

9.3

IBM MQ 實務範例

IBM

附註

使用本資訊及其支援的產品之前，請先閱讀第 197 頁的『[注意事項](#)』中的資訊。

除非新版中另有指示，否則此版本適用於 IBM® MQ 9.5.3 版及所有後續版本與修訂版。

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

© Copyright International Business Machines Corporation 2007, 2024.

目錄

實務練習	5
開始使用 IBM MQ.....	5
規劃解決方案.....	5
實作解決方案.....	6
下一步.....	16
點對點實務範例.....	17
規劃解決方案.....	17
實作解決方案.....	18
保護點對點拓撲.....	24
串流佇列.....	27
串流佇列配置.....	28
串流至遠端及別名佇列.....	29
串流佇列限制.....	30
串流佇列和交易.....	30
與叢集佇列之間的串流.....	31
使用串流佇列來儲存訊息歷程.....	31
發佈/訂閱實務.....	32
實務範例: 建立發佈/訂閱叢集.....	32
發佈/訂閱階層實務.....	38
交易式支援實務.....	47
介紹工作單元.....	47
實務範例 1: 佇列管理程式執行協調.....	48
實務範例 2: 其他軟體提供協調.....	69
即將到期的廣域工作單元.....	75
回復處置單元.....	76
安全實務範例.....	76
安全實務範例: z/OS 上的兩個佇列管理程式.....	76
安全實務範例: z/OS 上的佇列共用群組.....	83
伺服器至伺服器訊息通道攔截範例配置.....	88
使用 SSL/TLS 連接兩個佇列管理程式.....	89
將用戶端安全連接至佇列管理程式.....	96
在 Windows 上移轉.....	101
規劃解決方案.....	101
使用圖形使用者介面實作解決方案.....	106
在 Windows 上安裝更新版本的 IBM MQ 以與舊版共存.....	128
多個安裝的概觀.....	128
將較新版本的 IBM MQ 並列安裝至較舊版本。.....	128
使用 setmqenv 指令來執行 IBM MQ 的兩個版本.....	130
建立佇列管理程式.....	132
將佇列管理程式移轉至更新版本的 IBM MQ.....	133
在 IBM MQ 9.3 上安裝修正套件.....	135
Managed File Transfer 實務範例.....	136
MFT 一般拓撲.....	136
配置基本伺服器.....	139
開始使用 IBM MQ Internet Pass-Thru.....	146
驗證 MQIPT 是否正確運作.....	148
建立金鑰環檔案.....	149
建立測試憑證.....	151
鑑別 TLS 伺服器.....	152
鑑別 TLS 用戶端.....	155
鑑別 TLS 用戶端及伺服器.....	157
配置 HTTP 通道作業.....	160

配置存取控制.....	162
配置 SOCKS Proxy.....	164
配置 SOCKS 用戶端.....	165
配置 MQIPT 叢集作業支援.....	167
配置埠號.....	170
使用 LDAP 伺服器擷取 CRL.....	171
以 TLS Proxy 模式執行 MQIPT.....	174
使用安全管理程式以 TLS Proxy 模式執行 MQIPT.....	175
使用安全結束程式.....	178
使用安全結束程式將用戶端連線要求遞送至 IBM MQ 佇列管理程式伺服器.....	179
動態遞送用戶端連線要求.....	182
使用憑證結束程式來鑑別 TLS 伺服器.....	185
Kafka Connect 實務範例.....	187
Kafka 連接一般拓撲.....	188
正好一次支援.....	196
注意事項.....	197
程式設計介面資訊.....	198
商標.....	198

IBM MQ 實務範例

每一個實務範例都會逐步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。

下列子主題中說明可用的 IBM MQ 實務範例。

Windows 開始使用 IBM MQ

此實務範例說明如何在 Windows 平台上開始使用 IBM MQ。如果您從未使用 IBM MQ 且想要快速開始使用，請使用此實務範例。

如果您尚未在系統上安裝 IBM MQ，則此實務範例說明在 Windows 上安裝、配置及驗證的基本步驟。您可以使用圖形使用者介面或指令行介面來完成實務範例的步驟。

規劃解決方案

選擇在 Windows 上安裝 IBM MQ 的方法。使用圖形使用者介面及精靈來引導您完成安裝及配置程序，或使用指令行來處理無聲自動安裝。

概觀: 遞送的邏輯拓撲

完成實務範例之後交付的邏輯拓撲。

已安裝的 IBM MQ 伺服器實例容許建立 IBM MQ 物件: 佇列及佇列管理程式。您可以使用 IBM MQ Explorer，透過佇列管理程式從本端佇列中放置及取得訊息。完成此實務範例之後，交付的拓撲將類似於圖 1。

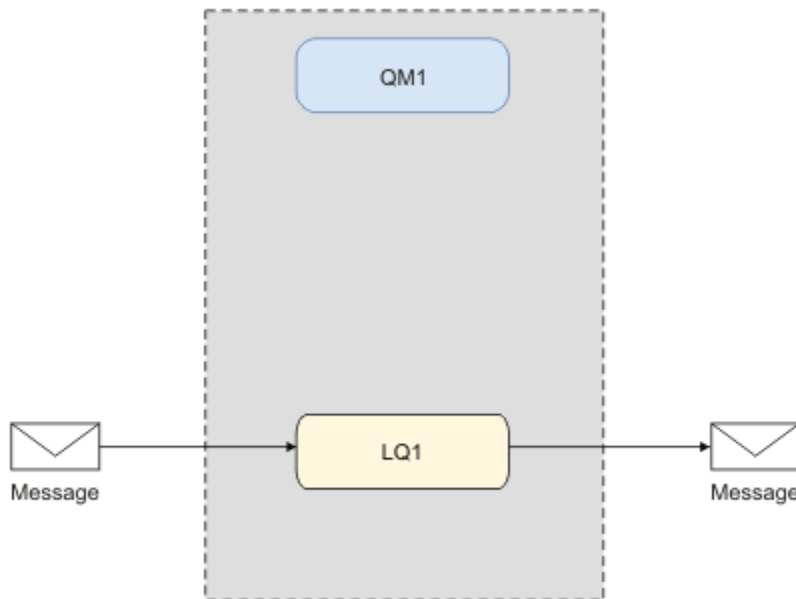


圖 1: 將訊息放置在 LQ1 上，從 LQ1 中取得訊息。

基本概念與重要詞彙

說明您在開始使用 IBM MQ 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 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 MQ 佇列，直到應用程式進行讀取。

在此實務範例中，您可以選擇使用下列其中一種方法來安裝與配置 IBM MQ：

第 6 頁的『使用圖形使用者介面安裝及配置』

使用圖形使用者介面進行安裝期間，系統會透過數個畫面和精靈來引導您，協助您套用相關選項和設定：

啟動程式

請檢查軟體需求，指定網路資訊，然後啟動 IBM MQ 安裝精靈。

IBM MQ 安裝精靈

安裝軟體並啟動「Prepare IBM MQ Wizard」。

Prepare IBM MQ Wizard

啟動 IBM MQ 服務和 IBM MQ Explorer。

IBM MQ Explorer

管理佇列和佇列管理程式。

第 10 頁的『使用指令行介面來安裝及配置』

指令行介面安裝可以是無聲自動安裝或互動式安裝。無聲自動安裝可供完整存取，是此實務範例所涵蓋的內容。使用指令行進行安裝期間，系統會透過數個步驟來引導您套用相關選項和設定：

- 安裝 IBM MQ
- 建立與配置 IBM MQ 物件；佇列管理程式和佇列。
- 驗證安裝，方法是使用 amqsput 來放置訊息，以及使用 amqsget 從佇列中取出訊息。

除了使用「IBM MQ Explorer」和指令行來建立 IBM MQ 物件以外，也能夠使用可程式化介面予以建立。目前的實務範例中不包括此部分內容。

重要詞彙

這是有關訊息佇列作業的重要詞彙清單。

有關訊息佇列作業的重要詞彙。

術語	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。

實作解決方案

實作實務範例的解決方案。使用安裝「啟動程式」在 Windows 上安裝 IBM MQ，然後使用 IBM MQ Explorer 來驗證安裝。

使用圖形使用者介面安裝及配置

使用安裝啟動程式在 Windows 上安裝 IBM MQ，並使用 IBM MQ Explorer 來驗證安裝。驗證安裝之後，請建立佇列管理程式及佇列，然後嘗試將訊息放入佇列並從佇列取得訊息。

使用啟動程式安裝

使用安裝啟動程式，在 Windows 上安裝 IBM MQ。

開始之前

在啟動此作業之前，請完成下列檢查：

- 安裝時必須具有本端管理者權限。透過 Windows 機能來定義此權限。
- 請確定機器名稱不包含任何空格。
- 請確定您有足夠的磁碟空間來完整安裝 IBM MQ for Windows。如需相關資訊，請參閱 [Multiplatforms 上的磁碟空間需求](#)。
- 判定是否需要為任何 IBM MQ 使用者定義 Windows 網域使用者 ID。

在安裝 IBM MQ 之前，請檢查您的系統是否符合軟體需求。如需所有受支援平台上軟體需求的相關資訊，請參閱 [IBM MQ 的系統需求](#)。

關於這項作業

啟動程式及後續精靈會引導您完成安裝程序，並協助您檢閱軟體需求及 IBM MQ 設定。

這項作業假設您是第一次在機器上安裝 IBM MQ，且您將使用預設位置。依預設，IBM MQ 程式檔案的位置為 C:\Program Files\IBM\MQ，資料及日誌檔位置為 C:\ProgramData\IBM\MQ。

註：如果您要安裝 IBM MQ 9.0，且您在機器上已安裝任何先前的 IBM MQ，則程式及資料檔的位置將不同於預設值。如需進一步資訊，請參閱 [程式和資料目錄位置](#)。如果您先前已完成此實務範例，且想要使用預設位置的單一全新安裝來重複它，請先移除先前的安裝，再重新啟動此實務範例。若要從機器解除安裝現有的 IBM MQ 實例，請參閱 [第 15 頁的『解除安裝 IBM MQ』](#)。

安裝程式包含進一步資訊的鏈結 (如果您在安裝程序期間需要的話)。

程序

1. 執行「啟動程式」，檢閱，必要的話，修改軟體需求及網路配置。
 - a) 導覽至 IBM MQ 軟體目錄，然後按兩下檔案 Setup.exe 以執行「啟動程式」。
 - b) 選取 **軟體需求** 標籤，以顯示 **軟體需求** 設定。
 - c) 檢查是否已符合軟體需求，以及需求項目是否顯示帶有 OK 單字的綠色勾號。進行任何指出的更正。

註：
如需任何需求的詳細資料，請按一下勾選框來展開資訊標籤。
 - d) 選取 **網路配置** 標籤，以顯示 **網路配置** 設定。
 - e) 選取否。

註：此實務範例假設您不需要配置 IBM MQ 的網域使用者 ID。如需配置 IBM MQ for Windows 網域使用者的相關資訊，請按一下 [相關資訊](#)。
 - f) 在「啟動程式」的 **IBM MQ 安裝** 標籤上，選取安裝語言，然後按一下 **啟動 IBM MQ 安裝程式** 以啟動 IBM MQ 安裝精靈。

您已完成透過符合或指定安裝需求來設定 IBM MQ，並已啟動 IBM MQ 安裝精靈。
2. 使用 IBM MQ 安裝精靈來安裝軟體，以及啟動 Prepare IBM MQ Wizard。
 - a) 在 IBM MQ 安裝精靈中，閱讀「授權合約」，並按一下 **我接受授權合約中的條款** 勾選框，然後按下一步。
 - b) 按一下 **一般**，然後按 **下一步**。
 - c) 在「**準備安裝 IBM MQ**」頁面中，檢閱安裝資訊，然後按一下 **安裝**。

註：請注意下列詳細資料：

- 安裝名稱
- 程式檔案的最上層資料夾

- 資料檔的最上層資料夾

已安裝下列特性:

- IBM MQ 伺服器
- IBM MQ: 用於管理及監視 IBM MQ 資源的圖形介面
- Java 和 .NET 傳訊及 Web 服務
- IBM MQ 開發工具箱

安裝程序開始。視您的系統而定，安裝程序可能需要數分鐘。

在安裝程序結束時，「IBM MQ 設定」視窗會顯示訊息 **Installation Wizard Completed Successfully**。

d) 按一下**完成**。

您已順利安裝 IBM MQ。Prepare IBM MQ Wizard 會自動啟動，並顯示 **Prepare IBM MQ Wizard** 頁面。

3. 使用 Prepare IBM MQ Wizard 來啟動 IBM MQ 服務。

a) 在「歡迎使用 Prepare IBM MQ Wizard」頁面上，選取**下一步**。

Prepare IBM MQ Wizard 會顯示訊息 **Status: Checking IBM MQ Configuration** 及進度列。當程序完成時，會顯示「IBM MQ 網路配置」頁面。

b) 在 Prepare IBM MQ Wizard 的「IBM MQ 網路配置」頁面上，選取**否**。

c) 按**下一步**。

Prepare IBM MQ Wizard 會顯示訊息 **Status: starting the IBM MQ Service** 及進度列。當程序完成時，精靈會顯示「完成 Prepare IBM MQ Wizard」頁面。

d) 選取 **啟動 IBM MQ 瀏覽器**，並選擇是否檢視版本注意事項，然後按一下 **完成** 按鈕。

即會啟動 IBM MQ Explorer。

您已安裝 IBM MQ。您也已啟動 IBM MQ Explorer。

結果

已安裝並驗證 IBM MQ，且您已準備好配置物件，例如佇列管理程式及佇列。

下一步

請遵循第 8 頁的『[建立稱為 QM1 的佇列管理程式](#)』中的指示。

相關概念

[光碟空間需求](#)

[Windows 系統上的軟硬體需求](#)

[IBM MQ 簡介](#)

相關工作

[在 Windows 上安裝 IBM MQ 伺服器](#)

[配置 IBM MQ 伺服器](#)

建立稱為 QM1 的佇列管理程式

使用 IBM MQ Explorer 來建立稱為 QM1 的佇列管理程式。佇列管理程式是 IBM MQ 傳訊網路中的主要元件。

開始之前

您必須已安裝 IBM MQ。如果沒有，請參閱第 7 頁的『[使用啟動程式安裝](#)』，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 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。

若要使用「IBM MQ Explorer」來建立及啟動佇列管理程式，請完成下列步驟。

程序

1. 以管理者身分啟動 IBM MQ Explorer。
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，IBM 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 9 頁的『建立稱為 LQ1 的佇列』](#)。

相關工作

在 [Multiplatforms](#) 上建立及管理佇列管理程式

建立稱為 LQ1 的佇列

使用 IBM MQ Explorer 來建立佇列。佇列是用來儲存訊息的資料結構，並且是 IBM MQ 佇列管理程式物件。

關於這項作業

在這項作業中，您可以使用 IBM MQ Explorer 來建立 IBM MQ 物件。

若要使用 IBM MQ Explorer 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 9 頁的『將訊息放入佇列 LQ1』](#)。

將訊息放入佇列 LQ1

使用 IBM MQ Explorer 將訊息放入佇列 LQ1。

關於這項作業

這項作業假設您已依照第 12 頁的『[建立稱為 QM1 的佇列管理程式](#)』中的說明，建立一個稱為 QM1 的佇列管理程式，以及依照第 9 頁的『[建立稱為 LQ1 的佇列](#)』中的說明，建立一個稱為 LQ1 的佇列。

若要使用 IBM MQ Explorer 將訊息放入佇列，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 10 頁的『[從佇列 LQ1 取得訊息](#)』。

從佇列 LQ1 取得訊息

使用 IBM MQ Explorer 從佇列 LQ1 取得訊息。

關於這項作業

此作業假設您已如第 9 頁的『[將訊息放入佇列 LQ1](#)』中所述放置訊息 QM1。

若要使用 IBM 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，用滑鼠右鍵按一下本端佇列 LQ1，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

遵循後續實務範例中的指示，以進一步探索 IBM MQ 特性。

若要瞭解如何撰寫佇列應用程式、連接佇列管理程式、發佈/訂閱以及開啟和關閉物件，請參閱 [撰寫佇列作業的程序化應用程式](#)。

使用指令行介面來安裝及配置

使用指令行在 Windows 上安裝 IBM MQ，以執行無聲自動安裝並設定環境變數。驗證安裝之後，請建立佇列管理程式及佇列，然後嘗試將訊息放入佇列並從佇列取得訊息。

使用無聲自動安裝來安裝

使用指令行在 Windows 上安裝 IBM MQ，以執行無聲自動安裝，並確認已正確設定安裝的環境。

開始之前

在開始這項作業之前，請先完成下列檢查：

- 安裝時必須具有本端管理者權限。透過 Windows 機能來定義此權限。
- 請確定機器名稱不包含任何空格。
- 請確定您有足夠的磁碟空間。如需相關資訊，請參閱 [Multiplatforms 上的磁碟空間需求](#)。
- 決定您是否需要為任何 IBM MQ 使用者定義 Windows 網域使用者 ID。

在安裝 IBM MQ 之前，請檢查您的系統是否符合軟硬體需求。如需所有受支援平台上軟硬體需求的最新詳細資料，請參閱 [IBM MQ 的系統需求](#)。

關於這項作業

此實務範例假設您是第一次在機器上安裝 IBM MQ，且您使用預設位置。依預設，IBM MQ 9.0 程式檔案的位置為 C:\Program Files\IBM\MQ，資料及日誌檔位置為 C:\ProgramData\IBM\MQ。

註：如果您在機器上具有任何先前安裝的 IBM MQ，則程式及資料檔的預設位置可能會變更。如需進一步資訊，請參閱 [程式和資料目錄位置](#)。如果您先前已完成此實務範例，且想要使用預設位置的單一全新安裝來重複它，請先移除先前的安裝，再重新啟動此實務範例。若要從機器解除安裝現有的 IBM MQ 實例，請參閱第 15 頁的『解除安裝 IBM MQ』。

Windows 上的 IBM MQ 使用 MSI 技術來安裝軟體。如需使用 MSI 技術進行安裝的相關資訊，請參閱 [使用 msixexec 進行進階安裝](#)。

若要使用指令行來安裝 IBM MQ，您必須指定下列參數：

- /i "MQ_INSTALLATION_MEDIA\MSI\IBM MQ.msi" 其中 MQ_INSTALLATION_MEDIA 是 IBM MQ.msi 檔案的位置。此引數指定 .msi 檔案的位置。
- /l*v USER_LOGFILE_LOCATION\install.log 其中 USER_LOGFILE_LOCATION 是您要寫入安裝日誌的位置。

註：在執行指令之前，您要在其中建立 install.log 的資料夾必須存在。

- /q[n|b|r|f] /q 必須與 n、b、r 或 f 其中之一配對。在命令提示字元執行 **msiexec** 指令會開啟說明檔，其中顯示正確的用法。
- USEINI="RESPONSE_FILE" 其中 RESPONSE_FILE 是無聲自動安裝要使用之回應檔的名稱和位置。此實務範例使用範例 Response.ini 檔案，該檔案包含在 IBM MQ 安裝媒體中。
- TRANSFORMS="TRANSFORM_FILE" 其中 TRANSFORM_FILE 是要套用至安裝的轉換檔案名稱。此實務範例使用美式英文轉換 1033.mst。
- AGREETOLICENSE="YES" 必須包含此參數，否則安裝無法完成。
- ADDLOCAL="Server" 此參數列出要安裝的元件。

程序

1. 使用指令行來處理無聲自動安裝。

- a) 若要從提升的命令提示字元呼叫無聲自動安裝，請按一下 **Windows 工作列** 上的 **啟動按鈕**，然後在 **搜尋程式及檔案** 欄位中鍵入 cmd。用滑鼠右鍵按一下 **cmd.exe** 程式，然後選取 **以管理者身分執行**。
- b) 在 Windows 命令提示字元中，輸入下列指令：

註：此指令在這裡以多行呈現，但必須在一行上鍵入。

```
msiexec /i "MQ_INSTALLATION_MEDIA\MSI\IBM MQ.msi"  
/l*v c:\wmi\logs\install.log  
/q USEINI="MQ_INSTALLATION_MEDIA\Response.ini"  
TRANSFORMS="1033.mst"  
AGREETOLICENSE="yes"  
ADDLOCAL="Server"
```

其中 MQ_INSTALLATION_MEDIA 是 IBM MQ 安裝媒體的路徑。

註: 在執行指令之前，您要在其中建立 `install.log` 的資料夾必須存在。

輸入指令之後，指令行會傳回提示。

- c) 若要檢視安裝進度，請開啟您指定的日誌檔。如果安裝順利完成，您會從日誌檔底端向上看到訊息 `Product: IBM MQ (Installation1) -- Installation operation completed successfully`. 兩個段落。
 - d) 安裝完成時，服務會啟動，且 IBM MQ 圖示會出現在系統匣中。
您已安裝 IBM MQ，且已啟動 IBM MQ 服務。
2. 使用 `setmqenv` 指令來設定安裝的環境變數。
- a) 在指令行中輸入下列指令：

```
"MQ_INSTALLATION_PATH/bin/setmqenv" -s
```

其中 `MQ_INSTALLATION_PATH` 是指 IBM MQ 的安裝位置。請確定您在 `bin` 資料夾中以引號括住 `setmqenv` 的路徑，以防止提示傳回錯誤。

註: 如果您使用預設位置，則安裝路徑會是 `C:\Program Files\IBM\MQ`。

- b) 輸入下列指令，以檢查環境是否正確設定：

```
dspmqr
```

如果指令順利完成，並傳回預期的版本號碼及安裝名稱，則會正確設定環境。訊息應該包含下列這一行：

```
Version: n.n.n.n
```

其中 `n.n.n.n` 是版本號碼，如果您未指定非預設安裝名稱，則這一行：

```
InstName: Installation1
```

您已使用無聲自動安裝順利安裝 IBM MQ。

結果

您已執行 IBM MQ 無聲自動安裝，並確認您的環境已正確設定。

下一步

- 您可以執行 [Prepare IBM MQ Wizard](#)。
- 請遵循第 12 頁的『[建立稱為 QM1 的佇列管理程式](#)』中的指示。

如果您在安裝期間遇到任何問題，請檢查安裝日誌，在您於 `msiexec` 指令中指定的位置，在此實務範例中，日誌檔的位置為：`c:\wmqinslogs\install.log`。請採取日誌中指定的任何動作，然後重新執行安裝。您也可以檢查您使用指令傳遞的參數，遮罩確定您包括所有必要的參數。

相關工作

[使用 msiexec 安裝伺服器](#)

[搭配使用轉換與 msiexec](#)

[安裝 IBM MQ -概觀](#)

建立稱為 QM1 的佇列管理程式

使用指令行介面建立佇列管理程式 QM1。佇列管理程式是 IBM MQ 傳訊網路中的主要元件。

開始之前

您必須已安裝 IBM MQ。如果沒有，請參閱 [第 10 頁的『使用無聲自動安裝來安裝』](#)，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 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。

程序

1. 以管理者身分開啟命令提示字元。
2. 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
IBM 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。

3. 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
IBM 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'.
IBM 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 13 頁的『建立稱為 LQ1 的佇列』](#)。

相關工作

在 [Multiplatforms](#) 上建立及管理佇列管理程式

建立稱為 LQ1 的佇列

使用指令行介面來建立佇列。佇列是用來儲存訊息的資料結構，並且是 IBM 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 MQ 物件：

- 指令行。
- IBM MQ Explorer。

- 使用可程式化介面。

在此作業中，您可以使用指令行建立 IBM MQ 物件。

指令行介面具有稱為「IBM 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: IBM 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 14 頁的『將訊息放入佇列 LQ1』](#)。

將訊息放入佇列 LQ1

使用指令行介面將訊息放入佇列 LQ1。

關於這項作業

IBM 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。

若要使用指令行介面將訊息放入佇列，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 15 頁的『從佇列 LQ1 取得訊息』](#)。

從佇列 LQ1 取得訊息

使用指令行介面從佇列 `LQ1` 取得訊息。

關於這項作業

IBM 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。

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 AMQSGET0 start
message <Hello World>
no more messages
Sample AMQSGET0 end
```

`amqsget` 應用程式在讀取訊息 30 秒之後結束。

下一步

遵循後續實務範例中的指示，以進一步探索 IBM MQ 特性。

若要瞭解如何撰寫佇列應用程式、連接佇列管理程式、發佈/訂閱以及開啟和關閉物件，請參閱 [撰寫佇列作業的程序化應用程式](#)。

解除安裝 IBM MQ

停止，然後解除安裝 IBM MQ，包括移除任何佇列管理程式及其物件。在此作業結束時，您已準備好重新安裝 IBM MQ。

關於這項作業

此作業說明使用下載的安裝映像檔在 Windows 上解除安裝 IBM MQ 的步驟。

「入門」實務範例會引導您完成使用啟動程式或指令行來安裝 IBM MQ 的選項。雖然您可以有多個 IBM MQ 安裝架構，但此實務範例是以單一伺服器上的新安裝架構為基礎。因此，如果您想要重複此實務範例，或嘗試不同的安裝方法，則必須先解除安裝現有的 IBM MQ 元件 (包括任何現有的佇列管理程式及其物件)，以便您可以使用全新安裝重新啟動。

您也可能需要解除安裝，以便針對本節中的部分其他實務範例執行全新安裝。

程序

1. 停止 IBM MQ 服務。

a) 用滑鼠右鍵按一下系統匣中的 **IBM MQ** 圖示，然後按一下 **停止 IBM MQ**，以停止 IBM MQ 服務。

即會顯示具有下列訊息的對話框：

關閉 IBM MQ 安裝 "Installation1" 會終止所有執行中的佇列管理程式及該安裝的 IBM MQ 程序，但受 Microsoft Failover Cluster 控制的程序除外。您確定要繼續嗎？

b) 按一下 **是**，然後等待 IBM MQ 停止。

c) 當 IBM MQ 停止時，用滑鼠右鍵按一下系統匣中的 **IBM MQ** 圖示，然後按一下 **結束**。

2. 開始解除安裝程序。

下載包含安裝映像檔的壓縮檔，然後將它解壓縮至暫存目錄。導覽至該目錄，然後按兩下 `setup.exe`。

即會顯示「**IBM MQ 安裝啟動程式**」視窗。

3. 移除 IBM MQ。

a) 按一下 **IBM MQ 安裝**。

b) 按 **啟動 IBM MQ 安裝程式**，然後按 **下一步**，直到顯示 **IBM MQ 程式維護窗格** 並顯示歡迎使用訊息為止。

如果未顯示此窗格，則目前未安裝 IBM MQ for Windows。

c) 按一下 **維護或升級現有實例**。選取 **Installation1** 來移除它。按 **下一步**，然後在「**程式維護窗格**」中按一下 **移除**，然後按 **下一步**。

即會顯示「**移除伺服器**」功能窗格。

d) 選取 **移除**：移除現有的佇列管理程式及其物件。

按 **下一步**。

即會顯示「**移除 IBM MQ**」窗格，其中包含要移除的安裝摘要。

e) 按一下 **移除** 以繼續。

如果出現訊息指出找到已鎖定的檔案，請確定沒有 IBM MQ 程式在執行中；請參閱 [在 Windows 系統上解除安裝 IBM MQ](#)。

解除安裝 IBM MQ 時，會出現一則訊息指出完成。

f) 按一下 **完成**。


您已順利解除安裝 IBM MQ。

相關工作

[在 Windows 系統上解除安裝 IBM MQ](#)

下一步

完成「開始使用 IBM MQ」實務範例時接下來要執行的動作。

如需協助您安裝及升級的指導教學，請參閱 [在 AIX、Linux® 及 Windows 上安裝及升級 IBM MQ 的指導教學集合](#)。指導教學涵蓋：

- 為 IBM MQ 準備主機。
- 下載 IBM MQ 程式碼。

- 安裝及解除安裝 IBM MQ 程式碼，並套用修正套件。
- 從 IBM MQ 的一個版本升級至另一個版本，並將佇列管理程式從一個主機移至另一個主機。

您還可以在 IBM MQ 產品說明文件中檢視其他主題。您可能想要查看下列區段：

- [管理 IBM MQ](#)

IBM 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。

- [使用 MQSC 指令管理 IBM MQ](#)

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 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，並向您顯示如何使用它們。這些指令在 [MQSC 指令](#) 參照小節中按字母順序詳細說明。

- [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。

點對點實務範例

在點對點拓撲中連接兩個 IBM MQ 佇列管理程式，以啟用分散式佇列作業。

關於這項作業

建立兩個佇列管理程式及適當的佇列和通道，以建立單向點對點傳訊基礎架構。在個別主機上建立佇列管理程式，以啟用透過網路進行通訊。作為實務範例的延伸，請將「傳輸層安全」新增至通道，以啟用安全的資料通訊。

規劃解決方案

點對點傳訊是 IBM MQ 中最簡單的傳訊形式。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特定資訊，才能傳送訊息。傳送端應用程式將需要定址遠端佇列的方法。使用點對點傳訊，將訊息傳送至具有範例應用程式的遠端佇列管理程式。

概觀: 遞送的邏輯拓撲

完成實務範例之後交付的邏輯拓撲。

點對點基礎架構容許在不同主機上的佇列管理程式之間進行單向傳訊。主機上的佇列管理程式 1，會將訊息傳送至主機 2 上的佇列管理程式 2。完成此實務範例之後，交付的拓撲將類似於 [圖 1](#)。

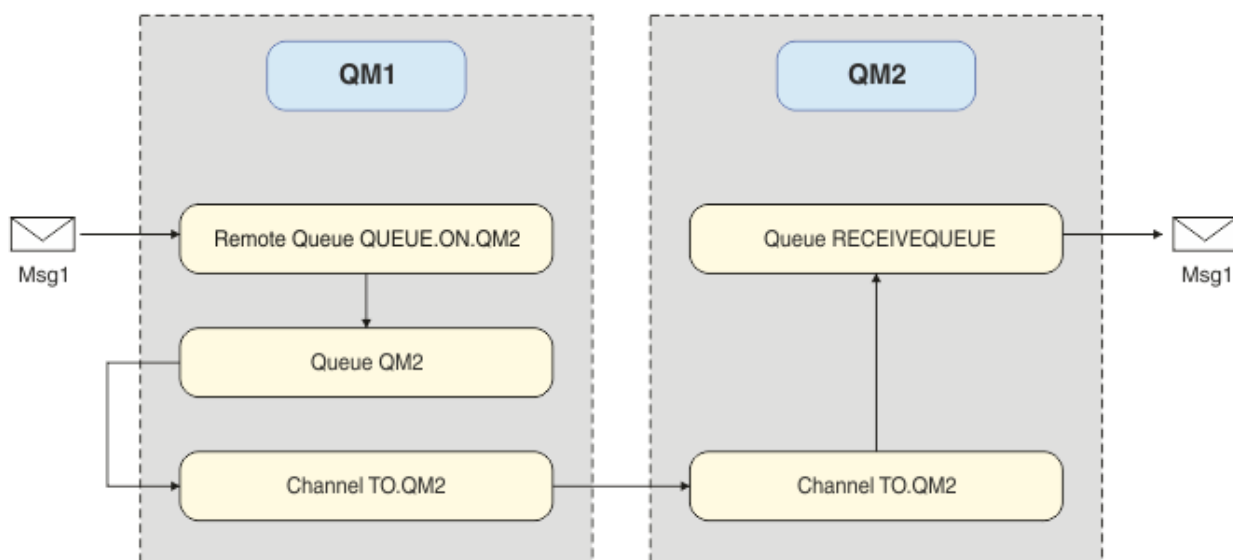


圖 2: QM1 將訊息傳送至 QM2

基本概念與重要詞彙

為了完成點對點實務範例，您必須知道的基本概念和重要術語的說明。

基本概念

IBM 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 MQ 佇列，直到應用程式進行讀取。

重要詞彙

這是有關訊息佇列作業的重要詞彙清單。

有關訊息佇列作業的重要詞彙。

術語	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。

實作解決方案

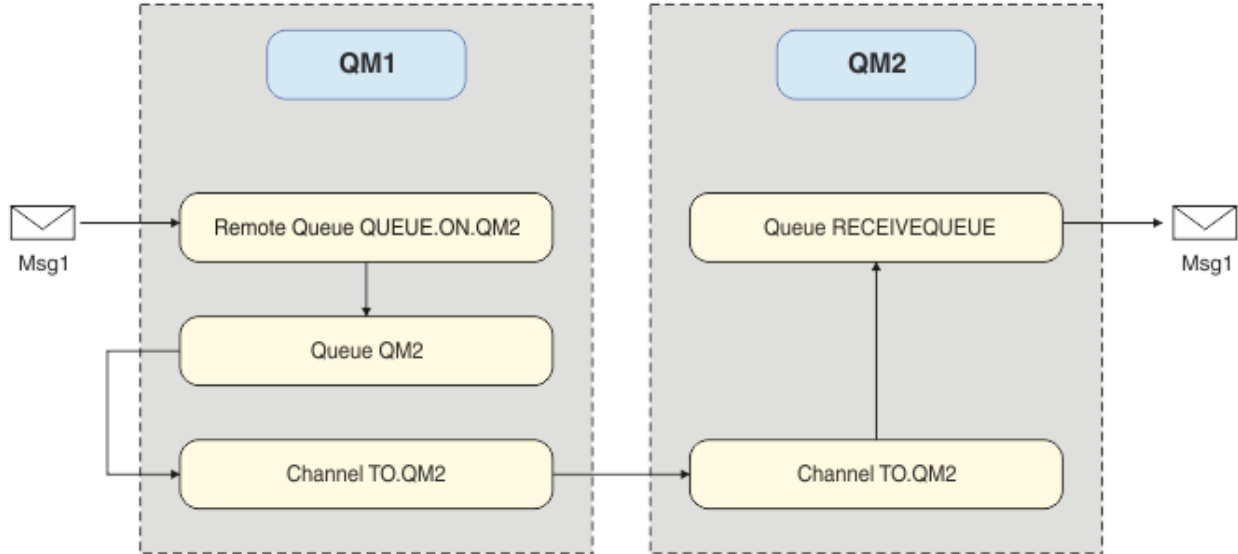
實作實務範例的解決方案。在兩個個別主機上建立兩個 IBM MQ 佇列管理程式，來源佇列管理程式用來傳送訊息，目標佇列管理程式用來接收訊息。

開始之前

此實務範例的起點是現有且已驗證的 IBM MQ 安裝。如需安裝 IBM MQ 的指示，請遵循 [在 Windows 上安裝 IBM MQ 伺服器](#) 中的步驟。

關於這項作業

使用指令行介面來建立兩個佇列管理程式，並定義必要的接聽器、佇列及通道。交付的邏輯拓撲會顯示實作解決方案所新增的功能。



建立佇列管理程式

建立 IBM MQ 佇列管理程式，以將訊息傳送至目標佇列管理程式。

開始之前

- 您必須已安裝 IBM MQ。如需安裝 IBM MQ 的相關資訊，請參閱 [安裝及解除安裝](#)。

關於這項作業

使用指令行介面來建立 IBM MQ 佇列管理程式。

程序

1. 建立名稱為 QM1 的佇列管理程式。在指令行上，鍵入：

```
crtmqm QM1
```

會顯示下列訊息，以確認已建立佇列管理程式：

```
IBM MQ queue manager created.  
Creating or replacing default objects for QM1.  
Default objects statistics : 61 created. 0 replaced. 0 failed.  
Completing setup.  
Setup completed.
```

2. 啟動佇列管理程式。在指令行上，鍵入：

```
strmqm QM1
```

會顯示下列訊息，以確認佇列管理程式已啟動：

```
IBM 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'.  
IBM MQ queue manager 'QM1' started.
```

結果

即會建立並啟動 IBM MQ 佇列管理程式 QM1。

下一步

若要建立與 QM1 搭配使用的佇列，請遵循 [第 20 頁的『建立佇列』](#) 中的指示。

建立佇列

建立 IBM MQ 佇列管理程式所管理的 IBM MQ 佇列。

開始之前

您必須具有如 [第 19 頁的『建立佇列管理程式』](#) 中所述設定的 IBM MQ 佇列管理程式。

關於這項作業

啟動 **MQSC** 介面，以管理連接至佇列管理程式的物件。建立傳輸佇列及遠端佇列定義。結束 **MQSC** 介面。

程序

1. 在指令行上，鍵入：

```
runmqsc QM1
```

在確認訊息之後，工具已準備好接受指令。

2. 建立稱為 QM2 的傳輸佇列。最好為傳輸佇列提供與遠端佇列管理程式相同的名稱。在 MQSC 介面中，鍵入：

```
DEFINE QLOCAL(QM2) DESC('Transmission queue to QM2') USAGE(XMITQ)
```

即會建立傳輸佇列。

3. 建立稱為 QUEUE.ON.QM2。遠端佇列定義必須參照提供給遠端主機上本端佇列的名稱。在 MQSC 介面中，鍵入：

```
DEFINE QREMOTE(QUEUE.ON.QM2) DESC('Remote queue for QM2') XMITQ(QM2) RNAME(RECEIVEQUEUE)  
RQMNAME(QM2)
```

已建立遠端佇列定義。

4. 鍵入 end 以結束 MQSC 介面。

下一步

若要建立用來連接至目標佇列管理程式的傳送端通道，請遵循 [第 20 頁的『建立傳送端通道』](#) 中的指示。

建立傳送端通道

在來源佇列管理程式上建立傳送端通道，該通道用來連接至目標佇列管理程式。

開始之前

若要建立使用 TLS 的通道，請遵循 [第 26 頁的『建立通道以使用 TLS』](#) 中的指示。如果您想要在沒有 TLS 安全的情況下測試解決方案，則可以在之後執行此動作。

關於這項作業

啟動 **MQSC** 介面，以管理連接至佇列管理程式的物件，並建立傳送端通道。此通道用來連接至稱為 QM2 的目標佇列管理程式。

程序

1. 在指令行上，鍵入：

```
runmqsc QM1
```

在確認訊息之後，工具已準備好接受指令。

2. 建立傳送端通道，稱為 TO.QM2。在 MQSC 介面中，鍵入：

```
DEFINE CHANNEL(TO.QM2) CHLTYPE(SDR) CONNAME(' remoteHost ') TRPTYPE(TCP) XMITQ(QM2)
```

註：變數 *remoteHost* 是目標佇列管理程式的主機名稱或 IP 位址。

已建立傳送端通道。

下一步

若要建立分散式佇列管理程式拓撲，請遵循第 21 頁的『[建立分散式佇列管理程式拓撲](#)』中的指示。

建立分散式佇列管理程式拓撲

點對點傳訊是 IBM MQ 中最簡單的傳訊形式。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特定資訊，才能傳送訊息。傳送端應用程式將需要定址遠端佇列的方法。使用點對點傳訊，將訊息傳送至具有範例應用程式的第二個佇列管理程式。

開始之前

您必須已依照第 19 頁的『[建立佇列管理程式](#)』中的說明來設定來源佇列管理程式。

關於這項作業

在遠端主機上建立目標佇列管理程式。使用範例應用程式來驗證來源與目標佇列管理程式之間的通訊。

建立佇列管理程式

建立 IBM MQ 佇列管理程式，以接收來自遠端佇列管理程式的訊息。

開始之前

您必須已安裝 IBM MQ。如需安裝 IBM MQ 的相關資訊，請參閱在 [Windows 上安裝 IBM MQ 伺服器](#)。

關於這項作業

使用指令行介面來建立 IBM MQ 佇列管理程式。

程序

1. 建立名稱為 QM2 的佇列管理程式。在指令行上，鍵入：

```
crtmqm QM2
```

會顯示下列訊息：

```
IBM MQ queue manager created.  
Creating or replacing default objects for QM2.  
Default objects statistics : 61 created. 0 replaced. 0 failed.
```

```
Completing setup.  
Setup completed.
```

2. 啟動佇列管理程式。在指令行上，鍵入：

```
strmqm QM2
```

會顯示下列訊息，以確認佇列管理程式已啟動：

```
IBM MQ queue manager 'QM2' starting.  
5 log records accessed on queue manager 'QM2' during the log replay phase.  
Log replay for queue manager 'QM2' complete.  
Transaction manager state recovered for queue manager 'QM2'.  
IBM MQ queue manager 'QM2' started.
```

結果

即會建立並啟動 IBM MQ 佇列管理程式 QM2。

下一步

若要建立佇列以與 QM2 搭配使用，請遵循 [第 22 頁的『建立佇列』](#) 中的指示。

建立佇列

建立用來在目標佇列管理程式上接收訊息的本端佇列，以及接受入埠通道連線的接聽器。

關於這項作業

啟動 **runmqsc** Scripting 工具之後，您可以使用 MQSC 指令來建立本端佇列和接聽器。

程序

1. 輸入下列指令來啟動 Scripting 工具：

```
runmqsc QM2
```

畫面上會顯示一則訊息，確認工具已啟動。

2. 建立稱為 RECEIVEQUEUE 的本端佇列。佇列必須與來源佇列管理程式上遠端佇列定義中所參照的名稱相同。在 MQSC 介面中，鍵入：

```
DEFINE QLOCAL(RECEIVEQUEUE) DESCR('Receiving queue')
```

已建立本端佇列。

3. 建立稱為 LISTENER1 的接聽器。在 MQSC 介面中，鍵入：

```
DEFINE LISTENER(LISTENER1) TRPTYPE(TCP) PORT(1414) CONTROL(QMGR)
```

註：埠 1414 是 IBM MQ 的預設埠。如果您選擇不同的埠號，則必須將它新增至傳送端佇列管理程式上傳送端通道的 CONNAME。

4. 啟動接聽器，以便它準備好接受入埠連線。在 MQSC 介面中，鍵入：

```
START LISTENER(LISTENER1)
```

註：因為接聽器是使用選項 CONTROL (QMGR) 建立的，所以下次啟動佇列管理程式時，也會自動啟動接聽器。

5. 鍵入 end 以結束 MQSC 介面。

下一步

若要建立接收端通道以建立來源與目標佇列管理程式之間的連線，請遵循 [第 23 頁的『建立接收端通道』](#) 中的指示。

建立接收端通道

建立目標佇列管理程式的接收端通道，以啟用來源與目標佇列管理程式之間的通訊。

開始之前

若要建立使用 TLS 的通道，請遵循 [第 26 頁的『建立通道以使用 TLS』](#) 中的指示。如果您想要在沒有 TLS 安全的情況下測試解決方案，則可以在之後執行此動作。

關於這項作業

使用 **MQSC** 介面來建立 QM2 所管理的接收端通道。

程序

1. 在指令行上，鍵入：

```
runmqsc QM2
```

在確認訊息之後，工具已準備好接受指令。

2. 建立稱為 TO.QM2。通道名稱必須與來源佇列管理程式上的傳送端通道名稱相同。在 MQSC 介面中，鍵入：

```
DEFINE CHANNEL(TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP)
```

已建立接收端通道。

下一步

若要在來源佇列管理程式上啟動傳送端通道，進而在目標佇列管理程式上起始接收端通道，請遵循 [第 23 頁的『啟動傳送端通道』](#) 中的指示。

啟動傳送端通道

啟動來源佇列管理程式上的傳送端通道，也會啟動目標佇列管理程式上的接收端通道。訊息可以從來源佇列管理程式傳送至目標佇列管理程式。

關於這項作業

啟動 **MQSC** 介面，以管理連接至佇列管理程式的物件。啟動傳送端通道以連接至目標佇列管理程式，並啟用通訊。當啟動來源通道時，接收端通道會自動啟動。

程序

1. 在指令行上，鍵入：

```
runmqsc QM1
```

在確認訊息之後，工具已準備好接受指令。

2. 在來源佇列管理程式上啟動傳送端通道。在 MQSC 介面中，鍵入：

```
START CHANNEL(TO.QM2)
```

傳送端通道會啟動，目標佇列管理程式上的接收端通道也會啟動。

3. 請檢查通道是否在執行中。在 MQSC 介面中，鍵入：

```
DISPLAY CHSTATUS(TO.QM2)
```

如果通道正在執行中，您會看到它報告 STATUS (RUNNING)。如果它報告 STATUS 中的任何其他值，請檢查 [錯誤日誌](#)。

下一步

若要驗證來源佇列管理程式可以將訊息傳送至目標佇列管理程式，請遵循 [第 24 頁的『驗證解決方案』](#) 中的指示。

驗證解決方案

請驗證來源佇列管理程式可以將訊息放入遠端佇列。請驗證目標佇列管理程式可以從佇列取得訊息。

關於這項作業

使用範例應用程式 **amqsput** 和 **amqsget** 來驗證解決方案。

程序

1. 從來源佇列管理程式傳送訊息至目標佇列管理程式 QM2。

- a) 在指令行介面中，鍵入：

```
amqsput QUEUE.ON.QM2 QM1
```

您必須使用遠端佇列定義的名稱，將訊息傳送至目標佇列管理程式。

此時會顯示下列訊息：

```
Sample AMQSPUT0 start  
target queue is QUEUE.ON.QM2
```

- b) 鍵入 Hello world.，按兩次 Enter 鍵。
2. 取得目標佇列管理程式上的訊息。

- a) 在指令行介面中，鍵入：

```
amqsget RECEIVEQUEUE QM2
```

此時會顯示下列訊息：

```
Sample AMQSGET0 start  
message <Hello world.>  
no more messages  
Sample AMQSGET0 end
```

結果

目標佇列管理程式收到來自來源佇列管理程式的訊息，確認已達到點對點通訊。

下一步

如果您要將安全新增至解決方案，請遵循 [第 24 頁的『保護點對點拓撲』](#) 中的指示。

保護點對點拓撲

保護點對點拓撲的安全，以便在正式作業環境中傳輸訊息。

關於這項作業

保護來源和目標佇列管理程式物件的安全，以便授與正確的存取層次。定義哪些使用者群組具有佇列及佇列管理程式的存取權。使用數位簽章憑證來使用「傳輸層安全 (TLS)」進行連接，以保護網路連線的安全。

保護來源佇列管理程式物件的安全

設定來源佇列管理程式上物件的授權值。

關於這項作業

使用 **setmqaut** 指令，將權限授與執行應用程式的使用者群組。

程序

1. 若要授與指定的使用者群組對佇列管理程式的 *connect* 權限，請在指令行介面上鍵入：

```
setmqaut -m QM1 -t qmgr -g userGroup +connect
```

2. 若要將遠端佇列定義上的 *put* 權限授與指定的使用者群組，請在指令行介面上鍵入：

```
setmqaut -m QM1 -t q -n "QUEUE.ON.QM2" -g userGroup +put
```

保護目標佇列管理程式物件的安全

設定目標佇列管理程式上物件的授權值。

關於這項作業

使用 **setmqaut** 指令，將權限授與執行應用程式的使用者群組。

程序

1. 若要授與指定的使用者群組對佇列管理程式的 *connect* 權限，請在指令行介面上鍵入：

```
setmqaut -m QM2 -t qmgr -g userGroup +connect
```

2. 若要在遠端佇列定義上授與指定的使用者群組 *get* 權限，請在指令行介面中鍵入：

```
setmqaut -m QM2 -t q -n "RECEIVEQUEUE" -g userGroup +get
```

保護網路安全

保護來源與遠端佇列管理程式之間的網路連線安全。

關於這項作業

使用已簽章的憑證來驗證來源及遠端佇列管理程式的確實性。使用 TLS 網路來傳送訊息以加密訊息。

準備佇列管理程式以使用 TLS

IBM MQ 佇列管理程式的金鑰儲存庫用來儲存佇列管理程式的個人憑證及公用憑證管理中心 (CA) 憑證。來自 IBM MQ 佇列管理程式的個人憑證申請必須由 CA 簽署，其他實體會使用公用憑證來鑑別 IBM MQ 佇列管理程式。

開始之前

您必須在檔案中具有公用憑證管理中心憑證。

關於這項作業

建立 IBM MQ 佇列管理程式的金鑰儲存庫，匯入憑證管理中心的簽章者憑證，並建立佇列管理程式的個人憑證申請。

程序

1. 為稱為 key.kdb 的佇列管理程式建立 CMS 金鑰儲存庫檔。導覽至 Qmgrs\QM1\ssl 目錄，並在指令行上鍵入：

```
runmqckm -keydb -create -db key.kdb -pw passw0rd -type cms -stash
```

註：在這個簡單範例中，我們使用了 passw0rd 密碼。您可能想要選擇不同的密碼，並變更下列每一個指令，以改用您自己的密碼。

2. 在指令行上，將您在檔案中擁有的 CA 憑證新增至金鑰儲存庫，鍵入：

```
runmqckm -cert -add -file CA-certificate-file -db key.kdb -pw passw0rd -label TrustedCA
```

3. 要求將寫入要求檔案 QM1req.req 的個人憑證。

在指令行上，輸入：

```
runmqckm -certreq -create -db key.kdb -pw passw0rd -label ibmwebspheremqmq1  
-dn CN="QM1" -size 1024 -file QM1req.req  
-sig_alg SHA1withRSA
```

此範例中顯示預設憑證標籤名稱。您可以設定自己的名稱 (如果您喜歡的話)。如需詳細資料，請參閱 [數位憑證標籤](#)。

4. 將憑證申請檔案傳送至 CA，他們會發出數位簽署憑證。將已接收且已簽章的憑證檔放在適當的位置，以接收到佇列管理程式的金鑰儲存庫中。
5. 將已簽署的個人憑證接收到佇列管理程式的金鑰儲存庫中。

```
runmqckm -cert -receive -file Signed-certificate-file -db key.kdb -pw passw0rd -format ascii
```

6. 針對每一個佇列管理程式完成下列步驟，並相應地變更佇列管理程式名稱。

下一步

若要啟用透過傳送端及接收端通道的安全通訊，請遵循 [第 26 頁的『建立通道以使用 TLS』](#) 中的指示。

建立通道以使用 TLS

建立使用 TLS 來建立連線的新通道。

開始之前

若要透過使用 TLS 的通道進行通訊，首先您必須具有連線每一端所需的憑證。若要建立必要的憑證，請遵循 [第 25 頁的『準備佇列管理程式以使用 TLS』](#) 中的指示。

關於這項作業

使用 MQSC 介面來定義已設定 TLS 屬性的通道。即使您在先前透過使用 REPLACE 關鍵字步驟中定義通道而不使用 TLS，也可以完成此作業。

程序

1. 在指令行上，鍵入：

```
runmqsc QM1
```

2. 在 QM1 上建立傳送端通道，稱為 T0.QM2，在 MQSC 介面中，鍵入：

```
DEFINE CHANNEL(TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP)
CONNNAME('remoteHost') XMITQ(QM2)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
DESCR('Sender channel using TLS from QM1 to QM2')
REPLACE
```

註：變數 *remoteHost* 是目標佇列管理程式的主機名稱或 IP 位址。

您可以指定通道的 CERTLABL 屬性。如果您這麼做，它必須符合您先前在 第 25 頁的『準備佇列管理程式以使用 TLS』的步驟 3 中執行之 **runmqckm** 指令的 **-label** 參數值。如需憑證標籤的相關資訊，請參閱 [數位憑證標籤](#)，瞭解需求。

3. 鍵入 end 以結束 MQSC 介面。
4. 在指令行上，鍵入：

```
runmqsc QM2
```

5. 在 QM2(稱為 TO.QM2，在 MQSC 介面中，鍵入：

```
DEFINE CHANNEL(TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256) SSLCAUTH(REQUIRED)
DESCR('Receiver channel using TLS from QM1 to QM2')
REPLACE
```

6. 鍵入 end 以結束 MQSC 介面。

下一步

若要驗證來源佇列管理程式可以使用 TLS 將訊息傳送至目標佇列管理程式，請遵循 第 24 頁的『驗證解決方案』中的指示。

V 9.3.0 串流佇列

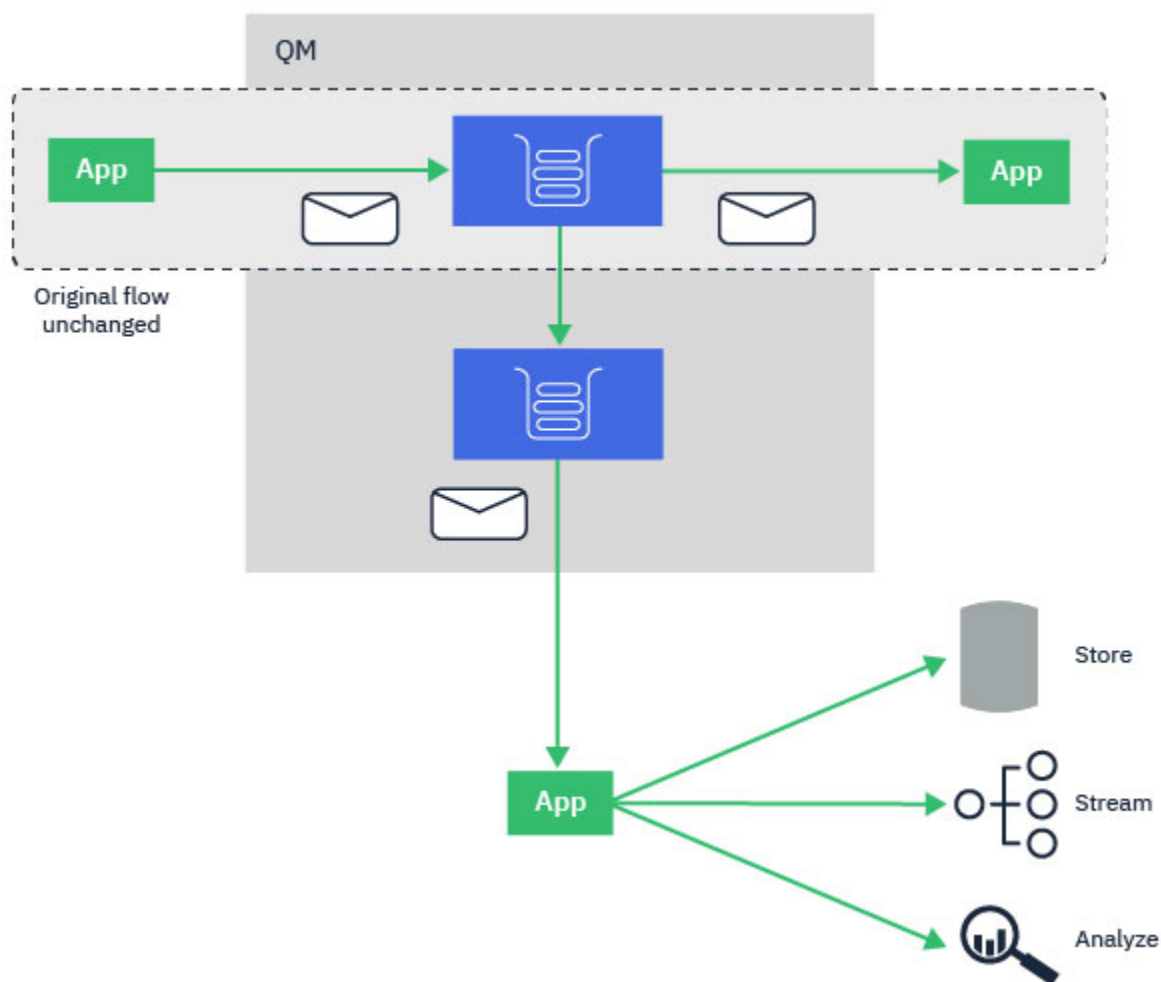
IBM MQ 的串流佇列功能可讓您配置佇列，將每一則訊息幾乎相同的副本放置到第二個佇列。

在您需要建立訊息副本的特定實務範例中，串流佇列可能很有用。例如：

- 使用 IBM MQ 的 Kafka Connect 來源連接器將訊息串流至 Apache Kafka。如需相關資訊，請參閱 [kafka_connect_mq_source](#)。
- 對通過系統的資料執行分析。
- 儲存稍後回復的訊息。
- 擷取一組要在開發及測試系統中使用的訊息。
- 耗用來自系統事件佇列的 IBM MQ 事件訊息，並將其他副本傳送至其他佇列或主題。

在所有這些實務範例中，您可以配置串流佇列，以確保原始訊息不受串流處理程序影響。這可確保核心商業應用程式不會觀察到串流的任何影響。

下列圖解說明：



相關概念

[串流佇列安全](#)

[串流佇列及 AMS](#)

V 9.3.0 串流佇列配置

IBM MQ 的串流佇列特性由管理者在個別佇列上配置，而訊息是由佇列管理程式而非應用程式本身進行串流處理。

這表示在幾乎所有情況下，應用程式將訊息放入原始佇列完全不知道正在進行串流。同樣地，從原始佇列耗用訊息的應用程式並不知道已發生訊息串流。

註： IBM MQ 用戶端程式庫的版本不需要升級即可使用串流佇列，且串流處理程序完全不會變更原始訊息。

您可以兩種模式之一來配置串流佇列：

最大努力

在此模式中，佇列管理程式會認為原始訊息的遞送不受串流訊息的遞送影響更為重要。

如果原始訊息可以遞送，但串流訊息無法遞送，則原始訊息仍會遞送至其佇列。此模式最適合那些應用程式，在這些應用程式中，原始商業應用程式必須不受串流處理程序影響。

必須重複

在此模式下，佇列管理程式會確保原始訊息及串流訊息都順利遞送至其佇列。

如果基於某些原因，串流訊息無法遞送至其佇列 (例如，因為第二個佇列已滿)，則原始訊息也不會遞送至其佇列。放置應用程式收到錯誤原因碼，必須重新嘗試放置訊息。

如需新增至啟用訊息串流之本端及模型佇列的其他屬性相關資訊，請參閱 [如何配置串流佇列](#)。

串流訊息

在大部分情況下，遞送至第二個佇列的訊息副本與原始訊息重複。這包括所有訊息描述子欄位，包括訊息 ID 和相關性 ID。串流訊息預期是非常接近原始訊息的副本，因此更容易找到它們，並在必要時將它們重播回另一個 IBM MQ 系統。

有部分訊息描述子欄位未保留在串流訊息上。在將串流訊息放置在第二個佇列之前，會對串流訊息進行下列變更：

- 不論原始訊息的期限為何，串流訊息的期限都會設為 **MQEI_UNLIMITED**。如果已在次要佇列上設定 **CAEXPRY**，則會使用其值來設定串流訊息的到期時間。
- 如果在原始訊息上設定下列任何報告選項，則不會在串流訊息上啟用這些選項。這是為了確保不會將非預期的報告訊息遞送至未設計用來接收它們的應用程式：
 - 活動報告
 - 到期報告
 - 異常狀況報告
 - 到達時確認 (COA)
 - 遞送時確認 (COD)

由於串流訊息的本質幾乎相同，因此次要佇列的大部分屬性不會影響串流訊息的訊息描述子欄位。例如，次要佇列的 **DEFPSIST** 及 **DEFPRTY** 屬性不會影響串流訊息。

下列異常狀況適用於串流訊息：

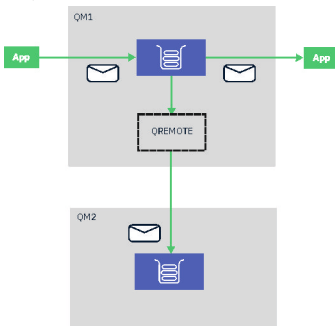
- **CAEXPRY** 屬性 (attribute)
如果已使用 **CAEXPRY** 屬性配置次要佇列，則會將此期限限制套用至串流訊息的期限。
- 叢集佇列的 **DEFBIND**
如果次要佇列是叢集佇列，則會使用次要佇列的 **DEFBIND** 屬性中設定的連結選項來放置串流訊息。

V 9.3.0 串流至遠端及別名佇列

可以將訊息串流至遠端佇列及別名佇列。例如，Q1 可能配置了 **STREAMQ (MY.REMOTE.Q)** 其中 **MY.REMOTE.Q** 是遠端佇列定義。

串流至遠端佇列

透過將訊息從本端佇列串流至遠端佇列，可以將重複訊息傳送至 IBM MQ 網路中另一個佇列管理程式上的佇列，如下圖所示：



串流至別名佇列

透過將訊息串流至別名佇列，可以將重複訊息傳送至別名佇列的目標。由於別名佇列的目標也可以是主題，因此可以將重複訊息傳送至發佈/訂閱主題。別名主題的任何訂閱者都會收到重複訊息的副本。透過此方式，您可以建立原始訊息的多個副本。不過，發佈/訂閱訊息的現有規則會套用至複製的訊息。這表示傳送至訂閱者的訊息將與原始訊息不同，包括：

- 具有新的訊息 ID。

- 具有產生的相關性 ID，視訂閱的配置而定。
- 將 `UserIdentifier` 欄位設為執行佇列管理程式的使用者，而不是放置訊息的使用者。
- `PutAppl` 名稱顯示佇列管理程式的名稱，而不是放置應用程式的名稱。

附註:

1. 無法在遠端佇列或別名佇列本身上配置 **STREAMQ** 屬性。您只能將訊息串流至它們，無法從它們進行串流。
2. 如果正在將訊息串流至佇列別名，則佇列別名的目標無法設定其 **STREAMQ** 屬性。

V 9.3.0 串流佇列限制

在 IBM MQ 中使用串流佇列時，不支援某些配置，這些配置在這裡記錄。

下列清單指定不受支援的配置：

- 定義彼此串流的佇列鏈，例如 Q1->Q2、Q2->Q3、Q3->Q4
- 定義串流佇列的迴圈，例如 Q1->Q2、Q2->Q1
- 使用提供的目的地來定義訂閱，其中該目的地已定義 **STREAMQ**
- 在使用 **USAGE (XMITQ)** 配置的佇列上定義 **STREAMQ**

註: **STREAMQ** 可以是遠端佇列，但您無法在遠端佇列定義上配置 **STREAMQ** 屬性。

- 修改動態佇列的 **STREAMQ** 屬性
- 將 **STREAMQ** 設為任何以 `SYSTEM.*` 開頭的值除了 `SYSTEM.DEFAULT.LOCAL.QUEUE`
- 在名為 `SYSTEM.*` 的任何佇列上定義 **STREAMQ**，但下列例外：

- `SYSTEM.DEFAULT.LOCAL.QUEUE`
- `SYSTEM.ADMIN.ACCOUNTING.QUEUE`
- `SYSTEM.ADMIN.ACTIVITY.QUEUE`
- `SYSTEM.ADMIN.CHANNEL.EVENT`
- `SYSTEM.ADMIN.COMMAND.EVENT`
- `SYSTEM.ADMIN.CONFIG.EVENT`
- `SYSTEM.ADMIN.LOGGER.EVENT`
- `SYSTEM.ADMIN.PERFM.EVENT`
- `SYSTEM.ADMIN.PUBSUB.EVENT`
- `SYSTEM.ADMIN.QMGR.EVENT`
- `SYSTEM.ADMIN.STATISTICS.QUEUE`
- `SYSTEM.DEFAULT.MODEL.QUEUE`
- `SYSTEM.JMS.TEMPQ.MODEL`

- 將 **STREAMQ** 設為模型佇列的名稱

- **V 9.3.1 z/OS** 未套用 APAR PH49686 時，在 IBM MQ 9.3.0 的共用佇列上定義 **STREAMQ**。當套用 APAR PH49686 或從 IBM MQ 9.3.1 時，已在 IBM MQ 9.3.0 解除此限制。

- **V 9.3.1 z/OS** 當未套用 APAR PH49686 時，將 **STREAMQ** 設為位於 IBM MQ 9.3.0 的共用佇列名稱。當套用 APAR PH49686 或從 IBM MQ 9.3.1 時，已在 IBM MQ 9.3.0 解除此限制。

V 9.3.0 串流佇列和交易

串流佇列特性容許將訊息放入一個佇列，以複製到第二個佇列。在大部分情況下，這兩個訊息會放置在工作單元下其各自的佇列中。

如果使用 `MQPMO_SYNCPOINT` 放置原始訊息，則會將重複訊息放置到串流佇列中針對原始放置啟動的相同工作單元下。

如果使用 MQPMO_NO_SYNCPOINT 放置原始工作單元，則即使原始工作單元未要求，也會啟動工作單元。這樣做有兩個原因：

1. 如果無法遞送原始訊息，它可確保不會遞送重複的訊息。如果原始訊息也已遞送，則串流佇列特性只會將訊息遞送至串流佇列。
2. 在工作單元內同時執行兩個工作單元，可改善效能

只有在原始 MQPUT 使用 MQPMO_NO_SYNCPOINT 非持續性且佇列的 STRMQOS 屬性設為 BESTEF (最大努力) 時，才不會在工作單元內遞送訊息。

附註：

1. 額外放入串流佇列的數量不會計入 MAXUMSGS 限制。
2. 如果佇列已配置 STRMQOS (BESTEF)，則無法遞送重複訊息不會導致回復工作單元。

V 9.3.0 與叢集佇列之間的串流

可以將訊息從本端佇列串流至叢集佇列，並將訊息從叢集佇列實例串流至本端佇列。

串流至叢集佇列

如果您有本端佇列來遞送原始訊息，且想要將每一則訊息的副本串流至叢集佇列的一或多個實例，則這會很有用。這可能是為了平衡重複訊息的處理工作量，或只是將重複訊息串流至叢集中其他位置的另一個佇列。

將訊息串流至叢集佇列時，會使用叢集工作量平衡演算法來配送訊息。根據叢集佇列的 DEFBIND 屬性來選擇叢集佇列實例。

例如，如果叢集佇列已配置 DEFBIND (OPEN)，則在開啟原始佇列時，會選擇叢集佇列的實例。所有重複訊息都會移至相同的叢集佇列實例，直到應用程式重新開啟原始佇列為止。

如果叢集佇列已配置 DEFBIND (NOTfixed)，則會針對每個 MQPUT 作業選擇叢集佇列的實例。

註：您應該使用相同的 DEFBIND 屬性值來配置所有叢集佇列實例。

從叢集佇列串流

如果您已傳送訊息至叢集佇列的數個實例，且想要將每一則訊息的副本遞送至串流佇列 (位於與叢集佇列實例相同的佇列管理程式上)，則這會很有用。

當原始訊息遞送至其中一個叢集佇列實例時，叢集接收端通道會將重複訊息遞送至串流佇列。

V 9.3.1 Multi 使用串流佇列來儲存訊息歷程

您可以使用串流佇列來保留訊息歷程一段有限的時間，並透過在訊息的串流目的地佇列上配置 CAPEXPY 屬性來達成此目的。

簡介

當訊息從一個佇列串流至另一個佇列時，對於重複副本，訊息上設定的任何期限值都會重設為 MQEI_UNQUIREN 值。依預設，如果沒有應用程式耗用訊息，則這會在您要串流至的佇列上穩定建置訊息。

在此實務範例中，您想要保留訊息的副本一段有限的時間，以便您能夠存取它們。例如，如果消費端應用程式意外移除原始訊息。

無法無限期保留每一則訊息的副本，且為了防止您要串流至的佇列已滿，有兩個選項：

- 執行應用程式以經常移除訊息
- 配置到期的訊息，使 IBM MQ 在一段時間之後將它們移除。

第二個選項可能更方便，因為它不需要您執行及維護應用程式，而只是為了防止佇列填滿。

配置 CAEXPRY

強制縮短有效期限 主題說明如何在佇列上配置 CAEXPRY。在此實務範例中，您必須在您要串流訊息的佇列上設定 CAEXPRY 屬性。

註：您不需要變更從中串流訊息之佇列上 CAEXPRY 屬性的值。

請考量下列考量，為重複訊息選擇適當的到期時間：

1. 您可能需要存取訊息的時間長度
2. 根據將訊息放入原始佇列的速率，佇列的 MAXDEPTH 屬性需要是什麼
3. 儲存重複訊息所需的儲存空間量。

這可能需要考量佇列管理程式檔案系統大小，以及佇列的 MAXFSIZE 屬性。

存取重複的訊息 (如果需要的話)

如果發生需要您存取及回復部分或所有重複訊息的問題，請使用 **dmpmqmsg** 指令來協助存取那些訊息。

dmpmqmsg 具有下列選項：

- 從佇列讀取訊息並將副本寫入檔案以稍後存取
- 從檔案讀取訊息並將它們寫回佇列以供應用程式使用

一旦 **dmpmqmsg** 將訊息標頭寫入檔案之後，就可以編輯訊息標頭。例如，在 **dmqmqmsg** 將訊息放回佇列進行處理之前，您可以將檔案中每一則訊息的 EXP 值變更為 -1，將訊息的期限重設回 MQEI_UNLIMITED。

效能考量

將重複訊息串流至另一個佇列，並在不再需要它們時使它們到期，成本較低。不過，成本遠小於手動將副本放入第二個佇列，並讓應用程式在一段時間之後破壞性地移除它們。[串流佇列效能報告](#) 提供此實務範例的效能相關資訊。

相關概念

[串流佇列安全](#)

[串流佇列及 AMS](#)

發佈/訂閱實務

示範使用發佈/訂閱叢集及發佈/訂閱階層的兩組實務範例。

下列子主題說明可用的發佈/訂閱實務範例：

實務範例: 建立發佈/訂閱叢集

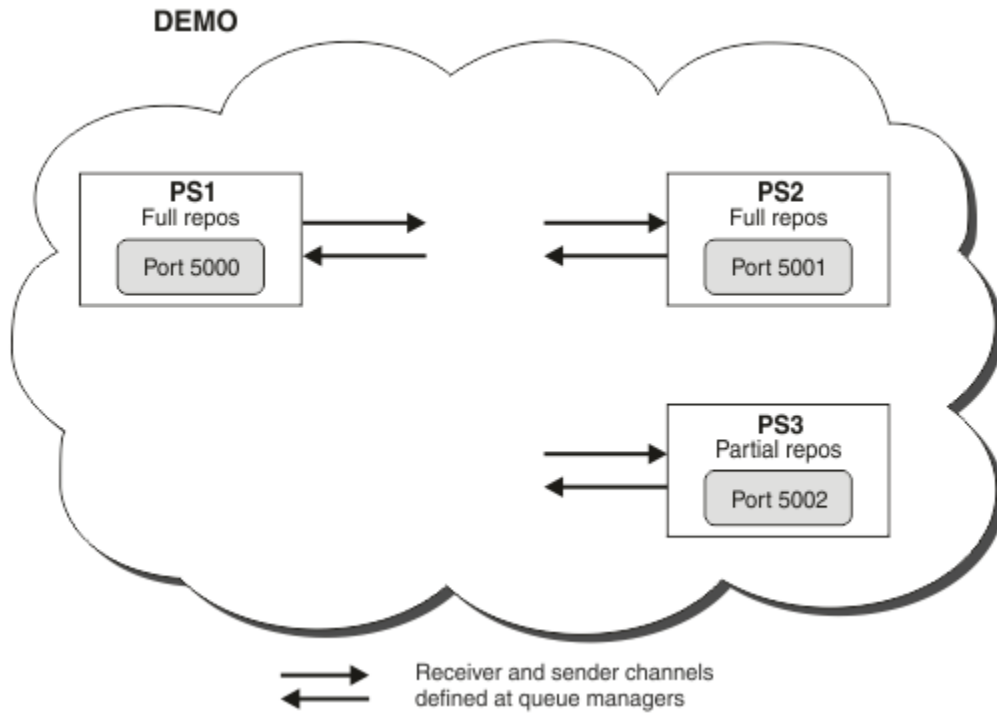
在此實務範例中，您建立一個簡單的三個佇列管理程式叢集，並將它配置成容許在一個佇列管理程式上建立的訂閱接收連接至另一個佇列管理程式的應用程式所發佈的訊息。

開始之前

此實務範例的起點是現有的 IBM MQ 安裝。如需安裝 IBM MQ 的指示，請遵循 [在 Windows 上安裝 IBM MQ 伺服器](#) 中的步驟。

關於這項作業

透過完成此實務範例中的步驟，您首先建立下列叢集：



此叢集由三個佇列管理程式組成，其中兩個定義為完整儲存庫佇列管理程式。

然後，在佇列管理程式 PS3 上定義叢集主題。透過建立叢集主題，您已使叢集成為發佈/訂閱叢集。若要測試發佈/訂閱叢集，您可以訂閱任何佇列管理程式上的主題，然後從另一個佇列管理程式將訊息發佈至主題，並檢查您的訂閱是否收到訊息。

相關工作

[設計發佈/訂閱叢集](#)

[配置佇列管理程式叢集](#)

建立及啟動佇列管理程式

建立並啟動三個佇列管理程式，稱為 PS1、PS2 及 PS3。

程序

1. 建立並啟動佇列管理程式 PS1。

a) 建立佇列管理程式。

在指令行中，輸入下列指令：

```
crtmqm PS1
```

b) 啟動佇列管理程式。

在指令行中，輸入下列指令：

```
strmqm PS1
```

2. 重複步驟 1，以建立並啟動佇列管理程式 PS2。

3. 重複步驟 1，以建立並啟動佇列管理程式 PS3。

下一步

現在您已準備好 [配置第一個佇列管理程式](#)。

配置第一個佇列管理程式

使用 MQSC 介面來定義 PS1 的接聽器及接收端通道，將佇列管理程式設為叢集的完整儲存庫，以及定義從 PS1 至 PS2 的傳送端通道，以便兩個完整儲存庫可以交換資訊。

開始之前

這項作業假設您已完成 [第 33 頁的『建立及啟動佇列管理程式』](#) 中的步驟。

程序

1. 定義並啟動 PS1 的接聽器。

- a) 啟動 MQSC 介面。

在指令行中，輸入下列指令：

```
runmqsc PS1
```

- b) 定義接聽器。

輸入下列 MQSC 指令：

```
DEFINE LISTENER(PS1_LS) TRPTYPE(TCP) CONTROL(QMGR) PORT(5000)
```

- c) 啟動接聽器。

輸入下列 MQSC 指令：

```
START LISTENER(PS1_LS)
```

2. 將佇列管理程式設為叢集的完整儲存庫。

輸入下列 MQSC 指令：

```
ALTER QMGR REPOS(DEMO)
```

3. 定義 PS1 的接收端通道，以容許叢集中的其他佇列管理程式與其通訊。

輸入下列 MQSC 指令：

```
DEFINE CHANNEL(DEMO.PS1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5000)')  
CLUSTER(DEMO)  
DESCR('TCP Cluster-receiver channel for queue manager PS1')
```

4. 定義從 PS1 到 PS2 的傳送端通道，以容許兩個完整儲存庫交換資訊。

輸入下列 MQSC 指令：

```
DEFINE CHANNEL(DEMO.PS2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5001)')  
CLUSTER(DEMO)  
DESCR('TCP Cluster-sender channel from PS1 to queue manager PS2')
```

下一步

現在您已準備好 [配置第二個佇列管理程式](#)。

配置第二個佇列管理程式

使用 MQSC 介面來定義 PS2 的接聽器及接收端通道，將佇列管理程式設為叢集的完整儲存庫，以及定義從 PS2 至 PS1 的傳送端通道，以便兩個完整儲存庫可以交換資訊。

開始之前

這項作業假設您已完成 [第 34 頁的『配置第一個佇列管理程式』](#) 中的步驟。

程序

1. 定義並啟動 PS2 的接聽器。

- a) 啟動 MQSC 介面。

在指令行中，輸入下列指令：

```
runmqsc PS2
```

- b) 定義接聽器。

輸入下列 MQSC 指令：

```
DEFINE LISTENER(PS2_LS) TRPTYPE(TCP) CONTROL(QMGR) PORT(5001)
```

- c) 啟動接聽器。

輸入下列 MQSC 指令：

```
START LISTENER(PS2_LS)
```

2. 將佇列管理程式設為叢集的完整儲存庫。

輸入下列 MQSC 指令：

```
ALTER QMGR REPOS(DEMO)
```

3. 定義 PS2 的接收端通道，以容許叢集中的其他佇列管理程式與其通訊。

輸入下列 MQSC 指令：

```
DEFINE CHANNEL(DEMO.PS2) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5001)')  
CLUSTER(DEMO)  
DESCR('TCP Cluster-receiver channel for queue manager PS2')
```

4. 定義從 PS2 到 PS1 的傳送端通道，以容許兩個完整儲存庫交換資訊。

輸入下列 MQSC 指令：

```
DEFINE CHANNEL(DEMO.PS1) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5000)')  
CLUSTER(DEMO)  
DESCR('TCP Cluster-sender channel from PS2 to PS1')
```

下一步

現在您已準備好 [配置第三個佇列管理程式](#)。

配置第三個佇列管理程式

使用 MQSC 介面來定義 PS3 的接聽器及接收端通道。透過定義從 PS3 到其中一個完整儲存庫佇列管理程式的傳送端通道，將 PS3 加入叢集。

開始之前

這項作業假設您已完成 [第 35 頁的『配置第二個佇列管理程式』](#) 中的步驟。

程序

1. 定義並啟動 PS3 的接聽器。

- a) 啟動 MQSC 介面。

在指令行中，輸入下列指令：

```
runmqsc PS3
```

- b) 定義接聽器。

輸入下列 MQSC 指令：

```
DEFINE LISTENER(PS3_LS) TRPTYPE(TCP) CONTROL(QMGR) PORT(5002)
```

- c) 啟動接聽器。

輸入下列 MQSC 指令：

```
START LISTENER(PS3_LS)
```

2. 定義 PS3 的接收端通道，以容許叢集中的其他佇列管理程式與其通訊。

輸入下列 MQSC 指令：

```
DEFINE CHANNEL(DEMO.PS3) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5002)')  
CLUSTER(DEMO)  
DESCR('TCP Cluster-receiver channel for queue manager PS3')
```

3. 定義從 PS3 到其中一個完整儲存庫佇列管理程式 (例如 PS1) 的傳送端通道。這會將 PS3 結合至叢集。

輸入下列 MQSC 指令：

```
DEFINE CHANNEL(DEMO.PS1) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5000)')  
CLUSTER(DEMO)  
DESCR('TCP Cluster-sender channel from PS3 to PS1')
```

4. 驗證 PS3 已順利加入叢集。

輸入下列 MQSC 指令：

```
DISPLAY CLUSQMGR(*) QMTYPE
```

此指令會傳回三個項目，分別適用於 QM1、QM2 及 QM3。QM1 及 QM2 應該具有 **QMTYPE REPOS**，而 QM3 應該具有 **QMTYPE NORMAL**。

下一步

現在，您已準備好 [定義叢集主題](#)。

定義叢集主題

發佈及訂閱應用程式可以發佈至任何主題字串，而不需要定義受管理主題物件。不過，如果發佈應用程式所連接的叢集佇列管理程式與建立訂閱的佇列管理程式不同，則必須定義受管理主題物件並新增至叢集。若要使主題成為叢集主題，您可以在其定義中指定叢集的名稱。

開始之前

這項作業假設您已完成 [第 35 頁的『配置第三個佇列管理程式』](#) 中的步驟。

關於這項作業

受管理主題物件會識別主題樹狀結構中透過其主題字串叢集化的點。發佈及訂閱應用程式可以使用該點或以下的任何主題字串，且其訊息會在佇列管理程式之間自動傳輸。

當您定義叢集主題時，也可以選擇其遞送模型。如需叢集中發佈遞送的相關資訊，請參閱 [設計發佈/訂閱叢集](#)。

在此情況下，我們使用預設遞送 *DIRECT*。這表示訊息會直接從發佈佇列管理程式傳送至訂閱佇列管理程式。

程序

1. 在 PS3 上定義叢集主題 SCORES。

若要使主題成為叢集主題，請指定叢集的名稱，並設定叢集遞送 (**CLROUTE**) 您要用於本主題的發佈和訂閱。

- a) 啟動 MQSC 介面。

在指令行中，輸入下列指令：

```
runmqsc PS3
```

- b) 定義叢集主題 SCORES。

輸入下列 MQSC 指令：

```
DEFINE TOPIC(SCORES) TOPICSTR('/Sport/Scores') CLUSTER(DEMO) CLROUTE(DIRECT)
```

- c) 輸入 end 以結束 PS3 的 MQSC 介面。

2. 驗證 PS1 上的主題定義。

- a) 啟動 PS1 的 MQSC 介面。

在指令行中，輸入下列指令：

```
runmqsc PS1
```

- b) 顯示叢集主題 SCORES 的叢集狀態。

輸入下列 MQSC 指令：

```
DISPLAY TCLUSTER(SCORES) CLSTATE
```

叢集主題 SCORES 的 **CLSTATE** 顯示為 ACTIVE。

下一步

如需這項作業的更詳細探索，請參閱 [配置發佈/訂閱叢集](#)。

您現在已準備好驗證解決方案。請參閱 [第 37 頁的『測試發佈/訂閱叢集』](#)。

測試發佈/訂閱叢集

從叢集中的不同佇列管理程式發佈及訂閱主題字串，以測試發佈/訂閱叢集。

開始之前

這項作業假設您已完成 [第 36 頁的『定義叢集主題』](#) 中的步驟。

關於這項作業

使用指令行，以及 IBM MQ 隨附的 `amqspub` 和 `amqssub` 範例應用程式，您可以從一個佇列管理程式發佈主題，並使用其他佇列管理程式訂閱該主題。當訊息發佈至主題時，訂閱端佇列管理程式會收到該訊息。

程序

1. 在指令行中，輸入下列指令：

```
amqspub /Sport/Scores/Football PS1
```

2. 同時，在個別指令行中輸入下列指令：

```
amqssub /Sport/Scores/Football PS2
```

```
amqssub /Sport/Scores/Football PS3
```

3. 在第一個指令行中，輸入訊息。

訊息會顯示在兩個訂閱指令行中。

註：如果 10 秒未收到發佈，則 `amqssub` 應用程式將逾時。

結果

發佈/訂閱叢集設定已完成。

下一步

請嘗試為主題樹狀結構的不同分支定義不同的主題物件，並使用不同的遞送模型。

發佈/訂閱階層實務

示範使用發佈/訂閱階層的三個實務範例。三個實務範例中的每一個都會設定相同的簡式發佈/訂閱拓撲。在每一個實務範例中，佇列管理程式依賴不同的方法來連接至階層中的鄰接佇列管理程式。

下列子主題說明可用的發佈/訂閱階層實務：

相關概念

[發佈/訂閱階層](#)

發佈/訂閱階層實務範例 1: 使用具有佇列管理程式別名的點對點通道

這是三個實務範例集的第一個實務範例，這些實務範例以不同方式設定發佈/訂閱階層，以建立佇列管理程式之間的連線。此實務範例設定發佈/訂閱階層，其使用具有佇列管理程式名稱別名的點對點通道。

關於這項作業

這組實務都使用一個稱為 QM1 的母項佇列管理程式，以及兩個稱為 QM2 和 QM3 的子項佇列管理程式。

實務範例 1 會分割成較小的區段，讓處理程序更容易遵循。

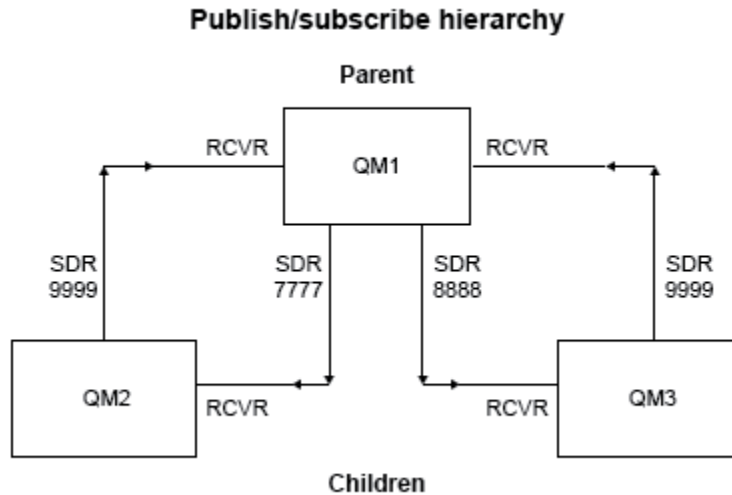


圖 3: 拓撲圖，顯示一般發佈者/訂閱階層中佇列管理程式之間的關係。

程序

1. 建立佇列管理程式。

- a) 使用下列指令，建立並啟動稱為 QM1、QM2 及 QM3 的三個佇列管理程式：

```

crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM1
strmqm QM1

crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM2
strmqm QM2

crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM3
strmqm QM3
  
```

- b) 在所有三個佇列管理程式上使用下列指令，啟用佇列管理程式發佈/訂閱模式：

```
ALTER QMGR PSMODE(ENABLED)
```

2. 使用與母項佇列管理程式同名的佇列管理程式別名，在佇列管理程式之間建立點對點通道連線。

- a) 將 QM2 上的傳輸佇列及佇列管理程式別名定義為 QM1。定義 QM1 的傳送端通道，以及在 QM1 for QM2 上建立之傳送端通道的接收端通道：

```

DEFINE QLOCAL(QM1.XMITQ) USAGE(XMITQ)

DEFINE QREMOTE (QM1) RNAME('') RQMNAME(QM1) XMITQ(QM1.XMITQ)

DEFINE CHANNEL('QM2.TO.QM1') CHLTYPE(SDR) CONNAME('localhost(9999)') XMITQ(QM1.XMITQ)
TRPTYPE(TCP)

DEFINE CHANNEL('QM1.TO.QM2') CHLTYPE(RCVR) TRPTYPE(TCP)
  
```

- b) 將 QM3 上的傳輸佇列及佇列管理程式別名定義為 QM1。定義 QM1 的傳送端通道，以及在 QM1 for QM3 上建立之傳送端通道的接收端通道：

```

DEFINE QLOCAL(QM1.XMITQ) USAGE(XMITQ)

DEFINE QREMOTE (QM1) RNAME('') RQMNAME(QM1) XMITQ(QM1.XMITQ)

DEFINE CHANNEL('QM3.TO.QM1') CHLTYPE(SDR) CONNAME('localhost(9999)') XMITQ(QM1.XMITQ)
TRPTYPE(TCP)
  
```

```
DEFINE CHANNEL('QM1.TO.QM3') CHLTYPE(RCVR) TRPTYPE(TCP)
```

- c) 將 QM1 上的傳輸佇列及佇列管理程式別名定義為 QM2 及 QM3。定義 QM2 和 QM3 的傳送端通道，以及在 QM2 和 QM3 for QM1 上建立之傳送端通道的接收端通道：

```
DEFINE QLOCAL(QM2.XMITQ) USAGE(XMITQ)
DEFINE QREMOTE (QM2) RNAME('') RQMNAME(QM2) XMITQ(QM2.XMITQ)
DEFINE CHANNEL('QM1.TO.QM2') CHLTYPE(SDR) CONNAME('localhost(7777)') XMITQ(QM2.XMITQ)
TRPTYPE(TCP)
DEFINE CHANNEL('QM2.TO.QM1') CHLTYPE(RCVR) TRPTYPE(TCP)
DEFINE QLOCAL(QM3.XMITQ) USAGE(XMITQ)
DEFINE QREMOTE (QM3) RNAME('') RQMNAME(QM3) XMITQ(QM3.XMITQ)
DEFINE CHANNEL('QM1.TO.QM3') CHLTYPE(SDR) CONNAME('localhost(8888)') XMITQ(QM3.XMITQ)
TRPTYPE(TCP)
DEFINE CHANNEL('QM3.TO.QM1') CHLTYPE(RCVR) TRPTYPE(TCP)
```

- d) 在佇列管理程式上啟動適當的接聽器：

```
runmqclsr -m QM1 -t TCP -p 9999 &
runmqclsr -m QM2 -t TCP -p 7777 &
runmqclsr -m QM3 -t TCP -p 8888 &
```

- e) 啟動下列通道：

- i) 在 QM1 上：

```
START CHANNEL('QM1.TO.QM2')
START CHANNEL('QM1.TO.QM3')
```

- ii) 在 QM2 上：

```
START CHANNEL('QM2.TO.QM1')
```

- iii) 在 QM3 上：

```
START CHANNEL('QM3.TO.QM1')
```

- f) 檢查所有通道是否已啟動：

```
DISPLAY CHSTATUS('QM1.TO.QM2')
DISPLAY CHSTATUS('QM1.TO.QM3')
DISPLAY CHSTATUS('QM2.TO.QM1')
DISPLAY CHSTATUS('QM3.TO.QM1')
```

- g)

3. 連接佇列管理程式並定義主題。

將子項佇列管理程式 QM2 及 QM3 連接至母項佇列管理程式 QM1。

- a) 在 QM2 和 QM3 上，將母項佇列管理程式設為 QM1：

```
ALTER QMGR PARENT (QM1)
```

- b) 在所有佇列管理程式上執行下列指令，以檢查子佇列管理程式是否連接至母項佇列管理程式：


```
DISPLAY PUBSUB TYPE(ALL)
```

顯示指令輸出。例如，以下是 QM1 的輸出，並強調顯示索引鍵詳細資料：

```
DISPLAY PUBSUB ALL
1 : DISPLAY PUBSUB ALL
AMQ8723: Display pub/sub status details.
QMNAME(QM1)          TYPE(LOCAL)
STATUS(ACTIVE)       SUBCOUNT(6)
TPCOUNT(9)
AMQ8723: Display pub/sub status details.
QMNAME(QM2) TYPE(CHILD)
STATUS(ACTIVE) SUBCOUNT(NONE)
TPCOUNT(NONE)
AMQ8723: Display pub/sub status details.
QMNAME(QM3) TYPE(CHILD)
STATUS(ACTIVE) SUBCOUNT(NONE)
TPCOUNT(NONE)
```

4. 使用 `amqspub.exe` 及 `amqssub.exe` 應用程式來發佈及訂閱主題。

a) 在第一個指令視窗中執行此指令：

```
amqspub Sport/Soccer QM2
```

b) 在第二個指令視窗中執行此指令：

```
amqssub Sport/Soccer QM1
```

c) 在第三個指令視窗中執行此指令：

```
amqssub Sport/Soccer QM3
```

結果

第二個及第三個指令視窗中的 `amqssub.exe` 應用程式會在第一個指令視窗中接收發佈的訊息。

發佈/訂閱階層實務範例 2: 使用傳輸佇列及遠端佇列管理程式具有相同名稱的點對點通道

這是以不同方式設定發佈/訂閱階層，以在佇列管理程式之間建立連線的三個實務範例集的第二個實務範例。此實務範例設定發佈/訂閱階層，其使用傳輸佇列名稱與遠端佇列管理程式相同的點對點通道。

關於這項作業

這組實務都使用一個稱為 QM1 的母項佇列管理程式，以及兩個稱為 QM2 和 QM3 的子項佇列管理程式。

此實務範例重複使用 [第 38 頁的『發佈/訂閱階層實務範例 1: 使用具有佇列管理程式別名的點對點通道』](#) 中的步驟 1、3 和 4。

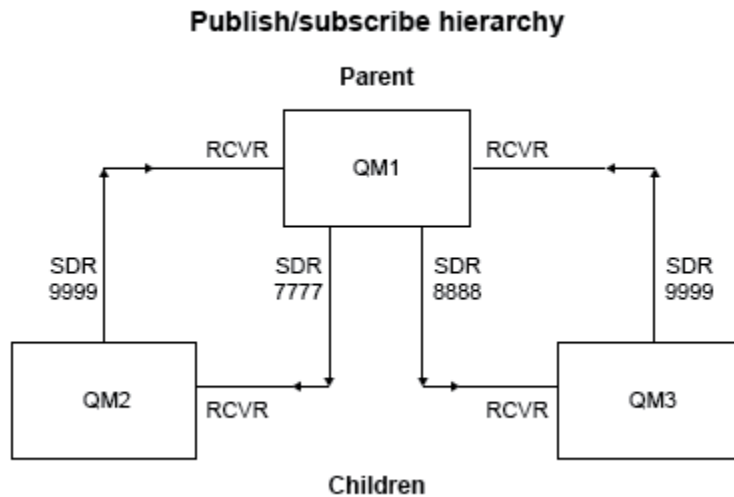


圖 4: 拓撲圖，顯示一般發佈者/訂閱階層中佇列管理程式之間的關係。

程序

1. 建立佇列管理程式。

- a) 使用下列指令，建立並啟動稱為 QM1、QM2 及 QM3 的三個佇列管理程式：

```

crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM1
strmqm QM1

crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM2
strmqm QM2

crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM3
strmqm QM3
  
```

- b) 在所有三個佇列管理程式上使用下列指令，啟用佇列管理程式發佈/訂閱模式：

```
ALTER QMGR PSMODE(ENABLED)
```

2. 使用與母項佇列管理程式同名的傳輸佇列，在佇列管理程式之間建立點對點通道連線。

- a) 將 QM2 上的傳輸佇列定義為 QM1。定義 QM1 的傳送端通道，以及 QM1 上所建立之 QM2 的傳送端通道的接收端通道：

```

DEFINE QLOCAL(QM1) USAGE(XMITQ)

DEFINE CHANNEL('QM2.TO.QM1') CHLTYPE(SDR) CONNAME('localhost(9999)') XMITQ(QM1)
TRPTYPE(TCP)

DEFINE CHANNEL('QM1.TO.QM2') CHLTYPE(RCVR) TRPTYPE(TCP)
  
```

- b) 將 QM3 上的傳輸佇列定義為 QM1。定義 QM1 的傳送端通道，以及在 QM1 for QM3 上建立之傳送端通道的接收端通道：

```

DEFINE QLOCAL(QM1) USAGE(XMITQ)

DEFINE CHANNEL('QM3.TO.QM1') CHLTYPE(SDR) CONNAME('localhost(9999)') XMITQ(QM1)
TRPTYPE(TCP)

DEFINE CHANNEL('QM1.TO.QM3') CHLTYPE(RCVR) TRPTYPE(TCP)
  
```

- c) 將 QM1 上的傳輸佇列定義為 QM2 和 QM3。定義 QM2 和 QM3 的傳送端通道，以及在 QM2 和 QM3 for QM1 上建立之傳送端通道的接收端通道:

```
DEFINE QLOCAL(QM2) USAGE(XMITQ)

DEFINE CHANNEL('QM1.TO.QM2') CHLTYPE(SDR) CONNAME('localhost(7777)') XMITQ(QM2)
TRPTYPE(TCP)

DEFINE CHANNEL('QM2.TO.QM1') CHLTYPE(RCVR) TRPTYPE(TCP)

DEFINE QLOCAL(QM3) USAGE(XMITQ)

DEFINE CHANNEL('QM1.TO.QM3') CHLTYPE(SDR) CONNAME('localhost(8888)') XMITQ(QM3)
TRPTYPE(TCP)

DEFINE CHANNEL('QM3.TO.QM1') CHLTYPE(RCVR) TRPTYPE(TCP)
```

- d) 在佇列管理程式上啟動適當的接聽器:

```
runmqclsr -m QM1 -t TCP -p 9999 &
runmqclsr -m QM2 -t TCP -p 7777 &
runmqclsr -m QM3 -t TCP -p 8888 &
```

- e) 啟動下列通道:

- i) 在 QM1 上:

```
START CHANNEL('QM1.TO.QM2')
START CHANNEL('QM1.TO.QM3')
```

- ii) 在 QM2 上:

```
START CHANNEL('QM2.TO.QM1')
```

- iii) 在 QM3 上:

```
START CHANNEL('QM3.TO.QM1')
```

- f) 檢查所有通道是否已啟動:

```
DISPLAY CHSTATUS('QM1.TO.QM2')
DISPLAY CHSTATUS('QM1.TO.QM3')
DISPLAY CHSTATUS('QM2.TO.QM1')
DISPLAY CHSTATUS('QM3.TO.QM1')
```

3. 連接佇列管理程式並定義主題。

將子項佇列管理程式 QM2 及 QM3 連接至母項佇列管理程式 QM1。

- a) 在 QM2 和 QM3 上，將母項佇列管理程式設為 QM1:

```
ALTER QMGR PARENT (QM1)
```

- b) 在所有佇列管理程式上執行下列指令，以檢查子佇列管理程式是否連接至母項佇列管理程式:

```
DISPLAY PUBSUB TYPE(ALL)
```

顯示指令輸出。例如，以下是 QM1 的輸出，並強調顯示索引鍵詳細資料:

```
DISPLAY PUBSUB ALL
1 : DISPLAY PUBSUB ALL
```

```
AMQ8723: Display pub/sub status details.  
QMNAME(QM1) TYPE(LOCAL)  
STATUS(ACTIVE) SUBCOUNT(6)  
TPCOUNT(9)  
AMQ8723: Display pub/sub status details.  
QMNAME(QM2) TYPE(CHILD)  
STATUS(ACTIVE) SUBCOUNT(NONE)  
TPCOUNT(NONE)  
AMQ8723: Display pub/sub status details.  
QMNAME(QM3) TYPE(CHILD)  
STATUS(ACTIVE) SUBCOUNT(NONE)  
TPCOUNT(NONE)
```

4. 使用 amqspub.exe 及 amqssub.exe 應用程式來發佈及訂閱主題。

- a) 在第一個指令視窗中執行此指令：

```
amqspub Sport/Soccer QM2
```

- b) 在第二個指令視窗中執行此指令：

```
amqssub Sport/Soccer QM1
```

- c) 在第三個指令視窗中執行此指令：

```
amqssub Sport/Soccer QM3
```

結果

第二個及第三個指令視窗中的 amqssub.exe 應用程式會在第一個指令視窗中接收發佈的訊息。

相關工作

第 38 頁的『[發佈/訂閱階層實務範例 1: 使用具有佇列管理程式別名的點對點通道](#)』

這是三個實務範例集的第一個實務範例，這些實務範例以不同方式設定發佈/訂閱階層，以建立佇列管理程式之間的連線。此實務範例設定發佈/訂閱階層，其使用具有佇列管理程式名稱別名的點對點通道。

第 44 頁的『[發佈/訂閱階層實務範例 3: 使用叢集通道來新增佇列管理程式](#)』

這是在三個實務範例集中以不同方式設定發佈/訂閱階層的第三個實務範例，以建立佇列管理程式之間的連線。此實務範例使用叢集通道，將佇列管理程式新增至階層。

[將佇列管理程式連接至發佈/訂閱階層](#)

發佈/訂閱階層實務範例 3: 使用叢集通道來新增佇列管理程式

這是在三個實務範例集中以不同方式設定發佈/訂閱階層的第三個實務範例，以建立佇列管理程式之間的連線。此實務範例使用叢集通道，將佇列管理程式新增至階層。

關於這項作業

這組實務都使用一個稱為 QM1 的母項佇列管理程式，以及兩個稱為 QM2 和 QM3 的子項佇列管理程式。

註：此實務範例僅使用叢集配置將佇列管理程式連接在一起，而不是透過叢集作業主題傳播發佈/訂閱資料流量。當定義相同叢集中佇列管理程式之間的子項/母項階層關係時，會根據主題樹狀結構中主題的發佈和訂閱範圍設定，在佇列管理程式之間傳播發佈。請務必不要使用主題的叢集名稱設定，將主題新增至叢集。如果使用叢集名稱，則拓撲會變成發佈/訂閱叢集，且不需要定義子項/母項階層關係。請參閱第 32 頁的『[實務範例: 建立發佈/訂閱叢集](#)』及 [規劃分散式發佈/訂閱網路](#)。

此實務範例重複使用第 38 頁的『[發佈/訂閱階層實務範例 1: 使用具有佇列管理程式別名的點對點通道](#)』中的步驟 1、3 和 4。

此實務範例會建立稱為 DEMO 的叢集，其中 QM1 和 QM2 是完整儲存庫，而 QM3 是局部儲存庫。佇列管理程式 QM1 是佇列管理程式 QM2 及 QM3 的母項。

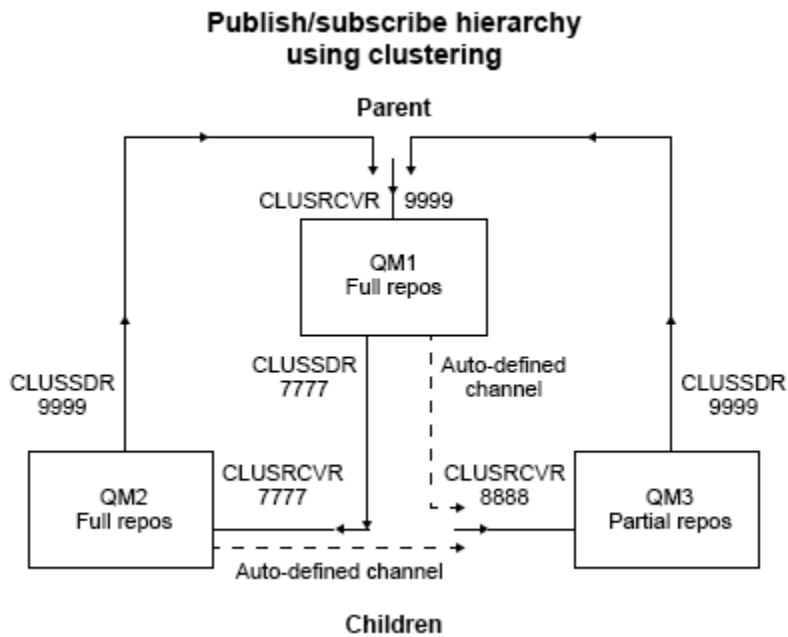


圖 5: 拓撲圖，顯示使用叢集通道的佇列管理程式之間的關係。

程序

1. 建立佇列管理程式。

- a) 使用下列指令，建立並啟動稱為 QM1、QM2 及 QM3 的三個佇列管理程式：

```

crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM1
strmqm QM1

crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM2
strmqm QM2

crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM3
strmqm QM3

```

- b) 在所有三個佇列管理程式上使用下列指令，啟用佇列管理程式發佈/訂閱模式：

```
ALTER QMGR PSMODE(ENABLED)
```

2. 在佇列管理程式與叢集之間建立點對點通道連線。

- a) 在 QM1 及 QM2 上，將 **REPOS** 參數設為叢集 DEMO 的名稱：

```
ALTER QMGR REPOS(DEMO)
```

- b) 在佇列管理程式上啟動適當的接聽器：

```

runmqclsr -m QM1 -t TCP -p 9999 &
runmqclsr -m QM2 -t TCP -p 7777 &
runmqclsr -m QM3 -t TCP -p 8888 &

```

- c) 在每一個佇列管理程式上定義叢集接收端通道：

- i) 在 QM1 上：

```
DEFINE CHANNEL(TO.QM1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('localhost(9999)')
CLUSTER(DEMO)
```

ii) 在 QM2 上:

```
DEFINE CHANNEL(TO.QM2) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('localhost(7777)')
CLUSTER(DEMO)
```

iii) 在 QM3 上:

```
DEFINE CHANNEL(TO.QM3) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('localhost(8888)')
CLUSTER(DEMO)
```

d) 定義叢集中每一個佇列管理程式上完整儲存庫的叢集傳送端通道:

i) 在 QM1 上:

```
DEFINE CHANNEL(TO.QM2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('localhost(7777)')
CLUSTER(DEMO)
```

ii) 在 QM2 上:

```
DEFINE CHANNEL(TO.QM1) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('localhost(9999)')
CLUSTER(DEMO)
```

iii) QM3 可以具有叢集傳送端通道，以連接 QM1 或 QM2 上的完整儲存庫。此範例定義 QM1 的通道:

```
DEFINE CHANNEL(TO.QM1) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('localhost(9999)')
CLUSTER(DEMO)
```

3. 連接佇列管理程式並定義主題。

將子項佇列管理程式 QM2 及 QM3 連接至母項佇列管理程式 QM1。

a) 在 QM2 和 QM3 上，將母項佇列管理程式設為 QM1:

```
ALTER QMGR PARENT (QM1)
```

b) 在所有佇列管理程式上執行下列指令，以檢查子佇列管理程式是否連接至母項佇列管理程式:

```
DISPLAY PUBSUB TYPE(ALL)
```

顯示指令輸出。例如，以下是 QM1 的輸出，並強調顯示索引鍵詳細資料:

```
DISPLAY PUBSUB ALL
1 : DISPLAY PUBSUB ALL
AMQ8723: Display pub/sub status details.
QMNAME(QM1)          TYPE(LOCAL)
STATUS(ACTIVE)      SUBCOUNT(6)
TPCOUNT(9)
AMQ8723: Display pub/sub status details.
QMNAME(QM2) TYPE(CHILD)
STATUS(ACTIVE) SUBCOUNT(NONE)
TPCOUNT(NONE)
AMQ8723: Display pub/sub status details.
QMNAME(QM3) TYPE(CHILD)
STATUS(ACTIVE) SUBCOUNT(NONE)
TPCOUNT(NONE)
```

4. 使用 amqspub.exe 及 amqssub.exe 應用程式來發佈及訂閱主題。

a) 在第一個指令視窗中執行此指令:

```
amqspub Sport/Soccer QM2
```

b) 在第二個指令視窗中執行此指令:

```
amqssub Sport/Soccer QM1
```

c) 在第三個指令視窗中執行此指令:

```
amqssub Sport/Soccer QM3
```

結果

第二個及第三個指令視窗中的 `amqssub.exe` 應用程式會在第一個指令視窗中接收發佈的訊息。

相關工作

第 38 頁的『[發佈/訂閱階層實務範例 1: 使用具有佇列管理程式別名的點對點通道](#)』

這是三個實務範例集的第一個實務範例，這些實務範例以不同方式設定發佈/訂閱階層，以建立佇列管理程式之間的連線。此實務範例設定發佈/訂閱階層，其使用具有佇列管理程式名稱別名的點對點通道。

第 41 頁的『[發佈/訂閱階層實務範例 2: 使用傳輸佇列及遠端佇列管理程式具有相同名稱的點對點通道](#)』

這是以不同方式設定發佈/訂閱階層，以在佇列管理程式之間建立連線的三個實務範例集的第二個實務範例。此實務範例設定發佈/訂閱階層，其使用傳輸佇列名稱與遠端佇列管理程式相同的點對點通道。

[將佇列管理程式連接至發佈/訂閱階層](#)

交易式支援實務

使用交易式支援，您可以讓應用程式可靠地使用資料庫。

本節介紹交易式支援。讓應用程式能夠將 IBM MQ 與資料庫產品搭配使用所需的工作跨越應用程式設計和系統管理的區域。請搭配使用這裡的資訊與 [確定及取消工作單元](#)。

我們從引進構成交易的工作單元開始，然後說明您讓 IBM MQ 能夠與資料庫協調交易的方式。

相關概念

第 47 頁的『[介紹工作單元](#)』

本主題介紹並定義工作單元、確定、取消及同步點的一般概念。它還包含兩個實務範例，用於說明廣域工作單元。

介紹工作單元

本主題介紹並定義工作單元、確定、取消及同步點的一般概念。它還包含兩個實務範例，用於說明廣域工作單元。

當程式將訊息放入工作單元內的佇列時，只有在程式確定工作單元時，其他程式才會看到這些訊息。若要確定工作單元，必須順利完成所有更新，以保留資料完整性。

如果程式偵測到錯誤，並決定不將放置作業設為永久，它可以退出工作單元。當程式執行取消時，IBM MQ 會移除該工作單元放置在佇列上的訊息，以還原佇列。

同樣地，當程式從工作單元內的一個以上佇列取得訊息時，這些訊息會保留在佇列上，直到程式確定工作單元為止，但其他程式無法擷取這些訊息。當程式確定工作單元時，會從佇列永久刪除訊息。如果程式取消工作單元，IBM MQ 會讓訊息可供其他程式擷取，以還原佇列。

在最簡單的情況下，會在作業結束時做出確定或取消變更的決策。不過，對於應用程式來說，同步化作業內其他邏輯點上的資料變更會更有用。這些邏輯點稱為同步點(或同步點)，而在兩個同步點之間處理一組更新項目的期間稱為工作單元。數個 MQGET 呼叫和 MQPUT 呼叫可以是單一工作單元的一部分。

使用 IBM MQ，我們需要區分本端與廣域工作單元:

本端工作單元

是指在佇列管理程式內使用單階段確定處理程序，提供對 IBM MQ 佇列放置及從中取得的唯一動作，以及每一個工作單元的協調。

當唯一要更新的資源是由單一 IBM MQ 佇列管理程式所管理的佇列時，請使用本端工作單元。使用 MQCMIT 動詞來確定更新項目，或使用 MQBACK 來取消。

除了使用本端工作單元所涉及的日誌管理之外，沒有其他系統管理作業。在您的應用程式中，搭配使用 MQPUT 及 MQGET 呼叫與 MQCMIT 及 MQBACK，請嘗試使用 MQPMO_SYNCPOINT 及 MQGMO_SYNCPOINT 選項。(如需日誌管理的相關資訊，請參閱 [管理日誌檔](#)。)

廣域工作單元

是否也會更新其他資源 (例如關聯式資料庫中的表格)。當涉及多個資源管理程式時，需要使用兩階段確定程序來協調廣域工作單元的交易管理程式軟體。

當您還需要包括關聯式資料庫管理程式軟體 (例如 Db2、Oracle、Sybase 及 Informix) 的更新項目時，請使用廣域工作單元。

使用廣域工作單元有數個可能的實務範例。這裡記載兩個實務範例：

1. 在第一個中，佇列管理程式本身會作為交易管理程式。在此實務範例中，MQI 動詞控制廣域工作單元；它們在應用程式中使用 MQBEGIN 動詞啟動，然後使用 MQCMIT 確定或使用 MQBACK 取消。
2. 在第二個中，交易管理程式角色由其他軟體 (例如 TXSeries、Encina 或 Tuxedo) 執行。在此實務範例中，交易管理程式軟體提供的 API 用來控制工作單元 (例如，EXEC CICS SYNCPOINT for TXSeries)。

下列各節說明使用廣域工作單元所需的所有步驟，依兩個實務範例組織：

- [第 48 頁的『實務範例 1: 佇列管理程式執行協調』](#)
- [第 69 頁的『實務範例 2: 其他軟體提供協調』](#)

Muti 實務範例 1: 佇列管理程式執行協調

在實務範例 1 中，佇列管理程式充當交易管理程式。在此實務範例中，MQI 動詞控制廣域工作單元；它們在應用程式中使用 MQBEGIN 動詞啟動，然後使用 MQCMIT 確定或使用 MQBACK 取消。

Muti 隔離層次

在 IBM MQ 中，視資料庫內實作的交易隔離設計而定，在資料庫更新之前可能會看到佇列上的訊息。

當 IBM MQ 佇列管理程式作為 XA 交易管理程式運作時，為了協調 XA 資源管理程式的更新項目，會遵循下列確定通訊協定：

1. 準備所有 XA 資源管理程式。
2. 確定 IBM MQ 佇列管理程式資源管理程式。
3. 確定其他資源管理程式。

在步驟 2 與 3 之間，應用程式可能會看到已確定至佇列的訊息，但資料庫中的對應列不會反映此訊息。

如果資料庫已配置成應用程式的資料庫 API 呼叫會等待擱置更新完成，則這不是問題。

您可以透過以不同方式配置資料庫來解決此問題。所需的配置類型稱為「隔離層次」。如需隔離層次的相關資訊，請參閱資料庫文件。或者，您可以將佇列管理程式配置成以下列相反順序來確定資源管理程式：

1. 準備所有 XA 資源管理程式。
2. 確定其他資源管理程式。
3. 確定 IBM MQ 佇列管理程式資源管理程式。

當您變更通訊協定時，最後會確定 IBM MQ 佇列管理程式，因此只有在完成對應的資料庫更新之後，讀取佇列中訊息的應用程式才會看到訊息。

若要配置佇列管理程式使用此已變更的通訊協定，請設定 **AMQ_REVERSE_COMMIT_ORDER** 環境變數。

在執行 **strmqm** 的環境中設定此環境變數，以啟動佇列管理程式。例如，在啟動佇列管理程式之前，請在 Shell 中執行下列指令：

```
export AMQ_REVERSE_COMMIT_ORDER=1
```

註：設定此環境變數可能會導致每個交易額外的日誌項目，因此這將對每個交易的效能產生較小的影響。

Multi 資料庫協調

當佇列管理程式協調廣域工作單元本身時，就可以整合工作單元內的資料庫更新項目。也就是說，可以撰寫混合 MQI 和 SQL 應用程式，且 MQCMIT 和 MQBACK 動詞可以一起用來確定或回復佇列和資料庫的變更。

佇列管理程式會使用 *X/Open Distributed Transaction Processing: The XA Specification* 中說明的兩段式確定通訊協定來達成此目的。當工作單元要確定時，佇列管理程式會先詢問每一個參與資料庫管理程式是否準備好確定其更新項目。只有在所有參與者 (包括佇列管理程式本身) 都準備好要確定時，才會確定所有佇列及資料庫更新項目。如果任何參與者無法準備其更新項目，則會改為取消工作單元。

一般而言，廣域工作單元是透過下列方法在應用程式中實作 (在虛擬碼中)：

```
MQBEGIN
MQGET (在訊息選項中包括旗標 MQGMO_SYNCPOINT)
MQPUT (在訊息選項中包括旗標 MQPMO_SYNCPOINT)
SQL INSERT
MQCMIT
```

MQBEGIN 的目的是表示廣域工作單元的開始。MQCMIT 的目的是表示廣域工作單元的結束，並使用兩階段確定通訊協定，以所有參與的資源管理程式來完成它。

當工作單元 (也稱為 交易) 已使用 MQCMIT 順利完成，在該工作單元內採取的所有動作都會變成永久或無法回復。如果由於任何原因，工作單元失敗，則會改為取消所有動作。無法將工作單元中的一個動作變成永久，而將另一個動作變成取消。這是工作單元的原則：工作單元內的所有動作都是永久的，或都不是永久的。

註：

1. 應用程式設計師可以呼叫 MQBACK 來強制取消工作單元。如果在呼叫 MQCMIT 之前，應用程式或資料庫失敗，佇列管理程式也會取消工作單元。
2. 如果應用程式呼叫 MQDISC 而未呼叫 MQCMIT，則佇列管理程式的行為會如同已呼叫 MQCMIT 一樣，並確定工作單元。

在 MQBEGIN 與 MQCMIT 之間，佇列管理程式不會對資料庫發出任何呼叫來更新其資源。也就是說，資料庫表格變更的唯一方式是您的程式碼 (例如，虛擬程式碼中的 SQL INSERT)。

如果佇列管理程式在確定通訊協定期間失去與任何資料庫管理程式的聯絡，則會提供完整回復支援。如果資料庫管理程式在不確定時變成無法使用，亦即，它已順利準備好進行確定，但尚未收到確定或取消決策，則佇列管理程式會記住工作單元的結果，直到該結果順利遞送至資料庫為止。同樣地，如果佇列管理程式因未完成的確定作業而終止，則會在佇列管理程式重新啟動時記住這些作業。如果應用程式非預期地終止，則工作單元的完整性不會受損，但結果取決於應用程式在處理程序中終止的位置，如 [第 50 頁的表 2](#) 中所述。

下列表格彙總了資料庫或應用程式失敗時所發生的情況：

失敗發生次數	結果
在應用程式呼叫 MQCMIT 之前。	工作單元已取消。
在應用程式呼叫 MQCMIT 期間，之前所有資料庫都已指出它們已順利準備好。	工作單元已取消，原因碼為 MQRC_BACKED_OUT。
在應用程式呼叫 MQCMIT 期間，在所有資料庫都指出已順利準備好之後，但在所有資料庫都指出已順利確定之前。	工作單元由佇列管理程式以可回復狀態保留，原因碼為 MQRC_OUTCOME_PENDING。
在應用程式呼叫 MQCMIT 期間，在所有資料庫都指出已順利確定之後。	工作單元已確定，原因碼為 MQRC_NONE。
在應用程式呼叫 MQCMIT 之後。	工作單元已確定，原因碼為 MQRC_NONE。

失敗發生次數	結果
在應用程式呼叫 MQCMIT 之前。	工作單元已取消。
在應用程式呼叫 MQCMIT 期間，在佇列管理程式收到應用程式的 MQCMIT 要求之前。	工作單元已取消。
在應用程式呼叫 MQCMIT 期間，在佇列管理程式收到應用程式的 MQCMIT 要求之後。	佇列管理程式會嘗試使用兩段式確定來確定 (取決於資料庫產品順利執行並確定其工作單元部分的情況)。

如果從 MQCMIT 傳回的原因碼是 MQRC_OUTCOME_PENDING，則佇列管理程式會記住工作單元，直到它能夠重新建立與資料庫伺服器的聯絡，並告知它確定其部分工作單元。如需如何及何時完成回復的相關資訊，請參閱第 64 頁的『失去與 XA 資源管理程式的聯絡時的考量』。

佇列管理程式會依照 *X/Open Distributed Transaction Processing: The XA Specification* 中的說明，使用 XA 介面與資料庫管理程式進行通訊。這些函數呼叫的範例有 `xa_open`、`xa_start`、`xa_end`、`xa_prepare` 和 `xa_commit`。我們使用的術語 交易管理程式 和 資源管理程式 與 XA 規格中使用的術語相同。

Multi 限制

資料庫協調支援有一些限制。

下列限制適用：

- 在 MQI 用戶端應用程式中 不 支援在 IBM MQ 工作單元內協調資料庫更新項目的能力。在用戶端應用程式中使用 MQBEGIN 失敗。呼叫 MQBEGIN 的程式必須在與佇列管理程式相同的機器上作為 伺服器 應用程式執行。

註： 伺服器 應用程式是已與必要的 IBM MQ 伺服器程式庫鏈結的程式；用戶端 應用程式是已與必要的 IBM MQ 用戶端程式庫鏈結的程式。如需編譯及鏈結以程序化語言撰寫的程式的詳細資料，請參閱 [建置 IBM MQ MQI clients 的應用程式及建置程序化應用程式](#)。

- 資料庫伺服器可以位於與佇列管理程式伺服器不同的機器上，只要資料庫用戶端安裝在與佇列管理程式相同的機器上，且它支援此功能。請參閱資料庫產品的說明文件，以判斷其用戶端軟體是否可用於兩段式確定系統。
- 雖然佇列管理程式的行為與資源管理程式相同 (基於涉及「實務範例 2」廣域工作單元的目的)，但無法讓一個佇列管理程式在其「實務範例 1」廣域工作單元內協調另一個佇列管理程式。

Multi 切換載入檔案

交換器載入檔案是由 IBM MQ 應用程式及佇列管理程式中的程式碼所載入的共用程式庫 (Windows 系統上的 DLL)。其目的是簡化資料庫用戶端共用程式庫的載入，並傳回 XA 函數的指標。

在啟動佇列管理程式之前，必須先指定交換器載入檔案的詳細資料。詳細資料放置在 AIX, Linux, and Windows 系統上的 `qm.ini` 檔案中。

- 在 Windows 和 Linux (x86 和 x86-64 平台) 系統上，請使用 IBM MQ Explorer 來更新 `qm.ini` 檔。
- 在所有其他系統上，直接編輯 `qm.ini` 檔案。

如果 IBM MQ 安裝支援「實務範例 1」廣域工作單元，則會提供交換器載入檔的 C 來源。來源包含稱為 `MQStart` 的函數。當載入交換器載入檔案時，佇列管理程式會呼叫此函數，其會傳回稱為 XA 交換器之結構的位址。

XA 參數結構存在於資料庫用戶端共用程式庫中，且包含許多函數指標，如第 50 頁的表 3 中所述：

函數指標名稱	XA 函數	用途
<code>xa_open_entry</code>	<code>xa_open</code>	連接至資料庫
<code>xa_close_entry</code>	<code>xa_close</code>	中斷與資料庫的連線

表 3: XA 交換器函數指標 (繼續)

函數指標名稱	XA 函數	用途
xa_start_entry	xa_start	啟動廣域工作單元的分支
xa_end_entry	xa_end	暫停廣域工作單元的分支
xa_rollback_entry	xa_rollback	回復廣域工作單元的分支
xa_prepare_entry	xa_prepare	準備確定廣域工作單元的分支
xa_commit_entry	xa_commit	確定廣域工作單元的分支
xa_recover_entry	xa_recover	從資料庫探索它是否具有不確定的工作單元
xa_forget_entry	xa_forget	容許資料庫忘記廣域工作單元的分支
xa_complete_entry	xa_complete	完成廣域工作單元的分支

在應用程式中第一個 MQBEGIN 呼叫期間，作為 MQBEGIN 一部分執行的 IBM MQ 程式碼會載入交換器載入檔案，並呼叫資料庫共用程式庫中的 xa_open 函數。同樣地，在佇列管理程式啟動期間及其他後續情況中，部分佇列管理程式處理程序會載入切換載入檔案並呼叫 xa_open。

您可以使用動態登錄來減少 xa_* 呼叫數。如需此最佳化技術的完整說明，請參閱第 68 頁的『XA 動態登錄』。

Multi 配置系統以進行資料庫協調

您必須先執行數個作業，然後資料庫管理程式才能參與由佇列管理程式協調的廣域工作單元。這些說明如下：

- 第 51 頁的『安裝及配置資料庫產品』
- 第 52 頁的『建立交換器載入檔案』
- 第 52 頁的『將配置資訊新增至佇列管理程式』
- 第 54 頁的『撰寫及修改應用程式』
- 第 54 頁的『測試系統』

Multi 安裝及配置資料庫產品

若要安裝及配置資料庫產品，請參閱產品自己的說明文件。本節中的主題說明一般配置問題，以及它們如何與 IBM MQ 與資料庫之間的交互作業相關。

資料庫連線

建立佇列管理程式標準連線的應用程式會與個別本端佇列管理程式代理程式處理程序中的執行緒相關聯。(在此環境定義中，不是捷徑連線的連線是標準連線。請參閱 [使用 MQCONNX 呼叫連接至佇列管理程式](#)。)

當應用程式發出 MQBEGIN 時，它和代理程式處理程序都會呼叫資料庫用戶端程式庫中的 xa_open 函數。為了回應此情況，資料庫用戶端程式庫程式碼連接會寫入要在工作單元中涉及的資料庫從應用程式及佇列管理程式處理程序。只要應用程式保持連接至佇列管理程式，就會維護這些資料庫連線。

如果資料庫僅支援有限數目的使用者或連線，則這是重要的考量，因為正在建立與資料庫的兩個連線，以支援一個應用程式。

主從架構配置

載入 IBM MQ 佇列管理程式及應用程式程序中的資料庫用戶端程式庫 **必須** 能夠傳送至其伺服器，以及從其伺服器接收。請確保：

- 資料庫的主從式配置檔具有正確的詳細資料
- 在佇列管理程式及應用程式程序的環境中設定相關環境變數

Multi 建立交換器載入檔案

IBM MQ 隨附一個範例 make 檔，用來建置受支援資料庫管理程式的切換載入檔。

`MQ_INSTALLATION_PATH` 代表 IBM MQ 安裝所在的高階目錄。

範例 make 檔與建置交換器載入檔案所需的所有相關聯 C 原始檔一起安裝在下列目錄中：

- 對於 IBM MQ for Windows，在 `MQ_INSTALLATION_PATH\tools\c\samples\xatm\` 目錄中
- 若為 IBM MQ for UNIX 和 Linux 系統，在 `MQ_INSTALLATION_PATH/samp/xatm/` 目錄中

用來建置切換載入檔的範例目標如下：

- 若為 Db2: `db2swit`。
- 若為 Oracle，請使用 `orasit`
- 對於 Informix，`infswit`
- 若為 Sybase，`sybswit`

Windows 產生的切換檔案放置在 `C:\Program Files\IBM\MQ\exits` 中。

Linux UNIX 如果您有 32 位元佇列管理程式，則範例 make 檔 `xaswit.mak` 會在 `/var/mqm/exits` 中安裝 32 位元切換載入檔。

Linux UNIX 如果您有 64 位元佇列管理程式，則範例 make 檔 `xaswit.mak` 會在 `/var/mqm/exits` 中安裝 32 位元切換載入檔，並在 `/var/mqm/exits64` 中安裝 64 位元切換載入檔。

V9.3.0 Linux UNIX 如果您的系統不支援 32 位元編譯，請使用資料庫的 64 位元唯一目標：

- 若為 Db2: `db2swit64`
- 若為 Oracle: `oraswit64`
- 若為 Informix: `infswit64`
- 若為 Sybase: `sybswit64`

檔案安全

由於 IBM MQ 無法控制的原因，您的作業系統可能無法由 IBM MQ 載入交換器載入檔案。如果發生此情況，錯誤訊息會寫入 IBM MQ 錯誤日誌，且 `MQBEGIN` 呼叫可能會失敗。為了協助確保您的作業系統不會使交換器載入檔案載入失敗，您必須滿足下列需求：

1. 交換器載入檔案必須位於 `qm.ini` 檔案中提供的位置。
2. 交換器載入檔案必須可供所有需要載入它的處理程序存取，包括佇列管理程式處理程序及應用程式處理程序。
3. 切換載入檔案所相依的所有檔案庫 (包括資料庫產品所提供的檔案庫) 都必須存在且可存取。

Multi 將配置資訊新增至佇列管理程式

當您已建立資料庫管理程式的交換器載入檔案，並將它放置在安全位置時，您必須將該位置指定給佇列管理程式。

若要指定位置，請執行下列步驟：

- 在 Windows 和 Linux (x86 和 x86-64 平台) 系統上，請使用 IBM MQ Explorer。在 XA 資源管理程式下的佇列管理程式內容畫面中，指定交換器載入檔的詳細資料。
- 在所有其他系統上，請在佇列管理程式 `qm.ini` 檔的 `XAResourceManager` 段落中指定交換器負載檔案的詳細資料。

針對佇列管理程式將要協調的資料庫，新增 XAResourceManager 段落。最常見的情況是只有一個資料庫，因此只有一個 XAResourceManager 段落。如需涉及多個資料庫之更複雜的配置的詳細資料，請參閱第 63 頁的『多個資料庫配置』。XAResourceManager 段落的屬性如下：

名稱 = 名稱

使用者選擇用來識別資源管理程式的字串。實際上，它會將名稱提供給 XAResourceManager 段落。名稱是必要的，長度最多可以為 31 個字元。

您選擇的名稱必須是唯一的；在此 qm.ini 檔案中，只能有一個 XAResourceManager 段落具有此名稱。該名稱也應該有意義，因為在使用 dspmqtrn 指令時，佇列管理程式會使用它在佇列管理程式錯誤日誌訊息及輸出中參照此資源管理程式。（如需相關資訊，請參閱第 65 頁的『使用 dspmqtrn 指令顯示未完成的工作單元』。）

一旦您已選擇名稱，並已啟動佇列管理程式，請不要變更名稱屬性。如需變更配置資訊的詳細資料，請參閱第 67 頁的『變更配置資訊』。

SwitchFile= name

這是您先前建置的 XA 交換器載入檔名稱。這是必要屬性。佇列管理程式及 IBM MQ 應用程式程序中的程式碼會兩次嘗試載入交換器載入檔案：

1. 在佇列管理程式啟動時
2. 當您在 IBM MQ 應用程式程序中第一次呼叫 MQBEGIN 時

交換器載入檔的安全及許可權屬性必須容許這些處理程序執行此動作。

XAOpenString= 字串

這是 IBM MQ 程式碼在其呼叫資料庫管理程式的 xa_open 函數時傳遞的資料字串。這是選用屬性；如果省略它，則會採用長度為零的字串。

佇列管理程式及 IBM MQ 應用程式程序中的程式碼會兩次呼叫 xa_open 函數：

1. 在佇列管理程式啟動時
2. 當您在 IBM MQ 應用程式程序中第一次呼叫 MQBEGIN 時

此字串的格式是每一個資料庫產品特有的格式，將在該產品的說明文件中說明。一般而言，xa_open 字串包含鑑別資訊（使用者名稱及密碼），以容許與佇列管理程式及應用程式程序中的資料庫連線。

從 IBM MQ 8.0.0 Fix Pack 4 開始，當 XAOpenString 包含密碼時，您可以取得 IBM MQ 來保護此資訊，而不是讓密碼在 qm.ini 檔案中以純文字顯示。IBM MQ 會將使用者名稱及密碼（以加密格式）儲存在不同的檔案中，並使用這些認證來連接至資料庫。如需詳細資料，請參閱 [保護資料庫鑑別詳細資料](#)。

XACloseString= 字串

這是 IBM MQ 程式碼在呼叫資料庫管理程式的 xa_close 函數時傳遞的資料字串。這是選用屬性；如果省略它，則會採用長度為零的字串。

佇列管理程式及 IBM MQ 應用程式程序中的程式碼會兩次呼叫 xa_close 函數：

1. 在佇列管理程式啟動時
2. 當您在 IBM MQ 應用程式程序中呼叫 MQDISC 時，先前已呼叫 MQBEGIN

此字串的格式是每一個資料庫產品特有的格式，將在該產品的說明文件中說明。一般而言，字串是空的，且通常會在 XAResourceManager 段落中省略 XACloseString 屬性。

ThreadOfControl=THREAD |PROCESS (預設值)

ThreadOf 控制項值可以是 THREAD 或 PROCESS。佇列管理程式會將它用於序列化。這是選用屬性；如果省略它，則會假設值 PROCESS。

如果資料庫用戶端程式碼容許執行緒呼叫 XA 函數而不進行序列化，則 ThreadOfControl 的值可以是 THREAD。必要的話，佇列管理程式會假設它可以同時從多個執行緒呼叫資料庫用戶端共用程式庫中的 XA 函數。

如果資料庫用戶端程式碼不容許執行緒以這種方式呼叫其 XA 函數，則 ThreadOf 控制項的值必須是 PROCESS。在此情況下，佇列管理程式會序列化對資料庫用戶端共用程式庫的所有呼叫，以便一次只從特定處理程序內進行一次呼叫。您可能還需要確保應用程式在使用多個執行緒執行時執行類似的序列化。

請注意，資料庫產品以這種方式處理多執行緒處理程序的能力問題，是該產品供應商的問題。如需您是否可以將 ThreadOfControl 屬性設為 THREAD 或 PROCESS 的詳細資料，請參閱資料庫產品說明文件。如果可以的話，我們建議您將 ThreadOfControl 設為 THREAD。如果不確定，則更安全選項是將它設為 PROCESS，但您會失去使用 THREAD 的潛在效能好處。

Multi 撰寫及修改應用程式

如何實作廣域工作單元。

第 47 頁的『介紹工作單元』中說明隨 IBM MQ 安裝提供的實務範例 1 廣域工作單元的範例應用程式。

一般而言，廣域工作單元是透過下列方法在應用程式中實作 (在虛擬碼中)：

```
MQBEGIN
MQGET
MQPUT
SQL INSERT
MQCMIT
```

MQBEGIN 的目的是表示廣域工作單元的開始。MQCMIT 的目的是表示廣域工作單元的結束，並使用兩階段確定通訊協定，以所有參與的資源管理程式來完成它。

在 MQBEGIN 與 MQCMIT 之間，佇列管理程式不會對資料庫發出任何呼叫來更新其資源。也就是說，資料庫表格變更的唯一方式是您的程式碼 (例如，虛擬程式碼中的 SQL INSERT)。

就資料庫而言，佇列管理程式的角色是告訴它廣域工作單元何時啟動、何時結束，以及是否應該確定或回復廣域工作單元。

就您的應用程式而言，佇列管理程式會執行兩個角色：資源管理程式 (其中資源是佇列上的訊息) 和廣域工作單元的交易管理程式。

從提供的範例程式開始，並完成在那些程式中所進行的各種 IBM MQ 及資料庫 API 呼叫。範例 IBM MQ 程序化程式、MQI 中使用的資料類型，以及 (如果是資料庫自己的 API) 資料庫自己的文件中，完全記載了相關的 API 呼叫。

Multi 測試系統

您知道應用程式和系統是否只有在測試期間執行時才正確配置。您可以建置並執行其中一個提供的範例程式，來測試系統的配置 (佇列管理程式與資料庫之間的成功通訊)。

Multi 正在配置 Db2

Db2 支援及配置資訊。

支援的 Db2 層次定義在 [IBM MQ 的系統需求](#) 頁面中。

註：在佇列管理程式為 64 位元的平台上，不支援 32 位元 Db2 實例。

請執行下列動作：

1. 請檢查環境變數設定。
2. 建立 Db2 交換器載入檔案。
3. 新增資源管理程式配置資訊。
4. 必要的話，請變更 Db2 配置參數。

請結合第 51 頁的『配置系統以進行資料庫協調』中提供的一般資訊來閱讀本資訊。

警告：如果您在 AIX and Linux 平台上執行 db2profile，則會設定環境變數 LIBPATH 及 LD_LIBRARY_PATH。建議 unset 這些環境變數。如需相關資訊，請參閱 [crtmqenv](#) 或 [setmqenv](#)。

檢查 Db2 環境變數設定

確定已為佇列管理程式處理程序以及中的應用程式程序設定 Db2 環境變數。特別是，您必須一律設定啟動佇列管理程式的 Db2INSTANCE 環境變數之前。DB2INSTANCE 環境變數會識別包含要更新之 Db2 資料庫的 Db2 實例。例如：

- 在 AIX and Linux 系統上，使用：

```
export DB2INSTANCE=db2inst1
```

- 在 Windows 系統上，使用：

```
set DB2INSTANCE=Db2
```

在具有 Db2 資料庫的 Windows 上，您必須將使用者 MUSR_MQADMIN 新增至 DB2USERS 群組，才能啟動佇列管理程式。

建立 Db2 交換器載入檔案

建立 Db2 切換載入檔案最簡單的方法是使用範例檔 xaswit.mak，IBM MQ 提供此範例檔來建置各種資料庫產品的切換載入檔案。

Windows 在 Windows 系統上，您可以在 `MQ_INSTALLATION_PATH\tools\c\samples\xatm` 目錄中找到 `xaswit.mak`。`MQ_INSTALLATION_PATH` 代表 IBM MQ 安裝所在的高階目錄。若要使用 Microsoft Visual C++ 建立 Db2 交換器載入檔案，請使用：

```
nmake /f xaswit.mak db2swit.dll
```

產生的切換檔案放置在 `C:\Program Files\IBM\MQ\exits` 中。

Linux **UNIX** 您可以在 `MQ_INSTALLATION_PATH/samp/xatm` 目錄中找到 `xaswit.mak`。`MQ_INSTALLATION_PATH` 代表 IBM MQ 安裝所在的高階目錄。

編輯 `xaswit.mak`，以解除註解您所使用的 Db2 版本所適用的行。然後使用下列指令執行 `make` 檔：

```
make -f xaswit.mak db2swit
```

產生的 32 位元交換器載入檔案放置在 `/var/mqm/exits` 中。

產生的 64 位元交換器載入檔案位於 `/var/mqm/exits64` 中。

V9.3.0 如果您的系統不支援 32 位元編譯，請僅使用 64 位元目標：

```
make -f xaswit.mak db2swit64
```

新增 Db2 的資源管理程式配置資訊

您必須修改佇列管理程式的配置資訊，以將 Db2 宣告為廣域工作單元中的參與者。如需以這種方式修改配置資訊的詳細資料，請參閱第 52 頁的『將配置資訊新增至佇列管理程式』。

- 在 Windows 和 Linux (x86 和 x86-64 平台) 系統上，請使用 IBM MQ Explorer。在 XA 資源管理程式下的佇列管理程式內容畫面中，指定交換器載入檔的詳細資料。
- 在所有其他系統上，請在佇列管理程式 `qm.ini` 檔的 `XAResourceManager` 段落中指定交換器負載檔案的詳細資料。

第 56 頁的圖 6 是 UNIX 範例，顯示 `XAResourceManager` 項目，其中要協調的資料庫稱為 `mydbname`，此名稱在 `XAOpenString` 中指定：

```
XAResourceManager:  
  Name=mydb2  
  SwitchFile=db2swit  
  XAOpenString=mydbname,myuser,mypasswd,toc=t  
  ThreadOfControl=THREAD
```

圖 6: UNIX 上 Db2 的範例 XAResourceManager 項目

註:

1. ThreadOfControl=THREAD 無法與 8 之前的 Db2 版本搭配使用。將 ThreadOfControl 及 XAOpenString 參數 toc 設為下列其中一種組合:

- ThreadOfControl=THREAD 及 toc=t
- ThreadOfControl=PROCESS 及 toc=p

如果您使用 jdbcdb2 XA 參數載入檔來啟用 JDBC/JTA 協調，則必須使用 ThreadOfControl=PROCESS 和 toc=p。

變更 Db2 配置參數

對於佇列管理程式正在協調的每一個 Db2 資料庫，您必須設定資料庫專用權，變更 tp_mon_name 參數，並重設 maxappls 參數。若要這麼做，請執行下列步驟:

設定資料庫專用權

在 AIX and Linux 系統上，佇列管理程式處理程序以有效使用者和群組 mqm 執行。在 Windows 系統上，它們以啟動佇列管理程式的使用者身分執行。這可以是下列其中一項:

1. 發出 stirmqm 指令的使用者，或
2. 執行 IBM MQ Service COM 伺服器的使用者

依預設，此使用者稱為 MUSR_MQADMIN。

如果您未在 xa_open 字串上指定使用者名稱和密碼，則 Db2 會使用 **執行佇列管理程式的使用者** 來鑑別 xa_open 呼叫。如果此使用者 (例如，AIX and Linux 系統上的使用者 mqm) 在資料庫中沒有最小專用權，則資料庫會拒絕鑑別 xa_open 呼叫。

相同的考量也適用於您的申請程序。如果您未在 xa_open 字串上指定使用者名稱和密碼，則 Db2 會使用執行應用程式的使用者來鑑別在第一個 MQBEGIN 期間進行的 xa_open 呼叫。同樣地，此使用者必須在資料庫中具有最小專用權，此作業才能運作。

例如，透過發出下列 Db2 指令，在 mydbname 資料庫中提供 mqm 使用者連接權限:

```
db2 connect to mydbname  
db2 grant connect on database to user mqm
```

如需安全的相關資訊，請參閱 [第 64 頁的『安全注意事項』](#)。

Windows 變更 TP_MON_NAME 參數

僅適用於 Windows 系統上的 Db2，請變更 TP_MON_NAME 配置參數，以命名 Db2 用來呼叫佇列管理程式以進行動態登錄的 DLL。

使用指令 db2 update dbm cfg using TP_MON_NAME mqmax 來命名 MQMAX.DLL 作為 Db2 用來呼叫佇列管理程式的檔案庫。這必須存在於 PATH 內的目錄中。

重設 maxappls 參數

您可能需要檢閱 maxappls 參數的設定，該參數會限制可連接至資料庫的應用程式數目上限。請參閱 [第 51 頁的『安裝及配置資料庫產品』](#)。

Multi 配置 Oracle

Oracle 支援及配置資訊。

請完成下列步驟：

1. 請檢查環境變數設定。
2. 建立 Oracle 交換器載入檔案。
3. 新增資源管理程式配置資訊。
4. 必要的話，變更 Oracle 配置參數。

如需 IBM MQ 所支援 Oracle 的現行層次清單，請參閱 [IBM MQ 的系統需求](#)。

檢查 Oracle 環境變數設定

請確定已為佇列管理程式處理程序以及應用程式程序中設定 Oracle 環境變數。尤其是在啟動佇列管理程式之前，請一律設定下列環境變數：

ORACLE_HOME

Oracle 起始目錄。例如，在 AIX and Linux 系統上，使用：

```
export ORACLE_HOME=/opt/oracle/product/8.1.6
```

在 Windows 系統上，使用：

```
set ORACLE_HOME=c:\oracle\ora81
```

ORACLE_SID

正在使用的 Oracle SID。如果您使用 Net8 來進行主從式連線功能，則可能不需要設定此環境變數。請參閱 Oracle 文件。

後續範例是在 AIX and Linux 系統上設定此環境變數的範例：

```
export ORACLE_SID=sid1
```

Windows 系統上的對等項目如下：

```
set ORACLE_SID=sid1
```

註：PATH 環境變數必須設為包含二進位檔目錄 (例如 ORACLE_INSTALL_DIR/VERSION/32BIT_NAME/bin 或 ORACLE_INSTALL_DIR/VERSION/64BIT_NAME/bin)，否則您可能會看到一則訊息，指出機器中遺漏 oraclient 程式庫。

如果您在 Windows 64 位元系統上執行佇列管理程式，則只能安裝 64 位元 Oracle 用戶端。由 64 位元佇列管理程式載入的交換器載入檔案必須存取 Oracle 64 位元用戶端程式庫。

建立 Oracle 交換器載入檔案

若要建立 Oracle 交換器載入檔案，請使用 IBM MQ 提供的範例檔案 xaswit.mak，以建置各種資料庫產品的交換器載入檔案。

Windows 在 Windows 系統上，您可以在 C:\Program Files\IBM\MQ\tools\c\samples\xatm 目錄中找到 xaswit.mak。若要使用 Microsoft Visual C++ 來建立 Oracle 交換器載入檔案，請使用：

```
nmake /f xaswit.mak oraswit.dll
```

註：這些交換器載入檔案只能與 C 應用程式搭配使用。若為 Java 應用程式，請參閱 [使用 IBM MQ classes for Java 的 JTA/JDBC 協調](#)。

產生的切換檔案放置在 `MQ_INSTALLATION_PATH\exits` 中。`MQ_INSTALLATION_PATH` 代表 IBM MQ 安裝所在的高階目錄。

Linux **UNIX** 您可以在 `MQ_INSTALLATION_PATH/samp/xatm` 目錄中找到 `xaswit.mak`。`MQ_INSTALLATION_PATH` 代表 IBM MQ 安裝所在的高階目錄。

編輯 `xaswit.mak`，以解除註解您所使用的 Oracle 版本所適用的行。然後使用下列指令執行 `make` 檔：

```
make -f xaswit.mak oraswit
```

安裝 IBM MQ 時，`MQ_INSTALLATION_PATH/samp/xatm` 的內容是唯讀的，因此若要編輯 `xaswit.mak`，請將 `samp/xatm` 中的所有檔案複製到另一個目錄，修改 `xaswit.mak`，然後從該目錄執行 `make -f xaswit.makoraswit`。

產生的 32 位元交換器載入檔案放置在 `/var/mqm/exits` 中。

產生的 64 位元交換器載入檔案放置在 `/var/mqm/exits64` 中。

V 9.3.0 如果您的系統不支援 32 位元編譯，請僅使用 64 位元目標：

```
make -f xaswit.mak oraswit64
```

新增 Oracle 的資源管理程式配置資訊

您必須修改佇列管理程式的配置資訊，以將 Oracle 宣告為廣域工作單元中的參與者。以這種方式修改佇列管理程式的配置資訊，在 [第 52 頁的『將配置資訊新增至佇列管理程式』](#) 中有更詳細的說明。

- 在 Windows 和 Linux (x86 和 x86-64 平台) 系統上，請使用 IBM MQ Explorer。在 XA 資源管理程式下的佇列管理程式內容畫面中，指定交換器載入檔的詳細資料。
- 在所有其他系統上，請在佇列管理程式 `qm.ini` 檔的 `XAResourceManager` 段落中指定交換器載入檔案的詳細資料。

[第 58 頁的圖 7](#) 是顯示 `XAResourceManager` 項目的 AIX 和 Linux 系統範例。您必須將 `LogDir` 新增至 XA 開放式字串，才能將所有錯誤及追蹤資訊記載至相同的位置。

```
XAResourceManager:  
  Name=myoracle  
  SwitchFile=oraswit  
  XAOpenString=Oracle_XA+Acc=P/myuser/mypasswd+SesTm=35+LogDir=/tmp+threads=true  
  ThreadOfControl=THREAD
```

圖 7: AIX 和 Linux 平台上 Oracle 的範例 `XAResourceManager` 項目

註：

1. 在 [第 58 頁的圖 7](#) 中，`xa_open` 字串已與四個參數搭配使用。可以包含其他參數，如 Oracle 的說明文件中所述。
2. 使用 IBM MQ 參數 `ThreadOfControl=THREAD` 時，您必須在 `XAResourceManager` 段落中使用 Oracle 參數 `+threads=true`。

如需 `xa_open` 字串的相關資訊，請參閱 *Oracle8 Server Application Developer 's Guide*。

變更 Oracle 配置參數

對於佇列管理程式正在協調的每一個 Oracle 資料庫，您必須檢閱階段作業數上限，並設定資料庫專用權。如果要這麼做，請完成下列步驟：

檢閱您的階段作業數上限

您可能需要檢閱 LICENSE_MAX_SESSIONS 及 PROCESSES 設定，以考量屬於佇列管理程式之處理程序所需的其他連線。如需詳細資料，請參閱 [第 51 頁的『安裝及配置資料庫產品』](#)。

設定資料庫專用權

xa_open 字串中指定的 Oracle 使用者名稱必須具有存取 DBA_PENDING_TRANSACTIONS 視圖的專用權，如 Oracle 文件中所述。

可以使用下列範例指令來提供必要的專用權：

```
grant select on DBA_PENDING_TRANSACTIONS to myuser;
```

Multi 正在配置 Informix

Informix 支援及配置資訊。

請完成下列步驟：

1. 確定您已安裝適當的 Informix 用戶端 SDK：
 - 32 位元佇列管理程式及應用程式需要 32 位元 Informix 用戶端 SDK。
 - 64 位元佇列管理程式及應用程式需要 64 位元 Informix 用戶端 SDK。
2. 請確定已正確建立 Informix 資料庫。
3. 請檢查環境變數設定。
4. 建置 Informix 交換器載入檔案。
5. 新增資源管理程式配置資訊。

如需 IBM MQ 所支援 Informix 的現行層次清單，請參閱 [IBM MQ 的系統需求](#)。

確定已正確建立 Informix 資料庫

必須指定 **log** 參數來建立每一個要由 IBM MQ 佇列管理程式協調的 Informix 資料庫。例如：

```
create database mydbname with log;
```

IBM MQ 佇列管理程式無法協調建立時未指定 log 參數的 Informix 資料庫。如果佇列管理程式嘗試協調在建立時未指定 log 參數的 Informix 資料庫，則對 Informix 的 xa_open 呼叫會失敗，並產生一些 FFST 錯誤。

檢查 Informix 環境變數設定

確定已為佇列管理程式處理程序 **以及** 中的應用程式程序設定 Informix 環境變數。尤其是在啟動佇列管理程式 **之前**，請一律設定下列環境變數：

INFORMIXDIR

Informix 產品安裝架構的目錄。

- 若為 32 位元 AIX and Linux 應用程式，請使用下列指令：

```
export INFORMIXDIR=/opt/informix/32-bit
```

- 若為 64 位元 AIX and Linux 應用程式，請使用下列指令：

```
export INFORMIXDIR=/opt/informix/64-bit
```

- 若為 Windows 應用程式，請使用下列指令：

```
set INFORMIXDIR=c:\informix
```

對於具有 64 位元佇列管理程式且必須同時支援 32 位元及 64 位元應用程式的系統，您需要同時安裝 Informix 32 位元及 64 位元用戶端 SDK。用於建立交換器載入檔案的範例 make 檔 xaswit.mak 也會同時設定兩個產品安裝目錄。

INFORMIXSERVER

Informix 伺服器名稱。例如，在 AIX and Linux 系統上，使用：

```
export INFORMIXSERVER=hostname_1
```

在 Windows 系統上，使用：

```
set INFORMIXSERVER=hostname_1
```

ONCONFIG

Informix 伺服器配置檔名稱。例如，在 AIX and Linux 系統上，使用：

```
export ONCONFIG=onconfig.hostname_1
```

在 Windows 系統上，使用：

```
set ONCONFIG=onconfig.hostname_1
```

建立 Informix 交換器載入檔案

若要建立 Informix 交換器載入檔案，請使用 IBM MQ 提供的範例檔 xaswit.mak，以建置各種資料庫產品的交換器載入檔案。

Windows 在 Windows 系統上，您可以在 `MQ_INSTALLATION_PATH\tools\c\samples\xatm` 目錄中找到 xaswit.mak。`MQ_INSTALLATION_PATH` 代表 IBM MQ 安裝所在的高階目錄。若要使用 Microsoft Visual C++ 建立 Informix 交換器載入檔案，請使用：

```
nmake /f xaswit.mak infswit.dll
```

產生的交換檔案放置在 `C:\Program Files\IBM\MQ\exits` 中。

Linux **UNIX** 您可以在 `MQ_INSTALLATION_PATH/samp/xatm` 目錄中找到 xaswit.mak。`MQ_INSTALLATION_PATH` 代表 IBM MQ 安裝所在的高階目錄。

編輯 xaswit.mak，以解除註解您所使用的 Informix 版本所適用的字行。然後使用下列指令執行 make 檔：

```
make -f xaswit.mak infswit
```

產生的 32 位元交換器載入檔案放置在 `/var/mqm/exits` 中。

產生的 64 位元交換器載入檔案放置在 `/var/mqm/exits64` 中。

V 9.3.0 如果您的系統不支援 32 位元編譯，請僅使用 64 位元目標：

```
make -f xaswit.mak infswit64
```

新增 Informix 的資源管理程式配置資訊

您必須修改佇列管理程式的配置資訊，以將 Informix 宣告為廣域工作單元中的參與者。以這種方式修改佇列管理程式的配置資訊，在 [第 52 頁的『將配置資訊新增至佇列管理程式』](#) 中有更詳細的說明。

- 在 Windows 和 Linux (x86 和 x86-64 平台) 系統上，請使用 IBM MQ Explorer。在 XA 資源管理程式下的佇列管理程式內容畫面中，指定交換器載入檔的詳細資料。
- 在所有其他系統上，請在佇列管理程式 qm.ini 檔的 XAResourceManager 段落中指定交換器載入檔案的詳細資料。

第 61 頁的圖 8 是 UNIX 範例，顯示 qm.ini XAResourceManager 項目，其中要協調的資料庫稱為 mydbname，此名稱在 XAOpenString 中指定：

```
XAResourceManager:
  Name=myinformix
  SwitchFile=infswit
  XAOpenString=DB=mydbname@myinformixserver\;USER=myuser\;PASSWD=mypasswd
  ThreadOfControl=THREAD
```

圖 8: UNIX 上 Informix 的範例 XAResourceManager 項目

註：依預設，UNIX 上的範例 xaswit.mak 會建立使用執行緒 Informix 程式庫的交換器載入檔案。使用這些 Informix 程式庫時，您必須確定 ThreadOfControl 設為 THREAD。在第 61 頁的圖 8 中，qm.ini 檔 XAResourceManager 段落屬性 ThreadOfControl 控制項設為 THREAD。當指定 THREAD 時，必須使用含執行緒作業的 Informix 程式庫及 IBM MQ 含執行緒作業的 API 程式庫來建置應用程式。

XAOpenString 屬性必須包含資料庫名稱，後面接著 @ 符號，然後再接著 Informix 伺服器名稱。

若要使用非執行緒 Informix 程式庫，您必須確保 qm.ini 檔 XAResourceManager 段落屬性 ThreadOfControl 設為 PROCESS。您也必須對範例 xaswit.mak 進行下列變更：

1. 解除註解非執行緒交換器載入檔案的產生。
2. 註銷產生執行緒交換器載入檔案。

Multi Sybase 配置

Sybase 支援及配置資訊。

請完成下列步驟：

1. 請確定您已安裝 Sybase XA 程式庫，例如安裝 XA DTM 選項。
2. 請檢查環境變數設定。
3. 啟用 Sybase XA 支援。
4. 建立 Sybase 交換器載入檔案。
5. 新增資源管理程式配置資訊。

如需 IBM MQ 所支援 Sybase 的現行層次清單，請參閱 [IBM MQ 的系統需求](#)。

檢查 Sybase 環境變數設定

請確定已針對佇列管理程式程序以及應用程式程序中的 Sybase 環境變數設定。尤其是在啟動佇列管理程式之前，請一律設定下列環境變數：

SYBASE

Linux

AIX

Sybase 產品安裝架構的位置。例如，在 AIX and Linux 系統上，使用：

```
export SYBASE=/sybase
```

Windows

在 Windows 系統上，使用：

```
set SYBASE=c:\sybase
```

SYBASE_OCS

SYBASE 下您已安裝 Sybase 用戶端檔案的目錄。

Linux **AIX** 例如，在 AIX and Linux 系統上，使用：

```
export SYBASE_OCS=OCS-12_0
```

Windows 在 Windows 系統上，使用：

```
set SYBASE_OCS=OCS-12_0
```

啟用 Sybase XA 支援

在 Sybase XA 配置檔 \$SYBASE/\$SYBASE_OCS/xa_config 內，為每一個要更新的 Sybase 伺服器連線定義一個「邏輯 Resource Manager (LRM)」。

以下是 \$SYBASE/\$SYBASE_OCS/xa_config 內容的範例：

```
# The first line must always be a comment
[xa]
LRM=lrname
server=servername
```

建立 Sybase 交換器載入檔案

若要建立 Sybase 交換器載入檔案，請使用 IBM MQ 隨附的範例檔案。

Windows 在 Windows 系統上，您可以在 C:\Program Files\IBM\MQ\tools\c\samples\xatm 目錄中找到 xaswit.mak。若要使用 Microsoft Visual C++ 建立 Sybase 交換器載入檔，請使用：

```
nmake /f xaswit.mak sybswit.dll
```

產生的交換檔案放置在 C:\Program Files\IBM\MQ\exits 中。

Linux **UNIX** 您可以在 MQ_INSTALLATION_PATH/samp/xatm 目錄中找到 xaswit.mak。MQ_INSTALLATION_PATH 代表 IBM MQ 安裝所在的高階目錄。

編輯 xaswit.mak，以解除註解您正在使用的 Sybase 版本所適用的行。然後使用下列指令執行 make 檔：

```
make -f xaswit.mak sybswit
```

產生的 32 位元交換器載入檔案放置在 /var/mqm/exits 中。

產生的 64 位元交換器載入檔案位於 /var/mqm/exits64 中。

V9.3.0 如果您的系統不支援 32 位元編譯，請僅使用 64 位元目標：

```
make -f xaswit.mak sybswit64
```

註： **AIX** 在 AIX 上，範例 make 檔已如下列範例所示進行修改，因此您可以選取不同的 SYBLINKFLAG64 值，視您是使用 Sybase 15 ESD#5 或更新版本或舊版 Sybase 而定。

```
SYBLINKFLAGS32=-brtl
# The following line is for Sybase 15
#SYBLINKFLAGS64=-brtl
```

```
# The following line is for Sybase 16
SYBLINKFLAGS64=-bstatic -bdynamic
```

您需要對 make 檔進行的唯一變更是確保只解除註解其中一個 SYBLINKFLAGS64 值。預設值為 Sybase 16，這是用於 15 #ESD5 以及更新版本的值。

所產生的任何 XA 交換器檔案都會鏈結至該特定版本的 Sybase，且不得移至其他平台。

如果變更 Sybase 的層次，則應該重建 XA 交換器檔案。

新增 Sybase 的資源管理程式配置資訊

您必須修改佇列管理程式的配置資訊，以將 Sybase 宣告為廣域工作單元中的參與者。如需修改配置資訊的詳細說明，請參閱第 52 頁的『將配置資訊新增至佇列管理程式』。

- **Windows** **Linux** 在 Windows 和 Linux (x86 和 x86-64 平台) 系統上，請使用 IBM MQ Explorer。在 XA 資源管理程式下的佇列管理程式內容畫面中，指定交換器載入檔的詳細資料。
- 在所有其他系統上，請在佇列管理程式的 `qm.ini` 檔案中的 XAResourceManager 段落中指定交換器負載檔案的詳細資料。

Linux **AIX** 下列範例顯示 AIX 和 Linux 上 Sybase 的範例 XAResourceManager 項目，其使用與 Sybase XA 配置檔 `$SYBASE/$SYBASE_OCS/xa_config` 中 `lrmname` LRM 定義相關聯的資料庫。如果您想要記載 XA 函數呼叫，請包含日誌檔名稱：

```
XAResourceManager:
  Name=mysybase
  SwitchFile=sybswit
  XAOpenString=-Uuser -Ppassword -Nlrmname -L/tmp/sybase.log -Txa
  ThreadOfControl=THREAD
```

搭配使用多執行緒程式與 Sybase

如果您使用具有 IBM MQ 廣域工作單元的多執行緒程式，並將更新項目合併至 Sybase，則您必須對 ThreadOf 控制參數使用值 THREAD。此外，請確定您將程式 (及切換載入檔案) 與安全執行緒 Sybase 程式庫 (_r 版本) 鏈結。前一個範例顯示對 ThreadOf 控制參數使用值 THREAD。

Multi 多個資料庫配置

如果您想要配置佇列管理程式，以便將多個資料庫的更新項目包含在廣域工作單元內，請為每一個資料庫新增 XAResourceManager 段落。

如果資料庫全部由相同的資料庫管理程式管理，則每一個段落都會定義個別資料庫。每一個段落都指定相同的 SwitchFile，但是 XAOpenString 的內容不同，因為它指定要更新的資料庫名稱。例如，第 64 頁的圖 9 中顯示在 AIX 和 Linux 系統上使用 Db2 資料庫 `MQBankDB` 及 `MQFeeDB` 來配置佇列管理程式的段落。

重要：您不能有多個段落指向相同的資料庫。此配置在任何情況下都無法運作，如果您嘗試此配置，則會失敗。

您將收到 when the MQ code makes its second xa_open call in any process in this environment, the database software fails the second xa_open with a -5 error, XAER_INVAL 格式的錯誤。

```
XAResourceManager:  
Name=DB2 MQBankDB  
SwitchFile=db2swit  
XAOpenString=MQBankDB
```

```
XAResourceManager:  
Name=DB2 MQFeeDB  
SwitchFile=db2swit  
XAOpenString=MQFeeDB
```

圖 9: 多個 Db2 資料庫的範例 XAResourceManager 項目

如果要更新的資料庫是由不同的資料庫管理程式所管理，請為每一個資料庫管理程式新增 XAResourceManager 段落。在此情況下，每一個段落都會指定不同的 *SwitchFile*。例如，如果 *MQFeeDB* 由 Oracle 而非 Db2 管理，請在 AIX and Linux 系統上使用下列段落：

```
XAResourceManager:  
Name=DB2 MQBankDB  
SwitchFile=db2swit  
XAOpenString=MQBankDB
```

```
XAResourceManager:  
Name=Oracle MQFeeDB  
SwitchFile=oraswit  
XAOpenString=Oracle_XA+Acc=P/myuser/mypassword+SesTm=35+LogDir=/tmp/ora.log+DB=MQFeeDB
```

圖 10: Db2 和 Oracle 資料庫的範例 XAResourceManager 項目

原則上，可以使用單一佇列管理程式來配置的資料庫實例數目沒有限制。

註：如需支援將 Informix 資料庫併入廣域工作單元內多個資料庫更新項目的相關資訊，請檢查產品 Readme 檔。

Multi 安全注意事項

在 XA 模型下執行資料庫的考量。

下列資訊僅作為指引提供。無論如何，請參閱資料庫管理程式所提供的文件，以判斷在 XA 模型下執行資料庫的安全含意。

應用程式程序表示使用 `MQBEGIN` 動詞來啟動廣域工作單元。應用程式發出的第一個 `MQBEGIN` 呼叫會在 `xa_open` 進入點呼叫其用戶端程式庫程式碼，以連接至所有參與資料庫。所有資料庫管理程式都提供在其 `XAOpenString` 中提供使用者 ID 和密碼的機制。這是鑑別資訊唯一流動的時間。

請注意，在 AIX and Linux 平台上，在進行 MQI 呼叫時，捷徑應用程式必須以有效使用者 ID `mqm` 來執行。

Multi 失去與 XA 資源管理程式的聯絡時的考量

佇列管理程式會容忍無法使用資料庫管理程式。這表示您可以獨立於資料庫伺服器啟動及停止佇列管理程式。當還原聯絡時，佇列管理程式和資料庫會重新同步化。您也可以使用 `rsvmqtrn` 指令來手動解決不確定的工作單元。

在一般作業中，完成配置步驟之後，只需要最少的管理量。管理工作更容易，因為佇列管理程式容忍資料庫管理程式無法使用。具體而言，這意味著：

- 佇列管理程式可以隨時啟動，而不需要先啟動每一個資料庫管理程式。
- 如果其中一個資料庫管理程式變成無法使用，則佇列管理程式不需要停止並重新啟動。

這可讓您獨立於資料庫伺服器來啟動和停止佇列管理程式。

每當佇列管理程式與資料庫之間失去聯絡時，當兩者再次可用時，它們需要重新同步化。重新同步是指涉及該資料庫的任何不確定工作單元完成的處理程序。一般而言，這會自動發生，而不需要使用者介入。佇列管理程式會向資料庫要求不確定的工作單元清單。然後，它會指示資料庫確定或回復每一個不確定的工作單元。

當佇列管理程式啟動時，它會與每一個資料庫重新同步化。當個別資料庫變成無法使用時，只有在下次佇列管理程式發現該資料庫再次可用時，才需要重新同步化該資料庫。

當使用 MQBEGIN 啟動新的廣域工作單元時，佇列管理程式會自動重新取得與先前無法使用之資料庫的聯絡。其作法是呼叫資料庫用戶端程式庫中的 xa_open 函數。如果此 xa_open 呼叫失敗，則 MQBEGIN 會傳回完成碼 MQCC_WARNING 及原因碼 MQRC_PARTICIPANT_NOT_AVAILABLE。您可以稍後重試 MQBEGIN 呼叫。

請勿繼續嘗試廣域工作單元，其中涉及在 MQBEGIN 期間指出失敗的資料庫更新。將不會有可用來進行更新的資料庫連線。您的唯一選項是結束程式，或定期重試 MQBEGIN，希望資料庫可能再次變成可用。

或者，您可以使用 rsvmqtrn 指令來明確解決所有不確定的工作單元。

Multi 不確定的工作單元

如果在指示資料庫管理程式準備之後失去與佇列管理程式的聯絡，則資料庫可能具有不確定的工作單元。在資料庫伺服器接收來自佇列管理程式的輸出(確定或回復)之前，它需要保留與更新項目相關聯的資料庫鎖定。

因為這些鎖定會阻止其他應用程式更新或讀取資料庫記錄，所以需要儘快進行重新同步。

如果基於某種原因，您無法等待佇列管理程式自動與資料庫重新同步化，則可以使用資料庫管理程式提供的機能來手動確定或回復資料庫更新項目。在「X/Open 分散式交易處理:XA 規格」中，這稱為制定探索性決策。僅將其用作最後手段，因為可能會損害資料完整性；例如，當所有其他參與者都已確定其更新項目時，您可能會錯誤地回復資料庫更新項目。

最好重新啟動佇列管理程式，或在資料庫重新啟動時使用 rsvmqtrn 指令來起始自動重新同步。

Multi 使用 dspmqtrn 指令顯示未完成的工作單元

您可以搭配使用 dspmqtrn 指令與 -i 參數，以顯示內部產生的不確定交易。

當資料庫管理程式無法使用時，您可以使用 dspmqtrn 指令來檢查涉及該資料庫之未完成廣域工作單元的狀態。

dspmqtrn 指令只會顯示一或多個參與者不確定的工作單元。參與者正在等待佇列管理程式的決定，以確定或回復準備好的更新項目。

對於其中每一個廣域工作單元，每一個參與者的狀態都會顯示在來自 dspmqtrn 的輸出中。如果工作單元未更新特定資源管理程式的資源，則不會顯示該工作單元。

關於不確定的工作單元，據說資源管理程式已完成下列其中一項：

已備妥

資源管理程式已準備好確定其更新項目。

已確定

資源管理程式已確定其更新項目。

已回復

資源管理程式已回復其更新項目。

已參與

資源管理程式是參與者，但尚未準備、確定或回復其更新項目。

當佇列管理程式重新啟動時，它會要求每一個具有 XAResourceManager 段落的資料庫，以取得其不確定的廣域工作單元清單。如果資料庫尚未重新啟動或無法使用，則佇列管理程式還無法將那些工作單元的最終結果遞送至資料庫。當資料庫再次可用時，會第一次將不確定工作單元的結果遞送至資料庫。

在此情況下，資料庫管理程式會報告為處於 prepared 狀態，直到發生重新同步為止。

每當 dspmqtrn 指令顯示不確定的工作單元時，它會先列出可能參與的所有可能的資源管理程式。這些會配置唯一 ID RMIId，用來取代資源管理程式在報告其與不確定工作單元相關的狀態時所使用的名稱。

範例 `dspmqrn` 輸出 顯示發出下列指令的結果:

```
dspmqrn -m MY_QMGR
```

```
AMQ7107: Resource manager 0 is MQSeries.  
AMQ7107: Resource manager 1 is DB2 MQBankDB.  
AMQ7107: Resource manager 2 is DB2 MQFeedB.  
  
AMQ7056: Transaction number 0,1.  
XID: formatID 5067085, gtrid_length 12, bqual_length 4  
gtrid [3291A5060000201374657374]  
bqual [00000001]  
AMQ7105: Resource manager 0 has committed.  
AMQ7104: Resource manager 1 has prepared.  
AMQ7104: Resource manager 2 has prepared.
```

其中 交易號碼 是可與 `rsvmqtrn` 指令搭配使用的交易 ID。如需進一步資訊，請參閱 AMQ7xxx: IBM MQ 產品訊息。XID 變數是 X/Open XA 規格的一部分; 如需此規格的最新相關資訊，請參閱: <https://publications.opengroup.org/c193>。

圖 11: `dspmqrn` 輸出範例

`dspmqrn` 輸出範例 中的輸出顯示有三個資源管理程式與佇列管理程式相關聯。第一個是資源管理程式 0，它是佇列管理程式本身。另外兩個資源管理程式實例是 MQBankDB 和 MQFeedB Db2 資料庫。

此範例僅顯示單一不確定的工作單元。這三個資源管理程式都會發出訊息，這表示已在工作單元內對佇列管理程式及這兩個 Db2 資料庫進行更新。

已 確定對佇列管理程式 (資源管理程式 0) 所做的更新。Db2 資料庫的更新項目處於 `prepared` 狀態，這表示在呼叫 Db2 以確定 MQBankDB 和 MQFeedB 資料庫的更新項目之前，必須已變成無法使用。

不確定的工作單元具有稱為 XID (交易 ID) 的外部 ID。這是佇列管理程式提供給 Db2 的資料片段，用來識別其在廣域工作單元中的部分。

如需相關資訊，請參閱 `dspmqrn`。

Multi 使用 `rsvmqtrn` 指令解決未完成的工作單元

當佇列管理程式與 Db2 重新同步化時，完成未完成的工作單元。

第 66 頁的圖 11 中顯示的輸出顯示單一不確定的工作單元，在該工作單元中，尚未將確定決策遞送至兩個 Db2 資料庫。

若要完成此工作單元，當 Db2 下一次變成可用時，佇列管理程式及 Db2 需要重新同步化。佇列管理程式會使用新工作單元的開始，作為重新聯絡 Db2 的機會。或者，您可以指示佇列管理程式使用 `rsvmqtrn` 指令明確地重新同步化。

請在重新啟動 Db2 之後立即執行此動作，以便儘快釋放與不確定工作單元相關聯的任何資料庫鎖定。請使用 `-a` 選項，它會告知佇列管理程式解決所有不確定的工作單元。在下列範例中，Db2 已重新啟動，因此佇列管理程式可以解決不確定的工作單元:

```
> rsvmqtrn -m MY_QMGR -a  
Any in-doubt transactions have been resolved.
```

Multi 混合結果和錯誤

雖然佇列管理程式使用兩段式確定通訊協定，但這不會完全排除部分工作單元完成混合輸出的可能性。這是部分參與者確定其更新項目及部分取消其更新項目的位置。

以混合結果完成的工作單元具有嚴重影響，因為應該作為單一工作單元更新的共用資源不再處於一致狀態。

混合結果主要是在對工作單元做出探索性決策時所造成，而不是讓佇列管理程式自行解決不確定的工作單元。這類決策不受佇列管理程式控制。

每當佇列管理程式偵測到混合結果時，它會產生 FFST 資訊，並在其錯誤日誌中記載失敗，其中包含下列兩則訊息之一：

- 如果資料庫管理程式回復而非確定：

```
AMQ7606 A transaction has been committed but one or more resource
managers have rolled back.
```

- 如果資料庫管理程式確定而非回復：

```
AMQ7607 A transaction has been rolled back but one or more resource
managers have committed.
```

進一步訊息會識別探索性損壞的資料庫。因此，您必須負責在本端還原受影響資料庫的一致性。這是複雜的程序，您需要先隔離錯誤確定或回復的更新，然後手動復原或重做資料庫變更。

Multi 變更配置資訊

順利啟動佇列管理程式來協調廣域工作單元之後，請勿變更任何資源管理程式配置資訊。

如果您需要變更配置資訊，您隨時可以這麼做，但在重新啟動佇列管理程式之後，變更才會生效。

如果您移除資料庫的資源管理程式配置資訊，則實際上會移除佇列管理程式聯絡該資料庫管理程式的能力。

絕不 在任何資源管理程式配置資訊中變更名稱屬性。此屬性唯一地向佇列管理程式識別該資料庫管理程式實例。如果您變更這個唯一 ID，佇列管理程式會假設資料庫已移除，且已新增一個全新的實例。佇列管理程式仍會將未完成的工作單元與舊的名稱相關聯，可能讓資料庫處於不確定的狀態。

Multi 移除資料庫管理程式實例

如果您需要永久地從配置中移除資料庫，在重新啟動佇列管理程式之前，請確定資料庫未處於不確定的狀態。

資料庫產品提供用於列出不確定交易的指令。如果有任何不確定的交易，請先容許佇列管理程式與資料庫重新同步化。透過啟動佇列管理程式來執行此動作。您可以使用 **rsvmqtrn** 指令或資料庫自己的指令來檢視不確定的工作單元，以驗證是否已進行重新同步。一旦您確定已進行重新同步，請結束佇列管理程式並移除資料庫的配置資訊。

如果您無法觀察此程序，佇列管理程式仍會記住涉及該資料庫的所有不確定工作單元。每次重新啟動佇列管理程式時，都會發出警告訊息 AMQ7623。如果您再也不要使用佇列管理程式來配置此資料庫，請使用 **rsvmqtrn** 指令的 **-r** 選項來指示佇列管理程式忘記資料庫參與其不確定的交易。只有在所有參與者都已完成不確定的交易時，佇列管理程式才會忘記這類交易。

有時您可能需要暫時移除某些資源管理程式配置資訊。在 AIX and Linux 系統上，這最好是透過註銷段落來達成，以便稍後可以輕鬆恢復該段落。如果每次佇列管理程式聯絡特定資料庫或資料庫管理程式時都發生錯誤，您可以決定這麼做。暫時移除相關的資源管理程式配置資訊，可讓佇列管理程式啟動涉及所有其他參與者的廣域工作單元。以下是註銷的 XAResourceManager 段落的範例：

```
# This database has been temporarily removed
#XAResourceManager:
# Name=mydb2
# SwitchFile=db2swit
# XAOpenString=mydbname,myuser,mypassword,toc=t
# ThreadOfControl=THREAD
```

圖 12: 在 AIX and Linux 系統上註銷 XAResourceManager 段落

在 Windows 系統上，使用 IBM MQ Explorer 來刪除資料庫管理程式實例的相關資訊。在重新安裝時，請小心在名稱欄位中鍵入正確的名稱。如果您輸入錯誤名稱，可能會遇到不確定的問題，如第 67 頁的『變更配置資訊』中所述。

Multi XA 動態登錄

XA 規格提供一種方法來減少交易管理程式對資源管理程式發出的 xa_* 呼叫數。此最佳化稱為 動態登錄。

Db2 支援動態登錄。其他資料庫可能支援它;如需詳細資料,請參閱資料庫產品的說明文件。

為何動態登錄最佳化有用? 在您的應用程式中,部分廣域工作單元可能包含資料庫表格的更新項目;其他可能不包含此類更新項目。未對資料庫的表格進行持續更新時,不需要在 MQCMIT 期間發生的確定通訊協定中併入該資料庫。

無論資料庫是否支援動態登錄,在 IBM MQ 連線上的第一個 MQBEGIN 呼叫期間,您的應用程式會呼叫 xa_open。它也會在後續 MQDISC 呼叫上呼叫 xa_close。後續 XA 呼叫的型樣取決於資料庫是否支援動態登錄:

如果您的資料庫不支援動態登錄 ...

每個廣域工作單元都涉及 IBM MQ 程式碼對資料庫用戶端程式庫進行的數個 XA 函數呼叫,不論您是否對工作單元內該資料庫的表格進行持續更新。這些包括:

- 應用程式程序中的 xa_start 和 xa_end。這些用來宣告廣域工作單元的開始和結束。
- 佇列管理程式代理程式處理程序 amqzlaa0 中的 xa_prepare、xa_commit 及 xa_rollback。這些是用來交付廣域工作單元的結果:確定或回復決策。

此外,在第一個 MQBEGIN 期間,佇列管理程式代理程式處理程序也會呼叫 xa_open。

如果您的資料庫支援動態登錄 ...

IBM MQ 程式碼只會發出那些必要的 XA 函數呼叫。對於不 涉及資料庫資源持續更新的廣域工作單元,沒有 資料庫的 XA 呼叫。對於已 涉及這類持續更新的廣域工作單元,呼叫如下:

- 來自應用程式程序的 xa_end,以宣告廣域工作單元的結束。
- 佇列管理程式代理程式處理程序 amqzlaa0 中的 xa_prepare、xa_commit 及 xa_rollback。這些是用來交付廣域工作單元的結果:確定或回復決策。

若要讓動態登錄運作,務必要讓資料庫具有告知 IBM MQ 它已執行持續更新的方式,該持續更新要包含在現行廣域工作單元中。IBM MQ 為此目的提供 ax_reg 函數。

在應用程式程序中執行的資料庫用戶端程式碼會尋找 ax_reg 函數並呼叫它,以動態登錄 它已在現行廣域工作單元內完成持續性工作的事實。為了回應此 ax_reg 呼叫,IBM MQ 會記錄資料庫已參與的記錄。如果這是此 IBM MQ 連線上的第一個 ax_reg 呼叫,則佇列管理程式代理程式會處理呼叫 xa_open。

當資料庫用戶端程式碼在您的處理程序中執行時(例如,在 SQL UPDATE 呼叫期間,或在資料庫用戶端 API 中的任何呼叫負責),它會發出此 ax_reg 呼叫。

Multi 錯誤條件

在 XA 動態登錄中,佇列管理程式中可能發生令人混淆的失敗。

常見的範例是如果您在啟動佇列管理程式之前忘記適當地設定資料庫環境變數,則佇列管理程式對 xa_open 的呼叫會失敗。無法使用廣域工作單元。

若要避免此情況,在啟動佇列管理程式之前,請確定您已設定相關的環境變數。檢閱資料庫產品的說明文件,以及第 54 頁的『正在配置 Db2』、第 57 頁的『配置 Oracle』和 第 61 頁的『Sybase 配置』中提供的建議。

在所有資料庫產品中,佇列管理程式會在佇列管理程式啟動時呼叫 xa_open 一次,以作為回復階段作業的一部分(如第 64 頁的『失去與 XA 資源管理程式的聯絡時的考量』中所述)。如果您不正確地設定資料庫環境變數,則此 xa_open 呼叫會失敗,但不會導致佇列管理程式無法啟動。這是因為資料庫用戶端程式庫使用相同的 xa_open 錯誤碼來指出資料庫伺服器無法使用。IBM MQ 不會將此視為嚴重錯誤,因為佇列管理程式必須能夠開始在涉及該資料庫的廣域工作單元之外繼續處理資料。

在 IBM MQ 連線上的第一個 MQBEGIN 期間(如果未使用動態登錄),或在資料庫用戶端程式碼呼叫 IBM MQ 提供的 ax_reg 函數期間(如果使用動態登錄),會從佇列管理程式對 xa_open 進行後續呼叫。

任何錯誤狀況(或偶爾 FFST 報告)的計時取決於您是否使用動態登錄:

- 如果您使用動態登錄,則 MQBEGIN 呼叫可能成功,但 SQL UPDATE (或類似) 資料庫呼叫將失敗。
- 如果您不是使用動態登錄,則 MQBEGIN 呼叫會失敗。

請確保在應用程式及佇列管理程式處理程序中正確設定環境變數。

Multi 彙總 XA 呼叫

以下是由於控制廣域工作單元的各種 MQI 呼叫而對資料庫用戶端程式庫中 XA 函數進行的呼叫清單。這不是 XA 規格中所說明之通訊協定的完整說明；其提供為簡要概觀。

請注意，在應用程式程序中，`xa_start` 和 `xa_end` 呼叫一律由 IBM MQ 程式碼呼叫，而 `xa_prepare`、`xa_commit` 和 `xa_rollback` 一律從佇列管理程式代理程式處理程序 `amqzlaa0` 呼叫。

此表格中顯示的 `xa_open` 及 `xa_close` 呼叫都是從應用程式程序進行。在 [第 68 頁的『錯誤條件』](#) 中說明的情況下，佇列管理程式代理程式處理程序會呼叫 `xa_open`。

MQI 呼叫	使用動態登錄所進行的 XA 呼叫	在沒有動態登錄的情況下進行的 XA 呼叫
第一個 MQBEGIN	<code>xa_open</code>	<code>xa_open</code> <code>xa_start</code>
後續的 MQBEGIN	無 XA 呼叫	<code>xa_start</code>
MQCMIT (在現行廣域工作單元期間未呼叫 <code>ax_reg</code>)	無 XA 呼叫	<code>xa_end</code> <code>xa_prepare</code> <code>xa_commit</code> <code>xa_rollback</code>
MQCMIT (在現行廣域工作單元期間呼叫與 <code>ax_reg</code>)	<code>xa_end</code> <code>xa_prepare</code> <code>xa_commit</code> <code>xa_rollback</code>	不適用。在非動態模式下，不會對 <code>ax_reg</code> 進行呼叫。
MQBACK (在現行廣域工作單元期間未呼叫 <code>ax_reg</code>)	無 XA 呼叫	<code>xa_end</code> <code>xa_rollback</code>
MQBACK (在現行廣域工作單元期間呼叫與 <code>ax_reg</code>)	<code>xa_end</code> <code>xa_rollback</code>	不適用。在非動態模式下，不會對 <code>ax_reg</code> 進行呼叫。
MQDISC，其中先呼叫 MQCMIT 或 MQBACK。如果不是，MQCMIT 處理程序會先在 MQDISC 期間完成。	<code>xa_close</code>	<code>xa_close</code>

附註:

- 對於 MQCMIT，如果 `xa_prepare` 成功，則會呼叫 `xa_commit`。否則，會呼叫 `xa_rollback`。

實務範例 2: 其他軟體提供協調

在實務範例 2 中，外部交易管理程式會協調廣域工作單元，在交易管理程式的 API 控制下啟動並確定它們。MQBEGIN、MQCMIT 及 MQBACK 動詞無法使用。

本節說明此實務範例，包括:

- [第 70 頁的『外部同步點協調』](#)

- [第 72 頁的『使用 CICS』](#)
- [第 75 頁的『使用 Microsoft Transaction Server \(COM +\)』](#)

外部同步點協調

廣域工作單元也可以由外部 X/Open XA 相容交易管理程式來協調。在這裡，IBM MQ 佇列管理程式會參與 (但不會協調) 工作單元。

外部交易管理程式協調的廣域工作單元中的控制流程如下：

1. 應用程式告知外部同步點協調程式 (例如 TXSeries) 它想要開始交易。
2. 同步點協調程式會將現行交易告知已知的資源管理程式 (例如 IBM MQ)。
3. 應用程式會對與現行交易相關聯的資源管理程式發出呼叫。例如，應用程式可以對 IBM MQ 發出 MQGET 呼叫。
4. 應用程式向外部同步點協調程式發出確定或取消要求。
5. 同步點協調程式會對每一個資源管理程式發出適當的呼叫 (通常使用兩階段確定通訊協定)，來完成交易。

支援的外部同步點協調程式層次，可針對 IBM MQ 參與定義於 [IBM MQ 的系統需求](#) 的交易，提供兩段式確定程序。

本節其餘部分說明如何啟用外部工作單元。

IBM MQ XA 交換器結構

參與外部協調工作單元的每一個資源管理程式必須提供 XA 交換器結構。這個結構定義資源管理程式的功能，以及同步點協調程式所要呼叫的功能。

IBM MQ 提供此結構的兩個版本：

- *MQRMIXASwitch* 代表靜態 XA 資源管理
- *MQRMIXASwitchDynamic*，用於動態 XA 資源管理

請參閱您的交易管理程式文件，以判斷是否要使用靜態或動態資源管理介面。每當交易管理程式支援它時，我們建議您使用動態 XA 資源管理。

部分 64 位元交易管理程式會將 XA 規格中的 *long* 類型視為 64 位元，而部分會將它視為 32 位元。IBM MQ 支援兩個模型：

- 如果交易管理程式是 32 位元，或交易管理程式是 64 位元，但將 *long* 類型視為 32 位元，請使用 [第 70 頁的表 5](#) 中列出的切換載入檔。
- 如果您的交易管理程式是 64 位元，並將 *long* 類型視為 64 位元，請使用 [第 71 頁的表 6](#) 中列出的切換載入檔。

部分 64 位元交易管理程式會將 *long* 類型視為 64 位元。下列 64 位元交易管理程式已知需要替代 64 位元交換器載入檔案：

- Tuxedo

如果您不確定交易管理程式使用的模型，請參閱您的交易管理程式文件。


平台	切換載入檔名稱 (伺服器)	切換載入檔名稱 (延伸交易式用戶端)
 Windows	<i>mqmx.dll</i>	<i>mqcxa.dll</i>
 AIX (無執行緒)	<i>libmqmx.a</i>	<i>libmqcxa.a</i>
 AIX (執行緒)	<i>libmqmx_r.a</i>	<i>libmqcxa_r.a</i>

表 5: XA 交換器載入檔名稱 (繼續)

平台	切換載入檔名稱 (伺服器)	切換載入檔名稱 (延伸交易式用戶端)
Linux Linux (無執行緒)	libmqmxa.so	libmqcxa.so
Linux Linux (執行緒)	libmqmxa_r.so	libmqcxa_r.so

表 6: 替代 64 位元 XA 交換器載入檔案名稱

平台	切換載入檔名稱 (伺服器)	切換載入檔名稱 (延伸交易式用戶端)
AIX AIX (無執行緒)	libmqmxa64.a	libmqcxa64.a
AIX AIX (執行緒)	libmqmxa64_r.a	libmqcxa64_r.a
Linux Linux (無執行緒)	libmqmxa64.so	libmqcxa64.so
Linux Linux (執行緒)	libmqmxa64_r.so	libmqcxa64_r.so

部分外部同步點協調程式 (非 CICS) 需要每一個參與工作單元的資源管理程式在 XA 交換器結構的名稱欄位中提供其名稱。IBM MQ 資源管理程式名稱是 MQSeries_XA_RMI。

同步點協調程式定義 IBM MQ XA 交換器結構鏈結至它的方式。第 72 頁的『使用 CICS』中提供鏈結 IBM MQ XA 交換器結構與 CICS 的相關資訊。如需將 IBM MQ XA 交換器結構與其他符合 XA 標準的同步點協調程式鏈結的相關資訊，請參閱這些產品隨附的文件。

下列考量適用於搭配使用 IBM MQ 與所有符合 XA 標準的同步點協調程式：

- 在呼叫 MQCONN 之前，預期交易管理程式程式庫程式碼 (在應用程式設計師所呼叫的 API 中執行) 會在某個時間點將 **xa_open** 呼叫至 IBM MQ。

xa_open 呼叫必須在發出 MQCONN 呼叫的相同執行緒上進行。這項需求的原因是 XA 規格要求使用執行緒來隱含環境定義。

請注意，這是在範例程式 amqstxsx.c 中採取的方法。此範例程式假設在 IBM MQ 的函數 tpopen 內，從交易管理程式的程式庫程式碼對進行 **xa_open** 呼叫。

在 MQCONN 呼叫之前，如果未在相同執行緒上進行 **xa_open** 呼叫，則 IBM MQ 佇列管理程式連線將不會與 XA 環境定義相關聯。

如需相關資訊，請參閱 [MQCTL](#)。

- 同步點協調程式在任何 **xa_open** 呼叫上傳遞的 **xa_info** 結構包括 IBM MQ 佇列管理程式的名稱。名稱的格式與傳遞至 MQCONN 呼叫的佇列管理程式名稱相同。如果在 **xa_open** 呼叫上傳遞的名稱為空白，則會使用預設佇列管理程式。

或者，**xa_info** 結構可以包含 *TPM* 和 *AXLIB* 參數的值。*TPM* 參數指定正在使用的交易管理程式。有效值為 CICS、TUXEDO 及 ENCINA。*AXLIB* 參數指定包含交易管理程式的 *ax_reg* 和 *ax_unreg* 函數的程式庫名稱。如需這些參數的相關資訊，請參閱配置延伸交易式用戶端。如果 **xa_info** 結構包含其中一個參數，則除非正在使用預設佇列管理程式，否則會在 *QMNAME* 參數中指定佇列管理程式名稱。

- 一次只能有一個佇列管理程式參與由外部同步點協調程式實例協調的交易。同步點協調程式實際上已連接至佇列管理程式，且遵循一次僅支援一個連線的規則。

- 所有包含外部同步點協調程式呼叫的應用程式，都只能連接至參與外部協調程式所管理之交易的佇列管理程式 (因為它們已有效地連接至該佇列管理程式)。不過，這類應用程式必須發出 MQCONN 呼叫來取得連線控點，並在結束之前發出 MQDISC 呼叫。
- 具有由外部同步點協調程式協調的資源更新項目的佇列管理程式必須在外部同步點協調程式之前啟動。同樣地，同步點協調程式必須在佇列管理程式之前結束。
- 如果外部同步點協調程式異常終止，請先停止並重新啟動佇列管理程式，然後再重新啟動同步點協調程式，以確保在失敗時未確定的任何傳訊作業都已適當解決。

使用 CICS

CICS 是 TXSeries 的其中一個元素。

符合 XA 標準 (並使用兩階段確定程序) 的 TXSeries 版本定義在 [IBM MQ 的系統需求 資訊](#) 中。

IBM MQ 也支援其他交易管理程式。如需受支援軟體相關資訊的鏈結，請參閱 [IBM MQ 的系統需求 網頁](#)。

兩階段確定程序的需求

當您將 CICS 兩階段確定程序與 IBM MQ 搭配使用時，兩階段確定程序的需求。這些需求不適用於 z/OS。

請注意下列需求：

- IBM MQ 和 CICS 必須位於相同的實體機器上。
- IBM MQ 在 IBM MQ MQI client 上不支援 CICS。
- 您必須啟動佇列管理程式，並在 XAD 資源定義段落中指定其名稱，**之前** 您嘗試啟動 CICS。如果您將 IBM MQ 的 XAD 資源定義段落新增至 CICS 區域，則無法啟動 CICS。
- 從單一 CICS 區域一次只能存取一個 IBM MQ 佇列管理程式。
- CICS 交易必須先發出 MQCONN 要求，才能存取 IBM MQ 資源。MQCONN 呼叫必須指定在 CICS 區域 XAD 資源定義段落的 XAOpen 項目上指定的 IBM MQ 佇列管理程式名稱。如果此項目空白，則 MQCONN 要求必須指定預設佇列管理程式。
- 存取 IBM MQ 資源的 CICS 交易必須先從交易發出 MQDISC 呼叫，才能回到 CICS。如果無法這樣做，可能表示 CICS 應用程式伺服器仍在連接中，並保持開放佇列。此外，如果您未安裝作業終止結束程式 (請參閱第 74 頁的『範例作業終止結束程式』)，CICS 應用程式伺服器稍後可能會異常結束，可能在後續交易期間。
- 您必須確保 CICS 使用者 ID (cics) 是 mqm 群組的成員，以便 CICS 程式碼有權呼叫 IBM MQ。



對於在 CICS 環境中執行的交易，佇列管理程式會調整其授權及判斷環境定義的方法，如下所示：

- 佇列管理程式會查詢 CICS 用來執行交易的使用者 ID。這是由「物件權限管理程式」檢查的使用者 ID，用於環境定義資訊。
- 在訊息環境定義中，應用程式類型是 MQAT_CICS。
- 環境定義中的應用程式名稱會從 CICS 交易名稱複製。

ALW 一般 XA 支援

提供 XA 交換器載入模組可讓您鏈結 CICS 與 IBM MQ for AIX, Linux, and Windows 系統。此外，還提供了範例原始碼檔案，可讓您開發其他交易訊息的 XA 交換器。 **IBM i** IBM i 不支援一般 XA。

提供的交換器載入模組名稱如下：

表 7: CICS 應用程式的基本程式碼 :XA 起始設定常式	
C (來源)	C (exec)-將下列其中一項新增至您的 XAD.Stanza
amqzscix.c	 amqzsc- TXSeries for AIX 5.1
amqzscin.c	 mqmc4swi - TXSeries for Windows 5.1

Multi

建置程式庫以與 *TXSeries for Multiplatforms* 搭配使用

建置程式庫以與 *TXSeries for Multiplatforms* 搭配使用時，請使用此資訊。

預先建置的交換器載入檔案是您可以與 CICS 程式搭配使用的共用程式庫 (在 Windows 系統上稱為 *DLL*)，這些程式需要使用 XA 通訊協定進行兩階段確定交易。這些預先建置程式庫的名稱位於表格 CICS 應用程式的必要程式碼 :XA 起始設定常式 中。下列目錄中也提供範例原始碼：

平台	目錄	來源檔
Linux and Linux AIX	<i>MQ_INSTALLATION_PATH</i> /samp/	amqzscix.c
Windows	<i>MQ_INSTALLATION_PATH</i> \Tools\ c\範例	amqzscin.c

其中 *MQ_INSTALLATION_PATH* 是您安裝 IBM MQ 的目錄。

註：您必須使用針對與交換器載入檔案搭配使用的 IBM MQ 版本所建置的 CICS 交換器載入檔案。

若要從範例來源建置交換器載入檔案，請遵循適合您作業系統的指示：

AIX

發出下列指令：

```
export MQM_HOME=/usr/mqm
echo "amqzscix" > tmp.exp
xlc_r $MQM_HOME/samp/amqzscix.c -I/usr/lpp/cics/include -I$MQM_HOME/inc -e amqzscix -bE:tmp.exp -bM:SRE
-o amqzsc /usr/lpp/cics/lib/regxa_swxa.o -L$MQM_HOME/lib -L/usr/lpp/cics/lib -lcicsrt -lEncina
-lEncServer -lpthreads -lsarpc -lmqmcics_r -lmqmx_r -lmqzi_r -lmqmc_r
rm tmp.exp
```

Linux

發出下列指令：

```
gcc -m32 -shared -fPIC -o amqzscix amqzscix.c
\ -IMQ_INSTALLATION_PATH/inc -I CICS_INSTALLATION_PATH/include
\ -LMQ_INSTALLATION_PATH/lib -Wl,-rpath=MQ_INSTALLATION_PATH/lib
\ -Wl,-rpath=/usr/lib -Wl,-rpath-link,/usr/lib -Wl,--no-undefined
-Wl,--allow-shlib-undefined -L CICS_LIB_PATH/regxa_swxa.o \-lpthread -ldl -lc
-shared -lmqzi_r -lmqmx_r -lmqmcics_r -ldl -lc
```

Windows

請遵循下列步驟：

1. 透過至少編譯下列變數，使用 `cl` 指令來建置 `amqzscin.obj`：

```
cl.exe -c -I EncinaPath\include -I MQ_INSTALLATION_PATH\include -Gz -LD amqzscin.c
```

2. 建立名為 `mqmc1415.def` 的模組定義檔，其中包含下列各行：

```
LIBRARY MQMC4SWI
EXPORTS
CICS_XA_Init
```

3. 至少使用下列選項，使用 `lib` 指令來建置匯出檔及匯入程式庫：

```
lib -def:mqmc4swi.def -out:mqmc4swi.lib
```

如果 `lib` 指令成功，也會建置 `mqmc4swi.exp` 檔。

4. 使用鏈結指令，至少使用下列選項來建置 mqmc4swi.dll：

```
link.exe -dll -nod -out:mqmc4swi.dll
amqzscin.obj CicsPath\lib\regxa_swxa.obj
mqmc4swi.exp mqmcics4.lib
CicsPath\lib\libcicsrt.lib
DcePath\lib\libdce.lib DcePath\lib\pthreads.lib
EncinaPath\lib\libEncina.lib
EncinaPath\lib\libEncServer.lib
msvcrt.lib kernel32.lib
```

IBM MQ XA 支援及 Tuxedo

AIX, Linux, and Windows 系統上的 IBM MQ 可以在 xa_start 中無限期地封鎖 Tuxedo 協調的 XA 應用程式。

只有在單一廣域交易中 Tuxedo 所協調的兩個以上處理程序嘗試使用相同的交易分支 ID (XID) 來存取 IBM MQ 時，才會發生此情況。如果 Tuxedo 為廣域交易中的每一個處理程序提供不同的 XID，以與 IBM MQ 搭配使用，則無法發生此情況。

若要避免此問題，請在 Tuxedo 中配置每一個應用程式，在其自己的 Tuxedo 伺服器群組內，以單一廣域交易 ID (gtrid) 來存取 IBM MQ。當代表單一 gtrid 存取資源管理程式時，相同伺服器群組中的處理程序會使用相同的 XID，因此在 IBM MQ 中很容易遭到 xa_start 中的封鎖。在存取資源管理程式時，不同伺服器群組中的程序會使用個別的 XID，因此不需要在 IBM MQ 中序列化其交易工作。

啟用 CICS 兩階段確定程序

若要讓 CICS 能夠使用兩階段確定處理程序來協調包括 MQI 呼叫的交易，請將 CICS XAD 資源定義段落項目新增至 CICS 區域。請注意，此主題不適用於 z/OS。

以下是新增 IBM MQ for Windows 的 XAD 段落項目的範例，其中 *Drive* 是安裝 IBM MQ 的磁碟機 (例如，D:)。

```
cicsadd -cxad -rcics_region \
ResourceDescription="MQM XA Product Description" \
SwitchLoadFile="Drive:\Program Files\IBM\IBM MQ\bin\mqmc4swi.dll" \
XAOpen=queue_manager_name
```

若為延伸交易式用戶端，請使用切換載入檔 mqcc4swi.dll。

以下是針對 IBM MQ for AIX or Linux 系統新增 XAD 段落項目的範例，其中 *MQ_INSTALLATION_PATH* 代表 IBM MQ 安裝所在的高階目錄：

```
cicsadd -cxad -rcics_region \
ResourceDescription="MQM XA Product Description" \
SwitchLoadFile="MQ_INSTALLATION_PATH/lib/amqzsc" \
XAOpen=queue_manager_name
```

若為延伸交易式用戶端，請使用交換器載入檔案 amqczsc。

如需使用 **cicsadd** 指令的相關資訊，請參閱 CICS 文件。

對 IBM MQ 的呼叫可以包含在 CICS 交易中，且 CICS 會依照指示來確定或回復 IBM MQ 資源。用戶端應用程式無法使用此支援。

您必須從 CICS 交易發出 MQCONN，才能存取 IBM MQ 資源，然後在結束時發出對應的 MQDISC。

啟用 CICS 使用者結束程式

CICS 使用者結束程式點 (通常稱為 使用者結束程式) 是 CICS 模組中的一個位置，CICS 可以在其中將控制傳送至您所撰寫的程式 (使用者結束程式 程式)，且 CICS 可以在結束程式完成其工作時回復控制。

在使用 CICS 使用者結束程式之前，請先閱讀適用於您平台的 CICS 管理手冊。

範例作業終止結束程式

IBM MQ 提供 CICS 作業終止結束程式的範例原始碼。

範例原始碼位於下列目錄中:

表 9: CICS 作業終止結束程式		
平台	目錄	來源檔
AIX and Linux 系統	<code>MQ_INSTALLATION_PATH/samp</code>	<code>amqzscgx.c</code>
Windows	<code>MQ_INSTALLATION_PATH\Tools\c\範例</code>	<code>amqzscgn.c</code>

`MQ_INSTALLATION_PATH` 代表 IBM MQ 安裝所在的高階目錄。

範例作業終止結束程式的建置指示包含在每一個原始檔的註解中。

CICS 會在正常及異常作業終止時 (在取得任何同步點之後) 呼叫此結束程式。跳出程式中不允許任何可回復的工作。

這些函數僅在 CICS 版本支援 XA 介面的 IBM MQ 及 CICS 環境定義中使用。CICS 將這些程式庫稱為 `programs` 或 `user exits`。

CICS 有一些使用者結束程式，且 `amqzscgx` (如果使用的話) 在 CICS 上定義並啟用為 `Task termination user exit (UE014015)`，即結束號碼 15。

當 CICS 呼叫作業終止結束程式時，CICS 已通知 IBM MQ 作業的終止狀態，且 IBM MQ 已採取適當的動作 (確定或回復)。結束程式只會發出 `MQDISC` 來清除。

安裝及配置 CICS 系統以使用作業終止結束程式的目的之一，是為了保護您的系統免受應用程式碼錯誤的部分結果影響。例如，如果 CICS 交易在未先呼叫 `MQDISC` 的情況下異常結束，且未安裝任何作業終止結束程式，則您可能會看到 (大約 10 秒內) CICS 區域的後續無法復原的失敗。這是因為在 `cicsas` 處理程序中執行的 IBM MQ 性能執行緒將不會公佈及提供清除及傳回的時間。症狀可能是在將 `FFST` 報告寫入 `/var/mqm/errors` 或 Windows 上的對等位置之後，`cicsas` 處理程序立即結束。

使用 Microsoft Transaction Server (COM +)

COM + (Microsoft Transaction Server) 是設計用來協助使用者在一般中層伺服器中執行商業邏輯應用程式。

如需重要資訊，請參閱 [只能與 Windows 上的主要安裝搭配使用的特性](#)。

COM + 會將工作劃分為活動，這些活動通常是商業邏輯的簡短獨立區塊，例如從帳戶 A 轉移資金至帳戶 B。COM + 高度依賴物件導向，特別是 COM; 鬆散的 COM + 活動由 COM (商業) 物件表示。

COM + 是作業系統的整合部分。

COM + 提供三項服務給商業物件管理者，免除商業物件程式設計師的疑慮:

- 交易管理
- 安全
- 資源儲存區

您通常會將 COM + 與前端程式碼 (亦即 COM + 中所保留物件的 COM 用戶端) 及後端服務 (例如資料庫) 搭配使用，並在 COM + 商業物件與後端之間橋接 IBM MQ。

前端程式碼可以是獨立式程式，或 Microsoft 網際網路 Information Server (IIS) 所管理的「作用中伺服器頁面 (ASP)」。前端程式碼可以位於與 COM + 及其商業物件相同的電腦上，並透過 COM 進行連線。或者，前端程式碼可以在不同的電腦上，透過 DCOM 進行連線。在不同狀況下，您可以使用不同的用戶端來存取相同的 COM + 商業物件。

後端程式碼可以與 COM + 及其商業物件位於同一部電腦上，或位於透過任何 IBM MQ 支援的通訊協定進行連線的不同電腦上。

即將到期的廣域工作單元

佇列管理程式可以配置為在預先配置的間置間隔之後使廣域工作單元到期。

若要啟用此行為，請設定下列環境變數:

- `AMQ_TRANSACTION_EXPIRY_RESCAN`=重新掃描間隔 (毫秒)
- `AMQ_XA_TRANSACTION_EXPIRY`=逾時間隔 (毫秒)



小心: 環境變數只會影響 `OPEN` 群組中 `XA` 規格的表格 6-4 中處於 **Idle** 狀態的交易。

亦即，未在任何應用程式執行緒上相關聯，但外部「交易管理程式」軟體尚未針對其呼叫 `xa_prepare` 函數呼叫的交易。

外部交易管理程式只會保留已準備、已確定或已回復的交易日誌。如果外部交易管理程式因任何原因而關閉，則在其返回時，它會驅動備妥、確定及回復交易完成，但任何尚未準備好的作用中交易會變成孤立交易。若要避免此情況，請設定 `AMQ_XA_TRANSACTION_EXPIRY`，以容許在應用程式進行 MQI 交易式 API 呼叫與完成交易 (已在其他資源管理程式上執行交易式工作) 之間的預期間隔。

為了確保在 `AMQ_XA_TRANSACTION_EXPIRY` 到期之後及時清理，請將 `AMQ_TRANSACTION_EXPIRY_RESCAN` 值設為低於 `AMQ_XA_TRANSACTION_EXPIRY` 間隔的值，最好是在 `AMQ_XA_TRANSACTION_EXPIRY` 間隔內多次重新掃描。

回復處置單元

IBM MQ for z/OS 提供回復單元處置。此特性可讓您配置當連接至相同佇列共用群組 (QSG) 內的另一個佇列管理程式時，是否可以驅動兩階段確定交易的第二階段 (例如，在回復期間)。

IBM MQ for z/OS V7.0.1 以及更新版本支援回復單元處置。

回復處置單元

回復單元處置與應用程式的連線以及隨後啟動的任何交易相關。有兩個可能的回復單元處置。

- `GROUP` 回復單元處置識別交易式應用程式在邏輯上連接至佇列共用群組，且與任何特定佇列管理程式沒有親緣性。當連接至 QSG 內的任何佇列管理程式時，它所啟動且已完成確定處理程序 `phase-1` 的任何兩階段確定交易 (亦即，它們是不確定的) 都可以進行查詢及解決。在回復實務範例中，這表示交易協調程式不需要重新連接至可能無法使用的相同佇列管理程式。
- `QMGR` 回復處置單元可識別應用程式與它所連接的佇列管理程式具有直接親緣性，且它所啟動的任何交易也具有此處置。

在回復實務中，不論佇列管理程式是否屬於佇列共用群組，交易協調程式必須重新連接至相同的佇列管理程式，才能查詢及解決任何不確定的交易。

z/OS 如需如何實作此特性的詳細資料，請參閱 [佇列共用群組中的回復單元處置](#)。

安全實務範例

一組實務範例，示範將安全套用至不同配置。

下列子主題說明可用的安全實務範例：

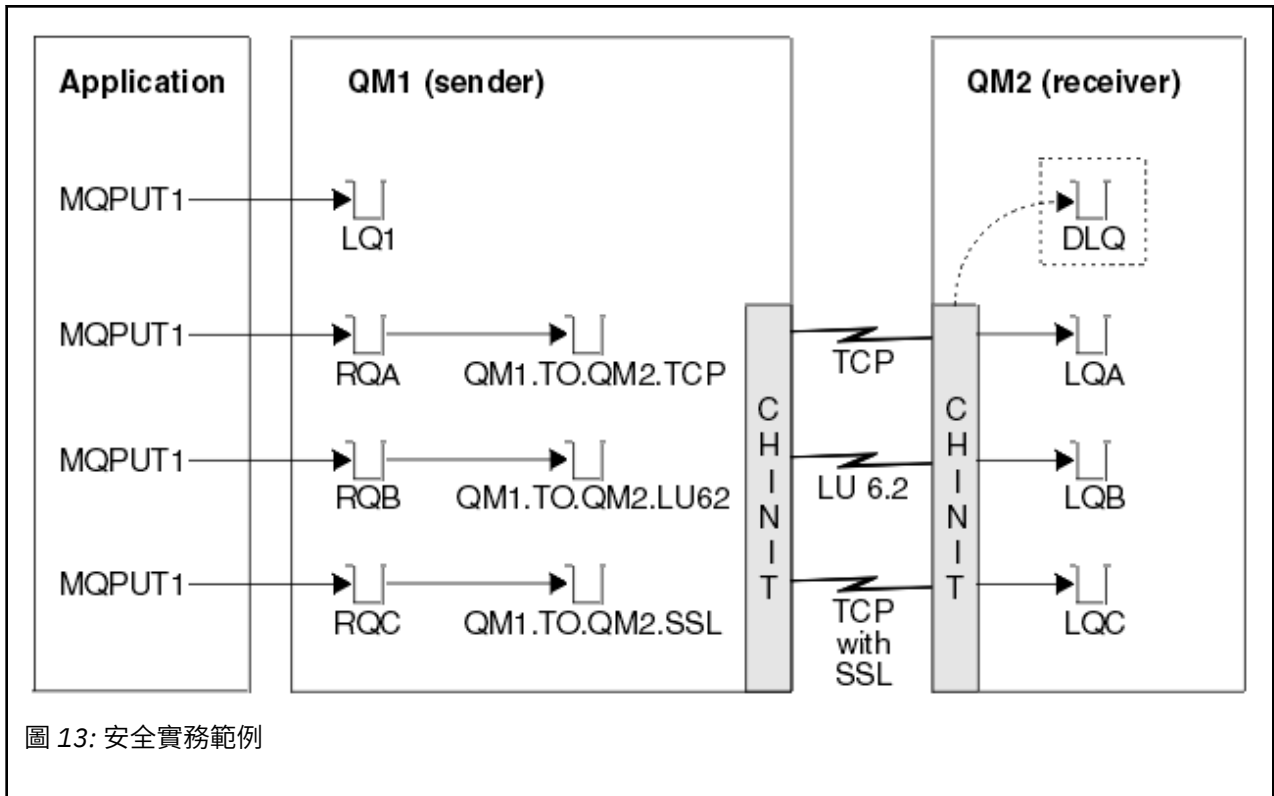
相關工作

z/OS [在 z/OS 上設定安全](#)

z/OS 安全實務範例: z/OS 上的兩個佇列管理程式

在此實務範例中，應用程式使用 `MQPUT1` 呼叫，將訊息放置到佇列管理程式 `QM1` 上的佇列。然後，部分訊息會使用 TCP 及 LU 6.2 通道轉遞至 `QM2` 上的佇列。TCP 通道是否可以與 SSL/TLS。應用程式可以是批次應用程式或 CICS 應用程式，並使用 `MQPMO_SET_ALL_CONTEXT` 選項放置訊息。

如第 77 頁的圖 13 中所說明。



下列是關於佇列管理程式的假設:

- 所有必要的 IBM MQ 定義都已預先定義，或已透過佇列管理程式啟動時所處理的 CSQINP2 資料集建立。如果沒有，您需要對定義這些物件所需的指令具有適當的存取權。
- 在佇列管理程式及通道起始程式啟動之前，已定義所有必要的 RACF 設定檔，並授與適當的存取權。如果沒有，您需要適當的權限來發出必要的 RACF 指令，以定義所有需要的設定檔，並授與這些設定檔適當的存取權。您也需要適當的權限來發出 MQSC 安全指令，以開始使用新的安全設定檔。
- 所有必要的數位憑證都已建立並連接至金鑰環。在 QM2 系統上，RACF 會辨識 QM1 作為 SSL/TLS 信號交換的一部分所傳送的數位憑證，因為它也安裝在該 RACF 設定檔中，或因為存在相符的「憑證名稱檔案 (CNF)」過濾器。

相關工作

z/OS 在 z/OS 上設定安全

z/OS 兩個佇列管理程式實務範例的安全切換設定

切換設定和 RACF 設定檔。

兩個佇列管理程式都設定下列安全切換參數:

- 上的子系統安全
- 上的佇列安全
- 上的替代使用者安全
- 上的環境定義安全
- 關閉處理程序安全
- 關閉名單安全
- 關閉主題安全
- 上的連線安全
- 上的指令安全

- 上的指令資源安全

下列設定檔定義在 MQADMIN 類別中，以關閉處理程序、名稱清單和主題安全：

```
QM1.NO.PROCESS.CHECKS
QM1.NO.NLIST.CHECKS
QM1.NO.TOPIC.CHECKS
QM2.NO.PROCESS.CHECKS
QM2.NO.NLIST.CHECKS
QM2.NO.TOPIC.CHECKS
```

兩個佇列管理程式實務範例中的佇列管理程式 QM1

QM1 的佇列及通道。

下列佇列定義於佇列管理程式 QM1：

LQ1

本端佇列。

RQA

遠端佇列定義，具有下列屬性：

- RNAME (LQA)
- RQMNAME (QM2)
- XMITQ (QM1.TO.QM2.TCP)

RQB

遠端佇列定義，具有下列屬性：

- RNAME (LQB)
- RQMNAME (QM2)
- XMITQ (QM1.TO.QM2.LU62)

RQC

遠端佇列定義，具有下列屬性：

- RNAME (LQC)
- RQMNAME (QM2)
- XMITQ (QM1.TO.QM2.TLS)

QM1.TO.QM2.TCP

傳輸佇列。

QM1.TO.QM2.LU62

傳輸佇列。

QM1.TO.QM2.TLS

傳輸佇列。

下列通道定義於 QM1：

QM1.TO.QM2.TCP

具有下列屬性的傳送端通道定義：

- CHLTYPE (SDR)
- TRPTYPE (TCP)
- XMITQ (QM1.TO.QM2.TCP)
- CONNAME (QM2TCP)

QM1.TO.QM2.LU62

具有下列屬性的傳送端通道定義：

- CHLTYPE (SDR)

- TRPTYPE (LU62)
- XMITQ (QM1.TO.QM2.LU62)
- CONNAME (QM2LU62)

(如需設定 APPC 安全的相關資訊，請參閱 [z/OS 上通道起始程式的安全考量](#)。)

QM1.TO.QM2.TLS

具有下列屬性的傳送端通道定義：

- CHLTYPE (SDR)
- TRPTYPE (TCP)
- XMITQ (QM1.TO.QM2.TLS)
- CONNAME (QM2TCP)
- SSLCIPH (TLS_RSA_WITH_AES_128_CBC_SHA256)

z/OS 兩個佇列管理程式實務範例中的佇列管理程式 QM2

QM2 的佇列及通道。

已在佇列管理程式 QM2: 上定義下列佇列：

LQA

本端佇列。

LQB

本端佇列。

LQC

本端佇列。

DLQ

用作無法傳送郵件之佇列的本端佇列。

下列通道已在 QM2:

QM1.TO.QM2.TCP

具有下列屬性的接收端通道定義：

- CHLTYPE (RCVR)
- TRPTYPE (TCP)
- PUTAUT (CTX)
- MCAUSER (MCATCP)

QM1.TO.QM2.LU62

具有下列屬性的接收端通道定義：

- CHLTYPE (RCVR)
- TRPTYPE (LU62)
- PUTAUT (CTX)
- MCAUSER (MCALU62)

(如需設定 APPC 安全的相關資訊，請參閱 [z/OS 上通道起始程式的安全考量](#)。)

QM1.TO.QM2.TLS

具有下列屬性的接收端通道定義：

- CHLTYPE (RCVR)
- TRPTYPE (TCP)
- PUTAUT (CTX)
- MCAUSER (MCASSL)
- SSLCIPH (TLS_RSA_WITH_AES_128_CBC_SHA256)

z/OS 在兩個佇列管理程式實務中使用的使用者 ID

實務範例中使用者 ID 的說明。

使用下列使用者 ID:

BATCHED

批次應用程式 (工作或 TSO ID)

MSGUSR

MQMD (環境定義使用者 ID) 中的 *UserIdentifier*

MOVER1

QM1 通道起始程式位址空間使用者 ID

MOVER2

QM2 通道起始程式位址空間使用者 ID

MCATCP

在 TCP/IP 上指定且沒有 SSL/TLS 接收端通道定義的 MCAUSER

MCALU62

LU 6.2 接收端通道定義上指定的 MCAUSER

MCASSL

在具有 SSL/TLS 接收端通道定義的 TCP/IP 上指定 MCAUSER

CICSAD1

CICS 位址空間 ID

CICSTX1

CICS 作業使用者 ID

CERTID

RACF 與已傳送憑證相關聯的使用者 ID。

z/OS 兩個佇列管理程式實務所需的安全設定檔及存取權

兩個佇列管理程式實務範例之批次或 CICS 實作所需的安全設定檔及存取權的相關資訊。

下表顯示讓兩個佇列管理程式實務範例運作所需的安全設定檔。還需要其他安全設定檔，視您是執行批次還是 CICS 實務範例實作而定。如需進一步資訊，請參閱第 81 頁的『批次應用程式所需的安全設定檔』和第 83 頁的『CICS 應用程式所需的安全設定檔』。

類別	設定檔	使用者 ID	存取
MQCONN	QM1.CHIN	MOVER1	READ
MQADMIN	QM1.RESLEVEL	BATCHED CICSAD1 MOVER1	無
MQADMIN	QM1.CONTEXT.**	MOVER1	CONTROL
MQQUEUE	QM1.SYSTEM.COMMAND.INPUT	MOVER1	UPDATE
MQQUEUE	QM1.SYSTEM.CHANNEL.SYNCQ	MOVER1	UPDATE
MQQUEUE	QM1.SYSTEM.CHANNEL.INITQ	MOVER1	UPDATE
MQQUEUE	QM1.SYSTEM.COMMAND.REPLY.MODEL	MOVER1	UPDATE
MQQUEUE	QM1.SYSTEM.ADMIN.CHANNEL.EVENT	MOVER1	UPDATE
MQQUEUE	QM1.QM1.TO.QM2.TCP	MOVER1	ALTER

表 10: 範例實務的安全設定檔。

此表格中的四個直欄顯示兩個佇列管理程式實務範例的類別、設定檔、使用者 ID 及存取權。

(繼續)

類別	設定檔	使用者 ID	存取
MQQUEUE	QM1.QM1.TO.QM2.LU62	MOVER1	ALTER
MQQUEUE	QM1.QM1.TO.QM2.TLS	MOVER1	ALTER
MQCONN	QM2.CHIN	MOVER2	READ
MQADMIN	QM2.RESLEVEL	MOVER2	無
MQADMIN	QM2.CONTEXT.**	MOVER2	CONTROL
MQQUEUE	QM2.SYSTEM.COMMAND.INPUT	MOVER2	UPDATE
MQQUEUE	QM2.SYSTEM.CHANNEL.SYNCQ	MOVER2	UPDATE
MQQUEUE	QM2.SYSTEM.CHANNEL.INITQ	MOVER2	UPDATE
MQQUEUE	QM2.SYSTEM.COMMAND.REPLY.MODEL	MOVER2	UPDATE
MQQUEUE	QM2.SYSTEM.ADMIN.CHANNEL.EVENT	MOVER2	UPDATE
MQQUEUE	QM2.DLQ	MOVER2	UPDATE

批次應用程式所需的安全設定檔

兩個佇列管理程式實務範例的批次實作所需的其他安全設定檔。

批次應用程式在 QM1 上以使用者 ID BATCHID 執行。它會連接至佇列管理程式 QM1，並將訊息放入下列佇列：

- LQ1
- RQA
- RQB
- RQC

它使用 MQPMO_SET_ALL_CONTEXT 選項。在訊息描述子 (MQMD) 的 *UserIdentifier* 欄位中找到的替代使用者 ID 是 MSGUSR。

佇列管理程式 QM1: 需要下列設定檔：

表 11: 佇列管理程式 QM1 上批次應用程式的安全設定檔範例

類別	設定檔	使用者 ID	存取
MQCONN	QM1.BATCH	BATCHED	READ
MQADMIN	QM1.CONTEXT.**	BATCHED	CONTROL
MQQUEUE	QM1.LQ1	BATCHED	UPDATE
MQQUEUE	QM1.RQA	BATCHED	UPDATE
MQQUEUE	QM1.RQB	BATCHED	UPDATE
MQQUEUE	QM1.RQC	BATCHED	UPDATE

對於放入佇列管理程式 QM1 (適用於不使用 TLS 的 TCP/IP 通道) 上佇列 RQA 的訊息，在佇列管理程式 QM2 上需要下列設定檔：

表 12: 使用 TCP/IP 而非 TLS 之佇列管理程式 QM2 的安全設定檔範例

類別	設定檔	使用者 ID	存取
MQADMIN	QM2.ALTERNATE.USER.MSGUSR	MCATCP MOVER2	UPDATE
MQADMIN	QM2.CONTEXT.**	MCATCP MOVER2	CONTROL
MQQUEUE	QM2.LQA	MOVER2 MSGUSR	UPDATE
MQQUEUE	QM2.DLQ	MOVER2 MSGUSR	UPDATE

附註:

1. 因為已使用 PUTAUT (CTX) 及 MCAUSER (MCATCP) 定義接收端通道，所以會使用傳入訊息 MQMD 的使用者 ID 作為佇列管理程式 QM2 上 MQPUT1 的使用者 ID。
2. 接收端通道定義的 MCAUSER 欄位設為 MCATCP; 除了針對替代使用者 ID 及環境定義設定檔執行的檢查之外，還會使用此使用者 ID。
3. 訊息描述子 (MQMD) 中的 MOVER2 使用者 ID 和 *UserIdentifier* 用於針對佇列進行資源檢查。
4. MOVER2 和 MSGUSR 使用者 ID 都需要存取無法傳送郵件的佇列，才能在該處傳送無法放入目的地佇列的訊息。
5. 因為 RESLEVEL 設為 NONE，所以所有三項執行的檢查都會檢查兩個使用者 ID。

對於放入佇列管理程式 QM1 (適用於 LU 6.2 通道) 上佇列 RQB 的訊息，在佇列管理程式 QM2 上需要下列設定檔:

表 13: 使用 LU 6.2 之佇列管理程式 QM2 的安全設定檔範例

類別	設定檔	使用者 ID	存取
MQADMIN	QM2.ALTERNATE.USER.MSGUSR	MCALU62 MOVER1	UPDATE
MQADMIN	QM2.CONTEXT.**	MCALU62 MOVER1	CONTROL
MQQUEUE	QM2.LQB	MOVER1 MSGUSR	UPDATE
MQQUEUE	QM2.DLQ	MOVER1 MSGUSR	UPDATE

附註:

1. 傳入訊息 MQMD 的使用者 ID 會作為佇列管理程式 QM2 上 MQPUT1 的使用者 ID，因為已使用 PUTAUT (CTX) 及 MCAUSER (MCALU62) 定義接收端通道。
2. MCA 使用者 ID 會設為接收端通道定義 (MCALU62) 的 MCAUSER 欄位值。
3. 因為 LU 6.2 支援通道通訊系統上的安全，所以會使用從網路收到的使用者 ID 作為通道使用者 ID (MOVER1)。
4. 因為 RESLEVEL 設為 NONE，所以所有三項執行的檢查都會檢查兩個使用者 ID。
5. MCALU62 和 MOVER1 用於針對替代使用者 ID 和環境定義設定檔執行的檢查，MSGUSR 和 MOVER1 用於針對佇列設定檔執行的檢查。
6. MOVER1 和 MSGUSR 使用者 ID 都需要存取無法傳送郵件的佇列，才能在該處傳送無法放入目的地佇列的訊息。

對於放入佇列管理程式 QM1 (適用於使用 TLS 的 TCP/IP 通道) 上佇列 RQC 的訊息，佇列管理程式 QM2 需要下列設定檔:

表 14: 使用 TCP/IP 及 TLS 之佇列管理程式 QM2 的安全設定檔範例

類別	設定檔	使用者 ID	存取
MQADMIN	QM2.ALTERNATE.USER.MSGUSR	MCASSL CERTID	UPDATE

表 14: 使用 TCP/IP 及 TLS 之佇列管理程式 QM2 的安全設定檔範例 (繼續)

類別	設定檔	使用者 ID	存取
MQADMIN	QM2.CONTEXT.**	MCASSL CERTID	CONTROL
MQQUEUE	QM2.LQC	CERTID MSGUSR	UPDATE
MQQUEUE	QM2.DLQ	CERTID MSGUSR	UPDATE

附註:

1. 傳入訊息 MQMD 的使用者 ID 會用作佇列管理程式 QM2 上 MQPUT1 的使用者 ID，因為接收端通道是以 PUTAUT (CTX) 及 MCAUSER (MCASSL) 定義。
2. MCA 使用者 ID 設為接收端通道定義 (MCASSL) 的 MCAUSER 欄位值。
3. 因為通道在 TLS 信號交換期間從 QM1 傳送的憑證可能安裝在 QM2 的系統上，或可能符合 QM2 的系統上的憑證名稱過濾器，所以在該相符期間找到的使用者 ID 會用作通道使用者 ID (CERTID)。
4. 因為 RESLEVEL 設為 NONE，所以所有三項執行的檢查都會檢查兩個使用者 ID。
5. MCASSL 和 CERTID 用於針對替代使用者 ID 和環境定義設定檔執行的檢查，MSGUSR 和 MOVER1 用於針對佇列設定檔執行的檢查。
6. CERTID 和 MSGUSR 使用者 ID 都需要存取無法傳送郵件的佇列，以便無法放入目的地佇列的訊息可以在該處傳送。

z/OS CICS 應用程式所需的安全設定檔

兩個佇列管理程式實務範例的 CICS 實作所需的其他安全設定檔。

CICS 應用程式使用 CICS 位址空間使用者 ID CICSAD1 及 CICS 作業使用者 ID CICSTX1。佇列管理程式 QM1 所需要的安全設定檔與批次應用程式所需要的設定檔不同。佇列管理程式 QM2 所需的設定檔與批次應用程式的設定檔相同。

佇列管理程式 QM1: 需要下列設定檔:

表 15: 佇列管理程式 QM1 上 CICS 應用程式的範例安全設定檔

類別	設定檔	使用者 ID	存取
MQCONN	QM1.CICS	CICSAD1	READ
MQADMIN	QM1.CONTEXT.**	CICSAD1 CICSTX1	CONTROL
MQQUEUE	QM1.LQ1	CICSAD1 CICSTX1	UPDATE
MQQUEUE	QM1.RQA	CICSAD1 CICSTX1	UPDATE
MQQUEUE	QM1.RQB	CICSAD1 CICSTX1	UPDATE

z/OS 安全實務範例: z/OS 上的佇列共用群組

在此實務範例中，應用程式使用 MQPUT1 呼叫，將訊息放置到佇列管理程式 QM1 上的佇列。然後，部分訊息會使用 TCP 及 LU 6.2 通道轉遞至 QM2 上的佇列。應用程式是批次應用程式，並使用 MQPMO_SET_ALL_CONTEXT 選項放置訊息。

如第 77 頁的圖 13 中所說明。

下列是關於佇列管理程式的假設:

- 所有必要的 IBM MQ 定義都已預先定義，或已透過佇列管理程式啟動時所處理的 CSQINP2 資料集建立。如果沒有，您需要對定義這些物件所需的指令具有適當的存取權。

- 在佇列管理程式及通道起始程式啟動之前，已定義所有必要的 RACF 設定檔，並授與適當的存取權。

如果沒有，您需要適當的權限來發出必要的 RACF 指令，以定義所有需要的設定檔，並授與這些設定檔適當的存取權。您也需要適當的權限來發出 MQSC 安全指令，以開始使用新的安全設定檔。

相關工作

z/OS 在 z/OS 上設定安全

z/OS 佇列共用群組實務範例的安全切換設定

切換設定和 RACF 設定檔。

下列是針對佇列共用群組所設定的安全切換參數：

- 上的子系統安全
- 上的佇列共用群組安全
- 關閉佇列管理程式安全
- 上的佇列安全
- 上的替代使用者安全
- 上的環境定義安全
- 關閉處理程序安全
- 關閉名單安全
- 關閉主題安全
- 上的連線安全
- 上的指令安全
- 上的指令資源安全

下列設定檔定義在 MQADMIN 類別中，以關閉處理程序、名單、主題及佇列管理程式層次安全：

```
QSGA.NO.PROCESS.CHECKS
QSGA.NO.NLIST.CHECKS
QSGA.NO.TOPIC.CHECKS
QSGA.NO.QMGR.CHECKS
```

z/OS 佇列共用群組實務中的佇列管理程式 QM1

QM1 的佇列及通道。

下列佇列定義於佇列管理程式 QM1：

LQ1

本端佇列。

RQA

遠端佇列定義，具有下列屬性：

- RNAME (LQA)
- RQMNAME (QM2)
- XMITQ (QM1.TO.QM2.TCP)

RQB

遠端佇列定義，具有下列屬性：

- RNAME (LQB)
- RQMNAME (QM2)
- XMITQ (QM1.TO.QM2.LU62)

QM1.TO.QM2.TCP

傳輸佇列。

QM1.TO.QM2.LU62

傳輸佇列。

下列通道定義於 QM1:

QM1.TO.QM2.TCP

具有下列屬性的傳送端通道定義:

- CHLTYPE (SDR)
- TRPTYPE (TCP)
- XMITQ (QM1.TO.QM2.TCP)
- CONNAME (QM2TCP)

QM1.TO.QM2.LU62

具有下列屬性的傳送端通道定義:

- CHLTYPE (SDR)
- TRPTYPE (LU62)
- XMITQ (QM1.TO.QM2.LU62)
- CONNAME (QM2LU62)

(如需設定 APPC 安全的相关資訊, 請參閱 [z/OS 上通道起始程式的安全考量](#)。)

z/OS 佇列共用群組實務中的佇列管理程式 QM2

QM2 的佇列及通道。

已在佇列管理程式 QM2: 上定義下列佇列:

LQA

本端佇列。

LQB

本端佇列。

DLQ

用作無法傳送郵件之佇列的本端佇列。

下列通道已在 QM2:

QM1.TO.QM2.TCP

具有下列屬性的接收端通道定義:

- CHLTYPE (RCVR)
- TRPTYPE (TCP)
- PUTAUT (CTX)
- MCAUSER (MCATCP)

QM1.TO.QM2.LU62

具有下列屬性的接收端通道定義:

- CHLTYPE (RCVR)
- TRPTYPE (LU62)
- PUTAUT (CTX)
- MCAUSER (MCALU62)

(如需設定 APPC 安全的相关資訊, 請參閱 [z/OS 上通道起始程式的安全考量](#)。)

z/OS 佇列共用群組實務中使用的使用者 ID

實務範例中使用者 ID 的說明。

使用下列使用者 ID:

BATCHED

批次應用程式 (工作或 TSO ID)

MSGUSR

MQMD (環境定義使用者 ID) 中的 *UserIdentifier*

MOVER1

QM1 通道起始程式位址空間使用者 ID

MOVER2

QM2 通道起始程式位址空間使用者 ID

MCATCP

TCP/IP 接收端通道定義上指定的 MCAUSER

MCALU62

LU 6.2 接收端通道定義上指定的 MCAUSER

z/OS 佇列共用群組實務範例所需的安全設定檔及存取權

佇列共用群組實務範例之批次或 CICS 實作的安全設定檔及存取權。

下表顯示讓佇列共用群組實務正常運作所需的安全設定檔。此實務範例的批次實作也需要 [第 87 頁的『批次應用程式所需的安全設定檔』](#) 中說明的其他安全設定檔。

表 16: 範例實務的安全設定檔。

此表格中的四個直欄顯示佇列共用群組實務範例的類別、設定檔、使用者 ID 及存取權。

類別	設定檔	使用者 ID	存取
MQCONN	QSGA.CHIN	MOVER1 MOVER2	READ
MQADMIN	QSGA.RESLEVEL	BATCHID MOVER1 MOVER2	無
MQADMIN	QSGA.CONTEXT.**	MOVER1 MOVER2	CONTROL
MQQUEUE	QSGA.SYSTEM.COMMAND.INPUT	MOVER1 MOVER2	UPDATE
MQQUEUE	QSGA.SYSTEM.CHANNEL.SYNCQ	MOVER1 MOVER	UPDATE
MQQUEUE	QSGA.SYSTEM.CHANNEL.INITQ	MOVER1 MOVER2	UPDATE
MQQUEUE	QSGA.SYSTEM.COMMAND.REPLY.MODEL	MOVER1 MOVER2	UPDATE
MQQUEUE	QSGA.SYSTEM.ADMIN.CHANNEL.EVENT	MOVER1 MOVER2	UPDATE
MQQUEUE	QSGA.SYSTEM.QSG.CHANNEL.SYNCQ	MOVER1 MOVER2	UPDATE
MQQUEUE	QSGA.SYSTEM.QSG.TRANSMIT.QUEUE	MOVER1 MOVER2	UPDATE

表 16: 範例實務的安全設定檔。

此表格中的四個直欄顯示佇列共用群組實務範例的類別、設定檔、使用者 ID 及存取權。

(繼續)

類別	設定檔	使用者 ID	存取
MQQUEUE	QSGA.QM1.TO.QM2.TCP	MOVER1	ALTER
MQQUEUE	QSGA.QM1.TO.QM2.LU62	MOVER1	ALTER
MQQUEUE	QSGA.DLQ	MOVER2	UPDATE

批次應用程式所需的安全設定檔

佇列共用群組實務範例的批次實作所需的其他安全設定檔。

批次應用程式在 QM1 上以使用者 ID BATCHID 執行。它會連接至佇列管理程式 QM1，並將訊息放入下列佇列：

- LQ1
- RQA
- RQB

它使用 MQPMO_SET_ALL_CONTEXT 選項。在訊息描述子 (MQMD) 的 *UserIdentifier* 欄位中找到的使用者 ID 是 MSGUSR。

佇列管理程式 QM1: 需要下列設定檔：

表 17: 佇列管理程式 QM1 上批次應用程式的安全設定檔範例

類別	設定檔	使用者 ID	存取
MQCONN	QSGA.BATCH	BATCHED	READ
MQADMIN	QSGA.CONTEXT.**	BATCHED	CONTROL
MQQUEUE	QSGA.LQ1	BATCHED	UPDATE
MQQUEUE	QSGA.RQA	BATCHED	UPDATE
MQQUEUE	QSGA.RQB	BATCHED	UPDATE

在佇列管理程式 QM2 上，將訊息放入佇列 RQA 的佇列管理程式 QM1 (適用於 TCP/IP 通道) 需要下列設定檔：

表 18: 使用 TCP/IP 之佇列管理程式 QM2 的安全設定檔範例

類別	設定檔	使用者 ID	存取
MQADMIN	QSGA.ALTERNATE.USER.MSGUSR	MCATCP MOVER2	UPDATE
MQADMIN	QSGA.CONTEXT.**	MCATCP MOVER2	CONTROL
MQQUEUE	QSGA.LQA	MOVER2 MSGUSR	UPDATE
MQQUEUE	QSGA.DLQ	MOVER2 MSGUSR	UPDATE

附註：

1. 因為已使用 PUTAUT (CTX) 及 MCAUSER (MCATCP) 定義接收端通道，所以會使用傳入訊息 MQMD 的使用者 ID 作為佇列管理程式 QM2 上 MQPUT1 的使用者 ID。

2. 接收端通道定義的 MCAUSER 欄位設為 MCATCP; 除了針對替代使用者 ID 及環境定義設定檔執行的檢查之外，還會使用此使用者 ID。
3. 訊息描述子 (MQMD) 中的 MOVER2 使用者 ID 和 *UserIdentifier* 用於針對佇列進行資源檢查。
4. MOVER2 和 MSGUSR 使用者 ID 都需要存取無法傳送郵件的佇列，才能在該處傳送無法放入目的地佇列的訊息。
5. 因為 RESLEVEL 設為 NONE，所以所有三項執行的檢查都會檢查兩個使用者 ID。

對於放入佇列管理程式 QM1 (適用於 LU 6.2 通道) 上佇列 RQB 的訊息，在佇列管理程式 QM2 上需要下列設定檔：

類別	設定檔	使用者 ID	存取
MQADMIN	QSGA.ALTERNATE.USER.MSGUSR	MCALU62 MOVER1	UPDATE
MQADMIN	QSGA.CONTEXT.**	MCALU62 MOVER1	CONTROL
MQQUEUE	QSGA.LQB	MOVER1 MSGUSR	UPDATE
MQQUEUE	QSGA.DLQ	MOVER1 MSGUSR	UPDATE

附註：

1. 傳入訊息 MQMD 的使用者 ID 會作為佇列管理程式 QM2 上 MQPUT1 的使用者 ID，因為已使用 PUTAUT (CTX) 及 MCAUSER (MCALU62) 定義接收端通道。
2. MCA 使用者 ID 會設為接收端通道定義 (MCALU62) 的 MCAUSER 欄位值。
3. 因為 LU 6.2 支援通道通訊系統上的安全，所以會使用從網路收到的使用者 ID 作為通道使用者 ID (MOVER1)。
4. 因為 RESLEVEL 設為 NONE，所以所有三項執行的檢查都會檢查兩個使用者 ID。
5. MCALU62 和 MOVER1 用於針對替代使用者 ID 和環境定義設定檔執行的檢查，MSGUSR 和 MOVER1 用於針對佇列設定檔執行的檢查。
6. MOVER1 和 MSGUSR 使用者 ID 都需要存取無法傳送郵件的佇列，才能在該處傳送無法放入目的地佇列的訊息。

z/OS 伺服器至伺服器訊息通道攔截範例配置

伺服器至伺服器訊息通道截取需要配置通道定義及 Advanced Message Security 原則，以確保入埠及出埠訊息可以受到正確保護或不受保護。配置視通道是入埠還是出埠而定。

入埠通道 (inbound channel)

下列範例顯示接收端類型之入埠通道的一般配置，並提供保護未受保護入埠訊息所需的 AMS 原則詳細資料：



圖 14: 入埠配置

範例顯示：

- 佇列管理程式 QMA
- 通道 TO.QMA
- 本端佇列 DESTQ

請使用下列程式碼:

```
DEFINE CHANNEL(TO.QMA) CHLTYPE(RCVR) SSLCAUTH(REQUIRED) SSLCIPH(ANY_TLS12) TRPTYPE(TCP)
SPLPROT(ASPOLICY)

DEFINE QLOCAL(DESTQ) DESCR('AMS PROTECTED QUEUE')

setmqsp1 -m QMA -p DESTQ -e AES256 -i CN=TEST,O=ORG,C=US
```

註: 前述文字中說明的原則只會加密訊息; 亦即, AMS 機密性。

如需在 z/OS 上使用 **setmqsp1** 的相關資訊, 請參閱 [setmqsp1](#) 及 [訊息安全原則 \(CSQOUTIL\)](#)。

出埠通道 (outbound channel)

下列範例顯示傳送端類型之出埠通道的一般配置。此範例提供 AMS 原則的詳細資料, 這些原則是用來保護放入遠端佇列的訊息, 以及解除保護並傳送從傳輸佇列取得的訊息:

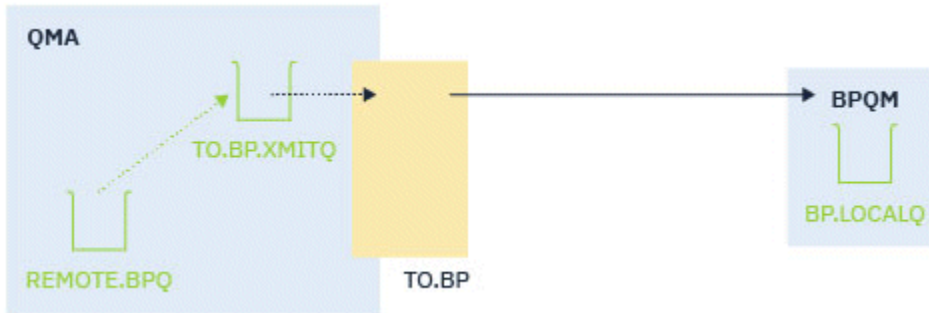


圖 15: 出埠配置

範例顯示:

- 佇列管理程式 QMA
- 通道 TO.BP
- 本端傳輸佇列 TO.BP.XMITQ
- 遠端佇列 REMOTE.BPQ

請使用下列程式碼:

```
DEFINE CHANNEL(TO.BP) CHLTYPE(SDR) SSLCAUTH(REQUIRED) SSLCIPH(ANY_TLS12) TRPTYPE(TCP)
SPLPROT(REMOVE) CONNAME('server(1414)') XMITQ(TO.BP.XMITQ)

DEFINE QLOCAL(TO.BP.XMITQ) DESCR('TRANSMISSION QUEUE FOR TO.BP') USAGE(XMITQ)

DEFINE QREMOTE(REMOTE.BPQ) DESCR('REMOTE QUEUE TO BP') RNAME(BP.LOCALQ) RQMANME(BPQM)
XMITQ(TO.BP.XMITQ)

setmqsp1 -m QMA -p TO.BP.XMITQ -e AES256 -i CN=TEST,O=ORG,C=US

setmqsp1 -m QMA -p REMOTE.BPQ -e AES256 -i CN=TEST,O=ORG,C=US
```

註: 前述文字中說明的原則只會加密訊息; 亦即, AMS 機密性。

使用 SSL/TLS 連接兩個佇列管理程式

使用 TLS 加密安全通訊協定的安全通訊包括設定通訊通道, 以及管理您將用於鑑別的數位憑證。

若要設定 SSL/TLS 安裝, 您必須定義通道以使用 TLS。您也必須取得並管理數位憑證。在測試系統上, 您可以使用自簽憑證或本端憑證管理中心 (CA) 發出的憑證。在正式作業系統上, 請勿使用自簽憑證。

如需建立及管理憑證的完整資訊，請參閱下列主題：

- ▶ **IBM i** 在 IBM i 上使用 SSL 或 TLS
- ▶ **ALW** 在 AIX, Linux, and Windows 系統上使用 SSL 或 TLS
- ▶ **z/OS** 在 z/OS 上使用 SSL 或 TLS

此主題集合介紹設定 SSL/TLS 通訊所涉及的作業，並提供完成這些作業的逐步指引。

您也可能想要測試 SSL/TLS 用戶端鑑別，這是通訊協定的選用部分。在 SSL/TLS 信號交換期間，SSL/TLS 用戶端一律會從伺服器取得並驗證數位憑證。使用 IBM MQ 實作，SSL/TLS 伺服器一律會向用戶端要求憑證。

附註：

1. 在此環境定義中，SSL/TLS 用戶端是指起始信號交換的連線。
2. ▶ **z/OS** 當 z/OS 佇列管理程式擔任 SSL/TLS 用戶端的角色時，佇列管理程式只會傳送憑證。

只有在找到具有相符標籤的憑證時，SSL/TLS 用戶端才會傳送憑證。如需詳細資料，請參閱 [數位憑證標籤](#)。

如果傳送用戶端憑證，SSL/TLS 伺服器一律會驗證用戶端憑證。如果用戶端未傳送憑證，則僅當作為 SSL/TLS 伺服器的通道結尾定義有 **SSLCAUTH** 參數設為 **REQUIRED** 或 **SSLPEER** 參數已設定值時，鑑別才會失敗。如需匿名連接佇列管理程式的相關資訊，亦即當 SSL/TLS 用戶端未傳送憑證時，請參閱 [第 94 頁的『使用單向鑑別來連接兩個佇列管理程式』](#)。

使用自簽憑證進行兩個佇列管理程式的交互鑑別

遵循下列範例指示，使用自簽 TLS 憑證來實作兩個佇列管理程式之間的交互鑑別。

關於這項作業

測試情境：

- 您有兩個佇列管理程式 QM1 及 QM2，它們需要安全地通訊。您需要在 QM1 與 QM2 之間執行交互鑑別。
- 您已決定使用自簽憑證來測試安全通訊。

產生的配置如下所示：

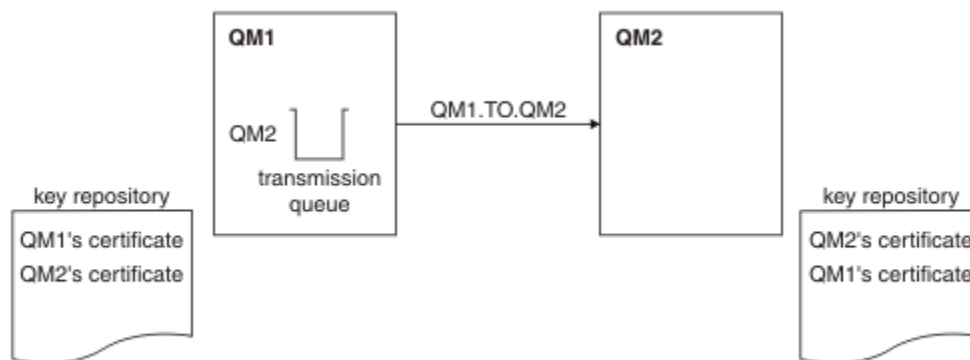


圖 16: 此作業產生的配置

在第 90 頁的圖 16 中，QM1 的金鑰儲存庫包含 QM1 的憑證，以及來自 QM2 的公用憑證。QM2 的金鑰儲存庫包含 QM2 的憑證，以及來自 QM1 的公用憑證。

程序

1. 根據作業系統，在每一個佇列管理程式上準備金鑰儲存庫：

- **ALW** 在 AIX, Linux, and Windows 系統上。
 - **z/OS** 在 z/OS 系統上。
2. 為每一個佇列管理程式建立自簽憑證:
- **ALW** 在 AIX, Linux, and Windows 系統上。
 - **z/OS** 在 z/OS 系統上。
3. 擷取每一個憑證的副本:
- **ALW** 在 AIX, Linux, and Windows 系統上。
 - **z/OS** 在 z/OS 系統上。
4. 使用 FTP **z/OS** 之類的公用程式，將 QM1 憑證的公用部分傳送至 QM2 系統，反之亦然，如 交換自簽憑證 中所述。
5. 將夥伴憑證新增至每一個佇列管理程式的金鑰儲存庫:
- **ALW** 在 AIX, Linux, and Windows 系統上。
 - **z/OS** 在 z/OS 系統上。
6. 在 QM1 上，發出類似下列範例的指令來定義傳送端通道及相關聯的傳輸佇列:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME(QM1.MACH.COM) XMITQ(QM2)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA) DESCR('Sender channel using TLS from QM1 to QM2')

DEFINE QLOCAL(QM2) USAGE(XMITQ)
```

此範例使用 CipherSpec TLS_RSA。通道每一端的 CipherSpecs 必須相同。

7. 在 QM2 上，發出類似下列範例的指令來定義接收端通道:

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using TLS from QM1 to QM2')
```

通道必須與您在步驟 第 91 頁的『6』 中定義的傳送端通道同名，並使用相同的 CipherSpec。

8. 啟動通道。

z/OS 在 z/OS 上，請參閱 啟動傳送端通道。

結果

金鑰儲存庫和通道如 第 90 頁的圖 16 中所示建立。

下一步

使用 DISPLAY 指令來檢查作業是否已順利完成。如果作業成功，則產生的輸出類似於下列範例中所示的輸出。

從佇列管理程式 QM1，輸入下列指令:

```
DISPLAY CHS(QM1.TO.QM2) SSLPEER SSLCERTI
```

產生的輸出類似下列範例:

```
DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
4 : DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
CHANNEL(QM1.TO.QM2)          CHLTYPE(SDR)
CONNAME(9.20.25.40)          CURRENT
RQMNAME(QM2)
```

```

SSLCERTI("CN=QM2,OU=<Department>,O=<Organization>,ST=<State>,C=<Country>")

SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5E:02,CN=QM2,OU=<Department>,O=<Organization>,ST=<State>,C=
<Country>")
STATUS(RUNNING)                                SUBSTATE(MQGET)
XMITQ(QM2)

```

從佇列管理程式 QM2 中，輸入下列指令：

```
DISPLAY CHS(QM1.TO.QM2) SSLPEER SSLCERTI
```

產生的輸出類似下列範例：

```

DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
 5 : DISPLAY CHSTATUS(QM1.TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
CHANNEL(QM2.TO.QM1)                                CHLTYPE(RCVR)
CONNAME(9.20.35.92)                                CURRENT
RQMNAME(QM1)
SSLCERTI("CN=QM1,OU=<Department>,O=<Organization>,ST=<State>,C=<Country>")

SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QM1,OU=<Department>,O=<Organization>,ST=<State>,C=
<Country>")
STATUS(RUNNING)                                SUBSTATE(RECEIVE)
XMITQ( )

```

在每一個情況下，SSLPEER 的值必須符合在步驟 [第 91 頁的『2』](#) 中建立之夥伴憑證中的 DN 值。發證者名稱符合同層級名稱，因為憑證是自簽憑證。

SSLPEER 是選用的。如果已指定，則必須設定其值，以便容許夥伴憑證 (在步驟 [第 91 頁的『2』](#) 中建立) 中的 DN。如需使用 SSLPEER 的相關資訊，請參閱 [SSLPEER 值的 IBM MQ 規則](#)。

使用 CA 簽章憑證來進行兩個佇列管理程式的交互鑑別

遵循下列範例指示，使用 CA 簽署的 TLS 憑證來實作兩個佇列管理程式之間的交互鑑別。

關於這項作業

測試情境：

- 您有兩個佇列管理程式稱為 QM1 及 QM2，它們需要安全通訊。您需要在 QM1 與 QM2 之間執行交互鑑別。
- 未來您計劃在正式作業環境中使用此網路，因此您已決定從頭開始使用 CA 簽章憑證。

產生的配置如下所示：

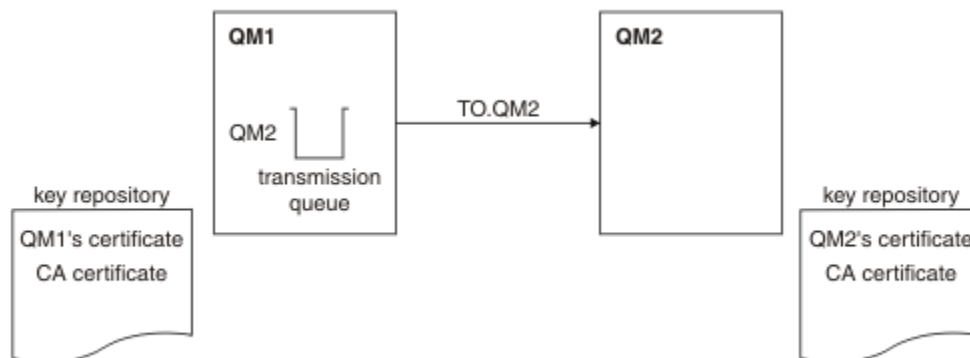


圖 17: 此作業產生的配置

在 [第 92 頁的圖 17](#) 中，QM1 的金鑰儲存庫包含 QM1 的憑證及 CA 憑證。QM2 的金鑰儲存庫包含 QM2 的憑證及 CA 憑證。

在此範例中，QM1 的憑證和 QM2 的憑證都由相同的 CA 發出。如果 QM1 的憑證及 QM2 的憑證是由不同的 CA 發出，則 QM1 及 QM2 的金鑰儲存庫必須同時包含這兩個 CA 憑證。

程序

1. 根據您企業使用的作業系統或系統，在每一個佇列管理程式上準備金鑰儲存庫：

- ▶ **IBM i** 在 [IBM i 系統上](#)。
- ▶ **ALW** 在 [AIX, Linux, and Windows 系統上](#)。
- ▶ **z/OS** 在 [z/OS 系統上](#)。

2. 要求每一個佇列管理程式的 CA 簽章憑證。
您可以對兩個佇列管理程式使用不同的 CA。

- ▶ **IBM i** 在 [IBM i 系統上](#)。
- ▶ **ALW** 在 [AIX, Linux, and Windows 系統上](#)。
- ▶ **z/OS** 在 [z/OS 系統上](#)。

3. 將憑證管理中心憑證新增至每一個佇列管理程式的金鑰儲存庫：

如果佇列管理程式使用不同的憑證管理中心，則必須將每一個憑證管理中心的 CA 憑證新增至兩個金鑰儲存庫。

- ▶ **IBM i** 請勿在 IBM i 系統上執行此步驟。
- ▶ **ALW** 在 [AIX, Linux, and Windows 系統上](#)。
- ▶ **z/OS** 在 [z/OS 系統上](#)。

4. 將 CA 簽章憑證接收至每一個佇列管理程式的金鑰儲存庫：

- ▶ **IBM i** 在 [IBM i 系統上](#)。
- ▶ **ALW** 在 [AIX, Linux, and Windows 系統上](#)。
- ▶ **z/OS** 在 [z/OS 系統上](#)。

5. 在 QM1 上，發出類似下列範例的指令來定義傳送端通道及相關聯的傳輸佇列：

```
DEFINE CHANNEL(TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP)
CONNNAME(QM2.MACH.COM) XMITQ(QM2) SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
DESCR('Sender channel using TLS from QM1 to QM2')

DEFINE QLOCAL(QM2) USAGE(XMITQ)
```

此範例使用 CipherSpec TLS_RSA_WITH_AES_128_CBC_SHA256。通道每一端的 CipherSpecs 必須相同。

6. 在 QM2 上，發出類似下列範例的指令來定義接收端通道：

```
DEFINE CHANNEL(TO.QM2) CHLTYPE(RCVR) TRPTYPE(TCP)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256) SSLCAUTH(REQUIRED)
DESCR('Receiver channel using TLS to QM2')
```

通道必須與您在步驟 [第 93 頁的『5』](#) 中定義的傳送端通道同名，並使用相同的 CipherSpec。

7. 啟動通道：

- ▶ **IBM i** 在 [IBM i 系統上](#)。
- ▶ **ALW** 在 [AIX, Linux, and Windows 系統上](#)。
- ▶ **z/OS** 在 [z/OS 系統上](#)。

結果

金鑰儲存庫和通道如 [第 92 頁的圖 17](#) 中所示建立。

下一步

使用 DISPLAY 指令來檢查作業是否已順利完成。如果作業成功，產生的輸出會如下列範例所示。

從佇列管理程式 QM1，輸入下列指令：

```
DISPLAY CHS(TO.QM2) SSLPEER SSLCERTI
```

產生的輸出類似下列範例：

```
DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
  4 : DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
CHANNEL(TO.QM2)                CHLTYPE(SDR)
CONNNAME(192.0.0.2)            CURRENT
RQMNAME(QM2)
SSLCERTI("CN=<Division> CA,OU=<Department>,O=<Organization>,ST=<State>,C=<Country>")

SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QM2,OU=<Department>,O=<Organization>,ST=<State>,C=
<Country>")
STATUS(RUNNING)                SUBSTATE(MQGET)
XMITQ(QM2)
```

從佇列管理程式 QM2 中，輸入下列指令：

```
DISPLAY CHS(TO.QM2) SSLPEER SSLCERTI
```

產生的輸出類似下列範例：

```
DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
  5 : DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
CHANNEL(TO.QM2)                CHLTYPE(RCVR)
CONNNAME(192.0.0.1)            CURRENT
RQMNAME(QM1)
SSLCERTI("CN=<Division> CA,OU=<Department>,O=<Organization>,ST=<State>,C=<Country>")

SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QM1,OU=<Department>,O=<Organization>,ST=<State>,C=
<Country>")
STATUS(RUNNING)                SUBSTATE(RECEIVE)
XMITQ( )
```

在每一個情況下，SSLPEER 的值必須符合在步驟 [第 93 頁的『2』](#) 中建立之夥伴憑證中的「識別名稱 (DN)」值。發證者名稱符合簽署步驟 [第 93 頁的『4』](#) 中所新增個人憑證之 CA 憑證的主體 DN。

使用單向鑑別來連接兩個佇列管理程式

請遵循這些範例指示來修改具有交互鑑別的系統，以容許佇列管理程式使用單向鑑別來連接至另一個系統；亦即，當 SSL/TLS 用戶端未傳送憑證時。

關於這項作業

測試情境：

- 您的兩個佇列管理程式 (QM1 及 QM2) 已設定為 [第 92 頁的『使用 CA 簽章憑證來進行兩個佇列管理程式的交互鑑別』](#)。
- 您想要變更 QM1，讓它使用單向鑑別連接至 QM2。

產生的配置如下所示：

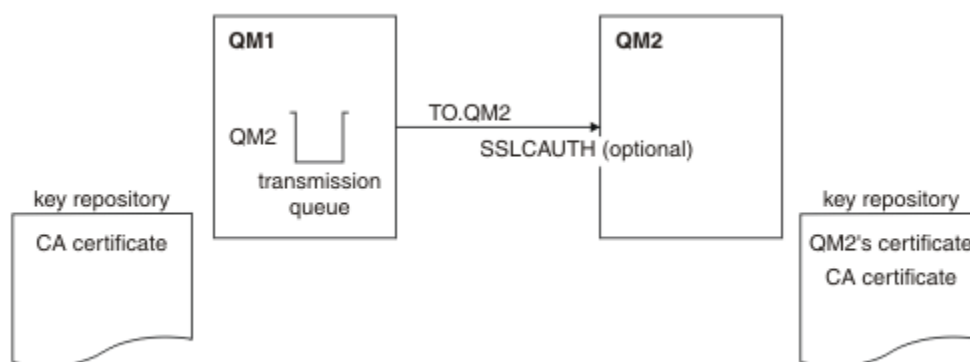


圖 18: 容許單向鑑別的佇列管理程式

程序

1. 從其金鑰儲存庫中移除 QM1 的個人憑證:

- **IBM i** 在 IBM i 系統上移除憑證。
- **ALW** 移除 AIX, Linux, and Windows 上的憑證。
- **z/OS** 移除 z/OS 系統上的憑證。執行此步驟兩次，以移除 QMA 的個人憑證及預設憑證。

如需如何標示憑證的詳細資料，請參閱 [數位憑證標籤](#)。

2. 選擇性的: 在 QM1 上，如果先前已執行任何 SSL/TLS 通道，請重新整理 SSL/TLS 環境，如 [重新整理 TLS 環境](#) 中所述。
3. 容許接收端上的匿名連線，如 [容許接收端通道上的匿名連線](#) 中所述。

金鑰儲存庫和通道已變更，如 [第 95 頁的圖 18](#) 中所示

4. 如果傳送端通道不在執行中，請啟動它。

註: 如果傳送端通道正在執行中，且您發出 REFRESH SECURITY TYPE (SSL) 指令 (在步驟 2 中)，則通道會自動重新啟動。

在通道的伺服器端，通道狀態顯示畫面上存在對等節點名稱參數值表示已傳送用戶端憑證。

5. 請發出一些 DISPLAY 指令來驗證作業已順利完成。

如果作業成功，則產生的輸出類似於下列範例中所示的輸出:

- 從 QM1 佇列管理程式中，輸入下列指令:

```
DISPLAY CHS(TO.QM2) SSLPEER SSLCERTI
```

產生的輸出將類似於下列範例:

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
  4 : DISPLAY CHSTATUS(TO.QMB) SSLPEER
AMQ8417: Display Channel Status details.
CHANNEL(TO.QM2)                CHLTYPE(SDR)
CONNAME(192.0.0.1)              CURRENT
QMNAME(QM2)
SSLCERTI("CN=IBM MQ CA,OU=IBM MQ Devt,O=IBM,ST=Hampshire,C=UK")
SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMB,OU=IBM MQ
Development,O=IBM,ST=Hampshire,C=UK")
STATUS(RUNNING)                 SUBSTATE(MQGET)
XMITQ(QM2)
```

- 從 QM2 佇列管理程式中，輸入下列指令:

```
DISPLAY CHS(TO.QM2) SSLPEER SSLCERTI
```

產生的輸出將類似於下列範例:

```
DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
  5 : DISPLAY CHSTATUS(TO.QM2) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
CHANNEL(TO.QM2)                CHLTYPE(RCVR)
CONNNAME(192.0.0.2)            CURRENT
RQMNAME(QMA)                  SSLCERTI( )
SSLPEER( )                     STATUS(RUNNING)
SUBSTATE(RECEIVE)             XMITQ( )
```

在 QM2 上，SSLPEER 欄位是空的，表示 QM1 未傳送憑證。在 QM1 上，SSLPEER 的值符合 QM2 個人憑證中 DN 的值。

將用戶端安全連接至佇列管理程式

使用 TLS 加密安全通訊協定的安全通訊包括設定通訊通道，以及管理您將用於鑑別的數位憑證。

若要設定 SSL/TLS 安裝，您必須定義通道以使用 TLS。您也必須取得並管理數位憑證。在測試系統上，您可以使用自簽憑證或本端憑證管理中心 (CA) 發出的憑證。在正式作業系統上，請勿使用自簽憑證。

如需建立及管理憑證的完整資訊，請參閱下列主題：

- ▶ **IBM i** [在 IBM i 上使用 SSL 或 TLS](#)
- ▶ **ALW** [在 AIX, Linux, and Windows 系統上使用 SSL 或 TLS](#)
- ▶ **z/OS** [在 z/OS 上使用 SSL 或 TLS](#)

此主題集合介紹設定 SSL/TLS 通訊所涉及的作業，並提供完成這些作業的逐步指引。

您可能想要測試 SSL/TLS 用戶端鑑別，這是通訊協定的選用部分。在 SSL/TLS 信號交換期間，SSL/TLS 用戶端一律會從伺服器取得並驗證數位憑證。使用 IBM MQ 實作，SSL/TLS 伺服器一律會向用戶端要求憑證。

▶ **IBM i** ▶ **ALW** 在 IBM i, AIX, Linux, and Windows 系統上，只有在 SSL/TLS 用戶端具有正確 IBM MQ 格式的標籤 (即 `ibmwebspheremq` 後接小寫的登入使用者 ID，或 **CERTLABL** 屬性的值) 時，才會傳送憑證。請參閱 [數位憑證標籤](#)。

如果傳送用戶端憑證，SSL/TLS 伺服器一律會驗證用戶端憑證。如果用戶端未傳送憑證，則只有在作為 SSL/TLS 伺服器的通道結尾定義為 **SSLCAUTH** 參數設為 **REQUIRED** 或已設定 **SSLPEER** 參數值時，鑑別才會失敗。如需匿名連接佇列管理程式的相關資訊，請參閱 [第 100 頁的『以匿名方式將用戶端連接至佇列管理程式』](#)。

相關概念

[IBM MQ classes for Java 中的 TLS CipherSpecs 及 CipherSuites](#)

[IBM MQ classes for JMS 中的 TLS CipherSpecs 及 CipherSuites](#)

相關工作

[使用受管理 .NET 用戶端的憑證](#)

使用自簽憑證進行用戶端及佇列管理程式的交互鑑別

遵循下列範例指示，使用自簽 TLS 憑證來實作用戶端與佇列管理程式之間的交互鑑別。

關於這項作業

▶ **IBM i** DCM on IBM i 不支援自簽憑證，因此此作業不適用於 IBM i 系統。

測試情境：

- 您具有需要安全通訊的用戶端 C1 及佇列管理程式 QM1。您需要在 C1 與 QM1 之間執行交互鑑別。

- 您已決定使用自簽憑證來測試安全通訊。

產生的配置如下所示：

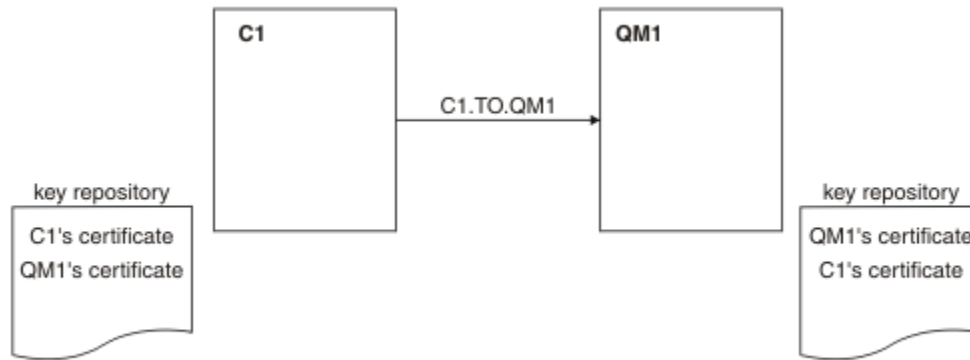


圖 19: 此作業產生的配置

在第 97 頁的圖 19 中，QM1 的金鑰儲存庫包含 QM1 的憑證及來自 C1 的公用憑證。C1 的金鑰儲存庫包含 C1 的憑證，以及來自 QM1 的公用憑證。

程序

1. 根據作業系統，在用戶端及佇列管理程式上準備金鑰儲存庫：
 - **ALW** 在 [AIX, Linux, and Windows 系統上](#)。
 - **z/OS** 在 [z/OS 系統上](#) (僅限佇列管理程式)。
2. 為用戶端及佇列管理程式建立自簽憑證：
 - **ALW** 在 [AIX, Linux, and Windows 系統上](#)。
 - **z/OS** 在 [z/OS 系統上](#) (僅限佇列管理程式)。
3. 擷取每一個憑證的副本：
 - **ALW** 在 [AIX, Linux, and Windows 系統上](#)。
 - **z/OS** 在 [z/OS 系統上](#)。
4. 使用 FTP 之類的公用程式，將 C1 憑證的公用部分傳送至 QM1 系統，反之亦然。
 - **z/OS** 若為 z/OS，請參閱 [交換自簽憑證](#)。
5. 將夥伴憑證新增至用戶端及佇列管理程式的金鑰儲存庫：
 - **ALW** 在 [AIX, Linux, and Windows 系統上](#)。
 - **z/OS** 在 [z/OS 系統上](#)。
6. 在佇列管理程式上發出指令 REFRESH SECURITY TYPE(SSL)。
7. 以下列其中一種方式來定義用戶端連線通道：
 - 在 C1 上搭配使用 MQCONNX 呼叫與 MQSCO 結構，如 [使用 MQCNO 在 IBM MQ MQI client 上建立用戶端連線通道](#) 中所述。
 - 使用用戶端通道定義表，如 [在伺服器上建立伺服器連線及用戶端連線定義](#) 中所述。
8. 在 QM1 上，發出類似下列範例的指令來定義伺服器連線通道：

```
DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA)
SSLCAUTH(REQUIRED) DESCR('Receiver channel using TLS from C1 to QM1')
```

通道必須與您在步驟 6 中定義的用戶端連線通道同名，並使用相同的 CipherSpec。

結果

金鑰儲存庫和通道如 [第 97 頁的圖 19](#) 中所示建立。

下一步

使用 **DISPLAY** 指令，檢查作業是否已順利完成。如果作業成功，則產生的輸出類似於下列範例中所示的輸出。

從佇列管理程式 QM1，輸入下列指令：

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
```

產生的輸出類似下列範例：

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
CHANNEL(C1.TO.QM1)                                CHLTYPE(SVRCONN)
CONNNAME(192.0.0.1)                                CURRENT
SSLCERTI("CN=QM1,OU=IBM MQ Development,O=IBM,ST=Hampshire,C=UK")
SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5E:02,CN=QM2,OU=IBM MQ
Development,O=IBM,ST=Hampshire,C=UK")
STATUS(RUNNING)                                SUBSTATE(RECEIVE)
```

您可以選擇設定通道定義的 **SSLPEER** 過濾器屬性。如果設定通道定義 **SSLPEER**，則其值必須符合在步驟 2 中建立的夥伴憑證中的主體 DN。成功連線之後，**DISPLAY CHSTATUS** 輸出中的 **SSLPEER** 欄位會顯示遠端用戶端憑證的主體 DN。

使用 CA 簽章憑證進行用戶端及佇列管理程式的交互鑑別

請遵循下列範例指示，使用 CA 簽署的 TLS 憑證來實作用戶端與佇列管理程式之間的交互鑑別。

關於這項作業

測試情境：

- 您具有需要安全通訊的用戶端 C1 及佇列管理程式 QM1。您需要在 C1 與 QM1 之間執行交互鑑別。
- 未來您計劃在正式作業環境中使用此網路，因此您已決定從頭開始使用 CA 簽章憑證。

產生的配置如下所示：

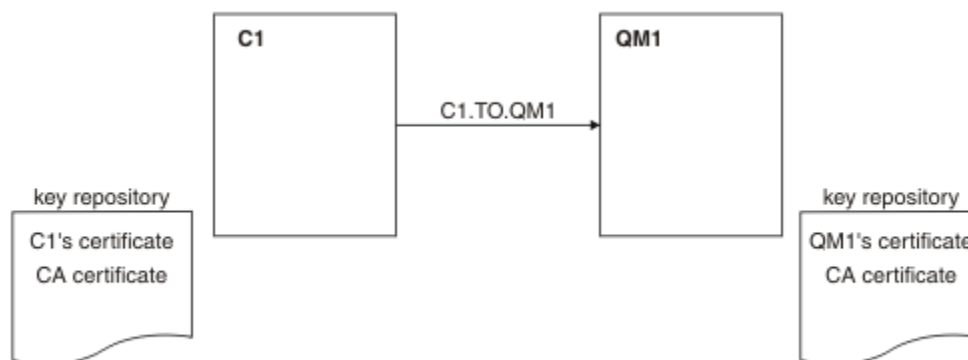





圖 20: 此作業產生的配置

在第 98 頁的圖 20 中，C1 的金鑰儲存庫包含 C1 的憑證及 CA 憑證。QM1 的金鑰儲存庫包含 QM1 的憑證及 CA 憑證。在此範例中，C1 的憑證和 QM1 的憑證都由相同的 CA 發出。如果 C1 的憑證和 QM1 的憑證是由不同的 CA 發出，則 C1 和 QM1 的金鑰儲存庫必須同時包含這兩個 CA 憑證。




程序

1. 根據作業系統，在用戶端及佇列管理程式上準備金鑰儲存庫：

-  在 [IBM i 系統上](#)。
-  在 [AIX, Linux, and Windows 系統上](#)。
-  在 [z/OS 系統上](#) (僅限佇列管理程式)。




2. 要求用戶端及佇列管理程式的 CA 簽章憑證。

您可以對用戶端及佇列管理程式使用不同的 CA。




-  在 [IBM i 系統上](#)。
-  在 [AIX, Linux, and Windows 系統上](#)。
-  在 [z/OS 系統上](#) (僅限佇列管理程式)。

3. 將憑證管理中心憑證新增至用戶端及佇列管理程式的金鑰儲存庫。

如果用戶端及佇列管理程式使用不同的「憑證管理中心」，則必須將每一個「憑證管理中心」的 CA 憑證新增至這兩個金鑰儲存庫。

-  請勿在 [IBM i 系統上](#) 執行此步驟。
-  在 [AIX, Linux, and Windows 系統上](#)。
-  在 [z/OS 系統上](#) (僅限佇列管理程式)。

4. 將 CA 簽章憑證接收至用戶端及佇列管理程式的金鑰儲存庫：

-  在 [IBM i 系統上](#)。
-  在 [AIX, Linux, and Windows 系統上](#)。
-  在 [z/OS 系統上](#) (僅限佇列管理程式)。

5. 以下列其中一種方式來定義用戶端連線通道：

- 在 C1 上搭配使用 MQCONNX 呼叫與 MQSCO 結構，如 [使用 MQCNO 在 IBM MQ MQI client 上建立用戶端連線通道](#) 中所述。
- 使用用戶端通道定義表，如 [在伺服器上建立伺服器連線及用戶端連線定義](#) 中所述。

6. 在 QM1 上，發出類似下列範例的指令來定義伺服器連線通道：

```
DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) TRPTYPE(TCP)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA) SSLCAUTH(REQUIRED)
DESCR('Receiver channel using TLS from C1 to QM1')
```

通道必須與您在步驟 6 中定義的用戶端連線通道同名，並使用相同的 CipherSpec。

結果

金鑰儲存庫和通道如 [第 98 頁的圖 20](#) 中所示建立。

下一步

使用 DISPLAY 指令來檢查作業是否已順利完成。如果作業成功，產生的輸出會如下列範例所示。

從佇列管理程式 QM1 中，輸入下列指令：

```
DISPLAY CHSTATUS(TO.QMB) SSLPEER SSLCERTI
```

產生的輸出類似下列範例:

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
CHANNEL(C1.TO.QM1)                CHLTYPE(SVRCONN)
CONNAME(192.0.0.1)                 CURRENT
SSLCERTI("CN=IBM MQ CA,OU=IBM MQ Devt,O=IBM,ST=Hampshire,C=UK")
SSLPEER("SERIALNUMBER=4C:D0:49:D5:02:5F:38,CN=QMA,OU=IBM MQ
Development,O=IBM,ST=Hampshire,C=UK")
STATUS(RUNNING)                    SUBSTATE(RECEIVE)
```

DISPLAY CHSTATUS 輸出中的 SSLPEER 欄位會顯示在步驟 2 中建立之遠端用戶端憑證的主體 DN。發證者名稱符合步驟 4 中所新增簽署個人憑證之 CA 憑證的主體 DN。

以匿名方式將用戶端連接至佇列管理程式

遵循這些範例指示來修改具有交互鑑別的系統，以容許佇列管理程式匿名連接至另一個。

關於這項作業

測試情境:

- 您的佇列管理程式及用戶端 (QM1 及 C1) 已設定在 [第 98 頁的『使用 CA 簽章憑證進行用戶端及佇列管理程式的交互鑑別』](#) 中。
- 您想要變更 C1，以便它以匿名方式連接至 QM1。

產生的配置如下所示:

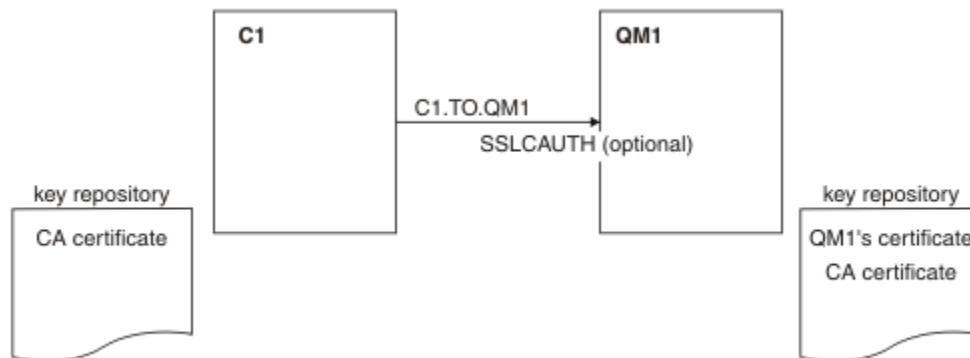




圖 21: 容許匿名連線的用戶端及佇列管理程式

程序

1. 根據作業系統，從 C1 的金鑰儲存庫中移除個人憑證:

-  IBM i 系統。
-  AIX, Linux, and Windows 系統。

憑證標籤是 `ibmwebsphermq` 後接小寫的登入使用者 ID，或 `CERTLABL` 屬性的值。請參閱 [數位憑證標籤](#)。

2. 重新啟動用戶端應用程式，或讓用戶端應用程式關閉並重新開啟所有 SSL/TLS 連線。
3. 發出下列指令，在佇列管理程式上容許匿名連線:

```
ALTER CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) SSLCAUTH(OPTIONAL)
```

結果

金鑰儲存庫和通道已變更，如 [第 100 頁的圖 21](#) 中所示

下一步

在通道的伺服器端，通道狀態顯示畫面上存在對等節點名稱參數值表示已傳送用戶端憑證。

請發出一些 DISPLAY 指令來驗證作業已順利完成。如果作業成功，則產生的輸出類似於下列範例中所示的輸出：

從佇列管理程式 QM1，輸入下列指令：

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
```

產生的輸出將類似於下列範例：

```
DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
5 : DISPLAY CHSTATUS(C1.TO.QM1) SSLPEER SSLCERTI
AMQ8417: Display Channel Status details.
CHANNEL(C1.TO.QM1)           CHLTYPE(SVRCONN)
CONNNAME(192.0.0.1)         CURRENT
SSLCERTI( )                 SSLPEER( )
STATUS(RUNNING)            SUBSTATE(RECEIVE)
```

SSLCERTI 和 SSLPEER 欄位是空的，表示 C1 未傳送憑證。

Windows 在 Windows 上移轉

從現有的 IBM MQ 9.2 安裝開始，此實務範例會引導您完成將資料升級及移轉至 IBM MQ 9.3 所需的主要作業。這兩個版本都安裝在相同的 Windows 環境中。

Windows 規劃解決方案

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，以及實務範例所提出解決方案的概觀。

相關工作

[規劃安裝](#)

Windows 假設

此實務範例會對您用來設定及使用範例 IT 配置的系統進行數個假設。這些假設包括您正在使用之產品的作業系統及版本，以及您是否已針對 IBM MQ 配置安全。

此實務範例假設下列要點：

- 在此實務範例中，您使用一部具有 Windows 作業系統的電腦，您將在其上安裝起始 IBM MQ 9.2 配置，然後安裝 IBM MQ 9.3。
註：此實務範例中未說明叢集作業。提供了安裝範例 IBM MQ 單一伺服器配置的指示，您可以使用該配置作為起點，以與最初開發的相同方式來試用實務範例。
- 您使用下列版本的 IBM MQ：
 - 對於起始範例配置，您是使用 IBM MQ 9.2。
 - 對於後置移轉配置，您是使用 IBM MQ 9.3。
- 此實務範例未說明 IBM MQ 的安全配置。如果您已配置安全，則仍應該能夠完成實務範例。

- 您可以使用 Windows 命令提示字元及圖形使用者介面 IBM MQ Explorer 來完成此實務範例中說明的作業。

相關概念

[移轉路徑](#)

Windows 業務概觀

公司想要將 Windows 作業系統上的現有 IBM MQ 9.2 IT 配置移轉至 IBM MQ 9.3。

公司決定將其商業解決方案移轉至 IBM MQ 9.3 以獲得商業價值，包括：

- 使用 IBM MQ 9.3 中提供的新增及更新功能。
- 探索可從 IBM MQ 9.3; Continuous Delivery Release (CDR) 取得的新版本類型。
- 受益於 Windows 平台上的 LDAP 授權。

相關概念

[IBM MQ 版本類型及版本化](#)

[相關資訊](#)

[IBM MQ 長期支援及持續交付版次的常見問題](#)

Windows 選擇移轉路徑

當您在 IBM MQ 9.2 與 IBM MQ 9.3 之間移轉時，您可以採用數個移轉路徑。

本主題提供下列移轉路徑的概觀：

- 單一階段路徑，也稱為獨立式移轉
- 並列移轉路徑
- 多階段移轉路徑

附註：此實務範例僅說明「單一階段」及「並列」移轉方法。

請考量每一個路徑的優點和限制，以判斷最適合您的需求：

單一階段移轉

在單一階段移轉中，產品最新版本的安裝會取代相同安裝位置中的舊版。

單一階段移轉的優點是它儘可能在舊版上變更佇列管理程式的配置。現有的應用程式會從舊版載入程式庫，自動切換至載入最新版本的程式庫。使用此方法，您的系統將無法用於整個處理程序。

並列移轉

在並列移轉中，您將 IBM MQ 的最新版本與繼續與舊版相關聯的佇列管理程式一起安裝。

當您備妥時，會將佇列管理程式及應用程式移轉至最新版本。

使用此方法，因為您在啟動任何佇列管理程式之前先解除安裝舊版，所以您可以將最新版本的安裝指派為主要安裝。

如需進一步資訊，請參閱 [選擇主要安裝](#)。

多階段移轉

在多階段移轉中，您可以在執行中佇列管理程式的同時安裝最新版本的產品，這些佇列管理程式會繼續與舊版相關聯。您可以使用最新版本安裝來建立佇列管理程式及執行新的應用程式。當您準備好開始從先前的移轉佇列管理程式和應用程式時，您可以逐一執行。當移轉至最新版本完成時，請解除安裝舊版，並將最新版本安裝設為主要安裝。

使用多階段方法，在解除安裝舊版之前，您必須配置環境來執行應用程式，以將佇列管理程式連接至最新版本。您也必須提供執行 IBM MQ 指令的路徑。這兩項作業都是使用 [setmqenv](#) 指令來達成。

相關概念

[移轉概念及方法概觀](#)

相關工作

[在 AIX and Linux 上移轉: 單一階段](#)

[在 AIX and Linux 上移轉: 並列](#)

[在 AIX and Linux 上移轉: 多階段](#)

相關參考

[影響移轉的變更清單](#)

Windows 技術解決方案

此實務範例說明兩種從舊版 IBM MQ 移轉至更新版本的方法，其中兩個版本都在 Windows 作業系統上執行，且位於相同伺服器上。

Windows 概觀: 起始 IT 配置

公司使用 IBM MQ 9.2 所提供的現有 IT 配置，該配置安裝在具有 Windows 作業系統的伺服器上。此實務範例說明將起始 IT 配置移轉至相同伺服器上 IBM MQ 9.3 所提供的對等 IT 配置。

起始 IT 配置包括管理者配置或使用的數個元件，如 [第 104 頁的圖 22](#) 所示：

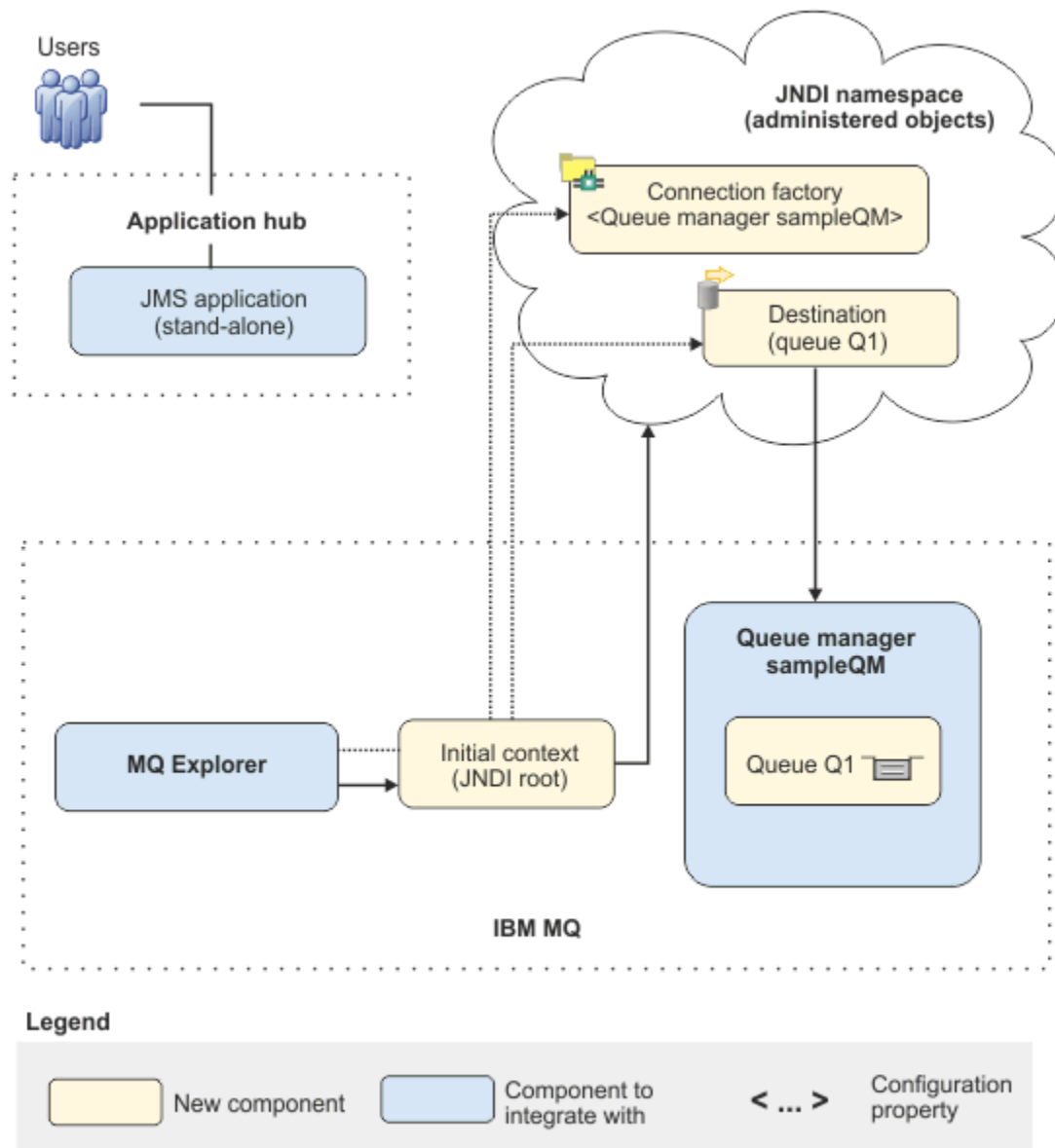


圖 22: 起始 IT 配置

JMS 應用程式

商業使用者與之互動的獨立式應用程式，例如登錄訂單。應用程式使用 Java Message Service (JMS) 進行非同步傳訊。

- JMS 是廣泛支援的 Java EE 傳訊標準。因此，JMS 型應用程式在許多傳訊產品之間是可攜的。
- JMS 提供傳訊層詳細資料的抽象層次，簡化應用程式開發程序。
- JMS 提供非同步通訊，可讓應用程式執行而無需等待回覆，與緊密連結的系統 (例如遠端程序呼叫 (RPC)) 不同。
- 使用 JMS 的應用程式不會直接指定存取資源的詳細資料。相反地，他們會查閱並使用受管理 JMS 物件，例如 Connection Factory 和目的地。

在某些狀況下，其他傳訊標準可能比 JMS 更適合。例如，IBM Message Service Client for C、C++ 及 .NET (也稱為 XMS) 是 API，提供類似於 JMS for 非 Java 應用程式的好處。因此，如果您使用 .NET 平台，或想要整合現有 C++ 應用程式與較新的 Java EE 應用程式，則 XMS 更適合。

應用程式使用點對點傳訊，將訊息傳送至基礎架構中的佇列，並處理回覆訊息，以提供商業使用者適當的回應。

在此傳訊模型中，應用程式會將訊息傳送至佇列，而另一個應用程式會從佇列接收訊息並確認接收訊息。此模型是最簡單的傳訊形式，因為它只涉及兩個端點。此模型也最適合實務範例應用程式：單一用戶端從單一伺服器要求資訊。

在替代傳訊模型（發佈/訂閱）中，發佈者會將訊息發佈至訊息主題。訂閱者訂閱主題以接收訊息。發佈者和訂閱者沒有彼此的任何相關資訊，且訊息由零個以上的收件者接收。

佇列管理程式 **sampleQM**

提供起始傳訊基礎架構的 IBM MQ 佇列管理程式。它會管理 JMS 應用程式使用的佇列。

Q1 [訊息佇列]

JMS 應用程式向其傳送訊息的 IBM MQ 佇列。

JNDI 名稱空間

Java Naming Directory Interface JNDI 名稱空間用來保留 JMS 受管理物件，應用程式可以使用這些物件來連接至 IBM MQ，以及存取目的地來傳送或接收訊息。

JNDI 是 Java EE 的一部分，並為應用程式提供標準方法來存取各種類型的命名和目錄服務，以擷取應用程式元件。例如，您可以使用 JNDI 來存取檔案系統上的命名服務，以擷取印表機物件的位置，或存取 LDAP 伺服器上的目錄服務，以擷取包含 ID 及密碼資訊的使用者物件。因此，JNDI 會加強 JMS 型應用程式的可攜性，並讓您更容易將那些應用程式彼此整合，以及整合到現有的系統中。對於 JMS 傳訊，您可以使用 JNDI 來儲存代表訊息目標目的地的物件，或儲存在應用程式與其傳訊目的地之間建立連線的 Connection Factory。

任何具有 JNDI 名稱空間存取權的應用程式或處理程序都可以使用相同的受管理物件。在「JNDI」中可以變更受管理物件的內容，所有應用程式或處理程序都可以受益於那些相同的變更。

起始環境定義

起始環境定義會定義 JNDI 名稱空間的根目錄。若要使用 IBM MQ Explorer 來建立及配置受管理物件，請先新增起始環境定義，以定義 JNDI 名稱空間的根目錄。同樣地，JMS 應用程式會先取得起始環境定義，然後才能從 JNDI 名稱空間擷取受管理物件。

Connection Factory **myCF**

JMS Connection Factory 物件會定義一組連線的標準配置內容。應用程式使用 Connection Factory 來建立與 IBM MQ 的連線。

目的地，**myQueue**

JMS 目的地可以是主題或佇列。在此實務範例中，目的地是佇列，並識別應用程式向其傳送訊息的 IBM MQ 佇列及/或應用程式從中接收訊息的佇列。應用程式會在 JNDI 名稱空間中查閱目的地，以建立與 IBM MQ 佇列的連線。

Windows 概觀: 遞送的邏輯拓撲

公司已從 IBM MQ 9.2 移轉至 IBM MQ 9.3。

移轉之後的 IT 配置保持不變，如第 106 頁的圖 23 所示。佇列管理程式及範例應用程式會移轉，且企業可以從最新版本 IBM MQ 中的新功能獲益。

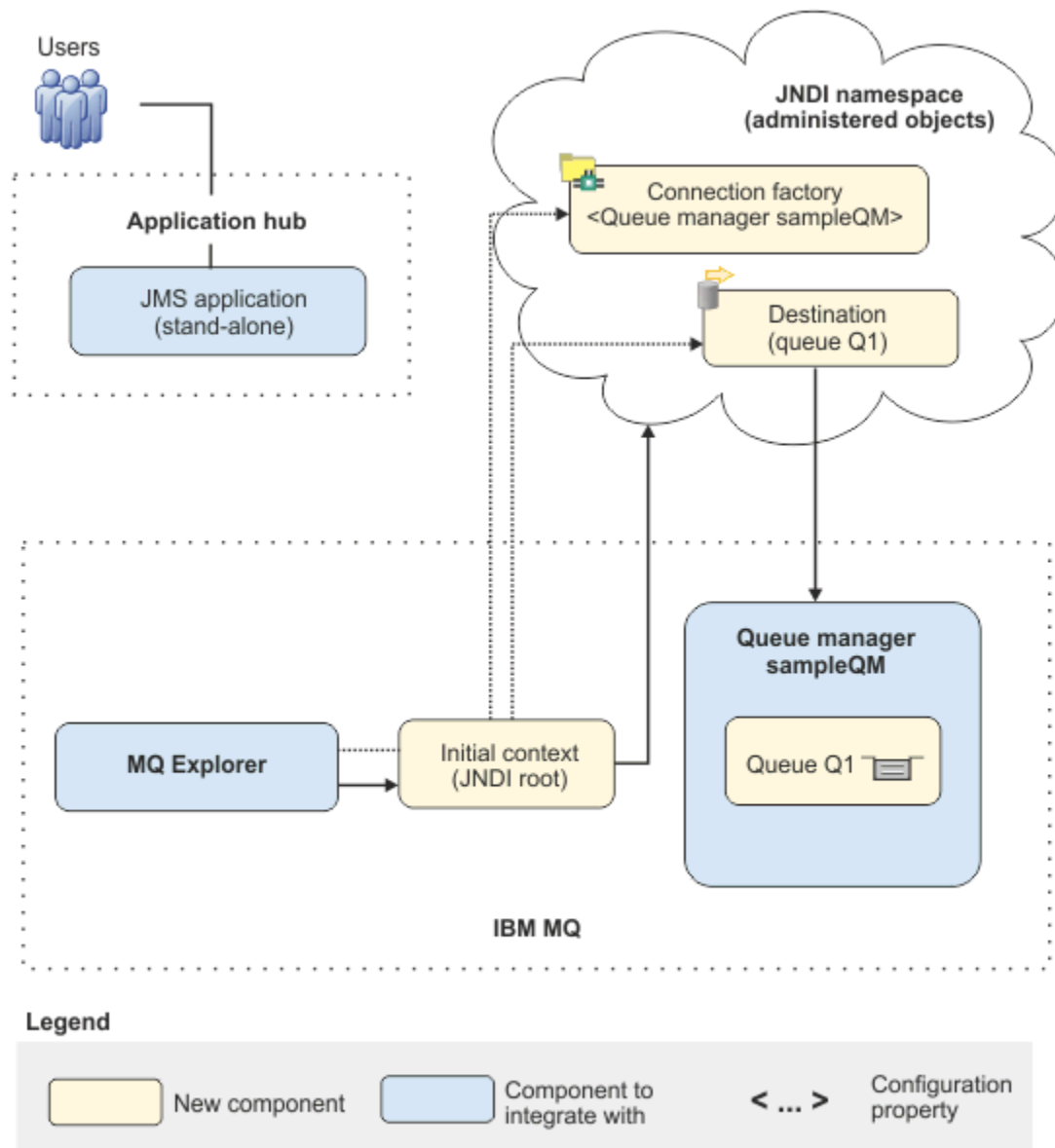


圖 23: 交付的 IT 配置

Windows 使用圖形使用者介面實作解決方案

在此實務範例中實作解決方案包括使用圖形使用者介面，從在 Windows 作業系統上執行的舊版 IBM MQ 移轉至也在 Windows 作業系統上執行的新版次，然後驗證佇列管理程式及佇列是否已順利移轉至新版次。

開始之前

如果您想要試用此實務範例，請先遵循指示來設定範例傳訊基礎架構的副本，如第 107 頁的『[建立起始 IT 配置](#)』中所述。此範例配置以 IBM MQ 9.2 為基礎。

關於這項作業

此實務中說明的移轉處理程序顯示將佇列管理程式從 IBM MQ 9.2 移轉至 IBM MQ 9.3。

此實務範例包括兩個用於移轉方法的選項。您可以選擇嘗試其中一個或另一個選項，或兩者。

選項 1: 單一階段移轉

在單一階段移轉中，產品更新版本的安裝會取代相同安裝位置中的舊版。

單一階段移轉的優點是它儘可能在舊版上變更佇列管理程式的配置。現有的應用程式會從舊版載入程式庫，自動切換至載入最新版本的程式庫。使用此方法，您的系統將無法用於整個處理程序。

選項 2: 並列移轉

在並列移轉中，您可以安裝更新版本的 IBM MQ 以及更舊版本。佇列管理程式會繼續與舊版相關聯，直到您準備好將它們移轉至新版本為止。

使用並列方法，因為您在更新版本中啟動任何已移轉的佇列管理程式之前，先解除安裝舊版，您可以將更新版本的安裝指派為主要安裝。

程序

1. 建立範例起始 IT 配置，以用作實務範例的起始點，如第 107 頁的『[建立起始 IT 配置](#)』中所述。
2. 選取您要用來移轉產品的方法，然後遵循所選選項的指示：
 - 第 115 頁的『[選項 1: 單一階段移轉](#)』
 - 第 121 頁的『[選項 2: 並列移轉](#)』

相關工作

[在 AIX and Linux 上移轉: 單一階段](#)

[在 AIX and Linux 上移轉: 並列](#)

[在 Windows 上將佇列管理程式從舊版移轉至最新版本](#)

[選擇主要安裝](#)

Windows 建立起始 IT 配置

此實務範例是使用範例起始 (IT) 配置來開發。遵循指示來設定此範例配置，以與最初開發的相同方式來試用實務範例。

關於這項作業

此實務範例的起始 IT 配置 (在第 103 頁的『[概觀: 起始 IT 配置](#)』中說明) 包含為 IBM MQ Explorer 新增的起始環境定義，以連接至 JNDI 名稱空間的根目錄。JNDI 名稱空間包含針對範例 JMS 應用程式新增用來連接 IBM MQ 的 Connection Factory，以及針對範例 JMS 應用程式新增用來連接 IBM MQ 佇列的目的地。該 IBM MQ 佇列也已新增至起始 IT 配置，並由範例 JMS 應用程式使用。

程序

1. [安裝 IBM MQ 9.2 並驗證安裝](#)。
2. [配置 JNDI 名稱空間和受管理物件](#)。
3. [驗證範例 IT 配置](#)。

Windows 使用啟動程式來安裝 IBM MQ 9.3

使用安裝啟動程式及精靈來安裝您要設定為起始 IT 配置的 IBM MQ 版本，以作為此實務範例的起點。

開始之前

在啟動此作業之前，請完成下列檢查：

- 安裝時必須具有本端管理者權限。
- 請確定機器名稱不包含任何空格。
- 請確定您有足夠的磁碟空間。如需相關資訊，請參閱 [Multiplatforms 上的磁碟空間要求](#)。

在此實務範例中，不需要判斷您是否需要為任何 IBM MQ 使用者定義 Windows 網域使用者 ID，因為此需求超出此實務範例的範圍。如需相關資訊，請參閱 [為 IBM MQ 建立 Active Directory 及 DNS 網域](#)。

在安裝 IBM MQ 之前，請檢查您的系統是否符合軟硬體需求。如需軟硬體需求的最新詳細資料，請參閱 [IBM MQ 的系統需求](#)。

關於這項作業

此作業說明在 Windows 作業系統上安裝 IBM MQ 的基本步驟。

安裝程式包含進一步資訊的鏈結。安裝程序具有下列組件：

1. 啟動安裝程序。
2. 使用「安裝啟動程式」來檢查及安裝軟體需求、指定網路資訊，以及啟動 IBM MQ 安裝精靈。
3. 使用 IBM MQ 安裝精靈來安裝軟體，並啟動 Prepare IBM MQ Wizard。
4. 使用 Prepare IBM MQ Wizard 來啟動 IBM MQ 服務。

程序

1. 啟動安裝程序。

在「Windows 檔案總管」中，導覽至已下載安裝映像檔的暫存資料夾，然後按兩下 setup.exe。

已啟動安裝啟動程式。

2. 使用啟動程式，檢閱並在必要時修改軟體需求及網路配置。

a) 按一下 **軟體需求** 按鈕，以顯示 **軟體需求** 標籤。

b) 檢查是否已符合軟體需求，以及需求項目是否顯示帶有 OK 單字的綠色勾號。進行任何指出的更正。

註：如果要查看任何需求的詳細資料，請按一下加號 (+) 按鈕。

c) 按一下 **網路配置** 按鈕，以顯示 **網路配置** 標籤。

d) 按一下 **否** 圓鈕。

註：此實務範例假設您不需要配置 IBM MQ 的網域使用者 ID。如需配置 IBM MQ for Windows 網域使用者的相關資訊，請按一下 **相關資訊** 按鈕。

e) 在「啟動程式」的 **IBM MQ 安裝** 標籤上，選取安裝語言，然後按一下 **啟動 IBM MQ 安裝程式** 按鈕以啟動 IBM MQ 安裝精靈。

您已完成 IBM MQ 安裝需求的檢閱，已進行任何必要的修改，並已啟動 IBM MQ 安裝精靈。

3. 使用 IBM MQ 安裝精靈來安裝軟體，以及啟動 Prepare IBM MQ Wizard。

a) 在 IBM MQ 安裝精靈中，閱讀「授權合約」，並選取 **我接受授權合約中的條款** 勾選框，然後按 **下一步**。

b) 按 **一般**，然後按 **下一步**。

c) 在「**準備安裝 IBM MQ**」頁面上，檢閱安裝資訊，然後按一下 **安裝**。

註：下列詳細資料：

- 安裝名稱
- 程式檔案的最上層資料夾
- 資料檔的最上層資料夾

將安裝下列特性：

- IBM MQ 伺服器
- IBM MQ: 用於管理及監視 IBM MQ 資源的圖形介面
- Java™ 和 .NET 傳訊及 Web 服務
- IBM MQ 開發工具箱

安裝程序開始。視您的系統而定，安裝程序可能需要數分鐘。

在安裝程序結束時，「IBM MQ 設定」視窗會顯示訊息 Installation Wizard Completed Successfully。

d) 按一下**完成**。

您已順利安裝 IBM MQ。Prepare IBM MQ Wizard 會自動啟動，並顯示 **歡迎使用 Prepare IBM MQ Wizard** 頁面。

4. 使用「準備 MQ」精靈來啟動 IBM MQ 服務。

a) 在「歡迎使用 Prepare IBM MQ Wizard」頁面上，選取**下一步**。

Prepare IBM MQ Wizard 會顯示訊息 Status: Checking IBM MQ Configuration 及進度列。當程序完成時，會顯示「IBM MQ 網路配置」頁面。

b) 在 Prepare IBM MQ Wizard 的「IBM MQ 網路配置」頁面上，選取**否**。

c) 按**下一步**。

Prepare IBM MQ Wizard 會顯示訊息 Status: starting the IBM MQ Service 及進度列。當程序完成時，精靈會顯示「完成 Prepare IBM MQ Wizard」頁面。

d) 選取 **啟動 IBM MQ Explorer**，並選擇是否要啟動「記事本」來檢視版本注意事項，然後按一下 **完成**。

IBM MQ Explorer 啟動。

您已安裝 IBM MQ，且已啟動 IBM MQ Explorer。

結果

IBM MQ 已安裝在您的電腦上。

下一步

您已準備好建立在此實務範例中使用的受管理物件，如 [第 109 頁的『配置 JNDI 名稱空間和受管理物件』](#) 中所述。

相關概念

[Windows 系統上的軟硬體需求](#)

[IBM MQ 簡介](#)

相關工作

[在 Windows 上安裝 IBM MQ 伺服器](#)

[配置 IBM MQ 伺服器](#)

Windows 配置 JNDI 名稱空間和受管理物件

在 IBM MQ Explorer 中定義 JNDI 名稱空間的起始環境定義，然後在名稱空間中定義範例應用程式可以使用的受管理物件。

關於這項作業

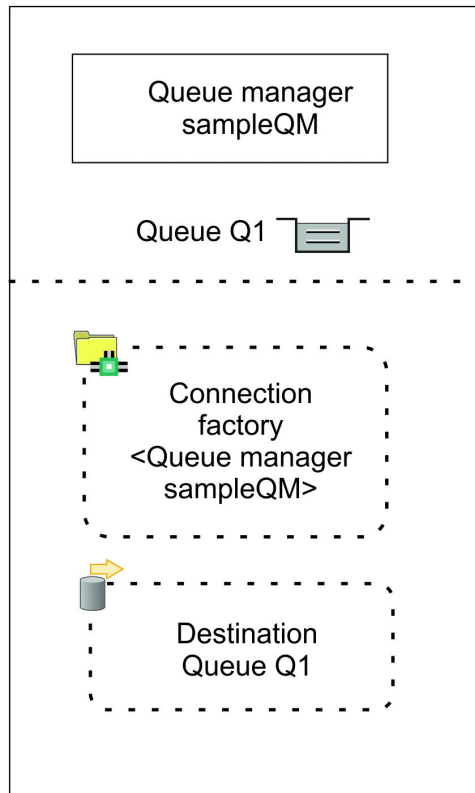
在這項作業中，您將在 IBM MQ 中建立下列物件。

- 位於本端檔案系統中的 JNDI 名稱空間。使用檔案系統，因為它是範例實務最簡單的 JNDI 機制。

JNDI 名稱空間可以位於檔案系統、輕量型目錄存取通訊協定 (LDAP) 伺服器或另一個 JNDI 實作上。如果您想要在 LDAP 伺服器或另一個 JNDI 實作上使用 JNDI 名稱空間，您必須配置 JNDI 名稱空間，並依照實作的需要，修改範例應用程式來參照 JNDI 名稱空間。

- JNDI 名稱空間中的受管理物件。JMS 應用程式可以查閱受管理物件，以連接至 IBM MQ，並存取用來傳送或接收訊息的 IBM MQ 目的地。

WebSphere MQ



WebSphere MQ JNDI Namespace

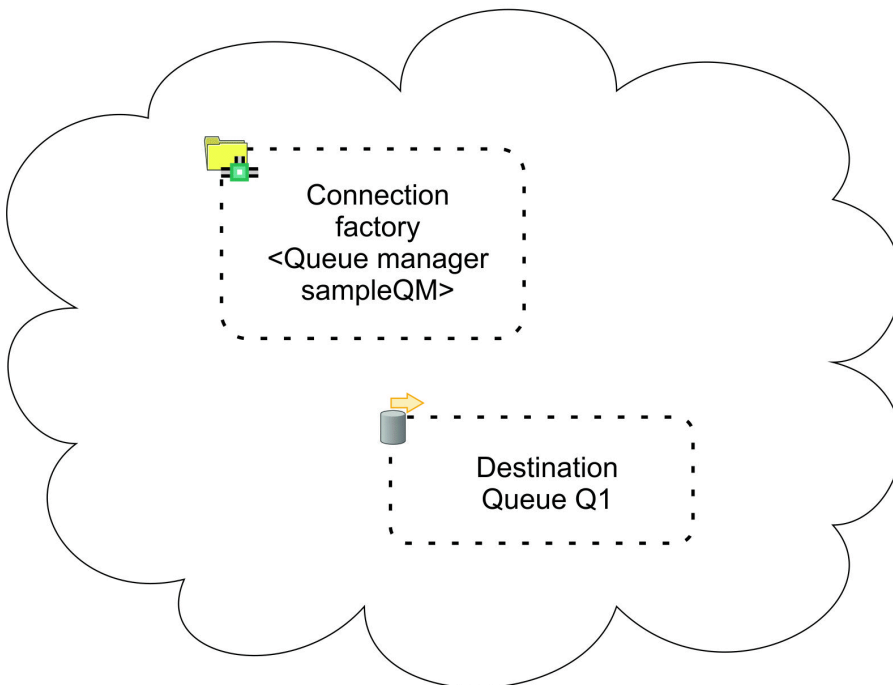


圖 24: 在 IBM MQ 中建立的物件

程序

- 按一下 **啟動 > 所有程式 > IBM MQ > IBM MQ Explorer**，以啟動 IBM MQ Explorer(如果尚未啟動)。如果 IBM MQ Explorer 正在執行並顯示「歡迎使用」頁面，請關閉「歡迎使用」頁面，以開始管理 IBM MQ 物件。
- 建立要用於範例應用程式的佇列管理程式。
 - 用滑鼠右鍵按一下 **佇列管理程式**，然後選取 **新建 > 佇列管理程式 ...**。即會啟動「建立佇列管理程式」精靈。
 - 在 **佇列管理程式名稱** 欄位中，輸入 `sampleQM`。
您可以為佇列管理程式選擇不同的名稱，但必須記得在稍後的配置步驟中使用它，以取代 `sampleQM`。
註：從下列字集中，名稱不得超過 48 個字元：
 - 大寫或小寫字元 (A-Z a-z)
 - 數值字元 (0-9)
 - 句點 (.)
 - 正斜線 (/)
 - 底線 (_)
 - 百分比符號 (%)名稱區分大小寫。相同類型的物件必須有不同的名稱。例如，兩個佇列不能同名，但佇列管理程式和佇列可以。
 - 在 **無法傳送郵件的佇列** 欄位中，輸入 `SYSTEM.DEAD.LETTER.QUEUE`。
此欄位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。
無法傳送郵件的佇列會儲存無法遞送至正確目的地的訊息，例如，因為佇列已滿。所有佇列管理程式都應該有相關聯的無法傳送郵件的佇列。
 - 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。
 - 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)。將指定的埠號 (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，「IBM MQ」會顯示一個對話框視窗。
- 新增 **JNDI** 名稱空間的起始環境定義，然後將 IBM MQ Explorer 連接至該環境定義
您必須先新增起始環境定義，以定義儲存受管理物件之 JNDI 名稱空間的根目錄，然後才能使用 IBM MQ Explorer 來建立及配置 JMS 受管理物件。
每當您想要使用 IBM MQ Explorer 來建立或管理 JNDI 名稱空間中的受管理物件時，必須將 IBM MQ Explorer 連接至 JNDI 名稱空間的起始環境定義。
 - 在 IBM MQ Explorer - **Navigator** 窗格中，用滑鼠右鍵按一下 **JMS 受管理物件**，然後選取 **新增起始環境定義 ...**。
此動作會顯示 " **連線詳細資料** " 頁面。
 - 在 " **JNDI 名稱空間位於何處?** " 下，選取 **檔案系統** 勾選框。
 - 在 **連結目錄** 欄位中，輸入 `C:\JNDI-Directory`。
此值符合範例 JMS 應用程式中指定的 JNDI 名稱空間位置。如果您必須指定不同的 JNDI 目錄，您必須修改要符合的應用程式。
如果系統上不存在該目錄，則視窗會顯示 `Specified location does not exist or is not readable` 訊息。按一下 **瀏覽 ...** 如果要開啟檔案系統視窗，請導覽至 Local Disk (C:)，然後按一下 **建立新資料夾** 以建立 JNDI-Directory 資料夾。按一下 **確定**。
按 **下一步**。
 - 在 **使用者喜好設定** 頁面上，保留預設值。

- **環境定義暱稱:** JNDI 名稱空間的位置將作為暱稱，以在 IBM MQ Explorer 中顯示起始環境定義。
- **完成時立即連接:** 當您完成建立起始環境定義時，這個選項會將 IBM MQ Explorer 連接至 JNDI 名稱空間，以便您可以立即建立受管理物件。
- **啟動時自動重新連接至環境定義:** 未選取此選項，因為您通常不需要在每次關閉並重新開啟 IBM MQ Explorer 時 IBM MQ Explorer 自動重新連接至起始環境定義。

如果您定期使用「IBM MQ Explorer」來建立或管理 JNDI 名稱空間中的受管理物件，您可以選取 **啟動時自動重新連接環境定義** 勾選框，讓 IBM MQ Explorer 在每次啟動 IBM MQ Explorer 時自動重新連接起始環境定義。此選項可讓您不必手動將 IBM MQ Explorer 連接至起始環境定義。

按一下 **完成**，以建立並顯示起始環境定義。

4. 建立 Connection Factory 受管理物件。

Connection Factory 受管理物件定義一組連線的標準配置內容。應用程式使用 Connection Factory 來建立與 IBM MQ 的連線。

a) 在 IBM MQ Explorer - **Navigator** 窗格中，展開 **JMS 受管理物件**，然後展開標籤為 **file:/C:/JNDI-Directory/** 的起始環境定義。

b) 用滑鼠右鍵按一下 **Connection Factory**，然後選取 **新建 > Connection Factory ...**。

此動作會顯示 **新建 Connection Factory** 精靈

c) 在「名稱」欄位中，輸入 myCF

範例 JMS 應用程式包含可查閱名稱為 myCF 的 Connection Factory 的程式碼。如果您必須使用不同的名稱，則必須修改應用程式以符合。

IBM MQ 用於傳訊提供者，因為範例應用程式使用點對點傳訊。

按下一步。

d) 將 Connection Factory 類型保留為 **Connection Factory**，因為此選項最適合一般 JMS 使用。

與網域無關的 Connection Factory 可讓 JMS 應用程式同時使用點對點傳訊和發佈/訂閱傳訊，尤其是當您想要 JMS 應用程式在相同交易下執行這兩種類型的傳訊時。

如果 JMS 應用程式預期只使用點對點傳訊或發佈/訂閱傳訊，您可以在建立 Connection Factory 時選取特定的傳訊網域，並建立網域特定的 (佇列或主題) Connection Factory。

e) 維持清除 XA 交易的支援。

範例應用程式不使用符合 XA 標準的交易。

IBM MQ JMS 在連結模式下支援符合 XA 標準的交易。如果您想要範例應用程式使用符合 XA 標準的交易，則必須修改範例應用程式。

按下一步。

f) 將傳輸保留為 **連結**。

使用 Connection Factory 的範例 JMS 應用程式與佇列管理程式在相同電腦上執行，因此可以使用「連結」模式傳輸。此選項表示 JMS 應用程式直接連接至佇列管理程式，並提供相對於替用戶端模式的效能優勢。

按下一步，然後再按下一步。

g) 在「變更內容」頁面上，從左側功能表中選取 **連線**，然後在 **連線** 窗格中選取 sampleQM 作為 **基本佇列管理程式**。

基本佇列管理程式是應用程式將連接的佇列管理程式。如果您想要應用程式能夠連接多個佇列管理程式，請將此值保留空白。

h) 按一下 **完成**。

IBM MQ 會顯示一個對話框視窗，以顯示已順利建立物件。按一下 **確定** 以關閉對話框視窗。

5. 建立目的地受管理物件。

目的地受管理物件可識別應用程式將訊息傳送至的 IBM MQ 佇列，以及應用程式從中接收訊息的佇列。應用程式會在 JNDI 名稱空間中查閱目的地，以建立與 IBM MQ 佇列的連線。

在發佈/訂閱傳訊中，目的地會識別主題而非佇列。

- a) 在 IBM MQ Explorer - **Navigator** 窗格中，展開 **JMS 受管理物件**，然後展開標籤為 **file:/C:/JNDI-Directory/** 的起始環境定義。
 - b) 用滑鼠右鍵按一下 **目的地**，然後選取 **新建 > 目的地 ...**。
即會顯示「**新建目的地**」精靈。
 - c) 在 **名稱** 欄位中，輸入 myQueue。
將 **類型** 保留為 **佇列**。
範例 JMS 應用程式包含的程式碼會查閱名為 myQueue 的目的地。範例 JMS 應用程式使用點對點傳訊，因此需要佇列類型的目的地。主題類型的目的地用於發佈/訂閱傳訊。
 - d) 選取 **啟動精靈以建立相符的 MQ 佇列** 勾選框。
目的地物件需要相符的 IBM MQ 佇列，且使用 IBM MQ Explorer 可方便兩者一起建立。當您完成「**新建目的地**」精靈時，會開啟「**建立 MQ 佇列**」精靈，其中有許多目的地詳細資料對映至 IBM MQ 佇列。
按下一步。
再按下一步。
 - e) 在 "變更內容" 頁面上，按一下 **選取 ... 佇列管理程式** 旁。選取您先前建立的 sampleQM 佇列管理程式，然後按一下 **確定**。
 - f) 指定 Q1 作為 IBM MQ 佇列的名稱。
您可以為佇列選擇不同的名稱，但必須記得在稍後的配置步驟中使用它，以取代 Q1。
註: 從下列字集中，名稱不得超過 48 個字元：
 - 大寫或小寫字元 (A-Z a-z)
 - 數值字元 (0-9)
 - 句點 (.)
 - 正斜線 (/)
 - 底線 (_)
 - 百分比符號 (%)名稱區分大小寫。相同類型的物件必須有不同的名稱。例如，兩個佇列不能同名，但佇列管理程式和佇列可以。
 - g) 按一下 **完成**。
即會啟動「**建立 MQ 佇列**」精靈。
如果精靈未啟動，您可能未在先前步驟中選取 **啟動精靈以建立相符的 MQ 佇列** 勾選框。在 IBM MQ Explorer - **Navigator** 窗格中，展開 **sampleQM** 佇列管理程式，用滑鼠右鍵按一下 **佇列**，然後選取 **新建 > 本端佇列 ...**。
6. 建立相符的 IBM MQ 佇列。
先前建立的目的地受管理物件代表 IBM MQ 佇列。此佇列是儲存 JMS 訊息的位置。
- a) 按 **下一步**，接受您先前指定的 sampleQM 佇列管理程式。
 - b) 按 **下一步**。
 - c) 按一下 **完成**，以使用您先前建立的目的地受管理物件中的資訊來建立 IBM MQ 佇列。
IBM MQ 會顯示一個對話框視窗，其中包含已順利建立物件的訊息。
現在可以在佇列管理程式下的 **佇列** 區段中看到新佇列。

結果

您現在已建立使用範例 JMS 應用程式所需的 IBM MQ 物件。

下一步

現在，您已準備好驗證已正確配置 IBM MQ 以與範例應用程式搭配使用，如 [第 114 頁的『驗證範例 IT 配置』](#) 中所述。

Windows 驗證範例 IT 配置

執行範例獨立式 JMS 應用程式，以透過 IBM MQ 傳送及接收訊息，並驗證您已正確配置 IBM MQ 以與範例應用程式搭配使用。

開始之前

下載範例應用程式套件。按下列鏈結，並將檔案儲存至您要安裝 IBM MQ 的電腦: [sampleJMSApp.zip](#)，然後解壓縮內容。套件包含用於執行應用程式的範例 JMS 應用程式 .jar 檔案及批次檔。

- 範例 sampleJMSApp.jar 檔和 .cmd 檔必須位於相同的目錄中。
- .cmd 檔使用環境變數來設定執行 JMS 應用程式的類別路徑。執行 JMS 應用程式時，如果您看到 Java java.lang.NoClassDefFoundError，則可能需要在指令檔中調整類別路徑行。

關於這項作業

JMS 應用程式包含傳送起始訊息的要求端用戶端，以及接收訊息並傳送回覆的回應端用戶端。提供的批次檔會執行下列動作：

- runresponder.cmd 會開啟命令提示字元視窗，回應者用戶端會在其中啟動，然後等待訊息。
- runrequester.cmd 會開啟個別的命令提示字元視窗，要求端用戶端會在其中啟動，然後傳送要求訊息並接收回覆。

使用兩個命令提示字元視窗，您可以分別且更清楚地查看要求者及回應者的動作。

程序

1. 按兩下 runresponder.cmd 檔案。

在標示為 **回應者視窗** 的命令提示字元視窗中，回應者用戶端會啟動，然後等待訊息。

```
> Connection factory located in JNDI.> Destination located in JNDI.> Creating connection to
QueueManager.> Created connection.
> Waiting for message.
```

2. 按兩下 runrequester.cmd 檔案。

在 **要求端視窗** 中，觀察要求端訊息。在「**回應者**」視窗中，觀察已更新的回應者訊息；它所接收的訊息（來自要求者用戶端），以及它所傳送的回覆訊息。

結果

在標示為 **要求端視窗** 的命令提示字元視窗中，要求端用戶端會顯示連線狀態、它傳送的訊息，以及它從回應端用戶端收到的回覆訊息：

```
> Connection factory located in JNDI.> Destination located in JNDI.> Creating connection to
QueueManager.> Connection created.
> Sending stock request for 'BakedBeans'> Sent Message
ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002b02
> Received Message ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002902 for 'B
akedBeans - 15 tins in stock'
> Closing connection to QueueManager.> Closed Connection.
-----
In this window, observe the messages sent through IBM MQ:
- The request message sent
- The reply message received
-----
When ready, press any key to close this window
Press any key to continue . . .
```

在「**回應者**」視窗中，觀察已更新的回應者訊息；它所接收的訊息（來自要求者用戶端）及它所傳送的回覆訊息：

```
> Connection factory located in JNDI.> Destination located in JNDI.> Creating connection to
QueueManager.> Created connection.
> Waiting for message.

> Received Message ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002b02 for 'B
akedBeans'
> Sending Reply Message 'BakedBeans - 15 tins in stock'> Sent Message
ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002902
> Closing connection to QueueManager.> Closed connection.
-----
In this window, observe the updated responder messages
- The request message received (from the requester)
- The reply message sent
-----
When ready, press any key to close this window
Press any key to continue . . .
```

兩個指令視窗中顯示的訊息會驗證範例應用程式的要求者及回應者用戶端可以透過 IBM MQ 彼此通訊。

下一步

現在，您已準備好使用下列兩個移轉選項之一，開始將範例 IBM MQ 9.2 安裝移轉至更新版本的 IBM MQ：

- 若要使用單一階段移轉方法進行移轉，請遵循第 115 頁的『[選項 1: 單一階段移轉](#)』中的指示。
- 如果要使用並列移轉方法來移轉，請遵循第 121 頁的『[選項 2: 並列移轉](#)』中的指示。

選項 1: 單一階段移轉

此實務範例的選項 1 顯示如何在使用單一階段移轉方法時，從舊版 IBM MQ 移轉至更新版本。透過單一階段移轉，更新版本 IBM MQ 的安裝會取代相同安裝位置中的舊版。

開始之前

此實務範例的起點是第 103 頁的『[概觀: 起始 IT 配置](#)』中說明的起始 IT 配置。

在啟動此作業之前，請遵循第 107 頁的『[建立起始 IT 配置](#)』中的指示來設定起始 IT 配置。

關於這項作業

使用單一階段移轉，您可以選擇在安裝更新版本之前先解除安裝舊版 IBM MQ，或安裝更新版本而不先解除安裝舊版（即就地移轉）。在這兩種情況下，較新的版本會安裝在與舊版相同的目錄中。此實務範例的選項 1 顯示單一階段移轉，在安裝較新版本之前，會先解除安裝較舊的版本。在解除安裝程序中不會移除佇列管理程式資料，這表示會保留此實務中使用的範例佇列管理程式，並在您安裝較新版本的 IBM MQ 時偵測到它們。

程序

1. 在舊版 IBM MQ 中執行的 [停止佇列管理程式](#)，並備份佇列管理程式資料。
2. 解除安裝您要從中移轉的舊版 IBM MQ，而不移除佇列管理程式資料。
3. 使用 [啟動程式安裝 IBM MQ 9.3](#)。
4. 使用 IBM MQ Explorer [來驗證新的 IBM MQ 9.3 安裝](#)。

請驗證已順利從舊版移轉佇列管理程式，且您可以在已移轉的佇列中放置訊息及取得訊息。

相關工作

在 AIX and Linux 上移轉: [單一階段](#)

Windows 準備移轉

在移轉至更新版本的 IBM MQ 之前，您必須先停止佇列管理程式並備份佇列管理程式資料。

關於這項作業

如果您在未先備份系統的情況下從舊版 IBM MQ 進行移轉，則無法回復至舊版，如果您決定不繼續移轉的話。在安裝新版本之前備份系統可讓您在必要時取消升級。不過，如果您確實回復升級，則無法回復較新版本 IBM MQ 所執行的任何工作，例如訊息及物件的變更。

在取得備份之前，請停止您要備份的佇列管理程式，對於此實務範例是 sampleQM。如果您嘗試建立執行中佇列管理程式的備份，備份可能不一致，因為在複製檔案時正在進行任何更新。

程序

1. 開啟 IBM MQ Explorer。

按一下 **啟動 > 所有應用程式 > IBM MQ > IBM MQ Explorer**。

2. 停止佇列管理程式 sampleQM。

a) 在「Navigator」視圖中，用滑鼠右鍵按一下佇列管理程式 sampleQM。

b) 按一下 **停止**。

即會開啟「**結束佇列管理程式**」視窗。

c) 選取 **控制項**，然後按一下 **確定**。

選取 **受控制** 選項會以受控制且有序的方式停止佇列管理程式。通常只有在受控制的停止無法順利完成時，才會使用強制佇列管理程式停止的 **立即** 選項。

佇列管理程式停止。在 IBM MQ 中，佇列管理程式 sampleQM 旁的圖示會變更為包含向下指向的紅色箭頭。

3. 關閉 IBM MQ Explorer。

4. 備份佇列管理程式資料。

取得下列所有資料的副本，確定您包括所有備份目錄。部分目錄可能是空的，但如果您需要在稍後日期還原備份，則全部都需要，因此也請儲存它們。

- 位於 C:\ProgramData\IBM\MQ\Qmgrs 中的佇列管理程式資料。
- 位於 C:\ProgramData\IBM\MQ\log 中之佇列管理程式的日誌檔目錄，包括日誌控制檔 amqh1ctl.lfh。
- 位於 C:\ProgramData\IBM\MQ\Config 中的配置檔。
- IBM MQ 9.2 .ini 檔案和登錄項目。佇列管理程式資訊儲存在 .ini 檔中，可用來回復成舊版產品。

5. 停止 IBM MQ。

a) 停止 IBM MQ 服務。

用滑鼠右鍵按一下系統匣中的 **IBM MQ** 圖示，然後按一下 **停止 IBM MQ**。

即會顯示具有下列訊息的對話框：

關閉 IBM MQ 會終止所有執行中的佇列管理程式，並
IBM MQ 程序。您確定要繼續嗎？(AMQ4102)

b) 按一下 **是**，然後等待 IBM MQ 停止。

c) 當 IBM MQ 停止時，用滑鼠右鍵按一下系統匣中的 **IBM MQ** 圖示，然後按一下 **結束**。

結果

您已停止要移轉至新版 IBM MQ 的佇列管理程式，並已備份佇列管理程式資料。

下一步

您現在已準備好解除安裝 IBM MQ，如 [第 117 頁的『解除安裝舊版』](#) 中所述。

相關工作

[備份佇列管理程式資料](#)

Windows 解除安裝舊版

使用控制台來解除安裝舊版。對於 Windows 上的單一階段移轉，在安裝更新版本之前解除安裝舊版產品是選用的。

開始之前

在啟動此作業之前，您必須先停止佇列管理程式，關閉 IBM MQ Explorer 並停止 IBM MQ，如 [第 115 頁的『準備移轉』](#) 中所述。

關於這項作業

在此作業中，您可以使用 Windows 控制台來解除安裝 IBM MQ。在解除安裝程序中不會移除佇列管理程式資料，這表示會保留此實務範例中使用的範例佇列管理程式，而且在您安裝較新版本的产品時可以偵測到它們。

在安裝更新版本之前，您是否必須先解除安裝舊版產品，取決於您的作業系統。在 Windows 系統上，解除安裝是選用的，您可以選擇安裝較新的版本，而不解除安裝先前的版本。請注意，在此情況下，您在安裝期間所看到的部分選項及訊息，會不同於您先解除安裝舊版時所出現的那些選項及訊息。如需在安裝較新版本之前必須在其上解除安裝較舊版本之平台的相關資訊，請參閱 [在 AIX and Linux 上移轉: 單一階段](#)。

程序

1. 按一下 **開始 > 控制台 > 解除安裝程式**，以開啟 Windows 控制台。
2. 在「**程式和特性**」視窗中，尋找您要移除的安裝項目，例如 IBM WebSphere MQ (Installation1)，然後按一下 **解除安裝**。

解除安裝程序會開始並執行至完成。當程序完成時，舊版 IBM MQ 會從您的電腦中移除，且不再顯示在程式清單中。

結果

舊版 IBM MQ 已從您的電腦中移除。不過，尚未移除佇列管理程式資料。

下一步

現在，您已準備好安裝更新版本的 IBM MQ，如 [第 117 頁的『使用啟動程式安裝 IBM MQ 9.3』](#) 中所述。

相關工作

[在 Windows 系統上解除安裝 IBM MQ](#)

Windows 使用啟動程式安裝 IBM MQ 9.3

使用安裝啟動程式及精靈，在與舊版安裝所在的相同 Windows 電腦上安裝更新版本的 IBM MQ。

開始之前

在啟動此作業之前，請下載包含安裝映像檔的壓縮檔，然後將它解壓縮至暫存目錄。

這項作業假設您先前已依照 [第 117 頁的『解除安裝舊版』](#) 中的說明，解除安裝您要移轉的舊版 IBM MQ。如果您在未先解除安裝舊版的情況下安裝更新版本，則您在安裝程序期間看到的部分選項及訊息將不同於此作業中說明的那些選項及訊息。

在啟動此作業之前，請完成下列檢查：

- 安裝時必須具有本端管理者權限。透過 Windows 機能來定義此權限。
- 請確定機器名稱不包含任何空格。
- 請確定您有足夠的磁碟空間。如需相關資訊，請參閱 [Multiplatforms 上的磁碟空間要求](#)。

在此實務範例中，不需要判斷您是否需要為任何 IBM MQ 使用者定義 Windows 網域使用者 ID，因為此需求超出此實務範例的範圍。如需相關資訊，請參閱 [為 IBM MQ 建立 Active Directory 及 DNS 網域](#)。

在安裝 IBM MQ 之前，請檢查您的系統是否符合軟體需求。如需軟體需求的最新詳細資料，請參閱 [IBM MQ 的系統需求](#)。

關於這項作業

此作業說明從舊版移轉時，在 Windows 作業系統上安裝 IBM MQ 的基本步驟。

註： IBM MQ 9.0 以及更新版本的預設程式和資料目錄位置相同。因此，從 IBM MQ 9.0 移轉至更新版本時，您不需要變更程式及資料目錄的規格。不過，從 IBM MQ 9.0 之前的版本移轉時，您需要考量預設位置的差異。如需相關資訊，請參閱 Windows 上的程式和資料目錄位置。

安裝程式包含進一步資訊的鏈結 (如果您在安裝程序期間需要的話)。安裝程序具有下列組件：

1. 使用「啟動程式」來檢查並安裝軟體需求、指定網路資訊，以及啟動 IBM MQ 安裝精靈。
2. 使用 IBM MQ 安裝精靈來安裝軟體，以及啟動 Prepare IBM MQ Wizard。
3. 使用 Prepare IBM MQ Wizard 來啟動 IBM MQ 服務。

程序

1. 啟動安裝程序。

在「Windows 檔案總管」中，導覽至已下載安裝映像檔的暫存資料夾，然後按兩下 `setup.exe`。
已啟動安裝啟動程式。

2. 執行「啟動程式」，檢閱，必要的話，修改軟體需求及網路配置。
 - a) 導覽至 IBM MQ 軟體目錄，然後按兩下檔案 `Setup.exe` 以執行「啟動程式」。
 - b) 按一下 **軟體需求** 按鈕，以顯示 **軟體需求** 標籤。
 - c) 檢查是否已符合軟體需求，以及需求項目是否顯示帶有 OK 單字的綠色勾號。進行任何指出的更正。

註：

如需任何需求的詳細資料，請按一下勾選框來展開資訊標籤。

- d) 按一下 **網路配置** 按鈕，以顯示 **網路配置** 標籤。
- e) 按一下 **否** 圓鈕。

註： 此實務範例假設您不需要配置 IBM MQ 的網域使用者 ID。如需配置 IBM MQ for Windows 網域使用者的相關資訊，請按一下 **相關資訊** 按鈕。
- f) 在「啟動程式」的 **IBM MQ 安裝** 標籤上，選取安裝語言，然後按一下 **啟動 IBM MQ 安裝程式** 以啟動 IBM MQ 安裝精靈。

您已完成透過符合或指定安裝需求來設定 IBM MQ，並已啟動 IBM MQ 安裝精靈。

3. 使用 IBM MQ 安裝精靈來安裝軟體，以及啟動 Prepare IBM MQ Wizard。
 - a) 在 IBM MQ 安裝精靈中，閱讀「授權合約」，並按一下 **我接受授權合約中的條款** 勾選框，然後按 **下一步**。
 - b) 按一下 **一般**，然後按 **下一步**。
 - c) 在「**準備安裝 IBM MQ**」頁面上，檢閱安裝資訊，然後按一下 **安裝**。

安裝資訊包括下列詳細資料：

- 安裝名稱
- 程式檔案的最上層資料夾
- 資料檔的最上層資料夾

已安裝下列特性：

- IBM MQ 伺服器
- IBM MQ: 用於管理及監視 IBM MQ 資源的圖形介面
- Java™ 和 .NET 傳訊及 Web 服務
- IBM MQ 開發工具箱

安裝程序開始。視您的系統而定，安裝程序可能需要數分鐘。

在安裝程序結束時，「IBM MQ 設定」視窗會顯示訊息 **Installation Wizard Completed Successfully**。

d) 按一下**完成**。

您已順利安裝 IBM MQ。Prepare IBM MQ Wizard 會自動啟動，並顯示 **歡迎使用 Prepare IBM MQ Wizard** 頁面。

4. 使用 Prepare IBM MQ Wizard 來啟動 IBM MQ 服務。

a) 在「歡迎使用 Prepare IBM MQ Wizard」頁面上，選取**下一步**。

Prepare IBM MQ Wizard 會顯示訊息 **Status: Checking IBM MQ Configuration** 及進度列。當程序完成時，會顯示「IBM MQ 網路配置」頁面。

b) 在 Prepare IBM MQ Wizard 的「IBM MQ 網路配置」頁面上，選取**否**。

c) 按**下一步**。

Prepare IBM MQ Wizard 會顯示訊息 **Status: starting the IBM MQ Service** 及進度列。當程序完成時，精靈會顯示「完成 Prepare IBM MQ Wizard」頁面。

d) 選取 **啟動 IBM MQ Explorer**，並選擇是否要啟動「記事本」來檢視版本注意事項，然後按一下 **完成** 按鈕。

IBM MQ Explorer 啟動。

結果

您已安裝 IBM MQ，且已啟動 IBM MQ Explorer。

下一步

現在，您已安裝較新版本的 IBM MQ，您已準備好檢查範例佇列管理程式是否已順利移轉，以及是否能夠依照第 119 頁的『[驗證 IBM MQ 9.3 安裝](#)』中的說明，將訊息放置在已移轉佇列中，以及從已移轉佇列中取得訊息。

相關概念

[在何處尋找可下載的安裝映像檔](#)

相關工作

[在 Windows 上安裝 IBM MQ 伺服器](#)

驗證 IBM MQ 9.3 安裝

安裝 IBM MQ 9.3 之後，請使用 IBM MQ Explorer 來驗證已順利從舊版移轉佇列管理程式及佇列，然後驗證您可以使用範例應用程式。

關於這項作業

當您檢查已移轉的佇列管理程式 `sampleQM` 在 IBM MQ Explorer 的「Navigator」視圖中可見時，請驗證您可以在已移轉的佇列中放置訊息，以及從已移轉的佇列中取得訊息，然後確認您仍然可以執行範例應用程式。

程序

1. 如果 IBM MQ Explorer 不在執行中，請立即啟動它。

按一下 **開始 > 所有程式 > IBM MQ > IBM MQ Explorer**。

2. 驗證您的佇列管理程式已順利移轉至較新版本的 IBM MQ:

a) 在「導覽器」視圖中，展開**佇列管理程式**資料夾。

b) 確認您可以在 **佇列管理程式** 資料夾中看到佇列管理程式 `sampleQM`。

c) 展開佇列管理程式 `sampleQM`，按一下 **佇列** 資料夾，並檢查您可以在「內容」視圖中看到佇列 `Q1`。

3. 如果佇列管理程式 `sampleQM` 尚未啟動，請立即啟動它。

- a) 在 Navigator 視圖中，展開佇列管理程式節點。
 - b) 用滑鼠右鍵按一下佇列管理程式 sampleQM，然後按一下 **啟動**。
4. 驗證您可以將訊息放入佇列 Q1。
- a) 在「瀏覽器」視圖中，展開**佇列管理程式**資料夾。
 - b) 展開佇列管理程式 sampleQM，然後按一下 **佇列** 資料夾。
 - c) 在「內容」視圖中，用滑鼠右鍵按一下佇列 Q1，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。
 - d) 在 **訊息資料** 欄位中，鍵入部分文字 (例如 Hello queue!)，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。
 - e) 按一下**關閉**。
在「內容」視圖中，請注意佇列的 **現行佇列深度** 值現在是 1。如果現行佇列深度直欄不可見，您可能需要捲動至「內容」視圖右側。
5. 驗證您可以從佇列 Q1 取得訊息。
- a) 在「Navigator」視圖中，展開 **佇列管理程式** 資料夾，
 - b) 展開佇列管理程式 sampleQM，然後按一下 **佇列** 資料夾。
 - c) 在「內容」視圖中，用滑鼠右鍵按一下佇列 Q1，然後按一下 **瀏覽訊息**。
即會開啟訊息瀏覽器，以顯示目前在佇列上的訊息清單。
 - d) 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。
6. 驗證您可以執行範例應用程式。

- a) 按兩下 runresponder.cmd 檔案。
在標示為 **回應者視窗**的命令提示字元視窗中，回應者用戶端會啟動，然後等待訊息。
> JNDI 中的 Connection Factory。> JNDI 中的目的地。> 建立與 QueueManager 的連線。> 已建立連線。
> 等待訊息。
- b) 按兩下 runrequester.cmd 檔案。
在 **要求端視窗**中，觀察要求端訊息。在「**回應者**」視窗中，觀察已更新的回應者訊息; 它所接收的訊息 (來自要求者用戶端)，以及它所傳送的回覆訊息。

在標示為 **要求端視窗**的命令提示字元視窗中，要求端用戶端會顯示連線狀態、它傳送的訊息，以及它從回應端用戶端收到的回覆訊息:

```
> JNDI 中的 Connection Factory。> JNDI 中的目的地。> 建立與 QueueManager 的連線。> 已建立連線。
> 正在傳送 'BakedBeans' 的庫存要求> 傳送訊息 ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002b02
> Received Message ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002902 for ' B
akedBeans -庫存中有 15 個 Tin '
> 正在關閉與 QueueManager 的連線。> 已關閉連線。
```

在此視窗中，觀察透過 IBM MQ 傳送的訊息:

```
-傳送的要求訊息
-收到回覆訊息
```

```
-----
備妥時，請按任何鍵以關閉此視窗
按任何鍵以繼續...
```

在「**回應者**」視窗中，觀察已更新的回應者訊息; 它所接收的訊息 (來自要求者用戶端) 及它所傳送的回覆訊息:

```
> JNDI 中的 Connection Factory。> JNDI 中的目的地。> 建立與 QueueManager 的連線。> 已建立連線。
> 等待訊息。
```

```
> Received Message ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002b02 for ' B
akedBeans '
> Sending Reply Message 'BakedBeans -15 tins in stock'> Sent Message
ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002902
> 正在關閉與 QueueManager 的連線。> 已關閉連線。
```

在此視窗中，觀察更新的回應者訊息

```
-接收的要求訊息 (來自要求者)
-傳送的回覆訊息
```

```
-----
備妥時，請按任何鍵以關閉此視窗
按任何鍵以繼續...
```


兩個指令視窗中顯示的訊息會驗證範例應用程式的要求者及回應者用戶端可以透過 IBM MQ 彼此通訊。

結果

您已順利移轉至更新版本的 IBM MQ。

Windows 選項 2: 並列移轉

此實務範例的選項 2 顯示如何在使用並列移轉方法時，從舊版 IBM MQ 移轉至新版本。透過並列移轉，您可以將 IBM MQ 的最新版本與您要從中移轉的舊版一起安裝。佇列管理程式及應用程式會繼續與舊版相關聯，直到您將它們移轉至新版本為止。

開始之前

此實務範例的起點是 [第 103 頁的『概觀: 起始 IT 配置』](#) 中說明的起始 IT 配置。

在啟動此作業之前，請遵循 [第 107 頁的『建立起始 IT 配置』](#) 中的指示來設定起始 IT 配置。

關於這項作業

當您遵循此實務範例中所說明的並列移轉方法時，您會在替代位置中安裝較新版本與較舊版本。因為您在更新版本中啟動任何佇列管理程式之前先解除安裝舊版，所以您可以將更新版本 IBM MQ 的安裝指派為主要安裝。如需主要安裝的相關資訊，請參閱 [主要安裝](#)。

程序

1. [使用啟動程式來安裝 IBM MQ 9.3](#)，然後驗證安裝。
2. [停止在舊版 IBM MQ 中執行的佇列管理程式](#)。
3. [解除安裝舊版 IBM MQ](#)。
4. [使 IBM MQ 9.3 成為主要安裝](#)。
5. 選擇性的: [建立佇列管理程式與 IBM MQ 9.3 的關聯](#)
6. 使用 IBM MQ Explorer 來 [驗證 IBM MQ 9.3 安裝](#)。

請確認已順利從舊版移轉佇列管理程式，且您可以將訊息放置在已移轉的佇列中，以及從已移轉的佇列中取得訊息。

相關工作

[在 Windows 上移轉: 並列](#)

Windows 使用啟動程式安裝 IBM MQ 9.3

使用安裝啟動程式及精靈，在 Windows 上安裝較新的 IBM MQ 版本及較舊版本。

開始之前

完成此作業之前，請先完成下列檢查：

- 安裝時必須具有本端管理者權限。透過 Windows 機能來定義此權限。
- 請確定機器名稱不包含任何空格。
- 請確定您有足夠的磁碟空間 (最多 1005 MB)，可完整安裝 IBM MQ for Windows。
- 判定是否需要為任何 IBM MQ 使用者定義 Windows 網域使用者 ID。

在安裝 IBM MQ 之前，請檢查您的系統是否符合軟硬體需求。如需所有受支援平台上軟硬體需求的最新詳細資料，請參閱 [IBM MQ 的系統需求](#)。

關於這項作業

如果您尚未在系統上安裝 IBM MQ，則此作業說明在 Windows 上安裝的基本步驟。

這項作業假設您將使用預設 IBM MQ 程式和資料檔位置。

註: IBM MQ 9.0 以及更新版本的預設程式和資料目錄位置相同。因此，從 IBM MQ 9.0 移轉至更新版本時，您不需要變更程式及資料目錄的規格。不過，從 IBM MQ 9.0 之前的版本移轉時，您需要考量預設位置的差異。如需相關資訊，請參閱 Windows 上的程式和資料目錄位置。

安裝程式包含進一步資訊的鏈結 (如果您在安裝程序期間需要的話)。安裝程序具有下列組件:

1. 使用「啟動程式」來檢查並安裝軟體需求、指定網路資訊，以及啟動 IBM MQ 安裝精靈。
2. 使用 IBM MQ 安裝精靈來安裝軟體，以及啟動 Prepare IBM MQ Wizard。
3. 使用 Prepare IBM MQ Wizard 來啟動 IBM MQ 服務。

程序

1. 啟動安裝程序。

在「Windows 檔案總管」中，導覽至已下載安裝映像檔的暫存資料夾，然後按兩下 setup.exe。

已啟動安裝啟動程式。

2. 使用啟動程式，檢閱並在必要時修改軟體需求及網路配置。

- a) 按一下 **軟體需求** 按鈕，以顯示 **軟體需求** 標籤。
- b) 檢查是否已符合軟體需求，以及需求項目是否顯示帶有 OK 單字的綠色勾號。進行任何指出的更正。

註: 如果要查看任何需求的詳細資料，請按一下加號 (+) 按鈕。

- c) 按一下 **網路配置** 按鈕，以顯示 **網路配置** 標籤。
- d) 按一下 **否** 圓鈕。

註: 此實務範例假設您不需要配置 IBM MQ 的網域使用者 ID。如需配置 IBM MQ for Windows 網域使用者的相關資訊，請按一下 **相關資訊** 按鈕。

- e) 在「啟動程式」的 **IBM MQ 安裝** 標籤上，選取安裝語言，然後按一下 **啟動 IBM MQ 安裝程式** 按鈕以啟動 IBM MQ 安裝精靈。

您已完成 IBM MQ 安裝需求的檢閱，已進行任何必要的修改，並已啟動 IBM MQ 安裝精靈。

3. 使用 IBM MQ 安裝精靈來安裝軟體，以及啟動 Prepare IBM MQ Wizard。

IBM MQ 安裝精靈會檢查現有安裝，並顯示可供您使用的升級或安裝選項。在此情況下，有兩個選項:

- 在現有安裝保持不變的情況下安裝
- 升級 8.0.0.5 安裝「安裝 1」

- a) 選取 **安裝讓現有的安裝維持不變**，然後按 **下一步**。
- b) 閱讀「授權合約」，並按一下 **我接受授權合約中的條款** 勾選框，然後按 **下一步**。
- c) 按 **一般**，然後按 **下一步**。
- d) 在 **準備安裝 IBM MQ** 頁面上，檢閱顯示的安裝資訊，然後按一下 **安裝**。

安裝資訊包括下列詳細資料:

- 安裝名稱
- 程式檔案的最上層資料夾
- 資料檔的最上層資料夾

已安裝下列特性:

- IBM MQ 伺服器
- IBM MQ: 用於管理及監視 IBM MQ 資源的圖形介面
- Java™ 和 .NET 傳訊及 Web 服務
- IBM MQ 開發工具箱

安裝程序開始。視您的系統而定，安裝程序可能需要數分鐘。

在安裝程序結束時，「IBM MQ 設定」視窗會顯示訊息 Installation Wizard Completed Successfully。

e) 按一下**完成**。

您已順利安裝 IBM MQ。Prepare IBM MQ Wizard 會自動啟動，並顯示 **歡迎使用 Prepare IBM MQ Wizard** 頁面。

4. 使用 Prepare IBM MQ Wizard 來啟動 IBM MQ 服務。

a) 在「歡迎使用 Prepare IBM MQ Wizard」頁面上，選取**下一步**。

Prepare IBM MQ Wizard 會顯示訊息 Status: Checking IBM MQ Configuration 及進度列。當程序完成時，會顯示「IBM MQ 網路配置」頁面。

b) 在 Prepare IBM MQ Wizard 的「IBM MQ 網路配置」頁面上，選取**否**。

c) 按**下一步**。

Prepare IBM MQ Wizard 會顯示訊息 Status: starting the IBM MQ Service 及進度列。當程序完成時，精靈會顯示「完成 Prepare IBM MQ Wizard」頁面。

d) 選取 **啟動 IBM MQ Explorer**，並選擇是否要啟動「記事本」來檢視版本注意事項，然後按一下 **完成** 按鈕。

IBM MQ Explorer 啟動。

下一步

您已隨舊版一起安裝 IBM MQ 的較新版本，但位於不同的安裝目錄中，並且已啟動 IBM MQ Explorer。

現在，您已準備好停止在舊版 IBM MQ 中執行的佇列管理程式，如 [第 123 頁的『停止佇列管理程式』](#) 中所述。

Windows 停止佇列管理程式

在移轉至更新版本的 IBM MQ 之前，您必須先停止佇列管理程式並備份佇列管理程式資料。

關於這項作業

在取得備份之前，請停止您要備份的佇列管理程式。如果您嘗試建立執行中佇列管理程式的備份，備份可能不一致，因為在複製檔案時正在進行任何更新。

程序

1. 開啟 IBM MQ Explorer。

按一下 **啟動 > 所有應用程式 > IBM MQ > IBM MQ Explorer**。

2. 停止佇列管理程式 sampleQM。

a) 在「Navigator」視圖中，用滑鼠右鍵按一下佇列管理程式 sampleQM。

b) 按一下**停止**。

即會開啟「**結束佇列管理程式**」視窗。

c) 選取 **控制項**，然後按一下 **確定**。

選取 **受控制** 選項會以受控制且有序的方式停止佇列管理程式。通常只有在受控制的停止無法順利完成時，才會使用強制佇列管理程式停止的 **立即** 選項。

佇列管理程式停止。在 IBM MQ 中，佇列管理程式 sampleQM 旁的圖示會變更為包含向下指向的紅色箭頭。

3. 關閉 IBM MQ Explorer。

4. 備份佇列管理程式資料。

取得下列所有資料的副本，確定您包括所有備份目錄。部分目錄可能是空的，但如果您需要在稍後日期還原備份，則全部都需要，因此也請儲存它們。

- 位於 C:\ProgramData\IBM\MQ\Qmgrs 中的佇列管理程式資料。

- 位於 C:\ProgramData\IBM\MQ\log 中之佇列管理程式的日誌檔目錄，包括日誌控制檔 amqhlctl.lfh。
- 位於 C:\ProgramData\IBM\MQ\Config 中的配置檔。
- IBM MQ 9.2 .ini 檔案和登錄項目。佇列管理程式資訊儲存在 .ini 檔中，可用來回復成舊版產品。

5. 停止 IBM MQ。

a) 停止 IBM MQ 服務。

用滑鼠右鍵按一下系統匣中的 **IBM MQ** 圖示，然後按一下 **停止 IBM MQ**。

即會顯示具有下列訊息的對話框：

關閉 IBM MQ 會終止所有執行中的佇列管理程式，並關閉 IBM MQ 程序。您確定要繼續嗎？(AMQ4102)

b) 按一下 **是**，然後等待 IBM MQ 停止。

c) 當 IBM MQ 停止時，用滑鼠右鍵按一下系統匣中的 **IBM MQ** 圖示，然後按一下 **結束**。

下一步

停止佇列管理程式之後，您就可以依照第 125 頁的『將佇列管理程式與 IBM MQ 9.3 相關聯』中的說明，將它們與新版 IBM MQ 的新安裝架構相關聯。

Windows 解除安裝舊版

使用 Windows 控制台來解除安裝舊版產品。

開始之前

在啟動此作業之前，您必須先停止佇列管理程式，關閉 IBM MQ Explorer 並停止 IBM MQ，如第 123 頁的『停止佇列管理程式』中所述。

關於這項作業

在此作業中，您可以使用 Windows 控制台來解除安裝 IBM MQ。在解除安裝程序中不會移除佇列管理程式資料，這表示會保留此實務範例中使用的範例佇列管理程式，而且在您安裝較新版本的產品時可以偵測到它們。

程序

1. 按一下 **開始 > 控制台 > 解除安裝程式**，以開啟 Windows 控制台。
2. 在「**程式和特性**」視窗中，尋找您要移除的安裝項目，例如 IBM WebSphere MQ (Installation1)，然後按一下 **解除安裝**。

解除安裝程序會開始並執行至完成。當程序完成時，舊版 IBM MQ 會從您的電腦中移除，且不再顯示在程式清單中。

結果

舊版產品已從您的電腦中移除。不過，尚未移除佇列管理程式資料。

下一步

現在，您已準備好讓 IBM MQ 的較新版本成為主要安裝架構，如第 124 頁的『使 IBM MQ 9.3 成為主要安裝』中所述。

相關工作

[在 Windows 系統上解除安裝 IBM MQ](#)

Windows 使 IBM MQ 9.3 成為主要安裝

在較新 IBM MQ 版本的新安裝中啟動任何佇列管理程式之前，您可以選擇性地將較新版本設為主要安裝。

關於這項作業

在支援多個 IBM MQ 安裝的系統上，主要安裝是 IBM MQ 系統範圍位置所參照的安裝。具有主要安裝是選用的，但非常方便。

當您遵循此實務中所說明的並列移轉方法時，因為在更新版本中啟動任何佇列管理程式之前，您會先解除安裝舊版，您可以將產品更新版本的安裝指派為主要安裝。

如需主要安裝的相關資訊，請參閱 [主要安裝](#)。

程序

1. 在命令提示字元中輸入 **dspmqinst** 指令，以檢查現行主要安裝。
命令提示字元會顯示任何現行安裝的詳細資料。現行主要安裝具有下列行 **Primary: Yes**。
2. 使用 **setmqinst** 指令來變更現行主要安裝架構。
在命令提示字元中，輸入：

```
setmqinst -x -n Installation_Name
```

其中 *Installation_Name*> 是現行主要安裝架構的名稱。

如果指令成功，則命令提示字元會顯示 '*Installation_Name*' (*Filepath*) has been unset as the Primary Installation 訊息。

3. 使用 **setmqinst** 指令，將新的 IBM MQ 9.3 安裝設為主要安裝。
在命令提示字元中，輸入：

```
setmqinst -i -n V9_Installation
```

其中 *V9_Installation* 是 IBM MQ 9.3 安裝架構的名稱。

如果指令成功，則命令提示字元會顯示訊息 '*V9_Installation*' (*Filepath*) has been set as the primary installation. You must restart the operating system to complete the update.

註: 如成功訊息中所指示，您必須重新啟動作業系統才能完成更新。

下一步

您已準備好將已移轉的佇列管理程式與更新版本的 IBM MQ 相關聯，如 [第 125 頁的『將佇列管理程式與 IBM MQ 9.3 相關聯』](#) 中所述。

將佇列管理程式與 IBM MQ 9.3 相關聯

使用「傳送佇列管理程式」精靈，將 sampleQM 佇列管理程式與您安裝的較新 IBM MQ 版本相關聯。

開始之前

在啟動這項作業之前，請確定您已依照 [第 123 頁的『停止佇列管理程式』](#) 中的說明來停止佇列管理程式，否則您將無法完成傳送。

關於這項作業

IBM MQ Explorer 的「傳送佇列管理程式」精靈特性可讓您將一或多個佇列管理程式從其他安裝架構傳送至現行安裝架構。此精靈相當於執行 **setmqm** 指令，但是不必鍵入必要的路徑及參數。只有已停止的佇列管理程式才能傳送；顯示的執行中佇列管理程式僅供參考。

在產品更新版本的安裝上傳送並啟動佇列管理程式之後，就無法移轉回舊版。

程序

1. 啟動 IBM MQ Explorer。
按一下 **啟動 > 所有應用程式 > IBM MQ > IBM MQ Explorer**。
2. 在「Navigator」視圖中，用滑鼠右鍵按一下佇列管理程式節點，然後選取 **傳送佇列管理程式**。
3. 按一下滑鼠右鍵選取佇列管理程式 sampleQM，然後按一下 **傳送**。
會以選取的佇列管理程式來呼叫 **setmqm** 指令。如果傳送成功，瀏覽器樹狀結構會更新以包含已傳送的佇列管理程式。如果發生任何問題，則會出現對話框，顯示指令所提供的錯誤訊息。
4. 啟動佇列管理程式 sampleQM。
 - a) 在 Navigator 視圖中，展開佇列管理程式節點。
 - b) 用滑鼠右鍵按一下佇列管理程式的名稱，然後按一下 **啟動**。

結果

您已順利將佇列管理程式 sampleQM 與更新版本的 IBM MQ 相關聯。

下一步

驗證已順利移轉佇列管理程式 sampleQM，方法是確認您可以依照 [第 126 頁的『驗證 IBM MQ 9.3 安裝』](#) 中的說明，將訊息放在佇列中，並從佇列中取得訊息。

驗證 IBM MQ 9.3 安裝

安裝較新版本的 IBM MQ 之後，請使用 IBM MQ Explorer 來驗證佇列管理程式及佇列已順利從舊版移轉，然後驗證您可以使用範例應用程式。

關於這項作業

當您檢查已移轉的佇列管理程式 sampleQM 在 IBM MQ Explorer 的「Navigator」視圖中可見時，請驗證您可以在已移轉的佇列中放置訊息，以及從已移轉的佇列中取得訊息，然後確認您仍然可以執行範例應用程式。

程序

1. 如果 IBM MQ Explorer 不在執行中，請立即啟動它。
按一下 **開始 > 所有程式 > IBM MQ > IBM MQ Explorer**。
2. 驗證您的佇列管理程式已順利移轉至較新版本的 IBM MQ：
 - a) 在「瀏覽器」視圖中，展開 **佇列管理程式** 資料夾。
 - b) 確認您可以在 **佇列管理程式** 資料夾中看到佇列管理程式 sampleQM。
 - c) 展開佇列管理程式 sampleQM，按一下 **佇列** 資料夾，並檢查您可以在「內容」視圖中看到佇列 Q1。
3. 如果佇列管理程式 sampleQM 尚未啟動，請立即啟動它。
 - a) 在 Navigator 視圖中，展開佇列管理程式節點。
 - b) 用滑鼠右鍵按一下佇列管理程式 sampleQM，然後按一下 **啟動**。
4. 驗證您可以將訊息放入佇列 Q1。
 - a) 在「瀏覽器」視圖中，展開 **佇列管理程式** 資料夾。
 - b) 展開佇列管理程式 sampleQM，然後按一下 **佇列** 資料夾。
 - c) 在「內容」視圖中，用滑鼠右鍵按一下佇列 Q1，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。
 - d) 在 **訊息資料** 欄位中，鍵入部分文字 (例如 Hello queue!)，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。
 - e) 按一下 **關閉**。

在「內容」視圖中，請注意佇列的**現行佇列深度**值現在是 1。如果現行佇列深度直欄不可見，您可能需要捲動至「內容」視圖右側。

5. 驗證您可以從佇列 Q1 取得訊息。

- a) 在「Navigator」視圖中，展開**佇列管理程式**資料夾，
- b) 展開佇列管理程式 sampleQM，然後按一下**佇列**資料夾。
- c) 在「內容」視圖中，用滑鼠右鍵按一下佇列 Q1，然後按一下**瀏覽訊息**。即會開啟訊息瀏覽器，以顯示目前在佇列上的訊息清單。
- d) 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。

6. 驗證您可以執行範例應用程式。

- a) 按兩下 runresponder.cmd 檔案。
在標示為**回應者視窗**的命令提示字元視窗中，回應者用戶端會啟動，然後等待訊息。
> JNDI 中的 Connection Factory。> JNDI 中的目的地。> 建立與 QueueManager 的連線。> 已建立連線。
> 等待訊息。
- b) 按兩下 runrequester.cmd 檔案。
在**要求端視窗**中，觀察要求端訊息。在「**回應者**」視窗中，觀察已更新的回應者訊息；它所接收的訊息(來自要求者用戶端)，以及它所傳送的回覆訊息。

在標示為**要求端視窗**的命令提示字元視窗中，要求端用戶端會顯示連線狀態、它傳送的訊息，以及它從回應端用戶端收到的回覆訊息：

```
> JNDI 中的 Connection Factory。> JNDI 中的目的地。> 建立與 QueueManager 的連線。> 已建立連線。  
> 正在傳送 'BakedBeans' 的庫存要求> 傳送訊息 ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002b02  
> Received Message ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002902 for ' B  
akedBeans -庫存中有 15 個 Tin '  
> 正在關閉與 QueueManager 的連線。> 已關閉連線。
```

在此視窗中，觀察透過 IBM MQ 傳送的訊息：

```
-傳送的要求訊息  
-收到回覆訊息
```

```
備妥時，請按任何鍵以關閉此視窗  
按任何鍵以繼續...
```

在「**回應者**」視窗中，觀察已更新的回應者訊息；它所接收的訊息(來自要求者用戶端)及它所傳送的回覆訊息：

```
> JNDI 中的 Connection Factory。> JNDI 中的目的地。> 建立與 QueueManager 的連線。> 已建立連線。  
> 等待訊息。
```

```
> Received Message ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002b02 for ' B  
akedBeans '  
> Sending Reply Message 'BakedBeans -15 tins in stock'> Sent Message  
ID=ID:414d5120514d5f4c33344c3238482020c3cd094d20002902  
> 正在關閉與 QueueManager 的連線。> 已關閉連線。
```

在此視窗中，觀察更新的回應者訊息

```
-接收的要求訊息 (來自要求者)  
-傳送的回覆訊息
```

```
備妥時，請按任何鍵以關閉此視窗  
按任何鍵以繼續...
```

兩個指令視窗中顯示的訊息會驗證範例應用程式的要求者及回應者用戶端可以透過 IBM MQ 彼此通訊。

結果

您已順利移轉至更新版本的 IBM MQ。

下一步

如需移轉及升級的相關資訊，請參閱 [維護及移轉](#)。

在 Windows 上安裝更新版本的 IBM MQ 以與舊版共存

此實務範例顯示安裝 IBM MQ Long Term Support (LTS) 版本的所有步驟，同時共存（「並列」）舊版本的產品。此外，這些步驟包括將修正套件安裝至更新版本。您應該一律安裝最新的 Fix Pack 層次。

多個安裝的概觀

此實務範例中使用的多個安裝以及軟硬體的說明。

關於這項作業

在相同主機中多個 IBM MQ 安裝 (多重安裝) 的這項特性的一個重要方面是，這種類型的安裝不需要在使用較新產品版本執行活動時停止舊版產品上的佇列管理程式和應用程式。

也就是說，產品更新版本的安裝不會影響在舊版產品上執行應用程式。當您嘗試將舊版產品上的佇列管理程式多階段移轉至更新版本時，這會很有用。

在此實務範例中，IBM MQ 8.0.0 用作產品的較舊版本，而 IBM MQ 9.1.0 用作產品的較新版本。

使用 IBM MQ 的 Long Term Support 版本

此實務範例將 IBM MQ 的 LTS 版本用於產品的較新版本。

CD 如果您將 Continuous Delivery (CD) 版本的 IBM MQ 用於產品更新版本，則必須先解除安裝您正在使用的 CD 版本 (例如 IBM MQ 9.1.1)，然後再安裝更新版本 (例如 IBM MQ 9.1.5)。如需相關資訊，請參閱 [從一個 Continuous Delivery 版本移轉至另一個](#)。

用於此實務範例的軟硬體

作業系統

Windows 10

主機名稱: johndoe1.fyre.<yourdomainname>.com

佇列管理程式

QM80

使用 IBM MQ 8.0.0 建立; 保留在 IBM MQ 8.0.0

QMMIG

使用 IBM MQ 8.0.0 建立; 以移轉至 IBM MQ 9.1.0

QM910

使用 IBM MQ 9.1.0 建立; 保留在 IBM MQ 9.1.0

將較新版本的 IBM MQ 並列安裝至較舊版本。

如何在相同機器上並列安裝 IBM MQ 9.3 至產品的現有版本。IBM MQ 9.1 安裝將不會指定為主要安裝。

開始之前

確保系統上已安裝 IBM MQ 9.1。如果未安裝 IBM MQ 9.1，請使用下列指示 [在 Windows 上安裝 IBM MQ 伺服器來安裝產品](#)。

您必須選取自訂選項，才能明確選取「用戶端」。如需相關資訊，請參閱 [Windows 的安裝方法](#)。

此外，您還需要使用 `-n` 參數來呼叫 `setmqenv` 指令，以將安裝名稱設為 `Installation1`。如果您使用批次檔來呼叫指令，它會很有用。

關於這項作業

所有佇列管理程式資料都儲存在共用目錄結構中。即使每一個 IBM MQ 版本的執行碼都儲存在不同的目錄結構中，所有佇列管理程式的資料 (不論版本為何) 都會儲存在 :MQ_DATA_PATH=C:\ProgramData\IBM\MQ 中。

若要安裝 IBM MQ 9.3:

程序

1. 以管理者身分登入。

IBM MQ 9.3 將安裝在預設目錄 C:\Program Files\IBM\MQ 中。

2. 移至下載檔案所在的目錄，例如 C:\downloads\mq9300。
3. 解壓縮下載的檔案。

這些檔案會解壓縮至稱為 MQServer 的新子目錄。

4. 切換至新目錄，並發出指令 **setup.exe** 以啟動安裝程式。

- a) 按一下軟體需求，以檢查您的企業是否已安裝必備軟體。

如需相關資訊，請參閱 [檢查 Windows 上的需求](#)。在此實務範例中，系統具有必要需求。

- b) 按一下網路配置。

在此實務範例中，機器不是「網域」的一部分，因此不需要指出網域使用者，問題的答案是 No。

- c) 按一下 *IBM MQ* 安裝。

- d) 按一下啟動 *IBM MQ* 安裝程式。

安裝程式會偵測到系統上有其他安裝架構，並顯示下列訊息：

升級或安裝

升級現有安裝或隨它一起安裝新版本

在此實務範例中，請保持其他安裝完整，因此選取第一個項目 安裝使現有安裝保持不變。

- e) 按 **下一步** 並接受授權。

- f) 針對安裝選項選取 **自訂**。

如需相關資訊，請參閱 [Windows 的安裝方法](#)。

- g) 按 **下一步**。

在

安裝詳細資料

定義安裝詳細資料

您會看到：

安裝名稱

Installation2

程式檔案的安裝資料夾

C:\ProgramFiles\IBM\MQ

顯示要安裝的特性清單。請注意，部分項目：

- 依預設未安裝
- 如果您要安裝它們，則需要選取它們。

對於此實務範例，請選取 *MQI* 用戶端，而且您無論如何都應該選取此選項。

- h) 按 **下一步**。

在標題為 準備安裝 *IBM MQ* 的畫面中，您會看到「安裝名稱」、位置、要安裝的元件等的摘要。

- i) 按一下 **安裝** 以繼續進行。

開始將檔案複製到安裝目錄結構。複製檔案之後，會出現下列對話框: The IBM MQ Installation Wizard has successfully installed IBM MQ。

- j) 按一下 **完成**。

安裝之後，您會看到下列對話框: Welcome to the Prepare IBM MQ Wizard。

因為此實務範例未使用 Windows 網域，所以接受預設值 否。

接受啟動 IBM MQ Explorer，除非您有非常明確的原因，否則您可以接受 使用新的工作區啟動 MQ 探險家的預設值。

結果

您已順利安裝 IBM MQ for Windows 的另一個版本以及產品的現有版本。

下一步

您需要執行 **setmqenv** 指令，才能在任一版本上使用指令。請參閱第 130 頁的『使用 setmqenv 指令來執行 IBM MQ 的兩個版本』，以取得詳細資料。

使用 setmqenv 指令來執行 IBM MQ 的兩個版本

安裝活動已完成，您現在可以驗證包含 IBM MQ 程式碼的目錄。

開始之前

請確定您已在 C:\Program Files\IBM\WebSphere MQ\ 中正確安裝 IBM MQ 8.0.0 產品 Installation1，並在 C:\Program Files\IBM\MQ 中正確安裝 IBM MQ 9.1 產品 Installation2。

關於這項作業

使用 **dspmqlnst** 指令來顯示系統中所安裝版本的安裝資訊，而不需要查看目錄，並使用 **dspmqver** 指令來顯示版本資訊。

在 Windows 系統上，資訊保留在登錄中，安裝配置資訊保留在下列機碼中：

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\IBM\WebSphere MQ\Installation。

重要: 您不得直接編輯或參照此金鑰。

程序

1. 使用 **dspmqlnst** 指令顯示安裝資訊。

```
InstName:      Installation1
InstDesc:
Identifier:    1
InstPath:     C:\Program Files\IBM\WebSphere MQ
Version:      8.0.0.9
Primary:      Yes
State:        Available
MSIProdCode:  {74F6B169-7CE6-4EFB-8A03-2AA7B2DBB57C}
MSIMedia:     8.0 Server
MSIInstanceId: 1

InstName:      Installation2
InstDesc:
Identifier:    2
InstPath:     C:\Program Files\IBM\MQ
Version:      9.1.0.0
Primary:      No
State:        Available
MSIProdCode:  {5D3ECA81-BF8D-4E80-B36C-CBB1D69BC110}
MSIMedia:     9.1 Server
MSIInstanceId: 1
```

註: 個別安裝名稱 (InstName) 很重要。

2. 針對預設 (或主要) 安裝，顯示使用 **dspmqver** 的 IBM MQ 版本:

```
C:\> dspmqver
Name:      WebSphere MQ
Version:   8.0.0.9
Level:    p800-009-180321.1
```

```
BuildType: IKAP - (Production)
Platform: WebSphere MQ for Windows (x64 platform)
Mode: 64-bit
O/S: Windows 10 Professional x64 Edition, Build 18363
InstName: Installation1
InstDesc:
Primary: Yes
InstPath: C:\Program Files\IBM\WebSphere MQ
DataPath: C:\ProgramData\IBM\MQ
MaxCmdLevel: 802
LicenseType: Production
```

執行指令之後，您會收到下列訊息: Note there are a number (1) of other installations, use the '-i' parameter to display them.

3. 發出指令 C:\> where dspmqver, 您會看到主要安裝架構的相關資訊:

```
C:\> where dspmqver
C:\Program Files\IBM\WebSphere MQ\bin64\dspmqver.exe
C:\Program Files\IBM\WebSphere MQ\bin\dspmqver.exe
```

4. 若要查看 IBM MQ 9.1 產品的相關資訊，請發出下列指令: C:\> setmqenv -n Installation2。
5. 重新發出指令 C:\> where dspmqver, 您會看到第二個安裝的相關資訊:

```
C:\> where dspmqver
C:\Program Files\IBM\MQ\bin64\dspmqver.exe
C:\Program Files\IBM\MQ\bin\dspmqver.exe
```

6. 重新發出指令 C:\> dspmqver。

您現在可以看到:

```
C:\> dspmqver
Name: IBM MQ
Version: 9.1.0.0
Level: p910-L180705
BuildType: IKAP - (Production)
Platform: IBM MQ for Windows (x64 platform)
Mode: 64-bit
O/S: Windows 10 Professional x64 Edition, Build 18363
InstName: Installation2
InstDesc:
Primary: No
InstPath: C:\Program Files\IBM\MQ
DataPath: C:\ProgramData\IBM\MQ
MaxCmdLevel: 910
LicenseType: Production
```

7. 發出指令 C:\> set MQ, 並且在使用 setmqenv 之後，您會看到第二個安裝的相關資訊。

```
C:\> set MQ
MQ_DATA_PATH=C:\ProgramData\IBM\MQ
MQ_ENV_MODE=64
MQ_FILE_PATH=C:\Program Files\IBM\MQ
MQ_INSTALLATION_NAME=Installation2
MQ_INSTALLATION_PATH=C:\Program Files\IBM\MQ
MQ_JAVA_DATA_PATH=C:\ProgramData\IBM\MQ
MQ_JAVA_INSTALL_PATH=C:\Program Files\IBM\MQ\java
MQ_JAVA_LIB_PATH=C:\Program Files\IBM\MQ\java\lib64
MQ_JRE_PATH=C:\Program Files\IBM\MQ\java\jre
```

您可以建立批次檔，以使用指定的語法來執行 setmqenv 指令。請確定您在 PATH 中的目錄中具有此批次檔，例如 C:\WinTools。

例如，您可以建立含有下列內容的批次檔 set-mq-910.bat：

```
REM Setup the environment to run MQ 9.1
CALL "C:\Program Files\IBM\MQ\bin\setmqenv" -n Installation2
REM Adding Samples to the path
SET PATH=%PATH%;%MQ_FILE_PATH%\Tools\c\Samples\Bin;%MQ_FILE_PATH%\Tools\c\Samples\Bin64
;%MQ_FILE_PATH%\Tools\jms\samples;%MQ_JAVA_INSTALL_PATH%\bin\ dspmqver -f 2
```

附註:

- a. 呼叫 **setmqenv** 時，您需要使用 "CALL" 引數。如果沒有這個引數，**setmqenv** 的處理程序會結束批次，且不容許執行下列陳述式。也就是說，當使用 CALL 引數時，您容許處理批次檔中的其他陳述式。
- b. 如果您在 PATH 中新增 IBM MQ 目錄，例如 C-samples 的位置 :PATH= ...;C:\Program Files\IBM\MQ\tools\c\Samples\Bin; ... 此目錄將由 **setmqenv** 在下次執行指令時移除。
如果您想要能夠從 IBM MQ 9.1 執行 C-samples，則需要前一個批次檔中的最後一行，才能將範例的目錄放回 PATH 中。
另請注意，使用 MQ_FILE_PATH 是為 IBM MQ 9.1 使用適當的目錄結構: SET PATH=%PATH%;%MQ_FILE_PATH%\tools\c\Samples\Bin。

建立佇列管理程式

如何使用 **crtmqm** 指令來建立佇列管理程式。您可以使用 IBM MQ Explorer 來執行這項作業。

開始之前

將 **dspmqs** 指令與 **-o** 安裝及 **-s** 參數搭配使用，以顯示現行佇列管理程式的安裝名稱及狀態。

```
C:\> dspmqs -o installation -s
QMNAME(QM80)                STATUS(Running) INSTNAME(Installation1)
INSTPATH(C:\Program Files\IBM\WebSphere MQ) INSTVER(9.1.0.9)
QMNAME(QMMIG)              STATUS(Running) INSTNAME(Installation1)
INSTPATH(C:\Program Files\IBM\WebSphere MQ) INSTVER(9.1.0.9)
```

關於這項作業

您必須開啟 Windows 命令提示字元，並將您自己設為 管理者，以執行下列處理程序。如果您在未使用命令提示字元的情況下嘗試發出 **crtmqm** 指令，則會收到訊息 AMQ7077: 您未獲授權執行所要求的作業。

程序

1. 選取 **開始 > Windows 系統 > 命令提示字元 > 更多 > 以管理者身分執行**
所建立視窗的標題為 管理者: 命令提示字元。
註: 命令提示字元中的 IBM MQ 版本是 IBM MQ 9.1。
2. 執行 **setmqenv** 指令或您可能已建立的批次檔 **set-mq-930**。
如需詳細資料，請參閱 第 130 頁的『使用 setmqenv 指令來執行 IBM MQ 的兩個版本』。
在任一情況下，您都會看到 9.3.0.0 版
3. 發出下列指令: C:\> **crtmqm -u SYSTEM.DEAD.LETTER.QUEUE QM930**
您將看到下列輸出:

```
IBM MQ queue manager created.
Directory 'C:\ProgramData\IBM\MQ\qmgrs\QM930' created.
The queue manager is associated with installation 'Installation2'.
Creating or replacing default objects for queue manager 'QM930'.
Default objects statistics : 87 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
```

4. 發出下列指令以啟動佇列管理程式 C:\> **strmqm QM930**
您會看到下列輸出。請注意指出佇列管理程式執行所在的安裝及版本的字行:

```
IBM MQ queue manager 'QM930' starting.
The queue manager is associated with installation 'Installation2'.
5 log records accessed on queue manager 'QM930' during the log replay phase.
Log replay for queue manager 'QM930' complete.
Transaction manager state recovered for queue manager 'QM910'.
IBM MQ queue manager 'QM930' started using V9.3.0.0.
```

5. 再次發出指令 C:\> **dspmqs -o installation -s**，以顯示已安裝的佇列管理程式:

```
C:\> dspmq -o installation -s
QMNAME(QM80) STATUS(Running) INSTNAME(Installation1)
INSTPATH(C:\Program Files\IBM\WebSphere MQ) INSTVER(9.1.0.9)
QMNAME(QMMIG) STATUS(Running) INSTNAME(Installation1)
INSTPATH(C:\Program Files\IBM\WebSphere MQ) INSTVER(9.1.0.9)
QMNAME(QM910) STATUS(Running) INSTNAME(Installation2)
INSTPATH(C:\Program Files\IBM\MQ) INSTVER(9.3.0.0)
++ Cannot use MQ 9.3.0 administrative commands to run an MQ 9.1 queue manager
```

重要: 您無法在與您所使用的版本不同的 IBM MQ 版本上使用管理指令。

如果您嘗試這樣做，則會收到訊息 AMQ5691E: Queue manager <qmname> is associated with a different installation (<installation name>)。

將佇列管理程式移轉至更新版本的 IBM MQ

安裝 IBM MQ 9.3 之後，您現在想要移轉及升級 IBM MQ 9.1 佇列管理程式，以與 IBM MQ 9.3 搭配使用。

關於這項作業

您需要執行兩個主要步驟：

1. 使用 **setmqm** 指令，將佇列管理程式與想要的安裝 (即 IBM MQ 版本) 相關聯。
2. 在想要的版本下使用 **strmqm** 指令，這會更新新版本的佇列管理程式資料。

在此實務範例中，使用 IBM MQ 9.1 建立的佇列管理程式 QMMIG 將用作移轉程序的圖解。



小心: 一旦佇列管理程式已移轉至較新版本的 IBM MQ，就無法再將它與舊版 IBM MQ 搭配使用。移轉處理程序會變更一些檔案及物件定義，且無法返回。

程序

1. 使用 **dmpmqcfig** 指令來備份佇列管理程式。

如需相關資訊，請參閱 [備份及還原 IBM MQ 佇列管理程式資料](#) 及 [備份佇列管理程式配置](#)。

- a) 若要指定所有屬性 (包括預設屬性) (**setmqaut** 除外，其未包含在輸出中)，請指定下列指令：

```
dmpmqcfig -m QMgr -a > QMgr.dmpmqcfig.out.all.mqsc
```

- b) 若要以 **setmqaut** 格式擷取屬性，請發出下列指令：

```
dmpmqcfig -m QMgr -o setmqaut > QMgr.dmpmqcfig.setmqaut.bat
```

註: 包含 **setmqaut** 指令的輸出檔包括每一個指令中佇列管理程式的名稱。因此，如果您要將指令還原至不同的佇列管理程式，則需要編輯檔案並指定所需的佇列管理程式名稱。

2. 若要還原，請執行下列動作：

- a) **runmqsc** 的指令，發出：

```
runmqsc Qmgr < QMgr.dmpmqcfig.out.mqsc
```

or

```
runmqsc Qmgr < QMgr.dmpmqcfig.out.all.mqsc
```

- b) **setmqaut** 指令發出：

```
QMgr.dmpmqcfig.setmqaut.bat
```

3. 要移轉的佇列管理程式位於 IBM MQ 9.1，因此您需要執行 Script，將執行環境設為 IBM MQ 9.3：

```
C:\> set-mq-80
```

- a) 發出指令 C:\> **dspmqver**，以檢查佇列管理程式在 IBM MQ 9.1 上執行的版本。
- b) 發出指令 C:\> **where dspmq**，以檢查佇列管理程式是否在執行中：

```
C:\Program Files\IBM\WebSphere MQ\bin64\dspmq.exe
C:\Program Files\IBM\WebSphere MQ\bin\dspmq.exe
```

- c) 發出以下指令: C:\> dspmq -m QMMIG -o installation -s

```
QMNAME(QMMIG)          STATUS(Running) INSTNAME(Installation1)
INSTPATH(C:\Program Files\IBM\WebSphere MQ) INSTVER(9.1.0.9)
```

- d) 發出以下指令: C:\> runmqsc QMMIG

```
display qmgr cmdlevel version
  1 : display qmgr cmdlevel
AMQ8408: Display Queue Manager details.
QMNAME(QMMIG)          CMDLEVEL(910)
VERSION(09001009)
end
```

請注意, CMDLEVEL 位於 IBM MQ 9.1。

- e) 發出指令 C:\ endmqm -i QMMIG 來停止佇列管理程式
您會收到下列訊息:

```
WebSphere MQ queue manager 'QMMIG' ending.
WebSphere MQ queue manager 'QMMIG' ended.
```

4. 變更環境以執行 IBM MQ 9.3 指令, 方法是發出指令 C:\> set-mq-930 (如果您已建立批次檔), 或使用 setmqenv 指令, 然後發出 dspmqver 指令來檢查版本。
5. 發出指令 C:\> dspmq -o installation -s, 以顯示佇列管理程式的狀態。
您會收到下列輸出:

```
QMNAME(QM91)          STATUS(Running)INSTNAME(Installation1)
INSTPATH(C:\Program Files\IBM\WebSphere MQ) INSTVER(9.1.0.9)

QMNAME(QMMIG)          STATUS(Ended immediately) INSTNAME(Installation1)
INSTPATH(C:\Program Files\IBM\WebSphere MQ) INSTVER(9.1.0.9)

QMNAME(QM910)          STATUS(Running) INSTNAME(Installation2)
INSTPATH(C:\Program Files\IBM\MQ) INSTVER(9.3.0.0)
```



小心: 當 Installation2 與 IBM MQ 9.3 相關聯時, 佇列管理程式 QMMIG 仍與 Installation1 (IBM MQ 9.1) 相關聯。

因此, 您需要取消佇列管理程式 QMMIG 與 Installation1 的關聯, 並將它與 Installation2 相關聯

6. 發出下列指令, 將佇列管理程式 QMMIG 與 Installation2 相關聯

```
C:\> setmqm -m QMMIG -n Installation2
```

您會收到下列訊息:

```
The setmqm command completed successfully.
```

這會通知您 QMMIG 現在與 IBM MQ 9.3 相關聯。

7. 發出指令 C:\> strmqm QMMIG 來啟動佇列管理程式 QMMIG

因為這是第一次在先前與舊版搭配使用的佇列管理程式上發出 IBM MQ 9.3 strmqm 指令, 所以會進行移轉。

您會看到類似下列的輸出:

```
IBM MQ queue manager 'QMMIG' starting.
The queue manager is associated with installation 'Installation2'.
5 log records accessed on queue manager 'QMMIG' during the log replay phase.
Log replay for queue manager 'QMMIG' complete.
Transaction manager state recovered for queue manager 'QMMIG'.
Migrating objects for queue manager 'QMMIG'.
```

```
Default objects statistics : 5 created. 0 replaced. 0 failed.
IBM MQ queue manager 'QMMIG' started using V9.3.0.0.
```

8. 發出下列指令 C:\> `runmqsc QMMIG`，以顯示佇列管理程式的屬性，並記下 VERSION 及 CMDLEVEL 欄位:

```
display qmgr cmdlevel version
1 : display qmgr cmdlevel version
AMQ8408I: Display Queue Manager details.          CMDLEVEL(930)
  QMNAME(QMMIG)
  VERSION(09300000)
end
```

結果

您已順利將佇列管理程式移轉至更新版本的產品。

在 IBM MQ 9.3 上安裝修正套件

如何在具有 IBM MQ 多版本安裝的系統上已安裝 IBM MQ 9.3 的基礎上安裝修正套件。

開始之前

確定您已將佇列管理程式 QMMIG 移轉至 IBM MQ 9.3。如需相關資訊，請參閱 [第 133 頁的『將佇列管理程式移轉至更新版本的 IBM MQ』](#)。

關於這項作業

在此實務範例中，有另一個已安裝的版本 (IBM MQ 9.1)，且在該其他版本下執行的佇列管理程式將不會停止，以示範您可以在執行 IBM MQ 9.3.0.n 的維護活動時繼續使用那些其他版本。

請注意，安裝更新項目時未選取任何進階選項。

程序

1. 以管理者身分登入。
 - a) 如果您已建立批次檔，或使用 `setmqenv` 指令，請發出指令 C:\> `set-mq-930`，以確定您是在 IBM MQ 9.3.0.n 上，其中 n 在此實務範例中是 0。
 - b) 發出指令 C:\> `dspmq -o installation -s`，以顯示佇列管理程式的狀態。
您會收到下列輸出:

```
QMNAME(QM80)                STATUS(Running)INSTNAME(Installation1)
INSTPATH(C:\Program Files\IBM\MQ) INSTVER(9.1.0.0)

QMNAME(QMMIG)                STATUS(Ended unexpectedly)
INSTNAME(Installation2)
INSTPATH(C:\Program Files\IBM\MQ) INSTVER(9.3.0.0)

QMNAME(QM910)                STATUS(Ended immediately) INSTNAME(Installation2)
INSTPATH(C:\Program Files\IBM\MQ) INSTVER(9.3.0.0)
```

2. 停止 IBM MQ 9.3.0 程序。
如需停止處理程序的進一步資訊，請參閱 [在 Windows 上套用及移除維護](#)。

總而言之：

- 對佇列管理程式 QMMIG 及 QM930 發出 `endmqm immediate` 指令。
 - 用滑鼠右鍵按一下工作列中的 IBM MQ 圖示，然後按一下 **停止 IBM MQ**，以停止安裝的 IBM MQ 服務。
3. 找出修正套件。此實務範例使用 IBM MQ 9.3.0.5。
如需最新的清單，請跳至 [建議的 IBM MQ 修正程式](#)：
 - Continuous Delivery 版次，以及 Continuous Delivery 版次的修正程式清單

- Long Term Support 版次的累加安全更新項目及 Long Term Support 版次的修正程式清單

註: 您應該造訪並下載適當的修正套件, 以確認您正在使用最新的修正套件。

- a) 按一下適當的標籤。
在此情況下, 它是 *V9.3.0.5 LTS*
- b) 從 Fix Central 或 Passport Advantage 下載 Windows 軟體。如果您需要下載整個產品, 在此實務範例中, 您將檔案集放入下列目錄 C:\downloads\mq9305; 檔名為 9.3.0-IBM-MQ-Win64-FP0005.zip。
4. 從 .zip 檔案解壓縮檔案, 並執行下列指令: **IBM-MQ-9.3.0-FP0005.exe**。
您會看到安裝 *Anywhere* 對話框, 其中包含下列資訊 *InstallAnywhere is preparing to install ...*
您需要等待直到準備結束, 這可能需要數分鐘。
5. 當視窗出現 *IBM MQ* (修正套件 9.3.0.5 檔案) 時, 按一下 **確定** 以繼續。
 - a) 當出現 簡介 區段時, 請按 **下一步**。
 - b) 當出現 安裝類型 區段時, 請選取更適合您企業的選項, 可能是 **載入檔案並套用修正套件**, 然後按 **下一步**。
 - c) 當出現 資訊 區段時, 按 **下一步**。
 - d) 當「目的地資料夾」區段出現時, 請選取預設位置 C:\Program Files\IBM\source\MQ 9.3.0.5, 然後按 **下一步**。
 - e) 當「進階選項」區段出現時, 請按 **下一步**。
 - f) 當出現 預先安裝摘要 區段時, 請檢查顯示的資訊, 然後按一下 **安裝**。
 - g) 載入程式碼時請稍候。
載入中 區段有一個進度指示器, 當程序完成時, 會標示 載入完成, 並出現 套用修正套件 區段。
 - h) 按一下 **完成**。
因為系統上有多個安裝, 所以會出現一個對話框, 您可以在其中選取要升級的安裝。在此情況下, 它是 *Installation2 (9.3.0.0)*。
 - i) 按一下 **確定**。
 - j) 接受 備份資料夾 的預設值, 然後按一下 **套用**。
即會出現包含進度指示器的其他對話框, 並且已套用最終對話框狀態 *Fix Pack 9.3.0.5*。按一下「**完成**」以結束。
 - k) 按一下 **完成**。

結果

您已順利升級 IBM MQ for Windows 版本及現有產品版本。

Managed File Transfer 實務範例

一般 Managed File Transfer 拓撲的簡介, 以及透過顯示如何設定系統及傳送測試訊息來示範如何使用 Managed File Transfer 功能的實務範例。

- [一般拓撲](#)
- [配置基本伺服器](#)

MFT 一般拓撲

本節說明一般 Managed File Transfer 拓撲。每一個圖表中的雙向箭頭都代表佇列管理程式的連線。如需佇列管理程式連線選項的相關資訊, 請參閱第 139 頁的『[連線功能考量](#)』。

含有一個佇列管理程式的基本拓撲

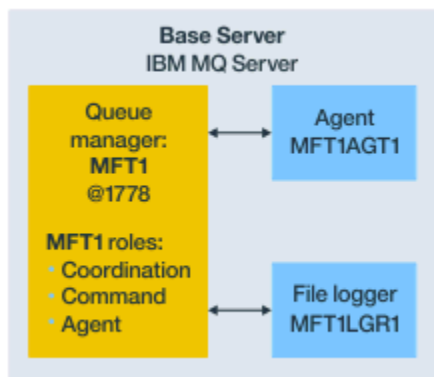


圖 25: 含有一個佇列管理程式的基本拓撲

基本拓撲代表一個完整配置，其中包括協調佇列管理程式。其配置名稱與協調佇列管理程式的名稱相同。如果協調佇列管理程式名稱是 MFT1，則配置名稱就是 MFT1。

基本拓撲是您完成的第一個 Managed File Transfer 配置。完成基本配置之後，來自遠端伺服器的友機代理程式即會新增至基本配置，以交換檔案。

基本拓撲不在基本拓撲伺服器之外交換檔案。不過，基本拓撲可讓您將檔案移至相同伺服器的不同位置，並可供開發用途使用。

含有一個友機代理程式的基本拓撲

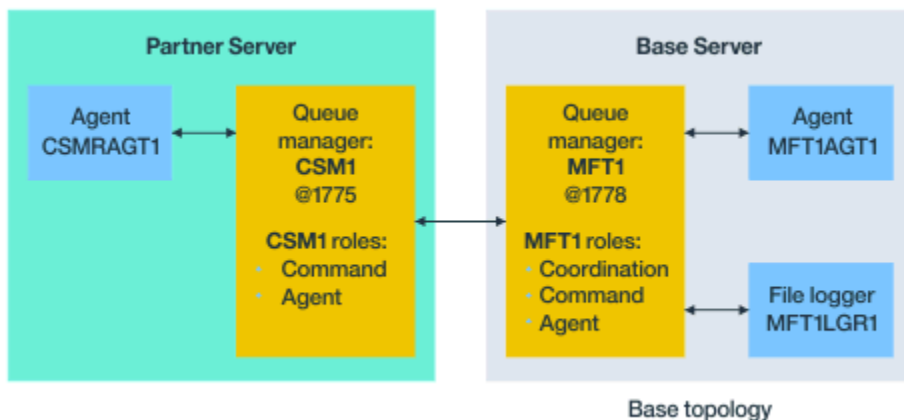


圖 26: 含有一個友機代理程式的基本拓撲

此拓撲可在兩個代理程式之間交換檔案。可以用新增第一個代理程式的類似方式來新增額外的友機代理程式。

您可以對所有這三個 Managed File Transfer 佇列管理程式角色使用單一佇列管理程式，或者，您可以對特定角色使用專用的佇列管理程式。

例如，您可以有一個佇列管理程式專用於協調佇列管理程式角色，而指令及代理程式角色可共用第二個佇列管理程式。

與基本配置不同的伺服器中的遠端代理程式佇列管理程式，與基本配置協調佇列管理程式，這兩者之間的連線必須配置成 IBM MQ 用戶端或 MQI 通道。

協調佇列管理程式的連線是由 **fteSetupCoordination** 指令建立。如果協調佇列管理程式連線未配置成 IBM MQ 用戶端通道，則在友機伺服器上，由友機代理程式伺服器發出的 **fteListAgents** 之類的指令會失敗。

含有不同的協調佇列管理程式和一個友機代理程式的基本拓撲

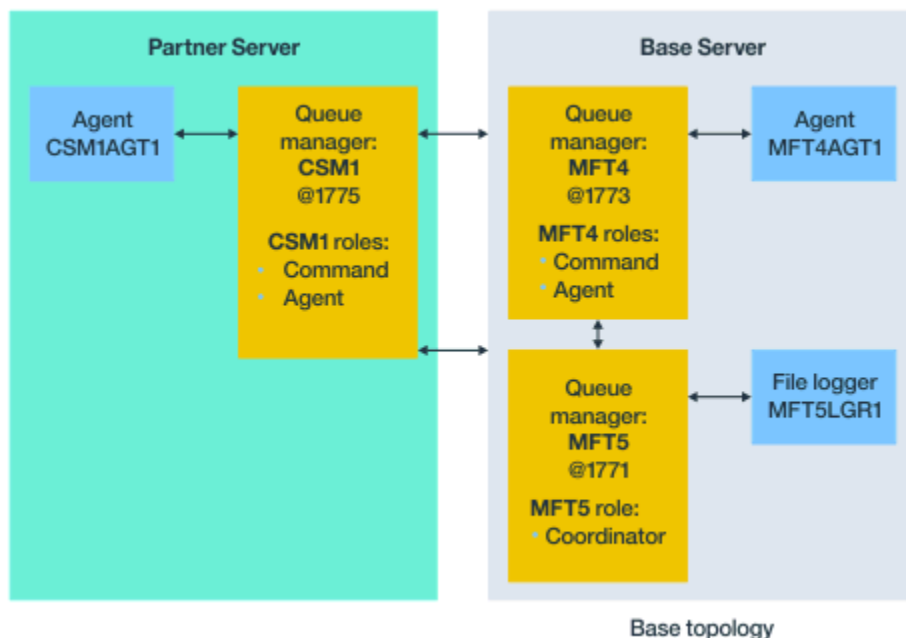


圖 27: 含有不同的協調佇列管理程式和一個友機代理程式的基本拓撲

在圖 3 的基本拓撲中，在基本伺服器上，佇列管理程式 MFT4 由指令與代理程式角色共用，佇列管理程式 MFT5 則為協調佇列管理程式角色專用。

連線功能必須存在於此拓撲中的所有佇列管理程式之間，包括基本拓撲中的佇列管理程式 MFT4 和 MFT5。

在友機伺服器佇列管理程式上，佇列管理程式 CSM1 具有代理程式和指令佇列管理程式的角色。

此拓撲可在兩個代理程式之間交換檔案。每一個友機代理程式都必須連接至佇列管理程式，如此圖所示。可以用新增第一個友機代理程式的類似方式來新增額外的友機代理程式。

含有 Managed File Transfer Agent 友機的基本拓撲

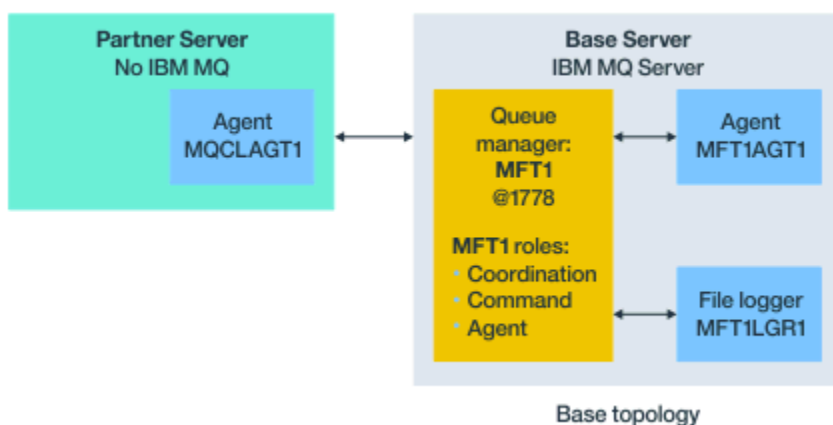


圖 28: 含有 Managed File Transfer Agent 友機的基本拓撲

此拓撲可在兩個代理程式之間交換檔案。

友機代理程式（在圖表中是描述為 MQCLAGT1）中的伺服器未安裝 IBM MQ 伺服器。

友機代理程式是使用與 IBM MQ 已安裝伺服器相同的指令進行配置，但有些例外情況：

- 此友機代理程式與基本佇列管理程式或佇列管理程式之間的配置必須使用 IBM MQ 用戶端連線。

- 不需要執行由友機代理程式伺服器中的配置指令所建立的協調佇列管理程式角色 IBM MQ 定義。協調佇列管理程式定義已存在於基本伺服器中。

但是，您必須：

- 複製在友機伺服器中建立代理程式時所產生的代理程式物件定義
- 將定義檔傳送至基本配置伺服器，以及
- 在識別為基本伺服器中的代理程式佇列管理程式的佇列管理程式中建立定義。

在此情況下，MFT1 負責處理所有這三個角色，而由您在 MFT1 佇列管理程式中建立代理程式 MQCLAGT1 的物件。

除了將物件定義複製到基本伺服器之外，您可以有別的替代方案，亦即在代理程式佇列管理程式所在的基本伺服器上，對 MQCLAGT1 代理程式執行 **fteDefine** 指令。請使用 **fteDefine** 指令產生的定義，在代理程式佇列管理程式上建立所需的代理程式定義。

例如，如圖表中所示，您可以將檔案 MQCLAGT1_create.mqsc 從友機伺服器中的代理程式目錄複製到基本配置伺服器，然後在 MFT1 佇列管理程式中建立所需的代理程式定義。

您在友機代理程式伺服器上完成的配置會建立 Managed File Transfer 配置目錄及所需的內容檔。

從 IBM MQ 9.1.0 開始，您可以在友機伺服器上安裝 [Managed File Transfer 可重新配送的用戶端來源 Fix Central](#)。

註：MQMFT 可重新配送的用戶端與 IBM MQ 可重新配送的用戶端的不同之處在於，它是套裝的，不需要使用 **genmqpkg** 公用程式。如需相關資訊，請參閱 [可重新配送的用戶端](#)。

連線功能考量

在之前的圖表中，代理程式與佇列管理程式之間的每一條線都代表佇列管理程式的連線。

此連線可能是：

- 本端連線
- 連結或訊息通道連線，或者
- IBM MQ 用戶端或 MQI 連線。

您在配置中選取的連線類型視您指定的參數而定

- 當您指定佇列管理程式名稱參數而沒有其他連線參數時，請指定連結連線。

如果所使用的佇列管理程式是 Managed File Transfer 配置的本端，它也代表使用於基本配置伺服器中的本端連線。

- 如果您指定佇列管理程式名稱參數以及對應的主機、埠及通道名稱等參數，請指定 IBM MQ 用戶端連線。

當代理程式位於與代理程式佇列管理程式相同的主機上時，指定產生本端連線的連結類型將更有效率。

配置基本伺服器

如何設定含有不同的配置佇列管理程式的基本伺服器。

開始之前

下列範例假設您：

- 已檢閱過第 139 頁的『[連線功能考量](#)』小節，並瞭解如何影響配置中的佇列管理程式的連線類型。
- 有運作中的 IBM MQ 基礎架構。如需設定佇列管理程式的相關資訊，請參閱 [配置 IBM MQ 佇列管理程式](#)。
- IBM MQ 安全作業已完成。

所有系統資源，例如存取檔案，都已配置足夠的安全性。

如需 Managed File Transfer 安全配置，請參閱 [保護 Managed File Transfer](#) 及 [限制 MFT 代理程式動作的使用者權限](#)。

- 不論是否透過以下哪一種方式來配置，在配置 IBM MQ 之後，所有 IBM MQ 連線都已經過測試：使用範例程式來傳送和接收訊息，或是使用 **amqscnxc** 範例來測試 IBM MQ 用戶端類型的連線。

amqscnxc 範例是透過在範例程式碼中定義通道連線來連接到佇列管理程式，此方式類似 Managed File Transfer 使用 MQI 或 IBM MQ 用戶端類型連線來進行連接的方式。

- 這些指示假設您用於基本配置的伺服器上已安裝一個 IBM MQ 版本。如果您在基本伺服器中有多個 IBM MQ 安裝，則必須對於您要使用的 IBM MQ 版本小心使用正確的檔案路徑。
- 這些指示中所使用的佇列管理程式不需要連線鑑別。

雖然不需要連線鑑別可以更容易完成第一次配置，但如果您的企業需要立即使用連線鑑別，請參閱 [MFT 和 IBM MQ 連線鑑別](#)，以取得如何配置 MQMFTCredentials.xml 認證檔的指示

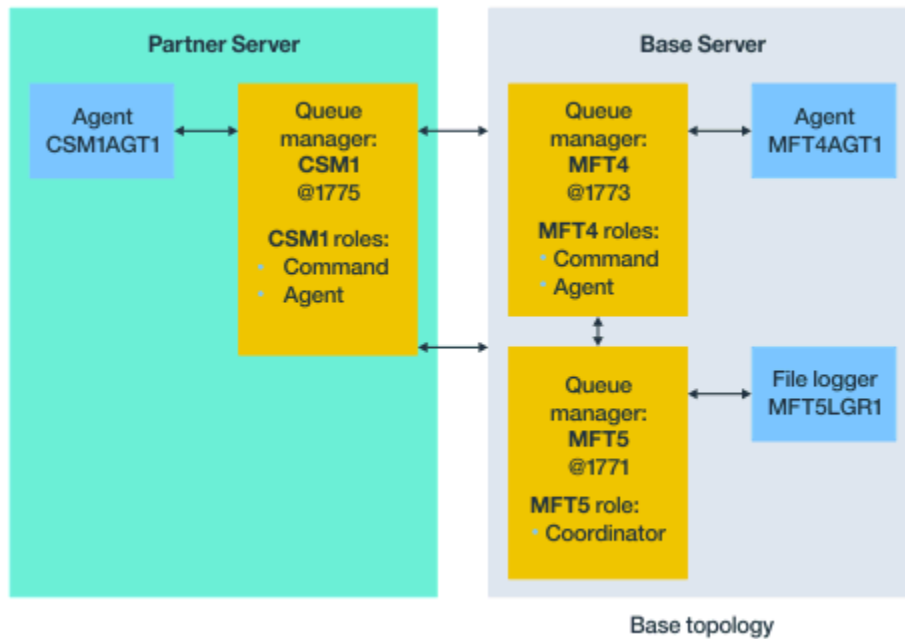


圖 29: 含有不同的協調佇列管理程式和一個友機代理程式的基本拓撲

關於這項作業

範例配置的佇列管理程式角色是：

- 基本伺服器
 - 佇列管理程式 MFT5 是協調佇列管理程式
 - 佇列管理程式 MFT4 是作為代理程式 MFT4AGT1 的代理程式佇列管理程式，並同時作為基本伺服器上的 MFT5 配置的指令佇列管理程式。
- 友機伺服器
 - 佇列管理程式 CSM1 兼作代理程式 CSM1AGT1 的代理程式佇列管理程式，以及友機伺服器上的 MFT5 配置的指令佇列管理程式。
 - 基本伺服器上的佇列管理程式 MFT5 是協調佇列管理程式。

程序

- 配置協調佇列管理程式
- 配置指令佇列管理程式
- 設定代理程式
- 設定日誌程式
- 配置友機伺服器

下一步

設定具有 MQMFT 的 MQExplorer，以測試您的範例設定。

配置協調佇列管理程式

如何配置協調佇列管理程式來協調檔案傳送。

開始之前

確保您對此實務練習設定的佇列管理程式之間有完整連線功能。

關於這項作業

這項作業設定協調佇列管理程式 MFT5，而且此章節中的指示假設您使用一個 IBM MQ 安裝。

如果您有多個安裝架構，則必須先使用 `setmqenv` 指令將 IBM MQ 路徑設定為所需的 IBM MQ 版本，然後再啟動任何配置作業。

程序

1. 以 Managed File Transfer 管理者身分登入。
2. 發出下列指令，以識別協調佇列管理程式並設定配置目錄結構：

```
fteSetupCoordination -coordinationQMGr MFT5
```

協調佇列管理程式目錄

```
C:\data\mqft\config\MFT5
```

coordination.properties 檔案

```
C:\data\mqft\config\MFT5\coordination.properties
```

此指令也會產生您必須對協調佇列管理程式 C:\data\mqft\config\MFT5\MFT5.mqsc 執行的 MQSC 指令檔：

3. 切換至 C:\data\mqft\config\MFT5 目錄。
4. 執行下列指令來配置佇列管理程式，以作為協調佇列管理程式。
您需要提供在步驟 第 141 頁的『2』中發出之指令所產生的 MQSC 指令檔：

```
runmqsc MFT5 < MFT5.mqsc > mft5.txt
```

5. 使用您偏好的編輯器來開啟 mft5.txt 結果檔案。同時確保已順利建立定義。

下一步

設定指令佇列管理程式。

配置指令佇列管理程式

如何配置指令佇列管理程式。

開始之前

請確定您已配置協調佇列管理程式。如需相關資訊，請參閱 第 141 頁的『配置協調佇列管理程式』。

關於這項作業

這項作業可識別指令佇列管理程式。

程序

發出下列指令：

```
fteSetupCommands -connectionQMgr MFT4
```

您會取得下列訊息 BFGCL0245I: 已順利建立 C:\data\mqft\config\MFT4\command.properties 檔案。

指令佇列管理程式不需要額外的 IBM MQ 定義。執行 **fteSetupCommands** 之後，會在 MFT5 配置目錄中建立 command.properties 檔案。

下一步

設定代理程式。

設定代理程式

如何準備檔案傳送代理程式 MFT4AGT1，包括您必須執行的 MQSC Script。

開始之前

您應該已設定指令佇列管理程式。如需相關資訊，請參閱 [第 141 頁的『配置指令佇列管理程式』](#)。

關於這項作業

這項作業會準備 Windows 檔案傳送代理程式 MFT4AGT1。

程序

1. 發出下列指令：

```
fteCreateAgent -agentName MFT4AGT1 -agentQMgr MFT4
```

當您使用 **fteCreateAgent** 指令建立代理程式之後，agents 目錄及代理程式 MFT4AGT1 的子目錄就會新增至 MFT5 目錄中。

您可以在 data\MFT5\agents\MFT4AGT1 目錄中找到下列項目：

- agent.properties 檔案
 - MFT4AGT1_create.mqsc 檔案，其中包含代理程式所需的 IBM MQ 定義。
2. 切換至 data\MFT5\agents\MFT4AGT1 目錄，並透過發出下列指令來建立所需的代理程式佇列管理程式定義：

```
runmqsc MFT4 < MFT4AGT1_create.mqsc > mft4.txt
```

3. 使用您偏好的編輯器來開啟 mft4.txt 結果檔案，並確定已順利建立這些定義。
4. 鍵入下列指令來啟動代理程式：**fteStartAgent** MFT4AGT1。
5. 鍵入下列指令來顯示代理程式：**fteListAgents**。

您應該會看到類似下列內容的輸出：

```
5655-MFT, 5724-H72 Copyright IBM Corp. 2008, 2024. ALL RIGHTS RESERVED
BFGPR0127W: No credentials file has been specified to connect to IBM MQ.
Therefore, the assumption is that IBM MQ authentication has been disabled.
Agent Name:      Queue Manager Name:   Status:
MFT4AGT1        MFT4                   READY
```

註：如果您尚未在 Managed File Transfer 環境中啟用連線鑑別，您可以忽略 BFGPR0127W 訊息。

如果您發出 **ftelistAgents** 指令並收到下列訊息: BFGCL0014W: 不存在符合現行選取準則的代理程式。，請參閱 [如果 fteListAgents 指令未列出您的 MFT 代理程式，怎麼辦。](#)

下一步

設定日誌程式。

設定日誌程式

需要有檔案或資料庫日誌程式，才能保留配置的傳送活動的歷程和審核相關資訊。在此範例中，您會建立檔案日誌程式。

開始之前

您必須已設定：

- 配置佇列管理程式
- 指令佇列管理程式
- 代理程式

程序

1. 發出下列指令：

```
fteCreateLogger -loggerQMgr MFT5 -loggerType FILE  
-fileLoggerMode CIRCULAR -fileSize 5MB -fileCount 3 MFT5lgr1
```

執行 **fteCreateLogger** 指令之後，會建立 `data\mqft\config\MFT5\loggers` 目錄以及 `MFT5LGR1` 子目錄。

`MFT5LGR1` 子目錄保留 `logger.properties` 檔案。另外，在此目錄中還有一個稱為 `MFT5LGR1_create.mqsc` 的檔案，它含有日誌程式所需的 IBM MQ 定義。

2. 切換至 `data\mqft\config\MFT5\loggers\MFT5LGR1` 目錄。
3. 執行關聯的 MQSC 指令檔。

```
runmqsc MFT5 < MFT5_create.mqsc
```

以建立日誌程式所需的定義。

- a) 檢閱物件定義的結果，確認已順利建立所需的物件。

4. 發出下列指令來啟動日誌程式：**fteStartLogger** MFT5LGR1。
5. 檢閱 `data\mqft\logs\MFT5\loggers\MFT5LGR1\logs` 中的 `output0.log` 檔案的內容。

在日誌程式的一些相關資訊之後，最後的陳述應包含此訊息：BFGDB0023I: 日誌程式已完成啟動活動，現在正在執行中。

有時候，日誌程式第一次啟動時，日誌資訊可能不會寫入 `output0.log`。如果 `output0.log` 檔案是空的，請鍵入 **fteStopLogger** MFT5LGR1，並按 **Enter** 鍵來重新啟動日誌程式。

鍵入 **fteStartLogger** MFT5LGR1，並按 **Enter** 鍵來重新啟動日誌程式。 `output0.log` 檔案現在會顯示資料。

第一次啟動代理程式時，相同的行為會延伸至 `output0.log` 檔案的代理程式版本。

使用 **fteStopAgent** 和 **fteStartAgent** 指令來停止及啟動代理程式。然後，您會看到日誌資料寫入至代理程式 `output0.log` 檔案。

結果

您已配置基本伺服器，其中包括此配置的協調佇列管理程式。

下一步

現在，您可以對友機伺服器（其包含遠端代理程式）執行類似工作。

配置友機伺服器

當基本伺服器有不同的協調佇列管理程式時該如何配置友機伺服器

開始之前

確保您已全部完成設定基本伺服器（包含配置佇列管理程式）的所有作業。

關於這項作業

對於 IBM MQ 採用相同的假設，同時安全配置以及 IBM MQ 路徑也會套用至友機伺服器。

開始時，先設定 MFT5 配置目錄，並使用 **fteSetupCoordination** 指令來識別協調佇列管理程式。

程序

1. 發出下列指令來建立友機伺服器配置目錄：

```
fteSetupCoordination -coordinationQMGr MFT5  
-coordinationQMGrHost 177.16.20.15 -coordinationQMGrPort 1771  
-coordinationQMGrChannel MQMFT.MFT5.SVRCONN
```

附註：

- a. 當協調佇列管理程式位於與友機伺服器不同的伺服器上時，基本伺服器協調佇列管理程式的連線必須定義為用戶端連線。

在友機伺服器上，若未能將協調佇列管理程式連線定義為 IBM MQ 用戶端連線，將導致任何連接至該協調佇列管理程式的 Managed File Transfer 指令失敗。

連接至協調佇列管理程式的指令範例：**fteListAgents**。

- b. 您不需要建立 IBM MQ 定義，因為當您配置基本伺服器時，就已完成協調佇列管理程式所需要的定義。

2. 發出下列指令來識別指令佇列管理程式：

```
fteSetupCommands -connectionQMGr CSM1
```

指令佇列管理程式不需要任何額外的 IBM MQ 定義。

3. 發出下列指令，以識別友機代理程式佇列管理程式並且建立友機代理程式佇列管理程式：

```
fteCreateAgent -agentName CSM1AGT1 -agentQMGr CSM1
```

4. 切換至 CSM1AGT1 目錄。

5. 發出下列指令來建立代理程式所需要的 IBM MQ 定義：

```
runmqsc CSM1 < CSM1AGT1_create.mqsc > csm1.txt
```

- a) 使用您偏好的編輯器來開啟 `csm1.txt` 檔案，確認已順利建立所有代理程式需要的定義。

6. 發出下列指令來啟動代理程式：

```
fteStartAgent CSM1AGT1
```

7. 鍵入 **fteListAgents** 來顯示代理程式

您應該會看到類似下列內容的輸出：

```
C:\>fteListAgents  
5655-MFT, 5724-H72 Copyright IBM Corp. 2008, 2024. ALL RIGHTS RESERVED  
BFGPR0127W: No credentials file has been specified to connect to IBM MQ. Therefo  
re, the assumption is that IBM MQ authentication has been disabled.  
Agent Name:      Queue Manager Name:      Status:
```


CSM1AGT1	CSM1	READY
MFT4AGT1	MFT4	READY

註: 如果您尚未在 Managed File Transfer 環境中啟用連線鑑別, 您可以忽略 BFGPR0127W 訊息。

如果您發出 **ftelistAgents** 指令並收到下列訊息: BFGCL0014W: 不存在符合現行選取準則的代理程式。 , 請參閱 [如果 **ftelistAgents** 指令未列出您的 MFT 代理程式, 怎麼辦。](#)

如果其中一個代理程式的狀態為 UNREACHABLE, 請參閱 [如果代理程式顯示為處於 UNKNOWN 狀態, 怎麼辦。](#)

使用 MFT 來設定 IBM MQ Explorer

此作業協助您將「IBM MQ Explorer」連接至 Managed File Transfer 配置。

程序

1. 啟動 IBM MQ Explorer。
2. 在左 Navigator 畫面中, 向下捲動並展開資料夾: Managed File Transfer。
您會看到協調佇列管理程式的項目: MFT5
3. 在 MFT5 上按一下滑鼠右鍵, 然後選取**連接**。
 - a) 在出現的下拉功能表中選取 Agents, 確保代理程式 MFT4AGT1 及 CSMAGT1 都處於 Ready 狀態。

下一步

使用「[IBM MQ Explorer](#)」來測試您的範例設定。

使用 IBM MQ Explorer 來測試檔案傳送

這項作業提供一個範例, 說明在您依照前一個主題所說明來設定 IBM MQ Explorer 之後, 如何搭配使用 IBM MQ Explorer 與 Managed File Transfer 來測試檔案傳送。

開始之前

確保您有工作中的系統, 代理程式是 READY 且 IBM MQ Explorer 在工作中。如需相關資訊, 請參閱第 145 頁的『[使用 MFT 來設定 IBM MQ Explorer](#)』。

關於這項作業

決定要用來測試傳送的檔案, 以及要將它複製到哪一個目錄。就此範例而言, 假設使用 C:\temp\mft 目錄中的 test-file.txt 檔案。

```
C:\temp\mft> dir *
Date stamp 61 test-file.txt
1 File(s) 61 bytes
```

程序

1. 在 Windows 中啟動 IBM MQ Explorer
2. 在左側導覽器畫面中, 展開此資料夾: Managed File Transfer。
您會看到協調佇列管理程式的項目: MFT5
3. 在 MFT5 上按一下滑鼠右鍵, 然後選取**連接**。
4. 連接之後, 在 MFT5 上按一下滑鼠右鍵, 然後選取**新傳送**。
 - a) 使用下拉功能表, 以選取代表來源代理程式的 MFT4AGT1 及代表目的地代理程式的 CSMAGT1。
 - b) 按**下一步**。
 - c) 在下一個視窗上, 按一下**新增**。

即會出現一個很大的對話框。左側用於 Source，右側用於 Destination。

5. 在 Source 畫面上:

- a) 選取**文字傳送**，因為檔案是文字檔。
- b) 選取**瀏覽**，以尋找檔案。

在此案例中，檔案是 C:\temp\mft\test-file.txt。



小心: 請勿按一下 **確定**，因為您需要完成 Destination 畫面。

6. 在 Destination 畫面上:

- a) 輸入您要提供給目的地上檔案的名稱，例如，test-file.txt。
支援使用相對路徑。完整路徑的上半部是啟動目的地代理程式的使用者 ID 的起始目錄。
- b) 如果您需要此選項，請選取 **Overwrite files if present**。
- c) 按一下**確定**。

您選取的檔案即會出現在「**新傳送**」畫面中。

7. 如果已關閉 MFT5 配置功能表，並顯示 +MFT5，請按一下 + 符號來展開此功能表。

8. 保持已選取的 MFT 配置。

接下來，請執行下列程序來檢查傳送的狀態。

9. 在協調佇列管理程式 MFT5 下，按一下**傳送日誌**。

10. 查看 Managed File Transfer - Current Transfer progress panel 中的狀態，緊接在 **傳送日誌** 頂端畫面下方，並等待傳送完成。

如果傳送顯示成功，並出現綠色背景，即表示您已順利完成配置的測試。

如果傳送失敗，並出現紅色背景，則表示發生錯誤。

在大部分情況下，您可以使用上半部**傳送日誌**畫面下方的捲軸來檢視失敗原因摘要。

- a) 如果您無法判定傳送失敗的原因，請在上半部**傳送日誌**畫面中按兩下代表此傳送的項目。
- b) 在出現的蹦現畫面的左窗格中，選取 XML。
- c) 捲動全部資訊，以判斷此錯誤的原因。
- d) 進行必要的更正，然後重新測試傳送。

開始使用 IBM MQ Internet Pass-Thru

這些實務範例顯示如何設定一些簡單的 IBM MQ Internet Pass-Thru (MQIPT) 配置。您也可以使用這些作業來確認已順利安裝產品。

開始之前

在開始使用這些實務範例之前，請確定已完成下列必要條件:

- 您熟悉如何在 IBM MQ 上定義佇列管理程式、佇列及通道。
- 您已安裝 IBM MQ 用戶端及伺服器。
- MQIPT 安裝在 Windows 系統上稱為 C:\mqipt 的目錄中。(這些範例是針對 Windows 系統撰寫的，但將在任何支援的平台上執行。) 如需安裝 MQIPT 的相關資訊，請參閱 [安裝 MQIPT](#)。
- MQIPT 的用戶端、伺服器及每一個實例安裝在個別電腦上。
- 您熟悉如何使用 **amqsputc** 指令將訊息放置在佇列上。
- 您熟悉如何使用 **amqsgetc** 指令從佇列取得訊息。
- 您熟悉如何在 IBM MQ 中設定用戶端權限。

關於這項作業

完成必要條件之後，請完成下列起始步驟，以設定系統備妥可透過實務範例運作。

程序

1. 在 IBM MQ 伺服器上，完成下列作業：

- 定義稱為 MQIPT.QM1 的佇列管理程式。
- 定義稱為 MQIPT.CONN.CHANNEL 的伺服器連線通道。
- 定義稱為 MQIPT.LOCAL.QUEUE 的本端佇列。
- 在埠 1414 上啟動 MQIPT.QM1 的 TCP/IP 接聽器。如果埠 1414 已由另一個應用程式使用，請選擇可用的埠位址，並在下列範例中替代它。
- 請確定連線鑑別和通道鑑別已配置成容許使用您的使用者 ID 從用戶端機器進行用戶端連線。如果連線鑑別設為需要應用程式提供用戶端連線的鑑別認證，則在執行 **amqsputc** 及 **amqsgetc** 指令之前，您需要設定下列其中一個環境變數：

MQSAMP_USER_ID

如果您想要使用使用者 ID 和密碼向佇列管理程式進行鑑別，請設為用於連線鑑別的使用者 ID。



如果您要使用鑑別記號向佇列管理程式進行鑑別，請設為非空白值。

2. 測試從 IBM MQ 用戶端到佇列管理程式的路徑，方法是使用 **amqsputc** 指令將訊息放置在佇列管理程式的本端佇列上，然後使用 **amqsgetc** 指令擷取它。

若要準備本節中的實務範例，請建立並編輯 `mqipt.conf` 檔案，如下所示：

- a. 將 `mqiptSample.conf` 檔案 (位於 MQIPT 安裝目錄的 `samples` 子目錄中) 複製到所選 MQIPT 起始目錄中的 `mqipt.conf`。下列實務範例使用 `C:\mqiptHome` 作為 MQIPT 起始目錄。
- b. 與 `mqipt.conf` `errors` 和 `logs` 一起建立兩個目錄。設定這些目錄的檔案許可權，以便執行 MQIPT 的使用者 ID 可以寫入它們。
- c. 從 `mqipt.conf` 檔案中刪除所有路徑。
- d. 在其餘 `[global]` 區段中，檢查 **ClientAccess** 是否存在且設為 `true`。

下一步

在設定系統之後，您已準備好開始下列實務範例：

- [第 148 頁的『驗證 MQIPT 是否正確運作』](#)
- [第 149 頁的『建立金鑰環檔案』](#)
- [第 151 頁的『建立測試憑證』](#)
- [第 152 頁的『鑑別 TLS 伺服器』](#)
- [第 155 頁的『鑑別 TLS 用戶端』](#)
- [第 160 頁的『配置 HTTP 通道作業』](#)
- [第 162 頁的『配置存取控制』](#)
- [第 164 頁的『配置 SOCKS Proxy』](#)
- [第 165 頁的『配置 SOCKS 用戶端』](#)
- [第 167 頁的『配置 MQIPT 叢集作業支援』](#)
- [第 170 頁的『配置埠號』](#)
- [第 171 頁的『使用 LDAP 伺服器擷取 CRL』](#)
- [第 174 頁的『以 TLS Proxy 模式執行 MQIPT』](#)
- [第 175 頁的『使用安全管理程式以 TLS Proxy 模式執行 MQIPT』](#)
- [第 178 頁的『使用安全結束程式』](#)
- [第 179 頁的『使用安全結束程式將用戶端連線要求遞送至 IBM MQ 佇列管理程式伺服器』](#)
- [第 182 頁的『動態遞送用戶端連線要求』](#)
- [第 185 頁的『使用憑證結束程式來鑑別 TLS 伺服器』](#)

驗證 MQIPT 是否正確運作

使用此簡式配置設定，以確保已正確安裝 MQIPT。

開始之前

- 在開始使用此實務範例之前，請確定您已完成 [第 146 頁的『開始使用 IBM MQ Internet Pass-Thru』](#) 中列出的必要作業。

關於這項作業

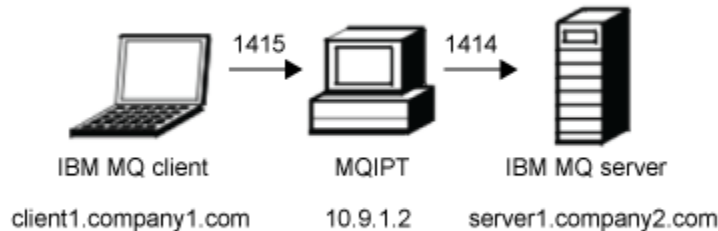


圖 30: 安裝驗證測試網路圖

此圖顯示從 IBM MQ 用戶端 (在埠 1415 上稱為 client1.company1.com) 到 MQIPT 到 IBM MQ 伺服器 (在埠 1414 上稱為 server1.company2.com) 的連線。

程序

若要驗證 MQIPT 是否正常運作，請完成下列步驟：

1. 定義 MQIPT 路徑。

在 MQIPT 電腦上，編輯 `mqipt.conf` 並新增路徑定義：

```
[route]
ListenerPort=1415
Destination=server1.company2.com
DestinationPort=1414
```

2. 啟動 MQIPT。

開啟命令提示字元，並輸入下列指令：

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
```

其中 `C:\mqiptHome` 指出 MQIPT 配置檔 `mqipt.conf` 的位置，`ipt1` 是要提供給 MQIPT 實例的名稱。

下列訊息指出 MQIPT 已順利啟動：

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt1
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to:
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....using MQ protocol
MQCPI078 Route 1415 ready for connection requests
```

3. 在 IBM MQ 用戶端系統上的命令提示字元中，輸入下列指令：

- a) 設定 `MQSERVER` 環境變數：

```
SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
```

b) 放置訊息:

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1  
Hello world
```

在鍵入訊息字串之後，按 Enter 鍵兩次。

c) 取得訊息:

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
```

即會傳回訊息 "Hello world"。

4. 若要停止 IBM MQ，請輸入下列指令:

```
mqiptAdmin -stop -n ipt1
```

建立金鑰環檔案

在此實務範例中，您可以要求憑證並建立金鑰環檔，以容許 MQIPT 來使用 TLS。

開始之前

在開始使用此實務範例之前，請確定您已完成 [第 146 頁的『開始使用 IBM MQ Internet Pass-Thru』](#) 中列出的必要作業。

這項作業假設您使用 **mqiptKeyman** (iKeyman) 向授信憑證管理中心 (CA) 要求新的憑證，且您的個人憑證會以檔案形式傳回給您 (例如 `server.cer`)。這足以執行伺服器鑑別。如果您需要用戶端鑑別，則必須要求第二個憑證 (例如，`client.cer`)，並執行下列步驟兩次，以建立兩個金鑰環檔案。

關於這項作業

您可以使用 **mqiptKeycmd** 指令行介面 (CLI) 或 **mqiptKeyman** GUI 來要求憑證。然後，您應該在 **SSLServerKeyRing** 或 **SSLClientKeyRing** MQIPT 路徑內容，視憑證是供入埠或出埠連線使用而定。

程序

請使用下列其中一種方法來建立金鑰環檔:

- 使用 **mqiptKeycmd** 指令行介面 (CLI)
 - a) 輸入下列指令，以建立新的 PKCS #12 金鑰環檔:

```
mqiptKeycmd -keydb -create -db server_name.pfx -pw password -type pkcs12
```

其中:

- **-db** 指定金鑰環檔案 (`server_name.pfx`) 的名稱。
- **-pw** 指定您稍後必須使用 **mqiptPW** 指令加密的金鑰環密碼 (`password`)。

b) 輸入下列指令，以建立新的憑證申請:

```
mqiptKeycmd -certreq -create -db server_name.pfx -pw password -type pkcs12  
-file cert_file_name.req -label label -dn DN_identity  
-sig_alg signature_algorithm -size key_size
```

其中:

- **-file** 指定所要求憑證的檔名。
- **-label** 指定您選擇的唯一名稱; 最好不要包含空格字元。
- **-dn** 指定 MQIPT 路徑的適當「識別名稱」身分; 例如，"CN=Test Certificate , OU=Sales , O=Example , C = US"。

- **-sig_alg** 指定雜湊演算法; 例如 SHA256WithRSA。
- **-size** 指定公開金鑰的大小; 例如 2048。

如果您使用給定的範例值，則此指令會建立具有 2048 位元 RSA 公開金鑰的數位憑證，以及使用 RSA 與 SHA-256 雜湊演算法的數位簽章。

建立憑證時，請小心選擇適當的公開金鑰加密演算法、金鑰大小及數位簽章演算法，以符合您組織的安全需求。如需相關資訊，請參閱 [MQIPT 的數位憑證考量](#)。

將指令所建立的憑證申請檔案 (*cert_file_name.req*) 傳送至 CA 進行簽署。

- c) 當您從 CA 接收已簽署的個人憑證時，請輸入下列指令，將憑證新增至伺服器金鑰環：

```
mqiptKeycmd -cert -receive -db server_name.pfx -pw password
            -type pkcs12 -file cert_file_name.crt
```

- 使用 **mqiptKeyman** GUI

- a) 執行下列指令來開啟 GUI:

```
mqiptKeyman
```

- b) 按一下 **金鑰資料庫檔 > 新建**。
- c) 選取金鑰資料庫的類型 PKCS12。
- d) 輸入新金鑰環檔案的檔名和位置。
按一下 **確定**。
- e) 輸入並確認新金鑰環檔案的密碼。
這是您稍後必須使用 **mqiptPW** 指令加密的金鑰環密碼。按一下 **確定**，以建立新的個人憑證金鑰環檔案。
- f) 按一下 **建立 > 新建憑證申請**，以建立憑證申請。
- g) 在 **金鑰標籤** 欄位中輸入新憑證的標籤。
標籤可以是您選擇的任何唯一名稱; 最好不要包含空格字元。
- h) 根據您組織的安全需求，選取適當的金鑰大小和數位簽章演算法。
如需相關資訊，請參閱 [MQIPT 的數位憑證考量](#)。
- i) 在選用 DN 欄位中輸入 MQIPT 路徑的適當「識別名稱」身分。
- j) 輸入要建立之憑證申請的檔名，然後按一下 **確定**。
即會產生憑證申請，並以您指定的名稱儲存。將此檔案傳送至 CA 進行簽署。
- k) 當您從 CA 接收已簽章的個人憑證時，必須在金鑰環檔案中接收它。
在「**金鑰資料庫內容**」畫面中，從下拉清單中選取 **個人憑證**。然後按一下 **接收**。
- l) 輸入儲存已簽章憑證的檔案名稱，然後按一下 **確定**。

下一步

您還必須確保簽署個人憑證之 CA 的 CA 憑證存在於 CA 金鑰環檔案中。視您的 MQIPT 配置而定，CA 金鑰環檔可能與個人憑證金鑰環檔不同。

若要使用個別 CA 金鑰環檔，您可以使用 MQIPT 隨附的名為 *sslCAdefault.pfx* 的範例 CA 金鑰環檔，或建立新的 PKCS #12 金鑰環檔。除非已在範例金鑰環檔案中，否則您需要將簽署個人憑證之 CA 的公用 CA 憑證新增至 CA 金鑰環。公用 CA 憑證可能已隨您的個人憑證一起傳回。如果沒有，則您必須從提供個人憑證的相同 CA 要求 CA 憑證，然後將它新增至金鑰環。

若要新增 CA 憑證，您可以使用 **mqiptKeycmd** CLI 或 **mqiptKeyman** GUI。

若要使用 **mqiptKeycmd** CLI 來新增 CA 憑證，請發出下列指令：

```
mqiptKeycmd -cert -add -db sslCAdefault.pfx -pw password -type pkcs12
            -file ca_file_name.crt -label label
```

其中：

- **-db** 指定 CA 金鑰環檔名，在此情況下為 `sslCAdefault.pfx`。
- **-pw** 指定金鑰環密碼。名為 `sslCAdefault.pfx` 的範例 CA 金鑰環檔案的密碼為 `mqiptSample`。
- **-file** 指定 CA 傳回的檔案名稱。
- **-label** 指定您選擇的唯一名稱；最好不要使用空格字元。

若要使用 iKeyman GUI 來新增 CA 憑證，請執行下列動作：

- 在「金鑰資料庫內容」畫面中，從下拉清單中選取 **簽章者憑證**
- 按一下**新增**。
- 輸入包含 CA 憑證的檔案名稱，然後按一下 **確定**。
- 輸入 CA 憑證的標籤。標籤可以是您選擇的任何唯一名稱；最好不要使用空格字元。按一下**確定**。

發出下列指令來加密金鑰環密碼：

```
mqiptPW
```

輸入要在提示時加密的金鑰環密碼。視入埠或出埠連線使用憑證而定，將 `mqipt.conf` 配置檔中適當內容的值設為 `mqiptPW` 指令所輸出的已加密密碼；例如 `SSLServerKeyRingPW` 或 `SSLClientKeyRingPW`。如需加密金鑰環密碼的相關資訊，請參閱 [加密儲存的密碼](#)，以加密金鑰環密碼。

若要使用這些新的金鑰環檔案進行伺服器鑑別，請將金鑰環檔案放置在 MQIPT 起始目錄下名為 `ssl` 的目錄中，並設定下列路由內容：

```
SSLClientCAKeyRing=C:\\mqiptHome\\ssl\\sslCAdefault.pfx
SSLClientCAKeyRingPW=encrypted_password
SSLServerKeyRing=C:\\mqiptHome\\ssl\\myServer.pfx
SSLServerKeyRingPW=encrypted_password
SSLServerCAKeyRing=C:\\mqiptHome\\ssl\\sslCAdefault.pfx
SSLServerCAKeyRingPW=encrypted_password
```

如需配置 MQIPT 以使用 TLS 的相關資訊，請參閱實務範例 [第 152 頁的『鑑別 TLS 伺服器』](#)。

建立測試憑證

在此實務範例中，您可以建立可用來測試 MQIPT 路徑的自簽憑證。MQIPT 路徑可以使用此憑證來向遠端對等節點識別其本身。

在測試情境中，如果您必須確保 TLS 連線功能，而不需支付憑證管理中心 (CA) 的費用，則自簽憑證可能很有用。不過，您不應在正式作業環境中使用自簽憑證。如果您需要用於正式作業的憑證，請參閱 [第 149 頁的『建立金鑰環檔案』](#)。

開始之前

- 在開始使用此實務範例之前，請確定您已完成 [第 146 頁的『開始使用 IBM MQ Internet Pass-Thru』](#) 中列出的必要作業。

關於這項作業

您可以使用 `mqiptkeycmd` (iKeyman) 指令行介面 (CLI) 或 `mqiptkeyman` GUI 來要求憑證。然後，您應該在 `SSLServerKeyRing` 或 `SSLClientKeyRing` MQIPT 路徑內容中指定包含憑證的金鑰環檔，視憑證是供入埠或出埠連線使用而定。

程序

請使用下列其中一種方法來建立測試憑證：

- 使用指令行介面 (CLI)
 - a) 輸入下列指令，以建立新的 PKCS #12 金鑰環檔：

```
mqiptKeycmd -keydb -create -db server_name.pfx -pw password -type pkcs12
```

其中：

- **-db** 指定金鑰環檔案 (*server_name.pfx*) 的名稱。
 - **-pw** 指定稍後必須使用 **mqiptPW** 公用程式加密的金鑰環密碼 (*password*)。
- b) 輸入下列指令，以建立用於測試的自簽個人憑證：

```
mqiptKeycmd -cert -create -db server_name.pfx -pw password -type pkcs12
            -label label -dn DN_identity
            -sig_alg signature_algorithm -size key_size
```

其中：

- **-label** 指定您選擇的唯一名稱；最好不要包含空格字元。
- **-dn** 指定 MQIPT 路徑的適當「識別名稱」身分；例如，"CN=Test Certificate , OU=Sales , O=Example , C = US"。
- **-sig_alg** 指定雜湊演算法；例如 SHA256WithRSA。
- **-size** 指定公開金鑰的大小；例如 2048。

如果您使用給定的範例值，則此指令會建立具有 2048 位元 RSA 公開金鑰的數位憑證，以及使用 RSA 與 SHA-256 雜湊演算法的數位簽章。

建立憑證時，請小心選擇適當的公開金鑰加密演算法、金鑰大小及數位簽章演算法，以符合您組織的安全需求。如需相關資訊，請參閱 [MQIPT 的數位憑證考量](#)。

- 使用 GUI

- a) 執行下列指令來開啟 GUI：

```
mqiptKeyman
```

- b) 按一下 **金鑰資料庫檔 > 新建**。
- c) 選取金鑰資料庫的類型；例如 PKCS12。
- d) 輸入新金鑰環檔案的檔名和位置。
按一下 **確定**。
- e) 輸入新金鑰環檔的密碼。
再次輸入密碼以確認。這是您稍後必須使用 **mqiptPW** 公用程式加密的金鑰環密碼。按一下 **確定**，以建立新的個人憑證金鑰環檔案。
- f) 按一下 **建立 > 新建自簽憑證**，以建立新的自簽個人憑證。
- g) 在 **金鑰標籤** 欄位中輸入新憑證的標籤。
標籤可以是您選擇的任何唯一名稱；最好不要包含空格字元。
- h) 根據您組織的安全需求，選取適當的金鑰大小和數位簽章演算法。
如需相關資訊，請參閱 [MQIPT 的數位憑證考量](#)。
- i) 在選用 DN 欄位中輸入 MQIPT 路徑的適當「識別名稱」身分，然後按一下 **確定**。

下一步

發出下列指令來加密金鑰環密碼：

```
mqiptPW
```

輸入要在提示時加密的金鑰環密碼。視入埠或出埠連線使用憑證而定，將 **mqipt.conf** 配置檔中適當內容的值設為 **mqiptPW** 指令所輸出的已加密密碼；例如 **SSLServerKeyRingPW** 或 **SSLClientKeyRingPW**。如需加密金鑰環密碼的相關資訊，請參閱 [加密儲存的密碼](#)，以加密金鑰環密碼。

鑑別 TLS 伺服器

在此實務範例中，您可以使用範例 (*sslSample.pfx*) 金鑰環檔中的測試憑證來測試 TLS 連線，該檔案隨附於 MQIPT 安裝目錄的 *samples/ssl* 子目錄中的 MQIPT。

開始之前

在開始使用此實務範例之前，請確定您已完成第 146 頁的『開始使用 IBM MQ Internet Pass-Thru』中列出的必要作業，並已閱讀 [MQIPT 中的 SSL/TLS 支援主題](#)。

關於這項作業

透過 MQIPT 的兩個實例，在 IBM MQ 用戶端與 IBM MQ 伺服器之間建立連線。MQIPT 1 與 MQIPT 2 之間的連線使用 TLS，MQIPT 1 充當 TLS 用戶端，MQIPT 2 充當 TLS 伺服器。

在 TLS 信號交換期間，伺服器會將其測試憑證傳送至用戶端，且用戶端會使用其憑證副本並設定「同層級信任」旗標來鑑別伺服器。使用 CipherSuite SSL_RSA_WITH_AES_256_CBC_SHA256。此實務範例中的 `mqipt.conf` 配置檔基於在第 148 頁的『驗證 MQIPT 是否正確運作』實務範例中建立的配置檔。如需如何建立測試憑證以在此範例中使用的詳細資料，請參閱第 151 頁的『建立測試憑證』。

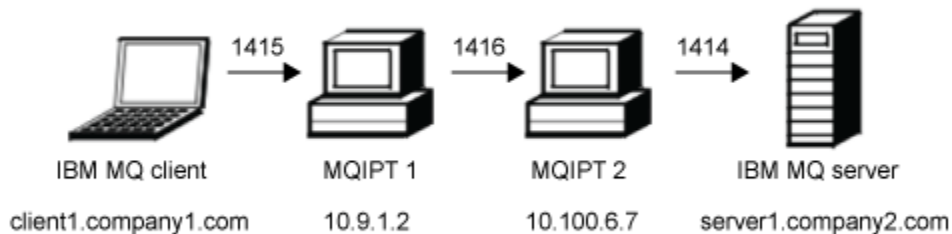


圖 31: SSL/TLS 伺服器網路圖

此圖顯示透過兩個 MQIPT 實例，從 IBM MQ 用戶端 (在埠 1415 上稱為 `client1.company1.com`) 到 IBM MQ 伺服器 (在埠 1414 上稱為 `server1.company2.com`) 的連線。

程序

若要鑑別 TLS 伺服器，請完成下列步驟：

1. 在 MQIPT 1 系統上：
 - a) 編輯 `mqipt.conf` 並新增下列路由定義：

```
[route]
ListenerPort=1415
Destination=10.100.6.7
DestinationPort=1416
SSLClient=true
SSLClientKeyRing=C:\mqipt\samples\ssl\sslSample.pfx
SSLClientKeyRingPW=<mqiptPW>!PCaB1HWiFMOp43ngjwgArg==!6N/vsbqru7iqMhFN+wozxQ==
SSLClientCipherSuites=SSL_RSA_WITH_AES_256_CBC_SHA256
```

- b) 開啟命令提示字元，並啟動 MQIPT：

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
```

其中 `C:\mqiptHome` 指出 MQIPT 配置檔 `mqipt.conf` 的位置，`ipt1` 是要提供給 MQIPT 實例的名稱。

下列訊息指出 MQIPT 已順利啟動：

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt1
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1415 is starting and will forward messages to :
```

```

MQCPI034 ....10.100.6.7(1416)
MQCPI035 ....using MQ protocol
MQCPI036 ....SSL Client side enabled with properties :
MQCPI139 .....secure socket protocols <NULL>
MQCPI031 .....cipher suites SSL_RSA_WITH_AES_256_CBC_SHA256
MQCPI032 .....key ring file C:\mqipt\samples\ssl\sslSample.pfx
MQCPI047 .....CA key ring file <NULL>
MQCPI071 .....site certificate uses
UID=*,CN=*,T=*,OU=*,DC=*,O=*,STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI038 .....peer certificate uses
UID=*,CN=*,T=*,OU=*,DC=*,O=*,STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI078 Route 1415 ready for connection requests

```

2. 在 MQIPT 2 系統上:

a) 編輯 mqipt.conf 並新增下列路由定義:

```

[route]
ListenerPort=1416
Destination=Server1.company2.com
DestinationPort=1414
SSLServer=true
SSLServerKeyRing=C:\mqipt\samples\ssl\sslSample.pfx
SSLServerKeyRingPW=<mqiptPW>!PCaB1HwrFM0p43ngjwgArg==!6N/vsbqru7iqMhFN+wozxQ==
SSLServerCipherSuites=SSL_RSA_WITH_AES_256_CBC_SHA256

```

b) 開啟命令提示字元，並啟動 MQIPT:

```

C:
cd \mqipt\bin
mqipt .. -n ipt2

```

其中 .. 指出 MQIPT 配置檔 mqipt.conf 位於上層目錄中，ipt2 是要提供給 MQIPT 實例的名稱。

下列訊息指出 MQIPT 已順利啟動:

```

5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt2
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqipt\logs will be used to store the log files
MQCPI006 Route 1416 is starting and will forward messages to :
MQCPI034 ....Server1.company2.com(1414)
MQCPI035 ....using MQ protocol
MQCPI037 ....SSL Server side enabled with properties :
MQCPI139 .....secure socket protocols <NULL>
MQCPI031 .....cipher suites SSL_RSA_WITH_AES_256_CBC_SHA256
MQCPI032 .....key ring file C:\mqipt\samples\ssl\sslSample.pfx
MQCPI047 .....CA key ring file <NULL>
MQCPI071 .....site certificate uses
UID=*,CN=*,T=*,OU=*,DC=*,O=*,STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI038 .....peer certificate uses
UID=*,CN=*,T=*,OU=*,DC=*,O=*,STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI033 .....client authentication set to false
MQCPI078 Route 1416 ready for connection requests

```

3. 在 IBM MQ 用戶端上的命令提示字元中，輸入下列指令:

a) 設定 MQSERVER 環境變數:

```
SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
```

b) 放置訊息:

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world
```

在鍵入訊息字串之後按 Enter 鍵兩次。

c) 取得訊息:

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
```

即會傳回訊息 "Hello world".

鑑別 TLS 用戶端

在此實務範例中，您可以使用範例測試憑證來測試 TLS 連線，以執行伺服器及用戶端鑑別。

開始之前

在開始使用此實務範例之前，請確定您已完成第 146 頁的『開始使用 IBM MQ Internet Pass-Thru』中列出的必要作業，並已閱讀 MQIPT 中的 SSL/TLS 支援主題。

關於這項作業

透過 MQIPT 的兩個實例，在 IBM MQ 用戶端與 IBM MQ 伺服器之間建立連線。MQIPT 1 與 MQIPT 2 之間的連線使用 TLS，MQIPT 1 充當 TLS 用戶端，MQIPT 2 充當 TLS 伺服器。

在 TLS 信號交換期間，伺服器會將其測試憑證傳送至用戶端。用戶端會使用其憑證副本與「同層級信任」旗標，來鑑別伺服器。然後，用戶端會將其測試憑證傳送至伺服器。伺服器會使用其憑證副本與「同層級信任」旗標，來鑑別用戶端。使用 CipherSuite SSL_RSA_WITH_AES_256_CBC_SHA256。此實務範例中的 mqipt.conf 配置檔基於在第 148 頁的『驗證 MQIPT 是否正確運作』實務範例中建立的配置檔。

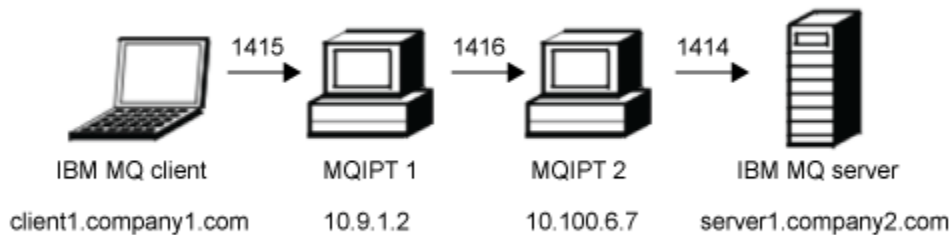


圖 32: SSL/TLS 用戶端網路圖

此圖顯示透過兩個 MQIPT 實例，從 IBM MQ 用戶端 (在埠 1415 上稱為 client1.company1.com) 到 IBM MQ 伺服器 (在埠 1414 上稱為 server1.company2.com) 的連線。

程序

若要鑑別 TLS 用戶端，請完成下列步驟：

1. 在 MQIPT 1 系統上：
 - a) 編輯 mqipt.conf 並新增下列路由定義：

```
[route]
ListenerPort=1415
Destination=10.100.6.7
DestinationPort=1416
SSLClient=true
SSLClientKeyRing=C:\mqipt\samples\ssl\sslSample.pfx
SSLClientKeyRingPW=<mqiptPW>1!PCaB1HWrFMOp43ngjwgArg==!6N/vsbqr7iqMhFN+wozxQ==
SSLClientCipherSuites=SSL_RSA_WITH_AES_256_CBC_SHA256
```

- b) 開啟命令提示字元，並啟動 MQIPT：

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
```

其中 C:\mqiptHome 指出 MQIPT 配置檔 mqipt.conf 的位置，ipt1 是要提供給 MQIPT 實例的名稱。

下列訊息指出 MQIPT 已順利啟動:

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt1
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1415 is starting and will forward messages to :
MQCPI034 ....10.100.6.7(1416)
MQCPI035 ...using MQ protocol
MQCPI036 ....SSL Client side enabled with properties :
MQCPI139 .....secure socket protocols <NULL>
MQCPI031 .....cipher suites SSL_RSA_WITH_AES_256_CBC_SHA256
MQCPI032 .....key ring file C:\mqipt\samples\ssl\sslSample.pfx
MQCPI047 .....CA key ring file <NULL>
MQCPI071 .....site certificate uses
UID=*,CN=*,T=*,OU=*,DC=*,O=*,STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI038 .....peer certificate uses
UID=*,CN=*,T=*,OU=*,DC=*,O=*,STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI078 Route 1415 ready for connection requests
```

2. 在 MQIPT 2 系統上:

a) 編輯 mqipt.conf 並新增下列路由定義:

```
[route]
ListenerPort=1416
Destination=Server1.company2.com
DestinationPort=1414
SSLServer=true
SSLServerAskClientAuth=true
SSLServerKeyRing=C:\mqipt\samples\ssl\sslSample.pfx
SSLServerKeyRingPW=<mqiptPW>1!PCaB1HWrFM0p43ngjwgArg==!6N/vsbqru7iqMhFN+wozxQ==
SSLServerCipherSuites=SSL_RSA_WITH_AES_256_CBC_SHA256
```

b) 開啟命令提示字元，並啟動 MQIPT:

```
C:
cd \mqipt\bin
mqipt .. -n ipt2
```

其中 .. 指出 MQIPT 配置檔 mqipt.conf 位於上層目錄中，ipt2 是要提供給 MQIPT 實例的名稱。

下列訊息指出 MQIPT 已順利啟動:

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt2
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqipt\logs will be used to store the log files
MQCPI006 Route 1416 is starting and will forward messages to :
MQCPI034 ....Server1.company2.com(1414)
MQCPI035 ...using MQ protocol
MQCPI037 ....SSL Server side enabled with properties :
MQCPI139 .....secure socket protocols <NULL>
MQCPI031 .....cipher suites SSL_RSA_WITH_AES_256_CBC_SHA256
MQCPI032 .....key ring file C:\mqipt\samples\ssl\sslSample.pfx
MQCPI047 .....CA key ring file <NULL>
MQCPI071 .....site certificate uses
UID=*,CN=*,T=*,OU=*,DC=*,O=*,STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI038 .....peer certificate uses
UID=*,CN=*,T=*,OU=*,DC=*,O=*,STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI033 .....client authentication set to true
MQCPI078 Route 1416 ready for connection requests
```

3. 在 IBM MQ 用戶端系統上的命令提示字元中，輸入下列指令:

a) 設定 MQSERVER 環境變數:

```
SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
```

b) 放置訊息:

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world
```

在鍵入訊息字串之後按 Enter 鍵兩次。

c) 取得訊息:

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
```

即會傳回訊息 "Hello world"。

鑑別 TLS 用戶端及伺服器

您可以執行 MQIPT 作為 TLS 伺服器及用戶端，以終止送入的 TLS 階段作業，並使用個別 TLS 連線將資料轉遞至目的地。

開始之前

在開始使用此實務範例之前，請確定您已完成第 146 頁的『開始使用 IBM MQ Internet Pass-Thru』中列出的必要作業，並已閱讀 MQIPT 中的 SSL/TLS 支援主題。

註: 為了方便起見，此實務範例使用自簽憑證。請勿在正式作業環境中使用任何自簽憑證。相反地，請取得授信憑證管理中心 (CA) 所簽署的憑證。

關於這項作業

透過 MQIPT 的單一實例，在 IBM MQ 用戶端與 IBM MQ 伺服器之間建立連線。IBM MQ 用戶端與 MQIPT 之間以及 MQIPT 與 IBM MQ 伺服器之間的連線都使用 TLS。因此，MQIPT 路徑同時是 TLS 伺服器和 TLS 用戶端。

在用戶端與 MQIPT 之間的 TLS 信號交換期間，用戶端與 MQIPT 會彼此傳送其憑證以鑑別連線。建立用戶端與 MQIPT 之間的連線時，MQIPT 會建立與 IBM MQ 伺服器的個別 TLS 連線。MQIPT 和 IBM MQ 伺服器會將它們的憑證彼此傳送，以鑑別連線。

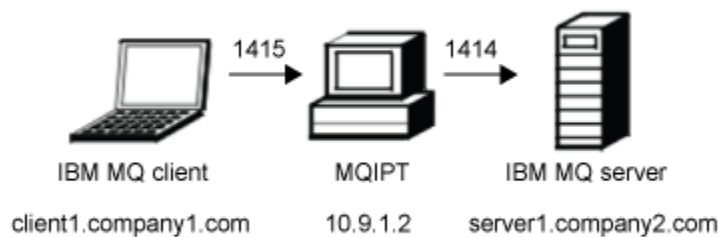


圖 33: SSL/TLS 伺服器及用戶端網路圖

此圖顯示從 IBM MQ 用戶端 (稱為 client1.company1.com) 透過 MQIPT 的單一實例到 IBM MQ 伺服器 (在埠 1414 上稱為 server1.company2.com) 的連線。

程序

若要將 MQIPT 路徑配置為 TLS 伺服器及 TLS 用戶端，請完成下列步驟:

1. 在用戶端執行所在的系統上，完成下列步驟，以將 IBM MQ 用戶端配置為使用 TLS 連線。

a) 建立用戶端的金鑰儲存庫。

輸入下列指令，在 C:\ProgramData\IBM\MQ 目錄中建立名為 clientkey.kdb 的新金鑰儲存庫:

```
runmqakm -keydb -create -db C:\ProgramData\IBM\MQ\clientkey.kdb -pw password -stash
```

其中 *password* 是金鑰儲存庫密碼。

如需相關資訊，請參閱 [在 AIX, Linux, and Windows 上設定金鑰儲存庫](#)。

- b) 在您的步驟 [第 157 頁](#) 的『1.a』中建立的用戶端金鑰儲存庫中，建立用戶端的個人憑證。
輸入下列指令，以針對用戶端建立具有標籤 `clientcert` 的新自簽憑證：

```
runmqakm -cert -create -db C:\ProgramData\IBM\MQ\clientkey.kdb -stashed  
-label clientcert -dn CN=Client
```

如需相關資訊，請參閱 [在 AIX, Linux, and Windows 上建立自簽個人憑證](#)。

- c) 從用戶端金鑰儲存庫擷取憑證。
輸入下列指令，以擷取用戶端憑證：

```
runmqakm -cert -extract -db C:\ProgramData\IBM\MQ\clientkey.kdb -stashed -label clientcert  
-target C:\ProgramData\IBM\MQ\client.crt -format ascii
```

如需相關資訊，請參閱 [從 AIX, Linux, and Windows 上的金鑰儲存庫擷取自簽憑證的公用部分](#)。

- d) 將擷取的憑證檔案複製到執行 MQIPT 的系統。
2. 在執行 IBM MQ 伺服器的系統上，完成下列步驟，以將佇列管理程式配置為使用 TLS 連線。
 - a) 建立佇列管理程式的金鑰儲存庫。
輸入下列指令，為佇列管理程式建立名為 `key.kdb` 的新金鑰儲存庫。

```
runmqakm -keydb -create -db C:\ProgramData\IBM\MQ\mqgrs\MQIPT!QM1\ssl\key.kdb -pw  
password -stash
```

其中 *password* 是金鑰儲存庫密碼。

請確定 mqm 使用者已獲授與對 `C:\ProgramData\IBM\MQ\mqgrs\MQIPT!QM1\ssl\key.sth` 隱藏檔的讀取權，指令會建立該隱藏檔以儲存加密金鑰儲存庫密碼。

如需相關資訊，請參閱 [在 AIX, Linux, and Windows 上設定金鑰儲存庫](#)。

- b) 在您的步驟 [第 158 頁](#) 的『2.a』中建立的佇列管理程式金鑰儲存庫中，建立佇列管理程式的個人憑證。
輸入下列指令，以針對佇列管理程式建立具有標籤 `ibmwebspheremqmqipt.qm1` 的新自簽憑證：

```
runmqakm -cert -create -db C:\ProgramData\IBM\MQ\mqgrs\MQIPT!QM1\ssl\key.kdb -stashed  
-label ibmwebspheremqmqipt.qm1 -dn CN=MQIPT.QM1
```

如需相關資訊，請參閱 [在 AIX, Linux, and Windows 上建立自簽個人憑證](#)。

- c) 從佇列管理程式金鑰儲存庫擷取憑證。
輸入下列指令，以擷取佇列管理程式憑證：

```
runmqakm -cert -extract -db C:\ProgramData\IBM\MQ\mqgrs\MQIPT!QM1\ssl\key.kdb -stashed  
-label ibmwebspheremqmqipt.qm1  
-target C:\ProgramData\IBM\MQ\mqgrs\MQIPT!QM1\ssl\mqipt.qm1.crt -format ascii
```

如需相關資訊，請參閱 [從 AIX, Linux, and Windows 上的金鑰儲存庫擷取自簽憑證的公用部分](#)。

- d) 將擷取的憑證檔案複製到執行 MQIPT 的系統。
- e) 發出下列 MQSC 指令，以變更 MQIPT.CONN.CHANNEL 伺服器連線通道使用 TLS：

```
ALTER CHANNEL(MQIPT.CONN.CHANNEL) CHLTYPE(SVRCONN) TRPTYPE(TCP)  
SSLCIPH(ANY_TLS12_OR_HIGHER)
```

3. 在 MQIPT 執行所在的系統上，完成下列步驟，以將 MQIPT 路徑配置為使用 TLS。
 - a) 輸入下列指令，以建立 MQIPT 的 PKCS #12 金鑰儲存庫：

```
mqiptKeycmd -keydb -create -db C:\mqiptHome\ssl\mqipt.p12 -pw password -type pkcs12
```

其中 *password* 是金鑰儲存庫密碼。

- b) 在您的步驟 [第 158 頁](#) 的『3.a』中建立的 MQIPT 金鑰儲存庫中，建立 MQIPT 的個人憑證。

輸入下列指令，以針對 MQIPT 建立標籤為 `mqiptycert` 的新自簽憑證：

```
mqiptykeycmd -cert -create -db C:\mqiptyHome\ssl\mqipty.p12 -pw password -type pkcs12
-label mqiptycert -dn "CN=MQIPT Test Certificate"
```

其中 `password` 是您在步驟 [第 158 頁的『3.a』](#) 中建立金鑰儲存庫時指定的金鑰儲存庫密碼。

- c) 將用戶端憑證及佇列管理程式憑證新增至 MQIPT 金鑰儲存庫。

輸入下列指令，將憑證新增至 MQIPT 金鑰儲存庫：

```
mqiptykeycmd -cert -add -db C:\mqiptyHome\ssl\mqipty.p12 -pw password -type pkcs12 -label
clientcert
-file client.crt -format ascii
mqiptykeycmd -cert -add -db C:\mqiptyHome\ssl\mqipty.p12 -pw password -type pkcs12 -label
qm1cert
-file mqipty.qm1.crt -format ascii
```

其中 `password` 是金鑰儲存庫密碼，`client.crt` 是您在步驟 [第 158 頁的『1.c』](#) 中建立的用戶端憑證檔，而 `mqipty.qm1.crt` 是您在步驟 [第 158 頁的『2.c』](#) 中建立的佇列管理程式憑證。

- d) 從金鑰儲存庫擷取 MQIPT 憑證。

輸入下列指令，以擷取 MQIPT 憑證：

```
mqiptykeycmd -cert -extract -db C:\mqiptyHome\ssl\mqipty.p12 -pw password -type pkcs12
-label mqiptycert -target C:\mqiptyHome\ssl\mqipty.crt -format ascii
```

其中 `password` 是金鑰儲存庫密碼。

- e) 將擷取的憑證檔案複製到用戶端執行所在的系統，以及 IBM MQ 伺服器執行所在的系統。

- f) 輸入下列指令，以加密 MQIPT 金鑰儲存庫密碼：

```
mqiptyPW
```

當系統提示時，請輸入您在步驟 [第 158 頁的『3.a』](#) 中建立金鑰儲存庫時指定的金鑰儲存庫密碼。

- g) 編輯 `mqipty.conf` 檔案，並新增下列路徑定義：

```
[route]
ListenerPort=1415
Destination=server1.company2.com
DestinationPort=1414
SSLServer=true
SSLServerKeyRing=C:\mqiptyHome\ssl\mqipty.p12
SSLServerKeyRingPW=encrypted_password
SSLClient=true
SSLClientKeyRing=C:\mqiptyHome\ssl\mqipty.p12
SSLClientKeyRingPW=encrypted_password
```

其中 `encrypted_password` 是在步驟 [第 159 頁的『3.f』](#) 中執行 `mqiptyPW` 指令所建立的已加密金鑰儲存庫密碼。

4. 將 MQIPT 憑證新增至用戶端金鑰儲存庫及佇列管理程式金鑰儲存庫。

- a) 在執行用戶端的系統上，輸入下列指令，將 MQIPT 憑證新增至用戶端金鑰儲存庫：

```
runmqakm -cert -add -db C:\ProgramData\IBM\MQ\clientkey.kdb -stashed
-label mqiptycert -file mqipty.crt -format ascii
```

其中 `mqipty.crt` 是您在步驟 [第 159 頁的『3.d』](#) 中建立的 MQIPT 憑證檔案。

- b) 在執行 IBM MQ 伺服器的系統上，輸入下列指令，以將 MQIPT 憑證新增至佇列管理程式金鑰儲存庫：

```
runmqakm -cert -add -db C:\ProgramData\IBM\MQ\qmgrs\MQIPT!QM1\ssl\key.kdb -stashed
-label mqiptycert -file mqipty.crt -format ascii
```

其中 `mqipty.crt` 是您在步驟 [第 159 頁的『3.d』](#) 中建立的 MQIPT 憑證檔案。

如需相關資訊，請參閱 [在 AIX, Linux, and Windows 系統上，將 CA 憑證 \(或自簽憑證的公用部分\) 新增至金鑰儲存庫。](#)

5. 在執行 MQIPT 的系統上，開啟命令提示字元，並輸入下列指令來啟動 MQIPT：

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
```

其中 C:\mqiptHome 指出 MQIPT 配置檔 mqipt.conf 的位置， ipt1 是要提供給 MQIPT 實例的名稱。

下列訊息指出 MQIPT 已順利啟動：

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt1
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1415 is starting and will forward messages to :
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....using MQ protocol
MQCPI036 ....SSL Client side enabled with properties :
MQCPI139 .....secure socket protocols <NULL>
MQCPI031 .....cipher suites <NULL>
MQCPI032 .....key ring file C:\mqiptHome\ssl\mqipt.p12
MQCPI047 .....CA key ring file <NULL>
MQCPI071 .....site certificate uses
UID=*,CN=*,T=*,OU=*,DC=*,O=*,STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI038 .....peer certificate uses
UID=*,CN=*,T=*,OU=*,DC=*,O=*,STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI037 ....SSL Server side enabled with properties :
MQCPI139 .....secure socket protocols <NULL>
MQCPI031 .....cipher suites <NULL>
MQCPI032 .....key ring file C:\mqiptHome\ssl\mqipt.p12
MQCPI047 .....CA key ring file <NULL>
MQCPI071 .....site certificate uses
UID=*,CN=*,T=*,OU=*,DC=*,O=*,STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI038 .....peer certificate uses
UID=*,CN=*,T=*,OU=*,DC=*,O=*,STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI033 .....client authentication set to false
MQCPI078 Route 1415 ready for connection requests
```

6. 在 IBM MQ 用戶端系統上的命令提示字元中，輸入下列指令以執行 TLS 範例程式：

```
AMQSSSLC -m MQIPT.QM1 -c MQIPT.CONN.CHANNEL -x 10.9.1.2(1415)
          -k "C:\ProgramData\IBM\MQ\clientkey" -l clientcert -s ANY_TLS12_OR_HIGHER
```

下列訊息指出應用程式已順利連接至佇列管理程式：

```
Connection established to queue manager MQIPT.QM1
```

配置 HTTP 通道作業

在此實務範例中，您可以透過 HTTP 測試 MQIPT 的兩個實例之間的簡式連線。

開始之前

在開始使用此實務範例之前，請確定您已完成 [第 146 頁的『開始使用 IBM MQ Internet Pass-Thru』](#) 中列出的必要作業。

關於這項作業

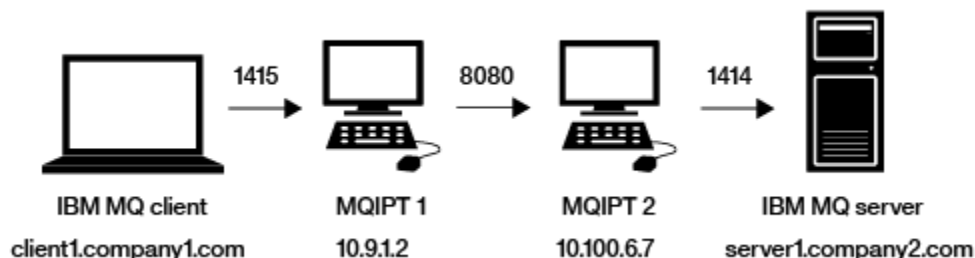


圖 34: HTTP 通道作業網路圖

此圖顯示從 IBM MQ 用戶端 (在埠 1415 上稱為 client1.company1.com) 到兩個 MQIPT 實例的連線，透過 HTTP 來隧道連線，最後到 IBM MQ 伺服器 (在埠 1414 上稱為 server1.company2.com)。

程序

若要在兩個 MQIPT 實例之間配置 HTTP 通道作業，請完成下列步驟：

1. 在 MQIPT 1 系統上：

a) 編輯 mqipt.conf 並新增下列路由定義：

```
[route]
ListenerPort=1415
Destination=10.100.6.7
DestinationPort=8080
HTTP=true
HTTPServer=10.100.6.7
HTTPServerPort=8080
```

b) 開啟命令提示字元，並啟動 MQIPT：

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
```

其中 C:\mqiptHome 指出 MQIPT 配置檔 mqipt.conf 的位置，ipt1 是要提供給 MQIPT 實例的名稱。

下列訊息指出 MQIPT 已順利啟動：

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt1
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1415 is starting and will forward messages to :
MQCPI034 ...10.100.6.7(8080)
MQCPI035 ...using HTTP
MQCPI066 ...and HTTP server at 10.100.6.7(8080)
MQCPI078 Route 1415 ready for connection requests
```

2. 在 MQIPT 2 系統上：

a) 編輯 mqipt.conf 並新增下列路由定義：

```
[route]
ListenerPort=8080
Destination=Server1.company2.com
DestinationPort=1414
```

b) 開啟命令提示字元，並啟動 MQIPT：

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt2
```

其中 C:\mqiptHome 指出 MQIPT 配置檔 mqipt.conf 的位置，ipt2 是要提供給 MQIPT 實例的名稱。

下列訊息指出 MQIPT 已順利啟動：

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt2
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 8080 is starting and will forward messages to :
MQCPI034 ...Server1.company2.com(1414)
MQCPI035 ...using MQ protocols
MQCPI078 Route 8080 ready for connection requests
```

3. 在 IBM MQ 用戶端上的命令提示字元中，輸入下列指令：

a) 設定 MQSERVER 環境變數：

```
SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
```

b) 放置訊息:

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1  
Hello world
```

在鍵入訊息字串之後按 Enter 鍵兩次。

c) 取得訊息:

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
```

即會傳回訊息 "Hello world"。

配置存取控制

在此實務範例中，您可以透過使用 Java security manager 在 MQIPT 接聽器埠上新增安全檢查，將 MQIPT 設定為僅接受來自特定用戶端的連線。

開始之前

- 在開始使用此實務範例之前，請確定您已完成 [第 146 頁的『開始使用 IBM MQ Internet Pass-Thru』](#) 中列出的必要作業。

關於這項作業

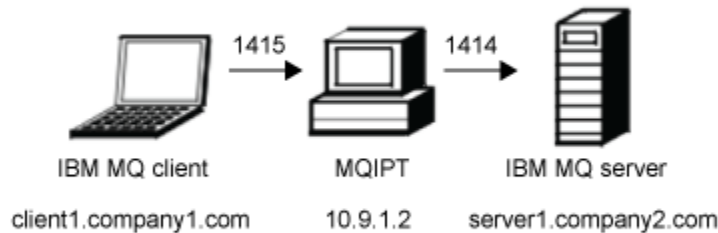


圖 35: 存取控制網路圖

此圖顯示從 IBM MQ 用戶端 (在埠 1415 上稱為 `client1.company1.com`) 到 MQIPT 到 IBM MQ 伺服器 (在埠 1414 上稱為 `server1.company2.com`) 的連線。

程序

若要配置存取控制，請完成下列步驟:

1. 設定 MQIPT:

- 在命令提示字元中輸入下列指令，將範例 Java security manager 原則複製到 MQIPT 起始目錄:

```
copy C:\mqipt\samples\mqiptSample.policy C:\mqiptHome\mqipt.policy
```

- 使用下列指令來啟動「原則工具」公用程式:

```
C:\mqipt\java\jre\bin\policytool
```

- 按一下 **檔案 > 開啟**，然後選取 `C:\mqiptHome\mqipt.policy`。
- 按一下 **編輯原則項目**，然後從下列項目變更 CodeBase:

```
file:/C:/Program Files/IBM/IBM MQ Internet Pass-Thru/lib/com.ibm.mq.ipt.jar
```

至:

```
file:/C:/mqipt/lib/com.ibm.mq.ipt.jar
```

- e) 從下列位置變更 IBM MQ Internet Pass-Thru、errors 及 logs 目錄的檔案許可權:

```
C:\Program Files\IBM\IBM MQ Internet Pass-Thru
```

至:

```
C:\mqiptHome
```

- f) 將其他檔案許可權從:

```
C:\Program Files\IBM\IBM MQ Internet Pass-Thru
```

至:

```
C:\mqipt
```

- g) 按一下 **新增許可權**

請依照下列的說明, 完成各欄位:

許可權: java.net.SocketPermission

目標: client1.company1.com:1024-

動作: accept、listen、resolve

- h) 按一下 **檔案 > 儲存**, 以儲存對原則檔案所做的變更。

- i) 編輯 mqipt.conf。

- i) 將下列兩個內容新增至 [global] 區段:

```
SecurityManager=true  
SecurityManagerPolicy=C:\mqiptHome\mqipt.policy
```

- ii) 新增下列路由定義:

```
[route]  
ListenerPort=1415  
Destination=server1.company2.com  
DestinationPort=1414
```

2. 啟動 MQIPT:

開啟命令提示字元, 並輸入下列指令:

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
```

其中 C:\mqiptHome 指出 MQIPT 配置檔 mqipt.conf 的位置, ipt1 是要提供給 MQIPT 實例的名稱。

下列訊息指出 MQIPT 已順利啟動:

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved  
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting  
MQCPI004 Reading configuration information from mqipt.conf  
MQCPI152 MQIPT name is ipt1  
MQCPI055 Setting the java.security.policy to C:\mqiptHome\mqipt.policy  
MQCPI053 Starting the Java Security Manager  
MQCPI021 Password checking has been enabled on the command port  
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files  
MQCPI006 Route 1415 has started and will forward messages to :  
MQCPI034 ....server1.company2.com(1414)  
MQCPI035 ....using MQ protocol  
MQCPI078 Route 1415 ready for connection requests
```

3. 在 IBM MQ 用戶端系統上的命令提示字元中, 輸入下列指令:

a) 設定 **MQSERVER** 環境變數:

```
SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
```

b) 放置訊息:

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1  
Hello world
```

在鍵入訊息字串之後按 Enter 鍵兩次。

c) 取得訊息:

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
```

即會傳回訊息 "Hello world"。

配置 SOCKS Proxy

在此實務範例中，您可以讓 MQIPT 充當 SOCKS Proxy。

開始之前

- 在開始使用此實務範例之前，請確定您已完成 [第 146 頁的『開始使用 IBM MQ Internet Pass-Thru』](#) 中列出的必要作業。
- 在整個 IBM MQ 電腦上啟用 SOCKS，或只在 IBM MQ 用戶端應用程式 **amqsputc** 及 **amqsgetc** 上啟用 SOCKS。
- 配置 SOCKS 用戶端，如下所示：
 1. 使用 MQIPT 作為 SOCKS Proxy。
 2. 啟用 SOCKS 5 支援。
 3. 停用使用者鑑別。
 4. 限制與 MQIPT 網址的連線。

關於這項作業

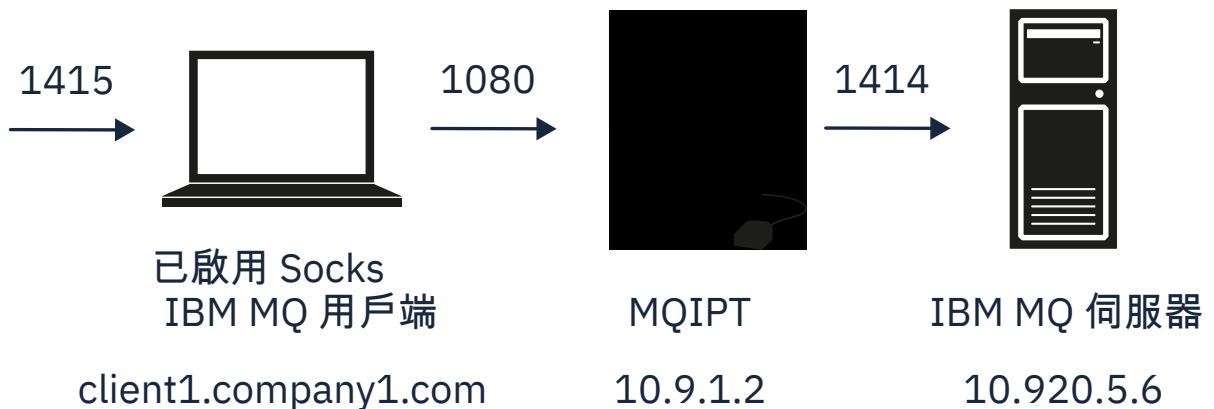


圖 36: SOCKS Proxy 網路圖

This diagram shows the connection flow from the IBM MQ client (called client1.company1.com on port 1415) through MQIPT to the IBM MQ server (called server1.company2.com on port 1414).

程序

若要配置 SOCKS Proxy，請完成下列步驟：

1. 配置並啟動 MQIPT：

- a) 編輯 `mqipt.conf` 並新增下列路由定義：

```
[route]
ListenerPort=1080
Destination=server1.company2.com
DestinationPort=1414
SocksServer=true
```

Destination 和 **DestinationPort** 路由內容的值會被忽略，因為在 SOCKS 信號交換程序期間，會從 IBM MQ 用戶端取得真正的目的地。

- b) 開啟命令提示字元，並啟動 MQIPT：

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
```

其中 `C:\mqiptHome` 指出 MQIPT 配置檔 `mqipt.conf` 的位置，`ipt1` 是要提供給 MQIPT 實例的名稱。

下列訊息指出 MQIPT 已順利啟動：

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt1
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1080 has started and will forward messages to :
MQCPI034 ...server1.company2.com(1414)
MQCPI035 ...using MQ protocol
MQCPI052 ...SOCKS server side enabled
MQCPI078 Route 1080 ready for connection requests
```

2. 在 IBM MQ 用戶端系統上的命令提示字元中，輸入下列指令：

- a) 設定 **MQSERVER** 環境變數：

```
SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.20.5.6(1414)
```

- b) 放置訊息：

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world
```

在鍵入訊息字串之後按 Enter 鍵兩次。

- c) 取得訊息：

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
```

即會傳回訊息 "Hello world"。

配置 SOCKS 用戶端

在此實務範例中，您可以使用現有的 SOCKS Proxy 來執行 MQIPT，就好像它已啟用 SOCKS 一樣。

這與實務範例第 164 頁的『[配置 SOCKS Proxy](#)』類似，不同之處在於 MQIPT 會建立啟用 SOCKS 的連線，而不是 IBM MQ 用戶端。

開始之前

在開始使用此實務範例之前，請確定您已完成 [第 146 頁的『開始使用 IBM MQ Internet Pass-Thru』](#) 中列出的必要作業

關於這項作業

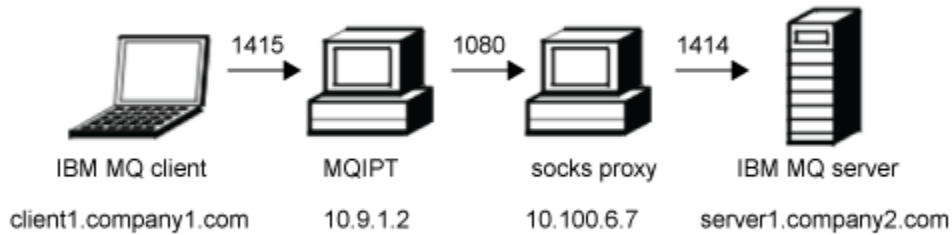


圖 37: SOCKS 用戶端網路圖

此圖顯示從 IBM MQ 用戶端 (在埠 1415 上稱為 client1.company1.com) 到 MQIPT，然後透過 SOCKS Proxy (在埠 1080 上) 到 IBM MQ 伺服器 (在埠 1414 上稱為 server1.company2.com) 的網路連線。

程序

若要配置 SOCKS 用戶端，請完成下列步驟：

1. 設定 MQIPT。

在 MQIPT 電腦上，編輯 `mqipt.conf` 並新增路徑定義：

```
[route]
ListenerPort=1415
Destination=server1.company2.com
DestinationPort=1414
SocksClient=true
SocksProxyHost=10.9.6.7
SocksProxyPort=1080
```

2. 啟動 MQIPT。

開啟命令提示字元並輸入：

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
```

其中 `C:\mqiptHome` 指出 MQIPT 配置檔 `mqipt.conf` 的位置，`ipt1` 是要提供給 MQIPT 實例的名稱。

下列訊息指出 MQIPT 已順利啟動：

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt1
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....using MQ protocol
MQCPI039 ....and SOCKS proxy at 10.9.6.7(1080)
MQCPI078 Route 1415 ready for connection requests
```

3. 在 IBM MQ 用戶端上的命令提示字元中，輸入下列指令：

a) 設定 **MQSERVER** 環境變數：

```
SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
```

b) 放置訊息：

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world
```

在鍵入訊息字串之後按 Enter 鍵兩次。

c) 取得訊息:

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
```

即會傳回訊息 "Hello world"。

配置 MQIPT 叢集作業支援

在此實務範例中，您可以設定叢集作業環境。

開始之前

- 在開始使用此實務範例之前，請確定您已完成 [第 146 頁的『開始使用 IBM MQ Internet Pass-Thru』](#) 中列出的必要作業。
- 在 IBM MQ 伺服器上倫敦：
 - 定義稱為「倫敦」的佇列管理程式。
 - 定義稱為 MQIPT.CONN.CHANNEL。
 - 已在埠 1414 上啟動適用於倫敦的 TCP/IP 接聽器。
 - SOCKS-已啟用佇列管理程式。
- 在 IBM MQ 伺服器上：
 - 定義名為 NEWNYK 的佇列管理程式。
 - 定義稱為 MQIPT.CONN.CHANNEL。
 - 在埠 1414 上啟動 NEWNYK 的 TCP/IP 接聽器。
 - SOCKS-已啟用佇列管理程式。

註：若要啟用 SOCKS-啟用佇列管理程式，請啟用整個電腦或只啟用 IBM MQ 伺服器應用程式。配置 SOCKS 用戶端，如下所示：

- 將用戶端指向 MQIPT 作為 SOCKS Proxy。
- 啟用 SOCKS V5 支援。
- 停用使用者鑑別。
- 僅建立與 MQIPT 的遠端連線。

關於這項作業

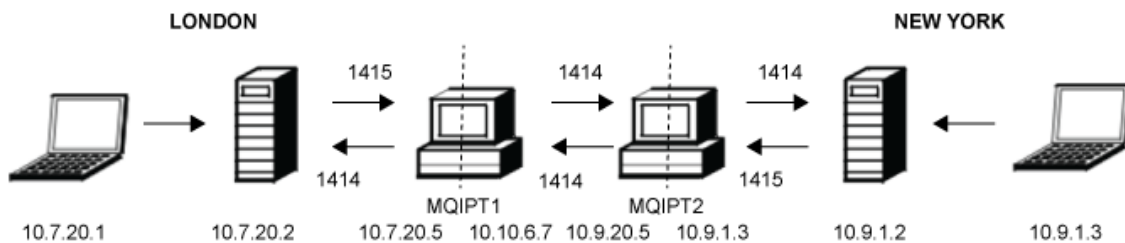


圖 38: 叢集作業網路圖

此圖顯示從 IBM MQ 用戶端透過 MQIPT 到 IBM MQ 伺服器的連線。

只有一個應用程式可以在相同電腦上的給定埠上接聽。如果埠 1414 已在使用中，請選擇可用埠並在範例中替換它。

然後，您可以測試佇列管理程式之間的路徑，方法是將訊息放置在倫敦伺服器上的本端佇列上，並從 NEWORK 伺服器擷取訊息。

程序

若要配置 MQIPT 叢集作業支援，請完成下列步驟：

1. 設定倫敦伺服器。

開啟命令提示字元，並輸入下列指令：

```
runmqsc
DEFINE CHANNEL(TO.LONDON) +
  CHLTYPE(CLUSRCVR) TRPTYPE(TCP) +
  CLUSTER(INVENTORY) +
  CONNAME('10.10.6.7(1414)')
DEFINE CHANNEL(TO.NEWYORK) +
  CHLTYPE(CLUSSDR) TRPTYPE(TCP) +
  CLUSTER(INVENTORY) +
  CONNAME('10.9.20.5(1414)')
```

2. 設定 NEWORK 伺服器

開啟命令提示字元，並輸入下列指令：

```
runmqsc
ALTER QMGR REPOS(INVENTORY)
DEFINE QLOCAL(MQIPT.LOCAL.QUEUE) +
  CLUSTER(INVENTORY)
DEFINE CHANNEL(TO.NEWYORK) +
  CHLTYPE(CLUSRCVR) TRPTYPE(TCP) +
  CLUSTER(INVENTORY) +
  CONNAME('10.9.20.5(1414)')
DEFINE CHANNEL(TO.LONDON) +
  CHLTYPE(CLUSSDR) TRPTYPE(TCP) +
  CLUSTER(INVENTORY) +
  CONNAME('10.10.6.7(1414)')
```

3. 設定 MQIPT 1.

編輯 mqipt.conf 並新增下列路由定義：

```
[route]
Name=LONDON to NEWYORK
ListenerPort=1415
Destination=10.9.20.5
DestinationPort=1414
SocksServer=true

[route]
Name=MQIPT1 to LONDON
ListenerPort=1414
Destination=10.7.20.2
DestinationPort=1414
```

4. 啟動 MQIPT 1.

開啟命令提示字元並輸入：

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
```

其中 C:\mqiptHome 指出 MQIPT 配置檔 mqipt.conf 的位置，ipt1 是要提供給 MQIPT 實例的名稱。

下列訊息指出 MQIPT 已順利啟動：

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt1
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....10.9.20.5(1414)
MQCPI035 ....using MQ protocol
MQCPI052 ....SOCKS server side enabled
MQCPI078 Route 1415 ready for connection requests
MQCPI006 Route 1414 has started and will forward messages to :
MQCPI034 ....10.7.20.2(1414)
```



```
MQCPI035 ....using MQ protocol
MQCPI078 Route 1414 ready for connection requests
```

5. 設定 MQIPT 2.

編輯 `mqipt.conf` 並新增下列路由定義:

```
[route]
Name=NEWYORK to LONDON
ListenerPort=1415
Destination=10.10.6.7
DestinationPort=1414
SocksServer=true

[route]
Name=MQIPT2 to NEWYORK
ListenerPort=1414
Destination=10.9.1.2
DestinationPort=1414
```

6. 啟動 MQIPT 2。

開啟命令提示字元，並輸入下列指令:

```
C:
cd \mqipt\bin
mqipt .. -n ipt2
```

其中 `..` 指出 MQIPT 配置檔 `mqipt.conf` 位於上層目錄中，`ipt2` 是要提供給 MQIPT 實例的名稱。

下列訊息指出 MQIPT 已順利啟動:

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt2
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqipt\logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....10.10.6.7(1414)
MQCPI035 ....using MQ protocol
MQCPI052 ....SOCKS server side enabled
MQCPI078 Route 1415 ready for connection requests
MQCPI006 Route 1414 has started and will forward messages to :
MQCPI034 ....10.9.1.2(1414)
MQCPI035 ....using MQ protocol
MQCPI078 Route 1414 ready for connection requests
```

7. 在倫敦 IBM MQ 用戶端 (10.7.20.1) 上的命令提示字元中，輸入下列指令:

a) 設定 **MQSERVER** 環境變數:

```
SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.7.20.2(1414)
```

b) 放置訊息:

```
amqsputc MQIPT.LOCAL.QUEUE LONDON
Hello world
```

在鍵入訊息字串之後按 Enter 鍵兩次。

這會導致倫敦佇列管理程式將訊息傳送至紐約佇列管理程式上的佇列。

8. 在紐約 IBM MQ 用戶端 (10.9.1.3) 上的命令提示字元中，輸入下列指令:

a) 設定 **MQSERVER** 環境變數:

```
SET MQSERVER=MQIPT.CONN.CHANNEL/TCP/10.9.1.2(1414)
```

b) 取得訊息:

```
amqsgetc MQIPT.LOCAL.QUEUE NEWYORK
```

即會傳回訊息 "Hello world"。

配置埠號

您可以控制建立送出連線時所使用的本端埠位址。例如，如果您的防火牆只容許特定範圍的埠號，則可以使用 MQIPT 來確保輸出源自有效埠。

開始之前

- 在開始使用此實務範例之前，請確定您已完成 [第 146 頁的『開始使用 IBM MQ Internet Pass-Thru』](#) 中列出的必要作業。
- 在多網卡電腦上安裝 MQIPT。

關於這項作業

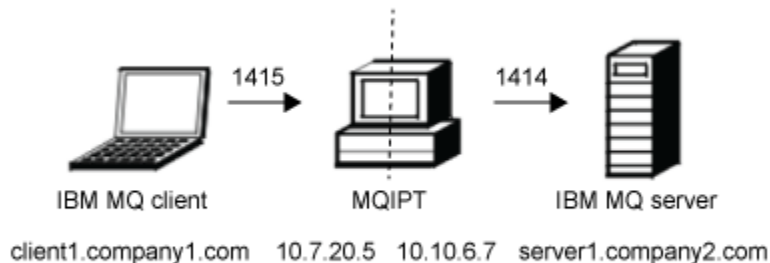


圖 39: 埠配置網路圖

此圖顯示從 IBM MQ 用戶端 (埠 1415 上的 client1.company1.com) 透過 MQIPT 到 IBM MQ 伺服器 (埠 1414 上的 server1.company2.com) 的連線。

程序

若要配置埠號，請完成下列步驟：

1. 設定 MQIPT。

編輯 mqipt.conf 並新增下列路由定義：

```
[route]
ListenerPort=1415
Destination=server1.company2.com
DestinationPort=1414
LocalAddress=10.10.6.7
OutgoingPort=2000
MaxConnectionThreads=20
```

2. 啟動 MQIPT。

在 IBM MQ 系統上開啟命令提示字元，並輸入下列指令：

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
```

其中 C:\mqiptHome 指出 MQIPT 配置檔 mqipt.conf 的位置，ipt1 是要提供給 MQIPT 實例的名稱。

下列訊息指出 MQIPT 已順利啟動：

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt1
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1415 is starting and will forward messages to :
MQCPI034 ....server1.company2.com(1414)
```

```
MQCPI035 ....using MQ protocol
MQCPI069 ....binding to local address 10.10.6.7 when making new connections
MQCPI070 ....using local port address range 2000-2019 when making new connections
MQCPI078 Route 1415 ready for connection requests
```

3. 在 IBM MQ 用戶端系統上的命令提示字元中，輸入下列指令：

a) 設定 **MQSERVER** 環境變數：

```
SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.7.20.5(1415)
```

b) 放置訊息：

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world
```

在鍵入訊息字串之後按 Enter 鍵兩次。

c) 取得訊息：

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
```

即會傳回訊息 "Hello world"。

使用 LDAP 伺服器擷取 CRL

您可以將 MQIPT 配置成使用 LDAP 伺服器來擷取憑證撤銷清冊 (CRL)。

開始之前

- 在開始使用此實務範例之前，請確定您已完成 [第 146 頁的『開始使用 IBM MQ Internet Pass-Thru』](#) 中列出的必要作業。
- 確保 MQIPT 2 具有由授信憑證管理中心 (CA) 發出的個人憑證，儲存在稱為 `myCert.pfx` 的金鑰環檔案中。
- 確保 MQIPT 1 具有授信 CA 憑證的副本，該憑證將用來鑑別 MQIPT 2 所傳送的憑證。此憑證儲存在稱為 `caCerts.pfx` 的金鑰環檔中。
- 用於存取金鑰環的密碼已使用 `mqiPTPW` 指令加密。

關於這項作業

在此實務範例中，您可以將 IBM MQ 用戶端連接至佇列管理程式 (QM)，並將 IBM MQ 訊息放在目標佇列上。在 MQIPT 1 上執行 MQIPT 追蹤將顯示正在使用的 LDAP 伺服器。

若要示範 CRL 如何運作，請確定授信 CA 已撤銷 MQIPT 2 所使用的個人憑證。然後，不容許 IBM MQ 用戶端連接至 QM，因為從 MQIPT 1 至 MQIPT 2 的連線遭到拒絕。

此實務範例無意說明如何安裝及設定 LDAP 伺服器，也無意說明如何建立包含個人或授信憑證的金鑰環檔案。它假設 LDAP 伺服器可從已知且受信任的 CA 使用。未使用備份 LDAP 伺服器，但可以透過新增適當的 Route 內容來實作。

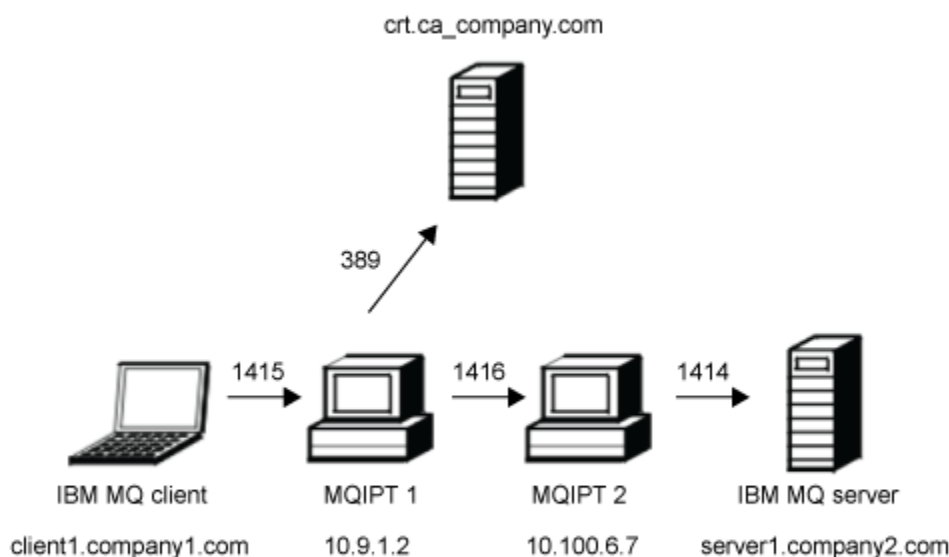


圖 40: LDAP 伺服器網路圖

此圖顯示透過兩個 MQIPT 實例，從 IBM MQ 用戶端 (埠 1415 上的 client1.company1.com) 到 IBM MQ 伺服器 (埠 1414 上的 server1.company2.com) 的連線。第一個 MQIPT 具有 LDAP 伺服器的連線 (埠 389 上的 crl.ca_company.com)。

程序

若要使用 LDAP 伺服器來擷取 CRL，請完成下列步驟：

1. 在 MQIPT 1 系統上：

a) 編輯 mqipt.conf 並新增下列路由定義：

```
[route]
ListenerPort=1415
Destination=10.100.6.7
DestinationPort=1416
SSLClient=true
SSLClientCAKeyRing=C:\mqiptHome\ssl\caCerts.pfx
SSLClientCAKeyRingPW=encrypted_key_ring_password
LDAP=true
LDAPServer1=crl.ca_company.com
LDAPServer1Timeout=4
```

其中 *encrypted_key_ring_password* 是使用 **mqiptPW** 指令加密之 caCerts.pfx 金鑰環的密碼。

b) 開啟命令提示字元，並啟動 MQIPT：

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
```

其中 C:\mqiptHome 指出 MQIPT 配置檔 mqipt.conf 的位置，ipt1 是要提供給 MQIPT 實例的名稱。

下列訊息指出 MQIPT 已順利啟動：

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
```

```

MQCPI152 MQIPT name is ipt1
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....10.100.6.7(1416)
MQCPI035 ...using MQ protocol
MQCPI036 ...SSL Client side enabled with properties :
MQCPI031 .....CipherSuites <NULL>
MQCPI032 .....key ring file <NULL>
MQCPI047 .....CA key ring file C:\mqiptHome\ssl\caCerts.pfx
MQCPI071 .....site certificate uses UID=*,CN=*,T=*,OU=*,DC=*,O=*,
                STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI038 .....peer certificate uses UID=*,CN=*,T=*,OU=*,DC=*,O=*,
                STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI075 ....LDAP main server at crl.ca\_company.com(389)
MQCPI086 .....timeout of 4 second(s)
MQCPI084 ....CRL cache expiry timeout is 1 hour(s)
MQCPI085 ....CRLs will be saved in the key-ring file(s)
MQCPI078 Route 1415 ready for connection requests

```

2. 在 MQIPT 2 系統上:

a) 編輯 mqipt.conf 並新增下列路由定義:

```

[route]
ListenerPort=1416
Destination=server1.company2.com
DestinationPort=1414
SSLServer=true
SSLServerKeyRing=C:\mqipt\ssl\myCert.pfx
SSLServerKeyRingPW=encrypted_key_ring_password

```

其中 `encrypted_key_ring_password` 是使用 `mqiptPW` 指令加密之 `myCert.pfx` 金鑰環的密碼。

b) 開啟命令提示字元，並啟動 MQIPT:

```

C:
cd \mqipt\bin
mqipt .. -n ipt2

```

其中 `..` 指出 MQIPT 配置檔 `mqipt.conf` 位於上層目錄中，`ipt2` 是要提供給 MQIPT 實例的名稱。

下列訊息指出順利完成:

```

5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt2
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqipt\logs will be used to store the log files
MQCPI006 Route 1416 is starting and will forward messages to :
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ...using MQ protocol
MQCPI037 ...SSL Server side enabled with properties :
MQCPI031 .....CipherSuites <NULL>
MQCPI032 .....key ring file C:\mqipt\ssl\myCert.pfx
MQCPI047 .....CA key ring file <NULL>
MQCPI071 .....site certificate uses UID=*,CN=*,T=*,OU=*,DC=*,O=*,
                STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI038 .....peer certificate uses UID=*,CN=*,T=*,OU=*,DC=*,O=*,
                STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI033 .....client authentication set to false
MQCPI078 Route 1416 ready for connection requests

```

3. 在 IBM MQ 用戶端系統上的命令提示字元中，輸入下列指令:

a) 設定 MQSERVER 環境變數:

```
SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
```

b) 放置訊息:

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world
```

在鍵入訊息字串之後按 Enter 鍵兩次。

c) 取得訊息:

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
```

即會傳回訊息 "Hello world"。

以 TLS Proxy 模式執行 MQIPT

您可以在 TLS Proxy 模式下執行 MQIPT，以便它接受來自 IBM MQ TLS 用戶端的 TLS 連線要求，並透過通道將它傳送至 IBM MQ TLS 伺服器。

開始之前

在開始使用此實務範例之前，請確定您已完成第 146 頁的『開始使用 IBM MQ Internet Pass-Thru』中列出的必要作業。

關於這項作業

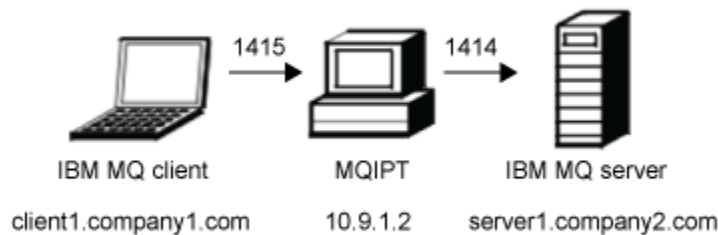


圖 41: SSL/TLS Proxy 模式網路圖

此圖顯示從 IBM MQ 用戶端 (埠 1415 上的 client1.company1.com) 到 MQIPT 到 IBM MQ 伺服器 (埠 1414 上的 server1.company2.com) 的連線流程。

如需為 IBM MQ 配置 TLS 的進一步資訊，請參閱 [使用 SSL/TLS](#)。

程序

若要以 TLS Proxy 模式執行 MQIPT，請完成下列步驟：

1. 將 IBM MQ 用戶端及伺服器配置為使用 TLS 連線。
 - a) 建立佇列管理程式的金鑰儲存庫。
如需相關資訊，請參閱 [在 AIX, Linux, and Windows 上設定金鑰儲存庫](#)。
 - b) 在 C:\ProgramData\IBM\MQ 目錄中建立用戶端的金鑰儲存庫。請將它稱為 *clientkey.kdb*。
 - c) 在您於步驟第 174 頁的『1.a』中建立的佇列管理程式金鑰儲存庫中，建立佇列管理程式的個人憑證。
如需相關資訊，請參閱 [在 AIX, Linux, and Windows 上建立自簽個人憑證](#)。
 - d) 在您於步驟第 174 頁的『1.b』中建立的用戶端金鑰儲存庫中，為用戶端建立個人憑證。
 - e) 從伺服器金鑰儲存庫擷取個人憑證，並將它新增至用戶端儲存庫。
如需相關資訊，請參閱 [從 AIX, Linux, and Windows 上的金鑰儲存庫擷取自簽憑證的公用部分](#)，以及在 [在 AIX, Linux, and Windows 系統上將 CA 憑證 \(或自簽憑證的公用部分\) 新增至金鑰儲存庫](#)。
 - f) 從用戶端金鑰儲存庫擷取個人憑證，並將它新增至伺服器金鑰儲存庫。
 - g) 變更 MQIPT.CONN.CHANNEL 伺服器連線通道：

```
ALTER CHANNEL(MQIPT.CONN.CHANNEL) CHLTYPE(SVRCONN) TRPTYPE(TCP)
SSLCPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
```

2. 若要以 TLS Proxy 模式執行 MQIPT，請完成下列步驟：

a) 編輯 `mqipt.conf` 並新增下列路由定義：

```
[route]
ListenerPort=1415
Destination=server1.company2.com
DestinationPort=1414
SSLProxyMode=true
```

b) 啟動 MQIPT。

開啟命令提示字元，並輸入下列指令：

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
```

其中 `C:\mqiptHome` 指出 MQIPT 配置檔 `mqipt.conf` 的位置，`ipt1` 是要提供給 MQIPT 實例的名稱。

下列訊息指出 MQIPT 已順利啟動：

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt1
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ...server1.company2.com(1414)
MQCPI035 ...using SSLProxyMode protocol
MQCPI078 Route 1415 ready for connection requests
```

3. 在 IBM MQ 用戶端系統上的命令提示字元中，輸入下列指令以執行 TLS 範例程式：

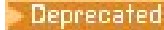
```
AMQSSSLC -m MQIPT.QM1 -c MQIPT.CONN.CHANNEL -x 10.9.1.2(1415)
          -k "C:\ProgramData\IBM\MQ\clientkey" -l cert_label -s
          TLS_RSA_WITH_AES_128_CBC_SHA256
```

其中 `cert_label` 是您在步驟 [第 174 頁的『1.d』](#) 中建立之用戶端憑證的標籤。

使用安全管理程式以 TLS Proxy 模式執行 MQIPT

您可以在 TLS Proxy 模式下執行 MQIPT，以便它接受來自 IBM MQ TLS 用戶端的 TLS 連線要求，並透過通道將它傳送至 IBM MQ TLS 伺服器。透過將安全管理程式與 MQIPT 搭配使用，您可以限制訊息可以傳送至的位址。

開始之前

註： **Deprecated** Java security manager 與 MQIPT 搭配使用已淘汰，因為在 Java 的未來版本中已淘汰 Java security manager 以供移除。

在開始使用此實務範例之前，請確定您已完成 [第 146 頁的『開始使用 IBM MQ Internet Pass-Thru』](#) 中列出的必要作業。

關於這項作業

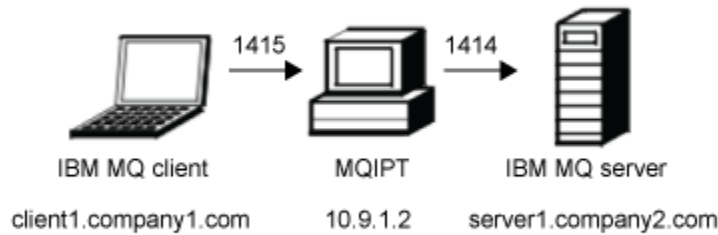


圖 42: SSL/TLS Proxy 模式網路圖

此圖顯示從 IBM MQ 用戶端 (埠 1415 上的 client1.company1.com) 到 MQIPT 到 IBM MQ 伺服器 (埠 1414 上的 server1.company2.com) 的連線流程。

如需為 IBM MQ 配置 TLS 的進一步資訊，請參閱 [使用 SSL/TLS](#)。

程序

若要使用安全管理程式以 TLS Proxy 模式執行 MQIPT，請完成下列步驟：

1. 將 IBM MQ 用戶端及伺服器配置為使用 TLS 連線。

a) 建立佇列管理程式的金鑰儲存庫。

如需相關資訊，請參閱 [在 AIX, Linux, and Windows 上設定金鑰儲存庫](#)。

b) 在 C:\ProgramData\IBM\MQ 目錄中建立用戶端的金鑰儲存庫。請將它稱為 *clientkey.kdb*。

c) 在您於步驟 [第 176 頁的『1.a』](#) 中建立的佇列管理程式金鑰儲存庫中，建立佇列管理程式的個人憑證。

如需相關資訊，請參閱 [在 AIX, Linux, and Windows 上建立自簽個人憑證](#)。

d) 在您於步驟 [第 176 頁的『1.b』](#) 中建立的用戶端金鑰儲存庫中，為用戶端建立個人憑證。

e) 從伺服器金鑰儲存庫擷取個人憑證，並將它新增至用戶端儲存庫。

如需相關資訊，請參閱 [從 AIX, Linux, and Windows 上的金鑰儲存庫擷取自簽憑證的公用部分](#)，以及在 [AIX, Linux, and Windows 系統上將 CA 憑證 \(或自簽憑證的公用部分\) 新增至金鑰儲存庫](#)。

f) 從用戶端金鑰儲存庫擷取個人憑證，並將它新增至伺服器金鑰儲存庫。

g) 變更 MQIPT.CONN.CHANNEL 伺服器連線通道：

```
ALTER CHANNEL(MQIPT.CONN.CHANNEL) CHLTYPE(SVRCONN) TRPTYPE(TCP)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
```

2. 在 MQIPT 電腦上 (請參閱圖表)，透過在命令提示字元中輸入下列指令，將範例 Java security manager 原則複製到 MQIPT 起始目錄：

```
copy C:\mqipt\samples\mqiptSample.policy C:\mqiptHome\mqipt.policy
```

3. 使用下列指令來啟動「原則工具」公用程式：

```
C:\mqipt\java\jre\bin\policytool
```

在原則工具中：

a) 按一下 **檔案 > 開啟**，然後選取 C:\mqiptHome\mqipt.policy。

b) 選擇：

```
file:/C:/Program Files/IBM/IBM MQ Internet Pass-Thru/lib/com.ibm.mq.ipt.jar
```


然後按一下 **編輯原則項目**

c) 將 CodeBase 從:

```
file:/C:/Program Files/IBM/IBM MQ Internet Pass-Thru/lib/com.ibm.mq.ipt.jar
```

至:

```
file:/C:/mqipt/lib/com.ibm.mq.ipt.jar
```

d) 從下列位置變更 IBM MQ Internet Pass-Thru、errors 及 logs 目錄的檔案許可權:

```
C:\Program Files\IBM\IBM MQ Internet Pass-Thru
```

至:

```
C:\mqiptHome
```

e) 將其他檔案許可權從:

```
C:\Program Files\IBM\IBM MQ Internet Pass-Thru
```

至:

```
C:\mqipt
```

f) 按一下 **新增許可權**

請依照下列的說明，完成各欄位:

許可權: java.net.SocketPermission

目標: client1.company1.com:1024-

動作: accept、listen、resolve

g) 按一下 **檔案 > 儲存**，以儲存對原則檔案所做的變更。

4. 編輯 mqipt.conf。將下列內容新增至 [global] 區段，並新增下列路徑定義:

```
[global]
SecurityManager=true
SecurityManagerPolicy=C:\mqiptHome\mqipt.policy

[route]
ListenerPort=1415
Destination=server1.company2.com
DestinationPort=1414
SSLProxyMode=true
```

5. 啟動 MQIPT。

開啟命令提示字元，並輸入下列指令:

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
```

其中 C:\mqiptHome 指出 MQIPT 配置檔 mqipt.conf 的位置，ipt1 是要提供給 MQIPT 實例的名稱。

下列訊息指出 MQIPT 已順利啟動:

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt1
MQCPI055 Setting the java.security.policy to C:\mqiptHome\mqipt.policy
MQCPI053 Starting the Java Security Manager
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\mqipt\logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....using SSLProxyMode protocol
MQCPI078 Route 1415 ready for connection requests
```

6. 在 IBM MQ 用戶端系統上的命令提示字元中，輸入下列指令以執行 TLS 範例程式：

```
AMQSSSLC -m MQIPT.QM1 -c MQIPT.CONN.CHANNEL -x 10.9.1.2(1415)
          -k "C:\ProgramData\IBM\MQ\clientkey" -l cert_label -s
          TLS_RSA_WITH_AES_128_CBC_SHA256
```

其中 cert_label 是您在步驟 [第 176 頁](#) 的『1.d』中建立之用戶端憑證的標籤。

使用安全結束程式

在此實務範例中，您可以使用提供的範例安全結束程式（稱為 SampleSecurity 結束程式），以便只容許用戶端連線使用以 MQIPT. 字元開頭的通道名稱。

開始之前

- 在開始使用此實務範例之前，請確定您已完成 [第 146 頁](#) 的『開始使用 IBM MQ Internet Pass-Thru』中列出的必要作業。
- 安裝 JavaJava 8.0 JDK。
- 將 Java bin 子目錄新增至 **PATH** 環境變數。

關於這項作業

此實務範例中使用的範例結束程式為 SampleSecurityExit.java。它隨 MQIPT 一起提供，位於 MQIPT 安裝目錄的 samples/exits 子目錄中。

如果您使用建議的伺服器連線通道名稱 MQIPT.CONN.CHANNEL（在大部分情況下使用），則容許完成用戶端連線，且可以將 IBM MQ 訊息放置在佇列上。

為了示範安全結束程式如預期般運作，請定義另一個伺服器連線通道，且其名稱不是以字元 MQIPT. 開頭（例如 TEST.CONN.CHANNEL），然後重試 **amqspuic** 指令，但已變更 **MQSERVER** 環境變數來使用新的通道名稱。此時會拒絕連線，並傳回 2059 (MQRC_Q_MGR_NOT_AVAILABLE) 錯誤。

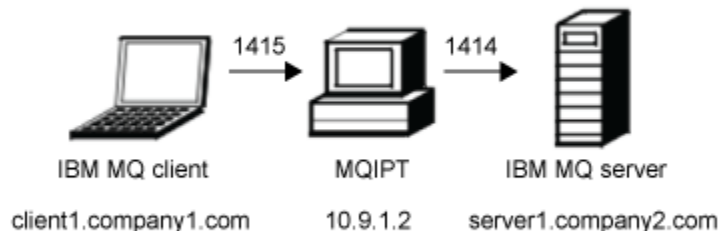


圖 43: 安全結束程式網路圖

This diagram shows the connection flow from the IBM MQ client (called client1.company1.com on port 1415) through MQIPT to the IBM MQ server (called server1.company2.com on port 1414).

程序

若要使用安全結束程式，請完成下列步驟：

1. 在 MQIPT 電腦上：
 - a) 透過在命令提示字元中發出下列指令，在 MQIPT 起始目錄中建立稱為 exits 的目錄：

```
md C:\mqiptHome\exits
```

- b) 輸入下列指令來編譯結束程式。如果您尚未變更結束碼，則不必這麼做，因為 MQIPT 會提供已編譯的範例結束程式。

```
C:
cd \mqipt\samples\exits
javac -classpath C:\mqipt\lib\com.ibm.mq.ipt.jar;. SampleSecurityExit.java
```

- c) 輸入下列指令，將已編譯的結束程式類別檔 `SampleSecurityExit.class` 複製到 `C:\mqiptHome\exits` 目錄：

```
copy C:\mqipt\samples\exits\SampleSecurityExit.class C:\mqiptHome\exits
```

- d) 編輯 `mqipt.conf` 並新增路徑定義：

```
[route]
ListenerPort=1415
Destination=server1.company2.com
DestinationPort=1414
SecurityExit=true
SecurityExitName=SampleSecurityExit
```

- e) 開啟命令提示字元，並啟動 MQIPT：

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
```

其中 `C:\mqiptHome` 指出 MQIPT 配置檔 `mqipt.conf` 的位置，`ipt1` 是要提供給 MQIPT 實例的名稱。

下列訊息指出 MQIPT 已順利啟動：

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt1
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....using MQ protocol
MQCPI079 ....using security exit C:\mqiptHome\exits\SampleSecurityExit
MQCPI080 .....and timeout of 30 seconds
MQCPI078 Route 1415 ready for connection requests
```

2. 在 IBM MQ 用戶端系統上的命令提示字元中，輸入下列指令：

- a) 設定 **MQSERVER** 環境變數：

```
SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
```

- b) 放置訊息：

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world
```

在鍵入訊息字串之後按 Enter 鍵兩次。

- c) 取得訊息：

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
```

即會傳回訊息 "Hello world"。

使用安全結束程式將用戶端連線要求遞送至 IBM MQ 佇列管理程式伺服器

在此實務範例中，您可以將用戶端連線要求以循環式方式動態遞送至由三個 IBM MQ 佇列管理程式伺服器組成的群組。群組中每一部伺服器上的佇列管理程式必須相同。

開始之前

- 在開始使用此實務範例之前，請確定您已完成第 146 頁的『開始使用 IBM MQ Internet Pass-Thru』中列出的必要作業。
- 安裝 Java 8.0 JDK。

- 將 Java bin 子目錄新增至 **PATH** 環境變數。

關於這項作業

此實務範例中使用的範例結束程式為 `SampleRoutingExit.java`。它隨 MQIPT 一起提供，位於 MQIPT 安裝目錄的 `samples/exits` 子目錄中。

已編譯結束程式類別檔的名稱及位置是使用 MQIPT **SecurityExitName** 及 **SecurityExitPath** 內容來定義。

從稱為 `SampleRoutingExit.conf` 的配置檔中讀取要使用的佇列管理程式及伺服器名稱清單。結束程式預期配置檔存在於與結束程式類別檔相同的目錄中。

第一次執行 **amqsputc** 指令時，IBM MQ 訊息會放置在 MQIPT.LOCAL.QUEUE 佇列。第二次執行時，訊息會放在第二部伺服器上的佇列上，依此類推。使用此設定時，**amqsgetc** 指令無法擷取剛剛放置在佇列上的訊息，因為 **amqsgetc** 指令所使用的用戶端連線要求會傳遞至清單中的下一個佇列。不過，執行 **amqsputc** 指令三次，接著執行三個 **amqsgetc** 指令，可確保以相同的順序擷取每一個訊息。

當然，透過使用另一個 IBM MQ 用戶端，直接連接至佇列管理程式 (亦即，不在此範例中使用 MQIPT)，您可以選擇性地從任何佇列管理程式擷取訊息。

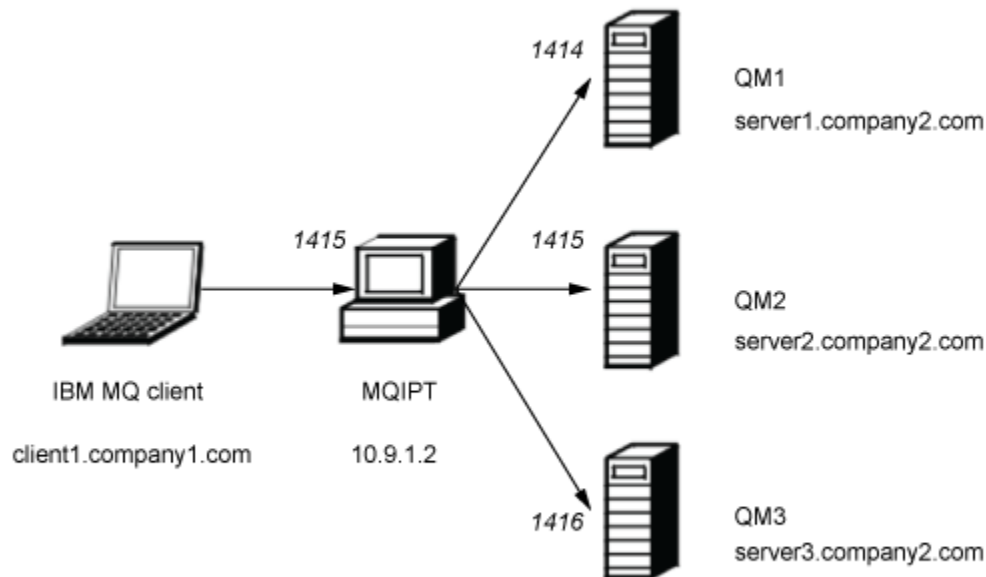


圖 44: 遞送安全結束程式網路圖

此圖顯示從 IBM MQ 用戶端 (在埠 1415 上稱為 client1.company1.com) 到 MQIPT 的三個 IBM MQ 伺服器 (稱為 server1.company2.com、server2.company2.com 及 server3.company2.com) 的連線流程。

程序

若要使用安全結束程式，將用戶端連線要求循序遞送至三個不同的 IBM MQ 佇列管理程式伺服器，請完成下列步驟：

1. 建立三個名為 MQIPT.QM1。
每一個佇列管理程式都有一個名為 MQIPT.CONN.CHANNEL 及名為 MQIPT.LOCAL.QUEUE。
2. 在 MQIPT 伺服器上：

- a) 透過在命令提示字元中發出下列指令，在 MQIPT 起始目錄中建立稱為 `exits` 的目錄：

```
md C:\mqiptHome\exits
```

- b) 在 `C:\mqiptHome\exits` 目錄 (其中 `C:\mqiptHome` 是 `mqipt.conf` 檔案所在的目錄) 中，建立一個稱為 `SampleRoutingExit.conf` 的範例配置檔，其中包含三個佇列管理程式的名稱。

例如，配置檔可能包含下列項目：

```
server1.company2.com:1414
server2.company2.com:1415
server3.company2.com:1416
```

請確定檔案中第一個項目之前沒有空白行，且每一個項目都是有效的伺服器名稱。如果您已使用不同的伺服器名稱，請變更這些名稱以符合您的環境。

- c) 開啟命令提示字元，並輸入下列指令來編譯結束程式。如果您尚未變更結束碼，則不必這麼做，因為 MQIPT 會提供已編譯的範例結束程式。

```
C:
cd \mqipt\samples\exits
javac -classpath C:\mqipt\lib\com.ibm.mq.ipt.jar;. SampleRoutingExit.java
```

- d) 輸入下列指令，將已編譯的結束程式類別檔 `SampleRoutingExit.class` 複製到 `C:\mqiptHome\exits` 目錄：

```
copy C:\mqipt\samples\exits\SampleRoutingExit.class C:\mqiptHome\exits
```

- e) 編輯 `mqipt.conf` 並新增路徑定義：

```
[route]
ListenerPort=1415
Destination=server1.company2.com
DestinationPort=1414
SecurityExit=true
SecurityExitPath=C:\mqiptHome\exits
SecurityExitName=SampleRoutingExit
```

請注意，如果您將 `SampleRoutingExit.conf` 放置在預設 `C:\mqiptHome\exits` 目錄中，則不需要設定 **SecurityExitPath**。

- f) 啟動 MQIPT。

開啟命令提示字元，並輸入下列指令：

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
```

其中 `C:\mqiptHome` 指出 MQIPT 配置檔 `mqipt.conf` 的位置，`ipt1` 是要提供給 MQIPT 實例的名稱。

下列訊息指出 MQIPT 已順利啟動：

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt1
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ...server1.company2.com(1414)
MQCPI035 ...using MQ protocol
MQCPI079 ...using security exit C:\mqiptHome\exits\SampleRoutingExit
MQCPI080 .....and timeout of 30 seconds
MQCPI078 Route 1415 ready for connection requests
```

3. 在 IBM MQ 用戶端系統上的命令提示字元中，輸入下列指令：

- a) 設定 **MQSERVER** 環境變數：

```
SET MQSERVER=MQIPT.CONN.CHANNEL/TCP/10.9.1.2(1415)
```

- b) 放置三則訊息：

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world 1
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world 2
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world 3
```

在鍵入每一個訊息字串之後，請按 Enter 鍵兩次。

c) 取得訊息:

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
```

即會傳回訊息 Hello world 1、Hello world 2 及 Hello world 3。

動態遞送用戶端連線要求

在此實務範例中，您可以根據所使用通道的名稱，將用戶端連線要求動態遞送至目標伺服器。

開始之前

- 在開始使用此實務範例之前，請確定您已完成 [第 146 頁的『開始使用 IBM MQ Internet Pass-Thru』](#) 中列出的必要作業。
- 安裝 Java 8.0 JDK。
- 將 Java bin 子目錄新增至 **PATH** 環境變數。

關於這項作業

如果您使用佇列管理程式的名稱作為通道名稱的第一部分，則只需要使用一個 MQIPT 路徑來處理所有連線要求。例如，若要連接至 QM1，SVRCONN 通道的名稱可能是 QM1.MQIPT.CHANNEL。

此實務範例中使用的範例結束程式為 `SampleOneRouteExit.java`。它隨 MQIPT 一起提供，位於 MQIPT 安裝目錄的 `samples/exits` 子目錄中。

已編譯結束程式類別檔的名稱及位置是使用 MQIPT **SecurityExitName** 及 **SecurityExitPath** 內容來定義。

從稱為 `SampleOneRouteExit.conf` 的配置檔中讀取要使用的佇列管理程式及伺服器名稱清單。結束程式預期配置檔存在於與結束程式類別檔相同的目錄中。

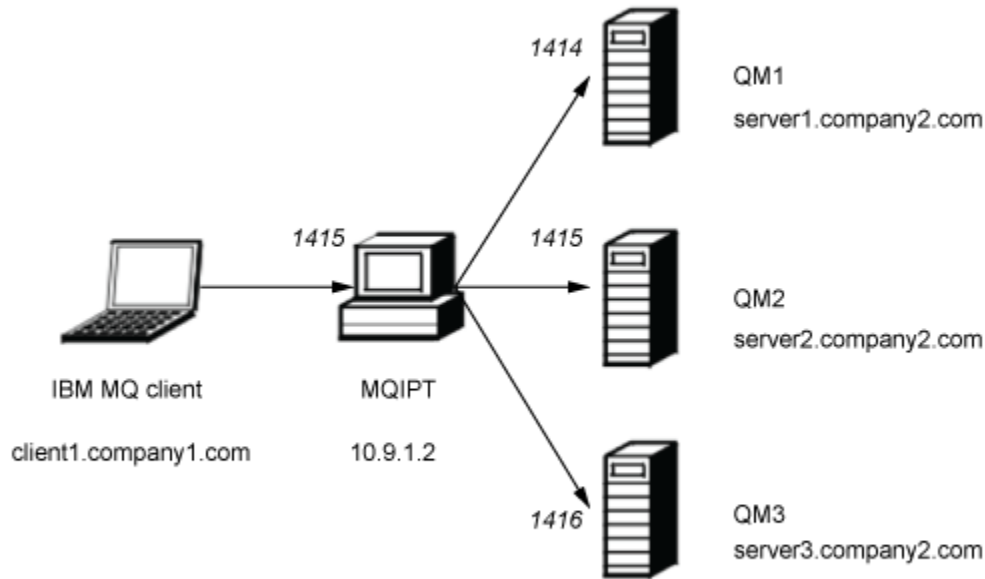


圖 45: 動態單一路由結束程式網路圖

此圖顯示從 IBM MQ 用戶端 (在埠 1415 上稱為 client1.company1.com) 到 MQIPT 的三個 IBM MQ 伺服器 (稱為 server1.company2.com、server2.company2.com 及 server3.company2.com) 的連線流程。

程序

若要動態遞送用戶端連線要求，請完成下列步驟：

1. 在三個個別伺服器上建立三個不同的佇列管理程式。
 每一個佇列管理程式都有一個以本身命名的 SVRCONN 通道 (例如，佇列管理程式 QM1 上的 QM1.MQIPT.CHANNEL)，以及一個名為 MQIPT.LOCAL.QUEUE 的空本端佇列。
2. 在 MQIPT 伺服器上：
 - a) 透過在命令提示字元中發出下列指令，在 MQIPT 起始目錄中建立稱為 exits 的目錄：

```
md C:\mqiptHome\exits
```

- b) 在 C:\mqiptHome\exits 目錄 (其中 C:\mqiptHome 是 mqipt.conf 檔案所在的目錄) 中，建立一個稱為 SampleOneRouteExit.conf 的範例配置檔，其中包含三個佇列管理程式的名稱。

例如，配置檔可能包含下列項目：

```
server1.company2.com:1414
server2.company2.com:1415
server3.company2.com:1416
```

請確定檔案中第一個項目之前沒有空白行，且每一個項目都是有效的伺服器名稱。如果您已使用不同的伺服器名稱，請變更這些名稱以符合您的環境。

請注意，清單中的所有佇列管理程式名稱都必須是唯一的。如果您多次列出相同名稱，即使佇列管理程式位於不同伺服器上，也只會登錄該名稱的最後一個項目。

- c) 開啟命令提示字元，並輸入下列指令來編譯結束程式。如果您尚未變更結束碼，則不必這麼做，因為 MQIPT 會提供已編譯的範例結束程式。

```
C:
cd \mqipt\samples\exits
javac -classpath C:\mqipt\lib\com.ibm.mq.ipt.jar;. SampleOneRouteExit.java
```

- d) 輸入下列指令，將已編譯的結束程式類別檔 `SampleOneRouteExit.class` 複製到 `C:\mqiptHome\exits` 目錄:

```
copy C:\mqipt\samples\exits\SampleOneRouteExit.class C:\mqiptHome\exits
```

- e) 編輯 `mqipt.conf` 並新增下列路由定義:

```
[route]
ListenerPort=1415
Destination=server1.company2.com
DestinationPort=1414
SecurityExit=true
SecurityExitName=SampleOneRouteExit
```

- f) 開啟命令提示字元，並啟動 MQIPT:

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
```

其中 `C:\mqiptHome` 指出 MQIPT 配置檔 `mqipt.conf` 的位置，`ipt2` 是要提供給 MQIPT 實例的名稱。

下列訊息指出 MQIPT 已順利啟動:

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt2
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1415 has started and will forward messages to :
MQCPI034 ....server1.company2.com(1414)
MQCPI035 ....using MQ protocol
MQCPI079 ....using security exit C:\mqiptHome\exits\SampleOneRouteExit
MQCPI080 .....and timeout of 5 seconds
MQCPI078 Route 1415 ready for connection requests
```

3. 在 IBM MQ 用戶端系統上的命令提示字元中，輸入下列指令:

- a) 設定 **MQSERVER** 環境變數:

```
SET MQSERVER=QM1.MQIPT.CHANNEL/TCP/10.9.1.2(1415)
```

- b) 放置訊息:

```
amqsputc MQIPT.LOCAL.QUEUE QM1
Hello world 1
```

在鍵入訊息字串之後按 Enter 鍵兩次。

當 `SVRCONN` 通道名稱以 `QM1` 開頭時，訊息會由 MQIPT 遞送至 `QM1`。

- c) 從 `QM1`: 取得訊息

```
amqsgetc MQIPT.LOCAL.QUEUE QM1
```

即會傳回訊息 `Hello world 1`。

- d) 重設 **MQSERVER** 環境變數:

```
SET MQSERVER=QM2.MQIPT.CHANNEL/TCP/10.9.1.2(1415)
```

- e) 放置訊息:

```
amqsputc MQIPT.LOCAL.QUEUE QM2
Hello world 2
```

在鍵入訊息字串之後按 Enter 鍵兩次。

當 SVRCONN 通道名稱以 QM2 開頭時，MQIPT 會將訊息遞送至 QM2。

f) 從 QM2: 取得訊息

```
amqsgetc MQIPT.LOCAL.QUEUE QM2
```

即會傳回訊息 Hello world 2。

g) 重新重設 MQSERVER 環境變數:

```
SET MQSERVER=QM3.MQIPT.CHANNEL/TCP/10.9.1.2(1415)
```

h) 放置訊息:

```
amqsputc MQIPT.LOCAL.QUEUE QM3  
Hello world 3
```

在鍵入訊息字串之後按 Enter 鍵兩次。

當 SVRCONN 通道名稱以 QM3 開頭時，訊息會由 MQIPT 遞送至 QM3。

i) 從 QM3: 取得訊息

```
amqsgetc MQIPT.LOCAL.QUEUE QM3
```

即會傳回訊息 Hello world 3。

使用憑證結束程式來鑑別 TLS 伺服器

在此實務範例中，您可以使用憑證結束程式來鑑別 TLS 連線。

開始之前

- 在開始使用此實務範例之前，請確定您已完成 [第 146 頁的『開始使用 IBM MQ Internet Pass-Thru』](#) 中列出的必要作業。
- 安裝 Java 8.0 JDK。
- 將 Java bin 子目錄新增至 **PATH** 環境變數。

關於這項作業

此實務範例執行與 [第 152 頁的『鑑別 TLS 伺服器』](#) 實務範例相同的功能，並新增憑證結束程式。

此實務範例中使用的範例結束程式為 `SampleCertificateExit.java`。它隨 MQIPT 一起提供，位於 MQIPT 安裝目錄的 `samples/exits` 子目錄中。

透過變更 **SSLExitData** 內容的值，可以容許或拒絕兩個 MQIPT 伺服器之間的 TLS 連線。

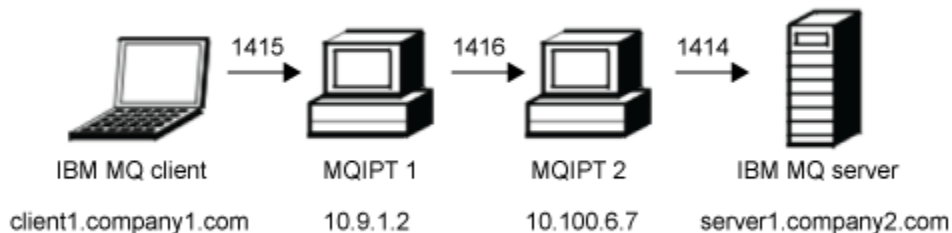


圖 46: SSL/TLS 伺服器網路圖

此圖顯示透過兩個 MQIPT 實例，從 IBM MQ 用戶端 (在埠 1415 上稱為 `client1.company1.com`) 到 IBM MQ 伺服器 (在埠 1414 上稱為 `server1.company2.com`) 的連線。

程序

若使用憑證結束程式來鑑別 TLS 伺服器，請完成下列步驟：

1. 在 MQIPT 1 系統上：

- a) 透過在命令提示字元中發出下列指令，在 MQIPT 起始目錄中建立稱為 `exits` 的目錄：

```
md C:\mqiptHome\exits
```

- b) 開啟命令提示字元，並輸入下列指令來編譯結束程式。如果您尚未變更結束碼，則不必這麼做，因為 MQIPT 會提供已編譯的範例結束程式。

```
C:  
cd \mqipt\samples\exits  
javac -classpath C:\mqipt\lib\com.ibm.mq.ipt.jar;. SampleCertificateExit.java
```

- c) 輸入下列指令，將已編譯的結束程式類別檔 `SampleCertificateExit.class` 複製到 `C:\mqiptHome\exits` 目錄：

```
copy C:\mqipt\samples\exits\SampleCertificateExit.class C:\mqiptHome\exits
```

- d) 編輯 `mqipt.conf` 並新增下列路由定義：

```
[route]  
ListenerPort=1415  
Destination=9.100.6.7  
DestinationPort=1416  
SSLClient=true  
SSLClientKeyRing=C:\mqipt\samples\ssl\sslSample.pfx  
SSLClientKeyRingPW=<mqiptPW>!PCaB1HWrFMOp43ngjwgArg==!6N/vsbqru7iqMhFN+wozxQ==  
SSLClientExit=true  
SSLExitName=SampleCertificateExit  
SSLExitPath=C:\mqiptHome\exits  
SSLExitData=allow
```

- e) 開啟命令提示字元，並啟動 MQIPT：

```
C:\mqipt\bin\mqipt C:\mqiptHome -n ipt1
```

其中 `C:\mqiptHome` 指出 MQIPT 配置檔 `mqipt.conf` 的位置，`ipt1` 是要提供給 MQIPT 實例的名稱。

下列訊息指出 MQIPT 已順利啟動：

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved  
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting  
MQCPI004 Reading configuration information from mqipt.conf  
MQCPI152 MQIPT name is ipt1  
MQCPI021 Password checking has been enabled on the command port  
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files  
MQCPI006 Route 1415 has started and will forward messages to :  
MQCPI034 ...9.100.6.7(1416)  
MQCPI035 ...using MQ protocol  
MQCPI036 ...SSL Client side enabled with properties :  
MQCPI031 .....CipherSuites <null>  
MQCPI032 .....keyring file C:\mqipt\samples\ssl\sslSample.pfx  
MQCPI047 .....CA keyring file <null>  
MQCPI038 .....peer certificate uses UID=*,CN=*,T=*,OU=*,DC=*,O=*,  
STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*  
MQCPI129 .....using certificate exit C:\mqiptHome\exits\SampleCertificateExit  
MQCPI131 .....and certificate exit data 'allow'  
MQCPI078 Route 1415 ready for connection requests
```

2. 在 MQIPT 2 系統上：

- a) 編輯 `mqipt.conf` 並新增下列路由定義：

```
[route]  
ListenerPort=1416  
Destination=Server1.company2.com  
DestinationPort=1414  
SSLServer=true
```

```
SSLServerKeyRing=C:\mqipt\samples\ssl\sslSample.pfx
SSLServerKeyRingPW=C:\mqipt\samples\ssl\sslSample.pwd
```

b) 開啟命令提示字元，並啟動 MQIPT:

```
C:
cd \mqipt\bin
mqipt .. -n ipt2
```

其中 .. 指出 MQIPT 配置檔 mqipt.conf 位於上層目錄中，ipt2 是要提供給 MQIPT 實例的名稱。

下列訊息指出 MQIPT 已順利啟動:

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is ipt2
MQCPI021 Password checking has been enabled on the command port
MQCPI011 The path C:\mqipt\logs will be used to store the log files
MQCPI006 Route 1416 has started and will forward messages to :
MQCPI034 ...server1.company2.com(1414)
MQCPI035 ...using MQ protocol
MQCPI037 ...SSL Server side enabled with properties :
MQCPI031 .....CipherSuites <null>
MQCPI032 .....key ring file C:\mqipt\samples\ssl\sslSample.pfx
MQCPI047 .....CA key ring file <null>
MQCPI038 .....peer certificate uses UID=*,CN=*,T=*,OU=*,DC=*,O=*,
STREET=*,L=*,ST=*,PC=*,C=*,DNQ=*
MQCPI033 .....client authentication set to false
MQCPI078 Route 1416 ready for connection requests
```

3. 在 IBM MQ 用戶端系統上的命令提示字元中，輸入下列指令:

a) 設定 MQSERVER 環境變數:

```
SET MQSERVER=MQIPT.CONN.CHANNEL/tcp/10.9.1.2(1415)
```

b) 放置訊息:

```
amqsputc MQIPT.LOCAL.QUEUE MQIPT.QM1
Hello world
```

在鍵入訊息字串之後按 Enter 鍵兩次。

c) 取得訊息:

```
amqsgetc MQIPT.LOCAL.QUEUE MQIPT.QM1
```

即會傳回訊息 "Hello world"。

IBM MQ 和 Apache Kafka 專門處理傳訊光譜的不同層面，其中一個是連線功能，另一個是資料，因此解決方案通常需要資料在這兩者之間流動。您可以使用 Kafka Connect 來達成此目的。

Kafka Connect 提供一個架構，可將資料從外部系統移至 Kafka 叢集，或從 Kafka 叢集移至外部系統。這是由連接器所達成。

有許多不同類型的連接器可用，且 IBM 提供連接器以與 IBM MQ 搭配使用。連接器有兩種不同的類型:

- 來源連接器會將資料從外部系統傳送至 Kafka。

IBM MQ 來源連接器會耗用來自 IBM MQ 佇列的訊息，並將它們以事件形式發佈至 Kafka 主題。

- 接收槽連接器會將資料從 Kafka 傳送至外部系統。

IBM MQ 接收槽連接器會耗用來自 Kafka 主題的事件，並將它們當作訊息傳送至 MQ 佇列。

如需相關資訊，請參閱 [Kafka 連接及連接器](#)。

Kafka Connect 實務範例可能包括:

- 以 IBM MQ 作為連線功能骨幹的核心銀行系統。您想要取得透過 IBM MQ 移動的訊息副本，並將它們推送至 Kafka 以進行分析
- 您想要延伸核心銀行系統以將資料發出至 Kafka，但只想要在銀行交易順利完成時輸入 Kafka，因此請使用 IBM MQ 作為交易式橋接器
- 您需要將資料從 Multiplatforms 取得至 z/OS。多平台開發團隊具有 Kafka 的經驗，z/OS 團隊想要利用 IBM MQ 與 CICS / IMS 的整合

從 IBM MQ 9.3.3 開始，如果您的企業具有 IBM MQ Advanced for z/OS VUE 授權、IBM MQ Advanced for Multiplatforms 授權或 IBM MQ Appliance 授權，則您可以取得所提供 IBM 的存取權，以及受支援的來源及接收槽連接器。

LTS **V 9.3.4** 如果您具有 IBM MQ Advanced 授權，則這也適用於 IBM MQ 9.3.4 或已套用 APAR PH56722 的 Long Term Support。

先前，您可以免費取得連接器而不提供支援，或取得 IBM Event Streams 授權的支援。

附註：

1. 這些方法可以與 Kafka 的任何變式 (例如，Apache Kafka 和 IBM Event Streams) 搭配使用。
2. 僅支援兩個 IBM 連接器，而不支援 Kafka Connect 架構本身。

V 9.3.3 **MQ Adv.** **MQ Adv. VUE** **Kafka 連接一般拓撲**

本節說明透過 IBM 連接器整合 IBM MQ 與 Kafka 時可以使用的三種方法。

如需取得連接器的相關資訊，請參閱 [第 194 頁的『取得連接器』](#)；如需佇列管理程式連線及配置選項的相關資訊，請參閱 [第 194 頁的『使用連接器』](#)。

直接到佇列 (來源)

想要使用 IBM MQ 將資料傳送至 Kafka 的應用程式可以將那些訊息傳送至 IBM MQ 來源連接器所使用的佇列。然後，IBM MQ 來源連接器會取得這些訊息，並將它們傳送至相關 Kafka 主題。

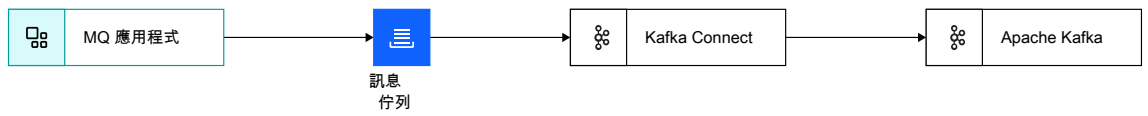


圖 47: 直接到佇列 (來源)

當應用程式需要將資料傳送至 Kafka，且該資料尚未傳送至 IBM MQ 時，應使用此方法。

使用 IBM MQ 傳送資料表示可以在與其他更新項目 (例如資料庫) 協調的交易內完成訊息傳送。此方法也可避免需要設定與 Kafka 的可能短暫存活連線，並改為使用與 IBM MQ 的現有連線。

串流佇列副本 (來源)

在許多情況下，需要取得在 IBM MQ 中移動的現有資料副本，並將其傳送至 Kafka (例如，用於分析)。從 IBM MQ 9.3 開始，可以使用串流佇列輕鬆達成此目的。串流佇列可讓佇列管理程式將訊息放入一個佇列中複製到第二個佇列，而不會影響使用第一個佇列的應用程式。如需相關資訊，請參閱第 27 頁的『串流佇列』。

例如：

```
DEF QL(TO.APP) STREAMQ(TO.KAFKA) STRMQOS(MUSTDUP)
DEF QL(TO.KAFKA)
```

表示當訊息傳送至 TO.APP 時，必須將該訊息的副本傳送至 TO.KAFKA。然後，IBM MQ 來源連接器會從 TO.KAFKA 並將它們傳送至相關 Kafka 主題。

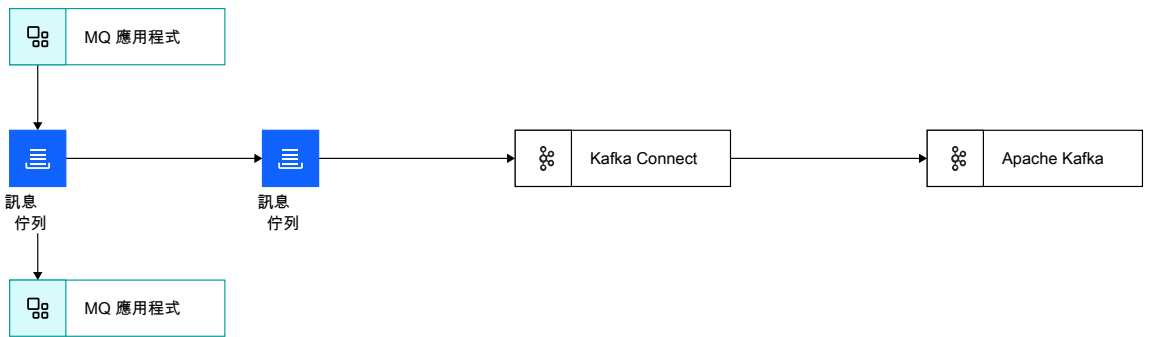


圖 48: 串流佇列副本 (來源)

啟用串流佇列對現有應用程式沒有影響，因為原始訊息不會變更。傳送至第二個佇列的訊息與具有相同有效負載、訊息 ID、相關性 ID 等的原始訊息相同。

直接到佇列 (接收槽)

與來源連接器一樣，可以配置接收槽連接器將資料從 Kafka 主題直接接收到佇列中。

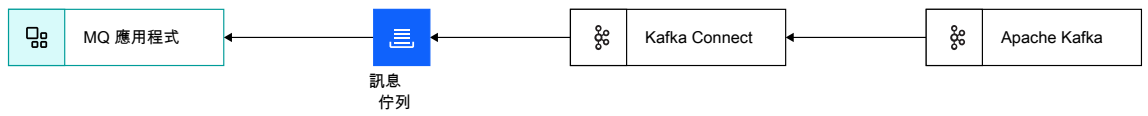


圖 49: 直接到佇列 (接收槽)

透過 IBM MQ 接收資料表示可以在與其他更新項目 (例如資料庫) 協調的交易內完成接收訊息。
此方法也可避免需要設定與 Kafka 的潛在短暫存活連線，並改為使用與 IBM MQ 的現有連線。

取得連接器

隨 IBM MQ 提供的連接器版本會隨時間變更。IBM MQ 9.3.3 隨附的第 1 版連接器提供至少一次訊息遞送。

V 9.3.4 IBM MQ 9.3.4 隨附的第 2 版連接器至少提供一次及正好一次訊息遞送。

V 9.3.4 如需至少一次遞送與僅一次遞送之間的差異，以及如何配置僅一次遞送的相關資訊，請參閱第 196 頁的『正好一次支援』。

在 IBM MQ Advanced for z/OS Value Unit Edition 中，**LTS** **V 9.3.4** 以及從 IBM MQ Advanced for z/OS 中的 IBM MQ 9.3.4 或已套用 APAR PH56722 的 Long Term Support 中，連接器及其範例提供於 z/OS UNIX System Services (USS) 中「連接器套件」元件的 kafka-connect 目錄中。

在 IBM MQ Advanced for Multiplatforms 和 IBM MQ Appliance 中，可以透過登入 Fix Central 並搜尋 V.R.M.F-IBM-MQ-Kafka-Connectors.tar.gz (例如 9.3.3.0-IBM-MQ-Kafka-Connectors.tar.gz) 來擷取這些連接器和必要的配置檔。

MQ Adv. **MQ Adv. VUE** **MQ Adv. z/OS** 以下是每一個 IBM MQ 版本隨附的連接器：

IBM MQ 版本號碼	IBM MQ for Multiplatforms tar 檔名	來源連接器版本	接收槽連接器版本	精確一次遞送支援
V 9.3.3 9.3.3	9.3.3.0-IBM-MQ-Kafka_Connectors.tar.gz	1.3.2	1.5.0	否
V 9.3.4 9.3.4	9.3.4.0-IBM-MQ-Kafka_Connectors.tar.gz	2.0	2.0	是

您也可以透過發行頁面從 GitHub 取得第 1 版連接器：

來源

[Kafka-connect-mq-source](#)

接收槽

[Kafka-connect-mq-sink](#)

請注意，IBM 支援從這些鏈結取得的任何版本。

附註：

- 不過，您應該一律採用連接器的最新版本，並定期檢查是否有更新項目。IBM MQ 隨附的連接器是產品出貨時的最新連接器，並定期更新至最新版本。
- 如果透過 IBM MQ Advanced for z/OS Value Unit Edition、IBM MQ Advanced for Multiplatforms 或 IBM MQ Appliance 授權提供 IBM MQ 連接器支援，則連接器必須連接至使用該授權執行的佇列管理程式。
- 早於 IBM MQ 9.3.3 的 IBM MQ Advanced for z/OS Value Unit Edition、IBM MQ Advanced for Multiplatforms 或 IBM MQ Appliance 版本也提供 IBM MQ 連接器的授權，但連接器必須從前述文字中參照的 GitHub 中的版本頁面下載。
- V 9.3.4** IBM MQ 9.3.4 之前的 IBM MQ Advanced for z/OS Continuous Delivery 版本也提供 IBM MQ 連接器的授權，但連接器必須從前述文字中所參照的 GitHub 中的版本頁面下載。

使用連接器

連接器是使用內容或 JSON 檔案來配置。範例檔隨連接器一起提供。

如需配置選項以及如何設定連接器的詳細資料，請參閱：

來源連接器: [Kafka IBM MQ 的連接來源連接器](#)

接收槽連接器: Kafka IBM MQ 的連接接收槽連接器

V 9.3.4 如果要在來源連接器中確切啟用一次支援，請參閱 [執行 MQ 來源連接器](#)，對於接收槽連接器，請參閱 [執行 MQ 接收槽連接器](#)。

V 9.3.4 如需至少一次遞送與僅一次遞送之間的差異，以及如何配置僅一次遞送的相關資訊，請參閱 [第 196 頁的『正好一次支援』](#)。

若要讓 Kafka Connect 執行 IBM MQ 連接器，它必須在其類別路徑上具有連接器 JAR 及各種 IBM MQ Jar 檔。需要下列 Jar 檔：

```
jms.jar  
com.ibm.mq.allclient.jar  
org.json.jar  
V 9.3.5 bcpkix-jdk18on.jar (從 IBM MQ 9.3.5)  
bcpkix-jdk15to18.jar (IBM MQ 9.3.3 和 IBM MQ 9.3.4)  
V 9.3.5 bcprov-jdk18on.jar (從 IBM MQ 9.3.5)  
bcprov-jdk15to18.jar (IBM MQ 9.3.3 和 IBM MQ 9.3.4)  
V 9.3.5 bcutil-jdk18on.jar (從 IBM MQ 9.3.5)  
bcutil-jdk15to18.jar (IBM MQ 9.3.3 和 IBM MQ 9.3.4)
```

例如：

來源連接器

V 9.3.5 從 IBM MQ 9.3.5:

```
export CLASSPATH=$CLASSPATH:/path-to-kafka-jars/kafka-connect-mq-source-1.3.2.jar:  
/path-to-mq-jars/jms.jar:/path-to-mq-jars/com.ibm.mq.allclient.jar:/path-to-mq-jars/org.json.jar:  
/path-to-mq-jars/bcpkix-jdk18on.jar:/path-to-mq-jars/bcprov-jdk18on.jar:/path-to-mq-jars/bcutil-  
jdk18on.jar
```

IBM MQ 9.3.3 和 IBM MQ 9.3.4

```
export CLASSPATH=$CLASSPATH:/path-to-kafka-jars/kafka-connect-mq-source-1.3.2.jar:  
/path-to-mq-jars/jms.jar:/path-to-mq-jars/com.ibm.mq.allclient.jar:/path-to-mq-jars/org.json.jar:  
/path-to-mq-jars/bcpkix-jdk15to18.jar:/path-to-mq-jars/bcprov-jdk15to18.jar:/path-to-mq-jars/bcutil-  
jdk15to18.jar
```

接收槽連接器

V 9.3.5 從 IBM MQ 9.3.5:

```
export CLASSPATH=$CLASSPATH:/path-to-kafka-jars/kafka-connect-mq-sink-1.5.0.jar:  
/path-to-mq-jars/jms.jar:/path-to-mq-jars/com.ibm.mq.allclient.jar:/path-to-mq-jars/org.json.jar:  
/path-to-mq-jars/bcpkix-jdk18on.jar:/path-to-mq-jars/bcprov-jdk18on.jar:/path-to-mq-jars/bcutil-  
jdk18on.jar
```

IBM MQ 9.3.3 和 IBM MQ 9.3.4

```
export CLASSPATH=$CLASSPATH:/path-to-kafka-jars/kafka-connect-mq-sink-1.5.0.jar:  
/path-to-mq-jars/jms.jar:/path-to-mq-jars/com.ibm.mq.allclient.jar:/path-to-mq-jars/org.json.jar:  
/path-to-mq-jars/bcpkix-jdk15to18.jar:/path-to-mq-jars/bcprov-jdk15to18.jar:/path-to-mq-jars/bcutil-  
jdk15to18.jar
```

其中：

path-to-kafka-jars 是 IBM MQ 連接器安裝所在位置的路徑
path-to-mq-jars 是 IBM JMS 用戶端安裝位置的路徑。

Z/OS 如果在 z/OS 上執行，則在「連接器套件」元件中，USS_ROOT/kafka-connect/source/kafka-connect-mq-source.jar 會指向來源連接器的最新版本，而 USS_ROOT/kafka-connect/sink/kafka-connect-mq-sink.jar 會指向接收槽連接器的最新版本。

Kafka Connect 和 IBM MQ 連接器可以在任何具有 Java 虛擬機器的平台上執行。它們不需要在與佇列管理程式或它們所連接的 Kafka 叢集相同的平台上執行。

不過，如果佇列管理程式與 Kafka 叢集之間有很長的距離，您應該將連接器定位在相對接近佇列管理程式的位置；最好位於相同的可用性區域或資料中心。

在 z/OS 上使用連接器

z/OS

所有平台 (包括 z/OS) 上執行的佇列管理程式都完全支援連接器。z/OS 佇列管理程式的連線可以透過伺服器連線通道，或透過本端連結。

在 IBM z/OS 和 IBM MQ for z/OS 上的效能測試環境中，透過在 z/OS UNIX System Services (USS) 中的 z/OS 上執行連接器，並使用本端連結連接至佇列管理程式，已取得最佳效能。這裡提供這些發現項目的詳細資料: [Kafka Connectors for IBM MQ - MQ for z/OS 視景](#)。

執行 Kafka Connect in USS on z/OS 需要一些額外的設定步驟; 這裡提供這些步驟的說明文件: [在 IBM z/OS 上執行連接器](#)。

MQ Adv.

V 9.3.4

MQ Adv. VUE

MQ Adv. z/OS

正好一次支援

IBM MQ Kafka 連接器有兩個版本: 1 和 2。第 2 版連接器提供僅一次及至少一次訊息遞送的支援，而第 1 版連接器則提供至少一次訊息遞送的支援。

至少一次訊息遞送表示在失敗的情況下，在 IBM MQ、IBM MQ Kafka 連接器或 Kafka 中:

- 對於來源連接器，IBM MQ 訊息不會遺失，但可能會多次遞送至 Kafka，導致 Kafka 訊息重複。
- 對於接收槽連接器，Kafka 訊息不會遺失，但可能會多次遞送至 IBM MQ，導致 IBM MQ 訊息重複。

正好一次訊息遞送表示在失敗的情況下，在 IBM MQ、IBM MQ Kafka 連接器或 Kafka 中:

- 對於來源連接器，IBM MQ 訊息不會遺失，且會遞送至 Kafka，且不會有重複的 Kafka 訊息。
- 對於接收槽連接器，Kafka 訊息不會遺失，且會遞送至 IBM MQ，且不會有重複的 IBM MQ 訊息。

正好一次支援僅在 IBM MQ 隨附的第 2 版連接器中提供，或在 IBM Event Streams 中提供。它在第 1 版連接器中無法使用。

第 2 版連接器可以在至少一次或完全一次模式下執行。透過適當的 Kafka 配置及利用「狀態佇列」來啟用支援正好一次。在正好一次模式中執行的連接器的每一個實例都需要自己的狀態佇列。

以精確一次模式執行之連接器的傳輸量及可調整性，小於以至少一次模式執行。如果您的應用程式不是設計成使用重複訊息，則只會啟用正好一次模式。

如需在來源連接器中配置正好一次模式的詳細資料，請參閱 [執行 MQ 來源連接器](#)，以及 [執行 MQ 接收槽連接器](#)，以取得在接收槽連接器中配置正好一次模式的詳細資料。

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提及的各項產品、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。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。

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：

Intellectual Property Licensing
Legal and Intellectual Property Law
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。

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。

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，在其他作業環境下取得的結果可能大不相同。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。本書的使用者應依自己的特定環境，查證適用的資料。

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 未來動向的任何陳述，僅代表 IBM 的目標而已，並可能於未事先聲明的情況下有所變動或撤回。

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、服務性或功能。

若貴客戶正在閱讀本項資訊的電子檔，可能不會有照片和彩色說明。

程式設計介面資訊

程式設計介面資訊 (如果有提供的話) 旨在協助您建立與此程式搭配使用的應用軟體。

本書包含預期程式設計介面的相關資訊，可讓客戶撰寫程式以取得 WebSphere MQ 的服務。

不過，本資訊也可能包含診斷、修正和調整資訊。提供診斷、修正和調整資訊，是要協助您進行應用軟體的除錯。

重要：請勿使用此診斷、修改及調整資訊作為程式設計介面，因為它可能會變更。

商標

IBM、IBM 標誌 ibm.com 是 IBM Corporation 在全球許多適用範圍的商標。IBM 商標的最新清單可在 Web 的 "Copyright and trademark information" www.ibm.com/legal/copytrade.shtml 中找到。其他產品及服務名稱可能是 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 (<https://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: