

7.5

IBM WebSphere MQ 概觀

IBM

附註

使用本資訊及其支援的產品之前，請先閱讀第 191 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® WebSphere MQ 7.5 版及所有後續版次與修訂。

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, 2024.

目錄

關於 IBM WebSphere MQ	5
IBM WebSphere MQ 簡介.....	5
IBM WebSphere MQ 授權資訊.....	8
IBM WebSphere MQ Telemetry 簡介.....	10
IBM WebSphere MQ Version 7.5 資訊導覽圖.....	13
IBM Documentation 離線 應用程式中的 IBM WebSphere MQ Version 7.5.....	15
IBM WebSphere MQ 7.5 PDF 文件.....	15
IBM WebSphere MQ 7.5 中的新增功能.....	16
IBM WebSphere MQ Client for HP Integrity NonStop 伺服器.....	18
IBM WebSphere MQ 7.5 中的變更內容.....	21
IBM WebSphere MQ Version 7.5 修正套件中的變更內容.....	21
舊版中的新增功能及變更內容.....	29
舊版 IBM WebSphere MQ 出版品與現行資訊結構之間的對映.....	29
AIX 的快速入門.....	29
HP-UX 的快速入門.....	29
Linux 的快速入門.....	30
Solaris 快速入門.....	30
Windows 的快速入門.....	30
應用程式設計手冊.....	31
應用程式設計參考手冊.....	31
用戶端.....	31
常數.....	32
交互通訊.....	32
訊息與訊息碼.....	33
移轉.....	33
監視.....	33
可程式指令格式及管理介面.....	33
發佈/訂閱使用手冊.....	33
佇列管理程式叢集.....	33
Script (MQSC) 指令參考手冊.....	34
安全.....	34
系統管理手冊.....	34
使用 .NET.....	34
使用 C++.....	35
使用 Java.....	35
Web 服務.....	35
使用元件物件模型介面.....	35
IBM WebSphere MQ 7.5、 IBM i 及 z/OS.....	35
注意事項.....	35
技術概觀.....	37
訊息佇列作業簡介.....	37
交互通訊概念.....	44
IBM WebSphere MQ Telemetry.....	59
管理物件.....	89
IBM WebSphere MQ 多重播送.....	108
安全.....	109
用戶端及伺服器.....	110
交易管理及支援.....	115
延伸佇列管理程式機能.....	116
IBM WebSphere MQ Client for HP Integrity NonStop Server 技術概觀.....	117
案例.....	118
IBM WebSphere MQ 7.5 入門.....	118

使用 Script 的基本檔案傳送.....	124
詳細的基本檔案傳送.....	132
使用 Script 進行兩台電腦檔案傳送.....	137
詳細傳送兩部電腦檔案.....	144
將審核功能新增至受管理檔案傳送.....	151
名詞解釋.....	153
A.....	153
B.....	156
C.....	157
D.....	161
E.....	163
F.....	163
G.....	165
H.....	165
I.....	166
J.....	167
K.....	168
L.....	168
M.....	170
N.....	173
O.....	174
P.....	175
Q.....	177
R.....	178
S.....	181
T.....	185
U.....	187
V.....	188
W.....	188
X.....	189
協助工具.....	189
Windows 上的協助工具.....	190
注意事項.....	191
程式設計介面資訊.....	192
商標.....	192

關於 IBM WebSphere MQ

本節提供介紹資訊，以協助您開始使用 IBM WebSphere MQ:

相關工作

[設計 WebSphere MQ 架構](#)

IBM WebSphere MQ

歡迎使用 IBM WebSphere MQ Version 7.5 產品說明文件，您可以在其中找到如何完成建立及維護 MQ 環境所需執行之作業的詳細指示。本說明文件也包含概念性資訊，協助您瞭解本產品以及如何使用本產品來解決商業問題。



IBM WebSphere MQ 是一套強大、安全又可靠的傳訊中介軟體。它使用訊息和佇列來支援在應用程式、系統、服務和檔案之間交換訊息。它可以跨多個平台來簡化並加速整合不同的應用程式和商業資料。

相關概念

[關於 IBM WebSphere MQ](#)

相關工作

[規劃](#)

[移轉及升級](#)

[安裝](#)

[安全](#)

[配置](#)

[管理](#)

[開發應用程式](#)

[監視及效能](#)

[疑難排解和支援](#)

[WebSphere MQ Managed File Transfer](#)

[WebSphere MQ 探險家](#)

[MQ GUI 精靈說明](#)

[WebSphere MQ 網際網路透通](#)

[Message Service Client for .NET](#)

[WebSphere MQ Hypervisor Edition](#)

[Mobile Messaging and M2M](#)

相關參考

[參照](#)

相關資訊

[IBM MQ on Cloud](#)

IBM WebSphere MQ 簡介

您可以使用 IBM WebSphere MQ，讓應用程式在不同時間及許多不同的運算環境中進行通訊。

什麼是 IBM WebSphere MQ?

- IBM WebSphere MQ 是應用程式的傳訊。它透過不同元件的網路傳送訊息。您的應用程式會連接至 IBM WebSphere MQ，以傳送或接收訊息。IBM WebSphere MQ 會處理它在傳送訊息時所遇到的不同處理器、作業系統、子系統及通訊協定。如果連線或處理器暫時無法使用，IBM WebSphere MQ 會將訊息排入佇列，並在連線回到線上時轉遞訊息。
- 應用程式可以選擇程式設計介面，以及連接至 IBM WebSphere MQ 的程式設計語言。
- IBM WebSphere MQ 是傳訊及佇列作業中介軟體，具有點對點、發佈/訂閱及檔案傳送作業模式。應用程式可以透過多重播送將訊息發佈給許多訂閱者。

傳訊

程式通訊的方式是在訊息中傳送彼此的資料，而不是直接互相呼叫。

佇列作業

訊息會放置在佇列上，因此程式可以在不同速度和時間、不同位置且彼此之間沒有直接連線的情況下，彼此獨立執行。

點對點 (*point-to-point*)

應用程式會將訊息傳送至佇列或佇列清單。傳送端必須知道目的地的名稱，但不知道它所在的位置。

發佈/訂閱 (*Publish/Subscribe*)

應用程式會發佈主題的訊息，例如團隊所玩的遊戲結果。IBM WebSphere MQ 會將訊息副本傳送至訂閱結果主題的應用程式。他們收到的訊息是球隊的比賽結果。發佈者不知道訂閱者的名稱或它們在何處。

多重播送

多重播送是一種有效率的發佈/訂閱傳訊形式，可擴充至許多訂閱者。它會將將發佈副本從 IBM WebSphere MQ 傳送至每一個訂閱者的工作傳送至網路。一旦在發佈者與訂閱者之間建立發佈的路徑，IBM WebSphere MQ 就不會涉及轉遞發佈。

檔案傳輸

在訊息中傳送檔案。IBM WebSphere MQ 檔案傳送版本可管理檔案傳送及管理，以設定自動化傳送並記載結果。您可以整合檔案傳送與其他檔案傳送系統、IBM WebSphere MQ 傳訊及 Web。

遙測

IBM WebSphere MQ Telemetry 是裝置的傳訊。IBM WebSphere MQ 會將裝置與應用程式傳訊連接在一起。它將網際網路、應用程式、服務及決策者與已檢測裝置的網路連接在一起。IBM WebSphere MQ 遙測具有有效率的傳訊通訊協定，可透過網路連接大量裝置。傳訊通訊協定已發佈，因此可以將它納入裝置中。您也可以使用通訊協定的其中一個已發佈程式設計介面來開發裝置程式。

它能為我做什麼？

- IBM WebSphere MQ 會透過網路在應用程式之間傳送及接收資料。
- 訊息遞送確定且與應用程式取消連結。請放心，因為 IBM WebSphere MQ 會以交易方式交換訊息，並取消連結，因為應用程式不需要檢查它們所傳送的訊息是否安全遞送。
- 您可以使用 SSL/TLS 來保護佇列管理程式之間的訊息遞送安全。
- 使用 Advanced Message Security (AMS)，您可以在由一個應用程式放置及由另一個應用程式擷取之間加密及簽署訊息。
- 應用程式設計師不需要具備通訊程式設計知識。

如何使用它？

- 使用 IBM WebSphere MQ Explorer GUI 或從指令視窗或應用程式執行指令來建立及管理 IBM WebSphere MQ。
- 透過呼叫其中一個程式設計介面來傳送及接收訊息的程式應用程式。程式設計介面是針對不同語言所提供，並且包括標準 JMS 程式設計介面，以及 Windows 通訊基礎的類別。
- 使用 HTTP 通訊協定從瀏覽器傳送及接收 IBM WebSphere MQ 訊息。

如何運作？

- 管理者使用指令來建立及啟動佇列管理程式。隨後，通常會在作業系統啟動時自動啟動佇列管理程式。然後，應用程式及其他佇列管理程式可以連接至該應用程式，以傳送及接收訊息。
- 應用程式或管理者建立佇列或主題。佇列及主題是佇列管理程式所擁有及儲存的物件。
- 當您應用程式要將資料轉移至另一個應用程式時，它會將資料放入訊息。它將訊息放入佇列，或將訊息發佈至主題。有三種主要方法可以擷取訊息：
 - 連接至相同佇列管理程式的點對點應用程式會從相同佇列擷取訊息。

例如，應用程式將訊息放置在佇列上作為儲存暫時或持續資料的方式。第二個範例：與在不同程序中執行的另一個應用程式共用資料的應用程式。
 - 連接至另一個佇列管理程式的點對點應用程式會從不同的佇列擷取相同的訊息。

應用程式透過交換佇列上的訊息來彼此通訊。IBM WebSphere MQ 的主要用途是傳送或交換訊息。一個應用程式將訊息放置在一部電腦上的佇列中，另一個應用程式則從另一部電腦上的另一個佇列中取得相同的訊息。兩台電腦上的佇列管理程式會一起運作，將訊息從第一個佇列傳送至第二個佇列。應用程式不會彼此通訊，而佇列管理程式會彼此通訊。
 - 連接至任何佇列管理程式的訂閱者應用程式會擷取一般主題的訊息。

發佈者應用程式會建立訊息，並將它發佈至一部電腦上的主題。任何數目的訂閱者應用程式都會訂閱不同電腦上的相同主題。IBM WebSphere MQ 會將發佈遞送至屬於訂閱者所連接之佇列管理程式的佇列。訂閱者會從佇列中擷取訊息。
- MQ 通道 會透過網路將一個佇列管理程式連接至另一個佇列管理程式。您可以自行建立 MQ 通道，或佇列管理程式叢集中的佇列管理程式會在需要時建立 MQ 通道。
- 您可以在一個佇列管理程式上具有許多佇列及主題。
- 一部電腦上可以有許多佇列管理程式。
- 應用程式可以在與佇列管理程式相同的電腦上執行，也可以在不同的電腦上執行。如果它在相同電腦上執行，則它是 IBM WebSphere MQ 伺服器應用程式。如果它在不同的電腦上執行，則它是 IBM WebSphere MQ 用戶端應用程式。不論它是 IBM WebSphere MQ 用戶端或伺服器，幾乎不會對應用程式造成任何影響。您可以使用 IBM WebSphere MQ 用戶端或伺服器來建置主從式應用程式。

IBM WebSphere MQ 隨附哪些工具和資源？

- 控制指令，從指令行執行。您可以使用控制指令來建立、啟動及停止佇列管理程式。您也可以使用控制指令來執行 IBM WebSphere MQ 管理及問題判斷程式。
- IBM WebSphere MQ Script 指令 (MQSC)，由直譯器執行。使用指令建立佇列及主題、配置及管理 IBM WebSphere MQ。編輯檔案中的指令，並將檔案傳遞至 `runmqsc` 程式以解譯它們。您可以在一個佇列管理程式上執行直譯器，它會將指令傳送至不同的電腦，以管理不同的佇列管理程式。
- 「可程式指令格式 (PCF)」指令，您可以在自己的應用程式中呼叫這些指令來管理 IBM WebSphere MQ。PCF 指令具有與 Script 指令相同的功能，但它們更容易程式設計。
- 程式範例。
- 在 Windows 和 Linux® x86 及 x86-64 平台上，您可以在其中執行下列公用程式：
 - IBM WebSphere MQ Explorer。瀏覽器執行的管理作業與 Script 指令相同，但更容易以互動方式使用。
 - *Postcard* 應用程式，用於示範傳訊並驗證您的安裝。
 - 指導教學。

相關概念

第 16 頁的『IBM WebSphere MQ Version 7.5 的新增功能』
瞭解 IBM WebSphere MQ Version 7.5 中的主要新功能。

WebSphere MQ 多重播送

IBM WebSphere MQ 多重播送提供低延遲、高扇出、可靠的多重播送傳訊功能。

WebSphere MQ Telemetry

人們、企業和政府越來越希望使用 IBM WebSphere MQ Telemetry 與我們生活和工作的環境進行更聰明的互動。IBM WebSphere MQ Telemetry 會將所有類型的裝置連接至網際網路及企業，並減少為智慧型裝置建置應用程式的成本。

[傳訊及佇列的技術簡介](#)

WebSphere MQ 產品可讓程式使用一致的應用程式設計介面，透過不同元件 (處理器、作業系統、子系統及通訊協定) 的網路彼此通訊。

[WebSphere MQ 用戶端和伺服器的技術簡介](#)

簡介 IBM WebSphere MQ 如何支援其應用程式的主從架構配置。

[佇列管理程式通訊的技術簡介](#)

在 WebSphere MQ 中，交互通訊表示將訊息從一個佇列管理程式傳送至另一個佇列管理程式。接收端佇列管理程式可以位於相同或另一部機器上; 附近或世界另一端。它可以在與本端佇列管理程式相同的平台上執行，也可以在 WebSphere MQ 支援的任何平台上執行。這稱為分散式環境。WebSphere MQ 會使用「分散式佇列管理 (DQM)」來處理分散式環境中的通訊。

相關工作

[WebSphere MQ Advanced 訊息安全 \(AMS\)](#)

[WebSphere MQ Managed File Transfer](#)

IBM WebSphere MQ 授權資訊

您可以使用 IBM WebSphere MQ 購買的項目，以及每次購買的項目授權您安裝。

注意事項: 本授權手冊提供增補資訊，以協助您在所購買的授權內部署從 IBM 取得授權的程式。您的授權合約 (例如 IBM 國際程式授權合約 (IPLA) 及其交易文件，包括 IBM WebSphere MQ 產品供應項目的授權資訊) 是您與 IBM 之間關於使用「本程式」的唯一完整合約。

您可以使用 IBM WebSphere MQ 購買的內容

分散式平台

若為分散式平台上的 IBM WebSphere MQ，產品與服務包含 11 個可獨立購買的元件：

5724-H72 IBM IBM WebSphere MQ

IBM IBM WebSphere MQ (伺服器)

IBM IBM WebSphere MQ 遙測

IBM IBM WebSphere MQ Advanced Message Security

IBM IBM WebSphere MQ 閒置待命

IBM IBM WebSphere MQ Advanced Message Security 閒置待命

IBM IBM WebSphere MQ 進階

IBM IBM WebSphere MQ 進階閒置待命

IBM IBM WebSphere MQ Advanced for Developers

IBM IBM WebSphere MQ Managed File Transfer 服務

IBM IBM WebSphere MQ Managed File Transfer 服務閒置待命

IBM IBM WebSphere MQ Managed File Transfer 受管理端點

我的企業有權安裝什麼？

對於分散式平台上的 IBM WebSphere MQ，下面的元件直接對映至 IBM WebSphere MQ 安裝程式可以安裝的元件，因此對於這些元件，您已購買的項目與您可以安裝的項目之間的對映很容易。

重要: IBM WebSphere MQ 安裝媒體包含所有元件，但您應該只安裝已購買授權的子集。

5724-H72 IBM IBM WebSphere MQ

IBM WebSphere (伺服器)

包括：

- ClientDevelopment 套件 (SDK)
- IBM 廣域安全套件 (UNIX)
- IBM WebSphere MQ 瀏覽者
- Java .NET 傳訊和 Web 服務
- 範例程式
- 伺服器/執行時期
- UNIX 線上指令說明

IBM IBM WebSphere MQ 遙測

包括：

- 遙測服務

IBM IBM WebSphere MQ Advanced Message Security

包括：

- Advanced Message Security

IBM IBM WebSphere MQ Managed File Transfer 服務

包括：

- IBM WebSphere MQ Managed File Transfer 日誌程式
- IBM WebSphere MQ Managed File Transfer 服務
- IBM WebSphere MQ Managed File Transfer 工具

IBM WebSphere IBM WebSphere MQ Managed File Transfer 受管理端點

包括：

- IBM WebSphere MQ Managed File Transfer 代理程式
- IBM WebSphere MQ Managed File Transfer 日誌程式
- IBM WebSphere MQ Managed File Transfer 工具

何謂 IBM IBM WebSphere MQ Advanced?

已引進 IBM WebSphere MQ Advanced，以簡化採購授權的處理程序。您的企業支付一個價格，並取得多個 IBM WebSphere MQ 元件的授權。

IBM WebSphere MQ 進階組件如下：

5724-H72 IBM IBM WebSphere MQ

IBM IBM WebSphere MQ 進階

IBM IBM WebSphere MQ Advanced for Developers

若為分散式平台上的 IBM IBM WebSphere MQ，購買 100 個處理器價值單位 (PVU) 的 IBM WebSphere MQ Advanced，即會提供您企業安裝的授權：

- 100 個 PVU 的 IBM IBM WebSphere MQ (伺服器)、及
- 100 個 PVU 的 IBM IBM WebSphere MQ Advanced Message Security 及
- 100 個 PVU 的 IBM IBM WebSphere MQ Managed File Transfer Service、及
- 無限制安裝 IBM IBM WebSphere MQ Telemetry

此外，您的企業可以視需要混合並符合 IBM 版本。因此，您的 100 個 PVU IBM IBM WebSphere MQ (伺服器) 授權可以分割成 50 個 PVU (IBM WebSphere MQ 7.1 版) 及 50 個 PVU (IBM WebSphere MQ 7.5 版) (此元件)。

IBM IBM WebSphere MQ Advanced for Developers 提供 IBM IBM WebSphere MQ Advanced 及 IBM IBM WebSphere MQ Managed File Transfer Managed Endpoint 的一切授權，僅用於開發目的。



小心: IBM 授權定義視為開發目的的項目。

何謂閒置待命組件?

已引進「閒置待命」組件來滿足高可用性環境，其中被動系統已 IBM WebSphere MQ 安裝且可用，但該系統未執行任何 IBM WebSphere MQ 處理工作或活動，但與作用中佇列管理程式的配置及活動保持最新。在此情況下，可能適用較低的費用。

附註:

1. 使用 IBM WebSphere MQ 多重實例佇列管理程式特性也需要「閒置待命」授權。
2. IBM WebSphere MQ Telemetry 元件沒有閒置待命組件。除非您有 IBM WebSphere MQ 「進階閒置待命」(在此情況下會包含它)，否則必須為主動及被動系統購買相同的 IBM WebSphere MQ Telemetry 組件。
3. IBM WebSphere 受管理檔案傳送受管理端點也沒有閒置待命組件，因為端點不是伺服器環境的一部分。

相關概念

第 21 頁的『IBM WebSphere MQ Version 7.5 中的變更內容』

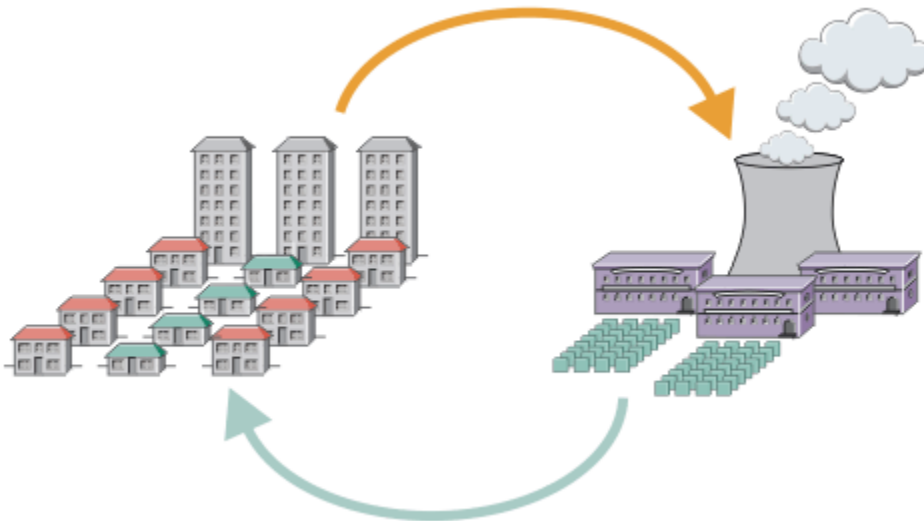
第 37 頁的『IBM WebSphere MQ 技術概觀』

使用 IBM WebSphere MQ 來連接您的應用程式，並管理跨組織的資訊配送。

簡介 IBM WebSphere MQ Telemetry

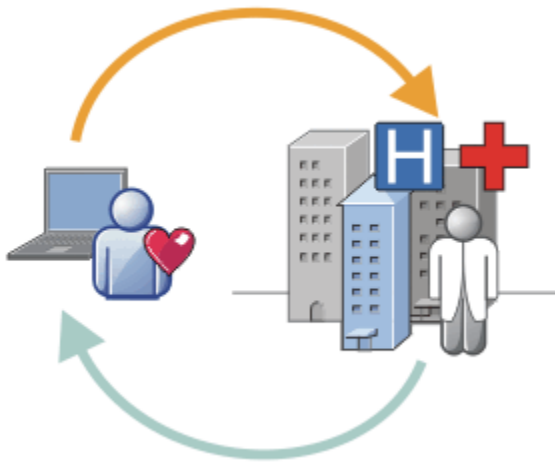
人們、企業和政府越來越希望使用 IBM WebSphere MQ Telemetry 與我們生活和工作的環境進行更聰明的互動。IBM WebSphere MQ Telemetry 會將所有類型的裝置連接至網際網路及企業，並減少為智慧型裝置建置應用程式的成本。

下列圖表示範 IBM WebSphere MQ Telemetry 的一些一般用法:



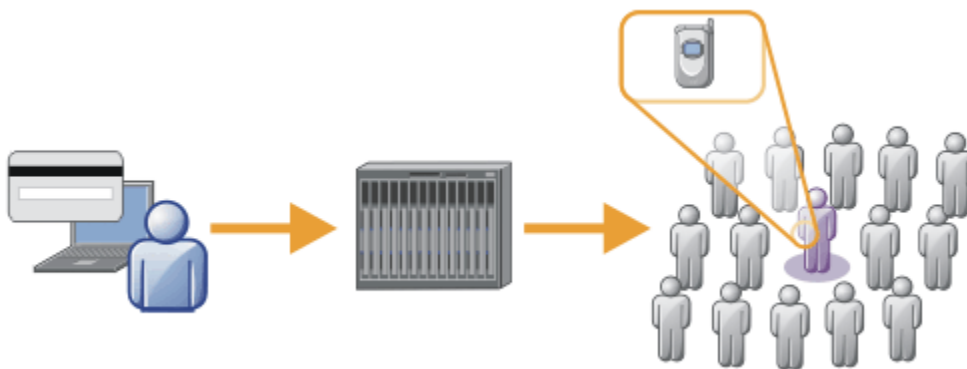
- MQTT 訊息，包含傳送至服務提供者的電力使用資料。
- 遙測應用程式傳送基於電力使用資料分析的控制指令。
- 如需相關資訊，請參閱第 63 頁的『遙測實務範例：家庭能源監視及控制』。

圖 1: 智慧型電力計量



- 遙測應用程式將您的健康資料傳送至醫院及醫生。
- 根據性能資料的分析，會傳回 MQTT 訊息警示或意見。
- 如需相關資訊，請參閱第 62 頁的『遙測實務範例：家庭病患監視』。

圖 2: 智慧型健康監視



- 簡式卡片交易會傳送至銀行的伺服器。
- IBM WebSphere MQ Telemetry 可識別數千中的一個人員，並警示客戶已使用其卡。
- IBM WebSphere MQ Telemetry 可以使用最簡單的資訊輸入，並尋找該個人。

圖 3: 遙測: 一個人在人群中

何謂 WebSphere MQ Telemetry?

- 它是 IBM WebSphere MQ 的一項特性，可將 IBM WebSphere MQ 所提供的通用傳訊骨幹延伸至各種遠端感應器、掣動器及遙測裝置。IBM WebSphere MQ Telemetry 延伸 IBM WebSphere MQ，以便它可以將智慧型企業應用程式、服務及決策者與已檢測裝置的網路交互連接。
- WebSphere MQ Telemetry 的兩個核心部分如下：
 1. 在 IBM WebSphere MQ 伺服器內執行的 IBM WebSphere MQ Telemetry 服務。
 2. 與應用程式一起配送至裝置的 IBM WebSphere MQ Telemetry 用戶端。

它能為我做什麼?

- MQ Telemetry 使用 MQ Telemetry Transport (MQTT)，在應用程式與 IBM WebSphere MQ 佇列管理程式之間傳送及接收資料。
- MQTT 是一種開放式傳訊傳輸，容許針對各種裝置建立 MQTT 實作。

- MQTT 用戶端可以在資源有限的小型覆蓋區裝置上執行。
- MQTT 在頻寬可能較低、傳送資料成本昂貴或可能脆弱的網路上有效運作。
- 訊息遞送已確定並與應用程式取消連結。
- 應用程式設計師不需要具備通訊程式設計知識。
- 訊息可以與其他傳訊應用程式交換。這些可能是其他遙測應用程式、MQI、JMS 或企業傳訊應用程式。

如何使用它？

- 使用 IBM WebSphere MQ Explorer 及其相關聯工具，可管理 MQ 的 WebSphere MQ Telemetry 特性。
- 使用應用程式中的 MQTT 用戶端來連接至佇列管理程式，發佈及訂閱訊息。
- 使用 MQTT 用戶端將應用程式配送至要執行應用程式的裝置。

如何運作？

- MQ Telemetry (MQXR) 服務將 IBM WebSphere MQ 佇列管理程式轉換為 MQTT 伺服器
- MQTT 伺服器瞭解 MQTT 訊息傳輸，並可接收來自 MQTT 用戶端的訊息，以及將訊息傳送至 MQTT 用戶端。
- MQ Telemetry 隨附許多實作 MQTT 訊息傳輸的遙測用戶端。這些通常稱為 MQTT 用戶端。
- 基本 Telemetry 用戶端的運作方式類似於標準 MQ 用戶端，但可以在更廣泛的各種平台及網路上執行。
- Advanced Telemetry Client 充當網路集中器，可將更多 MQTT 用戶端連接至單一佇列管理程式。它也可以為缺少在短暫網路中斷期間緩衝訊息的方法的小型裝置提供儲存及轉遞。
- IBM WebSphere MQ Telemetry daemon for devices 是屬於 IBM WebSphere MQ Telemetry 的 Advanced Telemetry 用戶端。如需相關資訊，請參閱第 84 頁的『用於裝置的 Telemetry 常駐程式』。
- MQTT 是發佈訂閱通訊協定：
 - MQTT 用戶端應用程式可以將訊息發佈至 MQTT 伺服器。
 - 當 IBM WebSphere MQ 佇列管理程式充當 MQTT 伺服器時，連接至佇列管理程式的其他應用程式可以訂閱及接收來自 MQTT 用戶端的訊息。
 - MQTT 用戶端可以訂閱連接至 MQ 佇列管理程式的應用程式所傳送的訊息。
 - 佇列管理程式充當路由器將訊息從發佈應用程式配送至訂閱應用程式。
 - 訊息可以在不同類型的用戶端應用程式之間配送。例如，在 Telemetry 用戶端與 JMS 用戶端之間。

IBM WebSphere MQ Telemetry 取代在 WebSphere Message Broker 第 7 版中撤銷並在 Windows、Linux 及 AIX 上執行的 SCADA 節點。將遙測應用程式從使用 WebSphere Message Broker 第 6 版移轉至使用 IBM WebSphere MQ Telemetry 及 WebSphere Message Broker 7.0 版提供資訊，可協助您從使用 WebSphere Message Broker V6 中的 SCADA 節點移轉應用程式。使用 WebSphere Message Broker 第 7 版的遙測應用程式會訂閱 MQTT 用戶端共用的主題。它們會使用 MQInput 節點從 MQTT 用戶端接收發佈，並使用發佈節點發佈至 MQTT 用戶端。

相關概念

[第 61 頁的『遙測概念及監視與控制實務範例』](#)

遙測可用於自動感應、測量資料及控制遠端裝置。主要用於將資料從裝置傳輸至中央控制點。遙測還包括將配置及控制資訊傳送至裝置。

相關工作

[安裝 WebSphere MQ Telemetry](#)

[管理 WebSphere MQ Telemetry](#)

[從使用 WebSphere Message Broker 第 6 版移轉遙測應用程式，以使用 WebSphere MQ Telemetry 及 WebSphere Message Broker 7.0](#)

[開發 WebSphere MQ Telemetry 應用程式](#)

[WebSphere MQ Telemetry 疑難排解](#)

相關參考

[WebSphere MQ Telemetry 參照](#)

相關資訊

第 59 頁的『IBM WebSphere MQ Telemetry』

IBM WebSphere MQ Version 7.5 資訊導覽圖

資訊導覽圖包含各種 IBM WebSphere MQ Version 7.5 資源的鏈結。

此導覽圖結合來自不同來源的資訊，以協助您進一步瞭解 IBM WebSphere MQ 的特定區域。按一下導覽圖中每一個區段的鏈結，以查看可用的資源。

- [產品概觀](#)
- [技術概觀](#)
- [實務](#)
- [規劃](#)
- [移轉及升級](#)
- [安裝](#)
- [安全](#)
- [配置](#)
- [管理](#)
- [開發應用程式](#)
- [監視及效能](#)
- [疑難排解與支援](#)
- [參考](#)

表 1: IBM WebSphere MQ 資訊導覽圖表格

種類	資訊資源
產品概觀	<p>IBM WebSphere MQ 的整體用途、功能及新增特性概觀。</p> <p> 第 5 頁的『關於 IBM WebSphere MQ』 協助您開始使用 IBM WebSphere MQ Version 7.5 的介紹資訊，包括產品簡介，以及此版本的新增功能及變更內容概觀。</p> <p>IBM WebSphere MQ Version 7.5 本 IBM Redbooks 出版品涵蓋 IBM WebSphere MQ Version 7.5 中所提供的核心加強功能，以及必須瞭解的概念。</p> <p>IBM WebSphere MQ 產品網頁 含有資源鏈結及其他資訊的產品網頁。</p> <p>IBM WebSphere MQ 系統需求 包含不同 IBM WebSphere MQ 版本之系統需求鏈結的網頁。</p> <p>第 15 頁的『IBM Documentation 離線 應用程式中的 IBM WebSphere MQ Version 7.5』 您可以將 IBM WebSphere MQ Version 7.5 傳訊文件下載到您在本機安裝的 IBM Documentation 離線版。</p>
技術概觀	<p> 第 37 頁的『IBM WebSphere MQ 技術概觀』 協助您瞭解 IBM WebSphere MQ Version 7.5 提供的訊息佇列作業及其他特性的相關資訊。</p>

表 1: IBM WebSphere MQ 資訊導覽圖表格 (繼續)

種類	資訊資源
實務	<p>每一個實務範例都會引導您完成一組重要的作業，並協助您配置主要產品特性。實務範例包括指向其他內容的有用鏈結，可協助您進一步瞭解您感興趣的區域。</p> <ul style="list-style-type: none"> ❖ <u>第 118 頁的『案例』</u> 入門實務說明如何開始使用 IBM WebSphere MQ。如果您之前未使用過 IBM WebSphere MQ 且想要快速開始使用，請使用此實務範例。進一步的實務範例會引導您執行適當的作業步驟，以協助您配置或使用產品特性。 ❖ <u>將 WebSphere Application Server 連接至 IBM WebSphere MQ</u> 包含在各種實務範例中引導您完成將 WebSphere Application Server 連接至 IBM WebSphere MQ 所需之主要作業的資訊。 ❖ <u>將 WebSphere Application Server Liberty 設定檔連接至 IBM WebSphere MQ</u> 包含資訊，可引導您在各種實務中完成將 WebSphere Application Server Liberty 設定檔連接至 IBM WebSphere MQ 所需的主要作業。 ❖ <u>將 IBM MessageSight 連接至 IBM WebSphere MQ 及 WebSphere Application Server</u> 包含的資訊可引導您在各種實務範例中完成將 IBM MessageSight 連接至 IBM WebSphere MQ 及 WebSphere Application Server 所需的重要作業。
規劃	<ul style="list-style-type: none"> ❖ <u>規劃</u> 規劃 IBM WebSphere MQ 環境時，請考量 IBM WebSphere MQ 為單一及多個佇列管理程式架構提供的支援，以及點對點及發佈/訂閱傳訊樣式的支援。同時規劃您的資源需求，以及使用記載和備份機能。
移轉及升級	<ul style="list-style-type: none"> ❖ <u>移轉及升級</u> 若要移轉佇列管理程式以在新的程式碼層次上執行，您必須先升級 IBM WebSphere MQ 以安裝新的程式碼層次。當您驗證升級成功時，請移轉佇列管理程式及其所有相關聯的應用程式和資源。在啟動此程序之前，請根據此移轉手冊中的資訊建立移轉計劃。如果您正在套用維護，則不需要移轉。不過，您應該使用新的 IBM WebSphere MQ 程式碼層次來測試應用程式 ❖ <u>WebSphere / IBM MQ 移轉手冊</u> 本手冊提供資訊來協助您規劃從舊版本移轉至新版本的程序。您可以在 Web 瀏覽器中檢視手冊，或將它下載為 PDF 檔。
安裝	<ul style="list-style-type: none"> ❖ <u>安裝及解除安裝</u> 協助您準備安裝、安裝產品及驗證安裝的資訊。還有一些資訊可協助您解除安裝產品。
安全	<ul style="list-style-type: none"> ❖ <u>安全</u> 在 IBM WebSphere MQ 安裝中要考量的安全層面，包括識別與鑑別、授權、審核、機密性及資料完整性。
配置	<ul style="list-style-type: none"> ❖ <u>配置</u> 在一或多部電腦上建立一或多個佇列管理程式，並在開發、測試及正式作業系統上配置它們及其相關資源，以處理包含商業資料的訊息。
管理	<ul style="list-style-type: none"> ❖ <u>管理 IBM WebSphere MQ</u> 管理佇列管理程式及相關聯的資源。

種類	資訊資源
開發應用程式	<p>❖ 開發應用程式 開發應用程式以傳送及接收訊息，以及管理佇列管理程式和相關資源。 IBM WebSphere MQ 支援以程序化語言撰寫的應用程式，以及物件導向的語言和架構。</p>
監視及效能	<p>❖ 監視及效能 監視資訊及指引，協助改善佇列管理程式網路的效能，以及協助改善佇列管理程式網路效能的調整提示。</p>
疑難排解與支援	<p>❖ 疑難排解及支援 協助您診斷及解決佇列管理程式網路或 IBM WebSphere MQ 應用程式問題的技術。</p> <p>IBM Support Assistant 網頁 IBM Support Assistant (ISA) 可讓您存取支援相關資訊及疑難排解工具，以協助您解決 IBM 軟體產品的問題。</p> <p>IBM 支援中心入口網站網頁 IBM IBM WebSphere MQ 的支援中心入口網站。</p>
參考	<p>❖ 參照 配置、管理、開發應用程式、遙測、安全、監視、疑難排解和支援以及診斷訊息的參照資訊。</p>

IBM Documentation 離線 應用程式中的 IBM WebSphere MQ Version 7.5

如果您在氣隙環境中無法存取網際網路，請使用我們的 Dark Shop 應用程式 "IBM Documentation 離線" 來檢視 IBM WebSphere MQ Version 7.5 產品說明文件的下載。

IBM Documentation 離線 有兩個元件：

- **IBM Documentation 離線 應用程式。** 這是 IBM Documentation 的本端可安裝離線版本。
- **您安裝至 IBM Documentation 離線 應用程式的說明文件套件。** 這些套件包含的文件與在 IBM Documentation 中線上發佈的文件相同。

若要下載應用程式及 IBM WebSphere MQ Version 7.5 文件套件，您需要登入 IBM Documentation。如需相關資訊，請參閱 [IBM Documentation 離線](#)。

IBM WebSphere MQ Version 7.5 PDF 說明文件

您可以將 IBM WebSphere MQ Version 7.5 產品說明文件下載為一系列 PDF 檔案。

IBM WebSphere MQ Version 7.5 PDF 檔案包含與 IBM Documentation 中發佈的 HTML 版本相同的資訊。您可以按下表中的鏈結來下載它們。

註：您也可以將 HTML 格式的 IBM WebSphere MQ Version 7.5 產品說明文件下載至本端安裝的 IBM Documentation 離線版本。如需相關資訊，請參閱第 15 頁的『[IBM Documentation 離線 應用程式中的 IBM WebSphere MQ Version 7.5](#)』。

PDF 檔名及下載鏈結	IBM Documentation 中的對等區段
mq75.overview.pdf	關於 IBM WebSphere MQ
mq75.plan.pdf	規劃
mq75.migrate.pdf	移轉及升級

表 2: 產品說明文件的 PDF 檔及其對應於 IBM Documentation 中各節的方式 (繼續)

PDF 檔名及下載鏈結	IBM Documentation 中的對等區段
mq75.install.pdf	安裝
mq75.secure.pdf	安全
mq75.configure.pdf	配置
mq75.administer.pdf	管理
mq75.develop.pdf	開發應用程式
mq75.monitor.pdf	監視及效能
mq75.troubleshoot.pdf	疑難排解和支援
mq75.reference.pdf	參照
mq75.refcon.pdf	配置參照
mq75.refadmin.pdf	管理參照
mq75.refdev.pdf	開發應用程式參照
mq75.mft.pdf	IBM WebSphere MQ Managed File Transfer
mq75.explorer.pdf	IBM WebSphere MQ Explorer
mq75.guiwizard.pdf	MQ GUI 精靈說明
mq75.xms.pdf	Message Service Client for .NET
mq75.hve.pdf	WebSphere MQ Hypervisor Edition
mq75.mmtc.pdf	Mobile Messaging and M2M
MQ_Migration_Guide.pdf	WebSphere / IBM MQ 移轉手冊

註: PDF 檔案必須位於相同的資料夾中, PDF 檔案之間的鏈結才能正確運作。

IBM WebSphere MQ Version 7.5 的新增功能

瞭解 IBM WebSphere MQ Version 7.5 中的主要新功能。

- 第 16 頁的『[IBM WebSphere MQ Managed File Transfer](#)』。
- 第 17 頁的『[IBM WebSphere MQ Advanced Message Security](#)』。
- 第 17 頁的『[訊息通道代理程式 \(MCA\) 截取](#)』。
- 第 17 頁的『[多個叢集傳輸佇列](#)』。
- 第 17 頁的『[延伸交易式功能現在是核心用戶端的一部分](#)』。
- 第 17 頁的『[透過設定應用程式名稱來識別佇列管理程式的連線](#)』。
- 第 17 頁的『[憑證驗證原則](#)』。
- 第 17 頁的『[更多交易式可見性](#)』。
- 第 17 頁的『[實務練習](#)』。
- 第 17 頁的『[IBM WebSphere MQ Explorer](#)』。

IBM WebSphere MQ Managed File Transfer

IBM WebSphere MQ 受管理檔案傳送使用 IBM WebSphere MQ 在佇列管理程式之間傳送檔案。您可以將其觸角延伸至沒有佇列管理程式的工作站及伺服器。您可以使用檔案傳送代理程式 Apache Ant 來延伸它, 並將其與 IBM Sterling Commerce:Direct、Web 閘道以及 SFTP、FTP 或 FTPS 通訊協定伺服器整合。

使用 IBM WebSphere MQ 受管理檔案傳送，您可以自動化、控制、保護及審核檔案傳送; 請參閱 [IBM WebSphere MQ 受管理檔案傳送簡介](#)。

IBM WebSphere MQ Advanced Message Security

IBM WebSphere MQ Advanced Message Security (AMS) 是個別安裝的元件，可個別收費。它為流經 IBM WebSphere MQ 網路的機密資料提供高階保護。您不需要修改現有應用程式即可利用 AMS，請參閱 [IBM WebSphere MQ Advanced Message Security](#)。

訊息通道代理程式 (MCA) 截取

MCA 攔截特性可讓在 IBM WebSphere MQ 下執行的佇列管理程式，以及 Advanced Message Security 的授權安裝，選擇性地啟用伺服器連線通道所套用的原則。MCA 截取可讓留在 IBM WebSphere MQ AMS 外部的用戶端仍連接至佇列管理程式，並將其訊息加密及解密。請參閱 [訊息通道代理程式 \(MCA\) 截取](#)。

多個叢集傳輸佇列

您可以變更新的佇列管理程式屬性 **DEFCLXQ**，將不同的叢集傳輸佇列指派給每一個叢集傳送端通道。每一個叢集傳送端通道要轉遞的訊息會放在個別叢集傳輸佇列上; 請參閱 [叢集傳輸佇列和叢集傳送端通道](#)。您也可以設定新的佇列屬性 **CLCHNAME**，以手動配置叢集傳輸佇列。您可以決定哪些叢集傳送端通道會共用哪些傳輸佇列、哪些具有個別傳輸佇列，以及哪些使用叢集傳輸佇列; 請參閱 [叢集作業: 規劃如何配置叢集傳輸佇列](#)。此變更會協助管理叢集佇列管理程式之間訊息傳送的系統管理者。

延伸交易式功能現在是核心用戶端的一部分

延伸交易式功能現在已納入 IBM WebSphere MQ 核心用戶端。您不需要購買個別延伸交易式用戶端授權，或安裝個別延伸交易式用戶端元件; 請參閱 [第 113 頁的『何謂延伸交易式用戶端?』](#)。

透過設定應用程式名稱來識別佇列管理程式的連線

應用程式可以設定名稱來識別其與佇列管理程式的連線。使用 **DISPLAY CONN** 指令顯示應用程式名稱。名稱會在 **APPLTAG** 欄位中傳回。您也可以 IBM WebSphere MQ Explorer [應用程式連線](#) 視窗中顯示名稱。此欄位稱為 **App name**; 請參閱 [設定 WebSphere MQ for Java 類別的 WebSphere MQ 環境](#)。您可以在所有平台上設定應用程式連線的名稱，但 z/OS 除外。

憑證驗證原則

在 UNIX, Linux, and Windows 上，您可以指定憑證鏈驗證符合 RFC 5280 業界安全標準的嚴格程度; 請參閱 [WebSphere MQ](#)。

更多交易式可見性

dspmqrn 指令具有兩個新參數: **-a** 及 **-q**，以在發生非同步回復時提供相關資訊。兩個新訊息 [AMQ7486](#) 及 [AMQ7487](#) 提供所回復交易的相關資訊，以及交易是否與連線相關聯。

實務練習

第 118 頁的『案例』顯示如何快速且輕鬆地使用及結合新的 IBM WebSphere MQ Version 7.5 函數。實務範例包括產品說明文件內容的有用鏈結，以協助您進一步瞭解您感興趣的區域。

IBM WebSphere MQ Explorer

IBM WebSphere MQ Explorer Version 7.5 中的新增特性包括 IBM WebSphere MQ Managed File Transfer 與 IBM WebSphere MQ Advanced Message Security 的整合、改良的多版本支援，以及減少額外負擔。

如需 IBM WebSphere MQ Explorer 中新增功能的詳細資料，請參閱 [「WebSphere MQ 探險家」](#) 中的新增功能及變更內容。

相關概念

[第 21 頁的『IBM WebSphere MQ Version 7.5 中的變更內容』](#)

[第 21 頁的『IBM WebSphere MQ Version 7.5 修正套件中的變更內容』](#)

本節說明 Version 7.5 Fix Pack 中功能及資源的變更。

HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端

IBM WebSphere MQ 現在支援 HP Integrity NonStop Server 平台的用戶端。

概觀

如需 IBM WebSphere MQ 用戶端 (包括 HP Integrity NonStop Server 平台的用戶端) 的概觀，請參閱 [第 110 頁的『IBM WebSphere MQ MQI 用戶端的概觀』](#)。

如需 HP Integrity NonStop Server 平台的 IBM WebSphere MQ 用戶端技術概觀，請參閱 [第 117 頁的『IBM WebSphere MQ Client for HP Integrity NonStop Server 技術概觀』](#)。

如需 HP Integrity NonStop Server 所支援環境及特性的 IBM WebSphere MQ 用戶端詳細資料，請參閱 [第 117 頁的『IBM WebSphere MQ 用戶端 \(適用於 HP Integrity NonStop Server 支援的環境和特性\)』](#)。

規劃

當您規劃 HP Integrity NonStop Server 環境的 IBM WebSphere MQ 用戶端時，如需說明，請參閱 [在 HP Integrity NonStop Server 上規劃 IBM WebSphere MQ 用戶端環境](#)。

安裝

安裝 HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端的相關說明。

- 選擇要安裝的項目，請參閱 [HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端元件](#)。
- 規劃安裝，請參閱 [規劃 HP Integrity NonStop Server 上的安裝](#)。
 - 檔案系統
- 軟硬體需求，請參閱 [HP Integrity NonStop Server 上的軟硬體需求](#)。
- 驗證您是否具有正確的軟體，請參閱 [驗證系統軟體必備項目](#)。
- 準備系統，請參閱 [在 HP Integrity NonStop Server 上設定使用者和群組](#)。
- 安裝用戶端，請參閱 [在 HP Integrity NonStop Server 系統上安裝 IBM WebSphere MQ 用戶端](#)。
- 驗證安裝，請參閱 [驗證用戶端安裝](#)。
- 解除安裝，請參閱 [在 HP Integrity NonStop Server 上解除安裝 IBM WebSphere MQ](#)。

HP Integrity NonStop Server 用戶端指令

下列指令適用於 HP Integrity NonStop Server OSS 及 Guardian 環境的 IBM WebSphere MQ 用戶端：

- [dspmqver](#)
- [endmqtrc](#)
- [mqrc](#) (MQ 回覆碼)
- [runmqras](#)
- [runmqtmc](#)
- [strmqtrc](#)

下列指令適用於 HP Integrity NonStop Server OSS 環境的 IBM WebSphere MQ 用戶端：

- [dspmqtrc](#)

新產品 ID MQNC，已新增至 [DISPLAY CHSTATUS](#) 指令的 [產品 ID 值](#) 表格。

安全

若要保護 IBM WebSphere MQ Client for HP Integrity NonStop Server 環境的安全，請參閱：

- [HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端如何向新增至 規劃用戶端應用程式的鑑別的佇列管理程式識別本身的相關資訊。](#)
- [在 HP Integrity NonStop Server 上設定安全](#)
 - [OpenSSL](#)
 - [熵常駐程式](#)
- [IBM WebSphere MQ 支援 SSL 和 TLS](#)
- [在 HP Integrity NonStop Server 上使用 SSL 或 TLS](#)
 - [憑證管理](#)
 - [個人憑證儲存庫](#)
 - [憑證信任儲存庫](#)
 - [通行詞組隱藏檔](#)
 - [憑證撤銷清冊檔案](#)

交易管理機能

如需「交易管理機能 (TMF)」的相關資訊，請參閱下列各節及主題。

- [在 HP Integrity NonStop Server 上規劃 IBM WebSphere MQ 用戶端環境](#)
 - [準備 HP Integrity NonStop Server 環境](#)
 - [IBM WebSphere MQ 和 HP NonStop TMF](#)
 - [使用 HP NonStop TMF](#)
 - [使用廣域工作單元](#)
 - [避免長時間執行交易](#)
 - [佇列管理程式配置的相關資訊，可讓廣域工作單元在預先配置的閒置間隔之後到期，並新增至 \[使廣域工作單元到期\]\(#\)。](#)
- [正在配置 HP Integrity NonStop Server](#)
 - [閘道處理程序概觀](#)
 - [配置閘道以在 Pathway 下執行](#)
 - [TMF 及 TMF/ 閘道段落](#)
 - [配置用戶端起始設定檔](#)
 - [將許可權授與通道](#)
- [管理 HP Integrity NonStop Server](#)
 - [從 Pathway 手動啟動 TMF/ 閘道](#)
 - [從 Pathway 停止 TMF/ 閘道](#)
- [HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端疑難排解](#)

開發應用程式

如需在 HP Integrity NonStop Server 平台上為 IBM WebSphere MQ 用戶端開發應用程式的相關資訊，請參閱：

- [在 HP Integrity NonStop Server 上建置應用程式](#)
 - [OSS 和 Guardian 標頭及公用程式庫](#)
 - [在 HP Integrity NonStop Server 中準備 C 程式](#)
 - [準備 COBOL 程式](#)

- [準備 pTAL 程式](#)
- 在 pTAL 中編碼。
- [為 HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端準備 JMS 程式。](#)

新訊息

以下是 HP Integrity NonStop Server 上 IBM WebSphere MQ 用戶端的新訊息：

- AMQ5000-5999: 可安裝服務

- [AMQ5370](#)
- [AMQ5371](#)
- [AMQ5372](#)
- [AMQ5373](#)
- [AMQ5374](#)
- [AMQ5375](#)
- [AMQ5376](#)
- [AMQ5377](#)
- [AMQ5378](#)
- [AMQ5379](#)
- [AMQ5380](#)
- [AMQ5390](#)
- [AMQ5391](#)
- [AMQ5392](#)
- [AMQ5393](#)
- [AMQ5394](#)
- [AMQ5395](#)
- [AMQ5396](#)
- [AMQ5397](#)
- [AMQ5398](#)
- [AMQ5399](#)

- AMQ9000-9999: 遠端

- [AMQ9816](#)
- [AMQ9817](#)
- [AMQ9818](#)
- [AMQ9819](#)
- [AMQ9820](#)
- [AMQ9821](#)
- [AMQ9823](#)
- [AMQ9824](#)

已修改 API 原因碼

下列現有的 API 原因碼現在包括 HP Integrity NonStop Server:

- [2354 \(0932\) \(RC2354\) :MQRC_UOW_ENLISTMENT_ERROR](#)
- [2355 \(0933\) \(RC2355\): MQRC_UOW_MIX_NOT_SUPPORTED](#)
- [2072 \(0818\) \(RC2072\) :XX_ENCODE_CASE_ONE mqrc_syncpoint_not_available](#)

- [2003 \(07D3\) \(RC2003\) :MQRC_BACKED_OUT](#)

範例

如需範例程式所示範技術的相關資訊，請參閱 [適用於 HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端的範例](#)。

疑難排解和支援

如需 HP Integrity NonStop Server 上 IBM WebSphere MQ 用戶端的疑難排解及支援資訊，請參閱下列主題：

- [HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端疑難排解](#)
- [HP Integrity NonStop Server 上的錯誤日誌](#)
- [在 HP Integrity NonStop Server 上使用追蹤](#)
- [FFST: IBM WebSphere MQ 適用於 HP Integrity NonStop Server](#)

IBM WebSphere MQ Version 7.5 中的變更內容

在將佇列管理程式升級至 IBM WebSphere MQ Version 7.5 之前，請仔細檢閱變更清單。決定在開始將系統移轉至 Version 7.5 之前，是否必須計劃對現有應用程式、Script 及程序進行變更。

下列鏈結指向產品說明文件的移轉及升級小節中的資訊。這裡未列出新功能及不影響現有應用程式、管理程序及管理 Script 的變更；請參閱第 16 頁的『[IBM WebSphere MQ Version 7.5 的新增功能](#)』。

[依版本、版次及維護層次列出的變更清單](#)

- [V7.1 至 V7.5 變更](#)

相關概念

第 16 頁的『[IBM WebSphere MQ Version 7.5 的新增功能](#)』
瞭解 IBM WebSphere MQ Version 7.5 中的主要新功能。

第 21 頁的『[IBM WebSphere MQ Version 7.5 修正套件中的變更內容](#)』
本節說明 Version 7.5 Fix Pack 中功能及資源的變更。

IBM WebSphere MQ Version 7.5 修正套件中的變更內容

本節說明 Version 7.5 Fix Pack 中功能及資源的變更。

- **V7.5.0.1** 第 23 頁的『[Version 7.5.0, Fix Pack 1: 支援 MQTT over WebSockets](#)』
- **V7.5.0.4** 第 23 頁的『[Version 7.5.0, Fix Pack 4: 使用 AMQ_DISABLE_CLIENT_AMS 在用戶端停用 IBM WebSphere MQ Advanced Message Security](#)』
- **V7.5.0.5** 第 23 頁的『[Version 7.5.0, Fix Pack 5: 使用 mqclient.ini 檔案在用戶端停用 IBM WebSphere MQ Advanced Message Security](#)』
- **Windows V7.5.0.6** 第 23 頁的『[Version 7.5.0, Fix Pack 6: 受管理 .NET 用戶端應用程式的使用者 ID 及密碼](#)』
- **V7.5.0.6** 第 24 頁的『[Version 7.5.0, Fix Pack 6: 已淘汰 cipherspecs](#)』
- **V7.5.0.6** 第 24 頁的『[Version 7.5.0, Fix Pack 6: IBM WebSphere MQ Managed File Transfer 的服務功能加強功能](#)』
- **V7.5.0.6** 第 24 頁的『[Version 7.5.0, Fix Pack 6 新建 IBM WebSphere MQ Managed File Transfer 代理程式內容 failTransferOnFirst 失敗](#)』
- **V7.5.0.7** 第 24 頁的『[Version 7.5.0, Fix Pack 7: 已淘汰 cipherspecs](#)』

- **Windows** **V 7.5.0.7** [第 25 頁的『Version 7.5.0, Fix Pack 7: 支援 JDBC 連線至 Oracle 12c 資料庫』](#)
- **Windows** **V 7.5.0.7** [第 25 頁的『Version 7.5.0, Fix Pack 7: 移除使用 .NET for MQCNO_CLIENT_BINDING 及 MQCNO_LOCAL_BINDING 的限制』](#)
- **V 7.5.0.8** [第 25 頁的『Version 7.5.0, Fix Pack 8: 在配送清單中使用主題別名佇列的限制』](#)
- **V 7.5.0.8** [第 25 頁的『Version 7.5.0, Fix Pack 8: GSKit 版本已更新』](#)
- **V 7.5.0.8** [第 25 頁的『Version 7.5.0, Fix Pack 8: 已淘汰 CipherSpecs』](#)
- **V 7.5.0.8** [第 26 頁的『Version 7.5.0, Fix Pack 8: 新常數 JMS_IBM_SUBSCRIPTION_USER_DATA 已新增至 JmsConstants 介面』](#)
- **V 7.5.0.8** [第 26 頁的『Version 7.5.0, Fix Pack 8: JMS 異常狀況接聽器更新項目』](#)
- **V 7.5.0.8** [第 26 頁的『Version 7.5.0, Fix Pack 8: 支援 IBM WebSphere MQ classes for JMS ObjectMessage 中的類別名稱允許清單』](#)
- **V 7.5.0.8** [第 26 頁的『Version 7.5.0, Fix Pack 8: 新的 IBM WebSphere MQ Managed File Transfer 代理程式內容 additionalWildcardSandboxChecking』](#)
- **V 7.5.0.8** [第 26 頁的『Version 7.5.0, Fix Pack 8: 變更為 IBM WebSphere MQ Managed File Transfer fteCleanAgent 指令的行為』](#)
- **V 7.5.0.9** [第 27 頁的『Version 7.5.0, Fix Pack 9: 新環境變數 AMQ_SSL_LDAP_SERVER_VERSION』](#)
- **V 7.5.0.9** [第 27 頁的『Version 7.5.0, Fix Pack 9: IBM WebSphere MQ classes for JMS 的範例配置檔』](#)
- **V 7.5.0.9** [第 27 頁的『Version 7.5.0, Fix Pack 9: 對 dmpmqcfg 指令的變更』](#)
- **V 7.5.0.9** [第 27 頁的『Version 7.5.0, Fix Pack 9: 對 strmqm 指令的變更』](#)
- **V 7.5.0.9** [第 27 頁的『runmqras 公用程式的加強功能』](#)
- **V 7.5.0.9** [第 28 頁的『Version 7.5.0, Fix Pack 9: 已還原訊息轉為檔案傳送的傳送 ID 與 groupId 屬性值的比較檢查』](#)
- **V 7.5.0.9** [第 27 頁的『Version 7.5.0, Fix Pack 9: 啟動規格上 targetClientMatching 內容的資源配接器更新項目』](#)
- **V 7.5.0.9** [第 28 頁的『Version 7.5.0, Fix Pack 9: runmqras 指令的新 -caseno 參數』](#)
- **V 7.5.0.9** [第 28 頁的『Version 7.5.0, Fix Pack 9: 變更為處理與佇列管理程式叢集元件所使用的內部 MQ 指令訊息內毀損字元資料相關的錯誤』](#)
- **V 7.5.0.9** **Windows** [第 28 頁的『Version 7.5.0, Fix Pack 9: 在 Windows 上新增 Microsoft Visual Studio 2013 C/C++ 執行時期』](#)
- **V 7.5.0.9** [第 28 頁的『Version 7.5.0, Fix Pack 9: 已還原訊息轉為檔案傳送的傳送 ID 與 groupId 屬性值的比較檢查』](#)
- **V 7.5.0.9** [第 28 頁的『Version 7.5.0, Fix Pack 9: 已還原訊息轉為檔案傳送的傳送 ID 與 groupId 屬性值的比較檢查』](#)
- **V 7.5.0.9** [第 28 頁的『Version 7.5.0, Fix Pack 9 外加 APAR IT26482: 變更為 IBM WebSphere MQ classes for JMS 所需的權限, 以查詢叢集佇列的 BackoutThreshold 及 BackoutRequeueQName』](#)

Version 7.5.0, Fix Pack 1: 支援 MQTT over WebSockets

V 7.5.0.1

新的通訊協定參數 (PROTOCOL) 已新增至 [MQTT 通道定義 \(DEFINE CHANNEL \(MQTT\)\)](#):

- 如果參數設為 MQTTV3, 則通道僅接受使用 MQ Telemetry Transfer 通訊協定第 3 版來自用戶端的連線。這是 IBM WebSphere MQ Version 7.5.0Fix Pack 1 之前唯一支援的通訊協定。
- 如果參數設為 HTTP, 通道只會接受頁面的 HTTP 要求, 或 IBM WebSphere MQ Telemetry 的 WebSockets 連線。
- 如果參數設為 MQTTV3, HTTP, 則通道會接受使用任一通訊協定來自用戶端的連線。這是使用 IBM WebSphere MQ Version 7.5.0Fix Pack 1 以及更新版本建立的新 MQTT 通道的預設行為。

如需相關資訊, 請參閱 [透過 SSL 連接 適用於 JavaScript 的 MQTT 傳訊用戶端 及 WebSockets](#)。

當用戶端使用 SSL 連接至 MQTT 通道時, 參數 SSLCAUTH 會決定 IBM WebSphere MQ 是否需要來自用戶端的憑證 (請參閱 [DEFINE CHANNEL \(MQTT\)](#))。在 IBM WebSphere MQ Version 7.5.0Fix Pack 1 之前, 對於 MQTT 通道, 此參數可以是 REQUIRED 或 OPTIONAL :

- REQUIRED 表示 IBM WebSphere MQ 向用戶端要求憑證, 且用戶端必須提供有效憑證。
- OPTIONAL 表示 IBM WebSphere MQ 會向用戶端要求憑證, 但用戶端不需要提供憑證。如果用戶端提供有效憑證或用戶端未提供憑證, 則容許用戶端連線。只有在用戶端提供無效憑證時, 才會禁止用戶端連線。

在 IBM WebSphere MQ Version 7.5.0Fix Pack 1 以及更新版本中, 對於 MQTT 通道, 參數 SSLCAUTH 可以設為 NEVER。NEVER 表示 IBM WebSphere MQ 永不從用戶端要求憑證。在支援使用 JavaScript 的 MQTT 傳訊用戶端的用戶端時, 新增了新值。它可容納將用戶端憑證要求視為通訊協定錯誤的部分 Web 瀏覽器的行為。

Version 7.5.0, Fix Pack 4: 使用 AMQ_DISABLE_CLIENT_AMS 在用戶端停用 IBM WebSphere MQ Advanced Message Security

V 7.5.0.4

從 Version 7.5.0, Fix Pack 4 開始, 您可以使用環境變數 AMQ_DISABLE_CLIENT_AMS, 在 IBM WebSphere MQ classes for JMS 及 IBM WebSphere MQ classes for Java 用戶端內停用 IBM WebSphere MQ Advanced Message Security (AMS), 以防止在連接至在舊版產品上執行的佇列管理程式時發生錯誤。或者, 您也可以從 Version 7.5.0, Fix Pack 4 使用 Java 系統內容 com.ibm.mq.cfg.AMQ_DISABLE_CLIENT_AMS, 在用戶端停用 AMS。

如需相關資訊, 請參閱 [在用戶端停用 IBM WebSphere MQ Advanced Message Security](#)。

Version 7.5.0, Fix Pack 5: 使用 mqclient.ini 檔案在用戶端停用 IBM WebSphere MQ Advanced Message Security

V 7.5.0.5

從 Version 7.5.0, Fix Pack 4 開始, 您可以透過設定配置變數 AMQ_DISABLE_CLIENT_AMS, 在用戶端停用 IBM WebSphere MQ Advanced Message Security (AMS), 以防止在連接至在舊版產品上執行的佇列管理程式時發生錯誤。從 Version 7.5.0, Fix Pack 5 中, 您也可以使用 mqclient.ini 檔案中 **Security** 段落下的 DisableClientAMS 內容來停用 AMS。如需相關資訊, 請參閱 [在用戶端停用 IBM WebSphere MQ Advanced Message Security](#)。





Version 7.5.0, Fix Pack 6: 受管理 .NET 用戶端應用程式的使用者 ID 及密碼

Windows V 7.5.0.6

從 IBM WebSphere MQ Version 7.5.0Fix Pack 6 開始, 隨受管理 .NET 用戶端應用程式指定的使用者 ID 及密碼會設定在傳遞至用戶端安全結束程式的 IBM WebSphere MQ .NET MQChannelDefinition 類別中。如需相關資訊, 請參閱 [在 IBM WebSphere MQ .NET 中使用通道結束程式](#)。

Version 7.5.0, Fix Pack 6: 已淘汰 cipherspecs

IBM WebSphere MQ Version 7.5.0Fix Pack 6 中已淘汰下列密碼規格:

- DES_SHA_EXPORT
-  DES_SHA_EXPORT1024
-  FIPS_WITH_DES_CBC_SHA
- NULL_MD5
- NULL_SHA
- TLS_RSA_WITH_DES_CBC_SHA
- ECDHE_ECDSA_NULL_SHA256
-  ECDHE_RSA_NULL_SHA256
-  TLS_RSA_WITH_NULL_NULL
- TLS_RSA_WITH_NULL_SHA256

如需相關資訊，請參閱 [已淘汰的密碼規格](#)。

Version 7.5.0, Fix Pack 6: IBM WebSphere MQ Managed File Transfer 的服務功能加強功能

V 7.5.0.6

從 IBM WebSphere MQ Version 7.5.0Fix Pack 6 開始，已對 IBM WebSphere MQ Managed File Transfer 進行下列變更:

- `installation.properties` 檔案中 `commandMessagePriority` 內容的預設值已變更為 8。如需相關資訊，請參閱 [installation.properties](#) 檔案。
- `agent.properties` 檔案中 `logTransferRecovery` 內容的預設值已變更為 `true`。如需相關資訊，請參閱 [agent.properties](#) 檔案。
- 在 `fteShowAgentDetails` 指令上使用 `-d` 參數時，不再需要在 `installation.properties` 檔案中設定 `enableFunctionalFixPack` 內容。如需相關資訊，請參閱 [fteShowAgentDetails](#)。
- 如果代理程式發現無法復原的錯誤，則會產生首次失敗資料擷取 (FDC)。



Version 7.5.0, Fix Pack 6 新建 IBM WebSphere MQ Managed File Transfer 代理程式內容 `failTransferOnFirst` 失敗

V 7.5.0.6 distributed

從 IBM WebSphere MQ Version 7.5.0Fix Pack 6 開始，您可以透過在 `agent.properties` 檔案中設定 `failTransferOnFirstFailure` 內容，將代理程式配置成只要受管理傳送中的傳送項目失敗，即會使該受管理傳送失敗。如需相關資訊，請參閱 [agent.properties](#) 檔案。

Version 7.5.0, Fix Pack 7: 已淘汰 cipherspecs

IBM WebSphere MQ Version 7.5.0Fix Pack 7 中已淘汰下列密碼規格:

- RC2_MD5_EXPORT
- RC4_MD5_EXPORT
- RC4_MD5_US
- RC4_SHA_US
-  RC4_56_SHA_EXPORT1024
-  ECDHE_ECDSA_RC4_128_SHA256

-    ECDHE_RSA_RC4_128_SHA256
-    TLS_RSA_WITH_RC4_128_SHA256

如需相關資訊，請參閱 [已淘汰的密碼規格](#)。

Version 7.5.0, Fix Pack 7: 支援 JDBC 連線至 Oracle 12c 資料庫

從 IBM WebSphere MQ Version 7.5.0 Fix Pack 7，IBM WebSphere MQ Windows 伺服器安裝映像檔隨附新的檔案 `jdbcora12.dll`，以支援 JDBC 與 Oracle 12c 資料庫的連線 (請參閱 [在 Windows 上配置 JTA/JDBC 協調](#))。

Version 7.5.0, Fix Pack 7: 移除使用 .NET for MQCNO_CLIENT_BINDING 及 MQCNO_LOCAL_BINDING 的限制

從 IBM WebSphere MQ Version 7.5.0 Fix Pack 7 開始，已更新 Microsoft Windows Communication Foundation (WCF) 的 IBM WebSphere MQ 自訂通道，以便在從僅限用戶端安裝執行時使用正確的用戶端連線配置。如需相關資訊，請參閱 [使用 MQCONN 呼叫連接至佇列管理程式](#)。

Version 7.5.0, Fix Pack 8: 在配送清單中使用主題別名佇列的限制



發佈清單不支援使用指向主題物件的別名佇列。從 Version 7.5.0, Fix Pack 8 開始，如果別名佇列指向配送清單中的主題物件，則 IBM WebSphere MQ 會傳回 `MQRC_ALIAS_BASE_Q_TYPE_ERROR`。

Version 7.5.0, Fix Pack 8: GSKit 版本已更新












GSKit 版本已更新。GSKit 的新版本會變更當您產生 `.sth` 檔案以隱藏金鑰資料庫密碼時所使用的隱藏檔格式。舊版 GSKit 無法讀取使用此 GSKit 版本產生的隱藏檔。若要確保使用 Version 7.5.0, Fix Pack 8 或更新版本產生的隱藏檔與您的應用程式及其他 IBM WebSphere MQ 安裝相容，您必須更新至包含 GSKit 相容版本的 IBM WebSphere MQ 版本。對於 IBM WebSphere MQ Version 7.5，這是 Version 7.5.0, Fix Pack 8。

如果無法更新應用程式或其他 IBM WebSphere MQ 安裝，您可以要求與舊版相容的隱藏檔格式。當您搭配使用 `runmqakm` 或 `runmqckm` 指令與 `-stash` 或 `-stashpw` 選項時，請包含 `-v1stash` 指令行參數。您無法使用 iKeyman GUI 來產生與舊版相容的隱藏檔。

Version 7.5.0, Fix Pack 8: 已淘汰 CipherSpecs



從 Version 7.5.0, Fix Pack 8 開始，下列 CipherSpecs 已淘汰：

-    FIPS_WITH_3DES_EDE_CBC_SHA
- TRIPLE_DES_SHA_US
- TLS_RSA_WITH_3DES_EDE_CBC_SHA
-    ECDHE_ECDSA_3DES_EDE_CBC_SHA256
-    ECDHE_RSA_3DES_EDE_CBC_SHA256

如需相關資訊，請參閱 [已淘汰的密碼規格](#)。

Version 7.5.0, Fix Pack 8: 新常數 JMS_IBM_SUBSCRIPTION_USER_DATA 已新增至 JmsConstants 介面

V 7.5.0.8

從 Version 7.5.0, Fix Pack 8 開始，會更新 IBM WebSphere MQ classes for JMS，以便當從包含 RFH2 標頭及 MQPS 資料夾的佇列中耗用訊息時，與 Sud 索引鍵相關聯的值 (如果存在的話) 會以「字串」內容新增至傳回 IBM WebSphere MQ classes for JMS 應用程式的「JMS 訊息」物件。為了讓應用程式能夠從訊息擷取這個內容，會將新的常數 JMS_IBM_SUBSCRIPTION_USER_DATA 新增至 JmsConstants 介面。這個新內容可以與 `javax.jms.Message.getStringProperty(java.lang.String)` 方法搭配使用，以擷取訂閱使用者資料。如需相關資訊，請參閱 [擷取使用者訂閱資料](#) 及 [DEFINE SUB](#)。

Version 7.5.0, Fix Pack 8: JMS 異常狀況接聽器更新項目

V 7.5.0.8

From IBM WebSphere MQ Version 7.5.0 Fix Pack 8, to maintain behavior for current JMS applications that configure a JMS MessageListener and a JMS ExceptionListener, the default value for the `ASYNC_EXCEPTIONS` JMS ConnectionFactory property has been changed to `ASYNC_EXCEPTIONS_CONNECTIONBROKEN` for the IBM WebSphere MQ classes for JMS for Version 7.5. 因此，依預設，只有對應於連線已岔斷錯誤碼的異常狀況才會遞送至應用程式的 JMS ExceptionListener。同步或非同步消費訊息時，連線已岔斷異常狀況會遞送至異常狀況接聽器。

Version 7.5 的 IBM WebSphere MQ classes for JMS 也已更新，當應用程式使用的 JMS ConnectionFactory 將 `ASYNC_EXCEPTIONS` 內容設為值 `ASYNC_EXCEPTIONS_ALL` 時，會將訊息遞送至非同步訊息消費者期間所發生的非連線中斷錯誤相關 JMS 異常狀況遞送至已登錄的 ExceptionListener。

如需相關資訊，請參閱 [JMS: 7.5 版中的異常狀況接聽器變更](#)，以及 [IBM WebSphere MQ for JMS 類別中的異常狀況](#)。

Version 7.5.0, Fix Pack 8: 支援 IBM WebSphere MQ classes for JMS ObjectMessage 中的類別名稱允許清單

V 7.5.0.8

從 IBM WebSphere MQ Version 7.5.0 Fix Pack 8 開始，IBM WebSphere MQ classes for JMS 支援將 JMS ObjectMessage 介面實作中的類別列入允許清單。允許清單定義哪些 Java 類別可以使用 `ObjectMessage.setObject()` 序列化，並使用 `ObjectMessage.getObject()` 解除序列化。

如需相關資訊，請參閱 [JMS ObjectMessage 中的類別名稱允許清單](#)，以及在 [Java Security Manager](#) 下執行 IBM WebSphere MQ classes for JMS 應用程式。

Version 7.5.0, Fix Pack 8: 新的 IBM WebSphere MQ Managed File Transfer 代理程式內容 additionalWildcardSandboxChecking

V 7.5.0.8

從 IBM WebSphere MQ Version 7.5.0 Fix Pack 8 開始，如果已使用使用者或代理程式沙盤推演配置代理程式，以限制代理程式可以來回傳送檔案的位置，您可以將 `additionalWildcardSandboxChecking` 內容設為 `true`，以指定對該代理程式的萬用字元傳送進行其他檢查。如需相關資訊，請參閱 [萬用字元傳送的其他檢查](#) 及 [agent.properties](#) 檔案。

Version 7.5.0, Fix Pack 8: 變更為 IBM WebSphere MQ Managed File Transfer fteCleanAgent 指令的行為

V 7.5.0.8

從 IBM WebSphere MQ Version 7.5.0 Fix Pack 8 開始，`fteCleanAgent` 指令已更新，因此您必須將適當的參數傳遞至指令，並提供代理程式名稱，以指定要清除的 IBM WebSphere MQ Managed File Transfer 狀態。行為中的這項變更可確保依預設，`fteCleanAgent` 不會清除所指定代理程式的所有進行中及擱置中傳送、資源監視器定義及排定傳送定義。

必要的話，您可以將 `command.properties` 檔中新的 `failCleanAgentWithNoArguments` 內容設為 `false`，以回復為 **fteCleanAgent** 的先前行為。

如需相關資訊，請參閱 [fteCleanAgent \(清除受管理檔案傳送代理程式\)](#) 及 `command.properties` 檔案。

Version 7.5.0, Fix Pack 9: 新環境變數 AMQ_SSL_LDAP_SERVER_VERSION

V 7.5.0.9

從 Version 7.5.0, Fix Pack 9 開始，您可以設定環境變數 `AMQ_SSL_LDAP_SERVER_VERSION`，以確保當針對更高版本的「LDAP 憑證撤銷清冊 (CRL)」伺服器驗證憑證時，IBM WebSphere MQ 會將必要的輕量型目錄存取通訊協定 (LDAP) 版本 (即 LDAP v2 或 LDAP v3) 傳遞至 GSKit。如需相關資訊，請參閱 [環境變數](#)。

Version 7.5.0, Fix Pack 9: IBM WebSphere MQ classes for JMS 的範例配置檔

V 7.5.0.9

從 Version 7.5.0, Fix Pack 9 開始，會在 IBM WebSphere MQ classes for JMS 安裝目錄的 `bin` 子目錄中提供範例配置檔 `mjava.config`。如需相關資訊，請參閱 [適用於 Java 的 IBM MQ 類別配置檔](#)。

Version 7.5.0, Fix Pack 9: 對 dmpmqcfg 指令的變更

V 7.5.0.9

從 Version 7.5.0, Fix Pack 9 開始，將指令行參數 `-w` 新增至 **dmpmqcfg**。這個切換參數可讓您增加 **dmpmqcfg** 等待來自指令伺服器的回覆訊息的時間 (以秒為單位)。

如需相關資訊，請參閱 [dmpmqcfg](#)。

Version 7.5.0, Fix Pack 9: 對 strmqm 指令的變更

V 7.5.0.9

從 Version 7.5.0, Fix Pack 9 開始，在完全啟動佇列管理程式之前，**strmqm** 指令會檢查 `qm.ini` 檔案中 CHANNELS 及 SSL 段落的語法，這可讓您更容易查看錯誤的內容，並在 **strmqm** 發現 `qm.ini` 檔案包含任何錯誤時快速更正。如需相關資訊，請參閱 [strmqm](#)。

runmqras 公用程式的加強功能

V 7.5.0.9

從 Version 7.5.0, Fix Pack 9 開始，**runmqras** 公用程式具有下列加強功能：

- **AIX** **Solaris** **Linux** 依預設會擷取環境變數資訊。
- **distributed** 依預設會擷取佇列管理程式資料目錄清單。
- 在 **runmqras** 指令中新增下列兩個區段：
 - **Linux** **UNIX** `leak` 區段，用於收集 IBM WebSphere MQ 處理程序資源使用情形資訊。
 - **distributed** `mft` 區段，用於擷取 **fteRas** 指令所取得的資料。

如需相關資訊，請參閱 [runmqras](#)。

Version 7.5.0, Fix Pack 9: 啟動規格上 targetClientMatching 內容的資源配接器更新項目

V 7.5.0.9

IBM WebSphere MQ JCA 資源配接器 (MQ-RA) 已更新，因此可以針對啟動規格配置 **targetClientMatching** 內容。您可以配置這個內容，以便在要求訊息不包含 MQRFH2 標頭時，在回覆訊息中包含 MQRFH2 標頭。這表示在傳送訊息時，會併入應用程式在回覆訊息上定義的任何訊息內容。如需相關資訊，請參閱 [配置啟動規格的 targetClientMatching 內容](#)。

Version 7.5.0, Fix Pack 9: runmqras 指令的新 -caseno 參數

V 7.5.0.9

-caseno 參數相當於 -pmrno，但它容許指定有效的 Salesforce 觀察值編號。-caseno 或 -pmrno 都是選用參數，但不允許同時提供兩者。如需相關資訊，請參閱 [runmqras \(收集 IBM MQ 診斷資訊\)](#)。

Version 7.5.0, Fix Pack 9: 變更為處理與佇列管理程式叢集元件所使用的內部 MQ 指令訊息內毀損字元資料相關的錯誤

V 7.5.0.9

從 Version 7.5.0, Fix Pack 9 開始，佇列管理程式處理與佇列管理程式叢集元件所使用的內部 MQ 指令訊息內毀損字元資料相關的錯誤的方式已變更。會將可能包含不可轉換字元的欄位清空，並重試轉換。如果轉換成功，則會以空白說明儲存更新，並寫入錯誤訊息 AMQ6174 和 AMQ9880，以通知管理者。如果訊息仍然無法轉換，則會將它移至「無法傳送的郵件佇列」，並寫入錯誤訊息。

Version 7.5.0, Fix Pack 9: 在 Windows 上新增 Microsoft Visual Studio 2013 C/C++ 執行時期

V 7.5.0.9 Windows

從 Version 7.5.0, Fix Pack 9 開始，IBM WebSphere MQ 的部分元素需要 Microsoft Visual Studio 2013 C/C++ (VS2013) 執行時期。這些執行時期是由 IBM WebSphere MQ Version 7.5.0 Fix Pack 9 的任何全新安裝所安裝。

如果機器上沒有 VS2013 執行時期，則會在您安裝全新用戶端、將用戶端更新至 7.5.0.9 維護層次或套用 7.5.0.9 修正套件時安裝它們。在大部分情況下，此處理程序不需要使用者動作，但如果您在套用伺服器修正套件時遇到安裝失敗，則可能需要指定環境變數，以便您可以手動安裝執行時期。如需相關資訊，請參閱 [在伺服器修正套件中安裝 Microsoft Visual Studio 2013 C/C++ 執行時期](#)。

Version 7.5.0, Fix Pack 9: 已還原訊息轉為檔案傳送的傳送 ID 與 groupId 屬性值的比較檢查

V 7.5.0.9

從 Version 7.5.0, Fix Pack 9 開始，IBM WebSphere MQ Managed File Transfer 會更新以還原比較檢查，此功能用於對傳送要求 XML 內容中 groupId 屬性的傳送 ID 和值進行比較檢查，先前由 APAR IT18213 在 Version 7.5.0, Fix Pack 6 移除。如果這兩個 ID 相同，則在「訊息轉為檔案」傳送的輸入佇列中第一次進行 MQGET 嘗試時，來源代理程式會使用此 ID 作為訊息 ID 比對選項（相對於群組 ID 比對選項）。另請參閱 [將資料從訊息傳送至檔案](#)。

Version 7.5.0, Fix Pack 9 外加 APAR IT26482: 變更為 IBM WebSphere MQ classes for JMS 所需的權限，以查詢叢集佇列的 BackoutThreshold 及 BackoutQueueName

V 7.5.0.9

對於 Version 7.5.0, Fix Pack 9 以及 APAR IT26482 的臨時修正程式，IBM WebSphere MQ classes for JMS 已更新，因此只需要查詢存取權即可查詢叢集佇列的 **BackoutThreshold** 及 **BackoutQueueName**。對於所有其他版本，也需要瀏覽及取得存取權。如需相關資訊，請參閱 [處理 IBM MQ for JMS 類別中的有害訊息](#)。

相關概念

第 16 頁的『[IBM WebSphere MQ Version 7.5 的新增功能](#)』
瞭解 IBM WebSphere MQ Version 7.5 中的主要新功能。

第 21 頁的『[IBM WebSphere MQ Version 7.5 中的變更內容](#)』

相關資訊

[IBM WebSphere MQ 的系統需求](#)

[IBM MQ、WebSphere MQ 及 MQSeries 產品 ReadMe 網頁](#)

[WebSphere MQ 的建議修正程式](#)

舊版中的新增功能及變更內容

鏈結至功能及資源的新增特性及變更的相關資訊，包括在 IBM WebSphere MQ Version 7.5 之前的產品版本中發生的穩定、淘汰及移除。

如需舊版產品中的新增功能及變更內容的相關資訊，請參閱該版本產品說明文件中的適當章節。

IBM WebSphere MQ Version 7.1

- [IBM WebSphere MQ 7.1 版中的新增功能](#)
- [IBM WebSphere MQ 7.1 版中的變更內容](#)
- [IBM WebSphere MQ 7.1 版 Fix Pack 中的變更功能](#)

IBM WebSphere MQ Version 7.0.1 及更早版本

若為舊版產品，其中文件是在 IBM Documentation 外部提供，請參閱 [Documentation 適用於舊版 WebSphere MQ](#)。

相關概念

[第 16 頁的『IBM WebSphere MQ Version 7.5 的新增功能』](#)
瞭解 IBM WebSphere MQ Version 7.5 中的主要新功能。

[第 21 頁的『IBM WebSphere MQ Version 7.5 中的變更內容』](#)

[第 21 頁的『IBM WebSphere MQ Version 7.5 修正套件中的變更內容』](#)
本節說明 Version 7.5 Fix Pack 中功能及資源的變更。

[WebSphere MQ 移轉簡介](#)

舊版 IBM WebSphere MQ 出版品與現行資訊結構之間的對映

IBM Documentation 中的資訊是根據一組一般活動來結構化。例如，安裝、開發、管理、配置、保護安全。早期出版品 (MQ 書籍) 的結構略有不同。本節提供這些舊版 IBM WebSphere MQ 出版品與現行資訊結構之間的對映。

AIX 的快速入門

本節提供從 AIX 書籍的舊「快速入門」到新產品說明文件結構的對映：

- [AIX: 規劃移轉至更新版本](#)
- [檢查需求](#)
- [正在準備系統](#)
- [在 AIX 上安裝 IBM WebSphere MQ 伺服器](#)
- [驗證伺服器安裝](#)
- [在 AIX 系統上安裝 IBM WebSphere MQ 用戶端](#)
- [驗證用戶端安裝](#)
- [AIX: 在 IBM WebSphere MQ Version 7.5](#)
- [在 AIX 上解除安裝 IBM WebSphere MQ](#)

HP-UX 的快速入門

本節提供從 HP-UX 書籍的舊「快速入門」到新產品說明文件結構的對映：

- [HP-UX: 規劃移轉至更新版本](#)
- [檢查需求](#)
- [正在準備系統](#)

- [在 HP-UX 上安裝 IBM WebSphere MQ 伺服器](#)
- [驗證伺服器安裝](#)
- [在 HP-UX 系統上安裝 IBM WebSphere MQ 用戶端](#)
- [驗證用戶端安裝](#)
- [HP-UX: 在 IBM WebSphere MQ Version 7.5](#)
- [在 HP-UX 上解除安裝 IBM WebSphere MQ](#)

Linux 的快速入門

本節提供從 Linux 書籍的舊「快速入門」到現行產品說明文件結構的對映:

- [Linux: 規劃從 IBM WebSphere MQ Version 7.0.1 至 IBM WebSphere MQ Version 7.5 進行移轉](#)
- [檢查需求](#)
- [正在準備系統](#)
- [在 Linux 上安裝 IBM WebSphere MQ 伺服器](#)
- [驗證伺服器安裝](#)
- [在 Linux 上安裝 WebSphere MQ 用戶端](#)
- [驗證用戶端安裝](#)
- [Linux: 在 IBM WebSphere MQ Version 7.5](#)
- [在 Linux 上解除安裝 IBM WebSphere MQ](#)

Solaris 快速入門

本節提供從 Solaris 書籍的舊「快速入門」到新產品說明文件結構的對映:

- [Solaris: 規劃移轉至更新版本](#)
- [檢查需求](#)
- [正在準備系統](#)
- [在 Solaris 上安裝 IBM WebSphere MQ 伺服器](#)
- [驗證伺服器安裝](#)
- [在 Solaris 上安裝 IBM WebSphere MQ 用戶端](#)
- [驗證用戶端安裝](#)
- [Solaris: 在 IBM WebSphere MQ Version 7.5 上套用維護層次更新](#)
- [在 Solaris 上解除安裝 IBM WebSphere MQ](#)

Windows 的快速入門

本節提供從 Windows 書籍的舊「快速入門」到新產品說明文件結構的對映:

- [Windows: 規劃移轉至更新版本](#)
- [檢查需求](#)
- [正在準備系統](#)
- [在 Windows 上安裝 IBM WebSphere MQ 伺服器](#)
- [驗證伺服器安裝](#)
- [在 Windows 系統上安裝 IBM WebSphere MQ 用戶端](#)
- [驗證用戶端安裝](#)
- [Windows: 在 IBM WebSphere MQ Version 7.5](#)
- [在 Solaris 上解除安裝 IBM WebSphere MQ](#)

應用程式設計手冊

本節提供從舊的「應用程式設計手冊」到新產品說明文件結構的對映:

- [應用程式開發概念](#)
- [設計 IBM WebSphere MQ 應用程式](#)
- [撰寫佇列作業應用程式](#)
- [範例 WebSphere MQ 程式](#)
- [C 語言範例](#)
- [COBOL 範例](#)
- [System/390 組譯語言範例](#) [System/390 組譯語言範例](#)
- [WebSphere MQ 資料定義檔](#)
- [64 位元平台上的編碼標準](#)

應用程式設計參考手冊

本節提供從舊應用程式設計參考手冊到新產品說明文件結構的對映:

- [MQI 中使用的資料類型](#)
- [函數呼叫](#)
- [物件的屬性](#)
- [回覆碼](#)
- [MQI 選項的驗證規則](#)
- [報告選項及訊息旗標](#)
- [報告選項及訊息旗標](#)
- [資料轉換](#)
- [指定為 MQRFH2 元素的內容](#)
- [字碼頁轉換](#)

用戶端

本節提供從舊「客戶」書籍到新產品說明文件結構的對映:

- [第 110 頁的『IBM WebSphere MQ MQI 用戶端的概觀』](#)
- [第 112 頁的『IBM WebSphere MQ 用戶端的平台支援』](#)
- [安裝 IBM WebSphere MQ 用戶端](#)
- [配置伺服器與用戶端之間的連線](#)
- [配置延伸交易式用戶端](#)
- [驗證用戶端安裝](#)
- [設定 WebSphere MQ MQI 用戶端安全](#)
- [第 99 頁的『通道』](#)
- [定義 MQI 通道](#)
- [在不同平台上建立伺服器連線和用戶端連線定義](#)
- [在伺服器上建立伺服器連線及用戶端連線定義](#)
- [MQI 通道的通道結束程式](#)
- [將用戶端連接至佇列共用群組](#)
- [使用配置檔來配置用戶端](#)
- [使用 WebSphere MQ 環境變數](#)
- [在用戶端應用程式中使用訊息佇列介面 \(MQI\)](#)

- [建置 WebSphere MQ MQI 用戶端的應用程式](#)
- [在 WebSphere MQ MQI 用戶端環境中執行應用程式](#)
- [準備及執行 CICS 和 Tuxedo 應用程式](#)
- [準備及執行 Microsoft Transaction Server 應用程式](#)
- [準備及執行 WebSphere MQ JMS 應用程式](#)
- [解決 IBM WebSphere MQ MQI 用戶端的問題](#)
- [使用儲存庫中的預先連接結束程式來參照連線定義](#)

常數

本節提供從舊 Constants 書籍到新產品說明文件結構的對映:

- [WebSphere MQ COPY、標頭、併入和模組檔案 WebSphere MQ COPY、標頭、併入和模組檔案](#)
- [常數](#)

交互通訊

本節提供從舊 Intercommunication book 到新產品說明文件結構的對映:

簡介

- [第 44 頁的『交互通訊概念』](#)
- [使用分散式佇列連接應用程式](#)
- [網路及網路規劃](#)
- [WebSphere MQ 分散式傳訊技術](#)
- [分散式佇列管理簡介](#)
- [通道屬性](#)
- [配置資訊範例](#)

WebSphere MQ for Windows 及 UNIX 平台中的分散式佇列管理

- [在 Windows、UNIX and Linux 平台上監視及控制通道](#)
- [建立傳輸佇列](#)
- [觸發通道](#)
- [通道程式](#)
- [遠端傳訊的安全](#)
- [分散式佇列管理的其他考量事項](#)
- [設定 Windows 的通訊](#)
- [配置範例- IBM WebSphere MQ for Windows](#)
- [配置範例- IBM WebSphere MQ for AIX](#)
- [配置範例- IBM WebSphere MQ for HP-UX](#)
- [配置範例- IBM WebSphere MQ for Solaris](#)
- [配置範例- IBM WebSphere MQ for Linux](#)
- [分散式平台的訊息通道規劃範例](#)

進一步的交互通訊考量

- [傳訊通道的通道結束程式](#)
- [通道結束程式呼叫及資料結構](#)
- [佇列名稱解析](#)

訊息與訊息碼

本節提供從舊的「訊息與訊息碼」書籍到新產品說明文件結構的對映:

- [診斷訊息: AMQ4000-9999](#)
- [API 完成及原因碼](#)
- [PCF 原因碼](#)
- [Secure Sockets Layer \(SSL\) 和傳輸層安全 \(TLS\) 回覆碼](#)
- [WCF 自訂通道異常狀況](#)

移轉

Version 7.5 中移轉區段的結構仍與 Version 7.1 中的結構相同。

已新增主題以移轉至 Version 7.5, 並移除主題以移轉至 Version 7.1。若要參照移轉至 Version 7.1 及更早版本, 請參閱 [Where to find a topic for a specific migration path](#)。

監視

本節提供從舊 Monitoring 書籍到新產品說明文件結構的對映:

- [事件監視](#)
- [訊息監視](#)
- [帳戶及統計資料訊息](#)
- [即時監視](#)
- [結構資料類型](#)
- [事件資料的物件屬性](#)

可程式指令格式及管理介面

本節提供從舊的「可程式指令格式及管理介面」書籍到新產品說明文件結構的對映:

- [可程式指令格式簡介](#)
- [WebSphere MQ 管理介面 \(MQAI\) 簡介](#)

發佈/訂閱使用手冊

本節提供從舊的「發佈/訂閱使用手冊」書籍到新產品說明文件結構的對映:

- [WebSphere MQ 發佈/訂閱傳訊簡介](#)
- [分散式發佈/訂閱](#)
- [撰寫發佈/訂閱應用程式](#)
- [發佈/訂閱安全](#)
- [從 Version 6.0 發佈/訂閱移轉](#)
- [WebSphere Event Broker 及 WebSphere Message Broker 中發佈/訂閱分配管理系統的移轉](#)

佇列管理程式叢集

本節提供從舊佇列管理程式叢集書籍到新產品說明文件結構的對映:

- [第 56 頁的『叢集運作方式』](#)
- [配置佇列管理程式叢集](#)
- [管理 IBM WebSphere MQ 叢集](#)
- [將訊息遞送至叢集以及從叢集遞送訊息](#)
- [使用叢集進行工作量管理](#)

- [保持叢集安全](#)
- [使用 MQI 及叢集](#)
- [WebSphere MQ 叢集指令](#)
- [解決佇列管理程式叢集的問題](#)

Script (MQSC) 指令參考手冊

本節提供從舊 Script (MQSC) Command Reference 一書到新產品說明文件結構的對映:

- [Script \(MQSC\) 指令](#)
- [通用值及具有特殊意義的字元](#)
- [建置指令 Script](#)
- [第 103 頁的『IBM WebSphere MQ 物件的命名規則』](#)
- [語法圖](#)
- [MQSC 指令](#)

安全

本節提供從舊「安全」書籍到新產品說明文件結構的對映:

- [安全](#)

系統管理手冊

This section provides a mapping from the old System Administration Guide book to the new product documentation structure:

- [第 37 頁的『IBM WebSphere MQ 技術概觀』](#)
- [管理 IBM WebSphere MQ](#)
- [管理本端 WebSphere MQ 物件](#)
- [使用 IBM WebSphere MQ Explorer 進行管理](#)
- [使用 WebSphere MQ 工作列應用程式 \(僅限 Windows\)](#)
- [WebSphere MQ 控制指令](#)

配置及管理

- [變更 IBM WebSphere MQ 及佇列管理程式配置資訊](#)
- [規劃檔案系統支援](#)
- [在 Windows、UNIX and Linux 系統上設定安全](#)
- [交易式支援](#)
- [使用 WebSphere MQ 無法傳送郵件的佇列處理程式來處理無法遞送的訊息](#)
- [可用性、回復及重新啟動](#)
- [疑難排解及支援](#)
- [WebSphere MQ 及 UNIX System V IPC 資源](#)
- [WebSphere MQ 和 UNIX 處理程序優先順序](#)
- [使用者結束程式、API 結束程式及 WebSphere MQ 可安裝服務](#)

使用 .NET

本節提供從舊的 Using .NET 書籍到新產品說明文件結構的對映:

- [使用 .NET](#)
- [撰寫及部署 WebSphere MQ .NET 程式](#)

- [WebSphere MQ .NET 類別和介面](#)
- [Microsoft Windows Communication Foundation \(WCF\) 的 IBM WebSphere MQ 自訂通道](#)

使用 C++

本節提供從舊的 Using C++ 書籍到新產品說明文件結構的對映:

- [使用 C++](#)
- [WebSphere MQ C++ 類別](#)

使用 Java

本節提供從舊的使用 Java 書籍到新產品說明文件結構的對映:

- [應該使用適用於 Java 的 WebSphere MQ 類別或適用於 JMS 的 WebSphere MQ 類別嗎?](#)
- [使用適用於 Java 的 WebSphere MQ 類別](#)
- [WebSphere MQ classes for JMS](#)
- [使用 WebSphere MQ classes for JMS](#)
- [WebSphere MQ classes for Java](#)

Web 服務

本節提供從舊 Web 服務到新產品說明文件結構的對映:

- [WebSphere MQ 傳輸](#)
- [適用於 HTTP 的 WebSphere MQ 橋接器](#)

使用元件物件模型介面

本節提供從舊的「使用元件物件模型介面」書籍到新產品說明文件結構的對映:

- [使用元件物件模型介面 \(適用於 ActiveX 的 WebSphere MQ 自動化類別\)](#)

IBM WebSphere MQ Version 7.5、IBM i 和 z/OS

IBM WebSphere MQ Version 7.5 不適用於 IBM i 和 z/OS。

這些平台在更新版本的產品中可用。

如需 IBM MQ for IBM i 和 z/OS 最新版本的相關資訊，請參閱 [IBM MQ 網站](#)。

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。

在其他國家或地區中，IBM 可能未提供本文件所包含的產品、服務或特性。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。本文件在提及 IBM 產品、程式或服務時，不表示或暗示只能使用 IBM 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 之產品、程式或服務，使用者必須自行負責作業之評估和驗證責任。

IBM 可能擁有涵蓋本文件所說明主題的專利或專利申請案。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：

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 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。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。

本程式之獲授權者若希望取得本程式之相關資訊，以便能夠：(i) 在獨立建立的程式與其他程式 (包括本程式) 之間交換資訊；以及 (ii) 相互使用已交換的資訊，則請聯絡：

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

該等資訊得依適當條款提供，在某些情況下需付費方得使用。

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的條款，提供本文件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，在其他作業環境中取得的結果可能會顯著不同。部分測量可能已在開發層次系統上進行，但不保證這些測量在一般可用系統上將會相同。此外，有些測量可能是透過推斷來預估。實際結果可能有所不同。本書的使用者應依自己的特定環境，查證適用的資料。

本書所提及之非 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。

一切關於 IBM 未來方針或目的之聲明，隨時可能變更或撤銷，不另行通知，且僅代表目標與目的。

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。所有這些名稱都是虛構的，如有任何類似實際企業所用的名稱及地址之處，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。您可以為了開發、使用、行銷或配送符合範例應用程式所適用的作業平台之應用程式設計介面的應用程式，以任何形式複製、修改及配送這些範例程式，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、服務性或功能。這些範例程式係依其「現狀」而提供，不附任何保證。IBM 不負擔任何因本程式範例之使用而產生的任何損害。

這些範例程式或任何衍生成果的每份複本或任何部分，都必須依照下列方式併入著作權聲明：©（貴客戶之公司名稱）（年）。此程式碼部分衍生自 IBM 公司範例程式。© Copyright IBM Corp. 1993, 2024. All rights reserved.

如果您正在檢視本資訊的電子檔，則照片和彩色圖例可能不會出現。

商標

IBM、IBM 標誌、ibm.com、Passport Advantage、WebSphere、MQSeries 及 z/OS 是 International Business Machines Corp. 的商標或註冊商標。在全球許多司法管轄區註冊。其他產品及服務名稱可能是 IBM 或其他公司的商標。IBM 商標最新清單可於下列網站之「著作權與商標資訊」(Copyright and trademark information) 網頁上取得，網址如下：<https://www.ibm.com/legal/copytrade.shtml>。

Microsoft、Windows、Windows NT 與 Windows 標誌是 Microsoft 公司在美國及/或其他國家或地區的商標。

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。

Adobe、Adobe 標誌、PostScript 與 PostScript 標誌是 Adobe Systems Incorporated 在美國及/或其他國家或地區的註冊商標或商標。

UNIX 是 The Open Group 在美國及/或其他國家/地區的註冊商標。

這裡包含的 Oracle Outside In Technology 受有限使用授權的約束，只能與此應用程式一起使用。

Intel 是 Intel Corporation 或其子公司在美國及其他國家或地區的商標或註冊商標。

註冊商標 Linux 是根據 Linux Foundation (全球標章擁有者 Linus Torvalds 的專用授權) 所提供的子授權使用。

IBM WebSphere MQ 技術概觀

使用 IBM WebSphere MQ 來連接您的應用程式，並管理跨組織的資訊配送。

IBM WebSphere MQ 可讓程式使用一致的應用程式設計介面，透過不同元件 (處理器、作業系統、子系統及通訊協定) 的網路彼此通訊。使用此介面設計及撰寫的應用程式稱為訊息佇列應用程式。

請使用下列子主題，以瞭解 IBM WebSphere MQ 所提供的訊息佇列作業及其他特性。

相關概念

[第 5 頁的『IBM WebSphere MQ 簡介』](#)

您可以使用 IBM WebSphere MQ，讓應用程式在不同時間及許多不同的運算環境中進行通訊。

相關工作

[設計 WebSphere MQ 架構](#)

[WebSphere MQ Managed File Transfer](#)

相關參考

[第 39 頁的『訊息佇列作業的主要特性及好處』](#)

此資訊強調顯示訊息佇列作業的部分特性及好處。它說明訊息佇列作業的安全及資料完整性等特性。

訊息佇列作業簡介

WebSphere MQ 產品可讓程式使用一致的應用程式設計介面，透過不同元件 (處理器、作業系統、子系統及通訊協定) 的網路彼此通訊。

使用此介面設計及撰寫的應用程式稱為 訊息佇列作業 應用程式，因為它們使用 傳訊 及 佇列作業 樣式：

傳訊	程式透過在訊息中傳送彼此的資料來進行通訊，而不是直接彼此呼叫。
佇列作業	訊息放置在儲存體中的佇列上，可讓程式以不同的速度和時間，在不同的位置，彼此之間沒有邏輯連線，彼此獨立執行。

訊息佇列作業已在資料處理中使用多年。目前最常用在電子郵件中。在沒有排隊的情況下，長距離傳送電子訊息需要路徑上的每個節點都可以轉遞訊息，而且收件人會登入並意識到您嘗試傳送訊息給他們的事實。在佇列作業系統中，訊息會儲存在中間節點，直到系統準備好轉遞它們為止。在其最終目的地，它們會儲存在電子郵件信箱中，直到收件人準備好閱讀它們為止。

即便如此，今天仍有許多複雜的商業交易在沒有排隊的情況下處理。在大型網路中，系統可能以現成可用的狀態維護數千條連線。如果系統的某個部分發生問題，則系統的許多部分會變成無法使用。

您可以將訊息佇列作業視為程式的電子郵件。在訊息佇列作業環境中，組成應用程式套組一部分的每一個程式都會執行完整定義且自行包含的函數，以回應特定要求。若要與另一個程式通訊，程式必須將訊息放置在預先定義的佇列上。另一個程式會從佇列中擷取訊息，並處理訊息中包含的要求及資訊。因此訊息佇列作業是一種程式對程式通訊的樣式。

佇列作業是在應用程式準備好處理訊息之前用來保留訊息的機制。佇列作業可讓您：

- 在程式之間進行通訊 (每個程式可能在不同環境中執行)，而不需要撰寫通訊程式碼。
- 選取程式處理訊息的順序。
- 當訊息數超出臨界值時，安排多個程式來處理佇列，以平衡系統上的負載。
- 如果主要系統無法使用，請安排替代系統來處理佇列，以增加應用程式的可用性。

何謂訊息佇列？

訊息佇列 (簡單稱為佇列) 是訊息可以傳送至其中的具名目的地。訊息會累積在佇列上，直到服務那些佇列的程式擷取這些訊息為止。

佇列位於佇列管理程式中，並由佇列管理程式管理 (請參閱 [第 41 頁的『訊息佇列作業術語』](#))。佇列的實體本質取決於執行佇列管理程式的作業系統。佇列可以是電腦記憶體中的暫時緩衝區，或是永久儲存裝置 (例如磁碟) 上的資料集。佇列的實體管理是佇列管理程式的責任，對於參與的應用程式並不明顯。

程式只會透過佇列管理程式的外部服務來存取佇列。他們可以開啟佇列、在佇列上放置訊息、從中取得訊息，以及關閉佇列。他們也可以設定及查詢佇列的屬性。

訊息佇列作業的不同樣式

點對點 (point-to-point)

一個訊息會放置在佇列上，且一個應用程式會接收該訊息。

在點對點傳訊中，傳送端應用程式必須知道接收端應用程式的相關資訊，才能將訊息傳送至該應用程式。例如，傳送端應用程式可能需要知道要將資訊傳送至其中的佇列名稱，也可能指定佇列管理程式名稱。

發佈/訂閱

發佈應用程式發佈的每一個訊息的副本會遞送至每一個感興趣的應用程式。可能有許多、一個或沒有感興趣的應用程式。在發佈/訂閱中，有興趣的應用程式稱為訂閱者，且訊息會排入訂閱所識別的佇列中。

發佈/訂閱傳訊可讓您將資訊提供者與該資訊的消費者取消連結。傳送端應用程式和接收端應用程式不需要彼此瞭解多少，即可傳送和接收資訊。如需發佈/訂閱傳訊的相關資訊，請參閱 [WebSphere 簡介 MQ 發佈/訂閱傳訊](#)

應用程式設計者和開發人員的訊息佇列作業好處

WebSphere MQ 可讓應用程式使用 訊息佇列作業 來參與訊息驅動處理。應用程式可以使用適當的訊息佇列軟體產品，在不同平台之間進行通訊。例如，HP-UX 和 z/OS 應用程式可以透過 WebSphere MQ for HP-UX 和 WebSphere MQ for z/OS 進行通訊。應用程式與基礎通訊的機制隔離。訊息佇列作業的部分其他好處如下：

- 您可以使用可在許多應用程式之間共用的小型程式來設計應用程式。
- 您可以重複使用這些建置區塊來快速建置新的應用程式。
- 寫入以使用訊息佇列作業技術的應用程式，不會受到佇列管理程式運作方式的變更影響。
- 您不需要使用任何通訊協定。佇列管理程式會為您處理通訊的所有層面。
- 接收訊息的程式在傳送訊息時不需要執行。訊息會保留在佇列上。

設計人員可以降低其應用程式的成本，因為開發速度更快，需要的開發人員較少，且對程式設計技能的需求低於不使用訊息佇列作業的應用程式。

WebSphere MQ 會實作一個一般應用程式設計介面，稱為 訊息佇列介面 (或 MQI)，不論應用程式在何處執行。這可讓您更容易將應用程式從一個平台移植到另一個平台。

如需 MQI 的詳細資料，請參閱 [訊息佇列介面概觀](#)。

訊息佇列作業的主要特性及好處

此資訊強調顯示訊息佇列作業的部分特性及好處。它說明訊息佇列作業的安全及資料完整性等特性。

使用訊息佇列作業技術之應用程式的主要特性如下：

- 程式之間沒有直接連線。
- 程式之間的通訊可以獨立於時間。
- 工作可以由小型自行包含的程式來執行。
- 通訊可以由事件驅動。
- 應用程式可以指派訊息的優先順序。
- 保安
- 資料完整性。
- 回復支援。

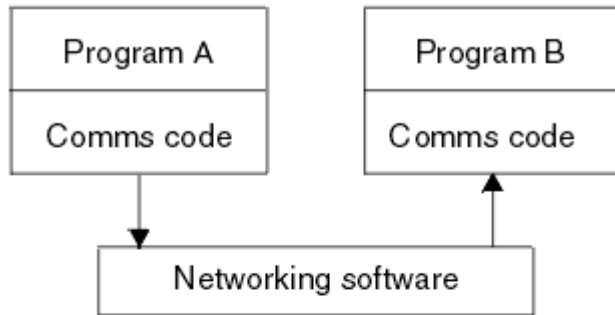
程式之間沒有直接連線

訊息佇列作業是一種間接程式對程式通訊的技術。它可以在程式彼此通訊的任何應用程式內使用。通訊是透過一個程式將訊息放入佇列 (由佇列管理程式所擁有)，以及另一個程式從佇列取得訊息來進行。

程式可以取得由其他程式放入佇列的訊息。其他程式可以連接至與接收端程式相同的佇列管理程式，或另一個佇列管理程式。這個其他佇列管理程式可能位於另一個系統、不同的電腦系統，甚至在不同的商業或企業內。

使用訊息佇列進行通訊的程式之間沒有實體連線。程式會將訊息傳送至佇列管理程式所擁有的佇列，而另一個程式會從佇列中擷取訊息 (請參閱 [第 40 頁的圖 4](#))。

Traditional communication between programs



Communication by message queuing

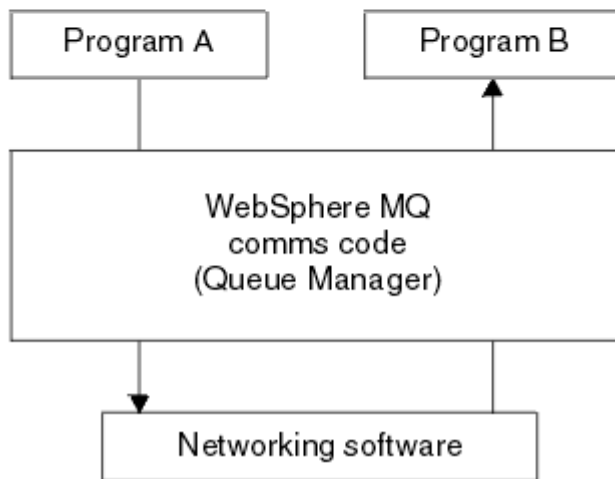


圖 4: 與傳統通訊相比的訊息佇列作業

如同電子郵件，屬於交易一部分的個別訊息會透過儲存及轉寄上的網路傳送。基礎。如果節點之間的鏈結失敗，則會保留訊息，直到還原鏈結或操作員或程式重新導向訊息為止。

在程式中隱藏訊息從佇列移至佇列的機制。因此程式更簡單。

與時間無關的通訊

要求他人執行工作的程式不必等待要求的回覆。他們可以執行其他工作，並在回覆到達時或稍後處理回覆。撰寫傳訊應用程式時，您不需要知道(或關注)程式何時傳送訊息，或目標何時能夠接收訊息。訊息不會遺失；它會由佇列管理程式保留，直到目標準備好處理它為止。訊息會一直留在佇列中，直到程式移除為止。這表示傳送和接收應用程式已取消連結；傳送端可以繼續處理，而不等待接收端確認接收訊息。當傳送訊息時，目標應用程式甚至不必在執行中。它可以在啟動之後擷取訊息。

小型程式

訊息佇列作業可讓您使用小型自行包含程式的優點。您可以將工作分散在數個較小且獨立的程式中，而不是單一大型程式循序執行工作的所有部分。要求程式會傳送訊息至每一個個別程式，要求它們執行其功能；當每一個程式完成時，結果會以一或多個訊息傳回。

訊息驅動處理

當訊息抵達佇列時，它們可以使用觸發來自動啟動應用程式。必要的話，可以在處理訊息時停止應用程式。

事件驅動處理

您可以根據佇列的狀態來控制程式。例如，您可以安排在訊息到達佇列時立即啟動程式，或者您可以指定在佇列上具有高於特定優先順序的 10 則訊息或佇列上具有任何優先順序的 10 則訊息之前不會啟動程式。

訊息優先順序

當程式將訊息放入佇列時，可以指派訊息的優先順序。這會決定在佇列中新增訊息的位置。

程式可以依照訊息在佇列中的順序，或取得特定訊息，從佇列中取得訊息。(如果程式正在尋找先前傳送之要求的回覆，則可能想要取得特定訊息。)

安全

使用 WebSphere MQ 管理者所設定及維護的表格，對每一個資源執行授權檢查。

- 在 WebSphere MQ for z/OS 上使用 Security Server (舊稱為 RACF) 或其他外部安全管理程式。
- 在 WebSphere MQ on UNIX 系統、Linux 系統、Windows 系統及 IBM i 上，提供稱為物件權限管理程式 (OAM) 的安全管理程式作為可安裝服務。依預設，OAM 為作用中。

資料完整性

資料完整性由工作單元提供。在每一個 MQGET 或 MQPUT 上，完全支援工作單元開始及結束的同步化，以容許確定或回復工作單元的結果。同步點支援在內部或外部對 WebSphere MQ 運作，視為應用程式選取的同步點協調形式而定。

回復支援

為了能夠回復，會記載所有持續性 WebSphere MQ 更新項目。如果需要回復，則會還原所有持續訊息，回復所有進行中交易，並以控制同步點管理程式的正常方式處理任何同步點確定及取消。如需持續訊息的相關資訊，請參閱 [訊息持續性](#)。

註: 在考量 WebSphere MQ 用戶端及伺服器時，您不需要變更伺服器應用程式來支援新平台上的其他 WebSphere MQ MQI 用戶端。同樣地，WebSphere MQ MQI 用戶端可以使用其他類型的伺服器來運作，而不會有任何變更。

訊息佇列作業術語

此資訊可讓您深入瞭解訊息佇列作業中使用的部分術語。

其中包括：

- [訊息](#)
- [訊息描述子](#)
- [佇列](#)
- [佇列管理程式](#)
- [通道](#)
- [訊息通道代理程式](#)
- [叢集](#)
- [WebSphere MQ MQI 用戶端](#)
- [點對點 \(point-to-point\)](#)
- [發佈/訂閱 \(Publish/Subscrib\)](#)
- [主題](#)
- [訂閱](#)

訊息

在訊息佇列作業中，訊息是由一個程式所傳送且預期用於另一個程式的資料集合。請參閱 [IBM WebSphere MQ 訊息](#)。如需訊息類型的相關資訊，請參閱 [訊息類型](#)。

訊息描述子

IBM WebSphere MQ 訊息包含控制資訊和應用程式資料。

控制資訊定義在訊息描述子結構 (MQMD) 中，並包含如下內容：

- 訊息類型
- 訊息的 ID
- 遞送訊息的優先順序

應用程式資料的結構及內容由參與程式決定，而不是由 IBM WebSphere MQ 決定。

佇列

可將訊息傳送至其中的具名目的地。訊息會累積在佇列上，直到服務那些佇列的程式擷取這些訊息為止。

佇列管理程式

佇列管理程式是向應用程式提供佇列作業服務的系統程式。

它提供應用程式設計介面，讓程式可以將訊息放置在佇列上，以及從佇列取得訊息。佇列管理程式提供其他功能，讓管理者可以建立新的佇列、變更現有佇列的內容，以及控制佇列管理程式的作業。

若要在系統上提供 IBM WebSphere MQ 訊息佇列作業服務，必須有佇列管理程式在執行中。您可以在單一系統上執行多個佇列管理程式 (例如，將測試系統與即時系統分開)。對於應用程式，每一個佇列管理程式都由連線控點 (*Hconn*) 來識別。

許多不同的應用程式可以同時使用佇列管理程式的服務，這些應用程式可以完全不相關。若要讓程式使用佇列管理程式的服務，它必須建立與該佇列管理程式的連線。

若要讓應用程式將訊息傳送至連接至其他佇列管理程式的應用程式，佇列管理程式必須能夠彼此通訊。IBM WebSphere MQ 實作儲存及轉遞通訊協定，以確保在這類應用程式之間安全遞送訊息。

通道

通道是提供從一個佇列管理程式到另一個佇列管理程式之通訊路徑的物件。在分散式佇列中使用通道，以將訊息從一個佇列管理程式移至另一個佇列管理程式，而且它們會保護應用程式免受基礎通訊協定的影響。佇列管理程式可能存在於相同或不同的平台上。

訊息通道代理程式

訊息通道代理程式會將訊息從一個佇列管理程式移至另一個佇列管理程式。

在處理報告訊息時，會對它們進行參照，您在設計應用程式時需要考量它們。如需相關資訊，請參閱 [撰寫您自己的訊息通道代理程式](#)。

叢集

叢集是邏輯上以某種方式關聯的佇列管理程式網路。在 IBM WebSphere MQ Version 7.0 以及更新版本中，佇列管理程式可以使用叢集作業。

在使用不含叢集作業的分散式佇列的 IBM WebSphere MQ 網路中，每個佇列管理程式都是獨立的。如果一個佇列管理程式需要將訊息傳送至另一個佇列管理程式，它必須已定義傳輸佇列及遠端佇列管理程式的通道。

使用叢集有兩個不同的原因：減少系統管理，以及改善可用性和工作量平衡。

只要您建立甚至最小的叢集，就可以從簡化的系統管理中獲益。屬於叢集的佇列管理程式需要較少的定義，因此會減少在定義中產生錯誤的風險。

如需叢集作業的相關資訊，請參閱 [第 56 頁的『叢集運作方式』](#)。

IBM WebSphere MQ MQI client

IBM WebSphere MQ MQI 用戶端是 IBM WebSphere MQ 的可獨立安裝元件。MQI 用戶端可讓您以通訊協定執行 IBM WebSphere MQ 應用程式，與其他平台的一個以上「訊息佇列介面 (MQI)」伺服器互動，並連接至其佇列管理程式。

如需如何安裝及使用 IBM WebSphere MQ MQI client 元件的完整詳細資料，請參閱 [安裝 IBM WebSphere MQ MQI client 及配置伺服器與用戶端之間的連線](#)。

點對點傳訊

在點對點傳訊中，每一個訊息會從一個生產端應用程式傳送至一個消費端應用程式。訊息會透過生產端應用程式傳送，並將訊息放到佇列中，而消費端應用程式會從該佇列中取得訊息。

發佈/訂閱傳訊

在發佈/訂閱傳訊中，發佈應用程式所發佈的每一個訊息的副本會遞送給每一個感興趣的應用程式。可能有許多、一個或沒有感興趣的應用程式。在發佈/訂閱中，有興趣的應用程式稱為訂閱者，且訊息會排入訂閱所識別的佇列中。如需發佈/訂閱的相關資訊，請參閱 [IBM WebSphere MQ 發佈/訂閱傳訊簡介](#)。

主題

主題是說明發佈/訂閱訊息中已發佈之資訊主旨的字串。

主題是在發佈/訂閱系統中順利遞送訊息的關鍵。發佈者不在每一個訊息中包含特定的目的地位址，而是指派主題給每一個訊息。佇列管理程式使主題符合已訂閱至該主題的訂閱者清單，並將訊息遞送至其中每一個訂閱者。

訂閱

發佈/訂閱應用程式可以在特定主題的相關訊息中登錄興趣。當應用程式執行此動作時，稱為訂閱者，且術語訂閱定義如何將相符訊息排入佇列以進行處理。

訂閱包含訂閱者身分及要放置發佈資訊之目的佇列身分的相關資訊。它也包含如何將發佈資訊放置在目的地佇列上的相關資訊。

訊息和佇列

訊息和佇列是訊息佇列系統的基本元件。

何謂訊息？

訊息是對使用它的應用程式有意義的位元組字串。訊息是用來將資訊從一個應用程式傳送至另一個應用程式(或在相同應用程式的不同部分之間)。應用程式可以在相同平台上執行，也可以在不同平台上執行。

IBM WebSphere MQ 訊息有兩個部分：

- 應用程式資料。應用程式資料的內容和結構由使用它的應用程式定義。
- 訊息描述子。訊息描述子會識別訊息並包含其他控制資訊，例如訊息類型及傳送應用程式指派給訊息的優先順序。

訊息描述子的格式由 IBM WebSphere MQ 定義。如需訊息描述子的完整說明，請參閱 [MQMD-訊息描述子](#)。

訊息長度

預設訊息長度上限是 4 MB，不過您可以將它增加至長度上限 100 MB (其中 1 MB 等於 1 048 576 個位元組)。實際上，訊息長度可能受到下列限制：

- 定義給接收佇列的訊息長度上限
- 定義給佇列管理程式的訊息長度上限
- 佇列定義的訊息長度上限
- 傳送或接收應用程式所定義的訊息長度上限
- 訊息可用的儲存體數量

可能需要數則訊息才能傳送應用程式所需的所有資訊。

應用程式如何傳送及接收訊息？

應用程式會使用 **MQI 呼叫** 來傳送及接收訊息。

例如，若要將訊息放入佇列，應用程式：

1. 發出 MQI MQOPEN 呼叫來開啟必要佇列
2. 發出 MQI MQPUT 呼叫，將訊息放入佇列中

另一個應用程式可以透過發出 MQI MQGET 呼叫，從相同佇列擷取訊息。

如需 MQI 呼叫的相關資訊，請參閱 [MQI 呼叫](#)。

何謂佇列？

佇列是用來儲存訊息的資料結構。

每一個佇列都是由佇列管理程式所擁有。佇列管理程式負責維護所擁有的佇列，並將其接收的所有訊息儲存至適當的佇列中。應用程式或佇列管理程式可能會將訊息放置在佇列中，作為其正常作業的一部分。

預先定義的佇列及動態佇列

佇列可以用其建立方式來描述：

- **預先定義佇列** 由管理者使用適當的 MQSC 或 PCF 指令建立。預先定義的佇列是永久的；它們的存在與使用它們的應用程式無關，並且在 IBM WebSphere MQ 重新啟動之後仍然存在。
- 當應用程式建立時，會建立 **動態佇列** 發出 MQOPEN 要求，並指定模型佇列的名稱。所建立的佇列是根據範本佇列定義（稱為模型佇列）。您可以使用 MQSC 指令 DEFINE QMODEL 來建立模型佇列。模型佇列的屬性（例如，可以儲存在它上的訊息數上限）由從它建立的任何動態佇列繼承。

模型佇列具有一個屬性，可指定動態佇列是永久還是暫時。永久佇列會在應用程式及佇列管理程式重新啟動後繼續存在；暫時佇列會在重新啟動時遺失。

從佇列擷取訊息

適當授權的應用程式可以根據下列擷取演算法，從佇列中擷取訊息：

- 先進先出 (FIFO)。
- 訊息優先順序，如訊息描述子中所定義。以 FIFO 為基準擷取具有相同優先順序的訊息。
- 特定訊息的程式要求。

來自應用程式的 MQGET 要求會決定使用的方法。

交互通訊概念

在 WebSphere MQ 中，交互通訊表示將訊息從一個佇列管理程式傳送至另一個佇列管理程式。接收端佇列管理程式可以位於相同或另一部機器上；附近或世界另一端。它可以在與本端佇列管理程式相同的平台上執行，也可以在 WebSphere MQ 支援的任何平台上執行。這稱為分散式環境。WebSphere MQ 會使用「分散式佇列管理 (DQM)」來處理分散式環境中的通訊。

本端佇列管理程式有時稱為來源佇列管理程式，遠端佇列管理程式有時稱為目標佇列管理程式或友機佇列管理程式。

分散式佇列如何運作？

「分散式佇列作業」可讓您將訊息從一個「佇列管理程式」傳送至另一個「佇列管理程式」。接收端佇列管理程式可能位於相同機器或遠端機器上。「佇列管理程式」、「佇列」、「通道」及相關聯的「定義」會與「叢集作業」（邏輯上關聯的「佇列管理程式」網路）一起概述。

第 45 頁的圖 5 顯示分散式佇列的元件概觀。

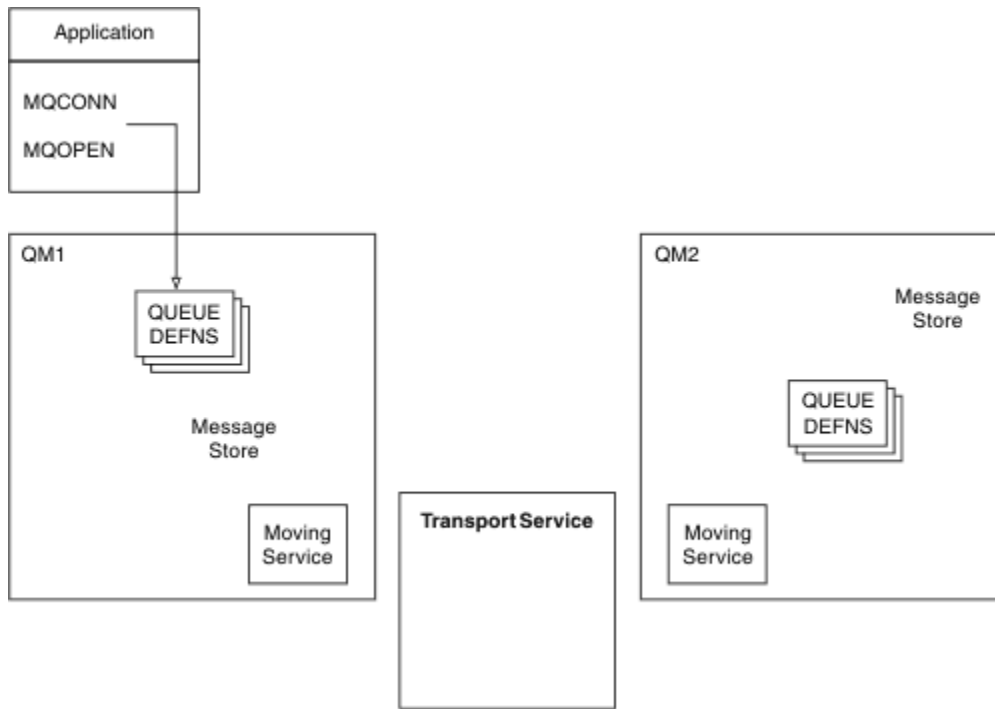


圖 5: 分散式佇列的元件概觀

1. 應用程式使用 MQCONN 呼叫來連接至佇列管理程式。
2. 然後，應用程式會使用 MQOPEN 呼叫來開啟佇列，以便它可以在其中放置訊息。
3. 佇列管理程式具有每一個佇列的定義，並指定佇列上容許的訊息數上限之類的資訊。它也可以具有位於遠端佇列管理程式上的佇列定義。
4. 如果訊息以遠端系統上的佇列為目的地，則本端佇列管理程式會將它們保留在訊息儲存庫中，直到它準備好將它們轉遞至遠端佇列管理程式為止。這對應用程式沒有影響。
5. 每一個佇列管理程式都包含稱為 移動服務 元件的通訊軟體；透過此，佇列管理程式可以與其他佇列管理程式進行通訊。
6. 傳輸服務 與佇列管理程式無關，可以是下列其中一項 (視平台而定):
 - 系統網路架構進階程式對程式通訊 (SNA APPC)
 - 傳輸控制通訊協定/網際網路通訊協定 (Transmission Control Protocol/Internet Protocol, (TCP/IP))
 - 網路基本輸入/輸出系統 (NetBIOS)
 - 循序封包交換 (SPX)

分散式佇列的元件有哪些?

WebSphere MQ 應用程式可以將訊息放入本端佇列，亦即，應用程式所連接之佇列管理程式上的佇列。

佇列管理程式具有其每一個佇列的定義。它也可以具有其他佇列管理程式所擁有的佇列定義。這些稱為 遠端佇列定義。WebSphere MQ 應用程式也可以放置以這些遠端佇列為目標的訊息。

如果訊息以遠端佇列管理程式為目的地，則本端佇列管理程式會將訊息儲存在 傳輸佇列 中，直到備妥將訊息傳送至遠端佇列管理程式為止。傳輸佇列是一種特殊類型的本端佇列，訊息會儲存在其中，直到能夠順利傳輸並儲存在遠端佇列管理程式為止。

處理訊息傳送及接收的軟體稱為 訊息通道代理程式 (MCA)。

在 通道上的佇列管理程式之間傳輸訊息。通道是兩個佇列管理程式之間的單向通訊鏈結。它可以在遠端佇列管理程式中傳送指定給任意數目佇列的訊息。

傳送訊息所需的元件

如果要將訊息傳送至遠端佇列管理程式，則本端佇列管理程式需要傳輸佇列及通道的定義。

通道的每一端都有個別的定義，例如，將它定義為傳送端或接收端。簡式通道包含本端佇列管理程式中的傳送端通道定義，以及遠端佇列管理程式中的接收端通道定義。這兩個定義必須具有相同的名稱，且一起構成一個通道。

在通道的每一端也有一個訊息通道代理程式 (MCA)。

每一個佇列管理程式都應該有一個無法傳送郵件的佇列 (也稱為未遞送訊息佇列)。如果訊息無法遞送至其目的地，則會將訊息放置在此佇列上。

第 46 頁的圖 6 顯示佇列管理程式、傳輸佇列、通道及 MCA 之間的關係。

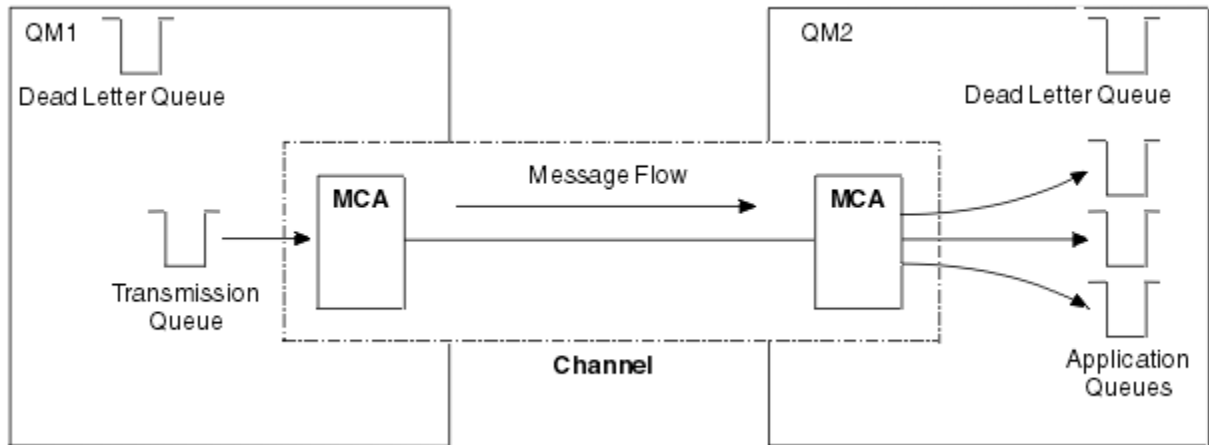


圖 6: 傳送訊息

傳回訊息所需的元件

如果您的應用程式需要從遠端佇列管理程式傳回訊息，您需要定義另一個通道，以便在佇列管理程式之間以相反方向執行，如第 46 頁的圖 7 所示。

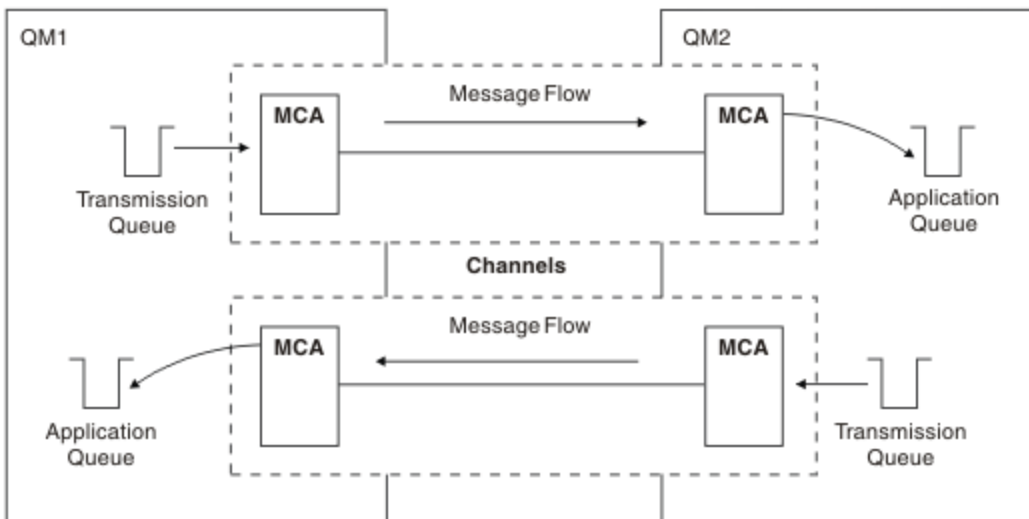


圖 7: 雙向傳送訊息

如需「分散式佇列管理」的相關資訊，請參閱 [分散式佇列管理簡介](#)。

叢集元件

傳統 WebSphere MQ 網路 (透過手動定義通道來交互連接) 的另一種替代方案是使用叢集。

叢集是以某種方式邏輯關聯的佇列管理程式網路。您可以將叢集中的佇列管理程式分組，讓佇列管理程式可以讓它們所管理的佇列可供叢集中的所有其他佇列管理程式使用。假設您已備妥必要的網路基礎架構，任何佇列管理程式都可以將訊息傳送至相同叢集中的任何其他佇列管理程式，而不需要每個目的地的明確通道定義、遠端佇列定義或傳輸佇列。叢集中的每個佇列管理程式都有單一傳輸佇列，可將訊息傳輸至叢集中的任何其他佇列管理程式。每一個佇列管理程式只需要定義一個叢集接收端通道和一個叢集傳送端通道，任何其他通道都會由叢集自動管理。

第 47 頁的圖 8 顯示稱為 CLUSTER 之叢集的元件：

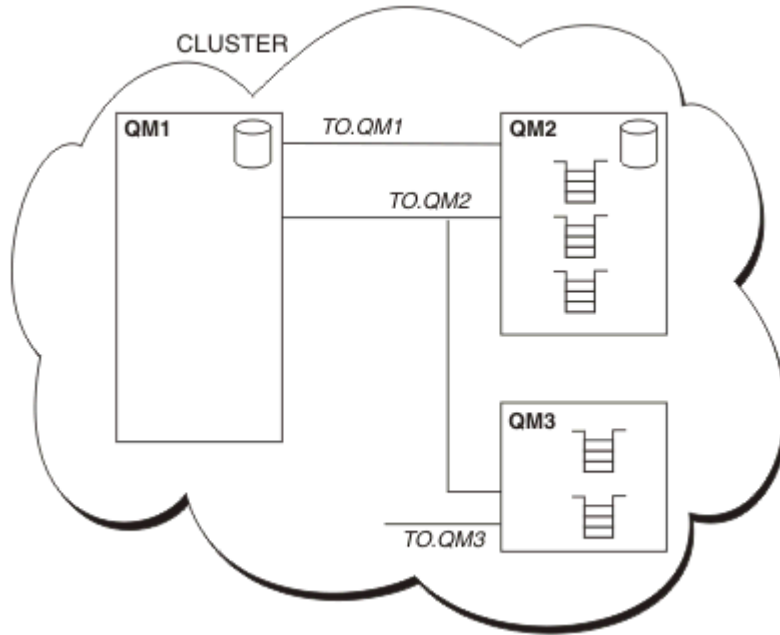


圖 8: 佇列管理程式的叢集

- CLUSTER 包含三個佇列管理程式: QM1、QM2 及 QM3。
- QM1 及 QM2 會管理叢集中佇列管理程式及佇列相關資訊的完整儲存庫。
- QM2 及 QM3 會管理部分叢集佇列，即叢集中任何其他佇列管理程式可存取的佇列。
- 每一個佇列管理程式都有一個稱為 TO.qmgr 的叢集接收端通道，它可以在該通道上接收訊息。
- 每一個佇列管理程式也都有一個叢集傳送端通道，可將資訊傳送至其中一個儲存庫佇列管理程式。
- QM1 和 QM3 會傳送至 QM2 的儲存庫，而 QM2 會傳送至 QM1 的儲存庫。

如同分散式佇列，您可以使用 MQPUT 呼叫，將訊息放入任何佇列管理程式的佇列中。您可以使用 MQGET 呼叫，從本端佇列擷取訊息。

如需叢集的相關資訊，請參閱第 33 頁的『佇列管理程式叢集』。

相關概念

第 48 頁的『分散式佇列元件』

這些是您啟用交互通訊所需的物件。

第 50 頁的『無法傳送郵件的佇列』

無法傳送郵件的佇列 (或無法遞送的訊息佇列) 是訊息無法遞送至正確目的地時傳送至其中的佇列。

第 51 頁的『遠端佇列定義』

遠端佇列定義是另一個佇列管理程式所擁有之佇列的定義。

第 51 頁的『如何取得遠端佇列管理程式』

您可能不會一律在每一個來源與目標佇列管理程式之間有一個通道。兩者之間還有許多其他連結方式，包括多跳、共用通道、使用不同通道及叢集。

[第 53 頁的『定址資訊』](#)

當應用程式放置以遠端佇列管理程式為目的地的訊息時，本端佇列管理程式會將傳輸標頭新增至它們，然後再將它們放置在傳輸佇列上。此標頭包含目的地佇列及佇列管理程式的名稱，即定址資訊。

[第 53 頁的『何謂別名?』](#)

別名用來提供訊息的服務品質。佇列管理程式別名可讓系統管理者變更目標佇列管理程式的名稱，而不會導致您必須變更應用程式。它還可讓系統管理者變更目的地佇列管理程式的路徑，或設定涉及通過許多其他佇列管理程式(多跳)的路徑。回覆目的地佇列別名提供回覆的服務品質。

[第 53 頁的『佇列管理程式別名定義』](#)

當應用程式開啟佇列以放置訊息、指定佇列名稱及佇列管理程式名稱時，會套用佇列管理程式別名定義。

[第 54 頁的『回覆目的地佇列別名定義』](#)

回覆目的地佇列別名定義會在訊息描述子中指定回覆資訊的替代名稱。其優點是您可以變更佇列或佇列管理程式的名稱，而不需要變更應用程式。

分散式佇列元件

這些是您啟用交互通訊所需的物件。

分散式佇列的元件如下：

- 訊息通道
- 訊息通道代理程式
- 傳輸佇列
- 通道起始程式及接聽器
- 通道結束程式

訊息通道是將訊息從一個佇列管理程式傳送至另一個佇列管理程式的通道。請勿將訊息通道與 MQI 通道混淆。MQI 通道有兩種類型：伺服器連線 (SVRCONN) 和用戶端連線 (CLNTCONN)。如需相關資訊，請參閱 [MQI 通道](#)。

訊息通道每一端的定義可以是下列其中一種類型：

- 傳送端 (SDR)
- 接收端 (RCVR)
- 伺服器 (SVR)
- 要求者 (RQSTR)
- 叢集傳送端 (CLUSDR)
- 叢集接收端 (CLUSRCVR)

訊息通道是使用定義在一端的其中一種類型，以及定義在另一端的相容類型來定義。可能的組合為：

- 傳送端-接收端
- 要求端-伺服器
- 要求端-傳送端 (回呼)
- 伺服器-接收端
- 叢集傳送端-叢集接收端

定義通道中包含建立傳送端-接收端通道的詳細指示(不適用於 z/OS)。如需設定傳送端-接收端通道所需的參數範例，請參閱適用於您平台的 [配置資訊範例](#)。如需定義任何類型通道所需的參數，請參閱 [DEFINE CHANNEL](#)。

傳送端-接收端通道

一個系統中的傳送端會啟動通道，以便它可以將訊息傳送至另一個系統。傳送端會要求通道另一端的接收端啟動。傳送端會將訊息從其傳輸佇列傳送至接收端。接收端將訊息放置在目的地佇列上。[第 49 頁的圖 9](#) 說明這一點。

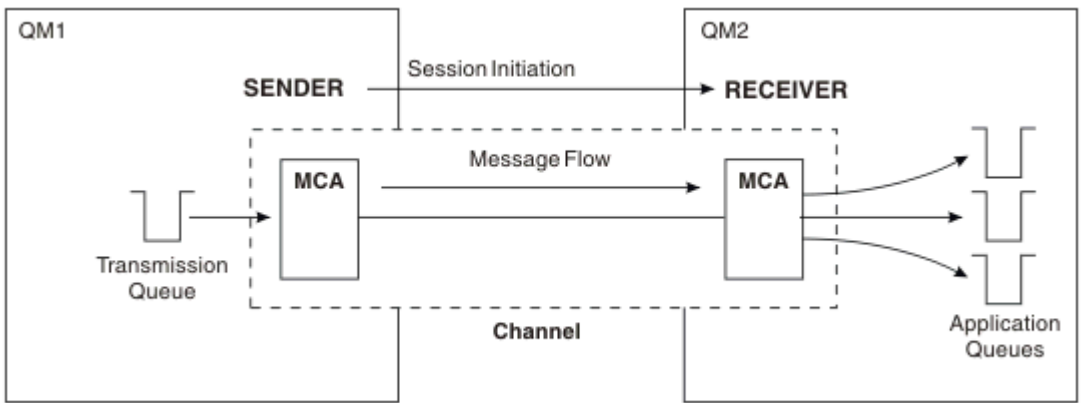


圖 9: 傳送端-接收端通道

要求端-伺服器通道

一個系統中的要求端會啟動通道，以便它可以從另一個系統接收訊息。要求端要求通道另一端的伺服器啟動。伺服器會從其通道定義中所定義的傳輸佇列，將訊息傳送給要求端。

伺服器通道也可以起始通訊，並將訊息傳送給要求端。這僅適用於完整伺服器，即具有通道定義中所指定友機連線名稱的伺服器通道。完整伺服器可以由要求者啟動，也可以起始與要求者的通訊。

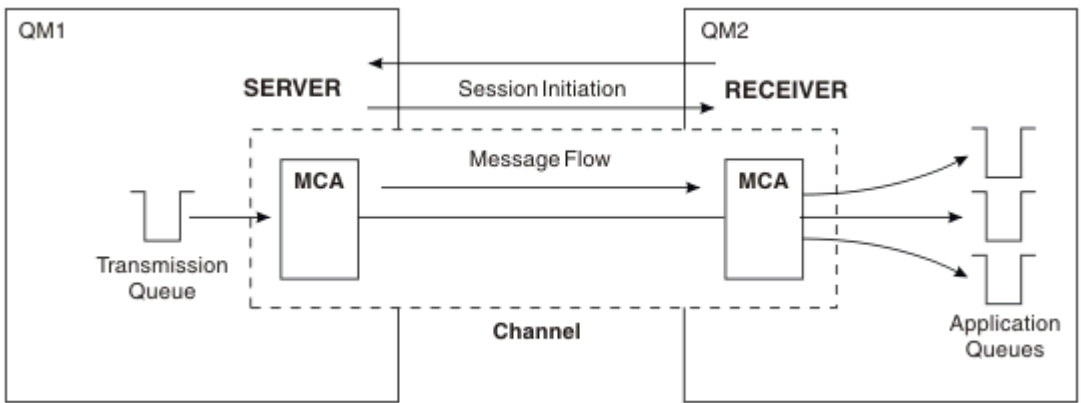


圖 10: 要求端-伺服器通道

要求端 - 傳送端通道

要求端會啟動通道，且傳送端會終止呼叫。然後，傳送端會根據其通道定義中的資訊 (稱為 回呼) 來重新啟動通訊。它會將訊息從傳輸佇列傳送至要求端。

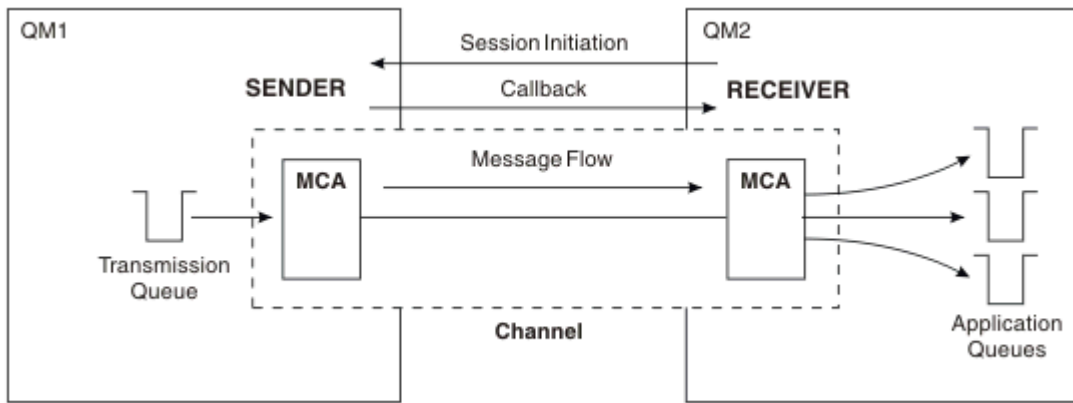


圖 11: 要求端-傳送端通道

伺服器-接收端通道

這類似於傳送端-接收端，但僅適用於完整伺服器，即具有通道定義中所指定友機連線名稱的伺服器通道。通道啟動必須在鏈結的伺服器端起始。此圖的圖解類似於第 49 頁的圖 9 中的圖解。

叢集傳送端通道

在叢集中，每一個佇列管理程式都有一個叢集傳送端通道，可將叢集資訊傳送至其中一個完整儲存庫佇列管理程式。佇列管理程式也可以將訊息傳送至叢集傳送端通道上的其他佇列管理程式。

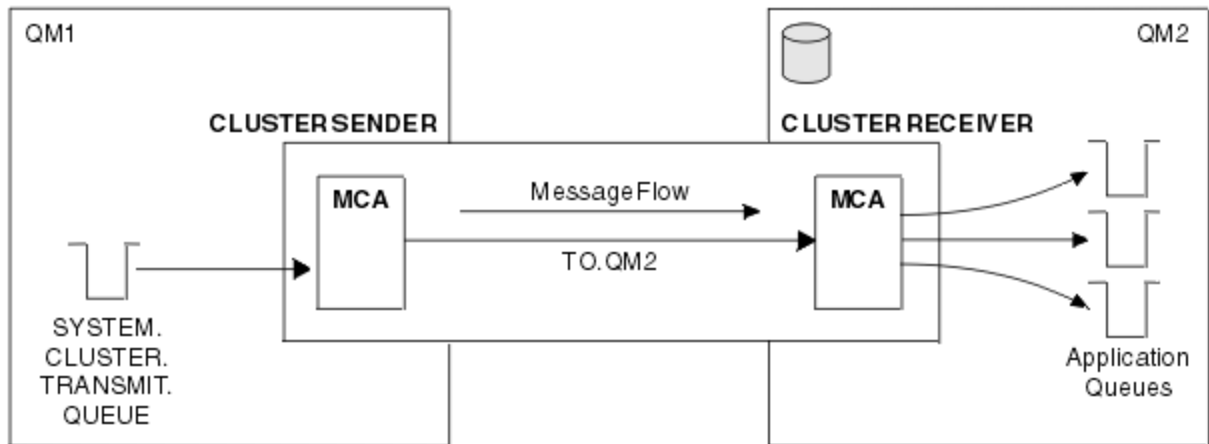


圖 12: 叢集傳送端通道

叢集接收端通道

在叢集中，每一個佇列管理程式都有一個叢集接收端通道，它可以在其中接收訊息及叢集的相關資訊。此圖的圖解類似於第 50 頁的圖 12 中的圖解。

無法傳送郵件的佇列

無法傳送郵件的佇列 (或無法遞送的訊息佇列) 是訊息無法遞送至正確目的地時傳送至其中的佇列。

當訊息無法放置在目的地佇列時，會將訊息放置在此佇列上。例如，因為佇列不存在，或因為佇列已滿。通道傳送端也會使用無法傳送郵件的佇列來處理資料轉換錯誤。

請考量為每一個佇列管理程式定義無法傳送郵件的佇列。如果您沒有，且 MCA 無法放置訊息，則它會留在傳輸佇列中，且通道會停止。

此外，如果無法遞送快速、非持續訊息 (請參閱 [快速、非持續訊息](#))，且目標系統上沒有無法傳送郵件的佇列，則會捨棄這些訊息。

不過，使用無法傳送郵件的佇列可能會影響訊息遞送的順序，因此您可以選擇不使用它們。

您可以使用 USEDLQ 通道屬性來判定無法遞送訊息時是否使用無法傳送郵件的佇列。此屬性可以配置為佇列管理程式的部分功能使用無法傳送郵件的佇列，而其他功能則不使用。如需在不同 MQSC 指令上使用 USEDLQ 通道屬性的相關資訊，請參閱 [DEFINE CHANNEL](#)、[DISPLAY CHANNEL](#)、[ALTER CHANNEL](#) 及 [DISPLAY CLUSQMGR](#)。

遠端佇列定義

遠端佇列定義是另一個佇列管理程式所擁有之佇列的定義。

雖然應用程式只能從本端佇列擷取訊息，但可以將訊息放置在本端佇列或遠端佇列上。因此，除了其每一個本端佇列的定義外，佇列管理程式也可以具有遠端佇列定義。遠端佇列定義的優點是它們可讓應用程式將訊息放入遠端佇列，而不需要指定遠端佇列或遠端佇列管理程式的名稱，或傳輸佇列的名稱。遠端佇列定義提供您位置獨立性。

遠端佇列定義還有其他用途，稍後會說明這些用途。

如何取得遠端佇列管理程式

您可能不會一律在每一個來源與目標佇列管理程式之間有一個通道。兩者之間還有許多其他連結方式，包括多跳、共用通道、使用不同通道及叢集。

多跳

如果來源佇列管理程式與目標佇列管理程式之間沒有直接通訊鏈結，則在前往目標佇列管理程式的途中，可以通過一或多個中繼佇列管理程式。這稱為多重中繼站。

您需要定義所有佇列管理程式之間的通道，以及中間佇列管理程式上的傳輸佇列。這顯示在 [第 51 頁的圖 13](#) 中。

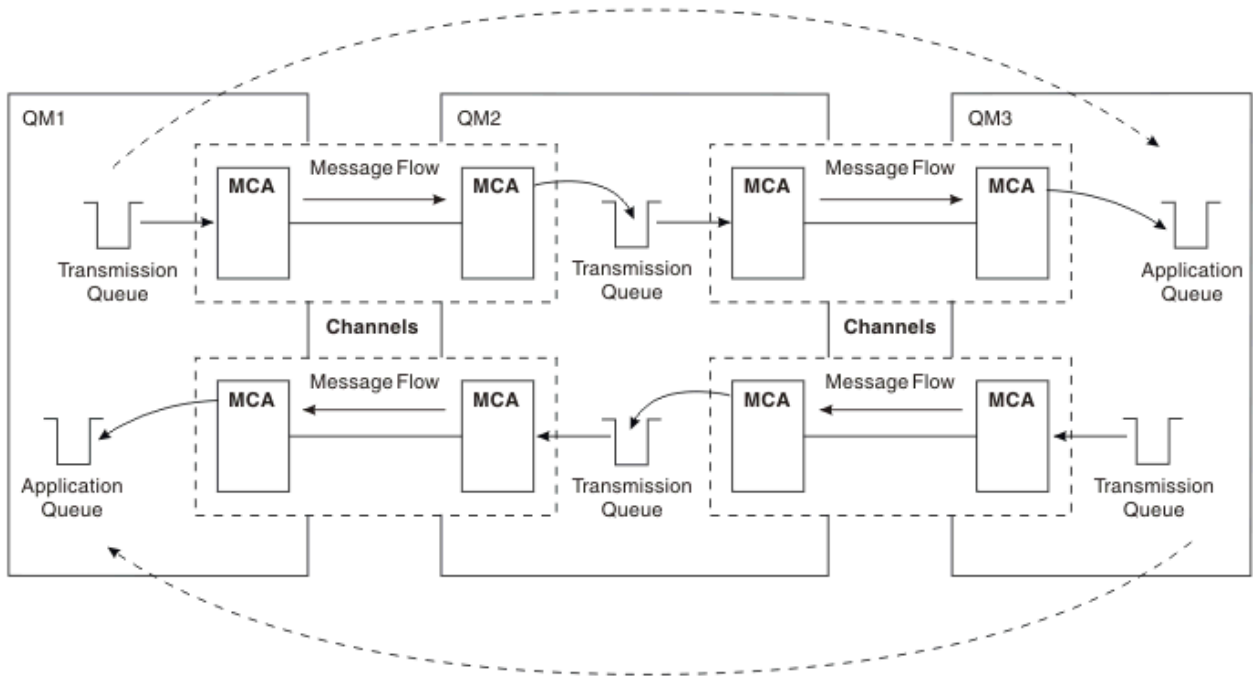


圖 13: 通過中繼佇列管理程式

共用通道

作為應用程式設計程式，您可以選擇強制應用程式指定遠端佇列管理程式名稱及佇列名稱，或為每一個遠端佇列建立遠端佇列定義。此定義會保留遠端佇列管理程式名稱、佇列名稱及傳輸佇列的名稱。無論如何，所有來自相同遠端位置之所有應用程式定址佇列的所有訊息，都會透過相同的傳輸佇列傳送其訊息。這顯示在第 52 頁的圖 14 中。

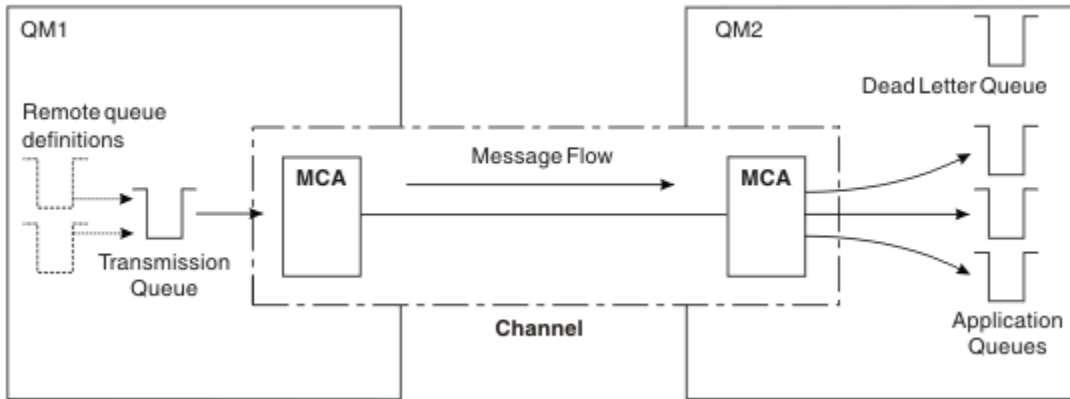


圖 14: 共用傳輸佇列

第 52 頁的圖 14 說明從多個應用程式到多個遠端佇列的訊息可以使用相同的通道。

使用不同的通道

如果您有不同類型的訊息要在兩個佇列管理程式之間傳送，則可以在兩個佇列管理程式之間定義多個通道。有時您需要替代通道 (可能為了安全起見)，或與龐大的訊息資料流量交換遞送速度。

若要設定第二個通道，您需要定義另一個通道及另一個傳輸佇列，並建立遠端佇列定義，以指定位置及傳輸佇列名稱。然後您的應用程式可以使用任一通道，但訊息仍會遞送至相同的目標佇列。這會顯示在第 52 頁的圖 15 中。

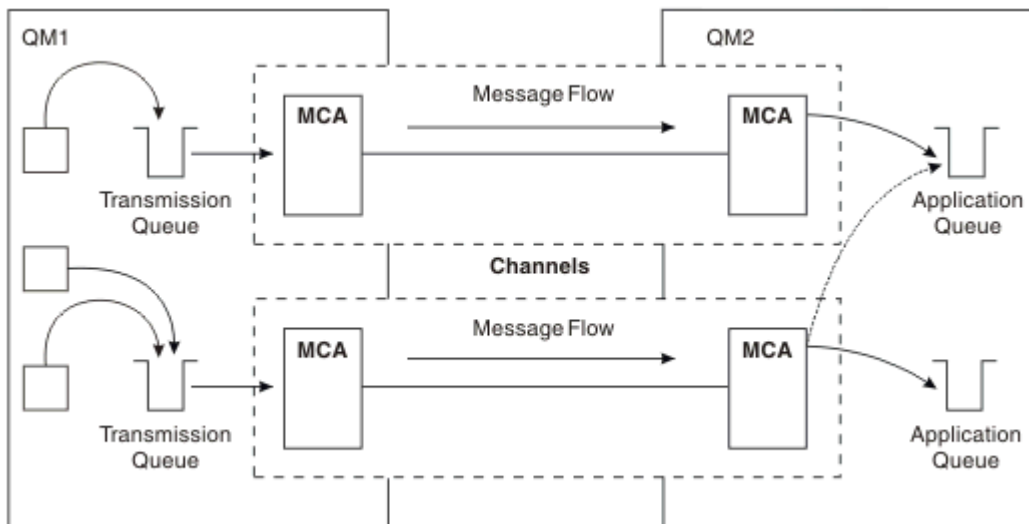


圖 15: 使用多個通道

當您使用遠端佇列定義來指定傳輸佇列時，您的應用程式 **不能** 自行指定位置 (即目的地佇列管理程式)。如果有的話，佇列管理程式不會使用遠端佇列定義。遠端佇列定義提供您位置獨立性。應用程式可以在不知道佇列位置的情況下將訊息放置到邏輯佇列中，並且您可以變更實體佇列，而不需要變更應用程式。

使用叢集作業

叢集內的每個佇列管理程式都會定義叢集接收端通道。當另一個佇列管理程式想要傳送訊息至該佇列管理程式時，它會自動定義對應的叢集傳送端通道。例如，如果叢集中有一個佇列的多個實例，則叢集傳送端通道可以定義給管理該佇列的任何佇列管理程式。WebSphere MQ 使用工作量管理演算法，該演算法使用循環式常式來選取可用的佇列管理程式，以將訊息遞送至該佇列管理程式。如需相關資訊，請參閱 [第 107 頁的『叢集』](#)。

定址資訊

當應用程式放置以遠端佇列管理程式為目的地的訊息時，本端佇列管理程式會將傳輸標頭新增至它們，然後再將它們放置在傳輸佇列上。此標頭包含目的地佇列及佇列管理程式的名稱，即定址資訊。

在單一佇列管理程式環境中，當應用程式開啟佇列以放置訊息時，會建立目的地佇列的位址。因為目的地佇列位於相同的佇列管理程式上，所以不需要任何定址資訊。

在分散式環境中，佇列管理程式不僅需要知道目的地佇列名稱，還需要知道該佇列的位置 (即佇列管理程式名稱)，以及該遠端位置 (即傳輸佇列) 的路徑。此定址資訊包含在傳輸標頭中。接收通道會移除傳輸標頭，並使用其中的資訊來尋找目的地佇列。

如果您使用遠端佇列定義，則可以避免應用程式需要指定目的地佇列管理程式的名稱。此定義指定遠端佇列的名稱、訊息目的地的遠端佇列管理程式名稱，以及用來傳輸訊息的傳輸佇列名稱。

何謂別名？

別名用來提供訊息的服務品質。佇列管理程式別名可讓系統管理者變更目標佇列管理程式的名稱，而不會導致您必須變更應用程式。它還可讓系統管理者變更目的地佇列管理程式的路徑，或設定涉及通過許多其他佇列管理程式 (多跳) 的路徑。回覆目的地佇列別名提供回覆的服務品質。

佇列管理程式別名及回覆目的地佇列別名是使用具有空白 RNAME 的遠端佇列定義來建立。這些定義不會定義實際佇列；佇列管理程式會使用它們來解析實體佇列名稱、佇列管理程式名稱及傳輸佇列。

別名定義的特徵是具有空白 RNAME。

佇列名稱解析

每次開啟佇列時，每個佇列管理程式都會進行佇列名稱解析。其目的是識別目標佇列、目標佇列管理程式 (可能是本端)，以及該佇列管理程式的路徑 (可能是空值)。已解析的名稱有三個部分：佇列管理程式名稱、佇列名稱，以及傳輸佇列 (如果佇列管理程式在遠端)。

當遠端佇列定義存在時，不會參照別名定義。應用程式提供的佇列名稱會解析為遠端佇列定義中指定的目的地佇列、遠端佇列管理程式及傳輸佇列的名稱。如需佇列名稱解析的詳細資訊，請參閱 [佇列名稱解析](#)。

如果沒有遠端佇列定義，且已指定佇列管理程式名稱，或由名稱服務解析，則佇列管理程式會查看是否有佇列管理程式別名定義符合所提供的佇列管理程式名稱。如果有，則會使用其中的資訊將佇列管理程式名稱解析為目的地佇列管理程式的名稱。佇列管理程式別名定義也可以用來決定傳送至目的地佇列管理程式的傳輸佇列。

如果解析的佇列名稱不是本端佇列，則佇列管理程式名稱及佇列名稱都會併入應用程式放置至傳輸佇列之每一則訊息的傳輸標頭中。

除非遠端佇列定義或佇列管理程式別名定義變更，否則所使用的傳輸佇列通常與已解析的佇列管理程式同名。如果您尚未定義此類傳輸佇列，但已定義預設傳輸佇列，則會使用此傳輸佇列。

在 z/OS 上執行的佇列管理程式名稱限制為四個字元。

佇列管理程式別名定義

當應用程式開啟佇列以放置訊息、指定佇列名稱及佇列管理程式名稱時，會套用佇列管理程式別名定義。

佇列管理程式別名定義有三種用法：

- 傳送訊息時，重新對映佇列管理程式名稱
- 傳送訊息時，變更或指定傳輸佇列

- 接收訊息時，判斷本端佇列管理程式是否為那些訊息的預期目的地

出埠訊息-重新對映佇列管理程式名稱

佇列管理程式別名定義可用來重新對映 MQOPEN 呼叫中指定的佇列管理程式名稱。例如，MQOPEN 呼叫指定佇列名稱 THISQ 和佇列管理程式名稱 YOURQM。在本端佇列管理程式中，有一個類似下列範例的佇列管理程式別名定義：

```
DEFINE QREMOTE (YOURQM) RQMNAME(REALQM)
```

這會顯示當應用程式將訊息放入佇列管理程式 YOURQM 時，要使用的實際佇列管理程式是 REALQM。如果本端佇列管理程式為 REALQM，則會將訊息放入佇列 THISQ(本端佇列)。如果本端佇列管理程式不是稱為 REALQM，它會將訊息遞送至稱為 REALQM 的傳輸佇列。佇列管理程式會將傳輸標頭變更為 REALQM，而非 YOURQM。

出埠訊息-變更或指定傳輸佇列

第 54 頁的圖 16 顯示訊息到達佇列管理程式 QM1 的實務範例，其中傳輸標頭顯示佇列管理程式中的佇列名稱 QM3。在此實務範例中，透過 QM2 可透過多個跳躍來呼叫到 QM3。

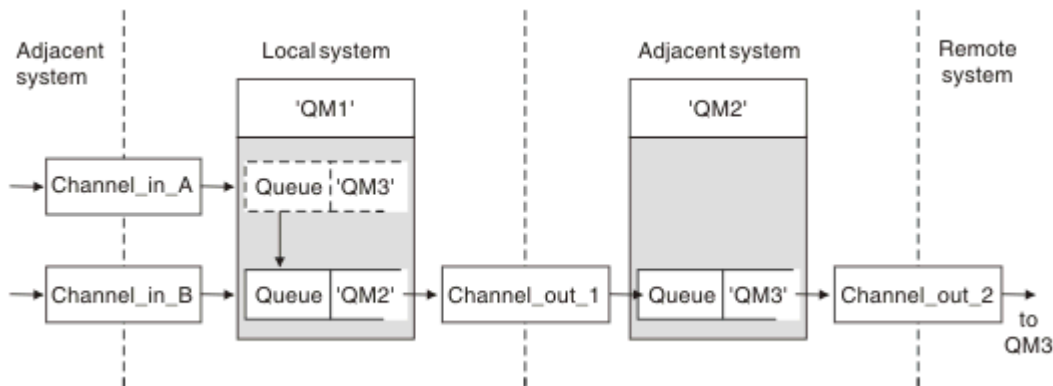


圖 16: 佇列管理程式別名

QM3 的所有訊息都會以佇列管理程式別名在 QM1 中擷取。佇列管理程式別名為 QM3，且包含 QM3 透過傳輸佇列 QM2 的定義。定義類似下列範例：

```
DEFINE QREMOTE (QM3) RNAME(' ') RQMNAME(QM3) XMITQ(QM2)
```

佇列管理程式會將訊息放入傳輸佇列 QM2，但不會變更傳輸佇列標頭，因為目的地佇列管理程式 QM3 的名稱不會變更。

所有到達 QM1 並顯示包含佇列名稱 (位於 QM2) 的傳輸標頭的訊息，也會放在 QM2 傳輸佇列上。以此方式，將具有不同目的地的訊息收集到共用傳輸佇列至適當的相鄰系統，以向前傳輸至其目的地。

入埠訊息-決定目的地

接收 MCA 會開啟傳輸標頭中所參照的佇列。如果存在與所參照的佇列管理程式同名的佇列管理程式別名定義，則會以該定義中的 RQMNAME 取代傳輸標頭中所收到的佇列管理程式名稱。

此處理程序有兩個用途：

- 將訊息導向至另一個佇列管理程式
- 將佇列管理程式名稱變更為與本端佇列管理程式相同

回覆目的地佇列別名定義

回覆目的地佇列別名定義會在訊息描述子中指定回覆資訊的替代名稱。其優點是您可以變更佇列或佇列管理程式的名稱，而不需要變更應用程式。

佇列名稱解析

當應用程式回覆訊息時，它會使用所收到訊息的訊息描述子中的資料來找出要回覆的佇列名稱。傳送端應用程式會指出回覆傳送至何處，並將此資訊附加至其訊息。此概念必須作為應用程式設計的一部分進行協調。

在將訊息放入佇列之前，會在應用程式傳送端進行佇列名稱解析。此實例不尋常地使用佇列名稱解析。只有在未開啟佇列時，才會進行名稱解析。因此，在與訊息傳送至的遠端應用程式互動之前，會先進行佇列名稱解析。

使用佇列管理程式別名解析佇列名稱

通常應用程式會指定回覆目的地佇列，並將回覆目的地佇列管理程式名稱留白。佇列管理程式會在放置時間完成它自己的名稱。除非您想要將替代通道用於回覆（例如，使用傳輸佇列 `QM1_relief` 的通道，而不是使用傳輸佇列 `QM1` 的預設傳回通道），否則此方法運作良好。在此狀況下，傳輸佇列標頭中指定的佇列管理程式名稱不符合「實際」佇列管理程式名稱，但會使用佇列管理程式別名定義重新指定。為了沿著替代路徑傳回回覆，也必須使用回覆目的地佇列別名定義來對映回覆目的地佇列資料。

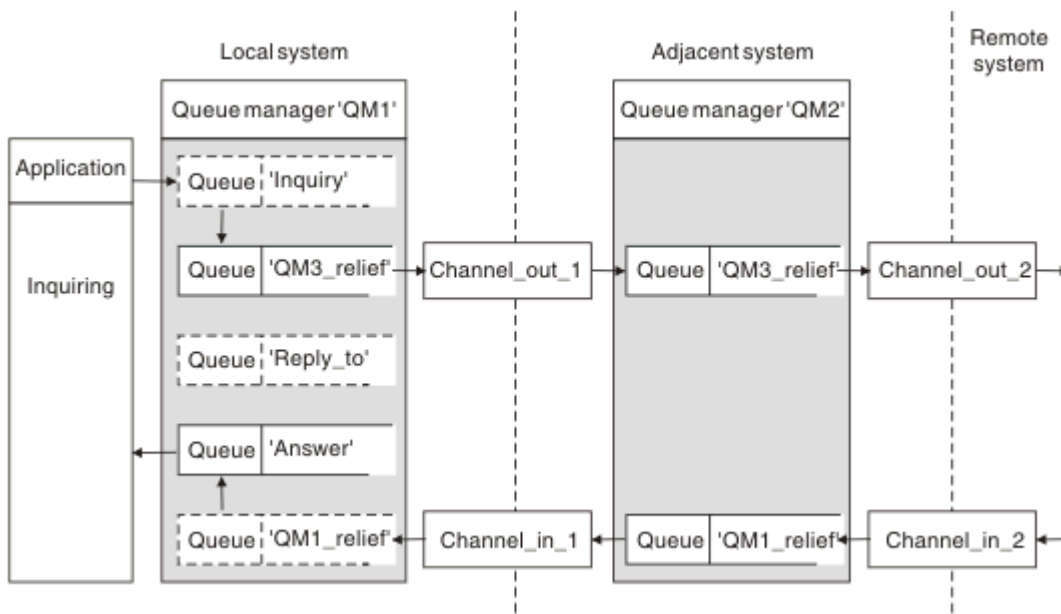


圖 17: 用於變更回覆位置的回覆目的地佇列別名

在 [第 55 頁的圖 17](#) 的範例中：

1. 應用程式會使用 `MQPUT` 呼叫並在訊息描述子中指定下列資訊來放置訊息：

```
ReplyToQ='Reply_to'  
ReplyToQMgr=' '
```

`ReplyTo` 佇列管理程式必須空白，才能使用回覆目的地佇列別名。

2. 您可以建立名為 `Reply_to` 的回覆目的地佇列別名定義，其中包含名稱 `Answer` 及佇列管理程式名稱 `QM1_relief`。

```
DEFINE QREMOTE ('Reply_to') RNAME ('Answer')  
RQMNAME ('QM1_relief')
```

3. 訊息會隨顯示 `ReplyToQ='Answer'` 和 `ReplyToQMgr='QM1_relief'` 的訊息描述子一起傳送。
4. 應用程式規格必須包含要在佇列 `Answer` 而非 `Reply_to` 中找到回覆的資訊。

若要準備回覆，您必須建立平行傳回通道，定義：

- 在 QM2，名為 QM1_relief 的傳輸佇列

```
DEFINE QLOCAL ('QM1_relief') USAGE(XMITQ)
```

- 在 QM1，佇列管理程式別名 QM1_relief

```
DEFINE QREMOTE ('QM1_relief') RNAME() RQMNAME(QM1)
```

此佇列管理程式別名會終止平行傳回通道的鏈結，並擷取 QM1 的訊息。

如果您認為您可能想要在未來某個時間執行此動作，請確保應用程式從一開始就使用別名。目前這是回覆目的地佇列的一般佇列別名，但稍後可以變更為佇列管理程式別名。

回覆目的地佇列名稱

命名回覆目的地佇列時需要小心。應用程式在訊息中放置回覆目的地佇列名稱的原因是它可以指定其回覆傳送至的佇列。當您建立具有此名稱的回覆目的地佇列別名定義時，您無法具有具有相同名稱的實際回覆目的地佇列 (即本端佇列定義)。因此，回覆目的地佇列別名定義必須包含新的佇列名稱及佇列管理程式名稱，且應用程式規格必須包含在此其他佇列中找到其回覆的資訊。

現在，當應用程式放置原始訊息時，它們必須從不同佇列中擷取訊息，該佇列不同於它們命名為回覆目的地佇列的佇列。

叢集運作方式

瞭解何謂叢集及其運作方式。

叢集是以某種方式邏輯關聯的佇列管理程式網路。叢集中的佇列管理程式實際上可能是遠端。例如，它們可能代表國際連鎖商店的分支，並實際位於不同國家。企業內的每一個叢集都必須具有唯一名稱。

通常叢集包含邏輯上以某種方式相關且需要共用部分資料或應用程式的佇列管理程式。例如，對於公司中的每一個部門，您可能有一個佇列管理程式，用於管理該部門特有的資料及應用程式。您可以將所有這些佇列管理程式分組到叢集中，以便它們全部饋送至薪資應用程式。或者，您可以針對鏈結商店的每一個分支各有一個佇列管理程式，以管理該分支的庫存層次及其他資訊。如果您將這些佇列管理程式分組到叢集中，它們都可以存取同一組銷售和購買應用程式。銷售和採購應用程式可以集中保留在總部佇列管理程式上。

設定叢集之後，其中的佇列管理程式可以彼此通訊，而不需要定義額外的通道定義或遠端佇列定義。

您可以將佇列管理程式的現有網路轉換為叢集，也可以在設定新網路的過程中建立叢集。

IBM WebSphere MQ 用戶端可以連接至屬於叢集的佇列管理程式，就像它可以連接至任何其他佇列管理程式一樣。

叢集也可以用於工作量管理。如需相關資訊，請參閱 [使用叢集進行工作量管理](#)。

如何在叢集中遞送訊息

如果您熟悉 IBM WebSphere MQ 及分散式佇列，請將叢集視為由依良心系統管理者維護的佇列管理程式網路。每當您定義叢集佇列時，系統管理者會視需要在其他佇列管理程式上自動建立對應的遠端佇列定義。

您不需要建立傳輸佇列定義，因為 IBM WebSphere MQ 會在叢集中的每一個佇列管理程式上提供傳輸佇列。此單一傳輸佇列可用來將訊息傳送至叢集中的任何其他佇列管理程式。您不限於使用單一傳輸佇列。佇列管理程式可以使用多個傳輸佇列來區隔進入叢集中每一個佇列管理程式的訊息。一般而言，佇列管理程式會使用單一叢集傳輸佇列。您可以變更佇列管理程式屬性 DEFCLXQ，以便佇列管理程式針對叢集中的每一個佇列管理程式使用不同的叢集傳輸佇列。您也可以手動定義叢集傳輸佇列。

加入叢集的所有佇列管理程式都同意以這種方式運作。它們會送出其本身及其所管理佇列的相關資訊，並接收叢集其他成員的相關資訊。

此資訊儲存在儲存庫中。大部分佇列管理程式只會保留它們需要的資訊，亦即它們需要與其通訊之佇列及佇列管理程式的相關資訊。每一個佇列管理程式都會將資訊保留在局部儲存庫中。部分指定的佇列管理程式會保留完整儲存庫，其中包含叢集中所有佇列管理程式的所有相關資訊。

若要成為叢集的一部分，佇列管理程式必須有兩個通道：叢集傳送端通道和叢集接收端通道

叢集傳送端通道是與傳送端通道類似的通訊通道。您必須在佇列管理程式上手動建立一個叢集傳送端通道，才能將它連接至已是叢集成員的完整儲存庫。

叢集接收端通道是類似於接收端通道的通訊通道。您必須手動建立一個叢集接收端通道。通道充當佇列管理程式接收叢集通訊的機制

然後會自動建立此佇列管理程式與叢集任何其他成員之間的通訊可能需要的所有其他通道

支援叢集的平台上的佇列管理程式不必成為叢集的一部分。您可以繼續使用分散式佇列技術，也可以使用叢集，或不使用叢集。

叢集範例

第 57 頁的圖 18 顯示稱為 CLSTR1 之叢集的元件。

- 在此叢集中，有三個佇列管理程式: QM1、QM2 及 QM3。
- QM1 和 QM2 管理叢集中所有佇列管理程式及叢集相關物件相關資訊的儲存庫。它們稱為完整儲存庫佇列管理程式。在圖表中，會以陰影化磁柱來代表儲存庫。
- QM2 及 QM3 管理叢集中任何其他佇列管理程式可存取的部分佇列。叢集中任何其他佇列管理程式可存取的佇列稱為叢集佇列。在圖表中，叢集佇列以陰影化佇列表示。可以從叢集中的任何位置存取叢集佇列。IBM WebSphere MQ 叢集作業程式碼可確保叢集佇列的遠端佇列定義建立在參照它們的任何佇列管理程式上。

與分散式佇列一樣，應用程式會使用 MQPUT 呼叫，將訊息放置在叢集中任何佇列管理程式的叢集佇列上。應用程式使用 MQGET 呼叫，只從佇列所在的佇列管理程式上的叢集佇列擷取訊息。

- 每一個佇列管理程式都有一個手動建立的定義，用於通道 (稱為 *cluster-name.queue-manager*) 的接收端，它可以在該通道上接收訊息。在接收端佇列管理程式上，*cluster-name.queue-manager* 是叢集接收端通道。叢集接收端通道類似於分散式佇列中使用的接收端通道; 它會接收佇列管理程式的訊息。此外，它還會接收叢集的相關資訊。

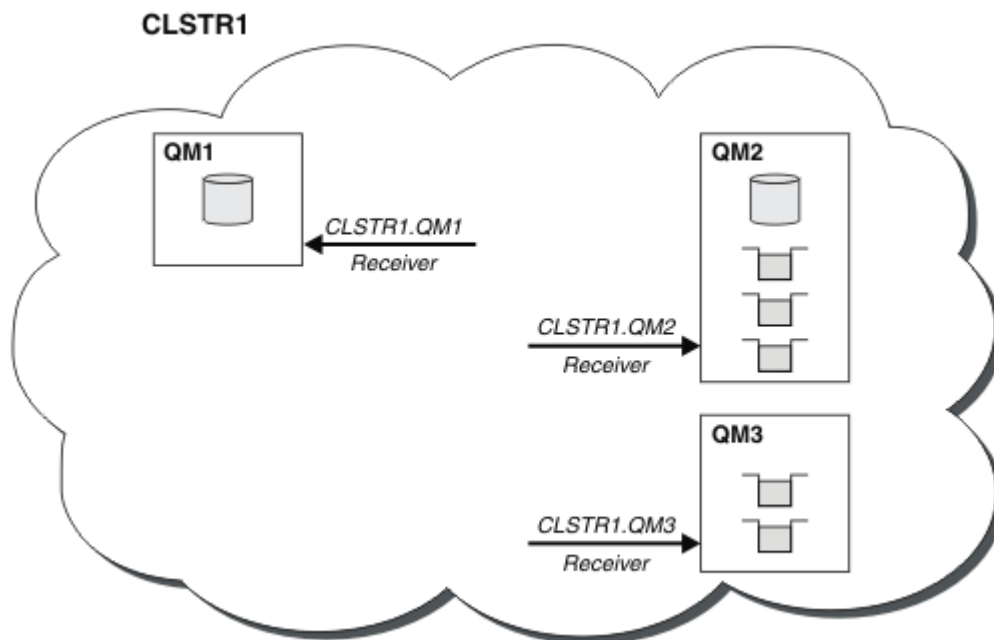


圖 18: 佇列管理程式的叢集

- 在第 58 頁的圖 19 中，每一個佇列管理程式也有通道傳送端的定義。它會連接至其中一個完整儲存庫佇列管理程式的叢集接收端通道。在傳送端佇列管理程式上，*cluster-name.queue-manager* 是叢集傳送端通道。QM1 及 QM3 具有連接至 CLSTR1.QM2 的叢集傳送端通道，請參閱點虛線 "2"。

QM2 具有連接至 CLSTR1.QM1 的叢集傳送端通道，請參閱點虛線 "3"。叢集傳送端通道類似於分散式佇列中使用的傳送端通道; 它會將訊息傳送至接收端佇列管理程式。此外，它還會傳送叢集的相關資訊。

一旦定義了通道的叢集接收端和叢集傳送端，通道會自動啟動。

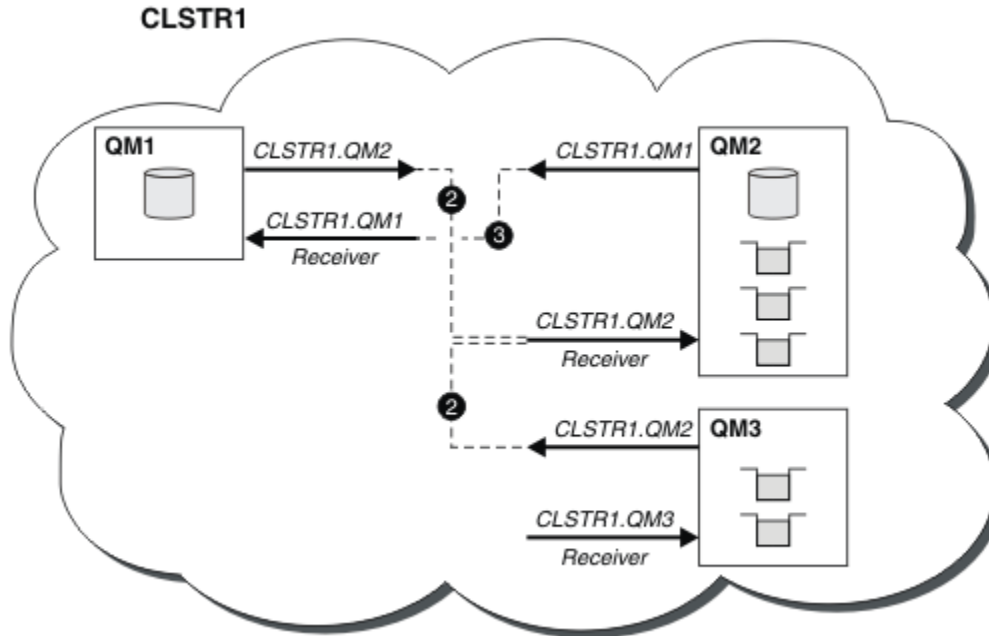


圖 19: 具有傳送端通道的佇列管理程式叢集

叢集作業的運作方式為何?

在本端佇列管理程式上定義叢集傳送端通道，會將該佇列管理程式引進其中一個完整儲存庫佇列管理程式。完整儲存庫佇列管理程式會相應地更新其完整儲存庫中的資訊。然後，它會自動建立回到原始佇列管理程式的叢集傳送端通道，並傳送該佇列管理程式的叢集相關資訊。因此，佇列管理程式會學習叢集，而叢集會學習佇列管理程式。

再次查看第 57 頁的圖 18。假設連接至佇列管理程式 QM3 的應用程式想要將部分訊息傳送至 QM2 中的佇列。QM3 第一次必須存取這些佇列時，它會透過諮詢完整儲存庫來探索這些佇列。在此情況下，完整儲存庫是 QM2，可使用傳送端通道 CLSTR1.QM2 來存取。利用儲存庫中的資訊，它可以自動建立那些佇列的遠端定義。如果佇列位於 QM1 上，則此機制仍可運作，因為 QM2 是完整儲存庫。完整儲存庫具有叢集中所有物件的完整記錄。在後一種情況下，QM3 也會自動建立叢集傳送端通道 (對應於 QM1 上的叢集接收端通道)，以容許兩者之間直接通訊。

第 59 頁的圖 20 顯示相同的叢集，以及兩個自動建立的叢集傳送端通道。叢集傳送端通道由與叢集接收端通道 CLSTR1.QM3 結合的兩條虛線代表。它也會顯示 QM1 用來傳送其訊息的叢集傳輸佇列 SYSTEM.CLUSTER.TRANSMIT.QUEUE。叢集中的所有佇列管理程式都有一個叢集傳輸佇列，它們可以從中將訊息傳送至相同叢集中的任何其他佇列管理程式。

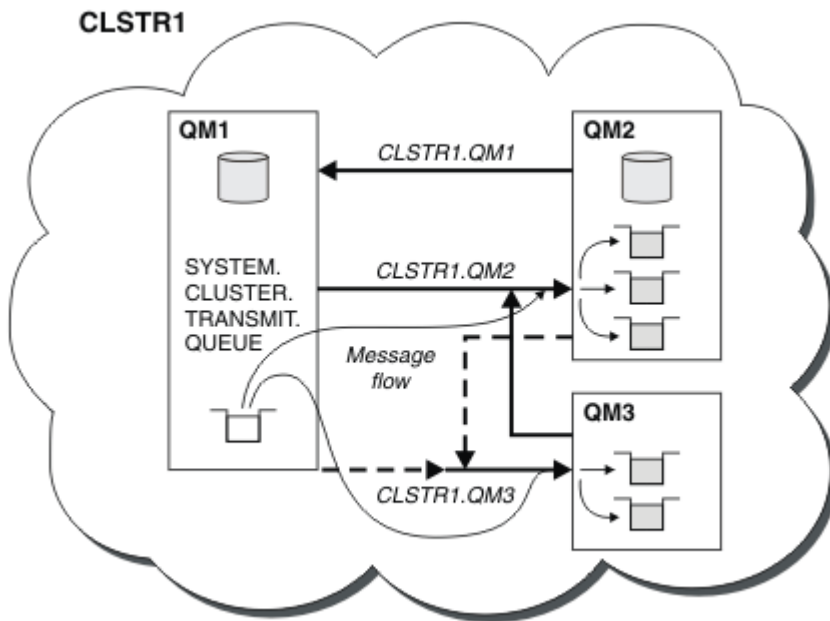


圖 20: 佇列管理程式的叢集，顯示自動定義的通道

註: 其他圖表只會顯示您為其進行手動定義之通道的接收端。會省略傳送端，因為它們大部分會在需要時自動定義。大部分叢集傳送端通道的自動定義對於叢集的功能和效率至關重要。

相關概念

叢集

您可以將叢集中的佇列管理程式分組。叢集中的佇列管理程式可以讓它們所管理的佇列可供叢集中的所有其他佇列管理程式使用。任何佇列管理程式都可以將訊息傳送至相同叢集中的任何其他佇列管理程式，而不需要標準分散式佇列作業所需的許多物件定義。

叢集作業與分散式佇列作業的比較

叢集的元件

相關工作

配置佇列管理程式叢集

設定新的叢集

管理 WebSphere MQ 叢集

IBM WebSphere MQ Telemetry

IBM WebSphere MQ Telemetry 包含屬於佇列管理程式一部分的遙測 (MQXR) 服務、您可以自行撰寫的遙測用戶端，或使用所提供的其中一個用戶端，以及指令行和瀏覽器管理介面。遙測是指從各種遠端裝置收集資料以及管理這些裝置。使用 IBM WebSphere MQ Telemetry，您可以將裝置的資料收集及控制與 Web 應用程式整合。

WebSphere Message Broker 或 WebSphere MQ 7.0.1 版先前提提供 MQTT 支援，其中 WebSphere MQ Telemetry 是個別特性。因為 WebSphere MQ Telemetry 是 WebSphere MQ 7.1 版及更新版本的元件，所以升級基本上是解除安裝 WebSphere MQ Telemetry 7.0.1 版及安裝 WebSphere MQ 7.1 版。WebSphere MQ Telemetry 可以隨主要產品一起安裝，或在已安裝 7.1 版或更新版本之後安裝。如需移轉資訊，請參閱 [從 Version 7.0.1 至 Version 7.5 移轉 IBM WebSphere MQ Telemetry 或使用 WebSphere Message Broker 第 6 版來移轉遙測應用程式](#)，以使用 IBM WebSphere MQ Telemetry 及 WebSphere Message Broker 7.0 版。

IBM WebSphere MQ Telemetry 中包含下列元件：

遙測通道

使用遙測通道來管理 MQTT 用戶端與 IBM WebSphere MQ 的連線。遙測通道會使用新的 IBM WebSphere MQ 物件 (例如 SYSTEM.MQTT.TRANSMIT.QUEUE) 來與 IBM WebSphere MQ 互動。

遙測 (MQXR) 服務

MQTT 用戶端使用 SYSTEM.MQXR.SERVICE 遙測服務來連接至遙測通道。

IBM WebSphere MQ Explorer IBM WebSphere MQ Telemetry

IBM WebSphere MQ Telemetry 可以使用 IBM WebSphere MQ Explorer 來管理。

用戶端軟體開發套件 (SDK)

用戶端 SDK 有四個部分：

1. 適用於 Java SE 及 Java ME 的 MQTT v3 用戶端程式庫。使用 Java 程式庫來撰寫支援 Java SE 或 Java ME 之裝置的 Java 用戶端。
2. 適用於 C 的 MQTT v3 程式庫。使用 C 程式庫來撰寫許多平台的 C 用戶端。
3. IBM WebSphere MQ Telemetry daemon for devices, 以 C 撰寫且在許多平台上執行的進階用戶端。
4. MQTT v3 通訊協定。MQTT v3 通訊協定已發佈並授權重複使用。使用通訊協定並參照 MQTT 用戶端實作，以撰寫適用於不同平台及語言的 MQTT 用戶端。

Documentation

IBM WebSphere MQ Telemetry 文件包含在來自 Version 7.1 的標準 IBM WebSphere MQ 產品說明文件中。Java 和 C 用戶端的 SDK 文件在產品說明文件中提供，並以 Javadoc 和 HTML 形式提供。

遙測概念

您可以從周圍的環境收集資訊，以決定要執行的動作。身為消費者，在決定要購買哪些食物之前，您先檢查商店中的商店。您想要知道如果您現在離開，在預訂連線之前，旅程將會花費多久時間。在決定是否去看醫生之前你先檢查一下你的症狀。在決定是否等待之前，您先檢查巴士何時抵達。這些決策的資訊直接來自儀表和裝置，來自紙上或螢幕上的書面文字，以及來自您的資訊。在任何地方，當您需要時，您可以收集資訊，將它結合在一起，分析它，並對它採取行動。

如果資訊來源廣泛分散或無法存取，則收集最準確的資訊會變得困難且代價高昂。如果您要進行許多變更，或很難進行變更，則不會進行變更，或會在變更無效時進行。

如果將具有數位技術的裝置連接至網際網路，以大幅降低從裝置收集資訊及控制廣泛分散裝置的成本，該怎麼辦？可以使用網際網路和企業的資源來分析資訊。您有更多機會做出明智決策並採取行動。

技術趨勢以及環境和經濟壓力正在推動這些變化的發生：

1. 由於與低成本數字處理器的標準化和連線，連線和控制感測器和致動器的成本正在降低。
2. 網際網路和網際網路技術越來越被用來連接裝置。在一些國家，行動電話在連線網際網路應用的數量上超過個人電腦。其他裝置肯定在追蹤。
3. 網際網路和網際網路技術讓應用程式更容易取得資料。輕鬆存取資料會推動使用資料分析，將資料從感應器轉換成在許多其他解決方案中很有用的資訊。
4. 智慧型資源使用通常是一種更快速且更便宜的方式來減少碳排放及成本。替代方案：尋找新資源或開發新技術以使用現有資源，可能是長期解決方案。短期而言，開發新技術或尋找新資源往往比改善現有解決方案風險更大、更緩慢且更昂貴。

範例

範例顯示這些趨勢如何創造新的機會，以智慧方式與環境互動。

The International Convention for the Safety of Life at Sea (SOLAS) requires Automatic Identification System (AIS) to be deployed on many ships. 它需要 300 噸以上的商船和客船。AIS 主要是沿海航運的避免碰撞系統。它被海洋當局用來監測和控制沿海水域。

世界各地的發燒友正在部署低成本 AIS 追蹤站，並將沿海航運資訊放到網際網路上。其他愛好者正在撰寫將來自 AIS 的資訊與來自網際網路的其他資訊相結合的應用程式。結果會放在網站上，並使用 Twitter 和 SMS 來發佈。

在一份申請中，來自南安普頓附近 AIS 站的資訊與船舶所有權和地理資訊相結合。應用程式會將渡船抵達及離開的相關即時資訊提供給 Twitter。使用南安普頓和懷特島之間渡輪的普通通勤者使用 Twitter 或 SMS 訂閱新聞資訊。如果訊號顯示他們的渡輪晚了，乘客可以推遲他們的出發時間，並在渡輪抵達時間晚於預定的到達時間時搭船。

如需其他範例，請參閱 [第 61 頁的『遙測概念及監視與控制實務範例』](#)。

相關工作

[安裝 WebSphere MQ Telemetry](#)

[管理 WebSphere MQ Telemetry](#)

[從使用 WebSphere Message Broker 第 6 版移轉遙測應用程式，以使用 WebSphere MQ Telemetry 及 WebSphere Message Broker 7.0](#)

[將 WebSphere MQ Telemetry 從 7.0.1 版移轉至 7.5 版](#)

[開發 WebSphere MQ Telemetry 應用程式](#)

[WebSphere MQ Telemetry 疑難排解](#)

相關參考

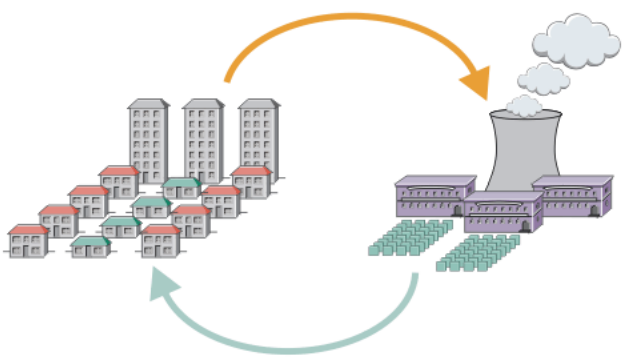
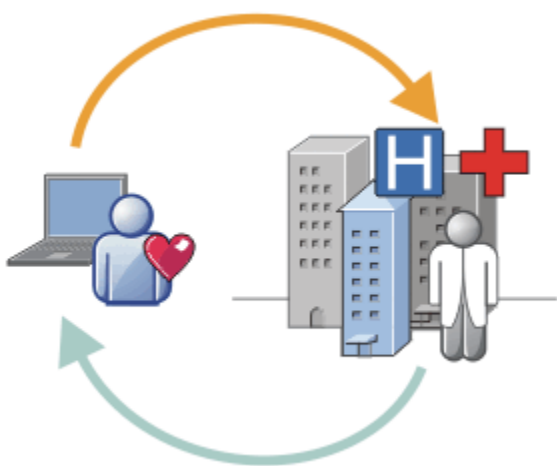
[WebSphere MQ Telemetry 參照](#)

遙測概念及監視與控制實務範例

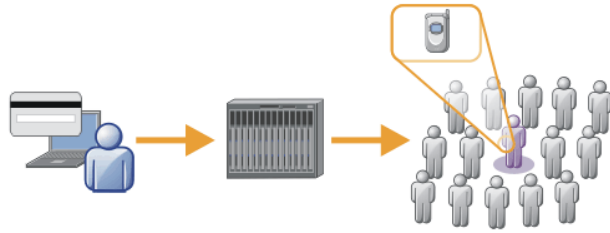
遙測可用於自動感應、測量資料及控制遠端裝置。主要用於將資料從裝置傳輸至中央控制點。遙測還包括將配置及控制資訊傳送至裝置。

IBM WebSphere MQ Telemetry 使用 MQTT 通訊協定連接小型裝置，並使用 IBM WebSphere MQ 將裝置連接至其他應用程式。IBM WebSphere MQ Telemetry 橋接裝置與網際網路之間的時間，讓您更容易建置「智慧型解決方案」。智慧型解決方案會針對監視及控制裝置的應用程式，解除鎖定網際網路上及企業應用程式中可用的豐富資訊。

下列圖表示範 IBM WebSphere MQ Telemetry 的一些一般用法：

遙測: 智慧型電力	
	<ul style="list-style-type: none">• 包含傳送至服務提供者之能源使用資料的 MQTT 訊息。• IBM WebSphere MQ Telemetry 會根據能源使用資料的分析來傳送 CONTROL COMMANDS。• 如需相關資訊，請參閱下列實務範例: 第 63 頁的『遙測實務範例：家庭能源監視及控制』
遙測: 智慧型健康服務	
<ul style="list-style-type: none">• IBM WebSphere MQ Telemetry 會將「健康資料」傳送至您的醫院與醫生。• 可以根據「性能資料」分析來傳送 MQTT 訊息警示或意見。• 如需相關資訊，請參閱下列實務範例: 第 62 頁的『遙測實務範例：家庭病患監視』	

遙測: 一個人在人群中



- 簡式卡片交易會傳送至銀行的伺服器。
- IBM WebSphere MQ Telemetry 可識別數千中的一個人員，並警示客戶已使用其卡。
- IBM WebSphere MQ Telemetry 可以使用最簡單的資訊輸入，並尋找該個人。

從實際範例中取得的後續實務範例說明使用遙測的一些方法，以及遙測技術必須解決的一些一般問題。

相關概念

第 62 頁的『[遙測實務範例: 家庭病患監視](#)』

在 IBM 與心臟病患者護理系統上的醫療保健提供者之間的協同作業中，植入的心臟病發作除顫器會與醫院進行通訊。病患及植入式裝置的相關資料，將透過使用 RF 遙測傳送至病患家中的 MQTT 裝置。

第 63 頁的『[遙測實務範例: 家庭能源監視及控制](#)』

第 64 頁的『[遙測實務範例: 無線電頻率識別 \(RFID\)](#)』

第 65 頁的『[遙測實務範例: 環境感應](#)』

環境感應使用遙測來收集河流水位及品質、大氣汙染物及其他環境資料的相關資訊。

第 65 頁的『[遙測實務範例: 行動式應用程式](#)』

行動式應用程式是在無線裝置上執行的應用程式。裝置是一般應用程式平台或自訂裝置。

遙測實務範例: 家庭病患監視

在 IBM 與心臟病患者護理系統上的醫療保健提供者之間的協同作業中，植入的心臟病發作除顫器會與醫院進行通訊。病患及植入式裝置的相關資料，將透過使用 RF 遙測傳送至病患家中的 MQTT 裝置。

通常是每晚傳送至位於床端的轉送器。轉送器會透過電話系統安全地將資料傳送至醫院，並在其中分析資料。

該系統減少了病人必須向醫生就診的次數。它會偵測病患或裝置何時需要注意，並在發生緊急狀況時警示待命醫師。

IBM 與醫療保健提供者之間的協同作業具有一些遙測實務範例常見的性質:

Invisibility

裝置不需要使用者介入，只需要提供電源、電話線，以及在一天中的部分時間接近裝置。本實用新型結構簡單，操作可靠，使用方便。

為了消除患者設定裝置的需求，裝置供應商預先配置裝置。病人只能把它插進去 患者消除配置簡化了裝置的操作，減少了裝置錯誤配置的機會。

MQTT 用戶端內嵌為裝置的一部分。裝置開發人員會將 MQTT 用戶端實作內嵌在裝置中，且開發人員或供應商會將 MQTT 用戶端配置為預先配置的一部分。

MQTT 用戶端以 Java SE and Java ME Jar 檔提供，開發人員在其 Java 應用程式中包括該檔案。對於非 Java 環境 (例如此環境)，裝置開發人員可以使用已發佈的 MQTT 格式及通訊協定，以不同語言實作用戶端。另外，開發人員也可以使用 Windows、Linux 和 ARM 平台所提供的其中一個 C 用戶端作為共用程式庫。

連線不平均

去纖顫器與醫院之間的通訊具有不平均的網路特性。採用兩種不同的網路，解決了從病人採集資料，並將資料傳送到醫院的不同問題。在專利與 MQTT 裝置之間，使用短程低功率 RF 網路。轉送器會透過低頻寬電話線，使用 VPN TCP/IP 連線來連接至醫院。

通常無法找到將每個裝置直接連接至 Internet Protocol 網路的方法。使用由中心連接的兩個網路是一般解決方案。MQTT 裝置是一個簡單的中心，儲存來自病患的資訊，並將其轉遞至醫院。

安全

醫生必須能夠信任病患資料的真實性，且病患想要尊重其資料的隱私。

在某些情況下，使用 VPN 或 SSL 來加密連線已足夠。在其他實務範例中，即使在儲存資料之後，也需要保持資料安全。

有時遙測裝置並不安全。例如，它可能位於共用住宅中。必須鑑別裝置的使用者，以確定資料來自正確的病患。裝置本身可以使用 SSL 向伺服器進行鑑別，而伺服器則可以向裝置進行鑑別。

裝置與佇列管理程式之間的遙測通道支援 JAAS 進行使用者鑑別，以及 SSL 進行通訊加密及裝置鑑別。發佈的存取權是由 WebSphere MQ 中的物件權限管理程式所控制。

用來鑑別使用者的 ID 可以對映至不同的 ID，例如一般病患身分。一般 ID 可簡化 WebSphere MQ 中發佈主題的配置授權。

連線功能

MQTT 裝置與醫院之間的連線使用撥號，並使用低到 300 波的頻寬。

為了以 300 Baud 有效運作，MQTT 通訊協定除了 TCP/IP 標頭之外，只會將幾個額外位元組新增至訊息。

MQTT 通訊協定提供單一傳輸「發動並忘記」傳訊，可保持低延遲。它也可以使用多個傳輸來保證「至少一次」和「正好一次」遞送，如果保證遞送比回應時間更重要的話。為了保證遞送，訊息會儲存在裝置中，直到順利遞送為止。如果裝置以無線方式連線，則保證遞送特別有用。

可調整性

遙測裝置通常會大量部署，從數萬個到數百萬個。

將許多裝置連接至系統會對解決方案提出大量需求。有一些商業需求，例如裝置及其軟體的成本，以及管理授權、裝置及使用者的管理需求。技術需求包括網路及伺服器上的負載。

開啟連線所使用的伺服器資源多於維護開啟連線所使用的伺服器資源。但在這類使用電話線的情境中，連線費用表示連線保持開啟的時間不會超過所需的時間。資料傳送基本上是批次性質。連線可以整晚排程，以避免在睡覺時突然出現連線尖峰。

在用戶端上，用戶端的可調整性是由所需的最小用戶端配置所協助。MQTT 用戶端內嵌在裝置中。不需要在將裝置部署至病患時內建配置或 MQTT 用戶端授權接受步驟。

在伺服器上，WebSphere MQ Telemetry 的起始目標是每個佇列管理程式開啟 50,000 個連線。

連線是使用「WebSphere MQ 探險家」來管理。「瀏覽器」會將要顯示的連線過濾成可管理的數字。透過適當選擇將 ID 配置給用戶端的方法，您可以根據地理位置或依病患名稱的字母順序來過濾連線。

遙測實務範例：家庭能源監視及控制

智慧型計量會收集比傳統計量更多關於能源消耗的詳細資料。

智慧型電表通常與本端遙測網路結合使用，以監視和控制家中的個別電器用品。一些智慧型電表也會以遠端方式連接，以便在遠方進行監視和控制。

遠端連線可以由個人、電源公用程式或中央控制點設定。遠端控制點可以讀取電源使用情形並提供使用情形資料。它可以提供資料來影響使用情形，例如連續定價和天氣資訊。可以限制負載，提高整體發電效率。

智慧型計量器已開始廣泛部署。例如，英國政府正在就 2020 年前在英國每個家庭部署智慧儀表進行磋商。

起始計量實務範例具有許多一般性質：

Invisibility

除非使用者想要使用計量來參與節省能源，否則計量不需要使用者人為介入。它不能降低個別電器的能源供應的可靠性。

MQTT 用戶端可以內嵌在隨計量器部署的軟體中，且不需要個別安裝或配置。

連線不平均

應用裝置與智慧型計量器之間的通訊需要與計量器與遠端連線點之間不同的連線標準。

從智慧型計量器到應用裝置的連線必須具有高可用性，且符合家庭區域網路的網路標準。

遠端網路可能使用各種實體連線。其中有些 (例如行動電話) 具有高傳輸成本，且可能是間歇性的。MQTT v3 規格是針對遠端連線，以及本端配接卡與智慧型計量之間的連線。

電源插座與應用程式與計量器之間的連線使用住家區域網路，例如 Zigbee。MQTT for sensor 網路 (MQTT-S) 設計為使用 Zigbee 及其他低頻寬網路通訊協定。WebSphere MQ Telemetry 不直接支援 MQTT-S。它需要闢道才能將 MQTT-S 連接至 MQTT v3。

如同家庭病患監視，家庭能源監視和控制的解決方案需要多個網路，使用智慧型計量作為中心來連接。

安全

有許多與智慧型計量相關聯的安全問題。這些問題包括交易不可否認性、已起始任何控制動作的授權，以及耗電量資料的隱私權。

為了確保隱私權，MQTT 在計量器與遠端控制點之間傳送的資料可以使用 SSL 進行加密。為了確保控制動作的授權，可以使用 SSL 來相互鑑別計量與遠端控制點之間的 MQTT 連線。

連線功能

遠端網路的實體本質可能會有很大的不同。它可能使用現有的寬頻連線，或使用具有高通話成本及間歇性可用性的行動網路。對於高成本、間歇性的連線，MQTT 是有效且可靠的通訊協定；請參閱第 62 頁的『遙測實務範例：家庭病患監視』。

可調整性

最終，電力公司 (或中央控制點) 計劃部署數千萬智慧電錶。一開始，每個部署的計量數是從數十到數十萬。此數目與每個佇列管理程式 50,000 個開放式用戶端連線的起始 MQTT 目標相當。

家用能源監視和控制架構的重要方面是使用智慧型計量器作為網路集中器。每一個應用裝置配接卡都是個別感應器。透過使用 MQTT 將它們連接至本端中心，該中心可以將資料流集中到具有中央控制點的單一 TCP/IP 階段作業，同時儲存訊息一段短時間以克服階段作業中斷。

遠端連線必須在家用能源實務範例中保持開啟狀態，原因有兩個。首先，因為相對於傳送要求，開啟連線需要很長的時間。在短間隔內開啟許多連線以傳送 "load-limitation" 要求的時間太長。其次，若要接收來自電力公司的負載限制要求，必須先由用戶端開啟連線。使用 MQTT，連線一律由用戶端起始，若要接收來自電力公司的負載限制要求，連線必須保持開啟。

如果開啟連線的速率是重要的，或伺服器起始時間重要要求，解決方案通常會維護許多開啟的連線。

遙測實務範例：無線電頻率識別 (RFID)

RFID 是使用內嵌式 RFID 標籤來無線識別及追蹤物件。RFID 標籤最多可以讀取數公尺的範圍，且不在 RFID 讀取器的視線範圍內。被動式標籤由 RFID 讀取器啟動。作用中標籤在沒有外部啟動的情況下傳輸。作用中標籤必須具有電源。被動標籤可以包括電源以增加其範圍。

RFID 在許多應用中被使用，情景的型別有很大的不同。RFID 情境，以及家庭病患監視與家庭能源監視與控制情境，有一些相似性與差異。

Invisibility

在許多實務範例中，RFID 讀取器會大量部署，且必須在沒有使用者介入的情況下運作。讀取器包括內嵌的 MQTT 用戶端，以與中央控制點進行通訊。

例如，在配送倉儲中，讀取器使用移動感應器來偵測棧板。它會啟動棧板上項目的 RFID 標籤，並將資料及要求傳送至中央應用程式。資料用來更新庫存的位置。要求會控制接下來棧板的狀況，例如將它移至特定機槽。航空公司和機場行李系統都以這種方式使用 RFID。

在某些 RFID 情境中，讀取器具有標準運算環境，例如 Java ME。在這些情況下，在製造之後，MQTT 用戶端可能會部署在不同的配置步驟中。

連線不平均

RFID 讀取器可能與包含 MQTT 用戶端的本端控制裝置分離，或者每一個讀取器可能內嵌 MQTT 用戶端。通常，地理或通訊因素會指出拓撲的選擇。

安全

隱私和真實性是 RFID 標籤附件中的安全問題。RFID 標籤是不明顯的，可以被祕密監視、欺騙或竄改。

RFID 安全問題解決方案增加部署新 RFID 解決方案的機會。雖然安全暴露在 RFID 標籤和當地讀取器中，但使用中央資訊處理會建議應對不同威脅的方法。例如，動態將庫存層次與遞送和分派產生關聯，即可偵測到標籤竄改。

連線功能

RFID 應用程式通常包含批次儲存及轉遞從 RFID 讀取器收集的資訊及立即查詢。在配送倉儲實務範例中，RFID 讀取器會一直連接。當讀取標籤時，它會連同讀取器的相關資訊一起發佈。倉儲應用程式會將回應發佈回讀取器。

在倉儲應用程式中，網路通常是可靠的，而立即要求可能會使用「發動並忘記」訊息，以獲得低延遲效能。批次儲存及轉遞資料可能使用「正好一次」傳訊，以將與鬆散資料相關聯的管理成本降到最低。

可調整性

如果 RFID 應用程式需要立即回應，則在第二個或第二個的順序中，RFID 讀取器必須保持連線。

遙測實務範例: 環境感應

環境感應使用遙測來收集河流水位及品質、大氣汙染物及其他環境資料的相關資訊。

感應器經常位於遠端位置，無法存取有線通訊。無線頻寬昂貴，可靠性低。通常，小地理區域中的許多環境感應器會連接至安全位置中的本端監視裝置。本端連線可能是有線或無線。

Invisibility

與中央監視裝置相比，感應器裝置可能較不容易存取、供電較低，且部署的數目較多。感應器有時會「啞巴」，而本端監視裝置包括用來轉換及儲存感應器資料的配接卡。監視裝置可能會納入支援 Java SE 或 ME 的通用電腦。在配置 MQTT 用戶端時，不可見性不太可能是主要需求。

連線不平均

感應器的功能，以及遠端連線的成本和頻寬，通常會導致本端監視中心連接至中央伺服器。

安全

除非在軍事或防禦情境中使用解決方案，否則安全不是主要需求。

連線功能

許多用途不需要持續監視或立即提供資料。異常狀況資料(例如洪水層次警示)需要立即轉遞。在本端監視器中聚集感應器資料，以減少連線及通訊成本，然後使用排程連線進行傳送。一旦在監視器上偵測到異常狀況資料，即會立即轉遞該異常狀況資料。

可調整性

感應器集中在本端中心周圍，並將感應器資料聚集到根據排程傳輸的封包中。這兩個因素都會減少使用直接連接感應器所強制的中央伺服器負載。

遙測實務範例: 行動式應用程式

行動式應用程式是在無線裝置上執行的應用程式。裝置是一般應用程式平台或自訂裝置。

一般平台包括手持裝置(如手機及個人資料助理)，以及行動式裝置(如筆記型電腦)。自訂裝置使用針對特定應用程式自訂的特殊用途硬體。記錄 "signed-for" 包裹遞送的裝置是自訂行動式裝置的範例。自訂行動式裝置上的應用程式通常建置在一般軟體平台上。

Invisibility

自訂行動式應用程式的部署是受管理的，並且可以包括 MQTT 用戶端應用程式的配置。在配置 MQTT 用戶端時，不可見性不太可能是主要需求。

連線不平均

與上述實務範例的本端中心拓撲不同，行動式用戶端會從遠端連接。用戶端應用程式層會直接連接至中央中心的應用程式。

安全

在幾乎沒有實體安全的情況下，必須鑑別行動式裝置及行動式使用者。SSL 用來確認裝置的身分，JAAS 用來鑑別使用者。

連線功能

如果移動應用依賴於無線覆蓋，它必須能夠離線操作，並且有效地處理中斷的連線。在此環境中，目標是保持連線，但應用程式必須能夠儲存及轉遞訊息。通常訊息是訂單或遞送確認，且具有重要商業價值。需要可靠地儲存並轉遞它們。

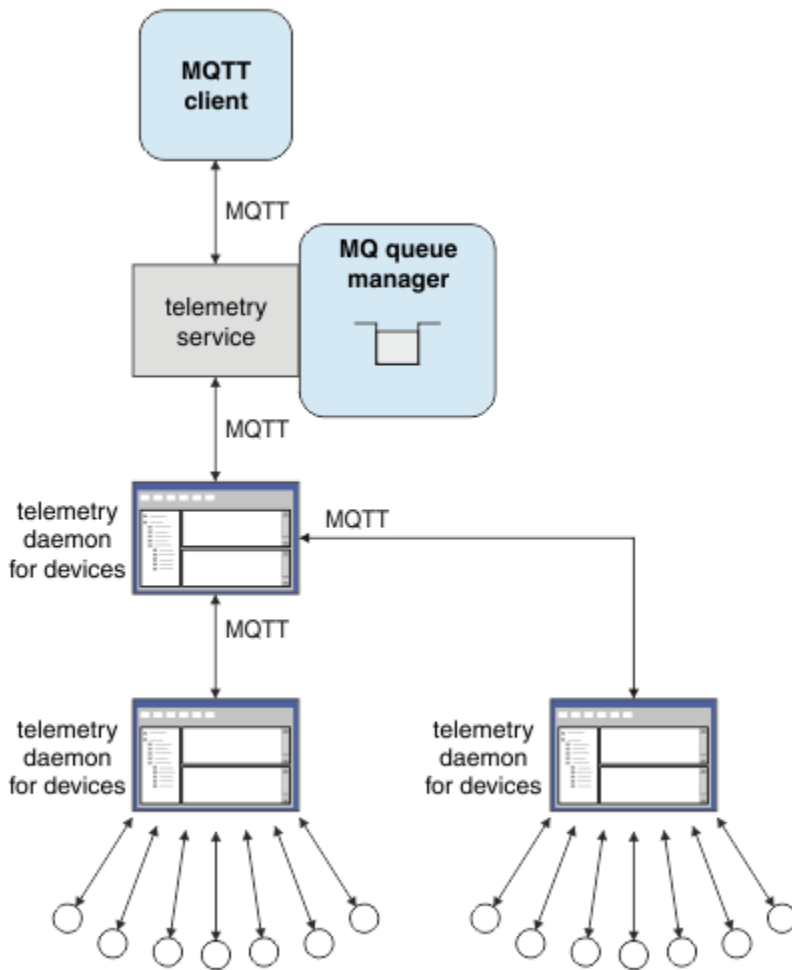
可調整性

可擴充性不是主要問題。在自訂行動式應用程式實務中，應用程式用戶端的數目可能不會超過數千或數萬個。

將遙測裝置連接至佇列管理程式

遙測裝置使用 MQTT v3 用戶端連接至佇列管理程式。MQTT v3 用戶端使用 TCP/IP 連接至稱為遙測 (MQXR) 服務的 TCP/IP 接聽器。

作為將遙測裝置直接連接至遙測服務的替代方案，您可以將裝置連接至裝置的 WebSphere MQ Telemetry 常駐程式。常駐程式自身是一個 MQTT 第 3 版用戶端。它會儲存裝置連線，並建立與遙測 (MQXR) 服務的單一連線。您可以連接階層中的常駐程式，將可間接連接至 IBM WebSphere MQ 的裝置數目增加許多數量級。



MQTT 用戶端使用 `MqttClient.connect` 方法起始 TCP/IP 連線。與 IBM WebSphere MQ 用戶端類似，MQTT 用戶端必須連接至佇列管理程式，才能傳送及接收訊息。使用隨 IBM WebSphere MQ Telemetry 安裝的 TCP/IP 接聽器 (稱為遙測 (MQXR) 服務) 在伺服器上建立連線。每一個佇列管理程式最多執行一項遙測 (MQXR) 服務。

遙測 (MQXR) 服務會使用 `MqttClient.connect` 方法中每一個用戶端所設定的遠端 Socket 位址，來配置與遙測通道的連線。Socket 位址是 TCP/IP 主機名稱與埠號的組合。遙測 (MQXR) 服務會將多個使用相同遠端 Socket 位址的用戶端連接至相同的遙測通道。

如果伺服器上有多個佇列管理程式，請在佇列管理程式之間分割遙測通道。在佇列管理程式之間配置遠端 Socket 位址。使用唯一的遠端 Socket 位址來定義每一個遙測通道。兩個遙測通道不得使用相同的 Socket 位址。

如果針對多個佇列管理程式上的遙測通道配置相同的遠端 Socket 位址，則第一個要連接的遙測通道會獲勝。在相同位址上連接的後續通道會失敗，並建立首次失敗資料擷取 (FDC) 檔案。

如果伺服器上有多個網路配接卡，請在遙測通道之間分割遠端 Socket 位址。只要只在一個遙測通道上配置任何特定 Socket 位址，Socket 位址的配置就完全是任意的。

使用 WebSphere MQ Telemetry supplement for IBM WebSphere MQ Explorer 中提供的精靈，配置 IBM WebSphere MQ 以連接 MQTT 用戶端。或者，遵循在 Linux 及 AIX 上配置遙測的佇列管理程式及在 Windows 上配置遙測的佇列管理程式中的指示，以手動配置遙測。

相關參考

[MQXR 內容](#)

遙測連線通訊協定

WebSphere MQ Telemetry 支援 TCP/IP IPv4 和 IPv6，以及 SSL。

遙測 (MQXR) 服務

遙測 (MQXR) 服務是一個 TCP/IP 接聽器，作為 IBM WebSphere MQ 服務進行管理。使用 IBM WebSphere MQ Explorer 精靈或 `runmqsc` 指令來建立服務。

IBM WebSphere MQ Telemetry (MQXR) 服務稱為 `SYSTEM.MQXR.SERVICE`。

IBM WebSphere MQ Telemetry supplement for IBM WebSphere MQ Explorer 中提供的 **Telemetry sample configuration** 精靈會建立遙測服務及遙測通道範例；請參閱 [使用 IBM WebSphere MQ Explorer 驗證 IBM WebSphere MQ Telemetry 的安裝](#)。從指令行建立配置範例；請參閱 [使用指令行驗證 IBM WebSphere MQ Telemetry 的安裝](#)。

遙測 (MQXR) 服務會隨佇列管理程式自動啟動及停止。使用 IBM WebSphere MQ Explorer 中的 `services` 資料夾來控制服務。若要查看服務，您必須按一下圖示，以停止瀏覽器從顯示畫面中濾除 `SYSTEM` 物件。

[installMQXRService_unix.mqsc](#) 顯示如何在 AIX 和 Linux 上手動建立服務的範例。

[installMQXRService_win.mqsc](#) 顯示如何在 Windows 中手動建立服務。

遙測通道

建立遙測通道以利用不同的內容（例如「Java 鑑別和授權服務 (JAAS)」或 SSL 鑑別）建立連線，或者管理用戶端群組。

使用 **New Telemetry Channel** 精靈建立遙測通道，該精靈在 IBM WebSphere MQ Explorer 的 IBM WebSphere MQ Telemetry 補充中提供。使用精靈來配置通道，以接受來自特定 TCP/IP 埠上 MQTT 用戶端的連線。從 Version 7.1 開始，您可以使用指令行程式 `runmqsc` 來配置 IBM WebSphere MQ Telemetry。

在不同埠上建立多個遙測通道，透過將用戶端分割成多個群組，讓大量用戶端連線更容易管理。每一個遙測通道都有不同的名稱。

您可以使用不同的安全屬性來配置遙測通道，以建立不同類型的連線。建立多個通道以接受不同 TCP/IP 位址上的用戶端連線。使用 SSL 來加密訊息並鑑別遙測通道及用戶端；請參閱 [MQTT 用戶端及遙測通道的 SSL 配置](#)。指定使用者 ID 以簡化 WebSphere MQ 物件的授權存取。指定 JAAS 配置，以使用 JAAS 來鑑別 MQTT 使用者；請參閱 [MQTT 用戶端識別、授權及鑑別](#)。

MQTT 通訊協定

MQ Telemetry Transport (MQTT) v3 通訊協定設計用於在低頻寬或高負載連線上的小型裝置之間交換訊息，以及可靠地傳送訊息。它使用 TCP/IP。

已發佈 MQTT 通訊協定; 請參閱 [MQ 遙測傳輸格式及通訊協定](#)。第 3 版通訊協定使用發佈/訂閱, 並支援三種服務品質:「發動並忘記」、「至少一次」及「正好一次」。

較小的通訊協定標頭及位元組陣列訊息有效負載會保留較小的訊息。標頭包含 2 個位元組的固定標頭, 以及最多 12 個位元組的其他變數標頭。通訊協定使用 12 位元組變數標頭來訂閱及連接, 而大部分發佈只使用 2 位元組變數標頭。

有了三種服務品質, 您可以在低延遲與可靠性之間進行交易; 請參閱 MQTT 用戶端所提供的服務品質。「隨發即忘」不使用持續裝置儲存體, 只使用一次傳輸來傳送或接收發佈。「至少一次」及「正好一次」需要裝置上的持續性儲存體, 以維護通訊協定狀態並儲存訊息, 直到確認為止。

通訊協定是其他產品中使用的 MQTT 通訊協定系列之一。

MQTT 用戶端

MQTT 用戶端應用程式負責從遙測裝置收集資訊、連接至伺服器, 以及將資訊發佈至伺服器。它也可以訂閱主題、接收發佈資訊, 以及控制遙測裝置。

與 IBM WebSphere MQ 用戶端應用程式不同, MQTT 用戶端應用程式不是 IBM WebSphere MQ 應用程式。它們未指定要連接的佇列管理程式。它們不限於使用特定的 IBM WebSphere MQ 程式設計介面。相反地, MQTT 用戶端會實作 MQTT version 3 通訊協定。您可以撰寫自己的用戶端程式庫, 以您選擇的程式設計語言及平台來連接至 MQTT 通訊協定。請參閱 [MQ Telemetry Transport 格式及通訊協定](#)。

若要簡化撰寫 MQTT 用戶端應用程式, 請使用封裝許多平台的 MQTT 通訊協定的 C、Java 及 JavaScript 用戶端程式庫。如需 MQTT 用戶端程式庫的用戶端 API 文件鏈結, 請參閱 [MQTT 用戶端程式設計參考手冊](#)。如果您在 MQTT 應用程式中納入這些程式庫, 則功能完整的 MQTT 用戶端可以短到 15 行程式碼。請參閱 [PubSync.java](#)。

已安裝兩個 `com.ibm.micro.client.mqttv3.jar` JAR 檔副本。一個副本具有版本號碼作為檔名的一部分。例如: `com.ibm.micro.client.mqttv3_3.0.2.0-20100723.jar`。在 OSGi 應用程式中使用版本化副本。JAR 檔的內容相同。

MQTT 用戶端應用程式一律負責起始與遙測通道的連線。連接之後, MQTT 用戶端應用程式或 IBM WebSphere MQ 應用程式可以開始交換訊息。

MQTT 用戶端應用程式及 IBM WebSphere MQ 應用程式會發佈及訂閱同一組主題。IBM WebSphere MQ 應用程式也可以直接將訊息傳送至 MQTT 用戶端應用程式, 而不需要用戶端應用程式先建立訂閱。請參閱 [配置分散式佇列以將訊息傳送至 MQTT 用戶端](#)。

MQTT 用戶端應用程式使用遙測通道連接至 IBM WebSphere MQ。遙測通道充當 MQTT 與 IBM WebSphere MQ 所使用的不同訊息類型之間的橋接器。它會代表 MQTT 用戶端應用程式在佇列管理程式中建立發佈及訂閱。遙測通道會將符合 MQTT 用戶端應用程式訂閱的發佈從佇列管理程式傳送至 MQTT 用戶端應用程式。

將訊息傳送至 MQTT 用戶端

WebSphere MQ 應用程式可以透過發佈至用戶端所建立的訂閱, 或直接傳送訊息, 來傳送 MQTT v3 用戶端訊息。MQTT 用戶端可以透過發佈至其他用戶端所訂閱的主題, 彼此傳送訊息。

MQTT 用戶端會訂閱它從 WebSphere MQ 接收的發佈。

執行第 70 頁的『[從 IBM WebSphere MQ Explorer 發佈訊息至 MQTT 用戶端公用程式](#)』作業, 以將發佈從 WebSphere MQ 傳送至 MQTT 用戶端。

MQTT v3 用戶端接收訊息的標準方式是建立主題或主題集的訂閱。在程式碼 Snippet 範例第 69 頁的圖 21 中, MQTT 用戶端使用主題字串 "MQTT Examples" 訂閱。WebSphere MQ C 應用程式第 70 頁的圖 22 使用主題字串 "MQTT Examples" 發佈至主題。在程式碼 Snippet 第 70 頁的圖 23 中, MQTT 用戶端會在回呼方法 `messageArrived` 中接收發佈。

如需如何配置 WebSphere MQ 以傳送發佈以回應 MQTT 用戶端訂閱的進一步相關資訊, 請參閱 [發佈訊息以回應 MQTT 用戶端訂閱](#)。

WebSphere MQ 應用程式會將訊息直接傳送至 MQTT 用戶端

執行第 74 頁的『[使用 IBM WebSphere MQ Explorer 將訊息傳送至 MQTT 用戶端](#)』作業，以將訊息直接從 WebSphere MQ 傳送至 MQTT 用戶端。

以此方式傳送至 MQTT 用戶端的訊息稱為自發訊息。MQTT v3 用戶端會將自發的訊息接收為已設定主題名稱的發佈。遙測 (MQXR) 服務會將主題名稱設為遠端佇列名稱。

您無法將自發的訊息傳送至裝置的 WebSphere MQ 常駐程式: 如果常駐程式收到自發的訊息，它可能會關閉。MQTT v3 用戶端無法將自發訊息傳送至另一個 MQTT v3 用戶端，也無法傳送至 WebSphere MQ 佇列。

如需如何配置 WebSphere MQ 以直接傳送訊息至 MQTT 用戶端的進一步相關資訊，請參閱 [直接傳送訊息至用戶端](#)。

MQTT 用戶端發佈訊息

MQTT v3 用戶端可以發佈由另一個 MQTT v3 用戶端接收的訊息，但它無法傳送自發訊息。程式碼 Snippet 第 70 頁的圖 24 顯示以 Java 撰寫的 MQTT v3 用戶端如何發佈訊息。

將訊息傳送至某個特定 MQTT v3 用戶端的一般型樣，是針對每一個用戶端建立其專屬 ClientIdentifier 的訂閱。執行作業第 76 頁的『[將訊息發佈至特定的 MQTT v3 用戶端](#)』，以使用 ClientIdentifier 作為主題字串，將訊息從某個 MQTT 用戶端發佈至另一個 MQTT 用戶端。

程式碼 Snippet 範例

第 69 頁的圖 21 中的程式碼 Snippet 顯示以 Java 撰寫的 MQTT 用戶端如何建立訂閱。它還需要回呼方法 `messageArrived` 才能接收訂閱的發佈。程式碼 Snippet 擷取自作業 [Creating a subscriber for MQ Telemetry Transport using Java](#)。

```
String    clientId = String.format("%-23.23s",
                                System.getProperty("user.name") + "_" +
                                (UUID.randomUUID().toString()).trim()).replace('-', '_');
MqttClient client = new MqttClient("localhost", clientId);
String topicString = "MQTT Examples";
int       QoS = 1;
client.subscribe(topicString, QoS);
```

圖 21: MQTT v3 用戶端訂閱者

第 70 頁的圖 22 中的程式碼 Snippet 顯示以 C 撰寫的 WebSphere MQ 應用程式如何傳送發佈。程式碼 Snippet 擷取自作業 [Create a publisher to a variable topic](#)

```

/* Define and set variables to defaults */
/* Omitted lines declaring variables */
char * topicName = ""
char * topicString = "MQTT Examples"
char * publication = "Hello world!";
do {
    MQCONN(qMgrName, &Hconn, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    td.ObjectType = MQOT_TOPIC; /* Object is a topic */
    td.Version = MQOD_VERSION_4; /* Descriptor needs to be V4 */
    strncpy(td.ObjectName, topicName, MQ_TOPIC_NAME_LENGTH);
    td.ObjectString.VSPtr = topicString;
    td.ObjectString.VSLength = (MQLONG)strlen(topicString);
    MQOPEN(Hconn, &td, MQOO_OUTPUT | MQOO_FAIL_IF QUIESCING, &Hobj, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    pmo.Options = MQPMO_FAIL_IF QUIESCING | MQPMO_RETAIN;
    MQPUT(Hconn, Hobj, &md, &pmo, (MQLONG)strlen(publication)+1, publication, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    MQCLOSE(Hconn, &Hobj, MQCO_NONE, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    MQDISC(&Hconn, &CompCode, &Reason);
} while (0);

```

圖 22: WebSphere MQ 發佈者

當發佈到達時，MQTT 用戶端會呼叫 MQTT 應用程式用戶端 `MqttCallback` 類別的 `messageArrived` 方法。程式碼 Snippet 擷取自作業 [Creating a subscriber for MQ Telemetry Transport using Java](#)。

```

public class CallBack implements MqttCallback {
    public void messageArrived(MqttTopic topic, MqttMessage message) {
        try {
            System.out.println("Message arrived: \" + message.toString()
                + "\" on topic \" + topic.toString() + "\"");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    // ... Other callback methods
}

```

圖 23: `messageArrived` 方法

第 70 頁的圖 24 顯示 MQTT v3 將訊息發佈至第 69 頁的圖 21 中所建立的訂閱。程式碼 Snippet 擷取自 [使用 Java 建立第一個 MQ Telemetry Transport Publisher 應用程式作業](#)。

```

String address = "localhost";
String clientId = String.format("%-23.23s",
    System.getProperty("user.name") + "_" +
    (UUID.randomUUID().toString()).trim()).replace('-', '_');
MqttClient client = new MqttClient(address, clientId);
String topicString = "MQTT Examples";
MqttTopic topic = client.getTopic(Example.topicString);
String publication = "Hello world";
MqttMessage message = new MqttMessage(publication.getBytes());
MqttDeliveryToken token = topic.publish(message);

```

圖 24: MQTT v3 用戶端發佈者

從 IBM WebSphere MQ Explorer 發佈訊息至 MQTT 用戶端公用程式

請遵循此作業中的步驟，使用 IBM WebSphere MQ Explorer 來發佈訊息，並使用 MQTT 用戶端公用程式來訂閱它。還有一項作業顯示如何配置佇列管理程式別名，而不是將預設傳輸佇列設為 `SYSTEM.MQTT.TRANSMIT.QUEUE`。

開始之前

作業假設您熟悉 IBM WebSphere MQ 和 IBM WebSphere MQ Explorer，且已安裝 IBM WebSphere MQ 和 IBM WebSphere MQ Telemetry 特性。

為這項作業建立佇列管理程式資源的使用者必須具備足夠的權限才能這麼做。為了示範，會假設 IBM WebSphere MQ Explorer 使用者 ID 是 mqm 群組的成員。

關於這項作業

在作業中，您在 IBM WebSphere MQ 中建立主題，並使用 MQTT 用戶端公用程式訂閱該主題。當您使用 IBM WebSphere MQ Explorer 發佈至主題時，MQTT 用戶端會接收發佈。

程序

執行下列其中一個作業：

- 您已安裝 WebSphere MQ Telemetry，但尚未啟動它。執行下列作業：[第 71 頁的『在尚未定義遙測 \(MQXR\) 服務的情況下啟動作業』](#)。
- 您之前已執行 WebSphere MQ 遙測，但想要使用新的佇列管理程式來執行示範。執行作業：[第 71 頁的『在尚未定義遙測 \(MQXR\) 服務的情況下啟動作業』](#)。
- 您想要使用未定義遙測資源的現有佇列管理程式來執行作業。您不想執行「**定義配置範例**」精靈。
 - a. 請執行下列其中一項作業來設定遙測：
 - 在 Linux 和 AIX 上配置遙測的佇列管理程式
 - 在 Windows 上配置遙測的佇列管理程式
 - b. 執行作業：[第 72 頁的『使用執行中遙測 \(MQXR\) 服務啟動作業』](#)
- 如果您想要使用已定義遙測資源的現有佇列管理程式來執行作業，請執行下列作業：[第 72 頁的『使用執行中遙測 \(MQXR\) 服務啟動作業』](#)。

下一步

執行 [第 74 頁的『使用 IBM WebSphere MQ Explorer 將訊息傳送至 MQTT 用戶端』](#)，將訊息直接傳送至用戶端公用程式。

相關工作

[使用 IBM WebSphere MQ Explorer 將訊息傳送至 MQTT 用戶端](#)

使用 IBM WebSphere MQ Explorer 將訊息放入 IBM WebSphere MQ 佇列，以將訊息傳送至 MQTT 用戶端公用程式。此作業顯示如何配置遠端佇列定義，以將訊息直接傳送至 MQTT 用戶端。

[將訊息發佈至特定的 MQTT v3 用戶端](#)

使用 ClientIdentifier 作為主題名稱，並使用 IBM WebSphere MQ 作為發佈/訂閱分配管理系統，將訊息從一個 MQTT v3 用戶端發佈至另一個 MQTT 用戶端。使用適用於裝置的 WebSphere MQ Telemetry 常駐程式作為發佈/訂閱分配管理系統，來重複此作業。

[在尚未定義遙測 \(MQXR\) 服務的情況下啟動作業](#)

建立佇列管理程式並執行 **定義配置範例**，以定義佇列管理程式的遙測資源範例。使用 IBM WebSphere MQ Explorer 發佈訊息，並使用 MQTT 用戶端公用程式訂閱該訊息。

關於這項作業

當您使用 **定義配置範例** 來設定遙測資源範例時，精靈會設定訪客使用者 ID 許可權。請仔細考量您是否要以這種方式授權來賓使用者 ID。guest (在 Windows 上) 及 nobody (在 Linux 上) 會被授與發佈及訂閱主題樹狀結構根目錄的許可權，以及將訊息放置在 SYSTEM.MQTT.TRANSMIT.QUEUE 上的許可權。

精靈也會將預設傳輸佇列設為 SYSTEM.MQTT.TRANSMIT.QUEUE，這可能會干擾在現有佇列管理程式上執行的應用程式。可以配置遙測，但不可以使用預設傳輸佇列；請執行下列作業：[第 73 頁的『使用佇列管理程式別名』](#)。在這項作業中，您將建立佇列管理程式，以避免干擾任何現有的預設傳輸佇列。

程序

1. 使用 IBM WebSphere MQ Explorer，建立並啟動新的佇列管理程式。
 - a) 用滑鼠右鍵按一下 Queue Managers 資料夾 > **新建** > **佇列管理程式 ...**。輸入佇列管理程式名稱 > **完成**。
組成佇列管理程式名稱; 例如, MQTTMGR。
2. 建立並啟動遙測 (MQXR) 服務，並建立遙測通道範例。
 - a) 開啟 Queue Managers\QmgrName\Telemetry 資料夾。
 - b) 按一下 **定義配置範例 ...** > **完成**
維持勾選 **啟動 MQTT 用戶端公用程式** 勾選框。
3. 使用 MQTT 用戶端公用程式建立 MQTT Example 的訂閱。
 - a) 按一下 **連接**。
用戶端歷程 會記錄 Connected 事件。
 - b) 在 **訂閱\主題** 欄位 > **訂閱** 中鍵入 MQTT Example。
用戶端歷程 會記錄 Subscribed 事件。
4. 在 WebSphere MQ 中建立 MQTTExampleTopic。
 - a) 用滑鼠右鍵按一下 WebSphere MQ 探險家 > **新建** > **主題** 中的 Queue Managers\QmgrName\Topics 資料夾。
 - b) 輸入 MQTTExampleTopic 作為 **名稱** > **下一步**。
 - c) 輸入 MQTT Example 作為 **主題字串** > **完成**。
 - d) 按一下 **確定** 以關閉確認視窗。
5. 使用 IBM WebSphere MQ Explorer 將 Hello World! 發佈至主題 MQTT Example。
 - a) 按一下 IBM WebSphere MQ Explorer 中的 Queue Managers\QmgrName\Topics 資料夾。
 - b) 用滑鼠右鍵按一下 MQTTExampleTopic > **測試發佈 ...**
 - c) 在 **訊息資料** 欄位 > **發佈訊息** > 切換至「**MQTT 用戶端公用程式**」視窗中鍵入 Hello World!。
用戶端歷程 會記錄 Received 事件。

使用執行中遙測 (MQXR) 服務啟動作業

建立遙測通道及主題。授權使用者使用主題及遙測傳輸佇列。使用 IBM WebSphere MQ Explorer 發佈訊息，並使用 MQTT 用戶端公用程式訂閱該訊息。

開始之前

在此版本的作業中，佇列管理程式 *QmgrName* 已定義且在執行中。遙測 (MQXR) 服務已定義且在執行中。遙測 (MQXR) 服務可能已手動建立，或透過執行「**定義配置範例**」精靈來建立。

關於這項作業

在此作業中，您可以配置現有的佇列管理程式，以將發佈資訊傳送至 MQTT 用戶端公用程式。

作業的步驟 [第 72 頁的『1』](#) 會將預設傳輸佇列設為 SYSTEM.MQTT.TRANSMIT.QUEUE，這可能會干擾在現有佇列管理程式上執行的應用程式。可以配置遙測，但不可以使用預設傳輸佇列; 請執行下列作業: [第 73 頁的『使用佇列管理程式別名』](#)。

程序

1. 將 SYSTEM.MQTT.TRANSMIT.QUEUE 設為預設傳輸佇列。
 - a) 用滑鼠右鍵按一下 Queue Managers\QmgrName folder > **內容 ...**
 - b) 在導覽器中按一下 **通訊**。
 - c) 按一下 **選取 ...** > 選取 SYSTEM.MQTT.TRANSMIT.QUEUE > **確定** > **確定**。

2. 建立遙測通道 MQTTExampleChannel，以將 MQTT 用戶端公用程式連接至 WebSphere MQ，並啟動 MQTT 用戶端公用程式。
 - a) 用滑鼠右鍵按一下 WebSphere MQ 探險家 > **新建** > **遙測通道 ...** 中的 Queue Managers\QmgrName\Telemetry\Channels 資料夾。
 - b) 在 **通道名稱** 欄位中鍵入 MQTTExampleChannel > **Next** > **Next**。
 - c) 將用戶端授權畫面上的 **固定使用者 ID** 變更為要發佈及訂閱 MQTTExample > **下一步** 的使用者 ID。
 - d) 保持 **啟動用戶端公用程式** 已勾選 > **完成**。
3. 使用 MQTT 用戶端公用程式建立 MQTT Example 的訂閱。
 - a) 按一下 **連接**。
用戶端歷程 會記錄 Connected 事件。
 - b) 在 **訂閱 \ 主題** 欄位 > **訂閱** 中鍵入 MQTT Example。
用戶端歷程 會記錄 Subscribed 事件。
4. 在 WebSphere MQ 中建立 MQTTExampleTopic。
 - a) 用滑鼠右鍵按一下 WebSphere MQ 探險家 > **新建** > **主題** 中的 Queue Managers\QmgrName\Topics 資料夾。
 - b) 輸入 MQTTExampleTopic 作為 **名稱** > **下一步**。
 - c) 輸入 MQTT Example 作為 **主題字串** > **完成**。
 - d) 按一下 **確定** 以關閉確認視窗。
5. 如果您想要使用者 (不在 mqm 群組中) 發佈及訂閱 MQTTExample 主題，請執行下列動作：
 - a) 授權使用者發佈及訂閱主題 MQTTExampleTopic:


```
setmqaut -m QmgrName -t topic -n MQTTExampleTopic -p User ID -all +pub +sub
```
 - b) 授權使用者將訊息放置到 SYSTEM.MQTT.TRANSMIT.QUEUE:


```
setmqaut -m QmgrName -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p User ID -all +put
```
6. 使用 IBM WebSphere MQ Explorer 將 Hello World! 發佈至主題 MQTT Example。
 - a) 按一下 IBM WebSphere MQ Explorer 中的 Queue Managers\QmgrName\Topics 資料夾。
 - b) 用滑鼠右鍵按一下 MQTTExampleTopic > **測試發佈 ...**
 - c) 在 **訊息資料** 欄位 > **發佈訊息** > 切換至「**MQTT 用戶端公用程式**」視窗中鍵入 Hello World!。
用戶端歷程 會記錄 Received 事件。

使用佇列管理程式別名

使用 IBM WebSphere MQ Explorer 將訊息發佈至 MQTT 用戶端公用程式，而不將預設傳輸佇列設為 SYSTEM.MQTT.TRANSMIT.QUEUE。

此作業是前一項作業的延續，並使用佇列管理程式別名來避免將預設傳輸佇列設為 SYSTEM.MQTT.TRANSMIT.QUEUE。

開始之前

完成作業 第 71 頁的『[在尚未定義遙測 \(MQXR\) 服務的情況下啟動作業](#)』或作業 第 72 頁的『[使用執行中遙測 \(MQXR\) 服務啟動作業](#)』。

關於這項作業

當 MQTT 用戶端建立訂閱時，IBM WebSphere MQ 會使用 ClientIdentifier 作為遠端佇列管理程式名稱來傳送其回應。在此作業中，它會使用 ClientIdentifier(MyClient)。

如果沒有稱為 MyClient 的傳輸佇列或佇列管理程式別名，則會將回應放置在預設傳輸佇列上。透過將預設傳輸佇列設為 SYSTEM.MQTT.TRANSMIT.QUEUE，MQTT 用戶端會取得回應。

您可以使用佇列管理程式別名，避免將預設傳輸佇列設為 `SYSTEM.MQTT.TRANSMIT.QUEUE`。您必須為每一個 `ClientIdentifier` 設定佇列管理程式別名。一般而言，用戶端太多，無法實際使用佇列管理程式別名。通常無法預期 `ClientIdentifier`，因此無法以這種方式配置遙測。

不過，在某些情況下，您可能必須將預設傳輸佇列配置成 `SYSTEM.MQTT.TRANSMIT.QUEUE` 以外的其他項目。程序中的步驟會配置佇列管理程式別名，而不是將預設傳輸佇列設為 `SYSTEM.MQTT.TRANSMIT.QUEUE`。

程序

1. 移除 `SYSTEM.MQTT.TRANSMIT.QUEUE` 作為預設傳輸佇列。
 - a) 用滑鼠右鍵按一下 `Queue Managers\QmgrName folder > 內容 ...`
 - b) 在導覽器中按一下 **通訊**。
 - c) 從 **預設傳輸佇列** 欄位 > **確定** 中移除 `SYSTEM.MQTT.TRANSMIT.QUEUE`。
2. 確認您無法再使用 MQTT 用戶端公用程式來建立訂閱：
 - a) 按一下 **連接**。
用戶端歷程 會記錄 `Connected` 事件。
 - b) 在 **訂閱 \ 主題** 欄位 > **訂閱** 中鍵入 `MQTT Example`。
用戶端歷程 會記錄 `Subscribe failed` 及 `Connection lost` 事件。
3. 建立 `ClientIdentifier` 的佇列管理程式別名 `MyClient`。
 - a) 用滑鼠右鍵按一下 `Queue Managers\QmgrName\Queues` 資料夾 > **新建** > **遠端佇列定義**。
 - b) 將定義命名為 `MyClient` > **Next**。
 - c) 在 **遠端佇列管理程式** 欄位中輸入 `MyClient`。
 - d) 在 **傳輸佇列** 欄位中鍵入 `SYSTEM.MQTT.TRANSMIT.QUEUE` > **完成**。
4. 重新連接 MQTT 用戶端公用程式。
 - a) 檢查 **用戶端 ID** 是否設為 `MyClient`。
 - b) **CONNECT**
用戶端歷程 會記錄 `Connected` 事件。
5. 使用 MQTT 用戶端公用程式建立 `MQTT Example` 的訂閱。
 - a) 按一下 **連接**。
用戶端歷程 會記錄 `Connected` 事件。
 - b) 在 **訂閱 \ 主題** 欄位 > **訂閱** 中鍵入 `MQTT Example`。
用戶端歷程 會記錄 `Subscribed` 事件。
6. 使用 IBM WebSphere MQ Explorer 將 `Hello World!` 發佈至主題 `MQTT Example`。
 - a) 按一下 IBM WebSphere MQ Explorer 中的 `Queue Managers\QmgrName\Topics` 資料夾。
 - b) 用滑鼠右鍵按一下 `MQTTExampleTopic` > **測試發佈 ...**
 - c) 在 **訊息資料** 欄位 > **發佈訊息** > 切換至「**MQTT 用戶端公用程式**」視窗中鍵入 `Hello World!`。
用戶端歷程 會記錄 `Received` 事件。

使用 IBM WebSphere MQ Explorer 將訊息傳送至 MQTT 用戶端

使用 IBM WebSphere MQ Explorer 將訊息放入 IBM WebSphere MQ 佇列，以將訊息傳送至 MQTT 用戶端公用程式。此作業顯示如何配置遠端佇列定義，以將訊息直接傳送至 MQTT 用戶端。

開始之前

執行作業第 70 頁的『從 IBM WebSphere MQ Explorer 發佈訊息至 MQTT 用戶端公用程式』。保持 MQTT 用戶端公用程式已連接。

關於這項作業

此作業示範使用佇列而非發佈至主題，將訊息傳送至 MQTT 用戶端。您未在用戶端中建立訂閱。作業的步驟 [第 75 頁的『2』](#) 會示範已刪除先前的訂閱。

程序

1. 中斷連線並重新連接 MQTT 用戶端公用程式，以捨棄任何現有的訂閱。

捨棄訂閱是因為除非您變更預設值，否則 MQTT 用戶端公用程式會使用全新階段作業進行連接；請參閱 [清除階段作業](#)。

若要让執行作業更容易，請鍵入您自己的 ClientIdentifier，而不是使用 MQTT 用戶端公用程式所建立的所產生 ClientIdentifier。

- a) 按一下 **中斷連線**，以中斷 MQTT 用戶端公用程式與遙測通道的連線。

用戶端歷程 會記錄 Disconnected 事件

- b) 將 **用戶端 ID** 變更為 MyClient。

- c) 按一下 **連接**。

用戶端歷程 會記錄 Connected 事件

2. 檢查 MQTT 用戶端公用程式是否不再接收 MQTTExampleTopic 的發佈。

- a) 按一下 IBM WebSphere MQ Explorer 中的 Queue Managers\QmgrName\Topics 資料夾。

- b) 用滑鼠右鍵按一下 MQTTExampleTopic > **測試發佈 ...**

- c) 在 **訊息資料** 欄位 > **發佈訊息** > 切換至「**MQTT 用戶端公用程式**」視窗中鍵入 Hello World!。

用戶端歷程中未記錄任何事件。

3. 建立用戶端的遠端佇列定義。

將 ClientIdentifier MyClient 設為遠端佇列定義中的遠端佇列管理程式名稱。使用您喜歡的任何名稱作為遠端佇列名稱。遠端佇列名稱會傳遞至 MQTT 用戶端作為主題名稱。

- a) 用滑鼠右鍵按一下 Queue Managers\QmgrName\Queues 資料夾 > **新建** > **遠端佇列定義**。

- b) 將定義命名為 MyClientRemoteQueue > **Next**。

- c) 在 **遠端佇列** 欄位中鍵入 MQTTExampleQueue。

- d) 在 **遠端佇列管理程式** 欄位中輸入 MyClient。

- e) 在 **傳輸佇列** 欄位中鍵入 SYSTEM.MQTT.TRANSMIT.QUEUE > **完成**。

4. 將測試訊息放置在 MyClientRemoteQueue 上。

- a) 用滑鼠右鍵按一下 **MyClientRemoteQueue** > **放置測試訊息 ...**

- b) 在「**訊息資料**」欄位中鍵入 Hello queue! > **Put message** > **Close**

用戶端歷程 會記錄 Received 事件。

5. 移除 SYSTEM.MQTT.TRANSMIT.QUEUE 作為預設傳輸佇列。

- a) 用滑鼠右鍵按一下 Queue Managers\QmgrName folder > **內容 ...**

- b) 在導覽器中按一下 **通訊**。

- c) 從 **預設傳輸佇列** 欄位 > **確定** 中移除 SYSTEM.MQTT.TRANSMIT.QUEUE。

6. 重做步驟 [第 75 頁的『4』](#)。

MyClientRemoteQueue 是明確命名傳輸佇列的遠端佇列定義。您不需要定義預設傳輸佇列，即可將訊息傳送至 MyClient。

下一步

當預設傳輸佇列不再設為 SYSTEM.MQTT.TRANSMIT.QUEUE 時，除非為 ClientIdentifier 定義佇列管理程式別名 (MyClient)，否則「MQTT 用戶端公用程式」無法建立新的訂閱。將預設傳輸佇列還原至 SYSTEM.MQTT.TRANSMIT.QUEUE。

相關工作

從 IBM WebSphere MQ Explorer 發佈訊息至 MQTT 用戶端公用程式

請遵循此作業中的步驟，使用 IBM WebSphere MQ Explorer 來發佈訊息，並使用 MQTT 用戶端公用程式來訂閱它。還有一項作業顯示如何配置佇列管理程式別名，而不是將預設傳輸佇列設為 SYSTEM.MQTT.TRANSMIT.QUEUE。

將訊息發佈至特定的 MQTT v3 用戶端

使用 ClientIdentifier 作為主題名稱，並使用 IBM WebSphere MQ 作為發佈/訂閱分配管理系統，將訊息從一個 MQTT v3 用戶端發佈至另一個 MQTT 用戶端。使用適用於裝置的 WebSphere MQ Telemetry 常駐程式作為發佈/訂閱分配管理系統，來重複此作業。

將訊息發佈至特定的 MQTT v3 用戶端

使用 ClientIdentifier 作為主題名稱，並使用 IBM WebSphere MQ 作為發佈/訂閱分配管理系統，將訊息從一個 MQTT v3 用戶端發佈至另一個 MQTT 用戶端。使用適用於裝置的 WebSphere MQ Telemetry 常駐程式作為發佈/訂閱分配管理系統，來重複此作業。

開始之前

執行作業 第 70 頁的『從 IBM WebSphere MQ Explorer 發佈訊息至 MQTT 用戶端公用程式』。保持 MQTT 用戶端公用程式已連接。

關於這項作業

此作業示範兩件事：

1. 訂閱某個 MQTT 用戶端中的主題，並從另一個 MQTT 用戶端接收發佈。
2. 使用 ClientIdentifier 作為主題字串來設定「點對點」訂閱。

另一項作業 第 77 頁的『使用適用於裝置的 WebSphere MQ Telemetry 常駐程式作為發佈/訂閱分配管理系統』使用適用於裝置的 WebSphere MQ Telemetry 常駐程式作為發佈/訂閱分配管理系統，而不是 WebSphere MQ。

程序

1. 中斷連線並重新連接 MQTT 用戶端公用程式，以捨棄任何現有的訂閱。

捨棄訂閱是因為除非您變更預設值，否則 MQTT 用戶端公用程式會使用全新階段作業進行連接；請參閱 [清除階段作業](#)。

若要讓執行作業更容易，請鍵入您自己的 ClientIdentifier，而不是使用 MQTT 用戶端公用程式所建立的所產生 ClientIdentifier。

- a) 按一下 **中斷連線**，以中斷 MQTT 用戶端公用程式與遙測通道的連線。

用戶端歷程 會記錄 Disconnected 事件

- b) 將 **用戶端 ID** 變更為 MyClient。
- c) 按一下 **連接**。

用戶端歷程 會記錄 Connected 事件

2. 建立主題 MyClient 的訂閱

MyClient 是此用戶端的 ClientIdentifier。

- a) 在 **訂閱 \ 主題 欄位 > 訂閱** 中鍵入 MyClient。

用戶端歷程 會記錄 Subscribed 事件。

3. 啟動另一個 MQTT 用戶端公用程式。

- a) 開啟 Queue Managers \ QmgrName \ Telemetry \ channels 資料夾。
- b) 用滑鼠右鍵按一下 **PlainText 通道 > 執行 MQTT 用戶端公用程式 ...**
- c) 按一下 **連接**。

用戶端歷程 會記錄 Connected 事件

4. 將 Hello MyClient! 發佈至主題 MyClient。
 - a) 從使用 ClientIdentifier(MyClient) 執行的 MQTT 用戶端公用程式中複製訂閱主題 MyClient。
 - b) 將 MyClient 貼至每一個 MQTT 用戶端公用程式實例的 **Publication \ Topic** 欄位。
 - c) 在 **Publication \ message** 欄位中鍵入 Hello MyClient!。
 - d) 在兩個實例中按一下 **發佈**。

結果

MQTT 用戶端公用程式中具有 ClientIdentifier MyClient 的用戶端歷程會記錄兩個 **Received** 事件及一個 **Published** 事件。其他 MQTT 用戶端公用程式實例會記錄一個 **已發佈** 事件。

如果您只看到一個 **已接收** 事件，請檢查下列可能的原因：

1. 佇列管理程式的預設傳輸佇列是否設為 SYSTEM.MQTT.TRANSMIT.QUEUE?
2. 您是否已在執行其他練習時建立佇列管理程式別名或參照 MyClient 的遠端佇列定義? 如果您有配置問題，請刪除任何參照 MyClient 的資源，例如佇列管理程式別名或傳輸佇列。中斷用戶端公用程式的連線，停止並重新啟動遙測 (MQXR) 服務。

下一步

執行下一項作業第 77 頁的『[使用適用於裝置的 WebSphere MQ Telemetry 常駐程式作為發佈/訂閱分配管理系統](#)』。MQTT 用戶端公用程式會連接至適用於裝置的 WebSphere MQ Telemetry 常駐程式，而不是遙測通道。

相關工作

從 IBM WebSphere MQ Explorer 發佈訊息至 MQTT 用戶端公用程式

請遵循此作業中的步驟，使用 IBM WebSphere MQ Explorer 來發佈訊息，並使用 MQTT 用戶端公用程式來訂閱它。還有一項作業顯示如何配置佇列管理程式別名，而不是將預設傳輸佇列設為 SYSTEM.MQTT.TRANSMIT.QUEUE。

使用 IBM WebSphere MQ Explorer 將訊息傳送至 MQTT 用戶端

使用 IBM WebSphere MQ Explorer 將訊息放入 IBM WebSphere MQ 佇列，以將訊息傳送至 MQTT 用戶端公用程式。此作業顯示如何配置遠端佇列定義，以將訊息直接傳送至 MQTT 用戶端。

使用適用於裝置的 *WebSphere MQ Telemetry* 常駐程式作為發佈/訂閱分配管理系統

使用裝置的 WebSphere MQ Telemetry 常駐程式作為發佈/訂閱分配管理系統，而不是 WebSphere MQ。透過使用其 ClientIdentifier 作為主題字串來訂閱，使用 MQTT 用戶端公用程式的一個實例來發佈訊息，以傳送至另一個實例。

開始之前

安裝常駐程式 (如果您尚未這麼做的話)。

請勿執行驗證; 它會使用埠 1883，該埠已由 PlainText 遙測通道使用。

關於這項作業

在作業中，您可以使用非預設 TCP/IP 埠，將 MQTT 用戶端公用程式連接至適用於裝置的 WebSphere MQ Telemetry 常駐程式。一個用戶端使用其 ClientIdentifier 作為主題字串來訂閱，而另一個用戶端則以與前一個作業完全相同的方式發佈至 ClientIdentifier，請參閱 [程序](#)。

註: 記載在 Windows 上執行常駐程式的作業。若要在 Linux 上執行常駐程式，請修改 amqtdd 的路徑及許可權。

程序

1. 在包含適用於裝置的 WebSphere MQ Telemetry 常駐程式的目錄中開啟指令視窗。

Windows 的目錄路徑為 *WebSphere MQ installation directory\mqxr\SDK\advanced\DeviceDaemon\windows_ia32*

2. 在不同的 TCP/IP 埠上執行常駐程式。
 - a) 在與常駐程式相同的目錄中建立稱為 `amqtdc.cfg` 的檔案。
 - b) 在檔案中新增一行，為常駐程式配置不同的預設埠。

```
port 1884
```

- c) 儲存檔案。
3. 啟動常駐程式。

```
amqtdc
```

常駐程式會將其主控台日誌寫入指令視窗：

```
20100712 123133.857 CWNAN9999I IBM WebSphere MQ Telemetry daemon for devices
20100712 123133.857 CWNAN9997I Licensed Materials - Property of IBM
20100712 123133.857 CWNAN9996I Copyright IBM Corp. 2007, 2024. All Rights Reserved
20100712 123133.857 CWNAN9995I US Government Users Restricted Rights ...
20100712 123133.857 CWNAN0049I Configuration file name is .\amqtdc.cfg
20100712 123133.873 CWNAN0054I Features included: bridge
20100712 123134.060 CWNAN0014I MQTT protocol starting, listening on port 1884
```

4. 啟動 MQTT 用戶端公用程式的實例。

僅從遙測通道啟動 MQTT 用戶端公用程式，然後您可以連接至常駐程式。或者，您也可以安裝 IBM WebSphere MQ SupportPac IA92。SupportPac 可從 [IA92: WBI Brokers- WebSphere MQ 遙測傳輸的 Java 實作取得](#)。

- a) 開啟 `Queue Managers\QmgrName\Telemetry\channels` 資料夾。
- b) 用滑鼠右鍵按一下 **PlainText** 通道 > 執行 MQTT 用戶端公用程式 ...
- c) 將埠變更為 1884。
- d) 將用戶端 ID 變更為 MyClient。
- e) 按一下連接。

用戶端歷程 會記錄 Connected 事件

5. 建立主題 MyClient 的訂閱

MyClient 是此用戶端的 ClientIdentifier。

- a) 在 訂閱 \ 主題 欄位 > 訂閱 中鍵入 MyClient。

用戶端歷程 會記錄 Subscribed 事件。

6. 啟動另一個 MQTT 用戶端公用程式。

- a) 開啟 `Queue Managers\QmgrName\Telemetry\channels` 資料夾。
- b) 用滑鼠右鍵按一下 **PlainText** 通道 > 執行 MQTT 用戶端公用程式 ...
- c) 將埠變更為 1884。
- d) 按一下連接。

用戶端歷程 會記錄 Connected 事件

7. 將 Hello MyClient! 發佈至主題 MyClient。

- a) 從使用 ClientIdentifier(MyClient) 執行的 MQTT 用戶端公用程式中複製訂閱主題 MyClient。
- b) 將 MyClient 貼至每一個 MQTT 用戶端公用程式實例的 **Publication \ Topic** 欄位。
- c) 在 **Publication \ message** 欄位中鍵入 Hello MyClient!。
- d) 在兩個實例中按一下 發佈。

結果

MQTT 用戶端公用程式中具有 ClientIdentifier MyClient 的用戶端歷程會記錄兩個 **Received** 事件及一個 **Published** 事件。其他 MQTT 用戶端公用程式實例會記錄一個 **已發佈** 事件。

您也可以指令視窗中監視裝置的 WebSphere MQ Telemetry 常駐程式的連線及斷線事件。

從 MQTT 用戶端傳送訊息至 IBM WebSphere MQ 應用程式

透過訂閱主題，IBM WebSphere MQ 應用程式可以從 MQTT v3 用戶端接收訊息。MQTT 用戶端會使用遙測通道連接至 IBM WebSphere MQ，並透過發佈至相同主題，將訊息傳送至 IBM WebSphere MQ 應用程式。

執行作業第 79 頁的『從 MQTT 用戶端發佈訊息至 IBM WebSphere MQ Explorer』，以瞭解如何將發佈從 MQTT 用戶端傳送至 IBM WebSphere MQ 中定義的訂閱。

如果主題已叢集化，或使用發佈/訂閱階層進行配送，則訂閱可以位於與 MQTT 用戶端所連接之佇列管理程式不同的佇列管理程式上。

從 MQTT 用戶端發佈訊息至 IBM WebSphere MQ Explorer

使用「WebSphere MQ 探險家」建立主題訂閱，並使用 WebSphere MQTT 用戶端公用程式發佈至主題。

開始之前

執行作業第 70 頁的『從 IBM WebSphere MQ Explorer 發佈訊息至 MQTT 用戶端公用程式』。保持 MQTT 用戶端公用程式已連接。

關於這項作業

此作業示範使用 MQTT 用戶端發佈訊息，以及使用未受管理的可延續訂閱 (使用「WebSphere MQ 探險家」建立) 來接收發佈。

程序

1. 建立主題字串 MQTT Example 的可延續訂閱。請執行下列一項程序：

- 執行 [結果](#) 中說明的指令 Script
- 執行下列步驟，以使用「WebSphere MQ 探險家」來建立佇列及訂閱。
 - a) 用滑鼠右鍵按一下 WebSphere MQ 探險家 > **新建** > **本端佇列 ...** 中的 Queue Managers \ *QmgrName* \ Queues 資料夾。
 - b) 輸入 MQTTExampleQueue 作為佇列名稱 > **完成**。
 - c) 用滑鼠右鍵按一下 WebSphere MQ 探險家 > **新建** > **訂閱 ...** 中的 Queue Managers \ *QmgrName* \ Subscriptions 資料夾。
 - d) 輸入 MQTTExampleSubscription 作為佇列名稱 > **下一步**。
 - e) 按一下 **選取 ...** > MQTTExampleTopic > **確定**。

您已在第 70 頁的『從 IBM WebSphere MQ Explorer 發佈訊息至 MQTT 用戶端公用程式』的步驟第 72 頁的『4』中建立主題 MQTTExampleTopic。

- f) 輸入 MQTTExampleQueue 作為目的地名稱 > **完成**。

2. 作為選用步驟，在沒有 mqm 權限的情況下，將佇列設定為供不同使用者使用。

如果您要為權限低於 mqm 的使用者設定配置，則必須將 put 及 get 權限授與 MQTTExampleQueue。已在第 70 頁的『從 IBM WebSphere MQ Explorer 發佈訊息至 MQTT 用戶端公用程式』中配置對主題及傳輸佇列的存取權。

- a) 授權使用者放置並進入佇列 MQTTExampleQueue:

```
setmqaut -m qMgrName -t queue -n MQTTExampleQueue -p User ID -all +put +get
```

3. 使用 MQTT 用戶端公用程式將 Hello WebSphere MQ! 發佈至主題 MQTT Example。

如果您尚未連接 MQTT 用戶端公用程式，請用滑鼠右鍵按一下 **PlainText** 通道 > 執行 MQTT 用戶端公用程式 ... > 連接。

- a) 在 **Publication \ Topic** 欄位中鍵入 MQTT Example。
 - b) 在 **發佈 \ 訊息** 欄位 > **發佈** 中鍵入 Hello WebSphere MQ!。
4. 開啟 Queue Managers \ QmgrName \ Queues 資料夾並尋找 MQTTExampleQueue。
- 現行佇列深度 欄位為 1
5. 用滑鼠右鍵按一下 MQTTExampleQueue > 瀏覽訊息 ... 並檢查該出版品。

在用於裝置的 IBM WebSphere MQ Telemetry 常駐程式與 IBM WebSphere MQ 之間傳送訊息

執行此作業，以瞭解如何將指令傳送至裝置的 IBM WebSphere MQ Telemetry 常駐程式。您撰寫的指令會建立一個橋接器，將訊息從 IBM WebSphere MQ 傳送至常駐程式，並將訊息從常駐程式傳送至 IBM WebSphere MQ。

開始之前

執行作業 [第 76 頁的『將訊息發佈至特定的 MQTT v3 用戶端』](#) 和 [第 77 頁的『使用適用於裝置的 WebSphere MQ Telemetry 常駐程式作為發佈/訂閱分配管理系統』](#)，以熟悉如何使用 MQTT 用戶端公用程式。當您完成作業時，請讓 MQTT 用戶端公用程式的一個實例連接至裝置的遙測常駐程式。保留另一個實例連接至遙測通道。

此作業假設您已定義遙測服務的通道，該服務會接聽位址 127.0.0.1 上的埠 1883。同樣地，預設常駐程式接聽器配置為接聽位址 127.0.0.1 上的埠 1884。amqtdc.cfg 檔案中的單行 (儲存在與常駐程式 amqtdc 相同的目錄中) 會配置預設常駐程式接聽器埠。

```
port 1884
```

關於這項作業

在此作業中，您將更新執行中常駐程式，以建立與 WebSphere MQ 遙測 (MQXR) 服務的連線橋接器，然後與常駐程式交換訊息。

提示: 更新檔案 amqtdc.upd 在使用之後會被常駐程式刪除。若要保留您建立的指令以供稍後使用，您可能想要在不同的檔案中建立指令，然後將它們傳送至 amqtdc.upd。

程序

1. 請確定您有兩個 MQTT 用戶端公用程式實例在執行中。其中一個連接至埠 1884 上的常駐程式，另一個連接至埠 1883 上執行的遙測通道。
2. 使用檔案中的下列指令，在與常駐程式 amqtdc 相同的目錄中建立檔案 amqtdc.upd。

```
connection daemon1
address 127.0.0.1:1883
topic # in import/ export/
topic # out export/ import/
try_private false
```

- 橋接器稱為 daemon1，它會連接至針對在 Socket 位址 127.0.0.1:1883 執行之遙測 (MQXR) 服務所配置的通道。try_private 指令是選用的; true 是預設值。如果沒有這一行，橋接器會先嘗試使用 WebSphere MQ Telemetry 常駐程式瞭解的專用通訊協定來連接裝置。在指令中包括 try_private false 可避免此步驟，並加快完成成功連線的時間。
- 這一行 topic # in import/ export/ 指示 daemon1 訂閱與佇列管理程式中建立的主題字串 export/# 相符的所有主題。它會將相符的發佈從佇列管理程式傳送至常駐程式，並將主題字串的開頭從 export/ 變更為 import/。這一行 topic # out export/ import/ 會在本端常駐程式建立訂閱。橋接器會訂閱所有符合常駐程式中所建立之主題字串 export/# 的主題。它會將發佈從常駐程式傳送至佇列管理程式，並將主題字串的開頭從 export/ 變更為 import/。

第 81 頁的圖 25 顯示產生的主控台日誌。

```
CWNAN0124I Starting bridge connection daemon1
CWNAN0133I Bridge connection daemon1 to 127.0.0.1:1883 now established
```

圖 25: 從啟動連線橋接器開始的主控台日誌

3. 在 MQTT 用戶端公用程式的每一個實例中，在 **訂閱/主題:** 輸入欄位 > **訂閱** 中鍵入 `import/#`。
4. 在 MQTT 用戶端公用程式的每一個實例中，在 **Publication/Topic:** 輸入欄位中鍵入 `export/#`。
 - a) 在連接至埠 1883 的 MQTT 用戶端公用程式中，遙測通道的 **發佈/訊息:** 輸入欄位 > **發佈** 中鍵入 `From the queue manager`。
 - b) 在連接至埠 1884 的 MQTT 用戶端公用程式中，遙測常駐程式會在 **Publication/Message:** 輸入欄位 > **Publish** 中鍵入 `From the daemon`。

每一個 MQTT 用戶端公用程式中的用戶端歷程會顯示已從一個分配管理系統傳送至另一個分配管理系統的發佈。

MQTT 發佈/訂閱應用程式

使用主題型發佈/訂閱來撰寫 MQTT 應用程式。

當 MQTT 用戶端連接時，發佈會以任一方向在用戶端與伺服器之間流動。在用戶端發佈資訊時，會從用戶端傳送發佈。當訊息發佈至符合用戶端所建立之訂閱的主題時，會在用戶端接收發佈。

WebSphere MQ 發佈/訂閱分配管理系統會管理 MQTT 用戶端所建立的主題及訂閱。MQTT 用戶端所建立的主題與 WebSphere MQ 應用程式所建立的主題共用相同的主題空間。

符合 MQTT 用戶端訂閱中主題字串的發佈會放置在 `SYSTEM.MQTT.TRANSMIT.QUEUE` 上，並將遠端佇列管理程式名稱設為用戶端的 `ClientIdentifier`。遙測 (MQXR) 服務會將發佈轉遞至建立訂閱的用戶端。它使用已設為遠端佇列管理程式名稱的 `ClientIdentifier` 來識別用戶端。

一般而言，`SYSTEM.MQTT.TRANSMIT.QUEUE` 必須定義為預設傳輸佇列。可以將 MQTT 配置為不使用預設傳輸佇列，但這很麻煩；請參閱 [配置分散式佇列以將訊息傳送至 MQTT 用戶端](#)。

MQTT 用戶端可以建立持續性階段作業；請參閱第 85 頁的『[MQTT stateless 和 stateful 階段作業](#)』。在持續性階段作業中建立的訂閱是可延續的。到達具有持續性階段作業之用戶端的發佈會儲存在 `SYSTEM.MQTT.TRANSMIT.QUEUE` 中，並在重新連接時轉遞至用戶端。

MQTT 用戶端也可以發佈及訂閱保留的發佈；請參閱 [保留的發佈及 MQTT 用戶端](#)。保留發佈主題的訂閱者會接收主題的最新發佈。當訂閱者建立訂閱時，或當它重新連接至先前的階段作業時，會收到保留的發佈。

相關工作

[使用 Java 來建立第一個 MQ Telemetry Transport Publisher 應用程式](#)

[使用 Java 來建立 MQ 遙測傳輸的訂閱者](#)

遙測應用程式

使用 WebSphere MQ 或 WebSphere Message Broker 訊息流程來撰寫遙測應用程式。

使用 JMS、MQI 或其他 WebSphere MQ 程式設計介面，在 WebSphere MQ 中對遙測應用程式進行程式設計。

遙測 (MQXR) 服務會在 MQTT v3 訊息與 WebSphere MQ 訊息之間進行轉換。它會代表 MQTT 用戶端建立訂閱及發佈，並將發佈轉遞至 MQTT 用戶端。發佈是 MQTT v3 訊息的有效負載。有效負載包含 `json-bytes` 格式的訊息標頭和位元組陣列。遙測伺服器會在 MQTT v3 訊息與 WebSphere MQ 訊息之間對映標頭；請參閱第 82 頁的『[WebSphere MQ Telemetry 與佇列管理程式的整合](#)』。

使用 `Publication`、`MQInput` 及 `JMSInput` 節點，在 WebSphere Message Broker 與 MQTT 用戶端之間傳送及接收發佈。

使用訊息流程，您可以使用 HTTP 來整合遙測與網站，以及使用 WebSphere MQ 和 WebSphere Adapters 來整合其他應用程式。

WebSphere MQ Telemetry 會取代 WebSphere Message Broker 第 7 版中的 SCADA 節點。如需如何使用 SCADAInput 和 SCADAOutput 節點將第 6 版移轉至第 7 版 WebSphere Message Broker 訊息流程的相關資訊，請參閱 [從使用 WebSphere Message Broker 第 6 版移轉遙測應用程式以使用 IBM WebSphere MQ Telemetry 及 WebSphere Message Broker 7.0](#)。

WebSphere MQ Telemetry 與佇列管理程式的整合

MQTT 用戶端與 WebSphere MQ 整合成發佈/訂閱應用程式。它可以發佈或訂閱 WebSphere MQ 中的主題、建立新主題或使用現有主題。它會因為 MQTT 用戶端 (包括本身) 或其他 WebSphere MQ 應用程式 (發佈至其訂閱的主題)，而從 WebSphere MQ 接收發佈。套用規則以決定發佈的屬性。

不支援與 WebSphere MQ 提供的主題、發佈、訂閱及訊息相關聯的許多屬性。第 82 頁的『MQTT 用戶端至 WebSphere MQ 發佈/訂閱分配管理系統』和第 83 頁的『WebSphere MQ 至 MQTT 用戶端』說明如何設定發佈資訊的屬性。這些設定視發佈是要前往或來自 WebSphere MQ 發佈/訂閱分配管理系統而定。

在 WebSphere MQ 中，發佈/訂閱主題與管理主題物件相關聯。MQTT 用戶端所建立的主題沒有不同。當 MQTT 用戶端建立發佈資訊的主題字串時，WebSphere MQ 發佈/訂閱分配管理系統會將它與管理主題物件相關聯。分配管理系統會將發佈資訊中的主題字串對映至最接近的管理主題物件母項。此對映與 WebSphere MQ 應用程式的對映相同。如果沒有使用者建立的主題，則發佈主題會對映至 SYSTEM.BASE.TOPIC。套用至發佈資訊的屬性衍生自主題物件。

當 WebSphere MQ 應用程式或管理者建立訂閱時，會命名訂閱。使用「WebSphere MQ 探險家」或使用 `runmqsc` 或 PCF 指令來列出訂閱。所有 MQTT 用戶端訂閱皆已命名。他們會取得下列格式的名稱：
`ClientIdentifier:Topic name`

MQTT 用戶端至 WebSphere MQ 發佈/訂閱分配管理系統

MQTT 用戶端已將發佈傳送至 WebSphere MQ。遙測 (MQXR) 服務會將發佈資訊轉換為 WebSphere MQ 訊息。WebSphere MQ 訊息包含三個部分：

1. MQMD
2. RFH2
3. 訊息

MQMD 內容會設為其預設值，除非在 [第 82 頁的表 3](#) 中註明。

表 3: MQMD		
MQMD 欄位	類型	值
Format	MQCHAR8	MQFMT_RF_HEADER_2
UserIdentifier	MQCHAR12	設為下列其中一項： MqttClient.ClientIdentifier MqttConnectOptions.UserName 由 WebSphere MQ 管理者為遙測通道設定的使用者 ID。
Priority	MQLONG	MQPRI_PRIORITY_AS_Q_DEF (WebSphere MQ 的預設值，不同於預設值為 4 的 JMS。)
Persistence	MQLONG	QoS=0→MQPER_NOT_PERSISTENT QoS=1→MQPER_PERSISTENT QoS=2→MQPER_PERSISTENT

RFH2 標頭不包含 <msd> 資料夾來定義 JMS 訊息的類型。遙測 (MQXR) 服務會將 WebSphere MQ 訊息建立為預設 JMS 訊息。預設 JMS 訊息類型是 `json-bytes` 訊息。應用程式可以存取其他標頭資訊作為訊息內容；請參閱 訊息內容。

RFH2 值的設定如 第 83 頁的表 4 所示。「格式」內容設定在 RFH2 固定標頭中，其他值設定在 RFH2 資料夾中。

RFH2 內容	類型/資料夾	標頭
格式	MQCHAR8	MQFMT_NONE
ClientIdentifier	mqtt/clientId	複製長度為 1...23 位元組的 <code>MqttClient.ClientIdentifier</code> 。
QoS	mqtt/qos	從送入的 MQTT 訊息複製 QoS。
訊息 ID	mqtt/msgid	如果 QoS 是 1 或 2，請從送入的 MQTT 訊息複製 訊息 ID。
MQIsRetained	mmps/Ret	設定是否在設定 RETAIN 內容的情況下傳送原始 MQTT 發佈，並將訊息接收為保留的發佈。
MQTopicString	mmps/Top	MQTT 訊息發佈至其中的主題。

MQTT 發佈中的有效負載會對映至 WebSphere MQ 訊息的內容：

訊息內容	類型	內容
緩衝區	MQBYTE n	來自送入 MQTT 訊息的位元組副本。長度可以為零。

WebSphere MQ 至 MQTT 用戶端

用戶端已訂閱發佈主題。WebSphere MQ 應用程式已發佈至主題，導致 WebSphere MQ 發佈/訂閱分配管理系統將發佈資訊傳送至 MQTT 訂閱者。或者，WebSphere MQ 應用程式已直接將自發的訊息傳送至 MQTT 用戶端。第 83 頁的表 6 說明如何在傳送至 MQTT 用戶端的訊息中設定固定訊息標頭。WebSphere MQ 訊息標頭或任何其他標頭中的任何其他資料都會被捨棄。WebSphere MQ 訊息中的訊息資料會以 MQTT 訊息中的訊息有效負載形式傳送，不會有任何變更。遙測 (MQXR) 服務會將 MQTT 訊息傳送至 MQTT 用戶端。

MQTT 欄位	類型	值
DUP	布林值	設定是 QoS = 1 或 2，且訊息已在前一個傳輸中傳送至此用戶端，且訊息在一段時間之後仍未確認。

表 6: MQTT 固定標頭內容 (繼續)

MQTT 欄位	類型	值
QoS	int	<p>在 WebSphere MQ 中，從發佈/訂閱分配管理系統送出的發佈資訊中設定 QoS 值的方式取決於送入的發佈資訊。它取決於送入的發佈是從 MQTT 用戶端傳送，還是從 WebSphere MQ 應用程式傳送。</p> <p>MQTT 在送入的發佈以及訂閱者所要求的 QoS 中，QoS 的較低值。</p> <p>WebSphere MQ 衍生自送入發佈的 QoS 較低值： MQPER_NOT_PERSISTENT→QoS=0 MQPER_PERSISTENT→QoS=2</p> <p>及訂閱者所要求的 QoS。如果訊息傳送至沒有訂閱的用戶端，依預設會將 QoS 設為 2。用戶端可以使用不同的 QoS 來訂閱 DEFAULT.QoS，以變更此值。</p>
RETAIN	布林值	如果送入的發佈具有保留的內容集，則設定。

第 84 頁的表 7 說明如何在傳送至 MQTT 用戶端的 MQTT 訊息中設定變數訊息標頭。

表 7: MQTT 變數標頭內容

MQTT 欄位	類型	值
Topic name	字串	用來發佈訊息的主題字串。
Message ID	字串	將發佈資訊放在 SYSTEM.MQTT.TRANSMIT.QUEUE 中時，發佈資訊的 MQMD.MsgId 內容的最後 2 個位元組。
Payload	byte[]	將送入發佈的位元組直接複製到發佈/訂閱分配管理系統。長度可以為零。

用於裝置的 Telemetry 常駐程式

適用於裝置的 WebSphere MQ Telemetry 常駐程式是進階 MQTT V3 用戶端應用程式。可以利用它來儲存及轉遞來自其他 MQTT 用戶端的訊息。它可以如 MQTT 用戶端一樣連接至 WebSphere MQ，但是，您也可以將其他 MQTT 用戶端連接至它。您也可以將它連接至其他 Telemetry 常駐程式。

它有四個基本目的：

1. 連接發佈/訂閱網路中的本端 MQTT 用戶端。

您可以連接裝置的感應器與掣動器，做為個別的 MQTT 用戶端，再將其連接到常駐程式。感應器會發佈其量規讀數，而掣動器會訂閱該讀數，並據以修改其行為。讀數會在本端處理。

2. 過濾要將哪些訂閱及訊息發佈到佇列管理程式與裝置。

在前一個範例中，WebSphere Message Broker 訊息流程可能會訂閱常駐程式將讀數發佈至其中的主題。流程會更新網頁並顯示裝置的狀態。

常駐程式可能也會將掣動器建立的訂閱轉遞至佇列管理程式。WebSphere Message Broker 流程會將訊息發佈至 MQTT 用戶端服務掣動器訂閱的主題。MQTT 用戶端會修改裝置設定。

訊息流程可能從使用 WebSphere Message Broker HTTPInput 節點的網頁開始。

3. 集中多部 MQTT 用戶端，匯集成對遙測伺服器的單一連線。

常駐程式會透過單一的 TCP/IP 連線轉遞發佈及訂閱，而不會個別將每部裝置連接至遙測伺服器。常駐程式會減少遙測 (MQXR) 服務所管理的 TCP/IP 連線數。

這些 MQTT 用戶端會連接到此常駐程式。這些用戶端對佇列管理程式而言為隱藏。此常駐程式會代表所有與其連線的用戶端，透過一條連線連接到佇列管理程式。

4. 在裝置與佇列管理程式之間儲存及轉遞訊息

常駐程式負責保護遙測裝置，避免發生與佇列管理程式間之連線時間過於短暫連線的失敗。

裝置可能僅支援「隨發即棄」傳訊。若對佇列管理程式的連線只能斷續使用或不可靠，裝置便無法如預期或可靠地傳輸資訊。

解決方案是使用穩定的本端連線，將裝置連線到常駐程式。常駐程式可以將流入佇列管理程式或從其中流出的訊息，儲存到其記憶體緩衝區中。如此一來即使在連線不穩定的情況下，也可有可靠的服務品質與佇列管理程式之來回傳送訊息。

註：常駐程式不會一直儲存"傳送中"訊息。這些訊息會儲存在記憶體緩衝區中。

MQTT stateless 和 stateful 階段作業

MQTT 用戶端可以使用佇列管理程式建立 stateful 階段作業。stateful MQTT 用戶端中斷連線時，佇列管理程式會保留用戶端建立的訂閱以及進行中的訊息。用戶端重新連接時，它會解析進行中的訊息。它會傳送排入佇列以進行遞送的所有訊息，並接收在其斷線時為其訂閱發佈的所有訊息。

當 MQTT 用戶端連接至遙測通道時，它會啟動新的階段作業，或回復舊的階段作業。新的階段作業沒有尚未確認的未完成訊息，沒有訂閱，也沒有等待遞送的發佈。當用戶端連接時，它會指定是使用全新階段作業啟動，還是回復現有的階段作業；請參閱 [清除階段作業](#)。

如果用戶端回復現有的階段作業，它會繼續進行，好像連線未中斷一樣。等待遞送的發佈會傳送至用戶端，且任何尚未確定的訊息傳送都會完成。當持續性階段作業中的用戶端與遙測 (MQXR) 服務中斷連線時，用戶端建立的任何訂閱都會保留。訂閱的發佈會在用戶端重新連接時傳送至用戶端。如果它重新連接而不回復舊的階段作業，遙測 (MQXR) 服務會捨棄發佈。

階段作業狀態資訊由佇列管理程式儲存在 `SYSTEM.MQTT.PERSISTENT.STATE` 佇列中。

WebSphere MQ 管理者可以中斷連線並清除階段作業。

未連接 MQTT 用戶端時

當用戶端未連接時，佇列管理程式可以繼續代表它接收發佈。當用戶端重新連接時，會將它們轉遞至用戶端。如果用戶端非預期地中斷連線，則用戶端可以建立佇列管理程式代表用戶端發佈的"最後留言"。

如果您想要在用戶端非預期地中斷連線時收到通知，則可以登錄最後留言發佈；請參閱 [最後留言發佈](#)。如果遙測 (MQXR) 服務偵測到用戶端連線中斷，而沒有用戶端要求它，則它會由遙測 (MQXR) 服務傳送。

用戶端可以隨時發佈保留的發佈資訊；請參閱 [保留的發佈資訊及 MQTT 用戶端](#)。主題的新訂閱可以要求傳送與主題相關聯的任何保留發佈資訊。如果您將最後留言建立為保留發佈，則可以使用它來監視用戶端的狀態。

例如，用戶端會在連接時發佈保留的發佈資訊，以廣告其可用性。同時，它會建立保留的最後留言發佈，以宣告其無法使用。此外，在進行計劃斷線之前，它會將其無效性發佈為保留發佈。若要瞭解用戶端是否可用，您可以訂閱保留發佈的主題。您一律會收到三種出版品中的其中一種。

如果用戶端要在斷線時接收發佈的訊息，請將用戶端重新連接至其前一個階段作業；請參閱 [第 85 頁的『MQTT stateless 和 stateful 階段作業』](#)。其訂閱在刪除之前或用戶端建立全新階段作業之前都處於作用中。

MQTT 用戶端與 WebSphere MQ 應用程式之間的鬆散連結

MQTT 用戶端與 WebSphere MQ 應用程式之間的發佈流程鬆散耦合。發佈可能源自 MQTT 用戶端或 WebSphere MQ 應用程式，且沒有設定順序。發佈者和訂閱者是鬆散耦合的。它們透過發佈和訂閱間彼此互動。您也可以從 WebSphere MQ 應用程式，將訊息直接傳送至 MQTT 用戶端。

MQTT 用戶端與 WebSphere MQ 應用程式在兩種意義上鬆散耦合：

1. 發佈者和訂閱者可透過發佈與訂閱與主題的關聯來鬆散結合。發佈者和訂閱者通常不知道發佈或訂閱的其他來源的位址或身分。
2. MQTT 用戶端在個別執行緒上發佈、訂閱、接收發佈及處理程序遞送確認通知。

MQTT 用戶端應用程式不會等到發佈遞送。應用程式會將訊息傳遞至 MQTT 用戶端，然後應用程式會在其自己的執行緒上繼續執行。遞送記號用來將應用程式與發佈的遞送同步化; 請參閱 [遞送記號](#)。

將訊息傳遞至 MQTT 用戶端之後，應用程式可以選擇等待遞送記號。用戶端可以提供在發佈遞送至 WebSphere MQ 時所呼叫的回呼方法，而不是等待。它也可以忽略 `delivery-token`。

視與訊息相關聯的服務品質而定，`delivery-token` 會立即傳回回呼方法，或可能在相當長的時間之後。在用戶端中斷連線並重新連接之後，甚至可能傳回 `delivery-token`。如果服務品質是「發動並忘記」，則會立即傳回遞送記號。在另外兩種情況下，只有在用戶端收到發佈已傳送給訂閱者的確認通知時，才會傳回遞送記號。

由於用戶端訂閱而傳送至 MQTT 用戶端的發佈，會遞送至 `messageArrived` 回呼方法。`messageArrived` 在與主要應用程式不同的執行緒上執行。

將訊息直接傳送至 MQTT 用戶端

您可以使用兩種方式之一，將訊息傳送至特定的 MQTT 用戶端。

1. WebSphere MQ 應用程式可以將訊息直接傳送至 MQTT 用戶端，而不需要訂閱; 請參閱 [直接傳送訊息至用戶端](#)。
2. 替代方法是使用 `ClientIdentifier` 命名慣例。讓所有 MQTT 訂閱者使用其唯一 `ClientIdentifier` 作為主題來建立訂閱。發佈至 `ClientIdentifier`。發佈會傳送至訂閱主題 `ClientIdentifier` 的用戶端。使用此技術，您可以將發佈傳送至特定 MQTT 訂閱者。

WebSphere MQ 遙測安全

維護遙測裝置安全可能很重要，因為裝置可能是可攜式，並會在無法對其進行謹慎控制的位置使用。您可以使用 VPN 來保護從 MQTT 裝置到遙測 (MQXR) 服務的連線。WebSphere MQ Telemetry 提供兩個其他安全機制: SSL 和 JAAS。

SSL 主要用於加密裝置與遙測通道之間的通訊，並鑑別裝置是否連接至正確的伺服器; 請參閱 [使用 SSL 進行遙測通道鑑別](#)。您也可以使用 SSL 來檢查是否允許用戶端裝置連接至伺服器; 請參閱 [使用 SSL 進行 MQTT 用戶端鑑別](#)。

JAAS 主要用來檢查裝置的使用者是否被允許使用伺服器應用程式; 請參閱 [使用密碼進行 MQTT 用戶端鑑別](#)。JAAS 可以與 LDAP 搭配使用，以使用單一登入目錄來檢查密碼。

SSL 和 JAAS 可以一起使用，以提供雙因素鑑別。您可以將 SSL 使用的密碼限制為符合 FIPS 標準的密碼。

至少有數萬個使用者，提供個別安全設定檔並不總是可行。使用設定檔來授權個別使用者存取 WebSphere MQ 物件也不是切實可行的作法。而是將使用者分組至類別，以授權發佈及訂閱主題，並將發佈傳送至用戶端。

配置每一個遙測通道，以將用戶端對映至一般用戶端使用者 ID。針對在特定通道上連接的每個用戶端使用一般使用者 ID; 請參閱 [MQTT 用戶端身分及授權](#)。

授權使用者群組不會危及每個個人的鑑別。每一個個別使用者都可以在用戶端或伺服器上使用其 **使用者名稱** 及 **密碼** 進行鑑別，然後在伺服器上使用一般使用者 ID 進行授權。

WebSphere MQ 遙測全球化

MQTT v3 通訊協定中的訊息有效負載編碼為 `byte-array`。一般而言，處理文字的應用程式會在 UTF-8 中建立訊息有效負載。遙測通道會將訊息有效負載說明為 UTF-8，但不會執行任何字碼頁轉換。發佈主題字串必須是 UTF-8。

應用程式負責將英文字母資料轉換成正確的字碼頁，並將數值資料轉換成正確的數字編碼。

MQTT Java 用戶端具有便利的 `MqttMessage.toString` 方法。此方法會將訊息有效負載視為以本端平台預設字集編碼，通常是 UTF-8。它會將有效負載轉換成 Java 字串。Java 具有字串方法 `getBytes`，可將字串轉換為使用本端平台預設字集編碼的位元組陣列。在具有相同預設字集的平台之間，兩個 MQTT Java 程式會在訊息有效負載中交換文字，在 UTF-8 中這樣容易且有效率。

如果其中一個平台的預設字集不是 UTF-8，則應用程式必須建立交換訊息的慣例。例如，發佈者使用 `getBytes("UTF8")` 方法指定從字串到 UTF-8 的轉換。為了接收訊息文字，訂閱者假設訊息是以 UTF-8 字集編碼。

遙測 (MQXR) 服務說明將來自 MQTT 用戶端訊息的所有送入發佈編碼為 UTF-8。它會設定 `MQMD.CodedCharSetId` 至 UTF-8，`RFH2.CodedCharSetId` 至 `MQCCSI_INHERIT`；請參閱第 82 頁的『[WebSphere MQ Telemetry 與佇列管理程式的整合](#)』。發佈的格式設為 `MQFMT_NONE`，因此通道或 `MQGET` 無法執行任何轉換。

WebSphere MQ Telemetry 的效能及可調整性

在管理大量用戶端及改善 WebSphere MQ Telemetry 的可調整性時，請考量下列因素。

容量規劃

如需 WebSphere MQ Telemetry 的效能報告相關資訊，請從 [WebSphere MQ 系列-效能報告](#) 中選取 WebSphere MQ 「遙測效能評估」報告。

連線

與連線相關的成本包括

- 根據處理器用量和時間來設定連線本身的成本。
- 網路成本。
- 保持連線開啟但不使用它時使用的記憶體。

用戶端保持連線會產生額外負載。如果連線保持開啟狀態，則 TCP/IP 流程及 MQTT 訊息會使用網路來檢查連線是否仍然存在。此外，伺服器中針對保持開啟的每一個用戶端連線使用記憶體。

如果您每分鐘傳送多個訊息，請保持連線開啟，以避免起始新連線的成本。如果您每 10-15 分鐘傳送少於一個訊息，請考量捨棄連線，以避免保持開啟的成本。您可能想要將 SSL 連線保持開啟，但閒置的時間較長，因為設定它更昂貴。

此外，請考量用戶端的功能。如果用戶端上有儲存及轉遞機能，則您可以批次處理訊息，並捨棄傳送批次之間的連線。不過，如果用戶端已斷線，則用戶端無法從伺服器接收訊息。因此，應用程式的目的會影響決策。

如果您的系統有一個用戶端傳送許多訊息 (例如檔案傳送)，請勿等待每個訊息的伺服器回應。相反地，請傳送所有訊息，並在結束時確認已收到所有訊息。或者，使用 [服務品質 \(QoS\)](#)。

您可以依訊息來改變 QoS，使用 QoS 0 來遞送不重要的訊息，以及使用 QoS 2 來遞送重要訊息。如果 QoS 為 0，則訊息傳輸量大約是 QoS 2 的兩倍。

命名慣例

如果您要為許多用戶端設計應用程式，請實作有效的命名慣例。若要將每一個用戶端對映至正確的 `ClientIdentifier`，請使 `ClientIdentifier` 有意義。良好的命名慣例可讓管理者更容易找出哪些用戶端在執行中。命名慣例可協助管理者在「WebSphere MQ 探險家」中過濾長清單的用戶端，並協助判斷問題；請參閱 [用戶端 ID](#)。

傳輸量

主題名稱的長度會影響流經網路的位元組數。當發佈或訂閱時，訊息中的位元組數可能很重要。因此，請限制主題名稱中的字元數。當 MQTT 用戶端訂閱主題 WebSphere MQ 時，會提供下列格式的名稱：

```
ClientIdentifier:TopicName
```

若要檢視 MQTT 用戶端的所有訂閱，您可以使用 WebSphere MQ MQSC **DISPLAY** 指令：

```
DISPLAY SUB('ClientID1:*')
```

在 WebSphere MQ 中定義資源以供 MQTT 用戶端使用

MQTT 用戶端會連接至 IBM WebSphere MQ 遠端佇列管理程式。WebSphere MQ 應用程式有兩種基本方法可將訊息傳送至 MQTT 用戶端：將預設傳輸佇列設為 `SYSTEM.MQTT.TRANSMIT.QUEUE` 或使用佇列管理程式別名。如果有大量 MQTT 用戶端，請定義佇列管理程式的預設傳輸佇列。使用預設傳輸佇列設定可簡化管理工作；請參閱 [配置分散式佇列以將訊息傳送至 MQTT 用戶端](#)。

透過避免訂閱來改善可調整性。

當 MQTT V3 用戶端訂閱主題時，遙測 (MQXR) 服務會在 WebSphere MQ 中建立訂閱。訂閱會將用戶端的發佈遞送至 `SYSTEM.MQTT.TRANSMIT.QUEUE`。每一個發佈項目的傳輸標頭中的遠端佇列管理程式名稱會設為進行訂閱之 MQTT 用戶端的 `ClientIdentifier`。如果有許多用戶端，每一個都建立自己的訂閱，這會導致在整個 WebSphere MQ 發佈/訂閱叢集或階層中維護許多 Proxy 訂閱。如需不使用發佈/訂閱，而是改用點對點型解決方案的相關資訊，請參閱 [直接傳送訊息至用戶端](#)。

管理大量用戶端

若要支援許多同時連接的用戶端，請設定 JVM 參數 `-Xms` 及 `-Xmx`，以增加可用於遙測 (MQXR) 服務的記憶體。請遵循下列步驟：

1. 在遙測服務配置目錄中尋找 `java.properties` 檔案；請參閱 [Windows 上的遙測 \(MQXR\) 服務配置目錄](#) 或 [Linux 上的遙測服務配置目錄](#)。
2. 遵循檔案中的指示；1 GB 的資料堆足夠 50,000 個同時連接的用戶端使用。

```
# Heap sizing options - uncomment the following lines to set the heap to 1G
#-Xmx1024m
#-Xms1024m
```

3. 在 `java.properties` 檔案中新增其他指令行引數，以傳遞至執行遙測 (MQXR) 服務的 JVM；請參閱 [將 JVM 參數傳遞至遙測 \(MQXR\) 服務](#)。

若要增加 Linux 上開啟檔案描述子的數目，請將下列行新增至 `/etc/security/limits.conf/`，然後重新登入。

```
@mqm soft nofile 65000
@mqm hard nofile 65000
```

每一個 Socket 都需要一個檔案描述子。遙測服務需要一些額外的檔案描述子，因此此數目必須大於所需的開啟 Socket 數目。

佇列管理程式會對每一個不可延續訂閱使用物件控點。為了支援許多作用中的不可延續訂閱，會增加佇列管理程式中的作用中控點數目上限；例如：

```
echo ALTER QMGR MAXHANDS(999999999) | runmqsc qMgrName
```

圖 26: 變更 Windows 上的控點數上限

```
echo "ALTER QMGR MAXHANDS(999999999)" | runmqsc qMgrName
```

圖 27: 變更 Linux 上的控點數上限

其他考量

規劃系統需求時，請考量重新啟動系統所花費的時間長度。計劃的關閉時間可能會影響佇列中等待處理的訊息數目。請配置系統，以便在可接受的時間內順利處理訊息。檢閱磁碟儲存體、記憶體及處理能力。對於部分用戶端應用程式，可能可以在用戶端重新連接時捨棄訊息。如果要捨棄訊息，請在用戶端連線參數中設定 `CleanSession`；請參閱 [清除階段作業](#)。或者，在 MQTT 用戶端中使用最佳效能服務品質 0 來發佈及訂閱；請參閱 [服務品質](#)。從 WebSphere MQ 傳送訊息時，請使用 [非持續性](#) 訊息。當系統或連線重新啟動時，不會回復具有這些服務品質的訊息。

IBM WebSphere MQ Telemetry 支援的裝置

MQTT 用戶端可以在一系列裝置上執行，從感應器和掣動器，到手持裝置和車輛系統。

MQTT 用戶端小型，且在受記憶體少及處理能力低限制的裝置上執行。MQTT 通訊協定可靠且具有小型標頭，適用於受低頻寬、高成本及間歇性可用性限制的網路。

IBM WebSphere MQ Telemetry 提供三個用戶端，這些用戶端全都實作 MQTT v3 通訊協定：

- 可從最小的 CLDC (已連接的受限裝置配置) / MIDP (行動式資訊裝置設定檔) 到 CDC (已連接裝置配置) / Foundation、J2SE (Java Platform Standard Edition) 及 J2EE (Java Platform, Enterprise Edition)，在所有 Java 變異上執行的 Java 用戶端。也支援 IBM jclRM 自訂類別庫。
- C 參照實作，以及 Windows 和 Linux 系統的預先建置原生用戶端。C 參照實作可讓 MQTT 移轉至廣泛的裝置和平台。
- 進階用戶端 IBM WebSphere MQ Telemetry daemon for devices，以 C 撰寫，而且可以在任何適當的 Linux 或 Windows 平台上執行。

Intel 上的部分 Windows 系統 (包括 Windows XP、RedHat、Ubuntu)，以及 ARM 平台上的部分 Linux 系統 (例如執行 C 用戶端的 Eurotech Viper 實作版本 Linux)，但 IBM 不提供平台的服務支援。如果您想要聯絡 IBM 支援中心，則必須在受支援平台上向用戶端重新產生問題。

Java ME 平台通常用於小型裝置，例如掣動器、感應器、行動電話及其他內嵌裝置。Java SE 平台通常安裝在高階內嵌裝置上，例如桌上型電腦和伺服器。

註：[Eclipse paho](#) 專案和 [mqtt.org](#) 具有最新遙測用戶端的免費下載，以及一系列程式設計語言的範例。

相關概念

第 84 頁的『用於裝置的 Telemetry 常駐程式』

適用於裝置的 WebSphere MQ Telemetry 常駐程式是進階 MQTT V3 用戶端應用程式。可以利用它來儲存及轉遞來自其他 MQTT 用戶端的訊息。它可以如 MQTT 用戶端一樣連接至 WebSphere MQ，但是，您也可以將其他 MQTT 用戶端連接至它。您也可以將它連接至其他 Telemetry 常駐程式。

管理物件

佇列管理程式會定義 IBM WebSphere MQ 物件的內容或屬性。這些屬性的值會影響 WebSphere MQ 處理這些物件的方式。從應用程式中，您可以使用「訊息佇列介面 (MQI)」來控制物件。從程式定址時，由 MQ 物件描述子 (MQOD) 來識別物件。

當您使用 WebSphere MQ 指令來執行物件管理作業 (例如定義、修改或刪除物件) 時，佇列管理程式會檢查您是否具有執行作業所需的權限層次。同樣地，當應用程式使用 MQOPEN 呼叫來開啟物件時，佇列管理程式會先檢查應用程式是否具有必要的權限層次，然後才容許存取該物件。會對正在開啟的物件名稱進行檢查。

物件管理包括下列作業：

- 啟動和停止佇列管理程式。
- 建立應用程式的物件，特別是佇列。
- 使用通道來建立通往其他 (遠端) 系統上佇列管理程式的通訊路徑。
- 建立佇列管理程式的叢集，以簡化整體管理程序及平衡工作量。

如需如何建立及管理 WebSphere MQ 物件的方法概觀，請參閱第 107 頁的『管理物件』。

如需物件管理作業的進一步詳細資料，請參閱下列子主題：

物件

許多管理作業涉及操作各種類型的 IBM WebSphere MQ 物件。

如需命名 IBM WebSphere MQ 物件的相關資訊，請參閱第 103 頁的『命名 IBM WebSphere MQ 物件』。

如需在佇列管理程式上建立之預設物件的相關資訊，請參閱第 108 頁的『系統預設物件』。

如需不同類型 IBM WebSphere MQ 物件的相關資訊，請參閱下列子主題：

相關概念

第 37 頁的『[訊息佇列作業簡介](#)』

WebSphere MQ 產品可讓程式使用一致的應用程式設計介面，透過不同元件 (處理器、作業系統、子系統及通訊協定) 的網路彼此通訊。

第 107 頁的『[物件屬性](#)』

物件的內容由其屬性定義。有些您可以指定，有些您只能檢視。

相關參考

[MQSC 指令](#)

佇列

WebSphere MQ 佇列及佇列屬性簡介。

WebSphere MQ 佇列 是一個具名物件，應用程式可以從中放置訊息，以及從中取得訊息。

訊息儲存在佇列中，因此如果放置應用程式預期會回覆其訊息，則可以在等待該回覆時執行其他工作。應用程式會使用「[訊息佇列介面 \(MQI\)](#)」來存取佇列，如「[訊息佇列介面](#)」概觀中所述。

必須已建立佇列，才能將訊息放置在佇列上。佇列是由佇列管理程式所擁有，且該佇列管理程式可以擁有許多佇列。不過，每一個佇列的名稱在該佇列管理程式中必須是唯一的。

佇列是透過佇列管理程式來維護。在大部分情況下，每一個佇列都由其佇列管理程式實際管理，但應用程式並不清楚這一點。WebSphere MQ for z/OS 共用佇列可以由佇列共用群組中的任何佇列管理程式來管理。

若要建立佇列，您可以使用 WebSphere MQ 指令 (MQSC)、PCF 指令或平台專用介面，例如 WebSphere MQ for z/OS 作業及控制台。

您可以從應用程式動態建立暫時工作的本端佇列。例如，您可以建立 *reply-to* 佇列 (在應用程式結束之後不需要)。如需相關資訊，請參閱 [第 94 頁的『動態和模型佇列』](#)。

在使用佇列之前，您必須先開啟佇列，並指定您要如何處理它。例如，您可以開啟下列項目的佇列：

- 僅瀏覽訊息 (不擷取訊息)
- 擷取訊息 (並與其他程式共用存取權，或具有獨佔性存取權)
- 將訊息放置在佇列上
- 查詢佇列的屬性
- 設定佇列的屬性

如需開啟佇列時可以指定之選項的完整清單，請參閱 [MQOPEN-開啟物件](#)。

佇列的屬性

當定義佇列時，會指定佇列的部分屬性，且之後無法變更 (例如，佇列類型)。佇列的其他屬性可以分組成可變更的屬性：

- 由佇列管理程式在處理佇列期間 (例如，佇列的現行深度)
- 僅透過指令 (例如，佇列的文字說明)
- 由應用程式使用 MQSET 呼叫 (例如，是否容許在佇列上放置作業)

您可以使用 MQINQ 呼叫來尋找所有屬性的值。

多種佇列類型共用的屬性如下：

QName

佇列的名稱

QType

佇列的類型

QDesc

佇列的文字說明

InhibitGet

是否容許程式從佇列取得訊息 (雖然您永遠無法從遠端佇列取得訊息)

InhibitPut

是否容許程式將訊息放置在佇列上

DefPriority

放置在佇列上之訊息的預設優先順序

DefPersistence

放置在佇列上之訊息的預設持續性

Scope (在 z/OS 上不受支援)

控制此佇列的項目是否也存在於名稱服務中

如需這些屬性的完整說明，請參閱 [佇列的屬性](#)。

相關概念

[第 92 頁的『遠端佇列』](#)

對程式而言，如果佇列是由與程式所連接之佇列管理程式不同的佇列管理程式所擁有，則該佇列是遠端佇列。

[第 93 頁的『別名佇列』](#)

別名佇列是一個 WebSphere MQ 物件，可用來存取另一個佇列或主題。這表示多個程式可以使用相同的佇列，並使用不同的名稱來存取該佇列。

[第 96 頁的『定義佇列』](#)

您可以使用 MQSC 指令 DEFINE 或 PCF 建立佇列指令，將佇列定義至 IBM WebSphere MQ。

[第 96 頁的『IBM WebSphere MQ 使用的佇列』](#)

IBM WebSphere MQ 會將某些本端佇列用於與其作業相關的特定用途。

相關工作

[開發應用程式參照](#)

相關參考**MQSC 指令**

[第 91 頁的『本端佇列』](#)

傳輸、起始、無法傳送的郵件、指令、預設、通道及事件佇列都是本端佇列的類型。

[第 93 頁的『共用及叢集佇列』](#)

此資訊定義並說明術語共用佇列及叢集佇列，以及提供兩者之間的比較。

[第 94 頁的『動態和模型佇列』](#)

此資訊可讓您深入瞭解動態佇列、暫時及永久動態佇列的內容、動態佇列的使用、使用動態佇列時的一些考量，以及模型佇列。

本端佇列

傳輸、起始、無法傳送的郵件、指令、預設、通道及事件佇列都是本端佇列的類型。

如果佇列是由程式連接到的佇列管理程式所擁有，則該佇列對程式而言是本端佇列。您可以從本端佇列中取得訊息，以及將訊息放置在本端佇列上。

佇列定義物件會保留佇列的定義資訊，以及放置在該佇列上的實體訊息。

每一個佇列管理程式都可以有一些本端佇列，用於特殊用途：

傳輸佇列

當應用程式將訊息傳送至遠端佇列時，本端佇列管理程式會將訊息儲存在特殊本端佇列中，稱為傳輸佇列。

訊息通道代理程式是與傳輸佇列相關聯的通道程式，它會將訊息遞送至下一個目的地。下一個目的地是訊息通道所連接的佇列管理程式。它不一定是與訊息最終目的地相同的佇列管理程式。當訊息遞送至其下一個目的地時，會從傳輸佇列中刪除它。訊息可能必須在其前往其最終目的地的旅程中通過許多佇列管理程式。您必須在路徑上的每一個佇列管理程式中定義傳輸佇列，每一個佇列管理程式都會保留等待傳輸至下一個目的地的訊息。雖然訊息可能有不同的最終目的地，但一般傳輸佇列會保留下一個目的地的訊息。叢集傳輸佇列會保留多個目的地的訊息。每則訊息的 `correlID` 會識別放置訊息的通道，以將它傳送至其下一個目的地。

您可以在佇列管理程式中定義數個傳輸佇列。您可以為相同的目的地定義數個傳輸佇列，每個傳輸佇列都用於不同的服務類別。例如，您可能想要為小訊息及進入相同目的地的大訊息建立不同的傳輸佇列。然後，您可以使用不同的訊息通道來傳送訊息，以便大型訊息不會保留較小的訊息。在 z/OS 以外的平台上，依預設訊息會進入單一叢集傳輸佇列 `SYSTEM.CLUSTER.TRANSMIT.QUEUE`。作為其他平台上的選項，您可以變更預設值，並將進入不同叢集佇列管理程式的訊息資料流量分隔到不同的叢集傳輸佇列。如果您將佇列管理程式屬性 `DEFCLXQ` 設為 `CHANNEL`，則每一個叢集傳送端通道會建立個別叢集傳輸佇列。其他平台上的另一個選項是手動定義叢集傳送端通道要使用的叢集傳輸佇列。

傳輸佇列可以觸發訊息通道代理程式向前傳送訊息；請參閱 [使用觸發程式來啟動 WebSphere MQ 應用程式](#)。

起始佇列

起始佇列 是一種本端佇列，當應用程式佇列上發生觸發事件時，佇列管理程式會在該佇列上放置觸發訊息。

觸發事件是要讓程式開始處理佇列的事件。例如，事件可能超過 10 則訊息送達。如需觸發如何運作的相關資訊，請參閱 [使用觸發程式來啟動 WebSphere MQ 應用程式](#)。

無法傳送郵件 (未遞送訊息) 佇列

無法傳送的郵件 (未遞送的訊息) 佇列 是佇列管理程式放置無法遞送之訊息的本端佇列。

當佇列管理程式將訊息放入無法傳送郵件的佇列時，它會將標頭新增至訊息。標頭資訊包括佇列管理程式將訊息放置在無法傳送郵件的佇列上的原因。它也包含原始訊息的目的地、日期，以及佇列管理程式將訊息放入無法傳送郵件的佇列的時間。

應用程式也可以使用無法遞送之訊息的佇列。如需相關資訊，請參閱 [使用無法傳送的郵件 \(未遞送訊息\) 佇列](#)。

系統指令佇列

系統指令佇列 是適當授權的應用程式可以將 WebSphere MQ 指令傳送至其中的佇列。這些佇列會接收平台上支援的 PCF、MQSC 及 CL 指令，以準備好讓佇列管理程式對它們執行動作。

系統預設佇列

系統預設佇列 包含系統的佇列起始定義。當您建立佇列定義時，佇列管理程式會從適當的系統預設佇列複製定義。建立佇列定義不同於建立動態佇列。動態佇列的定義基於您選擇作為動態佇列範本的模式佇列。

事件佇列

事件佇列 會保留事件訊息。這些訊息由佇列管理程式或通道報告。

遠端佇列

對程式而言，如果佇列是由與程式所連接之佇列管理程式不同的佇列管理程式所擁有，則該佇列是遠端佇列。

在已建立通訊鏈結的情況下，程式可以將訊息傳送至遠端佇列。程式永遠無法從遠端佇列取得訊息。

當您定義遠端佇列時所建立的佇列定義物件只會保留本端佇列管理程式所需的資訊，以尋找您要訊息前往的佇列。此物件稱為遠端佇列的本端定義。遠端佇列的所有屬性都由擁有它的佇列管理程式保留，因為它是該佇列管理程式的本端佇列。

開啟遠端佇列時，若要識別佇列，您必須指定下列任一項：

- 定義遠端佇列的本端定義名稱。

若要建立遠端佇列的本端定義，請使用 `DEFINE QREMOTE` 指令；在 WebSphere MQ for IBM i 上，使用 `CRTMQM` 指令。

從應用程式的觀點來看，這與開啟本端佇列相同。應用程式不需要知道佇列是本端或遠端。

- 遠端佇列管理程式的名稱，以及該遠端佇列管理程式已知的佇列名稱。

除了第 90 頁的『佇列的屬性』中說明的共同屬性之外，遠端佇列的本端定義還具有三個屬性。這些是 *RemoteQName* (佇列擁有端佇列管理程式的識別名稱)、*RemoteQMgrName* (擁有端佇列管理程式的名稱) 及 *XmitQName* (將訊息轉遞至其他佇列管理程式時所使用的本端傳輸佇列名稱)。如需這些屬性的完整說明，請參閱 佇列的屬性。

如果您針對遠端佇列的本端定義使用 MQINQ 呼叫，則佇列管理程式只會傳回本端定義的屬性，即遠端佇列名稱、遠端佇列管理程式名稱及傳輸佇列名稱，而不是遠端系統中相符本端佇列的屬性。

另請參閱 [傳輸佇列](#)。

別名佇列

別名佇列 是一個 WebSphere MQ 物件，可用來存取另一個佇列或主題。這表示多個程式可以使用相同的佇列，並使用不同的名稱來存取該佇列。

解析別名 (稱為基本佇列) 所產生的佇列可以是本端佇列、遠端佇列的本端定義或共用佇列 (一種本端佇列類型，只適用於 WebSphere MQ for z/OS)。它也可以是預先定義的佇列或動態佇列，如平台所支援。

別名也可以解析成主題。如果應用程式目前將訊息放入佇列，則可以讓佇列名稱成為主題的別名，以發佈至主題。不需要變更應用程式碼。

註: 別名無法解析為另一個本端定義的別名佇列。

別名佇列的使用範例是讓系統管理者對基本佇列名稱 (亦即，別名所解析的佇列) 及別名佇列名稱提供不同的存取權。這表示程式或使用者可以獲授權使用別名佇列，但不能使用基本佇列。

或者，授權可以設定為禁止對別名執行 put 作業，但允許對基本佇列執行這些作業。

在某些應用程式中，使用別名佇列表示系統管理者可以輕鬆變更別名佇列物件的定義，而不需要變更應用程式。

當程式嘗試使用別名時，WebSphere MQ 會根據別名進行授權檢查。它不會檢查程式是否有權存取別名所解析的名稱。因此，程式可以獲授權存取別名佇列名稱，但不能存取已解析的佇列名稱。

除了第 90 頁的『佇列』中說明的一般佇列屬性之外，別名佇列還具有 *BaseQName* 屬性。這是別名所解析成的基本佇列名稱。如需此屬性的完整說明，請參閱 [BaseQName \(MQCHAR48\)](#)。

別名佇列的 *InhibitGet* 及 *InhibitPut* 屬性 (請參閱第 90 頁的『佇列』) 屬於別名。例如，如果別名佇列名稱 ALIAS1 解析為基本佇列名稱 BASE，則 ALIAS1 只會影響 ALIAS1，且不禁止 BASE。不過，BASE 上的禁止也會影響 ALIAS1。

DefPriority 和 *DefPersistence* 屬性也屬於別名。因此，例如，您可以將不同的預設優先順序指派給相同基本佇列的不同別名。此外，您也可以變更這些優先順序，而不需要變更使用別名的應用程式。

共用及叢集佇列

此資訊定義並說明術語共用佇列及叢集佇列，以及提供兩者之間的比較。

共用佇列

共用佇列 是具有訊息的本端佇列類型，可由佇列共用群組中的一或多個佇列管理程式存取。**共用佇列僅適用於 WebSphere MQ for z/OS。** (這與由多個應用程式使用相同佇列管理程式共用的佇列不同。) 共用佇列由連結機能 (CF) 保留，且可由佇列共用群組中的任何佇列管理程式存取。佇列共用群組中每一個共用佇列的名稱在該群組內必須是唯一的。

叢集佇列數

叢集佇列是由叢集佇列管理程式所管理的佇列，並可提供給叢集的其他佇列管理程式使用。

叢集佇列管理程式會建立佇列的本端佇列定義，並指定要在其中提供佇列的叢集名稱。此定義會將佇列通告至叢集中的其他佇列管理程式。叢集中的其他佇列管理程式不需要相對應的遠端佇列定義，就可以將訊息放置在叢集佇列中。叢集佇列可以在多個叢集中通告。如需進一步資訊，請參閱 [叢集](#) 及 [配置佇列管理程式叢集](#)。

共用佇列與叢集佇列之間的比較

此資訊旨在協助您比較共用佇列和叢集佇列，並決定哪些可能更適合您的系統。

偏差成本

在叢集佇列中，訊息由移轉裝置傳送，因此除了應用程式成本之外，還容許移轉裝置成本。網路中有一些成本，因為通道會取得及放置訊息。共用佇列不存在這些成本，因此在「佇列共用群組」中的佇列管理程式之間移動訊息時，其處理能力會低於叢集佇列。

訊息可用性

當放入佇列時，叢集佇列會將訊息傳送至其中一個佇列管理程式，且作用中通道會連接至您的佇列管理程式。在遠端佇列管理程式上，如果用來處理訊息的應用程式無法運作，則不會處理訊息，並等待應用程式啟動。同樣地，如果佇列管理程式已關閉，則在佇列管理程式重新啟動之前，佇列管理程式上的任何訊息都無法使用。這些實例顯示的訊息可用性低於使用共用佇列時。

使用共用佇列時，佇列共用群組中的任何應用程式都可以取得傳送的訊息。如果您關閉佇列共用群組中的一個佇列管理程式，則訊息可供其他佇列管理程式使用，提供比使用叢集佇列更高的訊息可用性。

容量

連結機能比磁碟更貴；因此，將 1,000,000 則訊息儲存在本端佇列的成本低於具有足夠容量來儲存相同訊息數的連結機能。

傳送至其他佇列管理程式

共用佇列訊息只能在佇列共用群組內使用。如果您要使用佇列共用群組之外的佇列管理程式，則必須使用移轉裝置。您可以使用叢集作業來平衡多個遠端分散式佇列管理程式之間的工作量。

工作量平衡

您可以使用叢集作業來提供加權，讓哪些通道及佇列管理程式取得所傳送訊息的比例。例如，您可以將 60% 的訊息傳送至一個佇列管理程式，並將 40% 的訊息傳送至另一個佇列管理程式。此實例不取決於遠端佇列管理程式處理工作的能力。具有第一個佇列管理程式的系統可能超載，且具有第二個佇列管理程式的系統可能閒置，但大部分訊息仍會移至第一個佇列管理程式。

使用共用佇列，兩個 CICS 系統可以取得訊息。如果其中一個系統超載，則另一個系統會接管大部分工作量。

動態和模型佇列

此資訊可讓您深入瞭解動態佇列、暫時及永久動態佇列的內容、動態佇列的使用、使用動態佇列時的一些考量，以及模型佇列。

當應用程式發出 MQOPEN 呼叫來開啟模型佇列時，佇列管理程式會動態建立與模型佇列具有相同屬性的本端佇列實例。視模型佇列的 *DefinitionType* 欄位值而定，佇列管理程式會建立暫時或永久動態佇列 (請參閱 [建立動態佇列](#))。

暫時動態佇列的內容

暫時動態佇列 具有下列內容：

- 它們不能是共用佇列，可從佇列共用群組中的佇列管理程式存取。
請注意，佇列共用群組只能在 WebSphere MQ for z/OS 上使用。
- 它們只會保留非持續訊息。
- 它們是無法復原的。
- 當佇列管理程式啟動時，會刪除它們。
- 當發出 MQOPEN 呼叫 (建立佇列) 的應用程式關閉或終止佇列時，會刪除它們。
 - 如果佇列上有任何已確定訊息，則會刪除這些訊息。

- 如果此時有任何未確定的 MQGET、MQPUT 或 MQPUT1 呼叫未針對佇列執行，則該佇列會標示為邏輯刪除，且在關閉處理期間或應用程式終止時，只會實際刪除 (在確定這些呼叫之後)。
- 如果目前正在使用佇列 (由建立中或另一個應用程式使用)，則該佇列會標示為邏輯刪除中，且只有在最後一個使用該佇列的應用程式關閉時才會實際刪除。
- 嘗試存取邏輯刪除的佇列 (除了關閉它之外) 失敗，原因碼為 MQRC_Q_DELETED。
- 當 MQCLOSE 呼叫針對建立佇列的對應 MQOPEN 呼叫指定時，MQCO_NONE、MQCO_DELETE 及 MQCO_DELETE_PURGE 都被視為 MQCO_NONE。

永久動態佇列的內容

永久動態佇列 具有下列內容：

- 它們會保留持續或非持續訊息。
- 它們可以在系統失效時回復。
- 當應用程式 (不一定是發出建立佇列的 MQOPEN 呼叫的應用程式) 使用 MQCO_DELETE 或 MQCO_DELETE_PURGE 選項順利關閉佇列時，會刪除它們。
 - 如果佇列上仍有任何訊息 (已確定或未確定)，則具有 MQCO_DELETE 選項的關閉要求會失敗。具有 MQCO_DELETE_PURGE 選項的關閉要求會成功，即使佇列上有已確定訊息 (作為關閉的一部分而被刪除的訊息)，但如果佇列有任何未確定的 MQGET、MQPUT 或 MQPUT1 呼叫未完成，則會失敗。
 - 如果刪除要求成功，但佇列剛好在使用中 (由建立中或另一個應用程式)，則佇列會標示為邏輯刪除中，且只有在由使用該佇列的最後一個應用程式關閉時才會實際刪除。
- 如果由未獲授權刪除佇列的應用程式關閉，則不會刪除它們，除非關閉的應用程式發出建立佇列的 MQOPEN 呼叫。會針對用來驗證對應 MQOPEN 呼叫的使用者 ID (或替代使用者 ID，如果已指定 MQOO_ALTERNATE_USER_AUTHORITY) 執行授權檢查。
- 可以使用與一般佇列相同的方式來刪除它們。

使用動態佇列

您可以將動態佇列用於：

- 在應用程式終止之後，不需要保留佇列的應用程式。
- 需要另一個應用程式處理訊息之回覆的應用程式。此類應用程式可以透過開啟模型佇列來動態建立回覆目的地佇列。例如，用戶端應用程式可以：
 1. 建立動態佇列。
 2. 請在要求訊息之訊息描述子結構的 *ReplyToQ* 欄位中提供其名稱。
 3. 將要求放在伺服器正在處理的佇列上。

然後伺服器可以將回覆訊息放置在回覆目的地佇列上。最後，用戶端可以處理回覆，並使用刪除選項來關閉回覆目的地佇列。

使用動態佇列時的考量

使用動態佇列時，請考量下列要點：

- 在主從架構模型中，每一個用戶端必須建立並使用自己的動態回覆目的地佇列。如果在多個用戶端之間共用動態回覆目的地佇列，則刪除回覆目的地佇列可能會延遲，因為沒有針對佇列的未確定活動，或因為另一個用戶端正在使用該佇列。此外，佇列可能標示為邏輯上已刪除，且後續 API 要求 (非 MQCLOSE) 無法存取。
- 如果您的應用程式環境要求必須在應用程式之間共用動態佇列，請確保只有在已確定佇列的所有活動時，才會關閉佇列 (使用刪除選項)。這應該由最後一位使用者來執行。這可確保佇列的刪除不會延遲，並將佇列因已標示為邏輯刪除而無法存取的期間縮至最小。

模型佇列

模型佇列 是您在建立動態佇列時使用的佇列定義範本。

您可以從 WebSphere MQ 程式動態建立本端佇列，並命名您要用作佇列屬性範本的模型佇列。此時，您可以變更新佇列的部分屬性。不過，您無法變更 *DefinitionType*。例如，如果您需要永久佇列，請選取定義類

型設為永久的模型佇列。部分交談式應用程式可以使用動態佇列來保留其查詢的回覆，因為它們在處理回覆之後可能不需要維護這些佇列。

您可以在 MQOPEN 呼叫的物件描述子 (MQOD) 中指定模型佇列的名稱。使用模型佇列的屬性，佇列管理程式會為您動態建立本端佇列。

您可以指定動態佇列的名稱 (完整) 或名稱的詞幹 (例如 ABC)，並讓佇列管理程式將唯一組件新增至此，也可以讓佇列管理程式為您指派完整的唯一名稱。如果佇列管理程式指派名稱，則會將它放入 MQOD 結構中。

您無法直接對模型佇列發出 MQPUT1 呼叫，但可以對透過開啟模型佇列建立的動態佇列發出 MQPUT1。

無法針對模型佇列發出 MQSET 和 MQINQ。使用 MQOO_INQUIRE 或 MQOO_SET 開啟模型佇列會導致對動態建立的佇列進行後續 MQINQ 及 MQSET 呼叫。

模型佇列的屬性是本端佇列的子集。如需更完整的說明，請參閱 [佇列的屬性](#)。

定義佇列

您可以使用 MQSC 指令 DEFINE 或 PCF 建立佇列指令，將佇列定義至 IBM WebSphere MQ。

指令指定佇列類型及其屬性。例如，本端佇列物件具有屬性可指定當應用程式在 MQI 呼叫中參照該佇列時所發生的情況。屬性範例如下：

- 應用程式是否可以從佇列擷取訊息 (啟用 GET)
- 應用程式是否可以將訊息放置在佇列上 (啟用 PUT)
- 佇列的存取權是一個應用程式專用，還是在應用程式之間共用
- 可同時儲存在佇列上的訊息數上限 (佇列深度上限)
- 可以放置在佇列上的訊息長度上限

如需定義佇列物件的進一步詳細資料，請參閱 [Script \(MQSC\) 指令](#)。

IBM WebSphere MQ 使用的佇列

IBM WebSphere MQ 會將某些本端佇列用於與其作業相關的特定用途。

您必須先定義這些佇列，然後 IBM WebSphere MQ 才能使用它們。

起始佇列

起始佇列是在觸發時使用的佇列。發生觸發事件時，佇列管理程式會將觸發訊息放置在起始佇列上。觸發事件是佇列管理程式偵測到的條件邏輯組合。例如，當佇列上的訊息數達到預先定義的深度時，可能會產生觸發事件。此事件會導致佇列管理程式將觸發訊息放置在指定的起始佇列上。此觸發訊息由觸發監視器擷取，觸發監視器是監視起始佇列的特殊應用程式。然後，觸發監視器會啟動觸發訊息中指定的應用程式。

如果佇列管理程式要使用觸發，則必須為該佇列管理程式至少定義一個起始佇列。請參閱 [管理物件以使用觸發程式來觸發](#)、[runmqtrm](#) 及 [啟動 WebSphere MQ 應用程式](#)

傳輸佇列

傳輸佇列是暫時儲存以遠端佇列管理程式為目的地之訊息的佇列。您必須為本端佇列管理程式要直接傳送訊息的每一個遠端佇列管理程式至少定義一個傳輸佇列。這些佇列也用於遠端管理；請參閱 [從本端佇列管理程式進行遠端管理](#)。如需在分散式佇列中使用傳輸佇列的相關資訊，請參閱 [WebSphere MQ 分散式傳訊技術](#)。

每一個佇列管理程式都可以有預設傳輸佇列。如果不屬於叢集的佇列管理程式將訊息放入遠端佇列，則預設動作是使用預設傳輸佇列。如果有傳輸佇列與目的地佇列管理程式同名，則會將訊息放置在該傳輸佇列上。如果有佇列管理程式別名定義，其中 **RQMNAME** 參數符合目的地佇列管理程式，且指定 **XMITQ** 參數，則訊息會放置在 **XMITQ** 所命名的傳輸佇列上。如果沒有 **XMITQ** 參數，則訊息會放在訊息中指名的本端佇列上。

叢集傳輸佇列

叢集內的每一個佇列管理程式都有一個稱為 **SYSTEM.CLUSTER.TRANSMIT.QUEUE** 的叢集傳輸佇列，以及一個模型叢集傳輸佇列 **SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE**。依預設，當您定義佇列管理程式時，會建立這些佇列的定義。如果佇列管理程式屬性 **DEFCLXQ** 設為 **CHANNEL**，則會自動為所建立的每一個叢集傳送端通道建立永久動態叢集傳輸佇列。佇列稱為 **SYSTEM.CLUSTER.TRANSMIT.ChannelName**。您也可以手動定義叢集傳輸佇列。

屬於叢集的佇列管理程式會將其中一個佇列上的訊息傳送至相同叢集中的其他佇列管理程式。

在名稱解析期間，叢集傳輸佇列優先於預設傳輸佇列，而特定叢集傳輸佇列優先於 `SYSTEM.CLUSTER.TRANSMIT.QUEUE`。

無法傳送郵件的佇列

無法傳送郵件的 (無法遞送的訊息) 佇列是儲存無法遞送至其正確目的地之訊息的佇列。例如，當目的地佇列已滿時，無法遞送訊息。提供的無法傳送郵件的佇列稱為 `SYSTEM.DEAD.LETTER.QUEUE`。

對於分散式佇列，請在涉及的每一個佇列管理程式上定義無法傳送郵件的佇列。

指令佇列

指令佇列 `SYSTEM.ADMIN.COMMAND.QUEUE` 是本端佇列，適當授權的應用程式可以將 MQSC 指令傳送至其中進行處理。然後，稱為指令伺服器的 IBM WebSphere MQ 元件會擷取這些指令。指令伺服器會驗證指令，傳遞有效的指令以供佇列管理程式處理，並將任何回應傳回適當的回覆目的地佇列。

建立每一個佇列管理程式時，會自動建立該佇列管理程式的指令佇列。

回覆目的地佇列

當應用程式傳送要求訊息時，接收訊息的應用程式可以傳回回覆訊息給傳送應用程式。此訊息放置在佇列上，稱為回覆目的地佇列，通常是傳送端應用程式的本端佇列。回覆目的地佇列的名稱由傳送端應用程式指定為訊息描述子的一部分。

事件佇列

檢測事件可用來獨立監視佇列管理程式，與 MQI 應用程式無關。

發生檢測事件時，佇列管理程式會將事件訊息放置在事件佇列上。然後，監視應用程式可以讀取此訊息，如果事件指出問題，則可能會通知管理者或起始一些補救動作。

註：觸發程式事件不同於檢測事件。觸發事件不是由相同的條件所造成，且不會產生事件訊息。

如需設備測試事件的相關資訊，請參閱 [設備測試事件](#)。

IBM WebSphere MQ 佇列管理程式

佇列管理程式及其提供給應用程式之佇列服務的簡介。

程式必須具有與佇列管理程式的連線，才能使用該佇列管理程式的服務。程式可以明確地建立此連線 (使用 `MQCONN` 或 `MQCONNX` 呼叫)，也可以隱含地建立連線 (這取決於執行程式的平台及環境)。

佇列管理程式會提供佇列作業服務給應用程式，並管理屬於它們的佇列。佇列管理程式可確保下列動作：

- 根據接收的指令變更物件屬性。
- 當符合適當的條件時，會產生特殊事件 (例如觸發事件或設備測試事件)。
- 訊息會依照發出 `MQPUT` 呼叫的應用程式所要求，放置在正確的佇列中。如果無法執行此動作，則會通知應用程式，並提供適當的原因碼。

每一個佇列都屬於單一佇列管理程式，且被認為是該佇列管理程式的本端佇列。應用程式所連接的佇列管理程式被認為是該應用程式的本端佇列管理程式。對於應用程式，屬於其本端佇列管理程式的佇列是本端佇列。

遠端佇列是屬於另一個佇列管理程式的佇列。遠端佇列管理程式是本端佇列管理程式以外的任何佇列管理程式。遠端佇列管理程式可以存在於網路上的遠端機器上，也可以存在於與本端佇列管理程式相同的機器上。WebSphere MQ 支援相同機器上的多個佇列管理程式。

佇列管理程式物件可以在部分 MQI 呼叫中使用。例如，您可以使用 MQI 呼叫 `MQINQ` 來查詢佇列管理程式物件的屬性。

佇列管理程式的屬性

與每一個佇列管理程式相關聯的是一組定義其性質的屬性 (或內容)。佇列管理程式的部分屬性在建立時是固定的; 您可以使用 WebSphere MQ 指令來變更其他屬性。您可以使用 `MQINQ` 呼叫來查詢所有屬性的值，但用於 Secure Sockets Layer (SSL) 加密的屬性除外。

固定 屬性包括：

- 佇列管理程式的名稱

- 執行佇列管理程式的平台 (例如, Windows)
- 佇列管理程式支援的系統控制指令層次
- 您可以指派給佇列管理程式所處理訊息的優先順序上限
- 程式可以將 WebSphere MQ 指令傳送至其中的佇列名稱
- 佇列管理程式可以處理的訊息長度上限
- 當程式放置及取得訊息時, 佇列管理程式是否支援同步點

可變更 屬性包括:

- 佇列管理程式的文字說明
- 佇列管理程式在處理 MQI 呼叫時用於字串的字集 ID
- 佇列管理程式用來限制觸發訊息數目的時間間隔
- 佇列管理程式的無法傳送郵件 (未遞送訊息) 佇列名稱
- 佇列管理程式的預設傳輸佇列名稱
- 任何一個連線的開啟控點數目上限
- 啟用及停用事件報告的各種種類
- 工作單元內未確定的訊息數上限

佇列管理程式及工作量管理

您可以設定佇列管理程式的叢集, 該佇列管理程式具有相同佇列的多個定義 (例如, 叢集中的佇列管理程式可以是彼此的複本)。管理佇列實例的任何佇列管理程式都可以處理特定佇列的訊息。工作量管理演算法會決定哪個佇列管理程式處理訊息, 因此會在佇列管理程式之間分散工作量; 如需進一步資訊, 請參閱 [叢集工作量管理演算法](#)。

程序定義

程序定義物件可讓應用程式啟動, 而不需要操作員介入, 方法是定義應用程式的屬性以供佇列管理程式使用。

程序定義物件定義應用程式, 該應用程式啟動以回應 IBM WebSphere MQ 佇列管理程式上的觸發事件。程序定義屬性包括應用程式 ID、應用程式類型及應用程式特有的資料。如需相關資訊, 請參閱 [第 96 頁的『IBM WebSphere MQ 使用的佇列』](#) 下的「起始佇列」項目。

若要容許在不需要操作員介入的情況下啟動應用程式 (如 [使用觸發程式啟動 WebSphere MQ 應用程式](#) 中所述), 佇列管理程式必須知道應用程式的屬性。這些屬性定義在 [程序定義物件](#) 中。

建立物件時, *ProcessName* 屬性是固定的; 您可以使用 IBM WebSphere MQ 指令來變更其他屬性。

您可以使用 [MQINQ-查詢物件屬性](#) 來查詢 所有 屬性的值。

如需程序定義屬性的完整說明, 請參閱 [程序定義的屬性](#)。

名單

名單是 WebSphere MQ 物件, 其中包含叢集名稱、佇列名稱或鑑別資訊物件名稱的清單。在叢集中, 它可以用來識別佇列管理程式為其保留儲存庫的叢集清單。

名稱清單是 WebSphere MQ 物件, 其中包含其他 WebSphere MQ 物件的清單。通常名稱清單是提供給諸如觸發監視器之類的應用程式, 以用來識別佇列的群組。使用名單的優點是獨立於應用程式進行維護; 它可以在不停止任何使用它的應用程式的情況下進行更新。此外, 如果一個應用程式失敗, 則名單不受影響, 其他應用程式可以繼續使用它。

名稱清單也會與佇列管理程式叢集搭配使用, 以維護多個 WebSphere MQ 物件所參照的叢集清單。

您可以使用 MQSC 指令的作業及控制台, 來定義及修改名稱清單。

程式可以使用 MQI 來找出這些名稱清單中包含的佇列。名稱清單的組織是應用程式設計者和系統管理者的責任。

如需名稱清單屬性的完整說明, 請參閱 [名稱清單的屬性](#)。

鑑別資訊物件

佇列管理程式鑑別資訊物件的簡介，以及進一步資訊的鏈結。

佇列管理程式鑑別資訊物件是 WebSphere MQ 支援 Secure Sockets Layer (SSL) 和傳輸層安全 (TLS) 的一部分。它提供檢查已撤銷憑證所需的定義。憑證管理中心會撤銷無法再信任的憑證。

本節說明搭配使用 `setmqaut`、`dspmqaut`、`dmpmqaut`、`rcrmqobj`、`rcdmqing` 及 `dspmqfls` 指令與鑑別資訊物件。如需 SSL 和 TLS 以及使用鑑別資訊物件的概觀，請參閱 [WebSphere MQ SSL 和 TLS 支援](#)。

如需 SSL 和 TLS 的相關資訊，請參閱 [Secure Sockets Layer \(SSL\) 和傳輸層安全 \(TLS\) 概念](#)。

鑑別資訊物件提供執行憑證撤銷檢查所需的定義。

如需鑑別資訊物件屬性的完整說明，請參閱 [鑑別資訊物件](#)。

通訊資訊物件

IBM WebSphere MQ 多重播送提供低延遲、高扇出、可靠的多重播送傳訊功能。需要通訊資訊 (COMMINFO) 物件，才能使用「多重播送」傳輸。

COMMINFO 物件是包含與多重播送傳輸相關聯之屬性的 IBM WebSphere MQ 物件。如需這些屬性的相關資訊，請參閱 [DEFINE COMMINFO](#)。如需建立 COMMINFO 物件的相關資訊，請參閱 [開始使用多重播送](#)。

相關概念

[第 108 頁的『IBM WebSphere MQ 多重播送』](#)

IBM WebSphere MQ 多重播送提供低延遲、高扇出、可靠的多重播送傳訊功能。

通道

通道是分散式佇列管理程式所使用的通訊鏈結。

通道是提供從一個佇列管理程式到另一個佇列管理程式之通訊路徑的物件。在分散式佇列中使用通道，以將訊息從一個佇列管理程式移至另一個佇列管理程式，而且它們會保護應用程式免受基礎通訊協定的影響。佇列管理程式可能存在於相同或不同的平台上。

若要讓佇列管理程式彼此通訊，您必須在佇列管理程式中定義一個通道物件來傳送訊息，並在佇列管理程式中定義另一個互補的通道物件來接收訊息。

在 WebSphere MQ 中，通道有兩種種類：

- 訊息通道是單向的，並將訊息從一個佇列管理程式傳送至另一個佇列管理程式；如需相關資訊，請參閱 [通道結束程式呼叫及資料結構](#)。
- MQI 通道可雙向，並將 MQI 呼叫從 WebSphere MQ MQI 用戶端傳送至佇列管理程式，以及從佇列管理程式回應至 WebSphere MQ 用戶端；如需相關資訊，請參閱 [第 99 頁的『何謂通道？』](#)。

相關概念

[第 44 頁的『交互通訊概念』](#)

在 WebSphere MQ 中，交互通訊表示將訊息從一個佇列管理程式傳送至另一個佇列管理程式。接收端佇列管理程式可以位於相同或另一部機器上；附近或世界另一端。它可以在與本端佇列管理程式相同的平台上執行，也可以在 WebSphere MQ 支援的任何平台上執行。這稱為分散式環境。WebSphere MQ 會使用「分散式佇列管理 (DQM)」來處理分散式環境中的通訊。

相關工作

[管理遠端 WebSphere MQ 物件](#)

相關參考

[通道結束程式呼叫和資料結構](#)

[第 101 頁的『通訊』](#)

WebSphere MQ MQI 用戶端使用 MQI 通道來與伺服器進行通訊。

何謂通道？

通道是 WebSphere MQ MQI 用戶端與 WebSphere MQ 伺服器之間的邏輯通訊鏈結，或兩個 WebSphere MQ 伺服器之間的邏輯通訊鏈結。

通道有兩個定義：連線兩端各一個。必須在連線的每一端使用相同的通道名稱，且使用的通道類型必須相容。

WebSphere MQ 中有兩種通道種類，在這些種類內具有不同的通道類型：

相關概念

第 100 頁的『[訊息通道](#)』

訊息通道是單向鏈結。它會使用 訊息通道代理程式 (MCA) 來連接兩個佇列管理程式。

第 100 頁的『[MQI 通道](#)』

MQI 通道會將 WebSphere MQ MQI 用戶端連接至伺服器機器上的佇列管理程式，並在您從 WebSphere MQ MQI 用戶端應用程式發出 MQCONN 或 MQCONNX 呼叫時建立。

第 101 頁的『[停止通道](#)』

在 WebSphere MQ 中，當您對伺服器連線通道發出 STOP CHANNEL 指令時，您可以選擇要使用何種方法來停止用戶端連線通道。

訊息通道

訊息通道是單向鏈結。它會使用 訊息通道代理程式 (MCA) 來連接兩個佇列管理程式。

訊息通道的目的是將訊息從一個佇列管理程式傳送至另一個佇列管理程式。用戶端伺服器環境不需要訊息通道。

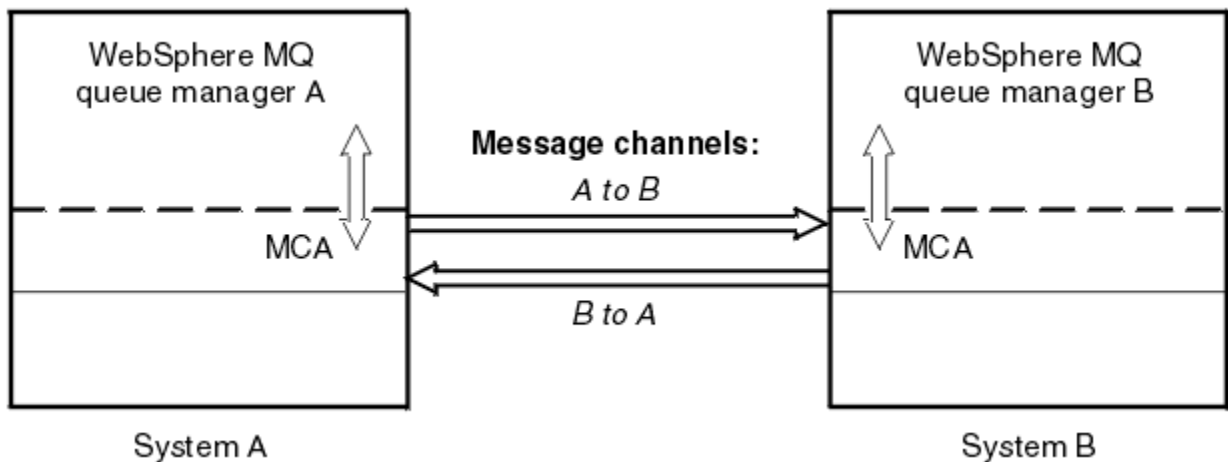


圖 28: 兩個佇列管理程式之間的訊息通道

MQI 通道

MQI 通道會將 WebSphere MQ MQI 用戶端連接至伺服器機器上的佇列管理程式，並在您從 WebSphere MQ MQI 用戶端應用程式發出 MQCONN 或 MQCONNX 呼叫時建立。

它是雙向鏈結，僅用於傳送 MQI 呼叫及回應，包括包含訊息資料的 MQPUT 呼叫，以及導致傳回訊息資料的 MQGET 呼叫。您可以使用不同的方式來建立及使用通道定義 (請參閱 [定義 MQI 通道](#))。

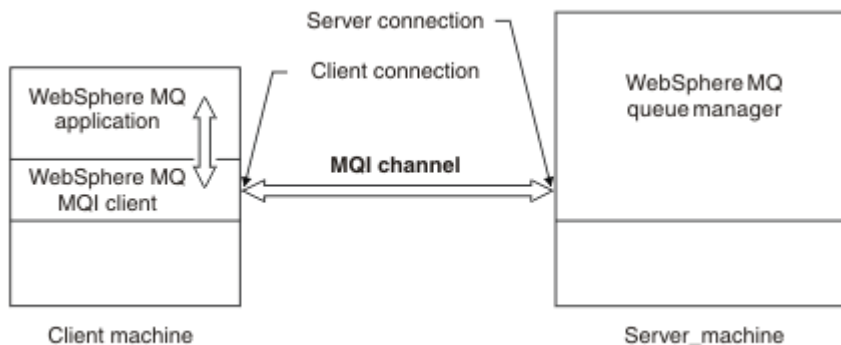


圖 29: MQI 通道上的用戶端連線及伺服器連線

MQI 通道可用來將用戶端連接至單一佇列管理程式，或連接至屬於佇列共用群組的佇列管理程式 (請參閱 [將用戶端連接至佇列共用群組](#))。

MQI 通道定義有兩種通道類型。它們定義雙向 MQI 通道。

用戶端連線通道

此類型適用於 WebSphere MQ MQI 用戶端。

伺服器連線通道

此類型適用於執行佇列管理程式的伺服器，在 WebSphere MQ MQI 用戶端環境中執行的 WebSphere MQ 應用程式會與之進行通訊。

停止通道

在 WebSphere MQ 中，當您對伺服器連線通道發出 STOP CHANNEL 指令時，您可以選擇要使用何種方法來停止用戶端連線通道。

這表示可以控制發出 MQGET 等待呼叫的用戶端通道，而且您可以決定停止通道的方式及時間。

可以用三種模式來發出 STOP CHANNEL 指令，指出如何停止通道：

靜止

在處理任何現行訊息之後停止通道。

如果啟用共用交談，WebSphere MQ MQI 用戶端會及時察覺停止要求；此時間取決於網路速度。由於對 WebSphere MQ 發出後續呼叫，用戶端應用程式會察覺到停止要求。

強制

立即停止通道。

終止

立即停止通道。如果通道作為處理程序執行，則可以終止通道的處理程序，或者如果通道作為執行緒執行，則為其執行緒。

這是一個多階段的程序。如果使用模式終止，則會嘗試停止伺服器連線通道，首先使用模式靜止，然後使用模式強制，必要的話，使用模式終止。在不同的終止階段期間，用戶端可以收到不同的回覆碼。如果處理程序或執行緒已終止，用戶端會收到通訊錯誤。

傳回應用程式的回覆碼會根據所發出的 MQI 呼叫及所發出的 STOP CHANNEL 指令而有所不同。用戶端將接收 MQRC_CONNECTION_QUIESCING 或 MQRC_CONNECTION_BROKEN 回覆碼。如果用戶端偵測到 MQRC_CONNECTION_QUIESCING，則應該嘗試完成現行交易並終止。這無法與 MQRC_CONNECTION_BROKEN 搭配使用。如果用戶端未完成交易且終止的速度足夠快，則會在幾秒後取得 CONNECTION_BROKEN。具有 MODE (FORCE) 或 MODE (TERMINATE) 的 STOP CHANNEL 指令比具有 MODE (QUIESCE) 更有可能導致 CONNECTION_BROKEN。

通訊

WebSphere MQ MQI 用戶端使用 MQI 通道來與伺服器進行通訊。

必須在 WebSphere MQ MQI 連線的用戶端及伺服器兩端建立通道定義。[定義 MQI 通道](#)中說明如何建立通道定義。

下表顯示可能的傳輸通訊協定：

用戶端平台	LU 6.2	TCP/IP	NetBIOS	SPX
UNIX and Linux 系統	是 ¹	是		
Windows	是	是	是	是
註： 1. Linux (POWER 平台)、Linux (x86-64 平台)、Linux (zSeries s390x 平台) 或 Solaris (x86-64 平台) 上不支援 LU6.2				

[傳輸通訊協定- WebSphere MQ MQI 用戶端與伺服器平台的組合](#) 顯示使用這些傳輸通訊協定的 WebSphere MQ MQI 用戶端與伺服器平台的可能組合。

WebSphere MQ MQI 用戶端上的 WebSphere MQ 應用程式可以採用與佇列管理程式本端相同的方式來使用所有 MQI 呼叫。**MQCONN** 或 **MQCONNX** 建立 WebSphere MQ 應用程式與所選佇列管理程式的關聯，以建立連線控點。然後，連接的佇列管理程式會處理使用該連線控點的其他呼叫。WebSphere MQ MQI 用戶端通訊需要用戶端與伺服器之間的作用中連線，不同於佇列管理程式之間的通訊，後者與連線無關且與時間無關。

傳輸通訊協定是使用通道定義來指定，不會影響應用程式。例如，Windows 應用程式可以透過 TCP/IP 連接至一個佇列管理程式，以及透過 NetBIOS 連接至另一個佇列管理程式。

效能考量

您使用的傳輸通訊協定可能會影響 WebSphere MQ 用戶端和伺服器系統的效能。對於透過慢速電話線的撥號支援，建議使用 WebSphere MQ 通道壓縮。

用戶端連線通道

用戶端連線通道物件的簡介，以及進一步資訊的鏈結。

用戶端連線通道是提供從 WebSphere MQ MQI 用戶端到佇列管理程式之通訊路徑的物件。在分散式佇列中使用用戶端連線通道，以在佇列管理程式與用戶端之間移動訊息。它們會從基礎通訊協定保護應用程式。用戶端可能存在於佇列管理程式的相同或不同的平台上。

如需用戶端連線通道及其使用方式的相關資訊，請參閱 [第 32 頁的『交互通訊』](#)。

接聽器

接聽器是接受來自其他佇列管理程式或用戶端應用程式的網路要求，並啟動相關聯通道的處理程序。

接聽器是接受來自其他佇列管理程式或用戶端應用程式的網路要求，並啟動相關聯通道的處理程序。可以使用 `runmqtsr` 控制指令來啟動接聽器程序。

接聽器物件是 WebSphere MQ 物件，可讓您從佇列管理程式範圍內管理接聽器程序的啟動和停止。透過定義接聽器物件的屬性，您可以執行下列動作：

- 配置接聽器程序。
- 指定當佇列管理程式啟動和停止時，接聽器處理程序是否自動啟動和停止。

在 **WebSphere MQ for z/OS** 上不支援接聽器物件。

服務

服務物件是定義當佇列管理程式啟動或停止時要執行之程式的一種方式。

程式可以是下列其中一種類型：

伺服器

伺服器是將參數 `SERVTYPE` 指定為 `SERVER` 的服務物件。伺服器服務物件是在啟動指定的佇列管理程式時將執行的程式定義。伺服器處理程序只能同時執行一個實例。執行時，可以使用 `MQSC` 指令 `DISPLAY SVSTATUS` 來監視伺服器處理程序的狀態。一般而言，伺服器服務物件是程式的定義，例如無法傳送的郵件處理程式或觸發監視器，但可執行的程式不限於 WebSphere MQ 所提供的程式。此外，可以定義伺服器服務物件，以包含在關閉指定的佇列管理程式以結束程式時將執行的指令。

指令

指令是將參數 `SERVTYPE` 指定為 `COMMAND` 的服務物件。指令服務物件是在啟動或停止指定的佇列管理程式時將執行的程式定義。一個指令處理程序的多個實例可以同時執行。指令服務物件不同於伺服器服務物件，因為一旦執行程式，佇列管理程式將不會監視程式。通常指令服務物件是短暫存活且將執行特定作業 (例如啟動一或多個其他作業) 的程式定義。

相關概念

[使用服務](#)

主題物件

主題物件是一個 WebSphere MQ 物件，可讓您將特定非預設屬性指派給主題。

主題是由發佈或訂閱特定主題字串的應用程式所定義。主題字串可以使用正斜線字元 (/) 來區隔主題，以指定主題的階層。這可以透過主題樹狀結構來視覺化。例如，如果應用程式發佈至主題字串 `/Sport/`

American Football 及 /Sport/Soccer，則會建立主題樹狀結構，其母節點 Sport 具有兩個子節點 American Football 及 Soccer。

主題會從在其主題樹狀結構中找到的第一個母項管理節點繼承其屬性。如果特定主題樹狀結構中沒有管理主題節點，則所有主題都會從基本主題物件 SYSTEM.BASE.TOPIC。

您可以在主題樹狀結構的任何節點上建立主題物件，方法是在主題物件的 TOPICSTR 屬性中指定該節點的主題字串。您也可以定義管理主題節點的其他屬性。如需這些屬性的相關資訊，請參閱 [MQSC 指令](#) 或 [自動化管理作業](#)。依預設，每一個主題物件都會從其最接近的上層管理主題節點繼承其屬性。

主題物件也可以用來向應用程式開發人員隱藏完整主題樹狀結構。如果針對主題 /Sport/American Football 建立名為 FOOTBALL.US 的主題物件，則應用程式可以發佈或訂閱名為 FOOTBALL.US 的物件，而不是具有相同結果的字串 /Sport/American Football。

如果您在主題物件的主題字串內輸入 #、+、/ 或 * 字元，則會將該字元視為字串內的一般字元，並視為與主題物件相關聯之主題字串的一部分。

如需主題物件的相關資訊，請參閱 [WebSphere 簡介 MQ 發佈/訂閱傳訊](#)。

命名 IBM WebSphere MQ 物件

WebSphere MQ 物件採用的命名慣例取決於物件。與 IBM WebSphere MQ 搭配使用的機器名稱及使用者 ID 也會受到一些命名限制。

佇列管理程式的每一個實例都以其名稱來識別。此名稱在交互連接的佇列管理程式網路內必須是唯一的，因此一個佇列管理程式可以明確地識別任何給定訊息傳送至其中的目標佇列管理程式。

對於其他類型的物件，每一個物件都有一個相關聯的名稱，且可以用該名稱來參照。這些名稱在一個佇列管理程式及物件類型中必須是唯一的。例如，您可以具有同名的佇列及處理程序，但不能具有同名的兩個佇列。

在 WebSphere MQ 中，名稱最多可以有 48 個字元，但 *channels* (最多 20 個字元) 除外。如需命名 IBM WebSphere MQ 物件的相關資訊，請參閱 [第 103 頁的『IBM WebSphere MQ 物件的命名規則』](#)。

與 IBM WebSphere MQ 搭配使用的機器名稱及使用者 ID 也會受到一些命名限制：

- 請確定機器名稱不包含任何空格。IBM WebSphere MQ 不支援包含空格的機器名稱。如果您在這類機器上安裝 IBM WebSphere MQ，則無法建立任何佇列管理程式。
- 對於 IBM WebSphere MQ 授權，使用者 ID 和群組的名稱不得超過 20 個字元 (不接受空格)。
- 如果用戶端以包含 @ 字元 (例如 abc@d.) 的使用者 ID 執行，則 WebSphere MQ for Windows 伺服器不支援 Windows 用戶端的連線。

相關概念

[第 105 頁的『瞭解 IBM WebSphere MQ 檔名』](#)

每一個 WebSphere MQ 佇列管理程式、佇列、程序定義、名單、通道、用戶端連線通道、接聽器、服務及鑑別資訊物件都由一個檔案代表。因為物件名稱不一定是有效的檔名，所以佇列管理程式會在必要時將物件名稱轉換成有效的檔名。

相關參考

[第 103 頁的『IBM WebSphere MQ 物件的命名規則』](#)

IBM WebSphere MQ 物件名稱具有長度上限，且區分大小寫。並非每個物件類型都支援所有字元，而且許多物件都有關於名稱唯一性的規則。

IBM WebSphere MQ 物件的命名規則

IBM WebSphere MQ 物件名稱具有長度上限，且區分大小寫。並非每個物件類型都支援所有字元，而且許多物件都有關於名稱唯一性的規則。

IBM WebSphere MQ 物件有許多不同類型，且每一種類型中的物件都可以具有相同的名稱，因為它們存在於個別物件名稱空間中：例如，本端佇列及傳送端通道都可以具有相同的名稱。但物件名稱不能與相同名稱空間中的另一個物件名稱相同：例如，本端佇列的名稱不能與模型佇列的名稱相同，且傳送端通道的名稱不能與接收端通道的名稱相同。

下列 IBM WebSphere MQ 物件存在於個別物件名稱空間中：

- 鑑別資訊
- 通道
- 用戶端通道
- 接聽器
- 名稱清單
- 處理程序
- 佇列
- 服務
- 儲存體類別
- 訂閱
- 主題

物件名稱的字元長度

一般而言，IBM WebSphere MQ 物件名稱最長可為 48 個字元。此規則適用於下列物件：

- 鑑別資訊
- 叢集
- 接聽器
- 名稱清單
- 程序定義
- 佇列
- 佇列管理程式
- 服務
- 訂閱
- 主題

有一些限制：

1. 通道物件名稱及用戶端連線通道名稱的長度上限為 20 個字元。如需通道的相關資訊，請參閱 [定義通道](#)。
2. 主題字串最多可以有 10240 個位元組。所有 IBM WebSphere MQ 物件名稱都區分大小寫。
3. 儲存類別名稱的長度上限為 8 個字元。
4. CF 結構名稱的長度上限為 12 個字元。

物件名稱中的字元

IBM WebSphere MQ 物件名稱的有效字元為：

個字元後	限制
大寫 A-Z	• 無
小寫 a-z	<ul style="list-style-type: none"> • 在 MQSC Script 中，必須以單引號括住具有小寫字元的名稱。這可防止小寫字元轉換成大寫。 • 使用 EBCDIC Katakana 的系統無法在物件名稱中使用小寫 a-z 字元。
數字 0-9	• 無
句點 (.)	• 無

個字元後	限制
底線 (_)	<ul style="list-style-type: none"> distributed 無
正斜線 (/)	<ul style="list-style-type: none"> Windows 在 Windows 系統上，佇列管理程式名稱的第一個字元不能是正斜線。
百分比符號 (%)	<ul style="list-style-type: none"> distributed 無

還有一些關於物件名稱字元的一般規則：

1. 不容許前導或內嵌空白。
2. 不接受國家語言字元。
3. 任何小於完整欄位長度的名稱都可以用空白填補右邊。佇列管理程式所傳回的所有簡稱一律以空白填補在右側。

佇列名稱

佇列名稱有兩個部分：

- 佇列管理程式的名稱
- 佇列管理程式已知的佇列本端名稱

佇列名稱的每一個部分長度為 48 個字元。

若要參照本端佇列，您可以省略佇列管理程式的名稱 (方法是將它取代為空白字元或使用前導空值字元)。不過，IBM WebSphere MQ 傳回給程式的所有佇列名稱都包含佇列管理程式的名稱。

若要參照遠端佇列，程式必須在完整佇列名稱中包含佇列管理程式的名稱，或必須有遠端佇列的本端定義。

當應用程式使用佇列名稱時，該名稱可以是本端佇列的名稱 (或別名) 或遠端佇列的本端定義名稱，但應用程式不需要知道該名稱，除非它需要從佇列取得訊息 (當佇列必須是本端佇列時)。當應用程式開啟佇列物件時，MQOPEN 呼叫會執行名稱解析函數，以決定要對哪個佇列執行後續作業。其重要性在於應用程式對於在佇列管理程式網路中的特定位置所定義的特定佇列沒有內建相依關係。因此，如果系統管理者重新定位網路中的佇列，並變更其定義，則不需要變更使用那些佇列的應用程式。

保留物件名稱

以 SYSTEM 開頭的物件名稱。保留給佇列管理程式所定義的物件。您可以使用 **Alter**、**Define** 及 **Replace** 指令來變更這些物件定義，以符合您的安裝。定義給 IBM WebSphere MQ 的名稱會完整列出在佇列名稱中。

相關工作

[選擇安裝名稱](#)

瞭解 IBM WebSphere MQ 檔名

每一個 WebSphere MQ 佇列管理程式、佇列、程序定義、名單、通道、用戶端連線通道、接聽器、服務及鑑別資訊物件都由一個檔案代表。因為物件名稱不一定是有效的檔名，所以佇列管理程式會在必要時將物件名稱轉換成有效的檔名。

佇列管理程式目錄的預設路徑如下：

- 在 WebSphere MQ 配置資訊中定義的字首：
 - 在 Windows 32 位元系統上，預設字首為 C:\Program Files\IBM\WebSphere MQ。在 Windows 64 位元系統上，預設字首為 C:\Program Files\IBM\WebSphere MQ (x86)\。這會配置在 mqs.ini 配置檔的 DefaultPrefix 段落中。

- 在 UNIX and Linux 系統上，預設字首為 /var/mqm。這會配置在 mqs.ini 配置檔的 DefaultPrefix 段落中。

如果可用的話，可以使用「IBM WebSphere MQ 探險家」中的 WebSphere MQ 內容頁面來變更字首，否則請手動編輯 mqs.ini 配置檔。

- 佇列管理程式名稱會轉換成有效的目錄名稱。例如，佇列管理程式：

```
queue.manager
```

將表示為：

```
queue!manager
```

此程序稱為 名稱轉換。

在 WebSphere MQ 中，您可以為佇列管理程式提供最多包含 48 個字元的名稱。

例如，您可以命名佇列管理程式：

```
QUEUE.MANAGER.ACCOUNTING.SERVICES
```

不過，每一個佇列管理程式都由一個檔案代表，且檔名的長度上限以及名稱中可以使用的字元有一些限制。因此，會自動轉換代表物件的檔案名稱，以符合檔案系統的需求。

控管佇列管理程式名稱轉換的規則如下：

1. 轉換個別字元：

- 從。到!
- From / to &

2. 如果名稱仍然無效：

- a. 將它截斷為 8 個字元
- b. 附加三個字元的數值字尾

例如，假設預設字首及名為 queue.manager 的佇列管理程式：

- 在 WebSphere MQ for Windows with NTFS 或 FAT32 中，佇列管理程式名稱會變成：

```
c:\Program Files\IBM\WebSphere MQ\mqgrs\queue!manager
```

- 在具有 FAT 的 WebSphere MQ for Windows 中，佇列管理程式名稱會變成：

```
c:\Program Files\IBM\WebSphere MQ\mqgrs\queue!ma
```

- 在 WebSphere MQ for UNIX and Linux 系統中，佇列管理程式名稱會變成：

```
/var/mqm/mqgrs/queue!manager
```

在不區分大小寫的檔案系統上，轉換演算法也會區分只有大小寫不同的名稱。

物件名稱轉換

物件名稱不一定是有效的檔案系統名稱。您可能需要轉換物件名稱。所使用的方法不同於佇列管理程式名稱的方法，因為雖然每一部機器上只有少數佇列管理程式名稱，但每一部佇列管理程式可能會有大量其他物件。檔案系統中代表佇列、程序定義、名稱清單、通道、用戶端連線通道、接聽器、服務及鑑別資訊物件。

當轉換程序產生新名稱時，與原始物件名稱沒有簡式關係。您可以使用 `dsqmqls` 指令，在實際物件名稱與轉換物件名稱之間進行轉換。

管理物件

如何建立、變更、顯示及刪除物件的概觀。


如需的相關資訊，請參閱 [第 89 頁的『物件』](#)。

除了動態佇列之外，這些物件必須先定義給佇列管理程式，您才能使用它們。




您可以使用下列指令來定義及管理物件：

- [可程式化指令格式參照及 自動化管理作業](#) 中說明的 PCF 指令
- [MQSC 指令](#) 中說明的 MQSC 指令
- WebSphere MQ 探險家 (僅適用於 Intel 系統 Windows、UNIX 及 Linux)

您也可以使用下列方法來管理物件：

- 控制指令，從鍵盤輸入。請參閱 [控制指令](#)。
- 程式中的 IBM WebSphere MQ 管理介面 (MQAI) 呼叫。請參閱 [WebSphere MQ 管理介面 \(MQAI\)](#)。
-  僅限 IBM WebSphere MQ for Windows：
 - 程式中的 MQAI 元件物件模型 (COM) 呼叫
 - Windows 預設配置應用程式

您也可以顯示或變更物件的屬性，或刪除物件。

   對於 Windows、UNIX and Linux 系統上的 WebSphere MQ 指令序列，您可以使用 MQSC 機能來執行檔案中所保留的一系列指令。

物件屬性

物件的內容由其屬性定義。有些您可以指定，有些您只能檢視。

例如，佇列可容納的訊息長度上限由其 *MaxMsgLength* 屬性定義；您可以在建立佇列時指定此屬性。*DefinitionType* 屬性指定如何建立佇列；您只能顯示此屬性。

在 WebSphere MQ 中，有兩種參照屬性的方式：

- 使用其 PCF 名稱，例如 *MaxMsgLength*。
- 使用其 MQSC 指令名稱，例如 MAXMSGL。

本手冊主要說明如何使用 MQSC 指令來指定屬性，因此它會使用 MQSC 指令名稱而非 PCF 名稱來參照大部分屬性。

叢集

您可以將叢集中的佇列管理程式分組。叢集中的佇列管理程式可以讓它們所管理的佇列可供叢集中的所有其他佇列管理程式使用。任何佇列管理程式都可以將訊息傳送至相同叢集中的任何其他佇列管理程式，而不需要標準分散式佇列作業所需的許多物件定義。

在使用分散式佇列的傳統 WebSphere MQ 網路中，每個佇列管理程式都是獨立的。如果一個佇列管理程式需要將訊息傳送至另一個佇列管理程式，它必須定義傳輸佇列、遠端佇列管理程式的通道，以及要傳送訊息的每個佇列的遠端佇列定義。

叢集是佇列管理程式的群組，其設定方式可讓佇列管理程式透過單一網路直接彼此通訊，而不需要傳輸佇列、通道及遠端佇列定義。

叢集中的每一個佇列管理程式都有單一傳輸佇列，可從中將訊息傳輸至叢集中的任何其他佇列管理程式。

相關概念

設計叢集

瞭解何謂叢集及其運作方式。

相關工作

[配置佇列管理程式叢集](#)

系統預設物件

系統預設物件的簡介，以及進一步資訊的鏈結。

系統預設物件是每當建立佇列管理程式時自動建立的一組物件定義。您可以複製並修改任何這些物件定義，以在安裝時用於應用程式中。

預設物件名稱具有詞幹 SYSTEM；例如，預設本端佇列為 SYSTEM.DEFAULT.LOCAL.QUEUE，且預設接收端通道是 SYSTEM.DEF.RECEIVER。您無法重新命名這些物件；需要這些名稱的預設物件。

當您定義物件時，會從適當的預設物件複製您未明確指定的任何屬性。例如，如果您定義本端佇列，則您未指定的那些屬性會從預設佇列 SYSTEM.DEFAULT.LOCAL.QUEUE。

如需系統預設值的相關資訊，請參閱 [系統及預設物件](#)。

IBM WebSphere MQ 多重播送

IBM WebSphere MQ 多重播送提供低延遲、高扇出、可靠的多重播送傳訊功能。

多重播送是發佈/訂閱傳訊的有效形式，因為它可以調整為大量訂閱者，而不會對效能造成不利影響。為了進行可靠的「多重播送」傳訊，WebSphere MQ 利用確認通知、負值確認通知及序號來達成高扇出的低延遲傳訊。

「WebSphere MQ 多重播送」的公平遞送可以達到近乎同時的遞送，確保沒有任何收件者可以佔到便宜。因為「WebSphere MQ 多重播送」使用網路來遞送訊息，所以不需要發佈/訂閱引擎來扇出資料。將主題對映至群組位址之後，不需要佇列管理程式，因為發佈者和訂閱者可以在對等模式下運作。這樣可以降低佇列管理程式伺服器的負載，且佇列管理程式伺服器就不會再成為可能的故障點。

起始多重播送概念

WebSphere MQ 多重播送可以使用「通訊資訊 (COMMINFO)」物件輕鬆整合至現有的系統和應用程式。兩個 TOPIC 物件欄位可啟用現有 TOPIC 物件的快速配置，以支援或忽略多重播送資料流量。

多重播送所需的物件

下列資訊是 WebSphere MQ 多重播送所需的兩個物件的簡要概觀：

COMMINFO 物件

COMMINFO 物件包含與多重播送傳輸相關聯的屬性。如需 COMMINFO 物件參數的相關資訊，請參閱 [DEFINE COMMINFO](#)。

「必須」設定的唯一 COMMINFO 欄位是 COMMINFO 物件的名稱。然後會使用這個名稱來識別主題的 COMMINFO 物件。必須檢查 COMMINFO 物件的 **GRPADDR** 欄位，以確保該值是有效的多重播送群組位址。

TOPIC 物件

主題是發佈/訂閱訊息中所發佈資訊的主旨，而主題是透過建立 TOPIC 物件來定義。如需 TOPIC 物件參數的相關資訊，請參閱 [DEFINE TOPIC](#)。

透過變更下列 TOPIC 物件參數的值，可以將現有主題與多重播送搭配使用：**COMMINFO** 及 **MCAST**。

- **COMMINFO** 此參數指定多重播送通訊資訊物件的名稱。
- **MCAST** 此參數指定主題樹狀結構中的這個位置是否容許多重播送。依預設，**MCAST** 會設為 **ASAPARENT**，表示主題的多重播送屬性繼承自母項。將 **MCAST** 設為 **ENABLED** 可容許此節點上的多重播送資料流量。

多重播送網路及主題

下列資訊是具有不同訂閱類型及主題定義之訂閱的概觀。這些範例都假設 TOPIC 物件 **COMMINFO** 參數設為有效 COMMINFO 物件的名稱：

已啟用多重播送的主題集

如果主題字串 **MCAST** 參數設為 **ENABLED**，則容許具有多重播送功能用戶端的訂閱，並建立多重播送訂閱，除非：

- 它是來自具有多重播送功能之用戶端的可延續訂閱。
- 它是來自具有多重播送功能之用戶端的非受管理訂閱。
- 它是來自具有非多重播送功能的用戶端的訂閱。

在這些情況下，會建立非多重播送訂閱，並將訂閱降級為正常發佈/訂閱。

已停用主題設為多重播送

如果主題字串 **MCAST** 參數設為 **DISABLED**，一律會建立非多重播送訂閱，且訂閱降級為正常發佈/訂閱。

主題僅設為多重播送

如果主題字串 **MCAST** 參數設為 **ONLY**，則容許從具有多重播送功能的用戶端進行訂閱，並建立多重播送訂閱，除非：

- 它是可延續訂閱: 拒絕可延續訂閱，原因碼為 [2436 \(0984\) \(RC2436\):MQRC_DURABILITY_NOT_ALLOWED](#)
- 它是非受管理訂閱: 拒絕非受管理訂閱，原因碼為 [2046 \(07FE\) \(RC2046\):MQRC_OPTIONS_ERROR](#)
- 它是來自具有非多重播送功能之用戶端的訂閱: 這些訂閱遭到拒絕，原因碼為 [2560 \(0A00\) \(RC2560\):MQRC_MULTICAST_only](#)
- 它是來自本端連結應用程式的訂閱: 這些訂閱遭到拒絕，原因碼為 [2560 \(0A00\) \(RC2560\):MQRC_MULTICAST_ONLY](#)

安全

在 IBM WebSphere MQ 中，有數種提供安全的方法: 授權服務介面; 使用者撰寫或協力廠商通道結束程式; 使用 Secure Sockets Layer (SSL) 的通道安全、通道鑑別記錄及訊息安全。

授權服務介面

使用 MQI 呼叫、指令及存取物件的授權是由 **物件權限管理程式 (OAM)** 提供，依預設會啟用。IBM WebSphere MQ 實體的存取權是透過 IBM WebSphere MQ 使用者群組和 OAM 來控制。管理者可以視需要使用指令行介面來授與或撤銷授權。

如需建立授權服務元件的相關資訊，請參閱 [在 Windows 上設定安全 UNIX and Linux 系統](#)。

使用者撰寫或協力廠商通道結束程式

通道可以使用使用者撰寫或協力廠商通道結束程式。如需相關資訊，請參閱 [傳訊通道的通道結束程式](#)。

使用 SSL 的通道安全

Secure Sockets Layer (SSL) 通訊協定提供業界標準通道安全，可防止竊聽、竄改及模擬。

SSL 使用公開金鑰和對稱技術來提供訊息機密性和完整性以及交互鑑別。

如需 IBM WebSphere MQ 中安全的綜合性檢閱 (包括 SSL 的詳細資訊)，請參閱 [安全](#)。如需 SSL 的概觀 (包括本節中所說明之指令的指標)，請參閱 [加密安全通訊協定: SSL 及 TLS](#)。

通道鑑別記錄

使用通道鑑別記錄，在通道層次對授與連接系統的存取權進行精確控制。如需相關資訊，請參閱 [通道鑑別記錄](#)。

訊息安全

使用 Advanced Message Security (這是 IBM WebSphere MQ 的個別安裝及授權元件)，為使用 IBM WebSphere MQ 傳送及接收的訊息提供加密保護。請參閱 [WebSphere MQ Advanced Message Security](#)。

相關工作

安全

規劃安全需求

用戶端及伺服器

簡介 IBM WebSphere MQ 如何支援其應用程式的主從架構配置。

IBM WebSphere MQ MQI 用戶端 是一個元件，可讓在系統上執行的應用程式對在另一個系統上執行的佇列管理程式發出 MQI 呼叫。呼叫的輸出會傳回給用戶端，而用戶端會將它傳回給應用程式。

IBM WebSphere MQ 伺服器 是為一個以上用戶端提供佇列作業服務的佇列管理程式。所有 IBM WebSphere MQ 物件 (例如佇列) 都只存在於佇列管理程式機器 (IBM WebSphere MQ 伺服器機器) 上，而不存在於用戶端上。IBM WebSphere MQ 伺服器也可以支援本端 IBM WebSphere MQ 應用程式。

IBM WebSphere MQ 伺服器與一般佇列管理程式之間的差異是伺服器與每一個用戶端都有專用通訊鏈結。如需為用戶端及伺服器建立通道的相關資訊，請參閱 [使用分散式佇列連接應用程式](#)。

如需一般用戶端的相關資訊，請參閱 [第 110 頁的『IBM WebSphere MQ MQI 用戶端的概觀』](#)。

主從架構環境中的 IBM WebSphere MQ 應用程式

鏈結至伺服器時，用戶端 IBM WebSphere MQ 應用程式可以使用與本端應用程式相同的方式發出大部分 MQI 呼叫。用戶端應用程式會發出 MQCONN 呼叫，以連接至指定的佇列管理程式。然後，此佇列管理程式會處理指定從連接要求傳回之連線控點的任何其他 MQI 呼叫。

您必須將應用程式鏈結至適當的用戶端程式庫。請參閱 [建置 WebSphere MQ MQI 用戶端的應用程式](#)。

相關概念

[第 115 頁的『交易管理及支援』](#)

交易管理及 WebSphere MQ 如何支援交易的簡介。

[第 116 頁的『延伸佇列管理程式機能』](#)

您可以使用使用者結束程式、API 結束程式或可安裝的服務來延伸佇列管理程式機能。

IBM WebSphere MQ MQI 用戶端的概觀

WebSphere MQ MQI 用戶端 是 IBM WebSphere MQ 產品的元件，可以安裝在沒有執行任何佇列管理程式的系統上。

使用 IBM WebSphere MQ MQI 用戶端，在與用戶端相同的系統上執行的應用程式可以連接至在另一個系統上執行的佇列管理程式。應用程式可以對該佇列管理程式發出 MQI 呼叫。這類應用程式稱為 *WebSphere MQ MQI 用戶端應用程式*，而佇列管理程式稱為 *伺服器佇列管理程式*。

IBM WebSphere MQ MQI 用戶端應用程式及伺服器佇列管理程式會使用 MQI 通道來彼此通訊。當用戶端應用程式發出 **MQCONN** 或 **MQCONNX** 呼叫來連接佇列管理程式時，會啟動 MQI 通道；當用戶端應用程式發出 **MQDISC** 呼叫來切斷與佇列管理程式的連線時，會結束 MQI 通道。MQI 呼叫流程在 MQI 通道上的一個方向的輸入參數，以及輸出參數在相反方向的輸入參數。

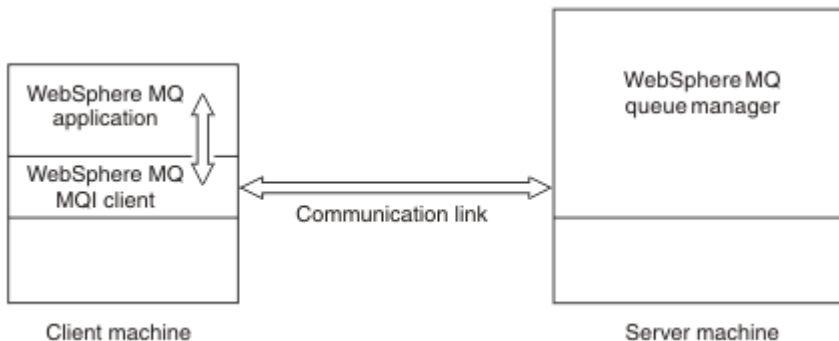


圖 30: 用戶端與伺服器之間的鏈結

可以使用下列平台。這些組合視您使用的 IBM WebSphere MQ 產品而定，並在 [第 112 頁的『IBM WebSphere MQ 用戶端的平台支援』](#) 中說明。

IBM WebSphere MQ MQI 用戶端

UNIX and Linux
Windows

HP Integrity NonStop Server

IBM WebSphere MQ 伺服器

UNIX and Linux
Windows

HP Integrity NonStop Server

MQI 可供在用戶端平台上執行的應用程式使用；佇列及其他 IBM WebSphere MQ 物件會保留在您已安裝在伺服器上的佇列管理程式上。

您要在 IBM WebSphere MQ MQI 用戶端環境中執行的應用程式必須先與相關用戶端程式庫鏈結。當應用程式發出 MQI 呼叫時，IBM WebSphere MQ MQI 用戶端會將要求導向至佇列管理程式，並會處理該要求，以及從該佇列管理程式將回覆傳回 IBM WebSphere MQ MQI 用戶端。

在執行時期會動態建立應用程式與 IBM WebSphere MQ MQI 用戶端之間的鏈結。

您也可以使用 IBM WebSphere MQ classes for .NET、IBM WebSphere MQ classes for Java 或 IBM WebSphere MQ classes for Java Message Service (JMS) 來開發用戶端應用程式。您可以在 UNIX、Linux 和 Windows 平台上使用 Java 和 JMS 用戶端。這裡未說明 Java 和 JMS 的用法。如需如何安裝、配置及使用 IBM WebSphere MQ classes for Java 和 IBM WebSphere MQ classes for JMS 的完整詳細資料，請參閱 [使用適用於 Java 的 WebSphere MQ 類別](#) 和 [使用適用於 JMS 的 WebSphere MQ 類別](#)。

相關概念

[第 111 頁的『為何使用 IBM WebSphere MQ 用戶端?』](#)

使用 IBM WebSphere MQ 用戶端是實作 IBM WebSphere MQ 傳訊及佇列作業的有效方式。

[第 112 頁的『如何設定 IBM WebSphere MQ MQI 用戶端?』](#)

請遵循下列指示來設定用戶端。

[第 113 頁的『何謂延伸交易式用戶端?』](#)

WebSphere MQ 延伸交易式用戶端可以在外部交易管理程式的控制下，更新另一個資源管理程式所管理的資源。

[第 114 頁的『用戶端如何連接至伺服器』](#)

用戶端使用 MQCONN 或 MQCONNX 連接至伺服器，並透過通道進行通訊。

為何使用 IBM WebSphere MQ 用戶端?

使用 IBM WebSphere MQ 用戶端是實作 IBM WebSphere MQ 傳訊及佇列作業的有效方式。

您可以讓應用程式使用在一部機器上執行的 MQI，以及在不同機器 (實體或虛擬) 上執行的佇列管理程式。這樣做的好處如下：

- 用戶端機器上不需要完整的 WebSphere MQ 實作。
- 減少用戶端系統上的硬體需求。
- 減少系統管理需求。
- 在用戶端上執行的 WebSphere MQ 應用程式可以連接至不同系統上的多個佇列管理程式。
- 可以使用使用不同傳輸通訊協定的替代通道。

相關參考

[第 112 頁的『哪些應用程式在 IBM WebSphere MQ MQI 用戶端上執行?』](#)

用戶端環境中支援完整 MQI。

[第 112 頁的『IBM WebSphere MQ 用戶端的平台支援』](#)

所有伺服器平台上的 IBM WebSphere MQ 都接受來自 UNIX 或 Linux 系統及 Windows 上 IBM WebSphere MQ MQI 用戶端的用戶端連線。

哪些應用程式在 IBM WebSphere MQ MQI 用戶端上執行？
用戶端環境中支援完整 MQI。

這可將 IBM WebSphere MQ MQI 用戶端上的應用程式鏈結至 MQIC 程式庫，而非 MQI 程式庫，將幾乎任何 WebSphere MQ 應用程式配置成在 IBM WebSphere MQ MQI 用戶端系統上執行。異常狀況如下：

- 具有信號的 MQGET
- 需要與其他資源管理程式同步點協調的應用程式必須使用延伸交易式用戶端

如果啟用先讀，為了改善非持續性傳訊效能，並非所有 MQGET 選項都可用。此表格顯示容許的選項，以及在 MQGET 呼叫之間是否可以變更這些選項。

表 9: 啟用先讀時允許 MQGET 選項

	已啟用先讀且可以在 MQGET 呼叫之間變更時允許	已啟用先讀但無法在 MQGET 呼叫之間變更時允許 ¹	啟用先讀時不允許的 MQGET 選項 ²
MQGET MD 值	訊息 ID ³ CorrelId ³	編碼 CodedCharSetId	
MQGET MQGMO 選項	MQGMO_WAIT MQGMO_NO_WAIT MQGMO_FAIL_IF_QUIESCING MQGMO_BROWSE_FIRST ⁴ MQGMO_BROWSE_NEXT ⁴ MQGMO_BROWSE_MESSAGE_UNDER_CURSOR ⁴	MQGMO_SYNCPOINT_IF_PERSISTENT MQGMO_NO_SYNCPOINT MQGMO_ACCEPT_TRUNCATED_MSG MQGMO_CONVERT MQGMO_LOGICAL_ORDER MQGMO_COMPLETE_MSG MQGMO_ALL_MSGS_AVAILABLE MQGMO_ALL_SEGMENTS_AVAILABLE MQGMO_MARK_BROWSE_HANDLE MQGMO_MARK_BROWSE_CO_OP MQGMO_UNMARK_BROWSE_CO_OP MQGMO_UNMARK_BROWSE_HANDLE MQGMO_UNMARKED_BROWSE_MSG MQGMO_PROPERTIES_FORCE_MQRFH2 MQGMO_NO_PROPERTIES MQGMO_PROPERTIES_IN_HANDLE MQGMO_PROPERTIES_COMPATIBILITY	MQGMO_SET_SIGNAL MQGMO_同步點 MQGMO_MARK_SKIP_BACKOUT MQGMO_MSG_UNDER_CURSOR ⁴ MQGMO_LOCK MQGMO_UNLOCK
MQGMO 值		MsgHandle	

1. 如果在 MQGET 呼叫之間變更了這些選項，則會傳回 MQRC_OPTIONS_CHANGED 原因碼。
2. 如果在第一個 MQGET 呼叫上指定這些選項，則會停用先讀功能。如果在後續的 MQGET 呼叫上指定了這些選項，則會傳回原因碼 MQRC_OPTIONS_ERROR。
3. 用戶端應用程式需要意識到，如果 MsgId 和 CorrelId 值在 MQGET 呼叫之間變更，則具有先前值的訊息可能已經傳送至用戶端，並保留在用戶端先讀緩衝區中，直到被取用（或自動清除）為止。
4. 第一個 MQGET 呼叫決定在啟用先讀功能時，是否要從佇列中瀏覽或取得訊息。如果應用程式嘗試同時執行瀏覽與取得動作，將傳回原因碼 MQRC_OPTIONS_CHANGED。
5. MQGMO_MSG_UNDER_CURSOR 不能與先讀功能一起使用。在啟用先讀功能之後，可瀏覽或取得訊息，但不能同時執行這兩項動作。

在 IBM WebSphere MQ MQI 用戶端上執行的應用程式可以同時連接至多個佇列管理程式，或在 MQCONN 或 MQCONNX 呼叫中使用具有星號 (*) 的佇列管理程式名稱 (請參閱 [將 IBM WebSphere MQ MQI 用戶端應用程式連接至佇列管理程式中的範例](#))。

IBM WebSphere MQ 用戶端的平台支援

所有伺服器平台上的 IBM WebSphere MQ 都接受來自 UNIX 或 Linux 系統及 Windows 上 IBM WebSphere MQ MQI 用戶端的用戶端連線。

安裝為基本產品和伺服器的 WebSphere MQ 可以接受來自下列平台上 IBM WebSphere MQ MQI 用戶端的連線：

- HP Integrity NonStop Server
- UNIX and Linux 系統
- Windows

用戶端連線會受到編碼字集 ID (CCSID) 及通訊協定的差異。

如何設定 IBM WebSphere MQ MQI 用戶端？

請遵循下列指示來設定用戶端。

若要設定 IBM WebSphere MQ MQI 用戶端，您必須已安裝並運作 IBM WebSphere MQ 伺服器，且用戶端將會連接此伺服器。設定用戶端所涉及的步驟如下：

1. 檢查您是否具有適用於 IBM WebSphere MQ MQI 用戶端的平台，以及軟硬體是否滿足需求。平台支援在第 112 頁的『[IBM WebSphere MQ 用戶端的平台支援](#)』中說明。
2. 決定如何在用戶端工作站上安裝 IBM WebSphere MQ，然後遵循特定用戶端與伺服器平台組合的指示。安裝說明於 [安裝 IBM WebSphere MQ 用戶端](#)。
3. 請確定已配置並連接您的通訊鏈結。[配置伺服器與用戶端之間的連線](#) 中說明了通訊鏈結的配置。
4. 請檢查您的安裝是否正常運作。[驗證用戶端安裝中說明驗證安裝](#)。
5. 當您已驗證 IBM WebSphere MQ MQI 用戶端安裝架構時，請考量是否必須保護用戶端安全。用戶端安全在 [設定 IBM WebSphere MQ MQI 用戶端安全](#) 中有說明。
6. 設定您要在用戶端上執行之 IBM WebSphere MQ 應用程式所需的 IBM WebSphere MQ MQI 用戶端與伺服器之間的通道。[定義 MQI 通道](#) 中說明了設定通道。如果您使用 SSL，還有一些其他考量。[指定 MQI 通道使用 SSL](#) 中會說明這些考量。您可能需要使用 IBM WebSphere MQ MQI 用戶端配置檔或 IBM WebSphere MQ 環境變數來設定通道。IBM WebSphere MQ 環境變數在 [使用 IBM WebSphere MQ 環境變數](#) 中有說明。
7. IBM WebSphere MQ 應用程式在 [開發應用程式](#) 中有完整說明。
8. 在 IBM WebSphere MQ MQI 用戶端環境中設計、建置及執行應用程式時，會考量佇列管理程式環境的一些差異。如需這些差異的相關資訊，請參閱：
 - [在用戶端應用程式中使用訊息佇列介面 \(MQI\)](#)
 - [建置 WebSphere MQ MQI 用戶端的應用程式](#)
 - [將 IBM WebSphere MQ MQI 用戶端應用程式連接至佇列管理程式](#)
 - [解決 IBM WebSphere MQ MQI 用戶端的問題](#)

何謂延伸交易式用戶端？

WebSphere MQ 延伸交易式用戶端可以在外部交易管理程式的控制下，更新另一個資源管理程式所管理的資源。

如果您不熟悉交易管理的概念，請參閱第 115 頁的『[交易管理及支援](#)』。

請注意，XA 交易式用戶端現在是作為 WebSphere MQ 的一部分提供。

用戶端應用程式可以參與由它所連接的佇列管理程式所管理的工作單元。在工作單元內，用戶端應用程式可以將訊息放置在該佇列管理程式所擁有的佇列中，以及從中取得訊息。然後用戶端應用程式可以使用 **MQCMIT** 呼叫來確定工作單元，或使用 **MQBACK** 呼叫來取消工作單元。不過，在相同的工作單元內，用戶端應用程式無法更新另一個資源管理程式的資源，例如 DB2 資料庫的表格。使用 WebSphere MQ 延伸交易式用戶端會移除此限制。

WebSphere MQ 延伸交易式用戶端是具有有一些額外功能的 IBM WebSphere MQ MQI client。使用此功能，在相同的工作單元內，用戶端應用程式可以執行下列作業：

- 將訊息放置在它所連接的佇列管理程式所擁有的佇列中，並從中取得訊息
- 更新非 WebSphere MQ 佇列管理程式的資源管理程式資源

此工作單元必須由與用戶端應用程式在相同系統上執行的外部交易管理程式管理。用戶端應用程式所連接的佇列管理程式無法管理工作單元。這表示佇列管理程式只能作為資源管理程式，而不能作為交易管理程式。它也表示用戶端應用程式只能使用外部交易管理程式所提供的應用程式設計介面 (API) 來確定或取消工作單元。因此，用戶端應用程式無法使用 MQI 呼叫、**MQBEGIN**、**MQCMIT** 及 **MQBACK**。

外部交易管理程式會使用連接至佇列管理程式的用戶端應用程式所使用的相同 MQI 通道，來與作為資源管理程式的佇列管理程式進行通訊。不過，在失敗之後的回復狀況中，當沒有應用程式正在執行時，交易管理程式可以使用專用 MQI 通道，來回復在失敗時佇列管理程式所參與的任何不完整工作單元。

在此區段中，沒有延伸交易式功能的 WebSphere MQ MQI 用戶端稱為 *WebSphere MQ* 基本用戶端。因此，您可以將 WebSphere MQ 延伸交易式用戶端視為包含 WebSphere MQ 基本用戶端及延伸交易式功能。

相關參考

第 114 頁的『[延伸交易式用戶端的平台支援](#)』

除了 z/OS 之外，所有支援基本用戶端的平台都可以使用 IBM WebSphere MQ 延伸交易式用戶端。

延伸交易式用戶端的平台支援

除了 z/OS 之外，所有支援基本用戶端的平台都可以使用 IBM WebSphere MQ 延伸交易式用戶端。

使用延伸交易式用戶端的用戶端應用程式只能連接至下列 IBM WebSphere MQ Version 7.5 產品的佇列管理程式：

- IBM WebSphere MQ 適用於 AIX
- IBM WebSphere MQ 適用於 HP-UX
- IBM WebSphere MQ 適用於 HP Integrity NonStop Server
- IBM WebSphere MQ 適用於 Linux
- IBM WebSphere MQ 適用於 Solaris
- IBM WebSphere MQ 適用於 Windows

雖然沒有在 z/OS 上執行的延伸交易式用戶端，但使用延伸交易式用戶端的用戶端應用程式可以連接至在 z/OS 上執行的佇列管理程式。

對於每一個平台，延伸交易式用戶端的軟硬體需求與 IBM WebSphere MQ 基本用戶端的那些需求相同。如果 IBM WebSphere MQ 基本用戶端及您使用的交易管理程式支援，則延伸交易式用戶端支援程式設計語言。

下列網頁列出每一個平台的外部交易管理程式。

延伸交易式用戶端平台	網頁
AIX	WebSphere MQ 7.5 版最低需求 - AIX
HP-UX	WebSphere MQ 7.5 版最低需求 - HP-UX
HP Integrity NonStop Server	在 HP Integrity NonStop Server 上規劃 IBM WebSphere MQ 用戶端環境
Linux	WebSphere MQ 7.5 版最低需求 - Linux
Solaris	WebSphere MQ 7.5 版最低需求 - Solaris
Windows	WebSphere MQ 7.5 版最低需求 - Windows

用戶端如何連接至伺服器

用戶端使用 MQCONN 或 MQCONNX 連接至伺服器，並透過通道進行通訊。

在 IBM WebSphere MQ 用戶端環境中執行的應用程式必須維護用戶端與伺服器機器之間的作用中連線。

連線是由發出 MQCONN 或 MQCONNX 呼叫的應用程式所建立。用戶端及伺服器會透過 MQI 通道進行通訊，或在使用共用交談時，每一個都會共用 MQI 通道實例。當呼叫成功時，MQI 通道實例或交談會維持連接狀態，直到應用程式發出 MQDISC 呼叫為止。這是應用程式需要連接的每個佇列管理程式的情況。

相關概念

第 114 頁的『[相同機器上的用戶端及佇列管理程式](#)』

當您的機器也已安裝佇列管理程式時，您也可以[在 WebSphere MQ MQI 用戶端環境中執行應用程式](#)。

第 115 頁的『[不同平台上的用戶端](#)』

以下是 WebSphere MQ MQI 用戶端及伺服器系統的另一個範例。在此範例中，伺服器機器在不同平台上與三個 WebSphere MQ MQI 用戶端進行通訊。

第 115 頁的『[使用不同版本的用戶端及伺服器軟體](#)』

如果您使用舊版 IBM WebSphere MQ 產品，請確定伺服器支援從用戶端的 CCSID 進行程式碼轉換。

相同機器上的用戶端及佇列管理程式

當您的機器也已安裝佇列管理程式時，您也可以[在 WebSphere MQ MQI 用戶端環境中執行應用程式](#)。

在此狀況下，您可以選擇鏈結至佇列管理程式庫或用戶端程式庫，但請記住，如果您鏈結至用戶端程式庫，仍需要定義通道連線。在應用程式的開發階段期間，這可能很有用。您可以在自己的機器上測試您的程式，而不依賴其他程式，並確信當您將它移至獨立的 WebSphere MQ MQI 用戶端環境時，它仍會運作。

不同平台上的用戶端

以下是 WebSphere MQ MQI 用戶端及伺服器系統的另一個範例。在此範例中，伺服器機器在不同平台上與三個 WebSphere MQ MQI 用戶端進行通訊。

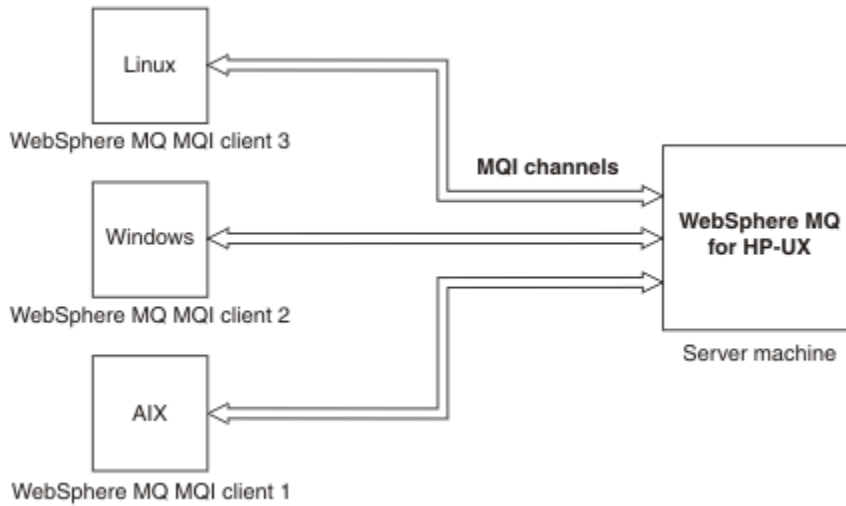


圖 31: WebSphere MQ 伺服器連接至不同平台上的用戶端

其他更複雜的環境是可能的。例如，WebSphere MQ 用戶端可以連接至多個佇列管理程式，或連接至任何數目的佇列管理程式 (作為佇列共用群組的一部分)。

使用不同版本的用戶端及伺服器軟體

如果您使用舊版 IBM WebSphere MQ 產品，請確定伺服器支援從用戶端的 CCSID 進行程式碼轉換。

IBM WebSphere MQ 用戶端可以連接至所有支援的佇列管理程式版本。如果您要連接至舊版佇列管理程式，則無法在用戶端上的 IBM WebSphere MQ 應用程式中使用較新版本產品的特性及結構。

IBM WebSphere MQ 佇列管理程式可以透過向下協議至最高相互支援的通訊協定層次，與不同版本的用戶端本身進行通訊。這表示較舊的用戶端可以與較新的佇列管理程式層次搭配使用。建議用戶端及伺服器都是目前支援的 IBM WebSphere MQ 版本，以協助問題診斷並啟用 IBM 的支援。

如需相關資訊，請參閱 [決定要使用的程式設計語言中支援的程式設計語言](#)。

交易管理及支援

交易管理及 WebSphere MQ 如何支援交易的簡介。

資源管理程式是一個電腦子系統，它擁有並管理可由應用程式存取及更新的資源。以下是資源管理程式的範例：

- WebSphere MQ 佇列管理程式，具有作為其佇列的資源
- DB2 資料庫，具有作為其表格的資源

當應用程式更新一或多個資源管理程式的資源時，可能有商業需求可確保某些更新全部以群組方式順利完成，或全部都未完成。這種需求的原因是如果其中部分更新順利完成，但其他更新未順利完成，則商業資料會保持不一致狀態。

以這種方式管理之資源的更新項目據說會在 工作單元或 交易內發生。應用程式可以將一組更新項目分組到工作單元。

在工作單元期間，應用程式會向資源管理程式發出要求，以更新其資源。當應用程式發出要求來確定所有更新項目時，工作單元即會結束。在確定更新項目之前，其他存取相同資源的應用程式都不會看到這些更新項目。或者，如果應用程式決定由於任何原因而無法完成工作單元，則它可以發出要求，以回復到該點為止所要求的所有更新項目。在此情況下，任何更新項目都不會對其他應用程式顯示。這些更新項目通常在邏輯上相關，且必須全部成功，才能保留資料完整性。如果一個更新成功，而另一個更新失敗，則資料完整性會遺失。

當工作單元順利完成時，即表示 確定。一旦確定，在該工作單元內所做的所有更新都會變成永久且無法回復。不過，如果工作單元失敗，則會改為 取消所有更新。此處理程序 (其中工作單元已確定或取消且具有完整性) 稱為 同步點協調。

確定或取消工作單元內所有更新項目的時間點稱為 同步點。工作單元內的更新在 同步點控制內進行。如果應用程式要求 超出同步點控制的更新，即使有工作單元在進行中，資源管理程式也會立即確定更新，且稍後無法取消更新。

管理工作單元的電腦子系統稱為 交易管理程式或 同步 點協調程式。

本端 工作單元是其中唯一更新的資源是 WebSphere MQ 佇列管理程式的資源。在這裡，佇列管理程式本身會使用一段式確定處理程序來提供同步點協調。

廣域 工作單元是其中屬於其他資源管理程式 (例如符合 XA 標準的資料庫) 的資源也會更新的工作單元。在這裡，必須使用兩階段確定程序，且工作單元可以由佇列管理程式本身或外部另一個符合 XA 標準的交易管理程式 (例如 IBM TXSeries 或 BEA Tuxedo) 來協調。

交易管理程式負責確保工作單元內資源的所有更新順利完成，或全部都不完成。對於交易管理程式，應用程式會發出要求來確定或取消工作單元。交易管理程式的範例有 CICS 和 WebSphere Application Server，雖然兩者也具有其他功能。

部分資源管理程式提供自己的交易管理功能。例如，WebSphere MQ 佇列管理程式可以管理工作單元，其中涉及對其專屬資源的更新，以及對 DB2 表格的更新。佇列管理程式不需要個別的交易管理程式來執行此功能，但如果是使用者需求，則可以使用此功能。如果使用個別交易管理程式，則稱為 外部交易管理程式。

如果要讓外部交易管理程式管理工作單元，交易管理程式與參與工作單元的每一個資源管理程式之間必須有標準介面。這個介面可讓交易管理程式和資源管理程式彼此通訊。其中一個介面是 XA 介面，它是一些交易管理程式和資源管理程式所支援的標準介面。XA 介面是由「開放式群組」在 分散式交易處理: XA 規格中發佈。

當多個資源管理程式參與工作單元時，即使系統失效，交易管理程式也必須使用 兩段式確定 通訊協定，以確保工作單元內的所有更新都順利完成，或沒有任何更新完成。當應用程式向交易管理程式發出確定工作單元的請求時，交易管理程式會執行下列動作：

階段 1 (準備確定)

交易管理程式會要求參與工作單元的每一個資源管理程式，以確保其資源之預期更新項目的所有相關資訊都處於可回復狀態。資源管理程式通常透過將資訊寫入日誌並確保將資訊寫入硬碟來執行此動作。當交易管理程式收到來自每一個資源管理程式的通知，指出其資源的預期更新項目相關資訊處於可回復狀態時，階段 1 即會完成。

階段 2 (確定)

當階段 1 完成時，交易管理程式會做出不可撤銷的決策來確定工作單元。它會要求參與工作單元的每一個資源管理程式確定其資源的更新。當資源管理程式收到這個要求時，它必須確定更新項目。在此階段，它無法選擇將它們退出。當交易管理程式收到來自每一個資源管理程式的通知，表示它已確定其資源的更新項目時，階段 2 即完成。

「XA 介面」使用兩段式確定通訊協定。

如需相關資訊，請參閱 [交易式支援](#)。

WebSphere MQ 也提供 Microsoft Transaction Server (COM+) 的支援。[使用 Microsoft Transaction Server \(COM+\) 提供如何設定 WebSphere MQ 以利用 COM+ 支援的相關資訊](#)。

延伸佇列管理程式機能

您可以使用使用者結束程式、API 結束程式或可安裝的服務來延伸佇列管理程式機能。

使用者結束程式

使用者結束程式提供一種機制，可讓您將您自己的程式碼插入佇列管理程式功能中。支援的使用者結束程式包括：

通道結束程式

這些結束程式會變更通道運作的方式。通道結束程式在 [傳訊通道的通道結束程式](#)中說明。

資料轉換結束程式

這些結束程式會建立可放入應用程式中的原始碼片段，以將資料從一種格式轉換成另一種格式。資料轉換結束程式在 [寫入資料轉換結束程式](#) 中有說明。

叢集工作量結束程式

這個結束程式所執行的功能是由結束程式的提供者所定義。呼叫定義資訊在 [MQ_CLUSTER_WORKLOAD_EXIT-呼叫說明](#) 中提供。

API 結束程式

API 結束程式可讓您撰寫程式碼來變更 WebSphere MQ API 呼叫 (例如 MQPUT 和 MQGET) 的行為，然後在那些呼叫之前或之後立即插入該程式碼。插入是自動的; 佇列管理程式會在登錄點驅動結束碼。如需 API 結束程式的相關資訊，請參閱 [使用及撰寫 API 結束程式](#)。

可安裝的服務

可安裝服務具有具有多個進入點的正規化介面 (API)。

可安裝服務的實作稱為 服務元件。您可以使用 WebSphere MQ 所提供的元件，也可以撰寫您自己的元件來執行您需要的功能。

目前提供下列可安裝服務：

授權服務

授權服務可讓您建置自己的安全機能。

實作服務的預設服務元件是物件權限管理程式 (OAM)。依預設，OAM 處於作用中，您不需要執行任何動作來配置它。您可以使用授權服務介面來建立其他元件，以取代或擴增 OAM。如需 OAM 的相關資訊，請參閱 [在 Windows 上設定安全，UNIX and Linux 系統](#)。

名稱服務

名稱服務可讓應用程式透過識別遠端佇列來共用佇列，如同它們是本端佇列一樣。

您可以撰寫自己的名稱服務元件。例如，如果您想要將名稱服務與 IBM WebSphere MQ 搭配使用，則可能想要執行此動作。若要使用名稱服務，您必須具有由使用者撰寫或由不同軟體供應商提供的元件。依預設，名稱服務為非作用中。

IBM WebSphere MQ Client for HP Integrity NonStop Server 技術概觀

HP Integrity NonStop Server 作業系統的技術概觀。

IBM WebSphere MQ HP Integrity NonStop Server SupportPac

HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端在 SupportPac MAT1 中發行。

HP Integrity NonStop Server 作業系統的技術概觀

HP Integrity NonStop Server 是一種作業系統，專門設計用來提供最高可能的可用性，即使有多個硬體或軟體故障，也不會有預定或未計劃的關閉時間。它可線性擴充，例如，如果您增加 20% 的硬體，則可獲得 20% 的可用效能。為了維護資料完整性，作業系統具有自己的交易管理程式及交易式檔案系統。

HP Integrity NonStop Server 作業系統通常由下列項目使用：

- 金融機構，例如 ATM 網路、線上銀行支援、信用授權、股票交換、交易及銀行對銀行交易。
- 例如，用於 Web 商店後端、庫存及程序控制的製造。
- 例如，電信，用於交換、緊急及其他網路服務。

IBM WebSphere MQ 用戶端 (適用於 HP Integrity NonStop Server 支援的環境和特性)

提供 HP Integrity NonStop Server 平台之 IBM WebSphere MQ 用戶端的詳細資料，並說明 HP Integrity NonStop Server 系統特有的受支援用戶端 API 和環境以及用戶端功能。

支援的用戶端 API 和環境

HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端支援下列執行環境:

表 10:		
	OSS	監護
C	✓	✓
JMS	✓	
COBOL	✓	✓
pTAL	✓	✓

功能摘要

用戶端功能的某些方面是主機作業系統特有的。下列摘要說明 IBM WebSphere MQ Client for HP Integrity NonStop Server 特有的用戶端功能層面:

- C (原生)、PTAL、COBOL (原生)
 - 網路通訊協定: TCP (IPv4 和 IPv6)
 - 傳輸類型: 僅限用戶端
 - 傳輸安全: SSL/TLS
 - 交易式支援: 由「交易管理機能 (TMF)」協調的兩階段確定 (需要連接至位於 IBM WebSphere MQ Version 7.1 或更新版本的佇列管理程式)
 - 定址模式: 32 位元
- Java 訊息服務 (Java Message Service, JMS)
 - 網路通訊協定: TCP (IPv4 和 IPv6)
 - 傳輸類型: 僅限用戶端 (不支援連結、直接及直接 HTTP)
 - 傳輸安全: SSL/TLS
 - 交易式支援: 單階段確定
 - 執行: 獨立式 (不支援 Application Support Facility (ASF) 和 Java 連接器架構 (JCA))
 - 結束程式: 僅限 Java 語言 (不支援以其他語言撰寫的原生結束程式)
 - IBM WebSphere MQ 標頭和 PCF: 不支援下列類別: com.ibm.mq.headers.* 及 com.ibm.mq.pcf.*

案例

本節提供實務範例的相關資訊，說明如何使用及結合新的 WebSphere MQ 7.5 功能。實務範例包括資訊中心內容的有用鏈結，可協助您進一步瞭解您感興趣的區域。

下列子主題中說明可用的實務範例:

IBM WebSphere MQ Version 7.5 入門

此實務範例說明如何在 Windows 平台上開始使用 IBM WebSphere MQ Version 7.5。如果您從未使用 IBM WebSphere MQ 且想要快速開始使用，請使用此實務範例。

此實務範例包含下列區段。您可以使用圖形或指令行介面來完成這些步驟，如本實務所示。

基本概念與重要詞彙

說明您在開始使用 IBM WebSphere MQ Version 7.5 實務範例之前必須瞭解的基本概念和重要詞彙。

基本概念

IBM WebSphere MQ 讓應用程式能夠讀取訊息並將其寫入佇列。讀取訊息的應用程式與寫入訊息的應用程式是互相獨立的。不需要同時執行兩個應用程式。如果沒有應用程式可讀取訊息，則會將訊息排入 IBM WebSphere MQ 佇列，直到應用程式進行讀取。

重要詞彙

以下是訊息佇列作業的相關重要術語清單。
關於訊息佇列的重要術語。

詞彙	說明
佇列管理程式	佇列管理程式負責維護所擁有的佇列，並將其接收的所有訊息儲存至適當的佇列中。
訊息	訊息是一個位元組字串，對於使用它的應用程式而言是有意義的。訊息用來將資訊從某個應用程式傳送到另一個應用程式。應用程式可在相同或不同的電腦上執行。
本端佇列	本端佇列是用來儲存訊息的資料結構。佇列可以是正常佇列或傳輸佇列。正常佇列會保留要被應用程式讀取的訊息，該應用程式可透過佇列管理程式直接讀取訊息。傳輸佇列會保留要傳送至另一個佇列管理程式的訊息。
遠端佇列	遠端佇列是用來將訊息定址至另一個佇列管理程式。
通道	通道用於在佇列管理程式之間傳送及接收訊息。
接聽器	接聽器是接受來自其他佇列管理程式或用戶端應用程式的網路要求，並啟動相關聯通道的處理程序。

建立稱為 QM1 的佇列管理程式

使用指令行介面或「WebSphere MQ 探險家」，建立一個稱為 QM1 的佇列管理程式，以與「IBM WebSphere MQ Version 7.5 入門」實務範例搭配使用。佇列管理程式是 WebSphere MQ 傳訊網路中的主要元件。

開始之前

您必須已安裝 IBM WebSphere MQ Version 7.5。如果沒有，請參閱 [安裝及解除安裝](#)，以取得如何執行此動作的相關資訊。

關於這項作業

在此範例中，所有名稱都以大寫形式輸入，而且由於 IBM WebSphere MQ 名稱區分大小寫，因此您也必須以大寫形式輸入所有名稱。

使用指令行介面來建立佇列管理程式

若要使用指令行介面來建立及啟動佇列管理程式，請完成下列步驟：

程序

1. 鍵入下列指令，以建立名為 QM1 的佇列管理程式：

```
crtmqm QM1
```

當系統建立佇列管理程式時，會顯示下列輸出：

```
C:\>crtmqm QM1
WebSphere MQ queue manager created.
Creating or replacing default objects for QM1.
Default objects statistics : 61 created. 0 replaced. 0 failed.
```

```
Completing setup.  
Setup completed.
```

即會建立並停止佇列管理程式。您必須先啟動佇列管理程式，然後才能管理它，以及從其佇列讀取及寫入訊息。

2. 輸入下列指令，以啟動佇列管理程式：

```
strmqm QM1
```

當佇列管理程式順利啟動時，會顯示下列輸出：

```
C:\>strmqm QM1  
WebSphere MQ queue manager 'QM1' starting.  
5 log records accessed on queue manager 'QM1' during the log replay phase.  
Log replay for queue manager 'QM1' complete.  
Transaction manager state recovered for queue manager 'QM1'.  
WebSphere MQ queue manager 'QM1' started.
```

佇列管理程式已啟動。

下一步

若要建立佇列，請參閱 [第 120 頁的『建立稱為 LQ1 的佇列』](#)。

使用「WebSphere MQ 探險家」建立佇列管理程式

如果要使用「WebSphere MQ 探險家」來建立及啟動佇列管理程式，請完成下列步驟：

程序

1. 啟動「WebSphere MQ 探險家」。
2. 在「Navigator」視圖中，用滑鼠右鍵按一下 **佇列管理程式** 資料夾，然後按一下 **新建> 佇列管理程式**。即會啟動「**建立佇列管理程式**」精靈。
3. 在 **佇列管理程式名稱** 欄位中，輸入 QM1。
4. 選取 **Make this the default queue manager** 勾選框。
5. 在 **無法傳送郵件的佇列** 欄位中，輸入 SYSTEM.DEAD.LETTER.QUEUE。
這是建立佇列管理程式時自動建立的無法傳送郵件的佇列名稱。
6. 將其他欄位留空並按一下 **完成**，或者如果該按鈕已停用，請按 **下一步**。
如果埠號與現有佇列管理程式 (例如作為預設配置的一部分建立的佇列管理程式) 相衝突，則會停用 **完成** 按鈕。您必須透過精靈繼續變更預設埠號。
7. 如果您按 **下一步**，請繼續接受預設值，並在每一個頁面上按 **下一步**，直到您到達精靈的最後一頁 (當 **完成** 按鈕變成可用時)。變更指定的埠號，例如變更為 1415，然後按一下 **完成**。

在建立並啟動佇列管理程式時，WebSphere MQ 會顯示「**建立佇列管理程式**」對話框視窗。

下一步

若要建立佇列，請參閱 [第 120 頁的『建立稱為 LQ1 的佇列』](#)。

建立稱為 LQ1 的佇列

使用指令行介面或「WebSphere MQ 探險家」，建立佇列以與 WebSphere IBM WebSphere MQ Version 7.5 入門實務範例搭配使用。佇列是用來儲存訊息的資料結構，並且是 IBM WebSphere MQ 佇列管理程式物件。

關於這項作業

有三種方式可以建立 IBM WebSphere MQ 物件：

- 指令行。

- IBM WebSphere MQ Explorer.
- 使用可程式化介面。

在這項作業中，您可以使用指令行或 IBM WebSphere MQ Explorer 來建立 IBM WebSphere MQ 物件。

使用指令行介面來建立佇列

指令行介面具有稱為「IBM WebSphere MQ Script 指令 (MQSC)」的 Scripting 語言。Scripting 工具 **runmqsc** 用來對佇列管理程式執行 Script。若要使用指令行介面來建立及啟動佇列，請完成下列步驟：

程序

1. 輸入下列指令來啟動 Scripting 工具：

```
runmqsc QM1
```

當 Scripting 工具啟動時，會顯示下列輸出：

```
C:\>runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
```

工具已準備好接受 MQSC 指令。

2. 鍵入下列 MQSC 指令，建立稱為 LQ1 的本端佇列：

```
define qlocal(LQ1)
```

建立佇列時，會顯示下列輸出：

```
define qlocal(LQ1)
  2 : define qlocal(LQ1)
AMQ8006: WebSphere MQ queue created.
```

3. 透過鍵入下列 MQSC 指令，停止 Scripting 工具：

```
end
```

當 Scripting 工具結束時，會顯示下列輸出：

```
One MQSC command read.
No commands have a syntax error.
All valid MQSC commands were processed.
C:\>
```

下一步

您已準備好將訊息放置在佇列上。若要將訊息放入佇列中，請參閱 [第 122 頁的『將訊息放入佇列 LQ1』](#)。

使用「WebSphere MQ 探險家」建立佇列

若要使用「WebSphere MQ 探險家」來建立及啟動佇列，請完成下列步驟：

程序

1. 在「**導覽器**」視圖中，展開**佇列管理程式**資料夾。
2. 展開佇列管理程式 **QM1**。
3. 用滑鼠右鍵按一下 **佇列** 資料夾，然後按一下 **新建> 本端佇列 ...** 即會啟動 **新建本端佇列** 精靈。
4. 在 **名稱** 欄位中，輸入 LQ1。
5. 按一下**完成**。

新的佇列 LQ1 會顯示在「**內容**」視圖中。如果佇列未顯示在「**內容**」視圖中，請按一下「**內容**」視圖頂端的 **重新整理** 按鈕。

下一步

您已準備好將訊息放置在佇列上。若要將訊息放入佇列中，請參閱 [第 122 頁的『將訊息放入佇列 LQ1』](#)。

將訊息放入佇列 LQ1

使用指令行介面或 IBM WebSphere MQ Explorer，將訊息放置在佇列 LQ1 中，以與 IBM WebSphere MQ Version 7.5 實務範例一起使用。

關於這項作業

IBM WebSphere MQ 隨附稱為 amqsput 的範例應用程式。此應用程式會將訊息放入預先定義的佇列。

使用指令行介面將訊息放入佇列

若要使用指令行介面將訊息放置在佇列上，請完成下列步驟：

程序

1. 鍵入下列指令，使用 amqsput 範例應用程式將訊息放入佇列 LQ1：

```
amqsput LQ1 QM1
```

當範例應用程式啟動時，會顯示下列輸出：

```
C:\>amqsput LQ1 QM1
Sample AMQSPUT0 start
target queue is LQ1
```

2. 鍵入 Hello World，然後按 Enter 鍵。您將包含文字 "Hello World" 的訊息放置在名為 QM1 的佇列管理程式所管理的佇列 LQ1 上。
3. 若要結束 amqsput，請按 Enter 鍵。會顯示下列輸出：

```
C:\>amqsput LQ1 QM1
Sample AMQSPUT0 start
target queue is LQ1
Hello World

Sample AMQSPUT0 end
```

下一步

若要從佇列取得訊息，請參閱 [第 123 頁的『從佇列 LQ1 取得訊息』](#)。

使用 IBM WebSphere MQ Explorer 將訊息放入佇列

若要使用 IBM WebSphere MQ Explorer 將訊息放置在佇列上，請完成下列步驟：

程序

1. 在「**導覽器**」視圖中，展開**佇列管理程式**資料夾。
2. 展開您建立的佇列管理程式 QM1。
3. 按一下**佇列**資料夾。佇列管理程式的佇列會列在「**內容**」視圖中。
4. 在「**內容**」視圖中，用滑鼠右鍵按一下本端佇列 LQ1，然後按一下 **放置測試訊息 ..**。這時會開啟「**放置測試訊息**」對話框。
5. 在 **訊息資料** 欄位中，鍵入部分文字 (例如 Hello World)，然後按一下 **放置訊息**。這時會清除**訊息資料**欄位，並且將訊息放置到佇列中。
6. 按一下**關閉**。

在「**內容**」視圖中，請注意 LQ1 **現行佇列深度** 值現在是 1。如果看不到 **現行佇列深度** 直欄，您可能需要捲動至「**內容視圖**」右側。

下一步

若要從佇列取得訊息，請參閱 [第 123 頁的『從佇列 LQ1 取得訊息』](#)。

從佇列 LQ1 取得訊息

使用指令行介面或 IBM WebSphere MQ Explorer，從佇列 LQ1 取得訊息，以與 IBM WebSphere MQ Version 7.5 入門實務範例搭配使用。

關於這項作業

IBM WebSphere MQ 隨附稱為 amqsget 的範例應用程式。此應用程式會從佇列讀取訊息。

使用指令行介面從佇列取得訊息

若要使用指令行介面從佇列取得訊息，請完成下列步驟：

程序

透過鍵入下列指令，使用 amqsget 範例應用程式來讀取佇列 LQ1 上的訊息：

```
amqsget LQ1 QM1
```

當範例應用程式啟動時，會顯示下列輸出：

```
C:\>amqsget LQ1 QM1
Sample AMQSGETO start
message <Hello World>
no more messages
Sample AMQSGETO end
```

amqsget 應用程式在讀取訊息之後 30 秒結束。

下一步

若要瞭解如何撰寫佇列作業應用程式、連接至佇列管理程式、發佈/訂閱以及開啟和關閉物件，請參閱 [撰寫佇列作業應用程式](#)。

使用 IBM WebSphere MQ Explorer 從佇列取得訊息

若要使用 IBM WebSphere MQ Explorer 從佇列取得訊息，請完成下列步驟：

程序

1. 在「**Navigator**」視圖中，展開 **佇列管理程式** 資料夾，然後展開 QM1。
2. 按一下 **佇列** 資料夾。
3. 在「**內容**」視圖中，用滑鼠右鍵按一下 QM1，然後按一下 **瀏覽訊息 ...**。即會開啟 **訊息瀏覽器**，以顯示目前位於 QM1 上的訊息清單。
4. 按兩下最後一則訊息以開啟內容對話框。

在內容對話框的**資料**頁面中，**訊息資料**欄位會以可閱讀的格式來顯示訊息內容。

下一步

若要瞭解如何撰寫佇列作業應用程式、連接至佇列管理程式、發佈/訂閱以及開啟和關閉物件，請參閱 [撰寫佇列作業應用程式](#)。

下一步

完成「開始使用 IBM WebSphere MQ Version 7.5」實務範例時接下來要執行的動作。

IBM WebSphere MQ 提供角色型訓練路徑，透過定義獲取特定 WebSphere 產品供應項目技能的路徑來協助您。IBM WebSphere MQ 有兩種訓練路徑：

- 應用程式開發人員

這些使用者負責建立使用佇列管理程式的應用程式。在此實務範例中，他們撰寫應用程式 `amqsput` 及 `amqsget`。

- 系統管理者

這些使用者負責建立佇列管理程式及其物件，他們通常會執行您在此實務範例中所涵蓋的類似作業。

如需 IBM WebSphere MQ 訓練路徑的相關資訊，請參閱：<https://www.ibm.com/software/websphere/education/paths/>。

若要檢視 IBM WebSphere MQ 課程的完整清單，請參閱：<https://www.ibm.com/software/websphere/education/curriculum/appint/wmq/>。

提供憑證計劃，以示範您在 IBM WebSphere MQ 中達到技能水準。如需相關資訊，請參閱：<https://www.ibm.com/certify/index>。

您可以與其他使用者分工合作，例如：

- 電子郵件型 IBM WebSphere MQ 專業人員社群，請參閱 <https://listserv.meduniwien.ac.at/archives/mqser-l.html>。
- 聚焦 IBM WebSphere MQ 產品系列的討論論壇，請參閱 <https://www.mqseries.net/>。
- 各種 IBM 傳訊產品的開發人員所提供的 IBM Developer 部落格，請參閱 <https://www.ibm.com/developerworks/blogs/page/messaging/>。
- IBM WebSphere MQ 的官方 IBM 代管討論區，請參閱 <https://www.ibm.com/developerworks/forums/forum.jspa?forumid=280>。
- IBM WebSphere MQ stackoverflow.com，請參閱 <https://stackoverflow.com/questions/tagged/websphere-mq>。

本產品說明文件中還有其他主題可供您檢視。您可能想要查看下列區段：

- [管理 IBM WebSphere MQ](#)

IBM WebSphere MQ 提供您可以使用的控制指令。在此實務範例中，您使用下列兩個指令：`crtmqm` 及 `strmqm`。本節也提供關於訊息佇列作業的良好概觀。

- [MQSC 參照](#)

在此實務範例中，您使用 `define qlocal('LQ1')` 指令來定義稱為 LQ1 的本端佇列；此指令是 MQSC 指令。IBM WebSphere MQ 系統管理者使用這些指令來管理其佇列管理程式。本節介紹指令，並顯示如何使用這些指令，然後再按字母順序詳細說明指令。

- [配置佇列管理程式叢集](#)

本節說明如何組織、使用及管理虛擬群組（稱為叢集）中的佇列管理程式。叢集作業可確保叢集內的每一個佇列管理程式都知道相同叢集中的所有其他佇列管理程式。叢集作業也可簡化複式佇列管理程式網路的管理。

「產品連線功能實務範例」產品說明文件提供資訊，引導您在各種實務範例中完成將 WebSphere Application Server 連接至 WebSphere MQ 所需的主要作業。每一個實務範例都包含在商業環境定義中實作解決方案的指示，可讓您在不需要使用其他資訊資源的情況下進行學習。https://www.ibm.com/docs/prodconn_1.0.0/com.ibm.prodconn.doc/infocenter_homepage/ic_home.htm

使用 Script 的基本檔案傳送

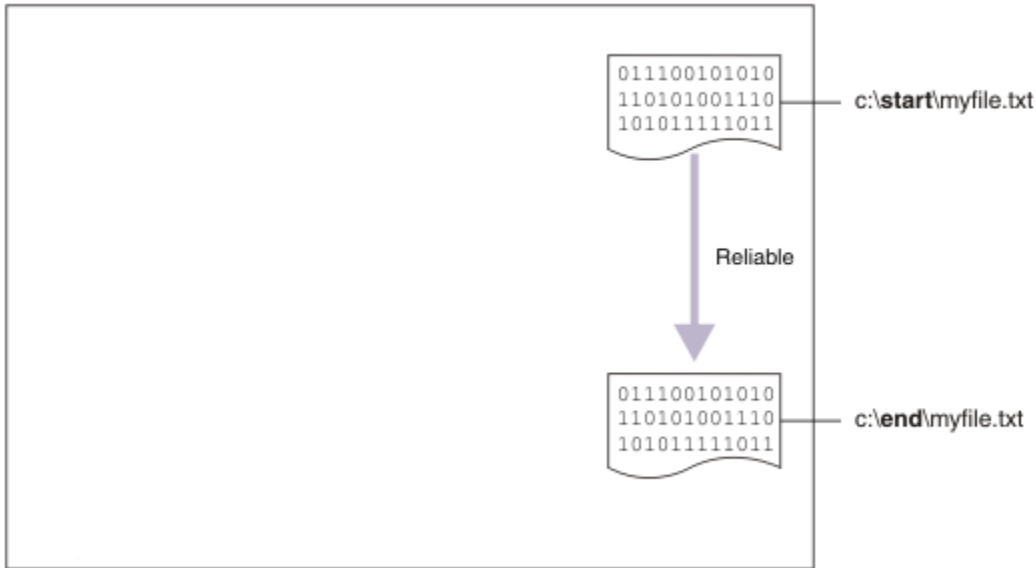
您可以使用 IBM WebSphere MQ Version 7.5，以多種不同的方式來傳送檔案。請檢閱本節中的主題，以瞭解此實務範例中涵蓋的內容、企業可能想要遵循實務範例的原因、涉及的使用者角色，以及實務範例所提出解決方案的概觀。

在企業中，具有控制、可靠且具有審核的傳送檔案可能是基本需求。IBM WebSphere MQ Version 7.5 提供「受管理檔案傳送」功能作為其整合傳訊平台的一部分。您可以使用「受管理檔案傳送」功能，透過基本檔案傳送或完全成熟的傳訊參與者，將檔案無縫整合至傳訊基礎架構。如需此功能的詳細資料，請參閱 [WebSphere MQ Managed File Transfer 簡介](#)。

此實務範例可讓您基本瞭解如何將檔案整合至最簡單的 IBM WebSphere MQ 傳訊拓撲。若要這樣做，您可以完成基本 IBM WebSphere MQ 實務範例，設計為將檔案從一個位置移至另一個位置。雖然此起始實務範例僅限於單一電腦，但它提供您配置環境的經驗，並形成後續實務範例的重要基礎。稍後的實務範例示範如何使用 IBM WebSphere MQ 來透過網路傳送檔案，然後開始顯示「受管理檔案傳送」元件如何解決實際的商業問題。

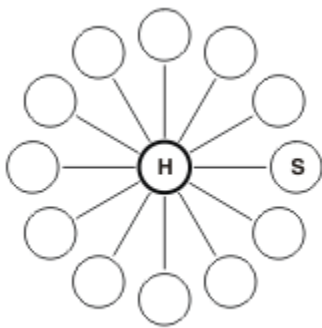
您應該對 IBM WebSphere MQ 有基本瞭解，特別是佇列管理程式的概念，以及 IBM WebSphere MQ 的基本配置和管理，直到使用 **runmqsc** 和「IBM WebSphere MQ 探險家」之類的指令。

在此實務範例中，您探索如何使用 IBM WebSphere MQ 在單一電腦上起始及追蹤檔案從一個位置到另一個位置的傳送，讓您在 IBM WebSphere MQ Version 7.5 中安裝、配置及使用可靠受管理檔案傳送功能的經驗。



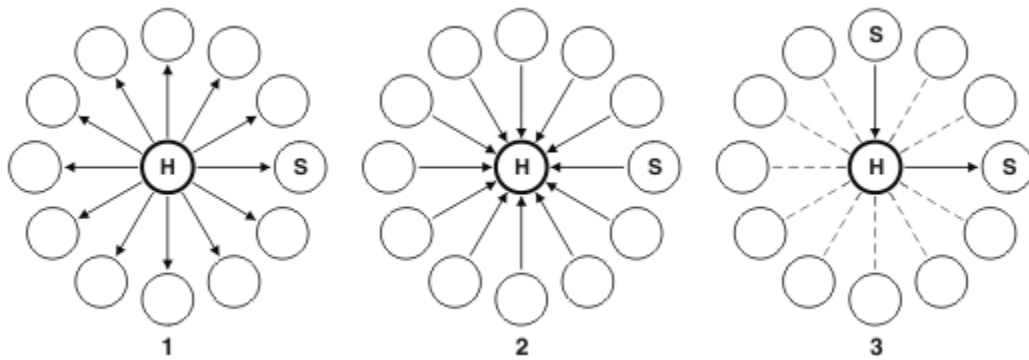
範例檔案傳送拓撲

此實務範例及後續實務範例以集中分支拓撲為基礎。此圖顯示包含中心 (H) 及多個分支 (S) 的概念性中心及分支拓撲。



這類拓撲會在許多商業實務範例中尋找應用程式，例如，您可能想要：

1. 將檔案從集中化 HQ (中心) 傳送至許多外圍節點 (輪輻)。
2. 將檔案從許多節點 (輪輻) 整理到單一位置 (中心)。
3. 將檔案從一個分支傳送至另一個分支。



您的拓撲可能不同於此範例，但此實務範例中的原則及指令可以隨時延伸，以滿足任何 IBM WebSphere MQ 網路的需求。

規劃解決方案

規劃使用 Script 與電腦之間來回傳送檔案。包括瞭解安全模型、必備項目及授權、安裝及配置 IBM WebSphere MQ，以及準備使用者和群組。

若要將檔案從點 A 傳送至點 B，請在單一電腦上建立中心分支拓撲。此拓撲包含中心、一個 IBM WebSphere MQ 佇列管理程式，以及每一個檔案傳送代理程式兩個輪輻。

檔案傳送代理程式是在電腦上執行並與其他代理程式來回傳送檔案的 Java 處理程序。在此實務範例中，定義使用這些檔案傳送代理程式的檔案傳送，以透過兩種機制將範例檔案從一個位置移至另一個位置：

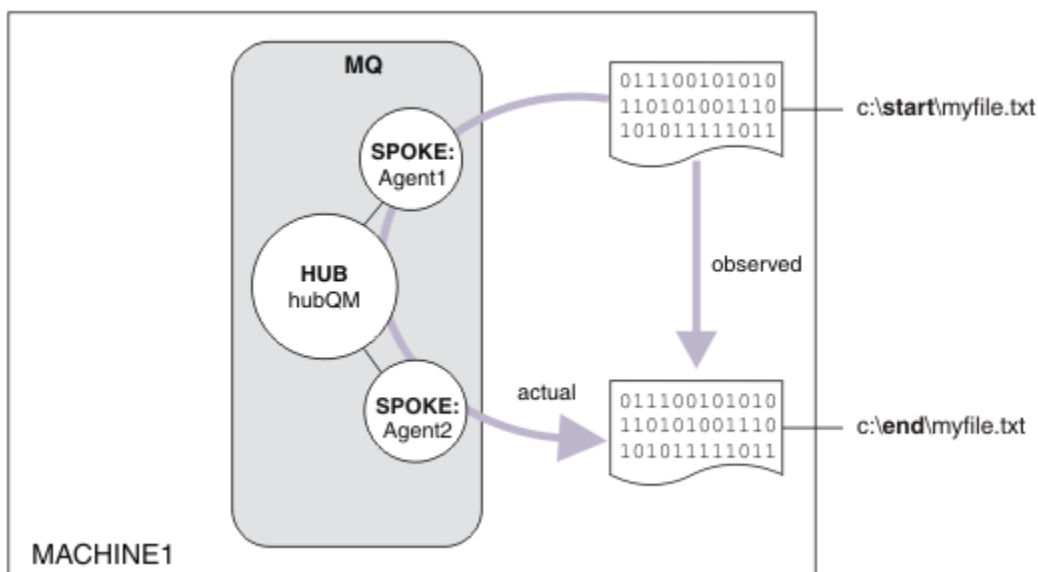
1. 使用圖形「IBM WebSphere MQ 檔案總管」。
2. 透過指令行。

此實務範例假設您具有 Windows 系統。對於 UNIX 系統，請替換適當的路徑和指令，並確保您具有所有相關目錄的讀取權和寫入權。此實務範例還假設您基本瞭解佇列管理程式執行的作業。

若要與電腦來回傳送檔案，您必須在該電腦上執行檔案傳送代理程式。每個代理程式都會連接至 IBM WebSphere MQ 佇列管理程式，並使用 IBM WebSphere MQ 與其他代理程式進行通訊。如需相關資訊，請參閱 [WebSphere MQ Managed File Transfer 拓撲概觀](#)。

IBM WebSphere MQ Version 7.5 有兩個範例 Script `createHub` 和 `addSpoke`，可協助您快速且輕鬆地建置檔案傳送拓撲。使用這些 Script，從中心向外建構此拓撲。這些 Script 是範例，您可以修改它們以符合您自己的需求。如果您要修改 Script，請先將它們複製到您自己的使用者目錄中的位置。

範例 Script `createHub` 和 `addSpoke` 用來建構拓撲。



啟動傳送時，您會觀察從一個位置複製到另一個位置的檔案。此簡式實務範例可以透過管理來源及目的地檔案的單一檔案傳送代理程式來達成。不過，會使用兩個檔案傳送代理程式，為您提供在移至多電腦拓撲之前此設定如何運作的較佳範例。

瞭解安全模型

createHub 和 addSpoke Script 會配置具有下列安全性質的檔案傳送拓撲：

- IBM WebSphere MQ 的存取權在三個角色之間分割：
 1. 配置 IBM WebSphere MQ 並執行 createHub 和 addSpoke Script 的 IBM WebSphere MQ 管理者。
 2. 啟動、停止代理程式並與代理程式互動的 FTAGENTS。
 3. 起始檔案傳送的 FTUSERS。

如需配置存取權的相關資訊，請參閱 [WebSphere MQ Managed File Transfer](#) 特定資源的權限。

- 來自代理程式的所有送入連線都會對映至中心的單一使用者 FTAGENT。
- 低保護性 IP 型鑑別是用來鑑別個別代理程式。

為了明確起見，Script 中的安全模型未完全強化。在正式作業中使用這些 Script 之前，您必須先瞭解自己的拓撲需求及安全威脅。因此，您必須考量是否要處理下列潛在漏洞及建議：

- 任何使用者都可以假冒任何其他使用者。請考量檔案傳送資源的物件存取模型中的更細精度。
- 任何代理程式都可以假冒任何其他使用者。請考量更強的鑑別，例如，TLS/SSL 及檔案傳送資源的物件存取模型中的更細精度。
- 未考量檔案系統與 IBM WebSphere MQ 安全之間的介面。請考量實作檔案沙盤推演，並瞭解代理程式配置檔的權限影響。如需沙盤推演的相關資訊，請參閱 [沙盤推演](#)。
- 未說明代理程式與作業系統之間的介面。請考量實作檔案沙盤推演。如需詳細資料，請參閱 [沙盤推演](#)。

如需安全、IBM WebSphere MQ 及檔案傳送的相關資訊，請參閱 [下一步](#)。

必備項目及授權

您需要下列項目：

- 滿足 IBM WebSphere MQ 的硬體及作業系統必要條件的測試電腦，如需詳細資料，請參閱 <https://www.ibm.com/support/docview.wss?uid=swg27006467>，且不存在 IBM WebSphere MQ 或 IBM WebSphere MQ 資料的現有安裝。
- IBM WebSphere MQ Version 7.5。您可以從 <https://www.ibm.com/developerworks/downloads/ws/wmq/> 下載試用版。

準備電腦

確保測試電腦滿足 IBM WebSphere MQ Version 7.5 安裝的需求，請參閱 [檢查需求](#)。

確保電腦已適當地準備好進行安裝，請參閱 [使用指令行驗證本端安裝](#)。

安裝 IBM WebSphere MQ Version 7.5

使用下列元件來安裝 IBM WebSphere MQ Version 7.5 伺服器：伺服器、IBM WebSphere MQ Explorer、IBM WebSphere MQ 進階受管理檔案傳送代理程式及 IBM WebSphere MQ 受管理檔案傳送指令行工具。如需詳細資料，請參閱 [選擇要安裝的項目](#)。

決定要管理 IBM WebSphere MQ 的方式。您可以透過下列方式管理 IBM WebSphere MQ：

- 使用 **setmqenv** 指令設定適當的環境。如需相關資訊，請參閱 [setmqenv](#)。
- 呼叫完整的 IBM WebSphere MQ 管理指令。

此實務範例假設您使用全新電腦，且未安裝先前的 IBM WebSphere MQ 或 IBM WebSphere MQ 檔案傳送版本安裝。否則，您必須決定是否支援共存性，或調整環境的安裝機制及配置。如需相關詳細資料，請參閱 [多重安裝](#)。

建立使用者和群組

此安全模型假設您已建立下列使用者和群組：

使用者

- mqmAdmin

IBM WebSphere MQ 管理者 (即 mqm 群組的成員)，或在 Windows 上為 Administrators 群組的成員。您必須建立此使用者，並使其成為 mqm 群組的成員。

- ftuser

您必須建立此使用者，並使其成為 FTUSERS 群組的成員。若要避免可能在佇列管理程式上執行管理層次安全動作，請不要將此使用者新增至 mqm 群組。

- ftagent

您必須建立此使用者，並使其成為 FTAGENTS 群組的成員。若要避免可能在佇列管理程式上執行管理層次安全動作，請不要將此使用者新增至 mqm 群組。

群組

- mqm

在 IBM WebSphere MQ 安裝期間自動建立。此群組的成員可以管理 IBM WebSphere MQ 及其資源。

- FTUSERS

您必須建立此群組。此群組的成員可以起始檔案傳送。

- FTAGENTS

您必須建立此群組。此群組的成員可以啟動及停止檔案傳送代理程式，這些代理程式是處理網路中檔案傳送的端點。

配置 IBM WebSphere MQ 以進行檔案傳送

使用範例 Script createHub 和 addSpoke 來配置 IBM WebSphere MQ 進行檔案傳送，以使用 Script 實務來建置基本檔案傳送的拓撲。

程序

1. 識別遠端代理程式 (在稍後的實務範例中定義) 可以連接至的可用埠，例如 1414。
2. 識別適合作為檔案傳送網路中心的佇列管理程式名稱，例如 hubQM。
3. As the user mqmAdmin, from the IBM WebSphere MQ bin directory, <MQ_INSTALL_ROOT>/bin:

- a) 建立中心。

執行下列指令：

```
<MQ_INSTALL_ROOT>\mqft\samples\scripts\createHub hubQmgr=hubQM hubPort=1414
```

- b) 新增呼叫代理程式 AGENT1 的第一個代理程式分支 (在 BINDINGS 中，因為它是在佇列管理程式本端)。

```
<MQ_INSTALL_ROOT>\mqft\samples\scripts\addSpoke agentName=AGENT1  
hubQmgr=hubQM connectionMode=BINDINGS
```

- c) 新增第二個代理程式分支 (在 BINDINGS 中，因為它位於佇列管理程式本端)，並呼叫代理程式 AGENT2。

```
<MQ_INSTALL_ROOT>\mqft\samples\scripts\addSpoke agentName=AGENT2  
hubQmgr=hubQM connectionMode=BINDINGS
```

4. 以使用者 ftagent 身分:

- a) 輸入下列指令，以啟動 AGENT1：


```
fteStartAgent -p hubQM AGENT1
```

如果您已配置多個中心，指令的 `-p hubQM` 部分可確保您挑選正確的中心。單一集線器不需要它。

b) 輸入下列指令，以啟動 AGENT2：

```
fteStartAgent -p hubQM AGENT2
```

c) 列出代理程式以確認它們正確執行。

```
fteListAgents -p hubQM
```

您將看到下列輸出：

Agent Name:	Queue Manager Name:	Status:
AGENT1	hubQM	READY
AGENT2	hubQM	READY

實作解決方案

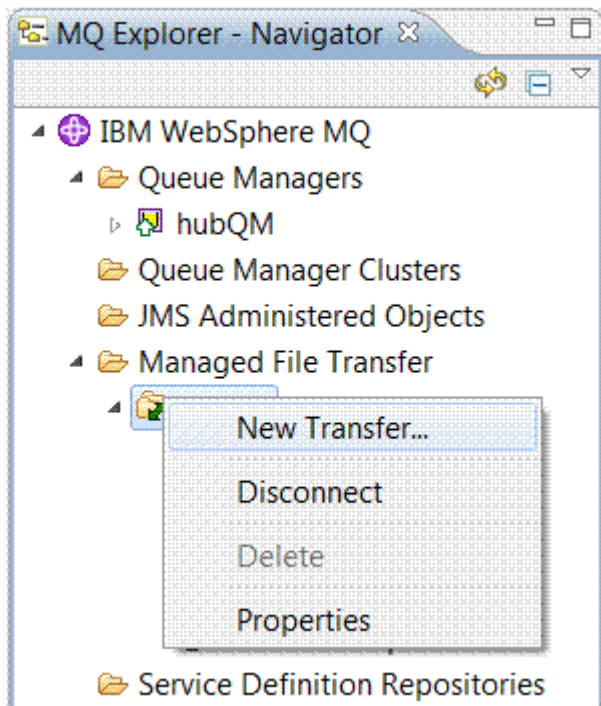
在此實務範例中實作解決方案涉及使用 IBM WebSphere MQ Explorer 來定義及起始檔案傳送。您可以監視傳送進度，並透過檢查檔案系統來確認其成功。

關於這項作業

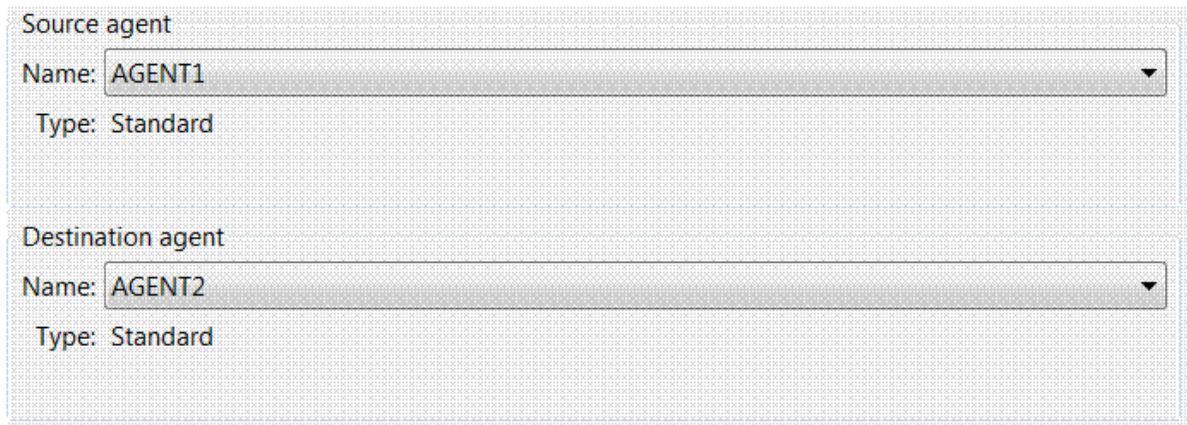
為了避免需要為非管理使用者 `ftuser` 配置 IBM WebSphere MQ Explorer，請以使用者 `mqmAdmin` 身分執行實務範例的這個部分。如果您要啟用此功能，請參閱 [安全](#)。

程序

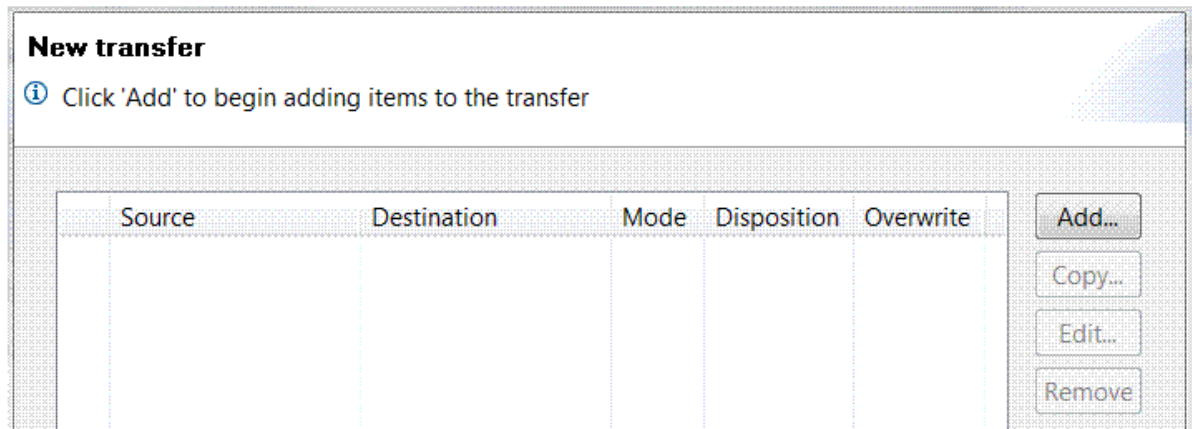
1. 識別來源檔和目標目錄：
 - a) 建立要傳送的範例檔，例如 `C:\start\myfile.txt`
 - b) 識別要將此檔案傳送至其中的現有目錄，例如，`C:\end\`
2. 以管理者身分，在此範例中，`mqmAdmin` 啟動 IBM WebSphere MQ Explorer。從「開始」功能表 (或對等項目) 啟動程式，或執行指令 **MQExplorer**。如需詳細資料，請參閱 [啟動 WebSphere MQ 探險家](#)。
3. 在 **受管理檔案傳送** 區段中，用滑鼠右鍵按一下名為 `hubQM` 的配置，然後選取 **連接**。
4. 在 IBM WebSphere MQ 瀏覽器中的 **受管理檔案傳送** 區段下，用滑鼠右鍵按一下 `hubQM`，然後選取 **新建傳送** 以啟動「新建傳送」精靈



5. 使用功能表來選取 AGENT1 作為來源代理程式，並選取 AGENT2 作為目的地代理程式。



6. 按下一步。



7. 按一下**新增...** 以開始選取要從 AGENT1 傳送至 AGENT2 的檔案。

8. 在「來源」頁框中，按一下 **瀏覽** 並瀏覽至先前識別的檔案，例如 C:\start\myFile.txt。確保已勾選 **如果傳送成功則移除來源檔案** 勾選框。

Source

Agent: AGENT1
Type: Standard agent
Type: File
File name: C:\start\myFile.txt
Browse...
 Remove source file if the transfer is successful

9. 在 **目的地** 頁框中，選取 **類型** 為 Directory，然後輸入您先前識別的目的地目錄，例如 C:\end\。

Destination

Agent: AGENT2
Type: Standard agent
Type: Directory
Directory: c:\end
Browse...
 Overwrite files if present

10. 按一下 **確定**，然後按一下 **完成**，即會開始傳送。
11. 您可以在「IBM WebSphere MQ 檔案總管」底端的「現行傳送進度」標籤中監視傳送進度：

IBM WebSphere MQ Explorer (Installation1)

MQ Explorer - Navigator

- IBM WebSphere MQ
 - Queue Managers
 - hubQM
 - Queue Manager Clusters
 - JMS Administered Objects
 - Managed File Transfer
 - hubQM
 - Agents
 - Monitors
 - Pending Transfers
 - Transfer Log
 - Transfer Templates
 - Service Definition Repositories

MQ Explorer - Content

Transfer Log

Source	Destination	Completion State	Owner
AGENT1	AGENT2	Successful	Ben Bakowski @ kitten.hursley.uk.ibm.com.

Filter the displayed log entries:

Managed File Transfer - Current Transfer Progress

Source	Destination	Current File	File Number	Progress	Rate
AGENT1	AGENT2	myFile.txt - (2KiB / 2KiB)	1 / 1	100%	# Successful #

12. 您也可以手動檢查檔案系統，以確認新檔案存在，例如 C:\end\myfile.txt。

使用指令行傳送檔案

在本節中，顯示如何使用指令行來起始傳送，以示範檔案傳送的彈性。雖然超出此實務範例的範圍，但您可以根據這些原則及 Ant Scripting 技術來定義及實作更強大的檔案傳送實務範例。

程序

1. 傳回您傳送至其原始位置的檔案。
2. 以使用者 `ftuser` 身分，輸入 **fteCreateTransfer** 指令以起始將檔案從 `C:\start\myfile.txt` 傳送至 `C:\end\myfile.txt`：

```
fteCreateTransfer -sa AGENT1 -sm hubQM -sd delete -da AGENT2 -dm hubQM -w -dd C:\end\  
C:\start\myfile.txt
```

- `-sa AGENT1` 將來源代理程式 (即從中傳送檔案的代理程式) 定義為 `AGENT1`。
- `-sm hubQM` 定義來源代理程式 `AGENT1` 所連接的佇列管理程式。
- `-sd delete` 指定在順利傳送之後刪除來源檔案。
- `-da AGENT2` 將目的地代理程式 (即檔案傳送至其中的代理程式) 定義為 `AGENT2`。
- `-w` 要求 **fteCreateTransfer** 指令等待其成功確認。
- `-dd C:\end\` 將目的地目錄定義為 `C:\end\`。
- `C:\start\myfile.txt` 定義要傳送的檔案。

如需相關資訊，請參閱 [fteCreateTransfer \(建立新的檔案傳送\)](#)。

3. 在 IBM WebSphere MQ Explorer 中檢查檔案傳送日誌，或手動檢查檔案系統，以確認檔案傳送成功。

下一步

您可能想要透過外部媒體來探索檔案傳送功能的更多特性。請參閱：

- [第 137 頁的『使用 Script 進行兩台電腦檔案傳送』](#)
- 進一步保護環境安全。您自己所需的存取模型，可能不同於此實務範例中使用的存取模型。如需此區域中最佳作法的相關資訊，請參閱 [保護 WebSphere MQ File Transfer Edition V7](#)。

詳細的基本檔案傳送

您可以使用 IBM WebSphere MQ Version 7.5，以多種不同的方式來傳送檔案。請閱讀本節中的主題，以瞭解此實務範例所涵蓋的內容、企業可能想要遵循實務範例的原因、涉及的使用者角色，以及實務範例所提出解決方案的概觀。

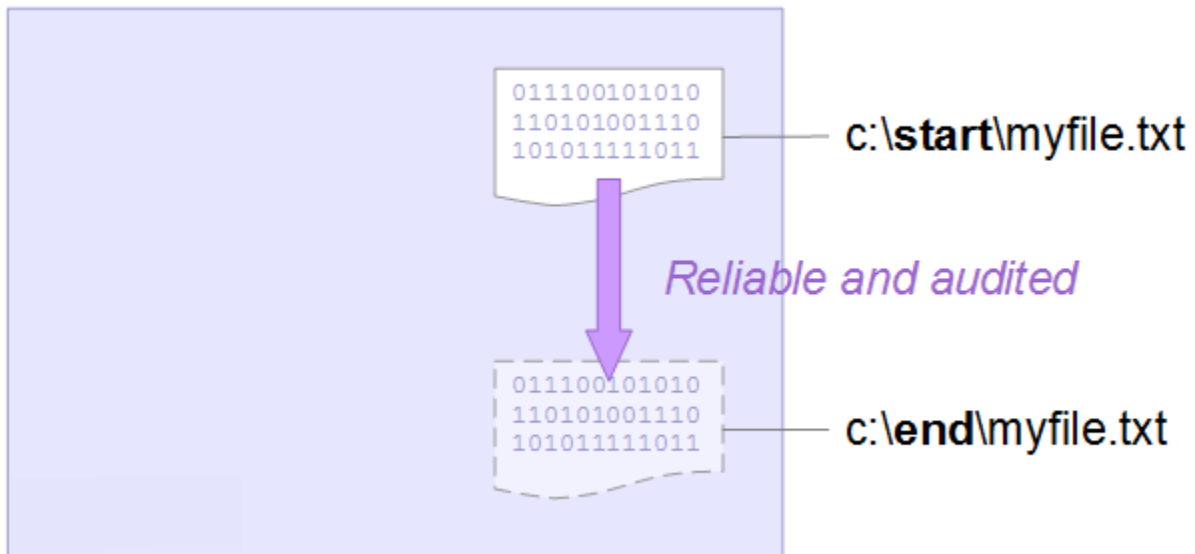
在企業中，具有控制、可靠且具有審核的傳送檔案可能是基本需求。IBM WebSphere MQ Version 7.5 提供「受管理檔案傳送」功能作為其整合傳訊平台的一部分。您可以使用「受管理檔案傳送」功能，透過基本檔案傳送或完全成熟的傳訊參與者，將檔案無縫整合至傳訊基礎架構。

此實務範例可讓您基本瞭解如何將檔案整合至最簡單的 IBM WebSphere MQ 傳訊拓撲。完成此基本 IBM WebSphere MQ 實務範例，設計為將檔案從一個位置移至另一個位置。雖然此起始實務範例僅限於單一電腦，但它提供您配置環境的經驗，並形成後續實務範例的重要基礎。此實務範例示範如何使用 IBM WebSphere MQ 透過網路傳送檔案，然後顯示「受管理檔案傳送」元件如何解決實際商業問題。

您應該對 IBM WebSphere MQ 有基本瞭解，特別是佇列管理程式的概念，以及 IBM WebSphere MQ 的基本配置和管理，直到使用 `runmqsc` 和「IBM WebSphere MQ 探險家」之類的指令。

概觀

IBM WebSphere MQ 可用來起始及追蹤單一電腦上從一個位置到另一個位置的檔案傳送。它提供您在 IBM WebSphere MQ Version 7.5 中安裝、配置及使用受管理檔案傳送功能的經驗，因此瞭解如何使用它來開始處理檔案傳送的實際商業問題。



如需規劃檔案傳送功能的詳細資料，請參閱 [WebSphere MQ Managed File Transfer 簡介](#)。

規劃解決方案

將檔案傳送至電腦、檔案傳送代理程式、必備項目、授權、準備電腦，以及針對基本檔案傳送實務範例安裝及配置 IBM WebSphere MQ 以進行檔案傳送。

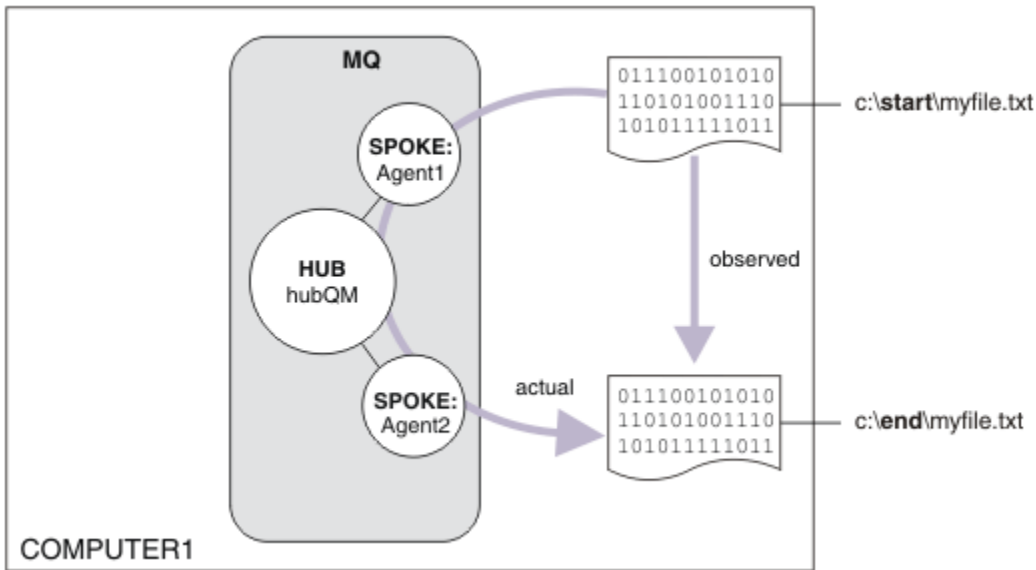
若要將檔案從 A 點傳送至 B 點，請定義 IBM WebSphere MQ 佇列管理程式及兩個檔案傳送代理程式。您已瞭解佇列管理程式是什麼，不過檔案傳送代理程式對您可能是新手。檔案傳送代理程式是在電腦上執行並與其他代理程式來回傳送檔案的 Java 處理程序。在此實務範例中，定義使用這些檔案傳送代理程式的檔案傳送，以透過兩種機制將範例檔案從一個位置移至另一個位置：

1. 使用圖形「IBM WebSphere MQ 檔案總管」。
2. 透過指令行。

逐步演練假設您有 Windows 系統。對於 UNIX 系統，請替換適當的路徑和指令，並確保您具有所有相關目錄的讀取權和寫入權。也假設您對佇列管理程式執行的動作有基本瞭解。

若要與電腦來回傳送檔案，您必須在該電腦上執行檔案傳送代理程式。每個代理程式都會連接至 IBM WebSphere MQ 佇列管理程式，並使用 IBM WebSphere MQ 來與其他代理程式進行通訊。如需相關資訊，請參閱 [WebSphere MQ Managed File Transfer 拓撲概觀](#)。

此實務範例可以使用管理來源及目的地檔案的單一檔案傳送代理程式。不過，此範例使用兩個檔案傳送代理程式，可讓您更實際瞭解檔案傳送在通常跨越網路的實際狀況下如何運作。



此圖顯示如何使用 IBM WebSphere MQ 來示範基本檔案傳送。觀察從一個位置複製到另一個位置的檔案。

必備項目及授權

您需要下列項目：

- 符合 IBM WebSphere MQ 的硬體及作業系統必要條件的測試電腦，如需詳細資料，請參閱 <https://www.ibm.com/support/docview.wss?uid=swg27006467>，且不存在 IBM WebSphere MQ 或 IBM WebSphere MQ 資料的現有安裝。
- IBM WebSphere MQ Version 7.5。您可以從 <https://www.ibm.com/developerworks/downloads/ws/wmq/> 下載試用版。

準備您的電腦

確保測試電腦滿足 IBM WebSphere MQ Version 7.5 安裝的需求，請參閱 [檢查需求](#)。

確保電腦已適當地準備好進行安裝，請參閱 [使用指令行驗證本端安裝](#)。

配置 WebSphere MQ 以進行檔案傳送

1. 建立佇列管理程式 hubQM。

輸入指令 `crtmqm hubQM`。如需詳細資料，請參閱 [crtmqm](#)。

2. 啟動佇列管理程式 hubQM。

輸入指令 `strmqm hubQM`。如需詳細資料，請參閱 [strmqm](#)。

3. 配置佇列管理程式 hubQM 以協調檔案傳送

- a. 輸入下列指令，以建立 IBM WebSphere MQ 的內容檔及協調佇列管理程式目錄：

```
fteSetupCoordination -coordinationQMgr hubQM
```

此指令會建立 IBM WebSphere MQ 的內容檔及協調佇列管理程式目錄。在此情況下，hubQM 會作為協調佇列管理程式來播送審核及檔案傳送資訊。如果 **fteSetupCoordination** 指令無法使用，則表示您已使用預設值安裝 WebSphere MQ。安裝額外套件 (例如，受管理檔案傳送) 可修正此問題。

如需相關資訊，請參閱 [IBM WebSphere MQ 受管理檔案傳送拓撲概觀](#) 及 [fteSetupCoordination \(設定協調詳細資料\)](#)。

- b. 輸入下列指令，以配置 hubQM 作為協調佇列管理程式：

```
runmqsc hubQM < <filepath from previous statement>
```

- c. 定義哪些佇列管理程式會處理檔案傳送指令，在此情況下為 hubQM。

```
fteSetupCommands -connectionQMgr hubQM
```

如需相關資訊，請參閱 [fteSetup 指令 \(建立 command.properties 檔案\)](#)。

4. 建立第一個檔案傳送代理程式 AGENT1。

- a. 透過輸入下列指令，準備檔案傳送代理程式 AGENT1，包括您必須針對代理程式連接至 hubQM 的佇列管理程式執行的 MQSC Script:

```
fteCreateAgent -agentName AGENT1 -agentQMgr hubQM
```

如需相關資訊，請參閱 [fteCreate 代理程式 \(建立 WebSphere MQ Managed File Transfer 代理程式\)](#)。

- b. 配置 hubQM 以處理您建立的代理程式。

```
runmqsc hubQM < <location of AGENT1_create.mqsc>
```

檔案的位置視您安裝 IBM WebSphere MQ 的位置而定。

5. 建立第二個檔案傳送代理程式 AGENT2。

```
fteCreateAgent -agentName AGENT2 -agentQMgr hubQM  
runmqsc hubQM < <location of AGENT2_create.mqsc>
```

6. 啟動 AGENT1。

```
fteStartAgent AGENT1
```

如需相關資訊，請參閱 [fteStartAgent \(啟動 WebSphere MQ Managed File Transfer 代理程式\)](#)。

7. 啟動 AGENT2。

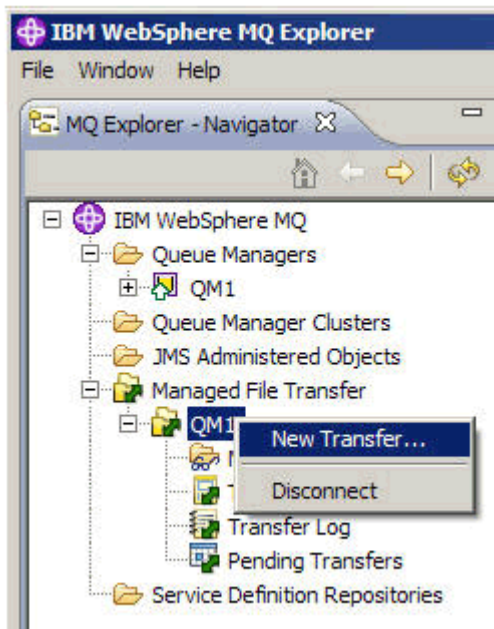
```
fteStartAgent AGENT2
```

實作解決方案

在此實務範例中實作解決方案包括使用 IBM WebSphere MQ Explorer 來定義及起始檔案傳送。您可以監視傳送進度，並透過檢查檔案系統來確認其成功。

程序

1. 識別來源檔和目標目錄:
 - a) 建立要傳送的範例檔，例如 C:\start\myfile.txt
 - b) 識別要將此檔案傳送至其中的現有目錄，例如，C:\end\
2. 啟動「IBM WebSphere MQ Explorer」。從「開始」功能表 (或對等項目) 啟動程式，或執行指令 **MQExplorer**。如需詳細資料，請參閱 [啟動 IBM WebSphere MQ Explorer](#)。
3. 按一下 IBM WebSphere MQ Explorer 導覽器中的 **受管理檔案傳送**，用滑鼠右鍵按一下 **QM**，然後選取 **新建傳送** 以啟動「新建傳送」精靈。



4. 在 **來源** 區段中選取 **AGENT1** 作為來源代理程式:

From:

Agent: AGENT1

Type: File

File:

Include subdirectories

5. 輸入您先前建立之檔案的路徑，例如 C:\start\myfile.txt。

From:

Agent: AGENT1

Type: File

File: C:\Users\Ben Bakowski\Desktop\FTETEST\START\myDemoTransfer.txt

Include subdirectories

6. 在 **目標** 區段中選取 **AGENT2** 作為目的地代理程式。

7. 輸入您先前識別的目的地目錄，例如 C:\end\。

To:

Agent: AGENT2

Type: File

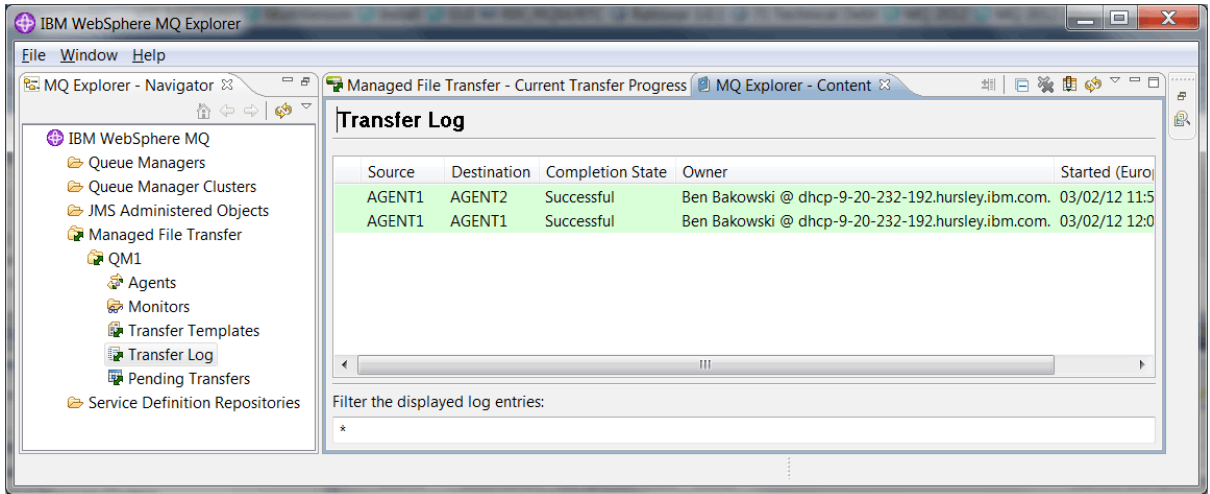
Directory: C:\Users\Ben Bakowski\Desktop\FTETEST\END

File name: myDemoTransfer.txt

Overwrite files on the destination file system that have the same name

8. 按一下 **立即完成**，即會開始傳送。

9. 您可以在 IBM WebSphere MQ Explorer 的 **現行傳送進度** 標籤中監視傳送進度:



10. 您也可以手動檢查檔案系統，以確認新檔案存在，例如 C:\end\myfile.txt。

使用指令行傳送檔案

您可以使用指令行來起始傳送，以示範檔案傳送的彈性。雖然超出此實務範例的範圍，但您可以根據這些原則及 Ant Scripting 技術來定義及實作更強大的檔案傳送實務範例。

程序

1. 從先前的示範中刪除已傳送的檔案，例如 C:\end\myfile.txt。
2. 使用 **fteCreateTransfer** 指令，以起始將檔案從 C:\start\myfile.txt 傳送至 C:\end\myfile.txt:

```
fteCreateTransfer -sa AGENT1 -sm hubQM -da AGENT2 -dm hubQM -w -dd C:\end\  
C:\start\myfile.txt
```

- -sa AGENT1 將來源代理程式 (即從中傳送檔案的代理程式) 定義為 AGENT1。
- -sm hubQM 定義來源代理程式 AGENT1 所連接的佇列管理程式。
- -da AGENT2 將目的地代理程式 (即檔案傳送至其中的代理程式) 定義為 AGENT2。
- -w 要求 **fteCreateTransfer** 指令等待其成功確認。
- -dd C:\end\ 將目的地目錄定義為 C:\end\。
- C:\start\myfile.txt 定義要傳送的檔案。

如需相關資訊，請參閱 [fteCreateTransfer \(建立新的檔案傳送\)](#)。

3. 在 IBM WebSphere MQ Explorer 中檢查檔案傳送日誌，或手動檢查檔案系統，以確認檔案傳送成功。

下一步

您可能想要透過外部媒體來探索檔案傳送功能的更多特性。請參閱：

- 第 137 頁的『[使用 Script 進行兩台電腦檔案傳送](#)』
- 進一步保護環境安全。您自己所需的存取模型，可能不同於此實務範例中使用的存取模型。如需此區域中最佳作法的相關資訊，請參閱 https://www.ibm.com/developerworks/websphere/library/techarticles/0902_wyatt/0902_wyatt.html。

使用 Script 進行兩台電腦檔案傳送

使用 Script 實務來延伸基本檔案傳送，以將檔案傳送整合至多電腦 IBM WebSphere MQ 傳訊拓撲。

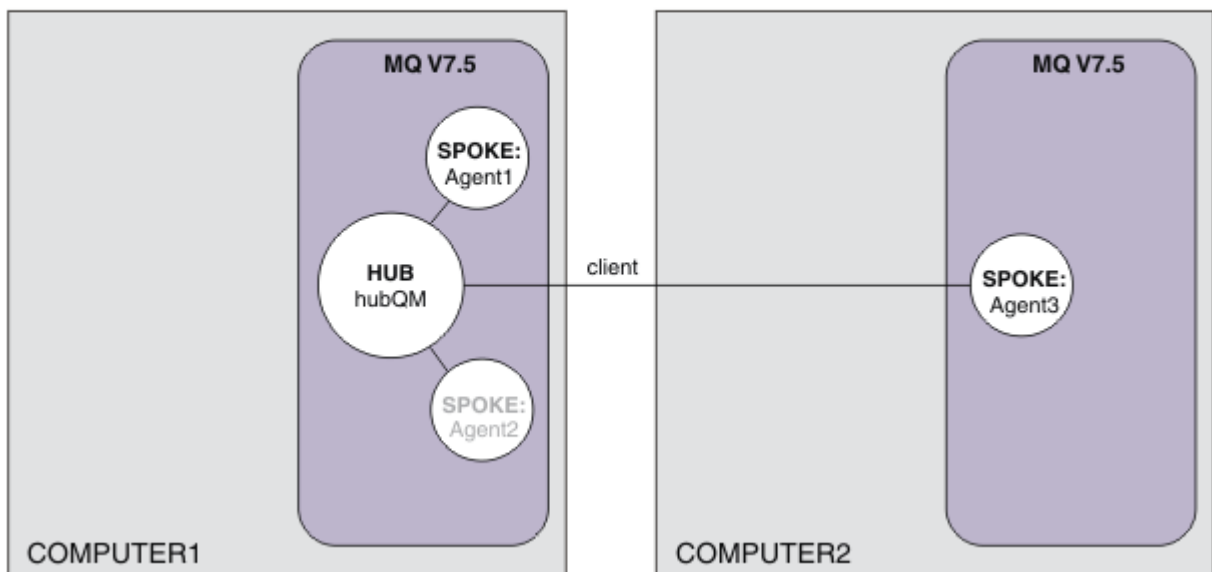
建置「受管理檔案傳送」功能的簡單示範(請參閱第 124 頁的『使用 Script 的基本檔案傳送』)之後,您現在已熟悉受管理檔案傳送背後的基本原則。您可以辨識在單一電腦上此功能的好處不大,因此現在您可以延伸實務範例,以探索如何將檔案傳送整合至多電腦 IBM WebSphere MQ 傳訊拓撲。

開始存取基礎 IBM WebSphere MQ 技術所提供的好處,亦即僅一次可靠的檔案遞送。為了達到此目的,拓撲會延伸至包括參與檔案傳送的第二部電腦。在此實務範例中,您安裝並配置個別可用的受管理檔案傳送代理程式,並開始瞭解適用於多電腦檔案傳送拓撲的安全考量。此實務範例透過示範從一部電腦到下一部電腦的檔案傳送,為您可以新增審核功能的下一個實務範例提供強大基礎,從而顯示為何這是受管理檔案傳送功能。

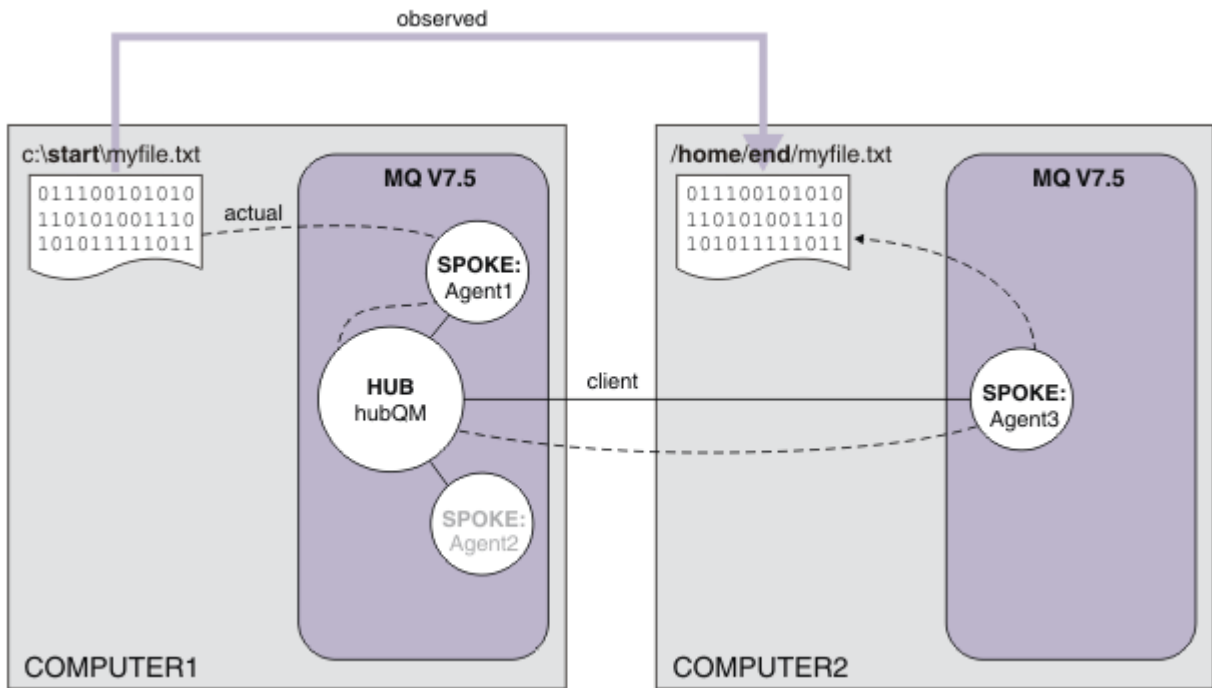
假設您已完成第 124 頁的『使用 Script 的基本檔案傳送』中提供的作業。

概觀

在此實務範例中,第 124 頁的『使用 Script 的基本檔案傳送』實務範例中配置的中心及分支拓撲延伸至包括第二部電腦(電腦 2)。電腦 2 已安裝「受管理檔案傳送代理程式」及「指令行工具」;這第二部電腦上沒有本端 IBM WebSphere MQ 伺服器安裝的必備項目。這類模型在中心及分支解決方案中很受歡迎,其中多個「受管理檔案傳送代理程式」透過用戶端連線與集中式 IBM WebSphere MQ 佇列管理程式互動,而不需要在每一個分支上安裝 IBM WebSphere MQ 伺服器及相關聯的授權。支援其他拓撲,且您自己的拓撲視您的特定需求及授權或授權而定。AGENT2 已淘汰,以支援單獨使用 AGENT1 來處理此電腦上的傳送。



此圖顯示跨越兩台電腦的中心及分支拓撲。AGENT1 和 AGENT2 配置成使用連結至(本端)中心,而 AGENT3 則透過用戶端連線進行連接。AGENT2 已停用,因為它在實務中沒有進一步的角色。AGENT1 會處理電腦 1 上的所有檔案傳送活動。請確定您瞭解安全模型及其限制,如需詳細資料,請參閱第 124 頁的『使用 Script 的基本檔案傳送』實務範例。配置時,此拓撲用於將檔案從電腦 1 傳送至電腦 2。



此圖顯示示範的檔案傳送路徑。同樣地，基礎檔案傳送是透過可靠且作用中的 IBM WebSphere MQ 連線進行。在此範例中，假設電腦 1 是 Windows 電腦，而電腦 2 是 Linux 電腦。您可能想要使用替代平台及架構，如需受支援平台的完整清單，請參閱 <https://www.ibm.com/support/docview.wss?uid=swg27006467#7.1>。

規劃解決方案

說明與電腦之間來回傳送檔案、檔案傳送代理程式、必備項目、授權、準備您的電腦，以及使用 Script 實務為兩部電腦檔案傳送配置 IBM WebSphere MQ。

必要條件

您需要下列項目：

- 電腦 1，來自第 124 頁的『使用 Script 的基本檔案傳送』的工作中配置。
- 電腦 2，滿足 IBM WebSphere MQ Version 7.5 的硬體及作業系統必要條件的第二部測試電腦。如需相關資訊，請參閱 <https://www.ibm.com/support/docview.wss?uid=swg27006467>。
- IBM WebSphere MQ Version 7.5。您可以從 <https://www.ibm.com/developerworks/downloads/ws/wmq/> 下載試用版。
- 瞭解電腦 1 和電腦 2 的 IP 位址，採用 nnn.nnn.nnn.nnn 格式。

安裝 IBM WebSphere MQ Version 7.5

使用下列元件來安裝 IBM WebSphere MQ Version 7.5 伺服器：伺服器、IBM WebSphere MQ Explorer、IBM WebSphere MQ 進階受管理檔案傳送代理程式及 IBM WebSphere MQ 受管理檔案傳送指令行工具。如需詳細資料，請參閱 [選擇要安裝的項目](#)。

決定要管理 IBM WebSphere MQ 的方式。您可以透過下列方式管理 IBM WebSphere MQ：

- 使用 **setmqenv** 指令設定適當的環境。如需相關資訊，請參閱 [setmqenv](#)。
- 呼叫完整的 IBM WebSphere MQ 管理指令。

準備使用者和群組

此安全模型假設您具有下列群組和使用者：

群組

- mqm

在 IBM WebSphere MQ 安裝期間自動建立。此群組的成員可以管理 IBM WebSphere MQ 及其資源。

- FTUSERS

您必須建立此群組。此群組的成員可以起始檔案傳送。

- FTAGENTS

您必須建立此群組。此群組的成員可以啟動及停止檔案傳送代理程式，這些代理程式是處理網路中檔案傳送的端點。

使用者

- mqmAdmin

IBM WebSphere MQ 管理者，即 mqm 群組的成員，或在 Windows 上是 Administrators 群組的成員。

- ftuser

您必須建立此使用者，並使其成為 FTUSERS 群組的成員。請勿將此使用者新增至 mqm 群組，以避免可能在佇列管理程式上執行管理安全動作。

- ftagent

您必須建立此使用者，並使其成為 FTAGENTS 群組的成員。請勿將此使用者新增至 mqm 群組，以避免可能在佇列管理程式上執行管理安全動作。

在電腦 2 上新增代理程式 AGENT3 作為分支

準備電腦 2，以使用 Script 實務範例來支援 IBM WebSphere MQ 兩台電腦檔案傳送的延伸拓撲。

關於這項作業

再次從電腦 1 使用 addSpoke 範例 Script，定義透過用戶端連線連接的代理程式分支。**addSpoke** 指令的完成方式是向您呈現一組指令，可從電腦 2 上的 IBM WebSphere MQ 安裝架構執行。

程序

1. 在電腦 1 上，以使用者 mqmAdmin 身分，從 IBM WebSphere MQ bin 目錄 <MQ_INSTALL_ROOT>\bin 執行 **addspoke** 指令。

```
<MQ_INSTALL_ROOT>\mqft\samples\scripts\addSpoke agentName=AGENT3
hubQmgr=hubQM connectionMode=CLIENT agentIPAddress=<IP address of computer 2>
hubIPAddress=<IP address of computer 1> hubPort=1414
```

2. 在電腦 2 上，以使用者 mqmAdmin 身分，從電腦 1 執行 **addSpoke** 指令所輸出的指令，例如：
 - a) `fteSetupCoordination -coordinationQMgr hubQM-coordinationQMgrHost <IP address of computer 1> -coordinationQMgrPort 1414 -coordinationQMgrChannel FTE.USER.SVRCONN -f`
 - b) `fteSetupCommands -p hubQM -connectionQMgr hubQM -connectionQMgrHost <IP address of computer 1> -connectionQMgrPort 1414 -connectionQMgrChannel FTE.USER.SVRCONN -f`
 - c) `fteCreateAgent -p hubQM -agentName AGENT3 -agentQMgr hubQM -agentQMgrHost <IP address of computer 1> -agentQMgrPort 1414 -agentQMgrChannel FTE.AGENT.SVRCONN -f`

請注意，您不需要執行產生的 MQSC Script。

3. 在電腦 2 上，以使用者 ftagent 身分啟動 AGENT3

```
fteStartAgent -p hubQM AGENT3
```

- 在電腦 2 上，以使用者 `ftagent` 身分確認三個代理程式可用 (必要的話，在使用者 `ftagent` 下啟動電腦 1 上的代理程式)。

```
ftelListAgents -p hubQM
```

您將看到下列輸出：

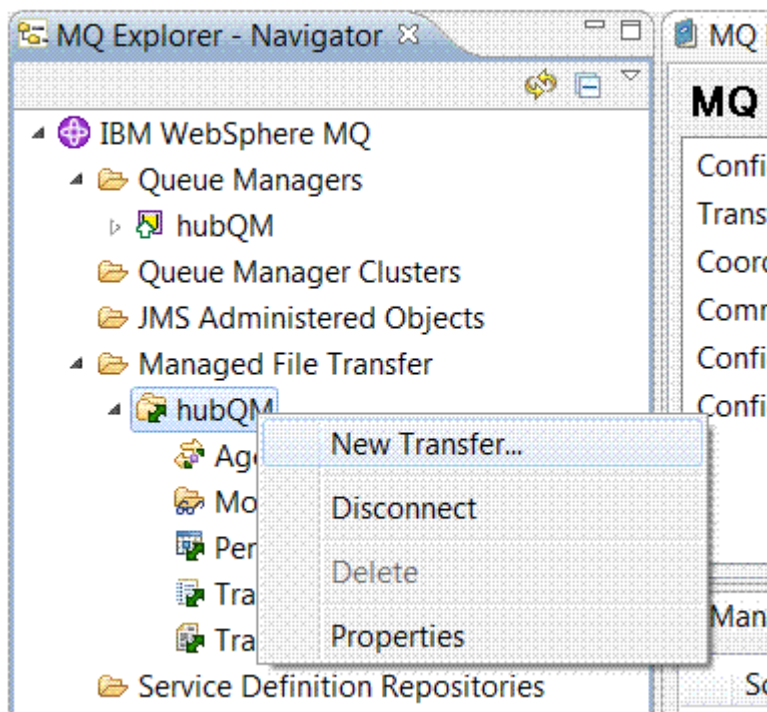
Agent Name:	Queue Manager Name:	Status:
AGENT1	hubQM	READY
AGENT2	hubQM	READY
AGENT2	hubQM	READY

實作解決方案

透過使用 IBM WebSphere MQ 來傳送檔案，開始使用 Script 實務範例進行兩台電腦檔案傳送的示範。監視傳送進度，並透過檢查檔案系統來確認其成功。

程序

- 識別來源檔和目標目錄：
 - 在電腦 1 上建立要傳送的範例檔，例如 `C:\start\myfile.txt`
 - 識別電腦 2 上要將此檔案傳送至其中的現有目錄，例如 `C:\end\`。確保使用者 `ftagent` 具有目錄的寫入權。
- 以使用者身分在電腦 1 上啟動 IBM WebSphere MQ Explorer `mqmAdmin`。請注意，如同先前的實務範例，請以 IBM WebSphere MQ 管理者身分執行此動作，以便將實務範例集中在檔案傳送，而不是配置 IBM WebSphere MQ Explorer。從「開始」功能表 (或對等項目) 啟動程式，或執行指令 `MQExplorer`。如需詳細資料，請參閱 [啟動 WebSphere MQ 探險家](#)。
- 在 IBM WebSphere MQ 導覽器中的 **受管理檔案傳送** 區段下，用滑鼠右鍵按一下 `hubQM`，然後選取 **新建傳送** 以啟動「新建傳送」精靈。



- 使用功能表來選取 AGENT1 作為來源代理程式，並選取 AGENT3 作為目的地代理程式：

Source agent

Name: AGENT1

Type: Standard

Destination agent

Name: AGENT3

Type: Standard

5. 按下一步，然後按一下 **新增 ...** 以及針對來源，輸入您要傳送之檔案的路徑，例如 C:\start\myfile.txt。確定已勾選 **如果傳送成功則移除來源檔** 勾選框。

Source

Agent: AGENT1

Type: Standard agent

Type: File

File name: C:\start\myFile.txt

Remove source file if the transfer is successful

6. 輸入目的地目錄。在此實務範例中，此目的地位於 Linux 平台上，因此請使用適當的 /home/end/ 表示法。

Destination

Agent: AGENT3

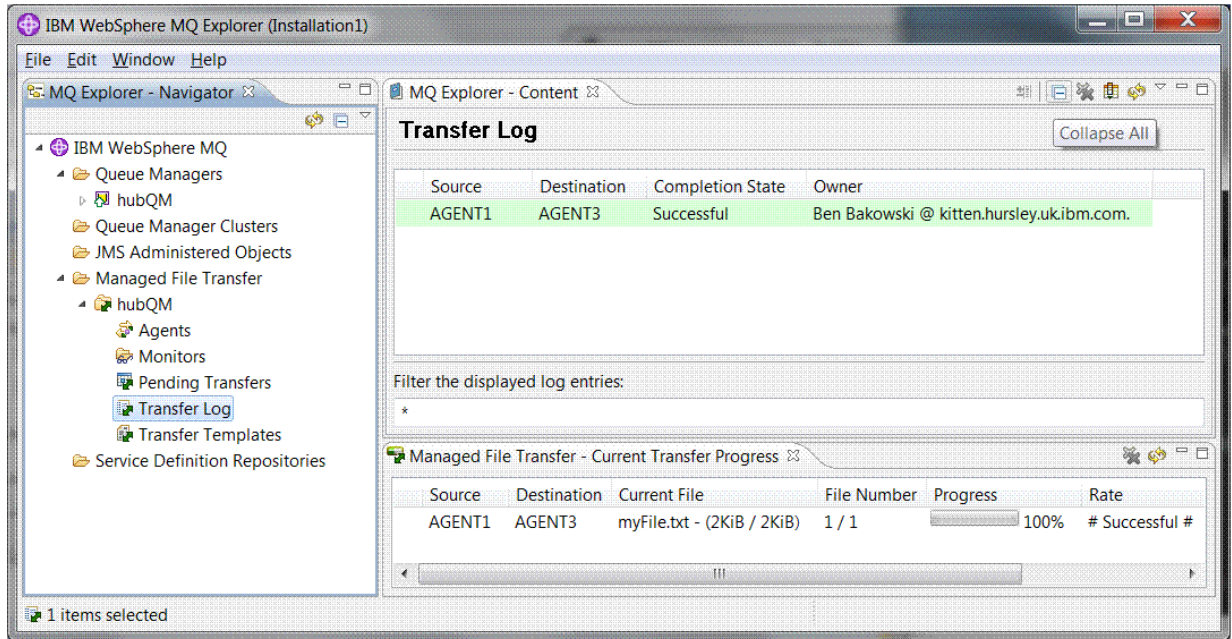
Type: Standard agent

Type: Directory

Directory: /home/end

Overwrite files if present

7. 按一下 **確定**，然後按一下 **完成**，即會開始傳送。
8. 您可以在 IBM WebSphere MQ Explorer 中的 **現行傳送進度** 標籤中監視傳送進度。



9. 您也可以手動檢查檔案系統，以確認新檔案存在，例如 `/home/end/myfile.txt`。

使用指令行傳送檔案

您可以使用指令行來起始傳送，以示範檔案傳送的彈性。雖然超出此實務範例的範圍，但您可以根據這些原則及 Ant Scripting 技術來定義及實作更強大的檔案傳送實務範例。

程序

1. 從先前的示範中刪除已傳送的檔案 (例如 `/home/end/myfile.txt`)，並在其原始位置重建該檔案。
2. 使用 `ftuser` 時，請使用中的 `fteCreateTransfer` 指令，以起始將檔案從電腦 1 上的 `C:\start\myfile.txt` 傳送至電腦 2 上的 `/home/end/myfile.txt`：

```
fteCreateTransfer -sa AGENT1 -sm hubQM -da AGENT3 -dm hubQM -w -dd "/home/end/"
"C:\start\myfile.txt"
```

3. 在 IBM WebSphere MQ Explorer 中檢查檔案傳送日誌，或手動檢查檔案系統，以確認檔案傳送成功。

下一步

如需 IBM WebSphere MQ 安全的詳細資料，請參閱 [安全](#)。

您可能想要透過外部媒體來探索檔案傳送功能的更多特性。

- 如需新增審核功能以提供受管理檔案傳送的受管理層面的詳細資料，請參閱 [第 151 頁的『將審核功能新增至受管理檔案傳送』](#)。
- 如需下列相關詳細資料：
 - 觸發：當新檔案出現時移動它們。
 - 觸發：配置單一檔案的外觀，以起始多個檔案的傳送。
 - 使用 Apache Ant 進行 Scripting 傳送。

請參閱 https://www.ibm.com/developerworks/websphere/library/techarticles/1003_phillips/1003_phillips.html。

- 您自己所需的存取模型，可能不同於此實務範例中使用的存取模型。如需進一步保護環境安全的最佳作法相關資訊，請參閱 https://www.ibm.com/developerworks/websphere/library/techarticles/0902_wyatt/0902_wyatt.html。

詳細傳送兩部電腦檔案

延伸基本檔案傳送實務範例，以將檔案傳送整合至多電腦 IBM WebSphere MQ 傳訊拓撲。

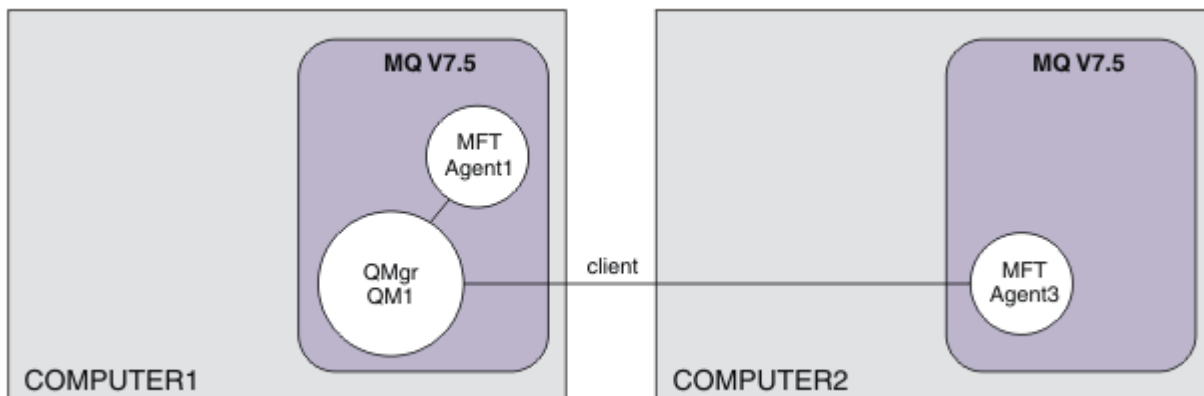
完成「受管理檔案傳送」功能的簡單示範(請參閱第 132 頁的『詳細的基本檔案傳送』)之後，您現在已熟悉受管理檔案傳送背後的基本原則。您可以辨識在單一電腦上此功能的好處不大，因此現在您可以延伸實務範例，以探索如何將檔案傳送整合至多電腦 IBM WebSphere MQ 傳訊拓撲。

開始存取基礎 IBM WebSphere MQ 技術所提供的好處，亦即僅一次可靠交付檔案。為了達到此目的，拓撲會延伸至包括參與檔案傳送的第二部電腦。在此實務範例中，您安裝並配置個別可用的受管理檔案傳送代理程式，並開始瞭解適用於多電腦檔案傳送拓撲的安全考量。此實務範例透過示範從一部電腦到下一部電腦的檔案傳送，為您可以新增審核功能的下一個實務範例提供強大基礎，從而顯示為何這是受管理檔案傳送功能。

假設您已完成第 132 頁的『詳細的基本檔案傳送』中提供的作業。

概觀

在此實務範例中，您將繼續使用基本檔案傳送實務範例中配置的現有 Windows 電腦。Agent2 已停用，因為您使用單一代理程式作為此第一部電腦上的檔案傳送端點。在第二部電腦上，您安裝「受管理檔案傳送代理程式」；在這第二部電腦上，本端 IBM WebSphere MQ 伺服器安裝不是必備項目。這類模型在中心及分支解決方案中很受歡迎，其中多個「受管理檔案傳送代理程式」透過用戶端連線與集中式 IBM WebSphere MQ 佇列管理程式互動，而不需要在每一個分支上安裝 IBM WebSphere MQ 伺服器及相關聯的授權。支援其他拓撲，且您自己的拓撲視您的特定需求及授權或授權而定。

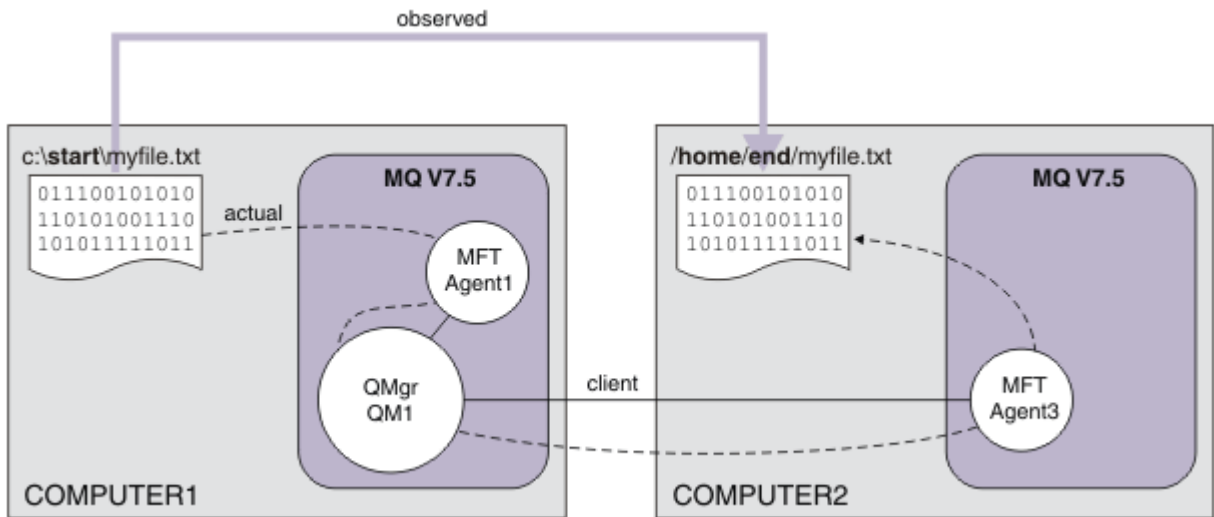


在檔案傳送網路中，會選取單一佇列管理程式作為網路中的單一點，以播送審核及檔案傳送資訊。在先前的第 132 頁的『詳細的基本檔案傳送』實務範例中，隱含地使用單一佇列管理程式作為協調佇列管理程式。在此實務範例中，您繼續使用電腦 1 上的佇列管理程式 QM1 作為協調佇列管理程式，且電腦 2 配置的一部分會設定環境以使用此佇列管理程式。

在此拓撲中，不需要能夠從電腦 2 起始檔案傳送，因此請不要在此第二部電腦上安裝選用的「受管理檔案傳送指令行工具」元件。

請務必注意，雖然考量部分基本安全特性，但檔案傳送拓撲未受到保護，無法達到您可能認為適當的層次。如需保護 IBM WebSphere MQ 安全，特別是檔案傳送的相關討論，請參閱 https://www.ibm.com/developerworks/websphere/library/techarticles/0902_wyatt/0902_wyatt.html。

安裝在第二部電腦上時，您可以建立實際代理程式 Agent3，然後透過將檔案從電腦 1 移至電腦 2 來示範解決方案。



基礎檔案傳送是透過可靠且執行 IBM WebSphere MQ 連線來進行。接下來的作業顯示如何配置及適當保護這些連線的安全。

在此範例中，假設電腦 1 是 Windows 電腦，而電腦 2 是 Linux 電腦。您可能想要使用替代平台及架構，如需受支援平台的完整清單，請參閱 <https://www.ibm.com/support/docview.wss?uid=swg27006467#7.1>。

規劃解決方案

規劃兩台電腦檔案傳送解決方案，包括適當基礎架構的說明，以及您需要建立的群組和使用者。

開始之前

您需要下列項目：

- 基本檔案傳送實務中的工作中配置，如需詳細資料，請參閱第 132 頁的『詳細的基本檔案傳送』。
- IBM WebSphere MQ Version 7.5。您可以從 <https://www.ibm.com/developerworks/downloads/ws/wmq/> 下載試用版。
- 第二部測試電腦滿足 IBM WebSphere MQ Version 7.5 的硬體及作業系統必要條件，如需詳細資料，請參閱 <https://www.ibm.com/support/docview.wss?uid=swg27006467>。

在此實務範例中，第二部電腦透過用戶端連線通道與佇列管理程式互動。使用通道鑑別記錄可確保：

- 送入的連線會鑑別為源自管理進一步受管理檔案傳送代理程式的新電腦。
- 送入要求會對映至對受管理檔案傳送資源具有適當存取權的使用者。

沙盤推演是一種限制檔案傳送以操作檔案系統特定區域的技術，例如 IBM WebSphere MQ 配置檔。

程序

在兩台電腦上：

1. 建立群組 FTEUSERS 和 FTEAGENTS。如需詳細資料，請參閱作業系統指示。
2. 建立使用者 fteuser，並將它新增至 FTEAGENTS 群組。
3. 建立使用者 fteagent，並將它新增至 FTEUSERS 群組。

使用者 fteuser 會起始檔案傳送，且使用者 fteagent 會啟動及停止代理程式處理程序。這些使用者不是 mqm (或 Windows 管理者) 群組的成員，因此有助於強化拓撲，防止不想要的管理攻擊。您可以透過在代理程式基礎上定義特定使用者 (例如，fteagent1、fteagent2) 來提供更精細的安全存取權。

修改電腦 1 以支援延伸拓撲

準備電腦 1 以支援 IBM WebSphere MQ 兩台電腦檔案傳送實務範例的延伸拓撲。

關於這項作業

在此作業中，您刪除 Agent2，因為不再需要它。然後，您可以建立並啟動接聽器，以接受來自電腦 2 上所建立代理程式的用戶端連線，從而在兩台電腦上建構簡單的 IBM WebSphere MQ 網路。假設電腦 1 正在執行 Windows 作業系統。如果您使用不同的平台來執行此實務範例，請替換適當的平台專用指令。

程序

1. 停止代理程式 Agent2。

```
fteStopAgent AGENT2
```

如需 **fteStopAgent** 指令的詳細資料，請參閱 [fteStop 代理程式 \(停止 WebSphere MQ Managed File Transfer 代理程式\)](#)。

2. 刪除代理程式 Agent2。

```
fteDeleteAgent AGENT2  
runmqsc QM1 < <output>
```

如需 **fteDeleteAgent** 指令的詳細資料，請參閱 [fteDelete 代理程式 \(刪除 WebSphere MQ Managed File Transfer 代理程式\)](#)。

配置 IBM WebSphere MQ 安全，以便電腦 2 上配置的新檔案傳送代理程式可以與協調佇列管理程式 QM1 互動。此新代理程式透過現有 SYSTEM.DEF.SVRCONN 通道連接至 QM1。您自己的安全需求可能不同，如需強化此拓撲的進一步詳細資料，請參閱 [下一步](#)。

3. 啟動 QM1 的 MQSC 介面。

```
runmqsc QM1
```

4. 建立兩個通道以處理來自使用者及代理程式的送入要求。

```
DEFINE CHANNEL(FTE.USER.SVRCONN) CHLTYPE(SVRCONN)  
DEFINE CHANNEL(FTE.AGENT.SVRCONN) CHLTYPE(SVRCONN)
```

5. 建立通道鑑別記錄，以容許從電腦 2 到 QM1 的連線，並指派所建立的使用者。

```
SET CHLAUTH('FTE.USER.SVRCONN') TYPE(ADDRESSMAP) ADDRESS('<IP address of computer2>')  
USERSRC (MAP) MCAUSER('fteuser' DESCR('Rule to allow file transfer users to communicate'))  
ACTION(ADD)  
  
SET CHLAUTH('FTE.AGENT.SVRCONN') TYPE(ADDRESSMAP) ADDRESS('<IP address of computer2>')  
USERSRC(MAP) MCAUSER('fteagent') DESCR('Rule to allow file transfer agent processes to  
communicate') ACTION(ADD)
```

如需詳細資料，請參閱 [通道鑑別記錄](#)。

此實務範例的目標不是鎖定及強化拓撲，而是示範基本檔案傳送。實作的安全模型支援此示範，但您必須瞭解自己的安全威脅，並在必要時採取適當的動作。如需考量選項的討論，請參閱 [下一步](#)。

6. 識別可用於與 IBM WebSphere MQ 進行網路通訊的可用埠。定義接聽器 LISTENER1 以使用此可用埠，例如 1414。

```
DEFINE LISTENER(LISTENER1) TRPTYPE(TCP) CONTROL(QMGR) PORT(1414)
```

7. 啟動接聽器 LISTENER1。

```
START LISTENER(LISTENER1)
```

8. 停止 QM1 的 MQSC 介面。

```
end
```

9. 檢查 FTEAGENTS 及 FTEUSERS 群組是否具有 IBM WebSphere MQ 物件的適當存取權，以針對要在電腦 2 上建立的代理程式 AGENT3 執行檔案傳送動作。您可能想要自訂此配置，以符合您自己的安全需求。

- a. `setmqaut -m QM1 -t qmgr -g FTEAGENTS +connect +inq`
- b. `setmqaut -m QM1 -t qmgr -g FTEUSERS +connect`
- c. `setmqaut -m QM1 -n "SYSTEM.FTE" -t q -g FTEAGENTS +get +put`
- d. `setmqaut -m QM1 -n "SYSTEM.FTE.COMMAND.AGENT1" -t q -g FTEUSERS +put`
- e. `setmqaut -m QM1 -n "SYSTEM.FTE.COMMAND.AGENT1" -t q -g FTEAGENTS +setid +get +put`
- f. `setmqaut -m QM1 -n "SYSTEM.FTE.COMMAND.AGENT3" -t q -g FTEUSERS +put`
- g. `setmqaut -m QM1 -n "SYSTEM.FTE.COMMAND.AGENT3" -t q -g FTEAGENTS +setid +get +put`
- h. `setmqaut -m QM1 -n "SYSTEM.FTE.DATA.AGENT1" -t q -g FTEAGENTS +get +put`
- i. `setmqaut -m QM1 -n "SYSTEM.FTE.DATA.AGENT3" -t q -g FTEAGENTS +get +put`
- j. `setmqaut -m QM1 -n "SYSTEM.FTE.EVENT.AGENT1" -t q -g FTEAGENTS +get +put`
- k. `setmqaut -m QM1 -n "SYSTEM.FTE.EVENT.AGENT3" -t q -g FTEAGENTS +get +put`
- l. `setmqaut -m QM1 -n "SYSTEM.FTE.REPLY.AGENT1" -t q -g FTEAGENTS +get +put`
- m. `setmqaut -m QM1 -n "SYSTEM.FTE.REPLY.AGENT3" -t q -g FTEAGENTS +get +put`
- n. `setmqaut -m QM1 -n "SYSTEM.FTE.STATE.AGENT1" -t q -g FTEAGENTS +get +put +inq`
- o. `setmqaut -m QM1 -n "SYSTEM.FTE.STATE.AGENT3" -t q -g FTEAGENTS +get +put +inq`
- p. `setmqaut -m QM1 -n "SYSTEM.FTE" -t topic -g FTEUSERS +sub`
- q. `setmqaut -m QM1 -n "SYSTEM.FTE" -t topic -g FTEAGENTS +pub +sub`
- r. `setmqaut -m QM1 -n "SYSTEM.DEFAULT.MODEL.QUEUE" -t q -g FTEUSERS +dsp +browse +get +put`
- s. `setmqaut -m QM1 -n "SYSTEM.DEFAULT.MODEL.QUEUE" -t q -g FTEAGENTS +dsp +browse +get +put`

如需 **setmqaut** 指令的詳細資料，請參閱 [setmqaut](#)。

如需將權限授與群組的詳細資料，請參閱 [WebSphere MQ Managed File Transfer](#) 特定資源的群組權限。

準備電腦 2 以進行檔案傳送

說明準備電腦 2 以進行 IBM WebSphere MQ 兩台電腦檔案傳送實務範例的檔案傳送。

關於這項作業

此作業假設電腦 2 正在執行 Linux 作業系統。如果您使用不同的平台來執行此實務範例，則必須替換適當的平台專用指令。

程序

1. 將 IBM WebSphere MQ 與「受管理檔案傳送代理程式」元件及任何適當的必備元件 (例如「受管理檔案傳送指令行工具」) 一起安裝在每一部電腦上。如需詳細資料，請參閱 [選擇要安裝的項目](#)。

此步驟假設您使用的是未安裝先前安裝 IBM WebSphere MQ 或 IBM WebSphere MQ 檔案傳送版本的全新電腦。如果沒有，您必須判斷是否支援共存性，並適當地調整環境的安裝機制或配置，如需詳細資料，請參閱 [多個安裝](#)。

安裝「受管理檔案傳送指令行工具」元件可讓您除了電腦 1 之外，還可以定義及起始電腦 2 的傳送，如本實務範例所示。

2. 以 mqm 群組中的使用者身分，將檔案傳送配置為使用電腦 1 上的 QM1 作為協調管理程式。輸入下列指令：

```
fteSetupCoordination -coordinationQMgr QM1 -coordinationQMgrHost <computer1_hostname>
-coordinationQMgrPort 1414 -coordinationQMgrChannel FTE.USER.SVRCONN
```

您不需要執行產生的 MQSC Script，因為您在配置電腦 1 時已執行它。

```
fteSetupCommands -connectionQMgr QM1 -connectionQMgrHost <computer1_hostname>
-connectionQMgrPort 1414 -connectionQMgrChannel FTE.USER.SVRCONN
```

如需相關資訊，請參閱：

- [WebSphere MQ Managed File Transfer 拓撲概觀](#)
- [fteSetupCoordination](#)（設定協調詳細資料）
- [fteSetupCommands](#)（建立 command.properties 檔案）

3. 列出向 QM1 登錄的代理程式，以確保用戶端連線功能的配置正確。執行下列指令：

```
fteListAgents
```

您會看到下列輸出：

Agent Name:	Queue Manager Name:	Status:
AGENT1	QM1	Ready

4. 建立第二個檔案傳送代理程式 AGENT3:

```
fteCreateAgent -agentName AGENT3 -agentQMgr QM1 -agentQMgrHost <computer1_hostname>
-agentQMgrPort 1414 -agentQMgrChannel FTE.AGENT.SVRCONN
```

切換至電腦 1，並輸入下列指令：

```
runmqsc QM1 < <AGENT3_create.mqsc>
```

5. 切換至電腦 2，並以使用者 fteagent 身分啟動新的檔案傳送代理程式 AGENT3。

```
fteStartAgent AGENT3
```

6. 選用項目: 切換至電腦 1，並以使用者 fteagent 而非第一個實務範例中使用的特許 IBM WebSphere MQ 管理者使用者身分重新啟動 AGENT1。
7. 列出向 QM1 登錄的代理程式，以檢查用戶端連線功能的配置是否正確。

```
fteListAgents
```

您會看到下列輸出：

Agent Name:	Queue Manager Name:	Status:
AGENT1	QM1	READY
AGENT3	QM1	READY

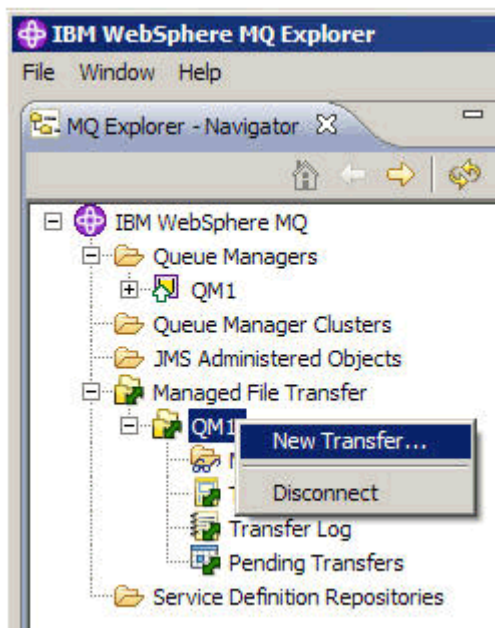
實作解決方案

在此實務範例中實作解決方案包括使用 IBM WebSphere MQ Explorer 來定義及起始檔案傳送。您可以監視傳送進度，並透過檢查檔案系統來確認其成功。

程序

1. 識別來源檔和目標目錄:
 - a) 在電腦 1 上建立要傳送的範例檔，例如 C:\start\myfile.txt。
 - b) 識別電腦 2 上要將此檔案傳送至其中的現有目錄，例如 /home/end/。請確保啟動代理程式的使用者具有該目錄的寫入權。
2. 在電腦 1 上啟動 IBM WebSphere MQ Explorer。從「開始」功能表(或對等項目)啟動程式，或執行指令 **MQExplorer**。如需詳細資料，請參閱 [啟動 IBM WebSphere MQ Explorer](#)。

3. 在「IBM WebSphere MQ 探險家」導覽器中按一下 **受管理檔案傳送**，用滑鼠右鍵按一下 **QM**，然後選取 **新建傳送** 以啟動「新建傳送」精靈。



4. 在 **來源** 區段中選取 **AGENT1** 作為來源代理程式:

From:

Agent: AGENT1

Type: File

File:

Include subdirectories

5. 輸入您先前建立之檔案的路徑，例如 C:\start\myfile.txt。

From:

Agent: AGENT1

Type: File

File: C:\Users\Ben Bakowski\Desktop\FTETEST\START\myDemoTransfer.txt

Include subdirectories

6. 在 **目標** 區段中選取 **AGENT2** 作為目的地代理程式。
7. 輸入您先前識別的目的地目錄，例如 /home/end/。

To:

Agent: AGENT2

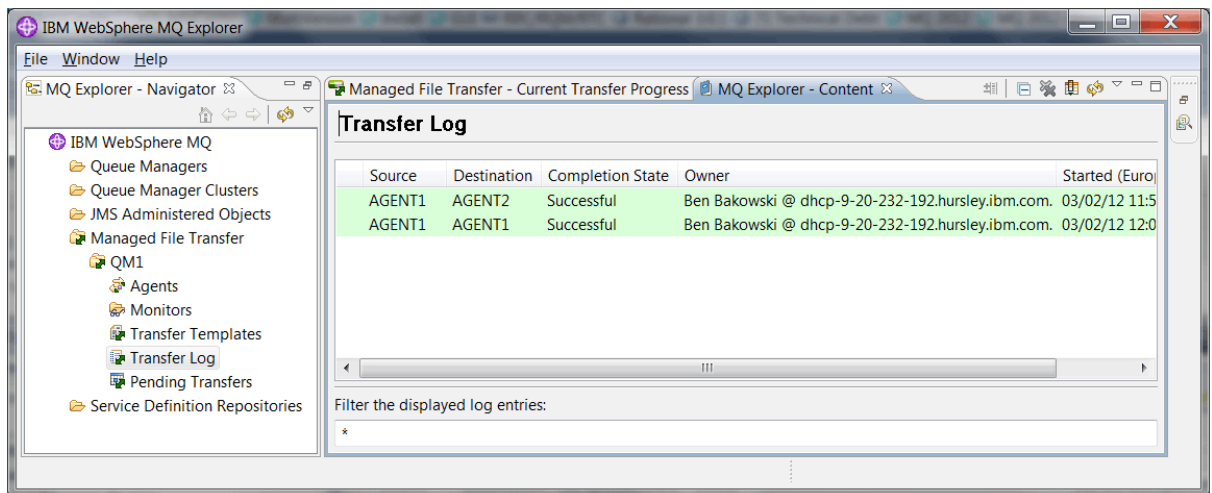
Type: File

Directory: /home/ben/FTETEST/END/

File name: myDemoTransfer.txt

Overwrite files on the destination file system that have the same name

- 按一下 **立即完成**，即會開始傳送。
- 您可以在 IBM WebSphere MQ Explorer 中的 **現行傳送進度** 標籤中監視傳送進度：



- 您也可以手動檢查電腦 2 的檔案系統，以確認新檔案存在，例如 /home/end/myfile.txt。

使用指令行傳送檔案

在此區段中，透過向您顯示如何使用指令行來起始傳送，開始示範檔案傳送的彈性。雖然超出此實務範例的範圍，但您可以根據這些原則及 Ant Scripting 技術來定義及實作更強大的檔案傳送實務範例。

程序

- 從先前的示範中刪除已傳送的檔案，例如 /home/end/myfile.txt。
- 以使用者 fteuser 身分輸入 **fteCreateTransfer** 指令，以起始將檔案從電腦 1 上的 C:\start\myfile.txt 傳送至電腦 2 上的 /home/end/myfile.txt：

```
fteCreateTransfer -sa AGENT1 -sm QM1 -da AGENT3 -dm QM1 -w -dd "/home/end/"
"C:\start\myfile.txt"
```

- 在 IBM WebSphere MQ Explorer 中檢查檔案傳送日誌，或手動檢查檔案系統，以確認檔案傳送成功。

下一步

如需 IBM WebSphere MQ 安全的詳細資料，請參閱 [安全](#)。

您可能想要透過外部媒體來探索檔案傳送功能的更多特性。

- 如需新增審核功能以提供受管理檔案傳送的受管理層面的詳細資料，請參閱第 151 頁的『[將審核功能新增至受管理檔案傳送](#)』。
- 如需下列相關詳細資料：
 - 觸發：當新檔案出現時移動它們。
 - 觸發：配置單一檔案的外觀，以起始多個檔案的傳送。

- 使用 Apache Ant 進行 Scripting 傳送。

請參閱 https://www.ibm.com/developerworks/websphere/library/techarticles/1003_phillips/1003_phillips.html。

- 您自己所需的存取模型，可能不同於此實務範例中使用的存取模型。如需進一步保護環境安全的最佳作法相關資訊，請參閱 https://www.ibm.com/developerworks/websphere/library/techarticles/0902_wyatt/0902_wyatt.html。

將審核功能新增至受管理檔案傳送

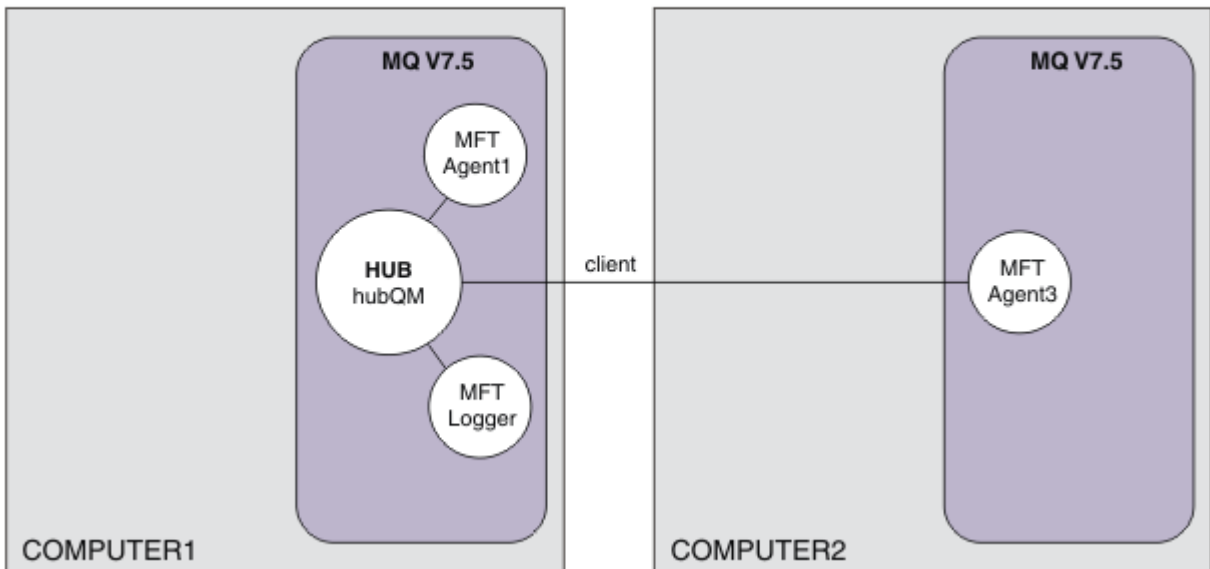
使用此實務範例來配置日誌程式，以及如何使用此功能來提供審核追蹤。

您已透過兩個先前的實務範例 第 132 頁的『詳細的基本檔案傳送』及第 144 頁的『詳細傳送兩部電腦檔案』，建置受管理檔案傳送功能的示範，並且您熟悉配置檔案傳送拓撲。您可以查看 IBM WebSphere MQ Version 7.5 如何提供記載及審核檔案傳送的功能，從而提供「受管理檔案傳送」的受管理層面。

假設您已根據 第 144 頁的『詳細傳送兩部電腦檔案』實務範例中的指示完成第二個受管理檔案傳送實務範例。

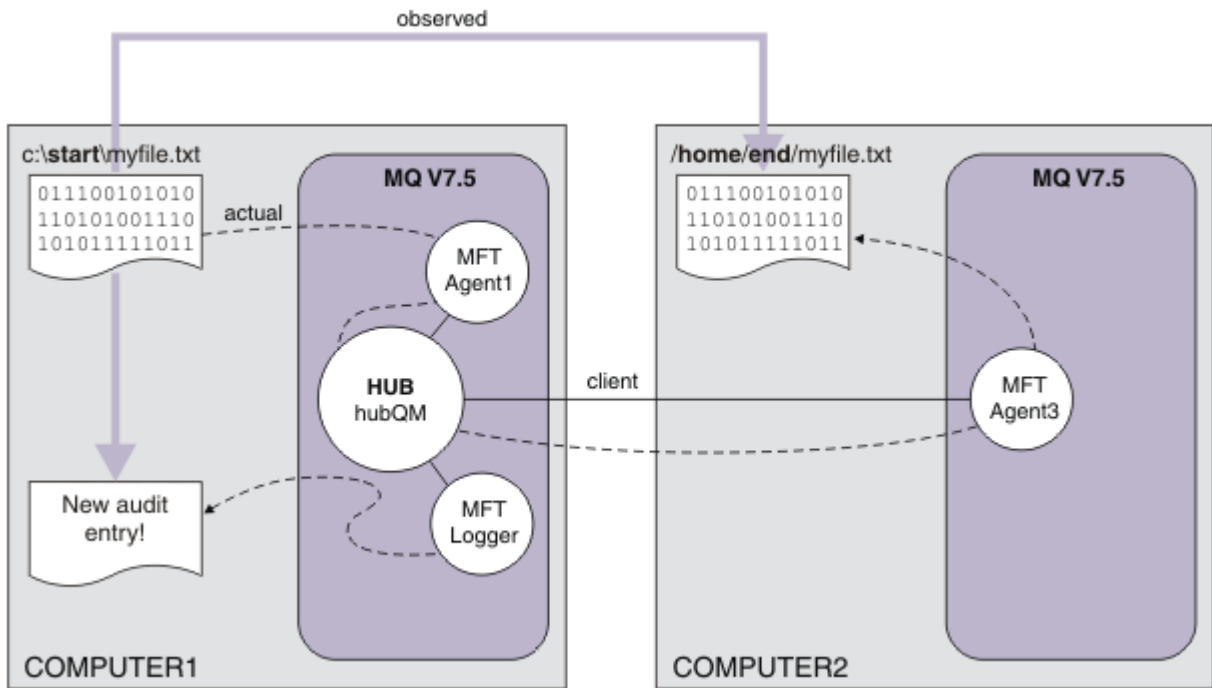
概觀

在此實務範例中，您繼續使用您在 第 144 頁的『詳細傳送兩部電腦檔案』中配置的現有 Windows 及 Linux 拓撲，並啟用檔案記載功能。檔案記載功能不需要安裝任何其他元件或產品，因此此實務範例不需要您考量兩個電腦檔案傳送實務範例的授權或授權變更。



您也可以實作資料庫日誌程式，這在正式作業環境中可能更適合，例如，用於可調整性及失效接手。不過，若要讓此實務範例保持簡單，且不需要安裝其他產品，請使用完整支援的檔案日誌程式。不會考量進一步的安全層面。因為此功能可能提供審核資訊，所以您可能想要考慮加強此安全。

在此實務範例中，會起始檔案傳送，並在日誌檔中擷取詳細資料。



規劃解決方案

說明 IBM WebSphere MQ 將審核功能新增至受管理檔案傳送實務範例的必要條件及授權需求。

在此實務範例中，您建立日誌程式以審核檔案傳送。日誌程式會尋找大部分與代理程式處理程序的類似之處，因此您可以使用現有的 `ftagent` 使用者來啟動及停止日誌程式。您可能想要建立自己的使用者或群組來管理日誌程式。

必備項目及授權

您需要來自 [第 144 頁的『詳細傳送兩部電腦檔案』](#) 實務範例的工作配置。

實作解決方案

修改電腦 1 以配置 IBM WebSphere MQ 的檔案日誌程式，將審核功能新增至受管理檔案傳送實務範例。

程序

1. 執行日誌程式處理程序時，請檢查 `ftagent` 群組是否具有 IBM WebSphere MQ 物件的適當存取權。輸入下列指令：

```
setmqaut -m hubQM -n "SYSTEM.FTE.LOG.RJCT.MYFILELOGGER" -t q -g FTAGENTS +put
setmqaut -m hubQM -n "SYSTEM.FTE.LOG.CMD.MYFILELOGGER" -t q -g FTAGENTS +get
```

如需詳細資料，請參閱 [資料庫日誌程式的權限](#)。

2. 以 IBM WebSphere MQ 管理者身分建立檔案日誌程式，並使用 `hubQM` 作為日誌程式佇列管理程式。

```
fteCreateLogger -loggerType FILE -loggerQMgr hubQM -fileLoggerMode LINEAR -fileSize 5MB
myFileLogger
runmqsc hubQM < <MYFILELOGGER_create.mqsc>
```

如需詳細資料，請參閱 [fteCreate 日誌程式 \(建立 WebSphere MQ Managed File Transfer 日誌程式\)](#)。您可能想要考量在正式作業中使用資料庫日誌程式。

3. 以使用者 `ftagent` 身分，啟動日誌程式。


```
fteStartLogger MYFILELOGGER
```

4. 確認日誌程式已啟動。

若要確認日誌程式已啟動，請檢查檔案系統。使用上述指令配置檔案日誌程式之後，可以在 <MQ INSTALL>/mqft/logs/hubQM/loggers/MYFILELOGGER/logs 中找到日誌。確認 output0.log 包含訊息 "BFGDB0023I: The logger has completed startup activities and is now running."。

5. 從先前的示範中刪除已傳送的檔案，例如 /home/end/myfile.txt。

6. 以使用者 ftuser 身分，從電腦 1 使用 fteCreateTransfer (建立新的檔案傳送) 指令，以開始將檔案從電腦 1 上的 C:\start\myfile.txt 傳送至電腦 2 上的 /home/end/myfile.txt。執行下列指令：

```
fteCreateTransfer -sa AGENT1 -sm hubQM -da AGENT3 -dm hubQM -w -dd "/home/end/"
"C:\start\myfile.txt"
```

7. 確認日誌程式擷取此傳送，並瞭解日誌項目的內容。

a. 開啟檔案 <MQ INSTALL>/mqft/logs/hubQM/loggers/MYFILELOGGER/MYFILELOGGER-XXXXXXXXX.log

b. 日誌項目會顯示您所起始的傳送，包括檔案來源和目的地位置，以及日期、時間和要求者 ID。例如：

```
2012-03-23T16:42:21;414d5120514d312020202020202020207a556b4f2000aa03;[TSTR]; ;
AGENT1;hubQM;STANDARD;AGENT3;hubQM;User;;;com.ibm.wmqfte.SourceAgent=AGENT1,
com.ibm.wmqfte.DestinationAgent=AGENT3, com.ibm.wmqfte.MqmdUser=User,
com.ibm.wmqfte.OriginatingUser=User, com.ibm.wmqfte.OriginatingHost=
dhcp-9-10-123-123.hursley.ibm.com., com.ibm.wmqfte.TransferId=
414d5120514d312020202020202020207a556b4f2000aa03, com.ibm.wmqfte.Priority=0;

2012-03-23T16:42:21;414d5120514d312020202020202020207a556b4f2000aa03;[TPRO];0 ;
C:\start\myfile.txt;51447;file;leave ;;;;/home/end/myfile.txt;51447;file;
;;;;;;

2012-03-23T16:42:21;414d5120514d312020202020202020207a556b4f2000aa03;[TCOM];0 ;
AGENT1;hubQM;STANDARD;AGENT3;hubQM;STANDARD;User;;BFGRP0032I: The file transfer
request has successfully completed.;com.ibm.wmqfte.SourceAgent=AGENT1,
com.ibm.wmqfte.DestinationAgent=AGENT3, com.ibm.wmqfte.MqmdUser=User,
com.ibm.wmqfte.OriginatingUser=User, com.ibm.wmqfte.OriginatingHost=
dhcp-9-20-123-123.hursley.ibm.com.,
com.ibm.wmqfte.TransferId=414d5120514d312020202020202020207a556b4f2000aa03,
com.ibm.wmqfte.Priority=0;
```

名詞解釋

此名詞解釋包含 IBM WebSphere MQ 的術語及定義。

本名詞解釋中會使用下列交互參照：

- 請參閱可讓您從術語參照偏好的同義字，或從字首語或縮寫參照已定義的完整形式。
- 另請參閱讓您參照相關或對照術語。

第 153 頁的『A』 第 156 頁的『B』 第 157 頁的『C』 第 161 頁的『D』 第 163 頁的『E』 第 163 頁的『F』 第 165 頁的『G』 第 165 頁的『H』 第 166 頁的『I』 第 167 頁的『J』 第 168 頁的『K』 第 168 頁的『L』 第 170 頁的『M』 第 173 頁的『N』 第 174 頁的『O』 第 175 頁的『P』 第 177 頁的『Q』 第 178 頁的『R』 第 181 頁的『S』 第 185 頁的『T』 第 187 頁的『U』 第 188 頁的『V』 第 188 頁的『W』 第 189 頁的『X』

A

異常終止原因碼 (abend reason code)

4 位元組十六進位碼，可唯一識別在 z/OS 作業系統上執行之程式的問題。

抽象類別 (abstract class)

在物件導向程式設計中，這是用來代表概念的類別；從其衍生的類別，代表概念的實作。無法從抽象類別建構物件；亦即，無法將其實例化。另請參閱母類別 (parent class)。

存取控制

在電腦安全中，這是用來確保使用者只能存取他們已獲授權之電腦系統資源的程序。

存取控制清單 (access control list, ACL)

在電腦安全中，這是一份關聯於某個物件的清單，用來識別所有能夠存取此物件的主體及其存取權。

責任

對自己的行為負責的品質。

ACL

請參閱存取控制清單 (access control list)。

作用中日誌 (active log)

這是一個具有固定大小的資料集，當發生回復事件時，會將回復事件記錄在該資料集中。當作用中日誌已滿時，會將作用中日誌的內容複製到保存日誌。

作用中佇列管理程式實例 (active queue manager instance)

這是一個用於執行多重實例佇列管理程式（正在處理要求）的實例。只有一個多重實例佇列管理程式的作用中實例。

配接器 (adapter)

這是供兩個其他軟體元件互相通訊的中介軟體元件。

位址空間 (address space, ASID)

這是適用於電腦程式或程序的位址範圍。位址空間可以指的是實體儲存體及/或虛擬儲存體。另請參閱聯合位址空間 (allied address space) 和緩衝池 (buffer pool)。

管理工具袋 (administration bag)

在「WebSphere MQ 管理介面 (MQAI)」中，這是為了管理 WebSphere MQ 而建立的一種資料工具袋類型，暗示它可以變更資料項目的順序、建立清單，以及檢查訊息內的選取器。

管理主題物件 (administrative topic object)

此物件能讓您對主題指派特定且非預設的屬性。

管理者指令 (administrator command)

用來管理 WebSphere MQ 物件 (例如佇列、處理程序及名稱清單) 的指令。

進階程式對程式通訊 (Advanced Program-to-Program Communication, APPC)

這是 SNA LU 6.2 通訊協定的實作，可讓交互連接的系統傳遞及共用程式的處理程序。

進階遙測用戶端

請參閱遙測進階用戶端。

親緣性 (affinity)

這是彼此具有某種關係或相依關係的物件之間的關聯。

警示

這是指出生成或即將發生某事件的訊息或其他指示。

警示監視器 (alert monitor)

在 WebSphere MQ for z/OS 中，這是 CICS 配接器的一個元件，用於處理由於對 WebSphere MQ for z/OS 的連線要求而發生的未排程事件。

別名佇列 (alias queue)

WebSphere MQ 物件，其名稱是定義給本端佇列管理程式之基本佇列或主題的別名。當應用程式或佇列管理程式使用別名佇列時，便會解析此別名，並且在相關聯的基本物件上執行所要求的作業。

別名佇列物件 (alias queue object)

WebSphere MQ 物件，其名稱是定義給本端佇列管理程式之基本佇列的別名。當應用程式或佇列管理程式使用別名佇列時，會解析該別名，並且會在相關聯的基本佇列上執行所要求的作業。

聯合位址空間 (allied address space)

連接至 WebSphere MQ for z/OS 的 z/OS 位址空間。

聯合 (ally)

請參閱聯合位址空間 (allied address space)。

替代使用者權限 (alternate user authority)

這是使用者 ID 能夠提供其他使用者 ID 進行安全檢查的能力。當應用程式開啟 WebSphere MQ 物件時，它可以在 MQOPEN、MQPUT1 或 MQSUB 呼叫上提供佇列管理程式用於權限檢查的使用者 ID，而不是與應用程式相關聯的使用者 ID。

替代使用者安全性 (alternate user security)

在 z/OS 上，當開啟 WebSphere MQ 物件時，當應用程式要求替代使用者權限時所執行的權限檢查。

APAR

請參閱[授權程式分析報告 \(authorized program analysis report\)](#)。

APF

請參閱[授權程式機能 \(authorized program facility\)](#)。

API 交叉結束程式 (API-crossing exit)

這是使用者撰寫的程式，在概念上與 API 結束程式類似。只有 WebSphere MQ for z/OS 上的 CICS 應用程式才支援它。

API 結束程式 (API exit)

這是使用者撰寫的程式，用於監視或修改 MQI 呼叫的功能。對於應用程式發出的每一個 MQI 呼叫，會在佇列管理程式開始處理呼叫之前，以及在佇列管理程式完成處理呼叫之後，呼叫 API 結束程式。API 結束程式可以檢查和修改 MQI 呼叫中的所有參數。

APPC

請參閱[進階程式對程式通訊 \(Advanced Program-to-Program Communication\)](#)。

應用程式定義的格式 (application-defined format)

這是訊息中的應用程式資料，使用者應用程式會定義其意義。另請參閱[內建格式 \(built-in format\)](#)。

應用程式環境 (application environment)

這是包括軟體及支援該軟體之伺服器或網路基礎架構的環境。

應用程式層次安全 (application level security)

當應用程式發出 MQI 呼叫時所呼叫的安全服務。

應用程式日誌 (application log)

在 Windows 系統中，這是記錄重要應用程式事件的日誌。

應用程式佇列 (application queue)

本端佇列，當它已設定觸發，且符合觸發條件時，需要寫入觸發訊息。

保存日誌 (archive log)

儲存裝置上的資料集，當作用中日誌達到其大小限制時，WebSphere MQ 會將每一個作用中日誌資料集的內容複製到該資料集。另請參閱[回復日誌 \(recovery log\)](#)。

ARM

請參閱[自動重新啟動管理程式 \(automatic restart manager\)](#)。

ASID

請參閱 [位址空間 \(address space\)](#)。

非對稱金鑰加密法 (asymmetric key cryptography)

這是使用兩個金鑰的加密系統：每個人都知道的公開金鑰，以及只有訊息接收端或傳送端知道的私密金鑰。另請參閱[對稱金鑰加密法 \(symmetric key cryptography\)](#)。

非同步使用 (asynchronous consumption)

這是一個使用 MQI 呼叫集的程序，可讓應用程式使用佇列集的訊息。透過呼叫應用程式所識別的程式碼單元，傳遞訊息或代表訊息的記號，將訊息遞送至應用程式。

非同步傳訊 (asynchronous messaging)

這是程式之間的一種通訊方法，程式先將訊息放到訊息佇列之後，便繼續自己的處理程序，不等待訊息的回覆。另請參閱[同步傳訊 \(synchronous messaging\)](#)。

非同步放置 (asynchronous put)

這是指由應用程式放置訊息，而不等候佇列管理程式的回應。

屬性 (attribute)

1. 在物件導向程式設計中，指物件或類別的內容，可與任何其他內容明確區分。屬性通常說明狀態資訊。
2. 說明實體的實體性質或特性；例如，員工的電話號碼是其中一個員工屬性。另請參閱[實體](#)。

鑑別 (authentication)

這是一種安全服務，負責證明電腦系統的使用者確實是此人所宣稱者。一般實作這項服務的機制是密碼和數位簽章。

鑑別資訊物件

這是一項物件，可提供在使用 LDAP 伺服器來檢查憑證撤銷清冊 (CRL) 時所需的定義，以便支援 Secure Sockets Layer (SSL) 安全。

權限檢查 (authority check)

請參閱[授權檢查 \(authorization check\)](#)。

授權 (authorization)

授與使用者、系統或程序對物件、資源或功能的完整或受限存取權的程序。

授權檢查 (authorization check)

當使用者或應用程式嘗試存取系統資源時所執行的安全檢查；例如，當管理者嘗試發出指令來管理 WebSphere MQ 時，或當應用程式嘗試連接至佇列管理程式時。

授權檔案 (authorization file)

這是一個用於提供物件、物件類別或物件的所有類別定義的檔案。

授權服務 (authorization service)

在 UNIX 及 Linux 系統上的 WebSphere MQ 及 WebSphere MQ for Windows 中，這是一項服務，可針對與指令或呼叫相關聯的使用者 ID，提供指令及 MQI 呼叫的權限檢查。

授權程式分析報告 (authorized program analysis report, APAR)

這是由 IBM 提供，且受支援之程式版本中的某個問題所提出的更正要求。

授權程式機能 (authorized program facility, APF)

在 z/OS 環境中，允許識別獲授權使用受限功能之程式的機能。

自動重新啟動管理程式 (automatic restart manager, ARM)

這是一種 z/OS 回復功能，在批次工作及已啟動的作業或執行這些工作和作業的系統非預期地結束之後，可以自動重新啟動這些工作及作業。

B

取消 (backout)

這是一項作業，可撤銷在現行工作單位期間對資源所做的所有變更。另請參閱[確定 \(commit\)](#)。

工具袋 (bag)

請參閱[資料工具袋 \(data bag\)](#)。

分隔線 (bar)

z/OS 記憶體限制，在 64 位元系統中設定為 2GB。長條會將低於 2 GB 位址的儲存體與高於 2 GB 位址的儲存體隔開。在分隔線上方的區域是用於資料；沒有程式會執行於分隔線上方。

基本對映支援 (basic mapping support, BMS)

CICS 與應用程式之間的介面，可格式化輸入及輸出顯示資料，並遞送多頁輸出訊息，而不考慮各種終端機所使用的控制字元。

行為 (behavior)

在物件導向程式設計中，這是體現在方法中的功能。

BMS

請參閱[基本對映支援 \(basic mapping support\)](#)。

Booch 方法論 (Booch methodology)

這是一種物件導向的方法論，用來協助使用者利用物件導向的參照範例設計系統。

Bootstrap 資料集 (bootstrap data set, BSDS)

VSAM 資料集，包含 WebSphere MQ for z/OS 已知的所有作用中及保存日誌資料集的庫存，以及所有最近 WebSphere MQ for z/OS 活動的四周庫存。需要 BSDS 才能重新啟動 WebSphere MQ for z/OS 子系統。

瀏覽

在訊息佇列作業中，用來複製訊息，而無需將其從佇列中移除。另請參閱[取得 \(get\)](#)、[放置 \(put\)](#)。

瀏覽游標 (browse cursor)

在訊息佇列作業中，這是瀏覽佇列時所使用的指示器，用來識別下一個訊息（依順序）。

BSDS

請參閱[引導資料集 \(bootstrap data set\)](#)。

緩衝池 (buffer pool)

這是記憶體區域，在其中讀取資料頁面，並在處理期間修改並保存它們。另請參閱[位址空間 \(address space\)](#)。

內建格式 (built-in format)

這是訊息中的應用程式資料，佇列管理程式會定義其意義。另請參閱[應用程式定義的格式 \(application-defined format\)](#)。

C**CA**

請參閱[憑證管理中心 \(certificate authority\)](#)。

CAF

請參閱[用戶端連接特性](#)。

回呼 (callback)

這是訊息消費者或事件處理程式的常式。

CCDT

請參閱[用戶端通道定義表 \(client channel definition table\)](#)。

CCF

請參閱[通道控制函數 \(channel control function\)](#)。

CCSID

請參閱[編碼字集 ID \(coded character set identifier\)](#)。

CDF

請參閱[通道定義檔 \(channel definition file\)](#)。

憑證管理中心 (certificate authority, CA)

授信協力廠商組織或公司，發出數位憑證以回應憑證簽署要求。憑證管理中心會驗證獲授與唯一憑證之個人的身分。另請參閱[Secure Sockets Layer](#)。

憑證鏈 (certificate chain)

在加密方面彼此相關的憑證階層，此階層起始於個人憑證，結束於鏈結頂端的主要憑證。

憑證有效期限 (certificate expiration)

包含憑證有效日期範圍的數位憑證。如果超出此有效日期範圍，即會將憑證視為已過期。

憑證撤消清冊 (certificate revocation list, CRL)

這是一份在排定到期日之前，已遭撤消的憑證清單。憑證撤消清冊由憑證管理中心維護，在 Secure Sockets Layer (SSL) 信號交換期間，用來確保涉及的憑證尚未撤消。

憑證儲存庫 (certificate store)

金鑰儲存庫的 Windows 名稱。

憑證簽署要求 (certificate signing request, CSR)

包含公程式或組織的公開金鑰和主體識別名稱的要求。傳送至 CA，以便 CA 向該公程式發出數位簽章。

CF

請參閱[連結機能 \(coupling facility\)](#)。

CFSTRUCT

用來說明佇列管理程式使用「連結機能」清單結構的 WebSphere MQ 物件

channel

這是一種 WebSphere MQ 物件，可定義兩個佇列管理程式之間的通訊鏈結（訊息通道）或用戶端與佇列管理程式之間的通訊鏈結（MQI 通道）。另請參閱[訊息通道 \(message channel\)](#)、[MQI 通道 \(MQI channel\)](#)。

通道回呼 (channel callback)

這是可確保對正確的機器建立通道連線的一種機制。在通道回呼中，傳送端通道會使用傳送端的定義來回呼原始要求端通道。

通道控制函數 (channel control function, CCF)

這是一個程式，透過操作員面板介面（容許設定和控制通道），將訊息從傳輸佇列移至通訊鏈結，並從通訊鏈結移至本端佇列。

通道定義檔 (channel definition file, CDF)

這是一個包含通訊通道定義的檔案，通訊通道定義可將傳輸佇列與通訊鏈結相關聯。

通道事件 (channel event)

這是在通道作業期間偵測到的事件報告狀況，例如，當通道實例啟動或停止時。通道事件產生於通道兩端的佇列管理程式上。

通道結束程式 (channel exit program)

這是使用者撰寫的程式，會從某個已定義位置號碼呼叫該程式，此號碼位於訊息通道代理程式 (MCA) 的處理順序中。

通道起始程式 (channel initiator)

WebSphere MQ 分散式佇列的元件，用來監視起始佇列，以查看何時符合觸發準則，然後啟動傳送端通道。

通道接聽器 (channel listener)

WebSphere MQ 分散式佇列作業的元件，用來監視網路是否有啟動要求，然後啟動接收端通道。

檢查點 (checkpoint)

這是在程式中執行檢查的位置，或是記錄資料的位置，以容許程式在岔斷的情況下重新啟動。

CI

請參閱控制間隔 ([control interval](#))。

CipherSpec

這是在鑑別完成之後，套用至 SSL 訊息之加密演算法和雜湊函數的組合。

密碼組合 (cipher suite)

這是鑑別、金鑰交換演算法及 Secure Sockets Layer (SSL) 密碼規格的組合，用於資料的安全交換。

密文 (ciphertext)

已加密的資料。除非使用金鑰將密文轉換成純文字 (解密)，否則無法讀取密文。另請參閱明碼 ([cleartext](#))。

循環式記載 (circular logging)

在 UNIX 及 Linux 系統上的 WebSphere MQ 及 WebSphere MQ for Windows 中，將所有重新啟動資料保留在日誌檔環中的處理程序。另請參閱線性記載 ([linear logging](#))。

CL

請參閱指令語言 ([Command Language](#))。

類別

在物件導向設計或程式設計中，這是可用來建立具有共同定義和共同內容、作業以及行為之物件的模型或範本。物件就是類別的實例。

類別階層 (class hierarchy)

這是共用單一繼承之類別間的關係。

類別庫 (class library)

在物件導向程式設計中，這是預先撰寫之類別或程式碼範本的集合，程式設計師可在開發應用程式時指定和使用其中任何一個。

明碼 (cleartext)

以可讀取形式透過網路傳送的一串字元。它們可以基於壓縮目的進行編碼，但可以輕鬆地解碼。另請參閱密文 ([ciphertext](#))。

用戶端

一種執行時期元件，可為本端使用者應用程式提供對伺服器上佇列服務的存取權。應用程式所使用的佇列位於伺服器上。另請參閱 [WebSphere MQ MQI 用戶端](#)、[WebSphere MQ Java 用戶端](#)、[WebSphere MQ 完全受管理 .NET 用戶端](#)。

用戶端應用程式 (client application)

這是執行於工作站、鏈結至用戶端，可供應用程式存取伺服器上佇列之服務的應用程式。

用戶端連接特性 (client attachment feature, CAF)

支援將用戶端連接至 z/OS 的選項。

用戶端通道定義表 (client channel definition table, CCDT)

這是包含一個以上用戶端連線通道定義的檔案。

用戶端連線通道類型 (client-connection channel type)

與 WebSphere MQ 用戶端相關聯的 MQI 通道定義類型。另請參閱 [伺服器連線通道類型 \(server-connection channel type\)](#)。

CLUSRCVR

請參閱 [叢集接收端通道 \(cluster-receiver channel\)](#)。

CLUSSDR

請參閱 [叢集傳送端通道 \(cluster-sender channel\)](#)。

叢集

在 WebSphere MQ 中，這是一部以上電腦上由兩個以上佇列管理程式所組成的群組，可提供自動互連，並容許在它們之間通告佇列及主題，以進行負載平衡及備援。

叢集佇列 (cluster queue)

這是由叢集佇列管理程式所管理的本端佇列，並已定義為從連線至叢集中任何佇列管理程式之應用程式所放置訊息的目標。所有擷取訊息的應用程式都必須在本端連接。

叢集佇列管理程式 (cluster queue manager)

佇列管理程式是叢集的一個成員。佇列管理程式可以是多個叢集的成員。

叢集接收端通道 (cluster-receiver channel, CLUSRCVR)

在這個通道中，叢集佇列管理程式可接收來自叢集中其他佇列管理程式的訊息，以及來自儲存庫佇列管理程式的叢集資訊。

叢集傳送端通道 (cluster-sender channel, CLUSSDR)

這是一個通道，在其中，叢集佇列管理程式可將訊息傳送至叢集中的其他佇列管理程式，並將叢集資訊傳送至儲存庫佇列管理程式。

叢集主題 (cluster topic)

這是一個管理主題，定義於叢集佇列管理程式上，並供叢集中其他佇列管理程式使用。

叢集傳輸佇列 (cluster transmission queue)

該傳輸佇列用來保存在相同叢集中兩個佇列管理程式之間流通的所有訊息。該佇列稱為 SYSTEM.CLUSTER.TRANSMIT.QUEUE。

CMS 金鑰資料庫 (CMS key database)

CMS 金鑰資料庫是 Windows 系統、UNIX 系統、Linux 及那些平台的用戶端所支援的資料庫格式。以 .kdb 結尾的檔案是 CMS 格式。 .kdb 檔案包含憑證和金鑰。

編碼字集 ID (coded character set identifier, CCSID)

這是一種 16 位元數字，其中包含一組特定編碼方法 ID、字集 ID、字碼頁 ID，以及用於唯一識別編碼圖形字元表示法的其他資訊。

共存性 (coexistence)

兩個以上不同版本 WebSphere MQ 在相同電腦上運作的能力。

指令

這是用來起始動作或啟動服務的陳述式。指令由指令名稱縮寫及其參數和旗標（如果適用）組成。

指令工具袋 (command bag)

在 MQAI 中，這是為了管理 WebSphere MQ 物件而建立的工具袋類型，但無法在訊息內變更資料項目的順序或建立清單。

指令事件 (command event)

已順利執行 MQSC 或 PCF 指令的通知。

指令語言 (command language, CL)

在 WebSphere MQ for iSeries 中，可在指令行或撰寫 CL 程式時用來發出指令的語言。

指令字首

1. 1 到 8 個字元的指令 ID。指令字首會將指令識別為屬於應用程式或子系統，而不是屬於 z/OS。
2. 在 WebSphere MQ for z/OS 中，這是一個字串，用來識別 WebSphere MQ for z/OS 指令所指向的佇列管理程式，以及從中接收 WebSphere MQ for z/OS 操作員訊息的佇列管理程式。

指令伺服器 (command server)

從系統指令輸入佇列讀取指令、驗證指令並將有效指令傳遞至指令處理器的 WebSphere MQ 元件。

確定

套用現行回復單元 (UR) 或工作單元 (UOW) 期間進行的所有變更。作業完成之後，便可開始新的 UR 或 UOW。

通用名稱 (common name, CN)

X.509 憑證的「識別名稱 (DN)」屬性中的元件，可代表通常與憑證擁有者相關聯的名稱。對於人而言，CN 通常是其實際名稱。對於 Web 伺服器，CN 是伺服器的完整主機及網域名稱。對於 WebSphere MQ，此欄位沒有特定需求，不過有許多管理者使用佇列管理程式的名稱。

另請參閱 [識別名稱](#)

完成碼 (completion code)

這是一種回覆碼，用來指出訊息佇列介面 (MQI) 呼叫的結束方式。

機密性 (confidentiality)

這是用來防止未獲授權而揭露機密性資訊的安全服務。加密是用來實作此服務的一般機制。

配置事件 (configuration event)

這是關於物件屬性的通知。這些通知會在建立、變更或刪除物件時產生，也會在明確要求時產生。

連線親緣性 (connection affinity)

這是一種通道屬性，在有多個連線可供使用時，指定用戶端應用程式用來連接佇列管理程式的用戶端通道定義。

Connection Factory

這是一組配置值，用來產生可讓 Java EE 元件存取資源的連線。Connection Factory 提供從應用程式至企業資訊系統 (EIS) 的隨選連線，可讓應用程式伺服器在分散式交易中登記 EIS。

連線控點 (connection handle)

這是一種 ID 或記號，程式透過它來存取所連接的佇列管理程式。

建構子 (constructor)

在物件導向程式設計中，這是用來起始設定物件的特殊方法。

使用 (consume)

從佇列中移除訊息，並將該訊息的內容傳回給呼叫應用程式。

消費者

這是接收和處理訊息的一種應用程式。另請參閱 [訊息消費者 \(message consumer\)](#)。

環境定義安全 (context security)

在 z/OS 上，當應用程式開啟佇列並指定它將在放入佇列的訊息中設定環境定義，或將環境定義從收到的訊息傳遞至放入佇列的訊息時所執行的權限檢查。

控制指令 (control command)

在 UNIX 及 Linux 系統上的 WebSphere MQ 及 WebSphere MQ for Windows 中，可以從作業系統指令行以互動方式輸入的指令。這類指令只需要安裝 WebSphere MQ 產品；它不需要特殊公用程式或程式來執行它。

控制間隔 (control interval, CI)

這是固定長度的直接存取儲存體區域，VSAM 會在該區域中儲存記錄並建立分散式可用空間。控制間隔是 VSAM 與直接存取儲存體之間傳輸的資訊單元。控制間隔一律包括整數數目的實體記錄。

控制關機 (controlled shutdown)

請參閱 [靜止關機 \(quiesced shutdown\)](#)。

相互關係 ID

這是訊息中的一個欄位，能提供識別相關訊息的方法。例如，會使用相關性 ID，來比對要求訊息和其對應的回覆訊息。

連結機能 (coupling facility, CF)

這是特殊的邏輯分割區，用來在 Sysplex 中提供高速快取、清單處理及鎖定功能。

CPF

請參閱 [指令字首 \(command prefix\)](#)。

CR (憑證申請)

憑證簽署要求的同義字。

CRL

請參閱 [憑證撤消清冊 \(certificate revocation list\)](#)。

跨系統連結機能 (XCF) (cross-system coupling facility, XCF)

提供功能以支援在 Sysplex 內執行的授權程式之間進行合作的元件。

加密法 (cryptography)

透過將資訊轉換 (加密) 為無法讀取的格式 (稱為密文) 來保護資訊。只有擁有秘密金鑰的人才能將訊息解密為純文字。

D**DAE**

請參閱傾出分析及排除 ([dump analysis and elimination](#))。

常駐程式 (daemon)

這是會連續或定期自動執行功能 (如網路控制) 的程式。

資料工具袋 (data bag)

這是物件內容的儲存器，供 MQAI 在管理佇列管理程式時使用。有 3 種類型的資料工具袋：使用者 (適用於使用者資料)、管理 (適用於具有假定選項的管理) 和指令 (適用於不具有假定選項的管理)。

資料轉換介面 (data-conversion interface, DCI)

WebSphere MQ 介面，在不同機器編碼和 CCSID 之間轉換應用程式資料的客戶或供應商撰寫程式必須符合該介面。WebSphere MQ 架構的一部分。

資料轉換服務 (data-conversion service)

該服務用來將應用程式資料轉換成字集及編碼，應用程式在其他平台上需要這些字集及編碼。

資料包 (datagram)

這是一種應用程式會傳送訊息，但不要求回應的非同步傳訊形式。另請參閱[要求/回覆 \(request/reply\)](#)。

資料完整性 (data integrity)

這是安全服務，用來偵測是否存在未獲授權的資料修改或竄改。該服務只偵測資料是否已修改；如果資料已修改，則它不會將資料還原為其原始狀態。

資料項目 (data item)

在 MQAI 中，這是資料工具袋中包含的項目。這可以是整數項目或字串項目，以及使用者項目或系統項目。

DCE

請參閱分散式運算環境 ([Distributed Computing Environment](#))。

DCE 主體 (DCE principal)

這是使用分散式運算環境的使用者 ID。

DCI

請參閱[資料轉換介面 \(data-conversion interface\)](#)。

DCM

請參閱數位憑證管理程式 ([Digital Certificate Manager](#))。

無法傳送郵件的佇列 (dead-letter queue, DLQ)

這是一種佇列，佇列管理程式或應用程式會將無法遞送至其正確目的地的訊息傳送到其中。

無法傳送郵件的佇列處理程式 (dead-letter queue handler)

這是一項公用程式，用來監視無法傳送郵件的佇列 (DLQ)，並根據使用者撰寫的規則表格來處理佇列中的訊息。WebSphere MQ 提供了無法傳送郵件的佇列處理程式範例。

解密 (decryption)

這是將已加密成秘密格式之資料解密的程序。解密需要秘密金鑰或密碼。

預設物件 (default object)

這是一個具有所有已定義屬性的物件定義 (例如，佇列)。如果使用者定義了物件，但未指定該物件所有可能的屬性，則佇列管理程式會使用預設屬性來替代未指定的任何屬性。

延遲連線 (deferred connection)

CICS 子系統在啟動之前嘗試連接至 WebSphere MQ for z/OS 時啟動的擱置事件。

衍生 (derivation)

在物件導向程式設計中，這是指從一個類別修正或延伸另一個類別。

目的地 (destination)

1. 在 JMS 中，指定應該傳送及接收訊息的位置及方式的物件。
2. 訊息傳送至的端點，例如佇列或主題。

Diffie-Hellman 金鑰交換 (Diffie-Hellman key exchange)

用於透過非安全通道來安全地建立共用密碼的公用金鑰交換演算法。

數位憑證 (digital certificate)

這是用來識別個人、系統、伺服器、公司或其他實體的電子文件，它會將公開金鑰關聯於此實體。數位憑證是憑證管理中心所發出，由此管理中心提供數位簽章。

數位憑證管理程式 (digital certificate manager, DCM)

在 IBM i 系統上，這是在 iSeries 伺服器上的安全應用程式中管理數位憑證及使用數位憑證的方法。「數位憑證管理程式」要求並處理來自憑證權限 (CA) 或其他協力廠商的數位憑證。

數位簽章 (digital signature)

這是一種利用私密金鑰來加密的資訊，它附加到某訊息或物件上，以向收件人保證此訊息或物件的真實性與完整性。數位簽章保證這個訊息或物件是擁有（或有權存取）私密金鑰或共用機密對稱金鑰的實體所簽署。

斷線 (disconnect)

中斷應用程式與佇列管理程式之間的連線。

識別名稱 (DN)

一組名稱/值配對 (例如 CN=person name 和 C=country)，用於唯一識別數位憑證中的實體。請注意，「識別名稱」僅在給定憑證管理中心的名稱空間內是唯一的。具有相同識別名稱的憑證完全可能由不同的憑證管理中心發出。因此，請確定金鑰儲存庫所包含的授信主要 CA 憑證儘可能少，最好不超過一個。另請參閱 [憑證管理中心](#)、[數位憑證](#)、[X509](#)。

分散式應用程式 (distributed application)

在訊息佇列作業中，這是一組應用程式，其中每一個都可以連接到不同的佇列管理程式，但集合在一起又可以組成單一應用程式。

分散式運算環境 (DCE)

在網路運算中，這是一組服務和工具，這些服務和工具支援透過異質作業系統和網路，來建立、使用和維護分散式應用程式。

分散式佇列管理 (distributed queue management)

在訊息佇列作業中，設定及控制其他系統上佇列管理程式的訊息通道。

發佈清單 (distribution list)

可以使用單一陳述式將訊息放入其中的佇列清單。

DLQ

請參閱無法傳送郵件的佇列 (dead-letter queue)。

DN

請參閱識別名稱 (distinguished name)。

雙重記載 (dual logging)

一種記錄 WebSphere MQ for z/OS 活動的方法，其中每一項變更都記錄在兩個資料集上，因此如果需要重新啟動，且其中一個資料集無法讀取，則可以使用另一個資料集。另請參閱 [單一記載 \(single logging\)](#)。

雙重模式 (dual mode)

請參閱 [雙重記載 \(dual logging\)](#)。

傾出分析與排除 (dump analysis and elimination, DAE)

z/OS 服務，可讓安裝暫停不需要的 SVC 傾出和 ABEND SYSUDUMP 傾出，因為它們會複製先前寫入的傾出。

可延續訂閱 (durable subscription)

當訂閱應用程式與佇列管理程式的連線關閉時所保留的訂閱。當訂閱應用程式中斷時，可延續訂閱作業會維持在適當位置，發佈作業會繼續進行分送。當應用程式重新連線時，它可以指定唯一的訂閱名稱，便可使用相同的訂閱。另請參閱 [不可延續訂閱 \(nondurable subscription\)](#)。

動態佇列 (dynamic queue)

這是在程式開啟模型佇列物件時所建立的本端佇列。

E

竊聽 (eavesdropping)

這是指通訊安全的中斷，在該中斷中，資訊保持完整，但其隱私權已受損。另請參閱模擬 (impersonation)、竄改 (tampering)。

Eclipse

這是一種開放程式碼的開端，它提供一種標準平台，供獨立軟體廠商 (ISV) 及其他工具開發人員用來開發與外掛程式相容的應用程式開發工具。

封裝作業 (encapsulation)

在物件導向程式設計中，這是用來從用戶端程式隱藏物件固有的詳細資料、函數或類別的技術。

加密

在電腦安全中，這是將資料轉換成晦澀難懂格式的程序，如此便無法取得原始資料，或只能使用解密程序才能取得原始資料。

放入佇列 (enqueue)

這是指將訊息或項目放入佇列。

實體 (entity)

定義給安全服務 (例如 RACF) 的使用者、群組或資源

環境變數 (environment variable)

這是指定作業系統或另一個程式如何執行或作業系統所辨識之裝置的變數。

ESM

請參閱外部安全管理程式 (external security manager)。

ESTAE

請參閱延伸指定作業異常終止 (extended specify task abnormal exit)。

事件資料 (event data)

在事件訊息中，這是包含事件之相關資訊的訊息資料部分 (例如，可引起事件的佇列管理程式名稱及應用程式)。另請參閱事件標頭 (event header)。

事件標頭 (event header)

在事件訊息中，這是用來識別事件原因碼之事件類型的訊息資料部分。另請參閱事件資料 (event data)。

事件訊息 (event message)

此訊息包含與 WebSphere MQ 系統網路中檢測事件原點相關的資訊 (例如事件種類、導致事件的應用程式名稱及佇列管理程式統計資料)。

事件佇列 (event queue)

這是偵測到事件以後，佇列管理程式將事件訊息放入其中的佇列。每一個種類的事件 (佇列管理程式、效能、配置、設備或通道事件) 都具有它專屬的事件佇列。

事件檢視器 (Event Viewer)

Windows 系統提供用來檢查及管理日誌檔的工具。

異常狀況接聽器 (exception listener)

這是可以由應用程式所登錄之類別的實例，並且對它呼叫 onException() 方法，以非同步方式將 JMS 異常狀況傳遞至應用程式。

排除法 (exclusive method)

在物件導向程式設計中，該方法並不預期出現多型性；每一個都具有特定效果。

延伸指定作業異常終止結束程式 (extended specify task abnormal exit, ESTAE)

z/OS 巨集提供回復功能，並提供控制權給使用者指定的結束常式來處理、診斷異常終止或指定重試位址。

外部安全管理程式 (external security manager, ESM)

這是對使用者和資源執行安全檢查的一種安全性產品。RACF 就是 ESM 的一個範例。

F

失效接手 (failover)

在軟體、硬體或網路岔斷時切換至備援或待命系統的自動作業。

FAP

請參閱格式和通訊協定 (Formats and Protocols)。

FFDC

請參閱首次失敗資料擷取 (first-failure data capture)。

FFST

請參閱 首次失敗支援技術 (First Failure Support Technology)。

FFST 檔案

請參閱 First Failure Support Technology 檔案。

先進先出

請參閱先入先出 (first-in first-out)。

FIPS

美國聯邦資訊處理標準

首次失敗資料擷取 (first-failure data capture, FFDC)

1. 一種問題診斷輔助工具，可識別錯誤、收集並記載這些錯誤的相關資訊，並將控制權傳回給受影響的執行時期軟體。
2. FFST 架構的 IBM i 實作，提供問題識別、選擇性傾出診斷資料、症狀字串產生及問題日誌項目。

首次失敗支援技術 (First Failure Support Technology, FFST)

一種 IBM 架構，可定義透過防禦程式設計技術進行錯誤偵測的單一方法。這些技術提供主動式（直到需要時才是被動的）問題識別，以及對軟體問題除錯時所需之診斷輸出的說明。

首次失敗支援技術檔案 (FFST 檔案)

這是一種檔案，包含用於偵測和診斷軟體問題的資訊。在 WebSphere MQ 中，FFST 檔案的檔案類型為 FDC。

先進先出 (first-in-first-out, FIFO)

這是一種佇列作業技術，在該技術中，即將擷取的下一個項目，是已處於佇列中最長時間的項目。

強制關機 (forced shutdown)

CICS 配接器的一種關機類型，不論任何目前作用中作業的狀態為何，配接器都會立即與 WebSphere MQ for z/OS 中斷連線。另請參閱靜止關機 (quiesced shutdown)。

格式 (format)

這是訊息佇列作業中的術語，可以用來識別訊息中應用資料的性質。

格式與通訊協定 (Formats and Protocols, FAP)

在訊息佇列作業中，它會定義佇列管理程式如何彼此通訊，以及用戶端如何與伺服器佇列管理程式通訊。

架構 (Framework)

在 WebSphere MQ 中，這是程式設計介面的集合，可讓客戶或供應商撰寫程式來延伸或取代 WebSphere MQ 產品所提供的某些功能。這些介面包括：資料轉換介面 (DCI)、訊息通道介面 (MCI)、名稱服務介面 (NSI)、安全啟用介面 (SEI) 及觸發監視器介面 (TMI)。

夥伴類別 (friend class)

在該類別中，會授予所有成員函數對另一個類別之專用及受保護成員的存取權。會在另一個類別的宣告中將其命名，它會使用關鍵字 friend 作為類別字首。

FRR

請參閱功能性回復常式 (functional recovery routine)。

完整儲存庫 (full repository)

這是叢集中每一個佇列管理程式的完整相關資訊集。此資訊集稱為儲存庫，有時也稱為完整儲存庫，通常由叢集中的 2 個佇列管理程式保留。另請參閱局部儲存庫 (partial repository)。

函數 (function)

一組指名的陳述式，它們可以加以呼叫及評估，然後傳回一個值給進行呼叫的陳述式。

功能性回復常式 (functional recovery routine, FRR)

z/OS 回復及終止管理程式，可讓回復常式在程式岔斷時取得控制權。

G

閘道佇列管理程式 (gateway queue manager)

這是叢集佇列管理程式，用於將訊息從應用程式遞送至叢集中的其他佇列管理程式。

一般性追蹤機能 (generalized trace facility, GTF)

一種 z/OS 服務程式，可記錄重要系統事件，例如 I/O 岔斷、SVC 岔斷、程式岔斷及外部岔斷。

一般安全服務 API (Generic Security Services API)

請參閱一般安全服務應用程式設計介面 ([Generic Security Services application programming interface](#))。

一般安全服務應用程式設計介面 (Generic Security Services application programming interface, Generic Security Services API, GSS API)

這是一種共用應用程式設計介面 (API)，用於存取安全服務。

取得

在訊息佇列作業中，這是指使用 MQGET 呼叫來移除佇列中的訊息，並將其內容傳回至呼叫應用程式。另請參閱 [瀏覽 \(browse\)](#)、[放置 \(put\)](#)。

廣域定義物件 (globally defined object)

在 z/OS 上，這是其定義儲存在共用儲存庫中的物件。佇列共用群組中的所有佇列管理程式，都可使用該物件。另請參閱 [區域定義物件 \(locally defined object\)](#)。

廣域追蹤 (global trace)

WebSphere MQ for z/OS 追蹤選項，其中追蹤資料來自整個 WebSphere MQ for z/OS 子系統。

廣域交易 (global transaction)

這是在分散式交易環境中，由一個以上資源管理程式來執行，由外部交易管理程式來協調的可回復工作單元。

GSS API

請參閱一般安全服務應用程式設計介面 ([Generic Security Services application programming interface](#))。

GTF

請參閱 [一般性追蹤機能 \(generalized trace facility\)](#)。

H

信號交換 (handshake)

這是指在啟動 Secure Sockets Layer 階段作業時交換訊息，使用戶端能夠利用公開金鑰技術鑑別伺服器（以及選擇性地使伺服器鑑別用戶端），進而使用戶端和伺服器能夠合作建立用來加密、解密和偵測竄改內容的對稱金鑰。

強化訊息 (hardened message)

這是寫入輔助設備（磁碟）儲存體的訊息，以便在發生系統失效時，不會遺失該訊息。

標頭 (header)

請參閱 [訊息標頭 \(message header\)](#)。

活動訊號 (heartbeat)

這是實體彼此之間傳送的一種信號，表達本身仍在作用中。

活動訊號流 (heartbeat flow)

這是當沒有訊息要傳送時，從傳送訊息通道代理程式 (MCA) 傳遞至接收 MCA 的脈衝。該脈衝會解除封鎖接收 MCA，除非訊息已抵達，或斷線間隔已過期，否則該接收 MCA 將保持等待狀態。

活動訊號間隔 (heartbeat interval)

活動訊號流之間將經歷的時間（以秒為單位）。

階層結構 (hierarchy)

在發佈/訂閱傳拓撲中，這是指連接至母項佇列管理程式的本端佇列管理程式。

HTTP

請參閱 [超文字傳送通訊協定 \(Hypertext Transfer Protocol\)](#)。

超文字轉送通訊協定 (HTTP)

用於在 Web 上傳送及顯示超文字和 XML 文件的網際網路通訊協定。

I

識別環境定義

識別應用程式使用者的資訊，該應用程式會先將訊息放置在佇列上。

識別 (identification)

這是可容許唯一識別電腦系統中每一個使用者的安全服務。實作此服務的一般機制，是將使用者 ID 與每一個使用者相關聯。

身分環境定義 (identity context)

可識別先將訊息放置在佇列上之應用程式使用者的資訊。

IFCID

請參閱設備機能元件 ID ([instrumentation facility component identifier](#))。

ILE

請參閱 [整合語言環境](#)。

立即關閉 (immediate shutdown)

在 WebSphere MQ 中，這是關閉不會等待應用程式中斷連線的佇列管理程式。容許完成現行訊息佇列介面 (MQI) 呼叫，但在要求立即關閉之後，新的 MQI 呼叫會失敗。另請參閱 [強制關機 \(preemptive shutdown\)](#) 和 [靜止關機 \(quiesced shutdown\)](#)。

模擬 (impersonation)

這是一種通訊安全中斷，在其中，會將資訊遞送至充當預期接收者的人員，或由充當其他人的人員傳送資訊。另請參閱 [竊聽 \(eavesdropping\)](#) 和 [竄改 \(tampering\)](#)。

入埠通道 (inbound channel)

這是從另一個佇列管理程式接收訊息的通道。

內嵌格式 (in-built format)

請參閱 [內建格式 \(built-in format\)](#)。

索引 (index)

在「WebSphere MQ 管理介面 (MQAI)」中，這是一種參照資料項目的方法。

不確定的回復單元

已要求同步點但尚未確認的回復單元狀態。

進行中 (inflight)

這是尚未完成確定程序準備階段之資源或回復單元的狀態。

繼承

這是一種物件導向程式設計技術，可讓您利用現有的類別來作為建立其他類別的基礎。在繼承中，較明確的元素納入了較一般元素的結構和行為。

起始設定輸入資料集 (initialization input data set)

WebSphere MQ for z/OS 在啟動時使用的資料集。

起始佇列 (initiation queue)

這是佇列管理程式在其中放置觸發訊息的本端佇列。

起始字元/起始程式 (initiator)

在分散式佇列中，這是在另一個系統上要求網路連線的程式。另請參閱 [回應者 \(responder\)](#)。

輸入參數 (input parameter)

MQI 呼叫的參數，您可以在發出呼叫時在其中提供資訊。

插入項目順序 (insertion order)

在「WebSphere MQ 管理介面 (MQAI)」中，將資料項目放入資料工具袋的順序。

可安裝服務 (installable service)

在 UNIX 及 Linux 系統上的 WebSphere MQ 及 WebSphere MQ for Windows 中，作為獨立元件提供的附加功能。每一個元件的安裝都是選用項目：可轉為使用內部或協力廠商元件。

實例 (instance)

這是指屬於某類別之物件的特定呈現。另請參閱 [物件 \(object\)](#)。

實例資料 (instance data)

在物件導向程式設計中，這是與物件相關聯的狀態資訊。

設備事件 (instrumentation event)

在 WebSphere MQ 系統的網路中監視佇列管理程式資源定義、效能條件及通道條件的方法。

應用機能元件 ID (instrumentation facility component identifier, IFCID)

在 Db2 for z/OS 中，這是用來命名及識別事件追蹤記錄的值。在作為 START TRACE 和 MODIFY TRACE 指令的參數時，這個值可以指定要追蹤的對應事件。

整合語言環境 (ILE)

一組建構及介面，為所有符合 ILE 的高階語言提供一般執行時期環境及執行時期可連結應用程式介面 (API)。

互動式問題控制系統 (Interactive Problem Control System, IPCS)

MVS 及 z/OS 的元件，允許線上問題管理、互動式問題診斷、磁碟常駐異常終止傾出的線上除錯、問題追蹤及問題報告。

互動式系統輔助機能 (Interactive System Productivity Facility, ISPF)

作為全螢幕編輯器和對話管理程式的 IBM 授權程式。用於撰寫應用程式，它提供一種在應用程式設計師與終端機使用者之間產生標準畫面及互動式對話框的方法。

中繼憑證

不是主要憑證的簽章者憑證。

介面 (interface)

在物件導向程式設計中，這是行為的抽象模型；函數或方法的集合。

網際網路通訊協定 (internet protocol, IP)

這是透過某個網路或互聯網路來遞送資料的通訊協定。這個通訊協定作為較高的通訊協定層和實體網路之間的中介。請參閱傳輸控制通訊協定 (Transmission Control Protocol)。

程序間通訊 (interprocess communication, IPC)

這是程式之間相互傳送訊息的程序。Socket、號誌、信號及內部訊息佇列，是交互程序通訊的一般方法。另請參閱用戶端 (client)。

跨系統通訊 (intersystem communication, ISC)

一種 CICS 機能，為來自其他電腦系統的通訊提供入埠及出埠支援。

IP

請參閱網際網路通訊協定 (Internet Protocol)。

IPC

請參閱交互程序通訊 (interprocess communication)。

IPCS

請參閱互動式問題控制系統 (Interactive Problem Control System)。

ISC

請參閱跨系統通訊 (intersystem communication)。

ISPF

請參閱互動式系統輔助機能 (Interactive System Productivity Facility)。

J**JAAS**

請參閱 Java 鑑別和授權服務 (Java Authentication and Authorization Service)。

Java 鑑別和授權服務 (Java Authentication and Authorization Service, JAAS)

在 Java EE 技術中，執行安全型作業的標準 API。透過 JAAS，服務可以鑑別和授權使用者，同時應用程式也能夠維持獨立於基礎技術之外。

Java 訊息服務 (Java Message Service, JMS)

這是一種應用程式設計介面，負責提供用來處理訊息的 Java 語言功能。另請參閱訊息佇列介面 (Message Queue Interface)。

Java 執行時期環境 (Java runtime environment, JRE)

這是 Java 開發者套件的子集，含有構成標準 Java 平台的核心可執行程式和檔案。JRE 包括 Java 虛擬機器 (JVM)、核心類別和支援檔案。

JMS

請參閱 Java 訊息服務 (Java Message Service)。

JMSAdmin

這是一項管理工具，能讓管理者定義 JMS 物件的內容，並且將它們儲存在 JNDI 名稱空間中。

異動日誌 (journal)

WebSphere MQ for iSeries 用來控制本端物件更新的 OS/400 特性。每一個佇列管理程式庫都包含該佇列管理程式的異動日誌。

JRE

請參閱 [Java 執行時期環境 \(Java runtime environment\)](#)。

K

保持作用中 (keepalive)

這是一種 TCP/IP 機制，小封包會在其中經由網路以預先定義的間隔傳送，以判斷 Socket 是否仍在正常工作中。

Kerberos

這是一種基於對稱金鑰加密法的網路鑑別通訊協定。Kerberos 將唯一金鑰（稱為通行證）指派給登入網路的每一個使用者。該通行證內嵌在透過網路傳送的訊息中。訊息的接收者會使用該通行證，來鑑別傳送者。

金鑰鑑別 (key authentication)

請參閱 [鑑別 \(authentication\)](#)。

金鑰儲存庫 (key repository)

數位憑證及其相關聯金鑰之儲存庫的通用術語。不同類型的金鑰儲存庫包括「憑證管理系統 (CMS)」、「Java 金鑰儲存庫 (JKS)」、「Java 加密法延伸金鑰儲存庫 (JCEKS)」、「公開金鑰加密法標準 12 (PKCS12) 金鑰儲存庫」及 RACF 金鑰環。當區分金鑰儲存庫類型很重要時，文件會依金鑰儲存庫類型的特定名稱來參照金鑰儲存庫類型。在適用於多個金鑰儲存庫類型的環境定義中，會使用通用術語金鑰儲存庫。

金鑰環 (key ring)

這是在電腦安全中，含有公開金鑰、私密金鑰、最高授信使用者和憑證的檔案。

金鑰儲存庫

私密金鑰及對應個人憑證的位置。另請參閱 [信任儲存庫 \(trust store\)](#)

L

最後留言 (last will and testament)

由用戶端向監視器登錄，並在用戶端非預期地結束時由監視器使用的物件。

LDAP

請參閱 [輕量型目錄存取通訊協定 \(Lightweight Directory Access Protocol\)](#)。

輕量型目錄存取通訊協定 (Lightweight Directory Access Protocol, LDAP)

這是利用 TCP/IP 來提供目錄存取的開放式通訊協定，它支援 X.500 模型，且不會引起更複雜的 X.500 目錄存取通訊協定 (DAP) 的資源需求。例如，LDAP 可在網際網路或企業內部網路目錄中，用來尋找人員、組織和其他資源。

線性記載 (linear logging)

在 UNIX 及 Linux 系統上的 WebSphere MQ 中，以及 WebSphere MQ for Windows 中，將重新啟動資料保存在一系列檔案中的處理程序。必要的話，會將新的檔案新增至序列中。在其中寫入資料的空間，不會被重複使用。另請參閱 [循環式記載 \(circular logging\)](#)。

鏈結層次安全 (link level security)

由訊息通道代理程式 (MCA)、通訊子系統或兩者一起運作的組合直接或間接呼叫的安全服務。

接聽器 (listener)

這是偵測送入要求和啟動相關通道的程式。

遠端佇列的本端定義 (local definition of a remote queue)

屬於本端佇列管理程式的 WebSphere MQ 物件，定義另一個佇列管理程式所擁有佇列的屬性。此外，它還用於佇列管理程式別名化及回覆目的地佇列別名化。

locale

這是用來識別語言或地理區及決定格式慣例（如對照、大小寫轉換、字元分類、訊息語言、日期和時間表示法，以及數值表示法）的設定。

區域定義物件 (locally defined object)

在 z/OS 上，這是其定義儲存在頁集零上的物件。該定義只可以由定義它的佇列管理程式來存取。另請參閱廣域定義物件 (globally defined object)。

本端佇列 (local queue)

這是屬於本端佇列管理程式的一種佇列。本端佇列可包含等待處理的訊息清單。另請參閱遠端佇列 (remote queue)。

本端佇列管理程式 (local queue manager)

這是程式所連接並對程式提供訊息佇列作業服務的佇列管理程式。另請參閱遠端佇列管理程式 (remote queue manager)。

log

在 WebSphere MQ 中，這是一個檔案，用來記錄佇列管理程式在接收、傳輸及遞送訊息時所完成的工作，以便在失敗時能夠回復。

日誌控制檔 (log control file)

在 UNIX 及 Linux 系統上的 WebSphere MQ 及 WebSphere MQ for Windows 中，包含監視日誌檔使用所需的資訊 (例如，其大小及位置，以及下一個可用檔案的名稱)。

日誌檔 (log file)

在 UNIX 上的 WebSphere MQ 中，這是 Linux 系統及 WebSphere MQ for Windows，其中會記錄佇列管理程式所控制資料的所有重要變更。如果主要日誌檔已滿，WebSphere MQ 便會配置次要日誌檔。

邏輯單元 (LU)

這是一個存取點，使用者或應用程式可透過該存取點來存取 SNA 網路，以與另一個使用者或應用程式通訊。

邏輯單元 6.2 (logical unit 6.2, LU 6.2)

這是一種 SNA 邏輯單元，可在分散式處理環境中支援程式之間的一般通訊。

邏輯工作單元 ID (LUWID)

這是網路中執行緒的唯一識別名稱。此名稱由完整的邏輯單元網路名稱、邏輯工作單元實例號碼及邏輯工作單元序號組成。

日誌記錄 (log record)

這是一組資料，在日誌檔中會視為單一單元。

日誌記錄序號 (log record sequence number, LRSN)

這是與資料共用成員相關聯之日誌記錄的唯一 ID。Db2 for z/OS 在資料共用環境中使用 LRSN 進行回復。

LRSN

請參閱日誌記錄序號 (log record sequence number)。

LU

請參閱邏輯單元 (logical unit)。

LU 6.2

請參閱邏輯單元 6.2 (logical unit 6.2)。

LU 6.2 交談 (LU 6.2 conversation)

在 SNA 中，這是兩個交易程式之間透過 LU 6.2 階段作業進行的邏輯連線，可讓它們彼此通訊。

LU 6.2 交談層次安全 (LU 6.2 conversation level security)

在 SNA 中，這是一個交談層次安全通訊協定，可讓友機交易程式鑑別起始該交談的交易程式。LU 6.2 交談層次安全也稱為一般使用者驗證。

LU 6.2 階段作業 (LU 6.2 session)

在 SNA 中，這是兩個類型為 6.2 的邏輯單元 (LU) 之間的階段作業。

LU 名稱

VTAM 用來參照網路中節點的名稱。

LUWID

請參閱邏輯工作單元 ID (logical unit of work identifier)。

M

受管理的目的地 (managed destination)

這是由佇列管理程式，為選擇使用受管理訂閱的應用程式所提供的一種佇列，作為傳送已發佈訊息的目的地。另請參閱受管理的訂閱 ([managed subscription](#))。

受管理的控點 (managed handle)

這是在已指定佇列管理程式來管理傳送至訂閱的訊息儲存時，由 MQSUB 呼叫所傳回的 ID。

受管理的訂閱 (managed subscription)

因為應用程式不需要使用特定的佇列，所以佇列管理程式已為此訂閱建立訂閱者佇列來接收發佈。另請參閱受管理的目的地 ([managed destination](#))。

配置 (marshalling)

請參閱序列化 ([serialization](#))。

MCA

請參閱訊息通道代理程式 ([message channel agent](#))。

MCI

請參閱訊息通道介面 ([message channel interface](#))。

媒體映像檔 (media image)

在 UNIX 及 Linux 系統上的 WebSphere MQ 及 WebSphere MQ for Windows 中，這是包含物件映像檔的日誌記錄順序。可從此映像檔重建該物件。

訊息

1. 從人員或程式傳送給另一個人或程式的通訊。
2. 在系統程式設計中，供終端機操作員或系統管理員使用的資訊。

訊息親緣性 (message affinity)

交談式訊息之間的一種關係，這些訊息在兩個應用程式之間進行交換，在這兩個應用程式中，必須由特定佇列管理程式或是在特定序列中處理訊息。

訊息通道 (message channel)

在分散式訊息佇列作業中，這是用於在佇列管理程式之間移動訊息的一種機制。訊息通道包含兩個訊息通道代理程式（一端是傳送端，另一端是接收端），和一個通訊鏈結。另請參閱通道 ([channel](#))。

訊息通道代理程式 (message channel agent, MCA)

這個程式會將已備妥的訊息從傳輸佇列傳輸至通訊鏈結，或從通訊鏈結傳輸至目的地佇列。另請參閱訊息佇列介面 ([Message Queue Interface](#))。

訊息通道介面 (message channel interface, MCI)

在 WebSphere MQ 佇列管理程式與另一個傳訊系統之間傳輸訊息的客戶或供應商撰寫程式必須符合的 WebSphere MQ 介面。WebSphere MQ 架構的一部分。另請參閱訊息佇列介面 ([Message Queue Interface](#))。

訊息消費者 (message consumer)

1. 取得及處理訊息的程式或函數。另請參閱消費者。
2. 在 JMS 中，這是在階段作業內建立以從目的地接收訊息的物件。

訊息環境定義 (message context)

這是訊息發送端的相關資訊，該訊息發送端保留在訊息描述子的欄位中。有 2 個種類的環境定義資訊：身分環境定義和來源環境定義。

訊息描述子 (message descriptor)

說明 WebSphere MQ 訊息中所附帶的訊息格式和呈現方式的控制資訊。訊息描述子的格式是由 MQMD 結構所定義。

訊息結束程式 (message exit)

這是一種通道結束程式，用來修改訊息內容。訊息結束程式通常成對工作，通道的每一端存在一個。在通道的傳送端，訊息通道代理程式 (MCA) 從傳輸佇列中取得訊息之後，會呼叫訊息結束程式。在通道的接收端，訊息通道代理程式 (MCA) 將訊息放置在其目的地佇列中之前，會呼叫訊息結束程式。

訊息流程控制 (message flow control)

這是一個分散式佇列管理作業，涉及設定和維護佇列管理程式之間的訊息遞送。

訊息格式服務 (Message Format Service, MFS)

這是可供應用程式處理簡式邏輯訊息，而不是處理與裝置有關之資料的 IMS 編輯機能，藉此來簡化應用程式的開發流程。

訊息群組 (message group)

這是相關訊息的邏輯群組。此關係由放置訊息的應用程式所定義，並確保如果生產者和消費者都允許分組，則會在順序放置中擷取訊息。

訊息控點 (message handle)

這是對訊息的參照。控點可用來取得訊息之訊息內容的存取權。

訊息標頭 (message header)

這是包含控制資訊（如唯一的訊息 ID）、訊息傳送端和接收端、訊息優先順序以及訊息類型的訊息部分。

訊息輸入描述子 (message input descriptor, MID)

這是「訊息格式服務 (MFS)」控制區塊，用來說明提供給應用程式的資料格式。另請參閱訊息輸出描述子 (message output descriptor)。

訊息接聽器 (message listener)

這是一種物件，扮演非同步訊息消費者的角色。

訊息輸出描述子 (message output descriptor, MOD)

這是「訊息格式服務 (MFS)」控制區塊，用來說明應用程式產生之輸出資料的格式。另請參閱訊息輸入描述子 (message input descriptor)。

訊息優先順序 (message priority)

在 WebSphere MQ 中，這是訊息的一個屬性，可影響佇列上訊息的擷取順序，以及是否產生觸發事件。

訊息生產者 (message producer)

在 JMS 中，這是由階段作業建立的一種物件，用來將訊息傳送至目的地。

訊息內容 (message property)

這是與訊息相關聯的資料，使用名稱/值配對格式。訊息內容可以用來作為訊息選取器，來過濾發佈或是選擇性地從佇列取得訊息。訊息內容可以用來包含關於處理的相關商業資料或狀態資訊，而不需要變更訊息內文。

訊息佇列介面 (Message Queue Interface, MQI)

WebSphere MQ 佇列管理程式所提供的程式設計介面。該程式設計介面容許應用程式存取訊息佇列作業服務。另請參閱 Java 訊息服務、訊息通道代理程式、訊息通道介面。

訊息佇列管理 (message queue management, MQM)

在 WebSphere MQ for HP Integrity NonStop Server 中，這是一種機能，可讓您存取 PCF 指令格式及控制指令，以管理佇列管理程式、佇列及通道。

訊息佇列作業 (message queuing)

這是一種程式設計技術，在該技術中，應用程式中的每一個程式，都會透過將訊息放置在佇列上，與其他程式通訊。

訊息重試 (message-retry)

這是一個選項，適用於無法放置訊息的 MCA。MCA 可以等待預先定義的時間量，然後嘗試再次放置訊息。

訊息區段 (message segment)

這是訊息的其中一個區段，該訊息太大，以致應用程式或佇列管理程式均無法處理。

訊息選取元 (message selector)

在應用程式設計中，這是一個可變長度字串，由應用程式用來僅在其內容滿足選取字串 represents.The 語法基於 SQL92 條件式表示式語法的子集。

訊息序號編定 (message sequence numbering)

這是一種程式設計技術，其中，會在透過通訊鏈結進行傳輸期間，給訊息指定唯一的號碼。這可讓接收程序檢查是否已接收所有訊息，以原始順序將它們放入佇列中，並捨棄重複的訊息。

訊息記號 (message token)

作用中佇列管理程式中訊息的唯一 ID。

方法 (method)

在物件導向的設計或程式設計中，實作作業指定操作行為的軟體。

MFS

請參閱[訊息格式服務 \(Message Format Service\)](#)。

MGAS

請參閱[最廣域位址空間 \(mostly global address space\)](#)。

Microsoft Cluster Server (MSCS)

透過將執行 Windows 的電腦分組到 MSCS 叢集來提供高可用性的技術。如果叢集中的其中一部電腦發生了某個問題，則 MSCS 會依序關閉已中斷的應用程式，將它的狀態資料傳送至叢集中的另一部電腦，並在該電腦上重新起始該應用程式。

Microsoft Transaction Server (MTS)

一種機能，可協助 Windows 使用者在中層伺服器中執行商業邏輯應用程式。MTS 會將工作分成許多活動，這些活動是簡短、獨立的商業邏輯片段。

MID

請參閱[訊息輸入描述子 \(message input descriptor\)](#)。

MOD

請參閱[訊息輸出描述子 \(message output descriptor\)](#)。

模型佇列物件 (model queue object)

這是一組佇列屬性，用來作為程式建立動態佇列時的範本。

最廣域位址空間 (mostly global address space, MGAS)

這是靈活的虛擬位址空間模型，在諸如 HP-UX 等系統中使用，可保留共用應用程式的大部分位址空間。這可以加強共用大量資料之處理程序的效能。另請參閱[最專用位址空間 \(mostly private address space\)](#)。

最專用位址空間 (mostly private address space, MPAS)

這是靈活的虛擬位址空間模型，在諸如 HP-UX 等系統中使用，可將較大的位址空間區塊配置給程序。這可以加強需要大量資料空間之處理程序的效能。另請參閱[最廣域位址空間 \(mostly global address space\)](#)。

MPAS

請參閱[最專用位址空間 \(mostly private address space\)](#)。

MQAI

請參閱[WebSphere MQ 管理介面](#)。

MQI

請參閱[訊息佇列介面 \(Message Queue Interface\)](#)。

MQI 通道 (MQI channel)

伺服器系統上 WebSphere MQ 用戶端與佇列管理程式之間的連線。MQI 通道只傳送 MQI 呼叫，並以雙向方式來回應。另請參閱[通道 \(channel\)](#)。

MQM

請參閱[訊息佇列管理 \(message queue management\)](#)。

MQSC

請參閱[WebSphere MQ Script 指令](#)。

MQSeries

WebSphere MQ 的前一個名稱。

MQ 遙測傳輸

MQ Telemetry Transport (MQTT) 是透過 TCP/IP 流動的開放式輕量型發佈/訂閱通訊協定，可連接大量裝置，例如何伺服器、掣動器、智慧型手機、車輛、住家、性能、遠端感應器及控制裝置。MQTT 設計為在網路可能受到頻寬限制，或裝置可能受到記憶體或處理器等限制的環境中運作。

MQTT

請參閱[MQ Telemetry Transport](#)。

MQTT 用戶端 (MQTT client)

MQTT 用戶端應用程式會連接至具有 MQTT 功能的伺服器，例如 WebSphere MQ Telemetry 通道。您可以撰寫自己的用戶端來使用已發佈的通訊協定，或使用 WebSphere MQ Telemetry 安裝隨附的其中一個用戶端。一般用戶端負責從遙測裝置收集資訊，然後將資訊發佈至伺服器。它還可以訂閱主題、接收訊息，並使用此資訊來控制遙測裝置。部分用戶端隨 WebSphere MQ Telemetry 提供；請參閱遙測用戶端及遙測進階用戶端。

MQTT 伺服器

MQTT 伺服器會處理 MQTT 通訊協定的伺服器端。它通常可讓許多 MQTT 用戶端同時與其連接，並為配送至 MQTT 用戶端的訊息提供一個中心。具有遙測 (MQXR) 服務的 WebSphere MQ 佇列管理程式是 MQTT 伺服器。

MSCS

請參閱 [Microsoft Cluster Server](#)。

MTS

請參閱 [Microsoft Transaction Server](#)。

多躍點 (multi-hop)

這是指當來源佇列管理程式與目標佇列管理程式之間，沒有直接通訊鏈結時，透過一個以上的中介佇列管理程式進行遞送。

多重實例佇列管理程式

該佇列管理程式配置成與其他佇列管理程式實例，共同使用佇列管理程式資料。執行中多重實例佇列管理程式的一個實例處於作用中，其他實例處於待用狀態，以準備從作用中實例中接管。另請參閱 [單一實例佇列管理程式 \(single instance queue manager\)](#)。

N

名單 (namelist)

包含物件名稱清單 (例如，佇列名稱) 的 WebSphere MQ 物件。

名稱服務 (name service)

在 UNIX 及 Linux 系統上的 WebSphere MQ 及 WebSphere MQ for Windows 中，這是決定哪個佇列管理程式擁有指定佇列的機能。

名稱服務介面 (name service interface, NSI)

解析佇列名稱所有權的客戶或供應商撰寫程式必須符合的 WebSphere MQ 介面。WebSphere MQ 架構的一部分。

名稱轉換 (name transformation)

在 UNIX 及 Linux 系統上的 WebSphere MQ 及 WebSphere MQ for Windows 中，這是一個內部處理程序，用於變更佇列管理程式名稱，使其對於所使用的系統是唯一且有效的。在外部，佇列管理程式名稱保持不變。

巢套式工具袋 (nested bag)

在 WebSphere MQ 管理介面 (MQAI) 中，這是插入另一個資料工具袋的系統工具袋

巢狀結構 (nesting)

在「WebSphere MQ 管理介面 (MQAI)」中，這是將 WebSphere MQ 傳回的資訊分組的方法。

網路基本輸入/輸出系統 (NetBIOS) (Network Basic Input/Output System, NetBIOS)

這是網路及個人電腦的標準介面，在區域網路中使用，以提供訊息、列印伺服器及檔案伺服器功能。使用 NetBIOS 的應用程式，不必處理 LAN 資料鏈結控制 (DLC) 通訊協定的詳細資料。

網路基本輸入/輸出系統 (Network Basic Input/Output System)

請參閱 [NetBIOS](#)。

新技術檔案系統 (New Technology File System, NTFS)

Windows 作業環境中的其中一個原生檔案系統。

node

在 Microsoft Cluster Server (MSCS) 中，叢集中的每一部電腦。

不可延續訂閱 (nondurable subscription)

只要訂閱應用程式與佇列管理程式的連線維持開啟，就會存在的訂閱。當訂閱應用程式有意地或由於遺失連線，而中斷與佇列管理程式的連線時，會移除訂閱。另請參閱 [可延續訂閱 \(durable subscription\)](#)。

非持續訊息 (nonpersistent message)

這種訊息在佇列管理程式重新啟動時不會存活。另請參閱 [持續訊息 \(persistent message\)](#)。

NSI

請參閱 [名稱服務介面 \(name service interface\)](#)。

NTFS

請參閱 [新技術檔案系統 \(New Technology File System\)](#)。

NUL

請參閱空值字元 (null character)。

空字元 (null character, NUL)

這是一種具有值 'X'00' 的控制字元，用來代表缺少的顯示字元或列印字元。

O

OAM

請參閱物件權限管理程式 (object authority manager)。

物件 (object)

1. 在 WebSphere MQ 中，佇列管理程式、佇列、程序定義、通道、名單、鑑別資訊物件、管理主題物件、接聽器、服務物件，或 (僅限 z/OS) CF 結構物件或儲存類別。
2. 在物件導向設計或程式設計中，這是由資料及與該資料相關聯的作業所組成之類別的具體實現 (實例)。物件含有類別所定義的實例資料，但類別擁有資料的相關作業。

Object Authority Manager (OAM)

在 UNIX 及 Linux 系統上的 WebSphere MQ 中，WebSphere MQ for IBM i，以及 WebSphere MQ for Windows，這是指令及物件管理的預設授權服務。OAM 可由客戶提供的安全服務取代，也可與其組合執行。

物件描述子 (object descriptor)

識別特定 WebSphere MQ 物件的資料結構。併入描述子中的是物件名稱及物件類型。

物件控點 (object handle)

程式用來存取其使用之 WebSphere MQ 物件的 ID 或記號。

物件導向程式設計 (object-oriented programming)

這是以資料抽象和繼承等概念為基礎的程式設計方法。與程序化程式設計技術不同，物件導向程式設計不著重於如何達成某件事，而是著重於哪些資料物件構成問題以及如何操作它們。

OCSP

線上憑證狀態通訊協定。檢查憑證是否已撤銷的方法。

卸載 (offloading)

在 WebSphere MQ for z/OS 中，這是將佇列管理程式的作用中日誌傳送至其保存日誌的自動處理程序。

單向鑑別 (one way authentication)

在此鑑別方法中，佇列管理程式會將憑證提交至用戶端，但不會檢查從用戶端至佇列管理程式的鑑別。

開啟 (open)

建立對物件 (例如佇列或主題) 的存取權

開放式系統互連 (open systems interconnection, OSI)

這是符合「國際標準組織 (ISO)」標準的開放式系統互連，用於交換資訊。

開放交易管理程式存取 (Open Transaction Manager Access, OTMA)

這是在 MVS Sysplex 環境中，IMS 用來實作交易型無連線主從式通訊協定的元件。通訊協定的網域只限於「z/OS 跨系統連結機能 (XCF)」的網域。OTMA 可以將用戶端連接到伺服器上，讓用戶端可以支援大型網路 (或數目極多的階段作業)，同時維持高效能。

OPM

請參閱原始程式模型 (original program model)。

原始程式模型 (original program model, OPM)

在引進「整合語言環境 (ILE)」模型之前，用來編譯原始碼及建立高階語言程式的函數集。

OSGi 聯盟 (OSGi Alliance)

由超過 20 家公司 (包括 IBM) 組成的聯盟，建立規格以概述語音、資料及多媒體無線和有線網路管理的開放式標準。

OSI

請參閱開放式系統互連 (open systems interconnection)。

OSI 目錄標準 (OSI directory standard)

這是稱為 X.500 的標準，用來定義綜合性的目錄服務，包括資訊模型、名稱空間、功能模型及鑑別架構。X.500 還定義了「目錄存取通訊協定 (DAP)」，用戶端使用該通訊協定來存取目錄。「輕量型目錄

存取通訊協定 (LDAP)」會從目錄用戶端移除 X.500 存取的部分負擔，讓目錄可用於較多種類的機器和應用程式。

OTMA

請參閱[開放式交易管理程式存取 \(Open Transaction Manager Access\)](#)。

出埠通道 (outbound channel)

這個通道用來從傳輸佇列中取得訊息，並將它們傳送至另一個佇列管理程式。

輸出日誌緩衝區 (output log-buffer)

在 WebSphere MQ for z/OS 中，這是在將回復日誌記錄寫入保存日誌之前，保留回復日誌記錄的緩衝區。

輸出參數 (output parameter)

這是 MQI 呼叫的一個參數，在該 MQI 呼叫中，佇列管理程式會在呼叫完成或失敗時傳回資訊。

超載 (overloading)

在物件導向程式設計中，運算子或方法能根據環境定義而具有不同意義的能力。例如，在 C++ 中，當函數和運算子搭配類別類型使用時，使用者可以重新定義函數與大部分標準的運算子。方法名稱或運算子會保持相同，但方法參數在類型、數目上或兩者皆不同。這項差異統稱為函數或運算子的簽章，而每一個簽章都需要個別的實作。

P

頁集 (page set)

當 WebSphere MQ for z/OS 將資料 (例如佇列和訊息) 從主儲存體中的緩衝區移至永久後端儲存體 (DASD) 時所使用的 VSAM 資料集。

母類別 (parent class)

其他類別從該類別繼承實例方法、屬性及實例變數。另請參閱[抽象類別 \(abstract class\)](#)。

局部儲存庫 (partial repository)

這是叢集中佇列管理程式的局部相關資訊集。局部儲存庫由不用來管理完整儲存庫的所有叢集佇列管理程式維護。另請參閱[完整儲存庫 \(full repository\)](#)。

友機佇列管理程式 (partner queue manager)

請參閱[遠端佇列管理程式 \(remote queue manager\)](#)。

PassTicket

在 RACF 維護的登入中，這是動態產生的隨機單次使用密碼替代方案，供工作站或其他用戶端用來登入主機，而不是透過網路來傳送 RACF 密碼。

PCF

請參閱[可程式指令格式 \(programmable command format\)](#)。

擱置事件 (pending event)

由於來自 CICS 配接器的連接要求而發生的未排程事件。

過濾 (percolation)

在錯誤回復中，這是指沿已預先建立的控制路徑，從回復常式至較高階回復常式的傳遞。

效能事件 (performance event)

這是一種事件，用來指出已發生某種限制狀況。

效能追蹤 (performance trace)

WebSphere MQ 追蹤選項，其中追蹤資料將用於效能分析及調整。

永久動態佇列 (permanent dynamic queue)

關閉該動態佇列時，除非明確地要求刪除，否則不會刪除該佇列。如果佇列管理程式失敗，則會回復永久動態佇列，因此它們可以包含持續訊息。另請參閱[暫時動態佇列 \(temporary dynamic queue\)](#)。

持續訊息 (persistent message)

這種訊息在佇列管理程式重新啟動時會存活。另請參閱[非持續訊息 \(nonpersistent message\)](#)。

個人憑證 (personal certificate)

您擁有其對應私密金鑰的憑證。與佇列管理程式或應用程式相關聯。

PGM

請參閱[實用一般多重播送 \(Pragmatic General Multicast\)](#)。

PID

請參閱[程序 ID \(process ID\)](#)。

ping

該指令將「網際網路控制訊息通訊協定 (ICMP)」回應要求的封包，傳送至閘道、路由器或主機，並期望收到回覆。

PKCS

公開金鑰密碼化標準。一組加密法標準，其中：

- 7 代表訊息
- 11 代表硬體安全模組
- 12 代表金鑰儲存庫中使用的檔案格式

PKI

請參閱[公開金鑰基礎架構 \(public key infrastructure\)](#)。

純文字 (plain text)

請參閱[明碼 \(cleartext\)](#)。

回復點 (point of recovery)

在 WebSphere MQ for z/OS 中，這是一組 WebSphere MQ for z/OS 頁集的備份副本，以及回復這些頁集所需的對應日誌資料集。如果發生頁集流失（例如，頁集 I/O 錯誤），則這些備份副本會提供可能的重新啟動點。

有害訊息 (poison message)

指的是在佇列中，接收應用程式無法處理之錯誤格式的訊息。此訊息可能會反覆地分送至輸入佇列，而應用程式會反覆地取消此訊息。

多型性 (polymorphism)

這是一種物件導向程式設計性質，可容許根據實作某個方法的類別，以不同方式來執行它。多型性可讓子類別置換繼承的方法，而不會影響母類別的方法。多型性還可讓用戶端從單一介面，存取物件的兩個以上實作。

實用一般多重播送 (Pragmatic General Multicast, PGM)

這是一種可靠的多重播送傳輸通訊協定，能同步對多個接收者提供可靠的連續封包。

強制關機 (preemptive shutdown)

在 WebSphere MQ 中，關閉佇列管理程式不會等待已連接的應用程式中斷連線，或等待現行 MQI 呼叫完成。另請參閱[立即關閉 \(immediate shutdown\)](#) 和 [靜止關機 \(quiesced shutdown\)](#)。

偏好的電腦 (preferred computer)

在 Microsoft Cluster Server 控制下執行的應用程式所使用的主要電腦。失效接手至另一部電腦之後，MSCS 會監視偏好的電腦，直到它修復好為止，並且只要它已再次正確地執行，就立即將應用程式移回該電腦。

主體 (principal)

這是可以和另一個實體安全通訊的實體。主體是由它的相關安全環境定義來識別，此安全環境定義確定了它的存取權。

私密定義物件 (privately defined object)

請參閱[區域定義物件 \(locally defined object\)](#)。

私密方法和實例資料 (private methods and instance data)

在物件導向程式設計中，這是只有相同類別的實作，才能存取的方法和實例資料。

程序定義物件 (process definition object)

包含 WebSphere MQ 應用程式定義的 WebSphere MQ 物件。例如，佇列管理程式在處理觸發訊息時，會使用該定義。

程序 ID (process ID, PID)

這是代表程序的唯一 ID。程序 ID 是一個正整數，直到此程序的生命期限結束之後才會重複使用它。

產生者 (producer)

這是建立和傳送訊息的一種應用程式。另請參閱 [發佈者](#)，[訊息產生者](#)。

可程式指令格式 (programmable command format, PCF)

由下列應用程式使用的 WebSphere MQ 訊息類型: 使用者管理應用程式、將 PCF 指令放入指定佇列管理程式的系統指令輸入佇列、使用者管理應用程式, 以從指定佇列管理程式及佇列管理程式取得 PCF 指令的結果, 作為發生事件的通知。另請參閱 [WebSphere MQ Script 指令](#)。

暫時修正程式 (program temporary fix, PTF)

在 System i、System p 和 System z 產品中, 這是包含個別或多個可供所有授權客戶使用之修正程式的套件。PTF 可解決問題, 甚至可提供加強功能。

內容 (property)

這是一項說明物件的性質。內容可變更或修改。除了別的以外, 內容還可以說明物件的名稱、類型、值或行為。

受保護的方法和實例資料 (protected methods and instance data)

在物件導向程式設計中, 這是只有相同類別或衍生類別的實作, 或從夥伴類別, 才能存取的方法和實例資料。

Proxy 訂閱 (proxy subscription)

Proxy 訂閱是指一個佇列管理程式針對在另一個佇列管理程式中發佈的主題所進行的訂閱。Proxy 訂閱會針對某個訂閱所訂閱的每個個別主題字串, 在佇列管理程式之間流動。您不會明確建立 Proxy 訂閱, 佇列管理程式會代表您執行此動作。

PTF

請參閱 [暫時修正程式 \(program temporary fix\)](#)。

公開金鑰 (public key)

每個人都知道的鑰匙 此金鑰通常內嵌在指定公開金鑰擁有者的數位憑證中。

公開金鑰加密法 (public key cryptography)

這是使用兩個金鑰的密碼系統: 每個人都知道的公開金鑰, 以及只有訊息收件人知道的私密金鑰。公開和私密金鑰的關聯方式是如此, 使用一個金鑰加密的任何內容都只能由對應的私密金鑰解密。

公開金鑰基礎架構 (public key infrastructure, PKI)

這是由數位憑證、憑證管理中心及其他註冊管理中心所組成的一種系統, 用於驗證及鑑別網路交易中所涉及之每一方的有效性。

公用方法和實例資料 (public methods and instance data)

在物件導向程式設計中, 這是所有類別均可存取的方法和實例資料。

發佈

讓指定主題的相關資訊, 可供發佈/訂閱系統中的佇列管理程式使用。

發佈者

這是一個應用程式, 可讓指定主題的一段相關資訊, 用於發佈/訂閱系統中的分配管理系統。

發佈/訂閱 (publish/subscribe)

一種傳訊互動類型, 由基礎架構將發佈應用程式所提供的資訊, 遞送給對該類型資訊有興趣的所有訂閱應用程式。

發佈/訂閱叢集 (publish/subscribe cluster)

這是一組完全交互連接的佇列管理程式, 組成發佈/訂閱應用程式的多重佇列管理程式網路一部分。

放置

這是在訊息佇列作業中, 使用 MQPUT 或 MQPUT1 呼叫將訊息放置於佇列的作業。另請參閱 [瀏覽 \(browse\)](#)、[取得 \(get\)](#)。

Q

佇列

這是為訊息佇列應用程式保留訊息的物件。佇列由佇列管理程式擁有及維護。

佇列索引 (queue index)

在 WebSphere MQ for z/OS 中, 這是可用來增加佇列上 MQGET 作業速度的訊息 ID 清單或相關性 ID 清單。

佇列管理程式 (queue manager)

這是訊息佇列系統的一個元件, 可為應用程式提供佇列服務。

佇列管理程式事件 (queue manager event)

指出下列其中一項的事件: 與佇列管理程式所使用的資源相關時發生錯誤狀況。例如, 佇列無法使用, 或佇列管理程式中發生重大變更。例如, 佇列管理程式已停止或已啟動。

佇列管理程式群組 (queue manager group)

在用戶端通道定義表 (CCDT) 中, 這是在用戶端在建立與伺服器的連線後, 嘗試連接的佇列管理程式群組。

佇列管理程式層次安全 (queue manager level security)

在 WebSphere MQ for z/OS 中, 使用特定於佇列管理程式的 RACF 設定檔來執行授權檢查。

佇列管理程式集 (queue manager set)

「WebSphere MQ 探險家」中的佇列管理程式群組, 可讓使用者對群組中的所有佇列管理程式執行動作。

佇列共用群組 (queue-sharing group)

在 WebSphere MQ for z/OS 中, 這是位於同一個 Sysplex 中的佇列管理程式群組, 可以存取儲存在共用儲存庫中的物件定義單一集合, 以及儲存在連結機能中的共用佇列單一集合。另請參閱 [共用佇列 \(shared queue\)](#)。

佇列共用群組層次安全 (queue-sharing group level security)

在 WebSphere MQ for z/OS 中, 使用由佇列共用群組中所有佇列管理程式共用的 RACF 設定檔所執行的授權檢查。

靜止 (quiesce)

這是指在允許作用中的作業正常完成之後, 結束程序或關閉系統。

靜止關機 (quiesced shutdown)

1. CICS 配接器的一種關機類型, 配接器會在完成所有目前作用中的作業之後從 WebSphere MQ 中斷連線。另請參閱 [強制關機 \(forced shutdown\)](#)。
2. 在 WebSphere MQ 中, 這是佇列管理程式的關閉, 容許所有已連接的應用程式中斷連線。另請參閱 [立即關閉 \(immediate shutdown\)](#) 和 [強制關機 \(preemptive shutdown\)](#)。

靜止 (quiescing)

在 WebSphere MQ 中, 佇列管理程式在停止之前的狀態。在此狀態下, 容許程式完成處理, 但不容許啟動新程式。

仲裁磁碟 (quorum disk)

由 Microsoft Cluster Server 專門存取的磁碟, 用來儲存叢集回復日誌, 以及判斷伺服器是已啟動還是已關閉。一次只有一個伺服器可以擁有仲裁磁碟。叢集中的伺服器可以協議所有權。

R

RACF

請參閱 [資源存取控制機能 \(Resource Access Control Facility\)](#)。

RAID

請參閱 [獨立磁碟的備用陣列 \(Redundant Array of Independent Disks\)](#)。

RBA

請參閱 [相對位元組位址 \(relative byte address\)](#)。

RC

請參閱 [回覆碼 \(return code\)](#)。

先讀 (read ahead)

這是一個選項, 容許在應用程式要求訊息之前, 先將訊息傳送至用戶端。

原因碼 (reason code)

這是一種回覆碼, 用來說明「訊息佇列介面 (MQI)」呼叫失敗或局部成功的原因。

接收結束程式 (receive exit)

這是一種通道結束程式, 在訊息通道代理程式 (MCA) 跟隨通訊接收取得控制, 並從通訊連線收到資料單元之後, 會呼叫該通道結束程式。另請參閱 [傳送結束程式 \(send exit\)](#)。

接收端通道 (receiver channel)

在訊息佇列作業中, 該通道會回應傳送端通道, 從通訊鏈結取得訊息, 並將它們放置在本端佇列中。

回復日誌 (recovery log)

在 WebSphere MQ for z/OS 中，這是包含回復訊息、佇列及 WebSphere MQ 子系統所需資訊的資料集。另請參閱保存日誌 (archive log)。

回復終止管理程式 (recovery termination manager, RTM)

這種程式透過將控制權傳遞至與終止功能相關聯的回復常式，來處理所有作業的正常和異常終止。

獨立磁碟的備用陣列 (RAID)

這是兩個以上實體磁碟機的集合，用來向主機呈現一個以上邏輯磁碟機的映像。如果發生實體裝置失敗，則由於資料冗餘，可以從陣列中的其他磁碟機中，讀取或重新產生資料。

參照訊息 (reference message)

這種訊息會參照要傳輸的一段資料。參照訊息由訊息結束程式處理，該結束程式會將資料與訊息連接，並將其從訊息分離，以便容許傳輸資料，而不需要在任何佇列上儲存該資料。

登錄

這是一種儲存庫，內含使用者、系統和軟體的存取和配置資訊。

登錄編輯器 (Registry Editor)

在 Windows 中，容許使用者編輯登錄的程式項目。

登錄區 (registry hive)

在 Windows 系統中，這是儲存在登錄中的資料結構。

相對位元組位址 (relative byte address, RBA)

這是資料記錄或控制間隔與儲存體空間的開始處之間的偏移，該儲存體空間配置給它所屬的資料集或檔案。

可靠多重播送傳訊 (reliable multicast messaging, RMM)

這是針對一對多資料遞送或是多對多資料交換，以訊息導向的中介軟體發佈/訂閱方式而設計的高傳輸量且低延遲的傳輸光纖。RMM 會利用 IP 多重播送基礎架構，以確保可調式資源保護及及時資訊配送。

遠端佇列 (remote queue)

這是屬於遠端佇列管理程式的一種佇列。程式可將訊息放置在遠端佇列上，但無法從遠端佇列中取得訊息。另請參閱本端佇列 (local queue)。

遠端佇列管理程式 (remote queue manager)

這是程式無法連接的一種佇列管理程式，即使它與該程式在相同系統中執行也一樣。另請參閱本端佇列管理程式 (local queue manager)。

遠端佇列物件 (remote queue object)

屬於本端佇列管理程式的 WebSphere MQ 物件。此物件用來定義由另一個佇列管理程式所擁有之佇列的屬性。此外，它還用於佇列管理程式別名化及回覆目的地佇列別名化。

遠端佇列作業 (remote queuing)

在訊息佇列作業中，這是所提供的服務，以讓應用程式能夠將訊息放置在屬於其他佇列管理程式的佇列中。

回覆訊息 (reply message)

這是用於回覆要求訊息的一種訊息類型。另請參閱報告訊息 (report message)、要求訊息 (request message)。

回覆目的地佇列 (reply-to queue)

這是發出 MQPUT 呼叫的程式想要將回覆訊息或報告訊息傳送至其中的佇列名稱。

報告訊息 (report message)

這是用於提供其他訊息相關資訊的一種訊息類型。報告訊息可指出訊息已分送、已送達其目的地、已過期，或基於某種原因而無法處理。另請參閱回覆訊息 (reply message)、要求訊息 (request message)。

儲存庫 (repository)

這是有關作為叢集成員之佇列管理程式的資訊集合。此資訊包括佇列管理程式的名稱、位置、通道以及其管理的佇列。

儲存庫佇列管理程式 (repository queue manager)

此佇列管理程式用來管理叢集相關資訊的完整儲存庫。

要求端通道 (requester channel)

在訊息佇列作業中，這個通道可以在本端啟動，以起始伺服器通道的作業。另請參閱伺服器通道 (server channel)。

要求訊息 (request message)

這是用來要求其他程式回覆的一種訊息類型。另請參閱[回覆訊息 \(reply message\)](#)、[報告訊息 \(report message\)](#)。

要求/回覆 (request/reply)

這是利用要求訊息來要求另一個應用程式回覆的一種傳訊應用程式。另請參閱[資料包 \(datagram\)](#)。

RESLEVEL

在 WebSphere MQ for z/OS 中，這是一個選項，用來控制針對 API 資源安全所檢查的使用者 ID 數目。

解析路徑 (resolution path)

這是當應用程式在 MQOPEN 呼叫的輸入中，指定別名或遠端佇列時，所開啟的佇列集。

資源

這是指運算系統或作業系統的一項機能，工作、作業或執行中程式需要該機能。資源包括主儲存體、輸入/輸出裝置、處理單元、資料集、檔案、程式庫、資料夾、應用程式伺服器，以及控制或處理程式。

資源存取控制機能 (Resource Access Control Facility, RACF)

提供存取控制的 IBM 授權程式，方法是識別系統的使用者；驗證系統的使用者；授權存取受保護資源；記載未獲授權進入系統的嘗試；以及記載受保護資源的存取權。

資源配接器 (resource adapter)

Java Enterprise Edition 連接器架構的實作，可讓在應用程式伺服器中執行的 JMS 應用程式和訊息驅動 Bean 存取 WebSphere MQ 佇列管理程式的資源。

資源管理程式 (resource manager)

這是管理和控制共用資源（如記憶體緩衝區和資料集）之存取的應用程式、程式或交易。WebSphere MQ、CICS 和 IMS 都是資源管理程式。

資源回復服務 (Resource Recovery Services, RRS)

這是在 z/OS 中，利用同步點管理程式，在參與的資源管理程式之間協調變更的元件。

回應者 (responder)

這是分散式佇列中的一種程式，可回覆另一個系統的網路連線要求。另請參閱[起始者 \(initiator\)](#)。

重新同步化 (resynch)

在 WebSphere MQ 中，此選項可引導通道啟動並解決任何不確定的狀態訊息，但不重新啟動訊息傳送。

回覆碼 (return code, RC)

這是程式所傳回，用來指示處理結果的值。回覆碼的例子包括完成碼和原因碼。

傳回至傳送端 (return-to-sender)

這是一個選項，適用於無法遞送訊息的 MCA。該 MCA 可以將訊息傳送回發送端。

RSA 演算法 (Rivest-Shamir-Adleman algorithm, RSA)

這是由 RSA Data Security, Inc 開發的公開金鑰加密技術，用於 SSL 的 IBM 實作中。

RMM

請參閱[可靠多重播送傳訊 \(reliable multicast messaging\)](#)。

回復 (rollback)

請參閱[取消 \(backout\)](#)。

主要憑證 (root certificate)

鏈結中的頂端憑證。如果這是自簽憑證，則僅用於簽署其他憑證。另請參閱[自簽憑證](#)

RRS

請參閱[資源回復服務 \(Resource Recovery Services\)](#)。

RSA

請參閱[Rivest-Shamir-Adleman 演算法 \(Rivest-Shamir-Adleman algorithm\)](#)。

RTM

請參閱[回復終止管理程式 \(recovery termination manager\)](#)。

規則表 (rules table)

這是一個包含一個以上規則的控制檔，無法傳送郵件的佇列 (DLQ) 處理程式會將這些規則套用至 DLQ 中的訊息。

S

可調式平行 2 (Scalable Parallel 2, SP2)

IBM 的平行 UNIX 系統: 在高速網路上有效平行 AIX 系統。

SDK

請參閱軟體開發套件 (software development kit)。

SDWA

請參閱系統診斷工作區 (system diagnostic work area)。

SECMEC

請參閱安全機制 (security mechanism)。

Secure Socket Layer (SSL)

這是提供通訊私密性的安全通訊協定。當使用 SSL 時，主從式應用程式可以依照專為了防止竊聽、竄改和訊息偽造而設計的方式來通訊。另請參閱憑證管理中心 (certificate authority)。

安全啟用介面 (security enabling interface, SEI)

檢查授權、提供使用者 ID 或執行鑑別的客戶或供應商撰寫程式必須符合的 WebSphere MQ 介面。WebSphere MQ 架構的一部分。

安全結束程式 (security exit)

這是一個通道結束程式，在通道啟動時，完成起始資料協商之後，會立即呼叫此通道結束程式。安全結束程式通常成對運作，並可在訊息通道及 MQI 通道上呼叫。安全結束程式的主要目的，就是在通道的兩端都啟用訊息通道代理程式 (MCA)，以鑑別其友機。

安全 ID (security identifier, SID)

在 Windows 系統上，這是使用者 ID 的補充，可識別在其中定義使用者的 Windows 安全帳戶管理程式資料庫上的完整使用者帳戶詳細資料。

安全機制 (security mechanism, SECMEC)

這是一種技術工具或技術，用來實作安全服務。機制可能自行運作，也可能與其他機制一起使用，以提供特定服務。安全機制的範例包括存取控制清單、加密法及數位簽章。

安全訊息 (security message)

這是由安全結束程式傳送的其中一則訊息，會在通道兩端呼叫安全結束程式，以與彼此通訊。安全訊息的格式未定義，由使用者決定。

安全服務 (security service)

這是電腦系統中用來保護其資源的服務。存取控制是安全服務的一個範例。

安全支援供應商介面 (Security Support Provider Interface, SSI)

這是網路化應用程式呼叫數個安全支援提供者 (SSP) 之一，以建立已鑑別連線並透過這些連線安全地交換資料的一種方法。它可以在 Windows 系統上使用。

自簽憑證 (self-signed certificate)

憑證中的數位簽章是使用對應於憑證中公開金鑰的私密金鑰來產生。

斷詞法 (segmentation)

這是因為佇列管理程式、佇列或應用程式的訊息太長，而分隔成一些較小的實體訊息，接著再由接收端佇列管理程式或應用程式將其重新組合。

SEI

請參閱安全啟用介面 (security enabling interface)。

選取元 (selector)

這是資料項目的 ID。在「WebSphere MQ 管理介面 (MQAI)」中，有兩種選取元類型: 使用者選取元和系統選取元。

號誌 (semaphore)

在 UNIX 及 Linux 系統中，這是在兩個處理程序之間進行通訊的一般方法，可延伸信號的特性。

傳送端通道 (sender channel)

在訊息佇列作業中，這個通道用來起始傳送，從傳輸佇列中移除訊息，並透過通訊鏈結將它們移至接收端或要求端通道。

傳送結束程式 (send exit)

這是一種通道結束程式，在訊息通道代理程式 (MCA) 發出通訊傳送，以透過通訊連線傳送資料單元之前，會呼叫該通道結束程式。另請參閱接收結束程式 (receive exit)。

序列封包交換通訊協定 (Sequenced Packet Exchange protocol, SPX)

這是一種階段作業導向的網路通訊協定，用來在網路中的兩個節點之間提供連線導向的服務，主要由主從式應用程式來使用。它依賴於「網際網路封包資料交換 (IPX)」通訊協定，提供流程控制及錯誤回復，並保證實體網路的可靠性。

序號覆蓋值 (sequence number wrap value)

在 WebSphere MQ 中，這是一種方法，可確保通訊鏈結兩端同時重設其現行訊息序號。使用序號傳輸訊息，將確保接收端通道在儲存訊息時，可以重新建立訊息序列。

序列化 (serialization)

在物件導向程式設計中，這是指依循序方式，將資料從程式記憶體寫到通訊媒體中。

伺服器

1. 一種佇列管理程式，為遠端工作站上執行的用戶端應用程式提供佇列服務。
2. 向其他軟體程式或其他電腦提供服務的軟體程式或電腦。另請參閱用戶端 (client)。

伺服器通道 (server channel)

在訊息佇列作業中，這個通道用來回應要求端通道，從傳輸佇列中移除訊息，並透過通訊鏈結將它們移至要求端通道。另請參閱要求端通道 (requester channel)。

伺服器連線通道類型 (server-connection channel type)

這是與執行佇列管理程式之伺服器相關聯的 MQI 通道定義類型。另請參閱用戶端連線通道類型 (client-connection channel type)。

服務間隔 (service interval)

這是一個時間間隔，針對該時間間隔，佇列管理程式會比較放置作業或取得作業與後續取得作業之間的歷時，以判定是否符合服務間隔事件的條件。佇列的服務間隔由佇列屬性來指定。

服務間隔事件 (service interval event)

這是與服務間隔相關的事件。

服務物件 (service object)

這是在佇列管理程式啟動時，可以啟動其他程序，而且在佇列管理程式停止時，可以停止程序的一種物件。

階段作業

網路上兩個工作站、軟體程式或裝置之間的邏輯或虛擬連線，可讓兩個元素在階段作業期間通訊及交換資料。

階段作業 ID (session ID)

在 WebSphere MQ for z/OS 中，CICS 唯一 ID，定義訊息通道代理程式將訊息從傳輸佇列移至鏈結時要使用的通訊鏈結。

階段作業層次鑑別 (session-level authentication)

在「系統網路架構 (SNA)」中，這是階段作業層次的安全通訊協定，可讓兩個邏輯單元 (LU) 在啟動階段作業時，彼此鑑別。階段作業層次鑑別亦稱為 LU-LU 驗證。

階段作業層次加密法 (session-level cryptography)

在「系統網路架構 (SNA)」中，這是一種方法，用來加密和解密在兩個邏輯單元 (LU) 之間的階段作業中流動的資料。

共用入埠通道 (shared inbound channel)

在 WebSphere MQ for z/OS 中，這是由接聽器使用群組埠啟動的通道。共用通道的通道定義，可以儲存在頁集零 (專用) 或共用儲存庫 (廣域) 中。

共用出埠通道 (shared outbound channel)

在 WebSphere MQ for z/OS 中，這是從共用傳輸佇列移動訊息的通道。共用通道的通道定義，可以儲存在頁集零 (專用) 或共用儲存庫 (廣域) 中。

共用佇列 (shared queue)

在 WebSphere MQ for z/OS 中，這是一種本端佇列類型。佇列中的訊息可以儲存在連結機能中，並可由佇列共用群組中的一個以上佇列管理程式存取。佇列的定義儲存在共用儲存庫中。另請參閱佇列共用群組 (queue-sharing group)。

共用儲存庫 (shared repository)

在 WebSphere MQ for z/OS 中，這是一個共用 Db2 資料庫，用來保留廣域定義的物件定義。

共用交談 (sharing conversations)

這是能讓多個交談共用通道實例，或是共用通道實例之交談的功能。

Shell

這是使用者與作業系統之間的軟體介面。Shell 通常屬於下列兩種種類之一：指令行 Shell，用來提供作業系統的指令行介面；圖形 Shell，用來提供圖形使用者介面 (GUI)。

SID

請參閱[安全 ID \(security identifier\)](#)。

信號 (signal)

這是一種機制，程序透過該機制可以知道系統中發生的事件，或受該事件影響。此類事件的範例包括硬體異常狀況及程序的特定動作。

信號 (signaling)

在 WebSphere MQ for z/OS 和 WebSphere MQ for Windows 中，這是一項特性，可讓作業系統在預期訊息到達佇列時通知程式。

簽章 (signature)

這是與方法相關聯的類型集合。簽章包括回覆值的類型（如果有的話），以及號碼、順序和每一個方法引數的類型。

簽章者憑證 (signer certificate)

用於加密或簽署的憑證。

單一實例佇列管理程式 (single instance queue manager)

這是沒有多個實例的佇列管理程式。另請參閱[多重實例佇列管理程式 \(multi-instance queue manager\)](#)。

單一記載 (single logging)

記錄 WebSphere MQ for z/OS 活動的方法，其中每一個變更只會記錄在一個資料集上。另請參閱[雙重記載 \(dual logging\)](#)。

一段式取消 (single-phase backout)

這是一種方法，其中，進行中的動作不得完成，且必須將隸屬於該動作的所有變更都復原。

一段式確定 (single-phase commit)

這是一種方法，其中，程式無需協調這些更新項目與由另一個資源管理程式控制的資源更新項目（由程式製作），就能確定「確定資源」的更新項目。

SIT

請參閱[系統起始設定表 \(system initialization table\)](#)。

SMF

請參閱[系統管理機能 \(System Management Facilities\)](#)。

SNA

請參閱[系統網路架構 \(Systems Network Architecture\)](#)。

軟體開發套件 (software development kit, SDK)

這是一組工具、API 和文件，可協助您以特定電腦語言或針對特定作業環境進行軟體開發。

來源佇列管理程式 (source queue manager)

請參閱[本端佇列管理程式 \(local queue manager\)](#)。

SP2

請參閱[可調式平行 2 \(Scalable Parallel 2\)](#)。

SPX

請參閱[序列封包交換通訊協定 \(Sequenced Packet Exchange protocol\)](#)。

SSI

請參閱[安全支援提供者介面 \(Security Support Provider Interface\)](#)。

SSL

請參閱[Secure Sockets Layer](#)。

SSLPeer

發證者中的值代表遠端個人憑證的識別名稱。

SSL 或 TLS 用戶端 (SSL or TLS client)

連線的起始端。來自佇列管理程式的其中一個出埠通道也是 SSL 或 TLS 用戶端。

待用佇列管理程式實例 (standby queue manager instance)

這是執行中多重實例佇列管理程式的一個實例，準備從作用中實例中接管。多重實例佇列管理程式有一個以上的待用實例。

段落 (stanza)

這是檔案中行的群組，這些行在一起具有一般功能或定義系統組件。段落通常由空白行或冒號區隔，且每一個段落都有一個名稱。

星形連接的通訊網路 (star-connected communications network)

在這種網路中，所有節點都連接至一個中心節點。

儲存類別 (storage class)

在 WebSphere MQ for z/OS 中，這是用來保留特定佇列之訊息的頁集。儲存類別是在定義佇列時指定。

儲存及轉遞 (store and forward)

這是指在將封包、訊息或訊框重新傳輸至它們的目的地之前，會暫時儲存它們。

串流 (streaming)

在物件導向程式設計中，這是類別資訊及物件實例資料的序列化。

訂閱

要求主題的相關資訊。

子系統 (subsystem)

在 z/OS 中，這是一個服務提供者，它會執行一或多項功能，但在提出要求之前不會執行任何動作。例如，每一個 WebSphere MQ for z/OS 佇列管理程式或 Db2 for z/OS 資料庫管理系統的實例都是 z/OS 子系統。

監督程式呼叫 (supervisor call, SVC)

這是一個指示，用來岔斷執行中的程式，並將控制傳遞給監督程式，以便該監督程式可以執行由指示所指出的特定服務。

SVC

請參閱監督程式呼叫 (supervisor call)。

切換 (switchover)

這是從作用中多重實例佇列管理程式實例，變成待用實例。切換是操作員有意地停止作用中多重實例佇列管理程式實例的結果。

交換器設定檔 (switch profile)

在 WebSphere MQ for z/OS 中，這是在 WebSphere MQ 啟動或發出重新整理安全指令時使用的 RACF 設定檔。WebSphere MQ 偵測到的每一個交換器設定檔都會關閉檢查指定的資源。

對稱金鑰加密法 (symmetric key cryptography)

這是一種加密系統，其中，訊息的傳送端和接收端共用單一且一般的秘密金鑰（用來加密和解密訊息）。此系統不提供任何鑑別。另請參閱非對稱金鑰加密法 (asymmetric key cryptography)。

表徵字串 (symptom string)

以結構化格式顯示的診斷資訊，設計用於搜尋 IBM 軟體支援資料庫。

同步傳訊 (synchronous messaging)

這是程式之間的一種通訊方法，程式將訊息放到訊息佇列之後，便等待其訊息的回覆，再回復它自己的處理。另請參閱非同步傳訊 (asynchronous messaging)。

同步點 (sync point)

這是在交易處理期間，受保護的資源都保持一致的點。

Sysplex

這是一組透過特定多重系統硬體元件和軟體服務來彼此通訊的 z/OS 系統。

系統工具袋 (system bag)

這是一種資料工具袋，由 MQAI 建立。

系統控制指令 (system control commands)

這是用來操作平台專用實體（例如，緩衝池、儲存類別及頁集）的指令。

系統診斷工作區 (system diagnostic work area, SDWA)

在 z/OS 環境中，這是記錄在 SYS1.LOGREC 項目。

系統起始設定表 (system initialization table, SIT)

包含 CICS 在啟動時所使用的參數的表格。

系統項目 (system item)

這是一種資料項目，由 MQAI 建立。

系統管理機能 (System Management Facilities, SMF)

z/OS 的元件，用於收集並記錄各種系統及工作相關資訊。

系統選取元 (system selector)

在「WebSphere MQ 管理介面 (MQAI)」中，這是在建立資料工具袋時包含在資料工具袋中的系統項目 ID。

系統網路架構 (Systems Network Architecture, SNA)

這是邏輯結構、格式、通訊協定及作業序列（用於透過網路傳輸資訊，及控制網路配置和作業）的說明。

T**竄改 (tampering)**

這是指通訊安全的中斷，在該中斷中，運送中的資訊已變更或被取代，然後將其傳送至收件者。另請參閱竊聽 ([eavesdropping](#)) 和模擬 ([impersonation](#))。

目標程式庫高階限定元 (target library high-level qualifier, thlqual)

z/OS 目標資料集名稱的高階限定元。

目標佇列管理程式 (target queue manager)

請參閱遠端佇列管理程式 ([remote queue manager](#))。

作業控制區塊 (task control block, TCB)

z/OS 控制區塊，用來傳達連接至子系統之位址空間內的作業相關資訊。

作業切換 (task switching)

這是指數個作業之間 I/O 作業與處理的重疊。

TCB

請參閱作業控制區塊 ([task control block](#))。

TCP

請參閱傳輸控制通訊協定 ([Transmission Control Protocol](#))。

TCP/IP

請參閱傳輸控制通訊協定/網際網路通訊協定 ([Transmission Control Protocol/Internet Protocol](#))。

technote

這是單一主題的相關簡短文件。

遙測通道 (telemetry channel)

遙測通道是 WebSphere MQ 上的佇列管理程式與 MQTT 用戶端之間的通訊鏈結。每一個通道可能會有一個以上的遙測裝置與其相連接。

遙測進階用戶端

進階遙測用戶端安裝在主要 WebSphere MQ 安裝的 mqxr 子資料夾中。它們是小覆蓋區，MQTT 伺服器容許多個 MQTT 用戶端連接至它，並提供 WebSphere MQ 的上行鏈路或橋接器。當上行鏈路連線中斷時，進階用戶端可以代表用戶端啟動訊息。

遙測用戶端

遙測用戶端是安裝在主要 WebSphere MQ 安裝的 mqxr 子資料夾內的 MQTT 用戶端。遙測用戶端會使用 MQTT 通訊協定來連接至 MQ。

遙測 (MQXR) 服務

處理 MQTT 通訊協定的伺服器一半的 MQ 服務 (請參閱「MQTT 伺服器」)。遙測 (MQXR) 服務會管理遙測通道。

暫時動態佇列 (temporary dynamic queue)

這是一種動態佇列，當該動態佇列關閉時，會將其刪除。如果佇列管理程式失敗，則不會回復暫時動態佇列，因此它們可以包含非持續訊息。另請參閱永久動態佇列 ([permanent dynamic queue](#))。

兆空間 (teraspace)

這是一個 TB 的暫時儲存體區域，可提供程序專用的儲存體。

終止通知 (termination notification)

CICS 子系統順利連接至 WebSphere MQ for z/OS 時啟動的擱置事件。

thlqual

請參閱目標程式庫高階限定元 ([target library high-level qualifier](#))。

執行緒 (thread)

這是在程序控制之下的電腦指令串流。在部分作業系統中，執行緒是程序中最小的作業單位。多個執行緒可以同時執行，處理不同的工作。

TID

請參閱交易 ID ([transaction identifier](#))。

與時間無關的傳訊 (time-independent messaging)

請參閱非同步傳訊 ([asynchronous messaging](#))。

TLS

傳輸層安全-SSL 的後置作業。

TMF

請參閱交易管理程式機能 ([Transaction Manager Facility](#))。

TMI

請參閱觸發監視器介面 ([trigger monitor interface](#))。

TP

請參閱交易程式 ([transaction program](#))。

trace

這是電腦程式或交易的處理記錄。從追蹤中收集的資訊，可用於評量問題及效能。

交易 ID (transaction ID)

請參閱交易 ID ([transaction identifier](#))。

交易 ID (transaction identifier, TID, transaction ID, XID)

這是指派給交易的唯一名稱，用來識別這項交易的相關動作。

交易管理程式 (transaction manager)

這是一個軟體單元，可透過管理廣域交易和協調決策（確定它們或將它們回復），來協調資源管理程式的活動。

交易管理程式機能 (Transaction Manager Facility, TMF)

在 IBM WebSphere MQ for HP Integrity NonStop Server 中，這是一個子系統，用來保護您的商業交易及資料庫的完整性。通常與 NonStop 交易管理程式 /MP 同義。

交易程式 (transaction program, TP)

這是在 SNA 網路中處理交易的程式。

傳輸控制通訊協定 (TCP)

這是在網際網路以及跨網路通訊協定遵循「網際網路工程工作小組 (IETF)」標準的任何網路中，所使用的通訊協定。在封包交換式的通訊網路中，以及在這類網路交互連接的系統中，TCP 提供了可靠的主機對主機通訊協定。另請參閱網際網路通訊協定 ([Internet Protocol](#))。

傳輸控制通訊協定/網際網路通訊協定 (Transmission Control Protocol/Internet Protocol, (TCP/IP)

這是一種非專利的業界標準通訊協定集，在交互連接的不同類型網路中，用來提供可靠的應用程式間點對點連線。

傳輸程式 (transmission program)

請參閱訊息通道代理程式 ([message channel agent](#))。

傳輸佇列 (transmission queue)

這是一個本端佇列，在其中會暫時儲存指向遠端佇列管理程式的備妥訊息。

觸發佇列 (triggered queue)

本端佇列，當它已設定觸發，且符合觸發條件時，需要寫入觸發訊息。

觸發事件 (trigger event)

這是一個事件（例如，訊息抵達佇列），會導致佇列管理程式在起始佇列中建立一則觸發訊息。

觸發 (triggering)

在 WebSphere MQ 中，這是一種機能，可讓佇列管理程式在滿足佇列上預先決定的條件時自動啟動應用程式。

觸發訊息 (trigger message)

這則訊息包含觸發監視器要啟動之程式的相關資訊。

觸發監視器 (trigger monitor)

這是一個持續執行的應用程式，可服務一個以上的起始佇列。當觸發訊息抵達起始佇列時，觸發監視器會擷取該訊息。它會使用觸發訊息中的資訊，來啟動一個程序，該程序會服務發生觸發事件的佇列。

觸發監視器介面 (trigger monitor interface, TMI)

客戶或供應商撰寫的觸發監視器程式必須符合的 WebSphere MQ 介面。WebSphere MQ 架構的一部分。

信任儲存庫

放置 CA 憑證以驗證來自遠端系統的憑證的位置。另請參閱 [金鑰儲存庫 \(key store\)](#)

雙向鑑別 (two way authentication)

在此鑑別方法中，佇列管理程式及用戶端會互相提交憑證。亦稱為交互鑑別。

兩階段確定 (two-phase commit)

兩個步驟的程序，經由此程序確定可回復的資源及外部子系統。第一步驟期間，會輪詢資料庫管理程式的次系統，以確認它們已備妥來確定。如果所有子系統都有正面的回應，則資料庫管理程式會指示它們進行確定。

類型

指定資料內部格式並決定資料使用方式的一種性質。

U

UDP

請參閱 [使用者資料包通訊協定 \(User Datagram Protocol\)](#)。

未獲授權的存取 (unauthorized access)

這是指無需許可權就能獲得電腦系統內資源的存取權。

無法遞送的訊息佇列 (undelivered message queue)

請參閱 [無法傳送郵件的佇列 \(dead-letter queue\)](#)。

復原/重做記錄 (undo/redo record)

這是回復中使用的日誌記錄。記錄的重做部分說明要對 WebSphere MQ 物件進行的變更。復原組件說明在工作尚未確定時，如何取消變更。

回復單元 (unit of recovery)

單一資源管理程式內可回復的作業序列，例如 Db2 for z/OS 實例。另請參閱 [工作單元 \(unit of work\)](#)。

工作單元 (UOW)

這是應用程式在一致性的兩點之間，所執行之作業的可回復序列。當交易開始或到達使用者要求的同步點時，工作單元便會開始。使用者要求的同步點或交易結束時，工作單元即會結束。

UOW

請參閱 [工作單元 \(unit of work\)](#)。

使用者工具袋 (user bag)

在 MQAII 中，這是一種資料工具袋，由使用者建立。

使用者資料封包通訊協定 (UDP)

這是一種網際網路通訊協定，可提供不可靠的無連線資料包服務。它會讓一部機器或程序上的應用程式，將資料包傳送至另一部機器或程序上的應用程式。

使用者項目 (user item)

在 MQAII 中，這是一種資料項目，由使用者建立。

使用者選取元 (user selector)

在「WebSphere MQ 管理介面 (MQAII)」中，這是與資料項目一起放置在資料工具袋中以識別資料項目的 ID。WebSphere MQ 提供 WebSphere MQ 物件的預先定義使用者選取元。

使用者記號 (user token, UTKEN)

封裝或代表使用者安全性質的 RACF 安全記號。RACF 會將 UTKEN 指派給系統中的每一個使用者。

公用程式 (utility)

在 WebSphere MQ 中，這是所提供的一組程式，除了 WebSphere MQ 指令所提供的機能之外，還提供系統操作員或系統管理者的機能。

UTOKEN

請參閱 [使用者記號 \(user token\)](#)。

V

value

這是資料項目的內容。這可以是整數、字串或另一個資料工具袋的控點。

虛擬方法 (virtual method)

在物件導向程式設計中，這是用來顯出多型性的方法。

W

WebSphere MQ

提供訊息佇列作業服務的 IBM 授權程式系列。

WebSphere MQ 管理介面 (MQAI)

透過使用資料工具袋，在 WebSphere MQ 佇列管理程式上執行管理作業的程式設計介面。資料工具袋可讓使用者處理 WebSphere MQ 物件的內容 (或參數)。

WebSphere MQ for .NET 類別

一組類別，容許在 .NET 程式設計架構中撰寫的程式連接至 WebSphere MQ 作為 WebSphere MQ 用戶端，或直接連接至 WebSphere MQ 伺服器。

適用於 C++ 的 WebSphere MQ 類別

一組類別，以 C++ 程式設計語言封裝「WebSphere MQ 訊息佇列介面 (MQI)」。

WebSphere MQ classes for Java

以 Java 程式設計語言封裝 WebSphere MQ 訊息佇列介面 (MQI) 的一組類別。

WebSphere MQ 完全受管理 .NET 用戶端

這是 WebSphere MQ 產品的一部分，可以安裝在系統上，而不需要安裝完整佇列管理程式。WebSphere MQ .NET 用戶端由完全受管理的 .NET 應用程式使用，並與伺服器系統上的佇列管理程式進行通訊。未完全受管理的 .NET 應用程式會使用 WebSphere MQ MQI 用戶端。另請參閱 [用戶端](#)、[WebSphere MQ MQI 用戶端](#)、[WebSphere MQ Java 用戶端](#)。

WebSphere MQ Java 用戶端

這是 WebSphere MQ 產品的一部分，可以安裝在系統上，而不需要安裝完整佇列管理程式。WebSphere MQ Java 用戶端供 Java 應用程式 (適用於 Java 的 WebSphere MQ 類別及適用於 JMS 的 WebSphere MQ 類別) 使用，並與伺服器系統上的佇列管理程式進行通訊。另請參閱 [用戶端](#)、[WebSphere MQ MQI 用戶端](#)、[WebSphere MQ 完全受管理 .NET 用戶端](#)。

WebSphere MQ MQI 用戶端

這是 WebSphere MQ 產品的一部分，可以安裝在系統上，而不需要安裝完整佇列管理程式。WebSphere MQ MQI 用戶端接受來自應用程式的 MQI 呼叫，並與伺服器系統上的佇列管理程式進行通訊。另請參閱 [用戶端](#)、[WebSphere MQ Java 用戶端](#)、[WebSphere MQ 完全受管理 .NET 用戶端](#)。

WebSphere MQ Script 指令 (MQSC)

在所有平台之間統一的人類可讀指令，用來操作 WebSphere MQ 物件。另請參閱 [可程式指令格式 \(programmable command format\)](#)。

WebSphere MQ 伺服器

這是一個佇列管理程式，可為一個以上的用戶端提供佇列作業服務。所有 WebSphere MQ 物件 (例如佇列) 都只存在於佇列管理程式系統上，即 MQI 伺服器機器上。伺服器也可以支援一般本端 MQI 應用程式。

WebSphere MQ Telemetry

WebSphere MQ Telemetry 提供小型用戶端程式庫，可內嵌至在許多不同裝置平台上執行的智慧型裝置。使用用戶端建置的應用程式會使用 MQ Telemetry Transport (MQTT) 及 WebSphere MQ Telemetry (MQXR) 服務，透過 WebSphere MQ 可靠地發佈及訂閱訊息。選取 WebSphere MQ 自訂安裝選項以安裝遙測時，它會安裝：1) 遙測 (MQXR) 服務 2) 遙測用戶端及 3) 遙測進階用戶端。

用於裝置的 WebSphere MQ Telemetry 常駐程式

適用於裝置的 WebSphere MQ Telemetry 常駐程式是進階 MQTT V3 用戶端。它是專為內嵌系統設計的超小型 MQTT 伺服器。

Windows NT 盤查/回應

在包括 Windows NT 系統及獨立式系統的網路上使用的鑑別通訊協定。

竊聽 (wiretapping)

這是用來取得資訊存取權的動作，此資訊沿佈線或通訊中使用的任何其他類型導體進行傳送。竊聽的目標是要取得資訊的未獲授權存取權，而不被偵測到。

X

X509

適用於 PKI 的「國際電信協會」標準。指定公開金鑰憑證的格式及公開金鑰加密法。

XCF

請參閱跨系統連結機能 (cross-system coupling facility)。

XID

請參閱交易 ID (transaction identifier)。

X/Open XA

這是指「X/Open 分散式交易處理 XA」介面。這是一項提議的分散式交易通訊標準。此標準指定在交易內可供存取共用資源的資源管理程式之間的雙向介面，以及在監視和解析交易的交易服務之間的雙向介面。

IBM WebSphere MQ 的協助工具特性

協助工具特性可幫助有身體障礙的使用者（例如行動不便或視力不良者）順利地使用資訊技術內容。

協助工具特性

IBM WebSphere MQ 包含下列主要協助工具特性：

- 僅限鍵盤作業
- 使用螢幕閱讀器的作業

IBM WebSphere MQ 使用最新的 W3C 標準 WAI-ARIA 1.0 (<https://www.w3.org/TR/wai-aria/>)，以確保符合 US Section 508 (<https://www.access-board.gov/guidelines-and-standards/communications-and-it/about-the-section-508-standards/section-508-standards>)，及 Web 內容協助工具準則 (WCAG) 2.0 (<https://www.w3.org/TR/WCAG20/>)。為了充分利用協助工具特性，請結合使用最新版的螢幕閱讀器與本產品支援的最新 Web 瀏覽器。

IBM Documentation 中的 IBM WebSphere MQ 線上產品資訊已啟用協助工具。IBM Documentation 的協助工具特性在 <https://www.ibm.com/docs/about/releasesnotes.html> 中有說明。

鍵盤導覽

本產品使用標準導覽鍵。

介面資訊

使用 IBM WebSphere MQ 的完全可存取方式是使用指令行介面。如需使用指令的相關資訊，請參閱 [如何使用 IBM WebSphere MQ 控制指令](#) 及 [使用 MQSC 指令進行管理](#)。

對於 Windows，安裝 IBM WebSphere MQ 的可存取方式是使用非互動式安裝。如需進一步資訊，請參閱 [使用 msixec 進行進階安裝](#)。

IBM WebSphere MQ 使用者介面沒有每秒閃動 2 - 55 次的內容。

IBM WebSphere MQ Web 使用者介面不依賴階式樣式表來正確呈現內容以及提供有用的體驗。不過，產品說明文件不依賴於階式樣式表。IBM WebSphere MQ 為視力不良的使用者提供同等方式來使用使用者的系統顯示設定，包括高對比模式。您可以使用裝置或瀏覽器設定來控制字型大小。

相關的協助工具資訊

除了標準 IBM 服務台與支援中心網站，IBM 也建立 TTY 電話服務，以方便失聰或聽力不佳的客戶存取銷售及支援服務：

TTY 服務
800-IBM-3383 (800-426-3383)
(北美洲內部)

IBM 和協助工具

如需 IBM 對協助工具之承諾的相關資訊，請參閱 [IBM 協助工具 \(www.ibm.com/able\)](http://www.ibm.com/able)。

Windows 上的協助工具

IBM WebSphere MQ 使用者介面不使用任何特殊按鍵，而是遵循快速功能表、對話框及對話框控制項 (例如按鈕) 等項目上快速鍵的 Windows 使用者介面準則。以一般方式存取快速鍵。如需相關資訊，請參閱 Windows 說明 (請參閱 鍵盤的 Windows 說明索引; 如需協助工具特性，請尋找 協助工具)。

協助工具的特殊特性

IBM WebSphere MQ 中的部分使用者介面通常是視覺化的，但在啟動協助工具特性時它們會有不同的行為，如下所示：

- 高對比模式

在此模式中，「啟動程式」、「準備 IBM WebSphere MQ 精靈」、「明信片」及「預設配置」都會隱藏其背景點陣圖，並確保它們使用系統文字顏色，以便它們易於顯示及讀取。

- 螢幕閱讀器模式

當螢幕閱讀器處於作用中狀態時，「準備 IBM WebSphere MQ 精靈」、「預設配置」及「明信片」會隱藏背景點陣圖、產生的效果、陰影框及其他可能讓螢幕閱讀器混淆的效果，以簡化其外觀。

- 瀏覽器物件狀態

「IBM WebSphere MQ」的「探險家」元件使用圖示來指出物件 (例如佇列管理程式) 的狀態。螢幕閱讀器無法解譯這些圖示，因此有一個選項，可用來顯示圖示的文字說明。如果要選取這個選項，請從「探險家」內按一下 **視窗 > 喜好設定 > WebSphere MQ 探險家**，然後選取 **在物件名稱之後顯示物件狀態**。

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 IBM 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](http://www.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: