają się na nowszą wersję bez konieczności ponownego kompilowania.

O tym zadaniu

Funkcja współistnienia wielu wersji zapewnia, że instalacja nowszej wersji produktu XMS nie spowoduje nadpisania poprzedniej wersji produktu XMS . Zamiast tego w globalnej pamięci podręcznej zespołu

(Global Assembly Cache-GAC) współistnieją wiele instancji podobnych zespołów XMS .NET , ale mają one różne numery wersji. Wewnętrznie, GAC używa pliku strategii do kierowania wywołań aplikacji do najnowszej wersji produktu XMS. Aplikacje działają bez konieczności ponownego kompilowania i mogą korzystać z nowych funkcji dostępnych w nowszej wersji produktu XMS .NET .

Jeśli jednak aplikacja jest wymagana do korzystania ze starszej wersji produktu XMS , należy ustawić atrybut `publisherpolicy` na wartość no w pliku konfiguracyjnym aplikacji.

Uwaga: Plik konfiguracyjny aplikacji to plik o nazwie, która składa się z nazwy programu wykonywalnego, do którego odnosi się plik, z przyrostkiem .config. Na przykład plik konfiguracyjny aplikacji dla pliku text.exe będzie miał nazwę text.exe.config.

Jednak w dowolnym momencie wszystkie aplikacje systemu korzystają z tej samej wersji produktu XMS .NET.

Konfigurowanie środowiska serwera przesyłania komunikatów

Tematy w tej sekcji opisują sposób konfigurowania środowiska serwera przesyłania komunikatów w celu umożliwienia aplikacjom produktu XMS nawiązywania połączenia z serwerem.

W przypadku aplikacji, które łączą się z menedżerem kolejek produktu IBM MQ , wymagany jest klient IBM MQ (lub menedżer kolejek w trybie powiązań).

Obecnie nie ma wymagań wstępnych dla aplikacji, które korzystają z połączenia w czasie rzeczywistym z brokerem.

Środowisko serwera przesyłania komunikatów musi zostać skonfigurowane przed uruchomieniem dowolnej aplikacji produktu XMS , w tym przykładowych aplikacji udostępnionych z produktem XMS.

Sekcja obejmuje następujące tematy:

- “Konfigurowanie menedżera kolejek i brokera dla aplikacji, która łączy się z menedżerem kolejek produktu IBM MQ” na stronie 14
- “Konfigurowanie brokera dla aplikacji, która korzysta z połączenia w czasie rzeczywistym z brokerem” na stronie 15
- “Konfigurowanie magistrali integracji usług dla aplikacji, która łączy się z produktem WebSphere Application Server” na stronie 16

Pojęcia pokrewne

Usługa Web Service wyszukiwania JNDI
Aby uzyskać dostęp do katalogu nazw COS z poziomu produktu XMS, usługa Web Service wyszukiwania produktu JNDI musi zostać wdrożona na serwerze WebSphere Application Server service integration bus . Ta usługa Web Service tłumaczy informacje o produkcie Java z usługi nazw COS na formularz, który może być odczytany przez aplikacje produktu XMS .

Korzystanie z przykładowych aplikacji produktu XMS
Użyj przykładowych aplikacji dostarczanych razem z produktem XMS w celu zweryfikowania instalacji i konfiguracji serwera przesyłania komunikatów, a także w celu ułatwienia budowania własnych aplikacji. Przykłady udostępniają przegląd wspólnych funkcji każdego interfejsu API.

Zadania pokrewne

Instalowanie produktu Message Service Client for .NET przy użyciu kreatora instalacji
Instalacja korzysta z instalatora MSI InstallShield X/Windows. Dostępne są dwie opcje konfiguracji, dzięki czemu można wybrać instalację kompletną lub niestandardową.

Odsyłacze pokrewne

Wymagania wstępne dla aplikacji XMS łączących się z produktem WebSphere MQ
Niektóre wymagania wstępne mają zastosowanie, jeśli aplikacja XMS łączy się z produktem WebSphere MQ.

Konfigurowanie menedżera kolejek i brokera dla aplikacji, która łączy się z menedżerem kolejek produktu IBM MQ

W tej sekcji założono, że użytkownik korzysta z produktu IBM WebSphere MQ 7.0.1lub jego nowszej wersji. Zanim możliwe będzie uruchomienie aplikacji łączącej się z menedżerem kolejek produktu IBM MQ , należy skonfigurować menedżer kolejek. W przypadku aplikacji publikowania/subskrypcji dodatkowa konfiguracja jest wymagana, jeśli używany jest interfejs publikowania/subskrybowania w kolejce.

Zanim rozpoczniesz

Produkt XMS działa z produktem IBM Integration Bus lub WebSphere Message Broker 6.1 lub nowszym.

Przed rozpoczęciem tej czynności należy wykonać następujące kroki:

- Upewnij się, że aplikacja ma dostęp do menedżera kolejek, który jest uruchomiony.
- Jeśli aplikacja jest aplikacją publikowania/subskrypcji i korzysta z interfejsu publikowania/subskrybowania w kolejce, należy upewnić się, że atrybut "PSMODE" jest ustawiony na wartość "ENABLED" w menedżerze kolejek.
- Upewnij się, że aplikacja korzysta z fabryki połączeń, której właściwości są odpowiednio ustawione w celu nawiązania połączenia z menedżerem kolejek. Jeśli aplikacja jest aplikacją publikowania/subskrypcji, należy upewnić się, że odpowiednie właściwości fabryki połączeń są ustawione na potrzeby używania brokera. For more information about the properties of a connection factory, “Właściwości obiektu ConnectionFactory” na stronie 182.

O tym zadaniu

Menedżer kolejek i broker można skonfigurować tak, aby uruchamiał aplikacje produktu XMS w taki sam sposób, jak menedżer kolejek i umieszczony w kolejce interfejs publikowania/subskrypcji w celu uruchamiania aplikacji IBM MQ JMS . Poniższe kroki podsumowują to, co należy zrobić.

Procedura

1. W menedżerze kolejek utwórz kolejki, których potrzebuje aplikacja.

   Informacje na temat tworzenia kolejek zawiera sekcja Definiowanie kolejek.

   Jeśli aplikacja jest aplikacją publikowania/subskrypcji i korzysta z interfejsu kolejkowania publikowania/subskrybowania, który wymaga dostępu do kolejek systemowych IBM MQ classes for JMS , przed utworzeniem kolejek poczekaj na krok 4a .
2. Nadaj identyfikatorowi użytkownika powiązanom z aplikacją uprawnienia do łączenia się z menedżerem kolejek, a także odpowiednie uprawnienia dostępu do kolejek.

   Więcej informacji na temat autoryzacji zawiera sekcja Zabezpieczanie. Jeśli aplikacja łączy się z menedżerem kolejek w trybie klienta, patrz także Klienty i serwery.
3. Jeśli aplikacja łączy się z menedżerem kolejek w trybie klienta, należy upewnić się, że kanał połączenia z serwerem jest zdefiniowany w menedżerze kolejek i że obiekt nasłuchiwania został uruchomiony.

   Nie ma potrzeby wykonywania tego kroku dla każdej aplikacji, która łączy się z menedżerem kolejek. Jedna definicja kanału połączenia z serwerem i jeden program nasłuchujący mogą obsługiwać wszystkie aplikacje, które łączą się w trybie klienta.
4. Jeśli aplikacja jest aplikacją publikowania/subskrypcji i korzysta z interfejsu publikowania/subskrybowania w kolejce, wykonaj następujące kroki.

   a) W menedżerze kolejek utwórz kolejki systemowe produktu IBM MQ classes for JMS , uruchamiając skrypt komend MQSC, które są dostarczane z produktem IBM MQ. Upewnij się, że ID użytkownika powiązany z IBM Integration Bus lub WebSphere Message Broker ma uprawnienia do uzyskiwania dostępu do kolejek.

   Więcej informacji na temat miejsca, w którym można znaleźć skrypt i sposób jego uruchomienia, zawiera sekcja Korzystanie z IBM MQ classes for Java.

Ten krok należy wykonać tylko raz dla menedżera kolejek. Ten sam zestaw kolejek systemowych IBM MQ classes for JMS może obsługiwać wszystkie aplikacje XMS i IBM MQ classes for JMS , które łączą się z menedżerem kolejek.

b) Nadaj użytkownikowi identyfikator powiązany z aplikacją, aby uzyskać dostęp do kolejek systemowych produktu IBM MQ classes for JMS .

   Informacje o tym, jakie uprawnienia są wymagane przez identyfikator użytkownika, zawiera sekcja Korzystanie z IBM MQ classes for JMS.

c) W przypadku brokera produktu IBM Integration Bus lub WebSphere Message Brokernależy utworzyć i wdrożyć przepływ komunikatów w celu obsługi kolejki, w której aplikacje wysyłają komunikaty, które publikują.

   Podstawowy przepływ komunikatów składa się z węzła przetwarzania komunikatów MQInput w celu odczytania opublikowanych komunikatów i węzła przetwarzania komunikatów publikowania w celu opublikowania komunikatów.

   Informacje na temat sposobu tworzenia i wdrażania przepływu komunikatów można znaleźć w dokumentacji produktu IBM Integration Bus lub WebSphere Message Broker dostępnej w Strona WWW biblioteki dokumentacji produktu IBM Integration Bus.

   Nie ma potrzeby wykonywania tego kroku, jeśli odpowiedni przepływ komunikatów jest już wdrożony w brokerze.

Wyniki

Teraz można uruchomić aplikację.

Zadania pokrewne

Konfigurowanie brokera dla aplikacji, która korzysta z połączenia w czasie rzeczywistym z brokerem
Przed uruchomieniem aplikacji, która korzysta z połączenia w czasie rzeczywistym z brokerem, należy skonfigurować ten broker.

Konfigurowanie magistrali integracji usług dla aplikacji, która łączy się z produktem WebSphere Application Server
Przed uruchomieniem aplikacji, która łączy się z magistralą integracji usług produktu WebSphere Application Server service integration technologies , należy skonfigurować integrację usług w taki sam sposób, w jaki magistrala integracji usług jest skonfigurowana do uruchamiania aplikacji produktu JMS , które korzystają z domyślnego dostawcy przesyłania komunikatów.

Konfigurowanie brokera dla aplikacji, która korzysta z połączenia w czasie rzeczywistym z brokerem

Przed uruchomieniem aplikacji, która korzysta z połączenia w czasie rzeczywistym z brokerem, należy skonfigurować ten broker.

Zanim rozpoczniesz

Przed rozpoczęciem tej czynności należy wykonać następujące kroki:

- Upewnij się, że aplikacja ma dostęp do brokera, który jest uruchomiony.
- Upewnij się, że aplikacja korzysta z fabryki połączeń, której właściwości są odpowiednio ustawione na potrzeby połączenia w czasie rzeczywistym z brokerem. Więcej informacji na temat właściwości fabryki połączeń zawiera sekcja “Właściwości obiektu ConnectionFactory” na stronie 182.

O tym zadaniu

Istnieje możliwość skonfigurowania brokera do uruchamiania aplikacji produktu XMS w taki sam sposób, jak w przypadku konfigurowania brokera do uruchamiania aplikacji produktu IBM MQ classes for JMS . Poniższe kroki podsumowują to, co należy wykonać:

Procedura

1. Utwórz i wdróż przepływ komunikatów w celu odczytania komunikatów z portu TCP/IP, na którym broker nasłuchuje i opublikuje komunikaty.

   Można to zrobić w jeden z następujących sposobów:

   - Utwórz przepływ komunikatów, który zawiera węzeł przetwarzania komunikatów produktu **Real-timeOptimizedFlow** .
   - Utwórz przepływ komunikatów, który zawiera węzeł przetwarzania komunikatów produktu **Real-timeInput** i węzeł przetwarzania komunikatów publikacji.

   Należy skonfigurować węzeł **Real-timeOptimizedFlow** lub **Real-timeInput** , aby nasłuchiwać na porcie używanym do połączeń w czasie rzeczywistym. W programie XMSdomyślnym numerem portu dla połączeń w czasie rzeczywistym jest 1506.

   Nie ma potrzeby wykonywania tego kroku, jeśli odpowiedni przepływ komunikatów jest już wdrożony w brokerze.
2. Jeśli wymagane są komunikaty, które mają zostać dostarczone do aplikacji przy użyciu produktu IBM MQ classes for JMS, należy skonfigurować broker, aby włączyć rozsyłanie grupowe. Należy skonfigurować tematy, które muszą mieć włączone rozsyłanie grupowe, określając niezawodną jakość usługi dla tych tematów, które wymagają niezawodnego rozsyłania grupowego.
3. Jeśli aplikacja udostępnia identyfikator użytkownika i hasło podczas nawiązywania połączenia z brokerem, a użytkownik chce, aby broker uwierzytelnia aplikację przy użyciu tych informacji, należy skonfigurować serwer nazw użytkowników i broker na potrzeby prostego uwierzytelniania przy użyciu hasła typu telnet.

Wyniki

Teraz można uruchomić aplikację.

Zadania pokrewne

Konfigurowanie menedżera kolejek i brokera dla aplikacji, która łączy się z menedżerem kolejek produktu IBM MQ

W tej sekcji założono, że użytkownik korzysta z produktu IBM WebSphere MQ 7.0.1lub jego nowszej wersji. Zanim możliwe będzie uruchomienie aplikacji łączącej się z menedżerem kolejek produktu IBM MQ , należy skonfigurować menedżer kolejek. W przypadku aplikacji publikowania/subskrypcji dodatkowa konfiguracja jest wymagana, jeśli używany jest interfejs publikowania/subskrybowania w kolejce.

Konfigurowanie magistrali integracji usług dla aplikacji, która łączy się z produktem WebSphere Application Server

Przed uruchomieniem aplikacji, która łączy się z magistralą integracji usług produktu WebSphere Application Server service integration technologies , należy skonfigurować integrację usług w taki sam sposób, w jaki magistrala integracji usług jest skonfigurowana do uruchamiania aplikacji produktu JMS , które korzystają z domyślnego dostawcy przesyłania komunikatów.

Konfigurowanie magistrali integracji usług dla aplikacji, która łączy się z produktem WebSphere Application Server

Przed uruchomieniem aplikacji, która łączy się z magistralą integracji usług produktu WebSphere Application Server service integration technologies , należy skonfigurować integrację usług w taki sam sposób, w jaki magistrala integracji usług jest skonfigurowana do uruchamiania aplikacji produktu JMS , które korzystają z domyślnego dostawcy przesyłania komunikatów.

Zanim rozpoczniesz

Przed rozpoczęciem tej czynności należy wykonać następujące kroki:

- Upewnij się, że została utworzona magistrala przesyłania komunikatów i że serwer jest dodawany do magistrali jako element magistrali.

- Upewnij się, że aplikacja ma dostęp do magistrali integracji usług, która zawiera co najmniej jeden działający mechanizm przesyłania komunikatów.
- Jeśli wymagane jest wykonanie operacji HTTP, należy zdefiniować kanał transportowy przychodzący mechanizmu przesyłania komunikatów HTTP. Domyślnie kanały dla protokołów SSL i TCP są definiowane podczas instalacji serwera.
- Upewnij się, że aplikacja korzysta z fabryki połączeń, której właściwości są odpowiednio ustawione w celu nawiązania połączenia z magistralą integracji usług przy użyciu serwera startowego. Minimalne wymagane informacje to:
  - Punkt końcowy dostawcy, który opisuje położenie i protokół, który ma być używany podczas negocjowania połączenia z serwerem przesyłania komunikatów (czyli za pośrednictwem serwera startowego). W najprostszej postaci, dla serwera zainstalowanego z ustawieniami domyślnymi, punkt końcowy udostępniania może być ustawiony na nazwę hosta serwera.
  - Nazwa magistrali, przez którą wysyłane są komunikaty.

  Więcej informacji na temat właściwości fabryki połączeń zawiera sekcja “Właściwości obiektu ConnectionFactory” na stronie 182.

O tym zadaniu

Należy zdefiniować wszystkie wymagane przez użytkownika obszary kolejki lub tematu. Domyślnie obszar tematu o nazwie Default.Topic.Space jest zdefiniowany podczas instalacji serwera, ale jeśli wymagane są dalsze obszary tematów, należy samodzielnie utworzyć te obszary tematów. Nie ma potrzeby wstępnego definiowania poszczególnych tematów w obrębie obszaru tematu, ponieważ serwer tworzy instancje tych pojedynczych tematów w sposób dynamiczny.

Poniższe kroki podsumowują to, co należy zrobić.

Procedura

1. Utwórz kolejki, które będą potrzebne aplikacji do przesyłania komunikatów w trybie punkt z punktem.
2. Utwórz dodatkowe obszary tematów, które będą potrzebne aplikacji do przesyłania komunikatów w trybie publikowania/subskrypcji.

Wyniki

Teraz można uruchomić aplikację.

Zadania pokrewne

Konfigurowanie menedżera kolejek i brokera dla aplikacji, która łączy się z menedżerem kolejek produktu IBM MQ

W tej sekcji założono, że użytkownik korzysta z produktu IBM WebSphere MQ 7.0.1lub jego nowszej wersji. Zanim możliwe będzie uruchomienie aplikacji łączącej się z menedżerem kolejek produktu IBM MQ , należy skonfigurować menedżer kolejek. W przypadku aplikacji publikowania/subskrypcji dodatkowa konfiguracja jest wymagana, jeśli używany jest interfejs publikowania/subskrybowania w kolejce.

Konfigurowanie brokera dla aplikacji, która korzysta z połączenia w czasie rzeczywistym z brokerem

Przed uruchomieniem aplikacji, która korzysta z połączenia w czasie rzeczywistym z brokerem, należy skonfigurować ten broker.

Korzystanie z przykładowych aplikacji produktu XMS

Użyj przykładowych aplikacji dostarczanych razem z produktem XMS w celu zweryfikowania instalacji i konfiguracji serwera przesyłania komunikatów, a także w celu ułatwienia budowania własnych aplikacji. Przykłady udostępniają przegląd wspólnych funkcji każdego interfejsu API.

Pojęcia pokrewne

Usługa Web Service wyszukiwania JNDI

Aby uzyskać dostęp do katalogu nazw COS z poziomu produktu XMS, usługa Web Service wyszukiwania produktu JNDI musi zostać wdrożona na serwerze WebSphere Application Server service integration bus .

Ta usługa Web Service tłumaczy informacje o produkcie Java z usługi nazw COS na formularz, który może być odczytany przez aplikacje produktu XMS .

Konfigurowanie środowiska serwera przesyłania komunikatów
Tematy w tej sekcji opisują sposób konfigurowania środowiska serwera przesyłania komunikatów w celu umożliwienia aplikacjom produktu XMS nawiązywania połączenia z serwerem.

Zadania pokrewne

Instalowanie produktu Message Service Client for .NET przy użyciu kreatora instalacji
Instalacja korzysta z instalatora MSI InstallShield X/Windows. Dostępne są dwie opcje konfiguracji, dzięki czemu można wybrać instalację kompletną lub niestandardową.

Przykładowe aplikacje

Przykładowe aplikacje zawierają przegląd wspólnych funkcji każdego interfejsu API. Można ich używać do weryfikowania konfiguracji instalacji i serwera przesyłania komunikatów oraz do budowania własnych aplikacji.

Jeśli potrzebna jest pomoc w tworzeniu własnych aplikacji, można użyć przykładowych aplikacji jako punktu początkowego. Zarówno źródło, jak i skompilowana wersja są udostępniane dla każdej aplikacji. Zapoznaj się z przykładowym kodem źródłowym i zidentyfikuj kroki kluczowe, aby utworzyć każdy wymagany obiekt dla aplikacji (ConnectionFactory, Connection, Session, Destination, Producer, Consumer lub both), a także aby ustawić wszystkie właściwości wymagane do określenia sposobu działania aplikacji. Więcej informacji na ten temat zawiera sekcja “Pisanie aplikacji produktu XMS” na stronie 21. Przykłady mogą ulec zmianie w przyszłych wersjach produktu XMS.

W poniższej tabeli przedstawiono trzy zestawy przykładowych aplikacji (po jednym dla każdego interfejsu API), które są dostarczane z produktem XMS.

Nazwa próbki	Opis
SampleConsumerCS	Aplikacja konsumująca komunikaty, która pobiera komunikaty z kolejki lub subskrybuje temat.
SampleProducerCS	Aplikacja producenta komunikatów, która generuje komunikaty do kolejki lub tematu.
SampleConfigCS	Aplikacja konfiguracyjna, której można użyć do utworzenia administrowanego repozytorium obiektów, które jest oparte na plikach. Aplikacja zawiera fabrykę połączeń i miejsce docelowe dla określonych ustawień połączenia. Administrowane repozytorium obiektów może być następnie używane razem z każdym przykładowym konsumentem i aplikacjami producenta.

Przykłady, które obsługują te same funkcje w różnych interfejsach API, mają różnice syntaktyczne.

- Przykładowe aplikacje konsumenta i producenta komunikatów obsługują następujące funkcje:
  - Połączenia z IBM MQ, IBM Integration Bus (przy użyciu połączenia w czasie rzeczywistym z brokerem) oraz WebSphere Application Server service integration bus
  - Wyszukiwanie administrowanych repozytorium obiektów przy użyciu początkowego interfejsu kontekstu
  - Połączenia z kolejkami (IBM MQ i WebSphere Application Server service integration bus) oraz tematy (IBM MQ, połączenie w czasie rzeczywistym z brokerem i WebSphere Application Server service integration bus)
  - Komunikaty podstawowe, bajtowe, mapy, obiekty, strumienie i tekst
- Przykładowa aplikacja konsumująca wiadomości obsługuje synchroniczne i asynchroniczne tryby odbierania oraz instrukcje Selector SQL.
- Przykładowa aplikacja producenta komunikatów obsługuje trwałe i nietrwałe tryby dostarczania.

Tryby pracy

Próbki mogą działać w jednym z dwóch trybów:

tryb prosty
Użytkownik może uruchomić przykłady z minimalnym wejściem użytkownika.

tryb zaawansowany
Bardziej precyzyjnie dostosować sposób działania przykładów.

Wszystkie próbki są kompatybilne i mogą w związku z tym działać w różnych językach.

Pojęcia pokrewne

Budowanie własnych aplikacji
Użytkownik buduje własne aplikacje, takie jak kompilacja przykładowych aplikacji.

Zadania pokrewne

Uruchamianie przykładowych aplikacji
Przykładowe aplikacje produktu .NET można uruchomić interaktywnie w trybie prostym lub zaawansowanym, albo w trybie nieinteraktywnym, używając automatycznie wygenerowanych lub dostosowanych plików odpowiedzi.

Budowanie przykładowych aplikacji produktu .NET
Podczas budowania przykładowej aplikacji produktu .NET tworzona jest wersja wykonywalna wybranej próby.

Uruchamianie przykładowych aplikacji

Przykładowe aplikacje produktu .NET można uruchomić interaktywnie w trybie prostym lub zaawansowanym, albo w trybie nieinteraktywnym, używając automatycznie wygenerowanych lub dostosowanych plików odpowiedzi.

Zanim rozpoczniesz

Przed uruchomieniem dowolnej z dostarczonych przykładowych aplikacji należy najpierw skonfigurować środowisko serwera przesyłania komunikatów, aby aplikacje mogły łączyć się z serwerem. Patrz “Konfigurowanie środowiska serwera przesyłania komunikatów” na stronie 13.

Procedura

Aby uruchomić przykładową aplikację .NET, wykonaj następujące kroki:

Wskazówka: Kiedy uruchamiasz przykładową aplikację, wpisz? w każdej chwili, aby uzyskać pomoc na temat tego, co dalej.

1. Wybierz tryb, w którym ma zostać uruchomiona aplikacja przykładowa.

   Wpisz `Advanced` lub `Simple`.
2. Odpowiedz na pytania.

   Aby wybrać wartość domyślną, która jest wyświetlana w nawiasach na końcu pytania, naciśnij klawisz Enter. Aby wybrać inną wartość, wpisz odpowiednią wartość i naciśnij klawisz Enter.

   Oto przykładowe pytanie:

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

   W tym przypadku wartością domyślną jest `wpm` (połączenie z serwerem WebSphere Application Server service integration bus).

Wyniki

Po uruchomieniu przykładowych aplikacji pliki odpowiedzi są generowane automatycznie w bieżącym katalogu roboczym. Nazwy plików odpowiedzi są w formacie *`connection_type-sample_type`*`.rsp`,

na przykład `wpm-producer.rsp`. Jeśli jest to wymagane, można użyć wygenerowanego pliku odpowiedzi, aby ponownie uruchomić aplikację przykładową z tymi samymi opcjami, tak aby nie było konieczne ponowne wprowadzanie opcji.

Pojęcia pokrewne

Przykładowe aplikacje
Przykładowe aplikacje zawierają przegląd wspólnych funkcji każdego interfejsu API. Można ich używać do weryfikowania konfiguracji instalacji i serwera przesyłania komunikatów oraz do budowania własnych aplikacji.

Zadania pokrewne

Budowanie przykładowych aplikacji produktu .NET
Podczas budowania przykładowej aplikacji produktu .NET tworzona jest wersja wykonywalna wybranej próby.

Budowanie przykładowych aplikacji produktu .NET

Podczas budowania przykładowej aplikacji produktu .NET tworzona jest wersja wykonywalna wybranej próby.

Zanim rozpoczniesz

Zainstaluj odpowiedni kompilator. W przypadku tego zadania założono, że użytkownik ma zainstalowany produkt Microsoft Visual Studio 2012 i że jest on zaznajomiony z jego używaniem.

Procedura

Aby zbudować przykładową aplikację produktu .NET , wykonaj następujące kroki:

1. Kliknij plik rozwiązania `Samples.sln` dostarczony wraz z przykładami produktu .NET .
2. Kliknij prawym przyciskiem myszy rozwiązanie `Przykłady` w oknie eksploratora rozwiązań i wybierz opcję **Zbuduj rozwiązanie**.

Wyniki

Program wykonywalny jest tworzony w odpowiednim podfolderze przykładu, `bin/Debug` lub `bin/Release`, w zależności od wybranej konfiguracji. Ten program ma taką samą nazwę, jak folder, z przyrostkiem CS. Jeśli na przykład tworzona jest wersja C# aplikacji przykładowej producenta komunikatów, produkt `SampleProducerCS.exe` jest tworzony w folderze `SampleProducer` .

Pojęcia pokrewne

Przykładowe aplikacje
Przykładowe aplikacje zawierają przegląd wspólnych funkcji każdego interfejsu API. Można ich używać do weryfikowania konfiguracji instalacji i serwera przesyłania komunikatów oraz do budowania własnych aplikacji.

“Budowanie własnych aplikacji” na stronie 44
Użytkownik buduje własne aplikacje, takie jak kompilacja przykładowych aplikacji.

Zadania pokrewne

Uruchamianie przykładowych aplikacji
Przykładowe aplikacje produktu .NET można uruchomić interaktywnie w trybie prostym lub zaawansowanym, albo w trybie nieinteraktywnym, używając automatycznie wygenerowanych lub dostosowanych plików odpowiedzi.

Tworzenie aplikacji produktu XMS

Tematy w tej sekcji zawierają informacje, które mogą być przydatne podczas pisania aplikacji XMS .

Informacje na temat pisania aplikacji XMS można znaleźć w następujących tematach:

Pisanie aplikacji produktu XMS

Tematy w tej sekcji zawierają informacje pomocne podczas pisania aplikacji XMS .

Ta sekcja zawiera ogólne pojęcia związane z pisaniem aplikacji produktu XMS . Więcej informacji na temat tworzenia aplikacji produktu .NET można znaleźć w sekcji “Pisanie aplikacji XMS .NET” na stronie 45 .

Sekcja obejmuje następujące tematy:

Pojęcia pokrewne

Pisanie aplikacji XMS .NET
Tematy zawarte w tej sekcji zawierają informacje pomocne podczas pisania aplikacji XMS .NET .

Odsyłacze pokrewne

.NET interfejsy
Ten sekcja dokumentuje interfejsy klasy .NET oraz ich właściwości i metody.

Model wielowątkowości

Ogólne reguły zarządzają sposobem, w jaki aplikacja wielowątkowa może używać obiektów XMS .

- Tylko obiekty z następujących typów mogą być używane współbieżnie w różnych wątkach:
  - ConnectionFactory
  - Połączenie
  - Dane ConnectionMeta
  - Miejsce docelowe
- Obiekt Session może być używany tylko w jednym wątku w dowolnym momencie.

Wyjątki od tych reguł są oznaczane przez wpisy z etykietą "Kontekst wątku" w definicjach interfejsu metod w produkcie “Informacje o klientach usługi komunikatów dla środowiska .NET” na stronie 92.

Pojęcia pokrewne

Warunki błędów, które mogą być obsługiwane w czasie wykonywania
Kody powrotu z wywołań funkcji API są warunkami błędów, które mogą być obsługiwane w czasie wykonywania. Sposób postępowania z tym typem błędu zależy od tego, czy używany jest interfejs API C, czy C++ .

Obiekty ConnectionFactories i obiekty Connection

Obiekt ConnectionFactory udostępnia szablon, który jest używany przez aplikację do tworzenia obiektu połączenia. Aplikacja korzysta z obiektu połączenia w celu utworzenia obiektu sesji.

W przypadku produktu .NETaplikacja XMS najpierw używa obiektu XMSFactoryFactory do uzyskania odwołania do obiektu ConnectionFactory , który jest odpowiedni do wymaganego typu protokołu. Ten obiekt ConnectionFactory może następnie generować połączenia tylko dla tego typu protokołu.

Aplikacja XMS może tworzyć wiele połączeń, a aplikacja wielowątkowa może korzystać jednocześnie z pojedynczego obiektu połączenia w wielu wątkach. Obiekt połączenia hermetyzuje połączenie komunikacyjne między aplikacją a serwerem przesyłania komunikatów.

Połączenie służy kilku celom:

- Gdy aplikacja tworzy połączenie, aplikacja może zostać uwierzytelniona.
- Aplikacja może powiązać unikalny identyfikator klienta z połączeniem. Identyfikator klienta jest używany do obsługi trwałych subskrypcji w domenie publikowania/subskrypcji. Identyfikator klienta można ustawić na dwa sposoby:

  Preferowanym sposobem przypisania identyfikatora klienta połączeń jest skonfigurowanie w obiekcie ConnectionFactory specyficznym dla klienta przy użyciu właściwości i w sposób przezroczysty przypisanie go do tworzonego połączenia.

  Alternatywnym sposobem przypisywania identyfikatora klienta jest użycie wartości specyficznej dla dostawcy, która jest ustawiona w obiekcie połączenia. Ta wartość nie przesłania identyfikatora, który został skonfigurowany administracyjnie. Jest on dostępny dla przypadku, w którym nie istnieje określony administracyjny identyfikator. Jeśli podany identyfikator administracyjny nie istnieje, próba przesłonięcia go z wartością specyficzną dla dostawcy powoduje zgłoszenie wyjątku. Jeśli aplikacja jawnie ustawi identyfikator, musi to zrobić natychmiast po utworzeniu połączenia, a przed innymi działaniami w tym połączeniu. W przeciwnym razie zgłaszany jest wyjątek.

Aplikacja XMS zwykle tworzy połączenie, jedną lub więcej sesji oraz liczbę producentów komunikatów i konsumentów komunikatów.

Tworzenie połączenia jest stosunkowo kosztowne pod względem zasobów systemowych, ponieważ wiąże się on z nawiązaniem połączenia komunikacyjnego, a także może wymagać uwierzytelnienia aplikacji.

Zadania pokrewne

Tworzenie obiektów administrowanych
Definicje obiektów ConnectionFactory i Destination, które są wymagane przez aplikacje produktu XMS w celu nawiązania połączenia z serwerem przesyłania komunikatów, muszą zostać utworzone przy użyciu odpowiednich narzędzi administracyjnych.

Odsyłacze pokrewne

IConnectionFactory (dla interfejsu .NET)
Aplikacja korzysta z fabryki połączeń w celu utworzenia połączenia.

Właściwości obiektu ConnectionFactory
Przegląd właściwości obiektu ConnectionFactory z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

IDestination (dla interfejsu .NET)
Miejsce docelowe to miejsce, w którym aplikacja wysyła komunikaty lub jest to źródło, z którego aplikacja odbiera komunikaty, lub oba te komunikaty.

Właściwości miejsca docelowego
Przegląd właściwości obiektu docelowego, z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

Uruchomiono i zatrzymano połączenie

Połączenie może działać w trybie uruchomionym lub zatrzymanego.

Gdy aplikacja tworzy połączenie, połączenie jest w trybie zatrzymanego. Gdy połączenie jest w trybie zatrzymanych, aplikacja może inicjować sesje i może wysyłać komunikaty, ale nie może ich odbierać, synchronicznie lub asynchronicznie.

Aplikacja może uruchomić połączenie, wywołując metodę `Start Connection` . Gdy połączenie jest w trybie uruchomionym, aplikacja może wysyłać i odbierać komunikaty. Aplikacja może następnie zatrzymać i zrestartować połączenie, wywołując metody zatrzymania połączenia i `Start Connection` .

Pojęcia pokrewne

Zamknięcie połączenia
Aplikacja zamknie połączenie, wywołując metodę Close Connection (Zamknij połączenie).

Obsługa wyjątków
Jeśli aplikacja korzysta z połączenia tylko w celu asynchronicznego korzystania z komunikatów, dowiaduje się o problemie z połączeniem tylko przy użyciu obiektu nasłuchiwania wyjątków.

Połączenie z magistralą integracji usług
Aplikacja XMS może łączyć się z magistralą integracji usług WebSphere Application Server za pomocą bezpośredniego połączenia TCP/IP lub za pomocą protokołu HTTP przy użyciu protokołu TCP/IP.

Zamknięcie połączenia

Aplikacja zamknie połączenie, wywołując metodę Close Connection (Zamknij połączenie).

Gdy aplikacja zamknie połączenie, program XMS wykonuje następujące działania:

- Zamyka ona wszystkie sesje powiązane z połączeniem i usuwa niektóre obiekty powiązane z tymi sesjami. Więcej informacji o tym, które obiekty są usuwane, zawiera sekcja “Usuwanie obiektu” na stronie 39. W tym samym czasie program XMS wycofuje wszystkie transakcje, które są aktualnie w toku w ramach sesji.
- Kończy on połączenie komunikacyjne z serwerem przesyłania komunikatów.
- Zwalnia ona pamięć i inne zasoby wewnętrzne używane przez połączenie.

Program XMS nie potwierdza odbioru żadnych komunikatów, które nie potwierdziły podczas sesji, przed zamknięciem połączenia. Więcej informacji na temat potwierdzania odbioru komunikatów zawiera sekcja “Potwierdzenie komunikatu” na stronie 26.

Pojęcia pokrewne

Uruchomiono i zatrzymano połączenie
Połączenie może działać w trybie uruchomionym lub zatrzymanego.

Obsługa wyjątków
Jeśli aplikacja korzysta z połączenia tylko w celu asynchronicznego korzystania z komunikatów, dowiaduje się o problemie z połączeniem tylko przy użyciu obiektu nasłuchiwania wyjątków.

Połączenie z magistralą integracji usług
Aplikacja XMS może łączyć się z magistralą integracji usług WebSphere Application Server za pomocą bezpośredniego połączenia TCP/IP lub za pomocą protokołu HTTP przy użyciu protokołu TCP/IP.

Obsługa wyjątków

Jeśli aplikacja korzysta z połączenia tylko w celu asynchronicznego korzystania z komunikatów, dowiaduje się o problemie z połączeniem tylko przy użyciu obiektu nasłuchiwania wyjątków.

Wyjątki XMS .NET wywodzi się z wyjątku System.Exception. Więcej informacji na ten temat zawiera sekcja “Obsługa błędów w produkcie .NET” na stronie 49.

Pojęcia pokrewne

Uruchomiono i zatrzymano połączenie
Połączenie może działać w trybie uruchomionym lub zatrzymanego.

Zamknięcie połączenia
Aplikacja zamknie połączenie, wywołując metodę Close Connection (Zamknij połączenie).

Połączenie z magistralą integracji usług

Aplikacja XMS może łączyć się z magistralą integracji usług WebSphere Application Server za pomocą bezpośredniego połączenia TCP/IP lub za pomocą protokołu HTTP przy użyciu protokołu TCP/IP.

Połączenie z magistralą integracji usług

Aplikacja XMS może łączyć się z magistralą integracji usług WebSphere Application Server za pomocą bezpośredniego połączenia TCP/IP lub za pomocą protokołu HTTP przy użyciu protokołu TCP/IP.

Protokół HTTP może być używany w sytuacjach, gdy bezpośrednie połączenie TCP/IP nie jest możliwe. Jedną z typowych sytuacji jest komunikacja za pośrednictwem firewalla, na przykład w przypadku wymiany komunikatów przez dwa przedsiębiorstwa. Używanie protokołu HTTP do komunikacji za pośrednictwem firewalla jest często określane jako *tunelowanie HTTP.* Tunelowanie HTTP jest jednak z natury wolniejsze niż użycie bezpośredniego połączenia TCP/IP, ponieważ nagłówki HTTP dodają znacznie do ilości przesyłanych danych, a protokół HTTP wymaga większej liczby przepływów komunikacji niż TCP/IP.

Aby utworzyć połączenie TCP/IP, aplikacja może używać fabryki połączeń, której właściwość XMSC_WPM_TARGET_TRANSPORT_CHAIN jest ustawiona na wartość XMSC_WPM_TARGET_TRANSPORT_CHAIN_BASIC. Jest to wartość domyślna właściwości. Jeśli połączenie zostanie utworzone pomyślnie, właściwość XMSC_WPM_CONNECTION_PROTOCOL połączenia zostanie ustawiona na wartość XMSC_WPM_CP_TCP.

Aby utworzyć połączenie, które korzysta z protokołu HTTP, aplikacja musi używać fabryki połączeń, której właściwość XMSC_WPM_TARGET_TRANSPORT_CHAIN jest ustawiona na nazwę łańcucha transportowego danych przychodzących, który jest skonfigurowany do korzystania z kanału transportowego HTTP. Jeśli połączenie zostanie pomyślnie utworzone, właściwość XMSC_WPM_CONNECTION_PROTOCOL połączenia zostanie ustawiona na wartość XMSC_WPM_CP_HTTP. Informacje na temat sposobu konfigurowania łańcuchów transportowych zawiera sekcja Konfigurowanie łańcuchów transportowych w dokumentacji produktu WebSphere Application Server .

Podczas nawiązywania połączenia z serwerem startowym aplikacja ma podobny wybór protokołów komunikacyjnych. Właściwość XMSC_WPM_PROVIDER_ENDPOINTS fabryki połączeń jest sekwencją jednego lub większej liczby adresów punktów końcowych serwerów startowych. Program startowy łańcucha transportowego każdego adresu punktu końcowego może być albo XMSC_WPM_BOOTSTRAP_TCP, dla połączenia TCP/IP z serwerem startowym lub XMSC_WPM_BOOTSTRAP_HTTP, dla połączenia, które używa protokołu HTTP.

Pojęcia pokrewne

Uruchomiono i zatrzymano połączenie
Połączenie może działać w trybie uruchomionym lub zatrzymanego.

Zamknięcie połączenia
Aplikacja zamknie połączenie, wywołując metodę Close Connection (Zamknij połączenie).

Obsługa wyjątków
Jeśli aplikacja korzysta z połączenia tylko w celu asynchronicznego korzystania z komunikatów, dowiaduje się o problemie z połączeniem tylko przy użyciu obiektu nasłuchiwania wyjątków.

Zadania pokrewne

Tworzenie obiektów administrowanych
Definicje obiektów ConnectionFactory i Destination, które są wymagane przez aplikacje produktu XMS w celu nawiązania połączenia z serwerem przesyłania komunikatów, muszą zostać utworzone przy użyciu odpowiednich narzędzi administracyjnych.

Odsyłacze pokrewne

IConnectionFactory (dla interfejsu .NET)
Aplikacja korzysta z fabryki połączeń w celu utworzenia połączenia.

Właściwości obiektu ConnectionFactory
Przegląd właściwości obiektu ConnectionFactory z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

IDestination (dla interfejsu .NET)

Miejsce docelowe to miejsce, w którym aplikacja wysyła komunikaty lub jest to źródło, z którego aplikacja odbiera komunikaty, lub oba te komunikaty.

Właściwości miejsca docelowego
Przegląd właściwości obiektu docelowego, z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

Sesje

Sesja jest jednowątkowym kontekstem do wysyłania i odbierania komunikatów.

Aplikacja może korzystać z sesji w celu tworzenia komunikatów, producentów komunikatów, konsumentów komunikatów, przeglądarek kolejek i miejsc docelowych tymczasowych. Aplikacja może również używać sesji do uruchamiania transakcji lokalnych.

Aplikacja może tworzyć wiele sesji, w których każda sesja generuje i konsumuje komunikaty niezależnie od innych sesji. Jeśli dwa konsumenty komunikatów w oddzielnych sesjach (lub nawet w tej samej sesji) zasubskrybują ten sam temat, każdy otrzymuje kopię dowolnego komunikatu opublikowanego w tym temacie.

W przeciwieństwie do obiektu połączenia, obiekt Session nie może być jednocześnie używany w różnych wątkach. Tylko metoda Close Session obiektu Session może być wywoływana z wątku innego niż ten, który jest używany przez obiekt Session w danym momencie. Metoda Close Session kończy sesję i zwalnia wszystkie zasoby systemowe przydzielone do sesji.

Jeśli aplikacja musi przetwarzać komunikaty współbieżnie w więcej niż jednym wątku, aplikacja musi utworzyć sesję dla każdego wątku, a następnie użyć tej sesji dla każdej operacji wysyłania lub odbierania w obrębie tego wątku.

Sesje transakcyjne

Aplikacje produktu XMS mogą uruchamiać transakcje lokalne. *Transakcja lokalna* to transakcja, która obejmuje zmiany tylko w przypadku zasobów menedżera kolejek lub magistrali integracji usług, z którymi połączona jest aplikacja.

Informacje zawarte w tym temacie są istotne tylko wtedy, gdy aplikacja łączy się z menedżerem kolejek produktu IBM MQ lub z magistralą integracji usług produktu WebSphere Application Server . Informacje te nie mają znaczenia dla połączenia w czasie rzeczywistym z brokerem.

Aby uruchomić transakcje lokalne, aplikacja musi najpierw utworzyć sesję transakcyjne, wywołując metodę Create Session obiektu połączenia, określając jako parametr, że sesja jest transakcjami transakcjami. Następnie wszystkie komunikaty wysłane i odebrane w ramach sesji są pogrupowane w sekwencję transakcji. Transakcja kończy się, gdy aplikacja zatwierdza lub wycofuje komunikaty, które zostały wysłane i odebrane od momentu rozpoczęcia transakcji.

Aby zatwierdzić transakcję, aplikacja wywołuje metodę zatwierdzania obiektu sesji. Jeśli transakcja zostanie zatwierdzona, wszystkie komunikaty wysłane w ramach transakcji stają się dostępne do dostarczenia do innych aplikacji, a wszystkie komunikaty odebrane w ramach transakcji zostaną potwierdzone, tak aby serwer przesyłania komunikatów nie próbował ponownie dostarczyć ich do aplikacji. W domenie typu punkt z punktem serwer przesyłania komunikatów usuwa również odebrane komunikaty z ich kolejek.

Aby wycofać transakcję, aplikacja wywołuje metodę Rollback obiektu Session. Po wycofaniu transakcji wszystkie komunikaty wysłane w ramach transakcji są odrzucane przez serwer przesyłania komunikatów, a wszystkie komunikaty odebrane w ramach transakcji stają się dostępne do dostarczenia ponownie. W domenie typu punkt z punktem komunikaty, które zostały odebrane, są ponownie umieszczane w ich kolejkach i ponownie stają się widoczne dla innych aplikacji.

Nowa transakcja jest uruchamiana automatycznie, gdy aplikacja tworzy sesję transakcyjna lub wywołuje metodę zatwierdzania lub wycofania zmian. Oznacza to, że sesja transakcyjna zawsze ma aktywną transakcję.

Gdy aplikacja zamknie sesję transakcyjne, następuje niejawne wycofanie zmian. Jeśli aplikacja zamknie połączenie, dla wszystkich sesji transakcyjnych połączenia zostanie wykonane niejawne wycofanie zmian.

Transakcja jest w całości zawarta w ramach sesji transakcyjnej. Transakcja nie może obejmować sesji. Oznacza to, że aplikacja nie może wysyłać i odbierać komunikatów w dwóch lub większej liczby sesji transakcyjnych, a następnie zatwierdzać lub wycofywanie wszystkich tych działań jako jednej transakcji.

Pojęcia pokrewne

Potwierdzenie komunikatu
Każda sesja, która nie jest transakcowana, ma tryb potwierdzenia, który określa sposób, w jaki odbierane są komunikaty odbierane przez aplikację. Dostępne są trzy tryby potwierdzenia, a wybór trybu potwierdzania wpływa na konstrukcję aplikacji.

Dostarczanie komunikatów asynchronicznych
Produkt XMS używa jednego wątku do obsługi wszystkich asynchronicznych dostaw komunikatów dla sesji. Oznacza to, że jednorazowo może być uruchomiona tylko jedna funkcja nasłuchiwania komunikatów lub jedna metoda `onMessage()` .

Synchroniczne dostarczanie komunikatów
Komunikaty są dostarczane synchronicznie do aplikacji, jeśli w aplikacji używane są metody odbierania obiektów MessageConsumer .

Tryb dostarczania wiadomości
Produkt XMS obsługuje dwa tryby dostarczania komunikatów.

Potwierdzenie komunikatu

Każda sesja, która nie jest transakcowana, ma tryb potwierdzenia, który określa sposób, w jaki odbierane są komunikaty odbierane przez aplikację. Dostępne są trzy tryby potwierdzenia, a wybór trybu potwierdzania wpływa na konstrukcję aplikacji.

Informacje zawarte w tym temacie są istotne tylko wtedy, gdy aplikacja łączy się z menedżerem kolejek produktu IBM MQ lub z magistralą integracji usług produktu WebSphere Application Server . Informacje te nie mają znaczenia dla połączenia w czasie rzeczywistym z brokerem.

Produkt XMS korzysta z tego samego mechanizmu do potwierdzania odbioru komunikatów, których używa JMS.

Jeśli sesja nie jest transakowana, sposób potwierdzania komunikatów odbieranych przez aplikację jest określany przez tryb potwierdzania sesji. Trzy tryby potwierdzenia są opisane w następujących akapitach:

XMSC_AUTO_ACKNOWLEDGE
Sesja automatycznie potwierdzi każdy komunikat otrzymany przez aplikację.

Jeśli komunikaty są dostarczane synchronicznie do aplikacji, sesja potwierdza otrzymanie komunikatu za każdym razem, gdy wywołanie Odbiór zakończy się pomyślnie.

Jeśli aplikacja pomyślnie odbierze komunikat, ale niepowodzenie uniemożliwia potwierdzenie wystąpienia, komunikat staje się dostępny do ponownego dostarczenia. Dlatego aplikacja musi mieć możliwość obsługi komunikatu, który został ponownie dostarczony.

XMSC_DUPS_OK_ACKNOWLEDGE
Sesja potwierdza komunikaty odebrane przez aplikację w czasie, gdy jest ona wybierana.

Użycie tego trybu potwierdzenia zmniejsza ilość pracy, jaką musi wykonać sesja, ale błąd, który uniemożliwia potwierdzenie komunikatu, może spowodować ponowne udostępnienie więcej niż jednego komunikatu do dostarczenia. Dlatego aplikacja musi mieć możliwość obsługi komunikatów, które są ponownie dostarczane.

POTWIERDZANIE Xmsc_client_acknowledge
Aplikacja potwierdza otrzymywane przez niego komunikaty, wywołując metodę Acknowledge klasy Message.

Aplikacja może potwierdzić odbiór każdego komunikatu indywidualnie lub może otrzymać partię komunikatów i wywołać metodę Acknowledge tylko dla ostatniego komunikatu, który otrzymuje. Gdy metoda Acknowledge jest nazywana wszystkimi wiadomościami otrzymanymi od czasu ostatniego wywołania metody, są one potwierdzane.

W połączeniu z dowolnym z tych trybów potwierdzania aplikacja może zatrzymać i ponownie uruchomić dostarczanie komunikatów w sesji, wywołując metodę Recover klasy Session. Wiadomości, których przyjęcie zostało wcześniej niepotwierdzone, są ponownie dostarczane. Mogą one jednak nie być dostarczane w tej samej kolejności, w jakiej zostały dostarczone wcześniej. W międzyczasie mogły zostać wysłane komunikaty o wyższym priorytecie, a niektóre z oryginalnych komunikatów mogły utraci ważność. W domenie typu punkt z punktem niektóre z oryginalnych komunikatów mogły zostać skonsumowane przez inną aplikację.

Aplikacja może określić, czy komunikat jest ponownie dostarczany, sprawdzając zawartość pola nagłówka JMSRedelivered komunikatu. Aplikacja wykonuje tę aplikację, wywołując metodę Get JMSRedelivered klasy Message.

Pojęcia pokrewne

Sesje transakcyjne
Aplikacje produktu XMS mogą uruchamiać transakcje lokalne. *Transakcja lokalna* to transakcja, która obejmuje zmiany tylko w przypadku zasobów menedżera kolejek lub magistrali integracji usług, z którymi połączona jest aplikacja.

Dostarczanie komunikatów asynchronicznych
Produkt XMS używa jednego wątku do obsługi wszystkich asynchronicznych dostaw komunikatów dla sesji. Oznacza to, że jednorazowo może być uruchomiona tylko jedna funkcja nasłuchiwania komunikatów lub jedna metoda `onMessage()`.

Synchroniczne dostarczanie komunikatów
Komunikaty są dostarczane synchronicznie do aplikacji, jeśli w aplikacji używane są metody odbierania obiektów MessageConsumer .

Tryb dostarczania wiadomości
Produkt XMS obsługuje dwa tryby dostarczania komunikatów.

Dostarczanie komunikatów asynchronicznych

Produkt XMS używa jednego wątku do obsługi wszystkich asynchronicznych dostaw komunikatów dla sesji. Oznacza to, że jednorazowo może być uruchomiona tylko jedna funkcja nasłuchiwania komunikatów lub jedna metoda `onMessage()`.

Jeśli więcej niż jeden konsument komunikatów w sesji odbiera komunikaty asynchronicznie, a funkcja nasłuchiwania komunikatów lub metoda `onMessage()` dostarcza komunikat do konsumenta komunikatów, to wszystkie inne konsumenty komunikatów, które oczekują na ten sam komunikat, muszą nadal czekać. Inne komunikaty, które oczekują na dostarczenie do sesji, muszą również czekać.

Jeśli aplikacja wymaga współbieżnego dostarczania komunikatów, utwórz więcej niż jedną sesję, tak aby produkt XMS używał więcej niż jednego wątku do obsługi asynchronicznego dostarczania komunikatów. W ten sposób współbieżnie może działać więcej niż jedna funkcja nasłuchiwania komunikatów lub metoda `onMessage()`.

Sesja nie jest asynchroniczna, przypisując obiekt nasłuchiwania komunikatów do konsumenta. Sesja staje się asynchroniczna tylko wtedy, gdy wywoływana jest metoda `Connection.Start`. Wszystkie wywołania synchroniczne są dozwolone, dopóki metoda `Connection.Start` nie zostanie wywołana. Dostarczanie komunikatów do konsumentów rozpoczyna się po wywołaniu `Connection.Start` .

Jeśli wywołania synchroniczne, takie jak tworzenie konsumenta lub producenta, muszą zostać wykonane w sesji asynchronicznej, należy wywołać funkcję `Connection.Stop` . Sesję można wznowić, wywołując metodę `Connection.Start` w celu rozpoczęcia dostarczania komunikatów. Jedynym wyjątkiem jest wątek dostarczania komunikatów sesji, który jest wątkiem, który dostarcza komunikaty do funkcji zwrotnej. Ten wątek może wywołać w funkcji zwrotnej komunikaty wywołanie sesji (z wyjątkiem wywołania Zamknij).

Uwaga: W trybie niezarządzanym wywołanie MQDISC w ramach funkcji połączenia telefocyjnego nie jest obsługiwane przez klienta IBM MQ .NET . Dlatego aplikacja kliencka nie może tworzyć ani zamykać sesji w ramach wywołania zwrotnego MessageListener w asynchronicznym trybie odbioru. Utwórz i rozdysponuj sesję poza metodą `MessageListener`.

Pojęcia pokrewne

Sesje transakcyjne
Aplikacje produktu XMS mogą uruchamiać transakcje lokalne. *Transakcja lokalna* to transakcja, która obejmuje zmiany tylko w przypadku zasobów menedżera kolejek lub magistrali integracji usług, z którymi połączona jest aplikacja.

Potwierdzenie komunikatu
Każda sesja, która nie jest transakcowana, ma tryb potwierdzenia, który określa sposób, w jaki odbierane są komunikaty odbierane przez aplikację. Dostępne są trzy tryby potwierdzenia, a wybór trybu potwierdzania wpływa na konstrukcję aplikacji.

Synchroniczne dostarczanie komunikatów
Komunikaty są dostarczane synchronicznie do aplikacji, jeśli w aplikacji używane są metody odbierania obiektów MessageConsumer .

Tryb dostarczania wiadomości
Produkt XMS obsługuje dwa tryby dostarczania komunikatów.

Synchroniczne dostarczanie komunikatów

Komunikaty są dostarczane synchronicznie do aplikacji, jeśli w aplikacji używane są metody odbierania obiektów MessageConsumer .

Za pomocą metod odbierania aplikacja może czekać na określony czas dla komunikatu lub może czekać na czas nieokreślony. Alternatywnie, jeśli aplikacja nie chce czekać na komunikat, może skorzystać z metody receive bez oczekiwania.

Pojęcia pokrewne

Sesje transakcyjne
Aplikacje produktu XMS mogą uruchamiać transakcje lokalne. *Transakcja lokalna* to transakcja, która obejmuje zmiany tylko w przypadku zasobów menedżera kolejek lub magistrali integracji usług, z którymi połączona jest aplikacja.

Potwierdzenie komunikatu
Każda sesja, która nie jest transakcowana, ma tryb potwierdzenia, który określa sposób, w jaki odbierane są komunikaty odbierane przez aplikację. Dostępne są trzy tryby potwierdzenia, a wybór trybu potwierdzania wpływa na konstrukcję aplikacji.

Dostarczanie komunikatów asynchronicznych
Produkt XMS używa jednego wątku do obsługi wszystkich asynchronicznych dostaw komunikatów dla sesji. Oznacza to, że jednorazowo może być uruchomiona tylko jedna funkcja nasłuchiwania komunikatów lub jedna metoda `onMessage()` .

Tryb dostarczania wiadomości
Produkt XMS obsługuje dwa tryby dostarczania komunikatów.

Tryb dostarczania wiadomości

Produkt XMS obsługuje dwa tryby dostarczania komunikatów.

- Komunikaty *Trwałe* są dostarczane jeden raz. Serwer przesyłania komunikatów podejmuje specjalne środki ostrożności, takie jak rejestrowanie komunikatów, w celu zapewnienia, że trwałe komunikaty nie zostaną utracone podczas przesyłania, nawet w przypadku niepowodzenia.
- Komunikaty *Nietrwałe* są dostarczane nie więcej niż raz. Komunikaty nietrwałe są mniej niezawodne niż komunikaty trwałe, ponieważ mogą zostać utracone podczas przesyłania w przypadku awarii.

Wybór trybu dostawy jest kompromisem między niezawodnością a wydajnością. Komunikaty nietrwałe są zwykle transportowane szybciej niż komunikaty trwałe.

Pojęcia pokrewne

Sesje transakcyjne
Aplikacje produktu XMS mogą uruchamiać transakcje lokalne. *Transakcja lokalna* to transakcja, która obejmuje zmiany tylko w przypadku zasobów menedżera kolejek lub magistrali integracji usług, z którymi połączona jest aplikacja.

Potwierdzenie komunikatu
Każda sesja, która nie jest transakcowana, ma tryb potwierdzenia, który określa sposób, w jaki odbierane są komunikaty odbierane przez aplikację. Dostępne są trzy tryby potwierdzenia, a wybór trybu potwierdzania wpływa na konstrukcję aplikacji.

Dostarczanie komunikatów asynchronicznych
Produkt XMS używa jednego wątku do obsługi wszystkich asynchronicznych dostaw komunikatów dla sesji. Oznacza to, że jednorazowo może być uruchomiona tylko jedna funkcja nasłuchiwania komunikatów lub jedna metoda `onMessage()`.

Synchroniczne dostarczanie komunikatów
Komunikaty są dostarczane synchronicznie do aplikacji, jeśli w aplikacji używane są metody odbierania obiektów MessageConsumer .

Miejsca docelowe

Aplikacja XMS używa obiektu docelowego do określenia miejsca docelowego wysyłanych komunikatów oraz źródła komunikatów, które są odbierane.

Aplikacja XMS może albo utworzyć obiekt docelowy w czasie wykonywania, albo uzyskać predefiniowane miejsce docelowe z repozytorium administrowanych obiektów.

Podobnie jak w przypadku elementu ConnectionFactory, najbardziej elastycznym sposobem na określenie miejsca docelowego przez aplikację XMS jest zdefiniowanie go jako obiektu administrowanego. Korzystając z tego podejścia, aplikacje napisane w językach C, C + + i .NET oraz Javamogą współużytkować definicje miejsca docelowego. Właściwości administrowanych obiektów docelowych można zmieniać bez zmieniania dowolnego kodu.

W przypadku aplikacji .NET miejsce docelowe jest tworzone przy użyciu metody CreateTopic lub CreateQueue . Te dwie metody są dostępne zarówno w obiektach ISession, jak i XMSFactoryFactory w interfejsie API produktu .NET . Więcej informacji na ten temat zawierają sekcje “Miejsca docelowe w środowisku .NET” na stronie 47 i “Miejsce docelowe” na stronie 110.

Odsyłacze pokrewne
IDestination (dla interfejsu .NET)
Miejsce docelowe to miejsce, w którym aplikacja wysyła komunikaty lub jest to źródło, z którego aplikacja odbiera komunikaty, lub oba te komunikaty.

Właściwości miejsca docelowego
Przegląd właściwości obiektu docelowego, z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

Identyfikatory ujednolicona zasobu tematu

Identyfikator URI (Uniform Resource Identifier) tematu określa nazwę tematu. Można również określić dla niego jedną lub więcej właściwości.

Identyfikator URI tematu rozpoczyna się od tematu sekwencji: //, po którym następuje nazwa tematu i (opcjonalnie) lista par nazwa-wartość, które ustawiają pozostałe właściwości tematu. Nazwa tematu nie może być pusta.

Poniżej przedstawiono przykład we fragmencie kodu .NET :

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

Więcej informacji na temat właściwości tematu, w tym nazwy i poprawnych wartości, których można używać w identyfikatorze URI, zawiera sekcja “Właściwości miejsca docelowego” na stronie 189.

Podczas określania identyfikatora URI tematu w celu użycia w subskrypcji można używać znaków wieloznacznych. Składnia dla tych znaków wieloznacznych zależy od typu połączenia i wersji brokera. Dostępne są następujące opcje:

- Menedżer kolejek produktu IBM WebSphere MQ 7.0 z użyciem znaków wieloznacznych na poziomie znaku

- Menedżer kolejek produktu IBM WebSphere MQ 7.0 z użyciem znaków wieloznacznych na poziomie tematu
- WebSphere Application Server magistrala integracji usług

Menedżer kolejek produktu IBM WebSphere MQ 7.0 z użyciem znaków wieloznacznych na poziomie znaku

W produkcie IBM WebSphere MQ 7.0 menedżer kolejek z użyciem znaków wieloznacznych na poziomie znaku używa następujących znaków wieloznacznych:

* dla 0 lub więcej znaków
? dla 1 znaku
% dla znaku zmiany znaczenia

W sekcji Tabela 1 na stronie 30 przedstawiono przykłady użycia tego schematu znaków wieloznacznych.

Tabela 1. Przykładowe identyfikatory URI korzystające ze znaków wieloznacznych na poziomie znaku dla menedżera kolejek produktu IBM WebSphere MQ 7.0

Jednolity identyfikator zasobu	**Zgodne**	**Przykłady**
"topic://Sport *Wyniki"	Wszystkie tematy zaczynając od "Sport" i kończąc w "Wyniki"	"topic://SportsResults" i "topic://Sport/Hockey/National/Div3/Results"
" topic://Sport?Wyniki "	Wszystkie tematy zaczynając od "Sport", po którym następuje jeden znak, po którym następuje "Results"	"topic://SportsResults" i "topic://SportXResults"
"topic://Sport/ * ball*/Div? / Results/*/???"	Tematy	"topic://Sport/Football/Div1/Results/2002/Nov" oraz "topic://Sport/Netball/National/Div3/Results/02/Jan"

Menedżer kolejek produktu IBM WebSphere MQ 7.0 z formatem dzikiego karty na poziomie tematu

Menedżer kolejek produktu IBM WebSphere MQ 7.0 z użyciem znaków wieloznacznych na poziomie tematu używa następujących znaków wieloznacznych:

aby dopasować wiele poziomów
+ dopasowanie do jednego poziomu

W programie Tabela 2 na stronie 30 przedstawiono przykłady użycia tego schematu znaków wieloznacznych.

Tabela 2. Przykładowe identyfikatory URI z użyciem schematu zastępczego na poziomie tematu dla menedżera kolejek produktu IBM WebSphere MQ 7.0

Jednolity identyfikator zasobu	**Zgodne**	**Przykłady**
"topic://Sport/ + /Wyniki"	Wszystkie tematy z pojedynczą hierarchiczną nazwą poziomu między Sport a Results	"topic://Sport/Football/Results" oraz "topic://Sport/Ju-Jitsu/Results"
"topic://Sport/#/Wyniki"	Wszystkie tematy zaczynając od "Sport/" i kończąc w "/Results"	"topic://Sport/Football/Results" oraz "topic://Sport/Hockey/National/Div3/Results"

Tabela 2. Przykładowe identyfikatory URI z użyciem schematu zastępczego na poziomie tematu dla menedżera kolejek produktu IBM WebSphere MQ 7.0 (kontynuacja)		
Jednolity identyfikator zasobu	**Zgodne**	**Przykłady**
" topic://Sport/Football/#"	Wszystkie tematy zaczynając od "Sport/Football/"	"topic://Sport/Football/Results" oraz "topic://Sport/Football/TeamNews/Signings/Managerial"

WebSphere Application Server magistrala integracji usług

Magistrala integracji usług produktu WebSphere Application Server korzysta z następujących znaków wieloznacznych:

*, aby dopasować dowolne znaki na jednym poziomie w hierarchii
// w celu dopasowania do 0 lub więcej poziomów
//. w celu dopasowania do wartości 0 lub większej liczby poziomów (na końcu wyrażenia tematu)

W programie Tabela 3 na stronie 31 przedstawiono przykłady użycia tego schematu znaków wieloznacznych.

Tabela 3. Przykładowe identyfikatory URI przy użyciu schematu zastępczego dla magistrali integracji usług produktu WebSphere Application Server		
Jednolity identyfikator zasobu	**Zgodne**	**Przykłady**
"topic://Sport/ * ball/Wyniki"	Wszystkie tematy z pojedynczą hierarchiczną nazwą poziomu kończącą się na "ball" pomiędzy Sport a Results	"topic://Sport/Football/Results" oraz "topic://Sport/Netball/Results"
"topic://Sport//Wyniki"	Wszystkie tematy zaczynając od "Sport/" i kończąc w "/Results"	"topic://Sport/Football/Results" oraz "topic://Sport/Hockey/National/Div3/Results"
"topic://Sport/Football//."	Wszystkie tematy zaczynając od "Sport/Football/"	"topic://Sport/Football/Results" oraz "topic://Sport/Football/TeamNews/Signings/Managerial"
"topic://Sport/ * ball//Results//."	Tematy	"topic://Sport/Football/Results" oraz "topic://Sport/Netball/National/Div3/Results/2002/November"

Pojęcia pokrewne

Identyfikatory URI jednorodnych zasobów
Identyfikator URI kolejki określa nazwę kolejki. Można również określić jedną lub więcej właściwości kolejki.

Tymczasowe miejsca docelowe
Aplikacje produktu XMS mogą tworzyć tymczasowe miejsca docelowe i korzystać z nich.

Znaki wieloznaczne miejsca docelowego
Produkt XMS udostępnia obsługę znaków wieloznacznych w celu zapewnienia, że znaki wieloznaczne mogą być przekazywane do miejsca, w którym są one potrzebne do dopasowania. Dla każdego typu serwera, z którym może pracować produkt XMS , istnieje inny schemat znaków wieloznacznych.

Odsyłacze pokrewne

IDestination (dla interfejsu .NET)
Miejsce docelowe to miejsce, w którym aplikacja wysyła komunikaty lub jest to źródło, z którego aplikacja odbiera komunikaty, lub oba te komunikaty.

Właściwości miejsca docelowego

Przegląd właściwości obiektu docelowego, z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

Identyfikatory URI jednorodnych zasobów

Identyfikator URI kolejki określa nazwę kolejki. Można również określić jedną lub więcej właściwości kolejki.

Identyfikator URI dla kolejki rozpoczyna się od kolejki sekwencji: //, po której następuje nazwa kolejki. Może ona także zawierać listę par nazwa-wartość, które ustawiają pozostałe właściwości kolejki.

W przypadku kolejek produktu IBM MQ (ale nie dla kolejek domyślnego dostawcy przesyłania komunikatów produktu WebSphere Application Server) menedżer kolejek, w którym rezyduje kolejka, może zostać określony przed kolejką, a nazwa menedżera kolejek zostanie oddzielona/oddzielająca od nazwy kolejki.

Jeśli określony jest menedżer kolejek, musi to być ten, do którego program XMS jest bezpośrednio połączony dla połączenia przy użyciu tej kolejki lub musi być dostępny z tej kolejki. Zdalne menedżery kolejek są obsługiwane tylko w przypadku pobierania komunikatów z kolejek, a nie w celu umieszczania komunikatów w kolejkach. Szczegółowe informacje na ten temat można znaleźć w dokumentacji menedżera kolejek produktu IBM MQ .

Jeśli nie został określony żaden menedżer kolejek, to dodatkowy/separator jest opcjonalny, a jego obecność lub brak nie ma znaczenia dla definicji kolejki.

Następujące definicje kolejek są równoważne dla kolejki IBM MQ o nazwie QB w menedżerze kolejek o nazwie QM_A, z którą XMS jest bezpośrednio połączone:

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

Pojęcia pokrewne

Identyfikatory ujednolicona zasobu tematu
Identyfikator URI (Uniform Resource Identifier) tematu określa nazwę tematu. Można również określić dla niego jedną lub więcej właściwości.

Tymczasowe miejsca docelowe
Aplikacje produktu XMS mogą tworzyć tymczasowe miejsca docelowe i korzystać z nich.

Znaki wieloznaczne miejsca docelowego
Produkt XMS udostępnia obsługę znaków wieloznacznych w celu zapewnienia, że znaki wieloznaczne mogą być przekazywane do miejsca, w którym są one potrzebne do dopasowania. Dla każdego typu serwera, z którym może pracować produkt XMS , istnieje inny schemat znaków wieloznacznych.

Odsyłacze pokrewne

IDestination (dla interfejsu .NET)
Miejsce docelowe to miejsce, w którym aplikacja wysyła komunikaty lub jest to źródło, z którego aplikacja odbiera komunikaty, lub oba te komunikaty.

Właściwości miejsca docelowego
Przegląd właściwości obiektu docelowego, z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

Tymczasowe miejsca docelowe

Aplikacje produktu XMS mogą tworzyć tymczasowe miejsca docelowe i korzystać z nich.

Aplikacja zwykle używa tymczasowego miejsca docelowego do odbierania odpowiedzi na komunikaty żądania. Aby określić miejsce docelowe, w którym ma zostać wysłana odpowiedź na komunikat żądania, aplikacja wywołuje metodę Set JMSReplyTo obiektu Message reprezentującego komunikat żądania. Miejsce docelowe określone w wywołaniu może być miejscem docelowym.

Chociaż sesja jest używana do tworzenia tymczasowego miejsca docelowego, zasięg tymczasowego miejsca docelowego jest w rzeczywistości połączeniem, który został użyty do utworzenia sesji.

Każdy z sesji połączenia może tworzyć producentów komunikatów i konsumentów komunikatów dla tymczasowego miejsca docelowego. Tymczasowe miejsce docelowe pozostaje do czasu jawnego usunięcia lub nastąpi zakończenie połączenia, w zależności od tego, co nastąpi wcześniej.

Gdy aplikacja tworzy kolejkę tymczasową, tworzona jest kolejka na serwerze przesyłania komunikatów, z którym połączona jest aplikacja. Jeśli aplikacja jest połączona z menedżerem kolejek, tworzona jest kolejka dynamiczna z kolejki modelowej, której nazwa jest określona za pomocą właściwości XMSC_WMQ_TEMPORARY_MODEL , a przedrostek używany do tworzenia nazwy kolejki dynamicznej jest określany przez właściwość XMSC_WMQ_TEMP_Q_PREFIX . Jeśli aplikacja jest połączona z magistralą integracji usług, w magistrali tworzona jest kolejka tymczasowa, a przedrostek używany do tworzenia nazwy kolejki tymczasowej jest określony przez właściwość XMSC_WPM_TEMP_Q_PREFIX .

Gdy aplikacja połączona z magistralą integracji usług tworzy temat tymczasowy, przedrostek używany do tworzenia nazwy tematu tymczasowego jest określany przez właściwość XMSC_WPM_TEMP_TOPIC_PREFIX .

Pojęcia pokrewne

Identyfikatory ujednolicona zasobu tematu
Identyfikator URI (Uniform Resource Identifier) tematu określa nazwę tematu. Można również określić dla niego jedną lub więcej właściwości.

Identyfikatory URI jednorodnych zasobów
Identyfikator URI kolejki określa nazwę kolejki. Można również określić jedną lub więcej właściwości kolejki.

Znaki wieloznaczne miejsca docelowego
Produkt XMS udostępnia obsługę znaków wieloznacznych w celu zapewnienia, że znaki wieloznaczne mogą być przekazywane do miejsca, w którym są one potrzebne do dopasowania. Dla każdego typu serwera, z którym może pracować produkt XMS , istnieje inny schemat znaków wieloznacznych.

Odsyłacze pokrewne

IDestination (dla interfejsu .NET)
Miejsce docelowe to miejsce, w którym aplikacja wysyła komunikaty lub jest to źródło, z którego aplikacja odbiera komunikaty, lub oba te komunikaty.

Właściwości miejsca docelowego
Przegląd właściwości obiektu docelowego, z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

Znaki wieloznaczne miejsca docelowego

Produkt XMS udostępnia obsługę znaków wieloznacznych w celu zapewnienia, że znaki wieloznaczne mogą być przekazywane do miejsca, w którym są one potrzebne do dopasowania. Dla każdego typu serwera, z którym może pracować produkt XMS , istnieje inny schemat znaków wieloznacznych.

Programy są następujące:

Typ połączenia	Schemat znaków wieloznacznych	Opis
WebSphere MQ menedżer kolejek	*	0 lub więcej znaków
	?	1 znak
	%	Znak zmiany znaczenia
Połączenie w czasie rzeczywistym z brokerem	#	Dopasuj wiele poziomów
	+	Zgodność z jednym poziomem

Typ połączenia	Schemat znaków wieloznacznych	Opis
WebSphere Usługa Integration Bus	* // //.	Dopasuj dowolne znaki na jednym poziomie w hierarchii Zgodność z 0 lub większą liczbą poziomów Zgodność z 0 lub większą liczbą poziomów (na końcu wyrażenia tematu)

Pojęcia pokrewne

Identyfikatory ujednolicona zasobu tematu
Identyfikator URI (Uniform Resource Identifier) tematu określa nazwę tematu. Można również określić dla niego jedną lub więcej właściwości.

Identyfikatory URI jednorodnych zasobów
Identyfikator URI kolejki określa nazwę kolejki. Można również określić jedną lub więcej właściwości kolejki.

Tymczasowe miejsca docelowe
Aplikacje produktu XMS mogą tworzyć tymczasowe miejsca docelowe i korzystać z nich.

Odsyłacze pokrewne

IDestination (dla interfejsu .NET)
Miejsce docelowe to miejsce, w którym aplikacja wysyła komunikaty lub jest to źródło, z którego aplikacja odbiera komunikaty, lub oba te komunikaty.

Właściwości miejsca docelowego
Przegląd właściwości obiektu docelowego, z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

Producenci komunikatów

W produkcie XMSproducent komunikatów może zostać utworzony z poprawnym miejscem docelowym lub bez powiązanego miejsca docelowego. Podczas tworzenia producenta komunikatów z miejscem docelowym o wartości NULL należy określić poprawne miejsce docelowe podczas wysyłania komunikatu.

Producenci komunikatów bez powiązanego miejsca docelowego

W programie XMS .NETproducent komunikatów może zostać utworzony z pustym miejscem docelowym.

Aby utworzyć producenta komunikatów bez powiązanego miejsca docelowego w przypadku korzystania z interfejsu API .NET , wartość NULL musi zostać przekazana jako parametr do metody `CreateProducer()` obiektu ISession (na przykład `session.CreateProducer(null)`). Jeśli komunikat jest wysyłany, należy jednak określić poprawne miejsce docelowe.

Producenci komunikatów z powiązanym miejscem docelowym

W tym scenariuszu producent komunikatów jest tworzony przy użyciu poprawnego miejsca docelowego. Podczas operacji wysyłania nie jest wymagane określenie miejsca docelowego.

Konsumenci komunikatów

Konsumenty komunikatów można sklasyfikować jako trwałe i nietrwałe subskrybenty oraz odbiorców komunikatów synchronicznych i asynchronicznych.

Trwali subskrybenci

Trwały subskrybent to konsument komunikatów, który odbiera wszystkie komunikaty opublikowane w temacie, w tym komunikaty opublikowane w czasie, gdy subskrybent jest nieaktywny.

Informacje zawarte w tym temacie są istotne tylko wtedy, gdy aplikacja łączy się z menedżerem kolejek produktu IBM MQ lub z magistralą integracji usług produktu WebSphere Application Server . Informacje te nie mają znaczenia dla połączenia w czasie rzeczywistym z brokerem.

Aby utworzyć trwały subskrybent dla tematu, aplikacja wywołuje metodę Create Durable Subskrybenta dla obiektu Session, określając jako parametry nazwę identyfikującą trwałą subskrypcję i obiekt docelowy reprezentujący dany temat. Aplikacja może utworzyć trwały subskrybent z selektorem komunikatów lub bez niego, a także może określić, czy trwały subskrybent ma odbierać komunikaty publikowane przez własne połączenie.

Sesja użyta do utworzenia trwałego subskrybenta musi mieć powiązany identyfikator klienta. Identyfikator klienta jest taki sam jak identyfikator powiązany z połączeniem, który jest używany do utworzenia sesji. Jest on określony w sposób opisany w sekcji “Obiekty ConnectionFactories i obiekty Connection” na stronie 22.

Nazwa, która identyfikuje trwałą subskrypcję, musi być unikalna w obrębie identyfikatora klienta, dlatego identyfikator klienta jest częścią pełnego, unikalnego identyfikatora trwałej subskrypcji. Serwer przesyłania komunikatów przechowuje rekord trwałej subskrypcji i zapewnia, że wszystkie komunikaty publikowane w tym temacie są zachowywane aż do momentu ich potwierdzenia przez trwałą subskrybent lub utracą ważność.

Serwer przesyłania komunikatów nadal utrzymuje rekord trwałej subskrypcji, nawet po zamknięciu trwałego subskrybenta. Aby ponownie wykorzystać trwałą subskrypcję, która została wcześniej utworzona, aplikacja musi utworzyć trwały subskrybent, określając tę samą nazwę subskrypcji i przy użyciu sesji z tym samym identyfikatorem klienta, co powiązane z trwałą subskrypcją. Tylko jedna sesja w danym momencie może mieć trwały subskrybent dla konkretnej trwałej subskrypcji.

Zasięgiem trwałej subskrypcji jest serwer przesyłania komunikatów, który jest rejestrowany w ramach subskrypcji. Jeśli dwa aplikacje połączone z różnymi serwerami przesyłania komunikatów tworzą trwały subskrybent przy użyciu tej samej nazwy subskrypcji i identyfikatora klienta, tworzone są dwie całkowicie niezależne subskrypcje trwałe.

Aby usunąć trwałą subskrypcję, aplikacja wywołuje metodę Unsubscribe obiektu Session, określając jako parametr nazwę identyfikującą trwałą subskrypcję. Identyfikator klienta powiązany z sesją musi być taki sam, jak identyfikator powiązany z trwałą subskrypcją. Serwer przesyłania komunikatów usuwa rekord trwałej subskrypcji, która jest konserwowana, i nie wysyła kolejnych komunikatów do trwałego subskrybenta.

Aby zmienić istniejącą subskrypcję, aplikacja może utworzyć trwały subskrybent przy użyciu tej samej nazwy subskrypcji i identyfikatora klienta, ale podając inny temat lub selektor komunikatów (lub oba te elementy). Zmiana trwałej subskrypcji jest równoznaczna z usunięciem subskrypcji i utworzeniem nowej subskrypcji.

W przypadku aplikacji, która łączy się z menedżerem kolejek produktu IBM WebSphere MQ 7.0 , produkt XMS zarządza kolejkami subskrybentów. Dlatego aplikacja nie jest wymagana do określania kolejki subskrybenta. Produkt XMS zignoruje kolejkę subskrybenta, jeśli zostanie określona.

Jednak w przypadku aplikacji, która łączy się z menedżerem kolejek produktu IBM WebSphere MQ 6.0 , każdy trwały subskrybent musi mieć wyznaczoną kolejkę subskrybenta. Aby określić nazwę kolejki subskrybenta dla tematu, należy ustawić właściwość XMSC_WMQ_DUR_SUBQ dla obiektu docelowego reprezentującego dany temat. Domyślną kolejką subskrybenta jest SYSTEM.JMS.D.SUBSCRIBER.QUEUE.

Trwałe subskrybenci łączący się z menedżerami kolejek produktu IBM WebSphere MQ 6.0 mogą współużytkować pojedynczą kolejkę subskrybenta lub każdy trwały subskrybent może pobierać swoje komunikaty z własnej, wyłącznej kolejki subskrybenta. Aby zapoznać się z podejściem do przyjęcia dla danej aplikacji, należy zapoznać się z *XMSIBM WebSphere MQ Using Java*.

Należy pamiętać, że nie można zmienić kolejki subskrybenta dla trwałej subskrypcji. Jedynym sposobem, aby zmienić kolejkę subskrybenta, jest usunięcie subskrypcji i utworzenie nowej.

W przypadku aplikacji, która łączy się z magistralą integracji usług, każdy trwały subskrybent musi mieć wyznaczoną stałą subskrypcję trwałą. Aby określić katalog główny trwałej subskrypcji dla wszystkich stałych subskrybentów, które korzystają z tego samego połączenia, należy ustawić właściwość XMSC_WPM_DUR_SUB_HOME obiektu ConnectionFactory , który jest używany do tworzenia połączenia.

Aby określić katalog główny trwałej subskrypcji dla danego tematu, należy ustawić właściwość XMSC_WPM_DUR_SUB_HOME obiektu docelowego reprezentującego dany temat. W przypadku połączenia przed utworzeniem trwałego subskrybenta, który korzysta z połączenia, musi zostać określony dom subskrypcji trwałej. Każda wartość określona dla miejsca docelowego przesłania wartość określoną dla połączenia.

Nietrwałe subskrybenty

Nietrwały subskrybent to konsument komunikatów, który odbiera tylko komunikaty, które są publikowane w czasie, gdy subskrybent jest aktywny. Komunikaty dostarczone w czasie, gdy subskrybent jest nieaktywny, są tracone.

Informacje zawarte w tym temacie są istotne tylko wtedy, gdy przesyłanie komunikatów w trybie publikowania/subskrypcji jest używane przez menedżer kolejek produktu IBM WebSphere MQ 6.0 .

Jeśli obiekty konsumenta nie zostaną usunięte przed lub w trakcie zamykania połączenia, komunikaty mogą pozostać w kolejkach brokera dla subskrybentów, które nie są już aktywne.

W takiej sytuacji kolejki te mogą zostać usunięte z tych komunikatów za pomocą programu narzędziowego do czyszczenia udostępnionego z klasami IBM WebSphere MQ classes for JMS dla usługi JMS. Szczegółowe informacje na temat korzystania z tego programu narzędziowego znajdują się w sekcji *IBM WebSphere MQ Using Java.* W przypadku dużej liczby komunikatów znajdujących się w tej kolejce konieczne może być również zwiększenie głębokości kolejki subskrybentów.

Synchroniczny konsument komunikatów

Synchroniczny konsument komunikatów odbiera komunikaty z kolejki synchronicznie.

Konsument komunikatów synchronicznych odbiera jeden komunikat w danym momencie. Gdy używana jest metoda `Receive(wait interval)` , wywołanie oczekuje tylko określonego czasu (w milisekundach) dla komunikatu lub do momentu zamknięcia konsumenta komunikatu.

Jeśli używana jest metoda oczekiwania ReceiveNoWait (), konsument komunikatów synchronicznych odbiera komunikaty bez opóźnienia. Jeśli następny komunikat jest dostępny, zostanie odebrany natychmiast, w przeciwnym razie zostanie zwrócony wskaźnik do obiektu komunikatu o wartości NULL.

Konsumenty komunikatów asynchronicznych

Asynchroniczny konsument komunikatów odbiera komunikat z kolejki asynchronicznie. Obiekt nasłuchiwania komunikatów zarejestrowany przez aplikację jest wywoływany za każdym razem, gdy w kolejce jest dostępny nowy komunikat.

Komunikaty nieprzetwarzalne XMS

Komunikat nieprzetwarzalny to komunikat, który nie może być przetwarzany przez odbierającą aplikację MDB. Jeśli zostanie napotkany komunikat nieprzetwarzalny, obiekt XMS MessageConsumer może go ponownie przekwalifikować zgodnie z dwoma właściwościami kolejki: BOQUEUE i BOTHRESH.

W niektórych przypadkach komunikat dostarczony do komponentu MDB może zostać wycofany w kolejce produktu IBM MQ . Może się to zdarzyć, na przykład, jeśli komunikat jest dostarczany w ramach jednostki pracy, która jest następnie wycofana. Komunikat, który jest wycofany, jest zwykle ponownie dostarczany, ale niepoprawnie sformatowany komunikat może spowodować, że komponent MDB nie powiedzie się i dlatego nie może zostać dostarczony. Taki komunikat jest nazywany komunikatem nieprzetwarzonym. Produkt IBM MQ można skonfigurować w taki sposób, aby komunikat nieprzetwarzalny był automatycznie przesyłany do innej kolejki w celu dalszego badania lub został odrzucony. Więcej informacji na temat sposobu konfigurowania produktu IBM MQ w ten sposób zawiera sekcja Obsługa komunikatów nieprzetwarzalnych w ASF.

Czasami w kolejce pojawia się źle sformatowany komunikat. W tym kontekście niepoprawnie sformatowany komunikat oznacza, że aplikacja odbierający nie może poprawnie przetworzyć komunikatu. Taki komunikat może spowodować, że aplikacja odbierający nie powiedzie się, a następnie wycofa się ze źle sformatowanego komunikatu. Następnie komunikat może być wielokrotnie dostarczany do kolejki wejściowej i wielokrotnie wycofany przez aplikację. Te komunikaty są znane jako komunikaty nieprzetwarzalne. Obiekt XMS MessageConsumer wykrywa komunikaty nieprzetwarzalne i przekierowuje je do alternatywnego miejsca docelowego.

Menedżer kolejek produktu IBM MQ przechowuje rekord, ile razy każdy komunikat został wycofany. Gdy ta liczba osiągnie konfigurowalną wartość progową, konsument komunikatu ponownie przeczy komunikat do nazwanej kolejki wycofania. Jeśli ponowne wykonanie nie powiedzie się z jakiegokolwiek powodu, komunikat zostanie usunięty z kolejki wejściowej i zostanie ponownie wysłany do kolejki niedostarczonych komunikatów lub usunięty.

Obiekty XMS ConnectionConsumer obsługują komunikaty nieprzetwarzalne w ten sam sposób i korzystają z tych samych właściwości kolejki. Jeśli wiele konsumentów połączeń monitoruje tę samą kolejkę, możliwe jest, że komunikat nieprzetwarzalny może zostać dostarczony do aplikacji więcej razy niż wartość progowa, zanim nastąpi ponowne wystąpienie kolejki. Takie zachowanie jest spowodowane sposobem, w jaki klienci połączenia indywidualnego monitorują kolejki i komunikaty nieprzetwarzalne w kolejce.

Wartość progowa i nazwa kolejki zaplecza są atrybutami kolejki produktu IBM MQ . Nazwy atrybutów to BackoutThreshold i BackoutRequeueQName. Kolejka, do której mają zastosowanie, jest następująca:

- W przypadku przesyłania komunikatów w trybie punkt z punktem jest to podstawowa kolejka lokalna. Jest to ważne, gdy konsumenci komunikatów i konsumenci połączą się z aliasami kolejek.
- W przypadku przesyłania komunikatów w trybie publikowania/subskrypcji w trybie normalnym dostawcy przesyłania komunikatów produktu IBM MQ jest to kolejka modelowa, z której tworzona jest kolejka zarządzana tematu.
- W przypadku przesyłania komunikatów publikowania/subskrypcji w trybie migracji dostawcy przesyłania komunikatów produktu IBM MQ jest to kolejka CCSUB zdefiniowana w obiekcie fabryki TopicConnectionlub w kolejce CCDSUB zdefiniowanej w obiekcie tematu.

Aby ustawić atrybuty nazwy QName BackoutThreshold i BackoutRequeue, należy wprowadzić następującą komendę MQSC:

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

W przypadku przesyłania komunikatów w trybie publikowania/subskrypcji, jeśli system tworzy kolejkę dynamiczną dla każdej subskrypcji, te wartości atrybutów są uzyskiwane z klas WebSphere MQ dla kolejki modelowej JMS, SYSTEM.JMS.MODEL.QUEUE. Aby zmienić te ustawienia, należy użyć następującej komendy:

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

Jeśli wartość progowa wycofania jest równa zero, obsługa komunikatów nieprzetwarzalnych jest wyłączona, a komunikaty nieprzetwarzalne pozostają w kolejce wejściowej. W przeciwnym razie, gdy liczba wycofań osiągnie wartość progową, komunikat jest wysyłany do nazwanej kolejki wycofanych komunikatów.

Jeśli licznik wycofań osiągnie wartość progową, ale komunikat nie może przejść do kolejki wycofania, komunikat jest wysyłany do kolejki niedostarczonych komunikatów lub, jeśli komunikat jest nietrwały, jest usuwany.

Taka sytuacja występuje wtedy, gdy kolejka wycofania nie została zdefiniowana lub jeśli obiekt MessageConsumer nie może wysłać komunikatu do kolejki wycofania.

Konfigurowanie systemu w celu przeprowadzenia obsługi komunikatów nieprzetwarzalnych

Kolejka używana przez produkt XMS .NET podczas uzyskiwania informacji o atrybutach **BOTHRESH** i **BOQNAME** zależy od stylu wykonywanego przesyłania komunikatów:

- W przypadku przesyłania komunikatów w trybie punkt z punktem jest to podstawowa kolejka lokalna. Jest to ważne, gdy aplikacja XMS .NET konsumuje komunikaty z kolejek aliasowych lub kolejek klastra.

- W przypadku przesyłania komunikatów w trybie publikowania/subskrypcji tworzona jest kolejka zarządzana w celu przechowywania komunikatów dla aplikacji. Program XMS .NET wysyła zapytania do zarządzanej kolejki w celu określenia wartości atrybutów **BOTHRESH** i **BOQNAME** .

  Kolejka zarządzana jest tworzona z kolejki modelowej powiązanej z obiektem tematu, który został zasubskrybowany przez aplikację, a także dziedziczy wartości atrybutów **BOTHRESH** i **BOQNAME** z kolejki modelowej. Używana kolejka modelowa jest zależna od tego, czy aplikacja odbierający podjęła trwałą lub nietrwałą subskrypcję:

  – Kolejka modelowa używana dla trwałych subskrypcji jest określona przez atrybut **MDURMDL** tematu. Wartością domyślną tego atrybutu jest `SYSTEM.DURABLE.MODEL.QUEUE`.
  – W przypadku subskrypcji nietrwałych używana jest kolejka modelowa, która jest używana przez atrybut **MNDURMDL** . Wartością domyślną atrybutu **MNDURMDL** jest `SYSTEM.NDURABLE.MODEL.QUEUE`.

Podczas uzyskiwania informacji o atrybutach **BOTHRESH** i **BOQNAME** , XMS .NET:

- Otwiera kolejkę lokalną lub kolejkę docelową dla kolejki aliasowej.
- Służy do sprawdzania atrybutów **BOTHRESH** i **BOQNAME** .
- Zamyka kolejkę lokalną lub kolejkę docelową dla kolejki aliasowej.

Opcje otwierania używane podczas otwierania kolejki lokalnej lub kolejki docelowej dla kolejki aliasowej są zależne od używanej wersji produktu IBM MQ :

- W przypadku bazy danych IBM MQ 9.0.0 Fix Pack 8i wcześniejszych, jeśli kolejka lokalna lub kolejka docelowa dla kolejki aliasowej jest kolejką klastra, program XMS .NET otwiera kolejkę przy użyciu opcji `MQOO_INPUT_AS_Q_DEF`, `MQOO_INQUIRE` i `MQOO_FAIL_IF_QUIESCING` . Oznacza to, że użytkownik uruchamiający aplikację odbierającą musi uzyskać dostęp do lokalnej instancji kolejki klastra i uzyskać dostęp do niej.

  Program XMS .NET otwiera wszystkie pozostałe typy kolejek lokalnych z opcjami otwartymi `MQOO_INQUIRE` i `MQOO_FAIL_IF_QUIESCING`. Aby program XMS .NET wykonał zapytania o wartości atrybutów, użytkownik uruchamiający aplikację odbierającą musi mieć dostęp do kolejki lokalnej, aby uzyskać dostęp do informacji o dostępie do niej.
- V 9.0.0.9 W przypadku korzystania z produktu XMS .NET z programu IBM MQ 9.0.0 Fix Pack 9użytkownik uruchamiający aplikację odbierającą musi mieć dostęp do kolejki lokalnej, niezależnie od typu kolejki.

Aby przenieść komunikaty nieprzetwarzalne do kolejki wycofanych komunikatów lub kolejki niewysłanych komunikatów menedżera kolejek, należy nadać użytkownikowi uprawnienia do uruchamiania uprawnień `put` i `passall` aplikacji.

Obsługa komunikatów nieprzetwarzalnych w ASF
Jeśli używane są narzędzia Application Server Facilities (ASF), ConnectionConsumer, a nie MessageConsumer, przetwarza komunikaty nieprzetwarzalne. Właściwości ConnectionConsumer są requiem komunikatów zgodnie z właściwościami QName BackoutThreshold i BackoutRequeuekolejki.

Jeśli aplikacja korzysta z opcji ConnectionConsumers, okoliczności, w których jest tworzona kopia zapasowa komunikatu, zależą od sesji, którą udostępnia serwer aplikacji:

- Gdy sesja jest nietransakowana, z AUTO_ACKNOWLEDGE lub DUPS_OK_ACKNOWLEDGE, komunikat jest wycofany tylko po wystąpieniu błędu systemowego lub nieoczekiwanie kończy działanie aplikacji.
- Gdy sesja nie jest transakowana za pomocą funkcji CLIENT_ACKNOWLEDGE, niepotwierdzone komunikaty mogą być wycofane przez serwer aplikacji wywołujący `Session.recover()`.

  Zazwyczaj implementacja klienta MessageListener lub serwer aplikacji wywołuje `Message.acknowledge()`. Program `Message.acknowledge()` potwierdza wszystkie komunikaty dostarczone do tej pory w sesji.
- Gdy sesja jest transmiana, komunikaty niepotwierdzone mogą być wycofane przez serwer aplikacji wywołujący `Session.rollback()`.

Przeglądarki kolejek

Aplikacja korzysta z przeglądarki kolejek w celu przeglądania komunikatów w kolejce bez usuwania ich.

Aby utworzyć przeglądarkę kolejek, aplikacja wywołuje metodę Create Queue Browser dla obiektu ISession, określając jako parametr obiekt docelowy, który identyfikuje kolejkę do przeglądania. Aplikacja może utworzyć przeglądarkę kolejek z selektorem komunikatów lub bez niego.

Po utworzeniu przeglądarki kolejki aplikacja może wywołać metodę GetEnumerator obiektu IQueueBrowser w celu uzyskania listy komunikatów w kolejce. Metoda zwraca obiekt wyliczeniowy, który hermetykuje listę obiektów komunikatu. Kolejność obiektów komunikatu na liście jest taka sama, jak kolejność, w jakiej komunikaty będą pobierane z kolejki. Aplikacja może następnie użyć wyliczenia, aby przejrzeć każdy komunikat z kolei.

Program wyliczeniowy jest aktualizowany dynamicznie, ponieważ komunikaty są umieszczane w kolejce i usuwane z kolejki. Za każdym razem, gdy aplikacja wywołuje komendę IEnumerator.MoveNext() w celu przeglądania następnego komunikatu w kolejce, komunikat ten odzwierciedla bieżącą zawartość kolejki.

Aplikacja może wywołać metodę GetEnumerator więcej niż jeden raz dla danej przeglądarki kolejki. Każde wywołanie zwraca nowy obiekt wyliczeniowy. W związku z tym aplikacja może używać więcej niż jednego wyliczenia do przeglądania komunikatów w kolejce i obsługi wielu pozycji w kolejce.

Aplikacja może użyć przeglądarki kolejek w celu wyszukania odpowiedniego komunikatu do usunięcia z kolejki, a następnie użyć konsumenta komunikatów z selektorem komunikatów, aby usunąć ten komunikat. Selektor komunikatów może wybrać komunikat zgodnie z wartością pola nagłówka JMSMessageID . Więcej informacji na temat tego i innych pól nagłówka komunikatu JMS zawiera sekcja “Pola nagłówka w komunikacie XMS” na stronie 72.

Requestery

Aplikacja korzysta z requestera w celu wysłania komunikatu żądania, a następnie oczekiwania na odebranie odpowiedzi i odebrania odpowiedzi.

Wiele aplikacji przesyłania komunikatów jest opartych na algorytmach, które wysyłają komunikat żądania, a następnie oczekują na odpowiedź. Produkt XMS udostępnia klasę o nazwie Requestor, która pomaga w opracowaniu tego stylu aplikacji.

W celu utworzenia requestera aplikacja wywołuje konstruktor żądania Create Requestor klasy Requestor, określając jako parametry obiekt Session i obiekt docelowy, który identyfikuje miejsca, w których mają być wysyłane komunikaty żądań. Sesja nie może być transakcjaowana ani nie ma trybu potwierdzania XMSC_CLIENT_ACKNOWLEDGE. Konstruktor automatycznie tworzy tymczasową kolejkę lub temat, w którym mają być wysyłane komunikaty odpowiedzi.

Po utworzeniu requestera aplikacja może wywołać metodę żądania obiektu Requestor w celu wysłania komunikatu żądania, a następnie oczekiwania na odpowiedź i odebranie odpowiedzi z aplikacji, która odbiera komunikat z żądaniem. Połączenie oczekuje na odebraną odpowiedź lub do momentu zakończenia sesji, w zależności od tego, co nastąpi wcześniej. Zamawiający wymaga tylko jednej odpowiedzi dla każdego komunikatu żądania.

Po zamknięciu przez aplikację requestera usuwana jest kolejka tymczasowa lub temat. Jednak powiązana sesja nie jest zamykana.

Usuwanie obiektu

Gdy aplikacja usunie utworzony obiekt XMS , produkt XMS zwalnia zasoby wewnętrzne, które zostały przydzielone do obiektu.

Gdy aplikacja tworzy obiekt XMS , produkt XMS przydziela pamięć i inne zasoby wewnętrzne do obiektu. Produkt XMS zachowuje te zasoby wewnętrzne do czasu, aż aplikacja jawnie usunie obiekt, wywołując metodę close lub delete obiektu, w którym punkt XMS zwalnia zasoby wewnętrzne. Jeśli aplikacja próbuje usunąć obiekt, który jest już usunięty, wywołanie jest ignorowane.

Gdy aplikacja usunie obiekt typu Połączenie lub Sesja, program XMS automatycznie usuwa niektóre powiązane obiekty i zwalnia ich zasoby wewnętrzne. Są to obiekty, które zostały utworzone przez obiekt

połączenia lub sesji i nie mają funkcji niezależnie od obiektu. Obiekty te są wyświetlane w programie Tabela 4 na stronie 40.

Uwaga: Jeśli aplikacja zamknie połączenie z sesjami zależnymi, wszystkie obiekty zależne od tych sesji również zostaną usunięte. Obiekty zależne mogą być zależne tylko od obiektu połączenia lub sesji.

Tabela 4. Obiekty, które są automatycznie usuwane

Usunięty obiekt	**Metoda**	**Obiekty zależne, które są usuwane automatycznie**
Połączenie	Zamknij połączenie	Obiekty danych i sesji ConnectionMeta
Sesja	Zamknij sesję	Obiekty MessageConsumer, MessageProducer, QueueBrowseri Requestor

Zarządzane transakcje XA IBM MQ za pomocą XMS

Zarządzane transakcje XA IBM MQ mogą być używane przez system XMS.

Aby można było używać transakcji XA za pomocą XMS, należy utworzyć sesję transakcyjne. Gdy transakcja XA jest używana, sterowanie transakcjami odbywa się za pośrednictwem transakcji globalnych rozproszonego koordynatora transakcji (Distributed Transaction Coordinator-DTC) i nie jest to prawda, ale sesje XMS . W przypadku korzystania z transakcji XA nie można wydać komendy `Session.commit` lub `Session.rollback` w sesji XMS . Zamiast tego należy użyć metod `Transscope.Commit` lub `Transscope.Rollback` DTC w celu zatwierdzenia lub wycofania transakcji. Jeśli sesja jest używana dla transakcji XA, producent lub konsument, który został utworzony za pomocą tej sesji, musi być częścią transakcji XA. Nie można ich używać dla żadnych operacji poza zasięgiem transakcji XA. Nie można ich używać dla operacji, takich jak `Producer.send` lub `Consumer.receive` poza transakcją XA.

Zgłaszany jest obiekt wyjątku `IllegalStateException` , jeśli

- Sesja transakcyjne XA jest używana dla produktu `Session.commit` lub `Session.rollback`.
- Obiekty producenta lub konsumenta, które są kiedyś używane w sesji transakcyjnej XA, są używane po stronie zasięgu transakcji XA.

Transakcje XA nie są obsługiwane w asynchronicznych konsumentach.

Uwaga:

1. Przed zatwierdzeniem transakcji XA możliwe jest wydanie zamknięcia na obiekcie `Producer`, `Consumer`, `Session`lub `Connection` . W takim przypadku komunikaty w transakcji są wycofywane. Podobnie, jeśli połączenie zostało zerwane przed zatwierdzeniem transakcji XA, wszystkie komunikaty w transakcji są wycofywane. W przypadku obiektu `Producer` wycofanie oznacza, że komunikaty nie są umieszczane w kolejce. W przypadku obiektu `Consumer` wycofanie oznacza, że komunikaty pozostają w kolejce.
2. Jeśli obiekt `Producer` wstawi komunikat `TimeToLive` (Czas życia) w `TransactionScope` , a `commit` zostanie wystawiony po upłynięciu czasu, komunikat może utracyć ważność przed wydaniem `commit` . W tym przypadku komunikat nie jest dostępny dla obiektów `Consumer` .
3. Obiekty `Session` nie są obsługiwane przez wątki. Użycie transakcji z obiektami `Session` , które są współużytkowane przez wątki, nie jest obsługiwane.

Typy podstawowe produktu XMS

Produkt XMS udostępnia odpowiedniki ośmiu typów podstawowych Java (byte, short, int, long, float, double, char i boolean). Pozwala to na wymianę komunikatów między XMS a JMS bez utraty lub uszkodzenia danych.

Tabela 5 na stronie 41 zawiera Java równoważny typ danych, wielkość oraz minimalną i maksymalną wartość każdego typu podstawowego XMS .

Tabela 5. Typy danych XMS i ich odpowiedniki Java				
Typ danych XMS	**Zgodny typ danychJava**	**Wielkość**	**Wartość minimalna**	**Maksymalna wartość**
System.Boolean	boolean (boolowskie)	32 bity	Fałsz	Prawda
System.SBYTE	B	8 bitów	-2^7 (-128)	2^7-1 (127)
System.BYTE	B	8 bitów	-2^7 (-128)	2^7-1 (127)
System.CHAR	B	8 bitów	-2^7 (-128)	2^7-1 (127)
System.Int16	short	16 bitów	-2^{15} (-32768)	2^{15}-1 (32767)
System.Int32	int	32 bity	-2^{31} (-2147483648)	2^{31}-1 (2147483647)
System.Int64	long	64 bity	-2^{63} (-9223372036854775808)	2^{63}-1 (9223372036854775807)
System.Single	liczba zmiennopozycyjna	32 bity	-3.402823E-38 (z dokładnością do 7 cyfr)	3.402823E+38 (z dokładnością do 7 cyfr)
System.Double	double (podwójna)	64 bity	-1.79769313486231E-308 (do 15 -cyfrowej precyzji)	1.79769313486231E+308 (do 15 -cyfrowej precyzji)

Pojęcia pokrewne

Atrybuty i właściwości obiektów
Obiekt XMS może mieć atrybuty i właściwości, które są właściwościami obiektu, które są implementowane na różne sposoby.

Niejawne przekształcenie wartości właściwości z jednego typu danych na inny.
Gdy aplikacja pobiera wartość właściwości, wartość może zostać przekształcona przez produkt XMS na inny typ danych. Wiele reguł decyduje o tym, które konwersje są obsługiwane i w jaki sposób program XMS wykonuje konwersje.

Odsyłacze pokrewne

Typy danych dla elementów danych aplikacji
Aby aplikacja XMS mogła wymieniać komunikaty z aplikacją IBM MQ classes for JMS , zarówno aplikacje, jak i aplikacje muszą być w stanie interpretować dane aplikacji w treści komunikatu w ten sam sposób.

Niejawne przekształcenie wartości właściwości z jednego typu danych na inny.

Gdy aplikacja pobiera wartość właściwości, wartość może zostać przekształcona przez produkt XMS na inny typ danych. Wiele reguł decyduje o tym, które konwersje są obsługiwane i w jaki sposób program XMS wykonuje konwersje.

Właściwość obiektu ma nazwę i wartość. Wartość ma powiązany typ danych, w którym wartość właściwości jest określana także jako *typ właściwości* .

Aplikacja korzysta z metod klasy PropertyContext w celu pobrania i ustawienia właściwości obiektów. W celu uzyskania wartości właściwości aplikacja wywołuje metodę odpowiednią dla typu właściwości. Na przykład, aby uzyskać wartość właściwości całkowitej, aplikacja zwykle wywołuje metodę GetIntProperty.

Jeśli jednak aplikacja pobiera wartość właściwości, wartość może zostać przekształcona przez produkt XMS na inny typ danych. Na przykład, aby uzyskać wartość właściwości całkowitej, aplikacja może wywołać metodę GetStringProperty, która zwraca wartość właściwości jako łańcuch. Konwersje obsługiwane przez produkt XMS są wyświetlane w programie Tabela 6 na stronie 42.

Tabela 6. Obsługiwane konwersje z typu właściwości do innych typów danych	
Typ właściwości	**Obsługiwane docelowe typy danych**
System.String	System.Boolean, System.Double, System.Float, System.Int32, System.Int64, System.SByte, System.Int16
System.Boolean	System.String, System.SByte, System.Int32, System.Int64, System.Int16
System.Char	System.String
System.Double	System.String
System.Float	System.String, System.Double
System.Int32	System.String, System.Int64
System.Int64	System.String
System.SByte	System.String, System.Int32, System.Int64, System.Int16
Tablica System.SByte	System.String
System.Int16	System.String, System.Int32, System.Int64

Obsługiwane konwersje określają następujące reguły ogólne:

- Wartości właściwości liczbowych mogą być przekształcane z jednego typu danych na inny, pod warunkiem, że podczas konwersji nie są tracone żadne dane. Na przykład wartość właściwości o typie danych System.Int32 może zostać przekształcona w wartość o typie danych System.Int64, ale nie może zostać przekształcona w wartość o typie danych System.Int16.
- Wartość właściwości dowolnego typu danych może zostać przekształcona w łańcuch.
- Wartość właściwości łańcuchowej może zostać przekształcona w dowolny inny typ danych pod warunkiem, że łańcuch zostanie poprawnie sformatowany w celu konwersji. Jeśli aplikacja podejmie próbę przekształcenia wartości właściwości łańcuchowej, która nie jest poprawnie sformatowana, program XMS może zwrócić błędy.
- Jeśli aplikacja podejmie próbę konwersji, która nie jest obsługiwana, program XMS może zwrócić błąd.

Następujące reguły mają zastosowanie, gdy wartość właściwości jest przekształcana z jednego typu danych na inny:

- Podczas konwersji wartości właściwości boolowskiej na łańcuch wartość true jest przekształcana w łańcuch "true", a wartość false jest przekształcana w łańcuch "false".
- Podczas konwersji wartości właściwości boolowskiej na liczbowy typ danych, w tym System.SByte, wartość true jest przekształcana na 1, a wartość false jest przekształcana na 0.
- Podczas przekształcania wartości właściwości łańcuchowej na wartość boolowskim łańcuch "true" (bez rozróżniania wielkości liter) lub "1" jest przekształcany na wartość true, a łańcuch "false" (bez rozróżniania wielkości liter) lub "0" jest przekształcany na wartość false. Nie można przekształcić wszystkich pozostałych łańcuchów.
- Podczas konwersji wartości właściwości łańcuchowej na wartość o typie danych System.Int32, System.Int64, System.SBytelub System.Int16łańcuch musi mieć następujący format:

  [*odstępy*] [*znak*]*cyfry*

  Komponenty łańcuchowe są definiowane w następujący sposób:

  wartości puste
  Opcjonalne wiodące puste znaki.

  znak
  Opcjonalny znak plus (+) lub znak minus (-).

  cyfry
  Ciągła sekwencja znaków cyfr (0-9). Musi istnieć co najmniej jeden znak cyfry.

Po sekwencji znaków cyfr łańcuch może zawierać inne znaki, które nie są znakami cyfr, ale konwersja zatrzymuje się, gdy tylko pierwszy z tych znaków zostanie osiągnięty. Przyjmuje się, że łańcuch reprezentuje dziesiętną liczbę całkowitą.

XMS może zwrócić błąd, jeśli łańcuch nie jest poprawnie sformatowany.

- Podczas przekształcania wartości właściwości łańcuchowej na wartość typu danych System.Double lub System.Floatłańcuch musi mieć następujący format:

  [*odstępy*] [*znak*] [*cyfry*] [*punkt*[*d_cyfry*]] [*znak e_char*[*znak*]*cyfra_cyfry*]

  Komponenty łańcuchowe są definiowane w następujący sposób:

  wartości puste
  (Opcjonalnie) Leading pustych znaków.

  znak
  (Opcjonalnie) Znak plus (+) lub znak minus (-).

  cyfry
  Ciągła sekwencja znaków cyfr (0-9). Co najmniej jedna cyfra musi być obecna w postaci *cyfr* lub *d_cyfr*.

  punkt
  (Opcjonalne) Punkt dziesiętny (.).

  d_cyfry
  Ciągła sekwencja znaków cyfr (0-9). Co najmniej jedna cyfra musi być obecna w postaci *cyfr* lub *d_cyfr*.

  znak
  Znak wykładnika, który ma wartość *E* lub *e*.

  znak
  (Opcjonalnie) Znak plus (+) lub znak minus (-) dla wykładnika.

  _cyfry
  Ciągła sekwencja znaków cyfr (0-9) dla wykładnika. Jeśli łańcuch zawiera znak wykładnika, musi być obecny co najmniej jeden znak cyfry.

  Po sekwencji znaków cyfr lub opcjonalnych znaków reprezentujących wykładnik, łańcuch może zawierać inne znaki, które nie są znakami cyfr, ale konwersja zatrzymuje się, gdy tylko pierwszy z tych znaków zostanie osiągnięty. Przyjmuje się, że łańcuch reprezentuje liczbę dziesiętną zmiennopozycyjną z wykładnikiem, który jest potęgową liczbą 10.

  XMS może zwrócić błąd, jeśli łańcuch nie jest poprawnie sformatowany.

- Podczas przekształcania wartości liczbowej właściwości w łańcuch, w tym wartość właściwości o typie danych System.SByte, wartość ta jest przekształcana w łańcuchową reprezentację wartości jako liczby dziesiętnej, a nie łańcucha zawierającego znak ASCII dla tej wartości. Na przykład liczba całkowita 65 jest przekształcana w łańcuch "65", a nie na łańcuch "A".
- Przekształcając wartość właściwości tablicy bajtów na łańcuch, każdy bajt jest przekształcany w 2 znaki szesnastkowe, które reprezentują bajt. Na przykład tablica bajtów {0xF1, 0x12, 0x00, 0xFF} jest konwertowana na łańcuch "F11200FF".

Konwersje z typu właściwości do innych typów danych są obsługiwane przez metody klas właściwości i PropertyContext .

Pojęcia pokrewne

Atrybuty i właściwości obiektów
Obiekt XMS może mieć atrybuty i właściwości, które są właściwościami obiektu, które są implementowane na różne sposoby.

Typy podstawowe produktu XMS
Produkt XMS udostępnia odpowiedniki ośmiu typów podstawowych Java (byte, short, int, long, float, double, char i boolean). Pozwala to na wymianę komunikatów między XMS a JMS bez utraty lub uszkodzenia danych.

Odsyłacze pokrewne

Komunikaty mapy
Treść komunikatu mapy zawiera zestaw par nazwa-wartość, w których każda wartość ma powiązany typ danych.

Komunikaty strumienia
Treść komunikatu strumienia zawiera strumień wartości, w którym każda wartość ma powiązany typ danych.

Typy danych dla elementów danych aplikacji
Aby aplikacja XMS mogła wymieniać komunikaty z aplikacją IBM MQ classes for JMS , zarówno aplikacje, jak i aplikacje muszą być w stanie interpretować dane aplikacji w treści komunikatu w ten sam sposób.

Iteratory

Iterator hermetyzuje listę obiektów i kursor, który utrzymuje bieżącą pozycję na liście. Pojęcie iteratora, jak jest dostępne w produkcie IBM Message Service Client for C/C++ , jest implementowane za pomocą interfejsu IEnumerator w produkcie IBM Message Service Client for .NET.

Gdy tworzony jest iterator, pozycja kursora jest przed pierwszym obiektem. Aplikacja korzysta z iteratora w celu pobrania każdego z nich z kolei.

Klasa Iterator IBM Message Service Client for C/C++ jest równoważna klasie Enumerator w produkcie Java. XMS .NET jest podobny do Java i korzysta z interfejsu IEnumerator.

Aplikacja może używać numeru IEnumerator do wykonywania następujących zadań:

- Aby uzyskać właściwości komunikatu
- Aby uzyskać pary nazwa-wartość w treści komunikatu mapy
- Przeglądanie komunikatów w kolejce
- Aby uzyskać nazwy zdefiniowanych właściwości komunikatu JMS obsługiwanych przez połączenie,

Identyfikatory kodowanego zestawu znaków

W programie XMS .NETwszystkie łańcuchy są przekazywane za pomocą rodzimego łańcucha .NET . Ze względu na to, że jest to kodowanie stałe, nie są wymagane żadne dodatkowe informacje w celu jego interpretacji. Dlatego właściwość XMSC_CLIENT_CCSID nie jest wymagana dla aplikacji XMS .NET .

Kody błędów i wyjątków produktu XMS

Produkt XMS używa szeregu kodów błędów do wskazania niepowodzeń. Te kody błędów nie są jawnie wymienione w tej dokumentacji, ponieważ mogą się one różnić w zależności od wydania do wydania. Tylko kody wyjątków XMS (w formacie XMS_X_ ...) są udokumentowane, ponieważ pozostają one takie same w różnych wersjach produktu XMS.

Budowanie własnych aplikacji

Użytkownik buduje własne aplikacje, takie jak kompilacja przykładowych aplikacji.

Zbuduj aplikację .NET zgodnie z opisem w sekcji “Budowanie przykładowych aplikacji produktu .NET” na stronie 20, która zawiera również listę wymagań wstępnych, które należy spełnić, aby zbudować własne aplikacje produktu .NET . Aby uzyskać dodatkowe wskazówki dotyczące sposobu budowania własnych aplikacji, należy użyć plików makefile udostępnionych dla każdej przykładowej aplikacji.

Wskazówka: Aby ułatwić diagnozowanie problemów w przypadku wystąpienia awarii, pomocne może być skompilowanie aplikacji z dołączonym symbolami.

Pojęcia pokrewne

Przykładowe aplikacje
Przykładowe aplikacje zawierają przegląd wspólnych funkcji każdego interfejsu API. Można ich używać do weryfikowania konfiguracji instalacji i serwera przesyłania komunikatów oraz do budowania własnych aplikacji.

Odsyłacze pokrewne

.NET interfejsy
Ten sekcja dokumentuje interfejsy klasy .NET oraz ich właściwości i metody.

Właściwości obiektów XMS
Ta rozdział dokumentuje właściwości obiektu zdefiniowane przez produkt XMS.

Automatyczne ponowne połączenie klienta IBM MQ za pomocą XMS

Skonfiguruj klienta produktu XMS w taki sposób, aby ponownie nawiązał połączenie po awarii sieci, menedżera kolejek lub serwera podczas korzystania z klienta IBM WebSphere MQ 7.1 , a następnie jako dostawcy komunikatów.

Użyj właściwości `WMQ_CONNECTION_NAME_LIST` i `WMQ_CLIENT_RECONNECT_OPTIONS` klasy `MQConnectionFactory` , aby skonfigurować połączenie klienta w celu automatycznego ponownego połączenia. Automatyczne ponowne połączenie klienta ponownie łączy klienta po awarii połączenia lub jako opcja po zatrzymaniu menedżera kolejek. Projektowanie niektórych aplikacji klienckich sprawia, że nie nadają się one do automatycznego ponownego połączenia.

Po nawiązaniu połączenia automatycznie połączalne połączenia klienckie stają się ponownie połączalne.

Uwaga: Właściwości `Client Reconnect Options`(Opcje ponownego połączenia klienta), `Client Reconnect Timeout`(Limit czasu ponownego połączenia klienta) i `Connection Namelist` (Lista nazw połączeń) można również ustawić za pomocą tabeli definicji kanału klienta (Client Channel Definitions Table-CCDT) lub przez włączenie ponownego połączenia klienta przy użyciu pliku `mqclient.ini` .

Uwaga: Jeśli właściwości ponownego połączenia są ustawione w obiekcie `ConnectionFactory` i podobnie jak w tabeli definicji kanału klienta, reguła kolejności wykonywania jest następująca. Jeśli wartość domyślna właściwości listy nazw połączeń jest ustawiona w obiekcie `ConnectionFactory` , to środowisko CCDT ma pierwszeństwo. Jeśli lista nazw połączeń nie jest ustawiona na wartość domyślną, pierwszeństwo mają wartości właściwości ustawione w obiekcie `ConnectionFactory` . Wartością domyślną listy nazw połączeń jest `localhost(1414)`.

Pisanie aplikacji XMS .NET

Tematy zawarte w tej sekcji zawierają informacje pomocne podczas pisania aplikacji XMS .NET .

Ta sekcja zawiera informacje specyficzne dla tworzenia aplikacji programu XMS .NET . Ogólne informacje na temat pisania aplikacji XMS zawiera sekcja “Pisanie aplikacji produktu XMS” na stronie 21.

Sekcja obejmuje następujące tematy:

Pojęcia pokrewne

Pisanie aplikacji produktu XMS
Tematy w tej sekcji zawierają informacje pomocne podczas pisania aplikacji XMS .

Odsyłacze pokrewne

.NET interfejsy
Ten sekcja dokumentuje interfejsy klasy .NET oraz ich właściwości i metody.

Typy danych dla .NET

Program XMS .NET obsługuje metody 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, i System.Object. Typy danych dla XMS .NET różnią się od typów danych dla XMS C/C + +. Można użyć tego tematu w celu zidentyfikowania odpowiednich typów danych.

W poniższej tabeli przedstawiono odpowiadające im typy danych XMS .NET i XMS C/C + +, a w skrócie opisano je w skrócie.

Tabela 7. Typy danych dla systemów XMS .NET i XMS C/C++

Typ XMS .NET	**XMS Typ C/C++**	**Opis**
System.SByte	xmsSBYTE xmsINT8	Podpisana 8-bitowa wartość
System.Byte	xmsBYTE xmsUINT8	Wartość 8-bitowa bez znaku
System.Int16	xmsINT16 xmsSHORT	16-bitowa wartość ze znakiem
System.UInt16	xmsUINT16 xmsUSHORT	16-bitowa wartość bez znaku
System.Int32	xmsINT32 xmsINT	Podpisana 32-bitowa wartość
System.UInt32	xmsUINT32 xmsUINT	32-bitowa wartość bez znaku
System.Int64	xmsLONG xmsINT64	Podpisana 64-bitowa wartość
System.UInt64	xmsULONG xmsUINT64	64-bitowa wartość bez znaku
System.Char	xmsCHAR16	16-bitowy znak bez znaku (Unicode dla środowiska .NET)
System.Single	xmsFLOAT	Liczba 32-bitowych liczb zmiennopozycyjnych IEEE
System.Double	xmsDOUBLE	Liczba zmiennopozycyjna IEEE 64-bit
System.Boolean	xmsBOOL	Wartość Prawda/Fałsz
Nie dotyczy	xmsCHAR	Podpisana lub niepodpisana wartość 8-bitowa (podpisana lub niepodpisana zależy od platformy)
System.Decimal	Nie dotyczy	96-bit podpisanych liczb całkowitych od 10^0 do 10^{28}
System.Object	Nie dotyczy	Podstawa wszystkich typów
System.String	Nie dotyczy	Typ łańcuchowy

Operacje zarządzane i niezarządzane w programie .NET

Kod zarządzany jest wykonywany wyłącznie w środowisku wykonawczym języka wspólnego produktu .NET i jest w pełni zależny od usług udostępnianych przez to środowisko wykonawcze. Aplikacja jest klasowana jako niezarządzana, jeśli dowolna część aplikacji działa lub wywołuje usługi poza środowiskiem wykonawczym wspólnego języka produktu .NET .

Niektóre zaawansowane funkcje nie mogą obecnie być obsługiwane w zarządzanym środowisku produktu .NET .

Jeśli aplikacja wymaga pewnych funkcji, które nie są obecnie obsługiwane w pełnym środowisku zarządzanym, można zmienić aplikację tak, aby korzystała z niezarządzanego środowiska bez konieczności wprowadzania istotnych zmian w aplikacji. Należy jednak pamiętać, że stos XMS korzysta z kodu niezarządzanego po dokonaniu wyboru.

Połączenia z menedżerem kolejek produktu IBM MQ

Połączenia zarządzane z programem WMQ_CM_CLIENT nie obsługują komunikacji innej niż TCP, a kompresja kanału. Połączenia te mogą jednak być obsługiwane przy użyciu połączenia niezarządzanego (WMQ_CM_CLIENT_UNMANAGED). Więcej informacji na ten temat zawiera sekcja Tworzenie aplikacji .NET.

Jeśli fabryka połączeń zostanie utworzona z obiektu administrowanego w środowisku niezarządzanym, należy ręcznie zmienić wartość dla trybu połączenia na XMSC_WMQ_CM_CLIENT_UNMANAGED.

Połączenia z mechanizmem przesyłania komunikatów magistrali integracji usług produktu WebSphere Application Server

Połączenia z mechanizmem przesyłania komunikatów magistrali integracji usług produktu WebSphere Application Server , które wymagają użycia protokołu SSL (w tym protokołu HTTPS), nie są obecnie obsługiwane jako kod zarządzany.

Odsyłacze pokrewne

XMSC_WMQ_CONNECTION_MODE

Miejsca docelowe w środowisku .NET

W programie .NETmiejsca docelowe są tworzone zgodnie z typem protokołu i mogą być używane tylko w przypadku typu protokołu, dla którego zostały utworzone.

Udostępniono dwie funkcje służące do tworzenia miejsc docelowych, jeden dla tematów i jeden dla kolejek:

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

Funkcje te są dostępne w następujących dwóch obiektach interfejsu API:

- ISesja
- XMSFactoryFactory

W obu przypadkach metody te mogą akceptować łańcuch w stylu URI, który może zawierać parametry w następującym formacie:

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

Alternatywnie metody te mogą akceptować tylko nazwę miejsca docelowego, to znaczy nazwę bez tematu: // lub kolejkę: // przedrostek i bez parametrów.

Oznacza to, że następujący łańcuch stylu URI:

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

spowodowałoby taki sam wynik, jak następująca nazwa docelowa:

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

Podobnie jak w przypadku WebSphere Application Server magistrali integracji usług JMS, tematy mogą być również określane w postaci skróconych, która zawiera zarówno *topicname* , jak i *topicspace* , ale nie może zawierać parametrów:

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

Właściwości w programie .NET

Aplikacja .NET korzysta z metod w interfejsie PropertyContext do pobierania i ustawiania właściwości obiektów.

Interfejs PropertyContext hermetyzuje metody, które dostają i ustawiają właściwości. Metody te są dziedziczone, bezpośrednio lub pośrednio, przez następujące klasy:

- BytesMessage
- Połączenie
- ConnectionFactory
- ConnectionMetaDane
- Cel
- MapMessage
- Komunikat
- MessageConsumer
- MessageProducer
- ObjectMessage
- QueueBrowser
- Sesja
- StreamMessage
- TextMessage

Jeśli aplikacja ustawi wartość właściwości, nowa wartość zastępuje poprzednią wartość posiadanej właściwości.

Więcej informacji na temat właściwości produktu XMS zawiera sekcja “Właściwości obiektów XMS” na stronie 181.

W przypadku łatwości używania nazwy właściwości i wartości właściwości XMS w produkcie XMS są predefiniowane jako stałe publiczne w strukturze o nazwie XMSC. Nazwy tych stałych są w formacie XMSC.*stała*, na przykład XMSC.USERID (stała nazwa właściwości) i XMSC.DELIVERY_AS_APP (stała wartość).

Dodatkowo można uzyskać dostęp do stałych IBM MQ za pomocą IBM.XMS.MQC . Jeśli IBM.XMS jest już zaimportowana, można uzyskać dostęp do wartości tych właściwości w postaci MQC.*stała*. Na przykład MQC.MQRO_COA_WITH_FULL_DATA.

Ponadto, jeśli istnieje aplikacja hybrydowa, która używa zarówno klas XMS .NET , jak i IBM MQ dla produktu .NET , a importuje to zarówno IBM.XMS i IBM.WMQ , a następnie należy w pełni kwalifikować przestrzeń nazw struktury MQC, aby upewnić się, że każde wystąpienie jest unikalne.

Niektóre zaawansowane funkcje nie są obecnie obsługiwane w zarządzanym środowisku .NET . Więcej informacji na ten temat zawiera sekcja “Operacje zarządzane i niezarządzane w programie .NET” na stronie 47 .

Obsługa właściwości nieistniejących w produkcie .NET

Obsługa nieistniejących właściwości w XMS .NET jest zasadniczo zgodna ze specyfikacją JMS, a także z implementacjami C i C + + XMS.

W usłudze JMS uzyskiwanie dostępu do nieistniejącej właściwości może spowodować wystąpienie wyjątku systemowego Java , gdy metoda próbuje przekształcić nieistniejącą (null) wartość w wymagany typ. Jeśli właściwość nie istnieje, występują następujące wyjątki:

- Właściwość getStringi właściwość getObjectzwracają wartość NULL
- getBooleanWłaściwość zwraca wartość false, ponieważ wartość Boolean.valueOf(null) zwraca wartość false.
- getIntWłaściwość etc.throw java.lang.NumberFormatException , ponieważ metoda Integer.valueOf(null) zgłasza wyjątek

Jeśli właściwość nie istnieje w programie XMS .NET , występują następujące wyjątki:

- Właściwość GetStringi właściwość GetObject(i właściwość GetBytes) zwracają wartość NULL (co jest takie samo, jak właściwość Java).
- GetBooleanWłaściwość zgłasza wyjątek System.NullReferenceException
- Właściwość GetIntpowoduje zgłoszenie wyjątku System.NullReferenceException .

Ta implementacja różni się od Java, ale jest ona ogólnie spójna ze specyfikacją JMS oraz z interfejsami XMS C i C++. Podobnie jak w przypadku implementacji Java , program XMS .NET propaguje wszystkie wyjątki z wywołania System.Convert do programu wywołującego. Jednak w przeciwieństwie do produktu Java , produkt XMS jawnie zgłasza wyjątki NullReference, a nie tylko za pomocą rodzimego zachowania środowiska .NET przez przekazanie wartości NULL do procedur konwersji systemu. Jeśli aplikacja ustawia właściwość na łańcuch "abc" i wywołuje właściwość GetInt, wyjątek System.FormatException zgłaszany przez wartość Convert.ToInt32("abc") jest propagowany do programu wywołującego, który jest spójny z programem Java. Wyjątek MessageFormatjest zgłaszany tylko wtedy, gdy typy używane dla właściwości setProperty i getProperty są niezgodne. To zachowanie jest również spójne z produktem Java.

Obsługa błędów w produkcie .NET

Wyjątki XMS .NET wywodzi się z wyjątku System.Exception. Wywołania metod XMS mogą zgłaszać konkretne wyjątki XMS , takie jak MessageFormatException, general XMSExceptions lub systemowe wyjątki, takie jak NullReferenceException.

Aplikacje napisz, aby wychwytywać dowolne z tych błędów, albo w określonych blokach catch, albo w ogólnych blokach catch `System.Exception` , zgodnie z wymaganiami aplikacji.

Programy nasłuchujące komunikatów i wyjątków w produkcie .NET

Aplikacja .NET używa obiektu nasłuchiwania komunikatów w celu asynchronicznego odbierania komunikatów i asynchronicznego powiadamiania o problemie z połączeniem korzysta z obiektu nasłuchiwania wyjątków.

Funkcjonalność programów nasłuchujących komunikatów i wyjątków jest taka sama dla produktu .NET , jak i dla języka C + +. Jednak niewielkie różnice w implementacji są niewielkie.

Obiekty nasłuchiwania komunikatów w produkcie .NET

Aby odbierać komunikaty asynchronicznie, należy wykonać następujące kroki:

1. Zdefiniuj metodę, która jest zgodna z sygnaturą delegata programu nasłuchującego komunikatów. Definiowana metoda może być metodą statyczną lub instancją i może być zdefiniowana w dowolnej dostępnej klasie. Podpis delegata jest następujący:

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

i tak można zdefiniować metodę jako:

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

2. Utwórz instancję tej metody jako delegata przy użyciu czegoś podobnego do następującego:

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

3. Zarejestruj delegata z jednym lub większą liczbę konsumentów, ustawiając go w właściwości MessageListener konsumenta:

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

Aby usunąć delegata, należy przywrócić wartość NULL przez ustawienie parametru MessageListener :

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

Obiekty nasłuchiwania wyjątków w środowisku .NET

Obiekt nasłuchiwania wyjątków działa w taki sam sposób, jak program nasłuchujący komunikatów, ale ma inną definicję delegowania i jest przypisywany do połączenia, a nie przez konsumenta komunikatów. Jest to takie samo, jak w przypadku C + +.

1. Zdefiniuj metodę. Podpis delegata jest następujący:

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

i tak zdefiniowana metoda może być:

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

2. Utwórz instancję tej metody jako delegata, używając czegoś podobnego do:

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

3. Zarejestruj delegata za pomocą połączenia, ustawiając jego właściwość ExceptionListener :

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

Istnieje możliwość usunięcia delegata przez zresetowanie obiektu ExceptionListener w celu:

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

Jeśli żadne odwołania do nich nie pozostaną, wyjątki lub komunikaty są automatycznie usuwane przez system czyszczenia pamięci systemów.

Poniżej znajduje się przykładowy kod:

```
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);
    }
}
```

Praca z administrowanymi obiektami

Tematy w tej sekcji zawierają informacje na temat administrowanych obiektów. Aplikacje produktu XMS mogą pobierać definicje obiektów z repozytorium administrowanych obiektów centralnych i używać ich do tworzenia fabryk połączeń i miejsc docelowych.

Ta sekcja zawiera informacje pomocne w tworzeniu obiektów administrowanych i zarządzaniu nimi, opisując typy administrowanych repozytorium obiektów obsługiwanych przez produkt XMS . W tej sekcji wyjaśniono również, w jaki sposób aplikacja XMS nawiąże połączenie z repozytorium obiektów administrowanych w celu pobrania wymaganych obiektów administrowanych.

Sekcja obejmuje następujące tematy:

Pojęcia pokrewne

Administrowane obiekty
Za pomocą administrowanych obiektów można administrować ustawieniami połączenia używalnymi przez aplikacje klienckie, które mają być administrowane z centralnego repozytorium. Aplikacja pobiera definicje obiektów z centralnego repozytorium i używa ich do tworzenia obiektów `ConnectionFactory` i `Destination` . Za pomocą administrowanych obiektów można usunąć kilka aplikacji z zasobów, które są używane w czasie wykonywania.

Obsługiwane typy administrowanego repozytorium obiektów

Obiekty administrowane przez system plików i LDAP mogą być używane do łączenia się z IBM MQ i WebSphere Application Server, podczas gdy nazwy COS mogą być używane tylko do łączenia się z serwerem WebSphere Application Server.

Katalogi obiektów systemu plików przyjmują postać serializowanych obiektów Java Naming Directory Interface (JNDI). Katalogi obiektów LDAP to katalogi, które zawierają obiekty JNDI . Katalogi obiektów systemu plików i LDAP można administrować za pomocą narzędzia JMSAdmin dostarczanego z produktem IBM WebSphere MQ 6.0lub IBM MQ Explorer, który jest dostarczany z produktem IBM WebSphere MQ 7.0 i nowszym. Zarówno system plików, jak i katalogi obiektów LDAP mogą być używane do administrowania połączeniami klientów poprzez scentralizowanie fabryk połączeń i miejsc docelowych produktu IBM WebSphere MQ . Administrator sieci może wdrożyć wiele aplikacji, które odwołują się do tego samego centralnego repozytorium, i które są automatycznie aktualizowane w celu odzwierciedlenia zmian ustawień połączeń wprowadzonych w centralnym repozytorium.

Katalog nazw COS zawiera fabryki połączeń i miejsca docelowe produktu WebSphere Application Server service integration bus i może być administrowany przy użyciu Konsoli administracyjnej serwera WebSphere Application Server . Aby aplikacja XMS mogła pobierać obiekty z katalogu nazw COS, musi zostać wdrożona usługa Web Service wyszukiwania produktu JNDI . Ta usługa Web Service nie jest dostępna na wszystkich serwerach WebSphere Application Server service integration technologies. Szczegółowe informacje można znaleźć w dokumentacji produktu.

Uwaga: Zrestartuj połączenia aplikacji, aby zmiany wprowadzone w katalogu obiektów zostały uwzględnione.

Pojęcia pokrewne

Odwzorowanie właściwości dla administrowanych obiektów
Aby umożliwić aplikacjom korzystanie z fabryki połączeń produktu IBM MQ JMS i WebSphere Application Server oraz definicji obiektów docelowych, właściwości pobrane z tych definicji muszą być odwzorowane na odpowiednie właściwości produktu XMS , które można ustawić w fabrykach połączeń i miejscach docelowych produktu XMS .

Właściwości obiektu InitialContext
Parametry konstruktora InitialContext obejmują położenie repozytorium administrowanych obiektów, podane jako jednolity wskaźnik zasobów (URI). Aby aplikacja mogła nawiązać połączenie z repozytorium, może być konieczne podanie większej ilości informacji niż informacje zawarte w identyfikatorze URI.

Format identyfikatora URI dla kontekstów początkowych produktu XMS
Położenie repozytorium administrowanych obiektów jest udostępniane jako jednolity wskaźnik zasobów (URI). Format identyfikatora URI zależy od typu kontekstu.

Usługa Web Service wyszukiwania JNDI
Aby uzyskać dostęp do katalogu nazw COS z poziomu produktu XMS, usługa Web Service wyszukiwania produktu JNDI musi zostać wdrożona na serwerze WebSphere Application Server service integration bus . Ta usługa Web Service tłumaczy informacje o produkcie Java z usługi nazw COS na formularz, który może być odczytany przez aplikacje produktu XMS .

Pobieranie administrowanych obiektów
Program XMS pobiera obiekt administrowany z repozytorium przy użyciu adresu udostępnionego podczas tworzenia obiektu InitialContext lub w właściwościach InitialContext .

Administrowane obiekty

Za pomocą administrowanych obiektów można administrować ustawieniami połączenia używalnymi przez aplikacje klienckie, które mają być administrowane z centralnego repozytorium. Aplikacja pobiera definicje obiektów z centralnego repozytorium i używa ich do tworzenia obiektów `ConnectionFactory` i `Destination` . Za pomocą administrowanych obiektów można usunąć kilka aplikacji z zasobów, które są używane w czasie wykonywania.

Zadania pokrewne

Tworzenie obiektów administrowanych
Definicje obiektów ConnectionFactory i Destination, które są wymagane przez aplikacje produktu XMS w celu nawiązania połączenia z serwerem przesyłania komunikatów, muszą zostać utworzone przy użyciu odpowiednich narzędzi administracyjnych.

Obiekty InitialContext
Aplikacja musi utworzyć kontekst początkowy, który będzie używany w celu nawiązania połączenia z repozytorium obiektów administrowanych w celu pobrania wymaganych obiektów administrowanych.

Odsyłacze pokrewne

Wymagane właściwości dla administrowanych obiektów ConnectionFactory
Gdy aplikacja tworzy fabrykę połączeń, należy zdefiniować pewną liczbę właściwości, aby utworzyć połączenie z serwerem przesyłania komunikatów.

Wymagane właściwości administrowanych obiektów docelowych
Aplikacja tworzący miejsce docelowe musi ustawić kilka właściwości, które aplikacja ma w administrowanego obiekcie docelowym.

Odwzorowanie właściwości dla administrowanych obiektów

Aby umożliwić aplikacjom korzystanie z fabryki połączeń produktu IBM MQ JMS i WebSphere Application Server oraz definicji obiektów docelowych, właściwości pobrane z tych definicji muszą być odwzorowane na odpowiednie właściwości produktu XMS , które można ustawić w fabrykach połączeń i miejscach docelowych produktu XMS .

Aby utworzyć na przykład fabrykę połączeń produktu XMS z właściwościami pobraną z fabryki połączeń JMS produktu IBM MQ , właściwości muszą być odwzorowane między tymi dwoma właściwościami.

Wszystkie odwzorowania właściwości są wykonywane automatycznie.

W poniższej tabeli przedstawiono odwzorowania między niektórymi najczęściej spotykanych właściwości fabryk połączeń i miejsc docelowych. Właściwości wyświetlane w tej tabeli są tylko małym zestawem przykładów, a nie wszystkie wyświetlane właściwości są istotne dla wszystkich typów połączeń i serwerów.

Tabela 8. Przykłady odwzorowania nazw dla właściwości fabryki połączeń i miejsca docelowego

Nazwa właściwości IBM MQ JMS	**XMS Nazwa właściwości**	**WebSphere Application Server service integration bus Nazwa właściwości**
TRWAŁOŚĆ (PER)	XMSC_DELIVERY_MODE	
TERMIN WAŻNOŚCI (EXP)	XMSC_TIME_TO_LIVE	
PRIORYTET (PRI)	XMSC_PRIORITY	
	XMSC_WPM_HOST_NAME	serverName
	XMSC_WPM_BUS_NAME	busName
	XMSC_WPM_TOPIC_SPACE	topicName

Tabela 9. Właściwości produktu XMS .NET

Właściwość	Typ obiektu				
	CF	**QCF**	**TCF**	**Kolejka**	**Temat**
APPLICATIONNAME	Y	Y	Y	Nie dotyczy	Nie dotyczy
ASYNCEXCEPTION	Y	Y	Y	Nie dotyczy	Nie dotyczy
CCDTURL	Y	Y	Y	Nie dotyczy	Nie dotyczy
CHANNEL	Y	Y	Y	Nie dotyczy	Nie dotyczy
CONNECTIONNAMELIST	Y	Y	Y	Nie dotyczy	Nie dotyczy
CLIENTRECONNECTOPTIONS	Y	Y	Y	Nie dotyczy	Nie dotyczy
CLIENTRECONNECTTIMEOUT	Y	Y	Y	Nie dotyczy	Nie dotyczy
CLIENTID	Nie dotyczy	Y	Nie dotyczy	Nie dotyczy	Nie dotyczy
COMPHDR "1" na stronie 55	Y	Nie dotyczy	Y	Nie dotyczy	Nie dotyczy
COMPMSG "1" na stronie 55	Y	Y	Y	Nie dotyczy	Nie dotyczy
CONNOPT "1" na stronie 55	Y	Y	Y	Nie dotyczy	Nie dotyczy
CONNTAG "1" na stronie 55	Y	Y	Y	Nie dotyczy	Nie dotyczy
OPIS "1" na stronie 55	Nie dotyczy	Y	Nie dotyczy	Y	Y
WAŻNOŚCI "1" na stronie 55	Nie dotyczy	Nie dotyczy	Nie dotyczy	Y	Y
FAILIFQUIESCE	Y	Y	Y	Y	Y
Nazwa hosta	Nie dotyczy	Y	Nie dotyczy	Nie dotyczy	Nie dotyczy
LOCALADDRESS	Nie dotyczy	Y	Nie dotyczy	Nie dotyczy	Nie dotyczy
PERSISTENCE	Nie dotyczy	Nie dotyczy	Nie dotyczy	Y	Y
PORT	Nie dotyczy	Y	Nie dotyczy	Nie dotyczy	Nie dotyczy
Priorytet"1" na stronie 55	Nie dotyczy	Nie dotyczy	Nie dotyczy	Y	Y
PROVIDERVERSION "1" na stronie 55	Nie dotyczy	Y	Nie dotyczy	Nie dotyczy	Nie dotyczy
QMANAGER	Y	Y	Y	Y	Nie dotyczy
kolejka"1" na stronie 55	Nie dotyczy	Nie dotyczy	Nie dotyczy	Y	Nie dotyczy

Tabela 9. Właściwości produktu XMS .NET (kontynuacja)					
Właściwość	**Typ obiektu**				
	CF	**QCF**	**TCF**	**Kolejka**	**Temat**
SHARECONVALLOWED	Y	Y	Y	Nie dotyczy	Nie dotyczy
Temat"1" na stronie 55	Nie dotyczy	Nie dotyczy	Nie dotyczy	Nie dotyczy	Y
TRANSPORT "1" na stronie 55	Nie dotyczy	Y	Nie dotyczy	Nie dotyczy	Nie dotyczy

Uwaga:

1. Te właściwości nie mają właściwości poziomu aplikacji, ale można je opcjonalnie ustawić za pomocą właściwości administrowanych.

Pojęcia pokrewne

Obsługiwane typy administrowanego repozytorium obiektów
Obiekty administrowane przez system plików i LDAP mogą być używane do łączenia się z IBM MQ i WebSphere Application Server, podczas gdy nazwy COS mogą być używane tylko do łączenia się z serwerem WebSphere Application Server.

Właściwości obiektu InitialContext
Parametry konstruktora InitialContext obejmują położenie repozytorium administrowanych obiektów, podane jako jednolity wskaźnik zasobów (URI). Aby aplikacja mogła nawiązać połączenie z repozytorium, może być konieczne podanie większej ilości informacji niż informacje zawarte w identyfikatorze URI.

Format identyfikatora URI dla kontekstów początkowych produktu XMS
Położenie repozytorium administrowanych obiektów jest udostępniane jako jednolity wskaźnik zasobów (URI). Format identyfikatora URI zależy od typu kontekstu.

Usługa Web Service wyszukiwania JNDI
Aby uzyskać dostęp do katalogu nazw COS z poziomu produktu XMS, usługa Web Service wyszukiwania produktu JNDI musi zostać wdrożona na serwerze WebSphere Application Server service integration bus . Ta usługa Web Service tłumaczy informacje o produkcie Java z usługi nazw COS na formularz, który może być odczytany przez aplikacje produktu XMS .

Pobieranie administrowanych obiektów
Program XMS pobiera obiekt administrowany z repozytorium przy użyciu adresu udostępnionego podczas tworzenia obiektu InitialContext lub w właściwościach InitialContext .

Zadania pokrewne

Tworzenie obiektów administrowanych
Definicje obiektów ConnectionFactory i Destination, które są wymagane przez aplikacje produktu XMS w celu nawiązania połączenia z serwerem przesyłania komunikatów, muszą zostać utworzone przy użyciu odpowiednich narzędzi administracyjnych.

Obiekty InitialContext
Aplikacja musi utworzyć kontekst początkowy, który będzie używany w celu nawiązania połączenia z repozytorium obiektów administrowanych w celu pobrania wymaganych obiektów administrowanych.

Odsyłacze pokrewne

Wymagane właściwości dla administrowanych obiektów ConnectionFactory
Gdy aplikacja tworzy fabrykę połączeń, należy zdefiniować pewną liczbę właściwości, aby utworzyć połączenie z serwerem przesyłania komunikatów.

Wymagane właściwości administrowanych obiektów docelowych
Aplikacja tworzący miejsce docelowe musi ustawić kilka właściwości, które aplikacja ma w administrowanego obiekcie docelowym.

IDestination (dla interfejsu .NET)

Miejsce docelowe to miejsce, w którym aplikacja wysyła komunikaty lub jest to źródło, z którego aplikacja odbiera komunikaty, lub oba te komunikaty.

Właściwości miejsca docelowego
Przegląd właściwości obiektu docelowego, z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

IConnectionFactory (dla interfejsu .NET)
Aplikacja korzysta z fabryki połączeń w celu utworzenia połączenia.

Właściwości obiektu ConnectionFactory
Przegląd właściwości obiektu ConnectionFactory z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

Wymagane właściwości dla administrowanych obiektów ConnectionFactory

Gdy aplikacja tworzy fabrykę połączeń, należy zdefiniować pewną liczbę właściwości, aby utworzyć połączenie z serwerem przesyłania komunikatów.

Właściwości wymienione w poniższych tabelach są minimalną wymaganą dla aplikacji, która ma zostać ustawiona w celu utworzenia połączenia z serwerem przesyłania komunikatów. Jeśli chcesz dostosować sposób tworzenia połączenia, aplikacja może w razie potrzeby ustawić dodatkowe właściwości obiektu ConnectionFactory . Więcej informacji na ten temat zawiera sekcja “Właściwości obiektu ConnectionFactory” na stronie 182. Dołączona jest pełna lista dostępnych właściwości.

Połączenie z menedżerem kolejek produktu IBM MQ

Tabela 10. Ustawienia właściwości dla administrowanych obiektów ConnectionFactory dla połączeń z menedżerem kolejek produktu IBM MQ

WymaganeXMS	**Wymagana jest równoważna właściwość IBM MQ JMS**
XMSC_CONNECTION_TYPE	Produkt XMS działa z tym produktem z właściwości Nazwa klasy fabryki połączeń i TRANSPORT (TRAN).
XMSC_WMQ_HOST_NAME	NAZWA HOSTA (HOST)
PORT XMSC_WMQ_PORT	PORT
MENEDŻER KOLEJEK XMSC_WMQ_QUEUE_MANAGER	Nazwa menedżera kolejek

Połączenie w czasie rzeczywistym z brokerem

Tabela 11. Ustawienia właściwości dla administrowanych obiektów ConnectionFactory dla połączeń w czasie rzeczywistym z brokerem

WymaganeXMS	**Wymagana jest równoważna właściwość IBM MQ JMS**
XMSC_CONNECTION_TYPE	Produkt XMS działa z tym produktem z właściwości Nazwa klasy fabryki połączeń i TRANSPORT (TRAN).
XMSC_RTT_HOST_NAME	NAZWA HOSTA (HOST)
PORT XMSC_RTT_PORT	PORT

Połączenie z WebSphere Application Server service integration bus

Tabela 12. Ustawienia właściwości dla administrowanych obiektów ConnectionFactory dla połączeń z serwerem WebSphere Application Server service integration bus

XMS właściwość	Opis
XMSC_CONNECTION_TYPE	Typ serwera przesyłania komunikatów, z którym łączy się aplikacja.. Wartość ta jest określana na podstawie nazwy klasy fabryki połączeń.
XMSC_WPM_BUS_NAME	W przypadku fabryki połączeń: nazwa magistrali integracji usług, z którą aplikacja nawiązuje połączenie. W przypadku miejsca docelowego: nazwa magistrali integracji usług, w której znajduje się miejsce docelowe.

Pojęcia pokrewne

Obsługiwane typy administrowanego repozytorium obiektów
Obiekty administrowane przez system plików i LDAP mogą być używane do łączenia się z IBM MQ i WebSphere Application Server, podczas gdy nazwy COS mogą być używane tylko do łączenia się z serwerem WebSphere Application Server.

Odwzorowanie właściwości dla administrowanych obiektów
Aby umożliwić aplikacjom korzystanie z fabryki połączeń produktu IBM MQ JMS i WebSphere Application Server oraz definicji obiektów docelowych, właściwości pobrane z tych definicji muszą być odwzorowane na odpowiednie właściwości produktu XMS , które można ustawić w fabrykach połączeń i miejscach docelowych produktu XMS .

Właściwości obiektu InitialContext
Parametry konstruktora InitialContext obejmują położenie repozytorium administrowanych obiektów, podane jako jednolity wskaźnik zasobów (URI). Aby aplikacja mogła nawiązać połączenie z repozytorium, może być konieczne podanie większej ilości informacji niż informacje zawarte w identyfikatorze URI.

Format identyfikatora URI dla kontekstów początkowych produktu XMS
Położenie repozytorium administrowanych obiektów jest udostępniane jako jednolity wskaźnik zasobów (URI). Format identyfikatora URI zależy od typu kontekstu.

Usługa Web Service wyszukiwania JNDI
Aby uzyskać dostęp do katalogu nazw COS z poziomu produktu XMS, usługa Web Service wyszukiwania produktu JNDI musi zostać wdrożona na serwerze WebSphere Application Server service integration bus . Ta usługa Web Service tłumaczy informacje o produkcie Java z usługi nazw COS na formularz, który może być odczytany przez aplikacje produktu XMS .

Pobieranie administrowanych obiektów
Program XMS pobiera obiekt administrowany z repozytorium przy użyciu adresu udostępnionego podczas tworzenia obiektu InitialContext lub w właściwościach InitialContext .

Bezpieczne połączenia z menedżerem kolejek produktu IBM MQ
Aby umożliwić aplikacji XMS .NET nawiąże bezpieczne połączenia z menedżerem kolejek produktu IBM MQ , odpowiednie właściwości muszą zostać zdefiniowane w obiekcie ConnectionFactory .

Bezpieczne połączenia z mechanizmem przesyłania komunikatów produktu WebSphere Application Server service integration bus
Aby umożliwić aplikacji XMS .NET nawiąże bezpieczne połączenia z mechanizmem przesyłania komunikatów produktu WebSphere Application Server service integration bus , odpowiednie właściwości muszą zostać zdefiniowane w obiekcie ConnectionFactory .

Zadania pokrewne

Tworzenie obiektów administrowanych
Definicje obiektów ConnectionFactory i Destination, które są wymagane przez aplikacje produktu XMS w celu nawiązania połączenia z serwerem przesyłania komunikatów, muszą zostać utworzone przy użyciu odpowiednich narzędzi administracyjnych.

Obiekty InitialContext

Aplikacja musi utworzyć kontekst początkowy, który będzie używany w celu nawiązania połączenia z repozytorium obiektów administrowanych w celu pobrania wymaganych obiektów administrowanych.

Odsyłacze pokrewne

Wymagane właściwości administrowanych obiektów docelowych
Aplikacja tworzący miejsce docelowe musi ustawić kilka właściwości, które aplikacja ma w administrowanego obiekcie docelowym.

IConnectionFactory (dla interfejsu .NET)
Aplikacja korzysta z fabryki połączeń w celu utworzenia połączenia.

Właściwości obiektu ConnectionFactory
Przegląd właściwości obiektu ConnectionFactory z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

Wymagane właściwości administrowanych obiektów docelowych

Aplikacja tworzący miejsce docelowe musi ustawić kilka właściwości, które aplikacja ma w administrowanego obiekcie docelowym.

Tabela 13. Ustawienia właściwości dla administrowanych obiektów docelowych

Typ połączenia	**Właściwość**	**Opis**
IBM MQ menedżer kolejek	KOLEJKA (QU) TOPIC (TOP)	Kolejka, z którą ma zostać nawiązane połączenie Temat używany przez aplikację jako miejsce docelowe
Połączenie w czasie rzeczywistym z brokerem	TOPIC (TOP)	Temat używany przez aplikację jako miejsce docelowe
WebSphere Application Server service integration bus	topicName queueName	Jeśli aplikacja łączy się z tematem Jeśli aplikacja łączy się z kolejką

Pojęcia pokrewne

Obsługiwane typy administrowanego repozytorium obiektów
Obiekty administrowane przez system plików i LDAP mogą być używane do łączenia się z IBM MQ i WebSphere Application Server, podczas gdy nazwy COS mogą być używane tylko do łączenia się z serwerem WebSphere Application Server.

Odwzorowanie właściwości dla administrowanych obiektów
Aby umożliwić aplikacjom korzystanie z fabryki połączeń produktu IBM MQ JMS i WebSphere Application Server oraz definicji obiektów docelowych, właściwości pobrane z tych definicji muszą być odwzorowane na odpowiednie właściwości produktu XMS , które można ustawić w fabrykach połączeń i miejscach docelowych produktu XMS .

Właściwości obiektu InitialContext
Parametry konstruktora InitialContext obejmują położenie repozytorium administrowanych obiektów, podane jako jednolity wskaźnik zasobów (URI). Aby aplikacja mogła nawiązać połączenie z repozytorium, może być konieczne podanie większej ilości informacji niż informacje zawarte w identyfikatorze URI.

Format identyfikatora URI dla kontekstów początkowych produktu XMS
Położenie repozytorium administrowanych obiektów jest udostępniane jako jednolity wskaźnik zasobów (URI). Format identyfikatora URI zależy od typu kontekstu.

Usługa Web Service wyszukiwania JNDI
Aby uzyskać dostęp do katalogu nazw COS z poziomu produktu XMS, usługa Web Service wyszukiwania produktu JNDI musi zostać wdrożona na serwerze WebSphere Application Server service integration bus . Ta usługa Web Service tłumaczy informacje o produkcie Java z usługi nazw COS na formularz, który może być odczytany przez aplikacje produktu XMS .

Pobieranie administrowanych obiektów
Program XMS pobiera obiekt administrowany z repozytorium przy użyciu adresu udostępnionego podczas tworzenia obiektu InitialContext lub w właściwościach InitialContext .

Zadania pokrewne

Tworzenie obiektów administrowanych
Definicje obiektów ConnectionFactory i Destination, które są wymagane przez aplikacje produktu XMS w celu nawiązania połączenia z serwerem przesyłania komunikatów, muszą zostać utworzone przy użyciu odpowiednich narzędzi administracyjnych.

Obiekty InitialContext
Aplikacja musi utworzyć kontekst początkowy, który będzie używany w celu nawiązania połączenia z repozytorium obiektów administrowanych w celu pobrania wymaganych obiektów administrowanych.

Odsyłacze pokrewne

Wymagane właściwości dla administrowanych obiektów ConnectionFactory
Gdy aplikacja tworzy fabrykę połączeń, należy zdefiniować pewną liczbę właściwości, aby utworzyć połączenie z serwerem przesyłania komunikatów.

IDestination (dla interfejsu .NET)
Miejsce docelowe to miejsce, w którym aplikacja wysyła komunikaty lub jest to źródło, z którego aplikacja odbiera komunikaty, lub oba te komunikaty.

Właściwości miejsca docelowego
Przegląd właściwości obiektu docelowego, z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

Tworzenie obiektów administrowanych

Definicje obiektów ConnectionFactory i Destination, które są wymagane przez aplikacje produktu XMS w celu nawiązania połączenia z serwerem przesyłania komunikatów, muszą zostać utworzone przy użyciu odpowiednich narzędzi administracyjnych.

Zanim rozpoczniesz

Szczegółowe informacje na temat różnych typów administrowanych repozytorium obiektów obsługiwanych przez produkt XMS zawiera sekcja “Obsługiwane typy administrowanego repozytorium obiektów” na stronie 52.

O tym zadaniu

Aby utworzyć administrowane obiekty dla produktu IBM MQ , należy użyć narzędzia administrowania JMS (JMSAdmin) produktu IBM MQ Explorer lub produktu IBM MQ .

Aby utworzyć administrowane obiekty dla produktu IBM MQ lub IBM Integration Bus, należy użyć narzędzia administrowania JMS produktu IBM MQ (JMSAdmin).

Aby utworzyć administrowane obiekty dla produktu WebSphere Application Server service integration bus, należy użyć Konsoli administracyjnej serwera WebSphere Application Server .

W poniższych krokach podsumowany jest to, co należy zrobić, aby utworzyć administrowane obiekty.

Procedura

1. Utwórz fabrykę połączeń i zdefiniuj niezbędne właściwości, aby utworzyć połączenie z aplikacji na wybrany serwer.

   Właściwości minimalne wymagane przez produkt XMS w celu nawiązania połączenia są zdefiniowane w produkcie “Wymagane właściwości dla administrowanych obiektów ConnectionFactory” na stronie 56.
2. Utwórz wymagane miejsce docelowe na serwerze przesyłania komunikatów, z którym łączy się aplikacja:

   - W przypadku połączenia z menedżerem kolejek produktu IBM MQ należy utworzyć kolejkę lub temat.

- W celu nawiązania połączenia w czasie rzeczywistym z brokerem należy utworzyć temat.
- W przypadku połączenia z serwerem WebSphere Application Server service integration busnależy utworzyć kolejkę lub temat.

Właściwości minimalne wymagane przez produkt XMS w celu nawiązania połączenia są zdefiniowane w produkcie “Wymagane właściwości administrowanych obiektów docelowych” na stronie 58.

Pojęcia pokrewne

Obsługiwane typy administrowanego repozytorium obiektów
Obiekty administrowane przez system plików i LDAP mogą być używane do łączenia się z IBM MQ i WebSphere Application Server, podczas gdy nazwy COS mogą być używane tylko do łączenia się z serwerem WebSphere Application Server.

Odwzorowanie właściwości dla administrowanych obiektów
Aby umożliwić aplikacjom korzystanie z fabryki połączeń produktu IBM MQ JMS i WebSphere Application Server oraz definicji obiektów docelowych, właściwości pobrane z tych definicji muszą być odwzorowane na odpowiednie właściwości produktu XMS , które można ustawić w fabrykach połączeń i miejscach docelowych produktu XMS .

Właściwości obiektu InitialContext
Parametry konstruktora InitialContext obejmują położenie repozytorium administrowanych obiektów, podane jako jednolity wskaźnik zasobów (URI). Aby aplikacja mogła nawiązać połączenie z repozytorium, może być konieczne podanie większej ilości informacji niż informacje zawarte w identyfikatorze URI.

Format identyfikatora URI dla kontekstów początkowych produktu XMS
Położenie repozytorium administrowanych obiektów jest udostępniane jako jednolity wskaźnik zasobów (URI). Format identyfikatora URI zależy od typu kontekstu.

Usługa Web Service wyszukiwania JNDI
Aby uzyskać dostęp do katalogu nazw COS z poziomu produktu XMS, usługa Web Service wyszukiwania produktu JNDI musi zostać wdrożona na serwerze WebSphere Application Server service integration bus . Ta usługa Web Service tłumaczy informacje o produkcie Java z usługi nazw COS na formularz, który może być odczytany przez aplikacje produktu XMS .

Pobieranie administrowanych obiektów
Program XMS pobiera obiekt administrowany z repozytorium przy użyciu adresu udostępnionego podczas tworzenia obiektu InitialContext lub w właściwościach InitialContext .

Administrowane obiekty
Za pomocą administrowanych obiektów można administrować ustawieniami połączenia używalnymi przez aplikacje klienckie, które mają być administrowane z centralnego repozytorium. Aplikacja pobiera definicje obiektów z centralnego repozytorium i używa ich do tworzenia obiektów `ConnectionFactory` i `Destination` . Za pomocą administrowanych obiektów można usunąć kilka aplikacji z zasobów, które są używane w czasie wykonywania.

Obiekty ConnectionFactories i obiekty Connection
Obiekt ConnectionFactory udostępnia szablon, który jest używany przez aplikację do tworzenia obiektu połączenia. Aplikacja korzysta z obiektu połączenia w celu utworzenia obiektu sesji.

Połączenie z magistralą integracji usług
Aplikacja XMS może łączyć się z magistralą integracji usług WebSphere Application Server za pomocą bezpośredniego połączenia TCP/IP lub za pomocą protokołu HTTP przy użyciu protokołu TCP/IP.

Zadania pokrewne

Obiekty InitialContext
Aplikacja musi utworzyć kontekst początkowy, który będzie używany w celu nawiązania połączenia z repozytorium obiektów administrowanych w celu pobrania wymaganych obiektów administrowanych.

Odsyłacze pokrewne

Wymagane właściwości dla administrowanych obiektów ConnectionFactory
Gdy aplikacja tworzy fabrykę połączeń, należy zdefiniować pewną liczbę właściwości, aby utworzyć połączenie z serwerem przesyłania komunikatów.

Wymagane właściwości administrowanych obiektów docelowych

Aplikacja tworzący miejsce docelowe musi ustawić kilka właściwości, które aplikacja ma w administrowanego obiekcie docelowym.

IConnectionFactory (dla interfejsu .NET)
Aplikacja korzysta z fabryki połączeń w celu utworzenia połączenia.

Właściwości obiektu ConnectionFactory
Przegląd właściwości obiektu ConnectionFactory z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

IDestination (dla interfejsu .NET)
Miejsce docelowe to miejsce, w którym aplikacja wysyła komunikaty lub jest to źródło, z którego aplikacja odbiera komunikaty, lub oba te komunikaty.

Właściwości miejsca docelowego
Przegląd właściwości obiektu docelowego, z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

Obiekty InitialContext

Aplikacja musi utworzyć kontekst początkowy, który będzie używany w celu nawiązania połączenia z repozytorium obiektów administrowanych w celu pobrania wymaganych obiektów administrowanych.

O tym zadaniu

Obiekt InitialContext hermetyzuje połączenie z repozytorium. Interfejs API produktu XMS udostępnia metody służące do wykonywania następujących zadań:

- Tworzenie obiektu InitialContext
- Wyszukaj administrowany obiekt w administrowanych repozytorium obiektów.

For further details about creating an InitialContext object, see “InitialContext” na stronie 113 for .NET and “Właściwości obiektu InitialContext” na stronie 192.

Pojęcia pokrewne

Obsługiwane typy administrowanego repozytorium obiektów
Obiekty administrowane przez system plików i LDAP mogą być używane do łączenia się z IBM MQ i WebSphere Application Server, podczas gdy nazwy COS mogą być używane tylko do łączenia się z serwerem WebSphere Application Server.

Odwzorowanie właściwości dla administrowanych obiektów
Aby umożliwić aplikacjom korzystanie z fabryki połączeń produktu IBM MQ JMS i WebSphere Application Server oraz definicji obiektów docelowych, właściwości pobrane z tych definicji muszą być odwzorowane na odpowiednie właściwości produktu XMS , które można ustawić w fabrykach połączeń i miejscach docelowych produktu XMS .

Właściwości obiektu InitialContext
Parametry konstruktora InitialContext obejmują położenie repozytorium administrowanych obiektów, podane jako jednolity wskaźnik zasobów (URI). Aby aplikacja mogła nawiązać połączenie z repozytorium, może być konieczne podanie większej ilości informacji niż informacje zawarte w identyfikatorze URI.

Format identyfikatora URI dla kontekstów początkowych produktu XMS
Położenie repozytorium administrowanych obiektów jest udostępniane jako jednolity wskaźnik zasobów (URI). Format identyfikatora URI zależy od typu kontekstu.

Usługa Web Service wyszukiwania JNDI
Aby uzyskać dostęp do katalogu nazw COS z poziomu produktu XMS, usługa Web Service wyszukiwania produktu JNDI musi zostać wdrożona na serwerze WebSphere Application Server service integration bus . Ta usługa Web Service tłumaczy informacje o produkcie Java z usługi nazw COS na formularz, który może być odczytany przez aplikacje produktu XMS .

Pobieranie administrowanych obiektów
Program XMS pobiera obiekt administrowany z repozytorium przy użyciu adresu udostępnionego podczas tworzenia obiektu InitialContext lub w właściwościach InitialContext .

Zadania pokrewne

Tworzenie obiektów administrowanych
Definicje obiektów ConnectionFactory i Destination, które są wymagane przez aplikacje produktu XMS w celu nawiązania połączenia z serwerem przesyłania komunikatów, muszą zostać utworzone przy użyciu odpowiednich narzędzi administracyjnych.

Odsyłacze pokrewne

Wymagane właściwości dla administrowanych obiektów ConnectionFactory
Gdy aplikacja tworzy fabrykę połączeń, należy zdefiniować pewną liczbę właściwości, aby utworzyć połączenie z serwerem przesyłania komunikatów.

Wymagane właściwości administrowanych obiektów docelowych
Aplikacja tworzący miejsce docelowe musi ustawić kilka właściwości, które aplikacja ma w administrowanego obiekcie docelowym.

InitialContext (dla interfejsu .NET)
Aplikacja korzysta z obiektu InitialContext do tworzenia obiektów na podstawie definicji obiektów pobranych z repozytorium obiektów administrowanych.

Właściwości obiektu InitialContext
Przegląd właściwości obiektu InitialContext , z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

Właściwości obiektu InitialContext

Parametry konstruktora InitialContext obejmują położenie repozytorium administrowanych obiektów, podane jako jednolity wskaźnik zasobów (URI). Aby aplikacja mogła nawiązać połączenie z repozytorium, może być konieczne podanie większej ilości informacji niż informacje zawarte w identyfikatorze URI.

W interfejsie JNDI i w implementacji .NET produktu XMSinformacje dodatkowe są udostępniane w środowisku Hashtable do konstruktora.

Położenie administrowanego repozytorium obiektów jest zdefiniowane we właściwości XMSC_IC_URL . Ta właściwość jest zwykle przekazywana w wywołaniu Create (tworzenie), ale może być modyfikowana w celu nawiązania połączenia z innym katalogiem nazw przed wyszukiwaniem. W przypadku kontekstów FileSystem lub LDAP ta właściwość definiuje adres katalogu. W przypadku nazewnictwa COS jest to adres usługi WWW, która korzysta z tych właściwości w celu nawiązania połączenia z katalogiem JNDI.

Następujące właściwości są przekazywane bez modyfikacji do usługi Web Service, która będzie używać ich do łączenia się z katalogiem JNDI.

- URI_DOSTAWCY XMSC_IC_PROVIDER_URL
- XMSC_IC_SECURITY_CREDENTIALS
- XMSC_IC_SECURITY_AUTHENTICATION
- XMSC_IC_SECURITY_PRINCIPAL
- XMSC_IC_SECURITY_PROTOCOL

Pojęcia pokrewne

Obsługiwane typy administrowanego repozytorium obiektów
Obiekty administrowane przez system plików i LDAP mogą być używane do łączenia się z IBM MQ i WebSphere Application Server, podczas gdy nazwy COS mogą być używane tylko do łączenia się z serwerem WebSphere Application Server.

Odwzorowanie właściwości dla administrowanych obiektów
Aby umożliwić aplikacjom korzystanie z fabryki połączeń produktu IBM MQ JMS i WebSphere Application Server oraz definicji obiektów docelowych, właściwości pobrane z tych definicji muszą być odwzorowane na odpowiednie właściwości produktu XMS , które można ustawić w fabrykach połączeń i miejscach docelowych produktu XMS .

Format identyfikatora URI dla kontekstów początkowych produktu XMS
Położenie repozytorium administrowanych obiektów jest udostępniane jako jednolity wskaźnik zasobów (URI). Format identyfikatora URI zależy od typu kontekstu.

Usługa Web Service wyszukiwania JNDI
Aby uzyskać dostęp do katalogu nazw COS z poziomu produktu XMS, usługa Web Service wyszukiwania produktu JNDI musi zostać wdrożona na serwerze WebSphere Application Server service integration bus . Ta usługa Web Service tłumaczy informacje o produkcie Java z usługi nazw COS na formularz, który może być odczytany przez aplikacje produktu XMS .

Pobieranie administrowanych obiektów
Program XMS pobiera obiekt administrowany z repozytorium przy użyciu adresu udostępnionego podczas tworzenia obiektu InitialContext lub w właściwościach InitialContext .

Zadania pokrewne

Tworzenie obiektów administrowanych
Definicje obiektów ConnectionFactory i Destination, które są wymagane przez aplikacje produktu XMS w celu nawiązania połączenia z serwerem przesyłania komunikatów, muszą zostać utworzone przy użyciu odpowiednich narzędzi administracyjnych.

Obiekty InitialContext
Aplikacja musi utworzyć kontekst początkowy, który będzie używany w celu nawiązania połączenia z repozytorium obiektów administrowanych w celu pobrania wymaganych obiektów administrowanych.

Odsyłacze pokrewne

Wymagane właściwości dla administrowanych obiektów ConnectionFactory
Gdy aplikacja tworzy fabrykę połączeń, należy zdefiniować pewną liczbę właściwości, aby utworzyć połączenie z serwerem przesyłania komunikatów.

Wymagane właściwości administrowanych obiektów docelowych
Aplikacja tworzący miejsce docelowe musi ustawić kilka właściwości, które aplikacja ma w administrowanego obiekcie docelowym.

InitialContext (dla interfejsu .NET)
Aplikacja korzysta z obiektu InitialContext do tworzenia obiektów na podstawie definicji obiektów pobranych z repozytorium obiektów administrowanych.

Właściwości obiektu InitialContext
Przegląd właściwości obiektu InitialContext , z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

Format identyfikatora URI dla kontekstów początkowych produktu XMS

Położenie repozytorium administrowanych obiektów jest udostępniane jako jednolity wskaźnik zasobów (URI). Format identyfikatora URI zależy od typu kontekstu.

Kontekst FileSystem

W kontekście FileSystem adres URL określa położenie katalogu opartego na systemie plików. Struktura adresu URL jest zdefiniowana w dokumencie RFC 1738, *Uniform Resource Locators (URL)*: adres URL ma przedrostek `file://`, a składnia następująca po tym przedrostku jest poprawną definicją pliku, który można otworzyć w systemie, w którym działa produkt XMS .

Ta składnia może być specyficzna dla platformy i może używać separatorów '/separatorów lub' \ '. W przypadku użycia znaku '\', każdy separator musi być poprzedzony znakiem zmiany znaczenia, używając dodatkowego znaku '\'. Dzięki temu środowisko .NET nie będzie interpretowane jako znak zmiany znaczenia dla tego, co następuje.

Poniższe przykłady ilustrują tę składnię:

```
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
```

Kontekst LDAP

W kontekście LDAP podstawowa struktura adresu URL jest zdefiniowana w dokumencie RFC 2255, *The LDAP URL Format*(Format adresu URL LDAP), z przedrostkiem bez rozróżniania wielkości liter `ldap://` .

Dokładna składnia jest ilustrowana w następującym przykładzie:

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

Ta składnia jest zdefiniowana w dokumencie RFC, ale bez obsługi żadnych atrybutów, zasięgu, filtrów lub rozszerzeń.

Przykłady tej składni to:

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

Kontekst WSS

W kontekście WSS adres URL jest w postaci punktu końcowego usług Web Service z przedrostkiem `http://`.

Alternatywnie można użyć przedrostka `cosnaming://` lub `wsvc://`,

Te dwa przedrostki są interpretowane w ten sposób, że używany jest kontekst WSS z adresem URL, do którego dostęp jest uzyskiwany za pośrednictwem protokołu http, co umożliwia łatwe uzyskiwanie początkowego typu kontekstu bezpośrednio z adresu URL.

Przykłady tej składni zawierają następujące elementy:

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

Pojęcia pokrewne

Obsługiwane typy administrowanego repozytorium obiektów
Obiekty administrowane przez system plików i LDAP mogą być używane do łączenia się z IBM MQ i WebSphere Application Server, podczas gdy nazwy COS mogą być używane tylko do łączenia się z serwerem WebSphere Application Server.

Odwzorowanie właściwości dla administrowanych obiektów
Aby umożliwić aplikacjom korzystanie z fabryki połączeń produktu IBM MQ JMS i WebSphere Application Server oraz definicji obiektów docelowych, właściwości pobrane z tych definicji muszą być odwzorowane na odpowiednie właściwości produktu XMS , które można ustawić w fabrykach połączeń i miejscach docelowych produktu XMS .

Właściwości obiektu InitialContext
Parametry konstruktora InitialContext obejmują położenie repozytorium administrowanych obiektów, podane jako jednolity wskaźnik zasobów (URI). Aby aplikacja mogła nawiązać połączenie z repozytorium, może być konieczne podanie większej ilości informacji niż informacje zawarte w identyfikatorze URI.

Usługa Web Service wyszukiwania JNDI
Aby uzyskać dostęp do katalogu nazw COS z poziomu produktu XMS, usługa Web Service wyszukiwania produktu JNDI musi zostać wdrożona na serwerze WebSphere Application Server service integration bus . Ta usługa Web Service tłumaczy informacje o produkcie Java z usługi nazw COS na formularz, który może być odczytany przez aplikacje produktu XMS .

Pobieranie administrowanych obiektów
Program XMS pobiera obiekt administrowany z repozytorium przy użyciu adresu udostępnionego podczas tworzenia obiektu InitialContext lub w właściwościach InitialContext .

Zadania pokrewne

Tworzenie obiektów administrowanych
Definicje obiektów ConnectionFactory i Destination, które są wymagane przez aplikacje produktu XMS w celu nawiązania połączenia z serwerem przesyłania komunikatów, muszą zostać utworzone przy użyciu odpowiednich narzędzi administracyjnych.

Obiekty InitialContext
Aplikacja musi utworzyć kontekst początkowy, który będzie używany w celu nawiązania połączenia z repozytorium obiektów administrowanych w celu pobrania wymaganych obiektów administrowanych.

Odsyłacze pokrewne

Wymagane właściwości dla administrowanych obiektów ConnectionFactory
Gdy aplikacja tworzy fabrykę połączeń, należy zdefiniować pewną liczbę właściwości, aby utworzyć połączenie z serwerem przesyłania komunikatów.

Wymagane właściwości administrowanych obiektów docelowych
Aplikacja tworzący miejsce docelowe musi ustawić kilka właściwości, które aplikacja ma w administrowanego obiekcie docelowym.

InitialContext (dla interfejsu .NET)
Aplikacja korzysta z obiektu InitialContext do tworzenia obiektów na podstawie definicji obiektów pobranych z repozytorium obiektów administrowanych.

Właściwości obiektu InitialContext
Przegląd właściwości obiektu InitialContext , z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

Usługa Web Service wyszukiwania JNDI

Aby uzyskać dostęp do katalogu nazw COS z poziomu produktu XMS, usługa Web Service wyszukiwania produktu JNDI musi zostać wdrożona na serwerze WebSphere Application Server service integration bus . Ta usługa Web Service tłumaczy informacje o produkcie Java z usługi nazw COS na formularz, który może być odczytany przez aplikacje produktu XMS .

Usługa Web Service jest udostępniana w pliku archiwum korporacyjnego SIBXJndiLookupEAR.ear, który znajduje się w katalogu instalacyjnym. Dla bieżącej wersji produktu IBM Message Service Client for .NETplik SIBXJndiLookupEAR.ear można znaleźć w katalogu *install_dir*\java\lib . Można to zainstalować na serwerze WebSphere Application Server service integration bus przy użyciu Konsoli administracyjnej lub narzędzia skryptowego wsaadmin. Więcej informacji na temat wdrażania aplikacji usług Web Service znajduje się w dokumentacji produktu.

Aby zdefiniować usługę Web Service w aplikacjach produktu XMS , należy po prostu ustawić właściwość XMSC_IC_URL obiektu InitialContext na adres URL punktu końcowego usługi Web Service. Jeśli na przykład usługa Web Service jest wdrażana na hoście serwera o nazwie MyServer, przykładem adresu URL punktu końcowego usługi Web Service jest:

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

Ustawienie właściwości XMSC_IC_URL umożliwia wywoływanie funkcji InitialContext Lookup w celu wywołania usługi Web Service w zdefiniowanym punkcie końcowym, co z kolei wyszukuje wymagany administrowany obiekt z usługi nazw COS.

Aplikacje produktu .NET mogą korzystać z usługi Web Service. The server-side deployment is the same for XMS C, /C++ and, XMS .NET. XMS Program .NET wywołuje usługę Web Service bezpośrednio za pomocą środowiska Microsoft .NET .

Pojęcia pokrewne

Obsługiwane typy administrowanego repozytorium obiektów
Obiekty administrowane przez system plików i LDAP mogą być używane do łączenia się z IBM MQ i WebSphere Application Server, podczas gdy nazwy COS mogą być używane tylko do łączenia się z serwerem WebSphere Application Server.

Odwzorowanie właściwości dla administrowanych obiektów
Aby umożliwić aplikacjom korzystanie z fabryki połączeń produktu IBM MQ JMS i WebSphere Application Server oraz definicji obiektów docelowych, właściwości pobrane z tych definicji muszą być odwzorowane na odpowiednie właściwości produktu XMS , które można ustawić w fabrykach połączeń i miejscach docelowych produktu XMS .

Właściwości obiektu InitialContext
Parametry konstruktora InitialContext obejmują położenie repozytorium administrowanych obiektów, podane jako jednolity wskaźnik zasobów (URI). Aby aplikacja mogła nawiązać połączenie z repozytorium, może być konieczne podanie większej ilości informacji niż informacje zawarte w identyfikatorze URI.

Format identyfikatora URI dla kontekstów początkowych produktu XMS
Położenie repozytorium administrowanych obiektów jest udostępniane jako jednolity wskaźnik zasobów (URI). Format identyfikatora URI zależy od typu kontekstu.

Pobieranie administrowanych obiektów
Program XMS pobiera obiekt administrowany z repozytorium przy użyciu adresu udostępnionego podczas tworzenia obiektu InitialContext lub w właściwościach InitialContext .

Konfigurowanie środowiska serwera przesyłania komunikatów
Tematy w tej sekcji opisują sposób konfigurowania środowiska serwera przesyłania komunikatów w celu umożliwienia aplikacjom produktu XMS nawiązywania połączenia z serwerem.

Korzystanie z przykładowych aplikacji produktu XMS
Użyj przykładowych aplikacji dostarczanych razem z produktem XMS w celu zweryfikowania instalacji i konfiguracji serwera przesyłania komunikatów, a także w celu ułatwienia budowania własnych aplikacji. Przykłady udostępniają przegląd wspólnych funkcji każdego interfejsu API.

Zadania pokrewne

Tworzenie obiektów administrowanych
Definicje obiektów ConnectionFactory i Destination, które są wymagane przez aplikacje produktu XMS w celu nawiązania połączenia z serwerem przesyłania komunikatów, muszą zostać utworzone przy użyciu odpowiednich narzędzi administracyjnych.

Obiekty InitialContext
Aplikacja musi utworzyć kontekst początkowy, który będzie używany w celu nawiązania połączenia z repozytorium obiektów administrowanych w celu pobrania wymaganych obiektów administrowanych.

Instalowanie produktu Message Service Client for .NET przy użyciu kreatora instalacji
Instalacja korzysta z instalatora MSI InstallShield X/Windows. Dostępne są dwie opcje konfiguracji, dzięki czemu można wybrać instalację kompletną lub niestandardową.

Odsyłacze pokrewne

Wymagane właściwości dla administrowanych obiektów ConnectionFactory
Gdy aplikacja tworzy fabrykę połączeń, należy zdefiniować pewną liczbę właściwości, aby utworzyć połączenie z serwerem przesyłania komunikatów.

Wymagane właściwości administrowanych obiektów docelowych
Aplikacja tworzący miejsce docelowe musi ustawić kilka właściwości, które aplikacja ma w administrowanego obiekcie docelowym.

Pobieranie administrowanych obiektów

Program XMS pobiera obiekt administrowany z repozytorium przy użyciu adresu udostępnionego podczas tworzenia obiektu InitialContext lub w właściwościach InitialContext .

Obiekty, które mają zostać pobrane, mogą mieć następujące typy nazw:

- Prosta nazwa opisująca obiekt docelowy, na przykład miejsce docelowe kolejki o nazwie SalesOrders .
- Nazwa złożona, która może być złożona z elementu SubContexts, oddzielona przez '/', i musi kończyć się nazwą obiektu. Przykładem nazwy złożonej jest "Warehouse/PickLists/DispatchQueue2", gdzie Warehouse and Picklists to SubContexts w katalogu nazw, a DispatchQueue2 jest nazwą obiektu docelowego.

Pojęcia pokrewne

Obsługiwane typy administrowanego repozytorium obiektów
Obiekty administrowane przez system plików i LDAP mogą być używane do łączenia się z IBM MQ i WebSphere Application Server, podczas gdy nazwy COS mogą być używane tylko do łączenia się z serwerem WebSphere Application Server.

Odwzorowanie właściwości dla administrowanych obiektów
Aby umożliwić aplikacjom korzystanie z fabryki połączeń produktu IBM MQ JMS i WebSphere Application Server oraz definicji obiektów docelowych, właściwości pobrane z tych definicji muszą być odwzorowane na odpowiednie właściwości produktu XMS , które można ustawić w fabrykach połączeń i miejscach docelowych produktu XMS .

Właściwości obiektu InitialContext
Parametry konstruktora InitialContext obejmują położenie repozytorium administrowanych obiektów, podane jako jednolity wskaźnik zasobów (URI). Aby aplikacja mogła nawiązać połączenie z repozytorium, może być konieczne podanie większej ilości informacji niż informacje zawarte w identyfikatorze URI.

Format identyfikatora URI dla kontekstów początkowych produktu XMS
Położenie repozytorium administrowanych obiektów jest udostępniane jako jednolity wskaźnik zasobów (URI). Format identyfikatora URI zależy od typu kontekstu.

Usługa Web Service wyszukiwania JNDI
Aby uzyskać dostęp do katalogu nazw COS z poziomu produktu XMS, usługa Web Service wyszukiwania produktu JNDI musi zostać wdrożona na serwerze WebSphere Application Server service integration bus . Ta usługa Web Service tłumaczy informacje o produkcie Java z usługi nazw COS na formularz, który może być odczytany przez aplikacje produktu XMS .

Zadania pokrewne

Tworzenie obiektów administrowanych
Definicje obiektów ConnectionFactory i Destination, które są wymagane przez aplikacje produktu XMS w celu nawiązania połączenia z serwerem przesyłania komunikatów, muszą zostać utworzone przy użyciu odpowiednich narzędzi administracyjnych.

Obiekty InitialContext
Aplikacja musi utworzyć kontekst początkowy, który będzie używany w celu nawiązania połączenia z repozytorium obiektów administrowanych w celu pobrania wymaganych obiektów administrowanych.

Odsyłacze pokrewne

Wymagane właściwości dla administrowanych obiektów ConnectionFactory
Gdy aplikacja tworzy fabrykę połączeń, należy zdefiniować pewną liczbę właściwości, aby utworzyć połączenie z serwerem przesyłania komunikatów.

Wymagane właściwości administrowanych obiektów docelowych
Aplikacja tworzący miejsce docelowe musi ustawić kilka właściwości, które aplikacja ma w administrowanego obiekcie docelowym.

InitialContext (dla interfejsu .NET)
Aplikacja korzysta z obiektu InitialContext do tworzenia obiektów na podstawie definicji obiektów pobranych z repozytorium obiektów administrowanych.

Właściwości obiektu InitialContext
Przegląd właściwości obiektu InitialContext , z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

Zabezpieczanie komunikacji dla aplikacji XMS

Ta sekcja zawiera informacje na temat konfigurowania bezpiecznej komunikacji w celu umożliwienia aplikacjom produktu XMS nawiązywania połączeń za pośrednictwem protokołu SSL (Secure Sockets Layer) z mechanizmem przesyłania komunikatów produktu WebSphere Application Server service integration bus lub menedżerem kolejek produktu IBM MQ .

Sekcja zawiera następujące tematy:

- “Bezpieczne połączenia z menedżerem kolejek produktu IBM MQ” na stronie 68

- “Odwzorowania nazw CipherSuite i CipherSpec na potrzeby połączeń z menedżerem kolejek produktu IBM MQ .” na stronie 69
- “Bezpieczne połączenia z mechanizmem przesyłania komunikatów produktu WebSphere Application Server service integration bus” na stronie 69
- “Odwzorowania nazw CipherSuite i CipherSpec na potrzeby połączeń z serwerem WebSphere Application Server service integration bus” na stronie 71

Bezpieczne połączenia z menedżerem kolejek produktu IBM MQ

Aby umożliwić aplikacji XMS .NET nawiąże bezpieczne połączenia z menedżerem kolejek produktu IBM MQ , odpowiednie właściwości muszą zostać zdefiniowane w obiekcie ConnectionFactory .

Protokół używany w negocjacjach szyfrowania może być protokołem SSL (Secure Sockets Layer) lub TLS (Transport Layer Security), w zależności od tego, który pakiet CipherSuite jest określony w obiekcie ConnectionFactory .

Jeśli używane są biblioteki klienta IBM WebSphere MQ 7.0.0 Fix Pack 1 i nowsze i nawiązano połączenie z menedżerem kolejek produktu IBM WebSphere MQ 7.0 , można utworzyć wiele połączeń z tym samym menedżerem kolejek w aplikacji XMS . Jednak połączenie z innym menedżerem kolejek nie jest dozwolone. W przypadku próby uzyskania informacji o błędzie `MQRC_SSL_ALREADY_INITIALIZED` .

Jeśli używane są biblioteki klienta IBM WebSphere MQ 6.0 i nowsze, można utworzyć połączenie SSL tylko wtedy, gdy poprzednie połączenie SSL jest zamykane. Wielokrotne współbieżne połączenia SSL z tego samego procesu do tych samych lub różnych menedżerów kolejek nie są dozwolone. W przypadku podjęcia próby wykonania więcej niż jednego żądania, zostanie wyświetlone ostrzeżenie `MQRC_SSL_ALREADY_INITIALIZED`, co może oznaczać, że niektóre żądane parametry połączenia SSL zostały zignorowane.

Właściwości ConnectionFactory dla połączeń za pośrednictwem protokołu SSL do menedżera kolejek produktu IBM MQ z krótkim opisem są przedstawione w poniższej tabeli:

Tabela 14. Właściwości obiektu ConnectionFactory dla połączeń z menedżerem kolejek produktu IBM MQ przy użyciu protokołu SSL

Nazwa właściwości	**Opis**
XMSC_WMQ_SSL_CERT_STORES	Lokalizacje serwerów, na których znajdują się listy odwołań certyfikatów (Certificate Revocation List – CRL) używane podczas nawiązywania połączenia SSL z menedżerem kolejek.
XMSC_WMQ_SSL_CIPHER_SPEC	Nazwa specyfikacji szyfrowania używanej w przypadku bezpiecznego połączenia z menedżerem kolejek.
XMSC_WMQ_SSL_CIPHER_SUITE	Nazwa zestawu algorytmów szyfrowania używanego w przypadku połączenia TLS z menedżerem kolejek. Na podstawie podanego zestawu algorytmów szyfrowania jest określany protokół używany do negocjowania bezpiecznego połączenia.
XMSC_WMQ_SSL_CRYPTO_HW	Szczegóły konfiguracji sprzętu szyfrującego połączonego z systemem klienta.
XMSC_WMQ_SSL_FIPS_REQUIRED	Wartość tej właściwości określa, czy aplikacja może lub nie może używać zestawów algorytmów szyfrowania niezgodnych ze standardem FIPS. Jeśli ta właściwość zostanie ustawiona na wartość true, w przypadku połączeń klient-serwer będzie można używać tylko algorytmów zgodnych ze standardem FIPS.
XMSC_WMQ_SSL_KEY_REPOSITORY	Położenie pliku bazy danych kluczy, w którym przechowywane są klucze i certyfikaty.

Tabela 14. Właściwości obiektu ConnectionFactory dla połączeń z menedżerem kolejek produktu IBM MQ przy użyciu protokołu SSL (kontynuacja)	
Nazwa właściwości	**Opis**
XMSC_WMQ_SSL_KEY_RESETCOUNT	Wartość KeyResetCount reprezentuje całkowitą liczbę nieszyfrowanych bajtów wysłanych i odebranych w ramach konwersacji SSL przed ponownym wynegocjowaniem klucza tajnego.
XMSC_WMQ_SSL_PEER_NAME	Nazwa węzła sieci używana na potrzeby połączenia SSL z menedżerem kolejek.

Odsyłacze pokrewne

IConnectionFactory (dla interfejsu .NET)
Aplikacja korzysta z fabryki połączeń w celu utworzenia połączenia.

Właściwości obiektu ConnectionFactory
Przegląd właściwości obiektu ConnectionFactory z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

Wymagane właściwości dla administrowanych obiektów ConnectionFactory
Gdy aplikacja tworzy fabrykę połączeń, należy zdefiniować pewną liczbę właściwości, aby utworzyć połączenie z serwerem przesyłania komunikatów.

Odwzorowania nazw CipherSuite i CipherSpec na potrzeby połączeń z menedżerem kolejek produktu IBM MQ .

Element InitialContext tłumaczy się między właściwością fabryki połączeń JMS SSLCIPHERSUITE a XMS odpowiednikiem XMSC_WMQ_SSL_CIPHER_SPEC w pobliżu. Podobne tłumaczenie jest konieczne, jeśli określono wartość dla XMSC_WMQ_SSL_CIPHER_SUITE, ale pominięto wartość dla XMSC_WMQ_SSL_CIPHER_SPEC.

Tabela 15 na stronie 69 zawiera listę dostępnych specyfikacji CipherSpecs i ich odpowiedników JSSE CipherSuite .

Tabela 15. Dostępne odpowiedniki CipherSpecs i ich odpowiedniki JSSE CipherSuite	
CipherSpec	**Odpowiednik 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

Uwaga:

- Parametr TLS_RSA_WITH_3DES_EDE_CBC_SHA jest nieaktualny. Jednak może być ona nadal używana do przesyłania do 32 GB danych, zanim połączenie zostanie zakończone z błędem AMQ9288. Aby uniknąć tego błędu, należy unikać używania potrójnego algorytmu szyfrowania DES lub włączyć resetowanie klucza tajnego podczas korzystania z tej specyfikacji CipherSpec.

Bezpieczne połączenia z mechanizmem przesyłania komunikatów produktu WebSphere Application Server service integration bus

Aby umożliwić aplikacji XMS .NET nawiąże bezpieczne połączenia z mechanizmem przesyłania komunikatów produktu WebSphere Application Server service integration bus , odpowiednie właściwości muszą zostać zdefiniowane w obiekcie ConnectionFactory .

Produkt XMS udostępnia obsługę protokołu SSL i HTTPS dla połączeń z serwerem WebSphere Usługa Integration Bus. Protokół SSL i HTTPS zapewniają bezpieczne połączenia na potrzeby uwierzytelniania i poufności.

Podobnie jak zabezpieczenia produktu WebSphere , zabezpieczenia produktu XMS są konfigurowane w odniesieniu do standardów zabezpieczeń JSSE oraz konwencji nazewnictwa, które obejmują użycie opcji CipherSuites w celu określenia algorytmów używanych podczas negocjowania bezpiecznego połączenia. Protokół używany w negocjacjach szyfrowania może być protokołem SSL lub TLS, w zależności od tego, który element CipherSuite jest określony w obiekcie ConnectionFactory .

Tabela 16 na stronie 70 zawiera listę właściwości, które muszą być zdefiniowane w obiekcie ConnectionFactory .

Tabela 16. Właściwości obiektu ConnectionFactory służące do zabezpieczania połączeń z mechanizmem przesyłania komunikatów produktu WebSphere Application Server service integration bus

Nazwa właściwości	**Opis**
XMSC_WPM_SSL_CIPHER_SUITE	Nazwa pakietu CipherSuite , która ma być używana przez połączenie TLS do mechanizmu przesyłania komunikatów produktu WebSphere Usługa Integration Bus . Na podstawie podanego zestawu algorytmów szyfrowania jest określany protokół używany do negocjowania bezpiecznego połączenia.
XMSC_WPM_SSL_KEYRING_LABEL	Certyfikat używany podczas uwierzytelniania na serwerze.

Poniżej znajduje się przykład właściwości ConnectionFactory dla bezpiecznych połączeń z mechanizmem przesyłania komunikatów 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);
```

Jeśli nazwa łańcucha powinna być ustawiona na wartość BootstrapTunneledSecureMessaging lub BootstrapSecureMessaging, a numer_portu to numer portu, na którym serwer startowy nasłuchuje nadchodzących żądań.

Poniżej znajduje się przykład właściwości ConnectionFactory dla bezpiecznych połączeń z mechanizmem przesyłania komunikatów produktu WebSphere Application Server service integration bus z wstawioną przykładową wartością:

```
  /* 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");
```

Odsyłacze pokrewne

IConnectionFactory (dla interfejsu .NET)
Aplikacja korzysta z fabryki połączeń w celu utworzenia połączenia.

Właściwości obiektu ConnectionFactory
Przegląd właściwości obiektu ConnectionFactory z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

Wymagane właściwości dla administrowanych obiektów ConnectionFactory
Gdy aplikacja tworzy fabrykę połączeń, należy zdefiniować pewną liczbę właściwości, aby utworzyć połączenie z serwerem przesyłania komunikatów.

Odwzorowania nazw CipherSuite i CipherSpec na potrzeby połączeń z serwerem WebSphere Application Server service integration bus

Ponieważ pakiet GSKit używa CipherSpecs , a nie CipherSuites, nazwy CipherSuite w stylu JSSE określone we właściwości XMSC_WPM_SSL_CIPHER_SUITE muszą być odwzorowane na nazwy CipherSpec w stylu GSKit.

Tabela 17 na stronie 71 zawiera równoważną wartość CipherSpec dla każdego rozpoznanego pakietu CipherSuite.

Tabela 17. Dostępne pakiety CipherSuites i ich odpowiedniki CipherSpecs

CipherSuite	**Odpowiednik 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

Uwaga:

- Parametr TLS_RSA_WITH_3DES_EDE_CBC_SHA jest nieaktualny. Jednak może być ona nadal używana do przesyłania do 32 GB danych, zanim połączenie zostanie zakończone z błędem AMQ9288. Aby uniknąć tego błędu, należy unikać używania potrójnego algorytmu szyfrowania DES lub włączyć resetowanie klucza tajnego podczas korzystania z tej specyfikacji CipherSpec.

Komunikaty produktu XMS

W tej sekcji opisano strukturę i treść komunikatów produktu XMS oraz wyjaśniono, w jaki sposób aplikacje przetwarzają komunikaty produktu XMS .

Sekcja obejmuje następujące tematy:

- “Części komunikatu produktu XMS” na stronie 71
- “Pola nagłówka w komunikacie XMS” na stronie 72
- “Właściwości komunikatu XMS” na stronie 73
- “Treść komunikatu produktu XMS” na stronie 77
- “Selektory komunikatów” na stronie 83
- “Odwzorowywanie komunikatów produktu XMS na komunikaty produktu IBM MQ” na stronie 84

Odsyłacze pokrewne

IMessage (dla interfejsu .NET)
Obiekt komunikatu reprezentuje komunikat, który jest wysyłany lub odbierany przez aplikację. IMessage jest nadklasą dla klas komunikatów, takich jak IMapMessage.

Części komunikatu produktu XMS

Komunikat XMS składa się z nagłówka, zestawu właściwości i treści.

Nagłówek
Nagłówek komunikatu zawiera pola, a wszystkie komunikaty zawierają ten sam zestaw pól nagłówka. Program XMS i aplikacje używają wartości pól nagłówka do identyfikowania i kierowania komunikatów. Więcej informacji na temat pól nagłówka zawiera sekcja “Pola nagłówka w komunikacie XMS” na stronie 72.

Zestaw właściwości
Właściwości komunikatu określają dodatkowe informacje na temat komunikatu. Mimo że wszystkie komunikaty mają ten sam zestaw pól nagłówka, każdy komunikat może mieć inny zestaw właściwości. Więcej informacji na ten temat zawiera sekcja “Właściwości komunikatu XMS” na stronie 73.

Treść

Treść komunikatu zawiera dane aplikacji. Więcej informacji na ten temat zawiera sekcja “Treść komunikatu produktu XMS” na stronie 77.

Aplikacja może wybrać, które komunikaty mają być odbierane. Za pomocą selektorów komunikatów, które określają kryteria wyboru. Kryteria mogą być oparte na wartościach niektórych pól nagłówka i wartościach dowolnej właściwości komunikatu. Więcej informacji na temat selektorów komunikatów zawiera sekcja “Selektory komunikatów” na stronie 83.

Odsyłacze pokrewne

Pola nagłówka w komunikacie XMS
Aby umożliwić aplikacji XMS wymianę komunikatów z aplikacją WebSphere JMS , nagłówek komunikatu XMS zawiera pola nagłówka komunikatu produktu JMS .

Właściwości komunikatu XMS
Produkt XMS obsługuje trzy rodzaje właściwości komunikatu: JMS zdefiniowane właściwości, IBM zdefiniowane właściwości i właściwości zdefiniowane przez aplikację.

Treść komunikatu produktu XMS
Treść komunikatu zawiera dane aplikacji. Jednak komunikat nie może mieć treści i zawierać tylko pola nagłówka i właściwości.

Selektory komunikatów
Aplikacja XMS używa selektorów komunikatów w celu wybrania komunikatów, które mają zostać odebrane.

Odwzorowywanie komunikatów produktu XMS na komunikaty produktu IBM MQ
Pola nagłówka JMS i właściwości komunikatu produktu XMS są odwzorowywane na pola w strukturach nagłówka komunikatu produktu IBM MQ .

Pola nagłówka w komunikacie XMS

Aby umożliwić aplikacji XMS wymianę komunikatów z aplikacją WebSphere JMS , nagłówek komunikatu XMS zawiera pola nagłówka komunikatu produktu JMS .

Nazwy tych pól nagłówka są rozpoczynanie z przedrostkiem JMS. Opis pól nagłówka komunikatu JMS znajduje się w publikacji *Java Message Service Specyfikacja*.

Produkt XMS implementuje pola nagłówka komunikatu produktu JMS jako atrybuty obiektu komunikatu. Każde pole nagłówka ma własne metody ustawiania i pobierania wartości. Opis tych metod znajduje się w sekcji “Komunikat IMessage” na stronie 126. Pole nagłówka jest zawsze czytelne i dostępne do zapisu.

Tabela 18 na stronie 72 zawiera listę pól nagłówka komunikatu produktu JMS i wskazuje, w jaki sposób wartość każdego pola jest ustawiana dla przesłanego komunikatu. Niektóre pola są ustawiane automatycznie przez produkt XMS , gdy aplikacja wysyła komunikat lub, w przypadku obiektu JMSRedelivered, gdy aplikacja odbierze komunikat.

Tabela 18. Pola nagłówka komunikatu produktu JMS

Nazwa pola nagłówka komunikatu produktu JMS .	**Sposób ustawiania wartości dla przesłanego komunikatu (w formacie *metoda* [*klasa*])**
JMSCorrelationID	Ustaw wartość JMSCorrelationID [Komunikat]
JMSDeliveryMode	Wyślij [MessageProducer]
JMSDestination	Wyślij [MessageProducer]
JMSExpiration	Wyślij [MessageProducer]
JMSMessageID	Wyślij [MessageProducer]
JMSPriority	Wyślij [MessageProducer]
JMSRedelivered	Receive [MessageConsumer]

Tabela 18. Pola nagłówka komunikatu produktu JMS (kontynuacja)	
Nazwa pola nagłówka komunikatu produktu JMS .	**Sposób ustawiania wartości dla przesłanego komunikatu (w formacie *metoda* [*klasa*])**
JMSReplyTo	Ustaw wartość JMSReplyTo [Komunikat]
JMSTimestamp	Wyślij [MessageProducer]
JMSType	Ustaw atrybut JMSType [Komunikat]

Odsyłacze pokrewne

Części komunikatu produktu XMS
Komunikat XMS składa się z nagłówka, zestawu właściwości i treści.

Właściwości komunikatu XMS
Produkt XMS obsługuje trzy rodzaje właściwości komunikatu: JMS zdefiniowane właściwości, IBM zdefiniowane właściwości i właściwości zdefiniowane przez aplikację.

Treść komunikatu produktu XMS
Treść komunikatu zawiera dane aplikacji. Jednak komunikat nie może mieć treści i zawierać tylko pola nagłówka i właściwości.

Selektory komunikatów
Aplikacja XMS używa selektorów komunikatów w celu wybrania komunikatów, które mają zostać odebrane.

Odwzorowywanie komunikatów produktu XMS na komunikaty produktu IBM MQ
Pola nagłówka JMS i właściwości komunikatu produktu XMS są odwzorowywane na pola w strukturach nagłówka komunikatu produktu IBM MQ .

Właściwości komunikatu XMS

Produkt XMS obsługuje trzy rodzaje właściwości komunikatu: JMS zdefiniowane właściwości, IBM zdefiniowane właściwości i właściwości zdefiniowane przez aplikację.

Aplikacja XMS może wymieniać komunikaty z aplikacją WebSphere JMS , ponieważ program XMS obsługuje następujące predefiniowane właściwości obiektu komunikatu:

- Te same właściwości zdefiniowane przez produkt JMS, które są obsługiwane przez program WebSphere JMS . Nazwy tych właściwości rozpoczynają się od przedrostka JMSX.
- Te same właściwości zdefiniowane przez produkt IBM, które są obsługiwane przez program WebSphere JMS . Nazwy tych właściwości rozpoczynają się od przedrostka JMS_IBM_.

Każda predefiniowana właściwość ma dwie nazwy:

- Nazwa JMS dla zdefiniowanej przez JMSwłaściwości lub nazwy WebSphere JMS dla właściwości zdefiniowanej w produkcie IBM.

  Jest to nazwa, pod którą właściwość jest znana w produkcie JMS lub WebSphere JMS, a także jest nazwą, która jest przesyłana z komunikatem o tej właściwości. Aplikacja XMS używa tej nazwy do identyfikowania właściwości w wyrażeniu selektora komunikatów.
- Nazwa XMS identyfikującą właściwość we wszystkich sytuacjach z wyjątkiem wyrażenia selektora komunikatów. Każda nazwa XMS jest zdefiniowana jako stała nazwana w klasie `IBM.XMS.XMSC` . The value of the named constant is the corresponding JMS or WebSphere JMS name.

Oprócz właściwości predefiniowanych aplikacja XMS może tworzyć i używać własnego zestawu właściwości komunikatu. Te właściwości są nazywane *właściwościami zdefiniowanymi przez aplikację.*

Po utworzeniu komunikatu przez aplikację właściwości komunikatu są czytelne i dostępne do zapisu. Po wysłaniu przez aplikację komunikatu właściwości pozostają czytelne i dostępne do zapisu. Gdy aplikacja odbierze komunikat, właściwości komunikatu są tylko do odczytu. Jeśli aplikacja wywołuje metodę `Clear Properties` klasy Message, gdy właściwości komunikatu są tylko do odczytu, wówczas właściwości stają się czytelne i dostępne do zapisu. Metoda kasuje również właściwości.

Odebrany komunikat po wyczyszczaniu właściwości komunikatu będzie zachowany w sposób zgodny z zachowaniem przekazywania standardowego produktu WMQ XMS dla platformy .NET BytesMessage z wyczyszczonym właściwościami komunikatu.

Jest to jednak niezalecane, ponieważ zostaną utracone następujące właściwości:

- Wartość właściwości JMS_IBM_Encoding, co oznacza, że dane komunikatu nie mogą być zdekodowane w sposób znaczący.
- Wartość właściwości JMS_IBM_Format, co oznacza, że nagłówek łączenia nagłówka (MQMD lub nowy MQRFH2) w nagłówku i istniejących nagłówkach zostanie zerwany.

Aby określić wartości wszystkich właściwości komunikatu, aplikacja może wywołać metodę Get Properties klasy Message. Metoda tworzy iterator, który hermetykuje listę obiektów właściwości, gdzie każdy obiekt właściwości reprezentuje właściwość komunikatu. Aplikacja może następnie użyć metod klasy Iterator w celu pobrania wszystkich obiektów właściwości z kolei, a także użyć metod klasy Property do pobrania nazwy, typu danych i wartości każdej właściwości.

Odsyłacze pokrewne

Części komunikatu produktu XMS
Komunikat XMS składa się z nagłówka, zestawu właściwości i treści.

Pola nagłówka w komunikacie XMS
Aby umożliwić aplikacji XMS wymianę komunikatów z aplikacją WebSphere JMS , nagłówek komunikatu XMS zawiera pola nagłówka komunikatu produktu JMS .

Treść komunikatu produktu XMS
Treść komunikatu zawiera dane aplikacji. Jednak komunikat nie może mieć treści i zawierać tylko pola nagłówka i właściwości.

Selektory komunikatów
Aplikacja XMS używa selektorów komunikatów w celu wybrania komunikatów, które mają zostać odebrane.

Odwzorowywanie komunikatów produktu XMS na komunikaty produktu IBM MQ
Pola nagłówka JMS i właściwości komunikatu produktu XMS są odwzorowywane na pola w strukturach nagłówka komunikatu produktu IBM MQ .

JMS-zdefiniowane właściwości komunikatu

Several JMS-defined properties of a message are supported by both XMS and WebSphere JMS.

Tabela 19 na stronie 74 lists the JMS-defined properties of a message that are supported by both XMS and WebSphere JMS. Opis właściwości zdefiniowanych w produkcie JMSmożna znaleźć w sekcji *Java Message Service Specyfikacja*. Właściwości zdefiniowane w produkcie JMSnie są poprawne dla połączenia w czasie rzeczywistym z brokerem.

W tabeli określa się typ danych każdej właściwości i wskazuje, w jaki sposób wartość właściwości jest ustawiona dla przekazanego komunikatu. Niektóre właściwości są ustawiane automatycznie przez produkt XMS , gdy aplikacja wysyła komunikat lub, w przypadku elementu JMSXDeliveryCount, gdy aplikacja odbiera komunikat.

Tabela 19. JMS-zdefiniowane właściwości komunikatu

Nazwa XMS zdefiniowanej właściwości JMS	**Nazwa JMS**	**Typ danych**	**Sposób ustawiania wartości dla przesłanego komunikatu (w formacie *metoda* [*klasa*])**
JMSX_APPID	JMSXAppID	System.String	Wyślij [MessageProducer]
JMSX_DELIVERY_COUNT	JMSXDeliveryCount	System.Int32	Receive [MessageConsumer]
Identyfikator JMSX_GROUPID	JMSXGroupID	System.String	Ustaw właściwość String [PropertyContext]

Tabela 19. JMS-zdefiniowane właściwości komunikatu (kontynuacja)			
Nazwa XMS zdefiniowanej właściwości JMS	**Nazwa JMS**	**Typ danych**	**Sposób ustawiania wartości dla przesłanego komunikatu (w formacie *metoda* [*klasa*])**
JMSX_GROUPSEQ	JMSXGroupSeq	System.Int32	Ustaw właściwość całkowitoliczbową [PropertyContext]
ID_UŻYTKOWNIKA JMSX_USERID	JMSXUserID	System.String	Wyślij [MessageProducer]

IBM-zdefiniowane właściwości komunikatu

Several IBM-defined properties of a message are supported by XMS and WebSphere JMS.

Tabela 20 na stronie 75 lists the IBM defined properties of a message that are supported by both XMS and WebSphere JMS. Więcej informacji na temat właściwości zdefiniowanych w produkcie IBMmożna znaleźć w dokumentacji produktu IBM MQ lub WebSphere Application Server .

W tabeli określa się typ danych każdej właściwości i wskazuje, w jaki sposób wartość właściwości jest ustawiona dla przekazanego komunikatu. Niektóre właściwości są ustawiane automatycznie przez produkt XMS , gdy aplikacja wysyła komunikat.

Tabela 20. Właściwości zdefiniowane przez IBMkomunikatu			
Nazwa XMS zdefiniowanej właściwości IBM	**Nazwa WebSphere JMS**	**Typ danych**	**Sposób ustawiania wartości dla przesłanego komunikatu (w formacie *metoda* [*klasa*])**
JMS_IBM_CHARACTER_SET	Zestaw znaków JMS_IBM_Character_Set	System.Int32	Ustaw właściwość całkowitoliczbową [PropertyContext]
JMS_IBM_Encoding	JMS_IBM_Encoding	System.Int32	Ustaw właściwość całkowitoliczbową [PropertyContext]
JMS_IBM_EXCEPTIONMESSAGE	JMS_IBM_ExceptionMessage	System.String	Receive [MessageConsumer]
JMS_IBM_EXCEPTIONREASON	JMS_IBM_ExceptionReason	System.Int32	Receive [MessageConsumer]
JMS_IBM_EXCEPTIONTIMESTAMP	JMS_IBM_ExceptionTimestamp	System.Int64	Receive [MessageConsumer]
JMS_IBM_EXCEPTIONPROBLEM DESTINATION	Miejsce docelowe JMS_IBM_ExceptionProblem	System.String	Receive [MessageConsumer]
JMS_IBM_FEEDBACK	Opinie JMS_IBM_Feedback	System.Int32	Ustaw właściwość całkowitoliczbową [PropertyContext]
FORMAT JMS_IBM_FORMAT	Format JMS_IBM_Format	System.String	Ustaw właściwość String [PropertyContext]

Tabela 20. Właściwości zdefiniowane przez IBMkomunikatu (kontynuacja)

Nazwa XMS zdefiniowanej właściwości IBM	**Nazwa WebSphere JMS**	**Typ danych**	**Sposób ustawiania wartości dla przesłanego komunikatu (w formacie *metoda* [*klasa*])**
GRUPA_JMS_IBM_LAST_MSG_IN_GROUP	Grupa JMS_IBM_Last_Msg_In_Group	System.Boolean	Ustaw właściwość całkowitoliczbową [PropertyContext]
TYP_JMS_IBM_MSGTYPE	JMS_IBM_MsgType	System.Int32	Ustaw właściwość całkowitoliczbową [PropertyContext]
TYP_JMS_IBM_PUTAPPLTYPE	Typ JMS_IBM_PutAppl	System.Int32	Wyślij [MessageProducer]
DATA JMS_IBM_PUTDATE	JMS_IBM_PutDate	System.String	Wyślij [MessageProducer]
CZAS JMS_IBM_PUTTIME	JMS_IBM_PutTime	System.String	Wyślij [MessageProducer]
JMS_IBM_REPORT_COA	Plik JMS_IBM_Report_COA	System.Int32	Ustaw właściwość całkowitoliczbową [PropertyContext]
JMS_IBM_REPORT_COD	JMS_IBM_Report_COD	System.Int32	Ustaw właściwość całkowitoliczbową [PropertyContext]
JMS_IBM_REPORT_DISCARD_MSG	JMS_IBM_Report_Discard_Msg	System.Int32	Ustaw właściwość całkowitoliczbową [PropertyContext]
JMS_IBM_REPORT_EXCEPTION	Wyjątek JMS_IBM_Report_Exception	System.Int32	Ustaw właściwość całkowitoliczbową [PropertyContext]
JMS_IBM_REPORT_EXPIRATION	Wygaśnięcie JMS_IBM_Report_Expiration	System.Int32	Ustaw właściwość całkowitoliczbową [PropertyContext]
JMS_IBM_REPORT_NAN	Plik JMS_IBM_Report_NAN	System.Int32	Ustaw właściwość całkowitoliczbową [PropertyContext]
JMS_IBM_REPORT_PAN	Plik JMS_IBM_Report_PAN	System.Int32	Ustaw właściwość całkowitoliczbową [PropertyContext]
JMS_IBM_REPORT_PASS_CORREL_ id	JMS_IBM_Report_Pass_Correl_ID	System.Int32	Ustaw właściwość całkowitoliczbową [PropertyContext]
JMS_IBM_REPORT_PASS_MSG_ID	JMS_IBM_Report_Pass_Msg_ID	System.Int32	Ustaw właściwość całkowitoliczbową [PropertyContext]
JMS_IBM_SYSTEM_MESSAGEID	JMS_IBM_System_MessageID	System.String	Wyślij [MessageProducer]

Właściwości komunikatu zdefiniowane przez aplikację

Aplikacja XMS może tworzyć i używać własnego zestawu właściwości komunikatu. Gdy aplikacja wysyła komunikat, te właściwości są również przesyłane razem z komunikatem. Aplikacja odbierający, korzystając z selektorów komunikatów, może następnie wybrać komunikaty, które mają być odbierane w oparciu o wartości tych właściwości.

Aby umożliwić aplikacji WebSphere JMS wybór i przetwarzanie komunikatów wysyłanych przez aplikację XMS , nazwa właściwości zdefiniowanej przez aplikację musi być zgodna z regułami tworzenia identyfikatorów w wyrażeniach selektora komunikatów, zgodnie z opisem w dokumentacji produktu IBM MQ . Wartość właściwości definiowanej przez aplikację musi mieć jeden z następujących typów danych: System.Boolean, System.SByte, System.Int16, System.Int32, System.Int64, System.Float, System.Doublelub System.String.

Treść komunikatu produktu XMS

Treść komunikatu zawiera dane aplikacji. Jednak komunikat nie może mieć treści i zawierać tylko pola nagłówka i właściwości.

Produkt XMS obsługuje pięć typów treści komunikatu:

Bajty
Treść zawiera strumień bajtów. Komunikat z tym typem treści jest nazywany *komunikatem bajtów.* Interfejs IBytesMessage zawiera metody przetwarzania treści komunikatu bajtów. Więcej informacji na ten temat zawiera sekcja “Komunikaty bajtowe” na stronie 79,

Odwzoruj
Treść zawiera zestaw par nazwa-wartość, w których każda wartość ma powiązany typ danych. Komunikat z tym typem treści jest nazywany *komunikatem mapy.* Interfejs IMapMessage zawiera metody przetwarzania treści komunikatu mapy. Więcej informacji na ten temat zawiera sekcja “Komunikaty mapy” na stronie 80,

Obiekt
Treść zawiera serializowany obiekt Java lub .NET . Komunikat z tym typem treści jest nazywany *komunikatem obiektu.* Interfejs IObjectMessage zawiera metody przetwarzania treści komunikatu obiektu. Więcej informacji na ten temat zawiera sekcja “Komunikaty obiektu” na stronie 80.

Strumień
Treść zawiera strumień wartości, w którym każda wartość ma powiązany typ danych. Komunikat z tym typem treści jest nazywany *komunikatem strumienia.* Interfejs IStreamMessage zawiera metody przetwarzania treści komunikatu strumienia. Więcej informacji na ten temat zawiera sekcja “Komunikaty strumienia” na stronie 81,

Tekstowy
Treść zawiera łańcuch. Komunikat z tym typem treści jest nazywany *komunikatem tekstowym.* Interfejs ITextMessage zawiera metody przetwarzania treści komunikatu tekstowego. Więcej informacji na ten temat zawiera sekcja “Komunikaty tekstowe” na stronie 82,

Interfejs IMessage jest nadrzędny względem wszystkich obiektów komunikatów i może być używany w funkcjach przesyłania komunikatów do reprezentowania dowolnego typu komunikatu produktu XMS .

Więcej informacji na temat wielkości oraz wartości maksymalnych i minimalnych dla każdego z tych typów danych zawiera sekcja Tabela 5 na stronie 41.

Odsyłacze pokrewne

Części komunikatu produktu XMS
Komunikat XMS składa się z nagłówka, zestawu właściwości i treści.

Pola nagłówka w komunikacie XMS
Aby umożliwić aplikacji XMS wymianę komunikatów z aplikacją WebSphere JMS , nagłówek komunikatu XMS zawiera pola nagłówka komunikatu produktu JMS .

Właściwości komunikatu XMS
Produkt XMS obsługuje trzy rodzaje właściwości komunikatu: JMS zdefiniowane właściwości, IBM zdefiniowane właściwości i właściwości zdefiniowane przez aplikację.

Selektory komunikatów
Aplikacja XMS używa selektorów komunikatów w celu wybrania komunikatów, które mają zostać odebrane.

Odwzorowywanie komunikatów produktu XMS na komunikaty produktu IBM MQ
Pola nagłówka JMS i właściwości komunikatu produktu XMS są odwzorowywane na pola w strukturach nagłówka komunikatu produktu IBM MQ .

Typy danych dla elementów danych aplikacji

Aby aplikacja XMS mogła wymieniać komunikaty z aplikacją IBM MQ classes for JMS , zarówno aplikacje, jak i aplikacje muszą być w stanie interpretować dane aplikacji w treści komunikatu w ten sam sposób.

Z tego powodu każdy element danych aplikacji zapisany w treści komunikatu przez aplikację XMS musi mieć jeden z typów danych wymienionych w sekcji Tabela 21 na stronie 78. Dla każdego typu danych w tabeli wyświetlany jest zgodny typ danych Java . Produkt XMS udostępnia metody do zapisywania elementów danych aplikacji tylko z tymi typami danych.

Tabela 21. Typy danych produktu XMS zgodne z typami danych produktu Java

Typ danych produktu XMS	**Reprezentuje**	**Zgodny typ danych Java**
System.Boolean	Wartość boolowska true lub false	boolean (boolowskie)
System.Char16	Znak dwubajtowy	char
System.SByte	8-bitowa liczba całkowita ze znakiem	B
System.Int16	16-bitowa liczba całkowita ze znakiem	short
System.Int32	32-bitowa liczba całkowita ze znakiem	int
System.Int64	64-bitowa liczba całkowita ze znakiem	long
System.Float	Liczba podpisanych zmiennopozycyjnych	liczba zmiennopozycyjna
System.Double	Liczba zmiennopozycyjna o podwójnej precyzji podpisanej	double (podwójna)
System.String	Łańcuch znaków	Łańcuch

Więcej informacji na temat wielkości, wartości maksymalnej i minimalnej wartości każdego z tych typów danych zawiera sekcja “Typy podstawowe produktu XMS” na stronie 40.

Pojęcia pokrewne

Atrybuty i właściwości obiektów
Obiekt XMS może mieć atrybuty i właściwości, które są właściwościami obiektu, które są implementowane na różne sposoby.

Typy podstawowe produktu XMS
Produkt XMS udostępnia odpowiedniki ośmiu typów podstawowych Java (byte, short, int, long, float, double, char i boolean). Pozwala to na wymianę komunikatów między XMS a JMS bez utraty lub uszkodzenia danych.

Niejawne przekształcenie wartości właściwości z jednego typu danych na inny.
Gdy aplikacja pobiera wartość właściwości, wartość może zostać przekształcona przez produkt XMS na inny typ danych. Wiele reguł decyduje o tym, które konwersje są obsługiwane i w jaki sposób program XMS wykonuje konwersje.

Odsyłacze pokrewne

Komunikaty bajtowe
Treść komunikatu bajtowego zawiera strumień bajtów. Jednostka zawiera tylko rzeczywiste dane, a odpowiedzialność za ich interpretowanie i odbieranie polega na wysyłaniu i odbierającym.

Komunikaty mapy
Treść komunikatu mapy zawiera zestaw par nazwa-wartość, w których każda wartość ma powiązany typ danych.

Komunikaty obiektu
Treść komunikatu obiektu zawiera serializowany obiektJava lub .NET .

Komunikaty strumienia
Treść komunikatu strumienia zawiera strumień wartości, w którym każda wartość ma powiązany typ danych.

Komunikaty tekstowe
Treść wiadomości tekstowej zawiera łańcuch.

Komunikaty bajtowe

Treść komunikatu bajtowego zawiera strumień bajtów. Jednostka zawiera tylko rzeczywiste dane, a odpowiedzialność za ich interpretowanie i odbieranie polega na wysyłaniu i odbierającym.

Komunikaty bajtowe są przydatne, jeśli aplikacja produktu XMS wymaga wymiany komunikatów z aplikacjami, które nie korzystają z interfejsu programistycznego aplikacji XMS lub JMS .

Po utworzeniu przez aplikację komunikatu bajtowego, treść komunikatu jest tylko do zapisu. Aplikacja zestawia dane aplikacji do treści, wywołując odpowiednie metody zapisu w interfejsie IBytesMessage dla produktu .NET. Za każdym razem, gdy aplikacja zapisuje wartość do strumienia komunikatów bajtów, wartość ta jest składana natychmiast po poprzedniej wartości zapisanej przez aplikację. XMS utrzymuje kursor wewnętrzny, aby pamiętać o pozycji ostatniego bajtu, który został zmontowany.

Po wysłaniu komunikatu przez aplikację treść komunikatu staje się dostępna tylko do odczytu. W tym trybie aplikacja może wielokrotnie wysyłać komunikat.

Gdy aplikacja odbierze komunikat w postaci bajtów, treść komunikatu jest tylko do odczytu. Aplikacja może użyć odpowiednich metod odczytu interfejsu IBytesMessage , aby odczytać zawartość strumienia komunikatów bajtów. Aplikacja odczytuje bajty w sekwencji, a program XMS utrzymuje kursor wewnętrzny, aby zapamiętać pozycję ostatniego odczytanego bajtu.

Jeśli aplikacja wywoła metodę Reset interfejsu IBytesMessage , gdy treść komunikatu bajtowego jest dostępna do zapisu, treść staje się dostępna tylko do odczytu. Metoda ta umożliwia również ponowne pozycjonowanie kursora na początku strumienia komunikatów.

Jeśli aplikacja wywoła metodę `Clear Body` interfejsu IMessage dla .NET , gdy treść komunikatu bajtowego jest tylko do odczytu, wówczas treść staje się dostępna do zapisu. Metoda również czyści ciało.

Odsyłacze pokrewne

Typy danych dla elementów danych aplikacji
Aby aplikacja XMS mogła wymieniać komunikaty z aplikacją IBM MQ classes for JMS , zarówno aplikacje, jak i aplikacje muszą być w stanie interpretować dane aplikacji w treści komunikatu w ten sam sposób.

Komunikaty mapy
Treść komunikatu mapy zawiera zestaw par nazwa-wartość, w których każda wartość ma powiązany typ danych.

Komunikaty obiektu
Treść komunikatu obiektu zawiera serializowany obiektJava lub .NET .

Komunikaty strumienia
Treść komunikatu strumienia zawiera strumień wartości, w którym każda wartość ma powiązany typ danych.

Komunikaty tekstowe
Treść wiadomości tekstowej zawiera łańcuch.

IBytesMessage (dla interfejsu .NET)
Komunikat bajtów to komunikat, którego treść zawiera strumień bajtów.

Komunikaty mapy

Treść komunikatu mapy zawiera zestaw par nazwa-wartość, w których każda wartość ma powiązany typ danych.

W każdej parze nazwa-wartość nazwa jest łańcuchem, który identyfikuje wartość, a wartość jest elementem danych aplikacji, które mają jeden z typów danych XMS wymienionych w Tabela 21 na stronie 78. Kolejność par nazwa-wartość nie jest zdefiniowana. Klasa MapMessage zawiera metody służące do ustawiania i pobierania par nazwa-wartość.

Aplikacja może losowo uzyskać dostęp do pary nazwa-wartość, określając jej nazwę.

Aplikacja .NET może użyć właściwości MapNames , aby uzyskać wyliczenie nazw w treści komunikatu mapy.

Jeśli aplikacja pobiera wartość pary nazwa-wartość, wartość może zostać przekształcona przez program XMS na inny typ danych. Na przykład, aby uzyskać liczbę całkowitą z treści komunikatu odwzorowania, aplikacja może wywołać metodę GetString klasy MapMessage , która zwraca liczbę całkowitą jako łańcuch. Obsługiwane konwersje są takie same, jak te, które są obsługiwane, gdy program XMS przekształca wartość właściwości z jednego typu danych na inny. Więcej informacji na temat obsługiwanych konwersji zawiera sekcja “Niejawne przekształcenie wartości właściwości z jednego typu danych na inny.” na stronie 41.

Po utworzeniu przez aplikację komunikatu odwzorowania treść komunikatu jest czytelna i dostępna do zapisu. Po wysłaniu przez aplikację wiadomości treść pozostaje czytelna i dostępna do zapisu. Gdy aplikacja odbierze komunikat mapy, treść komunikatu jest tylko do odczytu. Jeśli aplikacja wywoła metodę Clear Body klasy Message, gdy treść komunikatu mapy jest tylko do odczytu, wówczas treść staje się czytelna i dostępna do zapisu. Metoda również czyści ciało.

Pojęcia pokrewne

Niejawne przekształcenie wartości właściwości z jednego typu danych na inny.
Gdy aplikacja pobiera wartość właściwości, wartość może zostać przekształcona przez produkt XMS na inny typ danych. Wiele reguł decyduje o tym, które konwersje są obsługiwane i w jaki sposób program XMS wykonuje konwersje.

Odsyłacze pokrewne

Typy danych dla elementów danych aplikacji
Aby aplikacja XMS mogła wymieniać komunikaty z aplikacją IBM MQ classes for JMS , zarówno aplikacje, jak i aplikacje muszą być w stanie interpretować dane aplikacji w treści komunikatu w ten sam sposób.

Komunikaty bajtowe
Treść komunikatu bajtowego zawiera strumień bajtów. Jednostka zawiera tylko rzeczywiste dane, a odpowiedzialność za ich interpretowanie i odbieranie polega na wysyłaniu i odbierającym.

Komunikaty obiektu
Treść komunikatu obiektu zawiera serializowany obiektJava lub .NET .

Komunikaty strumienia
Treść komunikatu strumienia zawiera strumień wartości, w którym każda wartość ma powiązany typ danych.

Komunikaty tekstowe
Treść wiadomości tekstowej zawiera łańcuch.

IMapMessage (dla interfejsu .NET)
Komunikat odwzorowania to komunikat, którego treść składa się z zestawu par nazwa-wartość, w których każda wartość ma powiązany typ danych.

Komunikaty obiektu

Treść komunikatu obiektu zawiera serializowany obiektJava lub .NET .

Aplikacja XMS może odbierać komunikat obiektu, zmieniać jego pola nagłówka i właściwości, a następnie wysyłać je do innego miejsca docelowego. Aplikacja może również skopiować treść komunikatu obiektu i użyć go do utworzenia innego komunikatu obiektu. Program XMS traktuje treść komunikatu obiektu w postaci tablicy bajtów.

Po utworzeniu przez aplikację komunikatu obiektu, treść komunikatu jest czytelna i dostępna do zapisu. Po wysłaniu przez aplikację wiadomości treść pozostaje czytelna i dostępna do zapisu. Gdy aplikacja odbierze komunikat o obiekcie, treść komunikatu jest tylko do odczytu. Jeśli aplikacja wywoła metodę `Clear Body` interfejsu IMessage dla .NET , gdy treść komunikatu obiektu jest tylko do odczytu, wówczas treść staje się czytelna i dostępna do zapisu. Metoda również czyści ciało.

Odsyłacze pokrewne

Typy danych dla elementów danych aplikacji
Aby aplikacja XMS mogła wymieniać komunikaty z aplikacją IBM MQ classes for JMS , zarówno aplikacje, jak i aplikacje muszą być w stanie interpretować dane aplikacji w treści komunikatu w ten sam sposób.

Komunikaty bajtowe
Treść komunikatu bajtowego zawiera strumień bajtów. Jednostka zawiera tylko rzeczywiste dane, a odpowiedzialność za ich interpretowanie i odbieranie polega na wysyłaniu i odbierającym.

Komunikaty mapy
Treść komunikatu mapy zawiera zestaw par nazwa-wartość, w których każda wartość ma powiązany typ danych.

Komunikaty strumienia
Treść komunikatu strumienia zawiera strumień wartości, w którym każda wartość ma powiązany typ danych.

Komunikaty tekstowe
Treść wiadomości tekstowej zawiera łańcuch.

IObjectMessage (dla interfejsu .NET)
Komunikat obiektu to komunikat, którego treść składa się z serializowanego obiektu Java lub .NET.

Komunikaty strumienia

Treść komunikatu strumienia zawiera strumień wartości, w którym każda wartość ma powiązany typ danych.

Typ danych wartości to jeden z typów danych produktu XMS wymienionych w sekcji Tabela 21 na stronie 78.

Po utworzeniu przez aplikację komunikatu strumienia treść komunikatu jest zapisywalny. Aplikacja zestawia dane aplikacji do treści, wywołując odpowiednie metody zapisu w interfejsie IStreamMessage dla produktu .NET. Za każdym razem, gdy aplikacja zapisuje wartość do strumienia komunikatów, wartość, a jej typ danych są składane natychmiast po poprzedniej wartości zapisanej przez aplikację. XMS utrzymuje kursor wewnętrzny, aby pamiętać o pozycji ostatniej zmontowanej wartości.

Po wysłaniu komunikatu przez aplikację treść komunikatu staje się dostępna tylko do odczytu. W tym trybie aplikacja może wysłać komunikat wiele razy.

Gdy aplikacja odbierze komunikat strumienia, treść komunikatu jest tylko do odczytu. Aplikacja może użyć odpowiednich metod odczytu interfejsu IStreamMessage dla produktu .NET w celu odczytania treści strumienia komunikatów. Aplikacja odczyta wartości w sekwencji, a program XMS utrzymuje kursor wewnętrzny, aby pamiętać o pozycji ostatniej odczytanej wartości.

Gdy aplikacja odczytuje wartość ze strumienia komunikatów, wartość może zostać przekształcona przez program XMS na inny typ danych. Na przykład, aby odczytać liczbę całkowitą ze strumienia komunikatów, aplikacja może wywołać metodę ReadString , która zwraca liczbę całkowitą jako łańcuch. Obsługiwane konwersje są takie same, jak te, które są obsługiwane, gdy program XMS przekształca wartość właściwości z jednego typu danych na inny. Więcej informacji na temat obsługiwanych konwersji zawiera sekcja “Niejawne przekształcenie wartości właściwości z jednego typu danych na inny.” na stronie 41.

Jeśli błąd wystąpi podczas próby odczytania wartości ze strumienia komunikatów przez aplikację, kursor nie jest zaawansowany. Aplikacja może wykonać odtwarzanie po wystąpieniu błędu, próbując odczytać wartość jako inny typ danych.

Jeśli aplikacja wywoła metodę Reset interfejsu IStreamMessage dla XMS , gdy treść komunikatu strumienia jest tylko do zapisu, wówczas treść staje się tylko do odczytu. Metoda ta umożliwia również ponowne pozycjonowanie kursora na początku strumienia komunikatów.

Jeśli aplikacja wywoła metodę `Clear Body` interfejsu IMessage dla XMS , gdy treść komunikatu strumienia jest tylko do odczytu, wówczas treść staje się tylko do zapisu. Metoda również czyści ciało.

Pojęcia pokrewne

Niejawne przekształcenie wartości właściwości z jednego typu danych na inny.
Gdy aplikacja pobiera wartość właściwości, wartość może zostać przekształcona przez produkt XMS na inny typ danych. Wiele reguł decyduje o tym, które konwersje są obsługiwane i w jaki sposób program XMS wykonuje konwersje.

Odsyłacze pokrewne

Typy danych dla elementów danych aplikacji
Aby aplikacja XMS mogła wymieniać komunikaty z aplikacją IBM MQ classes for JMS , zarówno aplikacje, jak i aplikacje muszą być w stanie interpretować dane aplikacji w treści komunikatu w ten sam sposób.

Komunikaty bajtowe
Treść komunikatu bajtowego zawiera strumień bajtów. Jednostka zawiera tylko rzeczywiste dane, a odpowiedzialność za ich interpretowanie i odbieranie polega na wysyłaniu i odbierającym.

Komunikaty mapy
Treść komunikatu mapy zawiera zestaw par nazwa-wartość, w których każda wartość ma powiązany typ danych.

Komunikaty obiektu
Treść komunikatu obiektu zawiera serializowany obiektJava lub .NET .

Komunikaty tekstowe
Treść wiadomości tekstowej zawiera łańcuch.

IStreamMessage (dla interfejsu .NET)
Komunikat strumienia to komunikat, którego treść zawiera strumień wartości, w którym każda wartość ma powiązany typ danych. Treść treści jest zapisywana i odczytywaana sekwencyjnie.

Komunikaty tekstowe

Treść wiadomości tekstowej zawiera łańcuch.

Po utworzeniu przez aplikację wiadomości tekstowej treść wiadomości jest czytelna i dostępna do zapisu. Po wysłaniu przez aplikację wiadomości treść pozostaje czytelna i dostępna do zapisu. Gdy aplikacja otrzymuje wiadomość tekstową, treść wiadomości jest tylko do odczytu. Jeśli aplikacja wywoła metodę Clear Body interfejsu IMessage dla programu .NET , gdy treść komunikatu tekstowego jest tylko do odczytu, wówczas treść staje się czytelna i dostępna do zapisu. Metoda również czyści ciało.

Odsyłacze pokrewne

Typy danych dla elementów danych aplikacji
Aby aplikacja XMS mogła wymieniać komunikaty z aplikacją IBM MQ classes for JMS , zarówno aplikacje, jak i aplikacje muszą być w stanie interpretować dane aplikacji w treści komunikatu w ten sam sposób.

Komunikaty bajtowe
Treść komunikatu bajtowego zawiera strumień bajtów. Jednostka zawiera tylko rzeczywiste dane, a odpowiedzialność za ich interpretowanie i odbieranie polega na wysyłaniu i odbierającym.

Komunikaty mapy
Treść komunikatu mapy zawiera zestaw par nazwa-wartość, w których każda wartość ma powiązany typ danych.

Komunikaty obiektu
Treść komunikatu obiektu zawiera serializowany obiektJava lub .NET .

Komunikaty strumienia
Treść komunikatu strumienia zawiera strumień wartości, w którym każda wartość ma powiązany typ danych.

ITextMessage (dla interfejsu .NET)
Komunikat tekstowy jest komunikatem, którego treść składa się z łańcucha.

Selektory komunikatów

Aplikacja XMS używa selektorów komunikatów w celu wybrania komunikatów, które mają zostać odebrane.

Gdy aplikacja tworzy konsument komunikatów, może powiązać wyrażenie selektora komunikatów z konsumentem. Wyrażenie selektora komunikatów określa kryteria wyboru.

Gdy aplikacja łączy się z menedżerem kolejek produktu IBM WebSphere MQ 7.0 , wybór komunikatów jest wybierany po stronie menedżera kolejek. Produkt XMS nie wykonuje żadnego wyboru i po prostu dostarcza komunikat, który otrzymał od menedżera kolejek, co zapewnia lepszą wydajność.

Aplikacja może utworzyć więcej niż jeden konsument komunikatów, każdy z własnym wyrażeniem selektora komunikatów. Jeśli komunikat przychodzący spełnia kryteria wyboru więcej niż jednego konsumenta komunikatów, produkt XMS dostarcza komunikat do każdego z tych konsumentów.

Wyrażenie selektora komunikatów może odwoływać się do następujących właściwości komunikatu:

- Właściwości zdefiniowane przez JMS
- Właściwości zdefiniowane przez IBM
- Właściwości zdefiniowane przez aplikację

Może on również odwoływać się do następujących pól nagłówka komunikatu:

- JMSCorrelationID
- JMSDeliveryMode
- JMSMessageID
- JMSPriority
- JMSTimestamp
- JMSType

Wyrażenie selektora komunikatów nie może jednak odwoływać się do danych w treści komunikatu.

Poniżej przedstawiono przykład wyrażenia selektora komunikatów:

```
JMSPriority > 3 AND manufacturer = 'Jaguar' AND model in ('xj6','xj12')
```

Produkt XMS dostarcza komunikat do konsumenta komunikatów z tym wyrażeniem selektora komunikatów tylko wtedy, gdy ma on priorytet większy niż 3; właściwość zdefiniowana przez aplikację, producent, o wartości `Jaguar;` i innej zdefiniowanej właściwości aplikacji, modelu z wartością `xj6` lub `xj12.`.

Reguły składni służące do tworzenia wyrażenia selektora komunikatów w programie XMS są takie same, jak w przypadku produktu IBM MQ classes for JMS. Informacje na temat konstruowania wyrażenia selektora komunikatów można znaleźć w dokumentacji produktu IBM MQ . Należy zwrócić uwagę, że w wyrażeniu selektora komunikatów nazwy właściwości zdefiniowanych przez produkt JMSmuszą być nazwami JMS, a nazwy właściwości zdefiniowanych w produkcie IBMmuszą być nazwami IBM MQ classes for JMS . Nie można używać nazw XMS w wyrażeniu selektora komunikatów.

Odsyłacze pokrewne

Części komunikatu produktu XMS
Komunikat XMS składa się z nagłówka, zestawu właściwości i treści.

Pola nagłówka w komunikacie XMS
Aby umożliwić aplikacji XMS wymianę komunikatów z aplikacją WebSphere JMS , nagłówek komunikatu XMS zawiera pola nagłówka komunikatu produktu JMS .

Właściwości komunikatu XMS

Produkt XMS obsługuje trzy rodzaje właściwości komunikatu: JMS zdefiniowane właściwości, IBM zdefiniowane właściwości i właściwości zdefiniowane przez aplikację.

Treść komunikatu produktu XMS
Treść komunikatu zawiera dane aplikacji. Jednak komunikat nie może mieć treści i zawierać tylko pola nagłówka i właściwości.

Odwzorowywanie komunikatów produktu XMS na komunikaty produktu IBM MQ
Pola nagłówka JMS i właściwości komunikatu produktu XMS są odwzorowywane na pola w strukturach nagłówka komunikatu produktu IBM MQ .

Odwzorowywanie komunikatów produktu XMS na komunikaty produktu IBM MQ

Pola nagłówka JMS i właściwości komunikatu produktu XMS są odwzorowywane na pola w strukturach nagłówka komunikatu produktu IBM MQ .

Gdy aplikacja XMS jest połączona z menedżerem kolejek produktu IBM MQ , komunikaty wysyłane do menedżera kolejek są odwzorowywane na komunikaty produktu IBM MQ w taki sam sposób, w jaki komunikaty produktu IBM MQ classes for JMS są odwzorowywane na komunikaty produktu IBM MQ w podobnych okolicznościach.

Jeśli właściwość XMSC_WMQ_TARGET_CLIENT obiektu docelowego jest ustawiona na wartość XMSC_WMQ_TARGET_DEST_JMS, pola nagłówka JMS i właściwości komunikatu wysyłanego do miejsca docelowego są odwzorowywane na pola w strukturach nagłówka MQMD i MQRFH2 w komunikacie IBM MQ . Ustawienie w ten sposób właściwości XMSC_WMQ_TARGET_CLIENT zakłada, że aplikacja, która odbiera komunikat, może obsługiwać nagłówek MQRFH2 . Aplikacją odbierającą może być więc inna aplikacja XMS , aplikacja IBM MQ classes for JMS lub rodzima aplikacja IBM MQ , która została zaprojektowana do obsługi nagłówka MQRFH2 .

Jeśli właściwość XMSC_WMQ_TARGET_CLIENT obiektu docelowego jest ustawiona na wartość XMSC_WMQ_TARGET_DEST_MQ, pola nagłówka JMS i właściwości komunikatu wysyłanego do miejsca docelowego są odwzorowywane na pola w strukturze nagłówka MQMD komunikatu IBM MQ . Komunikat nie zawiera nagłówka MQRFH2 , a wszystkie pola nagłówka JMS i właściwości, które nie mogą zostać odwzorowane na pola w strukturze nagłówka MQMD, są ignorowane. Aplikacja, która odbiera komunikat, może być więc rodzimą IBM MQ , która nie jest przeznaczona do obsługi nagłówka MQRFH2 .

Komunikaty produktu IBM MQ odebrane z menedżera kolejek są odwzorowywane na komunikaty produktu XMS w taki sam sposób, w jaki komunikaty produktu IBM MQ są odwzorowywane na komunikaty produktu IBM MQ classes for JMS w podobnych okolicznościach.

Jeśli przychodzący komunikat IBM MQ ma nagłówek MQRFH2 , wynikowy komunikat XMS ma treść, której typ jest określany przez wartość właściwości **Msd** zawartej w folderze `mcd` nagłówka MQRFH2 . Jeśli właściwość **Msd** nie znajduje się w nagłówku MQRFH2 lub jeśli komunikat IBM MQ nie ma nagłówka MQRFH2 , wynikowy komunikat XMS będzie miał treść, której typ jest określany na podstawie wartości pola *`Format`* w nagłówku MQMD. Jeśli pole *`Format`* jest ustawione na wartość MQFMT_STRING, komunikat XMS jest komunikatem tekstowym. W przeciwnym razie komunikat XMS jest komunikatem bajtów. Jeśli w komunikacie IBM MQ nie ma nagłówka MQRFH2 , ustawiane są tylko te pola nagłówka JMS i właściwości, które mogą pochodzić z pól w nagłówku MQMD.

Więcej informacji na temat odwzorowywania komunikatów programu IBM MQ classes for JMS na komunikaty produktu IBM MQ można znaleźć w dokumentacji produktu IBM MQ .

Odsyłacze pokrewne

Części komunikatu produktu XMS
Komunikat XMS składa się z nagłówka, zestawu właściwości i treści.

Pola nagłówka w komunikacie XMS
Aby umożliwić aplikacji XMS wymianę komunikatów z aplikacją WebSphere JMS , nagłówek komunikatu XMS zawiera pola nagłówka komunikatu produktu JMS .

Właściwości komunikatu XMS

Produkt XMS obsługuje trzy rodzaje właściwości komunikatu: JMS zdefiniowane właściwości, IBM zdefiniowane właściwości i właściwości zdefiniowane przez aplikację.

Treść komunikatu produktu XMS
Treść komunikatu zawiera dane aplikacji. Jednak komunikat nie może mieć treści i zawierać tylko pola nagłówka i właściwości.

Selektory komunikatów
Aplikacja XMS używa selektorów komunikatów w celu wybrania komunikatów, które mają zostać odebrane.

Odczytywanie i zapisywanie deskryptora komunikatu z aplikacji produktu IBM Message Service Client for .NET

Można uzyskać dostęp do wszystkich pól deskryptora komunikatu (MQMD) komunikatu IBM MQ z wyjątkiem pól StrucId i Version; BackoutCount można odczytać, ale nie zapisywać do niego. Ta funkcja jest dostępna tylko wtedy, gdy nawiąże połączenie z menedżerem kolejek produktu IBM WebSphere MQ 6.0 lub nowszego i jest kontrolowana przez właściwości miejsca docelowego opisane w dalszej części.

Atrybuty komunikatów udostępniane przez produkt IBM Message Service Client for .NET ułatwiają aplikacjom produktu XMS ustawianie pól MQMD, a także na potrzeby obsługi aplikacji IBM WebSphere MQ .

Niektóre ograniczenia mają zastosowanie podczas przesyłania komunikatów w trybie publikowania/subskrypcji. Na przykład, pola MQMD, takie jak MsgID i CorrelId, jeśli są ustawione, są ignorowane.

Funkcja opisana w tym temacie jest niedostępna w przypadku przesyłania komunikatów w trybie publikowania/subskrypcji podczas nawiązywania połączenia z menedżerem kolejek produktu IBM WebSphere MQ 6.0 . Jest ona również niedostępna, gdy właściwość **PROVIDERVERSION** jest ustawiona na wartość 6.

Uzyskiwanie dostępu do danych komunikatu produktu IBM MQ z aplikacji IBM Message Service Client for .NET

Można uzyskać dostęp do pełnych danych komunikatu produktu IBM MQ , w tym nagłówka MQRFH2 (jeśli istnieje) i innych nagłówków produktu IBM MQ (jeśli istnieją) w aplikacji produktu IBM Message Service Client for .NET jako treści elementu JMSBytesMessage.

Funkcja opisana w tym temacie jest dostępna tylko wtedy, gdy nawiąże połączenie z menedżerem kolejek produktu IBM WebSphere MQ 7.0 lub późniejszym, a dostawca przesyłania komunikatów produktu WebSphere MQ jest w trybie normalnym.

Właściwości obiektu docelowego określają, w jaki sposób aplikacja XMS uzyskuje dostęp do całego komunikatu produktu IBM MQ (w tym nagłówek MQRFH2 , jeśli jest obecny) jako treść komunikatu JMSBytesMessage.

Rozwiązywanie problemów

Ta sekcja zawiera informacje pomocne w wykrywaniu i rozwiązywaniu problemów podczas korzystania z produktu IBM Message Service Client for .NET.

Sekcja obejmuje następujące tematy:

Konfiguracja śledzenia dla aplikacji produktu .NET

W przypadku aplikacji XMS .NET można skonfigurować śledzenie z pliku konfiguracyjnego aplikacji, a także ze zmiennych środowiskowych programu XMS . Istnieje możliwość wybrania komponentów, które mają być śledzeniem. Śledzenie jest zwykle używane zgodnie z wytycznymi działu wsparcia produktu IBM .

Śledzenie dla XMS .NET jest oparte na standardowej infrastrukturze śledzenia .NET .

Wszystkie śledzenie, z wyjątkiem śledzenia błędów, jest domyślnie wyłączone. Można włączyć śledzenie i skonfigurować ustawienia śledzenia w jeden z następujących sposobów:

- Za pomocą pliku konfiguracyjnego aplikacji o nazwie, która składa się z nazwy programu wykonywalnego, do którego odnosi się plik, z przyrostkiem `.config`. Na przykład plik konfiguracyjny aplikacji dla pliku text.exe będzie miał nazwę text.exe.config. Użycie pliku konfiguracyjnego aplikacji jest preferowanym sposobem włączania śledzenia dla aplikacji XMS .NET . Więcej informacji na ten temat zawiera sekcja "Śledzenie konfiguracji przy użyciu pliku konfiguracyjnego aplikacji" na stronie 87.
- Korzystanie ze zmiennych środowiskowych XMS w przypadku aplikacji XMS C lub C + +. Więcej informacji na ten temat zawiera sekcja "Śledzenie konfiguracji przy użyciu zmiennych środowiskowych XMS" na stronie 88.

Aktywny plik śledzenia ma nazwę formatu xms_trace*PID*.log, gdzie *PID* reprezentuje identyfikator procesu aplikacji. Wielkość aktywnego pliku śledzenia jest domyślnie ograniczona do 20 MB. Po osiągnięciu tego limitu nazwa pliku jest zmieniana i archiwizowana. Zarchiwizowane pliki mają nazwy w formacie xms_trace*PID_YY.MM.DD_HH.MM.SS*.log

Domyślnie liczba zachowanych plików śledzenia wynosi cztery, to znaczy jeden aktywny plik i trzy zarchiwizowane pliki. Te cztery pliki są używane jako bufor kroczący do momentu zatrzymania aplikacji, z najstarszym plikiem, który jest usuwany i zastępowany przez najnowszy plik. Liczbę plików śledzenia można zmienić, podając inną liczbę w pliku konfiguracyjnym aplikacji. Jednak muszą istnieć co najmniej dwa pliki (jeden aktywny plik i jeden zarchiwizowany plik).

Dostępne są dwa formaty plików śledzenia:

- Pliki śledzenia podstawowego formatu są czytelne dla użytkownika w formacie WebSphere Application Server . Ten format jest domyślnym formatem pliku śledzenia. Podstawowy format nie jest zgodny z narzędziami analizatora śledzenia.
- Pliki śledzenia zaawansowanego formatu są zgodne z narzędziami analizatora śledzenia. Konieczne jest określenie, że pliki śledzenia mają być generowane w formacie zaawansowanym w pliku konfiguracyjnym aplikacji.

Pozycje śledzenia zawierają następujące informacje:

- Data i godzina zarejestrowania śledzenia
- Nazwa klasy
- Typ śledzenia
- Komunikat śledzenia

W poniższym przykładzie przedstawiono wyciąg z pewnego śladu:

```
[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
```

W poprzednim przykładzie format jest następujący:

```
[Date Time:Microsecs]         Thread-id    Classname        Trace-type    Methodname    Entry
or Exit
```

gdzie `Trace-type` to:

> dla pozycji
< dla wyjścia
d dla informacji debugowania

Śledzenie konfiguracji przy użyciu pliku konfiguracyjnego aplikacji

Preferowanym sposobem konfigurowania śledzenia dla aplikacji XMS .NET jest plik konfiguracyjny aplikacji. Sekcja śledzenia tego pliku zawiera parametry, które definiują, co ma być śledzone, położenie pliku śledzenia i maksymalną dozwoloną wielkość, liczbę używanych plików śledzenia oraz format pliku śledzenia.

Aby włączyć śledzenie za pomocą pliku konfiguracyjnego aplikacji, należy po prostu umieścić ten plik w tym samym katalogu, co plik wykonywalny dla aplikacji.

Śledzenie może być włączone zarówno przez komponent, jak i typ śledzenia. Możliwe jest również włączenie śledzenia dla całej grupy śledzenia. Śledzenie komponentów w hierarchii można włączyć pojedynczo lub grupowo. Dostępne są następujące typy śledzenia:

- Śledzenie debugowania
- Dane śledzenia wyjątku
- Ostrzeżenia, komunikaty informacyjne i komunikaty o błędach
- Śledzenie wejścia i wyjścia metody

W poniższym przykładzie przedstawiono ustawienia śledzenia zdefiniowane w sekcji Śledzenie pliku konfiguracyjnego aplikacji:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <configSections>
        <sectionGroup name="IBM.XMS">
            <section name="Trace"
                     type="System.Configuration.SingleTagSectionHandler"/>
        </sectionGroup>
    </configSections>

    <IBM.XMS>
        <Trace traceSpecification="*=all=enabled" traceFilePath=""
               traceFileSize="20000000" traceFileNumber="3"
               traceFormat="advanced"/>
    </IBM.XMS>
 </configuration>
```

Tabela 22 na stronie 87 szczegółowo opisuje ustawienia parametrów.

Tabela 22. Ustawienia parametrów śledzenia pliku konfiguracyjnego aplikacji

Parametr	Opis
`traceSpecification=`*`ComponentName`*`=`*`type`*`=`*`state`*	*`ComponentName`* to nazwa klasy, która ma być śledzona. W nazwie tej można użyć znaku wieloznacznego *. Na przykład: `*=all=enabled` określa, że mają być śledzeniem wszystkich klas, a `IBM.XMS.impl.*=all=enabled` określa, że wymagane jest śledzenie interfejsu API. *`type`* może być dowolnym z następujących typów śledzenia: • Wszystkie • debuguj • zdarzenie • EntryExit Program *`state`* może być włączony lub wyłączony. Wiele elementów śledzenia można łańcuchować razem za pomocą separatora: ':' (dwukropek).

Tabela 22. Ustawienia parametrów śledzenia pliku konfiguracyjnego aplikacji (kontynuacja)	
Parametr	**Opis**
traceFilePath="*filename*"	Jeśli ścieżka traceFilenie zostanie określona, lub jeśli ścieżka traceFilejest obecna, ale zawiera pusty łańcuch, plik śledzenia zostanie umieszczony w bieżącym katalogu. Aby zapisać plik śledzenia w nazwanym katalogu, należy określić nazwę katalogu w ścieżce traceFile, na przykład: `traceFilePath="c:\somepath"`
traceFileSize="*size*"	Maksymalna dozwolona wielkość pliku śledzenia. Gdy plik osiągnie tę wielkość, zostanie zarchiwizowany i jego nazwa zostanie zmieniona. Domyślna wartość maksymalna to 20 kB, która jest określona jako `traceFileSize="20000000"`.
traceFileNumber="*number*"	Liczba plików śledzenia, które mają zostać zachowane. Wartością domyślną jest 4 (jeden aktywny plik i 3 pliki archiwum). Minimalna dozwolona liczba to 2.
traceFormat="*format*"	Domyślny format śledzenia jest podstawowy. Pliki śledzenia są tworzone w tym formacie, jeśli użytkownik określi opcję `traceFormat="basic"` lub jeśli nie zostanie określony parametr traceFormat, lub jeśli traceFormat jest obecny, ale zawiera pusty łańcuch. Jeśli wymagane jest śledzenie, które jest zgodne z narzędziami analizatora śledzenia, należy określić wartość `traceFormat="advanced"`.

Ustawienia śledzenia w pliku konfiguracyjnym aplikacji są dynamiczne. Są one reread za każdym razem, gdy plik jest składowany lub zastępowany. Jeśli błędy zostaną znalezione w pliku po jego zmodyfikowaniu, ustawienia pliku śledzenia zostaną przywróceni do wartości domyślnych.

Pojęcia pokrewne

Śledzenie konfiguracji przy użyciu zmiennych środowiskowych XMS

Alternatywą dla korzystania z pliku konfiguracyjnego aplikacji jest włączenie śledzenia przy użyciu zmiennych środowiskowych programu XMS . Te zmienne środowiskowe są używane tylko wtedy, gdy w pliku konfiguracyjnym aplikacji nie ma specyfikacji śledzenia.

Śledzenie konfiguracji przy użyciu zmiennych środowiskowych XMS

Alternatywą dla korzystania z pliku konfiguracyjnego aplikacji jest włączenie śledzenia przy użyciu zmiennych środowiskowych programu XMS . Te zmienne środowiskowe są używane tylko wtedy, gdy w pliku konfiguracyjnym aplikacji nie ma specyfikacji śledzenia.

Aby skonfigurować śledzenie dla aplikacji XMS .NET , przed uruchomieniem aplikacji należy ustawić następujące zmienne środowiskowe:

Tabela 23. Ustawienia zmiennych środowiskowych dla śledzenia środowiska .NET			
Zmienne środowiskowe	**Domyślny**	**Ustawienia**	**Znaczenie**
XMS_TRACE_ON	Nie dotyczy	Nie dotyczy: wartość tej zmiennej jest ignorowana	Jeśli ustawiona jest wartość XMS_TRACE_ON, domyślnie włączone są wszystkie dane śledzenia.

Tabela 23. Ustawienia zmiennych środowiskowych dla śledzenia środowiska .NET (kontynuacja)

Zmienne środowiskowe	Domyślny	Ustawienia	Znaczenie
XMS_TRACE_FILE_PATH	Bieżący katalog roboczy	/dirpath/	Ścieżka do katalogu, w którym zapisywane są rekordy śledzenia i FFDC. Program XMS tworzy pliki FFDC i pliki śledzenia w bieżącym katalogu roboczym, chyba że zostanie określone alternatywne położenie. Alternatywne położenie można określić, ustawiając zmienną środowiskową XMS_TRACE_FILE_PATH na pełną nazwę ścieżki do katalogu, w którym produkt XMS ma tworzyć pliki FFDC i pliki śledzenia. Przed uruchomieniem aplikacji, która ma być śledzona, należy ustawić tę zmienną środowiskową. Należy upewnić się, że identyfikator użytkownika, na podstawie którego działa aplikacja, ma uprawnienia do zapisu w katalogu, w którym produkt XMS tworzy pliki FFDC i pliki śledzenia.
FORMAT XMS_TRE_FORMAT	BASIC	BASIC, ZAAWANSOWANE	Określa wymagany format śledzenia, który może mieć wartość BASIC lub ADVANCED. Domyślny format to BASIC. Format ADVANCED jest zgodny z narzędziami analizatora śledzenia.
XMS_TRACE_SPECIFACJA	Nie dotyczy	Patrz sekcja “Śledzenie konfiguracji przy użyciu pliku konfiguracyjnego aplikacji” na stronie 87	Przesłania specyfikację śledzenia, która jest zgodna z formatem określonym w produkcie “Śledzenie konfiguracji przy użyciu pliku konfiguracyjnego aplikacji” na stronie 87.

Pojęcia pokrewne

Śledzenie konfiguracji przy użyciu pliku konfiguracyjnego aplikacji

Preferowanym sposobem konfigurowania śledzenia dla aplikacji XMS .NET jest plik konfiguracyjny aplikacji. Sekcja śledzenia tego pliku zawiera parametry, które definiują, co ma być śledzone, położenie

pliku śledzenia i maksymalną dozwoloną wielkość, liczbę używanych plików śledzenia oraz format pliku śledzenia.

Konfiguracja FFDC dla aplikacji produktu .NET

W przypadku implementacji .NET produktu XMSdla każdego FFDC tworzony jest jeden plik FFDC.

Pliki FFDC (First Failure Data Capture) są przechowywane w czytelnych plikach tekstowych. Te pliki mają nazwy w postaci `xmsffdc`*`processID`*`_Date`T*`Timestamp`*`.txt`. Przykładem nazwy pliku jest `xmsffdc264_2006.01.06T13.18.52.990955.txt`. Datownik zawiera rozdzielczość mikrosekund.

Pliki rozpoczynają się od daty i godziny wystąpienia wyjątku, po którym następuje typ wyjątku. Pliki te zawierają unikalny krótki identyfikator probeId, który może zostać użyty do znalezienia miejsca, w którym wystąpił ten mechanizm FFDC.

Nie ma potrzeby przeprowadzania żadnej konfiguracji w celu włączenia FFDC. Domyślnie wszystkie pliki FFDC są zapisywane w bieżącym katalogu. Jeśli jednak jest to wymagane, można określić inny katalog, zmieniając wartość ffdcDirectory w sekcji Śledzenie pliku konfiguracyjnego aplikacji. W poniższym przykładzie wszystkie pliki śledzenia są rejestrowane w katalogu c:client\ffdc:

```
<IBM.XMS>
       <Trace ffdc=true ffdcDirectory="c:\client\ffdc"/>
</IBM.XMS>
```

Śledzenie można wyłączyć, ustawiając wartość false na wartość false w sekcji Śledzenie pliku konfiguracyjnego aplikacji.

Jeśli nie jest używany plik konfiguracyjny aplikacji, narzędzie FFDC jest wyłączone, a śledzenie jest wyłączone.

Wskazówki dotyczące rozwiązywania problemów

Te wskazówki ułatwiają rozwiązywanie problemów związanych z używaniem produktu XMS.

Aplikacja XMS nie może nawiązać połączenia z menedżerem kolejek (MQRC_NOT_AUTHORIZED)

Klient XMS .NET może mieć różne zachowania związane z działaniem klienta IBM MQ JMS . Oznacza to, że aplikacja XMS nie może nawiązać połączenia z menedżerem kolejek, chociaż aplikacja JMS może być używana.

- Prostym rozwiązaniem tego problemu jest próba użycia identyfikatora użytkownika, którego długość nie przekracza 12 znaków i jest ona całkowicie autoryzowana na liście uprawnień menedżera kolejek. Jeśli to rozwiązanie nie jest idealne, innym, ale bardziej złożonym podejściem, byłoby korzystanie z wyjść bezpieczeństwa. Jeśli potrzebna jest dalsza pomoc w tym wydaniu, skontaktuj się z działem wsparcia IBM w celu uzyskania pomocy.
- Jeśli właściwość XMSC_USERID fabryki połączeń zostanie ustawiona, musi ona być zgodna z identyfikatorem użytkownika i hasłem zalogowanego użytkownika. Jeśli ta właściwość nie zostanie ustawiona, menedżer kolejek domyślnie użyje ID użytkownika zalogowanego użytkownika.
- Uwierzytelnianie użytkownika dla produktu IBM MQ jest wykonywane przy użyciu szczegółów aktualnie zalogowanego użytkownika, a nie informacji podanych w sekcji XMSC.USERID i XMSC.PASSWORD . Ma to na celu zachowanie spójności z produktem IBM MQ. Więcej informacji na temat uwierzytelniania można znaleźć w sekcji *Informacje o uwierzytelnianiu* w elektronicznej dokumentacji produktu IBM MQ .

Połączenie przekierowane do mechanizmu przesyłania komunikatów

Po nawiązaniu połączenia z magistralą integracji usług produktu WebSphere Application Server 6.0.2 wszystkie połączenia mogą zostać przekierowane z pierwotnego punktu końcowego dostawcy do mechanizmu przesyłania komunikatów wybranego przez magistralę dla tego połączenia klienta. W takim przypadku będzie zawsze przekierowuje połączenie do serwera hosta określonego przez nazwę hosta,

a nie przez adres IP. Z tego powodu mogą wystąpić problemy z połączeniem, jeśli nie można przetłumaczyć nazwy hosta.

Aby pomyślnie nawiązać połączenie z magistralą integracji usług produktu WebSphere Application Server 6.0.2 , może być konieczne udostępnienie odwzorowania między nazwami hostów i adresami IP na komputerze hosta klienta. Na przykład można określić odwzorowanie w tabeli hostów lokalnych na komputerze hosta klienta.

Obsługa uwierzytelniania przy użyciu hasła typu telnet

Protokół XMS .NET Real Time Transport obsługuje tylko proste uwierzytelnianie przy użyciu hasła typu telnet. Protokół XMS .NET Real Time Transport nie obsługuje jakości ochrony.

Ustawianie wartości typu double dla właściwości typu double

Na 64-bitowej platformie Windows metody SetDoubleProperty () lub GetDoubleProperty () mogą nie działać poprawnie podczas ustawiania lub pobierania wartości typu double, jeśli wartości te są mniejsze niż Double.Epsilon.

Na przykład, jeśli użytkownik spróbuje ustawić wartość 4.9E-324 dla właściwości typu double, 64-bitowe platformy Windows traktują ją jako 0.0. Tak więc w rozproszonym środowisku przesyłania komunikatów, jeśli aplikacja JMS lub inna ustawia wartość dla podwójnej właściwości jako 4.9E-324 na dowolnym komputerze z systemem UNIX lub Windows 32-bitowym, a program XMS .NET działa na komputerze 64-bitowym, wartość zwrócona przez właściwość GetDoubleProperty () wynosi 0.0. Jest to znany problem z produktem Microsoft .NET Framework 2.0 Framework.

Warunki błędów, które mogą być obsługiwane w czasie wykonywania

Kody powrotu z wywołań funkcji API są warunkami błędów, które mogą być obsługiwane w czasie wykonywania. Sposób postępowania z tym typem błędu zależy od tego, czy używany jest interfejs API C, czy C++ .

W jaki sposób wykrywać błędy w czasie wykonywania

Jeśli aplikacja wywołuje funkcję API języka C, a wywołanie nie powiedzie się, odpowiedź z kodem powrotu innym niż XMS_OK jest zwracana z blokiem błędu XMS zawierającym więcej informacji o przyczynie niepowodzenia.

Funkcja API C++ zgłasza wyjątek, gdy używana jest metoda.

Aplikacja korzysta z obiektu nasłuchiwania wyjątków, który jest powiadamiany asynchronicznie o problemie z połączeniem. Program nasłuchujący wyjątków jest dostarczany i inicjowany za pomocą interfejsu API XMS C lub C++ .

W jaki sposób obsługiwać błędy w czasie wykonywania

Niektóre warunki błędu wskazują, że niektóre zasoby są niedostępne, a działanie, które może zająć aplikacja, zależy od funkcji XMS , do której aplikacja jest wywoła. Na przykład, jeśli połączenie z serwerem nie powiedzie się, aplikacja może okresowo ponawiać próby, aż do momentu nawiązania połączenia. Blok błędu lub wyjątek produktu XMS może nie zawierać wystarczającej ilości informacji, aby określić działanie, które należy wykonać, a w takich sytuacjach często występuje powiązany blok błędów lub wyjątek, który zawiera bardziej szczegółowe informacje diagnostyczne.

W interfejsie API języka C zawsze testuj dla odpowiedzi z kodem powrotu innym niż XMS_OK i zawsze przekaz blok błędu w wywołaniu API. Działanie podejmowane zwykle zależy od tego, która funkcja API jest używana przez aplikację.

W interfejsie API produktu C++ zawsze dołączane są wywołania metod w bloku try, a w celu wychwytu wszystkich typów wyjątków produktu XMS należy określić klasę wyjątku w konstrukcji catch.

Obiekt nasłuchiwania wyjątków jest ścieżką do asynchronicznego warunku błędu, która może być uruchomiona w dowolnym momencie. Gdy funkcja nasłuchiwania wyjątków jest uruchomiona, we własnym wątku jest zwykle wskazaniem poważniejszej awarii niż normalny warunek błędu API XMS . Można podjąć odpowiednie działania, ale należy zachować ostrożność, aby postępować zgodnie z regułami modelu wielowątkowego XMS , zgodnie z opisem w sekcji “Model wielowątkowości” na stronie 21.

Pojęcia pokrewne

Model wielowątkowości
Ogólne reguły zarządzają sposobem, w jaki aplikacja wielowątkowa może używać obiektów XMS .

Informacje o klientach usługi komunikatów dla środowiska .NET

Ta sekcja zawiera informacje pomocne przy korzystaniu z programu Message Service Client for .NET. Te informacje ułatwiają wykonywanie zadań związanych z programowaniem z produktem XMS.

.NET interfejsy

Ten sekcja dokumentuje interfejsy klasy .NET oraz ich właściwości i metody.

Poniższa tabela zawiera podsumowanie wszystkich interfejsów, które są zdefiniowane w przestrzeni nazw `IBM.XMS` .

Tabela 24. Podsumowanie interfejsów klasy .NET

Interfejs	**Opis**
“IBytesMessage” na stronie 95	Komunikat bajtów to komunikat, którego treść zawiera strumień bajtów.
“IConnection” na stronie 105	Obiekt połączenia reprezentuje aktywne połączenie aplikacji z serwerem przesyłania komunikatów.
“IConnectionFactory” na stronie 108	Aplikacja korzysta z fabryki połączeń w celu utworzenia połączenia.
“Dane IConnectionMeta” na stronie 110	Obiekt danych ConnectionMetaudostępnia informacje na temat połączenia.
“Miejsce docelowe” na stronie 110	Miejsce docelowe to miejsce, w którym aplikacja wysyła komunikaty lub jest to źródło, z którego aplikacja odbiera komunikaty, lub oba te komunikaty.
“ExceptionListener” na stronie 112	Aplikacja korzysta z obiektu nasłuchiwania wyjątków, który jest powiadamiany asynchronicznie o problemie z połączeniem.
“Wyjątek IllegalState” na stronie 113	XMS zgłasza ten wyjątek, jeśli aplikacja wywołuje metodę w niepoprawnym lub nieodpowiednim czasie lub jeśli XMS nie znajduje się w odpowiednim stanie dla żądanej operacji.
“InitialContext” na stronie 113	Aplikacja korzysta z obiektu InitialContext do tworzenia obiektów na podstawie definicji obiektów pobranych z repozytorium obiektów administrowanych.
“InvalidClientIDException” na stronie 115	XMS zgłasza ten wyjątek, jeśli aplikacja próbuje ustawić identyfikator klienta dla połączenia, ale identyfikator klienta nie jest poprawny lub jest już używany.

Tabela 24. Podsumowanie interfejsów klasy .NET (kontynuacja)	
Interfejs	**Opis**
“Wyjątek InvalidDestination” na stronie 116	XMS zgłasza ten wyjątek, jeśli w aplikacji określono niepoprawne miejsce docelowe.
“InvalidSelectorWyjątek” na stronie 116	XMS zgłasza ten wyjątek, jeśli aplikacja udostępnia wyrażenie selektora komunikatów, którego składnia nie jest poprawna.
“IMapMessage” na stronie 116	Komunikat odwzorowania to komunikat, którego treść składa się z zestawu par nazwa-wartość, w których każda wartość ma powiązany typ danych.
“Komunikat IMessage” na stronie 126	Obiekt komunikatu reprezentuje komunikat, który jest wysyłany lub odbierany przez aplikację. IMessage jest nadklasą dla klas komunikatów, takich jak IMapMessage.
“IMessageConsumer” na stronie 132	Aplikacja używa konsumenta komunikatów do odbierania komunikatów wysyłanych do miejsca docelowego.
“MessageEOFException” na stronie 135	XMS zgłasza ten wyjątek, jeśli program XMS napotka koniec strumienia komunikatów w bajtach, gdy aplikacja odczyta treść komunikatu bajtów.
“Wyjątek MessageFormat” na stronie 135	XMS zgłasza ten wyjątek, jeśli program XMS napotka komunikat o niepoprawnym formacie.
“IMessageListener (delegat)” na stronie 135	Aplikacja korzysta z funkcji nasłuchiwania komunikatów w celu asynchronicznego odbierania komunikatów.
“MessageNotReadableException” na stronie 136	XMS zgłasza ten wyjątek, jeśli aplikacja próbuje odczytać treść komunikatu, który jest tylko do zapisu.
“MessageNotWritableException” na stronie 136	XMS zgłasza ten wyjątek, jeśli aplikacja podejmie próbę zapisu w treści komunikatu, który jest tylko do odczytu.
“IMessageProducer” na stronie 136	Aplikacja korzysta z producenta komunikatów w celu wysyłania komunikatów do miejsca docelowego.
“IObjectMessage” na stronie 142	Komunikat obiektu to komunikat, którego treść składa się z serializowanego obiektu Java lub .NET.
“IPropertyContext” na stronie 143	IPropertyContext jest abstrakcyjną nadklasą, która zawiera metody, które zawierają i ustawia właściwości. Metody te są dziedziczone przez inne klasy.
“IQueueBrowser” na stronie 152	Aplikacja korzysta z przeglądarki kolejek w celu przeglądania komunikatów w kolejce bez usuwania ich.
“Żądający” na stronie 154	Aplikacja korzysta z requestera w celu wysłania komunikatu żądania, a następnie czekania na odpowiedź i odebrania odpowiedzi.

Tabela 24. Podsumowanie interfejsów klasy .NET (kontynuacja)	
Interfejs	**Opis**
“Wyjątek ResourceAllocation” na stronie 156	XMS zgłasza ten wyjątek, jeśli program XMS nie może przydzielić zasobów wymaganych przez metodę.
“SecurityException” na stronie 156	XMS zgłasza ten wyjątek, jeśli identyfikator użytkownika i hasło udostępnione w celu uwierzytelnienia aplikacji są odrzucane. XMS zgłasza również ten wyjątek, jeśli sprawdzenie uprawnień nie powiedzie się i uniemożliwia wykonanie metody.
“ISesja” na stronie 156	Sesja jest jednowątkowym kontekstem do wysyłania i odbierania komunikatów.
“IStreamMessage” na stronie 167	Komunikat strumienia to komunikat, którego treść zawiera strumień wartości, w którym każda wartość ma powiązany typ danych.
“ITextMessage” na stronie 177	Komunikat tekstowy jest komunikatem, którego treść składa się z łańcucha.
“TransactionInProgressException” na stronie 178	XMS zgłasza ten wyjątek, jeśli aplikacja żąda operacji, która nie jest poprawna, ponieważ transakcja jest w toku.
“TransactionRolledBackException” na stronie 178	XMS zgłasza ten wyjątek, jeśli aplikacja wywołuje funkcję Session.commit() w celu zatwierdzenia bieżącej transakcji, ale transakcja jest wycofana.
XMSC	W przypadku bazy danych .NETnazwy i wartości właściwości XMS są zdefiniowane w tej klasie jako stałe publiczne. Więcej informacji na ten temat zawiera sekcja “Właściwości obiektów XMS” na stronie 181.
“Wyjątek XMSException” na stronie 178	Jeśli program XMS wykryje błąd podczas przetwarzania wywołania metody .NET , XMS zgłasza wyjątek. Wyjątek stanowi obiekt, który hermetyzuje informacje o błędzie. Istnieją różne typy wyjątków XMS , a obiekt XMSException to tylko jeden typ wyjątku. Klasa XMSException jest jednak nadklasą innych klas wyjątków XMS . XMS zgłasza obiekt XMSException w sytuacjach, w których żaden z pozostałych typów wyjątków nie jest odpowiedni.
“XMSFactoryFactory” na stronie 179	Jeśli aplikacja nie używa administrowanych obiektów, należy użyć tej klasy w celu utworzenia fabryk połączeń, kolejek i tematów.

Definicja każdej metody zawiera listę kodów wyjątków, które mogą zostać zwrócone przez produkt XMS , jeśli wykryje błąd podczas przetwarzania wywołania metody. Każdy kod wyjątku jest reprezentowany przez jego stałą nazwaną, która ma odpowiedni wyjątek.

Pojęcia pokrewne

Budowanie własnych aplikacji
Użytkownik buduje własne aplikacje, takie jak kompilacja przykładowych aplikacji.

Pisanie aplikacji produktu XMS
Tematy w tej sekcji zawierają informacje pomocne podczas pisania aplikacji XMS .

Pisanie aplikacji XMS .NET
Tematy zawarte w tej sekcji zawierają informacje pomocne podczas pisania aplikacji XMS .NET .

Odsyłacze pokrewne
Właściwości obiektów XMS
Ta rozdział dokumentuje właściwości obiektu zdefiniowane przez produkt XMS.

IBytesMessage

Komunikat bajtów to komunikat, którego treść zawiera strumień bajtów.

Hierarchia dziedziczenia:

```
IBM.XMS.IPropertyContext
   |
   +----IBM.XMS.IMessage
           |
           +----IBM.XMS.IBytesMessage
```

Odsyłacze pokrewne
Komunikaty bajtowe
Treść komunikatu bajtowego zawiera strumień bajtów. Jednostka zawiera tylko rzeczywiste dane, a odpowiedzialność za ich interpretowanie i odbieranie polega na wysyłaniu i odbierającym.

Właściwości produktu .NET

BodyLength -Pobieranie Długości Ciała

Interfejs:

```
Int64 BodyLength
    {
      get;
    }
```

Pobieranie długości treści komunikatu w bajtach, gdy treść komunikatu jest tylko do odczytu.

Zwrócona wartość jest długością całego ciała, niezależnie od miejsca, w którym znajduje się kursor do odczytu komunikatu.

Wyjątki:

- Wyjątek XMSException
- MessageNotReadableException

Metody

ReadBoolean -odczytywanie wartości boolowskiej

Interfejs:

```
Boolean ReadBoolean();
```

Odczytywanie wartości boolowskiej ze strumienia komunikatów bajtów.

Parametry:
Brak

Zwraca:
Wartość boolowska, która jest odczytywalna.

Wyjątki:

- Wyjątek XMSException
- MessageNotReadableException
- MessageEOFException

ReadSignedByte-Odczyt bajtu

Interfejs:

```
Int16   ReadSignedByte();
```

Czytaj następny bajt ze strumienia komunikatów bajtów jako 8-bitowa liczba całkowita ze znakiem.

Parametry:
Brak

Zwraca:
Bajt, który jest odczytany.

Wyjątki:

- Wyjątek XMSException
- MessageNotReadableException
- MessageEOFException

ReadBytes -odczytane bajty

Interfejs:

```
Int32   ReadBytes(Byte[] array);
Int32   ReadBytes(Byte[] array, Int32 length);
```

Odczytywanie tablicy bajtów z strumienia komunikatów bajtów począwszy od bieżącej pozycji kursora.

Parametry:

tablica (wyjście)
Bufor, który ma zawierać tablicę bajtów, które są odczytywanie. Jeśli liczba bajtów pozostających do odczytania ze strumienia przed wywołaniem jest większa lub równa długości buforu, bufor jest wypełniany. W przeciwnym razie bufor zostanie częściowo zapełniony wszystkimi pozostałymi bajtami.

Jeśli w danych wejściowych zostanie określony pusty wskaźnik, metoda pomija bajty bez ich odczytu. Jeśli liczba bajtów pozostających do odczytania ze strumienia przed wywołaniem jest większa lub równa długości buforu, liczba pominiętych bajtów jest równa długości buforu. W przeciwnym razie wszystkie pozostałe bajty zostaną pominięte. Kursor pozostaje na następnej pozycji do odczytania w strumieniu komunikatów bajtowych.

długość (wejście)
Długość buforu w bajtach

Zwraca:
Liczba bajtów, które zostały odczytane w buforze. Jeśli bufor jest zapełniony częściowo, wartość jest mniejsza niż długość buforu, co oznacza, że nie ma więcej bajtów do odczytania. Jeśli przed wywołaniem nie pozostały żadne bajty, które mają zostać odczytane ze strumienia, wartością jest XMSC_END_OF_STREAM.

Jeśli w danych wejściowych zostanie określony pusty wskaźnik, metoda nie zwróci żadnej wartości.

Wyjątki:

- Wyjątek XMSException
- MessageNotReadableException

ReadChar -Odczyt Znaku

Interfejs:

```
Char    ReadChar();
```

Odczytaj następne 2 bajty ze strumienia komunikatów bajtów jako znak.

Parametry:
Brak

Zwraca:
Znak, który jest odczytyowany.

Wyjątki:

- Wyjątek XMSException
- MessageNotReadableException
- MessageEOFException

ReadDouble -Odczyt dwukrotnego numeru zmiennopozycyjnego o podwójnej precyzji

Interfejs:

```
Double  ReadDouble();
```

Przeczytaj następne 8 bajtów ze strumienia komunikatów bajtów jako liczbę zmiennopozycyjną podwójnej precyzji.

Parametry:
Brak

Zwraca:
Liczba zmiennopozycyjna o podwójnej precyzji, która jest odczytywaana.

Wyjątki:

- Wyjątek XMSException
- MessageNotReadableException
- MessageEOFException

ReadFloat -odczyt numeru punktu zmiennopozycyjnego

Interfejs:

```
Single  ReadFloat();
```

Odczytaj następne 4 bajty strumienia komunikatów bajtów jako liczbę zmiennopozycyjną.

Parametry:
Brak

Zwraca:
Liczba zmiennopozycyjna, która jest odczytywaana.

Wyjątki:

- Wyjątek XMSException
- MessageNotReadableException
- MessageEOFException

ReadInt -Odczyt typu Integer

Interfejs:

```
Int32   ReadInt();
```

Przeczytaj kolejne 4 bajty ze strumienia komunikatów bajtów jako 32-bitową liczbę całkowitą ze znakiem.

Parametry:
Brak

Zwraca:
Liczba całkowita, która jest odczytywaana.

Wyjątki:

- Wyjątek XMSException
- MessageNotReadableException
- MessageEOFException

ReadLong -Odczyt Long Integer

Interfejs:

```
Int64   ReadLong();
```

Odczytaj następne 8 bajtów ze strumienia komunikatów bajtów jako 64-bitową liczbę całkowitą ze znakiem.

Parametry:
Brak

Zwraca:
Długa liczba całkowita, która jest odczytywaana.

Wyjątki:

- Wyjątek XMSException
- MessageNotReadableException
- MessageEOFException

ReadShort -Read Short Integer

Interfejs:

```
Int16   ReadShort();
```

Odczytaj następne 2 bajty ze strumienia komunikatów bajtów jako 16-bitowa liczba całkowita ze znakiem.

Parametry:
Brak

Zwraca:
Krótka liczba całkowita, która jest odczytywaana.

Wyjątki:

- Wyjątek XMSException
- MessageNotReadableException
- MessageEOFException

ReadByte -Odczyt niepodpisanego bajtu

Interfejs:

```
Byte    ReadByte();
```

Czytaj następny bajt ze strumienia komunikatów bajtów jako 8-bitowa liczba całkowita bez znaku.

Parametry:
Brak

Zwraca:
Bajt, który jest odczytany.

Wyjątki:

- Wyjątek XMSException
- MessageNotReadableException
- MessageEOFException

ReadUnsignedShort-Read Unsigned Short Integer

Interfejs:

```
Int32   ReadUnsignedShort();
```

Odczytaj następne 2 bajty ze strumienia komunikatów bajtów jako 16-bitowa liczba całkowita bez znaku.

Parametry:
Brak

Zwraca:
Niepodpisana krótka liczba całkowita, która jest odczytywaana.

Wyjątki:

- Wyjątek XMSException
- MessageNotReadableException
- MessageEOFException

ReadUTF -Odczyt łańcucha UTF

Interfejs:

```
String  ReadUTF();
```

Odczytaj łańcuch, zakodowany w UTF-8, z strumienia komunikatów bajtów.

Uwaga: Przed wywołaniem metody ReadUTF() należy upewnić się, że kursor buforu wskazuje na początek strumienia komunikatów bajtu.

Parametry:
Brak

Zwraca:
Obiekt typu String obudowujący odczytany łańcuch.

Wyjątki:

- Wyjątek XMSException
- MessageNotReadableException
- MessageEOFException

Resetuj-Resetuj

Interfejs:

```
void Reset();
```

Umieść treść komunikatu w trybie tylko do odczytu i przełóż kursor na początku strumienia komunikatów bajtów.

Parametry:
Brak

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- MessageNotReadableException

WriteBoolean -zapis wartości boolowskiej

Interfejs:

```
void  WriteBoolean(Boolean value);
```

Zapis wartości boolowskiej do strumienia komunikatów bajtów.

Parametry:

wartość (wejście)
Wartość boolowska, która ma zostać zapisana.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

WriteByte -Zapis Bajt

Interfejs:

```
void  WriteByte(Byte value);
void  WriteSignedByte(Int16 value);
```

Zapis bajtu do strumienia komunikatów bajtów.

Parametry:

wartość (wejście)
Bajt, który ma zostać zapisany.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

WriteBytes -Zapis Bajtów

Interfejs:

```
void  WriteBytes(Byte[] value);
```

Zapis tablicy bajtów do strumienia komunikatów bajtów.

Parametry:

wartość (wejście)
Tablica bajtów, które mają zostać zapisane.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

WriteBytes -Zapis częściowych bajtów tablicy

Interfejs:

```
void  WriteBytes(Byte[] value, int offset, int length);
```

Zapis częściowej tablicy bajtów do strumienia komunikatów bajtów zgodnie z definicją podaną przez podaną długość.

Parametry:

wartość (wejście)
Tablica bajtów, które mają zostać zapisane.

przesunięcie (wejście)
Punkt początkowy tablicy bajtów, która ma zostać zapisana.

długość (wejście)
Liczba bajtów do zapisu.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

WriteChar -Zapis Znaku

Interfejs:

```
void  WriteChar(Char value);
```

Napisz znak do strumienia komunikatów bajtów jako 2 bajty, pierwszy bajt o wysokiej kolejności.

Parametry:

wartość (wejście)
Znak, który ma zostać zapisany.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

WriteDouble -Liczba Zmiennopozycyjna O Podwójnej Precyzji

Interfejs:

```
void  WriteDouble(Double value);
```

Konwertuj liczbę zmiennopozycyjną podwójnej precyzji na długą liczbę całkowitą, a następnie zapisz długą liczbę całkowitą w strumieniu komunikatów bajtów jako 8 bajtów, pierwszy bajt o wysokiej kolejności (first order byte first-bajt o wysokiej kolejności).

Parametry:

wartość (wejście)
Liczba zmiennopozycyjna podwójnej precyzji, która ma zostać zapisana.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

WriteFloat -numer zmiennopozycyjnego zapisu

Interfejs:

```
void  WriteFloat(Single value);
```

Przekształć liczbę zmiennopozycyjną w liczbę całkowitą i wpisz liczbę całkowitą do strumienia komunikatów bajtów jako 4 bajty, pierwszy bajt o wysokiej kolejności.

Parametry:

wartość (wejście)
Liczba zmiennopozycyjna, która ma zostać zapisana.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

WriteInt -Zapis Integer

Interfejs:

```
void  WriteInt(Int32 value);
```

Wpisz liczbę całkowitą do strumienia komunikatów bajtów jako 4 bajty, pierwszy bajt o wysokiej kolejności.

Parametry:

wartość (wejście)
Liczba całkowita, która ma zostać zapisana.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

WriteLong -zapis Long Integer

Interfejs:

```
void  WriteLong(Int64 value);
```

Zapis długiej liczby całkowitej do strumienia komunikatów bajtów jako 8 bajtów, pierwszy bajt o wysokiej kolejności.

Parametry:

wartość (wejście)
Długa liczba całkowita, która ma zostać zapisana.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

WriteObject -Zapis Obiektu

Interfejs:

```
void  WriteObject(Object value);
```

Zapisz określony obiekt w strumieniu komunikatów bajtowych.

Parametry:

wartość (wejście)
Obiekt, który ma zostać zapisany, który musi być odwołaniem do typu podstawowego.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

WriteShort -Zapis Short Integer

Interfejs:

```
void  WriteShort(Int16 value);
```

Wpisz krótką liczbę całkowitą do strumienia komunikatów bajtów jako 2 bajty, pierwsze bajt o wysokiej kolejności.

Parametry:

wartość (wejście)
Krótka liczba całkowita, która ma zostać zapisana.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

WriteUTF -zapis łańcucha UTF

Interfejs:

```
void  WriteUTF(String value);
```

Wpisz łańcuch, zakodowany w UTF-8, do strumienia komunikatów bajtów.

Parametry:

wartość (wejście)
Obiekt typu String obudowujący łańcuch, który ma zostać zapisany.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

Odziedziczone właściwości i metody

Następujące właściwości zostały odziedziczone po interfejsie IMessage:

JMSCorrelationID, JMSDeliveryMode, JMSDestination, JMSExpiration, JMSMessageID, JMSPriority, JMSRedelivered, JMSReplyTo, JMSTimestamp, JMSType, Properties

Następujące metody zostały odziedziczone po interfejsie IMessage:

clearBody, clearProperties, PropertyExists

Następujące metody zostały odziedziczone po interfejsie IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

IConnection

Obiekt połączenia reprezentuje aktywne połączenie aplikacji z serwerem przesyłania komunikatów.

Hierarchia dziedziczenia:

```
IBM.XMS.IPropertyContext
   |
   +----IBM.XMS.IConnection
```

Listę właściwości zdefiniowanych przez XMS obiektu Connection można znaleźć w sekcji “Właściwości połączenia” na stronie 182.

Właściwości produktu .NET

ClientID -pobieranie i ustawianie identyfikatora klienta

Interfejs:

```
String ClientID
      {
        get;
        set;
      }
```

Pobierz i ustaw identyfikator klienta dla połączenia.

Identyfikator klienta może być wstępnie skonfigurowany przez administratora w obiekcie ConnectionFactorylub przypisany przez ustawienie ClientID.

Identyfikator klienta jest używany tylko do obsługi trwałych subskrypcji w domenie publikowania/subskrybowania i jest ignorowany w domenie typu punkt z punktem.

Jeśli aplikacja ustawia identyfikator klienta dla połączenia, musi to zrobić natychmiast po utworzeniu połączenia, a przed wykonaniem dowolnej innej operacji na połączeniu. Jeśli po tym punkcie aplikacja próbuje ustawić identyfikator klienta, wywołanie zgłasza wyjątek IllegalState.

Ta właściwość nie jest poprawna dla połączenia w czasie rzeczywistym z brokerem.

Wyjątki:

- Wyjątek XMSException
- Wyjątek IllegalState
- InvalidClientIDException

ExceptionListener -pobieranie i ustawianie obiektu nasłuchiwania wyjątków

Interfejs:

```
ExceptionListener ExceptionListener
    {
      get;
      set;
    }
```

Pobierz program nasłuchujący wyjątków, który jest zarejestrowany w połączeniu, i zarejestruj obiekt nasłuchiwania wyjątków za pomocą połączenia.

Jeśli program nasłuchujący wyjątków nie jest zarejestrowany w połączeniu, metoda zwraca wartość NULL. Jeśli obiekt nasłuchiwania wyjątków jest już zarejestrowany w połączeniu, można anulować rejestrację, podając wartość NULL zamiast obiektu nasłuchiwania wyjątków.

Więcej informacji na temat używania programów nasłuchujących wyjątków zawiera sekcja “Programy nasłuchujące komunikatów i wyjątków w produkcie .NET” na stronie 49.

Wyjątki:

- Wyjątek XMSException

Metadane-pobierz metadane

Interfejs:

```
IConnectionMetaData MetaData
    {
      get;
    }
```

Pobierz metadane dla połączenia.

Wyjątki:

- Wyjątek XMSException

Metody

Zamknij-Zamknij połączenie

Interfejs:

```
void  Close();
```

Zamknij połączenie.

Jeśli aplikacja próbuje zamknąć połączenie, które jest już zamknięte, wywołanie jest ignorowane.

Parametry:
Brak

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException

CreateSession -Tworzenie sesji

Interfejs:

```
ISession CreateSession(Boolean transacted,
                       AcknowledgeMode acknowledgeMode);
```

Utwórz sesję.

Parametry:

transacted (wejście)
Wartość `True` oznacza, że sesja jest transakowana. Wartość `False` oznacza, że sesja nie jest transakowana.

W przypadku połączenia w czasie rzeczywistym z brokerem wartość musi mieć wartość `False`.

acknowledgeMode (wejście)
Wskazuje, że komunikaty odebrane przez aplikację są potwierdzane. Wartość musi być jedną z następujących wartości: AcknowledgeMode enumerator:

`AcknowledgeMode.AutoAcknowledge`
`AcknowledgeMode.ClientAcknowledge`
`AcknowledgeMode.DupsOkAcknowledge`

W przypadku połączenia w czasie rzeczywistym z brokerem wartość musi mieć wartość `AcknowledgeMode.AutoAcknowledge` lub `AcknowledgeMode.DupsOkAcknowledge` .

Ten parametr jest ignorowany, jeśli sesja jest transakowana. Więcej informacji na temat trybów potwierdzania zawiera sekcja “Potwierdzenie komunikatu” na stronie 26.

Zwraca:
Obiekt sesji

Wyjątki:

- Wyjątek XMSException

Uruchom-Uruchom połączenie

Interfejs:

```
void  Start();
```

Uruchom lub zrestartuj dostarczanie komunikatów przychodzących dla połączenia. Wywołanie jest ignorowane, jeśli połączenie jest już uruchomione.

Parametry:
Brak

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException

Zatrzymaj-Zatrzymaj połączenie

Interfejs:

```
void  Stop();
```

Zatrzymaj dostarczanie komunikatów przychodzących dla połączenia. Wywołanie jest ignorowane, jeśli połączenie jest już zatrzymane.

Parametry:
Brak

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException

Odziedziczone właściwości i metody

Następujące metody zostały odziedziczone po interfejsie IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

IConnectionFactory

Aplikacja korzysta z fabryki połączeń w celu utworzenia połączenia.

Hierarchia dziedziczenia:

```
IBM.XMS.IPropertyContext
   |
   +----IBM.XMS.IConnectionFactory
```

Listę XMS zdefiniowanych właściwości obiektu ConnectionFactory można znaleźć w sekcji “Właściwości obiektu ConnectionFactory” na stronie 182.

Pojęcia pokrewne

Obiekty ConnectionFactories i obiekty Connection
Obiekt ConnectionFactory udostępnia szablon, który jest używany przez aplikację do tworzenia obiektu połączenia. Aplikacja korzysta z obiektu połączenia w celu utworzenia obiektu sesji.

Połączenie z magistralą integracji usług
Aplikacja XMS może łączyć się z magistralą integracji usług WebSphere Application Server za pomocą bezpośredniego połączenia TCP/IP lub za pomocą protokołu HTTP przy użyciu protokołu TCP/IP.

Bezpieczne połączenia z menedżerem kolejek produktu IBM MQ
Aby umożliwić aplikacji XMS .NET nawiąże bezpieczne połączenia z menedżerem kolejek produktu IBM MQ , odpowiednie właściwości muszą zostać zdefiniowane w obiekcie ConnectionFactory .

Bezpieczne połączenia z mechanizmem przesyłania komunikatów produktu WebSphere Application Server service integration bus
Aby umożliwić aplikacji XMS .NET nawiąże bezpieczne połączenia z mechanizmem przesyłania komunikatów produktu WebSphere Application Server service integration bus , odpowiednie właściwości muszą zostać zdefiniowane w obiekcie ConnectionFactory .

Odwzorowanie właściwości dla administrowanych obiektów
Aby umożliwić aplikacjom korzystanie z fabryki połączeń produktu IBM MQ JMS i WebSphere Application Server oraz definicji obiektów docelowych, właściwości pobrane z tych definicji muszą być odwzorowane na odpowiednie właściwości produktu XMS , które można ustawić w fabrykach połączeń i miejscach docelowych produktu XMS .

Zadania pokrewne

Tworzenie obiektów administrowanych

Definicje obiektów ConnectionFactory i Destination, które są wymagane przez aplikacje produktu XMS w celu nawiązania połączenia z serwerem przesyłania komunikatów, muszą zostać utworzone przy użyciu odpowiednich narzędzi administracyjnych.

Odsyłacze pokrewne

Wymagane właściwości dla administrowanych obiektów ConnectionFactory
Gdy aplikacja tworzy fabrykę połączeń, należy zdefiniować pewną liczbę właściwości, aby utworzyć połączenie z serwerem przesyłania komunikatów.

Metody

CreateConnection -Tworzenie fabryki połączeń (przy użyciu domyślnej tożsamości użytkownika)

Interfejs:

```
IConnection CreateConnection();
```

Utwórz fabrykę połączeń z właściwościami domyślnymi.

Jeśli połączenie z produktem WebSphere MQ i XMSC_USERID nie jest ustawione, wówczas menedżer kolejek domyślnie używa identyfikatora użytkownika userID zalogowanego użytkownika. Jeśli wymagane jest dalsze uwierzytelnianie na poziomie połączenia dla poszczególnych użytkowników, można napisać wyjście uwierzytelniania klienta skonfigurowane w produkcie WebSphere MQ.

Parametry:
Brak

Wyjątki:

- Wyjątek XMSException

CreateConnection -Tworzenie połączenia (przy użyciu określonej tożsamości użytkownika)

Interfejs:

```
IConnection CreateConnection(String userId, String password);
```

Utwórz połączenie przy użyciu określonej tożsamości użytkownika.

Jeśli połączenie z produktem WebSphere MQ i XMSC_USERID nie jest ustawione, wówczas menedżer kolejek domyślnie używa identyfikatora użytkownika userID zalogowanego użytkownika. Jeśli wymagane jest dalsze uwierzytelnianie na poziomie połączenia dla poszczególnych użytkowników, można napisać wyjście uwierzytelniania klienta skonfigurowane w produkcie WebSphere MQ.

Połączenie jest tworzone w trybie zatrzymania. Żadne komunikaty nie są dostarczane do momentu wywołania aplikacji **`Connection.start()`** przez aplikację.

Parametry:

userID (wejście)
Obiekt typu String hermetyzujący identyfikator użytkownika, który ma być używany do uwierzytelniania aplikacji. Jeśli zostanie udostępniona wartość NULL, podejmowana jest próba utworzenia połączenia bez uwierzytelniania.

hasło (wejście)
Obiekt typu String obudowujący hasło, które ma być używane do uwierzytelniania aplikacji. Jeśli zostanie udostępniona wartość NULL, podejmowana jest próba utworzenia połączenia bez uwierzytelniania.

Zwraca:
Obiekt połączenia.

Wyjątki:

- Wyjątek XMSException
- XMS_X_SECURITY_EXCEPTION

Odziedziczone właściwości i metody

Następujące metody zostały odziedziczone po interfejsie IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

Dane IConnectionMeta

Obiekt danych ConnectionMetaudostępnia informacje na temat połączenia.

Hierarchia dziedziczenia:

```
IBM.XMS.IPropertyContext
   |
   +----IBM.XMS.IConnectionMetaData
```

Aby uzyskać listę zdefiniowanych właściwości XMS danych ConnectionMeta, patrz “Właściwości danych ConnectionMeta” na stronie 189.

Właściwości produktu .NET

JMSXPropertyNames -Pobieranie Właściwości Komunikatu Zdefiniowanego Przez JMS

Interfejs:

```
System.Collections.IEnumerator JMSXPropertyNames
    {
      get;
    }
```

Zwraca wyliczenie nazw właściwości komunikatu zdefiniowanych przez produkt JMS obsługiwanych przez połączenie.

Zdefiniowane właściwości komunikatu produktu JMS nie są obsługiwane przez połączenie w czasie rzeczywistym z brokerem.

Wyjątki:

- Wyjątek XMSException

Odziedziczone właściwości i metody

Następujące metody zostały odziedziczone po interfejsie IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

Miejsce docelowe

Miejsce docelowe to miejsce, w którym aplikacja wysyła komunikaty lub jest to źródło, z którego aplikacja odbiera komunikaty, lub oba te komunikaty.

Hierarchia dziedziczenia:

```
IBM.XMS.IPropertyContext
   |
   +----IBM.XMS.IDestination
```

Listę XMS zdefiniowanych właściwości obiektu docelowego można znaleźć w sekcji “Właściwości miejsca docelowego” na stronie 189.

Pojęcia pokrewne

Obiekty ConnectionFactories i obiekty Connection
Obiekt ConnectionFactory udostępnia szablon, który jest używany przez aplikację do tworzenia obiektu połączenia. Aplikacja korzysta z obiektu połączenia w celu utworzenia obiektu sesji.

Połączenie z magistralą integracji usług
Aplikacja XMS może łączyć się z magistralą integracji usług WebSphere Application Server za pomocą bezpośredniego połączenia TCP/IP lub za pomocą protokołu HTTP przy użyciu protokołu TCP/IP.

Miejsca docelowe
Aplikacja XMS używa obiektu docelowego do określenia miejsca docelowego wysyłanych komunikatów oraz źródła komunikatów, które są odbierane.

Znaki wieloznaczne miejsca docelowego
Produkt XMS udostępnia obsługę znaków wieloznacznych w celu zapewnienia, że znaki wieloznaczne mogą być przekazywane do miejsca, w którym są one potrzebne do dopasowania. Dla każdego typu serwera, z którym może pracować produkt XMS , istnieje inny schemat znaków wieloznacznych.

Identyfikatory ujednolicona zasobu tematu
Identyfikator URI (Uniform Resource Identifier) tematu określa nazwę tematu. Można również określić dla niego jedną lub więcej właściwości.

Identyfikatory URI jednorodnych zasobów
Identyfikator URI kolejki określa nazwę kolejki. Można również określić jedną lub więcej właściwości kolejki.

Tymczasowe miejsca docelowe
Aplikacje produktu XMS mogą tworzyć tymczasowe miejsca docelowe i korzystać z nich.

Odwzorowanie właściwości dla administrowanych obiektów
Aby umożliwić aplikacjom korzystanie z fabryki połączeń produktu IBM MQ JMS i WebSphere Application Server oraz definicji obiektów docelowych, właściwości pobrane z tych definicji muszą być odwzorowane na odpowiednie właściwości produktu XMS , które można ustawić w fabrykach połączeń i miejscach docelowych produktu XMS .

Zadania pokrewne

Tworzenie obiektów administrowanych
Definicje obiektów ConnectionFactory i Destination, które są wymagane przez aplikacje produktu XMS w celu nawiązania połączenia z serwerem przesyłania komunikatów, muszą zostać utworzone przy użyciu odpowiednich narzędzi administracyjnych.

Odsyłacze pokrewne

Wymagane właściwości administrowanych obiektów docelowych
Aplikacja tworzący miejsce docelowe musi ustawić kilka właściwości, które aplikacja ma w administrowanego obiekcie docelowym.

Właściwości produktu .NET

Nazwa-Pobierz nazwę miejsca docelowego

Interfejs:

```
String Name
{
```

```
  get;
}
```

Pobierz nazwę miejsca docelowego. Nazwa jest łańcuchem hermetyzującym nazwę kolejki lub nazwę tematu.

Wyjątki:

- Wyjątek XMSException

TypeId -pobranie typu miejsca docelowego

Interfejs:

```
DestinationType TypeId
{
  get;
}
```

Pobierz typ miejsca docelowego. Typ miejsca docelowego to jedna z następujących wartości:

```
DestinationType.Queue
DestinationType.Topic
```

Wyjątki:

- Wyjątek XMSException

Odziedziczone właściwości i metody

Następujące metody zostały odziedziczone po interfejsie IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

ExceptionListener

Hierarchia dziedziczenia:

Brak

Aplikacja korzysta z obiektu nasłuchiwania wyjątków, który jest powiadamiany asynchronicznie o problemie z połączeniem.

Jeśli aplikacja korzysta z połączenia tylko w celu asynchronicznego korzystania z komunikatów i w żadnym innym celu, jedynym sposobem, w jaki aplikacja może uzyskać informacje na temat problemu z połączeniem, jest użycie obiektu nasłuchiwania wyjątków. W innych sytuacjach proces nasłuchiwania wyjątków może zapewnić bardziej natychmiastowy sposób uczenia się problemu z połączeniem, niż oczekiwanie na następne wywołanie synchroniczne do produktu XMS.

Delegowanie

ExceptionListener -obiekt nasłuchiwania wyjątków

Interfejs:

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

Powiadom aplikację o problemie z połączeniem.

Metody implementowane przez ten delegat mogą być rejestrowane w połączeniu.

Więcej informacji na temat używania programów nasłuchujących wyjątków zawiera sekcja “Programy nasłuchujące komunikatów i wyjątków w produkcie .NET” na stronie 49.

Parametry:

wyjątek (wejście)
Wskaźnik do wyjątku utworzonego przez produkt XMS.

Zwraca:
Unieważnione

Wyjątek IllegalState

Hierarchia dziedziczenia:

```
IBM.XMS.XMSException
   |
   +----IBM.XMS.Exception
           |
           +----IBM.XMS.IllegalStateException
```

XMS zgłasza ten wyjątek, jeśli aplikacja wywołuje metodę w niepoprawnym lub nieodpowiednim czasie lub jeśli XMS nie znajduje się w odpowiednim stanie dla żądanej operacji.

Odziedziczone właściwości i metody

Następujące metody zostały odziedziczone po interfejsie XMSException:

GetErrorCode, GetLinkedException

InitialContext

Aplikacja korzysta z obiektu InitialContext do tworzenia obiektów na podstawie definicji obiektów pobranych z repozytorium obiektów administrowanych.

Hierarchia dziedziczenia:
Brak

Pojęcia pokrewne
Właściwości obiektu InitialContext
Parametry konstruktora InitialContext obejmują położenie repozytorium administrowanych obiektów, podane jako jednolity wskaźnik zasobów (URI). Aby aplikacja mogła nawiązać połączenie z repozytorium, może być konieczne podanie większej ilości informacji niż informacje zawarte w identyfikatorze URI.

Format identyfikatora URI dla kontekstów początkowych produktu XMS
Położenie repozytorium administrowanych obiektów jest udostępniane jako jednolity wskaźnik zasobów (URI). Format identyfikatora URI zależy od typu kontekstu.

Pobieranie administrowanych obiektów
Program XMS pobiera obiekt administrowany z repozytorium przy użyciu adresu udostępnionego podczas tworzenia obiektu InitialContext lub w właściwościach InitialContext .

Zadania pokrewne
Obiekty InitialContext
Aplikacja musi utworzyć kontekst początkowy, który będzie używany w celu nawiązania połączenia z repozytorium obiektów administrowanych w celu pobrania wymaganych obiektów administrowanych.

Właściwości produktu .NET

Środowisko-pobierz środowisko

Interfejs:

```
Hashtable Environment
{
     get;
}
```

Pobierz środowisko.

Wyjątki:

- Wyjątki są specyficzne dla używanej usługi katalogowej.

Konstruktory

InitialContext -Tworzenie kontekstu początkowego

Interfejs:

```
InitialContext(Hashtable env);
```

Utwórz obiekt InitialContext .

Parametry:

Informacje wymagane do nawiązania połączenia z repozytorium administrowanych obiektów są dostarczane do konstruktora w środowisku Hashtable.

Wyjątki:

- Wyjątek XMSException

Metody

AddToŚrodowisko-Dodaj nową właściwość do środowiska

Interfejs:

```
Object AddToEnvironment(String propName, Object propVal);
```

Dodaj nową właściwość do środowiska.

Parametry:

propName (wejście)
Obiekt typu String hermetyzujący nazwę właściwości, która ma zostać dodana.

propVal (wejście)
Wartość właściwości, która ma zostać dodana.

Zwraca:
Stara wartość właściwości.

Wyjątki:

- Wyjątki są specyficzne dla używanej usługi katalogowej.

Zamknij-Zamknij ten kontekst

Interfejs:

```
void Close()
```

Zamknij ten kontekst.

Parametry:
Brak

Zwraca:
Brak

Wyjątki:

- Wyjątki są specyficzne dla używanej usługi katalogowej.

Wyszukiwanie-wyszukiwanie obiektu w kontekście początkowym

Interfejs:

```
Object Lookup(String name);
```

Utwórz obiekt z definicji obiektu pobranej z repozytorium administrowanych obiektów.

Parametry:

nazwa (wejście)
Obiekt typu String hermetyzujący nazwę administrowanego obiektu do pobrania. Nazwa może być nazwą prostą lub nazwą złożoną. Więcej informacji na ten temat zawiera sekcja "Pobieranie administrowanych obiektów" na stronie 66.

Zwraca:
IConnectionFactory lub IDestination, w zależności od typu odtwarzaczy obiektu. Jeśli funkcja może uzyskać dostęp do katalogu, ale nie może znaleźć wymaganego obiektu, zwracana jest wartość NULL.

Wyjątki:

- Wyjątki są specyficzne dla używanej usługi katalogowej.

RemoveFromenvironment-usuwanie właściwości z środowiska

Interfejs:

```
Object RemoveFromEnvironment(String propName);
```

Usuń właściwość ze środowiska.

Parametry:

propName (wejście)
Obiekt typu String hermetyzujący nazwę właściwości do usunięcia.

Zwraca:
Obiekt, który został usunięty.

Wyjątki:

- Wyjątki są specyficzne dla używanej usługi katalogowej.

InvalidClientIDException

Hierarchia dziedziczenia:

```
IBM.XMS.XMSException
   |
   +----IBM.XMS.XMSException
           |
           +----IBM.XMS.InvalidClientIDException
```

XMS zgłasza ten wyjątek, jeśli aplikacja próbuje ustawić identyfikator klienta dla połączenia, ale identyfikator klienta nie jest poprawny lub jest już używany.

Odziedziczone właściwości i metody

Następujące metody zostały odziedziczone po interfejsie XMSException:

GetErrorCode, GetLinkedException

Wyjątek InvalidDestination

Hierarchia dziedziczenia:

```
IBM.XMS.XMSException
   |
   +----IBM.XMS.XMSException
           |
           +----IBM.XMS.InvalidDestinationException
```

XMS zgłasza ten wyjątek, jeśli w aplikacji określono niepoprawne miejsce docelowe.

Odziedziczone właściwości i metody

Następujące metody zostały odziedziczone po interfejsie XMSException:

GetErrorCode, GetLinkedException

InvalidSelectorWyjątek

Hierarchia dziedziczenia:

```
IBM.XMS.XMSException
   |
   +----IBM.XMS.XMSException
           |
           +----IBM.XMS.InvalidSelectorException
```

XMS zgłasza ten wyjątek, jeśli aplikacja udostępnia wyrażenie selektora komunikatów, którego składnia nie jest poprawna.

Odziedziczone właściwości i metody

Następujące metody zostały odziedziczone po interfejsie XMSException:

GetErrorCode, GetLinkedException

IMapMessage

Komunikat odwzorowania to komunikat, którego treść składa się z zestawu par nazwa-wartość, w których każda wartość ma powiązany typ danych.

Hierarchia dziedziczenia:

```
IBM.XMS.IPropertyContext
   |
   +----IBM.XMS.IMessage
           |
           +----IBM.XMS.IMapMessage
```

Jeśli aplikacja pobiera wartość pary nazwa-wartość, wartość może zostać przekształcona przez program XMS na inny typ danych. Więcej informacji na temat tego rodzaju konwersji niejawnej zawiera sekcja “Komunikaty mapy” na stronie 80.

Odsyłacze pokrewne

Komunikaty mapy
Treść komunikatu mapy zawiera zestaw par nazwa-wartość, w których każda wartość ma powiązany typ danych.

Właściwości produktu .NET

MapNames -pobieranie nazw map

Interfejs:

```
System.Collections.IEnumerator MapNames
{
  get;
}
```

Pobierz wyliczenie nazw w treści komunikatu mapy.

Wyjątki:

- Wyjątek XMSException

Metody

GetBoolean -pobranie wartości boolowskiej

Interfejs:

```
Boolean GetBoolean(String name);
```

Pobiera wartość boolowką identyfikowana przez nazwę z treści komunikatu odwzorowania.

Parametry:

nazwa (wejście)
Obiekt typu String hermetyzujący nazwę identyfikującą wartość boolowką.

Zwraca:
Wartość boolowska pobrana z treści komunikatu mapy.

Wyjątki:

- Wyjątek XMSException

GetByte -Get Byte

Interfejs:

```
Byte    GetByte(String name);
    Int16   GetSignedByte(String name);
```

Pobierz bajt identyfikowany za pomocą nazwy z treści komunikatu mapy.

Parametry:

nazwa (wejście)
Obiekt typu String hermetyzujący nazwę identyfikującą bajt.

Zwraca:
Bajt pobrany z treści komunikatu mapy. Na bajcie nie jest wykonywana konwersja danych.

Wyjątki:

- Wyjątek XMSException

GetBytes -pobieranie bajtów

Interfejs:

```
Byte[]  GetBytes(String name);
```

Pobiera tablicę bajtów identyfikowanych za pomocą nazwy z treści komunikatu odwzorowania.

Parametry:

nazwa (wejście)
Obiekt typu String hermetyzujący nazwę, która identyfikuje tablicę bajtów.

Zwraca:
Liczba bajtów w tablicy.

Wyjątki:

- Wyjątek XMSException

GetChar -Pobieranie znaku

Interfejs:

```
Char    GetChar(String name);
```

Pobierz znak identyfikowany za pomocą nazwy z treści komunikatu mapy.

Parametry:

nazwa (wejście)
Obiekt typu String hermetyzujący nazwę identyfikującą znak.

Zwraca:
Znak pobrany z treści komunikatu mapy.

Wyjątki:

- Wyjątek XMSException

GetDouble -Uzyskanie Numeru Zmiennopozycyjnego Podwójnej Precyzji

Interfejs:

```
Double  GetDouble(String name);
```

Pobiera liczbę zmiennopozycyjną o podwójnej precyzji identyfikowanej za pomocą nazwy z treści komunikatu mapy.

Parametry:

nazwa (wejście)
Obiekt typu String hermetyzujący nazwę identyfikującą liczbę zmiennopozycyjną podwójnej precyzji.

Zwraca:
Liczba zmiennopozycyjna o podwójnej precyzji pobrana z treści komunikatu mapy.

Wyjątki:

- Wyjątek XMSException

GetFloat -pobranie numeru punktu zmiennopozycyjnego

Interfejs:

```
Single  GetFloat(String name);
```

Pobiera liczbę zmiennopozycyjną identyfikowanej przez nazwę z treści komunikatu odwzorowania.

Parametry:

nazwa (wejście)
Obiekt typu String hermetyzujący nazwę identyfikującą liczbę zmiennopozycyjną.

Zwraca:
Liczba zmiennopozycyjna pobrana z treści komunikatu mapy.

Wyjątki:

- Wyjątek XMSException

GetInt -Pobieranie liczby całkowitej

Interfejs:

```
Int32   GetInt(String name);
```

Pobiera liczbę całkowitą identyfikowana przez nazwę z treści komunikatu odwzorowania.

Parametry:

nazwa (wejście)
Obiekt typu String hermetyzujący nazwę, która identyfikuje liczbę całkowitą.

Zwraca:
Liczba całkowita pobrana z treści komunikatu mapy.

Wyjątki:

- Wyjątek XMSException

GetLong -pobieranie długiej liczby całkowitej

Interfejs:

```
Int64   GetLong(String name);
```

Pobieranie długiej liczby całkowitej identyfikowanej przez nazwę z treści komunikatu odwzorowania.

Parametry:

nazwa (wejście)
Obiekt typu String hermetyzujący nazwę, która identyfikuje długą liczbę całkowitą.

Zwraca:
Długa liczba całkowita pobrana z treści komunikatu mapy.

Wyjątki:

- Wyjątek XMSException

GetObject -Pobieranie obiektu

Interfejs:

```
Object  GetObject(String name);
```

Pobierz odwołanie do wartości pary nazwa-wartość z treści komunikatu odwzorowania. Para nazwa-wartość jest identyfikowana przez nazwę.

Parametry:

nazwa (wejście)
Obiekt typu String hermetyzujący nazwę pary nazwa-wartość.

Zwraca:
Wartość, która jest jednym z następujących typów obiektów:

```
Boolean
Byte
Byte[]
Char
Double
Single
Int32
Int64
Int16
String
```

Wyjątki:
Wyjątek XMSException

GetShort -pobieranie krótkiej liczby całkowitej

Interfejs:

```
Int16   GetShort(String name);
```

Pobierz krótką liczbę całkowitą identyfikowana przez nazwę z treści komunikatu mapy.

Parametry:

nazwa (wejście)
Obiekt typu String hermetyzujący nazwę, która identyfikuje krótką liczbę całkowitą.

Zwraca:
Krótka liczba całkowita pobrana z treści komunikatu mapy.

Wyjątki:

- Wyjątek XMSException

GetString -pobieranie łańcucha

Interfejs:

```
String  GetString(String name);
```

Pobierz łańcuch identyfikowany za pomocą nazwy z treści komunikatu odwzorowania.

Parametry:

nazwa (wejście)
Obiekt typu String hermetyzujący nazwę identyfikującą łańcuch w treści komunikatu mapy.

Zwraca:
Obiekt typu String hermetyzujący łańcuch pobrany z treści komunikatu mapy. Jeśli konwersja danych jest wymagana, wartość ta jest łańcuchem po konwersji.

Wyjątki:

- Wyjątek XMSException

ItemExists -Sprawdzanie Nazwy-Istnieje Para Wartości

Interfejs:

```
Boolean ItemExists(String name);
```

Sprawdź, czy treść komunikatu odwzorowania zawiera parę nazwa-wartość o podanej nazwie.

Parametry:

nazwa (wejście)
Obiekt typu String hermetyzujący nazwę pary nazwa-wartość.

Zwraca:

- `True`, jeśli treść komunikatu mapy zawiera parę nazwa-wartość o podanej nazwie.
- `False`, jeśli treść komunikatu mapy nie zawiera pary nazwa-wartość o podanej nazwie.

Wyjątki:

- Wyjątek XMSException

SetBoolean -ustawienie wartości boolowskiej

Interfejs:

```
void  SetBoolean(String name, Boolean value);
```

Umożliwia ustawianie wartości boolowskiej w treści komunikatu odwzorowania.

Parametry:

nazwa (wejście)
Obiekt typu String obudowujący nazwę w celu zidentyfikowania wartości boolowskiej w treści komunikatu mapy.

wartość (wejście)
Wartość boolowska, która ma zostać ustawiona.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException

SetByte -Ustawienie Bajtu

Interfejs:

```
void  SetByte(String name, Byte value);
void  SetSignedByte(String name, Int16 value);
```

Ustaw bajt w treści komunikatu mapy.

Parametry:

nazwa (wejście)
Obiekt typu String obudowujący nazwę w celu zidentyfikowania bajtu w treści komunikatu mapy.

wartość (wejście)
Bajt, który ma zostać ustawiony.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException

SetBytes -ustawianie bajtów

Interfejs:

```
void  SetBytes(String name, Byte[] value);
```

Umożliwia ustawianie tablicy bajtów w treści komunikatu mapy.

Parametry:

nazwa (wejście)
Obiekt typu String obudowujący nazwę w celu zidentyfikowania tablicy bajtów w treści komunikatu mapy.

wartość (wejście)
Tablica bajtów, które mają zostać ustawione.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException

SetChar -Ustaw znak

Interfejs:

```
void  SetChar(String name, Char value);
```

Ustaw dwubajtowy znak w treści komunikatu mapy.

Parametry:

nazwa (wejście)
Obiekt typu String obudowujący nazwę w celu zidentyfikowania znaku w treści komunikatu mapy.

wartość (wejście)
Znak, który ma zostać ustawiony.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException

SetDouble -ustawienie numeru zmiennopozycyjnego podwójnej precyzji

Interfejs:

```
void  SetDouble(String name, Double value);
```

Umożliwia ustawienie liczby zmiennopozycyjnej podwójnej precyzji w treści komunikatu mapy.

Parametry:

nazwa (wejście)
Obiekt typu String obudowujący nazwę w celu zidentyfikowania liczby zmiennopozycyjnej podwójnej precyzji w treści komunikatu mapy.

wartość (wejście)
Liczba zmiennopozycyjna podwójnej precyzji, która ma zostać ustawiona.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException

SetFloat -ustawienie liczby zmiennopozycyjnego

Interfejs:

```
void  SetFloat(String name, Single value);
```

Ustaw liczbę zmiennopozycyjną w treści komunikatu mapy.

Parametry:

nazwa (wejście)
Obiekt typu String obudowujący nazwę w celu zidentyfikowania liczby zmiennopozycyjnej w treści komunikatu mapy.

wartość (wejście)
Liczba zmiennopozycyjna, która ma zostać ustawiona.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException

SetInt -ustawienie liczby całkowitej

Interfejs:

```
void  SetInt(String name, Int32 value);
```

Umożliwia ustawianie liczby całkowitej w treści komunikatu mapy.

Parametry:

nazwa (wejście)
Obiekt typu String obudowujący nazwę w celu zidentyfikowania liczby całkowitej w treści komunikatu mapy.

wartość (wejście)
Liczba całkowita, która ma zostać ustawiona.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException

SetLong -ustawienie długiej liczby całkowitej

Interfejs:

```
void  SetLong(String name, Int64 value);
```

Umożliwia ustawienie długiej liczby całkowitej w treści komunikatu mapy.

Parametry:

nazwa (wejście)
Obiekt typu String obudowujący nazwę w celu zidentyfikowania długiej liczby całkowitej w treści komunikatu mapy.

wartość (wejście)
Długa liczba całkowita, która ma zostać ustawiona.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException

SetObject -ustawienie obiektu

Interfejs:

```
void  SetObject(String name, Object value);
```

Ustaw wartość, która musi być typem podstawowym XMS , w treści komunikatu mapy.

Parametry:

nazwa (wejście)
Obiekt typu String obudowujący nazwę w celu zidentyfikowania wartości w treści komunikatu mapy.

wartość (wejście)
Tablica bajtów zawierająca wartość, która ma zostać ustawiona.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException

SetShort -ustawienie Short Integer

Interfejs:

```
void  SetShort(String name, Int16 value);
```

Umożliwia ustawienie krótkiej liczby całkowitej w treści komunikatu mapy.

Parametry:

nazwa (wejście)
Obiekt typu String obudowujący nazwę w celu zidentyfikowania krótkiej liczby całkowitej w treści komunikatu mapy.

wartość (wejście)
Krótka liczba całkowita, która ma zostać ustawiona.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException

SetString -Ustaw łańcuch

Interfejs:

```
void  SetString(String name, String value);
```

Ustaw łańcuch w treści komunikatu mapy.

Parametry:

nazwa (wejście)
Obiekt typu String obudowujący nazwę w celu zidentyfikowania łańcucha w treści komunikatu mapy.

wartość (wejście)
Obiekt typu String obudowujący łańcuch, który ma zostać ustawiony.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException

Odziedziczone właściwości i metody

Następujące właściwości zostały odziedziczone po interfejsie IMessage:

JMSCorrelationID, JMSDeliveryMode, JMSDestination, JMSExpiration, JMSMessageID, JMSPriority, JMSRedelivered, JMSReplyTo, JMSTimestamp, JMSType, Properties

Następujące metody zostały odziedziczone po interfejsie IMessage:

clearBody, clearProperties, PropertyExists

Następujące metody zostały odziedziczone po interfejsie IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

Komunikat IMessage

Obiekt komunikatu reprezentuje komunikat, który jest wysyłany lub odbierany przez aplikację. IMessage jest nadklasą dla klas komunikatów, takich jak IMapMessage.

Hierarchia dziedziczenia:

```
IBM.XMS.IPropertyContext
   |
   +----IBM.XMS.IMessage
```

Listę pól nagłówka komunikatu JMS w obiekcie komunikatu można znaleźć w sekcji “Pola nagłówka w komunikacie XMS” na stronie 72. Listę zdefiniowanych właściwości JMS obiektu Message można znaleźć w sekcji “JMS-zdefiniowane właściwości komunikatu” na stronie 74. Listę IBM zdefiniowanych właściwości obiektu Message można znaleźć w sekcji “IBM-zdefiniowane właściwości komunikatu” na stronie 75. Aby uzyskać listę właściwości JMS_IBM_MQMD* dla obiektu komunikatu, patrz “Właściwości JMS_IBM_MQMD*” na stronie 194

Komunikaty są usuwane przez funkcję czyszczenia pamięci. Po usunięciu komunikatu zwolni on zasoby, które były używane.

Odsyłacze pokrewne

Komunikaty produktu XMS

W tej sekcji opisano strukturę i treść komunikatów produktu XMS oraz wyjaśniono, w jaki sposób aplikacje przetwarzają komunikaty produktu XMS .

Właściwości produktu .NET

GetJMSCorrelationID-Get i Set JMSCorrelationID

Interfejs:

```
String JMSCorrelationID
{
  get;
  set;
}
```

Pobierz i ustaw identyfikator korelacji komunikatu jako obiekt typu String (Łańcuch).

Wyjątki:

- Wyjątek XMSException

JMSDeliveryMode -pobieranie i ustawianie parametru JMSDeliveryMode

Interfejs:

```
DeliveryMode JMSDeliveryMode
{
  get;
  set;
}
```

Pobierz i ustaw tryb dostarczania wiadomości.

Tryb dostarczania komunikatu jest jedną z następujących wartości:

```
DeliveryMode.Persistent
DeliveryMode.NonPersistent
```

W przypadku nowo utworzonego komunikatu, który nie został wysłany, tryb dostarczania jest następujący: `DeliveryMode.Trwałe`, z wyjątkiem połączenia w czasie rzeczywistym z brokerem, dla

którego tryb dostarczania to `DeliveryMode.NonPersistent`(Nietrwały). W przypadku odebranego komunikatu metoda zwraca tryb dostarczania, który został ustawiony przez wywołanie metody IMessageProducer.send (), gdy komunikat został wysłany, chyba że aplikacja odbierający zmienia tryb dostarczania, ustawiając wartość JMSDeliveryMode.

Wyjątki:

- Wyjątek XMSException

Miejsce docelowe JMSDestination-pobieranie i ustawianie miejsca docelowego JMS

Interfejs:

```
IDestination JMSDestination
{
  get;
  set;
}
```

Pobierz i ustaw miejsce docelowe komunikatu.

Miejsce docelowe jest ustawiane przy użyciu wywołania funkcji IMessageProducer.send (), gdy komunikat jest wysyłany. Wartość JMSDestination jest ignorowana. Można jednak użyć obiektu JMSDestination, aby zmienić miejsce docelowe odebranego komunikatu.

W przypadku nowo utworzonego komunikatu, który nie został wysłany, metoda zwraca obiekt docelowy o wartości NULL, chyba że aplikacja wysyłający ustawia miejsce docelowe przez ustawienie miejsca docelowego JMSDestination. W przypadku komunikatu, który został odebrany, metoda zwraca obiekt docelowy dla miejsca docelowego, który został ustawiony przez wywołanie metody IMessageProducer.send (), gdy komunikat został wysłany, chyba że aplikacja odbierający zmieni miejsce docelowe przez ustawienie miejsca JMSDestination.

Wyjątki:

- Wyjątek XMSException

JMSExpiration-Get i Set JMSExpiration

Interfejs:

```
Int64 JMSExpiration
{
  get;
  set;
}
```

Pobierz i ustaw czas utraty ważności komunikatu.

Czas utraty ważności jest ustawiany przez wywołanie metody IMessageProducer.send (), gdy komunikat jest wysyłany. Jego wartość jest obliczana przez dodanie czasu do życia, określonego przez aplikację wysyłający, do czasu wysłania komunikatu. Czas utraty ważności jest wyrażony w milisekundach od godziny 00:00:00 GMT w dniu 1 stycznia 1970.

W przypadku nowo utworzonego komunikatu, który nie został wysłany, czas utraty ważności wynosi 0, chyba że aplikacja wysyłający ustawi inny czas utraty ważności przez ustawienie wartości JMSExpiration. W przypadku odebranego komunikatu metoda zwraca czas utraty ważności ustawiony przez wywołanie metody IMessageProducer.send (), gdy komunikat został wysłany, chyba że aplikacja odbierający zmienia czas utraty ważności przez ustawienie wartości JMSExpiration.

Jeśli czas życia wynosi 0, wywołanie funkcji IMessageProducer.send () ustawia czas utraty ważności na wartość 0, aby wskazać, że komunikat nie traci ważności.

Program XMS usuwa przedawnione komunikaty i nie dostarcza ich do aplikacji.

Wyjątki:

- Wyjątek XMSException

JMSMessageID -pobieranie i ustawianie wartości JMSMessageID

Interfejs:

```
String JMSMessageID
{
  get;
  set;
}
```

Pobierz i ustaw identyfikator komunikatu jako obiekt typu łańcuchowego hermetyzację identyfikatora komunikatu.

Identyfikator komunikatu jest ustawiany przez wywołanie metody IMessageProducer.send (), gdy komunikat jest wysyłany. W przypadku odebranego komunikatu metoda zwraca identyfikator komunikatu, który został ustawiony przez wywołanie metody IMessageProducer.send (), gdy komunikat został wysłany, chyba że aplikacja odbierający zmienia identyfikator komunikatu przez ustawienie JMSMessageID.

Jeśli komunikat nie ma identyfikatora komunikatu, metoda zwraca wartość NULL.

Wyjątki:

- Wyjątek XMSException

JMSPriority-Pobieranie i ustawianie właściwości JMSPriority.

Interfejs:

```
Int32 JMSPriority
{
  get;
  set;
}
```

Pobierz i ustaw priorytet komunikatu.

Priorytet jest ustawiany przez wywołanie metody IMessageProducer.send (), gdy komunikat jest wysyłany. Wartość jest liczbą całkowitą z zakresu 0, najniższym priorytetem, do 9, najwyższym priorytetem.

Dla nowo utworzonego komunikatu, który nie został wysłany, priorytetem jest 4 , chyba że aplikacja wysyłający ustawi inny priorytet przez ustawienie JMSPriority. W przypadku odebranego komunikatu metoda zwraca priorytet, który został ustawiony przez wywołanie metody IMessageProducer.send (), gdy komunikat został wysłany, chyba że aplikacja odbierający zmieni priorytet przez ustawienie JMSPriority.

Wyjątki:

- Wyjątek XMSException

JMSRedelivered-Get and Set JMSRedelivered

Interfejs:

```
Boolean JMSRedelivered
{
  get;
```

```
  set;
}
```

Uzyskaj informację o tym, czy komunikat jest ponownie dostarczany, i określ, czy komunikat jest ponownie dostarczany. Wskazanie jest ustawiane przez wywołanie metody IMessageConsumer.receive (), gdy komunikat zostanie odebrany.

Ta właściwość ma następujące wartości:

- `True`, jeśli komunikat jest ponownie dostarczany.
- `False`, jeśli komunikat nie jest ponownie dostarczany.

W przypadku połączenia w czasie rzeczywistym z brokerem wartość ta jest zawsze `False`.

Wskazanie ponownego dostarczania zestawu przez JMSRedelivered przed wysłaniem komunikatu jest ignorowane przez wywołanie metody IMessageProducer.send (), gdy komunikat jest wysyłany, a następnie jest ignorowany i zastępowany przez wywołanie metody IMessageConsumer.receive (), gdy komunikat zostanie odebrany. Można jednak użyć funkcji JMSRedelivered w celu zmiany wskazania dla otrzymanego komunikatu.

Wyjątki:

- Wyjątek XMSException

JMSReplyTo -pobieranie i ustawianie parametru JMSReplyTo .

Interfejs:

```
IDestination JMSReplyTo
{
  get;
  set;
}
```

Pobierz i ustaw miejsce docelowe, w którym ma zostać wysłana odpowiedź na komunikat.

Wartość tej właściwości to obiekt docelowy dla miejsca docelowego, w którym ma zostać wysłana odpowiedź na komunikat. Pusty obiekt docelowy oznacza, że nie oczekuje się odpowiedzi.

Wyjątki:

- Wyjątek XMSException

JMSTimestamp-pobieranie i ustawianie właściwości JMSTimestamp

Interfejs:

```
Int64 JMSTimestamp
{
  get;
  set;
}
```

Pobierz i ustaw godzinę wysłania komunikatu.

Znacznik czasu jest ustawiany przez wywołanie metody IMessageProducer.send (), gdy komunikat jest wysyłany i jest wyrażony w milisekundach od godziny 00:00:00 GMT w dniu 1 stycznia 1970.

W przypadku nowo utworzonego komunikatu, który nie został wysłany, znacznik czasu ma wartość 0, chyba że aplikacja wysyłający ustawia inny znacznik czasu, ustawiając wartość JMSTimestamp. W przypadku odebranego komunikatu metoda zwraca znacznik czasu, który został ustawiony przez wywołanie metody IMessageProducer.send (), gdy komunikat został wysłany, chyba że aplikacja odbierający zmienia znacznik czasu przez ustawienie właściwości JMSTimestamp.

Wyjątki:

- Wyjątek XMSException

Uwagi:

1. Jeśli znacznik czasu jest niezdefiniowany, metoda zwraca wartość 0, ale nie zgłasza wyjątku.

JMSType-pobranie i ustawienie JMSType

Interfejs:

```
String JMSType
{
  get;
  set;
}
```

Pobierz i ustaw typ komunikatu.

Wartość atrybutu JMSType jest łańcuchem hermetyzującym typ komunikatu. Jeśli konwersja danych jest wymagana, wartość ta jest typem po konwersji.

Wyjątki:

- Wyjątek XMSException

PropertyNames -pobieranie właściwości

Interfejs:

```
System.Collections.IEnumerator PropertyNames
{
  get;
}
```

Pobierz wyliczenie właściwości nazw komunikatu.

Wyjątki:

- Wyjątek XMSException

Metody

Potwierdzenie-Potwierdzenie

Interfejs:

```
void  Acknowledge();
```

Potwierdzanie tego komunikatu i wszystkich wcześniej niepotwierdzonych komunikatów odebranych przez sesję.

Aplikacja może wywołać tę metodę, jeśli trybem potwierdzania sesji jest AcknowledgeMode.ClientAcknowledge. Wywołania metody są ignorowane, jeśli sesja ma jakikolwiek inny tryb potwierdzania lub jest transdziałowana.

Komunikaty, które zostały odebrane, ale nie zostały potwierdzone, mogą zostać ponownie dostarczone.

Więcej informacji na temat potwierdzania komunikatów zawiera sekcja “Potwierdzenie komunikatu” na stronie 26.

Parametry:
Brak

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- Wyjątek IllegalState

ClearBody -Wyczyść treść

Interfejs:

```
void  ClearBody();
```

Wyczyść treść komunikatu. Pola nagłówka i właściwości komunikatu nie są czyszczone.

Jeśli aplikacja wyczyści treść komunikatu, treść pozostaje w tym samym stanie, co puste treści w nowo utworzonym komunikacie. Stan pustego treści w nowo utworzonym komunikacie zależy od typu treści komunikatu. Więcej informacji na ten temat zawiera sekcja “Treść komunikatu produktu XMS” na stronie 77.

Aplikacja może usunąć treść wiadomości w dowolnym momencie, bez względu na stan, w którym znajduje się treść. Jeśli treść komunikatu jest tylko do odczytu, jedynym sposobem, w jaki aplikacja może zapisywać do treści, jest aplikacja, aby najpierw wyczyścić treść.

Parametry:
Brak

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException

ClearProperties -Wyczyść właściwości

Interfejs:

```
void  ClearProperties();
```

Wyczyść właściwości komunikatu. Pola nagłówka i treść komunikatu nie są czyszczone.

Jeśli aplikacja wyczyści właściwości komunikatu, właściwości stają się czytelne i dostępne do zapisu.

Aplikacja może usunąć właściwości komunikatu w dowolnym momencie, niezależnie od stanu, w którym znajdują się właściwości. Jeśli właściwości komunikatu są dostępne tylko do odczytu, jedynym sposobem, w jaki właściwości mogą stać się dostępne do zapisu, jest to, że aplikacja będzie mogła wyczyścić właściwości w pierwszej kolejności.

Parametry:
Brak

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException

PropertyExists -sprawdzanie właściwości istnieje

Interfejs:

```
Boolean PropertyExists(String propertyName);
```

Sprawdź, czy komunikat ma właściwość o podanej nazwie.

Parametry:

propertyName (wejście)
Obiekt typu String hermetyzujący nazwę właściwości.

Zwraca:

- `True`, jeśli w komunikacie znajduje się właściwość o podanej nazwie.
- `False`, jeśli komunikat nie ma właściwości o podanej nazwie.

Wyjątki:

- Wyjątek XMSException

Odziedziczone właściwości i metody

Następujące metody zostały odziedziczone po interfejsie IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

IMessageConsumer

Aplikacja używa konsumenta komunikatów do odbierania komunikatów wysyłanych do miejsca docelowego.

Hierarchia dziedziczenia:

```
IBM.XMS.IPropertyContext
   |
   +----IBM.XMS.IMessageConsumer
```

Listę XMS zdefiniowanych właściwości obiektu MessageConsumer można znaleźć w sekcji “Właściwości obiektu MessageConsumer” na stronie 198.

Właściwości produktu .NET

MessageListener -pobieranie i ustawianie nasłuchiwania komunikatów

Interfejs:

```
MessageListener MessageListener
{
  get;
  set;
}
```

Pobierz obiekt nasłuchiwania komunikatów zarejestrowany w konsumencie komunikatów, a następnie zarejestruj obiekt nasłuchiwania komunikatów za pomocą konsumenta komunikatów.

Jeśli żaden obiekt nasłuchiwania komunikatów nie jest zarejestrowany w konsumencie komunikatów, obiekt MessageListener ma wartość NULL. Jeśli obiekt nasłuchiwania komunikatów jest już zarejestrowany w konsumencie komunikatów, można anulować rejestrację, podając zamiast niego wartość NULL.

Więcej informacji na temat korzystania z programów nasłuchujących komunikatów zawiera sekcja “Programy nasłuchujące komunikatów i wyjątków w produkcie .NET” na stronie 49.

Wyjątki:

- Wyjątek XMSException

MessageSelector -pobieranie Selektora komunikatów

Interfejs:

```
String MessageSelector
{
  get;
}
```

Pobierz selektor komunikatów dla konsumenta komunikatów. Wartością zwracanej wartości jest obiekt typu String obudowujący wyrażenie selektora komunikatów. Jeśli konwersja danych jest wymagana, ta wartość jest wyrażeniem selektora komunikatów po konwersji. Jeśli konsument komunikatów nie ma selektora komunikatów, wartość parametru MessageSelector jest pustym obiektem typu String.

Wyjątki:

- Wyjątek XMSException

Metody

Zamknij-Zamknij konsument komunikatów

Interfejs:

```
void  Close();
```

Zamknij konsument komunikatów.

Jeśli aplikacja próbuje zamknąć konsumenta komunikatów, który jest już zamknięty, wywołanie jest ignorowane.

Parametry:
Brak

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException

Odbieranie-odbieranie

Interfejs:

```
IMessage Receive();
```

Odbierz następny komunikat dla konsumenta komunikatów. Wywołanie oczekuje na czas nieokreślony dla komunikatu lub dopóki konsument komunikatów nie zostanie zamknięty.

Parametry:
Brak

Zwraca:
Wskaźnik do obiektu komunikatu. Jeśli konsument komunikatów jest zamknięty w czasie oczekiwania na komunikat, metoda zwraca wskaźnik do obiektu komunikatu o wartości NULL.

Wyjątki:

- Wyjątek XMSException

Odbieranie-odbieranie (z odstępem czasu oczekiwania)

Interfejs:

```
IMessage Receive(Int64 delay);
```

Odbierz następny komunikat dla konsumenta komunikatów. Wywołanie oczekuje tylko określonego okresu dla komunikatu lub do momentu zamknięcia konsumenta komunikatu.

Parametry:

opóźnienie (wejście)
Czas (w milisekundach), przez jaki wywołanie oczekuje na komunikat. Jeśli zostanie określony przedział czasu oczekiwania równy 0, wywołanie oczekuje na czas nieokreślony dla komunikatu.

Zwraca:
Wskaźnik do obiektu komunikatu. Jeśli w okresie oczekiwania nie nadejdzie żaden komunikat lub jeśli konsument komunikatu jest zamknięty w czasie oczekiwania na komunikat, metoda zwraca wskaźnik do obiektu komunikatu o wartości NULL, ale nie zgłasza wyjątku.

Wyjątki:

- Wyjątek XMSException

ReceiveNoWait-Odbiór bez oczekiwania

Interfejs:

```
IMessage ReceiveNoWait();
```

Odbierz następny komunikat dla konsumenta komunikatów, jeśli ten komunikat jest dostępny natychmiast.

Parametry:
Brak

Zwraca:
Wskaźnik do obiektu komunikatu. Jeśli żaden komunikat nie jest dostępny natychmiast, metoda zwraca wskaźnik do obiektu komunikatu o wartości NULL.

Wyjątki:

- Wyjątek XMSException

Odziedziczone właściwości i metody

Następujące metody zostały odziedziczone po interfejsie IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

MessageEOFException

Hierarchia dziedziczenia:

```
IBM.XMS.XMSException
   |
   +----IBM.XMS.XMSException
           |
           +----IBM.XMS.MessageEOFException
```

XMS zgłasza ten wyjątek, jeśli program XMS napotka koniec strumienia komunikatów w bajtach, gdy aplikacja odczyta treść komunikatu bajtów.

Odziedziczone właściwości i metody

Następujące metody zostały odziedziczone po interfejsie XMSException:

GetErrorCode, GetLinkedException

Wyjątek MessageFormat

Hierarchia dziedziczenia:

```
IBM.XMS.XMSException
   |
   +----IBM.XMS.XMSException
           |
           +----IBM.XMS.MessageFormatException
```

XMS zgłasza ten wyjątek, jeśli program XMS napotka komunikat o niepoprawnym formacie.

Odziedziczone właściwości i metody

Następujące metody zostały odziedziczone po interfejsie XMSException:

GetErrorCode, GetLinkedException

IMessageListener (delegat)

Hierarchia dziedziczenia:
Brak

Aplikacja korzysta z funkcji nasłuchiwania komunikatów w celu asynchronicznego odbierania komunikatów.

Delegowanie

MessageListener -obiekt nasłuchiwania komunikatów

Interfejs:

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

Dostarcz komunikat asynchronicznie do konsumenta komunikatów.

Metody implementowane przez ten delegat mogą być rejestrowane w połączeniu.

Więcej informacji na temat korzystania z programów nasłuchujących komunikatów zawiera sekcja “Programy nasłuchujące komunikatów i wyjątków w produkcie .NET” na stronie 49.

Parametry:

mesg (wejście)
Obiekt komunikatu.

Zwraca:
Unieważnione

MessageNotReadableException

Hierarchia dziedziczenia:

```
IBM.XMS.XMSException
   |
   +----IBM.XMS.XMSException
           |
           +----IBM.XMS.MessageNotReadableException
```

XMS zgłasza ten wyjątek, jeśli aplikacja próbuje odczytać treść komunikatu, który jest tylko do zapisu.

Odziedziczone właściwości i metody

Następujące metody zostały odziedziczone po interfejsie XMSException:

GetErrorCode, GetLinkedException

MessageNotWritableException

Hierarchia dziedziczenia:

```
IBM.XMS.XMSException
   |
   +----IBM.XMS.XMSException
           |
           +----IBM.XMS.MessageNotWritableException
```

XMS zgłasza ten wyjątek, jeśli aplikacja podejmie próbę zapisu w treści komunikatu, który jest tylko do odczytu.

Odziedziczone właściwości i metody

Następujące metody zostały odziedziczone po interfejsie XMSException:

GetErrorCode, GetLinkedException

IMessageProducer

Aplikacja korzysta z producenta komunikatów w celu wysyłania komunikatów do miejsca docelowego.

Hierarchia dziedziczenia:

```
IBM.XMS.IPropertyContext
   |
   +----IBM.XMS.IMessageProducer
```

Listę właściwości zdefiniowanych przez XMS obiektu MessageProducer można znaleźć w sekcji “Właściwości elementu MessageProducer” na stronie 198.

Właściwości produktu .NET

DeliveryMode -pobieranie i ustawianie domyślnego trybu dostarczania

Interfejs:

```
DeliveryMode DeliveryMode
{
  get;
  set;
}
```

Pobierz i ustaw domyślny tryb dostarczania dla komunikatów wysyłanych przez producenta komunikatów.

Domyślny tryb dostarczania ma jedną z następujących wartości:

`DeliveryMode.Persistent`
`DeliveryMode.NonPersistent`

W przypadku połączenia w czasie rzeczywistym z brokerem wartość musi mieć wartość `DeliveryMode.NonPersistent`.

Wartością domyślną jest `DeliveryMode.Persistent`, z wyjątkiem połączenia w czasie rzeczywistym z brokerem, dla którego wartością domyślną jest `DeliveryMode.NonPersistent`.

Wyjątki:

- Wyjątek XMSException

Miejsce docelowe-pobierz miejsce docelowe

Interfejs:

```
IDestination Destination
{
  get;
}
```

Pobierz miejsce docelowe dla producenta komunikatów.

Parametry:
Brak

Zwraca:
Obiekt docelowy. Jeśli producent komunikatu nie ma miejsca docelowego, metoda zwraca obiekt docelowy o wartości NULL.

Wyjątki:

- Wyjątek XMSException

DisableMsgID-Pobierz i ustaw flagę wyłączenia identyfikatora komunikatu

Interfejs:

```
Boolean DisableMessageID
{
  get;
  set;
}
```

Sprawdź, czy aplikacja odbierający wymaga, aby identyfikatory komunikatów były uwzględniane w komunikatach wysyłanych przez producenta komunikatów, a także wskazują, czy aplikacja odbierający wymaga, aby identyfikatory komunikatów były uwzględniane w komunikatach wysyłanych przez producenta komunikatów.

Ta opcja jest ignorowana w przypadku połączenia z menedżerem kolejek lub w czasie rzeczywistym połączenia z brokerem. W przypadku połączenia z magistralą integracji usług flaga jest honorowana.

Identyfikator DisabledMsgma następujące wartości:

- `True`, jeśli aplikacja odbierający nie wymaga, aby identyfikatory komunikatów były uwzględniane w komunikatach wysyłanych przez producenta komunikatów.
- `False`, jeśli aplikacja odbierający wymaga, aby identyfikatory komunikatów były uwzględniane w komunikatach wysyłanych przez producenta komunikatów.

Wyjątki:

- Wyjątek XMSException

DisableMsgTS-pobieranie i ustawianie flagi wyłączenia znacznika czasu

Interfejs:

```
Boolean DisableMessageTimestamp
{
  get;
  set;
}
```

Sprawdź, czy aplikacja odbierający wymaga, aby znaczniki czasu były uwzględniane w komunikatach wysyłanych przez producenta komunikatów, a także wskazują, czy aplikacja odbierający wymaga, aby znaczniki czasu były uwzględniane w komunikatach wysyłanych przez producenta komunikatów.

W czasie rzeczywistym połączenia z brokerem ta flaga jest ignorowana. W przypadku połączenia z menedżerem kolejek lub w przypadku połączenia z magistralą integracji usług flaga jest honorowana.

DisableMsgTS ma następujące wartości:

- `True`, jeśli aplikacja odbierający nie wymaga, aby znaczniki czasu były dołączane do komunikatów wysyłanych przez producenta komunikatów.
- `False`, jeśli aplikacja odbierający wymaga, aby znaczniki czasu były uwzględniane w komunikatach wysyłanych przez producenta komunikatów.

Zwraca:

Wyjątki:

- Wyjątek XMSException

Priorytet-Pobierz i ustaw priorytet domyślny

Interfejs:

```
Int32 Priority
{
  get;
  set;
}
```

Pobierz i ustaw priorytet domyślny dla komunikatów wysyłanych przez producenta komunikatów.

Wartość domyślnego priorytetu komunikatu jest liczbą całkowitą z zakresu od `0`, najniższym priorytetem, do 9, najwyższym priorytetem.

W czasie rzeczywistym połączenia z brokerem priorytet komunikatu jest ignorowany.

Wyjątki:

- Wyjątek XMSException

TimeToLive-Get i Set Default Time to Live

Interfejs:

```
Int64 TimeToLive
{
  get;
  set;
}
```

Pobierz i ustaw domyślny czas, przez który komunikat istnieje, zanim utraci ważność.

Czas jest mierzony od momentu wysłania komunikatu przez producenta komunikatu i jest to czas domyślny (w milisekundach). Wartość 0 oznacza, że komunikat nigdy nie traci ważności.

W przypadku połączenia w czasie rzeczywistym z brokerem ta wartość jest zawsze równa 0.

Wyjątki:

- Wyjątek XMSException

Metody

Zamknij-Zamknij producent komunikatów

Interfejs:

```
void  Close();
```

Zamknij producenta komunikatów.

Jeśli aplikacja próbuje zamknąć producenta komunikatów, który jest już zamknięty, wywołanie jest ignorowane.

Parametry:
Brak

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException

Wyślij-Wyślij

Interfejs:

```
void Send(IMessage msg) ;
```

Wyślij komunikat do miejsca docelowego, który został określony podczas tworzenia producenta komunikatów. Wyślij komunikat przy użyciu domyślnego trybu dostarczania, priorytetu i czasu producenta komunikatu, aby można było go użyć.

Parametry:

msg (wejście)
Obiekt komunikatu.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- Wyjątek MessageFormat
- Wyjątek InvalidDestination

Wysyłanie-wysyłanie (określanie trybu dostarczania, priorytetu i czasu życia)

Interfejs:

```
void Send(IMessage msg,
          DeliveryMode deliveryMode,
          Int32 priority,
          Int64 timeToLive);
```

Wyślij komunikat do miejsca docelowego, który został określony podczas tworzenia producenta komunikatów. Wyślij komunikat przy użyciu określonego trybu dostarczania, priorytetu i czasu życia.

Parametry:

msg (wejście)
Obiekt komunikatu.

deliveryMode (wejście)
Tryb dostarczania komunikatu, który musi mieć jedną z następujących wartości:

```
DeliveryMode.Persistent
DeliveryMode.NonPersistent
```

W przypadku połączenia w czasie rzeczywistym z brokerem wartość musi mieć wartość `DeliveryMode.NonPersistent`.

priorytet (wejście)
Priorytet komunikatu. Wartość może być liczbą całkowitą z zakresu 0, dla najniższego priorytetu, dla 9, dla najwyższego priorytetu. W czasie rzeczywistym połączenia z brokerem wartość ta jest ignorowana.

timeToLive (wejście)
Czas życia dla komunikatu (w milisekundach). Wartość 0 oznacza, że komunikat nigdy nie traci ważności. W przypadku połączenia w czasie rzeczywistym z brokerem wartość musi wynosić 0.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- Wyjątek MessageFormat
- Wyjątek InvalidDestination
- Wyjątek IllegalState

Send-Send (do określonego miejsca docelowego)

Interfejs:

```
void Send(IDestination dest, IMessage msg) ;
```

Wysyłaj komunikat do określonego miejsca docelowego, jeśli używany jest producent komunikatów, dla którego nie określono miejsca docelowego podczas tworzenia producenta komunikatów. Wyślij komunikat przy użyciu domyślnego trybu dostarczania, priorytetu i czasu producenta komunikatu, aby można było go użyć.

Zwykle określa się miejsce docelowe podczas tworzenia producenta komunikatów, ale jeśli nie, należy określić miejsce docelowe za każdym razem, gdy wysyłany jest komunikat.

Parametry:

docelowe (wejście)
Obiekt docelowy.

msg (wejście)
Obiekt komunikatu.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- Wyjątek MessageFormat
- Wyjątek InvalidDestination

Send-Send (do określonego miejsca docelowego, określanie trybu dostarczania, priorytetu i czasu życia)

Interfejs:

```
void Send(IDestination dest,
          IMessage msg,
          DeliveryMode deliveryMode,
          Int32 priority,
          Int64 timeToLive) ;
```

Wysyłaj komunikat do określonego miejsca docelowego, jeśli używany jest producent komunikatów, dla którego nie określono miejsca docelowego podczas tworzenia producenta komunikatów. Wyślij komunikat przy użyciu określonego trybu dostarczania, priorytetu i czasu życia.

Zwykle określa się miejsce docelowe podczas tworzenia producenta komunikatów, ale jeśli nie, należy określić miejsce docelowe za każdym razem, gdy wysyłany jest komunikat.

Parametry:

docelowe (wejście)
Obiekt docelowy.

msg (wejście)
Obiekt komunikatu.

deliveryMode (wejście)
Tryb dostarczania komunikatu, który musi mieć jedną z następujących wartości:

```
DeliveryMode.Persistent
DeliveryMode.NonPersistent
```

W przypadku połączenia w czasie rzeczywistym z brokerem wartość musi mieć wartość `DeliveryMode.NonPersistent`.

priorytet (wejście)
Priorytet komunikatu. Wartość może być liczbą całkowitą z zakresu `0`, dla najniższego priorytetu, dla 9, dla najwyższego priorytetu. W czasie rzeczywistym połączenia z brokerem wartość ta jest ignorowana.

timeToLive (wejście)
Czas życia dla komunikatu (w milisekundach). Wartość 0 oznacza, że komunikat nigdy nie traci ważności. W przypadku połączenia w czasie rzeczywistym z brokerem wartość musi wynosić 0.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- Wyjątek MessageFormat
- Wyjątek InvalidDestination
- Wyjątek IllegalState

Odziedziczone właściwości i metody

Następujące metody zostały odziedziczone po interfejsie IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

IObjectMessage

Komunikat obiektu to komunikat, którego treść składa się z serializowanego obiektu Java lub .NET.

Hierarchia dziedziczenia:

```
IBM.XMS.IPropertyContext
   |
   +----IBM.XMS.IMessage
           |
           +----IBM.XMS.IObjectMessage
```

Odsyłacze pokrewne
Komunikaty obiektu
Treść komunikatu obiektu zawiera serializowany obiektJava lub .NET .

Właściwości produktu .NET

Obiekt-Pobierz i ustaw obiekt jako bajty

Interfejs:

```
System.Object Object
    {
      get;
      set;
    }

    Byte[] GetObject();
```

Pobierz i ustaw obiekt, który tworzy treść komunikatu obiektu.

Wyjątki:

- Wyjątek XMSException
- MessageNotReadableException
- MessageEOFException

- MessageNotWritableException

Odziedziczone właściwości i metody

Następujące właściwości zostały odziedziczone po interfejsie IMessage:

JMSCorrelationID, JMSDeliveryMode, JMSDestination, JMSExpiration, JMSMessageID, JMSPriority, JMSRedelivered, JMSReplyTo, JMSTimestamp, JMSType, Properties

Następujące metody zostały odziedziczone po interfejsie IMessage:

clearBody, clearProperties, PropertyExists

Następujące metody zostały odziedziczone po interfejsie 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 jest abstrakcyjną nadklasą, która zawiera metody, które zawierają i ustawia właściwości. Metody te są dziedziczone przez inne klasy.

Hierarchia dziedziczenia:
Brak

Metody

Właściwość GetBoolean-Pobranie właściwości boolowskiej

Interfejs:

```
Boolean GetBooleanProperty(String property_name);
```

Pobierz wartość właściwości boolowskiej o określonej nazwie.

Parametry:

nazwa_właściwości (wejście)
Obiekt typu String hermetyzujący nazwę właściwości.

Zwraca:
Wartość właściwości.

Kontekst wątku:
Określona przez podklasę

Wyjątki:

- Wyjątek XMSException

Właściwość GetByte-Get Byte Property

Interfejs:

```
Byte    GetByteProperty(String property_name) ;
Int16   GetSignedByteProperty(String property_name) ;
```

Pobierz wartość właściwości bajtowej identyfikowanej przez nazwę.

Parametry:

nazwa_właściwości (wejście)
Obiekt typu String hermetyzujący nazwę właściwości.

Zwraca:
Wartość właściwości.

Kontekst wątku:
Określona przez podklasę

Wyjątki:

- Wyjątek XMSException

Właściwość GetBytes-Get Byte Array Property

Interfejs:

```
Byte[]  GetBytesProperty(String property_name) ;
```

Pobierz wartość właściwości tablicy bajtów identyfikowanej przez nazwę.

Parametry:

nazwa_właściwości (wejście)
Obiekt typu String hermetyzujący nazwę właściwości.

Zwraca:
Liczba bajtów w tablicy.

Kontekst wątku:
Określona przez podklasę

Wyjątki:

- Wyjątek XMSException

Właściwość GetChar-Pobieranie właściwości znaku

Interfejs:

```
Char    GetCharProperty(String property_name) ;
```

Pobierz wartość 2-bajtowej właściwości znakowej identyfikowanej przez nazwę.

Parametry:

nazwa_właściwości (wejście)
Obiekt typu String hermetyzujący nazwę właściwości.

Zwraca:
Wartość właściwości.

Kontekst wątku:
Określona przez podklasę

Wyjątki:

- Wyjątek XMSException

Właściwość GetDouble-Pobieranie Właściwości Zmiennopozycyjnego Podwójnej Precyzji

Interfejs:

```
Double  GetDoubleProperty(String property_name) ;
```

Pobierz wartość właściwości zmiennopozycyjnej podwójnej precyzji identyfikowanej za pomocą nazwy.

Parametry:

nazwa_właściwości (wejście)
Obiekt typu String hermetyzujący nazwę właściwości.

Zwraca:
Wartość właściwości.

Kontekst wątku:
Określona przez podklasę

Wyjątki:

- Wyjątek XMSException

Właściwość GetFloat-pobieranie właściwości punktu zmiennopozycyjnego

Interfejs:

```
Single  GetFloatProperty(String property_name) ;
```

Pobierz wartość właściwości zmiennopozycyjnej identyfikowanej przez nazwę.

Parametry:

nazwa_właściwości (wejście)
Obiekt typu String hermetyzujący nazwę właściwości.

Zwraca:
Wartość właściwości.

Kontekst wątku:
Określona przez podklasę

Wyjątki:

- Wyjątek XMSException

Właściwość GetInt- GetInt

Interfejs:

```
Int32   GetIntProperty(String property_name) ;
```

Pobierz wartość właściwości liczby całkowitej identyfikowanej przez nazwę.

Parametry:

nazwa_właściwości (wejście)
Obiekt typu String hermetyzujący nazwę właściwości.

Zwraca:
Wartość właściwości.

Kontekst wątku:
Określona przez podklasę

Wyjątki:

- Wyjątek XMSException

Właściwość GetLong-pobieranie właściwości Long Integer

Interfejs:

```
Int64   GetLongProperty(String property_name) ;
```

Pobierz wartość właściwości długiej liczby całkowitej identyfikowanej przez nazwę.

Parametry:

nazwa_właściwości (wejście)
Obiekt typu String hermetyzujący nazwę właściwości.

Zwraca:
Wartość właściwości.

Kontekst wątku:
Określona przez podklasę

Wyjątki:

- Wyjątek XMSException

Właściwość GetObject-Pobierz właściwość obiektu

Interfejs:

```
Object  GetObjectProperty( String property_name) ;
```

Pobierz wartość i typ danych właściwości identyfikowanej przez nazwę.

Parametry:

nazwa_właściwości (wejście)
Obiekt typu String hermetyzujący nazwę właściwości.

Zwraca:
Wartość właściwości, która jest jednym z następujących typów obiektów:

```
Boolean
Byte
Byte[]
Char
Double
Single
Int32
Int64
Int16
String
```

Kontekst wątku:
Określona przez podklasę

Wyjątki:

- Wyjątek XMSException

Właściwość GetShort-Pobierz właściwość Short Integer Integer

Interfejs:

```
Int16   GetShortProperty(String property_name) ;
```

Pobierz wartość właściwości krótkiej liczby całkowitej identyfikowanej przez nazwę.

Parametry:

nazwa_właściwości (wejście)
Obiekt typu String hermetyzujący nazwę właściwości.

Zwraca:
Wartość właściwości.

Kontekst wątku:
Określona przez podklasę

Wyjątki:

- Wyjątek XMSException

Właściwość GetString-właściwość GetString

Interfejs:

```
String  GetStringProperty(String property_name) ;
```

Pobierz wartość właściwości łańcuchowej identyfikowanej przez nazwę.

Parametry:

nazwa_właściwości (wejście)
Obiekt typu String hermetyzujący nazwę właściwości.

Zwraca:
Obiekt typu String hermetyzujący łańcuch, który jest wartością właściwości. Jeśli konwersja danych jest wymagana, wartość ta jest łańcuchem po konwersji.

Kontekst wątku:
Określona przez podklasę

Wyjątki:

- Wyjątek XMSException

SetBoolean-Ustawienie właściwości boolowskiej

Interfejs:

```
void SetBooleanProperty( String property_name,  Boolean value) ;
```

Umożliwia ustawianie wartości właściwości boolowskiej identyfikowanej przez nazwę.

Parametry:

nazwa_właściwości (wejście)
Obiekt typu String hermetyzujący nazwę właściwości.

wartość (wejście)
Wartość właściwości.

Zwraca:
Unieważnione

Kontekst wątku:
Określona przez podklasę

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

Właściwość SetByte-Ustawienie właściwości Byte

Interfejs:

```
void SetByteProperty( String property_name,  Byte value) ;
      void SetSignedByteProperty( String property_name,  Int16 value) ;
```

Umożliwia ustawianie wartości właściwości bajtowej identyfikowanej przez nazwę.

Parametry:

nazwa_właściwości (wejście)
Obiekt typu String hermetyzujący nazwę właściwości.

wartość (wejście)
Wartość właściwości.

Zwraca:
Unieważnione

Kontekst wątku:
Określona przez podklasę

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

Właściwość SetBytes-Set Byte Array Property

Interfejs:

```
void SetBytesProperty( String property_name, Byte[] value ) ;
```

Umożliwia ustawianie wartości właściwości tablicy bajtów identyfikowanej przez nazwę.

Parametry:

nazwa_właściwości (wejście)
Obiekt typu String hermetyzujący nazwę właściwości.

wartość (wejście)
Wartość właściwości, która jest tablicą bajtów.

Zwraca:
Unieważnione

Kontekst wątku:
Określona przez podklasę

Wyjątki:

- Wyjątek XMSException

- MessageNotWritableException

Właściwość SetChar-ustawienie właściwości znaku

Interfejs:

```
void SetCharProperty( String property_name,  Char value) ;
```

Ustaw wartość 2-bajtowej właściwości znakowej identyfikowanej przez nazwę.

Parametry:

nazwa_właściwości (wejście)
Obiekt typu String hermetyzujący nazwę właściwości.

wartość (wejście)
Wartość właściwości.

Zwraca:
Unieważnione

Kontekst wątku:
Określona przez podklasę

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

Właściwość SetDouble-ustawienie właściwości zmiennopozycyjnego podwójnej precyzji

Interfejs:

```
void SetDoubleProperty( String property_name,  Double value) ;
```

Umożliwia ustawianie wartości właściwości zmiennopozycyjnej podwójnej precyzji identyfikowanej przez nazwę.

Parametry:

nazwa_właściwości (wejście)
Obiekt typu String hermetyzujący nazwę właściwości.

wartość (wejście)
Wartość właściwości.

Zwraca:
Unieważnione

Kontekst wątku:
Określona przez podklasę

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

Właściwość SetFloat-ustawianie właściwości punktu zmiennopozycyjnego

Interfejs:

```
void SetFloatProperty( String property_name,  Single value) ;
```

Umożliwia ustawianie wartości właściwości zmiennopozycyjnej identyfikowanej przez nazwę.

Parametry:

nazwa_właściwości (wejście)
Obiekt typu String hermetyzujący nazwę właściwości.

wartość (wejście)
Wartość właściwości.

Zwraca:
Unieważnione

Kontekst wątku:
Określona przez podklasę

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

Właściwość SetInt-Ustaw właściwość typu Integer

Interfejs:

```
void SetIntProperty( String property_name,  Int32 value) ;
```

Umożliwia ustawianie wartości właściwości całkowitej identyfikowanej przez nazwę.

Parametry:

nazwa_właściwości (wejście)
Obiekt typu String hermetyzujący nazwę właściwości.

wartość (wejście)
Wartość właściwości.

Zwraca:
Unieważnione

Kontekst wątku:
Określona przez podklasę

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

Właściwość SetLong-ustawienie właściwości Long Integer

Interfejs:

```
void SetLongProperty( String property_name,  Int64 value) ;
```

Umożliwia ustawianie wartości właściwości długiej liczby całkowitej identyfikowanej przez nazwę.

Parametry:

nazwa_właściwości (wejście)
Obiekt typu String hermetyzujący nazwę właściwości.

wartość (wejście)
Wartość właściwości.

Zwraca:
Unieważnione

Kontekst wątku:
Określona przez podklasę

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

Właściwość obiektu SetObject-ustawienie właściwości obiektu

Interfejs:

```
void SetObjectProperty( String property_name,  Object value) ;
```

Ustaw wartość i typ danych właściwości identyfikowanej przez nazwę.

Parametry:

nazwa_właściwości (wejście)
Obiekt typu String hermetyzujący nazwę właściwości.

objectType (wejście)
Wartość właściwości, która musi być jednym z następujących typów obiektów:

```
Boolean
Byte
Byte[]
Char
Double
Single
Int32
Int64
Int16
String
```

wartość (wejście)
Wartość właściwości w postaci tablicy bajtów.

długość (wejście)
Liczba bajtów w tablicy.

Zwraca:
Unieważnione

Kontekst wątku:
Określona przez podklasę

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

SetShortProperty-ustawienie właściwości Short Integer Integer

Interfejs:

```
void SetShortProperty( String property_name,  Int16 value) ;
```

Umożliwia ustawianie wartości właściwości krótkiej liczby całkowitej identyfikowanej przez nazwę.

Parametry:

nazwa_właściwości (wejście)
Obiekt typu String hermetyzujący nazwę właściwości.

wartość (wejście)
Wartość właściwości.

Zwraca:
Unieważnione

Kontekst wątku:
Określona przez podklasę

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

SetStringWłaściwość-Ustaw właściwość String

Interfejs:

```
void SetStringProperty( String property_name,  String value);
```

Umożliwia ustawianie wartości właściwości łańcuchowej identyfikowanej przez nazwę.

Parametry:

nazwa_właściwości (wejście)
Obiekt typu String hermetyzujący nazwę właściwości.

wartość (wejście)
Obiekt typu String hermetyzujący łańcuch, który jest wartością właściwości.

Zwraca:
Unieważnione

Kontekst wątku:
Określona przez podklasę

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

IQueueBrowser

Aplikacja korzysta z przeglądarki kolejek w celu przeglądania komunikatów w kolejce bez usuwania ich.

Hierarchia dziedziczenia:

```
IBM.XMS.IPropertyContext
System.Collections.IEnumerable
   |
   +----IBM.XMS.IQueueBrowser
```

Właściwości produktu .NET

MessageSelector -pobieranie Selektora komunikatów

Interfejs:

```
String MessageSelector
{
  get;
}
```

Pobierz selektor komunikatów dla przeglądarki kolejek.

Selektor komunikatów to obiekt typu String obudowujący wyrażenie selektora komunikatów. Jeśli konwersja danych jest wymagana, ta wartość jest wyrażeniem selektora komunikatów po konwersji. Jeśli przeglądarka kolejek nie ma selektora komunikatów, metoda zwraca obiekt typu Łańcuch o wartości NULL.

Wyjątki:

- Wyjątek XMSException

Kolejka-Pobierz kolejkę

Interfejs:

```
IDestination Queue
{
  get;
}
```

Pobranie kolejki powiązanej z przeglądarką kolejki jako obiektu docelowego reprezentującego kolejkę.

Wyjątki:

- Wyjątek XMSException

Metody

Zamknij-Zamknij przeglądarkę kolejek

Interfejs:

```
void  Close();
```

Zamknij przeglądarkę kolejki.

Jeśli aplikacja próbuje zamknąć przeglądarkę kolejki, która jest już zamknięta, wywołanie jest ignorowane.

Parametry:
Brak

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException

GetEnumerator -pobieranie komunikatów

Interfejs:

```
IEnumerator GetEnumerator();
```

Pobierz listę komunikatów znajdujących się w kolejce.

Metoda zwraca obiekt wyliczeniowy, który hermetykuje listę obiektów komunikatu. Kolejność obiektów komunikatu jest taka sama, jak kolejność, w jakiej komunikaty będą pobierane z kolejki. Aplikacja może następnie użyć wyliczenia, aby przejrzeć każdy komunikat z kolei.

Program wyliczeniowy jest aktualizowany dynamicznie, ponieważ komunikaty są umieszczane w kolejce i usuwane z kolejki. Za każdym razem, gdy aplikacja wywołuje komendę IEnumerator.MoveNext() w celu przeglądania następnego komunikatu w kolejce, komunikat ten odzwierciedla bieżącą zawartość kolejki.

Jeśli aplikacja wywoła tę metodę więcej niż raz dla przeglądarki kolejek, każde wywołanie zwraca nowy obiekt wyliczeniowy. W związku z tym aplikacja może używać więcej niż jednego wyliczenia do przeglądania komunikatów w kolejce i obsługi wielu pozycji w kolejce.

Parametry:
Brak

Zwraca:
Obiekt iteratora.

Wyjątki:

- Wyjątek XMSException

Odziedziczone właściwości i metody

Następujące metody zostały odziedziczone po interfejsie IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

Żądający

Aplikacja korzysta z requestera w celu wysłania komunikatu żądania, a następnie czekania na odpowiedź i odebrania odpowiedzi.

Hierarchia dziedziczenia:
Brak

Konstruktory

Żądający-Utwórz Zamawiającego

Interfejs:

```
Requestor(ISession sess, IDestination dest);
```

Tworzenie requestera.

Parametry:

Sess (wejście)
Obiekt sesji. Sesja nie może być transakowana i musi mieć jeden z następujących trybów potwierdzania:

`AcknowledgeMode.AutoAcknowledge`
`AcknowledgeMode.DupsOkAcknowledge`

docelowe (wejście)
Obiekt docelowy reprezentujący miejsce docelowe, w którym aplikacja może wysyłać komunikaty żądania.

Kontekst wątku:
Sesja powiązana z requesterem

Wyjątki:

- Wyjątek XMSException

Metody

Zamknij-Zamawiający Zamawiający

Interfejs:

```
void  Close();
```

Zamknij zamawiającego.

Jeśli aplikacja próbuje zamknąć zgłaszającego żądanie, które jest już zamknięte, wywołanie jest ignorowane.

Uwaga: Gdy aplikacja zamknie zamawiającego, powiązana sesja również nie jest zamykana. W tym zakresie produkt XMS zachowuje się inaczej w porównaniu z usługą JMS.

Parametry:
Brak

Zwraca:
Unieważnione

Kontekst wątku:
Dowolna

Wyjątki:

- Wyjątek XMSException

Żądanie-odpowiedź na żądanie

Interfejs:

```
IMessage Request(IMessage requestMessage);
```

Wyślij komunikat żądania, a następnie odczekaj, a następnie odeślij odpowiedź z aplikacji, która odbiera komunikat z żądaniem.

Wywołanie tej metody blokuje do momentu odebrania odpowiedzi lub do momentu zakończenia sesji, w zależności od tego, która z tych dat jest wcześniejsza.

Parametry:

requestMessage (wejście)
Obiekt komunikatu hermetyzujący komunikat żądania.

Zwraca:
Wskaźnik do obiektu komunikatu hermetyzującego komunikat odpowiedzi.

Kontekst wątku:
Sesja powiązana z requesterem

Wyjątki:

- Wyjątek XMSException

Wyjątek ResourceAllocation

Hierarchia dziedziczenia:

```
IBM.XMS.XMSException
   |
   +----IBM.XMS.XMSException
           |
           +----IBM.XMS.ResourceAllocationException
```

XMS zgłasza ten wyjątek, jeśli program XMS nie może przydzielić zasobów wymaganych przez metodę.

Odziedziczone właściwości i metody

Następujące metody zostały odziedziczone po interfejsie XMSException:

GetErrorCode, GetLinkedException

SecurityException

Hierarchia dziedziczenia:

```
IBM.XMS.XMSException
   |
   +----IBM.XMS.XMSException
           |
           +----IBM.XMS.SecurityException
```

Program XMS zgłasza ten wyjątek, jeśli identyfikator użytkownika i hasło udostępnione w celu uwierzytelnienia aplikacji są odrzucane. XMS zgłasza również ten wyjątek, jeśli sprawdzenie uprawnień nie powiedzie się i uniemożliwia wykonanie metody.

Odziedziczone właściwości i metody

Następujące metody zostały odziedziczone po interfejsie XMSException:

GetErrorCode, GetLinkedException

ISesja

Sesja jest jednowątkowym kontekstem do wysyłania i odbierania komunikatów.

Hierarchia dziedziczenia:

```
IBM.XMS.IPropertyContext
   |
   +----IBM.XMS.ISession
```

Listę właściwości zdefiniowanych przez XMS obiektu Session można znaleźć w sekcji “Właściwości sesji” na stronie 198.

Właściwości środowiska .NET

AcknowledgeMode -Pobieranie Trybu Potwierdzenia

Interfejs:

```
AcknowledgeMode AcknowledgeMode
    {
```

```
      get;
    }
```

Pobierz tryb potwierdzania dla sesji.

Tryb potwierdzania jest określany podczas tworzenia sesji.

Jeśli sesja nie jest transakowana, tryb potwierdzania jest jedną z następujących wartości:

`AcknowledgeMode.AutoAcknowledge`
`AcknowledgeMode.ClientAcknowledge`
`AcknowledgeMode.DupsOkAcknowledge`

Więcej informacji na temat trybów potwierdzania zawiera sekcja “Potwierdzenie komunikatu” na stronie 26.

Sesja, która jest transakcana, nie ma trybu potwierdzania. Jeśli sesja jest transmiana, metoda zwraca zamiast niej wartość `AcknowledgeMode.SessionTransacted` .

Wyjątki:

- Wyjątek XMSException

Transakcyjne-Określenie, czy Transakcję

Interfejs:

```
Boolean Transacted
    {
      get;
    }
```

Określ, czy sesja jest transakowana.

Stwierdzona transakcja to:

- `True`, jeśli sesja jest transmiana.
- `False`, jeśli sesja nie jest transakowana.

W przypadku połączenia w czasie rzeczywistym z brokerem metoda zawsze zwraca wartość `False`.

Wyjątki:

- Wyjątek XMSException

Metody

Zamknij-Zamknij sesję

Interfejs:

```
void Close();
```

Zamknij sesję. Jeśli sesja jest transakcyjna, każda transakcja w toku jest wycofana.

Jeśli aplikacja próbuje zamknąć sesję, która jest już zamknięta, wywołanie jest ignorowane.

Parametry:
Brak

Zwraca:
Unieważnione

Kontekst wątku:
Dowolna

Wyjątki:

- Wyjątek XMSException

Zatwierdź-Zatwierdź

Interfejs:

```
void Commit();
```

Zatwierdź wszystkie komunikaty przetworzone w bieżącej transakcji.

Sesja musi być sesją transakcyjne.

Parametry:
Brak

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- Wyjątek IllegalState
- TransactionRolledBackException

CreateBrowser -Tworzenie przeglądarki kolejek

Interfejs:

```
IQueueBrowser CreateBrowser(IDestination queue) ;
```

Utwórz przeglądarkę kolejki dla podanej kolejki.

Parametry:

kolejka (wejście)
Obiekt docelowy reprezentujący kolejkę.

Zwraca:
Obiekt QueueBrowser .

Wyjątki:

- Wyjątek XMSException
- Wyjątek InvalidDestination

CreateBrowser -Tworzenie przeglądarki kolejek (z selektorem komunikatów)

Interfejs:

```
IQueueBrowser CreateBrowser(IDestination queue, String selector) ;
```

Utwórz przeglądarkę kolejki dla określonej kolejki przy użyciu selektora komunikatów.

Parametry:

kolejka (wejście)
Obiekt docelowy reprezentujący kolejkę.

selektor (input)
Obiekt typu String obudowujący wyrażenie selektora komunikatów. Do przeglądarki kolejek dostarczane są tylko te komunikaty o właściwościach zgodnych z wyrażeniem selektora komunikatów.

Pusty obiekt typu String oznacza, że nie istnieje selektor komunikatów dla przeglądarki kolejek.

Zwraca:
Obiekt QueueBrowser .

Wyjątki:

- Wyjątek XMSException
- Wyjątek InvalidDestination
- InvalidSelectorWyjątek

Komunikat CreateBytesMessage-Create Bytes

Interfejs:

```
IBytesMessage CreateBytesMessage();
```

Utwórz komunikat bajtów.

Parametry:
Brak

Zwraca:
Obiekt BytesMessage .

Wyjątki:

- Wyjątek XMSException
- IllegalStateWyjątek (sesja jest zamknięta)

CreateConsumer -Tworzenie konsumenta

Interfejs:

```
IMessageConsumer CreateConsumer(IDestination dest) ;
```

Utwórz konsument komunikatów dla określonego miejsca docelowego.

Parametry:

docelowe (wejście)
Obiekt docelowy.

Zwraca:
Obiekt MessageConsumer .

Wyjątki:

- Wyjątek XMSException
- Wyjątek InvalidDestination

CreateConsumer -tworzenie konsumenta (z selektorem komunikatów)

Interfejs:

```
IMessageConsumer CreateConsumer(IDestination dest,
                                String selector) ;
```

Utwórz konsument komunikatów dla określonego miejsca docelowego przy użyciu selektora komunikatów.

Parametry:

docelowe (wejście)
Obiekt docelowy.

selektor (input)
Obiekt typu String obudowujący wyrażenie selektora komunikatów. Do konsumenta komunikatów dostarczane są tylko te komunikaty o właściwościach zgodnych z wyrażeniem selektora komunikatów.

Pusty obiekt typu String oznacza, że nie istnieje selektor komunikatów dla konsumenta komunikatów.

Zwraca:
Obiekt MessageConsumer .

Wyjątki:

- Wyjątek XMSException
- Wyjątek InvalidDestination
- InvalidSelectorWyjątek

CreateConsumer -tworzenie konsumenta (z selektorem komunikatów i flagą komunikatu lokalnego)

Interfejs:

```
IMessageConsumer CreateConsumer(IDestination dest,
                                String selector,
                                Boolean noLocal) ;
```

Utwórz konsument komunikatów dla określonego miejsca docelowego przy użyciu selektora komunikatów, a jeśli miejscem docelowym jest temat, określ, czy konsument komunikatów odbiera komunikaty publikowane przez jego własne połączenie.

Parametry:

docelowe (wejście)
Obiekt docelowy.

selektor (input)
Obiekt typu String obudowujący wyrażenie selektora komunikatów. Do konsumenta komunikatów dostarczane są tylko te komunikaty o właściwościach zgodnych z wyrażeniem selektora komunikatów.

Pusty obiekt typu String oznacza, że nie istnieje selektor komunikatów dla konsumenta komunikatów.

noLocal (wejście)
Wartość `True` oznacza, że konsument komunikatów nie otrzymuje komunikatów publikowanych przez własne połączenie. Wartość `False` oznacza, że konsument komunikatów odbierze komunikaty opublikowane przez własne połączenie. Wartością domyślną jest `False`.

Zwraca:
Obiekt MessageConsumer .

Wyjątki:

- Wyjątek XMSException
- Wyjątek InvalidDestination
- InvalidSelectorWyjątek

CreateDurableSubskrybent-Tworzenie Trwałego Subskrybenta

Interfejs:

```
IMessageConsumer CreateDurableSubscriber(IDestination dest,
                                         String subscription) ;
```

Utwórz trwały subskrybent dla określonego tematu.

Ta metoda nie jest poprawna dla połączenia w czasie rzeczywistym z brokerem.

Więcej informacji na temat trwałych subskrybentów zawiera sekcja “Trwali subskrybenci” na stronie 34.

Parametry:

docelowe (wejście)
Obiekt docelowy reprezentujący temat. Temat nie może być tematem tymczasowym.

subskrypcja (wejście)
Obiekt typu String hermetyzujący nazwę, która identyfikuje trwałą subskrypcję. Nazwa musi być unikalna w obrębie identyfikatora klienta dla połączenia.

Zwraca:
Obiekt MessageConsumer reprezentujący trwały subskrybent.

Wyjątki:

- Wyjątek XMSException
- Wyjątek InvalidDestination

CreateDurableSubskrybent-Create Trwałego Subskrybenta (z selektorem komunikatów i flagą komunikatu lokalnego)

Interfejs:

```
IMessageConsumer CreateDurableSubscriber(IDestination dest,
                                         String subscription,
                                         String selector,
                                         Boolean noLocal) ;
```

Utwórz trwały subskrybent dla określonego tematu przy użyciu selektora komunikatów i określając, czy trwały subskrybent odbiera komunikaty publikowane przez własne połączenie.

Ta metoda nie jest poprawna dla połączenia w czasie rzeczywistym z brokerem.

Więcej informacji na temat trwałych subskrybentów zawiera sekcja “Trwali subskrybenci” na stronie 34.

Parametry:

docelowe (wejście)
Obiekt docelowy reprezentujący temat. Temat nie może być tematem tymczasowym.

subskrypcja (wejście)
Obiekt typu String hermetyzujący nazwę, która identyfikuje trwałą subskrypcję. Nazwa musi być unikalna w obrębie identyfikatora klienta dla połączenia.

selektor (input)
Obiekt typu String obudowujący wyrażenie selektora komunikatów. Tylko te komunikaty z właściwościami zgodnymi z wyrażeniem selektora komunikatów są dostarczane do trwałego subskrybenta.

Pusty obiekt typu String oznacza, że nie istnieje selektor komunikatów dla trwałego subskrybenta.

noLocal (wejście)
Wartość `True` oznacza, że trwały subskrybent nie odbiera komunikatów publikowanych przez własne połączenie. Wartość `False` oznacza, że trwały subskrybent odbiera komunikaty publikowane przez własne połączenie. Wartością domyślną jest `False`.

Zwraca:
Obiekt MessageConsumer reprezentujący trwały subskrybent.

Wyjątki:

- Wyjątek XMSException
- Wyjątek InvalidDestination
- InvalidSelectorWyjątek

Komunikat CreateMap-Tworzenie komunikatu odwzorowania

Interfejs:

```
IMapMessage CreateMapMessage();
```

Utwórz komunikat odwzorowania.

Parametry:
Brak

Zwraca:
Obiekt MapMessage .

Wyjątki:

- Wyjątek XMSException
- IllegalStateWyjątek (sesja jest zamknięta)

CreateMessage -Tworzenie komunikatu

Interfejs:

```
IMessage CreateMessage();
```

Utwórz komunikat, który nie ma treści.

Parametry:
Brak

Zwraca:
Obiekt komunikatu.

Wyjątki:

- Wyjątek XMSException
- IllegalStateWyjątek (sesja jest zamknięta)

Komunikat CreateObject-Tworzenie komunikatu obiektu

Interfejs:

```
IObjectMessage CreateObjectMessage();
```

Utwórz komunikat obiektu.

Parametry:
Brak

Zwraca:
Obiekt ObjectMessage .

Wyjątki:

- Wyjątek XMSException
- IllegalStateWyjątek (sesja jest zamknięta)

CreateProducer -Tworzenie producenta

Interfejs:

```
IMessageProducer CreateProducer(IDestination dest) ;
```

Utwórz producenta komunikatów w celu wysłania komunikatów do określonego miejsca docelowego.

Parametry:

docelowe (wejście)
Obiekt docelowy.

Jeśli zostanie określony pusty obiekt docelowy, producent komunikatów zostanie utworzony bez miejsca docelowego. W takim przypadku aplikacja musi określać miejsce docelowe za każdym razem, gdy jest używany przez producenta komunikatów w celu wysłania komunikatu.

Zwraca:
Obiekt MessageProducer .

Wyjątki:

- Wyjątek XMSException
- Wyjątek InvalidDestination

CreateQueue -Tworzenie kolejki

Interfejs:

```
IDestination CreateQueue(String queue) ;
```

Utwórz obiekt docelowy w celu reprezentowania kolejki na serwerze przesyłania komunikatów.

Ta metoda nie tworzy kolejki na serwerze przesyłania komunikatów. Aby aplikacja mogła wywołać tę metodę, należy utworzyć kolejkę.

Parametry:

kolejka (wejście)
Obiekt typu String obudowujący nazwę kolejki lub obudowujący jednolity identyfikator zasobu (URI), który identyfikuje kolejkę.

Zwraca:
Obiekt docelowy reprezentujący kolejkę.

Wyjątki:

- Wyjątek XMSException

Komunikat CreateStream-Tworzenie komunikatu strumienia

Interfejs:

```
IStreamMessage CreateStreamMessage();
```

Utwórz komunikat strumienia.

Parametry:
Brak

Zwraca:
Obiekt StreamMessage .

Wyjątki:

- Wyjątek XMSException
- XMS_ILLEGAL_STATE_EXCEPTION

CreateTemporaryKolejka-tworzenie kolejki tymczasowej

Interfejs:

```
IDestination CreateTemporaryQueue() ;
```

Utwórz kolejkę tymczasową.

Zasięg kolejki tymczasowej to połączenie. Tylko sesje utworzone przez połączenie mogą korzystać z kolejki tymczasowej.

Kolejka tymczasowa pozostaje do czasu jawnego usunięcia lub połączenia kończą się, w zależności od tego, co nastąpi wcześniej.

Więcej informacji na temat kolejek tymczasowych zawiera sekcja “Tymczasowe miejsca docelowe” na stronie 32.

Parametry:
Brak

Zwraca:
Obiekt docelowy reprezentujący kolejkę tymczasową.

Wyjątki:

- Wyjątek XMSException

Temat CreateTemporary-Tworzenie tematu tymczasowego

Interfejs:

```
IDestination CreateTemporaryTopic() ;
```

Utwórz temat tymczasowy.

Zasięgiem tematu tymczasowego jest połączenie. Tylko sesje utworzone przez połączenie mogą korzystać z tematu tymczasowego.

Wątek tymczasowy pozostaje do czasu jawnego usunięcia lub zakończenia połączenia, w zależności od tego, która z tych dat jest wcześniejsza.

Więcej informacji na temat tematów tymczasowych zawiera sekcja “Tymczasowe miejsca docelowe” na stronie 32.

Parametry:
Brak

Zwraca:
Obiekt docelowy reprezentujący temat tymczasowy.

Wyjątki:

- Wyjątek XMSException

Komunikat CreateText-Tworzenie komunikatu tekstowego

Interfejs:

```
ITextMessage CreateTextMessage();
```

Utwórz wiadomość tekstową z pustym ciałem.

Parametry:
Brak

Zwraca:
Obiekt TextMessage .

Wyjątki:

- Wyjątek XMSException

Komunikat CreateText-Tworzenie komunikatu tekstowego (zainicjowanego)

Interfejs:

```
ITextMessage CreateTextMessage(String initialValue);
```

Utwórz wiadomość tekstową, której treść jest inicjowana przy użyciu określonego tekstu.

Parametry:

initialValue (wejście)
Obiekt typu String hermetyzujący tekst w celu zainicjowania treści komunikatu tekstowego.

Brak

Zwraca:
Obiekt TextMessage .

Wyjątki:

- Wyjątek XMSException

CreateTopic -Tworzenie tematu

Interfejs:

```
IDestination CreateTopic(String topic) ;
```

Utwórz obiekt docelowy w celu reprezentowania tematu.

Parametry:

temat (wejście)
Obiekt typu String obudowujący nazwę tematu lub obudowujący jednolity identyfikator zasobu (URI), który identyfikuje dany temat.

Zwraca:
Obiekt docelowy reprezentujący temat.

Wyjątki:

- Wyjątek XMSException

Odzyskaj-Odzyskaj

Interfejs:

```
void Recover();
```

Odtwarzanie sesji. Dostarczanie komunikatów zostało zatrzymane, a następnie zrestartowane z najstarszym niepotwierdzonym komunikatem.

Sesja nie może być sesją transakcową.

Więcej informacji na temat odtwarzania sesji zawiera sekcja “Potwierdzenie komunikatu” na stronie 26.

Parametry:
Brak

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- Wyjątek IllegalState

Wycofaj-Wycofywanie zmian

Interfejs:

```
void Rollback();
```

wycofaj wszystkie komunikaty przetworzone w bieżącej transakcji.

Sesja musi być sesją transakcyjne.

Parametry:
Brak

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException

- Wyjątek IllegalState

Anuluj subskrypcję-Anuluj subskrypcję

Interfejs:

```
void Unsubscribe(String subscription);
```

Usuń trwałą subskrypcję. Serwer przesyłania komunikatów usuwa rekord trwałej subskrypcji, która jest konserwowana, i nie wysyła kolejnych komunikatów do trwałego subskrybenta.

Aplikacja nie może usunąć trwałej subskrypcji w żadnej z następujących sytuacji:

- Istnieje aktywny konsument komunikatów dla trwałej subskrypcji.
- Podczas gdy konsumowany komunikat jest częścią oczekującej transakcji
- Komunikat nie został potwierdzony, gdy nie został potwierdzony

Ta metoda nie jest poprawna dla połączenia w czasie rzeczywistym z brokerem.

Parametry:

subskrypcja (wejście)
Obiekt typu String hermetyzujący nazwę, która identyfikuje trwałą subskrypcję.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- Wyjątek InvalidDestination
- Wyjątek IllegalState

Odziedziczone właściwości i metody

Następujące metody zostały odziedziczone po interfejsie IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

IStreamMessage

Komunikat strumienia to komunikat, którego treść zawiera strumień wartości, w którym każda wartość ma powiązany typ danych. Treść treści jest zapisywana i odczytywaana sekwencyjnie.

Hierarchia dziedziczenia:

```
IBM.XMS.IPropertyContext
   |
   +----IBM.XMS.IMessage
           |
           +----IBM.XMS.IStreamMessage
```

Gdy aplikacja odczytuje wartość ze strumienia komunikatów, wartość może zostać przekształcona przez program XMS na inny typ danych. Więcej informacji na temat tego rodzaju konwersji niejawnej zawiera sekcja “Komunikaty strumienia” na stronie 81.

Odsyłacze pokrewne
Komunikaty strumienia

Treść komunikatu strumienia zawiera strumień wartości, w którym każda wartość ma powiązany typ danych.

Metody

ReadBoolean -odczytywanie wartości boolowskiej

Interfejs:

```
Boolean ReadBoolean();
```

Odczytywanie wartości boolowskiej ze strumienia komunikatów.

Parametry:
Brak

Zwraca:
Wartość boolowska, która jest odczytywalna.

Wyjątki:

- Wyjątek XMSException
- MessageNotReadableException
- MessageEOFException

ReadByte -Odczyt bajtu

Interfejs:

```
Int16   ReadSignedByte();
    Byte    ReadByte();
```

Przeczytaj 8-bitową liczbę całkowitą ze strumienia komunikatów.

Parametry:
Brak

Zwraca:
Bajt, który jest odczytany.

Wyjątki:

- Wyjątek XMSException
- MessageNotReadableException
- MessageEOFException

ReadBytes -odczytane bajty

Interfejs:

```
Int32   ReadBytes(Byte[] array);
```

Odczytywanie tablicy bajtów ze strumienia komunikatów.

Parametry:

tablica (wejście)
Bufor zawierający tablicę bajtów, która jest odczytywaana, oraz długość buforu w bajtach.

Jeśli liczba bajtów w tablicy jest mniejsza lub równa długości buforu, cała tablica jest odczytywaana do buforu. Jeśli liczba bajtów w tablicy jest większa niż długość buforu, bufor jest wypełniany częścią tablicy, a kursor wewnętrzny oznacza pozycję następnego bajtu, który ma zostać odczytany. Kolejne wywołanie funkcji readBytes(odczytywanie bajtów) odczytuje bajty z tablicy, począwszy od bieżącej pozycji kursora.

Jeśli w danych wejściowych zostanie określony pusty wskaźnik, wywołanie pomija tablicę bajtów bez jej odczytu.

Zwraca:
Liczba bajtów, które zostały odczytane w buforze. Jeśli bufor jest zapełniony częściowo, wartość jest mniejsza niż długość buforu, co oznacza, że nie ma więcej bajtów w tablicy, która ma zostać odczyta. Jeśli przed wywołaniem nie pozostały żadne bajty, które mają być odczytane z tablicy, wartością jest XMSC_END_OF_BYTEARRAY.

Jeśli w danych wejściowych zostanie określony pusty wskaźnik, metoda nie zwróci żadnej wartości.

Wyjątki:

- Wyjątek XMSException
- MessageNotReadableException
- MessageEOFException

ReadChar -Odczyt Znaku

Interfejs:

```
Char    ReadChar();
```

Odczytaj 2-bajtowy znak ze strumienia komunikatów.

Parametry:
Brak

Zwraca:
Znak, który jest odczytyowany.

Wyjątki:

- Wyjątek XMSException
- MessageNotReadableException
- MessageEOFException

ReadDouble -Odczyt dwukrotnego numeru zmiennopozycyjnego o podwójnej precyzji

Interfejs:

```
Double  ReadDouble();
```

Przeczytaj 8-bajtową liczbę zmiennopozycyjną o podwójnej precyzji z strumienia komunikatów.

Parametry:
Brak

Zwraca:
Liczba zmiennopozycyjna o podwójnej precyzji, która jest odczytywaana.

Wyjątki:

- Wyjątek XMSException

- MessageNotReadableException
- MessageEOFException

ReadFloat -odczyt numeru punktu zmiennopozycyjnego

Interfejs:

```
Single  ReadFloat();
```

Przeczytaj 4-bajtowy numer zmiennopozycyjny ze strumienia komunikatów.

Parametry:
Brak

Zwraca:
Liczba zmiennopozycyjna, która jest odczytywaana.

Wyjątki:

- Wyjątek XMSException
- MessageNotReadableException
- MessageEOFException

ReadInt -Odczyt typu Integer

Interfejs:

```
Int32   ReadInt();
```

Przeczytaj 32-bitową liczbę całkowitą ze strumienia komunikatów.

Parametry:
Brak

Zwraca:
Liczba całkowita, która jest odczytywaana.

Wyjątki:

- Wyjątek XMSException
- MessageNotReadableException
- MessageEOFException

ReadLong -Odczyt Long Integer

Interfejs:

```
Int64   ReadLong();
```

Przeczytaj podpisaną 64-bitową liczbę całkowitą ze strumienia komunikatów.

Parametry:
Brak

Zwraca:
Długa liczba całkowita, która jest odczytywaana.

Wyjątki:

- Wyjątek XMSException
- MessageNotReadableException
- MessageEOFException

ReadObject -Odczyt Obiektu

Interfejs:

```
Object  ReadObject();
```

Przeczytaj wartość ze strumienia komunikatów i zwróć jego typ danych.

Parametry:
Brak

Zwraca:
Wartość, która jest jednym z następujących typów obiektów:

```
Boolean
Byte
Byte[]
Char
Double
Single
Int32
Int64
Int16
String
```

Wyjątki:
Wyjątek XMSException

ReadShort -Read Short Integer

Interfejs:

```
Int16   ReadShort();
```

Przeczytaj 16-bitową liczbę całkowitą ze strumienia komunikatów.

Parametry:
Brak

Zwraca:
Krótka liczba całkowita, która jest odczytywaana.

Wyjątki:

- Wyjątek XMSException
- MessageNotReadableException
- MessageEOFException

ReadString -Odczyt łańcucha

Interfejs:

```
String  ReadString();
```

Przeczytaj łańcuch ze strumienia komunikatów. Jeśli jest to wymagane, program XMS przekształca znaki w łańcuchu w lokalną stronę kodową.

Parametry:
Brak

Zwraca:
Obiekt typu String obudowujący odczytany łańcuch. Jeśli konwersja danych jest wymagana, jest to łańcuch po konwersji.

Wyjątki:

- Wyjątek XMSException
- MessageNotReadableException
- MessageEOFException

Resetuj-Resetuj

Interfejs:

```
void  Reset();
```

Umieść treść komunikatu w trybie tylko do odczytu i przełóż kursor na początku strumienia komunikatów.

Parametry:
Brak

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- MessageNotReadableException
- MessageEOFException

WriteBoolean -zapis wartości boolowskiej

Interfejs:

```
void  WriteBoolean(Boolean value);
```

Zapis wartości boolowskiej do strumienia komunikatów.

Parametry:

wartość (wejście)
Wartość boolowska, która ma zostać zapisana.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException

- MessageNotWritableException

WriteByte -Zapis Bajt

Interfejs:

```
void  WriteByte(Byte value);
void  WriteSignedByte(Int16 value);
```

Zapis bajtu do strumienia komunikatów.

Parametry:

wartość (wejście)
Bajt, który ma zostać zapisany.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

WriteBytes -Zapis Bajtów

Interfejs:

```
void  WriteBytes(Byte[] value);
```

Zapis tablicy bajtów do strumienia komunikatów.

Parametry:

wartość (wejście)
Tablica bajtów, które mają zostać zapisane.

długość (wejście)
Liczba bajtów w tablicy.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

WriteChar -Zapis Znaku

Interfejs:

```
void  WriteChar(Char value);
```

Napisz znak do strumienia komunikatów jako 2 bajty, pierwszy bajt o wysokiej kolejności.

Parametry:

wartość (wejście)
Znak, który ma zostać zapisany.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

WriteDouble -Liczba Zmiennopozycyjna O Podwójnej Precyzji

Interfejs:

```
void  WriteDouble(Double value);
```

Przekształć liczbę zmiennopozycyjną podwójnej precyzji w długą liczbę całkowitą, a następnie zapisz długą liczbę całkowitą w strumieniu komunikatów jako 8 bajtów, pierwszy bajt o wysokiej kolejności.

Parametry:

wartość (wejście)
Liczba zmiennopozycyjna podwójnej precyzji, która ma zostać zapisana.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

WriteFloat -numer zmiennopozycyjnego zapisu

Interfejs:

```
void  WriteFloat(Single value);
```

Przekształć liczbę zmiennopozycyjną w liczbę całkowitą i napisz liczbę całkowitą do strumienia komunikatów jako 4 bajty, pierwsze bajt o wysokiej kolejności.

Parametry:

wartość (wejście)
Liczba zmiennopozycyjna, która ma zostać zapisana.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

WriteInt -Zapis Integer

Interfejs:

```
void  WriteInt(Int32 value);
```

Wpisz liczbę całkowitą do strumienia komunikatów jako 4 bajty, pierwszy bajt o wysokiej kolejności.

Parametry:

wartość (wejście)
Liczba całkowita, która ma zostać zapisana.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

WriteLong -zapis Long Integer

Interfejs:

```
void  WriteLong(Int64 value);
```

Zapis długiej liczby całkowitej do strumienia komunikatów jako 8 bajtów, najpierw w bajcie o wysokiej kolejności.

Parametry:

wartość (wejście)
Długa liczba całkowita, która ma zostać zapisana.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

WriteObject -Zapis Obiektu

Interfejs:

```
void  WriteObject(Object value);
```

Do strumienia komunikatów należy zapisać wartość, z określonym typem danych.

Parametry:

objectType (wejście)
Wartość, która musi być jednym z następujących typów obiektów:

```
Boolean
Byte
Byte[]
Char
Double
Single
Int32
Int64
Int16
String
```

wartość (wejście)
Tablica bajtów zawierająca wartość, która ma zostać zapisana.

długość (wejście)
Liczba bajtów w tablicy.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException

WriteShort -Zapis Short Integer

Interfejs:

```
void  WriteShort(Int16 value);
```

Wpisz krótką liczbę całkowitą do strumienia komunikatów jako 2 bajty, pierwsze bajt o wysokiej kolejności.

Parametry:

wartość (wejście)
Krótka liczba całkowita, która ma zostać zapisana.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

WriteString -zapis łańcucha

Interfejs:

```
void  WriteString(String value);
```

Napisz łańcuch do strumienia komunikatów.

Parametry:

wartość (wejście)
Obiekt typu String obudowujący łańcuch, który ma zostać zapisany.

Zwraca:
Unieważnione

Wyjątki:

- Wyjątek XMSException
- MessageNotWritableException

Odziedziczone właściwości i metody

Następujące właściwości zostały odziedziczone po interfejsie IMessage:

JMSCorrelationID, JMSDeliveryMode, JMSDestination, JMSExpiration, JMSMessageID, JMSPriority, JMSRedelivered, JMSReplyTo, JMSTimestamp, JMSType, Properties

Następujące metody zostały odziedziczone po interfejsie IMessage:

clearBody, clearProperties, PropertyExists

Następujące metody zostały odziedziczone po interfejsie IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

ITextMessage

Komunikat tekstowy jest komunikatem, którego treść składa się z łańcucha.

Hierarchia dziedziczenia:

```
IBM.XMS.IPropertyContext
   |
   +----IBM.XMS.IMessage
           |
           +----IBM.XMS.ITextMessage
```

Odsyłacze pokrewne

Komunikaty tekstowe
Treść wiadomości tekstowej zawiera łańcuch.

Właściwości produktu .NET

Tekst-Pobierz i ustaw tekst

Interfejs:

```
String Text
{
  get;
  set;
}
```

Pobierz i ustaw łańcuch, który tworzy treść wiadomości tekstowej.

Jeśli jest to wymagane, program XMS przekształca znaki w łańcuchu w lokalną stronę kodową.

Wyjątki:

- Wyjątek XMSException
- MessageNotReadableException
- MessageNotWritableException
- MessageEOFException

Odziedziczone właściwości i metody

Następujące właściwości zostały odziedziczone po interfejsie IMessage:

JMSCorrelationID, JMSDeliveryMode, JMSDestination, JMSExpiration, JMSMessageID, JMSPriority, JMSRedelivered, JMSReplyTo, JMSTimestamp, JMSType, Properties

Następujące metody zostały odziedziczone po interfejsie IMessage:

clearBody, clearProperties, PropertyExists

Następujące metody zostały odziedziczone po interfejsie IPropertyContext:

GetBooleanProperty, GetByteProperty, GetBytesProperty, GetCharProperty, GetDoubleProperty, GetFloatProperty, GetIntProperty, GetLongProperty, GetObjectProperty, GetShortProperty, GetStringProperty, SetBooleanProperty, SetByteProperty, SetBytesProperty, SetCharProperty, SetDoubleProperty, SetFloatProperty, SetIntProperty, SetLongProperty, SetObjectProperty, SetShortProperty, SetStringProperty

TransactionInProgressException

Hierarchia dziedziczenia:

```
IBM.XMS.XMSException
   |
   +----IBM.XMS.XMSException
           |
           +----IBM.XMS.TransactionInProgressException
```

XMS zgłasza ten wyjątek, jeśli aplikacja żąda operacji, która nie jest poprawna, ponieważ transakcja jest w toku.

Odziedziczone właściwości i metody

Następujące metody zostały odziedziczone po interfejsie XMSException:

GetErrorCode, GetLinkedException

TransactionRolledBackException

Hierarchia dziedziczenia:

```
IBM.XMS.XMSException
   |
   +----IBM.XMS.XMSException
           |
           +----IBM.XMS.TransactionRolledBackException
```

XMS zgłasza ten wyjątek, jeśli aplikacja wywołuje funkcję Session.commit() w celu zatwierdzenia bieżącej transakcji, ale transakcja jest wycofana.

Odziedziczone właściwości i metody

Następujące metody zostały odziedziczone po interfejsie XMSException:

GetErrorCode, GetLinkedException

Wyjątek XMSException

Jeśli program XMS wykryje błąd podczas przetwarzania wywołania metody .NET , XMS zgłasza wyjątek. Wyjątek stanowi obiekt, który hermetyzuje informacje o błędzie.

Hierarchia dziedziczenia:

```
System.Exception
   |
   +----IBM.XMS.XMSException
```

Istnieją różne typy wyjątków XMS , a obiekt XMSException to tylko jeden typ wyjątku. Klasa XMSException jest jednak nadklasą innych klas wyjątków XMS . XMS zgłasza obiekt XMSException w sytuacjach, w których żaden z pozostałych typów wyjątków nie jest odpowiedni.

Właściwości produktu .NET

ErrorCode -pobranie kodu błędu

Interfejs:

```
public String ErrorCode
{
  get {return errorCode_;}
}
```

Pobierz kod błędu.

Wyjątki:

- Wyjątek XMSException

LinkedException -Pobranie połączonego wyjątku

Interfejs:

```
public Exception LinkedException
{
  get { return linkedException_;}
  set { linkedException_ = value;}
}
```

Pobierz następny wyjątek w łańcuchu wyjątków.

Metoda zwraca wartość NULL, jeśli w łańcuchu nie ma więcej wyjątków.

Wyjątki:

- Wyjątek XMSException

XMSFactoryFactory

Jeśli aplikacja nie używa administrowanych obiektów, należy użyć tej klasy w celu utworzenia fabryk połączeń, kolejek i tematów.

Hierarchia dziedziczenia:
Brak

Właściwości produktu .NET

Metadane-pobieranie metadanych

Interfejs:

```
IConnectionMetaData MetaData
```

Pobierz metadane odpowiednie dla typu połączenia obiektu XMSFactoryFactory .

Wyjątki:
Brak

Metody

Fabryka połączeń CreateConnection-tworzenie fabryki połączeń

Interfejs:

```
IConnectionFactory CreateConnectionFactory();
```

Utwórz obiekt ConnectionFactory dla zadeklarowanego typu.

Parametry:
Brak

Zwraca:
Obiekt ConnectionFactory .

Wyjątki:

- Wyjątek XMSException

CreateQueue -Tworzenie kolejki

Interfejs:

```
IDestination CreateQueue(String name);
```

Utwórz obiekt docelowy w celu reprezentowania kolejki na serwerze przesyłania komunikatów.

Ta metoda nie tworzy kolejki na serwerze przesyłania komunikatów. Aby aplikacja mogła wywołać tę metodę, należy utworzyć kolejkę.

Parametry:

nazwa (wejście)
Obiekt typu String obudowujący nazwę kolejki lub obudowujący jednolity identyfikator zasobu (URI), który identyfikuje kolejkę.

Zwraca:
Obiekt docelowy reprezentujący kolejkę.

Wyjątki:

- Wyjątek XMSException

CreateTopic -Tworzenie tematu

Interfejs:

```
IDestination CreateTopic(String name);
```

Utwórz obiekt docelowy w celu reprezentowania tematu.

Parametry:

nazwa (wejście)
Obiekt typu String obudowujący nazwę tematu lub obudowujący jednolity identyfikator zasobu (URI), który identyfikuje dany temat.

Zwraca:
Obiekt docelowy reprezentujący temat.

Wyjątki:

- Wyjątek XMSException

GetInstance -pobieranie instancji klasy XMSFactoryFactory

Interfejs:

```
static XMSFactoryFactory GetInstance(int connectionType);
```

Utwórz instancję klasy XMSFactoryFactory. Aplikacja XMS korzysta z obiektu XMSFactoryFactory w celu uzyskania odniesienia do obiektu ConnectionFactory , który jest odpowiedni dla wymaganego typu protokołu. Ten obiekt ConnectionFactory może następnie generować połączenia tylko dla tego typu protokołu.

Parametry:

connectionType (wejście)
Typ połączenia, dla którego obiekt ConnectionFactory generuje połączenia:

- XMSC.CT_WPM
- XMSC.CT_RTT
- XMSC.CT_WMQ

Zwraca:
Obiekt XMSFactoryFactory dedykowany do zadeklarowanego typu połączenia.

Wyjątki:

- Wyjątek NotSupported

Właściwości obiektów XMS

Ta rozdział dokumentuje właściwości obiektu zdefiniowane przez produkt XMS.

rozdział zawiera następujące sekcje:

Każdy sekcja zawiera listę właściwości obiektu określonego typu i udostępnia krótki opis każdej właściwości.

Ten rozdział zawiera również "Definicje właściwości" na stronie 199 sekcja, który udostępnia definicję każdej właściwości.

Jeśli aplikacja definiuje własne właściwości obiektów opisanych w tym rozdział, nie powoduje to błędu, ale może spowodować nieprzewidywalne rezultaty.

Uwaga: Nazwy i wartości właściwości w tej sekcji są wyświetlane w postaci XMSC.*NAME*, która jest formularzem używanym dla języka C i C + +. Jednak w programie .NETnazwa właściwości może mieć postać XMSC.*NAME* lub XMSC_*NAME*, w zależności od tego, w jaki sposób jest używana.

- W przypadku określania właściwości nazwa właściwości musi mieć postać XMSC.*NAME* , tak jak przedstawiono to w poniższym przykładzie:

```
cf.SetStringProperty(XMSC.WMQ_CHANNEL, "DOTNET.SVRCONN");
```

- Jeśli zostanie podany łańcuch, nazwa właściwości musi mieć postać XMSC_*NAME* , tak jak pokazano w poniższym przykładzie:

```
cf.SetStringProperty("XMSC_WMQ_CHANNEL", "DOTNET.SVRCONN");
```

  W programie .NETnazwy właściwości i wartości są udostępniane jako stałe w klasie XMSC. Te stałe identyfikują łańcuchy i mogą być używane przez dowolną aplikację XMS .NET . Jeśli te predefiniowane stałe są używane, nazwy i wartości właściwości są w postaci XMSC.*NAZWA*, dlatego na przykład można użyć wyrażenia XMSC.USERID, a nie XMSC_USERID.

Typy danych znajdują się również w formularzu używanym dla C/C + +. Odpowiednie wartości można znaleźć w produkcie .NET w produkcie“Typy danych dla .NET” na stronie 45.

Pojęcia pokrewne

Budowanie własnych aplikacji
Użytkownik buduje własne aplikacje, takie jak kompilacja przykładowych aplikacji.

Odsyłacze pokrewne

.NET interfejsy
Ten sekcja dokumentuje interfejsy klasy .NET oraz ich właściwości i metody.

Właściwości połączenia

Przegląd właściwości obiektu Connection, z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

Tabela 25. Właściwości połączenia

Nazwa właściwości	Opis
“XMSC_WMQ_RESOLVED_QUEUE_MANAGER” na stronie 233	Ta właściwość jest używana do uzyskiwania nazwy menedżera kolejek, z którym jest on połączony.
“XMSC_WMQ_RESOLVED_QUEUE_MANAGER_ID” na stronie 233	Ta właściwość jest zapełniana za pomocą identyfikatora menedżera kolejek po połączeniu.
PROTOKÓŁ XMSC_WPM_CONNECTION_PROTOCOL	Protokół komunikacyjny używany na potrzeby połączenia z mechanizmem przesyłania komunikatów. Ta właściwość jest tylko do odczytu.
XMSC_WPM_HOST_NAME	Położenie mechanizmu przesyłania komunikatów, z którym połączona jest aplikacja (nazwa hosta lub adres IP systemu). Ta właściwość jest tylko do odczytu.
XMSC_WPM_ME_NAME	Nazwa mechanizmu przesyłania komunikatów, z którym połączona jest aplikacja. Ta właściwość jest tylko do odczytu.
PORT XMSC_WPM_PORT	Numer portu, na którym nasłuchuje mechanizm przesyłania komunikatów, z którym aplikacja nawiązała połączenie. Ta właściwość jest tylko do odczytu.

Obiekt połączenia ma również właściwości tylko do odczytu, które zostały utworzone na podstawie właściwości fabryki połączeń, która została użyta do utworzenia połączenia. Te właściwości są wyprowadzane nie tylko z właściwości fabryki połączeń, które zostały ustawione w momencie tworzenia połączenia, ale również z domyślnych wartości właściwości, które nie zostały ustawione. Właściwości obejmują tylko te, które są istotne dla typu serwera przesyłania komunikatów, z którym połączona jest aplikacja. Nazwy właściwości są takie same, jak nazwy właściwości fabryki połączeń.

Właściwości obiektu ConnectionFactory

Przegląd właściwości obiektu ConnectionFactory z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

Tabela 26. Właściwości obiektu ConnectionFactory	
Nazwa właściwości	**Opis**
“XMSC_ASYNC_EXCEPTIONS,” na stronie 208	Ta właściwość określa, czy aplikacja XMS ma informować interfejs ExceptionListener tylko wtedy, gdy połączenie zostało zerwane, czy też wtedy, gdy dowolny wyjątek wystąpi asynchronicznie w wywołaniu interfejsu API XMS. Ta właściwość ma zastosowanie względem wszystkich połączeń utworzonych w tej fabryce połączeń, które mają zarejestrowany interfejs ExceptionListener.
ID_KLIENTA XMSC_XML	Identyfikator klienta dla połączenia.
XMSC_CONNECTION_TYPE	Typ serwera przesyłania komunikatów, z którym łączy się aplikacja.
XMSC_PASSWORD	Hasło, które może być używane do uwierzytelniania aplikacji na etapie podejmowania próby nawiązania połączenia z serwerem przesyłania komunikatów.
“XMSC_RTT_BROKER_PING_INTERVAL” na stronie 214	Odstęp czasu (w milisekundach), po upływie którego klient XMS .NET sprawdza połączenie z serwerem przesyłania komunikatów w czasie rzeczywistym w celu wykrycia dowolnego działania.
PROTOKÓŁ XMSC_RTT_CONNECTION_PROTOCOL	Protokół komunikacyjny używany do nawiązywania połączenia z brokerem w czasie rzeczywistym.
XMSC_RTT_HOST_NAME	Miejsce działania brokera: nazwa hosta lub adres IP systemu.
ADRES_LOKALNY XMSC_RTT_LOCAL_ADDRESS	Nazwa hosta lub adres IP interfejsu sieci lokalnej na potrzeby nawiązania połączenia z brokerem w czasie rzeczywistym.
XMSC_RTT_MULTICAST	Ustawienie rozsyłania grupowego dla fabryki połączeń lub miejsca docelowego.
PORT XMSC_RTT_PORT	Numer portu, na którym broker nasłuchuje żądań przychodzących.
ID_UŻYTKOWNIKA XMSC_USERID	Identyfikator użytkownika, który może być używany do uwierzytelniania aplikacji na etapie podejmowania próby nawiązania połączenia z serwerem przesyłania komunikatów.
XMSC_WMQ_BROKER_CONTROLQ	Nazwa kolejki sterującej używanej przez broker. **Uwaga:** Ta właściwość może być używana z wersją 2.0 produktu IBM Message Service Client for .NET , ale nie ma wpływu na aplikację połączoną z menedżerem kolejek produktu IBM WebSphere MQ 7.0 , chyba że właściwość XMSC_WMQ_PROVIDER_VERSION fabryki połączeń jest ustawiona na numer wersji mniejszy niż 7.

Tabela 26. Właściwości obiektu ConnectionFactory (kontynuacja)	
Nazwa właściwości	**Opis**
XMSC_WMQ_BROKER_PUBQ	Nazwa kolejki monitorowanej przez broker, do której aplikacje wysyłają publikowane komunikaty. **Uwaga:** Ta właściwość może być używana z wersją 2.0 produktu IBM Message Service Client for .NET , ale nie ma wpływu na aplikację połączoną z menedżerem kolejek produktu IBM WebSphere MQ 7.0 , chyba że właściwość XMSC_WMQ_PROVIDER_VERSION fabryki połączeń jest ustawiona na numer wersji mniejszy niż 7.
XMSC_WMQ_BROKER_QMGR	Nazwa menedżera kolejek, z którym broker nawiązał połączenie. **Uwaga:** Ta właściwość może być używana z wersją 2.0 produktu IBM Message Service Client for .NET , ale nie ma wpływu na aplikację połączoną z menedżerem kolejek produktu IBM WebSphere MQ 7.0 , chyba że właściwość XMSC_WMQ_PROVIDER_VERSION fabryki połączeń jest ustawiona na numer wersji mniejszy niż 7.
XMSC_WMQ_BROKER_SUBQ	Nazwa kolejki subskrybenta dla konsumenta nietrwałych komunikatów. **Uwaga:** Ta właściwość może być używana z wersją 2.0 produktu IBM Message Service Client for .NET , ale nie ma wpływu na aplikację połączoną z menedżerem kolejek produktu IBM WebSphere MQ 7.0 , chyba że właściwość XMSC_WMQ_PROVIDER_VERSION fabryki połączeń jest ustawiona na numer wersji mniejszy niż 7.
XMSC_WMQ_BROKER_VERSION	Typ brokera używany na potrzeby połączenia lub miejsca docelowego. **Uwaga:** Ta właściwość może być używana z wersją 2.0 produktu IBM Message Service Client for .NET , ale nie ma wpływu na aplikację połączoną z menedżerem kolejek produktu IBM WebSphere MQ 7.0 , chyba że właściwość XMSC_WMQ_PROVIDER_VERSION fabryki połączeń jest ustawiona na numer wersji mniejszy niż 7.
“XMSC_WMQ_CCDTURL” na stronie 219	Adres URL identyfikujący nazwę i położenie pliku zawierającego tabelę definicji kanałów klienta oraz określający sposób dostępu do pliku.
XMSC_WMQ_CHANNEL	Nazwa kanału używanego do nawiązywania połączenia.
“XMSC_WMQ_CLIENT_RECONNECT_OPTIONS” na stronie 220	Ta właściwość określa opcje ponownego połączenia klienta dla nowych połączeń utworzonych przez tę fabrykę.
“XMSC_WMQ_CLIENT_RECONNECT_TIMEOUT” na stronie 220	Ta właściwość określa czas (w sekundach), przez który połączenie klienta próbuje ponownie nawiązać połączenie.
XMSC_WMQ_CONNECTION_MODE	Tryb, przy użyciu którego aplikacja nawiązuje połączenie z menedżerem kolejek.
“XMSC_WMQ_CONNECTION_NAME_LIST” na stronie 221	Ta właściwość określa hosty, do których klient próbuje ponownie nawiązać połączenie po zerowaniu połączenia.

Tabela 26. Właściwości obiektu ConnectionFactory (kontynuacja)	
Nazwa właściwości	**Opis**
XMSC_WMQ_FAIL_IF_QUIESCE	Określa, czy wywołania pewnych metod nie powiodą się, jeśli menedżer kolejek, z którym aplikacja nawiązała połączenie, będzie w stanie wyciszania.
XMSC_WMQ_HOST_NAME	Miejsce działania menedżera kolejek: nazwa hosta lub adres IP systemu.
XMSC_WMQ_LOCAL_ADDRESS	W przypadku połączenia z menedżerem kolejek: ta właściwość określa używaną wartość interfejsu sieci lokalnej albo portu lokalnego (lub zakresu portów lokalnych) bądź obie te wartości.
XMSC_WMQ_MESSAGE_SELECTION	Określa, czy wybór komunikatów jest dokonany przez klienta XMS , czy przez brokera. **Uwaga:** Ta właściwość może być używana z wersją 2.0 produktu IBM Message Service Client for .NET , ale nie ma wpływu na aplikację połączoną z menedżerem kolejek produktu IBM WebSphere MQ 7.0 , chyba że właściwość XMSC_WMQ_PROVIDER_VERSION fabryki połączeń jest ustawiona na numer wersji mniejszy niż 7.
XMSC_WMQ_MSG_BATCH_SIZE	Maksymalna liczba komunikatów pobieranych z kolejki w jednej partii, gdy komunikaty są dostarczane w trybie asynchronicznym. **Uwaga:** Ta właściwość może być używana z wersją 2.0 produktu IBM Message Service Client for .NET , ale nie ma wpływu na aplikację połączoną z menedżerem kolejek produktu IBM WebSphere MQ 7.0 , chyba że właściwość XMSC_WMQ_PROVIDER_VERSION fabryki połączeń jest ustawiona na numer wersji mniejszy niż 7.
XMSC_WMQ_POLLING_INTERVAL	Jeśli kolejka żadnego procesu nasłuchującego w ramach sesji nie zawiera odpowiedniego komunikatu, jest to maksymalny odstęp czasu (w milisekundach) między kolejnymi próbami pobrania komunikatu z kolejki przez każdy z procesów nasłuchujących komunikatów. **Uwaga:** Ta właściwość może być używana z wersją 2.0 produktu IBM Message Service Client for .NET , ale nie ma wpływu na aplikację połączoną z menedżerem kolejek produktu IBM WebSphere MQ 7.0 , chyba że właściwość XMSC_WMQ_PROVIDER_VERSION fabryki połączeń jest ustawiona na numer wersji mniejszy niż 7.
“XMSC_WMQ_PROVIDER_VERSION” na stronie 230	Wersja, wydanie, poziom modyfikacji i pakiet poprawek menedżera kolejek, z którym aplikacja ma nawiązać połączenie.
PORT XMSC_WMQ_PORT	Numer portu, na którym menedżer kolejek nasłuchuje żądań przychodzących.

Tabela 26. Właściwości obiektu ConnectionFactory (kontynuacja)	
Nazwa właściwości	**Opis**
XMSC_WMQ_PUB_ACK_INTERVAL	Liczba komunikatów publikowanych przez publikatora, zanim klient XMS zażąda potwierdzenia od brokera. **Uwaga:** Ta właściwość może być używana z wersją 2.0 produktu IBM Message Service Client for .NET , ale nie ma wpływu na aplikację połączoną z menedżerem kolejek produktu IBM WebSphere MQ 7.0 , chyba że właściwość XMSC_WMQ_PROVIDER_VERSION fabryki połączeń jest ustawiona na numer wersji mniejszy niż 7.
"XMSC_WMQ_PUT_ASYNC_ALLOWED" na stronie 226	Ta właściwość określa, czy producenci komunikatów mogą używać asynchronicznych operacji put w celu wysyłania komunikatów do tego miejsca docelowego.
CCSID XMSC_WMQ_QMGR_CCSID	Identyfikator (CCSID) kodowanego zestawu znaków lub strony kodowej, w którym pola danych znakowych zdefiniowane w interfejsie kolejki komunikatów (Message Queue Interface-MQI) są wymieniane między klientem XMS a klientem WebSphere MQ .
MENEDŻER KOLEJEK XMSC_WMQ_QUEUE_MANAGER	Nazwa menedżera kolejek, z którym ma zostać nawiązane połączenie.
XMSC_WMQ_RECEIVE_EXIT	Identyfikuje wyjście odbierania kanału, które ma zostać uruchomione.
XMSC_WMQ_RECEIVE_EXIT_INIT	Dane użytkownika przekazywane do wyjścia odbierania kanału w momencie jego wywołania.
XMSC_WMQ_SECURITY_EXIT	Identyfikuje wyjście zabezpieczeń kanału.
XMSC_WMQ_SECURITY_EXIT_INIT	Dane użytkownika przekazywane do wyjścia zabezpieczeń kanału w momencie jego wywołania.
"XMSC_WMQ_SEND_CHECK_COUNT" na stronie 235	Liczba wywołań wysyłania dozwolonych między sprawdzeniami błędów asynchronicznego umieszczania w ramach jednej sesji XMS bez transakcji.
XMSC_WMQ_SEND_EXIT	Identyfikuje wyjście wysyłania kanału.
XMSC_WMQ_SEND_EXIT_INIT	Dane użytkownika przekazywane do wyjść wysyłania kanału w momencie ich wywołania.
"XMSC_WMQ_SHARE_CONV_ALLOWED" na stronie 235	Określa, czy połączenie klienta może współużytkować swoje gniazdo z innymi połączeniami XMS najwyższego poziomu z tego samego procesu do tego samego menedżera kolejek, jeśli definicje kanałów są zgodne. Ta właściwość umożliwia pełną izolację połączeń w osobnych gniazdach (jeśli jest to konieczne z powodów związanych z tworzeniem, konserwacją lub działaniem aplikacji).
XMSC_WMQ_SSL_CERT_STORES	Lokalizacje serwerów, na których znajdują się listy odwołań certyfikatów (Certificate Revocation List – CRL) używane podczas nawiązywania połączenia SSL z menedżerem kolejek.
XMSC_WMQ_SSL_CIPHER_SPEC	Nazwa specyfikacji szyfrowania używanej w przypadku bezpiecznego połączenia z menedżerem kolejek.

Tabela 26. Właściwości obiektu ConnectionFactory (kontynuacja)	
Nazwa właściwości	**Opis**
XMSC_WMQ_SSL_CIPHER_SUITE	Nazwa zestawu algorytmów szyfrowania używanego w przypadku połączenia TLS z menedżerem kolejek. Na podstawie podanego zestawu algorytmów szyfrowania jest określany protokół używany do negocjowania bezpiecznego połączenia.
XMSC_WMQ_SSL_CRYPTO_HW	Szczegóły konfiguracji sprzętu szyfrującego połączonego z systemem klienta.
XMSC_WMQ_SSL_FIPS_REQUIRED	Wartość tej właściwości określa, czy aplikacja może lub nie może używać zestawów algorytmów szyfrowania niezgodnych ze standardem FIPS. Jeśli ta właściwość zostanie ustawiona na wartość true, w przypadku połączeń klient-serwer będzie można używać tylko algorytmów zgodnych ze standardem FIPS.
XMSC_WMQ_SSL_KEY_REPOSITORY	Położenie pliku bazy danych kluczy, w którym przechowywane są klucze i certyfikaty.
XMSC_WMQ_SSL_KEY_RESETCOUNT	Wartość KeyResetCount reprezentuje całkowitą liczbę nieszyfrowanych bajtów wysłanych i odebranych w ramach konwersacji SSL przed ponownym wynegocjowaniem klucza tajnego.
XMSC_WMQ_SSL_PEER_NAME	Nazwa węzła sieci używana na potrzeby połączenia SSL z menedżerem kolejek.
XMSC_WMQ_SYNCPOINT_ALL_GETS	Określa, czy wszystkie komunikaty muszą być pobierane z kolejek w ramach sterowania punktem synchronizacji.
“KLIENT XMSC_WMQ_TARGET_CLIENT” na stronie 242	
XMSC_WMQ_TEMP_Q_PREFIX	Przedrostek używany do utworzenia nazwy kolejki dynamicznej WebSphere MQ utworzonej podczas tworzenia kolejki tymczasowej produktu XMS przez aplikację.
XMSC_WMQ_TEMP_TOPIC_PREFIX	Podczas tworzenia tematów tymczasowych XMS generuje łańcuch tematu w postaci "`TEMP/TEMPTOPICPREFIX/unique_id`" lub jeśli ta właściwość zawiera wartość domyślną, to ten łańcuch, "`TEMP/unique_id`", zostanie wygenerowany. Podanie niepustej wartości umożliwia definiowanie konkretnych kolejek modelowych na potrzeby tworzenia zarządzanych kolejek dla subskrybentów tymczasowych tematów utworzonych w ramach tego połączenia.
MODEL XMSC_WMQ_TEMPORARY_MODEL	Nazwa kolejki modelowej WebSphere MQ , z której tworzona jest kolejka dynamiczna, gdy aplikacja tworzy kolejkę tymczasową XMS .
XMSC_WPM_BUS_NAME	W przypadku fabryki połączeń: nazwa magistrali integracji usług, z którą aplikacja nawiązuje połączenie. W przypadku miejsca docelowego: nazwa magistrali integracji usług, w której znajduje się miejsce docelowe.

Tabela 26. Właściwości obiektu ConnectionFactory (kontynuacja)	
Nazwa właściwości	**Opis**
XMSC_WPM_CONNECTION_ZBLIŻENIA	Ustawienie bliskości połączeń na potrzeby nawiązywania połączenia.
XMSC_WPM_DUR_SUB_HOME	Nazwa mechanizmu przesyłania komunikatów zarządzającego wszystkimi trwałymi subskrypcjami połączeń lub miejsc docelowych.
ADRES_LOKALNY_XMSC_WPM_	W przypadku połączenia z magistralą integracji usług: ta właściwość określa używaną wartość interfejsu sieci lokalnej albo portu lokalnego (lub zakresu portów lokalnych) bądź obie te wartości.
XMSC_WPM_NON_PERSISTENT_MAP	Poziom niezawodności komunikatów nietrwałych wysyłanych za pośrednictwem połączenia.
XMSC_WPM_PERSISTENT_MAP	Poziom niezawodności komunikatów trwałych wysyłanych za pośrednictwem połączenia.
PUNKTY KOŃCOWE XMSC_WPM_PROVIDER_ENDPOINTS	Jeden adres punktu końcowego serwera startowego lub sekwencja wielu adresów.
XMSC_WPM_TARGET_GROUP	Nazwa grupy docelowej mechanizmów przesyłania komunikatów.
XMSC_WPM_TARGET_ISTOTNOŚĆ	Znaczenie grupy docelowej mechanizmów przesyłania komunikatów.
XMSC_WPM_TARGET_TRANSPORT_CHAIN	Nazwa łańcucha transportowego danych przychodzących, który musi być używany przez aplikację do nawiązywania połączenia z mechanizmem przesyłania komunikatów.
XMSC_WPM_TARGET_TYPE	Typ grupy docelowej mechanizmów przesyłania komunikatów.
XMSC_WPM_TEMP_Q_PREFIX	Przedrostek używany do tworzenia nazwy kolejki tymczasowej, która jest tworzona na magistrali integracji usług, gdy aplikacja tworzy kolejkę tymczasową XMS .
XMSC_WPM_TEMP_TOPIC_PREFIX	Przedrostek używany do generowania nazwy tematu tymczasowego tworzonego przez aplikację.

Pojęcia pokrewne

Obiekty ConnectionFactories i obiekty Connection
Obiekt ConnectionFactory udostępnia szablon, który jest używany przez aplikację do tworzenia obiektu połączenia. Aplikacja korzysta z obiektu połączenia w celu utworzenia obiektu sesji.

Połączenie z magistralą integracji usług
Aplikacja XMS może łączyć się z magistralą integracji usług WebSphere Application Server za pomocą bezpośredniego połączenia TCP/IP lub za pomocą protokołu HTTP przy użyciu protokołu TCP/IP.

Bezpieczne połączenia z menedżerem kolejek produktu IBM MQ
Aby umożliwić aplikacji XMS .NET nawiąże bezpieczne połączenia z menedżerem kolejek produktu IBM MQ , odpowiednie właściwości muszą zostać zdefiniowane w obiekcie ConnectionFactory .

Bezpieczne połączenia z mechanizmem przesyłania komunikatów produktu WebSphere Application Server service integration bus
Aby umożliwić aplikacji XMS .NET nawiąże bezpieczne połączenia z mechanizmem przesyłania komunikatów produktu WebSphere Application Server service integration bus , odpowiednie właściwości muszą zostać zdefiniowane w obiekcie ConnectionFactory .

Odwzorowanie właściwości dla administrowanych obiektów

Aby umożliwić aplikacjom korzystanie z fabryki połączeń produktu IBM MQ JMS i WebSphere Application Server oraz definicji obiektów docelowych, właściwości pobrane z tych definicji muszą być odwzorowane na odpowiednie właściwości produktu XMS , które można ustawić w fabrykach połączeń i miejscach docelowych produktu XMS .

Zadania pokrewne

Tworzenie obiektów administrowanych
Definicje obiektów ConnectionFactory i Destination, które są wymagane przez aplikacje produktu XMS w celu nawiązania połączenia z serwerem przesyłania komunikatów, muszą zostać utworzone przy użyciu odpowiednich narzędzi administracyjnych.

Odsyłacze pokrewne

Wymagane właściwości dla administrowanych obiektów ConnectionFactory
Gdy aplikacja tworzy fabrykę połączeń, należy zdefiniować pewną liczbę właściwości, aby utworzyć połączenie z serwerem przesyłania komunikatów.

Właściwości danych ConnectionMeta

Przegląd właściwości obiektu danych ConnectionMeta, z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

Tabela 27. Właściwości danych ConnectionMeta

Nazwa właściwości	Opis
XMSC_JMS_MAJOR_VERSION	Numer wersji głównej specyfikacji JMS , na której oparty jest produkt XMS . Ta właściwość jest tylko do odczytu.
XMSC_JMS_MINOR_VERSION	Drugorzędny numer wersji specyfikacji JMS , na której oparty jest produkt XMS . Ta właściwość jest tylko do odczytu.
XMSC_JMS_VERSION	Identyfikator wersji specyfikacji JMS , na której oparty jest produkt XMS . Ta właściwość jest tylko do odczytu.
XMSC_MAJOR_VERSION	Numer wersji klienta XMS . Ta właściwość jest tylko do odczytu.
XMSC_MINOR_VERSION	Numer wydania klienta XMS . Ta właściwość jest tylko do odczytu.
XMSC_PROVIDER_NAME	Dostawca klienta XMS . Ta właściwość jest tylko do odczytu.
XMSC_VERSION	Identyfikator wersji klienta XMS . Ta właściwość jest tylko do odczytu.

Właściwości miejsca docelowego

Przegląd właściwości obiektu docelowego, z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

Tabela 28. Właściwości miejsca docelowego

Nazwa właściwości	Opis
XMSC_DELIVERY_MODE	Tryb dostarczania komunikatów wysyłanych do miejsca docelowego.
XMSC_PRIORITY	Priorytet komunikatów wysyłanych do miejsca docelowego.
XMSC_RTT_MULTICAST	Ustawienie rozsyłania grupowego dla fabryki połączeń lub miejsca docelowego.

Tabela 28. Właściwości miejsca docelowego (kontynuacja)	
Nazwa właściwości	**Opis**
XMSC_TIME_TO_LIVE	Czas życia komunikatów wysyłanych do miejsca docelowego.
XMSC_WMQ_BROKER_VERSION	Typ brokera używany na potrzeby połączenia lub miejsca docelowego.
XMSC_WMQ_CCSID	Identyfikator (identyfikator CCSID) kodowanego zestawu znaków lub strony kodowej, w którym łańcuchy danych znakowych w treści komunikatu są wyświetlane, gdy klient XMS przekazuje komunikat do miejsca docelowego.
XMSC_WMQ_DUR_SUBQ	Nazwa kolejki trwałego subskrybenta, który odbiera komunikaty z miejsca docelowego. **Uwaga:** Ta właściwość może być używana z wersją 2.0 produktu IBM Message Service Client for .NET , ale nie ma wpływu na aplikację połączoną z menedżerem kolejek produktu IBM WebSphere MQ 7.0 , chyba że właściwość XMSC_WMQ_PROVIDER_VERSION fabryki połączeń jest ustawiona na numer wersji mniejszy niż 7.
XMSC_WMQ_ENCODING	W jaki sposób dane liczbowe w treści komunikatu są reprezentowane, gdy klient XMS przekazuje komunikat do miejsca docelowego.
XMSC_WMQ_FAIL_IF_QUIESCE	Określa, czy wywołania pewnych metod nie powiodą się, jeśli menedżer kolejek, z którym aplikacja nawiązała połączenie, będzie w stanie wyciszania.
“XMSC_WMQ_MESSAGE_BODY” na stronie 224	Ta właściwość określa, czy aplikacja XMS przetwarza obiekt MQRFH2 komunikatu produktu IBM WebSphere MQ jako część ładunku komunikatu (czyli jako część treści komunikatu).
“XMSC_WMQ_MQMD_MESSAGE_CONTEXT” na stronie 224	Określa poziom kontekstu komunikatu, który ma być ustawiany przez aplikację XMS . Aby ta właściwość miała zastosowanie, aplikacja musi być uruchamiana z odpowiednimi uprawnieniami dotyczącymi kontekstu.
“XMSC_WMQ_MQMD_READ_ENABLED” na stronie 225	Ta właściwość określa, czy aplikacja XMS może wyodrębnić wartości pól MQMD, czy nie.
“XMSC_WMQ_MQMD_WRITE_ENABLED” na stronie 226	Ta właściwość określa, czy aplikacja XMS może mieć wartości pól MQMD, czy nie.
“XMSC_WMQ_READ_AHEAD_ALLOWED” na stronie 226	Ta właściwość określa, czy konsumenty komunikatów i przeglądarki kolejek mogą używać operacji odczytu z wyprzedzeniem do pobierania nietrwałych komunikatów nietransakcyjnych z tego miejsca docelowego do buforu wewnętrznego przed ich odebraniem.
“XMSC_WMQ_READ_AHEAD_CLOSE_POLICY” na stronie 227	W przypadku komunikatów dostarczanych do asynchronicznego procesu nasłuchującego komunikatów ta właściwość określa, co stanie się z komunikatami w wewnętrznym buforze odczytu z wyprzedzeniem, gdy konsument komunikatów zostanie zamknięty.

Tabela 28. Właściwości miejsca docelowego (kontynuacja)	
Nazwa właściwości	**Opis**
“CCSID XMSC_WMQ_RECEIVE_CCSID” na stronie 232	Właściwość docelowa, która ustawia docelowy identyfikator CCSID dla konwersji komunikatów menedżera kolejek. Wartość jest ignorowana, chyba że parametr XMSC_WMQ_RECEIVE_CONVERSION jest ustawiony na wartość WMQ_RECEIVE_CONVERSION_QMGR.
“XMSC_WMQ_RECEIVE_CONVERSION” na stronie 232	Właściwość miejsca docelowego, która określa, czy konwersja danych ma być wykonywana przez menedżer kolejek.
XMSC_WMQ_TARGET_CLIENT	Określa, czy komunikaty wysyłane do miejsca docelowego mają zawierać nagłówek MQRFH2.
XMSC_WMQ_TEMP_TOPIC_PREFIX	Podczas tworzenia tematów tymczasowych XMS generuje łańcuch tematu w postaci "TEMP/TEMPTOPICPREFIX/unique_id" lub jeśli ta właściwość zawiera wartość domyślną, to ten łańcuch, "TEMP/unique_id", zostanie wygenerowany. Podanie niepustej wartości umożliwia definiowanie konkretnych kolejek modelowych na potrzeby tworzenia zarządzanych kolejek dla subskrybentów tymczasowych tematów utworzonych w ramach tego połączenia.
XMSC_WPM_BUS_NAME	W przypadku fabryki połączeń: nazwa magistrali integracji usług, z którą aplikacja nawiązuje połączenie. W przypadku miejsca docelowego: nazwa magistrali integracji usług, w której znajduje się miejsce docelowe.
XMSC_WPM_TOPIC_SPACE	Nazwa obszaru tematu zawierającego dany temat.

Pojęcia pokrewne

Obiekty ConnectionFactories i obiekty Connection
Obiekt ConnectionFactory udostępnia szablon, który jest używany przez aplikację do tworzenia obiektu połączenia. Aplikacja korzysta z obiektu połączenia w celu utworzenia obiektu sesji.

Połączenie z magistralą integracji usług
Aplikacja XMS może łączyć się z magistralą integracji usług WebSphere Application Server za pomocą bezpośredniego połączenia TCP/IP lub za pomocą protokołu HTTP przy użyciu protokołu TCP/IP.

Miejsca docelowe
Aplikacja XMS używa obiektu docelowego do określenia miejsca docelowego wysyłanych komunikatów oraz źródła komunikatów, które są odbierane.

Znaki wieloznaczne miejsca docelowego
Produkt XMS udostępnia obsługę znaków wieloznacznych w celu zapewnienia, że znaki wieloznaczne mogą być przekazywane do miejsca, w którym są one potrzebne do dopasowania. Dla każdego typu serwera, z którym może pracować produkt XMS , istnieje inny schemat znaków wieloznacznych.

Identyfikatory ujednolicona zasobu tematu
Identyfikator URI (Uniform Resource Identifier) tematu określa nazwę tematu. Można również określić dla niego jedną lub więcej właściwości.

Identyfikatory URI jednorodnych zasobów
Identyfikator URI kolejki określa nazwę kolejki. Można również określić jedną lub więcej właściwości kolejki.

Tymczasowe miejsca docelowe
Aplikacje produktu XMS mogą tworzyć tymczasowe miejsca docelowe i korzystać z nich.

Odwzorowanie właściwości dla administrowanych obiektów

Aby umożliwić aplikacjom korzystanie z fabryki połączeń produktu IBM MQ JMS i WebSphere Application Server oraz definicji obiektów docelowych, właściwości pobrane z tych definicji muszą być odwzorowane na odpowiednie właściwości produktu XMS , które można ustawić w fabrykach połączeń i miejscach docelowych produktu XMS .

Zadania pokrewne

Tworzenie obiektów administrowanych
Definicje obiektów ConnectionFactory i Destination, które są wymagane przez aplikacje produktu XMS w celu nawiązania połączenia z serwerem przesyłania komunikatów, muszą zostać utworzone przy użyciu odpowiednich narzędzi administracyjnych.

Odsyłacze pokrewne

Wymagane właściwości administrowanych obiektów docelowych
Aplikacja tworzący miejsce docelowe musi ustawić kilka właściwości, które aplikacja ma w administrowanego obiekcie docelowym.

Właściwości obiektu InitialContext

Przegląd właściwości obiektu InitialContext , z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

Tabela 29. Właściwości obiektu InitialContext

Nazwa właściwości	Opis
URI_DOSTAWCY XMSC_IC_PROVIDER_URL	Umożliwia wskazanie katalogu nazw JNDI. Dzięki niej usługa nazw COS nie musi znajdować się na tym samym serwerze co usługa WWW.
XMSC_IC_SECURITY_AUTHENTICATION	W oparciu o interfejs kontekstu Java SECURITY_AUTHENTICATION. Ta właściwość ma zastosowanie tylko do kontekstu nazewnictwa COS.
XMSC_IC_SECURITY_CREDENTIALS	W oparciu o interfejs kontekstu Java SECURITY_CREDENTIALS. Ta właściwość ma zastosowanie tylko do kontekstu nazewnictwa COS.
XMSC_IC_SECURITY_PRINCIPAL	W oparciu o interfejs kontekstu Java SECURITY_PRINCIPAL. Ta właściwość ma zastosowanie tylko do kontekstu nazewnictwa COS.
XMSC_IC_SECURITY_PROTOCOL	Na podstawie interfejsu kontekstu Java SECURITY_PROTOCOL Ta właściwość ma zastosowanie tylko do kontekstu nazewnictwa COS.
XMSC_IC_URL	W przypadku kontekstów LDAP i FileSystem: adres repozytorium zawierającego obiekty administrowane. W przypadku kontekstów nazw COS: adres usługi WWW, która wyszukuje obiekty w katalogu.

Pojęcia pokrewne

Właściwości obiektu InitialContext
Parametry konstruktora InitialContext obejmują położenie repozytorium administrowanych obiektów, podane jako jednolity wskaźnik zasobów (URI). Aby aplikacja mogła nawiązać połączenie z repozytorium, może być konieczne podanie większej ilości informacji niż informacje zawarte w identyfikatorze URI.

Format identyfikatora URI dla kontekstów początkowych produktu XMS
Położenie repozytorium administrowanych obiektów jest udostępniane jako jednolity wskaźnik zasobów (URI). Format identyfikatora URI zależy od typu kontekstu.

Pobieranie administrowanych obiektów
Program XMS pobiera obiekt administrowany z repozytorium przy użyciu adresu udostępnionego podczas tworzenia obiektu InitialContext lub w właściwościach InitialContext .

Zadania pokrewne

Obiekty InitialContext

Aplikacja musi utworzyć kontekst początkowy, który będzie używany w celu nawiązania połączenia z repozytorium obiektów administrowanych w celu pobrania wymaganych obiektów administrowanych.

Właściwości komunikatu

Przegląd właściwości obiektu Message, z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

Tabela 30. Właściwości komunikatu

Nazwa właściwości	Opis
ZESTAW JMS_IBM_CHARACTER_SET	Identyfikator (CCSID) kodowanego zestawu znaków lub strony kodowej, w którym łańcuchy danych znakowych w treści komunikatu są używane, gdy klient XMS przekazuje komunikat do zamierzonego miejsca docelowego. W polu XMS ta właściwość ma wartość liczbową i jest odwzorowywać na identyfikator CCSID. Ta właściwość jest jednak oparta na właściwości JMS, dlatego przyjmuje wartość typu String, która jest odwzorowywana na zestaw znaków języka Java reprezentujący ten liczbowy identyfikator CCSID.
JMS_IBM_Encoding	W jaki sposób dane liczbowe w treści komunikatu są reprezentowane, gdy klient XMS przekazuje komunikat do zamierzonego miejsca docelowego.
JMS_IBM_EXCEPTIONMESSAGE	Tekst opisujący przyczynę, z powodu której komunikat został wysłany do miejsca docelowego wyjątków. Ta właściwość jest tylko do odczytu.
JMS_IBM_EXCEPTIONPROBLEMDESTINATION	Nazwa miejsca docelowego, w którym znajdował się komunikat, zanim został wysłany do miejsca docelowego wyjątków.
JMS_IBM_EXCEPTIONREASON	Kod przyczyny wskazujący powód wysłania komunikatu do miejsca docelowego wyjątków.
JMS_IBM_EXCEPTIONTIMESTAMP	Czas wysłania komunikatu do miejsca docelowego wyjątków.
JMS_IBM_FEEDBACK	Kod wskazujący rodzaj komunikatu raportu.
JMS_IBM_FORMAT	Przyroda danych aplikacji w komunikacie.
GRUPA_JMS_IBM_LAST_MSG_IN_GROUP	Wskazuje, czy komunikat jest ostatnim komunikatem w grupie komunikatów.
JMS_IBM_MSGTYPE	Typ komunikatu.
JMS_IBM_PUTAPPLTYPE	Typ aplikacji, która wysłała komunikat.
DATA JMS_IBM_PUTDATE	Data wysłania komunikatu.
JMS_IBM_PUTTIME	Godzina wysłania komunikatu.
JMS_IBM_REPORT_COA	Wysyła żądanie przesyłania komunikatów raportów 'potwierdzenie odebrania' i określa, ile danych aplikacji z oryginalnego komunikatu musi zostać dołączonych do komunikatu raportu.

Tabela 30. Właściwości komunikatu (kontynuacja)	
Nazwa właściwości	**Opis**
JMS_IBM_REPORT_COD	Wysyła żądanie przesyłania komunikatów raportów 'potwierdzenie dostarczenia' i określa, ile danych aplikacji z oryginalnego komunikatu musi zostać dołączonych do komunikatu raportu.
JMS_IBM_REPORT_DISCARD_MSG	Wysyła żądanie, aby komunikat był usuwany, jeśli nie może zostać dostarczony do wybranego miejsca docelowego.
JMS_IBM_REPORT_EXCEPTION	Wysyła żądanie przesyłania komunikatów raportów o wyjątkach i określa, ile danych aplikacji z oryginalnego komunikatu musi zostać dołączonych do komunikatu raportu.
JMS_IBM_REPORT_EXPIRATION	Wysyła żądanie przesyłania komunikatów raportów o utracie ważności i określa, ile danych aplikacji z oryginalnego komunikatu musi zostać dołączonych do komunikatu raportu.
JMS_IBM_REPORT_NAN	Wysyła żądanie przesyłania komunikatów raportów o powiadomieniach dotyczących działań negatywnych.
JMS_IBM_REPORT_PAN	Wysyła żądanie przesyłania komunikatów raportów o powiadomieniach dotyczących działań pozytywnych.
JMS_IBM_REPORT_PASS_CORREL_ID	Wysyła żądanie, aby identyfikator korelacji dowolnego komunikatu odpowiedzi lub raportu był taki sam jak identyfikator korelacji oryginalnego komunikatu.
JMS_IBM_REPORT_PASS_MSG_ID	Wysyła żądanie, aby identyfikator dowolnego komunikatu odpowiedzi lub raportu był taki sam jak identyfikator oryginalnego komunikatu.
JMS_IBM_RETAIN	Jeśli ta właściwość zostanie ustawiona, menedżer kolejek będzie traktował komunikat jako zachowaną publikację.
JMS_IBM_SYSTEM_MESSAGEID	Identyfikator, który identyfikuje komunikat w jednoznaczny sposób w obrębie magistrali integracji usług. Ta właściwość jest tylko do odczytu.
JMSX_APPID	Nazwa aplikacji, która wysłała komunikat.
JMSX_DELIVERY_COUNT	Liczba prób dostarczenia komunikatu.
ID_GROUP_JMS_JMS_JMS	Identyfikator grupy komunikatów, do której należy komunikat.
JMSX_GROUPSEQ	Numer kolejny komunikatu w obrębie grupy komunikatów.
JMSX_USERID	Identyfikator użytkownika powiązany z aplikacją, która wysłała komunikat.

Właściwości JMS_IBM_MQMD*

Produkt IBM Message Service Client for .NET umożliwia aplikacjom klienckim odczytywanie/zapisywanie pól MQMD za pomocą funkcji API. Umożliwia również dostęp do danych komunikatu produktu MQ . Domyślnie dostęp do deskryptora MQMD jest wyłączony i musi być jawnie włączony przez aplikację przy użyciu właściwości miejsca docelowego: XMSC_WMQ_MQMD_WRITE_ENABLED i XMSC_WMQ_MQMD_READ_ENABLED. Te dwie właściwości są niezależne od siebie.

Wszystkie pola MQMD z wyjątkiem StrucId i Version są ujawniane jako dodatkowe właściwości obiektu Message i są to prefixed JMS_IBM_MQMD.

Właściwości JMS_IBM_MQMD* mają wyższy priorytet niż inne właściwości, takie jak JMS_IBM* opisane w poprzedniej tabeli.

Wysyłanie komunikatów

Wszystkie pola MQMD z wyjątkiem StrucId i Version są reprezentowane. Te właściwości odnoszą się tylko do pól MQMD. W przypadku, gdy właściwość występuje zarówno w strukturze MQMD, jak i w nagłówku MQRFH2 , wersja w tabeli MQRFH2 nie jest ustawiona lub wyodrębniana. Możliwe jest ustawienie dowolnej z tych właściwości, z wyjątkiem właściwości JMS_IBM_MQMD_BackoutCount. Dowolna wartość ustawiona dla JMS_IBM_MQMD_BackoutCount jest ignorowana.

Jeśli właściwość ma maksymalną długość, a użytkownik poda wartość, która jest zbyt długa, wartość jest obcinana.

W przypadku niektórych właściwości należy również ustawić właściwość XMSC_WMQ_MQMD_MESSAGE_CONTEXT w obiekcie docelowym. Aby ta właściwość miała zastosowanie, aplikacja musi być uruchamiana z odpowiednimi uprawnieniami dotyczącymi kontekstu. Jeśli wartość właściwości XMSC_WMQ_MQMD_MESSAGE_CONTEXT nie zostanie ustawiona na odpowiednią wartość, wartość właściwości zostanie zignorowana. Jeśli wartość XMSC_WMQ_MQMD_MESSAGE_CONTEXT zostanie ustawiona na odpowiednią wartość, ale użytkownik nie ma wystarczających uprawnień kontekstowych dla menedżera kolejek, zostanie wygenerowany wyjątek. Właściwości wymagające konkretnych wartości atrybutu XMSC_WMQ_MQMD_MESSAGE_CONTEXT są następujące.

Następujące właściwości wymagają ustawienia XMSC_WMQ_MQMD_MESSAGE_CONTEXT na wartość XMSC_WMQ_MDCTX_SET_IDENTITY_CONTEXT lub XMSC_WMQ_MDCTX_SET_ALL_CONTEXT.

- JMS_IBM_MQMD_UserIdentifier
- JMS_IBM_MQMD_AccountingToken
- Dane JMS_IBM_MQMD_ApplIdentity

Następujące właściwości wymagają ustawienia XMSC_WMQ_MQMD_MESSAGE_CONTEXT na wartość XMSC_WMQ_MDCTX_SET_ALL_CONTEXT:

- Typ JMS_IBM_MQMD_PutAppl
- Nazwa JMS_IBM_MQMD_PutAppl
- JMS_IBM_MQMD_PutDate
- JMS_IBM_MQMD_PutTime
- Dane JMS_IBM_MQMD_ApplOrigin

Odbieranie komunikatów

Wszystkie te właściwości są dostępne w odebranym komunikacie, jeśli właściwość XMSC_WMQ_MQMD_READ_ENABLED jest ustawiona na wartość true, bez względu na rzeczywiste właściwości ustawione przez aplikację produkujący_aplikację. Aplikacja nie może modyfikować właściwości odebranego komunikatu, chyba że wszystkie właściwości zostaną wyczyszczone jako pierwsze zgodnie ze specyfikacją JMS. Odebrany komunikat może zostać przesłany bez modyfikowania właściwości.

Uwaga: Jeśli aplikacja odbierze komunikat z miejsca docelowego z właściwością XMSC_WMQ_MQMD_READ_ENABLED ustawioną na wartość true, a następnie przekazuje ją do miejsca docelowego z ustawioną wartością true dla parametru XMSC_WMQ_MQMD_WRITE_ENABLED, spowoduje to, że wszystkie wartości pól MQMD odebranego komunikatu są kopiowane do przekazanego komunikatu. Tabela właściwości

Tabela 31. Właściwości obiektu komunikatu reprezentującego pola MQMD		
Właściwość	**Opis**	**Typ**
Raport JMS_IBM_MQMD_REPORT	Opcje dla komunikatów raportu	System.Int32
JMS_IBM_MQMD_MSGTYPE	Typ komunikatu	System.Int32
JMS_IBM_MQMD_WAŻNOŚCI	czas życia komunikatu	System.Int32
JMS_IBM_MQMD_FEEDBACK	Informacja zwrotna lub kod przyczyny	System.Int32
JMS_IBM_MQMD_ENCODING	Kodowanie numeryczne danych komunikatu	System.Int32
JMS_IBM_MQMD_CODEDCHARSETID	Identyfikator zestawu znaków danych komunikatu	System.Int32
FORMAT JMS_IBM_MQMD_FORMAT	Nazwa formatu danych komunikatu	System.String
JMS_IBM_MQMD_PRIORITY **Uwaga:** Jeśli do wartości JMS_IBM_MQMD_PRIORITY zostanie przypisana wartość, która nie mieści się w zakresie od 0 do 9, ta wartość narusza specyfikację JMS.	Priorytet komunikatu	System.Int32
JMS_IBM_MQMD_PERSISTENCE	Trwałość komunikatu	System.Int32
Identyfikator MSGID JMS_IBM_MQMD_MSGID **Uwaga:** Specyfikacja JMS określa, że identyfikator komunikatu musi być ustawiony przez dostawcę JMS i musi być unikalny lub ma wartość NULL. Jeśli do wartości JMS_IBM_MQMD_MSGID zostanie przypisana wartość, ta wartość zostanie skopiowana do wartości JMSMessageID. Dlatego nie jest on ustawiany przez dostawcę JMS i może nie być unikalny: ta wartość narusza specyfikację JMS.	Identyfikator komunikatu	Tablica bajtów **Uwaga:** Użycie właściwości tablicy bajtów w komunikacie narusza specyfikację JMS.
JMS_IBM_MQMD_CORRELID **Uwaga:** Po przypisaniu wartości do wartości JMS_IBM_MQMD_CORRELID rozpoczynający się od łańcucha 'ID:' ta wartość narusza specyfikację JMS.	Identyfikator korelacji	Tablica bajtów **Uwaga:** Użycie właściwości tablicy bajtów w komunikacie narusza specyfikację JMS.
JMS_IBM_MQMD_BACKOUTCOUNT	Liczba wycofanych	System.Int32
JMS_IBM_MQMD_REPLYTOQ	Nazwa kolejki odpowiedzi	System.String
JMS_IBM_MQMD_REPLYTOQMGR	Nazwa menedżera kolejek odpowiedzi	System.String
JMS_IBM_MQMD_USERIDENTIFIER	Identyfikator użytkownika	System.String

Tabela 31. Właściwości obiektu komunikatu reprezentującego pola MQMD (kontynuacja)		
Właściwość	**Opis**	**Typ**
JMS_IBM_MQMD_ACCOUNTINGTOKEN,	Token rozliczania	Tablica bajtów **Uwaga:** Użycie właściwości tablicy bajtów w komunikacie narusza specyfikację JMS.
JMS_IBM_MQMD_APPLIDENTITYDATA	Dane aplikacji odnoszące się do tożsamości	System.String
JMS_IBM_MQMD_PUTAPPLTYPE	Typ aplikacji, która wstawiła komunikat	System.Int32
JMS_IBM_MQMD_PUTAPPLNAME	Nazwa aplikacji umieszczonej w komunikacie.	System.String
Data PUTDATE JMS_IBM_MQMD_PUTDATE	Data umieszczenia komunikatu	System.String
Czas PUTTIME JMS_IBM_MQMD_PUTTIME	Czas umieszczenia komunikatu	System.String
JMS_IBM_MQMD_APPLORIGINDATA	Dane dotyczące wniosku dotyczące pochodzenia	System.String
Identyfikator grupy JMS_IBM_MQMD_GROUPID	Identyfikator grupy	Tablica bajtów **Uwaga:** Użycie właściwości tablicy bajtów w komunikacie narusza specyfikację JMS.
JMS_IBM_MQMD_MSGSEQNUMBER	Numer kolejny komunikatu lokalnego w grupie	System.Int32
JMS_IBM_MQMD_OFFSET	Przesunięcie danych w komunikacie fizycznym od początku komunikatu logicznego	System.Int32
JMS_IBM_MQMD_MSGFLAGS	Flagi komunikatu	System.Int32
JMS_IBM_MQMD_ORIGINALLENGTH	Długość oryginalnego komunikatu	System.Int32

Więcej informacji na ten temat zawiera sekcja MQMD .

Przykłady

Ten przykład powoduje, że komunikat jest umieszczany w kolejce lub w temacie MQMD.UserIdentifier jest ustawiony na wartość "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
// ...
```

Konieczne jest ustawienie XMSC_WMQ_MQMD_MESSAGE_CONTEXT przed ustawieniem JMS_IBM_MQMD_USERIDENTIFIER. Więcej informacji na temat korzystania z obiektu XMSC_WMQ_MQMD_MESSAGE_CONTEXT zawiera sekcja Właściwości obiektu komunikatu.

Podobnie można wyodrębnić zawartość pól MQMD, ustawiając właściwość XMSC_WMQ_MQMD_READ_ENABLED na wartość true przed odebraniem komunikatu, a następnie za pomocą metod pobierania komunikatu, na przykład właściwości getString. Wszystkie otrzymane właściwości są przeznaczone tylko do odczytu.

Ten przykład powoduje, że w polu wartości znajduje się wartość MQMD.ApplIdentityData : komunikat został zwrócony z kolejki lub tematu.

```
// 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);
```

Właściwości obiektu MessageConsumer

Przegląd właściwości obiektu MessageConsumer , z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

Tabela 32. Właściwości obiektu MessageConsumer

Nazwa właściwości	Opis
XMSC_IS_SUBSCRIPTION_MULTICAST	Wskazuje, czy komunikaty są dostarczane do konsumenta komunikatów przy użyciu produktu WebSphere MQ Multicast Transport. Ta właściwość jest tylko do odczytu.
XMSC_IS_SUBSCRIPTION_RELIABLE_MULTICAST	Wskazuje, czy komunikaty są dostarczane do konsumenta komunikatów przy użyciu produktu WebSphere MQ Multicast Transport z niezawodną jakością usługi. Ta właściwość jest tylko do odczytu.

Patrz .Właściwości NET IMessageConsumer , aby uzyskać więcej szczegółów.

Właściwości elementu MessageProducer

Przegląd właściwości obiektu MessageProducer , z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

Patrz .Właściwości NET IMessageProducer , aby uzyskać więcej szczegółów.

Właściwości sesji

Przegląd właściwości obiektu Session, z odsyłaczami do bardziej szczegółowych informacji referencyjnych.

Patrz .NET properties of ISession , aby uzyskać więcej szczegółów.

Definicje właściwości

Ta sekcja udostępnia definicję każdej właściwości obiektu.

Każda definicja właściwości zawiera następujące informacje:

- typ danych właściwości;
- Typy obiektów, które mają właściwość
- W przypadku właściwości miejsca docelowego: nazwa, która może być używana w identyfikatorze URI
- Bardziej szczegółowy opis właściwości
- Poprawne wartości właściwości
- Wartość domyślna właściwości.

Właściwości, których nazwy rozpoczynały się od jednego z następujących przedrostków, są istotne tylko dla określonego typu połączenia:

XMSC_RTT
Właściwości są istotne tylko w przypadku połączenia w czasie rzeczywistym z brokerem. Nazwy właściwości są definiowane jako stałe nazwane w pliku nagłówkowego `xmsc_rtt.h`.

XMSC_WMQ
Właściwości są istotne tylko wtedy, gdy aplikacja łączy się z menedżerem kolejek produktu WebSphere MQ . Nazwy właściwości są definiowane jako stałe nazwane w pliku nagłówkowego `xmsc_wmq.h`.

XMSC_WPM
Właściwości są istotne tylko wtedy, gdy aplikacja łączy się z magistralą integracji usług produktu WebSphere . Nazwy właściwości są definiowane jako stałe nazwane w pliku nagłówkowego `xmsc_wpm.h`.

O ile nie określono inaczej w definicjach, pozostałe właściwości są istotne dla wszystkich typów połączeń. Nazwy właściwości są definiowane jako stałe nazwane w pliku nagłówkowego `xmsc.h`. Właściwości, których nazwy są rozpoczynanie od przedrostka JMSX, to JMS zdefiniowane właściwości komunikatu, a właściwości, których nazwy rozpoczyna się od przedrostka JMS_IBM, to IBM zdefiniowane właściwości komunikatu. Więcej informacji na temat właściwości komunikatów zawiera sekcja “Właściwości komunikatu XMS” na stronie 73.

Jeśli definicja nie stanowi inaczej, każda właściwość jest istotna zarówno w domenie punkt z punktem , jak i w domenach Publikowanie/subskrypcja .

Aplikacja może uzyskać i ustawić wartość dowolnej właściwości, chyba że właściwość ta jest oznaczona jako tylko do odczytu.

JMS_IBM_CHARACTER_SET

Typ danych:
System.Int32

Właściwość:
Komunikat

Identyfikator (CCSID) kodowanego zestawu znaków lub strony kodowej, w którym łańcuchy danych znakowych w treści komunikatu są używane, gdy klient XMS przekazuje komunikat do zamierzonego miejsca docelowego. W polu XMS ta właściwość ma wartość liczbową i jest odwzorowywać na identyfikator CCSID. Ta właściwość jest jednak oparta na właściwości JMS, dlatego przyjmuje wartość typu String, która jest odwzorowywana na zestaw znaków języka Java reprezentujący ten liczbowy identyfikator CCSID. Ta właściwość przesłania dowolny identyfikator CCSID określony dla miejsca docelowego przez właściwość XMSC_WMQ_CCSID .

Domyślnie właściwość nie jest ustawiona.

Ta właściwość nie ma znaczenia, gdy aplikacja łączy się z magistralą integracji usług.

JMS_IBM_Encoding

Typ danych:
System.Int32

Właściwość:
Komunikat

W jaki sposób dane liczbowe w treści komunikatu są reprezentowane, gdy klient XMS przekazuje komunikat do zamierzonego miejsca docelowego. Ta właściwość przesłania dowolne kodowanie określone dla miejsca docelowego przez właściwość XMSC_WMQ_ENCODING . Ta właściwość określa reprezentację binarnych liczb całkowitych, upakowanych liczb całkowitych dziesiętnych i liczb zmiennopozycyjnych.

Poprawne wartości właściwości są takie same, jak wartości, które można określić w polu **Encoding** deskryptora komunikatu.

Do ustawienia właściwości aplikacja może używać następujących stałych nazwanych:

Stała nazwana	Znaczenie
MQENC_INTEGER_NORMAL	Normalne kodowanie liczb całkowitych
MQENC_INTEGER_REVERSED	Odwrócone kodowanie liczb całkowitych
MQENC_DECIMAL_NORMAL	Normalne upakowane kodowanie dziesiętne
MQENC_DECIMAL_REVERSED	Odwrócone spakowane kodowanie dziesiętne
MQENC_FLOAT_IEEE_NORMAL	Normalne kodowanie zmiennopozycyjne IEEE
MQENC_FLOAT_IEEE_REVERSED	Odwrócone kodowanie zmiennopozycyjne IEEE
MQENC_FLOAT_S390	Kodowanie zmiennopozycyjne architektury produktu z/OS
MQENC_NATIVE	Kodowanie rodzimego komputera

Aby utworzyć wartość dla właściwości, aplikacja może dodać trzy następujące stałe w następujący sposób:

- Stała, której nazwa rozpoczyna się od wywołania MQENC_INTEGER, w celu określenia reprezentacji binarnych liczb całkowitych
- Stała, której nazwa rozpoczyna się od MQENC_DECIMAL, w celu określenia reprezentacji upakowanych liczb całkowitych.
- Stała, której nazwa rozpoczyna się od MQENC_FLOAT, w celu określenia reprezentacji liczb zmiennopozycyjnych.

Alternatywnie aplikacja może ustawić właściwość na wartość MQENC_NATIVE, której wartość jest zależna od środowiska.

Domyślnie właściwość nie jest ustawiona.

Ta właściwość nie ma znaczenia, gdy aplikacja łączy się z magistralą integracji usług.

JMS_IBM_EXCEPTIONMESSAGE

Typ danych:
Łańcuch

Właściwość:
Komunikat

Tekst opisujący przyczynę, z powodu której komunikat został wysłany do miejsca docelowego wyjątków. Ta właściwość jest tylko do odczytu.

Ta właściwość ma znaczenie tylko wtedy, gdy aplikacja łączy się z magistralą integracji usług i odbiera komunikat z miejsca docelowego wyjątków.

JMS_IBM_EXCEPTIONPROBLEMDESTINATION

Typ danych:
Łańcuch

Właściwość:
Komunikat

Nazwa miejsca docelowego, w którym znajdował się komunikat, zanim został wysłany do miejsca docelowego wyjątków.

Ta właściwość ma znaczenie tylko wtedy, gdy aplikacja łączy się z magistralą integracji usług i odbiera komunikat z miejsca docelowego wyjątków.

JMS_IBM_EXCEPTIONREASON

Typ danych:
System.Int32

Właściwość:
Komunikat

Kod przyczyny wskazujący powód wysłania komunikatu do miejsca docelowego wyjątków.

Ta właściwość ma znaczenie tylko wtedy, gdy aplikacja łączy się z magistralą integracji usług i odbiera komunikat z miejsca docelowego wyjątków.

JMS_IBM_EXCEPTIONTIMESTAMP

Typ danych:
System.Int64

Właściwość:
Komunikat

Czas wysłania komunikatu do miejsca docelowego wyjątków.

Czas jest wyrażony w milisekundach od godziny 00:00:00 GMT w dniu 1 stycznia 1970 r.

Ta właściwość ma znaczenie tylko wtedy, gdy aplikacja łączy się z magistralą integracji usług i odbiera komunikat z miejsca docelowego wyjątków.

JMS_IBM_FEEDBACK

Typ danych:
System.Int32

Właściwość:
Komunikat

Kod wskazujący rodzaj komunikatu raportu.

Poprawnymi wartościami właściwości są kody sprzężenia zwrotnego i kody przyczyny, które można określić w polu **Feedback** deskryptora komunikatu.

Domyślnie właściwość nie jest ustawiona.

FORMAT JMS_IBM_FORMAT

Typ danych:
Łańcuch

Właściwość:
Komunikat

Przyroda danych aplikacji w komunikacie.

Poprawne wartości właściwości są takie same, jak wartości, które można określić w polu **Format** deskryptora komunikatu.

Domyślnie właściwość nie jest ustawiona.

Ta właściwość nie ma znaczenia, gdy aplikacja łączy się z magistralą integracji usług.

GRUPA_JMS_IBM_LAST_MSG_IN_GROUP

Typ danych:
System.Boolean

Właściwość:
Komunikat

Wskazuje, czy komunikat jest ostatnim komunikatem w grupie komunikatów.

Ustaw właściwość na wartość true (prawda), jeśli komunikat jest ostatnim komunikatem w grupie komunikatów. W przeciwnym razie ustaw właściwość na wartość false lub nie ustawiaj właściwości. Domyślnie właściwość nie jest ustawiona.

Wartość true odpowiada flagi statusu MQMF_LAST_MSG_IN_GROUP, która może być określona w polu **MsgFlags** deskryptora komunikatu.

Ta właściwość jest ignorowana w domenie Publikowanie/subskrypcja i nie ma znaczenia, gdy aplikacja łączy się z magistralą integracji usług.

TYP_JMS_IBM_MSGTYPE

Typ danych:
System.Int32

Właściwość:
Komunikat

Typ komunikatu.

Poprawne wartości właściwości są następujące:

Poprawna wartość	Znaczenie
MQMT_DATAGRAM	Komunikat jest taki, że nie wymaga odpowiedzi.
MQMT_REQUEST	Komunikat jest taki, że wymaga odpowiedzi.
MQMT_REPLY	Komunikat jest komunikatem odpowiedzi.
Raport_menedżera_mQMT	Komunikat jest komunikatem raportu.

Te wartości odpowiadają typom komunikatów, które mogą być określone w polu **MsgType** deskryptora komunikatu.

Domyślnie właściwość nie jest ustawiona.

Ta właściwość nie ma znaczenia, gdy aplikacja łączy się z magistralą integracji usług.

TYP_JMS_IBM_PUTAPPLTYPE

Typ danych:
System.Int32

Właściwość:
Komunikat

Typ aplikacji, która wysłała komunikat.

Poprawne wartości właściwości to typy aplikacji, które można określić w polu **PutApplType** deskryptora komunikatu.

Domyślnie właściwość nie jest ustawiona.

Ta właściwość nie ma znaczenia, gdy aplikacja łączy się z magistralą integracji usług.

DATA JMS_IBM_PUTDATE

Typ danych:
Łańcuch

Właściwość:
Komunikat

Data wysłania komunikatu.

Poprawne wartości właściwości są takie same, jak wartości, które można określić w polu **PutDate** deskryptora komunikatu.

Domyślnie właściwość nie jest ustawiona.

Ta właściwość nie ma znaczenia, gdy aplikacja łączy się z magistralą integracji usług.

CZAS JMS_IBM_PUTTIME

Typ danych:
Łańcuch

Właściwość:
Komunikat

Godzina wysłania komunikatu.

Poprawne wartości właściwości są takie same, jak wartości, które można określić w polu **PutTime** deskryptora komunikatu.

Domyślnie właściwość nie jest ustawiona.

Ta właściwość nie ma znaczenia, gdy aplikacja łączy się z magistralą integracji usług.

JMS_IBM_REPORT_COA

Typ danych:
System.Int32

Właściwość:
Komunikat

Wysyła żądanie przesyłania komunikatów raportów 'potwierdzenie odebrania' i określa, ile danych aplikacji z oryginalnego komunikatu musi zostać dołączonych do komunikatu raportu.

Poprawne wartości właściwości są następujące:

Poprawna wartość	Znaczenie
MQRO_COA	Żądanie "potwierdź w dniu przyjazdu" komunikaty raportu, bez danych aplikacji z oryginalnego komunikatu dołączonego do komunikatu raportu.
MQRO_COA_WITH_DATA	Żądanie 'potwierdź w momencie przybycia' raportu, z pierwszych 100 bajtów danych aplikacji z oryginalnego komunikatu dołączonego do komunikatu raportu.
MQRO_COA_WITH_FULL_DATA	Żądanie 'potwierdź w momencie przybycia' raportu, ze wszystkimi danymi aplikacji z oryginalnego komunikatu dołączonego do komunikatu raportu.

Te wartości odpowiadają opcji raportu, które można określić w polu **Report** deskryptora komunikatu. Więcej informacji na temat tych opcji znajduje się w sekcji Raport (MQLONG).

Domyślnie właściwość nie jest ustawiona.

JMS_IBM_REPORT_COD

Typ danych:
System.Int32

Właściwość:
Komunikat

Wysyła żądanie przesyłania komunikatów raportów 'potwierdzenie dostarczenia' i określa, ile danych aplikacji z oryginalnego komunikatu musi zostać dołączonych do komunikatu raportu.

Poprawne wartości właściwości są następujące:

Poprawna wartość	Znaczenie
MQRO_COD	Żądanie "potwierdź na dostarczeniu" komunikaty raportu, bez danych aplikacji z oryginalnego komunikatu dołączonego do komunikatu raportu.
MQRO_COD_WITH_DATA	Żądanie 'potwierdź w trakcie dostarczania' komunikatów, z pierwszymi 100 bajtów danych aplikacji z oryginalnego komunikatu dołączonego do komunikatu raportu.
MQRO_COD_WITH_FULL_DATA	Żądanie 'potwierdź w przypadku dostarczania' komunikatów, ze wszystkimi danymi aplikacji z oryginalnego komunikatu dołączonego do komunikatu raportu.

Te wartości odpowiadają opcji raportu, które można określić w polu **Report** deskryptora komunikatu.

Domyślnie właściwość nie jest ustawiona.

JMS_IBM_REPORT_DISCARD_MSG

Typ danych:
System.Int32

Właściwość:
Komunikat

Wysyła żądanie, aby komunikat był usuwany, jeśli nie może zostać dostarczony do wybranego miejsca docelowego.

Ustaw właściwość na MQRO_DISCARD_MSG, aby zażądać, aby komunikat został usunięty, jeśli nie można go dostarczyć do jego planowanego miejsca docelowego. Jeśli wymagane jest, aby komunikat został umieszczony w kolejce niedostarczanych komunikatów lub został wysłany do miejsca docelowego wyjątków, nie należy ustawiać tej właściwości. Domyślnie właściwość nie jest ustawiona.

Wartość MQRO_DISCARD_MSG odpowiada opcji raportu, która może być określona w polu **Report** deskryptora komunikatu.

JMS_IBM_REPORT_EXCEPTION

Typ danych:
System.Int32

Właściwość:
Komunikat

Wysyła żądanie przesyłania komunikatów raportów o wyjątkach i określa, ile danych aplikacji z oryginalnego komunikatu musi zostać dołączonych do komunikatu raportu.

Poprawne wartości właściwości są następujące:

Poprawna wartość	Znaczenie
MQRO_EXCEPTION	Komunikaty raportu o wyjątkach żądań, bez danych aplikacji z oryginalnego komunikatu dołączonego do komunikatu raportu.
MQRO_EXCEPTION_WITH_DATA	Komunikaty raportu o wyjątkach żądań, z pierwszych 100 bajtów danych aplikacji z oryginalnego komunikatu dołączonego do komunikatu raportu.
MQRO_EXCEPTION_WITH_FULL_DATA	Komunikaty raportu o wyjątkach żądań, wraz ze wszystkimi danymi aplikacji z oryginalnego komunikatu dołączonego do komunikatu raportu.

Te wartości odpowiadają opcji raportu, które można określić w polu **Report** deskryptora komunikatu.

Domyślnie właściwość nie jest ustawiona.

JMS_IBM_REPORT_EXPIRATION

Typ danych:
System.Int32

Właściwość:
Komunikat

Wysyła żądanie przesyłania komunikatów raportów o utracie ważności i określa, ile danych aplikacji z oryginalnego komunikatu musi zostać dołączonych do komunikatu raportu.

Poprawne wartości właściwości są następujące:

Poprawna wartość	Znaczenie
MQRO_EXPIRATION	Komunikaty raportu o utracie ważności żądania, bez danych aplikacji z oryginalnego komunikatu dołączonego do komunikatu raportu.
MQRO_EXPIRATION_WITH_DATA	Komunikaty raportu o utracie ważności żądania, z pierwszych 100 bajtów danych aplikacji z oryginalnego komunikatu dołączonego do komunikatu raportu.
MQRO_EXPIRATION_WITH_FULL_DATA	Komunikaty raportu o utracie ważności żądania, wraz ze wszystkimi danymi aplikacji z oryginalnego komunikatu dołączonego do komunikatu raportu.

Te wartości odpowiadają opcji raportu, które można określić w polu **Report** deskryptora komunikatu.

Domyślnie właściwość nie jest ustawiona.

JMS_IBM_REPORT_NAN

Typ danych:
System.Int32

Właściwość:
Komunikat

Wysyła żądanie przesyłania komunikatów raportów o powiadomieniach dotyczących działań negatywnych.

Ustaw właściwość na MQRO_NAN, aby zażądać powiadomienia o negatywnym działaniu powiadomień o działaniu. Jeśli nie są wymagane komunikaty powiadomień o negatywnym działaniu, nie należy ustawiać tej właściwości. Domyślnie właściwość nie jest ustawiona.

Wartość MQRO_NAN odpowiada opcji raportu, która może zostać określona w polu **Report** deskryptora komunikatu.

JMS_IBM_REPORT_PAN

Typ danych:
System.Int32

Właściwość:
Komunikat

Wysyła żądanie przesyłania komunikatów raportów o powiadomieniach dotyczących działań pozytywnych.

Ustaw właściwość na MQRO_PAN, aby zażądać komunikatów raportu z powiadomieniem o pozytywnym działaniu. Jeśli nie są wymagane komunikaty z powiadomieniem o powiadomieniu o działaniu pozytywnym, nie należy ustawiać właściwości. Domyślnie właściwość nie jest ustawiona.

Wartość MQRO_PAN odpowiada opcji raportu, która może zostać określona w polu **Report** deskryptora komunikatu.

JMS_IBM_REPORT_PASS_CORREL_ID

Typ danych:
System.Int32

Właściwość:
Komunikat

Wysyła żądanie, aby identyfikator korelacji dowolnego komunikatu odpowiedzi lub raportu był taki sam jak identyfikator korelacji oryginalnego komunikatu.

Poprawne wartości właściwości są następujące:

Poprawna wartość	Znaczenie
MQRO_PASS_CORREL_ID	Wysyła żądanie, aby identyfikator korelacji dowolnego komunikatu odpowiedzi lub raportu był taki sam jak identyfikator korelacji oryginalnego komunikatu.
MQRO_COPY_MSG_ID_TO_CORREL_ID (Identyfikator CORREL_ID)	Zażądaj, aby identyfikator korelacji dowolnego komunikatu lub komunikatu odpowiedzi był taki sam, jak identyfikator komunikatu oryginalnego komunikatu.

Te wartości odpowiadają opcji raportu, które można określić w polu **Report** deskryptora komunikatu.

Domyślna wartość właściwości to MQRO_COPY_MSG_ID_TO_CORREL_ID.

JMS_IBM_REPORT_PASS_MSG_ID

Typ danych:
System.Int32

Właściwość:
Komunikat

Wysyła żądanie, aby identyfikator dowolnego komunikatu odpowiedzi lub raportu był taki sam jak identyfikator oryginalnego komunikatu.

Poprawne wartości właściwości są następujące:

Poprawna wartość	Znaczenie
MQRO_PASS_MSG_ID	Wysyła żądanie, aby identyfikator dowolnego komunikatu odpowiedzi lub raportu był taki sam jak identyfikator oryginalnego komunikatu.
MQRO_NEW_MSG_ID	Żądanie wygenerowania nowego identyfikatora komunikatu dla każdego komunikatu lub komunikatu odpowiedzi.

Te wartości odpowiadają opcji raportu, które można określić w polu Raport deskryptora komunikatu.

Wartością domyślną właściwości jest MQRO_NEW_MSG_ID.

JMS_IBM_RETAIN

Typ danych:
System.Int32

Właściwość:
Komunikat

Jeśli ta właściwość zostanie ustawiona, menedżer kolejek będzie traktował komunikat jako zachowaną publikację. Gdy subskrybent odbiera komunikaty z tematów, może otrzymywać dodatkowe komunikaty bezpośrednio po subskrybowaniu, poza komunikatami otrzymywanych w poprzednich wersjach. Te komunikaty są opcjonalnymi zachowanych publikacjami dla subskrybowanych tematów. W przypadku każdego tematu zgodnego z subskrypcją, jeśli istnieje zachowana publikacja, publikacja jest udostępniona do dostarczenia do konsumenta komunikatu subskrybującego.

Wartość RETAIN_PUBLICATION jest jedyną poprawną wartością tej właściwości. Domyślnie ta właściwość nie jest ustawiona.

Uwaga: Ta właściwość ma znaczenie tylko w domenie publikowania/subskrybowania

JMS_IBM_SYSTEM_MESSAGEID

Typ danych:
Łańcuch

Właściwość:
Komunikat

Identyfikator, który identyfikuje komunikat w jednoznaczny sposób w obrębie magistrali integracji usług. Ta właściwość jest tylko do odczytu.

Ta właściwość ma znaczenie tylko wtedy, gdy aplikacja łączy się z magistralą integracji usług.

JMSX_APPID

Typ danych:
Łańcuch

Właściwość:
Komunikat

Nazwa aplikacji, która wysłała komunikat.

Ta właściwość jest zdefiniowaną właściwością JMS o nazwie JMS o nazwie JMSXAppID. Więcej informacji na temat tej właściwości zawiera sekcja *Java Message Service Specification, wersja 1.1*.

Domyślnie właściwość nie jest ustawiona.

Ta właściwość nie jest poprawna dla połączenia w czasie rzeczywistym z brokerem.

JMSX_DELIVERY_COUNT

Typ danych:
System.Int32

Właściwość:
Komunikat

Liczba prób dostarczenia komunikatu.

Ta właściwość jest zdefiniowaną właściwością JMS o nazwie JMS o nazwie JMSXDeliveryCount. Więcej informacji na temat tej właściwości zawiera sekcja *Java Message Service Specification, wersja 1.1*.

Domyślnie właściwość nie jest ustawiona.

Ta właściwość nie jest poprawna dla połączenia w czasie rzeczywistym z brokerem.

Identyfikator JMSX_GROUPID

Typ danych:
Łańcuch

Właściwość:
Komunikat

Identyfikator grupy komunikatów, do której należy komunikat.

Ta właściwość jest zdefiniowaną właściwością JMS o nazwie JMS o nazwie JMSXGroupID. Więcej informacji na temat tej właściwości zawiera sekcja *Java Message Service Specification, wersja 1.1*.

Domyślnie właściwość nie jest ustawiona.

Ta właściwość nie jest poprawna dla połączenia w czasie rzeczywistym z brokerem.

JMSX_GROUPSEQ

Typ danych:
System.Int32

Właściwość:
Komunikat

Numer kolejny komunikatu w obrębie grupy komunikatów.

Ta właściwość jest zdefiniowaną właściwością JMS o nazwie JMS o nazwie JMSXGroupSeq. Więcej informacji na temat tej właściwości zawiera sekcja *Java Message Service Specification, wersja 1.1*.

Domyślnie właściwość nie jest ustawiona.

Ta właściwość nie jest poprawna dla połączenia w czasie rzeczywistym z brokerem.

ID_UŻYTKOWNIKA JMSX_USERID

Typ danych:
Łańcuch

Właściwość:
Komunikat

Identyfikator użytkownika powiązany z aplikacją, która wysłała komunikat.

Ta właściwość jest zdefiniowaną właściwością JMS o nazwie JMS o nazwie JMSXUserID. Więcej informacji na temat tej właściwości zawiera sekcja *Java Message Service Specification, wersja 1.1*.

Domyślnie właściwość nie jest ustawiona.

Ta właściwość nie jest poprawna dla połączenia w czasie rzeczywistym z brokerem.

XMSC_ASYNC_EXCEPTIONS,

Typ danych:
System.Int32

Właściwość:
ConnectionFactory

Obiekty mające zastosowanie:
Długa nazwa narzędzia administracyjnego JMS: ASYNCEXCEPTION

Krótka nazwa narzędzia administracyjnego JMS: AEX

Ta właściwość określa, czy aplikacja XMS ma informować interfejs ExceptionListener tylko wtedy, gdy połączenie zostało zerwane, czy też wtedy, gdy dowolny wyjątek wystąpi asynchronicznie w wywołaniu interfejsu API XMS. Ta właściwość ma zastosowanie względem wszystkich połączeń utworzonych w tej fabryce połączeń, które mają zarejestrowany interfejs ExceptionListener.

Poprawne wartości dla tej właściwości to:

XMSC_ASYNC_EXCEPTIONS_ALL
Wszystkie wyjątki wykryte asynchronicznie, poza zasięgiem wywołania synchronicznego interfejsu API, oraz wszystkie wyjątki zerwane połączenia są wysyłane do obiektu ExceptionListener.

XMSC_ASYNC_EXCEPTIONS_CONNECTIONBROKEN
Do obiektu ExceptionListenerwysyłane są tylko wyjątki wskazujące, że połączenie zerwane jest zerwane. Wszystkie inne wyjątki występujące podczas przetwarzania asynchronicznego nie są raportowane do obiektu ExceptionListeneri dlatego aplikacja nie jest powiadamiana o tych wyjątkach.

Domyślnie ta właściwość jest ustawiona na wartość XMSC_ASYNC_EXCEPTIONS_ALL.

XMSC_CLIENT_ID (Identyfikator klienta

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Obiekty mające zastosowanie:
Długa nazwa narzędzia administracyjnego JMS: CLIENTID

Krótka nazwa narzędzia administracyjnego JMS: CID

Identyfikator klienta dla połączenia.

Identyfikator klienta jest używany tylko do obsługi trwałych subskrypcji w domenie Publikowanie/subskrypcja i jest ignorowany w domenie punkt z punktem . Więcej informacji na temat ustawiania identyfikatorów klienta zawiera sekcja “Obiekty ConnectionFactories i obiekty Connection” na stronie 22.

Ta właściwość nie ma znaczenia dla połączenia w czasie rzeczywistym z brokerem.

XMSC_CONNECTION_TYPE

Typ danych:
System.Int32

Właściwość:
ConnectionFactory

Typ serwera przesyłania komunikatów, z którym łączy się aplikacja.

Poprawne wartości właściwości są następujące:

Poprawna wartość	Znaczenie
XMSC_CT_RTT	Połączenie w czasie rzeczywistym z brokerem.
XMSC_CT_WMQ	Połączenie z menedżerem kolejek produktu WebSphere MQ .
XMSC_CT_WPM	Połączenie z magistralą integracji usług WebSphere .

Domyślnie właściwość nie jest ustawiona.

XMSC_DELIVERY_MODE

Typ danych:
System.Int32

Właściwość:
Miejsce docelowe

Nazwa używana w identyfikatorze URI:

`persistence` (dla miejsca docelowego produktu WebSphere MQ)
`deliveryMode` (w przypadku domyślnego miejsca docelowego dostawcy przesyłania komunikatów produktu WebSphere)

Obiekty mające zastosowanie:
Długa nazwa narzędzia administracyjnego JMS: PERSISTENCE

Krótka nazwa narzędzia administracyjnego JMS: PER

Tryb dostarczania komunikatów wysyłanych do miejsca docelowego.

Poprawne wartości właściwości są następujące:

Poprawna wartość	Znaczenie
XMSC_DELIVERY_NOT_PERSISTENT	Komunikat wysłany do miejsca docelowego jest nietrwały. Domyślny tryb dostarczania dla producenta komunikatów lub dowolny tryb dostarczania określony w wywołaniu Send jest ignorowany. Jeśli miejscem docelowym jest kolejka WebSphere MQ , wartość atrybutu kolejki *DefPersistence* jest również ignorowana.
XMSC_DELIVERY_PERSISTENT	Komunikat wysłany do miejsca docelowego jest trwały. Domyślny tryb dostarczania dla producenta komunikatów lub dowolny tryb dostarczania określony w wywołaniu Send jest ignorowany. Jeśli miejscem docelowym jest kolejka WebSphere MQ , wartość atrybutu kolejki *DefPersistence* jest również ignorowana.
XMSC_DELIVERY_AS_APP	Komunikat wysłany do miejsca docelowego ma tryb dostarczania określony w wywołaniu Wyślij. Jeśli wywołanie Wyślij nie określa trybu dostarczania, zostanie użyty domyślny tryb dostarczania producenta komunikatów. Jeśli miejscem docelowym jest kolejka WebSphere MQ , wartość atrybutu kolejki *DefPersistence* jest ignorowana.
XMSC_DELIVERY_AS_DEST	Jeśli miejscem docelowym jest kolejka WebSphere MQ , komunikat umieszczany w kolejce ma tryb dostarczania określony przez wartość atrybutu kolejki *DefPersistence*. Domyślny tryb dostarczania dla producenta komunikatów lub dowolny tryb dostarczania określony w wywołaniu Send jest ignorowany. Jeśli miejsce docelowe nie jest kolejką produktu WebSphere MQ , oznacza to, że jest to samo znaczenie, jak w przypadku produktu XMSC_DELIVERY_AS_APP.

Wartością domyślną jest XMSC_DELIVERY_AS_APP.

XMSC_IC_PROVIDER_URL

Typ danych:
Łańcuch

Właściwość:
InitialContext

Umożliwia wskazanie katalogu nazw JNDI. Dzięki niej usługa nazw COS nie musi znajdować się na tym samym serwerze co usługa WWW.

XMSC_IC_SECURITY_AUTHENTACJA

Typ danych:
Łańcuch

Właściwość:
InitialContext

W oparciu o interfejs kontekstu Java SECURITY_AUTHENTICATION. Ta właściwość ma zastosowanie tylko do kontekstu nazewnictwa COS.

XMSC_IC_SECURITY_CREDENTIALS

Typ danych:
Łańcuch

Właściwość:
InitialContext

W oparciu o interfejs kontekstu Java SECURITY_CREDENTIALS. Ta właściwość ma zastosowanie tylko do kontekstu nazewnictwa COS.

XMSC_IC_SECURITY_PRINCIPAL

Typ danych:
Łańcuch

Właściwość:
InitialContext

W oparciu o interfejs kontekstu Java SECURITY_PRINCIPAL. Ta właściwość ma zastosowanie tylko do kontekstu nazewnictwa COS.

XMSC_IC_SECURITY_PROTOCOL

Typ danych:
Łańcuch

Właściwość:
InitialContext

Na podstawie interfejsu kontekstu Java SECURITY_PROTOCOL Ta właściwość ma zastosowanie tylko do kontekstu nazewnictwa COS.

Adres URL XMSC_IC_URL

Typ danych:
Łańcuch

Właściwość:
InitialContext

W przypadku kontekstów LDAP i FileSystem: adres repozytorium zawierającego obiekty administrowane.

W przypadku kontekstów nazw COS: adres usługi WWW, która wyszukuje obiekty w katalogu.

XMSC_IS_SUBSCRIPTION_MULTICAST

Typ danych:
System.Boolean

Właściwość:
MessageConsumer

Wskazuje, czy komunikaty są dostarczane do konsumenta komunikatów przy użyciu produktu WebSphere MQ Multicast Transport. Ta właściwość jest tylko do odczytu.

Wartość właściwości jest prawdziwa, jeśli komunikaty są dostarczane do konsumenta komunikatów przy użyciu produktu WebSphere MQ Multicast Transport. W przeciwnym razie wartość będzie mieć wartość false.

Ta właściwość ma znaczenie tylko w przypadku połączenia w czasie rzeczywistym z brokerem.

XMSC_IS_SUBSCRIPTION_RELIABLE_MULTICAST

Typ danych:
System.Boolean

Właściwość:
MessageConsumer

Wskazuje, czy komunikaty są dostarczane do konsumenta komunikatów przy użyciu produktu WebSphere MQ Multicast Transport z niezawodną jakością usługi. Ta właściwość jest tylko do odczytu.

Wartość tej właściwości jest prawdziwa, jeśli komunikaty są dostarczane do konsumenta komunikatów przy użyciu produktu WebSphere MQ Multicast Transport z niezawodną jakością usługi. W przeciwnym razie wartość będzie mieć wartość false.

Ta właściwość ma znaczenie tylko w przypadku połączenia w czasie rzeczywistym z brokerem.

XMSC_JMS_MAJOR_VERSION

Typ danych:
System.Int32

Właściwość:
Dane ConnectionMeta

Numer wersji głównej specyfikacji JMS , na której oparty jest produkt XMS . Ta właściwość jest tylko do odczytu.

XMSC_JMS_MINOR_VERSION

Typ danych:
System.Int32

Właściwość:
Dane ConnectionMeta

Drugorzędny numer wersji specyfikacji JMS , na której oparty jest produkt XMS . Ta właściwość jest tylko do odczytu.

XMSC_JMS_VERSION

Typ danych:
Łańcuch

Właściwość:
Dane ConnectionMeta

Identyfikator wersji specyfikacji JMS , na której oparty jest produkt XMS . Ta właściwość jest tylko do odczytu.

XMSC_MAJOR_VERSION

Typ danych:
System.Int32

Właściwość:
Dane ConnectionMeta

Numer wersji klienta XMS . Ta właściwość jest tylko do odczytu.

XMSC_MINOR_VERSION

Typ danych:
System.Int32

Właściwość:
Dane ConnectionMeta

Numer wydania klienta XMS . Ta właściwość jest tylko do odczytu.

HASŁO XMSC_PASSWORD

Typ danych:
Tablica bajtów

Właściwość:
ConnectionFactory

Hasło, które może być używane do uwierzytelniania aplikacji na etapie podejmowania próby nawiązania połączenia z serwerem przesyłania komunikatów. Hasło jest używane z właściwością XMSC_USERID .

Domyślnie właściwość nie jest ustawiona.

Multi Jeśli połączenie z produktem WebSphere MQ jest nawiązane z produktem Multiplatforms, a właściwość XMSC_USERID fabryki połączeń jest ustawiona, musi ona być zgodna z **userid** zalogowanego użytkownika. Jeśli te właściwości nie zostaną ustawione, menedżer kolejek domyślnie będzie używać programu **userid** zalogowanego użytkownika. Jeśli wymagane jest dalsze uwierzytelnianie na poziomie połączenia dla poszczególnych użytkowników, można napisać wyjście uwierzytelniania klienta skonfigurowane w produkcie WebSphere MQ. Więcej informacji na temat tworzenia wyjścia uwierzytelniania klienta można uzyskać w temacie Uwierzytelnianie w podręczniku WebSphere MQ Clients.

Aby uwierzytelnić użytkownika podczas nawiązywania połączenia z produktem WebSphere MQ w systemie z/OS , należy użyć wyjścia zabezpieczeń.

XMSC_PRIORITY

Typ danych:
System.Int32

Właściwość:
Miejsce docelowe

Nazwa używana w identyfikatorze URI:
priorytet

Priorytet komunikatów wysyłanych do miejsca docelowego.

Poprawne wartości właściwości są następujące:

Poprawna wartość	Znaczenie
Liczba całkowita z zakresu 0, najniższy priorytet, do 9, najwyższy priorytet	Komunikat wysłany do miejsca docelowego ma określony priorytet. Domyślny priorytet producenta komunikatów lub dowolny priorytet określony w wywołaniu Send jest ignorowany. Jeśli miejscem docelowym jest kolejka WebSphere MQ , wartość atrybutu kolejki **DefPriority** jest również ignorowana.

Poprawna wartość	Znaczenie
XMSC_PRIORITY_AS_APP	Komunikat wysłany do miejsca docelowego ma priorytet określony w wywołaniu Wyślij. Jeśli wywołanie Wyślij nie określa priorytetu, zamiast niego używany jest domyślny priorytet producenta komunikatów. Jeśli miejscem docelowym jest kolejka WebSphere MQ , wartość atrybutu kolejki **DefPriority** jest ignorowana.
XMSC_PRIORITY_AS_DEST	Jeśli miejscem docelowym jest kolejka WebSphere MQ , komunikat umieszczany w kolejce ma priorytet określony przez wartość atrybutu kolejki **DefPriority**. Domyślny priorytet producenta komunikatów lub dowolny priorytet określony w wywołaniu Send jest ignorowany. Jeśli miejsce docelowe nie jest kolejką produktu WebSphere MQ , oznacza to, że jest to samo znaczenie, jak w przypadku XMSC_PRIORITY_AS_APP.

Wartością domyślną jest XMSC_PRIORITY_AS_APP.

Produkty WebSphere MQ Real-Time Transport i WebSphere MQ Multicast Transport nie podejmują żadnych działań w oparciu o priorytet komunikatu.

NAZWA DOSTAWCY XMSC_DOSTAWCY

Typ danych:
Łańcuch

Właściwość:
Dane ConnectionMeta

Dostawca klienta XMS . Ta właściwość jest tylko do odczytu.

XMSC_RTT_BROKER_PING_INTERVAL

Typ danych:
System.Int32

Właściwość:
ConnectionFactory

Odstęp czasu (w milisekundach), po upływie którego klient XMS .NET sprawdza połączenie z serwerem przesyłania komunikatów w czasie rzeczywistym w celu wykrycia dowolnego działania. Jeśli żadne działanie nie zostanie wykryte, klient inicjuje komendę ping; połączenie jest zamykane, jeśli nie zostanie wykryta żadna odpowiedź na komendę ping.

Wartością domyślną właściwości jest 30000.

XMSC_RTT_CONNECTION_PROTOCOL

Typ danych:
System.Int32

Właściwość:
ConnectionFactory

Protokół komunikacyjny używany do nawiązywania połączenia z brokerem w czasie rzeczywistym.

Wartością tej właściwości musi być XMSC_RTT_CP_TCP, co oznacza połączenie w czasie rzeczywistym z brokerem za pośrednictwem protokołu TCP/IP. Wartością domyślną jest XMSC_RTT_CP_TCP.

XMSC_RTT_HOST_NAME

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Miejsce działania brokera: nazwa hosta lub adres IP systemu.

Ta właściwość jest używana z właściwością XMSC_RTT_PORT do identyfikowania brokera.

Domyślnie właściwość nie jest ustawiona.

XMSC_RTT_LOCAL_ADDRESS

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Nazwa hosta lub adres IP interfejsu sieci lokalnej na potrzeby nawiązania połączenia z brokerem w czasie rzeczywistym.

Ta właściwość jest przydatna tylko wtedy, gdy system, na którym działa aplikacja, ma dwa lub więcej interfejsów sieciowych i musi być w stanie określić, który interfejs musi być używany do połączenia w czasie rzeczywistym. Jeśli system ma tylko jeden interfejs sieciowy, tylko ten interfejs może być używany. Jeśli w systemie istnieją dwa lub więcej interfejsów sieciowych, a właściwość nie jest ustawiona, interfejs jest wybierany losowo.

Domyślnie właściwość nie jest ustawiona.

XMSC_RTT_MULTICAST

Typ danych:
System.Int32

Właściwość:
ConnectionFactory i miejsce docelowe

Nazwa używana w identyfikatorze URI:
mulicast

Ustawienie rozsyłania grupowego dla fabryki połączeń lub miejsca docelowego. Ta właściwość może mieć tylko miejsce docelowe, które jest tematem.

Aplikacja korzysta z tej właściwości w celu włączenia rozsyłania grupowego w czasie rzeczywistym połączenia z brokerem oraz, jeśli funkcja rozsyłania grupowego jest włączona, do określania precyzyjnego sposobu, w jaki rozsyłanie grupowe jest używane do dostarczania komunikatów z brokera do konsumenta komunikatów. Właściwość nie ma wpływu na to, w jaki sposób producent komunikatów wysyła komunikaty do brokera.

Poprawne wartości właściwości są następujące:

Poprawna wartość	Znaczenie
XMSC_RTT_MULTICAST_DISABLED	Komunikaty nie są dostarczane do konsumenta komunikatów przy użyciu produktu WebSphere MQ Multicast Transport. Ta wartość jest wartością domyślną dla obiektu ConnectionFactory .

Poprawna wartość	Znaczenie
XMSC_RTT_MULTICAST_ASCF	Komunikaty są dostarczane do konsumenta komunikatów zgodnie z ustawieniem rozsyłania grupowego dla fabryki połączeń powiązanej z konsumentem komunikatów. Ustawienie rozsyłania grupowego dla fabryki połączeń jest oznaczane w momencie tworzenia połączenia. Ta wartość jest poprawna tylko dla obiektu docelowego i jest to wartość domyślna dla obiektu docelowego.
XMSC_RTT_MULTICAST_ENABLED	Jeśli temat został skonfigurowany do rozsyłania grupowego w brokerze, komunikaty są dostarczane do konsumenta komunikatów przy użyciu produktu WebSphere MQ Multicast Transport. Jeśli temat został skonfigurowany pod kątem niezawodnego rozsyłania grupowego, używana jest niezawodna jakość usługi.
XMSC_RTT_MULTICAST_NIEZAWODNE	Jeśli temat został skonfigurowany pod kątem niezawodnego rozsyłania grupowego w brokerze, komunikaty są dostarczane do konsumenta komunikatów przy użyciu produktu WebSphere MQ Multicast Transport z niezawodną jakością usługi. Jeśli temat nie został skonfigurowany pod kątem niezawodnego rozsyłania grupowego, nie można utworzyć konsumenta komunikatów dla tego tematu.
XMSC_RTT_MULTICAST_NOT_RELIABLE	Jeśli temat został skonfigurowany do rozsyłania grupowego w brokerze, komunikaty są dostarczane do konsumenta komunikatów przy użyciu produktu WebSphere MQ Multicast Transport. Niezawodna jakość usługi nie jest używana, nawet jeśli temat został skonfigurowany pod kątem niezawodnego rozsyłania grupowego.

PORT XMSC_RTT_PORT

Typ danych:
System.Int32

Właściwość:
ConnectionFactory

Numer portu, na którym broker nasłuchuje żądań przychodzących. W brokerze należy skonfigurować węzeł przetwarzania komunikatów typu Real-timeInput lub Real-timeOptimizedFlow, aby nasłuchiwać na tym porcie.

Ta właściwość jest używana z właściwością XMSC_RTT_HOST_NAME do identyfikowania brokera.

Wartością domyślną tej właściwości jest XMSC_RTT_DEFAULT_PORT lub 1506.

XMSC_TIME_TO_LIVE

Typ danych:
System.Int32

Właściwość:
Miejsce docelowe

Nazwa używana w identyfikatorze URI:

utrata ważności (dla miejsca docelowego WebSphere MQ)
timeToLive (dla domyślnego miejsca docelowego dostawcy przesyłania komunikatów produktu WebSphere)

Czas życia komunikatów wysyłanych do miejsca docelowego.

Poprawne wartości właściwości są następujące:

Poprawna wartość	Znaczenie
0	Wiadomość wysłana do miejsca docelowego nigdy nie traci ważności.
Dodatnia liczba całkowita	Komunikat wysłany do miejsca docelowego ma określony czas życia (w milisekundach). Domyślny czas życia producenta komunikatów lub dowolny czas życia określony w wywołaniu Wyślij jest ignorowany.
XMSC_TIME_TO_LIVE_AS_APP	Komunikat wysłany do miejsca docelowego ma czas życia określony w wywołaniu Wyślij. Jeśli wywołanie Wyślij nie określa czasu życia, zamiast tego używany jest domyślny czas życia producenta komunikatów.

Wartością domyślną jest XMSC_TIME_TO_LIVE_AS_APP.

ID_UŻYTKOWNIKA XMSC_USERID

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Identyfikator użytkownika, który może być używany do uwierzytelniania aplikacji na etapie podejmowania próby nawiązania połączenia z serwerem przesyłania komunikatów. Identyfikator użytkownika jest używany z właściwością XMSC_PASSWORD .

Domyślnie właściwość nie jest ustawiona.

Multi Jeśli połączenie z produktem IBM MQ for Multiplatformsjest nawiązane, a właściwość XMSC_USERID fabryki połączeń jest ustawiona, musi ona być zgodna z **userid** zalogowanego użytkownika. Jeśli te właściwości nie zostaną ustawione, menedżer kolejek domyślnie będzie używać programu **userid** zalogowanego użytkownika. Jeśli wymagane jest dalsze uwierzytelnianie na poziomie połączenia dla poszczególnych użytkowników, można napisać wyjście uwierzytelniania klienta, które jest skonfigurowane w produkcie IBM MQ.

z/OS Aby uwierzytelnić użytkownika podczas nawiązywania połączenia z produktem IBM MQ for z/OS , należy użyć wyjścia zabezpieczeń.

XMSC_VERSION

Typ danych:
Łańcuch

Właściwość:
Dane ConnectionMeta

Identyfikator wersji klienta XMS . Ta właściwość jest tylko do odczytu.

XMSC_WMQ_BROKER_CONTROLQ

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Nazwa kolejki sterującej używanej przez broker.

Wartością domyślną tej właściwości jest SYSTEM.BROKER.CONTROL.QUEUE.

Ta właściwość jest odpowiednia tylko w domenie Publikowanie/subskrypcja .

XMSC_WMQ_BROKER_PUBQ

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Nazwa kolejki monitorowanej przez broker, do której aplikacje wysyłają publikowane komunikaty.

Wartością domyślną tej właściwości jest SYSTEM.BROKER.DEFAULT.STREAM.

Ta właściwość jest odpowiednia tylko w domenie Publikowanie/subskrypcja .

XMSC_WMQ_BROKER_QMGR

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Nazwa menedżera kolejek, z którym broker nawiązał połączenie.

Domyślnie właściwość nie jest ustawiona.

Ta właściwość jest odpowiednia tylko w domenie Publikowanie/subskrypcja .

XMSC_WMQ_BROKER_SUBQ

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Nazwa kolejki subskrybenta dla konsumenta nietrwałych komunikatów.

Nazwa kolejki subskrybenta musi rozpoczynać się od następujących znaków:

SYSTEM.JMS.ND.

Jeśli chcesz, aby wszystkie nietrwałe odbiorcy komunikatów współużytkował kolejkę subskrybenta, podaj pełną nazwę kolejki współużytkowanej. Kolejka o podanej nazwie musi istnieć, zanim aplikacja będzie mogła utworzyć nietrwały konsument komunikatów.

Jeśli każdy konsument komunikatów nietrwałych ma pobierać komunikaty z własnej, wyłącznej kolejki subskrybenta, należy podać nazwę kolejki, która kończy się gwiazdką (*). Następnie, gdy aplikacja tworzy nietrwały konsument komunikatów, klient XMS tworzy kolejkę dynamiczną do wyłącznego użytku przez konsumenta komunikatów. Klient XMS korzysta z wartości właściwości w celu ustawienia zawartości pola **DynamicQName** w deskryptorze obiektu, który jest używany do tworzenia kolejki dynamicznej.

Wartością domyślną tej właściwości jest SYSTEM.JMS.ND.SUBSCRIBER.QUEUE, co oznacza, że produkt XMS domyślnie korzysta z metody kolejki współużytkowanej.

Ta właściwość jest odpowiednia tylko w domenie Publikowanie/subskrypcja .

XMSC_WMQ_BROKER_VERSION

Typ danych:
System.Int32

Właściwość:
ConnectionFactory i miejsce docelowe

Nazwa używana w identyfikatorze URI:
brokerVersion

Typ brokera używany na potrzeby połączenia lub miejsca docelowego. Ta właściwość może mieć tylko miejsce docelowe, które jest tematem.

Poprawne wartości właściwości są następujące:

Poprawna wartość	Znaczenie
XMSC_WMQ_BROKER_V1	Aplikacja korzysta z brokera WebSphere MQ Publikowanie/subskrypcja . Aplikacja może również użyć tej wartości w przypadku migracji z produktu WebSphere MQ Publikowanie/subskrypcja do wersji WebSphere Message Broker , ale nie została ona zmieniona.
XMSC_WMQ_BROKER_V2	Aplikacja korzysta z brokera produktu IBM Integration Bus.
XMSC_WMQ_BROKER_UNSPECIFIED	Po przeprowadzeniu migracji brokera należy ustawić tę właściwość w taki sposób, aby nagłówki RFH2 nie były już używane. Po migracji właściwość ta nie jest już istotna.

Wartością domyślną fabryki connectionfactory jest XMSC_WMQ_BROKER_UNSPECIFIED, ale domyślnie właściwość nie jest ustawiona dla miejsca docelowego. Ustawienie właściwości dla miejsca docelowego przesłania dowolną wartość określoną przez właściwość fabryki połączeń.

XMSC_WMQ_CCDTURL

Typ danych:
System.String

Właściwość:
ConnectionFactory

Obiekty mające zastosowanie:
Długa nazwa narzędzia administracyjnego JMS: CCDTURL

Krótka nazwa narzędzia administracyjnego JMS: CCDT

Adres URL identyfikujący nazwę i położenie pliku zawierającego tabelę definicji kanałów klienta oraz określający sposób dostępu do pliku.

Domyślnie ta właściwość nie jest ustawiona.

CCSID XMSC_WMQ_CCSID

Typ danych:
System.Int32

Właściwość:
Miejsce docelowe

Nazwa używana w identyfikatorze URI:
CCSID

Identyfikator (identyfikator CCSID) kodowanego zestawu znaków lub strony kodowej, w którym łańcuchy danych znakowych w treści komunikatu są wyświetlane, gdy klient XMS przekazuje komunikat do miejsca docelowego. Jeśli zostanie ustawiony dla pojedynczego komunikatu, właściwość

JMS_IBM_CHARACTER_SET przesłania identyfikator CCSID określony dla miejsca docelowego przez tę właściwość.

Wartość domyślna właściwości to 1208.

Ta właściwość ma znaczenie tylko dla komunikatów wysyłanych do miejsca docelowego, a nie do komunikatów odebranych z miejsca docelowego.

XMSC_WMQ_CHANNEL

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Obiekty mające zastosowanie:
Długa nazwa narzędzia administracyjnego JMS: CHANNEL

Krótka nazwa narzędzia administracyjnego JMS: CHAN

Nazwa kanału używanego do nawiązywania połączenia.

Domyślnie właściwość nie jest ustawiona.

Ta właściwość ma znaczenie tylko wtedy, gdy aplikacja łączy się z menedżerem kolejek w trybie klienta.

XMSC_WMQ_CLIENT_RECONNECT_OPTIONS

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Obiekty mające zastosowanie:
Długa nazwa narzędzia administracyjnego JMS: CLIENTRECONNECTOPTIONS

Krótka nazwa narzędzia administracyjnego JMS: CROPT

Ta właściwość określa opcje ponownego połączenia klienta dla nowych połączeń utworzonych przez tę fabrykę. Znajduje się on w XMSC i jest jednym z:

- WMQ_CLIENT_RECONNECT_AS_DEF (wartość domyślna). Użyj wartości określonej w pliku `mqclient.ini` . Ustaw tę wartość, używając właściwości **DefRecon** w sekcji Kanały. Można ją ustawić na jedną z następujących opcji:
  1. Tak. Zachowuje się jak opcja WMQ_CLIENT_RECONNECT
  2. Nie. Domyślne. Nie określa żadnych opcji ponownego połączenia
  3. QMGR. Zachowuje się jak opcja WMQ_CLIENT_RECONNECT_Q_MGR
  4. Wyłączone. Zachowuje się jak opcja WMQ_CLIENT_RECONNECT_DISABLED
- WMQ_CLIENT_RECONNECT. Ponownie nawiąże połączenie z dowolnym menedżerem kolejek określonym na liście nazw połączeń.
- WMQ_CLIENT_RECONNECT_Q_MGR. Ponownie łączy się z tym samym menedżerem kolejek, z którym jest połączony. Zwraca ona `MQRC_RECONNECT_QMID_MISMATCH` , jeśli menedżer kolejek, z którym próbuje się połączyć (określony na liście nazw połączeń), ma inny identyfikator QMID do menedżera kolejek, do którego pierwotnie nawiązano połączenie.
- WMQ_CLIENT_RECONNECT_DISABLED. Ponowne połączenie jest wyłączone.

XMSC_WMQ_CLIENT_RECONNECT_TIMEOUT

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Obiekty mające zastosowanie:
Długa nazwa narzędzia administracyjnego JMS: CLIENTRECONNECTTIMEOUT

Krótka nazwa narzędzia administracyjnego JMS: CRT

Właściwość XMSC_WMQ_CLIENT_RECONNECT_TIMEOUT jest poprawna tylko dla zarządzanego klienta XMS .NET .

Ta właściwość określa czas (w sekundach), przez który połączenie klienta próbuje ponownie nawiązać połączenie.

Po próbie ponownego nawiązania połączenia przez ten czas klient nie powiedzie się i zostanie wykonana operacja MQRC_RECONNECT_FAILED. Domyślnym ustawieniem dla tej właściwości jest XMSC.WMQ_CLIENT_RECONNECT_TIMEOUT_DEFAULT.

Wartością domyślną tej właściwości jest 1800.

XMSC_WMQ_CONNECTION_MODE

Typ danych:
System.Int32

Właściwość:
ConnectionFactory

Tryb, przy użyciu którego aplikacja nawiązuje połączenie z menedżerem kolejek.

Poprawne wartości właściwości są następujące:

Poprawna wartość	Znaczenie
XMSC_WMQ_CM_BINDINGS	Połączenie z menedżerem kolejek w trybie powiązań w celu uzyskania optymalnej wydajności. Ta wartość jest wartością domyślną dla C/C + +.
KLIENT XMSC_WMQ_CM_CLIENT	Połączenie z menedżerem kolejek w trybie klienta w celu zapewnienia w pełni zarządzanego stosu. Ta wartość jest wartością domyślną dla środowiska .NET.
XMSC_WMQ_CM_CLIENT_UNMANAGED (tylko dla środowiska .NET)	Połączenie z menedżerem kolejek, które wymusza niezarządzany stos klienta.

Pojęcia pokrewne

Operacje zarządzane i niezarządzane w programie .NET
Kod zarządzany jest wykonywany wyłącznie w środowisku wykonawczym języka wspólnego produktu .NET i jest w pełni zależny od usług udostępnianych przez to środowisko wykonawcze. Aplikacja jest klasowana jako niezarządzana, jeśli dowolna część aplikacji działa lub wywołuje usługi poza środowiskiem wykonawczym wspólnego języka produktu .NET .

XMSC_WMQ_CONNECTION_NAME_LIST

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Obiekty mające zastosowanie:
Długa nazwa narzędzia administracyjnego JMS: CONNECTIONNAMELIST

Krótka nazwa narzędzia administracyjnego JMS: CNLIST

Ta właściwość określa hosty, do których klient próbuje ponownie nawiązać połączenie po zerowaniu połączenia.

Lista nazw połączeń jest rozdzielaną przecinkami listą par host/ip portów. Domyślnym ustawieniem dla tej właściwości jest WMQ_CONNECTION_NAME_LIST_DEFAULT.

Na przykład:`127.0.0.1 (1414),host2.example.com(1400)`

Domyślne ustawienie tej właściwości to `localhost (1414)`.

XMSC_WMQ_DUR_SUBQ

Typ danych:
Łańcuch

Właściwość:
Miejsce docelowe

Nazwa kolejki trwałego subskrybenta, który odbiera komunikaty z miejsca docelowego. Ta właściwość może mieć tylko miejsce docelowe, które jest tematem.

Nazwa kolejki subskrybenta musi rozpoczynać się od następujących znaków:

SYSTEM.JMS.D.

Jeśli chcesz, aby wszystkie trwałe subskrybenty współużytkował kolejkę subskrybenta, podaj pełną nazwę kolejki współużytkowanej. Kolejka o podanej nazwie musi istnieć, zanim aplikacja może utworzyć trwałą subskrybent.

Jeśli każdy z trwałych subskrybentów ma pobierać komunikaty z własnej, wyłącznej kolejki subskrybenta, należy określić nazwę kolejki, która kończy się gwiazdką (*). Następnie, gdy aplikacja tworzy trwałą subskrybent, klient XMS tworzy kolejkę dynamiczną do wyłącznego użytku przez trwałą subskrybent. Klient XMS korzysta z wartości właściwości w celu ustawienia zawartości pola **DynamicQName** w deskryptorze obiektu, który jest używany do tworzenia kolejki dynamicznej.

Wartością domyślną tej właściwości jest SYSTEM.JMS.D.SUBSCRIBER.QUEUE, co oznacza, że produkt XMS domyślnie korzysta z metody kolejki współużytkowanej.

Ta właściwość jest odpowiednia tylko w domenie Publikowanie/subskrypcja .

XMSC_WMQ_ENCODING

Typ danych:
System.Int32

Właściwość:
Miejsce docelowe

W jaki sposób dane liczbowe w treści komunikatu są reprezentowane, gdy klient XMS przekazuje komunikat do miejsca docelowego. Jeśli zostanie ustawiony dla pojedynczego komunikatu, właściwość JMS_IBM_ENCODING przesłania kodowanie określone dla miejsca docelowego przez tę właściwość. Ta właściwość określa reprezentację binarnych liczb całkowitych, upakowanych liczb całkowitych dziesiętnych i liczb zmiennopozycyjnych.

Poprawne wartości właściwości są takie same, jak wartości, które można określić w polu **Encoding** deskryptora komunikatu.

Do ustawienia właściwości aplikacja może używać następujących stałych nazwanych:

Stała nazwana	Znaczenie
MQENC_INTEGER_NORMAL	Normalne kodowanie liczb całkowitych
MQENC_INTEGER_REVERSED	Odwrócone kodowanie liczb całkowitych
MQENC_DECIMAL_NORMAL	Normalne upakowane kodowanie dziesiętne
MQENC_DECIMAL_REVERSED	Odwrócone spakowane kodowanie dziesiętne
MQENC_FLOAT_IEEE_NORMAL	Normalne kodowanie zmiennopozycyjne IEEE
MQENC_FLOAT_IEEE_REVERSED	Odwrócone kodowanie zmiennopozycyjne IEEE

Stała nazwana	Znaczenie
MQENC_FLOAT_S390	Kodowanie zmiennopozycyjne architektury z/OS
MQENC_NATIVE	Kodowanie rodzimego komputera

Aby utworzyć wartość dla właściwości, aplikacja może dodać trzy następujące stałe w następujący sposób:

- Stała, której nazwa rozpoczyna się od wywołania MQENC_INTEGER, w celu określenia reprezentacji binarnych liczb całkowitych
- Stała, której nazwa rozpoczyna się od MQENC_DECIMAL, w celu określenia reprezentacji upakowanych liczb całkowitych.
- Stała, której nazwa rozpoczyna się od MQENC_FLOAT, w celu określenia reprezentacji liczb zmiennopozycyjnych.

Alternatywnie aplikacja może ustawić właściwość na wartość MQENC_NATIVE, której wartość jest zależna od środowiska.

Wartością domyślną właściwości jest MQENC_NATIVE.

Ta właściwość ma znaczenie tylko dla komunikatów wysyłanych do miejsca docelowego, a nie do komunikatów odebranych z miejsca docelowego.

XMSC_WMQ_FAIL_IF_QUIESCE

Typ danych:
System.Int32

Właściwość:
ConnectionFactory i miejsce docelowe

Nazwa używana w identyfikatorze URI:
FAILIFQUIESCE

Obiekty mające zastosowanie:
Długa nazwa narzędzia administracyjnego JMS: FAILIFQUIESCE

Krótka nazwa narzędzia administracyjnego JMS: FIQ

Określa, czy wywołania pewnych metod nie powiodą się, jeśli menedżer kolejek, z którym aplikacja nawiązała połączenie, będzie w stanie wyciszania.

Poprawne wartości właściwości są następujące:

Poprawna wartość	Znaczenie
XMSC_WMQ_FIQ_YES	Wywołania niektórych metod nie powiodą się, jeśli menedżer kolejek znajduje się w stanie wygaszania. Gdy aplikacja wykryje, że menedżer kolejek jest wygaszany, aplikacja może zakończyć swoje natychmiastowe zadanie i zamknąć połączenie, co pozwoli menedżerowi kolejek zatrzymać.
XMSC_WMQ_FIQ_NO	Wywołania metody nie powiodą się, ponieważ menedżer kolejek jest w stanie wygaszania. Jeśli ta wartość zostanie określona, aplikacja nie będzie mogła wykryć, że menedżer kolejek jest wygaszany. Aplikacja może kontynuować wykonywanie operacji względem menedżera kolejek i w związku z tym zapobiec zatrzymaniu menedżera kolejek.

Wartością domyślną dla fabryki połączeń jest XMSC_WMQ_FIQ_YES, ale domyślnie właściwość ta nie jest ustawiona dla miejsca docelowego. Ustawienie właściwości dla miejsca docelowego przesłania dowolną wartość określoną przez właściwość fabryki połączeń.

XMSC_WMQ_MESSAGE_BODY

Typ danych:
System.Int32

Właściwość:
Miejsce docelowe

Ta właściwość określa, czy aplikacja XMS przetwarza obiekt MQRFH2 komunikatu produktu IBM WebSphere MQ jako część ładunku komunikatu (czyli jako część treści komunikatu).

Uwaga: Podczas wysyłania komunikatów do miejsca docelowego, właściwość XMSC_WMQ_MESSAGE_BODY zastępuje istniejące właściwości miejsca docelowego XMS XMSC_WMQ_TARGET_CLIENT.

Poprawne wartości dla tej właściwości to:

XMSC_WMQ_MESSAGE_BODY_JMS
Odbieranie: przychodzący typ i treść komunikatu XMS są określane przez treść komunikatu MQRFH2 (jeśli istnieje) lub MQMD (jeśli nie ma wartości MQRFH2) w odebranym komunikacie IBM WebSphere MQ .

Wyślij: Treść wychodzącego komunikatu XMS zawiera wstępnie wygenerowany nagłówek MQRFH2 i automatycznie wygenerowany nagłówek na podstawie właściwości komunikatu XMS i pól nagłówka.

XMSC_WMQ_MESSAGE_BODY_MQ
Odbieranie: przychodzący typ komunikatu XMS ma zawsze wartość ByteMessage, niezależnie od zawartości odebranego komunikatu IBM WebSphere MQ lub pola formatu odebranego deskryptora MQMD. Treść komunikatu XMS to niezmienione dane komunikatu zwracane przez bazowe wywołanie API dostawcy przesyłania komunikatów. Zestaw znaków i kodowanie danych w treści komunikatu jest określane przez pola CodedCharSetId i Kodowanie deskryptora MQMD. Format danych w treści komunikatu jest określany na podstawie pola Format deskryptora MQMD.

Wyślij: wychodzący treść komunikatu XMS zawiera ładunek aplikacji jako-is; i żaden automatycznie wygenerowany nagłówek IBM WebSphere MQ nie jest dodawany do treści.

XMSC_WMQ_MESSAGE_BODY_UNSPECIFIED
Odbieranie: Klient XMS określa odpowiednią wartość dla tej właściwości. W przypadku ścieżki odbierania ta wartość jest wartością właściwości WMQ_MESSAGE_BODY_JMS.

Wyślij: Klient XMS określa odpowiednią wartość dla tej właściwości. W przypadku ścieżki wysyłania ta wartość jest wartością właściwości XMSC_WMQ_TARGET_CLIENT.

Domyślnie ta właściwość jest ustawiona na wartość XMSC_WMQ_MESSAGE_BODY_UNSPECIFIED.

XMSC_WMQ_MQMD_MESSAGE_CONTEXT

Typ danych:
System.Int32

Właściwość:
Miejsce docelowe

Określa poziom kontekstu komunikatu, który ma być ustawiany przez aplikację XMS . Aby ta właściwość miała zastosowanie, aplikacja musi być uruchamiana z odpowiednimi uprawnieniami dotyczącymi kontekstu.

Poprawne wartości dla tej właściwości to:

XMSC_WMQ_MDCTX_DEFAULT
W przypadku komunikatów wychodzących wywołanie funkcji API MQOPEN i struktura MQPMO nie określają jawnych opcji kontekstu komunikatu.

XMSC_WMQ_MDCTX_SET_IDENTITY_CONTEXT
Wywołanie funkcji API MQOPEN określa opcję kontekstu komunikatu MQOO_SET_IDENTITY_CONTEXT, a struktura MQPMO określa wartość MQPMO_SET_IDENTITY_CONTEXT.

XMSC_WMQ_MDCTX_SET_ALL_CONTEXT
Wywołanie funkcji API MQOPEN określa opcję kontekstu komunikatu MQOO_SET_ALL_CONTEXT, a struktura MQPMO określa parametr MQPMO_SET_ALL_CONTEXT.

Domyślnie ta właściwość jest ustawiona na wartość XMSC_WMQ_MDCTX_DEFAULT.

Uwaga: Ta właściwość nie ma znaczenia, gdy aplikacja łączy się z systemem Integration Bus.

Następujące właściwości wymagają, aby właściwość XMSC_WMQ_MQMD_MESSAGE_CONTEXT została ustawiona na wartość właściwości XMSC_WMQ_MDCTX_SET_IDENTITY_CONTEXT lub wartość właściwości XMSC_WMQ_MDCTX_SET_ALL_CONTEXT podczas wysyłania komunikatu w celu uzyskania pożądanego efektu:

- JMS_IBM_MQMD_USERIDENTIFIER
- JMS_IBM_MQMD_ACCOUNTINGTOKEN,
- JMS_IBM_MQMD_APPLIDENTITYDATA

Następujące właściwości wymagają, aby właściwość XMSC_WMQ_MQMD_MESSAGE_CONTEXT została ustawiona na wartość właściwości XMSC_WMQ_MDCTX_SET_ALL_CONTEXT podczas wysyłania komunikatu w celu uzyskania pożądanego efektu:

- JMS_IBM_MQMD_PUTAPPLTYPE
- JMS_IBM_MQMD_PUTAPPLNAME
- Data PUTDATE JMS_IBM_MQMD_PUTDATE
- Czas PUTTIME JMS_IBM_MQMD_PUTTIME
- JMS_IBM_MQMD_APPLORIGINDATA

XMSC_WMQ_MQMD_READ_ENABLED

Typ danych:
System.Int32

Właściwość:
Miejsce docelowe

Ta właściwość określa, czy aplikacja XMS może wyodrębnić wartości pól MQMD, czy nie.

Poprawne wartości dla tej właściwości to:

XMSC_WMQ_READ_ENABLED_NO
Podczas wysyłania komunikatów właściwości JMS_IBM_MQMD* wysłanego komunikatu nie są aktualizowane w celu odzwierciedlenia zaktualizowanych wartości pól w strukturze MQMD.

Podczas odbierania komunikatów żaden z właściwości JMS_IBM_MQMD* nie jest dostępny dla odebranego komunikatu, nawet jeśli niektóre lub wszystkie z nich są ustawione przez nadawcę.

XMSC_WMQ_READ_ENABLED_YES
Podczas wysyłania komunikatów wszystkie właściwości JMS_IBM_MQMD* wysłanego komunikatu są aktualizowane w celu odzwierciedlenia zaktualizowanych wartości pól w strukturze MQMD, w tym tych właściwości, które nie zostały jawnie ustawione przez nadawcę.

Podczas odbierania komunikatów wszystkie właściwości JMS_IBM_MQMD* są dostępne w odebranym komunikacie, włącznie z właściwościami, które nie zostały jawnie ustawione przez nadawcę.

Domyślnie ta właściwość jest ustawiona na wartość XMSC_WMQ_READ_ENABLED_NO.

XMSC_WMQ_MQMD_WRITE_ENABLED

Typ danych:
System.Int32

Właściwość:
Miejsce docelowe

Ta właściwość określa, czy aplikacja XMS może mieć wartości pól MQMD, czy nie.

Poprawne wartości dla tej właściwości to:

XMSC_WMQ_WRITE_ENABLED_NO
Wszystkie właściwości JMS_IBM_MQMD* są ignorowane, a ich wartości nie są kopiowane do bazowej struktury MQMD.

XMSC_WMQ_WRITE_ENABLED_YES
Przetwarzane są właściwości JMS_IBM_MQMD*. Ich wartości są kopiowane do bazowej struktury MQMD.

Domyślnie ta właściwość jest ustawiona na wartość XMSC_WMQ_WRITE_ENABLED_NO.

XMSC_WMQ_PUT_ASYNC_ALLOWED

Typ danych:
System.Int32

Właściwość:
Miejsce docelowe

Ta właściwość określa, czy producenci komunikatów mogą używać asynchronicznych operacji put w celu wysyłania komunikatów do tego miejsca docelowego.

Poprawne wartości dla tej właściwości to:

XMSC_WMQ _PUT_ASYNC_ALLOWED_AS_DEST
Określ, czy dozwolone są operacje put asynchroniczne, odwołując się do definicji kolejki lub tematu.

XMSC_WMQ _PUT_ASYNC_ALLOWED_AS_Q_DEF
Określ, czy dozwolone są operacje put asynchroniczne, odwołując się do definicji kolejki.

XMSC_WMQ _PUT_ASYNC_ALLOWED_AS_TOPIC_DEF
Określ, czy dozwolone są operacje put asynchroniczne, odwołując się do definicji tematu.

XMSC_WMQ _PUT_ASYNC_ALLOWED_DISABLED
Asynchroniczne operacje put są niedozwolone.

XMSC_WMQ _PUT_ASYNC_ALLOWED_ENABLED
Dozwolone są asynchroniczne operacje put.

Domyślnie ta właściwość jest ustawiona na wartość XMSC_WMQ _PUT_ASYNC_ALLOWED_AS_DEST.

Uwaga: Ta właściwość nie ma znaczenia, gdy aplikacja łączy się z systemem Integration Bus.

XMSC_WMQ_READ_AHEAD_ALLOWED

Typ danych:
System.Int32

Właściwość:
Miejsce docelowe

Ta właściwość określa, czy konsumenty komunikatów i przeglądarki kolejek mogą używać operacji odczytu z wyprzedzeniem do pobierania nietrwałych komunikatów nietransakcyjnych z tego miejsca docelowego do buforu wewnętrznego przed ich odebraniem.

Poprawne wartości dla tej właściwości to:

XMSC_WMQ_READ_AHEAD_ALLOWED_AS_Q_DEF
Określ, czy odczyt z wyprzedzeniem jest dozwolony, odwołując się do definicji kolejki.

XMSC_WMQ_READ_AHEAD_ALLOWED_AS_ TEMAT _DEF
Określ, czy odczyt z wyprzedzeniem jest dozwolony, odwołując się do definicji tematu.

XMSC_WMQ_READ_AHEAD_ALLOWED_AS_DEST
Określ, czy odczyt z wyprzedzeniem jest dozwolony, odwołując się do definicji kolejki lub tematu.

XMSC_WMQ_READ_AHEAD_ALLOWED_DISABLED
Odczyt z wyprzedzeniem nie jest dozwolony podczas korzystania z komunikatów lub przeglądania komunikatów.

XMSC_WMQ_READ_AHEAD_ALLOWED_ENABLED
Odczyt z wyprzedzeniem jest dozwolony.

Domyślnie ta właściwość jest ustawiona na wartość XMSC_WMQ _READ_AHEAD_ALLOWED_AS_DEST.

XMSC_WMQ_READ_AHEAD_CLOSE_POLICY

Typ danych:
System.Int32

Właściwość:
Miejsce docelowe

W przypadku komunikatów dostarczanych do asynchronicznego procesu nasłuchującego komunikatów ta właściwość określa, co stanie się z komunikatami w wewnętrznym buforze odczytu z wyprzedzeniem, gdy konsument komunikatów zostanie zamknięty.

Ta właściwość ma zastosowanie w przypadku określania opcji kolejki zamykającej podczas odbierania komunikatów z miejsca docelowego i nie ma zastosowania podczas wysyłania komunikatów do miejsca docelowego.

Ta właściwość jest ignorowana w przypadku Przeglądarki kolejek, ponieważ podczas przeglądania komunikaty są nadal dostępne w kolejkach.

Poprawne wartości dla tej właściwości to:

XMSC_WMQ_READ_AHEAD_CLOSE_POLICY_DELIVER_CURRENT
Tylko bieżące wywołanie nasłuchiwania komunikatów kończy się przed zwróceniem, potencjalnie pozostawiając komunikaty w wewnętrznym buforze odczytu z wyprzedzeniem, które następnie są usuwane.

XMSC_WMQ_READ_AHEAD_CLOSE_POLICY_DELIVER_ALL
Wszystkie komunikaty w wewnętrznym buforze odczytu z wyprzedzeniem są dostarczane do obiektu nasłuchiwania komunikatów aplikacji przed zwróceniem.

Domyślnie ta właściwość jest ustawiona na wartość XMSC_WMQ _READ_AHEAD_CLOSE_POLICY_DELIVER_CURRENT.

Uwaga:

- **Nieprawidłowe zakończenie aplikacji**

  Wszystkie komunikaty w buforze odczytu z wyprzedzeniem są tracone, gdy aplikacja XMS przerywa przerywanie.

- **Konsekwencje dla transakcji**

  Funkcja odczytu z wyprzedzeniem jest wyłączona, gdy aplikacje korzystają z transakcji. Tak więc aplikacja nie widzi żadnej różnicy w zachowaniu podczas używania sesji transakcyjnych.

- **Konsekwencje trybów potwierdzania sesji**

Funkcja odczytu z wyprzedzeniem jest włączana w sesji bez transakcji, gdy tryby potwierdzenia są następujące: XMSC_AUTO_ACKNOWLEDGE lub XMSC_DUPS_OK_ACKNOWLEDGE. Funkcja odczytu z wyprzedzeniem jest wyłączona, jeśli tryb potwierdzenia sesji to XMSC_CLIENT_ACKNOWLEDGE, bez względu na transakcję lub sesje bez transakcji.

- **Konsekwencje dla przeglądarek kolejek i selektorów przeglądarki kolejek**

  Przeglądarki kolejek i selektory przeglądarki kolejek, używane w aplikacjach XMS , uzyskają przewagę wydajności z odczytu z wyprzedzeniem. Zamknięcie przeglądarki kolejek nie pogarsza wydajności, ponieważ komunikat jest nadal dostępny w kolejce do dalszego działania. Nie ma żadnych innych implikacji w przeglądarkach kolejek i selektorach przeglądarki kolejek poza korzyściami z wydajności odczytu z wyprzedzeniem.

XMSC_WMQ_HOST_NAME

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Obiekty mające zastosowanie:
Długa nazwa narzędzia administracyjnego JMS: HOSTNAME

Krótka nazwa narzędzia administracyjnego JMS: HOST

Miejsce działania menedżera kolejek: nazwa hosta lub adres IP systemu.

Ta właściwość jest używana tylko wtedy, gdy aplikacja łączy się z menedżerem kolejek w trybie klienta. Ta właściwość jest używana z właściwością XMSC_WMQ_PORT do identyfikowania menedżera kolejek.

Wartością domyślną tej właściwości jest `localhost`.

XMSC_WMQ_LOCAL_ADDRESS

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Obiekty mające zastosowanie:
Długa nazwa narzędzia administracyjnego JMS: LOCALADDRESS

Krótka nazwa narzędzia administracyjnego JMS: LA

W przypadku połączenia z menedżerem kolejek: ta właściwość określa używaną wartość interfejsu sieci lokalnej albo portu lokalnego (lub zakresu portów lokalnych) bądź obie te wartości.

Wartością właściwości jest łańcuch o następującym formacie:

[*nazwa_hosta*] [(*low_port*) [,*high_port*])]

Znaczenia zmiennych są następujące:

nazwa_hosta
Nazwa hosta lub adres IP lokalnego interfejsu sieciowego, który ma być używany dla połączenia.

Podanie tych informacji jest konieczne tylko wtedy, gdy system, na którym działa aplikacja, ma dwa lub więcej interfejsów sieciowych i użytkownik musi być w stanie określić, który interfejs musi być używany dla połączenia. Jeśli system ma tylko jeden interfejs sieciowy, tylko ten interfejs może być używany. Jeśli system ma dwa lub więcej interfejsów sieciowych i nie określono interfejsu, który musi być używany, interfejs jest wybierany losowo.

low_port
Numer portu lokalnego, który ma być używany dla połączenia.

Jeśli podano również wartość *high_port* , wartość *low_port* jest interpretowana jako najniższy numer portu w zakresie numerów portów.

wysoki_port
Najwyższy numer portu w zakresie numerów portów. Jeden z portów w podanym zakresie musi być używany dla połączenia.

Maksymalna długość łańcucha wynosi 48 znaków.

Poniżej przedstawiono kilka przykładów poprawnych wartości właściwości:

JUPITER
9.20.4.98
JUPITER (1000)
9.20.4.98(1000,2000)
(1000)
(1000,2000)

Domyślnie właściwość nie jest ustawiona.

Ta właściwość ma znaczenie tylko wtedy, gdy aplikacja łączy się z menedżerem kolejek w trybie klienta.

XMSC_WMQ_MESSAGE_SELECTION

Typ danych:
System.Int32

Właściwość:
ConnectionFactory

Określa, czy wybór komunikatów jest dokonany przez klienta XMS , czy przez brokera.

Poprawne wartości właściwości są następujące:

Poprawna wartość	Znaczenie
KLIENT XMSC_WMQ_MSEL_CLIENT	Wybór komunikatów jest dokonany przez klienta XMS .
XMSC_WMQ_MSEL_BROKER	Wybór komunikatu jest dokonany przez broker.

Wartością domyślną jest XMSC_WMQ_MSEL_CLIENT.

Ta właściwość jest odpowiednia tylko w domenie Publikowanie/subskrypcja . Wybór komunikatu przez broker nie jest obsługiwany, jeśli właściwość XMSC_WMQ_BROKER_VERSION jest ustawiona na wartość XMSC_WMQ_BROKER_V1.

XMSC_WMQ_MSG_BATCH_SIZE

Typ danych:
System.Int32

Właściwość:
ConnectionFactory

Maksymalna liczba komunikatów pobieranych z kolejki w jednej partii, gdy komunikaty są dostarczane w trybie asynchronicznym.

Gdy aplikacja używa asynchronicznego dostarczania komunikatów, pod pewnymi warunkami klient XMS pobiera partię komunikatów z kolejki przed przekazaniem poszczególnych komunikatów indywidualnie do aplikacji. Ta właściwość określa maksymalną liczbę komunikatów, które mogą znajdować się w zadaniu wsadowym.

Wartością właściwości jest dodatnia liczba całkowita, a wartością domyślną jest 10. Należy rozważyć ustawienie tej właściwości na inną wartość tylko wtedy, gdy występuje konkretny problem z wydajnością, który należy rozwiązać.

Jeśli aplikacja jest połączona z menedżerem kolejek za pośrednictwem sieci, podniesienie wartości tej właściwości może zmniejszyć koszty ogólne i czasy odpowiedzi sieci, ale zwiększyć ilość pamięci

wymaganej do zapisania komunikatów w systemie klienckim. I odwrotnie, obniżenie wartości tej właściwości może spowodować zwiększenie liczby przegród sieciowych i czasów odpowiedzi, ale zmniejszenie ilości pamięci wymaganej do przechowywania komunikatów.

XMSC_WMQ_POLLING_INTERVAL

Typ danych:
System.Int32

Właściwość:
ConnectionFactory

Jeśli kolejka żadnego procesu nasłuchującego w ramach sesji nie zawiera odpowiedniego komunikatu, jest to maksymalny odstęp czasu (w milisekundach) między kolejnymi próbami pobrania komunikatu z kolejki przez każdy z procesów nasłuchujących komunikatów.

Jeśli często zdarza się, że żaden odpowiedni komunikat nie jest dostępny dla żadnego z obiektów nasłuchiwania komunikatów w sesji, należy rozważyć zwiększenie wartości tej właściwości.

Wartość właściwości jest dodatnią liczbą całkowitą. Wartość domyślna to 5000.

PORT XMSC_WMQ_PORT

Typ danych:
System.Int32

Właściwość:
ConnectionFactory

Obiekty mające zastosowanie:
Długa nazwa narzędzia administracyjnego JMS: PORT

Krótka nazwa narzędzia administracyjnego JMS: PORT

Numer portu, na którym menedżer kolejek nasłuchuje żądań przychodzących.

Ta właściwość jest używana tylko wtedy, gdy aplikacja łączy się z menedżerem kolejek w trybie klienta. Ta właściwość jest używana z właściwością XMSC_WMQ_HOST_NAME do identyfikowania menedżera kolejek.

Wartością domyślną tej właściwości jest XMSC_WMQ_DEFAULT_CLIENT_PORT lub 1414.

XMSC_WMQ_PROVIDER_VERSION

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Wersja, wydanie, poziom modyfikacji i pakiet poprawek menedżera kolejek, z którym aplikacja ma nawiązać połączenie. Poprawne wartości dla tej właściwości to:

- Nieokreślona

Lub łańcuch w jednym z następujących formatów

- V.R.M.F
- V.R.M
- V.R
- V

Gdzie: V, R, M i F są wartościami całkowitymi większymi niż zero lub równymi zero.

Wartość 7 lub większa wskazuje, że ta wersja jest przeznaczona jako obiekt IBM WebSphere MQ 7.0 ConnectionFactory dla połączeń z menedżerem kolejek produktu IBM WebSphere MQ 7.0 . Wartość wcześniejsza niż 7 (na przykład "6.0.2.0"), wskazuje, że jest on przeznaczony do użytku z menedżerami

kolejek w wersjach wcześniejszych niż wersja 7.0. Wartość domyślna, nieokreślona, pozwala na połączenia z dowolnym poziomem menedżera kolejek, określając odpowiednie właściwości i funkcje dostępne w oparciu o możliwości menedżera kolejek.

Domyślnie ta właściwość jest ustawiona na wartość "unspecified" (nieokreślona).

Uwaga:

- Jeśli wartość XMSC_WMQ_PROVIDER_VERSION jest ustawiona na 6, nie jest wykonywane współużytkowanie przez gniazdo. 2.
- Połączenie nie powiedzie się, jeśli wartość XMSC_WMQ_PROVIDER_VERSION jest ustawiona na 7, a na serwerze SHARECNV dla kanału jest ustawiona wartość 0.
- Składniki specyficzne dla produktu IBM WebSphere MQ 7.0 są wyłączone, jeśli właściwość XMSC_WMQ_PROVIDER_VERSION jest ustawiona na wartość UNSPECIFIED, a wartość SHARECNV jest ustawiona na 0.

Wersja produktu IBM WebSphere MQ Client również odgrywa główną rolę w tym, czy aplikacja kliencka XMS może korzystać z funkcji specyficznych dla produktu IBM WebSphere MQ 7.0 . W poniższej tabeli opisano zachowanie.

Uwaga: Właściwość systemowa XMSC_WMQ_OVERRIDEPROVIDERVERSION przesłania właściwość XMSC_WMQ_PROVIDER_VERSION. Ta właściwość może być używana, jeśli nie można zmienić ustawienia fabryki połączeń.

Tabela 33. Klient XMS -możliwość korzystania z funkcji specyficznych dla produktu IBM WebSphere MQ 7.0 .

#	XMSC_WMQ_PROVIDER_VERSION	IBM WebSphere MQ Wersja klienta	IBM WebSphere MQ 7.0 opcje
1	nieokreślona	7	WŁĄCZ
2	nieokreślona	6	WYŁĄCZ
3	7	7	WŁĄCZ
4	7	6	Wyjątek
5	6	6	WYŁĄCZ
6	6	7	WYŁĄCZ

XMSC_WMQ_PUB_ACK_INTERVAL

Typ danych:
System.Int32

Właściwość:
ConnectionFactory

Liczba komunikatów publikowanych przez publikatora, zanim klient XMS zażąda potwierdzenia od brokera.

Jeśli wartość tej właściwości zostanie zmniejszona, klient będzie żądał potwierdzeń częściej, a w związku z tym wydajność publikatora maleje. Jeśli wartość zostanie zwiększona, zgłaszanie wyjątku przez klient w przypadku awarii brokera będzie trwać dłużej.

Wartość właściwości jest dodatnią liczbą całkowitą. Wartość domyślna wynosi 25.

CCSID XMSC_WMQ_QMGR_CCSID

Typ danych:
System.Int32

Właściwość:
ConnectionFactory

Identyfikator (CCSID) kodowanego zestawu znaków lub strony kodowej, w którym pola danych znakowych zdefiniowane w interfejsie kolejki komunikatów (Message Queue Interface-MQI) są wymieniane między klientem XMS a klientem WebSphere MQ . Ta właściwość nie ma zastosowania do łańcuchów danych znakowych w ciałach komunikatów.

Gdy aplikacja XMS łączy się z menedżerem kolejek w trybie klienta, klient XMS łączy się z klientem WebSphere MQ . Informacje wymieniane między dwoma klientami zawierają pola danych znakowych, które są zdefiniowane w interfejsie MQI. W normalnych okolicznościach klient WebSphere MQ zakłada, że te pola znajdują się na stronie kodowej systemu, na którym są uruchomione klienty. Jeśli klient XMS udostępnia te pola na innej stronie kodowej i oczekuje, że je otrzymają, należy ustawić tę właściwość w celu poinformowania klienta WebSphere MQ .

Gdy klient WebSphere MQ przekazuje te pola danych znakowych do menedżera kolejek, dane w nich muszą zostać przekształcone, jeśli to konieczne, na stronę kodową używaną przez menedżer kolejek. Podobnie, gdy klient WebSphere MQ odbiera te pola od menedżera kolejek, dane w nich muszą zostać przekształcone, jeśli jest to konieczne, na stronę kodową, w której klient XMS oczekuje otrzymania tych danych. Klient WebSphere MQ używa tej właściwości do wykonywania tych konwersji danych.

Domyślnie właściwość nie jest ustawiona.

Ustawienie tej właściwości jest równoznaczne z ustawieniem zmiennej środowiskowej MQCCSID dla klienta WebSphere MQ , który obsługuje rodzime aplikacje klienckie produktu WebSphere MQ . Więcej informacji na temat tej zmiennej środowiskowej można znaleźć w sekcji *KlientyWebSphere MQ*.

XMSC_WMQ_QUEUE_MANAGER

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Obiekty mające zastosowanie:
Długa nazwa narzędzia administracyjnego JMS: QMANAGER

Krótka nazwa narzędzia administracyjnego JMS: QMGR

Nazwa menedżera kolejek, z którym ma zostać nawiązane połączenie.

Domyślnie właściwość nie jest ustawiona.

CCSID XMSC_WMQ_RECEIVE_CCSID

Właściwość docelowa, która ustawia docelowy identyfikator CCSID dla konwersji komunikatów menedżera kolejek. Wartość jest ignorowana, chyba że parametr XMSC_WMQ_RECEIVE_CONVERSION jest ustawiony na wartość WMQ_RECEIVE_CONVERSION_QMGR.

Typ danych:
Liczba całkowita

Wartość:
Dowolna dodatnia liczba całkowita.

Wartość domyślna to 1208.

Podanie wartości `GMO_CONVERT` w komunikacie jest opcjonalne. Jeśli podana jest wartość `GMO_CONVERT` , konwersja odbywa się zgodnie z podaną wartością.

XMSC_WMQ_RECEIVE_CONVERSION

Właściwość miejsca docelowego, która określa, czy konwersja danych ma być wykonywana przez menedżer kolejek.

Typ danych:
Liczba całkowita

Wartości:

XMSC_WMQ_RECEIVE_CONVERSION_CLIENT_MSG (DEFAULT): Wykonywanie konwersji danych tylko na kliencie XMS . Konwersja jest zawsze wykonywana za pomocą strony kodowej 1208.

XMSC_WMQ_RECEIVE_CONVERSION_QMGR: Przed wysłaniem komunikatu do klienta XMS należy przeprowadzić konwersję danych w menedżerze kolejek.

XMSC_WMQ_RECEIVE_EXIT

Typ danych:

Łańcuch

Właściwość:

ConnectionFactory

Identyfikuje wyjście odbierania kanału, które ma zostać uruchomione.

Wartość właściwości jest łańcuchem, który identyfikuje wyjście odbierania kanału i ma następujący format:

libraryName(entryPointNazwa)

gdzie:

- **libraryName** to pełna ścieżka do zarządzanego wyjścia `.dll`
- `entryPointNazwa` to nazwa klasy kwalifikowana przez przestrzeń nazw.

Na przykład `C:\MyReceiveExit.dll(MyReceiveExitNameSpace.MyReceiveExitClassName)`

Domyślnie właściwość nie jest ustawiona.

Ta właściwość ma znaczenie tylko wtedy, gdy aplikacja łączy się z menedżerem kolejek w trybie klienta zarządzanego. Obsługiwane są również tylko wyjścia zarządzane.

XMSC_WMQ_RECEIVE_EXIT_INIT

Typ danych:

Łańcuch

Właściwość:

ConnectionFactory

Dane użytkownika przekazywane do wyjścia odbierania kanału w momencie jego wywołania.

Wartością właściwości jest łańcuch. Domyślnie właściwość nie jest ustawiona.

Ta właściwość ma znaczenie tylko wtedy, gdy aplikacja łączy się z menedżerem kolejek w trybie klienta zarządzanego i właściwość “XMSC_WMQ_RECEIVE_EXIT” na stronie 233 jest ustawiona.

XMSC_WMQ_RESOLVED_QUEUE_MANAGER

Typ danych:

Łańcuch

Właściwość:

ConnectionFactory

Ta właściwość jest używana do uzyskiwania nazwy menedżera kolejek, z którym jest on połączony.

W przypadku użycia z tabelą CCDT (Tabela definicji kanału klienta) ta nazwa może być inna niż nazwa menedżera kolejek określona w fabryce połączeń.

XMSC_WMQ_RESOLVED_QUEUE_MANAGER_ID

Typ danych:

Łańcuch

Właściwość:

ConnectionFactory

Ta właściwość jest zapełniana za pomocą identyfikatora menedżera kolejek po połączeniu.

XMSC_WMQ_SECURITY_EXIT

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Identyfikuje wyjście zabezpieczeń kanału.

Wartość właściwości to łańcuch, który identyfikuje wyjście zabezpieczeń kanału i ma następujący format:

libraryName(entryPointNazwa)

gdzie:

- **libraryName** jest pełną ścieżką do zarządzanego wyjścia .dll.
- entryPointNazwa to nazwa klasy kwalifikowana przez przestrzeń nazw.

Na przykład: C:\MySecurityExit.dll(MySecurityExitNameSpace.MySecurityExitClassName)

Maksymalna długość łańcucha wynosi 128 znaków.

Domyślnie właściwość nie jest ustawiona.

Ta właściwość ma znaczenie tylko wtedy, gdy aplikacja łączy się z menedżerem kolejek w trybie klienta zarządzanego. Obsługiwane są również tylko wyjścia zarządzane.

XMSC_WMQ_SECURITY_EXIT_INIT

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Dane użytkownika przekazywane do wyjścia zabezpieczeń kanału w momencie jego wywołania.

Maksymalna długość łańcucha danych użytkownika wynosi 32 znaki.

Domyślnie właściwość nie jest ustawiona.

Ta właściwość ma znaczenie tylko wtedy, gdy aplikacja łączy się z menedżerem kolejek w trybie klienta zarządzanego i właściwość "XMSC_WMQ_SECURITY_EXIT" na stronie 234 jest ustawiona.

XMSC_WMQ_SEND_EXIT

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Identyfikuje wyjście wysyłania kanału.

Wartością właściwości jest łańcuch. Wyjście wysyłania kanału ma następujący format:

libraryName(entryPointNazwa)

gdzie:

- **libraryName** jest pełną ścieżką do zarządzanego wyjścia .dll.
- entryPointNazwa to nazwa klasy kwalifikowana przez przestrzeń nazw.

Na przykład: C:\MySendExit.dll(MySendExitNameSpace.MySendExitClassName)

Domyślnie właściwość nie jest ustawiona.

Ta właściwość ma znaczenie tylko wtedy, gdy aplikacja łączy się z menedżerem kolejek w trybie klienta zarządzanego. Obsługiwane są również tylko wyjścia zarządzane.

XMSC_WMQ_SEND_EXIT_INIT

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Dane użytkownika przekazywane do wyjść wysyłania kanału w momencie ich wywołania.

Wartość właściwości to łańcuch zawierający jeden lub więcej elementów danych użytkownika oddzielonych przecinkami. Domyślnie właściwość nie jest ustawiona.

Reguły określania danych użytkownika, które są przekazywane do sekwencji wyjść wysyłania kanału, są takie same jak reguły określania danych użytkownika, które są przekazywane do sekwencji wyjść odbierania kanału. W związku z tym w przypadku reguł patrz “XMSC_WMQ_RECEIVE_EXIT_INIT” na stronie 233.

Ta właściwość ma znaczenie tylko wtedy, gdy aplikacja łączy się z menedżerem kolejek w trybie klienta zarządzanego i właściwość “XMSC_WMQ_SEND_EXIT” na stronie 234 jest ustawiona.

XMSC_WMQ_SEND_CHECK_COUNT

Typ danych:
System.Int32

Właściwość:
ConnectionFactory

Liczba wywołań wysyłania dozwolonych między sprawdzeniami błędów asynchronicznego umieszczania w ramach jednej sesji XMS bez transakcji.

Domyślnie ta właściwość jest ustawiona na 0.

XMSC_WMQ_SHARE_CONV_ALLOWED

Typ danych:
System.Int32

Właściwość:
ConnectionFactory

Obiekty mające zastosowanie:
Długa nazwa narzędzia administracyjnego JMS: SHARECONVALLOWED

Krótka nazwa narzędzia administracyjnego JMS: SCALD

Określa, czy połączenie klienta może współużytkować swoje gniazdo z innymi połączeniami XMS najwyższego poziomu z tego samego procesu do tego samego menedżera kolejek, jeśli definicje kanałów są zgodne. Ta właściwość umożliwia pełną izolację połączeń w osobnych gniazdach (jeśli jest to konieczne z powodów związanych z tworzeniem, konserwacją lub działaniem aplikacji). Ustawienie tej właściwości tylko wskazuje na wartość XMS , aby udostępnić gniazdo bazowe. Nie wskazuje na to, ile połączeń współużytkuje pojedyncze gniazdo. Liczba połączeń współużytkującej gniazdo jest określana przez wartość SHARECNV, która jest negocjowana między klientem WebSphere MQ a serwerem WebSphere MQ .

Aby ustawić właściwość, aplikacja może ustawić następujące stałe nazwane:

- XMSC_WMQ_SHARE_CONV_ALLOWED_FALSE-Połączenia nie współużytkowają gniazda.
- XMSC_WMQ_SHARE_CONV_ALLOWED_TRUE-Połączenia współużytkowają gniazdo.

Domyślnie właściwość ta jest ustawiona na wartość XMSC_WMQ_SHARE_CONV_ALLOWED_ENABLED.

Ta właściwość ma znaczenie tylko wtedy, gdy aplikacja łączy się z menedżerem kolejek w trybie klienta.

XMSC_WMQ_SSL_CERT_STORES

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Lokalizacje serwerów, na których znajdują się listy odwołań certyfikatów (Certificate Revocation List – CRL) używane podczas nawiązywania połączenia SSL z menedżerem kolejek.

Wartość właściwości to lista adresów URL oddzielonych przecinkami. Każdy adres URL ma następujący format:

```
[user[/password]@]ldap://[serveraddress][:portnum][,...]
```

Ten format jest zgodny z podstawowym formatem MQJMS, ale rozszerzonym z poziomu podstawowego.

Poprawne jest posiadanie pustego `serveraddress`. W tym przypadku program XMS przyjmuje, że wartością jest łańcuch "localhost".

Przykładowa lista:

```
myuser/mypassword@ldap://server1.mycom.com:389
ldap://server1.mycom.com
ldap://
ldap://:389
```

Tylko w przypadku produktu .NET : Z poziomu produktu IBM MQ 8.0połączenia zarządzane z produktem IBM MQ (WMQ_CM_CLIENT) i niezarządzane połączenia z serwerem IBM MQ (WMQ_CM_CLIENT_UNMANAGED) obsługują połączenia TLS/SSL.

Domyślnie właściwość nie jest ustawiona.

Informacje pokrewne
Obsługa protokołu SSL i TLS dla niezarządzanego klienta .NET
Obsługa protokołu SSL i TLS dla zarządzanego klienta .NET

XMSC_WMQ_SSL_CIPHER_SPEC

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Nazwa specyfikacji szyfrowania używanej w przypadku bezpiecznego połączenia z menedżerem kolejek.

Specyfikacje szyfru, które mogą być używane z obsługą protokołu IBM WebSphere MQ TLS, są wymienione w poniższej tabeli. Jeśli żądasz certyfikatu osobistego, należy podać wielkość klucza dla pary kluczy publicznego i prywatnego. Wielkość klucza używana podczas uzgadniania SSL jest wielkością zapisaną w certyfikacie, o ile nie jest ona określona przez atrybut CipherSpec, co zostało określone w tabeli. Domyślnie ta właściwość nie jest ustawiona.

Nazwa specyfikacji szyfrowania	Używany protokół	Algorytm mieszający	Algorytm szyfrowania	Bity szyfrowania	FIPS[1]	Zestaw B 128 bitów	Pakiet B 192 bit
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	SHA-1	AES	128	Tak	Nie	Nie
TLS_RSA_WITH_AES_256_CBC_SHA[2]	TLS 1.0	SHA-1	AES	256	Tak	Nie	Nie
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	SHA-1	DES	56	Nie	Nie	Nie

Nazwa specyfikacji szyfrowania	Używany protokół	Algorytm mieszający	Algorytm szyfrowania	Bity szyfrowania	FIPS[1]	Zestaw B 128 bitów	Pakiet B 192 bit
TLS_RSA_WITH_3DES_EDE_CBC_SHA[4]	TLS 1.0	SHA-1	3DES	168	Tak	Nie	Nie
TLS_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	SHA-256	AES	128	Tak	Nie	Nie
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	SHA-384	AES	256	Tak	Nie	Nie
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	SHA-256	AES	128	Tak	Nie	Nie
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	SHA-256	AES	256	Tak	Nie	Nie
ECDHE_ECDSA_RC4_128_SHA256	TLS 1.2	SHA-256	RC4	128	Nie	Nie	Nie
ECDHE_ECDSA_3DES_EDE_CBC_SHA256	TLS 1.2	SHA-256	3DES	168	Tak	Nie	Nie
ECDHE_RSA_RC4_128_SHA256	TLS 1.2	SHA-256	RC4	128	Nie	Nie	Nie
ECDHE_RSA_3DES_EDE_CBC_SHA256	TLS 1.2	SHA-256	3DES	168	Tak	Nie	Nie
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	SHA-256	AES	128	Tak	Nie	Nie
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	SHA-384	AES	256	Tak	Nie	Nie
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	SHA-256	AES	128	Tak	Nie	Nie
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	SHA-384	AES	256	Tak	Nie	Nie
ECDHE_ECDSA_AES_128_GCM_SHA256	TLS 1.2	SHA-256	AES	128	Tak	Tak	Nie
ECDHE_ECDSA_AES_256_GCM_SHA384	TLS 1.2	SHA-384	AES	256	Tak	Nie	Tak
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	SHA-256	AES	128	Tak	Nie	Nie
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	SHA-384	AES	256	Tak	Nie	Nie
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	SHA-256	Brak	0	Nie	Nie	Nie
ECDHE_RSA_NULL_SHA256	TLS 1.2	SHA-256	Brak	0	Nie	Nie	Nie
ECDHE_ECDSA_NULL_SHA256	TLS 1.2	SHA-256	Brak	0	Nie	Nie	Nie
TLS_RSA_WITH_NULL_NULL	TLS 1.2	Brak	Brak	0	Nie	Nie	Nie

Nazwa specyfikacji szyfrowania	Używany protokół	Algorytm mieszający	Algorytm szyfrowania	Bity szyfrowania	FIPS[1]	Zestaw B 128 bitów	Pakiet B 192 bit
TLS_RSA_WITH_RC4_128_SHA256	TLS 1.2	SHA-256	RC4	128	Nie	Nie	Nie

Uwagi:

1. Określa, czy specyfikacja CipherSpec jest zgodna ze standardem FIPS (Federal Information Processing Standards) 140-2. Wyjaśnienie standardu FIPS oraz informacje na temat sposobu konfigurowania produktu WebSphere MQ dla zgodnego ze standardem FIPS 140-2 znajdują się w dokumencie *Federal Information Processing Standards (FIPS)* w elektronicznej dokumentacji produktu IBM IBM WebSphere MQ .
2. Tej specyfikacji CipherSpec nie można użyć do zabezpieczenia połączenia z programu WebSphere MQ Explorer do menedżera kolejek, chyba że odpowiednie nieograniczone pliki strategii są stosowane w środowisku JRE używanym przez eksplorator.
3. Ta specyfikacja szyfrowania uzyskała certyfikat FIPS 140-2 przed 19 maja 2007.
4. Jeśli produkt WebSphere MQ jest skonfigurowany dla operacji zgodnej ze standardem FIPS 140-2, ta specyfikacja CipherSpec może zostać użyta do przesłania do 32 GB danych przed przerwaną połączeniem z błędem AMQ9288. Aby uniknąć tego błędu, należy unikać używania potrójnego algorytmu DES (który jest nieaktualny) lub włączyć resetowanie klucza tajnego podczas używania tej specyfikacji CipherSpec w konfiguracji FIPS 140-2.

Informacje pokrewne

Zabezpieczanie

Integralność danych komunikatów

Określanie parametru CipherSpecs

XMSC_WMQ_SSL_CIPHER_SUITE

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Nazwa zestawu algorytmów szyfrowania używanego w przypadku połączenia TLS z menedżerem kolejek. Na podstawie podanego zestawu algorytmów szyfrowania jest określany protokół używany do negocjowania bezpiecznego połączenia.

Ta właściwość ma następujące wartości kanoniczne:

- 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

Ta wartość może być podana jako alternatywa dla XMSC_WMQ_SSL_CIPHER_SPEC.

Jeśli dla parametru XMSC_WMQ_SSL_CIPHER_SPECzostanie podana niepusta wartość, ta wartość przesłania ustawienie dla elementu XMSC_WMQ_SSL_CIPHER_SUITE. Jeśli wartość XMSC_WMQ_SSL_CIPHER_SPEC nie ma wartości, jako zestaw algorytmów szyfrowania dostarczany jest pakiet GSKit, wartość XMSC_WMQ_SSL_CIPHER_SUITE. W tym przypadku wartość jest odwzorowywana na równoważną wartość atrybutu CipherSpec zgodnie z opisem w sekcji “Odwzorowania nazw CipherSuite i CipherSpec na potrzeby połączeń z menedżerem kolejek produktu IBM MQ .” na stronie 69.

Jeśli zarówno parametr XMSC_WMQ_SSL_CIPHER_SPEC, jak i XMSC_WMQ_SSL_CIPHER_SUITE są puste, to pole pChDef->SSLCipherSpec jest wypełnione spacjami.

Tylko w przypadku produktu .NET : Z poziomu produktu IBM MQ 8.0połączenia zarządzane z produktem IBM MQ (WMQ_CM_CLIENT) i niezarządzane połączenia z serwerem IBM MQ (WMQ_CM_CLIENT_UNMANAGED) obsługują połączenia TLS/SSL.

Domyślnie właściwość nie jest ustawiona.

Informacje pokrewne

Obsługa protokołu SSL i TLS dla niezarządzanego klienta .NET

Obsługa protokołu SSL i TLS dla zarządzanego klienta .NET

XMSC_WMQ_SSL_CRYPTO_HW

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Szczegóły konfiguracji sprzętu szyfrującego połączonego z systemem klienta.

Ta właściwość ma następujące wartości kanoniczne:

- GSK_ACCELERATOR_RAINBOW_CS_OFF
- GSK_ACCELERATOR_RAINBOW_CS_ON
- GSK_ACCELERATOR_NCIPHER_NF_OFF
- GSK_ACCELERATOR_NCIPHER_NF_ON

Dostępny jest specjalny format dla sprzętu szyfrującego PKCS11 (gdzie DriverPath, TokenLabeli TokenPassword są łańcuchami określonymi przez użytkownika):

```
GSK_PKCS11=PKCS#11 DriverPath; PKCS#11 TokenLabel;PKCS#11 TokenPassword
```

Program XMS nie interpretuje ani nie zmienia treści łańcucha. Kopiuje on podaną wartość do limitu 256 znaków jednobajtowych w MQSCO.CryptoHardware .

Tylko w przypadku produktu .NET : Z poziomu produktu IBM MQ 8.0połączenia zarządzane z produktem IBM MQ (WMQ_CM_CLIENT) i niezarządzane połączenia z serwerem IBM MQ (WMQ_CM_CLIENT_UNMANAGED) obsługują połączenia TLS/SSL.

Domyślnie właściwość nie jest ustawiona.

Informacje pokrewne

Obsługa protokołu SSL i TLS dla niezarządzanego klienta .NET

Obsługa protokołu SSL i TLS dla zarządzanego klienta .NET

XMSC_WMQ_SSL_FIPS_REQUIRED

Typ danych:
wartość boolowska

Właściwość:
ConnectionFactory

Wartość tej właściwości określa, czy aplikacja może lub nie może używać zestawów algorytmów szyfrowania niezgodnych ze standardem FIPS. Jeśli ta właściwość zostanie ustawiona na wartość true, w przypadku połączeń klient-serwer będzie można używać tylko algorytmów zgodnych ze standardem FIPS.

Ta właściwość może mieć następujące wartości, które przekładają się na dwie wartości kanoniczne dla MQSCO.FipsRequired:

Tabela 34. Tabela wartości dla MQSCO.FlipsRequired , właściwość

Wartość	**Opis**	**Odpowiednia wartość MQSCO.FipsRequired**
Fałsz	Można użyć dowolnego obiektu CipherSpec .	MQSSL_FIPS_NO (wartość domyślna)
Prawda	W specyfikacji CipherSpec stosowane do tego połączenia klienta mogą być używane tylko algorytmy szyfrowania certyfikowane przez FIPS.	MQSSL_FIPS_YES

Program XMS kopiuje odpowiednią wartość do produktu MQSCO.FipsRequired przed wywołaniem komendy MQCONNX.

Parametr MQSCO.FipsRequired jest dostępny tylko w programie WebSphere MQ w wersji 6. Jeśli WebSphere MQ wersja 5.3, jeśli ta właściwość jest ustawiona, program XMS nie podejmie próby nawiązania połączenia z menedżerem kolejek i zamiast niego zgłosi odpowiedni wyjątek.

Tylko w przypadku produktu .NET : Z poziomu produktu IBM MQ 8.0połączenia zarządzane z produktem IBM MQ (WMQ_CM_CLIENT) i niezarządzane połączenia z serwerem IBM MQ (WMQ_CM_CLIENT_UNMANAGED) obsługują połączenia TLS/SSL.

Informacje pokrewne

Obsługa protokołu SSL i TLS dla niezarządzanego klienta .NET

Obsługa protokołu SSL i TLS dla zarządzanego klienta .NET

XMSC_WMQ_SSL_KEY_REPOSITORY

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Położenie pliku bazy danych kluczy, w którym przechowywane są klucze i certyfikaty.

XMS kopiuje łańcuch, maksymalnie do 256 znaków jednobajtowych, do MQSCO.KeyRepository . WebSphere MQ interpretuje ten łańcuch jako nazwę pliku, włącznie z pełną ścieżką.

Tylko w przypadku produktu .NET : Z poziomu produktu IBM MQ 8.0połączenia zarządzane z produktem IBM MQ (WMQ_CM_CLIENT) i niezarządzane połączenia z serwerem IBM MQ (WMQ_CM_CLIENT_UNMANAGED) obsługują połączenia TLS/SSL.

Domyślnie właściwość nie jest ustawiona.

Informacje pokrewne

Obsługa protokołu SSL i TLS dla niezarządzanego klienta .NET

Obsługa protokołu SSL i TLS dla zarządzanego klienta .NET

XMSC_WMQ_SSL_KEY_RESETCOUNT

Typ danych:
System.Int32

Właściwość:
ConnectionFactory

Wartość KeyResetCount reprezentuje całkowitą liczbę nieszyfrowanych bajtów wysłanych i odebranych w ramach konwersacji SSL przed ponownym wynegocjowaniem klucza tajnego. Liczba bajtów obejmuje informacje sterujące wysłane przez agenta MCA.

Program XMS kopiuje wartość dostarczanej dla tej właściwości do MQSCO.KeyResetCount przed wywołaniem komendy MQCONNX.

Parametr MQSCO.KeyRestCount jest dostępny tylko w programie WebSphere MQ w wersji 6. Jeśli WebSphere MQ wersja 5.3, jeśli ta właściwość jest ustawiona, program XMS nie podejmie próby nawiązania połączenia z menedżerem kolejek i zamiast niego zgłosi odpowiedni wyjątek.

Tylko w przypadku produktu .NET : Z poziomu produktu IBM MQ 8.0połączenia zarządzane z produktem IBM MQ (WMQ_CM_CLIENT) i niezarządzane połączenia z serwerem IBM MQ (WMQ_CM_CLIENT_UNMANAGED) obsługują połączenia TLS/SSL.

Wartością domyślną tej właściwości jest zero, co oznacza, że klucze tajne nigdy nie są renegocjowane.

Informacje pokrewne
Obsługa protokołu SSL i TLS dla niezarządzanego klienta .NET
Obsługa protokołu SSL i TLS dla zarządzanego klienta .NET

XMSC_WMQ_SSL_PEER_NAME

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Nazwa węzła sieci używana na potrzeby połączenia SSL z menedżerem kolejek.

Dla tej właściwości nie ma listy wartości kanonicznych. Zamiast tego należy zbudować ten łańcuch zgodnie z regułami protokołu SSLPEER.

Przykład nazwy węzła sieci to:

```
"CN=John Smith, O=IBM ,OU=Test , C=GB"
```

Program XMS kopiuje łańcuch do poprawnej jednobajtowej strony kodowej i umieszcza poprawne wartości w plikach MQCD.SSLPeerNamePtr i MQCD.SSLPeerNameLength przed wywołaniem komendy MQCONNX.

Ta właściwość ma znaczenie tylko wtedy, gdy aplikacja łączy się z menedżerem kolejek w trybie klienta.

Tylko w przypadku produktu .NET : Z poziomu produktu IBM MQ 8.0połączenia zarządzane z produktem IBM MQ (WMQ_CM_CLIENT) i niezarządzane połączenia z serwerem IBM MQ (WMQ_CM_CLIENT_UNMANAGED) obsługują połączenia TLS/SSL.

Domyślnie właściwość nie jest ustawiona.

Informacje pokrewne
Obsługa protokołu SSL i TLS dla niezarządzanego klienta .NET
Obsługa protokołu SSL i TLS dla zarządzanego klienta .NET
SSLPEERNAME

XMSC_WMQ_SYNCPOINT_ALL_GETS

Typ danych:
System.Boolean

Właściwość:
ConnectionFactory

Określa, czy wszystkie komunikaty muszą być pobierane z kolejek w ramach sterowania punktem synchronizacji.

Poprawne wartości właściwości są następujące:

Poprawna wartość	Znaczenie
Fałsz	Jeśli okoliczności są odpowiednie, klient XMS może pobierać komunikaty z kolejek poza elementem sterującym punktu synchronizacji.
Prawda	Klient XMS musi pobrać wszystkie komunikaty z kolejek w ramach elementu sterującego punktu synchronizacji.

Wartością domyślną jest false.

KLIENT XMSC_WMQ_TARGET_CLIENT

Typ danych:
System.Int32

Właściwość:
Miejsce docelowe

Nazwa używana w identyfikatorze URI:
targetClient

Określa, czy komunikaty wysyłane do miejsca docelowego mają zawierać nagłówek MQRFH2.

Jeśli aplikacja wysyła komunikat zawierający nagłówek MQRFH2 , aplikacja odbierający musi być w stanie obsłużyć nagłówek.

Poprawne wartości właściwości są następujące:

Poprawna wartość	Znaczenie
XMSC_WMQ_TARGET_DEST_JMS	Komunikaty wysłane do miejsca docelowego zawierają nagłówek MQRFH2 . Tę wartość należy określić, jeśli aplikacja wysyła komunikaty do innej aplikacji XMS , aplikacji WebSphere JMS lub rodzimej aplikacji produktu WebSphere MQ zaprojektowanej do obsługi nagłówka MQRFH2 .
XMSC_WMQ_TARGET_DEST_MQ	Komunikaty wysłane do miejsca docelowego nie zawierają nagłówka MQRFH2 . Tę wartość należy określić, jeśli aplikacja wysyła komunikaty do rodzimej aplikacji produktu WebSphere MQ , która nie jest przeznaczona do obsługi nagłówka MQRFH2 .

Wartością domyślną jest XMSC_WMQ_TARGET_DEST_JMS.

XMSC_WMQ_TEMP_Q_PREFIX

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Przedrostek używany do utworzenia nazwy kolejki dynamicznej WebSphere MQ utworzonej podczas tworzenia kolejki tymczasowej produktu XMS przez aplikację.

The rules for forming the prefix are the same as the rules for forming the contents of the **DynamicQName** field in an object descriptor, but the last non-blank character must be an asterisk(*). If the property is not set, the value used is `CSQ.*` on z/OS and `AMQ.*` on the other platforms. Domyślnie właściwość nie jest ustawiona.

Ta właściwość jest odpowiednia tylko w domenie punkt z punktem .

XMSC_WMQ_TEMP_TOPIC_PREFIX

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory, Miejsce docelowe

Podczas tworzenia tematów tymczasowych XMS generuje łańcuch tematu w postaci "TEMP/TEMPTOPICPREFIX/unique_id" lub jeśli ta właściwość zawiera wartość domyślną, to ten łańcuch, "TEMP/unique_id", zostanie wygenerowany. Podanie niepustej wartości umożliwia definiowanie konkretnych kolejek modelowych na potrzeby tworzenia zarządzanych kolejek dla subskrybentów tymczasowych tematów utworzonych w ramach tego połączenia.

Dowolny łańcuch inny niż NULL składający się tylko z poprawnych znaków dla łańcucha tematu IBM WebSphere MQ jest poprawną wartością dla tej właściwości.

Domyślnie ta właściwość jest ustawiona na wartość "" (pusty łańcuch).

Uwaga: Ta właściwość jest odpowiednia tylko w domenie publikowania/subskrypcji.

MODEL XMSC_WMQ_TEMPORARY_MODEL

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Nazwa kolejki modelowej WebSphere MQ , z której tworzona jest kolejka dynamiczna, gdy aplikacja tworzy kolejkę tymczasową XMS .

Wartością domyślną tej właściwości jest SYSTEM.DEFAULT.MODEL.QUEUE.

Ta właściwość jest odpowiednia tylko w domenie punkt z punktem .

FORMAT XMSC_WMQ_WILDCARD_FORMAT

Typ danych:
System.Int32

Właściwość:
ConnectionFactory, Miejsce docelowe

Ta właściwość określa, która wersja składni ze znakami wieloznacznymi ma być używana.

W przypadku korzystania z publikowania/subskrypcji z IBM WebSphere MQ '*' i '?' są traktowane jako znaki wieloznaczne. Znaki '#' i '+' są traktowane jako znaki wieloznaczne, gdy opcja publikowania jest subskrybowana przy użyciu produktu IBM Integration Bus. Ta właściwość zastępuje właściwość XMSC_WMQ_BROKER_VERSION.

Poprawne wartości dla tej właściwości to:

XMSC_WMQ_WILDCARD_TOPIC_ONLY
Rozpoznaje tylko znaki zastępcze poziomu tematu, tzn. "#" i '+' są traktowane jako znaki wieloznaczne. Ta wartość jest taka sama jak wartość XMSC_WMQ_BROKER_V2.

XMSC_WMQ_WILDCARD_CHAR_ONLY
Rozpoznaje znaki zastępcze tylko w postaci znaków wieloznacznych. "*" i '?' są traktowane jako znaki wieloznaczne. Ta wartość jest taka sama jak wartość XMSC_WMQ_BROKER_V1.

Domyślnie ta właściwość jest ustawiona na wartość XMSC_WMQ_WILDCARD_TOPIC_ONLY.

XMSC_WPM_BUS_NAME

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory i miejsce docelowe

Nazwa używana w identyfikatorze URI:
busName

W przypadku fabryki połączeń: nazwa magistrali integracji usług, z którą aplikacja nawiązuje połączenie. W przypadku miejsca docelowego: nazwa magistrali integracji usług, w której znajduje się miejsce docelowe.

W przypadku miejsca docelowego, które jest tematem, ta właściwość jest nazwą magistrali integracji usług, w której istnieje powiązany obszar tematu. Ten obszar tematu jest określany przez właściwość XMSC_WPM_TOPIC_SPACE .

Jeśli właściwość nie jest ustawiona dla miejsca docelowego, zakłada się, że kolejka lub powiązany obszar tematu istnieją w magistrali integracji usług, z którą łączy się aplikacja.

Domyślnie właściwość nie jest ustawiona.

XMSC_WPM_CONNECTION_PROTOCOL

Typ danych:
System.Int32

Właściwość:
Połączenie

Protokół komunikacyjny używany na potrzeby połączenia z mechanizmem przesyłania komunikatów. Ta właściwość jest tylko do odczytu.

Możliwe wartości właściwości są następujące:

Wartość	Znaczenie
XMSC_WPM_CP_HTTP	Połączenie korzysta z protokołu HTTP przez TCP/IP.
XMSC_WPM_CP_TCP	Połączenie korzysta z protokołu TCP/IP.

XMSC_WPM_CONNECTION_BLISKOŚĆ

Typ danych:
System.Int32

Właściwość:
ConnectionFactory

Ustawienie bliskości połączeń na potrzeby nawiązywania połączenia. Ta właściwość określa, w jaki sposób mechanizm przesyłania komunikatów, z którym łączy się aplikacja, musi należeć do serwera startowego.

Poprawne wartości właściwości są następujące:

Poprawna wartość	Ustawienie bliskości połączenia
XMSC_WPM_CONNECTION_PROXIMITY_BUS	Magistrala
XMSC_WPM_CONNECTION_PROXIMITY_CLUSTER	Klaster
XMSC_WPM_CONNECTION_PROXIMITY_HOST	Host
XMSC_WPM_CONNECTION_PROXIMITY_SERVER	Serwer

Wartość domyślna to XMSC_WPM_CONNECTION_PROXIMITY_BUS.

XMSC_WPM_DUR_SUB_HOME

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Nazwa używana w identyfikatorze URI:
Strona główna durableSubscription

Nazwa mechanizmu przesyłania komunikatów zarządzającego wszystkimi trwałymi subskrypcjami połączeń lub miejsc docelowych. Komunikaty, które mają zostać dostarczone do trwałych subskrybentów, są zapisywane w punkcie publikacji tego samego mechanizmu przesyłania komunikatów.

W przypadku połączenia przed utworzeniem trwałego subskrybenta, który korzysta z połączenia, musi zostać określony dom subskrypcji trwałej. Każda wartość określona dla miejsca docelowego przesłania wartość określoną dla połączenia.

Domyślnie właściwość nie jest ustawiona.

Ta właściwość jest odpowiednia tylko w domenie Publikowanie/subskrypcja .

XMSC_WPM_HOST_NAME

Typ danych:
Łańcuch

Właściwość:
Połączenie

Położenie mechanizmu przesyłania komunikatów, z którym połączona jest aplikacja (nazwa hosta lub adres IP systemu). Ta właściwość jest tylko do odczytu.

XMSC_WPM_LOCAL_ADDRESS

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

W przypadku połączenia z magistralą integracji usług: ta właściwość określa używaną wartość interfejsu sieci lokalnej albo portu lokalnego (lub zakresu portów lokalnych) bądź obie te wartości.

Wartością właściwości jest łańcuch o następującym formacie:

[*nazwa_hosta*] [(*low_port*) [,*high_port*])]

Znaczenia zmiennych są następujące:

nazwa_hosta
Nazwa hosta lub adres IP lokalnego interfejsu sieciowego, który ma być używany dla połączenia.

Podanie tych informacji jest konieczne tylko wtedy, gdy system, na którym działa aplikacja, ma dwa lub więcej interfejsów sieciowych i użytkownik musi być w stanie określić, który interfejs musi być używany dla połączenia. Jeśli system ma tylko jeden interfejs sieciowy, tylko ten interfejs może być używany. Jeśli system ma dwa lub więcej interfejsów sieciowych i nie określono interfejsu, który musi być używany, interfejs jest wybierany losowo.

low_port
Numer portu lokalnego, który ma być używany dla połączenia.

Jeśli podano również wartość *high_port* , wartość *low_port* jest interpretowana jako najniższy numer portu w zakresie numerów portów.

wysoki_port
Najwyższy numer portu w zakresie numerów portów. Jeden z portów w podanym zakresie musi być używany dla połączenia.

Poniżej przedstawiono kilka przykładów poprawnych wartości właściwości:

JUPITER
9.20.4.98
JUPITER (1000)
9.20.4.98(1000,2000)
(1000)
(1000,2000)

Domyślnie właściwość nie jest ustawiona.

XMSC_WPM_ME_NAME

Typ danych:
Łańcuch

Właściwość:
Połączenie

Nazwa mechanizmu przesyłania komunikatów, z którym połączona jest aplikacja. Ta właściwość jest tylko do odczytu.

XMSC_WPM_NON_PERSISTENT_MAP

Typ danych:
System.Int32

Właściwość:
ConnectionFactory

Poziom niezawodności komunikatów nietrwałych wysyłanych za pośrednictwem połączenia.

Poprawne wartości właściwości są następujące:

Poprawna wartość	Poziom niezawodności
XMSC_WPM_MAPPING_AS_DESTINATION	Określony przez domyślny poziom niezawodności określony dla kolejki lub obszaru tematu w magistrali integracji usług.
XMSC_WPM_MAPPING_BEST_EFFORT_NON_ Trwałe	Optymalny nietrwały wysiłek
SPECYFIKACJA_ODWZOROWANIA_Z_XMSC_Z_XMSC__ Trwałe	Ekspresowa nietrwała
XMSC_WPM_MAPPING_RELIABLE_NON_ Trwałe	Niezawodny nietrwały
XMSC_WPM_MAPPING_RELIABLE_PERSISTENT	Niezawodny trwały
XMSC_WPM_MAPPING_ASSURED_PERSISTENT	Zapewniony trwały

Wartością domyślną jest XMSC_WPM_MAPPING_EXPRESS_NON_PERSISTENT.

XMSC_WPM_PERSISTENT_MAP

Typ danych:
System.Int32

Właściwość:
ConnectionFactory

Poziom niezawodności komunikatów trwałych wysyłanych za pośrednictwem połączenia.

Poprawne wartości właściwości są następujące:

Poprawna wartość	Poziom niezawodności
XMSC_WPM_MAPPING_AS_DESTINATION	Określony przez domyślny poziom niezawodności określony dla kolejki lub obszaru tematu w magistrali integracji usług.
XMSC_WPM_MAPPING_BEST_EFFORT_NON_ Trwałe	Optymalny nietrwały wysiłek
SPECYFIKACJA_ODWZOROWANIA_Z_XMSC_Z_XMSC__ Trwałe	Ekspresowa nietrwała
XMSC_WPM_MAPPING_RELIABLE_NON_ Trwałe	Niezawodny nietrwały
XMSC_WPM_MAPPING_RELIABLE_PERSISTENT	Niezawodny trwały
XMSC_WPM_MAPPING_ASSURED_PERSISTENT	Zapewniony trwały

Wartością domyślną jest XMSC_WPM_MAPPING_RELIABLE_PERSISTENT.

PORT XMSC_WPM_PORT

Typ danych:
System.Int32

Właściwość:
Połączenie

Numer portu, na którym nasłuchuje mechanizm przesyłania komunikatów, z którym aplikacja nawiązała połączenie. Ta właściwość jest tylko do odczytu.

PUNKTY KOŃCOWE XMSC_WPM_PROVIDER_ENDPOINTS

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Jeden adres punktu końcowego serwera startowego lub sekwencja wielu adresów. Adresy punktów końcowych są rozdzielane przecinkami.

Serwer startowy to serwer aplikacji, który jest odpowiedzialny za wybór mechanizmu przesyłania komunikatów, z którym aplikacja nawiązuje połączenie. Adres punktu końcowego serwera startowego ma następujący format:

nazwa_hosta:*numer_portu*:*nazwa_łańcucha*

Znaczenia komponentów adresu punktu końcowego są następujące:

nazwa_hosta
Nazwa hosta lub adres IP systemu, na którym znajduje się serwer startowy. Jeśli nie zostanie podana nazwa hosta lub adres IP, wartością domyślną jest `localhost`.

numer_portu
Numer portu, na którym serwer startowy nasłuchuje nadchodzących żądań. Jeśli nie zostanie podany numer portu, wartością domyślną jest 7276.

nazwa_łańcucha
Nazwa startowego łańcucha transportowego używanego przez serwer startowy. Poprawne wartości to:

Poprawna wartość	Nazwa startowego łańcucha transportowego
XMSC_WPM_BOOTSTRAP_HTTP	Przesyłanie komunikatów BootstrapTunneled
XMSC_WPM_BOOTSTRAP_HTTPS	BootstrapTunneledSecureMessaging
XMSC_WPM_BOOTSTRAP_SSL	Przesyłanie komunikatów programu BootstrapSecure
XMSC_WPM_BOOTSTRAP_TCP	Przesyłanie komunikatów BootstrapBasic

Jeśli nie zostanie podana żadna nazwa, wartością domyślną jest XMSC_WPM_BOOTSTRAP_TCP.

Jeśli nie zostanie podany adres punktu końcowego, wartością domyślną jest localhost:7276:BootstrapBasicMessaging.

XMSC_WPM_SSL_CIPHER_SUITE

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Nazwa pakietu CipherSuite , która ma być używana przez połączenie TLS do mechanizmu przesyłania komunikatów produktu WebSphere Usługa Integration Bus . Na podstawie podanego zestawu algorytmów szyfrowania jest określany protokół używany do negocjowania bezpiecznego połączenia.

Tabela 35. Opcje CipherSuite dla połączenia z mechanizmem przesyłania komunikatów produktu WebSphere Usługa Integration Bus

zestaw algorytmów szyfrowania	Używany protokół
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

Uwagi:

1. Opcje TLS_RSA_WITH_AES_128_CBC_SHA i TLS_RSA_WITH_AES_256_CBC_SHA CipherSuites są obsługiwane tylko w systemach Windows lub Solaris . (Jest to podyktowane przez pakiet GSKit).
2. Parametr TLS_RSA_WITH_3DES_EDE_CBC_SHA jest nieaktualny. Jednak może być ona nadal używana do przesyłania do 32 GB danych, zanim połączenie zostanie zakończone z błędem AMQ9288. Aby uniknąć tego błędu, należy unikać używania potrójnego algorytmu szyfrowania DES lub włączyć resetowanie klucza tajnego podczas korzystania z tej specyfikacji CipherSpec.

Dla tej właściwości nie ma wartości domyślnej. Aby używać protokołu SSL lub TLS, należy określić wartość tej właściwości, w przeciwnym razie aplikacja nie będzie mogła pomyślnie nawiązać połączenia z serwerem.

XMSC_WPM_SSL_FIPS_REQUIRED

Typ danych:
wartość boolowska

Właściwość:
ConnectionFactory

Wartość tej właściwości określa, czy aplikacja może lub nie może używać zestawów algorytmów szyfrowania niezgodnych ze standardem FIPS. Jeśli ta właściwość jest ustawiona na wartość true, dla połączenia klient-serwer używane są tylko algorytmy FIPS.Ustawienie wartości tej właściwości na TRUE uniemożliwia stosowanie zestawów algorytmów szyfrowania zgodnych ze standardem innym niż FIPS.

Domyślnie właściwość jest ustawiona na FALSE (jest to tryb FIPS wyłączony).

XMSC_WPM_SSL_KEY_REPOSITORY

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Ścieżka do pliku, który jest plikiem kluczy, w którym znajdują się klucze publiczne lub prywatne, które mają być używane w połączeniu chronionym.

Ustawienie właściwości pliku kluczy na wartość specjalną XMSC_WPM_SSL_MS_CERTIFICATE_STORE określa, że używana jest baza danych kluczy Microsoft Windows . Korzystając z bazy danych kluczy Microsoft Windows , która znajduje się w obszarze **Panel sterowania** > **Opcje internetowe** > **Treść** > **Certyfikaty**, usuwa potrzebę oddzielnej bazy danych pliku kluczy. Użycie tej stałej na platformie Windows x64 i innych platformach nie jest dozwolone.

Domyślnie właściwość nie jest ustawiona.

XMSC_WPM_SSL_KEYRING_LABEL,

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Certyfikat używany podczas uwierzytelniania na serwerze. Jeśli nie zostanie podana żadna wartość, zostanie użyty certyfikat domyślny.

Domyślnie właściwość nie jest ustawiona.

XMSC_WPM_SSL_KEYRING_PW

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Hasło dla pliku kluczy.

Ta właściwość może być używana jako alternatywa dla komendy XMSC_WPM_SSL_KEYRING_STASH_FILE w celu skonfigurowania hasła dla pliku kluczy.

Domyślnie właściwość nie jest ustawiona.

XMSC_WPM_SSL_KEYRING_STASH_FILE

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Nazwa pliku binarnego zawierającego hasło do pliku repozytorium kluczy.

Ta właściwość może być używana jako alternatywa dla komendy XMSC_WPM_SSL_KEYRING_PW w celu skonfigurowania hasła dla pliku kluczy.

Domyślnie właściwość nie jest ustawiona.

XMSC_WPM_TARGET_GROUP

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Nazwa grupy docelowej mechanizmów przesyłania komunikatów. Rodzaj grupy docelowej jest określany przez właściwość XMSC_WPM_TARGET_TYPE .

Tę właściwość należy ustawić, jeśli mechanizm przesyłania komunikatów ma zostać ograniczony do podgrupy mechanizmów przesyłania komunikatów w magistrali integracji usług. Jeśli aplikacja ma mieć możliwość łączenia się z dowolnym mechanizmem przesyłania komunikatów na magistrali integracji usług, nie należy ustawiać tej właściwości.

Domyślnie właściwość nie jest ustawiona.

XMSC_WPM_TARGET_ISTOTNOŚĆ

Typ danych:
System.Int32

Właściwość:
ConnectionFactory

Znaczenie grupy docelowej mechanizmów przesyłania komunikatów.

Poprawne wartości właściwości są następujące:

Poprawna wartość	Znaczenie
XMSC_WPM_TARGET_SIGNIFICANCE_ Preferowane	Mechanizm przesyłania komunikatów w grupie docelowej jest wybierany, jeśli dostępny jest jeden z nich. W przeciwnym razie wybierany jest mechanizm przesyłania komunikatów spoza grupy docelowej, pod warunkiem że znajduje się on w tej samej magistrali integracji usług.
XMSC_WPM_TARGET_SIGNIFICANCE_ WYMAGANE	Wybrany mechanizm przesyłania komunikatów musi należeć do grupy docelowej. Jeśli mechanizm przesyłania komunikatów w grupie docelowej nie jest dostępny, proces połączenia nie powiedzie się.

Wartością domyślną właściwości jest XMSC_WPM_TARGET_SIGNIFICANCE_PREFERRED.

XMSC_WPM_TARGET_TRANSPORT_CHAIN

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Nazwa łańcucha transportowego danych przychodzących, który musi być używany przez aplikację do nawiązywania połączenia z mechanizmem przesyłania komunikatów.

Wartością właściwości może być nazwa dowolnego łańcucha transportowego danych przychodzących, który jest dostępny na serwerze aplikacji, który udostępnia mechanizm przesyłania komunikatów. Następująca stała nazwana jest udostępniana dla jednego z predefiniowanych łańcuchów transportu przychodzącego:

Stała nazwana	Nazwa łańcucha transportowego
XMSC_WPM_TARGET_TRANSPORT_CHAIN_BASIC	Przesyłanie komunikatów InboundBasic

Domyślna wartość właściwości to XMSC_WPM_TARGET_TRANSPORT_CHAIN_BASIC.

XMSC_WPM_TARGET_TYPE

Typ danych:
System.Int32

Właściwość:
ConnectionFactory

Typ grupy docelowej mechanizmów przesyłania komunikatów. Ta właściwość służy do określania rodzaju grupy docelowej identyfikowanej przez właściwość XMSC_WPM_TARGET_GROUP .

Poprawne wartości właściwości są następujące:

Poprawna wartość	Znaczenie
XMSC_WPM_TARGET_TYPE_BUSMEMBER	Nazwa grupy docelowej to nazwa elementu magistrali. Grupa docelowa to wszystkie mechanizmy przesyłania komunikatów znajdujące się w elemencie magistrali.
XMSC_WPM_TARGET_TYPE_CUSTOM	Nazwa grupy docelowej jest nazwą grupy mechanizmów przesyłania komunikatów zdefiniowanej przez użytkownika. Grupa docelowa to wszystkie mechanizmy przesyłania komunikatów, które są zarejestrowane w grupie zdefiniowanej przez użytkownika.
XMSC_WPM_TARGET_TYPE_ME	Nazwa grupy docelowej to nazwa mechanizmu przesyłania komunikatów. Grupa docelowa jest podanym mechanizmem przesyłania komunikatów.

Domyślnie właściwość nie jest ustawiona.

XMSC_WPM_TEMP_Q_PREFIX

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Przedrostek używany do tworzenia nazwy kolejki tymczasowej, która jest tworzona na magistrali integracji usług, gdy aplikacja tworzy kolejkę tymczasową XMS . Przedrostek może zawierać do 12 znaków.

Nazwa kolejki tymczasowej zaczyna się od znaków "_Q", po których następuje przedrostek. Pozostała część nazwy składa się z wygenerowanych przez system znaków.

Domyślnie właściwość ta nie jest ustawiona, co oznacza, że nazwa kolejki tymczasowej nie ma przedrostka.

Ta właściwość jest odpowiednia tylko w domenie punkt z punktem .

XMSC_WPM_TEMP_TOPIC_PREFIX

Typ danych:
Łańcuch

Właściwość:
ConnectionFactory

Przedrostek używany do generowania nazwy tematu tymczasowego tworzonego przez aplikację. Przedrostek może zawierać do 12 znaków.

Nazwa tematu tymczasowego rozpoczyna się od znaków "_T", po których następuje przedrostek. Pozostała część nazwy składa się z wygenerowanych przez system znaków.

Domyślnie właściwość ta nie jest ustawiona, co oznacza, że nazwa tematu tymczasowego nie ma przedrostka.

Ta właściwość jest odpowiednia tylko w domenie Publikowanie/subskrypcja .

XMSC_WPM_TOPIC_SPACE

Typ danych:
Łańcuch

Właściwość:
Miejsce docelowe

Nazwa używana w identyfikatorze URI:
topicSpace

Nazwa obszaru tematu zawierającego dany temat. Ta właściwość może mieć tylko miejsce docelowe, które jest tematem.

Domyślnie właściwość nie jest ustawiona, co oznacza, że zakładany jest domyślny obszar tematu.

Ta właściwość jest odpowiednia tylko w domenie Publikowanie/subskrypcja .

Uwagi

Niniejsza publikacja została opracowana z myślą o produktach i usługach oferowanych w Stanach Zjednoczonych.

IBM może nie oferować w innych krajach produktów, usług lub opcji omawianych w tej publikacji. Informacje o produktach i usługach dostępnych w danym kraju można uzyskać od lokalnego przedstawiciela IBM. Odwołanie do produktu, programu lub usługi IBM nie oznacza, że można użyć wyłącznie tego produktu, programu lub usługi IBM. Zamiast nich można zastosować ich odpowiednik funkcjonalny pod warunkiem, że nie narusza to praw własności intelektualnej firmy IBM. Jednakże cała odpowiedzialność za ocenę przydatności i sprawdzenie działania produktu, programu lub usługi pochodzących od producenta innego niż IBM spoczywa na użytkowniku.

IBM może posiadać patenty lub złożone wnioski patentowe na towary i usługi, o których mowa w niniejszej publikacji. Używanie tego dokumentu nie daje żadnych praw do tych patentów. Pisemne zapytania w sprawie licencji można przesyłać na adres:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Zapytania w sprawie licencji dotyczących informacji kodowanych przy użyciu dwubajtowych zestawów znaków (DBCS) należy kierować do lokalnych działów IBM Intellectual Property Department lub zgłaszać na piśmie pod adresem:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

Poniższy akapit nie obowiązuje w Wielkiej Brytanii, a także w innych krajach, w których jego treść pozostaje w sprzeczności z przepisami prawa miejscowego: INTERNATIONAL BUSINESS MACHINES CORPORATION DOSTARCZA TĘ PUBLIKACJĘ W STANIE, W JAKIM SIĘ ZNAJDUJE ("AS IS"), BEZ JAKICHKOLWIEK GWARANCJI (RĘKOJMIĘ RÓWNIEŻ WYŁĄCZA SIĘ), WYRAŹNYCH LUB DOMNIEMANYCH, A W SZCZEGÓLNOŚCI DOMNIEMANYCH GWARANCJI PRZYDATNOŚCI HANDLOWEJ, PRZYDATNOŚCI DO OKREŚLONEGO CELU ORAZ GWARANCJI, ŻE PUBLIKACJA TA NIE NARUSZA PRAW OSÓB TRZECICH. Ustawodawstwa niektórych krajów nie dopuszczają zastrzeżeń dotyczących gwarancji wyraźnych lub domniemanych w odniesieniu do pewnych transakcji; w takiej sytuacji powyższe zdanie nie ma zastosowania.

Informacje zawarte w niniejszej publikacji mogą zawierać nieścisłości techniczne lub błędy typograficzne. Informacje te są okresowo aktualizowane, a zmiany te zostaną uwzględnione w kolejnych wydaniach tej publikacji. IBM zastrzega sobie prawo do wprowadzania ulepszeń i/lub zmian w produktach i/lub programach opisanych w tej publikacji w dowolnym czasie, bez wcześniejszego powiadomienia.

Wszelkie wzmianki w tej publikacji na temat stron internetowych innych podmiotów zostały wprowadzone wyłącznie dla wygody użytkowników i w żadnym wypadku nie stanowią zachęty do ich odwiedzania. Materiały dostępne na tych stronach nie są częścią materiałów opracowanych dla tego produktu IBM, a użytkownik korzysta z nich na własną odpowiedzialność.

IBM ma prawo do używania i rozpowszechniania informacji przysłanych przez użytkownika w dowolny sposób, jaki uzna za właściwy, bez żadnych zobowiązań wobec ich autora.

Licencjobiorcy tego programu, którzy chcieliby uzyskać informacje na temat programu w celu: (i) wdrożenia wymiany informacji między niezależnie utworzonymi programami i innymi programami (łącznie

z tym opisywanym) oraz (ii) wspólnego wykorzystywania wymienianych informacji, powinni skontaktować się z:

IBM Corporation
Koordynator współdziałania z oprogramowaniem, Dział 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Informacje takie mogą być udostępnione, o ile spełnione zostaną odpowiednie warunki, w tym, w niektórych przypadkach, zostanie uiszczona stosowna opłata.

Licencjonowany program opisany w niniejszej publikacji oraz wszystkie inne licencjonowane materiały dostępne dla tego programu są dostarczane przez IBM na warunkach określonych w Umowie IBM z Klientem, Międzynarodowej Umowie Licencyjnej IBM na Program lub w innych podobnych umowach zawartych między IBM i użytkownikami.

Wszelkie dane dotyczące wydajności zostały zebrane w kontrolowanym środowisku. W związku z tym rezultaty uzyskane w innych środowiskach operacyjnych mogą się znacząco różnić. Niektóre pomiary mogły być dokonywane na systemach będących w fazie rozwoju i nie ma gwarancji, że pomiary wykonane na ogólnie dostępnych systemach dadzą takie same wyniki. Niektóre z pomiarów mogły być estymowane przez ekstrapolację. Rzeczywiste wyniki mogą być inne. Użytkownicy powinni we własnym zakresie sprawdzić odpowiednie dane dla ich środowiska.

Informacje dotyczące produktów innych niż produkty IBM pochodzą od dostawców tych produktów, z opublikowanych przez nich zapowiedzi lub innych powszechnie dostępnych źródeł. Firma IBM nie testowała tych produktów i nie może potwierdzić dokładności pomiarów wydajności, kompatybilności ani żadnych innych danych związanych z tymi produktami. Pytania dotyczące możliwości produktów innych podmiotów należy kierować do dostawców tych produktów.

Wszelkie stwierdzenia dotyczące przyszłych kierunków rozwoju i zamierzeń IBM mogą zostać zmienione lub wycofane bez powiadomienia.

Publikacja ta zawiera przykładowe dane i raporty używane w codziennych operacjach działalności gospodarczej. W celu kompleksowego ich zilustrowania podane przykłady zawierają nazwiska osób prywatnych, nazwy przedsiębiorstw oraz nazwy produktów. Wszystkie te nazwy/nazwiska są fikcyjne i jakiekolwiek podobieństwo do istniejących nazw/nazwisk i adresów jest całkowicie przypadkowe.

LICENCJA W ZAKRESIE PRAW AUTORSKICH:

Niniejsza publikacja zawiera przykładowe aplikacje w kodzie źródłowym, ilustrujące techniki programowania w różnych systemach operacyjnych. Użytkownik może kopiować, modyfikować i dystrybuować te programy przykładowe w dowolnej formie bez uiszczania opłat na rzecz IBM, w celu projektowania, używania, sprzedaży lub dystrybucji aplikacji zgodnych z aplikacyjnym interfejsem programistycznym dla tego systemu operacyjnego, dla którego napisane zostały programy przykładowe. Programy przykładowe nie zostały gruntownie przetestowane. IBM nie może zatem gwarantować ani sugerować niezawodności, użyteczności i funkcjonalności tych programów.

W przypadku przeglądania niniejszych informacji w formie elektronicznej, zdjęcia i kolorowe ilustracje mogą nie być wyświetlane.

Informacje dotyczące interfejsu programistycznego

Informacje dotyczące interfejsu programistycznego, o ile są udostępniane, mają być pomocne podczas tworzenia oprogramowania aplikacji do użytku z tym programem.

Ten podręcznik zawiera informacje na temat planowanych interfejsów programistycznych, które umożliwiają klientom pisanie programów w celu uzyskania dostępu do usług produktu WebSphere MQ.

Informacje te mogą również zawierać informacje na temat diagnostyki, modyfikacji i strojenia. Tego typu informacje są udostępniane jako pomoc przy debugowaniu aplikacji.

Ważne: Informacji na temat diagnostyki, modyfikacji i strojenia nie należy używać jako interfejsu programistycznego, ponieważ może on ulec zmianie.

Znaki towarowe

IBM, logo IBM , ibm.com, są znakami towarowymi IBM Corporation, zarejestrowanymi w wielu systemach prawnych na całym świecie. Aktualna lista znaków towarowych IBM jest dostępna w serwisie WWW, w sekcji "Copyright and trademark information"(Informacje o prawach autorskich i znakach towarowych), pod adresemwww.ibm.com/legal/copytrade.shtml. Nazwy innych produktów lub usług mogą być znakami towarowymi IBM lub innych podmiotów.

Microsoft oraz Windows są znakami towarowymi Microsoft Corporation w Stanach Zjednoczonych i/lub w innych krajach.

UNIX jest zastrzeżonym znakiem towarowym The Open Group w Stanach Zjednoczonych i/lub w innych krajach.

Linux® jest zastrzeżonym znakiem towarowym Linusa Torvaldsa w Stanach Zjednoczonych i/lub w innych krajach.

Ten produkt zawiera oprogramowanie opracowane przez Eclipse Project (http://www.eclipse.org/).

Java oraz wszystkie znaki towarowe i logo dotyczące języka Java są znakami towarowymi lub zastrzeżonymi znakami towarowymi Oracle i/lub przedsiębiorstw afiliowanych Oracle.

Numer pozycji:

(1P) P/N: