

7.5

規劃 *IBM WebSphere MQ*

**IBM**

## 附註

使用本資訊及其支援的產品之前，請先閱讀第 133 頁的『[注意事項](#)』中的資訊。

除非新版中另有指示，否則此版本適用於 IBM® WebSphere MQ 7.5 版及所有後續版次與修訂。

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

© Copyright International Business Machines Corporation 2007, 2024.

# 目錄

<b>規劃.....</b>	<b>5</b>
IBM MQ 及 IBM MQ Appliance 內部部署對於 GDPR 就緒狀態的考量.....	5
設計 IBM WebSphere MQ 架構.....	12
單一佇列管理程式架構.....	13
多個佇列管理程式架構.....	14
點對點傳訊.....	17
發佈/訂閱簡介.....	17
使用無法傳送郵件的佇列處理程式處理未遞送的訊息.....	91
規劃多個安裝.....	99
選擇主要安裝.....	100
規劃儲存體及效能需求.....	104
磁碟空間需求.....	104
規劃檔案系統支援.....	105
IBM WebSphere MQ 及 UNIX System V IPC 資源.....	128
AIX 上的共用記憶體.....	129
設定 UNIX 處理程序優先順序值.....	129
在 HP Integrity NonStop Server 上規劃 IBM WebSphere MQ 用戶端環境.....	129
準備 HP Integrity NonStop Server 環境.....	129
IBM WebSphere MQ 和 HP NonStop TMF.....	130
使用 HP NonStop TMF.....	130
<b>注意事項.....</b>	<b>133</b>
程式設計介面資訊.....	134
商標.....	134



## 規劃

---

規劃 IBM WebSphere MQ 環境時，您必須考量您要配置的 IBM WebSphere MQ 架構、資源需求、記載需求及備份機能。使用本主題中的鏈結，以規劃 IBM WebSphere MQ 執行所在的環境。

在規劃 IBM WebSphere MQ 環境之前，請先熟悉基本 IBM WebSphere MQ 概念，請參閱 [技術概觀](#) 中的主題。

### 相關概念

[可用性、回復及重新啟動](#)

### 相關工作

[移轉](#)

[安裝](#)

[配置](#)

[管理 WebSphere MQ](#)

[確定訊息未遺失 \(記載\)](#)

## IBM MQ 及 IBM MQ Appliance 內部部署對於 GDPR 就緒狀態的考量

---

### 適用於下列 PID：

- 5724-H72 IBM MQ
- 5655-AV9 IBM MQ Advanced for z/OS
- 5655-AV1 IBM MQ Advanced for z/OS, 價值單位版本
- 5655-AM9 IBM MQ Advanced Message Security for z/OS
- 5725-Z09 IBM MQ Appliance M2001
- 5725-S14 IBM MQ Appliance M2000
- 5655-MQ9 IBM MQ for z/OS
- 5655-VU9 IBM MQ for z/OS Value Unit Edition
- 5639-L92 IBM MQ Internet Pass-Thru
- 5655-MF9 IBM MQ Managed File Transfer for z/OS
- 5655-ADV IBM WebSphere MQ Advanced for z/OS
- 5655-AMS IBM WebSphere MQ Advanced Message Security for z/OS
- 5724-R10 IBM WebSphere MQ 適用於 Multiplatforms 的檔案傳送版本
- 5724-A39 IBM WebSphere MQ for HP NonStop Server
- 5724-A38 IBM WebSphere MQ for HP OpenVMS
- 5655-W97 IBM WebSphere MQ for z/OS
- 5655-VU8 IBM WebSphere MQ for z/OS Value Unit Edition
- 5655-VUE IBM WebSphere MQ for z/OS Value Unit Edition
- 5725-C79 IBM WebSphere MQ Hypervisor Edition for Red Hat Enterprise Linux® for x86
- 5725-F22 IBM WebSphere MQ Hypervisor for AIX
- 5655-MFT IBM WebSphere MQ z/OS

### 注意事項：

本文件目的在協助您準備 GDPR 備妥。它提供您可以配置的 IBM MQ 特性的相關資訊，以及您應該考量以協助組織備妥 GDPR 的產品使用層面。本資訊不是完整清單，這是因為客戶可以選擇許多方式來配置功能，而且產品用法及搭配協力廠商應用程式和系統使用的方式多種多樣。

客戶負責確保他們自己符合各種法律法規，包括歐盟一般資料保護法規。就一切相關法律之確認與解釋，客戶應自行負責向法律顧問取得諮詢意見。該等確認與解釋可能影響客戶之業務，以及客戶為遵守該等法律古故所應採取之行為。

這裡描述的產品、服務及其他功能並不適合所有客戶情況，可能存在可用性限制。IBM 不提供法律、會計或審核意見，亦不聲明或保證其服務或產品可確保客戶符合任何法規。

## 目錄

1. [GDPR](#)
2. [GDPR 產品配置](#)
3. [資料生命週期](#)
4. [資料收集](#)
5. [資料儲存](#)
6. [資料存取](#)
7. [資料處理](#)
8. [資料刪除](#)
9. [資料監視](#)
10. [限制使用個人資料的功能](#)
11. [檔案處理](#)

## GDPR

歐盟 ("EU") 已採用《一般資料保護法 (GDPR)》，並自 2018 年 5 月 25 日起生效。

### 為什麼 GDPR 很重要？

GDPR 為處理個人資料建立了更強大的資料保護規範架構。GDPR 的特點：

- 新的和加強的個人權利
- 擴大的個人資料定義
- 處理者的新責任
- 可能對不合規行為採取的重大經濟處罰
- 強制通知資料洩漏

### 閱讀有關 GDPR 的更多內容：

- [EU GDPR 資訊入口網站](#)
- [ibm.com/GDPR 網站](http://ibm.com/GDPR)

## 產品配置 - GDPR 就緒的注意事項

下列各節提供配置 IBM MQ 的考量，以協助您的組織備妥 GDPR。

## 資料生命週期

IBM MQ 是交易式訊息導向中介軟體產品，可讓應用程式非同步交換應用程式提供的資料。IBM MQ 支援一系列傳訊 API、通訊協定及橋接器，以連接應用程式。因此，IBM MQ 可用來交換許多形式的資料，其中有些可能受限於 GDPR。IBM MQ 可能會與數個協力廠商產品交換資料。其中有些是 IBM 所擁有，但其他許多則是由其他技術供應商提供。軟體產品相容性報告網站提供相關聯軟體的清單。如需協力廠商產品的 GDPR 就緒狀態相關考量，您應該參閱該產品的說明文件。IBM MQ 管理者透過定義佇列、主題及訂閱，控制 IBM MQ 與透過它傳遞之資料的互動方式。

### 哪些類型的資料會流經 IBM MQ？

由於 IBM MQ 為應用程式資料提供非同步傳訊服務，因此這個問題沒有一個明確的答案，因為使用案例會因應用程式部署而異。應用程式訊息資料會持續保存在佇列檔 (z/OS 上的頁集或「連結機能」、日誌和保存

檔中，且訊息本身可能包含 GDPR 所控管的資料。針對問題判斷目的 (例如錯誤日誌、追蹤檔及 FFST) 所收集的檔案中也可能包含應用程式提供的訊息資料。在 z/OS 上，應用程式提供的訊息資料也可能包含在位址空間或「連結機能」傾出中。

以下是一些可能使用 IBM MQ 交換的一般個人資料範例：

- 客戶的員工 (例如: IBM MQ 可用來連接客戶的薪資或 HR 系統)
- 客戶自己的客戶個人資料 (例如: IBM MQ 可能由客戶用來在與其客戶相關的應用程式之間交換資料，例如取得銷售領導人並將資料儲存在其 CRM 系統內)。
- 客戶自己的客戶機密個人資料 (例如: IBM MQ 可能會在需要交換個人資料的產業環境定義內使用，例如整合臨床應用程式時的 HL7-based 醫療記錄)。

除了應用程式提供的訊息資料之外，IBM MQ 還會處理下列類型的資料：

- 鑑別認證 (例如使用者名稱和密碼、API 金鑰等)
- 技術可識別的個人資訊 (例如裝置 ID、基於使用的 ID、IP 位址等 - 當鏈結至個人時)

### 用於在線上與 IBM 聯絡的個人資料

IBM MQ 用戶端可以透過各種方式提交線上評論/意見回饋/要求，以聯絡 IBM MQ 主題，主要是：

- [IBM Developer](#) 上的 IBM MQ 區域 中頁面上的公開評論區域
- [IBM Documentation](#) 中的 IBM MQ 產品資訊 頁面上的公開評論區域
- [IBM 支援討論區](#) 中的公開評論
- [IBM RFE Community on IBM Developer](#) 中的公開評論

一般而言，只有客戶姓名及電子郵件位址才用來啟用聯絡主題的個人回覆，而且個人資料的使用符合 [IBM 線上隱私權聲明](#)。

## 資料收集

IBM MQ 可用來收集個人資料。在評量 IBM MQ 的使用以及滿足 GDPR 需求的需求時，您應該考量在您的情況下透過 IBM 傳遞的個人資料類型。您可能想要考量一些層面，例如：

- 資料如何到達您的佇列管理程式? (跨哪些通訊協定? 資料是否已加密? 資料是否已簽署?)
- 如何從佇列管理程式送出資料? (跨哪些通訊協定? 資料是否已加密? 資料是否已簽署?)
- 資料在通過佇列管理程式時如何儲存? (任何傳訊應用程式都有可能將訊息資料寫入有狀態媒體，即使訊息是非持續性。您是否瞭解傳訊特性如何可能公開透過產品傳遞之應用程式訊息資料的各個層面?)
- 如何收集並儲存 IBM MQ 存取協力廠商應用程式所需的認證?

IBM MQ 可能需要與需要鑑別的其他系統和服務 (例如 LDAP) 進行通訊。必要時，IBM MQ 會配置及儲存鑑別資料 (使用者 ID、密碼)，以用於這類通訊。盡可能避免使用個人認證進行 IBM MQ 鑑別。請考量保護用於鑑別資料的儲存體。(請參閱下面的「資料儲存體」。)

## 資料儲存

當訊息資料透過佇列管理程式傳送時，IBM MQ 會將該資料直接持續保存 (可能有多個副本) 到有狀態的媒體。IBM MQ 使用者可能想要考量在訊息資料處於靜止狀態時保護訊息資料安全。

下列項目強調顯示 IBM MQ 持續保存應用程式提供資料的區域，使用者在確保符合 GDPR 時可能想要考量這些資料。

- 應用程式訊息佇列：

IBM MQ 提供訊息佇列，以容許在應用程式之間進行非同步資料交換。儲存在佇列上的非持續性及持續訊息會寫入有狀態媒體。

- 檔案傳送代理程式佇列：

IBM MQ Managed File Transfer 會利用訊息佇列來協調檔案資料的可靠傳送，包含個人資料及傳送記錄的檔案會儲存在這些佇列上。

- 傳輸佇列：

為了在佇列管理程式之間可靠地傳送訊息，訊息會暫時儲存在傳輸佇列中。

- 無法傳送郵件的佇列:

在某些情況下，如果佇列管理程式已配置訊息，則訊息無法放入目的地佇列，且會儲存在無法傳送郵件的佇列中。

- 取消佇列:

JMS 和 XMS 傳訊介面提供一種功能，容許在發生若干取消之後，將有害訊息移至取消佇列，以容許處理其他有效訊息。

- AMS 錯誤佇列:

IBM MQ Advanced Message Security 會將不符合安全原則的訊息移至 SYSTEM.PROTECTION.ERROR.QUEUE 錯誤佇列的方式類似於無法傳送郵件的佇列作業。

- 保留的發佈資訊:

IBM MQ 提供保留的發佈特性，可讓訂閱應用程式恢復先前的發佈。

進一步閱讀:

- [記載: 確定訊息未遺失](#)
- [MFT 代理程式佇列設定](#)
- [定義傳輸佇列](#)
- [使用無法傳送郵件的佇列](#)
- [處理 IBM MQ for JMS 類別中的有害訊息](#)
- [AMS 錯誤處理](#)
- [保留的發佈](#)

下列項目強調顯示 IBM MQ 可能會間接持續保存應用程式所提供資料的區域，使用者在確保符合 GDPR 時也可能想要考量這些資料。

- 追蹤路徑傳訊:

IBM MQ 提供追蹤路徑功能，可記錄訊息在應用程式之間所採用的路徑。所產生的事件訊息可能包括技術上可識別的個人資訊，例如 IP 位址。

- 應用程式活動追蹤:

IBM MQ 提供應用程式活動追蹤，它會記錄應用程式和通道的傳訊 API 活動，應用程式活動追蹤可將應用程式提供的訊息資料內容記錄至事件訊息。

- 服務追蹤:

IBM MQ 提供服務追蹤特性，可記錄訊息資料流經的內碼路徑。作為這些特性的一部分，IBM MQ 可以記錄應用程式提供的訊息資料內容，以追蹤磁碟上儲存的檔案。

- 佇列管理程式事件:

IBM MQ 可以產生事件訊息，其中可能包含個人資料，例如權限、指令及配置事件。

進一步閱讀:

- [追蹤路徑傳訊](#)
- [使用追蹤](#)
- [事件監視](#)
- [佇列管理程式事件](#)

若要保護應用程式所提供訊息資料副本的存取權，請考量下列動作:

- 限制特許使用者存取檔案系統中的 IBM MQ 資料，例如限制 UNIX 平台上 'mqm' 群組的使用者成員資格。
- 限制應用程式透過專用佇列及存取控制來存取 IBM MQ 資料。在適當的情況下，避免不必要的資源共用 (例如應用程式之間的佇列)，並提供對佇列和主題資源的精細存取控制。
- 使用 IBM MQ Advanced Message Security 來提供訊息資料的端對端簽署及/或加密。

- 使用檔案或磁區層次加密來保護用來儲存追蹤日誌的目錄內容。
- 將服務追蹤上傳至 IBM 之後，如果您關心可能包含個人資料的內容，則可以刪除服務追蹤檔案及 FFST 資料。

進一步閱讀：

- [特許使用者](#)
- [規劃 Multiplatforms 上的檔案系統支援](#)

IBM MQ 管理者可以使用認證 (使用者名稱和密碼、API 金鑰等) 來配置佇列管理程式。適用於 3rd 服務，例如 LDAP、IBM Cloud Product Insights、Salesforce 等。此資料通常儲存在透過檔案系統許可權所保護的佇列管理程式資料目錄中。

建立 IBM MQ 佇列管理程式時，會使用群組型存取控制來設定資料目錄，以便 IBM MQ 可以讀取配置檔，並使用認證來連接至這些系統。IBM MQ 管理者被視為特許使用者，並且是此群組的成員，因此具有檔案的讀取權。部分檔案已模糊化，但未加密。基於此原因，若要完全保護對認證的存取權，您應該考量下列動作：

- 限制特許使用者對 IBM MQ 資料的存取權，例如限制 UNIX 平台上 'mqm' 群組的成員資格。
- 使用檔案或磁區層次加密來保護佇列管理程式資料目錄的內容。
- 加密正式作業配置目錄的備份，並使用適當的存取控制來儲存它們。
- 考量使用安全、指令及配置事件來提供鑑別失敗、存取控制及配置變更的審核追蹤。

進一步閱讀：

- [保護 IBM MQ](#)

## 資料存取

IBM MQ 佇列管理程式資料可以透過下列產品介面來存取，其中有些是設計為透過遠端連線來存取，而其他則是設計為透過本端連線來存取。

- IBM MQ 主控台 [僅限遠端]
- IBM MQ REST API [僅限遠端]
- MQI [本端及遠端]
- JMS [本端和遠端]
- XMS [本端和遠端]
- IBM MQ Telemetry (MQTT) [僅限遠端]
- IBM MQ Light (AMQP) [僅限遠端]
- IBM MQ IMS 橋接器 [僅限本端]
- IBM MQ CICS 橋接器 [僅限本端]
- IBM MQ for HTTP [僅限遠端]
- IBM MQ MFT 通訊協定橋接器 [僅限遠端]
- IBM MQ Connect:Direct 橋接器 [僅限遠端]
- IBM MQ Bridge to Salesforce [僅限遠端]
- IBM MQ Bridge to Blockchain [僅限遠端]
- IBM MQ MQAI [本端和遠端]
- IBM MQ PCF 指令 [本端和遠端]
- IBM MQ MQSC 指令 [本端和遠端]
- IBM MQ 探險家 [本端及遠端]

這些介面設計為容許使用者對 IBM MQ 佇列管理程式及儲存在其中的訊息進行變更。管理和傳訊作業受到保護，因此在提出要求時有三個階段；

- 鑑別
- 角色對映

- 授權

#### 鑑別:

如果從本端連線要求訊息或管理作業，則此連線的來源是相同系統上的執行中處理程序。執行處理程序的使用者必須已通過作業系統所提供的任何鑑別步驟。建立連線的處理程序擁有者的使用者名稱會主張為身分。例如，這可能是執行已從中啟動應用程式之 Shell 的使用者名稱。本端連線的可能鑑別形式如下:

1. 主張的使用者名稱 (本端 OS)
2. 選用使用者名稱及密碼 (OS、LDAP 或自訂 3rd 儲存庫)

如果從遠端連線要求管理動作，則會透過網路介面與 IBM MQ 進行通訊。可以呈現下列形式的身份，以透過網路連線進行鑑別;

1. 主張的使用者名稱 (來自遠端 OS)
2. 使用者名稱及密碼 (OS、LDAP 或自訂 3rd 儲存庫)
3. 來源網址 (例如 IP 位址)
4. X.509 數位憑證 (相互 SSL/TLS 鑑別)
5. 安全記號 (例如 LTPA2 記號)
6. 其他自訂安全 (由 3rd 結束程式提供的功能)

#### 角色對映:

在角色對映階段中，鑑別階段中提供的認證可以對映至替代使用者 ID。如果允許繼續進行對映的使用者 ID (例如，通道鑑別規則可能會封鎖管理使用者)，則在針對 IBM MQ 資源授權活動時，會將對映的使用者 ID 歸入最終階段。

#### 授權:

IBM MQ 可讓不同的使用者對不同的傳訊資源 (例如佇列、主題及其他佇列管理程式物件) 具有不同的權限。

#### 記載活動:

IBM MQ 的部分使用者可能需要建立 MQ 資源存取權的審核記錄。想要的審核日誌範例可能包括配置變更，除了要求變更的人員之外，還包含該變更的相關資訊。

下列資訊來源可用來實作此需求:

1. IBM MQ 佇列管理程式可以配置成在順利執行管理指令時產生指令事件。
2. IBM MQ 佇列管理程式可以配置成在建立、變更或刪除佇列管理程式資源時產生配置事件。
3. IBM MQ 佇列管理程式可以配置成在資源授權檢查失敗時產生權限事件。
4. 指出授權檢查失敗的錯誤訊息會寫入佇列管理程式錯誤日誌。
5. 當鑑別、授權檢查失敗或建立、啟動、停止或刪除佇列管理程式時，IBM MQ Web 主控台會將審核訊息寫入其日誌。

在考量這些類型的解決方案時，IBM MQ 使用者可能想要考量下列要點:

- 事件訊息是非持續性，因此當佇列管理程式重新啟動時，資訊會遺失。任何事件監視器都應該配置成持續使用任何可用的訊息，並將內容傳送至持續媒體。
- IBM MQ 特許使用者具有足夠專用權來停用事件、清除日誌或刪除佇列管理程式。

如需保護 IBM MQ 資料存取安全及提供審核追蹤的相關資訊，請參閱下列主題:

- [IBM MQ 安全機制](#)
- [配置事件](#)
- [指令事件](#)
- [錯誤日誌](#)

#### 資料處理

使用公開金鑰基礎架構進行加密:

您可以保護 IBM MQ 的網路連線安全，以使用 TLS，這也可以提供連線起始端的交互鑑別。

使用傳輸機制所提供的 PKI 安全機能，是使用 IBM MQ 來保護資料處理安全的第一步。不過，在不啟用進一步安全特性的情況下，消費端應用程式的行為是處理遞送給它的所有訊息，而不驗證訊息的來源或在傳輸中是否已變更。

授權使用 Advanced Message Security (AMS) 功能的 IBM MQ 使用者可以透過安全原則的定義及配置，控制應用程式處理訊息中所保留個人資料的方式。安全原則容許將數位簽章及/或加密套用至應用程式之間的訊息資料。

在使用訊息時，可以使用安全原則來要求及驗證數位簽章，以確保訊息是真實的。AMS 加密提供一種方法，透過此方法將訊息資料從可讀形式轉換為編碼版本，只有在另一個應用程式是預期的接收者或訊息且具有正確解密金鑰的存取權時，它才能被另一個應用程式解碼。

如需使用 SSL 及憑證來保護網路連線安全的相關資訊，請參閱 IBM MQ V9 產品說明文件中的下列主題：

- [配置 IBM MQ 的 TLS 安全](#)
- [AMS 概觀](#)

## 資料刪除

IBM MQ 提供指令及使用者介面動作，以刪除已提供給產品的資料。這可讓具有機能的 IBM MQ 使用者在需要時刪除與特定個體相關的資料。

- IBM MQ 行為考量符合 GDPR 用戶端資料刪除的區域
  - 透過下列方式刪除儲存在應用程式佇列上的訊息資料：
    - 使用傳訊 API 或工具或使用訊息期限來移除個別訊息。
    - 指定訊息是非持續性，保留在非持續性訊息類別正常的佇列上，並重新啟動佇列管理程式。
    - 以管理方式清除佇列。
    - 正在刪除佇列。
  - 透過下列方式刪除儲存在主題上的保留發佈資料：
    - 指定訊息是非持續性，並重新啟動佇列管理程式。
    - 將保留的資料取代為新資料或使用訊息期限。
    - 以管理方式清除主題字串。
  - 刪除整個佇列管理程式，以刪除儲存在佇列管理程式上的資料。
  - 刪除追蹤目錄中的檔案，以刪除服務追蹤指令所儲存的資料。
  - 刪除 errors 目錄中的檔案，以刪除儲存的 FFST 資料。
  - 刪除位址空間及「連結機能」傾出 (在 z/OS 上)。
  - 刪除這類資料的保存、備份或其他副本。
- IBM MQ 行為的區域，考量是否符合「GDPR 帳戶資料刪除」
  - 您可以刪除 (包括保存、備份或其他抄寫的副本)，以刪除 IBM MQ 所儲存的帳戶資料及喜好設定，以連接至佇列管理程式及 3rd 服務：
    - 儲存認證的佇列管理程式鑑別資訊物件。
    - 參照使用者 ID 的佇列管理程式權限記錄。
    - 對映或封鎖特定 IP 位址、憑證 DN 或使用使用者 ID 的佇列管理程式通道鑑別規則。
    - IBM MQ Managed File Transfer 代理程式、日誌程式及「MQ 探險家 MFT 外掛程式」所使用的認證檔，用於對佇列管理程式及檔案伺服器進行鑑別。
    - X.509 數位憑證，代表或包含金鑰儲存庫中可由 SSL/TLS 連線或 IBM MQ Advanced Message Security (AMS) 使用之個人的相關資訊。
    - IBM MQ Appliance 中的個別使用者帳戶，包括系統日誌檔中那些帳戶的參照。
    - IBM MQ 探險家工作區 meta 資料及 Eclipse 設定。

- IBM MQ 探險家密碼儲存庫，如 [密碼喜好設定](#) 中所指定。
- IBM MQ 主控台及 mqweb 伺服器配置檔。
- Salesforce 連線資料配置檔。
- 區塊鏈連線資料配置檔。
- IBM Cloud qm.ini 及 APIKeyFile 中 ReportingService 段落下的 Product Insights 連線資料。

進一步閱讀：

- [配置 IBM MQ Bridge 至 Salesforce](#)
- [配置 IBM MQ 以與區塊鏈搭配使用](#)
- [MFT 及 IBM MQ 連線鑑別](#)
- [使用 ProtocolBridgeCredentials.xml 檔案對映檔案伺服器的認證](#)
- [配置 IBM MQ 主控台使用者和角色](#)

## 資料監視

IBM MQ 提供一系列監視特性，使用者可以利用這些特性來進一步瞭解應用程式及佇列管理程式的執行方式。

IBM MQ 也提供一些特性，可協助管理佇列管理程式錯誤日誌。

進一步閱讀：

- [監視 IBM MQ 網路](#)
- [診斷訊息服務](#)
- [QMErrorLog 服務](#)

IBM MQ 提供一項特性，可讓使用者將資訊發佈至 IBM Cloud Product Insights 服務，以便 IBM MQ 使用者可以檢視佇列管理程式啟動及使用資訊。

進一步閱讀：

- [配置 IBM MQ 以與 IBM Cloud 中的 IBM Cloud Product Insights 服務搭配使用](#)

## 限制使用個人資料的功能

使用本文件中彙總的機能，IBM MQ 可讓一般使用者限制使用其個人資料。

IBM MQ 訊息佇列不應以資料庫的相同方式作為永久資料儲存庫，在處理受限於 GDPR 的應用程式資料時尤其如此。

與透過搜尋查詢找到資料的資料庫不同，除非您知道訊息的佇列、訊息及相關性 ID，否則很難找到訊息資料。

提供包含個人資料的訊息可以隨時識別及找到，可以使用標準 IBM MQ 傳訊特性來存取或修改訊息資料。

## 檔案處理

1. IBM MQ Managed File Transfer 不會對傳送的檔案執行惡意軟體掃描。檔案會依現狀傳送，並執行完整性檢查，以確保在傳送期間不會修改檔案資料。來源及目的地總和檢查會發佈為傳送狀態發佈的一部分。建議一般使用者在 MFT 傳送檔案之前，以及 MFT 將檔案遞送至遠端端點之後，針對其環境適當地實作惡意軟體掃描。
2. IBM MQ Managed File Transfer 不會根據 MIME 類型或副檔名採取動作。MFT 讀取檔案並傳送位元組，與從輸入檔讀取的位元組完全相同。

## 設計 IBM WebSphere MQ 架構

瞭解 IBM WebSphere MQ 支援點對點和發佈/訂閱傳訊樣式的不同架構。

在規劃 IBM WebSphere MQ 架構之前，請先熟悉基本 IBM WebSphere MQ 概念，請參閱 [IBM WebSphere MQ 技術概觀](#) 中的主題。

IBM WebSphere MQ 架構從使用單一佇列管理程式的簡單架構，到互連佇列管理程式的更複雜網路。使用分散式佇列技術將多個佇列管理程式連接在一起。如需規劃單一佇列管理程式及多個佇列管理程式架構的相關資訊，請參閱下列主題：

- [第 13 頁的『以單一佇列管理程式為基礎的架構』](#)
- [第 14 頁的『以多個佇列管理程式為基礎的架構』](#)
- [第 14 頁的『網路和網路規劃』](#)
- [WebSphere MQ 分散式傳訊技術](#)

如果您需要多個邏輯相關且需要共用資料及應用程式的佇列管理程式，可以在叢集中將它們分組在一起。使用叢集可以讓佇列管理程式彼此通訊，而不需要設定額外的通道定義或遠端佇列定義，以簡化其配置及管理。如需使用叢集的相關資訊，請參閱 [叢集如何運作](#)。

#### 相關概念

[第 5 頁的『規劃』](#)

規劃 IBM WebSphere MQ 環境時，您必須考量您要配置的 IBM WebSphere MQ 架構、資源需求、記載需求及備份機能。使用本主題中的鏈結，以規劃 IBM WebSphere MQ 執行所在的環境。

#### 相關工作

[配置](#)

## 以單一佇列管理程式為基礎的架構

最簡單的 IBM WebSphere MQ 架構涉及配置及使用單一佇列管理程式。

在規劃 IBM WebSphere MQ 架構之前，請先熟悉基本 IBM WebSphere MQ 概念，請參閱 [IBM WebSphere MQ 簡介](#)。

下列各節說明使用單一佇列管理程式的一些可能架構：

- [第 13 頁的『具有存取服務之本端應用程式的單一佇列管理程式』](#)
- [第 13 頁的『單一佇列管理程式，具有以用戶端形式存取服務的遠端應用程式』](#)
- [第 13 頁的『具有發佈/訂閱配置的單一佇列管理程式』](#)

## 具有存取服務之本端應用程式的單一佇列管理程式

基於單一佇列管理程式的第一個架構是存取服務的應用程式在與提供服務的應用程式相同的系統上執行。IBM WebSphere MQ 佇列管理程式在要求服務的應用程式與提供服務的應用程式之間提供非同步交互通訊。這表示即使其中一個應用程式長時間離線，也可以繼續進行應用程式之間的通訊。

## 單一佇列管理程式，具有以用戶端形式存取服務的遠端應用程式

以單一佇列管理程式為基礎的第二個架構可讓應用程式從提供服務的應用程式遠端執行。遠端應用程式在服務的不同系統上執行。應用程式會以用戶端身分連接至單一佇列管理程式。這表示可以透過單一佇列管理程式，將服務的存取權提供給多個系統。

此架構的限制是網路連線必須可供應用程式運作。應用程式與佇列管理程式之間透過網路連線的互動是同步的。

## 具有發佈/訂閱配置的單一佇列管理程式

使用單一佇列管理程式的替代架構是使用發佈/訂閱配置。在發佈/訂閱傳訊中，您可以將資訊提供者與該資訊的消費者取消連結。這不同於上述架構中的點對點傳訊樣式，其中應用程式必須知道目標應用程式的相關資訊，例如要放置訊息的佇列名稱。使用 IBM WebSphere MQ 發佈/訂閱，傳送端應用程式會根據資訊的主旨，發佈具有指定主題的訊息。IBM WebSphere MQ 會處理透過訂閱將訊息配送至已登錄關注該主題的應用程式。接收端應用程式也不需要知道訊息來源的任何資訊即可接收它們。如需發佈/訂閱傳訊的相關資訊，請參閱 [WebSphere MQ 發佈/訂閱傳訊簡介](#)。如需使用單一佇列管理程式的發佈/訂閱傳訊範例，請參閱 [單一佇列管理程式發佈/訂閱配置範例](#)。

## 相關概念

[第 12 頁的『設計 IBM WebSphere MQ 架構』](#)

瞭解 IBM WebSphere MQ 支援點對點和發佈/訂閱傳訊樣式的不同架構。

## 相關資訊

[WebSphere MQ 簡介](#)

[建立及管理佇列管理程式](#)

## 以多個佇列管理程式為基礎的架構

您可以使用分散式訊息佇列技術，來建立涉及配置及使用多個佇列管理程式的 IBM WebSphere MQ 架構。

在規劃 IBM WebSphere MQ 架構之前，請先熟悉基本 IBM WebSphere MQ 概念，請參閱 [IBM WebSphere MQ 簡介](#)。

可以透過新增其他佇列管理程式來變更 IBM WebSphere MQ 架構，而無需變更提供服務的應用程式。

應用程式可以在與佇列管理程式相同的機器上管理，然後與另一個系統上另一個佇列管理程式上管理的服務取得非同步通訊。或者，存取服務的應用程式可以作為用戶端連接至佇列管理程式，然後對另一個佇列管理程式上的服務提供非同步存取。

使用分散式佇列技術來定義連接不同佇列管理程式及其佇列的路徑。架構內的佇列管理程式會使用通道來連接。視佇列管理程式的配置而定，通道可用來將訊息從一個佇列管理程式自動移至另一個佇列管理程式。

如需規劃 IBM WebSphere MQ 網路的高階概觀，請參閱 [第 14 頁的『網路和網路規劃』](#)。

如需如何為 IBM WebSphere MQ 架構規劃通道的相關資訊，請參閱 [WebSphere MQ 分散式傳訊技術](#)。

分散式佇列管理可讓您建立及監視佇列管理程式之間的通訊。如需分散式佇列管理的相關資訊，請參閱 [分散式佇列管理簡介](#)。

## 相關概念

[WebSphere MQ 簡介](#)

[第 12 頁的『設計 IBM WebSphere MQ 架構』](#)

瞭解 IBM WebSphere MQ 支援點對點和發佈/訂閱傳訊樣式的不同架構。

## 相關工作

[建立及管理佇列管理程式](#)

## 網路和網路規劃

WebSphere MQ 會利用「佇列管理程式」和「通道」，在應用程式之間，以及透過網路來傳送和接收資料。網路規劃包括定義需求，以建立架構來透過網路連接這些系統。

可以在系統與您需要進行通訊的任何其他系統之間建立通道。可以建立多躍點通道，以連接至您沒有直接連線的系統。實務範例中說明的訊息通道連線在 [第 15 頁的圖 1](#) 中顯示為網路圖。

## 通道及傳輸佇列名稱

傳輸佇列可以指定任何名稱。但為了避免混淆，您可以視情況提供與目的地佇列管理程式名稱或佇列管理程式別名相同的名稱。這會將傳輸佇列與它們使用的路徑相關聯，並提供透過中間 (多中繼站) 佇列管理程式所建立之平行路徑的明確概觀。

頻道名稱並不明確。例如，對於送入及送出通道，[第 15 頁的圖 1 for QM2](#) 中的通道名稱必須不同。所有通道名稱仍然可以包含其傳輸佇列名稱，但它們必須是完整的，才能使它們成為唯一的。

例如，在 QM2 上，有一個來自 QM1 的 QM3 通道，以及一個前往 QM3 的 QM3 通道。若要使名稱成為唯一，第一個名稱可能命名為 'QM3\_from\_QM1'，第二個名稱可能命名為 'QM3\_from\_QM2'。如此一來，通道名稱會在名稱的第一部分顯示傳輸佇列名稱。名稱的第二部分會顯示方向及相鄰佇列管理程式名稱。

[第 15 頁的表 1](#) 中提供了 [第 15 頁的圖 1](#) 的建議通道名稱表格。

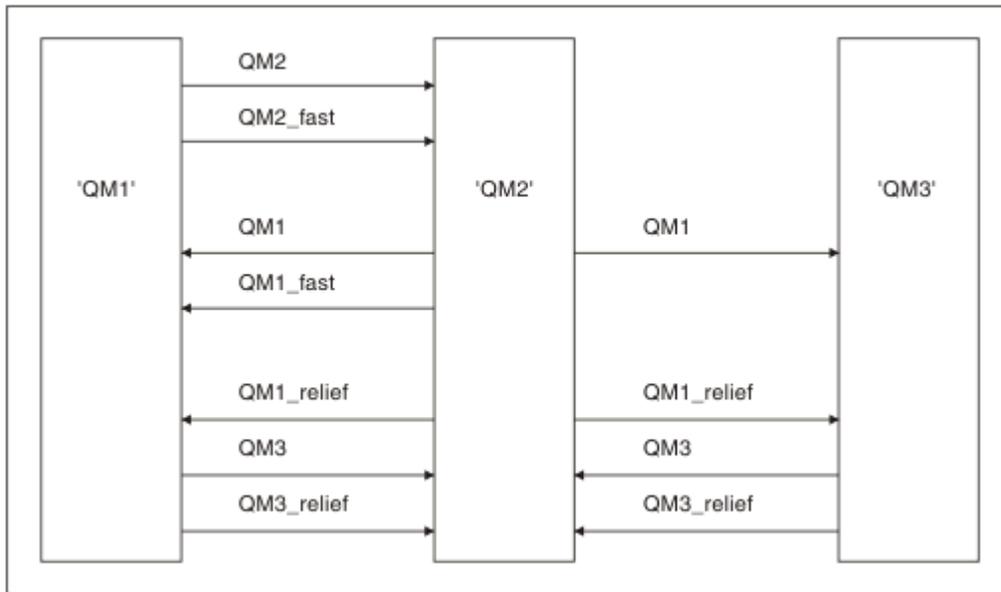


圖 1: 顯示所有通道的網路圖

路徑名稱	管理通道的佇列管理程式	傳輸佇列名稱	建議的通道名稱
QM1	QM1 & QM2	QM1 (在 QM2)	QM1.from.QM2
QM1	QM2 & QM3	QM1 (於 QM3)	QM1.from.QM3
QM1_fast	QM1 & QM2	QM1_fast (位於 QM2)	QM1_fast.from.QM2
QM1_relief	QM1 & QM2	QM1_relief (於 QM2)	QM1_relief.from.QM2
QM1_relief	QM2 & QM3	QM1_relief (於 QM3)	QM1_relief.from.QM3
QM2	QM1 & QM2	QM2 (位於 QM1)	QM2.from.QM1
QM2_fast	QM1 & QM2	QM2_fast (位於 QM1)	QM2_fast.from.QM1
QM3	QM1 & QM2	QM3 (於 QM1)	QM3.from.QM1
QM3	QM2 & QM3	QM3 (於 QM2)	QM3.from.QM2
QM3_relief	QM1 & QM2	QM3_relief (於 QM1)	QM3_relief.from.QM1
QM3_relief	QM2 & QM3	QM3_relief (於 QM2)	QM3_relief.from.QM2

註:

1. 在 WebSphere MQ for z/OS 上，佇列管理程式名稱限制為四個字元。
2. 唯一地命名網路中的所有通道。如 [第 15 頁的表 1](#) 所示，在通道名稱中包括來源和目標佇列管理程式名稱是很好的作法。

## 網路規劃程式

建立網路會假設有另一個更高層次的 網路規劃程式 功能，其計劃由團隊的其他成員實作。

對於廣泛使用的應用程式，使用本端存取網站之間的寬頻鏈結，以本端存取網站來考慮訊息資料流量的集中，會比較經濟，如 [第 16 頁的圖 2](#) 所示。

在此範例中，有兩個主要系統及數個衛星系統。實際配置取決於商業考量。有兩個集中器佇列管理程式位於方便的中心。每一個 QM-concentrator 都有本端佇列管理程式的訊息通道：

- QM-concentrator 1 具有三個本端佇列管理程式 (QM1、QM2 及 QM3) 的訊息通道。使用這些佇列管理程式的應用程式可以透過 QM-集中器彼此通訊。
- QM-concentrator 2 具有三個本端佇列管理程式 (QM4、QM5 及 QM6) 的訊息通道。使用這些佇列管理程式的應用程式可以透過 QM-集中器彼此通訊。
- QM 集中器之間有訊息通道，因此容許佇列管理程式中的任何應用程式與另一個佇列管理程式中的任何其他應用程式交換訊息。

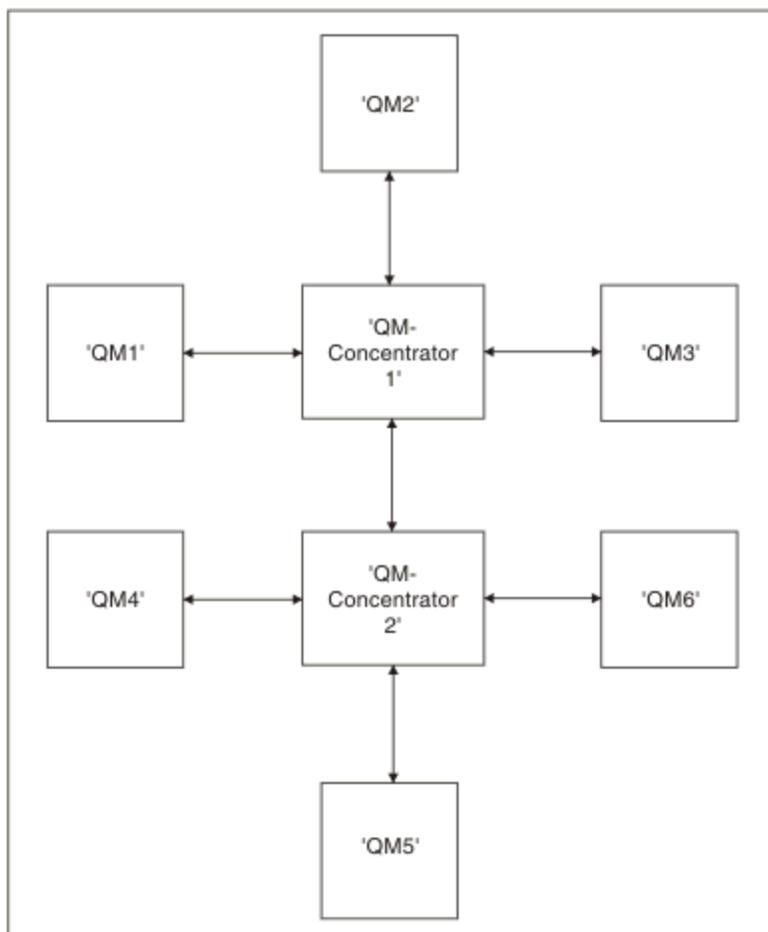


圖 2: 顯示 QM 集中器的網路圖

## 叢集

本主題提供規劃及管理 IBM WebSphere MQ 叢集的指引。此資訊是根據客戶的測試和意見而提供的指引。

下列資訊假設使用者對 IBM WebSphere MQ 叢集有基本瞭解。此資訊並非預期為 "單一大小適合所有" 解決方案，而是嘗試共用一般問題的一般方法。

叢集提供一種機制來交互連接佇列管理程式，以簡化設定系統所需的起始配置，以及所需的持續管理。配置越大，好處越大。

規劃叢集系統時需要小心，以確保叢集系統正確運作，並確保系統所需的可用性和回應性層次，特別是對於更大或更複雜的叢集系統。

成功的叢集設定取決於良好的規劃，以及對 IBM WebSphere MQ 基本概念的透徹瞭解，例如良好的應用程式管理及網路設計。確保您熟悉 [交互通訊的概念](#) 及 [叢集運作方式](#) 中的資訊。

## 何謂叢集以及使用它們的原因?

叢集作業提供兩個主要好處:

- 叢集可簡化 IBM WebSphere MQ 網路的管理，這些網路通常需要配置通道、傳輸佇列及遠端佇列的許多物件定義。在許多佇列管理程式需要交互連接的大型、潛在變更網路中，尤其如此。此架構特別難以配置及主動維護。
- 叢集可用來在叢集中的佇列及佇列管理程式之間配送訊息資料流量的工作量。這類配送可讓單一佇列的訊息工作量分散在位於多個佇列管理程式上該佇列的對等實例之間。此工作量分佈可用來提高系統故障的復原力，並增進系統中特別作用中訊息流程的調整效能。在這種環境中，分散式佇列的每一個實例都有消費端應用程式處理訊息。

### 相關資訊

[叢集作業: 最佳作法](#)

## 點對點傳訊

IBM WebSphere MQ 中最簡單的傳訊形式是點對點傳訊。

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

您可以與 IBM WebSphere MQ 搭配使用的替代傳訊樣式是發佈/訂閱傳訊。發佈/訂閱傳訊可讓您取消資訊提供者與該資訊消費者之間的連結。傳送應用程式及接收應用程式不必對彼此進行任何瞭解，即可傳送及接收資訊。如需發佈/訂閱傳訊的相關資訊，請參閱 [WebSphere MQ 發佈/訂閱傳訊簡介](#)。

### 相關資訊

[開發應用程式](#)

[WebSphere MQ 訊息](#)

## IBM WebSphere MQ 發佈/訂閱傳訊簡介

發佈/訂閱傳訊可讓您取消資訊提供者與該資訊消費者之間的連結。傳送應用程式及接收應用程式不必對彼此進行任何瞭解，即可傳送及接收資訊。

在點對點 IBM WebSphere MQ 應用程式可以將訊息傳送至另一個應用程式之前，它需要瞭解該應用程式的相關資訊。例如，它需要知道要將資訊傳送至其中的佇列名稱，也可能指定佇列管理程式名稱。

IBM WebSphere MQ 發佈/訂閱可讓您的應用程式不需要知道目標應用程式的任何相關資訊。所有傳送端應用程式都必須執行下列動作: 放置一則 IBM WebSphere MQ 訊息，其中包含它想要的資訊，並指派一個主題給它，以表示資訊的主旨，並讓 IBM WebSphere MQ 處理該資訊的配送。同樣地，目標應用程式不需要知道它所接收資訊的來源。

第 17 頁的圖 3 顯示最簡單的發佈/訂閱系統。有一個發佈者、一個佇列管理程式及一個訂閱者。訂閱會從訂閱者傳送至佇列管理程式，發佈會從發佈者傳送至佇列管理程式，然後由佇列管理程式將發佈轉遞至訂閱者。

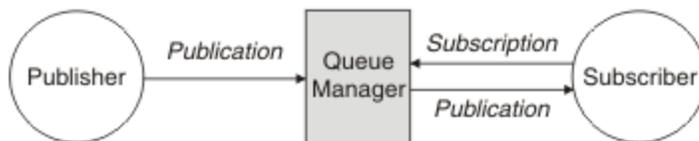


圖 3: 簡式發佈/訂閱配置

一般發佈/訂閱系統有多個發佈者和多個訂閱者，且通常有多個佇列管理程式。應用程式可以同時是發佈者和訂閱者。

## 發佈/訂閱元件概觀

發佈/訂閱是訂閱者可以從發佈者接收資訊 (以訊息形式) 的機制。發佈者與訂閱者之間的互動由佇列管理程式使用標準 WebSphere MQ 機能來控制。

一般發佈/訂閱系統有多個發佈者和多個訂閱者，且通常有多個佇列管理程式。應用程式可以同時是發佈者和訂閱者。

資訊的提供者稱為發佈者。發佈者提供主旨的相關資訊，而不需要知道對該資訊感興趣的應用程式的任何相關資訊。發佈者以訊息形式產生此資訊，稱為發佈，他們想要發佈並定義這些訊息的主題。

資訊的取用者稱為訂閱者。訂閱者建立訂閱，以說明訂閱者感興趣的主題。因此，訂閱會決定將哪些發佈轉遞給訂閱者。訂閱者可以執行多個訂閱，並且可以從許多不同的發佈者接收資訊。

已發佈資訊會在 WebSphere MQ 訊息中傳送，且資訊的主旨由其主題識別。發佈者會指定發佈資訊時的主題，而訂閱者會指定它要接收其發佈資訊的主題。訂閱者只會收到其訂閱之主題的相關資訊。

主題的存在可讓資訊提供者和消費者在發佈/訂閱傳訊中取消連結，因為不需要在每一個訊息中包含點對點傳訊所需要的特定目的地。

發佈者與訂閱者之間的互動全都由佇列管理程式控制。佇列管理程式會從發佈者接收訊息，並從訂閱者接收訂閱(主題範圍)。佇列管理程式的工作是將已發佈的訊息遞送至已登錄感興趣訊息主題的訂閱者。

標準 WebSphere MQ 機能用來配送訊息，因此您的應用程式可以使用現有 WebSphere MQ 應用程式可用的所有特性。這表示您可以使用持續訊息來取得僅一次的保證遞送，且您的訊息可以是交易式工作單元的一部分，以確保只有在發佈者確定訊息時才將訊息遞送至訂閱者。

## 單一佇列管理程式發佈/訂閱配置的範例

第 18 頁的圖 4 說明基本單一佇列管理程式發佈/訂閱配置。此範例顯示新聞服務的配置，其中發佈者提供數個主題的相關資訊：

- 發佈者 1 使用「體育」主題來發佈體育結果的相關資訊
- 發佈者 2 使用「股票」主題來發佈股價的相關資訊
- 發佈者 3 使用影片主題來發佈電影評論的相關資訊，以及使用電視主題來發佈電視清單的相關資訊

三個訂閱者已登錄對不同主題感興趣，因此佇列管理程式會將他們感興趣的資訊傳送給他們：

- 用戶 1 收到運動成績和股票價格
- 訂閱者 2 收到影片評論
- 訂閱者 3 會接收運動結果

沒有任何使用者對電視清單有興趣，因此這些都沒有被分發。

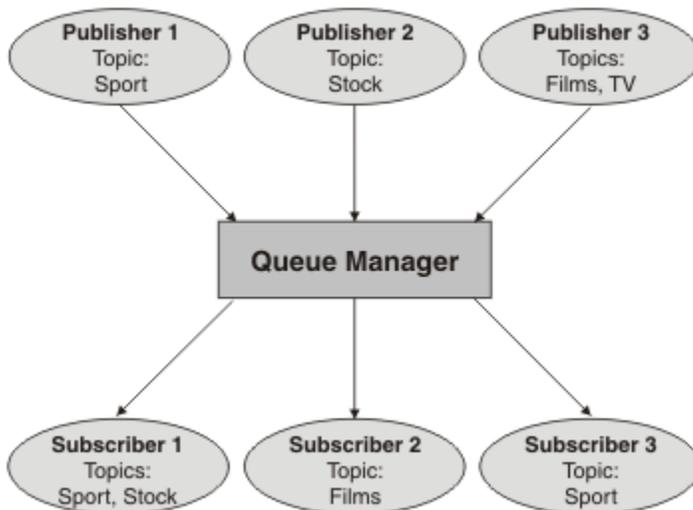


圖 4: 單一佇列管理程式發佈/訂閱範例

## 發佈者和發佈

在 WebSphere MQ 中，發佈/訂閱發佈者是一種應用程式，以標準 WebSphere MQ 訊息 (稱為發佈) 形式提供指定主題的相關資訊給佇列管理程式。發佈者可以發佈多個主題的相關資訊。

發佈者使用 MQPUT 動詞將訊息放置到先前開啟的主題，此訊息是發佈。然後，本端佇列管理程式會將發佈遞送給具有訂閱發佈主題的任何訂閱者。已發佈的訊息可由多個訂閱者耗用。

除了將發佈配送至具有適當訂閱的所有本端訂閱者之外，佇列管理程式也可以直接或透過具有主題訂閱者的佇列管理程式網路，將發佈配送至與其連接的任何其他佇列管理程式。

在 WebSphere MQ 發佈/訂閱網路中，發佈應用程式也可以是訂閱者。

### 同步點下的出版品

發佈者可以在同步點中發出 MQPUT 或 MQPUT1 呼叫，以將遞送給訂閱者的所有訊息併入工作單元中。如果指定 MQPMO\_RETAIN 選項或主題遞送選項 NPMGDLV 或 PMSGDLV，且值為 ALL 或 ALLDURR，則佇列管理程式會在發佈者 MQPUT 或 MQPUT1 呼叫的範圍內，使用同步點中的內部 MQPUT 或 MQPUT1 呼叫。

## 狀態和事件資訊

發佈可以分類為狀態發佈 (例如股票的現行價格) 或事件發佈 (例如該股票的交易)。

### 狀態發佈

狀態出版品 包含某項目現行狀態的相關資訊，例如股票價格或足球比賽中的現行評分。當發生某些情況 (比方說，股價變動或足球分數改變) 時，之前的狀態資訊就不再需要，因為它會被新的資訊所取代。

訂閱者會想要在啟動時接收狀態資訊的現行版本，並在狀態變更時傳送新資訊。

如果發佈包含狀態資訊，則通常會將它發佈為保留的發佈。新訂閱者通常會立即想要現行狀態資訊; 訂閱者不想要等待導致重新發佈資訊的事件。除非訂閱者使用 MQSO\_PUBLICATIONS\_ON\_REQUEST 或 MQSO\_NEW\_PUBLICATIONS\_ONLY 選項，否則訂閱者會在訂閱時自動接收主題的保留發佈。

### 事件發佈

事件出版品 包含所發生個別事件的相關資訊，例如部分股票的交易或特定目標的評分。每一個事件與其他事件都是互相獨立的。

訂閱者將想要在事件發生時接收事件的相關資訊。

### 保留的發佈

依預設，在將發佈傳送至所有感興趣的訂閱者之後，會捨棄它。不過，發佈者可以指定保留發佈的副本，以便將它傳送給未來對主題有興趣的訂閱者。

將發佈傳送至所有感興趣的訂閱者之後刪除發佈，適用於事件資訊，但並非一律適用於狀態資訊。透過保留訊息，新訂閱者不必在接收起始狀態資訊之前等待重新發佈資訊。例如，訂閱股票價格的訂閱者會立即收到現行價格，而不會等待股價變更 (因此重新發佈)。

佇列管理程式只能為每個主題保留一個發佈資訊，因此當新的保留發佈資訊到達佇列管理程式時，會刪除主題的現有保留發佈資訊。不過，刪除現有發佈資訊可能不會與新保留發佈資訊的到達同步發生。因此，在可能的情況下，最多只有一個發佈者傳送任何主題的保留發佈資訊。

訂閱者可以使用 MQSO\_NEW\_PUBLICATIONS\_ONLY 訂閱選項，指定他們不想要接收保留的發佈。現有訂閱者可以要求將保留發佈的副本傳送給他們。

有時您可能不想要保留發佈，即使是為了取得狀態資訊：

- 如果某個主題的所有訂閱都在對該主題進行任何發佈之前進行，且您不預期或不容許新的訂閱，則不需要保留發佈，因為它們在第一次發佈時遞送給完整的訂閱者集。
- 如果經常發生發佈，例如每秒，則新訂閱者 (或從失敗回復的訂閱者) 會在其起始訂閱之後幾乎立即收到現行狀態，因此不需要保留這些發佈。

- 如果發佈資訊很大，您可能最終需要大量儲存體空間來儲存每一個主題的保留發佈資訊。在多個佇列管理程式環境中，網路中具有相符訂閱的所有佇列管理程式都會儲存保留的發佈。

在決定是否使用保留的發佈資訊時，請考量訂閱應用程式如何從失敗中回復。如果發佈者未使用保留的發佈，則訂閱者應用程式可能需要在本地儲存其現行狀態。

若要確保保留發佈資訊，請使用 MQPMO\_RETAIN put-message 選項。如果使用此選項，且無法保留發佈，則不會發佈訊息，且呼叫會失敗並產生 MQRC\_PUT\_NOT\_RETAINED。

如果訊息是保留的發佈資訊，則由 MQIsRetained 訊息內容指出。訊息的持續性與最初發佈訊息時一樣。

## 同步點下的發佈

在 IBM WebSphere MQ 發佈/訂閱中，同步點可由發佈者使用，或由佇列管理程式在內部使用。

當發佈者使用 MQPMO\_SYNCPOINT 選項發出 MQPUT/MQPUT1 呼叫時，會使用同步點。遞送給訂閱者的所有訊息都會計入 work.The MAXUMSGS 佇列管理程式屬性會指定此限制。如果達到限制，發佈者會收到 2024 (07E8) (RC2024): MQRC\_SYNCPOINT\_LIMIT\_REACHED 原因碼。

當發佈者使用 MQPMO\_NO\_SYNCPOINT 搭配 MQPMO\_RETAIN 選項或主題遞送選項 NPMSGDLV/PMSGDLV (值為 ALL 或 ALLDURR) 發出 MQPUT/MQPUT1 呼叫時，佇列管理程式會使用內部同步點來保證訊息會依要求遞送。如果在發佈者 MQPUT/MQPUT1 呼叫的範圍內達到限制，發佈者可以接收 2024 (07E8) (RC2024) :XX\_ENCODE\_CASE\_ONE mqrc\_syncpoint\_limit\_reached 原因碼。

## 訂閱者和訂閱

在 WebSphere MQ 發佈/訂閱中，訂閱者是從發佈/訂閱網路中的佇列管理程式要求特定主題相關資訊的應用程式。訂閱者可以從多個發佈者接收關於相同或不同主題的訊息。

訂閱可以使用 MQSC 指令或由應用程式手動建立。這些訂閱會發出至本地佇列管理程式，並包含訂閱者想要接收之發佈的相關資訊：

- 訂閱者感興趣的主題；如果使用萬用字元，則這可以解析為多個主題。
- 要套用至已發佈訊息的選用選擇字串。
- 應該放置所選發佈的佇列控點（稱為 訂閱者佇列），以及選用的 CorrelId。

本地佇列管理程式會儲存訂閱資訊，當它收到發佈資訊時，會掃描資訊以判斷是否有訂閱符合發佈資訊的主題和選擇字串。對於每一個相符的訂閱，佇列管理程式會將發佈導向訂閱者的訂閱者佇列。可以使用 DIS SUB 及 DIS SBSTATUS 指令來檢視佇列管理程式儲存的訂閱相關資訊。

只有在發生下列其中一個事件時，才會刪除訂閱：

- 訂閱者使用 MQCLOSE 呼叫取消訂閱 (如果訂閱已非可延續的話)。
- 訂閱到期。
- 系統管理者會使用 DELETE SUB 指令來刪除訂閱。
- 訂閱者應用程式結束 (如果使訂閱無法持久)。
- 佇列管理程式會停止或重新啟動 (如果訂閱已非可延續的話)。

取得訊息時，請在 MQGET 呼叫上使用適當的選項。如果您的應用程式只處理一個訂閱的訊息，則您至少應該使用 get-by-correlid，如 C 範例程式 amqssbxa.c 及 未受管理 MQ 訂閱者中所示範。要使用的 **CorrelId** 會從 MQSD 中的 MQSUB 傳回。**SubCorrelId** 欄位。

## 受管理佇列及發佈/訂閱

當您建立訂閱時，可以選擇使用受管理佇列作業。如果您使用受管理佇列，則在建立訂閱時，會自動建立訂閱佇列。受管理佇列會根據訂閱的持續性自動進行整理。使用受管理佇列表示您不必擔心建立佇列來接收發佈，如果關閉不可延續訂閱連線，則會自動從訂閱者佇列中移除任何未耗用的發佈。

如果應用程式不需要使用特定佇列作為其訂閱者佇列 (它所接收發佈的目的地)，則可以使用 MQSQ\_MANAGED 訂閱選項來使用 受管理訂閱。如果您建立受管理訂閱，佇列管理程式會針對佇列管理程式建立的訂閱者佇列，將物件控點傳回給訂閱者，其中將會接收發佈。將會傳回佇列的物件控點，可讓您瀏覽、取得或查詢佇列 (除非明確授與您暫時動態佇列的存取權，否則無法放置或設定受管理佇列的屬性)。

訂閱的持續性會決定當訂閱應用程式與佇列管理程式的連線中斷時，受管理佇列是否仍然存在。

當與不可延續訂閱一起使用時，受管理訂閱特別有用，因為當應用程式連線結束時，未耗用的訊息會無限期保留在取用者佇列中，佔用佇列管理程式中的空間。如果您是使用受管理訂閱，則受管理佇列會是暫時動態佇列，因此當連線因下列任何原因而中斷時，會連同任何未耗用的訊息一併刪除：

- 使用具有 MQCO\_REMOVE\_SUB 的 MQCLOSE，並關閉受管理 Hobj。
- 使用不可延續訂閱 (MQSO\_NON\_DURABLE) 來失去與應用程式的連線。
- 已移除訂閱，因為它已過期且受管理 Hobj 已關閉。

受管理訂閱也可以與可延續訂閱搭配使用，但您可能想要將未耗用的訊息保留在訂閱者佇列上，以便在重新開啟連線時可以擷取它們。因此，可延續訂閱的受管理佇列會採用永久動態佇列的形式，當訂閱應用程式與佇列管理程式的連線中斷時仍會保留。

如果您要使用永久動態受管理佇列，則可以在訂閱上設定期限，這樣雖然在連線中斷之後佇列仍會存在，但不會無限期地繼續存在。

如果您刪除受管理佇列，則會收到錯誤訊息。

所建立的受管理佇列會以結束時的數字 (時間戳記) 來命名，因此每一個佇列都是唯一的。

## 訂閱延續性

訂閱可以配置成可延續或不可延續。訂閱延續性會決定當訂閱應用程式與佇列管理程式中斷連線時，訂閱會發生什麼情況。

## 可延續訂閱

關閉訂閱應用程式與佇列管理程式的連線時，可延續訂閱會繼續存在。如果訂閱是可延續的，當訂閱應用程式中斷連線時，訂閱會保留在原處，當訂閱應用程式使用建立訂閱時所傳回的 SubName 重新重新連接要求訂閱時，它可以使用該訂閱。

當持續訂閱時，需要訂閱名稱 (SubName)。在佇列管理程式內，訂閱名稱必須是唯一的，才能用來識別訂閱。如果您故意關閉訂閱的控點 (使用 MQCO\_KEEP\_SUB 選項) 或已從佇列管理程式中斷連線，則在指定您要回復的訂閱時，這是必要的識別方法。您可以使用 MQSUB 呼叫搭配 MQSO\_RESUME 選項來回復現有的訂閱。如果您搭配使用 DISPLAY SBSTATUS 指令與 SUBTYPE ALL 或 ADMIN，也會顯示訂閱名稱。

當應用程式不再需要可延續訂閱時，可以搭配使用 MQCLOSE 函數呼叫與 MQCO\_REMOVE\_SUB 選項來移除它，也可以使用 MQSC 指令 DELETE SUB 來手動刪除它。

是否可以使用 DURSUB 主題屬性來控制主題的可延續訂閱。

使用 MQSO\_RESUME 選項從 MQSUB 呼叫傳回時，訂閱期限會設為訂閱的原始期限，而不是剩餘到期時間。

即使該訂閱者應用程式未連接，佇列管理程式仍會繼續傳送發佈以滿足可延續訂閱。這會導致在訂閱者佇列上建立訊息。避免此問題的最簡單方法是在適當的情況下使用不可延續訂閱。不過，當需要使用可延續訂閱時，如果訂閱者使用保留的發佈選項來訂閱，則可以避免建立訊息。然後，訂閱者可以使用 MQSUBRQ 呼叫來控制何時接收發佈。

## 不可延續訂閱

只有在訂閱應用程式與佇列管理程式的連線維持開啟時，不可延續訂閱才會存在。當訂閱應用程式有意地或由於遺失連線，而中斷與佇列管理程式的連線時，會移除訂閱。當連線關閉時，會從佇列管理程式移除訂閱的相關資訊，如果您使用 DISPLAY SBSTATUS 指令來顯示訂閱，則不再顯示訂閱的相關資訊。不再將任何訊息放置到訂閱者佇列。

對於不可延續訂閱，訂閱者佇列上任何未耗用的發佈會發生什麼情況，如下所示。

- 如果訂閱應用程式使用受管理目的地，則會自動移除任何尚未使用的發佈。
- 如果訂閱應用程式在訂閱時提供其專屬訂閱者佇列的控點，則不會自動移除未耗用的訊息。如果適當的話，應用程式有責任清除佇列。如果佇列由多個訂閱者或其他點對點應用程式共用，則可能不適合完全清除佇列。

雖然不可延續訂閱不需要，但佇列管理程式會使用訂閱名稱 (如果有提供的話)。在佇列管理程式內，訂閱名稱必須是唯一的，才能用來識別訂閱。

## 選取字串

選取字串是套用至發佈的表示式，用來判斷它是否符合訂閱。選取字串可以包含萬用字元。

當您訂閱時，除了指定主題之外，您還可以指定選取字串，以根據發佈的訊息內容來選取發佈。

## 主題

主題是發佈/訂閱訊息中所發佈資訊的主旨。

點對點系統中的訊息會傳送至特定的目的地地址。根據說明訊息內容的主旨，將主旨型發佈/訂閱系統中的訊息傳送給訂閱者。在內容型系統中，訊息會根據訊息本身的內容來傳送給訂閱者。

IBM WebSphere MQ 發佈/訂閱系統是主體型發佈/訂閱系統。發佈者會建立訊息，並使用最適合發佈主題的主題字串來發佈訊息。為了接收發佈資訊，訂閱者會建立訂閱，此訂閱具有用來選取發佈主題的型樣相符主題字串。佇列管理程式會將發佈資訊，遞送至其訂閱與發佈主題相符且獲授權來接收發佈資訊的訂閱者。第 22 頁的『主題字串』一文說明識別發佈主題之主題字串的語法。訂閱者也會建立主題字串，以選取要接收的主題。訂閱者建立的主題字串可以包含兩種替代萬用字元架構中的任一種，以根據發佈中的主題字串進行型樣比對。型樣相符在第 23 頁的『萬用架構』中說明。

在主題型發佈/訂閱中，發佈者或管理者負責將主題分類成主題。通常主體會以階層方式組織成主題樹狀結構，並使用 '/' 字元在主題字串中建立子主題。如需主題樹狀結構的範例，請參閱第 28 頁的『主題樹狀結構』。主題是主題樹狀結構中的節點。主題可以是沒有任何子主題的葉節點，或者是具有子主題的中間節點。

與將主體組織成階層式主題樹狀結構一樣，您可以將主題與管理主題物件相關聯。您可以將屬性指派給主題，例如主題是否配送在叢集中，方法是將它與管理主題物件相關聯。建立關聯的方式是使用管理主題物件的 TOPICSTR 屬性來命名主題。如果您未明確關聯管理主題物件與主題，該主題會繼承與管理主題物件相關聯之主題樹狀結構中最接近上代的屬性。如果您根本未定義任何母項主題，它會繼承自 SYSTEM.BASE.TOPIC。管理主題物件在第 30 頁的『管理主題物件』中說明。

註：即使您從 SYSTEM.BASE.TOPIC，為直接繼承自 SYSTEM.BASE.TOPIC。例如，在美國各州的主題空間中，USA/Alabama USA/Alaska 等，USA 是根主題。根主題的主要目的是建立離散、非重疊的主題空間，以避免發佈與錯誤訂閱相符。這也表示您可以變更根主題的屬性，以影響整個主題空間。例如，您可以設定 CLUSTER 屬性的名稱。

當您將主題稱為發佈者或訂閱者時，您可以選擇提供主題字串、參照主題物件或兩者，在此情況下，您提供的主題字串會定義主題物件的子主題。佇列管理程式會識別主題，方法是將主題字串附加到主題物件中指定的主題字串字首，並在兩個主題字串之間插入其他 '/'，例如 *topic string/object string*。第 26 頁的『結合主題字串』進一步說明此情況。產生的主題字串用來識別主題，並將它與管理主題物件相關聯。管理主題物件不一定與對應於主要主題的主題物件相同。

在內容型發佈/訂閱中，您可以提供選項字串來搜尋每一則訊息的內容，以定義您要接收的訊息。WebSphere MQ 提供中間形式的內容型發佈/訂閱（使用訊息選取器來掃描訊息內容，而非訊息完整內容），請參閱選取器。訊息選取元的原型用法是訂閱主題，然後以數值內容來限定選項。選取元可讓您指定您只對特定範圍內的值感興趣；您無法使用字元或主題型萬用字元來執行此動作。如果您確實需要根據訊息的完整內容進行過濾，則需要使用 WebSphere Message Broker。

## 主題字串

使用主題字串來標示您發佈為主題的資訊。使用以字元為主或主題型萬用字元主題字串，來訂閱主題群組。

## 主題

主題字串是識別發佈/訂閱訊息主題的字串。當您建構主題字串時，可以使用任何您喜歡的字元。



在第 7 版發佈/訂閱中，有三個字元具有特殊意義。它們可以在主題字串中的任何位置使用，但請小心使用。特殊字元的用法在第 23 頁的『主題型萬用架構』中說明。

## 正斜線 (/)

主題層次分隔字元。使用 '/' 字元，將主題建構成主題樹狀結構。

如果可以的話，請避免空的主題層次 '/'/'。這些對應於主題階層中沒有主題字串的節點。主題字串中的前導或尾端 '/' 對應於前導或尾端空白節點，也應該避免。

## 雜湊符號 (#)

與 '/' 組合使用，以在訂閱中建構多層次萬用字元。在用來命名已發佈主題的主題字串中，請小心使用 '/' 旁的 '#'。第 23 頁的『主題字串範例』顯示 '#' 的合理用法。

字串 '.../#/...'、'#/...' 和 '.../#' 在訂閱主題字串中具有特殊意義。字串符合主題階層中一個以上層次的所有主題。因此，如果您建立具有其中一個序列的主題，則無法訂閱它，除非同時訂閱主題階層中多個層次的所有主題。

## 加號 (+)

與 '/' 組合使用，以在訂閱中建構單一層次萬用字元。在用來命名已發佈主題的主題字串中，請小心使用 '/' 旁的 '+'。

字串 '.../+/...'、'+/...' 和 '.../+' 在訂閱主題字串中具有特殊意義。字串符合主題階層中一個層次的所有主題。因此，如果您建立具有其中一個序列的主題，則無法訂閱它，除非同時訂閱主題階層中一個層次的所有主題。

## 主題字串範例

```
IBM/Business Area#/Results
IBM/Diversity/%African American
```

### 萬用架構

有兩種用於訂閱多個主題的萬用架構。架構選項是一個訂閱選項。

### MQSO\_WILDCARD\_TOPIC

使用主題型萬用字元架構來選取要訂閱的主題。

如果未明確選取萬用字元綱目，則這是預設值。

### MQSO\_WILDCARD\_CHAR

選取要使用文字萬用字元架構訂閱的主題。

透過在 DEFINE SUB 指令上指定 **wschema** 參數來設定任一架構。如需相關資訊，請參閱 [DEFINE SUB](#)。

註：在 WebSphere MQ 7.0 版之前建立的訂閱一律使用文字萬用字元架構。

## 範例

```
IBM/+/Results
#/Results
IBM/Software/Results
IBM/*ware/Results
```

### 主題型萬用架構

主題型萬用字元可讓訂閱者一次訂閱多個主題。

在 WebSphere MQ 發佈/訂閱中，主題型萬用字元是主題系統的一項強大特性。多層次萬用字元及單層次萬用字元可用於訂閱，但訊息發佈者不可在主題中使用這些萬用字元。

主題型萬用字元架構可讓您選取依主題層次分組的發佈。您可以針對主題階層中的每一個層次，選擇訂閱中該主題層次的字串是否必須完全符合發佈中的字串。例如，訂閱 `IBM/+/Results` 會選取所有主題，

```
IBM/Software/Results
IBM/Services/Results
IBM/Hardware/Results
```

萬用字元有兩種類型。

## 多層次萬用字元

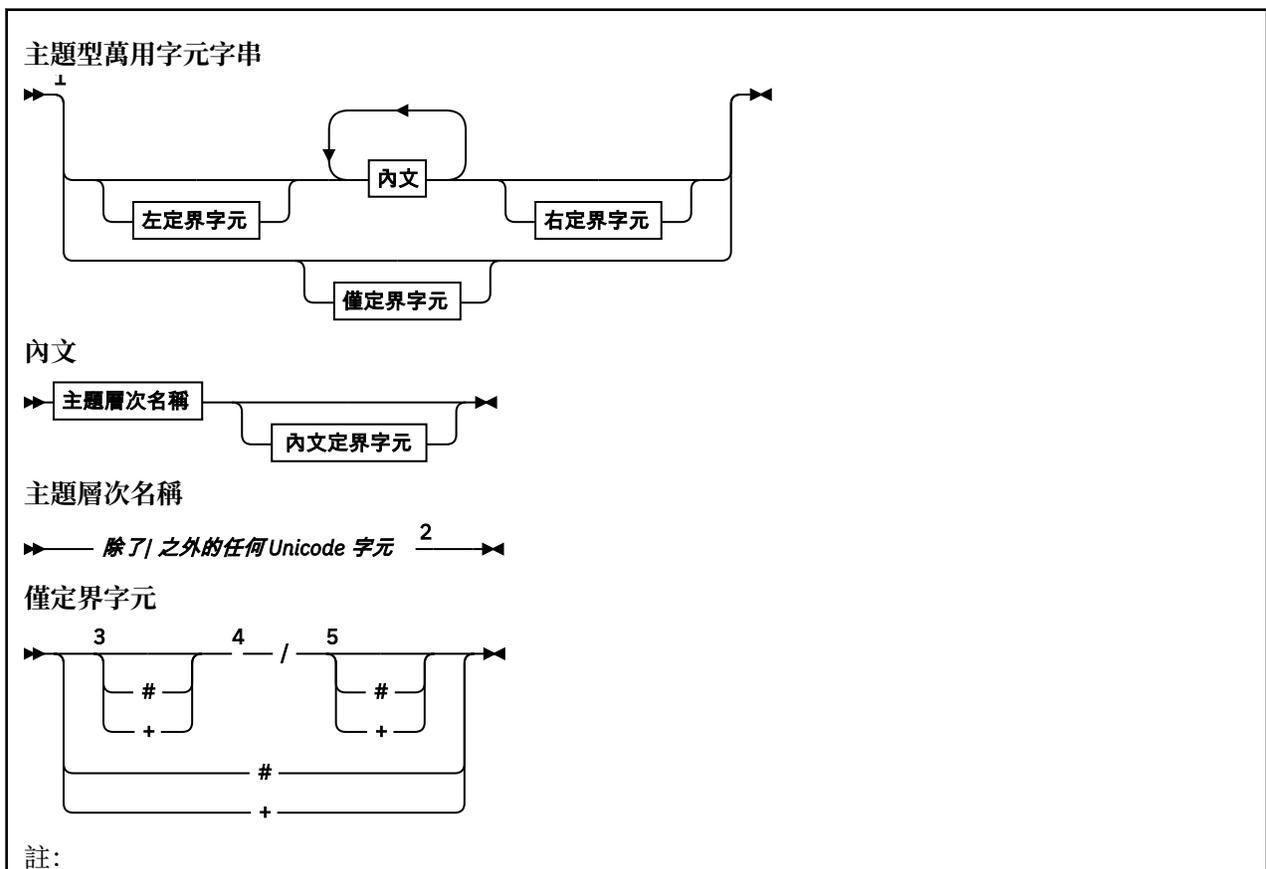
- 訂閱中使用多層次萬用字元。在發佈中使用時，會將它視為文字。
- 多層次萬用字元 '#' 用來符合主題內任意數目的層次。例如，使用範例主題樹狀結構，如果您訂閱 'USA/Alaska/#'，則會收到主題 'USA/Alaska' 及 'USA/Alaska/Juneau' 的相關訊息。
- 多層次萬用字元可以代表零或多個層次。因此，'USA/#' 也可以符合單數 'USA'，其中 '#' 代表零個層次。主題層次分隔字元在此環境定義中沒有意義，因為沒有可分隔的層次。
- 只有在自行指定或在主題層次分隔字元旁指定時，多層次萬用字元才有效。因此，'#' 和 'USA/#' 是將 '#' 字元視為萬用字元的有效主題。不過，雖然 'USA#' 也是有效的主题字串，但 '#' 字元不會被視為萬用字元，也沒有任何特殊意義。如需相關資訊，請參閱第 25 頁的『當主題型萬用字元不是萬用字元時』。

## 單一層次萬用字元

- 訂閱中使用單一萬用字元。在發佈中使用時，會將它視為文字。
- 單一層次萬用字元 '+' 符合一個 (且僅符合一個) 主題層次。例如，'USA/+' 符合 'USA/Alabama'，但不符合 'USA/Alabama/Auburn'。因為單一層次萬用字元只符合單一層次，所以 'USA/+' 不符合 'USA'。
- 單一層次萬用字元可以在主題樹狀結構中的任何層次使用，並與多層次萬用字元一起使用。必須在主題層次分隔字元旁邊指定單一層次萬用字元，但自行指定時除外。因此，'+' 和 'USA/+' 是將 '+' 字元視為萬用字元的有效主題。不過，雖然 'USA+' 也是有效的主题字串，但 '+' 字元不會被視為萬用字元，也沒有任何特殊意義。如需相關資訊，請參閱第 25 頁的『當主題型萬用字元不是萬用字元時』。

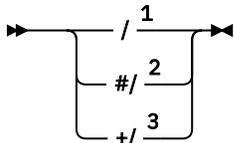
主題型萬用字元架構的語法沒有跳出字元。是否將 '#' 和 '+' 視為萬用字元視其環境定義而定。如需相關資訊，請參閱第 25 頁的『當主題型萬用字元不是萬用字元時』。

註：主題字串的開頭和結尾會以特殊方式處理。使用 '\$' 來表示字串結尾，則 '\$#/...' 是多層次萬用字元和 '\$/#/...'。是位於根目錄的空節點，後面接著多層次萬用字元。

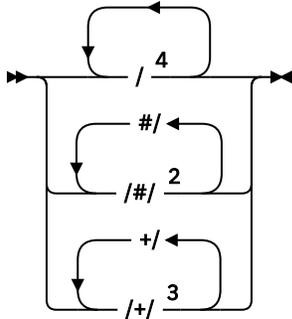


- 1 空值或零長度主題字串無效
- 2 基於字元型和主題型萬用字元架構之間的相容性，建議您不要在層次名稱字串中使用任何 \*, ?, % 。
- 3 這些觀察值相當於 左定界字元 型樣。
- 4 沒有萬用字元的 / 符合單一空主題。
- 5 這些觀察值相當於 右定界字元 型樣。
- 6 符合每個主題。
- 7 比對只有一個層次的每一個主題。

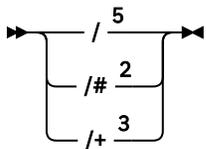
### 左定界字元



### 內文定界字元



### 右定界字元



註：

- 1 主題字串以空主題開頭
- 2 符合零個以上層次。多個多層級比對字串具有與一個多層級比對字串相同的效果。
- 3 完全符合一個層次。
- 4 // 是空的主題-沒有主題字串的主題物件。
- 5 主題字串以空主題結尾

## 當主題型萬用字元不是萬用字元時

在主題層次中與其他字元 (包括本身) 混合時，萬用字元 '+' 和 '#' 沒有特殊意義。

這表示可以發佈包含 '+' 或 '#' 以及主題層次中其他字元的主題。

例如，請考量下列兩個主題：

1. level0/level1/+/level4/#
2. level0/level1/#+/level4/level#

在第一個範例中，字元 '+' 和 '#' 被視為萬用字元，因此在要發佈至但在訂閱中有效的主題字串中無效。

在第二個範例中，字元 '+' 和 '#' 不會被視為萬用字元，因此主題字串可以同時發佈和訂閱。

## 範例

```
IBM/+/Results
#/Results
IBM/Software/Results
```

### 字元型萬用字元架構

文字萬用字元架構可讓您根據傳統字元比對來選取主題。

您可以使用字串 '\*' 來選取主題階層中多個層次的所有主題。在文字型萬用字元架構中使用 '\*' 相當於使用主題型萬用字元字串 '#'

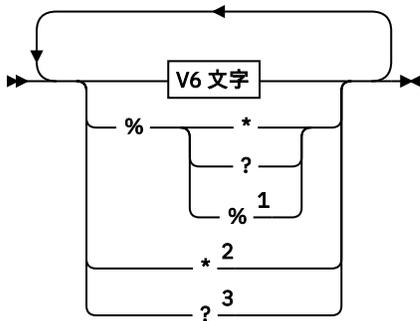
'x/\*/y' 相當於主題型架構中的 'x#/y'，並在層次 'x' 和 'y' 之間選取主題階層中的所有主題，其中 'x' 和 'y' 是不在萬用字元所傳回的層次集內的主題名稱。

主題型架構中的 '/+/' 在文字型架構中沒有確切的對等項目。'IBM/\*/Results' 也會選取 'IBM/Patents/Software/Results'。只有在階層的每一個層次上的主題名稱集都是唯一的時，您才能一律使用產生相同相符項的兩個方法來建構查詢。

以一般方式使用，文字架構中的 '\*' 及 '?' 在主題型架構中沒有對等項目。主題型架構不會使用萬用字元來執行局部比對。字元型萬用字元訂閱 'IBM/\*ware/Results' 沒有主題型對等項目。

註：使用萬用字元萬用字元訂閱的相符項比使用主題型訂閱的相符項慢。

### 字元型萬用字元字串



### V6 文字

► 除了\*? 以外的任何 Unicode 字元及 % ◄

註：

- 1 表示「跳出下列字元」，以便將它視為文字。 '%' 後面必須是 '\*'、 '?' 或 '%'。請參閱第 23 頁的『主題字串範例』。
- 2 表示訂閱中的「符合零個以上字元」。
- 3 表示訂閱中的「只符合一個字元」。

## 範例

```
IBM/*/Results
IBM/*ware/Results
```

### 結合主題字串

建立訂閱或開啟主題以向其發佈訊息時，可以結合兩個個別的子主題字串或 "子主題" 來形成主題字串。一個子主題由應用程式或管理指令以主題字串提供，另一個是與主題物件相關聯的主題字串。您可以單獨使用子主題作為主題字串，或結合子主題以形成新的主題名稱。

例如，當您使用 MQSC 指令 **DEFINE SUB** 定義訂閱時，指令可以將 **TOPICSTR** (主題字串) 及/或 **TOPICOBJ** (主題物件) 視為屬性。如果只提供 **TOPICOBJ**，則會使用與該主題物件相關聯的主題字串作為主題字串。如果只提供 **TOPICSTR**，則會使用它作為主題字串。如果同時提供兩者，則它們會連結以形成 **TOPICOBJ/**

**TOPICSTR** 格式的單一主題字串，其中 **TOPICOBJ** 配置的主題字串一律是第一個，且字串的兩個部分一律以 "/" 字元區隔。

同樣地，在 MQI 程式中，MQOPEN 會建立完整主題名稱。它由兩個在發佈/訂閱 MQI 呼叫中使用的欄位組成，依列出的順序：

1. 主題物件的 **TOPICSTR** 屬性，在 **ObjectName** 欄位中指定。
2. **ObjectString** 參數定義應用程式所提供的子主題。

產生的主題字串會在 **ResObjectString** 參數中傳回。

如果每一個欄位的第一個字元不是空白或空字元，且欄位長度大於零，則這些欄位會被視為存在。如果只存在其中一個欄位，則會使用未變更的欄位作為主題名稱。如果任一欄位都沒有值，則呼叫會失敗，原因碼為 MQRC\_UNKNOWN\_OBJECT\_NAME；如果完整主題名稱無效，則呼叫會失敗 MQRC\_TOPIC\_STRING\_ERROR。

如果這兩個欄位都存在，則會在產生的合併主題名稱的兩個元素之間插入 "/" 字元。

第 27 頁的表 2 顯示主題字串連結的範例：

主題物件的 TOPICSTR	應用程式或 DEFINE SUB 指令提供的主題字串	完整主題名稱	備註
足球/評分	' '	足球/評分	主題物件的 TOPICSTR 是單獨使用。
' '	足球/評分	足球/評分	單獨使用 ObjectString/TOPICSTR。
football	評分	足球/評分	在連結點新增 "/" 字元。
football	/Scores	足球//Scores	在兩個字串之間產生「空節點」。這與 "Football/Scores" 不同。
/Football	評分	/Football/Scores	主題以「空節點」開頭。這與 "Football/Scores" 不同。

"/" 字元被視為特殊字元，為第 28 頁的『主題樹狀結構』中的完整主題名稱提供結構。"/" 字元不得用於任何其他原因，因為主題樹狀結構會受到影響。主題 "/Football" 與主題 "Football" 不同。

註：如果您在建立訂閱時使用主題物件，則會在定義訂閱時修正主題物件主題字串的值。對主題物件所做的任何後續變更都不會影響對其定義訂閱的主題字串。

## 主題字串中的萬用字元

下列萬用字元是特殊字元：

- 加號 (+)
- # 記號 (#)
- 星號 (\*)
- 問號 (?)

只有在訂閱使用時，萬用字元才具有特殊意義。這些字元在其他位置使用時不會被視為無效，不過您必須確定您瞭解如何使用它們，且在發佈或定義主題物件時，您可能不想要在主題字串中使用這些字元。

如果您在主題層次內發佈主題字串，且 # 或 + 與其他字元 (包括本身) 混合，則可以使用任一萬用字元架構來訂閱主題字串。

如果您在主題字串上發佈，且 # 或 + 是兩個 / 字元之間的唯一字元，則應用程式無法使用萬用字元架構 MQSO\_WILDCARD\_TOPIC 來明確訂閱主題字串。此狀況會導致應用程式取得超出預期的發佈數。

您不應在已定義的主題物件的主題字串中使用萬用字元。如果您這樣做，當發佈者使用物件時，該字元會被視為文字字元，當訂閱使用時，該字元會被視為萬用字元。這可能會造成混淆。

## 程式碼 Snippet 範例

此程式碼 Snippet 擷取自範例程式 [Example 2: Publisher to a variable topic](#)，結合主題物件與變數主題字串：

```
MQOD      td = {MQOD_DEFAULT}; /* Object Descriptor          */
td.ObjectType = MQOT_TOPIC; /* Object is a topic    */
td.Version = MQOD_VERSION_4; /* Descriptor needs to be V4 */
strncpy(td.ObjectName, topicName, MQ_TOPIC_NAME_LENGTH);
td.ObjectString.VSPtr = topicString;
td.ObjectString.VSLength = (MQLONG)strlen(topicString);
td.ResObjectString.VSPtr = resTopicStr;
td.ResObjectString.VSBufSize = sizeof(resTopicStr)-1;
MQOPEN(Hconn, &td, MQOO_OUTPUT | MQOO_FAIL_IF_QUIESCING, &Hobj, &CompCode, &Reason);
```

## 主題樹狀結構

您定義的每一個主題就是主題樹狀結構中的一個元素或節點。主題樹狀結構可以是空的，以開始或包含先前使用 MQSC 或 PCF 指令定義的主題。您可以定義新主題，方式是使用「建立主題」指令，亦或在發佈或訂閱中第一次指定該主題。

雖然您可以使用任何字串來定義主題的主題字串，但建議您選擇適合階層式樹狀結構的主題字串。深入設計主題字串和主題樹狀結構可協助您執行下列作業：

- 訂閱多個主題。
- 建立安全原則。

雖然您可以將主題樹狀結構建構為平面線性結構，但最好以具有一個以上根主題的階層式結構來建置主題樹狀結構。如需安全規劃和主題的相關資訊，請參閱 [發佈/訂閱安全](#)。

第 28 頁的圖 5 顯示具有一個根主題的主題樹狀結構範例。

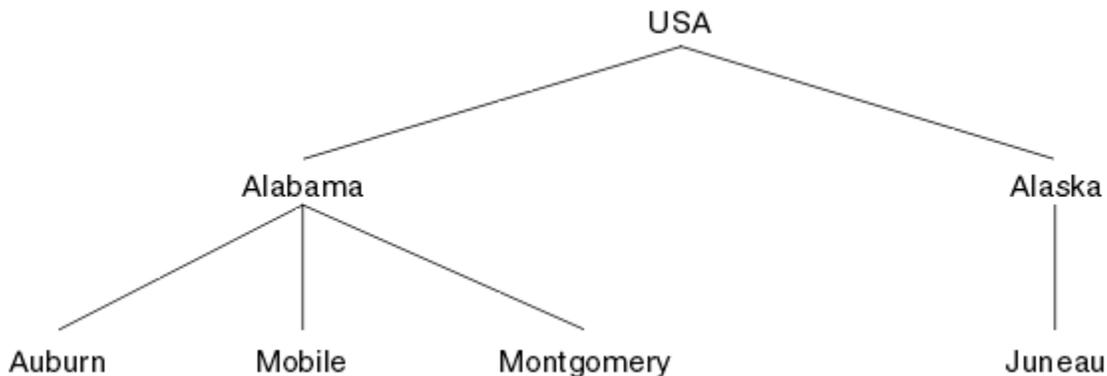


圖 5: 主題樹狀結構範例

圖中的每一個字串都代表主題樹狀結構中的一個節點。透過從主題樹狀結構中的一個以上層次聚集節點來建立完整主題字串。層次以 "/" 字元區隔。完整指定的主題字串格式為："root/level2/level3"。

第 28 頁的圖 5 所顯示主題樹狀結構中的有效主題如下：

```
"美國"
"美國/阿拉巴馬"
"美國/阿拉斯加"
"USA/Alabama/Auburn"
"USA/Alabama/Mobile"
"USA/Alabama/Montgomery"
"美國/阿拉斯加/朱諾"
```

當您設計主題字串和主題樹狀結構時，請記住佇列管理程式不會解譯或嘗試從主題字串本身衍生意義。它只會使用主題字串，將選取的訊息傳送給該主題的訂閱者。

下列原則適用於主題樹狀結構的建構和內容:

- 主題樹狀結構中的層次數沒有限制。
- 主題樹狀結構中層次名稱的長度沒有限制。
- 可以有任意數目的「根」節點; 亦即, 可以有任意數目的主題樹狀結構。

減少主題樹狀結構中不想要的主题數目

透過減少主題樹狀結構中不想要的主题數目, 可改善發佈/訂閱系統的效能。什麼是不想要的主题, 以及如何移除它們?

您可以建立大量主题, 而不會對效能產生不利影響。不過, 某些使用發佈/訂閱的方式會導致持續展開主题樹狀結構。異常大量的主题會建立一次, 且不再使用。越來越多的主题可能會變成效能問題。

如何避免設計導致大量且不斷增加的不想要主题? 您可以做什麼來協助佇列管理程式從主题樹狀結構中移除不想要的主题?

佇列管理程式可辨識不想要的主题, 因為它已 30 分鐘未使用。佇列管理程式會為您從主题樹狀結構中移除未用的主题。您可以變更佇列管理程式屬性 **TREELIFE** 來變更 30 分鐘的持續時間。您可以協助佇列管理程式移除不想要的主题, 方法是確定該主题對佇列管理程式顯示為未用。第 29 頁的『何謂未使用的主题?』一節說明未用的主题是什麼。

程式設計師, 設計任何應用程式, 特別是設計長時間執行的應用程式, 會考量其資源用量: 程式需要多少資源, 是否有任何無限制的需求, 以及任何資源洩漏? 主题是發佈/訂閱程式使用的資源。如同程式所使用的任何其他資源一樣, 仔細檢查主题的使用。

## 何謂未使用的主题?

在定義未使用的主题之前, 哪些確實是主题?

當主题字串 (例如 USA/Alabama/Auburn) 轉換為主题時, 該主题會新增至主题樹狀結構。必要的話, 會在樹狀結構中建立其他主题節點及其對應的主题。主题字串 USA/Alabama/Auburn 會轉換成具有三個主题的樹狀結構。

- USA
- USA/Alabama
- USA/Alabama/Auburn

若要顯示主题樹狀結構中的所有主题, 請使用 **runmqsc** 指令 `DISPLAY TPSTATUS('#') TYPE(TOPIC)`。

主题樹狀結構中未用的主题具有下列內容。

### 它未與主题物件相關聯

管理主题物件具有將其與主题相關聯的主题字串。當您定義主题物件 Alabama 時, 如果要關聯的主题 USA/Alabama 不存在, 則會從主题字串建立主题。如果主题確實存在, 則會使用主题字串將主题物件與主题關聯在一起。

### 它沒有保留的發佈

具有保留發佈資訊的主题是來自發佈者使用 `MQPMO_RETAIN` 選項將訊息放置到主题。

使用 **runmqsc** 指令 `DISPLAY TPSTATUS('USA/Alabama') RETAINED` 來檢查 USA/Alabama 是否具有保留的發佈資訊。回應為 YES 或 NO。

使用 **runmqsc** 指令 `CLEAR TOPICSTR('USA/Alabama') CLTRTYPE(RETAINED)`, 從 USA/Alabama 中移除保留的發佈資訊。

### 它沒有子主题

USA/Alabama/Auburn 是沒有子主题的主题。USA/Alabama/Auburn 是 USA/Alabama 的直屬子主题。

使用 **runmqsc** 指令 `DISPLAY TPSTATUS('USA/Alabama+')` 顯示 USA/Alabama 的直屬子項。

## 節點沒有作用中的發佈者

節點的作用中發佈者是開啟主題以供輸出的應用程式。

例如，應用程式以開啟選項 MQ00\_OUTPUT 開啟名為 **Alabama** 的主題物件。

若要顯示 USA/Alabama 及其所有子項的作用中發佈者，請使用 **runmqsc** 指令 `DISPLAY TPSTATUS('USA/Alabama/#') TYPE(PUB) ACTCONN`。

## 節點沒有作用中的訂閱者

作用中訂閱者可以是可延續訂閱，或是已向 MQSUB 登錄主題訂閱且未關閉它的應用程式。

若要顯示 USA/Alabama 的作用中訂閱，請使用 **runmqsc** 指令 `DISPLAY TPSTATUS('USA/Alabama') TYPE(SUB) ACTCONN`。

若要顯示 USA/Alabama 及其所有子項的作用中訂閱，請使用 **runmqsc** 指令 `DISPLAY TPSTATUS('USA/Alabama/#') TYPE(SUB) ACTCONN`。

## 減少主題樹狀結構中的主題數

總之，有許多方法可以減少主題樹狀結構中的主題數目。

### 修改 TREELIFE

依預設，未用的主題的生命期限為 30 分鐘。您可以減少未用主題的生命期限。

例如，**runmqsc** 指令 `ALTER QMGR TREELIFE(900)` 會將未用主題的生命期限從 30 分鐘減少至 15 分鐘。

### 異常地重新啟動佇列管理程式

當佇列管理程式重新啟動時，會從主題物件、具有保留發佈資訊的節點及可延續訂閱重新起始設定主題樹狀結構。已刪除發佈者及訂閱者程式作業所建立的主題。

請定期使用 **runmqsc** 指令 `DISPLAY TPSTATUS('#') TYPE(TOPIC)` 來列出所有主題，並檢查數目是否在增加。

作為最後手段，如果不想要的主題中的成長已成為過去效能問題的原因，請重新啟動佇列管理程式。

## 管理主題物件

使用管理主題物件，您可以將特定非預設屬性指派給主題。

第 30 頁的圖 6 顯示如何將 Sport 的高階主題分成涵蓋不同體育項目的個別主題視覺化為主題樹狀結構：

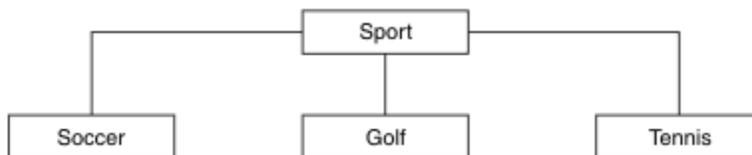


圖 6: 主題樹狀結構的視覺化

第 30 頁的圖 7 顯示如何進一步分割主題樹狀結構，以區隔每一個運動的不同類型相關資訊：



圖 7: 延伸主題樹狀結構

如果要建立圖解的主題樹狀結構，不需要定義任何管理主題物件。此樹狀結構中的每一個節點都是由發佈或訂閱作業中所建立的主題字串所定義。樹狀結構中的每一個主題都會從其母項繼承其屬性。屬性繼承自上

層主題物件，因為依預設所有屬性都設為 ASPARENT。在此範例中，每個主題都具有與 Sport 主題相同的屬性。Sport 主題沒有管理主題物件，並從 SYSTEM.BASE.TOPIC。

請注意，在主題樹狀結構的根節點 (即 SYSTEM.BASE.TOPIC，因為權限是繼承的，但無法限制。因此，透過在此層次授與權限，您會將權限授與整個樹狀結構。您應該在階層中的較低主題層次提供權限。

管理主題物件可用來定義主題樹狀結構中特定節點的特定屬性。在下列範例中，管理主題物件定義為將足球主題的可延續訂閱內容 DURSUB 設為值 NO:

```
DEFINE TOPIC(FOOTBALL.EUROPEAN)
  TOPICSTR('Sport/Soccer')
  DURSUB(NO)
  DESCR('Administrative topic object to disallow durable subscriptions')
```

現在可以將主題樹狀結構視覺化為:

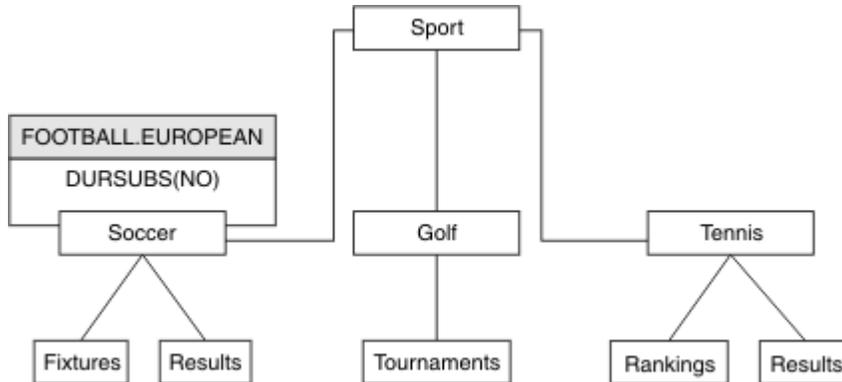


圖 8: 與 Sport/Soccer 主題相關聯之管理主題物件的視覺化

在樹狀結構中，訂閱 Soccer 之下主題的任何應用程式，在新增管理主題物件之前，仍然可以使用它們所使用的主題字串。不過，現在可以使用物件名稱 FOOTBALL.EUROPEAN 而非字串 /Sport/Soccer 來撰寫應用程式以訂閱。例如，若要訂閱 /Sport/Soccer/Results，應用程式可以將 MQSD.ObjectName 指定為 FOOTBALL.EUROPEAN，並將 MQSD.ObjectString 指定為 Results。

使用此特性，您可以對應用程式開發人員隱藏部分主題樹狀結構。在主題樹狀結構中的特定節點上定義管理主題物件，然後應用程式開發人員可以將他們自己的主題定義為節點的子項。開發人員必須知道上層主題，但不能知道上層樹狀結構中的任何其他節點。

## 繼承屬性

如果主題樹狀結構有許多管理主題物件，依預設，每一個管理主題物件會從其最近的上層管理主題繼承其屬性。前一個範例已在第 32 頁的圖 9 中延伸:

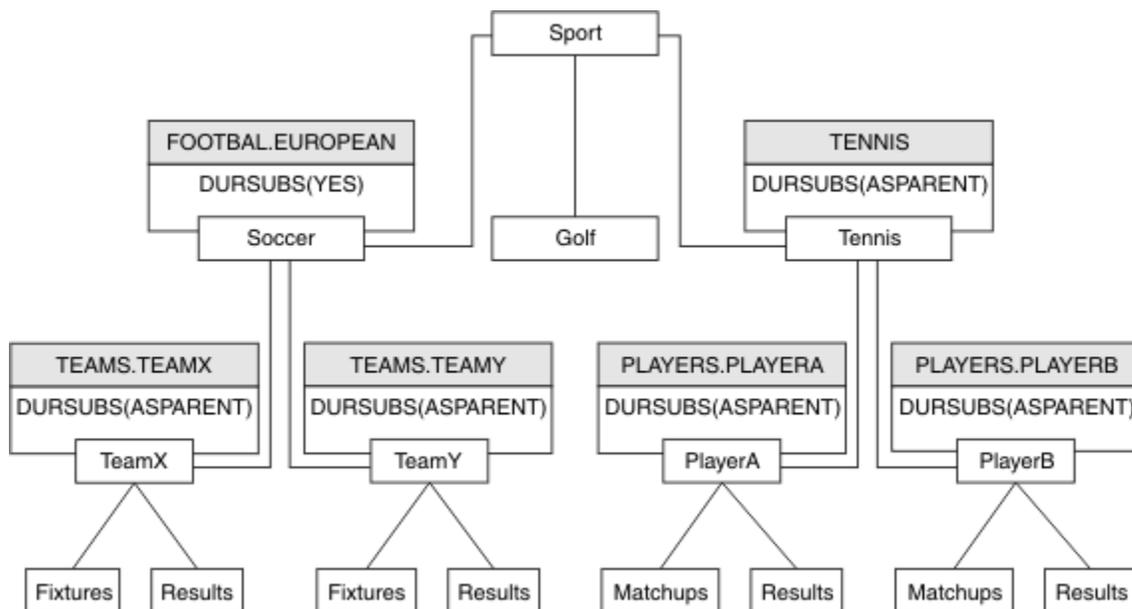


圖 9: 含有數個管理主題物件的主題樹狀結構

例如，使用繼承，為 /Sport/Soccer 的所有子主題提供訂閱不可延續的內容。將 FOOTBALL.EUROPEAN 的 DURSUB 屬性變更為 NO。

可以使用下列指令來設定此屬性：

```
ALTER TOPIC(FOOTBALL.EUROPEAN) DURSUB(NO)
```

Sport/Soccer 子主題的所有管理主題物件，都會將 DURSUB 內容設為預設值 ASPARENT。將 FOOTBALL.EUROPEAN 的 DURSUB 內容值變更為 NO 之後，Sport/Soccer 的子主題會繼承 DURSUB 內容值 NO。Sport/Tennis 的所有子主題都會從 SYSTEM.BASE.TOPIC 物件繼承 DURSUB 的值。SYSTEM.BASE.TOPIC 的值為 YES。

嘗試對主題 Sport/Soccer/TeamX/Results 進行可延續訂閱現在會失敗；不過，嘗試對 Sport/Tennis/PlayerB/Results 進行可延續訂閱將會成功。

## 使用 萬用字元 內容控制萬用字元用法

使用 MQSC **Topic** 萬用字元 內容或對等 PCF Topic WildcardOperation 內容來控制將發佈遞送至使用萬用字元主題字串名稱的訂閱者應用程式。WILDCARD 內容可以具有下列兩個可能值之一：

### WILDCARD

關於此主題的萬用字元訂閱的行為。

### PASSTHRU

對於比此主題物件的主題字串更不具體的萬用字元式主題所做的訂閱，將接收到對此主題以及比此主題更具體的主題字串所進行的發佈。

### BLOCK

對於比此主題物件的主題字串更不具體的萬用字元式主題所做的訂閱，不會接收到對此主題或比此主題更具體的主題字串所進行的發佈。

在定義訂閱時將使用此屬性的值。如果變更此屬性，則現有訂閱所涵蓋的主題集不會因為此修改而受到影響。如果在建立或刪除主題物件時拓撲發生變更，也適用此實務範例；將使用修改後的拓撲來建立與 WILDCARD 屬性修改後建立的訂閱相符的主題集。若要針對現有訂閱強制重新評估相符的主題集，則必須重新啟動佇列管理程式。

在範例 第 36 頁的『範例: 建立 Sport 發佈/訂閱叢集』中，您可以遵循步驟來建立 第 33 頁的圖 10 中所示的主題樹狀結構。

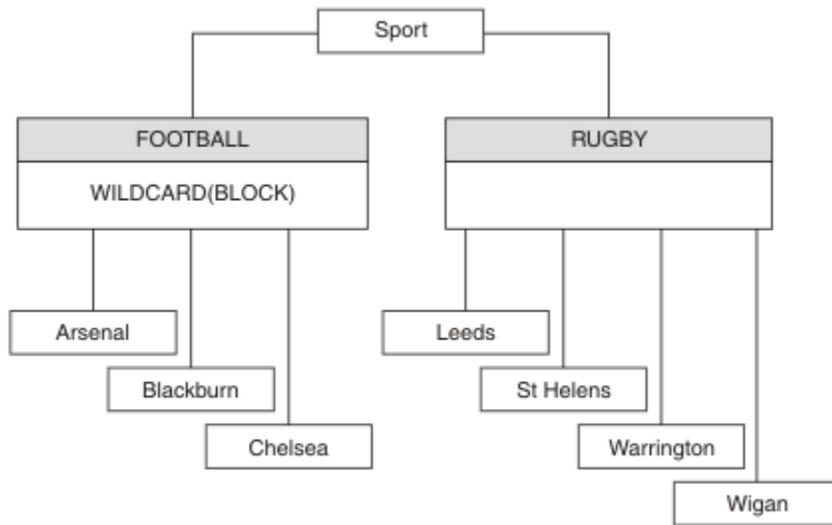


圖 10: 使用 萬用字元 內容 BLOCK 的主題樹狀結構

使用萬用字元主題字串 # 的訂閱者會接收 Sport 主題和 Sport/Rugby 子樹狀結構的所有發佈。訂閱者不會收到 Sport/Football 子樹狀結構的發佈，因為 Sport/Football 主題的 萬用字元 內容值是 BLOCK。

PASSTHRU 是預設值。您可以將 萬用字元 內容值 PASSTHRU 設為 Sport 樹狀結構中的節點。如果節點沒有 萬用字元 內容值 BLOCK，則設定 PASSTHRU 不會變更訂閱者對 Sports 樹狀結構中節點所觀察到的行為。

在範例中，建立訂閱以查看萬用字元設定如何影響遞送的發佈；請參閱 第 37 頁的圖 14。在 第 38 頁的圖 17 中執行發佈指令，以建立部分發佈。

pub QMA

圖 11: 發佈至 QMA

結果顯示在 第 33 頁的表 3 中。請注意如何設定 WILDCARD 內容值 BLOCK，防止使用萬用字元的訂閱接收對萬用字元範圍內主題的發佈。

訂閱	主題字串	收到的出版物	附註
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Sports/Football 上的 WILDCARD(BLOCK) 封鎖所有 Football 子樹狀結構的發佈
SARSENAL	Sports/#/Arsenal	-	WILDCARD(BLOCK) on Sports/Football 阻止在 Arsenal 上進行萬用字元訂閱
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Sports/Rugby 上的預設 WILDCARD 不會阻止 Leeds 上的萬用字元訂閱。

**註:**

假設訂閱具有萬用字元，其符合具有 WILDCARD 內容值 BLOCK 的主題物件。如果訂閱在相符萬用字元的右側也有主題字串，則訂閱永遠不會收到發佈。未封鎖的發佈資訊集是屬於已封鎖萬用字元母項之主題的發佈資訊。作為主題子項且具有 BLOCK 內容值的主題的發佈會被萬用字元封鎖。因此，如果訂閱主題字串包含萬用字元右側的主題，則永遠不會收到任何要符合的發佈。

將 WILDCARD 內容值設為 BLOCK 並不表示您無法使用包含萬用字元的主題字串來訂閱。這類訂閱是正常的。訂閱有一個明確主題符合主題，且主題物件具有 WILDCARD 內容值 BLOCK。對於作為主題母項或子項且具有 萬用字元 內容值 BLOCK 的主題，它會使用萬用字元。在 [第 33 頁的圖 10](#) 的範例中，Sports/Football/# 之類的訂閱可以接收發佈。

## 萬用字元和叢集主題

叢集主題定義會延伸到叢集中的每個佇列管理程式。訂閱叢集中某個佇列管理程式的叢集主題會導致佇列管理程式建立 Proxy 訂閱。在叢集中的所有其他佇列管理程式上建立 Proxy 訂閱。使用包含萬用字元且與叢集主題結合的主題字串的訂閱，可能會提供難以預測的行為。此行為在下列範例中說明。

在針對範例 [第 36 頁的『範例: 建立 Sport 發佈/訂閱叢集』](#) 所設定的叢集中，QMB 具有與 QMA 相同的訂閱集，但 QMB 在發佈者發佈至 QMA 之後未收到任何發佈，請參閱 [第 33 頁的圖 11](#)。雖然 Sports/Football 及 Sports/Rugby 主題是叢集主題，但 fullsubs.tst 中定義的訂閱不會參照叢集主題。沒有任何 Proxy 訂閱從 QMB 延伸到 QMA。如果沒有 Proxy 訂閱，則不會將任何對 QMA 的發佈轉遞至 QMB。

部分訂閱 (例如 Sports/#/Leeds) 似乎參照叢集主題，在此情況下為 Sports/Rugby。Sports/#/Leeds 訂閱實際上會解析為主題物件 SYSTEM.BASE.TOPIC。

用於解析訂閱 (例如 Sports/#/Leeds) 所參照的主題物件的規則如下。將主題字串截斷為第一個萬用字元。掃描左側的主題字串，以尋找具有相關聯管理主題物件的第一個主題。主題物件可以指定叢集名稱，或定義本端主題物件。在範例 Sports/#/Leeds 中，截斷之後的主題字串是 Sports，它沒有主題物件，因此 Sports/#/Leeds 繼承自 SYSTEM.BASE.TOPIC，它是本端主題物件。

若要查看訂閱叢集主題如何變更萬用字元傳播的運作方式，請執行批次 Script upsubs.bat。該 Script 會清除訂閱佇列，並在 fullsubs.tst 中新增叢集主題訂閱。再次執行 puba.bat 以建立一批出版品；請參閱 [第 33 頁的圖 11](#)。

[第 34 頁的表 4](#) 顯示將兩個新訂閱新增至發佈者所在之相同佇列管理程式的結果。結果如預期，新訂閱各會收到一個發佈，而其他訂閱所收到的發佈數則維持不變。其他叢集佇列管理程式上發生非預期的結果；請參閱 [第 35 頁的表 5](#)。

訂閱	主題字串	收到的出版物	附註
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Sports/Football 上的 WILDCARD (BLOCK) 封鎖所有 Football 子樹狀結構的發佈
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) on Sports/Football 阻止在 Arsenal 上進行萬用字元訂閱
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Sports/Rugby 上的預設 WILDCARD 不會阻止 Leeds 上的萬用字元訂閱。
FARSENAL	Sports/Football/Arsenal	Sports/Football/Arsenal	Arsenal 會接收發佈，因為訂閱沒有萬用字元。
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds 會在任何事件中收到發佈。

[第 35 頁的表 5](#) 顯示在 QMB 上新增兩個新訂閱，以及在 QMA 上發佈的結果。回想一下，如果沒有這兩個新訂閱，QMB 不會收到任何發佈。如預期，這兩個新的訂閱會接收發佈，因為 Sports/FootBall 和 Sports/Rugby 都是叢集主題。QMB 已將 Sports/Football/Arsenal 和 Sports/Rugby/Leeds 的 Proxy 訂閱轉遞至 QMA，然後將發佈傳送至 QMB。

非預期的結果是先前未收到任何發佈的兩個訂閱 Sports/# 及 Sports/#/Leeds 現在已收到發佈。原因是其他訂閱的 Sports/Football/Arsenal 及 Sports/Rugby/Leeds 轉遞至 QMB 的發佈現在可用於連接至 QMB 的任何訂閱者。因此，對本端主題 Sports/# 及 Sports/#/Leeds 的訂閱會收到 Sports/Rugby/Leeds 發佈。Sports/#/Arsenal 會繼續不接收發佈，因為「體育/足球」已將其 萬用字元 內容值設為 BLOCK。

表 5: 在 QMB 上收到的出版物

訂閱	主題字串	收到的出版物	附註
SPORTS	Sports/#	Sports/Rugby/Leeds	WILDCARD (BLOCK) 在 Sports/Football 上封鎖所有發佈至 Football 子樹狀結構
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) on Sports/Football 阻止在 Arsenal 上進行萬用字元訂閱
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Sports/Rugby 上的預設 WILDCARD 不會阻止 Leeds 上的萬用字元訂閱。
FARSENAL	Sports/Football/Arsenal	Sports/Football/Arsenal	Arsenal 會接收發佈，因為訂閱沒有萬用字元。
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds 會在任何事件中收到發佈。

在大部分應用程式中，一個訂閱不想要影響另一個訂閱的行為。WILDCARD 內容與值 BLOCK 的一個重要用途是讓相同主題字串 (包含萬用字元) 的訂閱統一運作。不論訂閱位於與發佈者相同的佇列管理程式上，或位於不同的佇列管理程式上，訂閱的結果都相同。

## 萬用字元和串流

WebSphere MQ 第 6 版串流由 WebSphere MQ 第 7 版對映至主題; 請參閱第 38 頁的『串流和主題』。在預設對映中 (在第 7 版中由 **strmqbrk** 執行)，串流 Sports 中的所有主題都會對映至主題 Sports。串流 Business 中的所有主題都將對映至主題 Business。

Sports 串流中 WebSphere MQ 第 6 版至 \* 的訂閱會接收 Sports 樹狀結構中的所有發佈，而 Business 樹狀結構中沒有發佈。第 7 版中的相同訂閱會接收 Sports 樹狀結構中的所有發佈，以及 Business 樹狀結構中的所有發佈。若要封鎖此行為，當串流移轉至第 7 版時，**strmqbrk** 會設定萬用字元內容。對於從串流移轉的每一個最上層主題，它會將它設為值 BLOCK。透過從稱為 Sports 和 Business 的第 6 版串流進行轉換，將 Sports 和 Business 的萬用字元內容設定為值 BLOCK。

對於寫入至發佈/訂閱 API 的新應用程式，其效果是 \* 的訂閱不會收到任何發佈。若要接收所有「體育」發佈資訊，您必須訂閱 Sports/\* 或 Sports/#，以及類似的 Business 發佈資訊。

當發佈/訂閱分配管理系統移轉至 WebSphere MQ 第 7 版時，現有已排入佇列的發佈/訂閱應用程式的行為不會變更。**Publish**、**Register Publisher** 或 **Subscriber** 指令中的 **StreamName** 內容對映至已移轉串流的主題名稱。

## 萬用字元和訂閱點

WebSphere Message Broker 訂閱點由 WebSphere MQ 第 7 版對映至主題; 請參閱第 40 頁的『訂閱點和主題』。在預設對映中 (由第 7 版中的 **migmqbrk** 執行)，訂閱點 Sports 中的所有主題都會對映至主題 Sports。訂閱點 Business 中的所有主題都會對映至主題 Business。

Sports 訂閱點中 WebSphere Message Broker 第 6 版至 \* 的訂閱會接收 Sports 樹狀結構中的所有發佈，而 Business 樹狀結構中沒有發佈。第 7 版中的相同訂閱會接收 Sports 樹狀結構中的所有發佈，以及 Business 樹狀結構中的所有發佈。若要封鎖此行為，當訂閱點移轉至第 7 版時，**migmqbrk** 會設定萬用字元內容。它會針對從訂閱點移轉的每一個最上層主題，將它設為值 BLOCK。透過從稱為 Sports 及 Business 的 WebSphere Message Broker 訂閱點進行轉換，將 Sports 及 Business 的萬用字元內容設為值 BLOCK。

對於寫入發佈/訂閱 API 的新應用程式，移轉的效果是 \* 的訂閱不會收到任何發佈。若要接收所有「體育」發佈資訊，您必須訂閱 Sports/\* 或 Sports/#，以及類似的 Business 發佈資訊。

當發佈/訂閱分配管理系統移轉至 WebSphere MQ 第 7 版時，現有已排入佇列的發佈/訂閱應用程式的行為不會變更。**Publish**、**Register Publisher** 或 **Subscriber** 指令中的 **SubPoint** 內容對映至已移轉訂閱的主題名稱。

## 範例: 建立 Sport 發佈/訂閱叢集

接下來的步驟會建立叢集 CL1，其中包含四個佇列管理程式: 兩個完整儲存庫、CL1A 和 CL1B，以及兩個局部儲存庫 QMA 和 QMB。完整儲存庫用來只保留叢集定義。QMA 指定為叢集主題主機。可延續訂閱同時定義在 QMA 和 QMB 上。

註: 此範例是針對 Windows 撰寫的。您必須重新編碼 建立 qmgrs.bat 及 建立 pub.bat，以在其他平台上配置及測試範例。

1. 建立 Script 檔。
  - a. 建立 topics.tst
  - b. 建立 wildsubs.tst
  - c. 建立 fullsubs.tst
  - d. 建立 qmgrs.bat
  - e. 建立 pub.bat
2. 執行 Create qmgrs.bat 以建立配置。

```
qmgrs
```

在 [第 33 頁的圖 10](#) 中建立主題。圖 5 中的 Script 會建立叢集主題 Sports/Football 和 Sports/Rugby。

註: REPLACE 選項不會取代主題的 TOPICSTR 內容。TOPICSTR 是在範例中用來測試不同主題樹狀結構的有用內容。若要變更主題，請先刪除主題。

```
DELETE TOPIC ('Sports')
DELETE TOPIC ('Football')
DELETE TOPIC ('Arsenal')
DELETE TOPIC ('Blackburn')
DELETE TOPIC ('Chelsea')
DELETE TOPIC ('Rugby')
DELETE TOPIC ('Leeds')
DELETE TOPIC ('Wigan')
DELETE TOPIC ('Warrington')
DELETE TOPIC ('St. Helens')

DEFINE TOPIC ('Sports') TOPICSTR('Sports')
DEFINE TOPIC ('Football') TOPICSTR('Sports/Football') CLUSTER(CL1) WILDCARD(BLOCK)
DEFINE TOPIC ('Arsenal') TOPICSTR('Sports/Football/Arsenal')
DEFINE TOPIC ('Blackburn') TOPICSTR('Sports/Football/Blackburn')
DEFINE TOPIC ('Chelsea') TOPICSTR('Sports/Football/Chelsea')
DEFINE TOPIC ('Rugby') TOPICSTR('Sports/Rugby') CLUSTER(CL1)
DEFINE TOPIC ('Leeds') TOPICSTR('Sports/Rugby/Leeds')
DEFINE TOPIC ('Wigan') TOPICSTR('Sports/Rugby/Wigan')
DEFINE TOPIC ('Warrington') TOPICSTR('Sports/Rugby/Warrington')
DEFINE TOPIC ('St. Helens') TOPICSTR('Sports/Rugby/St. Helens')
```

圖 12: 刪除並建立主題: topics.tst

註: 刪除主題，因為 REPLACE 不會取代主題字串。

使用萬用字元建立訂閱。萬用字元對應主題與 [第 33 頁的圖 10](#) 中的主題物件。為每一個訂閱建立佇列。執行或重新執行 Script 時，會清除佇列並刪除訂閱。

註: REPLACE 選項不會取代訂閱的 TOPICOBJ 或 TOPICSTR 內容。TOPICOBJ 或 TOPICSTR 是在測試不同訂閱的範例中有效改變的內容。若要變更它們，請先刪除訂閱。

```

DEFINE QLOCAL(QSPORTS) REPLACE
DEFINE QLOCAL(QSARSENAL) REPLACE
DEFINE QLOCAL(QSLEEDS) REPLACE
CLEAR QLOCAL(QSPORTS)
CLEAR QLOCAL(QSARSENAL)
CLEAR QLOCAL(QSLEEDS)

DELETE SUB (SPORTS)
DELETE SUB (SARSENAL)
DELETE SUB (SLEEDS)
DEFINE SUB (SPORTS) TOPICSTR('Sports/#') DEST(QSPORTS)
DEFINE SUB (SARSENAL) TOPICSTR('Sports+/Arsenal') DEST(QSARSENAL)
DEFINE SUB (SLEEDS) TOPICSTR('Sports+/Leeds') DEST(QSLEEDS)

```

圖 13: 建立萬用字元訂閱: *wildsubs.tst*

建立參照叢集主題物件的訂閱。

**註:**

定界字元 / 會自動插入 TOPICOBJ 所參照的主題字串與 TOPICSTR 所定義的主題字串之間。

DEFINE SUB(FARSENAL) TOPICSTR('Sports/Football/Arsenal') DEST(QFARSENAL) 定義會建立相同的訂閱。TOPICOBJ 是用來作為參照您已定義之主題字串的快速方式。建立的訂閱不再參照主題物件。

```

DEFINE QLOCAL(QFARSENAL) REPLACE
DEFINE QLOCAL(QRLEEDS) REPLACE
CLEAR QLOCAL(QFARSENAL)
CLEAR QLOCAL(QRLEEDS)

DELETE SUB (FARSENAL)
DELETE SUB (RLEEDS)
DEFINE SUB (FARSENAL) TOPICOBJ('Football') TOPICSTR('Arsenal') DEST(QFARSENAL)
DEFINE SUB (RLEEDS) TOPICOBJ('Rugby') TOPICSTR('Leeds') DEST(QRLEEDS)

```

圖 14: 刪除並建立訂閱: *fullsubs.tst*

建立具有兩個儲存庫的叢集。建立兩個局部儲存庫以進行發佈和訂閱。請重新執行 Script 以刪除所有項目，然後重新啟動。Script 也會建立主題階層及起始萬用字元訂閱。

**註:**

在其他平台上，撰寫類似的 Script，或輸入所有指令。使用 Script 可讓您快速刪除所有項目，並以相同的配置重新開始。

```

@echo off
set port.CL1B=1421
set port.CL1A=1420
for %%A in (CL1A CL1B QMA QMB) do call :createQM %%A
call :configureQM CL1A CL1B %port.CL1B% full
call :configureQM CL1B CL1A %port.CL1A% full
for %%A in (QMA QMB) do call :configureQM %%A CL1A %port.CL1A% partial
for %%A in (topics.tst wildsubs.tst) do runmqsc QMA < %%A
for %%A in (wildsubs.tst) do runmqsc QMB < %%A
goto:eof

:createQM
echo Configure Queue manager %1
endmqm -p %1
for %%B in (dlt crt str) do %%Bmqm %1
goto:eof

:configureQM
if %1==CL1A set p=1420
if %1==CL1B set p=1421
if %1==QMA set p=1422
if %1==QMB set p=1423
echo configure %1 on port %p% connected to repository %2 on port %3 as %4 repository
echo DEFINE LISTENER(LST%1) TRPTYPE(TCP) PORT(%p%) CONTROL(QMGR) REPLACE | runmqsc %1
echo START LISTENER(LST%1) | runmqsc %1
if full==%4 echo ALTER QMGR REPOS(CL1) DEADQ(SYSTEM.DEAD.LETTER.QUEUE) | runmqsc %1
echo DEFINE CHANNEL(TO.%2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('LOCALHOST(%3)') CLUSTER(CL1)
REPLACE | runmqsc %1
echo DEFINE CHANNEL(TO.%1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('LOCALHOST(%p%)')
CLUSTER(CL1) REPLACE | runmqsc %1
goto:eof

```

圖 15: 建立佇列管理程式: *qmgrs.bat*

透過將訂閱新增至叢集主題來更新配置。

```

@echo off
for %%A in (QMA QMB) do runmqsc %%A < wildsubs.tst
for %%A in (QMA QMB) do runmqsc %%A < upsubs.tst

```

圖 16: 更新訂閱: *upsubs.bat*

執行 *pub.bat*，並以佇列管理程式作為參數，以發佈包含發佈主題字串的訊息。 *Pub.bat* 使用範例程式 **amqspub**。

```

@echo off
@rem Provide queue manager name as a parameter
set S=Sports
set S=6 Sports/Football Sports/Football/Arsenal
set S=6 Sports/Rugby Sports/Rugby/Leeds
for %%B in (6) do echo %%B | amqspub %%B %1

```

圖 17: 發佈: *pub.bat*

## 串流和主題

已排入佇列的發佈/訂閱具有發佈串流的概念，該發佈串流不存在於整合的發佈/訂閱模型中。在排入佇列的發佈/訂閱中，串流提供一種方式來區隔不同主題的資訊流程。在 IBM WebSphere MQ Version 6.0 中，串流實作為佇列，在支援該串流的每一個分配管理系統上定義。每一個佇列都有相同的名稱（串流的名稱）。從 IBM WebSphere MQ Version 7.0 開始，串流會實作為最上層主題，可透過管理方式對映至不同的主題 ID。

會針對網路上的所有分配管理系統及佇列管理程式自動設定預設串流 **SYSTEM.BROKER.DEFAULT.STREAM**，並且不需要其他配置即可使用預設串流。將預設串流視為未命名的預設主題空間。發佈至預設串流的主題會立即提供給所有已連接的 Version 6.0 分配管理系統，以及從 Version 7.0 開始的所有佇列管理程式，並已啟用排入佇列的發佈/訂閱。具名串流類似於個別具名主題空間。指定的串流必須定義在使用它的每一個分配管理系統上。

當您定義主題時，Version 6.0 發佈/訂閱分配管理系統以及在較新版本 IBM WebSphere MQ 上執行的發佈者和訂閱者都可以使用該主題，而不需要特殊配置。

如果發佈者和訂閱者位於不同的佇列管理程式上，則在相同的分配管理系統階層中連接分配管理系統之後，不需要進一步配置發佈，以及在它們之間流動的訂閱。同樣的互操作性也是相反的。

## 具名串流

使用排入佇列的發佈/訂閱程式設計模型的解決方案設計程式，可能會決定將所有體育發佈放入名為 Sport 的具名串流中。在 Version 6.0 中，通常會將串流自動抄寫到使用模型佇列 SYSTEM.BROKER.MODEL.STREAM 的其他分配管理系統。不過，如果要讓在 Version 7.0 及更新版本上執行且已啟用佇列發佈/訂閱的佇列管理程式可以使用串流，則必須手動新增該串流。

如果您從 Version 6.0 移轉佇列管理程式，請執行指令 **strmqbrk** 將 Version 6.0 指名的串流移轉至主題。串流 Sport 對映至主題 Sport。這不適用於 z/OS。

在串流 Sport 上訂閱 Soccer/Results 的已排入佇列發佈/訂閱應用程式會運作而不會變更。使用 MQSUB 訂閱主題 Sport，並提供主題字串 Soccer/Results 的整合發佈/訂閱應用程式也會收到相同的發佈。

當 **strmqbrk** 建立主題 Soccer/Result 時，它會定義為主題 Sport 的子項，含有主題字串 Sport。Soccer/Results 的訂閱會實現為 Sport/Soccer/Results 的訂閱，因此 Sport 串流的發佈會對映至主題空間中的不同位置，而發佈會對映至不同的串流，例如 Business。

在某些情況下，**strmqbrk** 執行的自動移轉並不是答案，您需要手動新增串流。[新增串流](#)主題中說明了新增串流的作業。基於三個原因，您可能需要手動新增串流。

1. 您繼續在第 6 版佇列管理程式上維護發佈/訂閱應用程式，這會與在後續佇列管理程式上執行的新撰寫發佈/訂閱應用程式交互作業。
2. 您可以繼續開發在更新版本佇列管理程式上執行的已排入佇列的發佈/訂閱應用程式，而不是將應用程式移轉至整合發佈/訂閱 MQI 介面。
3. 串流至主題的預設對映會導致主題空間中的「衝突」，且串流上的發佈與來自其他位置的發佈具有相同的主題字串。

## 權限

依預設，在主題樹狀結構的根位置有多個主題物件：SYSTEM.BASE.TOPIC，SYSTEM.BROKER.DEFAULT.STREAM 和 SYSTEM.BROKER.DEFAULT.SUBPOINT。權限（例如，用於發佈或訂閱）由 SYSTEM.BASE.TOPIC 上的權限決定；SYSTEM.BROKER.DEFAULT.STREAM 或 SYSTEM.BROKER.DEFAULT.SUBPOINT 上的任何權限都會被忽略。如果使用非空主題字串刪除並重建 SYSTEM.BROKER.DEFAULT.STREAM 或 SYSTEM.BROKER.DEFAULT.SUBPOINT 之一，則會以與一般主題物件相同的方式使用對那些物件定義的權限。

## 串流與主題之間的對映

從 Version 7.0 開始，透過建立佇列並為其提供與串流相同的名稱，來模擬已排入佇列的發佈/訂閱串流。有時佇列稱為串流佇列，因為這就是佇列發佈/訂閱應用程式的顯示方式。將佇列新增至稱為 SYSTEM.QPUBSUB.QUEUE.NAMELIST 的特殊名稱清單，即可向發佈/訂閱引擎識別該佇列。您可以視需要新增任意數目的串流，方法是將其他特殊佇列新增至名稱清單。最後，您需要新增主題，其名稱與串流相同，且主題字串與串流名稱相同，因此您可以發佈及訂閱主題。

不過，在異常情況下，您可以為對應於串流的主題提供您在定義主題時選擇的任何主題字串。主題字串的目的在于提供主題空間中的唯一名稱。通常，串流名稱會完美地達到該目的。有時，串流名稱與現有主題名稱衝突。若要解決此問題，可以為與串流相關聯的主題選擇另一個主題字串。選擇任何主題字串，確定它是唯一的。

主題定義中定義的主題字串會以一般方式加上發佈者及訂閱者使用 MQOPEN 或 MQSUB MQI 呼叫所提供的主題字串的字首。使用主題物件來參照主題的應用程式不受選擇字首主題字串的影響-因此，您可以選擇任何主題字串，讓發佈資訊在主題空間中保持唯一。

將不同的串流重新對映至不同的主題，取決於主題字串所使用的字首是唯一的，以完全將一組主題與另一組主題分開。您必須定義嚴格遵循的通用主題命名慣例，對映才能運作。在 Version 7.0 中，如果主題字串衝突，您可以使用串流來區隔主題空間。從 Version 7.0 開始，您可以使用字首機制，將主題字串重新對映至主題空間中的另一個位置。

註: 當您刪除串流時, 請先刪除串流上的所有訂閱。如果任何訂閱源自分配管理系統階層中的其他分配管理系統, 則此動作最重要。

## 範例

在第 40 頁的圖 18 中, 主題 'Sport' 具有主題字串 'xyz', 導致源自串流 'Sport' 的發佈資訊在第 7 版佇列管理程式主題空間中以字串 'xyz' 作為字首。在第 7 版中發佈或訂閱主題字串的主題 'Sport' 字首 'xyz'。如果發佈流向第 6 版訂閱者, 則會從發佈中移除字首 'xyz', 並將其放置在 'Sport' 串流中。相反地, 當發佈資訊從第 6 版流向第 7 版時, 從 'Sport' 串流到 'Sport' 主題, 會將字首 'xyz' 新增至主題字串。

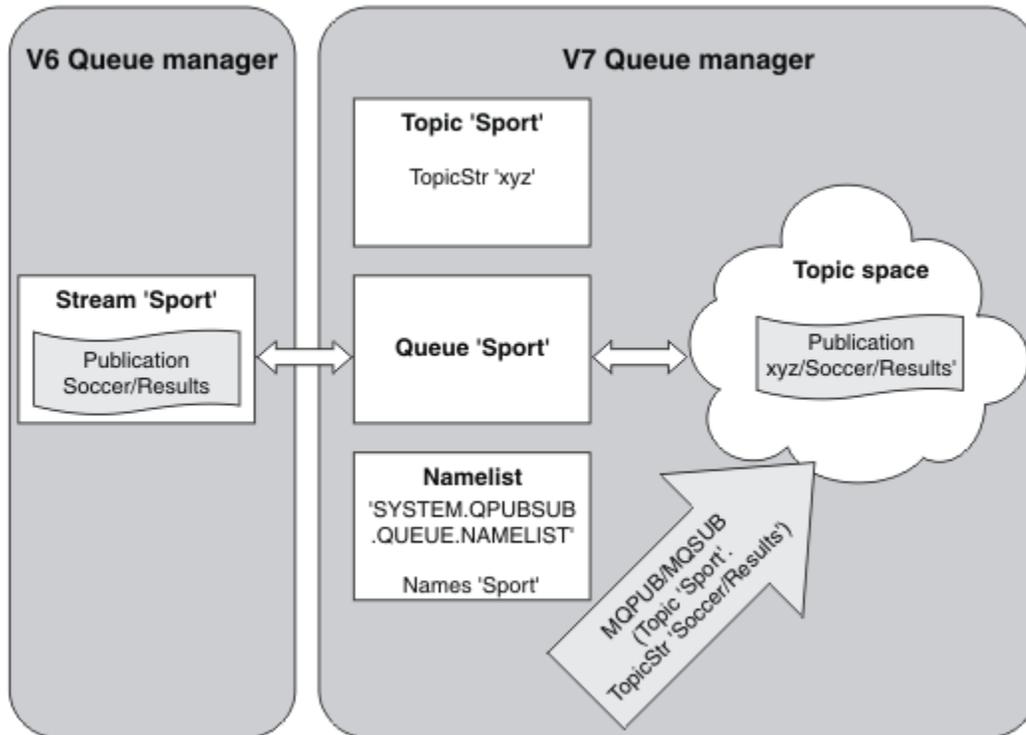


圖 18: 與第 7 版主題共存的第 6 版串流

## 訂閱點和主題

訂閱點, 用來從 WebSphere MQ Event Broker 及 Message Broker 中的一組特定發佈節點要求發佈。主題和主題物件會模擬具名訂閱點。

WebSphere MQ Event Broker V6.0 至 WebSphere MQ V7.0.1 移轉程序 **migmbbrk**, 會將具名訂閱點轉換成主題和主題物件。如果訂閱點具有保留的發佈資訊或已登錄的訂閱者, 則會自動移轉訂閱點。**migmbbrk** 會從具名訂閱點建立主題物件。訂閱點的名稱會變成主題物件的名稱, 以及主題字串本身。主題物件會新增至 `SYSTEM.QPUBSUB.SUBPOINT.NAMELIST`。

如果具有相同名稱的主題物件存在, **migmbbrk** 會執行兩件事之一。

1. 如果主題物件具有不同的主題字串, 或者訂閱點名稱長於物件名稱, 則 **migmbbrk** 會以產生的名稱建立主題物件。
2. 如果主題物件具有相同的主題字串, **migmbbrk** 會將現有物件新增至名單。

若要手動新增訂閱點, 請參閱 [新增訂閱點](#)。

## WebSphere MQ Event Broker 中的訂閱點

發佈節點用於 WebSphere MQ 事件及訊息分配管理系統訊息流程中，以過濾訊息並將訊息傳輸至訂閱者。發佈者通常不會在發佈節點上設定訂閱點。訂閱者會登錄對一組特定主題感興趣的項目，且通常也不會指定訂閱點。

訂閱點是一種選取哪些發佈節點將訊息轉遞至訂閱的方法。訂閱者使用訂閱點名稱來限定他們對一組主題的興趣。

將名稱指派給發佈節點的 **Subscription point** 內容，以設定其訂閱點名稱。

訂閱點內容會控制是否將主題的發佈轉遞給相同主題的訂閱者。來自具有指定訂閱點的發佈節點的發佈只會轉遞至相同訂閱點的訂閱者。來自沒有指定訂閱點 (預設值) 的發佈節點的發佈，只會轉遞至未指定訂閱點的訂閱者。

具有具名訂閱點的節點會以 MQRFH2 格式傳送 Publish 指令訊息，並設定 **SubPoint** 內容。具名訂閱點的訂閱必須在 MQRFH2 Register subscriber 指令訊息中設定 **SubPoint** 內容。

## WebSphere MQ 中的訂閱點

WebSphere MQ 會將訂閱對映至 WebSphere MQ 主題樹狀結構內的不同主題空間。指令訊息中沒有訂閱點的主題會對映至 WebSphere MQ 主題樹狀結構的根目錄，並繼承 SYSTEM.BASE.TOPIC 的內容。

具有訂閱點的指令訊息正在使用 SYSTEM.QPUBSUB.SUBPOINT.NAMELIST 中的主題物件清單進行處理。指令訊息中的訂閱點名稱會與清單中每一個主題物件的主題字串進行比對。如果找到相符項，則會在主題字串前面附加訂閱點名稱作為主題節點。主題會從 SYSTEM.QPUBSUB.SUBPOINT.NAMELIST 中找到的相關聯主題物件繼承其內容。

使用訂閱點的效果是為每一個訂閱點建立個別主題空間。主題空間根在與訂閱點同名的主題中。每一個主題空間中的主題會從與訂閱點同名的主題物件繼承其內容。

未在相符主題物件中設定的任何內容都會以正常方式從 SYSTEM.BASE.TOPIC 繼承。

現有排入佇列的發佈/訂閱應用程式 (使用 MQRFH2 訊息標頭) 可透過在 Publish 或 Register subscriber 指令訊息中設定 **SubPoint** 內容來繼續運作。訂閱點會與指令訊息中的主題字串結合，而產生的主題會像任何其他主題一樣處理。

新的 WebSphere MQ V7 應用程式不受訂閱點影響。如果它使用繼承自其中一個相符主題物件的主題，則它會使用相符訂閱點與已排入佇列的應用程式交互作業。

### 範例

群體中現有的 WebSphere MQ Event Broker 發佈/訂閱應用程式會使用訂閱點，以不同貨幣來發佈股價。IBM 股票的現貨價格使用訂閱點 USD 和主題 NYSE/IBM/SPOT 發佈。Sterling 價格是使用相同的主題和訂閱點 GBP 來發佈。

WebSphere MQ 上的移轉程序會建立兩個主題物件 GBP 和 USD，以及對應的主題字串 'GBP' 和 'USD'。

主題 NYSE/IBM/SPOT 的現有發佈者已移轉至在 WebSphere MQ 上執行，使用訂閱點 USD 在主題上建立發佈 USD/NYSE/IBM/SPOT。與 NYSE/IBM/SPOT 的現有訂閱者類似，使用訂閱點 USD 建立 USD/NYSE/IBM/SPOT 的訂閱。

透過呼叫 MQSUB，在第 7 版發佈/訂閱程式中訂閱美元現貨價格。使用 USD topic 物件和主題字串 'NYSE/IBM/SPOT' 來建立訂閱，如 'C' 程式碼片段中所示。

```
strncpy(sd.ObjectName, "USD", MQ_TOPIC_NAME_LENGTH);
sd.ObjectString.VSPtr = "NYSE/IBM/SPOT";
sd.ObjectString.VSLength = MQVS_NULL_TERMINATED;
MQSUB(Hconn, &sd, &Hobj, &Hsub, &CompCode, &Reason);
```

請考量群體中的 WebSphere MQ Event Broker 應用程式是否一律使用訂閱點 USD 和 GBP。如果他們這麼做，請只建立 USD 和 GBP 主題物件一次，作為叢集主題主機上的叢集主題。您不需要執行移轉程序的步驟 [../com.ibm.mq.mig.doc/q007670\\_dita#q007670\\_/clusterstep](#)，即可將叢集中每個佇列管理程式上的 SYSTEM.BASE.TOPIC 變更為叢集主題。請改為執行下列步驟：

1. 設定叢集主題主機上 USD 及 GBP 主題物件的 CLUSTER 屬性。

2. 刪除叢集中其他佇列管理程式上 USD 及 GBP 主題物件的所有副本。
3. 確保在叢集中每個佇列管理程式的 SYSTEM.QPUBSUB.SUBPOINT.NAMELIST 中定義 USD 和 GBP。

## 分散式發佈/訂閱

本節包含如何在佇列管理程式之間執行發佈/訂閱傳訊的相關資訊，以及可用來連接佇列管理程式、叢集和階層的兩種不同的佇列管理程式拓撲。

佇列管理程式可以與 WebSphere MQ 發佈/訂閱系統中的其他佇列管理程式通訊，讓訂閱者可以訂閱一個佇列管理程式，並接收最初發佈至另一個佇列管理程式的訊息。如第 42 頁的圖 19 中所說明。

第 42 頁的圖 19 顯示具有兩個佇列管理程式的發佈/訂閱系統。

- 「發佈者 4」使用「佇列管理程式 2」來發佈天氣預測資訊(使用「天氣」主題)，以及主要道路上交通狀況的相關資訊(使用「交通」主題)。
- 訂閱者 4 也會使用此佇列管理程式，並使用主題「資料流量」來訂閱資料流量狀況的相關資訊。
- 訂閱者 3 也會訂閱天氣狀況的相關資訊，即使它使用與發佈者不同的佇列管理程式也一樣。這可能是因為佇列管理程式彼此鏈結。

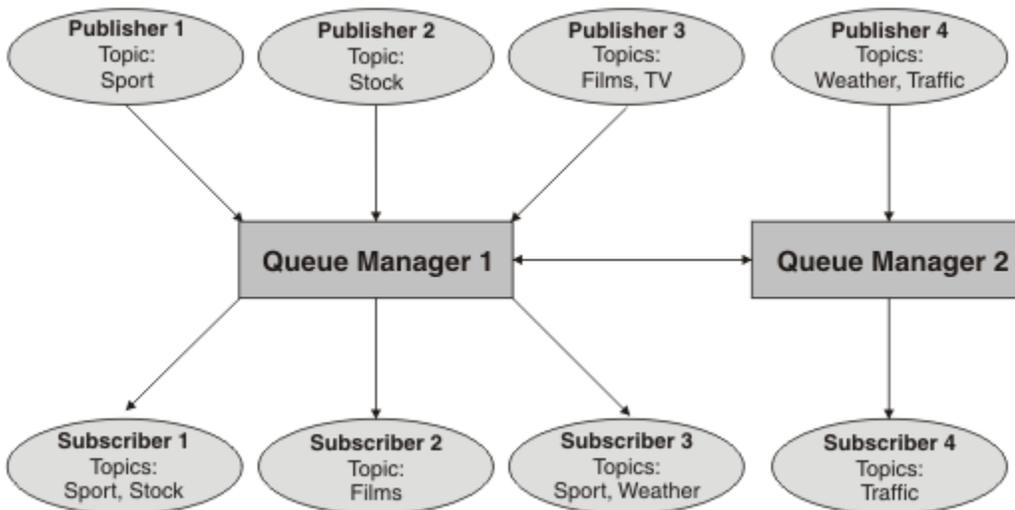


圖 19: 具有兩個佇列管理程式的發佈/訂閱範例

## 分散式發佈/訂閱如何運作?

WebSphere MQ 發佈/訂閱會使用 Proxy 訂閱，以確保訂閱者可以接收已發佈至遠端佇列管理程式的訊息。

分散式發佈/訂閱使用與分散式佇列相同的元件來連接佇列管理程式的網路，從而連接至那些佇列管理程式的應用程式。若要進一步瞭解佇列管理程式之間的傳訊，以及在佇列管理程式之間建立連線所涉及的元件，請參閱 交互通訊 文件。

在分散式發佈/訂閱系統中，訂閱者不需要執行任何超出標準訂閱作業的動作。在佇列管理程式上進行訂閱時，佇列管理程式會管理將訂閱延伸到所連接佇列管理程式的處理程序。Proxy 訂閱會流向網路中的所有佇列管理程式。建立它們是為了確保發佈會遞送回建立原始訂閱的佇列管理程式；請參閱 第 43 頁的圖 20。

只有在該遠端佇列管理程式上存在該主題的訂閱時，發佈才會延伸到遠端佇列管理程式。

佇列管理程式會合併在其上建立的所有訂閱，不論是從本端應用程式或遠端佇列管理程式。除非存在訂閱，否則它會針對訂閱及其鄰接項的主題建立 Proxy 訂閱；請參閱 第 43 頁的圖 21。

當應用程式發佈資訊時，接收端佇列管理程式會將它轉遞至在遠端佇列管理程式上具有有效訂閱的任何應用程式。它可能會透過一或多個中繼佇列管理程式來轉遞它；請參閱 第 44 頁的圖 22。

訂閱者 1 在亞洲佇列管理程式 (1) 上登錄特定主題的訂閱。此主題的訂閱會轉遞至網路中的所有其他佇列管理程式 (2,3, 4)。

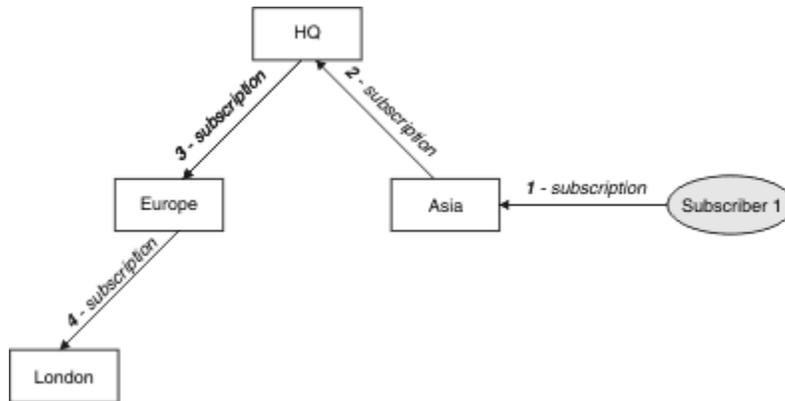


圖 20: 透過佇列管理程式網路傳播訂閱

訂閱者 2 會在 HQ 佇列管理程式 (5) 上向第 43 頁的圖 20 中的相同主題登錄訂閱。本主題的訂閱會轉遞至亞洲佇列管理程式，以便它知道網路上的其他位置有訂閱 (6)。訂閱不會轉遞至歐洲佇列管理程式，因為已登錄此主題的訂閱；請參閱第 43 頁的圖 20 中的步驟 3。

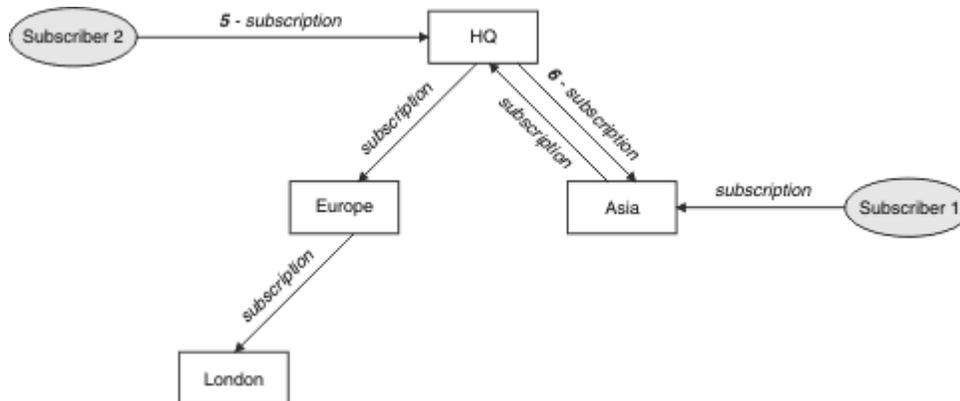


圖 21: 多個訂閱

發佈者根據與第 43 頁的圖 21 中相同的主题，將發佈資訊傳送至歐洲佇列管理程式 (7)。這個主题的訂閱是從 HQ 到歐洲，因此發佈會轉遞到 HQ 佇列管理程式 (8)。不過，從「倫敦」到「歐洲」(僅從「歐洲」到「倫敦」) 沒有訂閱，因此不會將發佈轉遞至「倫敦」佇列管理程式。HQ 佇列管理程式會將發佈直接傳送至訂閱者 2 及亞洲佇列管理程式 (9)。發佈會從亞洲 (10) 轉遞給訂閱者 1。

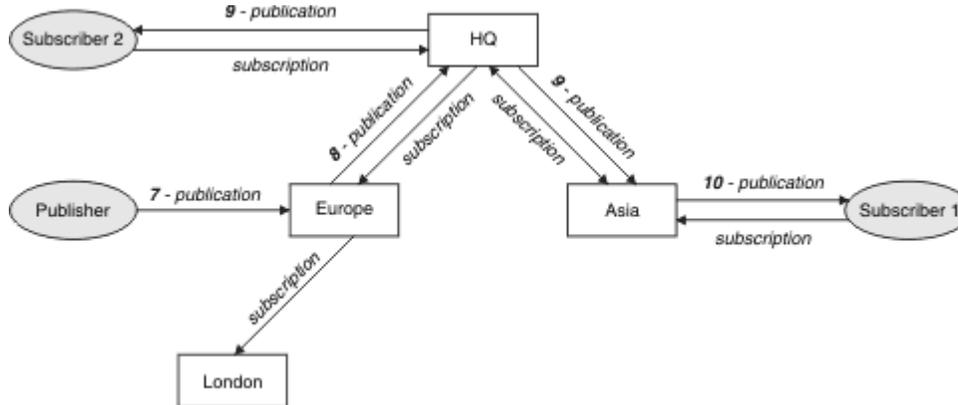


圖 22: 透過佇列管理程式網路傳播發佈資訊

當佇列管理程式將任何發佈或訂閱傳送至另一個佇列管理程式時，它會在訊息中設定自己的使用者 ID。如果您使用發佈/訂閱階層，且送入通道設定為在訊息中放置具有使用者 ID 權限的訊息，則必須授權傳送端佇列管理程式的使用者 ID; 請參閱第 89 頁的『將預設使用者 ID 與佇列管理程式階層搭配使用』。如果您使用發佈/訂閱叢集，則叢集會處理授權。

由於發佈/訂閱佇列管理程式的交互連接本質，Proxy 訂閱需要一些時間才能在網路中的所有節點周圍傳播。遠端發佈不一定會立即開始訂閱。您可以使用值為 FORCE 的 **Topic** 屬性 PROXYSUB，來消除訂閱延遲，如第 45 頁的『遞送機制的相關資訊』中所述。

當 Proxy 訂閱已放置在每一個直接連接的佇列管理程式的適當傳輸佇列上時，訂閱作業即會完成。訂閱作業不會等待 Proxy 訂閱延伸至拓撲的其餘部分。

Proxy 訂閱與建立它們的佇列管理程式名稱相關聯。如果階層中的佇列管理程式具有相同的名稱，則可能會導致發佈無法呼叫到它們。為了避免這個問題，如同點對點傳訊一樣，請提供佇列管理程式唯一名稱，特別是在 WebSphere MQ 網路中直接或間接連接的佇列管理程式。

在分散式發佈/訂閱網路內，可以使用發佈和訂閱範圍來控制發佈和訂閱的流程，並在適當時加以限制。

#### Proxy 訂閱聚集和發佈聚集

聚集分散式發佈/訂閱發佈及 Proxy 訂閱，以將發佈/訂閱佇列管理程式之間傳遞的訊息數量減至最少。

Proxy 訂閱是指一個佇列管理程式針對在另一個佇列管理程式中發佈的主题所進行的訂閱。您未明確建立 Proxy 訂閱，佇列管理程式會代表您建立 Proxy 訂閱; 請參閱第 42 頁的『分散式發佈/訂閱如何運作?』。

您可以將佇列管理程式連接至發佈/訂閱階層，或連接至發佈/訂閱叢集。Proxy 訂閱在連接的佇列管理程式之間流動。Proxy 訂閱會讓連接至某個佇列管理程式的發佈者所建立的主题發佈，由連接至其他佇列管理程式的該主题訂閱者接收; 請參閱第 52 頁的『發佈/訂閱拓撲』。

Proxy 訂閱會針對某個訂閱所訂閱的每個個別主题字串，在佇列管理程式之間流動。

您可以使用 **Topic** 屬性 PUBSCOPE 和 SUBSCOPE，來限制所連接佇列管理程式之間的 Proxy 訂閱和發佈流程。您也可以將 **Topic** 屬性 萬用字元 設為 BLOCK，以限制包含萬用字元的 Proxy 訂閱流程; 請參閱第 46 頁的『萬用字元規則』。

在建立訂閱時，會以非同步方式在佇列管理程式之間傳送 Proxy 訂閱。您可以將主题上的 **Topic** 屬性 PROXYSUB 設為 FORCE，或將訂閱的主题母項，以減少等待 Proxy 訂閱延伸到所有連接的佇列管理程式的延遲; 請參閱第 45 頁的『遞送機制的相關資訊』。

## Proxy 訂閱聚集

使用重複排除系統來聚集 Proxy 訂閱。對於特定的已解析主題字串，會在第一個本端訂閱或接收的 Proxy 訂閱上傳送 Proxy 訂閱。相同主題字串的後續訂閱會使用這個現有的 Proxy 訂閱。

在取消前次本端訂閱或收到的 Proxy 訂閱之後，會取消 Proxy 訂閱。

在發佈/訂閱拓撲中，如果個別主題字串有數千個訂閱，或這些訂閱的存在可能正在快速變更，則必須考量 Proxy 訂閱延伸的額外負擔。個別 Proxy 訂閱可以透過使用設為 FORCE 的主題屬性 **PROXYSUB** 進行合併。如需遞送機制和叢集主題效能的詳細資料，請參閱 [第 45 頁的『遞送機制的相關資訊』](#)。

## 發佈聚集

當佇列管理程式上有多個相同主題字串的訂閱時，在發佈/訂閱拓撲中，只會從其他佇列管理程式傳送每一個符合該主題字串之發佈的單一副本。訊息到達時，本端佇列管理程式會將訊息副本遞送至每一個相符的訂閱。

當 Proxy 訂閱包含萬用字元時，多個 Proxy 訂閱可以符合單一發佈的主題字串。如果在佇列管理程式上發佈的訊息符合單一連接佇列管理程式所建立的兩個以上 Proxy 訂閱，則只會將發佈的一個副本轉遞至遠端佇列管理程式，以滿足多個 Proxy 訂閱。

### 遞送機制的相關資訊

到處發佈是個別 Proxy 訂閱轉遞的替代遞送機制。個別 Proxy 訂閱轉遞表示只有在主題字串上具有相符訂閱的發佈才會傳送至遠端傳訊伺服器。將發佈至傳訊伺服器的所有發佈，轉遞至分散式發佈/訂閱網路中的所有其他傳訊伺服器，即可隨處發佈或播送。然後，接收端傳訊伺服器會遞送符合本端訂閱的那些發佈。

每個機制都有其優點，但也有其侷限性。

### 個別 Proxy 訂閱轉遞

此機制會導致佇列管理程式間發佈資料流量的數量最少，因為只會傳送那些符合佇列管理程式上訂閱的發佈。

不過：

- 所訂閱的每一個個別主題字串都會產生 Proxy 訂閱，該訂閱會傳送至發佈/訂閱拓撲中的所有其他佇列管理程式。如果要建立或刪除數以千計的訂閱 (例如，在重新啟動佇列管理程式之後所有不可延續的訂閱)，或如果訂閱集快速變更，且每一個訂閱都是不同的主題字串，則此傳訊額外負擔可能很重要。
- Proxy 訂閱會使用非同步傳訊來傳送至其他佇列管理程式，因此在建立訂閱與其他佇列管理程式建立、遞送及處理 Proxy 訂閱之間會有延遲。在該間隔中發佈在那些佇列管理程式的訊息不會遞送至遠端訂閱。

### 隨處發佈

使用此機制：

- 系統上沒有每個主題字串 Proxy 訂閱額外負擔，這表示快速建立、刪除或變更訂閱不會導致網路負載及處理程序增加。
- 建立訂閱與將發佈傳送至佇列管理程式之間沒有延遲，因為它們一律會傳送至所有佇列管理程式。因此，沒有任何視窗未將發佈資訊遞送至新建立的遠端訂閱。

不過：

- 所有發佈都會傳送至發佈/訂閱拓撲中的所有佇列管理程式，可能會導致網路資料流量過多，其中發佈在每一個佇列管理程式上沒有相符的訂閱。

當您預期從叢集或階層中相當大部分的佇列管理程式訂閱發佈，或 Proxy 訂閱額外需要因為訂閱變更頻率而太大時，您可能想要使用「隨處發佈」機制。在這些實例中，當發佈傳送至所有佇列管理程式，而不是傳送至具有相符訂閱的佇列管理程式時，此工作方法可能比在其他實例中遇到傳訊資料流量增加的情況更有效。

透過將高階主題物件的 **PROXYSUB** 屬性設為 FORCE，可以在 IBM WebSphere MQ 分散式發佈/訂閱拓撲中啟用隨處發佈機制。

如需停用個別 Proxy 訂閱的詳細資料，請參閱 [第 64 頁的『停用個別 Proxy 訂閱』](#)。

當這個強制 Proxy 訂閱延伸到整個拓撲時，任何新的訂閱都會立即從其他連接的佇列管理程式接收任何發佈，而不會有延遲。

配置這類系統時必須小心。在 **PROXYSUB** 設為 **FORCE** 的主題下，沒有任何主題物件必須與 **PROXYSUB** 設為 **FORCE** 的節點位於不同的叢集或階層串流中。同樣地，較低的主題物件不得將其 **WILDCARD** 屬性設為 **BLOCK**。在這兩種情況下，這可能導致已發佈的訊息無法正確從一個佇列管理程式流向另一個佇列管理程式。

即使 **PROXYSUB** 設為 **FORCE**，所訂閱的每一個個別主題字串的 Proxy 訂閱也會繼續延伸。如果訂閱的數目及頻率夠高，而導致系統的額外負荷，則可以針對佇列管理程式上的所有主題停用它們。如需停用個別 Proxy 訂閱的詳細資料，請參閱 [第 64 頁的『停用個別 Proxy 訂閱』](#)。

## 多重播送及訂閱延遲

訂閱延遲和 **PROXYSUB (FORCE)** 選項可用來維護 **PROXY** 訂閱。

例如，在切斷所有訂閱者的連線之後，從 **QM\_B** 到 **QM\_A** 的 Proxy 訂閱可能發生未建立的問題。如果您需要繼續多重播送資料流量，即使佇列管理程式的單點播送連線終止，也可能不需要此狀況。WebSphere MQ 的多重播送會在新訂閱者連接時，透過為每一個 Proxy 訂閱增加幾分鐘延遲來維護 Proxy 訂閱，以便它們不會在最後一個訂閱者終止時取消。

您也可以在主題上使用 **PROXYSUB (FORCE)** 選項，以確保未完成的 Proxy 訂閱一律未完成。在訂閱處於作用中的大部分時間內，您必須確保至少一個訂閱者需要在佇列中流動的訊息。如果設定 **PROXYSUB (FORCE)**，Proxy 訂閱可能會在第一個本端訂閱或收到的 Proxy 訂閱之前傳送，而且即使在最後一個本端訂閱或收到的 Proxy 訂閱取消之後也不會取消。

如果仍未建立訂閱，則可以使用對等式通訊來確保訊息傳送繼續；如需相關資訊，請參閱 [多重播送的高可用性](#)。

### 萬用字元規則

Proxy 訂閱中的萬用字元會轉換成使用主題萬用字元。

如果收到萬用字元訂閱，則可以是 WebSphere MQ 6.0 所使用的字元。它也可以是 WebSphere Message Broker 6.0 和 WebSphere MQ 7.0 所使用的主題。

- 字元萬用字元使用 **\*** 來代表任何字元，包括 **/**。
- 主題萬用字元使用 **#** 來代表 **/** 字元之間的主題空格部分。

在 WebSphere MQ 7.0 版中，所有 Proxy 訂閱都會轉換成使用主題萬用字元。如果找到萬用字元，則會將它取代為 **#** 字元，回到最接近的 **/**。例如，**/aaa/bbb/c\*d** 會轉換為 **/aaa/bbb/#**。轉換會導致遠端佇列管理程式傳送的發佈數比明確訂閱的發佈數略多。將發佈遞送給本端訂閱者時，本端佇列管理程式會過濾掉其他發佈。

## 使用 萬用字元 內容控制萬用字元用法

使用 **MQSC Topic** 萬用字元 內容或對等 **PCF Topic WildcardOperation** 內容來控制將發佈遞送至使用萬用字元主題字串名稱的訂閱者應用程式。WILDCARD 內容可以具有下列兩個可能值之一：

### WILDCARD

關於此主題的萬用字元訂閱的行為。

### PASSTHRU

對於比此主題物件的主題字串更不具體的萬用字元式主題所做的訂閱，將接收到對此主題以及比此主題更具體的主題字串所進行的發佈。

### BLOCK

對於比此主題物件的主題字串更不具體的萬用字元式主題所做的訂閱，不會接收到對此主題或比此主題更具體的主題字串所進行的發佈。

在定義訂閱時將使用此屬性的值。如果變更此屬性，則現有訂閱所涵蓋的主題集不會因為此修改而受到影響。如果在建立或刪除主題物件時拓撲發生變更，也適用此實務範例；將使用修改後的拓撲來建立與 WILDCARD 屬性修改後建立的訂閱相符的主題集。若要針對現有訂閱強制重新評估相符的主題集，則必須重新啟動佇列管理程式。

在範例 [第 36 頁的『範例: 建立 Sport 發佈/訂閱叢集』](#) 中，您可以遵循步驟來建立 [第 33 頁的圖 10](#) 中所示的主題樹狀結構。

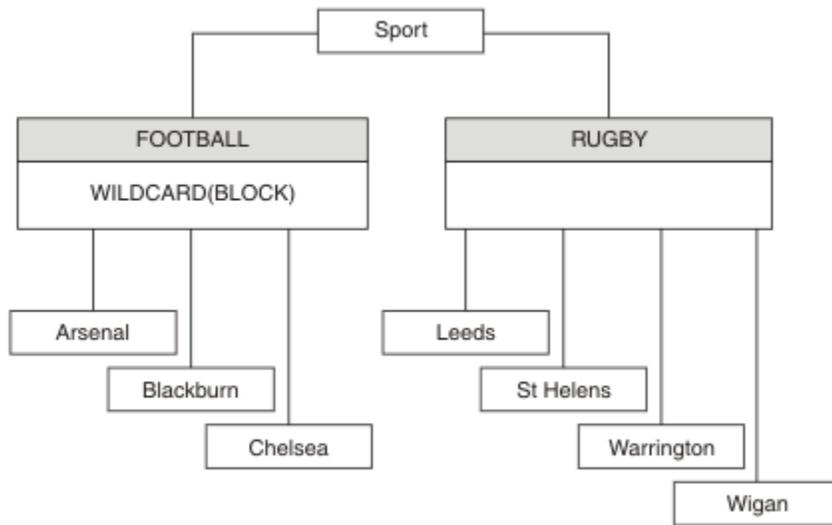


圖 23: 使用 萬用字元 內容 BLOCK 的主題樹狀結構

使用萬用字元主題字串 # 的訂閱者會接收 Sport 主題和 Sport/Rugby 子樹狀結構的所有發佈。訂閱者不會收到 Sport/Football 子樹狀結構的發佈，因為 Sport/Football 主題的 萬用字元 內容值是 BLOCK。

PASSTHRU 是預設值。您可以將 萬用字元 內容值 PASSTHRU 設為 Sport 樹狀結構中的節點。如果節點沒有 萬用字元 內容值 BLOCK，則設定 PASSTHRU 不會變更訂閱者對 Sports 樹狀結構中節點所觀察到的行為。

在範例中，建立訂閱以查看萬用字元設定如何影響遞送的發佈；請參閱 第 37 頁的圖 14。在 第 38 頁的圖 17 中執行發佈指令，以建立部分發佈。

pub QMA

圖 24: 發佈至 QMA

結果顯示在 第 33 頁的表 3 中。請注意如何設定 WILDCARD 內容值 BLOCK，防止使用萬用字元的訂閱接收對萬用字元範圍內主題的發佈。

訂閱	主題字串	收到的出版物	附註
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Sports/Football 上的 WILDCARD (BLOCK) 封鎖所有 Football 子樹狀結構的發佈
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) on Sports/Football 阻止在 Arsenal 上進行萬用字元訂閱
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Sports/Rugby 上的預設 WILDCARD 不會阻止 Leeds 上的萬用字元訂閱。

**註:**

假設訂閱具有萬用字元，其符合具有 WILDCARD 內容值 BLOCK 的主題物件。如果訂閱在相符萬用字元的右側也有主題字串，則訂閱永遠不會收到發佈。未封鎖的發佈資訊集是屬於已封鎖萬用字元母項之主題的發佈資訊。作為主題子項且具有 BLOCK 內容值的主題的發佈會被萬用字元封鎖。因此，如果訂閱主題字串包含萬用字元右側的主題，則永遠不會收到任何要符合的發佈。

將 WILDCARD 內容值設為 BLOCK 並不表示您無法使用包含萬用字元的主題字串來訂閱。這類訂閱是正常的。訂閱有一個明確主題符合主題，且主題物件具有 WILDCARD 內容值 BLOCK。對於作為主題母項或子項且具有 萬用字元 內容值 BLOCK 的主題，它會使用萬用字元。在 [第 33 頁的圖 10](#) 的範例中，Sports/Football/# 之類的訂閱可以接收發佈。

## 萬用字元和叢集主題

叢集主題定義會延伸到叢集中的每個佇列管理程式。訂閱叢集中某個佇列管理程式的叢集主題會導致佇列管理程式建立 Proxy 訂閱。在叢集中的所有其他佇列管理程式上建立 Proxy 訂閱。使用包含萬用字元且與叢集主題結合的主題字串的訂閱，可能會提供難以預測的行為。此行為在下列範例中說明。

在針對範例 [第 36 頁的『範例: 建立 Sport 發佈/訂閱叢集』](#) 所設定的叢集中，QMB 具有與 QMA 相同的訂閱集，但 QMB 在發佈者發佈至 QMA 之後未收到任何發佈，請參閱 [第 33 頁的圖 11](#)。雖然 Sports/Football 及 Sports/Rugby 主題是叢集主題，但 fullsubs.tst 中定義的訂閱不會參照叢集主題。沒有任何 Proxy 訂閱從 QMB 延伸到 QMA。如果沒有 Proxy 訂閱，則不會將任何對 QMA 的發佈轉遞至 QMB。

部分訂閱 (例如 Sports/#/Leeds) 似乎參照叢集主題，在此情況下為 Sports/Rugby。Sports/#/Leeds 訂閱實際上會解析為主題物件 SYSTEM.BASE.TOPIC。

用於解析訂閱 (例如 Sports/#/Leeds) 所參照的主題物件的規則如下。將主題字串截斷為第一個萬用字元。掃描左側的主題字串，以尋找具有相關聯管理主題物件的第一個主題。主題物件可以指定叢集名稱，或定義本端主題物件。在範例 Sports/#/Leeds 中，截斷之後的主題字串是 Sports，它沒有主題物件，因此 Sports/#/Leeds 繼承自 SYSTEM.BASE.TOPIC，它是本端主題物件。

若要查看訂閱叢集主題如何變更萬用字元傳播的運作方式，請執行批次 Script upsubs.bat。該 Script 會清除訂閱佇列，並在 fullsubs.tst 中新增叢集主題訂閱。再次執行 puba.bat 以建立一批出版品; 請參閱 [第 33 頁的圖 11](#)。

[第 34 頁的表 4](#) 顯示將兩個新訂閱新增至發佈者所在之相同佇列管理程式的結果。結果如預期，新訂閱各會收到一個發佈，而其他訂閱所收到的發佈數則維持不變。其他叢集佇列管理程式上發生非預期的結果; 請參閱 [第 35 頁的表 5](#)。

訂閱	主題字串	收到的出版物	附註
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Sports/Football 上的 WILDCARD (BLOCK) 封鎖所有 Football 子樹狀結構的發佈
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) on Sports/Football 阻止在 Arsenal 上進行萬用字元訂閱
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Sports/Rugby 上的預設 WILDCARD 不會阻止 Leeds 上的萬用字元訂閱。
FARSENAL	Sports/Football/ Arsenal	Sports/Football/ Arsenal	Arsenal 會接收發佈，因為訂閱沒有萬用字元。
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds 會在任何事件中收到發佈。

[第 35 頁的表 5](#) 顯示在 QMB 上新增兩個新訂閱，以及在 QMA 上發佈的結果。回想一下，如果沒有這兩個新訂閱，QMB 不會收到任何發佈。如預期，這兩個新的訂閱會接收發佈，因為 Sports/FootBall 和 Sports/Rugby 都是叢集主題。QMB 已將 Sports/Football/Arsenal 和 Sports/Rugby/Leeds 的 Proxy 訂閱轉遞至 QMA，然後將發佈傳送至 QMB。

非預期的結果是先前未收到任何發佈的兩個訂閱 Sports/# 及 Sports/#/Leeds 現在已收到發佈。原因是其他訂閱的 Sports/Football/Arsenal 及 Sports/Rugby/Leeds 轉遞至 QMB 的發佈現在可用於連接至 QMB 的任何訂閱者。因此，對本端主題 Sports/# 及 Sports/#/Leeds 的訂閱會收到 Sports/Rugby/Leeds 發佈。Sports/#/Arsenal 會繼續不接收發佈，因為「體育/足球」已將其 萬用字元 內容值設為 BLOCK。

表 8: 在 QMB 上收到的出版物

訂閱	主題字串	收到的出版物	附註
SPORTS	Sports/#	Sports/Rugby/Leeds	WILDCARD (BLOCK) 在 Sports/Football 上封鎖所有發佈至 Football 子樹狀結構
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) on Sports/Football 阻止在 Arsenal 上進行萬用字元訂閱
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Sports/Rugby 上的預設 WILDCARD 不會阻止 Leeds 上的萬用字元訂閱。
FARSENAL	Sports/Football/Arsenal	Sports/Football/Arsenal	Arsenal 會接收發佈，因為訂閱沒有萬用字元。
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds 會在任何事件中收到發佈。

在大部分應用程式中，一個訂閱不想要影響另一個訂閱的行為。WILDCARD 內容與值 BLOCK 的一個重要用途是讓相同主題字串 (包含萬用字元) 的訂閱統一運作。不論訂閱位於與發佈者相同的佇列管理程式上，或位於不同的佇列管理程式上，訂閱的結果都相同。

## 萬用字元和串流

WebSphere MQ 第 6 版串流由 WebSphere MQ 第 7 版對映至主題; 請參閱第 38 頁的『串流和主題』。在預設對映中 (在第 7 版中由 **strmqbrk** 執行)，串流 Sports 中的所有主題都會對映至主題 Sports。串流 Business 中的所有主題都將對映至主題 Business。

Sports 串流中 WebSphere MQ 第 6 版至 \* 的訂閱會接收 Sports 樹狀結構中的所有發佈，而 Business 樹狀結構中沒有發佈。第 7 版中的相同訂閱會接收 Sports 樹狀結構中的所有發佈，以及 Business 樹狀結構中的所有發佈。若要封鎖此行為，當串流移轉至第 7 版時，**strmqbrk** 會設定萬用字元內容。對於從串流移轉的每一個最上層主題，它會將它設為值 BLOCK。透過從稱為 Sports 和 Business 的第 6 版串流進行轉換，將 Sports 和 Business 的萬用字元內容設定為值 BLOCK。

對於寫入至發佈/訂閱 API 的新應用程式，其效果是 \* 的訂閱不會收到任何發佈。若要接收所有「體育」發佈資訊，您必須訂閱 Sports/\* 或 Sports/#，以及類似的 Business 發佈資訊。

當發佈/訂閱分配管理系統移轉至 WebSphere MQ 第 7 版時，現有已排入佇列的發佈/訂閱應用程式的行為不會變更。**Publish**、**Register Publisher** 或 **Subscriber** 指令中的 **StreamName** 內容對映至已移轉串流的主題名稱。

## 萬用字元和訂閱點

WebSphere Message Broker 訂閱點由 WebSphere MQ 第 7 版對映至主題; 請參閱第 40 頁的『訂閱點和主題』。在預設對映中 (由第 7 版中的 **migmqbrk** 執行)，訂閱點 Sports 中的所有主題都會對映至主題 Sports。訂閱點 Business 中的所有主題都會對映至主題 Business。

Sports 訂閱點中 WebSphere Message Broker 第 6 版至 \* 的訂閱會接收 Sports 樹狀結構中的所有發佈，而 Business 樹狀結構中沒有發佈。第 7 版中的相同訂閱會接收 Sports 樹狀結構中的所有發佈，以及 Business 樹狀結構中的所有發佈。若要封鎖此行為，當訂閱點移轉至第 7 版時，**migmqbrk** 會設定萬用字元內容。它會針對從訂閱點移轉的每一個最上層主題，將它設為值 BLOCK。透過從稱為 Sports 及 Business 的 WebSphere Message Broker 訂閱點進行轉換，將 Sports 及 Business 的萬用字元內容設為值 BLOCK。

對於寫入發佈/訂閱 API 的新應用程式，移轉的效果是 \* 的訂閱不會收到任何發佈。若要接收所有「體育」發佈資訊，您必須訂閱 Sports/\* 或 Sports/#，以及類似的 Business 發佈資訊。

當發佈/訂閱分配管理系統移轉至 WebSphere MQ 第 7 版時，現有已排入佇列的發佈/訂閱應用程式的行為不會變更。**Publish**、**Register Publisher** 或 **Subscriber** 指令中的 **SubPoint** 內容對映至已移轉訂閱的主題名稱。

## 範例: 建立 Sport 發佈/訂閱叢集

接下來的步驟會建立叢集 CL1，其中包含四個佇列管理程式: 兩個完整儲存庫、CL1A 和 CL1B，以及兩個局部儲存庫 QMA 和 QMB。完整儲存庫用來只保留叢集定義。QMA 指定為叢集主題主機。可延續訂閱同時定義在 QMA 和 QMB 上。

註: 此範例是針對 Windows 撰寫的。您必須重新編碼 建立 qmgrs.bat 及 建立 pub.bat，以在其他平台上配置及測試範例。

1. 建立 Script 檔。
  - a. 建立 topics.tst
  - b. 建立 wildsubs.tst
  - c. 建立 fullsubs.tst
  - d. 建立 qmgrs.bat
  - e. 建立 pub.bat
2. 執行 Create qmgrs.bat 以建立配置。

```
qmgrs
```

在 [第 33 頁的圖 10](#) 中建立主題。圖 5 中的 Script 會建立叢集主題 Sports/Football 和 Sports/Rugby。

註: REPLACE 選項不會取代主題的 TOPICSTR 內容。TOPICSTR 是在範例中用來測試不同主題樹狀結構的有用內容。若要變更主題，請先刪除主題。

```
DELETE TOPIC ('Sports')
DELETE TOPIC ('Football')
DELETE TOPIC ('Arsenal')
DELETE TOPIC ('Blackburn')
DELETE TOPIC ('Chelsea')
DELETE TOPIC ('Rugby')
DELETE TOPIC ('Leeds')
DELETE TOPIC ('Wigan')
DELETE TOPIC ('Warrington')
DELETE TOPIC ('St. Helens')

DEFINE TOPIC ('Sports') TOPICSTR('Sports')
DEFINE TOPIC ('Football') TOPICSTR('Sports/Football') CLUSTER(CL1) WILDCARD(BLOCK)
DEFINE TOPIC ('Arsenal') TOPICSTR('Sports/Football/Arsenal')
DEFINE TOPIC ('Blackburn') TOPICSTR('Sports/Football/Blackburn')
DEFINE TOPIC ('Chelsea') TOPICSTR('Sports/Football/Chelsea')
DEFINE TOPIC ('Rugby') TOPICSTR('Sports/Rugby') CLUSTER(CL1)
DEFINE TOPIC ('Leeds') TOPICSTR('Sports/Rugby/Leeds')
DEFINE TOPIC ('Wigan') TOPICSTR('Sports/Rugby/Wigan')
DEFINE TOPIC ('Warrington') TOPICSTR('Sports/Rugby/Warrington')
DEFINE TOPIC ('St. Helens') TOPICSTR('Sports/Rugby/St. Helens')
```

圖 25: 刪除並建立主題: *topics.tst*

註: 刪除主題，因為 REPLACE 不會取代主題字串。

使用萬用字元建立訂閱。萬用字元對應主題與 [第 33 頁的圖 10](#) 中的主題物件。為每一個訂閱建立佇列。執行或重新執行 Script 時，會清除佇列並刪除訂閱。

註: REPLACE 選項不會取代訂閱的 TOPICOBJ 或 TOPICSTR 內容。TOPICOBJ 或 TOPICSTR 是在測試不同訂閱的範例中有效改變的內容。若要變更它們，請先刪除訂閱。

```

DEFINE QLOCAL(QSPORTS) REPLACE
DEFINE QLOCAL(QSARSENAL) REPLACE
DEFINE QLOCAL(QSLEEDS) REPLACE
CLEAR QLOCAL(QSPORTS)
CLEAR QLOCAL(QSARSENAL)
CLEAR QLOCAL(QSLEEDS)

DELETE SUB (SPORTS)
DELETE SUB (SARSENAL)
DELETE SUB (SLEEDS)
DEFINE SUB (SPORTS) TOPICSTR('Sports/#') DEST(QSPORTS)
DEFINE SUB (SARSENAL) TOPICSTR('Sports+/Arsenal') DEST(QSARSENAL)
DEFINE SUB (SLEEDS) TOPICSTR('Sports+/Leeds') DEST(QSLEEDS)

```

圖 26: 建立萬用字元訂閱: *wildsubs.tst*

建立參照叢集主題物件的訂閱。

**註:**

定界字元 / 會自動插入 TOPICOBJ 所參照的主題字串與 TOPICSTR 所定義的主題字串之間。

DEFINE SUB(FARSENAL) TOPICSTR('Sports/Football/Arsenal') DEST(QFARSENAL) 定義會建立相同的訂閱。TOPICOBJ 是用來作為參照您已定義之主題字串的快速方式。建立的訂閱不再參照主題物件。

```

DEFINE QLOCAL(QFARSENAL) REPLACE
DEFINE QLOCAL(QRLEEDS) REPLACE
CLEAR QLOCAL(QFARSENAL)
CLEAR QLOCAL(QRLEEDS)

DELETE SUB (FARSENAL)
DELETE SUB (RLEEDS)
DEFINE SUB (FARSENAL) TOPICOBJ('Football') TOPICSTR('Arsenal') DEST(QFARSENAL)
DEFINE SUB (RLEEDS) TOPICOBJ('Rugby') TOPICSTR('Leeds') DEST(QRLEEDS)

```

圖 27: 刪除並建立訂閱: *fullsubs.tst*

建立具有兩個儲存庫的叢集。建立兩個局部儲存庫以進行發佈和訂閱。請重新執行 Script 以刪除所有項目，然後重新啟動。Script 也會建立主題階層及起始萬用字元訂閱。

**註:**

在其他平台上，撰寫類似的 Script，或輸入所有指令。使用 Script 可讓您快速刪除所有項目，並以相同的配置重新開始。

```

@echo off
set port.CL1B=1421
set port.CL1A=1420
for %%A in (CL1A CL1B QMA QMB) do call :createQM %%A
call :configureQM CL1A CL1B %port.CL1B% full
call :configureQM CL1B CL1A %port.CL1A% full
for %%A in (QMA QMB) do call :configureQM %%A CL1A %port.CL1A% partial
for %%A in (topics.tst wildsubs.tst) do runmqsc QMA < %%A
for %%A in (wildsubs.tst) do runmqsc QMB < %%A
goto:eof

:createQM
echo Configure Queue manager %1
endmqm -p %1
for %%B in (dlt crt str) do %%Bmqm %1
goto:eof

:configureQM
if %1==CL1A set p=1420
if %1==CL1B set p=1421
if %1==QMA set p=1422
if %1==QMB set p=1423
echo configure %1 on port %p% connected to repository %2 on port %3 as %4 repository
echo DEFINE LISTENER(LST%1) TRPTYPE(TCP) PORT(%p%) CONTROL(QMGR) REPLACE | runmqsc %1
echo START LISTENER(LST%1) | runmqsc %1
if full==%4 echo ALTER QMGR REPOS(CL1) DEADQ(SYSTEM.DEAD.LETTER.QUEUE) | runmqsc %1
echo DEFINE CHANNEL(TO.%2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('LOCALHOST(%3)') CLUSTER(CL1)
REPLACE | runmqsc %1
echo DEFINE CHANNEL(TO.%1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('LOCALHOST(%p%)')
CLUSTER(CL1) REPLACE | runmqsc %1
goto:eof

```

圖 28: 建立佇列管理程式: *qmgrs.bat*

透過將訂閱新增至叢集主題來更新配置。

```

@echo off
for %%A in (QMA QMB) do runmqsc %%A < wildsubs.tst
for %%A in (QMA QMB) do runmqsc %%A < upsubs.tst

```

圖 29: 更新訂閱: *upsubs.bat*

執行 *pub.bat*，並以佇列管理程式作為參數，以發佈包含發佈主題字串的訊息。 *Pub.bat* 使用範例程式 **amqspub**。

```

@echo off
@rem Provide queue manager name as a parameter
set S=Sports
set S=6 Sports/Football Sports/Football/Arsenal
set S=6 Sports/Rugby Sports/Rugby/Leeds
for %%B in (6) do echo %%B | amqspub %%B %1

```

圖 30: 發佈: *pub.bat*

## 發佈/訂閱拓撲

發佈/訂閱拓撲由佇列管理程式以及它們之間支援發佈/訂閱應用程式的連線組成。

發佈/訂閱應用程式可以由連接在一起的佇列管理程式網路組成。佇列管理程式可以全部位於相同的實體系統上，也可以分散在數個實體系統上。透過將佇列管理程式連接在一起，應用程式可以使用網路中的任何佇列管理程式來接收發佈。

這提供下列好處：

- 用戶端應用程式可以與鄰近的佇列管理程式進行通訊，而不是與較遠的佇列管理程式進行通訊，從而獲得更好的回應時間。
- 透過使用多個佇列管理程式，可以支援更多訂閱者。

您可以用兩種不同的方式來排列正在執行發佈/訂閱傳訊的佇列管理程式：叢集和階層。如需簡式叢集和簡式階層的範例，請參閱第 53 頁的圖 31 和第 53 頁的圖 32。如需這兩種拓撲的相關資訊，以及找出最適合您的拓撲，請參閱產品說明文件這一節中的資訊。

透過將階層中的叢集結合在一起，可以組合使用這兩種拓撲。

### Cluster

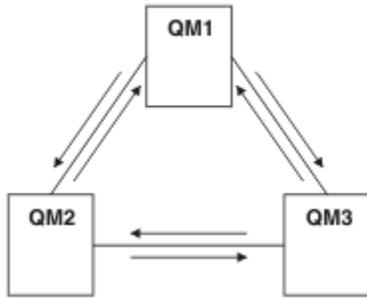


圖 31: 簡式發佈/訂閱叢集

### Hierarchy

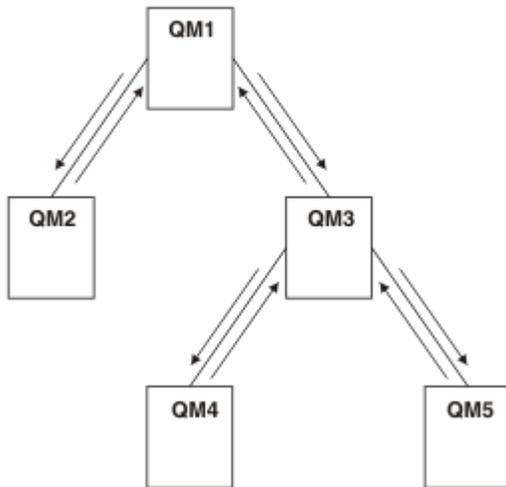


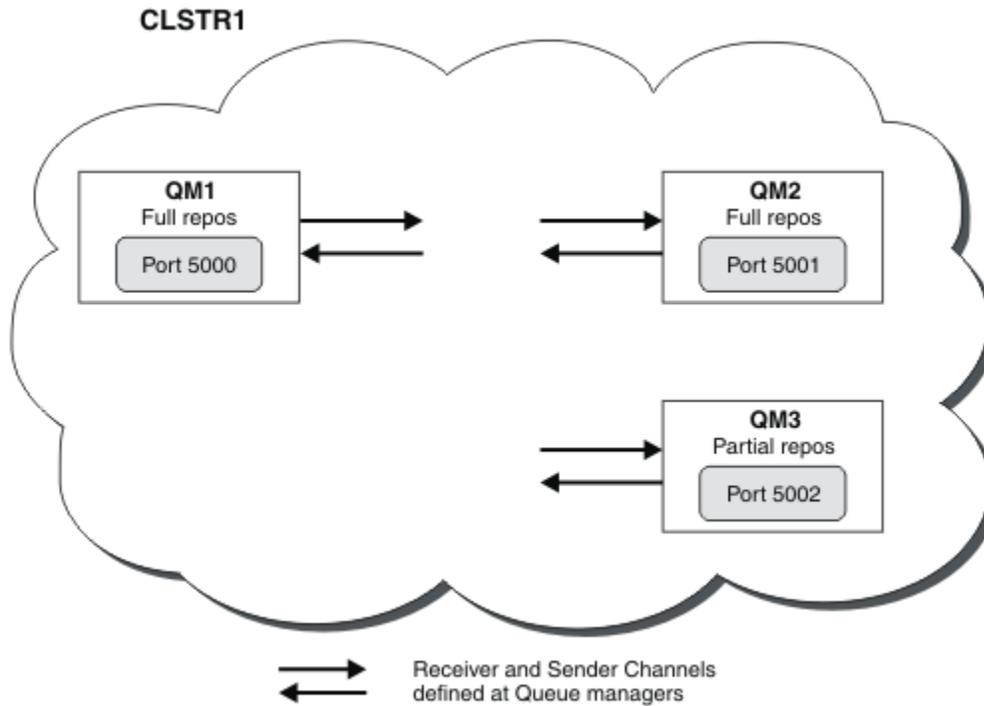
圖 32: 簡式發佈/訂閱階層

設定發佈/訂閱叢集: 實務 1

將兩個佇列管理程式作為完整儲存庫新增至叢集，並定義它們之間的通道。

## 關於這項作業

下列圖表有三個佇列管理程式: QM1、QM2 和 QM3:



QM1 和 QM2 是叢集中的完整儲存庫，QM3 是局部儲存庫。

實務範例 1 會將 QM1 和 QM2 新增至叢集 DEMO 作為完整儲存庫。

實務範例 2 會將 QM3 新增至叢集 DEMO 作為局部儲存庫。

這些作業至少需要一個指令視窗。

## 程序

1. 將 QM1 和 QM2 設為 DEMO 叢集的完整儲存庫:

```
alter QMGR REPOS(DEMO)
```

2. 定義並啟動 QM1 的接聽器:

```
define listener(QM1_LS) TRPTYPE(TCP) CONTROL(QMGR) PORT(5000)
start listener(QM1_LS)
```

3. 定義並啟動 QM2 的接聽器:

```
define listener(QM2_LS) TRPTYPE(TCP) CONTROL(QMGR) PORT(5001)
start listener(QM2_LS)
```

4. 定義 QM1 的接收端通道:

```
DEFINE CHANNEL(DEMO.QM1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5000)')
CLUSTER(DEMO) DESCR('TCP Cluster-receiver channel for queue manager QM1')
```

5. 定義從 QM1 到 QM2 的傳送端通道:

```
DEFINE CHANNEL(DEMO.QM2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5001)')
CLUSTER(DEMO) DESCR('TCP Cluster-sender channel from QM1 to queue manager QM2')
```

6. 定義 QM2 的接收端通道:

```
DEFINE CHANNEL(DEMO.QM2) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5001)')
CLUSTER(DEMO) DESCR('TCP Cluster-receiver channel for queue manager QM2')
```

7. 定義從 QM2 到 QM1 的傳送端通道:

```
DEFINE CHANNEL(DEMO.QM1) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5000)')
CLUSTER(DEMO) DESCR('TCP Cluster-sender channel from qm2 to qm1')
```

8. 在 QM1 上定義叢集主題 scores :

```
define topic(scores) TOPICSTR(/football) CLUSTER(DEMO)
```

9. 使用下列指令來驗證設定:

```
display topic(scores) type(all) clusinfo
display clusqmgr(*)
display chstatus(*)
```

10. 使用兩個指令視窗來測試設定:

a. 在第一個指令視窗中輸入此指令:

```
/opt/mqm/samp/bin/amqspub /FOOTBALL/scores QM1
```

b. 在第二個指令視窗中輸入下列指令:

```
/opt/mqm/samp/bin/amqssub /FOOTBALL/scores QM2
```

## 相關工作

[管理 WebSphere MQ 叢集](#)

[設定新的叢集](#)

[設定發佈/訂閱叢集: 實務 2](#)

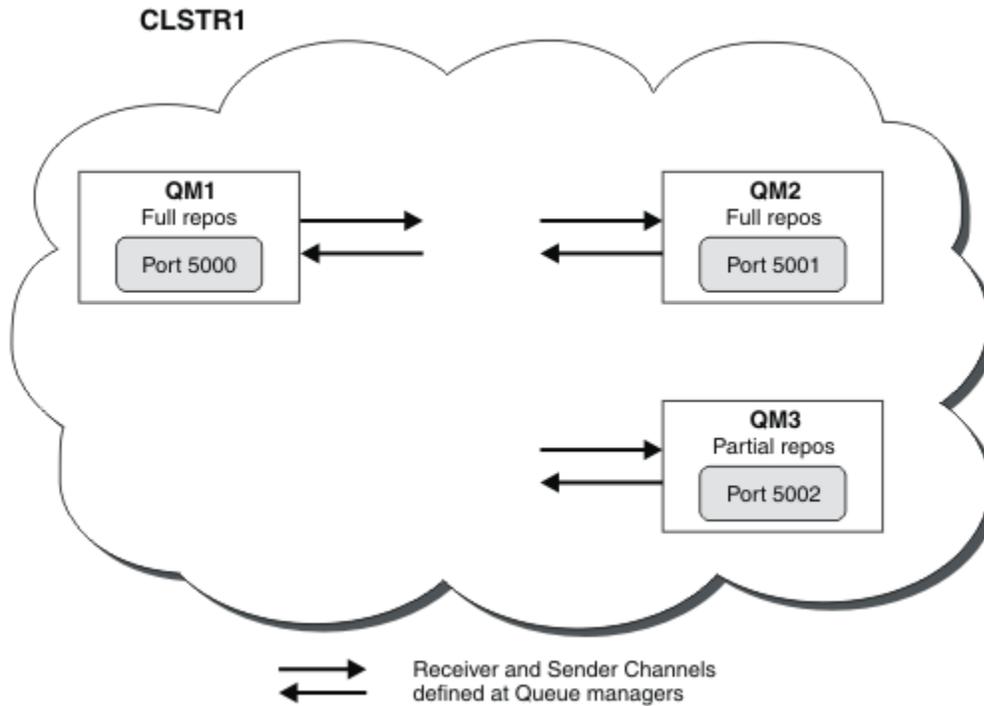
將第三個佇列管理程式作為局部儲存庫新增至叢集。

## 開始之前

在完成此作業之前，您必須已完成 [第 53 頁的『設定發佈/訂閱叢集: 實務 1』](#) 中的作業。

## 關於這項作業

下圖有 3 個佇列管理程式; QM1、QM2 和 QM3:



QM1 和 QM2 是叢集中的完整儲存庫，QM3 是局部儲存庫。

實務範例 1 會將 QM1 和 QM2 新增至叢集 DEMO 作為完整儲存庫。

實務範例 2 將 QM3 新增至叢集 DEMO 作為局部儲存庫。

這些作業至少需要 1 個指令視窗。

## 程序

1. 定義並啟動 QM3 的接聽器:

```
define listener(QM3_LS) TRPTYPE(TCP) CONTROL(QMGR) PORT(5002)
start listener(QM3_LS)
```

2. 定義 QM3 的接收端通道:

```
DEFINE CHANNEL(DEMO.QM3) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5002)') CLUSTER
(DEMO) DESCR('TCP Cluster-receiver channel for queue manager QM3')
```

3. 定義從 QM3 到 QM1 的傳送端通道:

```
DEFINE CHANNEL(DEMO.QM1) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('$HOSTNAME(5000)')
CLUSTER(DEMO) DESCR('TCP Cluster-sender channel from qm3 to qm1')
```

4. 使用下列指令來驗證設定:

```
display topic(scores) type(all) clusinfo
display clusqmgr(*)
display chstatus(*)
```

5. 使用 2 個指令視窗來測試設定:

- a. 在第一個指令視窗中輸入此指令:

```
/opt/mqm/samp/bin/amqspub /FOOTBALL/scores QM2
```

- b. 在第二個指令視窗中輸入下列指令:

## 發佈/訂閱叢集

發佈/訂閱叢集是交互連接的佇列管理程式的標準 IBM WebSphere MQ 叢集，其中發佈會自動從發佈應用程式移至存在於叢集中任何佇列管理程式上的訂閱。

用於發佈/訂閱傳訊的叢集與標準 IBM WebSphere MQ 叢集沒有不同。因此，發佈/訂閱叢集內的佇列管理程式可以存在於實際個別的電腦上，且每一對佇列管理程式會在必要時由叢集通道自動連接在一起。如需如何規劃及配置 IBM WebSphere MQ 叢集的相關資訊，請參閱 [叢集如何運作](#)。

透過在叢集中任何佇列管理程式所配置的主題上設定 **CLUSTER** 屬性，在定義叢集主題物件時建立發佈/訂閱叢集。主題定義會延伸到叢集的所有成員。您可以在叢集中的任何佇列管理程式上，發佈及訂閱主題，及其下方在主題樹狀結構中的任何主題字串。發佈會自動延伸到連接至叢集中其他佇列管理程式的訂閱者。

非叢集發佈/訂閱活動也可以在發佈/訂閱叢集中進行，方法是使用不屬於叢集主題物件的主題字串。此安排不同於發佈/訂閱階層，在此階層中，所有訂閱都會在整個階層中延伸。在這兩種情況下，都可以使用「訂閱」及「發佈」範圍來進行更細微的控制。

在發佈/訂閱拓撲中使用叢集提供下列好處：

- 以相同叢集中特定佇列管理程式上的訂閱為目的地的訊息會直接傳輸至該佇列管理程式，且不需要通過中繼佇列管理程式。相較於階層式拓撲，這可增進效能並最佳化佇列管理程式之間的發佈/訂閱資料流量。
- 因為所有佇列管理程式都直接彼此連接，所以此拓撲中沒有單一失敗點。如果一個佇列管理程式無法使用，叢集中其他佇列管理程式上的訂閱仍然能夠從可用佇列管理程式上的發佈者接收訊息。
- 在包含多個個別叢集的系統中，例如在地理上分散叢集的系統中，可以將叢集連接至叢集階層。此連線是透過結合每一個叢集中的單一佇列管理程式來建立，以啟用透過網路的發佈及訂閱流程；請參閱 [第 82 頁的『結合多個叢集的主題空間』](#)。您也可以控制哪些發佈資訊從一個叢集流向另一個叢集；請參閱 [第 83 頁的『結合並隔離多個叢集中的主題空間』](#)。
- 訂閱應用程式可以連接至其最接近的佇列管理程式，以改善其本身的效能。佇列管理程式會從叢集內的所有佇列管理程式接收符合用戶端訂閱登錄的所有訊息。

對於從這個佇列管理程式所要求的其他服務，也會改良用戶端應用程式的效能。用戶端應用程式可以使用發佈/訂閱及點對點傳訊。

- 透過將更多佇列管理程式新增至叢集以共用工作量，可以減少每一個佇列管理程式的用戶端及訂閱數目。發佈會自動配送至新佇列管理程式上的用戶端。對於部分使用型樣，此程序可以讓發佈/訂閱叢集拓撲具有高度可擴充性。

在發佈/訂閱中使用叢集時要考量的事項：

- 發佈/訂閱叢集中的所有佇列管理程式會自動瞭解叢集中的所有其他佇列管理程式。對於點對點叢集而言，此處理程序有所不同，其中只有佇列管理程式感興趣的佇列管理程式才是已知的。
- 發佈/訂閱叢集中管理叢集主題的一或多個訂閱的佇列管理程式，會自動建立叢集中所有其他佇列管理程式的叢集傳送端通道。佇列管理程式也會傳送每一個佇列管理程式的訂閱相關資訊，即使接收端佇列管理程式未發佈任何叢集化主題的訊息也一樣。
- 佇列管理程式上對叢集主題下的主題字串的第一個訂閱，會導致將訊息傳送至叢集中的所有其他佇列管理程式。同樣地，要刪除之主題字串的前次訂閱也會產生訊息。叢集主題下使用的個別主題字串越多，佇列管理程式之間的通訊就會越多。



### 注意：

基於本主題先前列出的原因，將叢集主題引進大型 IBM WebSphere MQ 叢集 (亦即，包含許多佇列管理程式的叢集) 可能會立即導致叢集中每一個佇列管理程式的額外負載，在某些情況下會導致效能降低。如需相關資訊，請參閱 [第 62 頁的『叢集主題效能』](#)。

必須仔細規劃將發佈/訂閱引進佇列管理程式叢集，特別是現有叢集，以因應這些效能降低。

在已知叢集無法容納發佈/訂閱效能降低的情況下，可以使用 **PSCLUS** 參數來停用佇列管理程式中的叢集發佈/訂閱功能。**PSCLUS** 參數主要是為了停止因意外或不正確地定義叢集主題而建立發佈/訂閱叢集時可能發生的嚴重問題。如需停用此功能的相關資訊，請參閱 [第 61 頁的『在叢集中禁止叢集化發佈/訂閱』](#)。

發佈/訂閱叢集作業: 最佳作法

本主題提供規劃及管理 IBM WebSphere MQ 發佈/訂閱叢集的指引。此資訊是根據客戶的測試和意見。

下列資訊假設使用者對 IBM WebSphere MQ 叢集、發佈/訂閱有基本瞭解，並且熟悉第 42 頁的『分散式發佈/訂閱』中的主題。這一資訊並不是“一刀切”的解決方案，而是試圖分享共同解決共同問題的方法。

## 發佈/訂閱叢集

使用叢集時，您在叢集中的佇列管理程式之間會有「任何對任何」直接連線功能 (必要時)。當叢集用於點對點傳訊時，叢集中的每一個佇列管理程式只知道其他叢集資源 (例如叢集中的其他佇列管理程式及叢集佇列) 的相關資訊，當應用程式連接至它們時，要求使用它們; 亦即，它們會以需要知道的方式運作。

發佈/訂閱叢集是佇列管理程式的叢集，具有一般 **CLUSDR** 及 **CLUSRCVR** 通道定義。不過，發佈/訂閱叢集也包含至少一個 **TOPIC** 物件，該物件定義在叢集中主題物件已識別叢集名稱的至少一個佇列管理程式上。

使用叢集中定義的主題物件，連接至叢集中某個佇列管理程式的應用程式可以訂閱該主題，或該主題下方主題樹狀結構中的任何節點，並從叢集中其他佇列管理程式接收該主題的發佈。在叢集中所有其他佇列管理程式上建立 **Proxy** 訂閱，以識別存在訂閱的佇列管理程式，即可達成此處理程序。因此，當有關主題的發佈在其佇列管理程式上發生時，他們知道將它轉遞至叢集的其他適當成員，並從該處將它遞送至個別應用程式訂閱。

為了達到此遞送，一旦將主題新增至叢集，叢集中的每個佇列管理程式都需要知道叢集中每個其他佇列管理程式的身分。此知識是透過叢集的完整儲存庫佇列管理程式來傳播。一個佇列管理程式上的已發佈訊息只會傳送至叢集中已知管理相同主題之訂閱的其他佇列管理程式。為了達到此程序，當應用程式建立對叢集主題的訂閱時，該佇列管理程式必須透過叢集傳送端通道，直接與叢集中的所有其他佇列管理程式進行通訊，以延伸 **Proxy** 訂閱。

此處理程序與您使用叢集進行點對點遞送時所需要的有限需要知道的資訊及連線功能有很大不同。因此，發佈/訂閱叢集上的需求與點對點叢集上的需求 (沒有任何叢集主題) 不同。

使用叢集主題可簡化佇列管理程式之間的發佈/訂閱網域延伸，但如果無法瞭解機制及含意，並考量用於發佈/訂閱的叢集，則可能會導致問題。下列最佳作法旨在協助進行此瞭解及準備。

總而言之，叢集式發佈/訂閱的效能含意可能對大型叢集有害，在嘗試使用現有叢集中的發佈/訂閱之前，需要仔細考量並瞭解。例如，即使是簡單建立叢集化主題物件。最好從專用於發佈/訂閱活動的小型新叢集開始，並從該處擴增叢集。

## 設計發佈/訂閱拓撲

如先前所述，當您在叢集中使用發佈/訂閱時，有一些容量和效能考量。因此，最佳作法是仔細考量佇列管理程式之間的發佈/訂閱需求，並將它限制為只需要佇列管理程式的數目。在識別需要發佈及訂閱一組主題的最小佇列管理程式集之後，它們可以成為只包含它們的叢集成員，而不包含其他佇列管理程式。

尤其在已建立的叢集中，對於點對點傳訊已運作良好。因此，當您將現有的大型叢集轉換成發佈/訂閱叢集時，最好是起始為發佈/訂閱工作建立個別叢集，以便在其中可以嘗試應用程式，而不是使用現行叢集。可以繼續使用已在一個以上點對點叢集中的現有佇列管理程式，這些佇列管理程式的子集需要成為新發佈/訂閱叢集的成員。不過，這個新叢集必須有個別的佇列管理程式配置成完整儲存庫，以便從現有的叢集完整儲存庫中隔離額外的負載。

如果您確定叢集不會因其大小或現行負載而用於發佈/訂閱，最好是透過在叢集中的任何佇列管理程式上簡單建立叢集主題，來防止此叢集非預期地變成發佈/訂閱叢集。使用 **PSCLUS** 佇列管理程式內容來達成此設計，如需詳細資料，請參閱 [禁止叢集中的叢集發佈/訂閱](#)。

謹慎選擇要新增至叢集的主題也很重要: 這些主題的主題樹狀結構越高，它們變得越廣泛。因此，不建議將主題根節點放入叢集中，而不考量所看到的行為。儘可能讓廣域主題顯而易見，例如在主題字串中使用高階限定元: `/global` 或 `/cluster`。

## 如何調整系統大小

發佈/訂閱叢集需要許多通道，因為模型與點對點傳訊不同: 每一個佇列管理程式都需要與該叢集中的所有其他佇列管理程式交談。點對點模型是「接受」的模型，但發佈/訂閱叢集具有不分青紅皂白的性質，且訂閱扇出。因此，完整儲存庫佇列管理程式，以及在發佈/訂閱叢集中管理本端訂閱的任何佇列管理程式，必須具有同時對叢集每一個成員建立通道的容量。

最好確保發佈/訂閱叢集中的每個佇列管理程式都可以達到此容量，但已確認主機訂閱永不知道的佇列管理程式不需要與其他佇列管理程式建立通道，因此不需要此容量層次。

不過，必須小心，因為在這類佇列管理程式上建立意外訂閱，或嘗試手動將這類佇列管理程式與叢集中的其他佇列管理程式重新同步化，都會導致同時啟動所有通道。如需相關資訊，請參閱第 59 頁的『重新同步化 Proxy 訂閱』。

叢集發佈/訂閱可將某個佇列管理程式上的已發佈訊息遞送至其他佇列管理程式上的訂閱。但對於點對點傳訊，在佇列管理程式之間傳輸訊息的成本可能對效能有害。因此，必須儘可能嘗試在發佈訊息的相同佇列管理程式上建立主題的訂閱。

另一個考量是對傳播 Proxy 訂閱系統效能的影響。通常，當建立特定叢集主題字串 (不只是已配置的主題物件) 的第一個訂閱時，佇列管理程式會將 Proxy 訂閱訊息傳送至叢集中的所有其他佇列管理程式。如果發佈/訂閱解決方案包含許多要訂閱的唯一主題字串，或主題經常訂閱及取消訂閱，則叢集中所有佇列管理程式之間可能會產生大量 Proxy 訂閱資料流量，對系統的整體效能產生不利影響。如需如何減少 Proxy 訂閱額外負擔的相關資訊，請參閱第 62 頁的『叢集主題效能』。

## 重新同步化 Proxy 訂閱

在正常情況下，佇列管理程式會自動確定系統中的 Proxy 訂閱正確反映叢集中每一個佇列管理程式上的訂閱。

不過，如果發生需要，您可以使用 `REFRESH QMGR TYPE (PROXYSUB)` 指令，手動重新同步化佇列管理程式的本端訂閱與它在叢集中延伸的 Proxy 訂閱。

**註：**重新同步會暫時在叢集上建立突然的額外 Proxy 訂閱負載，其源自發出指令的佇列管理程式。基於此原因，除非 IBM WebSphere MQ 服務、IBM WebSphere MQ 文件或錯誤記載指示您使用它，否則請勿使用它。

當需要重新同步化時的範例是當佇列管理程式無法正確傳播其 Proxy 訂閱時，可能是因為通道已停止，且所有訊息都無法排入佇列等待傳輸，或因為操作員錯誤導致訊息不正確地從 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 佇列中刪除。在此狀況下，請先更正原始問題 (例如重新啟動通道)，然後在佇列管理程式上發出 `REFRESH QMGR TYPE (PROXYSUB)` 指令。請注意，受影響的訂閱不會回復因未備妥 Proxy 訂閱而遺失的發佈。這個缺點必須加以考慮。

重新同步需要佇列管理程式啟動與叢集中所有其他佇列管理程式的通道。因此，您正在重新整理的佇列管理程式必須有足夠的能力來處理與叢集中所有其他佇列管理程式的通訊。

### 叢集主題

叢集主題是已定義 `cluster` 屬性的管理主題。叢集主題的相關資訊會推送至叢集的所有成員，並與本端主題結合，以在每一個佇列管理程式上建立不同的主題空間。

在佇列管理程式上定義叢集主題時，叢集主題定義會傳送到完整儲存庫佇列管理程式。然後，完整儲存庫會將叢集主題定義傳送到叢集內的所有佇列管理程式，使得相同的叢集主題可供叢集中任何佇列管理程式的發佈者和訂閱者使用。您在其中建立叢集主題的佇列管理程式稱為叢集主題主機。叢集主題可供叢集中的任何佇列管理程式使用，但必須在定義該主題的佇列管理程式 (主機) 上對叢集主題進行任何修改，此時修改會透過完整儲存庫傳送至叢集的所有成員。

在每一個佇列管理程式中，會從其察覺的本端及叢集主題定義建構單一主題名稱空間。當應用程式訂閱解析為叢集主題的主題時，IBM WebSphere MQ 會建立 Proxy 訂閱，並將它直接從建立訂閱的佇列管理程式傳送至叢集的所有其他成員。與叢集主題本身不同，Proxy 訂閱不會透過完整儲存庫佇列管理程式流動。

在主題上發佈的訊息會傳送至發佈者所連接的佇列管理程式已知的每個訂閱。如果其中任何訂閱是 Proxy 訂閱，則會將已發佈訊息的副本傳送至產生 Proxy 訂閱的佇列管理程式。然後，接收端佇列管理程式會將訊息副本傳送至每個本端訂閱。這項程序可確保叢集化主題的訂閱者會從叢集中任何佇列管理程式所連接的發佈者接收發佈資訊，並確保透過叢集傳送最少數目的已發佈訊息。

如果您有叢集主題及本端主題物件，則本端主題優先。如需相關資訊，請參閱第 60 頁的『多個叢集主題定義』。

如需用來顯示叢集主題之指令的相關資訊，請參閱下列相關鏈結：

## 萬用字元訂閱

當本端訂閱的主題字串解析於或低於叢集主題物件時，會建立 Proxy 訂閱。如果主題階層中的萬用字元訂閱高於任何叢集主題，則不會針對相符叢集主題在叢集周圍傳送 Proxy 訂閱，因此不會收到來自叢集其他成員的發佈。不過，它會從本端佇列管理程式接收發佈。

不過，如果另一個應用程式訂閱解析為叢集主題或低於叢集主題的主題字串，則會產生 Proxy 訂閱，並將發佈延伸到這個佇列管理程式。到達原始版本時，較高的萬用字元訂閱會被視為那些發佈的合法接受者，並接收副本。

此行為不同於相同主題上本端發佈的訊息。如果不需要此行為，則在叢集化主題上設定 **WILDCARD (BLOCK)** 會使原始萬用字元不被視為合法訂閱，且不會收到叢集主題或其子主題上的任何發佈 (本端或來自叢集中其他位置)。

### 相關概念

[使用管理主題](#)

[使用訂閱](#)

### 相關參考

[顯示主題](#)

[DISPLAY TPSTATUS](#)

[DISPLAY SUB](#)

### 叢集主題屬性

需要充分瞭解叢集主題屬性，才能設計及管理發佈/訂閱叢集。

主題物件具有許多適用於多佇列管理程式發佈/訂閱拓撲的屬性。當您使用 IBM WebSphere MQ 叢集來建立此類拓撲時，這些屬性具有下列行為。

### PROXYSUB

- **PROXYSUB** 是一個屬性，可控制何時進行 Proxy 訂閱。如需為何您可能想要將此屬性變更為 FIRSTUSE 預設值的詳細資料，請參閱第 45 頁的『遞送機制的相關資訊』。
- 如同叢集主題的其他屬性一樣，**PROXYSUB** 屬性會延伸到叢集中的每個佇列管理程式，而不只是主題定義所在的佇列管理程式。這會立即導致叢集中的每個佇列管理程式建立對所有其他佇列管理程式的萬用字元 Proxy 訂閱。此處理程序的結果是每個佇列管理程式都會建立與每個其他佇列管理程式的叢集傳送端通道，且任何已發佈的訊息都會傳送至每個佇列管理程式。

### PUBSCOPE 和 SUBSCOPE

**PUBSCOPE** 及 **SUBSCOPE** 決定此佇列管理程式是否將發佈資訊傳播至拓撲中的佇列管理程式 (發佈/訂閱叢集或階層)，或將範圍限制為僅其本端佇列管理程式。您可以使用 MQPMO\_SCOPE\_QMGR/MQSO\_SCOPE\_QMGR 以程式化方式執行對等工作。

- **PUBSCOPE** 如果使用 **PUBSCOPE (QMGR)** 定義叢集主題物件，則會與叢集共用定義，但基於該主題的發佈資訊範圍僅限本端，且不會傳送至叢集中的其他佇列管理程式。
- **SUBSCOPE** 如果使用 **SUBSCOPE (QMGR)** 定義叢集主題物件，則會與叢集共用定義，但基於該主題的訂閱範圍僅限本端，因此不會將任何 Proxy 訂閱傳送至叢集中的其他佇列管理程式。

這兩個屬性通常一起用來隔離佇列管理程式與叢集其他成員在特定主題上的互動。佇列管理程式不會與叢集其他成員之間來回發佈或接收關於這些主題的發佈。如果在子主題上定義主題物件，則此狀況不會阻止發佈或訂閱。

在主題的本端定義上將 **SUBSCOPE** 設為 QMGR，並不會阻止叢集中的其他佇列管理程式將其 Proxy 訂閱傳送至佇列管理程式 (如果它們使用主題的叢集版本，並搭配 **SUBSCOPE (ALL)**)。不過，如果本端定義也將 **PUBSCOPE** 設為 QMGR，則這些 Proxy 訂閱不會從此佇列管理程式傳送發佈。

### 多個叢集主題定義

本端主題定義會置換相同名稱的遠端定義叢集主題定義。也可以在叢集中的不同佇列管理程式上建立相同叢集主題的多重定義。不過，這兩種情況都需要小心，本主題會說明原因。

就像叢集佇列一樣，在叢集中具有相同叢集主題物件的多個定義，會產生在每一個上定義不同內容的可能性。很難判斷叢集中每一個佇列管理程式所看到的主題定義版本，因此很難判斷預期的行為。

如果單一主題字串有兩個以上叢集主題定義的屬性不同，或存在於多個叢集中，則會將訊息 (AMQ5465 & AMQ5466) 寫入錯誤日誌中，並使用最近收到的叢集主題定義。

叢集主題主機佇列管理程式不得刪除主題定義，且會保留在叢集中，以確保叢集的所有成員持續知道叢集主題。這個主機佇列管理程式持續可用並不重要，因為叢集主題定義是由完整儲存庫佇列管理程式及其局部叢集儲存庫中的所有其他佇列管理程式所快取。這項快取作業容許在無法使用主機佇列管理程式時，有至少 60 天的可用性。如需此主題的相關資訊，請參閱第 65 頁的『[發佈/訂閱叢集佇列管理程式的主要角色](#)』。

## 在本端置換叢集主題定義

在叢集中的特定佇列管理程式上，可能需要置換叢集主題的行為。透過定義本端主題物件來置換具有相同主題字串的叢集主題物件，並使用它來僅發佈至本端連接的訂閱者，可以達到此置換。

即使建立主題的本端定義來置換佇列管理程式上的叢集主題，佇列管理程式仍會使用叢集主題定義，繼續接收來自叢集其他成員的 Proxy 訂閱。依預設，本端發佈的訊息會繼續傳送至遠端佇列管理程式，以允許使用 Proxy 訂閱。如果不需要此安排，請在本端主題物件上指定 **PUBSCOPE(QMGR)**，以確保連接至此佇列管理程式的發佈者應用程式僅發佈至本端訂閱者。

## 修改叢集主題定義

如果您需要變更叢集主題定義，請在叢集主題主機所定義的相同佇列管理程式中修改它。請勿在叢集中的不同佇列管理程式上建立相同叢集主題的定義。再次定義主題會導致同一叢集主題有兩個叢集主題主機。

多次定義叢集主題會建立潛在衝突的定義，以及不同佇列管理程式在不同時間使用不同定義的可能性。

## 將叢集主題定義移至叢集中的不同佇列管理程式

您可能需要將叢集主題定義從叢集中的一個佇列管理程式移至另一個佇列管理程式，例如從叢集中解除佇列管理程式任務時。若要将叢集主題定義移至叢集中的不同佇列管理程式，而不岔斷發佈資訊的流程，您需要遵循下列步驟。此範例會將定義從 QM1 移至 QM2。

1. 在 QM2 上建立重複的叢集主題定義，其屬性與 QM1 的定義相同。
2. 等待完整儲存庫佇列管理程式將新定義延伸到整個叢集。您可以利用 (**DISPLAY CLUSTER**) 指令來顯示每一個叢集成員的叢集主題，以及檢查來自 QM2 的定義，來判斷傳送的方式。
3. 從 QM1 中刪除叢集主題定義。

從 QM1 刪除原始定義之後，必要的話，可以在 QM2 上修改定義，而不會在內容中造成衝突。

## 取代失敗佇列管理程式上的叢集主題定義

在前一個實務範例中，如果 QM1 有一段時間無法使用，則可能無法從 QM1 中刪除定義。在此情況下，可以接受在兩個定義同時存在的情況下執行。

如果隨後變成需要修改叢集主題定義，則可以修改 QM2 上的版本，因為知道 QM2 定義比 QM1 定義新，因此會優先使用。不過，在此期間，會將錯誤寫入佇列管理程式的錯誤日誌，因為有衝突的叢集主題定義。當可以重新啟動時，請從 QM1 中移除重複的叢集主題定義，以儘快解決錯誤。

或者，如果 QM1 永不會回到叢集 (例如，災難性硬體故障之後的非預期解除任務)，則可以使用 **RESET CLUSTER** 指令強制退出佇列管理程式。**RESET CLUSTER** 會自動刪除目標佇列管理程式上管理的所有主題物件。

### 在叢集中禁止叢集化發佈/訂閱

將發佈/訂閱引進佇列管理程式叢集，特別是現有的叢集，必須小心規劃，以因應效能的任何降低。

將叢集主題引入大型 IBM WebSphere MQ 叢集 (包含許多佇列管理程式的叢集) 可能會立即導致叢集中每一個佇列管理程式的額外負載，在某些情況下會降低效能。因此，必須仔細規劃發佈/訂閱的引進。如需相關資訊，請參閱第 62 頁的『[叢集主題效能](#)』。

如果已知叢集無法容納發佈/訂閱的額外需要，則可以透過將佇列管理程式屬性 **PSCLUS** 設為 **DISABLED** 來停用佇列管理程式中的叢集發佈/訂閱功能。

將 **PSCLUS** 設為 DISABLED 會修改佇列管理程式功能的三個層面:

- 此佇列管理程式的管理者不再能夠將 Topic 物件定義為叢集。
- 來自其他佇列管理程式的送入主題定義或 Proxy 訂閱會被拒絕 (會記載一則警告訊息, 以通知管理者配置不正確)。
- 完整儲存庫在收到主題定義時, 不再自動與其他局部儲存庫共用每個佇列管理程式的相關資訊。

雖然 **PSCLUS** 是叢集中每個個別佇列管理程式的參數, 但並不打算在叢集的佇列管理程式子集中選擇性地停用發佈/訂閱。除了任何其他方法之外, 此方法會導致將頻繁的錯誤訊息視為 Proxy 訂閱, 且會持續查看及拒絕主題定義。理想情況下, 當使用這個選項時, 會一致地將叢集中的所有佇列管理程式設為已停用。如果佇列管理程式參與一或多個發佈訂閱叢集, 以及一或多個傳統叢集, 則 **PSCLUS** 必須在該佇列管理程式上設為 ENABLED。請參閱下列在完整儲存庫中停用的相關資訊。

重要的是, 在叢集中所有完整儲存庫佇列管理程式上將 **PSCLUS** 設為 DISABLED, 可避免在未正確配置的局部儲存庫上, 任何叢集主題定義影響叢集中的其他佇列管理程式。在這種情況下, 會在完整儲存庫佇列管理程式的錯誤日誌中報告不一致。

將傳統點對點叢集與發佈訂閱叢集重疊時, 請務必在每一個叢集中使用一組個別的完整儲存庫。此安排容許主題定義及「所有佇列管理程式」資訊僅在發佈/訂閱叢集中流動。

使用此參數時有一些警告, 有助於避免配置不一致。從 ENABLED 修改為已停用时, 此佇列管理程式所屬的任何叢集中都不能存在任何叢集主題物件。停用此功能之前, 必須先刪除任何這類主題 (甚至遠端定義的主題)

如需 **PSCLUS** 的相關資訊, 請參閱 [ALTER QMGR \(PSCLUS\)](#)。

#### 叢集主題效能

叢集主題的效能性質需要特殊考量, 因為它們與叢集佇列的效能性質不同, 而且在大型或不平衡叢集中, 考量不足的使用情形可能是效能問題的來源。

## 減少發佈/訂閱對效能的影響

叢集中佇列管理程式有兩個工作量來源: 直接處理應用程式的訊息, 以及處理管理叢集所需的訊息和通道。在一般點對點叢集中, 叢集系統工作量主要受限於叢集成員視需要明確要求的資訊 (請參閱第 64 頁的『發佈/訂閱叢集的效能性質』中的比較)。因此, 在除了非常大的叢集 (例如包含數千個佇列管理程式的叢集) 之外的任何其他叢集中, 您可以在考量佇列管理程式效能時大幅降低管理叢集的效能效果。

在發佈/訂閱叢集中, 不論所有叢集佇列管理程式是否主動參與發佈/訂閱傳訊, 都會將資訊 (例如叢集主題及 Proxy 訂閱) 推送至叢集的所有成員。此處理程序可能會在系統上產生大量額外負載。因此, 您確實需要考量叢集管理對佇列管理程式效能的影響, 包括其計時及其大小。

若要減少發佈/訂閱叢集管理對叢集效能的影響, 請考量下列兩項建議:

1. 在一天中的離峰時間執行叢集、主題及訂閱更新。
2. 如果您只因為現有大型叢集已存在, 而考慮將發佈/訂閱主題新增至現有大型叢集, 請考量是否可以定義發佈/訂閱所涉及的較小佇列管理程式子集, 並使其成為「重疊」叢集。此叢集是定義叢集主題的叢集。雖然部分佇列管理程式現在位於兩個叢集中, 但發佈/訂閱的整體效果會減少:
  - a. 發佈/訂閱叢集的大小較小。
  - b. 不在發佈/訂閱叢集中的佇列管理程式受叢集管理資料流量的影響較小。

## 平衡生產者和消費者

非同步傳訊效能中的重要概念是平衡。除非訊息消費者與訊息產生者保持平衡, 否則未耗用訊息的待辦事項可能會累積並嚴重影響多個應用程式的效能。

在點對點傳訊拓撲中, 訊息消費者與訊息產生者之間的關係很容易理解。您可以取得訊息正式作業和耗用的估計值, 依佇列排列, 依通道排列。如果缺乏平衡, 則會立即找出瓶頸, 然後加以補救。

在發佈/訂閱拓撲中, 很難找出發佈者和訂閱者是否平衡。從每一個解析為叢集主題的訂閱開始, 然後回到具有該主題的發佈者的佇列管理程式。計算從每一個佇列管理程式流向每一個訂閱者的發佈數。

符合叢集中遠端佇列管理程式上的訂閱 (根據 Proxy 訂閱) 的每一個發佈都會放置到 `SYSTEM.CLUSTER.TRANSMIT.QUEUE`。如果多個遠端佇列管理程式具有該發佈的 Proxy 訂閱，則會將多個訊息副本放置到傳輸佇列，並以不同的叢集傳送端通道為目標。

這些發佈會以遠端佇列管理程式上的 `SYSTEM.INTER.QMGR.PUBS` 佇列為目標。每一個佇列管理程式都會處理到達該佇列的訊息，並將它們遞送至該佇列管理程式上的正確訂閱。

基於此原因，請在可能出現瓶頸的下列點監視負載：

- 個別訂閱本身會將佇列：
  - 這個瓶頸會暗示訂閱應用程式不會像發佈時一樣快速使用發佈。
- `SYSTEM.INTER.QMGR.PUBS` 佇列：
  - 佇列管理程式從一或多個遠端佇列管理程式接收發佈資訊的速度，比它將它們配送至本端訂閱的速度還要快。
- 發佈佇列管理程式、訂閱佇列管理程式及發佈佇列管理程式上叢集傳輸佇列之間的叢集通道 (依預設為 `SYSTEM.CLUSTER.TRANSMIT.QUEUE`)：
  - 一個以上叢集通道不在執行中，或正在將訊息發佈至本端佇列管理程式的速度比通道將它們遞送至遠端佇列管理程式的速度更快。
- 如果發佈應用程式正在使用已排入佇列的發佈/訂閱介面，則 `SYSTEM.BROKER.DEFAULT.STREAM` 佇列及 `SYSTEM.QPUBSUB.QUEUE.NAMELIST` 中列出的任何其他串流佇列，以及 `SYSTEM.BROKER.DEFAULT.SUBPOINT` 佇列及任何其他子點佇列 (如 `SYSTEM.QPUBSUB.SUBPOINT.NAMELIST` 中列出) 也需要考量：
  - 本端發佈應用程式放置訊息的速度比本端佇列管理程式處理訊息的速度還要快。

## 訂閱效能考量

如先前所述，當在佇列管理程式上針對解析為叢集主題的主題字串進行訂閱時，該佇列管理程式必須確保叢集中的每一個其他佇列管理程式都已備妥該主題的 Proxy 訂閱。為了達到此結果，佇列管理程式會建立 Proxy 訂閱訊息，並將其傳送至叢集中的所有其他佇列管理程式。

使用預設配置，只有在本端佇列管理程式上已有訂閱完全相同的主題字串時，建立叢集主題的訂閱並不會導致傳送新的 Proxy 訂閱。在此狀況下，不需要其他 Proxy 訂閱，因為到達的發佈會遞送至所有符合的訂閱，而不只是主題字串的原始訂閱。

如需預設配置的替代方案，請參閱第 64 頁的『停用個別 Proxy 訂閱』。

訂閱選取元不會納入考量，因此相同主題字串的兩個訂閱 (但具有不同的選取元) 仍會共用 Proxy 訂閱。此狀況也可能表示符合主題字串的發佈會延伸到訂閱者佇列管理程式，即使發佈不符合訂閱的選取器。

當從佇列管理程式刪除主題字串的最後一個訂閱時，會建立與 Proxy 訂閱訊息相等的訊息，並傳送至所有佇列管理程式。此程序會從遠端佇列管理程式移除 Proxy 訂閱。

基於這些原因，叢集大小及不同主題字串的訂閱頻率可能會對叢集本身造成重大負載，在規劃叢集及發佈/訂閱應用程式要使用的主題時必須考量。

考量來自 Proxy 訂閱資料流量的系統負載時，除了監視第 62 頁的『平衡生產者和消費者』區段中列出的佇列之外，還請監視下列佇列。

- `SYSTEM.INTER.QMGR.FANREQ` 佇列。
- `SYSTEM.INTER.QMGR.CONTROL` 佇列。

這些佇列上的任何重要訊息待辦事項都意味著訂閱變更率對系統而言太大，或佇列管理程式在叢集中無法正確運作。由於已停用發佈/訂閱支援 (請參閱 `ALTER QMGR` 中的 `PSMODE`)，或發生需要進一步調查的問題，此時請檢查佇列管理程式錯誤日誌。

## 減少 Proxy 訂閱資料流量

如果 Proxy 訂閱額外負擔很高，則應該採取步驟來減少此負擔。這可能透過一般主題合併或變更為佇列間管理程式發佈資訊的播送模型來達成。

一般的發佈/訂閱建議是評量主題字串的使用，以查看是否可以合併它們，以減少系統資源的整體負載。在連接發佈者或訂閱的系統中，使用許多不同的暫時性主題字串會在每一個佇列管理程式上引進某種層次的管理額外負擔。減少主題字串的數目和暫時性，因此它們的發佈者和訂閱會減少對系統的影響。

減少 Proxy 訂閱資料流量的一種方法是在相同佇列管理程式上尋找相同主題字串的訂閱。此方法可讓此佇列管理程式將單一 Proxy 訂閱傳送至其他佇列管理程式，而不是讓多個佇列管理程式傳送 Proxy 訂閱，每個佇列管理程式都在同一主題字串上針對自己的一組訂閱傳送 Proxy 訂閱。此作法也會將跨叢集的發佈遞送最佳化。

## 停用個別 Proxy 訂閱

在某些情況下，在叢集中訂閱的不同主題字串集很大且持續變更，最好從訂閱延伸模型變更為發佈播送模型。在此偏好模型中，任何叢集主題上的每個發佈都會自動傳送至叢集中的每個佇列管理程式，而不論那些佇列管理程式上是否存在訂閱。

然後，接收端佇列管理程式可以將訊息遞送至存在的本端訂閱，或捨棄訊息。在此模型中，不需要根據訂閱的存在來建立及刪除個別 Proxy 訂閱。在此模式下執行時，可能已發佈的訊息資源負載會隨著所有發佈資訊傳送至所有佇列管理程式而增加。因此，叢集中的佇列管理程式必須具有處理此額外負載的容量。

使用下列配置步驟來啟用廣播模型：

1. 每個管理訂閱的佇列管理程式都必須配置成不傳送符合叢集主題本端訂閱的 Proxy 訂閱。此配置需要在每一個佇列管理程式 `qm.ini` 檔中設定下列調整參數，然後才能定義叢集主題或在叢集中建立訂閱：

```
TuningParameters:  
pscProxySubFlags=1
```

2. 設定調整參數之後，必須重新啟動所有佇列管理程式。
3. 在重新啟動佇列管理程式之後，可以定義叢集主題。每一個叢集主題都必須將 **PROXYSUB** 設為 **FORCE**。

### 反轉行為

若要反轉 第 64 頁的『停用個別 Proxy 訂閱』中先前說明的作業模式，請使用下列步驟：

1. 從 `qm.ini` 檔案中移除每個佇列管理程式的調整參數。
2. 重新啟動每個佇列管理程式。
3. 在管理訂閱的每個佇列管理程式上發出 **REFRESH QMGR TYPE (PROXYSUB)** 指令。
4. 在叢集主題上，將 **PROXYSUB** 設為 **FIRSTUSE**。



**注意：**在此行為的啟用及反轉中，如果所有步驟未以記載的順序完成，則可能不會發生訂閱的正確發佈流程。

## 附註：隱含設定 PROXYSUB (設為 FORCE)

如本主題先前所述，**PROXYSUB (FORCE)** 主題屬性可以減少 Proxy 訂閱資料流量，但必須小心使用。**PROXYSUB (FORCE)** 屬性會延伸到叢集中的每個佇列管理程式，而不只是定義主題的佇列管理程式。這會立即導致叢集中的每個佇列管理程式建立對所有其他佇列管理程式的萬用字元 Proxy 訂閱。此處理程序的結果是每個佇列管理程式都會建立與每個其他佇列管理程式的叢集傳送端通道，並將任何已發佈訊息傳送至每個佇列管理程式。

在大型或忙碌叢集中設定此內容可能會導致系統資源上的額外負載。

## 發佈/訂閱叢集的效能性質

請務必考量變更發佈/訂閱叢集的屬性 (例如將佇列管理程式、主題或訂閱新增至叢集) 如何影響叢集中執行的應用程式效能。

比較兩個管理作業的點對點叢集與發佈/訂閱叢集。

首先，點對點叢集：

1. 當定義新的叢集佇列時，目的地資訊會推送至完整儲存庫佇列管理程式，且只會在其他叢集成員第一次參照叢集佇列時 (例如，應用程式嘗試開放它)，才會傳送至其他叢集成員。然後佇列管理程式會在本端快取此資訊，以移除每次存取佇列時從遠端擷取資訊的需求。

2. 將佇列管理程式新增至叢集不會直接影響其他佇列管理程式的負載。新佇列管理程式的相關資訊會推送至完整儲存庫，但只有在資料流量開始流入或流出新佇列管理程式時，才會建立及啟動從叢集中其他佇列管理程式到新佇列管理程式的通道。

簡言之，點對點叢集中佇列管理程式的負載與它針對應用程式所處理的訊息資料流量相關，且與叢集大小沒有直接相關。

其次，發佈/訂閱叢集：

1. 定義新的叢集主題時，會將資訊推送至完整儲存庫佇列管理程式，並從該處直接推送至叢集的所有成員，這會導致從完整儲存庫啟動通道至叢集的每一個成員 (如果尚未啟動的話)。
2. 在新主題字串上建立叢集主題的訂閱時，會立即將資訊從該佇列管理程式直接推送至叢集的所有其他成員，這會導致通道從該佇列管理程式啟動至叢集的每一個成員 (如果尚未啟動的話)。
3. 當新的佇列管理程式加入現有叢集時，會將所有叢集主題的相關資訊從完整儲存庫佇列管理程式推送至該叢集。然後，新的佇列管理程式會將叢集中叢集主題的所有訂閱的知識與叢集的所有成員同步化，從而建立通道並從新的佇列管理程式啟動至叢集的每一個成員。

總之，叢集中任何佇列管理程式的叢集管理負載會隨著叢集內的佇列管理程式、叢集主題及 Proxy 訂閱數目而增加，而不論每一個佇列管理程式上這些叢集主題的本端用途。

**發佈/訂閱叢集佇列管理程式的主要角色**

與點對點叢集類似，發佈/訂閱叢集中的佇列管理程式有兩個重要角色；作為完整儲存庫佇列管理程式及叢集主題主機。

#### **完整儲存庫**

完整儲存庫佇列管理程式具有將物件定義推送至叢集其他成員的角色；在發佈/訂閱叢集的情況下，將叢集主題物件定義推送至叢集其他成員。

#### **叢集主題主機**

叢集主題主機是在其中定義叢集主題物件的佇列管理程式。您可以在發佈/訂閱叢集中的任何佇列管理程式上定義叢集主題物件。叢集主題物件會推送至完整儲存庫佇列管理程式，然後將它推送至叢集中的所有其他佇列管理程式，供叢集中任何佇列管理程式上執行的發佈者及訂閱者使用。

## **可用性和管理**

您應該在叢集中定義兩個完整儲存庫，以最大化叢集中叢集主題定義的可用性。

對於已排入佇列的傳訊叢集，在許多電腦中只有兩部高可用性電腦的發佈/訂閱叢集中，最好將高可用性電腦定義為完整儲存庫。

在已排入佇列的叢集中，您可以在叢集中的多個佇列管理程式上定義相同的叢集佇列，以增加叢集佇列的可用性及傳輸量。然後會平衡訊息之間的工作量。相反地，在發佈/訂閱叢集中，叢集主題在叢集中的所有佇列管理程式上可用，但不會執行發佈/訂閱資料流量的工作量平衡。相反地，不同的訂閱和發佈者應該分散在不同的佇列管理程式中，以分散發佈/訂閱負載。如果您定義叢集主題所在的佇列管理程式變成無法使用，其他佇列管理程式會繼續處理主題的發佈/訂閱要求。

不過，如果您在其中定義叢集主題物件的佇列管理程式再也無法使用，則最終會刪除其他佇列管理程式上的快取主題物件，且主題會變成無法使用。從主題定義變成無法使用開始，至少 60 天之後 (視前次重新整理主題定義的時間而定) 就會發生此處理程序。

使用 60 天期間來回復已定義叢集主題物件的佇列管理程式，幾乎不需要採取特殊措施，讓叢集主題主機具有高可用性。60 天期限足以應付技術問題；60 天期限只能因行政錯誤而出。為了減少這種可能性，如果叢集主題主機無法使用，叢集的所有成員都會每小時寫入錯誤日誌訊息，指出其快取的叢集主題物件未重新整理。請確一定義叢集主題物件的佇列管理程式正在執行中，以回應此訊息。

您可以採用在其他佇列管理程式上定義相同叢集主題物件的作法。每一個定義都會導致將其他叢集主題物件推送至叢集中的其他佇列管理程式，包括其他叢集主題主機。現在，如果叢集主題主機變成無法使用超過 60 天，則只會從其他主機移除其叢集主題物件版本。仍會保留叢集主題物件的其他版本。叢集中特定主題的所有定義都必須相同，否則難以確定佇列管理程式正在使用哪個主題定義。任何主機上的最新副本一律是所使用的叢集主題物件。

權衡多重叢集主題定義的額外保護，以防止管理複雜性增加：隨著複雜性增加，人為錯誤的機會也會增加。

與管理叢集佇列不同，叢集主題定義的主機佇列管理程式不會引進任何其他應用程式訊息資料流量。該資料流量僅限於建立訂閱及發佈訊息所在的佇列管理程式。您可以在佇列管理程式上管理叢集主題，但兩者都不執行。此狀況表示雖然不是必要，但通常可管理叢集的完整儲存庫佇列管理程式上的叢集主題，因為這些佇列管理程式可能佈建有更高的可用性層次，並對它們具有更嚴格的管理控制。這種安排可減少錯誤地修改或刪除定義，甚至佇列管理程式的可能性。

#### 重疊叢集支援及發佈/訂閱

使用 IBM WebSphere MQ 叢集，單一佇列管理程式可以是多個叢集的成員。此排列稱為重疊叢集。當叢集在佇列管理程式中重疊時，發佈/訂閱叢集中的叢集主題的行為與佇列不同。使用具有重疊叢集的叢集發佈/訂閱時，必須清楚瞭解此行為。

與佇列不同，無法將主題定義與多個叢集相關聯。因此，在叢集中建立的 Proxy 訂閱範圍限制為在其中定義叢集主題的單一叢集。不過，每一個佇列管理程式都有單一主題樹狀結構，其中包含所有本端主題及任何已知叢集主題(來自它們所屬的任何叢集)。因此，可以建構這樣的系統，讓發佈/訂閱行為難以理解。

#### 整合多個發佈/訂閱叢集

對於點對點訊息，讓單一佇列管理程式成為多個叢集的成員的原因是在兩個叢集之間建立叢集閘道。如需此主題的相關資訊，請參閱 [重疊叢集](#)。這個叢集閘道可讓一個叢集中的點對點訊息遞送至另一個叢集中的查詢。發佈/訂閱叢集繼承傳統佇列管理程式叢集的重疊功能。不過，您無法使用此機制，將發佈和訂閱從一個叢集遞送至另一個叢集。

相反地，若要将發佈和訂閱從一個叢集中的佇列管理程式傳遞至另一個叢集，您必須使用發佈/訂閱階層將佇列管理程式鏈結在一起。透過在一個叢集中的一個佇列管理程式與另一個叢集中的另一個佇列管理程式之間明確建立上下代階層式關係，可以達成此安排。此關係會啟用叢集之間所有 Proxy 訂閱的流程，因此會啟用任何相符發佈。如需此關係的相關資訊，請參閱 [第 68 頁的『發佈/訂閱階層』](#)。

限制哪些發佈和訂閱在叢集之間流動的方法是使用既不在叢集中，也不在叢集中的閘道佇列管理程式；請參閱 [第 83 頁的『結合並隔離多個叢集中的主題空間』](#)。

#### 重疊叢集，單一主題樹狀結構

每一個佇列管理程式都有單一主題樹狀結構，其中包含本端主題及所有已知叢集主題。使用發佈/訂閱來重疊兩個叢集的進一步考量是，每一個叢集中的佇列管理程式可以定義具有相同名稱的叢集主題，或定義具有相同主題字串且名稱不同的叢集主題。在這兩個叢集的成員佇列管理程式上，當它們收到多個叢集主題定義的通知(每個叢集各一個)時，就會發生衝突。佇列管理程式會報告問題，但佇列管理程式會繼續運作，只使用最新的叢集主題定義。因此，行為會變成非決定性的，且無法依賴。

因此，使用叢集發佈/訂閱的重疊叢集必須考量其主題定義名稱空間來跨越所有叢集，並相應地命名其主題物件及建構其主題字串。然後，您可以可預測地使用重疊中的佇列管理程式來發佈及訂閱這兩個叢集。

在 [第 67 頁的圖 33](#) 中， $T_B$  和  $T_C$  是不重疊的主題定義。連接至叢集中的 QM3 的發佈者可以發佈至其各自叢集中的兩個主題。在重疊中連接至 QM3 的訂閱者能夠訂閱兩個叢集中的主題。

關於 [第 67 頁的圖 33](#) 的另一種思考方式是考量 Proxy 訂閱。連接至佇列管理程式 QM3 的應用程式，如果訂閱的主題解析為主題物件  $T_B$  (只存在於 CLUSTER 1 中)，則會導致將 Proxy 訂閱從佇列管理程式 QM3 傳送至佇列管理程式 QM1，且只 QM2。連接至佇列管理程式 QM3 的應用程式會在解析為主題物件  $T_C$  (只存在於 CLUSTER 2 中) 的主題上訂閱。訂閱會導致 Proxy 訂閱僅從佇列管理程式 QM3 傳送至佇列管理程式 QM4 及 QM5。

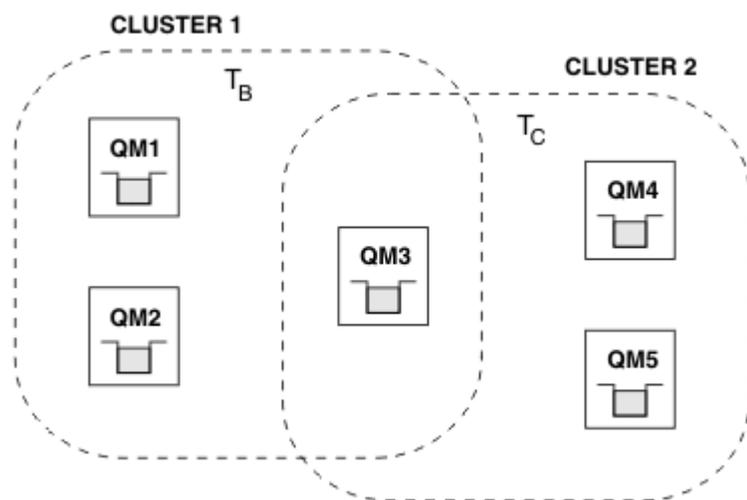


圖 33: 重疊叢集: 兩個叢集, 每個叢集訂閱不同的主題

沒有重疊的佇列管理程式的發佈者及訂閱者只能發佈及訂閱其叢集中的主題, 例如 QM2 上主題字串的訂閱者不會接收發佈至從 QM5 發佈之相同主題字串的訊息, 而不論主題叢集作業為何。若要達成此安排, 需要發佈/訂閱階層。

### 重疊叢集, 萬用字元訂閱

基於本主題前一節的原因, 當使用萬用字元來訂閱屬於多個叢集的佇列管理程式上的主題時, 必須小心。

在前一個範例中, 假設兩個主題物件配置為:

- $T_B$ : 主題名稱 'Football', 叢集 'CLUSTER1'. 主題字串 '/Sport/Football'
- $T_C$ : 主題名稱 'Tennis', 叢集 'CLUSTER2'. 主題字串 '/Sport/Tennis'

在此實務範例中, 兩個叢集化主題明顯是分開的, 主題名稱或主題字串沒有重疊。

連接至 QM3 的應用程式可以建立 '/Sport/Football' 的訂閱及 '/Sport/Tennis' 的訂閱。然後, 他們會從這兩個群組中收到任何出版品。不過, 如第 30 頁的『管理主題物件』中所述, 如果他們要訂閱 '/Sport/#', 且目的是同時接收 '/Sport/Football' 和 '/Sport/Tennis' 上的發佈, 則在任一叢集中都無法將此模型辨識為叢集主題, 因此不會建立任何 Proxy 訂閱。然後, 它們會遺漏來自任一叢集中其他佇列管理程式的發佈。

如已說明, 在 CLUSTER 1 和 CLUSTER 2 中為 '/Sport/#' 建立叢集主題是無效的, 因為這些叢集主題會衝突, 且參考訊息會寫入錯誤日誌中以指出這一點。不過, 「容許」只在其中一個叢集中建立這類主題, 例如 CLUSTER 1。現在 QM3 中 '/Sport/#' 的訂閱將導致 Proxy 訂閱僅傳送至 CLUSTER 1 中的佇列管理程式, 因此仍然無法從 QM4 或 QM5 收到 '/Sport/Tennis' 的發佈。

此實務範例中的唯一解決方案是繼續建立兩個個別訂閱。

### 發佈/訂閱叢集的 REFRESH CLUSTER 考量

發出 **REFRESH CLUSTER** 指令會導致佇列管理程式暫時捨棄本端保留的叢集相關資訊, 包括任何叢集主題及其相關聯的 Proxy 訂閱。

從發出 **REFRESH CLUSTER** 指令到佇列管理程式重新取得叢集發佈/訂閱必要資訊的完整知識, 所需的時間取決於叢集的大小、可用性及完整儲存庫佇列管理程式的回應性。

在重新整理處理程序期間, 發佈/訂閱叢集裡的發佈/訂閱資料流量會中斷。對於大型叢集, 使用 **REFRESH CLUSTER** 指令可能會在叢集進行中時中斷叢集, 此後每隔 27 天, 叢集物件會自動將狀態更新傳送至所有感興趣的佇列管理程式。請參閱在大型叢集中重新整理可能影響叢集的效能及可用性。基於這些原因, 只有在「IBM 支援中心」的指引下, 才必須在發佈/訂閱叢集中使用 **REFRESH CLUSTER** 指令。

對叢集的毀壞可能在外圍出現下列症狀:

- 此佇列管理程式上叢集主題的訂閱不會從連接至叢集中其他佇列管理程式的發佈者接收發佈資訊。
- 發佈到這個佇列管理程式上叢集主題的訊息不會延伸到其他佇列管理程式上的訂閱。

- 在此期間建立的此佇列管理程式上叢集主題的訂閱，不會一致地將 Proxy 訂閱傳送至叢集的其他成員。
- 在此期間刪除的此佇列管理程式上叢集主題的訂閱，不會一直從叢集的其他成員中移除 Proxy 訂閱。
- 在訊息遞送中暫停 10 秒或更長時間。
- **MQPUT** 失敗，例如 **MQRC\_PUBLICATION\_FAILURE**。
- 因 **MQRC\_UNKNOWN\_REMOTE\_Q\_MGR** 而放置在無法傳送郵件的佇列上的發佈數

基於這些原因，在發出 **REFRESH CLUSTER** 指令之前，必須先靜止發佈/訂閱應用程式。

另請參閱 **REFRESH CLUSTER** 的使用注意事項及 [叢集作業: 使用 REFRESH CLUSTER 最佳作法](#)。

在發佈/訂閱叢集中的佇列管理程式上發出 **REFRESH CLUSTER** 指令之後，請等待所有叢集佇列管理程式及叢集主題順利重新整理，然後如第 59 頁的『重新同步化 Proxy 訂閱』中所述重新同步化 Proxy 訂閱。這種安排需要啟動從這個佇列管理程式到叢集中所有其他佇列管理程式的叢集傳送端通道。當所有 Proxy 訂閱都已正確重新同步化時，請重新啟動發佈/訂閱應用程式。

如果 **REFRESH CLUSTER** 指令需要很長時間才能完成，請查看 **SYSTEM.CLUSTER.COMMAND.QUEUE** 的 **CURDEPTH** 來監視它。

### 相關概念

執行 **REFRESH CLUSTER** 時看到的應用程式問題

[叢集作業: 使用 REFRESH CLUSTER 最佳作法](#)

### 相關參考

[MQSC 指令參照: REFRESH CLUSTER](#)

### 發佈/訂閱階層

佇列管理程式可以在階層中分組在一起，其中階層包含一或多個直接連接的佇列管理程式。佇列管理程式會使用連線時間母項與子項關係連接在一起。當第一次將兩個佇列管理程式連接在一起時，子項佇列管理程式會連接至母項佇列管理程式。

當母項和子項佇列管理程式在階層中連接時，除非您切斷佇列管理程式與階層的連線，否則它們之間沒有功能差異。

註: IBM WebSphere MQ 階層式連線需要佇列管理程式屬性 **PSMODE** 設為 **ENABLED**。

### Hierarchy

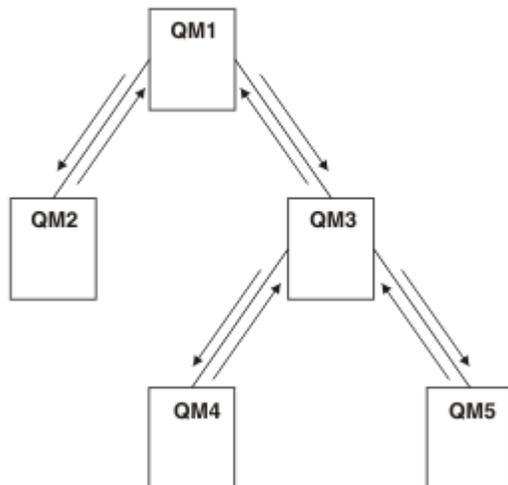


圖 34: 簡式發佈/訂閱階層

將佇列管理程式連接至分配管理系統階層

您可以將本端佇列管理程式連接至母項佇列管理程式，以修改分配管理系統階層。

## 開始之前

1. 啟用排入佇列的發佈/訂閱模式。請參閱 [啟動排入佇列的發佈/訂閱](#)。
2. 這項變更會利用 IBM WebSphere MQ 連線來延伸到母項佇列管理程式。有兩種方式可以建立連線。
  - 將佇列管理程式連接至 IBM WebSphere MQ 叢集，請參閱 [將佇列管理程式新增至叢集](#)
  - 使用與母項佇列管理程式同名的傳輸佇列或佇列管理程式別名來建立點對點通道連線。如需如何建立點對點通道連線的相關資訊，請參閱 [WebSphere MQ 分散式傳訊技術](#)。

## 關於這項作業

使用 ALTER QMGR PARENT (*PARENT\_NAME*) runmqsc 指令將子項連接至母項。

分散式發佈/訂閱是利用佇列管理程式叢集和叢集主題定義來實作。為了與 IBM WebSphere MQ Version 6.0 和 WebSphere Message Broker Version 6.1 及 WebSphere Event Broker Version 6.1 及更舊版本的交互作業能力，只要已啟用排入佇列的發佈/訂閱模式，您也可以將 Version 7.1 或更新版本的佇列管理程式連接至分配管理系統階層。

## 程序

ALTER QMGR PARENT (PARENT)

## 範例

第一個範例顯示如何將 QM2 附加為 QM1 的子項，然後查詢 QM2 的連線：

```
C:>runmqsc QM2
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM2
alter qmgr parent(QM1)
  1 : alter qmgr parent(QM1)
AMQ8005: WebSphere MQ queue manager changed.
display pubsub all
  2 : display pubsub all
AMQ8723: Display pub/sub status details.
QMNAME(QM2) TYPE(LOCAL)
STATUS(ACTIVE)
AMQ8723: Display pub/sub status details.
QMNAME(QM1) TYPE(PARENT)
STATUS(ACTIVE)
```

下一個範例顯示查詢 QM1 其連線的結果：

```
C:\Documents and Settings\Admin>runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
display pubsub all
  2 : display pubsub all
AMQ8723: Display pub/sub status details.
QMNAME(QM1) TYPE(LOCAL)
STATUS(ACTIVE)
AMQ8723: Display pub/sub status details.
QMNAME(QM2) TYPE(CHILD)
STATUS(ACTIVE)
```

## 下一步

您可以在一個分配管理系統或佇列管理程式上定義主題，這些主題可供所連接佇列管理程式上的發佈者及訂閱者使用。如需相關資訊，請參閱 [定義管理主題](#)

### 相關概念

[串流和主題](#)

[WebSphere MQ 發佈/訂閱傳訊簡介](#)

### 相關參考

[DISPLAY PUBSUB](#)

切斷佇列管理程式與分配管理系統階層的連線  
中斷子項佇列管理程式與分配管理系統階層中母項佇列管理程式的連線。

## 關於這項作業

使用 **ALTER QMGR** 指令來切斷佇列管理程式與分配管理系統階層的連線。您可以隨時依任何順序來切斷佇列管理程式的連線。

當佇列管理程式之間的連線執行時，會傳送更新母項的對應要求。

## 程序

```
ALTER QMGR PARENT('')
```

## 範例

```
C:\Documents and Settings\Admin>runmqsc QM2
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM2.
 1 : alter qmgr parent('')
AMQ8005: WebSphere MQ queue manager changed.
 2 : display pubsub type(child)
AMQ8147: WebSphere MQ object not found.
display pubsub type(parent)
 3 : display pubsub type(parent)
AMQ8147: WebSphere MQ object not found.
```

## 下一步

您可以刪除任何不再需要的串流、佇列及手動定義的通道。

發佈/訂閱階層範例: 實務 1

使用具有佇列管理程式別名的點對點通道來設定發佈/訂閱階層拓撲。

## 關於這項作業

這些實務範例以不同的方式設定發佈/訂閱階層，以建立佇列管理程式之間的連線。這些實務都使用稱為 QM1 的母項佇列管理程式，以及稱為 QM2 和 QM3 的兩個子項佇列管理程式。

實務範例 1 會分割成較小的區段，讓處理程序更容易遵循。

實務範例 1: 第 1 部分: 建立佇列管理程式

## 程序

1. 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
```

2. 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)
```

實務範例 1 第 2 部分: 點對點通道連線

## 關於這項作業

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。

## 程序

1. 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)
```

2. 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)
```

3. 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)
```

4. 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 &
```

5. 啟動下列通道：

- a. 在 QM1 上：

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

- b. 在 QM2 上：

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

- c. 在 QM3 上：

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

6. 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')
```

實務範例 1 第 3 部分: 連接佇列管理程式及定義主題

## 關於這項作業

將子佇列管理程式 QM2 及 QM3 連接至母項佇列管理程式 QM1。

## 程序

1. 在 QM2 和 QM3 上, 將母項佇列管理程式設為 QM1:

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

2. 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)
```

3. 定義主題物件:

```
define topic(FOOTBALL) TOPICSTR('Sport/Soccer')
```

實務範例 1 第 4 部分: 發佈及訂閱主題

## 關於這項作業

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

## 程序

1. 在第一個指令視窗中執行此指令:

```
amqspub Sport/Soccer QM2
```

2. 在第二個指令視窗中執行此指令:

```
amqssub Sport/Soccer QM1
```

3. 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 72 頁的『[發佈/訂閱階層範例: 實務 2](#)』

使用點對點通道 (傳輸佇列名稱與遠端佇列管理程式相同) 來設定發佈/訂閱階層拓撲。

第 75 頁的『[發佈/訂閱階層範例: 實務範例 3](#)』

使用叢集通道將佇列管理程式新增至階層拓撲。

發佈/訂閱階層範例: 實務 2

使用點對點通道 (傳輸佇列名稱與遠端佇列管理程式相同) 來設定發佈/訂閱階層拓撲。

## 關於這項作業

這些實務範例以不同的方式設定發佈/訂閱階層，以建立佇列管理程式之間的連線。這些實務都使用稱為 QM1 的母項佇列管理程式，以及稱為 QM2 和 QM3 的兩個子項佇列管理程式。

實務範例 2 分割成較小的區段，讓處理程序更容易遵循。此實務範例重複使用第 70 頁的『發佈/訂閱階層範例: 實務 1』中的實務範例 1 部分 1、實務範例 1 部分 3 及實務範例 1 部分 4。

實務範例 2 第 1 部分: 建立佇列管理程式並設定 *PSMODE*

## 程序

1. 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
```

2. 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)
```

實務範例 2 第 2 部分: 點對點通道連線

## 關於這項作業

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。

## 程序

1. 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)
```

2. 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)
```

3. 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)
```

4. 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 &
```

5. 啟動下列通道:

a. 在 QM1 上:

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

b. 在 QM2 上:

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

c. 在 QM3 上:

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

6. 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')
```

實務範例 2 第 3 部分: 連接佇列管理程式並定義主題

## 關於這項作業

將子項佇列管理程式 QM2 及 QM3 連接至母項佇列管理程式 QM1。

## 程序

1. 在 QM2 和 QM3 上, 將母項佇列管理程式設為 QM1:

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

2. 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)
```

3. 定義主題物件:

```
define topic(FOOTBALL) TOPICSTR('Sport/Soccer')
```

實務範例 2 第 4 部分: 發佈及訂閱主題

## 關於這項作業

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

## 程序

1. 在第一個指令視窗中執行此指令:

```
amqspub Sport/Soccer QM2
```

2. 在第二個指令視窗中執行此指令:

```
amqssub Sport/Soccer QM1
```

3. 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 70 頁的『[發佈/訂閱階層範例: 實務 1](#)』

使用具有佇列管理程式別名的點對點通道來設定發佈/訂閱階層拓撲。

第 75 頁的『[發佈/訂閱階層範例: 實務範例 3](#)』

使用叢集通道將佇列管理程式新增至階層拓撲。

發佈/訂閱階層範例: 實務範例 3

使用叢集通道將佇列管理程式新增至階層拓撲。

## 關於這項作業

這些實務範例以不同的方式設定發佈/訂閱階層，以建立佇列管理程式之間的連線。這些實務都使用稱為 QM1 的母項佇列管理程式，以及稱為 QM2 和 QM3 的兩個子項佇列管理程式。

實務範例 3 會分割成較小的區段，讓處理程序更容易遵循。此實務範例重複使用第 70 頁的『[發佈/訂閱階層範例: 實務 1](#)』中的實務範例 1 部分 1、實務範例 1 部分 3 及實務範例 1 部分 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。

實務範例 2 第 1 部分: 建立佇列管理程式並設定 *PSMODE*

## 程序

1. 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
```

2. 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)
```

實務範例 2 第 2 部分: 點對點通道連線

## 關於這項作業

在佇列管理程式與叢集之間建立點對點通道連線。

## 程序

1. 在 QM1 及 QM2 上，將 **REPOS** 參數設為叢集 DEMO 的名稱:

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

2. 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 &
```

3. 在每一個佇列管理程式上定義叢集接收端通道:

a. 在 QM1 上:

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

b. 在 QM2 上:

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

c. 在 QM3 上:

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

4. 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:

a. 在 QM1 上:

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

b. 在 QM2 上:

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

c. 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)
```

實務範例 2 第 3 部分: 連接佇列管理程式並定義主題

## 關於這項作業

將子項佇列管理程式 QM2 及 QM3 連接至母項佇列管理程式 QM1。

## 程序

1. 在 QM2 和 QM3 上，將母項佇列管理程式設為 QM1:

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

2. 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)
```

3. 定義主題物件:

```
define topic(FOOTBALL) TOPICSTR('Sport/Soccer')
```

實務範例 2 第 4 部分: 發佈及訂閱主題

## 關於這項作業

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

## 程序

1. 在第一個指令視窗中執行此指令:

```
amqspub Sport/Soccer QM2
```

2. 在第二個指令視窗中執行此指令:

```
amqssub Sport/Soccer QM1
```

3. 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 70 頁的『發佈/訂閱階層範例: 實務 1』

使用具有佇列管理程式別名的點對點通道來設定發佈/訂閱階層拓撲。

第 72 頁的『發佈/訂閱階層範例: 實務 2』

使用點對點通道 (傳輸佇列名稱與遠端佇列管理程式相同) 來設定發佈/訂閱階層拓撲。

### 控制發佈及訂閱的流程

連接至分散式發佈/訂閱拓撲的佇列管理程式會共用一般聯合主題空間。您可以透過選擇每一個發佈和訂閱是本端還是廣域，來控制拓撲內發佈和訂閱的流程。

本端發佈和訂閱不會延伸到發佈者或訂閱者所連接的佇列管理程式之外。

您可以控制透過將叢集或階層中的佇列管理程式連接在一起所建立的主題空間範圍。在發佈/訂閱叢集中，主題物件必須是「叢集化」，或所有元素都停留在本端，且發佈或訂閱沒有作用。

當訂閱符合不同發佈中的主題字串時，可以解析成不同的主題物件。這些稱為重疊主題。與特定相符項的發佈相關聯的主題物件會提供主題屬性，並例如決定訂閱者是否要接收發佈。

#### 發佈範圍

發佈範圍會控制佇列管理程式是否將發佈轉遞至遠端佇列管理程式。使用 **PUBSCOPE** 主題屬性來管理發佈範圍。

如果發佈未轉遞至遠端佇列管理程式，則只有本端訂閱者才會接收發佈。

**PUBSCOPE** 主題屬性用來決定對特定主題所做的發佈範圍。您可以將屬性設為下列其中一個值:

#### QMGR

發佈只會遞送給本端訂閱者。這些出版品稱為本端出版品。本端發佈不會轉遞至遠端佇列管理程式，因此連接至遠端佇列管理程式的訂閱者不會接收。

#### ALL

發佈會遞送至本端訂閱者，以及連接至遠端佇列管理程式的訂閱者。這些出版品稱為廣域出版品。

#### 如母項

使用母項的 **PUBSCOPE** 設定。

發佈者也可以使用 `MQPMO_SCOPE_QMGR` 放置訊息選項來指定發佈是本端還是廣域。如果使用此選項，它會置換已使用 **PUBSCOPE** topic 屬性設定的任何行為。

#### 訂閱範圍

訂閱的範圍可控制某個佇列管理程式上的訂閱是否接收發佈在發佈/訂閱叢集或階層中的另一個佇列管理程式上發佈的發佈，或只接收來自本端發佈者的發佈。

將訂閱範圍限制為佇列管理程式會停止將 Proxy 訂閱轉遞至發佈/訂閱拓撲中的其他佇列管理程式。這會減少佇列管理程式之間的發佈/訂閱傳訊資料流量。

**SUBSCOPE** 主題屬性用來決定對特定主題所做的訂閱範圍。您可以將屬性設為下列其中一個值:

## QMGR

訂閱只會接收本端發佈， Proxy 訂閱不會延伸到遠端佇列管理程式。

## ALL

Proxy 訂閱會延伸到遠端佇列管理程式， 且訂閱者會接收本端及遠端發佈。

## 如母項

使用母項的 **SUBSCOPE** 設定。

個別訂閱者可以在建立訂閱時指定 MQSO\_SCOPE\_QMGR 訂閱選項， 以置換 ALL 的 **SUBSCOPE** 設定。 訂閱可以置換主題的 **SUBSCOPE** 設定 ALL。

註: 個別訂閱者只能限制主題的 **SUBSCOPE**。 當個別訂閱的 **SUBSCOPE** 設定為 ALL 時， 訂閱允許使用相符主題的 **SUBSCOPE** 設定。

## 結合發佈及訂閱範圍

從 WebSphere MQ 第 7 版開始， 發佈和訂閱範圍會獨立運作， 以決定佇列管理程式之間的發佈流程。

發佈可以流向發佈/訂閱拓撲中所連接的所有佇列管理程式， 或僅流向本端佇列管理程式。 同樣地， 對於 Proxy 訂閱。 符合訂閱的發佈是由這兩個流程的組合所控管。

發佈及訂閱都可以限定為 QMGR 或 ALL。 如果發佈者和訂閱者都連接至相同的佇列管理程式， 則範圍設定不會影響訂閱者從該發佈者收到哪些發佈。

如果發佈者和訂閱者連接至不同的佇列管理程式， 則這兩個設定都必須是 ALL 才能接收遠端發佈。

假設發佈者連接至不同的佇列管理程式。 如果您想要訂閱者接收來自任何發佈者的發佈， 請將訂閱範圍設為 ALL。 然後， 您可以針對每一個發佈者， 決定是否將其發佈範圍限制為發佈者本端的訂閱者。

假設訂閱者連接至不同的佇列管理程式。 如果您想要將發佈者的發佈傳送至所有訂閱者， 請將發佈範圍設為 ALL。 如果您想要訂閱者僅從連接至相同佇列管理程式的發佈者接收發佈， 請將訂閱範圍設為 QMGR。

在第 6 版及更早版本中， 發佈及訂閱範圍不僅控管所傳送的發佈。 此外， 發佈的範圍必須符合訂閱的範圍。

## 範例 :football 結果服務

假設您是足球聯盟的成員。 每一個團隊都有一個佇列管理程式連接至發佈/訂閱叢集中的所有其他團隊。

球隊會使用主題 `Football/result/Home team name/Away team name` 來發佈所有主場比賽的結果。 以斜體表示的字串是變數主題名稱， 而發佈是符合的結果。

每一個俱樂部也會使用主題字串 `Football/myteam/Home team name/Away team name` 重新發佈僅適用於該俱樂部的結果。

這兩個主題都會發佈至整個叢集。

聯盟已設定下列訂閱， 因此任何球隊的球迷都可以使用三種有趣的方式訂閱結果。

請注意， 您可以使用 **SUBSCOPE** (QMGR) 來設定叢集主題。 主題定義會延伸到叢集的每一個成員， 但訂閱的範圍只是本端佇列管理程式。 因此， 每一個佇列管理程式的訂閱者會從相同的訂閱收到不同的發佈。

### 接收所有結果

```
DEFINE TOPIC(A) TOPICSTR('Football/result/') CLUSTER SUBSCOPE(ALL)
```

### 接收所有首頁結果

```
DEFINE TOPIC(B) TOPICSTR('Football/result/') CLUSTER SUBSCOPE(QMGR)
```

因為訂閱具有 QMGR 範圍， 所以只會比對在家中發佈的結果。

### 接收我的所有團隊結果

```
DEFINE TOPIC(C) TOPICSTR('Football/myteam/') CLUSTER SUBSCOPE(QMGR)
```

因為訂閱具有 QMGR 範圍， 所以只會比對本端重新發佈的本端團隊結果。

## 主題空間

主題空間是您可以訂閱的主題集。在分散式發佈/訂閱拓撲中連接至佇列管理程式的訂閱者具有主題空間，其中可能包括在所連接佇列管理程式上定義的主題。

當您定義主題物件或可延續訂閱時，或當應用程式動態建立發佈或訂閱時，一開始會以管理方式建立主題。

主題會透過 Proxy 訂閱，以及透過建立管理叢集主題物件，傳送至其他佇列管理程式。Proxy 訂閱會導致發佈從發佈者所連接的佇列管理程式轉遞至訂閱者的佇列管理程式。Proxy 訂閱是將不同佇列管理程式上所定義的主題結合到共同主題空間的機制。

Proxy 訂閱會在佇列管理程式階層中由上下代關係連接在一起的所有佇列管理程式之間延伸。結果是您可以在一個佇列管理程式上訂閱階層中任何其他佇列管理程式上定義的主題。只要佇列管理程式之間有連接路徑，就不會影響佇列管理程式的連接方式。

也會在叢集的所有成員之間傳送叢集主題的 Proxy 訂閱。叢集主題是連接至具有 **CLUSTER** 屬性的主題物件的主題，或從其母項繼承屬性的主題。非叢集主題的主題稱為本端主題，且不會抄寫至叢集。沒有 Proxy 訂閱會從本端主題的訂閱延伸到叢集。

總而言之，在兩種情況下會為訂閱者建立 Proxy 訂閱。

1. 佇列管理程式是階層的成員，Proxy 訂閱會轉遞至佇列管理程式的母項及子項。
2. 佇列管理程式是叢集的成員，訂閱主題字串會解析成與叢集主題物件相關聯的主題。Proxy 訂閱會轉遞至叢集的所有成員。如需併發症的相關資訊，請參閱第 86 頁的『重疊主題』。

如果佇列管理程式是叢集和階層的成員，這兩種機制會傳送 Proxy 訂閱，而不會將重複的發佈遞送給訂閱者。

建立叢集主題物件的效果是雙重的。當訂閱解析為叢集主題時，會將主題的 Proxy 訂閱傳送給叢集的其他成員。它也會將主題物件的副本傳送至叢集的其他成員。轉遞叢集主題物件的效果是簡化主題的管理。一般而言，叢集主題物件定義在叢集中的單一佇列管理程式上，稱為叢集主題主機。

下列清單中說明三個發佈/訂閱拓撲的主題空間：

- 第 79 頁的『[案例 1. 發佈/訂閱叢集](#)』。
- 第 80 頁的『[案件 2. 第 7 版中的發佈/訂閱階層](#)』。
- 第 80 頁的『[案件 3. 第 6 版中的發佈/訂閱階層和串流](#)』。

在個別主題中，下列作業說明如何結合主題空間。

- 第 81 頁的『[在發佈/訂閱叢集中建立單一主題空間](#)』。
- 第 81 頁的『[將第 7 版佇列管理程式新增至現有的第 6 版主題空間](#)』。
- 第 82 頁的『[結合多個叢集的主題空間](#)』。
- 第 83 頁的『[結合並隔離多個叢集中的主題空間](#)』。
- 第 85 頁的『[發佈及訂閱多個叢集中的主題空間](#)』。

### 案例 1. 發佈/訂閱叢集

在此範例中，假設佇列管理程式未連接至發佈/訂閱階層。

如果佇列管理程式是發佈/訂閱叢集的成員，則其主題空間是由本端主題及叢集主題組成。本端主題與不含 **CLUSTER** 屬性的主題物件相關聯。如果佇列管理程式具有本端主題物件定義，則其主題空間不同於叢集中的另一個佇列管理程式，該佇列管理程式也具有自己本端定義的主題物件。

在發佈/訂閱叢集中，除非您訂閱的主題解析為叢集主題物件，否則無法訂閱另一個佇列管理程式上定義的主題。

叢集中其他位置所定義叢集主題的衝突定義會得到解決，以支援最新的定義。在任何時間點，如果已多次定義叢集主題，則不同佇列管理程式上的叢集主題定義可能不同。

主題物件的本端定義（不論定義是用於叢集主題或本端主題）優先於叢集中其他位置所定義的相同主題物件。使用本端定義的主題，即使在其他位置定義的物件較新也一樣。

將 **PUBSCOPE** 及 **SUBSCOPE** 選項設為 **QMGR**，以防止叢集主題上的發佈或訂閱流向叢集中的不同佇列管理程式。

假設您在叢集主題主機上 Alabama 使用主題字串 USA/Alabama 來定義叢集主題物件。結果如下：

1. 叢集主題主機上的主題空間現在包括叢集主題物件 Alabama 及主題 USA/Alabama。
2. 叢集主題物件 Alabama 會抄寫到叢集中的所有佇列管理程式，在其中會結合每一個佇列管理程式上的主題空間。叢集中每一個佇列管理程式的情況取決於主題物件 Alabama 是否存在於佇列管理程式中。
  - 如果 Alabama 是新的主題物件，佇列管理程式會將叢集主題物件 Alabama 及主題 USA/Alabama 新增至其主題空間。
  - 如果 Alabama 是本端定義，則會新增叢集主題物件 Alabama。除非刪除本端定義，否則會忽略遠端定義的叢集主題物件。佇列管理程式會保留這兩個定義。
  - 如果 Alabama 是在其他位置定義的較舊叢集主題物件，則會由較新的叢集主題物件取代。
3. 應用程式或管理者可以透過參閱 Alabama 主題物件，在叢集中的任何位置建立 USA/Alabama 的訂閱。
4. 在叢集中任何位置使用主題字串 USA/Alabama 的應用程式可以直接建立訂閱來繼承主題物件 Alabama 的屬性。Alabama 主題物件是由從任何主題字串 (以 USA/Alabama 開頭) 所形成的訂閱所繼承。

如果其中一個其他佇列管理程式上有另一個 Alabama 主題物件定義，則其優先順序高於叢集主題主機上的定義。本端物件可能有叢集屬性，也可能沒有。叢集屬性可能參照相同的叢集或另一個叢集。請嘗試避免這些多重定義案例。它們會導致行為差異。

5. 如果主題物件 Alabama 具有 **PUBSCOPE** 屬性 ALL，則會將解析為 Alabama 的訂閱傳送至叢集中的所有其他佇列管理程式。

將 Alabama **PUBSCOPE** 屬性設為 QMGR，可防止發佈資訊從發佈者流向連接至叢集中不同佇列管理程式的訂閱者。

Alabama 主題物件會抄寫到叢集中的每個佇列管理程式，因此 **PUBSCOPE** 和 **PUBSCOPE** 屬性適用於叢集中的所有佇列管理程式。

叢集主題物件必須與叢集中每個位置的相同主題字串相關聯。您無法修改與主題物件相關聯的主題字串。若要將相同的主題物件與不同的主題字串相關聯，您必須刪除主題物件，並以新的主題字串重建它。如果主題已叢集化，則效果是刪除儲存在叢集其他成員上的主題物件副本，然後在叢集中到處建立新主題物件的副本。主題物件的副本都參照相同的主題字串。

不過，您可以使用不同的主題字串，在叢集中另一個佇列管理程式上建立主題物件的重複定義。一律嘗試透過管理一個佇列管理程式上的叢集主題主機來避免重複。如需此重要點的相關資訊，請參閱第 60 頁的『[多個叢集主題定義](#)』。視參照主題的方式及位置而定，相同主題物件與不同主題字串的多個定義可能會產生不同的結果。

## 案件 2. 第 7 版中的發佈/訂閱階層

在此範例中，假設佇列管理程式不是發佈/訂閱叢集的成員。

在第 7 版中，如果佇列管理程式是發佈/訂閱階層的成員，則其主題空間包含在本端及連接的佇列管理程式上定義的所有主題。階層中所有佇列管理程式的主題空間相同。沒有將主題劃分為本端主題和叢集主題。

將 **PUBSCOPE** 及 **SUBSCOPE** 選項任一設為 QMGR，以防止主題從發佈者傳送至階層中連接至不同佇列管理程式的訂閱者。

假設您在佇列管理程式 QMA 上使用主題字串 USA/Alabama 來定義主題物件 Alabama。結果如下：

1. QMA 中的主題空間現在包括主題物件 Alabama 及主題字串 USA/Alabama。
2. 應用程式或管理者可以使用主題物件名稱 Alabama 在 QMA 建立訂閱。
3. 應用程式可以在階層中的任何佇列管理程式上建立任何主題 (包括 USA/Alabama) 的訂閱。如果尚未在本端定義 QMA，則主題 USA/Alabama 會解析為主題物件 SYSTEM.BASE.TOPIC。

## 案件 3. 第 6 版中的發佈/訂閱階層和串流

在第 7 版之前，主題空間分為個別串流，其中包括存在於所有佇列管理程式上的預設串流。發佈無法在不同的串流之間流動。如果使用具名串流，則不同佇列管理程式中的主題空間可能不同。主題分為預設串流中的主題，以及不同具名串流中的主題。

**註:** 每一個具名串流都會形成個別主題空間。若要形成連接的拓撲，每一個具名串流必須存在於連接的佇列管理程式上。假設串流 X 定義在 QMA 和 QMC 上，但未定義在 QMB 上。如果 QMA 是 QMB 的母項，而 QMB 是 QMC 的母項，則串流 X 中的任何主題都無法在 QMA 與 QMC 之間流動。

同時將 **PUBSCOPE** 及 **SUBSCOPE** 選項設為 QMGR 或 ALL，需要主題的發佈者及訂閱者只交換本端使用的發佈，或只交換廣域使用的發佈。

從第 7 版開始，無法使用發佈/訂閱 API 來使用串流。如果您在第 7 版佇列管理程式上使用排入佇列的發佈/訂閱，則串流會對映至可模擬串流效果的不同主題物件。透過建立主題物件來模擬串流，該主題物件是串流中所有主題的根主題。佇列管理程式會在串流與每一個樹狀結構的對應根主題之間對映發佈和訂閱。

#### 結合主題空間

將佇列管理程式的主題空間與發佈/訂閱叢集或階層中的其他佇列管理程式相結合。結合發佈/訂閱叢集，以及發佈/訂閱叢集與階層。

您可以使用 **CLUSTER**、**PUBSCOPE** 及 **SUBSCOPE** 屬性、發佈/訂閱叢集及發佈/訂閱階層的建置區塊，來建立不同的發佈/訂閱主題空間。

從單一佇列管理程式擴增至發佈/訂閱叢集的範例開始，下列實務範例說明不同的發佈/訂閱拓撲。

#### 在發佈/訂閱叢集中建立單一主題空間

擴增發佈/訂閱系統，以在多個佇列管理程式上執行。使用發佈/訂閱叢集，為每一個發佈者和訂閱者提供單一相同的主題空間。

### 開始之前

您已在單一第 7 版佇列管理程式上實作發佈/訂閱系統。

一律使用自己的根主題來建立主題空間，而不是依賴繼承 `SYSTEM.BASE.TOPIC` 的屬性。如果您將發佈/訂閱系統擴增至叢集，則可以在叢集主題主機上將根主題定義為叢集主題，然後在整個叢集中共用所有主題。

### 關於這項作業

您現在想要擴增系統以支援更多發佈者和訂閱者，並在整個叢集中顯示每個主題。

### 程序

1. 建立叢集以與發佈/訂閱系統搭配使用。  
如果您有現有的傳統叢集，基於效能考量，最好為新的發佈訂閱系統設定新的叢集。您可以將相同的伺服器用於兩個叢集的叢集儲存庫。
2. 選擇一個佇列管理程式 (可能是其中一個儲存庫) 作為叢集主題主機。
3. 請確定在整個發佈/訂閱叢集中可見的每一個主題都解析為管理主題物件。  
設定命名發佈/訂閱叢集的 **CLUSTER** 屬性。

### 下一步

將發佈者及訂閱者應用程式連接至叢集中的任何佇列管理程式。

建立具有 **CLUSTER** 屬性的管理主題物件。這些主題也會延伸到整個叢集。發佈者和訂閱者程式使用管理主題，因此不會因為連接至叢集中的不同佇列管理程式而變更其行為。

如果您需要 `SYSTEM.BASE.TOPIC` 在每個佇列管理程式上扮演叢集主題的角色，則需要在每個佇列管理程式上修改它。

將第 7 版佇列管理程式新增至現有的第 6 版主題空間

延伸現有第 6 版發佈/訂閱系統，以與第 7 版佇列管理程式交互作業，並共用相同的主題空間。

### 開始之前

您有現有的第 6 版發佈/訂閱系統。

您已在新的伺服器上安裝 WebSphere MQ 第 7 版，並配置佇列管理程式。

## 關於這項作業

您想要延伸現有的第 6 版發佈/訂閱系統，以使用第 7 版佇列管理程式。

您已決定使用排入佇列的發佈/訂閱介面，來穩定第 6 版發佈/訂閱系統的開發。您打算使用第 7 版 MQI 將延伸規格新增至系統。您現在沒有計劃重寫已排入佇列的發佈/訂閱應用程式。

您打算在未來將第 6 版佇列管理程式升級至第 7 版。目前，您正在第 7 版佇列管理程式上繼續執行已排入佇列的現有發佈/訂閱應用程式。

## 程序

1. 建立一組傳送端-接收端通道，以雙向連接第 7 版佇列管理程式與第 6 版佇列管理程式之一。
2. 建立兩個具有目標佇列管理程式名稱的傳輸佇列。如果您因某些原因而無法使用目標佇列管理程式的名稱作為傳輸佇列名稱，請使用佇列管理程式別名。
3. 配置傳輸佇列以觸發傳送端通道。
4. 如果第 6 版發佈/訂閱系統使用串流，請依照 [新增串流](#) 中的說明，將串流新增至第 7 版佇列管理程式。
5. 檢查第 7 版佇列管理程式 **PSMODE** 是否設為 **ENABLE**。
6. 變更其 **PARENT** 屬性，以參照第 6 版佇列管理程式中的其中一個。
7. 請檢查佇列管理程式之間的上下代關係狀態是雙向作用中。

## 下一步

完成作業之後，第 6 版和第 7 版佇列管理程式會共用相同的主题空間。例如，您可以執行下列所有作業。

- 在第 6 版和第 7 版佇列管理程式之間交換發佈和訂閱。
- 在第 7 版佇列管理程式上執行現有第 6 版發佈/訂閱程式。
- 檢視及修改第 6 版或第 7 版佇列管理程式上的主题空間。
- 撰寫第 7 版發佈/訂閱應用程式，並在第 7 版佇列管理程式上執行它們。
- 使用第 7 版應用程式建立新的發佈和訂閱，並與第 6 版應用程式交換它們。

結合多個叢集的主题空間

建立跨越多個叢集的主题空間。發佈至某個叢集中的主题，並在另一個叢集中訂閱該主题。

## 開始之前

您具有現有的發佈/訂閱叢集，並且想要將部分叢集主题延伸至所有叢集。

## 關於這項作業

若要將發佈從一個叢集延伸到另一個叢集，您需要在階層中將叢集結合在一起；請參閱 [第 83 頁的圖 35](#)。階層式連線會在連接的佇列管理程式之間傳送訂閱和發佈，而叢集會在每一個叢集內傳送叢集主题，但不會在叢集之間。

這兩種機制的組合會在所有叢集之間傳播叢集主题。您需要在每一個叢集中重複叢集主题定義。

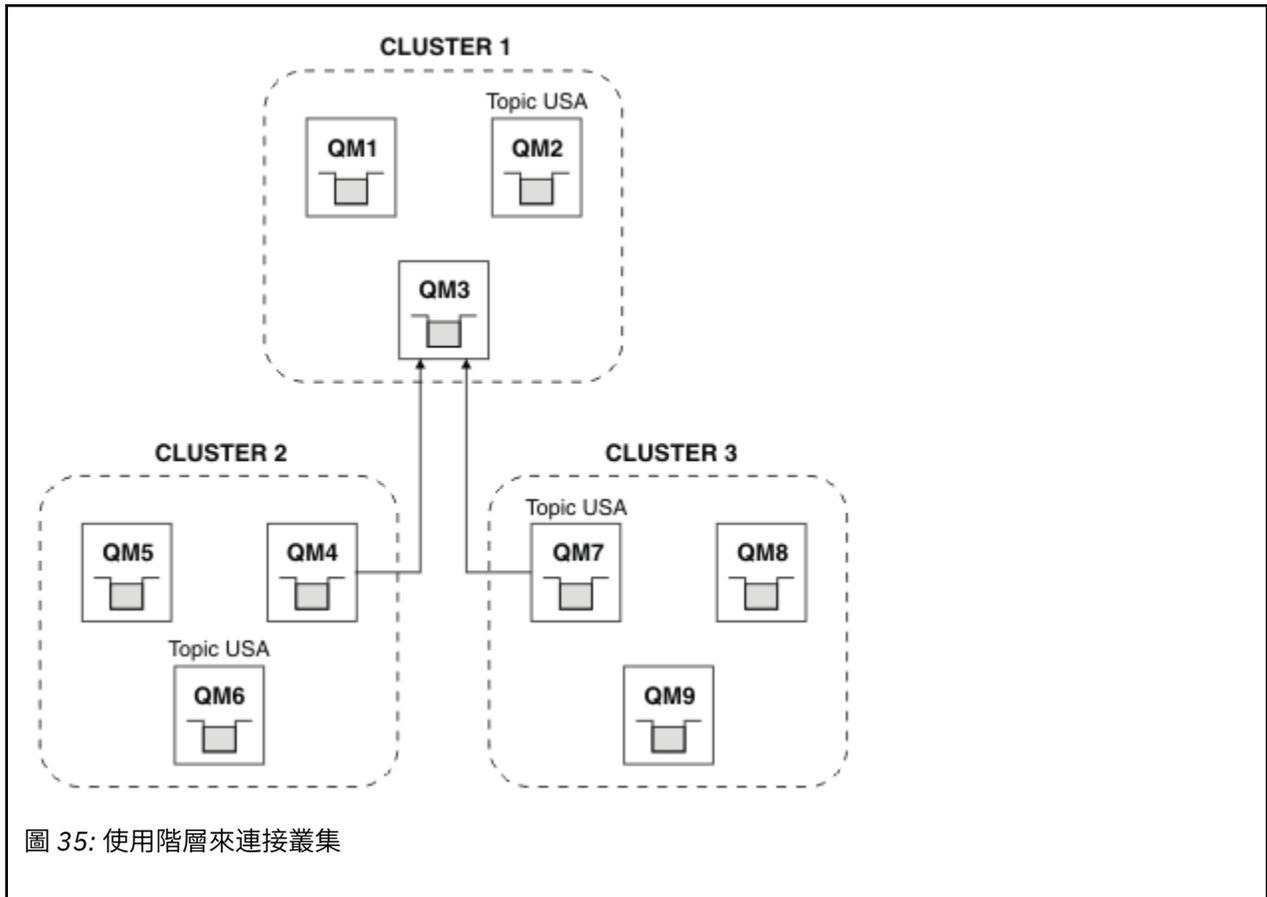


圖 35: 使用階層來連接叢集

下列步驟會將叢集連接至階層。

## 程序

1. 建立兩組傳送端-接收端通道，以雙向連接 QM3 和 QM4，以及 QM3 和 QM7。您必須使用傳統傳送端-接收端通道及傳輸佇列 (而非叢集) 來連接階層。
2. 使用目標佇列管理程式的名稱建立三個傳輸佇列。如果您因某些原因而無法使用目標佇列管理程式的名稱作為傳輸佇列名稱，請使用佇列管理程式別名。
3. 配置傳輸佇列以觸發傳送端通道。
4. 檢查 QM3、QM4 及 QM7 的 **PSMODE** 是否設為 **ENABLE**。
5. 將 QM4 和 QM7 的 **PARENT** 屬性變更為 QM3。
6. 請檢查佇列管理程式之間的上下代關係狀態是雙向作用中。
7. 在叢集 1、2 及 3 中三個叢集主題主機的每一個上建立管理主題 USA，其中包含屬性 **CLUSTER('CLUSTER 1')**、**CLUSTER('CLUSTER 2')** 及 **CLUSTER('CLUSTER 3')**。叢集主題主機不需要是階層式連接的佇列管理程式。

## 下一步

您現在可以 USA 在 [第 83 頁的圖 35](#) 中發佈或訂閱叢集主題。發佈訂閱會流向所有三個叢集中的發佈者及訂閱者。

假設您未將 USA 建立為其他叢集中的叢集主題。如果僅在 QM7 上定義 USA，則會在 QM7、QM8、QM9 及 QM3 之間交換 USA 的發佈及訂閱。在 QM7 QM8 上執行的發佈者和訂閱者 QM9 會繼承管理主題的屬性 USA。QM3 上的發佈者和訂閱者會繼承 QM3 上 **SYSTEM.BASE.TOPIC** 的屬性。

結合並隔離多個叢集中的主題空間

將部分主題空間隔離至特定叢集，並結合其他主題空間，讓它們在所有已連接叢集中都可存取。

## 開始之前

請檢查主題 第 82 頁的『結合多個叢集的主題空間』。它可能足以滿足您的需求，而不需要新增其他佇列管理程式作為橋接器。

## 關於這項作業

第 82 頁的『結合多個叢集的主題空間』中第 83 頁的圖 35 所顯示拓撲的潛在改進是隔離並非在所有叢集之間共用的叢集主題。透過建立不在任何叢集中的橋接佇列管理程式來隔離叢集，請參閱 第 84 頁的圖 36。使用橋接佇列管理程式來過濾哪些發佈和訂閱可以從一個叢集流向另一個叢集。

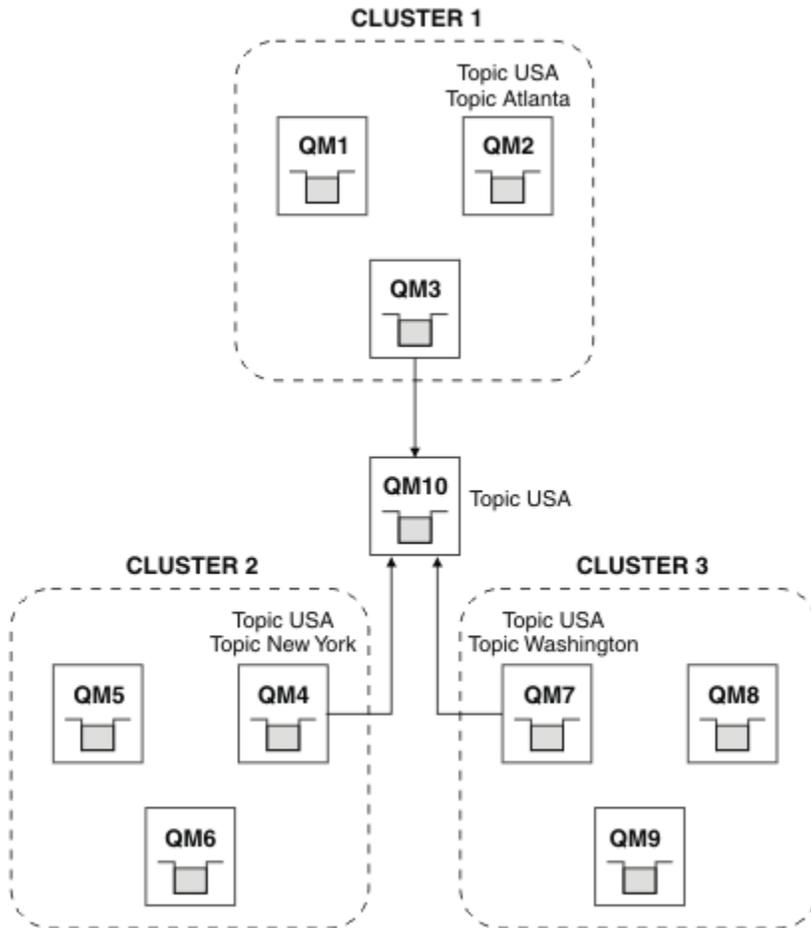


圖 36: 橋接叢集

使用橋接器來隔離您不想在其他叢集上透過橋接器公開的叢集主題。在第 84 頁的圖 36 中，USA 是所有叢集中共用的叢集主題，而 Atlanta、New York 和 Washington 是每個叢集中僅共用的叢集主題。

使用下列程序來建立配置模型：

## 程序

1. 修改所有 `SYSTEM.BASE.TOPIC` 主題物件，以在所有佇列管理程式上具有 **SUBSCOPE(QMGR)** 及 **PUBSCOPE(QMGR)**。

除非您在叢集主題的根主題上明確設定 **SUBSCOPE(ALL)** 和 **PUBSCOPE(ALL)**，否則不會將任何主題 (甚至叢集主題) 延伸到其他佇列管理程式。

2. 使用屬性 **CLUSTER(clustername)**、**SUBSCOPE(ALL)** 及 **PUBSCOPE(ALL)**，在每一個叢集中定義要共用的三個叢集主題主機上的主題。

如果您想要在所有叢集之間共用部分叢集主題，請在每一個叢集中定義相同的主題。使用每一個叢集的叢集名稱作為叢集屬性。

3. 對於您想要在所有叢集之間共用的叢集主題，請使用屬性 **SUBSCOPE(ALL)** 及 **PUBSCOPE(ALL)**，在橋接器佇列管理程式 (QM10) 上再次定義主題。

## 範例

在 [第 84 頁的圖 36](#) 的範例中，只有繼承自 USA 的主題會在所有三個叢集之間傳播。

## 下一步

在具有 **SUBSCOPE(ALL)** 和 **PUBSCOPE(ALL)** 的橋接器佇列管理程式上定義的主題訂閱，會在叢集之間傳送。

在每一個叢集內定義且具有屬性 **CLUSTER(clustername)**、**SUBSCOPE(ALL)** 及 **PUBSCOPE(ALL)** 之主題的訂閱會在每一個叢集內延伸。

任何其他訂閱都是佇列管理程式的本端訂閱。

發佈及訂閱多個叢集中的主題空間

使用重疊叢集來發佈及訂閱多個叢集中的主題。只要叢集中的主題空間不重疊，即可使用此技術。

## 開始之前

在叢集之間的交集中建立具有部分佇列管理程式的多個傳統叢集。

## 關於這項作業

基於各種不同的原因，您可能已選擇重疊叢集。

1. 您有有限數目的高可用性伺服器或佇列管理程式。您決定將所有叢集儲存庫及叢集主題主機部署至其中。
2. 您具有使用閘道佇列管理程式連接的現有傳統佇列管理程式叢集。您想要將發佈/訂閱應用程式部署至相同的叢集拓撲。
3. 您有數個自行包含的發佈/訂閱應用程式。基於效能原因，最好讓發佈/訂閱叢集保持較小且與傳統叢集分開。您已決定將應用程式部署至不同的叢集。不過，您也想要監視一個佇列管理程式上的所有發佈/訂閱應用程式，因為您只授權一個監視應用程式副本。此佇列管理程式必須有權存取所有叢集中叢集主題的發佈。

透過確保在非重疊主題空間中定義主題，您可以將主題部署至重疊發佈/訂閱叢集，請參閱 [第 85 頁的圖 37](#)。如果主題空間重疊，則部署至重疊叢集會導致問題。

因為發佈/訂閱叢集重疊，所以您可以使用重疊中的佇列管理程式來發佈及訂閱任何主題空間。

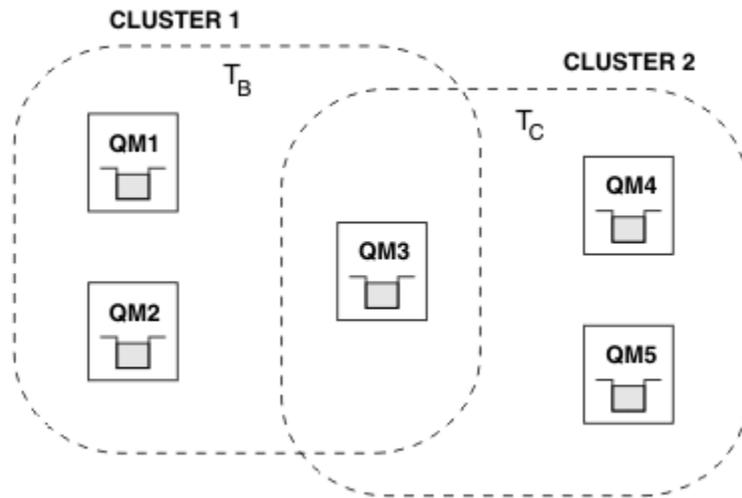


圖 37: 重疊叢集，非重疊主題空間

## 程序

建立方法來確保主題空間不會重疊。

例如，定義每一個主題空間的唯一根主題。建立根主題叢集主題。

- a) DEFINE TOPIC(B) TOPICSTR('B') CLUSTER('CLUSTER 1') ...
- b) DEFINE TOPIC(C) TOPICSTR('C') CLUSTER('CLUSTER 2') ...

## 範例

在 [第 85 頁的圖 37](#) 發佈者及連接至 QM3 的訂閱者中，可以發佈或訂閱 T<sub>B</sub> 或 T<sub>C</sub>

## 下一步

將使用這兩個叢集中的主題的發佈者和訂閱者連接至重疊中的佇列管理程式。

將必須僅使用特定叢集中主題的發佈者和訂閱者連接至不在重疊中的佇列管理程式。

### 重疊主題

根據分散式發佈/訂閱拓撲、發佈及訂閱主題字串，當發佈可以與不同的主題物件相關聯時，會發生重疊主題。

如果主題可以解析為多個主題物件，則必須考量主題之間的重疊。

### 叢集中的本端主題

主題可以定義在叢集中的任何佇列管理程式上。如果主題是在本端定義，則它優先於其他位置定義的叢集主題，並解析為相同的主題字串。

### 叢集中的叢集主題

主題可以定義在叢集中的任何佇列管理程式上。如果主題已叢集化，則會將它抄寫至叢集的其他成員。如果主題定義為叢集中另一個佇列管理程式上的叢集主題，則會發生錯誤。錯誤訊息會寫入具有現有叢集定義之佇列管理程式的錯誤日誌。

規則是只在叢集中的一個佇列管理程式（「叢集主題主機」）上定義叢集主題，以確保叢集主題只有一個定義。

如果您重新定義叢集主題，則變更會花費時間到達每一個佇列管理程式。最後，最新定義會置換先前已抄寫至非叢集主題主機的叢集主題定義。

如果您在叢集中具有不同屬性的多個佇列管理程式上定義叢集主題，則最新定義不會置換任何先前的本端定義。

## 萬用字元訂閱解析為多個主題字串

當訂閱包含萬用字元時，主題空間中可能不同的主題可以符合訂閱，並導致訂閱解析為不同的主題物件。

例如，請考量叢集 SPORTS 中的下列主題定義。

```
DEFINE TOPIC(A) TOPICSTR('Football/result/#') SUBSCOPE(QMGR) CLUSTER(SPORTS)
DEFINE TOPIC(B) TOPICSTR('Football/#') SUBSCOPE(ALL) CLUSTER(SPORTS)
DEFINE TOPIC(C) TOPICSTR('Football/result/Newport/Cardiff') PUBSCOPE(ALL) SUBSCOPE(ALL)
CLUSTER(SPORTS)
DEFINE TOPIC(D) TOPICSTR('Football/matches/Newport/Cardiff') PUBSCOPE(ALL) SUBSCOPE(QMGR)
CLUSTER(SPORTS)
```

假設叢集中有兩個佇列管理程式 QM1 及 QM2。主題 C 及 D 發佈於 QM1。

如果這些訂閱未分組，請考量 QM2 上的訂閱者所收到的內容。

- 對主題 A 的訂閱不會收到任何內容。
  - SUBSCOPE(QMGR)，且發佈位於另一個佇列管理程式上。
- 對主題 B 的訂閱會同時接收這兩個發佈。
  - 在這兩種情況下都是 SUBSCOPE(ALL) 和 PUBSCOPE(ALL)。
- 訂閱主題 C 會收到一個發佈資訊。

- SUBSCOPE (ALL) 和 PUBSCOPE (ALL), 以及與主題 C 上的發佈相符。
- 對主題 D 的訂閱不會收到任何內容。
  - SUBSCOPE (QMGR), 且發佈位於另一個佇列管理程式上。

如果這些訂閱已分組, 請考量 QM2 上的訂閱者所收到的內容。

- 訂閱者會收到一個關於主題 C 的發佈。
  - 主題 A 與 SUBSCOPE (QMGR) 的相符訂閱會被主題 C 與 SUBSCOPE (ALL) 的相符訂閱置換。更具體的訂閱獲勝, 並接收發佈。
  - 主題 B 上的相符訂閱遭到拒絕, 以支援主題 C 上的相符訂閱, 因為訂閱已分組, 且 C 更具體。已捨棄重複的發佈。
- 訂閱者未收到主題 D 的發佈資訊
  - 主題 B 與 SUBSCOPE (ALL) 的相符訂閱會被主題 D 與 SUBSCOPE (QMGR) 的相符訂閱置換。更具體的訂閱會獲勝, 且會捨棄發佈。

## 迴圈偵測如何運作

在分散式發佈/訂閱網路中, 發佈和 Proxy 訂閱不能迴圈是很重要的, 因為這會導致網路癱瘓, 連接的訂閱者會收到相同原始發佈的多個副本。

第 44 頁的『Proxy 訂閱聚集和發佈聚集』中說明的 Proxy 訂閱聚集系統不會阻止形成迴圈, 但它會防止 Proxy 訂閱的永久迴圈。因為發佈的傳播取決於 Proxy 訂閱的存在, 所以它們可以進入永久迴圈。

Websphere MQ V7.0 使用下列技術來防止出版品永久迴圈:

當發佈資訊在發佈/訂閱拓撲中移動時, 每一個佇列管理程式會將唯一指紋新增至訊息標頭。每當發佈/訂閱佇列管理程式從另一個發佈/訂閱佇列管理程式接收發佈時, 即會檢查訊息標頭中所保留的指紋。如果它自己的指紋已存在, 則發佈已在迴圈中完全循環, 因此佇列管理程式會捨棄訊息, 並將項目新增至錯誤日誌。

**註:** 在迴圈中, 會在迴圈周圍雙向傳播發佈資訊, 且迴圈中的每一個佇列管理程式會在原始佇列管理程式捨棄迴圈發佈資訊之前接收這兩個發佈資訊。這會導致訂閱應用程式接收重複的發佈副本, 直到迴圈中斷為止。

### 迴圈偵測指紋格式

迴圈偵測指紋會插入 RFH2 標頭或流程中, 作為 V7.0 通訊協定的一部分。RFH2 程式設計師需要瞭解標頭, 並完整傳遞指紋資訊。WebSphere MessageBroker 使用不包含指紋資訊的 RFH1 標頭。

```
<ibm>
  <Rfp>uuid1</Rfp>
  <Rfp>uuid2</Rfp>
  <Rfp>uuid3</Rfp>
  .
  .
  .
</ibm>
```

<ibm> is the name of the folder that holds the list of routing fingerprints containing the unique user identifier (uuid) of each queue manager that has been visited.

Every time that a message is published by a queue manager, it adds its uuid into the <ibm> folder using the <Rfp> (routing fingerprint) tag. Whenever a publication is received, WebSphere MQ uses the message properties API to iterate through the <Rfp> tags to see if that particular uuid value is present. 由於使用排入佇列的發佈/訂閱介面時, WebSphere MQ 的 WebSphere Platform Messaging 元件透過通道及 RFH2 訂閱連接至 Websphere Message Broker 的方式, WebSphere MQ 在透過該路徑接收發佈時也會建立指紋。

目標是不將任何 RFH2 遞送至應用程式 (如果它不預期), 只是因為我們已在指紋資訊中新增。

Whenever an RFH2 is converted into message properties, it will also be necessary to convert the <ibm> folder; this removes the fingerprint information from the RFH2 that is passed on or delivered to applications that have used the Websphere MQ V7.0 API.

每當具有指紋資訊的訊息遞送至 RFH1 訂閱者或傳遞至 Websphere Message Broker V6.0 時, 指紋資訊都會轉換為 RFH1。

當 Websphere Message Broker V6.0 將此訊息傳遞至 RFH2 訂閱者 (例如 SIB) 時, 它必須將指紋資訊轉換回 RFH2 格式。

JMS 應用程式不會看到指紋資訊，因為 JMS 介面不會從 RFH2 擷取該資訊，因此不會將它交給其應用程式。

Rfp 訊息內容是使用 `propDesc.CopyOptions = MQCOPY_FORWARD and MQCOPY_PUBLISH` 建立的。這會影響應用程式接收然後重新發佈相同的訊息。這表示此類應用程式可以使用 `PutMsgOpts.Action = MQACTP_FORWARD` 來繼續遞送指紋的鏈結，但必須適當地編碼以從鏈結中移除其自己的指紋。依預設，應用程式會使用 `PutMsgOpts.Action = MQACTP_NEW` 並啟動新的鏈結。

### 分散式發佈/訂閱拓撲中的保留發佈資訊

在分散式發佈/訂閱拓撲中使用保留的發佈資訊時，最佳作法是只從拓撲中的單一佇列管理程式發佈相同主題的保留發佈資訊。

否則，在相同主題的不同佇列管理程式中，可能會有不同的保留發佈資訊處於作用中，導致非預期的行為。當配送多個 Proxy 訂閱時，可能會收到多個保留的發佈。

### 佇列管理程式之間的發佈/訂閱安全

會使用一般通道安全規則，將發佈/訂閱內部訊息 (例如 Proxy 訂閱及發佈) 放置到發佈/訂閱系統佇列。本主題中的資訊和圖表強調顯示遞送這些訊息所涉及的各種程序 and 使用者 ID。

### 本端存取控制

發佈和訂閱主題的存取權由發佈/訂閱安全中說明的本端安全定義和規則控管。在 z/OS 上，不需要任何本端主題物件即可建立存取控制。其他平台上的存取控制也不需要本端主題。管理者可以選擇將存取控制套用到叢集主題物件，而不論它們是否存在於叢集中。

系統管理者負責其本端系統上的存取控制。他們必須信任階層或叢集群體的其他成員的管理者，以負責其存取控制原則。因為存取控制是針對每一個個別機器所定義，如果需要精細層次控制，則很可能是負擔。可能不需要強制任何存取控制，或可能在主題樹狀結構中的高階物件上定義存取控制。可以針對主題名稱空間的每一個子細目定義精細層次存取控制。

### 建立 Proxy 訂閱

一般通道鑑別方法會確認組織將其佇列管理程式連接至佇列管理程式的信任。如果也容許該授信組織執行分散式發佈/訂閱，則會執行權限檢查。當通道將訊息放入分散式發佈/訂閱佇列時，即會進行檢查。例如，如果將訊息放入 `SYSTEM.INTER.QMGR.CONTROL` 佇列。佇列權限檢查的使用者 ID 取決於接收端通道的 `PUTAUT` 值。例如，通道 `MCAUSER` 的使用者 ID (訊息環境定義)，視值和平台而定。如需通道安全的相關資訊，請參閱 [通道安全](#)。

使用遠端佇列管理程式上分散式發佈/訂閱代理程式的使用者 ID 進行 Proxy 訂閱。例如，第 88 頁的圖 38 中的 `QM2`。然後會輕鬆授與使用者對本端主題物件設定檔的存取權，因為該使用者 ID 已定義在系統中，因此沒有網域衝突。

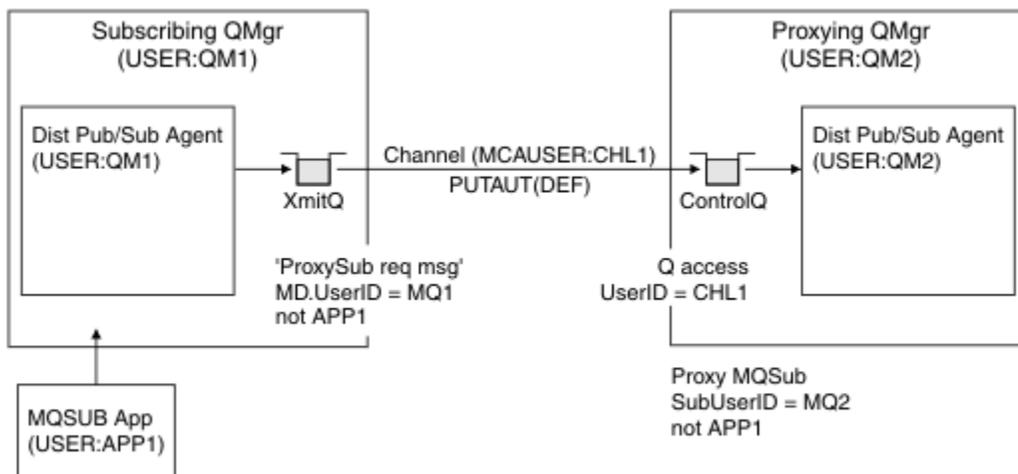


圖 38: Proxy 訂閱安全，建立訂閱

## 傳回遠端發佈

在發佈佇列管理程式上建立發佈時，會為任何 Proxy 訂閱建立發佈副本。所複製發佈的環境定義包含進行訂閱之使用者 ID 的環境定義；第 89 頁的圖 39 中的 QM2。建立 Proxy 訂閱時使用的目的地佇列是遠端佇列，因此發佈訊息會解析至傳輸佇列。

一般通道鑑別方法會確認信任組織將其佇列管理程式 QM2 連接至另一個佇列管理程式 QM1。如果接著容許該授信組織執行分散式發佈/訂閱，當通道將發佈訊息放入分散式發佈/訂閱發佈佇列 SYSTEM.INTER.QMGR.PUBS 時，會執行權限檢查。佇列權限檢查的使用者 ID 取決於接收端通道的 PUTAUT 值 (例如，通道的使用者 ID、MCAUSER、訊息環境定義及其他，視值及平台而定)。如需通道安全的相關資訊，請參閱 [通道安全](#)。

當發佈訊息到達訂閱佇列管理程式時，會在該佇列管理程式的權限下對主題執行另一個 MQPUT，且含有訊息的環境定義會取代為每一個本端訂閱者的環境定義，因為每一個本端訂閱者都有訊息。

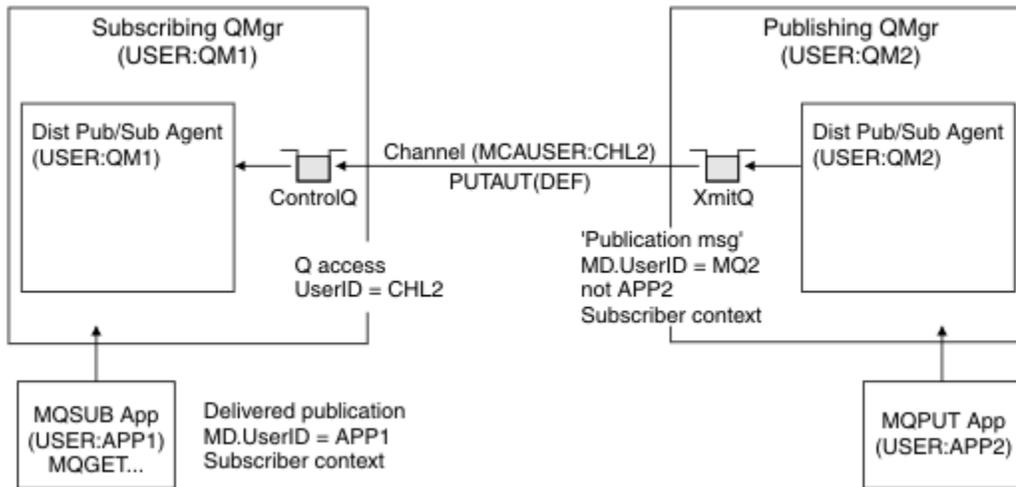


圖 39: Proxy 訂閱安全，轉遞發佈

在安全考量不多的系統上，分散式發佈/訂閱處理程序可能以 mqm 群組中的使用者 ID 執行，通道上的 MCAUSER 參數為空白 (預設值)，並視需要將訊息遞送至各種系統佇列。未受保護的系統可讓您輕鬆設定概念證明，以示範分散式發佈/訂閱。

在更認真考量安全的系統上，這些內部訊息會受到與任何通過通道的訊息相同的安全控制。

如果通道設定為非空白 MCAUSER 及指定必須勾選 MCAUSER 的 PUTAUT 值，則必須將 SYSTEM.INTER.QMGR.\* 佇列的存取權授與有問題的 MCAUSER。如果有多個不同的遠端佇列管理程式，且通道在不同的 MCAUSER ID 下執行，則需要授與所有這些使用者 ID 對 SYSTEM.INTER.QMGR.\* 佇列的存取權。例如，在單一佇列管理程式上配置多個階層式連線時，可能會出現以不同 MCAUSER ID 執行的通道。

如果通道設定了 PUTAUT 值，指定使用訊息的環境定義，則會根據內部訊息內的使用者 ID 來檢查對 SYSTEM.INTER.QMGR.\* 佇列的存取權。因為所有這些訊息都與來自傳送內部訊息或發佈訊息之佇列管理程式的分散式發佈/訂閱代理程式使用者 ID 一起放置 (請參閱第 89 頁的圖 39)，所以如果您想要以此方式設定分散式發佈/訂閱安全，則授與存取各種系統佇列的使用者 ID 集不會太大 (每個遠端佇列管理程式一個)。它仍然有通道環境定義安全一律會有的所有相同問題; 不同使用者 ID 網域的問題，以及訊息中的使用者 ID 可能未定義在接收系統上的事實。不過，如果需要的話，這是完全可以接受的執行方式。

為了分散式發佈/訂閱的目的，使用一般通道安全來執行所有佇列間管理程式傳訊。

如需在主題層次限制發佈和 Proxy 訂閱的相關資訊，請參閱 [發佈/訂閱安全](#)。

## 將預設使用者 ID 與佇列管理程式階層搭配使用

如果您具有在不同平台上執行且使用預設使用者 ID 的佇列管理程式階層，請注意，這些預設使用者 ID 在平台之間不同，且在目標平台上可能不明。因此，在一個平台上執行的佇列管理程式會拒絕從其他平台上的佇列管理程式收到的訊息，原因碼為 MQRC\_NOT\_AUTHORIZED。

為了避免拒絕訊息，至少需要將下列權限新增至其他平台上使用的預設使用者 ID：

- SYSTEM.BROKER。佇列
- \* SYSTEM.BROKER 上的 PUB \*SUB 權限。主題
- SYSTEM.BROKER.CONTROL.QUEUE 佇列。

預設使用者 ID 如下：

平台	預設使用者 ID
Windows	MUSR_MQADMIN 註: MUSR_MQADMIN 僅是第一次安裝的預設使用者 ID。對於後續安裝, 「準備 IBM WebSphere MQ 精靈」會建立名為 MUSR_MQADMINx 的使用者帳戶, 其中 x 是下一個可用的數字, 代表不存在的使用者 ID。
UNIX and Linux 系統	mqm
IBM i	QMQM
z/OS	通道起始程式位址空間使用者 ID

針對 Windows、UNIX、Linux 及 z/OS 平台上的佇列管理程式, 建立並授與存取權給 'mqm' 使用者 ID (如果以階層方式連接至 IBM i 上的佇列管理程式)。

針對 IBM i 及 z/OS 平台上的佇列管理程式, 建立並授與 'mqm' 使用者 ID 的存取權 (如果以階層方式連接至 Windows、UNIX 或 Linux 上的佇列管理程式)。

針對 Windows、UNIX、Linux 及 IBM i 平台上的佇列管理程式, 以階層方式連接至 z/OS 上的佇列管理程式, 建立並授與使用者對 z/OS 通道起始程式位址空間使用者 ID 的存取權。

使用者 ID 可以區分大小寫。原始佇列管理程式 (如果 IBM i、Windows、UNIX 或 Linux 系統) 會強制使用者 ID 全部大寫。接收端佇列管理程式 (如果 Windows、UNIX 或 Linux 系統) 會強制使用者 ID 全部小寫。因此, 在 UNIX and Linux 系統上建立的所有使用者 ID 都必須以小寫形式建立。如果已安裝訊息結束程式, 則不會強制使用者 ID 使用大寫或小寫。請小心瞭解訊息結束程式如何處理使用者 ID。

若要避免使用者 ID 轉換的潛在問題, 請執行下列動作:

- 在 UNIX、Linux 及 Windows 系統上, 確保以小寫形式指定使用者 ID。
- 在 IBM i 和 z/OS 上, 請確定以大寫指定使用者 ID。

## 分散式發佈/訂閱系統佇列

佇列管理程式會使用四個系統佇列來進行發佈/訂閱傳訊。您需要知道它們的存在, 僅用於問題判斷或產能規劃目的。

系統佇列	用途
SYSTEM.INTER.QMGR.CONTROL	WebSphere MQ 分散式發佈/訂閱控制佇列
SYSTEM.INTER.QMGR.FANREQ	WebSphere MQ 分散式發佈/訂閱內部 Proxy 訂閱扇出處理程序輸入佇列
SYSTEM.INTER.QMGR.PUBS	WebSphere MQ 分散式發佈/訂閱出版品
SYSTEM.HIERARCHY.STATE	WebSphere MQ 分散式發佈/訂閱階層關係狀態

發佈/訂閱系統佇列的屬性顯示在 第 90 頁的表 10 中。

屬性	預設值
DEFPSIST	是
DEFSOPT	EXC

表 10: 發佈/訂閱系統佇列的屬性 (繼續)

屬性	預設值
MAXMSGL	在 AIX、HP-UX、Linux、IBM i、Solaris 及 Windows 平台上: ALTER QMGR 指令的 MAXMSGL 參數值
MAXDEPTH	999999999
SHARE	無
STGCLASS	此屬性僅用於 z/OS 平台

**發佈/訂閱系統佇列錯誤**

當分散式發佈/訂閱佇列管理程式佇列無法使用時，可能會發生錯誤。

如果展開要求佇列 SYSTEM.INTER.QMGR.FANREQ 無法使用，在 Proxy 訂閱需要遞送至直接連接的佇列管理程式的情況下，MQSUB API 會收到原因碼及寫入錯誤日誌的錯誤訊息。

如果階層關係狀態佇列 SYSTEM.HIERARCHY.STATE 無法使用，錯誤訊息會寫入錯誤日誌，且發佈/訂閱引擎會進入 COMPAT 模式。

如果有任何其他 SYSTEM.INTER.QMGR 佇列無法使用，錯誤訊息會寫入錯誤日誌，而且雖然未停用功能，但發佈/訂閱訊息可能會建置在遠端佇列管理程式上的佇列上。

如果無法使用母項、子項或發佈/訂閱叢集佇列管理程式的傳輸佇列：

1. MQPUT API 會接收原因碼，且不會遞送發佈。
2. 如果達到取消臨界值，則會將接收到的佇列間管理程式發佈資訊取消至輸入佇列，然後重試放置在無法傳送的郵件佇列上。
3. 如果達到取消臨界值，Proxy 訂閱會取消至展開要求佇列，然後重試；在此情況下，Proxy 訂閱將不會遞送至任何連接的佇列管理程式。
4. 階層關係通訊協定訊息失敗，且 PUBSUB 指令上的連線狀態標示為 ERROR。

## 使用 WebSphere MQ 無法傳送郵件的佇列處理程式來處理未遞送的訊息

何謂無法傳送郵件的佇列、如何放置訊息，以及如何管理？

無法傳送郵件的佇列 (DLQ) (有時稱為 無法遞送的訊息佇列) 是無法遞送至其目的地佇列之訊息的保留佇列。網路中的每個佇列管理程式都應該有相關聯的 DLQ。

訊息可以由佇列管理程式、訊息通道代理程式 (MCA) 及應用程式放置在 DLQ 上。DLQ 上的所有訊息都必須以無法傳送郵件的標頭結構 MQDLH 作為字首。

佇列管理程式或訊息通道代理程式放置在 DLQ 上的訊息一律具有 MQDLH；將訊息放置在 DLQ 上的應用程式必須提供 MQDLH。MQDLH 結構的原因欄位包含原因碼，可識別訊息在 DLQ 上的原因。

所有 WebSphere MQ 環境都需要常式來定期處理 DLQ 上的訊息。WebSphere MQ 提供一個預設常式，稱為無法傳送郵件的佇列處理程式 (DLQ 處理程式)，您可以使用 runmqdlq 指令來呼叫它。

透過使用者撰寫的規則表格，將 DLQ 上處理訊息的指示提供給 DLQ 處理程式。也就是說，DLQ 處理程式會根據規則表格中的項目來比對 DLQ 上的訊息；當 DLQ 訊息符合規則表格中的項目時，DLQ 處理程式會執行與該項目相關聯的動作。

### 呼叫 DLQ 處理程式

使用 runmqdlq 指令來呼叫 DLQ 處理程式。您可以用兩種方式來命名您要處理的 DLQ 及您要使用的佇列管理程式。

這兩種方式如下：

- 作為命令提示字元中 `runmqdlq` 的參數。例如：

```
runmqdlq ABC1.DEAD.LETTER.QUEUE ABC1.QUEUE.MANAGER <qrule.rul
```

- 在規則表格中。例如：

```
INPUTQ(ABC1.DEAD.LETTER.QUEUE) INPUTQM(ABC1.QUEUE.MANAGER)
```

這些範例適用於稱為 `ABC1.DEAD.LETTER.QUEUE`，由佇列管理程式 `ABC1.QUEUE.MANAGER`。

如果您未如所示指定 DLQ 或佇列管理程式，則安裝的預設佇列管理程式會與屬於該佇列管理程式的 DLQ 一起使用。

`runmqdlq` 指令會從 `stdin` 取得其輸入；您可以從規則表格重新導向 `stdin`，使規則表格與 `runmqdlq` 產生關聯。

若要執行 DLQ 處理程式，您必須獲授權存取 DLQ 本身及 DLQ 上的訊息所轉遞至的任何訊息佇列。若要讓 DLQ 處理程式在訊息環境定義中具有使用者 ID 權限的佇列上放置訊息，您也必須獲得授權來假設其他使用者的身分。

如需 `runmqdlq` 指令的相關資訊，請參閱 [runmqdlq](#)。

### 範例 DLQ 處理程式 `amqsdlq`

除了使用 `runmqdlq` 指令所呼叫的 DLQ 處理程式之外，WebSphere MQ 還提供範例 DLQ 處理程式 `amqsdlq` 的來源，以及類似於 `runmqdlq` 所提供的函數。

您可以自訂 `amqsdlq`，以提供符合您需求的 DLQ 處理程式。例如，您可以決定想要一個 DLQ 處理程式，它可以處理沒有無法傳送的郵件標頭的訊息。（預設 DLQ 處理程式和範例 `amqsdlq` 都只會處理 DLQ 上以無法傳送郵件的標頭 `MQDLH` 開頭的訊息。不是以 `MQDLH` 開頭的訊息會被識別為錯誤，並無限期保留在 DLQ 上。）

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

在 WebSphere MQ for Windows 中，`amqsdlq` 的來源提供在下列目錄中：

```
MQ_INSTALLATION_PATH\tools\c\samples\dlq
```

並在下列目錄中提供已編譯版本：

```
MQ_INSTALLATION_PATH\tools\c\samples\bin
```

在 WebSphere MQ for UNIX and Linux 系統中，`amqsdlq` 的來源提供在下列目錄中：

```
MQ_INSTALLATION_PATH/samp/dlq
```

並在下列目錄中提供已編譯版本：

```
MQ_INSTALLATION_PATH/samp/bin
```

### DLQ 處理程式規則表格

DLQ 處理程式規則表格定義 DLQ 處理程式如何處理到達 DLQ 的訊息。

規則表格中有兩種類型的項目：

- 表格中的第一個項目 (選用) 包含控制資料。
- 表格中的所有其他項目都是要遵循 DLQ 處理程式的規則。每一個規則都包含符合訊息的型樣 (一組訊息性質)，以及當 DLQ 上的訊息符合指定型樣時要採取的動作。規則表格中必須至少有一個規則。

規則表格中的每一個項目都包含一個以上關鍵字。

#### 控制資料

本節說明您可以併入 DLQ 處理程式規則表格中控制資料項目的關鍵字。

註:

- 垂直線 (|) 會區隔替代方案，只能指定其中一個替代方案。
- 所有關鍵字都是選用的。

### **INPUTQ (QueueName|'\_\_\_')**

您要處理的 DLQ 名稱:

1. 您提供作為 runmqdlq 指令參數的任何 INPUTQ 值都會置換規則表格中的任何 INPUTQ 值。
2. 如果您未指定 INPUTQ 值作為 runmqdlq 指令的參數，但在規則表格中 **指定** 值，則會使用規則表格中的 INPUTQ 值。
3. 如果未指定 DLQ，或您在規則表格中指定 INPUTQ (")，則會使用屬於佇列管理程式的 DLQ 名稱，以及提供作為 runmqdlq 指令參數的名稱。
4. 如果您未將 INPUTQ 值指定為 runmqdlq 指令的參數或指定為規則表格中的值，則會使用屬於規則表格中 INPUTQM 關鍵字上所指名之佇列管理程式的 DLQ。

### **INPUTQM (QueueManagerName|'\_\_\_')**

擁有 INPUTQ 關鍵字上所指名之 DLQ 的佇列管理程式名稱:

1. 您提供作為 runmqdlq 指令參數的任何 INPUTQM 值都會置換規則表格中的任何 INPUTQM 值。
2. 如果您未指定 INPUTQM 值作為 runmqdlq 指令的參數，則會使用規則表格中的 INPUTQM 值。
3. 如果未指定佇列管理程式，或您在規則表中指定 INPUTQM (")，則會使用安裝的預設佇列管理程式。

### **RETRYINT (間隔|\_60)**

DLQ 處理程式應該重新處理 DLQ 上第一次嘗試無法處理且已要求重複嘗試的訊息的間隔 (以秒為單位)。依預設，重試間隔為 60 秒。

### **WAIT (YES| NO |nnn)**

當 DLQ 處理程式偵測到沒有可處理的進一步訊息時，DLQ 處理程式是否應該等待進一步訊息到達 DLQ。

#### **YES**

DLQ 處理程式會無限期等待。

#### **NO**

當 DLQ 處理程式偵測到 DLQ 是空的或未包含可處理的訊息時，即會結束 DLQ 處理程式。

#### **nnn**

在 DLQ 處理程式偵測到佇列是空的或沒有可處理的訊息之後，它會等待 *nnn* 秒，讓新工作在結束之前到達。

若為忙碌 DLQ，請指定 WAIT (YES)，若為活動層次低的 DLQ，請指定 WAIT (NO) 或 WAIT (*nnn*)。如果容許 DLQ 處理程式終止，請使用觸發來重新呼叫它。如需觸發的相關資訊，請參閱 [使用觸發程式來啟動 WebSphere MQ 應用程式](#)。

將控制資料併入規則表格的替代方案是提供 DLQ 及其佇列管理程式的名稱作為 runmqdlq 指令的輸入參數。如果您同時在規則表格中及作為 runmqdlq 指令的輸入指定值，則 runmqdlq 指令上指定的值優先。

如果您在規則表格中包括控制資料項目，則它必須是表格中的 **第一個** 項目。

## **規則 (型樣和動作)**

型樣相符關鍵字 (與 DLQ 上的訊息相符的關鍵字) 及動作關鍵字 (決定 DLQ 處理程式如何處理相符訊息的關鍵字) 的說明。也會提供範例規則。

## **型樣相符關鍵字**

您用來指定 DLQ 上的訊息相符值的型樣相符關鍵字如下。(所有型樣相符關鍵字都是選用的):

### **APPLIDAT (ApplIdentity 資料|\_\*)**

在 DLQ 上訊息的訊息描述子 MQMD 中指定的 *ApplIdentity* 資料值。

**APPLNAME (PutApplName|\_\*)**

發出 MQPUT 或 MQPUT1 呼叫的應用程式名稱，如 DLQ 上訊息的訊息描述子 MQMD 的 *PutAppl* 名稱欄位中所指定。

**APPLTYPE (PutAppl 類型|\_\*)**

*PutAppl* 類型值，在 DLQ 上訊息的訊息描述子 MQMD 中指定。

**DESTQ (QueueName|\_\*)**

指定訊息的訊息佇列名稱。

**DESTQM (QueueManagerName|\_\*)**

訊息目的地之訊息佇列的佇列管理程式名稱。

**回饋 (回饋|\_\*)**

當 *MsgType* 值為 MQFB\_REPORT 時，意見會說明報告的本質。

您可以使用符號名稱。例如，您可以使用符號名稱 MQFB\_COA 來識別 DLQ 上需要確認其到達目的地佇列的訊息。

**FORMAT (格式|\_\*)**

訊息傳送者用來說明訊息資料格式的名稱。

**MSGTYPE (MsgType|\_\*)**

DLQ 上訊息的訊息類型。

您可以使用符號名稱。例如，您可以使用符號名稱 MQMT\_REQUEST 來識別 DLQ 上需要回覆的那些訊息。

**持續保存 (持續性|\_\*)**

訊息的持續性值。(訊息的持續性會決定它是否在重新啟動佇列管理程式之後仍然存在。)

您可以使用符號名稱。例如，您可以使用符號名稱 MQPER\_PERSISTENT 來識別 DLQ 上持續存在的訊息。

**REASON (ReasonCode|\_\*)**

說明訊息放置到 DLQ 的原因碼。

您可以使用符號名稱。例如，您可以使用符號名稱 MQRC\_Q\_FULL 來識別放置在 DLQ 上的那些訊息，因為其目的地佇列已滿。

**REPLYQ (QueueName|\_\*)**

在 DLQ 上訊息的訊息描述子 MQMD 中指定的回覆目的地佇列名稱。

**REPLYQM (QueueManagerName|\_\*)**

回覆目的地佇列的佇列管理程式名稱，如 DLQ 上訊息的訊息描述子 MQMD 中所指定。

**USERID (UserIdentifier|\_\*)**

在 DLQ 上產生訊息之使用者的使用者 ID，如 DLQ 上訊息的訊息描述子 MQMD 中所指定。

**動作關鍵字**

用來說明如何處理相符訊息的動作關鍵字如下：

**ACTION (DISCARD | IGNORE | RETRY | FWD)**

DLQ 上任何符合此規則中定義之型樣的訊息所要採取的動作。

**捨棄**

從 DLQ 刪除訊息。

**IGNORE**

在 DLQ 上保留訊息。

**重試**

如果第一次嘗試將訊息放入其目的地佇列失敗，請重試。RETRY 關鍵字會設定嘗試實作動作的次數。控制資料的 RETRYINT 關鍵字控制兩次嘗試之間的時間。

**轉遞**

將訊息轉遞至 FWDQ 關鍵字上指定的佇列。

您必須指定 ACTION 關鍵字。

### **FWDQ (QueueName|&DESTQ|&REPLYQ)**

要求 ACTION (FWD) 時要將訊息轉遞至其中的訊息佇列名稱。

#### **QueueName**

訊息佇列的名稱。 FWDQ (") 無效。

#### **&DESTQ**

從 MQDLH 結構中的 *DestQName* 欄位取得佇列名稱。

#### **&REPLYQ**

從訊息描述子 MQMD 的 *ReplyToQ* 欄位中取得佇列名稱。

To avoid error messages when a rule specifying FWDQ (&REPLYQ) matches a message with a blank *ReplyToQ* field, specify REPLYQ (?\*) in the message pattern.

### **FWDQM (QueueManager 名稱|&DESTQM|&REPLYQM|' \_')**

要將訊息轉遞至其中之佇列的佇列管理程式。

#### **QueueManager 名稱**

要求 ACTION (FWD) 時要轉遞訊息之佇列的佇列管理程式名稱。

#### **&DESTQM**

從 MQDLH 結構中的 *DestQMGr* 名稱 欄位取得佇列管理程式名稱。

#### **&REPLYQM**

從訊息描述子 MQMD 的 *ReplyToQMGr* 欄位中取得佇列管理程式名稱。

..

FWDQM (") 是預設值，用於識別本端佇列管理程式。

### **標頭 (YES| NO)**

MQDLH 是否應該保留在要求 ACTION (FWD) 的訊息上。依預設，MQDLH 會保留在訊息上。HEADER 關鍵字對 FWD 以外的動作無效。

### **PUTAUT (DEF| CTX)**

DLQ 處理程式應用來放置訊息的權限:

#### **DEF**

放置具有 DLQ 處理程式本身權限的訊息。

#### **CTX**

將具有使用者 ID 權限的訊息放置在訊息環境定義中。如果您指定 PUTAUT (CTX)，則必須授權您採用其他使用者的身分。

### **RETRY (RetryCount| \_1)**

嘗試動作的次數，範圍為 1-999,999,999 (在控制資料的 RETRYINT 關鍵字上指定的間隔)。DLQ 處理程式嘗試實作任何特定規則的次數特定於 DLQ 處理程式的現行實例; 此計數不會在重新啟動之間持續保存。如果 DLQ 處理程式重新啟動，則嘗試套用規則的次數會重設為零。

## **範例規則**

以下是 DLQ 處理程式規則表格中的規則範例:

```
PERSIST(MQPER_PERSISTENT) REASON (MQRC_PUT_INHIBITED) +  
ACTION (RETRY) RETRY (3)
```

此規則指示 DLQ 處理程式三次嘗試將任何放置在 DLQ 上的持續訊息遞送至其目的地佇列，因為禁止 MQPUT 及 MQPUT1。

本節其餘部分會說明您可以在規則上使用的所有關鍵字。請注意下列項目:

- 關鍵字的預設值 (如果有的話) 會畫底線。對於大部分關鍵字，預設值為 \* (星號)，其符合任何值。
- 垂直線 (|) 會區隔替代方案，只能指定其中一個替代方案。
- ACTION 以外的所有關鍵字都是選用的。

## 規則表格使用慣例

DLQ 處理程式規則表格的語法、結構及內容必須遵守這些慣例。

規則表格必須遵循下列慣例：

- 規則表格必須至少包含一個規則。
- 關鍵字可以任意順序出現。
- 關鍵字只能在任何規則中併入一次。
- 關鍵字不區分大小寫。
- 關鍵字及其參數值必須與其他關鍵字至少以一個空白或逗點區隔。
- 規則的開頭或結尾，以及關鍵字、標點符號和值之間可以有任意數目的空白。
- 每一個規則必須從新行開始。
- 在 Windows 系統上，表格中的最後一個規則必須以換行字元結尾。若要達到此目的，您可以確保在規則結尾按 Enter 鍵，以便表格的最後一行是空白行。
- 基於可攜性原因，行的有效長度不得大於 72 個字元。
- 使用加號 (+) 作為行上的最後一個非空白字元，以指出規則從下一行中的第一個非空白字元繼續。使用減號 (-) 作為一行上的最後一個非空白字元，以指出規則從下一行開始繼續。在關鍵字和參數內可以出現接續字元。

例如：

```
APPLNAME('ABC+  
D')
```

產生 'ABCD'，以及

```
APPLNAME('ABC-  
D')
```

結果為 'ABC D'。

- 以星號 (\*) 開頭的註解行可以在規則表格中的任何位置出現。
- 空白行予以忽略。
- DLQ 處理程式規則表格中的每一個項目都包含一個以上關鍵字及其相關聯參數。參數必須遵循下列語法規則：
  - 每一個參數值必須至少包含一個有效字元。以引號括住的值中的定界單引號不被視為有效。例如，這些參數有效：

FORMAT('ABC')	3 個有效字元
FORMAT(ABC)	3 個有效字元
FORMAT('A')	1 個重要字元
FORMAT(A)	1 個重要字元
FORMAT(' ')	1 個重要字元

這些參數無效，因為它們不包含任何有效字元：

```
FORMAT('')  
FORMAT()  
FORMAT()  
FORMAT
```

- 萬用字元。您可以使用問號 (?) 代替任何單一字元，但尾端空白除外；您可以使用星號 (\*) 代替零或多個相鄰字元。星號 (\*) 及問號 (?) 在參數值中一律解譯為萬用字元。
- 在下列關鍵字之參數中不能包含萬用字元：ACTION、HEADER、RETRY、FWDQ、FWDQM 及 PUTAUT。
- 當執行萬用字元相符時，參數值中的尾端空白，以及 DLQ 上訊息中的對應欄位中的尾端空白並不重要。不過，字串內以單引號括住的前導和內嵌空白對萬用字元相符很重要。
- 數值參數不能包含問號 (?) 萬用字元。您可以使用星號 (\*) 來取代整個數值參數，但不能作為數值參數的一部分。例如，這些是有效的數值參數：

MSGTYPE(2)	只有回覆訊息合格
MSGTYPE(*)	任何訊息類型都適用
MSGTYPE('*')	任何訊息類型都適用

不過，MSGTYPE('2\*') 無效，因為它包含星號 (\*) 作為數值參數的一部分。

- 數值參數必須在 0-999 999 999 的範圍內。如果參數值在此範圍內，則即使它目前在關鍵字相關的欄位中無效，也會被接受。您可以對數值參數使用符號名稱。
- 如果字串值短於關鍵字相關的 MQDLH 或 MQMD 中的欄位，則該值會以空白填補欄位長度。如果值 (排除星號) 比欄位長，則會診斷錯誤。例如，這些都是 8 個字元欄位的有效字串值：

'ABCDEFGH'	8 個字元
'A*C*E*G*I'	5 個字元 (不含星號)
'*A*C*E*G*I*K*M*O*'	8 個字元 (不含星號)

- 以單引號括住包含空格、小寫字元或句點 (.)、正斜線 (?)、底線 ( ) 及百分比符號 (%) 以外的特殊字元的字串。未以單引號括住的小寫字元會轉換成大寫。如果字串包括引號，請使用兩個單引號來表示引號的開頭和結尾。計算字串長度時，每一個出現的雙引號都會計算為單一字元。

## 如何處理規則表格

DLQ 處理程式會在規則表格中搜尋型樣符合 DLQ 上訊息的規則。

搜尋會從表格中的第一個規則開始，並循序在表格中繼續執行。當 DLQ 處理程式找到具有相符型樣的規則時，它會從該規則採取動作。每當 DLQ 處理程式套用規則時，它會將該規則的重試次數增加 1。如果第一次嘗試失敗，DLQ 處理程式會重試，直到嘗試次數符合 RETRY 關鍵字上指定的數目為止。如果所有嘗試都失敗，DLQ 處理程式會搜尋表格中的下一個相符規則。

後續比對規則會重複此程序，直到動作成功為止。當每一個符合規則已嘗試其 RETRY 關鍵字上指定的次數，且所有嘗試都失敗時，會假設 ACTION (IGNORE)。如果找不到相符規則，也會假設 ACTION (IGNORE)。

註：

1. 只會針對 DLQ 上以 MQDLH 開頭的訊息來探查相符規則型樣。不是以 MQDLH 開頭的訊息會定期報告為發生錯誤，並無限期保留在 DLQ 上。
2. 所有型樣關鍵字都可以預設，以便規則只能由動作組成。不過請注意，僅動作規則會套用到佇列上具有 MQDLHs 且尚未根據表格中其他規則進行處理的所有訊息。
3. 當 DLQ 處理程式啟動時，會驗證規則表格，並在該時間標示錯誤。您可以隨時對規則表格進行變更，但這些變更要在 DLQ 處理程式重新啟動之前不會生效。
4. DLQ 處理程式不會變更訊息內容、MQDLH 或訊息描述子。DLQ 處理程式一律使用訊息選項 MQPMO\_PASS\_ALL\_CONTEXT 將訊息放入其他佇列。
5. 可能無法辨識規則表格中的連續語法錯誤，因為規則表格的設計旨在避免在驗證期間產生重複錯誤。
6. DLQ 處理程式會使用 MQOO\_INPUT\_AS\_Q\_DEF 選項開啟 DLQ。
7. 可以使用相同的規則表格，針對相同的佇列同時執行 DLQ 處理程式的多個實例。不過，DLQ 與 DLQ 處理程式之間的一對一關係更為常見。

## 確定已處理所有 DLQ 訊息

DLQ 處理程式會保留 DLQ 上已看到但未移除之所有訊息的記錄。

如果您使用 DLQ 處理程式作為過濾器，從 DLQ 擷取一小部分訊息，則 DLQ 處理程式仍必須保留 DLQ 上未處理之訊息的記錄。此外，即使 DLQ 定義為先進先出 (FIFO)，DLQ 處理程式也無法保證看到到達 DLQ 的新訊息。如果佇列不是空的，則會定期重新掃描 DLQ 以檢查所有訊息。

基於這些原因，請嘗試確保 DLQ 包含儘可能少的訊息；如果容許無法捨棄或轉遞至其他佇列的訊息 (不論原因為何) 累積在佇列上，則 DLQ 處理程式的工作量會增加，且 DLQ 本身可以填滿。

您可以採取特定措施，讓 DLQ 處理程式清空 DLQ。例如，嘗試不使用 ACTION (IGNORE)，這會在 DLQ 上留下訊息。(請記住，對於表格中其他規則未明確指出的訊息，會假設 ACTION (IGNORE)。) 相反地，對於您將忽略的那些訊息，請使用可將訊息移至另一個佇列的動作。例如：

```
ACTION (FWD) FWDQ (IGNORED.DEAD.QUEUE) HEADER (YES)
```

同樣地，將表格中的最終規則設為「擷取並更新」，以處理表格中先前規則未處理的訊息。例如，表格中的最終規則可能如下所示：

```
ACTION (FWD) FWDQ (REALLY.DEAD.QUEUE) HEADER (YES)
```

這會將符合表格中最終規則的訊息轉遞至佇列 REALLY.DEAD.QUEUE，在其中可以手動處理這些訊息。如果您沒有這類規則，則訊息可能會無限期保留在 DLQ 上。

## 範例 DLQ 處理程式規則表格

runmqdlq 指令的規則表格範例，包含單一控制資料項目及數個規則。

```
*****
*           An example rules table for the runmqdlq command           *
*****
* Control data entry
* -----
* If no queue manager name is supplied as an explicit parameter to
* runmqdlq, use the default queue manager for the machine.
* If no queue name is supplied as an explicit parameter to runmqdlq,
* use the DLQ defined for the local queue manager.
*
inputqm(' ') inputq(' ')

* Rules
* -----
* We include rules with ACTION (RETRY) first to try to
* deliver the message to the intended destination.
* If a message is placed on the DLQ because its destination
* queue is full, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_Q_FULL) ACTION(RETRY) RETRY(5)

* If a message is placed on the DLQ because of a put inhibited
* condition, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_PUT_INHIBITED) ACTION(RETRY) RETRY(5)

* The AAAA corporation are always sending messages with incorrect
* addresses. When we find a request from the AAAA corporation,
* we return it to the DLQ (DEADQ) of the reply-to queue manager
* (&REPLYQM).
* The AAAA DLQ handler attempts to redirect the message.

MSGTYPE(MQMT_REQUEST) REPLYQM(AAAA.*) +
  ACTION(FWD) FWDQ(DEADQ) FWDQM(&REPLYQM)

* The BBBB corporation never do things by half measures. If
* the queue manager BBBB.1 is unavailable, try to
* send the message to BBBB.2
```

```

DESTQM(bbbb.1) +
  action(fwd) fwdq(&DESTQ) fwdqm(bbbb.2) header(no)

* The CCCC corporation considers itself very security
* conscious, and believes that none of its messages
* will ever end up on one of our DLQs.
* Whenever we see a message from a CCCC queue manager on our
* DLQ, we send it to a special destination in the CCCC organization
* where the problem is investigated.

REPLYQM(CCCC.*) +
  ACTION(FWD) FWDQ(ALARM) FWDQM(CCCC.SYSTEM)

```

```

* Messages that are not persistent run the risk of being
* lost when a queue manager terminates. If an application
* is sending nonpersistent messages, it should be able
* to cope with the message being lost, so we can afford to
* discard the message. PERSIST(MQPER_NOT_PERSISTENT) ACTION(DISCARD)
* For performance and efficiency reasons, we like to keep
* the number of messages on the DLQ small.
* If we receive a message that has not been processed by
* an earlier rule in the table, we assume that it
* requires manual intervention to resolve the problem.
* Some problems are best solved at the node where the
* problem was detected, and others are best solved where
* the message originated. We don't have the message origin,
* but we can use the REPLYQM to identify a node that has
* some interest in this message.
* Attempt to put the message onto a manual intervention
* queue at the appropriate node. If this fails,
* put the message on the manual intervention queue at
* this node.

REPLYQM('?*') +
  ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION) FWDQM(&REPLYQM)

ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION)

```

## 多個安裝

在 UNIX, Linux, and Windows 上，系統上可以有多个 IBM WebSphere MQ 副本。

您可以選擇每一個 IBM WebSphere MQ 副本的安裝位置，但每一個副本必須位於個別安裝位置。一次最多可以在單一機器上存在 128 個 IBM WebSphere MQ 安裝。一個安裝可以是 Fix Pack 6 或更新版本的 IBM WebSphere MQ Version 7.0.1 安裝。您現在可以選擇：

- 保持在機器上維護及管理 IBM WebSphere MQ 單一安裝的簡單性。
- 利用啟用多個 IBM WebSphere MQ 安裝所提供的彈性。

在安裝多個 IBM WebSphere MQ 副本之前，您必須先做出數個決策：

- 系統上是否有 IBM WebSphere MQ Version 7.0.1 的副本？

當系統上安裝 Fix Pack 6 或更新版本的 IBM WebSphere MQ Version 7.0.1 時，有一些限制需要考量：

- 在 UNIX and Linux 系統上，IBM WebSphere MQ Version 7.0.1 必須安裝在預設位置。
- IBM WebSphere MQ Version 7.0.1 必須是系統上的第一個安裝。在安裝 7.1 版或更新版本之後，您無法安裝 IBM WebSphere MQ Version 7.0.1。如果您解除安裝 7.0.1 版，則在安裝更新版本的 WebSphere MQ 時無法重新安裝它。
- IBM WebSphere MQ Version 7.0.1 會自動成為主要安裝。安裝 IBM WebSphere MQ Version 7.0.1 時，您無法選取另一個安裝作為主要安裝。

- 您將在何處安裝 IBM WebSphere MQ 的每一個副本？

您可以選擇 7.1 版或更新版本安裝的安裝位置。如需相關資訊，請參閱 [選擇安裝位置](#)。

- 您需要主要安裝嗎？

主要安裝是指全系統位置所參照的安裝。如需相關資訊，請參閱第 100 頁的『[選擇主要安裝](#)』。

- 您的應用程式將如何連接？

您需要考量應用程式如何尋找適當的 IBM WebSphere MQ 程式庫。如需相關資訊，請參閱 [在多個安裝環境中連接應用程式](#)，以及 [在多個安裝環境中連接 .NET 應用程式](#)。

- 您現有的結束程式是否需要變更？

如果預設位置中未安裝 IBM WebSphere MQ，則需要更新您的結束程式。如需相關資訊，請參閱 [撰寫及編譯結束程式和可安裝的服務](#)。

- 哪個佇列管理程式將與哪個安裝相關聯？

每一個佇列管理程式都與特定安裝相關聯。佇列管理程式與佇列管理程式相關聯的安裝會限制該佇列管理程式，因此只能由該安裝中的指令來管理。如需相關資訊，請參閱 [建立佇列管理程式與安裝架構的關聯](#)。

- 您將如何設定環境以使用每一個安裝？

使用系統上的多個安裝，您需要考量如何使用特定安裝，以及如何從該安裝發出指令。您可以指定指令的完整路徑，也可以使用 `setmqenv` 或 `crtmqenv` 指令來設定環境變數。設定環境變數可讓您省略該安裝的指令路徑。如需相關資訊，請參閱 [setmqenv](#) 及 [crtmqenv](#)。

回答這些問題之後，您可以使用 [安裝 IBM WebSphere MQ](#) 中提供的步驟來安裝 IBM WebSphere MQ。

如果您具有 IBM WebSphere MQ 的現有安裝，並且想要使用多重安裝功能從一個 IBM WebSphere MQ 版本移轉至另一個版本，請參閱 [UNIX、Linux 及 Windows 上的多重安裝佇列管理程式共存性](#)。

## 適用於 .NET 支援套件及多個安裝架構的 IBM 訊息服務用戶端

如需多重版本支援，Java 和 .NET 傳訊及 Web 服務特性必須隨 IBM WebSphere MQ 產品一起安裝。此特性包含 *IBM Message Service Client for .NET* 支援套件 (IA9H) 中包含的所有功能。如果系統上已安裝支援套件，則不支援多個版本。在安裝 IBM WebSphere MQ 之前，您必須先解除安裝支援套件。如需安裝 .NET 特性的相關資訊，請參閱 [安裝 WebSphere MQ for .NET 類別](#)。

### 相關概念

[UNIX、Linux 及 Windows: 從 7.0.1 版並列移轉至 7.5 版](#)

[UNIX、Linux 及 Windows: 從 7.0.1 版至 7.5 版的多階段移轉](#)

### 相關工作

[配置多個安裝](#)

[在系統上尋找 WebSphere MQ 的安裝](#)

## 選擇主要安裝

在支援多個 IBM WebSphere MQ 安裝 (UNIX, Linux, and Windows) 的系統上，主要安裝是 IBM WebSphere MQ 全系統位置所參照的安裝。具有主要安裝是選用的，但方便使用。

在 IBM WebSphere MQ Version 7.1 之前，一次只能安裝一個產品實例。在 Windows 系統上，已設定數個廣域環境變數來指向該安裝。在 UNIX and Linux 系統上，符號鏈結已新增至 `/usr/lib`、`/usr/bin` 及 `/usr/include`，同時也指向該單一安裝。

從 Version 7.1 開始，您可以在 UNIX, Linux, and Windows 上安裝多個版本的 IBM WebSphere MQ。可以隨時在其中一個系統上安裝多個 IBM WebSphere MQ，並選擇性地將其中一個安裝配置為主要安裝。當存在多個版本時，指向單一安裝架構的環境變數和符號鏈結沒有那麼有意義。不過，有些功能需要這些全系統位置才能運作。例如，用於管理 IBM WebSphere MQ 及協力廠商產品的自訂使用者 Script。這些功能僅適用於主要安裝。

在 UNIX and Linux 系統上，如果您將安裝設為主要安裝，則該安裝的外部程式庫及控制指令的符號鏈結會新增至 `/usr/lib` 及 `/usr/bin`。如果您沒有主要安裝，則不會建立符號鏈結。如需主要安裝架構的符號鏈結清單，請參閱 [UNIX and Linux 上主要安裝架構的外部程式庫及控制指令鏈結](#)。

在 Windows 系統上，廣域環境變數指向已安裝主要安裝的目錄。這些環境變數是用來尋找 IBM WebSphere MQ 程式庫、控制指令及標頭檔。此外，在 Windows 系統上，作業系統的部分特性需要集中登錄介面程式庫，然後載入至單一處理程序。使用多個版本的 IBM WebSphere MQ 時，IBM WebSphere MQ 程式庫集將會發生衝突。這些特性會嘗試將這些衝突的程式庫集載入單一處理程序。因此，這類特性只能與主要安裝搭配使用。如需限制與主要安裝搭配使用之部分特性的詳細資料，請參閱 [Windows](#)。

如果您在系統上安裝 IBM WebSphere MQ Version 7.0.1，則此安裝會自動成為主要安裝。安裝 Version 7.0.1 時，無法變更主要安裝。如果系統上的所有安裝都位於 Version 7.1 或更新版本，您可以選擇是否具有主要安裝。請考量 第 101 頁的表 11 中的選項。

選項	有效的安裝配置		相關資訊
	主要	非主要	
Version 7.1 或更新版本的單一安裝。	Version 7.1 或更新版本。	無	如果您想要以與舊版相同的方式繼續使用單一安裝，請將您的安裝配置為主要安裝。如需此選項的相關資訊，請參閱 <a href="#">配置為主要安裝的 IBM WebSphere MQ Version 7.1 單一安裝或更新版本</a>
	無	Version 7.1 或更新版本。	如果您想要繼續使用單一安裝，但不想要為您建立符號鏈結或廣域環境變數，請將您的安裝配置為非主要安裝。如需此選項的含意相關資訊，請參閱 <a href="#">IBM WebSphere MQ Version 7.1 或更新版本的單一安裝，配置為非主要</a>
多個安裝: Version 7.0.1 及 Version 7.1 或更新版本。	Version 7.0.1	Version 7.1 或更新版本。	如果您想要具有多個 IBM WebSphere MQ 安裝 (其中一個是 7.0.1 版)，則 7.0.1 版安裝會自動成為主要安裝。雖然已安裝 IBM WebSphere MQ 7.0.1 版，但您無法變更哪個安裝是主要安裝。如需此選項及其含意的相關資訊，請參閱 <a href="#">多個 IBM WebSphere MQ 安裝，其中一個位於 Version 7.0.1</a>
多個安裝: Version 7.1 或更新版本。	Version 7.1 或更新版本。	Version 7.1 或更新版本。	如果您想要有多個 7.1 版或更新版本的 WebSphere MQ 安裝，您可以選擇是否將其中一個安裝設為主要。如需此選項的相關資訊，請參閱 <a href="#">多個 IBM WebSphere MQ Version 7.1 安裝或更新版本</a>
	無	Version 7.1 或更新版本。	

### 相關概念

[配置為主要安裝的 WebSphere MQ 7.1 版或更新版本的單一安裝](#)

[WebSphere MQ 7.1 版或更新版本的單一安裝，配置為非主要](#)

[WebSphere MQ 7.1 版或更新版本的多個安裝](#)

[多個 WebSphere MQ 安裝，一個 7.0.1 版](#)

### 相關工作

[變更主要安裝](#)

[選擇安裝位置](#)

[規劃安裝](#)

[選擇安裝名稱](#)

## IBM WebSphere MQ Version 7.1 或更新版本的單一安裝，配置為主要安裝

將 IBM WebSphere MQ 安裝標示為主要會將符號鏈結或廣域環境變數新增至系統，以便應用程式所使用的 IBM WebSphere MQ 指令及程式庫自動可用，並需要最少的系統設定。

您可以決定在何處安裝 IBM WebSphere MQ。

可能的話，請配置應用程式及 Script，以使用系統搜尋路徑來尋找 IBM WebSphere MQ 控制指令或 IBM WebSphere MQ 程式庫。此應用程式及 Script 配置提供最大彈性來執行未來的作業，例如移轉至下一版

IBM WebSphere MQ，或安裝第二個安裝。如需用於連接應用程式之選項的相關資訊，請參閱 [在多重安裝環境中連接應用程式](#)。

在 Windows 上，第一個安裝會自動配置為主要安裝。在 UNIX and Linux 平台上，第一次安裝到系統上必須手動配置成主要安裝。使用 **setmqinst** 指令設定主要安裝。如需相關資訊，請參閱 [解除安裝、升級及維護主要安裝](#)。

### 相關工作

[變更主要安裝](#)

[選擇安裝位置](#)

[規劃安裝](#)

[選擇安裝名稱](#)

## IBM WebSphere MQ Version 7.1 或更新版本的單一安裝，配置為非主要

如果您安裝 IBM WebSphere MQ Version 7.1 或更新版本，作為非主要的，您可能必須配置程式庫路徑，應用程式才能載入 IBM WebSphere MQ 程式庫。在 Windows 上，僅當 IBM WebSphere MQ 配置為主要時，部分產品功能才可用。

### UNIX 及 Linux 系統

在 UNIX and Linux 上執行非主要安裝的含意如下：

- 如果符合下列條件，則使用內嵌式程式庫路徑 (例如 RPATH) 尋找其 IBM WebSphere MQ 程式庫的應用程式會找不到那些程式庫：
  - IBM WebSphere MQ 安裝在與 RPATH 中指定的目錄不同的目錄中
  - /usr 中沒有符號鏈結
- 當應用程式使用外部程式庫路徑 (例如 LD\_LIBRARY\_PATH) 來尋找其程式庫時，您必須配置外部程式庫路徑以包含 `MQ_INSTALLATION_PATH/lib` 或 `MQ_INSTALLATION_PATH/lib64` 目錄。**setmqenv** 和 **crtmqenv** 指令可以在現行 Shell 中配置許多環境變數，包括外部程式庫路徑。
- 大部分 IBM WebSphere MQ 處理程序會以 `setuid/setgid` 執行。因此，當載入使用者結束程式時，它們會忽略外部程式庫路徑。只有在 IBM WebSphere MQ 程式庫內嵌的程式庫路徑中找到這些程式庫時，參照這些程式庫的使用者結束程式才能找到這些程式庫。如果 /usr 中有符號鏈結，則會解析它們。現在可以建置想要在 IBM WebSphere MQ Version 7.1 或更新版本上執行的使用者結束程式，讓它們完全不參照 IBM WebSphere MQ 程式庫。相反地，它們依賴 IBM WebSphere MQ 將函數指標傳入結束程式可以使用的 IBM WebSphere MQ 函數。如需相關資訊，請參閱 [撰寫及編譯結束程式和可安裝的服務](#)。

如需用於連接應用程式之選項的相關資訊，請參閱 [在多重安裝環境中連接應用程式](#)。

在 UNIX and Linux 平台上，系統上的第一個安裝不會自動配置為主要安裝。不過，/usr/bin 中包含單一符號鏈結，以尋找 **dspmqver** 指令。如果您不想要任何符號鏈結，則必須使用下列指令移除此鏈結：

```
setmqinst -x -p MQ_INSTALLATION_PATH
```

### Windows 系統

在 Windows 上執行非主要安裝的含意如下：

- 應用程式通常會使用外部程式庫路徑 PATH 來尋找其程式庫。沒有內嵌程式庫路徑或明確程式庫位置的概念。如果安裝是非主要安裝，則全域 PATH 環境變數不包含 IBM WebSphere MQ 安裝目錄。若要讓應用程式尋找 IBM WebSphere MQ 程式庫，請更新 PATH 環境變數以參照 IBM WebSphere MQ 安裝目錄。**setmqenv** 和 **crtmqenv** 指令可以在現行 Shell 中配置許多環境變數，包括外部程式庫路徑。
- 僅當安裝配置為主要安裝時，部分產品功能才可用；請參閱 [只能與 Windows](#)。

依預設，在 Windows 上，第一個安裝會自動配置為主要安裝。您必須手動取消選取它作為主要安裝。

### 相關工作

[變更主要安裝](#)

[選擇安裝位置](#)

[規劃安裝](#)

[選擇安裝名稱](#)

**相關參考**

[setmqenv](#)

[crtmqenv](#)

## IBM WebSphere MQ Version 7.1 或更新版本的多個安裝

您可以選擇將其中一個 IBM WebSphere MQ Version 7.1 或更新版本安裝配置為主要安裝。您的選擇取決於應用程式如何尋找程式庫。

隨附於 IBM WebSphere MQ Version 7.1 的 IBM WebSphere MQ 程式庫 (例如 mqm) 會自動使用它們所連接的佇列管理程式所需層次的程式庫。這表示只要應用程式從 IBM WebSphere MQ Version 7.1 安裝架構尋找其 IBM WebSphere MQ 程式庫，就可以連接至該系統上的任何佇列管理程式。將一個 IBM WebSphere MQ Version 7.1 安裝配置為主要，可確保如果應用程式找到其 IBM WebSphere MQ 介面程式庫，則應用程式可以連接至任何佇列管理程式。

如需在多個安裝環境中連接應用程式的相關資訊，請參閱 [在多個安裝環境中連接應用程式](#)。

當您解除安裝主要安裝時，主要安裝不會自動變更。如果您想要另一個安裝架構成為主要安裝架構，則必須使用 **setmqinst** 指令手動設定主要安裝架構。如需相關資訊，請參閱 [解除安裝、升級及維護主要安裝](#)。

**相關概念**

[多個安裝](#)

**相關工作**

[變更主要安裝](#)

[選擇安裝位置](#)

[規劃安裝](#)

[選擇安裝名稱](#)

## IBM WebSphere MQ 的多個安裝，一個位於 Version 7.0.1

IBM WebSphere MQ Version 7.1 或更新版本可以與 IBM WebSphere MQ Version 7.0.1 共存，但有一些限制。

- 在 UNIX and Linux 系統上，Version 7.0.1 只能安裝在固定預設位置，因此您無法將 Version 7.1 或更新版本安裝在該預設位置。
- IBM WebSphere MQ Version 7.0.1 會自動配置為主要安裝。在 UNIX and Linux 系統上，會自動建立指向適當 IBM WebSphere MQ 目錄的符號鏈結。在 Windows 上，產品提供的所有項目都已在全球登錄。IBM WebSphere MQ Version 7.0.1 必須以這種方式安裝才能運作。因此，在已安裝 IBM WebSphere MQ Version 7.0.1 的位置，IBM WebSphere MQ Version 7.1 或更新版本，無法將安裝設為主要。

來自 IBM WebSphere MQ Version 7.1 或更新版本的程式庫可以使用在 IBM WebSphere MQ Version 7.0.1 或更新版本下執行的任何佇列管理程式。如果應用程式需要連接至在 Version 7.0.1 及更新版本下執行的佇列管理程式，則在下列條件成立時，它可以繼續正常運作：

- 它會在執行時期尋找 IBM WebSphere MQ Version 7.1 或更新版本的程式庫。
- 它只使用 Version 7.0.1 中提供的功能。

如需在多個安裝環境中連接應用程式的相關資訊，請參閱 [在多個安裝環境中連接應用程式](#)。

當您解除安裝 IBM WebSphere MQ Version 7.0.1 時，主要安裝不會自動變更。如果您想要另一個安裝架構成為主要安裝架構，則必須使用 **setmqinst** 指令手動設定主要安裝架構。如需相關資訊，請參閱 [解除安裝、升級及維護主要安裝](#)。

**相關概念**

[多個安裝](#)

**相關工作**

[選擇安裝位置](#)

[規劃安裝](#)

## 規劃儲存體及效能需求

您必須為 IBM WebSphere MQ 系統設定實際且可達成的儲存體及效能目標。使用鏈結可找出影響平台上儲存體及效能的因素。

需求會因您使用 IBM WebSphere MQ 的系統以及您要使用的元件而有所不同。

如需受支援軟硬體環境的最新相關資訊，請參閱 [IBM WebSphere MQ 的系統需求](#) 網站：

[www.ibm.com/software/integration/wmq/requirements/](http://www.ibm.com/software/integration/wmq/requirements/)

IBM WebSphere MQ 會將佇列管理程式資料儲存在檔案系統中。使用下列鏈結，可瞭解如何規劃及配置與 IBM WebSphere MQ 搭配使用的目錄結構：

- [第 105 頁的『規劃檔案系統支援』](#)
- [第 106 頁的『共用檔案系統的需求』](#)
- [第 114 頁的『共用 IBM WebSphere MQ 檔案』](#)
- [第 116 頁的『UNIX and Linux 系統上的目錄結構』](#)
- [第 125 頁的『Windows 系統上的目錄結構』](#)

如需 UNIX and Linux 上系統資源、共用記憶體及處理程序優先順序的相關資訊，請使用下列鏈結：

- [第 128 頁的『IBM WebSphere MQ 及 UNIX System V IPC 資源』](#)
- [第 129 頁的『AIX 上的共用記憶體』](#)
- [第 129 頁的『WebSphere MQ 和 UNIX 處理程序優先順序』](#)

### 相關概念

[第 5 頁的『規劃』](#)

規劃 IBM WebSphere MQ 環境時，您必須考量您要配置的 IBM WebSphere MQ 架構、資源需求、記載需求及備份機能。使用本主題中的鏈結，以規劃 IBM WebSphere MQ 執行所在的環境。

[第 12 頁的『設計 IBM WebSphere MQ 架構』](#)

瞭解 IBM WebSphere MQ 支援點對點和發佈/訂閱傳訊樣式的不同架構。

[UNIX 及 Linux 上的軟硬體需求](#)

[Windows 上的軟硬體需求](#)

## 磁碟空間需求

WebSphere MQ 的儲存體需求取決於您安裝的元件，以及您需要的工作空間量。

您選擇安裝的選用元件需要磁碟儲存體，包括它們所需的任何必備元件。儲存體需求總計也取決於您使用的佇列數目、佇列上的訊息數目及大小，以及訊息是否持續。您也需要磁碟、磁帶或其他媒體上的保存容量，以及您自己應用程式的空間。

下表顯示在不同平台上安裝各種產品組合時所需的大約磁碟空間。(值會四捨五入至最接近的 5 MB，其中 MB 為 1,048,576 個位元組。)

平台	用戶端安裝 <sup>1</sup>	伺服器安裝 <sup>2</sup>	WebSphere MQ MFT 安裝 <sup>3</sup>	完整安裝 <sup>4</sup>
AIX	145 MB	190 MB	705 MB	915 MB
HP-UX	225 MB	310 MB	1075 MB	1340 MB
IBM i	215 MB	450 MB	80 MB	655 MB
Linux for System x (32 位元)	85 MB	N/A	N/A	120 MB

平台	用戶端安裝 <sup>1</sup>	伺服器安裝 <sup>2</sup>	WebSphere MQ MFT 安裝 <sup>3</sup>	完整安裝 <sup>4</sup>
Linux for System x (64 位元)	125 MB	170 MB	575 MB	935 MB
Linux on POWER Systems-Big Endian	130 MB	170 MB	565 MB	715 MB
Solaris x86-64、AMD64、EM64T 及相容處理器	105 MB <sup>5</sup>	150 MB <sup>5</sup>	695 MB	860 MB
Solaris SPARC	105 MB <sup>5</sup>	150 MB <sup>5</sup>	680 MB	820 MB
Windows (32 位元安裝) <sup>6</sup>	390 MB	N/A	N/A	475 MB
Windows (64 位元安裝) <sup>6</sup>	445 MB	555 MB	710 MB	1005 MB

## 使用注意事項

1. 用戶端安裝包括下列元件:

- 執行時期
- 用戶端

2. 伺服器安裝包括下列元件:

- 執行時期
- 伺服器

3. IBM WebSphere MQ Managed File Transfer 安裝包括下列元件:

- IBM WebSphere MQ Managed File Transfer 服務、日誌程式、代理程式、工具及基本元件
- 執行時期
- 伺服器
- Java
- JRE

4. 完整安裝包括所有可用的元件。

5.  在 Solaris 平台上，您必須無聲自動安裝才能取得此元件組合。

6.  並非這裡列出的所有元件都是 Windows 系統上的可安裝特性; 其功能有時包括在其他特性中。請參閱 [Windows 系統的 WebSphere MQ 特性](#)。

## 相關工作

選擇要安裝的項目

## 規劃檔案系統支援

佇列管理程式資料儲存在檔案系統中。佇列管理程式會利用檔案系統鎖定，以防止多重實例佇列管理程式的多個實例同時處於作用中。

## 共用檔案系統

共用檔案系統可讓多個系統同時存取相同的實體儲存裝置。如果多個系統直接存取相同的實體儲存裝置，而沒有強制鎖定及並行控制的方法，則會發生毀損。作業系統為本端檔案系統提供本端處理程序的鎖定及並行控制; 網路檔案系統為分散式系統提供鎖定及並行控制。

歷程上，網路檔案系統執行速度不夠快，或提供足夠的鎖定及並行控制，無法滿足記載訊息的需求。現在，網路檔案系統可以提供良好的效能，並實作可靠的網路檔案系統通訊協定 (例如 *RFC 3530*，網路檔案系統 (NFS) 第 4 版通訊協定)，以符合可靠記載訊息的需求。

## 共用檔案系統和 WebSphere MQ

多重實例佇列管理程式的佇列管理程式資料儲存在共用網路檔案系統中。在 Microsoft Windows 上，UNIX and Linux 系統上，佇列管理程式的資料檔及日誌檔必須放置在共用網路檔案系統中。

在 v7.0.1 之前，WebSphere MQ 不支援儲存在作為共用檔案系統存取之網路儲存體上的佇列管理程式資料。如果佇列管理程式資料放置在共用網路儲存體上，則您需要確保同時執行的另一個佇列管理程式實例不會存取佇列管理程式資料。

從 v7.0.1 開始，WebSphere MQ 會使用鎖定來防止相同多重實例佇列管理程式的多個實例同時處於作用中狀態。相同的鎖定也可確保兩個個別佇列管理程式不會不小心使用同一組佇列管理程式資料檔。一次只能有一個佇列管理程式實例具有其鎖定。因此，WebSphere MQ 確實支援儲存在作為共用檔案系統存取的網路儲存體上的佇列管理程式資料。

因為並非所有網路檔案系統的鎖定通訊協定都很健全，而且因為檔案系統可能配置為效能而非資料完整性，所以您必須執行 **amqmfscck** 指令，以測試網路檔案系統是否將正確控制佇列管理程式資料及日誌的存取權。此指令僅適用於 UNIX 及 IBM i 系統。在 Microsoft Windows 上，只有一個受支援的網路檔案系統，且不需要 **amqmfscck** 指令。

### 相關工作

第 107 頁的『驗證共用檔案系統行為』

執行 **amqmfscck**，以檢查 UNIX 系統上的共用檔案系統是否符合儲存多重實例佇列管理程式之佇列管理程式資料的需求。平行執行 IBM WebSphere MQ MQI client 範例程式 **amqsfhac** 與 **amqmfscck**，以示範佇列管理程式在失敗期間維護訊息完整性。

## 共用檔案系統的需求

共用檔案系統必須提供資料寫入完整性、保證檔案的專用存取權，以及在無法可靠地使用 IBM WebSphere MQ 時釋放鎖定。

### 共用檔案系統必須符合的需求

共用檔案系統必須符合三個基本需求，才能可靠地記載訊息：

#### 1. 資料寫入完整性

資料寫入完整性有時稱為清除時寫入磁碟。佇列管理程式必須能夠與順利確定至實體裝置的資料同步化。在交易式系統中，您需要確定已安全確定部分寫入，然後再繼續其他處理程序。

更具體而言，UNIX 平台上的 IBM WebSphere MQ 會使用 *O\_SYNC* 開啟選項及 *fsync()* 系統呼叫來明確地強制寫入可回復媒體，且取決於這些選項正確運作。



**小心：** Linux 您應該使用 *async* 選項來裝載檔案系統，該選項仍支援同步寫入選項，並提供比 *sync* 選項更好的效能。

不過請注意，如果檔案系統已從 Linux 匯出，您仍必須使用 **同步** 選項來匯出檔案系統。

#### 2. 保證對檔案的互斥存取

為了同步化多個佇列管理程式，需要有一種機制，讓佇列管理程式取得檔案的專用鎖定。

#### 3. 失敗時釋放鎖定

如果佇列管理程式失敗，或檔案系統發生通訊失敗，則佇列管理程式鎖定的檔案需要解除鎖定，並可供其他處理程序使用，而無需等待佇列管理程式重新連接至檔案系統。

共用檔案系統必須符合這些需求，IBM WebSphere MQ 才能可靠地運作。否則，在多重實例佇列管理程式配置中使用共用檔案系統時，佇列管理程式資料及日誌會毀損。

對於 Microsoft Windows 上的多重實例佇列管理程式，必須透過 Microsoft Windows 網路所使用的「共用網際網路檔案系統 (CIFS)」通訊協定來存取網路儲存體。「共用網際網路檔案系統 (CIFS)」用戶端不符合

IBM WebSphere MQ 在非 Microsoft Windows 平台上鎖定語意的需求，因此在非 Microsoft Windows 平台上執行的多重實例佇列管理程式不得使用「共用網際網路檔案系統 (CIFS)」作為其共用檔案系統。

對於其他受支援平台上的多重實例佇列管理程式，儲存體必須透過符合 Posix 標準且支援租賃型鎖定的網路檔案系統通訊協定來存取。現代檔案系統 (例如網路檔案系統 (NFS) 第 4 版) 使用租賃鎖定來偵測失敗，然後在失敗之後釋放鎖定。較舊的檔案系統 (例如「網路檔案系統第 3 版」) 在失敗之後沒有可靠的機制來釋放鎖定，因此不能與多重實例佇列管理程式搭配使用。

## 檢查共用檔案系統是否符合需求

您必須檢查計劃使用的共用檔案系統是否符合這些需求。您也必須檢查檔案系統是否已正確配置可靠性。共用檔案系統有時會提供配置選項，以犧牲可靠性來增進效能。

在一般情況下，IBM WebSphere MQ 會使用屬性快取正確地運作，而且不需要停用快取，例如在 NFS 裝載上設定 NOAC。當多個檔案系統用戶端競爭檔案系統伺服器上相同檔案的寫入權時，屬性快取可能會導致問題，因為每一個用戶端所使用的快取屬性可能與伺服器上的那些屬性不同。以這種方式存取的檔案範例是多重實例佇列管理程式的佇列管理程式錯誤日誌。在發生檔案輪替之前，作用中及待命佇列管理程式實例可能會寫入佇列管理程式錯誤日誌，且快取檔案屬性可能會導致錯誤日誌變得大於預期。

若要協助檢查檔案系統，請執行作業第 107 頁的『驗證共用檔案系統行為』。此作業會檢查共用檔案系統是否符合需求 2 及 3。您需要驗證共用檔案系統文件中的需求 1，或嘗試將資料記載至磁碟。

磁碟錯誤可能會導致寫入磁碟時發生錯誤，IBM WebSphere MQ 會報告為「首次失敗資料擷取」錯誤。您可以針對作業系統執行檔案系統檢查程式，以檢查共用檔案系統是否有任何磁碟錯誤。例如，在 UNIX 和 Linux 平台上，檔案系統檢查程式稱為 fsck。在 Windows 平台上，檔案系統檢查程式稱為 CHKDSK 或 SCANDISK。

## NFS 伺服器安全

註：您應該只將佇列管理程式資料放置在「網路檔案系統」(NFS) 伺服器上。在 NFS 上，搭配使用下列三個選項與裝載指令，以確保系統安全：

### noexec

透過使用此選項，您可以停止在 NFS 上執行二進位檔，以防止遠端使用者在系統上執行不想要的程式碼。

### 諾蘇伊德

透過使用此選項，您可以防止使用 set-user-identifier 及 set-group-identifier 位元，這會防止遠端使用者取得更高的專用權。

### 諾傑夫

透過使用此選項，您可以停止使用或定義字元及封鎖特殊裝置，這會防止遠端使用者從 chroot 監獄中越獄。

## 驗證共用檔案系統行為

執行 **amqmfsc**，以檢查 UNIX 系統上的共用檔案系統是否符合儲存多重實例佇列管理程式之佇列管理程式資料的需求。平行執行 IBM WebSphere MQ MQI client 範例程式 **amqsfhac** 與 **amqmfsc**，以示範佇列管理程式在失敗期間維護訊息完整性。

## 開始之前

您需要一部具有網路儲存體的伺服器，以及另外兩部已安裝 WebSphere MQ 的伺服器。您必須具有管理者 (root) 權限才能配置檔案系統，並且必須是 WebSphere MQ 管理者才能執行 **amqmfsc**。

## 關於這項作業

第 106 頁的『共用檔案系統的需求』說明將共用檔案系統與多重實例佇列管理程式搭配使用的檔案系統需求。IBM WebSphere MQ Technote [WebSphere MQ 多重實例佇列管理程式的測試及支援聲明](#) 列出 IBM 已測試過的共用檔案系統。此作業中的程序說明如何測試檔案系統，以協助您評量未列出的檔案系統是否維護資料完整性。

多重實例佇列管理程式的失效接手可以由硬體或軟體故障觸發，包括網路問題，這些問題會阻止佇列管理程式寫入其資料或日誌檔。主要而言，您有興趣在檔案伺服器上造成失敗。但是您也必須導致 IBM

WebSphere MQ 伺服器失敗，以測試是否已順利釋放任何鎖定。若要對共用檔案系統有信心，請測試下列所有失敗，以及您環境特有的任何其他失敗：

1. 正在關閉檔案伺服器上的作業系統，包括同步磁碟。
2. 中止檔案伺服器上的作業系統，而不同步磁碟。
3. 按下每一部伺服器上的重設按鈕。
4. 從每一個伺服器中拉出網路纜線。
5. 從每一個伺服器中拉出電源線。
6. 關閉每一個伺服器。

在您要用來共用佇列管理程式資料及日誌的網路儲存體上建立目錄。目錄擁有者必須是 WebSphere MQ 管理者，換句話說，是 UNIX 上 mqm 群組的成員。執行測試的使用者必須具備 WebSphere MQ 管理者權限。

使用在 Linux 上建立多重實例佇列管理程式來協助您配置檔案系統。不同的檔案系統需要不同的配置步驟。請閱讀檔案系統文件。

## 程序

在每一個檢查中，當檔案系統檢查程式執行時，會導致前一個清單中的所有失敗。如果您想要與 **amqmfsc** 同時執行 **amqsfhac**，請與此作業平行執行作業第 112 頁的『執行 amqsfhac 以測試訊息完整性』。

1. 在兩部 IBM WebSphere MQ 伺服器上裝載匯出的目錄。

在檔案系統伺服器上，建立共用目錄 `shared` 及子目錄，以儲存多重實例佇列管理程式 `qmdata` 的資料。如需在 Linux 上設定多重實例佇列管理程式的共用目錄範例，請參閱在 Linux 上建立多重實例佇列管理程式中的範例。

2. 請檢查基本檔案系統行為。

在一部 IBM WebSphere MQ 伺服器上，執行不含參數的檔案系統檢查程式。

```
amqmfsc /shared/qmdata
```

圖 40: 在 IBM WebSphere MQ 伺服器 1 上

3. 請檢查同時從兩部 IBM WebSphere MQ 伺服器寫入相同目錄的情況。

在兩部 IBM WebSphere MQ 伺服器上，使用 `-c` 選項同時執行檔案系統檢查程式。

```
amqmfsc -c /shared/qmdata
```

圖 41: 在 IBM WebSphere MQ 伺服器 1 上

```
amqmfsc -c /shared/qmdata
```

圖 42: 在 IBM WebSphere MQ 伺服器 2 上

4. 檢查等待及釋放兩個 IBM WebSphere MQ 伺服器上的鎖定。

在兩個 IBM WebSphere MQ 伺服器上，使用 `-w` 選項同時執行檔案系統檢查程式。

```
amqmfsc -w /shared/qmdata
```

圖 43: 在 IBM WebSphere MQ 伺服器 1 上

```
amqmfscck -w /shared/qmdata
```

圖 44: 在 IBM WebSphere MQ 伺服器 2 上

5. 檢查資料完整性。

a) 格式化測試檔案。

在要測試的目錄中建立大型檔案。檔案已格式化，因此後續階段可以順利完成。檔案必須夠大，才能有足夠時間岔斷第二個階段來模擬失效接手。請嘗試預設值 262144 頁 (1 GB)。程式會在慢速檔案系統上自動減少此預設值，以便在大約 60 秒內完成格式化

```
amqmfscck -f /shared/qmdata
```

伺服器會回應下列訊息：

```
Formatting test file for data integrity test.  
Test file formatted with 262144 pages of data.
```

圖 45: 在 IBM WebSphere MQ 伺服器 1 上

b) 導致失敗時，使用檔案系統檢查程式將資料寫入測試檔案。

同時在兩部伺服器上執行測試程式。在即將發生失敗的伺服器上啟動測試程式，然後在即將在失敗中存活的伺服器上啟動測試程式。造成您正在調查的失敗。

第一個測試程式停止，並出現錯誤訊息。第二測試程式取得測試檔案上的鎖定，並從第一測試程式離開的位置開始將資料寫入測試檔案。讓第二個測試程式執行完成。

表 12: 同時在兩部伺服器上執行資料完整性檢查

IBM WebSphere MQ 伺服器 1	IBM WebSphere MQ 伺服器 2
<pre>amqmfscck -a /shared/qmdata</pre>	

表 12: 同時在兩部伺服器上執行資料完整性檢查 (繼續)

IBM WebSphere MQ 伺服器 1	IBM WebSphere MQ 伺服器 2
<p>Please start this program on a second machine with the same parameters. File lock acquired. Start a second copy of this program with the same parameters on another server.</p> <p>Writing data into test file.</p> <p>To increase the effectiveness of the test, interrupt the writing by ending the process, temporarily breaking the network connection to the networked storage, rebooting the server or turning off the power. To increase the effectiveness of the test, interrupt the writing by ending the process, temporarily breaking the network connection to the networked storage, rebooting the server or turning off the power.</p>	<pre>amqmfscck -a /shared/qmdata</pre> <pre>Waiting for lock... Waiting for lock...</pre>
<p>Turn the power off here</p>	
	<pre>File lock acquired. Reading test file Checking the integrity of the data read. Appending data into the test file after data already found. The test file is full of data. It is ready to be inspected for data integrity.</pre>

測試的計時取決於檔案系統的行為。例如，檔案系統通常需要 30-90 秒，才能在停電之後釋放第一個程式所取得的檔案鎖定。如果在第一個測試程式填入檔案之前，您沒有足夠時間建立失敗，請使用 **amqmfscck** 的 **-x** 選項來刪除測試檔案。請從一開始使用較大的測試檔案來嘗試測試。

c) 驗證測試檔案中資料的完整性。

```
amqmfscck -i /shared/qmdata
```

伺服器會回應下列訊息:

```
File lock acquired
Reading test file checking the integrity of the data read.
The data read was consistent.
The tests on the directory completed successfully.
```

圖 46: 在 IBM WebSphere MQ 伺服器 2 上

6. 刪除測試檔案。

```
amqmfscck -x /shared/qmdata
Test files deleted.
```

圖 47: 在 IBM WebSphere MQ 伺服器 2 上

伺服器會回應下列訊息:

```
Test files deleted.
```

## 結果

如果測試順利完成，程式會傳回結束碼零，否則會傳回非零。

## 範例

第一組三個範例顯示產生最小輸出的指令。

### 成功測試一部伺服器上的基本檔案鎖定

```
> amqmfscck /shared/qmdata
The tests on the directory completed successfully.
```

### 在一部伺服器上測試基本檔案鎖定失敗

```
> amqmfscck /shared/qmdata
AMQ6245: Error Calling 'write()[2]' on file '/shared/qmdata/amqmfscck.lck' error '2'.
```

### 成功測試兩部伺服器上的鎖定

IBM WebSphere MQ 伺服器 1	IBM WebSphere MQ 伺服器 2
<pre>&gt; amqmfscck -w /shared/qmdata Please start this program on a second machine with the same parameters. Lock acquired. Press Return or terminate the program to release the lock.</pre>	
	<pre>&gt; amqmfscck -w /shared/qmdata Waiting for lock...</pre>
<pre>[ Return pressed ] Lock released.</pre>	
	<pre>Lock acquired. The tests on the directory completed successfully</pre>

三個範例的第二組顯示使用詳細模式的相同指令。

### 成功測試一部伺服器上的基本檔案鎖定

```
> amqmfscck -v /shared/qmdata
System call: stat("/shared/qmdata")
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call: fchmod(fd, 0666)
System call: fstat(fd)
System call:fcntl(fd, F_SETLK, F_WRLCK)
System call: write(fd)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call:fcntl(fd, F_SETLK, F_WRLCK)
System call: close(fd)
System call: fd1 = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call:fcntl(fd1, F_SETLK, F_RDLOCK)
System call: fd2 = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call:fcntl(fd2, F_SETLK, F_RDLOCK)
System call: close(fd2)
System call: write(fd1)
```

```
System call: close(fd1)
The tests on the directory completed successfully.
```

### 在一部伺服器上測試基本檔案鎖定失敗

```
> amqmfscck -v /shared/qmdata
System call: stat("/shared/qmdata")
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call: fchmod(fd, 0666)
System call: fstat(fd)
System call:fcntl(fd, F_SETLK, F_WRLCK)
System call: write(fd)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call:fcntl(fd, F_SETLK, F_WRLCK)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call:fcntl(fd, F_SETLK, F_RDLCK)
System call: fdSameFile = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call:fcntl(fdSameFile, F_SETLK, F_RDLCK)
System call: close(fdSameFile)
System call: write(fd)
AMQxxxx: Error calling 'write()[2]' on file '/shared/qmdata/amqmfscck.lck', errno 2
(Permission denied).
```

### 成功測試兩部伺服器上的鎖定

表 14: 在兩部伺服器上順利鎖定-詳細模式	
IBM WebSphere MQ 伺服器 1	IBM WebSphere MQ 伺服器 2
<pre>&gt; amqmfscck -wv /shared/qmdata Calling 'stat("/shared/qmdata")' Calling 'fd = open("/shared/qmdata/ amqmfscck.lkw", O_EXCL   O_CREAT   O_RDWR, 0666)' Calling 'fchmod(fd, 0666)' Calling 'fstat(fd)' Please start this program on a second machine with the same parameters. Calling 'fcntl(fd, F_SETLK, F_WRLCK)' Lock acquired. Press Return or terminate the program to release the lock.</pre>	
	<pre>&gt; amqmfscck -wv /shared/qmdata Calling 'stat("/shared/qmdata")' Calling 'fd = open("/shared/qmdata/ amqmfscck.lkw", O_EXCL   O_CREAT   O_RDWR,0666)' Calling 'fd = open("/shared/qmdata/amqmfscck.lkw, O_RDWR, 0666)' Calling 'fcntl(fd, F_SETLK, F_WRLCK) 'Waiting for lock...</pre>
<pre>[ Return pressed ] Calling 'close(fd)' Lock released.</pre>	
	<pre>Calling 'fcntl(fd, F_SETLK, F_WRLCK)' Lock acquired. The tests on the directory completed successfully</pre>

### 相關參考

[amqmfscck \(檔案系統檢查\)](#)

執行 **amqsfhac** 以測試訊息完整性

**amqsfhac** 會檢查使用網路儲存體的佇列管理程式是否在失敗之後維護資料完整性。

## 開始之前

此測試需要四部伺服器。兩部伺服器用於多重實例佇列管理程式，一部用於檔案系統，另一部用於作為 IBM WebSphere MQ MQI client 應用程式執行 **amqsfhac**。

遵循 [程序](#) 中的步驟 [第 108 頁的『1』](#)，以設定多重實例佇列管理程式的檔案系統。

## 關於這項作業

### 程序

1. 使用您在 [程序中](#) 步驟 [第 108 頁的『1』](#) 所建立的檔案系統，在另一部伺服器 QM1 上建立多重實例佇列管理程式。

請參閱 [建立多重實例佇列管理程式](#)。

2. 在兩部伺服器上啟動佇列管理程式，使其具有高可用性。

在伺服器 1 上:

```
strmqm -x QM1
```

在伺服器 2 上:

```
strmqm -x QM1
```

3. 設定用戶端連線以執行 **amqsfhac**。

- a) 使用 [驗證用戶端安裝](#) 中的程序來設定用戶端連線，或使用 [可重新連接的用戶端範例](#) 中的範例 Script。
- b) 修改用戶端通道以具有兩個 IP 位址，對應於執行 QM1 的兩部伺服器。

在範例 Script 中，修改:

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +  
CONNAME('LOCALHOST(2345)') QMNAME(QM1) REPLACE
```

至:

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +  
CONNAME('server1(2345),server2(2345)') QMNAME(QM1) REPLACE
```

其中 **server1** 和 **server2** 是兩部伺服器的主機名稱，2345 是通道接聽器接聽所在的埠。這通常預設為 1414。您可以將 1414 與預設接聽器配置搭配使用。

4. 在 QM1 上建立兩個本端佇列，以進行測試。  
執行下列 MQSC Script:

```
DEFINE QLOCAL(TARGETQ) REPLACE  
DEFINE QLOCAL(SIDEQ) REPLACE
```

5. 使用 **amqsfhac** 測試配置

```
amqsfhac QM1 TARGETQ SIDEQ 2 2 2
```

6. 測試檔案系統完整性時測試訊息完整性。

在 [程序的](#) 步驟 [第 109 頁的『5』](#) 期間執行 **amqsfhac**。

```
amqsfhac QM1 TARGETQ SIDEQ 10 20 0
```

如果您停止作用中佇列管理程式實例，「**amqsfhac**」會在變成作用中之後重新連接至另一個佇列管理程式實例。再次重新啟動已停止的佇列管理程式實例，以便您可以在下一次測試中反轉失敗。您可能需要根據您環境的實驗來增加反覆運算次數，以便測試程式執行足夠時間來進行失效接手。

## 結果

第 114 頁的圖 48 中顯示在步驟 第 113 頁的『6』中執行 **amqsfhac** 的範例。測試是成功的。

如果測試偵測到問題，輸出會報告失敗。在某些測試執行中，MQRC\_CALL\_INTERRUPTED 可能會報告 "Resolving to backed out"。這對結果沒有區別。結果取決於網路檔案儲存體是否在失敗發生之前或之後確定寫入磁碟。

```
Sample AMQSFHAC start
qmname = QM1
qname = TARGETQ
sidename = SIDEQ
transize = 10
iterations = 20
verbose = 0
Iteration 0
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Iteration 6
Resolving MQRC_CALL_INTERRUPTED
MQGET browse side tranid=14 pSideinfo->tranid=14
Resolving to committed
Iteration 7
Iteration 8
Iteration 9
Iteration 10
Iteration 11
Iteration 12
Iteration 13
Iteration 14
Iteration 15
Iteration 16
Iteration 17
Iteration 18
Iteration 19
Sample AMQSFHAC end
```

圖 48: 成功執行 **amqsfhac** 的輸出

## 相關參考

[高可用性範例程式](#)

## 共用 IBM WebSphere MQ 檔案

部分 IBM WebSphere MQ 檔案由作用中佇列管理程式專用存取，其他檔案則共用。

WebSphere MQ 檔案會分割成程式檔案和資料檔。程式檔案通常安裝在本端執行 WebSphere MQ 的每一部伺服器上。佇列管理程式會共用預設資料目錄中資料檔案及目錄的存取權。它們需要對自己的佇列管理程式目錄樹狀結構 (包含在 第 115 頁的圖 49 中顯示的每一個 qmgrs 及 log 目錄中) 具有互斥存取權。

第 115 頁的圖 49 是 WebSphere MQ 目錄結構的高階視圖。它會顯示可以在佇列管理程式之間共用，並使其成為遠端的目錄。詳細資料因平台而異。點虛線指示可配置路徑。

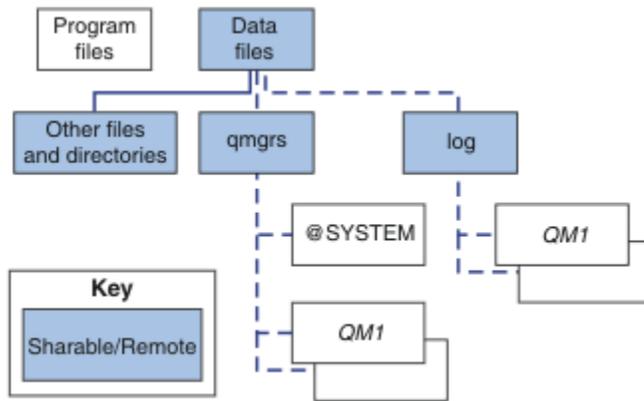


圖 49: WebSphere MQ 目錄結構的整體視圖

## 程式檔案

程式檔案目錄通常會保留在預設位置、位於本端，並由伺服器上的所有佇列管理程式共用。

## 資料檔

資料檔案目錄通常位於本端預設位置，在 UNIX and Linux 系統上為 `/var/mqm`，在 Windows 上安裝時為可配置。它在佇列管理程式之間共用。您可以將預設位置設為遠端，但不要在不同的 WebSphere MQ 安裝之間共用它。WebSphere MQ 配置中的 `DefaultPrefix` 屬性指向這個路徑。

## qmgrs

從 v7.0.1 開始，有兩種替代方式可指定佇列管理程式資料的位置。

### 使用字首

**字首** 屬性指定 `qmgrs` 目錄的位置。WebSphere MQ 會從佇列管理程式名稱建構佇列管理程式目錄名稱，並將它建立為 `qmgrs` 目錄的子目錄。

`Prefix` 屬性位於 `QueueManager` 段落中，並繼承自 `DefaultPrefix` 屬性中的值。依預設，為了簡化管理，佇列管理程式通常會共用相同的 `qmgrs` 目錄。

`QueueManager` 段落位於 `mqc.ini` 檔案中。

如果您變更任何佇列管理程式的 `qmgrs` 目錄位置，則必須變更其 **字首** 屬性的值。

UNIX and Linux 平台之 [第 115 頁的圖 49](#) 中 `QM1` 目錄的 `Prefix` 屬性為：

```
Prefix=/var/mqm
```

### 使用 DataPath

`DataPath` 屬性指定佇列管理程式資料目錄的位置。

`DataPath` 屬性指定完整路徑，包括佇列管理程式資料目錄的名稱。`DataPath` 屬性與 `Prefix` 屬性不同，後者指定佇列管理程式資料目錄的不完整路徑。

`DataPath` 屬性 (如果有指定的話) 位於 `QueueManager` 段落中。如果已指定，則它優先於 `Prefix` 屬性中的任何值。

`QueueManager` 段落位於 `mqc.ini` 檔案中。

如果您變更任何佇列管理程式的佇列管理程式資料目錄位置，則必須變更 `DataPath` 屬性的值。

UNIX 或 Linux 平台之 [第 115 頁的圖 49](#) 中 `QM1` 目錄的 `DataPath` 屬性為：

```
DataPath=/var/mqm/qmgrs/QM1
```

## log

在佇列管理程式配置的 **日誌** 段落中，會針對每一個佇列管理程式個別指定日誌目錄。佇列管理程式配置位於 `qm.ini` 中。

## DataPath/QmgrName/@IPCC 子目錄

`DataPath/QmgrName/@IPCC` 子目錄位於共用目錄路徑中。它們用來建構 IPC 檔案系統物件的目錄路徑。當在系統之間共用佇列管理程式時，他們需要識別佇列管理程式的名稱空間。在 V7.0.1 之前，佇列管理程式僅在一個系統上使用。一組子目錄足以定義 IPC 檔案系統物件的目錄路徑，請參閱第 116 頁的圖 50。

```
DataPath/QmgrName/@IPCC/esem
```

圖 50: IPC 子目錄範例 pre-V7.0.1

在 V7.0.1 以及更新版本中，IPC 檔案系統物件必須由系統識別。針對佇列管理程式執行所在的每一個系統，會將子目錄新增至目錄路徑，請參閱第 116 頁的圖 51。

```
DataPath/QmgrName/@IPCC/esem/myHostName/
```

圖 51: IPC 子目錄範例 V7.0.1 及後續版本

`myHostName` 最多是作業系統所傳回主機名稱的前 20 個字元。在部分系統上，在截斷之前，主機名稱長度最多可能為 64 個字元。產生的 `myHostName` 值可能會導致問題，原因有兩個：

1. 前 20 個字元不是唯一的。
2. 主機名稱是由 DHCP 演算法所產生，該演算法不一定會將相同的主機名稱配置給系統。

在這些情況下，請使用環境變數 `MQC_IPC_HOST` 來設定 `myHostName`；請參閱第 116 頁的圖 52。

```
export MQC_IPC_HOST=myHostName
```

圖 52: 範例: 設定 `MQC_IPC_HOST`

## 其他檔案和目錄

其他檔案及目錄 (例如包含追蹤檔的目錄及一般錯誤日誌) 通常會共用並保留在本端檔案系統上。

直到 v7.0.1 為止，WebSphere MQ 都依賴外部管理來保證佇列管理程式對佇列管理程式資料和日誌檔的互斥存取。從 v7.0.1 開始，在支援共用檔案系統的情況下，WebSphere MQ 會使用檔案系統鎖定來管理對這些檔案的互斥存取。檔案系統鎖定一次只容許特定佇列管理程式的一個實例處於作用中狀態。

當您啟動特定佇列管理程式的第一個實例時，它會取得其佇列管理程式目錄的所有權。如果您啟動第二個實例，則只有在第一個實例已停止時，它才能取得所有權。如果第一個佇列管理程式仍在執行中，則第二個實例無法啟動，並報告佇列管理程式正在其他位置執行。如果第一個佇列管理程式已停止，則第二個佇列管理程式會接管佇列管理程式檔案的所有權，並變成執行中的佇列管理程式。

您可以自動執行從第一個佇列管理程式接管的第二個佇列管理程式的程序。使用 `strmqm -x` 選項來啟動第一個佇列管理程式，該選項允許另一個佇列管理程式接管該佇列管理程式。然後，第二個佇列管理程式會等待直到佇列管理程式檔案解除鎖定，再嘗試接管佇列管理程式檔案的所有權，然後啟動。

## UNIX and Linux 系統上的目錄結構

UNIX and Linux 系統上的 WebSphere MQ 目錄結構可以對映至不同的檔案系統，以獲得更簡單的管理、更好的效能及更好的可靠性。

使用 WebSphere MQ 的彈性目錄結構，以利用共用檔案系統來執行多重實例佇列管理程式。

使用指令 `crtmqm QM1` 來建立第 117 頁的圖 53 中顯示的目錄結構，其中 R 是產品的版次。它是從 v7.0.1 開始在 WebSphere MQ 系統上建立之佇列管理程式的一般目錄結構。為了清楚起見，會省略部分目錄、檔案及 `.ini` 屬性設定，且另一個佇列管理程式名稱可能會因 mangling 而變更。檔案系統的名稱會在不同的系統上改變。

在一般安裝中，您建立的每個佇列管理程式都會指向本端檔案系統上的一般 `log` 及 `qmgrs` 目錄。在多重實例配置中，`log` 和 `qmgrs` 目錄位於與另一個 WebSphere MQ 安裝共用的網路檔案系統上。

第 117 頁的圖 53 顯示 WebSphere MQ v7.R on AIX，其中 R 是產品的版次。如需替代多重實例配置的範例，請參閱第 121 頁的『UNIX and Linux 系統上的目錄配置範例』。

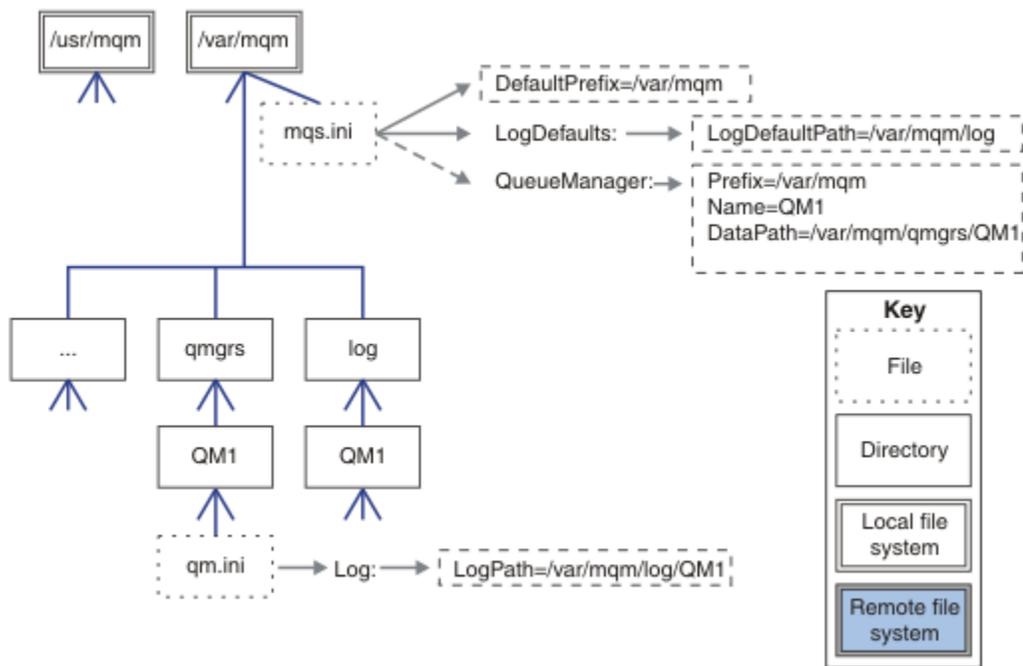


圖 53: UNIX and Linux 系統的預設 WebSphere MQ v7.R 目錄結構範例

依預設，產品會安裝至 AIX 上的 /usr/mqm 及其他系統上的 /opt/mqm。工作目錄會安裝到 /var/mqm 目錄中。

註: 如果您在安裝 IBM WebSphere MQ 之前已建立 /var/mqm 檔案系統，請確保 mqm 使用者具有完整目錄許可權，例如，檔案模式 755。

log 和 qmgrs 目錄會顯示在其預設位置中，如 mqs.ini 檔中 LogDefault 路徑和 DefaultPrefix 屬性的預設值所定義。建立佇列管理程式時，依預設會在 DefaultPrefix/qmgrs 中建立佇列管理程式資料目錄，並在 LogDefaultPath /log 中建立日誌檔目錄。LogDefault 路徑和 DefaultPrefix 只會影響依預設建立佇列管理程式和日誌檔的情況。佇列管理程式目錄的實際位置儲存在 mqs.ini 檔案中，日誌檔目錄的位置儲存在 qm.ini 檔案中。

佇列管理程式的日誌檔目錄定義在 LogPath 屬性的 qm.ini 檔案中。在 **crtmqm** 指令上使用 -ld 選項，以設定佇列管理程式的 LogPath 屬性; 例如 **crtmqm -ld LogPath QM1**。如果您省略 ld 參數，則會改用 LogDefault 路徑 的值。

佇列管理程式資料目錄定義在 mqs.ini 檔案中 QueueManager 段落的 DataPath 屬性中。在 **crtmqm** 指令上使用 -md 選項，以設定佇列管理程式的 DataPath; 例如 **crtmqm - md DataPath QM1**。如果您省略 md 參數，則會改用 DefaultPrefix 或 Prefix 屬性的值。Prefix 優先於 DefaultPrefix。

通常，在單一指令中同時指定日誌和資料目錄，以建立 QM1。

```
crtmqm
-md DataPath -ld
LogPath QM1
```

當佇列管理程式停止時，您可以編輯 qm.ini 檔中的 DataPath 及 LogPath 屬性，來修改佇列管理程式日誌及現有佇列管理程式資料目錄的位置。

errors 目錄的路徑 (如同 /var/mqm 中所有其他目錄的路徑) 無法修改。不過，目錄可以裝載在不同的檔案系統上，或以符號方式鏈結至不同的目錄。

## UNIX and Linux 系統上的目錄內容

與佇列管理程式相關聯的目錄內容。

如需產品檔案位置的相關資訊，請參閱 [選擇安裝位置](#)

如需替代目錄配置的相關資訊，請參閱 [第 105 頁的『規劃檔案系統支援』](#)。

在第 119 頁的圖 54 中，此佈置是在佇列管理程式已使用一段時間之後，WebSphere MQ 的代表。您具有的實際結構視佇列管理程式上發生的作業而定。

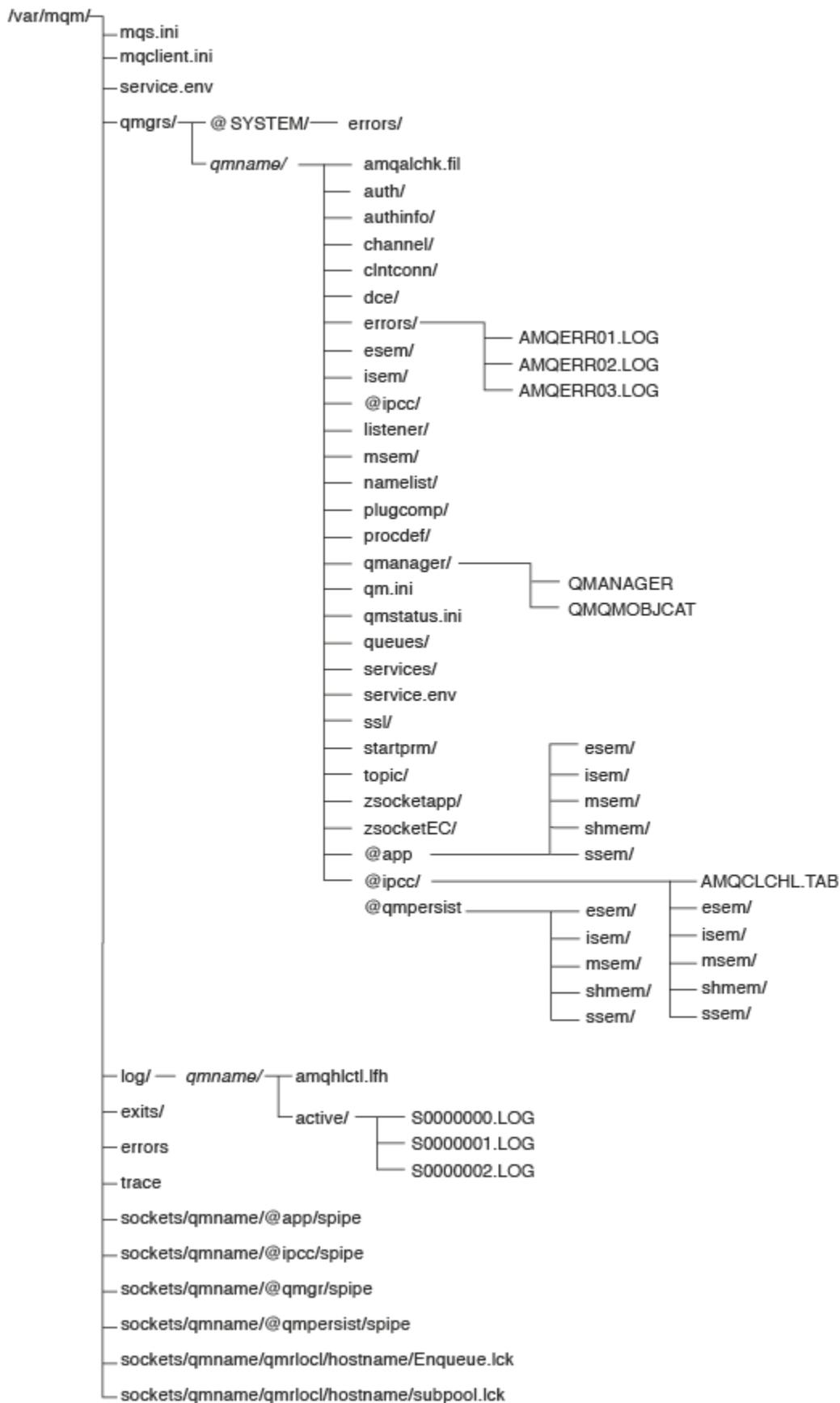


圖 54: 啟動佇列管理程式之後的預設目錄結構 (UNIX 系統)

## **`/var/mqm/`**

`/var/mqm` 目錄包含適用於整個 WebSphere MQ 安裝的配置檔及輸出目錄，而不適用於個別佇列管理程式。

表 15: UNIX 系統上 `/var/mqm` 目錄的記載內容

<b><u>mqs.ini</u></b>	WebSphere MQ 安裝層面配置檔，在佇列管理程式啟動時讀取。 可使用 <b>AMQ_MQS_INI_LOCATION</b> 環境變數修改檔案路徑。 請確定已在執行 <b>strmqm</b> 指令的 Shell 中設定並匯出此項。
<b><u>mqclient.ini</u></b>	WebSphere MQ MQI 用戶端程式所讀取的預設用戶端配置檔。 可使用 <b>MQCLNTCF</b> 環境變數修改檔案路徑。
<b><u>service.env</u></b>	包含服務程序的機器範圍環境變數。 檔案路徑已修正。
<b><u>錯誤/</u></b>	機器範圍錯誤日誌及 FFST 檔案。 已修正目錄路徑。 另請參閱 <a href="#">FFST: IBM WebSphere MQ 適用於 UNIX and Linux 系統</a> 。
<b><u>Socket/</u></b>	包含每一個佇列管理程式的資訊，僅供系統使用。
<b><u>trace/</u></b>	追蹤檔。 已修正目錄路徑。
<b><u>結束程式/</u></b>	包含使用者通道結束程式的預設目錄。
<b><u>exits64/</u></b>	在 <code>mqs.ini</code> 檔案的 <code>ApiExit</code> 段落中可修改的位置。

## **`/var/mqm/qmgrs/qmname/`**

`/var/mqm/qmgrs/qmname/` 包含佇列管理程式的目錄及檔案。目錄已鎖定，作用中佇列管理程式實例無法進行互斥存取。目錄路徑可在 `mqs.ini` 檔案中直接修改，或使用 **crtmqm** 指令的 **md** 選項來修改。

表 16: UNIX 系統上 `/var/mqm/qmgrs/qmname` 目錄的記載內容

<b><u>qm.ini</u></b>	佇列管理程式配置檔，在佇列管理程式啟動時讀取。
<b><u>錯誤/</u></b>	佇列管理程式範圍錯誤日誌。 <code>qmname = @system</code> 包含不明或無法使用佇列管理程式的通道相關訊息。
<b><u>@ipcc/AMQCLCHL.TAB</u></b>	預設用戶端通道控制表格，由 WebSphere MQ 伺服器建立，並由 WebSphere MQ MQI 用戶端程式讀取。 可使用 <b>MQCHLLIB</b> 和 <b>MQCHLTAB</b> 環境變數修改檔案路徑。
<b>QMANAGER</b>	佇列管理程式物件檔: QMANAGER 佇列管理程式物件型錄: QMQMOBJCAT

表 16: UNIX 系統上 `/var/mqm/qmgrs/qmname` 目錄的記載內容 (繼續)

<b>authinfo/</b>	佇列管理程式內定義的每一個物件都與這些目錄中的檔案相關聯。 檔名大約符合定義名稱; 請參閱 <a href="#">瞭解 WebSphere MQ 檔名</a> 。
<b>channel/</b>	
<b>clntconn/</b>	
<b>listener/</b>	
<b>namelist/</b>	
<b>procdef/</b>	
<b>queues/</b>	
<b>services/</b>	
<b>topics/</b>	
...	WebSphere MQ 使用的其他目錄 (例如 @ipcc) , 僅供 WebSphere MQ 修改。

### **`/var/mqm/log/qmname/`**

`/var/mqm/log/qmname/` 包含佇列管理程式日誌檔。目錄已鎖定, 作用中佇列管理程式實例無法進行互斥存取。目錄路徑可在 `qm.ini` 檔案中修改, 或使用 `crtmqm` 指令的 `ld` 選項來修改。

表 17: UNIX 系統上 `/var/mqm/log/qmname` 目錄的記載內容

<b>amqhlctl.lfh</b>	日誌控制檔。
<b>active/</b>	此目錄包含編號為 <code>S0000000.LOG</code> , <code>S0000001.LOG</code> , <code>S0000002.LOG</code> 等。

### **UNIX and Linux 系統上的目錄配置範例**

UNIX and Linux 系統上替代檔案系統配置的範例。

您可以用各種方式自訂 WebSphere MQ 目錄結構, 以達到許多不同的目標。

- 將 `qmgrs` 和 `log` 目錄放在遠端共用檔案系統上, 以配置多重實例佇列管理程式。
- 針對資料及日誌目錄使用個別檔案系統, 並將目錄配置給不同的磁碟, 以透過減少 I/O 競用來增進效能。
- 對對效能有更大影響的目錄使用更快的儲存裝置。在持續傳訊的效能中, 實體裝置延遲通常比在本端或遠端裝載裝置更重要。下列清單顯示哪些目錄對效能最敏感且最不敏感。

1. `log`
2. `qmgrs`
3. 其他目錄, 包括 `/usr/mqm`

- 在檔案系統上建立 `qmgrs` 和 `log` 目錄, 以配置給具有良好復原力的儲存體, 例如備援磁碟陣列。
- 最好將一般錯誤日誌儲存在 `var/mqm/errors` 本端, 而不是儲存在網路檔案系統上, 以便記載與網路檔案系統相關的錯誤。

第 122 頁的圖 55 是衍生替代 WebSphere MQ 目錄結構的範本。在範本中, 點虛線代表可配置的路徑。在範例中, 點虛線取代為實線, 其對應於儲存在 `AMQ_MQS_INI_LOCATION` 環境變數中以及 `mqs.ini` 和 `qm.ini` 檔案中的配置資訊。

註: 路徑資訊會顯示在 `mqs.ini` 或 `qm.ini` 檔案中。如果您在 `crtmqm` 指令中提供路徑參數, 請省略佇列管理程式目錄的名稱: 在佇列管理程式名稱被損壞之後, WebSphere MQ 會將它新增至路徑中。

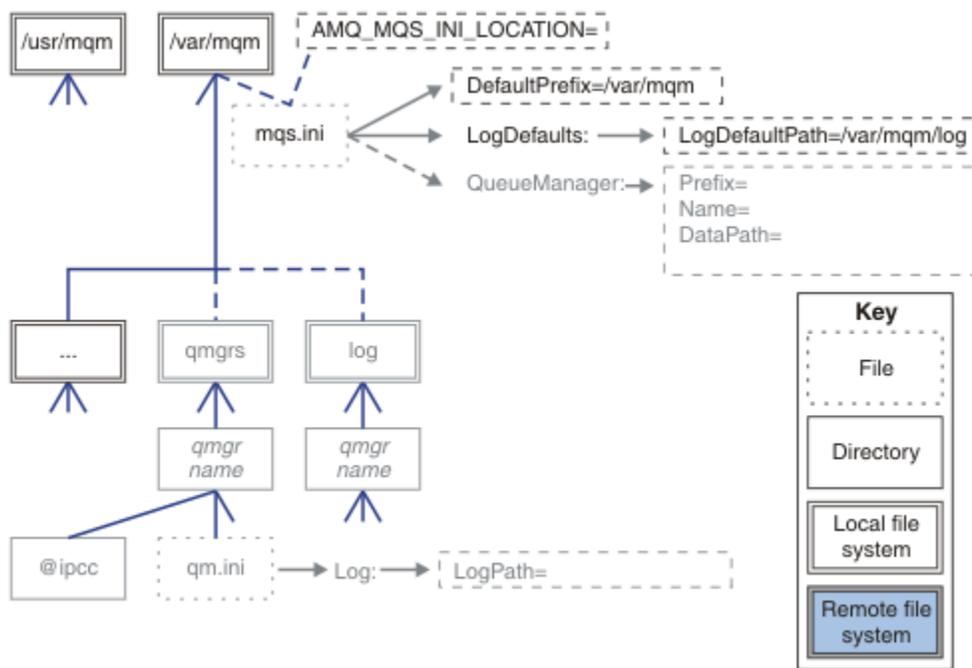


圖 55: 目錄結構型樣範本

以下是已配置的目錄結構範例。第一個範例顯示透過發出 `crtmqm QM1` 指令建立 WebSphere MQ v7.0.1 的一般預設目錄結構。第二個範例顯示如何針對使用 WebSphere 早於 v7.0.1 的 MQ 版本所建立的佇列管理程式，顯示一般目錄結構。目錄結構不會變更。

在 7.0.1 版中新建立的佇列管理程式與舊版 v7 具有不同的配置檔。如果您需要移除 v7.0.1 修正套件以回復至 v7.0.0.2，則需要重建配置檔。您可能只需要使用 `Prefix` 屬性來定義新佇列管理程式資料目錄的路徑，或者您可能需要將佇列管理程式資料目錄和日誌目錄移至不同位置。重新配置佇列管理程式最安全的方法是儲存佇列管理程式資料及日誌目錄，刪除並重建佇列管理程式，然後將其新位置中的資料及日誌目錄取代為已儲存的資料及日誌目錄。

### 從 v7.0.1 版開始的一般目錄結構

第 123 頁的圖 56 是透過發出指令 `crtmqm QM1` 在 v7.0.1 中建立的預設目錄結構。

`mqs.ini` 檔案具有 QM1 佇列管理程式的段落，其建立方式是參照 `DefaultPrefix` 的值。`qm.ini` 檔案中的 `Log` 段落具有 `LogPath` 的值，透過參照 `mqs.ini` 中的 `LogDefault` 路徑進行設定。

使用選用 `crtmqm` 參數來置換 `DataPath` 和 `LogPath` 的預設值。

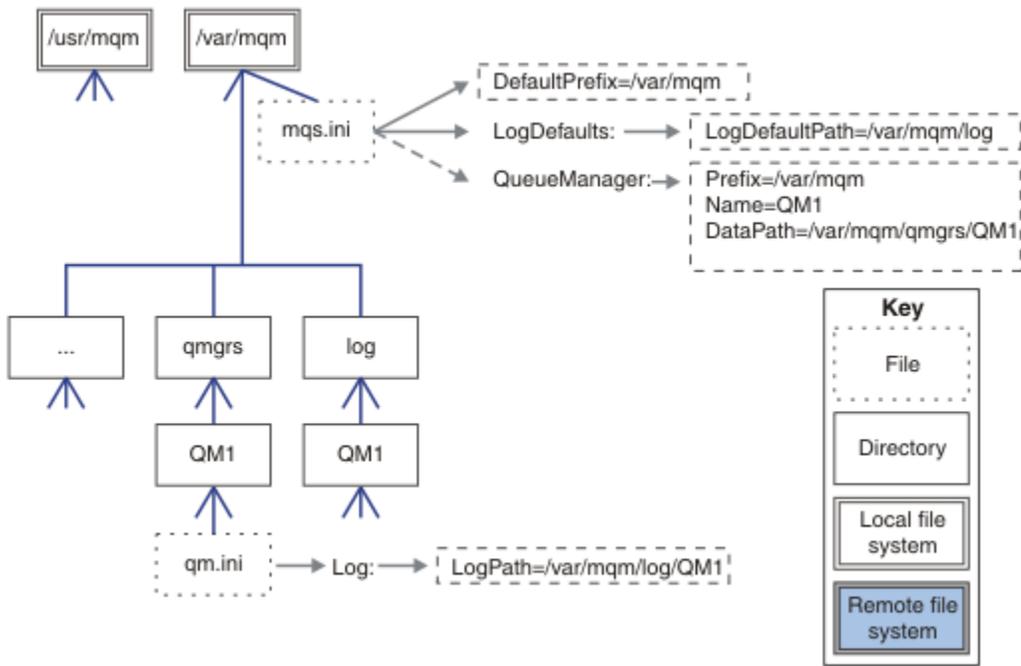


圖 56: UNIX and Linux 系統的預設 WebSphere MQ v7.R 目錄結構範例

### v7.0.1 之前版本的一般目錄結構

在 WebSphere MQ v7.0.1; 之前, DataPath 屬性不存在 mqs.ini 檔案中。已使用 Prefix 屬性來配置 qmgrs 目錄的位置。您可以使用符號鏈結來配置個別目錄的位置, 以指向不同的檔案系統位置。

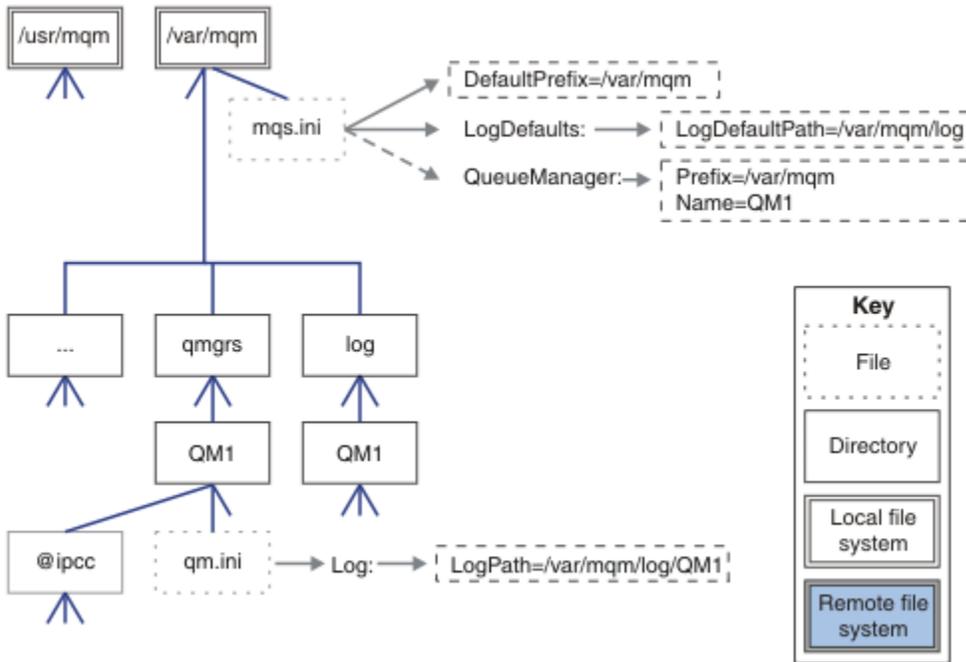


圖 57: v7.0.1 之前版本的一般目錄結構

### 共用預設 qmgrs 和 log 目錄 (從 v7.0.1 開始)

第 125 頁的『共用所有項目 (從 v7.0.1 版開始)』的替代方案是個別共用 qmgrs 和 log 目錄 (第 124 頁的圖 58)。在此配置中, 不需要設定 AMQ\_MQS\_INI\_LOCATION, 因為預設 mqs.ini 儲存在本端 /var/mqm 檔案系統中。檔案和目錄 (例如 mqclient.ini 和 mqserver.ini ) 也不會共用。

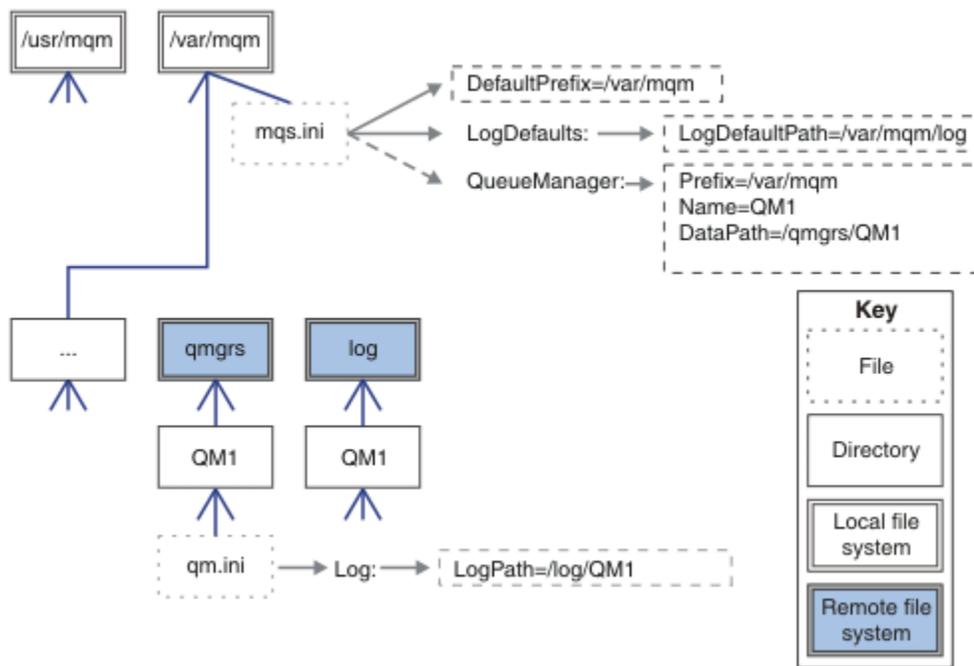


圖 58: 共用 *qmgrs* 和 *log* 目錄

#### 共用品名 *qmgrs* 及 *log* 目錄 (從 v7.0.1 開始)

第 124 頁的圖 59 中的配置會將 *log* 和 *qmgrs* 放在稱為 */ha* 的一般具名遠端共用檔案系統中。可以使用兩種不同的方式來建立相同的實體配置。

1. 設定 `LogDefaultPath=/ha`，然後執行指令 `crtmqm -md /ha/qmgrs QM1`。結果與第 124 頁的圖 59 中所說明的完全相同。
2. 保留預設路徑不變，然後執行指令 `crtmqm -ld /ha/log - md /ha/qmgrs QM1`。

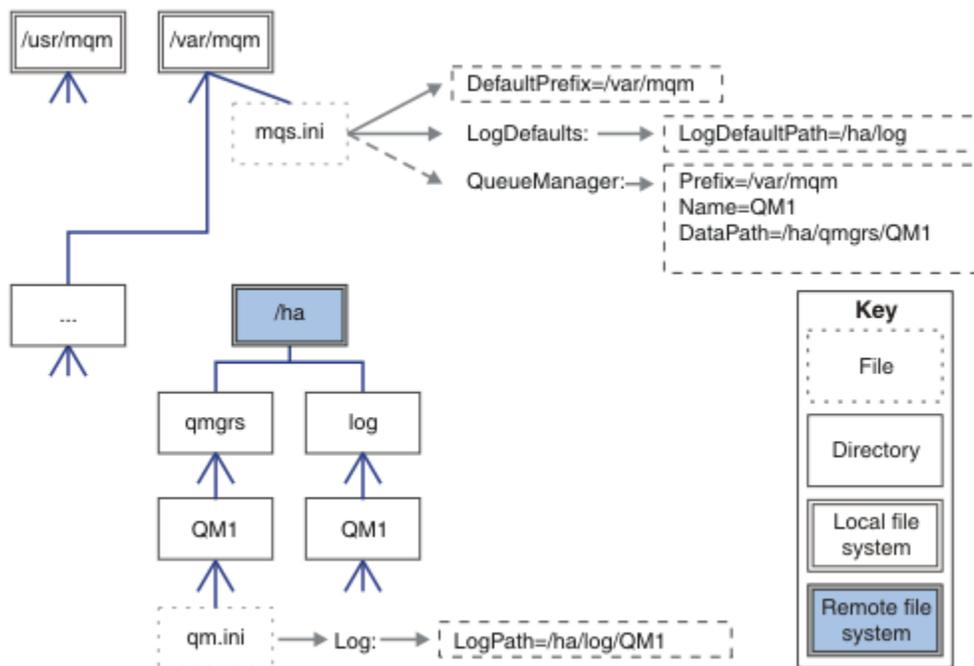


圖 59: 共用指名的 *qmgrs* 及 *log* 目錄

## 共用所有項目 (從 v7.0.1 版開始)

第 125 頁的圖 60 是具有快速網路檔案儲存體之系統的簡單配置。

將 /var/mqm 裝載為遠端共用檔案系統。依預設，當您啟動 QM1 時，它會尋找 /var/mqm，在共用檔案系統上尋找它，並讀取 /var/mqm 中的 mqs.ini 檔案。您可以在每一部伺服器上設定 AMQ\_MQS\_INI\_LOCATION 環境變數，以指向不同的 mqs.ini 檔案，而不是對所有伺服器上的佇列管理程式使用單一 /var/mqm/mqs.ini 檔案。

註: /var/mqm/errors/ 中一般錯誤檔的內容會在不同伺服器上的佇列管理程式之間共用。

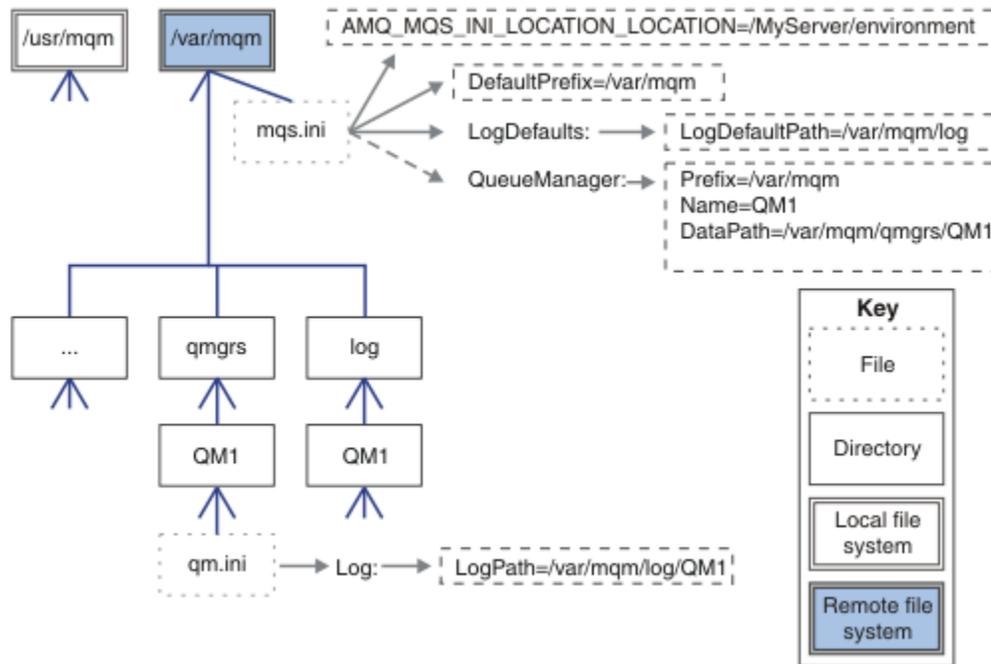


圖 60: 共用所有項目

請注意，您無法將此用於多重實例佇列管理程式。原因是多重實例佇列管理程式中的每一部主機都必須有自己的 /var/mqm 本端副本，才能追蹤本端資料 (例如號誌和共用記憶體)。這些實體無法在主機之間共用。

## Windows 系統上的目錄結構

如何在 Windows 上尋找佇列管理程式配置資訊及目錄。

IBM WebSphere MQ for Windows 安裝的預設目錄為:

### 32 位元

C:\Program Files\IBM\WebSphere MQ

### 64 位元

C:\Program Files (x86)\IBM\WebSphere MQ

安裝資訊儲存在 Windows 登錄中。儲存 IBM WebSphere MQ 資訊的登錄機碼為:

### 32 位元

My Computer\HKEY\_LOCAL\_MACHINE\SOFTWARE\IBM\WebSphere MQ\

### 64 位元

My Computer\HKEY\_LOCAL\_MACHINE\SOFTWARE\WOW6432Node\IBM\WebSphere MQ\

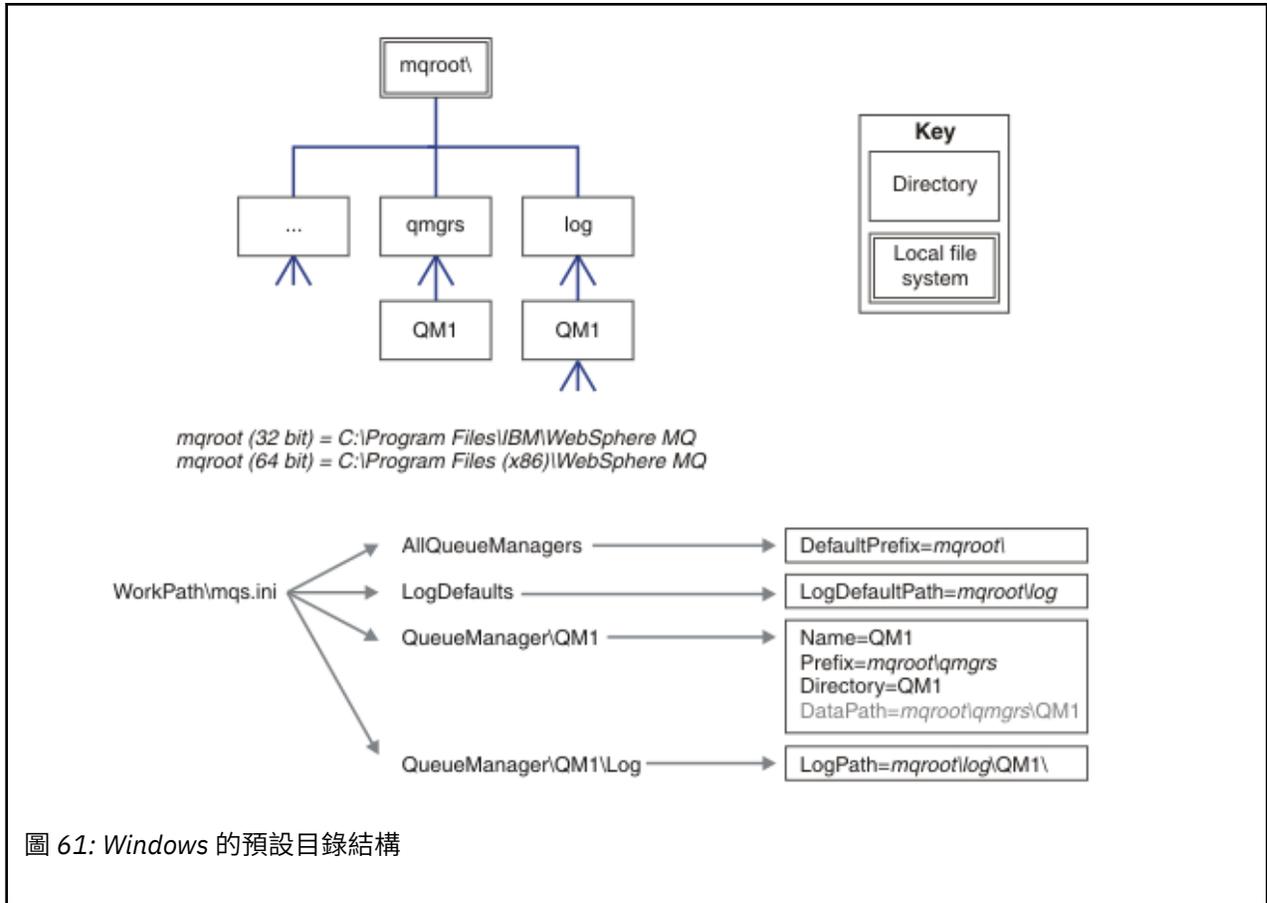
每一個安裝都有特定的子金鑰:

```
Installation\<InstallationName>\
```

指向 IBM WebSphere MQ 資料目錄的路徑儲存在名為 `WorkPath` 的字串值中，日誌的預設目錄儲存在 `LogDefaultPath` 中。佇列管理程式資料目錄建立在 `WorkPath\qmgrs\Qmgrname` 中。佇列管理程式日誌建立在 `LogDefaultPath\QmgrName` 中。請參閱 第 126 頁的圖 61。

如果您在安裝 IBM WebSphere MQ 時定義佇列管理程式資料及日誌目錄，則會使用自訂路徑資訊來更新 `WorkPath` 及 `LogDefault` 路徑。

`WorkPath` 和 `LogDefault` 路徑 只用來建立佇列管理程式。



## 多重實例佇列管理程式

若要配置多重實例佇列管理程式，必須將日誌及資料目錄放置在網路儲存體上，最好是放置在與執行佇列管理程式實例的任何伺服器不同的伺服器上。

在 `crtmqm` 指令 `-md` 及 `-ld` 上提供兩個參數，可讓您更輕鬆地指定佇列管理程式資料及日誌目錄的位置。指定 `-md` 參數的效果是四重：

1. `mq.ini` 段落 `QueueManager\QmgrName` 包含新的變數 `DataPath`，其指向佇列管理程式資料目錄。與 `Prefix` 變數不同，路徑包括佇列管理程式目錄的名稱。
2. 儲存在 `mq.ini` 檔案中的佇列管理程式配置資訊會縮減為 名稱、字首、目錄及 `DataPath`。

## 目錄內容

列出 WebSphere MQ 目錄的位置和內容。

WebSphere MQ 配置有三組主要檔案和目錄：

1. 執行檔及其他唯讀檔案 (例如 `Readme` 檔、「WebSphere MQ 探險家」外掛程式及說明檔，以及授權檔)，只有在套用維護時才會更新。這些檔案在 第 127 頁的表 18 中說明。
2. 非特定佇列管理程式專用的可能可修改檔案及目錄。這些檔案和目錄在 第 127 頁的表 19 中說明。
3. 伺服器上每一個佇列管理程式特定的檔案及目錄。這些檔案和目錄在 第 127 頁的表 18 中說明

## 資源目錄和檔案

資源目錄和檔案包含執行佇列管理程式的所有執行碼和資源。安裝特定 IBM WebSphere MQ 配置登錄機碼中的變數 *FilePath* 包含資源目錄的路徑。

檔案路徑	內容
<i>FilePath</i> \bin	指令和 DLL
<i>FilePath</i> \bin64	指令及 DLL (64 位元)
<i>FilePath</i> \conv	資料轉換表
<i>FilePath</i> \doc	精靈說明檔
<i>FilePath</i> \MQExplorer	「瀏覽器」和「瀏覽器」說明 Eclipse 外掛程式
<i>FilePath</i> \gskit8	廣域安全套件
<i>FilePath</i> \java	Java 資源, 包括 JRE
<i>FilePath</i> \licenses	授權資訊
<i>FilePath</i> \Non_IBM_License	授權資訊
<i>FilePath</i> \properties	內部使用
<i>FilePath</i> \Tivoli	
<i>FilePath</i> \tools	開發資源和範例
<i>FilePath</i> \Uninst	內部使用
<i>FilePath</i> \README.TXT	ReadMe 檔

## 非佇列管理程式特定的目錄

部分目錄包含並非特定佇列管理程式特有的檔案, 例如追蹤檔及錯誤日誌。 *DefaultPrefix* 變數包含這些目錄的路徑。 *DefaultPrefix* 是 *AllQueueManagers* 段落的一部分。

檔案路徑	內容
<i>DefaultPrefix</i> \Config	內部使用
<i>DefaultPrefix</i> \conv	ccsid.tbl 資料轉換控制檔, 如 <a href="#">資料轉換</a> 中所述
<i>DefaultPrefix</i> \errors	非佇列管理程式錯誤日誌 AMQERRnn.LOG
<i>DefaultPrefix</i> \exits	通道結束程式
<i>DefaultPrefix</i> \exits64	通道結束程式 (64 位元)
<i>DefaultPrefix</i> \ipc	未使用
<i>DefaultPrefix</i> \Qmgrs	在 <a href="#">第 128 頁的表 20</a> 中說明
<i>DefaultPrefix</i> \trace	追蹤檔案
<i>DefaultPrefix</i> \amqmjpse.txt	內部使用

## 佇列管理程式目錄

當您建立佇列管理程式時, 會建立佇列管理程式特有的一組新目錄。

如果您使用 `-md filepath` 參數建立佇列管理程式，則路徑會儲存在 `mqs.ini` 檔案的佇列管理程式段落中的 `DataPath` 變數。如果您在未設定 `-md filepath` 參數的情況下建立佇列管理程式，則會在 `DefaultPrefix` 所儲存的路徑中建立佇列管理程式目錄，並將路徑複製到 `mqs.ini` 檔案的佇列管理程式段落中的 `Prefix` 變數。

表 20: <code>DataPath</code> 和 <code>Prefix/Qmgrs/QmgrName</code> 目錄中的目錄和檔案	
檔案路徑	內容
<code>DataPath\@ipcc</code>	AMQCLCHL.TAB 用戶端連線表格的預設位置。
<code>DataPath\authinfo</code>	內部使用。
<code>DataPath\channel</code>	
<code>DataPath\clntconn</code>	
<code>DataPath\errors</code>	錯誤日誌, AMQERRnn.LOG
<code>DataPath\listener</code>	內部使用。
<code>DataPath\namelist</code>	
<code>DataPath\plugcomp</code>	
<code>DataPath\procdef</code>	
<code>DataPath\qmanager</code>	
<code>DataPath\queues</code>	
<code>DataPath\services</code>	
<code>DataPath\ssl</code>	
<code>DataPath\startprm</code>	
<code>DataPath\topic</code>	
<code>DataPath\active</code>	
<code>DataPath\active.dat</code>	
<code>DataPath\amqalchk.fil</code>	
<code>DataPath\master</code>	
<code>DataPath\master.dat</code>	
<code>DataPath\qm.ini</code>	佇列管理程式配置
<code>DataPath\qmstatus.ini</code>	佇列管理程式狀態
<code>Prefix\Qmgrs\QmgrName</code>	內部使用
<code>Prefix\Qmgrs\@SYSTEM</code>	未使用
<code>Prefix\Qmgrs\@SYSTEM\errors</code>	

## IBM WebSphere MQ 及 UNIX System V IPC 資源

佇列管理程式使用部分 IPC 資源。請使用 `ipcs -a` 來找出正在使用哪些資源。

此資訊僅適用於在 **UNIX and Linux** 系統上執行的 **IBM WebSphere MQ**。

IBM WebSphere MQ 使用 System V 跨處理程序通訊 (IPC) 資源 (號誌及 共用記憶體區段) 在系統元件之間儲存及傳遞資料。這些資源由連接至佇列管理程式的佇列管理程式處理程序及應用程式使用。IBM WebSphere MQ MQI 用戶端不會使用 IPC 資源，但 IBM WebSphere MQ 追蹤控制除外。使用 UNIX 指令 `ipcs -a` 可取得機器上目前使用的 IPC 資源數目及大小的完整資訊。

## AIX 上的共用記憶體

如果某些應用程式類型由於 AIX 記憶體限制而無法連接，在大部分情況下可以透過設定環境變數 `EXTSHM=ON` 來解決此問題。

AIX 上的部分 32 位元處理程序可能會遇到作業系統限制，這會影響其連接至 WebSphere MQ 佇列管理程式的能力。WebSphere MQ 的每個標準連線都使用共用記憶體，但與其他 UNIX and Linux 平台不同，AIX 只容許 32 位元處理程序連接 11 個共用記憶體集。

大部分 32 位元處理程序不會遇到此限制，但具有高記憶體需求的應用程式可能無法連接至 WebSphere MQ，原因碼為 2102: MQRC\_RESOURCE\_PROBLEM。下列應用程式類型可能會看到此錯誤：

- 在 32 位元 Java 虛擬機器中執行的程式
- 使用大型或非常大記憶體模型的程式
- 連接至許多佇列管理程式或資料庫的程式
- 自行連接至共用記憶體集的程式

AIX 提供 32 位元處理程序的延伸共用記憶體功能，可讓它們連接更多共用記憶體。若要使用此特性執行應用程式，請先匯出環境變數 `EXTSHM=ON`，然後再啟動佇列管理程式及程式。在大部分情況下，`EXTSHM=ON` 特性會防止此錯誤，但它與使用 `shmctl` 函數的 `SHM_SIZE` 選項的程式不相容。

WebSphere MQ MQI 用戶端應用程式及所有 64 位元處理程序不受此限制的影響。不論是否已設定 `EXTSHM`，他們都可以連接至 WebSphere MQ 佇列管理程式。

## WebSphere MQ 和 UNIX 處理程序優先順序

設定處理程序優先順序 *nice* 值時的良好作法。

此資訊僅適用於在 UNIX and Linux 系統上執行的 WebSphere MQ。

如果您在背景中執行處理程序，則可以透過呼叫 Shell 為該處理程序提供較高的 *nice* 值 (因此優先順序較低)。這可能具有一般 WebSphere MQ 效能含意。在高度緊張的狀況下，如果有許多備妥可執行的執行緒處於較高優先順序且有些處於較低優先順序，則作業系統排程性質會剝奪較低優先順序執行緒的處理器時間。

最好是獨立啟動與佇列管理程式相關聯的處理程序 (例如 `runmq1sr`) 具有與其相關聯的佇列管理程式相同的 *nice* 值。請確定 Shell 不會將較高的 *nice* 值指派給這些背景處理程序。例如，在 `ksh` 中，使用設定 `"set +o bgnice"` 來阻止 `ksh` 提高背景處理程序的 *nice* 值。您可以檢查 `"ps -efl"` 清單的 *NI* 直欄，以驗證執行中處理程序的 *nice* 值。

此外，啟動與佇列管理程式具有相同 *nice* 值的 WebSphere MQ 應用程式程序。如果它們以不同的 *nice* 值執行，應用程式執行緒可能會封鎖佇列管理程式執行緒，反之亦然，導致效能降低。

## 在 HP Integrity NonStop Server 上規劃 IBM WebSphere MQ 用戶端環境

當您規劃 IBM WebSphere MQ 環境時，必須考量 HP Integrity NonStop Server 環境及 HP NonStop TMF。使用此資訊來規劃 HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端執行所在的環境。

在規劃 IBM WebSphere MQ Client for HP Integrity NonStop Server 架構之前，請先熟悉基本 IBM WebSphere MQ Client for HP Integrity NonStop Server 概念，請參閱 [IBM WebSphere MQ Client for HP Integrity NonStop Server 技術概觀](#) 中的主題。

## 準備 HP Integrity NonStop Server 環境

在安裝之前，必須根據是否要立即驗證安裝，來準備環境。

對於安裝，您需要下列項目：

- 符合需求的使用者 ID。如需使用者 ID 需求的詳細資料，請參閱 [在 HP Integrity NonStop Server 上設定使用者和群組](#)。
- OSS 和 Guardian 檔案系統中可用於安裝檔案的已驗證位置。
- 可運作的 OSS Shell 及 OSS 檔案系統。您可以執行下列作業來驗證檔案系統：

- 登入 OSS 環境 (Shell)。請確定您對您想要使用的 OSS 安裝根目錄具有寫入權。
- 使用 MQM 群組中的使用者 ID 登入 TACL 環境。請驗證您要使用的磁區符合需求且可供您存取，且子磁區不存在。

您可以使用別名 (如果有的話) 或完整主體來登入 OSS 或 TACL。

如果您想要立即繼續進行以驗證安裝是否可用，則可能還需要下列選用項目：

- OSS 環境中可運作且可存取的「本端 Socket」子系統。
- 可運作的 TCP/IP 子系統。

如果您想要使用 TMF 協調的廣域工作單元，則需要下列項目：

- 可運作的 TMF 子系統。
- 作業 Pathway (TS/MP) 子系統。

如果您對這些重要子系統的狀態有任何疑問，請與系統管理者合作。

## IBM WebSphere MQ 和 HP NonStop TMF

HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端可以參與 HP NonStop 交易管理機能 (HP NonStop TMF) 協調工作單元。只有在佇列管理程式位於 IBM WebSphere MQ Version 7.1 或更新版本時，才支援與 HP NonStop TMF 協調交易。

IBM WebSphere MQ 提供的 TMF/ 閘道會將交易從 TMF 協調轉換為 eXtended Architecture (XA) 交易協調，以與遠端佇列管理程式進行通訊。IBM WebSphere MQ 提供的 TMF/ 閘道是 TMF 與佇列管理程式交易之間的橋接器，使用 HP NonStop TMF 提供的服務，並且設計成在 Pathway 環境中執行。

HP NonStop TMF 軟體在嚴苛環境中提供交易保護和資料庫一致性。如需 HP NonStop TMF 的相關資訊，請參閱 [HP NonStop TMF 簡介](#)。

如需如何配置 IBM WebSphere MQ 提供的 TMF/ 閘道的相關資訊，請參閱 [配置 HP Integrity NonStop Server](#)。

## 使用 HP NonStop TMF

HP NonStop 交易管理機能 (TMF) 是 HP Integrity NonStop Server 上的原生交易管理程式，並與檔案系統及關聯式資料庫管理程式、SQL/MP 和 SQL/MX 整合。

HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端可以使用 TMF 來協調廣域工作單元。

若要協調廣域工作單元，TMF 充當交易管理程式，應用程式必須使用 TMF 提供的 API 來啟動、確定及取消廣域工作單元。應用程式會呼叫 BEGINTRANSACTION 來啟動廣域工作單元，然後在同步點控制內發出 MQPUT、MQPUT1 和 MQGET 呼叫來更新廣域工作單元內的 IBM WebSphere MQ 資源。然後，應用程式可以透過呼叫 ENDTRANSACTION 來確定廣域工作單元，或透過呼叫 ABORTTRANSACTION 來取消它。

使用 TMF 交易的應用程式一次只能主動處理一個交易，不過，使用 RESUMETRANSACTION 可讓應用程式從一個作用中交易切換至另一個作用中交易，或與無 TMF 交易相關聯，而無需完成或中斷先前的作用中交易。對 MQPUT、MQPUT1 或 MQGET 的任何呼叫都是在目前作用中的 TMF 交易 (如果存在的話) 或本端工作單元 (如果不存在的話) 下進行。因此，必須在應用程式內小心，以確保這些呼叫是在正確的工作單元內進行。

在廣域工作單元內，以及更新 IBM WebSphere MQ 資源中，應用程式可以更新 Enscribe 檔案、SQL/MP 資料庫或 SQL/MX 資料庫。

## 使用廣域工作單元

廣域工作單元實作為 TMF 交易。應用程式透過呼叫 BEGINTRANSACTION 來啟動廣域工作單元，並透過呼叫 ENDTRANSACTION 來確定工作單元，或透過呼叫 ABORTTRANSACTION 來取消工作單元。應用程式也可以使用其他 TMF API 呼叫。

應用程式可以從另一個應用程式繼承 TMF 交易。例如，應用程式 (第一個應用程式) 可以在交易內執行工作，然後再回覆並將交易傳回第二個應用程式進行進一步處理。因此，第一個和第二個應用程式都可以參與相同的廣域工作單元，其中涉及 IBM WebSphere MQ 佇列的更新，以及檔案和資料庫的更新。在應用程式

之間傳遞 TMF 交易的能力，表示數個 IBM WebSphere MQ 應用程式可以在相同的廣域工作單元內執行傳訊作業。

應用程式可以同時管理及控制多個作用中 TMF 交易。交易可以由應用程式本身啟動，或繼承自其他應用程式，或兩者。這表示應用程式可以同時參與多個廣域工作單元。

每個處理程序的並行作用中 TMF 交易數目上限為 1000，這是架構限制。如果應用程式正在管理多個 TMF 交易，則在任何時間點只能有一個現行交易。或者，沒有任何交易可以是現行交易。應用程式可以使用 TMF API 呼叫 (例如 RESUMETRANSACTION、ACTIVATERECEIVETRANSID 及 TMF\_SET\_TX\_ID)，將現行狀態從一個交易移至另一個交易，或指定沒有現行交易。應用程式會使用這個控制層次來判斷傳訊作業是在本端工作單元、廣域工作單元內執行，還是在同步點控制之外執行：

- 當沒有現行 TMF 交易時，如果應用程式在同步點控制內呼叫 MQPUT、MQPUT1 或 MQGET，IBM WebSphere MQ 會處理本端工作單元內的呼叫。
- 當應用程式具有現行 TMF 交易時，如果應用程式在同步點控制內呼叫 MQPUT、MQPUT1 或 MQGET，IBM WebSphere MQ 會在現行 TMF 交易所實作的廣域工作單元內處理呼叫。
- 如果應用程式在同步點控制之外呼叫 MQPUT、MQPUT1 或 MQGET，不論應用程式在呼叫時是否有現行 TMF 交易，IBM WebSphere MQ 都會在同步點控制之外處理呼叫。

除非在處理期間發生軟體或硬體故障，且 IBM WebSphere MQ 或作業系統判定必須取消交易以保留資料完整性，否則 IBM WebSphere MQ 絕不會在 MQI 呼叫期間變更應用程式的 TMF 交易狀態。每個 MQI 呼叫都會在將控制權交還給應用程式之前，還原應用程式的交易狀態。

## 避免長時間執行交易

避免設計 TMF 交易保持作用中超過數十秒的應用程式。長時間執行交易可能會導致 TMF 的循環審核追蹤填滿。因為 TMF 是重要的全系統資源，所以 TMF 會透過取消作用過久的應用程式交易來保護自己。

假設應用程式內的處理是由從佇列取得訊息所驅動，且應用程式會從佇列取得訊息，並在工作單元內處理訊息。一般而言，應用程式會使用等待選項並在同步點控制內呼叫 MQGET，以從佇列中取得訊息。

如果應用程式改用廣域工作單元，則 MQGET 呼叫上指定的等待間隔必須短，以避免長時間執行交易。這表示應用程式可能需要在擷取訊息之前多次發出 MQGET 呼叫。



## 注意事項

本資訊係針對 IBM 在美國所提供之產品與服務所開發。

在其他國家中，IBM 可能不會提供本書中所提的各項產品、服務或功能。請洽當地 IBM 業務代表，以取得當地目前提供的產品和服務之相關資訊。這份文件在提及 IBM 的產品、程式或服務時，不表示或暗示只能使用 IBM 的產品、程式或服務。只要未侵犯 IBM 的智慧財產權，任何功能相當的產品、程式或服務都可以取代 IBM 的產品、程式或服務。不過，任何非 IBM 的產品、程式或服務，使用者必須自行負責作業的評估和驗證責任。

本文件所說明之主題內容，IBM 可能擁有其專利或專利申請案。提供本文件不代表提供這些專利的授權。您可以書面提出授權查詢，來函請寄到：

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

如果是有關雙位元組 (DBCS) 資訊的授權查詢，請洽詢所在國的 IBM 智慧財產部門，或書面提出授權查詢，來函請寄到：

智慧財產權授權  
法務部與智慧財產權法律  
IBM Japan, Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan

**下列段落不適用於英國，若與任何其他國家之法律條款抵觸，亦不適用於該國：** International Business Machines Corporation 只依 "現況" 提供本出版品，不提供任何明示或默示之保證，其中包括且不限於不侵權、可商用性或特定目的之適用性的隱含保證。有些地區在特定交易上，不允許排除明示或暗示的保證，因此，這項聲明不一定適合您。

這項資訊中可能有技術上或排版印刷上的訛誤。因此，IBM 會定期修訂；並將修訂後的內容納入新版中。IBM 隨時會改進及/或變更本出版品所提及的產品及/或程式，不另行通知。

本資訊中任何對非 IBM 網站的敘述僅供參考，IBM 對該網站並不提供任何保證。這些網站所提供的資料不是 IBM 本產品的資料內容，如果要使用這些網站的資料，您必須自行承擔風險。

IBM 得以各種適當的方式使用或散布由您提供的任何資訊，無需對您負責。

如果本程式的獲授權人為了 (i) 在個別建立的程式和其他程式（包括本程式）之間交換資訊，以及 (ii) 相互使用所交換的資訊，因而需要相關的資訊，請洽詢：

IBM Corporation  
軟體交互作業能力協調程式，部門 49XA  
3605 公路 52 N  
Rochester, MN 55901  
U.S.A.

在適當條款與條件之下，包括某些情況下（支付費用），或可使用此類資訊。

IBM 基於雙方之 IBM 客戶合約、IBM 國際程式授權合約或任何同等合約之條款，提供本資訊所提及的授權程式與其所有適用的授權資料。

本文件中所含的任何效能資料都是在受管制的環境下判定。因此不同作業環境之下所得的結果，可能會有很大的差異。有些測定已在開發階段系統上做過，不過這並不保證在一般系統上會出現相同結果。甚至有部分的測量，是利用插補法而得的估計值，實際結果可能有所不同。本文件的使用者應驗證其特定環境適用的資料。

本文件所提及之非 IBM 產品資訊，取自產品的供應商，或其發佈的聲明或其他公開管道。IBM 並未測試過這些產品，也無法確認這些非 IBM 產品的執行效能、相容性或任何對產品的其他主張是否完全無誤。有關非 IBM 產品的性能問題應直接洽詢該產品供應商。

有關 IBM 未來方針或目的之所有聲明，僅代表 IBM 的目標與主旨，隨時可能變更或撤銷，不必另行通知。

這份資訊含有日常商業運作所用的資料和報告範例。為了要使它們儘可能完整，範例包括個人、公司、品牌和產品的名稱。這些名稱全屬虛構，如與實際公司的名稱和住址雷同，純屬巧合。

著作權授權：

本資訊含有原始語言之範例應用程式，用以說明各作業平台中之程式設計技術。您可以基於研發、使用、銷售或散布符合作業平台（撰寫範例程式的作業平台）之應用程式介面的應用程式等目的，以任何形式複製、修改及散布這些範例程式，而不必向 IBM 付費。這些範例並未在所有情況下完整測試。因此，IBM 不保證或暗示這些程式的可靠性、有用性或功能。

若貴客戶正在閱讀本項資訊的電子檔，可能不會有照片和彩色說明。

## 程式設計介面資訊

---

程式設計介面資訊 (如果有提供的話) 旨在協助您建立與此程式搭配使用的應用軟體。

本書包含預期程式設計介面的相關資訊，可讓客戶撰寫程式以取得 IBM WebSphere MQ 的服務。

不過，本資訊也可能包含診斷、修正和調整資訊。提供診斷、修正和調整資訊，是要協助您進行應用軟體的除錯。

**重要：**請勿使用此診斷、修改及調整資訊作為程式設計介面，因為它可能會變更。

## 商標

---

IBM、IBM 標誌 [ibm.com](http://www.ibm.com) 是 IBM Corporation 在全球許多適用範圍的商標。IBM 商標的最新清單可在 Web 的 "Copyright and trademark information" [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml) 中找到。其他產品和服務名稱，可能是 IBM 或其他公司的商標。

Microsoft 及 Windows 是 Microsoft Corporation 在美國及/或其他國家或地區的商標。

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

Linux 是 Linus Torvalds 在美國及/或其他國家或地區的註冊商標。

本產品包含 Eclipse Project (<http://www.eclipse.org/>) 所開發的軟體。

Java 和所有以 Java 為基礎的商標及標誌是 Oracle 及/或其子公司的商標或註冊商標。





產品編號:

(1P) P/N: