

8.0

規劃 *IBM MQ*

IBM

附註

使用本資訊及其支援的產品之前，請先閱讀第 161 頁的『[注意事項](#)』中的資訊。

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

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

© Copyright International Business Machines Corporation 2007, 2023.

目錄

| | |
|---|------------|
| 規劃 | 5 |
| 針對 GDPR 就緒狀態的 IBM MQ 和 IBM MQ Appliance 內部部署考量..... | 5 |
| 以單一佇列管理程式為基礎的架構..... | 13 |
| 以多個佇列管理程式為基礎的架構..... | 14 |
| 規劃分散式佇列及叢集..... | 14 |
| 規劃分散式發佈/訂閱網路..... | 56 |
| 規劃儲存體及效能需求..... | 88 |
| 分散式平台上的磁碟空間需求..... | 89 |
| 規劃檔案系統支援..... | 90 |
| IBM MQ 及 UNIX System V IPC 資源..... | 117 |
| AIX 上的共用記憶體..... | 117 |
| IBM MQ 和 UNIX 程序優先順序..... | 117 |
| 在 HP Integrity NonStop Server 上規劃 IBM MQ 用戶端環境..... | 118 |
| 準備 HP Integrity NonStop Server 環境..... | 118 |
| IBM MQ 和 HP NonStop TMF..... | 118 |
| 使用 HP NonStop TMF..... | 118 |
| 在 z/OS 上規劃 IBM MQ 環境..... | 119 |
| 在 z/OS 上規劃儲存體及效能需求..... | 120 |
| 規劃頁面集及緩衝池..... | 125 |
| 規劃連結機能及卸載儲存體環境..... | 132 |
| 規劃記載環境..... | 141 |
| 規劃 IBM MQ Managed File Transfer..... | 148 |
| 規劃通道起始程式 SMF 資料..... | 151 |
| 規劃備份及回復..... | 152 |
| 規劃 z/OS UNIX 或 UNIX 系統服務環境..... | 159 |
| 規劃 z/OS TCP/IP 環境..... | 159 |
| 注意事項 | 161 |
| 程式設計介面資訊..... | 162 |
| 商標..... | 162 |

規劃

規劃 IBM MQ 環境時，請考量 IBM MQ 為單一及多個佇列管理程式架構提供的支援，以及點對點及發佈/訂閱傳訊樣式的支援。同時規劃您的資源需求，以及使用記載和備份機能。

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

IBM 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『規劃分散式佇列及叢集』](#)
 - [第 56 頁的『規劃分散式發佈/訂閱網路』](#)

 在 IBM MQ for z/OS 上，您可以使用共用佇列及佇列共用群組，讓您能夠實作工作量平衡，並讓 IBM MQ 應用程式具有可調式及高可用性。如需共用佇列及佇列共用群組的相關資訊，請參閱 [共用佇列及佇列共用群組](#)。

如需規劃多個安裝、儲存體和效能需求以及使用用戶端的相關資訊，請參閱其他子主題。

相關概念

[第 119 頁的『在 z/OS 上規劃 IBM MQ 環境』](#)

規劃 IBM MQ 環境時，您必須考量資料集、頁面集、Db2、「連結機能」的資源需求，以及記載及備份機能的需求。請利用這個主題來規劃 IBM MQ 執行所在的環境。

相關資訊

[檢查需求](#)

[確定訊息未遺失 \(記載\)](#)

[可用性、回復及重新啟動](#)

針對 GDPR 就緒狀態的 IBM MQ 和 IBM MQ Appliance 內部部署考量

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, Value Unit Edition
- 5655-AM9 IBM MQ Advanced Message Security for z/OS
- 5725-S14 IBM MQ Appliance M2000
- 5725-Z09 IBM MQ Appliance M2001
- 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 File Transfer Edition for Multiplatforms
- 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 Managed File Transfer for z/OS

注意事項：

本文件目的在協助您準備 GDPR 備妥。它提供為協助組織針對 GDPR 備妥，您應考量之可配置的 IBM MQ 特性以及產品使用方面的相關資訊。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。就一切相關法律法規之確認與解釋，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法規所應採取之行為。

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、會計或審核建議，也不表示或保證其服務或產品將確保客戶符合任何法律或法規。

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 以協助您的組織針對 GDPR 備妥的考量。

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 可以用來交換許多形式的資料，其中有些可能受到 GDPR 的限制。有數個協力廠商產品可與 IBM MQ 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所提供。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，因為使用案例會隨著應用程式部署而不同。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 應用程式上，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。

以下是可能使用 IBM MQ 交換的個人資料的一些典型範例：

- 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，客戶可以使用 IBM MQ 在與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)

用於與 IBM 線上聯絡的個人資料

IBM MQ 客戶可以透過各種方式提交線上評論/意見/要求，以聯絡 IBM 關於 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 MQ 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?)
- 在 IBM MQ 需要存取協力廠商應用程式時，如何收集並儲存認證?

IBM MQ 可能需要與需要鑑別的其他系統及服務 (例如 LDAP) 進行通訊。必要時，IBM MQ 會配置並儲存鑑別資料 (使用者 ID、密碼)，以便在這類通訊中使用。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」。)

資料儲存空間

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) 至有狀態媒體。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其安全。

下列項目強調顯示 IBM MQ 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 時可能想要考量這些區域。

- 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 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，可記錄應用程式及通道的傳訊 API 活動，應用程式活動追蹤可記錄應用程式提供給事件訊息之訊息資料的內容。

- 服務追蹤:

IBM MQ 提供服務追蹤功能，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，以追蹤儲存在磁碟上的檔案。

- 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：

- 限制特許使用者對檔案系統中 IBM MQ 資料的存取權，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

可以透過下列產品介面來存取 IBM MQ 佇列管理程式資料，其中部分設計用於透過遠端連線進行存取，而其他設計用於透過本端連線進行存取。

- 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 遙測 (MQTT) [僅限遠端]
- IBM MQ 燈 (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]
- 適用於 HTTP 的 IBM MQ 橋接器 [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 橋接至 Salesforce [僅限遠端]
- IBM MQ 橋接至 Blockchain [僅限遠端]
- IBM MQ MQAI [本端和遠端]
- IBM MQ PCF 指令 [本端及遠端]
- IBM MQ MQSC 指令 [本端及遠端]
- IBM MQ Explorer [本端和遠端]

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，管理使用者可能被通道鑑別規則封鎖)，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. 可以配置 IBM MQ 佇列管理程式，以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 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 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。

獲授權使用 Advanced Message Security (AMS) 功能的 IBM MQ 使用者可以透過定義及配置安全原則，來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。

- 遵循 GDPR 用戶端資料刪除考量的 IBM MQ 行為區域
 - 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。

- 遵循 GDPR 帳戶資料刪除考量的 IBM MQ 行為區域
 - 您可以透過刪除 (包括保存、備份或其他抄寫副本)，來刪除 IBM MQ 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 外掛程式用來鑑別佇列管理程式及檔案伺服器的認證檔。
 - 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。
 - 來自 IBM MQ Appliance 的個人使用者帳戶，包括系統日誌檔中那些帳戶的參照。
 - IBM MQ Explorer 工作區 meta 資料和 Eclipse 設定。
 - IBM MQ Explorer 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 qm.ini 及 APIKeyFile 中 ReportingService 段落下的 Cloud 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 提供一個特性，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。

如果提供包含個人資料的訊息可以隨時識別及找到，則可以使用標準 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，以及在 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 類型或副檔名來採取動作。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。

以單一佇列管理程式為基礎的架構

最簡單的 IBM MQ 架構涉及配置及使用單一佇列管理程式。

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

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 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 MQ 發佈/訂閱傳送應用程式會根據資訊的主旨，發佈具有指定主題的訊息。IBM 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。如需相關資訊，請參閱 [發佈/訂閱傳訊](#) 和 [單一佇列管理程式發佈/訂閱配置範例](#)。

相關概念

[第 5 頁的『規劃』](#)

規劃 IBM MQ 環境時，請考量 IBM MQ 為單一及多個佇列管理程式架構提供的支援，以及點對點及發佈/訂閱傳訊樣式的支援。同時規劃您的資源需求，以及使用記載和備份機能。

相關資訊

[IBM MQ 簡介](#)

[在分散式平台上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 MQ 架構。

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

可以透過新增其他佇列管理程式來變更 IBM 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 MQ 網路的高階概觀，請參閱 [第 15 頁的『設計分散式佇列管理程式網路』](#)。

如需如何規劃 IBM MQ 架構的通道的相關資訊，請參閱 [IBM MQ 分散式佇列技術](#)。

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

[第 5 頁的『規劃』](#)

規劃 IBM MQ 環境時，請考量 IBM MQ 為單一及多個佇列管理程式架構提供的支援，以及點對點及發佈/訂閱傳訊樣式的支援。同時規劃您的資源需求，以及使用記載和備份機能。

相關資訊

[在分散式平台上建立及管理佇列管理程式](#)

規劃分散式佇列及叢集

您可以手動連接在分散式佇列管理程式上管理的佇列，也可以建立佇列管理程式叢集，並讓產品為您連接佇列管理程式。若要選擇適合分散式傳訊網路的拓撲，您需要考量手動控制、網路大小、變更頻率、可用性及其可調整性的需求。

開始之前

這項作業假設您瞭解何謂分散式傳訊網路，以及它們的運作方式。如需技術概觀，請參閱 [分散式佇列及叢集](#)。

關於這項作業

若要建立分散式傳訊網路，您可以手動配置通道來連接在不同佇列管理程式上管理的佇列，也可以建立佇列管理程式叢集。叢集作業可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。

若要選擇適合分散式發佈/訂閱網路的拓撲，您需要考量下列廣泛問題：

- 您需要對網路中的連線進行多少手動控制？
- 您的網路將會有多大？
- 會有多動態？
- 您的可用性和可擴充性需求為何？

程序

- 請考量您需要對網路中的連線進行多少手動控制。

如果您只需要少數連線，或者如果需要非常精確地定義個別連線，則您可能應該手動建立網路。

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，則應該考量將它們分組在佇列管理程式叢集中。

- 預估您的網路需要有多大。

- a) 預估您需要的佇列管理程式數目。請記住，佇列可以在多個佇列管理程式上管理。
- b) 如果您考慮使用叢集，請新增兩個額外的佇列管理程式來作為完整儲存庫。
對於較大的網路，手動配置及維護連線可能非常耗時，您應該考慮使用叢集。
- 請考量網路活動的動態程度。
規劃在效能佇列管理程式上管理忙碌佇列。
如果您預期經常建立及刪除佇列，請考慮使用叢集。
- 考量您的可用性和可擴充性需求。
 - a) 決定您是否需要保證佇列管理程式的高可用性。如果是的話，請預估此需求適用的佇列管理程式數目。
 - b) 請考量部分佇列管理程式的功能是否低於其他佇列管理程式。
 - c) 請考量部分佇列管理程式的通訊鏈結是否比其他佇列管理程式更脆弱。
 - d) 請考量在多個佇列管理程式上管理佇列。手動配置的網路及叢集都可以配置為高可用性及可調式。如果您使用叢集，則需要將兩個額外的佇列管理程式定義為完整儲存庫。有兩個完整儲存庫可確保當其中一個完整儲存庫變成無法使用時，叢集會繼續運作。請確定完整儲存庫佇列管理程式健全、效能良好，且具有良好的網路連線功能。請勿計劃將完整儲存庫佇列管理程式用於任何其他工作。
- 根據這些計算，使用提供的鏈結可協助您決定是手動配置佇列管理程式之間的連線，還是使用叢集。

下一步

現在您已準備好配置分散式傳訊網路。

相關資訊

[配置分散式佇列](#)

[配置佇列管理程式叢集](#)

設計分散式佇列管理程式網路

IBM MQ 會使用「佇列管理程式」及「通道」，透過網路在應用程式之間傳送及接收資料。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 16 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 16 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 16 頁的表 1](#) 中提供了 [第 16 頁的圖 1](#) 建議通道名稱的表格。

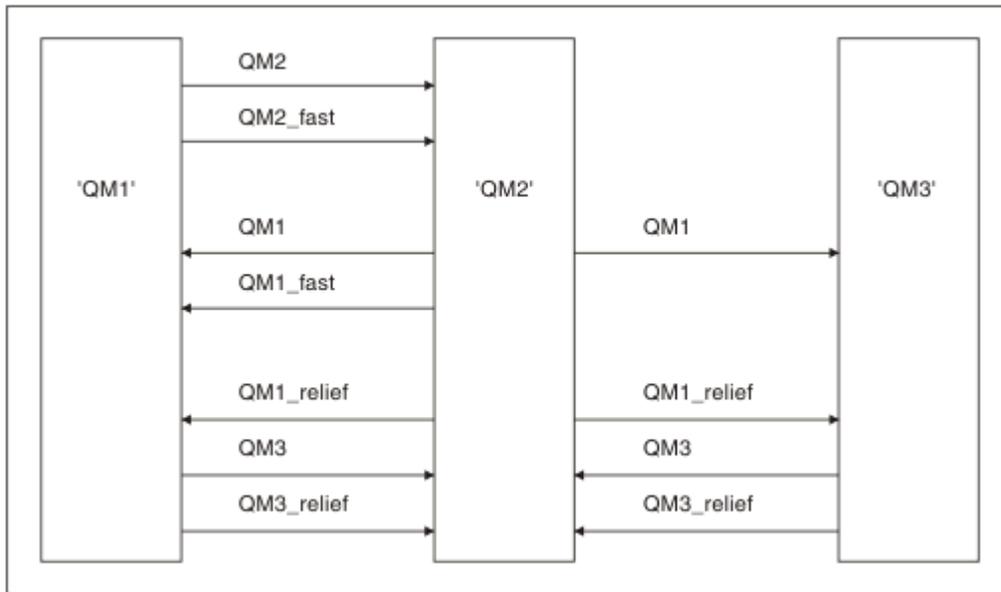


圖 1: 顯示所有通道的網路圖

| 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 IBM 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 16 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 17 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。

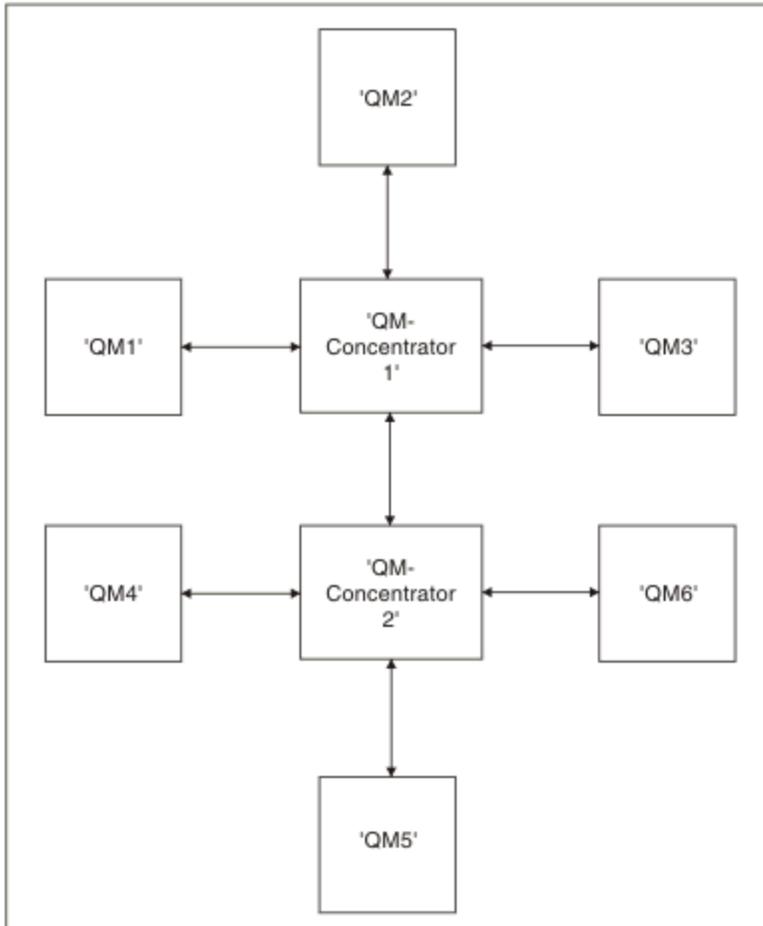


圖 2: 顯示 QM 集中器的網路圖

設計叢集

叢集提供一種機制，以簡化起始配置及進行中管理的方式來交互連接佇列管理程式。叢集必須小心設計，以確保它們正確運作，並達到所需的可用性和回應性層次。

開始之前

如需叢集作業概念的簡介，請參閱下列主題：

- [分散式佇列及叢集](#)
- [第 23 頁的『叢集作業與分散式佇列作業的比較』](#)
- [叢集的元素](#)

當您設計佇列管理程式叢集時，必須做出一些決策。您必須先決定叢集中哪些佇列管理程式要保留叢集資訊的完整儲存庫。您建立的任何佇列管理程式都可以在叢集中工作。您可以為此目的選擇任意數目的佇列管

理程式，但理想數目為 2。如需選取佇列管理程式來保留完整儲存庫的相關資訊，請參閱第 24 頁的『[如何選擇叢集佇列管理程式來保留完整儲存庫](#)』。

如需設計叢集的相關資訊，請參閱下列主題：

- [第 29 頁的『範例叢集』](#)
- [第 26 頁的『組織叢集』](#)
- [第 26 頁的『叢集命名慣例』](#)
- [第 27 頁的『佇列共用群組及叢集』](#)
- [第 27 頁的『重疊叢集』](#)

下一步

如需配置及使用叢集的相關資訊，請參閱下列主題：

- [在叢集中建立通訊](#)
- [配置佇列管理程式叢集](#)
- [將訊息遞送至叢集以及從叢集遞送訊息](#)
- [使用叢集進行工作量管理](#)

如需協助您配置叢集的相關資訊，請參閱第 28 頁的『[叢集提示](#)』。

規劃如何使用多個叢集傳輸佇列

您可以明確定義傳輸佇列，或讓系統為您產生傳輸佇列。如果您自行定義傳輸佇列，則對佇列定義有更大的控制權。  在 z/OS 上，您也可以進一步控制保留訊息的頁面集。

定義傳輸佇列

定義傳輸佇列有兩種方法：

- 自動使用佇列管理程式屬性 DEFCLXQ，如下所示：

```
ALTER QMGR DEFCLXQ(SCTQ | CHANNEL)
```

DEFCLXQ (SCTQ) 指出所有叢集傳送端通道的預設傳輸佇列是 SYSTEM.CLUSTER.TRANSMIT.QUEUE。這是預設值。

DEFCLXQ (CHANNEL) 指出依預設，每個叢集傳送端通道都會使用名為 SYSTEM.CLUSTER.TRANSMIT.<channel name>的個別傳輸佇列。每一個傳輸佇列由佇列管理程式自動定義。如需相關資訊，請參閱第 19 頁的『[自動定義的叢集傳輸佇列](#)』。

- 以手動方式定義傳輸佇列，並指定 CLCHNAME 屬性的值。CLCHNAME 屬性指出哪些叢集傳送端通道應該使用傳輸佇列。如需相關資訊，請參閱第 21 頁的『[規劃手動定義的叢集傳輸佇列](#)』。

我需要什麼安全？

若要自動或手動起始交換器，您需要啟動通道的權限。

若要定義用作傳輸佇列的佇列，您需要標準 IBM MQ 權限才能定義佇列。

何時才是適當時間執行變更？

變更叢集傳送端通道所使用的傳輸佇列時，您需要配置進行更新的時間，並考量下列要點：

- 通道切換傳輸佇列所需的時間取決於舊傳輸佇列上的訊息總數、需要移動的訊息數及訊息大小。
- 當發生變更時，應用程式可以繼續將訊息放入傳輸佇列。這可能會導致轉移時間增加。
- 您可以隨時變更任何傳輸佇列或 DEFCLXQ 的 CLCHNAME 參數，最好是在工作量偏低時。

請注意，不會立即發生任何事件。

- 只有在通道啟動或重新啟動時，才會發生變更。當通道啟動時，它會檢查現行配置，並在必要時切換至新的傳輸佇列。
- 有數個變更可能會變更叢集傳送端通道與傳輸佇列的關聯：
 - 變更傳輸佇列的 CLCHNAME 屬性值，使 CLCHNAME 較不明確或空白。
 - 變更傳輸佇列的 CLCHNAME 屬性值，使 CLCHNAME 更明確。
 - 刪除已指定 CLCHNAME 的佇列。
 - 變更佇列管理程式屬性 DEFCLXQ。

切換需要多久？

在轉移期間，通道的任何訊息都會從一個傳輸佇列移至另一個傳輸佇列。通道切換傳輸佇列所需的時間取決於舊傳輸佇列上的訊息總數，以及需要移動的訊息數。

對於包含數千則訊息的佇列，移動訊息應該需要幾秒的時間。實際時間取決於訊息的數目和大小。您的佇列管理程式應該可以每秒移動許多 MB 的訊息。

當發生變更時，應用程式可以繼續將訊息放入傳輸佇列。這可能會導致轉移時間增加。

必須重新啟動每一個受影響的叢集傳送端通道，變更才會生效。因此，最好在佇列管理程式不忙碌且叢集傳輸佇列上儲存少量訊息時變更傳輸佇列配置。

runswchl 指令， 或 z/OS 上的 CSQUTIL 中的 SWITCH CHANNEL (*) STATUS 指令，可用來查詢叢集傳送端通道的狀態，以及其傳輸佇列配置未完成的擱置變更。

如何實作變更

如需如何自動或手動變更更多個叢集傳輸佇列的詳細資料，請參閱 [使用多個叢集傳輸佇列實作系統](#)。

復原變更

如需發生問題時如何取消變更的詳細資料，請參閱 [復原變更](#)。

自動定義的叢集傳輸佇列
您可以讓系統為您產生傳輸佇列。

關於這項作業

如果通道沒有相關聯的手動定義叢集傳輸佇列，且您指定 DEFCLXQ (CHANNEL)，當通道啟動佇列管理程式時，會自動為叢集傳送端通道定義永久動態佇列。模型佇列 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE 用來自動定義名為 SYSTEM.CLUSTER.TRANSMIT.ChannelName。

 若要手動設定叢集傳輸佇列，請參閱 [第 21 頁的『規劃手動定義的叢集傳輸佇列』](#)。

重要： 

如果佇列管理程式已移轉至 IBM MQ 8.0，則佇列管理程式沒有 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE。

請先定義此佇列，讓指令 ALTER QGMGR DEFCLXQ (CHANNEL) 生效。

下列 JCL 是您可以用來定義模型佇列的程式碼範例：

```
//CLUSMODL JOB MSGCLASS=H,NOTIFY=&SYSUID
/*JOBPARM SYSAFF=(MVCC)
//MQCMD EXEC PGM=CSQUTIL,REGION=4096K,PARM='CDLK'
//STEPLIB DD DISP=SHR,DSN=SCEN.MQ.V000.COM.BASE.SCSQAUTH
// DD DISP=SHR,DSN=SCEN.MQ.V000.COM.BASE.SCSQANLE
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(CMDINP)
/*
//CMDINP DD *
DEFINE QMODEL( 'SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE' ) +
QSGDISP( QMGR ) +
* COMMON QUEUE ATTRIBUTES
```

```

DESCR( 'SYSTEM CLUSTERING TRANSMISSION MODEL QUEUE' ) +
PUT( ENABLED ) +
DEFPRTY( 5 ) +
DEFFPSIST( YES ) +

* MODEL QUEUE ATTRIBUTES
DEFTYPE( PERMDYN ) +

* LOCAL QUEUE ATTRIBUTES
GET( ENABLED ) +
SHARE +
DEFSOPT( EXCL ) +
MSGDLVSQ( PRIORITY ) +
RETINTVL( 999999999 ) +
MAXDEPTH( 999999999 ) +
MAXMSGL( 4194304 ) +
NOHARDENBO +
BOTHRESH( 0 ) +
BOQNAME( ' ' ) +
STGCLASS( 'REMOTE' ) +
USAGE( XMITQ ) +
INDXTYPE( CORRELID ) +
CFSTRUCT( ' ' ) +
MONQ( OFF ) ACCTQ( OFF ) +

* EVENT CONTROL ATTRIBUTES
QDPMAXEV( ENABLED ) +
QDPHIEV( DISABLED ) +
QDEPTHHI( 80 ) +
QDPLOEV( DISABLED ) +
QDEPTHLO( 40 ) +
QSVCIIEV( NONE ) +
QSVCIINT( 999999999 ) +

* TRIGGER ATTRIBUTES
TRIGGER +
TRIGTYPE( FIRST ) +
TRIGMPRI( 0 ) +
TRIGDPH( 1 ) +
TRIGDATA( ' ' ) +
PROCESS( ' ' ) +
INITQ( ' ' )
/*

```

程序

1. 使用 `DEFCLXQ` 佇列管理程式屬性。

如需此屬性的相關資訊，請參閱 [ALTER QMGR](#)。

有兩個選項：

Sctq

此選項是預設值，表示您使用單一 `SYSTEM.CLUSTER.TRANSMIT.QUEUE`。

CHANNEL

表示您使用多個叢集傳輸佇列。

2. 如果要切換至新的關聯，請執行下列動作：

- 停止並重新啟動通道。
- 通道會使用新的傳輸佇列定義。
- 訊息由轉移切換處理程序從舊佇列傳送至新傳輸佇列。

請注意，任何應用程式訊息都會放入舊定義中。

當舊佇列上的訊息數達到零時，新訊息會直接放置在新的傳輸佇列上。

3. 若要監視切換處理程序何時完成，請執行下列動作：

- a) 通道所起始的傳輸佇列切換在背景中執行，您的管理者可以監視佇列管理程式工作日誌，以判斷它何時完成。
- b) 監視工作日誌上的訊息，以顯示交換器的進度。
- c) 若要確保只有您想要的通道使用此傳輸佇列，請發出 `DIS CLUSQMGR(*)` 指令，例如，其中定義傳輸佇列的傳輸佇列內容是 `APPQMGR.CLUSTER1.XMITQ`。

d)

在 `CSQUTIL` 下使用 `SWITCH CHANNEL(*) STATUS` 指令。

這個選項會告訴您哪些擱置中的變更尚未完成，以及需要在傳輸佇列之間移動多少訊息。

結果

您已設定叢集傳輸佇列。

相關工作

第 21 頁的『[規劃手動定義的叢集傳輸佇列](#)』

如果您自行定義傳輸佇列，則對定義及保留訊息的頁集有更多控制權。

相關資訊

[ALTER QMGR](#)

[DISPLAY CLUSQMGR](#)

規劃手動定義的叢集傳輸佇列

如果您自行定義傳輸佇列，則對定義及保留訊息的頁集有更多控制權。

關於這項作業

您的管理者手動定義傳輸佇列，並使用新的佇列屬性 CLCHNAME 來定義哪些叢集傳送端通道或通道將使用此佇列作為其傳輸佇列。

請注意，CLCHNAME 可以在開頭或結尾包含萬用字元，以容許單一佇列用於多個通道。

若要自動設定叢集傳輸佇列，請參閱第 19 頁的『[自動定義的叢集傳輸佇列](#)』。

程序

1. 例如，請輸入下列指令：

```
DEFINE QLOCAL (APPQMGR.CLUSTER1.XMITQ)
CLCHNAME (CLUSTER1.TO.APPQMGR)
USAGE (XMITQ) STGCLASS (STG1)
INDXTYPE ( CORRELID ) SHARE

DEFINE STGCLASS (STG1) PSID (3)
DEFINE PSID (3) BUFFERPOOL (4)
```

提示：您需要規劃用於傳輸佇列的頁集 (及緩衝池)。您可以針對不同的佇列具有不同的頁集，並提供它們之間的隔離，因此一個頁集會填滿，不會影響其他頁集中的傳輸佇列。

如需每一個通道如何選取適當佇列的相關資訊，請參閱 [使用叢集傳輸佇列及叢集傳送端通道](#)。

當通道啟動時，它會將其關聯切換至新的傳輸佇列。為了確保不會遺失任何訊息，佇列管理程式會依序將訊息從舊叢集傳輸佇列自動傳送至新的傳輸佇列。

2. 請使用 CSQUTIL SWITCH 函數來變更為新的關聯。

如需進一步資訊，請參閱 [切換與叢集傳送端通道 \(SWITCH\) 相關聯的傳輸佇列](#)。

- a) 停止要變更其傳輸佇列的一或多個通道，使它們處於 STOPPED 狀態。

例如：

```
STOP CHANNEL (CLUSTER1.TO.APPQMGR)
```

- b) 變更傳輸佇列上的 CLCHNAME (XXXX) 屬性。
- c) 使用 SWITCH 函數來切換訊息或監視發生的情況。
使用指令

```
SWITCH CHANNEL (*) MOVEMSGS (YES)
```

移動訊息而不啟動通道。

- d) 請啟動一或多個通道，並檢查通道是否使用正確的佇列。

例如：

```
DIS CHS(CLUSTER1.TO.APPQMGR)
DIS CHS(*) where(XMITQ eq APPQMGR.CLUSTER1.XMITQ)
```

提示:

- 下列處理程序使用 CSQUTIL SWITCH 函數; 如需相關資訊, 請參閱 [切換與叢集傳送端通道 \(SWITCH\) 相關聯的傳輸佇列](#)。

您不需要使用此功能, 但使用此功能會提供更多選項:

- 使用 SWITCH CHANNEL (*) STATUS 可讓您輕鬆識別叢集傳送端通道的切換狀態。它可讓您的管理者查看目前正在切換的通道, 以及那些具有切換擱置的通道, 這些通道會在下次啟動時生效。

如果沒有此功能, 您的管理者需要使用多個 DISPLAY 指令, 然後處理產生的輸出以確定此資訊。您的管理者也可以確認配置變更具有想要的結果。

- 如果使用 CSQUTIL 來起始交換器, 則 CSQUTIL 會繼續監視此作業的進度, 且只有在交換器完成時才會結束。

這可讓您更容易以批次方式執行這些作業。此外, 如果執行 CSQUTIL 以切換多個通道, 則 CSQUTIL 會循序執行這些動作; 相較於平行執行的多個交換器, 這對企業的影響會較小。

結果

您已設定叢集傳輸佇列。

存取控制及多個叢集傳輸佇列

當應用程式將訊息放入遠端叢集佇列時, 請選擇三種檢查模式。這些模式包括對叢集佇列進行遠端檢查、對 SYSTEM.CLUSTER.TRANSMIT.QUEUE 進行本端檢查, 或對叢集佇列或叢集佇列管理程式的本端設定檔進行檢查。

IBM MQ 可讓您選擇在本端或本端及遠端檢查使用者是否有權將訊息放入遠端佇列。一般 IBM MQ 應用程式只會使用本端檢查, 並依賴遠端佇列管理程式來信任對本端佇列管理程式所進行的存取檢查。如果未使用遠端檢查, 則會以遠端訊息通道處理程序的權限將訊息放入目標佇列。若要使用遠端檢查, 您必須將接收通道的放置權限設為環境定義安全。

會針對應用程式開啟的佇列進行本端檢查。在分散式佇列中, 應用程式通常會開啟遠端佇列定義, 並針對遠端佇列定義進行存取檢查。如果放置具有完整遞送標頭的訊息, 則會針對傳輸佇列進行檢查。如果應用程式開啟不在本端佇列管理程式上的叢集佇列, 則沒有可檢查的本端物件。會針對叢集傳輸佇列 SYSTEM.CLUSTER.TRANSMIT.QUEUE 進行存取控制檢查。即使有多個叢集傳輸佇列, 也會從 7.5 版對 SYSTEM.CLUSTER.TRANSMIT.QUEUE 進行遠端叢集佇列的本端存取控制檢查。

選擇本端或遠端檢查是在兩個極端之間的選擇。遠端檢查是精細的。每一位使用者在叢集中的每一個佇列管理程式上都必須要有存取控制設定檔, 才能放入任何叢集佇列。本端檢查是粗略的。每一位使用者在他們所連接的佇列管理程式上, 只需要叢集傳輸佇列的一個存取控制設定檔。使用該設定檔, 他們可以將訊息放置到任何叢集中任何佇列管理程式上的任何叢集佇列。

從 7.1 版開始, 管理者有另一種方式來設定叢集佇列的存取控制。您可以使用 **setmqaut** 指令, 在叢集中的任何佇列管理程式上建立叢集佇列的安全設定檔。如果您在本端開啟遠端叢集佇列, 且只指定佇列名稱, 則設定檔會生效。您也可以設定遠端佇列管理程式的設定檔。如果您這麼做, 佇列管理程式可以提供完整名稱來檢查開啟叢集佇列之使用者的設定檔。

只有在您將佇列管理程式段落 **ClusterQueueAccessControl** 變更為 RQMName 時, 新的設定檔才會運作。預設值為 Xmitq。您必須為使用叢集佇列的所有叢集佇列現有應用程式建立設定檔。如果您將段落變更為 RQMName 而不建立設定檔, 則應用程式可能會失敗。

提示: 對 7.1 版中的叢集佇列存取檢查所做的變更不適用於遠端佇列作業。仍會針對本端定義進行存取權檢查。這些變更表示您可以遵循相同的方法來配置叢集佇列及叢集主題的存取權檢查。  這些變更也會使叢集佇列的存取權檢查方法與 z/OS 更接近。在 z/OS 上設定存取權檢查的指令不同, 但兩者都根據設定檔而非物件本身來檢查存取權。

相關資訊

叢集作業: 使用多個叢集傳輸佇列的應用程式隔離

叢集作業與分散式佇列作業的比較

比較需要定義的元件，以使用分散式佇列及叢集作業來連接佇列管理程式。

如果您不使用叢集，則佇列管理程式是獨立的，並使用分散式佇列作業進行通訊。如果一個佇列管理程式需要將訊息傳送至另一個佇列管理程式，您必須定義：

- 傳輸佇列
- 遠端佇列管理程式的通道

第 23 頁的圖 3 顯示分散式佇列所需的元件。

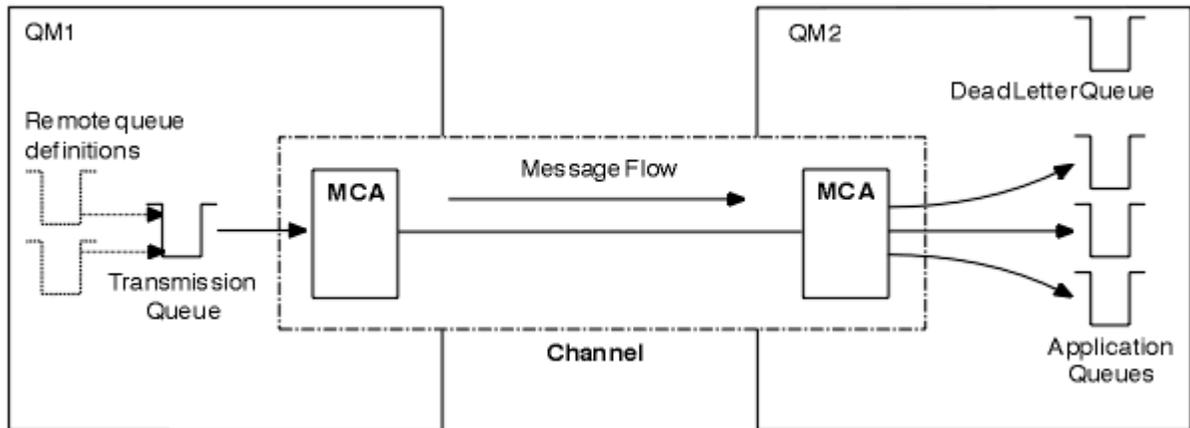


圖 3: 分散式佇列 (distributed queuing)

如果您將叢集中的佇列管理程式分組，則叢集中的任何其他佇列管理程式都可以使用任何佇列管理程式上的佇列。任何佇列管理程式都可以在沒有明確定義的情況下，將訊息傳送至相同叢集中的任何其他佇列管理程式。您不提供每一個目的地的通道定義、遠端佇列定義或傳輸佇列。叢集中的每個佇列管理程式都有單一傳輸佇列，可從中將訊息傳輸至叢集中的任何其他佇列管理程式。叢集中的每一個佇列管理程式只需要定義：

- 一個要接收訊息的叢集接收端通道
- 一個叢集傳送端通道，它用來建立自己並瞭解叢集

設定叢集與分散式佇列的定義

查看第 23 頁的圖 4，其中顯示四個佇列管理程式，各有兩個佇列。請考量需要多少定義才能使用分散式佇列來連接這些佇列管理程式。比較需要多少定義才能設定與叢集相同的網路。

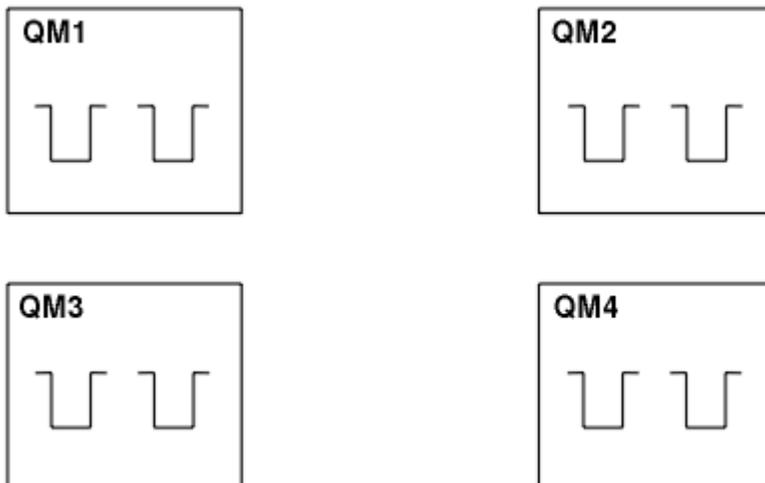


圖 4: 由四個佇列管理程式組成的網路

使用分散式佇列來設定網路的定義

若要使用分散式佇列來設定 [第 23 頁的圖 3](#) 中顯示的網路，您可能具有下列定義：

| 說明 | 每個佇列管理程式的數目 | 總數 |
|----------------------------------|-------------|----|
| 要將訊息傳送至每一個其他佇列管理程式之通道的傳送端-通道定義 | 3 | 12 |
| 要在其上接收來自所有其他佇列管理程式之訊息的通道的接收端通道定義 | 3 | 12 |
| 傳輸佇列至所有其他佇列管理程式的傳輸佇列定義 | 3 | 12 |
| 每一個本端佇列的本端佇列定義 | 2 | 8 |
| 此佇列管理程式要放置訊息之每一個遠端佇列的遠端佇列定義 | 6 | 24 |

您可以使用通用接收端通道定義來減少此定義數目。在每一個佇列管理程式上，定義數目上限可以多達 17 個，此網路總計 68 個。

使用叢集來設定網路的定義

若要使用叢集來設定 [第 23 頁的圖 3](#) 中顯示的網路，您需要下列定義：

| 說明 | 每個佇列管理程式的數目 | 總數 |
|-----------------------------------|-------------|----|
| 將訊息傳送至儲存庫佇列管理程式之通道的叢集傳送端通道定義 | 1 | 4 |
| 通道的叢集接收端通道定義，用來接收來自叢集中其他佇列管理程式的訊息 | 1 | 4 |
| 每一個本端佇列的本端佇列定義 | 2 | 8 |

若要設定此佇列管理程式叢集 (具有兩個完整儲存庫)，您在每一個佇列管理程式上需要四個定義，總共有 16 個定義。您也需要變更兩個佇列管理程式的佇列管理程式定義，讓它們成為叢集的完整儲存庫佇列管理程式。

只需要一個 CLUSSDR 及一個 CLUSRCVR 通道定義。當定義叢集時，您可以新增或移除佇列管理程式 (儲存庫佇列管理程式除外)，而不會對其他佇列管理程式造成任何毀壞。

使用叢集可減少設定包含許多佇列管理程式的網路所需的定義數目。

定義越少，錯誤風險就越低：

- 物件名稱一律相符，例如傳送端-接收端配對中的通道名稱。
- 通道定義中指定的傳輸佇列名稱一律符合正確的傳輸佇列定義或遠端佇列定義中指定的傳輸佇列名稱。
- QREMOTE 定義一律會指向遠端佇列管理程式上的正確佇列。

設定叢集之後，您可以將叢集佇列從一個佇列管理程式移至叢集內的另一個佇列管理程式，而不需要在任何其他佇列管理程式上執行任何系統管理工作。沒有機會忘記刪除或修改通道、遠端佇列或傳輸佇列定義。您可以將新的佇列管理程式新增至叢集，而不會對現有網路造成任何中斷。

如何選擇叢集佇列管理程式來保留完整儲存庫

在每一個叢集中，您必須至少選擇一個佇列管理程式 (最好是兩個佇列管理程式) 來保留完整儲存庫。除了最特殊的情況之外，兩個完整儲存庫都已足夠。可能的話，請選擇在健全且永久連接的平台上管理的佇列管理程式，這些佇列管理程式沒有巧合的中斷，且位於地理位置的中心位置。另請考量將系統專用作為完整儲存庫主機，而不要將這些系統用於任何其他作業。

完整儲存庫是佇列管理程式，可維護完整的叢集狀態。為了共用此資訊，每一個完整儲存庫都由 CLUSSDR 通道 (及其對應的 CLUSRCVR 定義) 連接至叢集中的所有其他完整儲存庫。您必須手動定義這些通道。



圖 5: 兩個已連接的完整儲存庫。

叢集中的每一個其他佇列管理程式都會維護其目前所知的局部儲存庫中叢集狀態的圖片。這些佇列管理程式會使用任何兩個可用的完整儲存庫，來發佈其本身的相關資訊，並要求其他佇列管理程式的相關資訊。如果選擇的完整儲存庫無法使用，則會使用另一個。當選擇的完整儲存庫再次變成可用時，它會從其他儲存庫收集最新的新資訊及變更的資訊，以便它們保持在步驟中。如果所有完整儲存庫都無法運作，其他佇列管理程式會使用它們在局部儲存庫中的資訊。不過，它們僅限於使用它們所擁有的資訊；無法處理新的資訊及更新要求。當完整儲存庫重新連接至網路時，會交換訊息，讓所有儲存庫 (完整和局部) 都保持最新。

規劃完整儲存庫的配置時，請包含下列考量：

- 選擇保留完整儲存庫的佇列管理程式需要可靠且受管理。選擇在健全且永久連接的平台上管理的佇列管理程式。
- 請考量管理完整儲存庫之系統的計劃性中斷執行，並確保它們沒有巧合的中斷執行。
- 考量網路效能：選擇位於地理位置中央的佇列管理程式，或與叢集中其他佇列管理程式共用相同系統的佇列管理程式。
- 請考量佇列管理程式是否為多個叢集的成員。在管理上方便使用相同的佇列管理程式來管理數個叢集的完整儲存庫，前提是此好處與您預期佇列管理程式的忙碌程度平衡。
- 請考慮讓部分系統只包含完整儲存庫，而不要將這些系統用於任何其他作業。這可確保這些系統只需要維護佇列管理程式配置，而不會從維護其他商業應用程式的服務中移除。它也可確保維護儲存庫的作業不會與應用程式競爭系統資源。這在大型叢集 (例如，超過一千個佇列管理程式的叢集) 中特別有用，其中完整儲存庫在維護叢集狀態時具有更高的工作量。

可能有兩個以上的完整儲存庫，但很少建議使用。雖然物件定義 (即佇列、主題及通道) 流向所有可用的完整儲存庫，但要求只會從局部儲存庫流向最多兩個完整儲存庫。這表示當定義超過兩個完整儲存庫，且任何兩個完整儲存庫變成無法使用時，部分局部儲存庫可能不會收到他們預期的更新項目。請參閱 [MQ 叢集: 為何只有兩個完整儲存庫?](#)

當您將現有的完整儲存庫移轉至新的硬體或新的佇列管理程式時，可能會發現定義兩個以上完整儲存庫會很有用的一種狀況。在此情況下，您應該先引進取代完整儲存庫，並確認它們已完全移入，然後再移除先前的完整儲存庫。每當您新增完整儲存庫時，請記住必須使用 CLUSSDR 通道將它直接連接至所有其他完整儲存庫。

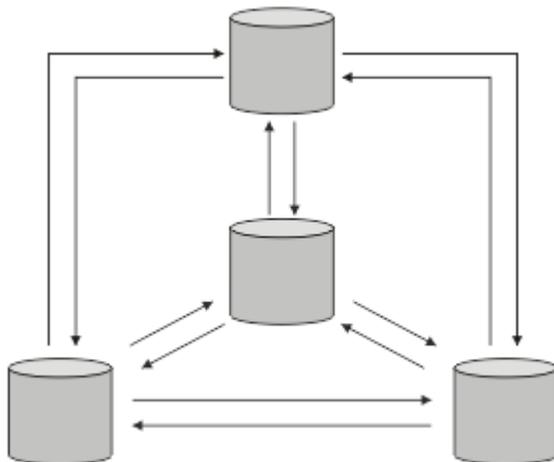


圖 6: 超過兩個已連接的完整儲存庫

相關資訊

[MQ 叢集: 為何只有兩個完整儲存庫?](#)

[「MQ 叢集」可以有多大?](#)

組織叢集

選取要鏈結至哪個完整儲存庫的佇列管理程式。請考量效能影響、佇列管理程式版本，以及是否需要多個 CLUSSDR 通道。

選取佇列管理程式來保留完整儲存庫之後，您需要決定要將哪些佇列管理程式鏈結至哪個完整儲存庫。CLUSSDR 通道定義會將佇列管理程式鏈結至完整儲存庫，並從中瞭解叢集中的其他完整儲存庫。從那時開始，佇列管理程式會將訊息傳送至任何兩個完整儲存庫。它一律會先嘗試使用其具有 CLUSSDR 通道定義的通道。您可以選擇將佇列管理程式鏈結至任一完整儲存庫。在選擇時，請考量配置的拓撲，以及佇列管理程式的實體或地理位置。

因為所有叢集資訊都會傳送至兩個完整儲存庫，所以在某些情況下，您可能想要建立第二個 CLUSSDR 通道定義。您可以在叢集中定義第二個 CLUSSDR 通道，其具有許多分佈於廣域範圍的完整儲存庫。然後，您可以控制要將資訊傳送至哪兩個完整儲存庫。

叢集命名慣例

請考慮使用識別佇列管理程式所屬叢集的命名慣例，來命名相同叢集中的佇列管理程式。通道名稱使用類似的命名慣例，並延伸它來說明通道性質。  在 z/OS 上定義叢集接收端通道時不要使用一般連線。

此資訊包含關於命名慣例的舊指引，以及現行指引。隨著 IBM MQ 技術的改善，以及隨著客戶以新的或不同的方式使用技術，必須針對這些實務範例提供新的建議及資訊。

叢集命名慣例: 現行最佳作法

良好的命名慣例有助於減少對叢集所有權及範圍的混淆。如果稍後合併叢集，則整個叢集拓撲中的明確命名慣例會造成較少的混淆。如果涉及的每個人都清楚擁有哪些佇列管理程式及哪些叢集，則也會改善此狀況。叢集命名慣例最重要的點可能是將佇列管理程式名稱放入通道名稱中，請參閱下列範例：

CLUSNAME.QMGRNAME

對於不熟悉叢集的有經驗 IBM MQ 使用者而言，此慣例可能並不明顯。此監督是因為 XXX.TO.YYY 格式是這類常見的方法。例如，CLUSTER.TO.XX 或 CLUSTER.X 是不建議用於叢集作業的常用格式，因為它們可以快速達到 20 個字元的限制。如果稍後新增另一個通道 (例如加入另一個叢集時)，則常用的 CLUSTER.TO.XX 格式會變得混淆。

其他物件也受益於可感應的規則，例如: LOB.PROJECT.QNAME 或 LOB.CLUSTER.ALIAS.NAME。

叢集命名慣例: 舊的最佳作法

設定新叢集時，請考量佇列管理程式的命名慣例。每個佇列管理程式都必須有不同的名稱。如果您為叢集中的佇列管理程式提供一組類似名稱，則可協助您記住將哪些佇列管理程式分組在何處。

定義通道時，請記住所有叢集傳送端通道與其對應的叢集接收端通道具有相同的名稱。通道名稱限制為最多 20 個字元。

每個叢集接收端通道也必須具有唯一名稱。其中一種可能是使用前面加上叢集名稱的佇列管理程式名稱。例如，如果叢集名稱為 CLUSTER1，且佇列管理程式為 QM1、QM2，則叢集接收端通道為 CLUSTER1.QM1、CLUSTER1.QM2。

如果通道具有不同的優先順序或使用不同的通訊協定 (例如，CLUSTER1.QM1.S1、CLUSTER1.QM1.N3 及 CLUSTER1.QM1.T4)，您可以延伸此慣例。在此範例中，S1 可能是第一個 SNA 通道，N3 可能是網路優先順序為 3 的 NetBIOS 通道。

最終限定元可以說明通道提供的服務類別。

 在 IBM MQ for z/OS 中，您可以定義 VTAM 通用資源或動態網域名稱伺服器 (DDNS) 通用名稱。您可以使用通用名稱來定義連線名稱。不過，當您建立叢集接收端定義時，請勿使用一般連線名稱。

叢集接收端定義使用同屬連線名稱的問題如下。如果您使用通用 CONNAME 定義 CLUSRCVR，則不保證您的 CLUSSDR 通道會指向您想要的佇列管理程式。您的起始 CLUSSDR 可能最終指向佇列共用群組中的任何佇列管理程式，但不一定指向管理完整儲存庫的佇列管理程式。如果通道再次開始嘗試連線，它可能會重新連接至具有相同通用名稱的不同佇列管理程式，從而中斷訊息流程。

z/OS 佇列共用群組及叢集

共用佇列可以是叢集佇列，佇列共用群組中的佇列管理程式也可以是叢集佇列管理程式。

在「IBM MQ for z/OS」上，您可以將佇列管理程式分組成佇列共用群組。佇列共用群組中的佇列管理程式可以定義要由最多 32 個佇列管理程式共用的本端佇列。

共用佇列也可以是叢集佇列。此外，佇列共用群組中的佇列管理程式也可以位於一個以上叢集中。

在 IBM MQ for z/OS 中，您可以定義 VTAM 通用資源或動態網域名稱伺服器 (DDNS) 通用名稱。您可以使用通用名稱來定義連線名稱。不過，當您建立叢集接收端定義時，請勿使用一般連線名稱。

叢集接收端定義使用同屬連線名稱的問題如下。如果您使用通用 CONNAME 定義 CLUSRCVR，則不保證您的 CLUSSDR 通道會指向您想要的佇列管理程式。您的起始 CLUSSDR 可能最終指向佇列共用群組中的任何佇列管理程式，但不一定指向管理完整儲存庫的佇列管理程式。如果通道再次開始嘗試連線，它可能會重新連接至具有相同通用名稱的不同佇列管理程式，從而中斷訊息流程。

無法啟動使用群組接聽器埠的 CLUSRCVR 通道，因為如果發生這種情況，則無法分辨 CLUSRCVR 將每次連接至哪個佇列管理程式。在其上保留叢集相關資訊的叢集系統佇列不會共用。每一個佇列管理程式都有自己的專屬佇列管理程式。

叢集通道不僅用來傳送應用程式訊息，還用來傳送關於叢集設定的內部系統訊息。叢集中的每一個佇列管理程式都必須接收這些內部系統訊息，才能適當地參與叢集作業，因此需要自己的唯一 CLUSRCVR 通道來接收它們。

共用 CLUSRCVR 可以在佇列共用群組 (QSG) 中的任何佇列管理程式上啟動，因此會導致提供不一致的內部系統訊息給 QSG 佇列管理程式，這表示沒有任何佇列管理程式可以適當地參與叢集。為了確保無法使用任何共用 CLUSRCVR 通道，任何嘗試都會失敗，並出現 `CSQX502E` 訊息。

重疊叢集

重疊叢集提供其他管理功能。使用名稱清單來減少管理重疊叢集所需的指令數目。

您可以建立重疊的叢集。您可以定義重疊叢集的原因有很多；例如：

- 容許不同組織擁有自己的管理。
- 容許個別管理獨立應用程式。
- 建立服務類別。

在第 28 頁的圖 7 中，佇列管理程式 STF2 是這兩個叢集的成員。當佇列管理程式是多個叢集的成員時，您可以利用名稱清單來減少您需要的定義數目。名稱清單包含名稱清單，例如叢集名稱。您可以建立命名叢集的名單。在 STF2 的 ALTER QMGR 指令上指定名稱清單，使它成為兩個叢集的完整儲存庫佇列管理程式。

如果網路中有多個叢集，則必須為它們提供不同的名稱。如果曾經合併兩個同名的叢集，則無法再次將它們分開。提供不同的名稱給叢集和通道也是不錯的建議。當您查看 DISPLAY 指令的輸出時，會更容易識別它們。佇列管理程式名稱在叢集內必須是唯一的，才能正確運作。

定義服務類別

想像一下，有一所大學，每一個教職員和學生都有一個佇列管理程式。員工之間的訊息是在高優先順序及高頻寬的通道上傳送。學生之間的訊息是透過更便宜、更慢的管道傳遞。您可以使用傳統分散式佇列技術來設定此網路。IBM MQ 會查看目的地佇列名稱及佇列管理程式名稱，以選取要使用的通道。

若要清楚區分人員與學生，您可以將他們的佇列管理程式分組為兩個叢集，如第 28 頁的圖 7 中所示。IBM MQ 只會透過該叢集中定義的通道，將訊息移至人員叢集中的會議佇列。學生叢集中 Gossip 佇列的訊息會透過該叢集中定義的通道，並接收適當的服務類別。

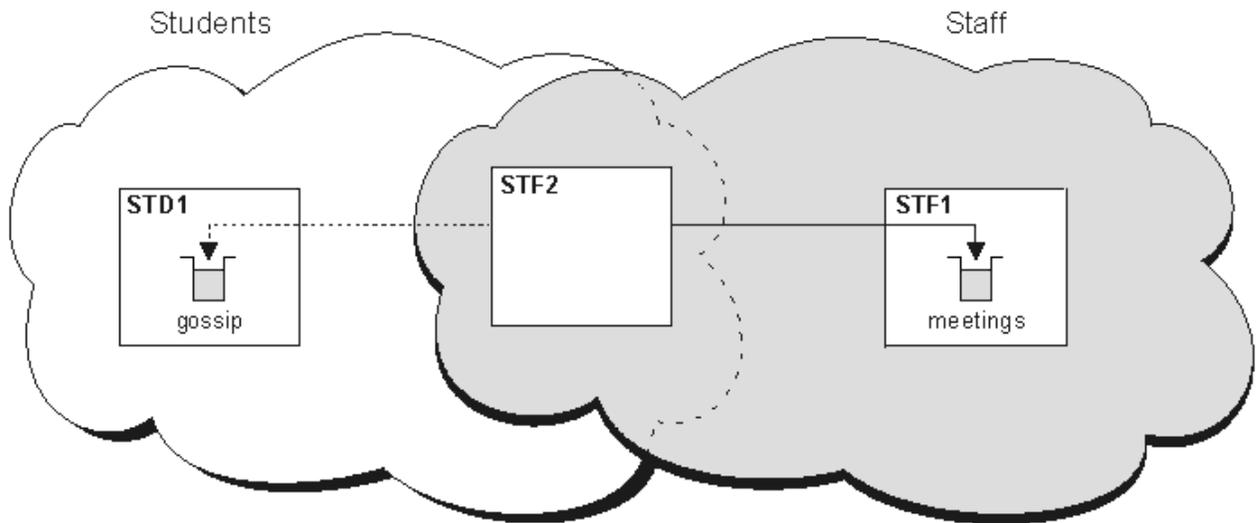


圖 7: 服務類別

叢集提示

在使用叢集作業之前，您可能需要對系統或應用程式進行一些變更。與分散式佇列的行為有相似性和差異。

- z/OS 「IBM MQ 探險家」無法直接管理 6.0 版之前版本的 IBM MQ for z/OS 佇列管理程式。
 - 您必須將手動配置定義新增至叢集外的佇列管理程式，它們才能存取叢集佇列。
 - 如果您合併兩個同名叢集，則無法再次區隔它們。因此，建議為所有叢集提供唯一名稱。
 - 如果訊息到達佇列管理程式，但沒有佇列可接收它，則會將訊息放置在無法傳送郵件的佇列上。如果沒有無法傳送郵件的佇列，通道會失敗並重試。無法傳送郵件的佇列與分散式佇列的使用相同。
 - 會維護持續訊息的完整性。由於使用叢集，訊息不會重複或遺失。
 - 使用叢集可減少系統管理。叢集可讓您輕鬆連接更大的網路與比您可以考慮使用分散式佇列作業更多的佇列管理程式。如果您嘗試啟用叢集中每個佇列管理程式之間的通訊，則可能會有耗用過多網路資源的風險。
 - 如果您使用以樹狀結構呈現佇列管理程式的「IBM MQ 探險家」，則大型叢集的視圖可能會很麻煩。
- z/OS 「IBM MQ 探險家」可以在 IBM MQ for z/OS 第 6 版或更新版本上管理具有儲存庫佇列管理程式的叢集。您不需要在個別系統上指定其他儲存庫。對於 z/OS 上舊版 IBM MQ，「IBM MQ 探險家」無法管理具有儲存庫佇列管理程式的叢集。您必須在 IBM MQ Explorer 可以管理的系統上提名其他儲存庫。
 - 配送清單的目的是使用單一 MQPUT 指令，將相同的訊息傳送至多個目的地。IBM MQ for AIX、IBM i、HP-UX、Solaris、Linux 及 Windows 支援配送清單。您可以將配送清單與佇列管理程式叢集搭配使用。在叢集中，所有訊息都會在 MQPUT 時展開。就網路資料流量而言，其優點不如非叢集作業環境中的優點。配送清單的優點是不需要手動定義許多通道和傳輸佇列。
 - 如果您要使用叢集來平衡工作量，請檢查應用程式。請查看它們是否需要由特定佇列管理程式或以特定順序處理訊息。這類應用程式據說具有訊息親緣性。您可能需要先修改應用程式，然後才能在複式叢集中使用它們。
 - 您可以選擇在 MQOPEN 上使用 MQOO_BIND_ON_OPEN 選項，以強制將訊息傳送至特定目的地。如果目的地佇列管理程式無法使用，則在佇列管理程式重新變成可用之前，不會遞送訊息。由於有重複的風險，訊息不會遞送至另一個佇列管理程式。
 - 如果佇列管理程式要管理叢集儲存庫，您需要知道其主機名稱或 IP 位址。當您在其他加入叢集的佇列管理程式上定義 CLUSSDR 時，必須在 CONNAME 參數中指定此資訊。如果您使用 DHCP，IP 位址可能會變更，因為 DHCP 可以在每次重新啟動系統時配置新的 IP 位址。因此，您不得在 CLUSSDR 定義中指定 IP 位址。即使所有 CLUSSDR 定義都指定主機名稱而非 IP 位址，這些定義仍不可靠。DHCP 不一定會以新位址來更新主機的 DNS 目錄項目。如果您必須在使用 DHCP 的系統上提名佇列管理程式作為完整儲存庫，請安裝可保證 DNS 目錄保持最新的軟體。

- 請勿使用一般名稱，例如 VTAM 一般資源或「動態網域名稱伺服器 (DDNS)」一般名稱作為通道的連線名稱。如果您這麼做，您的通道可能會連接至非預期的佇列管理程式。
- 您只能從本端叢集佇列取得訊息，但可以將訊息放入叢集中的任何佇列。如果您開啟佇列以使用 MQGET 指令，則佇列管理程式會開啟本端佇列。
- 如果您設定簡式 IBM MQ 叢集，則不需要變更任何應用程式。應用程式可以在 MQOPEN 呼叫上命名目標佇列，且不需要知道佇列管理程式的位置。如果您設定叢集以進行工作量管理，則必須檢閱您的應用程式，並視需要修改它們。
- 您可以使用 DISPLAY CHSTATUS 及 DISPLAY QSTATUS **runmqsc** 指令，來檢視通道或佇列的現行監視及狀態資料。監視資訊可用來協助測量系統的效能及性能。監視是由佇列管理程式、佇列及通道屬性所控制。可以使用 MONACLS 佇列管理程式屬性來監視自動定義的叢集傳送端通道。

相關概念

第 23 頁的『叢集作業與分散式佇列作業的比較』

比較需要定義的元件，以使用分散式佇列及叢集作業來連接佇列管理程式。

相關資訊

[叢集](#)

[配置佇列管理程式叢集](#)

[叢集的元件](#)

[設定新的叢集](#)

佇列管理程式儲存庫保留資訊的時間長度？

佇列管理程式儲存庫會保留資訊 30 天。自動處理程序可有效地重新整理正在使用的資訊。

當佇列管理程式送出一些本身的相關資訊時，完整和局部儲存庫佇列管理程式會儲存 30 天的資訊。例如，當佇列管理程式通告建立新佇列時，即會送出資訊。為了防止此資訊到期，佇列管理程式會在 27 天之後自動重新傳送其本身的所有相關資訊。如果局部儲存庫在 30 天生命期限期間傳送新的資訊要求，則到期時間會保留原始 30 天。

當資訊到期時，它不會立即從儲存庫中移除。相反地，它會保留 60 天的寬限期。如果在寬限期內未收到任何更新，則會移除資訊。寬限期容許下列事實：佇列管理程式可能在到期日暫時無法運作。如果佇列管理程式與叢集中斷連線超過 90 天，則會停止成為叢集的一部分。不過，如果它重新連接至網路，則它會再次成為叢集的一部分。完整儲存庫不會使用已過期的資訊來滿足其他佇列管理程式的新要求。

同樣地，當佇列管理程式從完整儲存庫傳送最新資訊的要求時，該要求會持續 30 天。在 27 天之後 IBM MQ 會檢查要求。如果在 27 天期間已參照它，則會自動重新整理它。如果沒有，則會讓它到期，並由佇列管理程式重新整理 (如果再次需要的話)。到期要求可防止建立來自休眠佇列管理程式的資訊要求。

註：對於大型叢集，如果許多佇列管理程式同時自動重新傳送其本身的所有相關資訊，則會造成干擾。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。

相關資訊

[叢集作業：使用 REFRESH CLUSTER 最佳作法](#)

範例叢集

第一個範例顯示兩個佇列管理程式的最小可能叢集。第二個和第三個範例顯示三個佇列管理程式叢集的兩個版本。

最小的可能叢集只包含兩個佇列管理程式。在此情況下，兩個佇列管理程式都包含完整儲存庫。您只需要幾個定義來設定叢集，但每一個佇列管理程式都有高度自主。

DEMOCLSTR

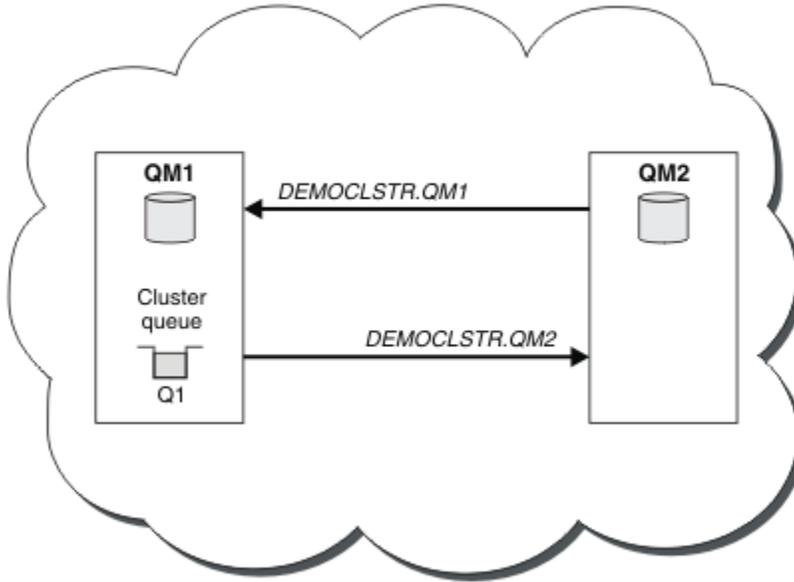


圖 8: 由兩個佇列管理程式組成的小型叢集

- 佇列管理程式可以有完整名稱，例如 LONDON 和 NEWYORK。在 IBM MQ for z/OS 上，佇列管理程式名稱限制為四個字元。
 - 每一個佇列管理程式通常配置在個別機器上。不過，您可以在相同機器上具有多個佇列管理程式。
- 如需設定類似範例叢集的指示，請參閱 [設定新叢集](#)。

第 31 頁的圖 9 顯示稱為 CLSTR1 之叢集的元件。

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

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

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

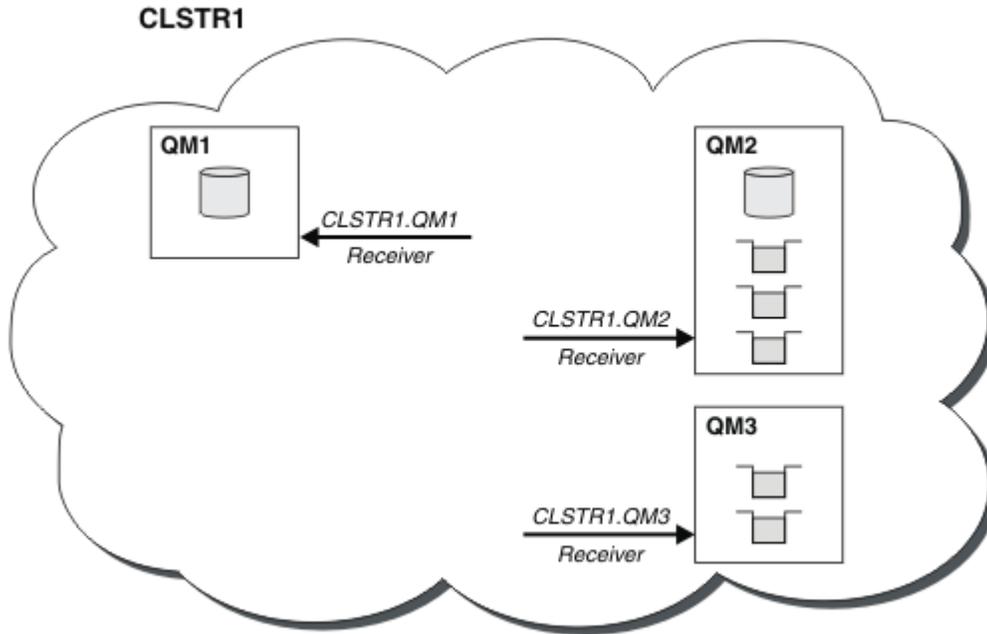


圖 9: 佇列管理程式的叢集

- 在第 31 頁的圖 10 中，每一個佇列管理程式也都有通道傳送端的定義。它會連接至其中一個完整儲存庫佇列管理程式的叢集接收端通道。在傳送端佇列管理程式上，*cluster-name.queue-manager* 是叢集傳送端通道。QM1 及 QM3 具有連接至 CLSTR1.QM2 的叢集傳送端通道，請參閱點虛線 "2"。

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

一旦定義了通道的叢集接收端和叢集傳送端，通道會自動啟動。

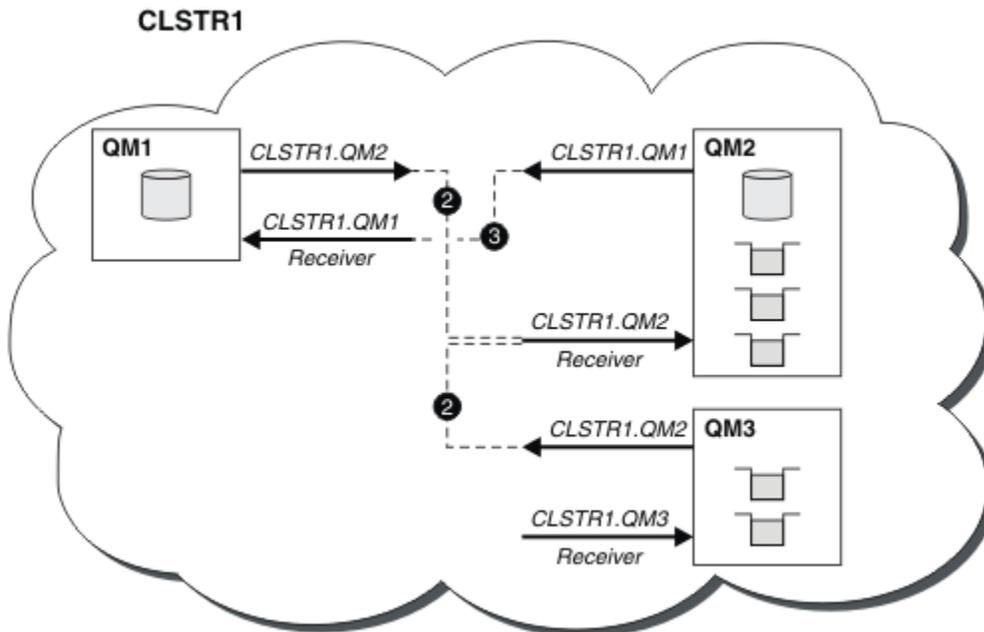


圖 10: 具有傳送端通道的佇列管理程式叢集

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

的叢集傳送端通道，並傳送該佇列管理程式的叢集相關資訊。因此，佇列管理程式會學習叢集，而叢集會學習佇列管理程式。

再次查看第 31 頁的圖 9。假設連接至佇列管理程式 QM3 的應用程式想要將部分訊息傳送至 QM2 中的佇列。QM3 第一次必須存取這些佇列時，它會透過諮詢完整儲存庫來探索這些佇列。在此情況下，完整儲存庫是 QM2，可使用傳送端通道 CLSTR1.QM2 來存取。利用儲存庫中的資訊，它可以自動建立那些佇列的遠端定義。如果佇列位於 QM1 上，則此機制仍可運作，因為 QM2 是完整儲存庫。完整儲存庫具有叢集中所有物件的完整記錄。在後一種情況下，QM3 也會自動建立叢集傳送端通道（對應於 QM1 上的叢集接收端通道），以容許兩者之間直接通訊。

第 32 頁的圖 11 顯示相同的叢集，以及兩個自動建立的叢集傳送端通道。叢集傳送端通道由與叢集接收端通道 CLSTR1.QM3 結合的兩條虛線代表。它也會顯示 QM1 用來傳送其訊息的叢集傳輸佇列 SYSTEM.CLUSTER.TRANSMIT.QUEUE。叢集中的所有佇列管理程式都有一個叢集傳輸佇列，它們可以從中將訊息傳送至相同叢集中的任何其他佇列管理程式。

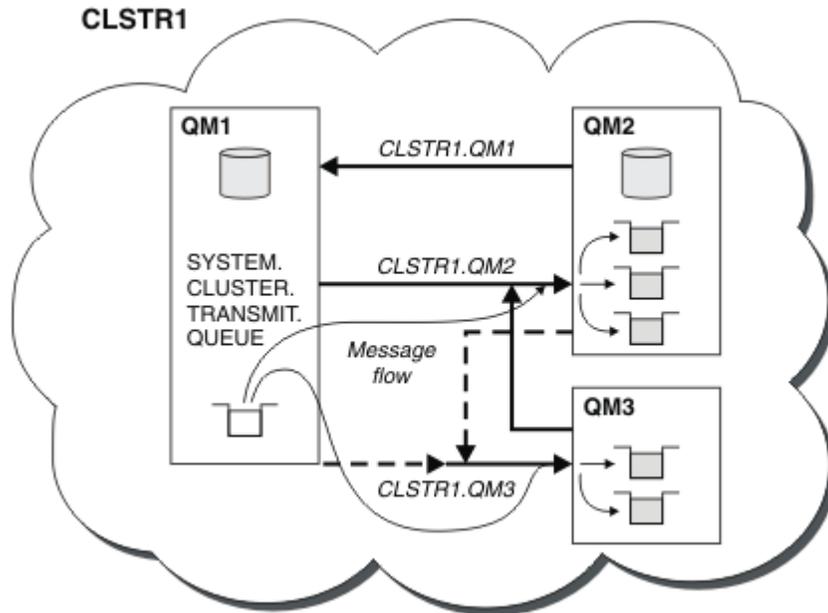


圖 11: 佇列管理程式的叢集，顯示自動定義的通道

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

相關概念

第 23 頁的『叢集作業與分散式佇列作業的比較』

比較需要定義的元件，以使用分散式佇列及叢集作業來連接佇列管理程式。

相關資訊

[配置佇列管理程式叢集](#)

[叢集的元件](#)

[設定新的叢集](#)

叢集作業: 最佳作法

叢集提供交互連接佇列管理程式的機制。本節中說明的最佳作法是根據客戶的測試和意見。

成功的叢集設定取決於良好的規劃，以及對 IBM 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。在繼續之前，請確定您熟悉相關主題中的資訊。

相關工作

第 17 頁的『設計叢集』

叢集提供一種機制，以簡化起始配置及進行中管理的方式來交互連接佇列管理程式。叢集必須小心設計，以確保它們正確運作，並達到所需的可用性和回應性層次。

相關資訊

[分散式佇列及叢集](#)

[叢集](#)

[監視叢集](#)

叢集作業: 重疊叢集的特殊考量

本主題提供規劃及管理 IBM 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。

- [第 33 頁的『叢集所有權』](#)
- [第 33 頁的『重疊叢集: 閘道』](#)
- [第 34 頁的『叢集命名慣例』](#)

叢集所有權

閱讀下列資訊之前，請先熟悉重疊的叢集。如需必要資訊，請參閱 [第 27 頁的『重疊叢集』](#) 及 [在叢集之間配置訊息路徑](#)。

配置及管理由重疊叢集組成的系統時，最好遵循下列各項：

- 雖然如先前所述，IBM MQ 叢集是「鬆散耦合」，但將叢集視為單一管理單元非常有用。使用此概念是因為個別佇列管理程式上定義之間的互動對於叢集順利運作至關重要。例如：使用工作量平衡叢集佇列時，請務必讓單一管理者或團隊瞭解訊息的完整可能目的地集，這取決於分散在整個叢集中的定義。更重要的是，叢集傳送端/接收端通道配對在整個中必須相容。
- 考量此先前的概念；當多個叢集符合（由個別團隊/個人管理）時，必須有明確的原則來控制閘道佇列管理程式的管理。
- 將重疊叢集視為單一名稱空間非常有用：通道名稱和佇列管理程式名稱在單一叢集中必須是唯一的。在整個拓撲中唯一時，管理更容易。最好遵循適當的命名慣例，[第 34 頁的『叢集命名慣例』](#) 中會說明可能的使用慣例。
- 有時行政和系統管理合作是必不可少/不可避免的：例如，擁有需要重疊的不同群組的組織之間的合作。清楚瞭解誰擁有哪些可強制執行的規則/慣例，有助於在重疊叢集時順利執行叢集作業。

重疊叢集: 閘道

一般而言，單一叢集比多個叢集更容易管理。因此，通常需要避免建立大量小型叢集（例如，每個應用程式都有一個叢集）。

不過，若要提供服務類別，您可以實作重疊叢集。例如：

- 同心叢集，其中較小的叢集用於「發佈/訂閱」。如需相關資訊，請參閱 [如何調整系統大小](#)。
- 部分佇列管理程式將由不同的團隊管理（請參閱 [第 33 頁的『叢集所有權』](#)）。
- 從組織或地理角度而言是否有意義。
- 相當於在現有叢集中實作 SSL（或 TLS）時使用名稱解析的叢集。

重疊叢集沒有安全好處；容許由兩個不同團隊管理的叢集重疊，有效地結合團隊及拓撲。任何：

- 在此類叢集中通告的名稱可供其他叢集存取。
- 在一個叢集中通告的名稱可以在另一個叢集中通告，以取得合格訊息。
- 可從閘道所屬的任何叢集解析與閘道相鄰之佇列管理程式上的非通告物件。

名稱空間是兩個叢集的聯集，必須視為單一名稱空間。因此，重疊叢集的所有權會在兩個叢集的所有管理者之間共用。

當系統包含多個叢集時，可能需要將訊息從一個叢集中的佇列管理程式遞送至另一個叢集中佇列管理程式上的佇列。在此情況下，多個叢集必須以某種方式交互連接：接下來要遵循的良好型樣是在叢集之間使用閘道佇列管理程式。此安排可避免建立難以管理的點對點通道網絡，並為管理安全原則之類的問題提供一個好地方。實現這一安排有兩種不同的方式：

1. 使用第二個叢集接收端定義，將一個 (或多個) 佇列管理程式放在兩個叢集中。此安排涉及較少的管理定義，但如先前所述，表示在兩個叢集的所有管理者之間共用重疊叢集的所有權。
2. 使用傳統點對點通道，將叢集 1 中的佇列管理程式與叢集 2 中的佇列管理程式配對。

在上述任一情況下，都可以使用各種工具來適當地遞送資料流量。特別是，佇列或佇列管理程式別名可以用來遞送至其他叢集，而佇列管理程式別名 (具有空白 **RQMNAME** 內容) 會在需要時重新驅動工作量平衡。

叢集命名慣例

此資訊包含先前的命名慣例指引，以及現行指引。隨著 IBM MQ 技術的改善，以及隨著客戶以新的或不同的方式使用技術，必須針對這些實務範例提供新的建議及資訊。

叢集命名慣例: 先前的指引

設定新叢集時，請考量佇列管理程式的命名慣例。每個佇列管理程式都必須有不同的名稱，但如果您提供一組類似名稱，則可協助您記住哪些佇列管理程式分組在何處。

每個叢集接收端通道也必須具有唯一名稱。

如果您有多個通道連接至相同的佇列管理程式 (每一個都具有不同的優先順序或使用不同的通訊協定)，您可以延伸名稱以包括不同的通訊協定; 例如 **QM1.S1**、**QM1.N3** 及 **QM1.T4**。在此範例中，**S1** 可能是第一個 SNA 通道，**N3** 可能是網路優先順序為 3 的 NetBIOS 通道。

最終限定元可能說明通道提供的服務類別。如需相關資訊，請參閱 [定義服務類別](#)。

請記住，所有叢集傳送端通道的名稱都與其對應的叢集接收端通道相同。

請勿在叢集接收端定義上使用通用連線名稱。在 IBM MQ for z/OS 中，您可以定義 VTAM 通用資源或動態網域名稱伺服器 (DDNS) 通用名稱，但如果您使用叢集，則不要這麼做。如果您使用一般 **CONNAME** 來定義 **CLUSRCVR**，則不保證您的 **CLUSDR** 通道會指向您想要的佇列管理程式。您的起始 **CLUSDR** 可能最終指向佇列共用群組中的任何佇列管理程式，但不一定指向管理完整儲存庫的佇列管理程式。此外，如果通道進入重試狀態，則可能會重新連接至具有相同通用名稱的不同佇列管理程式，且訊息流程會中斷。

叢集命名慣例: 現行指引

第 34 頁的『叢集命名慣例: 先前的指引』一節中的先前指引仍然有效。不過，下列指引旨在作為設計新叢集時的更新。這項更新的建議可確保通道在多個叢集之間的唯一性，讓多個叢集順利重疊。因為佇列管理程式及叢集最多可以有 48 個字元的名稱，且通道名稱限制為 20 個字元，所以在從頭開始命名物件時必須小心，以避免在中途透過專案變更命名慣例。

設定新叢集時，請考量佇列管理程式的命名慣例。每個佇列管理程式都必須有不同的名稱。如果您為叢集中的佇列管理程式提供一組類似名稱，則可協助您記住將哪些佇列管理程式分組在何處。

定義通道時，請記住叢集中任何佇列管理程式上所有自動建立的叢集傳送端通道，其名稱都與叢集中接收端佇列管理程式上所配置的對應叢集接收端通道相同，因此必須是唯一的，且在叢集中對該叢集的管理者有意義。通道名稱限制為最多 20 個字元。

一種可能是使用前面有叢集名稱的佇列管理程式名稱。例如，如果叢集名稱是 **CLUSTER1**，且佇列管理程式是 **QM1**、**QM2**，則叢集接收端通道是 **CLUSTER1.QM1**、**CLUSTER1.QM2**。

如果通道具有不同的優先順序或使用不同的通訊協定 (例如，**CLUSTER1.QM1.S1**、**CLUSTER1.QM1.N3** 及 **CLUSTER1.QM1.T4**)，您可以延伸此慣例。在此範例中，**S1** 可能是第一個 SNA 通道，**N3** 可能是網路優先順序為 3 的 NetBIOS 通道。

最終限定元可以說明通道提供的服務類別。

IBM MQ for z/OS 考量



在 IBM MQ for z/OS 中，您可以定義 VTAM 通用資源或動態網域名稱伺服器 (DDNS) 通用名稱。您可以使用通用名稱來定義連線名稱。不過，當您建立叢集接收端定義時，請勿使用一般連線名稱。

叢集接收端定義使用同屬連線名稱的問題如下。如果您使用通用 CONNAME 定義 CLUSRCVR，則不保證您的 CLUSSDR 通道會指向您想要的佇列管理程式。您的起始 CLUSSDR 可能最終指向佇列共用群組中的任何佇列管理程式，但不一定指向管理完整儲存庫的佇列管理程式。如果通道再次開始嘗試連線，它可能會重新連接至具有相同通用名稱的不同佇列管理程式，從而中斷訊息流程。

叢集作業: 拓撲設計考量

本主題提供規劃及管理 IBM 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。

透過預先考慮使用者應用程式及內部管理處理程序的位置，可以避免許多問題，或在稍後日期將問題減至最少。本主題包含設計決策的相關資訊，這些設計決策可以改善效能，並在叢集調整時簡化維護作業。

- [第 35 頁的『叢集作業基礎架構的效能』](#)
- [第 35 頁的『完整儲存庫』](#)
- [第 36 頁的『應用程式應該在完整儲存庫上使用佇列嗎?』](#)
- [第 36 頁的『管理通道定義』](#)
- [第 37 頁的『多個通道的工作量平衡』](#)

叢集作業基礎架構的效能

當應用程式嘗試在叢集中的佇列管理程式上開啟佇列時，佇列管理程式會向該佇列的完整儲存庫登錄其興趣，以便它可以瞭解佇列在叢集中的位置。完整儲存庫會自動將佇列位置或配置的任何更新項目傳送至感興趣的佇列管理程式。此感興趣登錄在內部稱為訂閱（這些訂閱與 IBM MQ 中用於發佈/訂閱傳訊的 IBM MQ 訂閱不同）

叢集的所有相關資訊都會通過每個完整儲存庫。因此，叢集中一律會使用完整儲存庫來進行管理訊息資料流量。管理這些訂閱時系統資源的高用量，以及它們的傳輸和產生的配置訊息，可能會對叢集基礎架構造成相當大的負載。在確保盡可能瞭解並最小化此負載時，有許多事項需要考量：

- 使用叢集佇列的個別佇列管理程式越多，系統中的訂閱越多，因此在發生變更且需要通知感興趣的訂閱者時，管理額外負擔越大，尤其是在完整儲存庫佇列管理程式上。將不必要的資料流量和完整儲存庫負載減至最少的方法之一，是將類似的應用程式（亦即，使用相同佇列的應用程式）連接至較少的佇列管理程式。
- 除了系統中影響效能的訂閱數之外，叢集物件配置中的變更率也會影響效能，例如經常變更叢集佇列配置。
- 當佇列管理程式是多個叢集（亦即，它是重疊叢集系統的一部分）的成員時，對佇列所產生的任何興趣都會導致它所屬每一個叢集的訂閱，即使相同佇列管理程式是多個叢集的完整儲存庫也一樣。此安排會增加系統上的負載，也是考量是否需要多個重疊叢集而非單一叢集的原因之一。
- 應用程式訊息資料流量（亦即，IBM MQ 應用程式傳送至叢集佇列的訊息）不會透過完整儲存庫到達目的地佇列管理程式。此訊息資料流量會直接在訊息進入叢集的佇列管理程式與叢集佇列所在的佇列管理程式之間傳送。因此，除非完整儲存庫佇列管理程式剛好是所提及的這兩個佇列管理程式之一，否則不需要容納與完整儲存庫佇列管理程式相關的高速率應用程式訊息資料流量。因此，建議不要將完整儲存庫佇列管理程式用於叢集作業基礎架構負載很重要的叢集中的應用程式訊息資料流量。

完整儲存庫

儲存庫是屬於叢集成員之佇列管理程式的相關資訊集合。管理叢集中每個佇列管理程式的完整資訊集的佇列管理程式具有完整儲存庫。如需完整儲存庫和局部儲存庫的相關資訊，請參閱 [叢集儲存庫](#)。

完整儲存庫必須保留在可靠且儘可能高可用性的伺服器上，且必須避免單一失敗點。叢集設計必須一律有兩個完整儲存庫。如果完整儲存庫失敗，叢集仍可運作。

叢集中佇列管理程式對叢集資源所做之任何更新的詳細資料；例如，叢集佇列會從該佇列管理程式傳送至該叢集中最多兩個完整儲存庫（如果叢集中只有一個完整儲存庫佇列管理程式，則傳送至一個完整儲存庫）。那些完整儲存庫會保留資訊，並將它延伸到叢集中任何有興趣的佇列管理程式（亦即，它們會訂閱它）。為了確保叢集的每一個成員都具有叢集資源的最新視圖，每一個佇列管理程式必須能夠隨時與至少一個完整儲存庫佇列管理程式進行通訊。

如果佇列管理程式因為任何原因而無法與任何完整儲存庫通訊，則它可以根據已快取的資訊層次在一段時間內繼續在叢集中運作，但沒有可用的新更新項目或對先前未用叢集資源的存取權。

因此，您必須持續保留兩個完整儲存庫。然而，這一安排並不意味著必須採取極端措施，因為叢集在沒有完整儲存庫的情況下在短時間內充分運作。

另一個原因是叢集必須有兩個完整儲存庫佇列管理程式，但叢集資訊的可用性除外：這個原因是為了確保保留在完整儲存庫快取中的叢集資訊存在於兩個位置，以進行回復。如果只有一個完整儲存庫，且遺失叢集的相關資訊，則需要對叢集內的所有佇列管理程式進行人為介入，才能讓叢集重新運作。不過，如果有兩個完整儲存庫，由於一律會將資訊發佈至兩個完整儲存庫，並從兩個完整儲存庫訂閱資訊，因此只要最少努力，就可以回復失敗的完整儲存庫。

- 在兩個完整儲存庫叢集設計中，可以對完整儲存庫佇列管理程式執行維護，而不會影響該叢集的使用者：叢集會繼續只使用一個儲存庫來運作，因此在可能的情況下，請關閉儲存庫，套用維護，然後一次重新備份一個儲存庫。即使第二個完整儲存庫中斷，執行中應用程式也不會受到至少三天的影響。
- 除非有充分理由使用第三個儲存庫，例如基於地理原因使用地理上的本端完整儲存庫，否則請使用兩個儲存庫設計。擁有三個完整儲存庫表示您永遠不知道目前使用中的兩個儲存庫，而且多個工作量管理參數之間的互動可能導致管理問題。不建議有兩個以上完整儲存庫。
- 如果您仍然需要更好的可用性，請考量將完整儲存庫佇列管理程式當作多重實例佇列管理程式來管理，或使用平台特定的高可用性支援來改善其可用性。
- 您必須將所有完整儲存庫佇列管理程式與手動定義的叢集傳送端通道完全交互連接。當叢集有兩個以上的完整儲存庫時，出於合理的原因，必須特別小心。在這種情況下，往往可能錯過一個或多個通道，而無法立即顯現。在未發生完整互連時，經常會出現難以診斷的問題。它們很難診斷，因為部分完整儲存庫並未保留所有儲存庫資料，因此導致叢集中的佇列管理程式具有不同的叢集視圖，視它們所連接的完整儲存庫而定。

應用程式應該在完整儲存庫上使用佇列嗎？

在大部分情況下，完整儲存庫與任何其他佇列管理程式完全相同，因此可以在完整儲存庫上管理應用程式佇列，並將應用程式直接連接至這些佇列管理程式。應用程式應該在完整儲存庫上使用佇列嗎？

普遍接受的答案是「否？」。雖然可以進行此配置，但許多客戶偏好讓這些佇列管理程式專用於維護完整儲存庫叢集快取。這裡說明在決定任一選項時要考量的點，但最終叢集架構必須適合環境的特定需求。

- 升級：通常，為了在 IBM MQ 新版本中使用新的叢集特性，必須先升級該叢集的完整儲存庫佇列管理程式。當叢集中的應用程式想要使用新特性時，如果能夠更新完整儲存庫（以及部分儲存庫的部分子集），而不測試一些並存的應用程式，可能很有用。
- 維護：同樣地，如果您必須將緊急維護套用至完整儲存庫，則可以使用 **REFRESH** 指令重新啟動或重新整理它們，而不需要接觸應用程式。
- 效能：當叢集不斷成長，且對完整儲存庫叢集快取維護的需求越來越高時，讓應用程式分開，可減少因競用系統資源而影響應用程式效能的風險。
- 硬體需求：一般而言，完整儲存庫不需要強大；例如，具有良好可用性預期的簡式 UNIX 伺服器就已足夠。或者，對於非常大或不斷變更的叢集，必須考量完整儲存庫電腦的效能。
- 軟體需求：需求通常是選擇在完整儲存庫上管理應用程式佇列的主要原因。在小型叢集中，並置可能表示所有佇列管理程式/伺服器都需要較少的佇列管理程式/伺服器。

管理通道定義

即使在單一叢集內，也可以存在多個通道定義，提供兩個佇列管理程式之間的多個路徑。

有時在單一叢集內具有平行通道有其優點，但必須徹底考量此設計決策；除了增加複雜性之外，此設計還可能導致通道未充分利用而降低效能。因為測試通常涉及以固定速率傳送大量訊息，所以會完全使用平行通道，所以會發生此狀況。但在訊息串流不固定的現實狀況下，當訊息流程從通道切換至通道時，工作量平衡演算法會導致效能降低。

當佇列管理程式是多個叢集的成員時，會有選項可搭配使用單一通道定義與叢集名單，而不是針對每一個叢集定義個別 CLUSRCVR 通道。不過，這項設定稍後可能會造成管理上的困難；例如，請考量將 SSL 套用至一個叢集而不是套用至第二個叢集的情況。因此，最好建立個別定義，且第 34 頁的『叢集命名慣例』中建議的命名慣例支援此作法。

多個通道的工作量平衡

本資訊旨在作為對主題的進階瞭解。如需此主題的基本說明 (在使用這裡的資訊之前必須先瞭解)，請參閱 [使用叢集進行工作量管理](#)、[叢集中的工作量平衡](#) 及 [叢集工作量管理演算法](#)。

叢集工作量管理演算法提供大量工具集，但它們不能全部一起使用，必須完全瞭解它們如何運作及互動。可能無法立即看出通道對工作量平衡處理程序的重要性：工作量管理循環式演算法的行為如同擁有叢集佇列之佇列管理程式的多個叢集通道被視為該佇列的多個實例。此處理程序在下列範例中有更詳細的說明：

1. 有兩個佇列管理程式管理叢集中的佇列：QM1 及 QM2。
2. QM1 有五個叢集接收端通道。
3. 只有一個叢集接收端通道可連接至 QM2。
4. 當 MQPUT 或 MQOPEN on QM3 選擇實例時，演算法將訊息傳送至 QM1 的可能性比傳送至 QM2 的可能性高五倍。
5. 發生步驟 4 中的狀況是因為演算法會看到六個選項可供選擇 (5 + 1)，並在所有五個通道之間來回轉換至 QM1，而在單一通道中來回轉換至 QM2。

另一個微妙的行為是，即使將訊息放入叢集佇列 (碰巧在本端佇列管理程式上配置了一個實例)，IBM MQ 也會使用本端叢集接收端通道的狀態來決定是否將訊息放入佇列的本端實例或佇列的遠端實例。在此實務範例中：

1. 放置訊息時，工作量管理演算法不會查看個別叢集佇列，它會查看可到達那些目的地的叢集通道。
2. 為了到達本端目的地，本端接收端通道會包含在此清單中 (雖然它們不是用來傳送訊息)。
3. 當本端接收端通道停止時，如果 CLUSRCVR 未停止，依預設，工作量管理演算法會偏好替代實例。如果目的地有多個本端 CLUSRCVR 實例，且至少有一個未停止，則本端實例仍然適用。

叢集作業：使用多個叢集傳輸佇列的應用程式隔離

您可以隔離叢集中佇列管理程式之間的訊息流程。您可以將由不同叢集傳送端通道傳輸的訊息放在不同的叢集傳輸佇列中。您可以在單一叢集或重疊叢集中使用此方法。本主題提供範例及一些最佳作法，指引您選擇要使用的方法。

當您部署應用程式時，您可以選擇它與其他應用程式共用哪些 IBM MQ 資源，以及它不共用哪些資源。有許多可以共用的資源類型，主要資源是伺服器本身、佇列管理程式、通道及佇列。您可以選擇使用較少的共用資源來配置應用程式；將個別佇列、通道、佇列管理程式，甚至伺服器配置給個別應用程式。如果這樣做，則整體系統配置會變得更大且更複雜。使用 IBM MQ 叢集可減少管理更多伺服器、佇列管理程式、佇列及通道的複雜性，但它會引進另一個共用資源，即叢集傳輸佇列 SYSTEM.CLUSTER.TRANSMIT.QUEUE。

第 38 頁的圖 12 是大型 IBM MQ 部署的截塊，說明共用 SYSTEM.CLUSTER.TRANSMIT.QUEUE 的重要性。在圖表中，應用程式 Client App 連接至 QM2 叢集中 CL1 的佇列管理程式。應用程式 Server App 會處理來自 Client App 的訊息。此訊息由 Server App 從 CLUSTER2 中的佇列管理程式 QM3 上的叢集佇列 Q1 擷取。因為用戶端與伺服器應用程式不在相同的叢集中，所以閘道佇列管理程式 QM1 會傳送訊息。

配置叢集閘道的正常方式是讓閘道佇列管理程式成為所有叢集的成員。在閘道佇列管理程式上，為所有叢集中的叢集佇列定義叢集別名佇列。叢集佇列別名可在所有叢集中使用。放入叢集佇列別名的訊息會透過閘道佇列管理程式遞送至其正確的目的地。閘道佇列管理程式會將傳送至叢集別名佇列的訊息放入 QM1 上的一般 SYSTEM.CLUSTER.TRANSMIT.QUEUE。

中心及分支架構需要叢集之間的所有訊息都通過閘道佇列管理程式。結果是所有訊息都流經 QM1 SYSTEM.CLUSTER.TRANSMIT.QUEUE 上的單一叢集傳輸佇列。

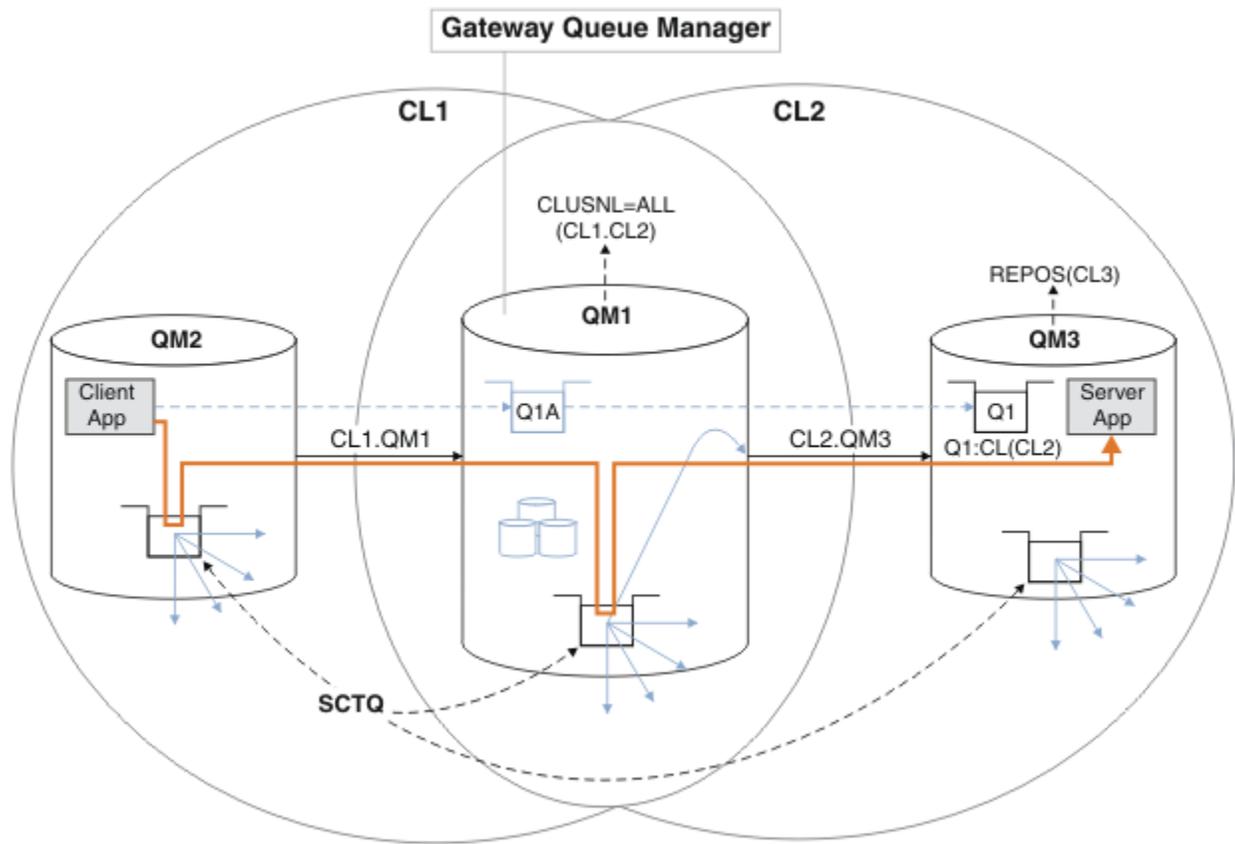
從效能角度來看，單一佇列不是問題。一般傳輸佇列通常不代表效能瓶頸。閘道上的訊息傳輸量很大程度上取決於與其連接之通道的效能。傳輸量通常不受佇列數目或使用通道之佇列上的訊息數目影響。

從某些其他角度來看，使用單一傳輸佇列來處理多個應用程式有一些缺點：

- 您無法將傳送至一個目的地的訊息流程與傳送至另一個目的地的訊息流程隔離。在轉遞訊息之前，即使目的地位於不同佇列管理程式上的不同叢集中，您也無法分隔訊息的儲存體。

如果一個叢集目的地變成無法使用，則單一傳輸佇列中該目的地建置的訊息，最終會填滿該訊息。一旦傳輸佇列已滿，就會停止將訊息放在任何叢集目的地的傳輸佇列中。

- 監視將訊息傳送至不同叢集目的地並不容易。所有訊息都在單一傳輸佇列上。顯示傳輸佇列的深度，可讓您很少指示是否將訊息傳送至所有目的地。



註: 第 38 頁的圖 12 及下圖中的箭頭屬於不同類型。實心箭頭代表訊息流程。實心箭頭上的標籤是訊息通道名稱。灰色實心箭頭是從 SYSTEM.CLUSTER.TRANSMIT.QUEUE 到叢集傳送端通道的潛在訊息流程。黑色虛線會將標籤連接至其目標。灰色虛線箭頭是參照; 例如從 MQOPEN Client App 呼叫叢集別名佇列定義 Q1A。

圖 12: 使用 IBM MQ 叢集部署至中心及分支架構的主從架構應用程式

在第 38 頁的圖 12 中, Server App 的用戶端開啟佇列 Q1A。訊息會放置到 QM2 上的 SYSTEM.CLUSTER.TRANSMIT.QUEUE, 傳送到 QM1 上的 SYSTEM.CLUSTER.TRANSMIT.QUEUE, 然後傳送到 QM3 上的 Q1, 在這裡由 Server App 應用程式接收。

來自 Client App 的訊息會通過 QM2 和 QM1 上的系統叢集傳輸佇列。在第 38 頁的圖 12 中, 目標是將閘道佇列管理程式上的訊息流程與用戶端應用程式隔離, 以便其訊息不會儲存在 SYSTEM.CLUSTER.TRANSMIT.QUEUE 上。您可以隔離任何其他叢集佇列管理程式上的流程。您也可以從另一個方向隔離流程, 回到用戶端。為了使解決方案的說明保持簡短, 說明僅考量來自用戶端應用程式的單一流程。

隔離叢集閘道佇列管理程式上叢集訊息資料流量的解決方案

解決此問題的方法之一是使用佇列管理程式別名或遠端佇列定義, 在叢集之間進行橋接。建立叢集遠端佇列定義、傳輸佇列及通道, 以區隔閘道佇列管理程式上的每一個訊息流程; 請參閱 [新增遠端佇列定義以隔離從閘道佇列管理程式傳送的訊息](#)。

從 7.5 版開始, 叢集佇列管理程式不限於單一叢集傳輸佇列。有兩個選項供您選擇:

1. 手動定義其他叢集傳輸佇列, 並定義哪些叢集傳送端通道會從每一個傳輸佇列傳送訊息; 請參閱 [新增叢集傳輸佇列以隔離從閘道佇列管理程式傳送的叢集訊息資料流量](#)。
2. 容許佇列管理程式自動建立及管理其他叢集傳輸佇列。它會為每一個叢集傳送端通道定義不同的叢集傳輸佇列; 請參閱 [將預設值變更為個別叢集傳輸佇列以隔離訊息資料流量](#)。

您可以結合部分叢集傳送端通道的手動定義叢集傳輸佇列, 以及管理其餘的佇列管理程式。傳輸佇列的組合是 [新增叢集傳輸佇列以隔離從閘道佇列管理程式傳送的叢集訊息資料流量](#) 中所採取的方法。在該解決方案

中，叢集之間的大部分訊息都使用一般 `SYSTEM.CLUSTER.TRANSMIT.QUEUE`。一個應用程式很重要，其所有訊息流程都使用一個手動定義的叢集傳輸佇列與其他流程隔離。

新增叢集傳輸佇列以隔離從閘道佇列管理程式傳送的叢集訊息資料流量中的配置受到限制。它不會將進入相同叢集中相同佇列管理程式上的叢集佇列的訊息資料流量，與另一個叢集佇列分開。您可以使用屬於分散式佇列的遠端佇列定義，來區隔個別佇列的訊息資料流量。使用叢集，使用多個叢集傳輸佇列，您可以區隔進入不同叢集傳送端通道的訊息資料流量。在相同佇列管理程式上，相同叢集中的多個叢集佇列會共用一個叢集傳送端通道。從閘道佇列管理程式轉遞之前，這些佇列的訊息會儲存在相同的傳輸佇列中。在新增叢集及叢集傳輸佇列以隔離閘道佇列管理程式所傳送的叢集訊息資料流量中的配置中，新增另一個叢集並使佇列管理程式及叢集佇列成為新叢集的成員，會遺漏此限制。新的佇列管理程式可能是叢集中唯一的佇列管理程式。您可以將更多佇列管理程式新增至叢集，並使用相同的叢集來隔離那些佇列管理程式上的叢集佇列。

相關概念

第 22 頁的『存取控制及多個叢集傳輸佇列』

當應用程式將訊息放入遠端叢集佇列時，請選擇三種檢查模式。這些模式包括對叢集佇列進行遠端檢查、對 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 進行本端檢查，或對叢集佇列或叢集佇列管理程式的本端設定檔進行檢查。

第 27 頁的『重疊叢集』

重疊叢集提供其他管理功能。使用名稱清單來減少管理重疊叢集所需的指令數目。

相關資訊

[授權將訊息放置在遠端叢集佇列上](#)

[使用叢集傳輸佇列及叢集傳送端通道](#)

[新增遠端佇列定義以隔離從閘道佇列管理程式傳送的訊息](#)

[新增叢集傳輸佇列以隔離從閘道佇列管理程式傳送的叢集訊息資料流量](#)

[新增叢集及叢集傳輸佇列，以隔離從閘道佇列管理程式傳送的叢集訊息資料流量](#)

[將預設值變更為個別叢集傳輸佇列以隔離訊息資料流量](#)

[使用閘道佇列管理程式建立兩個重疊的叢集](#)

[配置叢集之間的訊息路徑](#)

[安全](#)

[setmqaut](#)

叢集作業：規劃如何配置叢集傳輸佇列

系統會引導您選擇叢集傳輸佇列。您可以配置一個一般預設佇列、個別預設佇列或手動定義的佇列。

 將叢集傳送端通道配置成使用 `SYSTEM.CLUSTER.TRANSMIT.QUEUE`，您需要使用 `CSQ6SYSP` 參數中的作業模式 (`OPMODE`) 系統參數來啟用第 8 版新功能。

開始之前

檢閱 第 41 頁的『如何選擇要使用的叢集傳輸佇列類型』。

 請參閱作業模式 (`OPMODE`) 主題，以取得相關資訊。

關於這項作業

當您規劃如何配置佇列管理程式來選取叢集傳輸佇列時，有一些選項要做。

- 叢集訊息傳送的預設叢集傳輸佇列為何？
 - 一般叢集傳輸佇列 `SYSTEM.CLUSTER.TRANSMIT.QUEUE`。
 - 個別叢集傳輸佇列。佇列管理程式會管理個別叢集傳輸佇列。它會從模型佇列 `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE` 將它們建立為永久動態佇列。它會為它使用的每一個叢集傳送端通道建立一個叢集傳輸佇列。
- 對於您決定手動建立的叢集傳輸佇列，您還有兩個選項：
 - 針對您決定手動配置的每一個叢集傳送端通道，定義個別的傳輸佇列。在此情況下，請將傳輸佇列的 **CLCHNAME** 佇列屬性設為叢集傳送端通道的名稱。選取要從此傳輸佇列傳送訊息的叢集傳送端通道。

- b. 將叢集傳送端通道群組的訊息資料流量結合到相同的叢集傳輸佇列; 請參閱 第 40 頁的圖 13。在此情況下, 請將每一個一般傳輸佇列的 **CLCHNAME** 佇列屬性設為通用叢集傳送端通道名稱。通用叢集傳送端通道名稱是將叢集傳送端通道名稱分組的過濾器。例如, **SALES.*** 會將名稱以 **SALES.** 開頭的所有叢集傳送端通道分組。您可以在 **filter-string** 中的任何位置放置多個萬用字元。萬用字元是一個星號 *****。它代表從零到任意數目的字元。

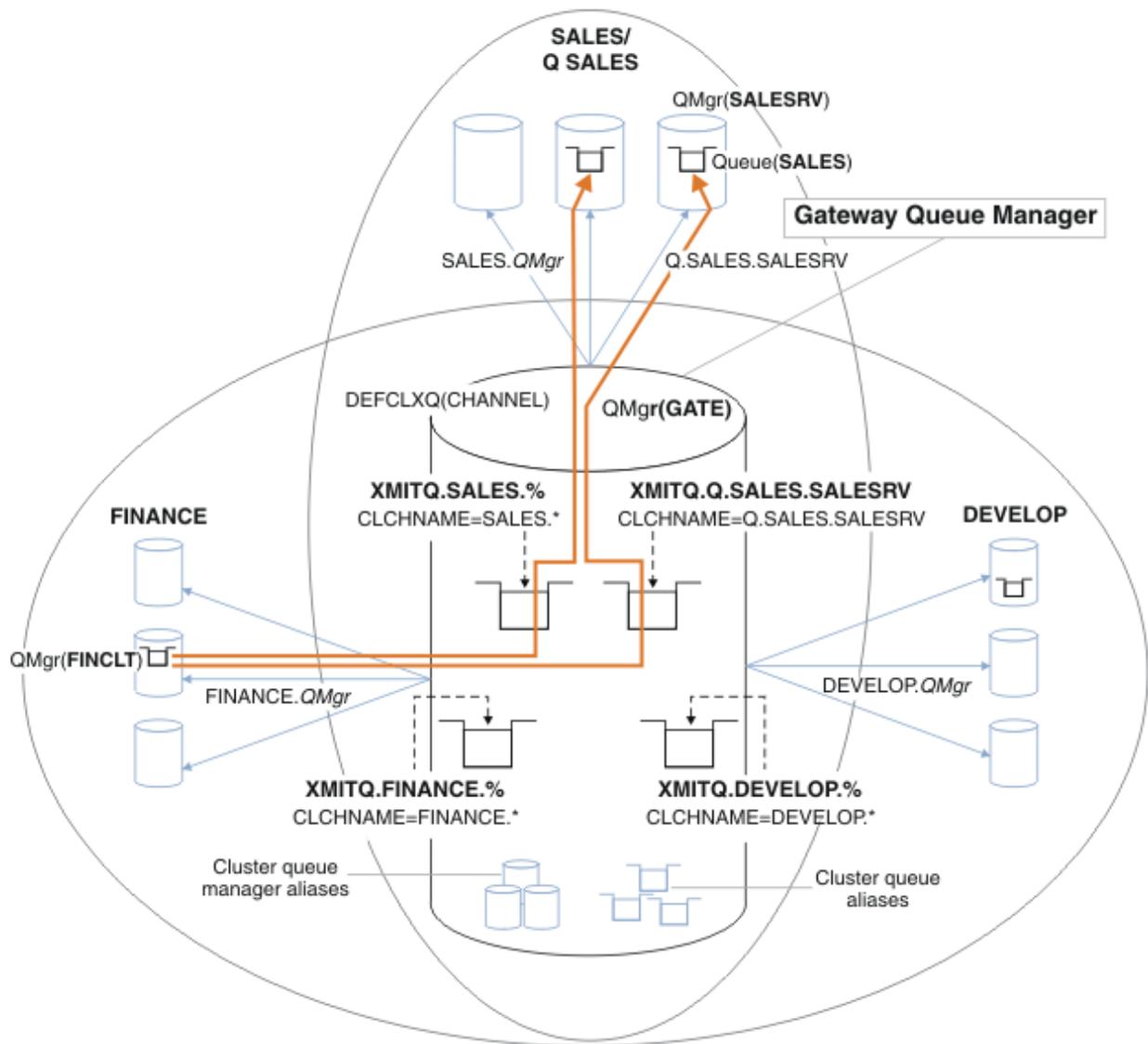


圖 13: 不同部門 IBM MQ 叢集的特定傳輸佇列範例

程序

1. 選取要使用的預設叢集傳輸佇列類型。
 - 選擇單一叢集傳輸佇列, 或為每個叢集連線選擇個別佇列。
- 保留預設值或執行 **MQSC** 指令:

```
ALTER QMGR DEFCLXQ(CHANNEL)
```

2. 隔離不得與其他流程共用叢集傳輸佇列的任何訊息流程。
 - 請參閱 第 43 頁的『叢集作業: 多個叢集傳輸佇列的範例配置』。在此範例中, **SALES** 佇列 (必須隔離) 是 **SALESRV** 上 **SALES** 叢集的成員。若要隔離 **SALES** 佇列, 請建立新的叢集 **Q.SALES**, 讓 **SALESRV** 佇列管理程式成為成員, 並將 **SALES** 佇列修改為屬於 **Q.SALES**。

- 將訊息傳送至 SALES 的佇列管理程式也必須是新叢集的成員。如果您使用叢集佇列別名及閘道佇列管理程式，在許多情況下，您可以限制變更，使閘道佇列管理程式成為新叢集的成員。
- 不過，將流程從閘道分隔至目的地，並不會將流程從來源佇列管理程式分隔至閘道。但有時它足以將流程與閘道分開，而不是流向閘道。如果不足，請將來源佇列管理程式新增至新叢集。如果您要透過閘道傳送訊息，請將叢集別名移至新叢集，並繼續將訊息傳送至閘道上的叢集別名，而不是直接傳送至目標佇列管理程式。

請遵循下列步驟來隔離訊息流程：

- 配置流程的目的地，讓每一個目標佇列都是該佇列管理程式上特定叢集中的唯一佇列。
 - 針對您遵循系統命名慣例所建立的任何新叢集，建立叢集傳送端和叢集接收端通道。
 - 請參閱第 33 頁的『叢集作業：重疊叢集的特殊考量』。
 - 為每個將訊息傳送至目標佇列的佇列管理程式上的每一個隔離目的地定義叢集傳輸佇列。
 - 叢集傳輸佇列的命名慣例是使用字首為 XMITQ. 的叢集通道名稱屬性 CLCHNAME 值。
3. 建立叢集傳輸佇列以符合控管或監視需求。
- 一般控管及監視需求會導致每個叢集的傳輸佇列或每個佇列管理程式的傳輸佇列。如果您遵循叢集通道 `ClusterName.QueueManagerName` 的命名慣例，則可以輕鬆地建立通用通道名稱來選取佇列管理程式的叢集，或佇列管理程式所屬的所有叢集；請參閱第 43 頁的『叢集作業：多個叢集傳輸佇列的範例配置』。
 - 將星號取代為百分比符號，以擴充叢集傳輸佇列的命名慣例，以滿足通用通道名稱。例如：

```
DEFINE QLOCAL(XMITQ.SALES.%) USAGE(XMITQ) CLCHNAME(SALES.*)
```

相關概念

第 22 頁的『存取控制及多個叢集傳輸佇列』

當應用程式將訊息放入遠端叢集佇列時，請選擇三種檢查模式。這些模式包括對叢集佇列進行遠端檢查、對 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 進行本端檢查，或對叢集佇列或叢集佇列管理程式的本端設定檔進行檢查。

第 27 頁的『重疊叢集』

重疊叢集提供其他管理功能。使用名稱清單來減少管理重疊叢集所需的指令數目。

相關資訊

[使用叢集傳輸佇列及叢集傳送端通道](#)

[新增遠端佇列定義以隔離從閘道佇列管理程式傳送的訊息](#)

[新增叢集傳輸佇列以隔離從閘道佇列管理程式傳送的叢集訊息資料流量](#)

[新增叢集及叢集傳輸佇列，以隔離從閘道佇列管理程式傳送的叢集訊息資料流量](#)

[將預設值變更為個別叢集傳輸佇列以隔離訊息資料流量](#)

[使用閘道佇列管理程式建立兩個重疊的叢集](#)

[配置叢集之間的訊息路徑](#)

[如何選擇要使用的叢集傳輸佇列類型](#)

[如何在不同的叢集傳輸佇列配置選項之間進行選擇。](#)

從 7.5 版開始，您可以選擇哪個叢集傳輸佇列與叢集傳送端通道相關聯。

1. 您可以讓所有叢集傳送端通道與單一預設叢集傳輸佇列 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 相關聯。此選項是預設值，是執行 7.1 版或更舊版本的佇列管理程式的唯一選項。
2. 您可以將所有叢集傳送端通道設為自動與個別叢集傳輸佇列相關聯。佇列由佇列管理程式從模型佇列 `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE` 建立，並命名為 `SYSTEM.CLUSTER.TRANSMIT.ChannelName`。如果佇列管理程式屬性 `DEFCLXQ` 設為 `CHANNEL`，則通道將使用其唯一命名的叢集傳輸佇列。



小心：如果您搭配使用專用 `SYSTEM.CLUSTER.TRANSMIT.QUEUES` 與從早於 IBM WebSphere MQ 7.5 的產品版本升級的佇列管理程式，請確保 `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE` 將 `SHARE/NOSHARES` 選項設為 **SHARE**。

3. 您可以設定要由單一叢集傳輸佇列提供服務的特定叢集傳送端通道。透過建立傳輸佇列並將其 **CLCHNAME** 屬性設為叢集傳送端通道的名稱，選取此選項。
4. 您可以選取要由單一叢集傳輸佇列提供服務的叢集傳送端通道群組。透過建立傳輸佇列並將其 **CLCHNAME** 屬性設為通用通道名稱 (例如 *ClusterName.**)，選取此選項。如果您遵循第 33 頁的『叢集作業: 重疊叢集的特殊考量』中的命名慣例來命名叢集通道，則此名稱會選取連接至叢集 *ClusterName* 中佇列管理程式的所有叢集通道。

您可以結合部分叢集傳送端通道的任一預設叢集傳輸佇列選項，以及任意數目的特定及通用叢集傳輸佇列配置。

最佳作法

在大部分情況下，對於現有 IBM MQ 安裝，預設配置是最佳選擇。叢集佇列管理程式會將叢集訊息儲存在單一叢集傳輸佇列 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 上。您可以選擇變更預設值，將不同佇列管理程式及不同叢集的訊息儲存在個別傳輸佇列上，或定義您自己的傳輸佇列。

在大部分情況下，對於新的 IBM MQ 安裝，預設配置也是最佳選擇。從預設配置切換至替代預設值 (每個叢集傳送端通道有一個傳輸佇列) 的處理程序是自動的。回復也是自動的。選擇一個或另一個並不重要，您可以反轉它。

選擇不同配置的原因更多是與控管和管理相關，而不是與功能或效能相關。有幾個異常狀況，配置多個叢集傳輸佇列對佇列管理程式的行為沒有好處。它會導致更多佇列，且需要您修改已設定且參照單一傳輸佇列的監視及管理程序。因此，在平衡上，保留預設配置是最佳選擇，除非您有強大的控管或管理理由，因為有不同的選擇。

這些異常狀況都與 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 上儲存的訊息數目增加時所發生的情況有關。如果您採取每個步驟來區隔一個目的地的訊息與另一個目的地的訊息，則一個目的地的通道及遞送問題應該不會影響另一個目的地的遞送。不過，儲存在 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 上的訊息數目可能會增加，因為將訊息遞送至一個目的地的速度不夠快。一個目的地在 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 上的訊息數可能會影響將訊息遞送至其他目的地。

為了避免因填滿單一傳輸佇列而造成的問題，請在您的配置中建置足夠的容量。然後，如果目的地失敗，且開始建置訊息待辦事項，您有時間修正問題。

如果訊息透過中心佇列管理程式 (例如叢集閘道) 遞送，則它們會共用一般傳輸佇列 `SYSTEM.CLUSTER.TRANSMIT.QUEUE`。如果閘道佇列管理程式上儲存在 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 上的訊息數達到其深度上限，則佇列管理程式會開始拒絕傳輸佇列的新訊息，直到其深度減少為止。壅塞會影響透過閘道遞送之所有目的地的訊息。訊息會備份將訊息傳送至閘道之其他佇列管理程式的傳輸佇列。問題本身表現在寫入佇列管理程式錯誤日誌的訊息、訊息傳輸量下降，以及傳送訊息與訊息到達其目的地之間的較長經歷時間。

即使在傳輸佇列已滿之前，壅塞對單一傳輸佇列的影響也會變得很明顯。如果您有混合的訊息資料流量，含有一些大型非持續訊息及一些小型訊息，則隨著傳輸佇列填滿，遞送小型訊息的時間會增加。延遲的原因是將大量非持續訊息寫入通常不會寫入磁碟的磁碟。如果您有時間關鍵訊息流程，與其他混合訊息流程共用叢集傳輸佇列，則可以配置特殊訊息路徑以將其與其他訊息流程隔離; 請參閱 [新增叢集及叢集傳輸佇列以隔離從閘道佇列管理程式傳送的叢集訊息資料流量](#)。

配置個別叢集傳輸佇列的其他原因是為了符合控管需求，或簡化傳送至不同叢集目的地的監視訊息。例如，您可能必須示範一個目的地的訊息絕不會與另一個目的地的訊息共用傳輸佇列。

變更控制預設叢集傳輸佇列的佇列管理程式屬性 **DEFCLXQ**，以針對每個叢集傳送端通道建立不同的叢集傳輸佇列。多個目的地可以共用叢集傳送端通道，因此您必須規劃叢集以完全符合此目標。將 [新增叢集及叢集傳輸佇列](#)，以有系統地隔離從閘道佇列管理程式傳送至所有叢集佇列的叢集訊息資料流量方法套用至所有叢集佇列。您的目標結果是沒有叢集目的地與另一個叢集目的地共用叢集傳送端通道。因此，叢集目的地的任何訊息都不會與另一個目的地的訊息共用其叢集傳輸佇列。

為某個特定訊息流程建立個別叢集傳輸佇列，可讓您輕鬆監視傳送至該目的地的訊息流程。如果要使用新的叢集傳輸佇列，請定義佇列，將它與叢集傳送端通道相關聯，然後停止並啟動通道。變更不必是永久的。您可以隔離訊息流程一段時間，以監視傳輸佇列，然後重新回復為使用預設傳輸佇列。

相關工作

[叢集作業: 多個叢集傳輸佇列的範例配置](#)

在此作業中，您可以套用步驟，將多個叢集傳輸佇列規劃至三個重疊叢集。需求是將訊息流程從所有其他訊息流程分隔至一個叢集佇列，並儲存不同叢集傳輸佇列上不同叢集的訊息。

叢集作業: 切換叢集傳輸佇列

規劃如何使現有正式作業佇列管理程式的叢集傳輸佇列變更生效。

叢集作業: 多個叢集傳輸佇列的範例配置

在此作業中，您可以套用步驟，將多個叢集傳輸佇列規劃至三個重疊叢集。需求是將訊息流程從所有其他訊息流程分隔至一個叢集佇列，並儲存不同叢集傳輸佇列上不同叢集的訊息。

關於這項作業

此作業中的步驟顯示如何在第 39 頁的『叢集作業: 規劃如何配置叢集傳輸佇列』中套用程序，並達到第 43 頁的圖 14 中所示的配置。它是三個重疊叢集 (具有閘道佇列管理程式) 的範例，配置了個別叢集傳輸佇列。第 45 頁的『建立範例叢集』中說明了用於定義叢集的 MQSC 指令。

例如，有兩個需求。其一是將訊息流程從閘道佇列管理程式分隔到記載銷售的銷售應用程式。第二個是詢問在任何時間點有多少訊息等待傳送至不同的部門區域。已定義 SALES、FINANCE 及 DEVELOP 叢集。目前從 SYSTEM.CLUSTER.TRANSMIT.QUEUE 轉遞叢集訊息。

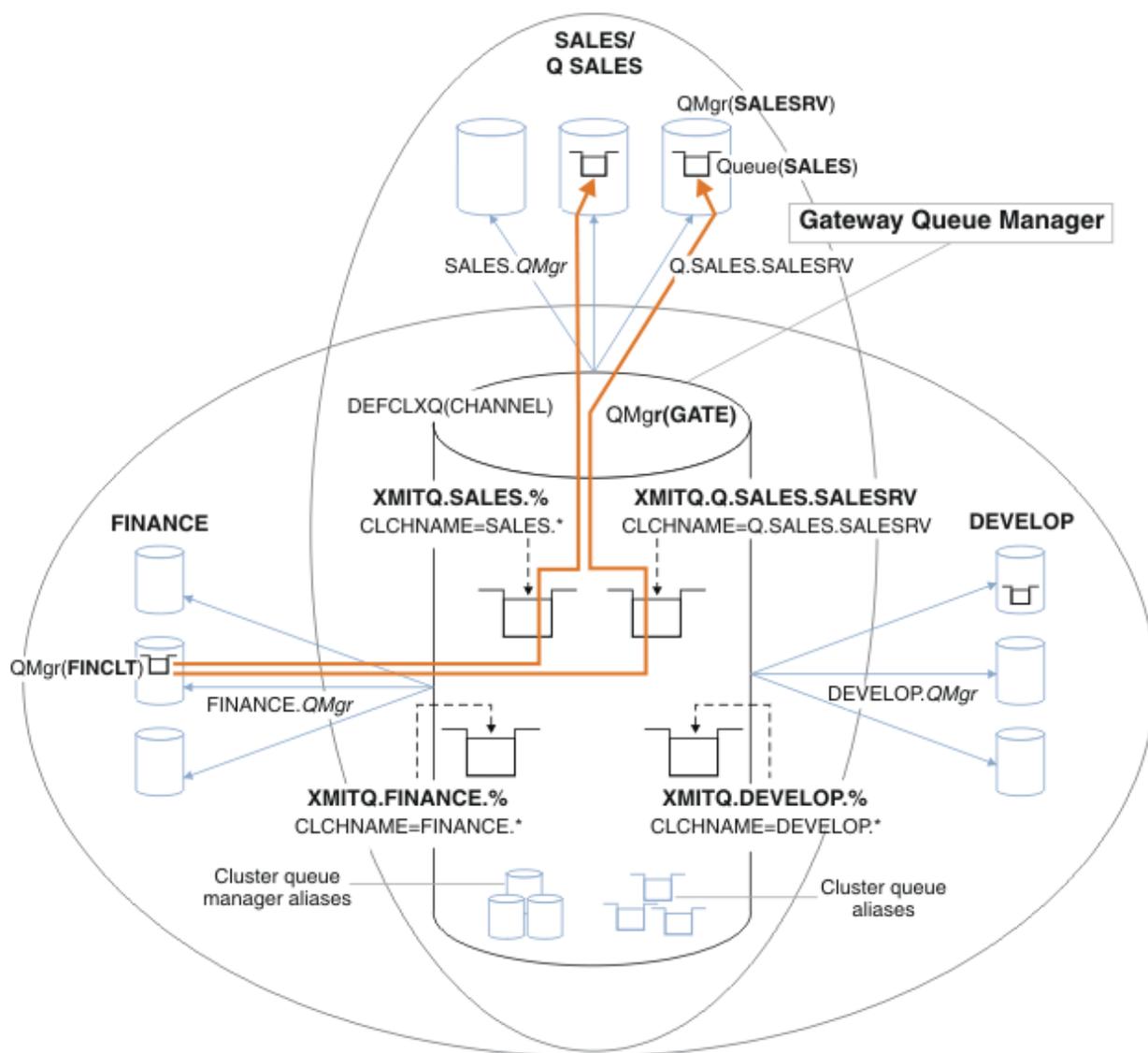


圖 14: 不同部門 IBM MQ 叢集的特定傳輸佇列範例

修改叢集的步驟如下所示; 如需定義，請參閱 [變更以隔離新叢集中的銷售佇列並區隔閘道叢集傳輸佇列](#)。

程序

1. 第一個配置步驟是 "選取要使用的預設叢集傳輸佇列類型"。

決策是在 GATE 佇列管理程式上執行下列 **MQSC** 指令，以建立個別的預設叢集傳輸佇列。

```
ALTER QMGR DEFCLXQ(CHANNEL)
```

沒有強烈理由選擇此預設值，因為目的是手動定義叢集傳輸佇列。選項確實具有弱診斷值。如果錯誤地執行手動定義，且訊息流經預設叢集傳輸佇列，則會在建立永久動態叢集傳輸佇列時顯示。

2. 第二個配置步驟是 "隔離不得與其他流程共用叢集傳輸佇列的任何訊息流程"。

在此情況下，從 SALESRV 上的佇列 SALES 接收訊息的銷售應用程式需要隔離。只需要隔離閘道佇列管理程式中的訊息。三個子步驟如下：

- a) "配置流程的目的地，讓每一個目標佇列都是該佇列管理程式上特定叢集中的唯一佇列"。

此範例需要將佇列管理程式 SALESRV 新增至銷售部門內的新叢集。如果您有幾個需要隔離的佇列，則可以決定為 SALES 佇列建立特定叢集。叢集名稱的可能命名慣例是將這類叢集命名為 **Q.QueueName**，例如 **Q.SALES**。如果您有大量要隔離的佇列，則另一種可能更實用的方法是在需要時隨時隨地建立隔離佇列的叢集。叢集名稱可能是 **QUEUES.n**。

在此範例中，新叢集稱為 **Q.SALES**。若要新增叢集，請參閱 變更以隔離新叢集中的銷售佇列並區隔閘道叢集傳輸佇列中的定義。定義變更的摘要如下：

- i) 將 **Q.SALES** 新增至儲存庫佇列管理程式上的叢集名單。在佇列管理程式 **REPOSNL** 參數中參照名單。
- ii) 將 **Q.SALES** 新增至閘道佇列管理程式上叢集的名單。在閘道佇列管理程式上的所有叢集佇列別名及叢集佇列管理程式別名定義中，會參照名稱清單。
- iii) 在佇列管理程式 SALESRV 上，針對其所屬的兩個叢集建立名稱清單，並變更 SALES 佇列的叢集成員資格：

```
DEFINE NAMELIST(CLUSTERS) NAMES(SALES, Q.SALES) REPLACE  
ALTER QLOCAL(SALES) CLUSTER(' ') CLUSNL(SALESRV.CLUSTERS)
```

SALES 佇列是這兩個叢集的成員，僅用於轉移。在新配置執行之後，您會從 SALES 叢集中移除 SALES 佇列；請參閱第 48 頁的圖 15。

- b) "針對您遵循系統命名慣例所建立的任何新叢集，建立叢集傳送端和叢集接收端通道"。

- i) 將叢集接收端通道 **Q.SALES.RepositoryQMGr** 新增至每一個儲存庫佇列管理程式
- ii) 將叢集傳送端通道 **Q.SALES.OtherRepositoryQMGr** 新增至每一個儲存庫佇列管理程式，以連接至其他儲存庫管理程式。啟動這些通道。
- iii) 將叢集接收端通道 **Q.SALES.SALESRV** 及 **Q.SALES.GATE** 新增至執行中的任一儲存庫佇列管理程式。
- iv) 將叢集傳送端通道 **Q.SALES.SALESRV** 及 **Q.SALES.GATE** 新增至 SALESRV 及 GATE 佇列管理程式。將叢集傳送端通道連接至您在其中建立叢集接收端通道的儲存庫佇列管理程式。

- c) "為每個將訊息傳送至目標佇列的佇列管理程式上的每一個隔離目的地定義叢集傳輸佇列"。

在閘道佇列管理程式上，定義 **Q.SALES.SALESRV** 叢集傳送端通道的叢集傳輸佇列 **XMITQ.Q.SALES.SALESRV**：

```
DEFINE QLOCAL(XMITQ.Q.SALES.SALESRV) USAGE(XMITQ) CLCHNAME(Q.SALES.SALESRV) REPLACE
```

3. 第三個配置步驟是 "建立叢集傳輸佇列以符合控管或監視需求"。

在閘道佇列管理程式上，定義叢集傳輸佇列：

```
DEFINE QLOCAL(XMITQ.SALES) USAGE(XMITQ) CLCHNAME(SALES.*) REPLACE  
DEFINE QLOCAL(XMITQ.DEVELOP) USAGE(XMITQ) CLCHNAME(DEVELOP.*) REPLACE  
DEFINE QLOCAL(XMITQ.FINANCE) USAGE(XMITQ) CLCHNAME(SALES.*) REPLACE
```

下一步

切換至閘道佇列管理程式上的新配置。

透過啟動新通道並重新啟動現在與不同傳輸佇列相關聯的通道來觸發交換器。或者，您可以停止並啟動閘道佇列管理程式。

1. 停止閘道佇列管理程式上的下列通道:

```
SALES. Qmgr
DEVELOP. Qmgr
FINANCE. Qmgr
```

2. 在閘道佇列管理程式上啟動下列通道:

```
SALES. Qmgr
DEVELOP. Qmgr
FINANCE. Qmgr
Q.SALES.SAVESRV
```

當交換器完成時，請從 SALES 叢集中移除 SALES 佇列; 請參閱 [第 48 頁的圖 15](#)。

相關概念

[如何選擇要使用的叢集傳輸佇列類型](#)

[如何在不同的叢集傳輸佇列配置選項之間進行選擇。](#)

相關工作

[叢集作業: 切換叢集傳輸佇列](#)

[規劃如何使現有正式作業佇列管理程式的叢集傳輸佇列變生效。](#)

建立範例叢集

建立範例叢集，並修改它以隔離閘道佇列管理程式上的 SALES 佇列及個別訊息的定義及指示。

關於這項作業

基本叢集的定義、變更以隔離新叢集中的銷售佇列並區隔閘道叢集傳輸佇列，以及從銷售叢集中移除佇列管理程式 SALESRV 上的銷售佇列中提供完整 **MQSC** 指令來建立 FINANCE、SALES 及 Q.SALES 叢集。定義會省略 DEVELOP 叢集，以保持較短的定義。

程序

1. 建立 SALES 和 FINANCE 叢集，以及閘道佇列管理程式。

- a) 建立佇列管理程式。

針對 [第 45 頁的表 4](#) 中的每一個佇列管理程式名稱執行指令: `critmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE QmgrName`。

| 說明 | 佇列管理程式名稱 | 埠號 |
|-------|----------|------|
| 財務儲存庫 | FINR1 | 1414 |
| 財務儲存庫 | FINR2 | 1415 |
| 財務用戶端 | FINCLT | 1418 |
| 銷售儲存庫 | SALER1 | 1416 |
| 銷售儲存庫 | SALER2 | 1417 |
| 銷售伺服器 | SALESRV | 1419 |
| 閘道 | GATE | 1420 |

b) 啟動所有佇列管理程式

針對 第 45 頁的表 4 中的每一個佇列管理程式名稱執行指令: `strmqm QmgrName`。

c) 建立每一個佇列管理程式的定義

Run the command: `runmqsc QmgrName < filename` where the files are listed in [基本叢集的定義](#), and the file name matches the queue manager name.

基本叢集的定義

finr1.txt

```
DEFINE LISTENER(1414) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1414) REPLACE
START LISTENER(1414)
ALTER QMGR REPOS(FINANCE)
DEFINE CHANNEL(FINANCE.FINR2) CHLTYPE(CLUSSDR) CONNAME('localhost(1415)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1414)')
CLUSTER(FINANCE) REPLACE
```

finr2.txt

```
DEFINE LISTENER(1415) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1415) REPLACE
START LISTENER(1415)
ALTER QMGR REPOS(FINANCE)
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSSDR) CONNAME('localhost(1414)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINR2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1415)')
CLUSTER(FINANCE) REPLACE
```

finclt.txt

```
DEFINE LISTENER(1418) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1418) REPLACE
START LISTENER(1418)
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSSDR) CONNAME('localhost(1414)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINCLT) CHLTYPE(CLUSRCVR) CONNAME('localhost(1418)')
CLUSTER(FINANCE) REPLACE
DEFINE QMODEL(SYSTEM.SAMPLE.REPLY) REPLACE
```

saler1.txt

```
DEFINE LISTENER(1416) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1416) REPLACE
START LISTENER(1416)
ALTER QMGR REPOS(SALES)
DEFINE CHANNEL(SALES.SALER2) CHLTYPE(CLUSSDR) CONNAME('localhost(1417)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1416)')
CLUSTER(SALES) REPLACE
```

saler2.txt

```
DEFINE LISTENER(1417) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1417) REPLACE
START LISTENER(1417)
ALTER QMGR REPOS(SALES)
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.SALER2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1417)')
CLUSTER(SALES) REPLACE
```

salesrv.txt

```
DEFINE LISTENER(1419) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1419) REPLACE
START LISTENER(1419)
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.SALESRV) CHLTYPE(CLUSRCVR) CONNAME('localhost(1419)')
CLUSTER(SALES) REPLACE
DEFINE QLOCAL(SALES) CLUSTER(SALES) TRIGGER INITQ(SYSTEM.DEFAULT.INITIATION.QUEUE)
```

```
PROCESS(ECHO) REPLACE
DEFINE PROCESS(ECHO) APPLICID(AMQSECH) REPLACE
```

gate.txt

```
DEFINE LISTENER(1420) TRPTYPE(TCP) IPADDR(LOCALHOST) CONTROL(QMGR) PORT(1420) REPLACE
START LISTENER(1420)
DEFINE NAMELIST(ALL) NAMES(SALES, FINANCE)
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSSDR) CONNAME('LOCALHOST(1414)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.GATE) CHLTYPE(CLUSRCVR) CONNAME('LOCALHOST(1420)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('LOCALHOST(1416)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.GATE) CHLTYPE(CLUSRCVR) CONNAME('LOCALHOST(1420)')
CLUSTER(SALES) REPLACE
DEFINE QALIAS(A.SALES) CLUSNL(ALL) TARGET(SALES) TARGTYPE(Queue) DEFBIND(NOTFIXED)
REPLACE
DEFINE QREMOTE(FINCLT) RNAME(' ') RQMNAME(FINCLT) CLUSNL(ALL) REPLACE
DEFINE QREMOTE(SALESRV) RNAME(' ') RQMNAME(SALESRV) CLUSNL(ALL) REPLACE
```

2. 執行範例要求程式來測試配置。

a) 在 SALESRV 佇列管理程式上啟動觸發監視器程式

在 Windows 上，開啟指令視窗並執行指令 `runmqtrm -m SALESRV`

b) 執行範例要求程式，並傳送要求。

在 Windows 上，開啟指令視窗並執行指令 `amqsreq A.SALES FINCLT`

要求訊息會回應，並在 15 秒之後完成範例程式。

3. 建立定義以隔離 Q.SALES 叢集中的 SALES 佇列，並針對閘道佇列管理程式上 SALES 及 FINANCE 叢集的個別叢集訊息。

Run the command: `runmqsc QmgrName < filename` where the files are listed in the following list, and the file name almost matches the queue manager name.

隔離新叢集中的銷售佇列並區隔閘道叢集傳輸佇列的變更 chgsaler1.txt

```
DEFINE NAMELIST(CLUSTERS) NAMES(SALES, Q.SALES)
ALTER QMGR REPOS(' ') REPOSNL(CLUSTERS)
DEFINE CHANNEL(Q.SALES.SALER2) CHLTYPE(CLUSSDR) CONNAME('localhost(1417)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
```

chgsaler2.txt

```
DEFINE NAMELIST(CLUSTERS) NAMES(SALES, Q.SALES)
ALTER QMGR REPOS(' ') REPOSNL(CLUSTERS)
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.SALER2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1417)')
CLUSTER(Q.SALES) REPLACE
```

chgsalesrv.txt

```
DEFINE NAMELIST (CLUSTERS) NAMES(SALES, Q.SALES)
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.SAVESRV) CHLTYPE(CLUSRCVR) CONNAME('localhost(1419)')
CLUSTER(Q.SALES) REPLACE
ALTER QLOCAL (SALES) CLUSTER(' ') CLUSNL(CLUSTERS)
```

chgate.txt

```
ALTER NAMELIST(ALL) NAMES(SALES, FINANCE, Q.SALES)
ALTER QMGR DEFCLXQ(CHANNEL)
```

```

DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.GATE) CHLTYPE(CLUSRCVR) CONNAME('localhost(1420)')
CLUSTER(Q.SALES) REPLACE
DEFINE QLOCAL (XMITQ.Q.SALES.SALESRV) USAGE(XMITQ) CLCHNAME(Q.SALES.SALESRV) REPLACE
DEFINE QLOCAL (XMITQ.SALES) USAGE(XMITQ) CLCHNAME(SALES.*) REPLACE
DEFINE QLOCAL (XMITQ.FINANCE) USAGE(XMITQ) CLCHNAME(FINANCE.*) REPLACE

```

4. 從 SALES 叢集中移除 SALES 佇列。

在 第 48 頁的圖 15 中執行 **MQSC** 指令:

```
ALTER QLOCAL(SALES) CLUSTER('Q.SALES') CLUSNL(' ')
```

圖 15: 從銷售叢集中移除佇列管理程式 SALESRV 上的銷售佇列

5. 將通道切換至新的傳輸佇列。

需求是停止並啟動 GATE 佇列管理程式正在使用的所有通道。若要使用最少數目的指令來執行此動作，請停止並啟動佇列管理程式

```
endmqm -i GATE
strmqm GATE
```

下一步

1. 重新執行範例要求程式以驗證新配置運作; 請參閱步驟 第 47 頁的『2』
2. 監視流經 GATE 佇列管理程式上所有叢集傳輸佇列的訊息:
 - a. 變更每一個叢集傳輸佇列的定義，以開啟佇列監視。

```
ALTER QLOCAL(SYSTEM.CLUSTER.TRANSMIT.
name) STATQ(ON)
```

- b. 檢查佇列管理程式統計資料監視是 OFF，以最小化輸出，並將監視間隔設為較低的值，以方便執行多個測試。

```
ALTER QMGR STATINT(60) STATCHL(OFF) STATQ(OFF) STATMQI(OFF) STATACLS(OFF)
```

- c. 重新啟動 GATE 佇列管理程式。
- d. 執行範例要求程式數次，以驗證流經 SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SALESRV 和 SYSTEM.CLUSTER.TRANSMIT.QUEUE 的訊息數目是否相等。要求會流經 SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SALESRV，並透過 SYSTEM.CLUSTER.TRANSMIT.QUEUE 回覆。

```
amqsmn -m GATE -t statistics
```

- e. 幾個間隔的結果如下:

```

C:\Documents and Settings\Admin>amqsmn -m GATE -t statistics
MonitoringType: QueueStatistics
QueueManager: 'GATE'
IntervalStartDate: '2012-02-27'
IntervalStartTime: '14.59.20'
IntervalEndDate: '2012-02-27'
IntervalEndTime: '15.00.20'
CommandLevel: 700
ObjectCount: 2
QueueStatistics: 0

```

```
QueueName: 'SYSTEM.CLUSTER.TRANSMIT.QUEUE'  
CreateDate: '2012-02-24'  
CreateTime: '15.58.15'  
...  
Put1Count: [0, 0]  
Put1FailCount: 0  
PutBytes: [435, 0]  
GetCount: [1, 0]  
GetBytes: [435, 0]  
...  
QueueStatistics: 1  
QueueName: 'SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV'  
CreateDate: '2012-02-24'  
CreateTime: '16.37.43'  
...  
PutCount: [1, 0]  
PutFailCount: 0  
Put1Count: [0, 0]  
Put1FailCount: 0  
PutBytes: [435, 0]  
GetCount: [1, 0]  
GetBytes: [435, 0]  
...  
MonitoringType: QueueStatistics  
QueueManager: 'GATE'  
IntervalStartDate: '2012-02-27'  
IntervalStartTime: '15.00.20'  
IntervalEndDate: '2012-02-27'  
IntervalEndTime: '15.01.20'  
CommandLevel: 700  
ObjectCount: 2  
QueueStatistics: 0  
QueueName: 'SYSTEM.CLUSTER.TRANSMIT.QUEUE'  
CreateDate: '2012-02-24'  
CreateTime: '15.58.15'  
...  
PutCount: [2, 0]  
PutFailCount: 0  
Put1Count: [0, 0]  
Put1FailCount: 0  
PutBytes: [863, 0]  
GetCount: [2, 0]  
GetBytes: [863, 0]  
...  
QueueStatistics: 1  
QueueName: 'SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV'  
CreateDate: '2012-02-24'  
CreateTime: '16.37.43'  
...  
PutCount: [2, 0]  
PutFailCount: 0  
Put1Count: [0, 0]  
Put1FailCount: 0  
PutBytes: [863, 0]  
GetCount: [2, 0]  
GetBytes: [863, 0]  
...  
2 Records Processed.
```

在第一個間隔中傳送了一個要求及回覆訊息，而在第二個間隔中傳送了兩個要求及回覆訊息。您可以推斷已在 SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV 上放置要求訊息，並在 SYSTEM.CLUSTER.TRANSMIT.QUEUE 上放置回覆訊息。

叢集作業: 切換叢集傳輸佇列

規劃如何使現有正式作業佇列管理程式的叢集傳輸佇列變生效。

開始之前

如果您減少切換處理程序必須傳送至新傳輸佇列的訊息數，則切換會更快完成。請閱讀 [將叢集傳送端通道切換至不同傳輸佇列的程序如何運作](#)，以瞭解在繼續任何進一步作業之前嘗試清空傳輸佇列的原因。

關於這項作業

您可以選擇兩種方式來使叢集傳輸佇列的變生效。

1. 讓佇列管理程式自動進行變更。這是預設值。當叢集傳送端通道下次啟動時，佇列管理程式會切換具有擱置傳輸佇列變更的叢集傳送端通道。
2. 手動進行變更。當叢集傳送端通道停止時，您可以對它進行變更。在叢集傳送端通道啟動之前，您可以將它從一個叢集傳輸佇列切換至另一個叢集傳輸佇列。

在決定要選擇兩個選項中的哪一個選項時，您會考量哪些因素，以及如何管理交換器？

程序

- 選項 1: 讓佇列管理程式自動進行變更; 請參閱 [第 51 頁的『將作用中叢集傳送端通道切換至另一組叢集傳輸佇列』](#)。

如果您想要佇列管理程式為您進行切換，請選擇此選項。

說明此選項的另一種方式是指出佇列管理程式會切換叢集傳送端通道，而不會強制通道停止。您可以選擇強制通道停止，然後啟動通道，讓交換器更快發生。交換器會在通道啟動時啟動，並在通道執行時執行，這與選項 2 不同。在選項 2 中，當通道停止時，會進行切換。

如果您透過讓切換自動發生來選擇此選項，則切換處理程序會在叢集傳送端通道啟動時啟動。如果通道未停止，則會在它變成非作用中之後啟動 (如果有要處理的訊息的話)。如果通道已停止，請使用 START CHANNEL 指令來啟動它。

一旦在通道所處理的傳輸佇列上沒有叢集傳送端通道的剩餘訊息，交換器處理程序即會完成。在這種情況下，叢集傳送端通道的新送達訊息會直接儲存在新的傳輸佇列中。在此之前，訊息會儲存在舊傳輸佇列上，而切換處理程序會將訊息從舊傳輸佇列傳送至新傳輸佇列。在整個切換處理程序期間，叢集傳送端通道會轉遞來自新叢集傳輸佇列的訊息。

交換器處理程序何時完成取決於系統的狀態。如果您在維護時間範圍中進行變更，請事先評量切換處理程序是否會及時完成。它是否會在時間內完成取決於等待從舊傳輸佇列傳送的訊息數是否達到零。

第一種方法的優點是它是自動的。缺點是如果進行配置變更的時間限制為維護時間，則您必須確信可以控制系統在維護時間範圍內完成交換器處理程序。如果您無法確定，則選項 2 可能是更好的選擇。

- 選項 2: 手動進行變更; 請參閱 [第 52 頁的『將已停止的叢集傳送端通道切換至另一個叢集傳輸佇列』](#)。

如果您要手動控制整個切換程序，或您要切換已停止或非作用中通道，請選擇此選項。如果您要切換一些叢集傳送端通道，且想要在維護時間期間執行切換，這是很好的選擇。

此選項的替代說明是指出您在叢集傳送端通道停止時切換叢集傳送端通道。

如果您選擇此選項，則可以完全控制交換器發生的時間。

您可以確定在固定時間量內，在維護時間範圍內完成切換處理程序。切換所花費的時間取決於必須將多少訊息從一個傳輸佇列傳送至另一個傳輸佇列。如果訊息持續送達，則處理程序可能需要一些時間來傳送所有訊息。

您可以選擇切換通道，而不從舊傳輸佇列傳送訊息。交換器是 "即時"。

當您重新啟動叢集傳送端通道時，它會開始處理您新指派給它的傳輸佇列上的訊息。

第二種方法的優點是您可以控制切換處理程序。缺點是您必須識別要切換的叢集傳送端通道，執行必要的指令，並解決任何可能阻止叢集傳送端通道停止的不確定通道。

相關概念

如何選擇要使用的叢集傳輸佇列類型
如何在不同的叢集傳輸佇列配置選項之間進行選擇。

相關工作

[叢集作業: 多個叢集傳輸佇列的範例配置](#)

在此作業中，您可以套用步驟，將多個叢集傳輸佇列規劃至三個重疊叢集。需求是將訊息流程從所有其他訊息流程分隔至一個叢集佇列，並儲存不同叢集傳輸佇列上不同叢集的訊息。

[第 51 頁的『將作用中叢集傳送端通道切換至另一組叢集傳輸佇列』](#)

此作業提供三個選項來切換作用中叢集傳送端通道。其中一個選項是讓佇列管理程式自動進行切換，這不會影響執行中的應用程式。其他選項是手動停止及啟動通道，或重新啟動佇列管理程式。

[第 52 頁的『將已停止的叢集傳送端通道切換至另一個叢集傳輸佇列』](#)

相關資訊

[將叢集傳送端通道切換至不同傳輸佇列的處理程序如何運作](#)

[將作用中叢集傳送端通道切換至另一組叢集傳輸佇列](#)

此作業提供三個選項來切換作用中叢集傳送端通道。其中一個選項是讓佇列管理程式自動進行切換，這不會影響執行中的應用程式。其他選項是手動停止及啟動通道，或重新啟動佇列管理程式。

開始之前

變更叢集傳輸佇列配置。您可以變更 **DEFCLXQ** 佇列管理程式屬性，或新增或修改傳輸佇列的 **CLCHNAME** 屬性。

如果您減少切換處理程序必須傳送至新傳輸佇列的訊息數，則切換會更快完成。請閱讀 [將叢集傳送端通道切換至不同傳輸佇列的程序如何運作](#)，以瞭解在繼續任何進一步作業之前嘗試清空傳輸佇列的原因。

關於這項作業

使用作業中的步驟作為基礎，以制定您自己的計劃來進行叢集傳輸佇列配置變更。

程序

1. 選擇性的: 記錄現行通道狀態

記錄負責處理叢集傳輸佇列之現行及已儲存通道的狀態。下列指令顯示與系統叢集傳輸佇列相關聯的狀態。新增您自己的指令，以顯示與您已定義的叢集傳輸佇列相關聯的狀態。使用慣例 (例如 XMITQ.ChannelName) 來命名您定義的叢集傳輸佇列，以方便顯示那些傳輸佇列的通道狀態。

```
DISPLAY CHSTATUS(*) WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
DISPLAY CHSTATUS(*) SAVED WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
```

2. 切換傳輸佇列。

- 不執行任何動作。當叢集傳送端通道在停止或非作用中之後重新啟動時，佇列管理程式會切換叢集傳送端通道。

如果您沒有變更佇列管理程式配置的規則或考量，請選擇這個選項。執行中應用程式不受變更影響。

- 重新啟動佇列管理程式。所有叢集傳送端通道會隨需應變停止並自動重新啟動。

選擇此選項以立即起始所有變更。當佇列管理程式關閉並重新啟動時，會岔斷執行中的應用程式。

- 停止個別叢集傳送端通道並重新啟動它們。

選擇此選項可立即變更一些通道。在停止和重新啟動訊息通道之間，執行中的應用程式在訊息傳送方面會遇到短暫的延遲。除非在您停止叢集傳送端通道期間，否則該通道仍在執行中。在切換過程中，訊息被傳送到舊傳輸佇列，由切換過程傳送到新傳輸佇列，並由叢集傳送端通道從新傳輸佇列轉發。

3. 選擇性的: 在通道切換時監視通道

顯示交換器期間的通道狀態及傳輸佇列深度。下列範例顯示系統叢集傳輸佇列的狀態。

```
DISPLAY CHSTATUS(*) WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
DISPLAY CHSTATUS(*) SAVED WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
DISPLAY QUEUE('SYSTEM.CLUSTER.TRANSMIT.*') CURDEPTH
```

4. 選擇性的: 監視訊息 " AMQ7341 通道 *ChannelName* 的傳輸佇列已從佇列 *QueueName* 切換至寫入佇列管理程式錯誤日誌的 *QueueName* "。

將已停止的叢集傳送端通道切換至另一個叢集傳輸佇列

開始之前

您可能會進行一些配置變更，現在想要讓它們生效，而不啟動受影響的叢集傳送端通道。或者，您也可以作為作業中的其中一個步驟來進行所需的配置變更。

如果您減少切換處理程序必須傳送至新傳輸佇列的訊息數，則切換會更快完成。請閱讀 [將叢集傳送端通道切換至不同傳輸佇列的程序如何運作](#)，以瞭解在繼續任何進一步作業之前嘗試清空傳輸佇列的原因。

關於這項作業

此作業會切換已停止或非作用中叢集傳送端通道所提供的傳輸佇列。您可以執行這項作業，因為叢集傳送端通道已停止，且您想要立即切換其傳輸佇列。例如，基於某些原因，叢集傳送端通道未啟動，或有其他配置問題。為了解決此問題，您決定建立叢集傳送端通道，並將舊叢集傳送端通道的傳輸佇列與您定義的新叢集傳送端通道相關聯。

更可能的情況是您想要控制何時執行叢集傳輸佇列的重新配置。若要完全控制重新配置，您可以停止通道，變更配置，然後切換傳輸佇列。

程序

1. 停止您想要切換的通道
 - a) 停止您想要切換的任何執行中或非作用中通道。停止非作用中叢集傳送端通道會阻止它在您進行配置變更時啟動。

```
STOP CHANNEL(ChannelName) MODE(QUIESCSE) STATUS(STOPPED)
```

2. 選擇性的: 進行配置變更。

例如，請參閱第 43 頁的『叢集作業: 多個叢集傳輸佇列的範例配置』。

3. 將叢集傳送端通道切換至新的叢集傳輸佇列。

  在分散式平台上，發出下列指令:

```
runswchl -m QmgrName -c ChannelName
```

 在 z/OS 上，請使用 CSQUTIL 指令的 SWITCH 函數來切換訊息或監視發生的情況。請使用下列指令。

```
SWITCH CHANNEL(channel_name) MOVEMSGS(YES)
```

如需相關資訊，請參閱 [SWITCH 函數](#)。

runswchl 或 CSQUTIL SWITCH 指令會將舊傳輸佇列上的任何訊息傳送至新的傳輸佇列。當此通道的舊傳輸佇列上的訊息數達到零時，即會完成切換。指令是同步的。此指令會在切換處理程序期間將進度訊息寫入視窗。

在傳送階段期間，會將針對叢集傳送端通道的現有及新訊息傳送至新的傳輸佇列。

由於叢集傳送端通道已停止，訊息會建置在新的傳輸佇列上。將已停止的叢集傳送端通道與第 51 頁的『將作用中叢集傳送端通道切換至另一組叢集傳輸佇列』中的步驟第 51 頁的『2』對照。在該步驟中，叢集傳送端通道正在執行中，因此訊息不一定會建置在新的傳輸佇列上。

4. 選擇性的: 在通道切換時監視通道

在不同的指令視窗中，顯示切換期間的傳輸佇列深度。下列範例顯示系統叢集傳輸佇列的狀態。

```
DISPLAY QUEUE('SYSTEM.CLUSTER.TRANSMIT.*') CURDEPTH
```

5. 選擇性的: 監視訊息 "AMQ7341 通道 ChannelName 的傳輸佇列已從佇列 QueueName 切換至寫入佇列管理程式錯誤日誌的 QueueName "。

6. 重新啟動您已停止的叢集傳送端通道。

當您停止通道時，通道不會自動啟動，通道會進入 STOPPED 狀態。

```
START CHANNEL(ChannelName)
```

相關資訊

[runswchl](#)

[解析通道](#)

[停止通道](#)

叢集作業: 移轉及修改最佳作法

本主題提供規劃及管理 IBM 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。

1. 第 53 頁的『[移動叢集中的物件](#)』(在叢集內四處移動物件而不安裝任何修正套件或 IBM MQ 新版本的最佳作法)。
2. 第 54 頁的『[升級及維護安裝](#)』(在套用維護或升級及測試新架構時，保持運作中叢集架構的最佳作法)。

移動叢集中的物件

應用程式及其佇列

當您必須將佇列管理程式上管理的佇列實例移至另一個佇列管理程式上管理時，您可以使用工作量平衡參數來確保順利轉移。

建立要在其中新管理它的佇列實例，但使用叢集工作量平衡設定來繼續將訊息傳送至原始實例，直到您的應用程式備妥可切換為止。這可透過下列步驟達成：

1. 將現有佇列的 **CLWLRANK** 內容設為高值，例如五。
2. 建立佇列的新實例，並將其 **CLWLRANK** 內容設為零。
3. 完成新系統的任何進一步配置，例如針對佇列的新實例部署及啟動耗用應用程式。
4. 將新佇列實例的 **CLWLRANK** 內容設為高於原始實例 (例如 9)。
5. 容許原始佇列實例處理系統中任何佇列訊息，然後刪除佇列。

移動整個佇列管理程式

如果佇列管理程式停留在相同主機上，但 IP 位址正在變更，則處理程序如下：

- 當正確使用 DNS 時，可以協助簡化處理程序。如需透過設定 [連線名稱 \(CONNAME\)](#) 通道屬性來使用 DNS 的相關資訊，請參閱 [ALTER CHANNEL](#)。
- 如果移動完整儲存庫，在進行變更之前，請確定您至少有一個其他完整儲存庫順利執行中 (例如，通道狀態沒有問題)。
- 使用 [SUSPEND QMGR](#) 指令暫停佇列管理程式，以避免資料流量增加。
- 修改電腦的 IP 位址。如果您的 CLUSRCVR 通道定義使用 CONNAME 欄位中的 IP 位址，請修改此 IP 位址登錄。可能需要清除 DNS 快取，以確保到處都有可用的更新項目。
- 當佇列管理程式重新連接至完整儲存庫時，通道自動定義會自動自行解析。

- 如果佇列管理程式管理完整儲存庫且 IP 位址變更，請務必確保盡快切換部分，以將任何手動定義的 CLUSSDR 通道指向新位置。在執行此切換之前，這些佇列管理程式可能只能聯絡剩餘 (未變更) 完整儲存庫，且可能會看到關於不正確通道定義的警告訊息。
- 使用 `RESUME QMGR` 指令回復佇列管理程式。

如果必須將佇列管理程式移至新主機，則可以複製佇列管理程式資料並從備份還原。不過，除非沒有其他選項，否則不建議此處理程序；最好在新機器上建立佇列管理程式，並依照前一節的說明來抄寫佇列及應用程式。此狀況提供平滑輪替/回復機制。

如果您決定使用備份來移動完整佇列管理程式，請遵循下列最佳作法：

- 將整個處理程序視為從備份還原的佇列管理程式，並套用您通常用於系統回復的任何處理程序，以適合您的作業系統環境。
- 移轉之後，請使用 `REFRESH CLUSTER` 指令來捨棄所有本端保留的叢集資訊 (包括任何不確定的自動定義通道)，並強制重建它。

註：對於大型叢集，使用 `REFRESH CLUSTER` 指令可能會對正在運行的叢集造成干擾，此後每隔 27 天在叢集物件向所有相關佇列管理程式自動傳送狀態更新項目時也可能造成干擾。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。

建立佇列管理程式並從叢集中的現有佇列管理程式抄寫設定 (如本主題先前所述) 時，絕不將兩個不同的佇列管理程式視為實際相同。尤其是不要為新的佇列管理程式提供相同的佇列管理程式名稱及 IP 位址。嘗試「捨棄」取代佇列管理程式是 IBM MQ 叢集中問題的常見原因。快取預期接收包括 `QMID` 屬性在內的更新項目，且狀態可能已毀損。

如果不小心以相同名稱建立了兩個不同的佇列管理程式，建議使用 `RESET CLUSTER QMID` 指令從叢集中退出不正確的項目。

升級及維護安裝

避免「大爆炸實務範例」(例如，停止所有叢集及佇列管理程式活動，將所有升級及維護套用至所有佇列管理程式，然後同時啟動所有項目)：叢集設計為仍然使用多個版本的佇列管理程式共存，因此建議使用計劃良好的分階段維護方法。

具有備份計劃：

-  在 z/OS 上，您是否已套用舊版移轉 PTF？
- 你有備份嗎？
- 避免立即使用新的叢集功能：請等到您確定所有佇列管理程式都已升級至新層次，且確定您不會回復任何佇列管理程式。在某些佇列管理程式仍在較早層次的叢集中使用新的叢集功能，可能會導致未定義的行為。例如，在從 IBM WebSphere MQ 6.0 移至 IBM WebSphere MQ 7.1 的過程中，如果佇列管理程式定義叢集主題，則 IBM WebSphere MQ 6.0 佇列管理程式將無法瞭解定義或能夠發佈此主題。

請先移轉完整儲存庫。雖然他們可以轉遞他們不瞭解的資訊，但他們無法持續保存它，因此除非絕對必要，否則不建議使用它。如需相關資訊，請參閱 [佇列管理程式叢集移轉](#)。

叢集作業：使用 `REFRESH CLUSTER` 最佳作法

您可以使用 `REFRESH CLUSTER` 指令來捨棄所有本端保留的叢集相關資訊，並從叢集中的完整儲存庫重建該資訊。除非發生異常狀況，否則您應該不需要使用此指令。如果您確實需要使用它，對於如何使用它，有一些特殊考量。此資訊是根據客戶的測試和意見而提供的指引。

只有在您確實需要這樣做時，才執行 REFRESH CLUSTER

IBM MQ 叢集技術可確保叢集配置的任何變更 (例如叢集佇列的變更) 會自動變成已知，讓任何需要知道資訊的叢集成員都知道。無需採取進一步的行政步驟來實現這種資訊傳播。

如果這類資訊未到達叢集中需要它的佇列管理程式，例如，當應用程式第一次嘗試開啟叢集佇列管理程式時，叢集中的另一個佇列管理程式並不知道叢集佇列，這表示叢集基礎架構中有問題。例如，可能無法在佇列管理程式與完整儲存庫佇列管理程式之間啟動通道。因此，必須調查任何觀察到不一致的狀況。可能的話，請在不使用 `REFRESH CLUSTER` 指令的情況下解決此狀況。

在本產品說明文件其他地方所記載的罕見情況下，或在 IBM 支援中心要求時，您可以使用 **REFRESH CLUSTER** 指令來捨棄所有本端保留的叢集相關資訊，並從叢集中的完整儲存庫重建該資訊。

在大型叢集中重新整理可能會影響叢集的效能及可用性

在叢集進行中時使用 **REFRESH CLUSTER** 指令可能會干擾叢集，例如，在完整儲存庫處理佇列管理程式叢集資源的重新傳送時，會突然增加完整儲存庫的工作。如果您是在大型叢集中重新整理 (亦即，數百個佇列管理程式)，則應該儘可能避免在日常工作中使用此指令，並使用替代方法來更正特定不一致。例如，如果叢集佇列未在整個叢集中正確延伸，則更新叢集佇列定義 (例如變更其說明) 的起始調查技術會在整個叢集中重新延伸佇列配置。此處理程序可協助識別問題，並可能解決暫時不一致的問題。

如果無法使用替代方法，且您必須在大型叢集中執行 **REFRESH CLUSTER**，則應該在離峰時間或維護時間期間執行，以避免影響使用者工作量。您也應該避免在單一批次中重新整理大型叢集，而是依照 [第 55 頁](#) 的『當叢集物件傳送自動更新時，避免效能和可用性問題』中的說明來錯開活動。

當叢集物件傳送自動更新時，避免效能和可用性問題

在佇列管理程式上定義新的叢集物件之後，此物件的更新會從定義時開始每 27 天產生一次，並傳送至叢集中的每個完整儲存庫，以及傳送至任何其他感興趣的佇列管理程式。當您對佇列管理程式發出 **REFRESH CLUSTER** 指令時，您會在指定叢集本端定義的所有物件上重設此自動更新的時鐘。

如果您在單一批次中重新整理大型叢集 (亦即數百個佇列管理程式)，或在其他情況下 (例如從配置備份重建系統)，則在 27 天之後，所有這些佇列管理程式會同時將其所有物件定義重新通告至完整儲存庫。在完成所有更新之前，這可能再次導致系統執行速度大幅降低，甚至變成無法使用。因此，當您必須重新整理或重建大型叢集中的多個佇列管理程式時，您應該錯開數小時或數天的活動，以便後續的自動更新不會定期影響系統效能。

系統叢集歷程佇列

當執行 **REFRESH CLUSTER** 時，佇列管理程式會在重新整理之前取得叢集狀態的 Snapshot，如果它定義在佇列管理程式上，則會將它儲存在 `SYSTEM.CLUSTER.HISTORY.QUEUE (SCHQ)` 上。此 Snapshot 僅用於 IBM 服務目的，以防稍後系統發生問題。

依預設，啟動時，會在分散式佇列管理程式上定義 SCHQ。對於 z/OS 移轉，必須手動定義 SCHQ。

SCHQ 上的訊息會在三個月後到期。

相關概念

[第 84 頁](#) 的『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。

相關資訊

[執行 REFRESH CLUSTER 時看到的應用程式問題](#)

[MQSC 指令參照: REFRESH CLUSTER](#)

叢集作業: 可用性、多重實例及災難回復

本主題提供規劃及管理 IBM 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。

IBM MQ 叢集作業本身不是高可用性解決方案，但在某些情況下，可以使用 IBM MQ 來改善服務的可用性，例如，在不同佇列管理程式上具有佇列的多個實例。本節提供指引，說明如何確保 IBM MQ 基礎架構盡可能高可用性，以便在這類架構中使用。

叢集資源的可用性

通常建議維護兩個完整儲存庫的原因，是失去一個儲存庫對叢集順利執行並不重要。即使兩者都變成無法使用，部分儲存庫所保留的現有知識仍有 60 天寬限期，雖然在此事件中無法使用新的或先前未存取的資源 (例如佇列)。

使用叢集來改善應用程式可用性

透過使用佇列及應用程式的多個實例，叢集可以協助設計高可用性應用程式 (例如要求/回應類型伺服器應用程式)。必要的話，優先順序屬性可以優先使用「即時」應用程式，除非佇列管理程式或通道 (例如，變成無法使用)。這可讓您快速切換，以便在發生問題時繼續處理新訊息。

不過，已遞送至叢集中特定佇列管理程式的訊息只會保留在該佇列實例上，且在回復該佇列管理程式之前無法進行處理。基於此原因，基於真實資料高可用性，您可能想要考量其他技術，例如多重實例佇列管理程式。

多重實例佇列管理程式

軟體高可用性(多重實例)是保持現有訊息可用的最佳內建供應項目。如需相關資訊，請參閱 [搭配使用 IBM MQ 與高可用性配置](#)、[建立多重實例佇列管理程式](#)，以及下列小節。只要叢集中的所有佇列管理程式都至少執行 IBM WebSphere MQ 7.0.1，叢集中的任何佇列管理程式都可以使用此技術來達到高可用性。如果叢集中有任何佇列管理程式處於先前層次，則在失效接手至次要 IP 時，它們可能會失去與多重實例佇列管理程式的連線功能。

如本主題先前所討論，只要配置兩個完整儲存庫，其本質幾乎都是高可用性。如果您需要，IBM MQ 軟體高可用性/多重實例佇列管理程式可以用於完整儲存庫。沒有充分理由使用這些方法，事實上對於暫時中斷，這些方法可能會在失效接手期間造成額外的效能成本。不建議使用軟體 HA 而不執行兩個完整儲存庫，因為例如，在單一通道中斷的情況下，它不一定會失效接手，但可能會讓局部儲存庫無法查詢叢集資源。

災難回復

災難回復(例如，從儲存佇列管理程式資料的磁碟變成毀損進行回復)很難做得好; IBM MQ 可以提供協助，但無法自動執行。IBM MQ 中唯一的 'true' 災難回復選項(排除任何作業系統或其他基礎抄寫技術)是從備份還原。在下列情況下，有一些叢集特定點需要考量：

- 測試災難回復實務範例時請小心。例如，如果測試備份佇列管理程式的作業，在相同網路中讓它們上線時請小心，因為可能會意外加入即時叢集，並透過管理與即時叢集佇列管理程式中相同的具名佇列來啟動「竊取」訊息。
- 災難回復測試不得干擾執行中的即時叢集。避免干擾的技術包括：
 - 在防火牆層次完成網路分隔或分隔。
 -  未啟動通道起始程式或 z/OS **chinit** 位址空間。
 - 除非發生實際災難回復實務範例，否則不會向災難回復系統發出即時 SSL 憑證。
- 在叢集中還原佇列管理程式的備份時，備份可能與叢集的其餘部分不同步。**REFRESH CLUSTER** 指令可以解析更新項目並與叢集同步化，但 **REFRESH CLUSTER** 指令必須作為最後手段使用。請參閱第 54 頁的『叢集作業：使用 REFRESH CLUSTER 最佳作法』。檢閱任何內部程序文件及 IBM MQ 文件，以查看在使用指令之前是否遺漏簡單的步驟。
- 至於任何回復，應用程式必須處理重播及資料流失。必須決定是要將佇列清除至已知狀態，還是要在其他位置有足夠資訊來管理重播。

規劃分散式發佈/訂閱網路

您可以建立佇列管理程式的網路，其中在一個佇列管理程式上建立的訂閱將接收由連接至網路中另一個佇列管理程式的應用程式所發佈的相符訊息。若要選擇合適的拓撲，您需要考量手動控制、網路大小、變更頻率、可用性及其可調整性的需求。

開始之前

這項作業假設您瞭解何謂分散式發佈/訂閱網路，以及它們的運作方式。如需技術概觀，請參閱 [分散式發佈/訂閱網路](#)。

關於這項作業

發佈/訂閱網路有三個基本拓撲：

- 直接遞送叢集
- 主題主機遞送叢集
- 階層

對於前兩個拓撲，起始點是 IBM MQ 叢集配置。可以使用或不使用叢集來建立第三個拓撲。如需規劃基礎佇列管理程式網路的相關資訊，請參閱第 14 頁的『[規劃分散式佇列及叢集](#)』。

當叢集已存在時，直接遞送叢集是最簡單配置的拓撲。您在任何佇列管理程式上定義的任何主題都會自動在叢集中的每一個佇列管理程式上提供，發佈會直接從發佈應用程式所連接的任何佇列管理程式，遞送至存在相符訂閱的每一個佇列管理程式。這種簡單的配置依賴 IBM MQ 維護叢集中每個佇列管理程式之間的高階資訊共用及連線功能。對於小型及簡式網路 (亦即，少量佇列管理程式，以及一組相當靜態的發佈者及訂閱者)，這是可接受的。不過，在更大或更動態的環境中使用時，額外負擔可能過高。請參閱第 60 頁的『發佈/訂閱叢集中的直接遞送』。

主題主機遞送叢集 提供與直接遞送叢集相同的好處，方法是讓您在叢集中任何佇列管理程式上定義的任何主題自動在叢集中每一個佇列管理程式上可用。不過，主題主機遞送叢集需要您仔細選擇管理每一個主題的佇列管理程式，因為該主題的所有資訊及發佈都會通過那些主題主機佇列管理程式。這表示系統不需要維護所有佇列管理程式之間的通道及資訊流程。不過，它也表示發佈可能不再直接傳送至訂閱者，但可能會透過主題主機佇列管理程式遞送。基於這些原因，可能會在系統上增加額外負擔，特別是在管理主題的佇列管理程式上，因此需要仔細規劃拓撲。此拓撲對於包含許多佇列管理程式或管理一組動態發佈者和訂閱者 (亦即，經常新增或移除的發佈者或訂閱者) 的網路特別有效。可以定義其他主題主機，以改善路徑的可用性及水平調整發佈工作量。請參閱第 65 頁的『發佈/訂閱叢集中的主題主機遞送』。

階層 需要最手動的配置來設定，且是最難修改的拓撲。您必須手動配置階層中每一個佇列管理程式與其直接關係之間的關係。配置關係之後，發佈會 (如同前兩種拓撲) 遞送至階層中其他佇列管理程式上的訂閱。使用階層關係來遞送發佈。這可讓您配置非常特定的拓撲，以符合不同的需求，但也可能導致發佈需要許多 "中繼站"，透過中間佇列管理程式才能達到訂閱。發佈一律只有一個透過階層的路徑，因此每個佇列管理程式的可用性都很重要。階層通常只適用於無法配置單一叢集的情況；例如，當跨越多個組織時。請參閱第 85 頁的『發佈/訂閱階層中的遞送』。

在必要時，可以結合上述三種拓撲來解決特定的拓撲需求。如需範例，請參閱 [結合多個叢集的主題空間](#)。

若要選擇適合分散式發佈/訂閱網路的拓撲，您需要考量下列廣泛問題：

- 您的網路將會有多大？
- 您需要對其配置進行多少手動控制？
- 在主題和訂閱方面，以及在佇列管理程式方面，系統的動態程度如何？
- 您的可用性和可擴充性需求為何？
- 所有佇列管理程式可以直接彼此連接嗎？

程序

- 預估您的網路需要有多大。
 - a) 預估您需要多少主題。
 - b) 預估您預期擁有的發佈者和訂閱者數目。
 - c) 預估發佈/訂閱活動將涉及的佇列管理程式數目。
 - 另請參閱第 74 頁的『發佈/訂閱叢集作業: 最佳作法』，特別是下列各節：
 - [如何調整系統大小](#)
 - [限制發佈/訂閱活動所涉及叢集佇列管理程式數目的原因](#)
 - [如何決定要叢集的主題](#)

如果您的網路將有許多佇列管理程式，並處理許多發佈者和訂閱者，則您可能需要使用主題主機遞送叢集或階層。直接遞送叢集幾乎不需要手動配置，可以是小型或靜態網路的良好解決方案。

- 請考量您需要對管理每一個主題、發佈者或訂閱者的佇列管理程式進行多少手動控制。
 - a) 請考量部分佇列管理程式的功能是否低於其他佇列管理程式。
 - b) 請考量部分佇列管理程式的通訊鏈結是否比其他佇列管理程式更脆弱。
 - c) 識別您預期主題有許多發佈和少數訂閱者的案例。
 - d) 識別您預期主題有許多訂閱者和很少發佈的案例。

在所有拓撲中，發佈會遞送至其他佇列管理程式上的訂閱。在直接遞送叢集中，這些發佈會採用訂閱的最短路徑。在主題主機遞送叢集或階層中，您可以控制發佈所採取的路徑。如果佇列管理程式的功能不

同，或具有不同層次的可用性及連線功能，則您可能想要將特定的工作量指派給特定的佇列管理程式。您可以使用主題主機遞送叢集或階層來執行此動作。

在所有拓撲中，只要可能，將發佈應用程式與訂閱位於相同的佇列管理程式上，即可將額外需要降到最低，並將效能最大化。對於主題主機遞送叢集，請考量將發佈者或訂閱者放置在管理主題的佇列管理程式上。這會移除佇列管理程式之間的任何額外"中繼站"，以將發佈傳遞給訂閱者。在主題有許多發佈者和很少訂閱者，或有許多訂閱者和很少發佈者的情況下，此方法特別有效。例如，請參閱 [使用集中式發佈者或訂閱者的主題主機遞送](#)。

另請參閱第 74 頁的『[發佈/訂閱叢集作業: 最佳作法](#)』，特別是下列各節：

- [如何決定要叢集的主題](#)
- [發佈者和訂閱位置](#)

- 請考量網路活動的動態程度。

- a) 預估在不同主題上新增及移除訂閱者的頻率。

每當在佇列管理程式中新增或移除訂閱時，只要它是該特定主題字串的第一個或最後一個訂閱，就會將該資訊傳遞給拓撲中的其他佇列管理程式。在直接遞送叢集和階層中，不論此訂閱資訊是否具有主題的發佈者，都會將此訂閱資訊傳送至拓撲中的每個佇列管理程式。如果拓撲是由許多佇列管理程式所組成，這可能是一個顯著的效能額外負擔。在主題主機遞送叢集中，此資訊只會延伸到管理對映至訂閱主題字串之叢集主題的那些佇列管理程式。

另請參閱第 74 頁的『[發佈/訂閱叢集作業: 最佳作法](#)』的 [訂閱變更及動態主題字串](#) 一節。

註：在非常動態的系統中，許多唯一主題字串集快速且持續變更，因此最好將模型切換至"隨處發佈"模式。請參閱 [發佈/訂閱網路中的訂閱效能](#)。

- b) 請考量佇列管理程式在拓撲中的動態程度。

階層需要手動從階層中插入或移除拓撲中佇列管理程式的每一項變更，在階層中的較高層次變更佇列管理程式時，請小心。階層中的佇列管理程式通常也會使用手動配置的通道連線。您必須維護這些連線，在從階層新增及移除佇列管理程式時，新增及移除通道。

在發佈/訂閱叢集中，佇列管理程式會自動連接至第一次加入叢集時所需的任何其他佇列管理程式，並自動察覺主題及訂閱。

- 請考量您的路徑可用性及發佈資料流量可調整性需求。

- a) 決定您是否需要一律具有從發佈佇列管理程式到訂閱佇列管理程式的可用路徑，即使佇列管理程式無法使用也一樣。
- b) 請考量您需要網路的可擴充性。決定發佈資料流量的層次是否太高而無法透過單一佇列管理程式或通道遞送，以及該層次的發佈資料流量是否必須由單一主題分支處理，或可以分散在多個主題分支中。
- c) 請考量您是否需要維護訊息排序。

因為直接遞送叢集會將訊息從發佈佇列管理程式直接傳送至訂閱佇列管理程式，所以您不需要考量中繼佇列管理程式在路徑上的可用性。同樣地，不考量調整至中繼佇列管理程式。不過，如先前所述，叢集中所有佇列管理程式之間自動維護通道及資訊流程的額外負擔可能會顯著影響效能，尤其是在大型或動態環境中。

主題主機遞送叢集可以針對個別主題進行調整。您可以確保在不同的佇列管理程式上定義主題樹狀結構的每一個分支(具有相當大的發佈工作量)，且每一個佇列管理程式都有足夠的效能，且可供主題樹狀結構該分支的預期工作量使用。您也可以在多個佇列管理程式上定義每一個主題，以進一步增進可用性與水平調整。這可讓系統繞過無法使用的主題主機佇列管理程式，並在它們之間平衡發佈資料流量的工作量。不過，當您在多個佇列管理程式上定義給定主題時，也會建立下列限制：

- 您會在發佈之間失去訊息訂購。
- 您無法使用保留的發佈資訊。請參閱第 83 頁的『[發佈/訂閱叢集中保留發佈的設計考量](#)』。

您無法透過多個路徑在階層中配置遞送的高可用性或可調整性。

另請參閱第 74 頁的『[發佈/訂閱叢集作業: 最佳作法](#)』的 [Publication traffic](#) 一節。

- 根據這些計算，使用提供的鏈結可協助您決定是否使用主題主機遞送叢集、直接遞送叢集、階層或這些拓撲的混合。

下一步

您現在已準備好配置分散式發佈/訂閱網路。

相關資訊

[配置佇列管理程式叢集](#)

[配置分散式佇列](#)

[配置發佈/訂閱叢集](#)

[將佇列管理程式連接至發佈/訂閱階層](#)

設計發佈/訂閱叢集

有兩種基本發佈/訂閱叢集拓撲: 直接遞送 和 主題主機遞送。 每一個都有不同的好處。 當您設計發佈/訂閱叢集時, 請選擇最符合預期網路需求的拓撲。

如需兩個發佈/訂閱叢集拓撲的概觀, 請參閱 [發佈/訂閱叢集](#)。 若要協助您評估網路需求, 請參閱 [第 56 頁的『規劃分散式發佈/訂閱網路』](#) 及 [第 74 頁的『發佈/訂閱叢集作業: 最佳作法』](#)。

一般而言, 兩個叢集拓撲都提供下列好處:

- 點對點叢集拓撲上的簡式配置。
- 自動處理加入及離開叢集的佇列管理程式。
- 透過新增額外佇列管理程式並在它們之間配送額外訂閱和發佈者, 輕鬆擴充其他訂閱和發佈者。

不過, 隨著需求變得更具體, 這兩種拓撲有不同的好處。

直接遞送發佈/訂閱叢集

使用直接遞送, 叢集中的任何佇列管理程式會將來自已連接應用程式的發佈, 直接傳送至叢集中具有相符訂閱的任何其他佇列管理程式。

直接遞送的發佈/訂閱叢集提供下列好處:

- 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。相較於主題主機遞送拓撲或階層式拓撲, 這可以增進效能。
- 因為所有佇列管理程式都直接彼此連接, 所以在此拓撲的遞送基礎架構中沒有單一失敗點。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。
- 配置非常簡單, 尤其是在現有叢集上。

使用直接遞送發佈/訂閱叢集時要考量的事項:

- 叢集中的所有佇列管理程式會感知叢集中的所有其他佇列管理程式。
- 管理叢集主題的一或多個訂閱的叢集中的佇列管理程式, 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, 即使那些佇列管理程式未在任何叢集主題上發佈訊息也一樣。
- 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。在叢集化主題下使用的個別主題字串越多, 訂閱變更率越高, 佇列管理程式之間的通訊就會越多。
- 即使佇列管理程式既未發佈也未訂閱這些主題, 叢集中的每個佇列管理程式都會維護其已通知之已訂閱主題字串的知識。

基於上述原因, 叢集中所有已定義直接遞送主題的佇列管理程式都會產生額外負擔。叢集中的佇列管理程式越多, 額外負擔就越大。同樣地, 訂閱的主題字串越多, 其變更率越高, 額外負擔也越大。這可能會導致在大型或動態直接遞送發佈/訂閱叢集中小型系統上執行的佇列管理程式負載過多。如需進一步資訊, 請參閱 [直接遞送發佈/訂閱效能](#)。

當您知道叢集無法容納直接遞送叢集發佈/訂閱的額外需要時, 您可以改用 [主題主機遞送發佈/訂閱](#)。或者, 在極端狀況下, 您可以在叢集中的每個佇列管理程式上, 將佇列管理程式屬性 **PSCLUS** 設為 **DISABLED**, 以完全停用叢集化發佈/訂閱功能。請參閱 [第 82 頁的『禁止叢集發佈/訂閱』](#)。這會防止建立任何叢集主題, 因此可確保您的網路不會產生與叢集發佈/訂閱相關聯的任何額外負擔。

主題主機遞送的發佈/訂閱叢集

使用主題主機遞送，以管理方式定義叢集主題的佇列管理程式會變成發佈的路由器。叢集中非管理佇列管理程式的發佈會透過管理佇列管理程式遞送至叢集中具有相符訂閱的任何佇列管理程式。

主題主機遞送的發佈/訂閱叢集比直接遞送的發佈/訂閱叢集提供下列額外好處：

- 只有定義主題主機遞送主題的佇列管理程式才會知道叢集中的所有其他佇列管理程式。
- 只有主題主機佇列管理程式需要能夠連接至叢集中的所有其他佇列管理程式，且通常只會連接至存在訂閱的那些佇列管理程式。因此，在佇列管理程式之間執行的通道會大幅減少。
- 管理叢集主題的一或多個訂閱的叢集佇列管理程式，只會自動建立叢集傳送端通道至管理叢集主題 (對映至訂閱的主題字串) 的佇列管理程式。
- 佇列管理程式中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管理叢集主題的佇列管理程式。同樣地，要刪除之主題字串的前次訂閱也會產生訊息。在叢集化主題下使用的個別主題字串越多，以及訂閱變更率越高，則會發生佇列管理程式之間的通訊越多，但只會在訂閱主機與主題主機之間進行。
- 對實體配置有更多控制。透過直接遞送，所有佇列管理程式都必須參與發佈/訂閱叢集，以增加其額外需要。使用主題主機遞送，只有主題主機佇列管理程式才知道其他佇列管理程式及其訂閱。您明確選擇主題主機佇列管理程式，因此您可以確保那些佇列管理程式在適當的設備上執行，並且您可以對其他佇列管理程式使用較不強大的系統。

使用主題主機遞送發佈/訂閱叢集時要考量的事項：

- 當發佈者或訂閱者不在管理佇列管理程式的主題上時，會在發佈佇列管理程式與訂閱佇列管理程式之間建立額外 "躍點"。額外 "躍點" 所造成的延遲，可能表示主題主機遞送的效率低於直接遞送。
- 在大型叢集上，主題主機遞送可讓您輕鬆解決直接遞送可能造成的重大效能和調整大小問題。
- 您可以選擇在單一佇列管理程式或極少數佇列管理程式上定義所有主題。如果這樣做，請確定主題主機佇列管理程式是在具有良好連線功能的功能強大系統上管理。
- 您可以在多個佇列管理程式上定義相同的主題。這會增進主題的可用性，也會增進可調整性，因為 IBM MQ 工作量會在該主題的所有主機之間平衡某個主題的發佈。不過請注意，在多個佇列管理程式上定義相同主題會失去該主題的訊息順序。
- 透過在不同佇列管理程式上管理不同的主題，您可以提高可調整性，而不會遺失訊息順序。

相關資訊

[發佈/訂閱叢集實務](#)

[配置發佈/訂閱叢集](#)

[調整分散式發佈/訂閱網路](#)

[分散式發佈/訂閱疑難排解](#)

發佈/訂閱叢集中的直接遞送

來自任何發佈佇列管理程式的發佈會直接遞送至叢集中具有相符訂閱的任何其他佇列管理程式。

如需如何在發佈/訂閱階層及叢集中的佇列管理程式之間遞送訊息的簡介，請參閱 [分散式發佈/訂閱網路](#)。

直接遞送的發佈/訂閱叢集行為如下：

- 所有佇列管理程式都會自動知道所有其他佇列管理程式。
- 所有具有叢集主題訂閱的佇列管理程式都會建立與叢集中所有其他佇列管理程式的通道，並將其訂閱通知給它們。
- 應用程式所發佈的訊息會從它所連接的佇列管理程式遞送至符合的訂閱所在的每一個佇列管理程式。

下圖顯示目前未用於發佈/訂閱或點對點活動的佇列管理程式叢集。請注意，叢集中的每一個佇列管理程式只會連接至完整儲存庫佇列管理程式，以及從完整儲存庫佇列管理程式連接。

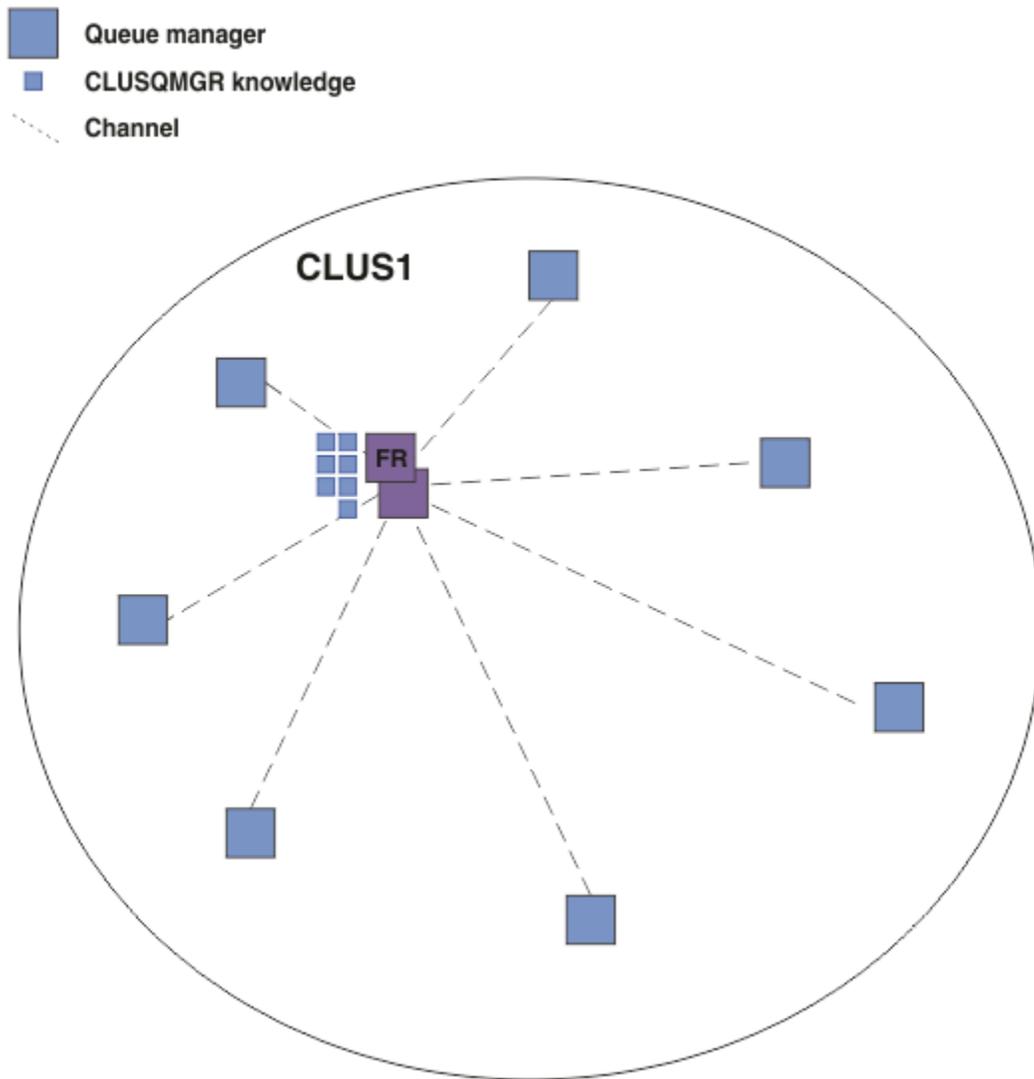


圖 16: 佇列管理程式叢集

為了讓發佈資訊在直接遞送叢集中的佇列管理程式之間流動，您將主題樹狀結構的分支建立叢集 (如 [配置發佈/訂閱叢集中](#) 所述)，並指定 [直接遞送](#) (預設值)。

在直接遞送發佈/訂閱叢集中，您可以在叢集中的任何佇列管理程式上定義主題物件。當您這樣做時，完整儲存庫佇列管理程式會自動將物件的知識及叢集中所有其他佇列管理程式的知識推送至叢集中的所有佇列管理程式。在任何佇列管理程式參照主題之前，會發生此情況：

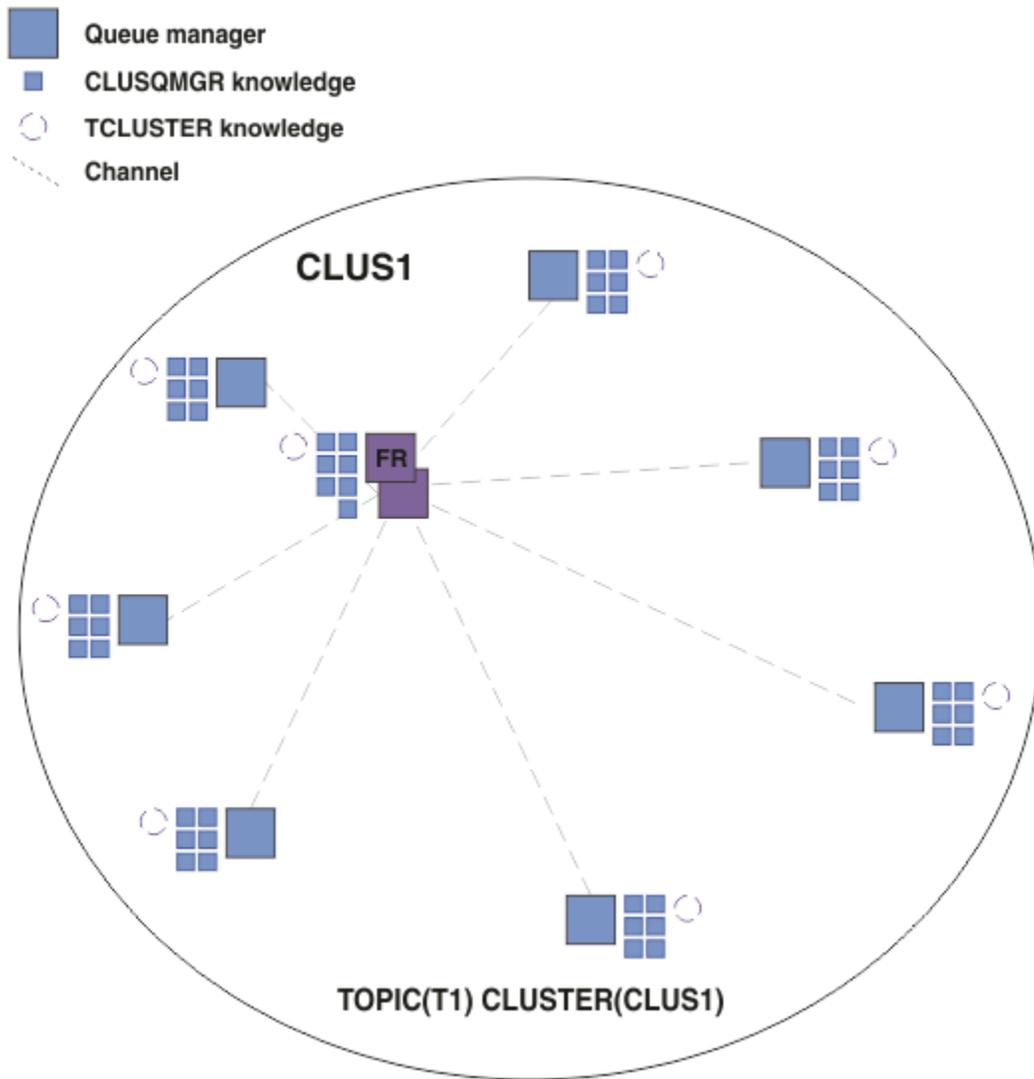


圖 17: 直接遞送的發佈/訂閱叢集

建立訂閱時，管理訂閱的佇列管理程式會建立與叢集中每個佇列管理程式的通道，並傳送訂閱的詳細資料。此分散式訂閱知識由每一個佇列管理程式上的 Proxy 訂閱 代表。在叢集中任何符合該 Proxy 訂閱主題字串的佇列管理程式上產生發佈資訊時，會建立從發佈者佇列管理程式到每一個管理訂閱的佇列管理程式的叢集通道，並將訊息傳送至每一個佇列管理程式。

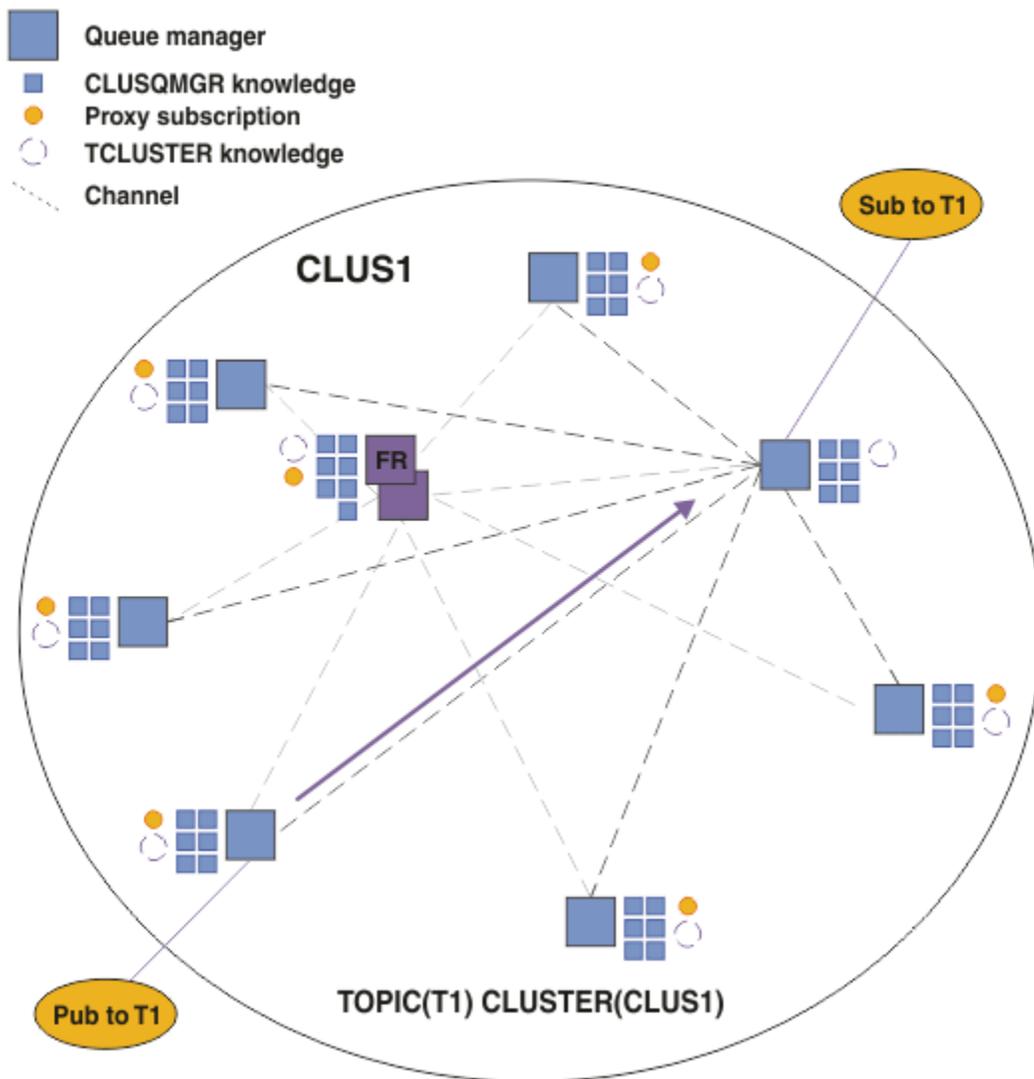


圖 18: 具有叢集主題的發佈者和訂閱者的直接遞送發佈/訂閱叢集

將發佈直接遞送至訂閱管理佇列管理程式，可簡化配置，並將將發佈遞送至訂閱的延遲降至最低。

不過，視訂閱及發佈者的位置而定，您的叢集可以快速變成完全交互連接，且每個佇列管理程式都具有與所有其他佇列管理程式的直接連線。這在您的環境中不一定可以接受。同樣地，如果要訂閱的主題字串集經常變更，則在所有佇列管理程式之間傳送該資訊的額外負擔也會變得很重要。直接遞送發佈/訂閱叢集中的所有佇列管理程式都必須能夠應付這些額外需要。

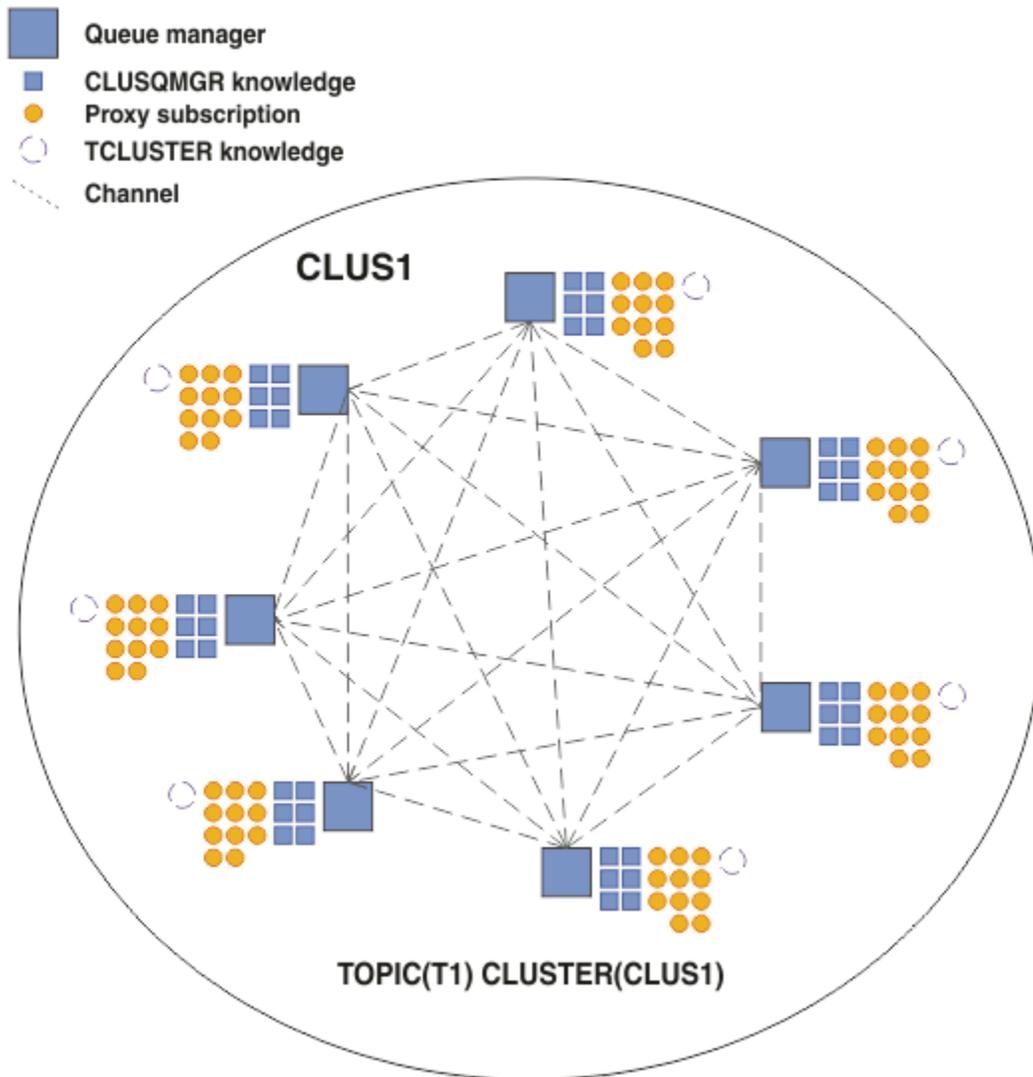


圖 19: 完全交互連接的直接遞送發佈/訂閱叢集

摘要及其他考量

直接遞送的發佈/訂閱叢集很少需要人為介入來建立或管理，並在發佈者與訂閱者之間提供直接遞送。對於某些配置，它通常是最適當的拓撲，尤其是具有較少佇列管理程式的叢集，或可接受高佇列管理程式連線功能且訂閱不常變更的叢集。不過，它也會對您的系統施加某些限制：

- 每一個佇列管理程式上的負載與叢集中的佇列管理程式總數成比例。因此，在較大的叢集中，個別佇列管理程式及整個系統可能會遇到效能問題。
- 依預設，所有訂閱的叢集主題字串都會延伸到整個叢集，發佈資訊只會延伸到訂閱相關聯主題的遠端佇列管理程式。因此，對訂閱集的快速變更可能成為限制因素。您可以變更此預設行為，並改為將所有發佈延伸到所有佇列管理程式，這樣就不需要 Proxy 訂閱。這會減少訂閱知識資料流量，但可能會增加發佈資料流量及每個佇列管理程式建立的通道數目。請參閱 [發佈/訂閱網路中的訂閱效能](#)。

註：類似的限制也適用於階層。

- 由於發佈/訂閱佇列管理程式的交互連接本質，Proxy 訂閱需要時間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，因此在訂閱新的主題字串之後，可能不會傳送早期發佈。您可以移除訂閱延遲所造成的問題，方法是讓所有發佈延伸到所有佇列管理程式，這樣就不需要 Proxy 訂閱。請參閱 [發佈/訂閱網路中的訂閱效能](#)。

註：此限制也適用於階層。

在使用直接遞送之前，請先探索 [第 65 頁的『發佈/訂閱叢集中的主題主機遞送』](#) 及 [第 85 頁的『發佈/訂閱階層中的遞送』](#) 中詳述的替代方法。

發佈/訂閱叢集中的主題主機遞送

叢集中非管理佇列管理程式的發佈會透過管理佇列管理程式遞送至叢集中具有相符訂閱的任何佇列管理程式。

如需如何在發佈/訂閱階層及叢集中的佇列管理程式之間遞送訊息的簡介，請參閱 [分散式發佈/訂閱網路](#)。

如果要瞭解主題主機遞送的行為和好處，最好先瞭解 [第 60 頁的『發佈/訂閱叢集中的直接遞送』](#)。

主題主機遞送的發佈/訂閱叢集行為如下：

- 叢集中的受管理主題物件是在叢集中的個別佇列管理程式上手動定義。這些稱為主題主機佇列管理程式。
- 在叢集佇列管理程式上進行訂閱時，會建立從訂閱主機佇列管理程式到主題主機佇列管理程式的通道，並且只會在管理主題的佇列管理程式上建立 Proxy 訂閱。
- 當應用程式將資訊發佈至主題時，所連接的佇列管理程式一律會將發佈轉遞至管理該主題的一個佇列管理程式，這會將它傳遞至叢集中具有該主題相符訂閱的所有佇列管理程式。

下列範例將更詳細地說明此程序。

使用單一主題主機遞送主題主機

若要讓發佈資訊在主題主機遞送叢集中的佇列管理程式之間流動，您可以建立主題樹狀結構分支的叢集，如 [配置發佈/訂閱叢集中所述](#)，並指定主題主機遞送。

在叢集中的多個佇列管理程式上定義主題主機遞送主題物件有許多原因。不過，為了簡單起見，我們從單一主題主機開始。

下圖顯示目前未用於發佈/訂閱或點對點活動的佇列管理程式叢集。請注意，叢集中的每一個佇列管理程式只會連接至完整儲存庫佇列管理程式，以及從完整儲存庫佇列管理程式連接。

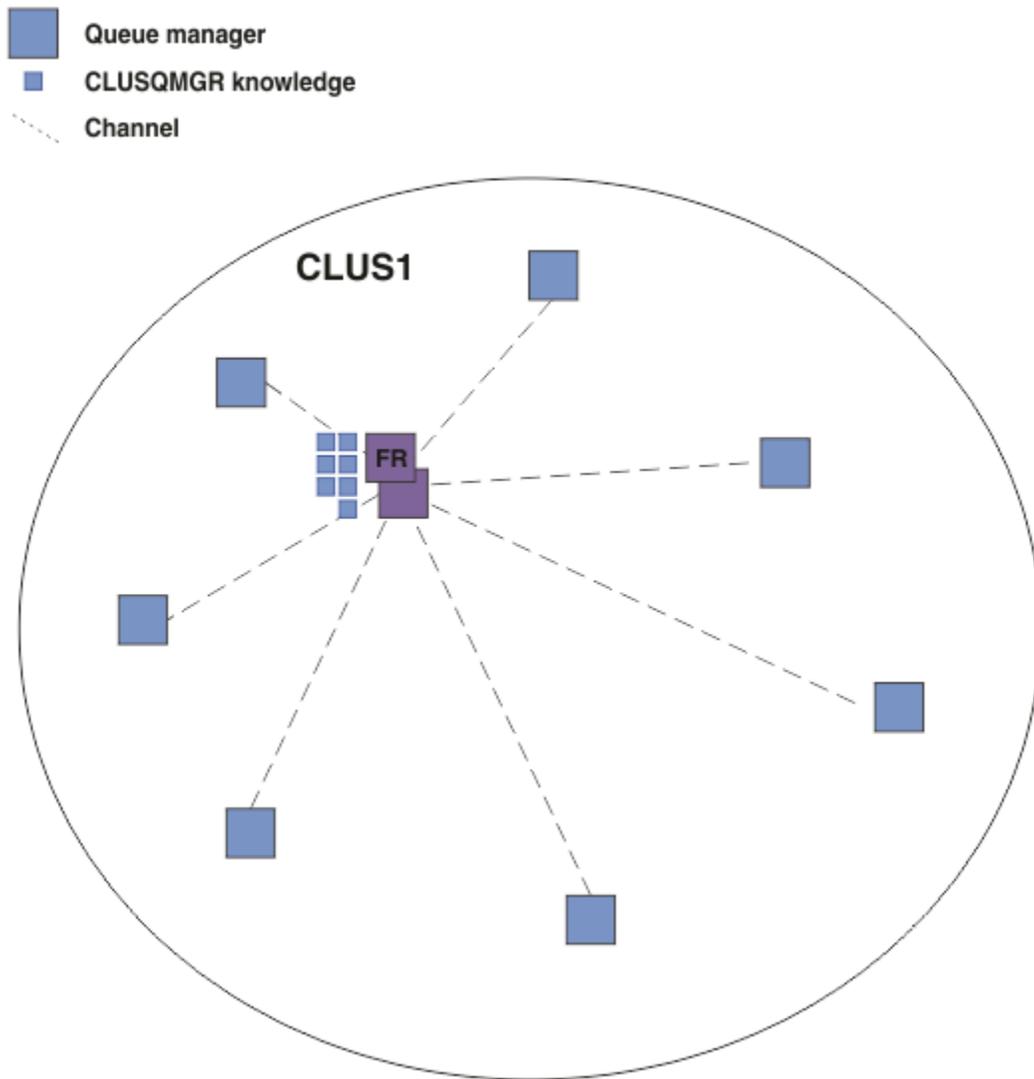


圖 20: 佇列管理程式叢集

在主題主機遞送的發佈/訂閱叢集中，您可以在叢集中的特定佇列管理程式上定義主題物件。然後，發佈/訂閱資料流量會流經該佇列管理程式，使其成為叢集中的重要佇列管理程式，並增加其工作量。基於這些原因，不建議使用完整儲存庫佇列管理程式，而是使用叢集中的另一個佇列管理程式。當您在主機佇列管理程式上定義主題物件時，完整儲存庫佇列管理程式會自動將該物件及其主機的知識推送至叢集中的所有其他佇列管理程式。請注意，與直接遞送不同，每一個佇列管理程式不會被告知叢集中的所有其他佇列管理程式。

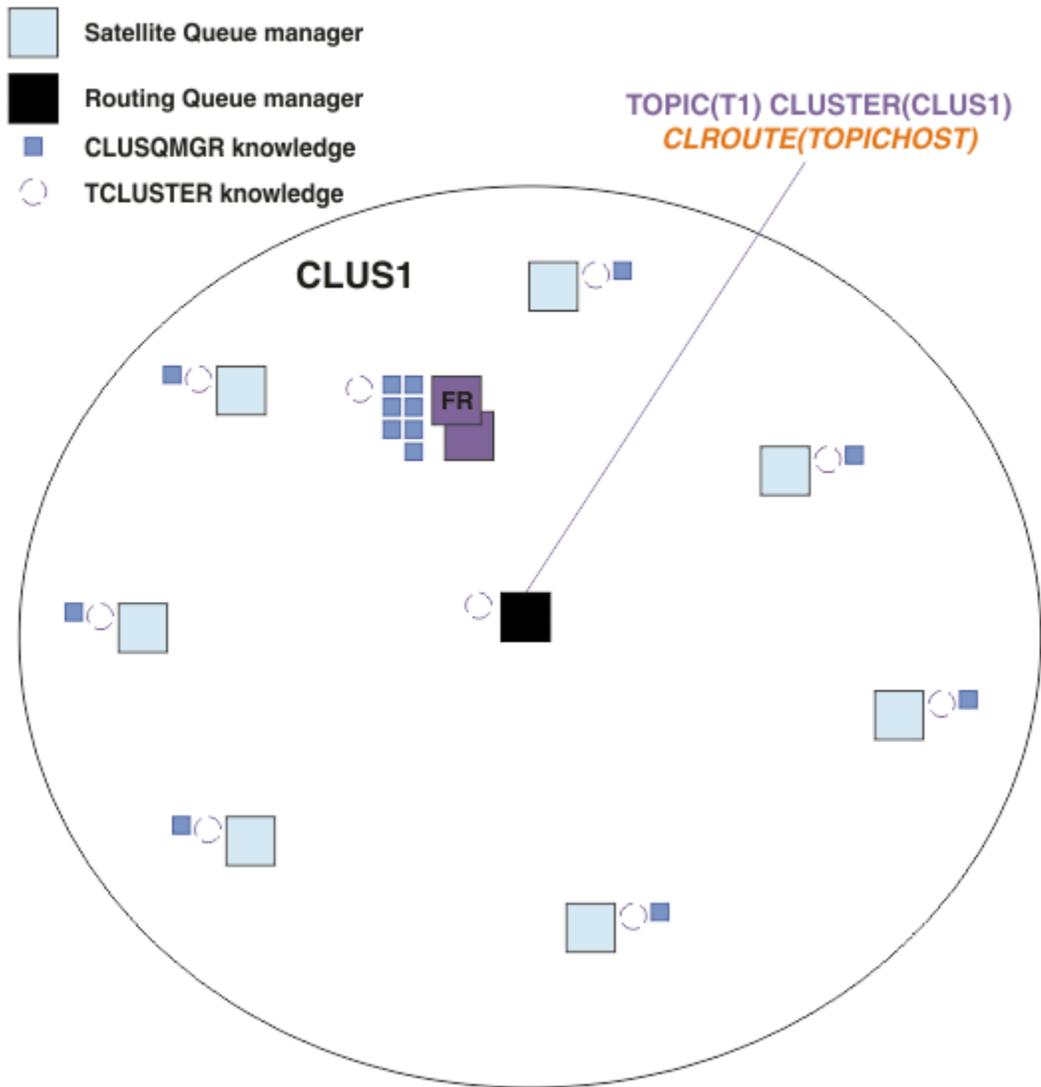


圖 21: 主題主機遞送的發佈/訂閱叢集，其中一個主題定義在一個主題主機上

在佇列管理程式上建立訂閱時，會在訂閱佇列管理程式與主題主機佇列管理程式之間建立通道。訂閱佇列管理程式僅連接至主題主機佇列管理程式，並傳送訂閱的詳細資料 (以 *Proxy* 訂閱形式)。主題主機佇列管理程式不會將此訂閱資訊轉遞至叢集中任何進一步的佇列管理程式。

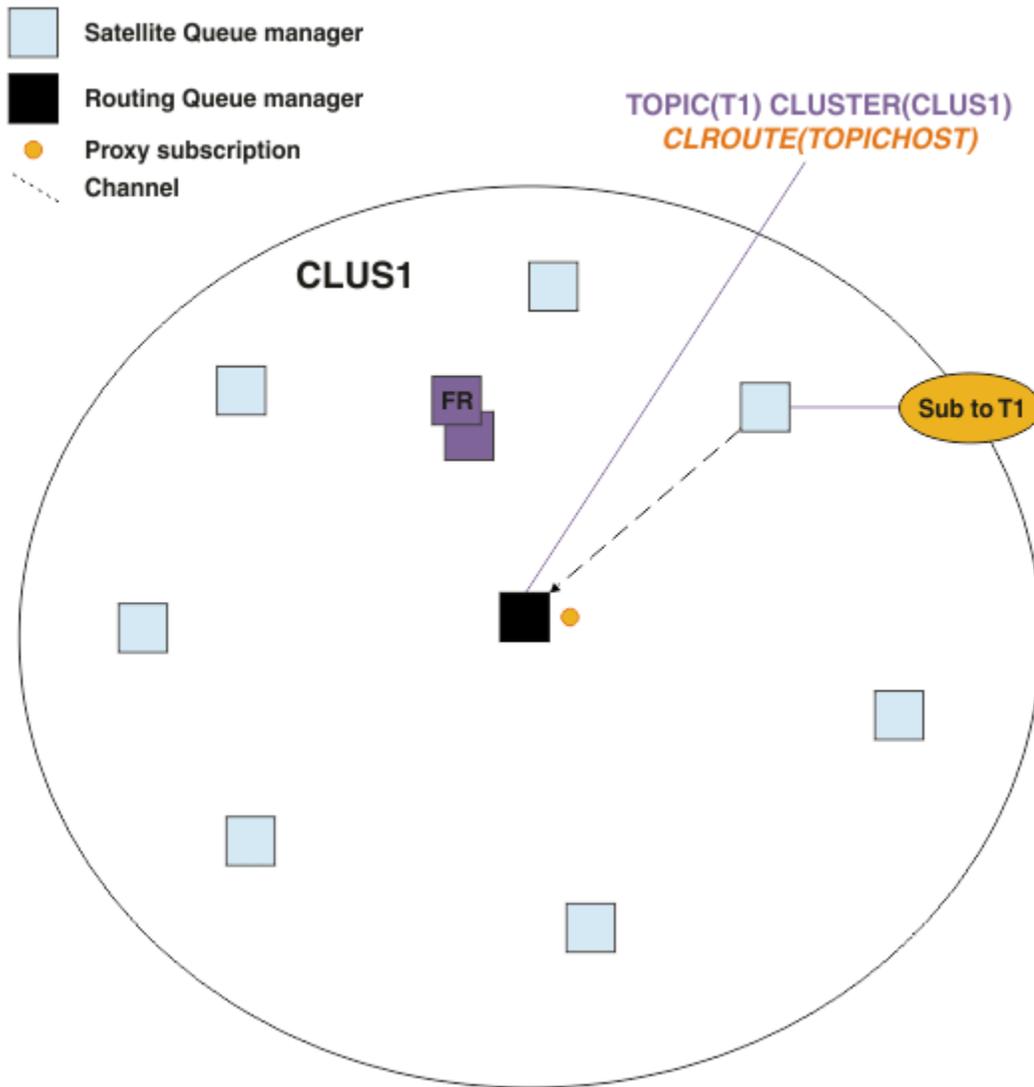


圖 22: 主題主機已遞送發佈/訂閱叢集，其中一個主題定義在一個主題主機上，一個訂閱者

當發佈應用程式連接至另一個佇列管理程式並發佈訊息時，會在發佈佇列管理程式與主題主機佇列管理程式之間建立通道，並將訊息轉遞至該佇列管理程式。發佈佇列管理程式不知道叢集中其他佇列管理程式上的任何訂閱，因此即使叢集中沒有該主題的訂閱者，訊息也會轉遞至主題主機佇列管理程式。發佈佇列管理程式只會連接至主題主機佇列管理程式。發佈會透過主題主機遞送至訂閱佇列管理程式 (如果有的話)。

直接滿足與發佈者相同的佇列管理程式上的訂閱，而不先將訊息傳送至主題主機佇列管理程式。

請注意，由於每一個主題主機佇列管理程式所扮演的重要角色，您必須選擇可處理主題主機作業之負載、可用性與連線功能需求的佇列管理程式。

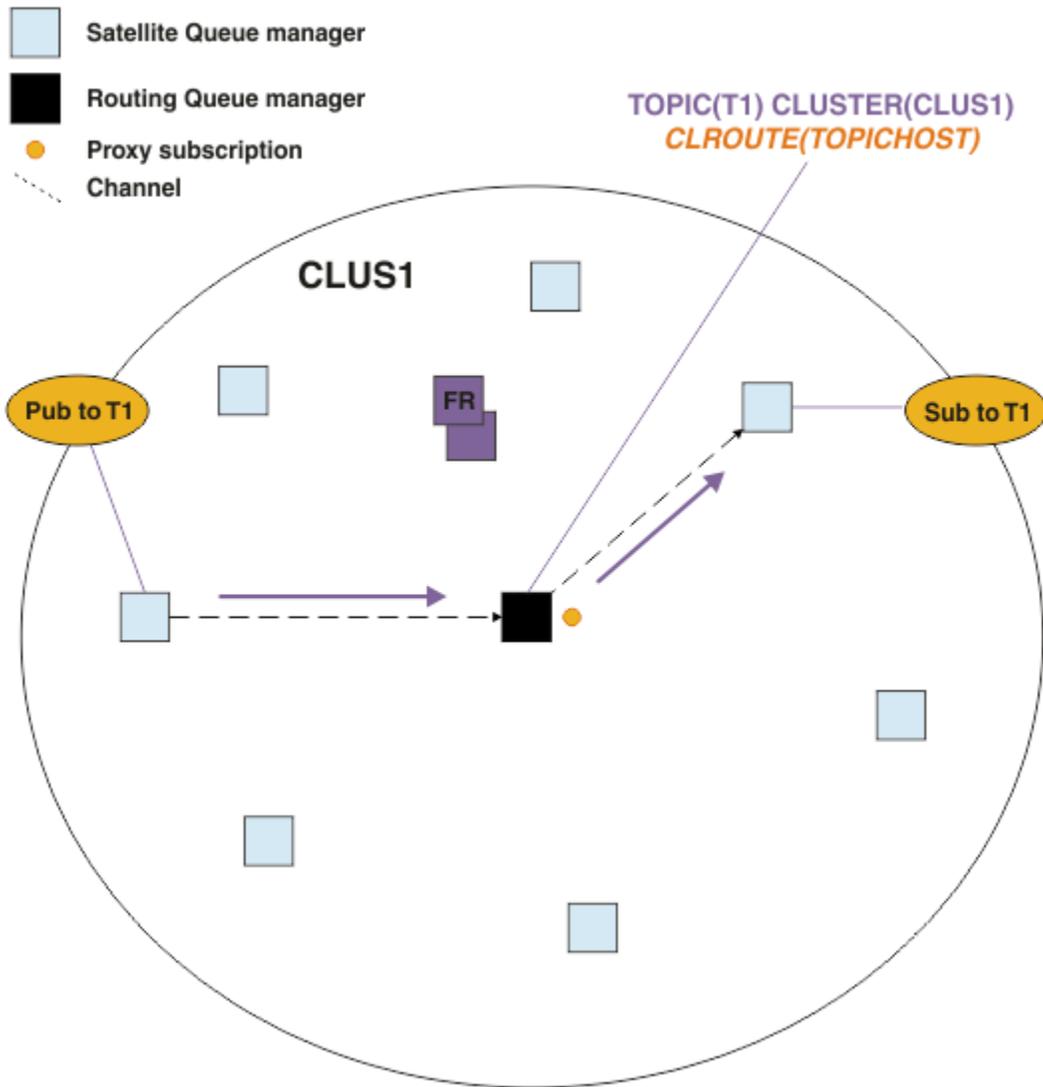


圖 23: 主題主機遞送發佈/訂閱叢集，具有一個主題、一個訂閱者及一個發佈者

在多個佇列管理程式之間劃分主題樹狀結構

遞送主題管理佇列管理程式只負責與配置其受管理主題物件之主題樹狀結構分支相關的訂閱知識及發佈訊息。如果叢集中的不同發佈/訂閱應用程式使用不同的主題，您可以配置不同的佇列管理程式來管理主題樹狀結構的不同叢集分支。這可透過減少叢集中每個主題主機佇列管理程式上的發佈資料流量、訂閱知識及通道來進行調整。您應該對主題樹狀結構的不同大量分支使用此方法：

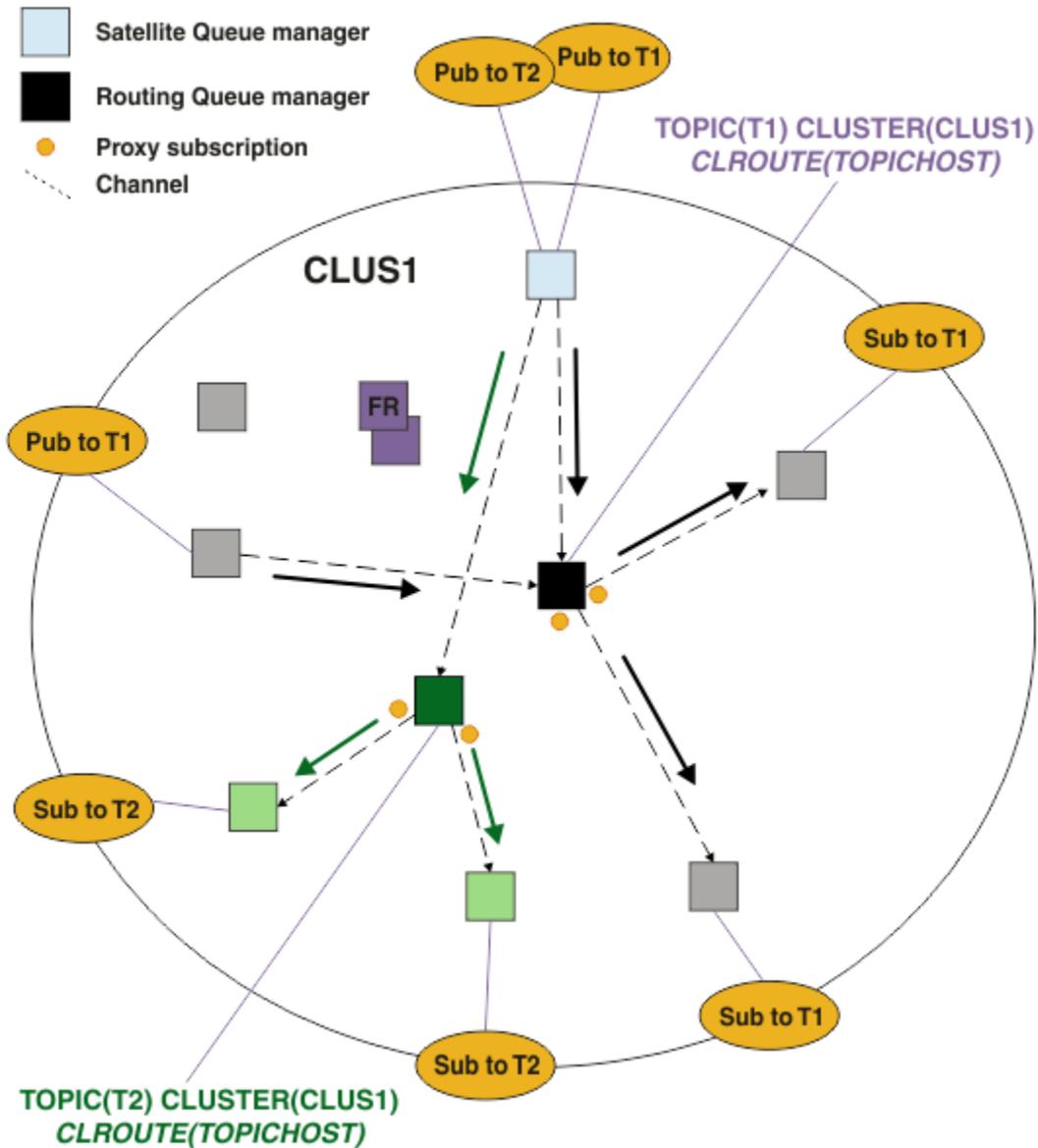


圖 24: 主題主機遞送發佈/訂閱叢集，具有兩個主題，每個主題定義在一個主題主機上

例如，使用 [主題樹狀結構](#) 中說明的主題，如果主題 T1 配置了主題字串 /USA/Alabama，而主題 T2 配置了主題字串 /USA/Alaska，則發佈至 /USA/Alabama/Mobile 的訊息將透過管理 T1 的佇列管理程式遞送。發佈至 /USA/Alaska/Juneau 的訊息會透過管理 T2 的佇列管理程式來遞送。

註: 您無法在主題樹狀結構中使用比叢集化點更高的萬用字元，讓單一訂閱跨越主題樹狀結構的多個叢集化分支。請參閱 [萬用字元訂閱](#)。

針對單一主題使用多個主題主機來遞送主題主機

如果單一佇列管理程式負責遞送主題，且該佇列管理程式變成無法使用或無法處理工作量，則發佈不會立即流向訂閱。

當您只在一個佇列管理程式上定義主題時，如果您需要更大的備援、可調整性及工作量平衡，則可以在多個佇列管理程式上定義主題。每一個發佈的個別訊息都會透過單一主題主機遞送。存在多個相符主題主機定義時，會選擇其中一個主題主機。選擇的方式與叢集佇列相同。這容許將訊息遞送至可用的主題主機，避免任何無法使用的訊息，並容許在多個主題主機佇列管理程式及通道之間平衡訊息負載的工作量。不過，當您針對叢集中的相同主題使用多個主題主機時，不會維護跨多則訊息的排序。

下圖顯示已在兩個佇列管理程式上定義相同主題的主題主機遞送叢集。在此範例中，訂閱端佇列管理程式會以 Proxy 訂閱形式，將已訂閱主題的相關資訊傳送至這兩個主題主機佇列管理程式：

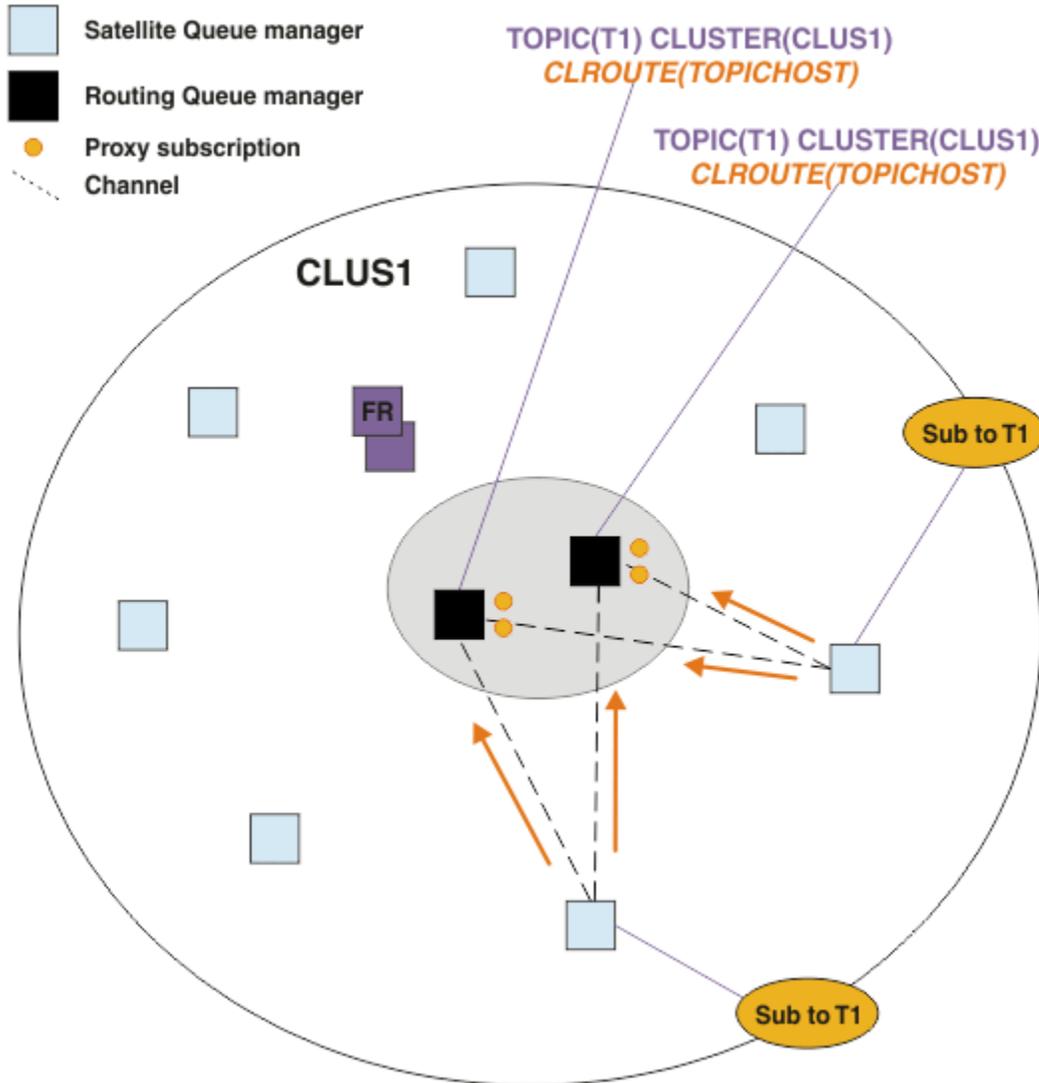


圖 25: 在多主題主機發佈/訂閱叢集中建立 Proxy 訂閱

從非管理的佇列管理程式建立發佈資訊時，佇列管理程式會將發佈資訊副本傳送至該主題的其中一個主題主機佇列管理程式。系統會根據叢集工作量管理演算法的預設行為來選擇主機。在一般系統中，這會近似於在每一個主題主機佇列管理程式之間的循環式分佈。來自相同發佈應用程式的訊息之間沒有親緣性；這等於使用 NOTfixed 叢集連結類型。

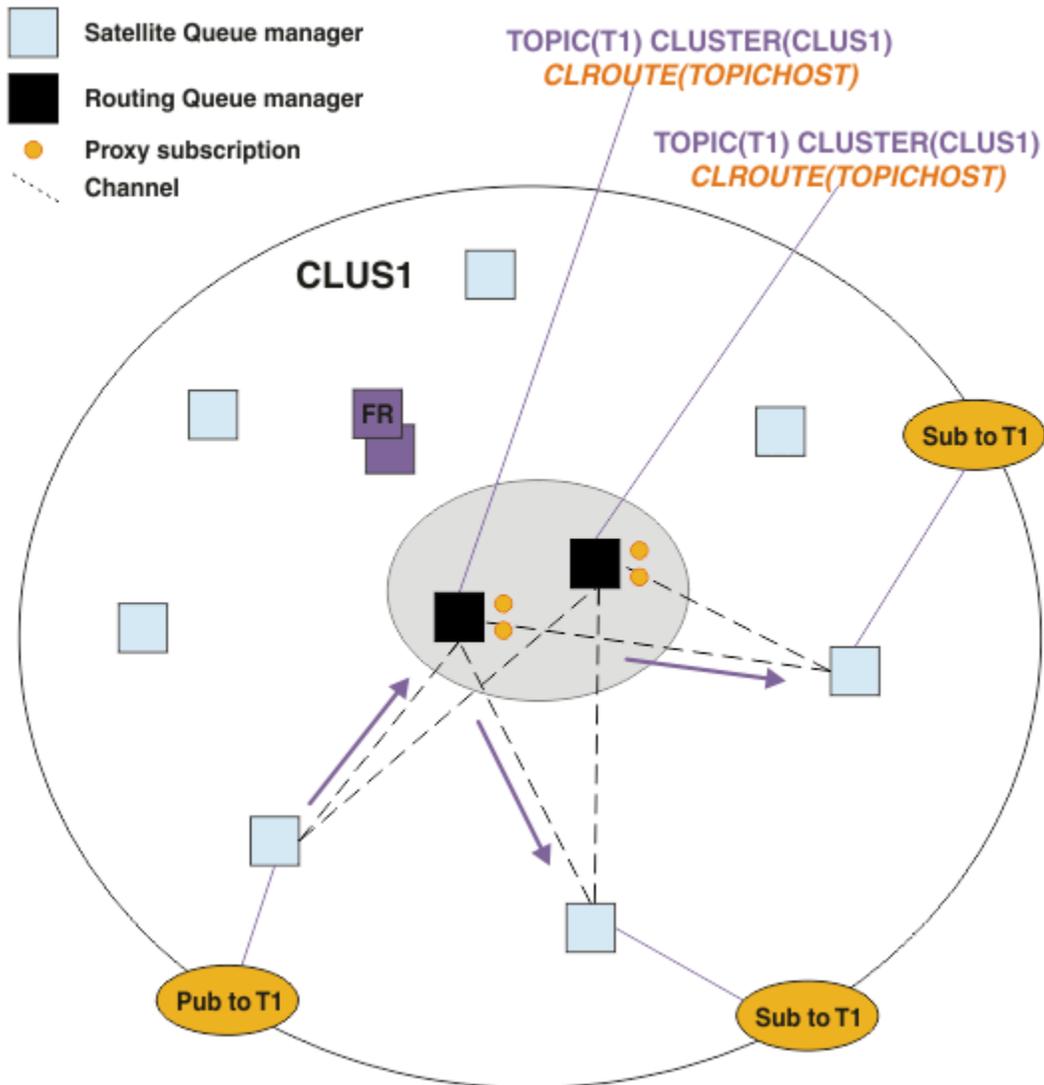


圖 27: 將發佈遞送至多主題主機發佈/訂閱叢集中的訂閱者

在主題主機佇列管理程式本端建立訂閱和發佈者

上述範例顯示在非管理受管理遞送主題物件的佇列管理程式上，發佈者與訂閱者之間的遞送。在這些拓撲中，訊息需要多個躍點才能達到訂閱。

如果不需要其他躍點，則可能適合將金鑰發佈者連接至管理佇列管理程式的主題。不過，如果一個主題有多個主題主機，且只有一個發佈者，則所有發佈資料流量都會透過發佈者所連接的主題主機佇列管理程式來遞送。

同樣地，如果有金鑰訂閱，這些可能位於主題主機佇列管理程式上。不過，如果遞送主題有多個主機，則只有部分發佈會避免額外的中繼站，其餘部分會先透過其他主題主機佇列管理程式遞送。

這裡會進一步說明這類拓撲: 使用集中式發佈者或訂閱者的主題主機遞送。

註: 如果在與遞送主題主機共同尋找發佈者或訂閱時變更遞送主題配置，則需要特殊規劃。例如，請參閱 [將額外主題主機新增至主題主機遞送叢集](#)。

摘要及其他考量

主題主機遞送發佈/訂閱叢集可讓您精確控制哪些佇列管理程式管理每一個主題，而那些佇列管理程式會變成主題樹狀結構該分支的遞送佇列管理程式。此外，沒有訂閱或發佈者的佇列管理程式不需要連接主題主機佇列管理程式，而具有訂閱的佇列管理程式不需要連接至未管理主題的佇列管理程式。此配置可大幅減少叢

集中佇列管理程式之間的連線數目，以及在佇列管理程式之間傳遞的資訊量。在只有一部分佇列管理程式執行發佈/訂閱工作的大型叢集中尤其如此。此配置也可讓您對叢集中個別佇列管理程式的負載進行部分控制，因此 (例如) 您可以選擇在更強大且更具復原力的系統上管理高度活躍的主題。對於某些配置 (特別是較大的叢集)，它通常是比直接遞送更適當的拓撲。

不過，主題主機遞送還是會對您的系統加諸一些限制：

- 系統配置和維護需要比直接遞送更多的規劃。您需要決定要在主題樹狀結構中形成叢集的時間點，以及主題定義在叢集的位置。
- 就像直接遞送主題一樣，在定義新的主題主機遞送主題時，該資訊會推送到完整儲存庫佇列管理程式，再從中導向至叢集的所有成員。此事件會導致從完整儲存庫啟動通往叢集每一個成員的通道 (如果尚未啟動)。
- 一律從非主機佇列管理程式將這些發佈資訊傳送至主機佇列管理程式，即使叢集中沒有訂閱也一樣。因此，當一般預期會有訂閱存在時，或當廣域連線功能和知識的超載量大於額外發佈資料流量的風險時，您應使用遞送主題。

註：如先前所述，讓發佈者在主題主機本端可降低此風險。

- 在非主機佇列管理程式上發佈的訊息不會直接進入管理訂閱的佇列管理程式，而是一律透過主題主機佇列管理程式來遞送。此方法可能會增加叢集的總超載量，因而增加訊息延遲時間並降低效能。

註：如先前所述，讓訂閱或發佈者在主題主機本端可降低此風險。

- 使用單一主題主機佇列管理程式，為發佈到某個主題的所有訊息建立單一失敗點。您可以透過定義多個主題主機來移除這個單一失敗點。不過，擁有多個主機將影響訂閱接收已發佈訊息的順序。
- 主題主機佇列管理程式將產生額外訊息負載，因為它們需要處理來自多個佇列管理程式的發佈資料流量。可以減輕這項負載：將多個主題主機用於單一主題 (在此情況下，不會維護訊息順序)，或使用不同的佇列管理程式來管理主題樹狀結構的不同分支的遞送主題。

在使用主題主機遞送之前，請先探索 [第 60 頁的『發佈/訂閱叢集中的直接遞送』](#) 及 [第 85 頁的『發佈/訂閱階層中的遞送』](#) 中詳述的替代方法。

發佈/訂閱叢集作業: 最佳作法

使用叢集主題可讓在佇列管理程式之間延伸發佈/訂閱網域很簡單，但如果無法完全瞭解機制和含意，則可能會導致問題。資訊共用和發佈遞送有兩個模型。實作最符合您個別商業需求的模型，並在您選擇的叢集上執行最佳。

下列各節中的最佳作法資訊未提供「一刀切」解決方案，而是共用一般方法來解決一般問題。它假設您對 IBM MQ 叢集及發佈/訂閱傳訊有基本瞭解，且您熟悉 [分散式發佈/訂閱網路](#) 及 [第 59 頁的『設計發佈/訂閱叢集』](#) 中的資訊。

當您使用叢集來進行點對點傳訊時，叢集中的每一個佇列管理程式都會以 "需要知道" 為基礎運作。也就是說，當應用程式連接至其他叢集資源 (例如叢集中的其他佇列管理程式及叢集佇列) 要求使用它們時，它只會發現這些其他叢集資源。當您將發佈/訂閱傳訊新增至叢集時，會引進叢集佇列管理程式之間增加的資訊共用層次及連線功能。若要能夠遵循發佈/訂閱叢集的最佳作法，您需要完全瞭解此行為變更的含意。

為了讓您根據精確需求來建置最佳架構，發佈/訂閱叢集中有兩種資訊共用和發佈遞送模型: 直接遞送 和 主題主機遞送。若要做出正確的選擇，您需要同時瞭解兩個模型，以及每個模型滿足的不同需求。這些需求與 [第 56 頁的『規劃分散式發佈/訂閱網路』](#) 一起在下列各節中討論：

- [第 75 頁的『限制發佈/訂閱活動中涉及的叢集佇列管理程式數目的原因』](#)
- [第 75 頁的『如何決定要叢集的主題』](#)
- [第 75 頁的『如何調整系統大小』](#)
- [第 76 頁的『發佈者和訂閱位置』](#)
- [第 76 頁的『發佈資料流量』](#)
- [第 77 頁的『訂閱變更及動態主題字串』](#)

限制發佈/訂閱活動中涉及的叢集佇列管理程式數目的原因

在叢集中使用發佈/訂閱傳訊時，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/訂閱活動需求，並將其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主題的佇列管理程式集下限之後，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。

如果您具有已建立的叢集，且已適當地運作來進行點對點傳訊，則此方法特別有用。當您將現有大型叢集轉換為發佈/訂閱叢集時，最好一開始為發佈/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。您可以使用已在一個以上點對點叢集中的現有佇列管理程式子集，並建立新發佈/訂閱叢集的此子集成員。不過，新叢集的完整儲存庫佇列管理程式不能是任何其他叢集的成員；這會隔離現有叢集完整儲存庫的額外負載。

如果您無法建立新叢集，且必須將現有大型叢集轉換為發佈/訂閱叢集，請不要使用直接遞送模型。主題主機遞送模型通常在較大的叢集中執行得更好，因為它通常會將發佈/訂閱資訊共用及連線功能限制為主動執行發佈/訂閱工作的佇列管理程式集，並集中在管理主題的佇列管理程式上。但例外是在管理主題定義的佇列管理程式上呼叫手動重新整理訂閱資訊，此時主題主機佇列管理程式將連接至叢集中的每個佇列管理程式。請參閱 [Proxy 訂閱的重新同步](#)。

如果您確定某個叢集由於其大小或現行負載而無法用於發佈/訂閱，則最好防止此叢集非預期地進入發佈/訂閱叢集。使用 **PSCLUS** 佇列管理程式內容，可停止任何人在叢集中的任何佇列管理程式上新增叢集主題。請參閱 [第 82 頁的『禁止叢集發佈/訂閱』](#)。

如何決定要叢集的主題

請務必小心選擇要新增至叢集的主題：這些主題的主題樹狀結構越高，它們的使用就越普遍。這可能會導致超出必要傳送更多訂閱資訊及發佈。如果主題樹狀結構有多個不同的分支（其中有些需要叢集化，有些則不需要），請在每一個需要叢集化之分支的根建立受管理主題物件，並將這些物件新增至叢集。例如，如果分支 /A、/B 和 /C 需要叢集作業，請為每一個分支定義個別叢集主題物件。

註：系統會阻止您在主題樹狀結構中巢狀內嵌叢集主題定義。在每個子分支的主題樹狀結構中，只允許您將主題叢集在一個點。例如，您無法定義 /A 及 /A/B 的叢集主題物件。巢狀叢集化主題可能會導致混淆哪些叢集化物件適用於哪些訂閱，特別是當訂閱使用萬用字元時。在使用主題主機遞送時，這甚至更重要，其中遞送決策是由主題主機的配置所精確定義。

如果必須在主題樹狀結構的上層新增叢集主題，但叢集點下方樹狀結構的部分分支不需要叢集行為，則您可以使用訂閱及發佈範圍屬性來減少進一步主題的訂閱及發佈共用層次。

在未考量所看到的行為之前，您不應將主題根節點放入叢集中。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。

還有另一個原因，就是不想將根主題節點設為叢集。這是因為每個佇列管理程式都有根節點 (SYSTEM.BASE.TOPIC 主題物件) 的本端定義。當此物件叢集在叢集中的一個佇列管理程式上時，所有其他佇列管理程式都會知道它。不過，當相同物件的本端定義存在時，其內容會置換叢集物件。這會導致這些佇列管理程式如同未叢集化主題一樣運作。若要解決此問題，您需要對 SYSTEM.BASE.TOPIC 的每個定義進行叢集作業。您可以針對直接遞送定義執行此動作，但不針對主題主機遞送定義執行此動作，因為它會導致每個佇列管理程式變成主題主機。

如何調整系統大小

發佈/訂閱叢集通常會導致叢集中點對點傳訊的不同叢集通道型樣。點對點模型是「接受」的模型，但發佈/訂閱叢集具有更不分青紅皂白的訂閱扇出本質，特別是在使用直接遞送主題時。因此，請務必識別發佈/訂閱叢集中的哪些佇列管理程式將使用叢集通道來連接至其他佇列管理程式，以及在何種情況下。

下表列出在正常執行下，發佈/訂閱叢集中每個佇列管理程式預期的一般叢集傳送端及接收端通道集，視發佈/訂閱叢集中的佇列管理程式角色而定。

| 佇列管理程式角色 | 直接叢集接收端 | 直接叢集傳送端 | 主題叢集接收端 | 主題叢集寄件者 |
|----------|----------|----------|---------------------|-------------|
| 完整儲存庫 | AllQmgrs | AllQmgrs | AllQmgrs | AllQMGRs |
| 主題定義的主機 | 不適用 | 不適用 | AllSubs+AllPubs (1) | AllSubs (1) |

| 表 5: 每一種遞送方法的叢集傳送端和接收端通道。(繼續) | | | | |
|-------------------------------|-------------|-------------|----------|----------|
| 佇列管理程式角色 | 直接叢集接收端 | 直接叢集傳送端 | 主題叢集接收端 | 主題叢集寄件者 |
| 已建立訂閱 | AllPubs (1) | AllQMgrs | AllHosts | AllHosts |
| 發佈者已連接 | AllSubs (1) | AllSubs (1) | AllHosts | AllHosts |
| 沒有發佈者或訂閱者 | AllSubs (1) | 無 (1) | 無 (2) | 無 (2) |

索引鍵:

AllQMgrs

與叢集中每個佇列管理程式之間的通道。

AllSubs

與已建立訂閱的每個佇列管理程式之間的通道。

AllPubs

與已連接發佈應用程式的每個佇列管理程式之間的通道。

AllHosts

與每個佇列管理程式之間的通道，其中已配置叢集主題物件的定義。

無

叢集中沒有與其他佇列管理程式之間的通道，僅用於發佈/訂閱傳訊。

附註:

1. 如果從這個佇列管理程式重新整理 Proxy 訂閱的佇列管理程式，可能會自動建立與叢集中所有其他佇列管理程式之間的通道。
2. 如果從這個佇列管理程式重新整理 Proxy 訂閱的佇列管理程式，可能會自動建立與叢集中管理叢集主題定義的任何其他佇列管理程式之間的通道。

上表顯示主題主機遞送所使用的叢集傳送端和接收端通道通常遠少於直接遞送。如果基於容量或建立特定通道(例如，透過防火牆)的能力，叢集中的特定佇列管理程式需要考量通道連線功能，因此主題主機遞送是偏好的解決方案。

發佈者和訂閱位置

叢集發佈/訂閱可將在一個佇列管理程式上發佈的訊息遞送至叢集中任何其他佇列管理程式上的訂閱。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不利於效能。因此，您應該考慮在發佈訊息的相同佇列管理程式上建立主題的訂閱。

在叢集內使用主題主機遞送時，也請務必考量與主題管理佇列管理程式相關的訂閱及發佈者位置。當發佈者未連接至作為叢集主題之主機之佇列管理程式時，所發佈的訊息一律會傳送至管理佇列管理程式的主題。同樣地，當在非叢集主題之主機之佇列管理程式上建立訂閱時，從叢集中其他佇列管理程式發佈的訊息一律會先傳送至管理佇列管理程式的主題。更具體地說，如果訂閱位於管理主題的佇列管理程式上，但有一或多個其他佇列管理程式也管理該相同主題，則來自其他佇列管理程式的發佈比例會透過那些其他主題管理佇列管理程式來遞送。如需設計主題主機遞送發佈/訂閱叢集以將發佈者與訂閱之間的距離縮至最小的相關資訊，請參閱 [使用集中式發佈者或訂閱者的主題主機遞送](#)。

發佈資料流量

連接至叢集中某個佇列管理程式的應用程式所發佈的訊息，會利用叢集傳送端通道來傳輸至其他佇列管理程式上的訂閱。

當您使用直接遞送時，發佈的訊息會採用佇列管理程式之間最短的路徑。亦即，它們會從發佈佇列管理程式直接傳送至每一個具有訂閱的佇列管理程式。訊息不會傳輸至沒有主題訂閱的佇列管理程式。請參閱 [發佈/訂閱網路中的 Proxy 訂閱](#)。

當叢集中任何一個佇列管理程式與另一個佇列管理程式之間的發佈訊息速率偏高時，這兩個點之間的叢集通道基礎架構必須能夠維持該速率。這可能涉及調整所使用的通道及傳輸佇列。

當您使用主題主機遞送時，在非主題主機的佇列管理程式上發佈的每一則訊息都會傳輸至主題主機佇列管理程式。這與叢集中其他任何位置是否存在一個以上訂閱無關。這引入了在規劃時需要考慮的進一步因素：

- 第一次將每一個發佈資訊傳送至主題主機佇列管理程式的額外延遲是否可接受？
- 每一個主題主機佇列管理程式是否可以維持入埠及出埠發佈率？請考量在許多不同佇列管理程式上具有發佈者的系統。如果它們都將訊息傳送至一組非常小的主題管理佇列管理程式，這些主題主機可能會成為處理這些訊息並將它們遞送至訂閱佇列管理程式的瓶頸。
- 是否預期大部分已發佈的訊息將沒有相符的訂閱者？如果是這樣，且發佈這類訊息的速率很高，則最好讓發佈者的佇列管理程式成為主題主機。在該狀況下，叢集中不存在任何訂閱的任何已發佈訊息將不會傳輸至任何其他佇列管理程式。

也可以引進多個主題主機來解決這些問題，以將發佈負載分散在它們之間：

- 如果有多個不同的主題（每個主題都有發佈資料流量的比例），請考量在不同的佇列管理程式上管理它們。
- 如果主題無法分隔至不同的主題主機，請考量在多個佇列管理程式上定義相同的主題物件。這會導致發佈在每一個發佈之間進行工作量平衡，以進行遞送。不過，只有在不需要發佈訊息排序時才適用。

訂閱變更及動態主題字串

另一個考量是延伸 Proxy 訂閱對系統效能的影響。通常，當在該佇列管理程式上建立特定叢集主題字串（不只是配置的主題物件）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特定其他佇列管理程式。同樣地，當刪除特定叢集主題字串的最後一個訂閱時，會傳送 Proxy 訂閱刪除訊息。

對於直接遞送，具有訂閱的每一個佇列管理程式會將這些 Proxy 訂閱傳送至叢集中的所有其他佇列管理程式。對於主題主機遞送，每一個具有訂閱的佇列管理程式只會將 Proxy 訂閱傳送至管理該叢集主題定義的每一個佇列管理程式。因此，使用直接遞送時，叢集中的佇列管理程式越多，在它們之間維護 Proxy 訂閱的額外負擔就越高。然而，使用主題主機遞送時，叢集中的佇列管理程式數目並不是因素。

在這兩個遞送模型中，如果發佈/訂閱解決方案包含許多要訂閱的唯一主題字串，或叢集中佇列管理程式上的主題經常訂閱及取消訂閱，則會在該佇列管理程式上看到大量額外負擔，因為持續產生訊息配送及刪除 Proxy 訂閱。使用直接遞送時，將這些訊息傳送至叢集中每個佇列管理程式的需求會使此情況更加複雜。

如果訂閱變更率太高，無法容納（即使在主題主機遞送系統內），請參閱 [發佈/訂閱網路中的訂閱效能](#)，以取得如何減少 Proxy 訂閱額外負擔的相關資訊。

定義叢集主題

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給叢集的所有成員，並與本端主題相結合，以建立橫跨多個佇列管理程式的主題空間的某些部分。這可讓一個佇列管理程式中的某個主題上發佈的訊息傳遞至叢集中其他佇列管理程式的訂閱。

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，但是對叢集主題所做的任何修改都必須在定義該主題的佇列管理程式（主機）上進行，此時所做的修改會透過完整儲存庫傳送給叢集的所有成員。

當您使用直接遞送時，叢集主題定義的位置不會直接影響系統的行為，因為叢集中的所有佇列管理程式都以相同方式使用主題定義。因此，只要需要主題，您應該在任何佇列管理程式上定義主題，該佇列管理程式將成為叢集的成員，且其位於系統上的可靠性足以定期聯絡完整儲存庫佇列管理程式。

當您使用主題主機遞送時，叢集主題定義的位置非常重要，因為叢集中的其他佇列管理程式會建立通道來指向這個佇列管理程式，並將訂閱資訊和發佈資訊傳送給它。若要選擇最佳佇列管理程式來管理主題定義，您需要瞭解主題主機遞送。請參閱 [第 65 頁的『發佈/訂閱叢集中的主題主機遞送』](#)。

如果您有叢集主題及本端主題物件，則本端主題優先。請參閱 [第 79 頁的『多個同名叢集主題定義』](#)。

如需用來顯示叢集主題的指令的相關資訊，請參閱相關資訊。

叢集主題繼承

一般而言，叢集發佈/訂閱拓撲中的發佈和訂閱應用程式，不論它們連接至叢集中的哪個佇列管理程式，都預期運作相同。這就是叢集受管理主題物件延伸到叢集中每個佇列管理程式的原因。

受管理主題物件會從主題樹狀結構中較高位置的其他受管理主題物件繼承其行為。當未設定主題參數的明確值時，即會發生此繼承。

在叢集化發佈/訂閱的情況下，考量這類繼承是很重要的，因為它會產生發佈者和訂閱者根據他們所連接的佇列管理程式而有不同行為的可能性。如果叢集主題物件讓任何參數繼承自較高主題物件，則該主題在叢集中的不同佇列管理程式上可能會有不同的行為。同樣地，在主題樹狀結構中叢集化主題物件下方定義的本端定義主題物件，將表示那些較低主題仍叢集化，但本端物件可能會以不同於叢集中其他佇列管理程式的方式變更其行為。

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。如果不需要此行為，請在叢集主題上設定 **WILDCARD (BLOCK)**。這會使原始萬用字元不被視為合法訂閱，並停止它在叢集主題或其子主題上接收任何發佈 (本端或來自叢集中其他位置)。

相關資訊

[使用管理主題](#)

[使用訂閱](#)

[顯示主題](#)

[DISPLAY TPSTATUS](#)

[DISPLAY SUB](#)

叢集主題屬性

當主題物件已設定叢集名稱屬性時，主題定義會延伸到叢集中的所有佇列管理程式。每一個佇列管理程式都會使用傳播的主題屬性來控制發佈/訂閱應用程式的行為。

主題物件具有許多適用於發佈/訂閱叢集的屬性。部分控制發佈及訂閱應用程式的一般行為，而部分控制如何在叢集中使用主題。

必須以叢集中所有佇列管理程式都可以正確使用叢集主題物件定義的方式來配置叢集主題物件定義。

例如，如果要用於受管理訂閱的模型佇列 (MDURMDL 及 MNDURMDL) 設為非預設佇列名稱，該具名模型佇列必須定義在將建立受管理訂閱的所有佇列管理程式上。

同樣地，如果任何屬性設為 ASPARENT，則主題的行為將取決於主題樹狀結構中的較高節點 (請參閱 管理主題物件) 在叢集中的每一個個別佇列管理程式上。當發佈或訂閱不同的佇列管理程式時，這可能會導致不同的行為。

與跨叢集發佈/訂閱行為直接相關的主要屬性如下：

CLROUTE

此參數控制在連接發佈者的佇列管理程式與存在相符訂閱的佇列管理程式之間遞送訊息。

- 您可以將路徑配置成直接在這些佇列管理程式之間，或透過管理叢集主題定義的佇列管理程式。如需詳細資料，請參閱 [發佈/訂閱叢集](#)。
- 當設定 **CLUSTER** 參數時，您無法變更 **CLROUTE**。若要變更 **CLROUTE**，請先將 **CLUSTER** 內容設為空白。這會阻止使用主題的應用程式以叢集化方式進行行為。這會導致發佈中斷遞送至訂閱，因此您也應該在進行變更時靜止發佈/訂閱傳訊。

PROXYSUB

此參數控制何時進行 Proxy 訂閱。

- **FIRSTUSE** 是預設值，會導致傳送 Proxy 訂閱，以回應分散式發佈/訂閱拓撲中佇列管理程式上的本端訂閱，並在不再需要時予以取消。如需為何您可能想要將此屬性變更為預設值 **FIRSTUSE** 的詳細資料，請參閱 [個別 Proxy 訂閱轉遞及 隨處發佈](#)。
- 若要啟用 隨處發佈，請將高階主題物件的 **PROXYSUB** 參數設為 **FORCE**。這會產生單一萬用字元 Proxy 訂閱，以符合主題樹狀結構中此主題物件下的所有主題。

註: 在大型或忙碌的發佈/訂閱叢集中設定 **PROXYSUB (FORCE)** 屬性，可能會導致系統資源負載過多。**PROXYSUB (FORCE)** 屬性會延伸到每個佇列管理程式，而不只是定義主題的佇列管理程式。這會導致叢集中的每個佇列管理程式建立萬用字元 Proxy 訂閱。

本主題的訊息副本 (發佈於叢集中的任何佇列管理程式) 會直接或透過主題主機佇列管理程式 (視 **CLROUTE** 設定而定) 傳送至叢集中的每個佇列管理程式。

當主題直接遞送時，每一個佇列管理程式都會建立通往每一個其他佇列管理程式的叢集傳送端通道。當主題主機遞送時，會從叢集中的每個佇列管理程式建立通往每個主題主機佇列管理程式的通道。

如需在叢集中使用 **PROXYSUB** 參數的相關資訊，請參閱 [直接遞送的發佈/訂閱效能](#)。

PUBSCOPE 和 SUBSCOPE

這些參數決定此佇列管理程式是否將發佈資訊傳送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** 和 **MQSO_SCOPE_QMGR**，以程式設計方式來執行對等工作。

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，則會與叢集共用該定義，但基於該主題的訂閱範圍僅為本端，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。

相關資訊

[發佈範圍](#)

[訂閱範圍](#)

多個同名叢集主題定義

您可以在叢集中的多個佇列管理程式上定義相同的具名叢集主題物件，在某些情況下，這會啟用特定行為。當存在多個同名叢集主題定義時，大部分內容應該相符。如果沒有，則會根據不符的顯著性來報告錯誤或警告。

一般而言，如果多個叢集主題定義的內容不符，則會發出警告，且叢集中每一個佇列管理程式都會使用其中一個主題物件定義。每一個佇列管理程式所使用的定義並不是決定性的，或在叢集中的佇列管理程式之間一致。這類不相符應盡快解決。

在叢集設定或維護期間，有時您需要建立多個不相同的叢集主題定義。不過，這只是暫時測量時才有用，因此會被視為潛在錯誤狀況。

偵測到不符時，下列警告訊息會寫入每一個佇列管理程式的錯誤日誌：

-  在分散式系統和 IBM i 上，是 [AMQ9465](#) 和 [AMQ9466](#)。
-  在 z/OS 系統上，是 [CSQX465I](#) 和 [CSQX466I](#)。

您可以檢視主題狀態，而非主題物件定義 (例如，使用 **DISPLAY TPSTATUS**)，來判斷每一個佇列管理程式上任何主題字串的所選內容。

在某些情況下，配置內容中的衝突嚴重到足以停止正在建立的主題物件，或導致不符的物件被標示為無效且不會在叢集中延伸 (請參閱 [DISPLAY TOPIC](#) 中的 **CLSTATE**)。當叢集遞送內容 (**CLROUTE**) 中發生衝突時，會發生這些狀況 主題定義的。此外，由於主題主機遞送定義之間一致性的重要性，本文後續各節會詳述進一步的不一致之處。

如果在定義物件時偵測到衝突，則會拒絕配置變更。如果稍後由完整儲存庫佇列管理程式偵測到，則下列警告訊息會寫入佇列管理程式錯誤日誌：

- **IBM i distributed** 在分散式系統和 IBM i 上: [AMQ9879](#)
- **z/OS** 在 z/OS 系統上: [CSQX879E](#)。

在叢集中定義相同主題物件的多項定義時，本端定義的定義優先於任何遠端定義的定義。因此，如果定義中存在任何差異，則管理多個定義的佇列管理程式彼此行為不同。

定義與另一個佇列管理程式中叢集主題同名的非叢集主題的效果

可以定義未在叢集中的佇列管理程式上叢集化的受管理主題物件，並同步將相同的具名主題物件定義為不同佇列管理程式上的叢集主題定義。在此情況下，本端定義的主題物件優先於同名的所有遠端定義。

這會防止從這個佇列管理程式使用主題的叢集作業行為。也就是說，訂閱可能不會接收來自遠端發佈者的發佈，且來自發佈者的訊息可能不會延伸到叢集中的遠端訂閱。

在配置這類系統之前，應該仔細考量，因為這可能導致令人混淆的行為。

註: 如果個別佇列管理程式需要防止發佈和訂閱在叢集周圍傳播，即使主題已在其他位置叢集化，替代方法是將發佈和訂閱範圍僅設為本端佇列管理程式。請參閱 [第 78 頁的『叢集主題屬性』](#)。

直接遞送叢集中有多個叢集主題定義

對於直接遞送，您通常不會在多個叢集佇列管理程式上定義相同的叢集主題。這是因為直接遞送會讓叢集中的所有佇列管理程式都可以使用主題，而不論在哪個佇列管理程式上定義該主題。此外，新增多個叢集主題定義會大幅增加系統活動及管理複雜性，而且隨著複雜性增加，發生人為錯誤的機會也會增加：

- 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佇列管理程式。
- 叢集中特定主題的所有定義必須相同，否則很難找出佇列管理程式正在使用哪個主題定義。

另外，唯一的主機佇列管理程式也不重要，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。如需相關資訊，請參閱 [使用直接遞送的主題主機佇列管理程式可用性](#)。

對於您可能需要在第二個佇列管理程式上暫時定義叢集主題的狀況，例如，當要從叢集中移除主題的現有主機時，請參閱 [將叢集主題定義移至不同的佇列管理程式](#)。

如果您需要變更叢集主題定義，請注意要在定義所在的相同佇列管理程式上加以修改。嘗試從另一個佇列管理程式修改它可能會意外地建立主題的第二個定義，具有衝突的主題屬性。

主題主機遞送叢集中有多個叢集主題定義

使用主題主機的叢集路徑定義叢集主題時，該主題會延伸到叢集中的所有佇列管理程式，就像直接遞送主題一樣。此外，該主題的所有發佈/訂閱傳訊都會透過定義該主題的佇列管理程式來遞送。因此，叢集中主題的位置和定義數目會變得很重要（請參閱 [第 65 頁的『發佈/訂閱叢集中的主題主機遞送』](#)）。

為了確保足夠的可用性和可調整性，如果可能的話，具有多個主題定義會很有用。請參閱 [使用主題主機遞送的主題主機佇列管理程式可用性](#)。

在叢集中新增或移除主題主機遞送主題的其他定義時，您應該考量在配置變更時的訊息流程。如果在變更時在叢集中發佈訊息至主題，則需要暫置處理程序來新增或移除主題定義。請參閱 [將叢集主題定義移至不同的佇列管理程式](#) 及 [將額外主題主機新增至主題主機遞送叢集](#)。

如先前所說明，多重定義的內容應該相符，但 **PUB** 參數可能例外，如下一節所述。當透過主題主機佇列管理程式遞送發佈資訊時，更重要的是讓多個定義保持一致。因此，如果已針對主題主機叢集遞送配置一或多個主題定義，則會拒絕主題字串或叢集名稱中偵測到的不一致。

註: 如果嘗試在主題樹狀結構中的另一個主題上方或下方配置叢集主題定義，則也會拒絕叢集主題定義，其中現有的叢集主題定義已配置用於主題主機遞送。這可防止在遞送與萬用字元訂閱相關的發佈時語義不明確。

PUB 參數的特殊處理

PUB 參數用來控制應用程式何時可以發佈至主題。在叢集中的主題主機遞送的情況下，它也可以控制使用哪些主題主機佇列管理程式來遞送發佈。基於此原因，允許在叢集中具有相同主題物件的多個定義，且 **PUB** 參數的設定不同。

如果主題的多個遠端叢集定義對此參數具有不同的設定，則在符合下列條件時，主題容許將發佈傳送及遞送至訂閱：

- 在發佈者所連接且設為 **PUB (DISABLED)** 的佇列管理程式上，未定義相符的主題物件。
- 叢集中多個主題定義中的一或多個設為 **PUB (ENABLED)**，或多個主題定義中的一或多個設為 **PUB (ASPARENT)**，並在主題樹狀結構中較高點將發佈者所連接的本端佇列管理程式及訂閱定義的本端佇列管理程式設為 **PUB (ENABLED)**。

對於主題主機遞送，當訊息由連接至非主題主機之佇列管理程式的應用程式發佈時，只會將訊息遞送至主題管理佇列管理程式，其中 **PUB** 參數尚未明確設為 **DISABLED**。因此，您可以使用 **PUB (DISABLED)** 設定來靜止透過特定主題主機的訊息資料流量。您可能想要這樣做，以準備維護或移除佇列管理程式，或基於將額外主題主機新增至主題主機遞送叢集中所述的原因。

叢集主題主機佇列管理程式的可用性

設計您的發佈/訂閱叢集，以將主題主機佇列管理程式變成無法使用時，叢集將無法再處理主題資料流量的風險降至最低。主題主機佇列管理程式變成無法使用的效果取決於叢集是使用主題主機遞送還是直接遞送。

使用直接遞送之主題主機佇列管理程式的可用性

對於直接遞送，您通常不會在多個叢集佇列管理程式上定義相同的叢集主題。這是因為直接遞送會讓叢集中的所有佇列管理程式都可以使用主題，而不論在哪個佇列管理程式上定義該主題。請參閱 [直接遞送叢集中的多個叢集主題定義](#)。

在叢集中，每當叢集物件 (例如叢集佇列或叢集主題) 的主機長時間無法使用時，叢集的其他成員最終會使那些物件的知識到期。在叢集主題的情況下，如果叢集主題主機佇列管理程式變成無法使用，則其他佇列管理程式會繼續以直接叢集方式 (亦即，將發佈傳送至遠端佇列管理程式上的訂閱) 處理主題的發佈/訂閱要求至少 60 天，從主題管理佇列管理程式前次與完整儲存庫佇列管理程式進行通訊開始。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，則最後會刪除其他佇列管理程式上的快取主題物件，且主題會回復為本端主題，在此情況下，訂閱會停止從連接至遠端佇列管理程式的應用程式接收發佈。

在 60 天期間內回復您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來保證叢集主題主機仍然可用 (不過請注意，在無法使用叢集主題主機上定義的任何訂閱都不會保留可用)。60 天期限足以應付技術問題，而且可能因行政錯誤而超出。為了降低這種可能性，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，指出未重新整理其快取叢集主題物件。請確定定義叢集主題物件的佇列管理程式正在執行中，以回應這些訊息。如果無法再次讓叢集主題主機佇列管理程式可用，請在叢集中的另一個佇列管理程式上，使用完全相同的屬性來定義相同的叢集主題定義。

使用主題主機遞送的主題主機佇列管理程式可用性

對於主題主機遞送，主題的所有發佈/訂閱傳訊都會透過定義該主題的佇列管理程式來遞送。基於此原因，考量叢集中這些佇列管理程式的持續可用性非常重要。如果主題主機變成無法使用，且該主題沒有其他主機存在，則從發佈者到叢集中不同佇列管理程式上的訂閱者的資料流量會立即中止該主題。如果有其他主題主機可用，則叢集佇列管理程式會透過這些主題主機遞送新的發佈資料流量，以提供訊息遞送的持續可用性。

對於直接主題，在 60 天之後，如果第一個主題主機仍然無法使用，則會從叢集中移除該主題主機主題的知識。如果這是叢集中此主題的最後一個剩餘定義，則所有其他佇列管理程式會停止將發佈轉遞至任何主題主機以進行遞送。

為了確保足夠的可用性和可調整性，因此如果可能的話，在至少兩個叢集佇列管理程式上定義每一個主題會很有用。這可防止任何給定主題主機佇列管理程式變成無法使用。另請參閱 [主題主機遞送叢集中的多個叢集主題定義](#)。

如果您無法配置多個主題主機 (例如，因為您需要保留訊息排序)，且無法只配置一個主題主機 (因為單一佇列管理程式的可用性不得影響跨叢集中所有佇列管理程式的訂閱發佈流程)，請考量將主題配置為直接遞送主題。這可避免整個叢集依賴單一佇列管理程式，但仍需要每一個個別佇列管理程式可供使用，才能處理本端管理的訂閱及發佈者。

禁止叢集發佈/訂閱

將第一個直接遞送叢集主題引進叢集中，會強制叢集中的每一個佇列管理程式瞭解其他每一個佇列管理程式，並可能導致它們建立彼此的通道。如果不想要這樣做，您應該改為配置主題主機遞送的發佈/訂閱。如果存在直接遞送叢集主題可能會危及叢集的穩定性，由於每一個佇列管理程式的調整大小考量，您可以將叢集中每一個佇列管理程式的 **PSCLUS** 設為 **DISABLED**，完全停用叢集發佈/訂閱功能。

如第 60 頁的『發佈/訂閱叢集中的直接遞送』中所述，當您將直接遞送叢集主題引進叢集時，所有局部儲存庫都會自動收到叢集所有其他成員的通知。叢集主題也可能會在所有其他節點上建立訂閱(例如，其中指定了 **PROXYSUB (FORCE)**)，並導致從佇列管理程式啟動大量通道，即使沒有本端訂閱也一樣。這會對叢集中的每一個佇列管理程式造成立即額外負載。對於包含許多佇列管理程式的叢集，這可能會導致效能大幅降低。因此，必須小心規劃直接遞送發佈/訂閱叢集的引進。

當您知道叢集無法容納直接遞送發佈/訂閱的額外需要時，您可以改用主題主機遞送發佈/訂閱。如需差異的概觀，請參閱第 59 頁的『設計發佈/訂閱叢集』。

如果您偏好完全停用叢集的發佈/訂閱功能，則可以透過將叢集中每個佇列管理程式上的佇列管理程式屬性 **PSCLUS** 設為 **DISABLED** 來執行此動作。此設定會修改佇列管理程式功能的三個層面，以停用叢集中直接遞送及主題主機遞送的發佈/訂閱：

- 此佇列管理程式的管理者無法再將 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 訂閱會遭到拒絕，且會記載一則警告訊息，通知管理者配置不正確。
- 完整儲存庫在收到主題定義時，不再自動與所有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。如果您以這種方式選擇性地停用，則會看到頻繁的錯誤訊息。這是因為如果主題叢集在已啟用 **PSCLUS** 的佇列管理程式上，則會持續看到並拒絕 Proxy 訂閱及主題定義。

因此，您應該針對叢集中的每個佇列管理程式，將 **PSCLUS** 設為 **DISABLED**。不過，實際上，這個狀態可能難以達到及維護，例如，佇列管理程式可以隨時加入並離開叢集。您至少必須確保在所有完整儲存庫佇列管理程式上，**PSCLUS** 都設為 **DISABLED**。如果您這麼做，且隨後在叢集中的 **ENABLED** 佇列管理程式上定義叢集主題，則不會導致完整儲存庫通知每個佇列管理程式所有其他佇列管理程式，因此您的叢集會受到保護，不會在所有佇列管理程式之間產生潛在的調整大小問題。在此實務範例中，叢集主題的原點會報告在完整儲存庫佇列管理程式的錯誤日誌中。

如果佇列管理程式參與一或多個發佈/訂閱叢集，以及一或多個點對點叢集，您必須在該佇列管理程式上將 **PSCLUS** 設為 **ENABLED**。因此，當點對點叢集與發佈訂閱叢集重疊時，您應該在每一個叢集中使用一組個別的完整儲存庫。此方法容許主題定義及「所有佇列管理程式」資訊僅在發佈/訂閱叢集中流動。

當您將 **PSCLUS** 從 **ENABLED** 變更為 **已停用**時，為了避免不一致的配置，任何叢集主題物件都不能存在於這個佇列管理程式所屬的任何叢集中。在將 **PSCLUS** 變更為 **DISABLED** 之前，必須先刪除任何這類主題(即使是遠端定義的主題也一樣)。

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

相關資訊

[直接遞送發佈/訂閱叢集效能](#)

發佈/訂閱及多個叢集

單一佇列管理程式可以是多個叢集的成員。這種排列有時稱為 重疊叢集。透過這類重疊，叢集佇列可以從多個叢集進行存取，點對點訊息資料流量可以從一個叢集中的佇列管理程式遞送至另一個叢集中的佇列管理程式。發佈/訂閱叢集中的叢集主題不提供相同的功能。因此，在使用多個叢集時，必須清楚瞭解其行為。

與佇列不同，您無法將主題定義與多個叢集相關聯。叢集主題的範圍僅限於定義主題之相同叢集中的那些佇列管理程式。這只容許將發佈資訊延伸到相同叢集中的那些佇列管理程式上的訂閱。

佇列管理程式的主題樹狀結構

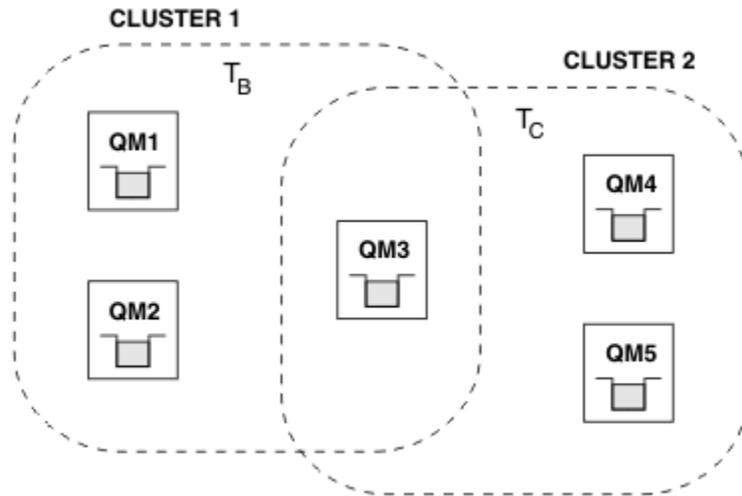


圖 28: 重疊叢集: 兩個叢集各訂閱不同的主題

當佇列管理程式是多個叢集的成員時，它會知道每一個叢集中所定義的所有叢集主題。例如，在上圖 QM3 中，同時知道 T_B 及 T_C 受管理叢集主題物件，而 QM1 只知道 T_B 。QM3 會將這兩個主題定義套用至其本端主題，因此對於特定主題具有與 QM1 不同的行為。基於此原因，來自不同叢集的叢集主題不會彼此干擾是很重要的。當一個叢集主題定義在不同叢集中另一個叢集主題的上方或下方時 (例如，它們具有 /Sport 及 /Sport/Football 的主題字串)，甚至對於這兩者中的相同主題字串，也會發生干擾。另一種干擾形式是當受管理叢集主題物件在不同叢集中以相同的物件名稱定義，但針對不同的主題字串。

如果進行這樣的配置，則將發佈遞送至相符訂閱會變得非常依賴發佈者及訂閱者相對於叢集的相對位置。因此，您無法依賴這類配置，您應該將它變更為移除干擾主題。

規劃具有發佈/訂閱傳訊的重疊叢集拓撲時，您可以將主題樹狀結構和叢集主題物件名稱視為跨越拓撲中所有重疊叢集，以避免任何干擾。

整合多個發佈/訂閱叢集

如果需要發佈/訂閱傳訊來跨越不同叢集中的佇列管理程式，則有兩個選項可用：

- 透過使用發佈/訂閱階層配置，將叢集連接在一起。請參閱 [結合多個叢集的主題空間](#)。
- 建立額外的叢集來覆蓋現有叢集，並包括需要發佈或訂閱特定主題的所有佇列管理程式。

使用後者選項時，您應該仔細考量叢集的大小，以及最有效的叢集遞送機制。請參閱第 59 頁的『[設計發佈/訂閱叢集](#)』。

發佈/訂閱叢集中保留發佈的設計考量

設計發佈/訂閱叢集以使用保留的發佈時，需要考量一些限制。

考量

考量 1: 下列叢集佇列管理程式一律儲存最新版本的保留發佈資訊：

- 發佈者的佇列管理程式
- 在主題主機遞送叢集中，主題主機 (假設主題只有一個主題主機，如本文章下一節所述)
- 訂閱符合保留發佈資訊主題字串的所有佇列管理程式

考量 2: 佇列管理程式沒有訂閱時，不會收到更新的保留發佈。因此，儲存在佇列管理程式中且不再訂閱主題的任何保留發佈資訊都會變成過時。

考量 3: 在建立任何訂閱時，如果主題字串有保留發佈資訊的本端副本，則會將本端副本遞送至訂閱。如果您是哪個給定主題字串的第一個訂閱者，則也會從下列其中一個叢集成員遞送符合的保留發佈資訊：

- 在直接遞送叢集中，發佈者的佇列管理程式
- 在主題主機遞送叢集中，給定主題的主題主機

從主題主機或發佈佇列管理程式，將保留的發佈資訊遞送至訂閱佇列管理程式，對於 MQSUB 呼叫是非同步的。因此，如果您使用 [MQSUBRQ](#) 呼叫，則在後續呼叫 MQSUBRQ 之前，可能會遺失最新保留的發佈資訊。

含意

對於任何發佈/訂閱叢集，當進行第一次訂閱時，本端佇列管理程式可能會儲存保留發佈的舊副本，而這是遞送至新訂閱的副本。本端佇列管理程式上存在訂閱表示這將在下次更新保留的發佈時解決。

對於主題主機遞送發佈/訂閱叢集，如果您為給定主題配置多個主題主機，則新訂閱者可能會從主題主機接收最新保留發佈，或者他們可能會從另一個主題主機接收過時保留發佈(最新已遺失)。對於主題主機遞送，通常會為給定主題配置多個主題主機。不過，如果您預期應用程式會使用保留的發佈資訊，則每一個主題應該只配置一個主題主機。

對於任何給定的主題字串，您應該只使用單一發佈者，並確定發佈者一律使用相同的佇列管理程式。如果您不這麼做，相同主題的不同佇列管理程式可能有不同的保留發佈資訊在作用中，導致非預期的行為。因為已配送多個 Proxy 訂閱，所以可能會收到多個保留的發佈。

如果您仍然擔心訂閱者使用過時發佈，請在建立每一個保留的發佈時考慮設定訊息到期。

您可以使用 **CLEAR TOPICSTR** 指令，從發佈/訂閱叢集中移除保留的發佈。在某些情況下，您可能需要在發佈/訂閱叢集的多個成員上發出指令，如 [CLEAR TOPICSTR](#) 中所述。

萬用字元訂閱及保留的發佈

如果您使用萬用字元訂閱，則發佈/訂閱叢集的其他成員所收到的對應 Proxy 訂閱會從第一個萬用字元之前的主題分隔字元萬用字元。請參閱 [萬用字元和叢集主題](#)。

因此，所使用的萬用字元可能符合更多主題字串，以及更多保留的發佈資訊，而不符合訂閱應用程式。

這會增加保留發佈所需的儲存體數量，因此您需要確保管理佇列管理程式具有足夠的儲存體容量。

相關資訊

[保留的發佈](#)

[個別 Proxy 訂閱轉遞及隨處發佈](#)

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** 和 第 54 頁的『叢集作業：使用 REFRESH CLUSTER 最佳作法』的使用注意事項。

在發佈/訂閱叢集中的佇列管理程式上發出 **REFRESH CLUSTER** 指令之後，請等到所有叢集佇列管理程式及叢集主題都已順利重新整理，然後依照重新同步化 Proxy 訂閱中的說明來重新同步化 Proxy 訂閱。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

第 54 頁的『叢集作業：使用 REFRESH CLUSTER 最佳作法』

您可以使用 **REFRESH CLUSTER** 指令來捨棄所有本端保留的叢集相關資訊，並從叢集中的完整儲存庫重建該資訊。除非發生異常狀況，否則您應該不需要使用此指令。如果您確實需要使用它，對於如何使用它，有一些特殊考量。此資訊是根據客戶的測試和意見而提供的指引。

相關資訊

執行 REFRESH CLUSTER 時看到的應用程式問題

MQSC 指令參照: REFRESH CLUSTER

發佈/訂閱階層中的遞送

如果您的分散式佇列管理程式拓撲是發佈/訂閱階層，且在佇列管理程式上進行訂閱，依預設會在階層中的每個佇列管理程式上建立 Proxy 訂閱。然後，在任何佇列管理程式上接收到的發佈會透過階層遞送至管理相符訂閱的每一個佇列管理程式。

如需如何在發佈/訂閱階層及叢集中的佇列管理程式之間遞送訊息的簡介，請參閱 [分散式發佈/訂閱網路](#)。

在分散式發佈/訂閱階層中的佇列管理程式上訂閱主題時，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。Proxy 訂閱提供佇列管理程式將發佈資訊轉遞至管理該主題之訂閱的那些佇列管理程式所需的資訊。發佈/訂閱階層中的每一個佇列管理程式只知道其直接關係。放置至一個佇列管理程式的發佈會透過其直接關係，傳送至具有訂閱的那些佇列管理程式。

下圖說明此情況，其中 *Subscriber 1* 會在亞洲佇列管理程式 (1) 上登錄特定主題的訂閱。亞洲佇列管理程式上此訂閱的 Proxy 訂閱會轉遞至網路中的所有其他佇列管理程式 (2,3, 4)。

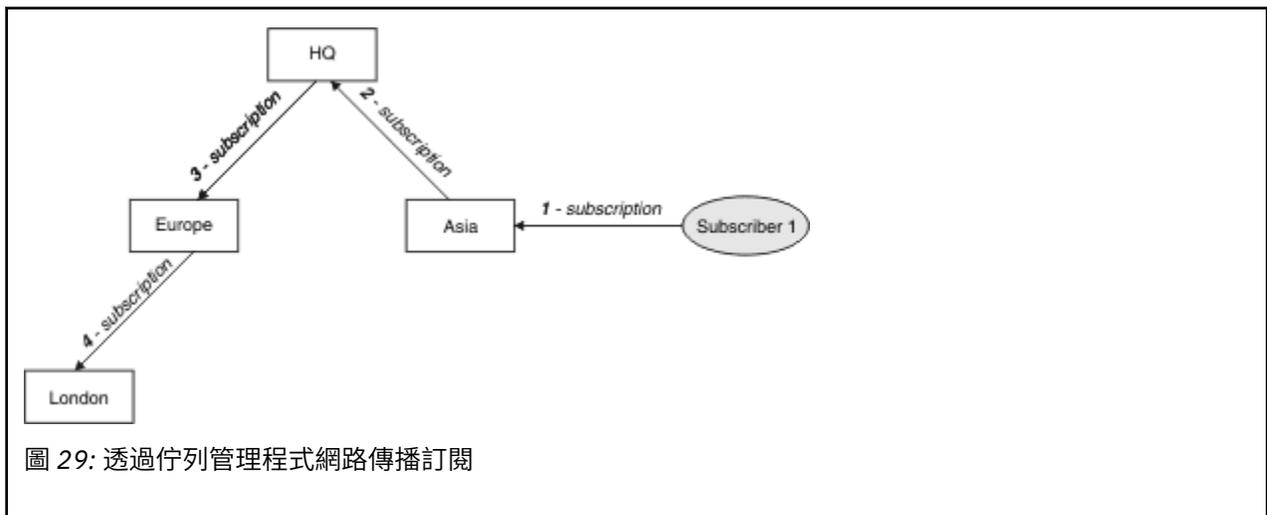


圖 29: 透過佇列管理程式網路傳播訂閱

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 Proxy 訂閱已存在，否則它會為訂閱及其鄰接項的主題建立 Proxy 訂閱。下圖說明 *Subscriber 2* 在 HQ 佇列管理程式 (5) 上向第 85 頁的圖 29 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，以便它知道訂閱存在於網路上的其他位置 (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 85 頁的圖 29 中的步驟 3。

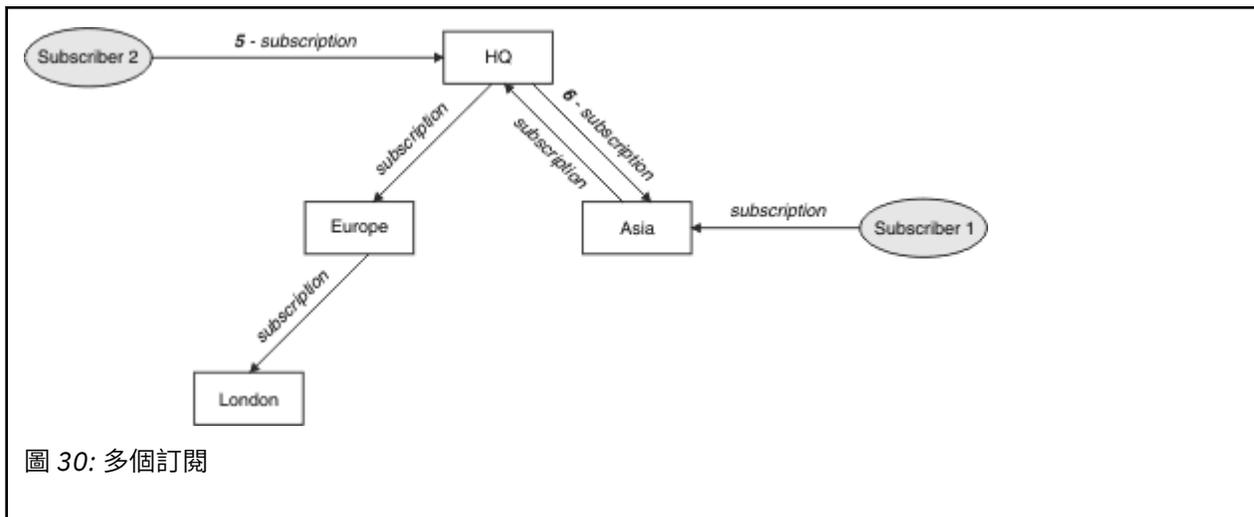


圖 30: 多個訂閱

當應用程式將資訊發佈至主題時，依預設，接收端佇列管理程式會將它轉遞至具有該主題有效訂閱的所有佇列管理程式。它可能透過一或多個中間佇列管理程式來轉遞它。下圖說明這一點，其中發佈者將與第 86 頁的圖 30 中相同主題的發佈傳送至歐洲佇列管理程式 (7)。此主題的訂閱是從 HQ 到歐洲，因此發佈會轉遞至 HQ 佇列管理程式 (8)。不過，沒有從倫敦至歐洲的訂閱 (僅從歐洲至倫敦)，因此發佈不會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 Subscriber 2 及亞洲佇列管理程式 (9)。發佈會從亞洲 (10) 轉遞至 Subscriber 1。

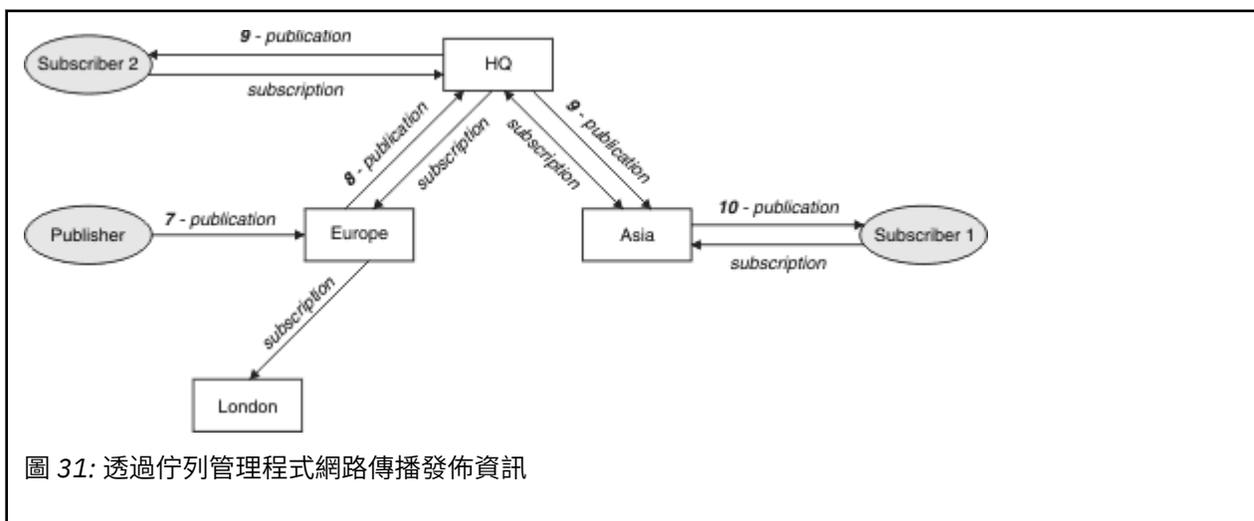


圖 31: 透過佇列管理程式網路傳播發佈資訊

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 [將預設使用者 ID 與佇列管理程式階層搭配使用](#)。

註: 如果您改用發佈/訂閱叢集，則叢集會處理授權。

摘要及其他考量

發佈/訂閱階層可讓您精確控制佇列管理程式之間的關係。在建立之後，它幾乎不需要人為介入來管理。不過，它也會對您的系統施加某些限制:

- 階層中的較高節點 (特別是根節點) 必須在健全、高可用性及高效能設備上管理。這是因為預期會有更多發佈資料流量流經這些節點。
- 階層中每個非葉節點佇列管理程式的可用性會影響網路將訊息從發佈者傳送至其他佇列管理程式上的訂閱者的能力。
- 依預設，所有訂閱的主題字串都會延伸到整個階層，發佈只會延伸到訂閱相關聯主題的遠端佇列管理程式。因此，對訂閱集的快速變更可能成為限制因素。您可以變更此預設行為，並改為將所有發佈延伸到所有佇列管理程式，這樣就不需要 Proxy 訂閱。請參閱 [發佈/訂閱網路中的訂閱效能](#)。

註:類似的限制也適用於直接遞送叢集。

- 由於發佈/訂閱佇列管理程式的交互連接本質，Proxy 訂閱需要時間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，因此在訂閱新的主題字串之後，可能不會傳送早期發佈。您可以移除訂閱延遲所造成的問題，方法是讓所有發佈延伸到所有佇列管理程式，這樣就不需要 Proxy 訂閱。請參閱 [發佈/訂閱網路中的訂閱效能](#)。

註:此限制也適用於直接遞送叢集。

- 對於發佈/訂閱階層，新增或移除佇列管理程式需要對階層進行手動配置，並仔細考量那些佇列管理程式的位置及其對其他佇列管理程式的依賴。除非您要新增或移除位於階層底端的佇列管理程式，因此在它們下方沒有其他分支，否則您也必須在階層中配置其他佇列管理程式。

在使用發佈/訂閱階層作為遞送機制之前，請先探索 [第 60 頁的『發佈/訂閱叢集中的直接遞送』](#) 和 [第 65 頁的『發佈/訂閱叢集中的主題主機遞送』](#) 中詳述的替代方法。

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。您需要注意它們的存在，僅用於問題判斷及產能規劃目的。

如需如何監視這些佇列的指引，請參閱 [在發佈/訂閱網路中平衡生產者和消費者](#)。

| 系統佇列 | 用途 |
|---------------------------|--------------------------------------|
| SYSTEM.INTER.QMGR.CONTROL | IBM MQ 分散式發佈/訂閱控制佇列 |
| SYSTEM.INTER.QMGR.FANREQ | IBM 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 | IBM MQ 分散式發佈/訂閱出版品 |
| SYSTEM.HIERARCHY.STATE | IBM MQ 分散式發佈/訂閱階層關係狀態 |

 在 z/OS 上，您可以在建立佇列管理程式時設定必要的系統物件，方法是將 CSQ4INSR 及 CSQ4INSG 範例併入 CSQINP2 起始設定輸入資料集。如需相關資訊，請參閱 [作業 13: 自訂起始設定輸入資料集](#)。

發佈/訂閱系統佇列的屬性顯示在 [第 87 頁的表 7](#) 中。

| 屬性 | 預設值 |
|--|--|
| DEFPSIST | 是 |
| DEFSOPT | EXC |
| MAXMSGL |  在 AIX、HP-UX、Linux、IBM i、Solaris 及 Windows 平台上:ALTER QMGR 指令的 MAXMSGL 參數值  在 z/OS: 上 (即 100 MB) |
| MAXDEPTH | 999999999 |
| SHARE | 無 |
|   STGCLASS | 此屬性僅在 z/OS 平台上使用 |

註: 唯一包含應用程式所放置訊息的佇列是 SYSTEM.INTER.QMGR.PUBS。MAXDEPTH 設為此佇列的上限值, 以容許在中斷或負載過多期間暫時建置已發佈的訊息。如果佇列管理程式在無法包含該佇列深度的系統上執行, 則應該調整此值。

相關資訊

[分散式發佈/訂閱疑難排解](#)

分散式發佈/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。這會影響訂閱知識在發佈/訂閱網路中的傳播, 以及發佈至遠端佇列管理程式上的訂閱。

如果無法使用扇出要求佇列 SYSTEM.INTER.QMGR.FANREQ, 則建立訂閱可能會產生錯誤, 當 Proxy 訂閱需要遞送至直接連接的佇列管理程式時, 錯誤訊息會寫入佇列管理程式錯誤日誌。

如果階層關係狀態佇列 SYSTEM.HIERARCHY.STATE 無法使用, 則會將錯誤訊息寫入佇列管理程式錯誤日誌, 且發佈/訂閱引擎會進入 COMPAT 模式。若要檢視發佈/訂閱模式, 請使用指令 DISPLAY QMGR PSMODE。

如果任何其他 SYSTEM.INTER.QMGR 佇列無法使用, 則會將錯誤訊息寫入佇列管理程式錯誤日誌, 而且雖然未停用此功能, 但發佈/訂閱訊息可能會建置在此佇列管理程式或遠端佇列管理程式上的佇列上。

如果發佈/訂閱系統佇列或母項、子項或發佈/訂閱叢集佇列管理程式的必要傳輸佇列無法使用, 則會發生下列結果:

- 發佈不會遞送, 發佈應用程式可能會收到錯誤。如需發佈應用程式收到錯誤的詳細資料, 請參閱 **DEFINE TOPIC** 指令的下列參數: **PMSGDLV**、**NPMSGDLV** 及 **USEDLQ**。
- 接收到的佇列管理程式間發佈資訊會取消至輸入佇列, 然後重試。如果達到取消臨限值, 則會將未遞送的發佈放置在無法傳送的郵件佇列上。佇列管理程式錯誤日誌將包含問題的詳細資料。
- 未遞送的 Proxy 訂閱會取消至展開要求佇列, 然後再次嘗試。如果達到取消臨限值, 則未遞送的 Proxy 訂閱不會遞送至任何連接的佇列管理程式, 且會放置在無法傳送的郵件佇列上。佇列管理程式錯誤日誌將包含問題的詳細資料, 包括任何必要的更正管理動作的詳細資料。
- 階層關係通訊協定訊息失敗, 且連線狀態標示為 ERROR。若要檢視連線狀態, 請使用指令 **DISPLAY PUBSUB**。

相關資訊

[分散式發佈/訂閱疑難排解](#)

規劃儲存體及效能需求

您必須為 IBM 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 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 MQ 的系統需求](#) 網站:

www.ibm.com/software/integration/wmq/requirements/

IBM 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, 以瞭解規劃及配置目錄結構以與 IBM MQ 搭配使用的相關資訊:

- [第 90 頁的『規劃檔案系統支援』](#)
- [第 91 頁的『共用檔案系統的需求』](#)
- [第 100 頁的『共用 IBM MQ 檔案』](#)
- [第 102 頁的『UNIX 和 Linux 系統上的目錄結構』](#)
- [第 111 頁的『Windows 系統上的目錄結構』](#)
-  [第 114 頁的『IBM i 上的目錄結構』](#)

如需 UNIX 和 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 117 頁的『IBM MQ 及 UNIX System V IPC 資源』](#)
- [第 117 頁的『AIX 上的共用記憶體』](#)
- [第 117 頁的『IBM MQ 和 UNIX 程序優先順序』](#)

如需日誌檔的相關資訊，請使用下列鏈結：

- [計算日誌大小](#)

相關概念

[第 5 頁的『規劃』](#)

規劃 IBM MQ 環境時，請考量 IBM MQ 為單一及多個佇列管理程式架構提供的支援，以及點對點及發佈/訂閱傳訊樣式的支援。同時規劃您的資源需求，以及使用記載和備份機能。

[第 119 頁的『在 z/OS 上規劃 IBM MQ 環境』](#)

規劃 IBM MQ 環境時，您必須考量資料集、頁面集、Db2、「連結機能」的資源需求，以及記載及備份機能的需求。請利用這個主題來規劃 IBM MQ 執行所在的環境。

相關資訊

[UNIX 和 Linux 上的軟硬體需求](#)

[Windows 上的軟硬體需求](#)

distributed 分散式平台上的磁碟空間需求

IBM 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。)

IBM i

附註：

1. 在 IBM i 上，您無法將原生用戶端與伺服器分開。表格中的伺服器圖例適用於不含 Java 的 5724H72*BASE，以及「英文語言載入」(2924)。有 22 個可能的唯一語言載入。
2. 表格中的圖例適用於不含 Java 的原生用戶端 5725A49 *BASE。
3. Java 和 JMS 類別可以同時新增至伺服器和用戶端連結。如果您想要包含這些特性，請新增 110 MB。
4. 將範例來源新增至用戶端或伺服器會額外新增 10 MB。
5. 新增範例至 Java 和 JMS 類別會額外新增 5 MB。

| 平台 | 用戶端安裝 ¹ | 伺服器安裝 ² | IBM MQ MFT 安裝 ³ | 完整安裝 ⁴ |
|---|--------------------|--------------------|----------------------------|--------------------------------------|
| 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 | 無 | 無 | 120 MB |
| Linux for System x (64 位元) | 125 MB | 170 MB | 575 MB | 935 MB |
| V 8.0.0.2 V 8.0.0.2 Linux on POWER Systems - Little Endian | 90 MB | 125 MB | 555 MB | 620 MB V 8.0.0.8 645 MB |

| 平台 | 用戶端安裝 ¹ | 伺服器安裝 ² | IBM MQ MFT 安裝 ³ | 完整安裝 ⁴ |
|-------------------------------------|---------------------|---------------------|----------------------------|-------------------|
| Linux on POWER Systems - Big Endian | 130 MB | 170 MB | 565 MB | 715 MB |
| Linux 適用於 IBM Z | 125 MB | 160 MB | 560 MB | 665 MB |
| Solaris x86-64、AMD64、EM64T 及相容處理器 | 105 MB ⁵ | 150 MB ⁵ | 695 MB | 860 MB |
| Solaris SPARC | 105 MB ⁵ | 150 MB ⁵ | 680 MB | 820 MB |
| Windows (32 位元安裝) ⁶ | 390 MB | 無 | 無 | 475 MB |
| Windows (64 位元安裝) ⁶ | 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 MQ Managed File Transfer 安裝包括下列元件:

- IBM 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 IBM MQ 特性](#)。

相關資訊

[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。

共用檔案系統及 IBM 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 UNIX、Linux 和 Windows 系統上，佇列管理程式的資料檔及日誌檔必須置於共用網路檔案系統中。在 IBM i 上，使用日誌登載而非日誌檔，且無法共用日誌登載。IBM i 上的多重實例佇列管理程式會使用日誌登載抄寫或可切換的日誌登載，讓日誌登載可在不同的佇列管理程式實例之間使用。

在 v7.0.1 之前，IBM 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，IBM 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，IBM 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 `amqmfsc` 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、Linux 及 IBM i 系統。在 Windows 上，只有一個受支援的網路檔案系統，且不需要 `amqmfsc` 指令。

相關工作

第 92 頁的『[驗證共用檔案系統行為](#)』

執行 `amqmfsc`，以檢查 UNIX、IBM i 及 IBM i 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 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。

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，並在無法可靠地使用 IBM 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。

更具體地說，UNIX 平台上的 IBM 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，且寫入作業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. 保證對檔案的專用存取權。

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 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，網路儲存體必須由 Microsoft Windows 網路所使用的「共用網際網路檔案系統 (CIFS)」通訊協定存取。「共用網際網路檔案系統 (CIFS)」用戶端不符合 Microsoft Windows 以外平台上鎖定語意的 IBM MQ 需求，因此在 Microsoft Windows 以外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。網路檔案系統第 4 版滿足此需求。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。

如需進一步資訊，請參閱 [IBM MQ 多重實例佇列管理程式的測試及支援陳述式](#)。

在一般情況下，IBM 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 92 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 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 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

附註：

- 對於用來保留 IBM MQ 安裝目錄的裝載點，您無法使用 **nosuid** 或 **noexec** 選項。這是因為 IBM MQ 包括 **setuid/setgid** 可執行程式，且不能阻止這些程式適當地執行。
- 當您只將佇列管理程式資料放置在「網路檔案系統 (NFS)」伺服器上時，您可以在 **mount** 指令中使用下列三個選項，使系統安全，且不會對佇列管理程式的執行產生有害影響：

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  及 IBM i 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 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，以及兩部已安裝 IBM MQ 的其他連接伺服器。您必須具有管理者 (root) 權限才能配置檔案系統，並且必須是 IBM MQ 管理者才能執行 **amqmfsc**。

關於這項作業

第 91 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 MQ Technote [IBM MQ 多重實例佇列管理程式的測試及支援陳述式](#) 列出 IBM 已測試的共用檔案系統。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 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 IBM MQ 管理者，也就是 UNIX 上 mqm 群組的成員。執行測試的使用者必須具備 IBM MQ 管理者權限。

使用在 Linux 上建立多重實例佇列管理程式 [IBM i](#) 中匯出及裝載檔案系統的範例，或使用 ADDMQMJRN 在 ASP 上建立 [鏡映日誌登載配置](#) 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 amqmfscck 同時執行 amqsfhac，請與此作業平行執行作業第 98 頁的『執行 amqsfhac 以測試訊息完整性』。

1. 在兩部 IBM 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 MQ 伺服器上，執行沒有參數的檔案系統檢查程式。

```
amqmfscck /shared/qmdata
```

圖 32: 在 IBM MQ 伺服器 1 上

3. 請檢查同時從兩個 IBM MQ 伺服器寫入相同目錄的情況。

在兩部 IBM MQ 伺服器上，使用 -c 選項同時執行檔案系統檢查程式。

```
amqmfscck -c /shared/qmdata
```

圖 33: 在 IBM MQ 伺服器 1 上

```
amqmfscck -c /shared/qmdata
```

圖 34: 在 IBM MQ 伺服器 2 上

4. 檢查等待及釋放兩部 IBM MQ 伺服器上的鎖定。

在兩部 IBM MQ 伺服器上，使用 -w 選項同時執行檔案系統檢查程式。

```
amqmfscck -w /shared/qmdata
```

圖 35: 在 IBM MQ 伺服器 1 上

```
amqmfscck -w /shared/qmdata
```

圖 36: 在 IBM 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 37: 在 IBM 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 8: 同時在兩部伺服器上執行資料完整性檢查

| IBM MQ 伺服器 1 | IBM MQ 伺服器 2 |
|--|--------------|
| <pre>amqmfscck -a /shared/qmdata</pre> | |

表 8: 同時在兩部伺服器上執行資料完整性檢查 (繼續)

| IBM MQ 伺服器 1 | IBM MQ 伺服器 2 |
|--|--|
| <p>Please start this program on a second machine with the same parameters.</p> <p>File lock acquired.</p> <p>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.</p> | <pre>amqmfsc -a /shared/qmdata</pre> <p>Waiting for lock...</p> |
| <p>Turn the power off here.</p> | |
| | <p>File lock acquired.</p> <p>Reading test file</p> <p>Checking the integrity of the data read.</p> <p>Appending data into the test file after data already found.</p> <p>The test file is full of data. It is ready to be inspected for data integrity.</p> |

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 **amqmfsc** 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 38: 在 IBM MQ 伺服器 2 上

6. 刪除測試檔案。

```
amqmfscck -x /shared/qmdata
Test files deleted.
```

圖 39: 在 IBM 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 MQ 伺服器 1 | IBM MQ 伺服器 2 |
|---|---|
| <pre>> 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>> 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_RDLCK)
System call: fd2 = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call:fcntl(fd2, F_SETLK, F_RDLCK)
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/amqmfsc.lck', errno 2
(Permission denied).
```

成功測試兩部伺服器上的鎖定

| 表 10: 在兩部伺服器上順利鎖定-詳細模式 | |
|--|--|
| IBM MQ 伺服器 1 | IBM MQ 伺服器 2 |
| <pre>> amqmfsc -wv /shared/qmdata Calling 'stat("/shared/qmdata")' Calling 'fd = open("/shared/qmdata/ amqmfsc.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>> amqmfsc -wv /shared/qmdata Calling 'stat("/shared/qmdata")' Calling 'fd = open("/shared/qmdata/ amqmfsc.lkw", O_EXCL O_CREAT O_RDWR,0666)' Calling 'fd = open("/shared/qmdata/amqmfsc.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> |

相關資訊

amqmfsc (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 MQ MQI client 應用程式執行 **amqsfhac**。

遵循 [程序](#) 中的步驟 [第 93 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 93 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](#)。

- 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
```

- 設定用戶端連線以執行 **amqsfhac**。
 - 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。
 - 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。

- 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
```

- 使用 **amqsfhac** 測試配置

```
amqsfhac QM1 TARGETQ SIDEQ 2 2 2
```

- 測試檔案系統完整性時測試訊息完整性。

在程序的步驟 [第 94 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 100 頁](#) 的圖 40 中顯示在步驟 [第 99 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 40: 成功執行 *amqsfhac* 的輸出

相關資訊

高可用性範例程式

共用 IBM MQ 檔案

部分 IBM MQ 檔案由作用中佇列管理程式專用存取，其他檔案則共用。

IBM MQ 檔案會分割成程式檔案和資料檔。程式檔案通常安裝在執行 IBM MQ 的每一部伺服器本端。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 100 頁的圖 41 中顯示的每一個 *qmgrs* 及 *log* 目錄中) 具有互斥存取權。

第 100 頁的圖 41 是 IBM 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。

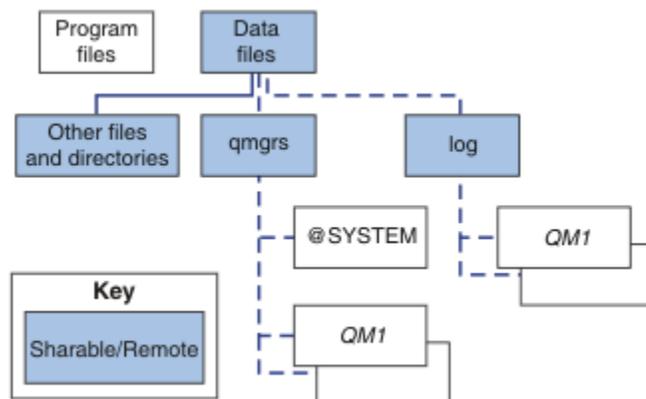


圖 41: IBM MQ 目錄結構的整體視圖

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預設位置本端，UNIX 和 Linux 系統上為 `/var/mqm`，安裝在 Windows 上可配置。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 IBM MQ 安裝之間共用它。IBM 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。IBM 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 `mqs.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。

在 UNIX 和 Linux 平台的 [第 100 頁的圖 41](#) 中，QM1 目錄的 `Prefix` 屬性是：

```
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 `mqs.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。

在 UNIX 或 Linux 平台的 [第 100 頁的圖 41](#) 中，QM1 目錄的 `DataPath` 屬性是：

```
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 101 頁的圖 42](#)。

```
DataPath/QmgrName/@IPCC/esem
```

圖 42: 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 102 頁的圖 43](#)。

```
DataPath/QmgrName/@IPCC/esem/myHostName/
```

圖 43: 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 `MQS_IPC_HOST` 設定 *myHostName* ；請參閱 第 102 頁的圖 44。

```
export MQS_IPC_HOST= myHostName
```

圖 44: 範例: 設定 `MQS_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，IBM 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及日誌檔的互斥存取權。從 v7.0.1 開始，在支援共用檔案系統的情況下，IBM 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 和 Linux 系統上的目錄結構

UNIX 和 Linux 系統上的 IBM MQ 目錄結構可以對映至不同的檔案系統，以更容易管理、效能更佳及可靠性更高。

使用 IBM 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 103 頁的圖 45 中顯示的目錄結構，其中 R 是產品的版次。它是從 IBM WebSphere MQ 7.0.1 開始在 IBM 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 及 qmgrs 目錄位於與另一個 IBM MQ 安裝共用的網路檔案系統上。

第 103 頁的圖 45 顯示 IBM 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 107 頁的『UNIX 和 Linux 系統上的目錄配置範例』。

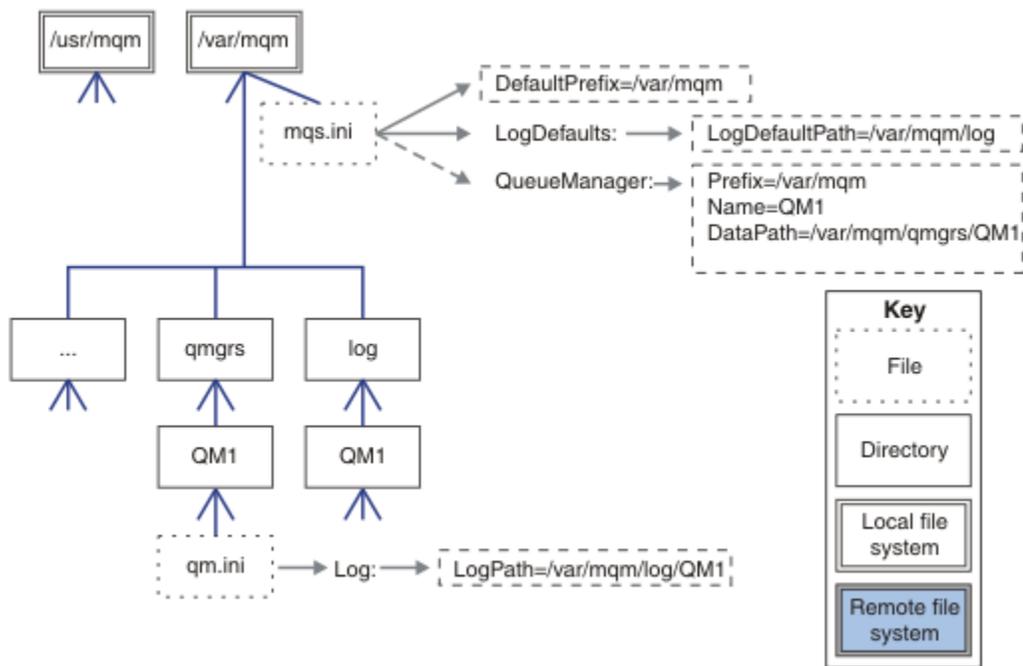


圖 45: UNIX 和 Linux 系統的預設 IBM MQ 目錄結構範例

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 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。

註: `/var/mqm/errors` 目錄應該是個別的檔案系統，以防止佇列管理程式所產生的 FFDC 填滿包含 `/var/mqm` 的檔案系統。

如需相關資訊，請參閱 [在 UNIX 和 Linux 系統上建立檔案系統](#)。

`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 和 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](#)

如需產品檔案位置的相關資訊，請參閱 [選擇安裝位置](#)

如需替換目錄配置的相關資訊，請參閱 [第 90 頁的『規劃檔案系統支援』](#)。

在 [第 105 頁的圖 46](#) 中，在佇列管理程式已使用一段時間之後，佈置會代表 IBM MQ。您具有的實際結構視佇列管理程式上發生的作業而定。

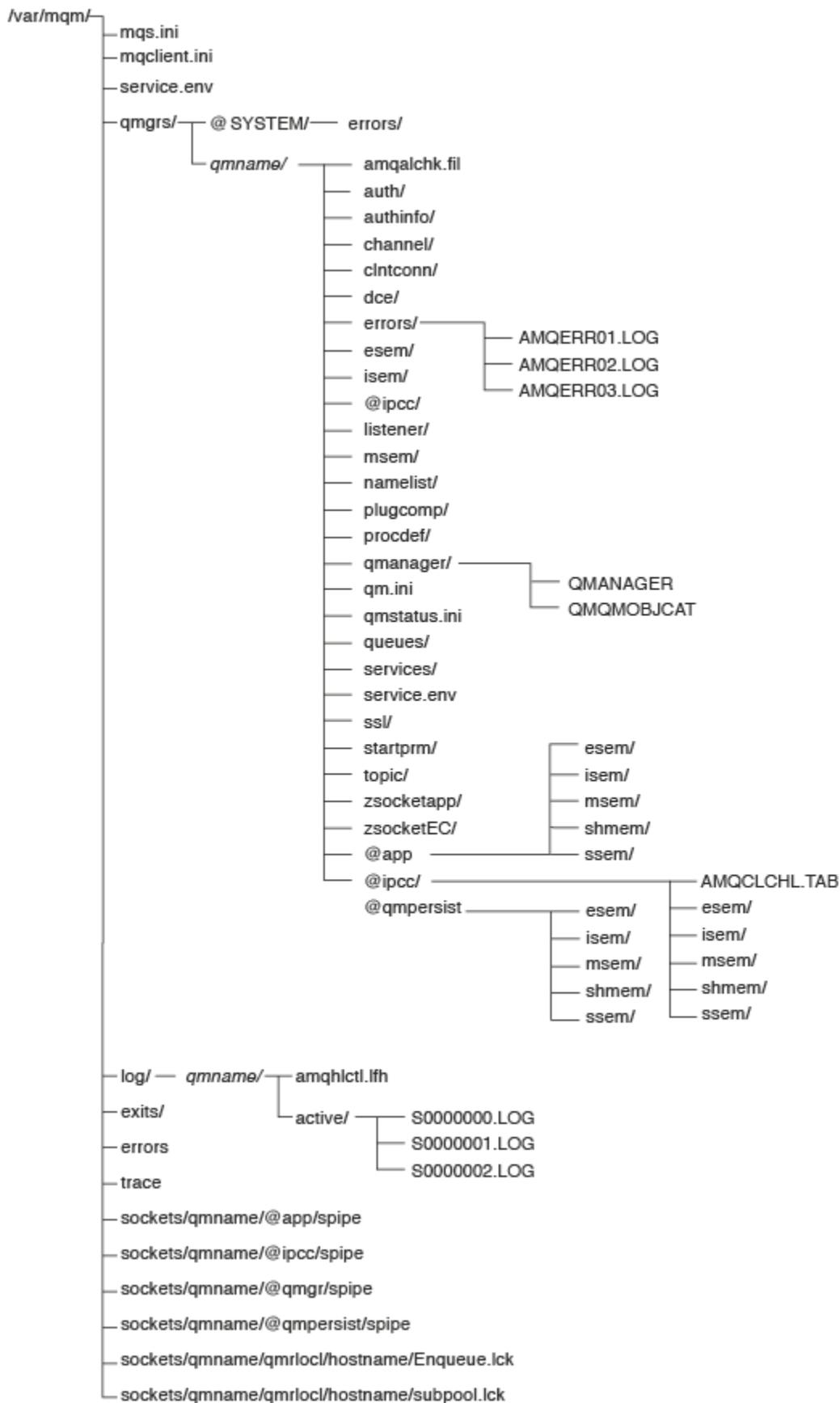


圖 46: 啟動佇列管理程式之後的預設目錄結構 (UNIX 系統)

/var/mqm/

/var/mqm 目錄包含適用於整個 IBM MQ 安裝架構的配置檔及輸出目錄，而不適用於個別佇列管理程式。

| | |
|----------------------------|--|
| <u>mqs.ini</u> | IBM MQ 安裝層面配置檔，在佇列管理程式啟動時讀取。 可使用 AMQ_MQS_INI_LOCATION 環境變數修改檔案路徑。 請確定已在執行 strmqm 指令的 Shell 中設定並匯出此項。 |
| <u>mqclient.ini</u> | IBM MQ MQI client 程式所讀取的預設用戶端配置檔。 可使用 MQCLNTCF 環境變數修改檔案路徑。 |
| <u>service.env</u> | 包含服務程序的機器範圍環境變數。 檔案路徑已修正。 |
| <u>錯誤/</u> | 機器範圍錯誤日誌及 FFST 檔案。 已修正目錄路徑。 另請參閱 FFST: IBM MQ for UNIX 及 Linux 系統 。 |
| <u>Socket/</u> | 包含每一個佇列管理程式的資訊，僅供系統使用。 |
| <u>trace/</u> | 追蹤檔。 已修正目錄路徑。 |
| <u>結束程式/</u> | 包含使用者通道結束程式的預設目錄。 |
| <u>exits64/</u> | 在 mqs.ini 檔案的 ApiExit 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。

| | |
|--|---|
| <u>qm.ini</u> | 佇列管理程式配置檔，在佇列管理程式啟動時讀取。 |
| <u>錯誤/</u> | 佇列管理程式範圍錯誤日誌。 qmname = @system 包含不明或無法使用佇列管理程式的通道相關訊息。 |
| <u>@ipcc/</u> <u>AMQCLCHL.TAB</u> | 由 IBM MQ 伺服器建立並由 IBM MQ MQI client 程式讀取的預設用戶端通道控制表格。 可使用 MQCHLLIB 和 MQCHLTAB 環境變數修改檔案路徑。 |
| <u>QMANAGER</u> | 佇列管理程式物件檔: QMANAGER 佇列管理程式物件型錄: QMQMOBJCAT |

表 12: UNIX 系統上 `/var/mqm/qmgrs/qmname` 目錄的記載內容 (繼續)

| | |
|------------------|---|
| authinfo/ | 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 瞭解 IBM MQ 檔名 。 |
| channel/ | |
| clntconn/ | |
| listener/ | |
| namelist/ | |
| procdef/ | |
| queues/ | |
| services/ | |
| topics/ | |
| ... | IBM MQ 使用的其他目錄 (例如 @ipcc) 僅由 IBM 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 13: UNIX 系統上 `/var/mqm/log/qmname` 目錄的記載內容

| | |
|---------------------|---|
| amqhlctl.lfh | 日誌控制檔。 |
| active/ | 此目錄包含編號為 <code>S0000000.LOG</code> , <code>S0000001.LOG</code> , <code>S0000002.LOG</code> 等。 |

`opt/mqm`

依預設, `opt/mqm` 是大部分平台上的安裝目錄。如需您企業在一或多個平台上安裝目錄所需空間量的相關資訊, 請參閱第 89 頁的『分散式平台上的磁碟空間需求』。

UNIX 和 Linux 系統上的目錄配置範例

UNIX 和 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 IBM 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 108 頁的圖 47 是衍生替代 IBM 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: IBM MQ 會將佇列管理程式名稱新增至路徑中。

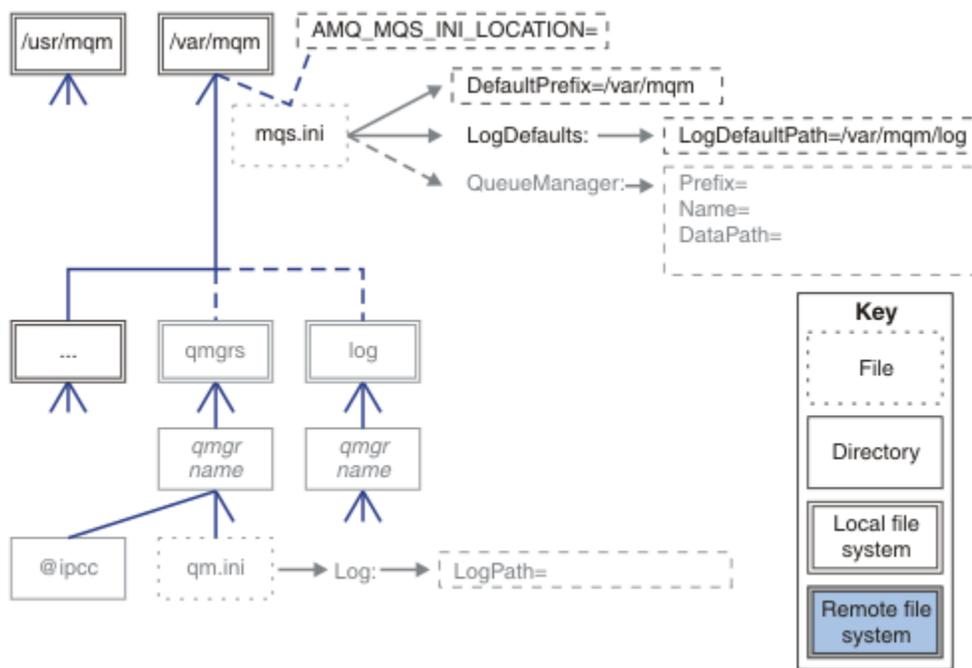


圖 47: 目錄結構型樣範本

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` 指令所建立 IBM 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 v7.0.1 之前的 IBM 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 109 頁的圖 48 是在 v7.0.1 中透過發出指令 `crtmqm QM1` 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。

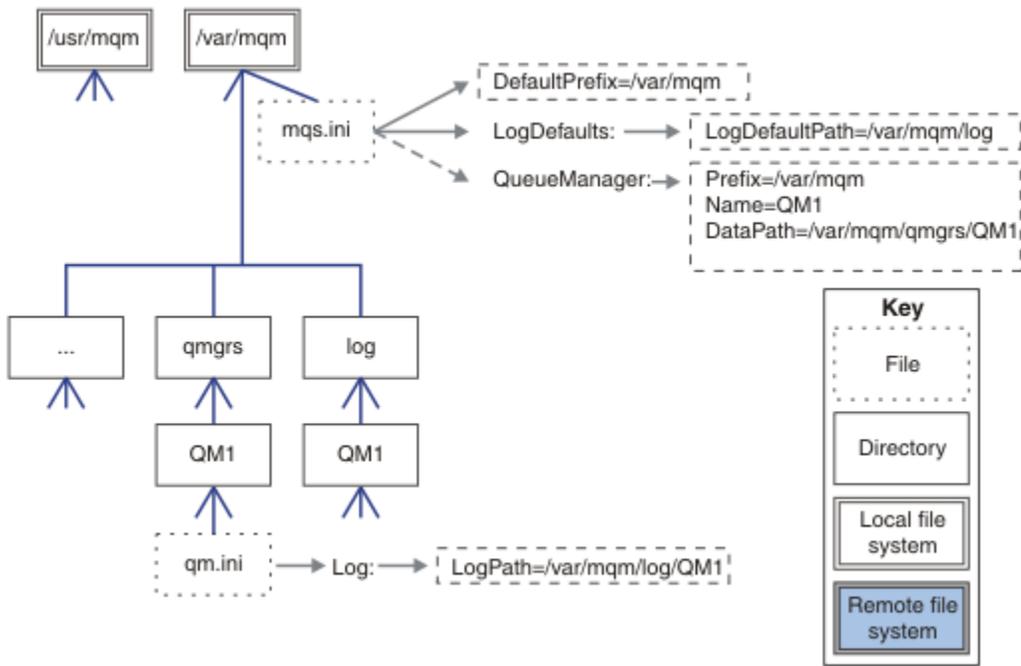


圖 48: UNIX 和 Linux 系統的預設 IBM MQ 目錄結構範例

共用預設 qmgrs 和 log 目錄 (從 v7.0.1 開始)

第 110 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 109 頁的圖 49)。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。

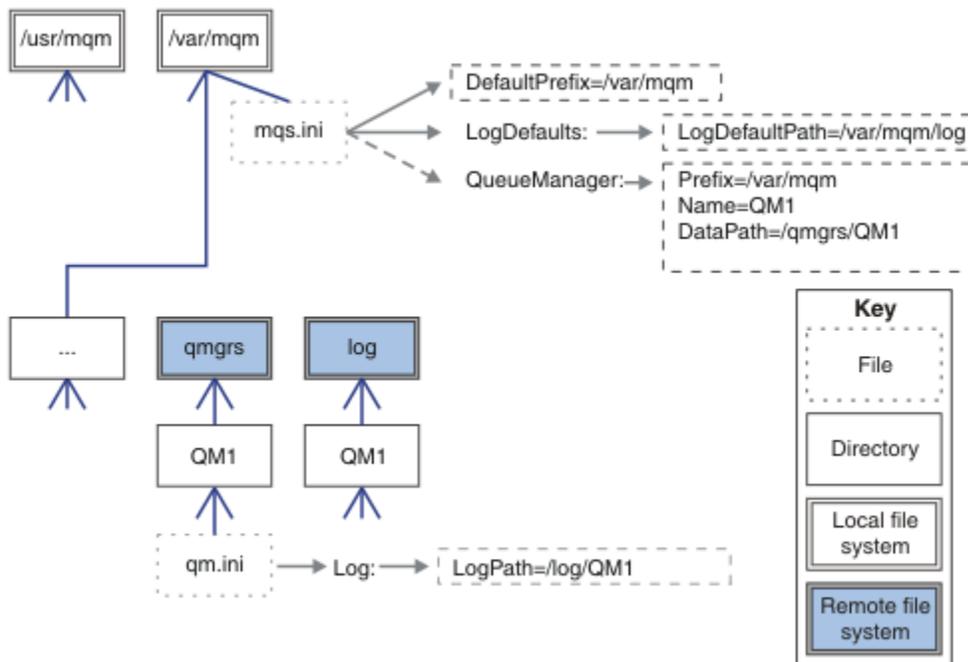


圖 49: 共用 qmgrs 和 log 目錄

共用具名 qmgrs 及 log 目錄 (從 v7.0.1 開始)

第 110 頁的圖 50 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 110 頁的圖 50 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`。

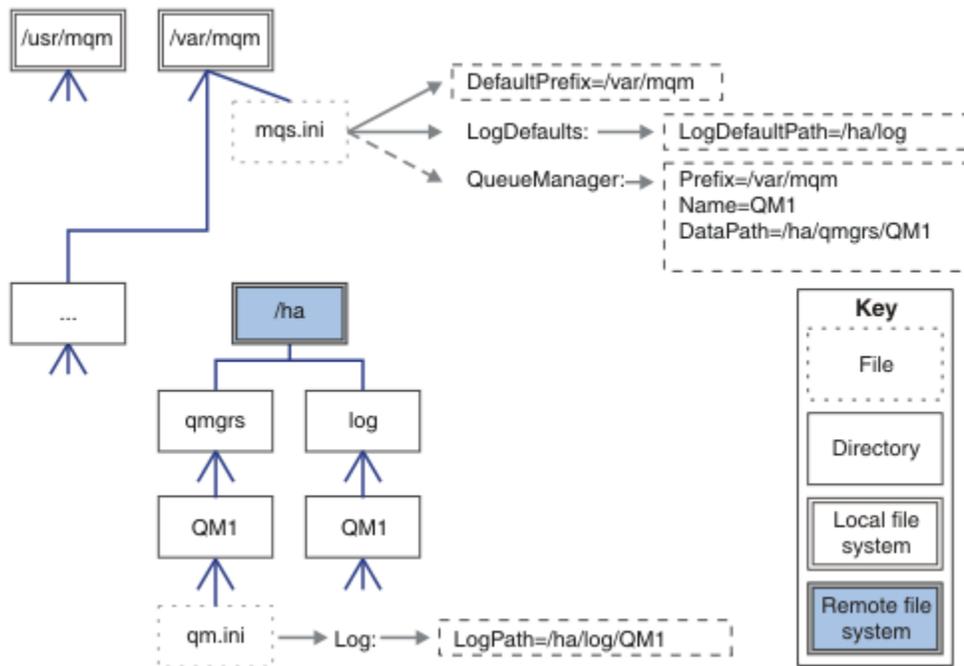


圖 50: 共用指名的 *qmgrs* 及 *log* 目錄

共用所有項目 (從 v7.0.1 版開始)

第 111 頁的圖 51 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。

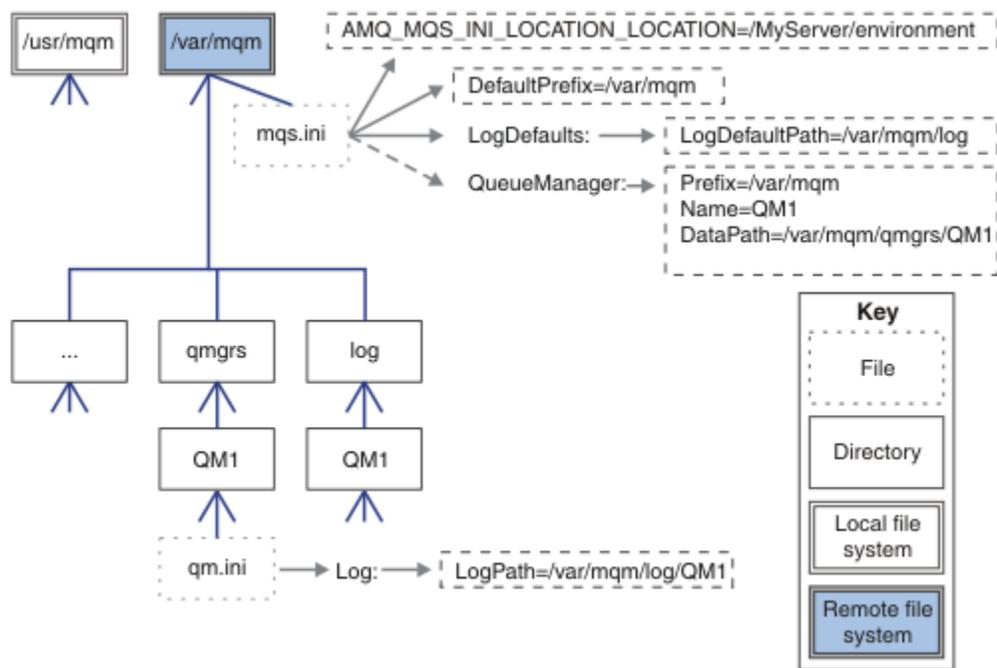


圖 51: 共用所有項目

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 MQ for Windows 安裝的預設目錄為：

32 位元

C:\Program Files (x86) \WebSphere MQ

64 位元程式目錄

C:\Program Files\IBM\WebSphere MQ

資料目錄

C:\ProgramData\IBM\MQ

重要：對於 Windows 安裝，會按照上述內容使用目錄，除非 IBM MQ 有先前的安裝仍包含登錄項目及/或佇列管理程式。在此狀況下，新的安裝將使用舊的資料目錄位置。如需相關資訊，請參閱程式和資料目錄位置。

Windows 32 位元用戶端安裝在 32 位元機器上，依預設，只會安裝到 C:\Program Files\IBM\WebSphere MQ 中。

Windows 64 位元的用戶端安裝在 64 位元的機器上，依預設，只會安裝到 C:\Program Files\IBM\WebSphere MQ 中。

如果您想要知道正在使用的安裝目錄及資料目錄，請執行 `dspmqr` 指令。

安裝目錄會列在 **InstPath** 欄位中，資料目錄會列在 **DataPath** 欄位中。

例如，執行 **dspmqr** 指令會顯示下列資訊：

```
>dspmqr
Name:      WebSphere MQ
Version:   8.0.0.0
Level:     p000-L140303.5
BuildType: IKAP - (Production)
Platform:  IBM MQ for Windows (x64 platform)
```

```

Mode:          64-bit
O/S:          Windows 7 Professional x64 Edition, Build 7601: SP1
InstName:     Installation1
InstDesc:
Primary:      Yes
InstPath:     C:\Program Files\IBM\WebSphere MQ
DataPath:     C:\ProgramData\IBM\MQ
MaxCmdLevel: 800
LicenseType:  Production

```

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. `mqs.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 `mqs.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 IBM MQ 目錄的位置和內容。

IBM MQ 配置有三組主要檔案和目錄：

1. 執行檔，以及只有在套用維護時才會更新的其他唯讀檔案。例如：
 - Readme 檔
 - IBM MQ Explorer 外掛程式和說明檔
 - 授權檔

這些檔案在 [第 112 頁的表 14](#) 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 113 頁的表 15](#) 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 113 頁的表 16](#) 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 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 |
|---------------------------------------|----------------------------|
| <code>FilePath\bin</code> | 指令和 DLL |
| <code>FilePath\bin64</code> | 指令及 DLL (64 位元) |
| <code>FilePath\conv</code> | 資料轉換表 |
| <code>FilePath\doc</code> | 精靈說明檔 |
| <code>FilePath\MQExplorer</code> | 「瀏覽器」和「瀏覽器」說明 Eclipse 外掛程式 |
| <code>FilePath\gskit8</code> | 廣域安全套件 |
| <code>FilePath\java</code> | Java 資源，包括 JRE |
| <code>FilePath\licenses</code> | 授權資訊 |
| <code>FilePath\Non_IBM_License</code> | 授權資訊 |

| 表 14: <i>FilePath</i> 目錄中的目錄和檔案 (繼續) | |
|--------------------------------------|----------|
| 檔案路徑 | 內容 |
| <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。

| 表 15: <i>DefaultPrefix</i> 目錄中的目錄和檔案 | |
|--------------------------------------|--|
| 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 資料轉換 中所述 |
| <i>DefaultPrefix</i> \errors | 非佇列管理程式錯誤日誌 AMQERR nn.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 第 113 頁的表 16 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 16: <i>DataPath</i> 和 <i>Prefix/Qmgrs/QmgrName</i> 目錄中的目錄和檔案 | |
|--|----------------------------|
| 檔案路徑 | 內容 |
| <i>DataPath</i> \@ipcc | AMQCLCHL.TAB 用戶端連線表格的預設位置。 |
| <i>DataPath</i> \authinfo | 內部使用。 |
| <i>DataPath</i> \channel | |
| <i>DataPath</i> \clntconn | |
| <i>DataPath</i> \errors | 錯誤日誌， AMQERR nn.LOG |

表 16: *DataPath* 和 *Prefix/Qmgrs/QmgrName* 目錄中的目錄和檔案 (繼續)

| 檔案路徑 | 內容 |
|-------------------------------------|----------|
| <i>DataPath</i> \listener | 內部使用。 |
| <i>DataPath</i> \namelist | |
| <i>DataPath</i> \plugcomp | |
| <i>DataPath</i> \procdef | |
| <i>DataPath</i> \qmanager | |
| <i>DataPath</i> \queues | |
| <i>DataPath</i> \services | |
| <i>DataPath</i> \ssl | |
| <i>DataPath</i> \startprm | |
| <i>DataPath</i> \topic | |
| <i>DataPath</i> \active | |
| <i>DataPath</i> \active.dat | |
| <i>DataPath</i> \amqalchk.fil | |
| <i>DataPath</i> \master | |
| <i>DataPath</i> \master.dat | |
| <i>DataPath</i> \qm.ini | 佇列管理程式配置 |
| <i>DataPath</i> \qmstatus.ini | 佇列管理程式狀態 |
| <i>Prefix</i> \Qmgrs\QmgrName | 內部使用 |
| <i>Prefix</i> \Qmgrs\@SYSTEM | 未使用 |
| <i>Prefix</i> \Qmgrs\@SYSTEM\errors | |

IBM i 上的目錄結構

提供 IFS 的說明，並說明伺服器、用戶端及 Java 的 IBM MQ IFS 目錄結構。

整合檔案系統 (IFS) 是 IBM i 的一部分，支援類似於個人電腦 (UNIX 和 Linux 作業系統) 的串流輸入/輸出及儲存體管理，同時為伺服器中儲存的所有資訊提供整合結構。

在 IBM i 上，目錄名稱以字元 & (ampersand) 開頭，而不是字元 @ (at)。例如，IBM i 上的 @system 是 &system。

IBM MQ 伺服器的 IFS 根檔案系統

當您安裝 IBM MQ Server for IBM i 時，會在 IFS 根檔案系統中建立下列目錄。

ProdData:

概觀

QIBM

```
'-- ProdData
    '-- mqm
    '-- doc
    '-- inc
    '-- lib
```

```
'-- samp
'-- licenses
'-- LicenseDoc
'-- 5724H72_V8R0M0
```

/QIBM/ProdData/mqm

下面的子目錄包含所有產品資料，例如 C++ 類別、追蹤格式檔案及授權檔。每次安裝產品時，都會刪除並取代此目錄中的資料。

/QIBM/ProdData/mqm/doc

CL 指令的「指令參考手冊」以 HTML 格式提供，並安裝在這裡。

/QIBM/ProdData/mqm/inc

用於編譯 C 或 C++ 程式的標頭檔。

/QIBM/ProdData/mqm/lib

MQ 使用的輔助檔案。

/QIBM/ProdData/mqm/samp

進一步的樣本。

/QIBM/ProdData/mqm/licenses

授權檔。每種語言的兩個檔案命名類似 LA_ *xx* 和 LI_ *xx*，其中 *xx* 是所提供每種語言的 2 個字元語言 ID。

此外，下列目錄會儲存授權合約檔案：

/QIBM/ProdData/LicenseDoc/5724H72_V8R0M0

授權檔。這些檔案的名稱類似於 5724H72_V8R0M0_ *xx*，其中 *xx* 是所提供每一種語言的 2 或 5 個字元語言 ID。

UserData:

概觀

QIBM

```
'-- UserData
'-- mqm
'-- errors
'-- trace
'-- qmgrs
'-- &system
'-- qmgrname1
'-- qmgrname2
'-- and so on
```

/QIBM/UserData/mqm

下面的子目錄包含與佇列管理程式相關的所有使用者資料。

當您安裝產品時，會在 /QIBM/UserData/mqm/ 目錄中建立 mqs.ini 檔案 (除非先前安裝已存在該檔案)。

當您建立佇列管理程式時，會在 /QIBM/UserData/mqm/qmgrs/ *QMGRNAME* / 目錄中建立 qm.ini 檔案 (其中 *QMGRNAME* 是佇列管理程式的名稱)。

刪除產品時，會保留目錄中的資料。

IBM MQ MQI client 的 IFS 根檔案系統

當您安裝 IBM MQ MQI client for IBM i 時，會在 IFS 根檔案系統中建立下列目錄：

ProdData:

概觀

QIBM

```
'-- ProdData
    '-- mqm
    '-- lib
```

/QIBM/ProdData/mqm

此目錄下的子目錄包含所有產品資料。每次取代產品時都會刪除並取代此目錄中的資料。

UserData:

概觀

QIBM

```
'-- UserData
    '-- mqm
    '-- errors
    '-- trace
```

/QIBM/UserData/mqm

此目錄下的子目錄包含所有使用者資料。

IBM MQ Java 的 IFS 根檔案系統

當您在 IBM i 上安裝 IBM MQ Java 時，會在 IFS 根檔案系統中建立下列目錄：

ProdData:

概觀

QIBM

```
'-- ProdData
    '-- mqm
    '-- java
    '-- samples
    '-- bin
    '-- lib
```

/QIBM/ProdData/mqm/java

此子目錄包含所有產品資料，包括 Java 類別。每次取代產品時都會刪除並取代此目錄中的資料。

/QIBM/ProdData/mqm/java/samples

下面的子目錄包含所有範例 Java 類別和資料。

伺服器及用戶端安裝所建立的檔案庫

安裝 IBM MQ 伺服器或用戶端會建立下列程式庫：

- QMQM
產品檔案庫。
- QMQMSAMP
範例程式庫 (如果您選擇安裝範例)。
- QMxxxx
僅限伺服器。

每次建立佇列管理程式時，IBM MQ 會自動建立相關聯的程式庫，其名稱類似於 QMxxxx，其中 xxxx 衍生自佇列管理程式名稱。此檔案庫包含佇列管理程式特定的物件，包括異動日誌及相關聯的接收端。依預設，此程式庫的名稱衍生自以 QM 字元為字首的佇列管理程式名稱。例如，對於稱為 TEST 的佇列管理程式，檔案庫稱為 QMTEST。

註: 當您建立佇列管理程式時, 如果您想要的話, 可以指定其檔案庫的名稱。例如:

```
CRTMQM MQMNAME(TEST) MQMLIB(TESTLIB)
```

您可以使用 `WRKLIB` 指令, 列出 IBM MQ for IBM i 已建立的所有檔案庫。針對佇列管理程式庫, 您會看到文字 `QMGR: QMGRNAME`。指令的格式為:

```
WRKLIB LIB(QM*)
```

刪除產品時, 會保留這些佇列管理程式相關聯的檔案庫。

IBM 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** 和 **Linux** 系統上執行的 **IBM MQ**。

IBM MQ 使用 System V 跨處理程序通訊 (IPC) 資源 (*semaphores* 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 MQ MQI clients 不使用 IPC 資源, IBM MQ 追蹤控制除外。請使用 UNIX 指令 `ipcs -a` 來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, 影響其連接至 IBM MQ 佇列管理程式的能力。IBM MQ 的每個標準連線都會使用共用記憶體, 但與其他 UNIX 和 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 IBM 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特性, 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。

IBM MQ MQI client 應用程式及所有 64 位元處理程序不受此限制影響。不論是否已設定 `EXTSHM`, 它們都可以連接至 IBM MQ 佇列管理程式。

IBM MQ 和 UNIX 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** 和 **Linux** 系統上執行的 **IBM 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 IBM 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 `runmqtsr`) 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，請使用與佇列管理程式相同的 *nice* 值來啟動 IBM 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 MQ 用戶端環境

當您規劃 IBM 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 MQ 用戶端執行所在的環境。

在規劃 IBM MQ Client for HP Integrity NonStop Server 架構之前，請先熟悉基本 IBM MQ Client for HP Integrity NonStop Server 概念，請參閱 [IBM 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 TAACL 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 TAACL。

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 MQ 和 HP NonStop TMF

HP Integrity NonStop Server 的 IBM 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 7.1 或更新版本時，才支援與 HP NonStop TMF 協調交易。

IBM 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 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 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 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 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 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 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 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 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 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 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 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 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。

在 z/OS 上規劃 IBM MQ 環境

規劃 IBM MQ 環境時，您必須考量資料集、頁面集、Db2、「連結機能」的資源需求，以及記載及備份機能的需求。請利用這個主題來規劃 IBM MQ 執行所在的環境。

在規劃 IBM MQ 架構之前，請先熟悉基本 IBM MQ for z/OS 概念，請參閱 [IBM MQ for z/OS 概念](#) 中的主題。

相關概念

[第 5 頁的『規劃』](#)

規劃 IBM MQ 環境時，請考量 IBM MQ 為單一及多個佇列管理程式架構提供的支援，以及點對點及發佈/訂閱傳訊樣式的支援。同時規劃您的資源需求，以及使用記載和備份機能。

相關資訊

[IBM MQ 技術概觀](#)

[配置 z/OS](#)

[管理 IBM MQ for z/OS](#)

z/OS 在 z/OS 上規劃儲存體及效能需求

您必須為 IBM MQ 系統設定實際且可達成的儲存體及效能目標。使用本主題可協助您瞭解影響儲存體及效能的因素。

本主題包含 IBM MQ for z/OS 的儲存體及效能需求的相關資訊。它包含下列區段：

- [IBM MQ 的 z/OS 效能選項](#)
- [判斷 z/OS 工作量管理重要性及速度目標](#)
- [第 120 頁的『磁帶庫儲存體』](#)
- [第 121 頁的『系統 LX 使用情形』](#)
- [第 121 頁的『位址空間儲存體』](#)
- [第 124 頁的『資料儲存』](#)

如需相關資訊，請參閱 [第 125 頁的『在何處尋找儲存體及效能需求的相關資訊』](#)。

IBM MQ 的 z/OS 效能選項

使用工作量管理，您可以定義效能目標，並將業務重要性指派給每一個目標。您以商業術語定義工作的目標，系統會決定應該提供多少資源 (例如處理器及儲存體) 給工作以符合其目標。工作量管理會根據您提供的目標來控制分派優先順序。工作量管理會視需要提高或降低優先順序，以符合指定的目標。因此，您不需要細部調整系統中每一項工作的確切優先順序，而可以專注於商業目標。

目標有三種：

回應時間

您希望處理工作的速度

執行速度

工作在備妥時應執行的速度，不會因處理器、儲存體、I/O 存取及佇列延遲而延遲

任意

沒有效能目標之低優先順序工作的種類

回應時間目標適用於一般使用者應用程式。例如，CICS 使用者可以將工作量目標設為回應時間目標。對於 IBM MQ 位址空間，速度目標更適合。在佇列管理程式中完成的少量工作會計入此速度目標，但此工作對於效能至關重要。佇列管理程式所完成的大部分工作都會計入一般使用者應用程式的效能目標。通道起始程式位址空間所完成的大部分工作都會計入自己的速度目標。通道起始程式所完成的 IBM MQ 訊息接收及傳送，對於使用它們的商業應用程式效能通常很重要。

判定 z/OS 工作量管理重要性及速度目標

如需相關資訊，請參閱 [第 121 頁的『判定 z/OS 工作量管理重要性』](#)。

磁帶庫儲存體

您必須為產品檔案庫配置儲存體。確切的數字視您的配置而定，但配送程式庫所需的空間預估為 80 MB。目標程式庫大約需要 72 MB。此外，您還需要 SMP/E 程式庫的空間。

IBM MQ for z/OS 使用的目標程式庫使用 PDS 或 PDSE 格式。請確定未在 Sysplex 外部共用任何 PDSE 目標程式庫。如需必要程式庫及其大小和必要格式的相關資訊，請參閱 [WebSphere MQ for z/OS 的程式目錄](#)。

系統 LX 使用情形

每一個已定義的 IBM MQ 子系統會在 IPL 時保留一個系統鏈結索引 (LX)，並在佇列管理程式啟動時保留一些非系統鏈結索引。當佇列管理程式停止並重新啟動時，會重複使用系統鏈結索引。同樣地，分散式佇列會保留一個非系統鏈結索引。在 z/OS 系統定義的系統 LX 不適當的情況下，您可能需要將這些保留系統 LX 納入考量。

判定 z/OS 工作量管理重要性

如需工作量管理及透過服務定義來定義目標的完整資訊，請參閱 [z/OS MVS 規劃: 工作量管理](#)。

本主題建議如何設定相對於系統中其他重要工作的 z/OS 工作量管理重要性及速度目標。

佇列管理程式位址空間需要以高優先順序定義，因為它提供子系統服務。通道起始程式是應用程式位址空間，但通常會獲得高優先順序，以確保傳送至遠端佇列管理程式的訊息不會延遲。Advanced Message Security (AMS) 也提供子系統服務，且需要以高優先順序來定義。

請使用下列服務類別：

預設 SYSSTC 服務類別

- VTAM 及 TCP/IP 位址空間
- IRLM 位址空間 (IRLMPROC)

註：VTAM、TCP/IP 及 IRLM 位址空間的分派優先順序必須高於所有 DBMS 位址空間、其連接的位址空間及其子層位址空間。不容許工作量管理將 VTAM、TCP/IP 或 IRLM 的優先順序降低至 (或以下) 其他 DBMS 位址空間的優先順序

對於具有您所定義名稱的服務類別 (例如 **PRODREGN**)，高速目標及重要性為 **1**，適用於下列項目：

- IBM MQ 佇列管理程式、通道起始程式及 AMS 位址空間
- Db2 (所有位址空間，但 Db2 建立的儲存程序位址空間除外)
- CICS (所有區域類型)
- IMS (BMP 以外的所有區域類型)

高速目標有助於確保對所有這些位址空間儘快執行啟動和重新啟動。

CICS 和 IMS 區域的速度目標僅在啟動或重新啟動期間才重要。在交易開始執行之後，工作量管理會忽略 CICS 或 IMS 速度目標，並根據區域中執行之交易的回應時間目標來指派優先順序。這些交易目標應反映其實作之商業應用程式的相對優先順序。它們通常具有重要性值 2。任何使用 IBM MQ 的批次應用程式都應該同樣具有速度目標和重要性，以反映其實作之商業應用程式的相對優先順序。通常重要性和速度目標會小於 PRODREGN 的重要性和速度目標。

位址空間儲存體

如需 IBM MQ 元件位址空間需求的基本指引，請使用本主題。

儲存體需求可以分為下列種類：

- [共用儲存體](#)
- [佇列管理程式專用區域儲存體用量](#)
- [通道起始程式儲存體用量](#)

如需詳細資料，請參閱 [建議的區域大小](#)。

使用 31 位元位址空間，虛擬「線」會標示 16 MB 位址，而 31 位元可定址儲存體通常稱為「高於 (16MB) 線」。使用 64 位元位址空間時，會有第二條稱為 'the bar' 的虛擬線路標示 2 GB 位址。條欄將低於 2 GB 位址 (稱為「低於條欄」) 的儲存體與高於 2 GB 位址 (稱為「高於條欄」) 的儲存體分開。低於限制的儲存體使用 31 位元定址能力，高於限制的儲存體使用 64 位元定址能力。

您可以在 JCL 上使用 REGION 參數來指定 31 位元儲存體的限制，以及使用 MEMLIMIT 參數來指定高於限制儲存體的限制。MVS 結束程式可以置換這些指定的值。



小心: 該系統的運作方式已經改變。現在，「跨系統延伸服務 (XES)」會為序列化清單結構的每一個連線配置 4 GB 的高虛擬儲存體，或為鎖定結構的每一個連線配置 36 GB 的儲存體。

在此變更之前，已在資料空間中配置此儲存體。在套用此 APAR 之後，根據 IBM MQ 計算儲存體使用情形的方式，可能會發出訊息 [CSQY225E](#) 及 [CSQY224I](#)，指出佇列管理程式缺少高於限制的本端儲存體。

您也會在訊息 [CSQY220I](#) 中看到上述條欄值增加。

如需相關資訊，請參閱 IBM 支援文件 [2017139](#)。

共用儲存體

每一個 IBM MQ for z/OS 子系統都具有下列大約儲存體需求：

- CSA 4 KB
- ECSA 800 KB，加上 CSQ6SYSP 系統參數巨集的 TRACTBL 參數中指定的追蹤表格大小。如需相關資訊，請參閱 [使用 CSQ6SYSP](#)。

此外，每一個並行 IBM MQ 邏輯連線需要大約 5 KB 的 ECSA。當作業結束時，其他 IBM MQ 作業可以重複使用此儲存體。在佇列管理程式關閉之前，IBM MQ 不會釋放儲存體，因此您可以將並行邏輯連線數目上限乘以 5 KB，來計算所需的 ECSA 數量上限。並行邏輯連線數目為：

- 批次、TSO、z/OS UNIX 和 Linux 系統服務、IMS 及 Db2 SPAS 區域中連接至 IBM MQ 但未斷線的作業 (TCB)。
- 已發出 IBM MQ 要求但尚未終止的 CICS 交易
- 已建立 (用於連結連線) 但尚未毀損或記憶體回收的 JMS 連線、階段作業、TopicSessions 或 QueueSessions。
- 作用中 IBM MQ 通道。

您可以使用 ACELIM 配置參數，將限制設為佇列管理程式的邏輯連線所使用的共用儲存體。對於 Db2 儲存程序在 IBM MQ 佇列上導致作業的網站而言，ACELIM 控制項主要很重要。

從儲存程序驅動時，每一個 IBM MQ 作業都可以產生與佇列管理程式的新邏輯連線。大型 Db2 工作單元 (例如由於表格載入) 可能會導致對共用儲存體的需求過多。

AKELIM 旨在限制一般儲存體使用，並保護 z/OS 系統。當超出限制時，使用 ACELIM 會導致 IBM MQ 失敗。如需相關資訊，請參閱 [使用 CSQ6SYSP](#) 中的 [ACELIM](#) 一節。

使用 [SupportPac MP1B](#) 來格式化 STATISTICS CLASS (3) 追蹤所產生的 SMF 115 子類型 5 記錄。

如需 SMF 115 統計資料記錄的相關資訊，請參閱 [解譯 IBM MQ 效能統計資料](#)。

在標題為 ACE/PEB 的字行上，輸出中會指出 ACELIM 值所控制的子儲存區中目前的儲存體數量。SupportPac MP1B 指出使用中的位元組數。

將正常值增加足夠的邊距，以提供空間來因應成長和工作量突增。將新值除以 1024，以產生用於 ACELIM 配置的儲存體大小上限 (KB)。

通道起始程式通常需要最多 160 KB 的 ECSA 用量。

佇列管理程式專用區域儲存體用量

對於部分內部控制區塊，IBM MQ for z/OS 可以使用高於 2 GB 限制的儲存體。您可以在此儲存體中具有緩衝池，這可讓您配置更大的緩衝池 (如果有足夠的儲存體可用的話)。一般而言，緩衝池是使用高於 2 GB 限制之儲存體的主要內部控制區塊。

每一個緩衝池大小在佇列管理程式起始設定時決定，並在連接使用該緩衝池的頁集時配置儲存體給緩衝池。新的參數 LOCATION (ABOVE | BELOW) 用來指定配置緩衝區的位置。您可以使用 ALTER BUFFPOOL 指令來動態變更緩衝池的大小。

若要使用高於條欄 (64 位元) 的儲存體，您可以在佇列管理程式 JCL 中的 EXEC PGM=CSQYASCP 參數上指定 MEMLIMIT 參數 (例如 MEMLIMIT=3G) 的值。您的安裝可能已設定預設值。

您應該指定 MEMLIMIT 並指定可感應的儲存體大小，而非 MEMLIMIT = NOLIMIT，以防止潛在問題。如果您指定 NOLIMIT 或非常大的值，則具有較大大小的 ALTER BUFFPOOL 指令可以使用所有可用的 z/OS 虛擬儲存體，這會導致在系統中分頁。

從 MEMLIMIT=3G 開始，當您需要增加緩衝池的大小時，請增加此大小。

指定 MEMLIMIT = 2 GB 加上高於條欄的緩衝池大小，四捨五入至最接近的 GB。例如，對於 2 個使用 LOCATION ABOVE 配置的緩衝池，緩衝池 1 具有 10,000 個緩衝區，緩衝池 2 具有 50,000 個緩衝區。高於限制的記憶體用量等於 60,000 (緩衝區總數) * 4096 = 245,760,000 位元組 = 234.375 MB。不論 LOCATION 為何，所有緩衝池都將使用 64 位元儲存體作為控制結構。隨著緩衝池數目及這些儲存區中的緩衝區數目增加，這可能會變得很重要。良好的經驗法則是每一個緩衝區都需要額外 200 個位元組的 64 位元儲存體。若為具有 10 個緩衝池的配置，每個緩衝池需要 20,000 個緩衝區: $200 * 10 * 20,000 = 40,000,000$ ，相當於 40 MB。您可以為 MEMLIMIT 大小指定 3 GB，這將容許成長範圍 (40MB + 200MB + 2 GB，四捨五入至 3 GB)。

對於部分配置，使用緩衝池 (其緩衝區永久由實際儲存體支援) 會有顯著的效能好處。您可以對緩衝池的 PAGECLAS 屬性指定 FIXED4KB 值來達到此目的。不過，只有在 LPAR 上有足夠的實際儲存體可用時，您才應該這麼做，否則其他位址空間可能會受到影響。如需您何時應該對 PAGECLAS 使用 FIXED4KB 值的相關資訊，請參閱 IBM MQ 支援套件 MP16: IBM MQ for z/OS -產能規劃與調整

若要最小化分頁，除了佇列管理程式及通道起始程式所使用的虛擬儲存體之外，請考量實際儲存體。

在使用高於限制的儲存體之前，您應該先與 MVS 系統程式設計師討論，以確保有足夠的輔助儲存體可供尖峰時間使用，以及有足夠的實際儲存體需求來防止分頁。

註: 可能必須增加記憶體傾出資料集的大小，才能處理增加的虛擬儲存體。

使緩衝池變得太大而有 MVS 分頁可能會對效能產生不利影響。您可以考慮使用較小且不分頁的緩衝池，IBM MQ 會將訊息移入及移出頁集。

您可以從 CSQY220I 訊息監視位址空間儲存體使用情形，該訊息指出高於及低於 2 GB 限制的專用區域儲存體使用中量，以及剩餘量。

通道起始程式儲存體用量

您必須考量通道起始程式儲存體用量的兩個區域:

- 專用區域
- 會計和統計

專用區域儲存體用量

您應該對 CHINIT 指定 REGION=0M，以容許它使用低於限制的儲存體上限。通道起始程式可用的儲存體會限制 CHINIT 可以擁有的並行連線數。

每個通道在通道起始程式位址空間中使用大約 170 KB 的延伸專用區域。如果傳輸大於 32 KB 的訊息，則會依訊息大小增加儲存體。在下列情況下，會釋放此增加的儲存體:

- 對於 10 個連續訊息，傳送端或用戶端通道需要小於現行緩衝區大小的一半。
- 傳送或接收活動訊號。

儲存體會釋放以供在 Language Environment 內重複使用，但 z/OS 虛擬儲存體管理程式不會將儲存體視為可用。這表示通道數目的上限取決於訊息大小及到達型樣，以及個別使用者系統在延伸專用區域大小上的限制。在許多系統上，通道數目的上限可能大約 9000，因為延伸區域大小不可能超過 1.6 GB。使用大於 32 KB 的訊息大小會減少系統中的通道數目上限。例如，如果傳輸長度為 100 MB 的訊息，且假設延伸區域大小為 1.6 GB，則通道數目上限為 15。

通道起始程式追蹤會寫入資料空間。資料庫儲存區的大小由 **TRAXTBL** 參數控制。請參閱 [ALTER QMGR](#)。

帳戶及統計資料儲存體用量

您應該容許通道起始程式存取至少 256 MB 虛擬儲存體，並且您可以透過指定 **MEMLIMIT=256M** 來執行此動作。

如果您未在通道起始程式 JCL 中設定 **MEMLIMIT** 參數，則可以使用 **SYS1.PARMLIB** 的 **SMFPRMxx** 成員中的 **MEMLIMIT** 參數，或從 **IEFUSI** 結束程式，來設定高於限制的虛擬儲存體數量。

如果您設定 **MEMLIMIT** 來限制上述限制儲存體低於所需的層次，則通道起始程式會發出 **CSQX124E** 訊息，且類別 4 結算及統計資料追蹤將無法使用。

建議的區域大小

下表顯示區域大小的建議值。

| 定義設定 | 系統 |
|--------|------------------------|
| 佇列管理程式 | REGION=0M, MEMLIMIT=3G |
| 通道起始程式 | REGION=0M |

管理 MEMLIMIT 和 REGION 大小

其他機制 (例如 **SYS1.PARMLIB** 的 **SMFPRMxx** 成員或 **IEFUSI** 結束程式中的 **MEMLIMIT** 參數) 可以在安裝時使用，以提供 z/OS 位址空間高於限制的預設虛擬儲存體量。如需限制高於限制的儲存體的完整詳細資料，請參閱 [高於限制的記憶體管理](#)。

資料儲存

當規劃日誌資料集、Db2 儲存體、連結機能儲存體及頁面資料集的資料儲存體需求時，請使用本主題。

請與儲存體管理者合作，以判定放置佇列管理程式資料集的位置。例如，您的儲存體管理者可能會為您提供特定 DASD 磁區，或不同資料集類型的 SMS 儲存類別、資料類別及管理類別。

- 日誌資料集必須位於 DASD 上。這些日誌可能具有高 I/O 活動且回應時間短，且不需要備份。
- 保存日誌可以在 DASD 或磁帶上。建立之後，除非發生異常狀況 (例如從備份回復頁集)，否則可能再也無法讀取它們。他們應該有很長的保留日期。
- 頁集可能具有低至中的活動，且應該定期備份。在高使用率系統上，應該每天備份兩次。
- BSDS 資料集應該每日備份; 它們沒有高 I/O 活動。

所有資料集都類似於 Db2 所使用的資料集，且類似的維護程序可用於 IBM MQ。

如需如何規劃資料儲存體的詳細資料，請參閱下列各節：

- **日誌和保存儲存體**

第 146 頁的『[我需要保留保存日誌的時間](#)』說明如何根據 IBM MQ 系統處理的訊息量，以及將作用中日誌卸載至保存資料集的頻率，來決定作用中日誌及保存資料集所需的儲存體量。

- **Db2 儲存體**

第 140 頁的『[Db2 儲存體](#)』說明如何判定 Db2 需要 IBM MQ 資料的儲存體量。

- **連結機能儲存體**

第 132 頁的『[定義連結機能資源](#)』說明如何判定建立連結機能結構的大小。

- **頁集和訊息儲存**

第 125 頁的『[規劃頁面集及緩衝池](#)』說明如何根據應用程式交換的訊息大小、這些訊息的數目，以及建立或交換這些訊息的速率，來決定頁面資料集所需的儲存體量。

在何處尋找儲存體及效能需求的相關資訊

請使用本主題作為參照，以尋找儲存體及效能需求的相關資訊。

您可以從下列來源找到相關資訊：

| 表 18: 何處可找到儲存體需求的相關資訊 | |
|------------------------|--|
| 主題 | 在何處尋找 |
| 系統參數 | 使用 CSQ6SYSP 及 自訂佇列管理程式 |
| 安裝 IBM MQ 所需的儲存體 | WebSphere MQ for z/OS 的程式目錄 |
| IEALIMIT 和 IEFUSI 結束程式 | <i>MVS Installation Exits</i> , 可從 zSeries 網站 z/OS 網際網路檔案庫 取得。 |
| 最新資訊 | IBM MQ SupportPac 網站 商業整合- WebSphere MQ SupportPacs 。 |
| 工作量管理及透過服務定義來定義目標 | <i>z/OS MVS 規劃: 工作量管理</i> |

規劃頁面集及緩衝池

協助您規劃頁面資料集及緩衝池的起始數目及大小的資訊。

本主題包含下列各節：

- [第 125 頁的『規劃您的頁面集』](#)
 - [頁集用法](#)
 - [頁集數目](#)
 - [頁集大小](#)
- [第 126 頁的『計算頁面集的大小』](#)
 - [頁集零](#)
 - [頁集 01-99](#)
 - [計算訊息的儲存體需求](#)
- [第 128 頁的『啟用動態頁集擴充』](#)
- [第 129 頁的『定義緩衝池』](#)

規劃您的頁面集

頁集使用情形

對於短期訊息，通常會在頁集上使用少數頁面，而且除了在啟動、檢查點期間或關閉時，資料集的 I/O 很少或沒有 I/O。

對於長期存在的訊息，包含訊息的那些頁面通常會寫出至磁碟。此作業由佇列管理程式執行，以減少重新啟動時間。

將短期訊息與長期訊息分開，方法是將它們放在不同的頁集及不同的緩衝池中。

頁集數目

使用數個大型頁面集可以讓 IBM MQ 管理者的角色更容易，因為這表示您需要較少的頁面集，讓佇列至頁面集的對映更簡單。

使用多個較小的頁面集有許多優點。例如，它們需要較少的時間來備份，而 I/O 可以在備份及重新啟動期間平行執行。不過，請考量這會增加 IBM MQ 管理者角色的顯著效能成本，該管理者需要將每一個佇列對映至更多頁集的其中一個。

至少定義五個頁面集，如下所示：

- 保留給物件定義的頁集 (頁集零)
- 系統相關訊息的頁面集
- 效能關鍵長期訊息的頁面集
- 效能關鍵短期訊息的頁面集
- 所有其他訊息的頁面集

第 129 頁的『定義緩衝池』說明以這種方式在頁集上配送訊息的效能優點。

頁集大小

在頁集中為預期的尖峰訊息容量定義足夠的空間。請考量任何非預期的尖峰容量，例如當因為佇列伺服器程式不在執行中而開發訊息建置時。您可以透過配置具有次要延伸範圍的頁集，或者透過啟用動態頁集擴充來執行此動作。如需相關資訊，請參閱第 128 頁的『啟用動態頁集擴充』。

規劃頁面集大小時，請考量可能產生的所有訊息，包括非應用程式訊息資料。例如，觸發應用程式所要求的訊息、事件訊息及任何報告訊息。

頁集的大小決定從備份還原回復頁集所花費的時間，因為大型頁集需要較長的還原時間。

註：頁集的回復也取決於佇列管理程式在處理自備份以來寫入的日誌記錄時所花費的時間；此時段由備份頻率決定。如需相關資訊，請參閱第 152 頁的『規劃備份及回復』。

註：大於 4 GB 的頁集需要使用 SMS 延伸定址能力。

計算頁面集的大小

對於佇列管理程式物件定義 (例如，佇列及處理程序)，很容易計算儲存體需求，因為這些物件是固定大小且是永久的。不過，對於訊息，由於下列原因，計算更為複雜：

- 訊息大小不同。
- 訊息是暫時性的。
- 非同步處理程序會定期收回已擷取訊息所佔用的空間。

如果網路停止，則可以建立大於 4 GB 的大型分頁集，以提供額外的訊息容量 (必要的話)。無法修改現有的頁面集。相反地，必須建立具有延伸定址能力及延伸格式屬性的新頁集。新頁集必須與舊頁集的實體大小相同，然後舊頁集必須複製到新頁集。如果需要反向移轉，則不得變更頁集零。如果少於 4 GB 的頁集已足夠，則不需要任何動作。

頁集零

對於頁集零，所需的儲存體為：

```
(maximum number of local queue definitions x 1010)
(excluding shared queues)
+ (maximum number of model queue definitions x 746)
+ (maximum number of alias queue definitions x 338)
+ (maximum number of remote queue definitions x 434)
+ (maximum number of permanent dynamic queue definitions x 1010)
+ (maximum number of process definitions x 674)
+ (maximum number of namelist definitions x 12320)
+ (maximum number of message channel definitions x 2026)
+ (maximum number of client-connection channel definitions x 5170)
+ (maximum number of server-connection channel definitions x 2026)
+ (maximum number of storage class definitions x 266)
+ (maximum number of authentication information definitions x 1010)
+ (maximum number of administrative topic definitions x 15000)
+ (total length of topic strings defined in administrative topic definitions)
```

將此值除以 4096，以決定要在叢集中為頁集資料集指定的記錄數。

您不需要容許儲存在共用儲存庫中的物件，但必須容許儲存或複製到頁集零的物件 (處置方式為 GROUP 或 QMGR 的物件)。

您可以建立的物件總數受限於頁集零的容量。您可以定義的本端佇列數目限制為 524 287。

頁集 01-99

對於頁集 01-99，每一個頁集所需的儲存體取決於該頁集上儲存的訊息數目及大小。(共用佇列上的訊息不會儲存在頁集上。)

將此值除以 4096，以決定要在叢集中為頁集資料集指定的記錄數。

計算訊息的儲存體需求

本節說明如何將訊息儲存在頁面上。瞭解這可協助您計算必須為訊息定義多少頁集儲存體。若要計算頁集上所有訊息所需的大約空間，您必須考量對映至頁集之所有佇列的佇列深度上限，以及那些佇列上訊息的平均大小。

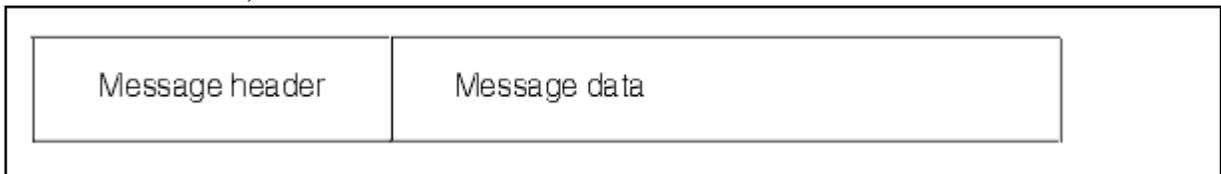
註：本節提供的結構及控制資訊的大小可能會在主要版本之間變更。如需 IBM MQ 版本特定的詳細資料，請參閱 [SupportPac MP16 - WebSphere MQ for z/OS 產能規劃與調整](#) 和 [MP1E / MP1F / MP1G - WebSphere MQ for z/OS Vx.x.x 效能報告](#)

您必須容許訊息「取得」可能因 IBM MQ 無法控制的原因 (例如，因為通訊協定發生問題) 而延遲。在此情況下，訊息的「放置」速率可能遠超過「取得」速率。這可能會導致儲存在頁面集中的訊息數目大幅增加，並導致所需儲存體大小增加。

頁集中的每一頁長度為 4096 個位元組。容許固定標頭資訊，每一個頁面都有 4057 個位元組的空間可用於儲存訊息。

在計算每一個訊息所需的空間時，您必須考量訊息是否適合一頁 (簡訊)，或是否需要分割成兩頁或多頁 (長訊息)。以這種方式分割訊息時，您必須在空間計算中容許其他控制資訊。

基於空間計算的目的，訊息可以如下所示：



訊息標頭區段包含訊息描述子及其他控制資訊，其大小視訊息大小而有所不同。訊息資料區段包含所有實際訊息資料及任何其他標頭 (例如，傳輸標頭或 IMS 橋接器標頭)。

頁集控制資訊至少需要兩頁，通常小於訊息所需空間總計的 1%。

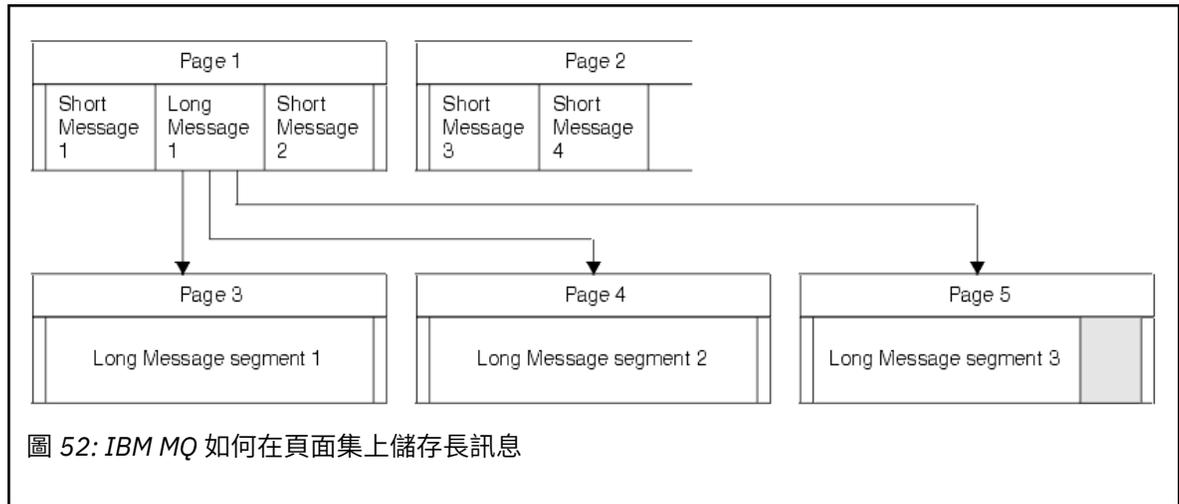
簡訊

短訊定義為適合一頁的訊息。

從 IBM WebSphere MQ 7.0.1 開始，小型訊息會儲存在每一個頁面上。

長訊息

如果訊息資料的大小大於 3596 個位元組，但不大於 4 MB，則會將訊息分類為長訊息。呈現長訊息時，IBM MQ 會將訊息儲存在一系列頁面上，並以儲存簡訊的相同方式儲存指向這些頁面的控制資訊。這在 [第 128 頁的圖 52](#) 中顯示：



很長的訊息

非常長的訊息是大大於 4 MB 的訊息。這些會儲存成每 4 MB 使用 1037 頁。任何餘數都以與長訊息相同的方式儲存，如上述。

啟用動態頁集擴充

在佇列管理程式執行時，可以動態延伸頁集。頁集可以有 119 個延伸範圍，且可以分散在多個磁碟區上。

每次展開頁面集時，都會使用新的資料集範圍。佇列管理程式會在必要時繼續擴充頁集，直到達到延伸範圍數目上限，或直到合格磁區上不再有儲存體可供配置為止。

一旦頁集擴充因上述其中一個原因而失敗，佇列管理程式會將頁集標示為不進一步擴充嘗試。可以透過將頁集變更為 EXPAND (SYSTEM) 來重設此標記。

當配置頁集中 90% 的現有空間時，頁集展開會以非同步方式進行所有其他頁集活動。

頁集擴充處理程序會格式化新配置的延伸範圍，並使它可供佇列管理程式使用。不過，在整個延伸範圍格式化之前，沒有任何空間可供使用。這表示在很大程度上擴充可能需要一些時間，如果應用程式在擴充完成之前填滿剩餘的 10% 頁集，則可能會「封鎖」應用程式。

範例 `thlqual.SCSQPROC(CSQ4PAGE)` 顯示如何定義次要延伸範圍。

若要控制新延伸範圍的大小，您可以使用 `DEFINE PSID` 及 `ALTER PSID` 指令之 `EXPAND` 關鍵字的下列其中一個選項：

- 使用者
- 系統
- 無

使用者

使用配置頁集時指定的次要範圍大小。如果未指定值，或指定零值，則無法進行動態頁集擴充。

當頁面中的空間已使用 90% 時，即會展開頁集，並與其他頁集活動非同步執行。

這可能導致一次擴充超過單一範圍。

請考量下列範例：您配置一個頁面集，其主要延伸範圍為 100000 頁，次要延伸範圍為 5000 頁。放置需要 9999 頁的訊息。如果頁集已使用 85,000 頁，則寫入訊息會跨越 90% 完整界限 (90,000 頁)。此時，會將另一個次要範圍配置給 100,000 頁的主要範圍，將頁面集大小設為 105,000 頁。訊息的其餘 4999 頁會繼續寫入。當已使用的頁面空間達到 94,500 頁 (等於已更新頁面集大小 105,000 頁的 90%) 時，會配置另一個 5000 頁範圍，將頁面集大小設為 110,000 頁。在 `MQPUT` 結束時，頁集已展開兩次，並使用 94,500 頁。未使用第二頁集展開中的任何頁面，雖然已配置它們。

在重新啟動時，若先前使用的頁集已取代為更小的資料集，則會擴充它，直到它達到先前使用的資料集大小為止。要達到此大小只需要一個範圍。

系統

忽略定義頁集時指定的次要延伸範圍大小。相反地，佇列管理程式所設定的值大約是現行頁集大小的 10%。該值會四捨五入至 DASD 最接近的磁柱。

如果未指定值，或指定零值，則仍可進行動態頁集擴充。佇列管理程式所設定的值大約是現行頁集大小的 10%。新值會根據 DASD 的性質來四捨五入。

當頁集中的空間大約使用 90% 時，即會展開頁集，並與其他頁集活動非同步執行。

在重新啟動時，若先前使用的頁集已取代為更小的資料集，則會擴充它，直到它達到先前使用的資料集大小為止。

無

不會進行進一步的頁集擴充。

相關資訊

[ALTER PSID](#)

[DEFINE PSID](#)

[顯示使用情形](#)

定義緩衝池

請利用這個主題來協助規劃您應該定義的緩衝池數目及其設定。

本主題分為下列各節：

1. [第 129 頁的『決定要定義的緩衝池數目』](#)
2. [第 130 頁的『決定每一個緩衝池的起始性質』](#)
3. [第 131 頁的『在預期負載下監視緩衝池的效能』](#)
4. [第 131 頁的『調整緩衝池性質』](#)

決定要定義的緩衝池數目

您一開始應該定義四個緩衝池：

緩衝池 0

用於物件定義 (在頁集零中) 及效能重要的系統相關訊息佇列，例如 SYSTEM.CHANNEL.SYNCQ 佇列及 SYSTEM.CLUSTER.COMMAND.QUEUE 和 SYSTEM.CLUSTER.REPOSITORY.QUEUE 佇列。

不過，如果要使用大量通道或叢集作業，請務必考量調整緩衝池性質中的點 [第 131 頁的『7』](#)。

將其餘三個緩衝池用於使用者訊息。

緩衝池 1

用於重要的長期訊息。

長時間存在的訊息是指在系統中停留超過兩個檢查點的那些訊息，此時會將這些訊息寫出至頁集。如果您有許多長期存在的訊息，則此緩衝池應該相對較小，因此頁集 I/O 會平均分佈 (每次緩衝池已滿 85% 時，會將較舊的訊息寫出至 DASD)。

如果緩衝池太大，且緩衝池永遠不會達到 85% 滿，則頁集 I/O 會延遲到檢查點處理程序為止。這可能會影響整個系統的回應時間。

如果您只預期幾個長期存在的訊息，請定義此緩衝池，使其足夠大，以保留所有這些訊息。

緩衝池 2

用於效能關鍵、短期訊息。

通常會有高程度的緩衝區重複使用，使用較少的緩衝區。不過，您應該讓這個緩衝池變得很大，以容許非預期的訊息累積，例如當伺服器應用程式失敗時。

緩衝池 3

用於所有其他 (一般為效能非重要) 訊息。

佇列，例如無法傳送郵件的佇列 SYSTEM.COMMAND.* 佇列及 SYSTEM.ADMIN.* 佇列也可以對映至緩衝池 3。

如果存在虛擬儲存體限制，且緩衝池需要較小，則緩衝池 3 是減少大小的第一個候選項。

在下列情況下，您可能需要定義其他緩衝池：

- 如果已知特定佇列需要隔離，可能是因為它在不同時間表現出不同的行為。
 - 在各種情況下，這類佇列可能需要可能的最佳效能，或需要隔離，使它不會對緩衝池中的其他佇列產生不利影響。
 - 每一個這類佇列都可以隔離至其自己的緩衝池及頁集。
- 基於服務程式類別的原因，您想要彼此隔離不同的佇列集。
 - 然後，每一組佇列可能需要一種或兩種緩衝池 1 或 2，如第 130 頁的表 20 中所述，需要建立數個特定類型的緩衝池。

在 IBM MQ 8.0 for z/OS 中，您可以定義的緩衝池數目上限取決於 OPMODE 參數。如果您指定：

- OPMODE (COMPAT,800) 或對等項目，最多可以定義 16 個緩衝池
- OPMODE (NEWFUNC,800) 最多可以定義 100 個緩衝池。

決定每一個緩衝池的起始性質

在決定您需要的緩衝池數目之後，您現在需要決定那些緩衝池的起始性質。可用的性質受 OPMODE 影響，並在第 130 頁的表 19 中彙總。

| 參數名稱 / OPMODE 設定 | 單一緩衝池中的緩衝區數目上限 | 緩衝池數目上限 | 佇列管理程式中緩衝區總數上限 | 緩衝池位置 | 緩衝池頁面類別 |
|-------------------------------------|----------------|---------|--|------------|----------------------------|
| DEFINE BUFFPOOL 或 ALTER BUFFPOOL 參數 | 緩衝區 | | | 位置 (上下) | PAGECLAS (4KB/FIXED4KB) |
| OPMODE (COMPAT,800) 或對等項目 | 500 000 | 16 | 受列下方可用空間限制。可能小於 262 144 | BELOW 列 | 可分頁 (4KB) |
| OPMODE (NEWFUNC, 800) | 999 999 999 | 100 | 99 999 999 900 (如果 MEMLIMIT / IEFUSI 結束程式容許) | 「高於」或「低於」列 | 可分頁 (4KB) 或固定頁面 (FIXED4KB) |

如果您使用第 129 頁的『決定要定義的緩衝池數目』中說明的四個緩衝池，則第 130 頁的表 20 會提供兩組緩衝池大小值。

第一組適用於測試系統，另一組適用於正式作業系統或最終將變成正式作業系統的系統。不論 OPMODE 設定為何，這些值都假設緩衝池將位於條欄下方，且緩衝區將可分頁。

| 定義設定 | 測試系統 | 生產系統 |
|------------|------------|-------------|
| BUFFPOOL 0 | 1 050 個緩衝區 | 50 000 個緩衝區 |
| BUFFPOOL 1 | 1 050 個緩衝區 | 20 000 個緩衝區 |
| BUFFPOOL 2 | 1 050 個緩衝區 | 50 000 個緩衝區 |

| 表 20: 建議的緩衝池設定定義 (繼續) | | |
|-----------------------|------------|-------------|
| 定義設定 | 測試系統 | 生產系統 |
| BUFFPOOL 3 | 1 050 個緩衝區 | 20 000 個緩衝區 |

如果您需要超過四個建議的緩衝池，請選取最精確說明緩衝池中佇列預期行為的緩衝池 (1 或 2)，並使用 [第 130 頁的表 20](#) 中的資訊來調整其大小。

您可能需要減少部分其他緩衝池的大小，或重新考量緩衝池的數目，尤其是如果您已指定 OPMODE (COMPAT,800) 或對等項目時。

在預期負載下監視緩衝池的效能

您可以透過分析緩衝池效能統計資料來監視緩衝池的使用情形。特別是，您應該確保緩衝池足夠大，以便 QPSTSOS、QPSTSTLA 及 QPSTDMC 的值保持為零。

如需進一步資訊，請參閱 [緩衝區管理程式資料記錄](#)。

調整緩衝池性質

必要的話，請使用下列點來調整 [第 130 頁的『決定每一個緩衝池的起始性質』](#) 中的緩衝池設定。

使用 [第 131 頁的『在預期負載下監視緩衝池的效能』](#) 中的效能統計資料作為指引。

註: 這些點都假設您已指定 OPMODE (NEWFUNC,800)，但點 2 除外。

1. 如果您是從舊版 IBM MQ 移轉，則只有在您有更多實際儲存體可用時，才會變更現有設定。
2. 一般而言，緩衝池越大，效能越好，如果緩衝池高於限制，則緩衝池可能更大。

不過，在任何時候，您都應該有足夠的實際儲存體可用，以便緩衝池常駐在實際儲存體中。與大型緩衝池相比，最好具有不會導致分頁的較小緩衝池。

此外，沒有任何點具有大於使用它的頁集大小總計的緩衝池，雖然您應該考量頁集擴充 (如果可能發生的話)。

3. 針對每個緩衝池設定一個緩衝區頁面，因為這提供更好的應用程式隔離。
4. 如果您有足夠的實際儲存體，使您的緩衝池永遠不會被作業系統頁出，請考慮在緩衝池中使用固定頁面的緩衝區。

如果緩衝池可能會經歷大量 I/O，這尤其重要，因為它會節省與頁面相關聯的 CPU 成本-在 I/O 之前先修正緩衝區，然後再進行頁面解除修正。

5. 將緩衝池定位在條欄上方有數個好處，即使它們的大小足以容納在條欄下方。它們是：

- 31 位元虛擬儲存體限制解除-例如，共用儲存體的更多空間。
- 如果大量使用緩衝池時需要非預期地增加緩衝池的大小，則對佇列管理程式及其工作量的影響及風險會較小，因為將更多緩衝區新增至已高於限制的緩衝池，而不是將緩衝池移至高於限制的緩衝池，然後再新增更多緩衝區。

6. 針對短期訊息調整緩衝池零及緩衝池 (緩衝池 2)，以便永不超出 15% 可用臨界值 (亦即，QPSTCBSL 除以 QPSTNBUF 一律大於 15%)。如果超過 15% 的緩衝區仍然可用，則在正常作業期間，使用這些緩衝池的頁集 I/O 大致可以避免，雖然超過兩個檢查點的訊息會寫入頁集。



小心: 這些參數的最佳值取決於個別系統的性質。給定的值僅作為準則，可能不適用於您的系統。

7. SYSTEM.* 非常深的佇列，例如 SYSTEM.CHANNEL.SYNCQ 放置在其自己的緩衝池中可能會有所助益。

IBM MQ SupportPac [MP16 - WebSphere MQ for z/OS 產能規劃與調整](#) 提供調整緩衝池的進一步相關資訊。

規劃連結機能及卸載儲存體環境

當規劃連結機能 (CF) 結構及共用訊息資料集 (SMDS) 環境或 Db2 環境的起始大小和格式時，請使用本主題。

本節包含下列主題的相關資訊：

- [第 132 頁的『定義連結機能資源』](#)
 - [決定卸載儲存機制](#)
 - [規劃結構](#)
 - [規劃結構的大小](#)
 - [將共用佇列對映至結構](#)
- [第 136 頁的『規劃共用訊息資料集 \(SMDS\) 環境』](#)
- [第 139 頁的『規劃 Db2 環境』](#)

定義連結機能資源

如果您打算使用共用佇列，則必須定義 IBM MQ 將在 CFRM 原則中使用的連結機能結構。若要這樣做，您必須先使用結構的相關資訊來更新 CFRM 原則，然後啟動原則。

您的安裝可能具有現有 CFRM 原則，說明可用的「連結機能」。IXCMIAPU z/OS 公用程式用來根據您提供的文字陳述式來修改原則的內容。該公用程式在 *MVS Setting up a Sysplex* 手冊中有說明。您必須將陳述式新增至原則，以定義新結構的名稱、在其中定義它們的「連結機能」，以及結構的大小。

CFRM 原則也會判定 IBM MQ 結構是否雙工，以及在失敗實務中如何重新配置它們。[共用佇列回復](#) 包含配置 CFRM 以進行「系統管理的重建」處理的建議。

決定卸載儲存體環境

共用佇列的訊息資料可以從連結機能卸載，並儲存在 Db2 表格或稱為共用訊息資料集 (SMDS) 的 IBM MQ 受管理資料集中。必須一律卸載太大而無法儲存在連結機能中 (即大於 63 KB) 的訊息，且可以選擇性地卸載較小的訊息，以減少連結機能空間使用情形。

如需相關資訊，請參閱 [指定共用訊息的卸載選項](#)。

規劃結構

佇列共用群組至少需要定義兩個結構。第一個結構 (稱為管理結構) 用來協調佇列共用群組中的 IBM MQ 內部活動。此結構中未保留任何使用者資料。它具有固定名稱 *qsg-name* CSQ_ADMIN (其中 *qsg-name* 是佇列共用群組的名稱)。後續結構用來保留 IBM MQ 共用佇列上的訊息。每一個結構最多可以保留 512 個共用佇列。

使用多個結構

佇列共用群組最多可以連接至 64 個連結機能結構。其中一個結構必須是管理結構，其中一個結構可能是 SYSAPPL 結構。因此，IBM MQ 資料最多可以使用 63 (62 個具有 SYSAPPL) 結構。基於下列任何原因，您可以選擇使用多個結構：

- 您有一些佇列可能保留大量訊息，因此需要整個連結機能的所有資源。
- 您需要大量共用佇列，因此必須在多個結構之間分割它們，因為每一個結構只能包含 512 個佇列。
- RMF 會報告結構的使用性質，建議您應該將它包含的佇列分散在許多「連結機能」中。
- 基於資料隔離的原因，您想要將部分佇列資料保留在與其他佇列資料實際上不同的連結機能中。

- 使用結構層次屬性及指令執行持續共用訊息的回復，例如 BACKUP CFSTRUCT。若要簡化備份及回復，您可以將保留非持續訊息的佇列指派給與保留持續訊息的那些結構不同的結構。

選擇要在其中配置結構的「連結機能」時，請考量下列要點：

- 您的資料隔離需求。
- 連結機能的波動性 (即透過停電來保留資料的能力)。
- 存取系統與連結機能之間或「連結機能」之間的失敗獨立性。
- 在連結機能上安裝的連結機能「控制碼 (CFCC)」層次 (IBM MQ 需要「層次 9」或更高層次)。

規劃結構的大小

管理結構 (*qsg-name* CSQ_ADMIN) 必須夠大，才能包含佇列共用群組中每一個佇列管理程式的 1000 個清單項目。當佇列管理程式啟動時，會檢查結構是否夠大，足以容納目前定義給佇列共用群組的佇列管理程式數目。如果佇列管理程式已由 CSQ5PQSG 公用程式新增，則會被視為定義給佇列共用群組。您可以使用 MQSC DISPLAY GROUP 指令來檢查定義給群組的佇列管理程式。

第 133 頁的表 21 顯示佇列共用群組中所定義之各種佇列管理程式數目的管理結構所需大小下限。這些大小是針對 CFCC 層次 14 連結機能結構所建立；對於較高層次的 CFCC，它們可能需要更大。

| 佇列共用群組中定義的佇列管理程式數 | 必需的儲存體 |
|-------------------|----------|
| 1 | 6144 KB |
| 2 | 6912 KB |
| 3 | 7976 KB |
| 4 | 8704 KB |
| 5 | 9728 KB |
| 6 | 10496 KB |
| 7 | 11520 KB |
| 8 | 12288 KB |
| 9 | 13056 KB |
| 10 | 14080 KB |
| 11 日 | 14848 KB |
| 12 | 15616 KB |
| 13 | 16640 KB |
| 14 | 17408 KB |
| 15 | 18176 KB |
| 16 | 19200 KB |
| 17 | 19968 KB |
| 18 | 20736 KB |
| 19 | 21760 KB |
| 20 | 22528 KB |
| 21 | 23296 KB |
| 22 | 24320 KB |

| 表 21: 管理結構大小下限 (繼續) | |
|---------------------|----------|
| 佇列共用群組中定義的佇列管理程式數 | 必需的儲存體 |
| 23 | 25088 KB |
| 24 | 25856 KB |
| 25 GB | 27136 KB |
| 26 | 27904 KB |
| 27 | 28672 KB |
| 28 | 29696 KB |
| 29 | 30464 KB |
| 30 | 31232 KB |
| 31 | 32256 KB |

當您將佇列管理程式新增至現有佇列共用群組時，儲存體需求可能已增加超過第 133 頁的表 21 中建議的大小。若是如此，請使用下列程序來預估 CSQ_ADMIN 結構所需的儲存體: 發出 MQSC 指令 /pf DISPLAY CFSTATUS(*), 其中 /cpf 是針對佇列共用群組的現有成員，並擷取 CSQ_ADMIN 結構的 ENTSMAX 資訊。如果此數目小於您要在佇列共用群組中定義之佇列管理程式總數的 1000 倍 (如 DISPLAY GROUP 指令所報告)，請增加結構大小。

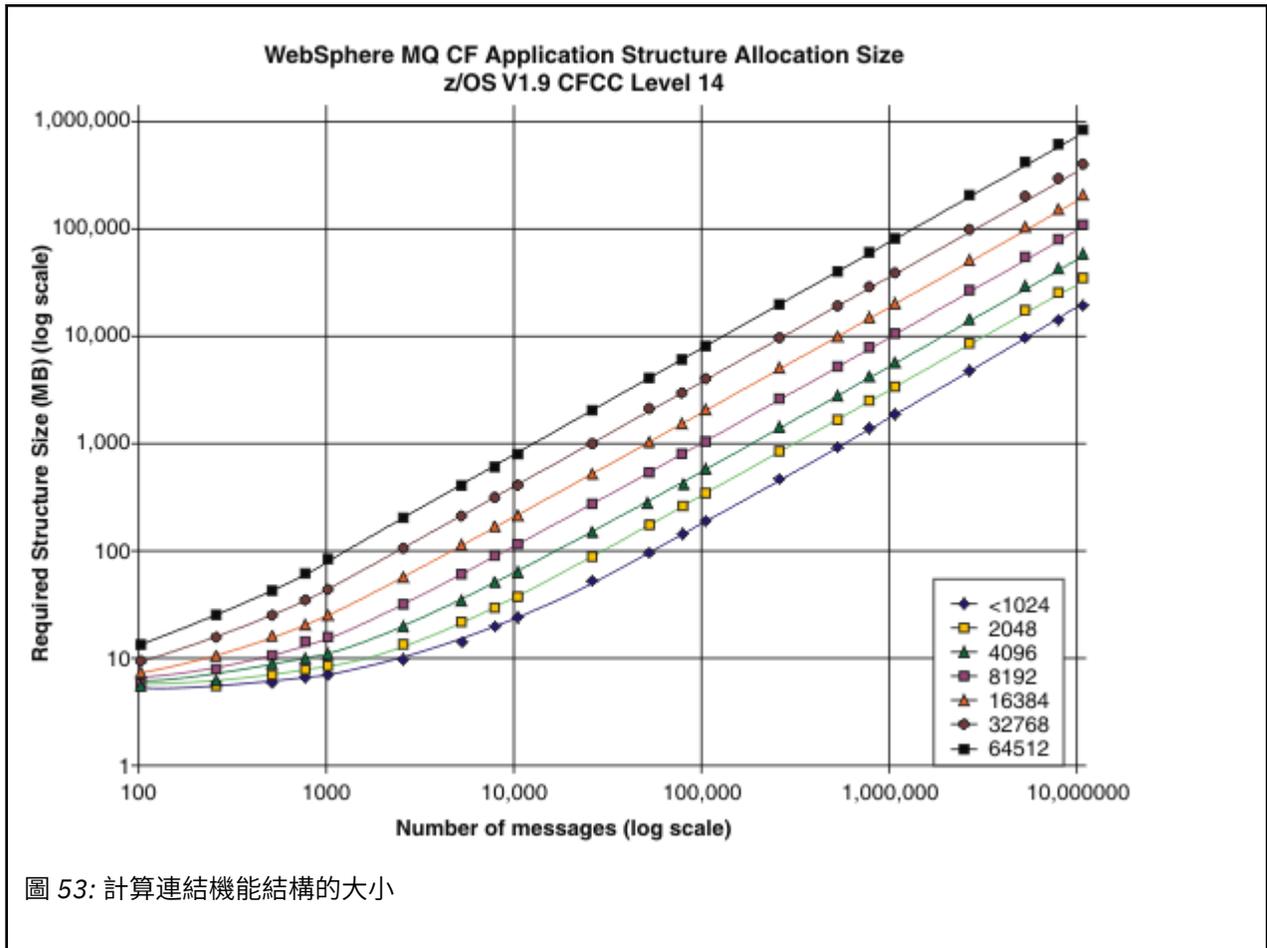
保留 IBM MQ 訊息所需的結構大小，取決於同時保留在結構上的可能訊息數目及大小，以及可能並行工作單元數目的估計值。

第 135 頁的圖 53 中的圖形顯示您應該將 CF 結構設為多大，以保留共用佇列上的訊息。若要計算配置大小，您需要知道

- 佇列上訊息的平均大小
- 可能儲存在結構中的訊息總數

尋找水平軸上的訊息數。(刻度為 2、5 和 8 的倍數。) 選取對應於訊息大小的曲線，並從垂直軸決定所需的值。例如，對於 200 000 則長度為 1 KB 的訊息，會提供 256 到 512MB 範圍內的值。

第 135 頁的表 22 以表格方式提供相同的資訊。



使用此表格來協助計算建立連結機能結構的大小:

表 22: 計算連結機能結構的大小

| 訊息數 | 1 KB | 2 KB | 4 KB | 8 KB | 16 KB | 32 KB | 63 KB |
|--------|-------|------|------|------|-------|-------|-------|
| 100 | 6 | 6 | 7 | 7 | 8 | 10 | 14 |
| 1000 | 8 | 9 | 12 | 17 | 27 | 48 | 88 |
| 10000 | 25 GB | 38 | 64 | 115 | 218 | 423 | 821 |
| 100000 | 199 | 327 | 584 | 1097 | 2124 | 4177 | 8156 |

您的 CFRM 原則應該包括下列陳述式:

INITSIZE 是 XES 在第一個連接器連接至結構時配置給結構的大小 (以 KB 為單位)。SIZE 是結構可以達到的大小上限。FULLTHRESHOLD 設定 XES 發出訊息 IXC585E 以指出結構已滿的臨界值百分比值。最佳作法是確保 INITSIZE 和 SIZE 在因數 2 內。

例如, 使用先前決定的數字, 您可以包括下列陳述式:

```
STRUCTURE NAME(structure-name)
INITSIZE(value from graph in KB, that is, multiplied by 1024)
SIZE(something larger)
FULLTHRESHOLD(85)
```

```
STRUCTURE NAME(QSG1APPLICATION1)
INITSIZE(262144) /* 256 MB */
SIZE(524288) /* 512 MB */
FULLTHRESHOLD(85)
```

如果使用的結構達到發出警告訊息的臨界值，則需要人為介入。您可以使用 IBM MQ 來禁止對結構中部分佇列的 MQPUT 作業，以防止應用程式寫入更多訊息、啟動更多應用程式以從佇列取得訊息，或靜止將訊息放入佇列的部分應用程式。

或者，您可以使用 XES 機能來變更適當的結構大小。下列 z/OS 指令：

```
SETXCF START,ALTER,STRNAME= structure-name,SIZE= newsize
```

會將結構大小變更為 *newsized*，其中 *newsized* 是小於結構 CFRM 原則上指定的 SIZE 值，但大於現行連結機能大小的值。

您可以使用 MQSC DISPLAY GROUP 指令來監視連結機能結構的使用。

如果未採取任何動作，且佇列結構填滿，則會將 MQRC_STORAGE_MEDIUM_FULL 回覆碼傳回給應用程式。如果管理結構已滿，則確切症狀取決於發生錯誤的處理程序，但它們可能包括下列問題：

- 沒有指令的回應。
- 佇列管理程式失敗，因為在確定處理期間發生問題。

某些系統佇列隨附 CFSTRUCT 屬性，這些屬性指定字首為佇列共用群組名稱的應用程式結構 CSQSYSAPPL。CSQSYSAPPL 結構是系統佇列的應用程式結構。如需建立連結機能結構的詳細資料，請參閱 [作業 10: 設定連結機能](#)。

使用預設定義，即 SYSTEM.QSG.CHANNEL.SYNCQ 和 SYSTEM.QSG.UR.RESOLUTION.QUEUE 使用此結構。表 3 示範如何預估預設佇列的訊息資料大小的範例。

| 表 23: 顯示根據調整大小的 CSQSYSAPPL 使用情形的表格。 | |
|-------------------------------------|----------------------------|
| <i>qsg-name</i> CSQSYSAPPL 用法 | 大小調整 |
| SYSTEM.QSG.CHANNEL.SYNCQ | 2 則訊息，每個共用通道的作用中實例 500 位元組 |
| SYSTEM.QSG.UR.RESOLUTION.QUEUE | 1000 則訊息 (2 KB) |

建議的起始結構定義值如下：

```
STRUCTURE NAME(qsgname CSQSYSAPPL)
INITSIZE(20480) /* 20 MB */
SIZE(30720) /* 30 MB */
FULLTHRESHOLD(85)
```

視您使用共用通道及群組回復單元而定，可以調整這些值。

將共用佇列對映至結構

佇列定義的 CFSTRUCT 屬性是用來將佇列對映至結構。

IBM MQ 會將佇列共用群組的名稱新增至 CFSTRUCT 屬性的開頭。對於 CFRM 原則中定義的結構 (名稱為 *qsg-name* SHAREDQ01)，使用此結構的佇列定義如下：

```
DEFINE QLOCAL( myqueue ) QSGDISP(SHARED) CFSTRUCT(SHAREDQ01)
```

規劃共用訊息資料集 (SMDS) 環境

如果您使用具有 SMDS 卸載功能的佇列共用群組，則 IBM MQ 需要連接至一組共用訊息資料集。請利用這個主題來協助瞭解資料集需求，以及儲存 IBM MQ 訊息資料所需的配置。

共用訊息資料集 (由關鍵字 SMDS 說明) 是佇列管理程式用來儲存連結機能結構中儲存之共用訊息的卸載訊息資料的資料集。

註: 定義結構的 SMDS 資料集時，每一個佇列管理程式必須各有一個。

當啟用這種形式的資料卸載時，**CFSTRUCT** 需要一組相關聯的共用訊息資料集，佇列共用群組中每一個佇列管理程式都有一個資料集。共用訊息資料集的群組是使用 **CFSTRUCT** 定義上的 **DSGROUP** 參數定義給 IBM MQ。其他參數可用來提供進一步的選用資訊，例如要使用的緩衝區數目及資料集的擴充屬性。

每一個佇列管理程式都可以寫入它所擁有的資料集，以儲存透過該佇列管理程式所寫入之訊息的共用訊息資料，並且可以讀取群組中的所有資料集

說明與結構相關聯之每一個資料集的狀態及屬性的清單，會在內部作為 **CFSTRUCT** 定義的一部分進行維護，因此每一個佇列管理程式都可以檢查定義，以找出目前可用的資料集。

此資料集資訊可以使用 **DISPLAY CFSTATUS TYPE(SMDS)** 指令顯示現行狀態和可用性，並使用 **DISPLAY SMDS** 指令顯示與指定 **CFSTRUCT** 相關聯的資料集的參數設定。

擁有端佇列管理程式名稱 (通常使用 **SMDS** 關鍵字指定) 與 **CFSTRUCT** 結構名稱的組合可有效地識別個別共用訊息 dta 集。

本節說明下列主題：

- [DSGROUP 參數](#)
- [DSBLOCK 參數](#)
- [共用訊息資料集性質](#)
- [共用訊息資料集空間管理](#)
- [存取共用訊息資料集](#)
- [建立共用訊息資料集](#)
- [共用訊息資料集效能及容量考量](#)
- [啟動共用訊息資料集](#)

如需這些參數的詳細資料，請參閱 [DEFINE CFSTRUCT](#)。

如需管理共用訊息資料集的相關資訊，請參閱 [管理共用訊息資料集](#)，以取得進一步詳細資料。

DSGROUP 參數

CFSTRUCT 定義上的 **DSGROUP** 參數識別要儲存該結構之大型訊息的資料集群組。其他參數可用來指定要用於空間配置目的的邏輯區塊大小，以及緩衝池大小及自動資料集擴充選項的值。

必須先設定 **DSGROUP** 參數，才能啟用卸載至資料集。

- 如果正在 **CFLEVEL (5)** 中定義新的 **CFSTRUCT**，且已指定或假設選項 **OFFLOAD(SMDS)**，則必須在相同的指令上指定 **DSGROUP** 參數。
- 如果正在變更現有的 **CFSTRUCT** 以將 **CFLEVEL** 增加至 **CFLEVEL (5)**，且已指定或假設 **OFFLOAD(SMDS)** 選項，則必須在相同指令上指定 **DSGROUP** 參數 (如果尚未設定的話)。

DSBLOCK 參數

每一個資料集內的空間會配置給佇列作為使用 **CFSTRUCT** 定義上 **DSBLOCK** 參數指定的固定大小 (通常是 256 KB) 的邏輯區塊，然後配置給個別訊息作為每一個邏輯區塊內 4 KB (對應於實體區塊大小及控制間隔大小) 的頁面範圍。邏輯區塊大小也會決定單一 I/O 作業中可讀取或寫入的訊息資料量上限，與 **SMDS** 緩衝池的緩衝區大小相同。

DSBLOCK 參數的較大值可以減少個別 I/O 作業的數目，以增進大型訊息的效能。不過，較小的值會減少每一個作用中要求所需的緩衝儲存體數量。**DSBLOCK** 參數的預設值為 256 KB，在這些需求之間提供合理的平衡，因此通常不需要指定此參數。

共用訊息資料集性質

共用訊息資料集定義為 VSAM 線性資料集 (LDS)。每一個卸載訊息都儲存在資料集中的一個以上區塊中。儲存的資料由連結機能項目中的資訊直接定址，例如延伸形式的虛擬儲存體。資料集本身沒有儲存個別索引或類似的控制資訊。

直接定址方法表示對於符合一個區塊的訊息，只需要單一 I/O 作業即可讀取或寫入區塊。當訊息跨越多個區塊時，只要有足夠的緩衝區可用，每個區塊的 I/O 作業可以完全重疊，以將經歷時間縮至最小。

共用訊息資料集也包含少量一般控制資訊，包括第一頁中的標頭 (包括回復及重新啟動狀態資訊)，以及空間對映檢查點區域 (用來在佇列管理程式正常終止時儲存可用區塊空間對映)。

共用訊息資料集空間管理

作為容量、效能及作業考量的背景資訊，瞭解共用訊息資料集中的空間如何由佇列管理程式管理的概念可能很有用。

每一個共用訊息資料集中的可用空間由其擁有的佇列管理程式使用空間對映來追蹤，空間對映指出每一個邏輯區塊內使用中的頁數。空間對映會在資料集開放時維護在主儲存體中，並在資料集正常關閉時儲存在資料集中。(在回復狀況下，會透過掃描連結機能結構中的訊息來自動重建空間對映，以找出目前使用中的資料集頁面)。

當寫入含有卸載訊息資料的共用訊息時，佇列管理程式會為每一個訊息區塊配置一個頁面範圍。如果指定佇列有部分使用的現行邏輯區塊，則佇列管理程式會從該區塊中的下一個可用頁面開始配置空間，否則會配置新的邏輯區塊。如果整個訊息不符合現行邏輯區塊，則佇列管理程式會在邏輯區塊結尾分割訊息資料，並為下一個訊息區塊配置新的邏輯區塊。這會重複執行，直到為整個訊息配置空間為止。最後一個邏輯區塊中的任何未用空間都會儲存為佇列的新現行邏輯區塊。正常關閉資料集時，現行邏輯區塊中任何未使用的頁面都會在儲存之前傳回空間對映。

當已讀取具有已卸載訊息資料的共用訊息且已備妥可刪除時，佇列管理程式會處理刪除要求，方法是將訊息的連結機能項目傳送至擁有端佇列管理程式 (可能是相同的佇列管理程式) 所監視的清除清單。當項目到達此清單時，擁有端佇列管理程式會讀取並刪除項目，並將釋放的頁面範圍傳回空間對映。當邏輯區塊中所有已使用的頁面都已釋放時，區塊會變成可供重複使用。

存取共用訊息資料集

每一個共用訊息資料集都必須位於共用直接存取儲存體上，該儲存體可供佇列共用群組中的所有佇列管理程式存取。

在正常執行期間，每一個佇列管理程式會開啟其自己的共用訊息資料集來進行讀寫存取，並開啟其他佇列管理程式的任何作用中共用訊息資料集來進行唯讀存取，以便它可以讀取那些佇列管理程式所儲存的訊息。這表示每一個佇列管理程式使用者 ID 至少需要其本身共用訊息資料集的 UPDATE 存取權，以及結構的所有其他共用訊息資料集的 READ 存取權。

如果需要使用 **RECOVER CFSTRUCT** 來回復共用訊息資料集，則可以從佇列共用群組中的任何佇列管理程式執行回復處理程序。可用來執行回復處理程序的佇列管理程式，需要對可能需要回復的所有資料集具有 UPDATE 存取權

建立共用訊息資料集

在建立或變更對應的 **CFSTRUCT** 定義之前，通常應該先建立每一個共用訊息資料集，以便能夠使用這種形式的訊息卸載，因為 **CFSTRUCT** 定義變更通常會立即生效，只要佇列管理程式嘗試存取已指派給該結構的共用佇列，就需要該資料集。SCSQPROC (CSQ4SMDS) 中提供配置及預先格式化共用訊息資料集的範例工作。必須自訂並執行工作，以針對使用 **CFSTRUCT** 搭配 **OFFLOAD (SMDS)** 的每一個佇列管理程式配置共用訊息資料集。

如果佇列管理程式發現已啟用卸載支援，並嘗試開啟其共用訊息資料集，但尚未建立它，則會將共用訊息資料集標示為無法使用。然後，在建立資料集並通知佇列管理程式重試 (例如使用 **START SMDSCONN** 指令) 之前，佇列管理程式將無法儲存任何大型訊息。

使用 Access Method Services **DEFINE CLUSTER** 指令將共用訊息資料集建立為 VSAM 線性資料集。定義必須指定「**SHAREOPTIONS(2 3)**」，以容許一個佇列管理程式同時開啟它來進行寫入存取，並容許任意數目的佇列管理程式同時讀取它。必須使用預設控制間隔大小 4 KB。如果資料集可能需要擴充超過 4 GB，則必須使用具有 VSAM 延伸定址能力屬性的 SMS 資料類別來定義它。共用訊息資料集可以位於延伸位址磁區 (EAV) 的延伸定址空間 (EAS) 部分。

在起始使用之前，每一個共用訊息資料集可以是空的或預先格式化為二進位零 (使用 **CSQJUFMT** 或類似公用程式，例如範例工作 **SCSQPROC (CSQ4SMDS)**)。如果它在開啟時是空的或僅部分格式化，則佇列管理程式會自動將剩餘空間格式化為二進位零。

共用訊息資料集效能及容量考量

每一個共用訊息資料集都用來儲存共用訊息的卸載資料，這些共用訊息是由擁有端佇列管理程式從相同系統內的區域寫入相關聯的 **CFSTRUCT**。每一個訊息的儲存資料包括描述子 (目前大約 350 個位元組)、訊息標頭及訊息內文。每一個卸載訊息都儲存在資料集中的一個以上頁面 (大小為 4 KB 的實體區塊) 中。

因此，將整體訊息大小 (包括描述子) 四捨五入至 4 KB 的下一個倍數，然後乘以訊息數目，即可預估給定卸載訊息數所需的資料集空間。

至於頁集，當共用訊息資料集幾乎已滿時，可以選擇性地自動展開。可以使用 **CFSTRUCT** 定義上的 **DSEXPAND** 參數來設定此自動擴充的預設行為。對於每一個佇列管理程式，可以在 **ALTER SMDS** 指令上使用 **DSEXPAND** 參數來置換此設定。當資料集達到 90% 已滿且需要更多空間時，會觸發自動擴充。如果容許擴充，但 **VSAM** 拒絕擴充嘗試，因為在定義資料集時未指定次要空間配置，則會使用資料集現行大小的 20% 次要配置來重試擴充。

如果使用延伸定址能力屬性來定義共用訊息資料集，則大小上限僅受限於最多 16 TB 或 59 個磁區的 **VSAM** 考量。這明顯大於本端頁集的 64 GB 大小上限。

啟動共用訊息資料集

當佇列管理程式順利連接至應用程式連結機能結構時，它會檢查該結構定義是否使用相關聯的 **DSGROUP** 參數指定卸載。若是如此，佇列管理程式會配置並開啟它自己的共用訊息資料集來進行寫入存取，然後開啟它來讀取其他佇列管理程式所擁有的任何現有共用訊息資料集。

第一次開啟共用訊息資料集 (在佇列共用群組內記錄為作用中之前) 時，第一頁將不會包含有效的標頭。佇列管理程式會填入標頭資訊，以識別佇列共用群組、結構名稱及擁有的佇列管理程式。

在完成標頭之後，佇列管理程式會將新的共用訊息資料集登錄為作用中，並播送事件以通知任何其他作用中佇列管理程式有關新的資料集。

每次佇列管理程式開啟共用訊息資料集時，都會驗證標頭資訊，以確保仍在使用的資料集，且尚未損壞。

規劃 Db2 環境

如果您使用佇列共用群組，則 **IBM MQ** 需要連接至屬於資料共用群組成員的 Db2 子系統。請利用這個主題來協助瞭解用來保留 **IBM MQ** 資料的 Db2 需求。

IBM MQ 需要知道要連接的資料共用群組名稱，以及要連接的 Db2 子系統 (或 Db2 群組) 名稱，才能呼叫到此資料共用群組。這些名稱指定在 **CSQ6SYSP** 系統參數巨集的 **QSGDATA** 參數中 (如 [使用 CSQ6SYSP](#) 中所述)。

IBM MQ 提供用於定義 Db2 表格空間、表格及索引的範例工作。提供兩組範例工作：

- 一個適用於與舊版 **IBM MQ** 的相容性
- 一個與 Db2 V12 以及更新版本搭配使用，利用「通用表格空間 (UTS)」

從 Db2 V11 開始，舊版 **IBM MQ** 所使用的各種類型表格空間已標示為已淘汰。可能的話，在設定新的資料共用群組時，建議您選擇一組利用 Db2 UTS 的工作。

依預設，Db2 會使用執行工作之人員的使用者 ID 作為 Db2 資源的擁有者。如果刪除此使用者 ID，則會刪除與其相關聯的資源，因此會刪除表格。請考慮使用群組 ID 來擁有表格，而不是個別使用者 ID。作法是將 **GROUP=groupname** 新增至 **JOB** 卡，並在任何 **SQL** 陳述式之前指定 **SET CURRENT SQLID='groupname'**。

IBM MQ 使用 Db2 的「RRS 連接」機能。這表示您可以指定要連接的 Db2 群組名稱。連接至 Db2 群組連接名稱 (而非特定的 Db2 子系統) 的優點是 **IBM MQ** 可以連接 (或重新連接) 屬於該群組成員的 z/OS 映像檔上任何可用的 Db2 子系統。在您要執行佇列共用 **IBM MQ** 子系統的每一個 z/OS 映像檔上，必須有一個 Db2 子系統是作用中資料共用群組的成員，且 **RRS** 必須是作用中。

Db2 儲存體

對於大部分安裝，在 3390 裝置上需要的 Db2 儲存體量大約是 20 或 30 個磁柱。不過，如果您要計算儲存體需求，下表提供一些資訊來協助您判斷 Db2 需要 IBM MQ 資料的儲存體量。此表格說明每一個 Db2 列的長度，以及在相關 Db2 表格中新增或刪除每一列的時間。將此資訊與計算 *Db2 for z/OS* 安裝手冊中 Db2 表格及其索引的空間需求相關資訊一起使用。

| Db2 表格名稱 | 列的長度 | 在下列情況下，會新增一列: | 在下列情況下，會刪除一列: |
|-----------------------|---------------|---|--|
| CSQ.ADMIN_B_QSG | 252 位元組 | 使用 CSQ5PQSG 公用程式的 ADD QSG 函數，將佇列共用群組新增至表格。 | 使用 CSQ5PQSG 公用程式的 REMOVE QSG 功能，從表格中移除佇列共用群組。(當刪除佇列共用群組記錄時，會自動從所有其他 Db2 表格中刪除與此佇列共用群組相關的所有列。) |
| CSQ.ADMIN_B_QMGR | 最多 3828 個位元組 | 佇列管理程式會使用 CSQ5PQSG 公用程式的 ADD QMGR 函數新增至表格。 | 使用 CSQ5PQSG 公用程式的 REMOVE QMGR 功能，從表格中移除佇列管理程式。 |
| CSQ.ADMIN_B_STRUCTURE | 1454 位元組 | 第一個本端佇列定義 (指定 QSGDISP (SHARED) 屬性) 會定義佇列共用群組內先前不明結構的名稱。 | 會刪除最後一個本端佇列定義 (指定 QSGDISP (SHARED) 屬性來命名佇列共用群組內的結構)。 |
| CSQ.ADMIN_B_SCST | 342 位元組 | 已啟動共用通道。 | 共用通道會變成非作用中。 |
| CSQ.ADMIN_B_SSKT | 254 位元組 | 已啟動具有 NPMSPEED (NORMAL) 屬性的共用通道。 | 具有 NPMSPEED (NORMAL) 屬性的共用通道會變成非作用中。 |
| CSQ.ADMIN_B_STRBACKUP | 514 位元組 | 即會將新列新增至 CSQ.ADMIN_B_STRUCTURE 表格。在執行 BACKUP CFSTRUCT 指令之前，每一個項目都是虛擬項目，這會改寫虛擬項目。 | 即會從 CSQ.ADMIN_B_STRUCTURE 表格。 |
| CSQ.OBJ_B_AUTHINFO | 3400 位元組 | 已定義具有 QSGDISP (GROUP) 的鑑別資訊物件。 | 已刪除具有 QSGDISP (GROUP) 的鑑別資訊物件。 |
| CSQ.OBJ_B_QUEUE | 最多 3707 位元組 | <ul style="list-style-type: none"> 已定義具有 QSGDISP (GROUP) 屬性的佇列。 已定義具有 QSGDISP (SHARED) 屬性的佇列。 即會開啟具有 DEFTYPE (SHAREDYN) 屬性的模型佇列。 | <ul style="list-style-type: none"> 刪除具有 QSGDISP (GROUP) 屬性的佇列。 已刪除具有 QSGDISP (SHARED) 屬性的佇列。 使用 DELETE 選項關閉具有 DEFTYPE (SHAREDYN) 屬性的動態佇列。 |
| CSQ.OBJ_B_NAMELIST | 最多 15127 個位元組 | 已定義具有 QSGDISP (GROUP) 屬性的名單。 | 已刪除具有 QSGDISP (GROUP) 屬性的名單。 |

| 表 24: 規劃 Db2 儲存體需求 (繼續) | | | |
|--|---------------|--|--|
| Db2 表格名稱 | 列的長度 | 在下列情況下，會新增一列: | 在下列情況下，會刪除一列: |
| CSQ.OBJ_B_CHANNEL | 最多 14127 個位元組 | 已定義具有 QSGDISP (GROUP) 屬性的通道。 | 刪除具有 QSGDISP (GROUP) 屬性的通道。 |
| CSQ.OBJ_B_STGCLASS | 最多 2865 個位元組 | 已定義具有 QSGDISP (GROUP) 屬性的儲存類別。 | 刪除具有 QSGDISP (GROUP) 屬性類別的儲存類別。 |
| CSQ.OBJ_B_PROCESS | 最多 3347 位元組 | 已定義具有 QSGDISP (GROUP) 屬性的處理程序。 | 刪除具有 QSGDISP (GROUP) 屬性的處理程序。 |
| CSQ.OBJ_B_TOPIC | 最多 14520 個位元組 | 已定義具有 QSGDISP (GROUP) 屬性的主題物件。 | 刪除具有 QSGDISP (GROUP) 屬性的主題物件。 |
| CSQ.EXTEND_B_QMGR | 小於 430 位元組 | 佇列管理程式會使用 CSQ5PQSG 公用程式的 ADD QMGR 函數新增至表格。 | 使用 CSQ5PQSG 公用程式的 REMOVE QMGR 功能，從表格中移除佇列管理程式。 |
| CSQ.ADMIN_B_MESSAGES | 87 位元組 | 若為大型訊息 PUT (每個 BLOB 1 個)。 | 若為大型訊息 GET (每個 BLOB 1 個)。 |
| CSQ.ADMIN_MSGS_BAUX1 CSQ.ADMIN_MSGS_BAUX2 CSQ.ADMIN_MSGS_BAUX3 CSQ.ADMIN_MSGS_BAUX4 | | 這 4 個表格包含大型訊息的訊息有效負載，新增至訊息每一個 BLOB 的這 4 個表格中的其中一個表格。BLOBS 的長度最多為 511 KB，因此如果訊息大小 > 711 KB，則此訊息會有多個 BLOB。 | |

使用大大於 63 KB 的大量共用佇列訊息，可能會對 IBM MQ 系統產生重大效能影響。如需相關資訊，請參閱 SupportPac MP16, Capacity Planning and Tuning for IBM MQ for z/OS, 網址為: [Business Integration- IBM MQ SupportPacs](#)。

規劃記載環境

請利用這個主題來規劃 IBM MQ 所使用的日誌和日誌保存檔的數目、大小和位置。

日誌用於:

- 寫入持續訊息的相關回復資訊
- 使用持續訊息記錄工作單元的相關資訊
- 記錄物件變更的相關資訊，例如定義佇列
- 備份 CF 結構

以及其他內部資訊。

使用系統參數巨集來建立 IBM MQ 記載環境，以指定選項，例如: 是否具有單一或雙重作用中日誌、保存日誌磁區要使用哪些媒體，以及要具有多少日誌緩衝區。

作業 14: 建立引導及日誌資料集 及 作業 17: 自訂系統參數模組中說明了這些巨集。

註: 如果您使用佇列共用群組，請確保使用 SHAREOPTIONS (2 3) 定義引導及日誌資料集。

本節包含下列主題的相關資訊:

請利用這個主題來決定日誌資料集最適當的配置。

本主題包含協助您回答下列問題的資訊：

- [您的安裝應該使用單一或雙重記載嗎？](#)
- [您需要多少個作用中日誌資料集？](#)
- [第 143 頁的『作用中日誌應該有多大？』](#)
- [作用中日誌放置](#)

您的安裝應該使用單一或雙重記載嗎？

一般而言，您應該對正式作業使用雙重記載，以將遺失資料的風險降到最低。如果您想要測試系統反映正式作業，則兩者都應該使用雙重記載，否則測試系統可以使用單一記載。

使用單一記載資料會寫入一組日誌資料集。使用雙重記載資料會寫入兩組日誌資料集，因此如果一個日誌資料集發生問題 (例如意外刪除資料集)，則可以使用另一組日誌中的對等資料集來回復資料。

使用雙重記載時，您需要的 DASD 數量是單一記載的兩倍。

如果您使用雙重記載，則也請使用雙重 BSDS 及雙重保存，以確保充分供應資料回復。

雙重作用中記載會增加少量效能成本。



小心：一律使用雙重記載和雙重 BSDS，而不是雙重寫入 DASD (鏡映)。如果意外刪除鏡映資料集，則兩個副本都會遺失。

如果您使用持續訊息，則單一記載可以將容量上限增加 10-30%，也可以改善回應時間。

單一記載使用 2-310 個作用中日誌資料集，而雙重記載使用 4-62 個作用中日誌資料集 0，以提供相同數目的作用中日誌。因此，單一記載會減少所記載的資料量，如果您的安裝受到 I/O 限制，這可能很重要。

您需要多少作用中日誌資料集？

日誌數目取決於佇列管理程式的活動。對於具有低傳輸量的測試系統，三個作用中日誌資料集可能適合。對於高傳輸量正式作業系統，您可能想要可用的日誌數目上限，因此，如果卸載日誌有問題，您有更多時間來解決問題。

您必須至少有三個作用中日誌資料集，但最好定義更多資料集。例如，如果填滿日誌所花費的時間可能接近在尖峰負載期間保存日誌所花費的時間，請定義更多日誌。

您也應該定義更多日誌，以抵銷日誌保存中可能的延遲。如果您在磁帶上使用保存日誌，請容許裝載磁帶所需的時間。

請考量有足夠的作用中日誌空間來保留一天的資料，以防系統因為缺少 DASD 或無法寫入磁帶而無法保存。

可以動態定義新的作用中日誌資料集，以將保存延遲或問題的影響降到最低。新的資料集可以快速回到線上，使用 [DEFINE LOG](#) 指令來避免佇列管理程式因為作用中日誌中缺少空間而「停滯」。

如果您想要定義超過 31 個作用中日誌資料集，則必須套用 APAR PI46853，並將記載環境配置為使用第 2 版格式 BSDS。使用第 2 版格式 BSDS 之後，每一個日誌副本環最多可以定義 310 個作用中日誌資料集。如需如何轉換為第 2 版格式 BSDS 的相關資訊，請參閱 [第 147 頁的『規劃增加可定址日誌範圍上限』](#)。

您可以執行列印日誌對映公用程式 (CSQJU004)，或從佇列管理程式起始設定期間發出的 CSQJ034I 訊息，來判斷佇列管理程式是否使用第 2 版或更新版本的 BSDS。在 CSQJ034I 訊息中，日誌 RBA 範圍的結尾 FFFFFFFFFFFFFFFF 指出正在使用第 2 版或更高版本的格式 BSDS。

當佇列管理程式使用第 2 版或更新版本的格式 BSDS 時，可以使用 [DEFINE LOG](#) 指令將超過 31 個作用中日誌資料集動態新增至日誌副本環。

作用中日誌應該有多大?

在 IBM MQ 8.0 上，保存至磁碟時支援的作用中日誌大小上限為 4 GB。在舊版產品中，保存至磁碟時支援的作用中日誌大小上限為 3 GB。

保存至磁帶時，作用中日誌大小上限為 4 GB。

您應該為正式作業和測試系統建立大小至少為 1 GB 的作用中日誌。

重要: 配置資料集時需要小心，因為 IDCAMS 會四捨五入您配置的大小。

若要配置 3 GB 日誌，請指定下列其中一個選項：

- 磁柱 (4369)
- MB (3071)
- TRACKS (65535)
- 記錄 (786420)

其中任何一個都會配置 2.99995 GB。

若要配置 4GB 日誌，請指定下列其中一個選項：

- 磁柱 (5825)
- MB (4095)
- TRACKS (87375)
- 記錄 (1048500)

其中任何一個都會配置 3.9997 GB。

使用分段資料集時，如果資料集分散在多個磁區中，則會在用於分段的每一個 DASD 磁區上配置指定的大小值。因此，如果您想要使用 4 GB 日誌和四個磁區進行分段，則應該指定：

- CYLinders (1456)
- MB (1023)

設定這些屬性會配置 $4 * 1456 = 5824$ 磁柱或 $4 * 1023 = 4092$ MB。

註: 使用延伸格式資料集時支援分段。這通常是由儲存體管理程式所設定。

如需執行此程序的相關資訊，請參閱 [增加作用中日誌的大小](#)。

作用中日誌放置

基於效能原因，您應該考慮分段作用中日誌資料集。I/O 會分散在多個磁區中，並減少 I/O 回應時間，導致更高的傳輸量。如需使用分段時配置作用中日誌大小的相關資訊，請參閱之前的文字。

您應該使用來自 RMF 或類似產品的報告來檢閱 I/O 統計資料。每月 (或更頻繁地) 檢閱 IBM MQ 資料集的這些統計資料，以確保不會因資料集的位置而延遲。

在某些狀況下，可能會有許多 IBM MQ 頁集 I/O，如果它們位於相同的 DASD 上，這可能會影響 IBM MQ 日誌效能。

如果您使用雙重記載，請確保每一組作用中及保存日誌保持分開。例如，在個別 DASD 子系統或不同裝置上配置它們。

如果其中一個磁區已毀損或毀損，則可減少兩者遺失的風險。如果日誌的兩個副本都遺失，則資料遺失的機率會很高。

當您建立新的作用中日誌資料時，您應該使用 CSQJUFMT 預先格式化它。如果日誌未預先格式化，則佇列管理程式會在第一次使用日誌時將其格式化，這會影響效能。

如果 DASD 較舊，且有大型旋轉磁碟，您必須小心使用哪些磁區來取得最佳效能。

有了現代 DASD，資料會分散在許多 PC 大小的磁碟上，您不需要太擔心使用哪些磁區。

您的儲存體管理程式應該檢查企業 DASD，以檢閱並解決任何效能問題。為了可用性，您可能想要在一個 DASD 子系統上使用一組日誌，並在不同的 DASD 子系統上使用雙重日誌。

規劃日誌保存儲存體

請利用這個主題來瞭解維護保存日誌資料集的不同方式。

您可以將保存日誌資料集放置在標準標籤磁帶或 DASD 上，並且可以透過資料機能階層式儲存體管理程式 (DFHSM) 來管理它們。保存日誌資料集中的每一筆 z/OS 邏輯記錄都是來自作用中日誌資料集的 VSAM 控制間隔。區塊大小是 4 KB 的倍數。

保存日誌資料集是以 IBM MQ 所選擇的名稱來動態配置。系統參數模組中指定這類配置所需的資料集名稱字首、區塊大小、單位名稱及 DASD 大小。您也可以選擇在安裝時，讓 IBM MQ 將日期和時間新增至保存日誌資料集名稱。

無法使用 IBM MQ 來指定新的保存日誌的特定磁區，但您可以使用「儲存體管理」常式來管理此作業。如果發生配置錯誤，則會延遲卸載，直到下一次觸發卸載為止。

如果您在安裝時指定雙重保存日誌，則從作用中日誌擷取的每一個日誌控制間隔都會寫入兩個保存日誌資料集。保存日誌資料集配對中包含的日誌記錄相同，但多容體資料集的磁區結尾點不同步。

您的保存日誌應該位於磁帶或 DASD 上嗎？

在決定要將磁帶或 DASD 用於保存日誌時，您應該考慮一些因素：

- 在決定磁帶或磁碟之前，請先檢閱您的作業程序。例如，如果您選擇保存至磁帶，則在需要時必須有足夠的磁帶機。發生災難之後，所有子系統都可能想要磁帶機，而且您可能沒有預期的可用磁帶機數目。
- 在回復期間，一旦裝載磁帶，即可使用磁帶上的保存日誌。如果已使用 DASD 保存檔，且使用階層式儲存體管理程式 (HSM) 將資料集移轉至磁帶，則 HSM 會延遲將每一個資料集恢復至磁碟。您可以在使用保存日誌之前恢復資料集。不過，並非一律可以預測需要它們的正確順序。
- 在 DASD 上使用保存日誌時，如果需要許多日誌 (從備份還原之後回復頁集時可能如此)，您可能需要大量 DASD 來保留所有保存日誌。
- 在低用量系統或測試系統中，在 DASD 上保存日誌可能會更方便，而不需要磁帶裝載。
- 發出 RECOVER CFSTRUCT 指令及取消持續性工作單元都會導致反向讀取日誌。具有硬體壓縮的磁帶機在往回讀取的作業上執行不正確。在 DASD 上規劃足夠的日誌資料，以避免從磁帶往回讀取。

保存至 DASD 可提供更快的可回復性，但比保存至磁帶更貴。如果您使用雙重記載，則可以指定將保存日誌的主要副本移至 DASD，並將次要副本移至磁帶。這會增加回復速度，而不會使用太多 DASD，而且您可以使用磁帶作為備份。

保存至磁帶

如果您選擇保存至磁帶機，則 IBM MQ 最多可以延伸至 20 個磁區。

如果您考慮變更作用中日誌資料集的大小，使該集適合一個磁帶磁區，請注意，BSDS 的副本與作用中日誌資料集的副本位於相同的磁帶磁區上。向下調整作用中日誌資料集的大小，以偏移磁帶磁區上 BSDS 所需的空間。

如果您在磁帶上使用雙重保存日誌，則通常會在本端保留一個副本，而另一個副本則會在離站保留以用於災難回復。

保存至 DASD 磁區

IBM MQ 需要您編目在非磁帶機 (DASD) 上配置的所有保存日誌資料集。如果您選擇保存至 DASD，則 CSQ6ARVP 巨集的 CATALOG 參數必須是 YES。如果此參數為 NO，且您決定將保存日誌資料集放在 DASD 上，則每次配置保存日誌資料集時，都會收到 CSQJ072E 訊息，但 IBM MQ 仍會將資料集編目。

如果保存日誌資料集保留在 DASD 上，則保存日誌資料集可以延伸至另一個磁區；支援多容體。

如果您選擇使用 DASD，請確定主要空間配置 (數量和區塊大小) 夠大，足以包含來自作用中日誌資料集的資料，或來自對應 BSDS 的資料 (兩者中的較大者)。

在卸載處理程序期間，這會將不想要的 z/OS X'B37'或 X'E37' 異常終止碼的可能性降到最低。主要空間配置是使用 [CSQ6ARVP](#) 巨集的 [PRIQTY](#) (主要數量) 參數來設定。

從 IBM MQ 8.0 開始，保存日誌資料集可以存在於大型或延伸格式的循序資料集上。SMS ACS 常式現在可以使用 [DSNTYPE \(LARGE\)](#) 或 [DSNTYPE \(EXT\)](#)。這些在 8.0 版之前不受支援。

IBM MQ 支援將保存日誌配置成延伸格式資料集。使用延伸格式時，保存日誌大小上限會從 65535 個磁軌增加至 4GB 的作用中日誌大小上限。保存日誌可以在延伸位址磁區 (EAV) 的延伸定址空間 (EAS) 中配置。

在必要軟硬體層次可供使用的情況下，使用 zEDC 將保存日誌配置給 [COMPACTION](#) 所定義的資料類別，可能會減少保留保存日誌所需的磁碟儲存體。如需相關資訊，請參閱 [IBM MQ for z/OS: 使用 IBM zEnterprise Data Compression \(zEDC\)](#) 來減少儲存體佔有率。

如需軟硬體層次以及 RACF 設定檔變更範例的詳細資料，請參閱 [使用 zEnterprise 資料壓縮 \(zEDC\) 加強功能](#)。

z/OS 資料集加密功能可套用至在 IBM MQ 8.0 或更新版本執行之佇列管理程式的保存日誌。這些保存日誌必須透過「自動選取類別 (ACS)」常式，配置給以 [EXTENDED](#) 屬性定義的資料類別，以及確保資料以 AES 加密的資料集金鑰標籤。

使用 SMS 與保存日誌資料集

如果您已安裝 MVS/DFP 儲存體管理子系統 (DFSMS)，則可以為保存日誌資料集撰寫「自動選取類別 (ACS)」使用者結束程式過濾器，以協助您針對 SMS 環境進行轉換。

例如，這類過濾器可以將您的輸出遞送至 DFSMS 可以管理的 DASD 資料集。如果您以此方式使用 ACS 過濾器，則必須特別小心。因為 SMS 需要編目 DASD 資料集，所以您必須確定 [CSQ6ARVP](#) 巨集的 [CATALOG DATA](#) 欄位包含 YES。如果沒有，則會傳回訊息 [CSQJ072E](#)；不過，資料集仍由 IBM MQ 編目。

如需 ACS 過濾器的相關資訊，請參閱 [DFSMSshm 動態配置的資料集](#)。

z/OS 變更保存日誌的儲存媒體

變更保存日誌所用儲存媒體的程序。

關於這項作業

此作業說明如何變更用於保存日誌的儲存媒體，例如從保存移至磁帶，再移至保存至 DASD。

您可以選擇如何進行變更：

1. 僅使用 [CSQ6ARVP](#) 巨集進行變更，以便從下次佇列管理程式重新啟動時開始套用這些變更。
2. 使用 [CSQ6ARVP](#) 巨集並使用 [SET ARCHIVE](#) 指令動態進行變更。這表示從下次佇列管理程式保存日誌檔開始，就會套用這些變更，並在佇列管理程式重新啟動之後持續保存。

程序

1. 變更使保存日誌儲存在 DASD 上而非磁帶上：
 - a) 閱讀 [第 144 頁的『規劃日誌保存儲存體』](#) 並檢閱 [CSQ6ARVP](#) 參數。
 - b) 在 [CSQ6ARVP](#) 中變更下列參數
 - 更新 [UNIT](#) 及 [UNIT2](#) 參數 (必要的話)。
 - 更新 [BLKSIZE](#) 參數，因為 DASD 的最佳設定不同於磁帶。
 - 將 [PRIQTY](#) 及 [SECQTY](#) 參數設為足夠大，以保留最大的作用中日誌或 BSDS。
 - 將 [CATALOG](#) 參數設為 YES。
 - 確認 [ALCUNIT](#) 設定是您想要的。您應該使用 [BLK](#)，因為它與裝置類型無關。
 - 如果 [ARCWTOR](#) 參數尚未設為 NO，請將它設為 NO。
2. 變更使保存日誌儲存在磁帶而非 DASD 上：

- a) 閱讀第 144 頁的『規劃日誌保存儲存體』，並檢閱 CSQ6ARVP 參數。
- b) 在 CSQ6ARVP: 中變更下列參數:
 - 更新 UNIT 及 UNIT2 參數 (必要的話)。
 - 更新 BLKSIZE 參數，因為磁帶的最佳設定不同於 DASD。
 - 確認 ALCUNIT 設定是您想要的。您應該使用 BLK，因為它與裝置類型無關。
 - 檢閱 ARCWTOR 參數的設定。

我需要保留保存日誌的時間

請使用本節中的資訊來協助您規劃備份策略。

您可以使用 USING CSQ6ARVP 中的 ARCRETN 參數或 SET SYSTEM 指令，指定保存日誌的保留時間 (以天為單位)。在此期間之後，z/OS 可以刪除資料集。

您可以手動刪除不再需要的保存日誌資料集。

- 佇列管理程式可能需要保存日誌才能進行回復。

佇列管理程式只能在 BSDS 中保留最新的 1000 個保存檔，當保存日誌不在 BSDS 中時，它們無法用於回復，且僅用於審核、分析或重播類型。
- 您可能想要保留保存日誌，以便可以從日誌擷取資訊。例如，從日誌擷取訊息，並檢閱放置或取得訊息的使用者 ID。

BSDS 包含日誌及其他回復資訊的相關資訊。此資料集是固定大小。當保存日誌數目達到 CSQ6LOGP 中的 MAXARCH 值時，或當 BSDS 填滿時，會改寫最舊的保存日誌資訊。

有一些公用程式可從 BSDS 中移除保存日誌項目，但一般而言，BSDS 會覆蓋最舊的保存日誌記錄。

何時需要保存日誌

您需要定期備份頁面集。備份頻率會決定在遺失頁集時需要哪些保存日誌。

您需要定期備份 CF 結構。備份頻率會決定在 CF 結構中遺失資料時需要哪些保存日誌。

回復可能需要保存日誌。下列資訊說明何時可能需要保存日誌，其中有不同 IBM MQ 資源的問題。

遺失頁集 0

您必須從備份回復系統，並重新啟動佇列管理程式。

您需要取得備份時的日誌，以及最多三個作用中日誌。

遺失任何其他頁集

您必須從備份回復系統，並重新啟動佇列管理程式。

您需要取得備份時的日誌，以及最多三個作用中日誌。

所有 LPARS 都失去與結構的連線功能，或該結構無法使用

使用 RECOVER CFSTRUCT 指令讀取日誌上的前次 CF 備份。

如果您經常執行 CF 備份，則資料應該在作用中日誌中。

您應該不需要保存日誌。

管理結構重建

如果您需要重建管理結構，則會從每一個佇列管理程式的日誌最後一個檢查點讀取資訊。

如果佇列管理程式不在作用中，則另一個佇列管理程式會讀取日誌。

您應該不需要保存日誌。

遺失 SMDS 資料集

如果您遺失 SMDS 資料集，或資料集毀損，則資料集會變成無法使用，且其狀態會設為 FAILED。CF 結構不變。

若要還原 SMDS 資料集，您需要：

1. 重新定義 SMDS 資料集，以及
2. 失敗，然後回復 CF 結構。

發出 [RECOVER CFSTRUCT](#) 指令兩次達到此處理程序。

第一次發出指令會將結構狀態設為失敗；第二次發出指令會執行實際回復。

註：CF 結構上的所有非持續訊息都會遺失；所有持續訊息都會還原。

您需要從發出 [BACKUP CFSTRUCT](#) 指令開始的日誌，因此這可能需要保存日誌。

如果所有 LPAR 都失去與結構的連線功能，則會重建結構，可能是在替代 CF 中。請注意，您的結構 CFRM PREFLIST 屬性必須包含多個 CF。

註：所有非持續訊息都將遺失；所有持續訊息將由下列項目重建：

1. 讀取前次 CF 備份的日誌
2. 從已使用結構的所有佇列管理程式讀取日誌，以及
3. 自備份以來合併更新項目

您需要自前次備份 (回到取得備份的時間) 以來已存取結構的所有佇列管理程式的日誌，以及在取得備份之佇列管理程式日誌中的結構備份本身。

BSDS

您需要單一或雙重 BSDS 嗎？

如果您使用雙重作用中日誌，則應該使用雙重 BSDS。

BSDS 需要多大？

BSDS 不需要非常大，且一個磁柱的主要和次要應該足夠。

規劃增加可定址日誌範圍上限

您可以將佇列管理程式配置成使用較大的日誌相對位元組位址 (RBA)，以增加可定址日誌範圍上限。

如需 IBM MQ 8.0 的日誌 RBA 變更的概觀，請參閱 [較大的日誌相對位元組位址](#)。

如果佇列管理程式不在佇列共用群組中，您可以將佇列管理程式升級至 IBM MQ 8.0，啟用 8.0 版新功能，並隨時將它轉換為使用 8 個位元組的日誌 RBA 值。一旦佇列管理程式已轉換為使用 8 個位元組的日誌 RBA 值，就無法將它回復為 COMPAT 模式。

對於佇列共用群組中的佇列管理程式，您可以將每一個佇列管理程式依序升級至 IBM MQ 8.0 並啟用 8.0 版新功能。一旦群組中的所有佇列管理程式都處於 IBM MQ 8.0 新功能模式，您可以依次變更每一個佇列管理程式，以使用 8 個位元組的日誌 RBA 值。同時變更所有佇列管理程式並不重要。

當佇列共用群組中的佇列管理程式已轉換為使用 8 個位元組的日誌 RBA 值時，佇列共用群組中的其他佇列管理程式可以使用已轉換佇列管理程式的日誌，即使它們尚未轉換為使用 8 個位元組的日誌 RBA 值也一樣。例如，這對於同層級回復很有用。

註：已轉換為使用 8 個位元組的日誌 RBA 值的佇列管理程式可以讀取具有以 6 個位元組或 8 個位元組的日誌 RBA 值寫入資料的日誌。因此，它的作用中日誌和保存日誌可以用來回復頁集和連結機能 (CF) 結構。

復原變更

無法取消變更。

要多久？

變更需要重新啟動佇列管理程式。停止佇列管理程式，對引導資料集 (BSDS) 執行 CSQJUCNV 公用程式，以建立新的資料集，重新命名這些引導資料集，然後重新啟動佇列管理程式。

這有什麼影響？

- 如果使用 8 個位元組的日誌 RBA，則每次將資料寫入日誌資料集都會有額外的位元組。因此，對於由持續訊息組成的工作量，寫入日誌的資料量會略有增加。
- 寫入頁集或連結機能 (CF) 結構的資料不受影響。

相關資訊

[實作較大的日誌相對位元組位址](#)

規劃 IBM MQ Managed File Transfer

請使用本主題作為指引，說明您需要如何設定系統來執行 IBM MQ Managed File Transfer。

一般配置

有三個一般 IBM MQ Managed File Transfer (MFT) 配置：

1. 單一佇列管理程式與一個以上使用本端連線的代理程式。這可能是用來將資料集的內容放置到 IBM MQ 佇列中。
2. 單一佇列管理程式與一個在分散式機器上使用用戶端連結的 MFT 用戶端。
3. 由通道連接的兩個佇列管理程式，且在每部機器上各有一個以上的代理程式。這些代理程式可以是用戶端或本端連結。

MFT 可以使用多個佇列管理程式：

- 一個以上用於傳送資料的佇列管理程式。
- 一個用於發出要求的指令佇列管理程式。例如，啟動傳送的要求傳送到此佇列管理程式之後，相關聯的指令將遞送至 MFT 代理程式。
- 一個用於管理工作的協調佇列管理程式。

附註：

1. 您可以使用相同的佇列管理程式來傳送資料、指令及協調。
2. 此設定雖然最簡單，但可能不是最有效率的，因為所有工作量都在一個佇列管理程式上。

如果您具有現有的 IBM MQ Managed File Transfer 配置，則您的指令及協調佇列管理程式可能已存在。

如果您沒有現有的 IBM MQ Managed File Transfer 配置，則可以使用一個佇列管理程式來傳送資料、指令及協調。請注意，即使您這麼做，也可以在相同機器上設定多個配置。

如果您使用多個佇列管理程式，則需要設定佇列管理程式之間的通道。您可以使用叢集作業或使用點對點連線來執行此動作。可以記載 IBM MQ Managed File Transfer 狀態及活動，並且可以儲存在 Db2 或 Oracle 資料庫中。

IBM MQ Managed File Transfer 是以 Java 撰寫，其中含有一些 Shell Script 和 JCL 來配置及操作程式。

重要：您必須熟悉 UNIX 系統服務 (USS)，才能配置 IBM MQ Managed File Transfer。例如：

- 檔案目錄結構，名稱如 /u/userID/myfile.txt
- USS 指令，例如：
 - cd (切換目錄)
 - ls (列出)
 - chmod (變更檔案許可權)
 - chown (變更檔案所有權或可存取檔案或目錄的群組)

您需要 USS 中的下列產品才能配置及執行 MFT：

1. Java，例如，在 /java/java71_bit64_GA/J7.1_64/ 目錄中
2. IBM MQ 8.0，例如，在 /mqm/V8R0M0 目錄中

3. 如果您要將 Db2 用於狀態和歷程，則需要安裝 Db2 JDBC 程式庫，例如，在 /db2/db2v10/jdbc/ libs 目錄中。

產品註冊

啟動時 IBM MQ Managed File Transfer 會檢查 sys1.parmlib 連結中的登錄。下列程式碼是如何登錄 MFT 的範例：

```
PRODUCT OWNER('IBM CORP')
NAME('WS MQ FILE TRANS')
ID(5655-MFT)
VERSION(*) RELEASE(*) MOD(*)
FEATURENAME('WS MQ FILE TRANS')
STATE(ENABLED)
```

磁碟空間

您將需要 100 MB 的 DASD (PDSE)，以及至少 50 MB 的 DASD (USS)。如果您使用追蹤來診斷問題，則 USS 需要額外的磁碟空間，例如 50 MB。

安全

您需要識別哪些使用者 ID 將用於 MFT 配置及 MFT 作業。

您需要識別您傳送的檔案或佇列，以及將向 MFT 提交傳送要求的使用者 ID。

當您自訂代理程式及日誌程式時，您可以指定容許執行 MFT 服務或執行 MFT 管理的使用者群組。

在開始自訂 MFT 之前，您應該先設定此群組。由於 MFT 使用 IBM MQ 佇列，如果您在佇列管理程式中已啟用安全，則 MFT 需要下列資源的存取權

表 25:

| 姓名 | 需要存取權 |
|---------------------------------------|-------|
| QUEUE.SYSTEM.FTE.EVENT.agent_name | 更新 |
| QUEUE.SYSTEM.FTE.COMMAND.agent_name | 更新 |
| CONTEXT.SYSTEM.FTE.COMMAND.agent_name | 更新 |
| QUEUE.SYSTEM.FTE.STATE.agent_name | 更新 |
| QUEUE.SYSTEM.FTE.DATA.agent_name | 更新 |
| QUEUE.SYSTEM.FTE.REPLY.agent_name | 更新 |
| QUEUE.SYSTEM.FTE.AUTHAGT1.agent_name | 更新 |
| QUEUE.SYSTEM.FTE.AUTHTRN1.agent_name | 更新 |
| QUEUE.SYSTEM.FTE.AUTHOPS1.agent_name | 更新 |
| QUEUE.SYSTEM.FTE.AUTHSCH1.agent_name | 更新 |
| QUEUE.SYSTEM.FTE.AUTHMON1.agent_name | 更新 |
| QUEUE.SYSTEM.FTE.AUTHADM1.agent_name | 更新 |

表 26:

| 姓名 | 需要存取權 |
|--------------------------------|-------|
| SYSTEM.FTE.AUTHAGT1.agent_name | 更新 |
| SYSTEM.FTE.AUTHTRN1.agent_name | 更新 |

| 表 26: (繼續) | |
|--------------------------------|-------|
| 姓名 | 需要存取權 |
| SYSTEM.FTE.AUTHOPS1.agent_name | 更新 |
| SYSTEM.FTE.AUTHSCH1.agent_name | 更新 |
| SYSTEM.FTE.AUTHMON1.agent_name | 更新 |

您可以使用使用者沙盤推演，來判斷要求傳送的使用者可以存取檔案系統的哪些部分。

若要啟用使用者沙盤推演，請將 `userSandboxes=true` 陳述式新增至您要限制之代理程式的 `agent.properties` 檔案，並將適當的值新增至 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name/UserSandboxes.xml` 檔案。

如需進一步資訊，請參閱 [使用使用者沙盤推演](#)。

這個使用者 ID 配置在 `UserSandboxes.xml` 檔中。

此 XML 檔案具有使用者 ID 或使用者 ID* 之類的資訊，以及可以使用 (已併入) 或無法使用 (已排除) 的資源清單。您需要定義可存取哪些資源的特定使用者 ID: 例如:

| 表 27: | | | |
|---------|----|-------|-------------------------|
| 使用者 ID | 存取 | 包含或排除 | 資源 |
| 管理 * | 讀取 | 併入 | /home/user/** |
| 管理 * | 讀取 | 排除 | /home/user/private/** |
| Sysprog | 讀取 | 併入 | /home/user/** |
| 管理 * | 讀取 | 併入 | Application.reply.queue |

附註:

1. 如果指定 `type=queue`，則資源是佇列名稱或 `queue@qmgr`。
2. 如果資源以 // 開頭，則資源是資料集; 否則資源是 USS 中的檔案。
3. 使用者 ID 是來自 MQMD 結構的使用者 ID，因此這可能不會反映實際放置訊息的使用者 ID。
4. 對於本端佇列管理程式上的要求，您可以使用 `MQADMIN CONTEXT.*` 以限制哪些使用者可以設定此值。
5. 對於透過遠端佇列管理程式進入的要求，您必須假設分散式佇列管理程式已啟用安全，以防止 MQMD 結構中使用者 ID 的未獲授權設定。
6. Linux 機器上 `SYSPROG1` 的使用者 ID，與 z/OS 安全檢查的使用者 ID `SYSPROG1` 相同。

我需要多少特工?

代理程式會執行傳送資料的工作，當您提出傳送資料的要求時，您可以指定代理程式的名稱。

依預設，代理程式可以同時處理 25 個傳送及 25 個接收要求。您可以配置這些程序。如需相關資訊，請參閱 [z/OS 上的 IBM MQ Managed File Transfer 配置選項](#)。

如果代理程式忙碌，則會將工作排入佇列。處理要求所花費的時間取決於多個因素，例如要傳送的資料量、網路頻寬及網路上的延遲。

您可能想要讓多個代理程式平行處理工作。

您也可以控制代理程式可以存取哪些資源，因此您可能想要部分代理程式使用有限的資料子集。

如果您想要處理具有不同優先順序的要求，則可以使用多個代理程式，並使用工作量管理程式來設定工作的優先順序。

執行代理程式

代理程式通常是長時間執行的處理程序。程序可以作為批次執行的工作提交，或作為已啟動作業提交。

規劃通道起始程式 SMF 資料

您需要規劃通道起始程式 (CHINIT) 的 SMF 資料實作。

CHINIT 會產生兩種類型的記錄：

- 含有 CHINIT 及其內作業相關資訊的統計資料。
- 通道統計資料，具有與 DIS CHSTATUS 指令類似的資訊。

您可以使用下列項目開始收集統計資料：

```
/CPF START TRACE(STAT) class(4)
```

並停止使用

```
/CPF STOP TRACE(STAT) class(4)
```

您可以使用下列方式開始收集統計資料：

```
/CPF START TRACE(ACCTG) class(4)
```

並停止使用

```
/CPF STOP TRACE(ACCTG) class(4)
```

在下列情況下會產生 SMF 記錄：

- 在 SMF 播送上已經歷 STATIME ZPARAM 參數中的時間間隔 (如果 STATIME 為零)。收集 CHINIT 和佇列管理程式的 SMF 資料的要求已同步。
- 發出 STOP TRACE(ACCTG) CLASS(4) 或 STOP TRACE(STAT) CLASS(4) 指令，或
- 當 CHINIT 關閉時。此時會寫出任何 SMF。

統計資料 SMF 資料通常適合一個 SMF 記錄，不過，如果使用大量作業，則可能會建立多個 SMF 記錄。

會針對已啟用帳戶資料的每一個通道 (使用 STATCHL 屬性) 收集帳戶資料，且通常適合一個 SMF 記錄。不過，如果有大量通道在作用中，則可能會建立多個 SMF 記錄。

如果通道在間隔中停止，下次 SMF 處理執行時，會將統計資料寫入 SMF。如果用戶端連接、執行一些工作及中斷連接，然後重新連接及中斷連接，則會產生兩組通道統計資料。

您可以控制哪些通道具有寫入 SMF 的資訊：

1. 在通道及佇列管理程式上使用 STATCHL 選項。
2. 對於用戶端通道，請注意，您必須在佇列管理程式層次設定 STATCHL。
3. 對於自動定義的叢集傳送端通道，您必須設定 STATACLS。

使用 CHINIT SMF 資料的成本很小。通常 CPU 使用率增加低於幾個百分比，且通常在測量錯誤內。

在使用此功能之前，您需要與 z/OS 系統程式設計師一起工作，以確保 SMF 具有額外記錄的容量，且他們會變更其擷取 SMF 記錄的處理程序，以包括新的 SMF 資料。

對於 CHINIT 統計資料，SMF 記錄類型為 115，次類型為 231。

對於 CHINIT 統計作業，SMF 記錄類型為 116，子類型為 10。

您可以撰寫自己的程式來處理此資料，或使用包含程式 MQSMF 的 SupportPac [MP1B](#) 來列印資料，並以適合匯入至展開工作表的逗點區隔值 (CSV) 來建立資料。

如果您在擷取通道起始程式 SMF 資料時遇到問題，請參閱 [在擷取通道起始程式 \(CHINIT\) 的 SMF 資料時處理問題](#)，以取得進一步資訊。

相關資訊

[解譯 IBM MQ 效能統計資料](#)
[通道統計資料疑難排解](#)

規劃備份及回復

在您的網站上開發備份及回復程序，對於避免代價高昂且耗時的資料流失非常重要。IBM MQ 提供在系統失敗之後，將佇列及訊息回復至其現行狀態的方法。

本主題包含下列各節：

- [第 152 頁的『回復程序』](#)
- [第 152 頁的『備份及回復提示』](#)
- [第 154 頁的『回復頁集』](#)
- [第 155 頁的『回復 CF 結構』](#)
- [第 155 頁的『達到特定的回復目標』](#)
- [第 157 頁的『其他產品的備份考量』](#)
- [第 157 頁的『回復及 CICS』](#)
- [第 157 頁的『回復及 IMS』](#)
- [第 157 頁的『在替代網站上準備回復』](#)
- [第 157 頁的『佇列管理程式備份活動的範例』](#)

回復程序

針對 IBM MQ 開發下列程序：

- 正在建立回復點。
- 正在備份頁面集。
- 備份 CF 結構。
- 正在回復頁集。
- 從空間不足狀況 (IBM MQ 日誌及頁集) 回復。
- 正在回復 CF 結構。

如需這些的相關資訊，請參閱 [管理 IBM MQ for z/OS](#)。

熟悉您網站所使用的下列程序：

- 從硬體或電源故障中回復。
- 從 z/OS 元件故障回復。
- 使用離站回復，從站台岔斷回復。

備份及回復提示

請利用這個主題來瞭解一些備份及回復作業。

佇列管理程式重新啟動處理程序會將日誌資訊套用至頁集，以將資料回復至一致狀態。如果頁集已損壞或無法使用，您可以使用頁集的備份副本來解決問題 (如果所有日誌都可用)。如果日誌資料集已損壞或無法使用，則可能無法完全回復。

請考量下列幾點：

- [定期取得備份副本](#)
- [不要捨棄您可能需要的保存日誌](#)

- [不要將 DDname 變更為頁面集關聯](#)

定期取得備份副本

回復點 這個術語用來說明一組 IBM MQ 頁集的備份副本，以及回復這些頁集所需的對應日誌資料集。如果發生頁集流失（例如，頁集 I/O 錯誤），則這些備份副本會提供可能的重新啟動點。如果您使用這些備份副本來重新啟動佇列管理程式，則在取得這些副本之前，IBM MQ 中的資料是一致的。如果從此點開始提供所有日誌，則 IBM MQ 可以回復至失敗點。

備份副本越新，IBM MQ 越快可以回復頁集中的資料。頁集的回復取決於所有可用的必要日誌資料集。

在規劃回復時，您需要決定備份副本的頻率，以及要保留的完整備份週期數。這些值告訴您必須保留日誌資料集及頁集備份副本以進行 IBM MQ 回復的時間。

在決定取得備份副本的頻率時，請考量回復頁集所需的時間。所需時間由下列各項決定：

- 要遍訪的日誌數量。
- 操作員裝載及移除保存磁帶磁區所花費的時間。
- 讀取回復所需日誌部分所需的時間。
- 重新處理已變更頁面所需的時間。
- 用於備份副本的儲存媒體。
- 用來建立及還原備份副本的方法。

一般而言，建立備份副本的頻率越高，回復所花費的時間就越少，但建立副本所花費的時間就越多。

對於每一個佇列管理程式，您應該建立下列項目的備份副本：

- 保存日誌資料集
- 保存時建立的 BSDS 副本
- 頁面集
- 您的物件定義
- 您的 CF 結構

若要減少備份副本遺失或損壞的風險，請考量：

- 將不同儲存磁區上的備份副本儲存至原始副本。
- 將不同站台的備份副本儲存至原始副本。
- 為頁面集的每一個備份至少建立兩個副本，如果您使用單一記載或單一 BSDS，則會為保存日誌和 BSDS 建立兩個副本。如果您使用雙重記載或 BSDS，請建立保存日誌或 BSDS 的單一副本。

將 IBM MQ 移至正式作業環境之前，請完整測試並記載備份程序。

備份物件定義

建立物件定義的備份副本。如果要這麼做，請使用公用程式之 COMMAND 函數的 MAKEDEF 特性（說明於 [使用 CSQUTIL 的 COMMAND 函數](#)）。

每當您取得佇列管理程式資料集的備份副本，並保留最新版本時，就應該這麼做。

備份連結機能結構

如果您已設定任何佇列共用群組，即使您未使用它們，也必須定期備份 CF 結構。若要執行此動作，請使用 IBM MQ BACKUP CFSTRUCT 指令。您只能在以 RECOVER (YES) 屬性定義的 CF 結構上使用此指令。如果持續共用訊息的任何 CF 項目參照儲存在共用訊息資料集 (SMDS) 或 Db2 中的已卸載訊息資料，則會擷取已卸載資料並與 CF 項目一起備份。共用訊息資料集不應個別備份。

建議您大約每小時執行一次所有 CF 結構的備份，以將還原 CF 結構所花費的時間縮至最小。

您可以在單一佇列管理程式上執行所有 CF 結構備份，其優點是將日誌使用的增加限制為單一佇列管理程式。或者，您可以在佇列共用群組中的所有佇列管理程式上執行備份，這樣做的優點是將工作量分散在

佇列共用群組中。不論您使用何種策略，IBM MQ 都可以尋找備份，並從佇列共用群組中的任何佇列管理程式執行 RECOVER CFSTRUCT。需要存取佇列共用群組中所有佇列管理程式的日誌，才能回復 CF 結構。

備份訊息安全原則

如果您使用 IBM MQ Advanced Message Security 來建立訊息安全原則的備份，請使用 訊息安全原則公用程式 (CSQ0UTIL) 來建立備份，以搭配 `-export` 參數來執行 `dspmqspl`，然後儲存輸出至 EXPORT DD 的原則定義。

每當您建立佇列管理程式資料集的備份副本，並保留最新版本時，都應該建立訊息安全原則的備份。

不要捨棄您可能需要的保存日誌

在重新啟動期間，IBM MQ 可能需要使用保存日誌。您必須保留足夠的保存日誌，才能完全還原系統。IBM MQ 可能會使用保存日誌，從已還原的備份副本回復頁集。如果您已捨棄該保存日誌，則 IBM MQ 無法將頁集還原為其現行狀態。捨棄保存日誌資料集中說明何時及如何捨棄保存日誌。

您可以使用 `/cpf DIS USAGE TYPE(ALL)` 指令來顯示回復佇列管理程式的頁集及佇列共用群組結構所需的日誌 RBA 及日誌範圍序號 (LRSN)。然後，您應該使用 列印日誌對映公用程式 (CSQJU004) 來列印佇列管理程式的引導資料集 (BSDS) 資訊，以尋找包含日誌 RBA 的日誌。

對於結構，您需要在佇列共用群組中的每一個佇列管理程式上執行 CSQJU004 公用程式，以尋找包含 LRSN 的日誌。您需要這些日誌及任何後續日誌，才能回復頁面集及結構。

請勿將 DDname 變更為頁面集關聯

IBM MQ 使頁集號碼 00 與 DDname CSQP0000 相關聯，頁集號碼 01 與 DDname CSQP0001 相關聯，依此類推，直到 CSQP0099。IBM MQ 會根據頁集相關聯的 DD 名稱來寫入頁集的回復日誌記錄。基於此原因，您不得移動已與 PSID DDname 相關聯的頁集。

回復頁集

請利用這個主題來瞭解回復頁面集時所涉及的因素，以及如何將重新啟動時間縮至最短。

回復策略中的關鍵因素是您可以容忍佇列管理程式中斷的時間。總中斷時間可能包括從備份回復頁集，或在異常終止之後重新啟動佇列管理程式所花費的時間。影響重新啟動時間的因素包括您備份頁面集的頻率，以及在檢查點之間寫入日誌的資料量。

為了將異常終止之後的重新啟動時間縮至最短，請保持工作單元簡短，以便在系統重新啟動時最多使用兩個作用中日誌。例如，如果您要設計 IBM MQ 應用程式，請避免放置在第一個同步點 MQI 呼叫與確定點之間具有較長等待間隔的 MQGET 呼叫，因為這可能會導致工作單元的持續時間較長。工作單元較長的另一個常見原因是通道起始程式的批次間隔超過 5 分鐘。

您可以使用 `DISPLAY THREAD` 指令來顯示工作單元的 RBA，並協助解決舊的工作單元。

您必須多久備份一次頁面集？

如果需要相當短的回復時間，則必須經常備份頁集。即使頁面集非常小，或該頁面集中的佇列有少量活動，也會套用此選項。

如果您在頁面集中使用持續訊息，備份頻率應該以小時而非日為單位。這也是頁集零的情況。

若要計算大約備份頻率，請從判定目標回復時間總計開始。這由下列各項組成：

1. 回應問題所花費的時間。
2. 還原頁集備份副本所花費的時間。

如果您使用 SnapShot 備份/還原，則執行此作業所花費的時間是幾秒鐘。如需 SnapShot 的相關資訊，請參閱 *DFSMSdss Storage Administration Guide*。

3. 佇列管理程式重新啟動所需的時間，包括回復頁集所需的額外時間。

這最主要取決於自前次備份該頁集以來，必須從作用中及保存日誌讀取的日誌資料量。除了直接與損壞頁集相關聯的日誌資料之外，還必須讀取所有這類日誌資料。

註: 使用 模糊備份 (在工作單元處於作用中狀態時取得日誌和頁面集的 Snapshot) 時，可能需要讀取最多三個其他檢查點，這可能導致需要讀取一個以上其他日誌。

在決定容許回復頁集的時間長度時，您需要考量的因素如下：

- 在正常處理期間將資料寫入作用中日誌的速率，除了訊息速率之外，還取決於訊息到達系統的方式。透過通道接收或傳送的訊息會導致比本端產生及擷取的訊息更多的資料記載。
- 可從保存及作用中日誌讀取資料的速率。

讀取日誌時，可達到的資料傳送速率取決於使用的裝置以及特定 DASD 子系統上的總負載。

使用大部分磁帶機時，對於具有大型區塊大小的保存日誌，可以達到更高的資料速率。不過，如果回復需要保存日誌，則也必須讀取作用中日誌上的所有資料。

回復 CF 結構

使用本主題可瞭解 CF 結構的回復處理程序。

佇列共用群組中至少必須有一個佇列管理程式在作用中，才能處理 RECOVER CFSTRUCT 指令。CF 結構回復不會影響佇列管理程式重新啟動時間，因為已在作用中的佇列管理程式會執行回復。

回復處理程序由 RECOVER CFSTRUCT 指令所管理的兩個邏輯步驟組成：

1. 尋找並還原備份。
2. 從已使用 CF 結構之佇列共用群組中所有佇列管理程式的日誌，將所有已記載的更新項目合併至 CF 結構上所保留的持續訊息，並將變更套用至備份。

第二個步驟可能需要更長的時間，因為可能需要讀取許多日誌資料。如果您經常備份及/或同時回復多個 CF 結構，則可以減少所花費的時間。

執行回復的佇列管理程式會使用 Db2 中的資料及引導資料集，在所有其他佇列管理程式日誌上尋找相關備份。佇列管理程式會在佇列共用群組中以正確的時間順序重播這些備份，從最後一個備份之前到失敗點。

回復 CF 結構所需的時間取決於必須重播的回復日誌資料量，而這又取決於備份的頻率。在最糟糕的情況下，讀取佇列管理程式日誌所花費的時間與寫入佇列管理程式日誌所花費的時間相同。例如，如果您有一個包含六個佇列管理程式的佇列共用群組，則一小時的日誌活動可能需要六個小時才能重播。一般而言，它所花費的時間會比這個少，因為讀取可以大量完成，而且可以平行讀取不同佇列管理程式的日誌。作為起點，我們建議您每小時備份 CF 結構一次。

當 CF 結構失敗時，所有佇列管理程式都可以繼續使用其他 CF 結構中的非共用佇列及佇列。如果管理結構也失敗，則必須先啟動佇列共用群組中至少一個佇列管理程式，然後才能發出 RECOVER CFSTRUCT 指令。

備份 CF 結構可能需要大量日誌寫入容量，因此可以對執行備份的佇列管理程式強制大量載入。選擇負載較少的佇列管理程式來執行備份；若為忙碌系統，請將其他佇列管理程式新增至佇列共用群組，並將它專門用於執行備份。

達到特定的回復目標

請使用本主題，以取得如何透過調整備份頻率來達到特定回復目標時間的指引。

如果您有特定的回復目標要達成，例如，除了正常啟動時間在 xx 秒內之外，還需要完成佇列管理程式回復及重新啟動處理程序，您可以使用下列計算來預估備份頻率 (以小時為單位)：

$$\text{Backup frequency (in hours)} = \frac{\text{Required restart time (in secs)} * \text{System recovery log read rate (in MB/sec)}}{\text{Application log write rate (in MB/hour)}} \quad \text{Formula (A)}$$

註: 接下來提供的範例旨在強調顯示經常備份頁面集的需求。計算假設大部分日誌活動衍生自大量持續訊息。不過, 在某些狀況下, 日誌活動的數量並不容易計算。例如, 在佇列共用群組環境中, 除了其他資源之外, 還更新共用佇列的工作單元可能會導致 UOW 記錄寫入 IBM MQ 日誌。基於此原因, 公式 (A) 中的「應用程式日誌寫入率」只能從 IBM MQ 日誌填滿的觀察速率精確衍生。

例如, 假設系統中 IBM MQ MQI clients 每秒產生 100 個持續訊息的總負載。在此情況下, 會在本端產生所有訊息。

如果每則訊息的使用者長度為 1 KB, 則每小時所記載的資料量大約為:

$$100 * (1 + 1.3) \text{ KB} * 3600 = \text{approximately } 800 \text{ MB}$$

where

$$\begin{aligned} 100 &= \text{the message rate a second} \\ (1 + 1.3) \text{ KB} &= \text{the amount of data logged for each 1 KB of persistent messages} \end{aligned}$$

考量整體目標回復時間 75 分鐘。如果您已容許 15 分鐘回應問題並還原頁集備份副本, 則佇列管理程式回復及重新啟動必須在 60 分鐘 (3600 秒) 內完成, 並套用公式 (A)。假設所有必要的日誌資料都在 RVA2-T82 DASD 上, 回復率大約為每秒 2.7 MB, 這需要頁集備份頻率至少每隔:

$$3600 \text{ seconds} * 2.7 \text{ MB a second} / 800 \text{ MB an hour} = 12.15 \text{ hours}$$

如果您的 IBM MQ 應用程式日誌持續大約 12 小時, 則每天一個備份是適當的。不過, 如果應用程式日誌持續 24 小時, 則每天兩次備份會更適合。

另一個範例可能是正式作業系統, 其中所有訊息都適用於要求/回覆應用程式 (亦即, 在接收端通道上接收持續訊息, 並產生持續回覆訊息並透過傳送端通道傳送)。

在此範例中, 達到的批次大小是 1, 因此每則訊息都有一個批次。如果每秒有 50 個要求回覆, 則總負載是每秒 100 個持續訊息。如果每則訊息的長度為 1 KB, 則每小時所記載的資料量大約為:

$$50((2 * (1+1.3) \text{ KB}) + 1.4 \text{ KB} + 2.5 \text{ KB}) * 3600 = \text{approximately } 1500 \text{ MB}$$

where:

$$\begin{aligned} 50 &= \text{the message pair rate a second} \\ (2 * (1 + 1.3) \text{ KB}) &= \text{the amount of data logged for each message pair} \\ 1.4 \text{ KB} &= \text{the overhead for each batch of messages received by each channel} \\ 2.5 \text{ KB} &= \text{the overhead for each batch of messages sent by each channel} \end{aligned}$$

若要達到佇列管理程式回復並在 30 分鐘 (1800 秒) 內重新啟動, 再次假設所有必要的日誌資料都在 RVA2-T82 DASD 上, 這需要至少每隔下列時間執行一次頁集備份:

$$1800 \text{ seconds} * 2.7 \text{ MB a second} / 1500 \text{ MB an hour} = 3.24 \text{ hours}$$

定期審查備份頻率

監視 IBM MQ 日誌用量 (以每小時 MB 為單位)。請定期執行此檢查，並在必要時修正您的頁集備份頻率。

其他產品的備份考量

如果您將 IBM MQ 與 CICS 或 IMS 搭配使用，則還必須考量這些產品的備份策略含意。資料機能階層式儲存體管理程式 (DFHSM) 可管理資料儲存體，並可與 IBM MQ 使用的儲存體互動。

使用 DFHSM 進行備份及回復

資料機能階層式儲存體管理程式 (DFHSM) 會在系統中的儲存裝置之間執行自動空間可用性與資料可用性管理。如果您使用它，則需要知道它會自動將資料移入及移出 IBM MQ 儲存體。

DFHSM 會將最近未使用的資料集移至替代儲存體，以有效地管理 DASD 空間。它也會自動將新的或變更的資料集複製到磁帶或 DASD 備份磁區，讓您的資料可供回復使用。它可以刪除資料集，或將它們移至另一個裝置。其作業會在每日的指定時間進行，並容許在刪除或移動資料集之前保留預定的期間。

您也可以手動執行所有 DFHSM 作業。 *Data Facility Hierarchical Storage Manager* 使用手冊說明如何使用 DFHSM 指令。如果您將 DFHSM 與 IBM MQ 搭配使用，請注意 DFHSM 會執行下列動作：

- 使用已編目資料集。
- 在頁面集和日誌上運作。
- 支援 VSAM 資料集。

回復及 CICS

CICS 資源的回復不受 IBM MQ 的存在影響。CICS 會將 IBM MQ 辨識為非 CICS 資源 (或外部資源管理程式)，並使用 CICS 資源管理程式介面 (RMI) 將 IBM MQ 併入作為任何同步點協調要求的參與者。如需 CICS 回復的相關資訊，請參閱 CICS 回復及重新啟動手冊。如需 CICS 資源管理程式介面的相關資訊，請參閱 CICS 自訂作業手冊。

回復及 IMS

IMS 可將 IBM MQ 辨識為外部子系統及同步點協調的參與者。IMS 自訂作業手冊中說明外部子系統資源的 IMS 回復。

在替代網站上準備回復

如果完全失去 IBM MQ 運算中心，您可以在回復站台的另一個 IBM MQ 系統上回復。

若要在回復站台回復 IBM MQ 系統，您必須定期備份頁集及日誌。與所有資料回復作業一樣，災難回復的目標是盡可能減少資料、工作量處理 (更新) 及時間。

在回復站台：

- 回復 IBM MQ 佇列管理程式 **必須** 具有與遺失佇列管理程式相同的名稱。
- 請確定回復佇列管理程式上使用的系統參數模組包含與遺失佇列管理程式相同的參數。

[管理 IBM MQ for z/OS](#) 中說明災難回復的處理程序。

佇列管理程式備份活動的範例

本主題顯示作為佇列管理程式備份活動的範例。

當您規劃佇列管理程式備份策略時，主要考量是保留正確數量的日誌資料。管理日誌說明如何透過參照佇列管理程式的系統回復 RBA 來判定需要哪些日誌資料集。IBM MQ 會使用下列相關資訊來判定系統回復 RBA：

- 目前作用中的工作單元。

- 尚未從緩衝池清除至磁碟的頁集更新。
- CF 結構備份，以及此佇列管理程式的日誌是否包含使用它們的任何回復作業所需的資訊。

您必須保留足夠的日誌資料，才能執行媒體回復。雖然系統回復 RBA 會隨著時間而增加，但只有在執行後續備份時，才會減少必須保留的日誌資料量。CF 結構備份是由 IBM MQ 管理，因此在報告系統回復 RBA 時，會將這些備份納入考量。這表示實際上，只有在執行頁集備份時，才會減少必須保留的日誌資料量。

第 158 頁的圖 54 顯示屬於佇列共用群組成員之佇列管理程式上的備份活動範例、回復 RBA 如何隨著每一個備份而改變，以及如何影響必須保留的日誌資料量。在此範例中，佇列管理程式會使用本端及共用資源：頁集，以及兩個 CF 結構：STRUCTURE1。

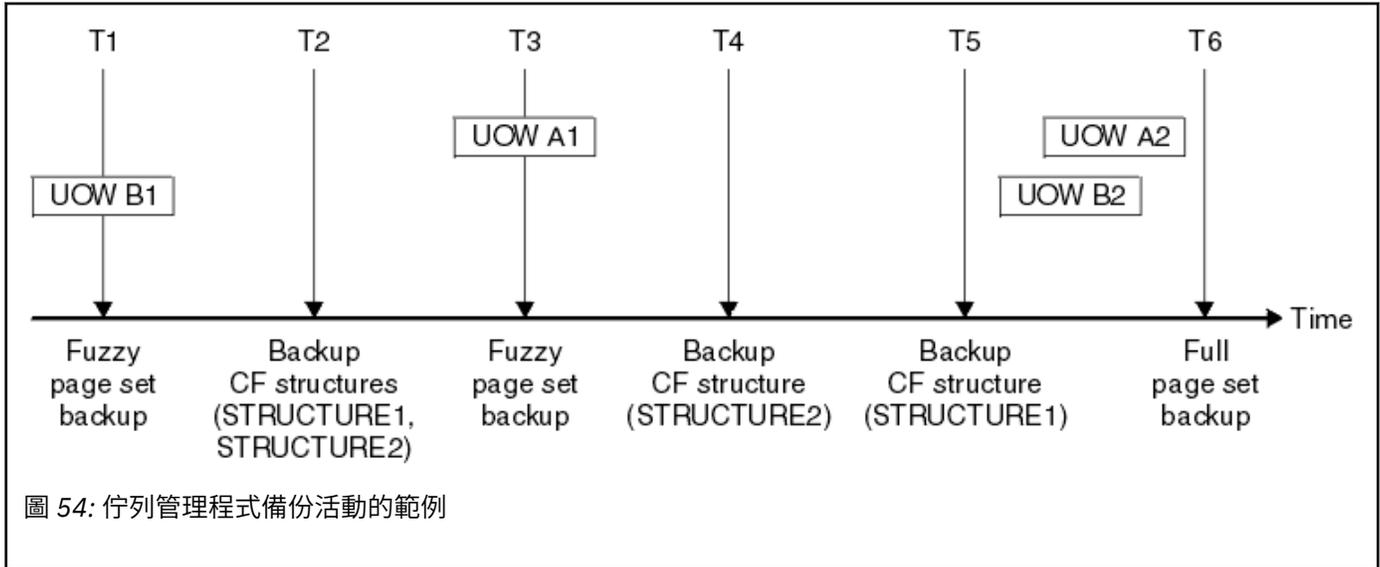


圖 54: 佇列管理程式備份活動的範例

這是在每一個時間點發生的情況：

時間點 T1

會建立頁面集的模糊備份，如 [如何備份及回復頁面集中所述](#)。

佇列管理程式的系統回復 RBA 是下列最低者：

- 此時正在備份之頁集的回復 RBA。
- 回復 CF 應用程式結構所需的最低回復 RBA。這與回復先前建立的 STRUCTURE1 的備份相關。
- 佇列管理程式 (UOWB1) 內最舊目前作用中工作單元的回復 RBA。

此時間點的系統回復 RBA 是由 DISPLAY USAGE 指令所發出的訊息所提供，該指令是模糊備份處理程序的一部分。

復原點 T2

建立 CF 結構的備份。會先備份 CF 結構 STRUCTURE1，然後再備份 STRUCTURE2。

必須保留的日誌資料量保持不變，因為使用 T1 所取得的頁集備份仍需要從 T1 的系統回復 RBA 所決定的相同資料。

時間點 T3

已建立另一個模糊備份。

佇列管理程式的系統回復 RBA 是下列最低者：

- 此時正在備份之頁集的回復 RBA。
- 回復 CF 結構 STRUCTURE1 所需的最低回復 RBA，因為在 STRUCTURE2 之前已備份 STRUCTURE1。
- 佇列管理程式 (UOWA1) 內最舊目前作用中工作單元的回復 RBA。

此時間點的系統回復 RBA 是由 DISPLAY USAGE 指令所發出的訊息所提供，該指令是模糊備份處理程序的一部分。

您現在可以減少所保留的日誌資料，如這個新的系統回復 RBA 所決定。

復原點 T4

建立 CF 結構 STRUCTURE2 的備份。回復最舊必要 CF 結構備份的回復 RBA 與 CF 結構 STRUCTURE1 的備份相關，該備份在時間 T2 時已備份。

建立此 CF 結構備份不會影響必須保留的日誌資料量。

時間點 T5

建立 CF 結構 STRUCTURE1 的備份。用於回復最舊必要 CF 結構備份的回復 RBA，現在與在 T4 時備份之 CF 結構 STRUCTURE2 的回復相關。

建立此 CF 結構備份不會影響必須保留的日誌資料量。

復原點 T6

會依照 [如何備份及回復頁集中的說明](#)，對您的頁集進行完整備份。

佇列管理程式的系統回復 RBA 是下列最低者：

- 此時正在備份之頁集的回復 RBA。
- 回復 CF 結構所需的最低回復 RBA。這與 CF 結構 STRUCTURE2 的回復相關。
- 佇列管理程式中最舊目前作用中工作單元的回復 RBA。在此情況下，沒有現行工作單元。

此時間點的系統回復 RBA 是由 DISPLAY USAGE 指令發出的訊息所提供，這是完整備份處理程序的一部分。

同樣地，可以減少保留的日誌資料，因為與完整備份相關聯的系統回復 RBA 更新。

規劃 z/OS UNIX 或 UNIX 系統服務環境

IBM MQ 佇列管理程式 (MSTR) 及通道起始程式 (CHIN) 內的某些處理程序會使用 z/OS UNIX 或 UNIX 系統服務來進行正常處理。如果您不想使用預設「UNIX 系統服務」配置，請規劃您的配置。

只要已設定全系統預設 OMVS 區段，就不需要任何特殊動作或自訂作業，IBM MQ 即可使用 UNIX 服務。

不想要 IBM MQ 使用訪客或預設 UID 及 OMVS 區段來呼叫 UNIX 系統服務的使用者，只需要根據預設區段來建立新 OMVS 區段的模型，因為 IBM MQ 不需要特殊許可權，也不會以超級使用者身分在 UNIX 內執行。

MSTR 和 CHIN 啟動作業使用者 ID 也需要 RACF OMVS 區段中的 UID。

規劃 z/OS TCP/IP 環境

若要透過網路取得最佳傳輸量，您必須使用大小為 64 KB 或以上的 TCP/IP 傳送及接收緩衝區。使用此大小，系統會最佳化其緩衝區大小。

如需相關資訊，請參閱 [高延遲網路的動態調整大小](#)。

您可以使用下列 Netstat 指令來檢查系統緩衝區大小，例如：

```
TSO NETSTAT ALL (CLIENT csq1CHIN
```

結果會顯示許多資訊，包括下列兩個值：

```
ReceiveBufferSize: 0000065536  
SendBufferSize: 0000065536
```

65536 是 64 KB。如果緩衝區大小小於 65536，您必須與網路團隊合作，在 TCPIP 程序中增加 PROFILE DDName 中的 **TCPSENDBFRSIZE** 和 **TCPRCVBUFRSIZE** 值。例如，您可以使用下列指令：

```
TCPCONFIG TCPSENDBFRSIZE 65536 TCPRCVBUFRSIZE 65536
```

如果您無法變更系統層面的 **TCPSENDBFRSIZE** 或 **TCPRCVBUFRSIZE** 設定，請聯絡 IBM 軟體支援中心。

注意事項

本資訊係針對 IBM 在美國所提供之產品與服務所開發。

在其他國家中，IBM 可能不會提供本書中所提的各項產品、服務或功能。請洽當地 IBM 業務代表，以取得當地目前提供的產品和服務之相關資訊。這份文件在提及 IBM 的產品、程式或服務時，不表示或暗示只能使用 IBM 的產品、程式或服務。只要未侵犯 IBM 的智慧財產權，任何功能相當的產品、程式或服務都可以取代 IBM 的產品、程式或服務。不過，任何非 IBM 的產品、程式或服務，使用者必須自行負責作業的評估和驗證責任。

本文件所說明之主題內容，IBM 可能擁有其專利或專利申請案。提供本文件不代表提供這些專利的授權。您可以書面提出授權查詢，來函請寄到：

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

如果是有關雙位元組 (DBCS) 資訊的授權查詢，請洽詢所在國的 IBM 智慧財產部門，或書面提出授權查詢，來函請寄到：

智慧財產權授權
法務部與智慧財產權法律
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

下列段落不適用於英國，若與任何其他國家之法律條款抵觸，亦不適用於該國： International Business Machines Corporation 只依 "現況" 提供本出版品，不提供任何明示或默示之保證，其中包括且不限於不侵權、可商用性或特定目的之適用性的隱含保證。有些地區在特定交易上，不允許排除明示或暗示的保證，因此，這項聲明不一定適合您。

這項資訊中可能有技術上或排版印刷上的訛誤。因此，IBM 會定期修訂；並將修訂後的內容納入新版中。IBM 隨時會改進及/或變更本出版品所提及的產品及/或程式，不另行通知。

本資訊中任何對非 IBM 網站的敘述僅供參考，IBM 對該網站並不提供任何保證。這些網站所提供的資料不是 IBM 本產品的資料內容，如果要使用這些網站的資料，您必須自行承擔風險。

IBM 得以各種適當的方式使用或散布由您提供的任何資訊，無需對您負責。

如果本程式的獲授權人為了 (i) 在個別建立的程式和其他程式（包括本程式）之間交換資訊，以及 (ii) 相互使用所交換的資訊，因而需要相關的資訊，請洽詢：

IBM Corporation
軟體交互作業能力協調程式，部門 49XA
3605 公路 52 N
Rochester, MN 55901
U.S.A.

在適當條款與條件之下，包括某些情況下（支付費用），或可使用此類資訊。

IBM 基於雙方之 IBM 客戶合約、IBM 國際程式授權合約或任何同等合約之條款，提供本資訊所提及的授權程式與其所有適用的授權資料。

本文件中所含的任何效能資料都是在受管制的環境下判定。因此不同作業環境之下所得的結果，可能會有很大的差異。有些測定已在開發階段系統上做過，不過這並不保證在一般系統上會出現相同結果。甚至有部分的測量，是利用插補法而得的估計值，實際結果可能有所不同。本文件的使用者應驗證其特定環境適用的資料。

本文件所提及之非 IBM 產品資訊，取自產品的供應商，或其發佈的聲明或其他公開管道。IBM 並未測試過這些產品，也無法確認這些非 IBM 產品的執行效能、相容性或任何對產品的其他主張是否完全無誤。有關非 IBM 產品的性能問題應直接洽詢該產品供應商。

有關 IBM 未來方針或目的之所有聲明，僅代表 IBM 的目標與主旨，隨時可能變更或撤銷，不必另行通知。

這份資訊含有日常商業運作所用的資料和報告範例。為了要使它們儘可能完整，範例包括個人、公司、品牌和產品的名稱。這些名稱全屬虛構，如與實際公司的名稱和住址雷同，純屬巧合。

著作權授權：

本資訊含有原始語言之範例應用程式，用以說明各作業平台中之程式設計技術。您可以基於研發、使用、銷售或散布符合作業平台（撰寫範例程式的作業平台）之應用程式介面的應用程式等目的，以任何形式複製、修改及散布這些範例程式，而不必向 IBM 付費。這些範例並未在所有情況下完整測試。因此，IBM 不保證或暗示這些程式的可靠性、有用性或功能。

若貴客戶正在閱讀本項資訊的電子檔，可能不會有照片和彩色說明。

程式設計介面資訊

程式設計介面資訊 (如果有提供的話) 旨在協助您建立與此程式搭配使用的應用軟體。

本書包含預期程式設計介面的相關資訊，可讓客戶撰寫程式以取得 WebSphere MQ 的服務。

不過，本資訊也可能包含診斷、修正和調整資訊。提供診斷、修正和調整資訊，是要協助您進行應用軟體的除錯。

重要：請勿使用此診斷、修改及調整資訊作為程式設計介面，因為它可能會變更。

商標

IBM、IBM 標誌 [ibm.com](http://www.ibm.com) 是 IBM Corporation 在全球許多適用範圍的商標。IBM 商標的最新清單可在 Web 的 "Copyright and trademark information" www.ibm.com/legal/copytrade.shtml 中找到。其他產品和服務名稱，可能是 IBM 或其他公司的商標。

Microsoft 及 Windows 是 Microsoft Corporation 在美國及/或其他國家或地區的商標。

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

Linux 是 Linus Torvalds 在美國及/或其他國家或地區的註冊商標。

本產品包含 Eclipse Project (<http://www.eclipse.org/>) 所開發的軟體。

Java 和所有以 Java 為基礎的商標及標誌是 Oracle 及/或其子公司的商標或註冊商標。



產品編號:

(1P) P/N: