

MQSeries® Everyplace



# 簡介

第 1 版



MQSeries® Everyplace



# 簡介

第 1 版

**請注意！**

在使用這份資訊及支援的產品之前，請務必先閱讀第61頁的『附錄. 注意事項』下的一般資訊。

**授權警告**

MQSeries Everyplace 第 1 版是一個工具箱，使用者可以利用它來撰寫 MQSeries Everyplace 應用程式，以及建立執行這些應用程式的環境。

這個工具箱的採購所遵循的授權條款決定了它所適用的環境：

如果採購 *MQSeries Everyplace* 是爲了當作一項裝置（從屬站），則不能利用它來建立 **MQSeries Everyplace 通道管理程式**或 **MQSeries Everyplace 通道接聽器**。

**MQSeries Everyplace 通道管理程式**或 **MQSeries Everyplace 通道接聽器**會定義**開道**（伺服器）環境，它需要開道授權。

第 1 版（2000 年 6 月）

這個修訂版適用於 MQSeries Everyplace 第 1.0 版及所有後續的版次和修正，直到新版本中另有說明爲止。

© Copyright International Business Machines Corporation 2000. All rights reserved.

# 目錄

圖 . . . . .	v	MQSeries Everyplace 本端安全 . . . . .	34
表 . . . . .	vii	MQSeries Everyplace 佇列型安全 . . . . .	34
關於本書 . . . . .	ix	訊息層次安全 . . . . .	35
本書適用對象 . . . . .	ix	登錄 . . . . .	36
必備知識 . . . . .	ix	MQSeries Everyplace 可鑑別實體 . . . . .	37
本書所用的詞彙 . . . . .	ix	專用登錄和憑證 . . . . .	37
第1章 總覽 . . . . .	1	自動登錄 . . . . .	37
第 1.0 版注意事項 . . . . .	1	公用登錄和憑證複製 . . . . .	38
第2章 必備條件 . . . . .	3	登錄服務的應用 . . . . .	38
第3章 MQSeries 系列 . . . . .	5	預設迷你憑證發行服務 . . . . .	38
第4章 基本需求 . . . . .	11	安全介面 . . . . .	39
功能 . . . . .	11	配置和自訂 . . . . .	39
應用程式 . . . . .	11	規則 . . . . .	39
客戶需求 . . . . .	12	連線樣式 . . . . .	41
第5章 產品概念 . . . . .	13	同層級對同層級連線 . . . . .	41
簡介 . . . . .	13	主從架構連線 . . . . .	42
訊息物件 . . . . .	14	多重連線樣式 . . . . .	42
傾出資料格式 . . . . .	17	類別 . . . . .	42
佇列 . . . . .	18	應用程式載入 . . . . .	43
佇列管理程式 . . . . .	21	第6章 MQSeries Everyplace 和 MQSeries	
佇列管理程式作業 . . . . .	23	網路 . . . . .	45
管理 . . . . .	25	MQSeries 介面 . . . . .	45
管理訊息 . . . . .	25	訊息轉換 . . . . .	52
選擇性管理 . . . . .	27	功能 . . . . .	54
監視和相關動作 . . . . .	27	相容性 . . . . .	54
動態通道 . . . . .	27	確定遞送 . . . . .	55
配接器 . . . . .	29	第7章 應用程式和公用程式 . . . . .	57
撥號連線管理 . . . . .	29	明信片 . . . . .	57
追蹤 . . . . .	29	MQSeries Everyplace 探險家 . . . . .	57
事件日誌 . . . . .	29	第8章 程式設計介面 . . . . .	59
MQSeries Everyplace 網路 . . . . .	30	附錄. 注意事項 . . . . .	61
配置和可調整性 . . . . .	30	商標 . . . . .	62
非同步訊息遞送 . . . . .	31	名詞解釋 . . . . .	63
同步訊息遞送 . . . . .	32	參考書目 . . . . .	67
安全 . . . . .	32	索引 . . . . .	69





1. MQSeries 系列 . . . . .	5	5. 簡式 MQSeries Everyplace 網路 . . . . .	30
2. 簡單的主電腦和分散式配置 . . . . .	6	6. 星形 MQSeries Everyplace 網路 . . . . .	31
3. 典型工作站配置 . . . . .	7	7. MQSeries 橋接物件階層 . . . . .	46
4. 典型裝置配置 . . . . .	8		





---

## 表

1. 第 1 版支援的軟體環境 . . . . .	3	12. 管理訊息的一般結構 . . . . .	26
2. MQSeries Everyplace 和 MQSeries 元素	8	13. 鑑別、加密和壓縮支援 . . . . .	33
3. 欄位物件及其構成欄位內容 . . . . .	15	14. 連線樣式 . . . . .	41
4. 屬性物件內容 . . . . .	15	15. 類別選項 . . . . .	42
5. 訊息物件內容 . . . . .	16	16. 橋接物件內容 . . . . .	46
6. 提供的訊息物件欄位 . . . . .	16	17. 橋接內容 . . . . .	48
7. 佇列內容 . . . . .	20	18. MQSeries 佇列管理程式 Proxy 內容	48
8. 本端佇列管理程式內容 . . . . .	23	19. 從屬站連線服務程式內容 . . . . .	49
9. 連線（遠端佇列管理程式）內容 . . . . .	23	20. 接聽器內容 . . . . .	50
10. MQSeries Everyplace 佇列中的傳訊作業	24	21. MQSeries 遠端佇列內容 . . . . .	51
11. 管理訊息類別 . . . . .	25		



---

## 關於本書

本書是 MQSeries Everyplace 的一般介紹。它涵蓋了產品概念以及與其他 MQSeries 產品的關係。

如果需要 MQSeries Everyplace API 及如何用它來建立 MQSeries Everyplace 應用程式的詳細資料，請參閱 *MQSeries Everyplace Programming Reference* 及 *MQSeries Everyplace Programming Guide*。

---

## 本書適用對象

本書適用於有意在感應器、電話、個人數位助理 (PDA) 和膝上型電腦等小型裝置上進行安全傳訊的任何人。

---

## 必備知識

閱讀這份資訊之前，不需要必備知識，不過，對於安全傳訊的概念，最好能先有些基礎瞭解。

如果您不具備這個基礎概念，請參閱下列 MQSeries 書籍，它們非常有用：

- *MQSeries An Introduction to Messaging and Queuing*
- *MQSeries for Windows NT® V5R1 快速入門*

您可以在線上 MQSeries 書庫的 Book 區段中，找到這些書籍的電子檔。請造訪 MQSeries 網站，網址如下：<http://www.ibm.com/software/ts/MQSeries/library/>

---

## 本書所用的詞彙

本書中會使用下列詞彙：

### **MQSeries 系列**

請參閱第5頁的『第3章 MQSeries 系列』中所說明的 MQSeries 產品集成。

### **MQSeries 傳訊**

請參閱第5頁的『第3章 MQSeries 系列』中所說明的四個傳訊產品群組。

### **MQSeries**

請參閱下面這三個 MQSeries 傳訊產品群組：

- 分散式傳訊
- 主電腦傳訊

- 工作站傳訊

### **MQSeries Everyplace**

請參閱第四個 MQSeries 傳訊產品群組，通用傳訊。

**裝置** 這是一部執行 MQSeries Everyplace 程式、但沒有安裝 **MQSeries Everyplace 通道管理程式**或 **MQSeries Everyplace 通道接聽器**物件的任何大小的電腦。

**註:** 就授權目的而言，裝置與 *MQSeries Everyplace 從屬站*是同義字。

**開道** 這是一部執行 MQSeries Everyplace 程式且安裝了 **MQSeries Everyplace 通道管理程式**或 **MQSeries Everyplace 通道接聽器**物件的任何大小的電腦。

**註:** 就授權目的而言，開道與 *MQSeries Everyplace 伺服器*是同義字。

---

## 第1章 總覽

MQSeries Everyplace 是 MQSeries 的商業特質傳訊產品系列的成員之一。它的設計既爲了滿足感應器、電話、PDA（個人數位助理）和膝上型電腦等小型裝置的需求，也爲了要符合行動式附屬設備的要求，以及使用脆弱的通信網路而產生之需求。它提供標準的 MQSeries 單次確定遞送，同時也能夠與其他系列成員交換訊息。由於許多 MQSeries Everyplace 應用程式在執行時，都沒有受到 Internet 防火牆的保護，因此，它也提供了非常精密的安全功能。

小型裝置需要傳訊子系統節約使用系統資源，因此，MQSeries Everyplace 也在系統軌跡和通信協定有效性上進行最佳化。它不提供 MQSeries 系列的其他傳訊成員的相同功能，但它提供無間隙的交互作業功能。在不可信賴的網路上傳訊之機動性、漫遊、本端與遠端訊息存取、安全及支援等方面，MQSeries Everyplace 都提供非常豐富的功能。

MQSeries Everyplace 是 IBM® 通用運算系列的成員之一，因此，它能與其他 IBM 通用和無線產品妥善地整合。

---

### 第 1.0 版注意事項

- MQSeries Everyplace 第 1.0 版是一個工具箱，使用者可以利用它來撰寫 MQSeries Everyplace 應用程式，以及建立執行這些應用程式的環境。
- 在這個版次中，應用程式和解決方案提供者必須負責將 MQSeries Everyplace 部署到通用裝置上。



## 第2章 必備條件

表1.顯示可用來執行 MQSeries Everyplace 第 1 版的軟體環境<sup>1</sup>。

表 1. 第 1 版支援的軟體環境

	作業系統
裝置	EPOC
	Palm OS
	Windows CE
	Windows <sup>®</sup> 95 Windows 98 Windows NT v4 Windows 2000
閘道	Windows NT v4 Windows 2000
<b>註:</b> <ol style="list-style-type: none"><li>第 1.0 版採用 Java 語言，可跨越所有支援 Java 的平台。</li><li>只能同步存取遠端佇列的限制功能從屬站，僅專用於 Palm OS 作為 C codebase 來使用。</li><li>建議採用平台所能使用的 Java 1.1 最新層次。請參閱 MQSeries Everyplace 網站 (<a href="http://www.ibm.com/software/mqseries/everyplace">www.ibm.com/software/mqseries/everyplace</a>)，以取得測試過的 Java 層次之詳細資料。</li></ol>	

1. MQSeries Everyplace 裝置碼可以在執行 Java<sup>®</sup> 的任何裝置上執行，但它只搭配表1.中所列出的作業系統進行過測試。





## 第3章 MQSeries 系列

MQSeries 系列包含許多產品，提供了某個範圍的各種功能，如圖1.所示。

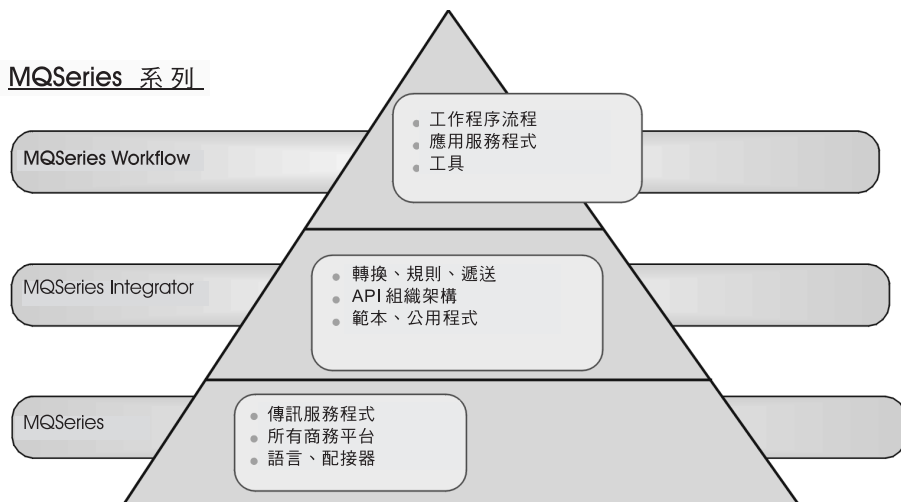


圖 1. MQSeries 系列

- **MQSeries Workflow** 利用自動化商業程序（涉及人和應用程式）來簡化整個企業的整合。
- **MQSeries Integrator** 是功能強大的訊息代理軟體，可提供及時的智慧化規則型訊息遞送，以及內容的轉換及其格式的建立。
- **MQSeries 傳訊**提供從桌面至大型電腦之間，透過商業品質傳訊來進行的「任何點至任何點」的連線，支援的平台超過 35 種。

MQSeries Workflow 和 MQSeries Integrator 產品都利用 MQSeries 傳訊層所提供的連線。

MQSeries (MQS) 和 MQSeries Everyplace 產品都提供有 MQSeries 系列傳訊；它們的設計是要支援一或多個硬體伺服器平台和（或）相關的作業系統。在平台功能具有高度多樣性的情況下，這些個別產品也組織成產品群組，來反映一般的功能和設計。產品群組有四個：

- **分散式傳訊**：MQSeries for Windows NT、AIX<sup>®</sup>、AS/400<sup>®</sup>、HP-UX、Sun Solaris，以及其他平台

- **主電腦傳訊**：MQSeries for OS/390®
- **工作站傳訊**：MQSeries for Windows 3.1、95、98
- **通用傳訊**：MQSeries Everyplace

傳訊本身的基礎在於佇列管理程式，而與特定產品或產品群組無關。佇列管理程式負責管理各個可儲存訊息的佇列。應用程式則會與本端佇列管理程式通信，並會取得佇列中的訊息，或將訊息放置在佇列中。如果訊息要放置到遠端佇列管理程式所擁有的遠端佇列，這時會透過通道，將訊息傳輸到遠端佇列管理程式。在這個方式下，可以經由一或多個中間佇列管理程式，將訊息傳送到目的地。傳訊的本質在於將傳送的應用程式和接收的應用程式分開，並在必要時，中途將訊息放入佇列中。所有 MQSeries 傳訊產品都會處理佇列、訊息和通道等相同的佇列管理程式基本元素，不過，它們的細節會有許多差異。

MQSeries 主電腦和分散式傳訊產品用來支援許多不同的網路配置，它們全都含有從屬站和伺服器。<sup>2</sup> 圖2.是其中的部份範例。

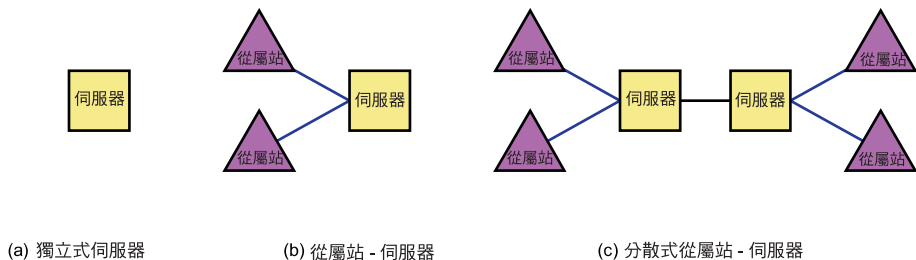


圖 2. 簡單的主電腦和分散式配置

最簡單的情況是配置執行一個佇列管理程式的獨立伺服器。在這部伺服器中，有一或多個應用程式在執行作業，且透過佇列來交換訊息。另一種配置是主從架構。這種配置的佇列管理程式只存在伺服器中，但每個從屬站都可以利用從屬站通道來存取它。從屬站通道是一種雙向通信鏈結，遵循唯一的 MQSeries 通信協定來進行類似遠端程序呼叫 (RPC) 的實作。應用程式可以在從屬站上執行，並存取伺服器佇列。主從架構的其中一項好處是相依於伺服器佇列管理程式，採用小型從屬站傳訊基礎架構。缺點是從屬站及其相關伺服器會進行同步作業，因此，從屬站通道必須固定在可用狀態中。

分散式主從架構配置涉及多部伺服器，情況更複雜。在這些配置中伺服器會利用訊息通道來交換訊息。訊息通道是單向的，採用專為了進行安全的非同步訊息資

2. 請注意，在 MQSeries 中，這些詞彙有非常明確的意義，不一定符合它們的一般用法。在本文件中，您一定要用它們在 MQSeries 的語意來使用它們。

料交換而設計的通信協定。雖然在無法通信時，伺服器之間無法流通訊息，但從屬站不需要利用這些訊息通道來繼續作業。

MQSeries 工作站傳訊產品提供有這些配置選項的子集。它們不支援伺服器，而支援有一個佇列管理程式的工作站，但不支援連接從屬站。不過，工作站可以跨越 MQSeries 訊息通道來連接到其他工作站，也可以連接到伺服器。因此，工作站通常被視為小型伺服器，當需要非同步功能時，用來取代從屬站。

圖3.顯示兩個典型的工作站配置。在 (b) 中，工作站應用程式的執行，與伺服器和從屬站是獨立而無關的：

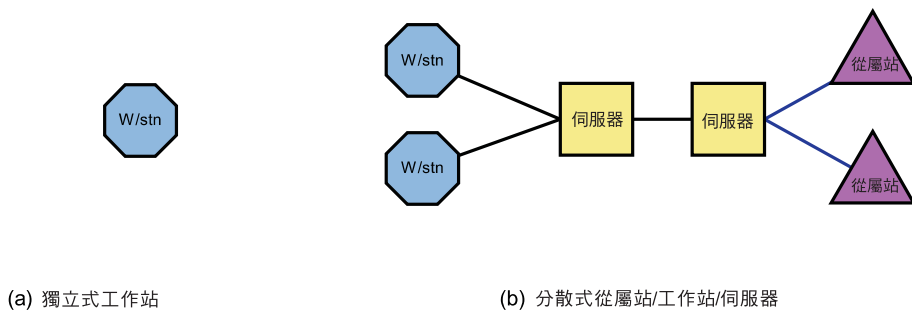


圖 3. 典型工作站配置

通用傳訊產品 MQSeries Everyplace 透過所提供的裝置和閘道來支援各種配置。

MQSeries Everyplace 裝置是一部執行 MQSeries Everyplace 程式碼，但不含通道管理程式的電腦。這表示裝置每次只能與一部其他裝置或閘道通信。MQSeries Everyplace 裝置範圍非常寬，它可以是非常小的裝置（如油料管線的感應器），也可以是比較大的裝置（如電話、個人資料助理 (PDA) 或膝上型電腦），最大的即是桌面機器和工作站。這些裝置電腦通常稱為通用裝置，雖然，這個詞彙隱含大小和功能上的限制，但本產品沒有這個限制。

閘道是一部執行 MQSeries Everyplace 程式碼，且配置了 MQSeries Everyplace 通道管理程式 (MQSeries Everyplace 通道接聽器) 的電腦。這提供了裝置程式碼的所有功能，外加同時與多部裝置閘道通信的能力。閘道也提供在 MQSeries Everyplace 網路和 MQSeries 網路之間交換訊息的機制。

大體上，我們可以說，裝置結合了從屬站和伺服器的許多屬性。它們可以配置完整的佇列功能來取得非同步作業的能力。它們也可以存取遠端保留的佇列，這種特性有些類似於從屬站存取伺服器佇列的特性。裝置和伺服器不同，不能連接從屬站。裝置可以利用同層級對同層級的傳訊功能來進行彼此間的直接通信。裝置也可以利用通道來通信，但它們是 MQSeries Everyplace 的專用通道，稱為動態通

道，以區別於 MQSeries 從屬站通道和 MQSeries 訊息通道。動態通道是雙向的，能夠支援 MQSeries Everyplace 所提供的完整功能，其中包括同步和非同步的傳訊。

閘道當然支援 MQSeries Everyplace 動態通道，以便和裝置通信。它們也可以選用地支援 MQSeries 從屬站通道，以便和伺服器通信。如同伺服器一般，閘道也有佇列管理程式，因此，能夠支援本端傳訊應用程式。

圖4.是一些典型的通用配置。

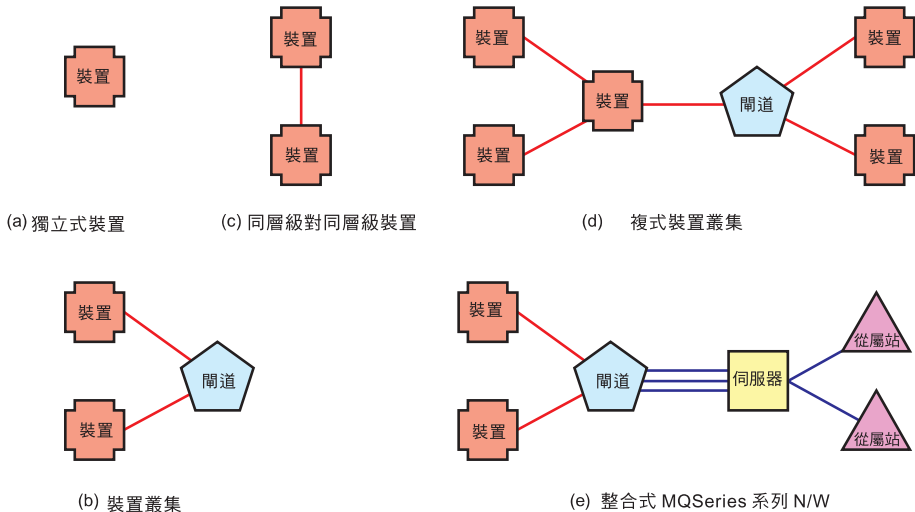


圖 4. 典型裝置配置

圖4 (b) 利用閘道，將各裝置連接一起。閘道的一項特性是它們能處理多個同時送入的連線要求，不像裝置每次只能處理一個這種要求。閘道和裝置都可以同時處理多個送出的要求。如果配置 (b) 將閘道改成裝置，這時雖然中間裝置可能已同時連接兩個終端裝置，但兩個終端裝置也必須輪流連接這個中間裝置。(d) 同時使用一個裝置和一個閘道來鏈結裝置。(e) 使用閘道將裝置網路鏈結到 MQSeries 伺服器，在這種配置中，訊息可以在所有構成要素之間流動，其中包括裝置、閘道、伺服器、工作站和從屬站。第9頁的表2.顯示這些元件最重要的特性。

表 2. MQSeries Everyplace 和 MQSeries 元素

元件	特性	提供者
裝置 (MQSeries Everyplace)	<p>提供應用程式透過動態通道來進行確定傳訊的功能</p> <ul style="list-style-type: none"> <li>容許同步進行的本端和遠端佇列存取</li> <li>容許向遠端佇列進行非同步遞送</li> <li>限制每次只能處理一個送入的要求</li> </ul>	通用
從屬站 (MQS)	<p>提供本端應用程式的確定傳訊功能</p> <ul style="list-style-type: none"> <li>需要通往伺服器的同步從屬站通道連線</li> <li>只容許同步存取相連伺服器中的佇列</li> <li>容許透過相連的伺服器，向遠端佇列進行非同步遞送</li> </ul>	分散式主電腦
閘道 (MQSeries Everyplace)	<p>提供應用程式透過動態通道來進行確定傳訊的功能</p> <ul style="list-style-type: none"> <li>容許同步進行的本端和遠端佇列存取</li> <li>容許向遠端佇列進行非同步遞送</li> <li>可同時處理多重送入的要求</li> </ul> <p>支援透過從屬站通道來連接多個 MQSeries 伺服器</p>	通用
伺服器 (MQS)	<p>提供應用程式透過訊息通道來進行確定傳訊的功能</p> <ul style="list-style-type: none"> <li>容許同步進行本端佇列存取</li> <li>容許向遠端佇列進行非同步遞送</li> </ul> <p>支援透過從屬站通道來連接多個 MQSeries 從屬站</p>	分散式主電腦
工作站 (MQS)	<p>提供應用程式透過訊息通道來進行確定傳訊的功能</p> <ul style="list-style-type: none"> <li>容許同步進行本端佇列存取</li> <li>容許向遠端佇列進行非同步遞送</li> </ul>	工作站



---

## 第4章 基本需求

本章說明 MQSeries Everyplace 之設計和實作的基本需求。

---

### 功能

MQSeries Everyplace 利用下列方式來延伸 MQSeries 系列的傳訊範圍：

- 支援低階裝置，如 PDA、電話和感應器等，使它們能夠參與 MQSeries 傳訊網路。它也支援中間裝置，如膝上型電腦、工作站和特定的分散式平台。MQSeries Everyplace 提供相同服務品質、訊息的單次確定遞送，且容許與其他系列成員交換訊息。
- 提供範圍更廣的安全特性來保護儲存體和傳輸中的訊息、佇列和相關資料。
- 在網路不穩定或頻寬非常有限的不良通信環境中，使作業更有效。它擁有一種很有效率的有線通信協定，以及通信鏈結失敗時的自動復原功能。
- 支援行動式使用者，讓網路連線功能的指向能隨著裝置的漫遊而變更。它也容許在電池資源和網路失敗或受到限制的狀況下控制行爲。
- 透過適當配置的防火牆來作業
- 將使用者管理工作縮到最小，因而大體上能隱藏裝置中的 MQSeries Everyplace。這使得 MQSeries Everyplace 非常適合作爲公用程式型應用程式的建置基礎。
- 非常容易自訂和延伸，利用應用程式提供的規則和其他修改行爲的類別，或透過建立基本物件類別之子類別，來完成各種目的，比方說，代表不同的訊息類型。

---

### 應用程式

由於有各型各樣的選項，因此，無法嚴格列出可能的 MQSeries Everyplace 應用程式清單，但我們可以預期有相當多是爲特定使用者群組而開發的自訂應用程式。下面列出一些思考過的範例：

- **消費者應用程式：** 使用 PDA 在家裡進行超市商品的選購，收集旅行者利用行動電話在航班、金融交易的偏好
- **控制應用程式：** 收集和整合透過衛星傳輸的油管感應器資料，安全地遠端操作設備（如活門）以保證操作員的有效
- **機動勞工：** 拜訪專家（保險公司）、快速開立郵包遞送公司的客戶收件證明、速食店服務生與廚房交換資訊、高爾夫球比賽計分、警察的機動安全系統傳訊系統、經常失去聯繫的基層勞工工作資訊，當地計量讀取。

- **個人生產力**：郵件/日曆複製、資料庫複製、膝上型電腦小型化。

---

## 客戶需求

影響 MQSeries Everyplace 設計的需求如下：

- **管理**：最少的安裝和維護工作；支援本端和遠端管理；延伸和自訂管理功能以符合特定應用程式需求；強調自動發現和復原；提供可選擇使用的獨立管理元素。
- **通信**：非常有效的有線通信協定；標題最少；訊息中沒有強制的欄位（唯一識別碼除外）；變更資料編碼的能力；壓縮、加密和鑑別支援；壓縮和安全特性的端點對端點協定；輕易通過防火牆的能力；可外掛的通信配接器。
- **相容性**：MQSeries 服務品質和無縫式的訊息交換；不透過應用程式變更來與現存 MQSeries 系統通信的能力；彈性控制 MQSeries 和 MQSeries Everyplace 之間的訊息交換。
- **軌跡**：最基本配置的裝置系統，大體上在 100KB 以下。
- **功能**：同步和非同步的傳訊功能，可存取本端或遠端佇列所保存的訊息；使用訊息內的任何欄位來進行選擇性擷取的能力；選擇性控制佇列的支持媒體。
- **規則支援**：透過規則來控制行為的許多方面，比方說，何時傳送訊息、通信鏈結的重試頻率、如何處理太大的訊息，或目標佇列已滿時要如何行動。
- **安全**：安全、鑑別和不拒絕的完整支援；訊息層次和佇列層次的安全；保護傳訊系統免受安全攻擊；利用業界標準演算法的可外掛安全；與作業系統使用者憑證整合的能力；符合國家安全需求，使安全支援能在訊息跨越國界時能隨著改變的能力。



---

## 第5章 產品概念

---

### 簡介

MQSeries Everyplace 程式設計模型的基本元素有訊息、佇列及佇列管理程式。MQSeries Everyplace 訊息是含有應用程式定義內容的物件。儲存時，它們會保留在佇列中，且您可以透過 MQSeries Everyplace 網路來移動它們。您可以指定一組目標佇列管理程式和佇列名稱，將訊息傳送到目標佇列中。應用程式會利用放置作業，將訊息放在佇列中，且通常會利用取得作業來擷取它們。佇列可以在本端或遠端，由佇列管理程式來管理它們。裝置和閘道都會將配置資料儲存在暫存器中。

裝置中的應用程式可以利用裝置所能取得的任何或全部 API 或功能，而不必受限於 MQSeries Everyplace 程式設計介面。MQSeries Everyplace 裝置可以透過動態通道來連接到其他 MQSeries Everyplace 裝置和（或）MQSeries Everyplace 閘道。

閘道中的應用程式也能利用閘道所能取得的任何或全部 API 或功能，而不只是 MQSeries Everyplace 程式設計介面。閘道可以透過動態通道來連接到其他閘道和（或）MQSeries Everyplace 裝置。閘道也可以透過 MQSeries 從屬站通道來連接到一或多個 MQSeries 伺服器（但不能連接到其他 MQSeries Everyplace 閘道）。MQSeries Everyplace 和 MQSeries 可共存於單一機器中，不過，MQSeries 的存在完全是選用性的。

除了下面這兩項之外，全功能的裝置和閘道功能相同：

- 閘道可以處理多個同時入埠的要求（來自其他裝置和/或閘道）
- 閘道可以同時作為多重 MQSeries 伺服器的介面

動態通道支援下列網路連線：

- 撥號連線
- 永久連線，比方說，傳統的 LAN、專線、紅外線或無線的 LAN。

通信協定由一組配接器來實作，每個支援的通信協定都有一個配接器。因此，必要時可以加入新的通信協定，特定環境內的記憶體軌跡也可以調整配合特定的配置。

佇列會透過另一組配接器來個別映射至儲存媒體。因此，佇列可以視所選配接器而定，儲存在檔案系統或記憶體中。

依 MQSeries Everyplace 程式設計介面的設計，撰寫的應用程式可以不依賴佇列的位置。因此，設計要存取本端佇列的程式應該不需要改變，便能從遠端佇列管理程式來執行作業，（能夠滿足任何強制的安全考量，不支援遠端佇列的特定 MQSeries Everyplace 作業）。這個獨立性包括任何管理功能的使用。

---

## 訊息物件

MQSeries Everyplace 訊息物件和 MQSeries 支援的訊息有基本差異。在 MQSeries 中，訊息是位元組陣列，可分成訊息標頭和訊息主體。MQSeries 能夠瞭解訊息標題，訊息標題中有非常重要的訊息，比方說，對佇列之回答的身份識別、對佇列管理程式之回答、訊息 ID，及相互關係 ID；但它不能瞭解訊息主體。

相對而言，MQSeries Everyplace 中的訊息是繼承自 MQSeries Everyplace 物件（稱為欄位物件）的訊息物件。訊息是真的物件，沒有標題或訊息主體的概念。在瞭解了基本欄位物件之後，訊息物件的真實性質會變得更清晰。這些欄位物件在 MQSeries Everyplace 中使用得非常廣泛，它們是欄位的彙集，欄位中含有名稱、資料類型和資料本身。欄位名稱是不限長度的 ASCII 字串（列出若干保留字元）。

欄位類型可以是：

- **ASCII** 字串或 ASCII 字串的動態陣列
- **Boolean** 值
- **位元組**，位元組值的固定陣列或動態陣列
- **倍精準浮點**，倍精準浮點值的固定陣列或動態陣列
- **欄位物件或欄位物件的動態陣列**（因此，支援欄位物件的巢狀結構）
- **浮點值**，浮點值的固定陣列或動態陣列
- **整數**（4 位元組），整數的固定陣列或動態陣列
- **長整數**（8 位元組），長整數的固定陣列或動態陣列
- **短整數**（2 位元組），短整數的固定陣列或動態陣列
- **UNICODE** 字串或 UNICODE 字串的動態陣列

欄位物件有類型，這個類型對應於程式設計物件類別名稱。應用程式會利用這個物件類別的後代作為訊息物件，必要時，MQSeries Everyplace 會利用它們的類型來建立正確物件類別的案例，比方說，在訊息物件從某通道中流過之後。

欄位物件可提供若干方法，比方說，欄位可以列舉，欄位的存在可以驗證。同樣地，欄位物件也可以進行相等性的比較。它們有能力與位元組陣列之間，來回進行欄位項目的傾出和還原，比方說，它們可以提供透過鏈結來進行的物件傳輸資料，以及在傳輸之後還原物件。傾出和還原方法是可置換的，因此，欄位物件可以採用其他方式來序列化它們本身，比方說，在傳輸之時查詢資料庫來找到欄位

內容。表3.列出欄位物件的內容和（或）它們的構成欄位。

表 3. 欄位物件及其構成欄位內容

內容	存在	
	欄位物件	欄位
相關屬性物件	選用	
構成欄位	是	
隱藏		是
名稱		是
類型	是	是
值		是

欄位的隱藏內容使得欄位能夠在比較作業中，得以被忽略。

屬性物件含有執行鑑別、加密和壓縮的機制，可建立和欄位物件的關聯性。

- **鑑別**：控制存取
- **壓縮**：縮減儲存體需求（用於傳輸和（或）儲存）
- **加密**：在傾出物件時保護內容（容許還原）

屬性物件是 MQSeries Everyplace 安全模型的基礎，除了其他用法之外，它容許在備份儲存體時，選擇性地存取內容和保護資料。表4.列出屬性物件的內容。當有「規則」值時，它會控制容許執行哪些作業。

表 4. 屬性物件內容

內容	存在
加密程式	選用（在某些情況下可能會需要）
鑑別程式	選用
壓縮程式	選用
規則	選用
類型	選用

訊息物件從欄位物件衍生而來，其中包括 MQSeries Everyplace 所產生的 UID（唯一識別碼）。這個 UID 用來唯一識別訊息物件，是根據下面兩個項目而建構成：

- 產生的佇列管理程式之**名稱**（佇列管理程式在收到物件時新增這個項目）。這個名稱必須是在廣域環境內的唯一名稱。
- 建立訊息物件的**時間**（在建立時新增）

除了從欄位物件繼承來的內容之外，訊息物件還有表5.中所列出的基本內容。

表 5. 訊息物件內容

內容	說明
Msg_OriginQMgr	這是傳送訊息之佇列管理程式的名稱
Msg_Time	這是應用程式建立訊息物件的時間

這兩個內容構成了訊息物件的唯一識別碼 (UID)。

在以另一個 MQSeries Everyplace 佇列管理程式為目標的訊息中，不需要任何其他資訊，雖然通常會包括攜帶訊息資訊內容的其他欄位。訊息通常是基本訊息物件類別的後代，因此，會有符合其目的之附加欄位。在這些附加欄位中，會有一個訊息應用程式所共用的號碼，比方說，對於佇列管理程式的回答。因此，MQSeries Everyplace 提供了某些支援它們的方式。

表6.列出支援的欄位。

表 6. 提供的訊息物件欄位

欄位名稱	用法
Msg_CorrelID	通常利用位元組字串來建立回答與原始訊息的關聯
Msg_ExpireTime	刪除訊息的時間（即使沒有遞送也一樣）
Msg_LockID	解除鎖定訊息所需要的索引鍵
Msg_MsgID	應用程式用來建立與原始訊息的關聯
Msg_Priority	訊息的優先順序
Msg_ReplyToQ	佇列回答應設定其位址的佇列名稱
Msg_ReplyToQMgr	佇列回答應設定其位址的佇列管理程式名稱
Msg_Resend	指出訊息是上一個訊息的重新遞送
Msg_Style	區分指令和要求/回答...等

不論在任何情況下，都會有一個定義好的常數可利用單位元組來攜帶欄位名稱。有些欄位會提供更多內容 - 比方說：優先順序（如果有的話）會影響訊息的傳輸次序；相互關係 ID 會觸發這些欄位值的佇索引作業，以加快擷取速度；到期時間會觸發訊息的到期，等等。

訊息物件中會定義許多方法，比方說，取出訊息 UID、產生的佇列管理程式名稱及物件建立時間的能力。其他非常有用的方法繼承自欄位物件類別，比方說，取得和放置欄位值的各種方法。傾出方法尤其有用，它用來將物件資料傾出到位元組字串中。當訊息儲存到持續性儲存體儲存體時，以及要透過動態通道來傳輸訊息時，MQSeries Everyplace 都會呼叫這個方法。在這個方式下，訊息物件本身要負責判斷資料值的外部表示法，這一點可以利用多種方式來做到。比方說，物件可

以只是傾出構成欄位的值，但也可以選擇查詢資料庫。在根據傾出格式來重建物件時，互補的還原方法可提供類似的控制能力。請注意，如果訊息物件有附加的屬性物件，則傾出時會呼叫屬性的資料編碼程式、加密程式和壓縮程式；同樣地，還原時也會呼叫解碼程式、解密程式和解壓縮程式。

當 MQSeries Everyplace 傳送訊息物件時，爲了要縮減在線路上的軌跡，它不會傳送相關聯的類別檔。因此，在要建立訊息物件案例的每個佇列管理程式中，必須能夠取得適當的訊息類別。

預設的訊息物件傾出方法已經過最佳化，它可以將產生的位元組字串縮到最小，以完成有效的訊息儲存和傳輸工作。

## 傾出資料格式

預設的傾出資料格式會依照下列方式來進行欄位編碼：

```
{Length Identifier Fence {Data}} {Length Identifier Fence {Data}} { ... }
```

其中：

- *Data*：這是資料值。整數會移除前導 0 和 F 來進行壓縮。Boolean 沒有相關的資料類型
- *Fence*：這是特殊的位元組，用來定出識別碼和選用資料項目的界限。這個位元組也指出資料項目類型
- *Identifier*：這會利用可變長度的 ASCII 位元組字串來保留欄位名稱，並終止於結束位元組
- *Length*：這指出資料欄位的長度。這會使用 1-4 之間的可變位元組數目。第一個位元組保留前兩個位元來指出長度欄位的長度。支援的長度在 0 - 1,073,741,823 之間

這會造成高度壓縮的資料串流。更進一步的節約，可以透過壓縮資料來達成。含有前一位元組串流的 XOR 壓縮可能被預期能產生好的結果，不過，由於這些欄位具有變動的性質，且欄位次序可以改變，因此，簡單的 XOR 不一定能產生所需要的效果。MQSeries Everyplace 併入了一種智慧型 XOR，能夠逐欄位作業，比較可能改進壓縮。

---

## 佇列

佇列通常在應用程式移除訊息物件時，用來保留訊息物件。和訊息一樣，佇列也衍生自欄位物件。應用程式不能直接存取佇列物件<sup>3</sup>；而佇列管理程式扮演了應用程式和佇列的中介角色。佇列由名稱來識別，名稱可以是無限長<sup>4</sup>的 ASCII 字串，但在特定佇列管理程式中，它必須是唯一的。

MQSeries Everyplace 支援許多不同的佇列類型：

### 本端佇列

應用程式利用本端佇列，以一種安全的方法來儲存訊息。它們會有一個透過配接器類別來存取的訊息儲存庫，通常是採用磁碟配接器類別。不過，MQSeries Everyplace 提供記憶體配接器類別，將訊息儲存庫保留在記憶體內，以便於進行快速存取（代價是系統毀損時，會失去其中的訊息）。由於可以建立適當的配接器，在佇列一個接一個的基礎上，訊息可以儲存在任何位置上；比方說，在關聯式資料庫中，可以將訊息儲存在可寫入的 CD 等。本端佇列可以連線或離線使用，也就是連接或不連接網路。存取和安全的擁有者是佇列，可由遠端佇列管理程式來授與使用權，當連接到網路時，可讓他者與佇列相互傳送和接收訊息。本端佇列存取固定採同步的方式。

### 遠端佇列

遠端佇列不在本端環境內；不過，本端環境內會有一個定義指出擁有的佇列管理程式和真實的佇列。遠端佇列可經由同步或非同步的方式來存取。如果本端環境內保留了遠端佇列的定義，這時存取模式會以這個定義為基礎。如果沒有的話，則會有發現佇列的動作，這時會去發現特性，且會強迫採取同步的存取模式。

如果要存取*同步*佇列，佇列必須已連接到網路，且網路有路徑通往擁有它的佇列管理程式。如果沒有建立起網路，則放置、取得和瀏覽等作業（請參閱第24頁的表10）都會造成異常狀況。擁有佇列的佇列管理程式控制了存取佇列所需要的許可權和安全需求。應用程式必須在傳送或接收訊息時，負責處理任何錯誤或重試作業；這時候，MQSeries Everyplace 不再負責單次確定遞送。

*非同步*佇列是可在其中放入訊息但不加以擷取的佇列。如果建立了網路連線，則訊息會遞送到擁有的佇列管理程式和佇列。不過，如果網路沒有連

---

3. 在佇列規則內可以直接存取。

4. 爲了能夠交互作業，建議您遵守 MQSeries 的命名限制，其中包括名稱長度不超出 48 個字元。您使用的檔案系統也可能會限制這個長度。

線，這時訊息會儲存在本端，直到網路連線之後，再傳輸訊息。這使得應用程式能在裝置離線時在佇列上操作；不過，它要求這些類型的佇列有訊息儲存庫，以便暫時儲存訊息。

### 儲存和轉遞佇列

這類型的佇列會將訊息儲存起來，直到訊息能轉遞到下個佇列管理程式為止（不一定是擁有的佇列管理程式）。這類型的佇列通常（不必然）是在閘道中定義的，當它連接到網路時，裝置必須收集它的訊息。儲存和轉遞佇列可以保留許多目標佇列管理程式的訊息，另外，也可能每個目標佇列管理程式各有一個儲存和轉遞佇列。當傳送端想要傳送訊息給可能已切斷的接受者時，傳送者仍會將訊息的位址設於接收者的佇列管理程式/佇列；中間伺服器會偵測出接收者沒有連線，並將訊息儲存在它的本端訊息儲存庫中。不論是將訊息傳送到連線或切斷連線的目標佇列，傳送的應用程式都不需要進行任何變更。

### 起始伺服器佇列

這類型的佇列通常位於裝置中（假設有時會連線），且會指向所謂起始伺服器的佇列管理程式其中的儲存和轉遞佇列。每當有裝置連接到網路時，起始伺服器佇列都會從起始伺服器中拉出訊息。當佇列從伺服器中拉出訊息時，它會利用 `putMessage` 和 `confirmputMessage` 方法呼叫（請參閱第23頁的『佇列管理程式作業』），將訊息交給本端佇列管理程式。之後，佇列管理程式要負責將訊息放到正確的本端佇列中。藉由網路串流從伺服器中取得訊息的拉出方法，比由伺服器來推送訊息更有效；因為起始伺服器佇列會將第一個訊息的確認視為對於第二個訊息（如果有的話）的要求，而伺服器推送的方法則需要透過要求/回應來送出訊息，之後，確認串流還需要第二個要求/回應。起始伺服器佇列通常會設定輪詢間隔，因此，當連著網路時，它會檢查伺服器中有沒有擱置中的訊息。輪詢間隔是一個管理配置選項。

### 管理佇列

這類型的佇列會接收 MQSeries Everyplace 管理訊息。選用的管理訊息回答佇列也可用來接收 MQSeries Everyplace 系統送出之管理訊息的回答。管理佇列不瞭解如何執行管理，但它們會處理封裝了管理詳細資料的訊息。

### MQSeries 橋接佇列

這是一種特殊形式的遠端佇列，它的定義在閘道及某 MQSeries 佇列管理程式上的目標佇列中。這種形式的佇列提供了在 MQSeries Everyplace 和 MQSeries 環境之間的路徑。任何需要重新建立資料或訊息格式時，都會用到轉換程式。MQSeries Everyplace 提供了一種非常基本的轉換程式；它預期程式設計師會自訂這個轉換程式來配合他們自己的需求。

MQSeries Everyplace 會安全地將資料儲存在佇列中，以確保訊息能夠實際寫入媒體，而不只是放到作業系統的緩衝區內。不過，MQSeries Everyplace 不會獨立記載訊息和佇列的變更。如果需要進行媒體失敗的復原作業，這時您必須部署硬體解決方案，比方說，使用 RAID 磁碟系統。另外，佇列也必須映射至可復原的儲存體，如特定的資料庫子系統。

MQSeries Everyplace 不要求佇列管理程式定義佇列。不過，必要時會提供四個系統佇列：

- **AdminQ**：接收管理訊息必須使用
- **AdminReplyQ**：在接收對於管理訊息的回答時，可以選擇性地使用
- **DeadLetterQ**：用來儲存其他方式所無法遞送的訊息
- **SYSTEM.DEFAULT.LOCAL.QUEUE**：與 MQSeries 伺服器中之必要系統佇列分享共用名稱的佇列

表7.顯示佇列內容，不過，請注意，並非所有顯示的內容都適用於所有佇列類型：

表 7. 佇列內容

內容	說明
Admin_Class	佇列類別
Admin_Name	ASCII 佇列名稱
Queue_Active	指出佇列在作用中
Queue_AttRule	控制安全作業的規則類別
Queue_Authenticator	鑑別程式類別
Queue_BridgeName	擁有 MQ 橋接名稱
Queue_ClientConnection	從屬站連線名稱
Queue_CreationDate	佇列的建立日期
Queue_Compressor	壓縮程式類別
Queue_Cryptor	加密程式類別
Queue_CurrentSize	佇列中的訊息數
Queue_Description	Unicode 說明
Queue_Expiry	訊息到期時間
Queue_FileDesc	儲存佇列的位置
Queue_MaxMsgSize	佇列所容許的最大訊息長度
Queue_MaxQSize	所容許的最大訊息數
Queue_Mode	同步或非同步
Queue_MQMgr	MQSeries 佇列管理程式 Proxy



表 7. 佇列內容 (繼續)

內容	說明
Queue_Priority	訊息要用的優先順序（除非被訊息值所置換）
Queue_QAliasNameList	佇列的替代名稱
Queue_QMgrName	擁有實際佇列的佇列管理程式
Queue_RemoteQName	遠端 MQSeries 欄位名稱
Queue_Rule	佇列作業的規則類別
Queue_TargetRegistry	目標登錄類型
Queue_Transporter	傳輸程式類別
Queue_Transformer	轉換程式類別

您也可以利用管理功能來建立和刪除佇列，以及查詢或修改它們的內容。

佇列不只能作為訊息的儲存庫。佇列的子類別還可以用在程序控制應用程式的實務中，比方說，佇列物件可以直接控制活門。類型正確的訊息可以開啓活門、改變流量...等。應用程式並不會從佇列中拉出訊息，再執行動作；佇列物件會自行控制這個動作。再比方說，其他佇列可以更新試算表，或進行本文與說話之間的轉換。這個技術的好處是佇列的安全層面仍存在且有效，另外，確定傳訊也是一項好處。因此，MQSeries Everyplace 仍會確定訊息的單次遞送，相關聯的鑑別程式和加密程式會保證只有獲授權的訊息傳送者可傳送這些訊息，且在傳輸過程中內容非常安全。沒有任何應用程式能夠或需要存取佇列。

## 佇列管理程式

MQSeries Everyplace 佇列管理程式可讓應用程式存取訊息和佇列，以及控制任何通道。在 MQSeries Everyplace 第 1.0 版中，不論任何時候，單一 Java 虛擬機器都只能有一個作用中的佇列管理程式。如果機器中有多個 JVM，這時可以有和 JVM 相同數目的佇列管理程式。佇列管理程式是依名稱來識別的，這個名稱必須具有廣域環境內的唯一性<sup>5</sup> 且必須是一個 ASCII 字串，長度沒有限制。<sup>6</sup> 佇列管理程式可以配置成含或不含本端佇列作業。所有佇列管理程式都支援同步的傳訊作業，含本端佇列作業的佇列管理程式也支援進行非同步的訊息遞送。

非同步和同步訊息遞送有非常不同的特性和結果：

- 
- MQSeries Everyplace 或 MQSeries 並不強制執行這項限制，但重複的佇列管理程式名稱可能會使訊息送到錯誤的佇列管理程式。
  - 爲了能夠交互作業，建議您遵守 MQSeries 佇列管理程式的命名規則，其中包括名稱長度不超出 48 個字元。您使用的檔案系統也可能會限制這個長度。

## 非同步訊息遞送<sup>7</sup>

應用程式利用非同步訊息遞送來傳遞訊息給 MQSeries Everyplace，以便遞送到遠端佇列。放置作業完成之後，會立即回覆應用程式。MQSeries Everyplace 會暫時將訊息保留在本端環境內，直到可以遞送它為止。遞送可以分階段進行，由 MQSeries Everyplace 來負責遞送。這個作業模式提供單次確定遞送。請參閱第31頁的『非同步訊息遞送』，找到進一步的討論。

### 同步訊息遞送：

同步傳訊可用來設定下列項目的位址：

- 透過 MQSeries Everyplace 網路來遞送的 MQSeries Everyplace 佇列管理程式中之目標佇列
- 直接連接到 MQSeries Everyplace 閘道的 MQSeries 佇列管理程式中之目標佇列
- 間接連接的 MQSeries 伺服器中之目標佇列

在同步傳訊之下，應用程式會將訊息放到 MQSeries Everyplace 中來進行遞送。MQSeries Everyplace 採用同步方式來聯絡目標遠端佇列及放置訊息。遞送之後，MQSeries Everyplace 會傳回應用程式。

與遠端佇列管理程式聯絡涉及透過中間裝置和（或）閘道來進行的 MQSeries Everyplace 遞送。請參閱第32頁的『同步訊息遞送』，找到進一步的討論。

因此，非同步訊息遞送表示本端應用程式會將訊息交給 MQSeries Everyplace，而從這個本端佇列管理程式開始的後續遞送，便由 MQSeries Everyplace 來負責。這表示不需要使用網路和（或）接收端應用程式。傳送端的應用程式不會知道實際遞送的時間。同步訊息遞送需要網路在執行中，但傳送端的應用程式會知道它已遞送到接收端應用程式的佇列中。不論在非同步或同步的情況中，都不需要能夠使用接收端應用程式。

本端佇列管理程式有反映本端佇列管理的內容。它也需要每個必須連接的遠端佇列管理程式的連線定義。因此，連線定義有時稱為遠端佇列管理程式定義。這些定義可能包括佇列管理程式之間直接通信所需要的所有資訊（包括網路位址），它們也可能只會指出通信是間接的，透過中間佇列管理程式來進行。在後面一種情況中，只需要有下個中繼佇列管理程式的名稱即可。

---

7. MQSeries Everyplace 無法分辨 MQSeries 所提供的持續性和非持續性模式的差異，它只支持持續性。不過，必要的話，您可以利用佇列備份儲存選項來取得效能和復原的折衷點。

表8.和表9.顯示佇列管理程式內容。

表 8. 本端佇列管理程式內容

內容	說明
QMgr_ChnlAttrRules	通道屬性規則
QMgr_ChnlTimeout	通道逾時
Admin_Class	佇列管理程式類別
QMgr_Description	Unicode 說明
Admin_Name	佇列管理程式名稱
QMgr_Rules	佇列管理程式作業的規則類別

表 9. 連線（遠端佇列管理程式）內容

內容	說明
Con_Adapter	配接器檔案描述子
Con_AdapterOptions	配接器選項（如：使用歷程）
Con_AdapterParm	配接器要用的 ASCII 資料（如 Servlet 名稱）
Con_Aliases	佇列管理程式的替代名稱
Con_Channel	這個連線應該使用的通道類型
Con_Description	Unicode 說明
Queue_QMgrName	定義的擁有者
Admin_Name	佇列管理程式名稱

連線定義支援多重配接器。

## 佇列管理程式作業

佇列管理程式支援傳訊作業，且可以選用性地管理佇列。應用程式可以透過執行的佇列管理程式作業來存取訊息。這時除非指定了過濾器，否則，會擷取佇列中第一個可用的訊息。過濾器是一個核對相等與否的欄位物件，訊息中的任何欄位都可用來進行選擇性的擷取作業。如同所有訊息傳送和擷取作業一樣，取得作業可以得到一個用來進行訊息之編碼和解碼的屬性物件。

在 MQSeries Everyplace 中，如同在 MQSeries 中，取得通常是一種破壞性的作業。如果在 MQSeries Everyplace 和應用程式之間需要進行確定傳訊，則取得之後，應該跟著確認方法序號。首先是以確認 ID（由應用程式選取它的值）來發出取得要求，這個作業會取得訊息，但會將它藏在佇列中，而不是立即刪除它。後續的確認作業會指定原始訊息 UID，指出應用程式的取得作業已順利完成，之後，會將訊息刪除。如果取得失敗，則可以復原訊息。放置作業也採用類似的方式來作業。

您可以指定 **UID** 來刪除而不擷取佇列中的訊息。

如果需要進行非破壞性的讀取動作，您可以瀏覽佇列來尋找訊息（選擇性地在過濾器的控制下）。瀏覽會擷取符合過濾器的所有訊息物件，但會將它們留在佇列中。另外，也支援鎖定瀏覽。這是一項新增功能，會鎖定佇列中相符的訊息。訊息可以個別地鎖定，也可以利用過濾器，以群組的方式來鎖定，鎖定作業會傳回鎖定 **ID**。只有在提供了鎖定 **ID** 時，才能取得或刪除鎖定的訊息。瀏覽器中有一個選項可選擇傳回所有訊息，或只傳回 **UID**。

應用程式可以在一段指定的時間內，等待訊息到達佇列。您可以選用性地使用過濾器來識別感興趣的訊息，您也可以指定**確認 ID**。另外，應用程式也可以接聽MQSeries Everyplace 訊息事件，同樣，您也可以選擇使用過濾器。當訊息到達佇列時，接聽器會獲得通知。

傳訊作業依表10 所示的方式來啓用佇列

表 10. MQSeries Everyplace 佇列中的傳訊作業

	本端佇列	遠端佇列 <sup>1</sup>	
		同步	非同步
瀏覽（±鎖定，±過濾器）	是	是	
刪除	是	是	
取得（±過濾器）	是	是	
接聽（±過濾器）	是		
放置	是	是	是
等待（±過濾器）	是	是	
<b>註：</b>			
1. 同步遠端等待作業是透過遠端佇列一起來實作的，因此，實際的等待時間是輪詢時間的倍數。			
2. <sup>1</sup> MQSeries Everyplace 第 1.0 版所提供的 MQSeries Everyplace MQSeries 橋接只支援放置 ('put') 作業。			

在起始之後，佇列管理程式可以選擇性地立即載入應用程式（類別）；同樣地，它們也可以在關機時終止應用程式。佇列管理程式會產生事件來反映狀態或錯誤；依預設，這些都會出現在事件日誌中。

## 管理

管理提供配置和管理 MQSeries Everyplace 資源（如：佇列與連線）的機能。訊息相關功能屬於應用程式的責任。管理是透過處理管理訊息之產生和接受的介面來啓用的，它的設計是要本端和遠端管理都能採用相同方式來處理。要求會傳送到目標佇列管理程式的管理佇列中，必要時，可能會收到回答。任何本端或遠端 MQSeries Everyplace 應用程式都可以直接或間接透過說明程式方法來建立和處理管理訊息。管理訊息也可以間接透過「MQSeries Everyplace 探險家」<sup>8</sup> 來產生，它是一個提供系統管理之圖形使用者介面的管理工具。

管理佇列不瞭解如何執行個別資源的管理；這項知識封裝在每個資源及其對應的管理訊息內。

## 管理訊息

管理訊息繼承基本的 MQSeries Everyplace 訊息物件。表11.列出提供給 MQSeries Everyplace 資源管理的訊息類別。這些基本管理訊息可以產生子類別來管理其他物件；比方說，不同類型的佇列可以利用 MQeQueueAdminMsg 的子類別來加以管理。MQSeries Everyplace 通往 MQSeries 的橋接會利用這個方式來使用 MQeAdminMsg 的子類別。

表 11. 管理訊息類別

管理訊息類別	使用
MQeAdminMsg	作為所有管理訊息之基礎的抽象類別
MQeQueueManagerAdminMsg	管理佇列管理程式
MQeQueueAdminMsg	管理本端佇列
MQeRemoteQueueAdminMsg	管理遠端佇列
MQeAdminQueueAdminMsg	管理「管理佇列」
MQeHomeServerQueueAdminMsg	管理起始伺服器佇列
MQeStoreAndForwardQueueAdminMsg	管理儲存和轉遞佇列
MQeConnectionAdminMsg	管理佇列管理程式之間的 od 連線
MQeClientConnectionAdminMsg	管理用來連接到 MQS 的橋接從屬站連線物件
MQeListenerAdminMsg	管理用來收集 MQS 訊息的橋接傳輸佇列接聽器物件
MQeBridgeAdminMsg	管理通往 MQS 的橋接
MQeMQBridgesAdminMsg	管理 MQ 橋接清單

8. 「MQSeries Everyplace 探險家」不包含於第 1.0 版，但您可以從全球資訊網的 MQSeries 軟體下載網站中取得，網址是：<http://www-4.ibm.com/software/ts/MQSeries/>。

表 11. 管理訊息類別 (繼續)

管理訊息類別	使用
MQeMQQMGrProxyAdminMsg	管理 MQSeries 佇列管理程式的橋接表示法
MQeMQBridgeQueueAdminMsg	管理 MQ 橋接佇列

管理訊息的結構依它的特定類別而定，也就是所管理之資源的性質，以及要執行於該資源之作業的詳細資料。不過，管理訊息通常會有如表12.所示的結構：

表 12. 管理訊息的一般結構

第 1 層次欄位	第 2 層次和以下的欄位	使用
Admin_Action		建立、刪除、查詢...等
Admin_Errors		欄位物件母項
	多重欄位	以個別錯誤為基礎的詳細資訊
Admin_MaxAttempts		應該採取的最大管理動作次數
Admin_Parameters		欄位物件母項
	資源	要管理的資源名稱
	多重欄位	訊息類別和動作的特定詳細參數資料
Admin_Reason		表示失敗原因的文字訊息
Msg_ReplyToQ		回應所應送達的佇列名稱
Msg_ReplyToQMGr		回應所應送達的佇列管理程式名稱
Admin_RC		表示輸出的數值回覆碼
Msg_Style		指令或要求/回答
Admin_TargetQMGr		擁有目標資源的佇列管理程式名稱

支援的管理訊息樣式有三種，它們是指出不需要回答之管理動作的指令（圖解）、需要回答的要求，以及回答本身。回答從原始訊息副本建構而來；因此，傳送者可以加入其他欄位供接收者使用。

除了基本的管理訊息支援之外，也為大部份一般管理作業提供了封裝回答之訊息建構和解譯的說明程式類別。這些類別可選用地提供使用者對話框，使它們在建置簡單管理工具<sup>9</sup> 上，變得非常有用。

## 選擇性管理

您可以利用管理佇列中的鑑別程式來控制對於管理的存取。如果是本端應用程式，提供的鑑別程式會將它們視為全部代表相同的本端使用者，因此，會全部接受或全部拒絕它們的管理。遠端管理應用程式的控制方式，是在傳送任何管理訊息之前，先在通道上呼叫鑑別程式。因此，可以區分出不同的遠端使用者，並個別地加以啟用或停用。在任何使用者的所有情況中，都會以整體的來啟用或停用管理。如果需要更精細層面的管理控制，比方說，要讓某些管理使用者存取某些佇列，但不能存取其他佇列，這時需要進行其他程式設計。更複雜的鑑別程式可以追蹤使用者身份所關聯的許可權，之後，便能在這些許可權的基礎上，後續處理管理訊息（請參閱安全一節）。您也可以利用佇列所關聯的規則，以類似的方式來容許或拒絕動作（請參閱第39頁的『規則』）。

## 監視和相關動作

管理所涉及的範圍通常比建立和修改物件還多，比方說，監視系統狀態及處理錯誤狀況；比方說，在佇列將滿之時通知操作員，或在訊息到達目標佇列而訊息太大時，採取適當動作。在 MQSeries Everyplace 中，這些問題都使用規則來處理；它們是一些在物件明顯變更其狀態時，或發生特定類型的錯誤狀況時，所呼叫的類別。MQSeries Everyplace 提供了一組預設的規則類別，但它們通常會被自訂類別取代（請參閱第39頁的『規則』）。

---

## 動態通道

MQSeries Everyplace 利用稱為動態通道的邏輯鏈結來進行裝置和（或）閘道佇列管理程式之間的通信。它們都支援雙向流動，佇列管理程式可以視需要來建立它們。非同步和同步的傳訊都會利用相同的通道，且所用的通信協定是 MQSeries Everyplace 的唯一通信協定。相反地，MQSeries 通常會利用從屬站通道來進行同步通訊，且會利用一對訊息通道來進行雙向的非同步傳訊。MQSeries 叢集訊息通道的某些特性類似於 MQSeries Everyplace 動態通道，但仍有些不同。

動態通道是兩個佇列管理程式之間的邏輯連線，它是為了傳送或接收資料而建立的。多個並行通道可同時存在，即使在相同的協力方之間也是如此。它們有各種

---

9. 第 1.0 版沒有併入這些類別，但您可以從全球資訊網的 MQSeries 軟體下載網站中找到它，網址是：  
<http://www-4.ibm.com/software/ts/MQSeries/>。

特性，比方說，所用的鑑別、加密、壓縮及傳輸通信協定。這些都是可外掛的特性，（不同的版本可能會使用不同的通道）因此，每個通道本身都有下列各項的服務屬性品質：

- **鑑別程式**：可能是空值，也可能是鑑別使用者或通道的**鑑別程式物件**。
- **通道**：提供傳輸服務的類別。
- **壓縮程式**：可能是空值，也可能是可壓縮或解壓縮資料的**壓縮程式物件**。
- **加密程式**：可能是空值，也可能是可加密或解密的**加密程式物件**。
- **目的地**：這個通道的目標，比方說，`SERVER.XYZ.COM`

鑑別程式通常只在設定通道時才會使用。壓縮程式和加密程式通常適用於所有串流。

最簡單的加密程式類型是 `MQeXorCryptor`，它會進行資料的互斥 OR 運算，來加密傳送的資料。這個加密法並不安全，但會使資料變成無法檢視。相對地，`MQe3DESCryptor` 會實作三層 DES。最簡單的壓縮程式類型是 `MQeRleCompressor`，它會以數量取代重複的字元來壓縮資料。另外還有其他鑑別程式、壓縮程式和加密程式，請參閱第33頁的表13。

通道的建立會使用通信協定配接器規格來判斷特定通道所用的鏈結和通信協定。在每個中間節點上，都會搜尋通道定義來解析下個鏈結所需要的定址。當沒有後續的定義存在時，通道會結束，且任何傳送中的訊息都會送交佇列管理程式。

應用程式設計師或管理者無法直接見到通道，他們是佇列管理程式在需要時所建立的。通道會將佇列管理程式鏈結起來，且 `MQSeries Everyplace` 會根據要傳送的資訊來協定和重新協定它們的特性。傳輸程式是利用通道來提供佇列層次通信的 `MQSeries Everyplace` 元件。再說明一次，這些都是應用程式設計師或管理者無法見到的。

當要求確定傳訊時，`MQSeries Everyplace` 會將訊息遞送到應用程式一次，而且只有一次。它是在傳輸結束時，刪除副本之前，先確保訊息已順利從一個佇列管理程式傳到另一佇列管理程式，並經過認可，來做到這一點。在通信失敗事件中，如果沒有收到認可，這時可能會重新傳輸訊息（只遞送一次，不表示只傳輸一次），但不會遞送複本。



---

## 配接器

配接器用來將 MQSeries Everyplace 映射至裝置介面。通道會利用通信協定配接器，透過 HTTP、原生 TCP/IP 及其他通信協定，來執行作業。同樣地，佇列會利用欄位儲存體配接器來形成記憶體或檔案系統這些儲存體子系統的介面。配接器為 MQSeries Everyplace 提供了一種機制，讓它延伸其裝置支援，且可以進行版本化的工作。

檔案描述子是用來識別、載入和啟動配接器的字串。

---

## 撥號連線管理

裝置的撥號網路支援由裝置的作業系統來處理。當未連線的裝置中之 MQSeries Everyplace 試圖使用網路（比方說，必須傳送一則訊息）時，如果網路堆疊不在作用中，作業系統本身會起始遠端存取服務 (RAS)。通常會顯示畫面給使用者來提供撥號連線設定檔。在建立好連線之前，由作業系統來控制。所以，裝置使用者必須確定作業系統有適當的撥號連線設定檔可用。因此，在 MQSeries Everyplace 裝置實作中，不需要有明確的撥號網路支援。

---

## 追蹤

當執行會採取追蹤動作的獨立程式時會啟用追蹤。MQSeries Everyplace 內嵌了若干追蹤呼叫，用來追蹤隨系統和使用者而不同的資訊、警告和錯誤狀況。應用程式也可以直接呼叫追蹤，且可以新增訊息或修改現存的追蹤訊息。所提供的範例追蹤程式可讓選取的訊息顯示、列印和（或）導向到事件日誌中。另外，您還可以撰寫含有其他功能的其他追蹤程式，且可以將它們設計成能夠採用其他方式來建立輸出格式和遞送輸出。

大部份 MQSeries Everyplace 異常狀況都會傳遞到應用程式來處理，應用程式異常狀況處理程式也會將這些遞送給追蹤。

---

## 事件日誌

MQSeries Everyplace 提供有事件日誌機制和介面，可用來記載狀態，比方說，已啟動佇列管理程式。您可以起始日誌的記載，依預設，它會寫入檔案中，不過，您也可以將它截取導向至其他位置。MQSeries Everyplace 事件日誌不會記載訊息資料，且不能用來復原訊息或佇列。

## MQSeries Everyplace 網路

MQSeries Everyplace 網路是連接的裝置和閘道。它們可以跨越多個實體網路，且可以在其間遞送訊息。大體上，它們利用與佇列位置無關的程式設計模型，來提供佇列的同步和非同步存取功能。

### 配置和可調整性

圖5顯示基本的 MQSeries Everyplace 網路拓撲選項。爲了這些目的，假設其中的每一項都同時配置了同步和非同步通信的功能。

最簡單的情況是如 (a) 所示，由一部獨立機器透過本端佇列來支援跨應用程式的同步通信。不過，(b) 比較有趣，它說明同層級對同層級的網路。這時兩個裝置必須採用相同的通信協定，且至少有一個裝置配置了接聽功能，以便回應試圖聯絡的另一裝置。很明顯，在這個簡單的例子中，兩方都必須在相同的網路中，才有可能通信。非同步傳訊可讓應用程式在裝置未連線的情況下執行，同步傳訊只在裝置已實際連線時，才可能發生。

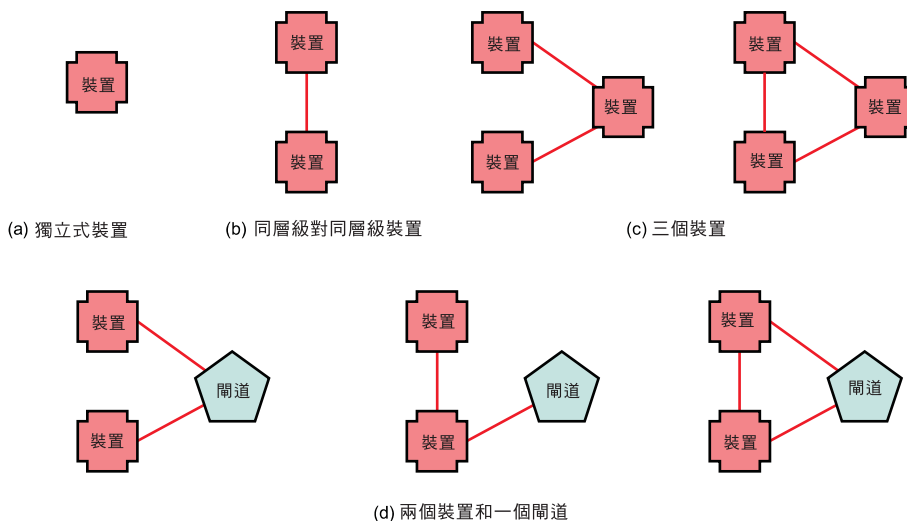


圖 5. 簡式 MQSeries Everyplace 網路

當有三個裝置時，可能性便多了許多。這時可能由其中一個裝置來扮演鏈結裝置的角色，也可能三個裝置全都交互連接起來。如果它們配置成能夠在同層級對同層級的基礎上，彼此互相交換訊息，則剛提供的考量也能夠適用。如果它們是透過第三個裝置來通信，彈性便大了許多。兩個通信裝置不需要採用相同的通信協定，但每一個都必須採用橋接裝置所能瞭解的通信協定。另外，非同步連線可在傳送端和接收端固定不會同時上網的情況下（不過，兩端都必須有一段能夠用到

鏈結裝置的上網時間)，讓訊息能夠傳送。同步通信會要求這三個裝置同時連線。經常不連接的裝置非常適合採用通過某種第三節點的配置，不過，中間媒體必須很容易取得。

第30頁的圖5(d) 顯示有可能利用閘道來鏈結的實際裝置之三個範例。鏈結節點閘道的偏好設定，基礎在於閘道支援多個並行送入的連線要求。顯示的第一個配置是最可能的配置，第二個和第三個配置也都可行，但有些不尋常。請注意，在第三個配置中，只能配置一個路徑給特定的遠端佇列管理程式，因此，雖然看似有兩個路徑，但只能選取其中一個。

較大型的網路可以使用許多閘道，每個閘道各支援若干裝置。閘道可以採用任何方式來交互連接，但如果已定義了完整的交互連線，則不會有任何裝置間的路徑含有超出兩個閘道。圖6.顯示較大型網路的範例。

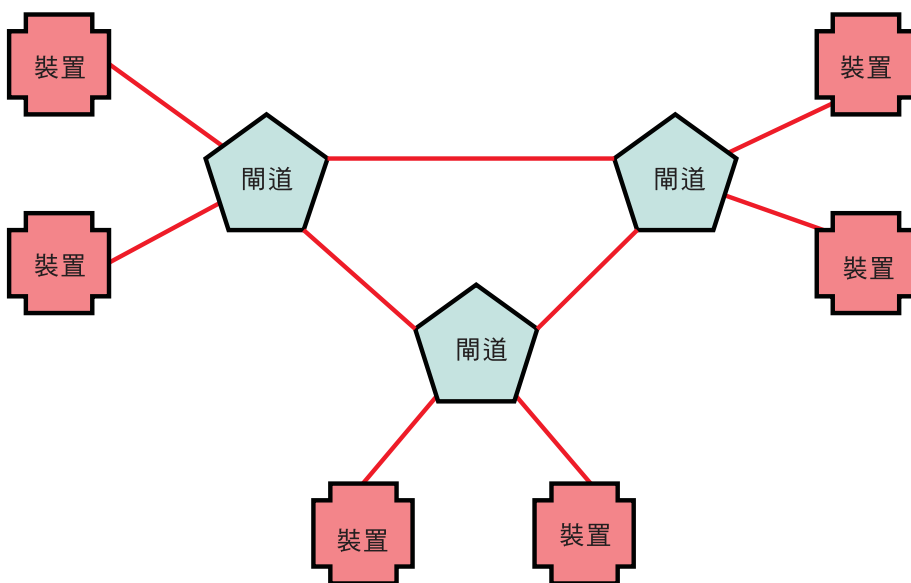


圖 6. 星形 MQSeries Everyplace 網路

## 非同步訊息遞送

當訊息採用非同步方式來放到遠端佇列時，訊息物件會以邏輯方式放在該佇列的本端定義所關聯的支持儲存庫中，同時還會附有目的地佇列管理程式和佇列的名稱，以及符合訊息目標目的地的壓縮程式、鑑別程式和加密程式等特性。當物件以目的地佇列所定義的安全格式儲存到持續性儲存體時，會呼叫物件的傾出方法。佇列管理程式會控制訊息的遞送。它會識別（或建立）含適當特性的通道來

指向下個中繼環節的佇列管理程式，之後，再建立（或重複使用）傳輸程式來指向目標佇列。傳輸程式會傾出物件，並傳輸所產生的位元組字串。請注意，目標佇列管理程式和佇列名稱不是這個訊息串流的一部份。

如果適當的話，會透過通道來加密和壓縮訊息。如果已到達目的地佇列管理程式，則會解密和解壓縮。這時會利用該物件類別的還原方法建立一個新的訊息物件，結果物件會放到目的地佇列中。如果訊息沒有到達目的地佇列管理程式，這時會將它解密和解壓縮，再放到含適當特性的儲存和轉遞佇列中，以進行後續的傳輸。在這兩種情況中，都會以目的地佇列所定義的安全格式將它保存在個別的佇列中。

非同步訊息遞送的一項特性是訊息會傳遞到中繼環節的佇列管理程式，再放到佇列中，以進行後續的傳輸。在中間佇列中，會先依照優先順序，再依照時間戳記順序，來取出訊息。

## 同步訊息遞送

同步訊息遞送類似於上述的非同步情況，不過，佇列管理程式在中繼環節的參與是在低了許多的層次上進行的，其中包括傳輸程式和通道。通道是利用通信協定規格所定義的配接器，在每個中間節點中，採用同層級對同層級的方式來建立，用來識別下個鏈結。在到達最後一個鏈結的尾端時，已沒有相關的檔案描述子，這時訊息會進入佇列管理程式較高的層次，以便進行處理。因此，傳送的節點不會將訊息放到佇列中，而是會沿著通道、通過中繼環節來傳遞它，之後，再將它交給目標佇列管理程式，以將它放在目標佇列中。

進入 MQSeries 的鏈結使用位於閘道的橋接佇列，它會將訊息轉換成 MQSeries 格式。這個機制表示同步的 MQSeries Everyplace 式傳訊有可能利用終止於閘道的動態通道，從裝置傳遞到 MQSeries。這時訊息會從閘道中，即時透過從屬站通道來遞送到 MQSeries 伺服器。到了該處，目的地可能會要求它以非同步方式，沿著 MQSeries 訊息通道來遞送。

透過類似的方式，只能進行同步傳訊的裝置也能夠將訊息傳送給非同步 MQSeries Everyplace 佇列，不過，這時必須有適當的中介者才行。

---

## 安全

MQSeries Everyplace 提供了一組整合的安全特性，既可保護保存在本端環境內的訊息資料，也可以保護在轉送過程中的訊息資料。

MQSeries Everyplace 安全特性所提供的保護有三個不同種類：

- 本端安全 - 訊息（或其他）資料的本端保護

- 佇列型安全 - 起始佇列管理程式和目標佇列之間的訊息保護
- 訊息層次安全 - 在起始者和接受者間之訊息的訊息層次保護

MQSeries Everyplace 在內部使用 MQSeries Everyplace 本端和訊息層次安全，但 MQSeries Everyplace 應用程式也可以使用它們。MQSeries Everyplace 佇列型安全是一個內部服務程式。

三個種類的 MQSeries Everyplace 安全特性全都利用屬性（MQeAttribute 或 descendent）來保護訊息資料。隨著種類的不同，屬性會採用明確或隱含的方式來套用。

每個屬性都含有下列物件：

- 鑑別程式
- 加密程式
- 壓縮程式
- 金鑰
- 目標實體名稱

隨著 MQSeries Everyplace 安全特性種類的不同，這些物件的使用方式也各不相同，不過，不論任何情況，當呼叫附加到某訊息物件的屬性時，都會套用 MQSeries Everyplace 安全特性的保護。當呼叫某 MQSeries Everyplace 訊息的 'dump' 方法時（當使用屬性的 'encodeData' 方法時，比方說，加密和壓縮訊息資料），便會套用這個保護。當呼叫 MQSeries Everyplace 訊息的 'restore' 方法時（當使用屬性的 'decodeData' 方法時，比方說，將訊息資料解壓縮和解密），MQSeries Everyplace 安全特性便不提供保護。

關於 MQSeries Everyplace 第 1.0 版支援的鑑別、加密和壓縮演算法的詳細資料，請參閱表 13。

表 13. 鑑別、加密和壓縮支援

函數	演算法
鑑別	以迷你憑證為基礎（衍生自 WAP 論壇的「WTLS 迷你憑證」）
	驗證 Windows NT/2000 身份
壓縮	LZW
	RLE

表 13. 鑑別、加密和壓縮支援 (繼續)

函數	演算法
加密	三層 DES
	DES
	MARS
	RC4
	RC6
	XOR

## MQSeries Everyplace 本端安全

本端安全會在本端環境內保護 MQSeries Everyplace 訊息（或 MQeFields 或 MQeFields 的後代）資料的安全。方法是建立一個含適當的對稱加密程式和壓縮程式的屬性，建立和設定適當的「金鑰」（提供密碼和密碼片語），再將金鑰明確附加到屬性上，再將屬性附加到 MQSeries Everyplace 訊息上。MQSeries Everyplace 提供 MQeLocalSecure 類別來協助設定本端安全，不過，不論在任何情況中，本端安全的使用者（內部的 MQSeries Everyplace，或 MQSeries Everyplace 應用程式）都必須負責設定適當的屬性，以及管理密碼和密碼片語金鑰。

## MQSeries Everyplace 佇列型安全

佇列型安全適用於同步和非同步訊息。

### 同步佇列型安全

應用程式可以利用同步佇列型安全，將所有訊息安全考量全部交給 MQSeries Everyplace。佇列有鑑別、加密和壓縮等特性，在判斷保護訊息串流（以及持續性儲存體）所需要的安全層次時，會用到這些特性。

當要傳送一則訊息時，會從本端登錄中擷取目標佇列的安全特性。如果沒有這些特性，佇列管理程式會試圖找出目標佇列管理程式的目標特性，並快取它們，以便後續重複使用。如果有指向這個佇列管理程式的通道，則會使用這個通道；如果沒有，則會建立一個新通道。這時會擷取目標佇列屬性。

目標佇列管理程式的通道屬性會依所需服務品質而動態改變。這項改變會遵循任何已建立的規則。規則通常會容許將安全層次升級（比方說，從無保護到弱保護，從弱保護到強保護）。如果通道無法升級，或認為安全層次已經過度（比方說，不需要保護，但可用的通道卻實作強的保護），這時會建立一個新的通道。通道有一個儲存池，可能的話，會重複使用，且它的特性會隨著通訊量的要求而動態變更。當不再需要通道時，會自動將通道摧毀。訊息固定會依照目標佇列特性所定義的安全層次而放在佇列中。

鑑別活動是在通道層次發生的，它將每個訊息的額外負荷縮到最小。同步佇列型安全通常也會搭配對稱的加密程式使用，因為這可以產生快速的加密/解密作業。不過，在這些對稱的情況中，開始時，MQSeries Everyplace 會使用 RSA 不對稱加密法來保護在傳送端和接收端建立共用金鑰所需要的串流。之後，會使用對稱的加密法來保護傳送的資料之機密性。MQSeries Everyplace 會隨著每個通道串流而動態改變金鑰，因此，這個資料的密碼攻擊便更加困難。MQSeries Everyplace 也會在傳送之前產生和附加摘要，並在接收之時重新產生和驗證它，來確保傳送的資料之完整性。

### 非同步佇列型安全

非同步傳訊不同於上述同步的情況，不同之處，在於執行 `putMessage` 時，無法保證能不能存取目標佇列。這時候，佇列管理程式無法立即傳送訊息，並將它放在傳輸佇列中；不過，它會根據目標佇列特性來加密訊息。等能夠傳輸它時，再將它解密，並以適當特性沿著通道來傳送它。因此，訊息固定會受到保護，即使在等待傳輸時也是如此。非同步傳訊需要遠端佇列定義 - 否則，無法判斷目標佇列特性。

在非同步的情況中，不可能進行發訊者和目標之間的鑑別。當鑑別非常重要時，比方說，接受者要判斷訊息的發訊者（決定要接受或確定不拒絕），或發訊者要確定只有接收者本人能夠處理訊息，這時必須使用訊息層次安全。

使用佇列型安全的同時，也可以使用訊息層次安全，不過，這是多餘的，因為訊息資料已受到保護。

## 訊息層次安全

在起始和接收 MQSeries Everyplace 應用程式之間，由訊息層次安全來保護訊息資料。

訊息層次安全是一種應用程式層的服務，起始 MQSeries Everyplace 應用程式必須設置一個訊息層次屬性，並在利用 `putMessage` 來放置訊息到目標佇列時，提供此屬性。接收應用程式也必須設置和傳遞一個相符的訊息層次屬性給接收佇列管理程式，以便在應用程式呼叫 `getMessage` 來取得目標佇列中的訊息時，能夠取得屬性。

訊息層次安全類似於本端安全，它會將屬性應用在訊息物件上。起始應用程式的佇列管理程式會利用 `'dump'` 方法來處理 `putMessage`，這個方法會利用屬性的 `'encodeData'` 方法來保護訊息資料。接收應用程式的佇列管理程式會利用 `'restore'` 方法來處理應用程式的 `getMessage`，這個方法會利用屬性的 `'decodeData'` 方法來復原原始的訊息資料。

MQSeries Everyplace 提供兩個訊息層次安全的替代屬性：

## MQeMAttribute

這用在商業對商業的通信中，這時應用程式層面上會極嚴密地管理相互的信任，且不需要任何可靠的第三者。所有可用的 MQSeries Everyplace 對稱加密程式和壓縮程式選項都可以使用。如同本端安全，屬性金鑰必須預先設定，之後，putMessage 或 getMessage 才能提供它。MQeAttribute 提供簡單而有力的訊息層次保護方法，能夠啟用強的加密法來維護訊息的機密性，而不會有任何公開金鑰基礎架構 (PKI) 上的過度負荷。

## MQeMTrustAttribute

這個屬性提供利用數位簽章和預設公開金鑰基礎架構的更進階的解決方案。它利用 ISO9796 數位簽章/驗證，讓接收端的應用程式能夠建立訊息來自真正傳送者的證明。它會利用提供的屬性之加密程式來保護訊息機密性。SHA1 摘要可以保證訊息的完整性，RSA 加密/解密可確保只有正確在接受者可以還原訊息。如同 MQeMAttribute，所有可用的 MQSeries Everyplace 對稱加密程式和壓縮程式選項都可以使用。爲了最佳化大小而選取使用的憑證，是以 WAP 論壇之「STLS 規格」所提議的「WTLS 認證」爲基礎的迷你憑證。鑑別（驗證簽名）和加密/解密所需要的資訊相互可用性由 MQSeries Everyplace 預設基礎架構提供。

典型的 MQeMTrustAttribute 保護訊息格式如下：

```
RSA-enc{SymKey}, SymKey-enc {Data, DataDigest, DataSignature}
```

其中：

- RSA-enc** : 以正確接受者的公開金鑰來加密的 RSA
- SymKey** 產生的虛擬隨機對稱金鑰
- SymKey-enc** 以 SymKey 來進行對稱加密
- Data** 訊息資料
- DataDigest** 訊息資料摘要
- DigSignature** 發訊者的訊息資料數位簽章

訊息層次安全和佇列層次安全無關。

## 登錄

登錄是佇列管理程式相關資訊的主要儲存庫；每個佇列管理程式都會有一個登錄。每個佇列管理程式都會利用登錄來保留它的：

- 佇列管理程式配置資料
- 佇列定義
- 遠端佇列定義



- 遠端佇列管理程式定義
- 使用者資料（包括與配置相依的安全資訊）

登錄通常會限制為只有合法的佇列管理程式使用者才能存取，且會受到 PIN 保護，不過，透過一個可配置的選項，可以使注重軌跡大小超過安全性的使用者略過這一項。

## MQSeries Everyplace 可鑑別實體

佇列型的安全會利用迷你憑證型的相互鑑別，訊息層次保護會利用數位簽章，兩者都引發了「可鑑別實體」這個概念。在相互鑑別中，通常是考慮兩位使用者（人）之間的鑑別；不過，一般而言，傳訊作業不含使用者的概念。這個概念通常是在應用程式層次來管理的，也就是說，由傳訊服務的使用者來管理。MQSeries Everyplace 特別將使用者的「鑑別目標」概念取出成「可鑑別實體」。這不排除可鑑別實體可能是人，不過，這會是應用程式選取的映射。從內部來看，MQSeries Everyplace 會定義所有能以可鑑別實體來產生或成為迷你鑑別相依服務之目標的佇列管理程式。另外，MQSeries Everyplace 也會定義佇列來使用迷你憑證型鑑別程式作為可鑑別實體。因此，支援這些服務程式的佇列管理程式可以有一個可鑑別實體（佇列管理程式）或一組可鑑別實體（佇列管理程式和每個使用憑證型鑑別程式的佇列）。

## 專用登錄和憑證

每個可鑑別實體都要有自己的憑證才有用。這帶來了兩項挑戰。首先，是如何執行登錄來取得憑證；其次，是要用安全方法在哪裡管理憑證。傳統上，相較於基礎的密碼化技術，這些是更難克服的挑戰。MQSeries Everyplace 提供了若干預設服務程式，讓可鑑別實體能夠執行自動登錄、專用登錄（基礎登錄的相依項）來啟用可鑑別實體專用憑證的安全管理，以及啟用公用登錄（也是基礎登錄的相依項）來管理一組公用憑證。專用登錄提供基本登錄，其中含有安全或密碼記號的許多特性，比方說，它可以是迷你憑證這類公用物件和專用金鑰這類專用物件的安全儲存庫。它提供一種只讓授權使用者存取專用物件的機制。它利用專用物件絕對不會離開專用儲存庫的方式來支援各服務程式（比方說，數位簽章、RSA 解密）。它提供一個通用介面來隱藏基礎的裝置支援，目前這個基礎裝置支援只以本端檔案系統為限，但將來可能會加以延伸，以映射至可攜的記號。

## 自動登錄

MQSeries Everyplace 提供支援自動登錄的預設服務程式。當配置可鑑別實體時，會自動觸發這些服務程式，比方說，當啟動佇列管理程式之時，或定義新佇列時。這兩種情況都會觸發登錄，並建立新憑證，儲存在可鑑別實體的專用登錄中。自動登錄步驟包括產生新的 RSA 金鑰配對，在專用登錄中保護和儲存專用金鑰；以及將「新憑證」要求中的公開金鑰套裝到預設的迷你憑證伺服器中。假設

已配置好且可以使用迷你憑證伺服器，它會隨著本身的迷你憑證來傳回可鑑別實體的新迷你憑證，這些憑證與所保護的專用金鑰都會儲存在可鑑別實體的專用登錄中，成為它的新憑證。自動登錄提供簡單機制來建立可鑑別實體的憑證，但在訊息層次的保護（MQeMTrustAttribute，參閱前文）上，也需要能夠存取正確接收者的公開金鑰（迷你憑證）。

## 公用登錄和憑證複製

MQSeries Everyplace 提供了使 MQSeries Everyplace 元件之間的可鑑別實體公用憑證（迷你憑證）能夠共用的預設服務程式。這些服務程式是 MQeMTrust 型訊息層次安全的必備條件。MQSeries Everyplace 公用登錄提供迷你憑證的公用可存取儲存庫。這類似於行動電話的個人電話目錄服務，不同之處，在於不是電話號碼，而是一組最常聯絡的可鑑別實體之迷你憑證。這個公用登錄的服務不完全是被動的。如果要求公用登錄提供沒有保存的迷你憑證，且公用登錄配置了有效的起始伺服器元件，它會自動試著從起始伺服器的公用登錄中提取所要求的迷你憑證。這些服務可用來提供智慧型自動化迷你憑證複製服務，以促進正確的迷你憑證在恰當時間裡的可用性。

## 登錄服務的應用

MQSeries Everyplace 佇列管理程式的設計是要利用專用和公用登錄服務的好處，這些服務的存取活動沒有限制。MQSeries Everyplace 解決方案可能會想定義和管理自己的可鑑別實體，比方說，使用者。之後，可以利用專用登錄服務來自動登錄和管理新的可鑑別實體之憑證，利用公用登錄服務，將公用憑證提供到任何有需要之處。所有登錄的可鑑別實體都可以作為 MQeMTrustAttribute 所保護之訊息層次服務的起始者或接受者。

## 預設迷你憑證發行服務

MQSeries Everyplace 提供可配置來滿足專用登錄自動登錄要求的預設迷你憑證發行服務。解決方案可以透過 MQSeries Everyplace 所提供的工具來設定和管理迷你憑證發行服務，將迷你憑證發給一組受到審慎控制的實體名稱。這個發行服務的特性如下：

- 管理已登錄的可鑑別實體組
- 發行迷你憑證（以 WAP WTLS 迷你憑證為基礎的迷你憑證）
- 管理「迷你憑證儲存庫」

迷你憑證發行服務管理者可以利用 MQSeries Everyplace 所提供的工具來登錄特定實體之實體名稱及登錄位址，及定義單次使用的憑證要求 PIN，而將迷你憑證發行授權給這個特定的實體。通常是在離線檢查作業已驗證了要求者的確實性之後，才會完成這個動作。憑證要求 PIN 會交給正確的使用者（比方說，採用類似於發出新金融卡時，將金融卡 PIN 交給使用者的方式）。之後，可以配置專用登錄的

使用者（比方說，MQSeries Everyplace 應用程式或 MQSeries Everyplace 佇列管理程式）在啟動時提供這個憑證要求 PIN。當專用登錄觸發自動登錄時，迷你憑證發行服務會驗證所產生的新憑證要求（根據呈現的實體名稱和憑證要求 PIN 及其預先登錄的值是否相符），發出新的迷你憑證，並重設登錄的憑證要求 PIN，因此，它無法重複使用。新迷你憑證所要求的所有自動登錄都是在安全通道中處理的。

迷你憑證發行服務所發出的迷你憑證組，保存在發行服務本身的登錄中。當重新發出迷你憑證（比方說，因過期之故）時，會將過期的迷你憑證保存起來。

## 安全介面

有一個提供的選用性介面，可來自訂安全管理程式來實作。安全管理程式可以利用各個方法來授權或拒絕關聯於下列各項的要求：

- 類別別名的新增或移除
- 配接器的定義
- 檔案描述子的映射
- 通道指令的處理

---

## 配置和自訂

### 規則

規則是一些 Java 類別，用來自訂發生各種狀態變更時，MQSeries Everyplace 所要採取的行為。在必要之處，會提供預設規則，但這些預設規則可以改成應用程式或安裝結構專用的規則，以符合客戶的需求。所支援的各種規則類型，不同之處，在於觸發它們的方式，而不是它們能夠執行的動作；規則含有邏輯，因此，可以執行非常多種功能。

#### 屬性規則

每當試圖要變更狀態時，都會將控制權交給這個規則類別，比方說，變更：

- 鑑別程式
- 壓縮程式
- 加密程式

規則通常會接受或拒絕變更。

#### MQSeries 橋接規則

當 MQSeries Everyplace 至 MQSeries 的橋接程式碼有狀態的改變時，這些規則會取得控制權。有一個個別的橋接規則類別，可用來判斷下列各項目：

- 當接聽器無法將訊息遞送到 MQSeries Everyplace 時，當訊息來自 MQSeries 時，要如何處理訊息。比方說，因為訊息太大，或佇列不存在。
- 在建立伺服器案例之後，應該立即啟動狀態橋接管理的物件
- 當橋接發現 MQSeries 中的 Sync Q（毀損復原所採用的持續性儲存庫）有錯誤時，要如何處理。預設規則只會顯示問題。
- 如何將 MQSeries Everyplace 訊息轉換成 MQ 訊息，反之亦然。進行 MQSeries Everyplace 和 MQ 訊息間之訊息轉換的轉換程式不是從任何 MQeRule 類別衍生而來，相反地，它們必須實作 MQeTransformerInterface 介面。另外，轉換程式會採取類似規則的行動，當訊息需要轉換格式時會呼叫它。

### **RAS 撥號程式規則**

當 RAS 撥號程式有狀態變更時，都會將控制權交給這個規則類別，比方說：

- 如果要呼叫的號碼沒有連線，要如何處理
- 超出錯誤臨界之後，要如何處理
- 試圖撥號，且根據日期時間，只應使用某些類型的連線。比方說，如果在離峰時間，只使用電話

### **佇列規則**

每當發生相關佇列的狀態變更時，都會將控制權交給這個規則類別，比方說：

- 新增訊息到佇列中。比方說，如果要知道是否已超出臨界值（訊息數目、訊息大小、無效優先順序）
- 指定或變更的佇列特性
- 開啓或關閉佇列
- 要刪除的佇列

### **佇列管理程式規則**

每當發生佇列管理程式的狀態變更時，都會將控制權交給這個規則類別，比方說：

- 開啓佇列管理程式。比方說，啟動背景計時器執行緒，以產生合時間的動作
- 已結束佇列管理程式。比方說，終止背景計時器執行緒
- 新增佇列

## 連線樣式

MQSeries Everyplace 可以支援主從架構（從屬站/伺服器<sup>10</sup>）和（或）同層級對同層級的作業。從屬站能夠起始與伺服器之間的通信；伺服器只能回應從屬站所起始的要求。在同層級對同層級的作業中，兩點可以依任何方向來起始串流。這些連線樣式需要能夠取得和使用不同的 MQSeries Everyplace 元件。涉及的元件包括：

- **通道接聽器**：接聽送入的連線要求。
- **通道管理程式**：支援端點之間的邏輯多重並行通信管線。
- **佇列管理程式**：提供傳訊和佇列功能來支援應用程式。

表14.顯示這些元件和連線樣式之間的關係。主從架構連線樣式說明 MQSeries Everyplace 可採用從屬站或伺服器模式來操作的情況。Servlet 選項說明將 MQSeries Everyplace 配置成 HTTP Servlet，HTTP 伺服器本身負責接聽送入的連線要求的情況。

表 14. 連線樣式

	佇列管理程式	通道管理程式	通道接聽器
從屬站	是		
主從架構	是	是	是
同層級	是		
伺服器	是	是	是
Servlet	是	是	

**MQSeries Everyplace 通道管理程式或 MQSeries Everyplace 通道接聽器的使用會因授權之目的，判斷 MQSeries Everyplace 案例是一個閘道。**

MQSeries Everyplace 應用程式不會直接知道佇列管理程式所用的連線樣式。不過，樣式非常重要，它會影響到各方能夠使用哪些資源、哪些佇列管理程式可連接其他佇列管理程式、裝置或閘道的 MQSeries Everyplace 軌跡，及哪些連線可以並行存在。

### 同層級對同層級連線

同層級對同層級通道包括單一通道的通道管理程式和通道接聽器的功能。當在兩個佇列管理程式之間建立同層級對同層級通道時，一個佇列管理程式要扮演接聽器的角色，另一個佇列管理程式要扮演連線起始者的角色。以同層級對同層級方式來連接的佇列管理程式可以起始多個同層級對同層級連線來連接其他佇列管理程式，但它只能回應一個送入的連線要求，之後，必須等待這個同層級對同層級

10. 在這一節裡，「從屬站」和「伺服器」兩詞採一般意義，而不是在 MQSeries 所專用的語意。

通道關閉，才能回應另一個這種要求。透過任何同層級對同層級通道，兩個參與的佇列管理程式都可以起始動作，因此，比方說，每個佇列管理程式中的應用程式都能存取另一佇列管理程式中的佇列。

同層級對同層級通道無法穿越防火牆，因為防火牆不一定會接受送入的連線要求之目標。

## 主從架構連線

主從架構連線樣式所用的標準通道沒有接聽的能力，它依賴於在伺服器的獨立接聽器，伺服器則需要透過通道管理程式來處理多重並行通道。從屬站會起始連線要求和伺服器回應。伺服器通常可以處理多個從屬站送來多重送入要求。從屬站則可以經由標準通道來取伺服器中的資源。如果伺服器中的應用程式需要同步存取從屬站中的資源，這時需要有保留角色的第二通道。不過，由於標準通道本身是雙向的，從伺服器傳輸佇列指向從屬站的訊息，會透過所起始的標準通道（主從架構）來遞送到從屬站。

從屬站可以同時是多部伺服器的從屬站。請注意，通道管理程式不需要支援這種配置，因為通道管理程式會處理多重埠通道。

主從架構連線樣式通常適合通過防火牆，因為送入連線的目標通常會以防火牆所能接受的方式來識別。

## 多重連線樣式

單一佇列管理程式可以起始同層級對同層級連線或主從架構連線，也能夠作為伺服器或對等的層級來回應。在這個情況下，同層級通道接聽器和標準通道接聽器必須有不同的埠號。

---

## 類別

MQSeries Everyplace 為某些功能提供了一種類別的選擇，讓 MQSeries Everyplace 的行為可以自訂，以符合特定的應用程式需求。在某些情況下，會產生類別介面的文件，以便於發展其他替代方案。表15彙總這些可能。您可以明確地識別類別，也可以利用別名來識別類別。

表 15. 類別選項

類別	提供替代方案	產生介面文件
管理	否	是
鑑別程式	是	否
通信配接器	是	是
通信樣式	是	否

表 15. 類別選項 (繼續)

類別	提供替代方案	產生介面文件
壓縮程式	是	否
加密程式	是	否
事件日誌	提供範例	是
訊息	否	是
佇列儲存體	是	否
規則	提供預設類別	是
追蹤	提供範例	是

## 應用程式載入

當配置 MQSeries Everyplace 作為從屬站（或同層級）來作業時，起始的應用程式要負責將任何其他應用程式載入 JVM 中。您可以利用標準 Java 機能來做到這一點，也可以利用 MQSeries Everyplace 所提供的類別載入器。因此，在相同的 JVM 中，可以同時針對單一佇列管理程式來執行多個應用程式。另外，您也可以使用多個 JVM，但每個 JVM 都會要求它自己的佇列管理程式，且每一個佇列管理程式都必須有唯一的名稱。

當配置 MQSeries Everyplace 作為伺服器時，MQSeries Everyplace 本身便是一個起始應用程式。MQSeries Everyplace 支援一份預載的類別清單，且在載入佇列管理程式本身之前，會先依次載入這些類別。





---

## 第6章 MQSeries Everyplace 和 MQSeries 網路

雖然 MQSeries Everyplace 網路可以單獨存在，不需要有 MQSeries 伺服器或網路，但實際上，MQSeries Everyplace 通常用來彌補現存 MQSeries 安裝結構之不足，使它的功能延伸到新平台和裝置，或提供佇列或訊息型的安全或同步傳訊這類進階功能。從 MQSeries Everyplace 應用程式的角度來看，MQSeries 佇列和佇列管理程式可以是簡單附加的遠端佇列和佇列管理程式。不過，由於這些佇列無法直接透過 MQSeries Everyplace 動態通道和 MQSeries Everyplace 佇列管理程式來存取，而需要 MQSeries Everyplace 閘道的介入，因此，仍有許多功能上的限制存在。閘道可以直接或間接透過 MQSeries 從屬站通道來傳送訊息給多個 MQSeries 佇列管理程式。如果是間接連線，這時訊息會透過 MQSeries 從屬站通道來傳送給中間 MQSeries 佇列管理程式，再繼續透過 MQSeries 訊息通道來傳送給目標佇列管理程式。

指定給 MQSeries Everyplace 的 MQSeries 應用程式之訊息，會照常將位址設定於 MQSeries Everyplace 佇列管理程式和佇列，且會定義好 MQSeries 遞送（遠端佇列管理程式定義），使 MQSeries Everyplace 訊息能到達特定的 MQSeries 傳輸佇列。這時不會依正常情況來定義傳輸佇列的 MQSeries 通道，相反地，MQSeries Everyplace 閘道會從這些佇列中拉出訊息，並確保它們能遞送到 MQSeries Everyplace 目的地。使用的傳輸佇列數目（也就是遞送數）是可配置的，且應該加以設定，以反映遞送的訊息量。

---

### MQSeries 介面

MQSeries Everyplace 的架構支援在 MQSeries Everyplace 和其他傳訊系統之間有一或多個選用性橋接的概念。

MQSeries Everyplace 第 1.0 版只支援一個這種橋接，也就是形成 MQSeries Everyplace 和 MQSeries 網路之介面的 *MQSeries* 橋接。這個橋接利用 MQSeries Java 從屬站作為介面來通往一或多個 MQSeries 佇列管理程式，使訊息能夠在 MQSeries Everyplace 和 MQSeries 之間流動，反之亦然。MQSeries Everyplace 第 1 版建議每個閘道採用一個這種橋接，再由每個橋接與多個 *MQSeries* 佇列管理程式 *Proxy* 相連結（MQSeries 佇列管理程式的定義）。每個與 MQSeries Everyplace 通信的 MQSeries 佇列管理程式都需要一個佇列管理程式 *Proxy* 定義。其中的每個定義都可以有一或多個相關聯的從屬站連線服務來分別代表與單一 MQSeries 佇列管理程式的連線。每個連線都會使用不同的 MQSeries 伺服器連線來連到佇列管理程式，且可以選用一組不同的內容，如使用者跳出程式或埠等。

一個閘道從屬站連線服務程式可以有多個接聽器，這些接聽器利用這個閘道服務程式來連接 MQSeries 佇列管理程式並將 MQSeries 的訊息擷取至 MQSeries Everyplace。一個接聽器只會利用一個服務程式來建立其連線，且每個接聽器都連接到 MQSeries 佇列管理程式中的單一傳輸佇列。每個接聽器都會透過它的母閘道佇列管理程式，將單一 MQSeries 傳輸佇列中的訊息移到 MQSeries Everyplace 網路中的任何位置。因此，單一閘道佇列管理程式可以如漏斗般，將多個 MQSeries 訊息來源傳送到 MQSeries Everyplace 網路中。

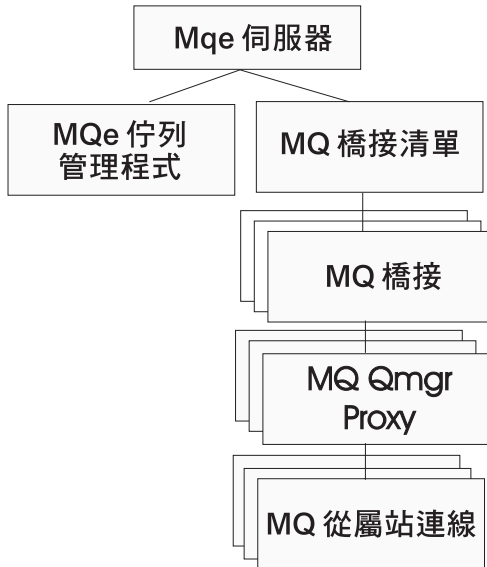


圖 7. MQSeries 橋接物件階層

當依另一方向將 MQSeries Everyplace 的訊息移到 MQSeries 時，閘道佇列管理程式會配置一或多個橋接佇列物件。每個橋接佇列物件都可以直接連到任何佇列管理程式，並將它的訊息傳送到目標佇列中。在這個方式下，閘道可以直接或間接將單一 MQSeries Everyplace 佇列管理程式所遞送的 MQSeries Everyplace 訊息分派給任何 MQSeries 佇列管理程式。橋接物件有表16.所顯示的內容。

表 16. 橋接物件內容

內容	說明
橋接名稱	橋接清單
執行狀態	狀態：執行中或已停止

橋接物件及其他閘道物件的啟動和停止，可以獨立於 MQSeries Everyplace 佇列管理程式來進行。如果啟動（或停止）這樣的閘道，這時動作也會套用到所有子項

上（所有橋接、佇列管理程式 Proxy、從屬站連線及傳輸佇列接聽器）。橋接物件有第48頁的表17.所顯示的內容。

表 17. 橋接內容

內容	說明
類別	橋接類別
預設轉換程式	在目的地佇列沒其他相關聯的轉換程式類別時，作為將 MQSeries Everyplace 訊息轉換至 MQSeries（反之亦然）的預設類別
心跳間隔	針對橋接物件來執行動作時，所採用的基本計時單位
名稱	橋接物件的名稱
執行狀態	狀態：執行中或已停止
啟動規則類別	啟動橋接物件時所使用的規則類別
MQSeries 佇列管理程式 Proxy 子項	這個橋接所擁有的所有「佇列管理程式 Proxy」之清單

在簡單情況下，您可以利用預設轉換程式（規則）來處理所有訊息轉換。另外，您也可以根據置換這個預設值的個別接聽器（MQSeries 至 MQSeries Everyplace 的訊息）來設定轉換程式。如果要有更明確的控制，您可以根據目標佇列，利用閘道中的橋接佇列定義來設定規則；這個方式適用於 MQSeries Everyplace 和 MQSeries 目標佇列兩者。

MQSeries 佇列管理程式 Proxy 會保留單一 MQSeries 佇列管理程式的特定內容。表18.顯示 Proxy 內容。

表 18. MQSeries 佇列管理程式 Proxy 內容

內容	說明
類別	MQSeries 佇列管理程式 Proxy 類別
MQSeries 主電腦名稱	建立透過 Java 從屬站類別來連接 MQSeries 佇列管理程式的連線時所採用的 IP 主電腦名稱。如果沒有指定的話，會假設 MQSeries 佇列管理程式在橋接的同一部機器上，且使用 Java 連結
MQSeries 佇列管理程式 Proxy 名稱	MQSeries 佇列管理程式的名稱
擁有者橋接的名稱	擁有這個 MQSeries 佇列管理程式 Proxy 之橋接物件的名稱
執行狀態	狀態：執行中或已停止
啟動規則類別	啟動 MQSeries 佇列管理程式物件時所使用的規則類別
從屬站連線子項	這個 Proxy 所擁有的所有「從屬站連線」物件之清單

橋接從屬站連線服務程式定義負責保留建立 MQSeries 佇列管理程式的連線時所需要的詳細資訊。表19.顯示連線內容。

表 19. 從屬站連線服務程式內容

內容	說明
配接器類別	作為閘道配接器的類別
CCSID*	要用的整數 MQSeries CCSID 值
類別	橋接從屬站連線服務程式類別
最大連線閒置時間	連線終止之前，容許閒置的最長時間
MQSeries 密碼*	Java 從屬站所用的密碼
MQSeries 埠*	透過 Java 從屬站類別來建立與 MQSeries 佇列管理程式的連線時，所採用的 IP 埠號。如果沒有指定的話，會假設 MQSeries 佇列管理程式在橋接的同一部機器上，且使用 Java 連結
MQSeries 接收跳出程式類別*	用以符合從屬站通道另一端所使用的接收跳出程式；這個跳出程式有相關聯的字串，可讓資料傳送到跳出程式碼
MQSeries 安全跳出程式類別*	用以符合從屬站通道另一端所使用的安全跳出程式；這個跳出程式有相關聯的字串，可讓資料傳送到跳出程式碼
MQSeries 傳送跳出程式類別*	用以符合從屬站通道另一端所使用的傳送跳出程式；這個跳出程式有相關聯的字串，可讓資料傳送到跳出程式碼
MQSeries 使用者 ID*	Java 從屬站所用的使用者 ID
從屬站連線服務名稱	MQSeries 機器中的伺服器連線通道名稱
擁有的佇列管理程式 Proxy 之名稱	擁有的佇列管理程式 Proxy 之名稱
啟動規則類別	啟動橋接從屬站連線服務程式物件時所使用的規則類別
同步佇列名稱	橋接為了同步而使用的 MQSeries 佇列之名稱
同步佇列清除器規則類別	在同步佇列中找到訊息時所使用的規則類別
執行狀態	狀態：執行中或已停止
擁有的橋接的名稱	擁有這個從屬站連線之橋接物件的名稱
MQ XmitQ Listener 子項	使用這個從屬站連線的所有接聽器物件清單
*關於這些參數的詳細資料，請參閱 <i>MQSeries 使用 Java</i> 文件。	

配接器類別用來從 MQSeries Everyplace 傳送訊息到 MQSeries，同步佇列用來追蹤這個程序的狀態。在復原狀況中，會使用它的內容來保證確定傳訊；在正常關機之後，佇列會變成空的。它可讓多個從屬站連線和多個橋接定義共用，不過，接收、傳送和安全跳出程式必須相同。視使用中的接聽器內容而定，這個佇列也

可用來儲存從 MQSeries 移至 MQSeries Everyplace 之訊息的相關狀態。當在同步佇列中找到訊息時，會使用*同步佇列清除器規則類別*，指出 MQSeries Everyplace 確認訊息失敗。

最大連線閒置時間用來控制橋接從屬站連線服務至其 MQSeries 系統所維護的 Java 從屬站連線儲存池。當 MQSeries 連線閒置時，會因沒有使用而啟動計時器，如果計時器在重新使用連線之前到期，這時會捨棄閒置的連線。MQSeries 連線的建立作業成本很高，但這個程序可確保能夠有效重複使用它們，且不會耗用過多的資源。零值表示不應該使用連線儲存池。

將訊息從 MQSeries 移至 MQSeries Everyplace 的接聽器物件有表20.所示的內容。

表 20. 接聽器內容

內容	說明
類別	接聽器類別
無法傳送的信件佇列名稱	用來保留 MQSeries 到 MQSeries Everyplace 無法遞送的訊息之佇列
接聽器狀態儲存配接器	用來儲存狀態資訊的配接器之類別名稱
接聽器名稱	提供訊息的 MQSeries XMIT 佇列的名稱
擁有的從屬站連線服務名稱	從屬站連線服務名稱
執行狀態	狀態：執行中或已停止
啟動規則類別	啟動接聽器物件之後所用的規則類別
傳輸程式類別	用來確定 MQSeries 訊息至 MQSeries Everyplace 之轉換的規則類別
未遞送的訊息規則類別	用來確定 MQSeries 至 MQSeries Everyplace 的訊息無法遞送時之動作的規則類別
等待訊息的秒數	可用來控制異常情況中之接聽器效能的進階選項

未遞送的訊息規則類別負責確定 MQSeries 至 MQSeries Everyplace 的訊息無法遞送時所要採取的動作。它通常放在 MQSeries 系統的無法傳送的信件佇列。

爲了要提供訊息的確定遞送，接聽器類別會使用接聽器狀態儲存庫配接器來儲存狀態資訊，可能是在 MQSeries Everyplace 系統中，也可能是在 MQSeries 系統的同步佇列中。

爲了要完成橋接的配置，遠端佇列管理程式和遠端佇列定義都不可或缺。遠端 MQSeries Everyplace 佇列管理程式的遠端佇列管理程式定義遵循標準的 MQSeries Everyplace 作法；遠端 MQSeries 佇列管理程式會將通道定義設爲空值，以表示不

使用正常的 MQSeries Everyplace 動態通道 - 相反地，這時會依上述詳細資料來定義通往 MQSeries 佇列管理程式的連線。

MQSeries Everyplace 佇列的遠端佇列定義也會遵循標準的作法；不過，對於 MQSeries 佇列而言，它會明顯改變 MQSeries Everyplace 佇列所採用者。表21. 顯示 MQSeries 遠端佇列的內容。

表 21. MQSeries 遠端佇列內容

內容	說明
別名	佇列的替代名稱
鑑別程式	必須是空值
類別	物件類別
從屬站連線	要使用的從屬站連線服務的名稱
壓縮程式	必須是空值
加密程式	必須是空值
期限	傳送至轉換程式
最大訊息大小	傳送至收件人規則
模式	必須是同步
MQ 佇列管理程式 Proxy	訊息應該先送達的 MQSeries 佇列管理程式名稱
MQSeries 橋接	將訊息傳送至 MQSeries 的橋接名稱
名稱	MQSeries Everyplace 用來識別遠端 MQSeries 佇列的名稱
擁有的佇列管理程式	擁有定義的佇列管理程式
優先順序	訊息要用的優先順序（除非訊息值已加以置換）
遠端 MQSeries 佇列名稱	遠端 MQSeries 佇列的名稱
規則	佇列作業所用的規則類別
佇列管理程式目標	擁有佇列的 MQSeries 佇列管理程式
轉換程式	將訊息從 MQSeries Everyplace 格式轉換成 MQSeries 格式的轉換程式類別名稱
類型	MQSeries 橋接佇列

加密程式、鑑別程式和壓縮程式類別會定義一組佇列屬性，用以指定傳送到這個佇列的任何訊息之安全層次。從開始傳送 MQSeries Everyplace 中之訊息，直到訊息送達 MQSeries 橋接佇列，訊息至少都受到佇列層次的安全保護。當 MQSeries 橋接佇列將訊息傳送到 MQSeries 系統時，便不會套用這些安全層次，在這個轉送中，會使用從屬站連線的安全傳送和接收跳出程式。不會進行任何安全檢查來確定佇列層次安全的維護。

MQSeries 橋接佇列只能採用同步方式；因此，非同步的應用程式必須透過「MQSeries Everyplace 儲存和轉遞/起始伺服器佇列」，或透過非同步遠端佇列定義，來將訊息送到這些佇列。

閘道的管理和正常 MQSeries Everyplace 佇列管理程式採用相同的方式 - 利用管理訊息。新的訊息類別會配合被管理的物件來定義。第25頁的表11.顯示閘道管理訊息類別。

---

## 訊息轉換

以 MQSeries 為目的地之 MQSeries Everyplace 訊息會通過橋接，並利用預設的轉換程式或目標佇列專用的轉換程式，來轉換成 MQSeries 格式。自訂的轉換程式彈性較大，比方說，利用 MQSeries Everyplace 訊息物件類別的子類別來表示通過 MQSeries Everyplace 網路的特定類型的訊息，便是好的作法。在閘道上，轉換程式可以利用欄位和適當的 MQSeries 值的任何對映，將訊息轉換成 MQSeries 格式，也可以新增特定資料來表示子類別的重要性。

從 MQSeries Everyplace 到 MQSeries 的預設轉換程式無法利用子類別資訊，但依照設計，它在許多情況中都非常有用。它有下列特性：

- **從 MQSeries Everyplace 到 MQS 的訊息串流：**

從 MQSeries Everyplace 到 MQSeries 的預設轉換程式會結合 MQeMQMsgObject 類別來運作。這個類別是您可以在 MQ 訊息標題中找到的所有欄位的表示法。您的應用程式可以利用 MQeMQMsgObject，以各種 set() 方法來設定值（比方說，優先順序）。因此，當 MQeMQMsgObject（或衍生自 MQeMQMsgObject 類別的物件）通過預設的 MQSeries Everyplace 轉換程式時，預設的轉換程式 (MQeBaseTransformer) 會從 MQeMMsgObject 內取得值，並在 MQSeries 訊息內設定對應的值（比方說，將優先順序值複製到 MQSeries 訊息）。

如果要傳送的訊息不是 MQeMQMsgObject，且不是衍生自 MQeMQMsgObject 類別，則整個 MQSeries Everyplace 訊息都會複製到 MQSeries 訊息的主體中（*funneled*）。這時會設定 MQSeries 訊息標題內的訊息格式欄位來指出 MQSeries 訊息以 MQSeries Everyplace "funneled" 格式來保留訊息。

- **MQSeries 至 MQSeries Everyplace 的訊息串流：**

MQSeries Everyplace 的 MQSeries 訊息的處理方式，類似於另一方向的訊息傳送方式。預設的轉換程式會檢查 MQSeries 標題的訊息類型欄位，並據此來採取行動。

如果 MQSeries 標題指出 "funneled" MQSeries Everyplace 訊息，則 MQSeries 訊息主體會依照原始 MQSeries Everyplace 訊息來重新建構訊息主體，再將它公佈到 MQSeries Everyplace 網路中。



如果訊息不是 “funneled” MQSeries Everyplace 訊息，則會取出 MQSeries 訊息標題內容，放到 MQeMQMsgObject 物件中。MQSeries 訊息主體會當作簡單的位元組欄位來處理，也會放到 MQeMQMsgObject 物件中。之後，再將 MQeMQMsgObject 公佈到 MQSeries Everyplace 網路。

此 MQeMQMsgObject 類別和預設的轉換程式行為表示：

- MQSeries Everyplace 訊息可跨越 MQSeries 網路直到 MQSeries Everyplace 網路，不需要改變。
- MQSeries 訊息可跨越 MQSeries Everyplace 網路直到 MQSeries 網路，不需要改變。
- MQSeries Everyplace 應用程式可以驅動任何現存的 MQSeries 應用程式，不需要改變 MQSeries 應用程式。

---

## 功能

MQSeries Everyplace 佇列管理程式能夠為同步的 MQSeries Everyplace 放置傳訊作業啓用 MQSeries 遠端佇列；但所有其他傳訊作業都必須是非同步。

MQSeries Everyplace 管理訊息不能傳送到 MQSeries 佇列管理程式。那裡沒有 AdminQ，且管理訊息格式不同於 MQSeries 所用的格式。

---

## 相容性

MQSeries Everyplace 網路可以獨立於 MQSeries 而存在，但在許多情況下，這兩個產品都必須符合應用程式需求。MQSeries Everyplace 可以整合到現存的 MQSeries 網路中，其相容性包含下面所摘要的各個方面：

- **定址和命名：**
  - 相同的定址語意，利用佇列管理程式/佇列位址
  - 通用 ASCII 名稱空間。
- **應用程式：**MQSeries Everyplace 能夠支援現存的 MQSeries 應用程式，不需要改變應用程式。
- **通道：**MQSeries Everyplace 閘道會使用 MQSeries 從屬站通道。
- **訊息交換和內容：**
  - MQSeries Everyplace 和 MQS 之間的訊息交換
  - 隱藏訊息網路（MQSeries Everyplace 或 MQSeries 的訊息都可以跨越另一網路，不需要變更）。
  - MQSeries 訊息標題中之識別欄位的相互支援
  - 單次確定訊息遞送

- **範例應用程式：** MQSeries 明信片 and MQSeries Everyplace 明信片<sup>11</sup> 應用程式的互用性

MQSeries Everyplace 第 1 版沒有支援 MQSeries 的所有功能。在環境、作業系統和通信考量之外，下面詳列一些更重要的差異。不過，請注意，在 MQSeries Everyplace 內，許多應用程式作業都能利用 MQSeries Everyplace 特性，透過替代方式來完成，或藉著利用子類別、取代提供的類別，或利用規則、介面和其他內建在產品中的自訂特性來完成。

- 沒有叢集支援。
- 沒有分送清單支援。
- 沒有分組/分段的訊息。
- 沒有平衡資料流量/暖待用功能。
- 沒有參照訊息。
- 沒有報告選項。
- 沒有共用佇列支援。
- 沒有觸發。
- 沒有工作單元支援，沒有 XA 對等關係。

可調整性和效能特性不同。

---

## 確定遞送

雖然 MQSeries Everyplace 和 MQSeries 都提供確定遞送，但它們提供的確定分屬不同的層次。當訊息從 MQSeries Everyplace 流向 MQSeries 時，必須結合 `putMessage` 和 `confirmPutMessage` 兩者，才能確定訊息的傳送（請參閱第23頁的『佇列管理程式作業』）。當訊息從 MQSeries 流向 MQSeries Everyplace 時，必須將 MQSeries 訊息定義為具有持續性，才能確定這項轉送作業。

---

11. 這個應用程式不包含在第 1.0 版中，但您可以在全球資訊網的 MQSeries 軟體下載網站 (<http://www-4.ibm.com/software/ts/MQSeries/>) 中找到它。



---

## 第7章 應用程式和公用程式

註：這些應用程式和公用程式不包含在第 1.0 版中，但在全球資訊網的 MQSeries 軟體下載網站 (<http://www-4.ibm.com/software/ts/MQSeries/>) 中，您將能找到它。

---

### 明信片

「明信片」是一個 MQSeries Everyplace 應用程式，可用來驗證獨立式 MQSeries Everyplace 網路作業，或 MQSeries Everyplace 和 MQSeries 網路的交互作業。「明信片」是一個 Java 應用程式，可讓文字訊息傳送給在遠端佇列管理程式的使用者。它會與「明信片」交互作業，以便傳送到 MQSeries 目標的「明信片」訊息能產生一份被接受的明信片，反之亦然。

「MQSeries Everyplace 明信片」應用程式也有可在 PalmOS 上執行 C 的版本，且能夠與 Java 版本交互作業。

---

### MQSeries Everyplace 探險家

「MQSeries Everyplace 探險家」是一個用 Java 語言撰寫的管理工具，可用來進行本端和遠端佇列管理程式、佇列和訊息的配置和探索。它利用 Microsoft® 基礎類別來呈現標準的 Windows 圖形使用者介面，因此，它無法在非 Windows 的平台上執行。不過，您可以利用它來管理所有 MQSeries Everyplace 佇列管理程式，因為它專門透過管理訊息的傳送和接收來作業。它會呈現兩個窗格的 MQSeries Everyplace 網路檢視畫面；左窗格是物件的樹狀檢視畫面，右窗格是物件明細的清單檢視畫面。

「MQSeries Everyplace 探險家」有下列功能：

- 顯示或修改佇列管理程式內容
- 建立、刪除或修改連線，及顯示其內容
- 建立、刪除或修改佇列，及顯示其內容和/或內涵
- 瀏覽或刪除訊息、顯示其內容，及檢查其欄位
- 傳送測試訊息
- 配置 MQSeries Everyplace 橋接來通往 MQSeries

「MQSeries Everyplace 探險家」通常會利用已配置的佇列管理程式，且可以載入其他類別來執行。如果沒有這些佇列管理程式存在，它會建立一個含有使用者選取特性的佇列管理程式。

在單一部機器中，可以執行工具的多個副本，每個副本都執行於自己的 JVM 中。這項安排容許模擬 MQSeries Everyplace 網路，可用來探究和示範 MQSeries Everyplace 的網路功能和作業。

---

## 第8章 程式設計介面

*MQSeries Everyplace* 系統程式設計介面 (SPI) 是 *MQSeries Everyplace* 的程式設計介面。隨著作業系統的不同，可以採用兩種不同的實作。Java 版可存取所有 *MQSeries Everyplace* 函數；*MQSeries Everyplace* 第 1.0 版的 C 介面只能存取子集。詳細的類別、方法和程序詳載於 *MQSeries Everyplace Programming Reference*；有關 *MQSeries Everyplace* 程式設計的範例，則請參閱 *MQSeries Everyplace Programming Guide*。





---

## 附錄. 注意事項

本資訊是針對 IBM 在美國所提供之產品與服務開發出來的。而在其他國家中，IBM 不見得有提供本書中所提的各項產品、服務、或功能。要知道在您所在之區是否可用到這些產品與服務時，請向當地的 IBM 服務代表查詢。亦不表示該產品、程式或服務為唯一的選擇。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，其他非 IBM 產品、程式、或服務在運作上的評價與驗證，其責任屬於使用者。

在這本書或文件中可能包含著 IBM 所擁有之專利或專利申請案。本書使用者並不享有前述專利之任何授權。您可以用書面方式來查詢授權，來函請寄到：

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

**下列段落若與該國之法律條款抵觸，即視為不適用：** IBM 僅以現狀提供本書，而不提供任何明示或默示之保證 (包括但不限於可售性或符合特定效用的保證)。若有些地區在某些交易上並不允許排除上述保證，則該排除無效。

本資訊中可能會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。同時，IBM 得隨時改進並 (或) 變動本書中所提及的產品及 (或) 程式。

本資訊中任何對非 IBM 網站的敘述僅供參考，IBM 對該網站並不提供保證。該 Web 站上的資料，並非本 IBM 產品所用資料的一部分，因使用該 Web 站造成之損害，由 貴客戶自行負責。

IBM 得以各種適當的方式使用或散布由 貴客戶提供的任何資訊，而無需對您負責。

本程式之獲授權者若希望取得相關資料，以便使用下列資訊者可洽詢 IBM。其下列資訊指的是：(1) 獨立建立的程式與其他程式 (包括此程式) 之間更換資訊的方式 (2) 相互使用已交換之資訊方法若有任何問題請聯絡：

IBM United Kingdom Laboratories,  
Mail Point 151,  
Hursley Park,  
Winchester,  
Hampshire  
England  
SO21 2JN

上述資料之取得有其特殊要件，在某些情況下必須付費方得使用。

IBM 基於雙方之「IBM 客戶合約」、「國際程式授權合約」或任何同等合約之條款，提供本書中所說的授權程式與其所有適用的授權資料。

---

## 商標

下列詞彙是國際商業機器股份有限公司 (IBM) 在美國和 (或) 其他國家的商標。

AIX    AS/400    IBM    MQSeries    OS/390

Microsoft、Windows、Windows NT 和 Windows 標誌圖是 Microsoft Corporation 在美國和 (或) 其他國家的商標。

Java 和所有以 Java 為基礎的商標和標誌圖是 Sun Microsystems, Inc. 在美國和 (或) 其他國家的商標或註冊商標。

其他公司、產品和服務名稱，可能是第三者的商標或服務標記。

---

## 名詞解釋

此名詞解釋說明本書所用的詞彙，以及搭配通用意義來使用的詞彙。在某些情況下，一個詞彙可能有不只一個適用的定義，不過，它會提供這個詞彙在本書內所採用的特定意義。

如果找不到您要尋找的詞彙，請參閱索引或 *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994。

### 三劃

**子類別 (subclass).** 子類別是繼承其他類別的類別。子類別會繼承其超類別的公用和受保護的方法和變數。

### 四劃

**介面 (interface).** 介面也是類別，但只含有抽象方法，不含案例變數。介面提供一組若干不同類別之子類別所能實作的共用方法。

**公用 (public).** 在任何位置中，都能見到公用類別或介面。只要是見到類別之處，也都能見到該類別的公用方法或變數。

**方法 (method).** 方法是一個物件導向程式設計術語，代表函數或程序。

### 五劃

**加密程式 (cryptor).** 這是為訊息加密以提供傳輸安全的程式。

### 六劃

**全球資訊網 (World Wide Web, WWW).** 「全球資訊網」是以一組通信協定為基礎的 Internet 服務，可讓特定配置的伺服器電腦採用標準方式，透過 Internet 來分送文件。

**同步傳訊 (synchronous messaging).** 這是一種各程式將訊息放到訊息佇列中來相互通信的通信方法。在同步傳訊中，傳送端的程式先等待訊息的回答，再繼續它本身的處理程序。請對照**非同步傳訊 (asynchronous messaging)**。

### 七劃

**佇列 (queue).** 佇列是一個 MQSeries 物件。訊息佇列作業應用程式可以將訊息放到佇列中，或取得佇列中的訊息。

**佇列管理程式 (queue manager).** 佇列管理程式是一個系統程式，負責提供訊息佇列作業服務給應用程式。

**伺服器 (server).** (1) MQSeries Everyplace 伺服器是配置了 MQSeries Everyplace 通道管理程式的裝置。(2) MQSeries 伺服器是一個佇列管理程式，負責提供訊息佇列作業服務給遠端工作站所執行的從屬站應用程式。(3) 在更通用的意義上，伺服器是在主從架構特定的兩個程式資訊串流模型中，負責回應資訊要求的程式。(4) 執行伺服器程式的電腦。

### 八劃

**受保護的 (protected).** 只有在類別本身之內，或該類別的子類別內，或該類別所屬套件內，才能見到該類別內受保護的欄位。

**物件 (object).** (1) 在 Java 中，物件是類別中的案例。類別是事物群組的模型；物件是該群組之特定成員的模型。(2) 在 MQSeries 中，物件是一個佇列管理程式、佇列或通道。

**非同步傳訊 (asynchronous messaging).** 這是一種各程式將訊息放到訊息佇列中來相互通信的通信方法。在非同步的傳訊中，傳送端的程式不需要等待訊息的回答，可以繼續它本身的處理程序。請對照**同步傳訊 (synchronous messaging)**。

## 九劃

**封裝 (encapsulation).** 封裝是一種物件導向程式設計技術，使物件資料成為專用或受保護的資料，讓程式設計師只能透過方法呼叫來存取和操作資料。

## 十劃

**個人數位助理 (personal digital assistant, PDA).** 這是一種口袋大小的個人電腦。

**套件 (package).** Java 中的套件是讓一組特定類別存取某 Java 程式碼片段的方法。在特定套件內的 Java 程式碼可以存取套件內的所有類別，以及各類別內所有非專用的方法或欄位。

**案例 (instance).** 案例是一個物件。當類別案例化以產生物件時，我們說，這個物件是類別的一個案例。

**訊息 (message).** 在訊息佇列作業應用程式中，訊息是在程式之間傳送的通信。

**訊息佇列 (message queue).** 請參考「佇列 (queue)」。

**訊息佇列作業 (message queuing).** 這是一種程式設計技術，在這裡，應用程式內的每個程式都是將訊息放到佇列中，來與其他程式通信。

## 十一劃

**動態通道 (dynamic channel).** 動態通道會連接 MQSeries Everyplace 裝置，並以雙向方式來轉送同步和非同步訊息與回應。

**專用 (private).** 在類別之外，無法見到類別內的專用欄位。

**從屬站 (client).** 在 MQSeries 中，從屬站是一個執行時期元件，用來為本端使用者應用程式存取伺服器中的佇列作業服務。

**通道 (channel).** 請參考**動態通道 (dynamic channel)** 和 **MQI 通道 (MQI channel)**。

**通道管理程式 (channel manager).** 這是一種 MQSeries Everyplace 物件，支援端點之間的邏輯多重並行通信管線。

## 十二劃

**超文字標記語言 (Hypertext Markup Language, HTML).** 這是一種用來定義要呈現於全球資訊網之資訊的語言。

**超類別 (superclass).** 超類別是其他類別所繼承的類別。子類別可以使用超類別的公用和受保護的方法與變數。

## 十三劃

**傳輸控制通信協定/Internet 通信協定 (Transmission Control Protocol/Internet Protocol, TCP/IP).** 這是一組支援本端和廣域網路之同層級對同層級連線功能的通信協定。

**閘道 (gateway).** MQSeries Everyplace 閘道（或伺服器）是一部電腦，負責執行包括通道管理程式在內的 MQSeries Everyplace 程式碼。

## 十四劃

**輕裝備目錄存取通信協定 (Lightweight Directory Access Protocol, LDAP).** LDAP 是一種用來存取目錄服務的主從架構通信協定。

## 十六劃

**橋接 (bridge).** 這是一種 MQSeries Everyplace 物件，可讓訊息在 MQSeries Everyplace 和包括 MQSeries 在內的其他傳訊系統之間流動。

## 十七劃

**壓縮程式 (compressor).** 這是壓縮訊息以縮小要傳輸之資料量的程式。

**應用程式設計介面 (Application Programming Interface, API).** 「應用程式設計介面」由若干函數和變數組成，程式設計師可以在他們的應用程式中使用這些函數和變數。

## 十九劃

**類別 (class).** 類別是一種資料和操作資料之方法的封裝集成。類別可以案例化，以產生類別案例物件。

## 二十二劃

**鑑別程式 (authenticator).** 這是一種檢查和驗證訊息之傳送者和接收者的程式。

## I

**Internet.** Internet 是共用資訊的協力公用網路。在實體上，Internet 會利用目前現存的所有公用電信網路之總資源的子集。在技術上，將 Internet 區分為協力公用網路的因素，在於它使用一組稱為 TCP/IP（傳輸控制通信協定/Internet 通信協定）的通信協定。

## J

**Java Developers Kit (JDK).** 這是 Sun Microsystems 分送給 Java 開發人員的一個軟體套件。它包含 Java 直譯器、Java 類別和各種 Java 開發工具：編譯器、除錯器、反組合格式、appletviewer、Stub 檔產生器，以及文件產生器。

**Java 命名和目錄服務 (Java Naming and Directory Service, JNDI).** 這是 Java 程式設計語言所指定的 API。它提供命名和目錄功能給用 Java 程式設計語言來撰寫的應用程式。

## M

**MQI 通道 (MQI channel).** MQI 通道會將 MQSeries 從屬站連接到伺服器系統中的佇列管理程式，並採用雙向方式來轉送 MQI 呼叫和回應。

## S

**servlet.** 這是一種設計來專門執行於 Web 伺服器的 Java 程式。

## W

**Web.** 請參考「全球資訊網 (World Wide Web)」。

**Web 瀏覽器 (Web browser).** 這是一種會建立全球資訊網所分送之資訊的格式並加以顯示的程式。

## 特殊字元

**MQSeries.** MQSeries 是一系列提供訊息佇列作業服務的 IBM 授權程式。



---

## 參考書目

相關出版品：

- *MQSeries Everyplace 請先閱讀我*, GC40-0542-00
- *MQSeries Everyplace Programming Reference*, SC34-5846-00
- *MQSeries Everyplace Programming Guide*, SC34-5845-00
- *MQSeries An Introduction to Messaging and Queuing*, GC33-0805-01
- *MQSeries for Windows NT V5R1 快速入門*, GC40-0314-00





# 索引

索引順序以中文字，英文字，及特殊符號之次序排列。

## 〔三劃〕

工作站傳訊 6

## 〔四劃〕

介面, 安全 39  
介面, 程式設計 59  
公用登錄 38  
公用程式 57  
公用程式, MQSeries Everyplace 探險家 57  
分散式傳訊 5  
支援的作業系統 3

## 〔五劃〕

主從架構連線 42  
主電腦傳訊 6  
以 MQSeries Everyplace 管理 25  
加密 32  
功能 11  
可調整性 30  
可鑑別實體 37  
必要的作業系統 3  
必備知識 ix  
必備條件 3  
本書適用對象 ix  
本端安全 34  
本端佇列 18  
本端, 安全 34

## 〔六劃〕

同步傳訊 22, 32  
同層級對同層級連線 41  
多重連線樣式 42  
安全介面 39

安全, MQSeries Everyplace 32  
安全, 佇列型 34  
安全, 訊息層次 35  
自行設定 39  
自動登錄 37

## 〔七劃〕

佇列型安全 34  
佇列規則 40  
佇列管理程式 6, 41  
佇列管理程式 Proxy 物件 48  
佇列管理程式作業 23  
佇列管理程式規則 40  
佇列管理程式, MQSeries Everyplace 21  
佇列, MQSeries Everyplace 18  
佇列, MQSeries 橋接 19  
佇列, 本端 18  
佇列, 遠端 18  
佇列, 儲存和轉遞 19  
伺服器, MQSeries 6  
作業系統, 支援 3  
作業, 佇列管理程式 23

## 〔八劃〕

事件日誌 29  
明信片 應用程式 57  
注意事項, 法務 61  
法務注意事項 61  
物件, MQSeries Everyplace 14  
物件, 訊息 14  
非同步傳訊 21, 31

## 〔九劃〕

客戶需求 12

## 〔十劃〕

訊息物件 14  
訊息遞送, 確定 55

訊息層次安全 35  
訊息轉換 52  
訊息, 管理 25  
起始伺服器佇列 19  
起始伺服器, MQSeries Everyplace 19  
迷你憑證 37  
迷你憑證發行服務 38  
追蹤 MQSeries Everyplace 29  
配接器, MQSeries Everyplace 29  
配置 39  
配置, MQSeries Everyplace 30

## 〔十一劃〕

動態通道 7, 13, 27  
商標 62  
專用登錄 37  
從屬站通道 7, 13  
從屬站, MQSeries 6  
接聽器物件 50, 51  
產品概念 13  
規則, MQSeries Everyplace 39  
通用傳訊 6  
通信 41  
通道接聽器 41  
通道管理程式 41  
通道, 動態 7, 13, 27  
通道, 從屬站 7, 13  
連線樣式 41  
連線樣式, 多重 42  
連線, 主從架構 42  
連線, 同層級對同層級 41

## 〔十二劃〕

登錄 36  
登錄, MQSeries Everyplace 13  
登錄, 公用 38  
登錄, 專用 37  
程式設計介面 59  
詞彙 ix

## 〔十三劃〕

傳訊, MQSeries 5  
傳訊, 同步 22, 32  
傳訊, 非同步 21, 31  
傾出資料格式 17  
概念, 產品 13  
裝置, MQSeries Everyplace 7, 13  
載入應用程式 43  
閘道, MQSeries Everyplace 7, 13

## 〔十四劃〕

實體, 可鑑別 37  
監視 27  
管理公用程式 57  
管理訊息 25  
網路, MQS 45  
網路, MQSeries Everyplace 30, 45  
與 MQSeries 的相容性 54  
說明 1  
遠端佇列 18  
需求, 客戶 12

## 〔十五劃〕

撥號連線管理 29  
確定訊息遞送 55

## 〔十六劃〕

憑證複製 38  
橋接物件 46, 47  
橋接, MQS 45

## 〔十七劃〕

儲存和轉遞佇列 19  
壓縮 32  
應用程式 57  
應用程式, MQSeries Everyplace 11  
應用程式, 明信片 57  
應用程式, 載入 43  
總覽 1

## 〔十八劃〕

轉換程式 52

## 〔十九劃〕

關於本書 ix  
類別, MQSeries Everyplace 42

## 〔二十一劃〕

屬性規則 39

## 〔二十二劃〕

讀者 ix

## M

MQAttribute 36  
MQMTrustAttribute 36  
MQSeries Everyplace 公用程式 57  
MQSeries Everyplace 安全 32  
MQSeries Everyplace 佇列 18  
MQSeries Everyplace 佇列管理程式  
21  
MQSeries Everyplace 物件 14  
MQSeries Everyplace 配接器 29  
MQSeries Everyplace 配置 30  
MQSeries Everyplace 探險家 公用程  
式 57  
MQSeries Everyplace 規則 39  
MQSeries Everyplace 登錄 13, 36  
MQSeries Everyplace 裝置 7, 13  
MQSeries Everyplace 閘道 7, 13  
MQSeries Everyplace 管理 25  
MQSeries Everyplace 管理公用程式  
57  
MQSeries Everyplace 網路 30, 45  
MQSeries Everyplace 應用程式 11,  
57  
MQSeries Everyplace 類別 42  
MQSeries Integrator 5  
MQSeries Workflow 5  
MQSeries 介面 45  
MQSeries 伺服器 6  
MQSeries 系列 5  
MQSeries 從屬站 6  
MQSeries 傳訊 5  
MQSeries 網路 45  
MQSeries 橋接 45

MQSeries 橋接佇列 19  
MQSeries 橋接規則 39  
MQSeries, 介面 45  
MQSeries, 相容性 54

## R

RAS 撥號程式規則 40

## S

SPI 59



折疊線

台北市敦化南路一段二號十二樓

臺灣國際商業機器股份有限公司  
中文支援中心 啟

廣告回信	號
臺灣北區郵政管理局 登記	號
北台字第	

免貼郵票

寄件人 姓名：  
地址：

寄

折疊線

讀者意見表





Printed in Singapore

GC40-0541-00

