

8.0

IBM Message Service Client for .NET

IBM

附註

使用本資訊及其支援的產品之前，請先閱讀第 219 頁的『[注意事項](#)』中的資訊。

除非新版中另有指示，否則此版本適用於 IBM® MQ 8.0.0 版及所有後續版本與修訂版。

當您將資訊傳送至 IBM 時，您授與 IBM 非專屬權利，以任何其認為適當的方式使用或散佈資訊，而無需對您負責。

© Copyright International Business Machines Corporation 2007, 2023.

目錄

IBM Message Service Client for .NET 簡介.....	5
IBM Message Service Client for .NET 簡介.....	5
傳訊樣式.....	6
XMS 物件模型.....	6
物件的屬性和內容.....	7
受管理物件.....	8
XMS 訊息模型.....	9
防止應用程式使用較新的 XMS 版本.....	9
設定傳訊伺服器環境.....	10
為連接至 WebSphere MQ 佇列管理程式的應用程式配置佇列管理程式及分配管理系統.....	10
為使用分配管理系統即時連線的應用程式配置分配管理系統.....	12
為連接至 WebSphere 服務 Integration Bus 的應用程式配置 服務整合匯流排.....	12
使用安裝精靈來安裝 Message Service Client for .NET.....	13
XMS 應用程式連接至 WebSphere MQ 的必要條件.....	15
使用 XMS 範例應用程式.....	16
範例應用程式.....	16
執行範例應用程式.....	17
建置 .NET 範例應用程式.....	18
開發 XMS 應用程式.....	18
撰寫 XMS 應用程式.....	18
撰寫 XMS .NET 應用程式.....	38
使用受管理物件.....	43
保護 XMS 應用程式的通訊安全.....	55
XMS 訊息.....	58
疑難排解.....	69
.NET 應用程式的追蹤配置.....	70
.NET 應用程式的 FFDC 配置.....	73
疑難排解提示.....	73
Message Service Client for .NET 參照.....	74
.NET 介面.....	74
XMS 物件的內容.....	159
注意事項.....	219
程式設計介面資訊.....	220
商標.....	220

IBM Message Service Client for .NET 簡介

IBM Message Service Client for .NET 提供稱為 XMS 的應用程式設計介面 (API)，其具有與 Java™ Message Service (JMS) 相同的介面集 API。IBM Message Service Client for .NET 包含 XMS 的完全受管理實作，可供任何符合 .NET 標準的語言使用。

XMS 支援：

- 點對點 (point-to-point) 傳訊
- 發佈/訂閱 傳訊
- 同步訊息遞送
- 非同步訊息遞送

XMS 應用程式可以與下列類型的應用程式交換訊息：

- XMS 應用程式
- IBM MQ classes for JMS 應用程式
- 原生 IBM MQ 應用程式
- 使用 IBM MQ 預設傳訊提供者的 JMS 應用程式

XMS 應用程式可以連接及使用下列任何傳訊伺服器的資源：

IBM MQ 佇列管理程式 (queue manager)

應用程式可以在連結或用戶端模式下連接。

WebSphere Application Server 服務 Integration Bus

應用程式可以使用直接 TCP/IP 連線，也可以使用透過 TCP/IP 的 HTTP。

IBM Integration Bus

使用 WebSphere MQ 即時傳輸在應用程式與分配管理系統之間傳輸訊息。可以使用 WebSphere MQ 多重播送傳輸將訊息遞送至應用程式。

透過連接至 IBM MQ 佇列管理程式，XMS 應用程式可以使用 WebSphere MQ 企業傳輸來與 IBM Integration Bus 進行通訊。或者，XMS 應用程式可以透過連接至 WebSphere MQ 來發佈及訂閱。

相關概念

[第 6 頁的『傳訊樣式』](#)

[第 6 頁的『XMS 物件模型』](#)

XMS API 是物件導向介面。XMS 物件模型基於 JMS 1.1 物件模型。

[第 9 頁的『XMS 訊息模型』](#)

XMS 訊息模型與 WebSphere MQ for JMS 類別 訊息模型相同。

IBM Message Service Client for .NET 簡介

IBM Message Service Client for .NET 提供稱為 XMS 的應用程式設計介面 (API)，其具有與 Java Message Service (JMS) 相同的介面集 API。IBM Message Service Client for .NET 包含 XMS 的完全受管理實作，可供任何符合 .NET 標準的語言使用。

XMS 支援：

- 點對點 (point-to-point) 傳訊
- 發佈/訂閱 傳訊
- 同步訊息遞送
- 非同步訊息遞送

XMS 應用程式可以與下列類型的應用程式交換訊息：

- XMS 應用程式

- IBM MQ classes for JMS 應用程式
- 原生 IBM MQ 應用程式
- 使用 IBM MQ 預設傳訊提供者的 JMS 應用程式

XMS 應用程式可以連接及使用下列任何傳訊伺服器的資源：

IBM MQ 佇列管理程式 (queue manager)

應用程式可以在連結或用戶端模式下連接。

WebSphere Application Server 服務 Integration Bus

應用程式可以使用直接 TCP/IP 連線，也可以使用透過 TCP/IP 的 HTTP。

IBM Integration Bus

使用 WebSphere MQ 即時傳輸在應用程式與分配管理系統之間傳輸訊息。可以使用 WebSphere MQ 多重播送傳輸將訊息遞送至應用程式。

透過連接至 IBM MQ 佇列管理程式，XMS 應用程式可以使用 WebSphere MQ 企業傳輸來與 IBM Integration Bus 進行通訊。或者，XMS 應用程式可以透過連接至 WebSphere MQ 來發佈及訂閱。

相關概念

[第 6 頁的『傳訊樣式』](#)

[第 6 頁的『XMS 物件模型』](#)

XMS API 是物件導向介面。XMS 物件模型基於 JMS 1.1 物件模型。

[第 9 頁的『XMS 訊息模型』](#)

XMS 訊息模型與 WebSphere MQ for JMS 類別 訊息模型相同。

傳訊樣式

XMS 支援 點對點 (point-to-point) 和 發佈/訂閱 傳訊樣式。

傳訊樣式也稱為傳訊網域。

點對點 (point-to-point) 傳訊

點對點 (point-to-point) 傳訊的一般形式會使用佇列作業。在最簡單的情況下，應用程式會隱含或明確地識別目的地佇列，以將訊息傳送至另一個應用程式。基礎傳訊及佇列作業系統會從傳送端應用程式接收訊息，並將訊息遞送至其目的地佇列。然後，接收端應用程式可以從佇列擷取訊息。

如果基礎傳訊及佇列作業系統包含 IBM Integration Bus，IBM Integration Bus 可能會將訊息及遞送訊息副本抄寫至不同的佇列。因此，多個應用程式可以接收訊息。IBM Integration Bus 也可以轉換訊息並將資料新增至其中。

點對點 (point-to-point) 傳訊的主要性質是應用程式在傳送訊息時將訊息放到本端佇列中。基礎傳訊及佇列作業系統會決定訊息傳送至哪個目的地佇列。接收端應用程式會從目的地佇列擷取訊息。

發佈/訂閱 傳訊

在發佈/訂閱傳訊中，有兩種類型的應用程式：發佈者和訂閱者。

發佈者以發佈訊息的形式提供資訊。當發佈者發佈訊息時，它會指定主題，以識別訊息內資訊的主旨。

訂閱者是已發佈資訊的消費者。訂閱者透過建立訂閱來指定其感興趣的主題。

發佈/訂閱系統會從發佈者接收發佈，並從訂閱者接收訂閱。它會將發佈遞送給訂閱者。訂閱者只會接收其訂閱的那些主題的發佈。

發佈/訂閱傳訊的主要性質是發佈者在發佈訊息時識別主題。它不會識別訂閱者。如果在沒有訂閱者的主題上發佈訊息，則沒有應用程式會收到訊息。

應用程式可以同時是發佈者和訂閱者。

XMS 物件模型

XMS API 是物件導向介面。XMS 物件模型基於 JMS 1.1 物件模型。

下列清單彙總主要 XMS 類別或物件類型:

ConnectionFactory

ConnectionFactory 物件會封裝連線的一組參數。應用程式使用 ConnectionFactory 來建立連線。應用程式可以在執行時期提供參數，並建立 ConnectionFactory 物件。或者，連線參數可以儲存在受管理物件的儲存庫中。應用程式可以從儲存庫擷取物件，並從中建立 ConnectionFactory 物件。

Connection

Connection 物件封裝從應用程式到傳訊伺服器的作用中連線。應用程式使用連線建立階段作業。

Destination

應用程式使用 Destination 物件來傳送訊息或接收訊息。在發佈/訂閱網域中，Destination 物件會封裝主題，而在點對點 (point-to-point) 網域中，Destination 物件會封裝佇列。應用程式可以提供參數，以在執行時期建立 Destination 物件。或者，它可以從儲存在受管理物件儲存庫中的物件定義建立 Destination 物件。

Session

Session 物件是用於傳送及接收訊息的單一執行緒環境定義。應用程式使用 Session 物件來建立 Message、MessageProducer 及 MessageConsumer 物件。

Message

Message 物件會封裝應用程式使用 MessageProducer 物件傳送或使用 MessageConsumer 物件接收的 Message 物件。

MessageProducer

應用程式使用 MessageProducer 物件將訊息傳送至目的地。

MessageConsumer

應用程式使用 MessageConsumer 物件來接收傳送至目的地的訊息。

第 7 頁的圖 1 顯示這些物件及其關係。此圖顯示 XMS 物件的主要類型: ConnectionFactory、連線、階段作業、MessageProducer、MessageConsumer、訊息及目的地。應用程式使用 Connection Factory 建立連線，使用連線建立階段作業。然後，應用程式可以使用階段作業建立訊息、訊息產生者及訊息消費者。應用程式使用訊息產生者將訊息傳送至目的地，使用訊息消費者接收已傳送至目的地的訊息。

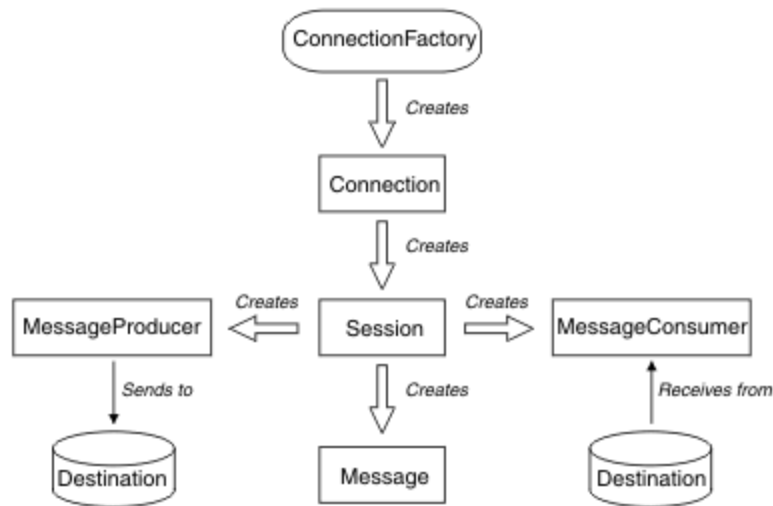


圖 1: XMS 物件及其關係

在 .NET 中，XMS 類別定義為一組 .NET 介面。當您撰寫 XMS .NET 應用程式的程式碼時，只需要宣告的介面。

XMS 物件模型基於 Java 訊息服務規格 1.1 中說明的與網域無關介面。未提供網域特定的類別，例如 Topic、TopicPublisher 及 TopicSubscriber。

物件的屬性和內容

XMS 物件可以具有以不同方式實作的屬性及內容，這些屬性及內容是物件的性質。

屬性

一律呈現並佔用儲存體的物件性質，即使屬性沒有值也是如此。在這方面，屬性類似於固定長度資料結構中的欄位。屬性的一個區別特性是每一個屬性都有自己的方法來設定及取得其值。

內容

物件的內容存在，且僅在設定其值之後才會佔用儲存體。在設定內容值之後，無法刪除內容或回復其儲存體。您可以變更其值。XMS 提供一組一般方法來設定及取得內容值。

相關概念

XMS 初始類型

XMS 提供八個 Java 初始類型 (byte、short、int、long、float、double、char 及 boolean) 的對等項目。這容許在 XMS 與 JMS 之間交換訊息，而不會失去或毀損資料。

將內容值從一個資料類型隱含轉換為另一個資料類型

當應用程式取得內容的值時，XMS 可以將該值轉換為另一個資料類型。許多規則會控管支援哪些轉換，以及 XMS 如何執行轉換。

相關參考

應用程式資料元素的資料類型

為了確保 XMS 應用程式可以與 WebSphere MQ for JMS 類別 應用程式交換訊息，這兩個應用程式必須能夠以相同方式解譯訊息內文中的應用程式資料。

受管理物件

使用受管理物件，您可以管理要從中央儲存庫管理的用戶端應用程式所使用的連線設定。應用程式會從中央儲存庫擷取物件定義，並使用它們來建立 `ConnectionFactory` 及 `Destination` 物件。使用受管理物件，您可以將應用程式與它們在執行時期使用的資源取消連結。

例如，可以使用參照測試環境中一組連線及目的地的受管理物件來撰寫及測試 XMS 應用程式。部署應用程式時，可以變更受管理物件，以將應用程式配置為參照正式作業環境中的連線及目的地。

XMS 支援兩種類型的受管理物件：

- `ConnectionFactory` 物件，供應用程式用來建立伺服器的起始連線。
- `Destination` 物件，由應用程式用來指定要傳送之訊息的目的地，以及要接收之訊息的來源。目的地是伺服器上應用程式所連接的主題或佇列。

管理工具 **JMSAdmin** 隨 WebSphere MQ 一起提供。它用來在受管理物件的中央儲存庫中建立及管理 的受管理物件。

WebSphere MQ for JMS 類別 和 XMS 應用程式可以使用儲存庫中的受管理物件。XMS 應用程式可以使用 `ConnectionFactory` 和 `Destination` 物件來連接至 WebSphere MQ 佇列管理程式。管理者可以變更保留在儲存庫中的物件定義，而不會影響應用程式碼。

下圖顯示 XMS 應用程式通常如何使用受管理物件。圖表左側顯示的儲存庫包含使用管理主控台管理的 `ConnectionFactory` 和「目的地」物件定義。圖表右側顯示的 XMS 應用程式會在儲存庫中查閱物件定義，然後在連接至傳訊伺服器時使用這些物件定義。

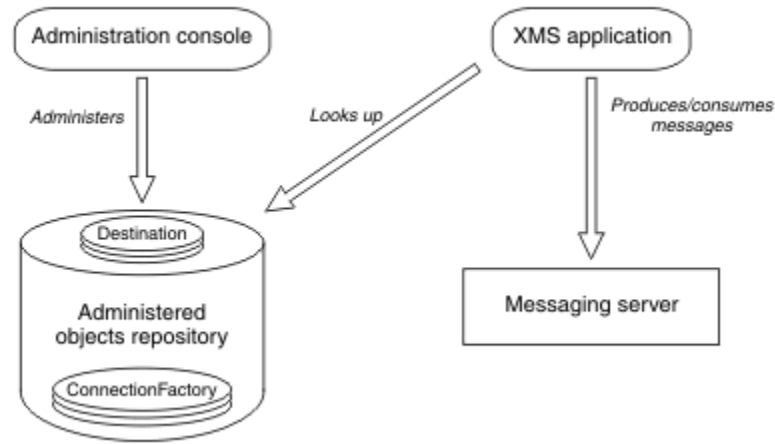


圖 2: XMS 應用程式一般使用受管理物件

相關概念

使用受管理物件

此章節提供受管理物件的相關資訊。XMS 應用程式可以從中央受管理物件儲存庫擷取物件定義，並使用它們來建立 Connection Factory 及目的地。

受管理物件儲存庫的受支援類型

「檔案系統」及 LDAP 受管理物件可以用來連接至 WebSphere MQ 及 WebSphere Application Server，而 COS 命名只能用來連接至 WebSphere Application Server。

相關工作

建立受管理物件

必須使用適當的管理工具來建立 XMS 應用程式建立傳訊伺服器連線所需的 ConnectionFactory 和「目的地」物件定義。

XMS 訊息模型

XMS 訊息模型與 WebSphere MQ for JMS 類別 訊息模型相同。

特別是，XMS 會實作 WebSphere MQ for JMS 類別 所實作的相同訊息標頭欄位及訊息內容：

- JMS 標頭欄位。這些欄位具有以 JMS 字首開頭的名稱。
- JMS 定義的內容。這些欄位具有名稱以字首 JMSX 開頭的內容。
- IBM 定義的內容。這些欄位具有名稱以字首 JMS_IBM_ 開頭的內容。

因此，XMS 應用程式可以與 WebSphere MQ for JMS 類別 應用程式交換訊息。在每一則訊息中，部分標頭欄位及內容由應用程式設定，其他則由 XMS 或 WebSphere MQ for JMS 類別設定。XMS 或 WebSphere MQ for JMS 類別 所設定的部分欄位會在傳送訊息時設定，而其他欄位則會在收到訊息時設定。在適當的情況下，標頭欄位和內容會透過傳訊伺服器隨訊息一起傳播。它們可供接收訊息的任何應用程式使用。

防止應用程式使用較新的 XMS 版本

依預設，當安裝較新的 XMS 版本時，使用舊版的應用程式會自動切換至較新的版本，而不需要重新編譯。

關於這項作業

多重版本共存性特性可確保較新 XMS 版本的安裝不會改寫先前的 XMS 版本。相反地，相似 XMS .NET 組件的多個實例同時存在於「廣域組件快取 (GAC)」中，但具有不同的版本號碼。在內部，GAC 會使用原則檔，將應用程式呼叫遞送至最新版本的 XMS。應用程式無需重新編譯即可執行，並且可以使用較新 XMS .NET 版本中提供的新特性。

不過，如果需要應用程式使用舊版 XMS，請在應用程式配置檔中，將 `publisherpolicy` 屬性設為 `no`。

註：應用程式配置檔是一個檔案，其名稱是由檔案相關的可執行程式名稱所組成，字尾為 `.config`。例如，`text.exe` 的應用程式配置檔將具有名稱 `text.exe.config`。

不過，在任何時候，系統的所有應用程式都會使用相同版本的 XMS .NET。

設定傳訊伺服器環境

此章節說明如何設定傳訊伺服器環境，以容許 XMS 應用程式連接至伺服器。

對於連接至 WebSphere MQ 佇列管理程式的應用程式，需要 WebSphere MQ 用戶端 (或連結模式的佇列管理程式)。

對於使用分配管理系統即時連線的應用程式，目前沒有必要條件。

在執行任何 XMS 應用程式之前，您必須先設定傳訊伺服器環境，包括 XMS 隨附的範例應用程式。

此章節包含下列區段：

- [第 10 頁的『為連接至 WebSphere MQ 佇列管理程式的應用程式配置佇列管理程式及分配管理系統』](#)
- [第 12 頁的『為使用分配管理系統即時連線的應用程式配置分配管理系統』](#)
- [第 12 頁的『為連接至 WebSphere 服務 Integration Bus 的應用程式配置 服務整合匯流排』](#)

相關概念

[JNDI 查閱 Web 服務](#)

若要從 XMS 存取 COS 命名目錄，必須在 WebSphere 服務 Integration Bus 伺服器上部署 JNDI 查閱 Web 服務。此 Web 服務會將 Java 資訊從 COS 命名服務轉換成 XMS 應用程式可以讀取的表單。

[使用 XMS 範例應用程式](#)

使用 XMS 隨附的範例應用程式來驗證安裝及傳訊伺服器設定，並協助您建置自己的應用程式。這些範例提供每一個 API 的一般特性概觀。

相關工作

[使用安裝精靈來安裝 Message Service Client for .NET](#)

安裝使用 InstallShield X/Windows MSI 安裝程式。有兩個設定選項可用，因此您可以選擇完整或自訂安裝。

相關參考

[XMS 應用程式連接至 WebSphere MQ 的必要條件](#)

如果 XMS 應用程式連接至 WebSphere MQ，則部分必要條件適用。

為連接至 WebSphere MQ 佇列管理程式的應用程式配置佇列管理程式及分配管理系統

本節假設您使用 WebSphere MQ 7.0 版或更新版本。您必須先配置佇列管理程式，才能執行連接至 WebSphere MQ 佇列管理程式的應用程式。對於發佈/訂閱應用程式，如果您使用「佇列發佈/訂閱」介面，則需要一些額外的配置。

開始之前

XMS 與 IBM Integration Bus 或 WebSphere Message Broker 6.1 或更新版本搭配使用

在啟動此作業之前，請執行下列步驟：

- 請確定您的應用程式可以存取執行中的佇列管理程式。
- 如果您的應用程式是發佈/訂閱應用程式，並使用排入佇列的發佈/訂閱介面，請確定佇列管理程式上的 "PSMODE" 屬性設為 "ENABLED"。
- 請確定您的應用程式使用其內容已適當設定的 Connection Factory，以連接至佇列管理程式。如果您的應用程式是發佈/訂閱應用程式，請確定已設定適當的 Connection Factory 內容來使用分配管理系統。如需 Connection Factory 內容的相關資訊，請參閱 [第 160 頁的『ConnectionFactory 的內容』](#)。

關於這項作業

您配置佇列管理程式和分配管理系統來執行 XMS 應用程式的方式，與您配置佇列管理程式和排入佇列的發佈/訂閱介面來執行 WebSphere MQ JMS 應用程式的方式相同。下列步驟彙總您需要執行的動作。

程序

1. 在佇列管理程式上，建立應用程式需要的佇列。
如需如何建立佇列的概觀，請參閱 [定義佇列](#)。
如果您的應用程式是發佈/訂閱應用程式，並使用需要存取 WebSphere MQ for JMS 類別系統佇列的「佇列發佈/訂閱」介面，請等到步驟 4a 之後再建立佇列。
2. 授與與應用程式相關聯的使用者 ID 連接至佇列管理程式的權限，以及存取佇列的適當權限。
如需授權的概觀，請參閱 [安全](#)。如果您的應用程式以用戶端模式連接至佇列管理程式，另請參閱 [用戶端及伺服器](#)。
3. 如果您的應用程式以用戶端模式連接至佇列管理程式，請確定已在佇列管理程式中定義伺服器連線通道，且已啟動接聽器。
您不需要對連接至佇列管理程式的每一個應用程式執行此步驟。一個伺服器連線通道定義和一個接聽器可以支援以用戶端模式連接的所有應用程式。
4. 如果您的應用程式是發佈/訂閱應用程式，並使用排入佇列的發佈/訂閱介面，請執行下列步驟。
 - a) 在佇列管理程式上，執行 WebSphere MQ 隨附的 MQSC 指令 Script，以建立 WebSphere MQ for JMS 類別系統佇列。請確定與 WebSphere Message Broker 相關聯的使用者 ID 具有存取佇列的權限。
如需在何處尋找 Script 以及如何執行 Script 的相關資訊，請參閱 [使用 Java](#)。
僅針對佇列管理程式執行一次此步驟。同一組 WebSphere MQ for JMS 類別系統佇列可支援連接至佇列管理程式的所有 XMS 及 WebSphere MQ for JMS 類別應用程式。
 - b) 授與與應用程式相關聯的使用者 ID 存取 WebSphere MQ for JMS 類別系統佇列的權限。
如需使用者 ID 需要哪些權限的相關資訊，請參閱 [使用 JMS](#)。
 - c) 若為 IBM Integration Bus 的分配管理系統，請建立並部署訊息流程，以處理應用程式在其中傳送所發佈訊息的佇列。
基本訊息流程包含用於讀取已發佈訊息的 MQInput 訊息處理節點，以及用於發佈訊息的 Publication 訊息處理節點。
如需如何建立及部署訊息流程的相關資訊，請參閱 [IBM Integration Bus 產品說明文件](#)
如果已在分配管理系統上部署適當的訊息流程，則不需要執行此步驟。

結果

您現在可以啟動應用程式。

相關工作

為使用分配管理系統即時連線的應用程式配置分配管理系統

您必須先配置分配管理系統，才能執行使用即時連線至分配管理系統的應用程式。

為連接至 WebSphere 服務 Integration Bus 的應用程式配置服務整合匯流排

您必須以配置服務整合匯流排來執行使用預設傳訊提供者的 JMS 應用程式的相同方式來配置服務整合匯流排，才能執行連接至 WebSphere 服務 Integration Bus 的應用程式。

使用安裝精靈來安裝 Message Service Client for .NET

安裝使用 InstallShield X/Windows MSI 安裝程式。有兩個設定選項可用，因此您可以選擇完整或自訂安裝。

相關參考

[XMS 應用程式連接至 WebSphere MQ 的必要條件](#)

如果 XMS 應用程式連接至 WebSphere MQ，則部分必要條件適用。

為使用分配管理系統即時連線的應用程式配置分配管理系統

您必須先配置分配管理系統，才能執行使用即時連線至分配管理系統的應用程式。

開始之前

在啟動此作業之前，請執行下列步驟：

- 請確定您的應用程式可以存取執行中的分配管理系統。
- 請確定您的應用程式使用 Connection Factory，其內容已適當地設定，可即時連接至分配管理系統。如需 Connection Factory 內容的相關資訊，請參閱第 160 頁的『ConnectionFactory 的內容』。

關於這項作業

配置分配管理系統以執行 XMS 應用程式的方式，與配置分配管理系統以執行 WebSphere MQ for JMS 類別應用程式的方式相同。下列步驟彙總您需要執行的動作：

程序

1. 建立並部署訊息流程，以從分配管理系統接聽及發佈訊息的 TCP/IP 埠讀取訊息。

您可以利用下列任何一種方式來執行這項作業：

- 建立包含 **Real-timeOptimizedFlow** 訊息處理節點的訊息流程。
- 建立包含 **Real-timeInput** 訊息處理節點及 Publication 訊息處理節點的訊息流程。

您必須將 **Real-timeOptimizedFlow** 或 **Real-timeInput** 節點配置成在用於即時連線的埠上接聽。在 XMS 中，即時連線的預設埠號是 1506。

如果已在分配管理系統上部署適當的訊息流程，則不需要執行此步驟。

2. 如果您需要使用 WebSphere MQ 多重播送傳輸將訊息遞送至應用程式，請配置分配管理系統以啟用多重播送。配置必須啟用多重播送的主題，為那些需要可靠多重播送的主題指定可靠的服務品質。
3. 如果您的應用程式在連接分配管理系統時提供使用者 ID 和密碼，且您想要分配管理系統使用此資訊來鑑別您的應用程式，請配置使用者名稱伺服器及分配管理系統，以進行簡單的 Telnet 型密碼鑑別。

結果

您現在可以啟動應用程式。

相關工作

[為連接至 WebSphere MQ 佇列管理程式的應用程式配置佇列管理程式及分配管理系統](#)

本節假設您使用 WebSphere MQ 7.0 版或更新版本。您必須先配置佇列管理程式，才能執行連接至 WebSphere MQ 佇列管理程式的應用程式。對於發佈/訂閱應用程式，如果您使用「佇列發佈/訂閱」介面，則需要一些額外的配置。

[為連接至 WebSphere 服務 Integration Bus 的應用程式配置服務整合匯流排](#)

您必須以配置服務整合匯流排來執行使用預設傳訊提供者的 JMS 應用程式的相同方式來配置服務整合匯流排，才能執行連接至 WebSphere 服務 Integration Bus 的應用程式。

[使用安裝精靈來安裝 Message Service Client for .NET](#)

安裝使用 InstallShield X/Windows MSI 安裝程式。有兩個設定選項可用，因此您可以選擇完整或自訂安裝。

相關參考

[XMS 應用程式連接至 WebSphere MQ 的必要條件](#)

如果 XMS 應用程式連接至 WebSphere MQ，則部分必要條件適用。

為連接至 WebSphere 服務 Integration Bus 的應用程式配置服務整合匯流排

您必須以配置服務整合匯流排來執行使用預設傳訊提供者的 JMS 應用程式的相同方式來配置服務整合匯流排，才能執行連接至 WebSphere 服務 Integration Bus 的應用程式。

開始之前

在啟動此作業之前，您必須執行下列步驟：

- 請確定已建立傳訊匯流排，且您的伺服器已新增至匯流排作為匯流排成員。
- 請確定您的應用程式有權存取至少包含一個執行中傳訊引擎的 服務整合匯流排。
- 如果需要 HTTP 作業，則必須定義 HTTP 傳訊引擎入埠傳輸通道。依預設，在伺服器安裝期間會定義 SSL 和 TCP 的通道。
- 請確定您的應用程式使用其內容已適當設定的 Connection Factory，以使用引導伺服器來連接至 服務整合匯流排。所需的資訊下限為：
 - 提供者端點，說明在與傳訊伺服器 (亦即，透過引導伺服器) 協議連線時要使用的位置和通訊協定。以最簡單的形式，對於使用預設值安裝的伺服器，可以將提供端點設為伺服器的主機名稱。
 - 用來傳送訊息的匯流排名稱。

如需 Connection Factory 內容的相關資訊，請參閱 [第 160 頁的『ConnectionFactory 的內容』](#)。

關於這項作業

必須定義您需要的任何佇列或主題空間。依預設，在伺服器安裝期間會定義稱為 Default.Topic.Space 的主題空間，但如果您需要進一步的主題空間，則必須自行建立這些主題空間。您不需要預先定義主題空間內的個別主題，因為伺服器會根據需要動態實例化這些個別主題。

下列步驟彙總您需要執行的動作。

程序

1. 建立應用程式用於 點對點 (point-to-point) 傳訊所需的佇列。
2. 建立應用程式用於 發佈/訂閱 傳訊所需的任何其他主題空間。

結果

您現在可以啟動應用程式。

相關工作

[為連接至 WebSphere MQ 佇列管理程式的應用程式配置佇列管理程式及分配管理系統](#)

本節假設您使用 WebSphere MQ 7.0 版或更新版本。您必須先配置 佇列管理程式，才能執行連接至 WebSphere MQ 佇列管理程式的應用程式。對於 發佈/訂閱 應用程式，如果您使用「佇列發佈/訂閱」介面，則需要一些額外的配置。

[為使用分配管理系統即時連線的應用程式配置分配管理系統](#)

您必須先配置分配管理系統，才能執行使用即時連線至分配管理系統的應用程式。

[使用安裝精靈來安裝 Message Service Client for .NET](#)

安裝使用 InstallShield X/Windows MSI 安裝程式。有兩個設定選項可用，因此您可以選擇完整或自訂安裝。

相關參考

[XMS 應用程式連接至 WebSphere MQ 的必要條件](#)

如果 XMS 應用程式連接至 WebSphere MQ，則部分必要條件適用。

使用安裝精靈來安裝 Message Service Client for .NET

安裝使用 InstallShield X/Windows MSI 安裝程式。有兩個設定選項可用，因此您可以選擇完整或自訂安裝。

關於這項作業

若要在 Windows 上安裝 Message Service Client for .NET，請遵循此程序。

程序

1. 如果您是從 SupportPac 安裝，請完成下列步驟，否則請直接前往步驟 [第 14 頁的『2』](#)。
 - a) 在 Windows 上，以管理者身分登入。
 - b) 執行 dotNETClientsetup.exe 安裝程式。
2. 等待安裝精靈開啟並顯示下列訊息：

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

按下一步。

精靈可能會要求您閱讀授權合約。

3. 如果系統要求您閱讀授權合約，且您接受授權合約的條款，請按一下 **我接受授權合約中的條款**，然後按下一步。

安裝精靈會要求您選擇最適合您需要的安裝類型。

4. 選取您需要的設定類型：

- 若要安裝所有程式特性，並將它們安裝在預設安裝目錄中，請按一下 **完成**。
- 若要選擇您要安裝的特性，並指定它們的安裝位置，請按一下 **自訂**。

5. 按下一步。

如果您選取完整安裝選項，則安裝精靈會顯示一則訊息，指出它已準備好開始安裝，如步驟 [第 14 頁的『8』](#) 中所述。如果您選取自訂安裝選項，則安裝精靈會要求您選取要安裝的特性，您必須先完成步驟 [第 14 頁的『6』](#) 及步驟 [第 14 頁的『7』](#)，然後再移至步驟 [第 14 頁的『8』](#)。

6. 僅針對自訂安裝，按一下特性清單中的圖示，以指定您要如何安裝 Message Service Client for .NET 特性的任何變更。如果您不想在建議的目錄中安裝 Message Service Client for .NET，請選擇另一個目錄。

如果您選擇將 Message Service Client for .NET 安裝在目前不存在的目錄中，安裝精靈會為您建立該目錄。

如果您想要開發 XMS 應用程式，請確定已選取 **開發工具及範例** 特性。此特性提供範例應用程式，以及編譯 .NET 應用程式所需的程式庫和任何其他檔案。如果您未選取此特性，則只會安裝執行 XMS 應用程式所需的檔案。

7. 如果您使用自訂安裝選項，請在選取您需要的選項之後按 **下一步**，如步驟 [第 14 頁的『6』](#) 中所述。

安裝精靈會顯示一則訊息，指出它已準備好開始安裝。

8. 按一下 **安裝** 以開始安裝。

安裝精靈會顯示一個顯示安裝進度的列。等待進度列完成。當安裝順利完成時，視窗會顯示下列訊息：

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

9. 按一下 **完成** 以關閉安裝精靈。

結果

您已順利安裝已備妥可供使用的 Message Service Client for .NET。

下一步

在執行任何 XMS 應用程式 (包括 XMS 隨附的範例應用程式) 之前，您必須先設定傳訊伺服器環境，如需詳細資料，請參閱：[第 10 頁的『設定傳訊伺服器環境』](#)。

相關概念

JNDI 查閱 Web 服務

若要從 XMS 存取 COS 命名目錄，必須在 WebSphere 服務 Integration Bus 伺服器上部署 JNDI 查閱 Web 服務。此 Web 服務會將 Java 資訊從 COS 命名服務轉換成 XMS 應用程式可以讀取的表單。

[設定傳訊伺服器環境](#)

此章節說明如何設定傳訊伺服器環境，以容許 XMS 應用程式連接至伺服器。

使用 XMS 範例應用程式

使用 XMS 隨附的範例應用程式來驗證安裝及傳訊伺服器設定，並協助您建置自己的應用程式。這些範例提供每一個 API 的一般特性概觀。

相關工作

為連接至 WebSphere MQ 佇列管理程式的應用程式配置佇列管理程式及分配管理系統

本節假設您使用 WebSphere MQ 7.0 版或更新版本。您必須先配置佇列管理程式，才能執行連接至 WebSphere MQ 佇列管理程式的應用程式。對於發佈/訂閱應用程式，如果您使用「佇列發佈/訂閱」介面，則需要一些額外的配置。

為使用分配管理系統即時連線的應用程式配置分配管理系統

您必須先配置分配管理系統，才能執行使用即時連線至分配管理系統的應用程式。

為連接至 WebSphere 服務 Integration Bus 的應用程式配置服務整合匯流排

您必須以配置服務整合匯流排來執行使用預設傳訊提供者的 JMS 應用程式的相同方式來配置服務整合匯流排，才能執行連接至 WebSphere 服務 Integration Bus 的應用程式。

相關參考

XMS 應用程式連接至 WebSphere MQ 的必要條件

如果 XMS 應用程式連接至 WebSphere MQ，則部分必要條件適用。

XMS 應用程式連接至 WebSphere MQ 的必要條件

如果 XMS 應用程式連接至 WebSphere MQ，則部分必要條件適用。

對於連接至 WebSphere MQ 佇列管理程式的應用程式，您必須在用來執行 XMS 應用程式的機器上安裝適當的 WebSphere MQ 用戶端程式庫。這些程式庫預先安裝在具有本端佇列管理程式的機器上。

若為 XMS Client for .NET，請使用 WebSphere MQ 7.0.1.0 版或更新版本隨附的用戶端程式庫。這些是 .NET 的 *WebSphere MQ* 類別。它們會啟用與 WebSphere MQ 7.0 版、WebSphere MQ 6.0 版及 WebSphere MQ 5.3 版佇列管理程式的用戶端模式連線，以及與本端佇列管理程式的連結模式連線 (如果它也是 7.0.1.0 版或更新版本)。

Microsoft .NET Framework 2.0 可重新配送套件必須安裝在要安裝 XMS 的電腦上。如果無法使用此套件，XMS 安裝將會失敗。然後，您需要退出安裝程序，在電腦上安裝 Microsoft .NET Framework 2.0 Redistributable Package，然後重新執行安裝程序。

在 Microsoft 下載網站上，您需要尋找適用於 Microsoft .NET Framework 2.0 Redistributable Package (x86) 的 dotnetfx.exe，以及適用於 Microsoft .NET Framework 2.0 Redistributable Package (x64) 的 NetFx64.exe (以何者適用)。

相關概念

設定傳訊伺服器環境

此章節說明如何設定傳訊伺服器環境，以容許 XMS 應用程式連接至伺服器。

相關工作

為連接至 WebSphere MQ 佇列管理程式的應用程式配置佇列管理程式及分配管理系統

本節假設您使用 WebSphere MQ 7.0 版或更新版本。您必須先配置佇列管理程式，才能執行連接至 WebSphere MQ 佇列管理程式的應用程式。對於發佈/訂閱應用程式，如果您使用「佇列發佈/訂閱」介面，則需要一些額外的配置。

為使用分配管理系統即時連線的應用程式配置分配管理系統

您必須先配置分配管理系統，才能執行使用即時連線至分配管理系統的應用程式。

為連接至 WebSphere 服務 Integration Bus 的應用程式配置服務整合匯流排

您必須以配置服務整合匯流排來執行使用預設傳訊提供者的 JMS 應用程式的相同方式來配置服務整合匯流排，才能執行連接至 WebSphere 服務 Integration Bus 的應用程式。

使用安裝精靈來安裝 Message Service Client for .NET

安裝使用 InstallShield X/Windows MSI 安裝程式。有兩個設定選項可用，因此您可以選擇完整或自訂安裝。

使用 XMS 範例應用程式

使用 XMS 隨附的範例應用程式來驗證安裝及傳訊伺服器設定，並協助您建置自己的應用程式。這些範例提供每一個 API 的一般特性概觀。

相關概念

JNDI 查閱 Web 服務

若要從 XMS 存取 COS 命名目錄，必須在 WebSphere 服務 Integration Bus 伺服器上部署 JNDI 查閱 Web 服務。此 Web 服務會將 Java 資訊從 COS 命名服務轉換成 XMS 應用程式可以讀取的表單。

設定傳訊伺服器環境

此章節說明如何設定傳訊伺服器環境，以容許 XMS 應用程式連接至伺服器。

相關工作

使用安裝精靈來安裝 Message Service Client for .NET

安裝使用 InstallShield X/Windows MSI 安裝程式。有兩個設定選項可用，因此您可以選擇完整或自訂安裝。

範例應用程式

範例應用程式提供每一個 API 的一般特性概觀。您可以使用它們來驗證安裝及傳訊伺服器設定，並協助您建置自己的應用程式。

如果您需要協助來建立自己的應用程式，您可以使用範例應用程式作為起點。會為每一個應用程式提供原始檔及已編譯版本。請檢閱範例原始碼，並識別建立應用程式 (ConnectionFactory、「連線」、「階段作業」、「目的地」及/或「生產者」或「消費者」) 的每一個必要物件的主要步驟，以及設定指定應用程式運作方式所需的任何特定內容。如需相關資訊，請參閱 [第 18 頁的『撰寫 XMS 應用程式』](#)。這些範例在未來的 XMS 版本中可能會變更。

下表顯示 XMS 隨附的三組範例應用程式 (每一個 API 各一個)。

範例名稱	說明
SampleConsumerCS	從佇列取得訊息或訂閱主題的訊息消費者應用程式。
SampleProducerCS	一種訊息產生者應用程式，可將訊息產生至佇列或主題。
SampleConfigCS	可用來建立檔案型受管理物件儲存庫的配置應用程式。應用程式包含特定連線設定的 Connection Factory 和目的地。然後，此受管理物件儲存庫可以與每一個範例消費者及生產者應用程式搭配使用。

在各種 API 中支援相同功能的範例具有語法差異。

- 訊息消費者和生產者應用程式範例都支援下列功能：
 - 與 WebSphere MQ、IBM Integration Bus (使用與分配管理系統的即時連線) 及 WebSphere 服務 Integration Bus 的連線
 - 使用起始環境定義介面來查閱受管理物件儲存庫
 - 與佇列 (WebSphere MQ 和 WebSphere 服務 Integration Bus) 及主題 (WebSphere MQ、與分配管理系統的即時連線及 WebSphere 服務 Integration Bus) 的連線
 - 基本、位元組、對映、物件、串流及文字訊息
- 範例訊息消費者應用程式支援同步及非同步接收模式，以及 SQL Selector 陳述式。
- 範例訊息產生者應用程式支援持續性和非持續性遞送模式。

作業模式

範例可以兩種模式之一運作：

簡式模式

您可以使用最少使用者輸入來執行範例。

進階模式

您可以更精細地自訂範例的操作方式。

所有範例都是相容的，因此可以跨語言運作。

相關概念

建置您自己的應用程式

您可以建置自己的應用程式，就像您建置範例應用程式一樣。

相關工作

執行範例應用程式

您可以在簡式或進階模式下以互動方式執行 .NET 範例應用程式，或使用自動產生或自訂的回應檔以非互動方式執行。

建置 .NET 範例應用程式

當您建置範例 .NET 應用程式時，會建立所選範例的執行檔版本。

執行範例應用程式

您可以在簡式或進階模式下以互動方式執行 .NET 範例應用程式，或使用自動產生或自訂的回應檔以非互動方式執行。

開始之前

在執行任何提供的範例應用程式之前，您必須先設定傳訊伺服器環境，讓應用程式可以連接伺服器。請參閱第 10 頁的『設定傳訊伺服器環境』。

程序

若要執行 .NET 範例應用程式，請完成下列步驟：

提示：當您執行範例應用程式時，請鍵入？隨時可以獲得接下來要做什麼的協助。

1. 選取您要執行範例應用程式的模式。

 鍵入 Advanced 或 Simple。

2. 回答問題。

 若要選取預設值 (顯示在問題結尾的方括弧中)，請按 Enter 鍵。若要選取不同的值，請鍵入適當的值，然後按 Enter 鍵。

 以下是範例問題：

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

在此情況下，預設值為 wpm (與 WebSphere 服務 Integration Bus 的連線)。

結果

當您執行範例應用程式時，會在現行工作目錄中自動產生回應檔。回應檔名稱的格式為 `connection_type-sample_type.rsp`；例如 `wpm-producer.rsp`。必要的話，您可以使用產生的回應檔，以相同的選項重新執行範例應用程式，以便您不必重新輸入選項。

相關概念

範例應用程式

範例應用程式提供每一個 API 的一般特性概觀。您可以使用它們來驗證安裝及傳訊伺服器設定，並協助您建置自己的應用程式。

相關工作

建置 .NET 範例應用程式

當您建置範例 .NET 應用程式時，會建立所選範例的執行檔版本。

建置 .NET 範例應用程式

當您建置範例 .NET 應用程式時，會建立所選範例的執行檔版本。

開始之前

安裝適當的編譯器。這項作業假設您已安裝 Visual Studio 2012，且您熟悉如何使用它。

程序

若要建置 .NET 範例應用程式，請完成下列步驟：

1. 按一下 .NET 範例隨附的 `Samples.sln` 解決方案檔案。
2. 在「解決方案瀏覽器」視窗中，用滑鼠右鍵按一下解決方案 **範例**，然後選取 **建置解決方案**。

結果

視您選擇的配置而定，會在範例的適當子資料夾 (`bin/Debug` 或 `bin/Release`) 中建立可執行程式。此程式具有與資料夾相同的名稱，字尾為 `CS`。例如，如果您要建置訊息產生者範例應用程式的 C# 版本，則會在 `SampleProducer` 資料夾中建立 `SampleProducerCS.exe`。

相關概念

範例應用程式

範例應用程式提供每一個 API 的一般特性概觀。您可以使用它們來驗證安裝及傳訊伺服器設定，並協助您建置自己的應用程式。

第 37 頁的『建置您自己的應用程式』

您可以建置自己的應用程式，就像您建置範例應用程式一樣。

相關工作

執行範例應用程式

您可以在簡式或進階模式下以互動方式執行 .NET 範例應用程式，或使用自動產生或自訂的回應檔以非互動方式執行。

開發 XMS 應用程式

此章節提供在撰寫 XMS 應用程式時可能有用的資訊。

如需撰寫 XMS 應用程式的相關資訊，請參閱下列主題：

撰寫 XMS 應用程式

此章節提供資訊來協助您撰寫 XMS 應用程式。

此章節包含用於撰寫 XMS 應用程式的一般概念。另請參閱第 38 頁的『撰寫 XMS .NET 應用程式』，以取得建立 .NET 應用程式的特定資訊。

此章節包含下列區段：

- [第 19 頁的『執行緒作業模型』](#)
- [第 20 頁的『ConnectionFactories 和 Connection 物件』](#)
- [第 22 頁的『階段作業』](#)
- [第 25 頁的『目的地』](#)
- [第 29 頁的『訊息產生者』](#)
- [第 29 頁的『訊息消費者』](#)
- [第 32 頁的『佇列瀏覽器』](#)
- [第 33 頁的『要求者』](#)
- [第 33 頁的『物件刪除』](#)
- [第 34 頁的『XMS 初始類型』](#)
- [第 34 頁的『將內容值從一個資料類型隱含轉換為另一個資料類型』](#)

- [第 36 頁的『反覆運算子』](#)
- [第 37 頁的『編碼字集 ID』](#)
- [第 37 頁的『XMS 錯誤碼和異常狀況碼』](#)
- [第 37 頁的『建置您自己的應用程式』](#)

相關概念

撰寫 XMS .NET 應用程式

此章節提供在撰寫 XMS.NET 應用程式時協助您的資訊。

相關參考

.NET 介面

此區段記載 .NET 類別介面及其內容和方法。

執行緒作業模型

一般規則控管多執行緒應用程式如何使用 XMS 物件。

- 在不同執行緒上只能同時使用下列類型的物件：
 - ConnectionFactory
 - 連線
 - ConnectionMeta 資料
 - 目的地
- Session 物件一次只能在單一執行緒上使用。

在 API 參照章節中，方法的介面定義中標示為 "執行緒環境定義" 的項目會指出這些規則的異常狀況。

可在執行時期處理的錯誤狀況

API 呼叫的回覆碼是可在執行時期處理的錯誤狀況。您處理此類型錯誤的方式取決於您是使用 C 還是 C++ API。

如何在執行時期偵測錯誤

如果應用程式呼叫 C API 函數但呼叫失敗，則會傳回回覆碼不是 XMS_OK 的回應，並包含 XMS 錯誤區塊，其中包含失敗原因的相關資訊。

當使用方法時，C++ API 會擲出異常狀況。

應用程式會使用異常狀況接聽器，以非同步方式通知連線發生問題。異常狀況接聽器提供給 XMS C 或 C++ API，並使用 API 來起始設定。

如何在執行時期處理錯誤

部分錯誤狀況指出部分資源無法使用，且應用程式可以採取的動作取決於應用程式正在呼叫的 XMS 函數。例如，如果連線無法連接至伺服器，則應用程式可能想要定期重試，直到建立連線為止。XMS 錯誤區塊或異常狀況可能未包含足夠資訊來決定要採取的動作，在這些狀況中，通常會有鏈結的錯誤區塊或異常狀況包含更具體的診斷資訊。

在 C API 中，一律測試回覆碼不是 XMS_OK 的回應，並一律在 API 呼叫上傳遞錯誤區塊。所採取的動作通常取決於應用程式所使用的 API 函數。

在 C++ API 中，一律在 try 區塊中包含方法的呼叫，如果要捕捉所有類型的 XMS 異常狀況，請在 catch 建構中指定 Exception 類別。

異常狀況接聽器是可隨時啟動的非同步錯誤狀況路徑。當啟動異常狀況接聽器函數時，在其自己的執行緒上，通常會指出比正常 XMS API 錯誤狀況更嚴重的失敗。可以採取任何適當的動作，但您必須小心遵循 XMS 執行緒作業模型的規則，如 [第 19 頁的『執行緒作業模型』](#) 中所述。

ConnectionFactory 和 Connection 物件

ConnectionFactory 物件提供應用程式用來建立 Connection 物件的範本。應用程式使用 Connection 物件來建立 Session 物件。

對於 .NET，XMS 應用程式會先使用 XMSFactoryFactory 物件來取得 ConnectionFactory 物件的參照，該物件適用於所需的通訊協定類型。然後，這個 ConnectionFactory 物件只能產生該通訊協定類型的連線。

XMS 應用程式可以建立多個連線，而多執行緒應用程式可以在多個執行緒上同時使用單一 Connection 物件。Connection 物件會封裝應用程式與傳訊伺服器之間的通訊連線。

連線有數個用途：

- 當應用程式建立連線時，可以鑑別應用程式。
- 應用程式可以將唯一用戶端 ID 與連線相關聯。用戶端 ID 用來支援發佈/訂閱網域中的可延續訂閱。有兩種方式可以設定用戶端 ID：

指派連線用戶端 ID 的偏好方式，是使用內容在用戶端特定的 ConnectionFactory 物件中配置，並將它透過地指派給它所建立的連線。

指派用戶端 ID 的替代方式是使用在 Connection 物件上設定的提供者特定值。此值不會置換已透過管理方式配置的 ID。它是針對不存在管理指定 ID 的情況所提供。如果管理上指定的 ID 確實存在，嘗試以提供者特定值置換它會導致擲出異常狀況。如果應用程式明確設定 ID，則必須在建立連線之後，並在對連線採取任何其他動作之前立即執行；否則，會擲出異常狀況。

XMS 應用程式通常會建立連線、一或多個階段作業，以及一些訊息產生者和訊息消費者。

就系統資源而言，建立連線相當昂貴，因為它涉及建立通訊連線，而且也可能涉及鑑別應用程式。

相關工作

[建立受管理物件](#)

必須使用適當的管理工具來建立 XMS 應用程式建立傳訊伺服器連線所需的 ConnectionFactory 和「目的地」物件定義。

相關參考

[IConnectionFactory \(適用於 .NET 介面\)](#)

應用程式使用 Connection Factory 來建立連線。

[ConnectionFactory 的內容](#)

ConnectionFactory 物件內容的概觀，以及指向更詳細參照資訊的鏈結。

[IDestination \(適用於 .NET 介面\)](#)

目的地是應用程式傳送訊息的地方，及（或）應用程式從中接收訊息的來源。

[目的地的內容](#)

「目的地」物件的內容概觀，以及指向更詳細參照資訊的鏈結。

連線已啟動和已停止模式

連線可以在已啟動或已停止模式下運作。

當應用程式建立連線時，連線會處於已停止模式。當連線處於已停止模式時，應用程式可以起始設定階段作業，它可以傳送訊息，但無法同步或非同步接收它們。

應用程式可以呼叫 Start Connection 方法來啟動連線。當連線處於已啟動模式時，應用程式可以傳送及接收訊息。然後，應用程式可以透過呼叫「停止連線」及 Start Connection 方法來停止並重新啟動連線。

相關概念

[連線關閉](#)

應用程式會呼叫「關閉連線」方法來關閉連線。

[異常狀況處理](#)

如果應用程式只使用連線來非同步使用訊息，它只會利用異常狀況接聽器來瞭解連線的問題。

[WebSphere 服務整合匯流排的連線](#)

XMS 應用程式可以使用直接 TCP/IP 連線或使用透過 TCP/IP 的 HTTP 來連接至 WebSphere 服務 Integration Bus。

連線關閉

應用程式會呼叫「關閉連線」方法來關閉連線。

當應用程式關閉連線時，XMS 會執行下列動作：

- 它會關閉與連線相關聯的所有階段作業，並刪除與這些階段作業相關聯的特定物件。如需刪除哪些物件的相關資訊，請參閱第 33 頁的『物件刪除』。同時，XMS 會回復階段作業內目前進行中的任何交易。
- 它會結束與傳訊伺服器的通訊連線。
- 它會釋放連線所使用的記憶體及其他內部資源。

在關閉連線之前，XMS 不會確認在階段作業期間收到任何它無法確認的訊息。如需確認接收訊息的相關資訊，請參閱第 23 頁的『訊息確認』。

相關概念

連線已啟動和已停止模式

連線可以在已啟動或已停止模式下運作。

異常狀況處理

如果應用程式只使用連線來非同步使用訊息，它只會利用異常狀況接聽器來瞭解連線的問題。

WebSphere 服務整合匯流排的連線

XMS 應用程式可以使用直接 TCP/IP 連線或使用透過 TCP/IP 的 HTTP 來連接至 WebSphere 服務 Integration Bus。

異常狀況處理

如果應用程式只使用連線來非同步使用訊息，它只會利用異常狀況接聽器來瞭解連線的問題。

XMS .NET 異常狀況全都衍生自 System.Exception。如需相關資訊，請參閱第 41 頁的『.NET 中的錯誤處理』。

相關概念

連線已啟動和已停止模式

連線可以在已啟動或已停止模式下運作。

連線關閉

應用程式會呼叫「關閉連線」方法來關閉連線。

WebSphere 服務整合匯流排的連線

XMS 應用程式可以使用直接 TCP/IP 連線或使用透過 TCP/IP 的 HTTP 來連接至 WebSphere 服務 Integration Bus。

WebSphere 服務整合匯流排的連線

XMS 應用程式可以使用直接 TCP/IP 連線或使用透過 TCP/IP 的 HTTP 來連接至 WebSphere 服務 Integration Bus。

HTTP 通訊協定可以在無法使用直接 TCP/IP 連線的情況下使用。一個常見的狀況是透過防火牆進行通訊時，例如當兩個企業交換訊息時。使用 HTTP 透過防火牆進行通訊通常稱為 HTTP 通道作業。不過，HTTP 通道作業本質上比使用直接 TCP/IP 連線更慢，因為 HTTP 標頭會大幅增加所傳送的資料量，而且因為 HTTP 通訊協定需要比 TCP/IP 更多的通訊流程。

若要建立 TCP/IP 連線，應用程式可以使用其 `XMSC_WPM_TARGET_TRANSPORT_CHAIN` 內容設為 `XMSC_WPM_TARGET_TRANSPORT_CHAIN_BASIC` 的 Connection Factory。這是內容的預設值。如果順利建立連線，連線的 `XMSC_WPM_CONNECTION_PROTOCOL` 內容會設為 `XMSC_WPM_CP_TCP`。

若要建立使用 HTTP 的連線，應用程式必須使用其 `XMSC_WPM_TARGET_TRANSPORT_CHAIN` 內容設為入埠傳輸鏈名稱 (配置為使用 HTTP 傳輸通道) 的 Connection Factory。如果順利建立連線，連線的 `XMSC_WPM_CONNECTION_PROTOCOL` 內容會設為 `XMSC_WPM_CP_HTTP`。如需如何配置傳輸鏈的相關資訊，請參閱 WebSphere Application Server 產品說明文件中的 配置傳輸鏈。

連接至引導伺服器時，應用程式具有類似的通訊協定選項。Connection Factory 的 `XMSC_WPM_PROVIDER_ENDPOINTS` 內容是引導伺服器的一或多個端點位址序列。每一個端點位址的引導

傳輸鏈元件可以是 XMSC_WPM_BOOTSTRAP_TCP (對於引導伺服器的 TCP/IP 連線) 或 XMSC_WPM_BOOTSTRAP_HTTP (對於使用 HTTP 的連線)。

相關概念

連線已啟動和已停止模式

連線可以在已啟動或已停止模式下運作。

連線關閉

應用程式會呼叫「關閉連線」方法來關閉連線。

異常狀況處理

如果應用程式只使用連線來非同步使用訊息，它只會利用異常狀況接聽器來瞭解連線的問題。

相關工作

建立受管理物件

必須使用適當的管理工具來建立 XMS 應用程式建立傳訊伺服器連線所需的 ConnectionFactory 和「目的地」物件定義。

相關參考

ConnectionFactory (適用於 .NET 介面)

應用程式使用 Connection Factory 來建立連線。

ConnectionFactory 的內容

ConnectionFactory 物件內容的概觀，以及指向更詳細參照資訊的鏈結。

IDestination (適用於 .NET 介面)

目的地是應用程式傳送訊息的地方，及（或）應用程式從中接收訊息的來源。

目的地的內容

「目的地」物件的內容概觀，以及指向更詳細參照資訊的鏈結。

階段作業

階段作業是用於傳送及接收訊息的單一執行緒環境定義。

應用程式可以使用階段作業來建立訊息、訊息產生者、訊息消費者、佇列瀏覽器及暫時目的地。應用程式也可以使用階段作業來執行區域交易。

應用程式可以建立多個階段作業，其中每一個階段作業會產生及耗用與其他階段作業無關的訊息。如果在個別階段作業 (或甚至在相同階段作業中) 中有兩個訊息消費者訂閱相同的主题，則每一個都會收到針對該主题發佈的任何訊息副本。

不同於 Connection 物件，Session 物件不能同時在不同的執行緒上使用。只能從「階段作業」物件當時正在使用的執行緒以外的執行緒呼叫「階段作業」物件的「關閉階段作業」方法。「關閉階段作業」方法會結束階段作業，並釋放配置給階段作業的任何系統資源。

如果應用程式必須在多個執行緒上同時處理訊息，應用程式必須在每一個執行緒上建立一個階段作業，然後在該執行緒內的任何傳送或接收作業中使用該階段作業。

交易式階段作業

XMS 應用程式可以執行區域交易。區域交易是只涉及變更應用程式所連接之佇列管理程式或服務整合匯流排的資源的交易。

只有在應用程式連接至 WebSphere MQ 佇列管理程式或 WebSphere 服務整合匯流排時，這個區段中的資訊才相關。此資訊與分配管理系統的即時連線無關。

如果要執行區域交易，應用程式必須先呼叫 Connection 物件的「建立階段作業」方法來建立交易式階段作業，並指定作為交易式階段作業的參數。隨後，在階段作業內傳送及接收的所有訊息會分組成一連串交易。當應用程式確定或回復自交易開始以來所傳送及接收的訊息時，交易會結束。

為了確定交易，應用程式會呼叫「階段作業」物件的「確定」方法。當確定交易時，在交易內傳送的所有訊息都會變成可供遞送給其他應用程式，且會確認交易內收到的所有訊息，以便傳訊伺服器不會再次嘗試將它們遞送給應用程式。在點對點 (point-to-point) 網域中，傳訊伺服器也會從其佇列中移除收到的訊息。

為了回復交易，應用程式會呼叫「階段作業」物件的「回復」方法。當回復交易時，傳訊伺服器會捨棄交易內所傳送的所有訊息，且交易內所接收的所有訊息會重新可供遞送。在點對點 (point-to-point) 網域中，已收到的訊息會放回其佇列中，並再次對其他應用程式可見。

當應用程式建立交易式階段作業或呼叫「確定」或「回復」方法時，新交易會自動啟動。因此，交易式階段作業一律具有作用中交易。

當應用程式關閉交易式階段作業時，會進行隱含回復。當應用程式關閉連線時，所有連線的交易階段作業都會進行隱含回復。

交易完全包含在交易式階段作業內。交易無法跨越階段作業。這表示應用程式無法在兩個以上交易階段作業中傳送及接收訊息，然後將所有這些動作確定或回復為單一交易。

相關概念

訊息確認

每一個未交易的階段作業都有一個確認模式，用來決定如何確認應用程式收到的訊息。有三種確認模式可用，而選擇確認模式會影響應用程式的設計。

非同步訊息遞送

XMS 使用一個執行緒來處理階段作業的所有非同步訊息遞送。這表示一次只能執行一個訊息接聽器函數或一個 `onMessage()` 方法。

同步訊息遞送

如果應用程式使用 `MessageConsumer` 物件的「接收」方法，則訊息會同步遞送至應用程式。

訊息遞送模式

XMS 支援兩種訊息遞送模式。

訊息確認

每一個未交易的階段作業都有一個確認模式，用來決定如何確認應用程式收到的訊息。有三種確認模式可用，而選擇確認模式會影響應用程式的設計。

只有在應用程式連接至 `WebSphere MQ` 佇列管理程式或 `WebSphere 服務 Integration Bus` 時，此區段中的資訊才相關。此資訊與分配管理系統的即時連線無關。

XMS 使用相同的機制來確認 JMS 使用的訊息接收。

如果階段作業未進行交易，則階段作業的確認模式會決定應用程式收到訊息的確認方式。下列段落說明這三種確認模式：

XMSC_AUTO_認可

階段作業會自動確認應用程式收到的每一則訊息。

如果訊息同步遞送至應用程式，階段作業會在每次「接收」呼叫順利完成時確認接收訊息。

如果應用程式順利收到訊息，但失敗導致無法進行確認，則訊息會再次變成可供遞送。因此，應用程式必須能夠處理重新遞送的訊息。

XMSC_DUPS_OK_認可

階段作業會確認應用程式在選取時所收到的訊息。

使用此確認模式可減少階段作業必須執行的工作量，但阻止訊息確認的失敗可能會導致多個訊息重新可供遞送。因此，應用程式必須能夠處理重新遞送的訊息。

XMSC_CLIENT_認可

應用程式透過呼叫 `Message` 類別的 `Acknowledge` 方法來確認它收到的訊息。

應用程式可以個別確認接收每一個訊息，也可以接收一批訊息，並僅針對它接收的最後一則訊息呼叫 `Acknowledge` 方法。當呼叫 `Acknowledge` 方法時，會確認自前次呼叫方法以來收到的所有訊息。

與任何這些確認模式一起使用，應用程式可以透過呼叫 `Session` 類別的 `Recover` 方法來停止並重新啟動階段作業中的訊息遞送。重新遞送其回執先前未確認的訊息。不過，它們可能不會以先前遞送的相同順序來遞送。同時，較高優先順序的訊息可能已到達，部分原始訊息可能已過期。在點對點網域中，部分原始訊息可能已由另一個應用程式使用。

應用程式可以檢查訊息的 `JMSRedelivered` 標頭欄位內容，來判斷是否正在重新遞送訊息。應用程式透過呼叫 `Message` 類別的 `Get JMSRedelivered` 方法來執行此動作。

相關概念

交易式階段作業

XMS 應用程式可以執行區域交易。區域交易是只涉及變更應用程式所連接之佇列管理程式或服務整合匯流排的資源的交易。

非同步訊息遞送

XMS 使用一個執行緒來處理階段作業的所有非同步訊息遞送。這表示一次只能執行一個訊息接聽器函數或一個 `onMessage()` 方法。

同步訊息遞送

如果應用程式使用 `MessageConsumer` 物件的「接收」方法，則訊息會同步遞送至應用程式。

訊息遞送模式

XMS 支援兩種訊息遞送模式。

非同步訊息遞送

XMS 使用一個執行緒來處理階段作業的所有非同步訊息遞送。這表示一次只能執行一個訊息接聽器函數或一個 `onMessage()` 方法。

如果階段作業中有多个訊息消費者非同步接收訊息，且訊息接聽器功能或 `onMessage()` 方法正在將訊息遞送至訊息消費者，則等待相同訊息的任何其他訊息消費者都必須繼續等待。等待遞送至階段作業的其他訊息也必須繼續等待。

如果應用程式需要並行遞送訊息，請建立多個階段作業，讓 XMS 使用多個執行緒來處理非同步訊息遞送。如此一來，就可以同時執行多個訊息接聽器函數或 `onMessage()` 方法。

將訊息接聽器指派給消費者，不會使階段作業變成非同步。只有在呼叫 `Connection.Start` 方法時，階段作業才會變成非同步。在呼叫 `Connection.Start` 方法之前，允許所有同步呼叫。當呼叫 `Connection.Start` 時，會開始將訊息遞送至消費者。

如果必須在非同步階段作業上進行同步呼叫 (例如建立消費者或生產者)，則必須呼叫 `Connection.Stop`。您可以呼叫 `Connection.Start` 方法來開始遞送訊息，以回復階段作業。唯一例外是「階段作業」訊息遞送執行緒，它是將訊息遞送至回呼函數的執行緒。此執行緒可以對訊息回呼函數中的階段作業 (「關閉」呼叫除外) 進行任何呼叫。

註: 在未受管理模式中，WMQ.NET 用戶端。因此，在「非同步接收」模式下，用戶端應用程式無法在 `MessageListener` 回呼內建立或關閉階段作業。在 `MessageListener` 方法之外建立並刪除階段作業。

相關概念

交易式階段作業

XMS 應用程式可以執行區域交易。區域交易是只涉及變更應用程式所連接之佇列管理程式或服務整合匯流排的資源的交易。

訊息確認

每一個未交易的階段作業都有一個確認模式，用來決定如何確認應用程式收到的訊息。有三種確認模式可用，而選擇確認模式會影響應用程式的設計。

同步訊息遞送

如果應用程式使用 `MessageConsumer` 物件的「接收」方法，則訊息會同步遞送至應用程式。

訊息遞送模式

XMS 支援兩種訊息遞送模式。

同步訊息遞送

如果應用程式使用 `MessageConsumer` 物件的「接收」方法，則訊息會同步遞送至應用程式。

使用「接收」方法，應用程式可以等待指定的訊息一段時間，也可以無限期地等待。或者，如果應用程式不想等待訊息，則可以使用「不等待接收」方法。

相關概念

交易式階段作業

XMS 應用程式可以執行區域交易。區域交易是只涉及變更應用程式所連接之佇列管理程式或服務整合匯流排的資源的交易。

訊息確認

每一個未交易的階段作業都有一個確認模式，用來決定如何確認應用程式收到的訊息。有三種確認模式可用，而選擇確認模式會影響應用程式的設計。

非同步訊息遞送

XMS 使用一個執行緒來處理階段作業的所有非同步訊息遞送。這表示一次只能執行一個訊息接聽器函數或一個 `onMessage()` 方法。

訊息遞送模式

XMS 支援兩種訊息遞送模式。

訊息遞送模式

XMS 支援兩種訊息遞送模式。

- 持續訊息會遞送一次。傳訊伺服器會採取特殊預防措施 (例如記載訊息)，以確保持續訊息不會在傳輸中遺失，即使發生失敗也一樣。
- 非持續性訊息只會遞送一次。非持續訊息比持續訊息更不可靠，因為一旦失敗，它們可能會在傳輸中遺失。

遞送模式的選擇是在可靠性與效能之間進行取捨。非持續訊息的傳輸速度通常比持續訊息快。

相關概念

交易式階段作業

XMS 應用程式可以執行區域交易。區域交易是只涉及變更應用程式所連接之佇列管理程式或服務整合匯流排的資源的交易。

訊息確認

每一個未交易的階段作業都有一個確認模式，用來決定如何確認應用程式收到的訊息。有三種確認模式可用，而選擇確認模式會影響應用程式的設計。

非同步訊息遞送

XMS 使用一個執行緒來處理階段作業的所有非同步訊息遞送。這表示一次只能執行一個訊息接聽器函數或一個 `onMessage()` 方法。

同步訊息遞送

如果應用程式使用 `MessageConsumer` 物件的「接收」方法，則訊息會同步遞送至應用程式。

目的地

XMS 應用程式使用「目的地」物件來指定所傳送訊息的目的地，以及所接收訊息的來源。

XMS 應用程式可以在執行時期建立「目的地」物件，或從受管理物件的儲存庫取得預先定義的目的地。

如同 `ConnectionFactory`，XMS 應用程式指定目的地的最靈活方式是將它定義為受管理物件。使用此方法，以 C、C++、.NET 語言及 Java 撰寫的應用程式可以共用目的地的定義。可以在不變更任何程式碼的情況下變更受管理目的地物件的內容。

對於 .NET 應用程式，您可以使用 `CreateTopic` 或 `CreateQueue` 方法來建立目的地。在 .NET API 的 `ISession` 和 `XMSFactoryFactory` 物件中提供這兩種方法。如需相關資訊，請參閱 [第 39 頁的『.NET 中的目的地』](#) 及 [第 91 頁的『IDestination』](#)。

相關參考

IDestination (適用於 .NET 介面)

目的地是應用程式傳送訊息的地方，及 (或) 應用程式從中接收訊息的來源。

目的地的內容

「目的地」物件的內容概觀，以及指向更詳細參照資訊的鏈結。

主題統一資源識別碼

主題統一資源識別碼 (URI) 指定主題的名稱; 它也可以為它指定一或多個內容。

主題的 URI 以序列 `topic://` 開頭，後面接著主題的名稱及 (選用) 設定其餘主題內容的名稱/值配對清單。主題名稱不能是空的。

以下是 .NET 程式碼片段中的範例:

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

如需主題內容的相關資訊 (包括您可以在 URI 中使用的名稱及有效值), 請參閱 [第 165 頁的『目的地的內容』](#)。

指定要在訂閱中使用的主題 URI 時, 可以使用萬用字元。這些萬用字元的語法視連線類型及分配管理系統版本而定; 可用的選項如下:

- WebSphere MQ V7.0 佇列管理程式 具有字元層次萬用字元格式
- WebSphere MQ V7.0 佇列管理程式 (含主題層次萬用字元格式)
- WebSphere 服務整合匯流排

WebSphere MQ V7.0 佇列管理程式 具有字元層次萬用字元格式

具有字元層次萬用字元格式的 WebSphere MQ V7.0 佇列管理程式 使用下列萬用字元:

* 代表 0 個以上字元

? 代表 1 個字元

% 表示跳出字元

[第 26 頁的表 1](#) 提供一些範例, 說明如何使用這個萬用字元架構。

統一資源識別碼 (URI)	符合	範例
"topic://Sport *Results"	以 "Sport" 開頭且以 "Results" 結尾的所有主題	"topic://SportsResults" 及 "topic://Sport/Hockey/National/Div3/Results"
" topic://Sport?結果 "	所有主題以 "Sport" 開頭, 後面接著單一字元, 後面接著 "Results"	"topic://SportsResults" 及 "topic://SportXResults"
"topic://Sport/ * ball*/Div? / Results*/???"	主題	"topic://Sport/Football/Div1/Results/2002/Nov" 及 "topic://Sport/Netball/National/Div3/Results/02/Jan"

WebSphere MQ V7.0 佇列管理程式 (含主題層次萬用字元格式)

具有主題層次萬用字元格式的 WebSphere MQ V7.0 佇列管理程式 使用下列萬用字元:

to match multiple levels

+ 以符合單一層次

[第 26 頁的表 2](#) 提供一些如何使用這個萬用字元架構的範例。

統一資源識別碼 (URI)	符合	範例
"topic://Sport/ + / Results"	Sport 與 Results 之間具有單一階層式層次名稱的所有主題	"topic://Sport/Football/Results" 及 "topic://Sport/Ju-Jitsu/Results"
"topic://Sport/#/Results"	以 "Sport/" 開頭且以 "/Results" 結尾的所有主題	"topic://Sport/Football/Results" 及 "topic://Sport/Hockey/National/Div3/Results"
" topic://Sport/ Fooball/#"	以 "Sport/Fooball/" 開頭的所有主題	"topic://Sport/Fooball/Results" 及 "topic://Sport/Fooball/TeamNews/Signings/Managerial"

WebSphere 服務整合匯流排

WebSphere 服務整合匯流排 使用下列萬用字元:

- * to match any characters at one level in the hierarchy
- // to match 0 or more levels
- //. to match 0 or more levels (at the end of a Topic expression)

第 27 頁的表 3 提供一些如何使用這個萬用字元架構的範例。

統一資源識別碼 (URI)	符合	範例
"topic://Sport/ * ball/ Results"	Sport 與 Results 之間具有單一階層式層次名稱以 "ball" 結尾的所有主題	"topic://Sport/Football/Results" 及 "topic://Sport/Netball/Results"
"topic://Sport// Results"	以 "Sport/" 開頭且以 "/Results" 結尾的所有主題	"topic://Sport/Football/Results" 及 "topic://Sport/Hockey/National/Div3/Results"
"topic://Sport/ Fooball//."	以 "Sport/Fooball/" 開頭的所有主題	"topic://Sport/Fooball/Results" 及 "topic://Sport/Fooball/TeamNews/Signings/Managerial"
"topic://Sport/ * ball// Results//."	主題	"topic://Sport/Football/Results" 及 "topic://Sport/Netball/National/Div3/Results/2002/November"

相關概念

[佇列統一資源識別碼](#)

佇列的 URI 指定佇列的名稱; 它也可以指定佇列的一或多個內容。

[暫時目的地](#)

XMS 應用程式可以建立及使用暫時目的地。

[目的地萬用字元](#)

XMS 提供對目的地萬用字元的支援, 確保萬用字元可以傳遞至比對所需的位置。XMS 可以使用的每一種伺服器類型都有不同的萬用字元架構。

相關參考

[IDestination \(適用於 .NET 介面\)](#)

目的地是應用程式傳送訊息的地方, 及 (或) 應用程式從中接收訊息的來源。

[目的地的內容](#)

「目的地」物件的內容概觀, 以及指向更詳細參照資訊的鏈結。

佇列統一資源識別碼

佇列的 URI 指定佇列的名稱; 它也可以指定佇列的一或多個內容。

佇列的 URI 以序列佇列: // 開頭, 後面接著佇列的名稱; 它也可能包含名稱/值配對清單, 可設定其餘佇列內容。

對於 WebSphere MQ 佇列 (但不適用於 WebSphere Application Server 預設傳訊提供者佇列), 佇列所在的佇列管理程式可以在佇列之前指定, 並以/區隔佇列管理程式名稱與佇列名稱。

如果指定佇列管理程式, 則必須是 XMS 使用此佇列直接連接的佇列管理程式, 或者必須可從這個佇列存取。僅支援遠端佇列管理程式從佇列擷取訊息, 不支援將訊息放入佇列。如需完整資料, 請參閱 WebSphere MQ 佇列管理程式文件。

如果未指定佇列管理程式, 則額外/分隔字元是選用的, 且其存在與否與佇列的定義沒有任何差異。

在稱為 QM_A 的佇列管理程式上, 下列佇列定義全都相等於 XMS 直接連接的 WebSphere MQ 佇列 QB:

```
queue://QB
```

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

相關概念

主題統一資源識別碼

主題統一資源識別碼 (URI) 指定主題的名稱; 它也可以為它指定一或多個內容。

暫時目的地

XMS 應用程式可以建立及使用暫時目的地。

目的地萬用字元

XMS 提供對目的地萬用字元的支援, 確保萬用字元可以傳遞至比對所需的位置。XMS 可以使用的每一種伺服器類型都有不同的萬用字元架構。

相關參考

IDestination (適用於 .NET 介面)

目的地是應用程式傳送訊息的地方, 及 (或) 應用程式從中接收訊息的來源。

目的地的內容

「目的地」物件的內容概觀, 以及指向更詳細參照資訊的鏈結。

暫時目的地

XMS 應用程式可以建立及使用暫時目的地。

應用程式通常會使用暫時目的地來接收要求訊息的回覆。為了指定要傳送要求訊息之回覆的目的地, 應用程式會呼叫代表要求訊息之訊息物件的 `Set JMSReplyTo` 方法。呼叫上指定的目的地可以是暫時目的地。

雖然階段作業是用來建立暫時目的地, 但暫時目的地的範圍實際上是用來建立階段作業的連線。任何連線的階段作業都可以建立暫時目的地的訊息產生者和訊息消費者。暫時目的地會保留到明確刪除或連線結束為止 (以先發生的情況為準)。

當應用程式建立暫時佇列時, 會在應用程式所連接的傳訊伺服器中建立佇列。如果應用程式連接至佇列管理程式, 則會從模型佇列建立動態佇列, 其名稱由 `XMSC_WMQ_TEMPORARY_MODEL` 內容指定, 且用來形成動態佇列名稱的字首由 `XMSC_WMQ_TEMP_Q_PREFIX` 內容指定。如果應用程式連接至服務整合匯流排, 則會在匯流排中建立暫時佇列, 且 `XMSC_WPM_TEMP_Q_PREFIX` 內容會指定用來形成暫時佇列名稱的字首。

當連接至服務整合匯流排的應用程式建立暫時主題時, `XMSC_WPM_TEMP_TOPIC_PREFIX` 內容會指定用來形成暫時主題名稱的字首。

相關概念

主題統一資源識別碼

主題統一資源識別碼 (URI) 指定主題的名稱; 它也可以為它指定一或多個內容。

佇列統一資源識別碼

佇列的 URI 指定佇列的名稱; 它也可以指定佇列的一或多個內容。

目的地萬用字元

XMS 提供對目的地萬用字元的支援, 確保萬用字元可以傳遞至比對所需的位置。XMS 可以使用的每一種伺服器類型都有不同的萬用字元架構。

相關參考

IDestination (適用於 .NET 介面)

目的地是應用程式傳送訊息的地方, 及 (或) 應用程式從中接收訊息的來源。

目的地的內容

「目的地」物件的內容概觀, 以及指向更詳細參照資訊的鏈結。

目的地萬用字元

XMS 提供對目的地萬用字元的支援, 確保萬用字元可以傳遞至比對所需的位置。XMS 可以使用的每一種伺服器類型都有不同的萬用字元架構。

這些計劃包括:

連線類型	萬用字元架構	說明
WebSphere MQ 佇列管理程式	* ? %	0 個以上字元 1 個字元 跳出字元
與分配管理系統的即時連線	# +	符合多個層次 符合單一層次
WebSphere 服務 Integration Bus	* // //.	符合階層中一個層次的任何字元 符合 0 個以上層次 符合 0 個以上層次 (在主題表示式結尾)

相關概念

主題統一資源識別碼

主題統一資源識別碼 (URI) 指定主題的名稱; 它也可以為它指定一或多個內容。

佇列統一資源識別碼

佇列的 URI 指定佇列的名稱; 它也可以指定佇列的一或多個內容。

暫時目的地

XMS 應用程式可以建立及使用暫時目的地。

相關參考

IDestination (適用於 .NET 介面)

目的地是應用程式傳送訊息的地方, 及 (或) 應用程式從中接收訊息的來源。

目的地的內容

「目的地」物件的內容概觀, 以及指向更詳細參照資訊的鏈結。

訊息產生者

在 XMS 中, 可以使用有效的目的地或沒有相關聯的目的地來建立訊息產生者。建立具有空值目的地的訊息產生者時, 在傳送訊息時必須指定有效的目的地。

沒有相關聯目的地的訊息產生者

在 XMS .NET 中, 可以使用空值目的地來建立訊息產生者。

當使用 .NET API 時, 如果要建立沒有關聯目的地的訊息產生者, 必須將 NULL 作為參數傳遞至 ISession 物件的 CreateProducer() 方法 (例如 session.CreateProducer(null))。不過, 在傳送訊息時必須指定有效的目的地。

具有相關聯目的地的訊息產生者

在此實務範例中, 會使用有效的目的地來建立訊息產生者。在傳送作業期間, 不需要指定目的地。

訊息消費者

訊息消費者可以分類為可延續及不可延續訂閱者, 以及同步及非同步訊息消費者。

延續訂閱者

可延續訂閱者是一種訊息消費者, 可接收針對主題發佈的所有訊息, 包括訂閱者處於非作用中狀態時發佈的訊息。

只有在應用程式連接至 WebSphere MQ 佇列管理程式或 WebSphere 服務整合匯流排時, 這個區段中的資訊才相關。此資訊與分配管理系統的即時連線無關。

為了建立主題的可延續訂閱者，應用程式會呼叫 `Session` 物件的 `Create Durable Subscriber` 方法，並將識別可延續訂閱的名稱和代表主題的 `Destination` 物件指定為參數。應用程式可以使用或不使用訊息選取器來建立可延續訂閱者，並且它可以指定可延續訂閱者是否要接收其自己的連線所發佈的訊息。

用來建立可延續訂閱者的階段作業必須有相關聯的用戶端 ID。用戶端 ID 與用來建立階段作業的連線相關聯的用戶端 ID 相同；其指定方式如第 20 頁的『[ConnectionFactories 和 Connection 物件](#)』中所述。

識別可延續訂閱的名稱在用戶端 ID 內必須是唯一的，因此用戶端 ID 會構成可延續訂閱完整唯一 ID 的一部分。傳訊伺服器會維護可延續訂閱的記錄，並確保在主題上發佈的所有訊息都會保留下來，直到可延續訂閱者確認它們或它們到期為止。

即使在可延續訂閱者關閉之後，傳訊伺服器仍會繼續維護可延續訂閱的記錄。若要重複使用先前建立的可延續訂閱，應用程式必須建立可延續訂閱者，並指定相同的訂閱名稱，並使用與可延續訂閱相關聯且具有相同用戶端 ID 的階段作業。一次只能有一個階段作業具有特定可延續訂閱的可延續訂閱者。

可延續訂閱的範圍是維護訂閱記錄的傳訊伺服器。如果連接至不同傳訊伺服器的兩個應用程式各使用相同的訂閱名稱及用戶端 ID 來建立可延續訂閱者，則會建立兩個完全獨立的可延續訂閱。

若要刪除可延續訂閱，應用程式會呼叫「階段作業」物件的「取消訂閱」方法，並將識別可延續訂閱的名稱指定為參數。與階段作業相關聯的用戶端 ID 必須與與可延續訂閱相關聯的用戶端 ID 相同。傳訊伺服器會刪除它所維護的可延續訂閱記錄，且不會再傳送任何訊息給可延續訂閱者。

若要變更現有訂閱，應用程式可以使用相同的訂閱名稱及用戶端 ID 來建立可延續訂閱者，但指定不同的主題及/或訊息選取器。變更可延續訂閱相當於刪除訂閱並建立新的訂閱。

對於連接至 WebSphere MQ V7.0 佇列管理程式的應用程式，XMS 會管理訂閱者佇列。因此，應用程式不需要指定訂閱者佇列。XMS 將忽略訂閱者佇列 (如果指定的話)。

不過，對於連接至 WebSphere MQ V6.0 佇列管理程式的應用程式，每一個可延續訂閱者都必須有指定的訂閱者佇列。若要指定主題的訂閱者佇列名稱，請設定代表主題之 `Destination` 物件的 `XMSC_WMQ_DUR_SUBQ` 內容。預設訂閱者佇列為 `SYSTEM.JMS.D.SUBSCRIBER.QUEUE`。

連接至 WebSphere MQ V6.0 佇列管理程式的可延續訂閱者可以共用單一訂閱者佇列，或者每一個可延續訂閱者可以從自己的專用訂閱者佇列擷取其訊息。如需討論應用程式採用的方法，請參閱 *WebSphere MQ 使用 Java*。

請注意，您無法變更可延續訂閱的訂閱者佇列。變更訂閱者佇列的唯一方法是刪除訂閱並建立新的訂閱。

對於連接至服務整合匯流排的應用程式，每一個可延續訂閱者必須具有指定的可延續訂閱起始目錄。如果要指定使用相同連線之所有可延續訂閱者的可延續訂閱起始目錄，請設定用來建立連線之 `ConnectionFactory` 物件的 `XMSC_WPM_DUR_SUB_HOME` 內容。若要指定個別主題的可延續訂閱起始目錄，請設定代表主題之 `Destination` 物件的 `XMSC_WPM_DUR_SUB_HOME` 內容。必須先指定連線的可延續訂閱起始目錄，應用程式才能建立使用連線的可延續訂閱者。指定給目的地的任何值都會置換指定給連線的值。

不可延續訂閱者

不可延續訂閱者是訊息消費者，只會接收在訂閱者作用中時所發佈的訊息。當訂閱者處於非作用中狀態時所遞送的訊息會遺失。

只有在您使用 WebSphere MQ V6.0 佇列管理程式上的發佈/訂閱傳訊時，這個區段中的資訊才相關。

如果在關閉連線之前或期間未刪除消費者物件，則對於不再作用中的訂閱者，訊息可以保留在分配管理系統佇列上。

在此狀況下，可以使用 WebSphere MQ Classes for JMS 隨附的 `Cleanup` 公用程式來清除這些訊息的佇列。*WebSphere MQ 使用 Java* 中提供如何使用此公用程式的詳細資料。如果此佇列上剩餘大量訊息，您可能需要增加訂閱者佇列的佇列深度。

同步訊息消費者

同步訊息消費者會同步從佇列接收訊息。

同步訊息消費者一次接收一則訊息。當使用 `Receive(wait interval)` 方法時，呼叫只會等待訊息的指定時段 (毫秒)，或直到訊息消費者關閉為止。

如果使用 `ReceiveNoWait ()` 方法，則同步訊息消費者會接收訊息而不會延遲；如果下一則訊息可用，則會立即接收它，否則會傳回空值訊息物件的指標。

非同步訊息消費者

非同步訊息消費者會非同步接收來自佇列的訊息。每當佇列上有新的訊息可用時，就會呼叫應用程式所登錄的訊息接聽器。

XMS 中的有害訊息

有害訊息是指無法由接收端 MDB 應用程式處理的訊息。如果發現有害訊息，XMS MessageConsumer 物件可以根據兩個佇列內容 (BOQUEUE 和 BOTHRESH) 將它重新排入佇列。

在某些情況下，遞送至 MDB 的訊息可能會回復到 IBM WebSphere MQ 佇列。例如，如果在後續回復的工作單元內遞送訊息，則會發生此情況。通常會重新遞送已回復的訊息，但格式不正確的訊息可能會反覆地導致 MDB 失敗，因此無法遞送。這類訊息稱為有害訊息。您可以配置 IBM WebSphere MQ，讓有害訊息自動傳送至另一個佇列，以進一步調查或捨棄。如需如何以此方式配置 IBM WebSphere MQ 的相關資訊，請參閱處理 ASF 中的有害訊息。

有時，格式不正確的訊息會到達佇列。在此環境定義中，格式不正確表示接收端應用程式無法正確處理訊息。這類訊息會導致接收端應用程式失敗，並退出這個格式不正確的訊息。然後，訊息可以反覆地遞送至輸入佇列，並由應用程式反覆地取消。這些訊息稱為有害訊息。XMS MessageConsumer 物件會偵測有毒訊息，並將它們重新遞送至替代目的地。

IBM WebSphere MQ 佇列管理程式會記錄每一則訊息已取消的次數。當此數目達到可配置的臨界值時，訊息消費者會將訊息重新排入指名的取消佇列。如果此重新排入佇列因任何原因而失敗，則會從輸入佇列中移除訊息，並重新排入無法傳送郵件的佇列，或捨棄該訊息。

XMS ConnectionConsumer 物件會以相同的方式並使用相同的佇列內容來處理有毒訊息。如果多個連線消費者正在監視相同的佇列，則在重新排入佇列之前，有害訊息可能遞送至應用程式的次數可能會超過臨界值。此行為是由於個別連線消費者監視佇列及重新排入佇列有害訊息的方式。

臨界值及回復佇列的名稱是 IBM WebSphere MQ 佇列的屬性。屬性名稱為 BackoutThreshold 及 BackoutRequeue 完整名稱。它們適用的佇列如下：

- 對於點對點傳訊，這是基礎本端佇列。當訊息消費者和連線消費者使用佇列別名時，這很重要。
- 對於 IBM WebSphere MQ 傳訊提供者標準模式中的發佈/訂閱傳訊，它是從中建立主題受管理佇列的模型佇列。
- 對於 IBM WebSphere MQ 傳訊提供者移轉模式中的發佈/訂閱傳訊，它是在 TopicConnectionFactory 物件上定義的 CCSUB 佇列，或在 Topic 物件上定義的 CCDSUB 佇列。

若要設定 BackoutThreshold 及 BackoutRequeue 完整名稱屬性，請發出下列 MQSC 指令：

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

對於發佈/訂閱傳訊，如果您的系統為每一個訂閱建立動態佇列，則會從 JMS 模型佇列 SYSTEM.JMS.MODEL.QUEUE 的 WebSphere MQ 類別取得這些屬性值。若要變更這些設定，請使用：

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

如果取消臨界值為零，則會停用有害訊息處理，且有害訊息會保留在輸入佇列上。否則，當取消計數達到臨界值時，會將訊息傳送至指名的取消佇列。

如果取消計數達到臨界值，但訊息無法跳至取消佇列，則會將訊息傳送至無法傳送郵件的佇列，或如果訊息是非持續性，則會捨棄該訊息。

如果未定義取消佇列，或如果 MessageConsumer 物件無法將訊息傳送至取消佇列，則會發生此狀況。

配置系統以執行有毒訊息處理

XMS .NET 在查詢 BOTHRESH 和 BOQNAME 屬性時所使用的佇列，取決於所執行的傳訊樣式：

- 對於點對點傳訊，這是基礎本端佇列。當 XMS .NET 應用程式使用別名佇列或叢集佇列中的訊息時，這很重要。

- 對於發佈/訂閱傳訊，會建立受管理佇列來保留應用程式的訊息。XMS .NET 會查詢受管理佇列，以判定 **BOTHRESH** 及 **BOQNAME** 屬性的值。

受管理佇列是從與應用程式已訂閱之「主題」物件相關聯的模型佇列建立，並從模型佇列繼承 **BOTHRESH** 及 **BOQNAME** 屬性的值。使用的模型佇列取決於接收端應用程式是否已取出可延續或不可延續訂閱：

- 用於可延續訂閱的模型佇列由主題的 **MDURMDL** 屬性指定。此屬性的預設值為 `SYSTEM.DURABLE.MODEL.QUEUE`。
- 對於不可延續訂閱，使用的模型佇列由 **MNDURMDL** 屬性指定。**MNDURMDL** 屬性的預設值是 `SYSTEM.NDURABLE.MODEL.QUEUE`。

查詢 **BOTHRESH** 和 **BOQNAME** 屬性時，XMS .NET：

- 開啟本端佇列或別名佇列的目標佇列。
- 查詢 **BOTHRESH** 及 **BOQNAME** 屬性。
- 關閉本端佇列或別名佇列的目標佇列。

開啟本端佇列或別名佇列的目標佇列時所使用的開啟選項，視所使用的 IBM MQ 版本而定：

- 對於 8.0.0 版 Fix Pack 13 及更早版本，如果本端佇列或別名佇列的目標佇列是叢集佇列，則 XMS .NET 會使用 `MQOO_INPUT_AS_Q_DEF`、`MQOO_INQUIRE` 及 `MQOO_FAIL_IF QUIESCING` 選項開啟佇列。這表示執行接收端應用程式的使用者必須具有叢集佇列本端實例的查詢及存取權。

XMS .NET 會使用開啟選項 `MQOO_INQUIRE` 及 `MQOO_FAIL_IF QUIESCING` 開啟所有其他類型的本端佇列。為了讓 XMS .NET 查詢屬性的值，執行接收端應用程式的使用者必須對本端佇列具有查詢存取權。

- V 8.0.0.14** 使用 8.0.0 版 Fix Pack 14 中的 XMS .NET 時，不論佇列類型為何，執行接收端應用程式的使用者必須具有本端佇列的查詢存取權。

若要將有害訊息移至取消佇列或佇列管理程式的無法傳送郵件佇列，您必須授與執行應用程式 `put` 及 `passall` 權限的使用者。

處理 ASF 中的有害訊息

當您使用「應用程式伺服器機能 (ASF)」時，`ConnectionConsumer` (而非 `MessageConsumer`) 會處理有毒訊息。`ConnectionConsumer` 會根據佇列的 `BackoutThreshold` 及 `BackoutRequeue` 完整名稱內容來要求訊息。

當應用程式使用 `ConnectionConsumers` 時，取消訊息的情況取決於應用程式伺服器提供的階段作業：

- 當階段作業非交易式時，若有 `AUTO_ANACK` 或 `DUPS_OK_ANACK`，則只有在系統錯誤之後，或應用程式非預期地終止時，才會取消訊息。
- 當階段作業與 `CLIENT_ACKNOWLEDGE` 非交易時，呼叫 `Session.recover()` 的應用程式伺服器可以取消未確認的訊息。

一般而言，`MessageListener` 或應用程式伺服器的用戶端實作會呼叫 `Message.acknowledge()`。`Message.acknowledge()` 確認到目前為止在階段作業上遞送的所有訊息。

- 當交易階段作業時，呼叫 `Session.rollback()` 的應用程式伺服器可以取消未確認的訊息。

佇列瀏覽器

應用程式使用佇列瀏覽器來瀏覽佇列上的訊息，而不移除它們。

若要建立佇列瀏覽器，應用程式會呼叫 `ISession` 物件的「建立佇列瀏覽器」方法，並將目的地物件指定為參數，以識別要瀏覽的佇列。應用程式可以使用或不使用訊息選取器來建立佇列瀏覽器。

建立佇列瀏覽器之後，應用程式可以呼叫 `IQueueBrowser` 物件的 `GetEnumerator` 方法，以取得佇列上的訊息清單。此方法會傳回封裝「訊息」物件清單的列舉元。清單中「訊息」物件的順序與從佇列擷取訊息的順序相同。然後，應用程式可以依序使用列舉元來瀏覽每一個訊息。

當訊息放置在佇列上並從佇列中移除時，會動態更新列舉元。每次應用程式呼叫 `IEnumerator.MoveNext()` 來瀏覽佇列上的下一個訊息時，該訊息會反映佇列的現行內容。

應用程式可以針對給定的佇列瀏覽器多次呼叫 `GetEnumerator` 方法。每一個呼叫都會傳回新的列舉元。因此，應用程式可以使用多個列舉元來瀏覽佇列上的訊息，並在佇列內維護多個位置。

應用程式可以使用佇列瀏覽器來搜尋要從佇列中移除的適當訊息，然後使用具有訊息選取器的訊息消費者來移除訊息。訊息選取器可以根據 `JMSMessageID` 標頭欄位的值來選取訊息。如需此及其他 JMS 訊息標頭欄位的相關資訊，請參閱第 59 頁的『XMS 訊息中的標頭欄位』。

要求者

應用程式使用要求者來傳送要求訊息，然後等待及接收回覆。

許多傳訊應用程式都是以傳送要求訊息然後等待回覆的演算法為基礎。XMS 提供稱為 Requestor 的類別，以協助開發這種應用程式樣式。

為了建立要求者，應用程式會呼叫 Requestor 類別的 Create Requestor 建構子，並指定 Session 物件和 Destination 物件作為參數，以識別要傳送要求訊息的位置。階段作業不得交易，也不得具有 XMSC_CLIENT_ACKNOWLEDGE 確認模式。建構子會自動建立要傳送回覆訊息的暫時佇列或主題。

建立要求者之後，應用程式可以呼叫 Requestor 物件的 Request 方法來傳送要求訊息，然後等待並接收來自接收要求訊息之應用程式的回覆。呼叫會等待直到收到回覆或直到階段作業結束 (以先出現者為準)。對於每一個要求訊息，要求者只需要一個回覆。

當應用程式關閉要求者時，會刪除暫時佇列或主題。不過，相關聯的階段作業不會關閉。

物件刪除

當應用程式刪除它所建立的 XMS 物件時，XMS 會釋放已配置給該物件的內部資源。

當應用程式建立 XMS 物件時，XMS 會將記憶體及其他內部資源配置給物件。XMS 會保留這些內部資源，直到應用程式透過呼叫物件的關閉或刪除方法明確刪除物件為止，此時 XMS 會釋放內部資源。如果應用程式嘗試刪除已刪除的物件，則會忽略呼叫。

當應用程式刪除「連線」或「階段作業」物件時，XMS 會自動刪除某些相關聯的物件，並釋放其內部資源。這些物件是由「連線」或「階段作業」物件所建立，且沒有與物件無關的功能。這些物件顯示在第 33 頁的表 4 中。

註: 如果應用程式關閉與相依階段作業的連線，則也會刪除所有相依於那些階段作業的物件。只有「連線」或「階段作業」物件可以有相依物件。

已刪除物件	方法	自動刪除的相依物件
連線	關閉連線	ConnectionMeta 資料和階段作業物件
Session	關閉階段作業	MessageConsumer、MessageProducer、QueueBrowser 及 Requestor 物件

透過 XMS 進行受管理 WebSphere MQ XA 交易

受管理 WebSphere MQ XA 交易可以透過 XMS 來使用。

若要透過 XMS 使用 XA 交易，必須建立交易式階段作業。當使用 XA 交易時，交易控制是透過「分散式交易協調程式 (DTC)」廣域交易，而不是透過 XMS 階段作業。使用 XA 交易時，無法在 XMS 階段作業上發出 `Session.commit` 或 `Session.rollback`。請改用 `Transscope.Commit` 或 `Transscope.Rollback` DTC 方法來確定或回復交易。如果階段作業用於 XA 交易，則使用階段作業建立的生產者或消費者必須是 XA 交易的一部分。它們無法用於 XA 交易範圍以外的任何作業。它們無法用於 XA 交易外部的作業，例如 `Producer.send` 或 `Consumer.receive`。

如果發生下列情況，則會擲出 `IllegalStateException` 異常狀況物件

- XA 交易式階段作業用於 `Session.commit` 或 `Session.rollback`。
- XA 交易階段作業中使用過一次的生產者或消費者物件會在 XA 交易範圍之外使用。

非同步消費者不支援 XA 交易。

註:

1. 在 XA 交易確定之前，可能會在 Producer、Consumer、Session 或 Connection 物件上發出關閉。在這種情況下，會回復交易中的訊息。同樣地，如果在 XA 交易確定之前中斷連線，則會回復交易中的所有訊息。對於 Producer 物件，回復表示訊息不會放置在佇列上。對於 Consumer 物件，回復表示訊息保留在佇列上。
2. 如果 Producer 物件將具有 TimeToLive 的訊息放置在 TransactionScope 中，並且在經歷時間之後發出 commit，則該訊息可能會在發出 commit 之前到期。在此情況下，訊息無法供 Consumer 物件使用。
3. 執行緒之間不支援 Session 物件。不支援使用與執行緒之間共用的 Session 物件的交易。

XMS 初始類型

XMS 提供八個 Java 初始類型 (byte、short、int、long、float、double、char 及 boolean) 的對等項目。這容許在 XMS 與 JMS 之間交換訊息，而不會失去或毀損資料。

第 34 頁的表 5 列出 Java 相等的資料類型、大小，以及每一個 XMS 初始類型的最小值和最大值。

XMS 資料類型	相容 Java 資料類型	大小	最小值	最大值
System.Boolean	布林值	32 位元	false	true
System.SBYTE	位元組	8 位元	-2^7 (-128)	2^7-1 (127)
System.BYTE	位元組	8 位元	-2^7 (-128)	2^7-1 (127)
System.CHAR	位元組	8 位元	-2^7 (-128)	2^7-1 (127)
System.Int16	短	16 位元	-2^{15} (-32768)	$2^{15}-1$ (32767)
System.Int32	整數	32 位元	-2^{31} (-2147483648)	$2^{31}-1$ (2147483647)
System.Int64	長整數	64 位元	-2^{63} (-9223372036854775808)	$2^{63}-1$ (9223372036854775807)
System.Single	浮點數	32 位元	-3.402823E-38 (至 7 位數精準度)	3.402823E+38 (至 7 位數精準度)
System.Double	倍精準數	64 位元	-1.79769313486231E-308 (至 15 位數精準度)	1.79769313486231E+308 (至 15 位數精準度)

相關概念

物件的屬性和內容

XMS 物件可以具有以不同方式實作的屬性及內容，這些屬性及內容是物件的性質。

將內容值從一個資料類型隱含轉換為另一個資料類型

當應用程式取得內容的值時，XMS 可以將該值轉換為另一個資料類型。許多規則會控管支援哪些轉換，以及 XMS 如何執行轉換。

相關參考

應用程式資料元素的資料類型

為了確保 XMS 應用程式可以與 WebSphere MQ for JMS 類別 應用程式交換訊息，這兩個應用程式必須能夠以相同方式解譯訊息內文中的應用程式資料。

將內容值從一個資料類型隱含轉換為另一個資料類型

當應用程式取得內容的值時，XMS 可以將該值轉換為另一個資料類型。許多規則會控管支援哪些轉換，以及 XMS 如何執行轉換。

物件的內容具有名稱和值；值具有相關聯的資料類型，其中內容的值也稱為 內容類型。

應用程式使用 `PropertyContext` 類別的方法來取得和設定物件的內容。為了取得內容的值，應用程式會呼叫適用於內容類型的方法。例如，若要取得整數內容的值，應用程式通常會呼叫 `GetIntProperty` 方法。

不過，當應用程式取得內容的值時，XMS 可以將該值轉換成另一個資料類型。例如，若要取得整數內容的值，應用程式可以呼叫 `GetStringProperty` 方法，它會以字串形式傳回內容的值。第 35 頁的表 6 顯示 XMS 支援的轉換。

內容類型	支援的目標資料類型
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
System.SByte 陣列	System.String
System.Int16	System.String, System.Int32, System.Int64

下列一般規則控管支援的轉換：

- 如果在轉換期間未遺失任何資料，則可以將數值內容值從一個資料類型轉換為另一個資料類型。例如，資料類型為 `System.Int32` 的內容值可以轉換成資料類型為 `System.Int64` 的值，但無法轉換成資料類型為 `System.Int16` 的值。
- 任何資料類型的內容值都可以轉換成字串。
- 字串內容值可以轉換為任何其他資料類型，前提是字串已正確格式化以進行轉換。如果應用程式嘗試轉換未正確格式化的字串內容值，XMS 可能會傳回錯誤。
- 如果應用程式嘗試不支援的轉換，XMS 可能會傳回錯誤。

當內容值從一種資料類型轉換為另一種資料類型時，會套用下列規則：

- 將布林內容值轉換成字串時，值 `true` 會轉換成字串 `"true"`，值 `false` 會轉換成字串 `"false"`。
- 將布林內容值轉換為數值資料類型 (包括 `System.SByte`) 時，值 `true` 會轉換為 `1`，而值 `false` 會轉換為 `0`。
- 將字串內容值轉換成布林值時，字串 `"true"` (不區分大小寫) 或 `"1"` 會轉換成 `true`，而字串 `"false"` (不區分大小寫) 或 `"0"` 會轉換成 `false`。無法轉換所有其他字串。
- 將字串內容值轉換成資料類型為 `System.Int32`、`System.Int64`、`System.SByte` 或 `System.Int16` 的值時，字串必須具有下列格式：

[空白] *[sign]digits*

字串元件定義如下：

空白

選用前導空白字元。

符號

選用加號 (+) 或減號 (-) 字元。

數字

一連串連續的數字字元 (0-9)。至少必須呈現一個數字字元。

在一連串數字字元之後，字串可以包含非數字字元的其他字元，但一旦達到這些字元的第一個字元，轉換就會停止。假設字串代表十進位整數。

如果字串格式不正確， XMS 可能會傳回錯誤。

- 當將字串內容值轉換成資料類型為 System.Double 或 System.Float 的值時，字串必須具有下列格式：

[空白] [sign] [digits] [point[d_digits]] [e_char[e_sign]e_digits]

字串元件定義如下：

空白

(選用) 前導空白字元。

符號

(選用) 加號 (+) 或減號 (-) 字元。

數字

一連串連續的數字字元 (0-9)。數字 或 *d_digits* 中必須至少存在一個數字字元。

點

(選用) 小數點 (.)。

數字

一連串連續的數字字元 (0-9)。數字 或 *d_digits* 中必須至少存在一個數字字元。

字元

指數字元，可以是 *E* 或 *e*。

_ 簽署

(選用) 指數的加號 (+) 或減號 (-) 字元。

數字 (*D*)

指數的連續數字字元 (0-9)。如果字串包含指數字元，則至少必須存在一個數字字元。

在一連串數字字元或代表指數的選用字元之後，字串可以包含不是數字字元的其他字元，但一旦達到這些字元的第一個字元，就會停止轉換。假設字串代表具有 10 次方的指數的十進位浮點數字。

如果字串格式不正確， XMS 可能會傳回錯誤。

- 將數值內容值轉換為字串 (包括資料類型為 System.SByte 的內容值) 時，該值會轉換為以十進位數表示的值字串，而不是包含該值的 ASCII 字元的字串。例如，整數 65 會轉換為字串 "65"，而不是字串 "A"。
- 將位元組陣列內容值轉換為字串時，每個位元組都會轉換為代表位元組的 2 個十六進位字元。例如，位元組陣列 {0xF1, 0x12, 0x00, 0xFF} 會轉換為字串 "F11200FF"。

Property 和 PropertyContext 類別的方法都支援從內容類型轉換為其他資料類型。

相關概念

物件的屬性和內容

XMS 物件可以具有以不同方式實作的屬性及內容，這些屬性及內容是物件的性質。

XMS 初始類型

XMS 提供八個 Java 初始類型 (byte、short、int、long、float、double、char 及 boolean) 的對等項目。這容許在 XMS 與 JMS 之間交換訊息，而不會失去或毀損資料。

相關參考

對映訊息

對映訊息的內文包含一組名稱/值配對，其中每一個值都有相關聯的資料類型。

串流訊息

串流訊息的內文包含值串流，其中每一個值都有相關聯的資料類型。

應用程式資料元素的資料類型

為了確保 XMS 應用程式可以與 WebSphere MQ for JMS 類別 應用程式交換訊息，這兩個應用程式必須能夠以相同方式解譯訊息內文中的應用程式資料。

反覆運算子

反覆運算子會封裝物件清單，以及維護清單中現行位置的游標。Message Service Client for C/C++ 中提供的「反覆運算子」概念是使用 Message Service Client for .NET 中的 IEnumerator 介面來實作。

建立反覆運算子時，游標的位置位於第一個物件之前。應用程式會使用反覆運算子來依序擷取每一個物件。

Message Service Client for C/C++ 的「反覆運算子」類別相當於 Java 中的「列舉元」類別。XMS .NET 類似於 Java，並使用 IEnumerator 介面。

應用程式可以使用 IEnumerator 來執行下列作業：

- 取得訊息的內容
- 取得對映訊息內文中的名稱/值配對
- 瀏覽佇列上的訊息
- 取得連線所支援之 JMS 定義訊息內容的名稱

編碼字集 ID

在 XMS .NET 中，所有字串都是使用原生 .NET 字串來傳遞。由於這是固定編碼，因此不需要進一步資訊來解譯它。因此，XMS .NET 應用程式不需要 XMSC_CLIENT_CCSD 內容。

XMS 錯誤碼和異常狀況碼

XMS 使用一系列錯誤碼來指出失敗。這些錯誤碼未在本文件中明確列出，因為它們可能會因不同版本而不同。只會記載 XMS 異常狀況碼 (格式為 XMS_X_...)，因為它們在 XMS 的不同版本之間仍然相同。

建置您自己的應用程式

您可以建置自己的應用程式，就像您建置範例應用程式一樣。

建置 .NET 應用程式，如第 18 頁的『建置 .NET 範例應用程式』中所述。這項作業也會列出建置您自己的 .NET 應用程式所需的必備項目。如需如何建置您自己的應用程式的其他指引，請使用為每一個範例應用程式提供的 make 檔。

提示: 如果要在失敗時協助診斷問題，您可能會發現編譯包含符號的應用程式會很有用。

相關概念

範例應用程式

範例應用程式提供每一個 API 的一般特性概觀。您可以使用它們來驗證安裝及傳訊伺服器設定，並協助您建置自己的應用程式。

相關參考

.NET 介面

此區段記載 .NET 類別介面及其內容和方法。

XMS 物件的內容

此章節記載 XMS 所定義的物件內容。

透過 XMS 自動重新連線 WebSphere MQ 用戶端

使用 WebSphere MQ V7.1 Client 以及更新版本作為訊息提供者時，將 XMS 用戶端配置成在網路、佇列管理程式或伺服器失敗之後自動重新連接。

使用 MQConnectionFactory 類別的 WMQ_CONNECTION_NAME_LIST 及 WMQ_CLIENT_RECONNECT_OPTIONS 內容，將用戶端連線配置為自動重新連接。自動用戶端重新連線會在連線失敗之後重新連接用戶端，或在停止佇列管理程式之後作為選項重新連線。部分用戶端應用程式的設計使得它們不適合自動重新連線。

建立連線之後，可自動重新連接的用戶端連線會變成可重新連接。

註: 也可以透過「用戶端通道定義表 (CCDT)」或透過 mqclient.ini 檔啟用用戶端重新連線，來設定內容用戶端重新連接選項、用戶端重新連接逾時及連線名單。

註: 如果在 ConnectionFactory 物件上以及 CCDT 中設定重新連線內容，則優先順序規則如下所示。如果在 ConnectionFactory 物件中設定連線名稱清單內容的預設值，則 CCDT 優先。如果連線名稱清單未設為其預設值，則 ConnectionFactory 物件中設定的內容值優先。連線名單的預設值為 localhost(1414)。

撰寫 XMS .NET 應用程式

此章節 提供在撰寫 XMS.NET 應用程式時協助您的資訊。

章節 包含下列 區段:

- [第 38 頁的『.NET 的資料類型』](#)
- [第 39 頁的『.NET 中的受管理及未受管理作業』](#)
- [第 39 頁的『.NET 中的目的地』](#)
- [第 40 頁的『.NET 中的內容』](#)
- [第 41 頁的『.NET 中不存在的內容處理』](#)
- [第 41 頁的『.NET 中的錯誤處理』](#)
- [第 41 頁的『.NET 中的訊息和異常狀況接聽器』](#)

相關概念

撰寫 XMS 應用程式

此 章節 提供資訊來協助您撰寫 XMS 應用程式。

相關參考

.NET 介面

此 區段 記載 .NET 類別介面及其內容和方法。

.NET 的資料類型

XMS .NET 支援 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、及 System.Object。XMS .NET 的資料類型不同於 XMS C++ 的資料類型。 您可以使用此 章節 來識別對應的資料類型。

下表顯示對應的 XMS .NET 及 XMS C++ 資料類型，並簡要說明它們。

XMS .NET 類型	XMS C++ 類型	說明
System.SByte	xmsSByte xmsINT8	帶正負號的 8 位元值
System.Byte	xmsBYTE xmsUINT8	不帶正負號的 8 位元值
System.Int16	xmsINT16 xmsSHORT	帶正負號的 16 位元值
System.UInt16	xmsUINT16 xmsUSHORT	不帶正負號的 16 位元值
System.Int32	xmsINT32 xmsINT	帶正負號的 32 位元值
System.UInt32	xmsUINT32 xmsUINT	不帶正負號的 32 位元值
System.Int64	xmsLONG xmsINT64	帶正負號的 64 位元值

表 7: XMS .NET 和 XMS C++ 的資料類型 (繼續)

XMS .NET 類型	XMS C++ 類型	說明
System.UInt64	xmsULONG xmsUINT64	不帶正負號的 64 位元值
System.Char	xmsCHAR16	不帶正負號的 16 位元字元 (Unicode for .NET)
System.Single	xmsFLOAT	IEEE 32 位元浮點數
System.Double	xmsDOUBLE	IEEE 64 位元浮點數
System.Boolean	xmsBOOL	True/False 值
不適用	xmsCHAR	已簽署或未簽署的 8 位元值 (已簽署或未簽署視平台而定)
System.Decimal	不適用	96 位元帶正負號整數乘以 10^0 到 10^{28}
System.Object	不適用	所有類型的基礎
System.String	不適用	字串類型

.NET 中的受管理及未受管理作業

受管理程式碼專門在 .NET 一般語言執行時期環境內執行，完全相依於該執行時期所提供的服務。如果應用程式的任何部分在 .NET 一般語言執行時期環境之外執行或呼叫服務，則應用程式會分類為未受管理。

受管理 .NET 環境目前無法支援某些進階功能。

如果您的應用程式需要在完全受管理環境中目前不支援的功能，則您可以將應用程式變更為使用未受管理的環境，而不需要對應用程式進行重大變更。不過，您應該注意，在進行這項選取時，XMS 堆疊會利用未受管理的程式碼。

與 WebSphere MQ 佇列管理程式的連線

WMQ_CM_CLIENT 的受管理連線不支援非 TCP 通訊及通道壓縮。不過，可以使用未受管理的連線 (WMQ_CM_CLIENT_UNMANAGED) 來支援這些連線。如需相關資訊，請參閱 [開發 .NET 應用程式](#)。

如果您在未受管理環境中從受管理物件建立 Connection Factory，則必須手動將連線模式的值變更為 XMSC_WMQ_CM_CLIENT_UNMANAGED。

WebSphere 服務 Integration Bus 傳訊引擎的連線

目前不支援連線至需要使用 SSL 通訊協定 (包括 HTTP) 的 WebSphere 服務整合匯流排傳訊引擎作為受管理程式碼。

相關參考

[XMSC_WMQ_CONNECTION_MODE](#)

.NET 中的目的地

在 .NET 中，目的地是根據通訊協定類型來建立，且只能在為其建立目的地的通訊協定類型上使用。

提供兩個功能來建立目的地，一個用於主題，另一個用於佇列：

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

這些函數可在 API 中的下列兩個物件上使用：

- `ISession`

- XMSFactoryFactory

在這兩種情況下，這些方法都可以接受 URI 樣式字串，其中可以包含參數，格式如下：

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

或者，這些方法只能接受目的地名稱，即沒有 `topic://` 或 `queue://` 字首且沒有參數的名稱。

這表示下列 URI 樣式字串：

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

會產生與下列目的地名稱相同的結果：

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

對於 WebSphere 服務 Integration Bus JMS，主題也可以速記形式指定，其中包括 `topicname` 和 `topicspace`，但不能包括參數：

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

.NET 中的內容

.NET 應用程式使用 `PropertyContext` 介面中的方法來取得和設定物件的內容。

`PropertyContext` 介面會封裝取得和設定內容的方法。這些方法由下列類別直接或間接繼承：

- [ByteMessage](#)
- [連線](#)
- [ConnectionFactory](#)
- [ConnectionMeta](#) 資料
- [目的地](#)
- [MapMessage](#)
- [訊息](#)
- [MessageConsumer](#)
- [MessageProducer](#)
- [ObjectMessage](#)
- [QueueBrowser](#)
- [階段作業](#)
- [StreamMessage](#)
- [TextMessage](#)

如果應用程式設定內容的值，新值會取代內容先前的任何值。

如需 XMS 內容的相關資訊，請參閱第 159 頁的『XMS 物件的內容』。

為了方便使用，.NET 中的 XMS 內容名稱和值預先定義為稱為 XMSC 的結構中的公用常數。這些常數的名稱格式為 `XMSC.<constant>`；例如，`XMSC.USERID` (內容名稱常數) 和 `XMSC.DELIVERY_AS_APP` (值常數)。

此外，您可以使用 `IBM.XMS.MQC` 結構來存取 WebSphere MQ 常數。如果已匯入 `IBM.XMS` 名稱空間，您可以使用 `MQC.<constant>` 格式來存取這些內容的值。例如，`MQC.MQRO_COA_WITH_FULL_DATA`。

此外，如果您有混合式應用程式同時使用 XMS .NET 和 WebSphere MQ classes for .NET，且同時匯入 `IBM.XMS` 和 `IBM.WMQ` 名稱空間，然後您必須完整限定 `MQC` 結構名稱空間，以確保每一個出現項目都是唯一的。

受管理 .NET 環境目前不支援部分進階功能。如需詳細資料，請參閱 [第 39 頁的『.NET 中的受管理及未受管理作業』](#)。

.NET 中不存在的內容處理

XMS .NET 中不存在內容的處理與 JMS 規格以及 XMS 的 C 和 C++ 實作大致一致。

在 JMS 中，當方法嘗試將不存在 (空值) 的值轉換成所需的類型時，存取不存在的內容可能會導致 Java 系統異常狀況。如果內容不存在，則會發生下列異常狀況：

- getString 內容和 getObject 內容傳回空值
- getBoolean 內容傳回 false，因為 Boolean.valueOf(null) 傳回 false
- getInt 內容 etc.throw java.lang.NumberFormatException，因為 Integer.valueOf(null) 擲出異常狀況

如果內容不存在於 XMS .NET 中，則會發生下列異常狀況：

- GetString 內容和 GetObject 內容 (以及 GetBytes 內容) 會傳回空值 (與 Java 相同)
- GetBoolean 內容擲出 System.NullReferenceException
- GetInt 內容等會擲出 System.NullReferenceException

此實作與 Java 不同，但與 JMS 規格以及 XMS C 和 C++ 介面大致一致。如同 Java 實作，XMS .NET 會將 System.Convert 呼叫中的任何異常狀況傳播給呼叫端。不過，與 Java 不同，XMS 會明確擲出 NullReference 異常狀況，而不只是透過將空值傳遞至系統轉換常式來使用 .NET 架構的原生行為。如果您的應用程式將內容設為 "abc" 之類的字串，並呼叫 GetInt 內容，則 Convert.ToInt32("abc") 會擲出 System.FormatException 會延伸到呼叫端，這與 Java 一致。MessageFormat 只有在用於 setProperty 和 getProperty 的類型不相容時，才會擲出異常狀況。此行為也與 Java 一致。

.NET 中的錯誤處理

XMS .NET 異常狀況全都衍生自 System.Exception。XMS 方法呼叫可能會擲出特定的 XMS 異常狀況，例如 MessageFormat 異常狀況、一般 XMSExceptions，或系統異常狀況，例如 NullReference 異常狀況。

撰寫應用程式以根據應用程式需求，在特定 catch 區塊或一般 System.Exception catch 區塊中捕捉任何這些錯誤。

.NET 中的訊息和異常狀況接聽器

.NET 應用程式使用訊息接聽器來非同步接收訊息，並使用異常狀況接聽器來非同步通知連線發生問題。

對於 .NET 和 C++，訊息和異常狀況接聽器的功能是相同的。不過，有一些小小的實作差異。

.NET 中的訊息接聽器

若要非同步接收訊息，您必須完成下列步驟：

1. 定義符合訊息接聽器委派簽章的方法。您定義的方法可以是靜態或實例方法，並且可以在任何可存取類別中定義。委派簽章如下：

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

因此您可以將方法定義為：

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

2. 使用與下列類似的項目，將此方法實例化為委派：

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

3. 將委派設為消費者的 MessageListener 內容，以向一或多個消費者登錄：

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

您可以將 MessageListener 設回空值，以移除委派：

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

.NET 中的異常狀況接聽器

異常狀況接聽器的運作方式與訊息接聽器相同，但具有不同的委派定義，且會指派給連線，而不是訊息消費者。這與 C++ 的相同。

1. 定義方法。委派簽章如下：

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

因此所定義的方法可能如下：

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

2. 使用類似下列的項目，將此方法實例化為委派：

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

3. 透過設定其 ExceptionListener 內容，向連線登錄委派：

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

您可以將 ExceptionListener 重設為：

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

當沒有任何參照保留時，系統記憶體回收器會自動刪除異常狀況或訊息。

下列是範例程式碼：

```
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);
}
}

```

使用受管理物件

此章節提供受管理物件的相關資訊。XMS 應用程式可以從中央受管理物件儲存庫擷取物件定義，並使用它們來建立 Connection Factory 及目的地。

章節包含下列區段：

- [第 43 頁的『受管理物件儲存庫的受支援類型』](#)
- [第 44 頁的『受管理物件的內容對映』](#)
- [第 45 頁的『受管理 Connection Factory 物件的必要內容』](#)
- [第 47 頁的『受管理目的地物件的必要內容』](#)
- [第 48 頁的『建立受管理物件』](#)
- [第 49 頁的『InitialContext 物件』](#)
- [第 50 頁的『InitialContext 內容』](#)
- [第 51 頁的『XMS 起始環境定義的 URI 格式』](#)
- [第 53 頁的『JNDI 查閱 Web 服務』](#)
- [第 54 頁的『擷取受管理物件』](#)

相關概念

受管理物件

使用受管理物件，您可以管理要從中央儲存庫管理的用戶端應用程式所使用的連線設定。應用程式會從中央儲存庫擷取物件定義，並使用它們來建立 Connection Factory 及 Destination 物件。使用受管理物件，您可以將應用程式與它們在執行時期使用的資源取消連結。

受管理物件儲存庫的受支援類型

「檔案系統」及 LDAP 受管理物件可以用來連接至 WebSphere MQ 及 WebSphere Application Server，而 COS 命名只能用來連接至 WebSphere Application Server。

「檔案系統」物件目錄採用序列化 Java 及命名目錄介面 (JNDI) 物件的形式。LDAP 物件目錄是包含 JNDI 物件的目錄。您可以使用 JMSAdmin 工具 (隨附於 WebSphere MQ v6.0) 或「WebSphere MQ 探險家」(隨附於 WebSphere MQ v7.0 以及更新版本) 來管理檔案系統及 LDAP 物件目錄。透過集中化 WebSphere MQ Connection Factory 和目的地，可以使用檔案系統和 LDAP 物件目錄來管理用戶端連線。網路管理者可以部署多個應用程式，這些應用程式會參照相同的中央儲存庫，且會自動更新以反映中央儲存庫中所做的連線設定變更。

COS 命名目錄包含 WebSphere 服務 Integration Bus Connection Factory 及目的地，並且可以使用 WebSphere Application Server 管理主控台來管理。若要讓 XMS 應用程式從 COS 命名目錄擷取物件，必須部署 JNDI 查閱 Web 服務。此 Web 服務並非在所有 WebSphere 服務整合技術上都可用。如需詳細資料，請參閱產品說明文件。

註: 重新啟動應用程式連線, 使物件目錄的變更生效。

相關概念

受管理物件的內容對映

如果要讓應用程式能夠使用 WebSphere MQ JMS 和 WebSphere Application Server Connection Factory 及目的地物件定義, 從這些定義擷取的內容必須對映至可以在 XMS Connection Factory 和目的地上設定的對應 XMS 內容。

InitialContext 內容

InitialContext 建構子的參數包括受管理物件的儲存庫位置, 以統一資源指示器 (URI) 形式提供。為了讓應用程式建立與儲存庫的連線, 可能需要提供比 URI 中包含的資訊更多的資訊。

XMS 起始環境定義的 URI 格式

受管理物件的儲存庫位置以統一資源指示器 (URI) 提供。URI 的格式視環境定義類型而定。

JNDI 查閱 Web 服務

若要從 XMS 存取 COS 命名目錄, 必須在 WebSphere 服務 Integration Bus 伺服器上部署 JNDI 查閱 Web 服務。此 Web 服務會將 Java 資訊從 COS 命名服務轉換成 XMS 應用程式可以讀取的表單。

擷取受管理物件

XMS 會使用建立 InitialContext 物件時所提供的位址, 或在 InitialContext 內容中, 從儲存庫擷取受管理物件。

受管理物件

使用受管理物件, 您可以管理要從中央儲存庫管理的用戶端應用程式所使用的連線設定。應用程式會從中央儲存庫擷取物件定義, 並使用它們來建立 ConnectionFactory 及 Destination 物件。使用受管理物件, 您可以將應用程式與它們在執行時期使用的資源取消連結。

相關工作

建立受管理物件

必須使用適當的管理工具來建立 XMS 應用程式建立傳訊伺服器連線所需的 ConnectionFactory 和「目的地」物件定義。

InitialContext 物件

應用程式必須建立起始環境定義, 以用來建立受管理物件儲存庫的連線, 以擷取必要的受管理物件。

相關參考

受管理 ConnectionFactory 物件的必要內容

當應用程式建立 Connection Factory 時, 必須定義一些內容來建立傳訊伺服器的連線。

受管理目的地物件的必要內容

建立目的地的應用程式必須在受管理的目的地物件上設定應用程式的數個內容。

受管理物件的內容對映

如果要讓應用程式能夠使用 WebSphere MQ JMS 和 WebSphere Application Server Connection Factory 及目的地物件定義, 從這些定義擷取的內容必須對映至可以在 XMS Connection Factory 和目的地上設定的對應 XMS 內容。

比方說, 如果要建立 XMS Connection Factory, 且其中含有從 WebSphere MQ JMS Connection Factory 擷取的內容, 這些內容必須在兩者之間對映。

會自動執行所有內容對映。

下表示範 Connection Factory 和目的地的部分最常用內容之間的對映。此表格中顯示的內容只是一小組範例, 並非所有顯示的內容都與所有連線類型及伺服器相關。

表 8: *Connection Factory* 和目的地內容的名稱對映範例

WebSphere MQ JMS 內容名稱	XMS 內容名稱	WebSphere 服務 Integration Bus 內容名稱
持續性 (每個)	<u>XMSC_DELIVERY_MODE</u>	
到期 (EXP)	<u>XMSC_TIME_TO_LIVE</u>	
優先順序 (PRI)	<u>XMSC_PRIORITY</u>	
	<u>XMSC_WPM_HOST_NAME</u>	serverName
	<u>XMSC_WPM_BUS_NAME</u>	busName
	<u>XMSC_WPM_TOPIC_SPACE</u>	topicName

相關概念

受管理物件儲存庫的受支援類型

「檔案系統」及 LDAP 受管理物件可以用來連接至 WebSphere MQ 及 WebSphere Application Server，而 COS 命名只能用來連接至 WebSphere Application Server。

InitialContext 內容

InitialContext 建構子的參數包括受管理物件的儲存庫位置，以統一資源指示器 (URI) 形式提供。為了讓應用程式建立與儲存庫的連線，可能需要提供比 URI 中包含的資訊更多的資訊。

XMS 起始環境定義的 URI 格式

受管理物件的儲存庫位置以統一資源指示器 (URI) 提供。URI 的格式視環境定義類型而定。

JNDI 查閱 Web 服務

若要從 XMS 存取 COS 命名目錄，必須在 WebSphere 服務 Integration Bus 伺服器上部署 JNDI 查閱 Web 服務。此 Web 服務會將 Java 資訊從 COS 命名服務轉換成 XMS 應用程式可以讀取的表單。

擷取受管理物件

XMS 會使用建立 InitialContext 物件時所提供的位址，或在 InitialContext 內容中，從儲存庫擷取受管理物件。

相關工作

建立受管理物件

必須使用適當的管理工具來建立 XMS 應用程式建立傳訊伺服器連線所需的 ConnectionFactory 和「目的地」物件定義。

InitialContext 物件

應用程式必須建立起始環境定義，以用來建立受管理物件儲存庫的連線，以擷取必要的受管理物件。

相關參考

受管理 ConnectionFactory 物件的必要內容

當應用程式建立 Connection Factory 時，必須定義一些內容來建立傳訊伺服器的連線。

受管理目的地物件的必要內容

建立目的地的應用程式必須在受管理的目的地物件上設定應用程式的數個內容。

IDestination (適用於 .NET 介面)

目的地是應用程式傳送訊息的地方，及（或）應用程式從中接收訊息的來源。

目的地的內容

「目的地」物件的內容概觀，以及指向更詳細參照資訊的鏈結。

ConnectionFactory (適用於 .NET 介面)

應用程式使用 Connection Factory 來建立連線。

ConnectionFactory 的內容

ConnectionFactory 物件內容的概觀，以及指向更詳細參照資訊的鏈結。

受管理 ConnectionFactory 物件的必要內容

當應用程式建立 Connection Factory 時，必須定義一些內容來建立傳訊伺服器的連線。

下表中列出的內容是應用程式設定以建立傳訊伺服器連線所需的最低內容。如果您想要自訂建立連線的方式，您的應用程式可以視需要設定 `ConnectionFactory` 物件的任何其他內容。如需相關資訊，請參閱第 160 頁的『`ConnectionFactory` 的內容』。包括可用內容的完整清單。

WebSphere MQ 佇列管理程式的連線

表 9: WebSphere MQ 佇列管理程式連線的受管理 <code>ConnectionFactory</code> 物件的內容設定	
必要 XMS	需要同等的 WebSphere MQ JMS 內容
<code>XMSC_CONNECTION_TYPE</code>	XMS 會從 <code>ConnectionFactory</code> 類別名稱和 <code>TRANSPORT (TRAN)</code> 內容來解決這個問題。
<code>XMSC_WMQ_HOST_NAME</code>	<code>HOSTNAME (HOST)</code>
<code>XMSC_WMQ_PORT</code>	<code>PORT</code>
<code>XMSC_WMQ_QUEUE_MANAGER</code>	佇列管理程式的名稱

與分配管理系統的即時連線

表 10: 即時連線至分配管理系統之受管理 <code>ConnectionFactory</code> 物件的內容設定	
必要 XMS	需要同等的 WebSphere MQ JMS 內容
<code>XMSC_CONNECTION_TYPE</code>	XMS 會從 <code>ConnectionFactory</code> 類別名稱和 <code>TRANSPORT (TRAN)</code> 內容來解決這個問題。
<code>XMSC_RTT_HOST_NAME</code>	<code>HOSTNAME (HOST)</code>
<code>XMSC_RTT_PORT</code>	<code>PORT</code>

與 WebSphere 服務 Integration Bus 的連線

表 11: WebSphere 服務 <code>Integration Bus</code> 連線的受管理 <code>ConnectionFactory</code> 物件的內容設定	
XMS 內容	說明
<code>XMSC_CONNECTION_TYPE</code>	應用程式所連接傳訊伺服器的類型。。這是由 <code>ConnectionFactory</code> 類別名稱來決定。
<code>XMSC_WPM_BUS_NAME</code>	對於 <code>ConnectionFactory</code> ，則應用程式所連接的服務整合匯流排的名稱，或者對於目的地，則為目的地所在的服務整合匯流排的名稱。

相關概念

受管理物件儲存庫的受支援類型

「檔案系統」及 LDAP 受管理物件可以用來連接至 WebSphere MQ 及 WebSphere Application Server，而 COS 命名只能用來連接至 WebSphere Application Server。

受管理物件的內容對映

如果要讓應用程式能夠使用 WebSphere MQ JMS 和 WebSphere Application Server `ConnectionFactory` 及目的地物件定義，從這些定義擷取的內容必須對映至可以在 XMS `ConnectionFactory` 和目的地上設定的對應 XMS 內容。

InitialContext 內容

`InitialContext` 建構子的參數包括受管理物件的儲存庫位置，以統一資源指示器 (URI) 形式提供。為了讓應用程式建立與儲存庫的連線，可能需要提供比 URI 中包含的資訊更多的資訊。

XMS 起始環境定義的 URI 格式

受管理物件的儲存庫位置以統一資源指示器 (URI) 提供。URI 的格式視環境定義類型而定。

JNDI 查閱 Web 服務

若要從 XMS 存取 COS 命名目錄，必須在 WebSphere 服務 Integration Bus 伺服器上部署 JNDI 查閱 Web 服務。此 Web 服務會將 Java 資訊從 COS 命名服務轉換成 XMS 應用程式可以讀取的表單。

擷取受管理物件

XMS 會使用建立 InitialContext 物件時所提供的位址，或在 InitialContext 內容中，從儲存庫擷取受管理物件。

WebSphere MQ 佇列管理程式的安全連線

如果要讓 XMS .NET 應用程式能夠建立與 WebSphere MQ 佇列管理程式的安全連線，必須在 ConnectionFactory 物件中定義相關內容。

WebSphere 服務 Integration Bus 傳訊引擎的安全連線

啟用 XMS 應用程式若要建立與 WebSphere 服務 Integration Bus 傳訊引擎的安全連線，必須在 ConnectionFactory 物件中定義相關內容。

相關工作

建立受管理物件

必須使用適當的管理工具來建立 XMS 應用程式建立傳訊伺服器連線所需的 ConnectionFactory 和「目的地」物件定義。

InitialContext 物件

應用程式必須建立起始環境定義，以用來建立受管理物件儲存庫的連線，以擷取必要的受管理物件。

相關參考

受管理目的地物件的必要內容

建立目的地的應用程式必須在受管理的目的地物件上設定應用程式的數個內容。

ConnectionFactory (適用於 .NET 介面)

應用程式使用 Connection Factory 來建立連線。

ConnectionFactory 的內容

ConnectionFactory 物件內容的概觀，以及指向更詳細參照資訊的鏈結。

受管理目的地物件的必要內容

建立目的地的應用程式必須在受管理的目的地物件上設定應用程式的數個內容。

連線類型	內容	說明
WebSphere MQ 佇列管理程式	佇列 (QU) 主題 (頂端)	您要連接的佇列 應用程式用作目的地的主題
與分配管理系統的即時連線	主題 (頂端)	應用程式用作目的地的主題
WebSphere 服務 Integration Bus	topicName queueName	如果您的應用程式正在連接至主題 如果您的應用程式正在連接至佇列

相關概念

受管理物件儲存庫的受支援類型

「檔案系統」及 LDAP 受管理物件可以用來連接至 WebSphere MQ 及 WebSphere Application Server，而 COS 命名只能用來連接至 WebSphere Application Server。

受管理物件的內容對映

如果要讓應用程式能夠使用 WebSphere MQ JMS 和 WebSphere Application Server Connection Factory 及目的地物件定義，從這些定義擷取的內容必須對映至可以在 XMS Connection Factory 和目的地上設定的對應 XMS 內容。

InitialContext 內容

InitialContext 建構子的參數包括受管理物件的儲存庫位置，以統一資源指示器 (URI) 形式提供。為了讓應用程式建立與儲存庫的連線，可能需要提供比 URI 中包含的資訊更多的資訊。

XMS 起始環境定義的 URI 格式

受管理物件的儲存庫位置以統一資源指示器 (URI) 提供。URI 的格式視環境定義類型而定。

JNDI 查閱 Web 服務

若要從 XMS 存取 COS 命名目錄，必須在 WebSphere 服務 Integration Bus 伺服器上部署 JNDI 查閱 Web 服務。此 Web 服務會將 Java 資訊從 COS 命名服務轉換成 XMS 應用程式可以讀取的表單。

擷取受管理物件

XMS 會使用建立 InitialContext 物件時所提供的位址，或在 InitialContext 內容中，從儲存庫擷取受管理物件。

相關工作

建立受管理物件

必須使用適當的管理工具來建立 XMS 應用程式建立傳訊伺服器連線所需的 ConnectionFactory 和「目的地」物件定義。

InitialContext 物件

應用程式必須建立起始環境定義，以用來建立受管理物件儲存庫的連線，以擷取必要的受管理物件。

相關參考

受管理 ConnectionFactory 物件的必要內容

當應用程式建立 Connection Factory 時，必須定義一些內容來建立傳訊伺服器的連線。

IDestination (適用於 .NET 介面)

目的地是應用程式傳送訊息的地方，及（或）應用程式從中接收訊息的來源。

目的地的內容

「目的地」物件的內容概觀，以及指向更詳細參照資訊的鏈結。

建立受管理物件

必須使用適當的管理工具來建立 XMS 應用程式建立傳訊伺服器連線所需的 ConnectionFactory 和「目的地」物件定義。

開始之前

如需 XMS 支援的不同受管理物件儲存庫類型的進一步詳細資料，請參閱 [第 43 頁的『受管理物件儲存庫的受支援類型』](#)。

關於這項作業

如果要建立 WebSphere MQ 的受管理物件，請使用 WebSphere MQ Explorer 或 WebSphere MQ JMS 管理 (JMSAdmin) 工具。

若要為 WebSphere MQ 或 IBM Integration Bus 建立受管理物件，請使用 WebSphere MQ JMS 管理 (JMSAdmin) 工具。

如果要建立「WebSphere 服務 Integration Bus」的受管理物件，請使用 WebSphere Application Server 管理主控台。

下列步驟彙總您在建立受管理物件時所執行的動作。

程序

1. 建立 Connection Factory 並定義必要的內容，以建立從應用程式到所選伺服器的連線。
XMS 建立連線所需的最少內容定義在 [第 45 頁的『受管理 ConnectionFactory 物件的必要內容』](#) 中。
2. 在您應用程式所連接的傳訊伺服器上建立必要的目的地：
 - 若要建立與 WebSphere MQ 佇列管理程式的連線，請建立佇列或主題。
 - 如需與分配管理系統的即時連線，請建立主題。
 - 若要建立與 WebSphere 服務 Integration Bus 的連線，請建立佇列或主題。XMS 建立連線所需的最少內容定義在 [第 47 頁的『受管理目的地物件的必要內容』](#) 中。

相關概念

受管理物件儲存庫的受支援類型

「檔案系統」及 LDAP 受管理物件可以用來連接至 WebSphere MQ 及 WebSphere Application Server，而 COS 命名只能用來連接至 WebSphere Application Server。

受管理物件的內容對映

如果要讓應用程式能夠使用 WebSphere MQ JMS 和 WebSphere Application Server Connection Factory 及目的地物件定義，從這些定義擷取的內容必須對映至可以在 XMS Connection Factory 和目的地上設定的對應 XMS 內容。

InitialContext 內容

InitialContext 建構子的參數包括受管理物件的儲存庫位置，以統一資源指示器 (URI) 形式提供。為了讓應用程式建立與儲存庫的連線，可能需要提供比 URI 中包含的資訊更多的資訊。

XMS 起始環境定義的 URI 格式

受管理物件的儲存庫位置以統一資源指示器 (URI) 提供。URI 的格式視環境定義類型而定。

JNDI 查閱 Web 服務

若要從 XMS 存取 COS 命名目錄，必須在 WebSphere 服務 Integration Bus 伺服器上部署 JNDI 查閱 Web 服務。此 Web 服務會將 Java 資訊從 COS 命名服務轉換成 XMS 應用程式可以讀取的表單。

擷取受管理物件

XMS 會使用建立 InitialContext 物件時所提供的位址，或在 InitialContext 內容中，從儲存庫擷取受管理物件。

受管理物件

使用受管理物件，您可以管理要從中央儲存庫管理的用戶端應用程式所使用的連線設定。應用程式會從中央儲存庫擷取物件定義，並使用它們來建立 ConnectionFactory 及 Destination 物件。使用受管理物件，您可以將應用程式與它們在執行時期使用的資源取消連結。

ConnectionFactories 和 Connection 物件

ConnectionFactory 物件提供應用程式用來建立 Connection 物件的範本。應用程式使用 Connection 物件來建立 Session 物件。

WebSphere 服務整合匯流排的連線

XMS 應用程式可以使用直接 TCP/IP 連線或使用透過 TCP/IP 的 HTTP 來連接至 WebSphere 服務 Integration Bus。

相關工作

InitialContext 物件

應用程式必須建立起始環境定義，以用來建立受管理物件儲存庫的連線，以擷取必要的受管理物件。

相關參考

受管理 ConnectionFactory 物件的必要內容

當應用程式建立 Connection Factory 時，必須定義一些內容來建立傳訊伺服器的連線。

受管理目的地物件的必要內容

建立目的地的應用程式必須在受管理的目的地物件上設定應用程式的數個內容。

IConnectionFactory (適用於 .NET 介面)

應用程式使用 Connection Factory 來建立連線。

ConnectionFactory 的內容

ConnectionFactory 物件內容的概觀，以及指向更詳細參照資訊的鏈結。

IDestination (適用於 .NET 介面)

目的地是應用程式傳送訊息的地方，及（或）應用程式從中接收訊息的來源。

目的地的內容

「目的地」物件的內容概觀，以及指向更詳細參照資訊的鏈結。

InitialContext 物件

應用程式必須建立起始環境定義，以用來建立受管理物件儲存庫的連線，以擷取必要的受管理物件。

關於這項作業

InitialContext 物件會封裝與儲存庫的連線。XMS API 提供方法來執行下列作業：

- 建立 InitialContext 物件
- 在受管理物件儲存庫中查閱受管理物件。

如需建立 InitialContext 物件的進一步詳細資料，請參閱 [.NET](#) 和 [第 167 頁的『InitialContext 的內容』](#) 的 [第 94 頁的『InitialContext』](#)。

相關概念

[受管理物件儲存庫的受支援類型](#)

「檔案系統」及 LDAP 受管理物件可以用來連接至 WebSphere MQ 及 WebSphere Application Server，而 COS 命名只能用來連接至 WebSphere Application Server。

[受管理物件的內容對映](#)

如果要讓應用程式能夠使用 WebSphere MQ JMS 和 WebSphere Application Server Connection Factory 及目的地物件定義，從這些定義擷取的內容必須對映至可以在 XMS Connection Factory 和目的地設定的對應 XMS 內容。

[InitialContext 內容](#)

InitialContext 建構子的參數包括受管理物件的儲存庫位置，以統一資源指示器 (URI) 形式提供。為了讓應用程式建立與儲存庫的連線，可能需要提供比 URI 中包含的資訊更多的資訊。

[XMS 起始環境定義的 URI 格式](#)

受管理物件的儲存庫位置以統一資源指示器 (URI) 提供。URI 的格式視環境定義類型而定。

[JNDI 查閱 Web 服務](#)

若要從 XMS 存取 COS 命名目錄，必須在 WebSphere 服務 Integration Bus 伺服器上部署 JNDI 查閱 Web 服務。此 Web 服務會將 Java 資訊從 COS 命名服務轉換成 XMS 應用程式可以讀取的表單。

[擷取受管理物件](#)

XMS 會使用建立 InitialContext 物件時所提供的位址，或在 InitialContext 內容中，從儲存庫擷取受管理物件。

相關工作

[建立受管理物件](#)

必須使用適當的管理工具來建立 XMS 應用程式建立傳訊伺服器連線所需的 ConnectionFactory 和「目的地」物件定義。

相關參考

[受管理 ConnectionFactory 物件的必要內容](#)

當應用程式建立 Connection Factory 時，必須定義一些內容來建立傳訊伺服器的連線。

[受管理目的地物件的必要內容](#)

建立目的地的應用程式必須在受管理的目的地物件上設定應用程式的數個內容。

[InitialContext \(適用於 .NET 介面\)](#)

應用程式使用 InitialContext 物件，從受管理物件儲存庫擷取的物件定義建立物件。

[InitialContext 的內容](#)

InitialContext 物件內容的概觀，以及更詳細參照資訊的鏈結。

InitialContext 內容

InitialContext 建構子的參數包括受管理物件的儲存庫位置，以統一資源指示器 (URI) 形式提供。為了讓應用程式建立與儲存庫的連線，可能需要提供比 URI 中包含的資訊更多的資訊。

在 XMS 的 JNDI 及 .NET 實作中，會在建構子的環境 Hashtable 中提供其他資訊。

受管理物件儲存庫的位置定義在 [XMSC_IC_URL](#) 內容中。此內容通常在「建立」呼叫上傳遞，但可以修改為在查閱之前連接至不同的命名目錄。對於 FileSystem 或 LDAP 環境定義，此內容定義目錄的位址。對於 COS 命名，這是使用這些內容來連接至 JNDI 目錄的 Web 服務位址。

下列內容會以未修改的方式傳遞至 Web 服務，而 Web 服務會使用這些內容來連接至 JNDI 目錄。

- [XMSC_IC_PROVIDER_URL](#)

- [XMSC_IC_SECURITY_CREDENTIALS](#)
- [XMSC_IC_SECURITY_AUTHENTICATION](#)
- [XMSC_IC_SECURITY_PRINCIPAL](#)
- [XMSC_IC_SECURITY_PROTOCOL](#)

相關概念

受管理物件儲存庫的受支援類型

「檔案系統」及 LDAP 受管理物件可以用來連接至 WebSphere MQ 及 WebSphere Application Server，而 COS 命名只能用來連接至 WebSphere Application Server。

受管理物件的內容對映

如果要讓應用程式能夠使用 WebSphere MQ JMS 和 WebSphere Application Server Connection Factory 及目的地物件定義，從這些定義擷取的內容必須對映至可以在 XMS Connection Factory 和目的地上設定的對應 XMS 內容。

XMS 起始環境定義的 URI 格式

受管理物件的儲存庫位置以統一資源指示器 (URI) 提供。URI 的格式視環境定義類型而定。

JNDI 查閱 Web 服務

若要從 XMS 存取 COS 命名目錄，必須在 WebSphere 服務 Integration Bus 伺服器上部署 JNDI 查閱 Web 服務。此 Web 服務會將 Java 資訊從 COS 命名服務轉換成 XMS 應用程式可以讀取的表單。

擷取受管理物件

XMS 會使用建立 InitialContext 物件時所提供的位址，或在 InitialContext 內容中，從儲存庫擷取受管理物件。

相關工作

建立受管理物件

必須使用適當的管理工具來建立 XMS 應用程式建立傳訊伺服器連線所需的 ConnectionFactory 和「目的地」物件定義。

InitialContext 物件

應用程式必須建立起始環境定義，以用來建立受管理物件儲存庫的連線，以擷取必要的受管理物件。

相關參考

受管理 ConnectionFactory 物件的必要內容

當應用程式建立 Connection Factory 時，必須定義一些內容來建立傳訊伺服器的連線。

受管理目的地物件的必要內容

建立目的地的應用程式必須在受管理的目的地物件上設定應用程式的數個內容。

InitialContext (適用於 .NET 介面)

應用程式使用 InitialContext 物件，從受管理物件儲存庫擷取的物件定義建立物件。

InitialContext 的內容

InitialContext 物件內容的概觀，以及更詳細參照資訊的鏈結。

XMS 起始環境定義的 URI 格式

受管理物件的儲存庫位置以統一資源指示器 (URI) 提供。URI 的格式視環境定義類型而定。

FileSystem 環境定義

對於 FileSystem 環境定義，URL 會提供檔案系統型目錄的位置。URL 的結構如 RFC 1738 (統一資源定址器 (URL)) 所定義: URL 具有字首 file://，且此字首後面的語法是可在執行 XMS 的系統上開啟之檔案的有效定義。

此語法可以是平台專用，並且可以使用 '/' 分隔字元或 '\' 分隔字元。如果您使用 '\'，則需要使用其他 '\' 來跳出每一個分隔字元。這可防止 .NET Framework 嘗試將分隔字元解譯為後續的跳出字元。

這些範例說明此語法:

```
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
```

LDAP 環境定義

對於 LDAP 環境定義，URL 的基本結構如 RFC 2255 所定義：*LDAP URL* 格式，字首不區分大小寫 `ldap://`

下列範例說明精確的語法：

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

此語法如 RFC 中所定義，但不支援任何屬性、範圍、過濾器或延伸。

此語法的範例包括：

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

WSS 環境定義

對於 WSS 環境定義，URL 是 Web 服務端點的形式，字首為 `http://`。

或者，您也可以使用字首 `cosnaming://` 或 `wsvc://`，

這兩個字首會解譯為您使用 WSS 環境定義與透過 `http` 存取的 URL，這可讓起始環境定義類型輕鬆直接從 URL 衍生。

此語法的範例包括下列：

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

相關概念

受管理物件儲存庫的受支援類型

「檔案系統」及 LDAP 受管理物件可以用來連接至 WebSphere MQ 及 WebSphere Application Server，而 COS 命名只能用來連接至 WebSphere Application Server。

受管理物件的內容對映

如果要讓應用程式能夠使用 WebSphere MQ JMS 和 WebSphere Application Server Connection Factory 及目的地物件定義，從這些定義擷取的內容必須對映至可以在 XMS Connection Factory 和目的地設定的對應 XMS 內容。

InitialContext 內容

InitialContext 建構子的參數包括受管理物件的儲存庫位置，以統一資源指示器 (URI) 形式提供。為了讓應用程式建立與儲存庫的連線，可能需要提供比 URI 中包含的資訊更多的資訊。

JNDI 查閱 Web 服務

若要從 XMS 存取 COS 命名目錄，必須在 WebSphere 服務 Integration Bus 伺服器上部署 JNDI 查閱 Web 服務。此 Web 服務會將 Java 資訊從 COS 命名服務轉換成 XMS 應用程式可以讀取的表單。

擷取受管理物件

XMS 會使用建立 InitialContext 物件時所提供的位址，或在 InitialContext 內容中，從儲存庫擷取受管理物件。

相關工作

建立受管理物件

必須使用適當的管理工具來建立 XMS 應用程式建立傳訊伺服器連線所需的 ConnectionFactory 和「目的地」物件定義。

InitialContext 物件

應用程式必須建立起始環境定義，以用來建立受管理物件儲存庫的連線，以擷取必要的受管理物件。

相關參考

受管理 ConnectionFactory 物件的必要內容

當應用程式建立 Connection Factory 時，必須定義一些內容來建立傳訊伺服器的連線。

受管理目的地物件的必要內容

建立目的地的應用程式必須在受管理的目的地物件上設定應用程式的數個內容。

InitialContext (適用於 .NET 介面)

應用程式使用 InitialContext 物件，從從受管理物件儲存庫擷取的物件定義建立物件。

InitialContext 的內容

InitialContext 物件內容的概觀，以及更詳細參照資訊的鏈結。

JNDI 查閱 Web 服務

若要從 XMS 存取 COS 命名目錄，必須在 WebSphere 服務 Integration Bus 伺服器上部署 JNDI 查閱 Web 服務。此 Web 服務會將 Java 資訊從 COS 命名服務轉換成 XMS 應用程式可以讀取的表單。

Web 服務在安裝目錄內的企業保存檔 SIBXJndiLookupEAR.ear 中提供。對於現行版本的 Message Service Client for .NET，可在 <install_dir>\java\lib 目錄中找到 SIBXJndiLookupEAR.ear。這可以使用管理主控台或 wsadmin Scripting 工具，安裝在 WebSphere 服務整合匯流排伺服器內。如需部署 Web 服務應用程式的進一步資訊，請參閱產品說明文件。

如果要在 XMS 應用程式內定義 Web 服務，您只需要將 InitialContext 物件的 XMSC_IC_URL 內容設為 Web 服務端點 URL。例如，如果 Web 服務部署在稱為 MyServer 的伺服器主機上，則 Web 服務端點 URL 的範例如下：

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

設定 XMSC_IC_URL 內容可讓 InitialContext 查閱呼叫在定義的端點上呼叫 Web 服務，進而從 COS 命名服務查閱所需的受管理物件。

.NET 應用程式可以使用 Web 服務。XMS C、/C++ 及 XMS .NET 的伺服器端部署相同。XMS .NET 直接透過 Microsoft .NET 架構呼叫 Web 服務。

相關概念

受管理物件儲存庫的受支援類型

「檔案系統」及 LDAP 受管理物件可以用來連接至 WebSphere MQ 及 WebSphere Application Server，而 COS 命名只能用來連接至 WebSphere Application Server。

受管理物件的內容對映

如果要讓應用程式能夠使用 WebSphere MQ JMS 和 WebSphere Application Server Connection Factory 及目的地物件定義，從這些定義擷取的內容必須對映至可以在 XMS Connection Factory 和目的地上設定的對應 XMS 內容。

InitialContext 內容

InitialContext 建構子的參數包括受管理物件的儲存庫位置，以統一資源指示器 (URI) 形式提供。為了讓應用程式建立與儲存庫的連線，可能需要提供比 URI 中包含的資訊更多的資訊。

XMS 起始環境定義的 URI 格式

受管理物件的儲存庫位置以統一資源指示器 (URI) 提供。URI 的格式視環境定義類型而定。

擷取受管理物件

XMS 會使用建立 InitialContext 物件時所提供的位址，或在 InitialContext 內容中，從儲存庫擷取受管理物件。

設定傳訊伺服器環境

此章節說明如何設定傳訊伺服器環境，以容許 XMS 應用程式連接至伺服器。

使用 XMS 範例應用程式

使用 XMS 隨附的範例應用程式來驗證安裝及傳訊伺服器設定，並協助您建置自己的應用程式。這些範例提供每一個 API 的一般特性概觀。

相關工作

建立受管理物件

必須使用適當的管理工具來建立 XMS 應用程式建立傳訊伺服器連線所需的 ConnectionFactory 和「目的地」物件定義。

InitialContext 物件

應用程式必須建立起始環境定義，以用來建立受管理物件儲存庫的連線，以擷取必要的受管理物件。

使用安裝精靈來安裝 Message Service Client for .NET

安裝使用 InstallShield X/Windows MSI 安裝程式。有兩個設定選項可用，因此您可以選擇完整或自訂安裝。

相關參考

受管理 ConnectionFactory 物件的必要內容

當應用程式建立 Connection Factory 時，必須定義一些內容來建立傳訊伺服器的連線。

受管理目的地物件的必要內容

建立目的地的應用程式必須在受管理的目的地物件上設定應用程式的數個內容。

擷取受管理物件

XMS 會使用建立 InitialContext 物件時所提供的位址，或在 InitialContext 內容中，從儲存庫擷取受管理物件。

要擷取的物件可以具有下列類型的名稱：

- 說明 Destination 物件的簡稱，例如，稱為 SalesOrders 的佇列目的地
- 複合名稱，可以由 SubContexts 組成，以 '/' 區隔，且必須以物件名稱結尾。複合名稱的範例為 "Warehouse/PickLists/DispatchQueue2"，其中 Warehouse 和 Picklists 是命名目錄中的 SubContexts，而 DispatchQueue2 是目的地物件的名稱。

相關概念

受管理物件儲存庫的受支援類型

「檔案系統」及 LDAP 受管理物件可以用來連接至 WebSphere MQ 及 WebSphere Application Server，而 COS 命名只能用來連接至 WebSphere Application Server。

受管理物件的內容對映

如果要讓應用程式能夠使用 WebSphere MQ JMS 和 WebSphere Application Server Connection Factory 及目的地物件定義，從這些定義擷取的內容必須對映至可以在 XMS Connection Factory 和目的地設定的對應 XMS 內容。

InitialContext 內容

InitialContext 建構子的參數包括受管理物件的儲存庫位置，以統一資源指示器 (URI) 形式提供。為了讓應用程式建立與儲存庫的連線，可能需要提供比 URI 中包含的資訊更多的資訊。

XMS 起始環境定義的 URI 格式

受管理物件的儲存庫位置以統一資源指示器 (URI) 提供。URI 的格式視環境定義類型而定。

JNDI 查閱 Web 服務

若要從 XMS 存取 COS 命名目錄，必須在 WebSphere 服務 Integration Bus 伺服器上部署 JNDI 查閱 Web 服務。此 Web 服務會將 Java 資訊從 COS 命名服務轉換成 XMS 應用程式可以讀取的表單。

相關工作

建立受管理物件

必須使用適當的管理工具來建立 XMS 應用程式建立傳訊伺服器連線所需的 ConnectionFactory 和「目的地」物件定義。

InitialContext 物件

應用程式必須建立起始環境定義，以用來建立受管理物件儲存庫的連線，以擷取必要的受管理物件。

相關參考

受管理 ConnectionFactory 物件的必要內容

當應用程式建立 Connection Factory 時，必須定義一些內容來建立傳訊伺服器的連線。

受管理目的地物件的必要內容

建立目的地的應用程式必須在受管理的目的地物件上設定應用程式的數個內容。

[InitialContext \(適用於 .NET 介面\)](#)

應用程式使用 InitialContext 物件，從受管理物件儲存庫擷取的物件定義建立物件。

[InitialContext 的內容](#)

InitialContext 物件內容的概觀，以及更詳細參照資訊的鏈結。

保護 XMS 應用程式的通訊安全

這個章節提供設定安全通訊的相關資訊，可讓 XMS 應用程式透過 Secure Sockets Layer (SSL) 連接至 WebSphere 服務 Integration Bus 傳訊引擎或 WebSphere MQ 佇列管理程式。

章節包含下列區段：

- [第 55 頁的『WebSphere MQ 佇列管理程式的安全連線』](#)
- [第 56 頁的『CipherSuite 及 CipherSpec 名稱對映，用於連接至 WebSphere MQ 佇列管理程式』](#)
- [第 57 頁的『WebSphere 服務 Integration Bus 傳訊引擎的安全連線』](#)
- [第 58 頁的『WebSphere 服務 Integration Bus 連線的 CipherSuite 及 CipherSpec 名稱對映』](#)

WebSphere MQ 佇列管理程式的安全連線

如果要讓 XMS .NET 應用程式能夠建立與 WebSphere MQ 佇列管理程式的安全連線，必須在 ConnectionFactory 物件中定義相關內容。

加密協議中使用的通訊協定可以是 Secure Sockets Layer (SSL) 或傳輸層安全 (TLS)，視您在 ConnectionFactory 物件中指定的 CipherSuite 而定。

如果您使用 WebSphere MQ 7.0.0.1 版以及更新版本的用戶端程式庫，並連接至 WebSphere MQ 第 7 版佇列管理程式，則可以在 XMS 應用程式中建立與相同佇列管理程式的多個連線。不過，不允許連線至不同的佇列管理程式。如果您嘗試，則會收到 MQRC_SSL_ALREADY_INITIALIZED 錯誤。

如果您使用 WebSphere MQ 第 6 版以及更新版本的用戶端程式庫，則只有在您先關閉任何先前的 SSL 連線時，才能建立 SSL 連線。不允許從相同處理程序到相同或不同佇列管理程式的多個並行 SSL 連線。如果您嘗試多個要求，則會收到警告 MQRC_SSL_ALREADY_INITIALIZED，這可能表示已忽略針對 SSL 連線所要求的部分參數。

下表顯示透過 SSL 連接至 IBM WebSphere MQ 管理程式之連線的 ConnectionFactory 內容，以及簡要說明：

內容名稱	說明
XMSC_WMQ_SSL_CERT_STORES	保留給佇列管理程式建立 SSL 連線時所使用憑證撤銷清冊 (CRL) 的伺服器位置。
XMSC_WMQ_SSL_CIPHER_SPEC	建立佇列管理程式的安全連線時所使用的 CipherSpec 名稱。
XMSC_WMQ_SSL_CIPHER_SUITE	要在與佇列管理程式的 SSL 或 TLS 連線所使用的 CipherSuite 名稱。在協議安全連線時使用的通訊協定取決於指定的 CipherSuite。
XMSC_WMQ_SSL_CRYPT_HW	用戶端系統所連接的加密硬體的配置詳細資料。
XMSC_WMQ_SSL_FIPS_REQUIRED	此內容的值決定應用程式是否可以使用不符合 FIPS 標準的密碼組合。如果此內容設為 true，則只將 FIPS 演算法用於用戶端/伺服器連線。
XMSC_WMQ_SSL_KEY_REPOSITORY	在其中儲存金鑰和憑證的金鑰資料庫檔的位置。
XMSC_WMQ_SSL_KEY_RESETCOUNT	KeyResetCount 代表在重新協議秘密金鑰之前 SSL 交談中傳送和收到的未加密位元組總數。
XMSC_WMQ_SSL_PEER_NAME	建立佇列管理程式的 SSL 連線時所使用的對等節點名稱。

相關參考

[IConnectionFactory](#) (適用於 .NET 介面)
應用程式使用 [ConnectionFactory](#) 來建立連線。

[ConnectionFactory](#) 的內容
[ConnectionFactory](#) 物件內容的概觀，以及指向更詳細參照資訊的鏈結。

受管理 [ConnectionFactory](#) 物件的必要內容
當應用程式建立 [ConnectionFactory](#) 時，必須定義一些內容來建立傳訊伺服器的連線。

CipherSuite 及 CipherSpec 名稱對映，用於連接至 WebSphere MQ 佇列管理程式

`InitialContext` 會在 `JMSAdmin Connection Factory` 內容 `SSLCIPHERSUITE` 與 `XMS` 接近對等的 `XMSC_WMQ_SSL_CIPHER_SPEC` 之間進行轉換。如果您指定 `XMSC_WMQ_SSL_CIPHER_SUITE` 的值，但省略 `XMSC_WMQ_SSL_CIPHER_SPEC` 的值，則需要類似的轉換。

第 56 頁的表 14 列出可用的 `CipherSpecs` 及其 `JSSE CipherSuite` 對等項目。

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

註: `CipherSuite` 名稱 `SSL_RSA_WITH_3DES_EDE_CBC_SHA` 或 `SSL_RSA_WITH_DES_CBC_SHA` 的一對一對映必須考量 `XMSC_WMQ_SSL_FIPSREQUIRED` 內容的設定，並套用探索性。

> 如果您針對內容 `XMSC_WMQ_SSL_CIPHER_SUITE` 指定 `SSL_RSA_WITH_3DES_EDE_CBC_SHA` 或 `SSL_RSA_WITH_DES_CBC_SHA`，且沒有 `XMSC_WMQ_SSL_CIPHER_SPEC` 的值，則會根據下列表格選擇 `XMSC_WMQ_SSL_CIPHER_SPEC` 的值。

下表顯示當您針對 `XMSC_WMQ_SSL_CIPHER_SUITE` 內容指定 `SSL_RSA_WITH_3DES_EDE_CBC_SHA` 時用於 `XMSC_WMQ_SSL_CIPHER_SPEC` 的值:

輸入: <code>XMSC_WMQ_SSL_FIPSREQUIRED</code> 值	輸出: 已選擇 <code>XMSC_WMQ_SSL_CIPHER_SPEC</code>
false (即 <code>MQSSL_FIPS_NO</code>)	<code>TRIPLE_DES_SHA_US</code>

表 15: 當您針對 <code>XMSC_WMQ_SSL_CIPHER_SUITE</code> 內容指定 <code>SSL_RSA_WITH_3DES_EDE_CBC_SHA</code> 時, <code>XMSC_WMQ_SSL_CIPHER_SPEC</code> 使用的值 (繼續)	
輸入 : <code>XMSC_WMQ_SSL_FIPSREQUIRED</code> 值	輸出: 已選擇 <code>XMSC_WMQ_SSL_CIPHER_SPEC</code>
true (即 <code>MQSSL_FIPS_YES</code>)	<code>TLS_RSA_WITH_3DES_EDE_CBC_SHA</code>

註:

- `TLS_RSA_WITH_3DES_EDE_CBC_SHA` 已淘汰。不過, 在連線因錯誤 AMQ9288 而終止之前, 它仍可用來傳送最多 32 GB 的資料。若要避免此錯誤, 您需要避免使用三重 DES 演算法, 或在使用此 CipherSpec 時啟用秘密金鑰重設。

下表顯示當您針對 `XMSC_WMQ_SSL_CIPHER_SUITE` 內容指定 `SSL_RSA_WITH_DES_CBC_SHA` 時, `XMSC_WMQ_SSL_CIPHER_SPEC` 使用的值:

表 16: 當您針對 <code>XMSC_WMQ_SSL_CIPHER_SUITE</code> 內容指定 <code>SSL_RSA_WITH_DES_CBC_SHA</code> 時, <code>XMSC_WMQ_SSL_CIPHER_SPEC</code> 使用的值	
輸入 : <code>XMSC_WMQ_SSL_FIPSREQUIRED</code> 值	輸出: 已選擇 <code>XMSC_WMQ_SSL_CIPHER_SPEC</code>
false (即 <code>MQSSL_FIPS_NO</code>)	<code>DES_SHA_EXPORT</code>
true (即 <code>MQSSL_FIPS_YES</code>)	<code>TLS_RSA_WITH_DES_CBC_SHA</code>

WebSphere 服務 Integration Bus 傳訊引擎的安全連線

啟用 XMS 應用程式若要建立與 WebSphere 服務 Integration Bus 傳訊引擎的安全連線, 必須在 `ConnectionFactory` 物件中定義相關內容。

XMS 為 WebSphere 服務 Integration Bus 的連線提供 SSL 及 HTTP 支援。SSL 和 HTTPS 提供安全連線, 以進行鑑別和機密性。

類似於 WebSphere 安全, XMS 安全是根據 JSSE 安全標準及命名慣例來配置, 其中包括使用 `CipherSuites` 來指定協議安全連線時所使用的演算法。加密協議中使用的通訊協定可以是 SSL 或 TLS, 視您在 `ConnectionFactory` 物件中指定的 `CipherSuite` 而定。

第 57 頁的表 17 列出必須在 `ConnectionFactory` 物件中定義的內容。

表 17: <code>ConnectionFactory</code> 的內容, 以建立與 WebSphere 服務整合匯流排傳訊引擎的安全連線	
內容名稱	說明
<code>XMSC_WPM_SSL_CIPHER_SUITE</code>	要在與 WebSphere 服務 Integration Bus 傳訊引擎的 SSL 或 TLS 連線上使用的 <code>CipherSuite</code> 名稱。在協議安全連線時使用的通訊協定取決於指定的 <code>CipherSuite</code> 。
<code>XMSC_WPM_SSL_KEYRING_LABEL</code>	向伺服器鑑別時使用的憑證。

以下是 WebSphere 整合傳訊引擎之安全連線的 `ConnectionFactory` 內容範例:

```
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);
```

其中 `chain_name` 應該設為 `BootstrapTunneledSecureMessaging` 或 `BootstrapSecureMessaging`, 而 `port_number` 是引導伺服器用來接聽送入要求的埠號。

以下是插入範例值之 WebSphere 整合傳訊引擎安全連線的 `ConnectionFactory` 內容範例:

```
/* 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");

```

相關參考

[IConnectionFactory \(適用於 .NET 介面\)](#)

應用程式使用 Connection Factory 來建立連線。

[ConnectionFactory 的內容](#)

ConnectionFactory 物件內容的概觀，以及指向更詳細參照資訊的鏈結。

[受管理 ConnectionFactory 物件的必要內容](#)

當應用程式建立 Connection Factory 時，必須定義一些內容來建立傳訊伺服器的連線。

WebSphere 服務 Integration Bus 連線的 CipherSuite 及 CipherSpec 名稱對映

因為 GSKit 使用 CipherSpecs 而非 CipherSuites，所以 XMSC_WPM_SSL_CIPHER_SUITE 內容中指定的 JSSE 樣式 CipherSuite 名稱必須對映至 GSKit 樣式 CipherSpec 名稱。

第 58 頁的表 18 列出每一個已辨識 CipherSuite 的對等 CipherSpec。

CipherSuite	CipherSpec 對等項目
SSL_RSA_WITH_NULL_MD5	NULL_MD5
SSL_RSA_EXPORT_WITH_RC4_40_MD5	RC4_MD5_EXPORT
SSL_RSA_WITH_RC4_128_MD5	RC4_MD5_US
SSL_RSA_WITH_NULL_SHA	NULL_SHA
SSL_RSA_EXPORT1024_WITH_RC4_56_SHA	RC4_56_SHA_EXPORT1024
SSL_RSA_WITH_RC4_128_SHA	RC4_SHA_US
SSL_RSA_WITH_DES_CBC_SHA	DES_SHA_EXPORT
SSL_RSA_EXPORT1024_WITH_DES_CBC_SHA	DES_SHA_EXPORT1024
SSL_RSA_FIPS_WITH_DES_CBC_SHA	FIPS_WITH_DES_CBC_SHA
SSL_RSA_WITH_3DES_EDE_CBC_SHA	TRIPLE_DES_SHA_US
SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA	FIPS_WITH_3DES_EDE_CBC_SHA
TLS_RSA_WITH_DES_CBC_SHA	TLS_RSA_WITH_DES_CBC_SHA
TLS_RSA_WITH_3DES_EDE_CBC_SHA	TLS_RSA_WITH_3DES_EDE_CBC_SHA
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

註:

- TLS_RSA_WITH_3DES_EDE_CBC_SHA 已淘汰。不過，在連線因錯誤 AMQ9288 而終止之前，它仍可用來傳送最多 32 GB 的資料。若要避免此錯誤，您需要避免使用三重 DES 演算法，或在使用此 CipherSpec 時啟用秘密金鑰重設。

XMS 訊息

此章節說明 XMS 訊息的結構及內容，並說明應用程式如何處理 XMS 訊息。

此章節包含下列區段:

- [第 59 頁的『XMS 訊息的組件』](#)

- [第 59 頁的『XMS 訊息中的標頭欄位』](#)
- [第 60 頁的『XMS 訊息的內容』](#)
- [第 63 頁的『XMS 訊息的內文』](#)
- [第 67 頁的『訊息選取器』](#)
- [第 68 頁的『將 XMS 訊息對映至 WebSphere MQ 訊息』](#)

相關參考

[IMessage](#) (適用於 .NET 介面)

訊息物件代表應用程式傳送或接收的訊息。IMessage 是訊息類別的超類別，例如 IMapMessage。

XMS 訊息的組件

XMS 訊息包含一個標頭、一組內容及一個內文。

標頭

訊息的標頭包含欄位，而所有訊息都包含相同的標頭欄位集。XMS 及應用程式使用標頭欄位的值來識別及遞送訊息。如需標頭欄位的相關資訊，請參閱 [第 59 頁的『XMS 訊息中的標頭欄位』](#)。

內容集

訊息的內容指定訊息的其他相關資訊。雖然所有訊息都有一組相同的標頭欄位，但每一則訊息可以有一組不同的內容。如需相關資訊，請參閱 [第 60 頁的『XMS 訊息的內容』](#)。

內文

訊息內文包含應用程式資料。如需相關資訊，請參閱 [第 63 頁的『XMS 訊息的內文』](#)。

應用程式可以選取要接收的訊息。使用訊息選取元來指定選取準則。準則可以根據特定標頭欄位的值，以及訊息任何內容的值。如需訊息選取器的相關資訊，請參閱 [第 67 頁的『訊息選取器』](#)。

相關參考

[XMS 訊息中的標頭欄位](#)

為了容許 XMS 應用程式與 WebSphere JMS 應用程式交換訊息，XMS 訊息的標頭包含 JMS 訊息標頭欄位。

[XMS 訊息的內容](#)

XMS 支援三種類型的訊息內容：JMS 定義的內容、IBM 定義的內容，以及應用程式定義的內容。

[XMS 訊息的內文](#)

訊息內文包含應用程式資料。不過，訊息可以沒有內文，且只包含標頭欄位和內容。

[訊息選取器](#)

XMS 應用程式使用訊息選取器來選取它要接收的訊息。

[將 XMS 訊息對映至 WebSphere MQ 訊息](#)

XMS 訊息的 JMS 標頭欄位和內容會對映至 WebSphere MQ 訊息標頭結構中的欄位。

XMS 訊息中的標頭欄位

為了容許 XMS 應用程式與 WebSphere JMS 應用程式交換訊息，XMS 訊息的標頭包含 JMS 訊息標頭欄位。

這些標頭欄位的名稱以字首 JMS 開始。如需 JMS 訊息標頭欄位的說明，請參閱 [Java 訊息服務規格 1.1](#)。

XMS 會將 JMS 訊息標頭欄位實作為訊息物件的屬性。每一個標頭欄位都有自己的方法來設定及取得其值。如需這些方法的說明，請參閱 [第 106 頁的『IMessage』](#)。標頭欄位一律可讀取及寫入。

[第 59 頁的表 19](#) 列出 JMS 訊息標頭欄位，並指出如何為傳輸的訊息設定每一個欄位的值。當應用程式傳送訊息時，或在 JMSRedelivered 的情況下，當應用程式接收訊息時，XMS 會自動設定部分欄位。

表 19: JMS 訊息標頭欄位	
JMS 訊息標頭欄位的名稱	如何設定所傳輸訊息的值 (格式為 <i>method [class]</i>)
JMSCorrelationID	設定 JMSCorrelationID [訊息]
JMSDeliveryMode	傳送 [MessageProducer]
JMSDestination	傳送 [MessageProducer]

表 19: JMS 訊息標頭欄位 (繼續)

JMS 訊息標頭欄位的名稱	如何設定所傳輸訊息的值 (格式為 <i>method [class]</i>)
JMSExpiration	傳送 [MessageProducer]
JMSMessageID	傳送 [MessageProducer]
JMSPriority	傳送 [MessageProducer]
JMSRedelivered	接收 [MessageConsumer]
JMSReplyTo	設定 JMSReplyTo [訊息]
JMSTimestamp	傳送 [MessageProducer]
JMSType	設定 JMSType [訊息]

相關參考

XMS 訊息的組件

XMS 訊息包含一個標頭、一組內容及一個內文。

XMS 訊息的內容

XMS 支援三種類型的訊息內容 :JMS 定義的內容、IBM 定義的內容，以及應用程式定義的內容。

XMS 訊息的內文

訊息內文包含應用程式資料。不過，訊息可以沒有內文，且只包含標頭欄位和內容。

訊息選取器

XMS 應用程式使用訊息選取器來選取它要接收的訊息。

將 XMS 訊息對映至 WebSphere MQ 訊息

XMS 訊息的 JMS 標頭欄位和內容會對映至 WebSphere MQ 訊息標頭結構中的欄位。

XMS 訊息的內容

XMS 支援三種類型的訊息內容 :JMS 定義的內容、IBM 定義的內容，以及應用程式定義的內容。

XMS 應用程式可以與 WebSphere JMS 應用程式交換訊息，因為 XMS 支援「訊息」物件的下列預先定義內容：

- WebSphere JMS 支援的相同 JMS 定義內容。這些內容的名稱以字首 JMSX 開頭。
- WebSphere JMS 支援的相同 IBM 定義內容。這些內容的名稱以字首 JMS_IBM_ 開頭。

每一個預先定義的內容都有兩個名稱：

- JMS 定義內容的 JMS 名稱，或 IBM 定義內容的 WebSphere JMS 名稱。

這是在 JMS 或 WebSphere JMS 中用來識別內容的名稱，它也是與具有此內容的訊息一起傳輸的名稱。XMS 應用程式使用此名稱來識別訊息選取元表示式中的內容。

- XMS 名稱，用於在訊息選取器表示式以外的所有狀況中識別內容。每一個 XMS 名稱都定義為 IBM.XMS.XMSC 類別中的已命名常數。具名常數的值是對應的 JMS 或 WebSphere JMS 名稱。

除了預先定義的內容之外，XMS 應用程式還可以建立及使用它自己的訊息內容集。這些內容稱為應用程式定義的內容。

在應用程式建立訊息之後，訊息的內容是可讀取和可寫入的。在應用程式傳送訊息之後，內容仍可讀取及寫入。當應用程式接收訊息時，訊息的內容是唯讀的。當訊息的內容是唯讀時，如果應用程式呼叫 Message 類別的 Clear Properties 方法，則內容會變成可讀取及可寫入。此方法也會清除內容。

在清除訊息內容之後轉遞收到的訊息時，其行為將與轉遞標準 WMQ XMS for .NET BytesMessage 的行為一致，並清除訊息內容。

不過，不建議這樣做，因為下列內容將會遺失：

- JMS_IBM_Encoding 內容值，暗示無法對訊息資料進行有意義的解碼。
- JMS_IBM_Format 內容值，暗示 (MQMD 或新的 MQRFH2) 訊息標頭與現有標頭之間的標頭鏈會中斷。

若要判定訊息所有內容的值，應用程式可以呼叫「訊息」類別的「取得內容」方法。此方法會建立一個反覆運算子來封裝「內容」物件的清單，其中每一個「內容」物件代表訊息的內容。然後，應用程式可以使用 Iterator 類別的方法依序擷取每一個 Property 物件，並且可以使用 Property 類別的方法來擷取每一個內容的名稱、資料類型及值。

相關參考

[XMS 訊息的組件](#)

XMS 訊息包含一個標頭、一組內容及一個內文。

[XMS 訊息中的標頭欄位](#)

為了容許 XMS 應用程式與 WebSphere JMS 應用程式交換訊息，XMS 訊息的標頭包含 JMS 訊息標頭欄位。

[XMS 訊息的內文](#)

訊息內文包含應用程式資料。不過，訊息可以沒有內文，且只包含標頭欄位和內容。

[訊息選取器](#)

XMS 應用程式使用訊息選取器來選取它要接收的訊息。

[將 XMS 訊息對映至 WebSphere MQ 訊息](#)

XMS 訊息的 JMS 標頭欄位和內容會對映至 WebSphere MQ 訊息標頭結構中的欄位。

訊息的 JMS 定義內容

XMS 和 WebSphere JMS 都支援訊息的數個 JMS 定義內容。

第 61 頁的表 20 列出 XMS 和 WebSphere JMS 支援之訊息的 JMS 定義內容。如需 JMS 定義內容的說明，請參閱 *Java 訊息服務規格 1.1*。JMS 定義的內容對分配管理系統的即時連線無效。

此表格指定每一個內容的資料類型，並指出如何設定所傳輸訊息的內容值。當應用程式傳送訊息時，或在 JMSXDeliveryCount 的情況下，當應用程式接收訊息時，XMS 會自動設定部分內容。

JMS 定義內容的 XMS 名稱	JMS 名稱	資料類型	如何設定所傳輸訊息的值 (格式為 <i>method [class]</i>)
JMSX_APPID	JMSXAppID	System.String	傳送 [MessageProducer]
JMSX_DELIVERY_COUNT	JMSXDeliveryCount	System.Int32	接收 [MessageConsumer]
JMSX_GROUPID	JMSXGroupID	System.String	設定字串內容 [PropertyContext]
JMSX_GROUPSEQ	JMSXGroupSeq	System.Int32	設定整數內容 [PropertyContext]
JMSX_USERID	JMSXUserID	System.String	傳送 [MessageProducer]

IBM 定義的訊息內容

XMS 和 WebSphere JMS 支援訊息的數個 IBM 定義內容。

第 61 頁的表 21 列出 XMS 和 WebSphere JMS 所支援之訊息的 IBM 定義內容。如需 IBM 定義內容的相關資訊，請參閱 *IBM WebSphere MQ 使用 Java 或 WebSphere Application Server 產品說明文件*。

此表格指定每一個內容的資料類型，並指出如何設定所傳輸訊息的內容值。當應用程式傳送訊息時，XMS 會自動設定部分內容。

IBM 定義內容的 XMS 名稱	WebSphere JMS 名稱	資料類型	如何設定所傳輸訊息的值 (格式為 <i>method [class]</i>)
JMS_IBM_CHARACTER_SET	JMS_IBM_Character_Set	System.Int32	設定整數內容 [PropertyContext]
JMS_IBM_Encoding	JMS_IBM_Encoding	System.Int32	設定整數內容 [PropertyContext]

表 21: IBM 定義的訊息內容 (繼續)			
IBM 定義內容的 XMS 名稱	WebSphere JMS 名稱	資料類型	如何設定所傳輸訊息的值 (格式為 <i>method [class]</i>)
JMS_IBM_EXCEPTIONMESSAGE	JMS_IBM_ExceptionMessage	System.String	接收 [MessageConsumer]
JMS_IBM_EXCEPTIONREASON	JMS_IBM_ExceptionReason	System.Int32	接收 [MessageConsumer]
JMS_IBM_EXCEPTIONTIMESTAMP	JMS_IBM_ExceptionTimestamp	System.Int64	接收 [MessageConsumer]
JMS_IBM_EXCEPTIONPROBLEM 目的地	JMS_IBM_ExceptionProblem 目的地	System.String	接收 [MessageConsumer]
JMS_IBM_FEEDBACK	JMS_IBM_Feedback	System.Int32	設定整數內容 [PropertyContext]
JMS_IBM_FORMAT	JMS_IBM_Format	System.String	設定字串內容 [PropertyContext]
JMS_IBM_LAST_MSG_IN_GROUP	JMS_IBM_Last_Msg_In_Group	System.Boolean	設定整數內容 [PropertyContext]
JMS_IBM_MSGTYPE	JMS_IBM_MsgType	System.Int32	設定整數內容 [PropertyContext]
JMS_IBM_PUTAPPLTYPE	JMS_IBM_PutAppl 類型	System.Int32	傳送 [MessageProducer]
JMS_IBM_PUTDATE	JMS_IBM_PutDate	System.String	傳送 [MessageProducer]
JMS_IBM_PUTTIME	JMS_IBM_PutTime	System.String	傳送 [MessageProducer]
JMS_IBM_REPORT_COA	JMS_IBM_Report_COA	System.Int32	設定整數內容 [PropertyContext]
JMS_IBM_REPORT_COD	JMS_IBM_Report_COD	System.Int32	設定整數內容 [PropertyContext]
JMS_IBM_REPORT_DISCARD_MSG	JMS_IBM_Report_Discard_Msg	System.Int32	設定整數內容 [PropertyContext]
JMS_IBM_REPORT_EXCEPTION	JMS_IBM_Report_Exception	System.Int32	設定整數內容 [PropertyContext]
JMS_IBM_REPORT_EXPIRATION	JMS_IBM_Report_Expiration	System.Int32	設定整數內容 [PropertyContext]
JMS_IBM_REPORT_NAN	JMS_IBM_Report_NAN	System.Int32	設定整數內容 [PropertyContext]
JMS_IBM_REPORT_PAN	JMS_IBM_Report_PAN	System.Int32	設定整數內容 [PropertyContext]
JMS_IBM_REPORT_PASS_CORREL_ID	JMS_IBM_Report_Pass_Correl_ID	System.Int32	設定整數內容 [PropertyContext]
JMS_IBM_REPORT_PASS_MSG_ID	JMS_IBM_Report_Pass_Msg_ID	System.Int32	設定整數內容 [PropertyContext]

表 21: IBM 定義的訊息內容 (繼續)

IBM 定義內容的 XMS 名稱	WebSphere JMS 名稱	資料類型	如何設定所傳輸訊息的值 (格式為 <i>method [class]</i>)
JMS_IBM_SYSTEM_MESS AGEID	JMS_IBM_System_Messa geID	System.String	傳送 [MessageProducer]

訊息的應用程式定義內容

XMS 應用程式可以建立及使用自己的訊息內容集。當應用程式傳送訊息時，這些內容也會隨訊息一起傳輸。然後，接收端應用程式 (使用訊息選取器) 可以根據這些內容的值來選取它想要接收的訊息。

若要容許 WebSphere JMS 應用程式選取及處理 XMS 應用程式所傳送的訊息，應用程式定義內容的名稱必須符合在訊息選取元表示式中形成 ID 的規則，如 *WebSphere MQ* 使用 *Java* 中所記載。應用程式定義內容的值必須具有下列其中一種資料類型: *System.Boolean*、*System.SByte*、*System.Int16*、*System.Int32*、*System.Int64*、*System.Float*、*System.Double* 或 *System.String*。

XMS 訊息的內文

訊息內文包含應用程式資料。不過，訊息可以沒有內文，且只包含標頭欄位和內容。

XMS 支援五種類型的訊息內文：

位元組

主體包含位元組串流。具有此類型內文的訊息稱為位元組訊息。*IBytesMessage* 介面包含用來處理位元組訊息內文的方法。如需相關資訊，請參閱第 64 頁的『位元組訊息』。

對映

內文包含一組名稱/值配對，其中每一個值都有相關聯的資料類型。具有此類型內文的訊息稱為對映訊息。*IMapMessage* 介面包含用來處理對映訊息內文的方法。如需相關資訊，請參閱第 65 頁的『對映訊息』。

物件

主體包含序列化 *Java* 或 *.NET* 物件。具有此類型主體的訊息稱為物件訊息。*IObjectMessage* 介面包含用來處理物件訊息內文的方法。如需相關資訊，請參閱第 66 頁的『物件訊息』。

串流

內文包含值串流，其中每一個值都有相關聯的資料類型。具有此類型內文的訊息稱為串流訊息。*IStreamMessage* 介面包含用來處理串流訊息內文的方法。如需相關資訊，請參閱第 66 頁的『串流訊息』。

Text

內文包含字串。具有此類型內文的訊息稱為文字訊息。*ITextMessage* 介面包含處理文字訊息內文的方法。如需相關資訊，請參閱第 67 頁的『文字訊息』。

IMessage 介面是所有訊息物件的母項，可在傳訊功能中用來代表任何 XMS 訊息類型。

如需每一種資料類型的大小及最大值和最小值的相關資訊，請參閱第 34 頁的表 5。

相關參考

XMS 訊息的組件

XMS 訊息包含一個標頭、一組內容及一個內文。

XMS 訊息中的標頭欄位

為了容許 XMS 應用程式與 WebSphere JMS 應用程式交換訊息，XMS 訊息的標頭包含 JMS 訊息標頭欄位。

XMS 訊息的內容

XMS 支援三種類型的訊息內容：JMS 定義的內容、IBM 定義的內容，以及應用程式定義的內容。

訊息選取器

XMS 應用程式使用訊息選取器來選取它要接收的訊息。

將 XMS 訊息對映至 WebSphere MQ 訊息

XMS 訊息的 JMS 標頭欄位和內容會對映至 WebSphere MQ 訊息標頭結構中的欄位。

應用程式資料元素的資料類型

為了確保 XMS 應用程式可以與 WebSphere MQ for JMS 類別 應用程式交換訊息，這兩個應用程式必須能夠以相同方式解譯訊息內文中的應用程式資料。

因此，XMS 應用程式在訊息內文中寫入的應用程式資料的每一個元素都必須具有第 64 頁的表 22 中列出的其中一個資料類型。對於每一種資料類型，表格會顯示相容的 Java 資料類型。XMS 提供僅使用這些資料類型寫入應用程式資料元素的方法。

XMS 資料類型	代表	相容 Java 資料類型
System.Boolean	布林值 true 或 false	布林值
System.Char16	雙位元組字元	char
System.SByte	帶正負號的 8 位元整數	位元組
System.Int16	帶正負號的 16 位元整數	短
System.Int32	帶正負號的 32 位元整數	整數
System.Int64	帶正負號的 64 位元整數	長整數
System.Float	帶正負號的浮點數字	浮點數
System.Double	帶正負號的倍精準度浮點數字	倍精準數
System.String	字串	字串

如需每一種資料類型的大小、最大值及最小值的相關資訊，請參閱第 34 頁的『XMS 初始類型』。

相關概念

物件的屬性和內容

XMS 物件可以具有以不同方式實作的屬性及內容，這些屬性及內容是物件的性質。

XMS 初始類型

XMS 提供八個 Java 初始類型 (byte、short、int、long、float、double、char 及 boolean) 的對等項目。這容許在 XMS 與 JMS 之間交換訊息，而不會失去或毀損資料。

將內容值從一個資料類型隱含轉換為另一個資料類型

當應用程式取得內容的值時，XMS 可以將該值轉換為另一個資料類型。許多規則會控管支援哪些轉換，以及 XMS 如何執行轉換。

相關參考

位元組訊息

位元組訊息的主體包含位元組串流。主體只包含實際資料，傳送及接收應用程式負責解譯此資料。

對映訊息

對映訊息的內文包含一組名稱/值配對，其中每一個值都有相關聯的資料類型。

物件訊息

物件訊息的主體包含序列化 Java 或 .NET 物件。

串流訊息

串流訊息的內文包含值串流，其中每一個值都有相關聯的資料類型。

文字訊息

文字訊息的內文包含字串。

位元組訊息

位元組訊息的主體包含位元組串流。主體只包含實際資料，傳送及接收應用程式負責解譯此資料。

如果 XMS 應用程式需要與未使用 XMS 或 JMS 應用程式設計介面的應用程式交換訊息，則位元組訊息很有用。

在應用程式建立位元組訊息之後，訊息內文是唯寫的。應用程式透過針對 .NET 呼叫 `IBytesMessage` 介面的適當寫入方法，將應用程式資料組合到主體中。每次應用程式將值寫入位元組訊息串流時，都會在應用程式所寫入的前一個值之後立即組合該值。XMS 會維護內部游標，以記住組合的最後一個位元組的位置。

當應用程式傳送訊息時，訊息內文會變成唯讀。在此模式中，應用程式可以反覆地傳送訊息。

當應用程式接收位元組訊息時，訊息內文是唯讀的。應用程式可以使用 `IBytesMessage` 介面的適當 `read` 方法來讀取位元組訊息串流的內容。應用程式會依序讀取位元組，且 XMS 會維護內部游標，以記住所讀取最後一個位元組的位置。

當位元組訊息的內文可寫入時，如果應用程式呼叫 `IBytesMessage` 介面的 `Reset` 方法，則內文會變成唯讀。此方法也會將游標重新定位在位元組訊息串流的開頭。

當位元組訊息的內文是唯讀時，如果應用程式呼叫 .NET 之 `IMessage` 介面的 `Clear Body` 方法，則內文會變成可寫入。此方法也會清除主體。

相關參考

應用程式資料元素的資料類型

為了確保 XMS 應用程式可以與 WebSphere MQ for JMS 類別 應用程式交換訊息，這兩個應用程式必須能夠以相同方式解譯訊息內文中的應用程式資料。

對映訊息

對映訊息的內文包含一組名稱/值配對，其中每一個值都有相關聯的資料類型。

物件訊息

物件訊息的主體包含序列化 Java 或 .NET 物件。

串流訊息

串流訊息的內文包含值串流，其中每一個值都有相關聯的資料類型。

文字訊息

文字訊息的內文包含字串。

`IBytesMessage` (適用於 .NET 介面)

位元組訊息是其內文包含位元組串流的訊息。

對映訊息

對映訊息的內文包含一組名稱/值配對，其中每一個值都有相關聯的資料類型。

在每一個名稱/值配對中，名稱是識別值的字串，值是應用程式資料的元素，具有第 64 頁的表 22 中列出的其中一個 XMS 資料類型。未定義名稱/值配對的順序。 `MapMessage` 類別包含設定及取得名稱/值配對的方法。

應用程式可以透過指定名稱來隨機存取名稱/值配對。

.NET 應用程式可以使用 `MapNames` 內容來取得對映訊息內文中的名稱列舉。

當應用程式取得名稱/值配對的值時，XMS 可以將該值轉換為另一個資料類型。例如，若要從對映訊息內文中取得整數，應用程式可以呼叫 `MapMessage` 類別的 `GetString` 方法，它會以字串形式傳回整數。支援的轉換與 XMS 將內容值從一種資料類型轉換為另一種資料類型時支援的轉換相同。如需受支援轉換的相關資訊，請參閱第 34 頁的『將內容值從一個資料類型隱含轉換為另一個資料類型』。

在應用程式建立對映訊息之後，訊息的內文是可讀取及可寫入的。在應用程式傳送訊息之後，內文仍可讀取及寫入。當應用程式接收對映訊息時，訊息內文是唯讀的。當對映訊息的主體是唯讀時，如果應用程式呼叫 `Message` 類別的 `Clear Body` 方法，則主體會變成可讀取及可寫入。此方法也會清除主體。

相關概念

將內容值從一個資料類型隱含轉換為另一個資料類型

當應用程式取得內容的值時，XMS 可以將該值轉換為另一個資料類型。許多規則會控管支援哪些轉換，以及 XMS 如何執行轉換。

相關參考

應用程式資料元素的資料類型

為了確保 XMS 應用程式可以與 WebSphere MQ for JMS 類別 應用程式交換訊息，這兩個應用程式必須能夠以相同方式解譯訊息內文中的應用程式資料。

位元組訊息

位元組訊息的主體包含位元組串流。主體只包含實際資料，傳送及接收應用程式負責解譯此資料。

物件訊息

物件訊息的主體包含序列化 Java 或 .NET 物件。

串流訊息

串流訊息的內文包含值串流，其中每一個值都有相關聯的資料類型。

文字訊息

文字訊息的內文包含字串。

IMapMessage (適用於 .NET 介面)

對映訊息是其內文包含一組名稱/值配對的訊息，其中每一個值都具有相關聯的資料類型。

物件訊息

物件訊息的主體包含序列化 Java 或 .NET 物件。

XMS 應用程式可以接收物件訊息，變更其標頭欄位和內容，然後將它傳送至另一個目的地。應用程式也可以複製物件訊息的內文，並使用它來形成另一個物件訊息。XMS 會將物件訊息的內文視為位元組陣列。

在應用程式建立物件訊息之後，訊息內文是可讀取及可寫入的。在應用程式傳送訊息之後，內文仍可讀取及寫入。當應用程式接收物件訊息時，訊息內文是唯讀的。當物件訊息的內文是唯讀時，如果應用程式呼叫 .NET IMessage 介面的 Clear Body 方法，則內文會變成可讀取及可寫入。此方法也會清除主體。

相關參考

應用程式資料元素的資料類型

為了確保 XMS 應用程式可以與 WebSphere MQ for JMS 類別 應用程式交換訊息，這兩個應用程式必須能夠以相同方式解譯訊息內文中的應用程式資料。

位元組訊息

位元組訊息的主體包含位元組串流。主體只包含實際資料，傳送及接收應用程式負責解譯此資料。

對映訊息

對映訊息的內文包含一組名稱/值配對，其中每一個值都有相關聯的資料類型。

串流訊息

串流訊息的內文包含值串流，其中每一個值都有相關聯的資料類型。

文字訊息

文字訊息的內文包含字串。

IObjectMessage (適用於 .NET 介面)

物件訊息是其內文包含序列化 Java 或 .NET 物件的訊息。

串流訊息

串流訊息的內文包含值串流，其中每一個值都有相關聯的資料類型。

值的資料類型是第 64 頁的表 22 中列出的其中一個 XMS 資料類型。

在應用程式建立串流訊息之後，訊息內文是可寫入的。應用程式會呼叫 .NET 的 IStreamMessage 介面的適當寫入方法，將應用程式資料組合到主體中。每次應用程式將值寫入訊息串流時，該值及其資料類型會緊接在應用程式所寫入的前一個值之後組合。XMS 會維護內部游標，以記住組合的最後一個值的位置。

當應用程式傳送訊息時，訊息內文會變成唯讀。在此模式中，應用程式可以多次傳送訊息。

當應用程式接收串流訊息時，訊息內文是唯讀的。應用程式可以使用適用於 .NET 的 IStreamMessage 介面的適當讀取方法，來讀取訊息串流的內容。應用程式依序讀取值，且 XMS 會維護內部游標，以記住所讀取最後一個值的位置。

當應用程式從訊息串流讀取值時，XMS 可以將值轉換為另一個資料類型。例如，若要從訊息串流讀取整數，應用程式可以呼叫 ReadString 方法，以字串形式傳回整數。支援的轉換與 XMS 將內容值從一種資料類型轉換為另一種資料類型時支援的轉換相同。如需受支援轉換的相關資訊，請參閱第 34 頁的『將內容值從一個資料類型隱含轉換為另一個資料類型』。

如果應用程式嘗試從訊息串流讀取值時發生錯誤，則游標不會進階。應用程式可以從錯誤中回復，方法是嘗試將值讀取為另一個資料類型。

當串流訊息的內文是唯讀時，如果應用程式針對 .NET 呼叫 `IStreamMessage` 介面的 `Reset` 方法，則內文會變成唯讀。此方法也會將游標重新定位在訊息串流的開頭。

當串流訊息的內文是唯讀時，如果應用程式呼叫 .NET 之 `IMessage` 介面的 `Clear Body` 方法，則內文會變成唯寫。此方法也會清除主體。

相關概念

將內容值從一個資料類型隱含轉換為另一個資料類型

當應用程式取得內容的值時，XMS 可以將該值轉換為另一個資料類型。許多規則會控管支援哪些轉換，以及 XMS 如何執行轉換。

相關參考

應用程式資料元素的資料類型

為了確保 XMS 應用程式可以與 WebSphere MQ for JMS 類別 應用程式交換訊息，這兩個應用程式必須能夠以相同方式解譯訊息內文中的應用程式資料。

位元組訊息

位元組訊息的主體包含位元組串流。主體只包含實際資料，傳送及接收應用程式負責解譯此資料。

對映訊息

對映訊息的內文包含一組名稱/值配對，其中每一個值都有相關聯的資料類型。

物件訊息

物件訊息的主體包含序列化 Java 或 .NET 物件。

文字訊息

文字訊息的內文包含字串。

`IStreamMessage` (適用於 .NET 介面)

串流訊息是其內文包含值串流的訊息，其中每一個值都具有相關聯的資料類型。主體的內容會循序寫入及讀取。

文字訊息

文字訊息的內文包含字串。

在應用程式建立文字訊息之後，訊息內文是可讀取及可寫入的。在應用程式傳送訊息之後，內文仍可讀取及寫入。當應用程式收到文字訊息時，訊息內文是唯讀的。當文字訊息的內文是唯讀時，如果應用程式呼叫 .NET 的 `IMessage` 介面的「清除內文」方法，則內文會變成可讀取及可寫入。此方法也會清除主體。

相關參考

應用程式資料元素的資料類型

為了確保 XMS 應用程式可以與 WebSphere MQ for JMS 類別 應用程式交換訊息，這兩個應用程式必須能夠以相同方式解譯訊息內文中的應用程式資料。

位元組訊息

位元組訊息的主體包含位元組串流。主體只包含實際資料，傳送及接收應用程式負責解譯此資料。

對映訊息

對映訊息的內文包含一組名稱/值配對，其中每一個值都有相關聯的資料類型。

物件訊息

物件訊息的主體包含序列化 Java 或 .NET 物件。

串流訊息

串流訊息的內文包含值串流，其中每一個值都有相關聯的資料類型。

`ITextMessage` (適用於 .NET 介面)

文字訊息是其內文包含字串的訊息。

訊息選取器

XMS 應用程式使用訊息選取器來選取它要接收的訊息。

當應用程式建立訊息消費者時，它可以建立訊息選取元表示式與消費者的關聯。訊息選取元表示式指定選取準則。

當應用程式連接至 IBM WebSphere MQ V7.0 佇列管理程式時，會在佇列管理程式端完成訊息選取。XMS 不會執行任何選擇，只會遞送它從佇列管理程式收到的訊息，從而提供更好的效能。

應用程式可以建立多個訊息消費者，每一個都有自己的訊息選取元表示式。如果送入訊息符合多個訊息消費者的選取準則，則 XMS 會將訊息遞送至其中每一個消費者。

訊息選取元表示式可以參照訊息的下列內容：

- JMS 定義的內容
- IBM 定義的內容
- 應用程式定義的內容

它也可以參照下列訊息標頭欄位：

- JMSCorrelationID
- JMSDeliveryMode
- JMSMessageID
- JMSPriority
- JMSTimestamp
- JMSType

不過，訊息選取元表示式無法參照訊息內文中的資料。

以下是訊息選取元表示式的範例：

```
JMSPriority > 3 AND manufacturer = 'Jaguar' AND model in ('xj6','xj12')
```

只有在訊息的優先順序大於 3 時，XMS 才會使用此訊息選取元表示式將訊息遞送給訊息消費者；應用程式定義的內容（製造商），其值為 Jaguar；；另一個應用程式定義的內容（模型），其值為 xj6 或 xj12。

在 XMS 中形成訊息選取器表示式的語法規則與在 WebSphere MQ for JMS 類別中形成訊息選取器表示式的語法規則相同。如需如何建構訊息選取元表示式的相關資訊，請參閱 *WebSphere MQ 使用 Java*。請注意，在訊息選取元表示式中，JMS 定義內容的名稱必須是 JMS 名稱，而 IBM 定義內容的名稱必須是 WebSphere MQ for JMS 類別名稱。您無法在訊息選取器表示式中使用 XMS 名稱。

相關參考

XMS 訊息的組件

XMS 訊息包含一個標頭、一組內容及一個內文。

XMS 訊息中的標頭欄位

為了容許 XMS 應用程式與 WebSphere JMS 應用程式交換訊息，XMS 訊息的標頭包含 JMS 訊息標頭欄位。

XMS 訊息的內容

XMS 支援三種類型的訊息內容：JMS 定義的內容、IBM 定義的內容，以及應用程式定義的內容。

XMS 訊息的內文

訊息內文包含應用程式資料。不過，訊息可以沒有內文，且只包含標頭欄位和內容。

將 XMS 訊息對映至 WebSphere MQ 訊息

XMS 訊息的 JMS 標頭欄位和內容會對映至 WebSphere MQ 訊息標頭結構中的欄位。

將 XMS 訊息對映至 WebSphere MQ 訊息

XMS 訊息的 JMS 標頭欄位和內容會對映至 WebSphere MQ 訊息標頭結構中的欄位。

當 XMS 應用程式連接至 WebSphere MQ 佇列管理程式時，在類似情況下，傳送至佇列管理程式的訊息會以 WebSphere MQ for JMS 類別 訊息對映至 WebSphere MQ 訊息的相同方式，對映至 WebSphere MQ 訊息。

如果「目的地」物件的 XMSC_WMQ_TARGET_CLIENT 內容設為 XMSC_WMQ_TARGET_DEST_JMS，傳送至目的地之訊息的 JMS 標頭欄位及內容會對映至 WebSphere MQ 訊息的 MQMD 及 MQRFH2 標頭結構中的欄位。以此方式設定 XMSC_WMQ_TARGET_CLIENT 內容，會假設接收訊息的應用程式可以處理 MQRFH2 標頭。因此，接收端應用程式可能是另一個 XMS 應用程式、WebSphere MQ for JMS 類別 應用程式，或設計用來處理 MQRFH2 標頭的原生 WebSphere MQ 應用程式。

如果「目的地」物件的 XMSC_WMQ_TARGET_CLIENT 內容設為 XMSC_WMQ_TARGET_DEST_MQ，則傳送至目的地之訊息的 JMS 標頭欄位及內容會對映至 WebSphere MQ 訊息的 MQMD 標頭結構中的欄位。訊息不包含 MQRFH2 標頭，且會忽略無法對映至 MQMD 標頭結構中的欄位的任何 JMS 標頭欄位和內容。因此，接收訊息的應用程式可以是原生 WebSphere MQ，其設計目的不是處理 MQRFH2 標頭。

從佇列管理程式接收到的 WebSphere MQ 訊息對映至 XMS 訊息的方式，與在類似情況下 WebSphere MQ 訊息對映至 WebSphere MQ for JMS 類別 訊息的方式相同。

如果送入的 WebSphere MQ 訊息具有 MQRFH2 標頭，則產生的 XMS 訊息具有一個內文，其類型由 MQRFH2 標頭的 mcd 資料夾中包含的 **Msd** 內容值決定。如果 MQRFH2 標頭中沒有 **Msd** 內容，或如果 WebSphere MQ 訊息沒有 MQRFH2 標頭，則產生的 XMS 訊息會有一個主體，其類型是由 MQMD 標頭中 *Format* 欄位的值所決定。如果 *Format* 欄位設為 MQFMT_STRING，則 XMS 訊息是文字訊息。否則，XMS 訊息是位元組訊息。如果 WebSphere MQ 訊息沒有 MQRFH2 標頭，則只會設定那些 JMS 標頭欄位，以及可從 MQMD 標頭中的欄位衍生的內容。

如需將 WebSphere MQ for JMS 類別 訊息對映至 WebSphere MQ 訊息的相關資訊，請參閱 *WebSphere MQ 使用 Java*。

相關參考

[XMS 訊息的組件](#)

XMS 訊息包含一個標頭、一組內容及一個內文。

[XMS 訊息中的標頭欄位](#)

為了容許 XMS 應用程式與 WebSphere JMS 應用程式交換訊息，XMS 訊息的標頭包含 JMS 訊息標頭欄位。

[XMS 訊息的內容](#)

XMS 支援三種類型的訊息內容：JMS 定義的內容、IBM 定義的內容，以及應用程式定義的內容。

[XMS 訊息的內文](#)

訊息內文包含應用程式資料。不過，訊息可以沒有內文，且只包含標頭欄位和內容。

[訊息選取器](#)

XMS 應用程式使用訊息選取器來選取它要接收的訊息。

從 *Message Service Client for .NET* 應用程式讀取及寫入訊息描述子

您可以存取 WebSphere MQ 訊息的所有訊息描述子 (MQMD) 欄位，但 StrucId 和版本除外；BackoutCount 可以讀取但無法寫入。只有在連接至 WebSphere MQ 佇列管理程式 第 6 版 以及更新版本時，才能使用此特性，並由稍後說明的目的地內容所控制。

Message Service Client for .NET 提供的訊息屬性可協助 XMS 應用程式設定 MQMD 欄位，以及驅動 WebSphere MQ 應用程式。

使用發佈/訂閱 傳訊時，有一些限制。例如，將忽略諸如 MsgID 和 CorrelId 之類的 MQMD 欄位 (如果已設定的話)。

當您連接至 WebSphere MQ V6 佇列管理程式時，發佈/訂閱 傳訊無法使用本主題中說明的功能。當 **PROVIDERVERSION** 內容設為 6 時也無法使用。

從 *Message Service Client for .NET* 應用程式存取 IBM WebSphere MQ 訊息資料

您可以存取完整 WebSphere MQ 訊息資料，包括 Message Service Client for .NET 應用程式內的 MQRFH2 標頭 (如果有的話) 及任何其他 WebSphere MQ 標頭 (如果有的話) 作為 JMSBytesMessage 內文。

只有在連接至第 7 版或更新版本的 WebSphere MQ 佇列管理程式，且 WebSphere MQ 傳訊提供者處於標準模式時，才能使用本主題中說明的功能。

目的地物件內容決定 XMS 應用程式如何存取整個 WebSphere MQ 訊息 (包括 MQRFH2 標頭，如果存在的話) 作為 JMSBytesMessage 的內文。

疑難排解

此 章節 提供的資訊可協助您在使用 Message Service Client for .NET 時偵測及處理問題。

此 章節 包含下列 區段：

- [第 70 頁的『.NET 應用程式的追蹤配置』](#)

- [第 73 頁的『.NET 應用程式的 FFDC 配置』](#)
- [第 73 頁的『疑難排解提示』](#)

.NET 應用程式的追蹤配置

對於 XMS .NET 應用程式，您可以從應用程式配置檔以及從 XMS 環境變數配置追蹤。您可以選取要追蹤的元件。追蹤通常在 IBM 支援中心的指引下使用。

XMS .NET 的追蹤是以標準 .NET 追蹤基礎架構為基礎。

依預設會停用錯誤追蹤以外的所有追蹤。您可以使用下列一種方式來開啟追蹤並配置追蹤設定：

- 透過使用應用程式配置檔，其名稱由檔案相關的可執行程式名稱組成，字尾為 `.config`。例如，`text.exe` 的應用程式配置檔將具有名稱 `text.exe.config`。使用應用程式配置檔是對 XMS .NET 應用程式啟用追蹤的偏好方式。如需詳細資料，請參閱第 70 頁的『[使用應用程式配置檔來追蹤配置](#)』。
- 使用 XMS 環境變數作為 XMS C 或 C++ 應用程式。如需詳細資料，請參閱第 72 頁的『[使用 XMS 環境變數的追蹤配置](#)』。

The active trace file has a name of the format `xms_trace<PID>.log` where `<PID>` represents the process ID of the application. 依預設，作用中追蹤檔的大小限制為 20 MB。當達到此限制時，會重新命名並保存檔案。Archived files have names of the format `xms_trace<PID>_YY.MM.DD_HH.MM.SS.log`

依預設，保留的追蹤檔數目是四個，即一個作用中檔案和三個保存檔。這四個檔案會作為捲動緩衝區，直到應用程式停止為止，其中最舊的檔案會被移除並取代為最新的檔案。您可以在應用程式配置檔中指定不同的數目，來變更追蹤檔的數目。不過，必須至少有兩個檔案（一個作用中檔案和一個保存檔）。

有兩種追蹤檔格式可用：

- 基本格式追蹤檔是人類可讀的 WebSphere Application Server 格式。此格式是預設追蹤檔格式。基本格式與追蹤分析器工具不相容。
- 進階格式追蹤檔與追蹤分析器工具相容。您必須在應用程式配置檔中指定要以進階格式產生追蹤檔。

追蹤項目包含下列資訊：

- 記載追蹤的日期和時間
- 類別名稱
- 追蹤類型
- 追蹤訊息

下列範例顯示從部分追蹤擷取：

```
[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
```

在前一個範例中，格式為：

```
[Date Time:Microsecs]      Thread-id  Classname      Trace-type  Methodname  Entry
or Exit
```

其中 Trace-type 是：

- > for Entry
- < for Exit
- d 代表除錯資訊

使用應用程式配置檔來追蹤配置

配置 XMS .NET 應用程式追蹤的偏好方式是使用應用程式配置檔。此檔案的追蹤區段包括定義要追蹤的項目、追蹤檔位置及容許的大小上限、使用的追蹤檔數目及追蹤檔格式的參數。

如果要使用應用程式配置檔來開啟追蹤，您只要將檔案放在與應用程式執行檔相同的目錄中即可。

可依元件及追蹤類型來啟用追蹤。也可以開啟整個追蹤群組的追蹤。您可以個別或集體開啟階層中元件的追蹤。可用的追蹤類型包括：

- 除錯追蹤
- 異常狀況追蹤
- 警告、參考訊息及錯誤訊息
- 方法進入和結束追蹤

下列範例顯示應用程式配置檔的「追蹤」區段中定義的追蹤設定：

```
<?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>
```

第 71 頁的表 23 更詳細地說明參數設定。

參數	說明
traceSpecification=<ComponentName>=<type>=<state>	<p><ComponentName> 是您要追蹤的類別名稱。您可以在此名稱中使用 * 萬用字元。例如，*=all=enabled 指定您要追蹤所有類別，而 IBM.XMS.impl.*=all=enabled 指定您只需要 API 追蹤。</p> <p><type> 可以是下列任何追蹤類型：</p> <ul style="list-style-type: none"> • 全部 • 除錯 • 事件 (event) • EntryExit <p><state> 可以是啟用或停用。</p> <p>您可以使用 ':' (冒號) 定界字元來將多個追蹤元素字串在一起。</p>
traceFilePath="<filename>"	<p>如果您未指定 traceFile 路徑，或者 traceFile 路徑存在但包含空字串，則會將追蹤檔放置在現行目錄中。如果要將追蹤檔儲存在指名的目錄中，請在 traceFile 路徑中指定目錄名稱，例如：</p> <pre>traceFilePath="c:\somepath"</pre>
traceFileSize="<size>"	<p>容許的追蹤檔大小上限。當檔案達到此大小時，即會保存並重新命名。預設上限為 20 KB，其指定為 traceFileSize="20000000"。</p>

參數	說明
traceFileNumber="<number>"	要保留的追蹤檔數目。預設值為 4 (一個作用中檔案及 3 個保存檔)。容許的數目下限為 2。
traceFormat="<format>"	預設追蹤格式是基本。如果您指定 traceFormat="basic", 或未指定 traceFormat, 或 traceFormat 存在但包含空字串, 則會以這種格式產生追蹤檔。 如果您需要與追蹤分析器工具相容的追蹤, 則必須指定 traceFormat="advanced"。

應用程式配置檔中的追蹤設定是動態的, 每次儲存或取代檔案時都會重新讀取。如果在編輯檔案之後在檔案中發現錯誤, 則追蹤檔案設定會回復為其預設值。

相關概念

使用 XMS 環境變數的追蹤配置

除了使用應用程式配置檔之外, 您也可以使用 XMS 環境變數來開啟追蹤。只有在應用程式配置檔中沒有追蹤規格時, 才會使用這些環境變數。

使用 XMS 環境變數的追蹤配置

除了使用應用程式配置檔之外, 您也可以使用 XMS 環境變數來開啟追蹤。只有在應用程式配置檔中沒有追蹤規格時, 才會使用這些環境變數。

若要配置 XMS .NET 應用程式的追蹤, 請在執行應用程式之前設定下列環境變數:

環境變數	預設值	設定	意義
XMS_TRACE_ON	不適用	不適用: 忽略此變數的值	如果設定 XMS_TRACE_ON, 依預設會啟用所有追蹤。
XMS_TRACE_FILE_PATH	現行工作目錄	/dirpath/	要寫入追蹤及 FFDC 記錄的目錄路徑。 除非您指定替代位置, 否則 XMS 會在現行工作目錄中建立 FFDC 及追蹤檔案。您可以將環境變數 XMS_TRACE_FILE_PATH 設為您要 XMS 在其中建立 FFDC 及追蹤檔案之目錄的完整路徑名稱, 以指定替代位置。您必須先設定此環境變數, 然後才能啟動您要追蹤的應用程式。您必須確定執行應用程式所使用的使用者 ID 有權寫入 XMS 建立 FFDC 及追蹤檔案的目錄。

表 24: .NET 追蹤的環境變數設定 (繼續)			
環境變數	預設值	設定	意義
XMS_TRACE_FORMAT	基本	基本, 進階	指定必要的追蹤格式, 可以是 BASIC 或 ADVANCED。預設格式為 BASIC。ADVANCED 格式與追蹤分析器工具相容。
XMS_ 追蹤_ 規格	不適用	請參閱第 70 頁的『使用應用程式配置檔來追蹤配置』。	置換追蹤規格, 其遵循第 70 頁的『使用應用程式配置檔來追蹤配置』中指定的格式。

相關概念

使用應用程式配置檔來追蹤配置

配置 XMS .NET 應用程式追蹤的偏好方式是使用應用程式配置檔。此檔案的追蹤區段包括定義要追蹤的項目、追蹤檔位置及容許的大小上限、使用的追蹤檔數目及追蹤檔格式的參數。

.NET 應用程式的 FFDC 配置

對於 XMS 的 .NET 實作, 會針對每一個 FFDC 產生一個 FFDC 檔案。

「首次失敗資料擷取 (FFDC)」檔案儲存在人類可讀的文字檔中。這些檔案的名稱格式為 `xmsffdc<processID>_<Date>T<Timestamp>.txt`。檔名的範例為 `xmsffdc264_2006.01.06T13.18.52.990955.txt`。時間戳記包含微秒解析度。

檔案以發生異常狀況的日期和時間開始, 後面接著異常狀況類型。這些檔案包括唯一的簡短 probeId, 可用來尋找發生此 FFDC 的位置。

您不需要執行任何配置來開啟 FFDC。依預設, 所有 FFDC 檔案都會寫入現行目錄。不過, 必要的話, 您可以在應用程式配置檔的「追蹤」區段中變更 `ffdcDirectory`, 以指定不同的目錄。在下列範例中, 所有追蹤檔都會記載至目錄 `c:\client\ffdc`:

```
<IBM.XMS>
  <Trace ffdc=true ffdcDirectory="c:\client\ffdc"/>
</IBM.XMS>
```

您可以在應用程式配置檔的「追蹤」區段中, 將 FFDC 設為 `false` 來停用追蹤。

如果您不是使用應用程式配置檔, 則 FFDC 已開啟且追蹤已關閉。

疑難排解提示

使用這些提示可協助您對使用 XMS 的問題進行疑難排解。

XMS 應用程式無法連接至佇列管理程式 (MQRC_NOT_AUTHORIZED)

XMS .NET 用戶端的行為可能不同於 WebSphere MQ JMS 用戶端的行為。因此, 您可能會發現 XMS 應用程式無法連接至佇列管理程式, 雖然 JMS 應用程式可以。

- 此問題的簡單解決方案是嘗試使用長度不超過 12 個字元且在佇列管理程式的權限清單中完全授權的使用者 ID。如果此解決方案不理想, 則不同的但更複雜的方法是使用安全結束程式。如果您需要此問題的進一步協助, 請聯絡 IBM 支援中心以取得協助。
- 如果您設定 Connection Factory 的 XMSC_USERID 內容, 它必須符合已登入使用者的使用者 ID 和密碼。如果您未設定此內容, 依預設, 佇列管理程式會使用已登入使用者的使用者 ID。
- WebSphere MQ 的使用者鑑別是使用目前登入之使用者的詳細資料, 而非 XMSC.USERID 及 XMSC.PASSWORD 欄位。這是設計用來維護與 WebSphere MQ 的一致性。如需鑑別的相關資訊, 請參閱線上 IBM WebSphere MQ 產品說明文件中的鑑別資訊。

重新導向至傳訊引擎的連線

當您連接至 WebSphere Application Server 6.0.2 版服務整合匯流排時，所有連線都可以從原始提供者端點重新導向至匯流排為該用戶端連線選擇的傳訊引擎。這樣做時，它一律會將連線重新導向至主機名稱指定的主機伺服器，而不是 IP 位址指定的主機伺服器。因此，如果無法解決主機名稱，您可能遇到連線問題。

如果要順利連接 WebSphere Application Server 6.0.2 版服務整合匯流排，您可能需要在用戶端主機上的主機名稱和 IP 位址之間提供對映。例如，您可以在用戶端主機上的本端主機表格中指定對映。

支援 Telnet 型密碼鑑別

XMS .NET 即時傳輸通訊協定僅支援簡式 Telnet 型密碼鑑別。XMS .NET 即時傳輸通訊協定不支援保護品質。

設定內容類型 double 的值

在 Windows 64 位元平台上，如果值小於 Double.Epsilon，則在設定或取得內容類型 double 的值時，SetDoubleProperty () 或 GetDoubleProperty () 方法可能無法正常運作。

例如，如果您嘗試針對類型為 double 的內容設定值 4.9E-324，則 Windows 64 位元平台會將它視為 0.0。因此，在分散式傳訊環境中，如果 JMS 或另一個應用程式在任何 Unix 或 Windows 32 位元機器上，將 double 內容的值設為 4.9E-324，且 XMS .Net 在 64 位元機器上執行，則 GetDouble 內容 () 所傳回的值為 0.0。這是 Microsoft .NET 2.0 Framework 的已知問題。

Message Service Client for .NET 參照

此參照小節提供資訊來協助您使用 Message Service Client for .NET。此資訊可協助您執行使用 XMS 進行程式設計所涉及的作業。

.NET 介面

此區段記載 .NET 類別介面及其內容和方法。

下表彙總定義在 IBM.XMS 名稱空間內的所有介面。

介面	說明
第 76 頁的 『IBytesMessage』	位元組訊息是其內文包含位元組串流的訊息。
第 86 頁的 『IConnection』	Connection 物件代表應用程式與傳訊伺服器的作用中連線。
第 89 頁的 『IConnectionFactory』	應用程式使用 Connection Factory 來建立連線。
第 91 頁的 『IConnectionMeta 資料』	ConnectionMeta 資料物件提供連線的相關資訊。
第 91 頁的 『IDestination』	目的地是應用程式傳送訊息的地方，及（或）應用程式從中接收訊息的來源。
第 93 頁的 『ExceptionListener』	應用程式使用異常狀況接聽器，以非同步方式收到連線問題的通知。
第 93 頁的 『IllegalState 異常狀況』	如果應用程式在不正確或不適當的時間呼叫方法，或 XMS 不是處於所要求作業的適當狀態，則 XMS 會擲出此異常狀況。
第 94 頁的 『InitialContext』	應用程式使用 InitialContext 物件，從從受管理物件儲存庫擷取的物件定義建立物件。
第 96 頁的 『InvalidClientIDException』	如果應用程式嘗試設定連線的用戶端 ID，但用戶端 ID 無效或已在使用中，則 XMS 會擲出此異常狀況。

表 25: .NET 類別介面的摘要 (繼續)

介面	說明
第 96 頁的『InvalidDestination 異常狀況』	如果應用程式指定無效的目的地，則 XMS 會擲出此異常狀況。
第 97 頁的『InvalidSelector 異常狀況』	如果應用程式提供語法無效的訊息選取元表示式，則 XMS 會擲出此異常狀況。
第 97 頁的『IMapMessage』	對映訊息是其內文包含一組名稱/值配對的訊息，其中每一個值都具有相關聯的資料類型。
第 106 頁的『IMessage』	訊息物件代表應用程式傳送或接收的訊息。IMessage 是訊息類別的超類別，例如 IMapMessage。
第 112 頁的『IMessageConsumer』	應用程式使用訊息消費者接收已傳送至目的地的訊息。
第 114 頁的『MessageEOFException』	當應用程式讀取位元組訊息的內文時，如果 XMS 發現位元組訊息串流結尾，則 XMS 會擲出此異常狀況。
第 115 頁的『MessageFormat 異常狀況』	如果 XMS 發現格式無效的訊息，則 XMS 會擲出此異常狀況。
第 115 頁的『IMessageListener (委派)』	應用程式使用訊息接聽器來非同步接收訊息。
第 115 頁的『MessageNotReadableException』	如果應用程式嘗試讀取僅寫入的訊息內文，則 XMS 會擲出此異常狀況。
第 116 頁的『MessageNotWritableException』	如果應用程式嘗試寫入唯讀訊息的內文，則 XMS 會擲出此異常狀況。
第 116 頁的『IMessageProducer』	應用程式使用訊息產生者將訊息傳送至目的地。
第 121 頁的『IObjectMessage』	物件訊息是其內文包含序列化 Java 或 .NET 物件的訊息。
第 122 頁的『IPropertyContext』	IPropertyContext 是抽象超類別，包含取得和設定內容的方法。這些方法由其他類別繼承。
第 131 頁的『IQueueBrowser』	應用程式使用佇列瀏覽器來瀏覽佇列上的訊息，而不移除它們。
第 133 頁的『要求者』	應用程式使用要求者來傳送要求訊息，然後等待並接收回覆。
第 135 頁的『ResourceAllocation 異常狀況』	如果 XMS 無法配置方法所需的資源，則 XMS 會擲出此異常狀況。
第 135 頁的『SecurityException』	XMS 會擲出此異常狀況。XMS 也會在權限檢查失敗並阻止方法完成時擲出此異常狀況。
第 135 頁的『ISession』	階段作業是用於傳送及接收訊息的單一執行緒環境定義。
第 146 頁的『IStreamMessage』	串流訊息是其內文包含值串流的訊息，其中每一個值都具有相關聯的資料類型。
第 155 頁的『ITextMessage』	文字訊息是其內文包含字串的訊息。
第 156 頁的『TransactionInProgressException』	如果應用程式因為交易進行中而要求無效的作業，則 XMS 會擲出此異常狀況。

表 25: .NET 類別介面的摘要 (繼續)	
介面	說明
第 156 頁的『 TransactionRolledBackException 』	如果應用程式呼叫 <code>Session.commit()</code> 來確定現行交易，但隨後又回復該交易，則 XMS 會擲出此異常狀況。
XMSC	對於 .NET，XMS 內容名稱及值在此類別中定義為公用常數。如需詳細資料，請參閱第 159 頁的『 XMS 物件的內容 』。
第 157 頁的『 XMSException 』	如果 XMS 在處理 .NET 方法的呼叫時偵測到錯誤，XMS 會擲出異常狀況。異常狀況是封裝錯誤相關資訊的物件。 有不同類型的 XMS 異常狀況，而 <code>XMSException</code> 物件只是一種異常狀況類型。不過， <code>XMSException</code> 類別是其他 XMS 異常狀況類別的超類別。XMS 在任何其他類型的異常狀況都不適當的情況下，會擲出 <code>XMSException</code> 物件。
第 157 頁的『 XMSFactoryFactory 』	如果應用程式未使用受管理物件，請使用此類別來建立 <code>Connection Factory</code> 、佇列及主題。

每一個方法的定義都會列出 XMS 在處理方法呼叫時偵測到錯誤時可能傳回的異常狀況碼。每一個異常狀況碼都由其具名常數代表，其具有對應的異常狀況。

相關概念

建置您自己的應用程式

您可以建置自己的應用程式，就像您建置範例應用程式一樣。

撰寫 XMS 應用程式

此章節提供資訊來協助您撰寫 XMS 應用程式。

撰寫 XMS .NET 應用程式

此章節提供在撰寫 XMS.NET 應用程式時協助您的資訊。

相關參考

XMS 物件的內容

此章節記載 XMS 所定義的物件內容。

IBytesMessage

位元組訊息是其內文包含位元組串流的訊息。

繼承階層：

```

IBM.XMS.IPropertyContext
|
+----IBM.XMS.IMessage
|
+----IBM.XMS.IBytesMessage

```

相關參考

位元組訊息

位元組訊息的主體包含位元組串流。主體只包含實際資料，傳送及接收應用程式負責解譯此資料。

.NET 內容

BodyLength -取得內文長度

介面：

```
Int64 BodyLength
{
    get;
}
```

取得訊息內文為唯讀時的訊息內文長度 (以位元組為單位)。

傳回的值是整個內文的長度，而不論用於讀取訊息的游標目前位於何處。

異常狀況：

- XMSException
- MessageNotReadableException

方法

ReadBoolean -讀取布林值

介面：

```
Boolean ReadBoolean();
```

從位元組訊息串流讀取布林值。

參數：

無

傳回：

所讀取的布林值。

異常狀況：

- XMSException
- MessageNotReadableException
- MessageEOFException

ReadSigned 位元組-讀取位元組

介面：

```
Int16 ReadSignedByte();
```

以帶正負號的 8 位元整數讀取位元組訊息串流中的下一個位元組。

參數：

無

傳回：

讀取的位元組。

異常狀況：

- XMSException
- MessageNotReadableException
- MessageEOFException

ReadBytes - 讀取位元組數

介面:

```
Int32 ReadBytes(Byte[] array);  
Int32 ReadBytes(Byte[] array, Int32 length);
```

從游標現行位置開始，從位元組訊息串流讀取位元組陣列。

參數:

陣列 (輸出)

包含所讀取位元組陣列的緩衝區。如果在呼叫之前要從串流讀取的剩餘位元組數大於或等於緩衝區長度，則會填入緩衝區。否則，緩衝區會部分填入所有剩餘位元組。

如果您在輸入上指定空值指標，方法會跳過位元組而不讀取它們。如果在呼叫之前要從串流讀取的剩餘位元組數大於或等於緩衝區的長度，則跳過的位元組數等於緩衝區的長度。否則，會跳過所有剩餘位元組。游標會保留在位元組訊息串流中要讀取的下一個位置。

長度 (輸入)

緩衝區的長度 (以位元組為單位)

傳回:

讀入緩衝區的位元組數。如果部分填入緩衝區，則該值會小於緩衝區的長度，指出沒有剩餘要讀取的位元組數。如果在呼叫之前沒有剩餘要從串流讀取的位元組數，則值為 `XMSC_END_OF_STREAM`。

如果您在輸入上指定空值指標，則方法不會傳回任何值。

異常狀況:

- `XMSEException`
- `MessageNotReadableException`

ReadChar - 讀取字元

介面:

```
Char ReadChar();
```

以字元形式從位元組訊息串流讀取接下來的 2 個位元組。

參數:

無

傳回:

讀取的字元。

異常狀況:

- `XMSEException`
- `MessageNotReadableException`
- `MessageEOFException`

ReadDouble - 讀取倍精準度浮點數

介面:

```
Double ReadDouble();
```

以倍精準度浮點數字從位元組訊息串流中讀取接下來的 8 個位元組。

參數：

無

傳回：

所讀取的倍精準度浮點數字。

異常狀況：

- XMSException
- MessageNotReadableException
- MessageEOFException

ReadFloat -讀取浮點數

介面：

```
Single ReadFloat();
```

以浮點數字從位元組訊息串流中讀取接下來的 4 個位元組。

參數：

無

傳回：

讀取的浮點數字。

異常狀況：

- XMSException
- MessageNotReadableException
- MessageEOFException

ReadInt -讀取整數

介面：

```
Int32 ReadInt();
```

以帶正負號的 32 位元整數從位元組訊息串流中讀取接下來的 4 個位元組。

參數：

無

傳回：

讀取的整數。

異常狀況：

- XMSException
- MessageNotReadableException
- MessageEOFException

ReadLong - 讀取 *Long* 整數

介面:

```
Int64 ReadLong();
```

以帶正負號的 64 位元整數從位元組訊息串流中讀取接下來的 8 個位元組。

參數:

無

傳回:

讀取的長整數。

異常狀況:

- *XMSEException*
- *MessageNotReadableException*
- *MessageEOFException*

ReadShort - 讀取短整數

介面:

```
Int16 ReadShort();
```

以帶正負號的 16 位元整數從位元組訊息串流中讀取接下來的 2 個位元組。

參數:

無

傳回:

讀取的短整數。

異常狀況:

- *XMSEException*
- *MessageNotReadableException*
- *MessageEOFException*

ReadByte - 讀取未簽署的位元組

介面:

```
Byte ReadByte();
```

以不帶正負號的 8 位元整數讀取位元組訊息串流中的下一個位元組。

參數:

無

傳回:

讀取的位元組。

異常狀況:

- *XMSEException*
- *MessageNotReadableException*
- *MessageEOFException*

ReadUnsignedShort-Read Unsigned Short Integer

介面:

```
Int32 ReadUnsignedShort();
```

以不帶正負號的 16 位元整數從位元組訊息串流中讀取接下來的 2 個位元組。

參數:

無

傳回:

讀取的不帶正負號短整數。

異常狀況:

- XMSEException
- MessageNotReadableException
- MessageEOFException

ReadUTF -讀取 UTF 字串

介面:

```
String ReadUTF();
```

從位元組訊息串流讀取以 UTF-8 編碼的字串。

註: 在呼叫 ReadUTF() 之前, 請確定緩衝區的游標指向位元組訊息串流的開頭。

參數:

無

傳回:

封裝所讀取字串的 String 物件。

異常狀況:

- XMSEException
- MessageNotReadableException
- MessageEOFException

重設-重設

介面:

```
void Reset();
```

將訊息內文置於唯讀模式, 並將游標重新定位在位元組訊息串流的開頭。

參數:

無

傳回:

無效

異常狀況:

- XMSEException

- `MessageNotReadableException`

WriteBoolean -寫入布林值

介面:

```
void WriteBoolean(Boolean value);
```

將布林值寫入位元組訊息串流。

參數:

值 (輸入)

要寫入的布林值。

傳回:

無效

異常狀況:

- `XMSEException`
- `MessageNotWritableException`

WriteByte -寫入位元組

介面:

```
void WriteByte(Byte value);  
void WriteSignedByte(Int16 value);
```

將位元組寫入至位元組訊息串流。

參數:

值 (輸入)

要寫入的位元組。

傳回:

無效

異常狀況:

- `XMSEException`
- `MessageNotWritableException`

WriteBytes -寫入位元組

介面:

```
void WriteBytes(Byte[] value);
```

將位元組陣列寫入位元組訊息串流。

參數:

值 (輸入)

要寫入的位元組陣列。

傳回:

無效

異常狀況：

- XMSEException
- MessageNotWritableException

WriteBytes -寫入局部位元組陣列**介面：**

```
void WriteBytes(Byte[] value, int offset, int length);
```

依照指定長度所定義，將部分位元組陣列寫入位元組訊息串流。

參數：**值 (輸入)**

要寫入的位元組陣列。

偏移 (輸入)

要寫入之位元組陣列的起始點。

長度 (輸入)

要寫入的位元組數。

傳回：

無效

異常狀況：

- XMSEException
- MessageNotWritableException

WriteChar -寫入字元**介面：**

```
void WriteChar(Char value);
```

將字元寫入位元組訊息串流，以 2 個位元組為單位，先高順序位元組。

參數：**值 (輸入)**

要寫入的字元。

傳回：

無效

異常狀況：

- XMSEException
- MessageNotWritableException

WriteDouble -寫入倍精準度浮點數**介面：**

```
void WriteDouble(Double value);
```

將倍精準度浮點數轉換為長整數，並將長整數寫入至位元組訊息串流，以 8 位元組、高順序位元組優先。

參數：**值 (輸入)**

要寫入的倍精準度浮點數字。

傳回：

無效

異常狀況：

- XMSEException
- MessageNotWritableException

WriteFloat - 寫入浮點數**介面：**

```
void WriteFloat(Single value);
```

將浮點數字轉換為整數，並將整數寫入位元組訊息串流為 4 個位元組，先高順序位元組。

參數：**值 (輸入)**

要寫入的浮點數字。

傳回：

無效

異常狀況：

- XMSEException
- MessageNotWritableException

WriteInt - 寫入整數**介面：**

```
void WriteInt(Int32 value);
```

將整數寫入位元組訊息串流，以 4 個位元組為單位，先高順序位元組。

參數：**值 (輸入)**

要寫入的整數。

傳回：

無效

異常狀況：

- XMSEException
- MessageNotWritableException

WriteLong - 寫入 Long 整數**介面：**

```
void WriteLong(Int64 value);
```

將長整數寫入位元組訊息串流，以 8 個位元組為單位，先高順序位元組。

參數：

值 (輸入)

要寫入的長整數。

傳回：

無效

異常狀況：

- XMSEException
- MessageNotWritableException

WriteObject - 寫入物件

介面：

```
void WriteObject(Object value);
```

將指定的物件寫入位元組訊息串流。

參數：

值 (輸入)

要寫入的物件，必須是初始類型的參照。

傳回：

無效

異常狀況：

- XMSEException
- MessageNotWritableException

WriteShort - 寫入短整數

介面：

```
void WriteShort(Int16 value);
```

將短整數寫入位元組訊息串流，以 2 個位元組為單位，先高順序位元組。

參數：

值 (輸入)

要寫入的短整數。

傳回：

無效

異常狀況：

- XMSEException
- MessageNotWritableException

WriteUTF -寫入 UTF 字串

介面:

```
void WriteUTF(String value);
```

將以 UTF-8 編碼的字串寫入位元組訊息串流。

參數:

值 (輸入)

封裝要寫入之字串的 String 物件。

傳回:

無效

異常狀況:

- XMSEException
- MessageNotWritableException

繼承的內容和方法

下列內容繼承自 [IMessage](#) 介面:

[JMSCorrelationID](#)、[JMSDeliveryMode](#)、[JMSDestination](#)、[JMSExpiration](#)、[JMSMessageID](#)、[JMSPriority](#)、[JMSRedelivered](#)、[JMSReplyTo](#)、[JMSTimestamp](#)、[JMSType](#)、[Properties](#)

下列方法繼承自 [IMessage](#) 介面:

[clearBody](#)、[clearProperties](#)、[PropertyExists](#)

下列方法繼承自 [IPropertyContext](#) 介面:

[GetBooleanProperty](#)、[GetByteProperty](#)、[GetBytesProperty](#)、[GetCharProperty](#)、[GetDoubleProperty](#)、[GetFloatProperty](#)、[GetIntProperty](#)、[GetLongProperty](#)、[GetObjectProperty](#)、[GetShortProperty](#)、[GetStringProperty](#)、[SetBooleanProperty](#)、[SetByteProperty](#)、[SetBytesProperty](#)、[SetCharProperty](#)、[SetDoubleProperty](#)、[SetFloatProperty](#)、[SetIntProperty](#)、[SetLongProperty](#)、[SetObjectProperty](#)、[SetShortProperty](#)、[SetStringProperty](#)

IConnection

Connection 物件代表應用程式與傳訊伺服器的作用中連線。

繼承階層:

```
IBM.XMS.IPropertyContext
|
+---- IBM.XMS.IConnection
```

如需 Connection 物件的 XMS 定義內容清單，請參閱 [第 160 頁的『連線的內容』](#)。

.NET 內容

ClientID -取得並設定用戶端 ID

介面:

```
String ClientID
{
    get;
    set;
}
```

取得並設定連線的用戶端 ID。

用戶端 ID 可以由管理者在 `ConnectionFactory` 中預先配置，或透過設定 `ClientID` 來指派。

用戶端 ID 只用來支援發佈/訂閱網域中的可延續訂閱，在點對點網域中會被忽略。

如果應用程式設定連線的用戶端 ID，則在建立連線之後，以及在對連線執行任何其他作業之前，應用程式必須立即這樣做。在這之後，如果應用程式嘗試設定用戶端 ID，呼叫會擲出異常狀況 `IllegalState` 異常狀況。

此內容對分配管理系統的即時連線無效。

異常狀況：

- `XMSEException`
- `IllegalState` 異常狀況
- `InvalidClientIDException`

ExceptionListener -取得並設定異常狀況接聽器

介面：

```
ExceptionListener ExceptionListener
{
    get;
    set;
}
```

取得向連線登錄的異常狀況接聽器，並向連線登錄異常狀況接聽器。

如果未向連線登錄任何異常狀況接聽器，則此方法會傳回空值。如果已向連線登錄異常狀況接聽器，您可以指定空值而非異常狀況接聽器來取消登錄。

如需使用異常狀況接聽器的相關資訊，請參閱 [第 41 頁的『.NET 中的訊息和異常狀況接聽器』](#)。

異常狀況：

- `XMSEException`

meta 資料-取得 *meta* 資料

介面：

```
IConnectionMetaData MetaData
{
    get;
}
```

取得連線的 *meta* 資料。

異常狀況：

- `XMSEException`

方法

關閉-關閉連線

介面：

```
void Close();
```

關閉連線。

如果應用程式嘗試關閉已關閉的連線，則會忽略該呼叫。

參數：

無

傳回：

無效

異常狀況：

- XMSEException

CreateSession - 建立階段作業

介面：

```
ISession CreateSession(Boolean transacted,  
                        AcknowledgeMode acknowledgeMode);
```

建立階段作業。

參數：

已交易 (輸入)

值 **True** 表示階段作業已進行交易。值 **False** 表示階段作業未交易。

對於與分配管理系統的即時連線，該值必須是 **False**。

acknowledgeMode (輸入)

指出如何確認應用程式收到的訊息。此值必須是來自 **AcknowledgeMode** 列舉元的下列其中一項：

AcknowledgeMode.AutoAcknowledge

AcknowledgeMode.ClientAcknowledge

AcknowledgeMode.DupsOkAcknowledge

若為與分配管理系統的即時連線，值必須是 **AcknowledgeMode.AutoAcknowledge** 或 **AcknowledgeMode.DupsOkAcknowledge**

如果階段作業已進行交易，則會忽略此參數。如需確認模式的相關資訊，請參閱 [第 23 頁的『訊息確認』](#)。

傳回：

Session 物件

異常狀況：

- XMSEException

啟動-啟動連線

介面：

```
void Start();
```

開始或重新啟動連線的送入訊息遞送。如果連線已啟動，則會忽略該呼叫。

參數：

無

傳回：

無效

異常狀況：

- XMSEException

停止-停止連線

介面：

```
void Stop();
```

停止遞送連線的送入訊息。如果連線已停止，則會忽略該呼叫。

參數：

無

傳回：

無效

異常狀況：

- XMSEException

繼承的內容和方法

下列方法繼承自 `IPropertyContext` 介面：

[GetBooleanProperty](#)、[GetByteProperty](#)、[GetBytesProperty](#)、[GetCharProperty](#)、[GetDoubleProperty](#)、[GetFloatProperty](#)、[GetIntProperty](#)、[GetLongProperty](#)、[GetObjectProperty](#)、[GetShortProperty](#)、[GetStringProperty](#)、[SetBooleanProperty](#)、[SetByteProperty](#)、[SetBytesProperty](#)、[SetCharProperty](#)、[SetDoubleProperty](#)、[SetFloatProperty](#)、[SetIntProperty](#)、[SetLongProperty](#)、[SetObjectProperty](#)、[SetShortProperty](#)、[SetStringProperty](#)

IConnectionFactory

應用程式使用 `ConnectionFactory` 來建立連線。

繼承階層：

```
IBM.XMS.IPropertyContext
|
+---- IBM.XMS.IConnectionFactory
```

如需 `ConnectionFactory` 物件的 XMS 已定義內容清單，請參閱 [第 160 頁的『ConnectionFactory 的內容』](#)。

相關概念

[ConnectionFactories 和 Connection 物件](#)

`ConnectionFactory` 物件提供應用程式用來建立 `Connection` 物件的範本。應用程式使用 `Connection` 物件來建立 `Session` 物件。

[WebSphere 服務整合匯流排的連線](#)

XMS 應用程式可以使用直接 TCP/IP 連線或使用透過 TCP/IP 的 HTTP 來連接至 WebSphere 服務 Integration Bus。

[WebSphere MQ 佇列管理程式的安全連線](#)

如果要讓 XMS .NET 應用程式能夠建立與 WebSphere MQ 佇列管理程式的安全連線，必須在 `ConnectionFactory` 物件中定義相關內容。

[WebSphere 服務 Integration Bus 傳訊引擎的安全連線](#)

啟用 XMS 應用程式若要建立與 WebSphere 服務 Integration Bus 傳訊引擎的安全連線，必須在 `ConnectionFactory` 物件中定義相關內容。

[受管理物件的內容對映](#)

如果要讓應用程式能夠使用 WebSphere MQ JMS 和 WebSphere Application Server Connection Factory 及目的地物件定義，從這些定義擷取的內容必須對映至可以在 XMS Connection Factory 和目的地設定的對應 XMS 內容。

相關工作

建立受管理物件

必須使用適當的管理工具來建立 XMS 應用程式建立傳訊伺服器連線所需的 ConnectionFactory 和「目的地」物件定義。

相關參考

受管理 ConnectionFactory 物件的必要內容

當應用程式建立 Connection Factory 時，必須定義一些內容來建立傳訊伺服器的連線。

方法

CreateConnection - 建立 Connection Factory (使用預設使用者身分)

介面：

```
IConnection CreateConnection();
```

使用預設內容建立 Connection Factory。

如果您連接至 WebSphere MQ 且未設定 XMSC_USERID，則依預設佇列管理程式會使用已登入使用者的 userID。如果您需要個別使用者的進一步連線層次鑑別，則可以撰寫在 WebSphere MQ 中配置的用戶端鑑別結束程式。

參數：

無

異常狀況：

- XMSEException

CreateConnection - 建立連線 (使用指定的使用者身分)

介面：

```
IConnection CreateConnection(String userId, String password);
```

使用指定的使用者身分建立連線。

如果您連接至 WebSphere MQ 且未設定 XMSC_USERID，則依預設佇列管理程式會使用已登入使用者的 userID。如果您需要個別使用者的進一步連線層次鑑別，則可以撰寫在 WebSphere MQ 中配置的用戶端鑑別結束程式。

以已停止模式建立連線。除非應用程式呼叫 **Connection.start()**，否則不會遞送任何訊息。

參數：

userID (輸入)

字串物件，封裝要用來鑑別應用程式的使用者 ID。如果您提供空值，則會嘗試在沒有鑑別的情況下建立連線。

密碼 (輸入)

字串物件封裝要用來鑑別應用程式的密碼。如果您提供空值，則會嘗試在沒有鑑別的情況下建立連線。

傳回：

Connection 物件。

異常狀況：

- XMSEException

- XMS_X_SECURITY_EXCEPTION

繼承的內容和方法

下列方法繼承自 [IPropertyContext](#) 介面：

[GetBooleanProperty](#)、[GetByteProperty](#)、[GetBytesProperty](#)、[GetCharProperty](#)、[GetDoubleProperty](#)、[GetFloatProperty](#)、[GetIntProperty](#)、[GetLongProperty](#)、[GetObjectProperty](#)、[GetShortProperty](#)、[GetStringProperty](#)、[SetBooleanProperty](#)、[SetByteProperty](#)、[SetBytesProperty](#)、[SetCharProperty](#)、[SetDoubleProperty](#)、[SetFloatProperty](#)、[SetIntProperty](#)、[SetLongProperty](#)、[SetObjectProperty](#)、[SetShortProperty](#)、[SetStringProperty](#)

IConnectionMeta 資料

ConnectionMeta 資料物件提供連線的相關資訊。

繼承階層：

```
IBM.XMS.IPropertyContext
|
+---- IBM.XMS.IConnectionMetaData
```

如需 ConnectionMeta 資料物件的 XMS 定義內容清單，請參閱 [第 165 頁的『ConnectionMeta 資料的內容』](#)。

.NET 內容

JMSXPropertyNames -取得 JMS 定義訊息內容

介面：

```
System.Collections.IEnumerator JMSXPropertyNames
{
    get;
}
```

傳回連線所支援 JMS 定義訊息內容的名稱列舉。

與分配管理系統的即時連線不支援 JMS 定義的訊息內容。

異常狀況：

- XMSException

繼承的內容和方法

下列方法繼承自 [IPropertyContext](#) 介面：

[GetBooleanProperty](#)、[GetByteProperty](#)、[GetBytesProperty](#)、[GetCharProperty](#)、[GetDoubleProperty](#)、[GetFloatProperty](#)、[GetIntProperty](#)、[GetLongProperty](#)、[GetObjectProperty](#)、[GetShortProperty](#)、[GetStringProperty](#)、[SetBooleanProperty](#)、[SetByteProperty](#)、[SetBytesProperty](#)、[SetCharProperty](#)、[SetDoubleProperty](#)、[SetFloatProperty](#)、[SetIntProperty](#)、[SetLongProperty](#)、[SetObjectProperty](#)、[SetShortProperty](#)、[SetStringProperty](#)

IDestination

目的地是應用程式傳送訊息的地方，及（或）應用程式從中接收訊息的來源。

繼承階層：

```
IBM.XMS.IPropertyContext
|
+---- IBM.XMS.IDestination
```


如需「目的地」物件的 XMS 定義內容清單，請參閱 [第 165 頁的『目的地的內容』](#)。

相關概念

ConnectionFactory 和 Connection 物件

ConnectionFactory 物件提供應用程式用來建立 Connection 物件的範本。應用程式使用 Connection 物件來建立 Session 物件。

WebSphere 服務整合匯流排的連線

XMS 應用程式可以使用直接 TCP/IP 連線或使用透過 TCP/IP 的 HTTP 來連接至 WebSphere 服務 Integration Bus。

目的地

XMS 應用程式使用「目的地」物件來指定所傳送訊息的目的地，以及所接收訊息的來源。

目的地萬用字元

XMS 提供對目的地萬用字元的支援，確保萬用字元可以傳遞至比對所需的位置。XMS 可以使用的每一種伺服器類型都有不同的萬用字元架構。

主題統一資源識別碼

主題統一資源識別碼 (URI) 指定主題的名稱；它也可以為它指定一或多個內容。

佇列統一資源識別碼

佇列的 URI 指定佇列的名稱；它也可以指定佇列的一或多個內容。

暫時目的地

XMS 應用程式可以建立及使用暫時目的地。

受管理物件的內容對映

如果要讓應用程式能夠使用 WebSphere MQ JMS 和 WebSphere Application Server Connection Factory 及目的地物件定義，從這些定義擷取的內容必須對映至可以在 XMS Connection Factory 和目的地上設定的對應 XMS 內容。

相關工作

建立受管理物件

必須使用適當的管理工具來建立 XMS 應用程式建立傳訊伺服器連線所需的 ConnectionFactory 和「目的地」物件定義。

相關參考

受管理目的地物件的必要內容

建立目的地的應用程式必須在受管理的目的地物件上設定應用程式的數個內容。

.NET 內容

名稱-取得目的地名稱

介面：

```
String Name
{
    get;
}
```

取得目的地的名稱。名稱是一個字串，封裝佇列名稱或主題名稱。

異常狀況：

- XMSException

TypeId -取得目的地類型

介面：

```
DestinationType TypeId
```

```
{
    get;
}
```

取得目的地類型。目的地類型是下列其中一個值：

```
DestinationType.Queue
DestinationType.Topic
```

異常狀況：

- XMSEException

繼承的內容和方法

下列方法繼承自 `IPropertyContext` 介面：

```
GetBooleanProperty、GetByteProperty、GetBytesProperty、GetCharProperty、GetDoubleProperty、
GetFloatProperty、GetIntProperty、GetLongProperty、GetObjectProperty、GetShortProperty、
GetStringProperty、SetBooleanProperty、SetByteProperty、SetBytesProperty、SetCharProperty、
SetDoubleProperty、SetFloatProperty、SetIntProperty、SetLongProperty、SetObjectProperty、
SetShortProperty、SetStringProperty
```

ExceptionListener

繼承階層：

無

應用程式使用異常狀況接聽器，以非同步方式收到連線問題的通知。

如果應用程式只使用連線來非同步使用訊息，且沒有其他目的，則應用程式瞭解連線問題的唯一方式是使用異常狀況接聽器。在其他情況下，異常狀況接聽器可以提供更直接的方式來瞭解連線問題，而不是等到下一次對 XMS 的同步呼叫。

代理人

ExceptionListener -異常狀況接聽器

介面：

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

連線發生問題時通知應用程式。

實作此委派的方法可以向連線登錄。

如需使用異常狀況接聽器的相關資訊，請參閱 [第 41 頁的『.NET 中的訊息和異常狀況接聽器』](#)。

參數：

異常狀況 (輸入)

指向 XMS 所建立之異常狀況的指標。

傳回：

無效

IllegalState 異常狀況

繼承階層：

```
IBM.XMS.XMSEException
|
+----+ IBM.XMS.Exception
```

```
|  
+----IBM.XMS.IllegalStateException
```

如果應用程式在不正確或不適當的時間呼叫方法，或 XMS 不是處於所要求作業的適當狀態，則 XMS 會擲出此異常狀況。

繼承的內容和方法

下列方法繼承自 `XMSEException` 介面：

[GetErrorCode](#)、[GetLinkedException](#)

InitialContext

應用程式使用 `InitialContext` 物件，從受管理物件儲存庫擷取的物件定義建立物件。

繼承階層：

無

相關概念

`InitialContext` 內容

`InitialContext` 建構子的參數包括受管理物件的儲存庫位置，以統一資源指示器 (URI) 形式提供。為了讓應用程式建立與儲存庫的連線，可能需要提供比 URI 中包含的資訊更多的資訊。

XMS 起始環境定義的 URI 格式

受管理物件的儲存庫位置以統一資源指示器 (URI) 提供。URI 的格式視環境定義類型而定。

擷取受管理物件

XMS 會使用建立 `InitialContext` 物件時所提供的位址，或在 `InitialContext` 內容中，從儲存庫擷取受管理物件。

相關工作

`InitialContext` 物件

應用程式必須建立起始環境定義，以用來建立受管理物件儲存庫的連線，以擷取必要的受管理物件。

.NET 內容

環境-取得環境

介面：

```
Hashtable Environment  
{  
    get;  
}
```

取得環境。

異常狀況：

- 所使用的目錄服務有特定的異常狀況。

建構子

`InitialContext` -建立起始環境定義

介面：

```
InitialContext(Hashtable env);
```

建立 `InitialContext` 物件。

參數：

建立與受管理物件儲存庫的連線所需的資訊，會提供給環境雜湊表中的建構子。

異常狀況：

- XMSEException

方法

AddTo 環境-將新內容新增至環境

介面：

```
Object AddToEnvironment(String propName, Object propVal);
```

將新內容新增至環境。

參數：**propName (輸入)**

封裝要新增之內容名稱的「字串」物件。

propVal (輸入)

要新增的內容值。

傳回：

內容的舊值。

異常狀況：

- 所使用的目錄服務有特定的異常狀況。

關閉-關閉此環境定義

介面：

```
void Close()
```

關閉此環境定義。

參數：

無

傳回：

無

異常狀況：

- 所使用的目錄服務有特定的異常狀況。

查閱-在起始環境定義中查閱物件

介面：

```
Object Lookup(String name);
```

從受管理物件儲存庫擷取的物件定義建立物件。

參數：**名稱 (輸入)**

封裝要擷取之受管理物件名稱的「字串」物件。名稱可以是簡式名稱或複式名稱。如需詳細資料，請參閱第 54 頁的『擷取受管理物件』。

傳回：

視所擷取物件的類型而定，可能是 `IConnectionFactory` 或 `IDestination`。如果函數可以存取目錄，但找不到所需的物件，則會傳回空值。

異常狀況：

- 所使用的目錄服務有特定的異常狀況。

`RemoveFrom` 環境-從環境移除內容

介面：

```
Object RemoveFromEnvironment(String propName);
```

從環境中移除內容。

參數：**propName (輸入)**

封裝要移除之內容名稱的「字串」物件。

傳回：

已移除的物件。

異常狀況：

- 所使用的目錄服務有特定的異常狀況。

InvalidClientIDException

繼承階層：

```
IBM.XMS.XMSException
|
+----IBM.XMS.XMSException
|
+----IBM.XMS.InvalidClientIDException
```

如果應用程式嘗試設定連線的用戶端 ID，但用戶端 ID 無效或已在使用中，則 XMS 會擲出此異常狀況。

繼承的內容和方法

下列方法繼承自 [XMSException](#) 介面：

[GetErrorCode](#)、[GetLinkedException](#)

InvalidDestination 異常狀況

繼承階層：

```
IBM.XMS.XMSException
|
+----IBM.XMS.XMSException
|
+----IBM.XMS.InvalidDestinationException
```

如果應用程式指定無效的目的地，則 XMS 會擲出此異常狀況。

繼承的內容和方法

下列方法繼承自 [XMSException](#) 介面：

[GetErrorCode](#)、[GetLinkedException](#)

InvalidSelector 異常狀況

繼承階層:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.InvalidSelectorException
```

如果應用程式提供語法無效的訊息選取元表示式，則 XMS 會擲出此異常狀況。

繼承的內容和方法

下列方法繼承自 [XMSEException](#) 介面:

[GetErrorCode](#)、[GetLinkedException](#)

IMapMessage

對映訊息是其內文包含一組名稱/值配對的訊息，其中每一個值都具有相關聯的資料類型。

繼承階層:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IMessage
|
+----IBM.XMS.IMapMessage
```

當應用程式取得名稱/值配對的值時，XMS 可以將該值轉換為另一個資料類型。如需這種隱含轉換形式的相關資訊，請參閱 [第 65 頁的『對映訊息』](#)。

相關參考

[對映訊息](#)

對映訊息的內文包含一組名稱/值配對，其中每一個值都有相關聯的資料類型。

.NET 內容

MapNames -取得對映名稱

介面:

```
System.Collections.IEnumerator MapNames
{
    get;
}
```

取得對映訊息內文的名稱列舉。

異常狀況:

- XMSEException

方法

GetBoolean -取得布林值

介面:

```
Boolean GetBoolean(String name);
```


從對映訊息內文中取得依名稱識別的布林值。

參數：

名稱 (輸入)

字串物件封裝用來識別布林值的名稱。

傳回：

從對映訊息內文擷取的布林值。

異常狀況：

- XMSEException

GetByte -取得位元組

介面：

```
Byte    GetByte(String name);  
Int16   GetSignedByte(String name);
```

從對映訊息內文中取得依名稱識別的位元組。

參數：

名稱 (輸入)

封裝識別位元組之名稱的「字串」物件。

傳回：

從對映訊息主體擷取的位元組。 不會對位元組執行任何資料轉換。

異常狀況：

- XMSEException

GetBytes -取得位元組

介面：

```
Byte[]  GetBytes(String name);
```

從對映訊息內文中取得依名稱識別的位元組陣列。

參數：

名稱 (輸入)

封裝名稱的「字串」物件，用來識別位元組陣列。

傳回：

陣列中的位元組數。

異常狀況：

- XMSEException

GetChar -取得字元

介面：

```
Char    GetChar(String name);
```

從對映訊息內文中取得依名稱識別的字元。

參數:

名稱 (輸入)

封裝識別字元之名稱的「字串」物件。

傳回:

從對映訊息內文中擷取的字元。

異常狀況:

- XMSEException

GetDouble -取得倍精準度浮點數

介面:

```
Double GetDouble(String name);
```

從對映訊息內文中取得依名稱識別的倍精準度浮點數字。

參數:

名稱 (輸入)

封裝識別倍精準度浮點數字之名稱的「字串」物件。

傳回:

從對映訊息內文擷取的倍精準度浮點數字。

異常狀況:

- XMSEException

GetFloat -取得浮點數

介面:

```
Single GetFloat(String name);
```

從對映訊息內文中取得依名稱識別的浮點數字。

參數:

名稱 (輸入)

封裝識別浮點數字之名稱的「字串」物件。

傳回:

從對映訊息內文中擷取的浮點數字。

異常狀況:

- XMSEException

GetInt -取得整數

介面:

```
Int32 GetInt(String name);
```

從對映訊息內文中取得依名稱識別的整數。

參數:

名稱 (輸入)

封裝識別整數的名稱的「字串」物件。

傳回:

從對映訊息內文中擷取的整數。

異常狀況:

- XMSEException

GetLong -取得 Long 整數

介面:

```
Int64 GetLong(String name);
```

從對映訊息內文中取得依名稱識別的長整數。

參數:

名稱 (輸入)

封裝識別長整數的名稱的「字串」物件。

傳回:

從對映訊息內文擷取的長整數。

異常狀況:

- XMSEException

GetObject -取得物件

介面:

```
Object GetObject(String name);
```

從對映訊息內文取得名稱/值配對值的參照。名稱/值配對由名稱識別。

參數:

名稱 (輸入)

封裝名稱/值配對名稱的 String 物件。

傳回:

該值是下列其中一個物件類型:

- Boolean
- Byte
- Byte[]
- Char
- Double
- Single
- Int32
- Int64
- Int16
- String

異常狀況:

XMSEException

GetShort -取得短整數

介面:

```
Int16 GetShort(String name);
```

從對映訊息內文中取得依名稱識別的短整數。

參數:

名稱 (輸入)

封裝識別短整數的名稱的「字串」物件。

傳回:

從對映訊息內文中擷取的短整數。

異常狀況:

- XMSEException

GetString -取得字串

介面:

```
String GetString(String name);
```

從對映訊息內文中取得依名稱識別的字串。

參數:

名稱 (輸入)

封裝名稱的「字串」物件，用來識別對映訊息內文中的字串。

傳回:

封裝從對映訊息主體擷取的字串的「字串」物件。如果需要資料轉換，則此值是轉換之後的字串。

異常狀況:

- XMSEException

ItemExists -檢查名稱/值配對是否存在

介面:

```
Boolean ItemExists(String name);
```

檢查對映訊息的內文是否包含具有指定名稱的名稱/值配對。

參數:

名稱 (輸入)

封裝名稱/值配對名稱的 String 物件。

傳回:

- True, 如果對映訊息的內文包含具有指定名稱的名稱/值配對。
- False, 如果對映訊息的內文不包含具有指定名稱的名稱/值配對。

異常狀況:

- XMSEException

SetBoolean - 設定布林值

介面:

```
void SetBoolean(String name, Boolean value);
```

在對映訊息的內文中設定布林值。

參數:

名稱 (輸入)

封裝名稱的「字串」物件，用來識別對映訊息內文中的布林值。

值 (輸入)

要設定的布林值。

傳回:

無效

異常狀況:

- *XMSEException*

SetByte - 設定位元組

介面:

```
void SetByte(String name, Byte value);  
void SetSignedByte(String name, Int16 value);
```

在對映訊息內文中設定位元組。

參數:

名稱 (輸入)

封裝名稱的「字串」物件，用來識別對映訊息內文中的位元組。

值 (輸入)

要設定的位元組。

傳回:

無效

異常狀況:

- *XMSEException*

SetBytes - 設定位元組

介面:

```
void SetBytes(String name, Byte[] value);
```

在對映訊息內文中設定位元組陣列。

參數:

名稱 (輸入)

封裝名稱的「字串」物件，用來識別對映訊息內文中的位元組陣列。

值 (輸入)

要設定的位元組陣列。

傳回：

無效

異常狀況：

- XMSEException

SetChar - 設定字元

介面：

```
void SetChar(String name, Char value);
```

在對映訊息內文中設定 2 個位元組的字元。

參數：

名稱 (輸入)

封裝名稱以識別對映訊息主體中的字元的「字串」物件。

值 (輸入)

要設定的字元。

傳回：

無效

異常狀況：

- XMSEException

SetDouble - 設定倍精準度浮點數

介面：

```
void SetDouble(String name, Double value);
```

在對映訊息內文中設定倍精準度浮點數字。

參數：

名稱 (輸入)

封裝名稱的「字串」物件，用來識別對映訊息內文中的倍精準度浮點數字。

值 (輸入)

要設定的倍精準度浮點數字。

傳回：

無效

異常狀況：

- XMSEException

SetFloat - 設定浮點數

介面：

```
void SetFloat(String name, Single value);
```

在對映訊息內文中設定浮點數字。

參數:

名稱 (輸入)

封裝名稱的「字串」物件，用來識別對映訊息內文中的浮點數字。

值 (輸入)

要設定的浮點數字。

傳回:

無效

異常狀況:

- XMSEException

SetInt -設定整數

介面:

```
void SetInt(String name, Int32 value);
```

在對映訊息的內文中設定整數。

參數:

名稱 (輸入)

封裝名稱以識別對映訊息內文中的整數的「字串」物件。

值 (輸入)

要設定的整數。

傳回:

無效

異常狀況:

- XMSEException

SetLong -設定長整數

介面:

```
void SetLong(String name, Int64 value);
```

在對映訊息內文中設定長整數。

參數:

名稱 (輸入)

封裝名稱以識別對映訊息內文中長整數的「字串」物件。

值 (輸入)

要設定的長整數。

傳回:

無效

異常狀況:

- XMSEException

SetObject - 設定物件

介面:

```
void SetObject(String name, Object value);
```

在對映訊息的內文中設定值 (必須是 XMS 初始類型)。

參數:

名稱 (輸入)

封裝名稱以識別對映訊息內文中的值的「字串」物件。

值 (輸入)

包含要設定之值的位元組陣列。

傳回:

無效

異常狀況:

- XMSEException

SetShort - 設定短整數

介面:

```
void SetShort(String name, Int16 value);
```

在對映訊息內文中設定一個短整數。

參數:

名稱 (輸入)

封裝名稱以識別對映訊息內文中的短整數的「字串」物件。

值 (輸入)

要設定的短整數。

傳回:

無效

異常狀況:

- XMSEException

SetString - 設定字串

介面:

```
void SetString(String name, String value);
```

在對映訊息的內文中設定字串。

參數:

名稱 (輸入)

封裝名稱的「字串」物件，用來識別對映訊息內文中的字串。

值 (輸入)

封裝要設定之字串的「字串」物件。

傳回:

無效

異常狀況：

- XMSException

繼承的內容和方法

下列內容繼承自 `IMessage` 介面：

`JMSCorrelationID`、`JMSDeliveryMode`、`JMSDestination`、`JMSExpiration`、`JMSMessageID`、`JMSPriority`、`JMSRedelivered`、`JMSReplyTo`、`JMSTimestamp`、`JMSType`、`Properties`

下列方法繼承自 `IMessage` 介面：

`clearBody`、`clearProperties`、`PropertyExists`

下列方法繼承自 `IPropertyContext` 介面：

`GetBooleanProperty`、`GetByteProperty`、`GetBytesProperty`、`GetCharProperty`、`GetDoubleProperty`、`GetFloatProperty`、`GetIntProperty`、`GetLongProperty`、`GetObjectProperty`、`GetShortProperty`、`GetStringProperty`、`SetBooleanProperty`、`SetByteProperty`、`SetBytesProperty`、`SetCharProperty`、`SetDoubleProperty`、`SetFloatProperty`、`SetIntProperty`、`SetLongProperty`、`SetObjectProperty`、`SetShortProperty`、`SetStringProperty`

IMessage

訊息物件代表應用程式傳送或接收的訊息。 `IMessage` 是訊息類別的超類別，例如 `IMapMessage`。

繼承階層：

```
IBM.XMS.IPropertyContext
|
+---- IBM.XMS.IMessage
```

如需訊息物件中 JMS 訊息標頭欄位的清單，請參閱第 59 頁的『[XMS 訊息中的標頭欄位](#)』。如需訊息物件的 JMS 定義內容清單，請參閱第 61 頁的『[訊息的 JMS 定義內容](#)』。如需訊息物件的 IBM 定義內容清單，請參閱第 61 頁的『[IBM 定義的訊息內容](#)』。如需訊息物件的 JMS_IBM_MQMD* 內容清單，請參閱第 169 頁的『[JMS_IBM_MQMD* 內容](#)』。

記憶體回收器會刪除訊息。當刪除訊息時，這會釋放它所使用的資源。

相關參考

XMS 訊息

此章節說明 XMS 訊息的結構及內容，並說明應用程式如何處理 XMS 訊息。

.NET 內容

`GetJMSCorrelationID`-取得並設定 `JMSCorrelationID`

介面：

```
String JMSCorrelationID
{
    get;
    set;
}
```

取得並將訊息的相關性 ID 設為 `String` 物件。

異常狀況：

- XMSException

JMSDeliveryMode -取得並設定 JMSDeliveryMode

介面:

```
DeliveryMode JMSDeliveryMode
{
    get;
    set;
}
```

取得並設定訊息的遞送模式。

訊息的遞送模式是下列其中一個值:

```
DeliveryMode.Persistent
DeliveryMode.NonPersistent
```

對於未傳送的新建立訊息，遞送模式為 `DeliveryMode`。持續性，但與分配管理系統的即時連線除外，該分配管理系統的遞送模式為 `DeliveryMode.NonPersistent`。對於收到的訊息，此方法會傳回傳送訊息時 `IMessageProducer.send ()` 呼叫所設定的遞送模式，除非接收端應用程式透過設定 `JMSDeliveryMode` 來變更遞送模式。

異常狀況:

- `XMSEException`

JMSDestination-取得及設定 JMSDestination

介面:

```
IDestination JMSDestination
{
    get;
    set;
}
```

取得並設定訊息的目的地。

傳送訊息時，由 `IMessageProducer.send ()` 呼叫設定目的地。會忽略 `JMSDestination` 的值。不過，您可以使用 `JMSDestination` 來變更所接收訊息的目的地。

對於未傳送的新建立訊息，除非傳送應用程式設定 `JMSDestination` 來設定目的地，否則此方法會傳回空值 `Destination` 物件。對於收到的訊息，除非接收端應用程式透過設定 `JMSDestination` 來變更目的地，否則此方法會傳回 `IMessageProducer.send ()` 呼叫所設定之目的地的 `Destination` 物件。

異常狀況:

- `XMSEException`

JMSExpiration-取得並設定 JMSExpiration

介面:

```
Int64 JMSExpiration
{
    get;
    set;
}
```

取得並設定訊息的有效期限。

當傳送訊息時，`IMessageProducer.send ()` 呼叫會設定有效期限。其值的計算方式是將傳送應用程式指定的存活時間加上傳送訊息的時間。有效期限以 1970 年 1 月 1 日 00:00:00 GMT 開始的毫秒數表示。

對於未傳送的新建立訊息，除非傳送應用程式透過設定 `JMSExpiration` 來設定不同的有效期限，否則有效期限為 0。對於收到的訊息，此方法會傳回傳送訊息時由 `IMessageProducer.send ()` 呼叫所設定的有效期限，除非接收端應用程式透過設定 `JMSExpiration` 來變更有效期限。

如果存活時間為 0，`IMessageProducer.send ()` 呼叫會將有效期限設為 0，以指出訊息不會到期。

XMS 會捨棄過期訊息，且不會將它們遞送至應用程式。

異常狀況：

- `XMSException`

JMSMessageID -取得並設定 *JMSMessageID*

介面：

```
String JMSMessageID
{
    get;
    set;
}
```

取得訊息的訊息 ID 並將其設為封裝訊息 ID 的字串物件。

訊息 ID 是在傳送訊息時由 `IMessageProducer.send ()` 呼叫所設定。對於收到的訊息，除非接收端應用程式設定 `JMSMessageID` 來變更訊息 ID，否則方法會傳回 `IMessageProducer.send ()` 呼叫在傳送訊息時所設定的訊息 ID。

如果訊息沒有訊息 ID，方法會傳回空值。

異常狀況：

- `XMSException`

JMSPriority-取得及設定 *JMSPriority*

介面：

```
Int32 JMSPriority
{
    get;
    set;
}
```

取得並設定訊息的優先順序。

傳送訊息時，`IMessageProducer.send ()` 呼叫會設定優先順序。該值是 0(最低優先順序) 到 9(最高優先順序) 範圍內的整數。

對於未傳送的新建立訊息，除非傳送端應用程式透過設定 `JMSPriority` 來設定不同的優先順序，否則優先順序為 4。對於收到的訊息，除非接收端應用程式透過設定 `JMSPriority` 來變更優先順序，否則此方法會傳回傳送訊息時 `IMessageProducer.send ()` 呼叫所設定的優先順序。

異常狀況：

- `XMSException`

JMSRedelivered-取得及設定 *JMSRedelivered*

介面：

```
Boolean JMSRedelivered
```

```
{
  get;
  set;
}
```

取得是否正在重新遞送訊息的指示，並指出是否正在重新遞送訊息。當收到訊息時，`IMessageConsumer.receive ()` 呼叫會設定此指示。

此內容具有下列值：

- `True`(如果正在重新遞送訊息)。
- `False`(如果未重新遞送訊息)。

對於與分配管理系統的即時連線，該值一律為 `False`。

當傳送訊息時，`IMessageProducer.send ()` 呼叫會忽略 `JMSRedelivered` 所設定的重新遞送指示，當接收訊息時，會忽略並取代為 `IMessageConsumer.receive ()` 呼叫。不過，您可以使用 `JMSRedelivered` 來變更所收到訊息的指示。

異常狀況：

- `XMSEException`

JMSReplyTo -取得並設定 *JMSReplyTo*

介面：

```
IDestination JMSReplyTo
{
  get;
  set;
}
```

取得並設定要傳送訊息回覆的目的地。

此內容的值是要傳送訊息回覆之目的地的「目的地」物件。空值 `Destination` 物件表示沒有預期的回覆。

異常狀況：

- `XMSEException`

JMSTimestamp-取得並設定 *JMSTimestamp*

介面：

```
Int64 JMSTimestamp
{
  get;
  set;
}
```

取得並設定傳送訊息的時間。

時間戳記是在傳送訊息時由 `IMessageProducer.send ()` 呼叫所設定，並以自 1970 年 1 月 1 日 00:00:00 GMT 以來的毫秒數表示。

對於未傳送的新建立訊息，除非傳送應用程式透過設定 `JMSTimestamp` 來設定不同的時間戳記，否則時間戳記為 0。對於收到的訊息，此方法會傳回傳送訊息時 `IMessageProducer.send ()` 呼叫所設定的時間戳記，除非接收端應用程式透過設定 `JMSTimestamp` 來變更時間戳記。

異常狀況：

- `XMSEException`

附註：

1. 如果未定義時間戳記，則方法會傳回 0，但不會擲出異常狀況。

JMSType-取得並設定 *JMSType*

介面：

```
String JMSType
{
    get;
    set;
}
```

取得並設定訊息的類型。

JMSType 的值是封裝訊息類型的字串。如果需要資料轉換，則此值是轉換之後的類型。

異常狀況：

- *XMSEException*

PropertyNames -取得內容

介面：

```
System.Collections.IEnumerator PropertyNames
{
    get;
}
```

取得訊息名稱內容的列舉。

異常狀況：

- *XMSEException*

方法

確認-確認

介面：

```
void Acknowledge();
```

確認此訊息及階段作業收到所有先前未確認的訊息。

如果階段作業的確認模式為 *AcknowledgeMode.ClientAcknowledge*，則應用程式可以呼叫此方法。如果階段作業具有任何其他確認模式或已進行交易，則會忽略對方法的呼叫。

已收到但未確認的訊息可能會重新遞送。

如需確認訊息的相關資訊，請參閱 [第 23 頁的『訊息確認』](#)。

參數：

無

傳回：

無效

異常狀況：

- *XMSEException*

- `IllegalState` 異常狀況

ClearBody -清除內文

介面:

```
void ClearBody();
```

清除訊息內文。不會清除標頭欄位和訊息內容。

如果應用程式清除訊息內文，則內文會與新建立訊息中的空內文保持相同狀態。新建立的訊息中空內文的狀態取決於訊息內文的類型。如需相關資訊，請參閱 [第 63 頁的『XMS 訊息的內文』](#)。

不論訊息內文處於何種狀態，應用程式都可以隨時清除訊息內文。如果訊息內文是唯讀的，應用程式寫入內文的唯一方式是讓應用程式先清除內文。

參數:

無

傳回:

無效

異常狀況:

- `XMSEException`

ClearProperties -清除內容

介面:

```
void ClearProperties();
```

清除訊息的內容。未清除標頭欄位及訊息內文。

如果應用程式清除訊息的內容，內容會變成可讀取及可寫入。

不論內容處於何種狀態，應用程式都可以隨時清除訊息的內容。如果訊息的內容是唯讀的，則內容變成可寫入的唯一方式是讓應用程式先清除內容。

參數:

無

傳回:

無效

異常狀況:

- `XMSEException`

PropertyExists -檢查內容存在

介面:

```
Boolean PropertyExists(String propertyName);
```

檢查訊息是否具有具有指定名稱的內容。

參數:

propertyName (輸入)

封裝內容名稱的 `String` 物件。

傳回：

- True(如果訊息具有具有指定名稱的內容)。
- False(如果訊息沒有具有指定名稱的內容)。

異常狀況：

- XMSException

繼承的內容和方法

下列方法繼承自 `IPropertyContext` 介面：

`GetBooleanProperty`、`GetByteProperty`、`GetBytesProperty`、`GetCharProperty`、`GetDoubleProperty`、`GetFloatProperty`、`GetIntProperty`、`GetLongProperty`、`GetObjectProperty`、`GetShortProperty`、`GetStringProperty`、`SetBooleanProperty`、`SetByteProperty`、`SetBytesProperty`、`SetCharProperty`、`SetDoubleProperty`、`SetFloatProperty`、`SetIntProperty`、`SetLongProperty`、`SetObjectProperty`、`SetShortProperty`、`SetStringProperty`

IMessageConsumer

應用程式使用訊息消費者接收已傳送至目的地的訊息。

繼承階層：

```
IBM.XMS.IPropertyContext
|
+---- IBM.XMS.IMessageConsumer
```

如需 `MessageConsumer` 物件的 XMS 定義內容清單，請參閱 [第 172 頁的『MessageConsumer 的內容』](#)。

.NET 內容

MessageListener -取得並設定訊息接聽器

介面：

```
MessageListener MessageListener
{
    get;
    set;
}
```

取得向訊息消費者登錄的訊息接聽器，並向訊息消費者登錄訊息接聽器。

如果未向訊息消費者登錄任何訊息接聽器，則 `MessageListener` 是空值。如果訊息接聽器已向訊息消費者登錄，您可以改為指定空值來取消登錄。

如需使用訊息接聽器的相關資訊，請參閱 [第 41 頁的『.NET 中的訊息和異常狀況接聽器』](#)。

異常狀況：

- XMSException

MessageSelector -取得訊息選取器

介面：

```
String MessageSelector
{
```

```
    get;  
}
```

取得訊息消費者的訊息選取器。回覆值是封裝訊息選取元表示式的「字串」物件。如果需要資料轉換，則此值是轉換之後的訊息選取元表示式。如果訊息消費者沒有訊息選取器，則 `MessageSelector` 的值是空值 `String` 物件。

異常狀況：

- `XMSEException`

方法

關閉-關閉訊息消費者

介面：

```
void Close();
```

關閉訊息消費者。

如果應用程式嘗試關閉已關閉的訊息消費者，則會忽略呼叫。

參數：

無

傳回：

無效

異常狀況：

- `XMSEException`

接收-接收

介面：

```
IMessage Receive();
```

接收訊息消費者的下一則訊息。呼叫會無限期地等待訊息，或直到訊息消費者關閉為止。

參數：

無

傳回：

訊息物件的指標。如果在呼叫等待訊息時關閉訊息消費者，則此方法會傳回空值訊息物件的指標。

異常狀況：

- `XMSEException`

接收-接收 (含等待間隔)

介面：

```
IMessage Receive(Int64 delay);
```

接收訊息消費者的下一則訊息。呼叫只會等待訊息的指定期間，或直到訊息消費者關閉為止。

參數：**延遲 (輸入)**

呼叫等待訊息的時間 (毫秒)。如果您指定等待間隔為 0，則呼叫會無限期地等待訊息。

傳回：

訊息物件的指標。如果在等待間隔期間沒有任何訊息到達，或如果在呼叫等待訊息時關閉訊息消費者，則此方法會傳回空值訊息物件的指標，但不會擲出異常狀況。

異常狀況：

- [XMSEException](#)

*ReceiveNoWait-Receive with No Wait***介面：**

```
IMessage ReceiveNoWait();
```

接收訊息消費者的下一個訊息 (如果立即可用的話)。

參數：

無

傳回：

訊息物件的指標。如果沒有立即可用的訊息，方法會傳回空值訊息物件的指標。

異常狀況：

- [XMSEException](#)

繼承的內容和方法

下列方法繼承自 [IPropertyContext](#) 介面：

[GetBooleanProperty](#)、[GetByteProperty](#)、[GetBytesProperty](#)、[GetCharProperty](#)、[GetDoubleProperty](#)、[GetFloatProperty](#)、[GetIntProperty](#)、[GetLongProperty](#)、[GetObjectProperty](#)、[GetShortProperty](#)、[GetStringProperty](#)、[SetBooleanProperty](#)、[SetByteProperty](#)、[SetBytesProperty](#)、[SetCharProperty](#)、[SetDoubleProperty](#)、[SetFloatProperty](#)、[SetIntProperty](#)、[SetLongProperty](#)、[SetObjectProperty](#)、[SetShortProperty](#)、[SetStringProperty](#)

MessageEOFException**繼承階層：**

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.MessageEOFException
```

當應用程式讀取位元組訊息的內文時，如果 XMS 發現位元組訊息串流結尾，則 XMS 會擲出此異常狀況。

繼承的內容和方法

下列方法繼承自 [XMSEException](#) 介面：

[GetErrorCode](#)、[GetLinkedException](#)

MessageFormat 異常狀況

繼承階層:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.MessageFormatException
```

如果 XMS 發現格式無效的訊息，則 XMS 會擲出此異常狀況。

繼承的內容和方法

下列方法繼承自 [XMSEException](#) 介面:

[GetErrorCode](#)、[GetLinkedException](#)

IMessageListener (委派)

繼承階層:

無

應用程式使用訊息接聽器來非同步接收訊息。

代理人

MessageListener - 訊息接聽器

介面:

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

以非同步方式將訊息遞送至訊息消費者。

實作此委派的方法可以向連線登錄。

如需使用訊息接聽器的相關資訊，請參閱 [第 41 頁的『.NET 中的訊息和異常狀況接聽器』](#)。

參數:

mesg (輸入)
訊息物件。

傳回:

無效

MessageNotReadableException

繼承階層:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.MessageNotReadableException
```

如果應用程式嘗試讀取僅寫入的訊息內文，則 XMS 會擲出此異常狀況。

繼承的內容和方法

下列方法繼承自 [XMSEException](#) 介面:

[GetErrorCode](#)、[GetLinkedException](#)

MessageNotWritableException

繼承階層:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
|
+----IBM.XMS.MessageNotWritableException
```

如果應用程式嘗試寫入唯讀訊息的內文，則 XMS 會擲出此異常狀況。

繼承的內容和方法

下列方法繼承自 [XMSEException](#) 介面:

[GetErrorCode](#)、[GetLinkedException](#)

IMessageProducer

應用程式使用訊息產生者將訊息傳送至目的地。

繼承階層:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IMessageProducer
```

如需 MessageProducer 物件的 XMS 已定義內容清單，請參閱 [第 172 頁的『MessageProducer 的內容』](#)。

.NET 內容

DeliveryMode -取得並設定預設遞送模式

介面:

```
DeliveryMode DeliveryMode
{
    get;
    set;
}
```

取得並設定訊息產生者所傳送訊息的預設遞送模式。

預設遞送模式具有下列其中一個值:

`DeliveryMode.Persistent`
`DeliveryMode.NonPersistent`

對於與分配管理系統的即時連線，該值必須是 `DeliveryMode.NonPersistent`。

預設值為 `DeliveryMode.Persistent`，但與分配管理系統的即時連線除外，該分配管理系統的預設值為 `DeliveryMode.NonPersistent`。

異常狀況:

- `XMSEException`

目的地-取得目的地

介面:

```
IDestination Destination
```



```
{
  get;
}
```

取得訊息產生者的目的地。

參數：

無

傳回：

目的地物件。如果訊息產生者沒有目的地，則此方法會傳回空值 Destination 物件。

異常狀況：

- XMSEException

DisableMsgID-取得並設定停用訊息 ID 旗標

介面：

```
Boolean DisableMessageID
{
  get;
  set;
}
```

取得接收端應用程式是否需要在訊息產生者所傳送的訊息中包含訊息 ID 的指示，並指出接收端應用程式是否需要在訊息產生者所傳送的訊息中包含訊息 ID。

在佇列管理程式的連線，或在與分配管理系統的即時連線，會忽略此旗標。在服務整合匯流排的連線，允許使用這個旗標。

DisabledMsgID 具有下列值：

- True，如果接收端應用程式不需要訊息 ID 包含在訊息產生者所傳送的訊息中。
- False，如果接收端應用程式確實需要訊息 ID 包含在訊息產生者所傳送的訊息中。

異常狀況：

- XMSEException

DisableMsgTS-取得並設定停用時間戳記旗標

介面：

```
Boolean DisableMessageTimestamp
{
  get;
  set;
}
```

取得接收端應用程式是否需要在訊息產生者所傳送的訊息中包含時間戳記的指示，並指出接收端應用程式是否需要在訊息產生者所傳送的訊息中包含時間戳記。

在與分配管理系統的即時連線，會忽略此旗標。在佇列管理程式的連線，或在服務整合匯流排的連線，允許使用旗標。

DisableMsgTS 具有下列值：

- True，如果接收端應用程式不需要在訊息產生者所傳送的訊息中包含時間戳記。
- False，如果接收端應用程式確實需要在訊息產生者所傳送的訊息中包含時間戳記。

傳回：

異常狀況：

- XMSEException

優先順序-取得及設定預設優先順序

介面：

```
Int32 Priority
{
    get;
    set;
}
```

取得並設定訊息產生者所傳送訊息的預設優先順序。

預設訊息優先順序的值是 0(最低優先順序) 到 9(最高優先順序) 範圍內的整數。

在與分配管理系統的即時連線上，會忽略訊息的優先順序。

異常狀況：

- XMSEException

TimeTo-取得並設定預設存活時間

介面：

```
Int64 TimeToLive
{
    get;
    set;
}
```

取得並設定訊息在到期之前存在的預設時間長度。

時間是從訊息產生者傳送訊息的時間開始測量，並且是預設存活時間 (毫秒)。值 0 表示訊息永不到期。

對於與分配管理系統的即時連線，此值一律為 0。

異常狀況：

- XMSEException

方法

關閉-關閉訊息生產者

介面：

```
void Close();
```

關閉訊息產生者。

如果應用程式嘗試關閉已關閉的訊息產生者，則會忽略該呼叫。

參數：

無

傳回：

無效

異常狀況:

- XMSEException

傳送-傳送

介面:

```
void Send(IMessage msg) ;
```

將訊息傳送至建立訊息產生者時指定的目的地。 使用訊息產生者預設遞送模式、優先順序及存活時間來傳送訊息。

參數:

訊息 (輸入)

訊息物件。

傳回:

無效

異常狀況:

- XMSEException
- MessageFormat 異常狀況
- InvalidDestination 異常狀況

傳送-傳送 (指定遞送模式、優先順序及存活時間)

介面:

```
void Send(IMessage msg,  
          DeliveryMode deliveryMode,  
          Int32 priority,  
          Int64 timeToLive);
```

將訊息傳送至建立訊息產生者時指定的目的地。 使用指定的遞送模式、優先順序及存活時間來傳送訊息。

參數:

訊息 (輸入)

訊息物件。

deliveryMode (輸入)

訊息的遞送模式，必須是下列其中一個值:

DeliveryMode.Persistent
DeliveryMode.NonPersistent

對於與分配管理系統的即時連線，該值必須是 DeliveryMode.NonPersistent。

優先順序 (輸入)

訊息的優先順序。 該值可以是 0(最低優先順序) 到 9(最高優先順序) 範圍內的整數。 在與分配管理系統的即時連線上，會忽略該值。

timeToLive (輸入)

訊息的存活時間 (毫秒)。 值 0 表示訊息永不到期。 對於與分配管理系統的即時連線，該值必須為 0。

傳回:

無效

異常狀況：

- XMSEException
- MessageFormat 異常狀況
- InvalidDestination 異常狀況
- IllegalState 異常狀況

傳送-傳送 (至指定的目的地)

介面：

```
void Send(IDestination dest, IMessage msg) ;
```

如果您使用的是建立訊息產生者時未指定目的地的訊息產生者，請將訊息傳送至指定的目的地。使用訊息產生者預設遞送模式、優先順序及存活時間來傳送訊息。

通常，您在建立訊息產生者時指定目的地，但如果未指定，則每次傳送訊息時都必須指定目的地。

參數：

目的地 (輸入)

目的地物件。

訊息 (輸入)

訊息物件。

傳回：

無效

異常狀況：

- XMSEException
- MessageFormat 異常狀況
- InvalidDestination 異常狀況

傳送-傳送 (至指定的目的地，指定遞送模式、優先順序及存活時間)

介面：

```
void Send(IDestination dest,
          IMessage msg,
          DeliveryMode deliveryMode,
          Int32 priority,
          Int64 timeToLive) ;
```

如果您使用的是建立訊息產生者時未指定目的地的訊息產生者，請將訊息傳送至指定的目的地。使用指定的遞送模式、優先順序及存活時間來傳送訊息。

通常，您在建立訊息產生者時指定目的地，但如果未指定，則每次傳送訊息時都必須指定目的地。

參數：

目的地 (輸入)

目的地物件。

訊息 (輸入)

訊息物件。

deliveryMode (輸入)

訊息的遞送模式，必須是下列其中一個值：

DeliveryMode.Persistent

DeliveryMode.NonPersistent

對於與分配管理系統的即時連線，該值必須是 `DeliveryMode.NonPersistent`。

優先順序 (輸入)

訊息的優先順序。該值可以是 0(最低優先順序) 到 9(最高優先順序) 範圍內的整數。在與分配管理系統的即時連線上，會忽略該值。

timeToLive (輸入)

訊息的存活時間 (毫秒)。值 0 表示訊息永不到期。對於與分配管理系統的即時連線，該值必須為 0。

傳回：

無效

異常狀況：

- `XMSEException`
- `MessageFormat` 異常狀況
- `InvalidDestination` 異常狀況
- `IllegalState` 異常狀況

繼承的內容和方法

下列方法繼承自 `IPropertyContext` 介面：

[GetBooleanProperty](#)、[GetByteProperty](#)、[GetBytesProperty](#)、[GetCharProperty](#)、[GetDoubleProperty](#)、[GetFloatProperty](#)、[GetIntProperty](#)、[GetLongProperty](#)、[GetObjectProperty](#)、[GetShortProperty](#)、[GetStringProperty](#)、[SetBooleanProperty](#)、[SetByteProperty](#)、[SetBytesProperty](#)、[SetCharProperty](#)、[SetDoubleProperty](#)、[SetFloatProperty](#)、[SetIntProperty](#)、[SetLongProperty](#)、[SetObjectProperty](#)、[SetShortProperty](#)、[SetStringProperty](#)

IObjectMessage

物件訊息是其內文包含序列化 Java 或 .NET 物件的訊息。

繼承階層：

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IMessage
|
+----IBM.XMS.IObjectMessage
```

相關參考

物件訊息

物件訊息的主體包含序列化 Java 或 .NET 物件。

.NET 內容

物件-取得並設定物件為位元組

介面：

```
System.Object Object
{
    get;
    set;
}

Byte[] GetObject();
```

取得並設定形成物件訊息內文的物件。

異常狀況：

- XMSEException
- MessageNotReadableException
- MessageEOFException
- MessageNotWritableException

繼承的內容和方法

下列內容繼承自 IMessage 介面：

JMSCorrelationID、JMSDeliveryMode、JMSDestination、JMSExpiration、JMSMessageID、JMSPriority、JMSRedelivered、JMSReplyTo、JMSTimestamp、JMSType、Properties

下列方法繼承自 IMessage 介面：

clearBody、clearProperties、PropertyExists

下列方法繼承自 IPropertyContext 介面：

GetBooleanProperty、GetByteProperty、GetBytesProperty、GetCharProperty、GetDoubleProperty、GetFloatProperty、GetIntProperty、GetLongProperty、GetObjectProperty、GetShortProperty、GetStringProperty、SetBooleanProperty、SetByteProperty、SetBytesProperty、SetCharProperty、SetDoubleProperty、SetFloatProperty、SetIntProperty、SetLongProperty、SetObjectProperty、SetShortProperty、SetStringProperty

IPropertyContext

IPropertyContext 是抽象超類別，包含取得和設定內容的方法。這些方法由其他類別繼承。

繼承階層：

無

方法

GetBoolean 內容-取得布林內容

介面：

```
Boolean GetBooleanProperty(String property_name);
```

取得具有指定名稱之布林內容的值。

參數：

property_name (輸入)

封裝內容名稱的 String 物件。

傳回：

內容的值。

執行緒環境定義：

由子類別決定

異常狀況：

- XMSEException

GetByte 內容-取得位元組內容

介面:

```
Byte    GetByteProperty(String property_name) ;  
Int16   GetSignedByteProperty(String property_name) ;
```

取得依名稱識別的位元組內容值。

參數:

property_name (輸入)

封裝內容名稱的 String 物件。

傳回:

內容的值。

執行緒環境定義:

由子類別決定

異常狀況:

- XMSEException

GetBytes 內容-取得位元組陣列內容

介面:

```
Byte[]  GetBytesProperty(String property_name) ;
```

取得依名稱識別的位元組陣列內容值。

參數:

property_name (輸入)

封裝內容名稱的 String 物件。

傳回:

陣列中的位元組數。

執行緒環境定義:

由子類別決定

異常狀況:

- XMSEException

GetChar 內容-取得字元內容

介面:

```
Char    GetCharProperty(String property_name) ;
```

取得依名稱識別的 2 位元組字元內容值。

參數:

property_name (輸入)

封裝內容名稱的 String 物件。

傳回:

內容的值。

執行緒環境定義:

由子類別決定

異常狀況:

- XMSEException

GetDouble 內容-取得倍精準度浮點內容

介面:

```
Double GetDoubleProperty(String property_name) ;
```

取得依名稱識別之倍精準度浮點內容的值。

參數:

property_name (輸入)

封裝內容名稱的 String 物件。

傳回:

內容的值。

執行緒環境定義:

由子類別決定

異常狀況:

- XMSEException

GetFloat 內容-取得浮點內容

介面:

```
Single GetFloatProperty(String property_name) ;
```

取得依名稱識別的浮點內容值。

參數:

property_name (輸入)

封裝內容名稱的 String 物件。

傳回:

內容的值。

執行緒環境定義:

由子類別決定

異常狀況:

- XMSEException

GetInt 內容- *GetInt* 內容

介面:

```
Int32 GetIntProperty(String property_name) ;
```

取得依名稱識別的整數內容值。

參數:

property_name (輸入)

封裝內容名稱的 String 物件。

傳回：

內容的值。

執行緒環境定義：

由子類別決定

異常狀況：

- XMSEException

GetLong 內容-取得 *Long* 整數內容

介面：

```
Int64 GetLongProperty(String property_name) ;
```

取得依名稱識別的長整數內容值。

參數：**property_name (輸入)**

封裝內容名稱的 *String* 物件。

傳回：

內容的值。

執行緒環境定義：

由子類別決定

異常狀況：

- XMSEException

GetObject 內容-取得物件內容

介面：

```
Object GetObjectProperty( String property_name) ;
```

取得依名稱識別之內容的值及資料類型。

參數：**property_name (輸入)**

封裝內容名稱的 *String* 物件。

傳回：

內容的值，它是下列其中一個物件類型：

Boolean
Byte
Byte[]
Char
Double
Single
Int32
Int64
Int16
String

執行緒環境定義：

由子類別決定

異常狀況:

- XMSEException

GetShort 內容-取得短整數內容

介面:

```
Int16 GetShortProperty(String property_name) ;
```

取得依名稱識別的短整數內容值。

參數:

property_name (輸入)

封裝內容名稱的 String 物件。

傳回:

內容的值。

執行緒環境定義:

由子類別決定

異常狀況:

- XMSEException

GetString 內容- GetString 內容

介面:

```
String GetStringProperty(String property_name) ;
```

取得名稱所識別字串內容的值。

參數:

property_name (輸入)

封裝內容名稱的 String 物件。

傳回:

字串物件封裝作為內容值的字串。如果需要資料轉換，則此值是轉換之後的字串。

執行緒環境定義:

由子類別決定

異常狀況:

- XMSEException

SetBoolean 內容-設定布林內容

介面:

```
void SetBooleanProperty( String property_name, Boolean value) ;
```

設定依名稱識別的布林內容值。

參數:

property_name (輸入)

封裝內容名稱的 String 物件。

值 (輸入)
內容的值。

傳回:
無效

執行緒環境定義:
由子類別決定

異常狀況:

- XMSEException
- MessageNotWritableException

SetByte 內容-設定位元組內容

介面:

```
void SetByteProperty( String property_name, Byte value) ;  
void SetSignedByteProperty( String property_name, Int16 value) ;
```

設定依名稱識別的位元組內容值。

參數:

property_name (輸入)
封裝內容名稱的 String 物件。

值 (輸入)
內容的值。

傳回:
無效

執行緒環境定義:
由子類別決定

異常狀況:

- XMSEException
- MessageNotWritableException

SetBytes 內容-設定位元組陣列內容

介面:

```
void SetBytesProperty( String property_name, Byte[] value ) ;
```

設定依名稱識別的位元組陣列內容值。

參數:

property_name (輸入)
封裝內容名稱的 String 物件。

值 (輸入)
內容的值，這是位元組陣列。

傳回:
無效

執行緒環境定義:
由子類別決定

異常狀況：

- XMSEException
- MessageNotWritableException

SetChar 內容-設定字元內容**介面：**

```
void SetCharProperty( String property_name, Char value) ;
```

設定由名稱識別的 2 位元組字元內容值。

參數：**property_name (輸入)**

封裝內容名稱的 String 物件。

值 (輸入)

內容的值。

傳回：

無效

執行緒環境定義：

由子類別決定

異常狀況：

- XMSEException
- MessageNotWritableException

SetDouble 內容-設定倍精準度浮點內容**介面：**

```
void SetDoubleProperty( String property_name, Double value) ;
```

設定名稱所識別之倍精準度浮點內容的值。

參數：**property_name (輸入)**

封裝內容名稱的 String 物件。

值 (輸入)

內容的值。

傳回：

無效

執行緒環境定義：

由子類別決定

異常狀況：

- XMSEException
- MessageNotWritableException

SetFloat 內容-設定浮點內容

介面:

```
void SetFloatProperty( String property_name, Single value) ;
```

設定依名稱識別的浮點內容值。

參數:

property_name (輸入)

封裝內容名稱的 String 物件。

值 (輸入)

內容的值。

傳回:

無效

執行緒環境定義:

由子類別決定

異常狀況:

- XMSEException
- MessageNotWritableException

SetInt 內容-設定整數內容

介面:

```
void SetIntProperty( String property_name, Int32 value) ;
```

設定依名稱識別的整數內容值。

參數:

property_name (輸入)

封裝內容名稱的 String 物件。

值 (輸入)

內容的值。

傳回:

無效

執行緒環境定義:

由子類別決定

異常狀況:

- XMSEException
- MessageNotWritableException

SetLong 內容-設定 Long 整數內容

介面:

```
void SetLongProperty( String property_name, Int64 value) ;
```

設定由名稱識別的長整數內容值。

參數：**property_name (輸入)**

封裝內容名稱的 String 物件。

值 (輸入)

內容的值。

傳回：

無效

執行緒環境定義：

由子類別決定

異常狀況：

- XMSEException
- MessageNotWritableException

SetObject 內容-設定物件內容

介面：

```
void SetObjectProperty( String property_name, Object value) ;
```

設定依名稱識別之內容的值及資料類型。

參數：**property_name (輸入)**

封裝內容名稱的 String 物件。

objectType (輸入)

內容的值，必須是下列其中一個物件類型：

Boolean
Byte
Byte[]
Char
Double
Single
Int32
Int64
Int16
String

值 (輸入)

內容的值，以位元組陣列表示。

長度 (輸入)

陣列中的位元組數。

傳回：

無效

執行緒環境定義：

由子類別決定

異常狀況：

- XMSEException
- MessageNotWritableException

SetShort 內容-設定短整數內容

介面:

```
void SetShortProperty( String property_name, Int16 value) ;
```

設定依名稱識別的短整數內容值。

參數:

property_name (輸入)

封裝內容名稱的 String 物件。

值 (輸入)

內容的值。

傳回:

無效

執行緒環境定義:

由子類別決定

異常狀況:

- XMSException
- MessageNotWritableException

SetString 內容-設定字串內容

介面:

```
void SetStringProperty( String property_name, String value);
```

設定依名稱識別的字串內容值。

參數:

property_name (輸入)

封裝內容名稱的 String 物件。

值 (輸入)

字串物件封裝作為內容值的字串。

傳回:

無效

執行緒環境定義:

由子類別決定

異常狀況:

- XMSException
- MessageNotWritableException

IQueueBrowser

應用程式使用佇列瀏覽器來瀏覽佇列上的訊息，而不移除它們。

繼承階層:

```
IBM.XMS.IPropertyContext  
System.Collections.IEnumerable  
|  
+---- IBM.XMS.IQueueBrowser
```

.NET 內容

MessageSelector -取得訊息選取器

介面:

```
String MessageSelector
{
    get;
}
```

取得佇列瀏覽器的訊息選取器。

訊息選取元是封裝訊息選取元表示式的「字串」物件。如果需要資料轉換，則此值是轉換之後的訊息選取元表示式。如果佇列瀏覽器沒有訊息選取器，則方法會傳回空值 String 物件。

異常狀況:

- XMSEException

佇列-取得佇列

介面:

```
IDestination Queue
{
    get;
}
```

取得與佇列瀏覽器相關聯的佇列作為代表佇列的目的地物件。

異常狀況:

- XMSEException

方法

關閉-關閉佇列瀏覽器

介面:

```
void Close();
```

關閉佇列瀏覽器。

如果應用程式嘗試關閉已關閉的佇列瀏覽器，則會忽略呼叫。

參數:

無

傳回:

無效

異常狀況:

- XMSEException

GetEnumerator - 取得訊息

介面：

```
IEnumerator GetEnumerator();
```

取得佇列上的訊息清單。

此方法會傳回封裝「訊息」物件清單的列舉元。「訊息」物件的順序與從佇列擷取訊息的順序相同。然後，應用程式可以依序使用列舉元來瀏覽每一個訊息。

當訊息放置在佇列上並從佇列中移除時，會動態更新列舉元。每次應用程式呼叫 `IEnumerator.MoveNext()` 來瀏覽佇列上的下一個訊息時，該訊息會反映佇列的現行內容。

如果應用程式針對佇列瀏覽器多次呼叫此方法，則每一個呼叫都會傳回新的列舉元。因此，應用程式可以使用多個列舉元來瀏覽佇列上的訊息，並在佇列內維護多個位置。

參數：

無

傳回：

反覆運算子物件。

異常狀況：

- `XMSEException`

繼承的內容和方法

下列方法繼承自 [IPropertyContext](#) 介面：

[GetBooleanProperty](#)、[GetByteProperty](#)、[GetBytesProperty](#)、[GetCharProperty](#)、[GetDoubleProperty](#)、[GetFloatProperty](#)、[GetIntProperty](#)、[GetLongProperty](#)、[GetObjectProperty](#)、[GetShortProperty](#)、[GetStringProperty](#)、[SetBooleanProperty](#)、[SetByteProperty](#)、[SetBytesProperty](#)、[SetCharProperty](#)、[SetDoubleProperty](#)、[SetFloatProperty](#)、[SetIntProperty](#)、[SetLongProperty](#)、[SetObjectProperty](#)、[SetShortProperty](#)、[SetStringProperty](#)

要求者

應用程式使用要求者來傳送要求訊息，然後等待並接收回覆。

繼承階層：

無

建構子

要求者-建立要求者

介面：

```
Requestor(ISession sess, IDestination dest);
```

建立要求者。

參數：

sess (輸入)

「階段作業」物件。階段作業不得交易，且必須具有下列其中一種確認模式：

`AcknowledgeMode.AutoAcknowledge`

`AcknowledgeMode.DupsOkAcknowledge`

目的地 (輸入)

這是一個 Destination 物件，代表應用程式可以在其中傳送要求訊息的目的地。

執行緒環境定義:

與要求者相關聯的階段作業

異常狀況:

- XMSEException

方法

關閉-關閉要求者

介面:

```
void Close();
```

關閉要求者。

如果應用程式嘗試關閉已關閉的要求者，則會忽略該呼叫。

註: 當應用程式關閉要求者時，相關聯的階段作業也不會關閉。在這方面，XMS 的行為與 JMS 不同。

參數:

無

傳回:

無效

執行緒環境定義:

任意

異常狀況:

- XMSEException

要求-要求回應

介面:

```
IMessage Request(IMessage requestMessage);
```

傳送要求訊息，然後等待並接收來自接收要求訊息之應用程式的回覆。

對這個方法的呼叫會封鎖，直到收到回覆或階段作業結束為止，兩者以較早的時間為準。

參數:

requestMessage (輸入)

封裝要求訊息的訊息物件。

傳回:

封裝回覆訊息之訊息物件的指標。

執行緒環境定義:

與要求者相關聯的階段作業

異常狀況:

- XMSEException

ResourceAllocation 異常狀況

繼承階層:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
      |
      +----IBM.XMS.ResourceAllocationException
```

如果 XMS 無法配置方法所需的資源，則 XMS 會擲出此異常狀況。

繼承的內容和方法

下列方法繼承自 [XMSEException](#) 介面:

[GetErrorCode](#)、[GetLinkedException](#)

SecurityException

繼承階層:

```
IBM.XMS.XMSEException
|
+----IBM.XMS.XMSEException
      |
      +----IBM.XMS.SecurityException
```

如果提供用來鑑別應用程式的使用者 ID 和密碼遭到拒絕，XMS 會擲出此異常狀況。XMS 也會在權限檢查失敗並阻止方法完成時擲出此異常狀況。

繼承的內容和方法

下列方法繼承自 [XMSEException](#) 介面:

[GetErrorCode](#)、[GetLinkedException](#)

ISession

階段作業是用於傳送及接收訊息的單一執行緒環境定義。

繼承階層:

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.ISession
```

如需 Session 物件的 XMS 定義內容清單，請參閱 [第 172 頁的『階段作業的內容』](#)。

.NET 內容

AcknowledgeMode -取得確認模式

介面:

```
AcknowledgeMode AcknowledgeMode
{
    get;
}
```

取得階段作業的確認模式。

建立階段作業時指定確認模式。

如果階段作業未進行交易，則確認模式是下列其中一個值：

```
AcknowledgeMode.AutoAcknowledge  
AcknowledgeMode.ClientAcknowledge  
AcknowledgeMode.DupsOkAcknowledge
```

如需確認模式的相關資訊，請參閱 [第 23 頁的『訊息確認』](#)。

交易的階段作業沒有確認模式。如果階段作業已進行交易，則方法會改為傳回 `AcknowledgeMode.SessionTransacted`。

異常狀況：

- `XMSEException`

交易-判定是否交易

介面：

```
Boolean Transacted  
{  
    get;  
}
```

判定階段作業是否已交易。

交易陳述如下：

- `True`(如果已交易階段作業)。
- `False`(如果階段作業未進行交易)。

對於與分配管理系統的即時連線，此方法一律會傳回 `False`。

異常狀況：

- `XMSEException`

方法

關閉-關閉階段作業

介面：

```
void Close();
```

關閉階段作業。如果階段作業已進行交易，則會回復進行中的任何交易。

如果應用程式嘗試關閉已關閉的階段作業，則會忽略呼叫。

參數：

無

傳回：

無效

執行緒環境定義：

任意

異常狀況：

- `XMSEException`

確定-確定

介面:

```
void Commit();
```

確定現行交易中處理的所有訊息。

階段作業必須是交易式階段作業。

參數:

無

傳回:

無效

異常狀況:

- XMSEException
- IllegalState 異常狀況
- TransactionRolledBackException

CreateBrowser -建立佇列瀏覽器

介面:

```
IQueueBrowser CreateBrowser(IDestination queue) ;
```

為指定的佇列建立佇列瀏覽器。

參數:

佇列 (輸入)

代表佇列的 Destination 物件。

傳回:

QueueBrowser 物件。

異常狀況:

- XMSEException
- InvalidDestination 異常狀況

CreateBrowser -建立佇列瀏覽器 (含訊息選取器)

介面:

```
IQueueBrowser CreateBrowser(IDestination queue, String selector) ;
```

使用訊息選取器為指定的佇列建立佇列瀏覽器。

參數:

佇列 (輸入)

代表佇列的 Destination 物件。

選取元 (輸入)

封裝訊息選取元表示式的「字串」物件。只有內容符合訊息選取器表示式的那些訊息才會遞送至佇列瀏覽器。

空字串物件表示沒有佇列瀏覽器的訊息選取器。

傳回：

QueueBrowser 物件。

異常狀況：

- XMSEException
- InvalidDestination 異常狀況
- InvalidSelector 異常狀況

CreateBytes 訊息-建立位元組訊息

介面：

```
IBytesMessage CreateBytesMessage();
```

建立位元組訊息。

參數：

無

傳回：

BytesMessage 物件。

異常狀況：

- XMSEException
- IllegalState 異常狀況 (階段作業已關閉)

CreateConsumer -建立消費者

介面：

```
IMessageConsumer CreateConsumer(IDestination dest) ;
```

建立指定目的地的訊息消費者。

參數：**目的地 (輸入)**

目的地物件。

傳回：

MessageConsumer 物件。

異常狀況：

- XMSEException
- InvalidDestination 異常狀況

CreateConsumer -建立消費者 (含訊息選取器)

介面：

```
IMessageConsumer CreateConsumer(IDestination dest,  
                                String selector) ;
```

使用訊息選取器為指定的目的地建立訊息消費者。

參數：**目的地 (輸入)**

目的地物件。

選取元 (輸入)

封裝訊息選取元表示式的「字串」物件。只有內容符合訊息選取器表示式的那些訊息才會遞送至訊息消費者。

空值 String 物件表示訊息消費者沒有訊息選取器。

傳回：

MessageConsumer 物件。

異常狀況：

- XMSEException
- InvalidDestination 異常狀況
- InvalidSelector 異常狀況

CreateConsumer - 建立消費者 (具有訊息選取器及本端訊息旗標)

介面：

```
IMessageConsumer CreateConsumer(IDestination dest,  
                                String selector,  
                                Boolean noLocal) ;
```

使用訊息選取器為指定的目的地建立訊息消費者，如果目的地是主題，則指定訊息消費者是否接收由自己的連線發佈的訊息。

參數：**目的地 (輸入)**

目的地物件。

選取元 (輸入)

封裝訊息選取元表示式的「字串」物件。只有內容符合訊息選取器表示式的那些訊息才會遞送至訊息消費者。

空值 String 物件表示訊息消費者沒有訊息選取器。

noLocal (輸入)

值 True 表示訊息消費者不會接收其自己的連線所發佈的訊息。值 False 表示訊息消費者確實會接收其自己的連線所發佈的訊息。預設值為 False。

傳回：

MessageConsumer 物件。

異常狀況：

- XMSEException
- InvalidDestination 異常狀況
- InvalidSelector 異常狀況

CreateDurable 訂閱者-建立可延續訂閱者

介面：

```
IMessageConsumer CreateDurableSubscriber(IDestination dest,  
                                          String subscription) ;
```

建立指定主題的可延續訂閱者。

此方法對分配管理系統的即時連線無效。

如需可延續訂閱者的相關資訊，請參閱 [第 29 頁的『延續訂閱者』](#)。

參數：

目的地 (輸入)

代表主題的 Destination 物件。主題不能是暫時主題。

訂閱 (輸入)

封裝名稱的「字串」物件，可識別可延續訂閱。此名稱在連線的用戶端 ID 內必須是唯一的。

傳回：

代表可延續訂閱者的 MessageConsumer 物件。

異常狀況：

- XMSEException
- InvalidDestination 異常狀況

CreateDurable 訂閱者-建立可延續訂閱者 (含訊息選取器和本端訊息旗標)

介面：

```
IMessageConsumer CreateDurableSubscriber(IDestination dest,  
                                           String subscription,  
                                           String selector,  
                                           Boolean noLocal) ;
```

使用訊息選取器並指定可延續訂閱者是否接收自己的連線所發佈的訊息，為指定主題建立可延續訂閱者。

此方法對分配管理系統的即時連線無效。

如需可延續訂閱者的相關資訊，請參閱 [第 29 頁的『延續訂閱者』](#)。

參數：

目的地 (輸入)

代表主題的 Destination 物件。主題不能是暫時主題。

訂閱 (輸入)

封裝名稱的「字串」物件，可識別可延續訂閱。此名稱在連線的用戶端 ID 內必須是唯一的。

選取元 (輸入)

封裝訊息選取元表示式的「字串」物件。只有內容符合訊息選取器表示式的那些訊息才會遞送至可延續訂閱者。

空值 String 物件表示可延續訂閱者沒有訊息選取器。

noLocal (輸入)

值 True 表示可延續訂閱者不會接收其自己的連線所發佈的訊息。值 False 表示可延續訂閱者確實會接收其自己的連線所發佈的訊息。預設值為 False。

傳回：

代表可延續訂閱者的 MessageConsumer 物件。

異常狀況：

- XMSEException
- InvalidDestination 異常狀況
- InvalidSelector 異常狀況

CreateMap 訊息-建立對映訊息

介面:

```
IMapMessage CreateMapMessage();
```

建立對映訊息。

參數:

無

傳回:

MapMessage 物件。

異常狀況:

- XMSEException
- IllegalState 異常狀況 (階段作業已關閉)

CreateMessage -建立訊息

介面:

```
IMessage CreateMessage();
```

建立沒有內文的訊息。

參數:

無

傳回:

訊息物件。

異常狀況:

- XMSEException
- IllegalState 異常狀況 (階段作業已關閉)

CreateObject 訊息-建立物件訊息

介面:

```
IObjectMessage CreateObjectMessage();
```

建立物件訊息。

參數:

無

傳回:

ObjectMessage 物件。

異常狀況:

- XMSEException
- IllegalState 異常狀況 (階段作業已關閉)

CreateProducer -建立生產者

介面:

```
IMessageProducer CreateProducer(IDestination dest) ;
```

建立訊息產生者以將訊息傳送至指定的目的地。

參數:

目的地 (輸入)

目的地物件。

如果您指定空值「目的地」物件，則會建立不含目的地的訊息產生者。在此情況下，每次應用程式使用訊息產生者傳送訊息時，都必須指定目的地。

傳回:

MessageProducer 物件。

異常狀況:

- XMSEException
- InvalidDestination 異常狀況

CreateQueue -建立佇列

介面:

```
IDestination CreateQueue(String queue) ;
```

建立目的地物件以代表傳訊伺服器中的佇列。

此方法不會在傳訊伺服器中建立佇列。您必須先建立佇列，然後應用程式才能呼叫此方法。

參數:

佇列 (輸入)

封裝佇列名稱的字串物件，或封裝識別佇列的統一資源識別碼 (URI)。

傳回:

代表佇列的 Destination 物件。

異常狀況:

- XMSEException

CreateStream 訊息-建立串流訊息

介面:

```
IStreamMessage CreateStreamMessage();
```

建立串流訊息。

參數:

無

傳回:

StreamMessage 物件。

異常狀況:

- XMSEException

- XMS_ILLEGAL_STATE_Exception

CreateTemporary 佇列-建立暫時佇列

介面:

```
IDestination CreateTemporaryQueue() ;
```

建立暫時佇列。

暫時佇列的範圍是連線。只有連線所建立的階段作業可以使用暫時佇列。

暫時佇列會保留到明確刪除或連線結束為止，兩者以較早的時間為準。

如需暫時佇列的相關資訊，請參閱 [第 28 頁的『暫時目的地』](#)。

參數:

無

傳回:

代表暫時佇列的 Destination 物件。

異常狀況:

- XMSEException

CreateTemporary 主題-建立暫時主題

介面:

```
IDestination CreateTemporaryTopic() ;
```

建立暫時主題。

暫時主題的範圍是連線。只有連線所建立的階段作業可以使用暫時主題。

暫時主題會保留下來，直到明確刪除或連線結束為止，兩者以較早的時間為準。

如需暫時主題的相關資訊，請參閱 [第 28 頁的『暫時目的地』](#)。

參數:

無

傳回:

代表暫時主題的 Destination 物件。

異常狀況:

- XMSEException

CreateText 訊息-建立文字訊息

介面:

```
ITextMessage CreateTextMessage();
```

建立內文空白的文字訊息。

參數:

無

傳回：

TextMessage 物件。

異常狀況：

- XMSEException

CreateText 訊息-建立文字訊息 (已起始設定)**介面：**

```
ITextMessage CreateTextMessage(String initialValue);
```

建立以指定文字來起始設定內文的文字訊息。

參數：**initialValue (輸入)**

封裝文字以起始設定文字訊息內文的「字串」物件。

無

傳回：

TextMessage 物件。

異常狀況：

- XMSEException

CreateTopic -建立主題**介面：**

```
IDestination CreateTopic(String topic) ;
```

建立目的地物件以代表主題。

參數：**topic (輸入)**

封裝主題名稱的「字串」物件，或封裝識別主題的統一資源識別碼 (URI)。

傳回：

代表主題的 Destination 物件。

異常狀況：

- XMSEException

回復-回復**介面：**

```
void Recover();
```

回復階段作業。訊息遞送已停止，然後以最舊的未確認訊息重新啟動。

階段作業不能是交易式階段作業。

如需回復階段作業的相關資訊，請參閱 [第 23 頁的『訊息確認』](#)。

參數：

無

傳回：

無效

異常狀況：

- XMSEException
- IllegalState 異常狀況

回復-回復**介面：**

```
void Rollback();
```

回復現行交易中處理的所有訊息。

階段作業必須是交易式階段作業。

參數：

無

傳回：

無效

異常狀況：

- XMSEException
- IllegalState 異常狀況

取消訂閱-取消訂閱**介面：**

```
void Unsubscribe(String subscription);
```

刪除可延續訂閱。傳訊伺服器會刪除它所維護的可延續訂閱記錄，且不會再傳送任何訊息給可延續訂閱者。

在下列任何情況下，應用程式無法刪除可延續訂閱：

- 當有可延續訂閱的作用中訊息消費者時
- 當耗用的訊息是擱置交易的一部分時
- 未確認已耗用的訊息時

此方法對分配管理系統的即時連線無效。

參數：**訂閱 (輸入)**

封裝用來識別可延續訂閱之名稱的 String 物件。

傳回：

無效

異常狀況：

- XMSEException
- InvalidDestination 異常狀況
- IllegalState 異常狀況

繼承的內容和方法

下列方法繼承自 [IPropertyContext](#) 介面：

[GetBooleanProperty](#)、[GetByteProperty](#)、[GetBytesProperty](#)、[GetCharProperty](#)、[GetDoubleProperty](#)、[GetFloatProperty](#)、[GetIntProperty](#)、[GetLongProperty](#)、[GetObjectProperty](#)、[GetShortProperty](#)、[GetStringProperty](#)、[SetBooleanProperty](#)、[SetByteProperty](#)、[SetBytesProperty](#)、[SetCharProperty](#)、[SetDoubleProperty](#)、[SetFloatProperty](#)、[SetIntProperty](#)、[SetLongProperty](#)、[SetObjectProperty](#)、[SetShortProperty](#)、[SetStringProperty](#)

IStreamMessage

串流訊息是其內文包含值串流的訊息，其中每一個值都具有相關聯的資料類型。主體的內容會循序寫入及讀取。

繼承階層：

```
IBM.XMS.IPropertyContext
|
+----IBM.XMS.IMessage
|
+----IBM.XMS.IStreamMessage
```

當應用程式從訊息串流讀取值時，XMS 可以將值轉換為另一個資料類型。如需這種隱含轉換形式的相關資訊，請參閱 [第 66 頁的『串流訊息』](#)。

相關參考

串流訊息

串流訊息的內文包含值串流，其中每一個值都有相關聯的資料類型。

方法

ReadBoolean - 讀取布林值

介面：

```
Boolean ReadBoolean();
```

從訊息串流讀取布林值。

參數：

無

傳回：

所讀取的布林值。

異常狀況：

- `XMSException`
- `MessageNotReadableException`
- `MessageEOFException`

ReadByte - 讀取位元組

介面：

```
Int16 ReadSignedByte();
Byte ReadByte();
```

從訊息串流讀取帶正負號的 8 位元整數。

參數：

無

傳回：

讀取的位元組。

異常狀況：

- XMSEException
- MessageNotReadableException
- MessageEOFException

ReadBytes - 讀取位元組數**介面：**

```
Int32 ReadBytes(Byte[] array);
```

從訊息串流讀取位元組陣列。

參數：**陣列 (輸入)**

包含讀取的位元組陣列及緩衝區長度 (以位元組為單位) 的緩衝區。

如果陣列中的位元組數小於或等於緩衝區的長度，則會將整個陣列讀取到緩衝區中。如果陣列中的位元組數大於緩衝區的長度，則緩衝區會填入陣列的一部分，且內部游標會標示要讀取的下一個位元組的位置。後續對 `readBytes()` 的呼叫會從游標的現行位置開始從陣列讀取位元組。

如果您在輸入上指定空值指標，則呼叫會跳過位元組陣列而不讀取它。

傳回：

讀入緩衝區的位元組數。如果部分填入緩衝區，則該值會小於緩衝區的長度，指出陣列中沒有剩餘要讀取的位元組數。如果在呼叫之前沒有剩餘要從陣列讀取的位元組，則值為 `XMSC_END_OF_BYTEARRAY`。

如果您在輸入上指定空值指標，則方法不會傳回任何值。

異常狀況：

- XMSEException
- MessageNotReadableException
- MessageEOFException

ReadChar - 讀取字元**介面：**

```
Char ReadChar();
```

從訊息串流讀取 2 個位元組的字元。

參數：

無

傳回：

讀取的字元。

異常狀況：

- XMSEException
- MessageNotReadableException
- MessageEOFException

ReadDouble -讀取倍精準度浮點數

介面:

```
Double ReadDouble();
```

從訊息串流讀取 8 位元組倍精準度浮點數字。

參數:

無

傳回:

所讀取的倍精準度浮點數字。

異常狀況:

- XMSEException
- MessageNotReadableException
- MessageEOFException

ReadFloat -讀取浮點數

介面:

```
Single ReadFloat();
```

從訊息串流讀取 4 位元組浮點數字。

參數:

無

傳回:

讀取的浮點數字。

異常狀況:

- XMSEException
- MessageNotReadableException
- MessageEOFException

ReadInt -讀取整數

介面:

```
Int32 ReadInt();
```

從訊息串流讀取帶正負號的 32 位元整數。

參數:

無

傳回:

讀取的整數。

異常狀況:

- XMSEException
- MessageNotReadableException

- MessageEOFException

ReadLong - 讀取 *Long* 整數

介面:

```
Int64 ReadLong();
```

從訊息串流讀取帶正負號的 64 位元整數。

參數:

無

傳回:

讀取的長整數。

異常狀況:

- XMSEException
- MessageNotReadableException
- MessageEOFException

ReadObject - 讀取物件

介面:

```
Object ReadObject();
```

從訊息串流讀取值，並傳回其資料類型。

參數:

無

傳回:

該值是下列其中一個物件類型:

Boolean
Byte
Byte[]
Char
Double
Single
Int32
Int64
Int16
String

異常狀況:

XMSEException

ReadShort - 讀取短整數

介面:

```
Int16 ReadShort();
```

從訊息串流讀取帶正負號的 16 位元整數。

參數：
無

傳回：
讀取的短整數。

異常狀況：

- XMSEException
- MessageNotReadableException
- MessageEOFException

ReadString -讀取字串

介面：

```
String ReadString();
```

從訊息串流讀取字串。必要的話，XMS 會將字串中的字元轉換成區域字碼頁。

參數：
無

傳回：
封裝所讀取字串的 String 物件。如果需要資料轉換，則這是轉換之後的字串。

異常狀況：

- XMSEException
- MessageNotReadableException
- MessageEOFException

重設-重設

介面：

```
void Reset();
```

將訊息內文置於唯讀模式，並將游標重新定位在訊息串流的開頭。

參數：
無

傳回：
無效

異常狀況：

- XMSEException
- MessageNotReadableException
- MessageEOFException

WriteBoolean -寫入布林值

介面:

```
void WriteBoolean(Boolean value);
```

將布林值寫入訊息串流。

參數:

值 (輸入)

要寫入的布林值。

傳回:

無效

異常狀況:

- XMSEException
- MessageNotWritableException

WriteByte -寫入位元組

介面:

```
void WriteByte(Byte value);  
void WriteSignedByte(Int16 value);
```

將位元組寫入訊息串流。

參數:

值 (輸入)

要寫入的位元組。

傳回:

無效

異常狀況:

- XMSEException
- MessageNotWritableException

WriteBytes -寫入位元組

介面:

```
void WriteBytes(Byte[] value);
```

將位元組陣列寫入訊息串流。

參數:

值 (輸入)

要寫入的位元組陣列。

長度 (輸入)

陣列中的位元組數。

傳回:

無效

異常狀況：

- XMSEException
- MessageNotWritableException

WriteChar -寫入字元**介面：**

```
void WriteChar(Char value);
```

將字元寫入訊息串流 2 個位元組，先高順序位元組。

參數：**值 (輸入)**

要寫入的字元。

傳回：

無效

異常狀況：

- XMSEException
- MessageNotWritableException

WriteDouble -寫入倍精準度浮點數**介面：**

```
void WriteDouble(Double value);
```

將倍精準度浮點數字轉換為長整數，並將長整數寫入訊息串流為 8 位元組，高順序位元組優先。

參數：**值 (輸入)**

要寫入的倍精準度浮點數字。

傳回：

無效

異常狀況：

- XMSEException
- MessageNotWritableException

WriteFloat -寫入浮點數**介面：**

```
void WriteFloat(Single value);
```

將浮點數字轉換為整數，並將整數寫入訊息串流為 4 位元組，先高順序位元組。

參數：**值 (輸入)**

要寫入的浮點數字。

傳回：

無效

異常狀況：

- XMSEException
- MessageNotWritableException

WriteInt -寫入整數

介面：

```
void WriteInt(Int32 value);
```

將整數寫入訊息串流，以 4 個位元組為單位，先高順序位元組。

參數：

值 (輸入)

要寫入的整數。

傳回：

無效

異常狀況：

- XMSEException
- MessageNotWritableException

WriteLong -寫入 Long 整數

介面：

```
void WriteLong(Int64 value);
```

將長整數寫入訊息串流，以 8 個位元組為單位，先高順序位元組。

參數：

值 (輸入)

要寫入的長整數。

傳回：

無效

異常狀況：

- XMSEException
- MessageNotWritableException

WriteObject -寫入物件

介面：

```
void WriteObject(Object value);
```

將具有指定資料類型的值寫入訊息串流。

參數:

objectType (輸入)

值, 必須是下列其中一個物件類型:

Boolean
Byte
Byte[]
Char
Double
Single
Int32
Int64
Int16
String

值 (輸入)

包含要寫入之值的位元組陣列。

長度 (輸入)

陣列中的位元組數。

傳回:

無效

異常狀況:

- XMSEException

WriteShort -寫入短整數

介面:

```
void WriteShort(Int16 value);
```

將短整數寫入訊息串流, 以 2 個位元組為單位, 先高順序位元組。

參數:

值 (輸入)

要寫入的短整數。

傳回:

無效

異常狀況:

- XMSEException
- MessageNotWritableException

WriteString -寫入字串

介面:

```
void WriteString(String value);
```

將字串寫入訊息串流。

參數：

值 (輸入)

封裝要寫入之字串的 String 物件。

傳回：

無效

異常狀況：

- XMSEException
- MessageNotWritableException

繼承的內容和方法

下列內容繼承自 [IMessage](#) 介面：

[JMSCorrelationID](#)、[JMSDeliveryMode](#)、[JMSDestination](#)、[JMSExpiration](#)、[JMSMessageID](#)、[JMSPriority](#)、[JMSRedelivered](#)、[JMSReplyTo](#)、[JMSTimestamp](#)、[JMSType](#)、[Properties](#)

下列方法繼承自 [IMessage](#) 介面：

[clearBody](#)、[clearProperties](#)、[PropertyExists](#)

下列方法繼承自 [IPropertyContext](#) 介面：

[GetBooleanProperty](#)、[GetByteProperty](#)、[GetBytesProperty](#)、[GetCharProperty](#)、[GetDoubleProperty](#)、[GetFloatProperty](#)、[GetIntProperty](#)、[GetLongProperty](#)、[GetObjectProperty](#)、[GetShortProperty](#)、[GetStringProperty](#)、[SetBooleanProperty](#)、[SetByteProperty](#)、[SetBytesProperty](#)、[SetCharProperty](#)、[SetDoubleProperty](#)、[SetFloatProperty](#)、[SetIntProperty](#)、[SetLongProperty](#)、[SetObjectProperty](#)、[SetShortProperty](#)、[SetStringProperty](#)

ITextMessage

文字訊息是其內文包含字串的訊息。

繼承階層：

```
IBM.XMS.IPropertyContext
|
+---- IBM.XMS.IMessage
      |
      +---- IBM.XMS.ITextMessage
```

相關參考

[文字訊息](#)

文字訊息的內文包含字串。

.NET 內容

文字-取得及設定文字

介面：

```
String Text
{
    get;
    set;
}
```

取得並設定形成文字訊息內文的字串。

必要的話， XMS 會將字串中的字元轉換成區域字碼頁。

異常狀況：

- XMSEException

- [MessageNotReadableException](#)
- [MessageNotWritableException](#)
- [MessageEOFException](#)

繼承的內容和方法

下列內容繼承自 [IMessage](#) 介面：

[JMSCorrelationID](#)、[JMSDeliveryMode](#)、[JMSDestination](#)、[JMSExpiration](#)、[JMSMessageID](#)、[JMSPriority](#)、[JMSRedelivered](#)、[JMSReplyTo](#)、[JMSTimestamp](#)、[JMSType](#)、[Properties](#)

下列方法繼承自 [IMessage](#) 介面：

[clearBody](#)、[clearProperties](#)、[PropertyExists](#)

下列方法繼承自 [IPropertyContext](#) 介面：

[GetBooleanProperty](#)、[GetByteProperty](#)、[GetBytesProperty](#)、[GetCharProperty](#)、[GetDoubleProperty](#)、[GetFloatProperty](#)、[GetIntProperty](#)、[GetLongProperty](#)、[GetObjectProperty](#)、[GetShortProperty](#)、[GetStringProperty](#)、[SetBooleanProperty](#)、[SetByteProperty](#)、[SetBytesProperty](#)、[SetCharProperty](#)、[SetDoubleProperty](#)、[SetFloatProperty](#)、[SetIntProperty](#)、[SetLongProperty](#)、[SetObjectProperty](#)、[SetShortProperty](#)、[SetStringProperty](#)

TransactionInProgressException

繼承階層：

```

IBM.XMS.XMSException
|
+----+IBM.XMS.XMSException
      |
      +----+IBM.XMS.TransactionInProgressException
  
```

如果應用程式因為交易進行中而要求無效的作業，則 XMS 會擲出此異常狀況。

繼承的內容和方法

下列方法繼承自 [XMSException](#) 介面：

[GetErrorCode](#)、[GetLinkedException](#)

TransactionRolledBackException

繼承階層：

```

IBM.XMS.XMSException
|
+----+IBM.XMS.XMSException
      |
      +----+IBM.XMS.TransactionRolledBackException
  
```

如果應用程式呼叫 `Session.commit()` 來確定現行交易，但隨後又回復該交易，則 XMS 會擲出此異常狀況。

繼承的內容和方法

下列方法繼承自 [XMSException](#) 介面：

[GetErrorCode](#)、[GetLinkedException](#)

XMSException

如果 XMS 在處理 .NET 方法的呼叫時偵測到錯誤，XMS 會擲出異常狀況。異常狀況是封裝錯誤相關資訊的物件。

繼承階層：

```
System.Exception
|
+----IBM.XMS.XMSException
```

有不同類型的 XMS 異常狀況，而 XMSException 物件只是一種異常狀況類型。不過，XMSException 類別是其他 XMS 異常狀況類別的超類別。XMS 在任何其他類型的異常狀況都不適當的情況下，會擲出 XMSException 物件。

.NET 內容

ErrorCode -取得錯誤碼

介面：

```
public String ErrorCode
{
    get {return errorCode_;}
}
```

取得錯誤碼。

異常狀況：

- XMSException

LinkedException -取得鏈結的異常狀況

介面：

```
public Exception LinkedException
{
    get { return linkedException_;}
    set { linkedException_ = value;}
}
```

取得異常狀況鏈中的下一個異常狀況。

如果鏈中沒有其他異常狀況，此方法會傳回空值。

異常狀況：

- XMSException

XMSFactoryFactory

如果應用程式未使用受管理物件，請使用此類別來建立 Connection Factory、佇列及主題。

繼承階層：

無

.NET 內容

meta 資料-擷取 *meta* 資料

介面:

```
IConnectionMetaData MetaData
```

取得適用於 XMSFactoryFactory 物件的連線類型的 *meta* 資料。

異常狀況:

無

方法

CreateConnectionFactory-建立 *ConnectionFactory*

介面:

```
IConnectionFactory CreateConnectionFactory();
```

建立宣告類型的 *ConnectionFactory* 物件。

參數:

無

傳回:

ConnectionFactory 物件。

異常狀況:

- *XMSEException*

CreateQueue -建立佇列

介面:

```
IDestination CreateQueue(String name);
```

建立目的地物件以代表傳訊伺服器中的佇列。

此方法不會在傳訊伺服器中建立佇列。您必須先建立佇列，然後應用程式才能呼叫此方法。

參數:

名稱 (輸入)

封裝佇列名稱的字串物件，或封裝識別佇列的統一資源識別碼 (URI)。

傳回:

代表佇列的 *Destination* 物件。

異常狀況:

- *XMSEException*

CreateTopic -建立主題

介面:

```
IDestination CreateTopic(String name);
```

建立目的地物件以代表主題。

參數：

名稱 (輸入)

封裝主題名稱的「字串」物件，或封裝識別主題的統一資源識別碼 (URI)。

傳回：

代表主題的 Destination 物件。

異常狀況：

- XMSException

GetInstance -取得 *XMSFactoryFactory* 的實例

介面：

```
static XMSFactoryFactory GetInstance(int connectionType);
```

建立 *XMSFactoryFactory* 的實例。XMS 應用程式使用 *XMSFactoryFactory* 物件來取得 *ConnectionFactory* 物件的參照，該物件適用於所需的通訊協定類型。然後，這個 *ConnectionFactory* 物件只能產生該通訊協定類型的連線。

參數：

connectionType (輸入)

ConnectionFactory 物件產生連線的連線類型：

- XMSC.CT_WPM
- XMSC.CT_RTT
- XMSC.CT_WMQ

傳回：

專用於宣告連線類型的 *XMSFactoryFactory* 物件。

異常狀況：

- NotSupported 異常狀況

XMS 物件的內容

此章節 記載 XMS 所定義的物件內容。

章節 包含下列 區段：

- [第 160 頁的『連線的內容』](#)
- [第 160 頁的『ConnectionFactory 的內容』](#)
- [第 165 頁的『ConnectionMeta 資料的內容』](#)
- [第 165 頁的『目的地的內容』](#)
- [第 167 頁的『InitialContext 的內容』](#)
- [第 168 頁的『訊息的內容』](#)
- [第 172 頁的『MessageConsumer 的內容』](#)
- [第 172 頁的『MessageProducer 的內容』](#)
- [第 172 頁的『階段作業的內容』](#)

每一個 區段 列出指定類型之物件的內容，並提供每一個內容的簡要說明。

此章節 也包含 [第 172 頁的『內容定義』](#) 區段，其提供每一個內容的定義。

如果應用程式定義其本身在此 章節中所說明的物件內容，則不會導致錯誤，但可能會導致無法預期的結果。

註: 此區段中的內容名稱及值以 `XMSC.NAME` 格式顯示, 這是用於 C 及 C++ 的格式。不過, 在 .NET 中, 內容名稱的格式可以是 `XMSC.NAME` 或 `XMSC_NAME`, 視您使用它的方式而定:

- 如果您指定內容, 則內容名稱的格式必須為 `XMSC.NAME`, 如下列範例所示:

```
cf.SetStringProperty(XMSC.WMQ_CHANNEL, "DOTNET.SVRCONN");
```

- 如果您指定字串, 內容名稱的格式必須是 `XMSC_NAME`, 如下列範例所示:

```
cf.SetStringProperty("XMSC_WMQ_CHANNEL", "DOTNET.SVRCONN");
```

在 .NET 中, 提供內容名稱和值作為 XMSC 類別中的常數。這些常數會識別字串, 並供任何 XMS.NET 應用程式使用。如果您使用這些預先定義的常數, 則內容名稱及值的格式為 `XMSC.NAME`, 例如, 您將使用 `XMSC.USERID`, 而不是 `XMSC_USERID`。

資料類型也採用 C/C++ 的格式。您可以在第 38 頁的『.NET 的資料類型』中找到 .NET 的對應值。

相關概念

建置您自己的應用程式

您可以建置自己的應用程式, 就像您建置範例應用程式一樣。

相關參考

.NET 介面

此區段記載 .NET 類別介面及其內容和方法。

連線的內容

Connection 物件內容的概觀, 以及指向更詳細參照資訊的鏈結。

內容名稱	說明
第 201 頁的『 XMSC_WMQ_RESOLVED_QUEUE_MANAGER 』	此內容用來取得它所連接的佇列管理程式名稱。
第 201 頁的『 XMSC_WMQ_RESOLVED_QUEUE_MANAGER_ID 』	在連線之後, 此內容會移入佇列管理程式的 ID。
XMSC_WPM_CONNECTION_PROTOCOL	建立傳訊引擎連線時使用的通訊協定。這是唯讀內容。
XMSC_WPM_HOST_NAME	包含應用程式所連接傳訊引擎的系統的主機名稱或 IP 位址。這是唯讀內容。
XMSC_WPM_ME_NAME	應用程式所連接的傳訊引擎的名稱。這是唯讀內容。
XMSC_WPM_PORT	應用程式所連接的傳訊引擎接聽的埠號。這是唯讀內容。

Connection 物件也具有唯讀內容, 這些內容衍生自用來建立連線之 Connection Factory 的內容。這些內容不僅衍生自建立連線時所設定的 Connection Factory 內容, 還衍生自未設定之內容的預設值。這些內容只包含與應用程式所連接的傳訊伺服器類型相關的內容。內容的名稱與 Connection Factory 內容的名稱相同。

ConnectionFactory 的內容

ConnectionFactory 物件內容的概觀, 以及指向更詳細參照資訊的鏈結。

內容名稱	說明
第 181 頁的『 XMSC_ASYNC_EXCEPTIONS 』	此內容決定 XMS 是否只在連線已損毀或者 XMS API 呼叫有發生任何非同步異常狀況時通知 ExceptionListener。此內容適用於從這個登錄了 ExceptionListener 的 ConnectionFactory 建立的所有連線。

表 27: <i>ConnectionFactory</i> 的內容 (繼續)	
內容名稱	說明
<u>XMSC_CLIENT_ID</u>	連線的用戶端 ID。
<u>XMSC_CONNECTION_TYPE</u>	應用程式所連接傳訊伺服器的類型。
<u>XMSC_PASSWORD</u>	應用程式在嘗試連接至傳訊伺服器時可以用來鑑別應用程式的密碼。
第 185 頁的 『 <u>XMSC_RTT_BROKER_PING_INTERVAL</u> 』	以毫秒為單位的時間間隔，在這段時間之後 XMS.NET 會檢查即時傳訊伺服器的連線以偵測任何活動。
<u>XMSC_RTT_CONNECTION_PROTOCOL</u>	用來即時連線至分配管理系統的通訊協定。
<u>XMSC_RTT_HOST_NAME</u>	分配管理系統在其中執行的系統的主機名稱或 IP 位址。
<u>XMSC_RTT_LOCAL_ADDRESS</u>	用於即時連線至分配管理系統的本端網路介面的主機名稱或 IP 位址。
<u>XMSC_RTT_MULTICAST</u>	Connection Factory 或目的地的多重播送設定。
<u>XMSC_RTT_PORT</u>	分配管理系統在其中接聽送入要求的埠的號碼。
<u>XMSC_USERID</u>	應用程式在嘗試連接至傳訊伺服器時可以用來鑑別應用程式的使用者 ID。
<u>XMSC_WMQ_BROKER_CONTROLQ</u>	分配管理系統使用的控制佇列的名稱。 註: 此內容可以與 IBM Message Service Client for .NET 2.0 版搭配使用，但對於連接至 IBM WebSphere MQ 7.0 版佇列管理程式的應用程式沒有作用，除非 Connection Factory 的 XMSC_WMQ_PROVIDER_VERSION 內容設為小於 7 的版本號碼。
<u>XMSC_WMQ_BROKER_PUBQ</u>	由分配管理系統監視且應用程式在其中傳送所發佈訊息的佇列的名稱。 註: 此內容可以與 IBM Message Service Client for .NET 2.0 版搭配使用，但對於連接至 IBM WebSphere MQ 7.0 版佇列管理程式的應用程式沒有作用，除非 Connection Factory 的 XMSC_WMQ_PROVIDER_VERSION 內容設為小於 7 的版本號碼。
<u>XMSC_WMQ_BROKER_QMGR</u>	分配管理系統所連接佇列管理程式的名稱。 註: 此內容可以與 IBM Message Service Client for .NET 2.0 版搭配使用，但對於連接至 IBM WebSphere MQ 7.0 版佇列管理程式的應用程式沒有作用，除非 Connection Factory 的 XMSC_WMQ_PROVIDER_VERSION 內容設為小於 7 的版本號碼。
<u>XMSC_WMQ_BROKER_SUBQ</u>	不可延續訊息消費者的訂閱者佇列的名稱。 註: 此內容可以與 IBM Message Service Client for .NET 2.0 版搭配使用，但對於連接至 IBM WebSphere MQ 7.0 版佇列管理程式的應用程式沒有作用，除非 Connection Factory 的 XMSC_WMQ_PROVIDER_VERSION 內容設為小於 7 的版本號碼。

表 27: <i>ConnectionFactory</i> 的內容 (繼續)	
內容名稱	說明
<u>XMSC_WMQ_BROKER_VERSION</u>	應用程式用於連線或目的地的分配管理系統的類型。 註: 此內容可以與 IBM Message Service Client for .NET 2.0 版搭配使用，但對於連接至 IBM WebSphere MQ 7.0 版佇列管理程式的應用程式沒有作用，除非 Connection Factory 的 <u>XMSC_WMQ_PROVIDER_VERSION</u> 內容設為小於 7 的版本號碼。
第 190 頁的 <u>『XMSC_WMQ_CCDURL』</u>	識別檔案名稱和位置的統一資源定址器 (URL)，該檔案包含用戶端通道定義表，而且還指定如何存取該檔案。
<u>XMSC_WMQ_CHANNEL</u>	要用於連線的通道名稱。
第 190 頁的 <u>『XMSC_WMQ_CLIENT_RECONNECT_OPTIONS』</u>	這個內容指定這個 Factory 所建立之新連線的用戶端重新連接選項
第 191 頁的 <u>『XMSC_WMQ_CLIENT_RECONNECT_TIMEOUT』</u>	此內容指定用戶端連線嘗試重新連接的持續時間 (以秒為單位)。
<u>XMSC_WMQ_CONNECTION_MODE</u>	應用程式據以連線至佇列管理程式的模式。
第 191 頁的 <u>『XMSC_WMQ_CONNECTION_NAME_LIST』</u>	此內容指定用戶端在其連線中斷之後嘗試重新連接的主機。
<u>XMSC_WMQ_FAIL_IF QUIESCE</u>	在應用程式所連接的佇列管理程式處於靜止狀態時對某些方法的呼叫是否失敗。
<u>XMSC_WMQ_HOST_NAME</u>	佇列管理程式在其中執行的系統的主機名稱或 IP 位址。
<u>XMSC_WMQ_LOCAL_ADDRESS</u>	為了連線至佇列管理程式，此內容會指定要使用的本端網路介面和/或要使用的本端埠 (或本端埠的範圍)。
<u>XMSC_WMQ_MESSAGE_SELECTION</u>	決定訊息選擇是由 XMS 用戶端還是分配管理系統完成。 註: 此內容可以與 IBM Message Service Client for .NET 2.0 版搭配使用，但對於連接至 IBM WebSphere MQ 7.0 版佇列管理程式的應用程式沒有作用，除非 Connection Factory 的 <u>XMSC_WMQ_PROVIDER_VERSION</u> 內容設為小於 7 的版本號碼。
<u>XMSC_WMQ_MSG_BATCH_SIZE</u>	使用非同步傳訊遞送時，透過一個批次從佇列中擷取的訊息數目上限。 註: 此內容可以與 IBM Message Service Client for .NET 2.0 版搭配使用，但對於連接至 IBM WebSphere MQ 7.0 版佇列管理程式的應用程式沒有作用，除非 Connection Factory 的 <u>XMSC_WMQ_PROVIDER_VERSION</u> 內容設為小於 7 的版本號碼。
<u>XMSC_WMQ_POLLING_INTERVAL</u>	如果階段作業內每個訊息接聽器的佇列上都沒有適當的訊息，這個值便是每個訊息接聽器在重新嘗試從它的佇列取得訊息之前，所經歷的間隔上限 (毫秒)。 註: 此內容可以與 IBM Message Service Client for .NET 2.0 版搭配使用，但對於連接至 IBM WebSphere MQ 7.0 版佇列管理程式的應用程式沒有作用，除非 Connection Factory 的 <u>XMSC_WMQ_PROVIDER_VERSION</u> 內容設為小於 7 的版本號碼。

表 27: <i>ConnectionFactory</i> 的內容 (繼續)	
內容名稱	說明
第 198 頁的 <u>『XMSC_WMQ_PROVIDER_VERSION』</u>	應用程式準備連接的佇列管理程式的版本、版次、修正層次和修正套件。
<u>XMSC_WMQ_PORT</u>	佇列管理程式在其中接聽送入要求的埠的號碼。
<u>XMSC_WMQ_PUB_ACK_INTERVAL</u>	在 XMS 用戶端要求分配管理系統確認之前，發佈者已發佈的訊息數。 註: 此內容可以與 IBM Message Service Client for .NET 2.0 版搭配使用，但對於連接至 IBM WebSphere MQ 7.0 版 佇列管理程式的應用程式沒有作用，除非 Connection Factory 的 XMSC_WMQ_PROVIDER_VERSION 內容設為小於 7 的版本號碼。
第 195 頁的 <u>『XMSC_WMQ_PUT_ASYNC_ALLOWED』</u>	此內容決定是否容許訊息產生者使用非同步放置，以傳送訊息至此目的地。
<u>XMSC_WMQ_QMGR_CCSID</u>	編碼字集 (或字碼頁) 的 ID (CCSID)，在 XMS 用戶端與 WebSphere MQ 用戶端之間交換「訊息佇列介面 (MQI)」中定義的字元資料欄位。
<u>XMSC_WMQ_QUEUE_MANAGER</u>	要連接的佇列管理程式的名稱。
<u>XMSC_WMQ_RECEIVE_EXIT</u>	識別要執行的通道接收結束程式。
<u>XMSC_WMQ_RECEIVE_EXIT_INIT</u>	通道接收結束程式在被呼叫時收到的使用者資料。
<u>XMSC_WMQ_SECURITY_EXIT</u>	識別通道安全結束程式。
<u>XMSC_WMQ_SECURITY_EXIT_INIT</u>	通道安全結束程式在被呼叫時收到的使用者資料。
第 203 頁的 <u>『XMSC_WMQ_SEND_CHECK_COUNT』</u>	在單一非交易式 XMS 階段作業內檢查非同步放置錯誤之間所容許的傳送呼叫次數。
<u>XMSC_WMQ_SEND_EXIT</u>	識別通道傳送結束程式。
<u>XMSC_WMQ_SEND_EXIT_INIT</u>	傳送給所呼叫之通道傳送結束程式的使用者資料。
第 203 頁的 <u>『XMSC_WMQ_SHARE_CONV_ALLOWED』</u>	如果通道定義相符，用戶端連線是否可以與從相同處理程序至相同佇列管理程式的其他最上層 XMS 連線共用其 Socket。系統提供此內容以容許根據需要出於應用程式開發、維護或作業原因，將連線完全隔離到單獨的 Socket 中。
<u>XMSC_WMQ_SSL_CERT_STORES</u>	保留給佇列管理程式建立 SSL 連線時所使用憑證撤銷清冊 (CRL) 的伺服器位置。
<u>XMSC_WMQ_SSL_CIPHER_SPEC</u>	建立佇列管理程式的安全連線時所使用的 CipherSpec 名稱。
<u>XMSC_WMQ_SSL_CIPHER_SUITE</u>	要在與佇列管理程式的 SSL 或 TLS 連線上使用 CipherSuite 名稱。在協議安全連線時使用的通訊協定取決於指定的 CipherSuite。
<u>XMSC_WMQ_SSL_CRYPT_HW</u>	用戶端系統所連接的加密硬體的配置詳細資料。
<u>XMSC_WMQ_SSL_FIPS_REQUIRED</u>	此內容的值決定應用程式是否可以使用不符合 FIPS 標準的密碼組合。如果此內容設為 true，則只將 FIPS 演算法用於用戶端/伺服器連線。
<u>XMSC_WMQ_SSL_KEY_REPOSITORY</u>	在其中儲存金鑰和憑證的金鑰資料庫檔的位置。

表 27: <i>ConnectionFactory</i> 的內容 (繼續)	
內容名稱	說明
XMSC_WMQ_SSL_KEY_RESETCOUNT	KeyResetCount 代表在重新協議秘密金鑰之前 SSL 交談中傳送和收到的未加密位元組總數。
XMSC_WMQ_SSL_PEER_NAME	建立佇列管理程式的 SSL 連線時所使用的對等節點名稱。
XMSC_WMQ_SYNCPOINT_ALL_GETS	是否必須在同步點控制之下從佇列中擷取所有訊息。
第 209 頁的『XMSC_WMQ_TARGET_CLIENT』	
XMSC_WMQ_TEMP_Q_PREFIX	用來構成 WebSphere MQ 動態佇列 (在應用程式建立 XMS 暫時佇列時建立) 名稱的字首。
XMSC_WMQ_TEMP_TOPIC_PREFIX	建立暫時主題時，XMS 會產生格式為 "TEMP/TEMPTOPICPREFIX/unique_id" 的主題字串，或者如果此內容包含預設值，則會產生此字串 "TEMP/unique_id"。如果指定非空白值，則容許定義特定模型佇列，以給訂閱者建立根據此連線建立的暫時主題的受管理佇列。
XMSC_WMQ_TEMPORARY_MODEL	當應用程式建立 XMS 暫時佇列時，從中建立動態佇列的 WebSphere MQ 模型佇列名稱。
XMSC_WPM_BUS_NAME	對於 Connection Factory，則應用程式所連接的服務整合匯流排的名稱，或者對於目的地，則為目的地所在的服務整合匯流排的名稱。
XMSC_WPM_CONNECTION_PROXIMITY	連線的連線近似性設定。
XMSC_WPM_DUR_SUB_HOME	在其中管理連線或目的地的所有可延續訂閱的傳訊引擎的名稱。
XMSC_WPM_LOCAL_ADDRESS	為了連線至服務整合匯流排，此內容會指定要使用的本端網路介面和/或要使用的本端埠（或本端埠的範圍）。
XMSC_WPM_NON_PERSISTENT_MAP	使用連線傳送的非持續訊息的可靠性層次。
XMSC_WPM_PERSISTENT_MAP	使用連線傳送的持續訊息的可靠性層次。
XMSC_WPM_PROVIDER_ENDPOINTS	引導伺服器的一個以上端點位址的序列。
XMSC_WPM_TARGET_GROUP	傳訊引擎的目標群組的名稱。
XMSC_WPM_TARGET_SIGNIFICANCE	傳訊引擎的目標群組的重要性。
XMSC_WPM_TARGET_TRANSPORT_CHAIN	應用程式在連接傳訊引擎時必須使用的入埠傳輸鏈的名稱。
XMSC_WPM_TARGET_TYPE	傳訊引擎的目標群組的類型。
XMSC_WPM_TEMP_Q_PREFIX	當應用程式建立 XMS 暫時佇列時，用來構成服務整合匯流排中所建立之暫時佇列名稱的字首。
XMSC_WPM_TEMP_TOPIC_PREFIX	用來構成應用程式所建立暫時主題的名稱的字首。

相關概念

[ConnectionFactory](#) 和 [Connection](#) 物件

[ConnectionFactory](#) 物件提供應用程式用來建立 [Connection](#) 物件的範本。應用程式使用 [Connection](#) 物件來建立 [Session](#) 物件。

[WebSphere 服務整合匯流排的連線](#)

XMS 應用程式可以使用直接 TCP/IP 連線或使用透過 TCP/IP 的 HTTP 來連接至 WebSphere 服務 Integration Bus。

[WebSphere MQ 佇列管理程式的安全連線](#)

如果要讓 XMS .NET 應用程式能夠建立與 WebSphere MQ 佇列管理程式的安全連線，必須在 ConnectionFactory 物件中定義相關內容。

WebSphere 服務 Integration Bus 傳訊引擎的安全連線

啟用 XMS 應用程式若要建立與 WebSphere 服務 Integration Bus 傳訊引擎的安全連線，必須在 ConnectionFactory 物件中定義相關內容。

受管理物件的內容對映

如果要讓應用程式能夠使用 WebSphere MQ JMS 和 WebSphere Application Server Connection Factory 及目的地物件定義，從這些定義擷取的內容必須對映至可以在 XMS Connection Factory 和目的地上設定的對應 XMS 內容。

相關工作

建立受管理物件

必須使用適當的管理工具來建立 XMS 應用程式建立傳訊伺服器連線所需的 ConnectionFactory 和「目的地」物件定義。

相關參考

受管理 ConnectionFactory 物件的必要內容

當應用程式建立 Connection Factory 時，必須定義一些內容來建立傳訊伺服器的連線。

ConnectionMeta 資料的內容

ConnectionMeta 資料物件內容的概觀，以及指向更詳細參照資訊的鏈結。

內容名稱	說明
XMSC_JMS_MAJOR_VERSION	作為 XMS 基礎之 JMS 規格的主要版本號碼。這是唯讀內容。
XMSC_JMS_MINOR_VERSION	XMS 所根據之 JMS 規格的次要版本號碼。這是唯讀內容。
XMSC_JMS_VERSION	XMS 所根據之 JMS 規格的版本 ID。這是唯讀內容。
XMSC_MAJOR_VERSION	XMS 用戶端的版本號碼。這是唯讀內容。
XMSC_MINOR_VERSION	XMS 用戶端的版次號碼。這是唯讀內容。
XMSC_PROVIDER_NAME	XMS 用戶端的提供者。這是唯讀內容。
XMSC_VERSION	XMS 用戶端的版本 ID。這是唯讀內容。

目的地的內容

「目的地」物件的內容概觀，以及指向更詳細參照資訊的鏈結。

內容名稱	說明
XMSC_DELIVERY_MODE	傳送至目的地的訊息遞送模式。
XMSC_PRIORITY	傳送至目的地的訊息的優先順序。
XMSC_RTT_MULTICAST	Connection Factory 或目的地的多重播送設定。
XMSC_TIME_TO_LIVE	傳送至目的地的訊息的存活時間。
XMSC_WMQ_BROKER_VERSION	應用程式用於連線或目的地的分配管理系統的類型。
XMSC_WMQ_CCSID	當 XMS 用戶端將訊息轉遞至目的地時，訊息內文中的字元資料字串所使用的編碼字集 (或字碼頁) ID (CCSID)。

表 29: 目的地的內容 (繼續)	
內容名稱	說明
<u>XMSC_WMQ_DUR_SUBQ</u>	從目的地接收訊息的可延續訂閱者的訂閱者佇列名稱。 註: 此內容可以與 IBM Message Service Client for .NET 2.0 版搭配使用, 但對於連接至 IBM WebSphere MQ 7.0 版佇列管理程式的應用程式沒有作用, 除非 Connection Factory 的 XMSC_WMQ_PROVIDER_VERSION 內容設為小於 7 的版本號碼。
<u>XMSC_WMQ_ENCODING</u>	當 XMS 用戶端將訊息轉遞至目的地時, 訊息內文中的數值資料如何呈現。
<u>XMSC_WMQ_FAIL_IF_QUIESCE</u>	在應用程式所連接的佇列管理程式處於靜止狀態時對某些方法的呼叫是否失敗。
第 193 頁的『 <u>XMSC_WMQ_MESSAGE_BODY</u> 』	此內容決定 XMS 應用程式是否將 IBM WebSphere MQ 訊息的 MQRFH2 作為訊息有效負載的一部分 (亦即, 作為訊息內文的一部分) 處理。
第 194 頁的『 <u>XMSC_WMQ_MQMD_MESSAGE_CONTEXT</u> 』	決定 XMS 應用程式要設定的訊息環境定義層次。應用程式必須以適當的環境定義權限執行, 才能使此內容生效。
第 194 頁的『 <u>XMSC_WMQ_MQMD_READ_ENABLED</u> 』	此內容決定 XMS 應用程式是否可以擷取 MQMD 欄位的值。
第 195 頁的『 <u>XMSC_WMQ_MQMD_WRITE_ENABLED</u> 』	此內容決定 XMS 應用程式是否可以使用 MQMD 欄位的值。
第 195 頁的『 <u>XMSC_WMQ_READ_Ahead_ALLOWED</u> 』	此內容決定是否容許訊息消費者和佇列瀏覽器在接收訊息之前使用先讀, 將這個目的地的非持續性、非交易性訊息讀到內部緩衝區。
第 196 頁的『 <u>XMSC_WMQ_READ_AHEAD_CLOSE_policy</u> 』	對於將遞送至非同步訊息接聽器的訊息, 此內容決定在消息消費者關閉時內部先讀緩衝區中的訊息發生的情況。
第 200 頁的『 <u>XMSC_WMQ_RECEIVE_CCsid</u> 』	設定佇列管理程式訊息轉換的目標 CCSID 的目的地內容。除非 XMSC_WMQ_RECEIVE_CONVERSION 設為 WMQ_RECEIVE_CONVERSION_QMGR, 否則會忽略此值。
第 200 頁的『 <u>XMSC_WMQ_RECEIVE_CONVERSION</u> 』	決定佇列管理程式是否要執行資料轉換的目的地內容。
<u>XMSC_WMQ_TARGET_CLIENT</u>	傳送至目的地的訊息是否包含 MQRFH2 標頭。
<u>XMSC_WMQ_TEMP_TOPIC_PREFIX</u>	建立暫時主題時, XMS 會產生格式為 "TEMP/TEMPTOPICPREFIX/unique_id" 的主題字串, 或者如果此內容包含預設值, 則會產生此字串 "TEMP/unique_id"。如果指定非空白值, 則容許定義特定模型佇列, 以給訂閱者建立根據此連線建立的暫時主題的受管理佇列。
<u>XMSC_WPM_BUS_NAME</u>	對於 Connection Factory, 則應用程式所連接的服務整合匯流排的名稱, 或者對於目的地, 則為目的地所在的服務整合匯流排的名稱。
<u>XMSC_WPM_TOPIC_SPACE</u>	包含主題的主題空間名稱。

相關概念

ConnectionFactories 和 Connection 物件

ConnectionFactory 物件提供應用程式用來建立 Connection 物件的範本。應用程式使用 Connection 物件來建立 Session 物件。

WebSphere 服務整合匯流排的連線

XMS 應用程式可以使用直接 TCP/IP 連線或使用透過 TCP/IP 的 HTTP 來連接至 WebSphere 服務 Integration Bus。

目的地

XMS 應用程式使用「目的地」物件來指定所傳送訊息的目的地，以及所接收訊息的來源。

目的地萬用字元

XMS 提供對目的地萬用字元的支援，確保萬用字元可以傳遞至比對所需的位置。XMS 可以使用的每一種伺服器類型都有不同的萬用字元架構。

主題統一資源識別碼

主題統一資源識別碼 (URI) 指定主題的名稱；它也可以為它指定一或多個內容。

佇列統一資源識別碼

佇列的 URI 指定佇列的名稱；它也可以指定佇列的一或多個內容。

暫時目的地

XMS 應用程式可以建立及使用暫時目的地。

受管理物件的內容對映

如果要讓應用程式能夠使用 WebSphere MQ JMS 和 WebSphere Application Server Connection Factory 及目的地物件定義，從這些定義擷取的內容必須對映至可以在 XMS Connection Factory 和目的地上設定的對應 XMS 內容。

相關工作

建立受管理物件

必須使用適當的管理工具來建立 XMS 應用程式建立傳訊伺服器連線所需的 ConnectionFactory 和「目的地」物件定義。

相關參考

受管理目的地物件的必要內容

建立目的地的應用程式必須在受管理的目的地物件上設定應用程式的數個內容。

InitialContext 的內容

InitialContext 物件內容的概觀，以及更詳細參照資訊的鏈結。

內容名稱	說明
<u>XMSC_IC_PROVIDER_URL</u>	用來找出 JNDI 命名目錄，這樣 COS 命名服務就不需要與 Web 服務位在同一伺服器上。
<u>XMSC_IC_SECURITY_AUTHENTICATION</u>	基於 Java 環境定義介面 SECURITY_AUTHENTICATION。此內容僅適用於 COS 命名環境定義。
<u>XMSC_IC_SECURITY_CREDENTIALS</u>	根據 Java 環境定義介面 SECURITY_CREDENTIALS。此內容僅適用於 COS 命名環境定義。
<u>XMSC_IC_SECURITY_PRINCIPAL</u>	基於 Java 環境定義介面 SECURITY_PRINCIPAL。此內容僅適用於 COS 命名環境定義。
<u>XMSC_IC_SECURITY_PROTOCOL</u>	基於 Java 環境定義介面 SECURITY_PROTOCOL 此內容僅適用於 COS 命名環境定義。
<u>XMSC_IC_URL</u>	對於 LDAP 和 FileSystem 環境定義，則為管理物件所在儲存庫的位址。對於 COS 命名環境定義，這是查閱目錄中物件的 Web 服務位址。

相關概念

InitialContext 內容

InitialContext 建構子的參數包括受管理物件的儲存庫位置，以統一資源指示器 (URI) 形式提供。為了讓應用程式建立與儲存庫的連線，可能需要提供比 URI 中包含的資訊更多的資訊。

XMS 起始環境定義的 URI 格式

受管理物件的儲存庫位置以統一資源指示器 (URI) 提供。URI 的格式視環境定義類型而定。

擷取受管理物件

XMS 會使用建立 InitialContext 物件時所提供的位址，或在 InitialContext 內容中，從儲存庫擷取受管理物件。

相關工作

InitialContext 物件

應用程式必須建立起始環境定義，以用來建立受管理物件儲存庫的連線，以擷取必要的受管理物件。

訊息的內容

Message 物件內容的概觀，以及指向更詳細參照資訊的鏈結。

內容名稱	說明
JMS_IBM_CHARACTER_SET	當 XMS 用戶端將訊息轉遞至其預期目的地時，訊息內文中的字元資料字串所使用的編碼字集 (或字碼頁) ID (CCSID)。在 XMS 中，此內容具有數值並對映至 CCSID。但是，此內容以 JMS 內容為基礎，因此具有字串類型值且對映至代表此數值 CCSID 的 Java 字集。
JMS_IBM_Encoding	當 XMS 用戶端將訊息轉遞至其預期目的地時，訊息內文中的數值資料如何呈現。
JMS_IBM_EXCEPTIONMESSAGE	用來說明訊息傳送至異常目的地的原因的文字。這是唯讀內容。
JMS_IBM_EXCEPTIONPROBLEMDESTINATION	訊息傳送至異常目的地之前訊息所在目的地的名稱。
JMS_IBM_EXCEPTIONREASON	用來說明訊息傳送至異常目的地的原因的原因碼。
JMS_IBM_EXCEPTIONTIMESTAMP	訊息傳送至異常目的地的時間。
JMS_IBM_FEEDBACK	指出報告訊息本質的代碼。
JMS_IBM_FORMAT	訊息中應用程式資料的本質。
JMS_IBM_LAST_MSG_IN_GROUP	指出訊息是否為訊息群組中的最後一則訊息。
JMS_IBM_MSGTYPE	訊息的類型。
JMS_IBM_PUTAPPLTYPE	傳送訊息的應用程式類型。
JMS_IBM_PUTDATE	傳送訊息的日期。
JMS_IBM_PUTTIME	傳送訊息的時間。
JMS_IBM_REPORT_COA	要求「到達時確認」報告訊息，指定必須將原始訊息中的多少應用程式資料併入到報告訊息中。
JMS_IBM_REPORT_COD	要求「遞送時確認」報告訊息，指定必須將原始訊息中的多少應用程式資料併入到報告訊息中。
JMS_IBM_REPORT_DISCARD_MSG	要求在訊息無法遞送至其預期目的地的情況下捨棄訊息。
JMS_IBM_REPORT_EXCEPTION	要求異常狀況報告訊息，指定必須將原始訊息中的多少應用程式資料併入到報告訊息中。
JMS_IBM_REPORT_EXPIRATION	要求有效期限報告訊息，指定必須將原始訊息中的多少應用程式資料併入到報告訊息中。
JMS_IBM_REPORT_NAN	要求負面動作通知報告訊息。

表 31: 訊息的內容 (繼續)	
內容名稱	說明
<u>JMS_IBM_REPORT_PAN</u>	要求正面動作通知報告訊息。
<u>JMS_IBM_REPORT_PASS_CORREL_ID</u>	要求任何報告或回覆訊息的相關性 ID 與原始訊息的相關性 ID 相同。
<u>JMS_IBM_REPORT_PASS_MSG_ID</u>	要求任何報告或回覆訊息的訊息 ID 與原始訊息的訊息 ID 相同。
<u>JMS_IBM_RETAIN</u>	設定此內容會向佇列管理程式指出如何將訊息作為保留的發佈資訊加以處理。
<u>JMS_IBM_SYSTEM_MESSAGEID</u>	在服務整合匯流排中唯一識別訊息的 ID。這是唯讀內容。
<u>JMSX_APPID</u>	傳送訊息的應用程式名稱。
<u>JMSX_DELIVERY_COUNT</u>	嘗試遞送訊息的次數。
<u>JMSX_GROUPID</u>	訊息所屬的訊息群組的 ID。
<u>JMSX_GROUPSEQ</u>	訊息群組內訊息的序號。
<u>JMSX_USERID</u>	與傳送訊息的應用程式相關聯的使用者 ID。

JMS_IBM_MQMD* 內容

IBM Message Service Client for .NET 可讓用戶端應用程式使用 API 讀取/寫入 MQMD 欄位。它也容許存取 MQ 訊息資料。依預設會停用 MQMD 的存取權，且應用程式必須使用「目的地」內容 XMSC_WMQ_MQMD_WRITE_ENABLED 及 XMSC_WMQ_MQMD_READ_ENABLED 來明確啟用。這兩個內容彼此獨立。

除了 StrucId 和 Version 之外，所有 MQMD 欄位都公開為其他訊息物件內容，並以 JMS_IBM_MQMD 為字首。

JMS_IBM_MQMD* 內容優先於上表所說明的其他內容 (例如 JMS_IBM*)。

傳送訊息

除了 StrucId 和 Version 之外的所有 MQMD 欄位都已呈現。這些內容僅參照 MQMD 欄位; 如果內容同時出現在 MQMD 和 MQRFH2 標頭中，則不會設定或擷取 MQRFH2 中的版本。可以設定任何這些內容，但 JMS_IBM_MQMD_BackoutCount 除外。會忽略為 JMS_IBM_MQMD_BackoutCount 設定的任何值。

如果內容具有長度上限，且您提供的值太長，則會截斷該值。

對於某些內容，您也必須在 Destination 物件上設定 XMSC_WMQ_MQMD_MESSAGE_CONTEXT 內容。應用程式必須以適當的環境定義權限執行，才能使此內容生效。如果您未將 XMSC_WMQ_MQMD_MESSAGE_CONTEXT 設為適當的值，則會忽略內容值。如果您將 XMSC_WMQ_MQMD_MESSAGE_CONTEXT 設為適當的值，但沒有佇列管理程式的足夠環境定義權限，則會發出異常狀況。需要特定 XMSC_WMQ_MQMD_MESSAGE_CONTEXT 值的內容如下所示。

下列內容需要將 XMSC_WMQ_MQMD_MESSAGE_CONTEXT 設為 XMSC_WMQ_MDCTX_SET_IDENTITY_CONTEXT 或 XMSC_WMQ_MDCTX_SET_ALL_CONTEXT:

- JMS_IBM_MQMD_UserIdentifier
- JMS_IBM_MQMD_AccountingToken
- JMS_IBM_MQMD_ApplIdentity 資料

下列內容需要將 XMSC_WMQ_MQMD_MESSAGE_CONTEXT 設為 XMSC_WMQ_MDCTX_SET_ALL_CONTEXT:

- JMS_IBM_MQMD_PutAppl 類型
- JMS_IBM_MQMD_PutAppl 名稱

- JMS_IBM_MQMD_PutDate
- JMS_IBM_MQMD_PutTime
- JMS_IBM_MQMD_ApplOrigin 資料

接收訊息

如果 XMSC_WMQ_MQMD_READ_ENABLED 內容設為 true，則不論產生端應用程式所設定的實際內容為何，都可以在收到的訊息上使用所有這些內容。除非先根據 JMS 規格清除所有內容，否則應用程式無法修改所接收訊息的內容。可以在不修改內容的情況下轉遞接收到的訊息。

註: 如果應用程式從 XMSC_WMQ_MQMD_READ_ENABLED 內容設為 true 的目的地接收訊息，並將它轉遞至 XMSC_WMQ_MQMD_WRITE_ENABLED 設為 true 的目的地，這會導致將所接收訊息的所有 MQMD 欄位值複製到轉遞的訊息。內容表

表 32: 代表 MQMD 欄位之訊息物件的內容		
內容	說明	類型
JMS_IBM_MQMD_REPORT	報告訊息的選項	System.Int32
JMS_IBM_MQMD_MSGTYPE	訊息類型	System.Int32
JMS_IBM_MQMD_EXPIRY	訊息期限	System.Int32
JMS_IBM_MQMD_FEEDBACK	回饋碼或原因碼	System.Int32
JMS_IBM_MQMD_ENCODING	訊息資料的數字編碼	System.Int32
JMS_IBM_MQMD_CODEDCHARSETID	訊息資料的字集 ID	System.Int32
JMS_IBM_MQMD_FORMAT	訊息資料的格式名稱	System.String
JMS_IBM_MQMD_PRIORITY 註: 如果您指派值給不在 0-9 範圍內的 JMS_IBM_MQMD_PRIORITY，此值會違反 JMS 規格。	訊息優先順序	System.Int32
JMS_IBM_MQMD_PERSISTENCE	訊息持續性	System.Int32
JMS_IBM_MQMD_MSGID 註: JMS 規格指出訊息 ID 必須由 JMS 提供者設定，且必須是唯一或空值。如果您指派值給 JMS_IBM_MQMD_MSGID，此值會複製到 JMSMessageID。因此，它不是由 JMS 提供者所設定，且可能不是唯一的: 此值違反 JMS 規格。	訊息 ID	位元組陣列 註: 在訊息上使用位元組陣列內容違反 JMS 規格。
JMS_IBM_MQMD_CORRELID 註: 如果您將值指派給以字串 'ID:' 開頭的 JMS_IBM_MQMD_CORRELID，則此值會違反 JMS 規格。	相關性 ID	位元組陣列 註: 在訊息上使用位元組陣列內容違反 JMS 規格。
JMS_IBM_MQMD_BACKOUTCOUNT	取消計數器	System.Int32
JMS_IBM_MQMD_REPLYTOQ	回覆佇列的名稱	System.String
JMS_IBM_MQMD_REPLYTOQMGR	回覆佇列管理程式的名稱	System.String
JMS_IBM_MQMD_USERIDENTIFIER	使用者 ID	System.String

表 32: 代表 MQMD 欄位之訊息物件的內容 (繼續)		
內容	說明	類型
JMS_IBM_MQMD_ACCOUNTINGtoken	帳戶記號	位元組陣列 註: 在訊息上使用位元組陣列內容違反 JMS 規格。
JMS_IBM_MQMD_APPLIDENTITYDATA	與身分相關的應用程式資料	System.String
JMS_IBM_MQMD_PUTAPPLTYPE	放置訊息的應用程式類型	System.Int32
JMS_IBM_MQMD_PUTAPPLNAME	放置訊息的應用程式名稱	System.String
JMS_IBM_MQMD_PUTDATE	放置訊息的日期	System.String
JMS_IBM_MQMD_PUTTIME	放置訊息的時間	System.String
JMS_IBM_MQMD_APPLORIGINDATA	與出處相關的應用程式資料	System.String
JMS_IBM_MQMD_GROUPID	群組 ID	位元組陣列 註: 在訊息上使用位元組陣列內容違反 JMS 規格。
JMS_IBM_MQMD_MSGSEQNUMBER	群組內本端訊息的序號	System.Int32
JMS_IBM_MQMD_OFFSET	從邏輯訊息開始的實體訊息中的資料偏移	System.Int32
JMS_IBM_MQMD_MSGFLAGS	訊息旗標	System.Int32
JMS_IBM_MQMD_ORIGINALLENGTH	原始訊息的長度	System.Int32

如需進一步詳細資料，請參閱 [MQMD](#)。

範例

此範例會導致將訊息放入具有 MQMD.UserIdentifer 設為 "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
// ...
```

在設定 JMS_IBM_MQMD_USERIDENTIFIER 之前，必須先設定 XMSC_WMQ_MQMD_MESSAGE_CONTEXT。如需使用 XMSC_WMQ_MQMD_MESSAGE_CONTEXT 的相關資訊，請參閱訊息物件內容。

同樣地，您可以在接收訊息之前將 XMSC_WMQ_MQMD_READ_ENABLED 設為 true，然後使用訊息的 get 方法 (例如 getStringProperty)，來解壓縮 MQMD 欄位的內容。任何收到的內容都是唯讀的。

此範例會導致值欄位保留 MQMD.ApplIdentityData 欄位。

```
// Create a ConnectionFactory, connection, session, consumer
// ...

// Create a destination
// ...

// Enable MQMD read
dest.setBooleanProperty(XMSC_WMQ_MQMD_READ_ENABLED, XMSC_WMQ_MQMD_READ_ENABLED_YES);

// Receive a message
// ...

// Get desired MQMD field value using a property
System.String value = rcvMsg.getStringProperty(JMS_IBM_MQMD_APPLIDENTITYDATA);
```

MessageConsumer 的內容

MessageConsumer 物件內容的概觀，以及指向更詳細參照資訊的鏈結。

內容名稱	說明
XMSC_IS_SUBSCRIPTION_MULTICAST	指出是否使用 WebSphere MQ 多重播送傳輸將訊息遞送至訊息消費者。這是唯讀內容。
XMSC_IS_SUBSCRIPTION_RELIABLE_MULTICAST	指出是否使用具有可靠服務品質的 WebSphere MQ 多重播送傳輸將訊息遞送至訊息消費者。這是唯讀內容。

請參閱 [IMessageConsumer](#) 的 NET 內容，以取得詳細資料。

MessageProducer 的內容

MessageProducer 物件內容概觀，以及指向更詳細參照資訊的鏈結。

請參閱 [IMessageProducer](#) 的 NET 內容，以取得詳細資料。

階段作業的內容

「階段作業」物件的內容概觀，以及指向更詳細參照資訊的鏈結。

請參閱 [ISession](#) 的 NET 內容，以取得詳細資料。

內容定義

此區段提供每一個物件內容的定義。

每一個內容定義都包含下列資訊：

- 內容的資料類型
- 具有內容的物件類型
- 對於 Destination 的內容，可以在統一資源識別碼 (URI) 中使用的名稱
- 內容的更詳細說明
- 內容的有效值
- 內容的預設值

名稱以下列其中一個字首開始的內容僅與指定的連線類型相關：

XMSC_RTT

這些內容僅適用於與分配管理系統的即時連線。內容的名稱在標頭檔 `xmsc_rtt.h` 中定義為具名常數。

XMSC_WMQ

僅當應用程式連接至 WebSphere MQ 佇列管理程式時，這些內容才相關。內容的名稱在標頭檔 `xmsc_wmq.h` 中定義為具名常數。

XMSC_WPM

只有在應用程式連接至 WebSphere 服務整合匯流排時，這些內容才相關。內容的名稱在標頭檔 `xmsc_wpm.h` 中定義為具名常數。

除非在其定義中另有說明，否則其餘內容會與所有連線類型相關。內容的名稱在標頭檔 `xmsc.h` 中定義為具名常數。名稱以字首 `JMSX` 開頭的內容是 JMS 定義的訊息內容，名稱以字首 `JMS_IBM` 開頭的內容是 IBM 定義的訊息內容。如需訊息內容的相關資訊，請參閱第 60 頁的『XMS 訊息的內容』。

除非在其定義中另有說明，否則每一個內容都在點對點 (point-to-point) 和發佈/訂閱網域中相關。

除非內容指定為唯讀，否則應用程式可以取得並設定任何內容的值。

JMS_IBM_CHARACTER_SET

資料類型：

`System.Int32`

下列項目的內容：

訊息

當 XMS 用戶端將訊息轉遞至其預期目的地時，訊息內文中的字元資料字串所使用的編碼字集 (或字碼頁) ID (CCSID)。在 XMS 中，此內容具有數值並對映至 CCSID。但是，此內容以 JMS 內容為基礎，因此具有字串類型值且對映至代表此數值 CCSID 的 Java 字集。此內容會置換 `XMSC_WMQ_CCSID` 內容為目的地指定的任何 CCSID。

依預設，不會設定內容。

當應用程式連接至服務整合匯流排時，這個內容不相關。

JMS_IBM_Encoding

資料類型：

`System.Int32`

下列項目的內容：

訊息

當 XMS 用戶端將訊息轉遞至其預期目的地時，訊息內文中的數值資料如何呈現。此內容會置換 `XMSC_WMQ_ENCODING` 內容為目的地指定的任何編碼。此內容指定二進位整數、聚集十進位整數及浮點數字的表示法。

內容的有效值與可以在訊息描述子的 **Encoding** 欄位中指定的值相同。

應用程式可以使用下列具名常數來設定內容：

已命名的常數	意義
<code>MQENC_INTEGER_NORMAL</code>	正常整數編碼
已反轉 <code>MQENC_INTEGER_REVERSED</code>	反向整數編碼
<code>MQENC_DECIMAL_NORMAL</code>	一般聚集十進位編碼
<code>MQENC_DECIMAL_REVERED</code>	反向壓縮十進位編碼
<code>MQENC_FLOAT_IEE_NORMAL</code>	一般 IEEE 浮點數編碼
<code>MQENC_FLOAT_IEE_REVERSED</code>	反向 IEEE 浮點數編碼
<code>MQENC_FLOAT_S390</code>	z/OS 架構浮點數編碼
<code>MQENC_NATIVE</code>	原生機器編碼

若要形成內容的值，應用程式可以新增下列三個常數，如下所示：

- 名稱以 `MQENC_INTEGER` 開始的常數，用來指定二進位整數的表示法

- 名稱以 MQENC_DECIMAL 開始的常數，用來指定聚集十進位整數的表示法
 - 名稱以 MQENC_FLOAT 開始的常數，用來指定浮點數字的表示法
- 或者，應用程式可以將內容設為 MQENC_NATIVE，其值與環境相關。
依預設，不會設定內容。
當應用程式連接至服務整合匯流排時，這個內容不相關。

JMS_IBM_EXCEPTIONMESSAGE

資料類型：

字串

下列項目的內容：

訊息

用來說明訊息傳送至異常目的地的原因的文字。這是唯讀內容。

只有在應用程式連接至服務整合匯流排，並從異常狀況目的地接收訊息時，這個內容才相關。

JMS_IBM_EXCEPTIONPROBLEMDESTINATION

資料類型：

字串

下列項目的內容：

訊息

訊息傳送至異常目的地之前訊息所在目的地的名稱。

只有在應用程式連接至服務整合匯流排，並從異常狀況目的地接收訊息時，這個內容才相關。

JMS_IBM_EXCEPTIONREASON

資料類型：

System.Int32

下列項目的內容：

訊息

用來說明訊息傳送至異常目的地的原因的原因碼。

只有在應用程式連接至服務整合匯流排，並從異常狀況目的地接收訊息時，這個內容才相關。

JMS_IBM_EXCEPTIONTIMESTAMP

資料類型：

System.Int64

下列項目的內容：

訊息

訊息傳送至異常目的地的時間。

時間以 1970 年 1 月 1 日 00:00:00 GMT 開始的毫秒數表示。

只有在應用程式連接至服務整合匯流排，並從異常狀況目的地接收訊息時，這個內容才相關。

JMS_IBM_FEEDBACK

資料類型：

System.Int32

下列項目的內容：

訊息

指出報告訊息本質的代碼。

內容的有效值是在於訊息描述子的 **Feedback** 欄位中指定的回饋碼及原因碼。

依預設，不會設定內容。

JMS_IBM_FORMAT

資料類型：

字串

下列項目的內容：

訊息

訊息中應用程式資料的本質。

內容的有效值與可以在訊息描述子的 **Format** 欄位中指定的值相同。

依預設，不會設定內容。

當應用程式連接至服務整合匯流排時，這個內容不相關。

JMS_IBM_LAST_MSG_IN_GROUP

資料類型：

System.Boolean

下列項目的內容：

訊息

指出訊息是否為訊息群組中的最後一則訊息。

如果訊息是訊息群組中的最後一則訊息，請將內容設為 **true**。否則，請將內容設為 **false**，或不要設定內容。依預設，不會設定內容。

true 值對應於狀態旗標 **MQMF_LAST_MSG_IN_GROUP**，可以在訊息描述子的 **MsgFlags** 欄位中指定。

此內容在 發佈/訂閱 網域中被忽略，且在應用程式連接至服務整合匯流排時不相關。

JMS_IBM_MSGTYPE

資料類型：

System.Int32

下列項目的內容：

訊息

訊息的類型。

內容的有效值如下：

有效值	意義
MQMT_DATAGRAM	訊息是不需要回覆的訊息。
MQMT_REQUEST	訊息是需要回覆的訊息。
MQMT_REPLY	訊息是回覆訊息。
MQMT_REPORT	訊息是報告訊息。

這些值對應於可以在訊息描述子的 **MsgType** 欄位中指定的訊息類型。

依預設，不會設定內容。

當應用程式連接至服務整合匯流排時，這個內容不相關。

JMS_IBM_PUTAPPLTYPE

資料類型：

System.Int32

下列項目的內容：

訊息

傳送訊息的應用程式類型。

內容的有效值是在可以在訊息描述子的 **PutApplType** 欄位中指定的應用程式類型。

依預設，不會設定內容。

當應用程式連接至服務整合匯流排時，這個內容不相關。

JMS_IBM_PUTDATE

資料類型：

字串

下列項目的內容：

訊息

傳送訊息的日期。

內容的有效值與可以在訊息描述子的 **PutDate** 欄位中指定的值相同。

依預設，不會設定內容。

當應用程式連接至服務整合匯流排時，這個內容不相關。

JMS_IBM_PUTTIME

資料類型：

字串

下列項目的內容：

訊息

傳送訊息的時間。

內容的有效值與可以在訊息描述子的 **PutTime** 欄位中指定的值相同。

依預設，不會設定內容。

當應用程式連接至服務整合匯流排時，這個內容不相關。

JMS_IBM_REPORT_COA

資料類型：

System.Int32

下列項目的內容：

訊息

要求「到達時確認」報告訊息，指定必須將原始訊息中的多少應用程式資料併入到報告訊息中。

內容的有效值如下：

有效值	意義
MQRO_COA	要求「到達時確認」報告訊息，報告訊息中未包含原始訊息中的應用程式資料。
MQRO_COA_WITH_DATA	要求「到達時確認」報告訊息，原始訊息中的前 100 個位元組應用程式資料包括在報告訊息中。
MQ Ro_COA_WITH_FULL_DATA	要求「到達時確認」報告訊息，原始訊息中的所有應用程式資料都包含在報告訊息中。

這些值對應於可以在訊息描述子的 **Report** 欄位中指定的報告選項。如需這些選項的相關資訊，請參閱 [報告 \(MQLONG\)](#)。

依預設，不會設定內容。

JMS_IBM_REPORT_COD

資料類型:

System.Int32

下列項目的內容:

訊息

要求「遞送時確認」報告訊息，指定必須將原始訊息中的多少應用程式資料併入到報告訊息中。

內容的有效值如下:

有效值	意義
MQRO_COD	要求「遞送時確認」報告訊息，報告訊息中未包含原始訊息中的應用程式資料。
MQRO_COD_WITH_DATA	要求「遞送時確認」報告訊息，報告訊息中包含原始訊息中應用程式資料的前 100 個位元組。
MQRO_COD_WITH_FULL_DATA	要求「遞送時確認」報告訊息，且原始訊息中的所有應用程式資料都包含在報告訊息中。

這些值對應於可以在訊息描述子的 **Report** 欄位中指定的報告選項。

依預設，不會設定內容。

JMS_IBM_REPORT_DISCARD_MSG

資料類型:

System.Int32

下列項目的內容:

訊息

要求在訊息無法遞送至其預期目的地的情況下捨棄訊息。

將此內容設為 MQRO_DISCARD_MSG，以要求捨棄無法遞送至其預期目的地的訊息。如果您需要將訊息放置在無法傳送的郵件佇列中，或將訊息傳送至異常狀況目的地，請不要設定內容。依預設，不會設定內容。

值 MQRO_DISCARD_MSG 對應於可在訊息描述子的 **Report** 欄位中指定的報告選項。

JMS_IBM_REPORT_EXCEPTION

資料類型:

System.Int32

下列項目的內容:

訊息

要求異常狀況報告訊息，指定必須將原始訊息中的多少應用程式資料併入到報告訊息中。

內容的有效值如下:

有效值	意義
MQRO_Exception	要求異常狀況報告訊息，報告訊息中未包含原始訊息中的應用程式資料。
MQRO_EXCEPTION_WITH_DATA	要求異常狀況報告訊息，報告訊息中包含原始訊息中應用程式資料的前 100 個位元組。
MQRO_EXCEPTION_WITH_FULL_DATA	要求異常狀況報告訊息，原始訊息中的所有應用程式資料都包含在報告訊息中。

這些值對應於可以在訊息描述子的 **Report** 欄位中指定的報告選項。

依預設，不會設定內容。

JMS_IBM_REPORT_EXPIRATION

資料類型:

System.Int32

下列項目的內容:

訊息

要求有效期限報告訊息，指定必須將原始訊息中的多少應用程式資料併入到報告訊息中。

內容的有效值如下:

有效值	意義
MQRO_EXPIRATION	要求到期報告訊息，報告訊息中未包含原始訊息中的應用程式資料。
MQRO_EXPIRATION_WITH_DATA	要求到期報告訊息，報告訊息中包含原始訊息中應用程式資料的前 100 個位元組。
MQRO_EXPIRATION_WITH_FULL_DATA	要求有效期限報告訊息，原始訊息中的所有應用程式資料都包含在報告訊息中。

這些值對應於可以在訊息描述子的 **Report** 欄位中指定的報告選項。

依預設，不會設定內容。

JMS_IBM_REPORT_NAN

資料類型:

System.Int32

下列項目的內容:

訊息

要求負面動作通知報告訊息。

將內容設為 MQRO_NAN，以要求負面動作通知報告訊息。如果您不需要負面動作通知報告訊息，請不要設定內容。依預設，不會設定內容。

MQRO_NAN 值對應於可在訊息描述子的 **Report** 欄位中指定的報告選項。

JMS_IBM_REPORT_PAN

資料類型:

System.Int32

下列項目的內容:

訊息

要求正面動作通知報告訊息。

將內容設為 MQRO_PAN，以要求正向動作通知報告訊息。如果您不需要正面動作通知報告訊息，請不要設定內容。依預設，不會設定內容。

MQRO_PAN 值對應於可在訊息描述子的 **Report** 欄位中指定的報告選項。

JMS_IBM_REPORT_PASS_CORREL_ID

資料類型:

System.Int32

下列項目的內容:

訊息

要求任何報告或回覆訊息的相關性 ID 與原始訊息的相關性 ID 相同。

內容的有效值如下:

有效值	意義
MQRO_PASS_CORREL_ID	要求任何報告或回覆訊息的相關性 ID 與原始訊息的相關性 ID 相同。
MQRO_COPY_MSG_ID_TO_CORREL_ID	要求任何報告或回覆訊息的相關性 ID 與原始訊息的訊息 ID 相同。

這些值對應於可在訊息描述子的 **Report** 欄位中指定的報告選項。

內容的預設值為 MQRO_COPY_MSG_ID_TO_CORREL_ID。

JMS_IBM_REPORT_PASS_MSG_ID

資料類型：

System.Int32

下列項目的內容：

訊息

要求任何報告或回覆訊息的訊息 ID 與原始訊息的訊息 ID 相同。

內容的有效值如下：

有效值	意義
MQRO_PASS_MSG_ID	要求任何報告或回覆訊息的訊息 ID 與原始訊息的訊息 ID 相同。
MQRO_NEW_MSG_ID	要求為每一個報告或回覆訊息產生新的訊息 ID。

這些值對應於可在訊息描述子的 **報告** 欄位中指定的報告選項。

此內容的預設值是 MQRO_NEW_MSG_ID。

JMS_IBM_RETAIN

資料類型：

System.Int32

下列項目的內容：

訊息

設定此內容會向佇列管理程式指出如何將訊息作為保留的發佈資訊加以處理。當訂閱者接收來自主題的訊息時，除了舊版中所收到的訊息之外，它可能會在訂閱之後立即收到其他訊息。這些訊息是所訂閱主題的選用保留發佈資訊。對於每一個符合訂閱的主題，如果有保留的發佈資訊，則發佈資訊可供遞送給訂閱訊息消費者。

RETAIN_PUBLICATION 是此內容的唯一有效值。依預設，未設定此內容。

註：此內容僅在發佈/訂閱網域中相關

JMS_IBM_SYSTEM_MESSAGEID

資料類型：

字串

下列項目的內容：

訊息

在服務整合匯流排中唯一識別訊息的 ID。這是唯讀內容。

只有在應用程式連接至服務整合匯流排時，這個內容才相關。

JMSX_APPID

資料類型：

字串

下列項目的內容:

訊息

傳送訊息的應用程式名稱。

此內容是具有 JMS 名稱 JMSXAppID 的 JMS 定義內容。如需內容的相關資訊，請參閱 *Java 訊息服務規格 1.1 版*。

依預設，不會設定內容。

此內容對分配管理系統的即時連線無效。

JMSX_DELIVERY_COUNT**資料類型:**

System.Int32

下列項目的內容:

訊息

嘗試遞送訊息的次數。

此內容是具有 JMS 名稱 JMSXDeliveryCount 的 JMS 定義內容。如需內容的相關資訊，請參閱 *Java 訊息服務規格 1.1 版*。

依預設，不會設定內容。

此內容對分配管理系統的即時連線無效。

JMSX_GROUPID**資料類型:**

字串

下列項目的內容:

訊息

訊息所屬的訊息群組的 ID。

此內容是具有 JMS 名稱 JMSXGroupID 的 JMS 定義內容。如需內容的相關資訊，請參閱 *Java 訊息服務規格 1.1 版*。

依預設，不會設定內容。

此內容對分配管理系統的即時連線無效。

JMSX_GROUPSEQ**資料類型:**

System.Int32

下列項目的內容:

訊息

訊息群組內訊息的序號。

此內容是具有 JMS name JMSXGroupSeq 的 JMS 定義內容。如需內容的相關資訊，請參閱 *Java 訊息服務規格 1.1 版*。

依預設，不會設定內容。

此內容對分配管理系統的即時連線無效。

JMSX_USERID**資料類型:**

字串

下列項目的內容:

訊息

與傳送訊息的應用程式相關聯的使用者 ID。

此內容是具有 JMS 名稱 JMSXUserID 的 JMS 定義內容。如需內容的相關資訊，請參閱 *Java 訊息服務規格 1.1 版*。

依預設，不會設定內容。

此內容對分配管理系統的即時連線無效。

XMSC_ASYNC_EXCEPTIONS

資料類型：

System.Int32

下列項目的內容：

ConnectionFactory

此內容決定 XMS 是否只在連線已損毀或者 XMS API 呼叫有發生任何非同步異常狀況時通知 ExceptionListener。此內容適用於從這個登錄了 ExceptionListener 的 ConnectionFactory 建立的所有連線。

此內容的有效值如下：

XMSC_ASYNC_EXCEPTIONS_ALL

非同步偵測到的任何異常狀況、同步 API 呼叫的範圍之外，以及所有連線中斷異常狀況都會傳送至 ExceptionListener。

XMSC_ASYNC_EXCEPTIONS_CONNECTIONBROKEN

只有指出連線中斷的異常狀況才會傳送至 ExceptionListener。在非同步處理期間發生的任何其他異常狀況都不會向 ExceptionListener 報告，因此應用程式不會收到這些異常狀況的通知。

依預設，此內容設為 XMSC_ASYNC_EXCEPTIONS_ALL。

XMSC_CLIENT_ID

資料類型：

字串

下列項目的內容：

ConnectionFactory

連線的用戶端 ID。

用戶端 ID 只用來支援 發佈/訂閱 網域中的可延續訂閱，在 點對點 (point-to-point) 網域中會被忽略。如需設定用戶端 ID 的相關資訊，請參閱 [第 20 頁的『ConnectionFactory 和 Connection 物件』](#)。

此內容與分配管理系統的即時連線無關。

XMSC_CONNECTION_TYPE

資料類型：

System.Int32

下列項目的內容：

ConnectionFactory

應用程式所連接傳訊伺服器的類型。

內容的有效值如下：

有效值

XMSC_CT_RTT

XMSC_CT_WMQ

XMSC_CT_WPM

意義

與分配管理系統的即時連線。

WebSphere MQ 佇列管理程式的連線。

WebSphere 服務整合匯流排的連線。

依預設，不會設定內容。

XMSC_DELIVERY_MODE

資料類型:

System.Int32

下列項目的內容:

目的地

URI 中使用的名稱:

persistence (適用於 WebSphere MQ 目的地)

deliveryMode (適用於 WebSphere 預設傳訊提供者目的地)

傳送至目的地的訊息遞送模式。

內容的有效值如下:

有效值

XMSC_DELIVERY_NOT_持續性

XMSC_DELIVERY_持續性

XMSC_DELIVERY_AS_APP

XMSC_DELIVERY_AS_DEST

意義

傳送至目的地的訊息是非持續性。系統會忽略訊息產生者的預設遞送模式，或「傳送」呼叫上指定的任何遞送模式。如果目的地是 WebSphere MQ 佇列，也會忽略佇列屬性 *DefPersistence* 的值。

傳送至目的地的訊息是持續性的。系統會忽略訊息產生者的預設遞送模式，或「傳送」呼叫上指定的任何遞送模式。如果目的地是 WebSphere MQ 佇列，也會忽略佇列屬性 *DefPersistence* 的值。

傳送至目的地的訊息具有在「傳送」呼叫上指定的遞送模式。如果「傳送」呼叫指定無遞送模式，則會改用訊息產生者的預設遞送模式。如果目的地是 WebSphere MQ 佇列，則會忽略佇列屬性 *DefPersistence* 的值。

如果目的地是 WebSphere MQ 佇列，則放置在佇列上的訊息具有由佇列屬性 *DefPersistence* 值指定的遞送模式。系統會忽略訊息產生者的預設遞送模式，或「傳送」呼叫上指定的任何遞送模式。

如果目的地不是 WebSphere MQ 佇列，則其意義與 XMSC_DELIVERY_AS_APP 的意義相同。

預設值為 XMSC_DELIVERY_AS_APP。

XMSC_IC_PROVIDER_URL

資料類型:

字串

下列項目的內容:

InitialContext

用來找出 JNDI 命名目錄，這樣 COS 命名服務就不需要與 Web 服務位在同一伺服器上。

XMSC_IC_SECURITY_AUTHORIZATION

資料類型:

字串

下列項目的內容:

InitialContext

基於 Java 環境定義介面 SECURITY_AUTHENTICATION。此內容僅適用於 COS 命名環境定義。

XMSC_IC_SECURITY_CREDENTIALS

資料類型：

字串

下列項目的內容：

InitialContext

根據 Java 環境定義介面 SECURITY_CREDENTIALS。此內容僅適用於 COS 命名環境定義。

XMSC_IC_SECURITY_PRINCIPAL

資料類型：

字串

下列項目的內容：

InitialContext

基於 Java 環境定義介面 SECURITY_PRINCIPAL。此內容僅適用於 COS 命名環境定義。

XMSC_IC_SECURITY_PROTOCOL

資料類型：

字串

下列項目的內容：

InitialContext

基於 Java 環境定義介面 SECURITY_PROTOCOL 此內容僅適用於 COS 命名環境定義。

XMSC_IC_URL

資料類型：

字串

下列項目的內容：

InitialContext

對於 LDAP 和 FileSystem 環境定義，則為管理物件所在儲存庫的位址。

對於 COS 命名環境定義，這是查閱目錄中物件的 Web 服務位址。

XMSC_IS_SUBSCRIPTION_MULTICAST

資料類型：

System.Boolean

下列項目的內容：

MessageConsumer

指出是否使用 WebSphere MQ 多重播送傳輸將訊息遞送至訊息消費者。這是唯讀內容。

如果正在使用 WebSphere MQ 多重播送傳輸將訊息遞送至訊息消費者，則此內容的值為 true。否則，此值為 false。

此內容僅適用於與分配管理系統的即時連線。

XMSC_IS_SUBSCRIPTION_RELIABLE_MULTICAST

資料類型：

System.Boolean

下列項目的內容：

MessageConsumer

指出是否使用具有可靠服務品質的 WebSphere MQ 多重播送傳輸將訊息遞送至訊息消費者。這是唯讀內容。

如果使用 WebSphere MQ 多重播送傳輸 將訊息遞送至具有可靠服務品質的訊息消費者，則此內容的值为 true。否則，此值为 false。

此內容僅適用於與分配管理系統的即時連線。

XMSC_JMS_MAJOR_VERSION

資料類型：

System.Int32

下列項目的內容：

ConnectionMeta 資料

作為 XMS 基礎之 JMS 規格的主要版本號碼。這是唯讀內容。

XMSC_JMS_MINOR_VERSION

資料類型：

System.Int32

下列項目的內容：

ConnectionMeta 資料

XMS 所根據之 JMS 規格的次要版本號碼。這是唯讀內容。

XMSC_JMS_VERSION

資料類型：

字串

下列項目的內容：

ConnectionMeta 資料

XMS 所根據之 JMS 規格的版本 ID。這是唯讀內容。

XMSC_MAJOR_VERSION

資料類型：

System.Int32

下列項目的內容：

ConnectionMeta 資料

XMS 用戶端的版本號碼。這是唯讀內容。

XMSC_MINOR_VERSION

資料類型：

System.Int32

下列項目的內容：

ConnectionMeta 資料

XMS 用戶端的版次號碼。這是唯讀內容。

XMSC_PASSWORD

資料類型：

位元組陣列

下列項目的內容：

ConnectionFactory

應用程式在嘗試連接至傳訊伺服器時可以用來鑑別應用程式的密碼。密碼與 [XMSC_USERID](#) 內容一起使用。

依預設，不會設定內容。

如果您在分散式平台上連接至 WebSphere MQ，並且設定 Connection Factory 的 XMSC_USERID 內容，則它必須符合已登入使用者的 **userid**。如果您未設定這些內容，依預設，佇列管理程式會使用已登入使用者

的 **userid**。如果您需要個別使用者的進一步連線層次鑑別，則可以撰寫在 WebSphere MQ 中配置的用戶端鑑別結束程式。您可以在 WebSphere MQ Client 手冊的「鑑別」主題中，進一步瞭解如何建立用戶端鑑別結束程式。

若要在連接至 z/OS 上的 WebSphere MQ 時鑑別使用者，您需要使用安全結束程式。

XMSC_PRIORITY

資料類型：

System.Int32

下列項目的內容：

目的地

URI 中使用的名稱：

priority

傳送至目的地的訊息的優先順序。

內容的有效值如下：

有效值	意義
0 (最低優先順序) 到 9 (最高優先順序) 範圍內的整數	傳送至目的地的訊息具有指定的優先順序。系統不處理訊息產生者的預設優先順序，或「傳送」呼叫上指定的任何優先順序。如果目的地是 WebSphere MQ 佇列，也會忽略佇列屬性 DefPriority 的值。
XMSC_PRIORITY_AS_APP	傳送至目的地的訊息具有在「傳送」呼叫上指定的優先順序。如果「傳送」呼叫未指定優先順序，則會改用訊息產生者的預設優先順序。如果目的地是 WebSphere MQ 佇列，則會忽略佇列屬性 DefPriority 的值。
XMSC_PRIORITY_AS_DEST	如果目的地是 WebSphere MQ 佇列，則放置在佇列上的訊息具有佇列屬性 DefPriority 值所指定的優先順序。系統不處理訊息產生者的預設優先順序，或「傳送」呼叫上指定的任何優先順序。 如果目的地不是 WebSphere MQ 佇列，則其意義與 XMSC_PRIORITY_AS_APP 的意義相同。

預設值為 XMSC_PRIORITY_AS_APP。

WebSphere MQ 即時傳輸 和 WebSphere MQ 多重播送傳輸 不會根據訊息的優先順序採取任何動作。

XMSC_PROVIDER_NAME

資料類型：

字串

下列項目的內容：

ConnectionMeta 資料

XMS 用戶端的提供者。這是唯讀內容。

XMSC_RTT_BROKER_PING_INTERVAL

資料類型：

System.Int32

下列項目的內容：

ConnectionFactory

以毫秒為單位的時間間隔，在這段時間之後 XMS .NET 會檢查即時傳訊伺服器的連線以偵測任何活動。如果未偵測到任何活動，則用戶端會起始連線測試；如果未偵測到對連線測試的回應，則會關閉連線。

此內容的預設值為 30000。

XMSC_RTT_CONNECTION_PROTOCOL

資料類型:

System.Int32

下列項目的內容:

ConnectionFactory

用來即時連線至分配管理系統的通訊協定。

此內容的值必須是 XMSC_RTT_CP_TCP，這表示透過 TCP/IP 與分配管理系統的即時連線。預設值為 XMSC_RTT_CP_TCP。

XMSC_RTT_HOST_NAME

資料類型:

字串

下列項目的內容:

ConnectionFactory

分配管理系統在其中執行的系統的主機名稱或 IP 位址。

此內容與 [XMSC_RTT_PORT](#) 內容一起使用，以識別分配管理系統。

依預設，不會設定內容。

XMSC_RTT_LOCAL_ADDRESS

資料類型:

字串

下列項目的內容:

ConnectionFactory

用於即時連線至分配管理系統的本端網路介面的主機名稱或 IP 位址。

只有在執行應用程式的系統具有兩個以上網路介面，且您需要能夠指定必須用於即時連線的介面時，此內容才有用。如果系統只有一個網路介面，則只能使用該介面。如果系統有兩個以上網路介面，且未設定內容，則會隨機選取介面。

依預設，不會設定內容。

XMSC_RTT_MULTICAST

資料類型:

System.Int32

下列項目的內容:

ConnectionFactory 和目的地

URI 中使用的名稱:

穆利卡斯特

ConnectionFactory 或目的地的多重播送設定。只有主題的目的地可以有這個內容。

應用程式使用此內容來啟用與分配管理系統的即時連線相關聯的多重播送，並且如果已啟用多重播送，則指定使用多重播送將訊息從分配管理系統遞送至訊息消費者的精確方式。此內容不會影響訊息產生者如何將訊息傳送至分配管理系統。

內容的有效值如下:

有效值

XMSC_RTT_MULTICAST_DISABLED

意義

訊息不會使用 WebSphere MQ 多重播送傳輸遞送至訊息消費者。這個值是 ConnectionFactory 物件的預設值。

有效值

XMSC_RTT_MULTICAST_ASCF

已啟用 XMSC_RTT_MULTICAST_ENABLED

XMSC_RTT_MULTICAST_RELIABLE

XMSC_RTT_MULTICAST_NOT_RELIABLE

意義

根據與訊息消費者相關聯的 Connection Factory 的多重播送設定，將訊息遞送至訊息消費者。在建立連線時，會記下 Connection Factory 的多重播送設定。此值僅對「目的地」物件有效，且是「目的地」物件的預設值。

如果在分配管理系統中配置主題進行多重播送，則會使用 WebSphere MQ 多重播送傳輸將訊息遞送至訊息消費者。如果主題配置為可靠的多重播送，則會使用可靠的服務品質。

如果主題配置為在分配管理系統中進行可靠的多重播送，則會使用具有可靠服務品質的 WebSphere MQ 多重播送傳輸，將訊息遞送至訊息消費者。如果主題未配置為可靠多重播送，則無法建立主題的訊息消費者。

如果在分配管理系統中配置主題進行多重播送，則會使用 WebSphere MQ 多重播送傳輸將訊息遞送至訊息消費者。即使主題配置為可靠的多重播送，也不會使用可靠的服務品質。

XMSC_RTT_PORT

資料類型：

System.Int32

下列項目的內容：

ConnectionFactory

分配管理系統在其中接聽送入要求的埠的號碼。在分配管理系統上，您必須配置 Real-timeInput 或 Real-timeOptimized 流程訊息處理節點，才能在此埠上接聽。

此內容與 [XMSC_RTT_HOST_NAME](#) 內容一起使用，以識別分配管理系統。

此內容的預設值為 XMSC_RTT_DEFAULT_PORT 或 1506。

XMSC_TIME_TO_LIVE

資料類型：

System.Int32

下列項目的內容：

目的地

URI 中使用的名稱：

期限 (適用於 WebSphere MQ 目的地)

timeTo(適用於 WebSphere 預設傳訊提供者目的地)

傳送至目的地的訊息的存活時間。

內容的有效值如下：

有效值

0

正整數

意義

傳送至目的地的訊息永不到期。

傳送至目的地的訊息具有指定的存活時間 (毫秒)。系統會忽略訊息產生者的預設存活時間，或在「傳送」呼叫上指定的任何存活時間。

有效值

XMSC_TIME_TO_LIVE_AS_APP

意義

傳送至目的地的訊息具有在「傳送」呼叫上指定的存活時間。如果「傳送」呼叫未指定存活時間，則會改用訊息產生者的預設存活時間。

預設值為 XMSC_TIME_TO_LIVE_AS_APP。

XMSC_USERID

資料類型：

字串

下列項目的內容：

ConnectionFactory

應用程式在嘗試連接至傳訊伺服器時可以用來鑑別應用程式的使用者 ID。使用者 ID 與 XMSC_PASSWORD 內容一起使用。

依預設，不會設定內容。

如果您要連接至 WebSphere MQ 分散式平台，並設定 Connection Factory 的 XMSC_USERID 內容，它必須符合已登入使用者的 **userid**。如果您未設定這些內容，依預設，佇列管理程式會使用已登入使用者的 **userid**。如果您需要個別使用者的進一步連線層次鑑別，您可以撰寫在 WebSphere MQ 中配置的用戶端鑑別結束程式。

若要在連接至 z/OS 上的 WebSphere MQ 時鑑別使用者，您需要使用安全結束程式。

XMSC_VERSION

資料類型：

字串

下列項目的內容：

ConnectionMeta 資料

XMS 用戶端的版本 ID。這是唯讀內容。

XMSC_WMQ_BROKER_CONTROLQ

資料類型：

字串

下列項目的內容：

ConnectionFactory

分配管理系統使用的控制佇列的名稱。

此內容的預設值為 SYSTEM.BROKER.CONTROL.QUEUE。

此內容僅在 發佈/訂閱 網域中相關。

XMSC_WMQ_BROKER_PUBQ

資料類型：

字串

下列項目的內容：

ConnectionFactory

由分配管理系統監視且應用程式在其中傳送所發佈訊息的佇列的名稱。

此內容的預設值為 SYSTEM.BROKER.DEFAULT.STREAM。

此內容僅在 發佈/訂閱 網域中相關。

XMSC_WMQ_BROKER_QMGR

資料類型:

字串

下列項目的內容:

ConnectionFactory

分配管理系統所連接佇列管理程式的名稱。

依預設，不會設定內容。

此內容僅在 發佈/訂閱 網域中相關。

XMSC_WMQ_BROKER_SUBQ

資料類型:

字串

下列項目的內容:

ConnectionFactory

不可延續訊息消費者的訂閱者佇列的名稱。

訂閱者佇列的名稱必須以下列字元開頭:

SYSTEM.JMS.ND.

如果您想要所有不可延續訊息消費者共用訂閱者佇列，請指定共用佇列的完整名稱。具有指定名稱的佇列必須存在，應用程式才能建立不可延續訊息消費者。

如果您想要每一個不可延續訊息消費者從自己的專用訂閱者佇列擷取訊息，請指定以星號 (*) 結尾的佇列名稱。然後，當應用程式建立不可延續的訊息消費者時，XMS 用戶端會建立動態佇列供訊息消費者專用。XMS 用戶端會使用內容的值，在用來建立動態佇列的物件描述子中設定 **DynamicQName** 欄位的內容。

此內容的預設值為 SYSTEM.JMS.ND.SUBSCRIBER.QUEUE，表示依預設 XMS 會使用共用佇列方法。

此內容僅在 發佈/訂閱 網域中相關。

XMSC_WMQ_BROKER_VERSION

資料類型:

System.Int32

下列項目的內容:

ConnectionFactory 和目的地

URI 中使用的名稱:

brokerVersion

應用程式用於連線或目的地的分配管理系統的類型。只有主題的目的地可以有這個內容。

內容的有效值如下:

有效值

XMSC_WMQ_BROKER_V1

意義

應用程式正在使用 WebSphere MQ 發佈/訂閱 分配管理系統。

如果您從 WebSphere MQ 發佈/訂閱 移轉至 WebSphere Message Broker，但未變更應用程式，則應用程式也可以使用此值。

XMSC_WMQ_BROKER_V2

應用程式正在使用 IBM Integration Bus 的分配管理系統。

XMSC_WMQ_BROKER_UNSPECIFIED

移轉分配管理系統之後，請設定此內容，以便不再使用 RFH2 標頭。移轉之後，此內容不再相關。

connectionfactory 的預設值為 XMSC_WMQ_BROKER_UNSPECIFIED，但依預設不會設定目的地的內容。設定目的地的內容會置換 Connection Factory 內容所指定的任何值。

XMSC_WMQ_CCDTURL

資料類型:

System.String

下列項目的內容:

ConnectionFactory

識別檔案名稱和位置的統一資源定址器 (URL)，該檔案包含用戶端通道定義表，而且還指定如何存取該檔案。

依預設，不會設定此內容。

XMSC_WMQ_CC SID

資料類型:

System.Int32

下列項目的內容:

目的地

URI 中使用的名稱:

CCSID

當 XMS 用戶端將訊息轉遞至目的地時，訊息內文中的字元資料字串所使用的編碼字集 (或字碼頁) ID (CCSID)。如果針對個別訊息設定，JMS_IBM_CHARACTER_SET 內容會置換此內容為目的地指定的 CCSID。

此內容的預設值為 1208。

此內容僅與傳送至目的地的訊息相關，與從目的地接收的訊息無關。

XMSC_WMQ_CHANNEL

資料類型:

字串

下列項目的內容:

ConnectionFactory

要用於連線的通道名稱。

依預設，不會設定內容。

僅當應用程式以用戶端模式連接至佇列管理程式時，此內容才相關。

XMSC_WMQ_CLIENT_RECONNECT_OPTIONS

資料類型:

字串

下列項目的內容:

ConnectionFactory

這個內容指定這個 Factory 所建立之新連線的用戶端重新連接選項。它位於 XMSC 中，並且是下列其中一項：

- WMQ_CLIENT_RECONNECT_AS_DEF (預設值)。使用 mqclient.ini 檔案中指定的值。使用「通道」段落內的 **DefRecon** 內容來設定值。它可以設為下列其中一項：
 1. 是。行為與 WMQ_CLIENT_RECONNECT 選項相同
 2. 沒有。預設值。不指定任何重新連線選項
 3. QMGR。行為與 WMQ_CLIENT_RECONNECT_Q_MGR 選項相同
 4. 已停用。行為與 WMQ_CLIENT_RECONNECT_DISABLED 選項相同
- WMQ_CLIENT_RECONNECT。重新連接連線名稱清單中指定的任何佇列管理程式。

- WMQ_CLIENT_RECONNECT_Q_MGR。重新連接至原先連接的相同佇列管理程式。如果它嘗試連接的佇列管理程式 (在連線名稱清單中指定) 與最初連接的佇列管理程式具有不同的 QMID，則它會傳回 MQRC_RECONNECT_QMID_MISMATCH。
- WMQ_CLIENT_RECONNECT_DISABLED。已停用重新連線。

XMSC_WMQ_CLIENT_RECONNECT_TIMEOUT

資料類型：

字串

下列項目的內容：

ConnectionFactory

XMSC_WMQ_CLIENT_RECONNECT_TIMEOUT 內容僅適用於受管理 XMS .NET 用戶端。

此內容指定用戶端連線嘗試重新連接的持續時間 (以秒為單位)。

在此期間嘗試重新連接之後，用戶端將會失敗，並出現 MQRC_RECONNECT_FAILED。此內容的預設值為 XMSC_WMQ_CLIENT_RECONNECT_TIMEOUT_DEFAULT。

此內容的預設值為 1800。

XMSC_WMQ_CONNECTION_MODE

資料類型：

System.Int32

下列項目的內容：

ConnectionFactory

應用程式據以連線至佇列管理程式的模式。

內容的有效值如下：

有效值	意義
XMSC_WMQ_CM_BINDINGS	以連結模式連接至佇列管理程式，以取得最佳效能。此值是 C/C++ 的預設值。
XMSC_WMQ_CM_CLIENT	以用戶端模式連接至佇列管理程式，以確保完全受管理堆疊。此值是 .NET 的預設值。
XMSC_WMQ_CM_CLIENT_UNMANAGED (僅適用於 .NET)	與佇列管理程式的連線，該佇列管理程式會強制執行未受管理的用戶端堆疊。

相關概念

.NET 中的受管理及未受管理作業

受管理程式碼專門在 .NET 一般語言執行時期環境內執行，完全相依於該執行時期所提供的服務。如果應用程式的任何部分在 .NET 一般語言執行時期環境之外執行或呼叫服務，則應用程式會分類為未受管理。

XMSC_WMQ_CONNECTION_NAME_LIST

資料類型：

字串

下列項目的內容：

ConnectionFactory

此內容指定用戶端在其連線中斷之後嘗試重新連接的主機。

連線名稱清單是以逗點區隔的主機 /IP 埠配對清單。此內容的預設值為 WMQ_CONNECTION_NAME_LIST_DEFAULT。

例如，127.0.0.1 (1414) , host2.example.com(1400)

此內容的預設值是 localhost (1414)。

XMSC_WMQ_DUR_SUBQ

資料類型：

字串

下列項目的內容：

目的地

從目的地接收訊息的可延續訂閱者的訂閱者佇列名稱。只有主題的目的地可以有這個內容。

訂閱者佇列的名稱必須以下列字元開頭：

SYSTEM.JMS.D.

如果您想要所有可延續訂閱者共用訂閱者佇列，請指定共用佇列的完整名稱。具有指定名稱的佇列必須存在，應用程式才能建立可延續訂閱者。

如果您想要每一個可延續訂閱者從它自己的專用訂閱者佇列擷取訊息，請指定以星號 (*) 結尾的佇列名稱。然後，當應用程式建立可延續訂閱者時，XMS 用戶端會建立動態佇列供可延續訂閱者專用。XMS 用戶端會使用內容的值，在用來建立動態佇列的物件描述子中設定 **DynamicQName** 欄位的內容。

此內容的預設值為 SYSTEM.JMS.D.SUBSCRIBER.QUEUE，表示依預設 XMS 會使用共用佇列方法。

此內容僅在 發佈/訂閱 網域中相關。

XMSC_WMQ_ENCODING

資料類型：

System.Int32

下列項目的內容：

目的地

當 XMS 用戶端將訊息轉遞至目的地時，訊息內文中的數值資料如何呈現。如果針對個別訊息設定，JMS_IBM_ENCODING 內容會置換此內容為目的地指定的編碼。此內容指定二進位整數、聚集十進位整數及浮點數字的表示法。

內容的有效值與可以在訊息描述子的 **Encoding** 欄位中指定的值相同。

應用程式可以使用下列具名常數來設定內容：

已命名的常數	意義
MQENC_INTEGER_NORMAL	正常整數編碼
已反轉 MQENC_INTEGER_REVERSED	反向整數編碼
MQENC_DECIMAL_NORMAL	一般聚集十進位編碼
MQENC_DECIMAL_REVERED	反向壓縮十進位編碼
MQENC_FLOAT_IEE_NORMAL	一般 IEEE 浮點數編碼
MQENC_FLOAT_IEE_REVERSED	反向 IEEE 浮點數編碼
MQENC_FLOAT_S390	z/OS 架構浮點數編碼
MQENC_NATIVE	原生機器編碼

若要形成內容的值，應用程式可以新增下列三個常數，如下所示：

- 名稱以 MQENC_INTEGER 開始的常數，用來指定二進位整數的表示法
- 名稱以 MQENC_DECIMAL 開始的常數，用來指定聚集十進位整數的表示法
- 名稱以 MQENC_FLOAT 開始的常數，用來指定浮點數字的表示法

或者，應用程式可以將內容設為 MQENC_NATIVE，其值與環境相關。

此內容的預設值為 MQENC_NATIVE。

此內容僅與傳送至目的地的訊息相關，與從目的地接收的訊息無關。

XMSC_WMQ_FAIL_IF_QUIESCE

資料類型:

System.Int32

下列項目的內容:

ConnectionFactory 和目的地

URI 中使用的名稱:

FAILIFQUIESCE

在應用程式所連接的佇列管理程式處於靜止狀態時對某些方法的呼叫是否失敗。

內容的有效值如下:

有效值	意義
XMSC_WMQ_FIQ_YES	如果佇列管理程式處於靜止狀態，則對特定方法的呼叫會失敗。當應用程式偵測到佇列管理程式正在靜止時，應用程式可以完成其立即作業並關閉連線，讓佇列管理程式停止。
XMSC_WMQ_FIQ_NO	沒有方法呼叫失敗，因為佇列管理程式處於靜止狀態。如果您指定此值，則應用程式無法偵測到佇列管理程式正在靜止。應用程式可能會繼續對佇列管理程式執行作業，因此防止佇列管理程式停止。

ConnectionFactory 的預設值是 XMSC_WMQ_FIQ_YES，但依預設不會設定目的地的內容。設定目的地的內容會置換 Connection Factory 內容所指定的任何值。

XMSC_WMQ_MESSAGE_BODY

資料類型:

System.Int32

下列項目的內容:

目的地

此內容決定 XMS 應用程式是否將 IBM WebSphere MQ 訊息的 MQRFH2 作為訊息有效負載的一部分 (亦即，作為訊息內文的一部分) 處理。

註: 將訊息傳送至目的地時，XMSC_WMQ_MESSAGE_BODY 內容會取代現有的 XMS 目的地內容 XMSC_WMQ_TARGET_CLIENT。

此內容的有效值如下:

XMSC_WMQ_MESSAGE_BODY_JMS

接收: 入埠 XMS 訊息類型和內文由接收的 IBM WebSphere MQ 訊息中 MQRFH2 (如果有的話) 或 MQMD (如果沒有 MQRFH2) 的內容所決定。

傳送: 出埠 XMS 訊息內文包含以 XMS 訊息內容及標頭欄位為基礎，預先附加且自動產生的 MQRFH2 標頭。

XMSC_WMQ_MESSAGE_BODY_MQ

接收: 入埠 XMS 訊息類型一律為 ByteMessage，與接收的 IBM WebSphere MQ 訊息內容或接收的 MQMD 格式欄位無關。XMS 訊息內文是基礎傳訊提供者 API 呼叫所傳回的未變更訊息資料。訊息內文中資料的字集及編碼由 MQMD 的 CodedCharSetId 及「編碼」欄位決定。訊息內文中資料的格式由 MQMD 的「格式」欄位決定。

傳送: 出埠 XMS 訊息內文依現狀包含應用程式有效負載; 且不會將任何自動產生的 IBM WebSphere MQ 標頭新增至內文。

XMSC_WMQ_MESSAGE_BODY_UNSPECIFIED

接收: XMS 用戶端決定適合這個內容的值。在接收路徑上，此值是 WMQ_MESSAGE_BODY_JMS 內容值。

傳送: XMS 用戶端決定適合這個內容的值。在傳送路徑上，此值是 XMSC_WMQ_TARGET_CLIENT 內容值。

依預設，此內容設為 XMSC_WMQ_MESSAGE_BODY_UNSPECIFIED。

XMSC_WMQ_MQMD_MESSAGE_CONTEXT

資料類型：

System.Int32

下列項目的內容：

目的地

決定 XMS 應用程式要設定的訊息環境定義層次。應用程式必須以適當的環境定義權限執行，才能使此內容生效。

此內容的有效值如下：

XMSC_WMQ_MDCTX_DEFAULT

對於出埠訊息，MQOPEN API 呼叫及 MQPMO 結構不會指定明確訊息環境定義選項。

XMSC_WMQ_MDCTX_SET_IDENTITY_CONTEXT

MQOPEN API 呼叫會指定訊息環境定義選項 MQOO_SET_IDENTITY_CONTEXT，而 MQPMO 結構會指定 MQPMO_SET_IDENTITY_CONTEXT。

XMSC_WMQ_MDCTX_SET_ALL_CONTEXT

MQOPEN API 呼叫指定訊息環境定義選項 MQOO_SET_ALL_CONTEXT，MQPMO 結構指定 MQPMO_SET_ALL_CONTEXT。

依預設，此內容設為 XMSC_WMQ_MDCTX_DEFAULT。

註：當應用程式連接至系統 Integration Bus 時，此內容不相關。

下列內容需要在傳送訊息時，將 XMSC_WMQ_MQMD_MESSAGE_CONTEXT 內容設為 XMSC_WMQ_MDCTX_SET_IDENTITY_CONTEXT 內容值或 XMSC_WMQ_MDCTX_SET_ALL_CONTEXT 內容值，以達到想要的效果：

- JMS_IBM_MQMD_USERIDENTIFIER
- JMS_IBM_MQMD_ACCOUNTINGtoken
- JMS_IBM_MQMD_APPLIDENTITYDATA

下列內容需要在傳送的訊息時，將 XMSC_WMQ_MQMD_MESSAGE_CONTEXT 內容設為 XMSC_WMQ_MDCTX_SET_ALL_CONTEXT 內容值，以便具有想要的效果：

- JMS_IBM_MQMD_PUTAPPLTYPE
- JMS_IBM_MQMD_PUTAPPLNAME
- JMS_IBM_MQMD_PUTDATE
- JMS_IBM_MQMD_PUTTIME
- JMS_IBM_MQMD_APPLORIGINDATA

XMSC_WMQ_MQMD_READ_ENABLED

資料類型：

System.Int32

下列項目的內容：

目的地

此內容決定 XMS 應用程式是否可以擷取 MQMD 欄位的值。

此內容的有效值如下：

XMSC_WMQ_READ_ENABLED_NO

傳送訊息時，不會更新已傳送訊息上的 JMS_IBM_MQMD* 內容，以反映 MQMD 中已更新的欄位值。

接收訊息時，所接收訊息上沒有可用的 JMS_IBM_MQMD* 內容，即使其中部分或全部由寄件者設定也一樣。

XMSC_WMQ_READ_ENABLED_YES

傳送訊息時，會更新已傳送訊息上的所有 JMS_IBM_MQMD* 內容，以反映 MQMD 中已更新的欄位值，包括寄件者未明確設定的那些內容。

接收訊息時，所有 JMS_IBM_MQMD* 內容 (包括寄件者未明確設定的那些內容) 可用於接收的訊息。

依預設，此內容設為 XMSC_WMQ_READ_ENABLED_NO。

XMSC_WMQ_MQMD_WRITE_ENABLED**資料類型：**

System.Int32

下列項目的內容：

目的地

此內容決定 XMS 應用程式是否可以使用 MQMD 欄位的值。

此內容的有效值如下：

XMSC_WMQ_WRITE_ENABLED_NO

會忽略所有 JMS_IBM_MQMD* 內容，且其值不會複製到基礎 MQMD 結構。

XMSC_WMQ_WRITE_ENABLED_YES

會處理 JMS_IBM_MQMD* 內容。其值會複製到基礎 MQMD 結構。

依預設，此內容設為 XMSC_WMQ_WRITE_ENABLED_NO。

XMSC_WMQ_PUT_ASYNC_ALLOWED**資料類型：**

System.Int32

下列項目的內容：

目的地

此內容決定是否容許訊息產生者使用非同步放置，以傳送訊息至此目的地。

此內容的有效值如下：

XMSC_WMQ_PUT_ASYNC_ALLOWED_AS_DEST

透過參照佇列或主題定義來判定是否容許非同步放置。

XMSC_WMQ_PUT_ASYNC_ALLOWED_AS_Q_DEF

請參照佇列定義來判斷是否容許非同步放置。

XMSC_WMQ_PUT_ASYNC_ALLOWED_AS_TOPIC_DEF

請參照主題定義來判斷是否容許非同步放置。

XMSC_WMQ_PUT_ASYNC_ALLOWED_DISABLED

不容許非同步放置。

XMSC_WMQ_PUT_ASYNC_ALLOWED_ENABLED

容許非同步放置。

依預設，此內容設為 XMSC_WMQ_PUT_ASYNC_ALLOWED_AS_DEST。

註：當應用程式連接至系統 Integration Bus 時，此內容不相關。

XMSC_WMQ_READ_Ahead_ALLOWED**資料類型：**

System.Int32

下列項目的內容：

目的地

此內容決定是否容許訊息消費者和佇列瀏覽器在接收訊息之前使用先讀，將這個目的地的非持續性、非交易性訊息讀到內部緩衝區。

此內容的有效值如下：

XMSC_WMQ_READ_AHEAD_ALLOWED_AS_Q_DEF

請參照佇列定義來決定是否容許先讀。

XMSC_WMQ_READ_AHEAD_ALLOWED_AS_TOPIC_DEF

請參閱主題定義來判斷是否容許先讀。

XMSC_WMQ_READ_AHEAD_ALLOWED_AS_DEST

請參照佇列或主題定義來決定是否容許先讀。

XMSC_WMQ_READ_AHEAD_ALLOWED_DISABLED

耗用或瀏覽訊息時不容許先讀。

XMSC_WMQ_READ_AHEAD_ALLOWED_ENABLED

容許先讀。

依預設，此內容設為 XMSC_WMQ_READ_AHEAD_ALLOWED_AS_DEST。

XMSC_WMQ_READ_AHEAD_CLOSE_policy

資料類型：

System.Int32

下列項目的內容：

目的地

對於將遞送至非同步訊息接聽器的訊息，此內容決定在消息消費者關閉時內部先讀緩衝區中的訊息發生的情況。

此內容適用於在使用來自目的地的訊息時指定關閉佇列選項，而在將訊息傳送至目的地時不適用。

「佇列瀏覽器」會忽略此內容，因為在瀏覽期間，訊息仍可在佇列中使用。

此內容的有效值如下：

XMSC_WMQ_READ_AHEAD_CLOSE_POLICY_DELIVER_CURRENT

只有現行訊息接聽器呼叫在傳回之前完成，可能會將訊息留在內部先讀緩衝區中，然後捨棄這些訊息。

XMSC_WMQ_READ_AHEAD_CLOSE_POLICY_DELIVER_ALL

在傳回之前，內部先讀緩衝區中的所有訊息都會遞送至應用程式訊息接聽器。

依預設，此內容設為 XMSC_WMQ_READ_AHEAD_CLOSE_POLICY_DELIVER_CURRENT。

註：

• **異常應用程式終止**

當 XMS 應用程式突然終止時，先讀緩衝區中的所有訊息都會遺失。

• **交易的含意**

當應用程式使用交易時，會停用先讀。因此，應用程式在使用交易式階段作業時不會看到任何行為差異。

• **階段作業確認模式的含意**

當確認模式為 XMSC_AUTO_ACKNOWLEDGE 或 XMSC_DUPS_OK_ACKNOWLEDGE 時，會在非交易式階段作業上啟用先讀。如果階段作業確認模式是 XMSC_CLIENT_ACKNOWLEDGE，則不論交易式或非交易式階段作業為何，都會停用先讀。

• **佇列瀏覽器及佇列瀏覽器選取器上的含意**

在 XMS 應用程式中使用的「佇列瀏覽器」及「佇列瀏覽器選取器」，可從先讀取得效能優勢。關閉「佇列瀏覽器」不會降低效能，因為佇列中仍有訊息可供進一步作業使用。除了先讀的效能優點之外，在佇列瀏覽器及佇列瀏覽器選取器上沒有任何其他含意。

XMSC_WMQ_HOST_NAME

資料類型：

字串

下列項目的內容：

ConnectionFactory

佇列管理程式在其中執行的系統的主機名稱或 IP 位址。

只有在應用程式以用戶端模式連接至佇列管理程式時，才會使用此內容。此內容與 [XMSC_WMQ_PORT](#) 內容一起使用，以識別佇列管理程式。

此內容的預設值為 localhost。

XMSC_WMQ_LOCAL_ADDRESS

資料類型：

字串

下列項目的內容：

ConnectionFactory

為了連線至佇列管理程式，此內容會指定要使用的本端網路介面和/或要使用的本端埠（或本端埠的範圍）。

內容的值是具有下列格式的字串：

```
[host_name] [(low_port) [, high_port]]
```

變數的意義如下：

host_name

要用於連線之本端網路介面的主機名稱或 IP 位址。

只有在執行應用程式的系統具有兩個以上網路介面，且您需要能夠指定連線必須使用哪個介面時，才需要提供此資訊。如果系統只有一個網路介面，則只能使用該介面。如果系統有兩個以上網路介面，且您未指定必須使用哪個介面，則會隨機選取該介面。

low_port

要用於連線的本端埠號。

如果也指定 *high_port*，則會解譯埠號範圍中的最低埠號 *low_port*。

高埠

埠號範圍中的最高埠號。指定範圍內的其中一個埠必須用於連線。

字串的長度上限為 48 個字元。

以下是內容有效值的一些範例：

木星

9.20.4.98

JUPITER (1000)

9.20.4.98(1000,2000)

(1000)

(1000,2000)

依預設，不會設定內容。

僅當應用程式以用戶端模式連接至佇列管理程式時，此內容才相關。

XMSC_WMQ_MESSAGE_SELECTION

資料類型：

System.Int32

下列項目的內容：

ConnectionFactory

決定訊息選擇是由 XMS 用戶端還是分配管理系統完成。

內容的有效值如下：

有效值	意義
XMSC_WMQ_MSEL_CLIENT	訊息選擇由 XMS 用戶端完成。
XMSC_WMQ_MSEL_BROKER	訊息選擇由分配管理系統完成。

預設值為 XMSC_WMQ_MSEL_CLIENT。

此內容僅在 發佈/訂閱 網域中相關。如果 `XMSC_WMQ_BROKER_VERSION` 內容設為 `XMSC_WMQ_BROKER_V1`，則不支援分配管理系統選取訊息。

XMSC_WMQ_MSG_BATCH_SIZE

資料類型：

System.Int32

下列項目的內容：

ConnectionFactory

使用非同步傳訊遞送時，透過一個批次從佇列中擷取的訊息數目上限。

當應用程式使用非同步訊息遞送時，在特定條件下，XMS 用戶端會先從佇列擷取一批訊息，然後再個別將每一則訊息轉遞至應用程式。此內容指定可以在批次中的訊息數上限。

內容的值是正整數，預設值是 10。只有在您有需要解決的特定效能問題時，才考量將內容設為不同的值。

如果應用程式透過網路連接至佇列管理程式，則提高此內容的值可以減少網路額外需要及回應時間，但會增加在用戶端系統上儲存訊息所需的記憶體數量。相反地，降低此內容的值可能會增加網路額外需要及回應時間，但會減少儲存訊息所需的記憶體數量。

XMSC_WMQ_POLLING_INTERVAL

資料類型：

System.Int32

下列項目的內容：

ConnectionFactory

如果階段作業內每個訊息接聽器的佇列上都沒有適當的訊息，這個值便是每個訊息接聽器在重新嘗試從它的佇列取得訊息之前，所經歷的間隔上限（毫秒）。

如果經常發生階段作業中的任何訊息接聽器都沒有可用的適當訊息，請考量增加此內容的值。

內容的值是正整數。預設值是 5000。

XMSC_WMQ_PORT

資料類型：

System.Int32

下列項目的內容：

ConnectionFactory

佇列管理程式在其中接聽送入要求的埠的號碼。

只有在應用程式以用戶端模式連接至佇列管理程式時，才會使用此內容。此內容與 `XMSC_WMQ_HOST_NAME` 內容一起使用，以識別佇列管理程式。

此內容的預設值為 `XMSC_WMQ_DEFAULT_CLIENT_PORT` 或 1414。

XMSC_WMQ_PROVIDER_VERSION

資料類型：

字串

下列項目的內容:

ConnectionFactory

應用程式準備連接的佇列管理程式的版本、版次、修正層次和修正套件。此內容的有效值如下:

- 未指定

或下列其中一種格式的字串

- V.R.M.F
- V.R.M
- V.R
- V

其中 V、R、M 和 F 是大於或等於零的整數值。

值為 7 或以上表示此版本預期作為 IBM WebSphere MQ 7.0 版 佇列管理程式連線的 IBM WebSphere MQ 7.0 版 ConnectionFactory。7 之前的值 (例如 "6.0.2.0") 指出它預期與 7.0 版之前的佇列管理程式搭配使用。預設值 (未指定) 容許連線至任何層次的佇列管理程式, 並根據佇列管理程式的功能來決定可用的適用內容及功能。

依預設, 此內容會設為 "unspecified"。

註:

- 如果 XMSC_WMQ_PROVIDER_VERSION 設為 6, 則不會發生 Socket 共用。2。
- 如果通道的 XMSC_WMQ_PROVIDER_VERSION 設為 7, 且在伺服器 SHARECNV 上設為 0, 則連線失敗。
- 如果 XMSC_WMQ_PROVIDER_VERSION 設為 UNSPECIFIED 且 SHARECNV 設為 0, 則會停用 IBM WebSphere MQ 7.0 版 特定特性。

IBM WebSphere MQ Client 的版本也在 XMS 用戶端應用程式是否可以與 IBM WebSphere MQ 7.0 版 特定特性中扮演主要角色。下表說明行為。

註: 系統內容 XMSC_WMQ_OVERRIDEPROVIDERVERSION 會置換 XMSC_WMQ_PROVIDER_VERSION 內容。如果您無法變更 Connection Factory 設定, 則可以使用此內容。

表 34: XMS 用戶端-能夠使用 IBM WebSphere MQ 7.0 版 特定特性。

#	XMSC_WMQ_PROVIDER_VERSION	IBM WebSphere MQ 用戶端版本	IBM WebSphere MQ 7.0 版 功能
1	未指定	7	開啟
2	未指定	6	關閉
3	7	7	開啟
4	7	6	異常狀況
5	6	6	關閉
6	6	7	關閉

XMSC_WMQ_PUB_ACK_INTERVAL

資料類型:

System.Int32

下列項目的內容:

ConnectionFactory

在 XMS 用戶端要求分配管理系統確認之前, 發佈者已發佈的訊息數。

如果您減少此內容的值, 則用戶端會更頻繁地要求確認, 因此發佈者的效能會降低。如果您提高此值, 萬一分配管理系統失敗, 則用戶端會花更久的時間擲出異常。

內容的值是正整數。預設值是 25。

XMSC_WMQ_QMGR_CCSID

資料類型：

System.Int32

下列項目的內容：

ConnectionFactory

編碼字集 (或字碼頁) 的 ID (CCSID)，在 XMS 用戶端與 WebSphere MQ 用戶端之間交換「訊息佇列介面 (MQI)」中定義的字元資料欄位。此內容不適用於訊息內文中的字元資料字串。

當 XMS 應用程式以用戶端模式連接至佇列管理程式時，XMS 用戶端會鏈結至 WebSphere MQ 用戶端。在兩個用戶端之間交換的資訊包含在 MQI 中定義的字元資料欄位。在一般情況下，WebSphere MQ 用戶端會假設這些欄位是使用用戶端執行所在系統的字碼頁。如果 XMS 用戶端提供並預期以不同的字碼頁接收這些欄位，則您必須設定此內容以通知 WebSphere MQ 用戶端。

當 WebSphere MQ 用戶端將這些字元資料欄位轉遞至佇列管理程式時，必要的話，必須將其中的資料轉換成佇列管理程式所使用的字碼頁。同樣地，當 WebSphere MQ 用戶端從佇列管理程式接收這些欄位時，必要的話，必須將其中的資料轉換成 XMS 用戶端預期用來接收資料的字碼頁。WebSphere MQ 用戶端使用此內容來執行這些資料轉換。

依預設，不會設定內容。

設定此內容相當於為支援原生 WebSphere MQ 用戶端應用程式的 WebSphere MQ 用戶端設定 MQCCSID 環境變數。如需此環境變數的相關資訊，請參閱 *WebSphere MQ* 用戶端。

XMSC_WMQ_QUEUE_MANAGER

資料類型：

字串

下列項目的內容：

ConnectionFactory

要連接的佇列管理程式的名稱。

依預設，不會設定內容。

XMSC_WMQ_RECEIVE_CCSID

設定佇列管理程式訊息轉換的目標 CCSID 的目的地內容。除非 XMSC_WMQ_RECEIVE_CONVERSION 設為 WMQ_RECEIVE_CONVERSION_QMGR，否則會忽略此值。

資料類型：

整數

值：

任何正整數。

預設值為 1208。

在訊息中指定 GMO_CONVERT 值是選用的。如果指定 GMO_CONVERT 值，則會根據指定的值進行轉換。

XMSC_WMQ_RECEIVE_CONVERSION

決定佇列管理程式是否要執行資料轉換的目的地內容。

資料類型：

整數

值：

XMSC_WMQ_RECEIVE_CONVERSION_CLIENT_MSG (DEFAULT): 僅在 XMS 用戶端上執行資料轉換。一律使用字碼頁 1208 來完成轉換。

XMSC_WMQ_RECEIVE_CONVERSION_QMGR: 將訊息傳送至 XMS 用戶端之前，請先在佇列管理程式上執行資料轉換。

XMSC_WMQ_RECEIVE_EXIT

資料類型：

字串

下列項目的內容：

ConnectionFactory

識別要執行的通道接收結束程式。

此內容的值是一個字串，用於識別通道接收結束程式並具有下列格式：

libraryName(entryPoint 名稱)

其中：

- **libraryName** 是受管理結束程式 .dll 的完整路徑
- **entryPoint 名稱** 是名稱空間所限定的類別名稱

例如：C:\MyReceiveExit.dll(MyReceiveExitNameSpace.MyReceiveExitClassName)

依預設，不會設定內容。

只有在應用程式以受管理用戶端模式連接至佇列管理程式時，此內容才相關。此外，僅支援受管理結束程式。

XMSC_WMQ_RECEIVE_EXIT_INIT

資料類型：

字串

下列項目的內容：

ConnectionFactory

通道接收結束程式在被呼叫時收到的使用者資料。

內容的值是字串。依預設，不會設定內容。

只有在應用程式以受管理用戶端模式連接至佇列管理程式，且已設定 [第 201 頁的『XMSC_WMQ_RECEIVE_EXIT』](#) 內容時，此內容才相關。

XMSC_WMQ_RESOLVED_QUEUE_MANAGER

資料類型：

字串

下列項目的內容：

ConnectionFactory

這個內容用來取得它所連接的佇列管理程式名稱。

與 CCDT (用戶端通道定義表) 搭配使用時，此名稱可能與 Connection Factory 中指定的佇列管理程式名稱不同。

XMSC_WMQ_RESOLVED_QUEUE_MANAGER_ID

資料類型：

字串

下列項目的內容：

ConnectionFactory

此內容會在連線之後移入佇列管理程式的 ID。

XMSC_WMQ_SECURITY_EXIT

資料類型：

字串

下列項目的內容:

ConnectionFactory

識別通道安全結束程式。

此內容的值是識別通道安全結束程式的字串，格式如下:

libraryName(entryPoint 名稱)

其中:

- **libraryName** 是受管理結束程式 .dll 的完整路徑
- **entryPoint 名稱** 是名稱空間所限定的類別名稱

例如, C:\MySecurityExit.dll(MySecurityExitNameSpace.MySecurityExitClassName)

字串的長度上限為 128 個字元。

依預設, 不會設定內容。

只有在應用程式以受管理用戶端模式連接至佇列管理程式時, 此內容才相關。此外, 僅支援受管理結束程式。

XMSC_WMQ_SECURITY_EXIT_INIT

資料類型:

字串

下列項目的內容:

ConnectionFactory

通道安全結束程式在被呼叫時收到的使用者資料。

使用者資料的字串長度上限為 32 個字元。

依預設, 不會設定內容。

只有在應用程式以受管理用戶端模式連接至佇列管理程式, 且已設定 [第 201 頁的『XMSC_WMQ_SECURITY_EXIT』](#) 內容時, 此內容才相關。

XMSC_WMQ_SEND_EXIT

資料類型:

字串

下列項目的內容:

ConnectionFactory

識別通道傳送結束程式。

內容的值是字串。通道傳送結束程式具有下列格式:

libraryName(entryPoint 名稱)

其中:

- **libraryName** 是受管理結束程式 .dll 的完整路徑
- **entryPoint 名稱** 是名稱空間所限定的類別名稱

例如: C:\MySendExit.dll(MySendExitNameSpace.MySendExitClassName)

依預設, 不會設定內容。

只有在應用程式以受管理用戶端模式連接至佇列管理程式時, 此內容才相關。此外, 僅支援受管理結束程式。

XMSC_WMQ_SEND_EXIT_INIT

資料類型:

字串

下列項目的內容:

ConnectionFactory

傳送給所呼叫之通道傳送結束程式的使用者資料。

此內容的值是一或多個使用者資料項目的字串，以逗點區隔。依預設，不會設定內容。

指定傳給一連串通道傳送結束程式的使用者資料的規則，與指定傳給一連串通道接收結束程式的使用者資料的規則相同。因此，如需規則，請參閱第 201 頁的『[XMSC_WMQ_RECEIVE_EXIT_INIT](#)』。

只有在應用程式以受管理用戶端模式連接至佇列管理程式，且已設定第 202 頁的『[XMSC_WMQ_SEND_EXIT](#)』內容時，此內容才相關。

XMSC_WMQ_SEND_CHECK_COUNT

資料類型:

System.Int32

下列項目的內容:

ConnectionFactory

在單一非交易式 XMS 階段作業內檢查非同步放置錯誤之間所容許的傳送呼叫次數。

依預設，此內容設為 0。

XMSC_WMQ_SHARE_CONV_ALLOWED

資料類型:

System.Int32

下列項目的內容:

ConnectionFactory

如果通道定義相符，用戶端連線是否可以與從相同處理程序至相同佇列管理程式的其他最上層 XMS 連線共用其 Socket。系統提供此內容以容許根據需要出於應用程式開發、維護或作業原因，將連線完全隔離到單獨的 Socket 中。設定此內容只會指示 XMS 讓基礎 Socket 成為共用。它不會指出有多少連線共用單一 Socket。共用 Socket 的連線數目由 WebSphere MQ 用戶端與 WebSphere MQ 伺服器之間協議的 SHARECNV 值決定。

應用程式可以設定下列具名常數來設定內容:

- XMSC_WMQ_SHARE_CONV_ALLOWED_FALSE-連線不共用 Socket。
- XMSC_WMQ_SHARE_CONV_ALLOWED_TRUE-連線共用 Socket。

依預設，此內容會設為 XMSC_WMQ_SHARE_CONV_ALLOWED_ENABLED。

僅當應用程式以用戶端模式連接至佇列管理程式時，此內容才相關。

XMSC_WMQ_SSL_CERT_STORES

資料類型:

字串

下列項目的內容:

ConnectionFactory

保留給佇列管理程式建立 SSL 連線時所使用憑證撤銷清冊 (CRL) 的伺服器位置。

此內容的值是以逗點區隔的一或多個 URL 清單。每一個 URL 都具有下列格式:

```
[user[/password]@]ldap://[serveraddress][:portnum][,...]
```

此格式與基本 MQJMS 格式相容，但延伸自基本 MQJMS 格式。

有效的空 serveraddress。在此情況下，XMS 會假設值是字串 "localhost"。

範例清單如下:

```
myuser/mypassword@ldap://server1.mycom.com:389
ldap://server1.mycom.com
ldap://
ldap://:389
```

僅適用於 .NET : 從 IBM MQ 8.0, 與 IBM MQ (WMQ_CM_CLIENT) 的受管理連線及與 IBM MQ (WMQ_CM_CLIENT_UNMANAGED) 的未受管理連線都支援 TLS/SSL 連線。

依預設, 不會設定內容。

相關資訊

[未受管理 .NET 用戶端的 SSL 及 TLS 支援](#)

[受管理 .NET 用戶端的 SSL 及 TLS 支援](#)

XMSC_WMQ_SSL_CIPHER_SPEC

資料類型:

字串

下列項目的內容:

ConnectionFactory

建立佇列管理程式的安全連線時所使用的 CipherSpec 名稱。

下表列出可與 IBM WebSphere MQ SSL 及 TLS 支援搭配使用的密碼規格。當您要求個人憑證時, 要指定公開與私密金鑰組之金鑰大小。SSL 信號交換期間使用的金鑰大小是儲存在憑證中的大小, 除非如表格中所述由 CipherSpec 決定。依預設, 不會設定此內容。

CipherSpec 名稱	使用的通訊協定	雜湊演算法	加密演算法	加密位元	FIPS ¹	套組 B 128 位元	套組 B 192 位元
NULL_MD5	SSL 3.0	MD5	無	0	否	否	否
NULL_SHA	SSL 3.0	SHA-1	無	0	否	否	否
RC4_MD5_EXPORT ²	SSL 3.0	MD5	RC4	40	否	否	否
RC4_MD5_US	SSL 3.0	MD5	RC4	128	否	否	否
RC4_SHA_US	SSL 3.0	SHA-1	RC4	128	否	否	否
RC2_MD5_EXPORT ²	SSL 3.0	MD5	RC2	40	否	否	否
DES_SHA_EXPORT ²	SSL 3.0	SHA-1	DES	56	否	否	否
RC4_56_SHA_EXPORT1024 ³	SSL 3.0	SHA-1	RC4	56	否	否	否
DES_SHA_EXPORT1024 ³	SSL 3.0	SHA-1	DES	56	否	否	否
TRIPLE_DES_SHA_US	SSL 3.0	SHA-1	3DES	168	否	否	否
TLS_RSA_WITH_AES_128_CBC_SHA	TLS 1.0	SHA-1	AES	128	是	否	否
TLS_RSA_WITH_AES_256_CBC_SHA ⁴	TLS 1.0	SHA-1	AES	256	是	否	否
TLS_RSA_WITH_DES_CBC_SHA	TLS 1.0	SHA-1	DES	56	否 ⁵	否	否
TLS_RSA_WITH_3DES_EDE_CBC_SHA ⁸	TLS 1.0	SHA-1	3DES	168	是	否	否
FIPS_WITH_DES_CBC_SHA	SSL 3.0	SHA-1	DES	56	否 ⁶	否	否
FIPS_WITH_3DES_EDE_CBC_SHA	SSL 3.0	SHA-1	3DES	168	否 ⁷	否	否
TLS_RSA_WITH_AES_128_GCM_SHA256 ⁶	TLS 1.2	SHA-256	AES	128	是	否	否

CipherSpec 名稱	使用的通訊協定	雜湊演算法	加密演算法	加密位元	FIPS ¹	套組 B 128 位元	套組 B 192 位元
TLS_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	SHA-384	AES	256	是	否	否
TLS_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	SHA-256	AES	128	是	否	否
TLS_RSA_WITH_AES_256_CBC_SHA256	TLS 1.2	SHA-256	AES	256	是	否	否
ECDHE_ECDSA_RC4_128_SHA256	TLS 1.2	SHA-256	RC4	128	否	否	否
ECDHE_ECDSA_3DES_EDE_CBC_SHA256	TLS 1.2	SHA-256	3DES	168	是	否	否
ECDHE_RSA_RC4_128_SHA256	TLS 1.2	SHA-256	RC4	128	否	否	否
ECDHE_RSA_3DES_EDE_CBC_SHA256	TLS 1.2	SHA-256	3DES	168	是	否	否
ECDHE_ECDSA_AES_128_CBC_SHA256	TLS 1.2	SHA-256	AES	128	是	否	否
ECDHE_ECDSA_AES_256_CBC_SHA384	TLS 1.2	SHA-384	AES	256	是	否	否
ECDHE_RSA_AES_128_CBC_SHA256	TLS 1.2	SHA-256	AES	128	是	否	否
ECDHE_RSA_AES_256_CBC_SHA384	TLS 1.2	SHA-384	AES	256	是	否	否
ECDHE_ECDSA_AES_128_GCM_SHA256	TLS 1.2	SHA-256	AES	128	是	是	否
ECDHE_ECDSA_AES_256_GCM_SHA384	TLS 1.2	SHA-384	AES	256	是	否	是
ECDHE_RSA_AES_128_GCM_SHA256	TLS 1.2	SHA-256	AES	128	是	否	否
ECDHE_RSA_AES_256_GCM_SHA384	TLS 1.2	SHA-384	AES	256	是	否	否
TLS_RSA_WITH_NULL_SHA256	TLS 1.2	SHA-256	無	0	否	否	否
ECDHE_RSA_NULL_SHA256	TLS 1.2	SHA-256	無	0	否	否	否
ECDHE_ECDSA_NULL_SHA256	TLS 1.2	SHA-256	無	0	否	否	否
TLS_RSA_WITH_NULL_NULL	TLS 1.2	無	無	0	否	否	否
TLS_RSA_WITH_RC4_128_SHA256	TLS 1.2	SHA-256	RC4	128	否	否	否

CipherSpec 名稱	使用的通訊協定	雜湊演算法	加密演算法	加密位元	FIPS ¹	套組 B 128 位元	套組 B 192 位元
<p>附註:</p> <ol style="list-style-type: none"> 指定 CipherSpec 是否符合「聯邦資訊存取安全標準 (FIPS) 140-2」。如需 FIPS 的說明，以及如何針對符合 FIPS 140-2 標準的作業配置 WebSphere MQ 的相關資訊，請參閱線上 IBM WebSphere MQ 產品說明文件中的聯邦資訊存取安全標準 (FIPS)。 信號交換金鑰大小上限是 512 位元。如果在 SSL 信號交換期間交換的兩個憑證中有一個金鑰大小超出 512 位元，則在信號交換期間會產生一個臨時的 512 位元金鑰以供使用。 信號交換金鑰大小是 1024 位元。 這個 CipherSpec 無法用來保護從「WebSphere MQ 探險家」到佇列管理程式的連線，除非將適當的未限定原則檔套用至「探險家」所使用的 JRE。 在 2007 年 5 月 19 日之前，這個 CipherSpec 已經過 FIPS 140-2 認證。 在 2007 年 5 月 19 日之前，這個 CipherSpec 已經過 FIPS 140-2 認證。FIPS_WITH_DES_CBC_SHA 名稱是歷史名稱，反映出這個 CipherSpec 先前（但已不再）符合 FIPS 標準的事實。此 CipherSpec 已淘汰。 FIPS_WITH_3DES_EDE_CBC_SHA 名稱是歷史名稱，反映出這個 CipherSpec 先前（但已不再）符合 FIPS 標準的事實。這個 CipherSpec 的用法已淘汰。 為符合 FIPS 140-2 標準的作業配置 WebSphere MQ 時，此 CipherSpec 可用於在連線終止之前傳送最多 32 GB 資料，並發生錯誤 AMQ9288。若要避免此錯誤，請避免使用三重 DES 演算法 (已淘汰)，或在 FIPS 140-2 配置中使用此 CipherSpec 時啟用秘密金鑰重設。 							

相關資訊

安全

訊息的資料完整性

指定 CipherSpecs

XMSC_WMQ_SSL_CIPHER_SUITE

資料類型:

字串

下列項目的內容:

ConnectionFactory

要在與佇列管理程式的 SSL 或 TLS 連線上使用的 CipherSuite 名稱。在協議安全連線時使用的通訊協定取決於指定的 CipherSuite。

此內容具有下列標準值:

- SSL_RSA_WITH_DES_CBC_SHA
- SSL_RSA_EXPORT1024_WITH_DES_CBC_SHA
- SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA
- SSL_RSA_FIPS_WITH_DES_CBC_SHA
- SSL_RSA_WITH_NULL_MD5
- SSL_RSA_WITH_NULL_SHA
- SSL_RSA_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

可以提供此值作為 `XMSC_WMQ_SSL_CIPHER_SPEC` 的替代方案。

如果為 `XMSC_WMQ_SSL_CIPHER_SPEC` 指定非空白值，則此值會置換 `XMSC_WMQ_SSL_CIPHER_SUITE` 的設定。如果 `XMSC_WMQ_SSL_CIPHER_SPEC` 沒有值，則會使用 `XMSC_WMQ_SSL_CIPHER_SUITE` 值作為提供給 GSKit 的密碼組合。在此情況下，會將值對映至對等的 CipherSpec 值，如第 56 頁的『CipherSuite 及 CipherSpec 名稱對映，用於連接至 WebSphere MQ 佇列管理程式』中所述。

如果 `XMSC_WMQ_SSL_CIPHER_SPEC` 和 `XMSC_WMQ_SSL_CIPHER_SUITE` 都是空的，則 `pChDef->SSLCipherSpec` 欄位會填入空格。

僅適用於 .NET：從 IBM MQ 8.0，與 IBM MQ (WMQ_CM_CLIENT) 的受管理連線及與 IBM MQ (WMQ_CM_CLIENT_UNMANAGED) 的未受管理連線都支援 TLS/SSL 連線。

依預設，不會設定內容。

相關資訊

[未受管理 .NET 用戶端的 SSL 及 TLS 支援](#)

[受管理 .NET 用戶端的 SSL 及 TLS 支援](#)

XMSC_WMQ_SSL_CRYPTO_HW

資料類型：

字串

下列項目的內容：

ConnectionFactory

用戶端系統所連接的加密硬體的配置詳細資料。

此內容具有下列標準值：

- GSK_ACCELERATOR_RAINBOW_CS_OFF
- GSK_ACCELERATOR_RAINBOW_CS_ON
- GSK_ACCELERATOR_NCIPHER_NF_OFF
- GSK_ACCELERATOR_NCIPHER_NF_ON

PKCS11 加密硬體有特殊格式 (其中 DriverPath、TokenLabel 和 TokenPassword 是使用者指定的字串)：

```
GSK_PKCS11=PKCS#11 DriverPath; PKCS#11 TokenLabel;PKCS#11 TokenPassword
```

XMS 不會解譯或變更字串的內容。它會將提供的值 (最多 256 個單位元組字元) 複製到 `MQSCO.CryptoHardware` 欄位。

僅適用於 .NET：從 IBM MQ 8.0，與 IBM MQ (WMQ_CM_CLIENT) 的受管理連線及與 IBM MQ (WMQ_CM_CLIENT_UNMANAGED) 的未受管理連線都支援 TLS/SSL 連線。

依預設，不會設定內容。

相關資訊

[未受管理 .NET 用戶端的 SSL 及 TLS 支援](#)

[受管理 .NET 用戶端的 SSL 及 TLS 支援](#)

XMSC_WMQ_SSL_FIPS_REQUIRED

資料類型：

布林

下列項目的內容：

ConnectionFactory

此內容的值決定應用程式是否可以使用不符合 FIPS 標準的密碼組合。如果此內容設為 true，則只將 FIPS 演算法用於用戶端/伺服器連線。

此內容可以具有下列值，這會轉換為 MQSCO.FipsRequired:

表 35: MQSCO.FlipsRequired 內容		
值	說明	MQSCO.FipsRequired
false	可以使用任何 CipherSpec。	MQSSL_FIPS_NO (預設值)
true	在套用至此用戶端連線的 CipherSpec 中，只能使用 FIPS 認證的加密演算法。	MQSSL_FIPS_YES

XMS 會將相關值複製到 MQSCO.FipsRequired。

MQSCO.FipsRequired 參數只能從 WebSphere MQ 第 6 版開始使用。若為 WebSphere MQ 5.3 版，如果設定此內容，則 XMS 不會嘗試建立與佇列管理程式的連線，而是會擲出適當的異常狀況。

僅適用於 .NET : 從 IBM MQ 8.0，與 IBM MQ (WMQ_CM_CLIENT) 的受管理連線及與 IBM MQ (WMQ_CM_CLIENT_UNMANAGED) 的未受管理連線都支援 TLS/SSL 連線。

相關資訊

[未受管理 .NET 用戶端的 SSL 及 TLS 支援](#)

[受管理 .NET 用戶端的 SSL 及 TLS 支援](#)

XMSC_WMQ_SSL_KEY_REPOSITORY

資料類型:

字串

下列項目的內容:

ConnectionFactory

在其中儲存金鑰和憑證的金鑰資料庫檔的位置。

XMS 會將字串 (最多 256 個單位元組字元) 複製到 MQSCO.KeyRepository 欄位。WebSphere MQ 會將此字串解譯為檔名，包括完整路徑。

僅適用於 .NET : 從 IBM MQ 8.0，與 IBM MQ (WMQ_CM_CLIENT) 的受管理連線及與 IBM MQ (WMQ_CM_CLIENT_UNMANAGED) 的未受管理連線都支援 TLS/SSL 連線。

依預設，不會設定內容。

相關資訊

[未受管理 .NET 用戶端的 SSL 及 TLS 支援](#)

[受管理 .NET 用戶端的 SSL 及 TLS 支援](#)

XMSC_WMQ_SSL_KEY_RESETCOUNT

資料類型:

System.Int32

下列項目的內容:

ConnectionFactory

KeyResetCount 代表在重新協議秘密金鑰之前 SSL 交談中傳送和收到的未加密位元組總數。位元組數包括 MCA 所傳送的控制資訊。

XMS 會將您為此內容提供的值複製到 MQSCO.KeyResetCount。

參數 MQSCO.KeyRestCount 只能從 WebSphere MQ 第 6 版開始使用。若為 WebSphere MQ 5.3 版，如果設定此內容，則 XMS 不會嘗試建立與佇列管理程式的連線，而是會擲出適當的異常狀況。

僅適用於 .NET : 從 IBM MQ 8.0，與 IBM MQ (WMQ_CM_CLIENT) 的受管理連線及與 IBM MQ (WMQ_CM_CLIENT_UNMANAGED) 的未受管理連線都支援 TLS/SSL 連線。

此內容的預設值為零，表示永不重新協議秘密金鑰。

相關資訊

[未受管理 .NET 用戶端的 SSL 及 TLS 支援](#)

[受管理 .NET 用戶端的 SSL 及 TLS 支援](#)

XMSC_WMQ_SSL_PEER_NAME

資料類型：

字串

下列項目的內容：

ConnectionFactory

建立佇列管理程式的 SSL 連線時所使用的對等節點名稱。

此內容沒有標準值清單。相反地，您必須根據 SSLPEER 的規則來建置此字串。

對等名稱的範例如下：

```
"CN=John Smith, O=IBM ,OU=Test , C=GB"
```

在呼叫 MQCONNX 之前，XMS 會將字串複製到正確的單位元組字碼頁，並將正確的值放入 MQCD.SSLPeerNamePtr 和 MQCD.SSLPeerNameLength。

僅當應用程式以用戶端模式連接至佇列管理程式時，此內容才相關。

僅適用於 .NET：從 IBM MQ 8.0，與 IBM MQ (WMQ_CM_CLIENT) 的受管理連線及與 IBM MQ (WMQ_CM_CLIENT_UNMANAGED) 的未受管理連線都支援 TLS/SSL 連線。

依預設，不會設定內容。

相關資訊

[未受管理 .NET 用戶端的 SSL 及 TLS 支援](#)

[受管理 .NET 用戶端的 SSL 及 TLS 支援](#)

[SSLPEERNAME](#)

XMSC_WMQ_SYNCPOINT_ALL_GETS

資料類型：

System.Boolean

下列項目的內容：

ConnectionFactory

是否必須在同步點控制之下從佇列中擷取所有訊息。

內容的有效值如下：

有效值

false

true

意義

當情況適當時，XMS 用戶端可以從同步點控制之外的佇列擷取訊息。

XMS 用戶端必須從同步點控制內的佇列中擷取所有訊息。

預設值是 false。

XMSC_WMQ_TARGET_CLIENT

資料類型：

System.Int32

下列項目的內容：

目的地

URI 中使用的名稱:

targetClient

傳送至目的地的訊息是否包含 MQRFH2 標頭。

如果應用程式傳送包含 MQRFH2 標頭的訊息，則接收端應用程式必須能夠處理標頭。

內容的有效值如下:

有效值	意義
XMSC_WMQ_TARGET_DEST_JMS	傳送至目的地的訊息包含 MQRFH2 標頭。如果應用程式將訊息傳送至另一個 XMS 應用程式、WebSphere JMS 應用程式或設計用來處理 MQRFH2 標頭的原生 WebSphere MQ 應用程式，請指定此值。
XMSC_WMQ_TARGET_DEST_MQ	傳送至目的地的訊息不包含 MQRFH2 標頭。如果應用程式將訊息傳送至未設計用來處理 MQRFH2 標頭的原生 WebSphere MQ 應用程式，請指定此值。

預設值為 XMSC_WMQ_TARGET_DEST_JMS。

XMSC_WMQ_TEMP_Q_PREFIX**資料類型:**

字串

下列項目的內容:

ConnectionFactory

用來構成 WebSphere MQ 動態佇列 (在應用程式建立 XMS 暫時佇列時建立) 名稱的字首。

構成字首的規則與構物件描述子中 **DynamicQName** 欄位內容的規則相同，但最後一個非空白字元必須是星號 (*)。如果未設定此內容，則在 z/OS 上使用的值為 CSQ.*，在其他平台上使用的值為 AMQ.*。依預設，不會設定內容。

此內容僅在 點對點 (point-to-point) 網域中相關。

XMSC_WMQ_TEMP_TOPIC_PREFIX**資料類型:**

字串

下列項目的內容:

ConnectionFactory, 目的地

建立暫時主題時，XMS 會產生格式為 "TEMP/TEMPTOPICPREFIX/unique_id" 的主題字串，或者如果此內容包含預設值，則會產生此字串 "TEMP/unique_id"。如果指定非空白值，則容許定義特定模型佇列，以給訂閱者建立根據此連線建立的暫時主題的受管理佇列。

任何僅包含 IBM WebSphere MQ 主題字串有效字元的非空值字串，都是此內容的有效值。

依預設，此內容設為 "" (空字串)。

註: 此內容僅在發佈/訂閱網域中相關。

XMSC_WMQ_TEMPORARY_MODEL**資料類型:**

字串

下列項目的內容:

ConnectionFactory

當應用程式建立 XMS 暫時佇列時，從中建立動態佇列的 WebSphere MQ 模型佇列名稱。

此內容的預設值為 SYSTEM.DEFAULT.MODEL.QUEUE。

此內容僅在 點對點 (point-to-point) 網域中相關。

XMSC_WMQ_WILDCARD_FORMAT

資料類型:

System.Int32

下列項目的內容:

ConnectionFactory, 目的地

此內容決定要使用的萬用字元語法版本。

搭配使用發佈/訂閱與 IBM WebSphere MQ '*' 及 '?' 時視為萬用字元。而當搭配使用發佈訂閱與 IBM Integration Bus 時, 會將 '#' 和 '+' 視為萬用字元。這個內容會取代 XMSC_WMQ_BROKER_VERSION 內容。

此內容的有效值如下:

XMSC_WMQ_WILDCARD_TOPIC_ONLY

僅辨識主題層次萬用字元, 亦即 '#' 及 '+' 被視為萬用字元。此值與 XMSC_WMQ_BROKER_V2 相同。

XMSC_WMQ_WILDCARD_CHAR_ONLY

僅辨識字元萬用字元, 亦即 '*' 及 '?' 視為萬用字元。此值與 XMSC_WMQ_BROKER_V1 相同。

依預設, 此內容設為 XMSC_WMQ_WILDCARD_TOPIC_ONLY。

XMSC_WPM_BUS_NAME

資料類型:

字串

下列項目的內容:

ConnectionFactory 和目的地

URI 中使用的名稱:

busName

對於 Connection Factory, 則應用程式所連接的服務整合匯流排的名稱, 或者對於目的地, 則為目的地所在的服務整合匯流排的名稱。

如果目的地是主題, 這個內容是相關聯主題空間所在的服務整合匯流排名稱。此主題空間由 XMSC_WPM_TOPIC_SPACE 內容指定。

如果未設定目的地的內容, 則會假設佇列或相關聯的主題空間存在於應用程式所連接的服務整合匯流排中。

依預設, 不會設定內容。

XMSC_WPM_CONNECTION_XX_ENCODE_CASE_ONE Protocol

資料類型:

System.Int32

下列項目的內容:

連線

建立傳訊引擎連線時使用的通訊協定。這是唯讀內容。

內容的可能值如下:

值	意義
XMSC_WPM_CP_HTTP	連線使用透過 TCP/IP 的 HTTP。
XMSC_WPM_CP_TCP	連線使用 TCP/IP。

XMSC_WPM_CONNECTION_PROXIMITY

資料類型:

System.Int32

下列項目的內容:

ConnectionFactory

連線的連線近似性設定。這個內容決定應用程式連接的傳訊引擎必須接近引導伺服器的程度。

內容的有效值如下：

有效值	連線近似性設定
XMSC_WPM_CONNECTION_PROXIMITY_BUS	匯流排
XMSC_WPM_CONNECTION_PROXIMITY_CLUSTER	叢集
XMSC_WPM_CONNECTION_PROXIMITY_HOST	主機
XMSC_WPM_CONNECTION_PROXIMITY_SERVER	伺服器

預設值為 XMSC_WPM_CONNECTION_PROXIMITY_BUS。

XMSC_WPM_DUR_SUB_HOME

資料類型：

字串

下列項目的內容：

ConnectionFactory

URI 中使用的名稱：

durableSubscription 首頁

在其中管理連線或目的地的所有可延續訂閱的傳訊引擎的名稱。要遞送至可延續訂閱者的訊息會儲存在相同傳訊引擎的發佈點。

必須先指定連線的可延續訂閱起始目錄，應用程式才能建立使用連線的可延續訂閱者。指定給目的地的任何值都會置換指定給連線的值。

依預設，不會設定內容。

此內容僅在 發佈/訂閱 網域中相關。

XMSC_WPM_HOST_NAME

資料類型：

字串

下列項目的內容：

連線

包含應用程式所連接傳訊引擎的系統的主機名稱或 IP 位址。這是唯讀內容。

XMSC_WPM_LOCAL_ADDRESS

資料類型：

字串

下列項目的內容：

ConnectionFactory

為了連線至服務整合匯流排，此內容會指定要使用的本端網路介面和/或要使用的本端埠（或本端埠的範圍）。

內容的值是具下列格式的字串：

[*host_name*] [(*low_port*) [, *high_port*]]

變數的意義如下：

host_name

要用於連線之本端網路介面的主機名稱或 IP 位址。

只有在執行應用程式的系統具有兩個以上網路介面，且您需要能夠指定連線必須使用哪個介面時，才需要提供此資訊。如果系統只有一個網路介面，則只能使用該介面。如果系統有兩個以上網路介面，且您未指定必須使用哪個介面，則會隨機選取該介面。

low_port

要用於連線的本端埠號。

如果也指定 *high_port*，則會解譯埠號範圍中的最低埠號 *low_port*。

高埠

埠號範圍中的最高埠號。指定範圍內的其中一個埠必須用於連線。

以下是內容有效值的一些範例：

木星
9.20.4.98
JUPITER (1000)
9.20.4.98(1000,2000)
(1000)
(1000,2000)

依預設，不會設定內容。

XMSC_WPM_ME_NAME

資料類型：

字串

下列項目的內容：

連線

應用程式所連接的傳訊引擎的名稱。這是唯讀內容。

XMSC_WPM_NON_PERSISTENT_MAP

資料類型：

System.Int32

下列項目的內容：

ConnectionFactory

使用連線傳送的非持續訊息的可靠性層次。

內容的有效值如下：

有效值

XMSC_WPM_MAPPING_AS_DESTINATION

XMSC_WPM_MAPPING_BEST_EFFORT_NON_持續

XMSC_WPM_MAPPING_EXPRESS_NON_持續

XMSC_WPM_MAPPING_RELIABLE_NON_持續

XMSC_WPM_MAPPING_RELIABLE_PERSISTENT

XMSC_WPM_MAPPING_ASSURED_PERSISTENT

可靠性層次

由服務整合匯流排中指定給佇列或主題空間的預設可靠性層次來決定
最大努力非持續性

快速非持續性

可靠非持續性

可靠持續性

保證持續性

預設值為 XMSC_WPM_MAPPING_EXPRESS_NON_PERSISTENT。

XMSC_WPM_PERSISTENT_MAP

資料類型:

System.Int32

下列項目的內容:

ConnectionFactory

使用連線傳送的持續訊息的可靠性層次。

內容的有效值如下:

有效值

XMSC_WPM_MAPPING_AS_DESTINATION

XMSC_WPM_MAPPING_BEST_EFFORT_NON_持續

XMSC_WPM_MAPPING_EXPRESS_NON_持續

XMSC_WPM_MAPPING_RELIABLE_NON_持續

XMSC_WPM_MAPPING_RELIABLE_PERSISTENT

XMSC_WPM_MAPPING_ASSURED_PERSISTENT

可靠性層次

由服務整合匯流排中指定給佇列或主題空間的預設可靠性層次來決定

最大努力非持續性

快速非持續性

可靠非持續性

可靠持續性

保證持續性

預設值為 XMSC_WPM_MAPPING_RELIABLE_PERSISTENT。

XMSC_WPM_PORT

資料類型:

System.Int32

下列項目的內容:

連線

應用程式所連接的傳訊引擎接聽的埠號。這是唯讀內容。

XMSC_WPM_PROVIDER_ENDPOINTS

資料類型:

字串

下列項目的內容:

ConnectionFactory

引導伺服器的一個以上端點位址的序列。端點位址以逗點區隔。

引導伺服器是一部應用程式伺服器，負責選取應用程式所連接的傳訊引擎。引導伺服器的端點位址具有下列格式:

host_name:port_number:chain_name

端點位址的元件意義如下:

host_name

引導伺服器所在系統的主機名稱或 IP 位址。如果未指定主機名稱或 IP 位址，則預設值為 localhost。

port_number

引導伺服器用來接聽送入要求的埠號。如果未指定埠號，則預設值為 7276。

鏈名稱

引導伺服器使用的引導傳輸鏈名稱。有效值如下所示：

有效值	引導傳輸鏈的名稱
XMSC_WPM_BOOTSTRAP_HTTP	BootstrapTunneled 傳訊
XMSC_WPM_BOOTSTRAP_HTTPS	BootstrapTunneledSecureMessaging
XMSC_WPM_BOOTSTRAP_SSL	BootstrapSecure 傳訊
XMSC_WPM_BOOTSTRAP_TCP	BootstrapBasic 傳訊

如果未指定名稱，則預設值為 XMSC_WPM_BOOTSTRAP_TCP。

如果未指定端點位址，預設值為 localhost:7276:BootstrapBasicMessaging。

XMSC_WPM_SSL_CIPHER_SUITE

資料類型：

字串

下列項目的內容：

ConnectionFactory

要在與 WebSphere 服務 Integration Bus 傳訊引擎的 SSL 或 TLS 連線上使用的 CipherSuite 名稱。在協議安全連線時使用的通訊協定取決於指定的 CipherSuite。

密碼組合	使用的通訊協定
SSL_RSA_WITH_NULL_MD5	SSLv3
SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5	SSLv3
SSL_RSA_EXPORT_WITH_RC4_40_MD5	SSLv3
SSL_RSA_WITH_RC4_128_MD5	SSLv3
SSL_RSA_WITH_NULL_SHA	SSLv3
SSL_RSA_EXPORT1024_WITH_RC4_56_SHA	SSLv3
SSL_RSA_WITH_RC4_128_SHA	SSLv3
SSL_RSA_WITH_DES_CBC_SHA	SSLv3
SSL_RSA_EXPORT1024_WITH_DES_CBC_SHA	SSLv3
SSL_RSA_FIPS_WITH_DES_CBC_SHA	SSLv3
SSL_RSA_WITH_3DES_EDE_CBC_SHA	SSLv3
SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA	SSLv3
TLS_RSA_WITH_DES_CBC_SHA	TLSv1
TLS_RSA_WITH_3DES_EDE_CBC_SHA (已淘汰)	TLSv1
TLS_RSA_WITH_AES_128_CBC_SHA	TLSv1
TLS_RSA_WITH_AES_256_CBC_SHA	TLSv1

註: TLS_RSA_WITH_AES_128_CBC_SHA 和 TLS_RSA_WITH_AES_256_CBC_SHA CipherSuites 僅在 Windows 或 Solaris 上受支援。(這由 GSKit 指定。)

此內容沒有預設值。如果您想要使用 SSL 或 TLS，則必須指定此內容的值，否則您的應用程式無法順利連接至伺服器。

XMSC_WPM_SSL_FIPS_REQUIRED

資料類型：

布林

下列項目的內容：

ConnectionFactory

此內容的值決定應用程式是否可以使用不符合 FIPS 標準的密碼組合。如果此內容設為 true，則只會將 FIPS 演算法用於主從架構連線。將此內容的值設為 TRUE 會防止應用程式使用不符合 FIPS 標準的密碼組合。

依預設，此內容設為 FALSE (即關閉 FIPS 模式)。

XMSC_WPM_SSL_KEY_REPOSITORY

資料類型：

字串

下列項目的內容：

ConnectionFactory

檔案的路徑，該檔案是包含要在安全連線中使用的公開或私密金鑰的金鑰環檔案。

將金鑰環檔案內容設為特殊值 XMSC_WPM_SSL_MS_CERTIFICATE_STORE 會指定使用 Microsoft Windows 金鑰資料庫。使用 Microsoft Windows 金鑰資料庫 (可在 **控制台 > 網際網路選項 > 內容 > 憑證** 下找到)，不需要個別金鑰檔資料庫。不允許在 Windows x64 及其他平台上使用此常數。

依預設，不會設定內容。

XMSC_WPM_SSL_KEYRING_LABEL

資料類型：

字串

下列項目的內容：

ConnectionFactory

向伺服器鑑別時使用的憑證。如果未指定任何值，則會使用預設憑證。

依預設，不會設定內容。

XMSC_WPM_SSL_KEYRING_PW

資料類型：

字串

下列項目的內容：

ConnectionFactory

金鑰環檔案的密碼。

此內容可用來替代使用 [XMSC_WPM_SSL_KEYRING_STASH_FILE](#) 來配置金鑰環檔案的密碼。

依預設，不會設定內容。

XMSC_WPM_SSL_KEYRING_STASH_FILE

資料類型：

字串

下列項目的內容：

ConnectionFactory

包含金鑰儲存庫檔密碼的二進位檔名稱。

此內容可用來替代使用 [XMSC_WPM_SSL_KEYRING_PW](#) 來配置金鑰環檔案的密碼。

依預設，不會設定內容。

XMSC_WPM_TARGET_GROUP

資料類型：

字串

下列項目的內容：

ConnectionFactory

傳訊引擎的目標群組的名稱。目標群組的本質由 `XMSC_WPM_TARGET_TYPE` 內容決定。

如果您想將傳訊引擎的搜尋限制在服務整合匯流排中傳訊引擎的子群組，請設定這個內容。如果您希望應用程式能夠連接至服務整合匯流排中的任何傳訊引擎，請勿設定這個內容。

依預設，不會設定內容。

XMSC_WPM_TARGET_SIGNIFICANCE

資料類型：

System.Int32

下列項目的內容：

ConnectionFactory

傳訊引擎的目標群組的重要性。

內容的有效值如下：

有效值

XMSC_WPM_TARGET_顯著性_ 偏好

XMSC_WPM_TARGET_顯著性_ 必要

意義

如果有可用的傳訊引擎，則會選取目標群組中的傳訊引擎。否則，如果傳訊引擎位於相同的服務整合匯流排中，則會選取目標群組之外的傳訊引擎。

選取的傳訊引擎必須在目標群組中。如果目標群組中的傳訊引擎無法使用，連線程序會失敗。

內容的預設值為 `XMSC_WPM_TARGET_SIGNIFICANCE_PREFERRED`。

XMSC_WPM_TARGET_TRANSPORT_CHAIN

資料類型：

字串

下列項目的內容：

ConnectionFactory

應用程式在連接傳訊引擎時必須使用的入埠傳輸鏈的名稱。

這個內容的值可以是管理傳訊引擎之應用程式伺服器中可用的任何入埠傳輸鏈名稱。下列具名常數提供給其中一個預先定義的入埠傳輸鏈：

已命名的常數

XMSC_WPM_TARGET_TRANSPORT_CHAIN_BASIC

傳輸鏈的名稱

InboundBasic 傳訊

此內容的預設值為 `XMSC_WPM_TARGET_TRANSPORT_CHAIN_BASIC`。

XMSC_WPM_TARGET_TYPE

資料類型：

System.Int32

下列項目的內容：

ConnectionFactory

傳訊引擎的目標群組的類型。此內容決定 `XMSC_WPM_TARGET_GROUP` 內容所識別目標群組的本質。

內容的有效值如下：

有效值

XMSC_WPM_TARGET_TYPE_BUSMEMBER

XMSC_WPM_TARGET_TYPE_CUSTOM

XMSC_WPM_TARGET_TYPE_ME

意義

目標群組的名稱是匯流排成員的名稱。目標群組是匯流排成員中的所有傳訊引擎。

目標群組的名稱是使用者定義的傳訊引擎群組名稱。目標群組是向使用者定義群組登錄的所有傳訊引擎。

目標群組的名稱是傳訊引擎的名稱。目標群組是指定的傳訊引擎。

依預設，不會設定內容。

XMSC_WPM_TEMP_Q_PREFIX

資料類型：

字串

下列項目的內容：

ConnectionFactory

當應用程式建立 XMS 暫時佇列時，用來構成服務整合匯流排中所建立之暫時佇列名稱的字首。字首最多可以包含 12 個字元。

暫時佇列的名稱以字元 "_Q" 開頭，後面接著字首。名稱的其餘部分由系統產生的字元組成。

依預設，不會設定此內容，這表示暫時佇列的名稱沒有字首。

此內容僅在 點對點 (point-to-point) 網域中相關。

XMSC_WPM_TEMP_TOPIC_PREFIX

資料類型：

字串

下列項目的內容：

ConnectionFactory

用來構成應用程式所建立暫時主題的名稱的字首。字首最多可以包含 12 個字元。

暫時主題的名稱以字元 "_T" 開頭，後面接著字首。名稱的其餘部分由系統產生的字元組成。

依預設，不會設定內容，這表示暫時主題的名稱沒有字首。

此內容僅在 發佈/訂閱 網域中相關。

XMSC_WPM_TOPIC_SPACE

資料類型：

字串

下列項目的內容：

目的地

URI 中使用的名稱：

topicSpace

包含主題的主題空間名稱。只有主題的目的地可以有這個內容。

依預設，不會設定內容，這表示會假設預設主題空間。

此內容僅在 發佈/訂閱 網域中相關。

注意事項

本資訊係針對 IBM 在美國所提供之產品與服務所開發。

在其他國家中，IBM 可能不會提供本書中所提的各項產品、服務或功能。請洽當地 IBM 業務代表，以取得當地目前提供的產品和服務之相關資訊。這份文件在提及 IBM 的產品、程式或服務時，不表示或暗示只能使用 IBM 的產品、程式或服務。只要未侵犯 IBM 的智慧財產權，任何功能相當的產品、程式或服務都可以取代 IBM 的產品、程式或服務。不過，任何非 IBM 的產品、程式或服務，使用者必須自行負責作業的評估和驗證責任。

本文件所說明之主題內容，IBM 可能擁有其專利或專利申請案。提供本文件不代表提供這些專利的授權。您可以書面提出授權查詢，來函請寄到：

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

如果是有關雙位元組 (DBCS) 資訊的授權查詢，請洽詢所在國的 IBM 智慧財產部門，或書面提出授權查詢，來函請寄到：

智慧財產權授權
法務部與智慧財產權法律
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

下列段落不適用於英國，若與任何其他國家之法律條款抵觸，亦不適用於該國： International Business Machines Corporation 只依 "現況" 提供本出版品，不提供任何明示或默示之保證，其中包括且不限於不侵權、可商用性或特定目的之適用性的隱含保證。有些地區在特定交易上，不允許排除明示或暗示的保證，因此，這項聲明不一定適合您。

這項資訊中可能有技術上或排版印刷上的訛誤。因此，IBM 會定期修訂；並將修訂後的內容納入新版中。IBM 隨時會改進及/或變更本出版品所提及的產品及/或程式，不另行通知。

本資訊中任何對非 IBM 網站的敘述僅供參考，IBM 對該網站並不提供任何保證。這些網站所提供的資料不是 IBM 本產品的資料內容，如果要使用這些網站的資料，您必須自行承擔風險。

IBM 得以各種適當的方式使用或散布由您提供的任何資訊，無需對您負責。

如果本程式的獲授權人為了 (i) 在個別建立的程式和其他程式（包括本程式）之間交換資訊，以及 (ii) 相互使用所交換的資訊，因而需要相關的資訊，請洽詢：

IBM Corporation
軟體交互作業能力協調程式，部門 49XA
3605 公路 52 N
Rochester, MN 55901
U.S.A.

在適當條款與條件之下，包括某些情況下（支付費用），或可使用此類資訊。

IBM 基於雙方之 IBM 客戶合約、IBM 國際程式授權合約或任何同等合約之條款，提供本資訊所提及的授權程式與其所有適用的授權資料。

本文件中所含的任何效能資料都是在受管制的環境下判定。因此不同作業環境之下所得的結果，可能會有很大的差異。有些測定已在開發階段系統上做過，不過這並不保證在一般系統上會出現相同結果。甚至有部分的測量，是利用插補法而得的估計值，實際結果可能有所不同。本文件的使用者應驗證其特定環境適用的資料。

本文件所提及之非 IBM 產品資訊，取自產品的供應商，或其發佈的聲明或其他公開管道。IBM 並未測試過這些產品，也無法確認這些非 IBM 產品的執行效能、相容性或任何對產品的其他主張是否完全無誤。有關非 IBM 產品的性能問題應直接洽詢該產品供應商。

有關 IBM 未來方針或目的之所有聲明，僅代表 IBM 的目標與主旨，隨時可能變更或撤銷，不必另行通知。

這份資訊含有日常商業運作所用的資料和報告範例。為了要使它們儘可能完整，範例包括個人、公司、品牌和產品的名稱。這些名稱全屬虛構，如與實際公司的名稱和住址雷同，純屬巧合。

著作權授權：

本資訊含有原始語言之範例應用程式，用以說明各作業平台中之程式設計技術。您可以基於研發、使用、銷售或散布符合作業平台（撰寫範例程式的作業平台）之應用程式介面的應用程式等目的，以任何形式複製、修改及散布這些範例程式，而不必向 IBM 付費。這些範例並未在所有情況下完整測試。因此，IBM 不保證或暗示這些程式的可靠性、有用性或功能。

若貴客戶正在閱讀本項資訊的電子檔，可能不會有照片和彩色說明。

程式設計介面資訊

程式設計介面資訊 (如果有提供的話) 旨在協助您建立與此程式搭配使用的應用軟體。

本書包含預期程式設計介面的相關資訊，可讓客戶撰寫程式以取得 WebSphere MQ 的服務。

不過，本資訊也可能包含診斷、修正和調整資訊。提供診斷、修正和調整資訊，是要協助您進行應用軟體的除錯。

重要：請勿使用此診斷、修改及調整資訊作為程式設計介面，因為它可能會變更。

商標

IBM、IBM 標誌 ibm.com 是 IBM Corporation 在全球許多適用範圍的商標。IBM 商標的最新清單可在 Web 的 "Copyright and trademark information" www.ibm.com/legal/copytrade.shtml 中找到。其他產品和服務名稱，可能是 IBM 或其他公司的商標。

Microsoft 及 Windows 是 Microsoft Corporation 在美國及/或其他國家或地區的商標。

UNIX 是 The Open Group 在美國及/或其他國家/地區的註冊商標。

Linux® 是 Linus Torvalds 在美國及/或其他國家或地區的註冊商標。

本產品包含 Eclipse Project (<http://www.eclipse.org/>) 所開發的軟體。

Java 和所有以 Java 為基礎的商標及標誌是 Oracle 及/或其子公司的商標或註冊商標。



產品編號:

(1P) P/N: