

7.5

配置 *IBM WebSphere MQ*

IBM

附註

使用本資訊及其支援的產品之前，請先閱讀第 379 頁的『[注意事項](#)』中的資訊。

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

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

© Copyright International Business Machines Corporation 2007, 2024.

目錄

配置	5
在 UNIX、Linux 及 Windows 上配置多個安裝.....	5
在多重安裝環境中連接應用程式.....	5
變更主要安裝.....	12
將佇列管理程式與安裝相關聯.....	14
尋找系統上 IBM WebSphere MQ 的安裝.....	15
建立及管理佇列管理程式.....	16
建立預設佇列管理程式.....	18
將現有的佇列管理程式設為預設值.....	19
在建立佇列管理程式之後備份配置檔.....	20
啟動佇列管理程式.....	20
停止佇列管理程式.....	21
重新啟動佇列管理程式.....	22
刪除佇列管理程式.....	22
使用分散式佇列連接應用程式.....	23
IBM WebSphere MQ 分散式傳訊技術.....	24
分散式佇列管理簡介.....	41
在 UNIX、Linux 及 Windows 上監視及控制通道.....	63
配置伺服器與用戶端之間的連線.....	84
決定要使用的通訊類型.....	85
配置延伸交易式用戶端.....	87
定義 MQI 通道.....	95
在不同平台上建立伺服器連線和用戶端連線定義.....	96
在伺服器上建立伺服器連線及用戶端連線定義.....	99
通道結束程式.....	103
將用戶端連接至佇列共用群組.....	106
使用配置檔來配置用戶端.....	107
使用環境變數來配置用戶端.....	121
控制排入佇列的發佈/訂閱.....	127
設定排入佇列的發佈/訂閱訊息屬性.....	127
正在啟動排入佇列的發佈/訂閱.....	128
正在停止排入佇列的發佈/訂閱.....	129
新增串流.....	129
刪除串流.....	130
新增訂閱點.....	131
將佇列管理程式連接至階層.....	132
切斷佇列管理程式與階層的連線.....	133
配置佇列管理程式叢集.....	133
存取控制及多個叢集傳輸佇列.....	135
與分散式佇列比較.....	135
叢集的元件.....	137
如何選擇叢集佇列管理程式來保留完整儲存庫.....	148
組織叢集.....	149
叢集命名慣例.....	149
重疊叢集.....	150
叢集提示.....	151
在叢集中建立通訊.....	152
保留儲存庫資訊.....	154
管理 IBM WebSphere MQ 叢集.....	154
將訊息遞送至叢集以及從叢集遞送訊息.....	210
使用叢集進行工作量管理.....	221
叢集作業: 最佳作法.....	234

可用性、回復及重新啟動.....	260
自動用戶端重新連線.....	260
主控台訊息監視.....	265
高可用性配置.....	265
確定訊息未遺失 (記載).....	334
備份及還原 IBM WebSphere MQ 佇列管理程式資料.....	345
變更配置資訊.....	350
變更 UNIX、Linux 及 Windows 系統上的配置資訊.....	350
用於變更 IBM WebSphere MQ 配置資訊的屬性.....	355
變更佇列管理程式配置資訊.....	360
配置 HP Integrity NonStop Server.....	375
閘道處理程序概觀.....	375
配置閘道以在 Pathway 下執行.....	375
配置用戶端起始設定檔.....	376
將許可權授與通道.....	377
注意事項.....	379
程式設計介面資訊.....	380
商標.....	380

配置

在一或多部電腦上建立一或多個佇列管理程式，並在開發、測試及正式作業系統上配置它們，以處理包含商業資料的訊息。

在配置 IBM WebSphere MQ 之前，請先閱讀 [IBM WebSphere MQ 技術概觀](#) 中的 IBM WebSphere MQ 概念。瞭解如何在 [規劃](#) 中規劃 IBM WebSphere MQ 環境。

您可以使用許多不同的方法，在「IBM WebSphere MQ」中建立、配置及管理佇列管理程式及其相關資源。這些方法包括指令行介面、圖形使用者介面及管理 API。如需這些介面的相關資訊，請參閱 [管理 IBM WebSphere MQ](#)。

如需如何建立、啟動、停止及刪除佇列管理程式的指示，請參閱 [第 16 頁的『建立及管理佇列管理程式』](#)。

如需如何建立將 IBM WebSphere MQ 安裝與應用程式連接在一起所需的元件的相關資訊，請參閱 [第 23 頁的『使用分散式佇列連接應用程式』](#)。

如需如何使用不同方法將用戶端連接至 IBM WebSphere MQ 伺服器的指示，請參閱 [第 84 頁的『配置用戶端與伺服器之間的連線』](#)。

如需如何配置佇列管理程式叢集的指示，請參閱 [第 133 頁的『配置佇列管理程式叢集』](#)。

您可以透過變更配置資訊來變更 IBM WebSphere MQ 或佇列管理程式的行為。如需相關資訊，請參閱 [第 350 頁的『變更 IBM WebSphere MQ 及佇列管理程式配置資訊』](#)。一般而言，您不需要重新啟動佇列管理程式，任何配置變更就會生效，但在本產品說明文件中說明時除外。

相關概念

[WebSphere MQ 技術概觀](#)

相關工作

[管理本端 WebSphere MQ 物件](#)

[管理遠端 WebSphere MQ 物件](#)

[規劃](#)

在 UNIX, Linux, and Windows 上配置多個安裝

在相同系統上使用多個安裝時，您必須配置安裝及佇列管理程式。

此資訊適用於 UNIX, Linux®, and Windows。

使用下列鏈結中的資訊來配置安裝：

- [第 12 頁的『變更主要安裝』](#)
- [第 14 頁的『將佇列管理程式與安裝相關聯』](#)
- [第 5 頁的『在多重安裝環境中連接應用程式』](#)

相關概念

[多個安裝](#)

相關工作

[選擇主要安裝](#)

[選擇安裝名稱](#)

在多重安裝環境中連接應用程式

在 UNIX, Linux, and Windows 系統上，如果載入 IBM WebSphere MQ Version 7.1 或更新版本的程式庫，IBM WebSphere MQ 會自動使用適當的程式庫，而不需要採取任何進一步動作。IBM WebSphere MQ 使用與應用程式所連接之佇列管理程式相關聯之安裝中的程式庫。

下列概念用來說明應用程式連接至 IBM WebSphere MQ 的方式：

鏈結

編譯應用程式時，應用程式會鏈結至 IBM WebSphere MQ 程式庫，以取得函數匯出，然後在應用程式執行時載入這些函數匯出。

載入中

執行應用程式時，會找到並載入 IBM WebSphere MQ 程式庫。用來尋找程式庫的特定機制會因作業系統及建置應用程式的方式而異。如需如何在多個安裝環境中尋找及載入程式庫的相關資訊，請參閱 [第 7 頁的『載入 IBM WebSphere MQ Version 7.1 或更新版本的程式庫』](#)。

連接中

當應用程式連接執行中的佇列管理程式時 (例如，使用 MQCONN 或 MQCONNX 呼叫)，它會使用載入的 IBM WebSphere MQ 程式庫來連接。

當伺服器應用程式連接至佇列管理程式時，載入的程式庫必須來自與佇列管理程式相關聯的安裝。如果系統上有多個安裝架構，當選擇作業系統用來尋找要載入之 IBM WebSphere MQ 程式庫的機制時，這項限制會帶來新的挑戰：

- 當使用 **setmqm** 指令來變更與佇列管理程式相關聯的安裝時，需要載入的程式庫會變更。
- 當應用程式連接至不同安裝所擁有的多個佇列管理程式時，需要載入多組檔案庫。

不過，如果找到並載入 IBM WebSphere MQ Version 7.1 或更新版本的程式庫，則 IBM WebSphere MQ 會載入並使用適當的程式庫，而不需要採取任何進一步動作。當應用程式連接至佇列管理程式時，IBM WebSphere MQ 會從與佇列管理程式相關聯的安裝中載入程式庫。

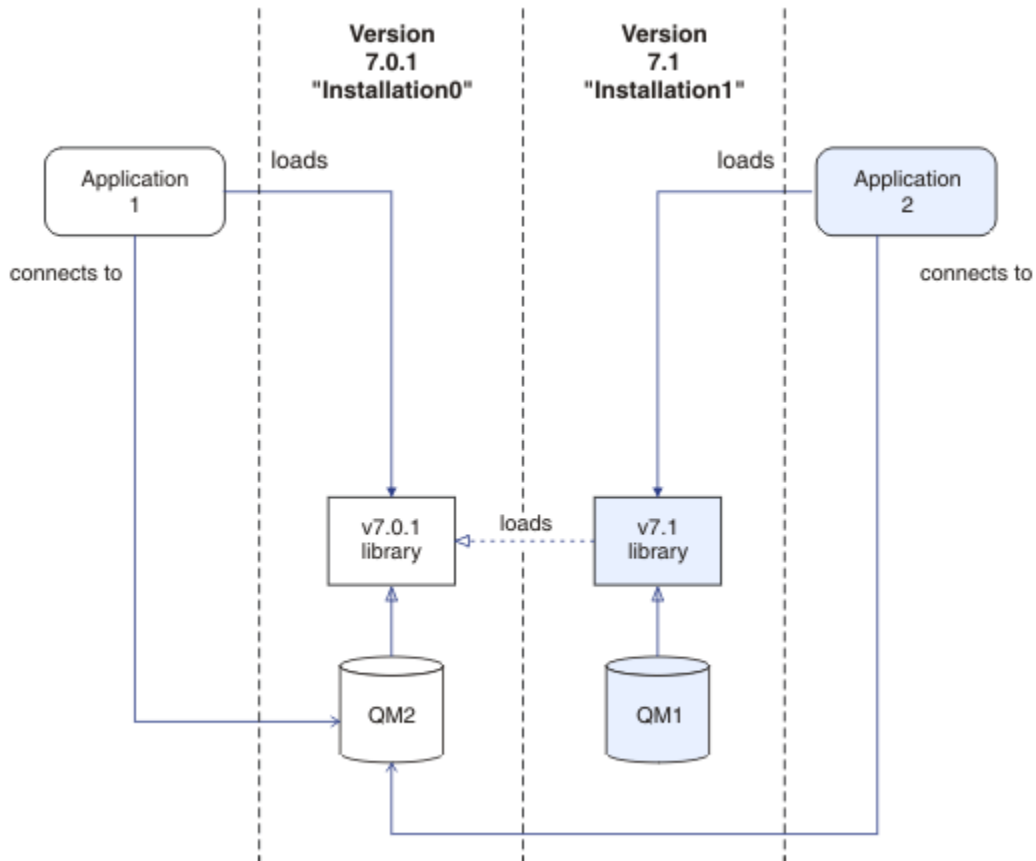


圖 1: 在多重安裝環境中連接應用程式

例如，[第 6 頁的圖 1](#) 顯示具有 7.0.1 版安裝 (Installation0) 及 7.1 版安裝 (Installation1) 的多重安裝環境。有兩個應用程式連接至這些安裝架構，但它們會載入不同的程式庫版本。

Application 1 會直接載入 7.0.1 版程式庫。當 application 1 連接至 QM2 時，會使用 7.0.1 版程式庫。如果 application 1 嘗試連接至 QM1，或 QM2 與 Installation1 相關聯，則 application 1 會失敗，並產生 `2059 (080B) (RC2059) :MQRC_Q_MGR_NOT_AVAILABLE` 錯誤。應用程式失敗，因為 7.0.1

版程式庫無法載入其他程式庫版本。也就是說，如果直接載入 7.0.1 版程式庫，則無法使用與 IBM WebSphere MQ 更新版本的安裝相關聯的佇列管理程式。

Application 2 會直接載入 7.1 版程式庫。當 application 2 連接至 QM2 時，7.1 版程式庫會載入並使用 7.0.1 版程式庫。如果 application 2 連接至 QM1，或 QM2 與 Installation1 相關聯，則會載入 7.1 版程式庫，且應用程式會如預期般運作。

在 UNIX、Linux 和 Windows 上的多重安裝佇列管理程式共存性中，會更詳細地考量移轉實務和連接具有多個安裝架構的應用程式。

如需如何載入 IBM WebSphere MQ Version 7.1 程式庫的相關資訊，請參閱 [第 7 頁的『載入 IBM WebSphere MQ Version 7.1 或更新版本的程式庫』](#)。

支援及限制

如果找到並載入下列任何 7.1 版或更新版本的程式庫，IBM WebSphere MQ 可以自動載入並使用適當的程式庫：

- C 伺服器檔案庫
- C++ 伺服器程式庫
- XA 伺服器程式庫
- COBOL 伺服器程式庫
- COM + 伺服器程式庫
- 未受管理模式中的 .NET

IBM WebSphere MQ 也會以連結模式自動載入及使用 Java 和 JMS 應用程式的適當程式庫。

使用多個安裝架構的應用程式有一些限制。如需相關資訊，請參閱 [第 10 頁的『使用多個安裝架構的應用程式限制』](#)。

相關概念

[第 14 頁的『將佇列管理程式與安裝相關聯』](#)

當您建立佇列管理程式時，它會自動與發出 `crtmqm` 指令的安裝相關聯。在 UNIX, Linux, and Windows 上，您可以使用 `setmqm` 指令來變更與佇列管理程式相關聯的安裝。

[第 10 頁的『使用多個安裝架構的應用程式限制』](#)

在多重安裝環境中使用 CICS 伺服器程式庫、捷徑連線、訊息控點及結束程式時，有一些限制。

[第 7 頁的『載入 IBM WebSphere MQ Version 7.1 或更新版本的程式庫』](#)

在決定如何載入 IBM WebSphere MQ 程式庫時，您需要考量許多因素，包括：您的環境、您是否可以變更現有的應用程式、您是否想要主要安裝架構、IBM WebSphere MQ 的安裝位置，以及 IBM WebSphere MQ 的位置是否可能變更。

相關工作

[選擇主要安裝](#)

[第 12 頁的『變更主要安裝』](#)

您可以使用 `setmqinst` 指令，將安裝設定或取消設定為主要安裝。

載入 IBM WebSphere MQ Version 7.1 或更新版本的程式庫

在決定如何載入 IBM WebSphere MQ 程式庫時，您需要考量許多因素，包括：您的環境、您是否可以變更現有的應用程式、您是否想要主要安裝架構、IBM WebSphere MQ 的安裝位置，以及 IBM WebSphere MQ 的位置是否可能變更。

IBM WebSphere MQ Version 7.1 程式庫的定位和載入方式視您的安裝環境而定：

- 在 UNIX and Linux 系統上，如果 IBM WebSphere MQ Version 7.1 的副本安裝在預設位置，則現有應用程式會繼續以與舊版相同的方式運作。不過，如果應用程式在 `/usr/lib` 中需要符號鏈結，您必須選取 7.1 版安裝架構作為主要安裝架構，或手動建立符號鏈結。
- 如果 IBM WebSphere MQ Version 7.1 安裝在非預設位置 (如果也安裝 IBM WebSphere MQ Version 7.0.1)，您可能需要變更現有的應用程式，以便載入正確的程式庫。

如何找到及載入 IBM WebSphere MQ Version 7.1 或更新版本的程式庫，也取決於如何設定任何現有的應用程式來載入程式庫。如需如何載入程式庫的相關資訊，請參閱第 9 頁的『作業系統程式庫載入機制』。

最理想的情況是，您應該確定佇列管理程式與作業系統所載入的 IBM WebSphere MQ 程式庫相關聯。

載入 IBM WebSphere MQ 程式庫的方法因平台而異，而且每一種方法都有優點和缺點。

平台	選項	好處	缺點
UNIX and Linux 系統	<p>設定或變更應用程式的內嵌式執行時期搜尋路徑 (RPath)。</p> <p>此選項需要您重新編譯並鏈結應用程式。如需編譯及鏈結應用程式的相關資訊，請參閱 建置 WebSphere MQ 應用程式。</p>	<ul style="list-style-type: none"> 變更的範圍很明確。 	<ul style="list-style-type: none"> 您必須能夠重新編譯及鏈結應用程式。 如果 IBM WebSphere MQ 的位置變更，您必須變更 RPath。
UNIX and Linux 系統	<p>使用 setmqenv 或 crtmqenv 搭配 -k 或 -l 選項來設定 <code>LD_LIBRARY_PATH</code> 環境變數 (AIX 上的 <code>LIBPATH</code>)。</p>	<ul style="list-style-type: none"> 不需要變更現有應用程式。 置換應用程式中的內嵌 RPaths。 如果 IBM WebSphere MQ 的位置變更，則容易變更變數。 	<ul style="list-style-type: none"> 基於安全理由，<code>setuid</code> 和 <code>setgid</code> 應用程式或以其他方式建置的應用程式可能會忽略 <code>LD_LIBRARY_PATH</code>。 環境專用，因此必須在執行應用程式的每一個環境中設定。 對其他依賴 <code>LD_LIBRARY_PATH</code> 的應用程式可能造成影響。 HP-UX: 編譯應用程式時使用的選項可能會停用 <code>LD_LIBRARY_PATH</code>。如需相關資訊，請參閱 HP-UX 的執行時期鏈結考量。 Linux: 用來建置應用程式的編譯器可能會停用 <code>LD_LIBRARY_PATH</code>。如需相關資訊，請參閱 Linux 的執行時期鏈結考量。
Windows 系統	<p>使用 setmqenv 或 crtmqenv 設定 <code>PATH</code> 變數。</p>	<ul style="list-style-type: none"> 現有應用程式不需要變更。 如果 IBM WebSphere MQ 的位置變更，則容易變更變數。 	<ul style="list-style-type: none"> 環境專用，因此必須在執行應用程式的每一個環境中設定。 對其他應用程式的可能影響。

表 1: 載入程式庫的選項的優點和缺點 (繼續)

平台	選項	好處	缺點
UNIX, Linux, and Windows 系統	<p>將主要安裝設為 7.1 版或更新版本安裝。請參閱第 12 頁的『變更主要安裝』。</p> <p>如需主要安裝的相關資訊，請參閱 選擇主要安裝。</p>	<ul style="list-style-type: none"> 現有應用程式不需要變更。 如果 IBM WebSphere MQ 的位置變更，則容易變更主要安裝。 將類似行為提供給舊版 IBM WebSphere MQ。 	<ul style="list-style-type: none"> 安裝 WebSphere MQ 7.0.1 版時，您無法將主要安裝設為 7.1 版或更新版本。 UNIX and Linux: 如果 /usr/lib 不在預設搜尋路徑中，則無法運作。

HP-UX 的程式庫載入考量

舊版 IBM WebSphere MQ 的產品說明文件中的範例編譯指令包括 64 位元應用程式的 `-w1, +noenvvar` 鏈結選項。這個選項會停用 `LD_LIBRARY_PATH`，以載入共用程式庫。如果您想要應用程式從 RPath 中指定的位置以外的位置載入 IBM WebSphere MQ 程式庫，則必須更新應用程式。您可以在不使用 `-w1, +noenvvar` 鏈結選項的情況下重新編譯及鏈結，或使用 `chatr` 指令來更新應用程式。

若要瞭解應用程式目前如何載入程式庫，請參閱 [第 9 頁的『作業系統程式庫載入機制』](#)。

Linux 的程式庫載入考量

使用部分版本的 gcc (例如 3.2.x 版) 編譯的應用程式可以具有無法使用 `LD_LIBRARY_PATH` 環境變數置換的內嵌 RPath。您可以使用 `readelf -d applicationName` 指令來判斷應用程式是否受到影響。如果 RPATH 符號存在，且 RUNPATH 符號不存在，則無法置換 RPath。

Solaris 的程式庫載入考量

舊版 IBM WebSphere MQ 的產品說明文件中的範例編譯指令包括 `-lmqmc -lmqzse` 鏈結選項。IBM WebSphere MQ 現在會自動載入這些程式庫的適當版本。如果 IBM WebSphere MQ 安裝在非預設位置，或系統上有多個安裝，您必須更新應用程式。您可以重新編譯並鏈結，而不使用 `-lmqmc -lmqzse` 鏈結選項來更新應用程式。

作業系統程式庫載入機制

在 Windows 系統上，會搜尋數個目錄來尋找程式庫：

- 從中載入應用程式的目錄。
- 現行目錄。
- `PATH` 環境變數中的目錄，包括現行使用者的廣域 `PATH` 變數及 `PATH` 變數。

在 UNIX and Linux 系統上，可能已使用一些方法來尋找要載入的程式庫：

- 使用 `LD_LIBRARY_PATH` 環境變數 (在 AIX 上亦為 `LIBPATH`，在 HP-UX 上為 `SHLIB_PATH`)。如果設定此變數，則會定義一組目錄，以搜尋所需的 WebSphere MQ 程式庫。如果在這些目錄中找到任何檔案庫，則會優先使用可能使用其他方法找到的任何檔案庫。
- 使用內嵌搜尋路徑 (RPath)。應用程式可能包含一組目錄來搜尋 IBM WebSphere MQ 程式庫。如果未設定 `LD_LIBRARY_PATH`，或使用變數找不到必要的檔案庫，則會在 RPath 中搜尋檔案庫。如果您現有的應用程式使用 RPath，但無法重新編譯及鏈結應用程式，則必須在預設位置安裝 IBM WebSphere MQ Version 7.1，或使用其他方法來尋找程式庫。
- 使用預設程式庫路徑。在搜尋 `LD_LIBRARY_PATH` 變數及 RPath 位置之後，如果找不到 WebSphere MQ 程式庫，則會搜尋預設程式庫路徑。通常，此路徑包含 `/usr/lib` 或 `/usr/lib64`。如果在搜尋預設程式庫路徑之後找不到程式庫，應用程式會因為遺漏相依關係而無法啟動。

您可以使用作業系統機制來找出應用程式是否有內嵌的搜尋路徑。例如：

- AIX: `dump`

- HP-UX: **chatr**
- Linux: **readelf**
- Solaris: **elfdump**

相關概念

第 14 頁的『[將佇列管理程式與安裝相關聯](#)』

當您建立佇列管理程式時，它會自動與發出 **crtmqm** 指令的安裝相關聯。在 UNIX, Linux, and Windows 上，您可以使用 **setmqm** 指令來變更與佇列管理程式相關聯的安裝。

第 10 頁的『[使用多個安裝架構的應用程式限制](#)』

在多重安裝環境中使用 CICS 伺服器程式庫、捷徑連線、訊息控點及結束程式時，有一些限制。

第 5 頁的『[在多重安裝環境中連接應用程式](#)』

在 UNIX, Linux, and Windows 系統上，如果載入 IBM WebSphere MQ Version 7.1 或更新版本的程式庫，IBM WebSphere MQ 會自動使用適當的程式庫，而不需要採取任何進一步動作。IBM WebSphere MQ 使用與應用程式所連接之佇列管理程式相關聯之安裝中的程式庫。

相關工作

選擇主要安裝

第 12 頁的『[變更主要安裝](#)』

您可以使用 **setmqinst** 指令，將安裝設定或取消設定為主要安裝。

使用多個安裝架構的應用程式限制

在多重安裝環境中使用 CICS 伺服器程式庫、捷徑連線、訊息控點及結束程式時，有一些限制。

CICS 伺服器程式庫

如果您使用 CICS 伺服器檔案庫，IBM WebSphere MQ 不會自動為您選取正確的檔案庫層次。您必須針對應用程式所連接的佇列管理程式，編譯並鏈結您的應用程式與適當的程式庫層次。如需相關資訊，請參閱 [建置程式庫以與 TXSeries for Multiplatforms 第 5 版搭配使用](#)。

訊息控點

使用特殊值 MQHC_UNASSOCIATED_HCONN 的訊息控點限制為與程序中載入的第一個安裝搭配使用。如果特定安裝無法使用訊息控點，則會傳回原因碼 MQRC_HMSG_NOT_AVAILABLE。

此限制會影響訊息內容。您無法使用訊息控點從某個安裝上的佇列管理程式取得訊息內容，並將它們放置到不同安裝上的佇列管理程式。如需訊息控點的相關資訊，請參閱 [MQCRTMH-建立訊息控點](#)。

結束程式

在多個安裝環境中，必須更新現有結束程式，以與 IBM WebSphere MQ Version 7.1 或更新版本安裝搭配使用。使用 **crtmqcvx** 指令產生的資料轉換結束程式必須使用更新的指令重新產生。

所有結束程式都必須使用 MQIEP 結構來撰寫，無法使用內嵌的 RPATH 來尋找 IBM WebSphere MQ 程式庫，也無法鏈結至 IBM WebSphere MQ 程式庫。如需相關資訊，請參閱 [撰寫及編譯結束程式和可安裝的服務](#)。

捷徑

在具有多個安裝的伺服器上，使用快速路徑連線至 IBM WebSphere MQ Version 7.1 或更新版本的應用程式必須遵循下列規則：

1. 佇列管理程式必須與應用程式從中載入 IBM WebSphere MQ 執行時期程式庫的同一個安裝相關聯。應用程式不能使用與另一個安裝相關聯的佇列管理程式的捷徑連線。嘗試建立連線會產生錯誤，原因碼為 MQRC_INSTALLATION_MISMATCH。
2. 將非捷徑連接至與應用程式已載入 IBM WebSphere MQ 執行時期程式庫的相同安裝相關聯的佇列管理程式，以防止應用程式連接捷徑，除非下列其中一項條件為真：
 - 應用程式第一次連線到與同一個安裝相關聯的佇列管理程式就是使用捷徑連線。

- 已設定環境變數 AMQ_SINGLE_INSTALLATION。
3. 以非捷徑連接與 Version 7.1 或更新版本安裝相關聯的佇列管理程式，對應用程式是否可以透過捷徑連線沒有影響。
 4. 您無法既連接與 Version 7.0.1 安裝相關聯的佇列管理程式，又以捷徑連接與 Version 7.1 或更新版本安裝相關聯的佇列管理程式。

設定 AMQ_SINGLE_INSTALLATION 之後，您可以將與佇列管理程式的任何連線設為捷徑連線。否則，將適用幾乎相同的限制：

- 安裝必須是從中載入 IBM WebSphere MQ 執行時期程式庫的同一個安裝。
- 相同程序上的每個連線必須指向同一個安裝。如果您嘗試連接至與不同安裝相關聯的佇列管理程式，則連線會失敗，原因碼為 MQRC_INSTALLATION_MISMATCH。請注意，在設定 AMQ_SINGLE_INSTALLATION 的情況下，此限制適用於所有連線，而不只是捷徑連線。
- 使用捷徑連線只會連接一個佇列管理程式。

相關參考

[MQCONN-連接佇列管理程式 \(延伸\)](#)

[MQIEP 結構](#)

[2583 \(0A17\) \(RC2583\) :MQRC_INSTALL_MISMATCH](#)

[2587 \(0A1B\) \(RC2587\): MQRC_HMSG_NOT_AVAILABLE](#)

[2590 \(0A1E\) \(RC2590\): MQRC_FASTPATH_NOT_AVAILABLE](#)

在多重安裝環境中連接 .NET 應用程式

依預設，應用程式會使用主要安裝架構中的 .NET 組件。如果沒有主要安裝，或您不想要使用主要安裝組件，則必須更新應用程式配置檔或 *DEVPATH* 環境變數。

如果系統上有主要安裝，則該安裝的 .NET 組件及原則檔會登錄至廣域組件快取 (GAC)。在每一個安裝的安裝路徑中可以找到所有其他安裝的 .NET 組件，但這些組件未向 GAC 登錄。因此，依預設，應用程式會使用主要安裝架構中的 .NET 組件來執行。如果符合下列任何情況，您必須更新應用程式配置檔：

- 您沒有主要安裝。
- 您不想要應用程式使用主要安裝組件。
- 主要安裝架構的 IBM WebSphere MQ 版本低於用來編譯應用程式的版本。

如需如何更新應用程式配置檔的相關資訊，請參閱 [第 11 頁的『使用應用程式配置檔連接 .NET 應用程式』](#)。

如果符合下列情況，您必須更新 *DEVPATH* 環境變數：

- 您想要應用程式使用非主要安裝架構中的組件，但主要安裝架構的版本與非主要安裝架構相同。

如需如何更新 *DEVPATH* 變數的相關資訊，請參閱 [第 12 頁的『使用 DEVPATH 連接 .NET 應用程式』](#)。

使用應用程式配置檔連接 .NET 應用程式

在應用程式配置檔內，您必須設定各種標籤來重新導向應用程式，以使用非來自主要安裝架構的組件。

下表顯示需要對應用程式配置檔進行的特定變更，以容許 .NET 應用程式使用特定組件進行連接：

表 2: 配置應用程式以使用特定組件		
	使用較低版本 IBM WebSphere MQ 編譯的應用程式	使用更高版本 IBM WebSphere MQ 編譯的應用程式
執行具有更高版本 IBM WebSphere MQ 主要安裝的應用程式。(GAC 中的更高版本組件):	不需要變更	不需要變更

表 2: 配置應用程式以使用特定組件 (繼續)

	使用較低版本 IBM WebSphere MQ 編譯的應用程式	使用更高版本 IBM WebSphere MQ 編譯的應用程式
使用較低版本 IBM WebSphere MQ 主要安裝來執行應用程式。 (GAC 中的較低版本組件):	不需要變更	在應用程式配置檔中: <ul style="list-style-type: none"> • 使用 <code><bindingRedirect></code> 標籤來指出使用 GAC 中較低版本的組件
使用更高版本的 IBM WebSphere MQ 非主要安裝來執行應用程式。 (安裝資料夾中的更高版本組件):	在應用程式配置檔中: <ul style="list-style-type: none"> • 使用 <code><codebase></code> 標籤來指向更高版本組件的位置 • 使用 <code><bindingRedirect></code> 標籤來指出使用更高版本組件 	在應用程式配置檔中: <ul style="list-style-type: none"> • 使用 <code><codebase></code> 標籤來指向更高版本組件的位置
使用較低版本的 IBM WebSphere MQ 非主要安裝來執行應用程式。 (安裝資料夾中的較低版本組件):	在應用程式配置檔中: <ul style="list-style-type: none"> • 使用 <code><codebase></code> 標籤來指向較低版本組件的位置 • 包含標籤 <code><publisherpolicy Apply=no></code> 	在應用程式配置檔中: <ul style="list-style-type: none"> • 使用 <code><codebase></code> 標籤來指向較低版本組件的位置 • 使用 <code><bindingRedirect></code> 標籤來指出使用較低版本組件 • 包含標籤 <code><publisherpolicy Apply=no></code>

資料夾 `MQ_INSTALLATION_PATH\tools\dotnet\samples\base` 中提供了範例應用程式配置檔 `NonPrimaryRedirect.config`。可以使用任何非主要安裝架構的 IBM WebSphere MQ 安裝路徑來修改此檔案。該檔案也可以使用 `<linkedConfiguration>` 標籤直接包含在其他配置檔中。提供 `nmqsget.exe.config` 和 `nmqsput.exe.config` 的範例。這兩個範例都使用 `<linkedConfiguration>` 標籤，並包含 `NonPrimaryRedirect.config` 檔案。

使用 DEVPATH 連接 .NET 應用程式

您可以使用 `DEVPATH` 環境變數來尋找組件。`DEVPATH` 變數指定的組件會優先於 GAC 中的任何組件使用。如需何時使用此變數的相關資訊，請參閱 `DEVPATH` 上的適當 Microsoft 文件。

若要使用 `DEVPATH` 環境變數來尋找組件，您必須將 `DEVPATH` 變數設為包含您要使用之組件的資料夾。然後，您必須更新應用程式配置檔，並新增下列執行時期配置資訊：

```
<configuration>
  <runtime>
    <developmentMode developerInstallation="true" />
  </runtime>
</configuration>
```

相關概念

第 5 頁的『在多重安裝環境中連接應用程式』

在 UNIX, Linux, and Windows 系統上，如果載入 IBM WebSphere MQ Version 7.1 或更新版本的程式庫，IBM WebSphere MQ 會自動使用適當的程式庫，而不需要採取任何進一步動作。IBM WebSphere MQ 使用與應用程式所連接之佇列管理程式相關聯之安裝中的程式庫。

[多個安裝](#)

[相關工作](#)

[選擇主要安裝](#)

[使用 .NET](#)

變更主要安裝

您可以使用 `setmqinst` 指令，將安裝設定或取消設定為主要安裝。

關於這項作業

此作業適用於 UNIX, Linux, and Windows。

主要安裝是必要的全系統位置所參照的安裝。如需主要安裝的相關資訊，以及選擇主要安裝的考量，請參閱 [選擇主要安裝](#)。

如果 IBM WebSphere MQ Version 7.1 或更新版本的安裝與 IBM WebSphere MQ Version 7.0.1 的安裝共存，則 IBM WebSphere MQ Version 7.0.1 安裝必須是主要安裝。當已安裝 IBM WebSphere MQ Version 7.1 或更新版本，且 IBM WebSphere MQ Version 7.1 或更新版本安裝無法成為主要安裝時，會將它標示為主要安裝。

在 Windows 上安裝程序期間，您可以指定安裝是主要安裝。在 UNIX and Linux 系統上，您必須在安裝之後發出 `setmqinst` 指令，以將安裝設為主要安裝。

第 13 頁的『設定主要安裝』。

第 13 頁的『取消設定主要安裝』。

設定主要安裝

程序

若要將安裝設為主要安裝，請執行下列動作：

1. 透過輸入下列指令，檢查安裝是否已成為主要安裝：

```
MQ_INSTALLATION_PATH/bin/dspmqinst
```

其中 `MQ_INSTALLATION_PATH` 是 IBM WebSphere MQ Version 7.1 或更新版本安裝架構的安裝路徑。

2. 如果現有 IBM WebSphere MQ Version 7.1 或更新版本安裝設為主要安裝，請遵循第 13 頁的『取消設定主要安裝』中的指示將它取消設定。如果 IBM WebSphere MQ Version 7.0.1 安裝在系統上，則無法變更主要安裝。
3. 以 root 使用者身分在 UNIX and Linux 系統上，或在 Windows 系統上以 Administrators 群組成員身分，輸入下列其中一個指令：

- 若要使用安裝路徑來設定主要安裝，您想要成為主要安裝：

```
MQ_INSTALLATION_PATH/bin/setmqinst -i -p MQ_INSTALLATION_PATH
```

- 若要使用您要作為主要安裝的安裝名稱來設定主要安裝，請執行下列動作：

```
MQ_INSTALLATION_PATH/bin/setmqinst -i -n installationName
```

4. 在 Windows 系統上，重新啟動系統。

取消設定主要安裝

程序

若要將安裝取消設為主要安裝，請執行下列動作：

1. 透過輸入下列指令，檢查哪個安裝是主要安裝：

```
MQ_INSTALLATION_PATH/bin/dspmqinst
```

其中 `MQ_INSTALLATION_PATH` 是 IBM WebSphere MQ Version 7.1 或更新版本安裝架構的安裝路徑。

如果 IBM WebSphere MQ Version 7.0.1 是主要安裝，則無法取消設定主要安裝。

2. 以 root 使用者身分在 UNIX and Linux 系統上，或在 Windows 系統上以 Administrators 群組成員身分，輸入下列其中一個指令：
 - 若要使用安裝路徑取消設定主要安裝，您不再想要作為主要安裝：


```
MQ_INSTALLATION_PATH/bin/setmqinst -x -p MQ_INSTALLATION_PATH
```

- 若要使用安裝名稱取消設定主要安裝，您不再想要作為主要安裝：

```
MQ_INSTALLATION_PATH/bin/setmqinst -x -n installationName
```

相關概念

[只能與 Windows 上的主要安裝搭配使用的功能](#)

[UNIX 及 Linux 上主要安裝的外部程式庫及控制指令鏈結](#)

相關工作

[解除安裝、升級及維護主要安裝](#)

[選擇安裝名稱](#)

相關參考

[setmqinst](#)

將佇列管理程式與安裝相關聯

當您建立佇列管理程式時，它會自動與發出 **crtmqm** 指令的安裝相關聯。在 UNIX, Linux, and Windows 上，您可以使用 **setmqm** 指令來變更與佇列管理程式相關聯的安裝。

您可以透過下列方式來使用 **setmqm** 指令：

- 在同等版本的 WebSphere MQ 之間移動個別佇列管理程式。例如，將佇列管理程式從測試移至正式作業系統。
- 將個別佇列管理程式從舊版 WebSphere MQ 移轉至新版 WebSphere MQ。在版本之間移轉佇列管理程式具有您必須瞭解的各種含意。如需移轉的相關資訊，請參閱 [移轉及升級 WebSphere MQ](#)。

若要將佇列管理程式與安裝相關聯，請執行下列動作：

1. 從目前與佇列管理程式相關聯的安裝架構中，使用 **endmqm** 指令來停止佇列管理程式。
2. 使用該安裝中的 **setmqm** 指令，將佇列管理程式與另一個安裝相關聯。

例如，若要將佇列管理程式 QMB 設為與名為 Installation2 的安裝相關聯，請從 Installation2: 輸入下列指令：

```
MQ_INSTALLATION_PATH/bin/setmqm -m QMB -n Installation2
```

其中 *MQ_INSTALLATION_PATH* 是 Installation2 的安裝路徑。

3. 從現在與佇列管理程式相關聯的安裝架構中，使用 **strmqm** 指令來啟動佇列管理程式。

此指令會執行任何必要的佇列管理程式移轉，並導致佇列管理程式備妥可供使用。

佇列管理程式與佇列管理程式相關聯的安裝會限制該佇列管理程式，因此只能由該安裝中的指令來管理。有三個主要異常狀況：

- **setmqm** 會變更與佇列管理程式相關聯的安裝。此指令必須從您要與佇列管理程式相關聯的安裝中發出，而不是從目前與佇列管理程式相關聯的安裝中發出。**setmqm** 指令指定的安裝名稱必須符合從中發出指令的安裝。
- **strmqm** 通常必須從與佇列管理程式相關聯的安裝發出。不過，當第一次在 V7.1 或更新版本安裝上啟動 V7.0.1 或更早版本佇列管理程式時，可以使用 **strmqm**。在此情況下，**strmqm** 會啟動佇列管理程式，並將它與從中發出指令的安裝相關聯。
- 「**dspmq**」會顯示系統上所有佇列管理程式的相關資訊，而不只是與 **dspmq** 指令相同安裝相關聯的那些佇列管理程式。**dspmq -o installation** 指令會顯示哪些佇列管理程式與哪些安裝相關聯的相關資訊。

HA 環境中的佇列管理程式關聯

對於 HA 環境，**addmqinf** 指令會自動將佇列管理程式與從中發出 **addmqinf** 指令的安裝相關聯。只要從與 **addmqinf** 指令相同的安裝發出 **strmqm** 指令，就不需要進一步設定。若要使用不同的安裝來啟動佇列管理程式，您必須先使用 **setmqm** 指令來變更相關聯的安裝。

與已刪除安裝相關聯的佇列管理程式

如果已刪除與佇列管理程式相關聯的安裝，或佇列管理程式狀態資訊無法使用，則 **setmqm** 指令無法將佇列管理程式與另一個安裝相關聯。在此狀況下，請採取下列動作：

1. 使用 **dspmqinst** 指令來查看系統上的其他安裝。
2. 手動修改 `mqm.ini` 中 `QueueManager` 段落的 `InstallationName` 欄位，以指定另一個安裝。
3. 從該安裝中使用 **dltmqm** 指令來刪除佇列管理程式。

相關概念

第 15 頁的『[尋找系統上 IBM WebSphere MQ 的安裝](#)』

如果系統上有多個 IBM WebSphere MQ 安裝，您可以檢查哪些版本已安裝及其位置。

第 351 頁的『[IBM WebSphere MQ 配置檔 `mqm.ini`](#)』

IBM WebSphere MQ 配置檔 `mqm.ini` 包含與節點上所有佇列管理程式相關的資訊。在安裝期間會自動建立它。

相關工作

[選擇主要安裝](#)

相關參考

[setmqm](#)

[strmqm](#)

[dspmq](#)

[dspmqinst](#)

尋找系統上 IBM WebSphere MQ 的安裝

如果系統上有多個 IBM WebSphere MQ 安裝，您可以檢查哪些版本已安裝及其位置。

您可以使用下列方法來尋找系統上的 IBM WebSphere MQ 安裝：

- 使用 **dspmqver** 指令。如果是從 Version 7.0.1 安裝架構發出，這個指令不會提供系統上所有安裝架構的詳細資料。
- 使用平台安裝工具來查詢已安裝 IBM WebSphere MQ 的位置。然後從 Version 7.1 或更新版本安裝中使用 **dspmqver** 指令。下列指令是您可以用來查詢已安裝 IBM WebSphere MQ 的指令範例：

- 在 AIX 系統上，您可以使用 **lslpp** 指令：

```
lslpp -R ALL -l mqm.base.runtime
```

- 在 HP-UX 系統上，您可以使用 **swlist** 指令：

```
swlist -a location -a revision -l product MQSERIES
```

- 在 Linux 系統上，您可以使用 **rpm** 指令：

```
rpm -qa --qf "%{NAME}-%{VERSION}-%{RELEASE}\t%{INSTPREFIXES}\n" | grep MQSeriesRuntime
```

- 在 Solaris 系統上，您可以使用 **pkginfo** 和 **pkgparam** 指令：

1. 輸入下列指令來列出已安裝的套件：

```
pkginfo | grep -w mqm
```

2. 對於列出的每一個套件，輸入下列指令：

```
pkgparam pkgname BASEDIR
```

- 在 Windows 系統上，您可以使用 **wmic** 指令。此指令可能會安裝 wmic 用戶端：

```
wmic product where "(Name like 'MQ%') AND (not Name like '%bitSupport')\" get Name, Version, InstallLocation
```

- 在 UNIX and Linux 系統上，發出下列指令以找出 IBM WebSphere MQ 的安裝位置:

```
cat /etc/opt/mqm/mqinst.ini
```

然後從 Version 7.1 或更新版本安裝中使用 **dspmqr** 指令。

- 若要在 32 位元 Windows 上顯示系統上安裝的詳細資料，請發出下列指令:

```
reg.exe query "HKEY_LOCAL_MACHINE\SOFTWARE\IBM\WebSphere MQ\Installation" /s
```

- 在 64 位元 Windows 上，發出下列指令:

```
reg.exe query "HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\IBM\WebSphere MQ\Installation" /s
```

附註: **reg.exe** 指令只會顯示 Version 7.1 或更新版本安裝的資訊。

相關概念

多個安裝

相關參考

[dspmqr](#)

[dspmqrinst](#)

建立及管理佇列管理程式

您必須先建立並啟動至少一個佇列管理程式及其相關物件，然後才能使用訊息及佇列。

建立佇列管理程式

佇列管理程式會管理與其相關聯的資源，特別是它所擁有的佇列。它為「訊息佇列作業介面 (MQI)」呼叫及指令提供佇列作業服務，以建立、修改、顯示及刪除 IBM WebSphere MQ 物件。

若要建立佇列管理程式，請使用 IBM WebSphere MQ 控制指令 **crtmqm** (在 **crtmqm** 中說明)。**crtmqm** 指令會自動建立必要的預設物件及系統物件 (如系統預設物件中所述)。預設物件會形成您所建立之任何物件定義的基礎; 佇列管理程式作業需要系統物件。當您已建立佇列管理程式及其物件時，請使用 **strmqm** 指令來啟動佇列管理程式。

註: IBM WebSphere MQ 不支援包含空格的機器名稱。如果您在機器名稱包含空格的電腦上安裝 IBM WebSphere MQ，則無法建立任何佇列管理程式。

開啟

在建立佇列管理程式之前，您必須考量數個要點 (尤其是在正式作業環境中)。完成下列核對清單:

與佇列管理程式相關聯的安裝

crtmqm 指令會自動將佇列管理程式與從中發出 **crtmqm** 指令的安裝相關聯。對於在佇列管理程式上操作的指令，您必須從與佇列管理程式相關聯的安裝中發出指令。您可以使用 **setmqm** 指令來變更佇列管理程式的相關聯安裝。請注意，Windows 安裝程式不會將執行安裝的使用者新增至 **mqm** 群組，如需詳細資料，請參閱在 [UNIX、Linux 及 Windows 系統上管理 IBM WebSphere MQ 的權限](#)。

命名慣例

使用「大寫」名稱以便與所有平台上的佇列管理程式通訊。請記住，在您輸入名稱時，會完全指派名稱。為了避免大量輸入的不便，請不要使用不必要的長名稱。

指定唯一佇列管理程式名稱

當您建立佇列管理程式時，請確定網路中沒有其他佇列管理程式具有相同的名稱 *anywhere*。建立佇列管理程式時，不會檢查佇列管理程式名稱，且名稱不是唯一的，會阻止您建立分散式佇列的通道。

確保唯一性的一種方法是在每一個佇列管理程式名稱前面加上其自己的唯一節點名稱。例如，如果節點稱為 **ACCOUNTS**，您可以將佇列管理程式命名為 **ACCOUNTS.SATURN.QUEUE.MANAGER**，其中 **SATURN** 會識別特定佇列管理程式，**QUEUE.MANAGER** 是您可以提供給所有佇列管理程式的延伸。或者，您可以省略此項目，但請注意 **ACCOUNTS.SATURN** 和 **ACCOUNTS.SATURN.QUEUE.MANAGER** 是不同的佇列管理程式名稱。

如果您使用 IBM WebSphere MQ 來與其他企業通訊，您也可以併入自己的企業名稱作為字首。在範例中沒有這樣做，因為它會使它們更難以遵循。

註：控制指令中的佇列管理程式名稱區分大小寫。這表示您可以建立兩個名稱為 `jupiter.queue.manager` 和 `JUPITER.queue.manager` 的佇列管理程式。不過，最好避免出現這些複雜情況。

限制佇列管理程式數目

您可以建立資源所容許的佇列管理程式數目。不過，由於每一個佇列管理程式都需要自己的資源，因此在節點上有一個佇列管理程式 (100 個佇列) 通常會比有 10 個佇列管理程式 (每個佇列有 10 個佇列) 來得好。

在正式作業系統中，許多處理器可以利用單一佇列管理程式，但較大的伺服器機器可以更有效地執行多個佇列管理程式。

指定預設佇列管理程式

雖然可以在沒有預設佇列管理程式的節點上配置 IBM WebSphere MQ，但每個節點都應該有一個預設佇列管理程式。如果應用程式未在 `MQCONN` 呼叫中指定佇列管理程式名稱，則預設佇列管理程式是應用程式所連接的佇列管理程式。當您在未指定佇列管理程式名稱的情況下呼叫 `runmqsc` 指令時，它也是處理 `MQSC` 指令的佇列管理程式。

將佇列管理程式指定為預設會取代節點的任何現有預設佇列管理程式規格。

變更預設佇列管理可能會影響其他使用者或應用程式。變更對目前連接的應用程式沒有影響，因為它們可以在任何進一步的 `MQI` 呼叫中使用來自其原始連接呼叫的控點。此控點可確保將呼叫導向至相同的佇列管理程式。在您變更預設佇列管理程式之後，任何連接的應用程式都會連接至新的預設佇列管理程式。這可能是您想要的，但在變更預設值之前，您應該將此納入考量。

第 18 頁的『[建立預設佇列管理程式](#)』中說明建立預設佇列管理程式。

指定無法傳送郵件的佇列

無法傳送郵件的佇列是一個本端佇列，如果訊息無法遞送至其預期的目的地，則會在其中放置訊息。

在您的網路中，每一個佇列管理程式都必須有無法傳送郵件的佇列，這很重要。如果您沒有定義，則應用程式中的錯誤可能會導致通道關閉，而且可能無法接收管理指令的回覆。

例如，如果應用程式嘗試將訊息放置在另一個佇列管理程式的佇列上，但提供錯誤的佇列名稱，則通道會停止，且訊息會保留在傳輸佇列上。然後，其他應用程式就無法將此通道用於其訊息。

如果佇列管理程式具有無法傳送郵件的佇列，則通道不受影響。無法遞送的訊息會放在接收端的無法傳送郵件的佇列中，讓通道及其傳輸佇列可供使用。

當您建立佇列管理程式時，請使用 `-u` 旗標來指定無法傳送郵件的佇列名稱。您也可以使用 `MQSC` 指令來變更已定義的佇列管理程式的屬性，以指定要使用的無法傳送郵件的佇列。如需 `MQSC` 指令 `ALTER` 的範例，請參閱 [使用佇列管理程式](#)。

指定預設傳輸佇列

傳輸佇列是一種本端佇列，傳輸至遠端佇列管理程式的訊息會在傳輸之前排入佇列。預設傳輸佇列是尚未明確定義傳輸佇列時所使用的佇列。每個佇列管理程式都可以指定預設傳輸佇列。

當您建立佇列管理程式時，請使用 `-d` 旗標來指定預設傳輸佇列的名稱。這不會實際建立佇列；您必須稍後明確地執行此動作。如需相關資訊，請參閱 [使用本端佇列](#)。

指定您需要的記載參數

您可以在 `crtmqm` 指令上指定記載參數，包括記載類型，以及日誌檔的路徑和大小。

在開發環境中，預設記載參數應該足夠。不過，您可以變更預設值，例如：

- 您的低階系統配置無法支援大型日誌。
- 您預期佇列上同時會有大量長訊息。
- 您預期會有許多持續訊息通過佇列管理程式。

一旦設定了記載參數，其中部分只能透過刪除佇列管理程式並使用相同名稱但不同的記載參數重建它來變更。

如需記載參數的相關資訊，請參閱第 260 頁的『[可用性、回復及重新啟動](#)』。

僅適用於 IBM WebSphere MQ for UNIX 系統

在使用 `crtmqm` 指令之前，您可以先建立佇列管理程式目錄 `/var/mqm/qmgrs/<qmgr>`，即使是在個別本端檔案系統上。當您使用 `crtmqm` 時，如果 `/var/mqm/qmgrs/<qmgr>` 目錄存在，它是空的，且由 `mqm` 擁有，則會用於佇列管理程式資料。如果該目錄不是由 `mqm` 所擁有，則建立會失敗，並顯示 First Failure Support Technology (FFST) 訊息。如果目錄不是空的，則會建立新目錄。

相關概念

[第 5 頁的『配置』](#)

在一或多部電腦上建立一或多個佇列管理程式，並在開發、測試及正式作業系統上配置它們，以處理包含商業資料的訊息。

[第 20 頁的『在建立佇列管理程式之後備份配置檔』](#)

IBM WebSphere MQ 配置資訊儲存在 Windows UNIX and Linux 系統上的配置檔中。

[第 20 頁的『啟動佇列管理程式』](#)

當您建立佇列管理程式時，必須啟動它，才能讓它處理指令或 MQI 呼叫。

[第 21 頁的『停止佇列管理程式』](#)

有三種方法可以停止佇列管理程式：靜止關機、立即關機，以及強制關機。

[第 22 頁的『重新啟動佇列管理程式』](#)

您可以使用 `strmqm` 指令來重新啟動佇列管理程式，或在 IBM WebSphere MQ for Windows 及 IBM WebSphere MQ for Linux (x86 及 x86-64 平台) 系統上，從「IBM WebSphere MQ 探險家」重新啟動佇列管理程式。

[第 350 頁的『變更 IBM WebSphere MQ 及佇列管理程式配置資訊』](#)

變更 IBM WebSphere MQ 或個別佇列管理程式的行為，以符合您安裝的需求。

[系統及預設物件](#)

相關工作

[第 19 頁的『將現有的佇列管理程式設為預設值』](#)

您可以讓現有的佇列管理程式成為預設佇列管理程式。您執行此動作的方式視您使用的平台而定。

[第 22 頁的『刪除佇列管理程式』](#)

您可以使用 `dlmqm` 指令或使用「WebSphere MQ 探險家」來刪除佇列管理程式。

distributed 建立預設佇列管理程式

如果應用程式未在 MQCONN 呼叫中指定佇列管理程式名稱，則預設佇列管理程式是應用程式所連接的佇列管理程式。當您呼叫 `runmqsc` 指令而不指定佇列管理程式名稱時，它也是處理 MQSC 指令的佇列管理程式。若要建立佇列管理程式，請使用 IBM WebSphere MQ 控制指令 `crtmqm`。

開始之前

在建立預設佇列管理程式之前，請先閱讀 [第 16 頁的『建立及管理佇列管理程式』](#) 中說明的考量。

UNIX 當您使用 `crtmqm` 在 UNIX and Linux 上建立佇列管理程式時，如果 `/var/mqm/qmgrs/<qmgr>` 目錄已存在，則由 `mqm` 擁有且為空白，它會用於佇列管理程式資料。如果該目錄不是 `mqm` 所擁有，則建立佇列管理程式會失敗，並產生 First Failure Support Technology (FFST) 訊息。如果目錄不是空的，則會為佇列管理程式資料建立新目錄。

即使 `/var/mqm/qmgrs/<qmgr>` 目錄已存在於個別本端檔案系統上，也會套用此考量。

關於這項作業

當您使用 `crtmqm` 指令來建立佇列管理程式時，指令會自動建立必要的預設物件及系統物件。預設物件構成您所建立之任何物件定義的基礎，且佇列管理程式作業需要系統物件。

例如，透過在指令中包括相關參數，您也可以定義佇列管理程式要使用的預設傳輸佇列名稱，以及無法傳送的郵件佇列名稱。

Windows 在 Windows 上，您可以使用 `crtmqm` 指令的 `sax` 選項來啟動佇列管理程式的多個實例。

如需 **crtmqm** 指令及其語法的相關資訊，請參閱 [crtmqm](#)。

程序

- 若要建立預設佇列管理程式，請搭配使用 **crtmqm** 指令與 **-q** 旗標。
下列 **crtmqm** 指令範例會建立稱為 **SATURN.QUEUE.MANAGER** 的預設佇列管理程式：

```
crtmqm -q -d MY.DEFAULT.XMIT.QUEUE -u SYSTEM.DEAD.LETTER.QUEUE SATURN.QUEUE.MANAGER
```

其中：

-q

指出此佇列管理程式是預設佇列管理程式。

-d MY.DEFAULT.XMIT.QUEUE

是此佇列管理程式要使用的預設傳輸佇列名稱。

註：IBM WebSphere MQ 不會為您建立預設傳輸佇列；您必須自行定義它。

-u SYSTEM.DEAD.LETTER.QUEUE

是 IBM WebSphere MQ 在安裝時建立的預設無法傳送郵件的佇列名稱。

SATURN.QUEUE.MANAGER

是此佇列管理程式的名稱。這必須是 **crtmqm** 指令上指定的最後一個參數。

下一步

當您已建立佇列管理程式及其物件時，請使用 **strmqm** 指令來 [啟動佇列管理程式](#)。

相關概念

第 20 頁的『在建立佇列管理程式之後備份配置檔』

IBM WebSphere MQ 配置資訊儲存在 Windows UNIX and Linux 系統上的配置檔中。

[使用佇列管理程式](#)

[使用本端佇列](#)

相關參考

[系統及預設物件](#)

將現有的佇列管理程式設為預設值

您可以讓現有的佇列管理程式成為預設佇列管理程式。您執行此動作的方式視您使用的平台而定。

WebSphere MQ for Windows 及 WebSphere MQ for Linux (x86 及 x86-64 平台) 系統

關於這項作業

使用下列指示，使現有的佇列管理程式成為 WebSphere MQ for Windows 及 WebSphere MQ for Linux (x86 及 x86-64 平台) 系統上的預設佇列管理程式：

程序

1. 開啟 IBM WebSphere MQ Explorer。
2. 用滑鼠右鍵按一下 IBM WebSphere MQ，然後選取 Properties...。即會顯示 WebSphere MQ 畫面。
3. 在預設佇列管理程式名稱欄位中輸入預設佇列管理程式的名稱。
4. 按一下確定。

UNIX and Linux 系統

關於這項作業

當您建立預設佇列管理程式時，其名稱會插入 WebSphere MQ 配置檔 (mqs.ini) 中 DefaultQueueManager 段落的 Name 屬性中。如果段落及其內容不存在，則會自動建立它們。

程序

- 若要將現有佇列管理程式設為預設值，請將 Name 屬性上的佇列管理程式名稱變更為新預設佇列管理程式的名稱。您可以使用文字編輯器來手動執行此動作。
- 如果節點上沒有預設佇列管理程式，且您想要使現有的佇列管理程式成為預設值，請自行建立具有所需名稱的 DefaultQueueManager 段落。
- 如果您不小心將另一個佇列管理程式設為預設值，並想要回復至原始預設佇列管理程式，請編輯 mqs.ini 中的 DefaultQueueManager 段落，將不想要的預設佇列管理程式取代為您想要的佇列管理程式。

下一步

如需配置檔的相關資訊，請參閱 [第 350 頁的『變更 IBM WebSphere MQ 及佇列管理程式配置資訊』](#)。

在建立佇列管理程式之後備份配置檔

IBM WebSphere MQ 配置資訊儲存在 Windows UNIX and Linux 系統上的配置檔中。

在 Windows 和 Linux (x86 和 x86-64) 系統上，請使用 IBM WebSphere MQ Explorer 來變更配置檔。

在 Windows 系統上，您也可以使用 amqmdain 指令來變更配置檔。請參閱 [amqmdain](#)

配置檔有兩種類型：

- 安裝產品時，會建立 IBM WebSphere MQ 配置檔 (mqs.ini)。它包含每次建立或刪除佇列管理程式時都會更新的佇列管理程式清單。每個節點有一個 mqs.ini 檔案。
- 當您建立新的佇列管理程式時，會自動建立新的佇列管理程式配置檔 (qm.ini)。這包含佇列管理程式的配置參數。

建立佇列管理程式之後，請備份配置檔。然後，如果您建立導致問題的另一個佇列管理程式，則可以在移除問題來源之後恢復備份。作為一般規則，每次建立新的佇列管理程式時，請備份您的配置檔。

如需配置檔的相關資訊，請參閱 [第 350 頁的『變更 IBM WebSphere MQ 及佇列管理程式配置資訊』](#)。

啟動佇列管理程式

當您建立佇列管理程式時，必須啟動它，才能讓它處理指令或 MQI 呼叫。

若要啟動佇列管理程式，請使用 **strmqm** 指令。

註：您必須從與您使用之佇列管理程式相關聯的安裝中使用 **strmqm** 指令。您可以使用 **dspmqs -o installation** 指令找出與佇列管理程式相關聯的安裝。

例如，若要啟動佇列管理程式 QMB，請輸入下列指令：

```
strmqm QMB
```

在 WebSphere MQ for Windows 及 WebSphere MQ for Linux (x86 及 x86-64 平台) 系統上，您可以啟動佇列管理程式，如下所示：

1. 開啟 IBM WebSphere MQ Explorer。
2. 從「Navigator」視圖中選取佇列管理程式。
3. 按一下 Start。即會啟動佇列管理程式。

如果佇列管理程式啟動花費數秒以上的時間，WebSphere MQ 會間歇性發出參考訊息，詳述啟動進度。

在佇列管理程式已啟動並準備好接受連線要求之前， `strmqm` 指令不會傳回控制權。

自動啟動佇列管理程式

在 WebSphere MQ for Windows 中，當系統使用「IBM WebSphere MQ 探險家」啟動時，您可以自動啟動佇列管理程式。如需相關資訊，請參閱 [使用 IBM WebSphere MQ Explorer 管理](#)。

停止佇列管理程式

有三種方法可以停止佇列管理程式：靜止關機、立即關機，以及強制關機。

使用 `endmqm` 指令來停止佇列管理程式。

註：您必須從與您使用之佇列管理程式相關聯的安裝中使用 `endmqm` 指令。您可以使用 `dspmqs -o installation` 指令找出與佇列管理程式相關聯的安裝。

例如，若要停止稱為 QMB 的佇列管理程式，請輸入下列指令：

```
endmqm QMB
```

在 WebSphere MQ for Windows 及 WebSphere MQ for Linux (x86 及 x86-64 平台) 系統上，您可以如下停止佇列管理程式：

1. 開啟 IBM WebSphere MQ Explorer。
2. 從「Navigator」視圖中選取佇列管理程式。
3. 按一下 Stop...。即會顯示「結束佇列管理程式」畫面。
4. 選取受控制或立即。
5. 按一下 OK。佇列管理程式停止。

靜止關機 (quiesced shutdown)

依預設， `endmqm` 指令會對指定的佇列管理程式執行靜止關閉。這可能需要一些時間才能完成。靜止關閉會等到所有已連接的應用程式都已斷線為止。

使用這種類型的關機來通知應用程式停止。如果您發出：

```
endmqm -c QMB
```

當所有應用程式都停止時，系統不會告訴您。（`endmqm -c QMB` 指令相當於 `endmqm QMB` 指令。）

不過，如果您發出：

```
endmqm -w QMB
```

指令會等到所有應用程式都已停止且佇列管理程式已結束為止。

立即關閉 (immediate shutdown)

若為立即關閉，任何現行 MQI 呼叫都可以完成，但任何新呼叫都會失敗。這種類型的關閉不會等待應用程式與佇列管理程式中斷連線。

若為立即關閉，請鍵入：

```
endmqm -i QMB
```

強制關機 (preemptive shutdown)

註：除非使用 `endmqm` 指令停止佇列管理程式的所有其他嘗試都失敗，否則請勿使用此方法。此方法可能會對連接的應用程式產生無法預期的結果。

如果立即關閉無法運作，您必須透過指定 `-p` 旗標來強制 強制 關閉。例如：

```
endmqm -p QMB
```

這會立即停止佇列管理程式。如果此方法仍然無法運作，請參閱 [手動停止佇列管理程式](#)，以取得替代解決方案。

如需 **endmqm** 指令及其選項的詳細說明，請參閱 [endmqm](#)。

如果您在關閉佇列管理程式時發生問題

關閉佇列管理程式的問題通常是由應用程式所造成。例如，當應用程式：

- 不要適當地檢查 MQI 回覆碼
- 不要求靜止通知
- 終止而不中斷與佇列管理程式的連線 (透過發出 MQDISC 呼叫)

如果在停止佇列管理程式時發生問題，您可以使用 Ctrl-C 來中斷 **endmqm** 指令。然後，您可以發出另一個 **endmqm** 指令，但這次帶有指定所需關機類型的旗標。

重新啟動佇列管理程式

您可以使用 **strmqm** 指令來重新啟動佇列管理程式，或在 IBM WebSphere MQ for Windows 及 IBM WebSphere MQ for Linux (x86 及 x86-64 平台) 系統上，從「IBM WebSphere MQ 探險家」重新啟動佇列管理程式。

若要重新啟動佇列管理程式，請鍵入：

```
strmqm saturn.queue.manager
```

在 IBM WebSphere MQ for Windows 及 IBM WebSphere MQ for Linux (x86 及 x86-64 平台) 系統上，您可以使用與啟動佇列管理程式相同的方式來重新啟動佇列管理程式，如下所示：

1. 開啟 IBM WebSphere MQ Explorer。
2. 從「Navigator」視圖中選取佇列管理程式。
3. 按一下 Start。佇列管理程式會重新啟動。

如果佇列管理程式重新啟動花費數秒以上的時間，IBM WebSphere MQ 會間歇性地發出參考訊息，詳述啟動進度。

刪除佇列管理程式

您可以使用 **dltmqm** 指令或使用「WebSphere MQ 探險家」來刪除佇列管理程式。

開始之前

停止佇列管理程式。

程序

- 發出下列指令：`dltmqm QMB`

註：您必須從與您使用之佇列管理程式相關聯的安裝中使用 **dltmqm** 指令。您可以使用 `dspm -o installation` 指令找出與佇列管理程式相關聯的安裝。

刪除佇列管理程式的步驟

關於這項作業

在 WebSphere MQ for Windows 及 WebSphere MQ for Linux (x86 及 x86-64 平台) 系統上，您可以刪除佇列管理程式，如下所示：

程序

1. 開啟「WebSphere MQ 探險家」。
2. 在「Navigator」視圖中，選取佇列管理程式。
3. 如果佇列管理程式未停止，請停止它。
 - a) 用滑鼠右鍵按一下佇列管理程式。
 - b) 按一下**停止**。
4. 用滑鼠右鍵按一下佇列管理程式。
5. 按一下**刪除**。

結果

已刪除佇列管理程式。



小心：

- 刪除佇列管理程式是一個重大步驟，因為您也會刪除與佇列管理程式相關聯的所有資源，包括所有佇列及其訊息和所有物件定義。如果您使用 **dltmqm** 指令，則沒有可讓您改變主意的顯示提示；當您按 Enter 鍵時，所有相關聯的資源都會遺失。
- 在 WebSphere MQ for Windows 中，刪除佇列管理程式也會從自動啟動清單中移除佇列管理程式（如第 20 頁的『啟動佇列管理程式』中所述）。當指令完成時，會顯示 WebSphere MQ queue manager ending 訊息；系統不會告訴您佇列管理程式已刪除。
- 刪除叢集佇列管理程式並不會將它從叢集中移除。如需相關資訊，請參閱 **dltmqm** 說明中的附註。

如需 **dltmqm** 指令及其選項的說明，請參閱 **dltmqm**。請確定只有授信管理者才有權使用此指令。（如需安全的相關資訊，請參閱 [在 Windows 上設定安全](#)，[UNIX and Linux 系統](#)。）

使用分散式佇列連接應用程式

本節提供 WebSphere MQ 安裝之間交互通訊的更詳細資訊，包括佇列定義、通道定義、觸發及同步點程序。閱讀本節之前，請先瞭解 [交互通訊的概念](#) 中引進的通道、佇列及其他概念。

使用下列鏈結中的資訊，以使用分散式佇列來連接應用程式：

- [第 43 頁的『如何將訊息傳送至另一個佇列管理程式』](#)
- [第 58 頁的『觸發通道』](#)
- [第 56 頁的『訊息安全』](#)
- [第 24 頁的『IBM WebSphere MQ 分散式傳訊技術』](#)
- [第 41 頁的『分散式佇列管理簡介』](#)
- [Windows](#) [UNIX](#) [Linux](#) [第 63 頁的『在 UNIX, Linux, and Windows 上監視及控制通道』](#)

相關概念

[第 84 頁的『配置用戶端與伺服器之間的連線』](#)

若要配置 WebSphere MQ MQI 用戶端與伺服器之間的通訊鏈結，請決定您的通訊協定，定義鏈結兩端的連線，啟動接聽器，以及定義通道。

[第 350 頁的『變更 IBM WebSphere MQ 及佇列管理程式配置資訊』](#)

變更 IBM WebSphere MQ 或個別佇列管理程式的行為，以符合您安裝的需求。

相關工作

[第 133 頁的『配置佇列管理程式叢集』](#)

請使用本主題中的鏈結，以瞭解叢集如何運作、如何設計叢集配置，以及取得如何設定簡式叢集的範例。

IBM WebSphere MQ 分散式傳訊技術

本節中的子主題說明規劃通道時使用的技術。這些子主題說明技術，可協助您規劃如何將佇列管理程式連接在一起，以及管理應用程式之間的訊息流程。

如需訊息通道規劃範例，請參閱：

- [分散式平台的訊息通道規劃範例](#)

相關概念

[第 23 頁的『使用分散式佇列連接應用程式』](#)

本節提供 WebSphere MQ 安裝之間交互通訊的更詳細資訊，包括佇列定義、通道定義、觸發及同步點程序通道

[訊息佇列作業簡介](#)

[交互通訊概念](#)

相關參考

[配置資訊範例](#)

訊息流程控制 (message flow control)

訊息流程控制是一項涉及在佇列管理程式之間設定及維護訊息路徑的作業。對於透過多個佇列管理程式的多重中繼站路徑而言，這很重要。本節說明如何在系統上使用佇列、別名佇列定義及訊息通道來達到訊息流程控制。

您可以使用 [第 23 頁的『使用分散式佇列連接應用程式』](#) 中引進的許多技術來控制訊息流程。如果佇列管理程式位於叢集中，則會使用不同的技術來控制訊息流程，如 [第 24 頁的『訊息流程控制 \(message flow control\)』](#) 中所述。

您可以使用下列物件來達到訊息流程控制：

- 傳輸佇列
- 訊息通道
- 遠端佇列定義
- 佇列管理程式別名定義
- 回覆目的地佇列別名定義

佇列管理程式及佇列物件在 [物件](#) 中說明。訊息通道在 [分散式佇列元件](#) 中有說明。下列技術使用這些物件在系統中建立訊息流程：

- 將訊息放入遠端佇列
- 透過特定傳輸佇列遞送
- 接收訊息
- 透過系統傳遞訊息
- 區隔訊息流程
- 將訊息流程切換至另一個目的地
- 將回覆目的地佇列名稱解析為別名

附註

本節中說明的所有概念都與網路中的所有節點相關，並且包括訊息通道的傳送和接收端。基於此原因，在大部分範例中只會說明一個節點。異常狀況是範例需要管理者在訊息通道的另一端進行明確合作。

在繼續使用個別技術之前，請先總結名稱解析的概念，以及三種使用遠端佇列定義的方式。請參閱 [交互通訊的概念](#)。

相關概念

[第 25 頁的『傳輸標頭中的佇列名稱』](#)

目的地佇列名稱會與傳輸標頭中的訊息一起傳送，直到到達目的地佇列為止。

第 25 頁的『如何建立佇列管理程式及回覆目的地別名』

本主題說明三種您可以建立遠端佇列定義的方式。

傳輸標頭中的佇列名稱

目的地佇列名稱會與傳輸標頭中的訊息一起傳送，直到到達目的地佇列為止。

佇列管理程式會將應用程式所使用的佇列名稱 (邏輯佇列名稱) 解析為目的地佇列名稱。換句話說，是實體佇列名稱。此目的地佇列名稱會與訊息一起在個別資料區 (傳輸標頭) 中傳送，直到到達目的地佇列為止。然後會除去傳輸標頭。

當您建立平行服務類別時，您可以變更此佇列名稱的佇列管理程式部分。當到達服務程式類別轉移的結尾時，請記得將佇列管理程式名稱傳回原始名稱。

如何建立佇列管理程式及回覆目的地別名

本主題說明三種您可以建立遠端佇列定義的方式。

遠端佇列定義物件以三種不同方式使用。第 25 頁的表 3 說明如何定義三種方式中的每一種：

- 使用遠端佇列定義來重新定義本端佇列名稱。

當開啟佇列時，應用程式只會提供佇列名稱，且此佇列名稱是遠端佇列定義的名稱。

遠端佇列定義包含目標佇列及佇列管理程式的名稱。定義可以選擇性地包含要使用的傳輸佇列名稱。如果未提供傳輸佇列名稱，則佇列管理程式會使用從遠端佇列定義取得的佇列管理程式名稱作為傳輸佇列名稱。如果未定義此名稱的傳輸佇列，但已定義預設傳輸佇列，則會使用預設傳輸佇列。

- 使用遠端佇列定義來重新定義佇列管理程式名稱。

當開啟佇列時，應用程式或通道程式會提供佇列名稱以及遠端佇列管理程式名稱。

如果您已提供與佇列管理程式名稱同名的遠端佇列定義，且將定義中的佇列名稱留白，則佇列管理程式會將開放式呼叫中的佇列管理程式名稱替換為定義中的佇列管理程式名稱。

此外，定義可以包含要使用的傳輸佇列名稱。如果未提供傳輸佇列名稱，則佇列管理程式會採用從遠端佇列定義取得的佇列管理程式名稱作為傳輸佇列名稱。如果未定義此名稱的傳輸佇列，但已定義預設傳輸佇列，則會使用預設傳輸佇列。

- 使用遠端佇列定義來重新定義回覆目的地佇列名稱。

每次應用程式將訊息放入佇列時，它可以提供回答訊息的回覆目的地佇列名稱，但佇列管理程式名稱為空白。

如果您提供與回覆目的地佇列同名的遠端佇列定義，則本端佇列管理程式會將回覆目的地佇列名稱取代為您定義中的佇列名稱。

您可以在定義中提供佇列管理程式名稱，但不能提供傳輸佇列名稱。

使用情形	佇列管理程式名稱	佇列名稱	傳輸佇列名稱
1. 遠端佇列定義 (在 OPEN 呼叫時)			
在通話中提供	空白或本端 QM	(*) 必要	-
在定義中提供	必要	必要	選用
2. 佇列管理程式別名 (在 OPEN 呼叫時)			
在通話中提供	(*) 必要且不是本端 QM	必要	-
在定義中提供	必要	空白	選用
3. 回覆佇列別名 (PUT 呼叫時)			
在通話中提供	空白	(*) 必要	-

表 3: 三種使用遠端佇列定義物件的方式 (繼續)			
使用情形	佇列管理程式名稱	佇列名稱	傳輸佇列名稱
在定義中提供	選用	選用	空白
註: (*) 表示此名稱是定義物件的名稱			

如需正式說明，請參閱 [佇列名稱解析](#)。

將訊息放置在遠端佇列上

您可以使用遠端佇列定義物件，將佇列名稱解析為相鄰佇列管理程式的傳輸佇列。

在分散式佇列環境中，不論訊息是來自本端系統中的應用程式，還是來自相鄰系統的通道，傳輸佇列及通道都是所有訊息到達位置的焦點。第 26 頁的圖 2 顯示應用程式將訊息放置在名為 'QA_norm' 的邏輯佇列上。名稱解析使用遠端佇列定義 'QA_norm' 來選取傳輸佇列 QMB。然後，它會將傳輸標頭新增至訊息，指出 'QA_norm at QMB'。

例如，從 'Channel_back' 上相鄰系統到達的訊息具有實體佇列名稱為 'QA_norm at QMB' 的傳輸標頭。這些訊息在傳輸佇列 QMB 上保持不變。

通道會將訊息移至相鄰的佇列管理程式。

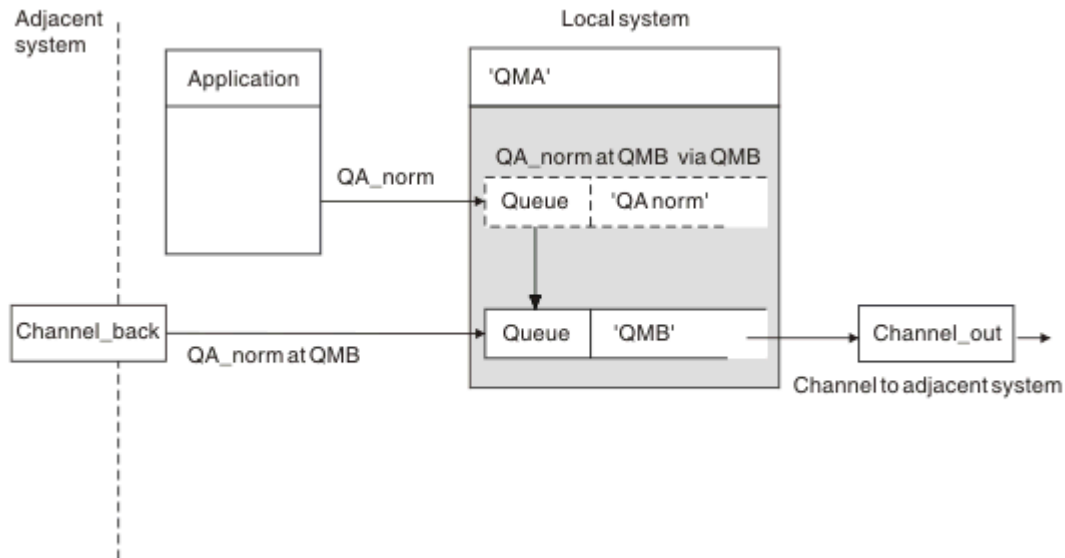


圖 2: 遠端佇列定義用來將佇列名稱解析成相鄰佇列管理程式的傳輸佇列

如果您是 WebSphere MQ 系統管理者，則必須：

- 從相鄰系統定義訊息通道
- 定義相鄰系統的訊息通道
- 建立傳輸佇列 QMB
- 定義遠端佇列物件 'QA_norm'，以將應用程式使用的佇列名稱解析為目的地佇列名稱、目的地佇列管理程式名稱及傳輸佇列名稱

在叢集環境中，您只需要在本端佇列管理程式中定義叢集接收端通道。您不需要定義傳輸佇列或遠端佇列物件。如需相關資訊，請參閱 [叢集](#)。

名稱解析的相關資訊

遠端佇列定義的作用是定義實體目的地佇列名稱及佇列管理程式名稱。這些名稱會放在訊息的傳輸標頭中。

來自相鄰系統的送入訊息已由原始佇列管理程式執行此類型的名稱解析。因此，它們具有顯示實體目的地佇列名稱及佇列管理程式名稱的傳輸標頭。這些訊息不受遠端佇列定義的影響。

選擇傳輸佇列

您可以使用遠端佇列定義，容許不同的傳輸佇列將訊息傳送至相同的相鄰佇列管理程式。

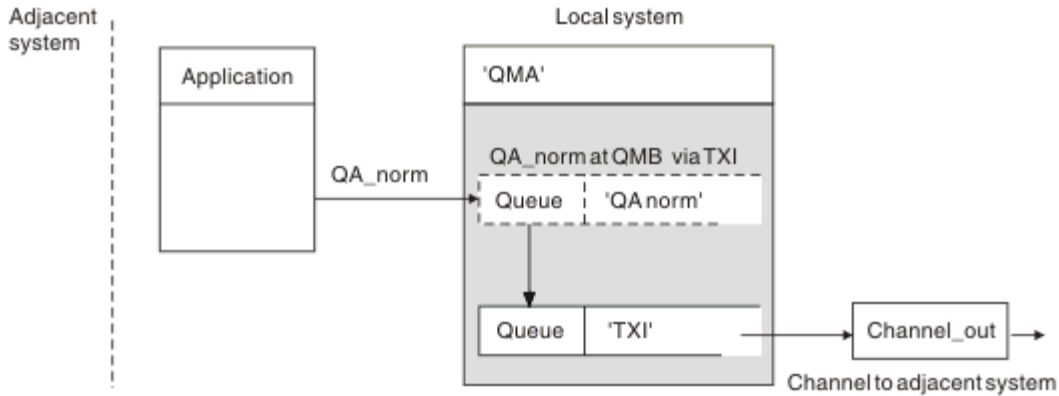


圖 3: 遠端佇列定義容許使用不同的傳輸佇列

在分散式佇列環境中，當您需要將訊息流程從一個通道變更為另一個通道時，請使用第 26 頁的『將訊息放置在遠端佇列上』中的第 26 頁的圖 2 所顯示的相同系統配置。本主題中的第 27 頁的圖 3 顯示如何使用遠端佇列定義，透過不同的傳輸佇列，並因此透過不同的通道，將訊息傳送至相同的相鄰佇列管理程式。

對於第 27 頁的圖 3 中顯示的配置，您必須提供遠端佇列物件 'QA_norm' 及傳輸佇列 'TX1'。您必須提供 'QA_norm'，以選擇遠端佇列管理程式上的 'QA_norm' 佇列、傳輸佇列 'TX1' 及佇列管理程式 'QMB_priority'。在系統旁的通道定義中指定 'TX1'。

訊息放置在傳輸佇列 'TX1' 上，其傳輸標頭包含 'QA_norm at QMB_priority'，並透過通道傳送至相鄰系統。

channel_back 已從這個圖解中遺漏，因為它需要佇列管理程式別名。

在叢集環境中，您不需要定義傳輸佇列或遠端佇列定義。如需相關資訊，請參閱第 139 頁的『叢集佇列數』。

接收訊息

您可以配置佇列管理程式來接收來自其他佇列管理程式的訊息。您必須確保不會發生無意的名稱解析。

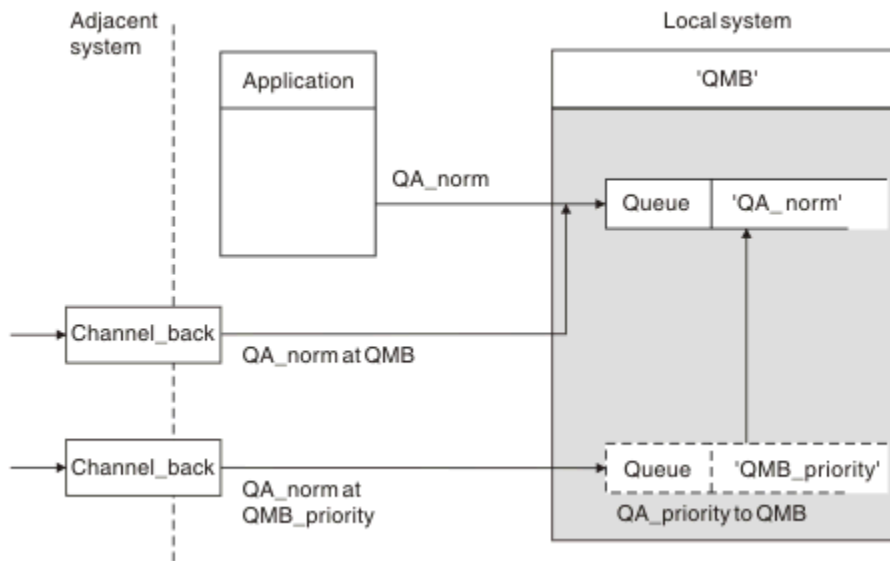


圖 4: 直接接收訊息，並解析別名佇列管理程式名稱

除了安排傳送訊息之外，系統管理者也必須安排從相鄰佇列管理程式接收訊息。接收的訊息在傳輸標頭中包含目的地佇列管理程式及佇列的實體名稱。它們會被視為來自同時指定佇列管理程式名稱及佇列名稱之本端應用程式的訊息。由於這種處理方式，您需要確保進入系統的訊息不會執行無意的名稱解析。如需此實務範例，請參閱第 28 頁的圖 4。

對於此配置，您必須準備：

- 從相鄰佇列管理程式接收訊息的訊息通道
- 佇列管理程式別名定義，用於將送入訊息流程 'QMB_priority' 解析為本端佇列管理程式名稱 'QMB'
- 本端佇列 'QA_norm' (如果不存在)

接收別名佇列管理程式名稱

在此圖解中使用佇列管理程式別名定義未選取不同的目的地佇列管理程式。通過此本端佇列管理程式並定址至 'QMB_priority' 的訊息預期用於佇列管理程式 'QMB'。別名佇列管理程式名稱用來建立個別訊息流程。

透過系統傳遞訊息

您可以用三種方式透過系統傳遞訊息：使用位置名稱、使用佇列管理程式的別名，或選取傳輸佇列。

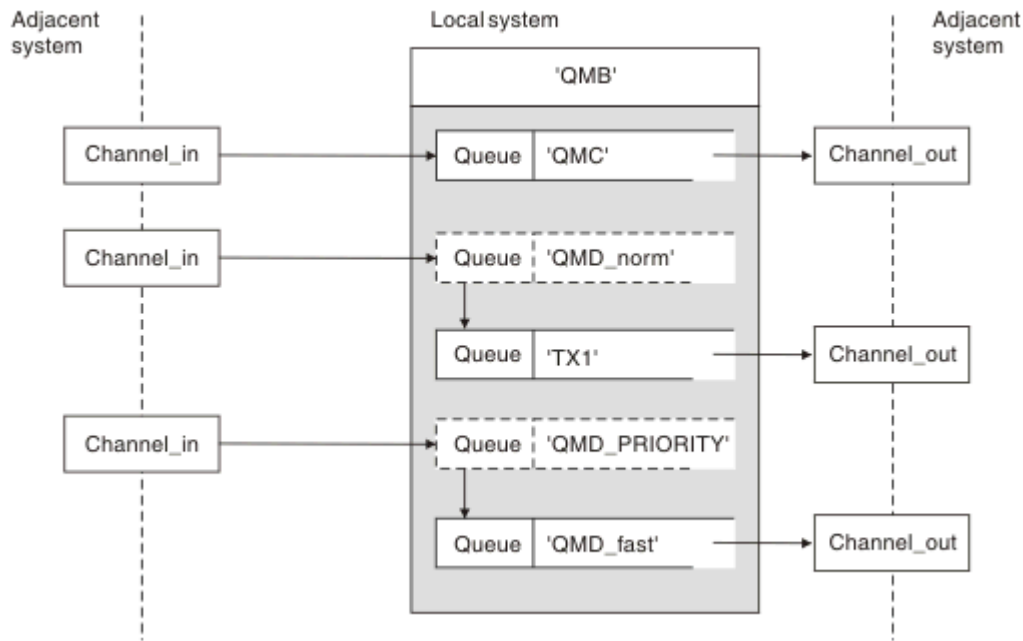


圖 5: 透過系統傳遞訊息的三種方法

第 27 頁的『接收訊息』中第 28 頁的圖 4 所顯示的技術顯示如何擷取別名流程。第 29 頁的圖 5 說明結合先前說明的技術來建立網路的方式。

此配置顯示通道遞送具有不同目的地的三則訊息：

1. QB 於 QMC
2. QB 於 QMD_norm
3. QB 於 QMD_PRIORITY

您必須在系統中傳遞第一個未變更的訊息流程。您必須透過不同的傳輸佇列及通道來傳遞第二個訊息流程。對於第二個訊息流程，您也必須將別名佇列管理程式名稱 QMD_norm 的訊息解析為佇列管理程式 QMD。第三個訊息流程會選擇不同的傳輸佇列，而不會有任何其他變更。

在叢集環境中，訊息會透過叢集傳輸佇列傳遞。一般而言，單一傳輸佇列 SYSTEM.CLUSTER.TRANSMIT.QUEUE 會將所有訊息傳送至佇列管理程式所屬之所有叢集中的所有佇列管理程式；請參閱佇列管理程式的叢集。您可以為佇列管理程式所屬叢集中的所有或部分佇列管理程式定義個別傳輸佇列。

下列方法說明適用於分散式佇列環境的技術。

使用這些方法

對於這些配置，您必須準備：

- 輸入通道定義
- 輸出通道定義
- 傳輸佇列：
 - QMC
 - TX1
 - QMD_fast
- 佇列管理程式別名定義：
 - QMD_norm 搭配 QMD_norm 至 QMD 到 TX1
 - QMD_PRIORITY 搭配 QMD_PRIORITY 至 QMD_PRIORITY 到 QMD_fast

註: 範例中顯示的任何訊息流程都不會變更目的地佇列。佇列管理程式名稱別名提供訊息流程的分隔。

方法 1: 使用送入的位置名稱

您將接收其傳輸標頭包含另一個位置名稱 (例如 QMC) 的訊息。最簡單的配置是建立具有該名稱 QMC 的傳輸佇列。處理傳輸佇列的通道會將未變更的訊息遞送至下一個目的地。

方法 2: 使用佇列管理程式的別名

第二個方法是使用佇列管理程式別名物件定義, 但指定新的位置名稱 QMD 及特定的傳輸佇列 TX1。此動作:

- 終止佇列管理程式名稱別名 QMD_norm(即指名的服務類別 QMD_norm) 所設定的別名訊息流程。
- 將這些訊息上的傳輸標頭從 QMD_norm 變更為 QMD。

方法 3: 選取傳輸佇列

第三個方法是使用與目的地位置 QMD_PRIORITY 相同的名稱來定義佇列管理程式別名物件。使用佇列管理程式別名定義來選取特定的傳輸佇列 QMD_fast, 以及另一個通道。這些訊息上的傳輸標頭保持不變。

區隔訊息流程

您可以使用佇列管理程式別名來建立個別訊息流程, 以將訊息傳送至相同的佇列管理程式。

在分散式佇列環境中, 由於許多原因, 可能需要將訊息分隔至相同的佇列管理程式, 以產生不同的訊息流程。例如:

- 您可能需要為大型、中型及小型訊息提供個別流程。此需求也適用於叢集作業環境, 在此情況下, 您可以建立重疊的叢集。您可以這麼做的原因有很多, 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。例如, 您可以有一個稱為 STANSER 的叢集, 它是稱為 STURES 的叢集子集。當您將訊息放入 STAFF 叢集中所通告的佇列時, 會使用受限通道。當您將訊息放入 STURNES 叢集中所通告的佇列時, 可以使用一般通道或受限通道。
 - 建立測試和正式作業環境。
- 可能需要從本端產生訊息的路徑, 以不同的路徑來遞送送入訊息。
- 您的安裝可能需要排定在特定時間 (例如, 夜間) 移動訊息, 然後訊息需要儲存在保留佇列中, 直到排程為止。

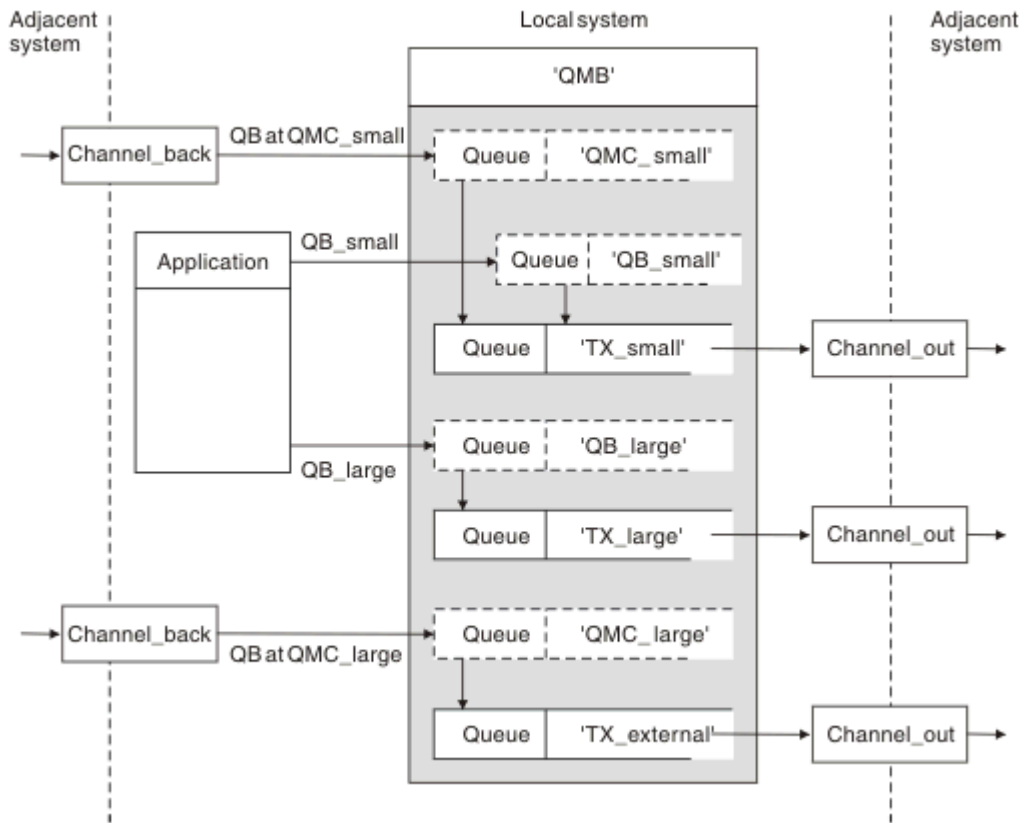


圖 6: 區隔訊息流程

在第 31 頁的圖 6 中顯示的範例中，兩個送入流程是針對別名佇列管理程式名稱 'QMC_small' 及 'QMC_large'。您可以為這些流程提供佇列管理程式別名定義，以擷取本端佇列管理程式的這些流程。您有一個應用程式會處理兩個遠端佇列，且您需要將這些訊息流程分開。您提供兩個遠端佇列定義，指定相同的位置 'QMC'，但指定不同的傳輸佇列。此定義會保持流程分隔，且遠端不需要任何額外的，因為它們在傳輸標頭中具有相同的目的地佇列管理程式名稱。您提供：

- 送入通道定義
- 兩個遠端佇列定義 QB_small 及 QB_large
- 兩個佇列管理程式別名定義 QMC_small 及 QMC_large
- 三個傳送端通道定義
- 三個傳輸佇列: TX_small、TX_large 及 TX_external

與相鄰系統協調

當您使用佇列管理程式別名來建立個別訊息流程時，您需要與訊息通道遠端的系統管理者協調此活動，以確保對應的佇列管理程式別名可在該處使用。

將訊息集中到不同的位置

您可以將送往不同位置的訊息集中到單一通道。

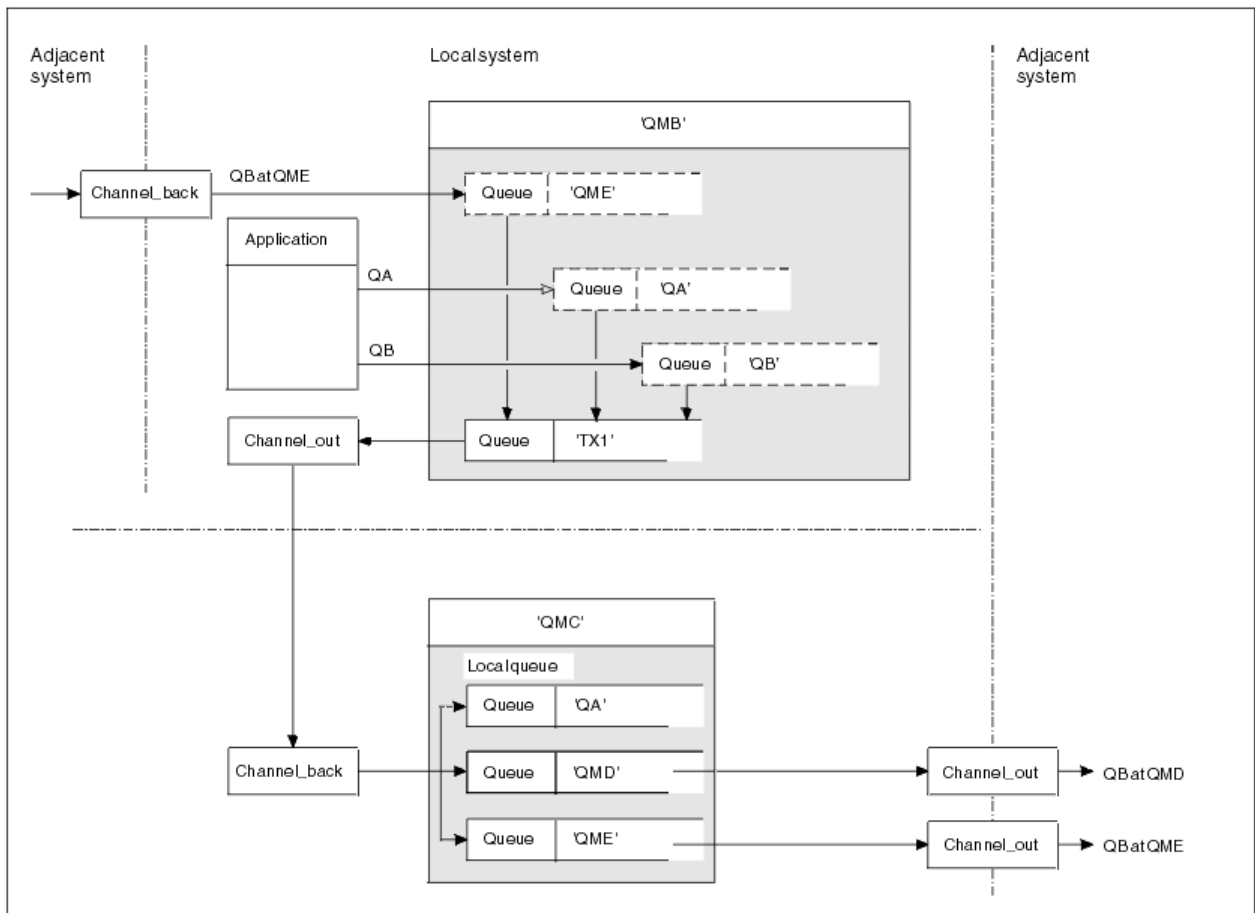


圖 7: 將訊息流程結合至通道

第 32 頁的圖 7 說明將送往不同位置的訊息集中在一個通道上的分散式佇列技術。兩種可能的用途如下：

- 透過閘道集中訊息資料流量
- 在節點之間使用寬頻寬高速公路

在此範例中，來自不同來源 (本端及相鄰且具有不同目的地佇列及佇列管理程式) 的訊息會透過傳輸佇列 'TX1' 傳送至佇列管理程式 QMC。佇列管理程式 QMC 會根據目的地來遞送訊息。一個設定為傳輸佇列 'QMD'，用於向前傳輸至佇列管理程式 QMD。另一個設定為傳輸佇列 'QME'，用於向前傳輸至佇列管理程式 QME。其他訊息會放置在本端佇列 'QA' 上。

您必須提供：

- 通道定義
- 傳輸佇列 TX1
- 遠端佇列定義：
 - 具有 'QA at QMC 到 TX1' 的 QA
 - 具有 'QB at QMD 到 TX1' 的 QB
- 佇列管理程式別名定義：
 - 具有 'QME 到 TX1' 的 QME

配置 QMC 的補充管理者必須提供：

- 具有相同通道名稱的接收端通道定義
- 傳輸佇列 QMD 與相關聯的傳送通道定義
- 傳輸佇列 QME 與相關聯的傳送通道定義

- 本端佇列物件 QA。

將訊息流程轉向至另一個目的地

您可以使用佇列管理程式別名及傳輸佇列來重新定義特定訊息的目的地。

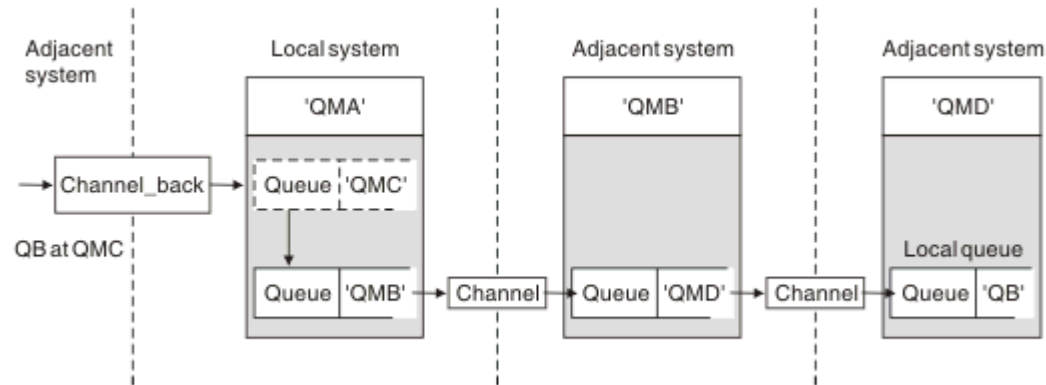


圖 8: 將訊息串流轉向另一個目的地

第 33 頁的圖 8 說明如何重新定義特定訊息的目的地。傳送至 QMA 的送入訊息指定給 QMC 上的 QB。它們通常會到達 QMA，並放置在稱為 QMC 的傳輸佇列上，該佇列已成為 QMC 通道的一部分。QMA 必須將訊息轉移至 QMD，但只能透過 QMB 到達 QMD。當您需要將服務從一個位置移至另一個位置，並容許訂閱者繼續暫時傳送訊息，直到他們調整到新位址為止，此方法非常有用。

將特定佇列管理程式的送入訊息重新遞送至不同佇列管理程式的方法使用：

- 佇列管理程式別名，用於將目的地佇列管理程式變更為另一個佇列管理程式，並選取相鄰系統的傳輸佇列
- 為相鄰佇列管理程式提供服務的傳輸佇列
- 位於相鄰佇列管理程式的傳輸佇列，用於向前遞送至目的地佇列管理程式

您必須提供：

- Channel_back 定義
- 佇列管理程式別名物件定義 QMC (QB 為 QMD 到 QMB)
- 通道輸出定義
- 相關聯的傳輸佇列 QMB

配置 QMB 的補充管理者必須提供：

- 對應的 channel_back 定義
- 傳輸佇列，QMD
- 與 QMD 相關聯的通道定義

您可以在叢集作業環境內使用別名。如需相關資訊，請參閱第 219 頁的『佇列管理程式別名及叢集』。

將訊息傳送至配送清單

您可以使用單一 MQPUT 呼叫，讓應用程式將訊息傳送至數個目的地。

在所有平台 (z/OS 除外) 上的 WebSphere MQ 中，應用程式可以使用單一 MQPUT 呼叫，將訊息傳送至數個目的地。您可以在分散式佇列環境及叢集作業環境中執行此動作。您必須依照 配送清單 中的說明，在配送清單中定義目的地。

並非所有佇列管理程式都支援配送清單。當 MCA 建立與友機的連線時，它會判斷友機是否支援配送清單，並相應地在傳輸佇列上設定旗標。如果應用程式嘗試傳送以配送清單為目的地的訊息，但夥伴不支援配送清單，則傳送端 MCA 會截取該訊息，並針對每一個預期的目的地將它放入傳輸佇列一次。

接收 MCA 可確保在所有預期的目的地安全地接收傳送至配送清單的訊息。如果任何目的地失敗，MCA 會建立哪些目的地失敗。然後，它可以為它們產生異常狀況報告，並可以嘗試重新傳送訊息給它們。

回覆目的地佇列

您可以使用回覆目的地佇列來建立完整遠端佇列處理迴圈。

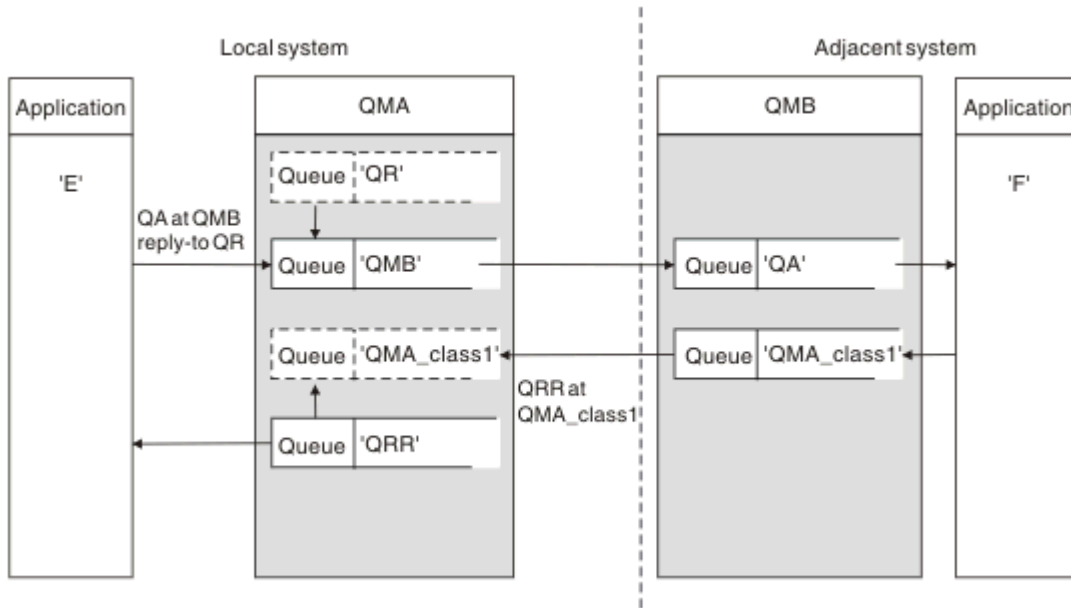


圖 9: PUT 呼叫期間的回覆目的地佇列名稱替代

第 34 頁的圖 9 顯示使用回覆目的地佇列的完整遠端佇列處理迴圈。此迴圈同時適用於分散式佇列環境及叢集作業環境。詳細資料如第 40 頁的表 7 所示。

應用程式會在 QMB 開啟 QA，並將訊息放置在該佇列上。訊息會取得回覆目的地佇列名稱 QR，而不指定佇列管理程式名稱。佇列管理程式 QMA 會尋找回覆目的地佇列物件 QR，並從中擷取別名 QRR 及佇列管理程式名稱 QMA_class1。這些名稱會放入訊息的回覆目的地欄位中。

來自 QMB 應用程式的回覆訊息會定址至 QMA_class1 中的 QRR。佇列管理程式會使用佇列管理程式別名定義 QMA_class1，將訊息傳送至佇列本身及佇列 QRR。

此實務範例描述您為應用程式提供選擇回覆訊息的服務類別之機能的方式。此類別由 QMB 的傳輸佇列 QMA_class1 與 QMA 的佇列管理程式別名定義 QMA_class1 實作。以此方式，您可以變更應用程式的回覆目的地佇列，以便隔離流程而不涉及應用程式。應用程式一律選擇此特定服務類別的 QR。您有機會使用回覆目的地佇列定義 QR 來變更服務類別。

您必須建立：

- 回覆目的地佇列定義 QR
- 傳輸佇列物件 QMB
- 通道輸出定義
- Channel_back 定義
- 佇列管理程式別名定義 QMA_class1
- 本端佇列物件 QRR (如果不存在)

相鄰系統上的補充管理者必須建立：

- 接收端通道定義
- 傳輸佇列物件 QMA_class1
- 相關聯的傳送端通道
- 本端佇列物件 QA。

您的應用程式使用：

- 放置呼叫中的回覆目的地佇列名稱 QR

- 取得呼叫中的佇列名稱 QRR

如此一來，您可以視需要變更服務類別，而不涉及應用程式。您將回覆目的地別名 'QR' 與傳輸佇列 'QMA_class1' 及佇列管理程式別名 'QMA_class1' 一起變更。

如果在佇列上放置訊息時找不到回覆目的地別名物件，則會在空白回覆目的地佇列管理程式名稱欄位中插入本端佇列管理程式名稱。回覆目的地佇列名稱保持不變。

名稱解析限制

因為在放置原始訊息時，已在 'QMA' 執行回覆目的地佇列的名稱解析，所以在 'QMB' 不容許進一步的名稱解析。回覆應用程式會以回覆目的地佇列的實體名稱來放置訊息。

應用程式必須知道它們用於回覆目的地佇列的名稱與要在其中找到回覆訊息的實際佇列名稱不同。

例如，為使用回覆目的地佇列別名為 'C1_alias' 及 'C2_alias' 的應用程式提供兩個服務類別時，應用程式會在訊息放置呼叫中使用這些名稱作為回覆目的地佇列名稱。不過，應用程式實際上預期訊息會出現在佇列 'C1' (對於 'C1_alias') 及 'C2' (對於 'C2_alias')。

不過，應用程式能夠對回覆別名佇列進行查詢呼叫，以檢查其本身是否有取得回覆訊息必須使用的實際佇列名稱。

相關概念

第 25 頁的『如何建立佇列管理程式及回覆目的地別名』
本主題說明三種您可以建立遠端佇列定義的方式。

第 35 頁的『回覆目的地佇列別名範例』

此範例說明如何使用回覆別名，為傳回的訊息選取不同的路徑 (傳輸佇列)。使用此機能需要與應用程式合作變更回覆目的地佇列名稱。

第 37 頁的『範例如何運作』

範例的說明，以及佇列管理程式如何使用回覆目的地佇列別名。

第 37 頁的『回覆目的地佇列別名逐步演練』

程序的逐步演練，從應用程式將訊息放置在遠端佇列，直到從別名回覆目的地佇列中移除回覆訊息的相同應用程式。

回覆目的地佇列別名範例

此範例說明如何使用回覆別名，為傳回的訊息選取不同的路徑 (傳輸佇列)。使用此機能需要與應用程式合作變更回覆目的地佇列名稱。

如 第 35 頁的圖 10 所示，回覆路徑必須可用於回覆訊息，包括傳輸佇列、通道及佇列管理程式別名。

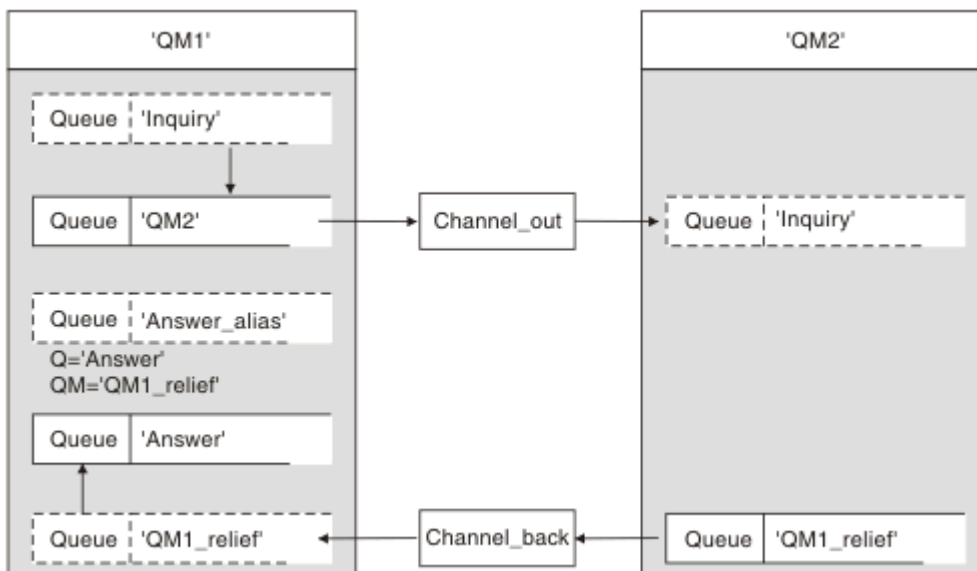


圖 10: 回覆目的地佇列別名範例

此範例適用於位於 'QM1' 的要求者應用程式，它會將訊息傳送至位於 'QM2' 的伺服器應用程式。伺服器上的訊息將透過使用傳輸佇列 'QM1_relief' 的替代通道傳回 (預設傳回通道將使用傳輸佇列 'QM1' 提供服務)。

回覆目的地佇列別名是遠端佇列定義 'Answer_alias' 的特定用法。位於 QM1 的應用程式在其放入佇列「查詢」的所有訊息的回覆目的地欄位中包括此名稱 'Answer_alias'。

回覆目的地佇列定義 'Answer_alias' 定義為 'Answer at QM1_relief'。位於 QM1 的應用程式預期其回覆會出現在名為 'Answer' 的本端佇列中。

QM2 中的伺服器應用程式使用已接收訊息的回覆目的地欄位，以取得 QM1 中傳送至要求端之回覆訊息的佇列及佇列管理程式名稱。

此範例中在 QM1 中使用的定義

位於 QM1 的 WebSphere MQ 系統管理者必須確保回覆目的地佇列「回答」與其他物件一起建立。以 '*' 標示的佇列管理程式別名名稱必須與回覆目的地佇列別名定義中的佇列管理程式名稱一致，也必須以 '*' 標示。

物件	定義	
本端傳輸佇列	QM2	
遠端佇列定義	物件名稱	查詢
	遠端佇列管理程式名稱	QM2
	遠端佇列名稱	查詢
	傳輸佇列名稱	QM2 (DEFAULT)
佇列管理程式別名	物件名稱	QM1_relief *
	佇列管理程式名稱	QM1
	佇列名稱	(空白)
回覆目的地佇列別名	物件名稱	Answer_alias
	遠端佇列管理程式名稱	QM1_relief *
	遠端佇列名稱	答案

將定義放置在 QM1

應用程式會以回覆目的地佇列別名填入回覆目的地欄位，並將佇列管理程式名稱欄位留白。

欄位	內容
佇列名稱	查詢
佇列管理程式名稱	(空白)
回覆目的地佇列名稱	Answer_alias
回覆目的地佇列管理程式	(空白)

此範例中在 QM2 中使用的定義

QM2 的 WebSphere MQ 系統管理者必須確定送入訊息的本端佇列存在，且正確命名的傳輸佇列可供回覆訊息使用。

物件	定義
本端佇列	查詢
傳輸佇列	QM1_relief

將定義放置在 QM2

位於 QM2 的應用程式會從原始訊息中擷取回覆目的地佇列名稱及佇列管理程式名稱，並在將回覆訊息放入回覆目的地佇列時使用它們。

欄位	內容
佇列名稱	答案
佇列管理程式名稱	QM1_relief

範例如何運作

範例的說明，以及佇列管理程式如何使用回覆目的地佇列別名。

在此範例中，QM1 中的要求者應用程式一律使用 'Answer_alias' 作為 put 呼叫相關欄位中的回覆目的地佇列。他們一律從名為 'Answer' 的佇列中擷取其訊息。

回覆目的地佇列別名定義可供 QM1 系統管理者使用，以變更回覆目的地佇列 'Answer' 及傳回路徑 'QM1_relief' 的名稱。

變更佇列名稱 'Answer' 通常沒有用，因為 QM1 應用程式在此佇列中預期其回答。不過，QM1 系統管理者可以視需要變更返回路徑 (服務類別)。

佇列管理程式如何使用回覆目的地佇列別名

當回覆目的地佇列名稱 (由應用程式併入放置呼叫中) 與回覆目的地佇列別名相同且佇列管理程式組件空白時，佇列管理程式 QM1 會從回覆目的地佇列別名擷取定義。

佇列管理程式會將 put 呼叫中的回覆目的地佇列名稱取代為定義中的佇列名稱。它會將放置呼叫中的空白佇列管理程式名稱取代為定義中的佇列管理程式名稱。

這些名稱會隨訊息描述子中的訊息一起傳送。

欄位名稱	放置呼叫	傳輸標頭
回覆目的地佇列名稱	Answer_alias	答案
回覆目的地佇列管理程式名稱	(空白)	QM1_relief

回覆目的地佇列別名逐步演練

程序的逐步演練，從應用程式將訊息放置在遠端佇列，直到從別名回覆目的地佇列中移除回覆訊息的相同應用程式。

若要完成此範例，讓我們查看處理程序。

1. 應用程式會開啟名為 'Inquiry' 的佇列，並將訊息放入其中。應用程式會將訊息描述子的回覆目的地欄位設為：

回覆目的地佇列名稱	Answer_alias
回覆目的地佇列管理程式名稱	(空白)

2. 佇列管理程式 'QM1' 會檢查名為 'Answer_alias' 的遠端佇列定義，以回應空白佇列管理程式名稱。如果找不到，佇列管理程式會在訊息描述子的回覆目的地佇列管理程式欄位中放置自己的名稱 'QM1'。
3. 如果佇列管理程式找到名為 'Answer_alias' 的遠端佇列定義，則會從定義中擷取佇列名稱及佇列管理程式名稱 (佇列名稱 = 'Answer' 及佇列管理程式名稱 = 'QM1_relief')。然後，它會將它們放入訊息描述子的 reply-to 欄位中。
4. 佇列管理程式 'QM1' 使用遠端佇列定義 'Inquiry' 來判定預期的目的地佇列位於佇列管理程式 'QM2'，且訊息放置在傳輸佇列 'QM2' 上。'QM2' 是目的地為佇列管理程式 'QM2' 中佇列之訊息的預設傳輸佇列名稱。
5. 當佇列管理程式 'QM1' 將訊息放入傳輸佇列時，它會將傳輸標頭新增至訊息。此標頭包含目的地佇列 'Inquiry' 及目的地佇列管理程式 'QM2' 的名稱。

6. 訊息到達佇列管理程式 'QM2'，並放置在「查詢」本端佇列上。
7. 應用程式從此佇列中取得訊息並處理訊息。應用程式會準備回覆訊息，並從原始訊息的訊息描述子將此回覆訊息放置在回覆目的地佇列名稱上：

回覆目的地佇列名稱

答案

回覆目的地佇列管理程式名稱

QM1_relief

8. 佇列管理程式 'QM2' 會執行 put 指令。發現佇列管理程式名稱 'QM1_relief' 是遠端佇列管理程式，它會將訊息放置在具有相同名稱 'QM1_relief' 的傳輸佇列上。訊息會獲得傳輸標頭，其中包含目的地佇列 'Answer' 及目的地佇列管理程式 'QM1_relief' 的名稱。
9. 訊息會傳送至佇列管理程式 'QM1'。佇列管理程式會辨識佇列管理程式名稱 'QM1_relief' 是別名，並從別名定義 'QM1_relief' 中擷取實體佇列管理程式名稱 'QM1'。
10. 然後，佇列管理程式 'QM1' 會將訊息放置在傳輸標頭 'Answer' 中包含的佇列名稱上。
11. 應用程式從佇列 'Answer' 擷取其回覆訊息。

網路考量

在分散式佇列環境中，由於訊息目的地只會以佇列名稱和佇列管理程式名稱來定址，因此會套用某些規則。

1. 其中提供佇列管理程式名稱，且名稱不同於本端佇列管理程式名稱：

- 傳輸佇列必須具有相同的名稱。此傳輸佇列必須是將訊息移至另一個佇列管理程式的訊息通道的一部分，或
- 佇列管理程式別名定義必須存在，才能將佇列管理程式名稱解析成相同或另一個佇列管理程式名稱，以及選用的傳輸佇列，或
- 如果無法解析傳輸佇列名稱，且已定義預設傳輸佇列，則會使用預設傳輸佇列。

2. 如果只提供佇列名稱，則本端佇列管理程式上必須提供任何類型但具有相同名稱的佇列。此佇列可以是遠端佇列定義，其解析為：相鄰佇列管理程式的傳輸佇列、佇列管理程式名稱，以及選用的傳輸佇列。

若要查看如何在叢集作業環境中運作，請參閱產品說明文件的 [叢集如何運作](#) 一節中的適當主題。

請考量在分散式佇列環境中，訊息通道將訊息從一個佇列管理程式移至另一個佇列管理程式的情況。

正在移動的訊息源自網路中的任何其他佇列管理程式，部分訊息可能到達目的地時具有不明佇列管理程式名稱。例如，當佇列管理程式名稱已變更或已從系統中移除時，即會發生此問題。

通道程式會在找不到這些訊息的傳輸佇列時辨識此狀況，並將訊息放置在無法遞送的訊息 (無法傳送郵件) 佇列上。您有責任尋找這些訊息，並安排將它們轉遞至正確的目的地。或者，將它們傳回給發送端，其中可以確定發送端。

在這些情況下，如果在原始訊息中要求報告訊息，則會產生異常狀況報告。

名稱解析慣例

名稱解析會變更目的地佇列的身分 (亦即，邏輯對實體名稱變更)，只會發生一次，且只會在原始佇列管理程式上發生。

只有在分隔及結合訊息流程時，才必須使用後續使用各種可能的別名。

退貨遞送

訊息可以包含佇列及佇列管理程式名稱形式的回覆位址。此傳回位址表單可以在分散式佇列環境及叢集環境中使用。

此位址通常由建立訊息的應用程式指定。然後，任何處理訊息的應用程式 (包括使用者結束程式應用程式) 都可以修改它。

不論此位址的來源為何，任何處理訊息的應用程式都可以選擇使用此位址，將回答、狀態或報告訊息傳回給原始應用程式。

遞送這些回應訊息的方式與遞送原始訊息的方式沒有不同。您需要注意，您建立給其他佇列管理程式的訊息流程需要對應的傳回流程。

實體名稱衝突

目的地回覆目的地佇列名稱已解析為原始佇列管理程式上的實體佇列名稱。不得在回應佇列管理程式中再次解析它。

名稱衝突問題的可能性只能透過全網路的實體及邏輯佇列名稱協議來防止。

管理佇列名稱轉換

當您建立佇列管理程式別名定義或遠端佇列定義時，每個包含該名稱的訊息都會執行名稱解析。必須管理此狀況。

此說明提供給應用程式設計者及通道規劃者，其與具有相鄰系統之訊息通道的個別系統相關。它採用通道規劃及控制的本端視圖。

當您建立佇列管理程式別名定義或遠端佇列定義時，不論訊息來源為何，都會針對每個包含該名稱的訊息執行名稱解析。若要監督此狀況(可能涉及佇列管理程式網路中的大量佇列)，您可以保留下列表格：

- 與已解析佇列名稱、已解析佇列管理程式名稱及已解析傳輸佇列名稱相關的來源佇列及來源佇列管理程式名稱，具有解析方法
- 與下列相關的來源佇列名稱：
 - 已解析目的地佇列名稱
 - 已解析目的地佇列管理程式名稱
 - 傳輸佇列
 - 訊息通道名稱
 - 相鄰系統名稱
 - 回覆目的地佇列名稱

註：在此環境定義中使用 來源 一詞，是指應用程式所提供的佇列名稱或佇列管理程式名稱，或是在開啟佇列以放置訊息時的通道程式。

其中每一個表格的範例顯示在 [第 39 頁的表 5](#)、[第 39 頁的表 6](#) 及 [第 40 頁的表 7](#) 中。

這些表格中的名稱衍生自本節中的範例，而此表格並非預期作為一個節點中佇列名稱解析的實際範例。

開啟佇列時指定的來源佇列	開啟佇列時指定的來源佇列管理程式	解析的佇列名稱	解析的佇列管理程式名稱	已解析傳輸佇列名稱	解析類型
QA_norm	-	QA_norm	QMB	QMB	遠端佇列
(任何)	QMB	-	-	QMB	(無)
QA_norm	-	QA_norm	QMB	TX1	遠端佇列
QB	QMC	QB	QMD	QMB	佇列管理程式別名

開啟佇列時指定的來源佇列	開啟佇列時指定的來源佇列管理程式	解析的佇列名稱	解析的佇列管理程式名稱	已解析傳輸佇列名稱	解析類型
QA_norm	-	QA_norm	QMB	-	(無)
QA_norm	QMB	QA_norm	QMB	-	(無)
QA_norm	QMB_優先順序	QA_norm	QMB	-	佇列管理程式別名
(任何)	QMC	(任何)	QMC	QMC	(無)

開啟佇列時指定的來源佇列	開啟佇列時指定的來源佇列管理程式	解析的佇列名稱	解析的佇列管理程式名稱	已解析傳輸佇列名稱	解析類型
(任何)	QMD_norm	(任何)	QMD_norm	TX1	佇列管理程式別名
(任何)	QMD_PRIORITY	(任何)	QMD_PRIORITY	QMD_fast	佇列管理程式別名
(任何)	QMC_small	(任何)	QMC_small	TX_small	佇列管理程式別名
(任何)	QMC_large	(任何)	QMC_large	TX_external	佇列管理程式別名
QB 小型	QMC	QB 小型	QMC	TX_small	遠端佇列
QB_large	QMC	QB_large	QMC	TX_large	遠端佇列
(任何)	QME	(任何)	QME	TX1	佇列管理程式別名
QA	QMC	QA	QMC	TX1	遠端佇列
QB	QMD	QB	QMD	TX1	遠端佇列

應用程式設計		回覆別名定義	
本端 QMGR	訊息的佇列名稱	回覆目的地佇列別名	重新定義為
QMA	QRR	QR	QRR 位於 QMA_class1

通道訊息序號

通道會使用序號來確保訊息遞送、遞送而不重複，並以從傳輸佇列取得訊息的相同順序儲存。

序號會在通道傳送端產生，並在使用之前增加一個序號，這表示現行序號是前次傳送的訊息號碼。可以使用 DISPLAY CHSTATUS 來顯示此資訊 (請參閱 MQSC 參照)。序號和稱為 LUWID 的 ID 會儲存在持續儲存體中，以批次傳送最後一則訊息。在通道啟動期間會使用這些值，以確保鏈結兩端同意哪些訊息已順利傳送。

循序擷取訊息

如果應用程式將一連串訊息放入相同的目的地佇列，且符合下列條件，則單一應用程式可以依序擷取這些訊息，並具有一連串 MQGET 作業：

- 所有放置要求都是從相同的應用程式完成。
- 所有放置要求都來自相同的工作單元，或所有放置要求都是在工作單元之外提出。
- 這些訊息都具有相同的優先順序。
- 這些訊息都具有相同的持續性。
- 對於遠端佇列作業，配置是從提出放置要求的應用程式，透過其佇列管理程式，透過交互通訊，到目的地佇列管理程式及目標佇列，只能有一個路徑。
- 訊息不會放入無法傳送郵件的佇列 (例如，如果佇列暫時已滿)。
- 取得訊息的應用程式不會故意變更擷取順序，例如透過指定特定 *MsgId* 或 *CorrelId* 或使用訊息優先順序。
- 只有一個應用程式正在執行取得作業，以從目的地佇列擷取訊息。如果有多個應用程式，這些應用程式必須設計成取得傳送端應用程式所放置的每一個序列中的所有訊息。

註： 來自其他作業及工作單元的訊息可能會與序列混雜在一起，即使序列是從單一工作單元內放置。

如果無法符合這些條件，且目標佇列上的訊息順序很重要，則可以將應用程式編碼為使用其自己的訊息序號作為訊息的一部分，以確保訊息的順序。

快速、非持續訊息的擷取順序

快速通道上的非持續訊息可能會超過相同通道上的持續訊息，因此會超出順序。接收端 MCA 會立即將非持續訊息放置在目的地佇列上，並使其可見。直到下一個同步點才會顯示持續訊息。

迴圈測試

Loopback 測試是在非 z/OS 平台上的一種技術，可讓您測試通訊鏈結，而無需實際鏈結至另一部機器。

您可以設定兩個佇列管理程式之間的連線，就像它們位於個別機器上一樣，但您可以透過迴圈回到相同機器上的另一個處理程序來測試連線。此技術表示您可以在不需要作用中網路的情況下測試通訊碼。

您這樣做的方式取決於您使用的產品和通訊協定。

在 Windows 系統上，您可以使用「迴圈」配接器。

如需相關資訊，請參閱您正在使用之產品的說明文件。

遞送追蹤和活動記錄

您可以兩種方式來確認訊息在一系列佇列管理程式中所採用的路徑。

您可以使用透過控制指令 dspmqrte 提供的 WebSphere MQ 顯示路徑應用程式，也可以使用活動記錄。這兩個主題都在 [監視參照](#) 中說明。

分散式佇列管理簡介

分散式佇列管理 (DQM) 用來定義及控制佇列管理程式之間的通訊。

分散式佇列管理：


- 可讓您定義及控制佇列管理程式之間的通訊通道
- 提供訊息通道服務，可讓您將訊息從本端佇列類型 (稱為傳輸佇列) 移至本端系統上的通訊鏈結，以及從通訊鏈結移至目的地佇列管理程式上的本端佇列
- 提供您使用畫面、指令及程式來監視通道作業及診斷問題的機能

通道定義會將通道名稱與傳輸佇列、通訊鏈結 ID 及通道屬性相關聯。通道定義在不同平台上以不同方式實作。訊息傳送及接收由稱為訊息通道代理程式 (MCA) 的程式所控制，這些程式使用通道定義來啟動及控制通訊。

然後，MCA 由 DQM 本身控制。此結構與平台相關，但通常包括接聽器和觸發監視器，以及操作員指令和畫面。

訊息通道是將訊息從一個佇列管理程式移至另一個佇列管理程式的單向管道。因此，訊息通道具有兩個端點，由一對 MCA 代表。每一個端點都有其訊息通道結尾的定義。例如，一端定義傳送端，另一端定義接收端。

如需如何定義通道的詳細資料，請參閱：

-  [第 63 頁的『在 UNIX, Linux, and Windows 上監視及控制通道』](#)

如需訊息通道規劃範例，請參閱：

-  [分散式平台的訊息通道規劃範例](#)

如需通道結束程式的相關資訊，請參閱 [傳訊通道的通道結束程式](#)。

相關概念

[第 42 頁的『訊息傳送及接收』](#)

下圖顯示分散式佇列管理模型，詳述傳輸訊息時實體之間的關係。它也會顯示控制項的流程。

[第 46 頁的『通道控制功能』](#)

通道控制功能可讓您定義、監視及控制通道。

[第 56 頁的『無法遞送訊息時會發生什麼情況?』](#)

當無法遞送訊息時，MCA 可以數種方式來處理它。它可以重試，它可以傳回給傳送端，也可以將它放在無法傳送郵件的佇列中。

第 60 頁的『起始設定和配置檔』

通道起始設定資料的處理取決於 WebSphere MQ 平台。

第 61 頁的『訊息的資料轉換』

在不同佇列管理程式上的佇列之間傳送 WebSphere MQ 訊息時，可能需要進行資料轉換。

第 61 頁的『撰寫您自己的訊息通道代理程式』

WebSphere MQ 可讓您撰寫自己的訊息通道代理程式 (MCA) 程式，或安裝來自獨立軟體供應商的訊息通道代理程式。

第 61 頁的『分散式佇列管理要考量的其他事項』

準備 WebSphere MQ 以進行分散式佇列管理時要考量的其他主題。本主題涵蓋「未遞送訊息佇列」、「使用中佇列」、「系統延伸及使用者結束程式」，以及「作為授信應用程式執行通道及接聽器」。

相關參考

配置資訊範例

訊息傳送及接收

下圖顯示分散式佇列管理模型，詳述傳輸訊息時實體之間的關係。它也會顯示控制項的流程。

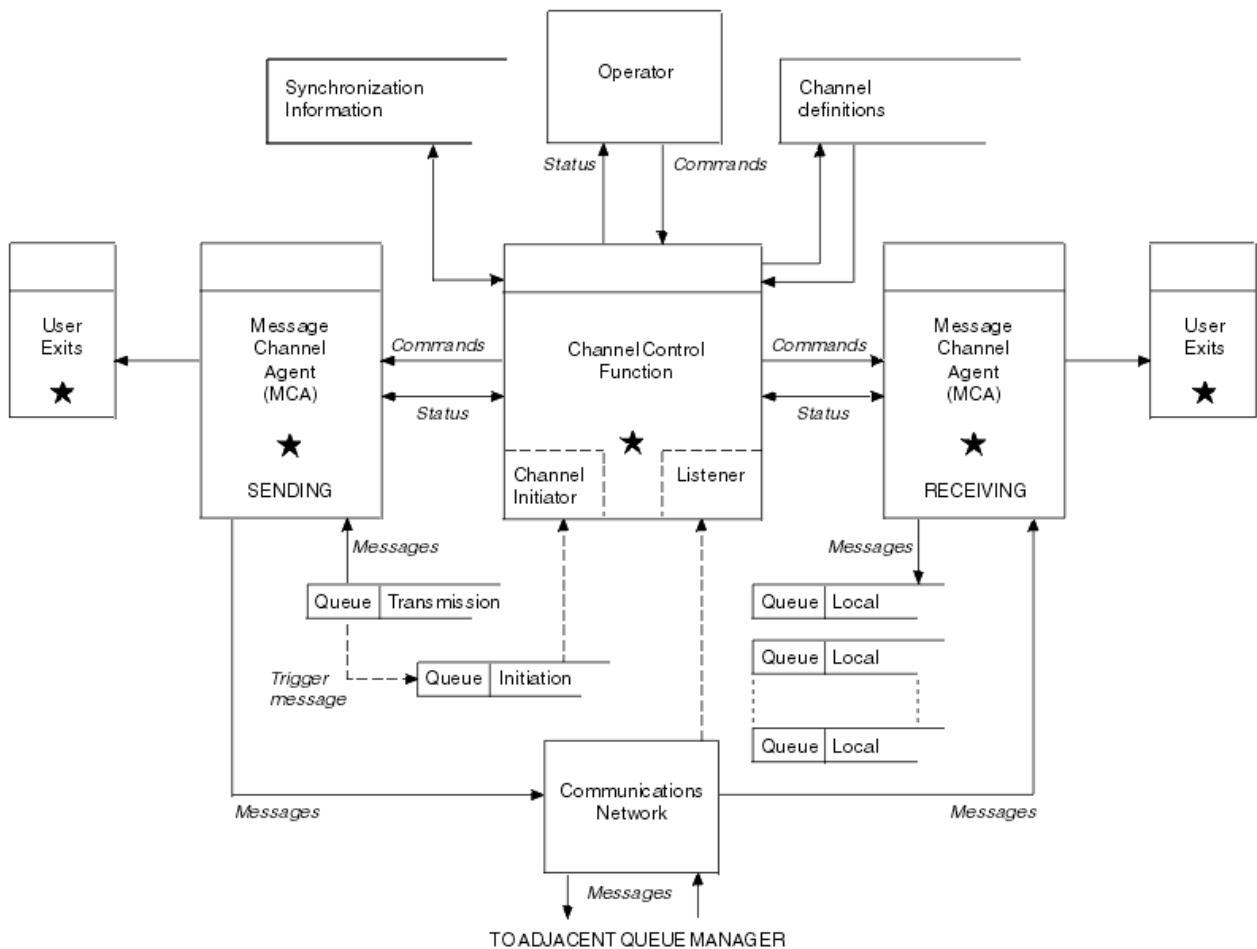


圖 11: 分散式佇列管理模型

註:

1. 視平台而定，每個通道都有一個 MCA。特定佇列管理程式可能有一或多個通道控制功能。
2. MCA 及通道控制功能的實作高度依賴於平台。它們可以是程式或程序或執行緒，也可以是單一實體或多個包含數個獨立或鏈結組件的實體。

3. 標示星號的所有元件都可以使用 MQI。

通道參數

MCA 會以下列數種方式之一來接收其參數：

- 如果由指令啟動，則會在資料區中傳遞通道名稱。然後 MCA 會直接讀取通道定義，以取得其屬性。
- 對於傳送端，以及在某些情況下，佇列管理程式觸發程式可以自動啟動 MCA。通道名稱是從觸發程式程序定義中擷取(如果適用的話)，並傳遞至 MCA。其餘處理與先前所說明的相同。只有在伺服器通道是完整的(亦即，指定要連接的 CONNAME)時，才必須將伺服器通道設定為觸發。
- 如果由傳送端、伺服器、要求端或用戶端連線從遠端啟動，則通道名稱會從友機訊息通道代理程式傳入起始資料。MCA 會直接讀取通道定義以取得其屬性。

通道定義中未定義的某些屬性也可協議：

分割訊息

如果一端不支援分割訊息，則不會傳送分割訊息。

轉換功能

如果一端無法在需要時執行必要的字碼頁轉換或數值編碼轉換，則另一端必須處理它。如果任一端都不支援它，則在必要時，通道無法啟動。

配送清單支援

如果一端不支援配送清單，則夥伴 MCA 會在其傳輸佇列中設定旗標，以便它知道截取預期用於多個目的地的訊息。

通道狀態及序號

訊息通道代理程式會保留每一個通道現行序號及邏輯工作單元號碼的記錄，以及通道一般狀態的記錄。部分平台可讓您顯示此狀態資訊，以協助您控制通道。

如何將訊息傳送至另一個佇列管理程式

本節說明在佇列管理程式之間傳送訊息的最簡單方法，包括所需的必備項目及授權。其他方法也可以用來將訊息傳送至遠端佇列管理程式。

在將訊息從一個佇列管理程式傳送至另一個佇列管理程式之前，您需要執行下列步驟：

1. 請檢查您選擇的通訊協定是否可用。
2. 啟動佇列管理程式。
3. 啟動通道起始程式。
4. 啟動接聽器。

您也需要具有正確的 WebSphere MQ 安全授權，才能建立所需的物件。

如果要將訊息從一個佇列管理程式傳送至另一個佇列管理程式，請執行下列動作：

- 在來源佇列管理程式上定義下列物件：
 - 傳送端通道
 - 遠端佇列定義
 - 起始佇列(選用項目)
 - 傳輸佇列
 - 無法傳送郵件的佇列
- 在目標佇列管理程式上定義下列物件：
 - 接收端通道
 - 目標佇列
 - 無法傳送郵件的佇列

視 WebSphere MQ 平台而定，您可以使用數種不同的方法來定義這些物件：

- 在所有平台上，您可以使用 WebSphere MQ Script 指令 (MQSC)，如 [MQSC 指令 自動化管理作業](#) 中所說明的可程式化指令格式 (PCF) 指令，或「WebSphere MQ 探險家」中所說明。

如需建立元件以將訊息傳送至另一個佇列管理程式的相關資訊，請參閱下列子主題：

相關概念

[第 16 頁的『建立及管理佇列管理程式』](#)

您必須先建立並啟動至少一個佇列管理程式及其關聯物件，然後才能使用訊息及佇列。

[第 24 頁的『IBM WebSphere MQ 分散式傳訊技術』](#)

本節中的子主題說明規劃通道時使用的技術。這些子主題說明技術，可協助您規劃如何將佇列管理程式連接在一起，以及管理應用程式之間的訊息流程。

[第 41 頁的『分散式佇列管理簡介』](#)

分散式佇列管理 (DQM) 用來定義及控制佇列管理程式之間的通訊。

[第 58 頁的『觸發通道』](#)

WebSphere MQ 提供一項機能，可在符合佇列上的特定條件時自動啟動應用程式。此機能稱為觸發。

[第 56 頁的『訊息安全』](#)

除了 WebSphere MQ 的一般回復特性之外，分散式佇列管理還可確保使用訊息通道兩端之間協調的同步點程序來適當地遞送訊息。如果此程序偵測到錯誤，則會關閉通道，以便您可以調查問題，並安全地將訊息保留在傳輸佇列中，直到通道重新啟動為止。

[第 63 頁的『在 UNIX, Linux, and Windows 上監視及控制通道』](#)

對於 DQM，您需要建立、監視及控制遠端佇列管理程式的通道。您可以使用指令、程式、IBM WebSphere MQ Explorer、通道定義的檔案，以及同步化資訊的儲存區來控制通道。

[第 84 頁的『配置用戶端與伺服器之間的連線』](#)

若要配置 WebSphere MQ MQI 用戶端與伺服器之間的通訊鏈結，請決定您的通訊協定，定義鏈結兩端的連線，啟動接聽器，以及定義通道。

相關工作

[第 133 頁的『配置佇列管理程式叢集』](#)

請使用本主題中的鏈結，以瞭解叢集如何運作、如何設計叢集配置，以及取得如何設定簡式叢集的範例。

定義通道

若要將訊息從一個佇列管理程式傳送至另一個佇列管理程式，您必須定義兩個通道。您必須在來源佇列管理程式上定義一個通道，並在目標佇列管理程式上定義一個通道。

在來源佇列管理程式上

定義通道類型為 SENDER 的通道。您需要指定下列項目：

- 要使用的傳輸佇列名稱 (XMITQ 屬性)。
- 友機系統的連線名稱 (CONNNAME 屬性)。
- 您使用的通訊協定名稱 (TRPTYPE 屬性)。在 WebSphere MQ for z/OS 上，通訊協定必須是 TCP 或 LU6.2。在其他平台上，您不需要指定這個。您可以讓它從預設通道定義中挑選值。

所有通道屬性的詳細資料都在 [通道屬性](#) 中提供。

在目標佇列管理程式上

定義通道類型為 RECEIPT 且名稱與傳送端通道相同的通道。

指定您使用的通訊協定名稱 (TRPTYPE 屬性)。在 WebSphere MQ for z/OS 上，通訊協定必須是 TCP 或 LU6.2。在其他平台上，您不需要指定這個。您可以讓它從預設通道定義中挑選值。

接收端通道定義可以是通用的。這表示如果您有數個佇列管理程式與相同的接收端進行通訊，則傳送端通道可以全部指定相同的接收端名稱，且一個接收端定義適用於它們全部。

註：回應訊息通道代理程式會忽略 TRPTYPE 參數值。例如，傳送端通道定義上 TCP 的 TRPTYPE 會以接收端通道定義上 LU62 的 TRPTYPE 順利啟動作為友機。

當您已定義通道時，可以使用 PING CHANNEL 指令來測試它。這個指令會將特殊訊息從傳送端通道傳送至接收端通道，並檢查是否傳回。

定義佇列

若要將訊息從一個佇列管理程式傳送至另一個佇列管理程式，您必須定義最多六個佇列。您必須在來源佇列管理程式上定義最多四個佇列，並在目標佇列管理程式上定義最多兩個佇列。

在來源佇列管理程式上

- 遠端佇列定義

在此定義中，指定下列：

遠端佇列管理程式名稱

目標佇列管理程式的名稱。

遠端佇列名稱

目標佇列管理程式上的目標佇列名稱。

傳輸佇列名稱

傳輸佇列的名稱。您不需要指定此傳輸佇列名稱。如果沒有，則會使用與目標佇列管理程式同名的傳輸佇列。如果不存在，則會使用預設傳輸佇列。建議您為傳輸佇列提供與目標佇列管理程式相同的名稱，以便依預設找到佇列。

- 起始佇列定義

在 z/OS 上是必要項目，在其他平台上是選用項目。請考慮命名起始佇列 SYSTEM.CHANNEL.INITQ。在其他平台上。

- 傳輸佇列定義

USAGE 屬性設為 XMITQ 的本端佇列。

- 無法傳送郵件的佇列定義

定義可寫入未遞送訊息的無法傳送郵件佇列。

在目標佇列管理程式上

- 本端佇列定義

目標佇列。此佇列的名稱必須與來源佇列管理程式上遠端佇列定義的遠端佇列名稱欄位中指定的名稱相同。

- 無法傳送郵件的佇列定義

定義可寫入未遞送訊息的無法傳送郵件佇列。

相關概念

第 45 頁的『建立傳輸佇列』

在可以啟動通道 (非要求端通道) 之前，必須依照本節的說明來定義傳輸佇列。傳輸佇列必須在通道定義中命名。

建立傳輸佇列

在可以啟動通道 (非要求端通道) 之前，必須依照本節的說明來定義傳輸佇列。傳輸佇列必須在通道定義中命名。

針對每一個傳送訊息通道，定義 USAGE 屬性設為 XMITQ 的本端佇列。如果您要在遠端佇列定義中使用特定的傳輸佇列，請建立所顯示的遠端佇列。

若要建立傳輸佇列，請使用 WebSphere MQ 指令 (MQSC)，如下列範例所示：

建立傳輸佇列範例

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

建立遠端佇列範例

```
DEFINE QREMOTE(PAYROLL) DESCR('Remote queue for QM2') +  
XMITQ(QM2) RNAME(PAYROLL) RQMNAME(QM2)
```

請考慮將傳輸佇列命名為遠端系統上的佇列管理程式名稱，如範例所示。

啟動通道

當您將訊息放置在來源佇列管理程式所定義的遠端佇列上時，訊息會儲存在傳輸佇列中，直到通道啟動為止。當通道已啟動時，訊息會遞送至遠端佇列管理程式上的目標佇列。

使用 `START CHANNEL` 指令在傳送端佇列管理程式上啟動通道。當您啟動傳送端通道時，接收端通道會自動啟動 (由接聽器)，並將訊息傳送至目標佇列。訊息通道的兩端必須在執行中，才能傳送訊息。

因為通道兩端位於不同的佇列管理程式上，所以可以使用不同的屬性來定義它們。為了解決任何差異，當通道啟動時，兩端之間會有起始資料協議。一般而言，通道兩端使用需要較少資源的屬性來運作。這可讓較大的系統在訊息通道的另一端容納較小系統的較小資源。

傳送端 MCA 會先分割大型訊息，然後再透過通道傳送它們。它們會在遠端佇列管理程式中重新組合。這對使用者而言並不明顯。

MCA 可以使用多個執行緒來傳送訊息。此處理程序稱為管線化，可讓 MCA 以較少等待狀態更有效率地傳送訊息。管線化可改善通道效能。

通道控制功能

通道控制功能可讓您定義、監視及控制通道。

指令是透過畫面、程式或從指令行發出至通道控制功能。畫面介面也會顯示通道狀態及通道定義資料。您可以使用「可程式指令格式」或第 63 頁的『[在 UNIX, Linux, and Windows 上監視及控制通道](#)』中詳述的 WebSphere MQ 指令 (MQSC) 及控制指令。

這些指令屬於下列群組：

- 通道管理
- 通道控制
- 通道狀態監視

通道管理指令會處理通道的定義。它們可讓您：

- 建立通道定義
- 複製通道定義
- 變更通道定義
- 刪除通道定義

通道控制指令會管理通道的作業。它們可讓您：

- 啟動通道
- 停止通道
- 與夥伴重新同步化 (在某些實作中)
- 重設訊息序號
- 解決不確定的訊息批次
- 連線測試; 透過通道傳送測試通訊

通道監視會顯示通道的狀態，例如：

- 現行通道設定
- 通道是作用中還是非作用中
- 通道是否在已同步狀態下終止

如需定義、控制及監視通道的相關資訊，請參閱下列子主題：

準備通道

在嘗試啟動訊息通道或 MQI 通道之前，您必須先準備通道。您必須確定本端及遠端通道定義的所有屬性都正確且相容。

[通道屬性](#) 說明通道定義及屬性。

雖然您設定明確通道定義，但通道啟動時所執行的通道協議可能會置換其中一個或其他定義的值。此行為是正常的，對使用者來說並不明顯，且已以這種方式來安排，因此其他不相容的定義可以一起運作。

自動定義接收端和伺服器連線通道

在 z/OS 以外所有平台上的 WebSphere MQ 中，如果沒有適當的通道定義，則對於已啟用自動定義的接收端或伺服器連線通道，會自動建立定義。使用下列方式來建立定義：

1. 適當的模型通道定義 SYSTEM.AUTO.RECEIVER 或 SYSTEM.AUTO.SVRCONN。自動定義的模型通道定義與系統預設值 SYSTEM.DEF.RECEIVER 和 SYSTEM.DEF.SVRCONN，但說明欄位除外，它是「自動定義者」，後面接著 49 個空白。系統管理者可以選擇變更所提供模型通道定義的任何部分。
2. 來自夥伴系統的資訊。來自友機的值用於通道名稱及序號折返。
3. 通道結束程式，可用來變更自動定義所建立的值。請參閱 [通道自動定義結束程式 \(Channel auto-definition exit program\)](#)。

然後會檢查說明，以判定它是否已由自動定義結束程式變更，或因為模型定義已變更。如果前 44 個字元仍然是「自動定義」，後面接著 29 個空白，則會新增佇列管理程式名稱。如果最後 20 個字元仍全為空白，則會新增當地時間和日期。

當已建立並儲存定義時，通道啟動會繼續進行，如同定義一直存在一樣。與夥伴協議批次大小、傳輸大小及訊息大小。

定義其他物件

在可以啟動訊息通道之前，必須在其佇列管理程式中定義 (或啟用自動定義) 兩端。它要提供的傳輸佇列必須在傳送端定義給佇列管理程式。通訊鏈結必須已定義且可用。您可能需要準備其他 WebSphere MQ 物件 (例如遠端佇列定義、佇列管理程式別名定義及回覆目的地佇列別名定義)，以實作 [第 23 頁的『使用分散式佇列連接應用程式』](#) 中說明的實務範例。

如需定義 MQI 通道的相關資訊，請參閱 [第 95 頁的『定義 MQI 通道』](#)。

每個傳輸佇列有多個訊息通道

每個傳輸佇列可以定義多個通道，但每次只能有一個通道處於作用中狀態。請考量此選項，在佇列管理程式之間供應替代路徑以進行資料流量平衡及鏈結失敗更正動作。如果要使用的前一個通道已終止，則另一個通道無法使用傳輸佇列，在傳送端留下一批不確定的訊息。如需相關資訊，請參閱 [第 54 頁的『不確定的通道』](#)。

啟動通道

通道可能以四種方式之一開始傳輸訊息。它可能是：

- 由操作員啟動 (非接收端、叢集接收端或伺服器連線通道)。
- 從傳輸佇列觸發。此方法僅適用於傳送端通道及完整伺服器通道 (那些指定 CONNAME 的通道)。您必須準備觸發通道的必要物件。
- 從應用程式 (非接收端、叢集接收端或伺服器連線通道) 啟動。
- 由傳送端、叢集傳送端、要求端、伺服器或用戶端連線通道從遠端啟動。接收端、叢集接收端以及可能的伺服器和要求端通道傳輸會以這種方式啟動；伺服器連線通道也會以這種方式啟動。通道本身必須已啟動 (亦即，已啟用)。

註：因為通道「已啟動」，所以不一定會傳輸訊息。相反地，當發生先前說明的四個事件之一時，可能會「啟用」以開始傳輸。可使用 START 及 STOP 操作員指令來啟用及停用通道。

通道狀態

通道隨時可以處於許多狀態之一。部分狀態也有子狀態。從給定狀態，通道可以移至其他狀態。

第 48 頁的圖 12 顯示所有可能通道狀態的階層，以及套用至每一個通道狀態的子狀態。

第 49 頁的圖 13 顯示通道狀態之間的鏈結。這些鏈結適用於所有類型的訊息通道及伺服器連線通道。

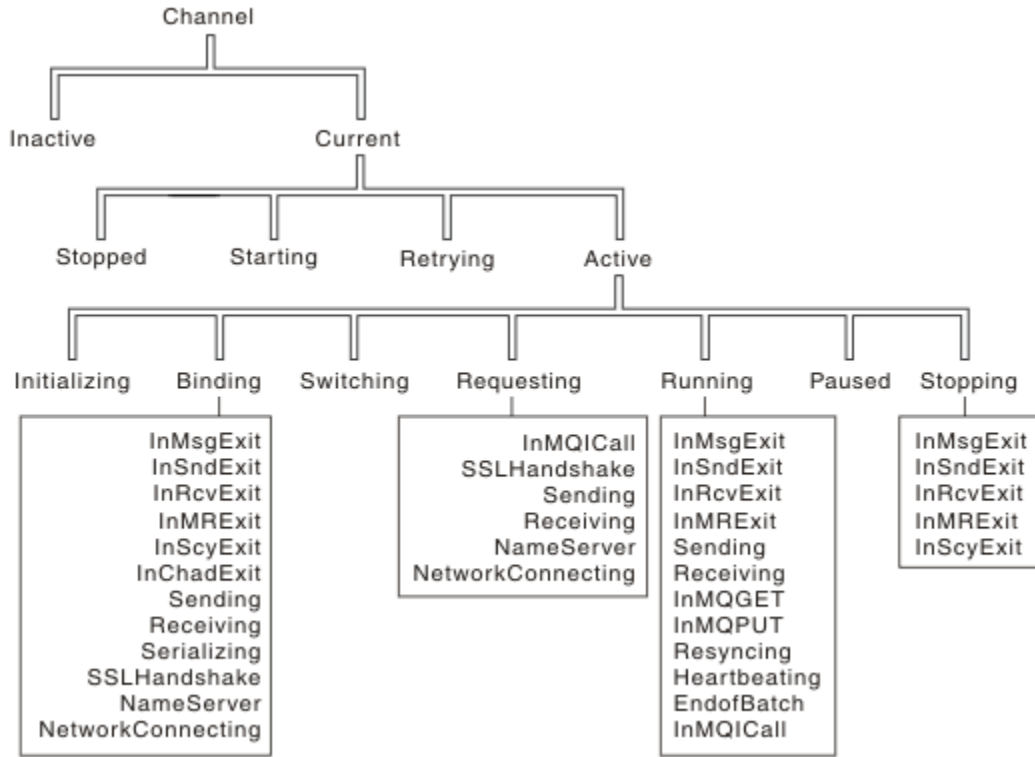


圖 12: 通道狀態及子狀態

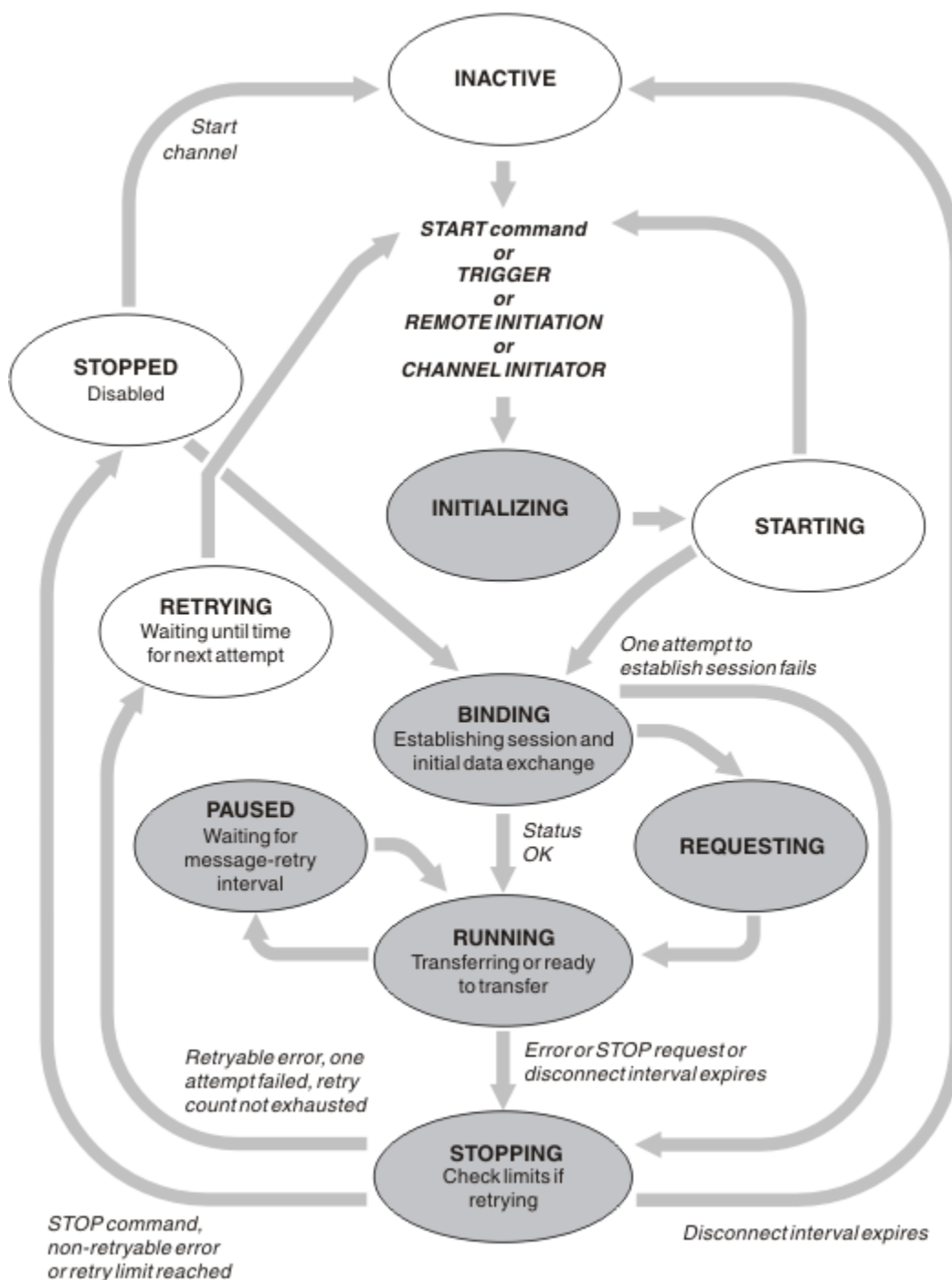


圖 13: 通道狀態之間的流程

現行及作用中

如果通道處於非作用中以外的任何狀態，則通道是現行。除非現行通道處於「重試中」、「已停止」或「啟動中」狀態，否則它是作用中。當通道處於作用中狀態時，它會耗用資源，且處理程序或執行緒正在執行中。在 [第 49 頁的圖 13](#) 中強調顯示作用中通道 (INITIALIZING、BINDING、SWITCHING、REQUESTING、RUNNING、PAUSED 或 STOPPING) 的七個可能狀態。

作用中通道也可以顯示子狀態，以提供通道確切執行的詳細資料。每一個狀態的子狀態會顯示在 [第 48 頁的圖 12](#) 中。

現行及作用中

如果通道處於非作用中以外的任何狀態，則通道是 "現行"。除非現行通道處於「重試中」、「已停止」或「啟動中」狀態，否則它是 "作用中"。

如果通道處於「作用中」狀態，則也可能顯示子狀態，以提供通道確切執行的詳細資料。

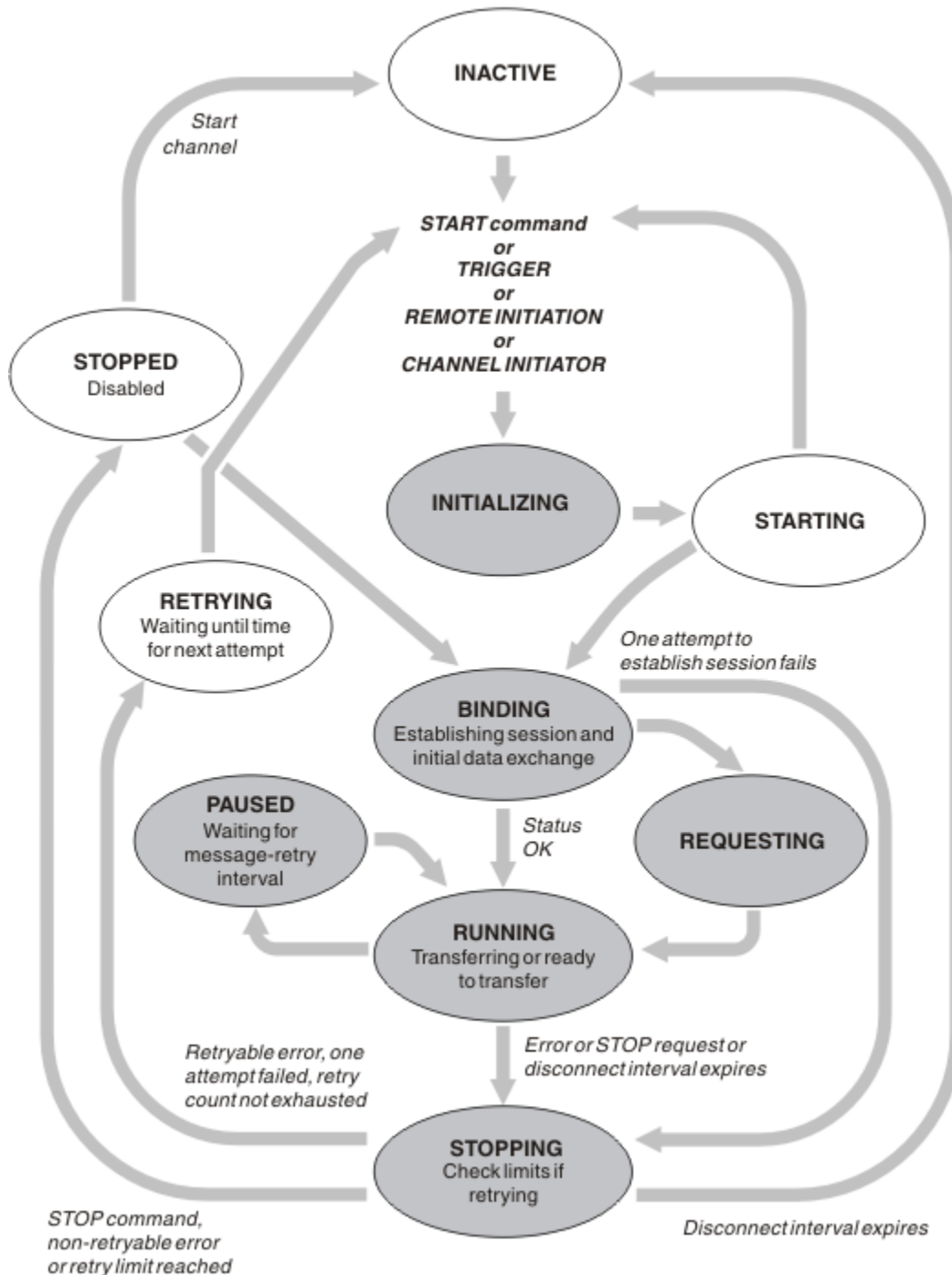


圖 14: 通道狀態之間的流程

註:

1. 當通道處於第 50 頁的圖 14 中強調顯示的六種狀態 (INITIALIZING、BINDING、REQUESTING、RUNNING、PAUSED 或 STOPPING) 之一時，它會耗用資源，且處理程序或執行緒正在執行中；通道是作用中。
2. 當通道處於 STOPPED 狀態時，階段作業可能處於作用中狀態，因為下一個狀態不明。

指定現行通道數目上限

您可以指定一次可以是現行通道數上限。此數目是通道狀態表格中具有項目的通道數目，包括重試中的通道及已停止的通道。請使用 UNIX and Linux 系統的佇列管理程式配置檔，或「WebSphere MQ 探險家」來指定此選項。如需使用起始設定或配置檔所設定值的相關資訊，請參閱 [分散式佇列的配置檔段落](#)。如需指定通道數目上限的相關資訊，請參閱 [管理 IBM WebSphere MQ for WebSphere MQ for UNIX and Linux 系統及 Windows 系統](#)。

註：

1. 這個數目包含伺服器連線通道。
2. 通道必須是現行通道，才能變成作用中。如果通道已啟動，但無法變成現行通道，則啟動會失敗。

指定作用中通道數目上限

您也可以指定作用中通道數目上限，以防止系統因許多起始通道而超載。如果您使用此方法，請將斷線間隔屬性設為低值，以容許等待通道在其他通道終止時立即啟動。

每次重試的通道嘗試建立與其友機的連線時，它必須變成作用中通道。如果嘗試失敗，則它會保持非作用中的現行通道，直到下次嘗試時為止。通道重試的次數，以及重試頻率，由重試次數和重試間隔通道屬性決定。這兩個屬性都有短值和長值。如需相關資訊，請參閱 [通道屬性](#)。

當通道必須變成作用中通道時（因為已發出 START 指令，或因為已觸發它，或因為現在是另一次重試嘗試的時間），但由於作用中通道的數目已達到最大值而無法這樣做時，通道會等到另一個停止作用中的通道實例釋放其中一個作用中槽為止。不過，如果通道因為在遠端起始而啟動中，且當時沒有可用的作用中插槽，則會拒絕遠端起始。

每當通道（非要求端通道）嘗試變成作用中，它會進入 STARTING 狀態。即使有立即可用的作用中插槽，也會發生此狀態，雖然它只在短時間內處於 STARTING 狀態。不過，如果通道必須等待作用中的插槽，則在等待時它會處於 STARTING 狀態。

要求端通道不會進入 STARTING 狀態。如果要求端通道因為作用中通道數已達到限制而無法啟動，則通道會異常結束。

每當通道（非要求端通道）無法取得作用中插槽時，就會將訊息寫入日誌，並產生事件。稍後釋放空位且通道能夠獲得它時，會產生另一個訊息及事件。如果通道能夠立即獲得空位，則不會產生任何這些事件及訊息。

如果在通道等待變成作用中時發出 STOP CHANNEL 指令，則通道會進入 STOPPED 狀態。引發「通道-已停止」事件。

作用中通道數上限包含伺服器連線通道。

如需指定作用中通道數目上限的相關資訊，請參閱 [管理 IBM WebSphere MQ for WebSphere MQ for UNIX and Linux 系統及 Windows 系統](#)。

通道錯誤

通道上的錯誤會導致通道停止進一步傳輸。如果通道是傳送端或伺服器，則會進入 RETRY 狀態，因為問題可能自行清除。如果無法進入 RETRY 狀態，通道會進入 STOPPED 狀態。

對於傳送通道，相關聯的傳輸佇列會設為 GET (DISABLED)，並關閉觸發。(具有 STATUS (STOPPED) 的 STOP 指令會使發出它的一端進入 STOPPED 狀態；只有斷線間隔到期，或具有 STATUS (INACTIVE) 的 STOP 指令會使它正常結束並變成非作用中。) 處於 STOPPED 狀態的通道需要操作員介入才能重新啟動 (請參閱 [第 54 頁的『重新啟動已停止的通道』](#))。

註：若為 UNIX、Linux 及 Windows 系統，通道起始程式必須在執行中，才能嘗試重試。如果通道起始程式無法使用，則通道會變成非作用中，且必須手動重新啟動。如果您使用 Script 來啟動通道，請確保通道起始程式正在執行中，然後再嘗試執行 Script。

[長重試次數 \(LONGRTY\)](#) 說明重試的運作方式。如果錯誤清除，通道會自動重新啟動，且會重新啟用傳輸佇列。如果達到重試限制而未清除錯誤，則通道會進入 STOPPED 狀態。操作員必須手動重新啟動已停止的通道。如果錯誤仍然存在，則不會重試。當它順利啟動時，會重新啟用傳輸佇列。

當通道處於「重試中」或「已停止」狀態時，如果佇列管理程式停止，則重新啟動佇列管理程式時，會記住通道狀態。不過，如果佇列管理程式在通道處於 STOPPED 狀態時停止，則 SVRCONN 通道類型的通道狀態會重設。

如果通道因為佇列已滿或禁止放置而無法將訊息放置到目標佇列，則通道可以在某個時間間隔 (在訊息重試間隔屬性中指定) 重試作業數次 (在訊息重試次數屬性中指定)。或者，您可以撰寫自己的訊息重試結束程式，以判斷哪些情況會導致重試，以及嘗試的次數。等待訊息重試間隔完成時，通道會進入 PAUSED 狀態。

如需通道屬性的相關資訊，請參閱 [通道屬性](#)，以及 [傳訊通道的通道結束程式](#)，以取得訊息重試結束程式的相關資訊。

伺服器連線通道限制

您可以設定伺服器連線通道限制，以防止用戶端應用程式耗盡佇列管理程式通道資源 **MAXINST**，並防止單一用戶端應用程式耗盡伺服器連線通道容量 **MAXINSTC**。

在個別佇列管理程式上隨時可以處於作用中的通道總數上限。作用中通道數目上限包含伺服器連線通道實例總數。

如果您未指定可啟動的伺服器連線通道同時實例數上限，則單一用戶端應用程式 (連接至單一伺服器連線通道) 可能會耗盡可用的作用中通道數上限。當達到作用中通道數上限時，它會阻止在佇列管理程式上啟動任何其他通道。若要避免此狀況，您必須限制可同時啟動個別伺服器連線通道的實例數，而不論哪個用戶端啟動它們。

如果限制值減少到低於伺服器連線通道目前執行中的實例數，即使是零，則不會影響執行中通道。除非已停止執行足夠的現有實例，否則無法啟動新的實例，因此目前執行中的實例數小於限制值。

此外，許多不同的用戶端連線通道可以連接至個別伺服器連線通道。個別伺服器連線通道可同時啟動的實例數限制 (不論哪個用戶端啟動它們)，可防止任何用戶端耗盡佇列管理程式的作用中通道容量上限。如果您未同時限制可從個別用戶端啟動之個別伺服器連線通道的同時實例數，則單一錯誤用戶端應用程式可能會開啟如此多連線，而耗盡配置給個別伺服器連線通道的通道容量，因此阻止其他需要使用該通道的用戶端連接至該通道。若要避免此狀況，您必須限制可從個別用戶端啟動之個別伺服器連線通道的同時實例數。

如果個別用戶端限制的值低於目前從個別用戶端執行的伺服器連線通道實例數 (即使是零)，則不會影響執行中通道。不過，伺服器連線通道的新實例無法從超出新限制的個別用戶端啟動，直到該用戶端的足夠現有實例停止執行，使得目前執行中的實例數小於此參數的值為止。

正在檢查通道的另一端是否仍然可用

您可以使用活動訊號間隔、保持作用中間隔及接收逾時，來檢查通道的另一端是否可用。

活動訊號

您可以使用活動訊號間隔通道屬性來指定當傳輸佇列上沒有訊息時，要從傳送端 MCA 傳遞流程，如 [活動訊號間隔 \(HBINT\)](#) 中所述。

保持作用中

在 WebSphere MQ for UNIX、Linux 及 Windows 系統中，如果您使用 TCP 作為傳輸通訊協定，則可以設定 `keepalive=yes`。如果您指定此選項，TCP 會定期檢查連線的另一端是否仍然可用。不是，通道已終止。此選項在 [保持作用中間隔 \(KAINT\)](#) 中有說明。

如果您有不可靠的通道報告 TCP 錯誤，則使用 **Keepalive** 選項表示您的通道更有可能回復。

您可以指定時間間隔來控制 **Keepalive** 選項的行為。當您變更時間間隔時，只會影響在變更之後啟動的 TCP/IP 通道。請確定您選擇的時間間隔值小於通道的斷線間隔值。

如需使用 **Keepalive** 選項的相關資訊，請參閱 [DEFINE CHANNEL](#) 指令中的 [KAINT](#) 參數。

接收逾時

如果您使用 TCP 作為傳輸通訊協定，則在一段時間內未收到任何資料時，也會關閉閒置非 MQI 通道連線的接收端。此期間接收逾時值是根據 HBINT (活動訊號間隔) 值來決定。

在 WebSphere MQ for UNIX、Linux 及 Windows 系統中，接收逾時值設定如下：

1. 對於起始流程數，在進行任何協議之前，接收逾時值是通道定義中 HBINT 值的兩倍。
2. 在通道協議 HBINT 值之後，如果 HBINT 設為小於 60 秒，則接收逾時值會設為此值的兩倍。如果 HBINT 設為 60 秒以上，則接收逾時值會設為大於 HBINT 值 60 秒。

註：

1. 如果任一值為零，則沒有逾時。
2. 對於不支援活動訊號的連線，HBINT 值在步驟 2 中協議為零，因此沒有逾時，因此您必須使用 TCP/IP KEEPALIVE。
3. 對於使用共用交談的用戶端連線，活動訊號可以一直在通道中（從兩端）流動，而不只是在 MQGET 未完成時。
4. 對於未使用共用交談的用戶端連線，只有在用戶端發出具有等待的 MQGET 呼叫時，才會從伺服器傳送活動訊號。因此，不建議您將用戶端通道的活動訊號間隔設得太小。例如，如果活動訊號設為 10 秒，則 MQCMIT 呼叫會失敗（具有 MQRC_CONNECTION_BROKEN），因為在此期間未傳送任何資料。這可能發生在大型工作單元上。不過，如果針對活動訊號間隔選擇適當的值，則不會發生此情況，因為只有具有等待的 MQGET 才會花費大量時間。

提供的 SHARECNV 不是零，用戶端會使用全雙工連線，這表示用戶端在所有 MQI 呼叫期間可以（且會）活動訊號

5. 在 WebSphere MQ 第 7 版用戶端通道中，活動訊號可以從伺服器及用戶端傳送。任一端的逾時基於 $2 \times \text{HBINT}$ （對於少於 60 秒的 HBINT），以及 $\text{HBINT} + 60$ （對於超過 60 秒的 HBINT）。
6. 在活動訊號間隔兩次之後取消連線是有效的，因為預期資料或活動訊號流程至少在每個活動訊號間隔。不過，將活動訊號間隔設得太小可能會造成問題，尤其是當您使用通道結束程式時。比方說，如果 HBINT 值為一秒，且使用傳送或接收結束程式，則接收端只會等待 2 秒，然後再取消通道。如果 MCA 正在執行諸如加密訊息之類的作業，則此值可能太短。

採用 MCA

「採用 MCA」功能可讓 IBM WebSphere MQ Explorer 取消接收端通道，並在其位置啟動新的通道。

如果通道發生通訊失敗，則接收端通道可能處於「通訊接收」狀態。重新建立通訊時，傳送端通道會嘗試重新連接。如果遠端佇列管理程式發現接收端通道已在執行中，則不容許啟動相同接收端通道的另一個版本。此問題需要使用者人為介入來更正問題或使用系統保留作用中。

採用 MCA 功能可自動解決問題。它可讓 IBM WebSphere MQ Explorer 取消接收端通道，並在其位置啟動新的通道。

可以使用各種選項來設定此功能。 **distributed** 若為分散式平台，請參閱 [管理](#)。

停止及靜止通道

本主題說明如何在斷線時間間隔到期之前停止及靜止通道。

訊息通道設計為佇列管理程式之間的長時間執行連線，只有中斷連線間隔通道屬性才會控制循序終止。除非操作員需要在斷線時間間隔到期之前終止通道，否則此機制運作良好。在下列狀況中可能發生此需要：

- 系統靜止
- 資源節省
- 通道一端的單方面動作

在此情況下，您可以停止通道。您可以使用下列方式來執行此動作：

- STOP CHANNEL MQSC 指令
- 「停止通道 PCF」指令
- IBM WebSphere MQ 檔案總管

使用這些指令來停止通道有三個選項：

QUIESCE

QUIESCE 選項會在停止通道之前嘗試結束現行訊息批次。

強制

FORCE 選項會嘗試立即停止通道，而且在通道重新啟動時可能需要通道重新同步化，因為通道可能處於不確定的狀態。

TERMINATE

TERMINATE 選項會嘗試立即停止通道，並終止通道的執行緒或處理程序。

所有這些選項都會讓通道處於 STOPPED 狀態，需要操作員介入來重新啟動它。

在傳送端停止通道是有效的，但需要操作員介入才能重新啟動。在通道的接收端，情況會更困難，因為 MCA 正在等待來自傳送端的資料，且無法從接收端起始通道的依序終止；停止指令會擱置到 MCA 從等待資料中返回為止。

因此，根據所需的作業性質，有三種建議使用通道的方式：

- 如果您想要通道長時間執行，請注意只能從傳送端依序終止。當通道岔斷 (即已停止) 時，需要操作員介入 (START CHANNEL 指令) 才能重新啟動它們。
- 如果您希望通道只有在有訊息要傳輸時才處於作用中狀態，請將斷線間隔設為相當低的值。預設值為高，因此不建議用於需要此控制層次的通道。因為很難岔斷接收通道，所以最經濟的選項是讓通道根據工作量需求自動斷線並重新連接。對於大部分通道，可以探索性地建立斷線間隔的適當設定。
- 您可以使用活動訊號間隔屬性，讓傳送端 MCA 在沒有訊息可傳送的期間，將活動訊號流程傳送至接收端 MCA。此動作會釋放接收 MCA 的等待狀態，並讓它有機會靜止通道，而不等待斷線間隔到期。請提供低於斷線間隔值的活動訊號間隔值。

註：

1. 建議您將伺服器通道的斷線間隔設為低值，或使用活動訊號。此低值是在沒有伺服器通道要傳送的訊息時，容許要求端通道異常結束 (例如，因為通道已取消) 的情況。如果斷線間隔設為高，且活動訊號不在使用中，則伺服器不會偵測要求端已結束 (它只會在下次嘗試傳送訊息給要求端時才會執行)。當伺服器仍在執行中時，它會將傳輸佇列保持開啟以供專用輸入，以便取得任何其他到達佇列的訊息。如果嘗試從要求端重新啟動通道，則啟動要求會收到錯誤，因為伺服器仍開啟傳輸佇列以進行專用輸入。必須停止伺服器通道，然後重新從要求端重新啟動通道。

重新啟動已停止的通道

當通道進入 STOPPED 狀態時，您必須手動重新啟動通道。

若要重新啟動通道，請發出下列其中一個指令：

- START CHANNEL MQSC 指令
- 「啟動通道 PCF」指令
- IBM WebSphere MQ 檔案總管

對於傳送端或伺服器通道，當通道進入 STOPPED 狀態時，會將相關聯的傳輸佇列設為 GET (DISABLED)，並將觸發設為 off。當收到啟動要求時，會自動重設這些屬性。

如果佇列管理程式 (在分散式平台上) 在通道處於 RETRYING 或 STOPPED 狀態時停止，則當佇列管理程式重新啟動時，會記住通道狀態。不過，如果佇列管理程式在通道處於 STOPPED 狀態時停止，則 SVRCONN 通道類型的通道狀態會重設。

不確定的通道

不確定通道是具有已傳送及接收訊息之遠端通道的不確定通道。

請注意，此佇列管理程式與佇列管理程式之間的區別在於哪些訊息應該確定至佇列。

您可以使用「批次活動訊號通道」參數 (BATCHHB) 來減少通道不確定的機會。當指定這個參數值時，傳送端通道會在採取任何進一步動作之前，檢查遠端通道是否仍在作用中。如果未收到任何回應，則會將接收端通道視為不再作用中。訊息可以回復及重新遞送，且傳送端通道不會處於不確定的狀態。這會減少通道可能處於不確定狀態的時間，介於傳送端通道驗證接收端通道是否仍在作用中，以及驗證接收端通道是否已接收已傳送的訊息之間。如需批次活動訊號參數的相關資訊，請參閱 [通道屬性](#)。

不確定的通道問題通常會自動解決。即使通訊遺失，且傳送端的通道不確定，且接收狀態不明，當重新建立通訊時，也會解決此狀況。為此目的保留序號及 LUWID 記錄。在 LUWID 資訊交換之前，通道是不確定的，且通道只能有一批訊息是不確定的。

必要時，您可以手動重新同步化通道。手動一詞包括使用包含 WebSphere MQ 系統管理指令的運算子或程式。手動重新同步處理程序的運作方式如下。此說明使用 MQSC 指令，但您也可以使用 PCF 對等項目。

1. 請使用 DISPLAY CHSTATUS 指令來尋找通道的 **每一個** 端前次確定的邏輯工作單元 ID (LUWID)。請使用下列指令來執行此動作：

- 對於通道的不確定端：

```
DISPLAY CHSTATUS(name) SAVED CURLUWID
```

您可以使用 CONNAME 及 XMITQ 參數來進一步識別通道。

- 對於通道的接收端：

```
DISPLAY CHSTATUS(name) SAVED LSTLUWID
```

您可以使用 CONNAME 參數來進一步識別通道。

指令不同，因為只有通道的傳送端可能不確定。接收端是毋庸置疑的。

在 WebSphere MQ for IBM i 上，可以使用 STRMQMMQSC 指令或「使用 MQM 通道狀態 CL」指令 WRKMQMCHST，從檔案執行 DISPLAY CHSTATUS 指令

2. 如果兩個 LUWID 相同，則接收端已確定傳送端視為不確定的工作單元。傳送端現在可以從傳輸佇列中移除不確定的訊息，並重新啟用它。使用下列通道 RESOLVE 指令可完成此動作：

```
RESOLVE CHANNEL(name) ACTION(COMMIT)
```

3. 如果兩個 LUWID 不同，則接收端未確定傳送端視為不確定的工作單元。傳送端需要在傳輸佇列上保留不確定的訊息並重新傳送。使用下列通道 RESOLVE 指令可完成此動作：

```
RESOLVE CHANNEL(name) ACTION(BACKOUT)
```

此處理程序完成之後，通道就不再是不確定的。如果需要，另一個通道現在可以使用傳輸佇列。

問題判斷

問題判斷有兩個不同層面：提交指令時發現的問題，以及通道作業期間發現的問題。

指令驗證

在接受指令及畫面資料進行處理之前，必須先避免錯誤。驗證所發現的任何錯誤都會立即以錯誤訊息通知使用者。

問題診斷從解譯這些錯誤訊息開始，並採取更正動作。

處理問題

在通道正常作業期間發現的問題會通知系統主控台或系統日誌。問題診斷從收集日誌中的所有相關資訊開始，並繼續進行分析以識別問題。

可能的話，會將確認及錯誤訊息傳回給起始指令的終端機。

WebSphere MQ 會產生帳戶和統計資料，您可以用來識別使用率和效能的趨勢。 **distributed** 在分散式平台上，此資訊會產生為 PCF 記錄，如需詳細資料，請參閱 [結構資料類型](#)。

訊息與訊息碼

如需協助主要診斷問題的訊息及訊息碼，請參閱 [診斷訊息及原因碼](#)。

訊息安全

除了 WebSphere MQ 的一般回復特性之外，分散式佇列管理還可確保使用訊息通道兩端之間協調的同步點程序來適當地遞送訊息。如果此程序偵測到錯誤，則會關閉通道，以便您可以調查問題，並安全地將訊息保留在傳輸佇列中，直到通道重新啟動為止。

同步點程序的另一個好處是它會在通道啟動時嘗試回復不確定狀況。(不確定是已要求同步點但要求結果不明的回復單元狀態。)同時與此機能相關聯的還有兩個功能：

1. 以確定或取消來解決
2. 重設序號

只有在異常情況下，才會使用這些功能，因為在大部分情況下，通道會自動回復。

快速、非持續訊息

非持續訊息速度 (NPMSPEED) 通道屬性可用來指定要更快速遞送通道上的任何非持續訊息。如需此屬性的相關資訊，請參閱 [非持續訊息速度 \(NPMSPEED\)](#)。

如果通道在快速傳送非持續訊息時終止，則訊息可能會遺失，必要時由應用程式安排回復。

如果接收端通道無法將訊息放置到其目的地佇列，則會將訊息放置在無法傳送的郵件佇列上 (如果已定義的話)。如果沒有，則會捨棄訊息。

註：如果通道的另一端不支援此選項，則通道會以正常速度執行。

未遞送的訊息

如需無法遞送訊息時所發生狀況的相關資訊，請參閱 [第 56 頁的『無法遞送訊息時會發生什麼情況?』](#)。

無法遞送訊息時會發生什麼情況？

當無法遞送訊息時，MCA 可以數種方式來處理它。它可以重試，它可以傳回給傳送端，也可以將它放在無法傳送郵件的佇列中。

[第 57 頁的圖 15](#) 顯示當 MCA 無法將訊息放入目的地佇列時所進行的處理。(顯示的選項不適用於所有平台。)

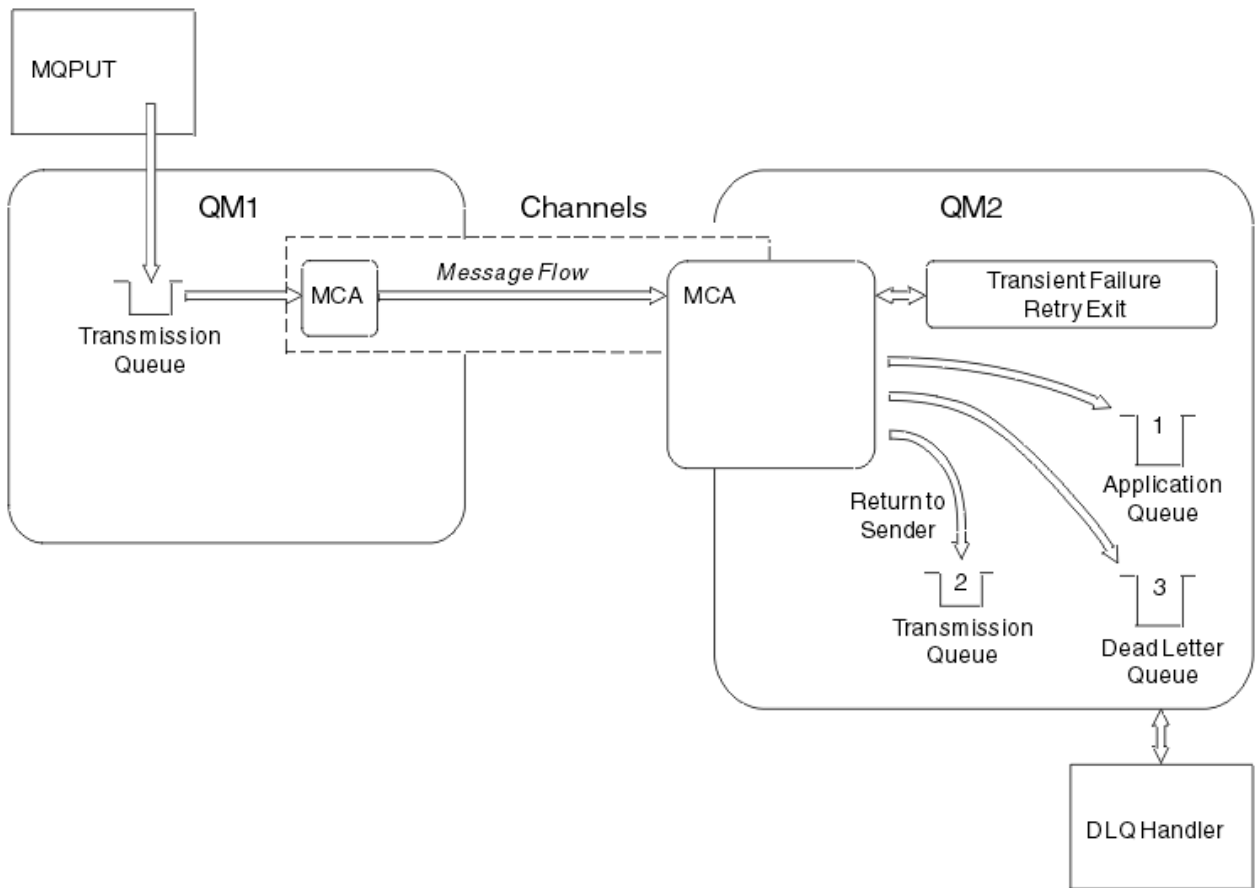


圖 15: 無法遞送訊息時會發生什麼情況

如圖所示，MCA 可以使用它無法遞送的訊息來執行數個動作。所採取的動作取決於定義通道時指定的選項，以及訊息的 MQPUT 報告選項。

1. 訊息重試 (message-retry)

如果 MCA 無法將訊息放入目標佇列，原因可能是暫時 (例如，因為佇列已滿)，MCA 可以等待稍後再重試作業。您可以決定 MCA 是否等待、等待時間長度及嘗試次數。

- 您可以在定義通道時指定 MQPUT 錯誤的訊息重試時間及間隔。如果因為佇列已滿或禁止放置而無法將訊息放入目的地佇列，則 MCA 會以指定的時間間隔嘗試指定的作業次數。
- 您可以撰寫自己的訊息-重試結束程式。結束程式可讓您指定您要 MCA 在哪些條件下重試 MQPUT 或 MQOPEN 作業。當您定義通道時，請指定結束程式的名稱。

2. 傳回至傳送端 (return-to-sender)

如果 message-retry 不成功，或發現不同類型的錯誤，MCA 可以將訊息傳回給發送端。若要啟用傳回給傳送端，當您將訊息放入原始佇列時，需要在訊息描述子中指定下列選項：

- MQRO_EXCEPTION_WITH_FULL_DATA 報告選項
- MQRO_DISCARD_MSG 報告選項
- 回覆目的地佇列及回覆目的地佇列管理程式的名稱

如果 MCA 無法將訊息放入目的地佇列，它會產生包含原始訊息的異常狀況報告，並將它放置在傳輸佇列中，以傳送至原始訊息中指定的回覆目的地佇列。(如果回覆目的地佇列位於與 MCA 相同的佇列管理模式上，則訊息會直接放置在該佇列中，而不是放置在傳輸佇列中。)

3. 無法傳送郵件的佇列

如果訊息無法遞送或傳回，則會放入無法傳送郵件的佇列 (DLQ)。您可以使用 DLQ 處理程式來處理訊息。此處理在 [使用 WebSphere MQ 無法傳送郵件的佇列處理程式處理未遞送的訊息 for IBM WebSphere](#)

MQ for UNIX Linux 中有說明。如果無法使用無法傳送郵件的佇列，則傳送端 MCA 會將訊息保留在傳輸佇列上，且通道會停止。在快速通道上，無法寫入無法傳送郵件的佇列的非持續訊息會遺失。

在 IBM WebSphere MQ Version 7.0 上，如果未定義本端無法傳送郵件的佇列，遠端佇列無法使用或定義，且沒有遠端無法傳送郵件的佇列，則傳送端通道會進入 RETRY，訊息會自動回復至傳輸佇列。

相關參考

[使用無法傳送郵件的佇列 \(USEDLQ\)](#)

觸發通道

WebSphere MQ 提供一項機能，可在符合佇列上的特定條件時自動啟動應用程式。此機能稱為觸發。

此說明旨在作為觸發概念的概觀。如需完整說明，請參閱 [使用觸發程式啟動 WebSphere MQ 應用程式](#)。

如需平台專用資訊，請參閱下列：

- 若為 Windows，請參閱 UNIX and Linux 系統: 第 59 頁的『在 UNIX、Linux 及 Windows 系統上觸發通道。』

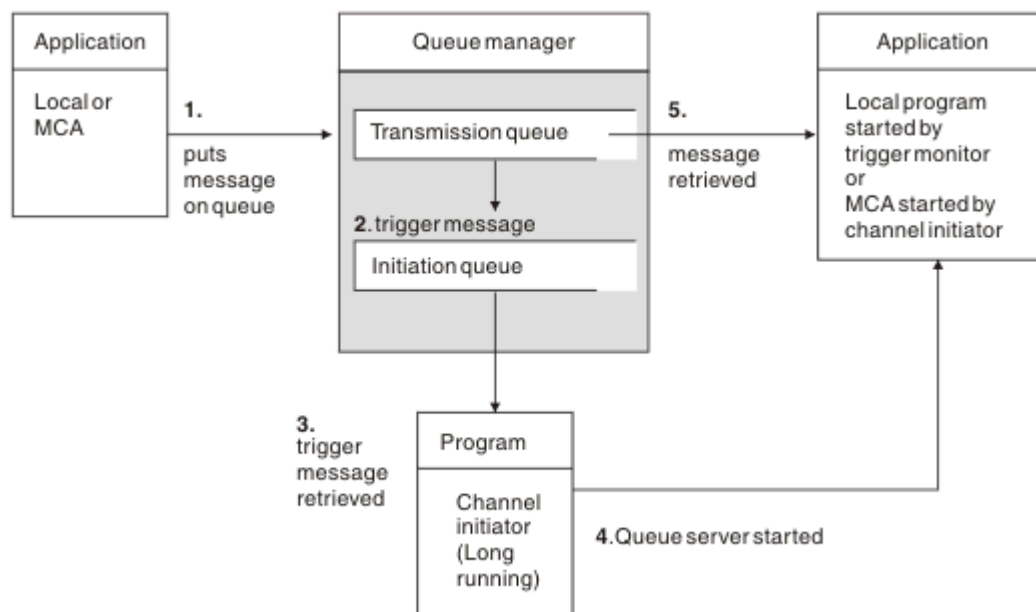


圖 16: 觸發的概念

觸發所需的物件會顯示在 [第 58 頁的圖 16](#) 中。它顯示下列事件順序：

1. 本端佇列管理程式會將來自應用程式或來自訊息通道代理程式 (MCA) 的訊息放置在傳輸佇列上。
2. 當滿足觸發條件時，本端佇列管理程式會在起始佇列上放置觸發訊息。
3. 長時間執行的通道起始程式會監視起始佇列，並在訊息到達時擷取訊息。
4. 通道起始程式會根據觸發訊息中包含的資訊來處理觸發訊息。此資訊可能包括通道名稱，在此情況下，會啟動對應的 MCA。
5. 本端應用程式或已觸發的 MCA 會從傳輸佇列中擷取訊息。

若要設定此實務範例，您需要：

- 以起始佇列名稱 (即 SYSTEM.CHANNEL.INITQ)。
- 確定起始佇列 (SYSTEM.CHANNEL.INITQ)。
- 請確定通道起始程式可用且在執行中。通道起始程式必須在其啟動指令中隨起始佇列名稱一起提供。
- 選擇性地建立觸發程式的程序定義 (如果不存在的話)，並確定 *UserData* 欄位包含它所提供的通道名稱。您可以在傳輸佇列的 *TriggerData* 屬性中指定通道名稱，而不是建立程序定義。WebSphere MQ for

UNIX、Linux 及 Windows 系統容許將通道名稱指定為空白，在此情況下會使用此傳輸佇列的第一個可用通道定義。

- 請確定傳輸佇列定義包含要提供它的程序定義名稱 (如果適用的話)、起始佇列名稱，以及您認為最適合的觸發性質。觸發控制屬性容許視需要啟用或不用觸發。

註：

- 通道起始程式會作為「觸發監視器」來監視用來啟動通道的起始佇列。
- 起始佇列及觸發程序可用來觸發任意數目的通道。
- 可以定義任何數目的起始佇列及觸發程序。
- 建議使用 FIRST 觸發類型，以避免通道啟動導致系統氾濫。

在 UNIX、Linux 及 Windows 系統上觸發通道。

您可以在 WebSphere MQ 中建立程序定義，以定義要觸發的程序。使用 MQSC 指令 DEFINE PROCESS 來建立處理程序定義，命名當訊息到達傳輸佇列時要觸發的處理程序。程序定義的 USERDATA 屬性包含傳輸佇列所提供的通道名稱。

定義本端佇列 (QM4)，指定將觸發訊息寫入起始佇列 (IQ)，以觸發啟動通道 (QM3.TO.QM4)：

```
DEFINE QLOCAL(QM4) TRIGGER INITQ(SYSTEM.CHANNEL.INITQ) PROCESS(P1) USAGE(XMITQ)
```

定義要啟動的應用程式 (程序 P1)：

```
DEFINE PROCESS(P1) USERDATA(QM3.TO.QM4)
```

或者，對於 WebSphere MQ for UNIX、Linux 及 Windows 系統，您可以透過在傳輸佇列的 TRIGDATA 屬性中指定通道名稱來消除對程序定義的需求。

定義本端佇列 (QM4)。指定將觸發訊息寫入預設起始佇列 SYSTEM.CHANNEL.INITQ，以觸發啟動通道 (QM3.TO.QM4) 的應用程式 (處理程序 P1)：

```
DEFINE QLOCAL(QM4) TRIGGER INITQ(SYSTEM.CHANNEL.INITQ)  
USAGE(XMITQ) TRIGDATA(QM3.TO.QM4)
```

如果您沒有指定通道名稱，通道起始程式會搜尋通道定義檔，直到找到與指名傳輸佇列相關聯的通道為止。

相關概念

第 59 頁的『[啟動及停止通道起始程式](#)』
使用通道起始程式處理程序來實作觸發。

第 23 頁的『[使用分散式佇列連接應用程式](#)』

本節提供 WebSphere MQ 安裝之間交互通訊的更詳細資訊，包括佇列定義、通道定義、觸發及同步點程序

相關參考

[UNIX、Linux 及 Windows 系統上的通道程式](#)

啟動及停止通道起始程式

使用通道起始程式處理程序來實作觸發。

此通道起始程式處理程序是使用 MQSC 指令 START CHINIT 來啟動。除非您使用預設起始佇列，否則請在指令上指定起始佇列的名稱。例如，若要使用 START CHINIT 指令來啟動預設佇列管理程式的佇列 IQ，請輸入：

```
START CHINIT INITQ(IQ)
```

依預設，通道起始程式會使用預設起始佇列 SYSTEM.CHANNEL.INITQ。如果您要手動啟動所有通道起始程式，請遵循下列步驟：

1. 建立並啟動佇列管理程式。
2. 將佇列管理程式的 SCHINIT 內容變更為 MANUAL
3. 結束並重新啟動佇列管理程式

在 Linux 及 Windows 系統中，通道起始程式會自動啟動。您可以啟動的通道起始程式數目受到限制。預設及上限值為 3。您可以在 UNIX and Linux 系統的 qm.ini 檔案中，以及在 Windows 系統的登錄中，使用 MAXINITIATORS 來變更此項。

如需執行通道起始程式指令 `runmqchi` 及其他控制指令的詳細資料，請參閱 [WebSphere MQ 控制指令](#)。

停止通道起始程式

當您啟動佇列管理程式時，會自動啟動預設通道起始程式。當佇列管理程式停止時，所有通道起始程式都會自動停止。

起始設定和配置檔

通道起始設定資料的處理取決於 WebSphere MQ 平台。

Windows、UNIX and Linux 系統

在 WebSphere MQ for Windows、UNIX and Linux 系統中，有配置檔可保留 WebSphere MQ 安裝的基本配置資訊。

有兩個配置檔：一個適用於機器，另一個適用於個別佇列管理程式。

WebSphere MQ 配置檔

這個檔案存放 WebSphere MQ 系統上所有佇列管理程式的相關資訊。該檔案稱為 `mqs.ini`。如需完整說明，請參閱 [管理 for WebSphere MQ for Windows、UNIX and Linux 系統](#)。

佇列管理程式配置檔

此檔案保留與一個特定佇列管理程式相關的配置資訊。該檔案稱為 `qm.ini`。

它是在佇列管理程式建立期間建立的，並且可以保留與佇列管理程式任何層面相關的配置資訊。檔案中保留的資訊包括日誌配置與 WebSphere MQ 配置檔中的預設值有何不同的詳細資料。

佇列管理程式配置檔保留在佇列管理程式所佔用的目錄樹狀結構根目錄中。例如，對於 `DefaultPath` 屬性，稱為 `QMNAME` 之佇列管理程式的佇列管理程式配置檔會是：

若為 UNIX and Linux 系統：

```
/var/mqm/qmgrs/QMNAME/qm.ini
```

以下是 `qm.ini` 檔案的摘錄。它指定 TCP/IP 接聽器在埠 2500 上接聽，現行通道數上限為 200，作用中通道數上限為 100。

```
TCP:
  Port=2500
CHANNELS:
  MaxChannels=200
  MaxActiveChannels=100
```

您可以指定埠通道要使用的 TCP/IP 埠範圍。一種方法是使用 `qm.ini` 檔案來指定埠值範圍的開始及結束。下列範例顯示指定通道範圍的 `qm.ini` 檔案：

```
TCP:
  StrPort=2500
  EndPort=3000
CHANNELS:
  MaxChannels=200
  MaxActiveChannels=100
```

如果您指定 StrPort 或 EndPort 的值，則必須同時指定兩者的值。EndPort 的值必須一律大於 StrPort 的值。

通道會嘗試使用指定範圍內的每一個埠值。當連線成功時，埠值是通道隨後使用的埠。

若為 Windows 系統：

```
C:\Program Files\IBM\WebSphere MQ\qmgrs\QMNAME\qm.ini
```

如需 qm.ini 檔案的相關資訊，請參閱 [分散式佇列的配置檔段落](#)。

訊息的資料轉換

在不同佇列管理程式上的佇列之間傳送 WebSphere MQ 訊息時，可能需要進行資料轉換。

WebSphere MQ 訊息由兩個部分組成：

- 訊息描述子中的控制資訊
- 應用程式資料

在不同佇列管理程式上的佇列之間傳送時，這兩個部分中的任一部分可能需要資料轉換。如需應用程式資料轉換的相關資訊，請參閱 [應用程式資料轉換](#)。

撰寫您自己的訊息通道代理程式

WebSphere MQ 可讓您撰寫自己的訊息通道代理程式 (MCA) 程式，或安裝來自獨立軟體供應商的訊息通道代理程式。

您可能想要撰寫您自己的 MCA 程式，讓 WebSphere MQ 透過您自己的專有通訊協定交互作業，或透過 WebSphere MQ 不支援的通訊協定來傳送訊息。(您無法撰寫自己的 MCA，以在另一端與 WebSphere MQ 提供的 MCA 交互作業。)

如果您決定使用 WebSphere MQ 未提供的 MCA，則必須考量下列要點。

訊息傳送及接收

您必須撰寫傳送端應用程式，從應用程式放置訊息的任何地方 (例如從傳輸佇列) 取得訊息，並以您要通訊的通訊協定將訊息送出。您也必須撰寫接收端應用程式，以從這個通訊協定取得訊息，並將它們放在目的地佇列中。傳送及接收應用程式使用訊息佇列介面 (MQI) 呼叫，而非任何特殊介面。

您必須確定訊息只遞送一次。同步點協調可用來協助進行此遞送。

通道控制功能

您必須提供自己的管理功能來控制通道。您無法使用 WebSphere MQ 通道管理功能來配置 (例如，DEFINE CHANNEL 指令) 或監視 (例如，DISPLAY CHSTATUS) 通道。

起始設定檔

如果您需要起始設定檔，則必須提供您自己的起始設定檔。

應用程式資料轉換

您可能想要容許對傳送至不同系統的訊息進行資料轉換。若是如此，當從應用程式放置訊息的任何地方 (例如傳輸佇列) 擷取訊息時，請在 MQGET 呼叫上使用 MQGMO_CONVERT 選項。

使用者結束程式

請考量您是否需要使用者結束程式。若是如此，您可以使用 WebSphere MQ 所使用的相同介面定義。

觸發

如果您的應用程式將訊息放入傳輸佇列，您可以設定傳輸佇列屬性，以便在訊息到達佇列時觸發傳送 MCA。

通道起始程式

您可能必須提供自己的通道起始程式。

分散式佇列管理要考量的其他事項

準備 WebSphere MQ 以進行分散式佇列管理時要考量的其他主題。本主題涵蓋「未遞送訊息佇列」、「使用中佇列」、「系統延伸及使用者結束程式」，以及「作為授信應用程式執行通道及接聽器」。

無法遞送的訊息佇列

若要確保處理到達未遞送訊息佇列 (也稱為無法傳送郵件的佇列或 DLQ) 的訊息，請建立可定期觸發或執行以處理這些訊息的程式。在 UNIX and Linux 系統上，WebSphere MQ 提供 DLQ 處理程式；如需相關資訊，請參閱 [範例 DLQ 處理程式 amqsdldq](#)。

使用中佇列

即使未傳輸訊息，接收端通道的 MCA 也可以保持目的地佇列開啟。這會導致佇列出現 "使用中"。

通道數目上限

請參閱 [分散式佇列的配置檔段落](#)。

系統擴充及使用者結束程式

在通道定義中提供機能，可讓額外程式在訊息處理期間的定義時間執行。這些程式不是隨 WebSphere MQ 一起提供，但每一個安裝架構可以根據本端需求來提供。

若要執行，這些使用者結束程式必須具有預先定義的名稱，且可以在呼叫通道程式時使用。訊息通道定義中包括使用者結束程式的名稱。

有一個已定義的控制區塊介面，用於將控制權移交給這些程式，以及處理從這些程式傳回的控制權。

在 [傳訊通道的通道結束程式](#) 中，會找到呼叫這些程式的確切位置，以及控制區塊和名稱的詳細資料。

將通道和接聽器當作授信應用程式來執行

如果效能是環境中的重要考量，且您的環境是穩定的，則可以使用 FASTPATH 連結，將通道及接聽器執行為受信任。有兩個因素會影響通道及接聽器是否以受信任的方式執行：

- 環境變數 MQ_CONNECT_TYPE=FASTPATH 或 MQ_CONNECT_TYPE = STANDARD。這會區分大小寫。如果您指定無效的值，則會忽略它。
- qm.ini 或登錄檔的「通道」段落中的 MQIBindType。您可以將此設定為 FASTPATH 或 STANDARD，且不區分大小寫。預設值為「標準」。

您可以使用與環境變數相關聯的 MQIBindType 來達到所需的效果，如下所示：

MQIBindType	環境變數	結果
標準	未定義	標準
Fastpath	未定義	Fastpath
標準	標準	標準
Fastpath	標準	標準
標準	Fastpath	標準
Fastpath	Fastpath	Fastpath
標準	用戶端	用戶端
Fastpath	用戶端	標準
標準	本端	標準
Fastpath	本端	標準

總之，只有兩種方法可讓通道和接聽器實際以信任的方式執行：

1. 在 qm.ini 或登錄中指定 MQIBindType= FASTPATH，而不指定環境變數。
2. 在 qm.ini 或登錄中指定 MQIBindType= FASTPATH，並將環境變數設為 FASTPATH。

請將執行接聽器視為受信任，因為接聽器是穩定的處理程序。除非您使用不穩定通道結束程式或 STOP CHANNEL MODE (TERMINATE) 指令，否則請將執行通道視為授信。

在 UNIX, Linux, and Windows 上監視及控制通道

對於 DQM，您需要建立、監視及控制遠端佇列管理程式的通道。您可以使用指令、程式、IBM WebSphere MQ Explorer、通道定義的檔案，以及同步化資訊的儲存區來控制通道。

您可以使用下列類型的指令：

IBM WebSphere MQ 指令 (MQSC)

在 Windows、UNIX and Linux 系統中，您可以使用 MQSC 作為 MQSC 階段作業中的單一指令。若要發出更複雜的或多個指令，可以將 MQSC 建置在您隨後從指令行執行的檔案中。如需詳細資料，請參閱 [MQSC 指令](#)。本節提供使用 MQSC 進行分散式佇列作業的一些簡單範例。

通道指令是「IBM WebSphere MQ 指令 (MQSC)」的子集。您可以使用 MQSC 及控制指令來執行下列動作：

- 建立、複製、顯示、變更及刪除通道定義
- 當無法重新建立鏈結時，啟動和停止通道、連線測試、重設通道序號，以及解決不確定的訊息
- 顯示通道的狀態資訊

控制指令

您也可以指令行針對部分這些功能發出控制指令。如需詳細資料，請參閱 [控制指令](#)。

可程式化指令格式指令

如需詳細資料，請參閱 [PCF 指令](#)。

IBM WebSphere MQ Explorer

在 UNIX、Linux 及 Windows 系統上，您可以使用 IBM WebSphere MQ Explorer。這提供圖形管理介面來執行管理作業，作為使用控制指令或 MQSC 指令的替代方案。通道定義會保留為佇列管理程式物件。

每一個佇列管理程式都有一個 DQM 元件，用來控制與相容遠端佇列管理程式的交互連線。儲存區保留序號及邏輯工作單元 (LUW) ID。這些用於通道同步化。

如需使用不同類型指令來設定及控制訊息通道時可供您使用的功能清單，請參閱 [第 64 頁的表 8](#)。

相關概念

[第 66 頁的『開始使用物件』](#)

必須先定義通道，且其關聯物件必須存在且可供使用，才能啟動通道。這一節教您如何。

[第 71 頁的『設定 Windows 的通訊』](#)

當啟動分散式佇列管理通道時，它會嘗試使用通道定義中指定的連線。若要順利完成此作業，必須定義連線並使其可用。本節說明如何使用 WebSphere MQ for Windows 系統的四種通訊形式之一來執行此作業。

[第 79 頁的『在 UNIX and Linux 系統上設定通訊』](#)

DQM 是 IBM WebSphere MQ 的遠端佇列作業機能。它為佇列管理程式提供通道控制程式，這些佇列管理程式形成通訊鏈結的介面，可由系統操作員控制。分散式佇列管理所保留的通道定義會使用這些連線。

相關參考

[UNIX、Linux 及 Windows 系統上的通道程式](#)

[分散式平台的訊息通道規劃範例](#)

[配置資訊範例](#)

[通道屬性](#)

設定及控制通道所需的功能

可能需要一些 IBM WebSphere MQ 功能才能設定及控制通道。本主題說明通道功能。

您可以使用 IBM WebSphere MQ 提供的預設值來建立通道定義，並指定通道名稱、您要建立的通道類型、要使用的通訊方法、傳輸佇列名稱及連線名稱。

通道名稱在通道兩端必須相同，且在網路內是唯一的。不過，您必須將使用的字元限制為適用於 IBM WebSphere MQ 物件名稱的字元。

如需其他通道相關功能，請參閱下列主題：

- [第 66 頁的『開始使用物件』](#)
- [第 66 頁的『建立相關聯的物件』](#)
- [第 66 頁的『建立預設物件』](#)
- [第 66 頁的『建立頻道』](#)
- [第 67 頁的『顯示通道』](#)
- [第 67 頁的『顯示通道狀態』](#)
- [第 68 頁的『使用連線測試檢查鏈結』](#)
- [第 68 頁的『啟動通道』](#)
- [第 69 頁的『停止通道』](#)
- [第 70 頁的『重新命名通道』](#)
- [第 70 頁的『重設通道』](#)
- [第 70 頁的『解決通道上不確定的訊息』](#)

第 64 頁的表 8 顯示您可能需要的 IBM WebSphere MQ 函數完整清單。

表 8: UNIX, Linux, and Windows 系統中所需的機能			
函數	控制指令	MQSC	WebSphere MQ 探險家對等項目?
佇列管理程式功能			
變更佇列管理程式		ALTER QMGR	是
建立佇列管理程式	crtmqm		是
刪除佇列管理程式	dlmqm		是
顯示佇列管理程式		DISPLAY QMGR	是
結束佇列管理程式	endmqm		是
Ping 佇列管理程式		PING 佇列管理程式	否
啟動佇列管理程式	strmqm		是
指令伺服器功能			
顯示指令伺服器	dspmqcsv		否
結束指令伺服器	endmqcsv		否
啟動指令伺服器	strmqcsv		否
佇列功能			
變更佇列		ALTER QALIAS ALTER QLOCAL ALTER QMODEL ALTER QREMOTE 請參閱 ALTER 佇列 。	是
清除佇列		CLEAR QLOCAL	是

表 8: UNIX, Linux, and Windows 系統中所需的機能 (繼續)			
函數	控制指令	MQSC	WebSphere MQ 探險家對等項目?
建立佇列		DEFINE QALIAS DEFINE QLOCAL DEFINE QMODEL DEFINE QREMOTE 請參閱 DEFINE 佇列 。	是
刪除佇列		DELETE QALIAS DELETE QLOCAL DELETE QMODEL DELETE QREMOTE 請參閱 DELETE 佇列 。	是
顯示佇列		顯示佇列	是
處理程序函數			
變更處理程序		ALTER PROCESS	是
建立處理程序		DEFINE PROCESS	是
刪除處理程序		刪除處理程序	是
顯示程序		DISPLAY PROCESS	是
Channel functions			
變更通道		ALTER CHANNEL	是
建立通道		定義通道	是
刪除通道		刪除通道	是
顯示通道		顯示通道	是
顯示通道狀態		DISPLAY CHSTATUS	是
結束通道		停止通道	是
Ping 通道		Ping 通道	是
重設通道		重設通道	是
解析通道		解析通道	是
執行通道	runmqchl	啟動通道	是
執行通道起始程式	runmqchi	開始 CHINIT	否
執行接聽器 ¹	runmqlsr	啟動接聽器	否
結束接聽器	endmqlsr (僅限 Windows 系統、AIX、HP-UX 及 Solaris)		否
註:			
1. 當佇列管理程式啟動時，可能會自動啟動接聽器。			

開始使用物件

必須先定義通道，且其關聯物件必須存在且可供使用，才能啟動通道。這一節教您如何。

使用 WebSphere MQ 指令 (MQSC) 或「IBM WebSphere MQ 探險家」來執行下列動作：

1. 定義訊息通道及相關聯的物件
2. 監視及控制訊息通道

您可能需要定義的相關聯物件如下：

- 傳輸佇列
- 遠端佇列定義
- 佇列管理程式別名定義
- 回覆目的地佇列別名定義
- 回覆本端佇列
- 觸發程序 (MCA)
- 訊息通道定義

必須先定義並提供每一個通道的特定通訊鏈結，然後才能執行通道。如需如何定義 LU 6.2、TCP/IP、NetBIOS、SPX 及 DECNet 鏈結的說明，請參閱安裝的特定通訊手冊。另請參閱 [配置資訊範例](#)。

如需建立及使用物件的相關資訊，請參閱下列子主題：

建立相關聯的物件

MQSC 用來建立相關聯的物件。

使用 MQSC 來建立佇列及別名物件：傳輸佇列、遠端佇列定義、佇列管理程式別名定義、回覆目的地佇列別名定義，以及回覆目的地本端佇列。

此外，也會以類似的方式建立觸發 (MCA) 的處理程序定義。

如需顯示如何建立所有必要物件的範例，請參閱 [分散式平台的訊息通道規劃範例](#)。

建立預設物件

建立佇列管理程式時，會自動建立預設物件。這些物件是佇列、通道、程序定義及管理佇列。建立預設物件之後，您可以隨時執行 `strmqm` 指令並指定 `-c` 選項來取代它們。

當您使用 `crtmqm` 指令來建立佇列管理程式時，該指令也會起始一個程式來建立一組預設物件。

1. 會依序建立每一個預設物件。程式會保留已順利定義的物件數、存在及已被取代的物件數，以及失敗嘗試次數的計數。
2. 程式會將結果顯示給您，如果發生任何錯誤，則會將您引導至適當的錯誤日誌，以取得詳細資料。

當程式完成執行時，您可以使用 `strmqm` 指令來啟動佇列管理程式。

如需 `crtmqm` 及 `strmqm` 指令的相關資訊，請參閱 [控制指令](#)。

變更預設物件

當您指定 `-c` 選項時，佇列管理程式會在建立物件時暫時啟動，然後再次關閉。使用 `-c` 選項發出 `strmqm` 會以預設值重新整理現有系統物件 (例如，通道定義的 `MCAUSER` 屬性設為空白)。如果您要啟動佇列管理程式，則必須再次使用不含 `-c` 選項的 `strmqm` 指令。

如果您想要變更預設物件，則可以建立您自己的舊 `amqscoma.tst` 檔案版本並編輯它。

建立頻道

建立 **兩個** 通道定義，連線兩端各一個。您可以在第一個佇列管理程式建立第一個通道定義。然後在鏈結另一端的第二個佇列管理程式上建立第二個通道定義。

兩端必須使用 **相同的** 通道名稱來定義。兩端必須具有 **相容** 通道類型，例如：傳送端和接收端。

若要為鏈結的一端建立通道定義，請使用 MQSC 指令 DEFINE CHANNEL。包括通道名稱、連線此端的通道類型、連線名稱、說明 (必要的話)、傳輸佇列名稱 (必要的話)，以及傳輸通訊協定。也請使用您先前收集的資訊，併入您要不同於所需通道類型之系統預設值的任何其他屬性。

系統會提供協助來決定 通道屬性中通道屬性的值。

註：建議您將網路中的所有通道唯一命名。在通道名稱中包含來源和目標佇列管理程式名稱是很好的作法。

建立通道範例

```
DEFINE CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) +
DESCR('Sender channel to QM2') +
CONNNAME(QM2) TRPTYPE(TCP) XMITQ(QM2) CONVERT(YES)
```

在 MQSC 的所有範例中，指令會顯示在指令檔中，以及在 Windows 或 UNIX 或 Linux 系統中所輸入的指令。這兩種方法看起來相同，除了以互動方式發出指令之外，您必須先啟動 MQSC 階段作業。輸入 `runmqsc`，代表預設佇列管理程式，或輸入 `runmqsc qmname`，其中 `qmname` 是所需佇列管理程式的名稱。然後鍵入任意數目的指令，如範例中所示。

為了可攜性，請將指令的行長度限制為 72 個字元。請使用連結字元 `+`，如所示，以在多行上繼續。在 Windows 上，使用 `Ctrl-z` 來結束指令行的項目。在 UNIX and Linux 系統上，使用 `Ctrl-d`。或者，在 UNIX、Linux 或 Windows 系統上，使用 `end` 指令。

顯示通道

使用 MQSC 指令 DISPLAY CHANNEL 來顯示通道的屬性。

如果未要求任何特定屬性，且指定的通道名稱不是同屬，則依預設會採用 DISPLAY CHANNEL 指令的 ALL 參數。

這些屬性在 [通道屬性](#) 中有說明。

顯示通道範例

```
DISPLAY CHANNEL(QM1.TO.QM2) TRPTYPE,CONVERT
DISPLAY CHANNEL(QM1.TO.*) TRPTYPE,CONVERT
DISPLAY CHANNEL(*) TRPTYPE,CONVERT
DISPLAY CHANNEL(QM1.TO.QMR34) ALL
```

顯示通道狀態

請使用 MQSC 指令 DISPLAY CHSTATUS，指定通道名稱，以及您是否想要通道的現行狀態或已儲存資訊的狀態。

DISPLAY CHSTATUS 會套用至所有訊息通道。它不適用於伺服器連線通道以外的 MQI 通道。

顯示的資訊包括：

- 通道名稱
- 通訊連線名稱
- 通道的不確定狀態 (適當時)
- 最後序號
- 傳輸佇列名稱 (適當時)
- 不確定的 ID (適當時)
- 前次確定的序號
- 邏輯工作單元 ID
- 處理程序 ID
- 執行緒 ID (僅限 Windows)

顯示通道狀態範例

```
DISPLAY CHSTATUS(*) CURRENT
DISPLAY CHSTATUS(QM1.TO.*) SAVED
```

在通道上傳至少一個批次的訊息之前，不會套用已儲存的狀態。當通道停止 (使用 STOP CHL 指令) 及佇列管理程式結束時，也會儲存狀態。

使用連線測試檢查鏈結

使用 MQSC 指令 PING CHANNEL，與遠端系統交換固定資料訊息。

Ping 可讓系統監督者確信鏈結可用且正常運作。

連線測試不涉及使用傳輸佇列和目標佇列。它使用通道定義、相關通訊鏈結及網路設定。只有在通道目前不在作用中時，才能使用它。

它只能從傳送端和伺服器通道使用。對應的通道會在鏈結的遠端啟動，並執行啟動參數協議。正常通知錯誤。

訊息交換的結果會呈現為 Ping complete 或錯誤訊息。

使用 LU 6.2 進行連線測試

當呼叫連線測試時，依預設不會有任何使用者 ID 或密碼流向接收端。如果需要使用者 ID 和密碼，則可以在通道定義中的起始端建立它們。如果在通道定義中輸入密碼，則會先由 WebSphere MQ 加密再儲存。然後會在流經交談之前將它解密。

啟動通道

對傳送端、伺服器及要求端通道使用 MQSC 指令 START CHANNEL。為了讓應用程式能夠交換訊息，您必須針對入埠連線啟動接聽器程式。

如果已設定通道來觸發佇列管理程式，則不需要「啟動通道」。

啟動時，傳送端 MCA 會讀取通道定義並開啟傳輸佇列。發出通道啟動序列，其遠端啟動接收端或伺服器通道的對應 MCA。當訊息已啟動時，傳送端和伺服器處理程序會等待訊息到達傳輸佇列，並在訊息到達時傳輸訊息。

當您使用觸發或執行通道作為執行緒時，請確定通道起始程式可用來監視起始佇列。依預設，通道起始程式會作為佇列管理程式的一部分來啟動。

不過，TCP 及 LU 6.2 確實提供其他功能：

- 對於 UNIX and Linux 系統上的 TCP，inetd 可以配置為啟動通道。inetd 作為個別處理程序啟動。
- 若為 UNIX and Linux 系統中的 LU 6.2，請配置您的 SNA 產品以啟動 LU 6.2 回應者處理程序。
- 對於 Windows 系統中的 LU 6.2，您可以使用 SNA Server 來啟動通道。TpStart (SNA Server 隨附的公用程式)。TpStart 作為個別處理程序啟動。

必要的話，使用「啟動」選項一律會導致通道重新同步化。

若要開始成功，請執行下列動作：

- 通道定義 (本端及遠端) 必須存在。如果接收端或伺服器連線通道沒有適當的通道定義，則在自動定義通道時，會自動建立預設通道定義。請參閱 [通道自動定義結束程式 \(Channel auto-definition exit program\)](#)。
- 傳輸佇列必須存在，且沒有其他通道使用它。
- MCA (本端及遠端) 必須存在。
- 通訊鏈結必須可用。
- 佇列管理程式必須在執行中、本端及遠端。
- 訊息通道不得已在執行中。

畫面會傳回一則訊息，確認已接受啟動通道的要求。如需確認啟動指令已成功，請檢查錯誤日誌，或使用 DISPLAY CHSTATUS。錯誤日誌如下：

Windows

```
MQ_INSTALLATION_PATH\qmgrs\qmname\errors\AMQERR01.LOG (針對稱為 qmname 的每一個佇列管理程式)
```

```
MQ_INSTALLATION_PATH\qmgrs\@SYSTEM\errors\AMQERR01.LOG (適用於一般錯誤)
```

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

註：在 Windows 系統上，您仍會在 Windows 系統應用程式事件日誌中收到訊息。

UNIX and Linux 系統

```
/var/mqm/qmgrs/qmname/errors/AMQERR01.LOG (針對稱為 qmname 的每一個佇列管理程式)
```

```
/var/mqm/qmgrs/@SYSTEM/errors/AMQERR01.LOG (適用於一般錯誤)
```

在 Windows、UNIX and Linux 系統上，請使用 runmqslsr 指令來啟動 WebSphere MQ 接聽器程序。依預設，通道連接的任何入埠要求都會導致接聽器處理程序以 amqrmppa 處理程序的執行緒形式啟動 MCA。

```
runmqslsr -t tcp -m QM2
```

對於出埠連線，您必須使用下列三種方式之一來啟動通道：

1. 視 MCATYPE 參數而定，使用 MQSC 指令 START CHANNEL 來指定通道名稱，以啟動通道作為處理程序或執行緒。(如果通道啟動為執行緒，則它們是通道起始程式的執行緒。)

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

2. 使用控制指令 runmqchl 將通道作為處理程序啟動。

```
runmqchl -c QM1.TO.QM2 -m QM1
```

3. 使用通道起始程式來觸發通道。

停止通道

使用 MQSC 指令 STOP CHANNEL 來要求通道停止活動。在操作員再次啟動通道之前，通道不會啟動新的訊息批次。

如需重新啟動已停止通道的相關資訊，請參閱第 54 頁的『[重新啟動已停止的通道](#)』。

此指令可以發出至 MQCHT_CLNTCONN 以外任何類型的通道。

您可以選取您需要的停止類型：

停止靜止範例

```
STOP CHANNEL(QM1.TO.QM2) MODE(QUIESCE)
```

此指令會要求通道依序關閉。訊息的現行批次已完成，同步點程序與通道的另一端一起執行。如果通道閒置，則此指令不會終止接收端通道。

停止強制範例

```
STOP CHANNEL(QM1.TO.QM2) MODE(FORCE)
```

此選項會立即停止通道，但不會終止通道的執行緒或處理程序。通道未完成處理現行訊息批次，因此可以讓通道不確定。一般而言，請考量使用靜止停止選項。

停止終止範例

```
STOP CHANNEL(QM1.TO.QM2) MODE(TERMINATE)
```

此選項會立即停止通道，並終止通道的執行緒或處理程序。

停止 (靜止) 已停止範例

```
STOP CHANNEL(QM1.TO.QM2) STATUS(STOPPED)
```

此指令未指定 MODE，因此預設為 MODE (QUIESCE)。它會要求停止通道，以便它無法自動重新啟動，但必須手動啟動。

停止 (靜止) 非作用中範例

```
STOP CHANNEL(QM1.TO.QM2) STATUS(INACTIVE)
```

此指令未指定 MODE，因此預設為 MODE (QUIESCE)。它會要求將通道設為非作用中，以便在必要時自動重新啟動。

重新命名通道

請使用 MQSC 來重新命名訊息通道。

請使用 MQSC 來執行下列步驟：

1. 請使用 STOP CHANNEL 來停止通道。
2. 使用 DEFINE CHANNEL 來建立具有新名稱的重複通道定義。
3. 使用 DISPLAY CHANNEL 來檢查它是否已正確建立。
4. 使用 DELETE CHANNEL 來刪除原始通道定義。

如果您決定重新命名訊息通道，請記住通道有 **兩個** 通道定義，每一端一個。請確定您同時在兩端重新命名通道。

重設通道

請使用 MQSC 指令 RESET CHANNEL 來變更訊息序號。

RESET CHANNEL 指令適用於任何訊息通道，但不適用於 MQI 通道 (用戶端連線或伺服器連線)。下次啟動通道時，第一則訊息會啟動新序列。

如果在傳送端或伺服器通道上發出指令，當通道重新啟動時，它會通知另一端發生變更。

相關概念

[第 66 頁的『開始使用物件』](#)

必須先定義通道，且其關聯物件必須存在且可供使用，才能啟動通道。這一節教您如何。

[第 46 頁的『通道控制功能』](#)

通道控制功能可讓您定義、監視及控制通道。

[第 23 頁的『使用分散式佇列連接應用程式』](#)

本節提供 WebSphere MQ 安裝之間交互通訊的更詳細資訊，包括佇列定義、通道定義、觸發及同步點程序

相關參考

[重設通道](#)

解決通道上不確定的訊息

當傳送端或伺服器保留不確定的訊息時，請使用 MQSC 指令 RESOLVE CHANNEL。例如，因為鏈結的一端已終止，且無法進行回復。

RESOLVE CHANNEL 指令接受下列兩個參數之一：BACKOUT 或 COMMIT。取消會將訊息還原至傳輸佇列，而「確定」會捨棄它們。

通道程式不會嘗試建立與友機的階段作業。相反地，它會決定代表不確定訊息的邏輯工作單元 ID (LUWID)。然後，它會根據要求發出下列其中一項：

- BACKOUT-將訊息還原至傳輸佇列; 或
- COMMIT 以刪除傳輸佇列中的訊息。

若要讓解決方案成功，請執行下列動作：

- 通道必須處於非作用中
- 通道必須不確定
- 通道類型必須是傳送端或伺服器
- 本端通道定義必須存在
- 本端佇列管理程式必須在執行中

相關概念

[第 66 頁的『開始使用物件』](#)

必須先定義通道，且其關聯物件必須存在且可供使用，才能啟動通道。這一節教您如何。

[第 46 頁的『通道控制功能』](#)

通道控制功能可讓您定義、監視及控制通道。

[第 23 頁的『使用分散式佇列連接應用程式』](#)

本節提供 WebSphere MQ 安裝之間交互通訊的更詳細資訊，包括佇列定義、通道定義、觸發及同步點程序

相關參考

[解析通道](#)

設定 Windows 的通訊

當啟動分散式佇列管理通道時，它會嘗試使用通道定義中指定的連線。若要順利完成此作業，必須定義連線並使其可用。本節說明如何使用 WebSphere MQ for Windows 系統的四種通訊形式之一來執行此作業。

您可能會發現參閱 [範例配置 - IBM WebSphere MQ for Windows](#) 會很有用。

若為 UNIX and Linux 系統，請參閱 [第 79 頁的『在 UNIX and Linux 系統上設定通訊』](#)。

決定連線

針對 WebSphere MQ for Windows 系統，從下列四種通訊形式中選擇：

- [第 72 頁的『在 Windows 上定義 TCP 連線』](#)
- [第 73 頁的『在 Windows 上定義 LU 6.2 連線』](#)
- [第 74 頁的『在 Windows 上定義 NetBIOS 連線』](#)
- [第 77 頁的『在 Windows 上定義 SPX 連線』](#) (僅限 Windows XP 及 Windows 2003 Server)

每一個通道定義只能指定一個通訊協定作為「傳輸通訊協定 (傳輸類型)」屬性。佇列管理程式可以使用一或多個通訊協定。

對於 WebSphere MQ 用戶端，使用不同傳輸通訊協定的替代通道可能很有用。如需 WebSphere MQ 用戶端的相關資訊，請參閱 [用戶端概觀](#)。

相關概念

[第 23 頁的『使用分散式佇列連接應用程式』](#)

本節提供 WebSphere MQ 安裝之間交互通訊的更詳細資訊，包括佇列定義、通道定義、觸發及同步點程序

[第 63 頁的『在 UNIX, Linux, and Windows 上監視及控制通道』](#)

對於 DQM，您需要建立、監視及控制遠端佇列管理程式的通道。您可以使用指令、程式、IBM WebSphere MQ Explorer、通道定義的檔案，以及同步化資訊的儲存區來控制通道。

[第 84 頁的『配置用戶端與伺服器之間的連線』](#)

若要配置 WebSphere MQ MQI 用戶端與伺服器之間的通訊鏈結，請決定您的通訊協定，定義鏈結兩端的連線，啟動接聽器，以及定義通道。

在 Windows 上定義 TCP 連線

在傳送端配置通道以指定目標位址，並在接收端執行接聽器程式，以定義 TCP 連線。

傳送端

在通道定義的「連線名稱」欄位中，指定目標機器的主機名稱或 TCP 位址。

要連接的埠預設為 1414。埠號 1414 由「網際網路指派號碼」權限指派給 IBM WebSphere MQ。

如果要使用預設值以外的埠號，請在通道物件定義的連線名稱欄位中指定它：

```
DEFINE CHANNEL('channel name') CHLTYPE(SDR) +  
  TRPTYPE(TCP) +  
  CONNAME('OS2ROG3(1822)') +  
  XMITQ('XMitQ name') +  
  REPLACE
```

其中 OS2ROG3 是遠端佇列管理程式的 DNS 名稱，而 1822 是所需的埠。（這必須是接收端接聽器接聽所在的埠。）

必須停止並重新啟動執行中通道，才能取得通道物件定義的任何變更。

您可以在 IBM WebSphere MQ for Windows 的 .ini 檔中指定預設埠號，以變更預設埠號：

```
TCP:  
Port=1822
```

註：若要選取要使用的 TCP/IP 埠號，IBM WebSphere MQ 會依下列順序使用它找到的第一個埠號：

1. 通道定義或指令行中明確指定的埠號。此數字容許針對通道置換預設埠號。
2. 在 .ini 檔案的 TCP 段落中指定的埠屬性。此數字容許置換佇列管理程式的預設埠號。
3. 預設值 1414。這是「網際網路指派號碼機構」針對入埠和出埠連線指派給 IBM WebSphere MQ 的號碼。

如需您使用 qm.ini 所設定值的相關資訊，請參閱 [分散式佇列的配置檔段落](#)。

在 TCP 上接收

若要啟動接收端通道程式，必須啟動接聽器程式，以偵測送入的網路要求並啟動相關聯的通道。您可以使用 IBM WebSphere MQ 接聽器。

接收通道程式會啟動，以回應來自傳送通道的啟動要求。

若要啟動接收端通道程式，必須啟動接聽器程式，以偵測送入的網路要求並啟動相關聯的通道。您可以使用 IBM WebSphere MQ 接聽器。

若要執行隨 IBM WebSphere MQ 提供的接聽器（以執行緒方式啟動新通道），請使用 [runmqtsr](#) 指令。

使用 **runmqtsr** 指令的基本範例：

```
runmqtsr -t tcp [-m QMNAME] [-p 1822]
```

方括弧指出選用參數；預設佇列管理程式不需要 QMNAME，如果您使用預設值（1414），則不需要埠號。埠號不可超過 65535。

註：若要選取要使用的 TCP/IP 埠號，IBM WebSphere MQ 會依下列順序使用它找到的第一個埠號：

1. 通道定義或指令行中明確指定的埠號。此數字容許針對通道置換預設埠號。
2. 在 .ini 檔案的 TCP 段落中指定的埠屬性。此數字容許置換佇列管理程式的預設埠號。

3. 預設值 1414。這是「網際網路指派號碼機構」針對入埠和出埠連線指派給 IBM WebSphere MQ 的號碼。

若要取得最佳效能，請如第 62 頁的『將通道和接聽器當作授信應用程式來執行』中所述，將 IBM WebSphere MQ 接聽器作為授信應用程式執行。如需授信應用程式的相關資訊，請參閱 [授信應用程式的限制](#)

使用 TCP/IP SO_KEEPALIVE 選項

如果您要使用 Windows SO_KEEPALIVE 選項，則必須將下列項目新增至登錄：

```
TCP:
KeepAlive=yes
```

如需 SO_KEEPALIVE 選項的相關資訊，請參閱第 52 頁的『正在檢查通道的另一端是否仍然可用』。

在 Windows 上，Windows KeepAlive 選項的 HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters 登錄值會控制在檢查連線之前所經歷的間隔。預設值為兩小時。

在 Windows 上定義 LU 6.2 連線

必須配置 SNA，才能在兩部機器之間建立 LU 6.2 交談。

配置 SNA 之後，請按如下所示繼續進行。

如需相關資訊，請參閱下表。

遠端平台	TPNAME	TPPATH
z/OS 或 MVS/ESA (不含 CICS)	與遠端佇列管理程式的對應端資訊相同。	-
z/OS 或使用 CICS 的 MVS/ESA	CKRC (傳送端) CKSV (要求端) CKRC (伺服器)	-
IBM i	與 IBM i 系統上遞送登錄中的比較值相同。	-
UNIX and Linux 系統	與遠端佇列管理程式的對應端資訊相同。	MQ_INSTALLATION_PATH/bin/amqcrs6a
Windows	如 Windows 執行接聽器指令中所指定，或在 Windows 上使用 TpSetup 定義的可呼叫交易程式。	MQ_INSTALLATION_PATH\bin\amqcrs6a

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

如果您在同一部機器上有多個佇列管理程式，請確定通道定義中的 TPname 是唯一的。

如需透過 TCP/IP 配置 AnyNet SNA 的最新資訊，請參閱下列線上 IBM 文件: [AnyNet SNA over TCP/IP 及 SNA 節點作業](#)。

相關概念

第 74 頁的『在 LU 6.2 上傳送結束』

從您正在使用之 LU 6.2 產品的管理應用程式建立 CPI-C 端物件 (符號式目的地)。在通道定義的連線名稱欄位中輸入此名稱。另請建立指向夥伴的 LU 6.2 鏈結。

第 74 頁的『在 LU 6.2 上接收』

接收通道程式會啟動，以回應來自傳送通道的啟動要求。

在 LU 6.2 上傳送結束

從您正在使用之 LU 6.2 產品的管理應用程式建立 CPI-C 端物件 (符號式目的地)。在通道定義的連線名稱欄位中輸入此名稱。另請建立指向夥伴的 LU 6.2 鏈結。

在 CPI-C 端物件中，輸入接收機器上的友機 LU 名稱、TP 名稱及模式名稱。例如：

```
Partner LU Name      OS2R0G2
Partner TP Name     recv
Mode Name           #INTER
```

在 LU 6.2 上接收

接收通道程式會啟動，以回應來自傳送通道的啟動要求。

若要啟動接收端通道程式，必須啟動接聽器程式，以偵測送入的網路要求並啟動相關聯的通道。您可以使用 RUNMQLSR 指令來啟動此接聽器程式，並提供要接聽的 TpName。或者，您可以在 SNA Server for Windows 下使用 TpStart。

使用 RUNMQLSR 指令

用於啟動接聽器的指令範例：

```
RUNMQLSR -t LU62 -n RECV [-m QMNAME]
```

其中 RECV 是在另一個 (傳送中) 端指定為「TpName」的 TpName。方括弧中的最後一部分是選用的，預設佇列管理程式不需要。

可以在一部機器上執行多個佇列管理程式。您必須將不同的 TpName 指派給每一個佇列管理程式，然後為每一個佇列管理程式啟動接聽器程式。例如：

```
RUNMQLSR -t LU62 -m QM1 -n TpName1
RUNMQLSR -t LU62 -m QM2 -n TpName2
```

為了取得最佳效能，請依照 [將通道和接聽器當作授信應用程式來執行中的說明](#)，將 WebSphere MQ 接聽器當作授信應用程式來執行。如需授信應用程式的相關資訊，請參閱 [授信應用程式的限制](#)。

您可以使用下列指令，停止在非作用中佇列管理程式上執行的所有 WebSphere MQ 接聽器：

```
ENDMQLSR [-m QMNAME]
```

在 Windows 上使用 Microsoft SNA Server

您可以使用 TpSetup (從 SNA 伺服器 SDK) 來定義可呼叫的 TP，然後驅動 amqcrs6a.exe，也可以手動設定各種登錄值。應該傳遞至 amqcrs6a.exe 的參數如下：

```
-m QM -n TpName
```

其中 QM 是佇列管理程式名稱，而 TpName 是 TP 名稱。如需相關資訊，請參閱 [微軟 SNA 伺服器 APPC 程式設計師手冊](#) 或 [微軟 SNA 伺服器 CPI-C 程式設計師手冊](#)。

如果您未指定佇列管理程式名稱，則會採用預設佇列管理程式。

在 Windows 上定義 NetBIOS 連線

WebSphere MQ 在建立與另一個 WebSphere MQ 產品的 NetBIOS 連線時，使用三種類型的 NetBIOS 資源：階段作業、指令及名稱。每一個資源都有限制，依預設或在安裝 NetBIOS 期間選擇建立。

每一個執行中通道 (不論類型為何) 都會使用一個 NetBIOS 階段作業及一個 NetBIOS 指令。IBM NetBIOS 實作容許多個處理程序使用相同的本端 NetBIOS 名稱。因此，只有一個 NetBIOS 名稱可供 WebSphere MQ 使用。其他供應商的實作 (例如 Novell 的 NetBIOS 模擬) 需要每個處理程序不同的本端名稱。請從您正在使用之 NetBIOS 產品的說明文件中驗證您的需求。

在所有情況下，請確定每一種類型都有足夠的資源可用，或增加配置中指定的上限。對值所做的任何變更都需要重新啟動系統。

在系統啟動期間，NetBIOS 裝置驅動程式會顯示可供應用程式使用的階段作業、指令及名稱數目。這些資源可供在相同系統上執行的任何 NetBIOS 型應用程式使用。因此，在 WebSphere MQ 需要獲得這些資源之前，其他應用程式可能會耗用這些資源。您的 LAN 網路管理者應該能夠為您釐清這一點。

相關概念

第 75 頁的『[定義 IBM WebSphere MQ 本端 NetBIOS 名稱](#)』

IBM WebSphere MQ 通道處理程序使用的本端 NetBIOS 名稱有三種指定方式。

第 75 頁的『[建立佇列管理程式 NetBIOS 階段作業、指令及名稱限制](#)』

NetBIOS 階段作業、指令及名稱的佇列管理程式限制可以用兩種方式來指定。

第 76 頁的『[建立 LAN 配接卡號碼](#)』

若要讓通道在 NetBIOS 中順利運作，每一端的配接卡支援必須相容。IBM WebSphere MQ 可讓您使用 qm.ini 檔案的 NETBIOS 段落中的 AdapterNum 值，並在 runmqsr 指令上指定 -a 參數，來控制 LAN 配接卡 (LANA) 號碼的選擇。

第 76 頁的『[起始 NetBIOS 連線](#)』

定義起始連線所需的步驟。

第 76 頁的『[NetBIOS 連線的目標接聽器](#)』

定義要在 NetBIOS 連線接收端執行的步驟。

定義 IBM WebSphere MQ 本端 NetBIOS 名稱

IBM WebSphere MQ 通道處理程序使用的本端 NetBIOS 名稱有三種指定方式。

依優先順序，有三種方式：

1. RUNMQSR 指令的 -l 參數中指定的值，例如：

```
RUNMQSR -t NETBIOS -l my_station
```

2. 具有指令所建立值的 MQNAME 環境變數：

```
SET MQNAME=my_station
```

您可以設定每一個處理程序的 MQNAME 值。或者，您可以在 Windows 登錄中的系統層次設定它。

如果您使用需要唯一名稱的 NetBIOS 實作，則必須在每一個啟動 IBM WebSphere MQ 處理程序的視窗中發出 SET MQNAME 指令。MQNAME 值是任意的，但對於每一個處理程序必須是唯一的。

3. 佇列管理程式配置檔 qm.ini 中的 NETBIOS 段落。例如：

```
NETBIOS:  
LocalName=my_station
```

註：

1. 由於支援 NetBIOS 產品實作的變異，建議您讓每一個 NetBIOS 名稱在網路中是唯一的。如果沒有，可能會發生無法預期的結果。如果您在建立 NetBIOS 通道時發生問題，且佇列管理程式錯誤日誌中有錯誤訊息顯示 NetBIOS 回覆碼 X'15'，請檢閱您使用的 NetBIOS 名稱。
2. 在 Windows 上，您無法使用機器名稱作為 NetBIOS 名稱，因為 Windows 已使用它。
3. 傳送端通道起始需要使用 MQNAME 環境變數或 qm.ini 檔案中的 LocalName 來指定 NetBIOS 名稱。

建立佇列管理程式 NetBIOS 階段作業、指令及名稱限制

NetBIOS 階段作業、指令及名稱的佇列管理程式限制可以用兩種方式來指定。

依優先順序，這些方式如下：

1. RUNMQSR 指令中指定的值：

```
-s Sessions
-e Names
-o Commands
```

如果指令中未指定 `-m` 運算元，則值只會套用至預設佇列管理程式。

2. 佇列管理程式配置檔 `qm.ini` 中的 NETBIOS 段落。例如：

```
NETBIOS:

NumSess=Qmgr_max_sess
NumCmds=Qmgr_max_cmds
NumNames=Qmgr_max_names
```

建立 LAN 配接卡號碼

若要讓通道在 NetBIOS 中順利運作，每一端的配接卡支援必須相容。IBM WebSphere MQ 可讓您使用 `qm.ini` 檔案的 NETBIOS 段落中的 `AdapterNum` 值，並在 `runmqslr` 指令上指定 `-a` 參數，來控制 LAN 配接卡 (LANA) 號碼的選擇。

IBM WebSphere MQ 用於 NetBIOS 連線的預設 LAN 配接卡號碼是 0。驗證系統上使用的號碼，如下所示：

在 Windows 上，無法透過作業系統直接查詢 LAN 配接卡號碼。相反地，您可以使用 `LANACFG.EXE` 指令行公用程式，可從 Microsoft 取得。工具的輸出會顯示虛擬 LAN 配接卡號碼及其有效連結。如需 LAN 配接卡號碼的進一步相關資訊，請參閱 Microsoft 知識庫文章 138037 *HOWTO: 在 32 位元環境中使用 LANA 號碼*。

請在佇列管理程式配置檔 `qm.ini` 的 NETBIOS 段落中指定正確值。

```
NETBIOS:
AdapterNum=n
```

其中 `n` 是此系統的正确 LAN 配接卡號碼。

起始 NetBIOS 連線

定義起始連線所需的步驟。

若要起始連線，請在傳送端遵循下列步驟：

1. 使用 `MQNAME` 或 `LocalName` 值定義 NetBIOS 工作站名稱。
2. 請驗證系統上正在使用的 LAN 配接卡號碼，並使用 `AdapterNum` 指定正確的檔案。
3. 在通道定義的 `ConnectionName` 欄位中，指定目標接聽器程式所使用的 NetBIOS 名稱。在 Windows 上，NetBIOS 通道 **必須** 以執行緒形式執行。透過在通道定義中指定 `MCATYPE (THREAD)` 來執行此動作。

```
DEFINE CHANNEL (chname) CHLTYPE(SDR) +
TRPTYPE(NETBIOS) +
CONNNAME(your_station) +
XMITQ(xmitq) +
MCATYPE(THREAD) +
REPLACE
```

NetBIOS 連線的目標接聽器

定義要在 NetBIOS 連線接收端執行的步驟。

在接收端，遵循下列步驟：

1. 使用 `MQNAME` 或 `LocalName` 值定義 NetBIOS 工作站名稱。
2. 請驗證系統上正在使用的 LAN 配接卡號碼，並使用 `AdapterNum` 指定正確的檔案。
3. 定義接收端通道：

```
DEFINE CHANNEL (chname) CHLTYPE(RCVR) +
```



```
TRPTYPE (NETBIOS) +  
REPLACE
```

4. 啟動 WebSphere MQ 接聽器程式以建立工作站並讓它可以聯絡。例如：

```
RUNMQLSR -t NETBIOS -l your_station [-m qmgr]
```

此指令會將 `your_station` 建立為等待聯絡的 NetBIOS 工作站。NetBIOS 工作站名稱在整個 NetBIOS 網路中必須是唯一的。

為了取得最佳效能，請依照第 62 頁的『將通道和接聽器當作授信應用程式來執行』中的說明，將 WebSphere MQ 接聽器當作授信應用程式來執行。如需授信應用程式的相關資訊，請參閱 [授信應用程式的限制](#)。

您可以使用下列指令，停止在非作用中佇列管理程式上執行的所有 WebSphere MQ 接聽器：

```
ENDMQLSR [-m QMNAME]
```

如果您未指定佇列管理程式名稱，則會採用預設佇列管理程式。

在 Windows 上定義 SPX 連線

SPX 連線僅適用於執行 Windows XP 及 Windows 2003 Server 的用戶端及伺服器。

傳送端的通道定義指定目標的位址。接聽器程式必須在接收端執行。

相關概念

第 77 頁的『在 SPX 上傳送結束』

如果目標機器是遠端，請在通道定義的連線名稱欄位中指定目標機器的 SPX 位址。

第 78 頁的『在 SPX 上接收』

接收通道程式會啟動，以回應來自傳送通道的啟動要求。

第 79 頁的『IPX/SPX 參數』

在大部分情況下，IPX/SPX 參數的預設值將符合您的需求。不過，您可能需要在環境中修改其中部分，以調整其對 WebSphere MQ 的使用。

在 SPX 上傳送結束

如果目標機器是遠端，請在通道定義的連線名稱欄位中指定目標機器的 SPX 位址。

SPX 位址以下列格式指定：

```
network.node(socket)
```

其中：

network

是遠端機器所在網路的 4 位元組網址，

node

是 6 位元組節點位址，這是遠端機器中 LAN 配接卡的 LAN 位址

socket

是遠端機器在其上接聽的 2 位元組 Socket 號碼。

如果本端與遠端機器位於相同的網路上，則不需要指定網址。如果遠端系統正在預設 Socket (5E86) 上接聽，則不需要指定 Socket。

MQSC 指令的 CONNAME 參數中指定的完整指定 SPX 位址範例如下：

```
CONNAME('00000001.08005A7161E5(5E87)')
```

在預設情況下，當機器兩者都在相同網路上時，這會變成：

```
CONNNAME (08005A7161E5)
```

您可以在佇列管理程式配置檔 (qm.ini) 中指定預設 Socket 號碼來變更它：

```
SPX:  
Socket=5E87
```

如需您使用 qm.ini 所設定值的相關資訊，請參閱 [分散式佇列的配置檔段落](#)。

在 SPX 上接收

接收通道程式會啟動，以回應來自傳送通道的啟動要求。

若要啟動接收端通道程式，必須啟動接聽器程式，以偵測送入的網路要求並啟動相關聯的通道。

使用 WebSphere MQ 接聽器。

使用 SPX 接聽器待辦事項選項

在 SPX 上接收時，會設定未完成的連線要求數目上限。這可以視為在 SPX 埠上等待接聽器接受要求的要求待辦事項。預設接聽器待辦事項值顯示在 [第 78 頁的表 10](#) 中。

平台	預設接聽器待辦事項值
Windows 伺服器	5
Windows 工作站	5

如果待辦事項達到 [第 78 頁的表 10](#) 中的值，當嘗試使用 MQCONN 或 MQCONNX 連接至佇列管理程式時，會收到原因碼 MQRC_Q_MGR_NOT_AVAILABLE。如果發生這種情況，則可以嘗試重新連接。

不過，若要避免此錯誤，您可以在 qm.ini 檔案或 Windows 的登錄中新增項目：

```
SPX:  
ListenerBacklog = n
```

這會置換 SPX 接聽器未完成要求的預設數目上限 (請參閱 [第 78 頁的表 10](#))。

註：部分作業系統支援大於預設值的值。必要的話，這可用來避免達到連線限制。

如果要在開啟 backlog 選項的情況下執行接聽器，請執行下列步驟：

- 使用 RUNMQLSR -b 指令，或
- 使用 MQSC 指令 **DEFINE LISTENER**，並將 BACKLOG 屬性設為必要值。

如需 **RUNMQLSR** 指令的相關資訊，請參閱 [runmqlsr](#)。如需 **DEFINE LISTENER** 指令的相關資訊，請參閱 [DEFINE LISTENER](#)。

使用 WebSphere MQ 接聽器

若要執行隨 WebSphere MQ 提供的接聽器 (以執行緒方式啟動新通道)，請使用 RUNMQLSR 指令。例如：

```
RUNMQLSR -t spx [-m QMNAME] [-x 5E87]
```

方括弧指出選用參數；預設佇列管理程式不需要 QMNAME，如果您使用預設值 (5E86)，則不需要 Socket 號碼。

為了取得最佳效能，請依照 [第 62 頁的『將通道和接聽器當作授信應用程式來執行』](#) 中的說明，將 WebSphere MQ 接聽器當作授信應用程式來執行。如需授信應用程式的相關資訊，請參閱 [授信應用程式的限制](#)。

您可以使用下列指令，停止在非作用中佇列管理程式上執行的所有 WebSphere MQ 接聽器：

```
ENDMQLSR [-m QMNAME]
```

如果您未指定佇列管理程式名稱，則會採用預設佇列管理程式。

IPX/SPX 參數

在大部分情況下，IPX/SPX 參數的預設值將符合您的需求。不過，您可能需要在環境中修改其中部分，以調整其對 WebSphere MQ 的使用。

實際參數及變更它們的方法會根據 SPX 通訊支援的平台及提供者而有所不同。範例小節說明其中部分參數，特別是可能影響 WebSphere MQ 通道及用戶端連線作業的參數。

Windows 系統

如需 NWLink IPX 及 SPX 參數的使用及設定的完整資料，請參閱 Microsoft 文件。IPX/SPX 參數位於登錄中的下列路徑中：

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service\NWLinkSPX\Parameters  
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Service\NWLinkIPX\Parameters
```

在 UNIX and Linux 系統上設定通訊

DQM 是 IBM WebSphere MQ 的遠端佇列作業機能。它為佇列管理程式提供通道控制程式，這些佇列管理程式形成通訊鏈結的介面，可由系統操作員控制。分散式佇列管理所保留的通道定義會使用這些連線。

當啟動分散式佇列管理通道時，它會嘗試使用通道定義中指定的連線。若要成功，連線必須定義且可用。本節說明如何執行此動作。您也可能會發現參閱下列各節會很有用：

- [配置範例- IBM WebSphere MQ for AIX](#)
- [配置範例- IBM WebSphere MQ for HP-UX](#)
- [配置範例- IBM WebSphere MQ for Solaris](#)
- [配置範例- IBM WebSphere MQ for Linux](#)

若為 Windows，請參閱第 71 頁的『設定 Windows 的通訊』。

在 UNIX and Linux 系統上，您可以選擇兩種 WebSphere MQ 通訊形式：

- [第 79 頁的『在 UNIX and Linux 上定義 TCP 連線』](#)
- [第 83 頁的『在 UNIX and Linux 上定義 LU 6.2 連線』](#)

每一個通道定義只能指定一個作為傳輸通訊協定 (傳輸類型) 屬性。佇列管理程式可以使用一或多個通訊協定。

對於 IBM WebSphere MQ Explorer MQI 用戶端，具有使用不同傳輸通訊協定的替代通道可能非常有用。如需 IBM WebSphere MQ Explorer MQI 用戶端的相關資訊，請參閱 [IBM WebSphere MQ MQI 用戶端概觀](#)。

相關概念

[第 23 頁的『使用分散式佇列連接應用程式』](#)

本節提供 WebSphere MQ 安裝之間交互通訊的更詳細資訊，包括佇列定義、通道定義、觸發及同步點程序

[第 63 頁的『在 UNIX, Linux, and Windows 上監視及控制通道』](#)

對於 DQM，您需要建立、監視及控制遠端佇列管理程式的通道。您可以使用指令、程式、IBM WebSphere MQ Explorer、通道定義的檔案，以及同步化資訊的儲存區來控制通道。

[第 84 頁的『配置用戶端與伺服器之間的連線』](#)

若要配置 WebSphere MQ MQI 用戶端與伺服器之間的通訊鏈結，請決定您的通訊協定，定義鏈結兩端的連線，啟動接聽器，以及定義通道。

在 UNIX and Linux 上定義 TCP 連線

傳送端的通道定義指定目標的位址。接聽器或 inet 常駐程式是針對接收端的連線所配置。

傳送端

在通道定義的連線名稱欄位中，指定目標機器的主機名稱或 TCP 位址。要連接的埠預設為 1414。埠號 1414 由「網際網路指派號碼」權限指派給 WebSphere MQ。

若要使用非預設埠號，請變更連線名稱欄位：

```
Connection Name REMHOST(1822)
```

其中 REMHOST 是遠端機器的主機名稱，1822 是所需的埠號。(這必須是接收端接聽器接聽所在的埠。) 或者，您可以在佇列管理程式配置檔 (qm.ini) 中指定埠號來變更埠號：

```
TCP:  
Port=1822
```

如需您使用 qm.ini 所設定值的相關資訊，請參閱 [分散式佇列的配置檔段落](#)。

在 TCP 上接收

您可以使用 TCP/IP 接聽器 (inet 常駐程式 (inetd)) 或 WebSphere MQ 接聽器。

部分 Linux 發行套件現在使用延伸 inet 常駐程式 (xinetd)，而非 inet 常駐程式。如需如何在 Linux 系統上使用延伸 inet 常駐程式的相關資訊，請參閱 [在 Linux 上建立 TCP 連線](#)。

相關概念

[第 80 頁的『使用 TCP/IP 接聽器』](#)

若要在 UNIX and Linux 上啟動通道，必須編輯 /etc/services 檔案及 inetd.conf 檔案

[第 81 頁的『使用 TCP 接聽器待辦事項選項』](#)

在 TCP 中，除非在伺服器與用戶端之間進行三向信號交換，否則會將連線視為不完整。這些連線稱為未完成的要求。這些未完成的要求會設定上限值，且可視為在 TCP 埠上等待接聽器接受要求的要求待辦事項。

[第 82 頁的『使用 WebSphere MQ 接聽器』](#)

若要執行隨 WebSphere MQ 提供的接聽器 (以執行緒方式啟動新通道)，請使用 runmqclsr 指令。

[第 82 頁的『使用 TCP/IP SO_KEEPALIVE 選項』](#)

在部分 UNIX and Linux 系統上，您可以定義 TCP 在檢查連線是否仍然可用之前等待的時間長度，以及它在第一次檢查失敗時再次嘗試連線的頻率。這是核心可調整參數，或可以在指令行輸入。

使用 TCP/IP 接聽器

若要在 UNIX and Linux 上啟動通道，必須編輯 /etc/services 檔案及 inetd.conf 檔案

請遵循下列指示：

1. 編輯 /etc/services 檔案：

註：若要編輯 /etc/services 檔案，您必須以超級使用者或 root 身分登入。您可以變更此項，但它必須符合傳送端指定的埠號。

將下面這一行加到檔案中：

```
MQSeries 1414/tcp
```

其中 1414 是 WebSphere MQ 所需的埠號。埠號不可超過 65535。

2. 在 inetd.conf 檔案中新增一行以呼叫程式 amqcrsta，其中 MQ_INSTALLATION_PATH 代表 WebSphere MQ 安裝所在的高階目錄：

```
MQSeries stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta  
[-m Queue_Man_Name]
```

在 inetd 重新讀取配置檔之後，更新會在作用中。若要執行此動作，請從 root 使用者 ID 發出下列指令：

- 在 AIX 上:

```
refresh -s inetd
```

- 在 HP-UX 上, 從 mqm 使用者 ID:

```
inetd -c
```

- 在 Solaris 10 或更新版本上:

```
inetconv
```

- 在其他 UNIX and Linux 系統 (包括 Solaris 9) 上:

```
kill -1 <process number>
```

inetd 啟動的接聽器程式從 inetd 繼承語言環境時, 可能不允許使用 (合併) MQMDE, 並將其作為訊息資料放置在佇列上。若要確保允許使用 MQMDE, 您必須正確地設定語言環境。inetd 所設定的語言環境可能不符合 WebSphere MQ 處理程序所使用的其他語言環境所選擇的語言環境。如果要設定語言環境, 請執行下列動作:

1. 建立 Shell Script, 將語言環境變數 LANG、LC_COLLATE、LC_CTYPE、LC_MONETARY、LC_NUMERIC、LC_TIME 及 LC_MESSAGES 設為用於其他 WebSphere MQ 處理程序的語言環境。
2. 在相同的 Shell Script 中, 呼叫接聽器程式。
3. 修改 inetd.conf 檔案, 以呼叫您的 Shell Script 來取代接聽器程式。

伺服器上可以有多個佇列管理程式。您必須針對每一個佇列管理程式, 在這兩個檔案中各新增一行。例如:

```
MQSeries1    1414/tcp
MQSeries2    1822/tcp
```

```
MQSeries2 stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta -m QM2
```

其中 MQ_INSTALLATION_PATH 代表 WebSphere MQ 安裝所在的高階目錄。

如果在單一 TCP 埠上排入佇列的未完成連線要求數目有限制, 這可避免產生錯誤訊息。如需未完成的連線要求數的相關資訊, 請參閱 [第 81 頁的『使用 TCP 接聽器待辦事項選項』](#)。

使用 TCP 接聽器待辦事項選項

在 TCP 中, 除非在伺服器與用戶端之間進行三向信號交換, 否則會將連線視為不完整。這些連線稱為未完成的連線要求。這些未完成的連線要求會設定上限值, 且可視為在 TCP 埠上等待接聽器接受要求的要求待辦事項。

預設接聽器待辦事項值顯示在 [第 81 頁的表 11](#) 中。

表 11: 在 TCP/IP 埠上佇列的未完成連線要求數上限	
伺服器平台	連線要求數上限
AIX	100
HP-UX	20
Linux	100
IBM i	255
Solaris	100

表 11: 在 TCP/IP 埠上佇列的未完成連線要求數上限 (繼續)	
伺服器平台	連線要求數上限
Windows 伺服器	100
Windows 工作站	100
z/OS	255

如果待辦事項達到 [第 81 頁的表 11](#) 中顯示的值，則會拒絕 TCP/IP 連線，且通道無法啟動。

對於 MCA 通道，這會導致通道進入 RETRY 狀態，並在稍後重試連線。

不過，若要避免此錯誤，您可以在 qm.ini 檔案中新增項目：

```
TCP:
ListenerBacklog = n
```

這會置換 TCP/IP 接聽器的未完成要求預設數目上限 (請參閱 [第 81 頁的表 11](#))。

註：部分作業系統支援大於預設值的值。必要的話，可以使用此值來避免達到連線限制。

如果要在開啟 backlog 選項的情況下執行接聽器，請執行下列步驟：

- 使用 `runmqclsr -b` 指令，或
- 使用 MQSC 指令 **DEFINE LISTENER**，並將 BACKLOG 屬性設為必要值。

如需 `runmqclsr` 指令的相關資訊，請參閱 [runmqclsr](#)。如需 DEFINE LISTENER 指令的相關資訊，請參閱 [DEFINE LISTENER](#)。

使用 WebSphere MQ 接聽器

若要執行隨 WebSphere MQ 提供的接聽器 (以執行緒方式啟動新通道)，請使用 `runmqclsr` 指令。

例如：

```
runmqclsr -t tcp [-m QMNAME] [-p 1822]
```

方括弧指出選用參數；預設佇列管理程式不需要 QMNAME，如果您使用預設值 (1414)，則不需要埠號。埠號不可超過 65535。

為了取得最佳效能，請依照 [第 62 頁的『將通道和接聽器當作授信應用程式來執行』](#) 中的說明，將 WebSphere MQ 接聽器當作授信應用程式來執行。如需授信應用程式的相關資訊，請參閱 [授信應用程式的限制](#)。

您可以使用下列指令，停止在非作用中佇列管理程式上執行的所有 WebSphere MQ 接聽器：

```
endmqclsr [-m QMNAME]
```

如果您未指定佇列管理程式名稱，則會採用預設佇列管理程式。

使用 TCP/IP SO_KEEPALIVE 選項

在部分 UNIX and Linux 系統上，您可以定義 TCP 在檢查連線是否仍然可用之前等待的時間長度，以及它在第一次檢查失敗時再次嘗試連線的頻率。這是核心可調整參數，或可以在指令行輸入。

如果您想要使用 SO_KEEPALIVE 選項 (如需相關資訊，請參閱 [第 52 頁的『正在檢查通道的另一端是否仍然可用』](#))，您必須將下列項目新增至佇列管理程式配置檔 (qm.ini)：

```
TCP:
KeepAlive=yes
```

如需相關資訊，請參閱 UNIX and Linux 系統的說明文件。

在 UNIX and Linux 上定義 LU 6.2 連線

必須配置 SNA，才能在兩部機器之間建立 LU 6.2 交談。

如需透過 TCP/IP 配置 SNA 的最新資訊，請參閱下列線上 IBM 說明文件: [Communications Server](#)。

必須配置 SNA，才能在兩個系統之間建立 LU 6.2 交談。

如需相關資訊，請參閱 *Multiplatform APPC Configuration Guide* 及下表。

遠端平台	TPNAME	TPPATH
z/OS, 不含 CICS	與遠端佇列管理程式相關資訊中的對應 TPName 相同。	-
z/OS 使用 CICS	CKRC (傳送端) CKSV (要求端) CKRC (伺服器)	-
IBM i	與 IBM i 系統上遞送登錄中的比較值相同。	-
UNIX and Linux 系統	與遠端佇列管理程式相關資訊中的對應 TPName 相同。	<code>MQ_INSTALLATION_PATH/bin/amqcrs6a</code>
Windows	如 Windows 執行接聽器指令中所指定, 或在 Windows 上使用 TpSetup 定義的可呼叫交易程式。	<code>MQ_INSTALLATION_PATH\bin\amqcrs6a</code>

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

如果您在同一部機器上有多個佇列管理程式，請確定通道定義中的 TPname 是唯一的。

相關概念

第 83 頁的『傳送端』

在 UNIX and Linux 系統上，建立 CPI-C 端物件 (符號式目的地)，並在通道定義的「連線名稱」欄位中輸入此名稱。另請建立指向夥伴的 LU 6.2 鏈結。

第 83 頁的『在 LU 6.2 上接收』

在 UNIX and Linux 系統上，於接收端建立接聽附件、LU 6.2 邏輯連線設定檔及 TPN 設定檔。

傳送端

在 UNIX and Linux 系統上，建立 CPI-C 端物件 (符號式目的地)，並在通道定義的「連線名稱」欄位中輸入此名稱。另請建立指向夥伴的 LU 6.2 鏈結。

在 CPI-C 端物件中，輸入接收機器上的友機 LU 名稱、交易程式名稱及模式名稱。例如：

```
Partner LU Name          REMHOST
Remote TP Name           recv
Service Transaction Program no
Mode Name                #INTER
```

在 HP-UX 上，請使用 `APPCLLU` 環境變數來命名傳送者應該使用的本端 LU。在 Solaris 上，將 `APPC_LOCAL_LU` 環境變數設為本端 LU 名稱。

當 WebSphere MQ 嘗試建立 SNA 階段作業時，會在 CPI-C 支援的情況下使用 SECURITY PROGRAM。

在 LU 6.2 上接收

在 UNIX and Linux 系統上，於接收端建立接聽附件、LU 6.2 邏輯連線設定檔及 TPN 設定檔。

在 TPN 設定檔中，輸入執行檔的完整路徑及「交易程式」名稱：

```
Full path to TPN executable  MQ_INSTALLATION_PATH/bin/amqcrs6a
Transaction Program name     recv
User ID                       0
```

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

在您可以設定使用者 ID 的系統上，指定屬於 mqm 群組成員的使用者。在 AIX、Solaris 及 HP-UX 上，設定 APPCTPN (交易名稱) 及 APPCLLU (本端 LU 名稱) 環境變數 (您可以使用所呼叫交易程式的配置畫面)。

您可能需要使用預設佇列管理程式以外的佇列管理程式。若是如此，請定義一個指令檔來呼叫：

```
amqcrs6a -m Queue_Man_Name
```

然後呼叫指令檔。

配置用戶端與伺服器之間的連線

若要配置 WebSphere MQ MQI 用戶端與伺服器之間的通訊鏈結，請決定您的通訊協定，定義鏈結兩端的連線，啟動接聽器，以及定義通道。

在 WebSphere MQ 中，物件之間的邏輯通訊鏈結稱為通道。用來將 WebSphere MQ MQI 用戶端連接至伺服器的通道稱為 MQI 通道。您可以在鏈結每一端設定通道定義，讓 WebSphere MQ MQI 用戶端上的 WebSphere MQ 應用程式可以與伺服器上的佇列管理程式進行通訊。如需如何執行此動作的詳細說明，請參閱 [使用者定義的通道](#)。

在定義 MQI 通道之前，您必須：

1. 決定您要使用的通訊形式。請參閱第 84 頁的『要使用的通訊類型』。
2. 定義通道每一端的連線：

若要定義連線，您必須：

- 配置連線。
- 記錄通道定義所需的參數值。
- 啟動接聽器，讓伺服器能夠偵測來自 WebSphere MQ MQI 用戶端的送入網路要求。

要使用的通訊類型

不同的平台支援不同的傳輸通訊協定。您選擇的傳輸通訊協定取決於 WebSphere MQ MQI 用戶端與伺服器平台的組合。

視您的用戶端及伺服器平台而定，MQI 通道最多有四種傳輸通訊協定類型：

- LU 6.2
- NetBIOS
- SPX
- TCP/IP

當您定義 MQI 通道時，每一個通道定義都必須指定傳輸通訊協定 (傳輸類型) 屬性。伺服器不受限於一個通訊協定，因此不同的通道定義可以指定不同的通訊協定。對於 WebSphere MQ MQI 用戶端，具有使用不同傳輸通訊協定的替代 MQI 通道可能非常有用。

您選擇的傳輸通訊協定可能受到 WebSphere MQ MQI 用戶端及伺服器平台的特定組合所限制。下表顯示可能的組合。

傳輸通訊協定	WebSphere MQ MQI 用戶端	WebSphere MQ 伺服器
TCP/IP	UNIX 系統 Windows	UNIX 系統 Windows z/OS
LU 6.2	UNIX 系統 ¹ Windows	UNIX 系統 ¹ Windows

表 13: 傳輸通訊協定- WebSphere MQ MQI 用戶端和伺服器平台的組合 (繼續)		
傳輸通訊協定	WebSphere MQ MQI 用戶端	WebSphere MQ 伺服器
NetBIOS	Windows	Windows
SPX	Windows	Windows
註: 1. 除了 Linux for Power Systems 之外		

如需設定不同類型連線的相關資訊，請參閱下列鏈結：

- [第 72 頁的『在 Windows 上定義 TCP 連線』](#)
- [第 79 頁的『在 UNIX and Linux 上定義 TCP 連線』](#)
- [第 87 頁的『TCP/IP 連線限制』](#)
- [第 73 頁的『在 Windows 上定義 LU 6.2 連線』](#)
- [第 83 頁的『在 UNIX and Linux 上定義 LU 6.2 連線』](#)
- [第 74 頁的『在 Windows 上定義 NetBIOS 連線』](#)
- [第 77 頁的『在 Windows 上定義 SPX 連線』](#)

相關概念

[第 87 頁的『配置延伸交易式用戶端』](#)

這個主題集合說明如何配置每一個交易管理程式種類的延伸交易式功能。

[第 95 頁的『定義 MQI 通道』](#)

若要建立新的通道，您必須使用相同的通道名稱及相容的通道類型來建立 **兩個** 通道定義，每個連線端都有一個通道定義。在此情況下，通道類型為 *server-connection* 及 *client-connection*。

[第 96 頁的『在不同平台上建立伺服器連線和用戶端連線定義』](#)

您可以在套用它的電腦上建立每一個通道定義。在用戶端電腦上建立通道定義的方式有一些限制。

[第 99 頁的『在伺服器上建立伺服器連線及用戶端連線定義』](#)

您可以在伺服器上建立這兩個定義，然後讓用戶端可以使用用戶端連線定義。

[第 103 頁的『MQI 通道的通道結束程式』](#)

在 UNIX、Linux 及 Windows 系統上，WebSphere MQ MQI 用戶端環境可以使用三種類型的通道結束程式。

[第 106 頁的『將用戶端連接至佇列共用群組』](#)

在屬於佇列共用群組成員的伺服器上，您可以在用戶端與佇列管理程式之間建立 MQI 通道，以將用戶端連接至佇列共用群組。

[第 107 頁的『使用配置檔來配置用戶端』](#)

使用文字檔中的屬性來配置用戶端。這些屬性可以由環境變數置換，或以其他平台專用方式置換。

相關工作

[將 IBM MQ MQI 用戶端應用程式連接至佇列管理程式](#)

相關參考

[DISPLAY CHLAUTH](#)

[SET CHLAUTH](#)

要使用的通訊類型

不同的平台支援不同的通訊協定。您選擇的傳輸通訊協定取決於 WebSphere MQ MQI 用戶端與伺服器平台的組合。

在不同平台上，MQI 通道有四種通訊類型：

- LU 6.2
- NetBIOS
- SPX
- TCP/IP

當您定義 MQI 通道時，每一個通道定義都必須指定傳輸通訊協定 (傳輸類型) 屬性。伺服器不受限於一個通訊協定，因此不同的通道定義可以指定不同的通訊協定。對於 WebSphere MQ MQI 用戶端，具有使用不同傳輸通訊協定的替代 MQI 通道可能非常有用。

您選擇的傳輸通訊協定也取決於 WebSphere MQ 用戶端和伺服器平台的特定組合。下表顯示可能的組合。

傳輸通訊協定	WebSphere MQ MQI 用戶端	WebSphere MQ 伺服器
TCP/IP	UNIX 系統 Windows	UNIX 系統 Windows
LU 6.2	UNIX 系統 ¹ Windows	UNIX 系統 ¹ Windows
NetBIOS	Windows	Windows
SPX	Windows	Windows

註：
1. 除了 Linux (POWER 平台)

相關概念

第 72 頁的『[在 Windows 上定義 TCP 連線](#)』

在傳送端配置通道以指定目標位址，並在接收端執行接聽器程式，以定義 TCP 連線。

第 79 頁的『[在 UNIX and Linux 上定義 TCP 連線](#)』

傳送端的通道定義指定目標的位址。接聽器或 inet 常駐程式是針對接收端的連線所配置。

第 73 頁的『[在 Windows 上定義 LU 6.2 連線](#)』

必須配置 SNA，才能在兩部機器之間建立 LU 6.2 交談。

第 83 頁的『[在 UNIX and Linux 上定義 LU 6.2 連線](#)』

必須配置 SNA，才能在兩部機器之間建立 LU 6.2 交談。

第 74 頁的『[在 Windows 上定義 NetBIOS 連線](#)』

WebSphere MQ 在建立與另一個 WebSphere MQ 產品的 NetBIOS 連線時，使用三種類型的 NetBIOS 資源：階段作業、指令及名稱。每一個資源都有限制，依預設或在安裝 NetBIOS 期間選擇建立。

第 77 頁的『[在 Windows 上定義 SPX 連線](#)』

SPX 連線僅適用於執行 Windows XP 及 Windows 2003 Server 的用戶端及伺服器。

相關參考

第 87 頁的『[TCP/IP 連線限制](#)』

可以在單一 TCP/IP 埠上排入佇列的未完成連線要求數，視平台而定。如果達到限制，則會發生錯誤。

定義 TCP/IP 連線

在用戶端的通道定義上指定傳輸類型 TCP。在伺服器上啟動接聽器程式。

在通道定義上指定傳輸類型 TCP，以在用戶端指定 TCP/IP 連線。

接收通道程式會啟動，以回應來自傳送通道的啟動要求。若要這樣做，必須啟動接聽器程式，以偵測送入的網路要求並啟動相關聯的通道。啟動接聽器程式的程序視伺服器平台而定。

請參閱用戶端和伺服器平台的相關主題。

TCP/IP 連線限制

可以在單一 TCP/IP 埠上排入佇列的未完成連線要求數，視平台而定。如果達到限制，則會發生錯誤。

此連線限制與您可以連接至 IBM WebSphere MQ 伺服器的用戶端數目上限不同。您可以將更多用戶端連接至伺服器，最高到伺服器系統資源所決定的層次。下表顯示連線要求的待辦事項值：

伺服器平台	連線要求數上限
AIX	100
HP-UX	20
Linux	100
IBM	255
Solaris	100
Windows 伺服器	100
Windows 工作站	100
z/OS	255

如果達到連線限制，用戶端會收到來自 MQCONN 呼叫的 MQRC_HOST_NOT_AVAILABLE 回覆碼，以及用戶端錯誤日誌中的 AMQ9202 錯誤 (UNIX and Linux 系統上的 /var/mqm/errors/AMQERR0n.LOG 或 Windows 上 IBM WebSphere MQ 用戶端安裝的錯誤子目錄中的 amqerr0n.log)。如果用戶端重試 MQCONN 要求，則可能成功。

若要增加您可以提出的連線要求數目，並避免由此限制產生錯誤訊息，您可以有多個接聽器，每個接聽器在不同的埠上接聽，或有多個佇列管理程式。

定義 NetBIOS 或 SPX 連線

NetBIOS 及 SPX 連線僅適用於 Windows 系統。

NetBIOS 連線僅適用於執行 Windows 的用戶端及伺服器。請參閱 [定義 NetBIOS 連線](#)。

SPX 連線僅適用於執行 Windows XP 或 Windows 2003 Server 的用戶端及伺服器。請參閱 [定義 SPX 連線](#)。

配置延伸交易式用戶端

這個主題集合說明如何配置每一個交易管理程式種類的延伸交易式功能。

對於每一個平台，延伸交易式用戶端提供下列外部交易管理程式的支援：

符合 XA 標準的交易管理程式

延伸交易式用戶端提供 XA 資源管理程式介面，以支援符合 XA 標準的交易管理程式，例如 CICS 和 Tuxedo。

Microsoft Transaction Server (僅限 Windows 系統)

僅在 Windows 系統上，XA 資源管理程式介面也支援 Microsoft Transaction Server (MTS)。延伸交易式用戶端所提供的 WebSphere MQ MTS 支援提供 MTS 與 XA 資源管理程式介面之間的橋接器。

WebSphere Application Server

舊版 WebSphere MQ 支援 WebSphere Application Server 第 4 版或第 5 版，且需要您執行某些配置作業，才能使用延伸交易式用戶端。WebSphere Application Server 第 6 版以及更新版本包含 WebSphere MQ 傳訊提供者，因此您不需要使用延伸交易式用戶端。

相關概念

第 88 頁的『[配置符合 XA 標準的交易管理程式](#)』

首先配置 WebSphere MQ 基本用戶端，然後使用這些主題中的資訊來配置延伸交易式功能。

第 94 頁的『[Microsoft Transaction Server](#)』

在您可以使用 MTS 作為交易管理程式之前，不需要其他配置。不過，有幾點需要注意。

配置符合 XA 標準的交易管理程式

首先配置 WebSphere MQ 基本用戶端，然後使用這些主題中的資訊來配置延伸交易式功能。

註：本節假設您已基本瞭解「分散式交易處理: XA 規格」中的「開放式群組」所發佈的 XA 介面。

如果要配置延伸交易式用戶端，您必須先依照 [安裝 IBM WebSphere MQ 用戶端](#) 中的說明來配置 WebSphere MQ 基本用戶端。然後，您可以使用本節中的資訊，為符合 XA 標準的交易管理程式 (例如 CICS 和 Tuxedo) 配置延伸交易式功能。

交易管理程式會使用與連接至佇列管理程式的用戶端應用程式所使用的相同 MQI 通道，來與作為資源管理程式的佇列管理程式進行通訊。當交易管理程式發出資源管理程式 (xa_) 函數呼叫時，MQI 通道會用來將呼叫轉遞至佇列管理程式，以及接收來自佇列管理程式的輸出。

交易管理程式可以透過發出 xa_open 呼叫以開啟佇列管理程式作為資源管理程式來啟動 MQI 通道，或者用戶端應用程式可以透過發出 MQCONN 或 MQCONNX 呼叫來啟動 MQI 通道。

- 如果交易管理程式啟動 MQI 通道，且用戶端應用程式稍後在相同執行緒上呼叫 MQCONN 或 MQCONNX，則 MQCONN 或 MQCONNX 呼叫會順利完成，且連線控點會傳回至應用程式。應用程式不會收到 MQRC_ALREADY_CONNECTED 原因碼的 MQCC_WARNING 完成碼。
- 如果用戶端應用程式啟動 MQI 通道，且交易管理程式稍後會在相同執行緒上呼叫 xa_open，則會使用 MQI 通道將 xa_open 呼叫轉遞至佇列管理程式。

在失敗之後的回復狀況下，當沒有任何用戶端應用程式執行時，交易管理程式可以使用專用 MQI 通道來回復失敗時佇列管理程式所參與的任何不完整工作單元。

將延伸交易式用戶端與符合 XA 標準的交易管理程式搭配使用時，請注意下列條件：

- 在單一執行緒內，用戶端應用程式一次只能連接至一個佇列管理程式。只有在使用延伸交易式用戶端時，這項限制才適用；在單一執行緒內，使用 WebSphere MQ 基本用戶端的用戶端應用程式可以同時連接至多個佇列管理程式。
- 用戶端應用程式的每一個執行緒可以連接至不同的佇列管理程式。
- 用戶端應用程式無法使用共用連線控點。

如果要配置延伸交易式功能，您必須針對作為資源管理程式的每一個佇列管理程式，提供下列資訊給交易管理程式：

- xa_open 字串
- XA 交換器結構的指標

當交易管理程式呼叫 xa_open 以將佇列管理程式開啟為資源管理程式時，它會將 xa_open 字串傳遞至延伸交易式用戶端作為呼叫上的引數 xa_info。延伸交易式用戶端以下列方式使用 xa_open 字串中的資訊：

- 啟動伺服器佇列管理程式的 MQI 通道 (如果用戶端應用程式尚未啟動)
- 檢查交易管理程式開啟作為資源管理程式的佇列管理程式是否與用戶端應用程式所連接的佇列管理程式相同
- 如果佇列管理程式使用動態登錄，則尋找交易管理程式的 ax_reg 及 ax_unreg 函數

如需 xa_open 字串的格式，以及延伸交易式用戶端如何使用 xa_open 字串中資訊的詳細資料，請參閱 [第 89 頁的『xa_open 字串的格式』](#)。

XA 交換器結構可讓交易管理程式尋找延伸交易式用戶端所提供的 xa_ 函數，並指定佇列管理程式是否使用動態登錄。如需延伸交易式用戶端所提供 XA 交換器結構的相關資訊，請參閱 [第 92 頁的『XA 交換器結構』](#)。

如需如何為特定交易管理程式配置延伸交易式功能的相關資訊，以及使用交易管理程式與延伸交易式用戶端搭配的任何其他相關資訊，請參閱下列各節：

- [第 93 頁的『配置 CICS 的延伸交易式用戶端』](#)
- [第 94 頁的『配置 Tuxedo 的延伸交易式用戶端』](#)

相關概念

[第 90 頁的『xa_open 字串的 CHANNEL、TRPTYPE、CONNNAME 及 QMNAME 參數』](#)
使用此資訊來瞭解延伸交易式用戶端如何使用這些參數來決定要連接的佇列管理程式。

[第 91 頁的『xa_open 的其他錯誤處理』](#)
xa_open 呼叫在特定情況下失敗。

相關工作

[第 92 頁的『將延伸交易式用戶端與 SSL 通道搭配使用』](#)
您無法使用 xa_open 字串來設定 SSL 通道。請遵循下列指示來使用用戶端通道定義表 (ccdt)。

相關參考

[第 91 頁的『TPM 和 AXLIB 參數』](#)
延伸交易式用戶端使用 TPM 和 AXLIB 參數來尋找交易管理程式的 ax_reg 和 ax_unreg 函數。只有在佇列管理程式使用動態登錄時，才會使用這些功能。

[第 92 頁的『延伸交易式處理程序失敗之後的回復』](#)
在失敗之後，交易管理程式必須能夠回復任何未完成的工作單元。如果要這麼做，交易管理程式必須能夠在失敗時，以資源管理程式身分開啟任何參與不完整工作單元的佇列管理程式。

xa_open 字串的格式

xa_open 字串包含已定義參數名稱和值的配對。

xa_open 字串具有下列格式：

```
parm_name1=parm_value1,parm_name2=parm_value2, ...
```

其中 *parm_name* 是參數的名稱，*parm_value* 是參數的值。參數名稱不區分大小寫，但除非另有說明，否則參數值會區分大小寫。您可以依任何順序來指定參數。

參數的名稱、意義及有效值如下：

姓名

意義和有效值

CHANNEL

MQI 通道的名稱。

這是選用參數。如果提供此參數，則也必須提供 CONNNAME 參數。

TRPTYPE

MQI 通道的通訊協定。有效值是：

LU62

SNA LU 6.2

NETBIOS

NetBIOS

SPX

IPX/SPX

TCP

TCP/IP

這是選用參數。如果省略它，則會採用預設值 TCP。參數的值不區分大小寫。

CONNNAME

位於 MQI 通道伺服器端的佇列管理程式網址。此參數的有效值視 TRPTYPE 參數的值而定：

LU62

識別 CPI-C 端資訊項目的符號式目的地名稱。

友機 LU 的網路完整名稱不是有效值，也不是友機 LU 別名。這是因為沒有其他參數可指定異動程式 (TP) 名稱及模式名稱。

NETBIOS

NetBIOS 名稱。

SPX

4 位元組網址、6 位元組節點位址及選用的 2 位元組 socket 號碼。這些值必須以十六進位表示法指定。句點必須分隔網路和節點位址，並且 Socket 號碼 (如果提供的話) 必須以括弧括住。例如：

```
0a0b0c0d.804abcde23a1(5e86)
```

如果省略 Socket 號碼，則會採用預設值 5e86。

TCP

主機名稱或 IP 位址，後面選擇性地接著以括弧括住的埠號。如果省略埠號，則會採用預設值 1414。

這是選用參數。如果提供此參數，則也必須提供 CHANNEL 參數。

QMNAME

MQI 通道伺服器端的佇列管理程式名稱。名稱不能是空白或單一星號 (*)，名稱也不能以星號開頭。這表示參數必須依名稱來識別特定的佇列管理程式。

這是必要參數。

當用戶端應用程式連接至特定的佇列管理程式時，任何交易回復都必須由相同的佇列管理程式處理。

如果應用程式連接至 z/OS 佇列管理程式，則應用程式可以指定特定佇列管理程式的名稱或佇列共用群組 (QSG) 的名稱。透過使用佇列管理程式名稱或 QSG 名稱，應用程式會控制它是否參與具有 QMGR 回復處置單元或 GROUP 回復處置單元的交易。GROUP 回復單元處置可讓您在 QSG 的任何成員上處理交易的回復。若要使用 GROUP 回復單元，必須啟用 **GROUPUR** 佇列管理程式屬性。

TPM

正在使用的交易管理程式。有效值為 CICS 及 TUXEDO。

延伸交易式用戶端基於相同目的使用此參數及 AXLIB 參數。如需這些參數的相關資訊，請參閱 [TPM](#) 及 [AXLIB](#) 參數。

這是選用參數。參數的值不區分大小寫。

AXLIB

包含交易管理程式的 ax_reg 和 ax_unreg 函數的程式庫名稱。

這是選用參數。

以下是 xa_open 字串的範例：

```
channel=MARS.SVR,trptype=tcp,conname=MARS(1415),qmname=MARS,tpm=cics
```

xa_open 字串的 CHANNEL、TRPTYPE、CONNAME 及 QMNAME 參數

使用此資訊來瞭解延伸交易式用戶端如何使用這些參數來決定要連接的佇列管理程式。

如果在 xa_open 字串中提供 CHANNEL 和 CONNAME 參數，則延伸交易式用戶端會使用這些參數和 TRPTYPE 參數來啟動伺服器佇列管理程式的 MQI 通道。

如果未在 xa_open 字串中提供 CHANNEL 和 CONNAME 參數，則延伸交易式用戶端會使用 MQSERVER 環境變數的值來啟動 MQI 通道。如果未定義 MQSERVER 環境變數，延伸交易式用戶端會使用 QMNAME 參數所識別用戶端通道定義中的項目。

在上述每一種情況下，延伸交易式用戶端會檢查 QMNAME 參數的值是否為 MQI 通道伺服器端的佇列管理程式名稱。如果不是，則 xa_open 呼叫會失敗，且交易管理程式會向應用程式報告失敗。

如果應用程式用戶端連接至 V7.0.1 或更新版本的 z/OS 佇列管理程式，則可以為 QMNAME 參數指定佇列共用群組 (QSG) 名稱。這可讓應用程式用戶端參與具有 GROUP 單元回復處置的交易。

如果應用程式在 QMNAME 參數欄位中使用 QSG 名稱，且它所連接的佇列管理程式上已停用 GROUPUR 內容，則 xa_open 呼叫會失敗。

如果應用程式連接至早於 V7.0.1 的佇列管理程式，則 xa_open 呼叫會成功，但交易具有 QMGR 回復單元。確保需要 GROUP 回復單元處置的應用程式僅連接至 V7.0.1 或更新版本中的佇列管理程式。

當用戶端應用程式稍後在交易管理程式用來發出 xa_open 呼叫的相同執行緒上呼叫 MQCONN 或 MQCONNX 時，應用程式會接收 xa_open 呼叫所啟動 MQI 通道的連線控點。未啟動第二個 MQI 通道。延伸交易式用戶端會檢查 MQCONN 或 MQCONNX 呼叫的 QMgrName 參數值是否為 MQI 通道伺服器端的佇列管理程式名稱。否則，MQCONN 或 MQCONNX 呼叫會失敗，原因碼為 MQRC_ANOTHER_Q_MGR_CONNECTED。如果 QMgrName 參數的值為空白或單一星號 (*)，或以星號開頭，則 MQCONN 或 MQCONNX 呼叫會失敗，原因碼為 MQRC_Q_MGR_NAME_ERROR。

如果在交易管理程式呼叫相同執行緒上的 xa_open 之前，用戶端應用程式已透過呼叫 MQCONN 或 MQCONNX 來啟動 MQI 通道，則交易管理程式會改用此 MQI 通道。未啟動第二個 MQI 通道。延伸交易式用戶端會檢查 xa_open 字串中 QMNAME 參數的值是否為伺服器佇列管理程式的名稱。如果不是，xa_open 呼叫會失敗。

如果用戶端應用程式先啟動 MQI 通道，則 MQCONN 或 MQCONNX 呼叫中 QMgrName 參數的值可以是空白或單一星號 (*)，也可以以星號開頭。不過，在這些情況下，您必須確定應用程式所連接的佇列管理程式，與交易管理程式稍後在相同執行緒上呼叫 xa_open 時，打算開啟作為資源管理程式的佇列管理程式相同。因此，如果 QMgrName 參數的值依名稱明確地識別佇列管理程式，則您可能會遇到較少的問題。

TPM 和 AXLIB 參數

延伸交易式用戶端使用 TPM 和 AXLIB 參數來尋找交易管理程式的 ax_reg 和 ax_unreg 函數。只有在佇列管理程式使用動態登錄時，才會使用這些功能。

如果在 xa_open 字串中提供 TPM 參數，但未提供 AXLIB 參數，則延伸交易式用戶端會根據 TPM 參數的值來假設 AXLIB 參數的值。如需 AXLIB 參數的假設值，請參閱第 91 頁的表 16。

TPM 的值	平台	AXLIB 的假設值
CICS	AIX	/usr/lpp/encina/lib/libEncServer.a(EncServer_shr.o)
CICS	HP-UX	/opt/encina/lib/libEncServer.sl
CICS	Solaris	/opt/encina/lib/libEncServer.so
CICS	Windows 系統	libEnc 伺服器
Tuxedo	AIX	/usr/lpp/tuxedo/lib/libtux.a(libtux.so.60)
Tuxedo	HP-UX	/opt/tuxedo/lib/libtux.sl
Tuxedo	Solaris	/opt/tuxedo/lib/libtux.so.60
Tuxedo	Windows 系統	libtux

如果在 xa_open 字串中提供 AXLIB 參數，則延伸交易式用戶端會使用其值，根據 TPM 參數的值來置換任何假設值。AXLIB 參數也可以用於 TPM 參數沒有指定值的交易管理程式。

xa_open 的其他錯誤處理

xa_open 呼叫在特定情況下失敗。

本節中的主題說明 xa_open 呼叫失敗的狀況。如果發生下列任何狀況，它也會失敗：

- xa_open 字串中有錯誤。
- 資訊不足，無法啟動 MQI 通道。
- 嘗試啟動 MQI 通道時發生問題 (例如，伺服器佇列管理程式不在執行中)。

延伸交易式處理程序失敗之後的回復

在失敗之後，交易管理程式必須能夠回復任何未完成的工作單元。如果要這麼做，交易管理程式必須能夠在失敗時，以資源管理程式身分開啟任何參與不完整工作單元的佇列管理程式。

如果您曾經需要變更任何配置資訊，因此，在進行變更之前，您必須確定已解決所有未完成的工作單元。或者，您必須確保配置變更不會影響交易管理程式開啟它需要開啟的佇列管理程式的能力。以下是這類配置變更的範例：

- 變更 xa_open 字串的內容
- 變更 MQSERVER 環境變數的值
- 變更用戶端通道定義表 (CCDT) 中的項目
- 刪除伺服器連線通道定義

XA 交換器結構

在每一個平台上，延伸交易式用戶端提供兩個 XA 交換器結構。

這些交換器結構如下：

MQRMIXASwitch

當作為資源管理程式的佇列管理程式未使用動態登錄時，交易管理程式會使用這個切換結構。

MQRMIXASwitchDynamic

當作為資源管理程式的佇列管理程式使用動態登錄時，交易管理程式會使用這個切換結構。

這些交換器結構位於 第 92 頁的表 17 所顯示的檔案庫中。

平台	包含 XA 交換器結構的程式庫
AIX HP-UX Linux Solaris	MQ_INSTALLATION_PATH/lib/libmqcxa
Windows 系統	MQ_INSTALLATION_PATH\bin\mqcxa.dll ¹
MQ_INSTALLATION_PATH 代表 WebSphere MQ 安裝所在的高階目錄。	

每一個交換器結構中的 WebSphere MQ 資源管理程式名稱是 MQSeries_XA_RMI，但許多佇列管理程式可以共用相同的交換器結構。

相關概念

第 92 頁的『動態登錄及延伸交易式處理』

使用動態登錄是最佳化的一種形式，因為它可以減少交易管理程式發出的 xa_ 函數呼叫數目。

動態登錄及延伸交易式處理

使用動態登錄是最佳化的一種形式，因為它可以減少交易管理程式發出的 xa_ 函數呼叫數目。

如果佇列管理程式不使用動態登錄，交易管理程式會在每個工作單元中包含佇列管理程式。交易管理程式透過呼叫 xa_start、xa_end 及 xa_prepare 來執行此動作，即使佇列管理程式沒有在工作單元內更新的資源。

如果佇列管理程式使用動態登錄，則交易管理程式會透過假設佇列管理程式未涉及工作單元，且不會呼叫 xa_start 來啟動。然後，只有在同步點控制內更新其資源時，佇列管理程式才會涉及工作單元。如果發生這種情況，延伸交易式用戶端會呼叫 ax_reg 來登錄佇列管理程式的參與。

將延伸交易式用戶端與 SSL 通道搭配使用

您無法使用 xa_open 字串來設定 SSL 通道。請遵循下列指示來使用用戶端通道定義表 (ccdt)。

關於這項作業

由於 xa_open xa_info 字串的大小有限，無法使用 xa_open string 方法來連接至佇列管理程式，以傳遞設定 SSL 通道所需的所有資訊。因此，您必須使用用戶端通道定義表，或者如果您的交易管理程式容許，則在發出 xa_open 呼叫之前，先使用 MQCONNX 建立通道。

若要使用用戶端通道定義表，請遵循下列步驟：

程序

1. 指定僅包含必要 qmname (佇列管理程式名稱) 參數的 xa_open 字串，例如：
XA_Open_String=qmname=MYQM
2. 使用佇列管理程式，以必要的 SSL 參數來定義 CLNTCONN (用戶端連線) 通道。在 CLNTCONN 定義的 QMNAME 屬性中併入佇列管理程式名稱。這將與 xa_open 字串中的 qmname 相符。
3. 在用戶端通道定義表 (CCDT) 中，或在 Windows 上，在作用中目錄中，讓用戶端系統可以使用 CLNTCONN 定義。
4. 如果您使用 CCDT，請使用環境變數 MQCHLLIB 及 MQCHLTAB 來識別包含 CLNTCONN 通道定義的 CCDT。在用戶端應用程式及交易管理程式使用的環境中設定這些變數。

結果

這為交易管理程式提供通道定義給適當的佇列管理程式，其中具有正確鑑別所需的 SSL 屬性，包括 SSLCIPH CipherSpec。

配置 CICS 的延伸交易式用戶端

您可以透過將 XAD 資源定義新增至 CICS 區域，來配置供 CICS 使用的延伸交易式用戶端。

使用線上 CICS 資源定義 (RDO) 指令 **cicsadd** 來新增 XAD 資源定義。XAD 資源定義指定下列資訊：

- xa_open 字串
- 交換器載入檔案的完整路徑名稱

在下列每一個平台上，提供一個交換器載入檔案供 CICS 使用：AIX、HP-UX、Solaris 及 Windows 系統。每一個交換器載入檔案都包含一個函數，可將指標傳回用於動態登錄的 XA 交換器結構 MQRMIASwitchDynamic。如需每一個交換器載入檔案的完整路徑名稱，請參閱第 93 頁的表 18。

平台	切換載入檔案
AIX HP-UX Linux Solaris	MQ_INSTALLATION_PATH/lib/amqcsc
Windows 系統	MQ_INSTALLATION_PATH\bin\mqcc4swi.dll ¹

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

以下是 Windows 系統的 XAD 資源定義範例：

```
cicsadd -c xad -r REGION1 WMQXA \  
ResourceDescription="WebSphere MQ queue manager MARS" \  
XAOpen="channel=MARS.SVR,trptype=tcp,connname=MARS(1415),qmname=MARS,tpm=cics" \  
SwitchLoadFile="C:\Program Files\IBM\WebSphere MQ\bin\mqcc4swi.dll"
```

如需將 XAD 資源定義新增至 CICS 區域的相關資訊，請參閱適用於您平台的 *CICS Administration Reference* 及 *CICS Administration Guide*。

請注意下列搭配使用 CICS 與延伸交易式用戶端的相關資訊：

- 您只能將 WebSphere MQ 的一個 XAD 資源定義新增至 CICS 區域。此表示一個區域只能有一個相關聯的佇列管理程式，且在該區域中執行的所有 CICS 應用程式只能連接至該佇列管理程式。如果您想要執行連接至不同佇列管理程式的 CICS 應用程式，則必須在不同區域中執行這些應用程式。
- 區域中的每一個應用程式伺服器都會在起始設定時呼叫 `xa_open`，並啟動與區域相關聯的佇列管理程式的 MQI 通道。這表示必須在應用程式伺服器啟動之前啟動佇列管理程式，否則 `xa_open` 呼叫會失敗。稍後由應用程式伺服器處理的所有 WebSphere MQ MQI 用戶端應用程式都會使用相同的 MQI 通道。
- 當 MQI 通道啟動，且通道的用戶端沒有安全結束程式時，從用戶端系統流向伺服器連線 MCA 的使用者 ID 是 `cics`。在特定情況下，當伺服器連線 MCA 後續嘗試代表用戶端應用程式存取佇列管理程式資源時，佇列管理程式會使用此使用者 ID 進行權限檢查。如果使用這個使用者 ID 來進行權限檢查，您必須確定它有權存取它需要存取的所有資源。

如需佇列管理程式何時使用此使用者 ID 進行權限檢查的相關資訊，請參閱 [安全](#)。

- 第 94 頁的表 19 中列出提供在 WebSphere MQ 用戶端系統上使用的 CICS 作業終止結束程式。您配置這些結束程式的方式與配置 WebSphere MQ 伺服器系統的對應結束程式的方式相同。因此，如需此資訊，請參閱 [啟用 CICS 使用者結束程式](#)。

表 19: CICS 作業終止結束程式		
平台	來源	檔案庫
AIX HP-UX Linux Solaris	amqzscgx.c	amqczscg
Windows 系統	amqzscgn.c	mqcc1415.dll

配置 Tuxedo 的延伸交易式用戶端

若要配置 XAD 資源定義供 Tuxedo 使用，請更新 UBBCONFIG 檔案及資源管理程式表格。

若要配置 XAD 資源定義以供 Tuxedo 使用，請執行下列動作：

- 在應用程式的 Tuxedo UBBCONFIG 檔案的 GROUPS 區段中，使用 OPENINFO 參數來指定 `xa_open` 字串。

如需如何執行此動作的範例，請參閱提供與 Tuxedo 範例程式搭配使用的範例 UBBCONFIG 檔案。在 AIX、HP-UX 和 Solaris 上，檔案名稱是 `ubbstxcx.cfg`，在 Windows 系統上，檔案名稱是 `ubbstxcn.cfg`。

- 在 Tuxedo 資源管理程式表格的佇列管理程式項目中：
 - `udataobj/RM` (AIX、HP-UX 和 Solaris)
 - `udataobj\rm` (Windows 系統)

指定 XA 交換器結構的名稱，以及包含該結構之程式庫的完整路徑名稱。如需如何針對每一個平台執行此動作的範例，請參閱 [TUXEDO 範例](#)。Tuxedo 支援資源管理程式的動態登錄，因此您可以使用 `MQRMIASwitch` 或 `MQRMIASwitchDynamic`。

Microsoft Transaction Server

在您可以使用 MTS 作為交易管理程式之前，不需要其他配置。不過，有幾點需要注意。

請注意下列與延伸交易式用戶端搭配使用 MTS 的相關資訊：

- 當 MTS 應用程式連接至伺服器佇列管理程式時，一律會啟動 MQI 通道。然後，MTS 會以其交易管理程式的角色，使用相同的 MQI 通道來與佇列管理程式進行通訊。
- 在失敗之後，MTS 必須能夠回復任何未完成的工作單元。若要這樣做，MTS 必須能夠與在失敗時參與不完整工作單元的任何佇列管理程式通訊。

當 MTS 應用程式連接至伺服器佇列管理程式並啟動 MQI 通道時，必要的話，延伸交易式用戶端會從 `MQCONN` 或 `MQCONNX` 呼叫的參數擷取足夠資訊，以便在失敗之後重新啟動通道。延伸交易式用戶端會將資訊傳遞至 MTS，而 MTS 會將資訊記錄在其日誌中。

如果 MTS 應用程式發出 MQCONN 呼叫，則此資訊只是佇列管理程式的名稱。如果 MTS 應用程式發出 MQCONNX 呼叫並提供通道定義結構 MQCD，則資訊還包括 MQI 通道名稱、伺服器佇列管理程式的網址，以及通道的通訊協定。

在回復狀況下，MTS 會將此資訊傳回延伸交易式用戶端，而延伸交易式用戶端會使用它來重新啟動 MQI 通道。

因此，如果您需要變更任何配置資訊，請確保在進行變更之前已解決所有未完成的工作單元。或者，請確定配置變更不會影響延伸交易式用戶端使用 MTS 所記錄的資訊來重新啟動 MQI 通道的能力。以下是這類配置變更的範例：

- 變更 MQSERVER 環境變數的值
- 變更用戶端通道定義表 (CCDT) 中的項目
- 刪除伺服器連線通道定義
- 將延伸交易式用戶端與 MTS 搭配使用時，請注意下列條件：
 - 在單一執行緒內，用戶端應用程式一次只能連接至一個佇列管理程式。
 - 用戶端應用程式的每一個執行緒可以連接至不同的佇列管理程式。
 - 用戶端應用程式無法使用共用連線控點。

定義 MQI 通道

若要建立新的通道，您必須使用相同的通道名稱及相容的通道類型來建立 **兩個** 通道定義，每個連線端都有一個通道定義。在此情況下，通道類型為 *server-connection* 及 *client-connection*。

使用者定義通道

當伺服器未自動定義通道時，有兩種方法可以建立通道定義，並在 WebSphere MQ MQI 用戶端機器上，提供通道存取權給 WebSphere MQ 應用程式。

這兩種方法詳細說明：

1. 在 WebSphere MQ 用戶端上建立一個通道定義，並在伺服器上建立另一個通道定義。

這適用於 WebSphere MQ MQI 用戶端及伺服器平台的任何組合。當您開始使用系統時，請使用它，或測試您的設定。

如需如何使用此方法的詳細資料，請參閱 [第 96 頁的『在不同平台上建立伺服器連線和用戶端連線定義』](#)。

2. 在伺服器機器上建立兩個通道定義。

當您同時設定多個通道及 WebSphere MQ MQI 用戶端機器時，請使用此方法。

如需如何使用此方法的詳細資料，請參閱 [第 99 頁的『在伺服器上建立伺服器連線及用戶端連線定義』](#)。

自動定義的通道

z/OS 以外的平台上的 WebSphere MQ 產品包含一項特性，可自動在伺服器上建立通道定義 (如果不存在的話)。

如果從用戶端收到入埠連接要求，且在該佇列管理程式上找不到適當的伺服器連線定義，WebSphere MQ 會自動建立定義，並將它新增至佇列管理程式。自動定義是根據預設伺服器連線通道 SYSTEM.AUTO.SVRCONN。您可以使用 ALTER QMGR 指令搭配 CHAD 參數 (或 PCF 指令「變更佇列管理程式」搭配 ChannelAutoDef 參數) 來更新佇列管理程式物件，以啟用伺服器連線定義的自動定義。

如需自動建立通道定義的相關資訊，請參閱 [接收端及伺服器連線通道的自動定義](#)。

相關概念

[第 96 頁的『自動定義的通道』](#)

z/OS 以外的平台上的 WebSphere MQ 產品包含一項特性，可自動在伺服器上建立通道定義 (如果不存在的話)。

[第 96 頁的『使用者定義通道』](#)

當伺服器未自動定義通道時，有兩種方法可以建立通道定義，並在 WebSphere MQ MQI 用戶端機器上，提供通道存取權給 WebSphere MQ 應用程式。

第 46 頁的『[通道控制功能](#)』

通道控制功能可讓您定義、監視及控制通道。

自動定義的通道

z/OS 以外的平台上的 WebSphere MQ 產品包含一項特性，可自動在伺服器上建立通道定義 (如果不存在的話)。

如果從用戶端收到入埠連接要求，且在該佇列管理程式上找不到適當的伺服器連線定義，WebSphere MQ 會自動建立定義，並將它新增至佇列管理程式。自動定義是根據預設伺服器連線通道 SYSTEM.AUTO.SVRCONN。您可以使用 ALTER QMGR 指令搭配 CHAD 參數 (或 PCF 指令「變更佇列管理程式」搭配 ChannelAutoDef 參數) 來更新佇列管理程式物件，以啟用伺服器連線定義的自動定義。

使用者定義通道

當伺服器未自動定義通道時，有兩種方法可以建立通道定義，並在 WebSphere MQ MQI 用戶端機器上，提供通道存取權給 WebSphere MQ 應用程式。

這兩種方法詳細說明：

1. 在 WebSphere MQ 用戶端上建立一個通道定義，並在伺服器上建立另一個通道定義。

這適用於 WebSphere MQ MQI 用戶端及伺服器平台的任何組合。當您開始使用系統時，請使用它，或測試您的設定。

如需如何使用此方法的詳細資料，請參閱第 96 頁的『[在不同平台上建立伺服器連線和用戶端連線定義](#)』。

2. 在伺服器機器上建立兩個通道定義。

當您同時設定多個通道及 WebSphere MQ MQI 用戶端機器時，請使用此方法。

如需如何使用此方法的詳細資料，請參閱第 99 頁的『[在伺服器上建立伺服器連線及用戶端連線定義](#)』。

在不同平台上建立伺服器連線和用戶端連線定義

您可以在套用它的電腦上建立每一個通道定義。在用戶端電腦上建立通道定義的方式有一些限制。

在所有平台上，您可以使用 WebSphere MQ Script (MQSC) 指令、可程式指令格式 (PCF) 指令或「IBM WebSphere MQ 探險家」，在伺服器機器上定義伺服器連線通道。

因為在已將 WebSphere MQ 安裝為 WebSphere MQ MQI 用戶端的機器上無法使用 MQSC 指令，您必須使用不同的方式在用戶端機器上定義用戶端連線通道。

相關概念

第 97 頁的『[在 IBM WebSphere MQ MQI 用戶端上建立用戶端連線通道](#)』

您可以使用 MQSERVER 或 MQCONN 呼叫 MQCNO 結構，在用戶端工作站上定義用戶端連線通道。

相關工作

第 96 頁的『[在伺服器上定義伺服器連線通道](#)』

必要的話，請啟動 MQSC，然後定義伺服器連線通道。

在伺服器上定義伺服器連線通道

必要的話，請啟動 MQSC，然後定義伺服器連線通道。

程序

1. 選擇性的: 如果您的伺服器平台不是 z/OS，請先建立並啟動佇列管理程式，然後啟動 MQSC 指令。
 - a) 建立稱為 QM1 的佇列管理程式，例如：

```
ctmqm QM1
```

b) 啟動佇列管理程式:

```
stmqm QM1
```

c) 啟動 MQSC 指令:

```
runmqsc QM1
```

2. 使用您選擇的名稱及通道類型 *server-connection* 來定義通道。

```
DEFINE CHANNEL(CHAN1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +  
DESCR('Server-connection to Client_1')
```

此通道定義與伺服器上執行的佇列管理程式相關聯。

3. 使用下列指令，以容許對佇列管理程式的入埠連接存取:

```
SET CHLAUTH(CHAN1) TYPE(ADDRESSMAP) ADDRESS('IP address') MCAUSER('userid')
```

- 其中 SET CHLAUTH 使用前一個步驟中定義的通道名稱。
- 其中 'IP address' 是用戶端的 IP 位址。
- 其中 'userid' 是您要提供給目標佇列之存取控制通道的 ID。此欄位區分大小寫。

您可以選擇使用許多不同的屬性來識別入埠連線。此範例使用 IP 位址。替代屬性包括用戶端使用者 ID 及 SSL 或 TLS 主體識別名稱。如需相關資訊，請參閱 [通道鑑別記錄](#)

在 IBM WebSphere MQ MQI 用戶端上建立用戶端連線通道

您可以使用 MQSERVER 或 MQCONN 呼叫 MQCNO 結構，在用戶端工作站上定義用戶端連線通道。

使用 MQSERVER

您可以使用 MQSERVER 環境變數來指定用戶端連線通道的簡式定義。簡單的意思是您只能使用此方法來指定通道的幾個屬性。

- 在 Windows 上指定簡式通道定義，如下所示:

```
SET MQSERVER=ChannelName/TransportType/ConnectionName
```

- 在 UNIX and Linux 系統上指定簡式通道定義，如下所示:

```
export MQSERVER=ChannelName/TransportType/ConnectionName
```

其中:

- ChannelName 必須與伺服器上定義的名稱相同。它不能包含正斜線。
- 視您的 IBM WebSphere MQ MQI 用戶端平台而定，TransportType 可以是下列其中一個值:
 - LU62
 - TCP
 - NETBIOS
 - SPX

註: 在 UNIX and Linux 系統上，TransportType 區分大小寫且必須大寫。如果無法辨識 TransportType，則 MQCONN 或 MQCONN 呼叫會傳回 2058

- ConnectionName 是定義給通訊協定 (TransportType) 的伺服器名稱。

例如，在 Windows 上:

```
SET MQSERVER=CHANNEL1/TCP/MCID66499
```

或者，在 UNIX and Linux 系統上：

```
export MQSERVER=CHANNEL1/TCP/'MCID66499'
```

註：若要變更 TCP/IP 埠號，請參閱 第 124 頁的『MQSERVER』。

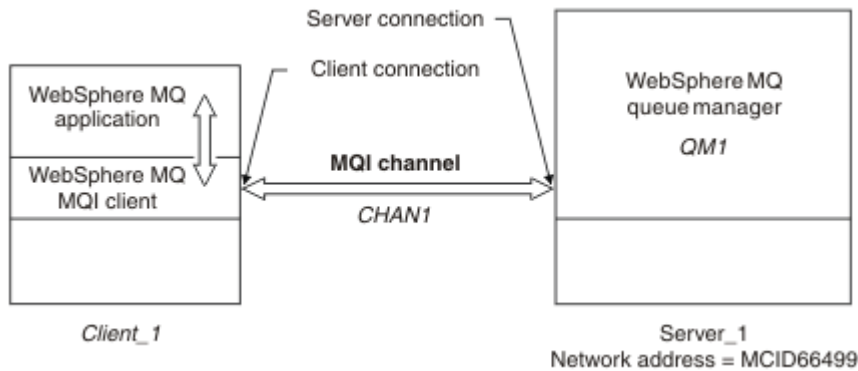


圖 17: 簡式通道定義

以下是一些簡單通道定義的其他範例：

- 在 Windows 上：

```
SET MQSERVER=CHANNEL1/TCP/9.20.4.56  
SET MQSERVER=CHANNEL1/NETBIOS/BOX643
```

- 在 UNIX and Linux 系統上：

```
export MQSERVER=CHANNEL1/TCP/'9.20.4.56'  
export MQSERVER=CHANNEL1/LU62/BOX99
```

其中 BOX99 是 LU 6.2 ConnectionName。

在 IBM WebSphere MQ MQI 用戶端上，除非在提供給 **MQCONN** 的 **MQCNO** 結構所參照的 **MQCD** 結構中置換通道，否則所有 **MQCONN** 或 **MQCONN** 要求都會嘗試使用您已定義的通道。

註：如需 **MQSERVER** 環境變數的相關資訊，請參閱 第 124 頁的『MQSERVER』。

在 **MQCONN** 呼叫中使用 **MQCNO** 結構

IBM WebSphere MQ MQI 用戶端應用程式可以在 **MQCONN** 呼叫中使用連接選項結構 **MQCNO**，來參照包含用戶端連線通道定義的通道定義結構 **MQCD**。

以此方式，用戶端應用程式可以在執行時期指定通道的 **ChannelName**、**TransportType** 及 **ConnectionName** 屬性，讓用戶端應用程式可以同時連接至多個伺服器佇列管理程式。

請注意，如果您使用 **MQSERVER** 環境變數來定義通道，則無法在執行時期指定 **ChannelName**、**TransportType** 及 **ConnectionName** 屬性。

用戶端應用程式也可以指定通道的屬性，例如 **MaxMsgLength** 和 **SecurityExit**。指定這類屬性可讓用戶端應用程式指定非預設值的屬性值，並在 MQI 通道的用戶端端呼叫通道結束程式。

如果通道使用 Secure Sockets Layer (SSL) 或傳輸層安全 (TLS)，則用戶端應用程式也可以在 **MQCD** 結構中提供與 SSL 或 TLS 相關的資訊。與 SSL 或 TLS 相關的其他資訊可以在 SSL 或 TLS 配置選項結構 **MQSCO** 中提供，**MQCNO** 結構也會在 **MQCONN** 呼叫中參照該結構。

如需 **MQCNO**、**MQCD** 及 **MQSCO** 結構的相關資訊，請參閱 [MQCNO](#)、[MQCD](#) 及 [MQSCO](#)。

註：**MQCONN** 的範例程式稱為 **amqscnxc**。另一個稱為 **amqssslc** 的範例程式會示範 **MQSCO** 結構的使用。

在伺服器上建立伺服器連線及用戶端連線定義

您可以在伺服器上建立這兩個定義，然後讓用戶端可以使用用戶端連線定義。

首先定義伺服器連線通道，然後定義用戶端連線通道。在所有平台上，您可以使用 WebSphere MQ Script (MQSC) 指令、可程式指令格式 (PCF) 指令或「IBM WebSphere MQ 探險家」，在伺服器機器上定義伺服器連線通道。

在伺服器上建立的用戶端連線通道定義可使用用戶端通道定義表 (CCDT) 供用戶端使用。

相關概念

第 99 頁的『用戶端通道定義表』

用戶端通道定義表 (CCDT) 決定用戶端應用程式用來連接佇列管理程式的通道定義及鑑別資訊。在 z/OS 以外的平台上，會自動建立 CCDT。然後，您必須讓它可供用戶端應用程式使用。

相關工作

第 100 頁的『在伺服器上定義伺服器連線通道』

建立佇列管理程式的伺服器連線通道定義。

第 101 頁的『在伺服器上定義用戶端連線通道』

在定義伺服器連線通道之後，您現在定義對應的用戶端連線通道。

第 102 頁的『存取用戶端連線通道定義』

透過複製或共用用戶端應用程式，讓用戶端通道定義表 (CCDT) 可供用戶端應用程式使用，然後在用戶端電腦上指定其位置及名稱。

用戶端通道定義表

用戶端通道定義表 (CCDT) 決定用戶端應用程式用來連接佇列管理程式的通道定義及鑑別資訊。在 z/OS 以外的平台上，會自動建立 CCDT。然後，您必須讓它可供用戶端應用程式使用。

用戶端通道定義表 (CCDT) 的用途是決定用戶端應用程式用來連接佇列管理程式的通道定義。通道定義也會指定適用於連線的鑑別資訊。

CCDT 是二進位檔。它是由佇列管理程式所產生。佇列管理程式不會讀取 CCDT 檔案。

在 z/OS 以外的平台上，會在建立佇列管理程式時建立 CCDT。當您使用 **DEFINE CHANNEL** 指令時，會將用戶端連線通道新增至表格，並在您發出 **ALTER CHANNEL** 指令時變更其定義。

您可以使用 CCDT 來提供鑑別資訊給用戶端，以檢查 SSL 憑證撤銷。定義包含鑑別資訊物件的名稱清單，並將佇列管理程式屬性 **SSLCRLNameList** 設為名稱清單的名稱。

用戶端應用程式使用 CCDT 的方式有許多種。CCDT 可以複製到用戶端電腦。您可以將 CCDT 複製到多個用戶端共用的位置。當 CCDT 仍位於伺服器上時，您可以讓用戶端以共用檔案的形式來存取 CCDT。

如果您使用 FTP 來複製檔案，請使用 **bin** 選項來設定二進位模式；不要使用預設 ASCII 模式。不論您選擇讓 CCDT 可供使用的方法，都必須確保位置安全，以防止對通道進行未獲授權的變更。

z/OS 以外的平台

當您建立佇列管理程式時，會建立稱為 AMQCLCHL.TAB 的預設 CCDT。

依預設，AMQCLCHL.TAB 位於伺服器上的下列目錄中：

-   在 UNIX and Linux 系統上：

```
/prefix/qmgrs/QUEUMANAGERNAME/@ipcc
```

在 UNIX and Linux 系統上，*QUEUMANAGERNAME* 所參照的目錄名稱會區分大小寫。如果佇列管理程式名稱中有特殊字元，則目錄名稱可能與佇列管理程式名稱不同。

-  在 Windows 上：

```
MQ_INSTALLATION_PATH\data\qmgrs\QUEUMANAGERNAME\@ipcc
```

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

不過，您可能已選擇對佇列管理程式資料使用不同的目錄。當您使用 `crtmqm` 指令時，可以指定參數 `-md DataPath`。如果您這麼做，`AMQCLCHL.TAB` 會位於您指定之 `DataPath` 的 `@ipcc` 目錄中。

您可以設定 `MQCHLLIB` 來變更 `CCDT` 的路徑。如果您設定 `MQCHLLIB`，請注意，如果您在相同伺服器上有許多個佇列管理程式，它們會共用相同的 `CCDT` 位置。

當建立佇列管理程式時，會建立 `CCDT`。`CCDT` 的每一個項目都代表與特定佇列管理程式的用戶端連線。當您使用 **DEFINE CHANNEL** 指令來定義用戶端連線通道時，會新增一個項目，當您使用 **ALTER CHANNEL** 指令來變用戶端連線通道時，會更新這個項目。

如何指定 CCDT 在用戶端上的位置

在用戶端系統上，您可以透過兩種方式來指定 `CCDT` 的位置：

- 使用環境變數 `MQCHLLIB` 來指定表格所在的目錄，並使用 `MQCHLTAB` 來指定表格的檔名。
- 使用用戶端配置檔。在 `CHANNELS` 段落中，使用屬性 `ChannelDefinitionDirectory` 來指定表格所在的目錄，並使用 `ChannelDefinitionFile` 來指定檔名。

如果在用戶端配置檔中以及使用環境變數來指定位置，則環境變數會優先處理。您可以使用此特性在用戶端配置檔中指定標準位置，並在必要時使用環境變數來置換它。

相關參考

[第 122 頁的『MQCHLLIB』](#)

`MQCHLLIB` 指定包含用戶端通道定義表 (`CCDT`) 之檔案的目錄路徑。該檔案建立在伺服器上，但可以複製到 `WebSphere MQ MQI` 用戶端工作站。

相關資訊

[使用已撤銷的憑證](#)

移轉及用戶端通道定義表 (CCDT)

一般而言，用戶端通道定義表的內部格式可能會從 `IBM WebSphere MQ` 的一個版次層次變更為下一個版次層次。因此，只有在伺服器佇列管理程式已準備用戶端通道定義表時，`IBM WebSphere MQ MQI` 用戶端才會使用該定義表，且該伺服器佇列管理程式的版本層次與用戶端相同，或更早版本層次。

7.1 `IBM WebSphere MQ MQI` 用戶端可以使用 6.0 佇列管理程式所準備的用戶端通道定義表。但是 6.0 版用戶端無法使用 7.1 版佇列管理程式所準備的用戶端通道定義表。

Active Directory 中的用戶端連線通道

在支援 `Active Directory` 的 `Windows` 系統上，`IBM WebSphere MQ` 會在 `Active Directory` 中發佈用戶端連線通道，以提供動態主從架構連結。

當定義用戶端連線通道物件時，它們會寫入用戶端通道定義檔中，稱為 `AMQCLCHL.TAB`。如果用戶端連線通道使用 `TCP/IP` 通訊協定，則 `IBM WebSphere MQ` 伺服器也會在 `Active Directory` 中發佈它們。當 `IBM WebSphere MQ` 用戶端決定如何連接至伺服器時，它會使用下列搜尋順序來尋找相關的用戶端連線通道物件定義：

1. `MQCONN` `MQCD` 資料結構
2. `MQSERVER` 環境變數
3. 用戶端通道定義檔
4. `Active Directory`

此順序表示任何現行應用程式都不受任何變更影響。您可以將 `Active Directory` 中的這些項目視為用戶端通道定義檔中的記錄，且 `IBM WebSphere MQ` 用戶端會以相同的方式處理它們。若要配置及管理在 `Active Directory` 中發佈用戶端連線通道定義的支援，請使用 `setmqscp` 指令，如 [setmqscp](#) 中所述。

在伺服器上定義伺服器連線通道

建立佇列管理程式的伺服器連線通道定義。

程序

1. 在伺服器機器上，使用您選擇的名稱及通道類型 *server-connection* 來定義通道。
例如：

```
DEFINE CHANNEL(CHAN2) CHLTYPE(SVRCONN) TRPTYPE(TCP) +  
DESCR('Server-connection to Client_2')
```

2. 使用下列指令，以容許對佇列管理程式的入埠連接存取：

```
SET CHLAUTH(CHAN2) TYPE(ADDRESSMAP) ADDRESS('IP address') MCAUSER('userid')
```

- 其中 SET CHLAUTH 使用前一個步驟中定義的通道名稱。
- 其中 'IP address' IP 位址是用戶端的 IP 位址。
- 其中 'userid' 是您要提供給目標佇列之存取控制通道的 ID。此欄位區分大小寫。

您可以選擇使用許多不同的屬性來識別入埠連線。此範例使用 IP 位址。替代屬性包括用戶端使用者 ID 及 SSL 或 TLS 主體識別名稱。如需相關資訊，請參閱 [通道鑑別記錄](#)

此通道定義與伺服器上執行的佇列管理程式相關聯。

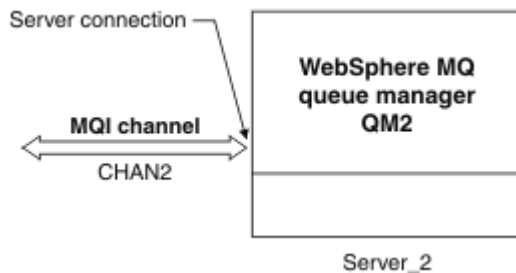


圖 18: 定義伺服器連線通道

在伺服器上定義用戶端連線通道

在定義伺服器連線通道之後，您現在定義對應的用戶端連線通道。

開始之前

定義伺服器連線通道。

程序

1. 定義與伺服器連線通道同名的通道，但通道類型為 *client-connection*。您必須陳述連線名稱 (CONNNAME)。對於 TCP/IP，連線名稱是伺服器機器的網址或主機名稱。也建議指定您要 IBM WebSphere MQ 應用程式 (在用戶端環境中執行) 連接的佇列管理程式名稱 (QMNAME)。透過改變佇列管理程式名稱，您可以定義一組通道來連接至不同的佇列管理程式。

```
DEFINE CHANNEL(CHAN2) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +  
CONNNAME(9.20.4.26) QMNAME(QM2) DESCR('Client-connection to Server_2')
```

2. 使用下列指令，以容許對佇列管理程式的入埠連接存取：

```
SET CHLAUTH(CHAN2) TYPE(ADDRESSMAP) ADDRESS('IP-address') MCAUSER('userid')
```

- 其中 SET CHLAUTH 使用前一個步驟中定義的通道名稱。
- 其中 'IP address' 是用戶端的 IP 位址。
- 其中 'userid' 是您要提供給目標佇列之存取控制通道的 ID。此欄位區分大小寫。

您可以選擇使用許多不同的屬性來識別入埠連線。此範例使用 IP 位址。替代屬性包括用戶端使用者 ID 及 SSL 或 TLS 主體識別名稱。如需相關資訊，請參閱 [通道鑑別記錄](#)

結果

在 z/OS 以外的平台上，此通道定義會儲存在稱為用戶端通道定義表 (CCDT) 的檔案中，該檔案與佇列管理程式相關聯。用戶端通道定義表可以包含多個用戶端連線通道定義。如需用戶端通道定義表的相關資訊，以及用戶端連線通道定義如何儲存在 z/OS 上的對應相關資訊，請參閱 [第 99 頁的『用戶端通道定義表』](#)。

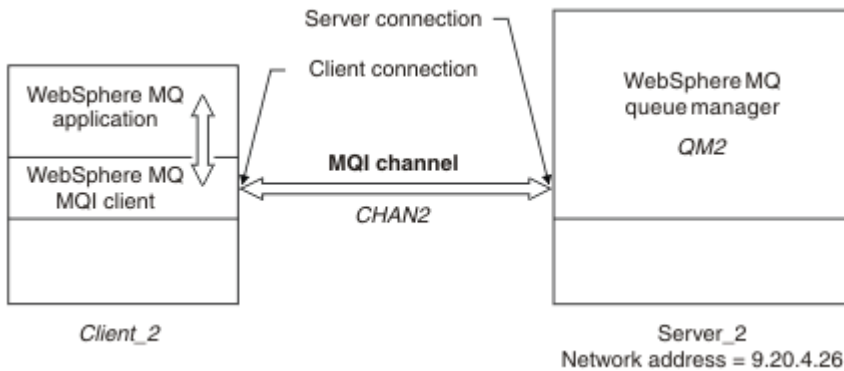


圖 19: 定義用戶端連線通道

存取用戶端連線通道定義

透過複製或共用用戶端應用程式，讓用戶端通道定義表 (CCDT) 可供用戶端應用程式使用，然後在用戶端電腦上指定其位置及名稱。

開始之前

您已定義您需要的用戶端連線通道。

在 z/OS 上，您已建立 CCDT。在其他平台上，會自動建立並更新 CCDT。

關於這項作業

若要讓用戶端應用程式使用用戶端通道定義表 (CCDT)，您必須讓 CCDT 可供它使用，並指定其位置和名稱

程序

1. 以下列三種方式之一，將 CCDT 提供給用戶端應用程式：

- 選擇性的：將 CCDT 複製到用戶端電腦。
- 選擇性的：將 CCDT 複製到多個用戶端共用的位置。
- 選擇性的：將 CCDT 保留在伺服器上，但讓用戶端可共用它。

不論您為 CCDT 選擇哪一個位置，該位置必須是安全的，以防止對通道進行未獲授權的變更。

2. 在用戶端上，以下列三種方式之一指定包含 CCDT 之檔案的位置和名稱：

- 選擇性的：使用用戶端配置檔的 CHANNELS 段落。如需相關資訊，請參閱 [第 113 頁的『用戶端配置檔的 CHANNELS 段落』](#)。
- 選擇性的：使用環境變數 MQCHLLIB 及 MQCHLTAB。

例如，您可以鍵入下列指令來設定環境變數：

- 在 HP Integrity NonStop Server 和 UNIX and Linux 系統上：

```
export MQCHLLIB=MQ_INSTALLATION_PATH/qmgrs/QUEUEMANAGERNAME/@ipcc
export MQCHLTAB=AMQCLCHL.TAB
```

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

- 選擇性的：僅在 Windows 上，使用 **setmqscp** 控制指令來發佈 Active Directory 中的用戶端連線通道定義

如果設定 MQSERVER 環境變數， WebSphere MQ 用戶端會優先於用戶端通道定義表中的任何定義，使用 MQSERVER 指定的用戶端連線通道定義。

MQI 通道的通道結束程式

在 UNIX、Linux 及 Windows 系統上， WebSphere MQ MQI 用戶端環境可以使用三種類型的通道結束程式。

它們是：

- 傳送結束程式
- 接收結束
- 安全結束程式

在通道的用戶端和伺服器端都可以使用這些結束程式。如果您是使用 MQSERVER 環境變數，則應用程式無法使用結束程式。通道結束程式在 [傳訊通道的通道結束程式](#) 中說明。

傳送和接收結束程式一起運作。有幾種可能的方式可以使用它們：

- 分割及重新組合訊息
- 壓縮及解壓縮訊息中的資料 (此功能作為 WebSphere MQ 的一部分提供，但您可能想要使用不同的壓縮技術)
- 加密及解密使用者資料 (此功能作為 WebSphere MQ 的一部分提供，但您可能想要使用不同的加密技術)
- 登載傳送及接收的每一個訊息

您可以使用安全結束程式來確保正確識別 WebSphere MQ 用戶端和伺服器，以及控制存取權。

如果通道實例的伺服器連線端上的傳送或接收結束程式需要在它們相關聯的連線上執行 MQI 呼叫，則它們會使用 MQCXP Hconn 欄位中提供的連線控點。您必須注意，用戶端連線傳送及接收結束程式無法發出 MQI 呼叫。

相關概念

第 104 頁的『[用戶端連線上的安全結束程式](#)』

您可以使用安全結束程式來驗證通道另一端的夥伴是否真實。將安全結束程式套用至用戶端連線時，會有特殊考量。

[使用者結束程式、API 結束程式及 WebSphere MQ 可安裝服務](#)

相關工作

[延伸佇列管理程式機能](#)

相關參考

第 103 頁的『[結束程式的路徑](#)』

通道結束程式位置的預設路徑定義在用戶端配置檔中。起始設定通道時，會載入通道結束程式。

第 105 頁的『[識別傳送或接收結束程式中的 API 呼叫](#)』

當您對用戶端使用 MQI 通道時，代理程式緩衝區的位元組 10 會識別在呼叫傳送或接收結束程式時所使用的 API 呼叫。這有助於識別哪些通道流程包含使用者資料，且可能需要處理 (例如加密或數位簽章)。

結束程式的路徑

通道結束程式位置的預設路徑定義在用戶端配置檔中。起始設定通道時，會載入通道結束程式。

在 UNIX、Linux 及 Windows 系統上，在安裝 WebSphere MQ MQI 用戶端期間，會將用戶端配置檔新增至您的系統。用戶端上通道結束程式位置的預設路徑定義在此檔案中，使用下列段落：

```
ClientExitPath:  
  ExitsDefaultPath=string  
  ExitsDefaultPath64=string
```

其中 *string* 是適合平台之格式的檔案位置

當起始設定通道時，在 MQCONN 或 MQCONNX 呼叫之後，會搜尋用戶端配置檔。會讀取 ClientExit 路徑段落，並載入通道定義中指定的任何通道結束程式。

用戶端連線上的安全結束程式

您可以使用安全結束程式來驗證通道另一端的夥伴是否真實。將安全結束程式套用至用戶端連線時，會有特殊考量。

第 105 頁的圖 20 說明在用戶端連線中使用安全結束程式，並使用 WebSphere MQ 物件權限管理程式來鑑別使用者。SecurityParmsPtr 或 SecurityParms 偏移設定在用戶端上的 MQCNO 結構中，且通道兩端都有安全結束程式。在正常安全訊息交換已結束且通道已備妥可執行之後，從 MQCXP SecurityParms 欄位存取的 MQCSP 結構會傳遞至用戶端上的安全結束程式。結束程式類型設為 MQXR_SEC_PARMS。安全結束程式可以選擇不對使用者 ID 及密碼執行任何動作，也可以變更其中一個或兩個。然後，從結束程式傳回的資料會傳送至通道的伺服器連線端。MQCSP 結構會在通道的伺服器連線端重建，並傳遞至從 MQCXP SecurityParms 欄位存取的伺服器連線安全結束程式。安全結束程式會接收並處理此資料。此處理通常是要反轉對用戶端結束程式中使用者 ID 及密碼欄位所做的任何變更，然後使用這些變更來授權佇列管理程式連線。在佇列管理程式系統上的 MQCNO 結構中使用 SecurityParmsPtr 來參照產生的 MQCSP 結構。

如果在 MQCNO 結構中設定 SecurityParmsPtr 或 SecurityParms 偏移，且通道的一端只有安全結束程式，則安全結束程式會接收並處理 MQCSP 結構。加密之類的動作不適用於單一使用者結束程式，因為沒有結束程式可執行補充動作。

如果未在 MQCNO 結構中設定 SecurityParmsPtr 及 SecurityParmsOffset，且通道任一端或兩端都有安全結束程式，則會呼叫安全結束程式。任一安全結束程式都可以傳回其自己的 MQCSP 結構，透過 SecurityParmsPtr；除非終止安全結束程式 (MQXR_TERM 的 ExitReason)，否則不會再次呼叫安全結束程式。結束程式寫出器可以釋放該階段用於 MQCSP 的記憶體。

當伺服器連線通道實例共用多個交談時，安全結束程式的呼叫型樣會限制在第二個及後續的交談上。

對於第一次交談，該型樣與通道實例未共用交談時的型樣相同。對於第二次及後續的交談，絕不會使用 MQXR_INIT、MQXR_INIT_SEC 或 MQXR_SEC_MSG 來呼叫安全結束程式。它與 MQXR_SEC_PARMS 一起呼叫。

在具有共用交談的通道實例中，只會針對前次執行的交談呼叫 MQXR_TERM。

每一個交談都有機會在結束程式的 MQXR_SEC_PARMS 呼叫中變更 MQCD；在通道的伺服器連線端，例如，在建立與佇列管理程式的連線之前，改變 MCAUserIdentifier 或 LongMCAUserIdPtr 值可能會很有用。

Server-connection exit	Client-connection exit
	Invoked with MQXR_INIT Responds with MQXCC_OK
Invoked with MQXR_INIT Responds with MQXCC_OK	
	Invoked with MQXR_INIT_SEC Responds with MQXCC_OK
Invoked with MQXR_INIT_SEC Responds with MQXCC_OK	
	Invoked with MQXR_SEC_PARMS Responds with MQXCC_OK
Invoked with MQXR_SEC_PARMS Responds with MQXCC_OK	
Data transfer begins	
Invoked with MQXR_TERM Responds with MQXCC_OK	Invoked with MQXR_TERM Responds with MQXCC_OK

圖 20: 用戶端連線-使用安全參數起始與用戶端連線協議的交換

註: 在 WebSphere MQ v7.1 之前所建構的安全結束程式應用程式可能需要更新。如需相關資訊, 請參閱 [通道安全結束程式](#)。

識別傳送或接收結束程式中的 API 呼叫

當您對用戶端使用 MQI 通道時, 代理程式緩衝區的位元組 10 會識別在呼叫傳送或接收結束程式時所使用的 API 呼叫。這有助於識別哪些通道流程包含使用者資料, 且可能需要處理 (例如加密或數位簽章)。

下表顯示在處理 API 呼叫時通道流程的位元組 10 中出現的資料。

註: 這些不是此位元組的唯一值。還有其他 **保留** 值。

API 呼叫	要求的位元組 10 值	回覆的位元組 10 值
MQCONN 第 106 頁的『1』 、 第 106 頁的『2』	X'81 '	X' 91 '
MQDISC 第 106 頁的『1』	X'82 '	X' 92 '
MQOPEN 第 106 頁的『3』	X'83 '	X' 93 '
MQCLOSE	X'84 '	X' 94 '
MQGET 第 106 頁的『4』	X'85 '	X' 95 '
MQPUT 第 106 頁的『4』	X'86 '	X' 96 '

表 20: 識別 API 呼叫 (繼續)

API 呼叫	要求的位元組 10 值	回覆的位元組 10 值
MQPUT1 要求 第 106 頁的『4』	X'87'	X'97'
MQSET 要求	X'88'	X'98'
MQINQ 要求	X'89'	X'99'
MQCMIT 要求	X'8A'	X'9A'
MQBACK 要求	X'8B'	X'9B'
MQSTAT 要求	X'8D'	X'9D'
MQSUB 要求	X'8E'	X'9E'
MQSUBRQ 要求	X'8F'	X'9F'
xa_start 要求	X'A1'	X'B1'
xa_end 要求	X'A2'	X'B2'
xa_open 要求	X'A3'	X'B3'
xa_close 要求	X'A4'	X'B4'
xa_prepare 要求	X'A5'	X'B5'
xa_commit 要求	X'A6'	X'B6'
xa_rollback 要求	X'A7'	X'B7'
xa_forget 要求	X'A8'	X'B8'
xa_recover 要求	X'A9'	X'B9'
xa_complete 要求	X'AA'	X'BA'

附註:

1. 用戶端與伺服器之間的連線由用戶端應用程式使用 MQCONN 起始。因此，尤其針對此指令，有數個其他網路流程。這也適用於 MQDISC，它會終止網路連線。
2. 基於主從架構連線的目的，MQCONN 的處理方式與 MQCONN 相同。
3. 如果開啟大型配送清單，則每個 MQOPEN 呼叫可能有多個網路流程，以便將所有必要資料傳遞至 SVRCONN MCA。
4. 大型訊息可能超出傳輸區段大小。如果發生這種情況，則單一 API 呼叫可能會產生許多網路流程。

將用戶端連接至佇列共用群組

在屬於佇列共用群組成員的伺服器上，您可以在用戶端與佇列管理程式之間建立 MQI 通道，以將用戶端連接至佇列共用群組。

佇列共用群組是由一組可存取同一組共用佇列的佇列管理程式所組成。

放入共用佇列的用戶端可以連接至佇列共用群組的任何成員。連接至佇列共用群組的好處可能增加前端及後端可用性，以及增加容量。您可以連接至特定佇列管理程式或一般介面。

直接連接至佇列共用群組中的佇列管理程式，可讓您將訊息放到共用目標佇列中，這會增加後端可用性。

連接至佇列共用群組的通用介面會開啟與群組中其中一個佇列管理程式的階段作業。這會增加前端系統可用性，因為用戶端佇列管理程式可以與群組中的任何佇列管理程式連接。當您不想要連接至佇列共用群組內的特定佇列管理程式時，可以使用通用介面來連接至群組。

一般介面可以是 WLM/DNS 群組名稱或 VTAM 一般資源名稱，或是佇列共用群組的另一個一般介面。

若要連接至佇列共用群組的一般介面，您需要建立可由群組中任何佇列管理程式存取的通道定義。若要這樣做，您必須在群組中的每一個佇列管理程式上具有相同的定義。

定義 SVRCONN 通道，如下所示：

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
MCAUSER(' ') QSGDISP(GROUP)
```

伺服器上的通道定義儲存在共用 DB2 儲存庫中。佇列共用群組中的每一個佇列管理程式都會建立定義的本端副本，確保在您發出 MQCONN 或 MQCONNX 呼叫時一律連接至正確的伺服器連線通道。

定義 CLNTCONN 通道，如下所示：

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNNAME(WLM/DNS_groupname) QMNAME(QSG1) +
DESCR('Client-connection to Queue Sharing Group QSG1') QSGDISP(GROUP)
```

因為佇列共用群組的一般介面儲存在用戶端連線通道的 CONNAME 欄位中，您現在可以連接至群組中的任何佇列管理程式，並將其放置在該群組所擁有的共用佇列中。

相關概念

第 107 頁的『[建立通道定義](#)』

若要連接至佇列共用群組的一般介面，您需要建立可由群組中任何佇列管理程式存取的通道定義。若要這樣做，您必須在群組中的每一個佇列管理程式上具有相同的定義。

建立通道定義

若要連接至佇列共用群組的一般介面，您需要建立可由群組中任何佇列管理程式存取的通道定義。若要這樣做，您必須在群組中的每一個佇列管理程式上具有相同的定義。

定義 SVRCONN 通道，如下所示：

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
MCAUSER(' ') QSGDISP(GROUP)
```

伺服器上的通道定義儲存在共用 DB2 儲存庫中。佇列共用群組中的每一個佇列管理程式都會建立定義的本端副本，確保在您發出 MQCONN 或 MQCONNX 呼叫時一律連接至正確的伺服器連線通道。

定義 CLNTCONN 通道，如下所示：

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNNAME(WLM/DNS_groupname) QMNAME(QSG1) +
DESCR('Client-connection to Queue Sharing Group QSG1') QSGDISP(GROUP)
```

因為佇列共用群組的一般介面儲存在用戶端連線通道的 CONNAME 欄位中，您現在可以連接至群組中的任何佇列管理程式，並將其放置在該群組所擁有的共用佇列中。

使用配置檔來配置用戶端

使用文字檔中的屬性來配置用戶端。這些屬性可以由環境變數置換，或以其他平台專用方式置換。

您可以使用文字檔（類似於 UNIX and Linux 平台上使用的佇列管理程式配置檔 `qm.ini`）來配置 IBM WebSphere MQ MQI client。檔案包含許多段落，每一個段落都包含許多行，格式為 **attribute-name=value**。

在本文件中，此檔案稱為 *WebSphere MQ MQI* 用戶端配置檔；其檔名通常是 `mqclient.ini`，但您可以選擇為它提供另一個名稱。此檔案中的配置資訊適用於所有平台，以及使用 MQI、IBM WebSphere MQ classes for Java、IBM WebSphere MQ classes for JMS、IBM WebSphere MQ classes for .NET 及 XMS 的用戶端。

雖然 IBM WebSphere MQ MQI client 配置檔中的屬性適用於大部分 IBM WebSphere MQ 用戶端，但有些屬性未由受管理 .NET 及 XMS .NET 用戶端讀取，或由使用 IBM WebSphere MQ classes for Java 或 IBM

WebSphere MQ classes for JMS 的用戶端讀取。如需相關資訊，請參閱第 109 頁的『[哪些 IBM WebSphere MQ 用戶端可以讀取每一個屬性](#)』。

配置特性適用於用戶端應用程式對任何佇列管理程式所建立的所有連線，而不是特定於佇列管理程式的個別連線。與個別佇列管理程式連線相關的屬性可以程式設計方式配置，例如使用 MQCD 結構或使用「用戶端通道定義表 (CCDT)」。

繼續支援 IBM WebSphere MQ 7.0 版之前的版本所支援的環境變數，如果這類環境變數符合用戶端配置檔中的對等值，則環境變數會置換用戶端配置檔值。

對於使用 IBM WebSphere MQ classes for JMS 的用戶端應用程式，您也可以使用下列方式來置換用戶端配置檔：

- 在 JMS 配置檔中設定內容
- 設定 Java 系統內容，也會置換 JMS 配置檔

對於 .NET 用戶端，您也可以使用 .NET 應用程式配置檔來置換用戶端配置檔及對等環境變數。

請注意，您無法使用用戶端配置檔來設定多個通道連線。

用戶端配置檔範例

```
##* Module Name: mqclient.ini                                ##*
##* Type       : WebSphere MQ MQI client configuration file  ##*
##* Function   : Define the configuration of a client        ##*
##*           :                                             ##*
##* Notes     :                                             ##*
##* 1) This file defines the configuration of a client      ##*
##*           :                                             ##*
##*           :                                             ##*
ClientExitPath:
  ExitsDefaultPath=/var/mqm/exits
  ExitsDefaultPath64=/var/mqm/exits64

TCP:
  Library1=DLLName1
  KeepAlive = Yes
  ClntSndBuffSize=32768
  ClntRcvBuffSize=32768
  Connect_Timeout=0

MessageBuffer:
  MaximumSize=-1
  Updatepercentage=-1
  PurgeTime=0

LU62:
  TPName
  Library1=DLLName1
  Library2=DLLName2

PreConnect:
  Module=amqldapi
  Function=myFunc
  Data=ldap://myLDAPServer.com:389/cn=wmq,ou=ibm,ou=com
  Sequence=1

CHANNELS:
  DefRecon=YES
  ServerConnectionParms=SALES.SVRCONN/TCP/hostname.x.com(1414)
```

相關參考

第 109 頁的『[用戶端配置檔的位置](#)』

IBM WebSphere MQ MQI 用戶端配置檔可以保留在許多位置。

第 113 頁的『[用戶端配置檔的 CHANNELS 段落](#)』

請使用 CHANNELS 段落來指定用戶端通道的相關資訊。

第 115 頁的『[ClientExit 用戶端配置檔的路徑段落](#)』

使用 ClientExit 路徑段落來指定用戶端上通道結束程式的預設位置。

第 116 頁的『用戶端配置檔的 LU62、NETBIOS 及 SPX 段落』
僅在 Windows 系統上，使用這些段落為指定的網路通訊協定指定配置參數。

第 116 頁的『用戶端配置檔的 MessageBuffer 段落』
使用 MessageBuffer 段落來指定訊息緩衝區的相關資訊。

第 118 頁的『用戶端配置檔的 SSL 段落』
請使用 SSL 段落來指定使用 SSL 或 TLS 的相關資訊。

第 120 頁的『用戶端配置檔的 TCP 段落』
使用 TCP 段落來指定 TCP 網路通訊協定配置參數。

第 121 頁的『使用 WebSphere MQ 環境變數』
本節說明可與 WebSphere MQ MQI 用戶端應用程式搭配使用的環境變數。

第 360 頁的『變更佇列管理程式配置資訊』
這裡說明的屬性會修改個別佇列管理程式的配置。它們會置換 WebSphere MQ 的任何設定。

用戶端配置檔的位置

IBM WebSphere MQ MQI 用戶端配置檔可以保留在許多位置。

用戶端應用程式會使用下列搜尋路徑來尋找 IBM WebSphere MQ MQI 用戶端配置檔：

1. 環境變數 MQCLNTCF 指定的位置。

此環境變數的格式是完整 URL。這表示檔名不一定是 mqclient.ini，有助於將檔案放置在網路連接的檔案系統上。

請注意下列項目：

- C、.NET 及 XMS 用戶端僅支援 file: 通訊協定；如果 URL 字串不是以 protocol: 開頭，則會假設 file: 通訊協定
- 若要容許不支援讀取環境變數的 Java 1.4.2 JRE，可以使用 MQCLNTCF Java 系統內容置換 MQCLNTCF 環境變數。

2. 應用程式的現行工作目錄中稱為 mqclient.ini 的檔案。

3. 在 Windows、UNIX and Linux 系統的 IBM WebSphere MQ 資料目錄中，稱為 mqclient.ini 的檔案。

請注意下列項目：

- IBM WebSphere MQ 資料目錄不存在於特定平台上 (例如，IBM i 和 z/OS)，或在用戶端已隨另一個產品提供的情況下。
- 在 UNIX and Linux 系統上，目錄為 /var/mqm
- 在 Windows 平台上，您可以在安裝期間配置環境變數 MQ_FILE_PATH，以指向資料目錄。它通常是 C:\Program Files\IBM\WebSphere MQ
- 若要容許不支援讀取環境變數的 Java 1.4.2 JRE，您可以手動將 MQ_FILE_PATH 環境變數置換為 MQ_FILE_PATH Java 系統內容。

4. 在適用於平台且可供使用者存取的標準目錄中，稱為 mqclient.ini 的檔案：

- 對於所有 Java 用戶端，這是 user.home Java 系統內容的值。
- 對於 UNIX and Linux 平台上的 C 用戶端，這是 HOME 環境變數的值。
- 對於 Windows 上的 C 用戶端，這是 HOMEDRIVE 和 HOMEPATH 環境變數的連結值。

註：對於 HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端，mqclient.ini 檔案必須位於 OSS 檔案系統中。監護應用程式必須將 mqclient.ini 檔案放置在 IBM WebSphere MQ 資料目錄中，或將 MQCLNTCF 環境變數設為 OSS 檔案系統中的位置。

哪些 IBM WebSphere MQ 用戶端可以讀取每一個屬性

IBM WebSphere MQ MQI client 配置檔中的大部分屬性都可以由 C 用戶端及未受管理的 .NET 用戶端使用。不過，有些屬性未由受管理 .NET 及 XMS .NET 用戶端讀取，或由使用 IBM WebSphere MQ classes for Java 或 IBM WebSphere MQ classes for JMS 的用戶端讀取。

表 21: 哪些屬性適用於每一種類型的用戶端

mqclient.ini 段落名稱及屬性	說明	C 及未受管理的 .NET	Java	JMS	受管理.NET	受管理 XMS .NET
通道段落						
<u>CCSID</u>	要使用的編碼字集號碼。	是	否	否	是	是
<u>ChannelDefinition 名錄</u>	包含用戶端通道定義表之檔案的目錄路徑。	是	否	否	是	是
<u>ChannelDefinition 檔案</u>	包含用戶端通道定義表的檔案名稱。	是	否	否	是	是
<u>ReconDelay</u>	針對可自動重新連接的用戶端程式，配置重新連接延遲的管理選項。	是	否	是	是	是
<u>DefRecon</u>	一種管理選項，可讓用戶端程式自動重新連接，或停用已寫入以自動重新連接之用戶端程式的自動重新連線。	是	否	是	是	是
<u>MQReconnectTimeout</u>	重新連接至用戶端的逾時 (以秒為單位)。	是	否	否	是	否
<u>ServerConnection 參數</u>	IBM WebSphere MQ 伺服器的位置，以及要使用的通訊方法。	是	否	否	是	是
<u>Put1DefaultAlwaysSync</u>	使用選項 MQPMO_RESPONSE_AS_Q_DEF 控制 MQPUT1 函數呼叫的行為。	是	是	是	是	是
ClientExit 路徑段落						
<u>ExitsDefaultPath</u>	指定用戶端 32 位元通道結束程式的位置。	是	否	否	是	是

表 21: 哪些屬性適用於每一種類型的用戶端 (繼續)

mqclient.ini 段落名稱及屬性	說明	C 及未受管理的 .NET	Java	JMS	受管理.NET	受管理 XMS .NET
ExitsDefaultPath64	指定用戶端 64 位元通道結束程式的位置。	是	否	否	是	是
JavaExitsClassPath	執行 Java 結束程式時要新增至類別路徑的值。	否	是	是	否	否
MessageBuffer 段落						
MaximumSize	先讀緩衝區的大小 (以 KB 為單位), 範圍為 1 到 999 999。	是	是	是	是	是
PurgeTime	間隔 (秒), 在此之後會清除先讀緩衝區中剩餘的訊息。	是	是	是	是	是
UpdatePercentage	在 1-100 範圍內的更新百分比值, 用於計算臨界值, 以判定用戶端應用程式何時向伺服器提出新要求。	是	是	是	是	是
SSL 段落						
CDPCheckExtensions	指定此佇列管理程式上的 SSL 或 TLS 通道是否嘗試檢查 CrlDistribution 點憑證延伸中指定的 CDP 伺服器。	是	否	否	否	否
CertificateLabel	通道定義的憑證標籤。	是	否	否	否	否
CertificateValidationPrinciples	決定使用的憑證驗證類型。	是	否	否	否	否

表 21: 哪些屬性適用於每一種類型的用戶端 (繼續)

mqclient.ini 段落名稱及屬性	說明	C 及未受管理的 .NET	Java	JMS	受管理.NET	受管理 XMS .NET
ClientRevocation 檢查	判定如果用戶端連接呼叫使用 SSL/TLS 通道, 如何配置憑證撤銷檢查。	是	否	否	否	否
EncryptionPolicySuiteB	決定通道是否使用 Suite-B 相容加密法, 以及要使用的強度層次。	是	否	否	否	否
OCSPAuthentication	定義當 OCSP 已啟用且 OCSP 撤銷檢查無法判斷憑證撤銷狀態時, IBM WebSphere MQ 的行為。	是	否	否	否	否
OCSPCheckExtensions	控制 IBM WebSphere MQ 是否作用於 AuthorityInfo Access 憑證延伸。	是	否	否	否	否
SSLCryptoHardware	設定在系統上配置 PKCS #11 加密硬體所需的參數字串。	是	否	否	否	否
SSLFipsRequired	指定在 IBM WebSphere MQ 中執行加密法時是否只使用 FIPS 認證的演算法。	是	否	否	否	否
SSLHTTPProxyName	字串是 GSKit for OCSP 檢查要使用之 HTTP Proxy 伺服器的主機名稱或網址。	是	否	否	否	否
SSLKeyRepository	保留使用者數位憑證的金鑰儲存庫位置 (以詞幹格式表示)。	是	否	否	否	否

表 21: 哪些屬性適用於每一種類型的用戶端 (繼續)

mqclient.ini 段落名稱及屬性	說明	C 及未受管理的 .NET	Java	JMS	受管理.NET	受管理 XMS .NET
SSLKeyReset 計數	在重新協議秘密金鑰之前, 在 SSL 或 TLS 通道上傳送及接收的未加密位元組數。	是	否	否	否	否
TCP 段落						
ClntRcvBufferSize	用戶端連線伺服器連線通道的用戶端所使用的 TCP/IP 接收緩衝區大小 (以位元組為單位)。	是	是	是	是	是
ClntSndBufferSize	用戶端連線伺服器連線通道的用戶端所使用的 TCP/IP 傳送緩衝區大小 (以位元組為單位)。	是	是	是	是	是
連接逾時	嘗試連接 Socket 逾時之前的秒數。	是	是	是	否	否
IPAddressVersion	指定要用於通道連線的 IP 通訊協定。	是	否	否	是	是
KeepAlive	開啟或關閉 KeepAlive 功能。	是	是	是	是	是
 Library1	僅在 Windows 上, 這是 TCP/IP Socket DLL 的名稱。	是	否	否	否	否

對於 HP Integrity NonStop Server, 您可以使用 [TMF](#) 及 [TmfGateway](#) 段落與 TMF/ 閘道進行通訊。

用戶端配置檔的 CHANNELS 段落

請使用 CHANNELS 段落來指定用戶端通道的相關資訊。

在 CHANNELS 段落中可以包含下列屬性:

CCSID=數字

要使用的編碼字集號碼。

CCSID 號碼相當於 MQCCSID 環境參數。

ChannelDefinitionDirectory=path

包含用戶端通道定義表之檔案的目錄路徑。

在 Windows 系統上，預設值是 IBM WebSphere MQ 安裝目錄，通常是 C:\Program Files\IBM\WebSphere MQ。在 UNIX and Linux 系統上，預設值為 /var/mqm。

ChannelDefinition 名錄路徑相當於 MQCHLLIB 環境參數。

ChannelDefinitionFile=檔名| AMQCLCHL.TAB

包含用戶端通道定義表的檔案名稱。

用戶端通道定義表相當於 MQCHLTAB 環境參數。

ReconDelay=(delay [, rand])(delay [, rand]) ...

ReconDelay 屬性提供管理選項，可針對可自動重新連接的用戶端程式配置重新連接延遲。以下是配置範例：

```
ReconDelay=(1000,200)(2000,200)(4000,1000)
```

所顯示的範例定義一秒的起始延遲，以及最多 200 毫秒的隨機間隔。下一個延遲是 2 秒加上隨機間隔（最多 200 毫秒）。所有後續延遲都是四秒，加上最多 1000 毫秒的隨機間隔。

DefRecon=NO|YES|QMGR|已停用

DefRecon 屬性提供管理選項，可讓用戶端程式自動重新連接，或停用已撰寫成自動重新連接之用戶端程式的自動重新連線。如果程式使用與重新連線不相容的選項（例如 MQPMO_LOGICAL_ORDER），則您可以選擇設定後者。

DefRecon 選項的解釋取決於用戶端程式中是否也設定 MQCNO_RECONNECT_* 值，以及設定的值。

如果用戶端程式使用 MQCONN 來連接，或使用 MQCONNX 來設定 MQCNO_RECONNECT_AS_DEF 選項，則 DefRecon 所設定的重新連接值會生效。如果程式中沒有設定重新連接值，或由 DefRecon 選項設定重新連接值，則不會自動重新連接用戶端程式。

適用於 Java 的 IBM WebSphere MQ 類別不支援自動用戶端重新連線。

NO

除非以 MQCONNX 置換，否則不會自動重新連接用戶端。

YES

除非被 MQCONNX 置換，否則用戶端會自動重新連接。

QMGR

除非以 MQCONNX 置換，否則用戶端會自動重新連接，但只會重新連接至相同的佇列管理程式。

QMGR 選項具有與 MQCNO_RECONNECT_Q_MGR 相同的效果。

DISABLED

即使用戶端程式使用 MQCONNX MQI 呼叫來要求，也會停用重新連線。

自動用戶端重新連線取決於兩個值：

- 應用程式中設定的重新連接選項
- mqclient.ini 檔案中的 DefRecon 值

mqclient.ini 中的 DefRecon 值	應用程式中設定的重新連線選項			
	MQCNO_RECONNECT	MQCNO_RECONNECT_Q_MGR	MQCNO_RECONNECT_AS_DEF	MQCNO_RECONNECT_DISABLED
NO	YES	QMGR	NO	NO
YES	YES	QMGR	YES	NO
QMGR	YES	QMGR	QMGR	NO

表 22: 自動重新連線取決於在應用程式及 *mqclient.ini* 檔案中設定的值 (繼續)

mqclient.ini 中的 DefRecon 值	應用程式中設定的重新連線選項			
DISABLED	NO	NO	NO	NO

MQReconnectTimeout

重新連接至用戶端的逾時 (以秒為單位)。預設值是 1800 秒 (30 分鐘)。

IBM WebSphere MQ classes for XMS .NET 用戶端可以使用內容 `XMSC.WMQ_CLIENT_RECONNECT_TIMEOUT`。此內容的預設值是 1800 秒 (30 分鐘)。

ServerConnectionParms

`ServerConnection` 參數相當於 `MQSERVER` 環境參數，並指定 IBM WebSphere MQ 伺服器的位置及要使用的通訊方法。`ServerConnectionParms` 屬性只定義簡式通道; 您無法使用它來定義 SSL 通道或具有通道結束程式的通道。它是 `ChannelName/TransportType/ConnectionName` 格式的字串，`ConnectionName` 必須是完整網路名稱。`ChannelName` 不能包含正斜線 ("/") 字元，因為此字元是用來區隔通道名稱、傳輸類型及連線名稱。

當使用 `ServerConnection` 參數來定義用戶端通道時，使用的訊息長度上限為 100 MB。因此，通道有效的訊息大小上限是伺服器上 `SVRCONN` 通道中指定的值。

請注意，只能建立單一用戶端通道連線。例如，如果您有兩個項目：

```
ServerConnectionParms=R1.SVRCONN/TCP/localhost(1963)
ServerConnectionParms=R2.SVRCONN/TCP/localhost(1863)
```

只使用第二個。

指定 `ConnectionName` 作為以逗點區隔的指定傳輸類型名稱清單。通常只需要一個名稱。您可以提供多個主機名稱，以配置具有相同內容的多個連線。會依照連線清單中指定的順序來嘗試連線，直到順利建立連線為止。如果沒有連線成功，用戶端會重新開始處理。連線清單是佇列管理程式群組的替代方案，用來配置可重新連接用戶端的連線。

Put1DefaultAlwaysSync=NO|YES

使用選項 `MQPMO_RESPONSE_AS_Q_DEF` 控制 `MQPUT1` 函數呼叫的行為。

NO

如果 `MQPUT1` 與 `MQPMO_SYNCPOINT` 一起設定，則其行為與 `MQPMO_ASYNC_RESPONSE` 相同。同樣地，如果使用 `MQPMO_NO_SYNCPOINT` 設定 `MQPUT1`，則其行為與 `MQPMO_SYNC_RESPONSE` 相同。這是預設值。

YES

不論是否設定 `MQPMO_SYNCPOINT` 或 `MQPMO_NO_SYNCPOINT`，`MQPUT1` 都會像設定 `MQPMO_SYNC_RESPONSE` 一樣運作。

ClientExit 用戶端配置檔的路徑段落

使用 `ClientExit` 路徑段落來指定用戶端上通道結束程式的預設位置。

`ClientExit` 路徑段落中可以包含下列屬性：

ExitsDefaultPath=字串

指定用戶端 32 位元通道結束程式的位置。

ExitsDefaultPath64=字串

指定用戶端 64 位元通道結束程式的位置。

JavaExitsClassPath=字串

執行 Java 結束程式時要新增至類別路徑的值。任何其他語言的結束程式都會忽略此情況。

在 JMS 配置檔中，為 `JavaExitsClassPath` 名稱提供標準 `com.ibm.mq.cfg`。在 WebSphere MQ V7.0 系統內容上也會使用字首和這個完整名稱。在 6.0 版中，這個屬性是使用系統內容 `com.ibm.mq.exitClasspath` 來指定，6.0 版 `Readme` 中有記載。已淘汰使用

com.ibm.mq.exitClasspath。如果同時存在 JavaExitsClassPath 和 exitClasspath，則允許使用 JavaExitsClassPath。如果只存在 exitClasspath 用法，在 Websphere MQ V7.0 中仍然允許使用它。

用戶端配置檔的 LU62、NETBIOS 及 SPX 段落

僅在 Windows 系統上，使用這些段落為指定的網路通訊協定指定配置參數。

LU62

使用 LU62 段落來指定 SNA LU 6.2 通訊協定配置參數。此段落中可以包含下列屬性：

Library1=DLLName|_WCPIC32

APPC DLL 的名稱。

Library2=DLLName|_WCPIC32

與 Library1 相同，如果程式碼儲存在兩個個別的程式庫中，則會使用。

TPName

要在遠端站台上啟動的 TP 名稱。

NETBIOS

使用 NETBIOS 段落來指定 NetBIOS 通訊協定配置參數。此段落中可以包含下列屬性：

AdapterNum=number|_0

LAN 配接卡的號碼。

Library1=DLLName|_NETAPI32

NetBIOS DLL 的名稱。

LocalName=name

在 LAN 上用來識別此電腦的名稱。

這相當於 MQNAME 環境參數。

NumCmds=number|_1

要配置的指令數。

NumSess=number|_1

要配置的階段作業數。

SPX

使用 SPX 段落來指定 SPX 通訊協定配置參數。此段落中可以包含下列屬性：

BoardNum=number|_0

LAN 配接卡號碼。

KeepAlive=YES|NO

開啟或關閉 KeepAlive 函數。

KeepAlive=YES 會導致 SPX 定期檢查連線的另一端是否仍然可用。如果不是，則會關閉通道。

Library1=DLLName|_WSOCK32.DLL

SPX DLL 的名稱。

Library2=DLLName|_WSOCK32.DLL

與 Library1 相同，用於將程式碼儲存在兩個個別的程式庫中。

Socket=number|_5E86

以十六進位表示法表示的 SPX Socket 號碼。

用戶端配置檔的 MessageBuffer 段落

使用 MessageBuffer 段落來指定訊息緩衝區的相關資訊。

MessageBuffer 段落中可以包含下列屬性：

MaximumSize=integer|_1

先讀緩衝區的大小 (以 KB 為單位)，範圍為 1 到 999 999。

存在下列特殊值:

-1

用戶端決定適當的值。

0

用戶端已停用先讀。

PurgeTime=integer|_600

間隔 (秒)，在此之後會清除先讀緩衝區中剩餘的訊息。

如果用戶端應用程式根據 `MsgId` 或 `CorrelId` 選取訊息，則先讀緩衝區可能包含傳送至具有先前所要求 `MsgId` 或 `CorrelId` 的用戶端的訊息。然後這些訊息會滯留在先讀緩衝區中，直到以適當的 `MsgId` 或 `CorrelId` 發出 `MQGET` 為止。您可以設定 `PurgeTime`，從先讀緩衝區清除訊息。在先讀緩衝區中保留超過清除間隔的任何訊息都會自動清除。這些訊息已從佇列管理程式上的佇列中移除，因此除非正在瀏覽它們，否則它們會遺失。

有效範圍在 1 到 999 999 秒的範圍內，或特殊值 0，表示不會進行清除。

UpdatePercentage=integer|_-1

在 1-100 範圍內的更新百分比值，用於計算臨界值，以判定用戶端應用程式何時向伺服器提出新要求。特殊值 -1 表示用戶端決定適當的值。

用戶端會定期傳送要求給伺服器，指出用戶端應用程式已耗用多少資料。當用戶端透過 `MQGET` 呼叫所擷取的位元組數 n 超出臨界值 T 時，會傳送要求。每次將新要求傳送至伺服器時， n 都會重設為零。

臨界值 T 的計算方式如下:

$$T = \text{Upper} - \text{Lower}$$

上限與 `MaximumSize` 屬性指定的先讀緩衝區大小 (以 KB 為單位) 相同。其預設值為 100 Kb。

「下限」低於「上限」，且由 `UpdatePercentage` 屬性指定。此屬性是 1 到 100 範圍內的數字，預設值為 20。「下限」的計算方式如下:

$$\text{Lower} = \text{Upper} \times \text{UpdatePercentage} / 100$$

範例 1:

`MaximumSize` 和 `UpdatePercentage` 屬性會採用其預設值 100 Kb 和 20 Kb。

用戶端會呼叫 `MQGET` 來擷取訊息，並反覆地這樣做。這會繼續執行，直到 `MQGET` 已耗用 n 個位元組為止。

使用計算

$$T = \text{Upper} - \text{Lower}$$

T 是 $(100-20) = 80$ Kb。

因此，當 `MQGET` 呼叫已從佇列中移除 80 Kb 時，用戶端會自動提出新的要求。

範例 2:

`MaximumSize` 屬性採用其預設值 100 Kb，並且為 `UpdatePercentage` 選擇值 40。

用戶端會呼叫 `MQGET` 來擷取訊息，並反覆地這樣做。這會繼續執行，直到 `MQGET` 已耗用 n 個位元組為止。

使用計算

$$T = \text{Upper} - \text{Lower}$$

T is $(100-40) = 60$ Kb

因此當 `MQGET` 呼叫已從佇列中移除 60 Kb 時，用戶端會自動提出新的要求。這比使用預設值的 `EXAMPLE 1` 還早。

因此，選擇較大的臨界值 T 會降低從用戶端傳送至伺服器的要求頻率。相反地，選擇較小的臨界值 T 會增加從用戶端傳送至伺服器的要求頻率。

不過，選擇大臨界值 *T* 可能表示先讀的效能增益會隨著先讀緩衝區變成空的機會增加而降低。當發生此情況時，MQGET 呼叫可能必須暫停，等待資料從伺服器送達。

用戶端配置檔的 SSL 段落

請使用 SSL 段落來指定使用 SSL 或 TLS 的相關資訊。

SSL 段落中可以包含下列屬性：

CDPCheckExtensions=YES|NO

CDPCheckExtensions 指定此佇列管理程式上的 SSL 或 TLS 通道是否嘗試檢查在 CrlDistribution 點憑證延伸中命名的 CDP 伺服器。

此屬性有下列可能值：

- YES: SSL 或 TLS 通道嘗試檢查 CDP 伺服器，以判定是否撤銷數位憑證。
- NO: SSL 或 TLS 通道不會嘗試檢查 CDP 伺服器。此值是預設值。

CertificateLabel = 字串

通道定義的憑證標籤。

此屬性可由 C 及未受管理的 .NET 用戶端讀取。

CertificateValPolicy=字串

決定使用的憑證驗證類型。

ANY

使用基礎 Secure Socket Library 所支援的任何憑證驗證原則。此設定為預設值。

RFC5280

僅使用符合 RFC 5280 標準的憑證驗證。

ClientRevocationChecks = REQUIRED | OPTIONAL | DISABLED

判定如果用戶端連接呼叫使用 SSL/TLS 通道，如何配置憑證撤銷檢查。另請參閱

OCSPAuthentication

此屬性可由 C 及未受管理的 .NET 用戶端讀取。

此屬性有下列可能值：

REQUIRED (預設值)

嘗試從 CCDT 載入憑證撤銷配置，並依照配置來執行撤銷檢查。如果無法開啟 CCDT 檔案，或無法驗證憑證 (例如，因為 OCSP 或 CRL 伺服器無法使用)，MQCONN 呼叫會失敗。如果 CCDT 未包含撤銷配置，但這不會導致通道失敗，則不會執行撤銷檢查。

Windows 在 Windows 系統上，您也可以使用 Active Directory 來進行 CRL 撤銷檢查。您無法使用 Active Directory 進行 OCSP 撤銷檢查。

OPTIONAL

至於 REQUIRED，如果無法載入憑證撤銷配置，通道不會失敗。

已停用

不會嘗試從 CCDT 載入憑證撤銷配置，也不會執行憑證撤銷檢查。

註: 如果您是使用 MQCONNX 而非 MQCONN 呼叫，則可以選擇透過 MQSCO 提供鑑別資訊記錄 (MQAIR)。因此，如果無法開啟 CCDT 檔案，則 MQCONNX 的預設行為不會失敗，而是假設您提供 MQAIR (即使您選擇不這麼做)。

EncryptionPolicySuiteB=字串

決定通道是否使用 Suite-B 相容加密法，以及要使用的強度層次。可能值包括：

無

不使用 Suite-B 相容加密法。此設定為預設值。

128_BIT,192_BIT

將安全強度同時設為 128 位元及 192 位元層次。

128_BIT

將安全強度設為 128 位元層次。

192_BIT

將安全強度設為 192 位元層次。

OCSPAuthentication=OPTIONAL|REQUIRED|WARN

定義當 OCSP 已啟用且 OCSP 撤銷檢查無法判斷憑證撤銷狀態時，WebSphere MQ 的行為。有三個可能的值：

OPTIONAL

任何具有 OCSP 檢查無法判斷之撤銷狀態的憑證都會被接受，且不會產生警告或錯誤訊息。SSL 或 TLS 連線會繼續進行，就像未進行撤銷檢查一樣。

必要

OCSP 檢查必須針對所檢查的每一個 SSL 或 TLS 憑證產生最終撤銷結果。任何具有撤銷狀態且無法驗證的 SSL 或 TLS 憑證都會遭到拒絕，並出現錯誤訊息。如果已啟用佇列管理程式 SSL 事件訊息，則會產生 MQRC_SSL_HANDSHAKE_ERROR 為 ReasonQualifier 的 MQRC_CHANNEL_SSL_ERROR 訊息。連線已關閉。

此值為預設值。

警告

如果 OCSP 撤銷檢查無法判斷任何 SSL 或 TLS 憑證的撤銷狀態，則會在佇列管理程式錯誤日誌中報告警告。如果已啟用佇列管理程式 SSL 事件訊息，則會產生具有 MQRC_SSL_UNKNOWN_REVOCATION 之 ReasonQualifier 的 MQRC_CHANNEL_SSL_WARNING 訊息。容許連線繼續。

OCSPCheckExtensions=YES|NO

控制 WebSphere MQ 是否作用於 AuthorityInfoAccess 憑證延伸。如果值設為 NO，WebSphere MQ 會忽略 AuthorityInfo 存取憑證延伸，且不會嘗試 OCSP 安全檢查。預設值為 YES。

SSLCryptoHardware=字串

設定在系統上配置 PKCS #11 加密硬體所需的參數字串。

以下列格式指定字串: GSK_PKCS11=驅動程式路徑和檔名; 記號標籤; 記號密碼; 對稱密碼設定;

例如: GSK_PKCS11=/usr/lib/pkcs11/
PKCS11_API.so;tokenlabel;passw0rd;SYMMETRIC_CIPHER_ON

驅動程式路徑是提供 PKCS #11 卡支援之共用程式庫的絕對路徑。驅動程式檔名是共用程式庫的名稱。PKCS #11 驅動程式路徑和檔名所需的值範例為 /usr/lib/pkcs11/PKCS11_API.so。若要透過 GSKit 存取對稱密碼作業，請指定對稱密碼設定參數。此參數的值為：

SYMMETRIC_CIPHER_OFF

不存取對稱密碼作業。此設定為預設值。

SYMMETRIC_CIPHER_ON

存取對稱密碼作業。

字串的長度上限為 256 個字元。預設值為空白。如果您指定的字串格式不正確，則會產生錯誤。

SSLFipsRequired=YES|NO

指定在 WebSphere MQ 中執行加密法時，是否只使用 FIPS 認證的演算法。如果已配置加密硬體，則使用的加密模組是硬體產品所提供的那些模組。視使用中的硬體產品而定，這些可能符合或不符合 FIPS 認證的特定層次。

SSLHTTPProxyName=字串

字串是 GSKit for OCSP 檢查要使用之 HTTP Proxy 伺服器的主機名稱或網址。此位址後面可以接著以括弧括住的選用埠號。如果您未指定埠號，則會使用預設 HTTP 埠 80。在 HP-UX PA-RISC 和 Sun Solaris SPARC 平台上，對於 AIX 上的 32 位元用戶端，網址只能是 IPv4 位址；在其他平台上，它可以是 IPv4 或 IPv6 位址。

例如，如果防火牆阻止存取 OCSP 回應端的 URL，則可能需要此屬性。

SSLKeyRepository=路徑名稱

保留使用者數位憑證的金鑰儲存庫位置 (以詞幹格式表示)。也就是說，它包括不含副檔名的完整路徑和檔名。

SSLKeyResetCount=integer|0

在重新協議秘密金鑰之前，在 SSL 或 TLS 通道上傳送及接收的未加密位元組數。

值必須在 0-999999999 範圍內。

預設值為 0，表示永不重新協議秘密金鑰。

如果您指定值 1-32768，則 SSL 或 TLS 通道會使用秘密金鑰重設計數 32768 (32Kb)。這是為了避免過多的金鑰重設，因為小型秘密金鑰重設值會發生這種情況。

用戶端配置檔的 TCP 段落

使用 TCP 段落來指定 TCP 網路通訊協定配置參數。

TCP 段落中可以包含下列屬性：

ClntRcvBuffSize=number| 32768

用戶端連線伺服器連線通道的用戶端所使用的 TCP/IP 接收緩衝區大小 (以位元組為單位)。值零表示作業系統將管理緩衝區大小，相對於 WebSphere MQ 所修正的緩衝區大小。

ClntSndBuffSize=number| 32768

用戶端連線伺服器連線通道的用戶端所使用的 TCP/IP 傳送緩衝區大小 (以位元組為單位)。值零表示作業系統將管理緩衝區大小，相對於 WebSphere MQ 所修正的緩衝區大小。

Connect_Timeout=數字

嘗試連接 Socket 逾時之前的秒數；除非通道已配置非零用戶端通道加權 (在此情況下，預設值為 5)，否則預設值為 0。

IPAddressVersion=MQIPADDR_IPV4|MQIPADDR_IPV6

指定要用於通道連線的 IP 通訊協定。

它具有可能的字串值 MQIPADDR_IPV4 或 MQIPADDR_IPV6。這些值的意義與 **ALTER QMGR IPADDRV** 中的 IPV4 及 IPV6 相同。

KeepAlive=YES|NO

開放或關閉 KeepAlive 函數。KeepAlive=YES 會使 TCP/IP 定期檢查連線的另一端是否仍然可用。如果不是，則會關閉通道。

Windows Library1=DLLName| WSOCK32

(僅限 Windows) TCP/IP Socket DLL 的名稱。

TMF 及 TMF/ 閘道段落

IBM WebSphere MQ 提供的 TMF/ 閘道在 Pathway 環境中執行。使用 TMF 及 TMF/ 閘道段落，為 HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端指定必要的配置參數，以與 TMF/ 閘道進行通訊。

如果您想要使用 TMF，則必須針對與您通訊的每一個佇列管理程式定義一個 TMF 段落及一個 TmfGateway 段落。所有值都衍生自您的配置。

TMF 段落

PathMon=name

定義 TMF/ 閘道的伺服器類別所定義 Pathmon 處理程序的名稱。

TMF Gateway 段落

此段落中可以包含下列屬性：

QManager=name

佇列管理程式的名稱。

Server=name

為該佇列管理程式配置之 TMF/ 閘道的伺服器類別名稱。

範例

以下是針對不同伺服器上兩個不同佇列管理程式的兩個 TmfGateway 段落所定義的 TMF 段落範例：

```
TMF:
  PathMon=$PSD1P
```

```
TmfGateway:
  QManager=MQ5B
  Server=MQ-MQ5B

TmfGateway:
  QManager=MQ5C
  Server=MQ-MQ5C
```

使用 WebSphere MQ 環境變數

本節說明可與 WebSphere MQ MQI 用戶端應用程式搭配使用的環境變數。

您可以使用下列方式來使用環境變數：

- 在系統設定檔中設定變數以進行永久變更
- 從指令行發出指令，只對此階段作業進行變更
- 若要為一或多個變數提供特定值 (視執行中的應用程式而定)，請將指令新增至應用程式所使用的指令 Script 檔

WebSphere MQ 會對您尚未設定的那些變數使用預設值。

除非另有說明，否則可在所有 WebSphere MQ MQI 用戶端平台上使用指令。

對於每一個環境變數，請使用與平台相關的指令來顯示現行設定或重設變數值。

例如：

設定或重設環境變數的值		
作用	指令	
	Windows	UNIX and Linux 系統
移除變數	set MQSERVER=	unset MQSERVER
顯示現行設定	SET MQSERVER	echo \$MQSERVER
顯示階段作業的所有環境變數	設定	設定

如需個別變數的相關資訊，請參閱下列子主題：

相關概念

第 107 頁的『[使用配置檔來配置用戶端](#)』

使用文字檔中的屬性來配置用戶端。這些屬性可以由環境變數置換，或以其他平台專用方式置換。

相關參考

[環境變數](#)

MQCCSID

MQCCSID 會指定要使用的編碼字集號碼，並置換已配置伺服器的 CCSID 值。

如需相關資訊，請參閱 [選擇用戶端或伺服器編碼字集 ID \(CCSID\)](#)。

若要設定此變數，請使用下列其中一個指令：

- 若為 Windows:

```
SET MQCCSID=number
```

- 若為 UNIX and Linux 系統:

```
export MQCCSID=number
```

MQCERTVPOL

MQCERTVPOL 指定使用的憑證驗證原則。

如需 WebSphere MQ 中憑證驗證原則的相關資訊，請參閱 [WebSphere MQ](#)。

此環境變數會置換用戶端 ini 檔案 SSL 段落中的 *CertificateValPolicy* 設定。變數可以設為下列兩個值之一：

ANY

使用基礎 Secure Socket Library 所支援的任何憑證驗證原則。

RFC5280

僅使用符合 RFC 5280 標準的憑證驗證。

若要設定此變數，請使用下列其中一個指令：

- 若為 Windows:

```
SET MQCERTVPOL=value
```

- 若為 UNIX and Linux 系統:

```
export MQCERTVPOL=value
```

MQCHLLIB

MQCHLLIB 指定包含用戶端通道定義表 (CCDT) 之檔案的目錄路徑。該檔案建立在伺服器上，但可以複製到 WebSphere MQ MQI 用戶端工作站。

如果未設定 MQCHLLIB，用戶端的路徑預設為：

- **Windows** 若為 Windows，請執行下列動作：*MQ_INSTALLATION_PATH*
- **UNIX Linux** 若為 UNIX and Linux 系統：*/var/mqm/*

對於 *crtmqm* 和 *strmqm* 指令，路徑預設為兩組路徑之一。如果設定 *datapath*，則路徑預設為第一個集合的其中一個。如果未設定 *datapath*，則路徑預設為第二個集合之一。

- **Windows** 若為 Windows：*datapath\@ipcc*
- **UNIX Linux** 若為 UNIX 和 Linux 系統：*datapath/@ipcc*

或：

- **Windows** 若為 Windows：*MQ_INSTALLATION_PATH\data\qmgrs\qmgrname\@ipcc*
- **UNIX Linux** 若為 UNIX and Linux 系統：*/prefix/qmgrs/qmgrname/@ipcc*

其中：

- *MQ_INSTALLATION_PATH* 代表 IBM WebSphere MQ 安裝所在的高階目錄。
- 如果存在，*datapath* 是在佇列管理程式段落中定義的 *DataPath* 值。
- *prefix* 是在佇列管理程式段落中定義的「字首」值。字首通常是 */var/mqm* 在 UNIX 和 Linux 平台上。
- *qmgrname* 是在佇列管理程式段落中定義的 *Directory* 屬性值。此值可能與實際佇列管理程式名稱不同。可能已變更值來取代特殊字元。
- 佇列管理程式段落定義在 UNIX 及 Linux 上的 *mqs.ini* 檔案中，以及 Windows 上的登錄中

附註：

1. 如果設定，MQCHLLIB 會置換用來尋找 CCDT 的路徑。
2. 環境變數 (例如 MQCHLLIB) 可以特定平台專用方式將範圍限定為處理程序或工作，或整個系統。
3. 如果您在伺服器上設定 MQCHLLIB 系統層面，它會針對伺服器上的所有佇列管理程式，設定 CCDT 檔案的相同路徑。如果您未設定 MQCHLLIB 環境變數，則每一個佇列管理程式的路徑都不同。佇列管理程式會在 *crtmqm* 或 *strmqm* 指令上讀取 MQCHLLIB 的值 (如果有設定的話)。

4. 如果您在一部伺服器上建立多個佇列管理程式，則區別很重要，原因如下。如果您設定 MQCHLLIB 系統層面，每一個佇列管理程式都會更新相同的 CCDT 檔案。此檔案包含來自伺服器上所有佇列管理程式的用戶端連線定義。例如，如果相同的定義存在於多個佇列管理程式上 (SYSTEM.DEF.CLNTCONN)，則檔案會包含最新定義。當您建立佇列管理程式時，如果設定 MQCHLLIB，則會在 CCDT 中更新 SYSTEM.DEF.CLNTCONN。更新會改寫不同佇列管理程式所建立的 SYSTEM.DEF.CLNTCONN。如果您已修改先前的定義，則您的修改會遺失。因此，您必須考慮尋找替代方案，將 MQCHLLIB 設為伺服器上的系統範圍環境變數。
5. 用戶端連線定義上的 MQSC 及 PCF NOREPLACE 選項不會檢查 CCDT 檔案的內容。不論 NOREPLACE 選項為何，都會取代先前建立但此佇列管理程式未建立之相同名稱的用戶端連線通道定義。如果定義先前的由相同的佇列管理程式所建立，則不會取代該定義。
6. 指令 `rcrimqobj -t clchltab` 會刪除並重建 CCDT 檔案。僅使用在執行指令的佇列管理程式上建立的用戶端連線定義來重建檔案。
7. 其他更新 CCDT 的指令只會修改具有相同通道名稱的用戶端連線通道。不會變更檔案中的其他用戶端連線通道。
8. MQCHLLIB 的路徑不需要引號。

範例

若要設定此變數，請使用下列其中一個指令：

-  若為 Windows:

```
SET MQCHLLIB=pathname
```

例如：

```
SET MQCHLLIB=C:\wmqtest
```

-   若為 UNIX and Linux 系統:

```
export MQCHLLIB=pathname
```

MQCHLTAB

MQCHLTAB 指定包含用戶端通道定義表 (ccdt) 的檔案名稱。預設檔名為 AMQCLCHL.TAB。

如需用戶端通道定義表在伺服器上的位置相關資訊，請參閱 [第 99 頁的『用戶端通道定義表』](#)。

若要設定此變數，請使用下列其中一個指令：

- 在 Windows 上:

```
SET MQCHLTAB=filename
```

- 在 UNIX and Linux 系統上:

```
export MQCHLTAB=filename
```

例如：

```
SET MQCHLTAB=ccdf1.tab
```

如同用戶端一樣，伺服器上的 MQCHLTAB 環境變數會指定用戶端通道定義表的名稱。

MQIPADDRV

MQIPADDRV 指定通道連線使用的 IP 通訊協定。它具有可能的字串值 "MQIPADDR_IPV4" 或 "MQIPADDR_IPV6"。這些值具有與 ALTER QMGR IPADDRV 中 IPV4 及 IPV6 相同的意義。如果未設定，則會採用 "MQIPADDR_IPV4"。

若要設定此變數，請使用下列其中一個指令：

- 若為 Windows:

```
SET MQIPADDRV=MQIPADDR_IPV4|MQIPADDR_IPV6
```

- 若為 UNIX and Linux 系統:

```
export MQIPADDRV=MQIPADDR_IPV4|MQIPADDR_IPV6
```

MQNAME

MQNAME 指定 WebSphere MQ 處理程序可以使用的本端 NetBIOS 名稱。

如需用戶端及伺服器上的完整說明及優先順序規則，請參閱第 74 頁的『在 Windows 上定義 NetBIOS 連線』。

若要設定此變數，請使用下列指令：

```
SET MQNAME=Your_env_Name
```

例如：

```
SET MQNAME=CLIENT1
```

如果您在 WebSphere MQ MQI 用戶端上同步執行多個 WebSphere MQ 應用程式，部分平台上的 NetBIOS 需要每一個應用程式不同的名稱 (由 MQNAME 設定)。

MQSERVER

MQSERVER 環境變數可用來定義最小通道。MQSERVER 指定 WebSphere MQ 伺服器的位置，以及要使用的通訊方法。

您無法使用 MQSERVER 來定義 SSL 通道或具有通道結束程式的通道。如需如何定義 SSL 通道的詳細資料，請參閱 [使用 SSL 保護通道](#)。

ConnectionName 必須是完整網路名稱。*ChannelName* 不能包含正斜線 (/) 字元，因為這個字元用來區隔通道名稱、傳輸類型和連線名稱。當使用 MQSERVER 環境變數來定義用戶端通道時，會使用 100 MB 的訊息長度上限 (MAXMSGL)。因此，通道有效的訊息大小上限是伺服器上 SVRCONN 通道中指定的值。

若要設定此變數，請使用下列其中一個指令：

- 若為 Windows:

```
SET MQSERVER=ChannelName/TransportType/ConnectionName
```

- 若為 UNIX and Linux 系統:

```
export MQSERVER='ChannelName/TransportType/ConnectionName'
```

TransportType 可以是下列其中一個值，視您的 IBM WebSphere MQ 用戶端平台而定：

- LU62
- TCP
- NETBIOS
- SPX

ConnectionName 可以是以逗點區隔的連線名稱清單。清單中連線名稱的使用方式與用戶端連線表格中多個連線的使用方式類似。*ConnectionName* 清單可用來作為佇列管理程式群組的替代方案，以指定多個連線供用戶端嘗試。如果您要配置多重實例佇列管理程式，您可以使用 *ConnectionName* 清單來指定不同的佇列管理程式實例。

TCP/IP 預設埠

依預設，對於 TCP/IP，WebSphere MQ 會假設通道將連接至埠 1414。

您可以透過下列方式變更此項目：

- 在方括弧中新增埠號作為 `ConnectionName` 的最後一部分：
 - 若為 Windows：

```
SET MQSERVER=ChannelName/TransportType/ConnectionName(PortNumber)
```

- 若為 UNIX and Linux 系統：

```
export MQSERVER='ChannelName/TransportType/ConnectionName(PortNumber)'
```

- 透過將埠號新增至通訊協定名稱來變更 `mqclient.ini` 檔案，例如：

```
TCP:  
port=2001
```

- 將 WebSphere MQ 新增至 `services` 檔，如第 80 頁的『使用 TCP/IP 接聽器』中所述。

SPX 預設 Socket

依預設，對於 SPX，WebSphere MQ 會假設通道將連接至 Socket 5E86。

您可以透過下列方式變更此項目：

- 新增方括弧中的 Socket 號碼作為 `ConnectionName` 的最後部分：

```
SET MQSERVER=ChannelName/TransportType/ConnectionName(SocketNumber)
```

對於 SPX 連線，請以 `network.node(socket)` 格式指定 `ConnectionName` 和 `Socket`。如果 WebSphere MQ 用戶端和伺服器位於相同網路上，則不需要指定網路。如果您使用預設 `Socket`，則不需要指定 `Socket`。

- 透過將埠號新增至通訊協定名稱來變更 `qm.ini` 檔案，例如：

```
SPX:  
socket=5E87
```

使用 MQSERVER

如果您使用 `MQSERVER` 環境變數來定義 WebSphere MQ MQI 用戶端機器與伺服器機器之間的通道，這是應用程式唯一可用的通道，不會參照用戶端通道定義表 (CCDT)。

在此狀況下，您在伺服器機器上執行的接聽器程式會決定應用程式將連接的佇列管理程式。它將與接聽器程式所連接的佇列管理程式相同。

如果 `MQCONN` 或 `MQCONNX` 要求指定的佇列管理程式不是接聽器所連接的佇列管理程式，或者如果無法辨識 `MQSERVER` 參數 `TransportType`，則 `MQCONN` 或 `MQCONNX` 要求會失敗，且回覆碼為 `MQRC_Q_MGR_NAME_ERROR`。

在 UNIX and Linux 系統上，您可以定義 `MQSERVER`，如下列其中一個範例所示：

```
export MQSERVER=CHANNEL1/TCP/'9.20.4.56(2002)'  
export MQSERVER=CHANNEL1/LU62/BOX99
```

然後所有 `MQCONN` 或 `MQCONNX` 要求都會嘗試使用您已定義的通道，除非已從提供給 `MQCONNX` 的 `MQCNO` 結構參照 `MQCD` 結構，在此情況下，`MQCD` 結構指定的通道優先於 `MQSERVER` 環境變數指定的任何通道。

`MQSERVER` 環境變數優先於 `MQCHLLIB` 及 `MQCHLTAB` 所指向的任何用戶端通道定義。

正在取消 MQSERVER

若要取消 MQSERVER 並回到 MQCHLLIB 及 MQCHLTAB 所指向的用戶端通道定義表，請輸入下列指令：

- 在 Windows 上：

```
SET MQSERVER=
```

- 在 UNIX and Linux 系統上：

```
unset MQSERVER
```

MQSSLCRYP

MQSSLCRYP 會保留參數字串，可讓您配置系統上呈現的加密硬體。允許的值與 ALTER QMGR 指令的 SSLCRYP 參數相同。

若要設定此變數，請使用下列其中一個指令：

- 在 Windows 系統上：

```
SET MQSSLCRYP=string
```

- 在 UNIX and Linux 系統上：

```
export MQSSLCRYP=string
```

相關參考

[ALTER QMGR 指令 SSLCRYP 參數](#)

MQSSLFIPS

MQSSLFIPS 指定在 WebSphere MQ 中執行加密法時，是否只使用 FIPS 認證的演算法。這些值與 ALTER QMGR 指令的 SSLFIPS 參數相同。

使用加密硬體會影響 FIPS 認證演算法的使用，請參閱 [指定在 MQI 用戶端上執行時期只使用 FIPS 認證的 CipherSpecs](#)。

若要設定此變數，請使用下列其中一個指令：

- 在 Windows 系統上：

```
SET MQSSLFIPS=YES|NO
```

- 在 UNIX and Linux 系統上：

```
export MQSSLFIPS=YES|NO
```

預設值為 NO。

MQSSLKEYR

MQSSLKEYR 以 stem 格式指定金鑰儲存庫的位置，該金鑰儲存庫保留屬於使用者的數位憑證。詞幹格式表示它包括不含副檔名的完整路徑和檔名。如需完整資料，請參閱 ALTER QMGR 指令的 SSLKEYR 參數。

若要設定此變數，請使用下列其中一個指令：

- 在 Windows 系統上：

```
SET MQSSLKEYR=pathname
```

- 在 UNIX and Linux 系統上:

```
export MQSSLKEYR=pathname
```

沒有預設值。

MQSSLPROXY

MQSSLPROXY 指定 GSKit 用於 OCSP 檢查之 HTTP Proxy 伺服器的主機名稱及埠號。

若要設定此變數，請使用下列其中一個指令：

- 在 Windows 系統上:

```
SET MQSSLPROXY=string
```

- 在 UNIX and Linux 系統上:

```
export MQSSLPROXY="string"
```

字串是由 GSKit 用於 OCSP 檢查的 HTTP Proxy 伺服器的主機名稱或網址。此位址後面可以接著以括弧括住的選用埠號。如果您未指定埠號，則會使用預設 HTTP 埠 80。

例如，在 UNIX and Linux 系統上，您可以使用下列其中一個指令：

- ```
export MQSSLPROXY="proxy.example.com(80)"
```

- ```
export MQSSLPROXY="127.0.0.1"
```

MQSSLRESET

MQSSLRESET 代表在重新協議秘密金鑰之前，在 SSL 或 TLS 通道上傳送及接收的未加密位元組數。

如需秘密金鑰重新協議的相關資訊，請參閱 [重設 SSL 及 TLS 秘密金鑰](#)。

它可以設為 0 到 999 999 999 範圍內的整數。預設值為 0，表示永不重新協議秘密金鑰。如果您指定 SSL 或 TLS 秘密金鑰重設計數，範圍從 1 位元組到 32 KB，則 SSL 或 TLS 通道會使用秘密金鑰重設計數 32 KB。此秘密重設計數是為了避免針對小型 SSL 或 TLS 秘密金鑰重設值進行過多的金鑰重設。

若要設定此變數，請使用下列其中一個指令：

- 在 Windows 系統上:

```
SET MQSSLRESET=integer
```

- 在 UNIX and Linux 系統上:

```
export MQSSLRESET=integer
```

控制排入佇列的發佈/訂閱

您可以啟動、停止及顯示佇列發佈/訂閱的狀態。您也可以分配管理系統階層中新增和移除串流，以及新增和刪除佇列管理程式。

如需控制佇列發佈/訂閱的相關資訊，請參閱下列子主題：

設定排入佇列的發佈/訂閱訊息屬性

您可以使用佇列管理程式屬性來控制部分發佈/訂閱訊息屬性的行為。您在 `qm.ini` 檔的分配管理系統段落中控制的其他屬性。

關於這項作業

您可以設定下列發佈/訂閱屬性：如需詳細資料，請參閱 [佇列管理程式參數](#)

表 23: 發佈/訂閱配置參數	
說明	MQSC 參數名稱
指令訊息重試次數	PSRTYCNT
捨棄無法遞送的指令輸入訊息	PSNPMSG
無法遞送的指令回應訊息之後的行為	PSNPRES
在同步點下處理指令訊息	PSSYNCPT

分配管理系統段落用來管理下列配置設定:

- `PersistentPublishRetry=yes | force`

如果您指定 **是**，則如果透過排入佇列的發佈/訂閱介面發佈持續訊息失敗，且未要求負面回覆，則會重試發佈作業。

如果您要求負面回應訊息，則會傳送負面回應，且不會再重試。

如果您指定 **強制**，則當透過排入佇列的發佈/訂閱介面發佈持續訊息失敗時，會重試發佈作業，直到順利處理為止。未傳送負面回應。

- `NonPersistentPublishRetry= yes | force`

如果您指定 **是**，則如果透過排入佇列的發佈/訂閱介面發佈非持續訊息失敗，且未要求負面回覆，則會重試發佈作業。

如果您要求負面回應訊息，則會傳送負面回應，且不會再重試。

如果您指定 **強制**，則如果透過排入佇列的發佈/訂閱介面發佈非持續訊息失敗，則會重試發佈作業，直到順利處理為止。未傳送負面回應。

註: 如果您要對非持續訊息啟用此功能，則除了設定 `NonPersistentPublishRetry` 值之外，還必須確保佇列管理程式屬性 `PSSYNCPT` 設為 **是**。

這樣做也可能會影響處理非持續性發佈的效能，因為現在來自 `STREAM` 佇列的 `MQGET` 在同步點下發生。

- `PublishBatch 大小 =數字`

分配管理系統通常會在同步點內處理發佈訊息。個別確定每一個發佈可能沒有效率，在某些情況下，分配管理系統可以在單一工作單元中處理多個發佈訊息。此參數指定可在單一工作單元中處理的發佈訊息數上限

`PublishBatchSize` 的預設值為 5。

- `PublishBatchInterval =number`

分配管理系統通常會在同步點內處理發佈訊息。個別確定每一個發佈可能沒有效率，在某些情況下，分配管理系統可以在單一工作單元中處理多個發佈訊息。此參數指定批次中第一個訊息與相同批次中所包含的任何後續發佈資訊之間的時間上限 (毫秒)。

批次間隔 0 表示只要訊息立即可用，最多可以處理 `PublishBatch 大小` 訊息。

`PublishBatch 間隔` 的預設值為零。

程序

使用「WebSphere MQ 探險家」、程式化指令或 `runmqsc` 指令，來變更控制發佈/訂閱行為的佇列管理程式屬性。

範例

```
ALTER QMGR PSNPRES (SAFE)
```

正在啟動排入佇列的發佈/訂閱

開始之前

請閱讀 [PSMODE](#) 的說明，以瞭解三種發佈/訂閱模式：

- COMPAT
- DISABLED
- ENABLED

註：如果您已從 Version 6.0 移轉，則在使用已升級的佇列管理程式時，必須使用 **strmqbrk** 來移轉 Version 6.0 發佈/訂閱分配管理系統狀態。這不適用於 z/OS。

關於這項作業

設定 QMGR PSMODE 屬性，以啟動排入佇列的發佈/訂閱介面 (也稱為分配管理系統) 及/或發佈/訂閱引擎 (也稱為第 7 版發佈/訂閱)。若要啟動排入佇列的發佈/訂閱，您需要將 PSMODE 設為 ENABLED。預設值為 ENABLED。

程序

如果尚未啟用排入佇列的發佈/訂閱介面，請使用「WebSphere MQ 探險家」或 **runmqsc** 指令來啟用該介面。

範例

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

下一步

WebSphere MQ 會處理排入佇列的發佈/訂閱指令，以及發佈/訂閱「訊息佇列介面 (MQI)」呼叫。

正在停止排入佇列的發佈/訂閱

開始之前

已淘汰排入佇列的發佈/訂閱。

請閱讀 [PSMODE](#) 的說明，以瞭解三種發佈/訂閱模式：

- COMPAT
- 已停用
- ENABLED

關於這項作業

設定 QMGR PSMODE 屬性，以停止已排入佇列的發佈/訂閱介面 (也稱為分配管理系統) 及/或發佈/訂閱引擎 (也稱為第 7 版發佈/訂閱)。若要停止排入佇列的發佈/訂閱，您需要將 PSMODE 設為 COMPAT。若要完全停止發佈/訂閱引擎，請將 PSMODE 設為 DISABLED。

程序

使用「WebSphere MQ 探險家」或 **runmqsc** 指令來停用排入佇列的發佈/訂閱介面。

範例

```
ALTER QMGR PSMODE(COMPAT)
```

新增串流

您可以手動新增串流，以便它們與從 Version 6.0 佇列管理程式移轉的串流同時存在。

開始之前

閱讀主題、[串流和主題](#)，以熟悉發佈/訂閱串流的運作方式。

關於這項作業

使用 PCF 指令、**runmqsc** 或 IBM WebSphere MQ Explorer 來執行這些步驟。

註: 您可以按任何順序執行步驟 1 和 2。只有在完成步驟 1 和 2 之後，才執行步驟 3。

程序

1. 定義與 Version 6.0 串流同名的本端佇列。
2. 定義與 Version 6.0 串流同名的本端主題。
3. 將佇列名稱新增至名單 SYSTEM.QPUBSUB.QUEUE.NAMELIST
4. 針對位於 Version 7.1 或更高版本且位於發佈/訂閱階層中的所有佇列管理程式，重複上述步驟。

新增中 'Sport'

在共用串流 'Sport' 的範例中，Version 6.0 和 Version 7.1 佇列管理程式在相同的發佈/訂閱階層中運作。Version 6.0 佇列管理程式會共用稱為 'Sport' 的串流。此範例顯示如何在 Version 7.1 佇列管理程式 'Sport' 上建立佇列及主題，主題字串 'Sport' 與第 6 版串流共用 'Sport'。

Version 7.1 發佈應用程式 (發佈至主題 'Sport'，含主題字串 'Soccer/Results') 會建立產生的主題字串 'Sport/Soccer/Results'。在 Version 7.1 佇列管理程式上，主題 'Sport' (主題字串為 'Soccer/Results') 的訂閱者會接收發佈。

在 Version 6.0 佇列管理程式上，串流 'Sport' (主題字串為 'Soccer/Results') 的訂閱者會接收發佈。

```
runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
define qlocal('Sport')
  1 : define qlocal('Sport')
AMQ8006: WebSphere MQ queue created.
define topic('Sport') topicstr('Sport')
  2 : define topic('Sport') topicstr('Sport')
AMQ8690: WebSphere MQ topic created.
alter namelist(SYSTEM.QPUBSUB.QUEUE.NAMELIST) NAMES('Sport', 'SYSTEM.BROKER.DEFAULT.STREAM',
'SYSTEM.BROKER.ADMIN.STREAM')
  3 : alter namelist(SYSTEM.QPUBSUB.QUEUE.NAMELIST) NAMES('Sport',
'SYSTEM.BROKER.DEFAULT.STREAM', 'SYSTEM.BROKER.ADMIN.STREAM')
AMQ8551: WebSphere MQ namelist changed.
```

註: 您需要在 **alter namelist** 指令中同時提供名稱清單物件中的現有名稱，以及您要新增的新名稱。

下一步

串流的相關資訊會傳遞給階層中的其他分配管理系統。

如果分配管理系統是 Version 6.0，請以 Version 6.0 分配管理系統來管理它。也就是說，您可以選擇手動建立串流佇列，或讓分配管理系統在需要時動態建立串流佇列。佇列基於模型佇列定義 SYSTEM.BROKER.MODEL.STREAM。

如果分配管理系統是 Version 7.1，您必須在階層中手動配置每一個 Version 7.1 佇列管理程式。

刪除串流

您可以從 IBM WebSphere MQ Version 7.1 或更新版本的佇列管理程式中刪除串流。

開始之前

在 IBM WebSphere MQ Version 7.1 中已淘汰使用排入佇列的發佈/訂閱。

在刪除串流之前，您必須確定沒有剩餘的串流訂閱，並靜止所有使用該串流的應用程式。如果發佈繼續流向已刪除的串流，則需要大量的管理努力才能將系統還原至正常工作狀態。

關於這項作業

如需從其連接的任何 Version 6.0 佇列管理程式中刪除串流的指示，請參閱 [刪除串流 \(v6.0 說明文件中的 ps11870_.htm\)](#)。

程序

1. 尋找管理此串流的所有已連接分配管理系統。
2. 取消所有分配管理系統上串流的所有訂閱。
3. 從名單 `SYSTEM.QPUBSUB.QUEUE.NAMELIST` 中移除佇列 (與串流同名)。
4. 從佇列中刪除或清除與串流同名的所有訊息。
5. 刪除與串流同名的佇列。
6. 刪除相關聯的主題物件。

下一步

1. 在管理串流的所有其他已連接 Version 7.1 或更新版本佇列管理程式上，重複步驟 3 至 5。
2. 從所有其他連接的 Version 6.0 或更早版本佇列管理程式中移除串流。

新增訂閱點

如何新增未由 **migmbbrk** 從 IBM WebSphere MQ Event Broker 或 IBM WebSphere MQ Message Broker 移轉的訂閱點。使用新的訂閱點，延伸您已從 IBM WebSphere MQ Event Broker 或 IBM WebSphere MQ Message Broker 移轉的現有已排入佇列的發佈/訂閱應用程式。

開始之前

1. 完成從 IBM WebSphere MQ Event Broker 和 IBM WebSphere MQ Message Broker Version 6.0 至 IBM WebSphere MQ Version 7.1 的移轉。
2. 請檢查是否尚未在 `SYSTEM.QPUBSUB.SUBPOINT.NAMELIST` 中定義訂閱點。
3. 請檢查是否有主題物件或主題字串與訂閱點同名。

關於這項作業

現有的 IBM WebSphere MQ Event Broker 應用程式會使用訂閱點。新的 IBM WebSphere MQ Version 7.1 應用程式不使用訂閱點，但它們可以使用訂閱點移轉機制，與執行的現有應用程式交互作業。

如果在移轉時未使用訂閱點，則 **migmbbrk** 可能尚未移轉該訂閱點。

您可能想要將訂閱點新增至從 IBM WebSphere MQ Event Broker 移轉的現有已排入佇列的發佈/訂閱程式。

如果已排入佇列的發佈/訂閱程式使用已從 IBM WebSphere MQ Version 6.0 或更早版本移轉的 MQRFH1 標頭，則訂閱點無法運作。

無需新增訂閱點，即可使用針對 IBM WebSphere MQ Version 7.1 撰寫的整合發佈/訂閱應用程式。

程序

1. 將訂閱點的名稱新增至 `SYSTEM.QPUBSUB.SUBPOINT.NAMELIST`。
 - 在 z/OS 上，**NLTYPE** 是 NONE(預設值)。
 - 在相同發佈/訂閱拓撲中連接的每個佇列管理程式上重複該步驟。
2. 以符合訂閱點名稱的主題字串來新增主題物件，最好為它提供訂閱點的名稱。
 - 如果訂閱點位於叢集中，請將主題物件新增為叢集主題主機上的叢集主題。

- 如果存在具有與訂閱點名稱相同的主題字串的主題物件，請使用現有的主題物件。您必須瞭解重複使用現有主題的訂閱點結果。如果現有主題是現有應用程式的一部分，則必須解決兩個同名主題之間的衝突。
 - 如果存在與訂閱點同名的主題物件，但主題字串不同，請建立不同名稱的主題。
3. 將 **Topic** 屬性 WILDCARD 設為值 BLOCK。
- 封鎖訂閱 # 或 * 會隔離訂閱點的萬用字元訂閱，請參閱 [萬用字元和訂閱點](#)。
4. 在主題物件中設定您需要的任何屬性。

範例

此範例顯示新增兩個訂閱點 (USD 和 GBP) 的 **runmqsc** 指令檔。

```
DEFINE TOPIC(USD) TOPICSTR(USD)
DEFINE TOPIC(GBP) TOPICSTR(GBP) WILDCARD(BLOCK)
ALTER NL(SYSTEM.QPUBSUB.SUBPOINT.NAMELIST) NAMES(SYSTEM.BROKER.DEFAULT.SUBPOINT, USD, GBP)
```

註:

1. 在使用 **ALTER** 指令新增的訂閱點清單中包括預設訂閱點。 **ALTER** 會刪除名單中的現有名稱。
2. 在變更名單之前定義主題。只有在佇列管理程式啟動及變更名稱清單時，佇列管理程式才會檢查名稱清單。

將佇列管理程式連接至分配管理系統階層

您可以將本端佇列管理程式連接至母項佇列管理程式，以修改分配管理系統階層。

開始之前

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

關於這項作業

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

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

程序

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

範例

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

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

```

QMNAME(QM2)                TYPE(LOCAL)
STATUS(ACTIVE)
AMQ8723: Display pub/sub status details.
QMNAME(QM1)                TYPE(PARENT)
STATUS(ACTIVE)

```

下一個範例顯示查詢 QM1 其連線的結果:

```

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

```

下一步

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

相關概念

[串流和主題](#)

[WebSphere MQ 發佈/訂閱傳訊簡介](#)

相關參考

[DISPLAY PUBSUB](#)

切斷佇列管理程式與分配管理系統階層的連線

中斷子項佇列管理程式與分配管理系統階層中母項佇列管理程式的連線。

關於這項作業

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

當佇列管理程式之間的連線執行時，會傳送更新母項的對應要求。

程序

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

範例

```

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

```

下一步

您可以刪除任何不再需要的串流、佇列及手動定義的通道。

配置佇列管理程式叢集

請使用本主題中的鏈結，以瞭解叢集如何運作、如何設計叢集配置，以及取得如何設定簡式叢集的範例。

開始之前

如需叢集作業概念的簡介，請參閱下列主題：

- [叢集運作方式](#)
- [第 135 頁的『叢集作業與分散式佇列作業的比較』](#)
- [第 137 頁的『叢集的元件』](#)

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 148 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 149 頁的『組織叢集』](#)
- [第 149 頁的『叢集命名慣例』](#)
- [第 150 頁的『重疊叢集』](#)

範例

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。

第 134 頁的圖 21 顯示稱為 DEMOCLSTR 的叢集，其中兩個佇列管理程式稱為 QM1 及 QM2。

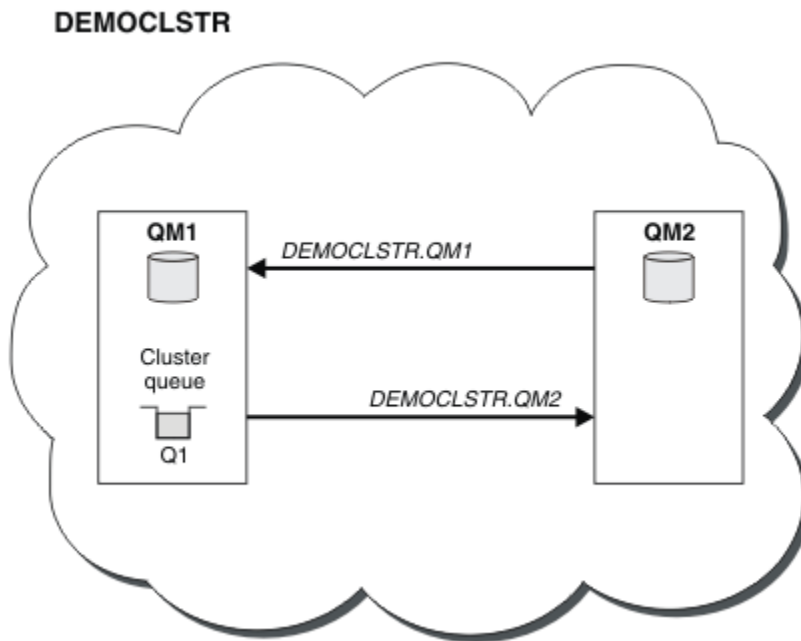


圖 21: 由兩個佇列管理程式組成的小型叢集

- 佇列管理程式具有完整名稱，例如 LONDON 及 NEWYORK。在進階及工作量平衡作業中使用相同的名稱。在 IBM WebSphere MQ for z/OS 上，佇列管理程式名稱限制為四個字元。
- 佇列管理程式的名稱暗示每一個佇列管理程式都位於個別機器上。您可以使用相同機器上的所有佇列管理程式來執行這些作業。
- 這些作業會使用 IBM WebSphere MQ Script 指令，因為系統管理者會使用 **MQSC** 指令來輸入它們。還有其他輸入指令的方法，包括使用較簡單的 IBM WebSphere MQ Explorer。使用 WebSphere MQ Script 指令的目的是示範作業中使用的 IBM WebSphere MQ 指令。

如需設定類似範例叢集的指示，請參閱 [第 154 頁的『設定新的叢集』](#)。

下一步

如需配置及使用叢集的相關資訊，請參閱下列主題：

- [第 152 頁的『在叢集中建立通訊』](#)
- [第 154 頁的『管理 IBM WebSphere MQ 叢集』](#)
- [第 210 頁的『將訊息遞送至叢集以及從叢集遞送訊息』](#)
- [第 221 頁的『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 151 頁的『叢集提示』](#)。

相關概念

叢集

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 WebSphere 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 WebSphere 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，並對遠端佇列定義進行存取權檢查。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 Version 7.5 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 Version 7.1 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 Version 7.1 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。

相關概念

[第 238 頁的『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。

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 136 頁的圖 22 顯示分散式佇列所需的元件。

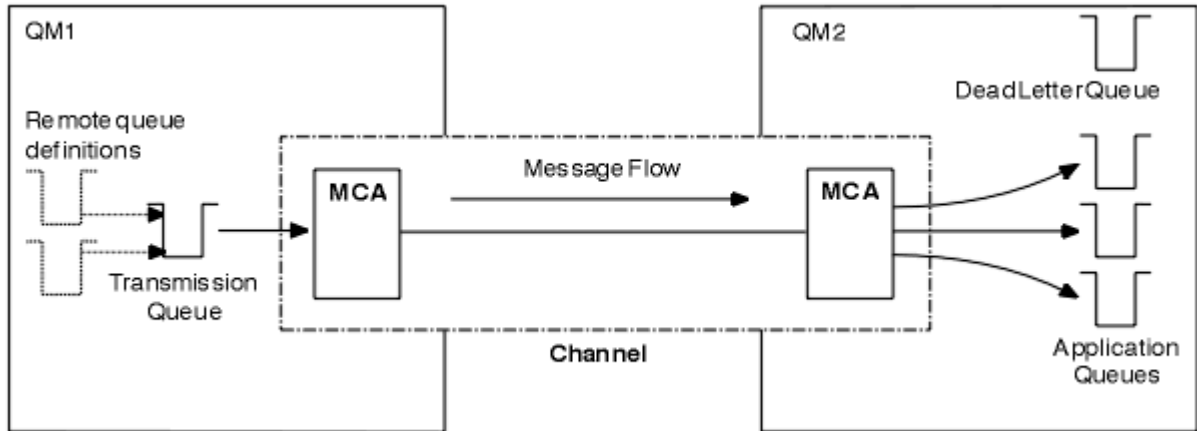


圖 22: 分散式佇列 (*distributed queuing*)

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 136 頁的圖 23，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。

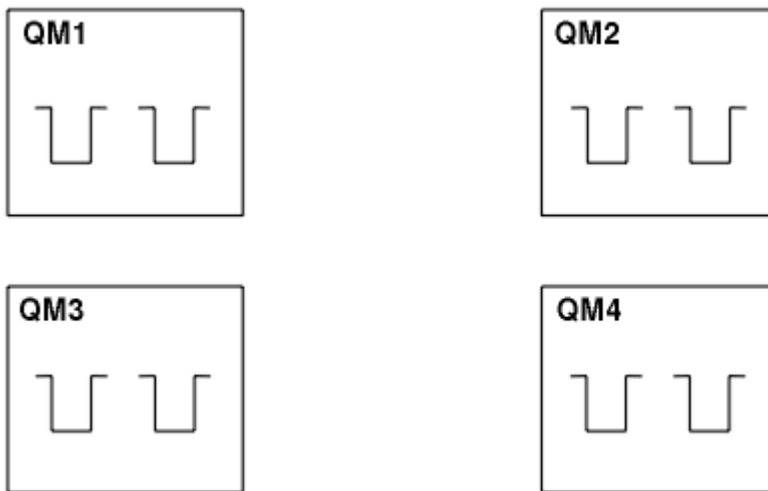


圖 23: 由四個佇列管理程式組成的網路

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 136 頁的圖 22 中顯示的網路，您可能具有下列定義：

表 24: 分散式佇列的定義		
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 136 頁的圖 22](#) 中顯示的網路，您需要下列定義：

表 25: 叢集作業的定義		
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。

叢集的元件

叢集由佇列管理程式、叢集儲存庫、叢集通道及叢集佇列組成。

如需每一個叢集元件的相關資訊，請參閱下列子主題：

相關概念

叢集

[第 135 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。

第 154 頁的『[管理 IBM WebSphere MQ 叢集](#)』
您可以建立、延伸及維護 IBM WebSphere MQ 叢集。

相關工作

第 133 頁的『[配置佇列管理程式叢集](#)』
請使用本主題中的鏈結，以瞭解叢集如何運作、如何設計叢集配置，以及取得如何設定簡式叢集的範例。

第 154 頁的『[設定新的叢集](#)』
請遵循下列指示來設定範例叢集。個別指示說明在 TCP/IP、LU 6.2 上設定叢集，以及使用單一傳輸佇列或多個傳輸佇列。測試叢集的運作方式是將訊息從一個佇列管理程式傳送至另一個佇列管理程式。

叢集儲存庫

儲存庫是屬於叢集成員之佇列管理程式的相關資訊集合。

儲存庫資訊包括佇列管理程式名稱、其位置、其通道、其管理的佇列，以及其他資訊。資訊以訊息形式儲存在稱為 `SYSTEM.CLUSTER.REPOSITORY.QUEUE` 的佇列中。佇列是其中一個預設物件。當您建立 WebSphere MQ 佇列管理程式時，會定義此佇列管理程式，但 WebSphere MQ for z/OS 除外。

通常，叢集中的兩個佇列管理程式會保留完整儲存庫。其餘佇列管理程式都會保留局部儲存庫。

完整儲存庫和局部儲存庫

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。叢集中的其他佇列管理程式具有局部儲存庫，其中包含完整儲存庫中的部分資訊。

局部儲存庫只包含佇列管理程式需要與其交換訊息之那些佇列管理程式的相關資訊。佇列管理程式會要求更新它們所需的資訊，因此如果它變更，完整儲存庫佇列管理程式會傳送新資訊給它們。在大部分時間內，局部儲存庫包含佇列管理程式在叢集內執行所需的所有資訊。當佇列管理程式需要某些其他資訊時，它會查詢完整儲存庫，再更新其局部儲存庫。佇列管理程式會使用稱為 `SYSTEM.CLUSTER.COMMAND.QUEUE` 的佇列來要求及接收儲存庫的更新項目。此佇列是其中一個預設物件。

叢集佇列管理程式

叢集佇列管理程式是屬於叢集成員的佇列管理程式。

佇列管理程式可以是多個叢集的成員。每一個叢集佇列管理程式在其所屬的所有叢集中都必須具有唯一的名稱。

叢集佇列管理程式可以管理向叢集中其他佇列管理程式通告的佇列。叢集佇列管理程式不需要管理或通告任何佇列。它可以將訊息饋送至叢集，且只會接收明確導向至叢集的回應，而不會接收至通告的佇列。

在 WebSphere MQ for z/OS 中，叢集佇列管理程式可以是佇列共用群組的成員。在此情況下，它會與相同佇列共用群組中的其他佇列管理程式共用其佇列定義。

叢集佇列管理程式是自主的。他們可以完全控制他們定義的佇列和通道。其他佇列管理程式 (相同佇列共用群組中的佇列管理程式除外) 無法修改其定義。儲存庫佇列管理程式不會控制叢集中其他佇列管理程式中的定義。它們會保留一組完整的所有定義，以便在必要時使用。叢集是佇列管理程式的聯合。

在叢集佇列管理程式上建立或變更定義之後，會將資訊傳送至完整儲存庫佇列管理程式。稍後會更新叢集中的其他儲存庫。

完整儲存庫佇列管理程式

完整儲存庫佇列管理程式是一種叢集佇列管理程式，可保留叢集資源的完整呈現。為了確保可用性，請在每一個叢集中設定兩個以上完整儲存庫佇列管理程式。完整儲存庫佇列管理程式會接收叢集中其他佇列管理程式所傳送的資訊，並更新其儲存庫。它們會彼此傳送訊息，以確保它們都保持最新的叢集相關新資訊。

佇列管理程式和儲存庫

每個叢集具有至少一個 (最好是兩個) 佇列管理程式，其中包含叢集中佇列管理程式、佇列及通道相關資訊的完整儲存庫。這些儲存庫也包含來自叢集中其他佇列管理程式的要求，以更新資訊。

其他佇列管理程式每一個都保留局部儲存庫，其中包含需要與其通訊之佇列及佇列管理程式子集的相關資訊。佇列管理程式會在第一次需要存取另一個佇列或佇列管理程式時進行查詢，以建置其局部儲存庫。他們要求收到有關該佇列或佇列管理程式的任何新資訊通知。

每一個佇列管理程式都會將其儲存庫資訊儲存在稱為 `SYSTEM.CLUSTER.REPOSITORY.QUEUE` 的佇列上的訊息中。佇列管理程式會在稱為 `SYSTEM.CLUSTER.COMMAND.QUEUE` 的佇列上交換訊息中的儲存庫資訊。

每一個加入叢集的佇列管理程式都會定義一個叢集傳送端 `CLUSDR` 通道至其中一個儲存庫。它會立即得知叢集中哪些其他佇列管理程式保留完整儲存庫。從那時開始，佇列管理程式可以從任何儲存庫要求資訊。當佇列管理程式將資訊傳送至選擇的儲存庫時，它也會將資訊傳送至另一個儲存庫 (如果有的話)。

當管理完整儲存庫的佇列管理程式從鏈結至它的其中一個佇列管理程式接收新資訊時，會更新完整儲存庫。新資訊也會傳送至另一個儲存庫，以減少儲存庫佇列管理程式無法運作時延遲它的風險。因為所有資訊都會傳送兩次，所以儲存庫必須捨棄重複項目。每一個資訊項目都有一個序號，供儲存庫用來識別重複項目。透過交換訊息，所有儲存庫彼此保持一致。

叢集佇列數

叢集佇列是由叢集佇列管理程式所管理的佇列，並可提供給叢集的其他佇列管理程式使用。

在管理佇列的叢集佇列管理程式上，將叢集佇列定義為本端佇列。指定佇列所屬的叢集名稱。下列範例顯示使用 `CLUSTER` 選項來定義叢集佇列的 `runmqsc` 指令：

```
DEFINE QLOCAL(Q1) CLUSTER(SALES)
```

叢集佇列定義會通告至叢集中的其他佇列管理程式。叢集中的其他佇列管理程式不需要相對應的遠端佇列定義，就可以將訊息放置在叢集佇列中。可以使用叢集名稱清單在多個叢集中通告叢集佇列。

在通告佇列時，叢集中的任何佇列管理程式都可以將訊息置入該佇列中。若要放置訊息，佇列管理程式必須從完整儲存庫中找出管理該佇列的位置。然後，它會將一些遞送資訊新增到訊息中，並將訊息放置在叢集傳輸佇列上。

叢集佇列可以是 IBM WebSphere MQ for z/OS 中的佇列共用群組成員所共用的佇列。

連結中

您可以建立叢集，其中多個佇列管理程式會管理相同叢集佇列的實例。請確定序列中的所有訊息都已傳送至佇列的相同實例。您可以在 `MQOPEN` 呼叫上使用 `MQOO_BIND_ON_OPEN` 選項，將一系列訊息連結至特定佇列。

叢集傳輸佇列

除了 z/OS 之外，佇列管理程式可以將叢集中其他佇列管理程式的訊息儲存在多個傳輸佇列上。您可以採用兩種不同的方式來配置佇列管理程式，以將訊息儲存在多個叢集傳輸佇列上。如果您將佇列管理程式屬性 `DEFCLXQ` 設定為 `CHANNEL`，則會從 `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE` 中為每一個叢集傳送端通道自動建立一個不同的叢集傳輸佇列。如果將 `CLCHNAME` 傳輸佇列選項設定為符合一個以上的叢集傳送端通道，則佇列管理程式可以將相符通道的訊息儲存在該傳輸佇列上。



小心：如果您將專用 `SYSTEM.CLUSTER.TRANSMIT.QUEUES` 與從舊版產品升級的佇列管理程式搭配使用，請確保 `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE` 已將 `SHARE/NOSHARE` 選項設為 **SHARE**。

在傳送之前，會將不同佇列管理程式上叢集佇列的訊息放置在叢集傳輸佇列上。叢集傳送端通道會將訊息從叢集傳輸佇列傳送至其他佇列管理程式上的叢集接收端通道。依預設，一個系統定義的叢集傳輸佇列會保留要傳送至其他叢集佇列管理程式的所有訊息。佇列稱為 `SYSTEM.CLUSTER.TRANSMIT.QUEUE`。屬於叢集的佇列管理程式可以將此叢集傳輸佇列上的訊息傳送至相同叢集中的任何其他佇列管理程式。

依預設，除了 z/OS 之外，每個佇列管理程式都會建立單一 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 佇列的定義。

在 z/OS 以外的平台上，您可以配置佇列管理程式，以使用多個傳輸佇列將訊息傳送至其他叢集佇列管理程式。您可以手動定義其他叢集傳輸佇列，或讓佇列管理程式自動建立佇列。

若要讓佇列管理程式自動建立佇列，請將佇列管理程式屬性 DEFCLXQ 從 SCTQ 變更為 CHANNEL。結果是佇列管理程式會為所建立的每一個叢集傳送端通道建立個別叢集傳輸佇列。傳輸佇列會從模型佇列 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE 建立為永久動態佇列。每一個永久動態佇列的名稱都是 SYSTEM.CLUSTER.TRANSMIT.ChannelName。在本端傳輸佇列屬性 CLCHNAME 中設定與每一個永久動態叢集傳輸佇列相關聯的叢集傳送端通道名稱。遠端叢集佇列管理程式的訊息會放置在相關聯叢集傳送端通道的永久動態叢集傳輸佇列上，而不是放置在 SYSTEM.CLUSTER.TRANSMIT.QUEUE 上。

若要手動建立叢集傳輸佇列，請建立本端佇列，並將 USAGE 屬性設為 XMITQ，並將 CLCHNAME 屬性設為可解析為一或多個叢集傳送端通道的通用通道名稱；請參閱 ClusterChannel 名稱。如果您手動建立叢集傳輸佇列，您可以選擇將傳輸佇列與單一叢集傳送端通道或多個叢集傳送端通道相關聯。CLCHNAME 屬性是通用名稱，表示您可以在名稱中放置多個萬用字元 "*"。

除了您手動建立以將佇列管理程式連接至完整儲存庫的起始叢集傳送端通道之外，會自動建立叢集傳送端通道。當有訊息要傳送至叢集佇列管理程式時，會自動建立它們。它們的建立名稱與目的地佇列管理程式上接收該特定叢集之叢集訊息的叢集接收端通道名稱相同。

如果您遵循叢集接收端通道的命名慣例，則可以定義 CLCHNAME 的通用值，以將不同類型的叢集訊息過濾到不同的傳輸佇列。例如，如果您遵循 ClusterName.QmgrName 叢集接收端通道的命名慣例，則通用名稱 ClusterName.* 會將不同叢集的訊息過濾到不同的傳輸佇列上。您必須手動定義傳輸佇列，並將每一個傳輸佇列中的 CLCHNAME 設為 ClusterName.*。

叢集傳輸佇列與叢集傳送端通道的關聯變更不會立即生效。叢集傳送端通道所服務的目前相關聯傳輸佇列可能包含正在由叢集傳送端通道傳送的訊息。只有在叢集傳送端通道未處理目前相關聯傳輸佇列上的訊息時，佇列管理程式才能將叢集傳送端通道的關聯變更為不同的傳輸佇列。當傳輸佇列中沒有要由叢集傳送端通道處理的訊息時，或當訊息處理已暫停且叢集傳送端通道沒有 "進行中" 訊息時，就會發生這種情況。發生此情況時，叢集傳送端通道的任何未處理訊息都會傳送至新關聯的傳輸佇列，且叢集傳送端通道的關聯會變更。

您可以建立遠端佇列定義，以解析為叢集傳輸佇列。在定義中，佇列管理程式 QMX 與本端佇列管理程式位於相同的叢集中，且沒有傳輸佇列 QMX。

```
DEFINE QREMOTE(A) RNAME(B) RQMNAME(QMX)
```

在佇列名稱解析期間，叢集傳輸佇列優先於預設傳輸佇列。放入 A 的訊息會儲存在叢集傳輸佇列中，然後傳送至 QMX 上的遠端佇列 B。

佇列管理程式也可以與不屬於叢集的其他佇列管理程式進行通訊。您必須以在分散式佇列環境中的相同方式，將通道及傳輸佇列定義至另一個佇列管理程式。

註：應用程式必須寫入解析為叢集傳輸佇列的佇列，且不得直接寫入叢集傳輸佇列。

自動定義遠端佇列

叢集中的佇列管理程式不需要叢集中遠端佇列的遠端佇列定義。叢集佇列管理程式會從完整儲存庫中尋找遠端佇列的位置。它會將遞送資訊新增至訊息，並將它放在叢集傳輸佇列中。WebSphere MQ 會自動建立相當於遠端佇列定義的定義，以便可以傳送訊息。

您無法變更或刪除自動建立的遠端佇列定義。不過，搭配使用 DISPLAY QUEUE runmqsc 指令與 CLUSINFO 屬性，您可以檢視佇列管理程式上的所有本端佇列，以及所有叢集佇列 (包括遠端佇列管理程式上的叢集佇列)。例如：

```
DISPLAY QUEUE(*) CLUSINFO
```

相關參考

[ClusterChannel 名稱 \(MQCHAR20\)](#)

叢集通道

您必須為叢集中的佇列管理程式定義叢集接收端和叢集傳送端通道。特殊考量適用於完整儲存庫。

在叢集內，訊息會在特殊類型通道上的叢集佇列管理程式之間配送，您需要叢集接收端通道定義及叢集傳送端通道定義。

叢集傳送端通道: CLUSSDR

手動將叢集傳送端通道定義至叢集中每一個佇列管理程式的完整儲存庫。叢集傳送端定義可讓佇列管理程式起始聯絡叢集。它會將佇列管理程式優先選擇傳送叢集資訊的完整儲存庫佇列管理程式命名為該佇列管理程式。叢集傳送端通道用來通知儲存庫佇列管理程式的狀態有任何變更。例如，如果新增或移除佇列。它也可以用來傳輸訊息。

完整儲存庫佇列管理程式本身有指向彼此的叢集傳送端通道。它們會使用這些通道來互相溝通叢集狀態變更。

CLUSSDR 通道定義指向哪個完整儲存庫不重要。一旦完成起始聯絡，就會視需要自動定義進一步的叢集佇列管理程式物件。佇列管理程式可以將叢集資訊傳送至每個完整儲存庫，並將訊息傳送至每個佇列管理程式。

在完整儲存庫佇列管理程式上所做的 CLUSSDR 定義是特殊的。完整儲存庫所交換的所有更新項目都只會在這些通道上流動。管理者明確控制完整儲存庫的網路。管理者必須定義從每個完整儲存庫佇列管理程式到叢集中每個其他完整儲存庫佇列管理程式的 CLUSSDR 通道。管理者必須手動在完整儲存庫佇列管理程式上建立 CLUSSDR 定義，而不是將它們保留為自動定義。

叢集傳送端通道只能定義為將局部儲存庫連接至完整儲存庫，或將兩個完整儲存庫連接在一起。手動配置用於處理局部儲存庫或不在叢集中的佇列管理程式的 CLUSSDR 通道會導致發出錯誤訊息，例如 AMQ9427 及 AMQ9428。

雖然有時這種暫時狀況（例如修改完整儲存庫的位置時）可能無法避免，但應該儘快刪除不正確的定義，以停止發出這些錯誤。

叢集接收端通道: CLUSRCVR

叢集接收端通道定義通道的結尾，叢集佇列管理程式可以在此通道上接收來自叢集中其他佇列管理程式的訊息。

叢集接收端通道也可以傳送指向本端儲存庫之叢集資訊的相關資訊。透過定義叢集接收端通道，佇列管理程式會顯示可用來接收訊息的其他叢集佇列管理程式。每一個叢集佇列管理程式至少需要一個叢集接收端通道。

CLUSRCVR 定義可讓其他佇列管理程式自動定義對應的 CLUSSDR 通道定義。

相關概念

第 141 頁的『自動定義叢集通道』

佇列管理程式必須有叢集傳送端通道的定義，才能將訊息傳送至遠端目的地。建立叢集的佇列管理程式並建立其起始 CLUSSDR 及 CLUSRCVR 定義之後，WebSphere MQ 會在需要時自動建立叢集傳送端通道定義。您無法修改自動定義的叢集傳送端通道。您可以使用通道自動定義結束程式來修改其行為。

自動定義叢集通道

佇列管理程式必須有叢集傳送端通道的定義，才能將訊息傳送至遠端目的地。建立叢集的佇列管理程式並建立其起始 CLUSSDR 及 CLUSRCVR 定義之後，WebSphere MQ 會在需要時自動建立叢集傳送端通道定義。您無法修改自動定義的叢集傳送端通道。您可以使用通道自動定義結束程式來修改其行為。

當定義通道的叢集傳送端和叢集接收端時，會啟動通道。自動定義通道會保持作用中，直到不再需要它，並使用正常斷線間隔規則來關閉它為止。

自動定義的叢集傳送端通道會從接收端佇列管理程式上對應的叢集接收端通道定義取得其屬性。即使有手動定義的叢集傳送端通道，也會自動修改其屬性，以確保它們符合對應的叢集接收端定義。例如，假設您定義 CLUSRCVR 而未在 CONNAME 參數中指定埠號，並手動定義 CLUSSDR 而確實指定埠號。當自動定義的 CLUSSDR 取代手動定義的一個時，埠號（取自 CLUSRCVR）會變成空白。使用預設埠號且通道失敗。

您無法修改自動定義的叢集傳送端定義。

您無法使用 DISPLAY CHANNEL **runmqsc** 指令來查看自動定義的通道。若要查看自動定義的通道，請使用下列指令：

```
DISPLAY CLUSQMGR(qMgrName)
```

若要顯示自動定義的 CLUSSDR 通道 (對應於您所建立的 CLUSRCVR 通道定義) 的狀態, 請使用下列指令:

```
DISPLAY CHSTATUS(channelname)
```

如果您想要撰寫使用者結束程式來自訂叢集傳送端通道或叢集接收端通道, 您可以使用 WebSphere MQ 通道自動定義結束程式。您可以在叢集環境中使用通道自動定義結束程式來執行下列動作:

- 自訂通訊定義, 亦即 SNA LU6.2 名稱
- 新增或移除其他結束程式, 例如安全結束程式
- 變更通道結束程式的名稱。您必須變更 CLUSSDR 通道結束程式的名稱, 因為 CLUSSDR 通道結束程式名稱是從 CLUSRCVR 通道定義自動產生。自動產生的名稱可能錯誤, 如果通道兩端位於不同平台上, 幾乎可以肯定是錯誤的。在不同平台上, 結束程式名稱的格式不同。例如, 在 Windows 上是 SCYEXIT('drive:\path\library(secexit)')。

非 z/OS 平台上的結束程式名稱一般格式為 *path/library(function)*。如果 *function* 存在, 則最多使用八個字元。否則, 會使用截斷為八個字元的 *library*。例如:

- /var/mqm/exits/myExit.so(MsgExit) 轉換為 MSGEXIT
- /var/mqm/exits/myExit 轉換為 MYEXIT
- /var/mqm/exits/myExit.so(ExitLongName) 轉換為 EXITLONG

若要讓出埠 (TCP) 通道使用特定 IP 位址、埠或埠範圍, 請使用通道屬性 LOCLADDR。如果您有多個網路卡, 且想要通道使用特定網路卡來進行出埠通訊, 則 LOCLADDR 非常有用。

若要在 CLUSSDR 通道上指定虛擬 IP 位址, 請在手動定義的 CLUSSDR 上使用 LOCLADDR 中的 IP 位址。若要指定埠範圍, 請使用 CLUSRCVR 中的埠範圍。

如果叢集需要使用 LOCLADDR 來取得出埠通訊通道以連結至特定 IP 位址, 您必須撰寫「通道自動定義結束程式」, 以強制 LOCLADDR 值進入任何自動定義的 CLUSSDR 通道, 且必須在手動定義的 CLUSSDR 通道中指定它。

除非所有佇列管理程式都位於相同的伺服器上, 否則請勿將 IP 位址放置在 CLUSRCVR 通道的 LOCLADDR 欄位中。LOCLADDR IP 位址會延伸到使用 CLUSRCVR 通道連接之所有佇列管理程式的自動定義 CLUSSDR 通道。

如果您想要叢集中的所有佇列管理程式在其所有出埠通訊中使用特定埠或埠範圍, 請將埠號或埠範圍放置在 CLUSRCVR 通道的 LOCLADDR 中。

distributed 在分散式平台上, 可以設定預設本端位址值, 該值將用於未定義本端位址的所有傳送端通道。在啟動佇列管理程式之前, 請先設定 MQ_LCLADDR 環境變數來定義預設值。值的格式符合 MQSC 屬性 LOCLADDR 的格式。

自動定義叢集傳送端通道定義不是實際通道物件。在 z/OS 以外的平台上, OAM (物件權限管理程式) 不知道它們是否存在。如果您嘗試在自動定義的叢集傳送端通道上發出 start、stop、ping、reset 或 resolve 指令, 則 OAM 會檢查您是否獲授權在叢集的叢集接收端通道上執行相同的動作。

如果叢集需要使用 PROPCTL, 從 WebSphere MQ 第 7 版佇列管理程式傳送至舊版 WebSphere MQ 上佇列管理程式的訊息中移除應用程式標頭 (例如 RFH2), 您必須撰寫「通道自動定義結束程式」, 強制 PROPCTL 值為 NONE。因為叢集傳送端通道具有根據對應叢集接收端通道的定義, 所以需要結束程式。由於較早層次叢集接收端通道沒有 PROPCTL 屬性, 自動叢集傳送端通道會將屬性設為 COMPAT。不論在手動叢集傳送端通道上設定的內容為何, 都會將屬性設為 COMPAT。

相關參考

[本端位址 \(LOCLADDR\)](#)

預設叢集物件

使用 WebSphere MQ 叢集時建立預設叢集物件。它們包含在您定義佇列管理程式除外。

您可以執行 MQSC 或 PCF 指令, 以與任何其他通道定義相同的方式變更預設通道定義。

請勿變更預設佇列定義, SYSTEM.CLUSTER.HISTORY.QUEUE 除外。

SYSTEM.CLUSTER.COMMAND.QUEUE

叢集中的每一個佇列管理程式都有一個稱為 SYSTEM.CLUSTER.COMMAND.QUEUE 的本端佇列，用來將訊息傳送至完整儲存庫。此訊息包含任何新的或已變更的佇列管理程式相關資訊，或其他佇列管理程式相關資訊的任何要求。SYSTEM.CLUSTER.COMMAND.QUEUE 通常是空的。

SYSTEM.CLUSTER.HISTORY.QUEUE

叢集中的每一個佇列管理程式都有一個稱為 SYSTEM.CLUSTER.HISTORY.QUEUE 的本端佇列。SYSTEM.CLUSTER.HISTORY.QUEUE 用來儲存叢集狀態資訊的歷程，以作為服務用途。

在預設物件設定中，SYSTEM.CLUSTER.HISTORY.QUEUE 設為 PUT(ENABLED)。若要暫停歷程收集，請將設定變更為 PUT(DISABLED)。

SYSTEM.CLUSTER.REPOSITORY.QUEUE

叢集中的每一個佇列管理程式都有一個稱為 SYSTEM.CLUSTER.REPOSITORY.QUEUE 的本端佇列。此佇列用來儲存所有完整儲存庫資訊。此佇列通常不是空的。

SYSTEM.CLUSTER.TRANSMIT.QUEUE

每一個佇列管理程式都有一個稱為 SYSTEM.CLUSTER.TRANSMIT.QUEUE 的本端佇列定義。SYSTEM.CLUSTER.TRANSMIT.QUEUE 是所有訊息到叢集內所有佇列及佇列管理程式的預設傳輸佇列。您可以變更佇列管理程式屬性 DEFXMITQ (除外)，將每一個叢集傳送端通道的預設傳輸佇列變更為 SYSTEM.CLUSTER.TRANSMIT.ChannelName。您無法刪除 SYSTEM.CLUSTER.TRANSMIT.QUEUE。它也用來定義授權檢查所使用的預設傳輸佇列是 SYSTEM.CLUSTER.TRANSMIT.QUEUE 還是 SYSTEM.CLUSTER.TRANSMIT.ChannelName。

SYSTEM.DEF.CLUSRCVR

每一個叢集都有一個稱為 SYSTEM.DEF.CLUSRCVR 的預設 CLUSRCVR 通道定義。SYSTEM.DEF.CLUSRCVR 用來為您在叢集中的佇列管理程式上建立叢集接收端通道時未指定的任何屬性提供預設值。

SYSTEM.DEF.CLUSSDR

每一個叢集都有一個稱為 SYSTEM.DEF.CLUSSDR 的預設 CLUSSDR 通道定義。SYSTEM.DEF.CLUSSDR 用來為您在叢集中的佇列管理程式上建立叢集傳送端通道時未指定的任何屬性提供預設值。

叢集傳輸佇列及叢集傳送端通道

叢集佇列管理程式之間的訊息儲存在叢集傳輸佇列上，並由叢集傳送端通道轉遞。

當您顯示叢集傳送端通道時，會看到它與傳輸佇列相關聯。在任何時間點，叢集傳送端通道都會與一個傳輸佇列相關聯。如果您變更通道的配置，則下次啟動時可能會切換至不同的傳輸佇列。執行下列 MQSC 指令，以顯示與叢集傳送端通道相關聯的傳輸佇列：

```
DISPLAY CHSTATUS(*) WHERE(CHLTYPE EQ CLUSSDR)
```

```
AMQ8417: Display Channel Status details.
CHANNEL(TO.QM2)          CHLTYPE(CLUSSDR)
CONNNAME(9.146.163.190(1416))  CURRENT
RQMNAME(QM2)            STATUS(STOPPED)
SUBSTATE( )              XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

當通道再次啟動時，已停止叢集傳送端通道的已儲存通道狀態中所顯示的傳輸佇列可能會變更。第 144 頁的『依叢集傳送端通道選取預設傳輸佇列』說明選取預設傳輸佇列的處理程序；第 144 頁的『依叢集傳送端通道選取手動定義的傳輸佇列』說明選取手動定義的傳輸佇列的處理程序。

當任何叢集傳送端通道啟動時，它會重新檢查其與傳輸佇列的關聯。如果傳輸佇列或佇列管理程式預設值的配置變更，它可能會重新關聯通道與不同的傳輸佇列。如果通道由於配置變更而以不同的傳輸佇列重新啟動，則會進行將訊息傳送至新關聯傳輸佇列的處理程序。第 145 頁的『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』說明將叢集傳送端通道從一個傳輸佇列傳送至另一個傳輸佇列的處理程序。

叢集傳送端通道的行為不同於傳送端和伺服器通道。它們會保持與相同的傳輸佇列相關聯，直到通道屬性 XMITQ 變更為止。如果您變更傳送端或伺服器通道上的傳輸佇列屬性，並重新啟動它，則不會將訊息從舊傳輸佇列傳送至新傳輸佇列。

叢集傳送端通道與傳送端或伺服器通道之間的另一個差異是多個叢集傳送端通道可以開啟叢集傳輸佇列，但只有一個傳送端或伺服器通道可以開啟正常傳輸佇列。在 Version 7.5 叢集連線共用單一叢集傳輸佇列

SYSTEM.CLUSTER.TRANSMIT.QUEUE 之前。從 Version 7.5 開始，您可以選擇叢集傳送端通道不共用傳輸佇列。不強制專屬性；它是配置的結果。您可以配置訊息在叢集中所採用的路徑，使其不會與在其他應用程式之間流動的訊息共用任何傳輸佇列或通道。請參閱第 241 頁的『叢集作業：規劃如何配置叢集傳輸佇列』及第 173 頁的『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』。

依叢集傳送端通道選取預設傳輸佇列

叢集傳輸佇列是系統預設佇列（名稱以 SYSTEM.CLUSTER.TRANSMIT 開頭）或手動定義的佇列。叢集傳送端通道以兩種方式之一與叢集傳輸佇列相關聯：透過預設叢集傳輸佇列機制或手動配置。

預設叢集傳輸佇列設為佇列管理程式屬性 **DEFCLXQ**。其值為 SCTQ 或 CHANNEL。新的及已移轉的佇列管理程式會設為 SCTQ。您可以將值變更為 CHANNEL。

如果設定 SCTQ，則預設叢集傳輸佇列為 SYSTEM.CLUSTER.TRANSMIT.QUEUE。每個叢集傳送端通道都可以開啟此佇列。開啟佇列的叢集傳送端通道是與手動定義的叢集傳輸佇列無關的通道。

如果設定 CHANNEL，則佇列管理程式可以為每個叢集傳送端通道建立個別的永久動態傳輸佇列。每一個佇列都命名為 SYSTEM.CLUSTER.TRANSMIT.ChannelName，並從模型佇列 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE 建立。每一個未與手動定義的叢集傳輸佇列相關聯的叢集傳送端通道，都會與永久動態叢集傳輸佇列相關聯。當佇列管理程式需要這個叢集傳送端通道所提供之叢集目的地的個別叢集傳輸佇列，且沒有任何佇列存在時，就會建立佇列。

部分叢集目的地可以由與手動定義的傳輸佇列相關聯的叢集傳送端通道提供，其他則由一或多個預設佇列提供。在叢集傳送端通道與傳輸佇列的關聯中，手動定義的傳輸佇列一律優先於預設傳輸佇列。

第 144 頁的圖 24 中說明叢集傳輸佇列的優先順序。唯一與手動定義的叢集傳輸佇列沒有關聯的叢集傳送端通道是 CS.QM1。它不會與手動定義的傳輸佇列相關聯，因為傳輸佇列的 **CLCHNAME** 屬性中沒有任何通道名稱符合 CS.QM1。

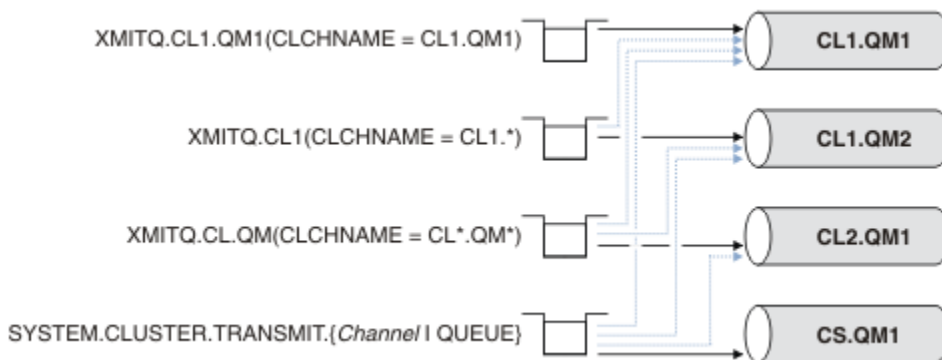


圖 24: 傳輸佇列/叢集傳送端通道優先順序

依叢集傳送端通道選取手動定義的傳輸佇列

手動定義的佇列將傳輸佇列屬性 **USAGE** 屬性設為 XMITQ，並將叢集通道名稱屬性 **CLCHNAME** 設為特定或通用通道名稱。

如果 **CLCHNAME** 佇列屬性中的名稱符合叢集傳送端通道名稱，則通道會與佇列相關聯。如果名稱不包含萬用字元，則名稱完全相符；如果名稱包含萬用字元，則名稱最符合。

如果多個傳輸佇列上的 **CLCHNAME** 定義符合相同的叢集傳送端通道，則這些定義會被認為重疊。若要解決語義不明確，相符項之間有優先順序。完全相符一律優先。第 144 頁的圖 24 顯示傳輸佇列與叢集傳送端通道之間的關聯。黑色箭頭顯示實際關聯，而灰色箭頭則顯示潛在關聯。第 144 頁的圖 24 中傳輸佇列的優先順序為：

XMITQ.CL1.QM1

傳輸佇列 XMITQ.CL1.QM1 將其 **CLCHNAME** 屬性設為 CL1.QM1。**CLCHNAME** 屬性 CL1.QM1 的定義沒有萬用字元，其優先順序高於其他傳輸佇列上定義且符合萬用字元的任何其他 **CLCHNAME** 屬性。佇列管理程式會儲存 XMITQ.CL1.QM1 傳輸佇列上 CL1.QM1 叢集傳送端通道所要傳送的任何叢集訊息。唯一例外是多個傳輸佇列將其 **CLCHNAME** 屬性設為 CL1.QM1。在該情況下，佇列管理程式會將 CL1.QM1 叢

集傳送端通道的訊息儲存在其中任何佇列上。當通道啟動時，它會任意選取佇列。當通道再次啟動時，它可能會選取不同的佇列。

XMITQ.CL1

傳輸佇列 XMITQ.CL1 將其 **CLCHNAME** 屬性設為 CL1.*。 **CLCHNAME** 屬性 CL1.* 的定義有一個尾端萬用字元，它符合任何以 CL1. 開頭的叢集傳送端通道名稱。佇列管理程式會在傳輸佇列 XMITQ.CL1 上儲存任何叢集傳送端通道 (其名稱以 CL1. 開頭) 所要傳送的任何叢集訊息，除非有傳輸佇列具有更具體的相符項 (例如佇列 XMITQ.CL1.QM1)。一個尾端萬用字元會使定義比不含萬用字元的定義更不明確，且比含有多個萬用字元或後面有多個尾端字元的萬用字元的定義更明確。

XMITQ.CL.QM

XMITQ.CL.QM 是傳輸佇列的名稱，其 **CLCHNAME** 屬性設為 CL*.QM*。CL*.QM* 的定義有兩個萬用字元，它們符合任何以 CL. 開頭且包括或結束於 QM 的叢集傳送端通道名稱。比對小於具有一個萬用字元的比對。

SYSTEM.CLUSTER.TRANSMIT.channelName | QUEUE

如果沒有傳輸佇列具有符合佇列管理程式要使用之叢集傳送端通道名稱的 **CLCHNAME** 屬性，則佇列管理程式會使用預設叢集傳輸佇列。預設叢集傳輸佇列是單一系統叢集傳輸佇列 SYSTEM.CLUSTER.TRANSMIT.QUEUE，或佇列管理程式為特定叢集傳送端通道 SYSTEM.CLUSTER.TRANSMIT.channelName 建立的系統叢集傳輸佇列。哪個佇列是預設值取決於佇列管理程式 **DEFXMITQ** 屬性的設定。

提示: 除非您明確需要重疊定義，否則請避免它們，因為它們可能會導致難以理解的複雜配置。

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

若要變更叢集傳送端通道與叢集傳輸佇列的關聯，請隨時變更任何傳輸佇列的 **CLCHNAME** 參數或佇列管理程式參數 **DEFCLXQ**。沒有立即發生的事。只有在通道啟動時才會發生變更。當它啟動時，它會檢查是否繼續轉遞來自相同傳輸佇列的訊息。三種變更會變更叢集傳送端通道與傳輸佇列的關聯。

1. 重新定義叢集傳送端通道目前相關聯之傳輸佇列的 **CLCHNAME** 參數，使其不明確或空白，或在通道停止時刪除叢集傳輸佇列。

部分其他叢集傳輸佇列現在可能更符合通道名稱。或者，如果沒有其他傳輸佇列符合叢集傳送端通道的名稱，則關聯必須回復為預設傳輸佇列。

2. 重新定義任何其他叢集傳輸佇列的 **CLCHNAME** 參數，或新增叢集傳輸佇列。

另一個傳輸佇列的 **CLCHNAME** 參數現在可能比叢集傳送端通道目前相關聯的傳輸佇列更符合叢集傳送端通道。如果叢集傳送端通道目前與預設叢集傳輸佇列相關聯，它可能會變成與手動定義的叢集傳輸佇列相關聯。

3. 如果叢集傳送端通道目前與預設叢集傳輸佇列相關聯，請變更 **DEFCLXQ** 佇列管理程式參數。

如果叢集傳送端通道的關聯變更，當通道啟動時，它會將其關聯切換至新的傳輸佇列。在切換期間，它可確保不會遺失任何訊息。訊息會依照通道將訊息傳送至遠端佇列管理程式的順序，傳送至新的傳輸佇列。

記住: 與叢集中的任何訊息轉遞一樣，您必須將訊息放入群組中，以確保依序遞送必須依序遞送的訊息。在極少數情況下，叢集中的訊息可能不正常。

切換程序會經歷下列交易式步驟。如果交換器處理程序岔斷，則當通道重新重新啟動時，會回復現行交易式步驟。

步驟 1-處理來自原始傳輸佇列的訊息

叢集傳送端通道與新的傳輸佇列相關聯，它可能與其他叢集傳送端通道共用。叢集傳送端通道的訊息會繼續放置在原始傳輸佇列上。過渡交換器處理程序會將訊息從原始傳輸佇列傳送至新的傳輸佇列。叢集傳送端通道會將訊息從新的傳輸佇列轉遞至叢集接收端通道。通道狀態顯示仍與舊傳輸佇列相關聯的叢集傳送端通道。

交換器處理程序也會繼續傳送新送達的訊息。此步驟會繼續，直到交換器處理程序要轉遞的剩餘訊息數達到零為止。當訊息數達到零時，程序會移至步驟 2。

在步驟 1 期間，通道的磁碟活動會增加。持續訊息從第一個傳輸佇列確定到第二個傳輸佇列。在正常傳送訊息的過程中，當訊息放置在傳輸佇列上並從傳輸佇列中移除時，除了此磁碟活動之外，還會確定其他訊息。理想情況下，在切換處理程序期間不會有任何訊息到達，因此可以盡快進行轉移。如果訊息確實到達，則由交換器處理程序進行處理。

步驟 2-處理來自新傳輸佇列的訊息

只要叢集傳送端通道的原始傳輸佇列中沒有任何訊息，新訊息就會直接放在新的傳輸佇列中。通道狀態顯示叢集傳送端通道與新的傳輸佇列相關聯。下列訊息會寫入佇列管理程式錯誤日誌："AMQ7341 通道 `ChannelName` 的傳輸佇列為 `QueueName`。"

多個叢集傳輸佇列及叢集傳輸佇列屬性

您可以選擇將叢集訊息轉遞至不同的佇列管理程式，以儲存單一叢集傳輸佇列或多個佇列上的訊息。使用一個佇列，您有一組要設定及查詢的叢集傳輸佇列屬性，使用多個佇列，您有多個集。對於部分屬性，具有多個集是優點：例如，查詢佇列深度會告訴您有多少訊息正在等待由一個或一組通道轉遞，而不是由所有通道轉遞。對於其他屬性，具有多個集是缺點：例如，您可能不想為每個叢集傳輸佇列配置相同的存取權。因此，一律會針對 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 的設定檔來檢查存取權，而不會針對特定叢集傳輸佇列的設定檔來檢查存取權。如果您想要套用更精細的安全檢查，請參閱第 135 頁的『存取控制及多個叢集傳輸佇列』。

多個叢集傳送端通道及多個傳輸佇列

在叢集傳送端通道上轉遞訊息之前，佇列管理程式會將訊息儲存在叢集傳輸佇列中。它會選取連接至訊息目的地的叢集傳送端通道。它可以選擇所有連接至相同目的地的叢集傳送端通道。目的地可能是相同的實體佇列，由多個叢集傳送端通道連接至單一佇列管理程式。目的地也可能是許多相同佇列名稱的實體佇列，在相同叢集中的不同佇列管理程式上管理。其中，選擇連接至目的地的叢集傳送端通道，工作量平衡演算法會選擇一個。此選項取決於許多因素；請參閱 [叢集工作量管理演算法](#)。

在第 147 頁的圖 25 中，`CL1.QM1`、`CL1.QM2` 和 `CS.QM1` 都是可能通往相同目的地的所有通道。比方說，如果您在 `QM1` 和 `QM2` 上的 `CL1` 中定義 `Q1`，則 `CL1.QM1` 和 `CL1.QM2` 都會提供通往兩個不同佇列管理程式上相同目的地 `Q1` 的路徑。如果通道 `CS.QM1` 也位於 `CL1` 中，則它也是 `Q1` 訊息可以採用的通道。`CS.QM1` 的叢集成員資格可能由叢集名單定義，這就是通道名稱未在其建構中包含叢集名稱的原因。視工作量平衡參數及傳送端應用程式而定，`Q1` 的部分訊息可能會放置在每一個傳輸佇列 `XMITQ.CL1.QM1`、`XMITQ.CL1` 及 `SYSTEM.CLUSTER.TRANSMIT.CS.QM1` 上。

如果您想要分隔訊息資料流量，以便相同目的地的訊息不會與不同目的地的訊息共用佇列或通道，則必須考量如何先將資料流量劃分到不同的叢集傳送端通道，然後如何將特定通道的訊息分隔到不同的傳輸佇列。相同叢集、相同佇列管理程式上的叢集佇列通常會共用相同的叢集通道。單獨定義多個叢集傳輸佇列不足以將叢集訊息資料流量分隔到不同的佇列。除非您將不同目的地佇列的訊息分隔到不同的通道，否則這些訊息會共用相同的叢集傳輸佇列。

分隔訊息所採用通道的直接明確方法是建立多個叢集。在每一個叢集中的任何佇列管理程式上，只定義一個叢集佇列。然後，如果您為每一個叢集/佇列管理程式組合定義不同的叢集接收端通道，則每一個叢集佇列的訊息不會與其他叢集佇列的訊息共用叢集通道。如果您為叢集通道定義個別傳輸佇列，則傳送端佇列管理程式只會將訊息儲存在每一個傳輸佇列上儲存一個叢集佇列的訊息。比方說，如果您想要兩個叢集佇列不共用資源，您可以將它們放在相同佇列管理程式的不同叢集中，或放在相同叢集的不同佇列管理程式中。

選擇叢集傳輸佇列不會影響工作量平衡演算法。工作量平衡演算法會選擇要轉遞訊息的叢集傳送端通道。它會將訊息放在該通道所處理的傳輸佇列上。如果再次呼叫工作量平衡演算法來選擇，例如，如果通道停止，它可能可以選取不同的通道來轉遞訊息。如果它選擇不同的通道，且新通道會從不同的叢集傳輸佇列轉遞訊息，則工作量平衡演算法會將訊息傳送至另一個傳輸佇列。

在第 147 頁的圖 25 中，兩個叢集傳送端通道 `CS.QM1` 和 `CS.QM2` 與預設系統傳輸佇列相關聯。當工作量平衡演算法將訊息儲存在 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 或任何其他叢集傳輸佇列時，要轉遞訊息的叢集傳送端通道名稱會儲存在訊息的相關性 ID 中。每一個通道只會轉遞那些符合具有通道名稱之相關性 ID 的訊息。

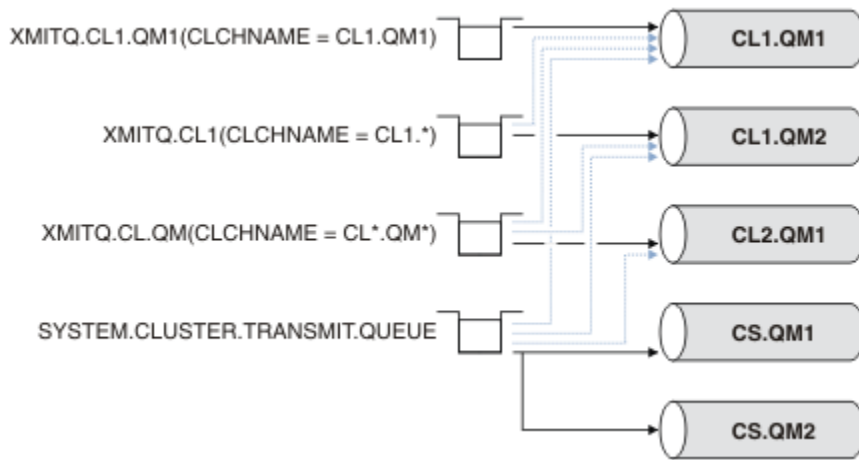


圖 25: 多個叢集傳送端通道

如果 CS.QM1 停止，則會檢查該叢集傳送端通道的傳輸佇列上的訊息。工作量平衡演算法會重新處理可由另一個通道轉遞的那些訊息。其相關性 ID 會重設為替代叢集傳送端通道名稱。如果替代叢集傳送端通道是 CS.QM2，則訊息會保留在 SYSTEM.CLUSTER.TRANSMIT.QUEUE 上。如果替代通道為 CL1.QM1，則工作量平衡演算法會將訊息傳送至 XMITQ.CL1.QM1。當叢集傳送端通道重新啟動時，通道會重新傳送新訊息，以及未針對不同叢集傳送端通道標示的訊息。

您可以在執行中系統上變更傳輸佇列與叢集傳送端通道之間的關聯。您可以變更傳輸佇列上的 CLCHNAME 參數，或變更 DEFCLXQ 佇列管理程式參數。當受變更影響的通道重新啟動時，它會啟動傳輸佇列切換處理程序；請參閱第 145 頁的『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』。

當通道重新啟動時，會啟動切換傳輸佇列的處理程序。當通道停止時，工作量重新平衡處理程序即會啟動。兩個處理程序可以平行執行。

簡單的情況是當停止叢集傳送端通道時，不會導致重新平衡處理程序變更要轉遞佇列上任何訊息的叢集傳送端通道。在這種情況下，沒有其他叢集傳送端通道可以將訊息轉遞至正確的目的地。由於沒有替代的叢集傳送端通道可將訊息轉遞至其目的地，在叢集傳送端通道停止之後，仍會針對相同的叢集傳送端通道標示訊息。當通道啟動時，如果交換器在擱置中，則切換處理程序會將訊息移至不同的傳輸佇列，由相同的叢集傳送端通道處理它們。

更複雜的情況是多個叢集傳送端通道可以處理部分訊息至相同目的地。您可以停止並重新啟動叢集傳送端通道，以觸發傳輸佇列切換。在許多情況下，當您重新啟動通道時，工作量平衡演算法已將訊息從原始傳輸佇列移至不同叢集傳送端通道所提供的不同傳輸佇列。只有那些無法由不同叢集傳送端通道轉遞的訊息，才會繼續傳送至新的傳輸佇列。在某些情況下，如果通道快速重新啟動，部分可由工作量平衡演算法傳送的訊息仍會保留。在此情況下，部分剩餘訊息會由工作量平衡處理程序切換，部分則由傳輸佇列切換處理程序切換。

相關概念

[第 238 頁的『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。

[第 337 頁的『計算日誌的大小』](#)

預估佇列管理程式所需的日誌大小。

相關工作

[第 241 頁的『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。配置多個叢集傳輸佇列適用於 z/OS 以外的平台。

[第 182 頁的『使用閘道佇列管理程式建立兩個重疊的叢集』](#)

遵循作業中的指示，使用閘道佇列管理程式來建構重疊叢集。使用叢集作為下列範例的起點，將訊息從一個應用程式隔離至叢集中的其他應用程式。

[第 165 頁的『將佇列管理程式新增至叢集: 個別傳輸佇列』](#)

請遵循下列指示，將佇列管理程式新增至您建立的叢集。叢集佇列和主題的訊息會使用多個叢集傳輸佇列來傳送。

第 170 頁的『[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](#)』

修改使用閘道佇列管理程式之重疊叢集的配置。從閘道佇列管理程式將修改訊息傳送至應用程式之後，不會使用與其他叢集訊息相同的傳輸佇列或通道。將修改訊息傳送至應用程式之後，解決方案會使用額外的叢集傳輸佇列，將訊息資料流量分隔至叢集中的單一佇列管理程式。

第 173 頁的『[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](#)』

修改使用閘道佇列管理程式之重疊叢集的配置。從閘道佇列管理程式將修改訊息傳送至應用程式之後，不會使用與其他叢集訊息相同的傳輸佇列或通道。將修改訊息傳送至應用程式之後，解決方案會使用其他叢集，將訊息與特定叢集佇列隔離。

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。

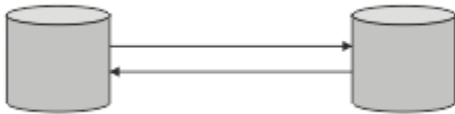


圖 26: 兩個已連接的完整儲存庫。

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。

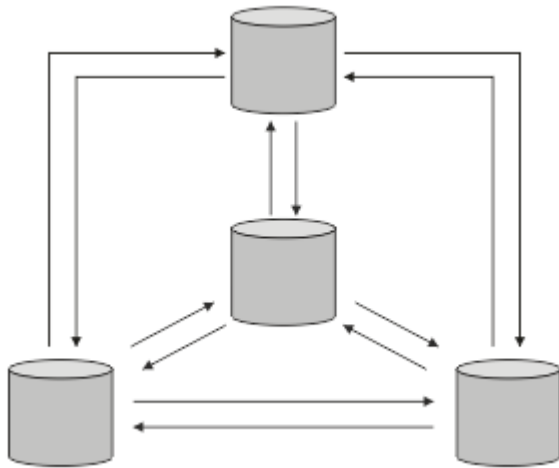


圖 27: 超過兩個已連接的完整儲存庫

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。

命名 MQ 叢集時的最佳作法

雖然叢集名稱最多可以有 48 個字元，但在將命名慣例套用至其他物件時，相對較短的叢集名稱會很有用。請參閱第 150 頁的『選擇叢集通道名稱時的最佳作法』。

選擇叢集名稱時，通常會有助於代表叢集的「目的」(可能長期存在)，而不是「內容」。例如 'B2BPROD' 或 'ACTTEST'，而不是 'QM1_QM2_QM3_CLUS'。

選擇叢集佇列管理程式名稱時的最佳作法

如果您是從頭開始建立新的叢集及其成員，請考量佇列管理程式的命名慣例，以反映其叢集使用情形。每個佇列管理程式都必須有不同的名稱。不過，您可以為叢集中的佇列管理程式提供一組類似名稱，以協助識別及記住邏輯分組 (例如 'ACTTQM1, ACTTQM2)。

如果您選擇使用下一節中說明的慣例，或類似的通道名稱，則相對較短的佇列管理程式名稱 (例如少於 8 個字元) 會有幫助。

選擇叢集通道名稱時的最佳作法

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，所以在第一次命名物件時請小心，以避免在中途透過專案變更命名慣例 (請參閱前一節)。

在定義通道時，請記住叢集中任何佇列管理程式上自動建立的叢集傳送端通道，會從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中取得其名稱，因此這些通道必須是唯一的，在叢集中遠端佇列管理程式上才有意義。

一種常見的方法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，您可以延伸此慣例。例如：

- CLUSTER1.QM1.S1
- CLUSTER1.QM1.N3
- CLUSTER1.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，T4 可能是使用 IPV4 網路的 TCP IP。

命名共用通道定義

單一通道定義可以在多個叢集之間共用，在此情況下，這裡建議的命名慣例需要修改。不過，如 [管理通道定義](#) 中所述，在任何情況下通常最好為每個叢集定義離散通道。

較舊的通道命名慣例

在叢集環境之外，使用 'FROMQM.TO.TARGETQM' 命名慣例已在歷史上很常見，因此您可能會發現現有叢集已使用類似的名稱 (例如 CLUSTER.TO.TARGET)。這不建議作為新叢集命名方法的一部分，因為它會進一步減少可用字元，以在通道名稱內傳送「有用」資訊。

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 151 頁的圖 28](#) 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。WebSphere MQ 會查看目的地佇列名稱和佇列管理程式名稱，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 151 頁的圖 28](#) 中所示。WebSphere 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。

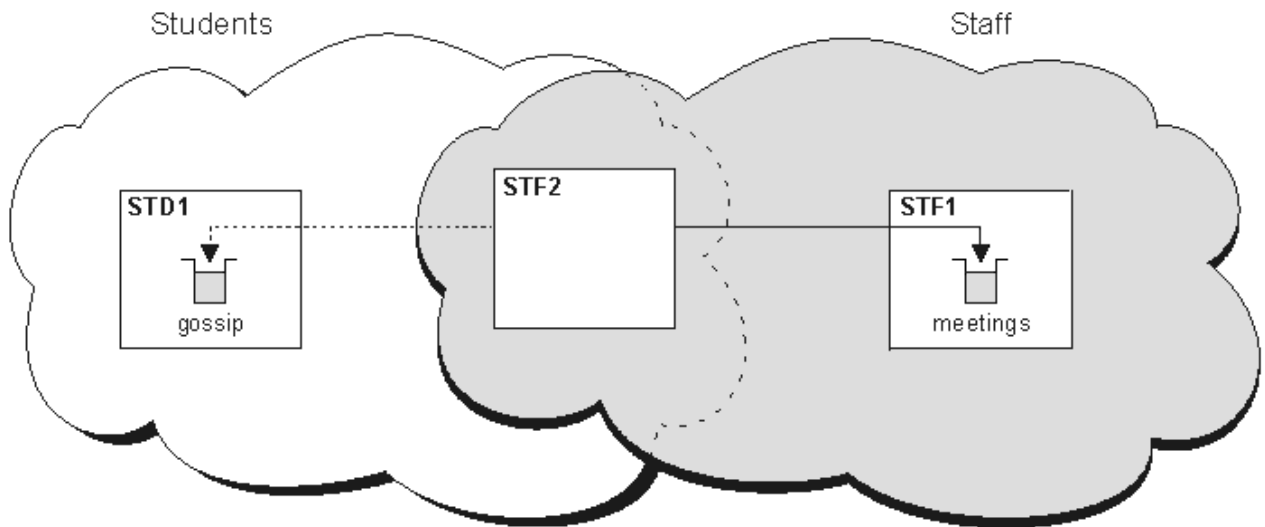


圖 28: 服務類別

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。

- 「WebSphere MQ 探險家」無法直接管理 WebSphere MQ for z/OS 早於 6.0 版的佇列管理程式。
- 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，則可能有耗用過多網路資源的風險。
- 如果您使用「WebSphere MQ 探險家」（以樹狀結構呈現佇列管理程式），則大型叢集的視圖可能會很麻煩。
- WebSphere MQ 探險家可以在 WebSphere 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 WebSphere MQ，「WebSphere 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「WebSphere MQ 探險家」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。WebSphere 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 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 WebSphere 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

叢集

叢集運作方式

第 135 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。

第 137 頁的『叢集的元件』

叢集由佇列管理程式、叢集儲存庫、叢集通道及叢集佇列組成。

第 154 頁的『管理 IBM WebSphere MQ 叢集』

您可以建立、延伸及維護 IBM WebSphere MQ 叢集。

相關工作

第 133 頁的『配置佇列管理程式叢集』

請使用本主題中的鏈結，以瞭解叢集如何運作、如何設計叢集配置，以及取得如何設定簡式叢集的範例。

第 154 頁的『設定新的叢集』

請遵循下列指示來設定範例叢集。個別指示說明在 TCP/IP、LU 6.2 上設定叢集，以及使用單一傳輸佇列或多個傳輸佇列。測試叢集的運作方式是將訊息從一個佇列管理程式傳送至另一個佇列管理程式。

在叢集中建立通訊

當有訊息要遞送時，需要通道起始程式來啟動通訊通道。通道接聽器會等待啟動通道的另一端來接收訊息。

開始之前

若要在叢集中的佇列管理程式之間建立通訊，請使用其中一個支援的通訊協定來配置鏈結。支援的通訊協定是任何平台上的 TCP 或 LU 6.2，以及 Windows 系統上的 NetBIOS 或 SPX。在此配置中，您也需要通道起始程式和通道接聽器，就像使用分散式佇列一樣。

關於這項作業

所有叢集佇列管理程式都需要通道起始程式，才能監視系統定義的起始佇列 SYSTEM.CHANNEL.INITQ。SYSTEM.CHANNEL.INITQ 是所有傳輸佇列 (包括叢集傳輸佇列) 的起始佇列。

每一個佇列管理程式都必須有一個通道接聽器。通道接聽器程式會等待送入的網路要求，並在需要時啟動適當的接收端通道。頻道接聽器的實作是平台專用的，不過有一些一般特性。在所有 WebSphere MQ 平台上，可以使用 START LISTENER 指令來啟動接聽器。在 WebSphere MQ for IBM i、Windows、UNIX and Linux 系統上，您可以與佇列管理程式同時自動啟動接聽器。若要自動啟動接聽器，請將 LISTENER 物件的 CONTROL 屬性設為 QMGR 或 STARTONLY。

程序

1. 啟動通道起始程式。



IBM WebSphere MQ 適用於 Windows, UNIX and Linux 系統

當您啟動佇列管理程式時，如果佇列管理程式屬性 SCHINIT 設為 QMGR，則會自動啟動通道起始程式。否則，可以使用 **runmqsc** START CHINIT 指令或 **runmqchi** 控制指令來啟動它。

2. 啟動通道接聽器。

Windows

IBM WebSphere MQ 適用於 Windows

請使用 WebSphere MQ 所提供的通道接聽器程式，或使用作業系統所提供的機能。

如果要啟動 WebSphere MQ 通道接聽器，請使用 RUNMQLSR 指令。例如：

```
RUNMQLSR -t tcp -p 1414 -m QM1
```

UNIX Linux

UNIX and Linux 系統上的 IBM WebSphere MQ

請使用 WebSphere MQ 所提供的通道接聽器程式，或使用作業系統所提供的機能；例如 **inetd** 來進行 TCP 通訊。

如果要啟動 WebSphere MQ 通道接聽器，請使用 **runmqclsr** 指令。例如：

```
runmqclsr -t tcp -p 1414 -m QM1
```

若要使用 **inetd** 來啟動通道，請配置兩個檔案：

- 編輯檔案 `/etc/services`。您必須以超級使用者或 `root` 身分登入。如果下列行不在檔案中，請如下所示新增：

```
MQSeries      1414/tcp      # Websphere MQ channel listener
```

其中 1414 是 IBM WebSphere MQ 所需的埠號。您可以變更埠號，但它必須符合傳送端指定的埠號。

- 編輯檔案 `/etc/inetd.conf`。如果您在該檔案中沒有下列這一行，請如下所示新增它：

```
MQSeries stream tcp nowait mqm MQ_INSTALLATION_PATH/bin/amqcrsta amqcrsta  
-m queue.manager.name
```

其中 `MQ_INSTALLATION_PATH` 取代為 WebSphere MQ 安裝所在的高階目錄。

在 **inetd** 重新讀取配置檔之後，更新項目會變成作用中。從 `root` 使用者 ID 發出下列指令：

在 AIX 上：

```
refresh -s inetd
```

在 HP-UX 上：

```
inetd -c
```

在 Solaris 或 Linux 上：

- 使用下列指令尋找 **inetd** 的程序 ID：

```
ps -ef | grep inetd
```

- 執行適當的指令，如下所示：

- 若為 Solaris 9 及 Linux：

```
kill -1 inetd processid
```

- 若為 Solaris 10 或更新版本：

```
inetconv
```


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 WebSphere 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。請參閱第 257 頁的『在大型叢集中重新整理可能會影響叢集的效能及可用性』。

相關概念

第 257 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。

管理 IBM WebSphere MQ 叢集

您可以建立、延伸及維護 IBM WebSphere MQ 叢集。

如需如何管理 IBM WebSphere MQ 叢集的詳細資料，請參閱下列子主題：

相關概念

叢集

叢集運作方式

第 135 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。

第 137 頁的『叢集的元件』

叢集由佇列管理程式、叢集儲存庫、叢集通道及叢集佇列組成。

相關工作

第 133 頁的『配置佇列管理程式叢集』

請使用本主題中的鏈結，以瞭解叢集如何運作、如何設計叢集配置，以及取得如何設定簡式叢集的範例。

第 154 頁的『設定新的叢集』

請遵循下列指示來設定範例叢集。個別指示說明在 TCP/IP、LU 6.2 上設定叢集，以及使用單一傳輸佇列或多個傳輸佇列。測試叢集的運作方式是將訊息從一個佇列管理程式傳送至另一個佇列管理程式。

設定新的叢集

請遵循下列指示來設定範例叢集。個別指示說明在 TCP/IP、LU 6.2 上設定叢集，以及使用單一傳輸佇列或多個傳輸佇列。測試叢集的運作方式是將訊息從一個佇列管理程式傳送至另一個佇列管理程式。

開始之前

- 無需遵循這些指示，您可以使用 IBM WebSphere MQ Explorer 隨附的其中一個精靈來建立叢集，就像此作業所建立的叢集一樣。用滑鼠右鍵按一下「佇列管理程式叢集」資料夾，然後按一下 **新建 > 佇列管理程式叢集**，並遵循精靈中提供的指示。

- 如需背景資訊，可協助您瞭解設定叢集所採取的步驟，請參閱 [第 139 頁的『叢集佇列數』](#)、[通道](#) 及 [接聽器](#)。

關於這項作業

您正在為鏈商店設定新的 IBM WebSphere MQ 網路。該商店有兩個分店，一個在倫敦，另一個在紐約。每一個商店的資料和應用程式是由執行個別佇列管理程式的系統所管理。這兩個佇列管理程式稱為 LONDON 和 NEWYORK。庫存應用程式在紐約的系統上執行，並連接至佇列管理程式 NEWYORK。應用程式由訊息到達 INVENTQ 佇列 (由 NEWYORK 管理) 所驅動。這兩個佇列管理程式 LONDON 和 NEWYORK 將在稱為 INVENTORY 的叢集中鏈結，以便它們都可以將訊息放置到 INVENTQ。

第 155 頁的圖 29 顯示此叢集的外觀。

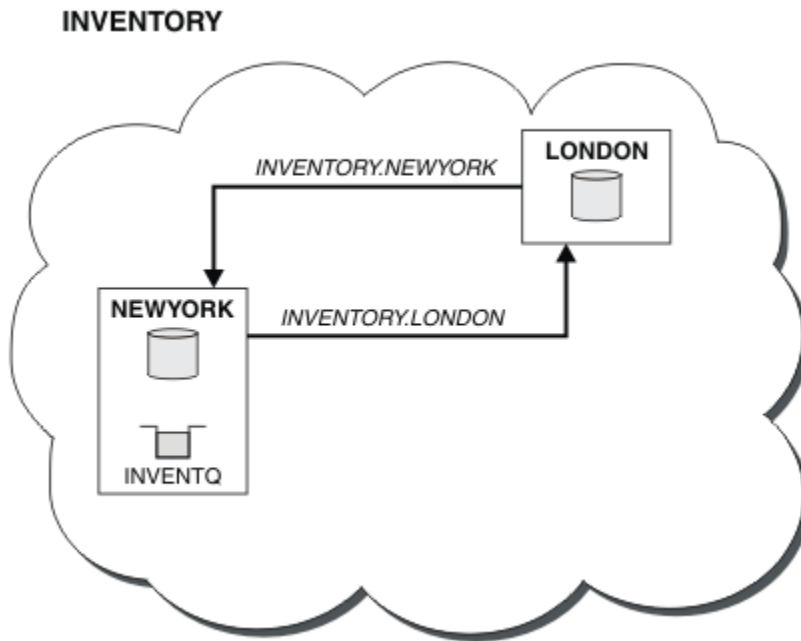


圖 29: 具有兩個佇列管理程式的 *INVENTORY* 叢集

您可以在不在 z/OS 上的叢集中配置每一個佇列管理程式，以使用不同的叢集傳輸佇列，將訊息傳送至叢集中的其他佇列管理程式。

設定叢集的指示因傳輸通訊協定、傳輸佇列數或平台而異。您可以選擇三種組合。所有組合的驗證程序維持相同。

程序

- [第 156 頁的『使用 TCP/IP 搭配每個佇列管理程式的單一傳輸佇列來設定叢集』](#)
- [第 158 頁的『在 TCP/IP 上使用每個佇列管理程式多個傳輸佇列來設定叢集』](#)
- [第 160 頁的『在 z/OS 上使用 LU 6.2 設定叢集』](#)
- [第 162 頁的『驗證叢集』](#)

結果

第 155 頁的圖 29 顯示此作業的 *INVENTORY* 叢集設定。

顯然，*INVENTORY* 是一個小型叢集。然而，它作為概念的證明是有用的。瞭解此叢集的重要事項是它為未來加強功能提供的範圍。

相關概念

[叢集](#)

叢集運作方式

第 135 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。

第 137 頁的『叢集的元件』
叢集由佇列管理程式、叢集儲存庫、叢集通道及叢集佇列組成。

第 154 頁的『管理 IBM WebSphere MQ 叢集』
您可以建立、延伸及維護 IBM WebSphere MQ 叢集。

相關工作

第 133 頁的『配置佇列管理程式叢集』
請使用本主題中的鏈結，以瞭解叢集如何運作、如何設計叢集配置，以及取得如何設定簡式叢集的範例。

使用 TCP/IP 搭配每個佇列管理程式的單一傳輸佇列來設定叢集

開始之前

- 在 AIX, HP-UX, IBM i, Linux, Solaris, and Windows 上，佇列管理程式屬性 **DEFCLXQ** 必須保留為其預設值 SCTQ。

關於這項作業

遵循下列步驟，使用傳輸通訊協定 TCP/IP 在 AIX, HP-UX, IBM i, Linux, Solaris, and Windows 上設定叢集。

程序

1. 決定叢集的組織及其名稱。

您決定將兩個佇列管理程式 LONDON 及 NEWYORK 鏈結至叢集。只有兩個佇列管理程式的叢集只會在使用分散式佇列的網路上提供邊際好處。這是一個好的開始，為未來的擴展提供了空間。當您開啟儲存庫的新分支時，您可以輕鬆地將新的佇列管理程式新增至叢集。新增佇列管理程式不會中斷現有網路；請參閱第 164 頁的『將佇列管理程式加入叢集中』。

目前，您執行的唯一應用程式是庫存應用程式。叢集名稱為 INVENTORY。

2. 決定要保留完整儲存庫的佇列管理程式。

在任何叢集中，您必須至少指定一個佇列管理程式，或最好指定兩個佇列管理程式，以保留完整儲存庫。在此範例中，只有兩個佇列管理程式 LONDON 和 NEWYORK，兩者都保留完整儲存庫。

- a. 您可以依任何順序執行其餘步驟。
- b. 當您繼續執行這些步驟時，可能會將警告訊息寫入佇列管理程式日誌。這些訊息是您尚未新增之遺漏定義的結果。

Examples of the responses to the commands are shown in a box like this after each step in this task. These examples show the responses returned by WebSphere MQ for AIX. The responses vary on other platforms.

- c. 在繼續執行這些步驟之前，請確定已啟動佇列管理程式。
3. 變更佇列管理程式定義以新增儲存庫定義。

在每一個要保留完整儲存庫的佇列管理程式上，使用 ALTER QMGR 指令並指定 REPOS 屬性，來變更本端佇列管理程式定義：

```
ALTER QMGR REPOS(INVENTORY)
```

```
1 : ALTER QMGR REPOS(INVENTORY)
AMQ8005: Websphere MQ queue manager changed.
```

例如，如果您輸入：

- a. runmqsc LONDON
- b. ALTER QMGR REPOS(INVENTORY)

LONDON 已變更為完整儲存庫。

4. 定義接聽器。

定義接聽器，以針對叢集中每個佇列管理程式接受來自其他佇列管理程式的網路要求。在 LONDON 佇列管理程式上，發出下列指令：

```
DEFINE LISTENER(LONDON_LS) TRPTYPE(TCP) CONTROL(QMGR)
```

註：當您定義接聽器時，如果您在 CONNAME 欄位中使用 IP 位址，且埠號不是預設埠 (1414)，則應該定義埠號。例如：

```
DEFINE LISTENER(LONDON_LS) TRPTYPE(TCP) CONTROL(QMGR) PORT(1415)
```

CONTROL 屬性可確保當佇列管理程式啟動時，接聽器會停止。

接聽器在定義時不會啟動，因此第一次必須使用下列 MQSC 指令手動啟動它：

```
START LISTENER(LONDON_LS)
```

對叢集中的所有其他佇列管理程式發出類似指令，並變更每一個佇列管理程式的接聽器名稱。

有數種方法可定義這些接聽器，如 接聽器 中所示。

5. 定義 LONDON 佇列管理程式的 CLUSRCVR 通道。

在叢集中的每個佇列管理程式上，定義佇列管理程式可以在其中接收訊息的叢集接收端通道。CLUSRCVR 定義佇列管理程式的連線名稱。連線名稱儲存在儲存庫中，其他佇列管理程式可以在其中參照它。CLUSTER 關鍵字顯示佇列管理程式從叢集中其他佇列管理程式接收訊息的可用性。

在此範例中，通道名稱為 INVENTORY.LONDON，連線名稱 (CONNAME) 是佇列管理程式所在機器的網址，即 LONDON.CHSTORE.COM。網址可以輸入為英數 DNS 主機名稱，或 IPv4 帶點十進位格式的 IP 位址。例如，192.0.2.0 或 IPv6 十六進位格式；例如 2001:DB8:0204:acff:fe97:2c34:fde0:3485。未指定埠號，因此使用預設埠 (1414)。

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)  
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)  
DESCR('TCP Cluster-receiver channel for queue manager LONDON')
```

```
1 : DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)  
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)  
DESCR('TCP Cluster-receiver channel for queue manager LONDON')  
AMQ8014: Websphere MQ channel created.  
07/09/98 12:56:35 No repositories for cluster 'INVENTORY'
```

6. 定義 NEWYORK 佇列管理程式的 CLUSRCVR 通道。

如果通道接聽器使用預設埠 (通常是 1414)，且叢集在 z/OS 上不包含佇列管理程式，則您可以省略 CONNAME

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CLUSTER(INVENTORY)  
DESCR('TCP Cluster-receiver channel for queue manager NEWYORK')
```

7. 在 LONDON 佇列管理程式上定義 CLUSSDR 通道。

在叢集中的每個佇列管理程式上，定義一個叢集傳送端通道。佇列管理程式會將訊息傳送至叢集傳送端通道上的其中一個完整儲存庫佇列管理程式。在此情況下，只有兩個佇列管理程式，兩者都保留完整儲存庫。它們必須各自具有 CLUSSDR 定義，該定義指向在其他佇列管理程式中定義的 CLUSRCVR 通道。CLUSSDR 定義上提供的通道名稱必須符合對應 CLUSRCVR 定義上的通道名稱。一旦佇列管理程式在相同叢集中同時具有叢集接收端通道及叢集傳送端通道的定義，即會啟動叢集傳送端通道。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
```

```
1 : DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
AMQ8014: Websphere MQ channel created.
07/09/98 13:00:18 Channel program started.
```

8. 在 NEWYORK 佇列管理程式上定義 CLUSSDR 通道。

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from NEWYORK to repository at LONDON')
```

9. 定義叢集佇列 INVENTQ

指定 CLUSTER 關鍵字，在 NEWYORK 佇列管理程式上定義 INVENTQ 佇列。

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

```
1 : DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
AMQ8006: Websphere MQ queue created.
```

CLUSTER 關鍵字會將佇列通告給叢集。一旦定義佇列，即可供叢集中的其他佇列管理程式使用。他們可以將訊息傳送給它，而不需要為它建立遠端佇列定義。

所有定義都已完成。在所有平台上，在每一個佇列管理程式上啟動接聽器程式。接聽器程式會等待送入的網路要求，並在需要時啟動叢集接收端通道。

在 TCP/IP 上使用每個佇列管理程式多個傳輸佇列來設定叢集

關於這項作業

遵循下列步驟，使用傳輸通訊協定 TCP/IP 在 AIX, HP-UX, IBM i, Linux, Solaris, and Windows 上設定叢集。儲存庫佇列管理程式會配置成使用不同的叢集傳輸佇列來彼此傳送訊息，以及傳送訊息至叢集中的其他佇列管理程式。如果您將佇列管理程式新增至也要使用不同傳輸佇列的叢集，請遵循作業第 165 頁的『將佇列管理程式新增至叢集: 個別傳輸佇列』。您無法在 z/OS 上設定佇列管理程式來使用個別叢集傳輸佇列。

程序

1. 決定叢集的組織及其名稱。

您決定將兩個佇列管理程式 LONDON 及 NEWYORK 鏈結至叢集。只有兩個佇列管理程式的叢集只會在使用分散式佇列的網路上提供邊際好處。這是一個好的開始，為未來的擴展提供了空間。當您開啟儲存庫的新分支時，您可以輕鬆地將新的佇列管理程式新增至叢集。新增佇列管理程式不會中斷現有網路；請參閱第 164 頁的『將佇列管理程式加入叢集中』。

目前，您執行的唯一應用程式是庫存應用程式。叢集名為 INVENTORY。

2. 決定要保留完整儲存庫的佇列管理程式。

在任何叢集中，您必須至少指定一個佇列管理程式，或最好指定兩個佇列管理程式，以保留完整儲存庫。在此範例中，只有兩個佇列管理程式 LONDON 和 NEWYORK，兩者都保留完整儲存庫。

- a. 您可以依任何順序執行其餘步驟。
- b. 當您繼續執行這些步驟時，可能會將警告訊息寫入佇列管理程式日誌。這些訊息是您尚未新增之遺漏定義的結果。

Examples of the responses to the commands are shown in a box like this after each step in this task. These examples show the responses returned by WebSphere MQ for AIX. The responses vary on other platforms.

- c. 在繼續執行這些步驟之前，請確定已啟動佇列管理程式。
3. 變更佇列管理程式定義以新增儲存庫定義。

在每一個要保留完整儲存庫的佇列管理程式上，使用 ALTER QMGR 指令並指定 REPOS 屬性，來變更本端佇列管理程式定義：

```
ALTER QMGR REPOS(INVENTORY)
```

```
1 : ALTER QMGR REPOS(INVENTORY)
AMQ8005: Websphere MQ queue manager changed.
```

例如，如果您輸入：

- a. runmqsc LONDON
- b. ALTER QMGR REPOS(INVENTORY)

LONDON 已變更為完整儲存庫。

4. 變更佇列管理程式定義，以針對每一個目的地建立個別叢集傳輸佇列。

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

在您新增至叢集的每一個佇列管理程式上，決定是否使用個別傳輸佇列。請參閱主題 [第 164 頁的『將佇列管理程式加入叢集中』](#) 和 [第 165 頁的『將佇列管理程式新增至叢集: 個別傳輸佇列』](#)。

5. 定義接聽器。

定義接聽器，以針對叢集中每個佇列管理程式接受來自其他佇列管理程式的網路要求。在 LONDON 佇列管理程式上，發出下列指令：

```
DEFINE LISTENER(LONDON_LS) TRPTYPE(TCP) CONTROL(QMGR)
```

註：當您定義接聽器時，如果您在 CONNAME 欄位中使用 IP 位址，且埠號不是預設埠 (1414)，則應該定義埠號。例如：

```
DEFINE LISTENER(LONDON_LS) TRPTYPE(TCP) CONTROL(QMGR) PORT(1415)
```

CONTROL 屬性可確保當佇列管理程式啟動時，接聽器會停止。

接聽器在定義時不會啟動，因此第一次必須使用下列 MQSC 指令手動啟動它：

```
START LISTENER(LONDON_LS)
```

對叢集中的所有其他佇列管理程式發出類似指令，並變更每一個佇列管理程式的接聽器名稱。

有數種方法可定義這些接聽器，如 [接聽器](#) 中所示。

6. 定義 LONDON 佇列管理程式的 CLUSRCVR 通道。

在叢集中的每個佇列管理程式上，定義佇列管理程式可以在其中接收訊息的叢集接收端通道。CLUSRCVR 定義佇列管理程式的連線名稱。連線名稱儲存在儲存庫中，其他佇列管理程式可以在其中參照它。CLUSTER 關鍵字顯示佇列管理程式從叢集中其他佇列管理程式接收訊息的可用性。

在此範例中，通道名稱為 INVENTORY.LONDON，連線名稱 (CONNAME) 是佇列管理程式所在機器的網址，即 LONDON.CHSTORE.COM。網址可以輸入為英數 DNS 主機名稱，或 IPv4 帶點十進位格式的 IP 位址。例如，192.0.2.0 或 IPv6 十六進位格式；例如 2001:DB8:0204:acff:fe97:2c34:fde0:3485。未指定埠號，因此使用預設埠 (1414)。

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager LONDON')
```

```
1 : DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager LONDON')
AMQ8014: Websphere MQ channel created.
07/09/98 12:56:35 No repositories for cluster 'INVENTORY'
```

7. 定義 NEWYORK 佇列管理程式的 CLUSRCVR 通道。

如果通道接聽器使用預設埠 (通常是 1414)，且叢集在 z/OS 上不包含佇列管理程式，則您可以省略 CONNNAME

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CLUSTER(INVENTORY)
DESCR('TCP Cluster-receiver channel for queue manager NEWYORK')
```

8. 在 LONDON 佇列管理程式上定義 CLUSSDR 通道。

在叢集中的每個佇列管理程式上，定義一個叢集傳送端通道。佇列管理程式會將訊息傳送至叢集傳送端通道上的其中一個完整儲存庫佇列管理程式。在此情況下，只有兩個佇列管理程式，兩者都保留完整儲存庫。它們必須各自具有 CLUSSDR 定義，該定義指向在其他佇列管理程式中定義的 CLUSRCVR 通道。CLUSSDR 定義上提供的通道名稱必須符合對應 CLUSRCVR 定義上的通道名稱。一旦佇列管理程式在相同叢集中同時具有叢集接收端通道及叢集傳送端通道的定義，即會啟動叢集傳送端通道。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
```

```
1 : DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from LONDON to repository at NEWYORK')
AMQ8014: Websphere MQ channel created.
07/09/98 13:00:18 Channel program started.
```

9. 在 NEWYORK 佇列管理程式上定義 CLUSSDR 通道。

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('TCP Cluster-sender channel from NEWYORK to repository at LONDON')
```

10. 定義叢集佇列 INVENTQ

指定 CLUSTER 關鍵字，在 NEWYORK 佇列管理程式上定義 INVENTQ 佇列。

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

```
1 : DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
AMQ8006: Websphere MQ queue created.
```

CLUSTER 關鍵字會將佇列通告給叢集。一旦定義佇列，即可供叢集中的其他佇列管理程式使用。他們可以將訊息傳送給它，而不需要為它建立遠端佇列定義。

所有定義都已完成。在所有平台上，在每一個佇列管理程式上啟動接聽器程式。接聽器程式會等待送入的網路要求，並在需要時啟動叢集接收端通道。

在 z/OS 上使用 LU 6.2 設定叢集

程序

1. 決定叢集的組織及其名稱。

您決定將兩個佇列管理程式 LONDON 及 NEWYORK 鏈結至叢集。只有兩個佇列管理程式的叢集只會在使用分散式佇列的網路上提供邊際好處。這是一個好的開始，為未來的擴展提供了空間。當您開啟儲存庫的新分支時，您可以輕鬆地將新的佇列管理程式新增至叢集。新增佇列管理程式不會中斷現有網路；請參閱第 164 頁的『將佇列管理程式加入叢集中』。

目前，您執行的唯一應用程式是庫存應用程式。叢集名稱為 INVENTORY。

2. 決定要保留完整儲存庫的佇列管理程式。

在任何叢集中，您必須至少指定一個佇列管理程式，或最好指定兩個佇列管理程式，以保留完整儲存庫。在此範例中，只有兩個佇列管理程式 LONDON 和 NEWYORK，兩者都保留完整儲存庫。

- a. 您可以依任何順序執行其餘步驟。
- b. 當您繼續執行這些步驟時，可能會在 z/OS 系統主控台中寫入警告訊息。這些訊息是您尚未新增之遺漏定義的結果。
- c. 在繼續執行這些步驟之前，請確定已啟動佇列管理程式。

3. 變更佇列管理程式定義以新增儲存庫定義。

在每一個要保留完整儲存庫的佇列管理程式上，使用 ALTER QMGR 指令並指定 REPOS 屬性，來變更本端佇列管理程式定義：

```
ALTER QMGR REPOS(INVENTORY)
```

```
1 : ALTER QMGR REPOS(INVENTORY)
AMQ8005: Websphere MQ queue manager changed.
```

例如，如果您輸入：

- a. runmqsc LONDON
- b. ALTER QMGR REPOS(INVENTORY)

LONDON 已變更為完整儲存庫。

4. 定義接聽器。

接聽器在定義時不會啟動，因此第一次必須使用下列 MQSC 指令手動啟動它：

```
START LISTENER(LONDON_LS)
```

對叢集中的所有其他佇列管理程式發出類似指令，並變更每一個佇列管理程式的接聽器名稱。

5. 定義 LONDON 佇列管理程式的 CLUSRCVR 通道。

在叢集中的每個佇列管理程式上，定義佇列管理程式可以在其中接收訊息的叢集接收端通道。CLUSRCVR 定義佇列管理程式的連線名稱。連線名稱儲存在儲存庫中，其他佇列管理程式可以在其中參照它。CLUSTER 關鍵字顯示佇列管理程式從叢集中其他佇列管理程式接收訊息的可用性。

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(LU62)
CONNAME(LONDON.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-receiver channel for queue manager LONDON')
```

```
1 : DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) TRPTYPE(LU62)
CONNAME(LONDON.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-receiver channel for queue manager LONDON')
AMQ8014: Websphere MQ channel created.
07/09/98 12:56:35 No repositories for cluster 'INVENTORY'
```

6. 定義 NEWYORK 佇列管理程式的 CLUSRCVR 通道。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSRCVR) TRPTYPE(LU62)
CONNAME(NEWYORK.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-receiver channel for queue manager NEWYORK')
```

7. 在 LONDON 佇列管理程式上定義 CLUSSDR 通道。

在叢集中的每個佇列管理程式上，定義一個叢集傳送端通道。佇列管理程式會將訊息傳送至叢集傳送端通道上的其中一個完整儲存庫佇列管理程式。在此情況下，只有兩個佇列管理程式，兩者都保留完整儲存庫。它們必須各自具有 CLUSSDR 定義，該定義指向在其他佇列管理程式中定義的 CLUSRCVR 通道。CLUSSDR 定義上提供的通道名稱必須符合對應 CLUSRCVR 定義上的通道名稱。一旦佇列管理程式在相同叢集中同時具有叢集接收端通道及叢集傳送端通道的定義，即會啟動叢集傳送端通道。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNAME(CPIC) CLUSTER(INVENTORY)
DESCR('LU62 Cluster-sender channel from LONDON to repository at NEWYORK')
```

```
1 : DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNAME(NEWYORK.LUNAME) CLUSTER(INVENTORY)
MODENAME('#INTER') TPNAME('MQSERIES')
DESCR('LU62 Cluster-sender channel from LONDON to repository at NEWYORK')
AMQ8014: Websphere MQ channel created.
07/09/98 13:00:18 Channel program started.
```

8. 在 NEWYORK 佇列管理程式上定義 CLUSSDR 通道。

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNAME(LONDON.LUNAME) CLUSTER(INVENTORY)
DESCR('LU62 Cluster-sender channel from NEWYORK to repository at LONDON')
```

9. 定義叢集佇列 INVENTQ

指定 CLUSTER 關鍵字，在 NEWYORK 佇列管理程式上定義 INVENTQ 佇列。

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

```
1 : DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
AMQ8006: Websphere MQ queue created.
```

CLUSTER 關鍵字會將佇列通告給叢集。一旦定義佇列，即可供叢集中的其他佇列管理程式使用。他們可以將訊息傳送給它，而不需要為它建立遠端佇列定義。

所有定義都已完成。在所有平台上，在每一個佇列管理程式上啟動接聽器程式。接聽器程式會等待送入的網路要求，並在需要時啟動叢集接收端通道。

驗證叢集

關於這項作業

您可以使用下列一或多種方式來驗證叢集：

1. 執行管理指令以顯示叢集和通道屬性。
2. 執行範例程式，以在叢集佇列上傳送及接收訊息。
3. 撰寫您自己的程式以將要求訊息傳送至叢集佇列，並將回應訊息回覆至非叢集回覆佇列。

程序

發出 DISPLAY **runmqsc** 指令以驗證叢集。

您看到的回應應該與後續步驟中的回應類似。

1. 從 NEWYORK 佇列管理程式中，執行 **DISPLAY CLUSQMGR** 指令：

```
dis clusqmgr(*)
```

```

1 : dis clusqmgr(*)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(NEWYORK)          CLUSTER(INVENTORY)
CHANNEL(INVENTORY.NEWYORK)
AMQ8441: Display Cluster Queue Manager details.
CLUSQMGR(LONDON)          CLUSTER(INVENTORY)
CHANNEL(INVENTORY.LONDON)

```

2. 從 NEWYORK 佇列管理程式中，執行 **DISPLAY CHANNEL STATUS** 指令：

```
dis chstatus(*)
```

```

1 : dis chstatus(*)
AMQ8417: Display Channel Status details.
CHANNEL(INVENTORY.NEWYORK)  XMITQ( )
CONNAME(192.0.2.0)         CURRENT
CHLTYPE(CLUSRCVR)         STATUS(RUNNING)
RQMNAME(LONDON)
AMQ8417: Display Channel Status details.
CHANNEL(INVENTORY.LONDON)  XMITQ(SYSTEM.CLUSTER.TRANSMIT.INVENTORY.LONDON)
CONNAME(192.0.2.1)         CURRENT
CHLTYPE(CLUSSDR)          STATUS(RUNNING)
RQMNAME(LONDON)

```

使用 **amqsput** 在兩個佇列管理程式之間傳送訊息。

3. 在 LONDON 上，執行 **amqsput INVENTQ LONDON** 指令。

鍵入一些訊息，後面接著空白行。

4. 在 NEWYORK 上，執行 **amqsget INVENTQ NEWYORK** 指令。

現在您會看到您在 LONDON 上輸入的訊息。在 15 秒之後，程式結束。

使用您自己的程式在兩個佇列管理程式之間傳送訊息。

在下列步驟中，LONDON 會將訊息放置到 INVENTQ (位於 NEWYORK)，並在其佇列上接收回覆 LONDON_reply。

5. 在 LONDON 上，將訊息放置到叢集佇列。
 - a) 定義稱為 LONDON_reply 的本端佇列。
 - b) 將 MQOPEN 選項設為 MQ00_OUTPUT。
 - c) 發出 MQOPEN 呼叫以開啟佇列 INVENTQ。
 - d) 將訊息描述子中的 *ReplyToQ* 名稱設為 LONDON_reply。
 - e) 發出 MQPUT 呼叫以放置訊息。
 - f) 確定訊息。
6. 在 NEWYORK 上，接收叢集佇列上的訊息，並將回覆放置到回覆佇列。
 - a) 將 MQOPEN 選項設為 MQ00_BROWSE。
 - b) 發出 MQOPEN 呼叫以開啟佇列 INVENTQ。
 - c) 發出 MQGET 呼叫，以從 INVENTQ 取得訊息。
 - d) 從訊息描述子擷取 *ReplyToQ* 名稱。
 - e) 將 *ReplyToQ* 名稱放置在物件描述子的 *ObjectName* 欄位中。
 - f) 將 MQOPEN 選項設為 MQ00_OUTPUT。
 - g) 發出 MQOPEN 呼叫，以在佇列管理程式 LONDON 中開啟 LONDON_reply。
 - h) 發出 MQPUT 呼叫，將訊息放置到 LONDON_reply。
7. 在 LONDON 上，收到回覆。
 - a) 將 MQOPEN 選項設為 MQ00_BROWSE。
 - b) 發出 MQOPEN 呼叫以開啟佇列 LONDON_reply。
 - c) 發出 MQGET 呼叫，以從 LONDON_reply 取得訊息。

將佇列管理程式加入叢集中

請遵循下列指示，將佇列管理程式新增至您建立的叢集。叢集佇列及主題的訊息會使用單一叢集傳輸佇列 `SYSTEM.CLUSTER.TRANSMIT.QUEUE` 來傳送。

開始之前

註: 如果要將叢集的變更延伸到整個叢集，至少必須一律有一個完整儲存庫可用。在啟動這項作業之前，請確定您的儲存庫可供使用。

測試情境：

- INVENTORY 叢集如第 154 頁的『設定新的叢集』中所述進行設定。它包含兩個佇列管理程式 LONDON 和 NEWYORK，這兩個佇列管理程式都保留完整儲存庫。
- 佇列管理程式 PARIS 由主要安裝架構所擁有。如果不是，您必須執行 `setmqenv` 指令，以針對 PARIS 所屬的安裝架構設定指令環境。
- 所有三個系統之間都存在 TCP 連線功能，且佇列管理程式配置有在佇列管理程式控制下啟動的 TCP 接聽器。

關於這項作業

1. 正在巴黎設定鏈結儲存庫的新分支，且您想要將稱為 PARIS 的佇列管理程式新增至叢集。
2. 佇列管理程式 PARIS 透過將訊息放置在 INVENTQ 佇列上，將庫存更新項目傳送至在紐約系統上執行的應用程式。

請遵循下列步驟，將佇列管理程式新增至叢集。

程序

1. 決定哪一個完整儲存庫 PARIS 會先參照。

叢集中的每個佇列管理程式都必須參照其中一個或其他完整儲存庫。它會從完整儲存庫收集叢集的相關資訊，因此會建置它自己的局部儲存庫。選擇任一儲存庫作為完整儲存庫。一旦將新的佇列管理程式新增至叢集，它也會立即瞭解其他儲存庫。佇列管理程式的變更相關資訊會直接傳送至兩個儲存庫。在此範例中，您將 PARIS 鏈結至佇列管理程式 LONDON(純粹基於地理原因)。

註: 在佇列管理程式 PARIS 啟動之後，以任何順序執行其餘步驟。

2. 在佇列管理程式 PARIS 上定義 CLUSRCVR 通道。

叢集中的每個佇列管理程式都必須定義叢集接收端通道，它可以在該通道上接收訊息。在 PARIS 上，定義：

```
DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(PARIS.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for queue manager PARIS')
```

叢集接收端通道會通告佇列管理程式的可用性，以接收來自叢集 INVENTORY 中其他佇列管理程式的訊息。對於叢集接收端通道 INVENTORY.PARIS 的傳送端，不需要在其他佇列管理程式上進行定義。其他定義會在需要時自動建立。

3. 在佇列管理程式 PARIS 上定義 CLUSSDR 通道。

叢集中的每個佇列管理程式都必須定義一個叢集傳送端通道，它可以在其中將訊息傳送至其起始完整儲存庫。

在 PARIS 上，針對名為 INVENTORY.LONDON 的通道，對網址為 LONDON.CHSTORE.COM 的佇列管理程式進行下列定義。

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from PARIS to repository at LONDON')
```

4. 選擇性的: 如果此佇列管理程式正在重新加入叢集，請完成一些額外步驟。

- a) 如果您要將佇列管理程式新增至先前已從相同叢集中移除的叢集，請檢查它現在是否顯示為叢集成員。如果沒有，請完成下列額外步驟：
- i) 在您要新增的佇列管理程式上發出 **REFRESH CLUSTER** 指令。此步驟會停止叢集通道，並為本端叢集快取提供一組全新的序號，確保在叢集其餘部分內保持最新。

```
REFRESH CLUSTER(INVENTORY) REPOS(YES)
```

註：對於大型叢集，使用 **REFRESH CLUSTER** 指令會干擾進行中的叢集，而此後每隔 27 天，當叢集物件自動將狀態更新傳送給所有相關的佇列管理程式時，會再次造成干擾。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。

- ii) 重新啟動 CLUSSDR 通道 (例如，使用 START CHANNEL 指令)。
 - iii) 重新啟動 CLUSRCVR 通道。
- b) 如果叢集是發佈/訂閱叢集，且重新加入的佇列管理程式有訂閱，請發出下列指令，以確保在叢集中正確同步化 Proxy 訂閱：

```
REFRESH QMGR TYPE(PROXYSUB)
```

結果

下圖顯示此作業所設定的叢集。

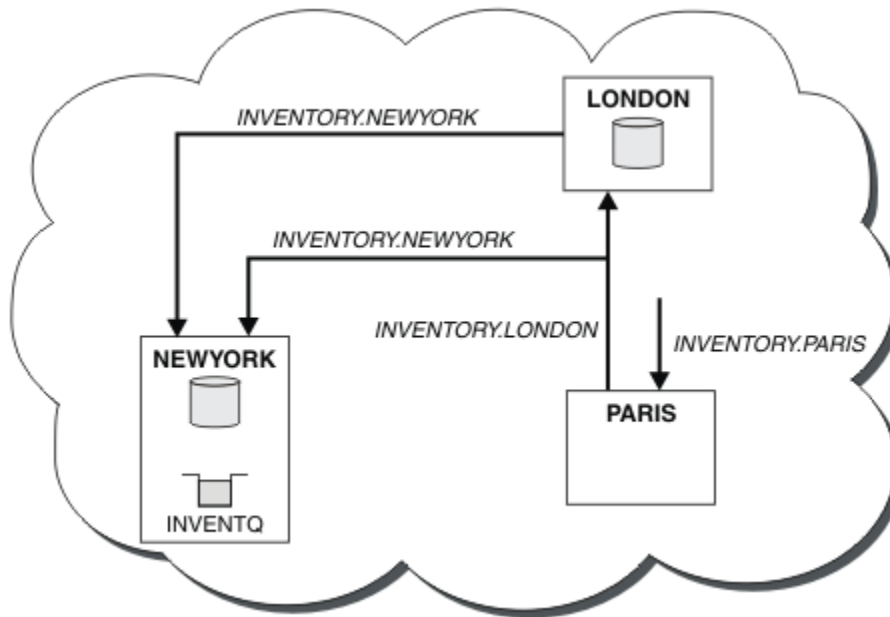


圖 30: 具有三個佇列管理程式的 INVENTORY 叢集

藉由只建立兩個定義: CLUSRCVR 定義和 CLUSSDR 定義，我們將佇列管理程式 PARIS 新增至叢集。

現在，PARIS 佇列管理程式會從 LONDON 的完整儲存庫中，得知 INVENTQ 佇列是由佇列管理程式管理 NEWYORK。當巴黎系統所管理的應用程式嘗試將訊息放置到 INVENTQ 時，PARIS 會自動定義叢集傳送端通道以連接至叢集接收端通道 INVENTORY.NEWYORK。當應用程式的佇列管理程式名稱指定為目標佇列管理程式並提供回覆目的地佇列時，應用程式可以接收回應。

將佇列管理程式新增至叢集: 個別傳輸佇列

請遵循下列指示，將佇列管理程式新增至您建立的叢集。叢集佇列和主題的訊息會使用多個叢集傳輸佇列來傳送。

開始之前

- 佇列管理程式定義在非 z/OS 的平台上。
- 佇列管理程式不是任何叢集的成員。
- 叢集存在; 這個佇列管理程式可以直接連接完整儲存庫, 且儲存庫可供使用。如需建立叢集的步驟, 請參閱 [第 154 頁的『設定新的叢集』](#)。

關於這項作業

此作業是 [第 164 頁的『將佇列管理程式加入叢集中』](#) 的替代方案, 您可以將佇列管理程式新增至將叢集訊息放置在單一傳輸佇列上的叢集。

在這項作業中, 您可以將佇列管理程式新增至叢集, 以針對每一個叢集傳送端通道自動建立個別叢集傳輸佇列。

為了保持佇列的定義數目較小, 預設值是使用單一傳輸佇列。如果您想要監視送往不同佇列管理程式及不同叢集的資料流量, 則使用個別傳輸佇列是有利的。您也可能想要將資料流量分隔至不同的目的地, 以達到隔離或效能目標。

程序

1. 變更預設叢集通道傳輸佇列類型。

變更佇列管理程式 PARIS:

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

每次佇列管理程式建立叢集傳送端通道以將訊息傳送至佇列管理程式時, 它都會建立叢集傳輸佇列。傳輸佇列僅供此叢集傳送端通道使用。傳輸佇列是永久動態的。它是從模型佇列

SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE 建立的, 名稱為
SYSTEM.CLUSTER.TRANSMIT.ChannelName。



小心: 如果您將專用 SYSTEM.CLUSTER.TRANSMIT.QUEUES 與從舊版產品升級的佇列管理程式搭配使用, 請確保 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE 已將 [SHARE/NOSHARE](#) 選項設為 **SHARE**。

2. 決定哪一個完整儲存庫 PARIS 會先參照。

叢集中的每個佇列管理程式都必須參照其中一個或其他完整儲存庫。它會從完整儲存庫收集叢集的相關資訊, 因此會建置它自己的局部儲存庫。選擇任一儲存庫作為完整儲存庫。一旦將新的佇列管理程式新增至叢集, 它也會立即瞭解其他儲存庫。佇列管理程式的變更相關資訊會直接傳送至兩個儲存庫。在此範例中, 您將 PARIS 鏈結至佇列管理程式 LONDON(純粹基於地理原因)。

註: 在佇列管理程式 PARIS 啟動之後, 以任何順序執行其餘步驟。

3. 在佇列管理程式 PARIS 上定義 CLUSRCVR 通道。

叢集中的每個佇列管理程式都必須定義叢集接收端通道, 它可以在該通道上接收訊息。在 PARIS 上, 定義:

```
DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)  
CONNAME(PARIS.CHSTORE.COM) CLUSTER(INVENTORY)  
DESCR('Cluster-receiver channel for queue manager PARIS')
```

叢集接收端通道會通告佇列管理程式的可用性, 以接收來自叢集 INVENTORY 中其他佇列管理程式的訊息。對於叢集接收端通道 INVENTORY.PARIS 的傳送端, 不需要在其他佇列管理程式上進行定義。其他定義會在需要時自動建立。

4. 在佇列管理程式 PARIS 上定義 CLUSSDR 通道。

叢集中的每個佇列管理程式都必須定義一個叢集傳送端通道, 它可以在其中將訊息傳送至其起始完整儲存庫。

在 PARIS 上，針對名為 INVENTORY.LONDON 的通道，對網址為 LONDON.CHSTORE.COM 的佇列管理程式進行下列定義。

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from PARIS to repository at LONDON')
```

佇列管理程式會 SYSTEM.CLUSTER.TRANSMIT.INVENTORY.LONDON 從模型佇列 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE 自動建立永久動態叢集傳輸佇列。它會將傳輸佇列的 CLCHNAME 屬性設為 INVENTORY.LONDON。

結果

下圖顯示此作業所設定的叢集。

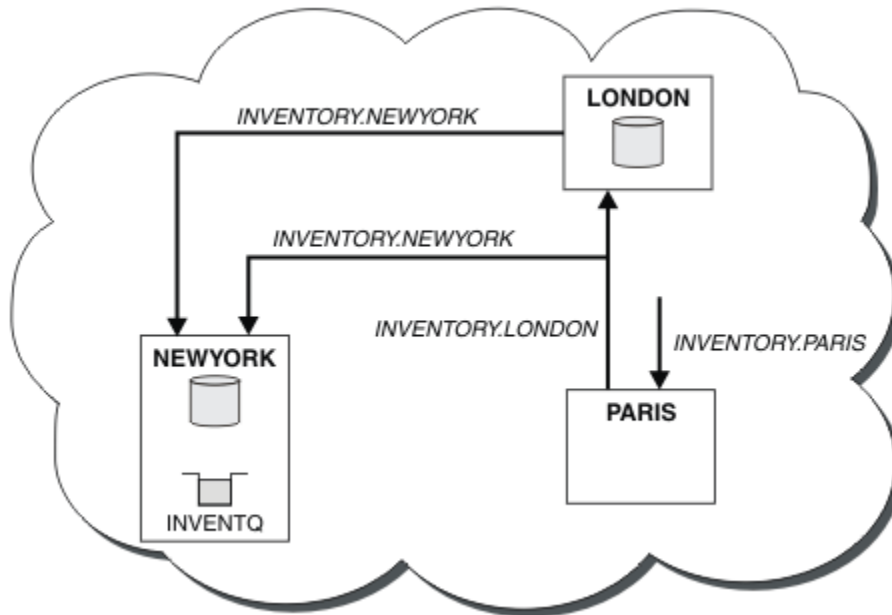


圖 31: 具有三個佇列管理程式的 INVENTORY 叢集

藉由只建立兩個定義: CLUSRCVR 定義和 CLUSSDR 定義，我們將佇列管理程式 PARIS 新增至叢集。

現在，PARIS 佇列管理程式會從 LONDON 的完整儲存庫中，得知 INVENTQ 佇列是由佇列管理程式管理 NEWYORK。當巴黎系統所管理的應用程式嘗試將訊息放置到 INVENTQ 時，PARIS 會自動定義叢集傳送端通道以連接至叢集接收端通道 INVENTORY.NEWYORK。當應用程式的佇列管理程式名稱指定為目標佇列管理程式並提供回覆目的地佇列時，應用程式可以接收回應。

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

修改使用閘道佇列管理程式之重疊叢集的配置。從閘道佇列管理程式將修改訊息傳送至應用程式之後，不會使用與其他叢集訊息相同的傳輸佇列或通道。解決方案使用叢集佇列遠端定義，以及個別傳送端通道和傳輸佇列。

開始之前

遵循第 182 頁的『使用閘道佇列管理程式建立兩個重疊的叢集』中的步驟，在該作業中建構第 183 頁的圖 37 中顯示的重疊叢集。

關於這項作業

解決方案使用分散式佇列，將 Server App 應用程式的訊息與閘道佇列管理程式上的其他訊息資料流量分開。您必須在 QM1 上定義叢集遠端佇列定義，以將訊息轉移至不同的傳輸佇列及不同的通道。遠端佇列定

義必須包含對特定傳輸佇列的參照，該特定傳輸佇列僅儲存 Q1 on QM3 的訊息。在 第 168 頁的圖 32 中，叢集佇列別名 Q1A 由遠端佇列定義 Q1R 補充，並新增傳輸佇列及傳送端通道。

在此解決方案中，會使用一般 SYSTEM.CLUSTER.TRANSMIT.QUEUE 傳回任何回覆訊息。

此解決方案的優點是很容易將相同叢集中相同佇列管理程式上多個目的地佇列的資料流量分開。解決方案的缺點是您無法在不同佇列管理程式上的多個 Q1 副本之間使用叢集工作量平衡。若要克服此缺點，請參閱 第 170 頁的『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』。您也必須管理從一個傳輸佇列到另一個傳輸佇列的交換器。

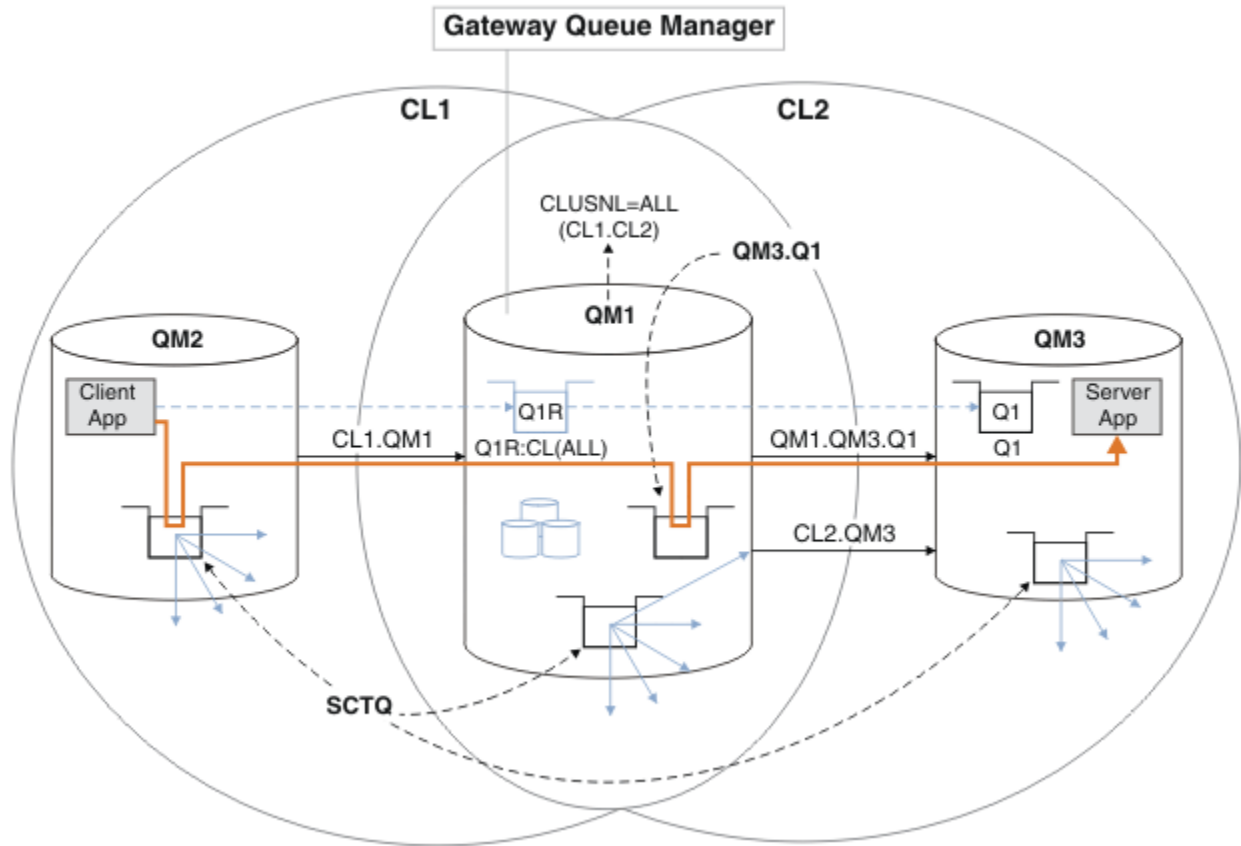


圖 32: 使用遠端佇列定義部署至中心及分支叢集架構的主從架構應用程式

程序

1. 建立通道以區隔 Q1 與閘道佇列管理程式的訊息資料流量

- a) 在閘道佇列管理程式 QM1 上建立傳送端通道，以傳送至目標佇列管理程式 QM3。

```
DEFINE CHANNEL(QM1.QM3.Q1) CHLTYPE(SDR) CONNAME(QM3HostName(1413)) XMITQ(QM3.Q1) REPLACE
```

- b) 在目標佇列管理程式 QM3 上建立接收端通道。

```
DEFINE CHANNEL(QM1.QM3.Q1) CHLTYPE(RCVR) REPLACE
```

2. 在閘道佇列管理程式上建立傳輸佇列，以用於傳送至 Q1 的訊息資料流量

```
DEFINE QLOCAL(QM3.Q1) USAGE(XMITQ) REPLACE
START CHANNEL(QM1.QM3.Q1)
```

啟動與傳輸佇列相關聯的通道，使傳輸佇列與通道相關聯。一旦傳輸佇列與通道相關聯，即會自動啟動通道。

3. 使用叢集遠端佇列定義補充閘道佇列管理程式上 Q1 的叢集佇列別名定義。

下一步

測試配置，方法是使用 Q1R 閘道佇列管理程式上 QM1 的叢集佇列遠端定義，將訊息從 QM2 傳送至 QM3 上的 Q1。

1. 在 QM2 上執行範例程式 **amqsput**，以放置訊息。

```
C:\IBM\MQ>amqsput Q1R QM2
Sample AMQSPUT0 start
target queue is Q1R
Sample request message from QM2 to Q1 using Q1R
```

```
Sample AMQSPUT0 end
```

2. 執行範例程式 **amqsget**，以從 QM3 上的 Q1 取得訊息

```
C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1R>
no more messages
Sample AMQSGET0 end
```

相關概念

[第 135 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 238 頁的『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。

相關工作

[第 165 頁的『將佇列管理程式新增至叢集: 個別傳輸佇列』](#)

請遵循下列指示，將佇列管理程式新增至您建立的叢集。叢集佇列和主題的訊息會使用多個叢集傳輸佇列來傳送。

[第 182 頁的『使用閘道佇列管理程式建立兩個重疊的叢集』](#)

遵循作業中的指示，使用閘道佇列管理程式來建構重疊叢集。使用叢集作為下列範例的起點，將訊息從一個應用程式隔離至叢集中的其他應用程式。

[第 167 頁的『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』](#)

修改使用閘道佇列管理程式之重疊叢集的配置。從閘道佇列管理程式將修改訊息傳送至應用程式之後，不會使用與其他叢集訊息相同的傳輸佇列或通道。解決方案使用叢集佇列遠端定義，以及個別傳送端通道和傳輸佇列。

[第 187 頁的『將預設值變更為個別叢集傳輸佇列以隔離訊息資料流量』](#)

您可以變更佇列管理程式將叢集佇列或主題的訊息儲存在傳輸佇列上的預設方式。變更預設值可讓您隔離閘道佇列管理程式上的叢集訊息。

[第 170 頁的『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』](#)

修改使用閘道佇列管理程式之重疊叢集的配置。從閘道佇列管理程式將修改訊息傳送至應用程式之後，不會使用與其他叢集訊息相同的傳輸佇列或通道。將修改訊息傳送至應用程式之後，解決方案會使用額外的叢集傳輸佇列，將訊息資料流量分隔至叢集中的單一佇列管理程式。

[第 173 頁的『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』](#)

修改使用閘道佇列管理程式之重疊叢集的配置。從閘道佇列管理程式將修改訊息傳送至應用程式之後，不會使用與其他叢集訊息相同的傳輸佇列或通道。將修改訊息傳送至應用程式之後，解決方案會使用其他叢集，將訊息與特定叢集佇列隔離。

[第 241 頁的『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。配置多個叢集傳輸佇列適用於 z/OS 以外的平台。

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

修改使用閘道佇列管理程式之重疊叢集的配置。從閘道佇列管理程式將修改訊息傳送至應用程式之後，不會使用與其他叢集訊息相同的傳輸佇列或通道。將修改訊息傳送至應用程式之後，解決方案會使用額外的叢集傳輸佇列，將訊息資料流量分隔至叢集中的單一佇列管理程式。

開始之前

1. 閘道佇列管理程式必須位於 Version 7.5 或更新版本，以及 z/OS 以外的平台上。
2. 遵循第 182 頁的『[使用閘道佇列管理程式建立兩個重疊的叢集](#)』中的步驟，在該作業中建構第 183 頁的圖 37 中顯示的重疊叢集。

關於這項作業

在閘道佇列管理程式 QM1 上，新增傳輸佇列並設定其佇列屬性 CLCHNAME。將 CLCHNAME 設為 QM3 上叢集接收端通道的名稱；請參閱第 171 頁的圖 33。

與第 167 頁的『[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](#)』中說明的解決方案相比，此解決方案具有許多優點：

- 它需要較少的其他定義。
- 它支援在相同叢集 CL2 中不同佇列管理程式上目標佇列 Q1 的多個副本之間進行工作量平衡。
- 當通道重新啟動時，閘道佇列管理程式會自動切換至新配置，而不會失去任何訊息。
- 閘道佇列管理程式會以收到訊息的相同順序繼續轉遞訊息。它會這樣做，即使交換器是在佇列 Q1 (位於 QM3) 的訊息仍在 SYSTEM.CLUSTER.TRANSMIT.QUEUE 上進行。

在第 171 頁的圖 33 中隔離叢集訊息資料流量的配置不會像在第 167 頁的『[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](#)』中使用遠端佇列的配置那樣對資料流量進行很好的隔離。如果 CL2 中的佇列管理程式 QM3 管理許多不同的叢集佇列及伺服器應用程式，則所有這些佇列都會共用連接 QM1 至 QM3 的叢集通道 CL2.QM3。其他流程在第 171 頁的圖 33 中以灰色箭頭說明，代表從 SYSTEM.CLUSTER.TRANSMIT.QUEUE 到叢集傳送端通道 CL2.QM3 的潛在叢集訊息資料流量。

補救方法是將佇列管理程式限制為管理特定叢集中的一個叢集佇列。如果佇列管理程式已在管理一些叢集佇列，則為了符合此限制，您必須建立另一個佇列管理程式，或建立另一個叢集；請參閱第 173 頁的『[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](#)』。

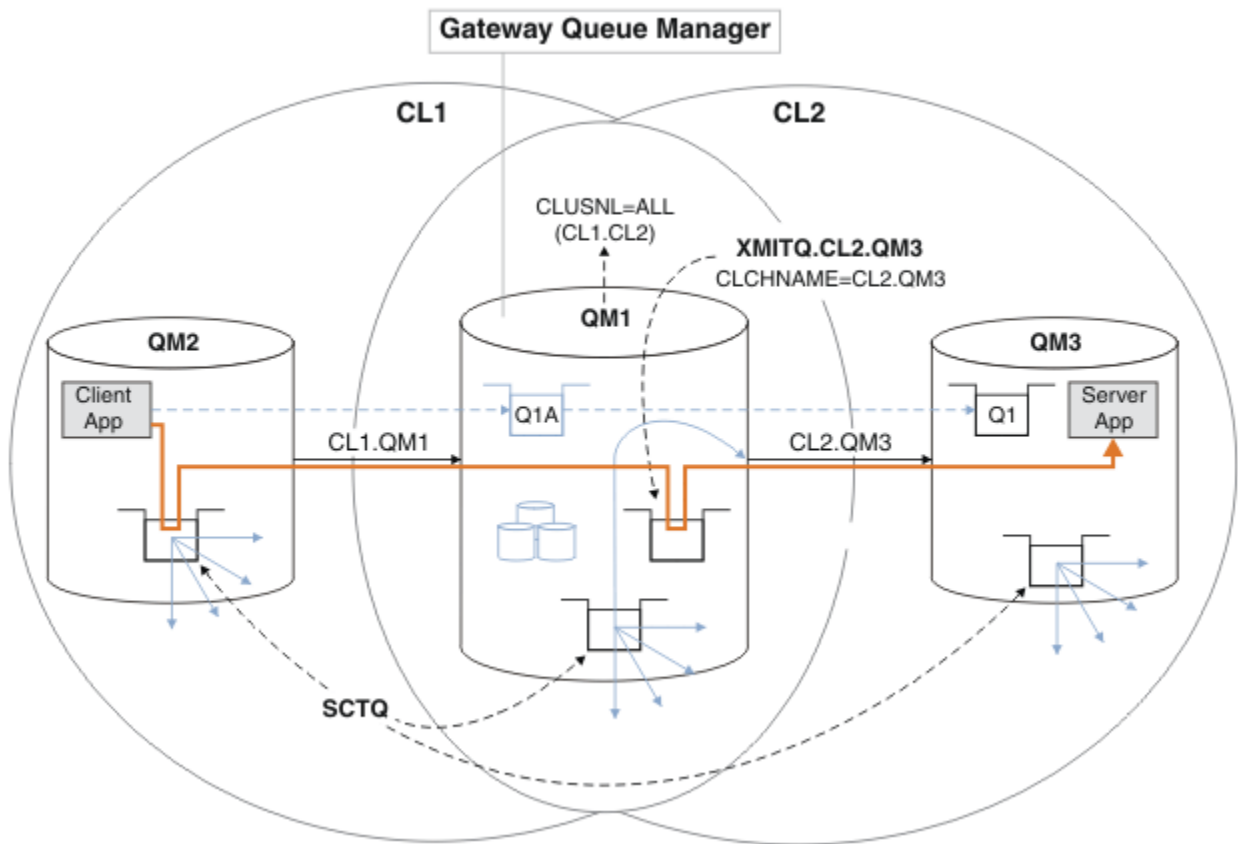


圖 33: 使用其他叢集傳輸佇列部署至中心及分支架構的主從架構應用程式。

程序

1. 在閘道佇列管理程式 QM1 上為叢集傳送端通道 CL2.QM3 建立額外的叢集傳輸佇列。

```
*... on QM1
DEFINE QLOCAL(XMITQ.CL2.QM3) USAGE(XMITQ) CLCHNAME(CL2.QM3)
```

2. 切換至使用傳輸佇列 XMITQ.CL2.QM3。

- a) 停止叢集傳送端通道 CL2.QM3。

```
*... On QM1
STOP CHANNEL(CL2.QM3)
```

回應是接受指令:

AMQ8019: Stop WebSphere MQ channel accepted.

- b) 檢查通道 CL2.QM3 是否已停止

如果通道未停止，您可以使用 FORCE 選項重新執行 **STOP CHANNEL** 指令。設定 FORCE 選項的範例如下: 通道未停止，且您無法重新啟動另一個佇列管理程式來同步化通道。

```
*... On QM1
start
```

回應是通道狀態的摘要

```
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)                CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1413))        CURRENT
```

```
RQMNAME(QM3)                STATUS(STOPPED)
SUBSTATE(MQGET)              XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
```

c) 啟動通道 CL2.QM3。

```
*... On QM1
START CHANNEL(CL2.QM3)
```

回應是接受指令:

```
AMQ8018: Start WebSphere MQ channel accepted.
```

d) 請檢查通道是否已啟動。

```
*... On QM1
DISPLAY CHSTATUS(CL2.QM3)
```

回應是通道狀態的摘要:

```
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)                CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1413))      CURRENT
RQMNAME(QM3)                   STATUS(RUNNING)
SUBSTATE(MQGET)                XMITQ(XMITQ.CL2.QM3)
```

e) 請檢查傳輸佇列是否已切換。

監視閘道佇列管理程式錯誤日誌，以取得訊息 "AMQ7341 通道 CL2.QM3 是 XMITQ.CL2.QM3"。

下一步

使用佇列別名定義 Q1A 將訊息從 QM2 傳送至 QM3 上的 Q1，以測試個別傳輸佇列

1. 在 QM2 上執行範例程式 **amqsput**，以放置訊息。

```
C:\IBM\MQ>amqsput Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
Sample request message from QM2 to Q1 using Q1A
```

```
Sample AMQSPUT0 end
```

2. 執行範例程式 **amqsget**，以從 QM3 上的 Q1 取得訊息

```
C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1A>
no more messages
Sample AMQSGET0 end
```

相關概念

[第 135 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 238 頁的『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。

[第 143 頁的『叢集傳輸佇列及叢集傳送端通道』](#)

叢集佇列管理程式之間的訊息儲存在叢集傳輸佇列上，並由叢集傳送端通道轉遞。

相關工作

[第 165 頁的『將佇列管理程式新增至叢集: 個別傳輸佇列』](#)

請遵循下列指示，將佇列管理程式新增至您建立的叢集。叢集佇列和主題的訊息會使用多個叢集傳輸佇列來傳送。

第 182 頁的『使用閘道佇列管理程式建立兩個重疊的叢集』

遵循作業中的指示，使用閘道佇列管理程式來建構重疊叢集。使用叢集作為下列範例的起點，將訊息從一個應用程式隔離至叢集中的其他應用程式。

第 167 頁的『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』

修改使用閘道佇列管理程式之重疊叢集的配置。從閘道佇列管理程式將修改訊息傳送至應用程式之後，不會使用與其他叢集訊息相同的傳輸佇列或通道。解決方案使用叢集佇列遠端定義，以及個別傳送端通道和傳輸佇列。

第 187 頁的『將預設值變更為個別叢集傳輸佇列以隔離訊息資料流量』

您可以變更佇列管理程式將叢集佇列或主題的訊息儲存在傳輸佇列上的預設方式。變更預設值可讓您隔離閘道佇列管理程式上的叢集訊息。

第 170 頁的『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』

修改使用閘道佇列管理程式之重疊叢集的配置。從閘道佇列管理程式將修改訊息傳送至應用程式之後，不會使用與其他叢集訊息相同的傳輸佇列或通道。將修改訊息傳送至應用程式之後，解決方案會使用額外的叢集傳輸佇列，將訊息資料流量分隔至叢集中的單一佇列管理程式。

第 173 頁的『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』

修改使用閘道佇列管理程式之重疊叢集的配置。從閘道佇列管理程式將修改訊息傳送至應用程式之後，不會使用與其他叢集訊息相同的傳輸佇列或通道。將修改訊息傳送至應用程式之後，解決方案會使用其他叢集，將訊息與特定叢集佇列隔離。

第 241 頁的『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。配置多個叢集傳輸佇列適用於 z/OS 以外的平台。

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

修改使用閘道佇列管理程式之重疊叢集的配置。從閘道佇列管理程式將修改訊息傳送至應用程式之後，不會使用與其他叢集訊息相同的傳輸佇列或通道。將修改訊息傳送至應用程式之後，解決方案會使用其他叢集，將訊息與特定叢集佇列隔離。

開始之前

撰寫作業中的步驟以修改 [第 171 頁的圖 33](#) 中說明的配置。

1. 閘道佇列管理程式必須位於 Version 7.5 或更新版本，以及 z/OS 以外的平台上。
2. 遵循 [第 182 頁的『使用閘道佇列管理程式建立兩個重疊的叢集』](#) 中的步驟，在該作業中建構 [第 183 頁的圖 37](#) 中顯示的重疊叢集。
3. 在 [第 170 頁的『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』](#) 中執行 [第 171 頁的圖 33](#) 中的步驟，以在沒有其他叢集的情況下建立解決方案。使用此作為此作業中步驟的基礎。

關於這項作業

如果目標叢集佇列是佇列管理程式上唯一的叢集佇列，則在 [第 170 頁的『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』](#) 中隔離單一應用程式的訊息資料流量的解決方案可以運作。如果不是，你有兩個選擇。請將佇列移至不同的佇列管理程式，或建立叢集，將佇列與佇列管理程式上的其他叢集佇列隔離。

此作業會引導您完成新增叢集以隔離目標佇列的步驟。只會針對該目的新增叢集。實際上，當您正在設計叢集及叢集命名方法時，請有系統地執行隔離特定應用程式的作業。每次佇列需要隔離時新增叢集，最後可能會有許多要管理的叢集。在此作業中，您可以透過新增叢集 CL3 以隔離 QM3 上的 Q1，來變更 [第 170 頁的『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。在整個變更期間，應用程式會繼續執行。

新的和已變更的定義會在 [第 174 頁的圖 34](#) 中強調顯示。變更摘要如下: 建立叢集，這表示您也必須建立新的完整叢集儲存庫。在此範例中，QM3 會成為 CL3 的其中一個完整儲存庫。建立 QM1 的叢集傳送端及叢集接收端通道，以將閘道佇列管理程式新增至新叢集。變更 Q1 的定義，以將它切換至 CL3。修改閘道佇列管

程式上的叢集名單，並新增叢集傳輸佇列以使用新的叢集通道。最後，將佇列別名 Q1A 切換至新的叢集名稱清單。

IBM WebSphere MQ 無法自動將訊息從您在第 170 頁的『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』中新增的傳輸佇列 XMITQ.CL2.QM3 傳送至新的傳輸佇列 XMITQ.CL3.QM3。只有在兩個傳輸佇列都由相同的叢集傳送端通道提供服務時，它才能自動傳送訊息。相反地，此作業說明一種手動執行交換器的方法，這可能適合您。當傳送完成時，您可以選擇回復為針對 QM3 上的其他 CL2 叢集佇列使用預設叢集傳輸佇列。或者，您可以繼續使用 XMITQ.CL2.QM3。如果您決定回復至預設叢集傳輸佇列，閘道佇列管理程式會自動為您管理交換器。

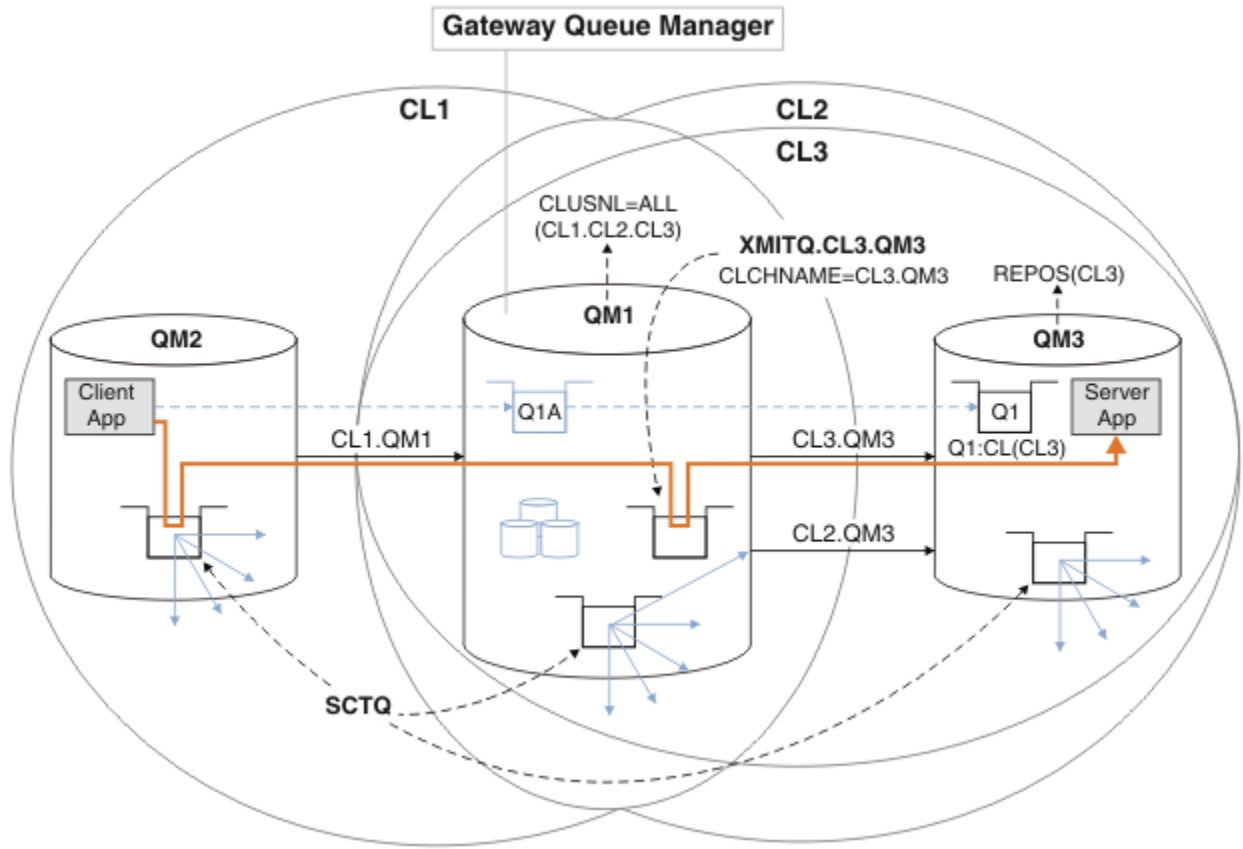


圖 34: 使用額外叢集來區隔閘道佇列管理程式中的訊息資料流量，該閘道佇列管理程式會進入相同佇列管理程式上的許多叢集佇列之一

程序

1. 變更佇列管理程式 QM3 和 QM5，使它們同時成為 CL2 和 CL3 的儲存庫。

若要让佇列管理程式成為多個叢集的成員，它必須使用叢集名稱清單來識別其所屬的叢集。

```
*... On QM3 and QM5
DEFINE NAMELIST(CL23) NAMES(CL2, CL3) REPLACE
ALTER QMGR REPOS(' ') REPOSNL(CL23)
```

2. 針對 CL3，定義佇列管理程式 QM3 與 QM5 之間的通道。

```
*... On QM3
DEFINE CHANNEL(CL3.QM5) CHLTYPE(CLUSSDR) CONNAME('localhost(1415)') CLUSTER(CL3) REPLACE
DEFINE CHANNEL(CL3.QM3) CHLTYPE(CLUSRCVR) CONNAME('localhost(1413)') CLUSTER(CL3) REPLACE

*... On QM5
DEFINE CHANNEL(CL3.QM3) CHLTYPE(CLUSSDR) CONNAME('localhost(1413)') CLUSTER(CL3) REPLACE
DEFINE CHANNEL(CL3.QM5) CHLTYPE(CLUSRCVR) CONNAME('localhost(1415)') CLUSTER(CL3) REPLACE
```

3. 將閘道佇列管理程式新增至 CL3。

將 QM1 新增至 CL3 作為局部儲存庫，以新增閘道佇列管理程式。將叢集傳送端和叢集接收端通道新增至 QM1，以建立局部儲存庫。

此外，將 CL3 新增至連接至閘道佇列管理程式之所有叢集的名稱清單。

```
*... On QM1
DEFINE CHANNEL(CL3.QM3) CHLTYPE(CLUSSDR) CONNAME('localhost(1413)') CLUSTER(CL3) REPLACE
DEFINE CHANNEL(CL3.QM1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1411)') CLUSTER(CL3) REPLACE
ALTER NAMELIST(ALL) NAMES(CL1, CL2, CL3)
```

4. 針對移至 QM3 上 CL3 的訊息，將叢集傳輸佇列新增至閘道佇列管理程式 QM1。

一開始，停止叢集傳送端通道從傳輸佇列傳送訊息，直到您準備好切換傳輸佇列為止。

```
*... On QM1
DEFINE QLOCAL(XMITQ.CL3.QM3) USAGE(XMITQ) CLCHNAME(CL3.QM3) GET(DISABLED) REPLACE
```

5. 從現有叢集傳輸佇列 XMITQ.CL2.QM3 排除訊息。

此子程序旨在保留 Q1 中的訊息順序，以符合它們到達閘道佇列管理程式的順序。使用叢集時，無法完全保證訊息排序，但很可能如此。如果需要保證訊息排序，應用程式必須定義訊息順序；請參閱 [從佇列擷取訊息的順序](#)。

- a) 將 QM3 上的目標佇列 Q1 從 CL2 變更為 CL3。

```
*... On QM3
ALTER QLOCAL(Q1) CLUSTER(CL3)
```

- b) 監視 XMITQ.CL3.QM3，直到開始將訊息遞送給它為止。

當將 Q1 至 CL3 的切換延伸到閘道佇列管理程式時，開始將訊息遞送至 XMITQ.CL3.QM3。

```
*... On QM1
DISPLAY QUEUE(XMITQ.CL3.QM3) CURDEPTH
```

- c) 監視 XMITQ.CL2.QM3，直到它沒有訊息等待遞送至 QM3 上的 Q1。

註：XMITQ.CL2.QM3 可能正在儲存 QM3 上屬於 CL2 成員之其他佇列的訊息，在此情況下深度可能不會變成零。

```
*... On QM1
DISPLAY QUEUE(XMITQ.CL2.QM3) CURDEPTH
```

- d) 啟用從新的叢集傳輸佇列 XMITQ.CL3.QM3 取得

```
*... On QM1
ALTER QLOCAL(XMITQ.CL3.QM3) GET(ENABLED)
```

6. 移除舊叢集傳輸佇列 XMITQ.CL2.QM3(如果不再需要的話)。

QM3 上 CL2 中叢集佇列的訊息會回復為使用閘道佇列管理程式 QM1 上的預設叢集傳輸佇列。預設叢集傳輸佇列為 SYSTEM.CLUSTER.TRANSMIT.QUEUE 或 SYSTEM.CLUSTER.TRANSMIT.CL2.QM3。哪一個取決於 QM1 上佇列管理程式屬性 **DEFCLXQ** 的值是 SCTQ 還是 CHANNEL。當叢集傳送端通道 CL2.QM3 下次啟動時，佇列管理程式會自動從 XMITQ.CL2.QM3 傳送訊息。

- a) 將傳輸佇列 XMITQ.CL2.QM3 從叢集傳輸佇列變更為正常傳輸佇列。

這會岔斷傳輸佇列與任何叢集傳送端通道的關聯。在下次啟動叢集傳送端通道時，IBM WebSphere MQ 會自動將訊息從 XMITQ.CL2.QM3 傳送至預設叢集傳輸佇列。在此之前，QM3 上 CL2 的訊息會繼續放置在 XMITQ.CL2.QM3 上。

```
*... On QM1
ALTER QLOCAL(XMITQ.CL2.QM3) CLCHNAME('')
```

- b) 停止叢集傳送端通道 CL2.QM3。

停止並重新啟動叢集傳送端通道會起始將訊息從 XMITQ.CL2.QM3 傳送至預設叢集傳輸佇列。通常您會停止並手動啟動通道，以啟動傳送。如果通道在其斷線間隔到期時關閉之後重新啟動，則會自動啟動傳送。

```
*... On QM1
STOP CHANNEL(CL2.QM3)
```

回應是接受指令：

AMQ8019: Stop WebSphere MQ channel accepted.

c) 檢查通道 CL2.QM3 是否已停止

如果通道未停止，您可以使用 FORCE 選項重新執行 **STOP CHANNEL** 指令。設定 FORCE 選項的範例如下：通道未停止，且您無法重新啟動另一個佇列管理程式來同步化通道。

```
*... On QM1
DISPLAY CHSTATUS(CL2.QM3)
```

回應是通道狀態的摘要

AMQ8417: Display Channel Status details.

CHANNEL(CL2.QM3)	CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1413))	CURRENT
RQMNAME(QM3)	STATUS(STOPPED)
SUBSTATE(MQGET)	XMITQ(XMITQ.CL2.QM3)

d) 啟動通道 CL2.QM3。

```
*... On QM1
START CHANNEL(CL2.QM3)
```

回應是接受指令：

AMQ8018: Start WebSphere MQ channel accepted.

e) 請檢查通道是否已啟動。

```
*... On QM1
DISPLAY CHSTATUS(CL2.QM3)
```

回應是通道狀態的摘要：

```
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)          CHLTYPE(CLUSSDR)
CONNNAME(127.0.0.1(1413)) CURRENT
RQMNAME(QM3)             STATUS(RUNNING)
SUBSTATE(MQGET)          XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE|CL2.QM3)
```

f) 監視閘道佇列管理程式錯誤日誌，以取得訊息 "AMQ7341 通道 CL2.QM3 是 SYSTEM.CLUSTER.TRANSMIT.QUEUE | CL2.QM3"。

g) 刪除叢集傳輸佇列 XMITQ.CL2.QM3。

```
*... On QM1
DELETE QLOCAL(XMITQ.CL2.QM3)
```

下一步

使用佇列別名定義 Q1A 將訊息從 QM2 傳送至 QM3 上的 Q1，以測試個別叢集佇列。

1. 在 QM2 上執行範例程式 **amqspu**t，以放置訊息。

```
C:\IBM\MQ>amqspu Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
```

Sample request message from QM2 to Q1 using Q1A

Sample AMQSPUT0 end

2. 執行範例程式 **amqsget**，以從 QM3 上的 Q1 取得訊息

```
C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1A>
no more messages
Sample AMQSGET0 end
```

相關概念

[第 135 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 238 頁的『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。

[第 143 頁的『叢集傳輸佇列及叢集傳送端通道』](#)

叢集佇列管理程式之間的訊息儲存在叢集傳輸佇列上，並由叢集傳送端通道轉遞。

相關工作

[第 165 頁的『將佇列管理程式新增至叢集: 個別傳輸佇列』](#)

請遵循下列指示，將佇列管理程式新增至您建立的叢集。叢集佇列和主題的訊息會使用多個叢集傳輸佇列來傳送。

[第 182 頁的『使用閘道佇列管理程式建立兩個重疊的叢集』](#)

遵循作業中的指示，使用閘道佇列管理程式來建構重疊叢集。使用叢集作為下列範例的起點，將訊息從一個應用程式隔離至叢集中的其他應用程式。

[第 167 頁的『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』](#)

修改使用閘道佇列管理程式之重疊叢集的配置。從閘道佇列管理程式將修改訊息傳送至應用程式之後，不會使用與其他叢集訊息相同的傳輸佇列或通道。解決方案使用叢集佇列遠端定義，以及個別傳送端通道和傳輸佇列。

[第 187 頁的『將預設值變更為個別叢集傳輸佇列以隔離訊息資料流量』](#)

您可以變更佇列管理程式將叢集佇列或主題的訊息儲存在傳輸佇列上的預設方式。變更預設值可讓您隔離閘道佇列管理程式上的叢集訊息。

[第 170 頁的『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』](#)

修改使用閘道佇列管理程式之重疊叢集的配置。從閘道佇列管理程式將修改訊息傳送至應用程式之後，不會使用與其他叢集訊息相同的傳輸佇列或通道。將修改訊息傳送至應用程式之後，解決方案會使用額外的叢集傳輸佇列，將訊息資料流量分隔至叢集中的單一佇列管理程式。

[第 173 頁的『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』](#)

修改使用閘道佇列管理程式之重疊叢集的配置。從閘道佇列管理程式將修改訊息傳送至應用程式之後，不會使用與其他叢集訊息相同的傳輸佇列或通道。將修改訊息傳送至應用程式之後，解決方案會使用其他叢集，將訊息與特定叢集佇列隔離。

[第 241 頁的『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。配置多個叢集傳輸佇列適用於 z/OS 以外的平台。

使用 DHCP 將佇列管理程式新增至叢集

使用 DHCP 將佇列管理程式新增至叢集。此作業示範在 CLUSRCVR 定義中省略 CONNAME 值。

開始之前

註: 如果要將叢集的變更延伸到整個叢集，至少必須一律有一個完整儲存庫可用。在啟動這項作業之前，請確定您的儲存庫可供使用。

此作業示範兩個特殊特性：

- 在 CLUSRCVR 定義上省略 CONNAME 值的能力。
- 在 CLUSSDR 定義上使用 +QMNAME+ 的能力。

z/OS 上未提供任何特性。

測試情境：

- 已依照第 154 頁的『設定新的叢集』中的說明來設定 INVENTORY 叢集。它包含兩個佇列管理程式 LONDON 和 NEWYORK，這兩個佇列管理程式都保留完整儲存庫。
- 正在巴黎設定鏈結儲存庫的新分支，且您想要將稱為 PARIS 的佇列管理程式新增至叢集。
- 佇列管理程式 PARIS 會將訊息放入 INVENTQ 佇列，以將庫存更新項目傳送至在紐約系統上執行的應用程式。
- 所有三個系統之間都存在網路連線功能。
- 網路通訊協定是 TCP。
- PARIS 佇列管理程式系統會使用 DHCP，這表示 IP 位址可能會在系統重新啟動時變更。
- PARIS 與 LONDON 系統之間的通道會根據已定義的命名慣例來命名。慣例會使用 LONDON 上完整儲存庫佇列管理程式的佇列管理程式名稱。
- PARIS 佇列管理程式的管理者沒有 LONDON 儲存庫上佇列管理程式名稱的相關資訊。LONDON 儲存庫上的佇列管理程式名稱可能會變更。

關於這項作業

請遵循下列步驟，使用 DHCP 將佇列管理程式新增至叢集。

程序

1. 決定哪一個完整儲存庫 PARIS 會先參照。

叢集中的每個佇列管理程式都必須參照其中一個或其他完整儲存庫。它會從完整儲存庫收集叢集的相關資訊，因此會建置它自己的局部儲存庫。選擇任一儲存庫作為完整儲存庫。一旦將新的佇列管理程式新增至叢集，它也會立即瞭解其他儲存庫。佇列管理程式的變更相關資訊會直接傳送至兩個儲存庫。在此範例中，我們選擇將 PARIS 鏈結至佇列管理程式 LONDON，純粹是基於地理原因。

註: 在佇列管理程式 PARIS 啟動之後，以任何順序執行其餘步驟。

2. 在佇列管理程式 PARIS 上定義 CLUSRCVR 通道。

叢集中的每個佇列管理程式都需要定義可接收訊息的叢集接收端通道。在 PARIS 上，定義：

```
DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSRCVR)
TRPTYPE(TCP) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for queue manager PARIS')
```

叢集接收端通道會通告佇列管理程式的可用性，以接收來自叢集 INVENTORY 中其他佇列管理程式的訊息。您不需要在叢集接收端通道上指定 CONNAME。您可以透過省略 CONNAME 或指定 CONNAME(' ')，來要求 IBM WebSphere MQ 從系統中找出連線名稱。IBM WebSphere MQ 會使用系統的現行 IP 位址來產生 CONNAME 值；請參閱 CONNAME。對於叢集接收端通道 INVENTORY.PARIS 的傳送端，不需要在其他佇列管理程式上進行定義。其他定義會在需要時自動建立。

3. 在佇列管理程式 PARIS 上定義 CLUSSDR 通道。

叢集中的每個佇列管理程式都需要定義一個叢集傳送端通道，它可以在其中將訊息傳送至其起始完整儲存庫。在 PARIS 上，針對名為 INVENTORY.+QMNAME+ 的通道，對網址為 LONDON.CHSTORE.COM 的佇列管理程式進行下列定義。

```
DEFINE CHANNEL(INVENTORY.+QMNAME+) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from PARIS to repository at LONDON')
```

4. 選擇性的: 如果此佇列管理程式正在重新加入叢集，請完成一些額外步驟。

a) 如果您要將佇列管理程式新增至先前已從相同叢集中移除的叢集，請檢查它現在是否顯示為叢集成員。如果沒有，請完成下列額外步驟:

i) 在您要新增的佇列管理程式上發出 **REFRESH CLUSTER** 指令。此步驟會停止叢集通道，並為本端叢集快取提供一組全新的序號，確保在叢集其餘部分內保持最新。

```
REFRESH CLUSTER(INVENTORY) REPOS(YES)
```

註: 對於大型叢集，使用 **REFRESH CLUSTER** 指令會干擾進行中的叢集，而此後每隔 27 天，當叢集物件自動將狀態更新傳送給所有相關的佇列管理程式時，會再次造成干擾。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。

ii) 重新啟動 CLUSSDR 通道 (例如，使用 **START CHANNEL** 指令)。

iii) 重新啟動 CLUSRCVR 通道。

b) 如果叢集是發佈/訂閱叢集，且重新加入的佇列管理程式有訂閱，請發出下列指令，以確保在叢集中正確同步化 Proxy 訂閱:

```
REFRESH QMGR TYPE(PROXYSUB)
```

結果

此作業所設定的叢集與第 164 頁的『將佇列管理程式加入叢集中』的叢集相同:

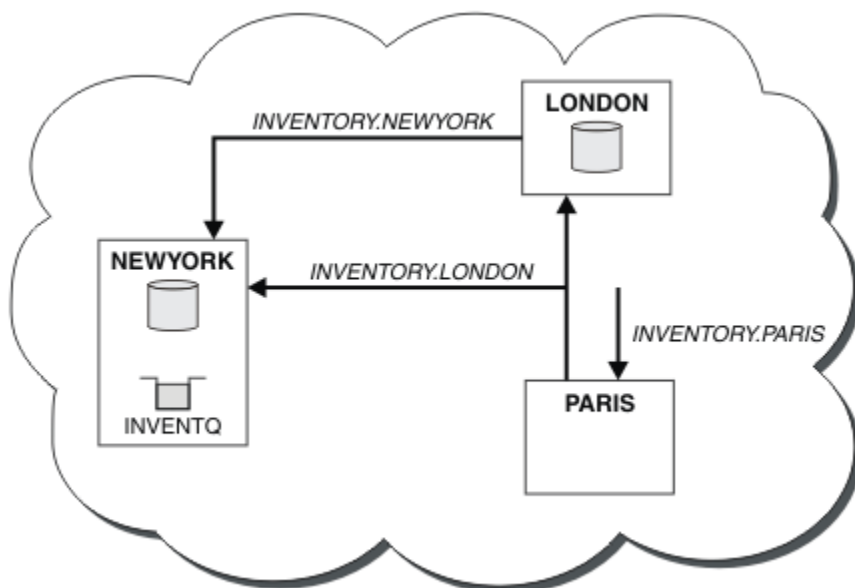


圖 35: 具有三個佇列管理程式的 INVENTORY 叢集

透過只建立兩個定義: CLUSRCVR 定義及 CLUSSDR 定義，我們已將佇列管理程式 PARIS 新增至叢集。

在 PARIS 佇列管理程式上，會啟動包含字串 +QMNAME+ 的 CLUSSDR。在 LONDON 系統上，IBM WebSphere MQ 會將 +QMNAME+ 解析為佇列管理程式名稱 (LONDON)。然後，IBM WebSphere MQ 會比對稱為 INVENTORY.LONDON 之通道的定義與對應的 CLUSRCVR 定義。

WebSphere MQ 會將已解析的通道名稱傳回 PARIS 佇列管理程式。在 PARIS 中，稱為 INVENTORY.+QMNAME+ 之通道的 CLUSSDR 通道定義會取代為 INVENTORY.LONDON 的內部產生 CLUSSDR 定義。此定義包含已解析的通道名稱，但否則與您所做的 +QMNAME+ 定義相同。叢集儲存庫也會隨具有新解析之通道名稱的通道定義一起更新。

註：

1. 以 +QMNAME+ 名稱建立的通道會立即變成非作用中。它絕不會用來傳輸資料。
2. 通道結束程式可能會看到在一次呼叫與下一次呼叫之間的通道名稱變更。

現在，PARIS 佇列管理程式會從 LONDON 中的儲存庫，得知 INVENTQ 佇列是由佇列管理程式 NEWYORK 所管理。當巴黎系統所管理的應用程式嘗試將訊息放置到 INVENTQ，PARIS 時，會自動定義叢集傳送端通道，以連接至叢集接收端通道 INVENTORY.NEWYORK。當應用程式的佇列管理程式名稱指定為目標佇列管理程式並提供回覆目的地佇列時，應用程式可以接收回應。

相關參考

[定義通道](#)

新增管理佇列的佇列管理程式

將另一個佇列管理程式新增至叢集，以管理另一個 INVENTQ 佇列。要求會交替傳送至每一個佇列管理程式上的佇列。不需要對現有 INVENTQ 主機進行任何變更。

開始之前

註：如果要將叢集的變更延伸到整個叢集，至少必須一律有一個完整儲存庫可用。在啟動這項作業之前，請確定您的儲存庫可供使用。

測試情境：

- 已依照第 164 頁的『將佇列管理程式加入叢集中』中的說明來設定 INVENTORY 叢集。它包含三個佇列管理程式：LONDON 和 NEWYORK 都保留完整儲存庫，PARIS 則保留局部儲存庫。庫存應用程式在紐約的系統上執行，並連接至 NEWYORK 佇列管理程式。應用程式是由 INVENTQ 佇列上的訊息到達所驅動。
- 多倫多正在建立一家新商店。為了提供額外容量，您想要在多倫多及紐約的系統上執行庫存應用程式。
- 所有四個系統之間都存在網路連線功能。
- 網路通訊協定是 TCP。

註：佇列管理程式 TORONTO 只包含局部儲存庫。如果您要將完整儲存庫佇列管理程式新增至叢集，請參閱第 193 頁的『將完整儲存庫移至另一個佇列管理程式』。

關於這項作業

請遵循下列步驟來新增管理佇列的佇列管理程式。

程序

1. 決定哪一個完整儲存庫 TORONTO 會先參照。

叢集中的每個佇列管理程式都必須參照其中一個或其他完整儲存庫。它會從完整儲存庫收集叢集的相關資訊，因此會建置它自己的局部儲存庫。您選擇的儲存庫沒有特別重要的意義。在此範例中，我們選擇 NEWYORK。一旦新佇列管理程式已加入叢集，它就會與這兩個儲存庫進行通訊。

2. 定義 CLUSRCVR 通道。

叢集中的每個佇列管理程式都需要定義可接收訊息的叢集接收端通道。在 TORONTO 上，定義 CLUSRCVR 通道：

```
DEFINE CHANNEL(INVENTORY.TORONTO) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(TORONTO.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for TORONTO')
```

TORONTO 佇列管理程式會通告其可用性，以使用其叢集接收端通道來接收來自 INVENTORY 叢集中其他佇列管理程式的訊息。

3. 在佇列管理程式 TORONTO 上定義 CLUSSDR 通道。

叢集中的每個佇列管理程式都需要定義一個叢集傳送端通道，以將訊息傳送至其第一個完整儲存庫。在此情況下，請選擇 NEWYORK。TORONTO 需要下列定義：

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from TORONTO to repository at NEWYORK')
```

4. 選擇性的: 如果此佇列管理程式正在重新加入叢集，請完成一些額外步驟。

- a) 如果您要將佇列管理程式新增至先前已從相同叢集中移除的叢集，請檢查它現在是否顯示為叢集成員。如果沒有，請完成下列額外步驟：
 - i) 在您要新增的佇列管理程式上發出 **REFRESH CLUSTER** 指令。此步驟會停止叢集通道，並為本端叢集快取提供一組全新的序號，確保在叢集其餘部分內保持最新。

```
REFRESH CLUSTER(INVENTORY) REPOS(YES)
```

註: 對於大型叢集，使用 **REFRESH CLUSTER** 指令會干擾進行中的叢集，而此後每隔 27 天，當叢集物件自動將狀態更新傳送給所有相關的佇列管理程式時，會再次造成干擾。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。

- ii) 重新啟動 CLUSSDR 通道 (例如，使用 [START CHANNEL](#) 指令)。

- iii) 重新啟動 CLUSRCVR 通道。

- b) 如果叢集是發佈/訂閱叢集，且重新加入的佇列管理程式有訂閱，請發出下列指令，以確保在叢集中正確同步化 Proxy 訂閱：

```
REFRESH QMGR TYPE(PROXYSUB)
```

5. 請檢閱庫存應用程式，以取得訊息親緣性。

在繼續之前，請確定庫存應用程式對訊息處理順序沒有任何相依關係，並在多倫多的系統上安裝應用程式。

6. 定義叢集佇列 INVENTQ。

INVENTQ 佇列 (已由 NEWYORK 佇列管理程式管理) 也將由 TORONTO 管理。在 TORONTO 佇列管理程式上定義它，如下所示：

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

結果

第 182 頁的圖 36 顯示此作業所設定的 INVENTORY 叢集。

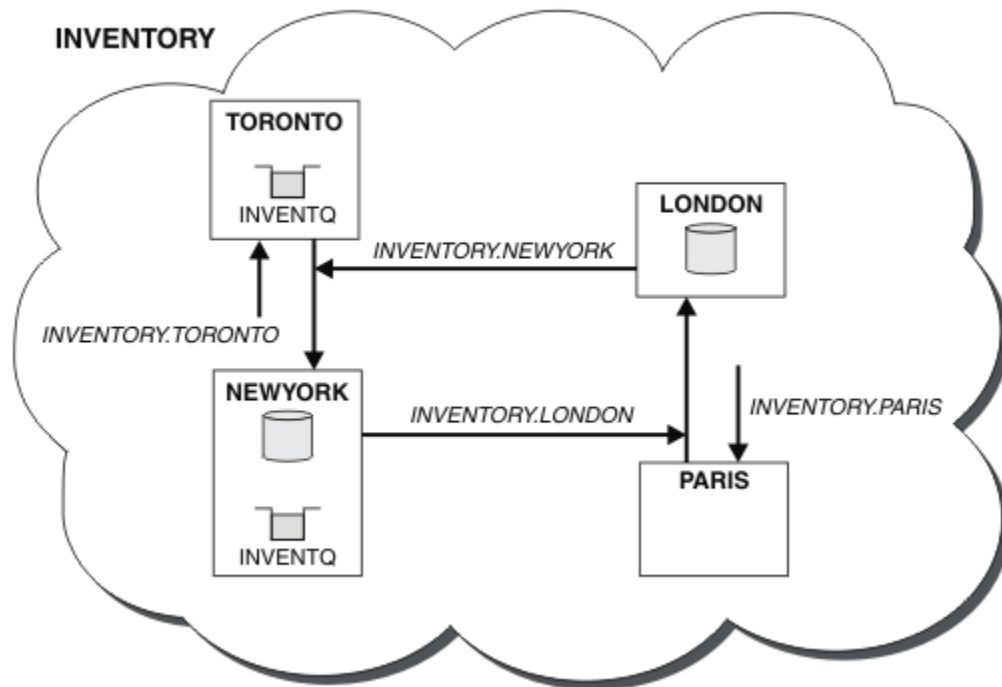


圖 36: 具有四個佇列管理程式的 INVENTORY 叢集

現在，INVENTQ 佇列和庫存應用程式是在叢集中的兩個佇列管理程式上管理。這會增加其可用性，加快訊息的傳輸量，並容許在兩個佇列管理程式之間分配工作量。由 TORONTO 或 NEWYORK 放置至 INVENTQ 的訊息，會盡量由本端佇列管理程式上的實例處理。LONDON 或 PARIS 所放置的訊息會交替遞送至 TORONTO 或 NEWYORK，以平衡工作量。

叢集的這項修改已完成，您無需變更佇列管理程式 NEWYORK、LONDON 及 PARIS 上的定義。這些佇列管理程式中的完整儲存庫會自動更新為能夠傳送訊息至 INVENTQ 所需的資訊，網址為 TORONTO。如果其中一個 NEWYORK 或 TORONTO 佇列管理程式變成無法使用，且具有足夠容量，則庫存應用程式會繼續運作。如果庫存應用程式同時在兩個位置中管理，則必須能夠正確運作。

您可以從這項作業的結果中看到，您可以讓相同的應用程式在多個佇列管理程式上執行。您可以形成叢集以平均分配工作量。

應用程式可能無法處理兩個位置中的記錄。例如，假設您決定新增客戶帳戶查詢，並更新在 LONDON 和 NEWYORK 中執行的應用程式。帳戶記錄只能保留在一個位置。您可以決定使用資料分割技術來控制要求的配送。您可以分割記錄的分佈。您可以安排將半數記錄 (例如帳號 00000-49999) 保留在 LONDON 中。另一半 (在 50000-99999 範圍內) 保留在 NEWYORK 中。然後您可以撰寫叢集工作量結束程式，以檢查所有訊息中的帳戶欄位，並將訊息遞送至適當的佇列管理程式。

下一步

既然您已完成所有定義，如果您尚未完成，請在 IBM WebSphere MQ for z/OS 上啟動通道起始程式。在所有平台上，在佇列管理程式 TORONTO 上啟動接聽器程式。接聽器程式會等待送入的網路要求，並在需要時啟動叢集接收端通道。

使用閘道佇列管理程式建立兩個重疊的叢集

遵循作業中的指示，使用閘道佇列管理程式來建構重疊叢集。使用叢集作為下列範例的起點，將訊息從一個應用程式隔離至叢集中的其他應用程式。

關於這項作業

第 183 頁的圖 37 中顯示用來說明隔離叢集訊息資料流量的範例叢集配置。第 238 頁的『叢集作業: 使用多個叢集傳輸佇列的應用程式隔離』中說明此範例。

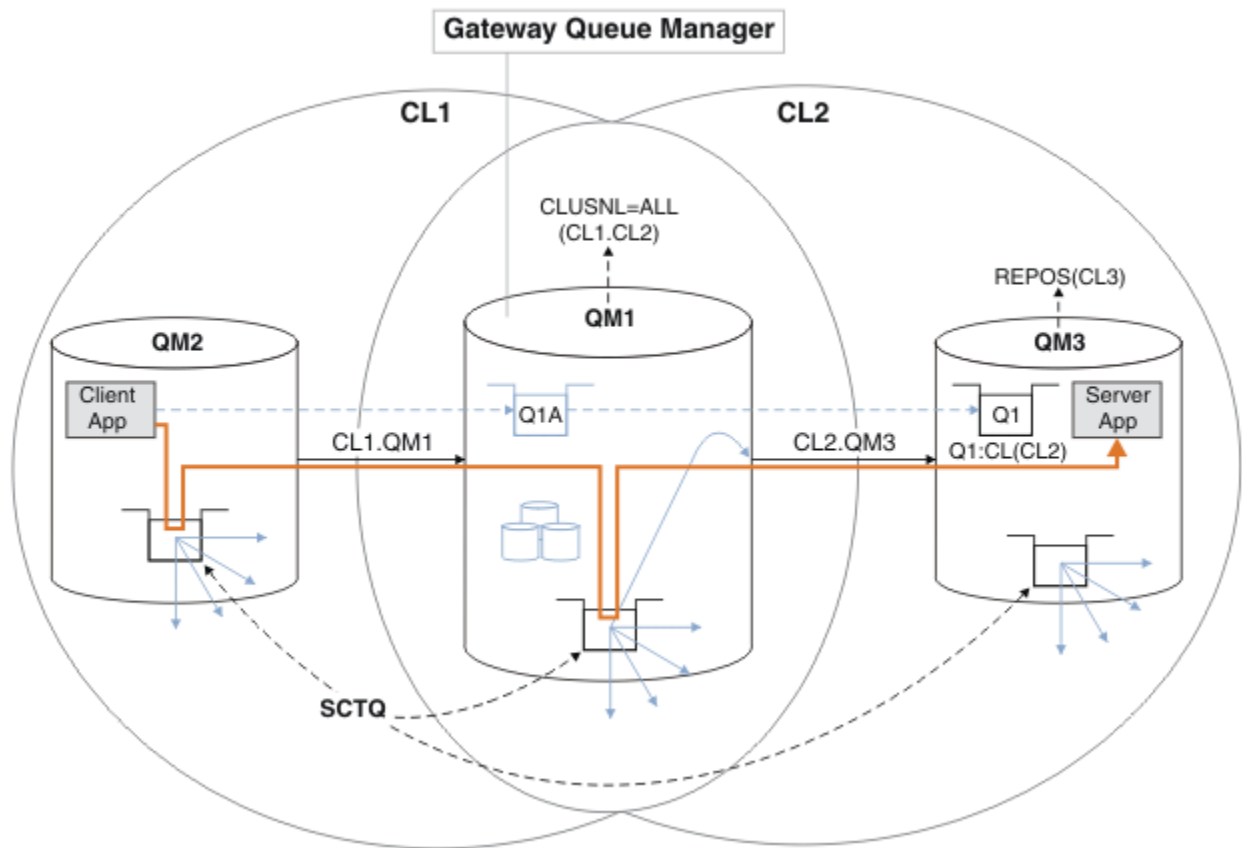


圖 37: 使用 IBM WebSphere MQ 叢集部署至中心及分支架構的主從架構應用程式

為了儘可能減少建構範例的步驟數，配置會保持簡單，而不是實際。此範例可能代表兩個不同組織所建立之兩個叢集的整合。如需更實際的實務範例，請參閱第 241 頁的『叢集作業: 規劃如何配置叢集傳輸佇列』。

請遵循步驟來建構叢集。在下列範例中使用這些叢集，以隔離從用戶端應用程式到伺服器應用程式的訊息資料流量。

這些指示會新增一些額外的佇列管理程式，以便每一個叢集都有兩個儲存庫。基於效能原因，開道佇列管理程式未用作儲存庫。

程序

1. 建立並啟動佇列管理程式 QM1、QM2、QM3、QM4、QM5。

```
crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE QMn
strmqm QmgrName
```

註: QM4 和 QM5 是叢集的備份完整儲存庫。

2. 定義並啟動每一個佇列管理程式的接聽器。

```
*... On QMn
DEFINE LISTENER(TCP141n) TRPTYPE(TCP) IPADDR(hostname) PORT(141n) CONTROL(QMGR) REPLACE
START LISTENER(TCP141n)
```

3. 建立所有叢集的叢集名稱清單。

```
*... On QM1
DEFINE NAMELIST(ALL) NAMES(CL1, CL2) REPLACE
```

4. 建立 CL1、QM3 的 QM2 和 QM4 完整儲存庫，以及 CL2 的 QM5 完整儲存庫。

a) 若為 CL1:

```
*... On QM2 and QM4
ALTER QMGR REPOS(CL1) DEFCLXQ(SCTQ)
```

b) 若為 CL2:

```
*... On QM3 and QM5
ALTER QMGR REPOS(CL2) DEFCLXQ(SCTQ)
```

5. 新增每一個佇列管理程式和叢集的叢集傳送端和叢集接收端通道。

在 QM2、QM3、QM4 及 QM5 上執行下列指令，其中 *c*、*n* 及 *m* 會採用 [第 184 頁的表 26](#) 中針對每一個佇列管理程式所顯示的值:

佇列管理程式	叢集 <i>c</i>	其他儲存庫 <i>n</i>	此儲存庫 <i>m</i>
QM2	1	4	2
QM4	1	2	4
QM3	2	5	3
QM5	2	3	5

```
*... On QMm
DEFINE CHANNEL(CLc.QMn) CHLTYPE(CLUSSDR) CONNAME('localhost(141n)') CLUSTER(CLc) REPLACE
DEFINE CHANNEL(CLc.QMm) CHLTYPE(CLUSRCVR) CONNAME('localhost(141m)') CLUSTER(CLc) REPLACE
```

6. 將閘道佇列管理程式 QM1 新增至每一個叢集。

```
*... On QM1
DEFINE CHANNEL(CL1.QM2) CHLTYPE(CLUSSDR) CONNAME('localhost(1412)') CLUSTER(CL1) REPLACE
DEFINE CHANNEL(CL1.QM1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1411)') CLUSTER(CL1) REPLACE
DEFINE CHANNEL(CL2.QM3) CHLTYPE(CLUSSDR) CONNAME('localhost(1413)') CLUSTER(CL2) REPLACE
DEFINE CHANNEL(CL2.QM1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1411)') CLUSTER(CL2) REPLACE
```

7. 將本端佇列 Q1 新增至叢集 CL2 中的佇列管理程式 QM3。

```
*... On QM3
DEFINE QLOCAL(Q1) CLUSTER(CL2) REPLACE
```

8. 將叢集佇列管理程式別名 Q1A 新增至閘道佇列管理程式。

```
*... On QM1
DEFINE QALIAS(Q1A) CLUSNL(ALL) TARGET(Q1) TARGTYPE(Queue) DEFBIND(NOTFIXED) REPLACE
```

註: 在 QM1 以外的任何其他佇列管理程式上使用佇列管理程式別名的應用程式，在開啟別名佇列時必須指定 DEFBIND(NOTFIXED)。DEFBIND 指定當應用程式開啟佇列時，是否修正訊息標頭中的遞送資訊。如果設為預設值 OPEN，則訊息會遞送至 Q1@QM1。Q1@QM1 不存在，因此來自其他佇列管理程式的訊息會結束在無法傳送郵件的佇列上。透過將佇列屬性設為 DEFBIND(NOTFIXED)，應用程式 (例如 **amqspout**，其預設為佇列設定 DEFBIND) 會以正確的方式運作。

9. 將所有叢集佇列管理程式的叢集佇列管理程式別名定義新增至閘道佇列管理程式 QM1。

```
*... On QM1
DEFINE QREMOTE(QM2) RNAME(' ') RQMNAME(QM2) CLUSNL(ALL) REPLACE
DEFINE QREMOTE(QM3) RNAME(' ') RQMNAME(QM3) CLUSNL(ALL) REPLACE
```

提示: 閘道佇列管理程式上的佇列管理程式別名定義會參照另一個叢集中的佇列管理程式; 請參閱 [叢集佇列管理程式別名](#)。

下一步

1. 使用佇列別名定義 Q1A，將訊息從 QM2 傳送至 QM3 上的 Q1，以測試佇列別名定義。

a. 在 QM2 上執行範例程式 **amqsput**，以放置訊息。

```
C:\IBM\MQ>amqsput Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
Sample request message from QM2 to Q1 using Q1A
```

```
Sample AMQSPUT0 end
```

b. 執行範例程式 **amqsget**，以從 QM3 上的 Q1 取得訊息

```
C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1A>
no more messages
Sample AMQSGET0 end
```

2. 傳送要求訊息，並在暫時動態回覆佇列上接收回覆訊息，以測試佇列管理程式別名定義。

此圖顯示回覆訊息回到暫時動態佇列 (稱為 RQ) 所採用的路徑。連接至 QM3 的伺服器應用程式會使用佇列管理程式名稱 QM2 開啟回覆佇列。佇列管理程式名稱 QM2 在 QM1 上定義為叢集佇列管理程式別名。QM3 會將回覆訊息遞送至 QM1。QM1 會將訊息遞送至 QM2。

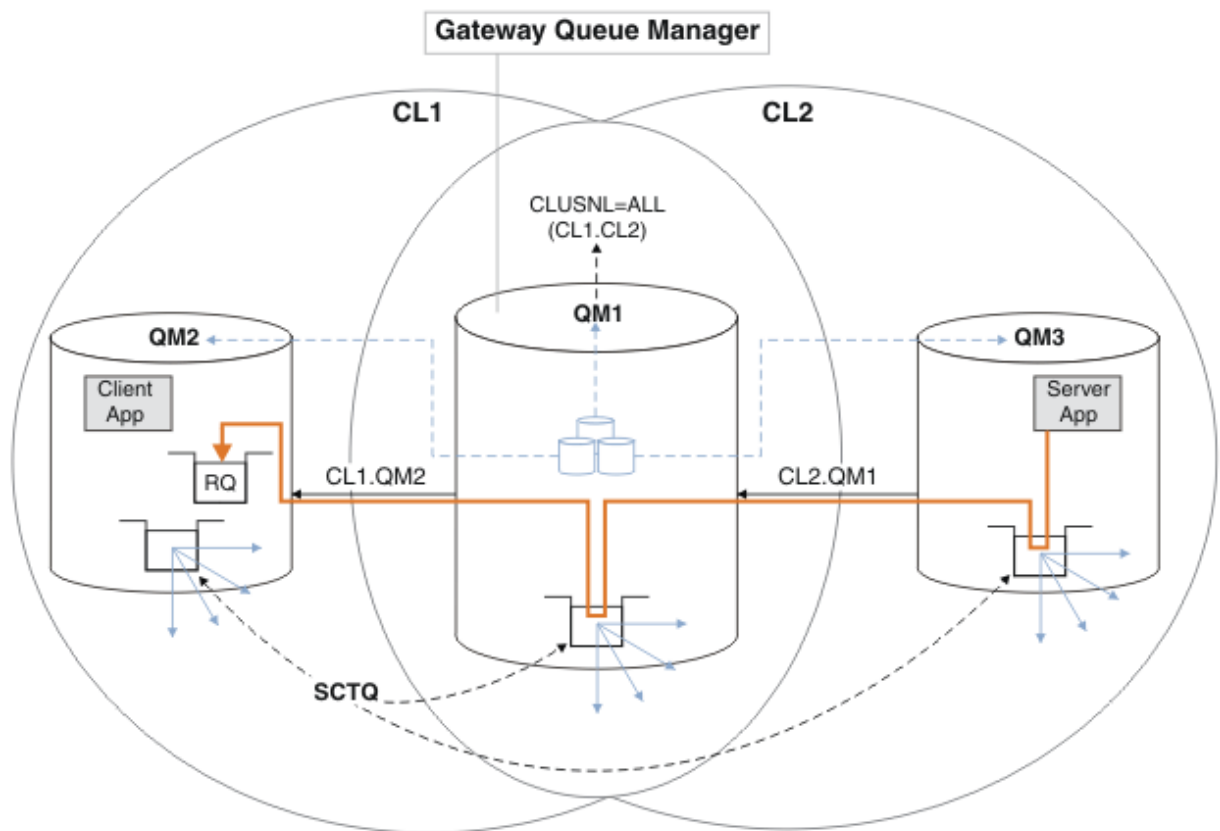


圖 38: 使用佇列管理程式別名將回覆訊息傳回至不同的叢集

遞送的運作方式如下。每個叢集中的每個佇列管理程式在 QM1 上都有一個佇列管理程式別名定義。別名會在所有叢集中形成叢集。從每一個別名到佇列管理程式的灰色虛線箭頭顯示每一個佇列管理程式別名都已解析為至少其中一個叢集中的實際佇列管理程式。在此情況下，QM2 別名同時在叢集 CL1 和 CL2 中形成叢集，並解析為 CL1 中的實際佇列管理程式 QM2。伺服器應用程式會使用回覆目的地佇列名稱 RQ 及回覆目的地佇列管理程式名稱 QM2 來建立回覆訊息。訊息會遞送至 QM1，因為佇列管理程式別名定義 QM2 定義在叢集 CL2 中的 QM1 上，且佇列管理程式 QM2 不在叢集 CL2 中。因為訊息無法傳送至目標佇列管理程式，所以會傳送至具有別名定義的佇列管理程式。

QM1 會將訊息放在 QM1 上的叢集傳輸佇列中，以便傳送至 QM2。QM1 會將訊息遞送至 QM2，因為 QM1 for QM2 上的佇列管理程式別名定義會將 QM2 定義為實際目標佇列管理程式。定義不是循環，因為別名定義只能參照實際定義；別名不能指向本身。實際定義由 QM1 解析，因為 QM1 和 QM2 都位於相同的叢集 CL1 中。QM1 會從 CL1 的儲存庫中找出 QM2 的連線資訊，並將訊息遞送至 QM2。若要讓 QM1 重新遞送訊息，伺服器應用程式必須已開啟回覆佇列，並將選項 DEFBIND 設為 MQBND_BIND_NOT_FIXED。如果伺服器應用程式已使用選項 MQBND_BIND_ON_OPEN 開啟回覆佇列，則訊息不會重新遞送，且會在無法傳送的郵件佇列上結束。

- a. 在 QM3 上建立具有觸發程式的叢集要求佇列。

```
*... On QM3
DEFINE QLOCAL(QR) CLUSTER(CL2) TRIGGER INITQ(SYSTEM.DEFAULT.INITIATION.QUEUE)
PROCESS(ECHO) REPLACE
```

- b. 在閘道佇列管理程式 QM1 上建立 QR 的叢集佇列別名定義。

```
*... On QM1
DEFINE QALIAS(QRA) CLUSNL(ALL) TARGET(QR) TARGTYPE(QUEUE) DEFBIND(NOTFIXED) REPLACE
```

- c. 建立程序定義以在 QM3 上啟動範例回應程式 **amqsech**。

```
*... On QM3
DEFINE PROCESS(ECHO) APPLICID(AMQSECH) REPLACE
```

- d. 在 QM2 上為範例程式 **amqsreq** 建立模型佇列，以建立暫時動態回覆佇列。

```
*... On QM2
DEFINE QMODEL(SYSTEM.SAMPLE.REPLY) REPLACE
```

- e. 測試佇列管理程式別名定義，方法是使用佇列別名定義 QRA，將要求從 QM2 傳送至 QM3 上的 QR。

- i) 在 QM3 上執行觸發監視器程式。

```
runmqtrm -m QM3
```

輸出為

```
C:\IBM\MQ>runmqtrm -m QM3
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
01/02/2012 16:17:15: WebSphere MQ trigger monitor started.
```

```
-----
01/02/2012 16:17:15: Waiting for a trigger message
```

- ii) 在 QM2 上執行範例程式 **amqsreq**，以放置要求並等待回覆。

```
C:\IBM\MQ>amqsreq QRA QM2
Sample AMQSREQ0 start
server queue is QRA
replies to 4F2961C802290020
A request message from QM2 to QR on QM3

response <A request message from QM2 to QR on QM3>
no more replies
Sample AMQSREQ0 end
```

相關概念

第 135 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 238 頁的『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。

相關工作

第 165 頁的『[將佇列管理程式新增至叢集: 個別傳輸佇列](#)』

請遵循下列指示，將佇列管理程式新增至您建立的叢集。叢集佇列和主題的訊息會使用多個叢集傳輸佇列來傳送。

第 182 頁的『[使用閘道佇列管理程式建立兩個重疊的叢集](#)』

遵循作業中的指示，使用閘道佇列管理程式來建構重疊叢集。使用叢集作為下列範例的起點，將訊息從一個應用程式隔離至叢集中的其他應用程式。

第 167 頁的『[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](#)』

修改使用閘道佇列管理程式之重疊叢集的配置。從閘道佇列管理程式將修改訊息傳送至應用程式之後，不會使用與其他叢集訊息相同的傳輸佇列或通道。解決方案使用叢集佇列遠端定義，以及個別傳送端通道和傳輸佇列。

第 187 頁的『[將預設值變更為個別叢集傳輸佇列以隔離訊息資料流量](#)』

您可以變更佇列管理程式將叢集佇列或主題的訊息儲存在傳輸佇列上的預設方式。變更預設值可讓您隔離閘道佇列管理程式上的叢集訊息。

第 170 頁的『[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](#)』

修改使用閘道佇列管理程式之重疊叢集的配置。從閘道佇列管理程式將修改訊息傳送至應用程式之後，不會使用與其他叢集訊息相同的傳輸佇列或通道。將修改訊息傳送至應用程式之後，解決方案會使用額外的叢集傳輸佇列，將訊息資料流量分隔至叢集中的單一佇列管理程式。

第 173 頁的『[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](#)』

修改使用閘道佇列管理程式之重疊叢集的配置。從閘道佇列管理程式將修改訊息傳送至應用程式之後，不會使用與其他叢集訊息相同的傳輸佇列或通道。將修改訊息傳送至應用程式之後，解決方案會使用其他叢集，將訊息與特定叢集佇列隔離。

第 241 頁的『[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。配置多個叢集傳輸佇列適用於 z/OS 以外的平台。

將預設值變更為個別叢集傳輸佇列以隔離訊息資料流量

您可以變更佇列管理程式將叢集佇列或主題的訊息儲存在傳輸佇列上的預設方式。變更預設值可讓您隔離閘道佇列管理程式上的叢集訊息。

開始之前

1. 閘道佇列管理程式必須位於 Version 7.5 或更新版本，以及 z/OS 以外的平台上。
2. 遵循第 182 頁的『[使用閘道佇列管理程式建立兩個重疊的叢集](#)』中的步驟，在該作業中建構第 183 頁的圖 37 中顯示的重疊叢集。

關於這項作業

若要使用多個叢集佇列來實作架構，您的閘道佇列管理程式必須位於 Version 7.5 或更新版本上。您只要變更閘道佇列管理程式上的預設叢集傳輸佇列類型，即可使用多個叢集傳輸佇列。將 QM1 上佇列管理程式屬性 DEFCLXQ 的值從 SCTQ 變更為 CHANNEL; 請參閱第 188 頁的圖 39。此圖顯示一個訊息流程。對於流向其他佇列管理程式或其他叢集的流程，佇列管理程式會建立其他永久動態叢集傳輸佇列。每一個叢集傳送端通道會從不同的叢集傳輸佇列傳送訊息。

除非您第一次將閘道佇列管理程式連接至叢集，否則變更不會立即生效。此作業包括管理現有配置之變更的一般案例的步驟。若要設定佇列管理程式在第一次加入叢集時使用個別叢集傳輸佇列，請參閱第 165 頁的『[將佇列管理程式新增至叢集: 個別傳輸佇列](#)』。

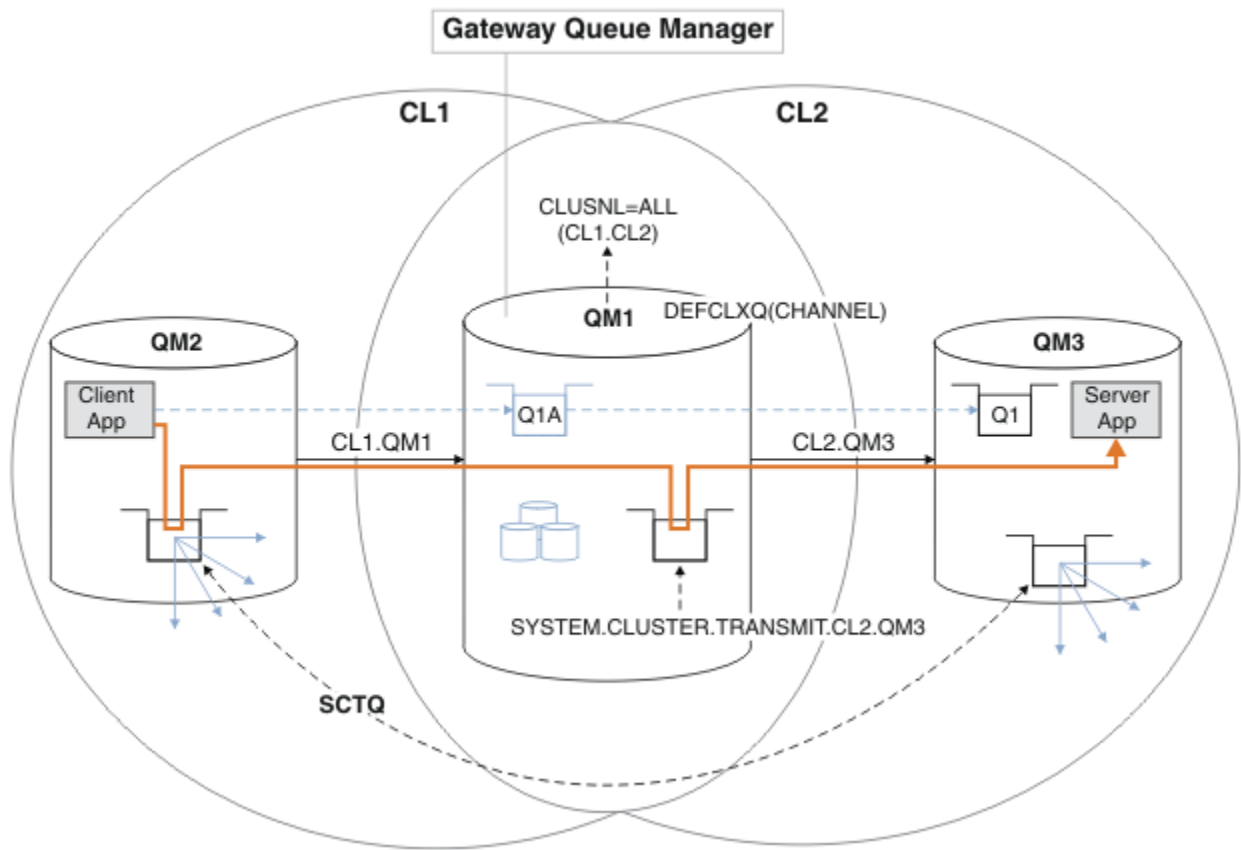


圖 39: 將主從架構應用程式部署至閘道佇列管理程式上具有個別叢集傳輸佇列的 hub 及輪輻架構。

程序

1. 將閘道佇列管理程式變更為使用個別叢集傳輸佇列。

```
*... On QM1
ALTER QMGR DEFCLXQ(CHANNEL)
```

2. 切換至個別叢集傳輸佇列。

任何未執行的叢集傳送端通道，會在下次啟動時切換至使用個別叢集傳輸佇列。

若要切換執行中通道，請重新啟動佇列管理程式，或遵循下列步驟：

- a) 列出正在使用 SYSTEM.CLUSTER.TRANSMIT.QUEUE 執行的叢集傳送端通道。

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ EQ 'SYSTEM.CLUSTER.TRANSMIT.QUEUE')
```

回應是通道狀態報告的清單：

```
AMQ8417: Display Channel Status details.
CHANNEL(CL1.QM2)           CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1412))  CURRENT
RQMNAME(QM2)              STATUS(RUNNING)
SUBSTATE(MQGET)           XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL(CL2.QM3)           CHLTYPE(CLUSSDR)
CONNAME(127.0.0.1(1413))  CURRENT
RQMNAME(QM3)              STATUS(RUNNING)
SUBSTATE(MQGET)           XMITQ(SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
```

```

CHANNEL (CL2.QM5)           CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1415)) CURRENT
RQMNAME (QM5)              STATUS (RUNNING)
SUBSTATE (MQGET)           XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL (CL1.QM4)           CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1414)) CURRENT
RQMNAME (QM4)              STATUS (RUNNING)
SUBSTATE (MQGET)           XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)

```

b) 停止執行中的通道

針對清單中的每一個通道，執行下列指令：

```

*... On QM1
STOP CHANNEL(ChannelName)

```

其中 *ChannelName* 是每一個 CL1.QM2, CL1.QM4, CL1.QM3, CL1.QM5。

回應是接受指令：

```
AMQ8019: Stop WebSphere MQ channel accepted.
```

c) 監視哪些通道已停止

```

*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ EQ 'SYSTEM.CLUSTER.TRANSMIT.QUEUE')

```

回應是仍在執行中的通道及已停止的通道清單：

```

AMQ8417: Display Channel Status details.
CHANNEL (CL1.QM2)           CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1412)) CURRENT
RQMNAME (QM2)              STATUS (STOPPED)
SUBSTATE ( )                XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL (CL2.QM3)           CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1413)) CURRENT
RQMNAME (QM3)              STATUS (STOPPED)
SUBSTATE ( )                XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL (CL2.QM5)           CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1415)) CURRENT
RQMNAME (QM5)              STATUS (STOPPED)
SUBSTATE ( )                XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)
AMQ8417: Display Channel Status details.
CHANNEL (CL1.QM4)           CHLTYPE (CLUSSDR)
CONNNAME (127.0.0.1(1414)) CURRENT
RQMNAME (QM4)              STATUS (STOPPED)
SUBSTATE ( )                XMITQ (SYSTEM.CLUSTER.TRANSMIT.QUEUE)

```

d) 啟動每一個已停止的通道。

針對所有執行中的通道執行此步驟。如果通道未停止，您可以使用 **FORCE** 選項重新執行 **STOP CHANNEL** 指令。設定 **FORCE** 選項的範例如下：通道未停止，且您無法重新啟動另一個佇列管理程式來同步化通道。

```

*... On QM1
START CHANNEL (CL2.QM5)

```

回應是接受指令：

```
AMQ8018: Start WebSphere MQ channel accepted.
```

e) 監視正在切換的傳輸佇列。

監視閘道佇列管理程式錯誤日誌，以取得訊息 "AMQ7341 通道 CL2.QM3 是 SYSTEM.CLUSTER.TRANSMIT.QUEUE / CL2.QM3"。

- f) 檢查是否不再使用 SYSTEM.CLUSTER.TRANSMIT.QUEUE

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ EQ 'SYSTEM.CLUSTER.TRANSMIT.QUEUE')
DISPLAY QUEUE(SYSTEM.CLUSTER.TRANSMIT.QUEUE) CURDEPTH
```

回應是通道狀態報告的清單，以及 SYSTEM.CLUSTER.TRANSMIT.QUEUE 的深度：

```
AMQ8420: Channel Status not found.
AMQ8409: Display Queue details.
      QUEUE(SYSTEM.CLUSTER.TRANSMIT.QUEUE)      TYPE(QLOCAL)
      CURDEPTH(0)
```

- g) 監視哪些通道已啟動

```
*... On QM1
DISPLAY CHSTATUS(*) WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
```

回應是通道清單，在此情況下，已使用新的預設叢集傳輸佇列執行：

```
AMQ8417: Display Channel Status details.
      CHANNEL(CL1.QM2)                CHLTYPE(CLUSSDR)
      CONNAME(127.0.0.1(1412))        CURRENT
      RQMNAME(QM2)                    STATUS(RUNNING)
      SUBSTATE(MQGET)
      XMITQ(SYSTEM.CLUSTER.TRANSMIT.CL1.QM2)
AMQ8417: Display Channel Status details.
      CHANNEL(CL2.QM3)                CHLTYPE(CLUSSDR)
      CONNAME(127.0.0.1(1413))        CURRENT
      RQMNAME(QM3)                    STATUS(RUNNING)
      SUBSTATE(MQGET)
      XMITQ(SYSTEM.CLUSTER.TRANSMIT.CL2.QM3)
AMQ8417: Display Channel Status details.
      CHANNEL(CL2.QM5)                CHLTYPE(CLUSSDR)
      CONNAME(127.0.0.1(1415))        CURRENT
      RQMNAME(QM5)                    STATUS(RUNNING)
      SUBSTATE(MQGET)
      XMITQ(SYSTEM.CLUSTER.TRANSMIT.CL2.QM5)
AMQ8417: Display Channel Status details.
      CHANNEL(CL1.QM4)                CHLTYPE(CLUSSDR)
      CONNAME(127.0.0.1(1414))        CURRENT
      RQMNAME(QM4)                    STATUS(RUNNING)
      SUBSTATE(MQGET)
      XMITQ(SYSTEM.CLUSTER.TRANSMIT.CL1.QM4)
```

下一步

1. 測試自動定義的叢集傳輸佇列，方法是將訊息從 QM2 傳送至 QM3 上的 Q1，並使用佇列別名定義來解析佇列名稱 Q1A
 - a. 在 QM2 上執行範例程式 **amqsput**，以放置訊息。

```
C:\IBM\MQ>amqsput Q1A QM2
Sample AMQSPUT0 start
target queue is Q1A
Sample request message from QM2 to Q1 using Q1A

Sample AMQSPUT0 end
```
 - b. 執行範例程式 **amqsget**，以從 QM3 上的 Q1 取得訊息


```
C:\IBM\MQ>amqsget Q1 QM3
Sample AMQSGET0 start
message <Sample request message from QM2 to Q1 using Q1A>
no more messages
Sample AMQSGET0 end
```

2. 請考量是否重新配置安全，方法是在產生叢集佇列訊息的佇列管理程式上配置叢集佇列的安全。

相關概念

[第 135 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 238 頁的『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。

相關工作

[第 165 頁的『將佇列管理程式新增至叢集: 個別傳輸佇列』](#)

請遵循下列指示，將佇列管理程式新增至您建立的叢集。叢集佇列和主題的訊息會使用多個叢集傳輸佇列來傳送。

[第 182 頁的『使用閘道佇列管理程式建立兩個重疊的叢集』](#)

遵循作業中的指示，使用閘道佇列管理程式來建構重疊叢集。使用叢集作為下列範例的起點，將訊息從一個應用程式隔離至叢集中的其他應用程式。

[第 167 頁的『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』](#)

修改使用閘道佇列管理程式之重疊叢集的配置。從閘道佇列管理程式將修改訊息傳送至應用程式之後，不會使用與其他叢集訊息相同的傳輸佇列或通道。解決方案使用叢集佇列遠端定義，以及個別傳送端通道和傳輸佇列。

[第 187 頁的『將預設值變更為個別叢集傳輸佇列以隔離訊息資料流量』](#)

您可以變更佇列管理程式將叢集佇列或主題的訊息儲存在傳輸佇列上的預設方式。變更預設值可讓您隔離閘道佇列管理程式上的叢集訊息。

[第 170 頁的『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』](#)

修改使用閘道佇列管理程式之重疊叢集的配置。從閘道佇列管理程式將修改訊息傳送至應用程式之後，不會使用與其他叢集訊息相同的傳輸佇列或通道。將修改訊息傳送至應用程式之後，解決方案會使用額外的叢集傳輸佇列，將訊息資料流量分隔至叢集中的單一佇列管理程式。

[第 173 頁的『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』](#)

修改使用閘道佇列管理程式之重疊叢集的配置。從閘道佇列管理程式將修改訊息傳送至應用程式之後，不會使用與其他叢集訊息相同的傳輸佇列或通道。將修改訊息傳送至應用程式之後，解決方案會使用其他叢集，將訊息與特定叢集佇列隔離。

[第 241 頁的『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。配置多個叢集傳輸佇列適用於 z/OS 以外的平台。

從佇列管理程式移除叢集佇列

停用多倫多的 INVENTQ 佇列。將所有庫存訊息傳送至紐約，並刪除空的多倫多 INVENTQ 佇列。

開始之前

註: 如果要將叢集的變更延伸到整個叢集，至少必須一律有一個完整儲存庫可用。在啟動這項作業之前，請確定您的儲存庫可供使用。

測試情境:

- 已依照第 180 頁的『新增管理佇列的佇列管理程式』中的說明來設定 INVENTORY 叢集。它包含四個佇列管理程式。LONDON 和 NEWYORK 都保留完整儲存庫。PARIS 和 TORONTO 保留局部儲存庫。庫存應用程式在紐約和多倫多的系統上執行，並由 INVENTQ 佇列上的訊息到達所驅動。
- 由於工作量減少，您不再想要在多倫多執行庫存應用程式。您想要停用佇列管理程式 TORONTO 所管理的 INVENTQ 佇列，並讓 TORONTO 將訊息饋送至 NEWYORK 中的 INVENTQ 佇列。
- 所有四個系統之間都存在網路連線功能。
- 網路通訊協定是 TCP。

關於這項作業

請遵循下列步驟來移除叢集佇列。

程序

1. 指出佇列不再可用。

若要從叢集中移除佇列，請從本端佇列定義中移除叢集名稱。變更 TORONTO 上的 INVENTQ，使其無法從叢集的其餘部分存取：

```
ALTER QLOCAL(INVENTQ) CLUSTER('')
```

2. 請檢查佇列是否不再可用。

在完整儲存庫佇列管理程式 (LONDON 或 NEWYORK) 上，發出下列指令來檢查佇列是否不再由佇列管理程式 TORONTO 管理：

```
DIS QCLUSTER (INVENTQ)
```

如果 ALTER 指令已順利完成，則結果中不會列出 TORONTO。

3. 停用佇列。

停用位於 TORONTO 的 INVENTQ 佇列，以便沒有進一步的訊息可以寫入其中：

```
ALTER QLOCAL(INVENTQ) PUT(DISABLED)
```

現在，使用 MQOO_BIND_ON_OPEN 傳送至此佇列的訊息會移至無法傳送郵件的佇列。您需要停止所有應用程式將訊息明確放入此佇列管理程式上的佇列。

4. 監視佇列直到它是空的。

使用 DISPLAY QUEUE 指令並指定屬性 IPPROCS、OPPROCS 及 CURDEPTH 來監視佇列，或在 IBM i 上使用 **WRKMQMSTS** 指令。當輸入及輸出處理程序的數目及佇列的現行深度都為零時，佇列是空的。

5. 監視通道以確保沒有不確定的訊息。

為了確保通道 INVENTORY.TORONTO 上沒有不確定的訊息，請監視其他每一個佇列管理程式上稱為 INVENTORY.TORONTO 的叢集傳送端通道。從每一個佇列管理程式發出 DISPLAY CHSTATUS 指令，並指定 INDOUBT 參數：

```
DISPLAY CHSTATUS(INVENTORY.TORONTO) INDOUBT
```

如果有任何不確定的訊息，您必須先解決這些訊息，才能繼續進行。例如，您可以嘗試發出 RESOLVE 通道指令，或停止並重新啟動通道。

6. 刪除本端佇列。

當您滿意沒有其他訊息可遞送至 TORONTO 上的庫存應用程式時，您可以刪除佇列：

```
DELETE QLOCAL(INVENTQ)
```

7. 您現在可以從多倫多的系統中移除庫存應用程式
移除應用程式可避免重複，並節省系統上的空間。

結果

此作業所設定的叢集類似於前一個作業所設定的叢集。差異在於 INVENTQ 佇列不再適用於佇列管理程式 TORONTO。

當您在步驟 1 中讓佇列無法運作時，TORONTO 佇列管理程式會將訊息傳送至兩個完整儲存庫佇列管理程式。它已通知他們狀態變更。完整儲存庫佇列管理程式會將此資訊傳遞給叢集中已要求更新 INVENTQ 相關資訊的其他佇列管理程式。

當佇列管理程式將訊息放置在 INVENTQ 佇列上時，更新的局部儲存庫會指出 INVENTQ 佇列只能在 NEWYORK 佇列管理程式上使用。訊息會傳送至 NEWYORK 佇列管理程式。

下一步

在此作業中，只有一個佇列要移除，且只有一個叢集要從中移除。

假設有許多佇列參照包含許多叢集名稱的名單。例如，TORONTO 佇列管理程式不僅可以管理 INVENTQ，還可以管理 PAYROLLQ、SALESQ 和 PURCHASESQ。TORONTO 可讓這些佇列在所有適當的叢集、INVENTORY、PAYROLL、SALES 及 PURCHASES 中可供使用。定義 TORONTO 佇列管理程式上叢集名稱的名單：

```
DEFINE NAMELIST(TOROLIST)
  DESCR('List of clusters TORONTO is in')
  NAMES(INVENTORY, PAYROLL, SALES, PURCHASES)
```

將名單新增至每一個佇列定義：

```
DEFINE QLOCAL(INVENTQ) CLUSNL(TOROLIST)
DEFINE QLOCAL(PAYROLLQ) CLUSNL(TOROLIST)
DEFINE QLOCAL(SALESQ) CLUSNL(TOROLIST)
DEFINE QLOCAL(PURCHASESQ) CLUSNL(TOROLIST)
```

現在假設您想要從 SALES 叢集中移除所有那些佇列，因為 SALES 作業將由 PURCHASES 作業接管。您只需要變更 TOROLIST 名稱清單，即可從中移除 SALES 叢集的名稱。

如果您要從名單中的其中一個叢集移除單一佇列，請建立一個名單，其中包含剩餘的叢集名稱清單。然後變更佇列定義，以使用新的名稱清單。若要從 INVENTORY 叢集中移除 PAYROLLQ，請執行下列動作：

1. 建立名單：

```
DEFINE NAMELIST(TOROSHORTLIST)
  DESCR('List of clusters TORONTO is in other than INVENTORY')
  NAMES(PAYROLL, SALES, PURCHASES)
```

2. 變更 PAYROLLQ 佇列定義：

```
ALTER QLOCAL(PAYROLLQ) CLUSNL(TOROSHORTLIST)
```

將完整儲存庫移至另一個佇列管理程式

將完整儲存庫從一個佇列管理程式移至另一個佇列管理程式，從第二個儲存庫所保留的資訊來建置新的儲存庫。

開始之前

註：如果要將叢集的變更延伸到整個叢集，至少必須一律有一個完整儲存庫可用。在啟動這項作業之前，請確定您的儲存庫可供使用。

測試情境：

- 已依照第 164 頁的『將佇列管理程式加入叢集中』中的說明來設定 INVENTORY 叢集。
- 基於商業原因，您現在想要從佇列管理程式 LONDON 中移除完整儲存庫，並將它取代為佇列管理程式 PARIS 中的完整儲存庫。NEWYORK 佇列管理程式會繼續保留完整儲存庫。

關於這項作業

請遵循下列步驟，將完整儲存庫移至另一個佇列管理程式。

程序

1. 變更 PARIS，使它成為完整儲存庫佇列管理程式。

在 PARIS 上，發出下列指令：

```
ALTER QMGR REPOS(INVENTORY)
```

2. 在 PARIS 上新增 CLUSSDR 通道

PARIS 目前有一個叢集傳送端通道指向 LONDON。LONDON 不再保留叢集的完整儲存庫。PARIS 必須具有指向 NEWYORK 的新叢集傳送端通道，其中現在保留了其他完整儲存庫。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)  
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)  
DESCR('Cluster-sender channel from PARIS to repository at NEWYORK')
```

3. 在 NEWYORK 上定義指向 PARIS 的 CLUSSDR 通道

目前 NEWYORK 有一個指向 LONDON 的叢集傳送端通道。既然其他完整儲存庫已移至 PARIS，您需要在 NEWYORK 中新增指向 PARIS 的叢集傳送端通道。

```
DEFINE CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSSDR) TRPTYPE(TCP)  
CONNAME(PARIS.CHSTORE.COM) CLUSTER(INVENTORY)  
DESCR('Cluster-sender channel from NEWYORK to repository at PARIS')
```

當您將叢集傳送端通道新增至 PARIS 時，PARIS 會從 NEWYORK 瞭解叢集。它會使用 NEWYORK 中的資訊來建置自己的完整儲存庫。

4. 確認佇列管理程式 PARIS 現在具有完整儲存庫

檢查佇列管理程式 PARIS 是否已從佇列管理程式 NEWYORK 上的完整儲存庫建置自己的完整儲存庫。發出下列指令：

```
DIS QCLUSTER(*) CLUSTER (INVENTORY)  
DIS CLUSQMGR(*) CLUSTER (INVENTORY)
```

請檢查這些指令是否顯示此叢集中與 NEWYORK 上相同資源的詳細資料。

註：如果佇列管理程式 NEWYORK 無法使用，則此資訊建置無法完成。在作業完成之前，請勿移至下一步。

5. 變更 LONDON 上的佇列管理程式定義

最後，變更位於 LONDON 的佇列管理程式，使其不再保留叢集的完整儲存庫。在 LONDON 上，發出下列指令：

```
ALTER QMGR REPOS(' ')
```

佇列管理程式不再接收任何叢集資訊。在 30 天之後，儲存在其完整儲存庫中的資訊會到期。佇列管理程式 LONDON 現在會建置自己的局部儲存庫。

6. 移除或變更任何未完成的定義。

當您確定叢集的新安排如預期般運作時，請移除或變更不再正確的手動定義 CLUSSDR 定義。

- 在 PARIS 佇列管理程式上，您必須停止並刪除 LONDON 的叢集傳送端通道，然後發出 start channel 指令，讓叢集可以重新使用自動通道：

```
STOP CHANNEL(INVENTORY.LONDON)  
DELETE CHANNEL(INVENTORY.LONDON)  
START CHANNEL(INVENTORY.LONDON)
```

- 在 NEWYORK 佇列管理程式上，您必須停止並刪除 LONDON 的叢集傳送端通道，然後發出 start channel 指令，讓叢集可以重新使用自動通道：

```
STOP CHANNEL(INVENTORY.LONDON)
DELETE CHANNEL(INVENTORY.LONDON)
START CHANNEL(INVENTORY.LONDON)
```

- 將叢集中所有指向 LONDON 的其他叢集傳送端通道，取代為指向 NEWYORK 或 PARIS 的通道。在這個小例子中，沒有其他例子。若要檢查是否有任何您忘記的其他項目，請從每一個佇列管理程式中指定 TYPE(CLUSSDR)，發出 DISPLAY CHANNEL 指令。例如：

```
DISPLAY CHANNEL(*) TYPE(CLUSSDR)
```

請務必在將完整儲存庫從 LONDON 移至 PARIS 之後儘快執行此作業。在執行此作業之前，已手動定義 CLUSSDR 通道 (名稱為 INVENTORY.LONDON) 的佇列管理程式可能會使用此通道來傳送資訊要求。

在 LONDON 不再是完整儲存庫之後，如果收到這類要求，它會將錯誤訊息寫入其佇列管理程式錯誤日誌。下列範例顯示在 LONDON 上可能看到哪些錯誤訊息：

- AMQ9428: Unexpected publication of a cluster queue object received
- AMQ9432: Query received by a non-repository queue manager

佇列管理程式 LONDON 不會回應資訊要求，因為它不再是完整儲存庫。從 LONDON 要求資訊的佇列管理程式必須依賴 NEWYORK 來取得叢集資訊，直到將其手動定義的 CLUSSDR 定義更正為指向 PARIS 為止。長期而言，不得容忍此狀況作為有效的配置。

結果

第 195 頁的圖 40 顯示此作業所設定的叢集。

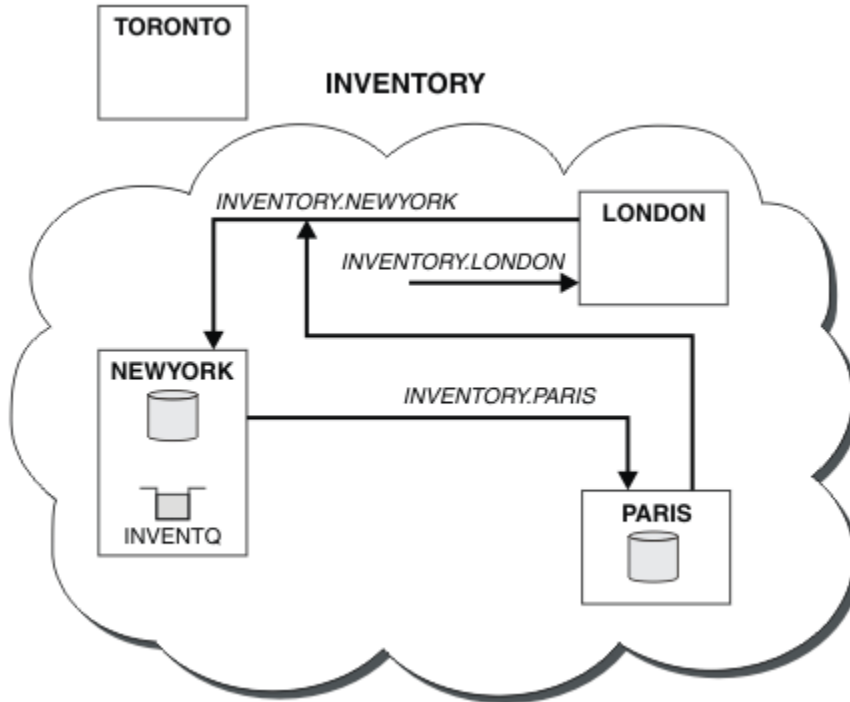


圖 40: 具有完整儲存庫的 INVENTORY 叢集已移至 PARIS

將現有網路轉換為叢集

將現有的分散式佇列網路轉換為叢集，並新增額外的佇列管理程式以增加容量。

開始之前

在第 154 頁的『設定新的叢集』中，透過第 193 頁的『將完整儲存庫移至另一個佇列管理程式』，您已建立並延伸新的叢集。接下來兩項作業會探索不同的方法：將現有的佇列管理程式網路轉換成叢集。

註：如果要將叢集的變更延伸到整個叢集，至少必須一律有一個完整儲存庫可用。在啟動這項作業之前，請確定您的儲存庫可供使用。

測試情境：

- IBM WebSphere MQ 網路已就緒，可連接連鎖商店的全國性分支。它具有中心分支結構：所有佇列管理程式都連接至一個中央佇列管理程式。中央佇列管理程式位於執行庫存應用程式的系統上。應用程式是由 INVENTQ 佇列上的訊息到達所驅動，每一個佇列管理程式都有遠端佇列定義。

第 196 頁的圖 41 中說明此網路。

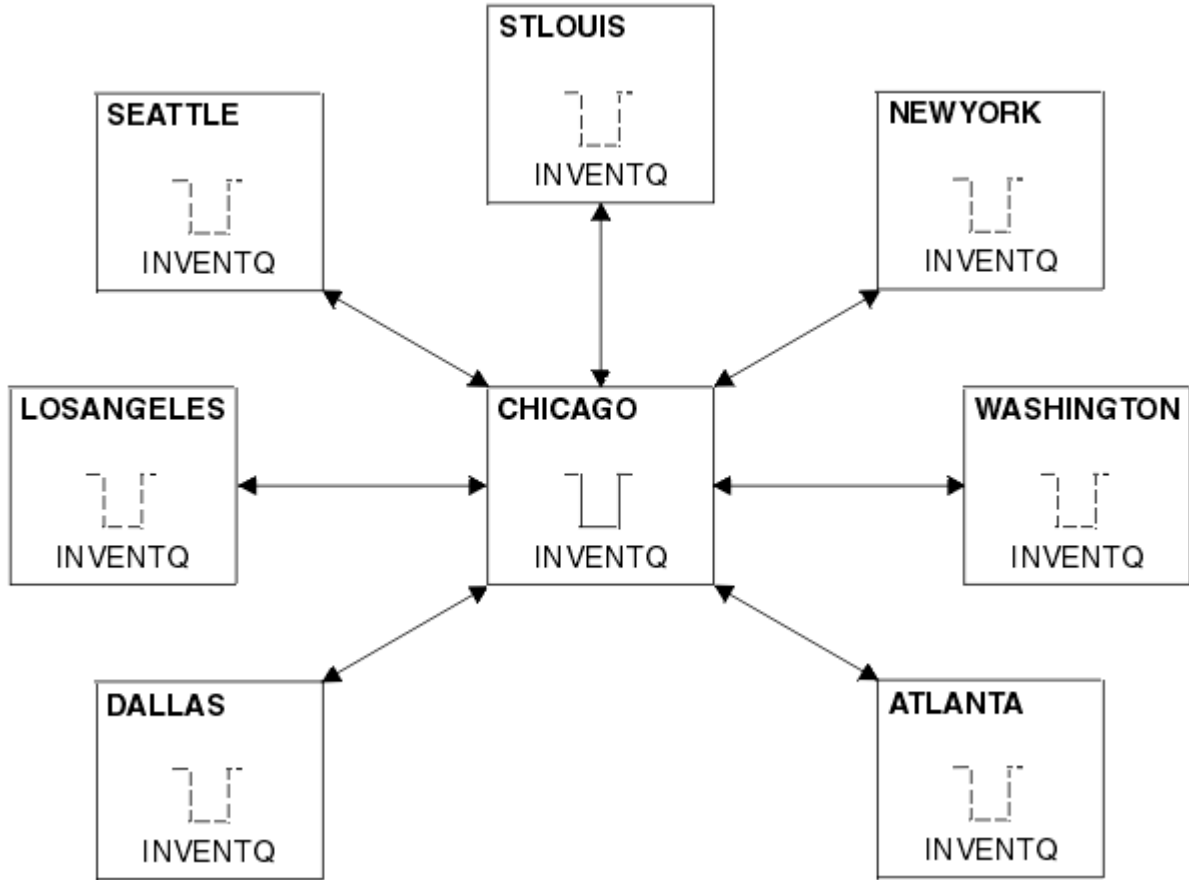


圖 41: 中心及分支網路

- 為了簡化管理，您將將此網路轉換為叢集，並在中央站台建立另一個佇列管理程式以共用工作量。

叢集名稱為 CHNSTORE。

註：選取叢集名稱 CHNSTORE 以容許使用 *cluster-name.queue-manager* 格式的名稱來建立叢集接收端通道名稱，這些名稱不超過 20 個字元的長度上限，例如 CHNSTORE.WASHINGTON。

- 這兩個中央佇列管理程式都會管理完整儲存庫，且可供庫存應用程式存取。
- 庫存應用程式是由訊息到達由任一中央佇列管理程式所管理的 INVENTQ 佇列所驅動。
- 庫存應用程式是唯一平行執行且可由多個佇列管理程式存取的應用程式。所有其他應用程式繼續像之前一樣執行。
- 所有分支都具有與兩個中央佇列管理程式的網路連線功能。
- 網路通訊協定是 TCP。

關於這項作業

請遵循下列步驟，將現有網路轉換為叢集。

程序

1. 請檢閱庫存應用程式，以取得訊息親緣性。

在繼續之前，請確定應用程式可以處理訊息親緣性。訊息親緣性是兩個應用程式之間交換的交談式訊息之間的關係，其中訊息必須由特定佇列管理程式或以特定順序處理。如需訊息親緣性的相關資訊，請參閱：第 233 頁的『處理訊息親緣性』

2. 變更兩個中央佇列管理程式，使它們成為完整儲存庫佇列管理程式。

兩個佇列管理程式 CHICAGO 及 CHICAGO2 位於此網路的中心。您已決定將與鏈結儲存庫叢集相關聯的所有活動集中於這兩個佇列管理程式。除了庫存應用程式和 INVENTQ 佇列的定義，您也希望這些佇列管理程式管理叢集的兩個完整儲存庫。在兩個佇列管理程式的每一個上，發出下列指令：

```
ALTER QMGR REPOS(CHNSTORE)
```

3. 在每一個佇列管理程式上定義 CLUSRCVR 通道。

在叢集中的每一個佇列管理程式上，定義叢集接收端通道及叢集傳送端通道。您先定義哪個通道並不重要。

製作 CLUSRCVR 定義，將每一個佇列管理程式、其網址及其他資訊通告給叢集。例如，在佇列管理程式 ATLANTA 上：

```
DEFINE CHANNEL(CHNSTORE.ATLANTA) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)  
CONNNAME(ATLANTA.CHSTORE.COM) CLUSTER(CHNSTORE)  
DESCR('Cluster-receiver channel')
```

4. 在每一個佇列管理程式上定義 CLUSSDR 通道

在每一個佇列管理程式上建立 CLUSSDR 定義，以將該佇列管理程式鏈結至其中一個或其他完整儲存庫佇列管理程式。例如，您可以將 ATLANTA 鏈結至 CHICAGO2：

```
DEFINE CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR) TRPTYPE(TCP)  
CONNNAME(CHICAGO2.CHSTORE.COM) CLUSTER(CHNSTORE)  
DESCR('Cluster-sender channel to repository queue manager')
```

5. 在 CHICAGO2 上安裝庫存應用程式。

您已在佇列管理程式 CHICAGO 上具有庫存應用程式。現在您需要在佇列管理程式 CHICAGO2 上建立此應用程式的副本。

6. 在中央佇列管理程式上定義 INVENTQ 佇列。

在 CHICAGO 上，修改佇列 INVENTQ 的本端佇列定義，使佇列可供叢集使用。發出指令：

```
ALTER QLOCAL(INVENTQ) CLUSTER(CHNSTORE)
```

在 CHICAGO2 上，建立相同佇列的定義：

```
DEFINE QLOCAL(INVENTQ) CLUSTER(CHNSTORE)
```

在 z/OS 上，您可以使用 **CSQUTIL** 的 **COMMAND** 函數的 **MAKEDEF** 選項，在 INVENTQ on CHICAGO 的 CHICAGO2 上建立確切副本。

當您建立這些定義時，會將訊息傳送至位於 CHICAGO 和 CHICAGO2 的完整儲存庫，並更新其中的資訊。當佇列管理程式將訊息放置到 INVENTQ 時，會從完整儲存庫中發現訊息有目的地可供選擇。

7. 請檢查是否已傳播叢集變更。

請檢查您在前一個步驟中建立的定義是否已透過叢集傳播。在完整儲存庫佇列管理程式上發出下列指令：

```
DIS QCLUSTER(INVENTQ)
```

新增交互連接的叢集

新增與現有叢集共用部分佇列管理程式的新叢集。

開始之前

註:

1. 如果要將叢集的變更延伸到整個叢集，至少必須一律有一個完整儲存庫可用。在啟動這項作業之前，請確定您的儲存庫可供使用。
2. 在啟動此作業之前，請先檢查佇列名稱衝突，並瞭解結果。您可能需要重新命名佇列或設定佇列別名，才能繼續進行。

測試情境:

- 已依照第 195 頁的『將現有網路轉換為叢集』中的說明來設定 WebSphere MQ 叢集。
- 將實作為 MAILORDER 的新叢集。此叢集包含四個位於 CHNSTORE 叢集中的佇列管理程式; CHICAGO、CHICAGO2、SEATTLE 及 ATLANTA，以及兩個額外的佇列管理程式; HARTFORD 及 OMAHA。MAILORDER 應用程式在連接至佇列管理程式 OMAHA 的 Omaha 系統上執行。它由叢集中的其他佇列管理程式驅動，將訊息放置在 MORDERQ 佇列上。
- MAILORDER 叢集的完整儲存庫是在兩個佇列管理程式 CHICAGO 和 CHICAGO2 上維護。
- 網路通訊協定是 TCP。

關於這項作業

請遵循下列步驟來新增交互連接的叢集。

程序

1. 建立叢集名稱的名稱清單。

現在，位於 CHICAGO 和 CHICAGO2 的完整儲存庫佇列管理程式將同時保留 CHNSTORE 和 MAILORDER 這兩個叢集的完整儲存庫。首先，建立包含叢集名稱的名單。在 CHICAGO 和 CHICAGO2 上定義名稱清單，如下所示:

```
DEFINE NAMELIST(CHAINMAIL)
  DESCR('List of cluster names')
  NAMES(CHNSTORE, MAILORDER)
```

2. 變更兩個佇列管理程式定義。

現在，請變更位於 CHICAGO 及 CHICAGO2 的兩個佇列管理程式定義。目前這些定義顯示佇列管理程式保留叢集 CHNSTORE 的完整儲存庫。變更該定義，以顯示佇列管理程式會保留 CHAINMAIL 名單中列出之所有叢集的完整儲存庫。變更 CHICAGO 及 CHICAGO2 佇列管理程式定義:

```
ALTER QMGR REPOS(' ') REPOSNL(CHAINMAIL)
```

3. 變更 CHICAGO 和 CHICAGO2 上的 CLUSRCVR 通道。

CHICAGO 及 CHICAGO2 中的 CLUSRCVR 通道定義顯示這些通道可在叢集 CHNSTORE 中使用。您需要變更叢集接收端定義，以顯示通道可供 CHAINMAIL 名單中列出的所有叢集使用。變更位於 CHICAGO 的叢集接收端定義:

```
ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSRCVR)
  CLUSTER(' ') CLUSNL(CHAINMAIL)
```

在 CHICAGO2 中，輸入下列指令:

```
ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSRCVR)
  CLUSTER(' ') CLUSNL(CHAINMAIL)
```

4. 變更 CHICAGO 及 CHICAGO2 上的 CLUSSDR 通道。

變更兩個 CLUSSDR 通道定義以新增名稱清單。在 CHICAGO 中，輸入下列指令：

```
ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

在 CHICAGO2 中，輸入下列指令：

```
ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSSDR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

5. 在 SEATTLE 和 ATLANTA 上建立名單。

因為 SEATTLE 和 ATLANTA 將成為多個叢集的成員，所以您必須建立包含叢集名稱的名稱清單。在 SEATTLE 和 ATLANTA 上定義名稱清單，如下所示：

```
DEFINE NAMELIST(CHAINMAIL)
DESCR('List of cluster names')
NAMES(CHNSTORE, MAILORDER)
```

6. 變更 SEATTLE 和 ATLANTA 上的 CLUSRCVR 通道。

SEATTLE 及 ATLANTA 中的 CLUSRCVR 通道定義顯示這些通道可在叢集 CHNSTORE 中使用。變更叢集接收通道定義，以顯示這些通道可供 CHAINMAIL 名單中列出的所有叢集使用。在 SEATTLE 中，輸入下列指令：

```
ALTER CHANNEL(CHNSTORE.SEATTLE) CHLTYPE(CLUSRCVR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

在 ATLANTA 中，輸入下列指令：

```
ALTER CHANNEL(CHNSTORE.ATLANTA) CHLTYPE(CLUSRCVR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

7. 變更 SEATTLE 及 ATLANTA 上的 CLUSSDR 通道。

變更兩個 CLUSSDR 通道定義以新增名稱清單。在 SEATTLE 中，輸入下列指令：

```
ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSSDR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

在 ATLANTA 中，輸入下列指令：

```
ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR)
CLUSTER(' ') CLUSNL(CHAINMAIL)
```

8. 在 HARTFORD 及 OMAHA 上定義 CLUSRCVR 及 CLUSSDR 通道。

在兩個新的佇列管理程式 HARTFORD 和 OMAHA 上，定義叢集接收端和叢集傳送端通道。您建立定義的順序並不重要。在 HARTFORD 中，輸入：

```
DEFINE CHANNEL(MAILORDER.HARTFORD) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(HARTFORD.CHSTORE.COM) CLUSTER(MAILORDER)
DESCR('Cluster-receiver channel for HARTFORD')

DEFINE CHANNEL(MAILORDER.CHICAGO) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(CHICAGO.CHSTORE.COM) CLUSTER(MAILORDER)
DESCR('Cluster-sender channel from HARTFORD to repository at CHICAGO')
```

在 OMAHA 中，輸入：

```
DEFINE CHANNEL(MAILORDER.OMAHA) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(OMAHA.CHSTORE.COM) CLUSTER(MAILORDER)
DESCR('Cluster-receiver channel for OMAHA')

DEFINE CHANNEL(MAILORDER.CHICAGO) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(CHICAGO.CHSTORE.COM) CLUSTER(MAILORDER)
DESCR('Cluster-sender channel from OMAHA to repository at CHICAGO')
```

9. 在 OMAHA 上定義 MORDERQ 佇列。

完成此作業的最終步驟是 MORDERQ 在佇列管理程式上 OMAHA 定義佇列。在 OMAHA 中，輸入：

```
DEFINE QLOCAL(MORDERQ) CLUSTER(MAILORDER)
```

10. 請檢查是否已傳播叢集變更。

請檢查您以先前步驟所建立的定義是否已透過叢集傳播。在完整儲存庫佇列管理程式上發出下列指令：

```
DIS QCLUSTER (MORDERQ)  
DIS CLUSQMGR
```

11.

結果

此作業所設定的叢集顯示在 [第 201 頁的圖 42](#) 中。

現在我們有兩個重疊的叢集。這兩個叢集的完整儲存庫都保留在 CHICAGO 和 CHICAGO2。在 OMAHA 上執行的郵件訂單應用程式與在 CHICAGO 上執行的庫存應用程式無關。不過，CHNSTORE 叢集中的部分佇列管理程式也位於 MAILORDER 叢集中，因此它們可以將訊息傳送至任一應用程式。在執行此作業以重疊兩個叢集之前，請注意佇列名稱衝突的可能性。

假設在叢集 CHNSTORE 中的 NEWYORK 及叢集 MAILORDER 中的 OMAHA 上，有一個稱為 ACCOUNTQ 的佇列。如果您重疊叢集，然後 SEATTLE 上的應用程式將訊息放置到佇列 ACCOUNTQ，則訊息可以跳至 ACCOUNTQ 的任一實例。

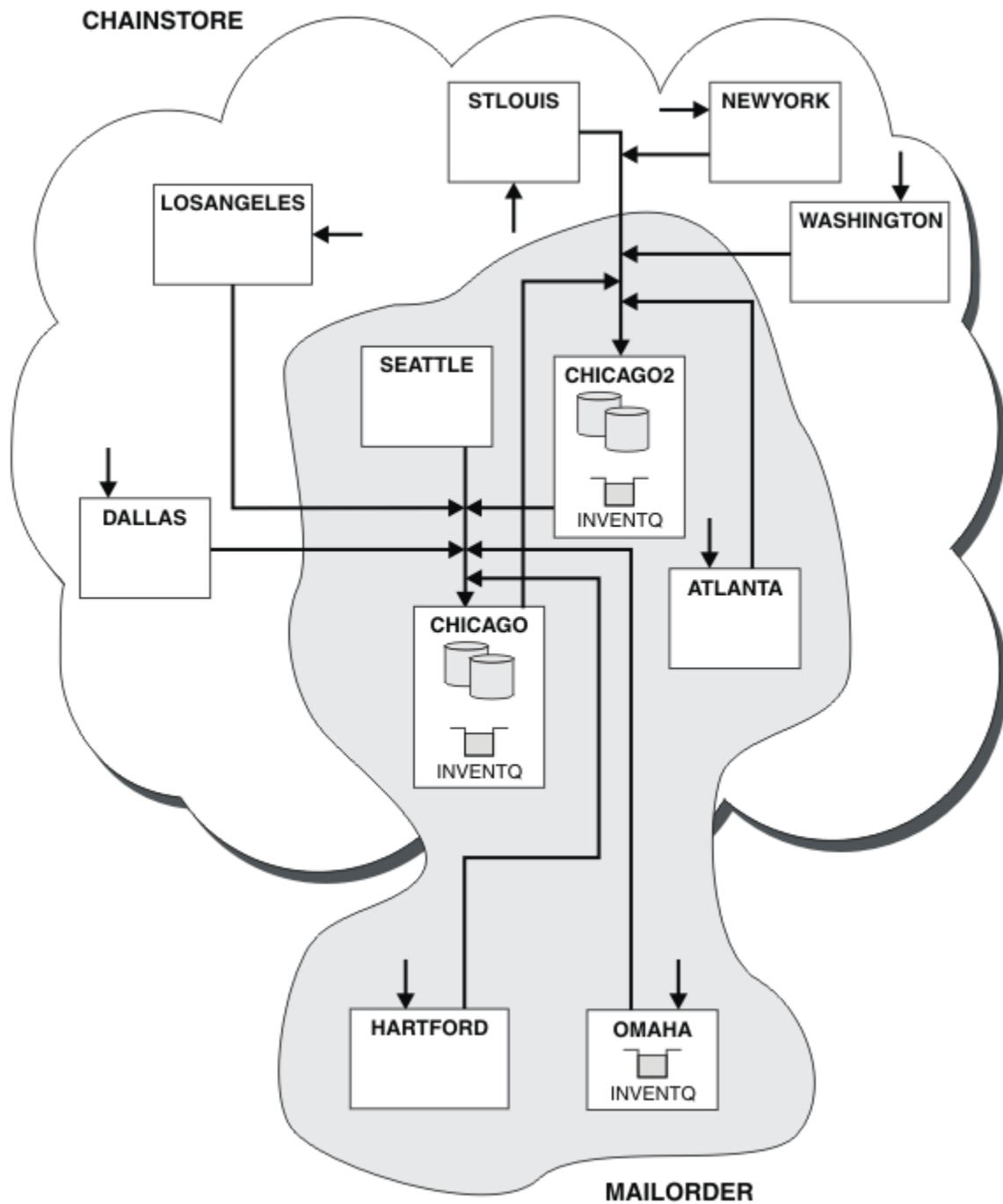


圖 42: 交互連接叢集

下一步

假設您決定將 MAILORDER 叢集與 CHNSTORE 叢集合併，以形成一個稱為 CHNSTORE 的大型叢集。

如果要將 MAILORDER 叢集與 CHNSTORE 叢集合併，以便 CHICAGO 和 CHICAGO2 保留完整儲存庫，請執行下列動作：

- 變更 CHICAGO 及 CHICAGO2 的佇列管理程式定義，移除指定名稱清單 (CHAINMAIL) 的 REPOSNL 屬性，並將它取代為指定叢集名稱 (CHNSTORE) 的 REPOS 屬性。例如：

```
ALTER QMGR(CHICAGO) REPOSNL(' ') REPOS(CHNSTORE)
```

- 在 MAILORDER 叢集中的每一個佇列管理程式上，變更所有通道定義及佇列定義，以將 CLUSTER 屬性值從 MAILORDER 變更為 CHNSTORE。例如，在 HARTFORD 中，輸入：

```
ALTER CHANNEL(MAILORDER.HARTFORD) CLUSTER(CHNSTORE)
```

在 OMAHA 中，輸入：

```
ALTER QLOCAL(MORDERQ) CLUSTER(CHNSTORE)
```

- 變更指定叢集名單 CHAINMAIL(即位於 CHICAGO、CHICAGO2、SEATTLE 及 ATLANTA 的 CLUSRCVR 及 CLUSSDR 通道定義)的所有定義，以改為指定叢集 CHNSTORE。

從這個範例中，您可以看到使用名稱清單的優點。您可以變更名稱清單 CHAINMAIL 的值，而不是變更 CHICAGO 及 CHICAGO2 的佇列管理程式定義。同樣地，您可以透過變更名稱清單來達到所需結果，而不是變更 CHICAGO、CHICAGO2、SEATTLE 及 ATLANTA 中的 CLUSRCVR 及 CLUSSDR 通道定義。

移除叢集網路

從網路中移除叢集，並還原分散式佇列配置。

開始之前

註：如果要將叢集的變更延伸到整個叢集，至少必須一律有一個完整儲存庫可用。在啟動這項作業之前，請確定您的儲存庫可供使用。

測試情境：

- 已依照第 195 頁的『將現有網路轉換為叢集』中的說明來設定 IBM WebSphere MQ 叢集。
- 現在將從系統中移除此叢集。佇列管理程式網路會像實作叢集之前一樣繼續運作。

關於這項作業

請遵循下列步驟來移除叢集網路。

程序

1. 從 CHNSTORE 叢集中移除叢集佇列。

在 CHICAGO 及 CHICAGO2 上，修改佇列 INVENTQ 的本端佇列定義，以從叢集中移除佇列。發出指令：

```
ALTER QLOCAL(INVENTQ) CLUSTER('')
```

當您變更佇列時，完整儲存庫中的資訊會更新並延伸到整個叢集。使用 MQOO_BIND_NOT_FIXED 的作用中應用程式，以及使用已使用 DEFBIND(NOTFIXED)定義佇列的 MQOO_BIND_AS_Q_DEF 的應用程式，會在下一次嘗試 MQPUT 或 MQPUT1 呼叫時失敗。會傳回原因碼 MQRC_UNKNOWN_OBJECT_NAME。

您不需要先執行步驟 1，但如果沒有，請在步驟 4 之後執行它。

2. 停止所有可存取叢集佇列的應用程式。

停止所有可存取叢集佇列的應用程式。如果沒有，當您在步驟 5 中重新整理叢集時，部分叢集資訊可能會保留在本端佇列管理程式上。當所有應用程式都已停止且叢集通道已斷線時，會移除此資訊。

3. 從完整儲存庫佇列管理程式中移除儲存庫屬性。

在 CHICAGO 及 CHICAGO2 上，修改佇列管理程式定義以移除儲存庫屬性。若要執行此動作，請發出下列指令：

```
ALTER QMGR REPOS('')
```


佇列管理程式會通知叢集中的其他佇列管理程式，它們不再保留完整儲存庫。當其他佇列管理程式收到這項資訊時，您會看到一則訊息，指出完整儲存庫已結束。您也會看到一則以上訊息，指出叢集 CHNSTORE 不再有任何可用的儲存庫。

4. 移除叢集通道。

在 CHICAGO 上，移除叢集通道：

```
ALTER CHANNEL(CHNSTORE.CHICAGO2) CHLTYPE(CLUSSDR) CLUSTER(' ')
ALTER CHANNEL(CHNSTORE.CHICAGO) CHLTYPE(CLUSRCVR) CLUSTER(' ')
```

註：請務必先發出 CLUSSDR 指令，然後再發出 CLUSRCVR 指令。請不要先發出 CLUSRCVR 指令，然後再發出 CLUSSDR 指令。這樣做會建立狀態為 STOPPED 的不確定通道。然後，您需要發出 START CHANNEL 指令，以回復已停止的通道；例如 START CHANNEL(CHNSTORE.CHICAGO)。

您會看到訊息，指出叢集 CHNSTORE 沒有儲存庫。

如果您未如步驟 1 所述移除叢集佇列，請立即執行。

5. 停止叢集通道。

在 CHICAGO 上，使用下列指令停止叢集通道：

```
STOP CHANNEL(CHNSTORE.CHICAGO2)
STOP CHANNEL(CHNSTORE.CHICAGO)
```

6. 針對叢集中的每一個佇列管理程式，重複步驟 4 和 5。

7. 停止叢集通道，然後從每一個佇列管理程式中移除叢集通道及叢集佇列的所有定義。

8. 選擇性的：清除佇列管理程式所保留的快取叢集資訊。

雖然佇列管理程式不再是叢集的成員，但它們會保留叢集相關資訊的快取副本。如果您要移除此資料，請參閱作業第 206 頁的『將佇列管理程式還原至其叢集前狀態』。

9. 取代 INVENTQ 的遠端佇列定義

為了讓網路可以繼續運作，請取代每個佇列管理程式上 INVENTQ 的遠端佇列定義。

10. 清理叢集。

刪除任何不再需要的佇列或通道定義。

從叢集中移除佇列管理程式

在佇列管理程式可以與叢集中至少一個完整儲存庫正常通訊的情況下，從叢集中移除佇列管理程式。

開始之前

對於至少有一個完整儲存庫可用，且正在移除的佇列管理程式可以聯絡的實務範例，此方法是最佳作法。此方法涉及最少的人為介入，並容許佇列管理程式從叢集協議受控制的撤銷。如果要移除的佇列管理程式無法聯絡完整儲存庫，請參閱第 205 頁的『從叢集中移除佇列管理程式：替代方法』。

在從叢集中移除佇列管理程式之前，您必須確定佇列管理程式不再管理叢集所需的資源：

- 如果佇列管理程式管理完整儲存庫，請從第 193 頁的『將完整儲存庫移至另一個佇列管理程式』完成步驟 1-4。
- 如果佇列管理程式管理叢集佇列，請從第 191 頁的『從佇列管理程式移除叢集佇列』完成步驟 1-7。
- 如果佇列管理程式管理叢集主題，請刪除主題（例如，使用 `DELETE TOPIC` 指令），或將它們移至其他主機。

註：如果您從叢集中移除佇列管理程式，且佇列管理程式仍管理叢集主題，則佇列管理程式可能會繼續嘗試將發佈資訊遞送至叢集中剩餘的佇列管理程式，直到刪除主題為止。

關於這項作業

此範例作業會從 INVENTORY 叢集中移除佇列管理程式 LONDON。INVENTORY 叢集設定如第 164 頁的『將佇列管理程式加入叢集中』中所述，並如第 191 頁的『從佇列管理程式移除叢集佇列』中所述進行修改。

從叢集中移除佇列管理程式的程序比新增佇列管理程式的程序更複雜。

當佇列管理程式加入叢集時，叢集的現有成員不會知道新的佇列管理程式，因此不會與它互動。必須在結合的佇列管理程式上建立新的傳送端和接收端通道，它才能連接完整儲存庫。

從叢集中移除佇列管理程式時，連接至佇列管理程式的應用程式可能正在使用物件 (例如叢集中其他位置所管理的佇列)。此外，連接至叢集中其他佇列管理程式的應用程式可能正在使用目標佇列管理程式上管理的物件。由於這些應用程式，現行佇列管理程式可能會建立其他傳送端通道，以建立與叢集成員的通訊，但它用來加入叢集的完整儲存庫除外。叢集中的每個佇列管理程式都有一份說明其他叢集成員的資料快取副本。這可能包括正在移除的項目。

程序

1. 在佇列管理程式 LONDON 上，變更手動定義的叢集接收端通道，以從叢集中移除它們：

```
ALTER CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSRCVR) CLUSTER(' ')
```

2. 在佇列管理程式 LONDON 上，變更手動定義的叢集傳送端通道，以從叢集中移除它們：

```
ALTER CHANNEL(INVENTORY.PARIS) CHLTYPE(CLUSSDR) CLUSTER(' ')
```

叢集中的其他佇列管理程式會瞭解此佇列管理程式及其叢集資源不再是叢集的一部分。

3. 監視佇列管理程式 LONDON 上的叢集傳輸佇列，直到沒有訊息正在等待流向叢集中的任何完整儲存庫為止。

```
DISPLAY CHSTATUS(INVENTORY.LONDON) XQMSGSA
```

如果訊息仍留在傳輸佇列中，請先判斷訊息未傳送至 PARIS 和 NEWYORK 完整儲存庫的原因，再繼續進行。

結果

佇列管理程式 LONDON 不再是叢集的一部分。不過，它仍然可以作為獨立佇列管理程式來運作。

下一步

在叢集的其餘成員上發出下列指令，即可確認這些變更的結果：

```
DISPLAY CLUSQMGR(LONDON)
```

佇列管理程式會繼續顯示，直到自動定義的叢集傳送端通道停止為止。您可以發出下列指令，等待發生此情況，或繼續監視作用中實例：

```
DISPLAY CHANNEL(INVENTORY.LONDON)
```

當您確定不再將任何其他訊息遞送至這個佇列管理程式時，您可以在叢集的其餘成員上發出下列指令，停止叢集傳送端通道至 LONDON：

```
STOP CHANNEL(INVENTORY.LONDON) STATUS(INACTIVE)
```

在變更延伸到整個叢集之後，且不再將其他訊息遞送到這個佇列管理程式，請停止並刪除 LONDON 上的 CLUSRCVR 通道：

```
STOP CHANNEL(INVENTORY.LONDON)  
DELETE CHANNEL(INVENTORY.LONDON)
```

移除的佇列管理程式可以稍後再新增回叢集中，如第 164 頁的『將佇列管理程式加入叢集中』中所述。已移除的佇列管理程式會繼續快取叢集其餘成員的知識，最多 90 天。如果您不想等到這個快取到期，可以依照第 206 頁的『將佇列管理程式還原至其叢集前狀態』中的說明來強制移除它。

從叢集中移除佇列管理程式: 替代方法

從叢集中移除佇列管理程式，在由於嚴重系統或配置問題，導致佇列管理程式無法與叢集中的任何完整儲存庫進行通訊的情況下。

開始之前

這個從叢集中手動移除佇列管理程式的替代方法會停止並刪除所有將已移除佇列管理程式鏈結至叢集的叢集通道，並強制從叢集中移除佇列管理程式。在所移除佇列管理程式無法與任何完整儲存庫通訊的情況下，會使用此方法。例如，這可能是因為佇列管理程式已停止運作，或因為佇列管理程式與叢集之間的通訊長時間失敗。否則，請使用最常用的方法: [第 203 頁的『從叢集中移除佇列管理程式』](#)。

在從叢集中移除佇列管理程式之前，您必須確定佇列管理程式不再管理叢集所需的資源:

- 如果佇列管理程式管理完整儲存庫，請從 [第 193 頁的『將完整儲存庫移至另一個佇列管理程式』](#) 完成步驟 1-4。
- 如果佇列管理程式管理叢集佇列，請從 [第 191 頁的『從佇列管理程式移除叢集佇列』](#) 完成步驟 1-7。
- 如果佇列管理程式管理叢集主題，請刪除主題 (例如，使用 `DELETE TOPIC` 指令)，或將它們移至其他主機。

註: 如果您從叢集中移除佇列管理程式，且佇列管理程式仍管理叢集主題，則佇列管理程式可能會繼續嘗試將發佈資訊遞送至叢集中剩餘的佇列管理程式，直到刪除主題為止。

關於這項作業

此範例作業會從 `INVENTORY` 叢集中移除佇列管理程式 `LONDON`。`INVENTORY` 叢集設定如 [第 164 頁的『將佇列管理程式加入叢集中』](#) 中所述，並如 [第 191 頁的『從佇列管理程式移除叢集佇列』](#) 中所述進行修改。

從叢集中移除佇列管理程式的程序比新增佇列管理程式的程序更複雜。

當佇列管理程式加入叢集時，叢集的現有成員不會知道新的佇列管理程式，因此不會與它互動。必須在結合的佇列管理程式上建立新的傳送端和接收端通道，它才能連接完整儲存庫。

從叢集中移除佇列管理程式時，連接至佇列管理程式的應用程式可能正在使用物件 (例如叢集中其他位置所管理的佇列)。此外，連接至叢集中其他佇列管理程式的應用程式可能正在使用目標佇列管理程式上管理的物件。由於這些應用程式，現行佇列管理程式可能會建立其他傳送端通道，以建立與叢集成員的通訊，但它用來加入叢集的完整儲存庫除外。叢集中的每個佇列管理程式都有一份說明其他叢集成員的資料快取副本。這可能包括正在移除的項目。

當無法等待佇列管理程式循序離開叢集時，此程序可能適用於緊急狀況。

程序

1. 停止用來與叢集中其他佇列管理程式通訊的所有通道。使用 `MODE (FORCE)` 來停止佇列管理程式 `LONDON` 上的 `CLUSRCVR` 通道。否則，您可能需要等待傳送端佇列管理程式停止通道:

```
STOP CHANNEL (INVENTORY.LONDON) MODE (FORCE)
STOP CHANNEL (INVENTORY.TORONTO)
STOP CHANNEL (INVENTORY.PARIS)
STOP CHANNEL (INVENTORY.NEWYORK)
```

2. 監視佇列管理程式 `LONDON` 上的通道狀態，直到通道停止為止:

```
DISPLAY CHSTATUS (INVENTORY.LONDON)
DISPLAY CHSTATUS (INVENTORY.TORONTO)
DISPLAY CHSTATUS (INVENTORY.PARIS)
DISPLAY CHSTATUS (INVENTORY.NEWYORK)
```

在通道停止之後，不會再與叢集中其他佇列管理程式之間傳送其他應用程式訊息。

3. 刪除佇列管理程式 `LONDON` 上手動定義的叢集通道:

```
DELETE CHANNEL (INVENTORY.NEWYORK)
DELETE CHANNEL (INVENTORY.TORONTO)
```

4. 叢集中的其餘佇列管理程式仍會保留已移除佇列管理程式的知識，且可能會繼續向它傳送訊息。如果要從其餘佇列管理程式清除知識，請在其中一個完整儲存庫上，從叢集重設已移除的佇列管理程式：

```
RESET CLUSTER(INVENTORY) ACTION(FORCEREMOVE) QMNAME(LONDON) QUEUES(YES)
```

如果叢集中可能有另一個佇列管理程式與已移除的佇列管理程式同名，請指定已移除佇列管理程式的 **QMID**。

結果

佇列管理程式 LONDON 不再是叢集的一部分。不過，它仍然可以作為獨立佇列管理程式來運作。

下一步

在叢集的其餘成員上發出下列指令，即可確認這些變更的結果：

```
DISPLAY CLUSQMGR(LONDON)
```

佇列管理程式會繼續顯示，直到自動定義的叢集傳送端通道停止為止。您可以發出下列指令，等待發生此情況，或繼續監視作用中實例：

```
DISPLAY CHANNEL(INVENTORY.LONDON)
```

在將變更延伸到整個叢集，且不再將任何訊息遞送到這個佇列管理程式之後，請刪除 LONDON 上的 CLUSRCVR 通道：

```
DELETE CHANNEL(INVENTORY.LONDON)
```

移除的佇列管理程式可以稍後再新增回叢集中，如第 164 頁的『將佇列管理程式加入叢集中』中所述。已移除的佇列管理程式會繼續快取叢集其餘成員的知識，最多 90 天。如果您不想等到這個快取到期，可以依照第 206 頁的『將佇列管理程式還原至其叢集前狀態』中的說明來強制移除它。

將佇列管理程式還原至其叢集前狀態

從叢集中移除佇列管理程式時，它會保留其餘叢集成員的知識。此知識最終到期並自動刪除。不過，如果您偏好立即刪除它，則可以使用本主題中的步驟。

開始之前

假設佇列管理程式已從叢集中移除，且不再於叢集中執行任何工作。例如，其佇列不再接收來自叢集的訊息，且沒有應用程式正在等待訊息到達這些佇列。

重要：如果您從叢集中移除佇列管理程式，並使用 REPOS (YES) 來重新整理它，則只要變更其 CLUSRCVR 的 CLUSTER 屬性，就無法重新將它新增回叢集。將其 CLUSRCVR 的 CLUSTER 屬性變更為非空白 (即叢集名稱) 之後，您還需要使用 REPOS (NO) 來發出重新整理叢集，此時 CLUSRCVR 上的內部序號將會更新。然後佇列管理程式會順利將自己重新引進「完整儲存庫」及其餘叢集成員。(請注意，在提供正確的叢集名稱給 CLUSRCVR 通道之後，必須執行 REPOS (NO) 版本的指令。)

此限制僅適用於 IBM WebSphere MQ Version 7.5。

關於這項作業

從叢集中移除佇列管理程式時，它會保留其餘叢集成員的知識最多 90 天。這可能會有系統好處，尤其是當佇列管理程式快速重新加入叢集時。當此知識最終到期時，會自動刪除它。不過，有一些您可能偏好手動刪除此資訊的原因。例如：

- 您可能想要確認已停止此佇列管理程式上先前使用叢集資源的每個應用程式。在剩餘叢集成員的知識到期之前，任何這類應用程式都會繼續寫入傳輸佇列。刪除叢集知識之後，當這類應用程式嘗試使用叢集資源時，系統會產生錯誤訊息。
- 當您顯示佇列管理程式的狀態資訊時，您可能不希望看到其餘叢集成員的相關到期資訊。

此作業使用 INVENTORY 叢集作為範例。LONDON 佇列管理程式已從 INVENTORY 叢集中移除，如第 203 頁的『從叢集中移除佇列管理程式』中所述。若要刪除叢集其餘成員的知識，請在 LONDON 佇列管理程式上發出下列指令。

程序

1. 從此佇列管理程式中移除叢集中其他佇列管理程式的所有記憶體:

```
REFRESH CLUSTER(INVENTORY) REPOS(YES)
```

2. 監視佇列管理程式，直到所有叢集資源都消失為止:

```
DISPLAY CLUSQMGR(*) CLUSTER(INVENTORY)
DISPLAY QCLUSTER(*) CLUSTER(INVENTORY)
DISPLAY TOPIC(*) CLUSTER(INVENTORY)
```

相關概念

叢集

第 135 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。

第 137 頁的『叢集的元件』

叢集由佇列管理程式、叢集儲存庫、叢集通道及叢集佇列組成。

第 154 頁的『管理 IBM WebSphere MQ 叢集』

您可以建立、延伸及維護 IBM WebSphere MQ 叢集。

維護佇列管理程式

從叢集暫停及回復佇列管理程式，以執行維護。

關於這項作業

有時，您可能需要在屬於叢集的佇列管理程式上執行維護。例如，您可能需要備份其佇列中的資料，或將修正程式套用至軟體。如果佇列管理程式管理任何佇列，則必須暫停其活動。當維護完成時，可以回復其活動。

程序

1. 發出 SUSPEND QMGR **runmqsc** 指令來暫停佇列管理程式:

```
SUSPEND QMGR CLUSTER(SALES)
```

SUSPEND **runmqsc** 指令會通知 SALES 叢集中的佇列管理程式，此佇列管理程式已暫停。

SUSPEND QMGR 指令的目的只是建議其他佇列管理程式盡可能避免將訊息傳送給這個佇列管理程式。這並不表示佇列管理程式已停用。某些必須由這個佇列管理程式處理的訊息仍會傳送給它，例如，當這個佇列管理程式是叢集佇列的唯一主機時。

當佇列管理程式暫停時，工作量管理常式會避免傳送訊息給它。必須由該佇列管理程式處理的訊息包括本端佇列管理程式所傳送的訊息。

WebSphere MQ 使用工作量平衡演算法來判斷哪些目的地適合，而不是儘可能選取本端佇列管理程式。

- a) 在 SUSPEND QMGR 指令上使用 FORCE 選項，強制暫停佇列管理程式:

```
SUSPEND QMGR CLUSTER(SALES) MODE(FORCE)
```


MODE (FORCE) 會強制停止來自叢集中其他佇列管理程式的所有入埠通道。如果未指定 MODE (FORCE)，則會套用預設 MODE (QUIESCE)。

2. 執行任何必要的維護作業。
3. 發出 RESUME QMGR **runmqsc** 指令來回復佇列管理程式:

```
RESUME QMGR CLUSTER(SALES)
```

結果

RESUME **runmqsc** 指令會通知完整儲存庫，佇列管理程式會再次可用。完整儲存庫佇列管理程式會將此資訊傳送給其他已要求更新此佇列管理程式相關資訊的佇列管理程式。

維護叢集傳輸佇列

盡一切努力保持叢集傳輸佇列可用。它們對於叢集的效能至關重要。

開始之前

- 請確定叢集傳輸佇列未滿。
- 請小心不要發出 ALTER **runmqsc** 指令，將它設為「已停用」或「意外停用」。
- 確保儲存在上的叢集傳輸佇列中的媒體不會變滿。

重新整理叢集佇列管理程式

您可以使用 REFRESH CLUSTER 指令，從本端儲存庫移除自動定義的通道及自動定義的叢集物件。不會遺失任何訊息。

開始之前

IBM 支援中心可能會要求您使用該指令。請不要在未仔細考量的情況下使用指令。例如，對於大型叢集，使用 **REFRESH CLUSTER** 指令可能會對進行中的叢集造成干擾，此後每隔 27 天，叢集物件會自動將狀態更新傳送至所有相關的佇列管理程式。請參閱第 257 頁的『叢集作業：使用 REFRESH CLUSTER 最佳作法』。

關於這項作業

佇列管理程式可以在叢集中進行全新啟動。在一般情況下，您不需要使用 REFRESH CLUSTER 指令。

程序

從佇列管理程式發出 REFRESH CLUSTER **MQSC** 指令，以從本端儲存庫移除自動定義的叢集佇列管理程式及佇列物件。

指令只會移除參照其他佇列管理程式的物件，不會移除與本端佇列管理程式相關的物件。此指令也會移除自動定義的通道。它會移除在叢集傳輸佇列中沒有訊息且未連接至完整儲存庫佇列管理程式的通道。

結果

實際上，REFRESH CLUSTER 指令容許對佇列管理程式的完整儲存庫內容進行冷啟動。IBM WebSphere MQ 可確保不會從佇列中遺失任何資料。

相關概念

第 257 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。

回復佇列管理程式

使用 `REFRESH CLUSTER runmqsc` 指令，使佇列管理程式的相關叢集資訊保持最新。從復原點備份回復佇列管理程式之後，請遵循此程序。

開始之前

您已從復原點備份還原叢集佇列管理程式。

關於這項作業

若要回復叢集中的佇列管理程式，請還原佇列管理程式，然後使用 `REFRESH CLUSTER runmqsc` 指令讓叢集資訊保持最新。

註：對於大型叢集，使用 `REFRESH CLUSTER` 指令會干擾進行中的叢集，而此後每隔 27 天，當叢集物件自動將狀態更新傳送給所有相關的佇列管理程式時，會再次造成干擾。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。

程序

針對佇列管理程式參與的所有叢集，在還原的佇列管理程式上發出 `REFRESH CLUSTER` 指令。

下一步

不需要在任何其他佇列管理程式上發出 `REFRESH CLUSTER` 指令。

相關概念

第 257 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。

配置叢集通道的可用性

如果有間歇性網路停止，請遵循良好的配置作法，讓叢集通道保持順暢執行。

開始之前

叢集可讓您不需要定義通道，但您仍需要維護它們。叢集中佇列管理程式之間的通訊使用相同的通道技術，與分散式佇列中使用的相同。若要瞭解叢集通道，您需要熟悉下列事項：

- 通道運作方式
- 如何尋找其狀態
- 如何使用通道結束程式

關於這項作業

您可能想要對下列要點提供一些特殊考量：

程序

配置叢集通道時，請考量下列要點

- 選擇叢集傳送端通道及叢集接收端通道上的 `HBINT` 或 `KAINT` 值，這些通道不會對具有大量活動訊號或保持作用中流程的網路造成負擔。如果您的網路有時會變慢，並導致此長度的延遲，則小於大約 10 秒的間隔會提供假性失敗。
- 設定 `BATCHHB` 值，以減少造成孤立訊息的視窗，因為它在失敗通道上不確定。如果為批次提供較長的填入時間，則失敗通道上的不確定批次更有可能發生。如果通道上的訊息資料流量是零星的，且在訊息激增之間有很長一段時間，則失敗批次更有可能。

- 如果通道的叢集傳送端失敗，然後在活動訊號或保持作用中偵測到失敗之前嘗試重新啟動，則會發生問題。如果通道的叢集接收端仍在作用中，則通道傳送端重新啟動會被拒絕。為了避免失敗，請安排在叢集傳送端通道嘗試重新啟動時終止並重新啟動叢集接收端通道。

在 z/OS 以外的平台上

使用 `qm.ini` 檔案或「Windows NT 登錄」中的 `AdoptNewMCA`、`AdoptNewMCATimeout` 及 `AdoptNewMCACheck` 屬性，控制通道的叢集接收端仍在作用中的問題。

將訊息遞送至叢集以及從叢集遞送訊息

使用佇列別名、佇列管理程式別名及遠端佇列定義，將叢集連接至外部佇列管理程式及其他叢集。

如需與叢集之間遞送訊息的詳細資料，請參閱下列子主題：

相關概念

叢集

叢集運作方式

第 135 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。

第 137 頁的『叢集的元件』

叢集由佇列管理程式、叢集儲存庫、叢集通道及叢集佇列組成。

第 154 頁的『管理 IBM WebSphere MQ 叢集』

您可以建立、延伸及維護 IBM WebSphere MQ 叢集。

第 219 頁的『佇列管理程式別名及叢集』

在叢集中傳送訊息時，請使用佇列管理程式別名來隱藏佇列管理程式的名稱，以及平衡傳送至叢集的訊息工作量。

第 221 頁的『佇列別名及叢集』

使用佇列別名來隱藏叢集佇列的名稱、建立佇列的叢集、採用不同的屬性或採用不同的存取控制。

第 221 頁的『回覆目的地佇列別名及叢集』

回覆目的地佇列別名定義是用來指定回覆資訊的替代名稱。回覆目的地佇列別名定義可以與叢集搭配使用，就像在分散式佇列環境中一樣。

相關工作

第 133 頁的『配置佇列管理程式叢集』

請使用本主題中的鏈結，以瞭解叢集如何運作、如何設計叢集配置，以及取得如何設定簡式叢集的範例。

第 154 頁的『設定新的叢集』

請遵循下列指示來設定範例叢集。個別指示說明在 TCP/IP、LU 6.2 上設定叢集，以及使用單一傳輸佇列或多個傳輸佇列。測試叢集的運作方式是將訊息從一個佇列管理程式傳送至另一個佇列管理程式。

配置叢集的要求/回覆

配置來自叢集外部佇列管理程式的要求/回覆訊息路徑。使用閘道佇列管理程式作為與叢集之間的通訊路徑，以隱藏叢集的內部詳細資料。

開始之前

第 211 頁的圖 43 顯示稱為 QM3 的佇列管理程式，它位於稱為 DEMO 的叢集外部。QM3 可能是 WebSphere MQ 產品上不支援叢集的佇列管理程式。QM3 管理稱為 Q3 的佇列，其定義如下：

```
DEFINE QLOCAL(Q3)
```

在叢集內，有兩個稱為 QM1 和 QM2 的佇列管理程式。QM2 管理稱為 Q2 的叢集佇列，其定義如下：

```
DEFINE QLOCAL(Q2) CLUSTER(DEMO)
```

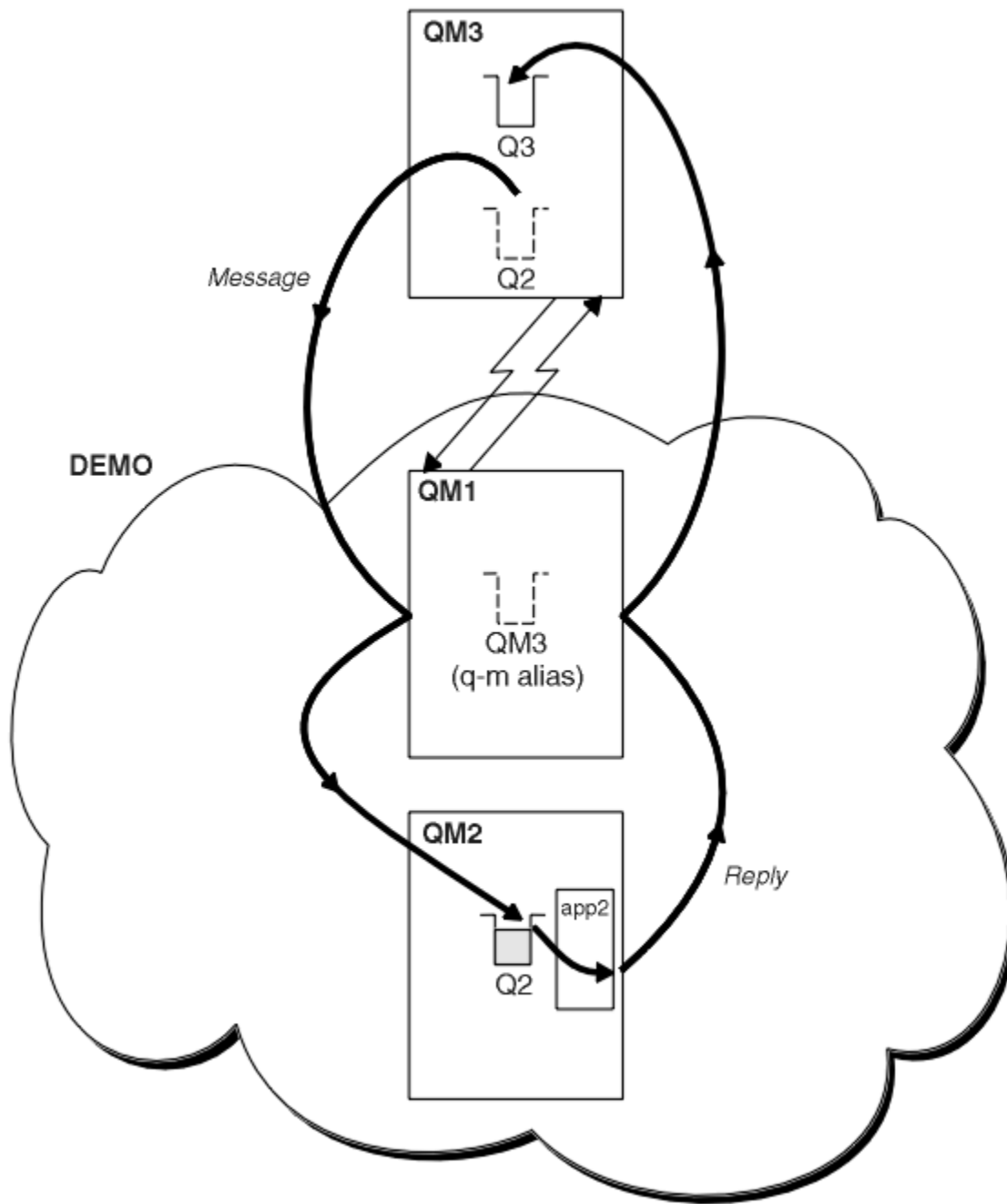


圖 43: 從叢集外部的佇列管理程式放置

關於這項作業

遵循程序中的建議，以設定要求及回覆訊息的路徑。

程序

1. 將要求訊息傳送至叢集。

請考量叢集外部的佇列管理程式如何將訊息放入叢集內部的佇列 Q2 (位於 QM2)。叢集外部的佇列管理程式必須對其放置訊息的叢集中的每一個佇列具有 QREMOTE 定義。

- a) 在 QM3 上定義 Q2 的遠端佇列。

```
DEFINE QREMOTE(Q2) RNAME(Q2) RQMNAME(QM2) XMITQ(QM1)
```

因為 QM3 不是叢集的一部分，所以它必須使用分散式佇列技術進行通訊。因此，它還必須具有 QM1 的傳送端通道及傳輸佇列。QM1 需要對應的接收端通道。通道及傳輸佇列未明確顯示在 [第 211 頁的圖 43](#) 中。

在此範例中，位於 QM3 的應用程式會發出 MQPUT 呼叫，將訊息放置到 Q2。QREMOTE 定義會使用從 QM1 傳輸佇列取得訊息的傳送端通道，將訊息遞送至位於 QM2 的 Q2。

2. 從叢集接收回覆訊息。

使用佇列管理程式別名來建立回覆至叢集外部佇列管理程式的傳回路徑。開道 QM1 會針對叢集以外的佇列管理程式 QM3 通告佇列管理程式別名。它透過將叢集屬性新增至 QM3 的佇列管理程式別名定義，將 QM3 通告至叢集內的佇列管理程式。佇列管理程式別名定義類似於遠端佇列定義，但具有空白 RNAME。

a) 在 QM1 上定義 QM3 的佇列管理程式別名。

```
DEFINE QREMOTE(QM3) RNAME(' ') RQMNAME(QM3) CLUSTER(DEMO)
```

我們必須考慮為用來將回覆從 QM1 轉遞回 QM3 的傳輸佇列選擇名稱。在 QREMOTE 定義中，由於省略 XMITQ 屬性，隱含的是傳輸佇列的名稱是 QM3。但 QM3 與我們預期使用佇列管理程式別名向叢集其餘部分通告的名稱相同。WebSphere MQ 不容許您同時為傳輸佇列和佇列管理程式別名提供相同的名稱。一個解決方案是建立傳輸佇列，以將具有不同名稱的訊息轉遞至佇列管理程式別名的 QM3。

b) 在 QREMOTE 定義中提供傳輸佇列名稱。

```
DEFINE QREMOTE(QM3) RNAME(' ') RQMNAME(QM3) CLUSTER(DEMO) XMITQ(QM3.XMIT)
```

新的佇列管理程式別名會連結稱為 QM3.XMIT 的新傳輸佇列與 QM3 佇列管理程式別名。這是一個簡單而正確的解決方法，但並不完全令人滿意。它已違反傳輸佇列的命名慣例，為它們提供與目標佇列管理程式相同的名稱。是否有任何替代方案保留傳輸佇列命名慣例？

發生問題的原因是要求端預設為在從 QM3 傳送的要求訊息中傳遞 QM3 作為回覆目的地佇列管理程式名稱。QM2 上的伺服器在其回覆中使用 QM3 回覆目的地佇列管理程式名稱來位址 QM3。解決方案需要 QM1 通告 QM3 作為佇列管理程式別名，以將回覆訊息傳回給 QM1，並防止使用 QM3 作為傳輸佇列的名稱。

QM3 上的應用程式需要將回覆目的地佇列管理程式別名傳遞至 QM1，而不是預設提供 QM3 作為回覆目的地佇列管理程式名稱。開道佇列管理程式 QM1 會向 QM3 而非 QM3 本身通告回覆的佇列管理程式別名，避免與傳輸佇列的名稱發生衝突。

c) 在 QM1 上定義 QM3 的佇列管理程式別名。

```
DEFINE QREMOTE(QM3.ALIAS) RNAME(' ') RQMNAME(QM3) CLUSTER(DEMO)
```

需要對配置指令進行兩項變更。

- i) QREMOTE at QM1 現在會將我們的佇列管理程式別名 QM3.ALIAS 通告至叢集的其餘部分，並將它連結至實際佇列管理程式 QM3 的名稱。QM3 再次是要將回覆佇列傳送回 QM3 的傳輸佇列名稱
- ii) 用戶端應用程式在建構要求訊息時，必須提供 QM3.ALIAS 作為回覆目的地佇列管理程式的名稱。您可以使用兩種方式之一，將 QM3.ALIAS 提供給用戶端應用程式。
 - 在 MQMD 中由 MQPUT 建構的回覆目的地佇列管理程式名稱欄位中，撰寫程式碼 QM3.ALIAS。如果您使用動態佇列來取得回覆，則必須這樣做。
 - 在提供回覆目的地佇列名稱時，請使用回覆目的地佇列別名 Q3.ALIAS，而非回覆目的地佇列。

```
DEFINE QREMOTE(Q3.ALIAS) RNAME(Q3) RQMNAME(QM3.ALIAS)
```

下一步

註：您無法使用 **AMQSREQ0** 來示範回覆目的地佇列別名的使用。它會使用參數 3 中提供的佇列名稱或預設 SYSTEM.SAMPLE.REPLY 模型佇列來開啟回覆目的地佇列。您需要修改範例，以提供另一個包含回覆目的地佇列別名的參數，來命名 MQPUT 的回覆目的地佇列管理程式別名。

相關工作

第 213 頁的『隱藏叢集目標佇列管理程式的名稱』

將訊息遞送至叢集中任何佇列管理程式上定義的叢集佇列，而不命名佇列管理程式。

隱藏叢集目標佇列管理程式的名稱

將訊息遞送至叢集中任何佇列管理程式上定義的叢集佇列，而不命名佇列管理程式。

開始之前

- 避免向叢集外的佇列管理程式顯示叢集內的佇列管理程式名稱。
 - 解析對叢集內管理佇列之佇列管理程式的參照，會移除執行工作量平衡的彈性。
 - 這也會讓您難以變更管理叢集中佇列的佇列管理程式。
 - 替代方案是將 RQMNAME 取代為叢集管理者所提供的佇列管理程式別名。
 - 第 213 頁的『隱藏叢集目標佇列管理程式的名稱』說明如何使用佇列管理程式別名，將叢集外的佇列管理程式與叢集內的佇列管理程式管理取消連結。
- 不過，建議將傳輸佇列命名為目標佇列管理程式的名稱。傳輸佇列的名稱會顯示叢集中佇列管理程式的名稱。您必須選擇要遵循的規則。您可以選擇使用佇列管理程式名稱或叢集名稱來命名傳輸佇列：

使用閘道佇列管理程式名稱來命名傳輸佇列

向叢集外部的佇列管理程式揭露閘道佇列管理程式名稱，是隱藏叢集佇列管理程式名稱規則的合理例外。

使用叢集名稱來命名傳輸佇列

如果您未遵循使用目標佇列管理程式名稱來命名傳輸佇列的慣例，請使用叢集名稱。

關於這項作業

修改作業 [第 210 頁的『配置叢集的要求/回覆』](#)，以隱藏叢集內目標佇列管理程式的名稱。

程序

在範例中，請參閱 [第 214 頁的圖 44](#)，在閘道佇列管理程式 QM1 上定義稱為 DEMO 的佇列管理程式別名：

```
DEFINE QREMOTE(DEMO) RNAME(' ') RQMNAME(' ')
```

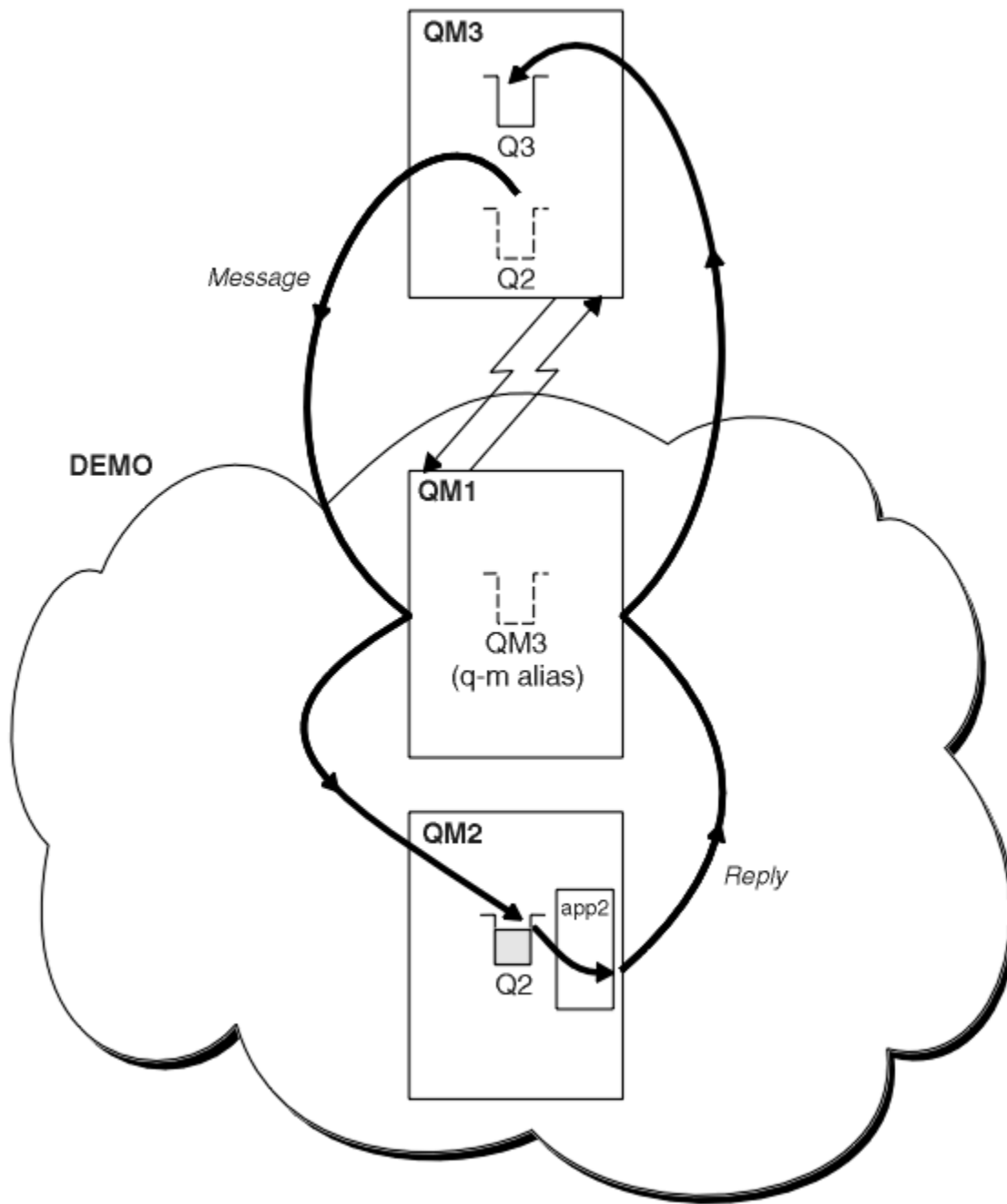


圖 44: 從叢集外部的佇列管理程式放置

QM1 上的 QREMOTE 定義會讓閘道佇列管理程式已知佇列管理程式別名 DEMO。QM3，叢集外部的佇列管理程式可以使用佇列管理程式別名 DEMO，將訊息傳送至 DEMO 上的叢集佇列，而不必使用實際的佇列管理程式名稱。

如果您採用使用叢集名稱來命名連接至叢集的傳輸佇列的慣例，則 Q2 的遠端佇列定義會變成：

```
DEFINE QREMOTE(Q2) RNAME(Q2) RQMNAME(DEMO)
```

結果

以 DEMO 上的 Q2 為目的地的訊息會放置在 DEMO 傳輸佇列上。從傳輸佇列，傳送端通道會將它們傳送至閘道佇列管理程式 QM1。閘道佇列管理程式會將訊息遞送至叢集中管理叢集佇列 Q2 的任何佇列管理程式。

從叢集配置要求/回覆

配置從叢集到叢集外部佇列管理程式的要求/回覆訊息路徑。隱藏叢集內佇列管理程式如何使用閘道佇列管理程式在叢集外進行通訊的詳細資料。

開始之前

第 215 頁的圖 45 顯示叢集內的佇列管理程式 QM2 DEMO。它會將要求傳送至叢集外的佇列管理程式上所管理的佇列 Q3。回覆會回到叢集內位於 QM2 的 Q2。

若要與叢集外的佇列管理程式進行通訊，叢集內的一或多個佇列管理程式會充當閘道。閘道佇列管理程式具有與叢集外部佇列管理程式的通訊路徑。在此範例中，QM1 是閘道。

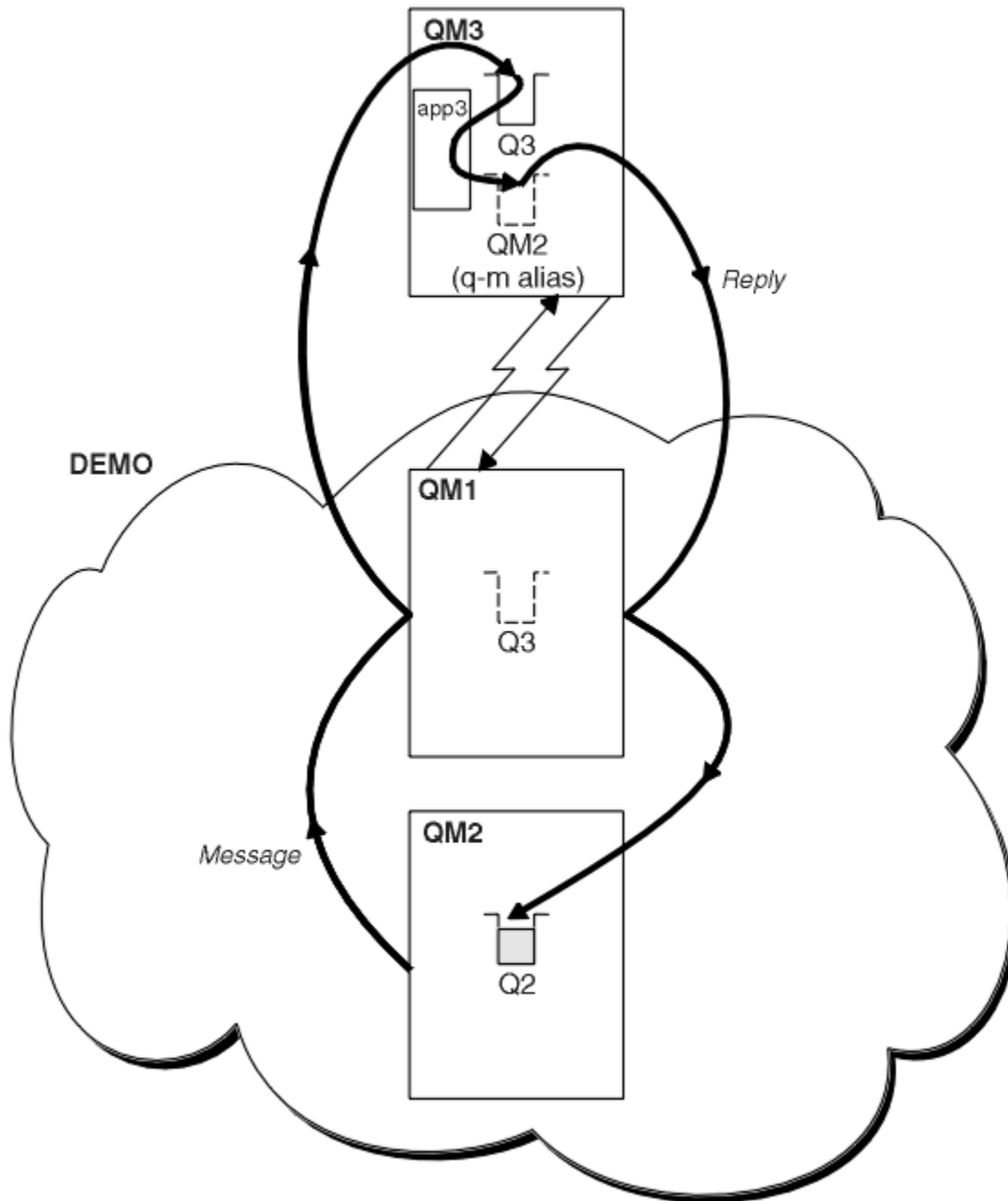


圖 45: 放入叢集外部的佇列管理程式

關於這項作業

遵循指示來設定要求及回覆訊息的路徑

程序

1. 從叢集傳送要求訊息。

請考量位於叢集內的佇列管理程式 QM2 如何將訊息放入位於 QM3 的佇列 Q3 (位於叢集外)。

- a) 在 QM1 上建立 QREMOTE 定義，以將遠端佇列 Q3 通告至叢集

```
DEFINE QREMOTE(Q3) RNAME(Q3) RQMNAME(QM3) CLUSTER(DEMO)
```

它也有傳送端通道及傳輸佇列，可連接至叢集外部的佇列管理程式。QM3 具有對應的接收端通道。通道不會顯示在 [第 215 頁的圖 45](#) 中。

QM2 上的應用程式會發出 MQPUT 呼叫，指定要傳送回覆的目標佇列及佇列。目標佇列為 Q3，回覆目的地佇列為 Q2。

訊息會傳送至 QM1，會使用其遠端佇列定義，將佇列名稱解析為 Q3 (位於 QM3)。

2. 從叢集外部的佇列管理程式接收回覆訊息。

叢集外部的佇列管理程式必須對其傳送訊息的叢集中的每一個佇列管理程式具有佇列管理程式別名。佇列管理程式別名也必須指定閘道佇列管理程式的傳輸佇列名稱。在此範例中，QM3 需要 QM2 的佇列管理程式別名定義：

- a) 在 QM3 上建立佇列管理程式別名 QM2

```
DEFINE QREMOTE(QM2) RNAME(' ') RQMNAME(QM2) XMITQ(QM1)
```

QM3 也需要 QM1 的傳送端通道和傳輸佇列，QM1 也需要對應的接收端通道。

然後，QM3 上的應用程式 **app3** 可以透過發出 MQPUT 呼叫並指定佇列名稱 Q2 及佇列管理程式名稱 QM2，將回覆傳送至 QM2。

下一步

您可以在叢集外定義多個路徑。

從叢集外部配置工作量平衡

配置從叢集外的佇列管理程式到任何叢集佇列副本的訊息路徑。結果是從叢集外部到叢集佇列的每一個實例的工作量平衡要求。

開始之前

配置範例，如 [第 210 頁的『配置叢集的要求/回覆』](#) 中的 [第 211 頁的圖 43](#) 所示。

關於這項作業

在此實務範例中，叢集外的佇列管理程式 QM3 ([第 217 頁的圖 46](#) 中) 將要求傳送至佇列 Q2。Q2 在叢集 DEMO 內的兩個佇列管理程式上管理，以使用工作量平衡。名為 Q2 的佇列定義在佇列管理程式 QM2 及 QM4 上，但未定義在閘道佇列管理程式 QM1 上。來自 QM3(叢集外部的佇列管理程式) 的要求會傳送至 Q2 的任一實例。

QM3 不是叢集的一部分，並使用分散式佇列技術進行通訊。它必須具有 QM1 的傳送端通道及傳輸佇列。QM1 需要對應的接收端通道。通道及傳輸佇列未明確顯示在 [第 217 頁的圖 46](#) 中。

此程序會延伸 [第 210 頁的『配置叢集的要求/回覆』](#) 中 [第 211 頁的圖 43](#) 的範例。

程序

1. 在 QM2 和 QM4 上各定義一個稱為 Q2 的本端佇列。

```
DEFINE QLOCAL(Q2) CLUSTER(DEMO) DEFBIND(NOTFIXED)
```

2. 在 QM3 上建立 Q2 的 QREMOTE 定義。

```
DEFINE QREMOTE(Q2) RNAME(Q2) RQMNAME(Q3) XMITQ(QM1)
```

針對叢集中 QM3 將訊息放入其中的每一個佇列，建立 QREMOTE 定義。

3. 在 QM3 上建立佇列管理程式別名 Q3。

```
DEFINE QREMOTE(Q3) RNAME(' ') RQMNAME(' ') CLUSTER(DEMO) DEFBIND(NOTFIXED)
```

Q3 不是實際佇列管理程式名稱。它是叢集中佇列管理程式別名定義的名稱，它會將佇列管理程式別名 Q3 與空白 ' ' 相等

4. QM1(閘道佇列管理程式) 沒有特殊定義。

結果

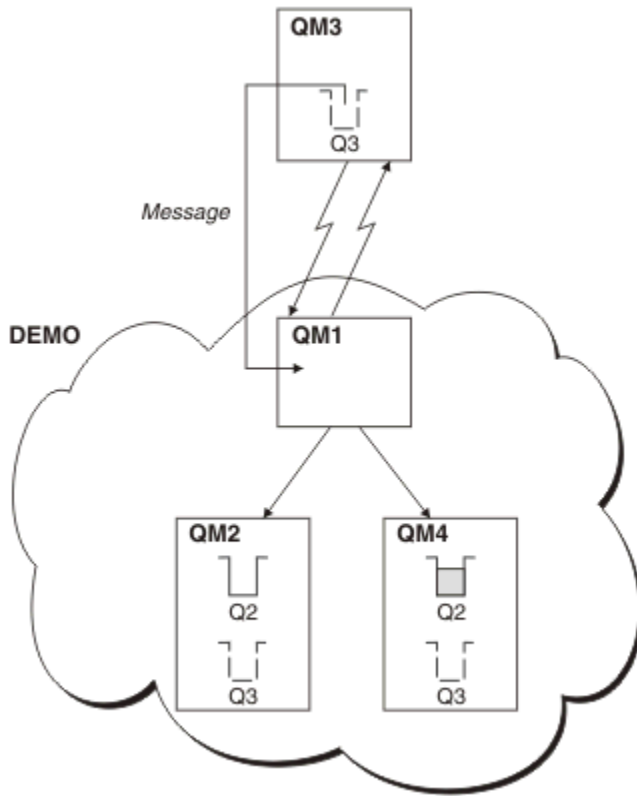


圖 46: 從叢集外部的佇列管理程式放置

當位於 QM3 的應用程式發出 MQPUT 呼叫，將訊息放置到 Q2 時，QREMOTE 定義會導致透過閘道佇列管理程式 QM1 遞送訊息。QM1 會使用工作量平衡，在兩個佇列管理程式 QM2 及 QM4(具有 Q3 的叢集佇列管理程式別名) 上稱為 Q2 的佇列之間配送以 Q2 為目標的訊息。

配置叢集之間的訊息路徑

使用閘道佇列管理程式將叢集連接在一起。透過在閘道佇列管理程式上定義叢集佇列或叢集佇列管理程式別名，讓所有叢集都可以看見佇列或佇列管理程式。

關於這項作業

您可以有許多較小的叢集，而不是將所有佇列管理程式分組在一個大型叢集中。每一個叢集都有一或多個作為橋接器的佇列管理程式。其優點是您可以限制佇列及佇列管理程式名稱在叢集中的可見性。請參閱 [第](#)

150 頁的『重疊叢集』。使用別名來變更佇列及佇列管理程式的名稱，以避免名稱衝突或符合本端命名慣例。

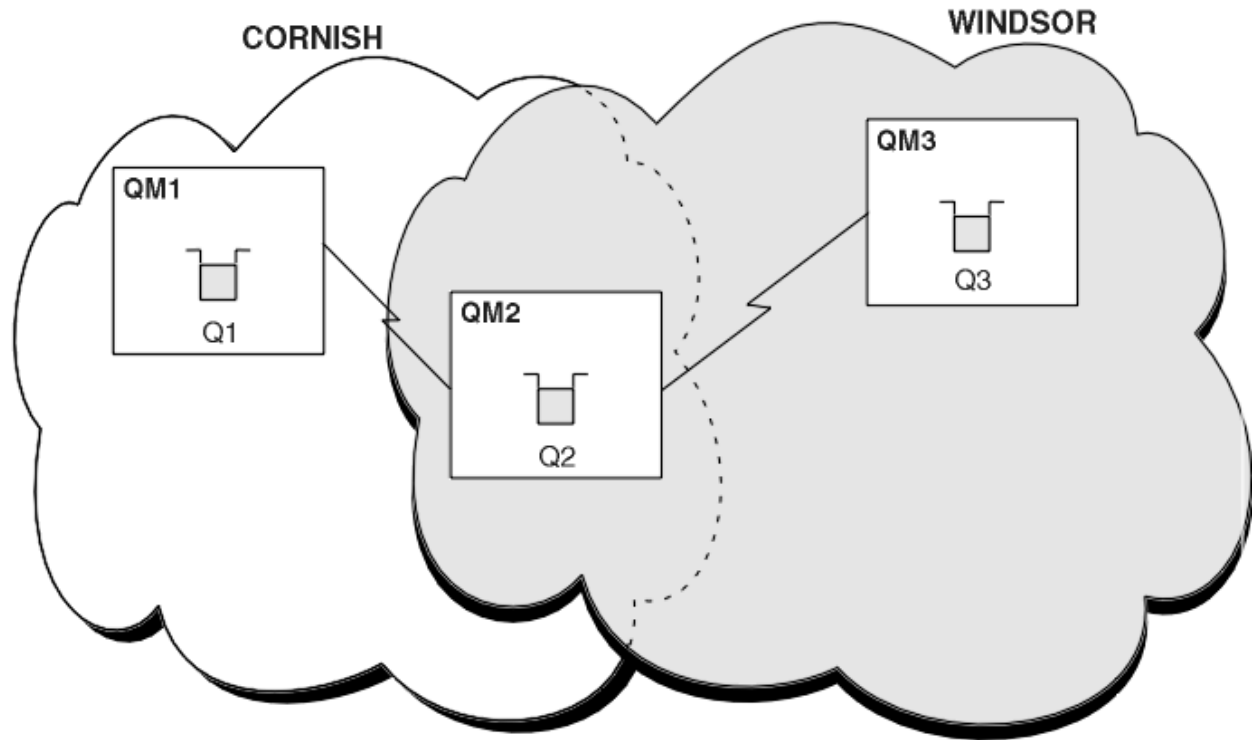


圖 47: 跨叢集橋接

第 218 頁的圖 47 顯示兩個叢集之間具有橋接器。可能有不止一座橋。使用下列程序來配置叢集:

程序

1. 在 QM1 上定義叢集佇列 Q1。

```
DEFINE QLOCAL(Q1) CLUSTER(CORNISH)
```

2. 在 QM3 上定義叢集佇列 Q3。

```
DEFINE QLOCAL(Q3) CLUSTER(WINDSOR)
```

3. 建立名為 CORNISHWINDSOR on QM2 的名單，其中包含兩個叢集的名稱。

```
DEFINE NAMELIST(CORNISHWINDSOR) DESCR('CornishWindsor namelist')  
  NAMES(CORNISH, WINDSOR)
```

4. 在 QM2 上定義叢集佇列 Q2

```
DEFINE QLOCAL(Q2) CLUSNL(CORNISHWINDSOR)
```

下一步

QM2 是這兩個叢集的成員，並且是它們之間的橋接器。對於您要在橋接器上顯示的每一個佇列，您需要在橋接器上具有 QALIAS 定義。例如，在第 218 頁的圖 47 中，在 QM2 上，您需要:

```
DEFINE QALIAS(MYQ3) TARGET(Q3) CLUSTER(CORNISH) DEFBIND(NOTFIXED)
```

使用佇列別名，連接至 CORNISH 中佇列管理程式的應用程式 (例如 QM1) 可以將訊息放置到 Q3。它將 Q3 稱為 MYQ3。訊息會遞送至位於 QM3 的 Q3。

當您開啟佇列時，需要將 DEFBIND 設為 NOTFIXED 或 QDEF。如果保留 DEFBIND 作為預設值 OPEN，則佇列管理程式會將別名定義解析為管理它的橋接器佇列管理程式。橋接器不會轉遞訊息。

對於您要設為可見的每一個佇列管理程式，您需要佇列管理程式別名定義。例如，在 QM2 上，您需要：

```
DEFINE QREMOTE(QM1) RNAME(' ') RQMNAME(QM1) CLUSTER(WINDSOR)
```

連接至 WINDSOR 中任何佇列管理程式的應用程式 (例如 QM3) 可以透過在 MQOPEN 呼叫上明確命名 QM1，將訊息放置到 QM1 上的任何佇列。

佇列管理程式別名及叢集

在叢集中傳送訊息時，請使用佇列管理程式別名來隱藏佇列管理程式的名稱，以及平衡傳送至叢集的訊息工作量。

佇列管理程式別名 (使用含空白 RNAME 的遠端佇列定義建立) 有五種用法：

傳送訊息時重新對映佇列管理程式名稱

佇列管理程式別名可用來將 MQOPEN 呼叫中指定的佇列管理程式名稱重新對映至另一個佇列管理程式。它可以是叢集佇列管理程式。例如，佇列管理程式可能具有佇列管理程式別名定義：

```
DEFINE QREMOTE(YORK) RNAME(' ') RQMNAME(CLUSQM)
```

YORK 可以用作稱為 CLUSQM 之佇列管理程式的別名。當建立此定義的佇列管理程式上的應用程式將訊息放入佇列管理程式 YORK 時，本端佇列管理程式會將名稱解析為 CLUSQM。如果本端佇列管理程式不是稱為 CLUSQM，則會將訊息放置在叢集傳輸佇列上，以移至 CLUSQM。它也會將傳輸標頭變更為指出 CLUSQM，而不是 YORK。

註：此定義僅適用於建立它的佇列管理程式。若要向整個叢集通告別名，您需要將 CLUSTER 屬性新增至遠端佇列定義。然後，來自其他佇列管理程式且以 YORK 為目的地的訊息會傳送至 CLUSQM。

傳送訊息時變更或指定傳輸佇列

別名化可用來將叢集結合至非叢集系統。例如，叢集 ITALY 中的佇列管理程式可以與叢集外部稱為 PALERMO 的佇列管理程式進行通訊。若要通訊，叢集中的其中一個佇列管理程式必須充當閘道。從閘道佇列管理程式發出下列指令：

```
DEFINE QREMOTE(ROME) RNAME(' ') RQMNAME(PALERMO) XMITQ(X) CLUSTER(ITALY)
```

指令是佇列管理程式別名定義。它將 ROME 定義並通告為佇列管理程式，來自叢集 ITALY 中任何佇列管理程式的訊息可以透過此佇列管理程式多重躍點到達其目的地 (PALERMO)。在佇列管理程式名稱設為 ROME 的情況下開啟的佇列中放置的訊息，會以佇列管理程式別名定義傳送至閘道佇列管理程式。在此之後，訊息會放置在傳輸佇列 X 上，並由非叢集通道移至佇列管理程式 PALERMO。

在此範例中，選擇名稱 ROME 並不重要。QREMOTE 和 RQMNAME 的值可能相同。

接收訊息時決定目的地

當佇列管理程式接收訊息時，它會從傳輸標頭擷取目的地佇列及佇列管理程式的名稱。它會在傳輸標頭中尋找與佇列管理程式同名的佇列管理程式別名定義。如果它找到一個，則會將佇列管理程式別名定義中的 RQMNAME 替換為傳輸標頭中的佇列管理程式名稱。

以這種方式使用佇列管理程式別名有兩個原因：

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

在閘道佇列管理程式中使用佇列管理程式別名，在不同叢集中的佇列管理程式之間遞送訊息。

應用程式可以使用佇列管理程式別名，將訊息傳送至不同叢集中的佇列。佇列不必是叢集佇列。佇列定義在一個叢集中。應用程式連接至不同叢集中的佇列管理程式。閘道佇列管理程式會連接兩個叢集。如果佇列未定義為叢集，為了進行正確的遞送，應用程式必須使用佇列名稱和叢集佇列管理程式別名來開啟佇列。如需配置範例，請參閱第 182 頁的『使用閘道佇列管理程式建立兩個重疊的叢集』，從中取得圖 1 中所說明的回覆訊息流程。

此圖顯示回覆訊息回到暫時動態佇列 (稱為 RQ) 所採用的路徑。連接至 QM3 的伺服器應用程式會使用佇列管理程式名稱 QM2 開啟回覆佇列。佇列管理程式名稱 QM2 在 QM1 上定義為叢集佇列管理程式別名。QM3 會將回覆訊息遞送至 QM1。QM1 會將訊息遞送至 QM2。

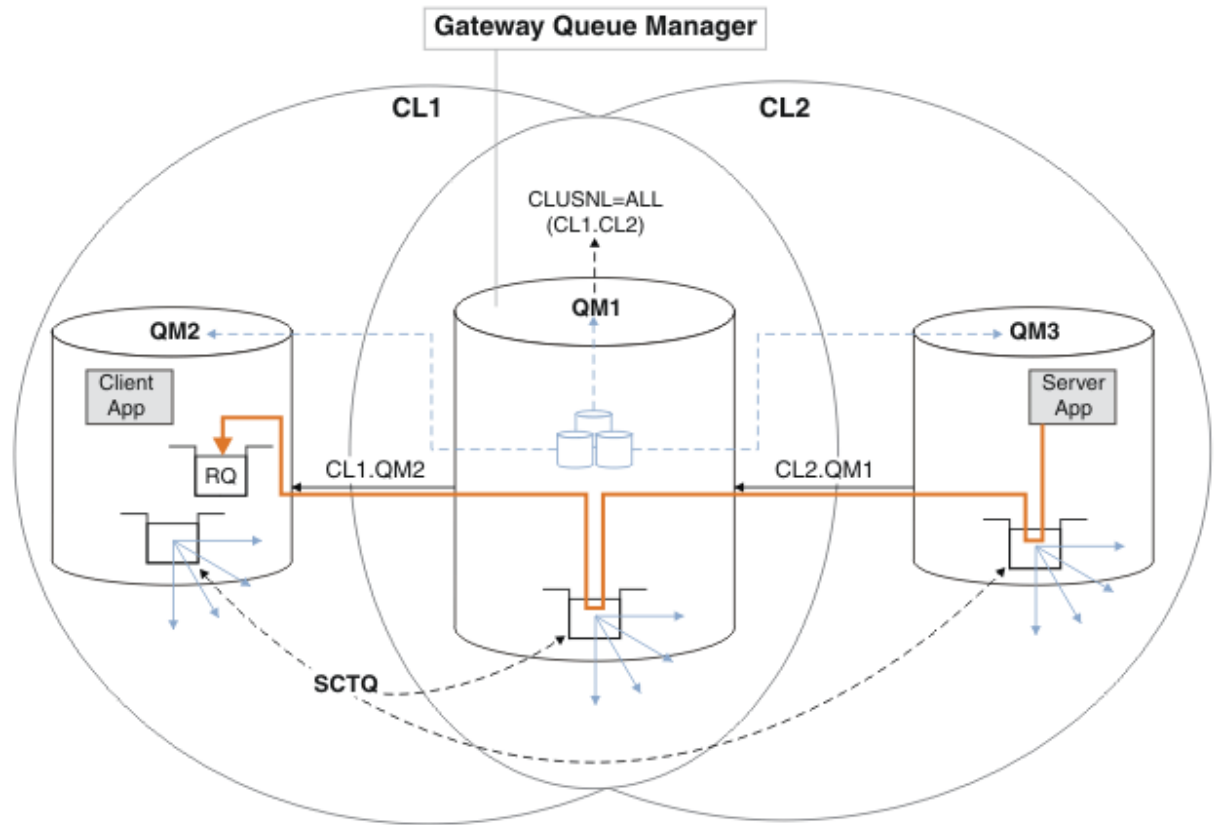


圖 48: 使用佇列管理程式別名將回覆訊息傳回至不同的叢集

遞送的運作方式如下。每個叢集中的每個佇列管理程式在 QM1 上都有一個佇列管理程式別名定義。別名會在所有叢集中形成叢集。從每一個別名到佇列管理程式的灰色虛線箭頭顯示每一個佇列管理程式別名都已解析為至少其中一個叢集中的實際佇列管理程式。在此情況下，QM2 別名同時在叢集 CL1 和 CL2 中形成叢集，並解析為 CL1 中的實際佇列管理程式 QM2。伺服器應用程式會使用回覆目的地佇列名稱 RQ 及回覆目的地佇列管理程式名稱 QM2 來建立回覆訊息。訊息會遞送至 QM1，因為佇列管理程式別名定義 QM2 定義在叢集 CL2 中的 QM1 上，且佇列管理程式 QM2 不在叢集 CL2 中。因為訊息無法傳送至目標佇列管理程式，所以會傳送至具有別名定義的佇列管理程式。

QM1 會將訊息放在 QM1 上的叢集傳輸佇列中，以便傳送至 QM2。QM1 會將訊息遞送至 QM2，因為 QM1 for QM2 上的佇列管理程式別名定義會將 QM2 定義為實際目標佇列管理程式。定義不是循環，因為別名定義只能參照實際定義；別名不能指向本身。實際定義由 QM1 解析，因為 QM1 和 QM2 都位於相同的叢集 CL1 中。QM1 會從 CL1 的儲存庫中找出 QM2 的連線資訊，並將訊息遞送至 QM2。若要讓 QM1 重新遞送訊息，伺服器應用程式必須已開啟回覆佇列，並將選項 DEFBIND 設為 MQBND_BIND_NOT_FIXED。如果伺服器應用程式已使用選項 MQBND_BIND_ON_OPEN 開啟回覆佇列，則訊息不會重新遞送，且會在無法傳送的郵件佇列上結束。

使用佇列管理程式作為叢集的閘道，以平衡來自叢集外部的訊息工作量。

您在叢集中的多個佇列管理程式上定義稱為 EDINBURGH 的佇列。您想要叢集機制平衡從叢集外部進入該佇列之訊息的工作量。

來自叢集外部的佇列管理程式需要叢集中一個佇列管理程式的傳輸佇列及傳送端通道。此佇列稱為閘道佇列管理程式。若要利用預設工作量平衡機制，必須套用下列其中一個規則：

- 閘道佇列管理程式不得包含 EDINBURGH 佇列的實例。
- 閘道佇列管理程式會指定 ALTER QMGR 上的 CLWLUSEQ(ANY)。

如需從叢集外部進行工作量平衡的範例，請參閱第 216 頁的『從叢集外部配置工作量平衡』

回覆目的地佇列別名及叢集

回覆目的地佇列別名定義是用來指定回覆資訊的替代名稱。回覆目的地佇列別名定義可以與叢集搭配使用，就像在分散式佇列環境中一樣。

例如：

- 位於佇列管理程式 VENICE 的應用程式會使用 MQPUT 呼叫，將訊息傳送至佇列管理程式 PISA。應用程式在訊息描述子中提供下列回覆目的地佇列資訊：

```
ReplyToQ='QUEUE'  
ReplyToQMgr=''
```

- 為了能夠在 OTHERQ (PISA) 上接收傳送至 QUEUE 的回覆，請在 VENICE 上建立遠端佇列定義，以作為回覆目的地佇列別名。別名僅在其建立所在的系統上有效。

```
DEFINE QREMOTE(QUEUE) RNAME(OTHERQ) RQMNAME(PISA)
```

即使 RQMNAME 本身是叢集佇列管理程式，RQMNAME 和 QREMOTE 也可以指定相同的名稱。

佇列別名及叢集

使用佇列別名來隱藏叢集佇列的名稱、建立佇列的叢集、採用不同的屬性或採用不同的存取控制。

QALIAS 定義用來建立用來識別佇列的別名。您可以基於下列數個原因來建立別名：

- 您想要開始使用不同的佇列，但不想要變更您的應用程式。
- 您不希望應用程式知道其將訊息放入其中的佇列實際名稱。
- 您的命名慣例可能不同於定義佇列的命名慣例。
- 您的應用程式可能未獲授權依其實際名稱來存取佇列，但只能依其別名來存取佇列。

使用 DEFINE QALIAS 指令在佇列管理程式上建立 QALIAS 定義。例如，執行下列指令：

```
DEFINE QALIAS(PUBLIC) TARGET(LOCAL) CLUSTER(C)
```

此指令會將稱為 PUBLIC 的佇列通告給叢集 C 中的佇列管理程式。PUBLIC 是一個別名，解析為稱為 LOCAL 的佇列。傳送至 PUBLIC 的訊息會遞送至稱為 LOCAL 的佇列。

您也可以使用佇列別名定義，將佇列名稱解析為叢集佇列。例如，執行下列指令：

```
DEFINE QALIAS(PRIVATE) TARGET(PUBLIC)
```

此指令可讓佇列管理程式使用名稱 PRIVATE，以存取叢集中其他位置以 PUBLIC 名稱通告的佇列。因為此定義不包含 CLUSTER 屬性，所以它僅適用於建立它的佇列管理程式。

使用叢集進行工作量管理

透過在叢集中的不同佇列管理程式上定義佇列的多個實例，您可以將服務佇列的工作分散到多部伺服器。有數個因素可以防止在失敗時將訊息重新排入不同佇列管理程式。

除了設定叢集以減少系統管理之外，您還可以建立多個佇列管理程式管理相同佇列實例的叢集。

您可以組織叢集，使其中的佇列管理程式彼此複製。每一個佇列管理程式都能夠執行相同的應用程式，且具有相同佇列的本端定義。您可以透過具有數個應用程式實例，在佇列管理程式之間分散工作量。應用程式的每一個實例都會接收訊息，並彼此獨立執行。

以這種方式使用叢集的優點如下：

- 增加佇列及應用程式的可用性
- 更快的訊息傳輸量
- 在您的網路中更平均的工作量分佈

管理特定佇列實例的任何一個佇列管理程式都可以處理針對該佇列的訊息。應用程式在傳送訊息時不會命名佇列管理程式。工作量管理演算法會決定哪個佇列管理程式會處理訊息。

如需工作量管理之叢集配置的相關資訊，請參閱下列子主題：

相關概念

叢集

叢集運作方式

[第 135 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。

[第 137 頁的『叢集的元件』](#)

叢集由佇列管理程式、叢集儲存庫、叢集通道及叢集佇列組成。

[第 154 頁的『管理 IBM WebSphere MQ 叢集』](#)

您可以建立、延伸及維護 IBM WebSphere MQ 叢集。

[第 210 頁的『將訊息遞送至叢集以及從叢集遞送訊息』](#)

使用佇列別名、佇列管理程式別名及遠端佇列定義，將叢集連接至外部佇列管理程式及其他叢集。

相關工作

[第 133 頁的『配置佇列管理程式叢集』](#)

請使用本主題中的鏈結，以瞭解叢集如何運作、如何設計叢集配置，以及取得如何設定簡式叢集的範例。

[第 154 頁的『設定新的叢集』](#)

請遵循下列指示來設定範例叢集。個別指示說明在 TCP/IP、LU 6.2 上設定叢集，以及使用單一傳輸佇列或多個傳輸佇列。測試叢集的運作方式是將訊息從一個佇列管理程式傳送至另一個佇列管理程式。

[撰寫及編譯叢集工作量結束程式](#)

具有多個佇列實例的叢集範例

在此具有多個佇列實例的叢集範例中，訊息會遞送至佇列的不同實例。您可以強制將訊息傳送至佇列的特定實例，並且可以選擇將一連串訊息傳送至其中一個佇列管理程式。

[第 223 頁的圖 49](#) 顯示其中有多個佇列 Q3 定義的叢集。如果位於 QM1 的應用程式將訊息放置到 Q3，則不一定知道 Q3 的哪個實例將處理其訊息。如果應用程式在有 Q3 本端實例的 QM2 或 QM4，上執行，則依預設會開啟 Q3 本端實例。透過設定 CLWLUSEQ 佇列屬性，可以將佇列的本端實例視為佇列的遠端實例。

MQOPEN 選項 DefBind 可控制在發出 MQOPEN 呼叫時，還是從傳輸佇列傳送訊息時，是否選擇目標佇列管理程式。

如果您將 DefBind 設為 MQBND_BIND_NOT_FIXED，則訊息可以傳送至傳輸訊息時可用的佇列實例。這可避免下列問題：

- 當訊息到達目標佇列管理程式時，無法使用目標佇列。
- 佇列的狀態已變更。
- 已使用叢集佇列別名放置訊息，且在定義叢集佇列別名實例的佇列管理程式上沒有目標佇列的實例。

如果在傳輸時發現這些問題，則會探查目標佇列的另一個可用實例，並重新遞送訊息。如果沒有可用的佇列實例，則會將訊息放置在無法傳送郵件的佇列上。

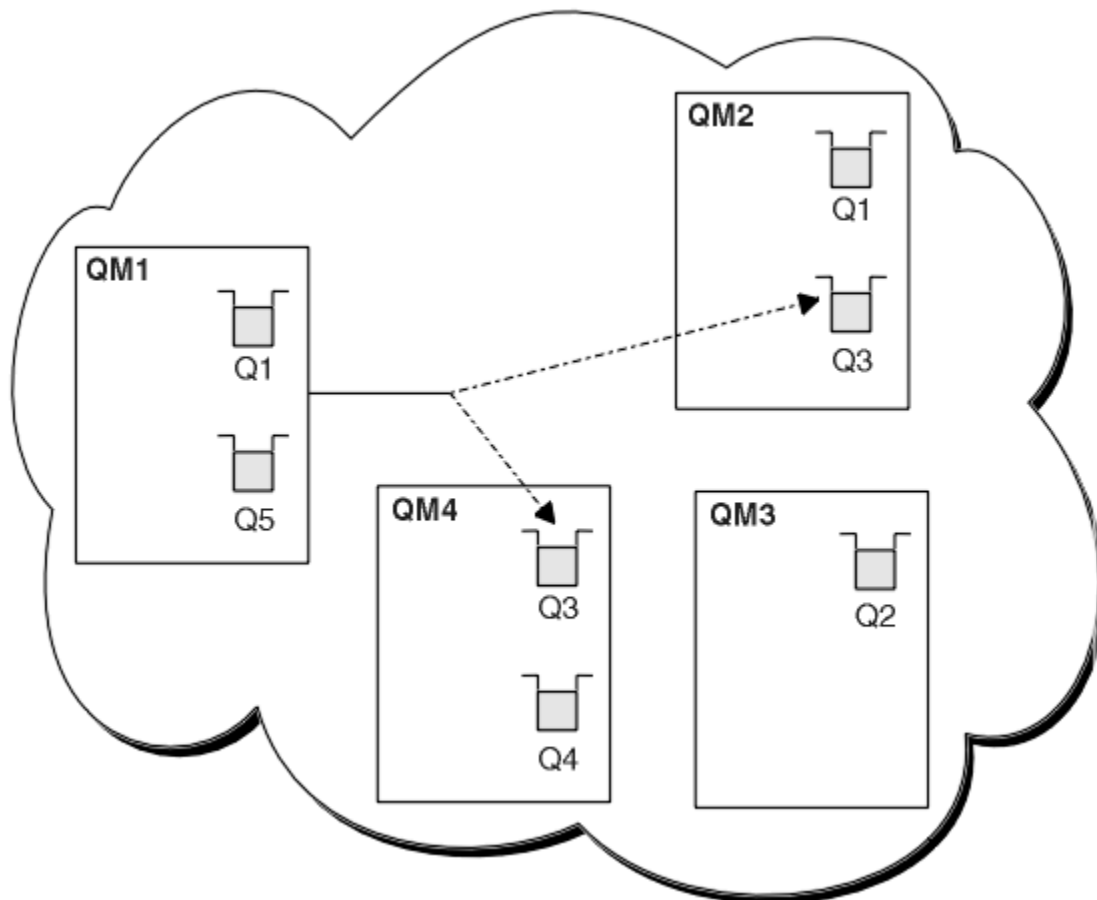


圖 49: 具有相同佇列的多個實例的叢集

如果訊息已指派給具有 MQBND_BIND_ON_OPEN 的固定佇列管理程式或通道，則可防止重新遞送訊息的因素之一。MQOPEN 上所連結的訊息絕不會重新配置到另一個通道。另請注意，只有在叢集通道實際失敗時，才會進行訊息重新配置。如果通道已失敗，則不會重新配置。

如果目的地佇列管理程式無法運作，系統會嘗試重新遞送訊息。在這樣做時，它不會因為執行遺失訊息的風險或建立重複訊息而影響訊息的完整性。如果佇列管理程式失敗並讓訊息不確定，則不會重新遞送該訊息。

新增在本端管理佇列的佇列管理程式

請遵循下列指示來新增 INVENTQ 實例，以提供額外容量在巴黎和紐約執行庫存應用程式系統。

開始之前

註: 如果要將叢集的變更延伸到整個叢集，至少必須一律有一個完整儲存庫可用。在啟動這項作業之前，請確定您的儲存庫可供使用。

測試情境：

- 已依照 將新的佇列管理程式新增至叢集中的說明來設定 INVENTORY 叢集。它包含三個佇列管理程式：LONDON 和 NEWYORK 都保留完整儲存庫，PARIS 保留局部儲存庫。庫存應用程式在紐約的系統上執行，並連接至 NEWYORK 佇列管理程式。應用程式是由 INVENTQ 佇列上的訊息到達所驅動。
- 我們想要新增 INVENTQ 的實例，以提供在巴黎和紐約執行庫存應用程式系統的額外容量。

關於這項作業

請遵循下列步驟，在本端新增管理佇列的佇列管理程式。

程序

1. 變更 PARIS 佇列管理程式。

若要讓巴黎的應用程式使用巴黎的 INVENTQ 及紐約的應用程式，我們必須通知佇列管理程式。在 PARIS 上，發出下列指令：

```
ALTER QMGR CLWLUSEQ(ANY)
```

2. 請檢閱庫存應用程式，以取得訊息親緣性。

在繼續之前，請確定庫存應用程式對訊息的處理順序沒有任何相依關係。如需相關資訊，請參閱第 233 頁的『處理訊息親緣性』。

3. 在巴黎的系統上安裝庫存應用程式。
4. 定義叢集佇列 INVENTQ。

已由 NEWYORK 佇列管理程式管理的 INVENTQ 佇列也將由 PARIS 管理。在 PARIS 佇列管理程式上定義它，如下所示：

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY)
```

既然您已完成所有定義(如果尚未完成的話)，請在 WebSphere MQ for z/OS 上啟動通道起始程式。在所有平台上，在佇列管理程式 PARIS 上啟動接聽器程式。接聽器會接聽送入的網路要求，並在需要時啟動叢集接收端通道。

結果

第 224 頁的圖 50 顯示此作業所設定的叢集。

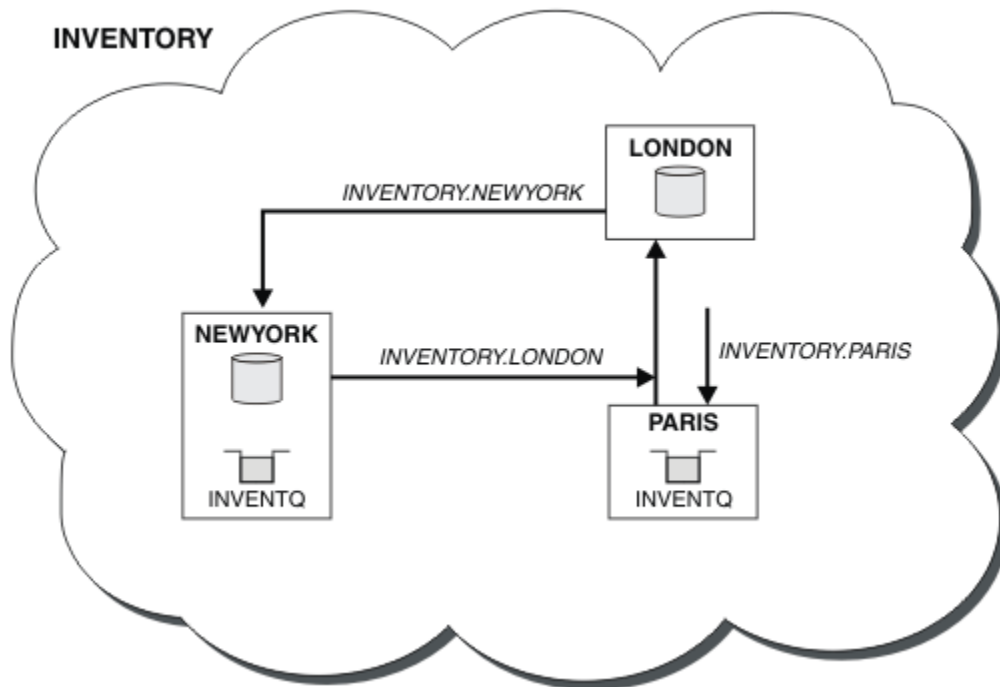


圖 50: INVENTORY 叢集，具有三個佇列管理程式

已完成對此叢集的修改，但未變更佇列管理程式 NEWYORK 或 LONDON。這些佇列管理程式中的完整儲存庫會自動更新為能夠傳送訊息至 INVENTQ 所需的資訊，網址為 PARIS。

下一步

現在，INVENTQ 佇列和庫存應用程式是在叢集中的兩個佇列管理程式上管理。這會增加其可用性，加快訊息的傳輸量，並容許在兩個佇列管理程式之間分配工作量。由任何佇列管理程式 LONDON、NEWYORK、PARIS 放置到 INVENTQ 的訊息會交替遞送至 PARIS 或 NEWYORK，以平衡工作量。

在叢集中使用兩個網路

遵循下列指示，在有兩個不同網路的 TOKYO 中新增商店。兩者都必須可供使用，才能與東京的佇列管理程式通訊。

開始之前

註：如果要將叢集的變更延伸到整個叢集，至少必須一律有一個完整儲存庫可用。在啟動這項作業之前，請確定您的儲存庫可供使用。

測試情境：

- 已依照「將佇列管理程式新增至叢集」中的說明來設定 INVENTORY 叢集。它包含三個佇列管理程式：LONDON 和 NEWYORK 都保留完整儲存庫，PARIS 保留局部儲存庫。庫存應用程式在紐約的系統上執行，並連接至 NEWYORK 佇列管理程式。應用程式是由 INVENTQ 佇列上的訊息到達所驅動。
- 正在 TOKYO 中新增商店，其中有兩個不同的網路。兩者都必須可供使用，才能與東京的佇列管理程式通訊。

關於這項作業

請遵循下列步驟，在叢集中使用兩個網路。

程序

1. 決定哪一個完整儲存庫 TOKYO 會先參照。

叢集中的每個佇列管理程式都必須參照一個或其他完整儲存庫，才能收集叢集的相關資訊。它會建置自己的局部儲存庫。您選擇的儲存庫沒有特別重要的意義。在此範例中，選擇 NEWYORK。一旦新佇列管理程式已加入叢集，它就會與這兩個儲存庫進行通訊。

2. 定義 CLUSRCVR 通道。

叢集中的每個佇列管理程式都需要定義可接收訊息的叢集接收端。此佇列管理程式需要能夠在每一個網路上進行通訊。

```
DEFINE CHANNEL(INVENTORY.TOKYO.NETB) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME('TOKYO.NETB.CMSTORE.COM') CLUSTER(INVENTORY) DESCR('Cluster-receiver
channel using network B for TOKYO')
```

```
DEFINE CHANNEL(INVENTORY.TOKYO.NETA) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME('TOKYO.NETA.CMSTORE.COM') CLUSTER(INVENTORY) DESCR('Cluster-receiver
channel using network A for TOKYO')
```

3. 在佇列管理程式 TOKYO 上定義 CLUSSDR 通道

叢集中的每個佇列管理程式都需要定義一個叢集傳送端通道，以將訊息傳送至其第一個完整儲存庫。在此情況下，我們已選擇 NEWYORK，因此 TOKYO 需要下列定義：

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY) DESCR('Cluster-sender
channel from TOKYO to repository at NEWYORK')
```

現在，您已完成所有定義，如果尚未完成，請在 WebSphere MQ for z/OS 上啟動通道起始程式。在所有平台上，在佇列管理程式 PARIS 上啟動接聽器程式。接聽器程式會接聽送入的網路要求，並在需要時啟動叢集接收端通道。

結果

第 226 頁的圖 51 顯示此作業所設定的叢集。

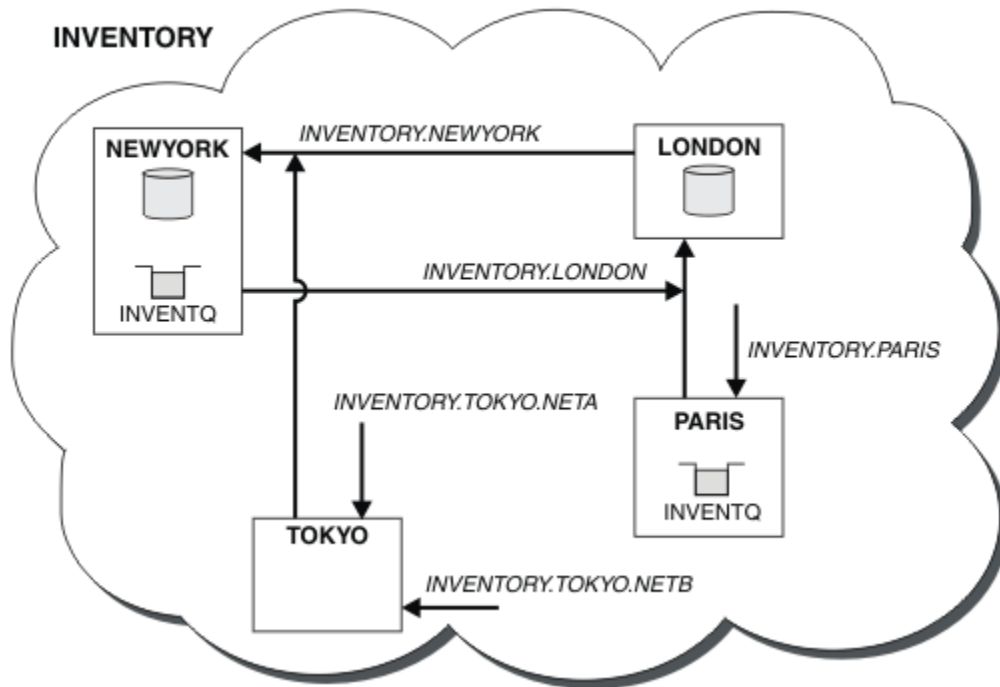


圖 51: INVENTORY 叢集，具有四個佇列管理程式

透過僅建立三個定義，我們已將佇列管理程式 TOKYO 新增至具有兩個不同網路路徑可供其使用的叢集。

相關工作

第 164 頁的『將佇列管理程式加入叢集中』

請遵循下列指示，將佇列管理程式新增至您建立的叢集。叢集佇列及主題的訊息會使用單一叢集傳輸佇列 SYSTEM.CLUSTER.TRANSMIT.QUEUE 來傳送。

在叢集中使用主要及次要網路

請遵循下列指示，使一個網路成為主要網路，而另一個網路成為備份網路。如果主要網路有問題，請使用備份網路。

開始之前

註: 如果要將叢集的變更延伸到整個叢集，至少必須一律有一個完整儲存庫可用。在啟動這項作業之前，請確定您的儲存庫可供使用。

測試情境:

- 已依照第 225 頁的『在叢集中使用兩個網路』中的說明來設定 INVENTORY 叢集。它包含四個佇列管理程式: LONDON 及 NEWYORK 都保留完整儲存庫; PARIS 及 TOKYO 保留局部儲存庫。庫存應用程式在紐約系統上執行，並連接至佇列管理程式 NEWYORK。TOKYO 佇列管理程式可以在兩個不同的網路上進行通訊。
- 您想要使其中一個網路成為主要網路，而另一個網路成為備份網路。如果主要網路有問題，您計劃使用備份網路。

關於這項作業

使用 NETPRTY 屬性來配置叢集中的主要及次要網路。

程序

變更 TOKYO 上現有的 CLUSRCVR 通道。

若要指出網路 A 通道是主要通道，而網路 B 通道是次要通道，請使用下列指令：

- a) ALTER CHANNEL(INVENTORY.TOKYO.NETA) CHLTYPE(CLUSRCVR) NETPRTY(2) DESCR('Main cluster-receiver channel for TOKYO')
- b) ALTER CHANNEL(INVENTORY.TOKYO.NETB) CHLTYPE(CLUSRCVR) NETPRTY(1) DESCR('Backup cluster-receiver channel for TOKYO')

下一步

透過配置具有不同網路優先順序的通道，您現在已定義給具有主要網路及次要網路的叢集。叢集中使用這些通道的佇列管理程式會在主要網路可用時自動使用它。當主要網路無法使用時，佇列管理程式會失效接手以使用次要網路。

新增佇列以作為備份

遵循這些指示，在芝加哥為現在在紐約執行的庫存系統提供備份。只有在紐約系統出現問題時，才會使用芝加哥系統。

開始之前

註：如果要將叢集的變更延伸到整個叢集，至少必須一律有一個完整儲存庫可用。在啟動這項作業之前，請確定您的儲存庫可供使用。

測試情境：

- 已依照第 164 頁的『將佇列管理程式加入叢集中』中的說明來設定 INVENTORY 叢集。它包含三個佇列管理程式：LONDON 和 NEWYORK 都保留完整儲存庫，PARIS 保留局部儲存庫。庫存應用程式在紐約的系統上執行，並連接至 NEWYORK 佇列管理程式。應用程式是由 INVENTQ 佇列上的訊息到達所驅動。
- 芝加哥正在建立一家新商店，為現在在紐約執行的庫存系統提供備份。只有在紐約系統出現問題時，才會使用芝加哥系統。

關於這項作業

請遵循下列步驟來新增佇列以作為備份。

程序

1. 決定哪一個完整儲存庫 CHICAGO 會先參照。

叢集中的每個佇列管理程式都必須參照一個或其他完整儲存庫，才能收集叢集的相關資訊。它會建置自己的局部儲存庫。對於任何特定佇列管理程式，您選擇哪個儲存庫並不特別重要。在此範例中，選擇 NEWYORK。一旦新佇列管理程式已加入叢集，它就會與這兩個儲存庫進行通訊。

2. 定義 CLUSRCVR 通道。

叢集中的每個佇列管理程式都需要定義可接收訊息的叢集接收端。在 CHICAGO 上，定義：

```
DEFINE CHANNEL(INVENTORY.CHICAGO) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(CHICAGO.CMSTORE.COM) CLUSTER(INVENTORY) DESCR('Cluster-receiver
channel for CHICAGO')
```

3. 在佇列管理程式 CHICAGO 上定義 CLUSSDR 通道。

叢集中的每個佇列管理程式都需要定義一個叢集傳送端通道，以將訊息傳送至其第一個完整儲存庫。在此情況下，我們已選擇 NEWYORK，因此 CHICAGO 需要下列定義：

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY) DESCR('Cluster-sender
channel from CHICAGO to repository at NEWYORK')
```

4. 變更現有叢集佇列 INVENTQ。

已由 NEWYORK 佇列管理程式管理的 INVENTQ 是佇列的主要實例。

```
ALTER QLOCAL(INVENTQ) CLWLPRTY(2)
```

5. 請檢閱庫存應用程式，以取得訊息親緣性。

在繼續之前，請確定庫存應用程式對訊息的處理順序沒有任何相依關係。

6. 在 CHICAGO 中的系統上安裝庫存應用程式。

7. 定義備份叢集佇列 INVENTQ

INVENTQ 已由 NEWYORK 佇列管理程式管理，也將由 CHICAGO 作為備份進行管理。在 CHICAGO 佇列管理程式上定義它，如下所示：

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY) CLWLPRTY(1)
```

現在，您已完成所有定義，如果尚未完成，請在 WebSphere MQ for z/OS 上啟動通道起始程式。在所有平台上，在佇列管理程式 CHICAGO 上啟動接聽器程式。接聽器程式會接聽送入的網路要求，並在需要時啟動叢集接收端通道。

結果

第 228 頁的圖 52 顯示此作業所設定的叢集。

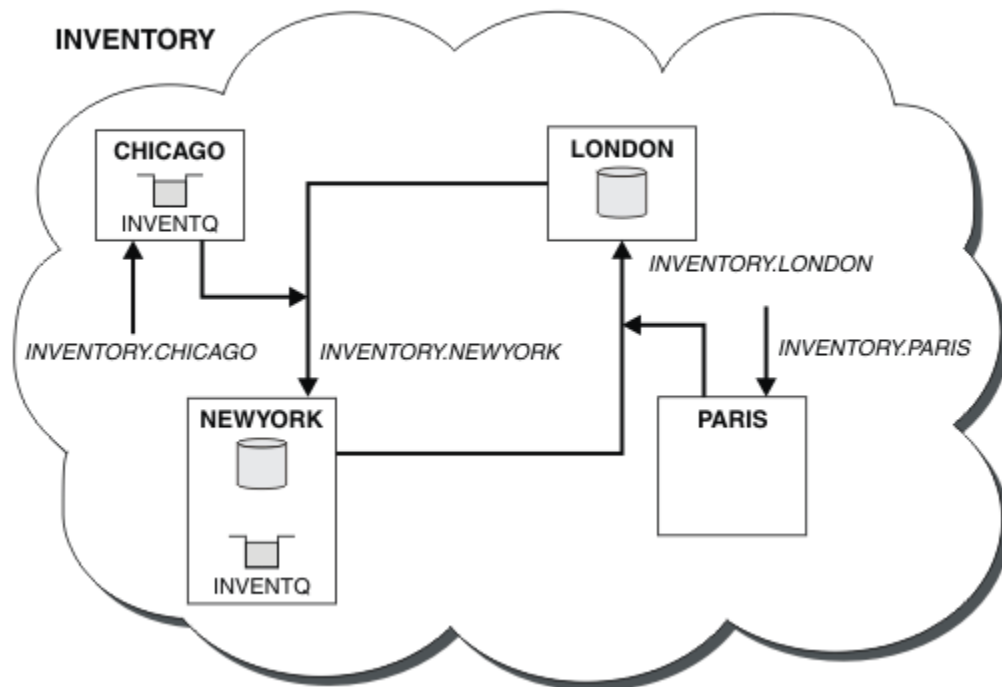


圖 52: INVENTORY 叢集，具有四個佇列管理程式

現在，INVENTQ 佇列和庫存應用程式是在叢集中的兩個佇列管理程式上管理。CHICAGO 佇列管理程式是備份。放置至 INVENTQ 的訊息會遞送至 NEWYORK，除非當它們改為傳送至 CHICAGO 時無法使用。

註：

遠端佇列管理程式的可用性取決於該佇列管理程式的通道狀態。當通道啟動時，它們的狀態會變更數次，部分狀態對叢集工作量管理演算法較不優先。實際上，這表示在啟動通往高優先順序(主要)目的地的通道時，可以選擇低優先順序(備份)目的地。

如果您需要確保沒有任何訊息前往備份目的地，請不要使用 CLWLPRTY。請考慮使用個別佇列，或使用手動從主要切換至備份的 CLWLRANK。

限制使用的通道數

遵循下列指示，以限制在各種佇列管理程式上安裝價格檢查應用程式時，每一部伺服器執行的作用中通道數。

開始之前

註: 如果要將叢集的變更延伸到整個叢集，至少必須一律有一個完整儲存庫可用。在啟動這項作業之前，請確定您的儲存庫可供使用。

測試情境：

- 價格檢查應用程式將安裝在各種佇列管理程式上。為了將使用的通道數目維持在較低的數目，每一部伺服器執行的作用中通道數目會受到限制。應用程式是由 PRICEQ 佇列上的訊息到達所驅動。
- 四個伺服器佇列管理程式負責管理價格檢查應用程式。兩個查詢佇列管理程式會將訊息傳送至 PRICEQ，以查詢價格。另外還有兩個佇列管理程式配置成完整儲存庫。

關於這項作業

請遵循下列步驟來限制使用的通道數。

程序

1. 選擇兩個完整儲存庫。

選擇兩個佇列管理程式作為價格檢查叢集的完整儲存庫。它們稱為 REPOS1 和 REPOS2。

發出下列指令：

```
ALTER QMGR REPOS(PRICECHECK)
```

2. 在每一個佇列管理程式上定義 CLUSRCVR 通道。

在叢集中的每一個佇列管理程式上，定義叢集接收端通道及叢集傳送端通道。先定義哪一個並不重要。

```
DEFINE CHANNEL(PRICECHECK.SERVE1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)  
CONNNAME(SERVER1.COM) CLUSTER(PRICECHECK) DESCR('Cluster-receiver channel')
```

3. 在每一個佇列管理程式上定義 CLUSSDR 通道。

在每一個佇列管理程式上建立 CLUSSDR 定義，以將該佇列管理程式鏈結至其中一個或其他完整儲存庫佇列管理程式。

```
DEFINE CHANNEL(PRICECHECK.REPOS1) CHLTYPE(CLUSSDR) TRPTYPE(TCP)  
CONNNAME(REPOS1.COM) CLUSTER(PRICECHECK) DESCR('Cluster-sender channel to  
repository queue manager')
```

4. 安裝價格檢查應用程式。
5. 在所有伺服器佇列管理程式上定義 PRICEQ 佇列。

在每一個上發出下列指令：

```
DEFINE QLOCAL(PRICEQ) CLUSTER(PRICECHECK)
```

6. 限制查詢所使用的通道數

在查詢佇列管理程式上，我們會透過對每一個查詢佇列管理程式發出下列指令來限制使用的作用中通道數目：

```
ALTER QMGR CLWLMRUC(2)
```

7. 如果您尚未這樣做，請在 WebSphere MQ for z/OS 上啟動通道起始程式。在所有平台上，啟動接聽器程式。

接聽器程式會接聽送入的網路要求，並在需要時啟動叢集接收端通道。

結果

第 230 頁的圖 53 顯示此作業所設定的叢集。

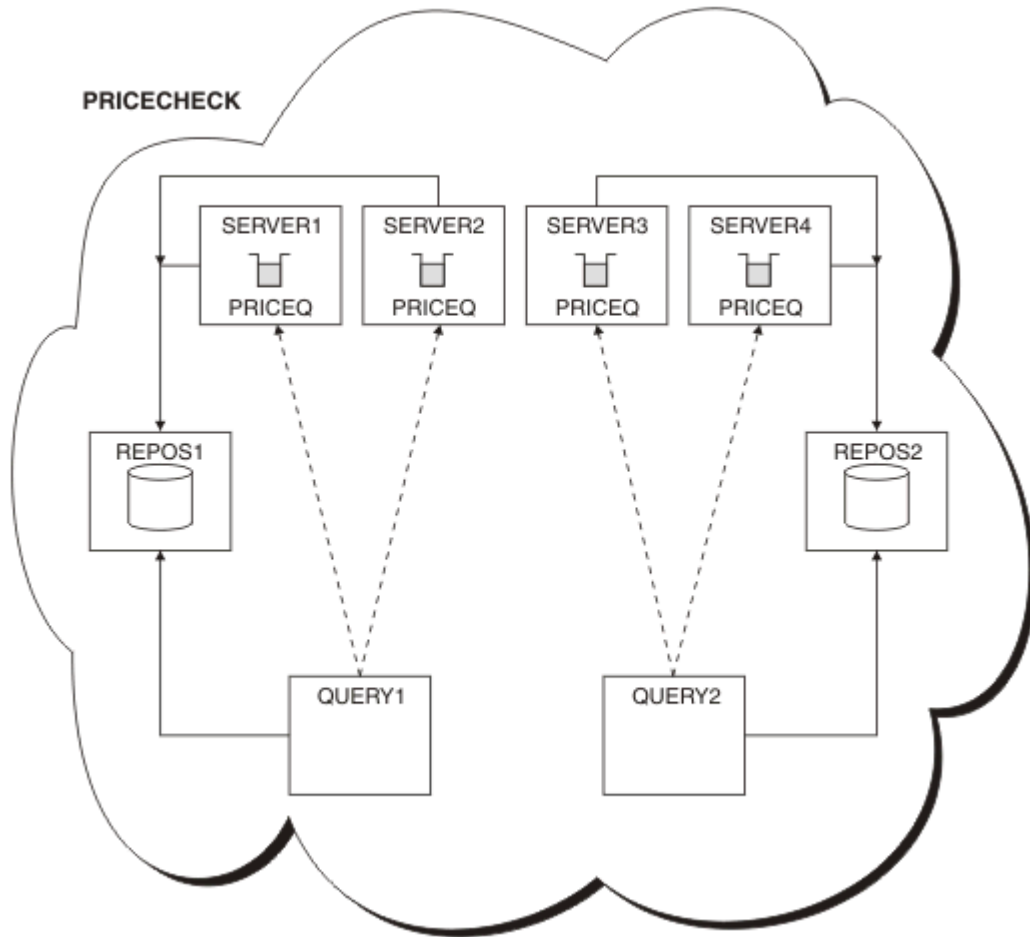


圖 53: PRICECHECK 叢集，具有四個伺服器佇列管理程式、兩個儲存庫及兩個查詢佇列管理程式

雖然 PRICECHECK 叢集中有四個 PRICEQ 佇列實例可用，但每一個查詢佇列管理程式只會使用其中兩個實例中的兩個實例。例如，QUERY1 佇列管理程式只具有 SERVER1 及 SERVER2 佇列管理程式的作用中通道。如果 SERVER1 變成無法使用，則 QUERY1 佇列管理程式會開始使用另一個佇列管理程式，例如 SERVER3。

下一步

雖然 PRICECHECK 叢集中有四個 PRICEQ 佇列實例可用，但每一個查詢佇列管理程式只會使用其中兩個實例中的兩個實例。例如，QUERY1 佇列管理程式只具有 SERVER1 及 SERVER2 佇列管理程式的作用中通道。如果 SERVER1 變成無法使用，則 QUERY1 佇列管理程式會開始使用另一個佇列管理程式，例如 SERVER3。

新增管理佇列的更強大佇列管理程式

遵循這些指示，透過在洛杉磯及紐約執行庫存系統來提供額外容量，洛杉磯可以處理兩倍於紐約的訊息數目。

開始之前

註: 如果要將叢集的變更延伸到整個叢集，至少必須一律有一個完整儲存庫可用。在啟動這項作業之前，請確定您的儲存庫可供使用。

測試情境：

- 已依照第 164 頁的『將佇列管理程式加入叢集中』中的說明來設定 INVENTORY 叢集。它包含三個佇列管理程式：LONDON 及 NEWYORK 兩者都保留完整儲存庫，PARIS 保留局部儲存庫並放置來自 INVENTQ 的訊息。庫存應用程式在紐約連接至 NEWYORK 佇列管理程式的系統上執行。應用程式是由 INVENTQ 佇列上的訊息到達所驅動。
- 洛杉磯正在開設一家新商店。為了提供額外容量，您想要在洛杉磯及紐約執行庫存系統。新的佇列管理程式可以處理兩倍於紐約的訊息。

關於這項作業

請遵循下列步驟，以新增管理佇列的功能更強大的佇列管理程式。

程序

1. 決定哪一個完整儲存庫 LOSANGELES 會先參照。
2. 叢集中的每個佇列管理程式都必須參照一個或其他完整儲存庫，才能收集叢集的相關資訊。它會建置自己的局部儲存庫。您選擇的儲存庫沒有特別重要的意義。在此範例中，選擇 NEWYORK。一旦新佇列管理程式已加入叢集，它就會與這兩個儲存庫進行通訊。

```
DEFINE CHANNEL (INVENTORY.NEWYORK) CHLTYPE (CLUSSDR) TRPTYPE (TCP)
CONNNAME (NEWYORK.CHSTORE.COM) CLUSTER (INVENTORY)
DESCR ('Cluster-sender channel from LOSANGELES to repository at NEWYORK')
```

3. 在佇列管理程式 LOSANGELES 上定義 CLUSRCVR 通道。

叢集中的每個佇列管理程式都必須定義叢集接收端通道，它可以在該通道上接收訊息。在 LOSANGELES 上，定義：

```
DEFINE CHANNEL (INVENTORY.LOSANGELES) CHLTYPE (CLUSRCVR) TRPTYPE (TCP)
CONNNAME (LOSANGELES.CHSTORE.COM) CLUSTER (INVENTORY)
DESCR ('Cluster-receiver channel for queue manager LOSANGELES')
CLWLWGHT (2)
```

叢集接收端通道會通告佇列管理程式的可用性，以接收來自叢集 INVENTORY 中其他佇列管理程式的訊息。將 CLWLWGHT 設為 2，可確保洛杉磯佇列管理程式取得的庫存訊息數是紐約的兩倍（當 NEWYORK 的通道設為 1 時）。

4. 變更佇列管理程式 NEWYORK 上的 CLUSRCVR 通道。

請確定洛杉磯佇列管理程式取得的庫存訊息數是紐約的兩倍。變更叢集接收端通道的定義。

```
ALTER CHANNEL (INVENTORY.NEWYORK) CHLTYPE (CLUSRCVR) CLWLWGHT (1)
```

5. 請檢閱庫存應用程式，以取得訊息親緣性。

在繼續之前，請確定庫存應用程式對訊息的處理順序沒有任何相依關係。

6. 在洛杉磯的系統上安裝庫存應用程式

7. 定義叢集佇列 INVENTQ。

INVENTQ 佇列（已由 NEWYORK 佇列管理程式管理）也將由 LOSANGELES 管理。在 LOSANGELES 佇列管理程式上定義它，如下所示：

```
DEFINE QLOCAL (INVENTQ) CLUSTER (INVENTORY)
```

現在，您已完成所有定義，如果尚未完成，請在 WebSphere MQ for z/OS 上啟動通道起始程式。在所有平台上，在佇列管理程式 LOSANGELES 上啟動接聽器程式。接聽器程式會接聽送入的網路要求，並在需要時啟動叢集接收端通道。

結果

第 230 頁的『新增管理佇列的更強大佇列管理程式』顯示此作業所設定的叢集。

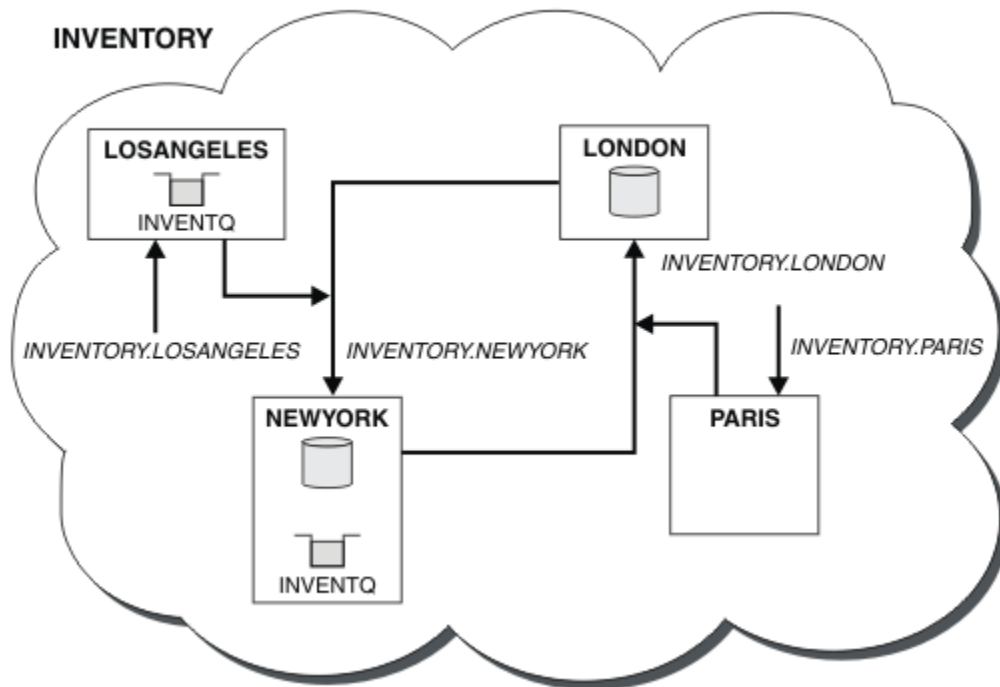


圖 54: 具有四個佇列管理程式的 *INVENTORY* 叢集

叢集的這項修改已完成，您無需變更佇列管理程式 *LONDON* 及 *PARIS*。這些佇列管理程式中的儲存庫會自動更新為能夠傳送訊息至 *INVENTQ* 所需的資訊，網址為 *LOSANGELES*。

下一步

INVENTQ 佇列及庫存應用程式在叢集中的兩個佇列管理程式上進行管理。此配置會增加其可用性，加快訊息的傳輸量，並容許在兩個佇列管理程式之間分散工作量。由 *LOSANGELES* 或 *NEWYORK* 放置至 *INVENTQ* 的訊息，會盡可能由本端佇列管理程式上的實例處理。*LONDON* 或 *PARIS* 放置的訊息會遞送至 *LOSANGELES* 或 *NEWYORK*，並將兩倍的訊息傳送至 *LOSANGELES*。

應用程式設計和叢集

您不需要進行任何程式設計變更，即可利用相同佇列的多個實例。不過，除非將一連串訊息傳送至佇列的相同實例，否則部分程式無法正常運作。

應用程式可以使用 *MQOPEN* 呼叫來開啟佇列。應用程式會使用 *MQPUT* 呼叫，將訊息放入開啟的佇列中。應用程式可以使用 *MQPUT1* 呼叫，將單一訊息放到尚未開啟的佇列中。

如果您設定的叢集具有相同佇列的多個實例，則沒有特定的應用程式設計考量。不過，若要從叢集作業的工作量管理層面獲益，您可能需要修改應用程式。如果您設定的網路中有多個相同佇列的定義，請檢閱應用程式的訊息親緣性。

例如，假設您有兩個應用程式以問答形式依賴在它們之間流動的一系列訊息。您可能想要回答回到傳送問題的相同佇列管理程式。工作量管理常式不會將訊息傳送至任何管理回覆佇列副本的佇列管理程式，這是很重要的。

您可能有需要依序處理訊息的應用程式 (例如，傳送必須依序擷取之訊息批次的資料庫抄寫應用程式)。使用分段訊息也可能導致親緣性問題。

開啟目標佇列的本端或遠端版本

請注意佇列管理程式如何選擇使用目標佇列的本端或遠端版本。

1. 佇列管理程式會開啟目標佇列的本端版本以讀取訊息，或設定佇列的屬性。

2. 如果至少符合下列其中一個條件，則佇列管理程式會開啟要寫入訊息的目標佇列的任何實例：

- 目標佇列的本端版本不存在。
- 佇列管理程式會在 ALTER QMGR 上指定 CLWLUSEQ (ANY)。
- 佇列管理程式上的佇列指定 CLWLUSEQ (ANY)。

處理訊息親緣性

訊息親緣性很少是良好程式設計的一部分。您需要移除訊息親緣性，才能完全使用叢集作業。如果無法移除訊息親緣性，您可以強制使用相同的通道將相關訊息遞送至相同的佇列管理程式。

如果您有具有訊息親緣性的應用程式，請先移除親緣性，然後再開始使用叢集。

移除訊息親緣性可改善應用程式的可用性。應用程式會將具有訊息親緣性的訊息批次傳送至佇列管理程式。在只接收部分批次之後，佇列管理程式會失敗。傳送端佇列管理程式必須等待它回復並處理不完整的訊息批次，才能傳送更多訊息。

移除訊息親緣性也會改善應用程式的可調整性。當等待後續訊息時，具有親緣性的訊息批次可以鎖定目的地佇列管理程式中的資源。這些資源可能長時間保持鎖定，導致其他應用程式無法執行其工作。

此外，訊息親緣性會阻止叢集工作量管理常式對佇列管理程式進行最佳選擇。

若要移除親緣性，請考量下列可能性：

- 在訊息中攜帶狀態資訊
- 維護永久儲存體中可供任何佇列管理程式存取的狀態資訊，例如在 Db2 資料庫中
- 抄寫唯讀資料，以便多個佇列管理程式可以存取它

如果不適合修改應用程式來移除訊息親緣性，則有許多可能的問題解決方案。

命名 MQOPEN 呼叫上的特定目的地

指定每一個 MQOPEN 呼叫的遠端佇列名稱和佇列管理程式名稱，使用該物件控點放入佇列中的所有訊息都會移至相同的佇列管理程式，這可能是本端佇列管理程式。

在每一個 MQOPEN 呼叫上指定遠端佇列名稱和佇列管理程式名稱有缺點：

- 不執行工作量平衡。您不會利用叢集工作量平衡的好處。
- 如果目標佇列管理程式位於遠端，且有多個通道通往它，則訊息可能採用不同的路徑，且仍不會保留訊息序列。
- 如果您的佇列管理程式具有與目的地佇列管理程式同名之傳輸佇列的定義，則訊息會進入該傳輸佇列，而非叢集傳輸佇列。

在回覆目的地佇列管理程式欄位中傳回佇列管理程式名稱

容許接收批次中第一個訊息的佇列管理程式在其回應中傳回其名稱。它會使用訊息描述子的 ReplyToQMgr 欄位來執行此動作。然後，傳送端的佇列管理程式可以擷取回覆目的地佇列管理程式名稱，並在所有後續訊息上指定它。

使用回應中的 ReplyToQMgr 資訊有下列缺點：

- 發出要求的佇列管理程式必須等待其第一則訊息的回應
- 在傳送後續訊息之前，您必須撰寫其他程式碼來尋找並使用 ReplyToQMgr 資訊
- 如果佇列管理程式有多個路徑，則可能不會保留訊息序列

在 MQOPEN 呼叫上設定 MQOO_BIND_ON_OPEN 選項

在 MQOPEN 呼叫上使用 MQOO_BIND_ON_OPEN 選項，強制將所有訊息放置到相同的目的地。當搭配使用 訊息群組 與叢集時，必須指定 MQOO_BIND_ON_OPEN 或 MQOO_BIND_ON_GROUP，以確保在相同目的地處理群組中的所有訊息。

透過開啟佇列並指定 MQOO_BIND_ON_OPEN，您可以強制將傳送至此佇列的所有訊息傳送至佇列的相同實例。MQOO_BIND_ON_OPEN 會將所有訊息連結至相同的佇列管理程式，也會連結至相同的路徑。例如，如

果有一個 IP 路徑和一個 NetBIOS 路徑通往相同的目的地，當開啟佇列時，會選取其中一個，且使用所取得的物件控點，將所有放入相同佇列的訊息都允許使用這個選項。

透過指定 MQOO_BIND_ON_OPEN，您可以強制將所有訊息遞送至相同的目的地。因此，具有訊息親緣性的應用程式不會中斷。如果目的地無法使用，則訊息會保留在傳輸佇列上，直到再次變成可用為止。

當您開啟佇列時，如果在物件描述子中指定佇列管理程式名稱，也會套用 MQOO_BIND_ON_OPEN。可能有多個指向指定佇列管理程式的路徑。例如，可能有多個網路路徑，或另一個佇列管理程式可能已定義別名。如果您指定 MQOO_BIND_ON_OPEN，則會在開啟佇列時選取路徑。

註：這是建議的技術。不過，在佇列管理程式通告叢集佇列別名的多重躍點配置中，它無法運作。在應用程式針對不同訊息群組使用相同佇列管理程式上的不同佇列的情況下，它也不會有所幫助。

在 MQOPEN 呼叫上指定 MQOO_BIND_ON_OPEN 的替代方案是修改佇列定義。在佇列定義上，指定 DEFBIND(OPEN)，並容許 MQOPEN 呼叫上的 DefBind 選項預設為 MQOO_BIND_AS_Q_DEF。

在 MQOPEN 呼叫上設定 MQOO_BIND_ON_GROUP 選項

在 MQOPEN 呼叫上使用 MQOO_BIND_ON_GROUP 選項，強制將群組中的所有訊息放置到相同的目的地。當搭配使用 訊息群組 與叢集時，必須指定 MQOO_BIND_ON_OPEN 或 MQOO_BIND_ON_GROUP，以確保在相同目的地處理群組中的所有訊息。

透過開啟佇列並指定 MQOO_BIND_ON_GROUP，您可以強制將群組中傳送至此佇列的所有訊息傳送至佇列的相同實例。MQOO_BIND_ON_GROUP 會將群組中的所有訊息連結至相同的佇列管理程式，也會連結至相同的路徑。例如，如果有一個 IP 路徑和 NetBIOS 路徑指向相同的目的地，則會在開啟佇列時選取其中一個，並允許使用取得的物件控點，將群組中的所有訊息放入相同的佇列中。

透過指定 MQOO_BIND_ON_GROUP，您可以強制將群組中的所有訊息遞送至相同的目的地。因此，具有訊息親緣性的應用程式不會中斷。如果目的地無法使用，則訊息會保留在傳輸佇列上，直到再次變成可用為止。

當您開啟佇列時，如果在物件描述子中指定佇列管理程式名稱，也會套用 MQOO_BIND_ON_GROUP。可能有多個指向指定佇列管理程式的路徑。例如，可能有多個網路路徑，或另一個佇列管理程式可能已定義別名。如果您指定 MQOO_BIND_ON_GROUP，則會在開啟佇列時選取路徑。

若要讓 MQOO_BIND_ON_GROUP 生效，您必須在 MQPUT 上包括 MQPMO_LOGICAL_ORDER 放置選項。您可以在訊息的 MQMD 中設定 **GroupId** 為 MQGI_NONE，並且必須在訊息的 MQMD **MsgFlags** 欄位內包括下列訊息旗標：

- 群組中的最後一則訊息: MQMF_LAST_MSG_IN_GROUP
- 群組中的所有其他訊息: MQMF_MSG_IN_GROUP

如果指定 MQOO_BIND_ON_GROUP 但未分組訊息，則行為相當於 [MQOO_BIND_NOT_FIXED](#)。

註：這是確保群組中的訊息傳送至相同目的地的建議技術。不過，在佇列管理程式通告叢集佇列別名的多重躍點配置中，它無法運作。

在 MQOPEN 呼叫上指定 MQOO_BIND_ON_GROUP 的替代方案是修改佇列定義。在佇列定義上，指定 DEFBIND(GROUP)，並容許 MQOPEN 呼叫上的 DefBind 選項預設為 MQOO_BIND_AS_Q_DEF。

撰寫自訂叢集工作量結束程式

您可以撰寫叢集工作量結束程式來規避訊息親緣性問題，而不修改應用程式。撰寫叢集工作量結束程式並不容易，也不是建議的解決方案。程式必須設計成透過檢查訊息內容來辨識親緣性。辨識親緣性之後，程式必須強制工作量管理公用程式將所有相關訊息遞送至相同的佇列管理程式。

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 WebSphere MQ 基本概念的透徹瞭解，例如良好的應用程式管理及網路設計。在繼續之前，請確定您已熟悉下列相關主題中的資訊。

相關概念

[叢集](#)

[交互通訊概念](#)

[叢集運作方式](#)

叢集作業: 重疊叢集的特殊考量

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

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 150 頁的『重疊叢集』](#) 及 [第 217 頁的『配置叢集之間的訊息路徑』](#)。

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 WebSphere 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 149 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：

- 如果您有同心叢集，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](#)。
- 如果部分佇列管理程式將由不同團隊管理。如需相關資訊，請參閱前一節 [第 235 頁的『叢集所有權』](#)。
- 從組織或地理角度而言是否有意義。
- 如果對等叢集使用名稱解析，例如在現有叢集中實作 SSL 或 TLS 時。

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 teo 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。

相關概念

第 149 頁的『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。

叢集作業: 拓撲設計考量

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

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 236 頁的『叢集作業基礎架構的效能』](#)
- [第 236 頁的『完整儲存庫』](#)
- [第 237 頁的『應用程式應該在完整儲存庫上使用佇列嗎?』](#)
- [第 237 頁的『管理通道定義』](#)
- [第 238 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 WebSphere MQ 中用於發佈/訂閱傳訊的 IBM WebSphere 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 WebSphere 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 [第 138 頁的『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 WebSphere 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 149 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），請參閱第 221 頁的『使用叢集進行工作量管理』、[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 WebSphere 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 WebSphere 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 WebSphere 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 239 頁的圖 55 是大型 IBM WebSphere 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。

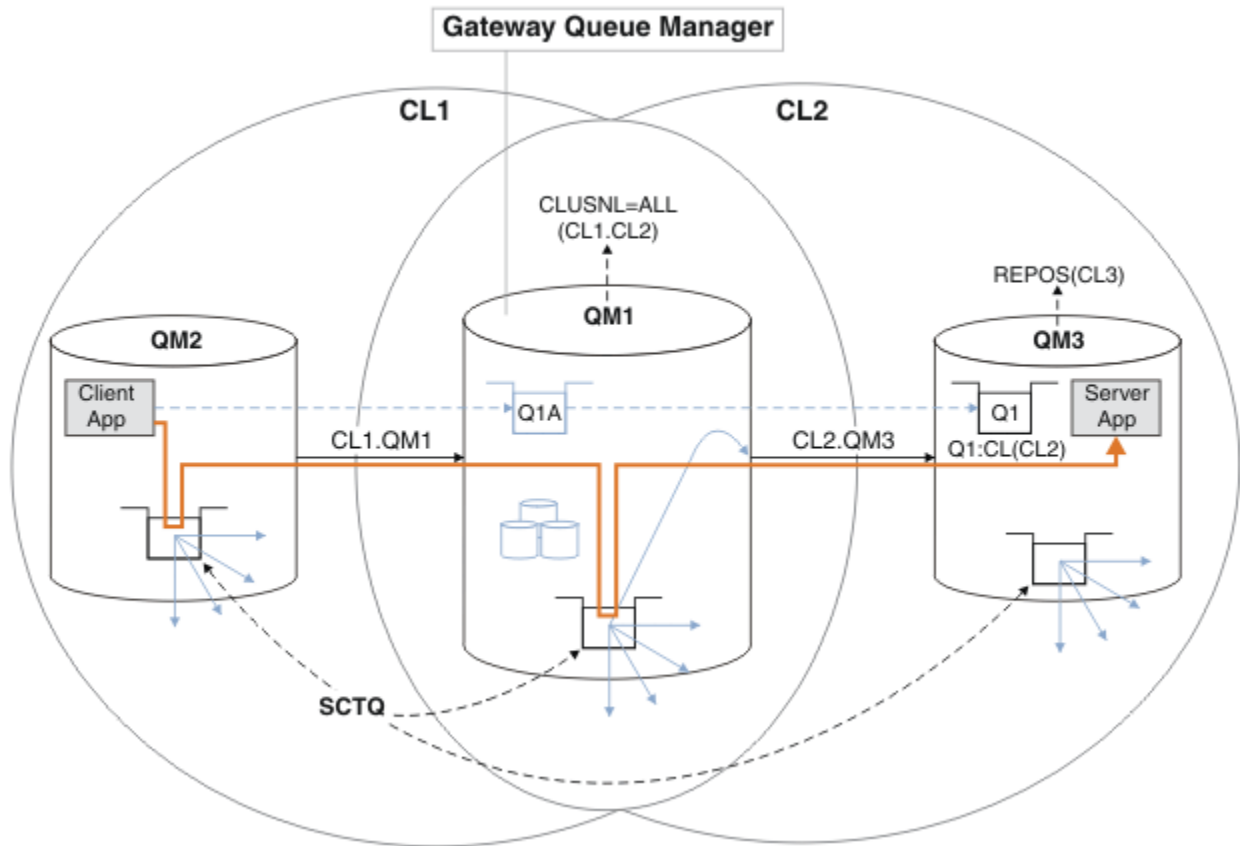
從效能角度來看，單一佇列不是問題。一般傳輸佇列通常不代表效能瓶頸。閘道上的訊息傳輸量很大程度上取決於與其連接之通道的效能。傳輸量通常不受佇列數目或使用通道之佇列上的訊息數目影響。

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 239 頁的圖 55 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 55: 使用 IBM WebSphere MQ 叢集部署至中心及分支架構的主從架構應用程式

在第 239 頁的圖 55 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 239 頁的圖 55 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；請參閱第 167 頁的『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 Version 7.5 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，並定義從每一個傳輸佇列傳送訊息的叢集傳送端通道；請參閱 [第 170 頁的『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。它為每個叢集傳送端通道定義不同的叢集傳輸佇列；請參閱 [第 187 頁的『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。傳輸佇列的組合是在 [第 170 頁的『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。

[第 170 頁的『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。您可以使用分散式佇列中的遠端佇列定義，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 [第 173 頁的『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』](#) 的配置中，新增另一個叢集，並使佇列管理程式和叢集佇列成為新叢集的成員，以產生限制。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 135 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 235 頁的『叢集作業: 重疊叢集的特殊考量』](#)

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

[第 143 頁的『叢集傳輸佇列及叢集傳送端通道』](#)

叢集佇列管理程式之間的訊息儲存在叢集傳輸佇列上，並由叢集傳送端通道轉遞。

[第 150 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。

相關工作

[授權將訊息放置在遠端叢集佇列上](#)

[第 167 頁的『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』](#)

修改使用閘道佇列管理程式之重疊叢集的配置。從閘道佇列管理程式將修改訊息傳送至應用程式之後，不會使用與其他叢集訊息相同的傳輸佇列或通道。解決方案使用叢集佇列遠端定義，以及個別傳送端通道和傳輸佇列。

[第 170 頁的『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』](#)

修改使用閘道佇列管理程式之重疊叢集的配置。從閘道佇列管理程式將修改訊息傳送至應用程式之後，不會使用與其他叢集訊息相同的傳輸佇列或通道。將修改訊息傳送至應用程式之後，解決方案會使用額外的叢集傳輸佇列，將訊息資料流量分隔至叢集中的單一佇列管理程式。

[第 173 頁的『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』](#)

修改使用閘道佇列管理程式之重疊叢集的配置。從閘道佇列管理程式將修改訊息傳送至應用程式之後，不會使用與其他叢集訊息相同的傳輸佇列或通道。將修改訊息傳送至應用程式之後，解決方案會使用其他叢集，將訊息與特定叢集佇列隔離。

[第 187 頁的『將預設值變更為個別叢集傳輸佇列以隔離訊息資料流量』](#)

您可以變更佇列管理程式將叢集佇列或主題的訊息儲存在傳輸佇列上的預設方式。變更預設值可讓您隔離閘道佇列管理程式上的叢集訊息。

[第 182 頁的『使用閘道佇列管理程式建立兩個重疊的叢集』](#)

遵循作業中的指示，使用閘道佇列管理程式來建構重疊叢集。使用叢集作為下列範例的起點，將訊息從一個應用程式隔離至叢集中的其他應用程式。

[第 217 頁的『配置叢集之間的訊息路徑』](#)

使用閘道佇列管理程式將叢集連接在一起。透過在閘道佇列管理程式上定義叢集佇列或叢集佇列管理程式別名，讓所有叢集都可以看見佇列或佇列管理程式。

第 241 頁的『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。配置多個叢集傳輸佇列適用於 z/OS 以外的平台。

相關參考

第 360 頁的『安全』

使用 `qm.ini` 檔案中的 Security 段落來指定「物件權限管理程式 (OAM)」的選項。

`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。配置多個叢集傳輸佇列適用於 z/OS 以外的平台。

開始之前

檢閱 [第 244 頁的『如何選擇要使用的叢集傳輸佇列類型』](#)。

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。

1. 叢集訊息傳送的預設叢集傳輸佇列為何?
 - a. 一般叢集傳輸佇列 `SYSTEM.CLUSTER.TRANSMIT.QUEUE`。
 - b. 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。
2. 對於您決定手動建立的叢集傳輸佇列，您還有兩個選項:
 - a. 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 242 頁的圖 56](#)。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。

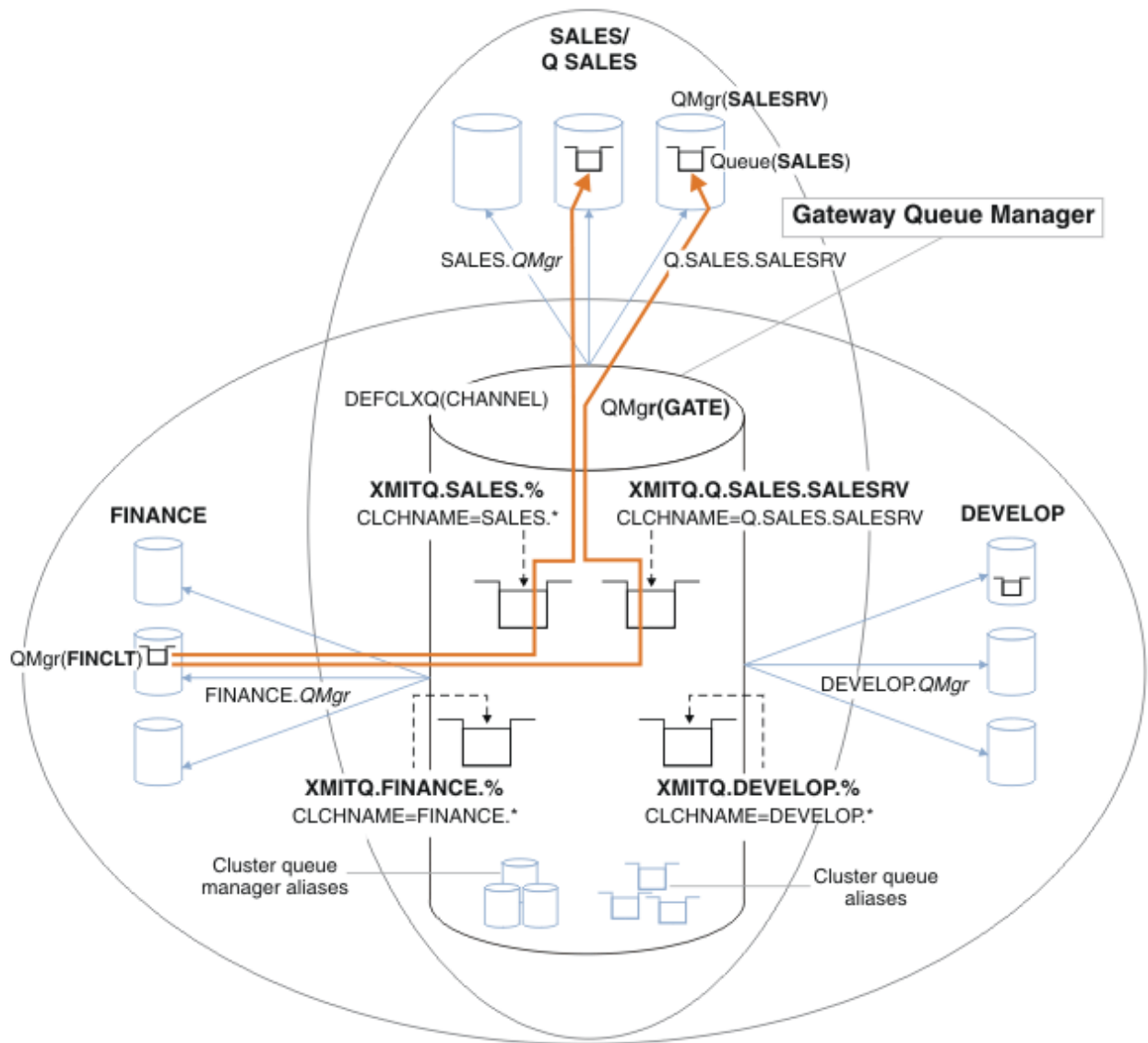


圖 56: 不同部門 IBM WebSphere MQ 叢集的特定傳輸佇列範例

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 245 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:

- 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。
 - 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。
 - 請參閱 [第 235 頁的『叢集作業: 重疊叢集的特殊考量』](#)。
 - 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。
 - 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 245 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 143 頁的『叢集傳輸佇列及叢集傳送端通道』](#)

叢集佇列管理程式之間的訊息儲存在叢集傳輸佇列上，並由叢集傳送端通道轉遞。

[第 238 頁的『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。

[第 135 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 235 頁的『叢集作業: 重疊叢集的特殊考量』](#)

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

[第 150 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。

相關工作

[第 167 頁的『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』](#)

修改使用閘道佇列管理程式之重疊叢集的配置。從閘道佇列管理程式將修改訊息傳送至應用程式之後，不會使用與其他叢集訊息相同的傳輸佇列或通道。解決方案使用叢集佇列遠端定義，以及個別傳送端通道和傳輸佇列。

[第 170 頁的『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』](#)

修改使用閘道佇列管理程式之重疊叢集的配置。從閘道佇列管理程式將修改訊息傳送至應用程式之後，不會使用與其他叢集訊息相同的傳輸佇列或通道。將修改訊息傳送至應用程式之後，解決方案會使用額外的叢集傳輸佇列，將訊息資料流量分隔至叢集中的單一佇列管理程式。

[第 173 頁的『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』](#)

修改使用閘道佇列管理程式之重疊叢集的配置。從閘道佇列管理程式將修改訊息傳送至應用程式之後，不會使用與其他叢集訊息相同的傳輸佇列或通道。將修改訊息傳送至應用程式之後，解決方案會使用其他叢集，將訊息與特定叢集佇列隔離。

[第 187 頁的『將預設值變更為個別叢集傳輸佇列以隔離訊息資料流量』](#)

您可以變更佇列管理程式將叢集佇列或主題的訊息儲存在傳輸佇列上的預設方式。變更預設值可讓您隔離閘道佇列管理程式上的叢集訊息。

[第 182 頁的『使用閘道佇列管理程式建立兩個重疊的叢集』](#)

遵循作業中的指示，使用閘道佇列管理程式來建構重疊叢集。使用叢集作為下列範例的起點，將訊息從一個應用程式隔離至叢集中的其他應用程式。

[第 217 頁的『配置叢集之間的訊息路徑』](#)

使用閘道佇列管理程式將叢集連接在一起。透過在閘道佇列管理程式上定義叢集佇列或叢集佇列管理程式別名，讓所有叢集都可以看見佇列或佇列管理程式。

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 Version 7.5 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 Version 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: 如果您將專用 `SYSTEM.CLUSTER.TRANSMIT.QUEUES` 與從舊版產品升級的佇列管理程式搭配使用，請確保 `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE` 已將 `SHARE/NOSHARE` 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 235 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 WebSphere 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 WebSphere 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，並與其他混合訊息流程共用叢集傳輸佇列，則值得配置特殊訊息路徑，以將它與其他訊息流程隔離；請參閱 [第 173 頁的『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。有系統地將方法 [第 173 頁的『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』](#) 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 241 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 246 頁的圖 57 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 248 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。

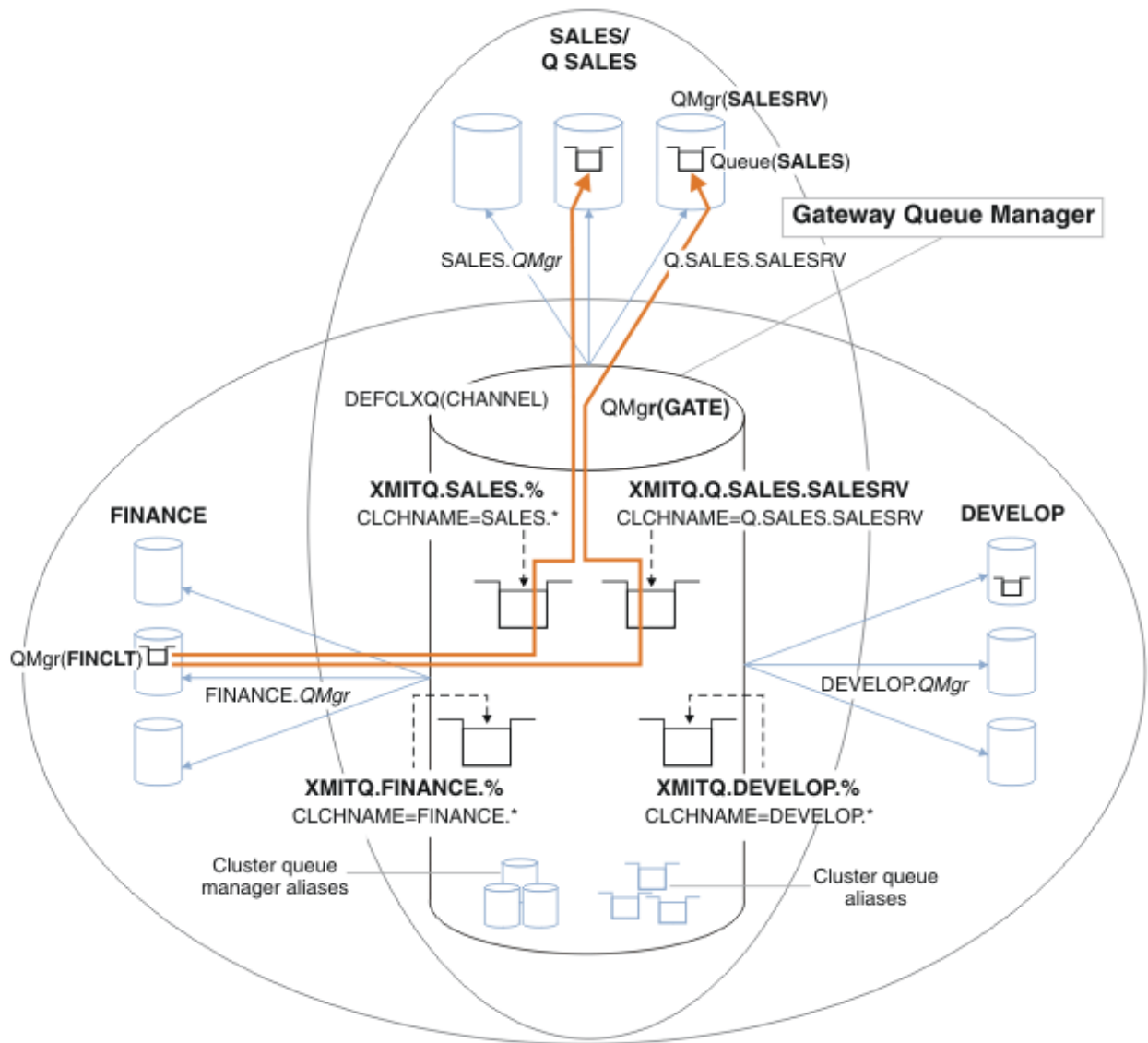


圖 57: 不同部門 IBM WebSphere MQ 叢集的特定傳輸佇列範例

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 250 頁的圖 58](#)。

- 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) FINANCE(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
```

當交換器完成時，請從 SALES 叢集中移除 SALES 佇列；請參閱 [第 250 頁的圖 58](#)。

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更生效。

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 248 頁的表 27](#) 中的每一個佇列管理程式名稱執行指令: `crtmqm -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 248 頁的表 27](#) 中的每一個佇列管理程式名稱執行指令: `stmqm 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 250 頁的圖 58](#) 中執行 **MQSC** 指令：

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

圖 58: 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 249 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。

```
amqsmon -m GATE -t statistics
```

- e. 幾個間隔的結果如下:

```
C:\Documents and Settings\Admin>amqsmon -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更生效。

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。請閱讀第 145 頁的『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』，以瞭解嘗試清空傳輸佇列的原因，然後再繼續任何進一步的動作。

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更生效。

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 253 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 254 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 253 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 254 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

[第 145 頁的『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。請閱讀第 145 頁的『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』，以瞭解嘗試清空傳輸佇列的原因，然後再繼續任何進一步的動作。

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
```

- 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。請閱讀第 145 頁的『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』，以瞭解嘗試清空傳輸佇列的原因，然後再繼續任何進一步的動作。

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 245 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。

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

runswchl 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對照到第 253 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 254 頁的『2』。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)
```

相關參考

[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 WebSphere MQ 叢集的指引。此資訊是根據客戶的測試和意見而提供的指引。

1. 第 256 頁的『[移動叢集中的物件](#)』(在叢集內移動物件而不安裝任何修正套件或 IBM WebSphere MQ 新版本的最佳作法)。
2. 第 257 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 **CLUSDR** 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 指令會干擾進行中的叢集，而此後每隔 27 天，當叢集物件自動將狀態更新傳送給所有相關的佇列管理程式時，會再次造成干擾。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 WebSphere 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 Version 6.0 移至 IBM WebSphere MQ Version 7.1 時, 如果佇列管理程式定義叢集主題, 則 IBM WebSphere MQ Version 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](#)。

相關概念

第 257 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。

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 WebSphere 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位置記載的罕見情況下, 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 257 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，以備日後系統發生問題時使用。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

[執行 REFRESH CLUSTER 時看到的應用程式問題](#)

[發佈/訂閱叢集的 REFRESH CLUSTER 考量](#)

相關參考

[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 WebSphere MQ 叢集的指引。此資訊是根據客戶的測試和意見而提供的指引。

IBM WebSphere MQ 叢集作業本身不是「高可用性」解決方案，但在某些情況下，可以使用它來改善使用 IBM WebSphere MQ 的服務可用性，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 WebSphere 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 [第 265 頁的『搭配使用 WebSphere MQ 與高可用性配置』](#)、[第 289 頁的『建立多重實例佇列管理程式』](#) 及下一節。只要叢集中的所有佇列管理程式都至少執行 IBM WebSphere MQ Version 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 WebSphere 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 WebSphere MQ 可以提供協助，但無法自動執行。IBM WebSphere 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。
 - 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 257 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 WebSphere 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。

叢集作業：監視

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

監視叢集中的應用程式訊息

通常，離開佇列管理程式的所有叢集訊息都會通過 `SYSTEM.CLUSTER.TRANSMIT.QUEUE`，而不論使用哪個叢集傳送端通道來傳輸訊息。每一個通道都會與所有其他叢集傳送端通道平行排除該通道的目標訊息。在此佇列上不斷增加的訊息建置可能指出一或多個通道有問題，必須加以調查：

- 必須針對叢集設計適當地監視佇列深度。
- 下列指令會傳回在傳輸佇列上等待多個訊息的所有通道：

```
DIS CHSTATUS(*) WHERE(XQMSGSA GT 1)
```

對於單一佇列上的所有叢集訊息，當通道開始填滿時，並不總是容易看到哪個通道有問題。使用此指令是查看哪個通道負責的簡單方法。

您可以將叢集佇列管理程式配置成具有多個傳輸佇列。如果您將佇列管理程式屬性 `DEFCLXQ` 變更為 `CHANNEL`，則每個叢集傳送端通道都會與不同的叢集傳輸佇列相關聯。或者，您可以手動配置個別傳輸佇列。若要顯示與叢集傳送端通道相關聯的所有叢集傳輸佇列，請執行下列指令：

```
DISPLAY CLUSQMGR (qmgrName) XMITQ
```

定義叢集傳輸佇列，以便它們遵循左側佇列名稱的固定詞幹型樣。然後，您可以使用通用佇列名稱來查詢 **DISPLAY CLUSMGR** 指令所傳回所有叢集傳輸佇列的深度：

```
DISPLAY QUEUE (qname*) CURDEPTH
```

監視叢集中的控制訊息

`SYSTEM.CLUSTER.COMMAND.QUEUE` 佇列用於處理佇列管理程式的所有叢集控制訊息，這些訊息由本端佇列管理程式產生，或從叢集中的其他佇列管理程式傳送至這個佇列管理程式。當佇列管理程式正確維護其叢集狀態時，此佇列會趨向零。不過，在某些狀況下，此佇列上的訊息深度可能會暫時增加：

- 佇列上有大量訊息表示叢集狀態流失。
- 進行重大變更時，容許佇列在這些變更之間安定下來。例如，移動儲存庫時，在移動第二個儲存庫之前，容許佇列達到零。

當此佇列上存在訊息待辦事項時，不會處理叢集狀態或叢集相關指令的更新。如果長時間未從此佇列中移除訊息，則需要進行進一步調查，起始方式是檢驗佇列管理程式錯誤日誌，這可能會說明造成此狀況的處理程序。

`SYSTEM.CLUSTER.REPOSITORY.QUEUE` 會將叢集儲存庫快取資訊保留為一些訊息。訊息通常一律存在於此佇列中，而更多訊息則適用於較大的叢集。因此，此佇列上的訊息深度不是值得關注的問題。

監視日誌

在最初發生問題之後，由於快取資訊及叢集作業的分散式本質，叢集中發生的問題可能在許多天 (甚至數個月) 內不會對應用程式顯示外部症狀。不過，通常會在 IBM WebSphere MQ 錯誤日誌中報告原始問題。基於此原因，請務必主動監視這些日誌中所撰寫的任何與叢集作業相關的訊息。必須閱讀並瞭解這些訊息，並在必要時採取任何動作。

例如：與叢集中佇列管理程式的通訊中斷可能會導致瞭解由於叢集透過重新發佈資訊而定期重新驗證叢集資源的方式而被刪除的特定叢集資源。[AMQ9465](#) 訊息會報告可能發生此類事件的警告。此訊息指出需要調查問題。

負載平衡的特殊考量

當叢集在佇列的兩個以上實例之間進行負載平衡時，消費端應用程式必須正在處理每一個實例上的訊息。如果一或多個消費端應用程式終止或停止處理訊息，叢集作業可能會繼續將訊息傳送至那些佇列實例。在此狀況下，除非應用程式再次正確運作，否則不會處理這些訊息。因此，監視應用程式是解決方案的重要部分，在該狀況下必須採取動作來重新遞送訊息。在此範例中可以找到自動化此類監視的機制範例：[The Cluster Queue Monitoring sample program \(AMQSCLM\)](#)。

可用性、回復及重新啟動

如果佇列管理程式失敗，請維護佇列可用性，讓您的應用程式具有高可用性，並在伺服器或儲存體失敗之後回復訊息。

透過使用用戶端重新連線，在佇列管理程式群組之間自動切換用戶端，或在佇列管理程式失敗之後切換至多重實例佇列管理程式的新作用中實例，來改善用戶端應用程式可用性。Java 的 WebSphere MQ 類別不支援自動用戶端重新連接。

在 Windows、UNIX、Linux 及 IBM i 平台上，將伺服器應用程式部署至多重實例佇列管理程式，該佇列管理程式配置為在多部伺服器上作為單一佇列管理程式執行；如果執行作用中實例的伺服器失敗，則執行會自動切換至不同伺服器上相同佇列管理程式的待命實例。如果您將伺服器應用程式配置成以佇列管理程式服務方式執行，當待命實例變成主動執行中的佇列管理程式實例時，它們會重新啟動。

您可以將 WebSphere MQ 配置為平台專用叢集作業解決方案的一部分，例如 微軟 Cluster Server 或 PowerHA for AIX (先前稱為 AIX 上的 HACMP)，以及其他 UNIX and Linux 叢集作業解決方案。

增加伺服器應用程式可用性的另一種方法是將伺服器應用程式部署至佇列管理程式叢集中的多部電腦。

傳訊系統可確保輸入系統的訊息會遞送至其目的地。WebSphere MQ 可以在訊息從一個佇列管理程式移至另一個佇列管理程式時，使用 `dspmqrte` 指令來追蹤訊息的路徑。如果系統失敗，視失敗類型及系統的配置方式而定，可以用各種方式來回復訊息。

WebSphere MQ 可透過維護處理訊息接收、傳輸及遞送之佇列管理程式活動的回復日誌，來確保訊息不會遺失。它使用這些日誌來進行三種類型的回復：

1. 當您以計劃的方式停止 WebSphere MQ 時，請重新啟動回復。
2. 失敗回復，當失敗停止 WebSphere MQ 時。
3. 媒體回復，用來還原損壞的物件。

在所有情況下，回復會將佇列管理程式還原至佇列管理程式停止時的狀態，但會回復任何進行中的交易，並從佇列中移除佇列管理程式停止時進行中的任何更新項目。回復會還原所有持續訊息；在處理程序期間可能會遺失非持續訊息。

自動用戶端重新連線

您可以透過配置多個元件，讓用戶端應用程式自動重新連接，而無需撰寫任何其他程式碼。

自動用戶端重新連線是行內作業。會在用戶端應用程式中的任意時間點自動還原連線，並將用於開啟物件的控點全部還原。

相反地，手動重新連線需要用戶端應用程式使用 `MQCONN` 或 `MQCONNX` 重建連線，並重新開啟物件。自動用戶端重新連線適合許多（但並非全部）用戶端應用程式。

第 261 頁的表 28 列出必須安裝在用戶端工作站上的最早 IBM WebSphere MQ 用戶端支援版本。您必須將用戶端工作站升級至下列其中一個層次，應用程式才能使用自動用戶端重新連線。第 261 頁的表 29 列出啟用自動用戶端重新連線的其他需求。

使用重新連線選項的程式存取權，用戶端應用程式可以設定重新連線選項。除了 JMS 和 XMS 用戶端之外，如果用戶端應用程式可以存取重新連線選項，它也可以建立事件處理程式來處理重新連線事件。

現有的用戶端應用程式可能可以受益於重新連線支援，而無需重新編譯及鏈結：

- 若為非 JMS 用戶端，請設定 `mqclient.ini` 環境變數 `DefRecon`，以設定重新連線選項。使用 CCDT 來連接佇列管理程式。如果用戶端要連接多重實例佇列管理程式，請在 CCDT 中提供作用中及待命佇列管理程式實例的網址。
- 若為 JMS 用戶端，請在 Connection Factory 配置中設定重新連線選項。當您使用 WebSphere MQ 資源配接器或整合在 Java EE 環境中的 JMS 用戶端時，可能無法使用自動用戶端重新連線。部分受管理環境有一些限制，如需相關資訊，請參閱在 Java SE 及 Java EE 環境中使用自動用戶端重新連線。

註：適用於 Java 的 WebSphere MQ 類別不支援自動用戶端重新連線。

用戶端介面	用戶端	重新連線選項的程式存取權	重新連線支援
傳訊 API	C、C++、COBOL、Unmanaged Visual Basic、XMS (Windows 上未受管理的 XMS)	7.0.1	7.0.1
	JMS (JSE 及 Java EE 用戶端儲存器和受管理儲存器)	7.0.1.3	7.0.1.3
	WebSphere MQ for Java 類別	不支援	不支援
	受管理 XMS 及受管理 .NET 用戶端 :C#、Visual Basic	7.1	7.1
其他 API	Windows Communication Foundation (未受管理 ¹)	不支援	7.0.1
	Windows Communication Foundation (受管理 ¹)	不支援	不支援
	軸 1	不支援	不支援
	軸 2	不支援	7.0.1.3
	HTTP (Web 2.0)	不支援	7.0.1.3

1. 在 WCF 連結配置中設定受管理或未受管理模式。

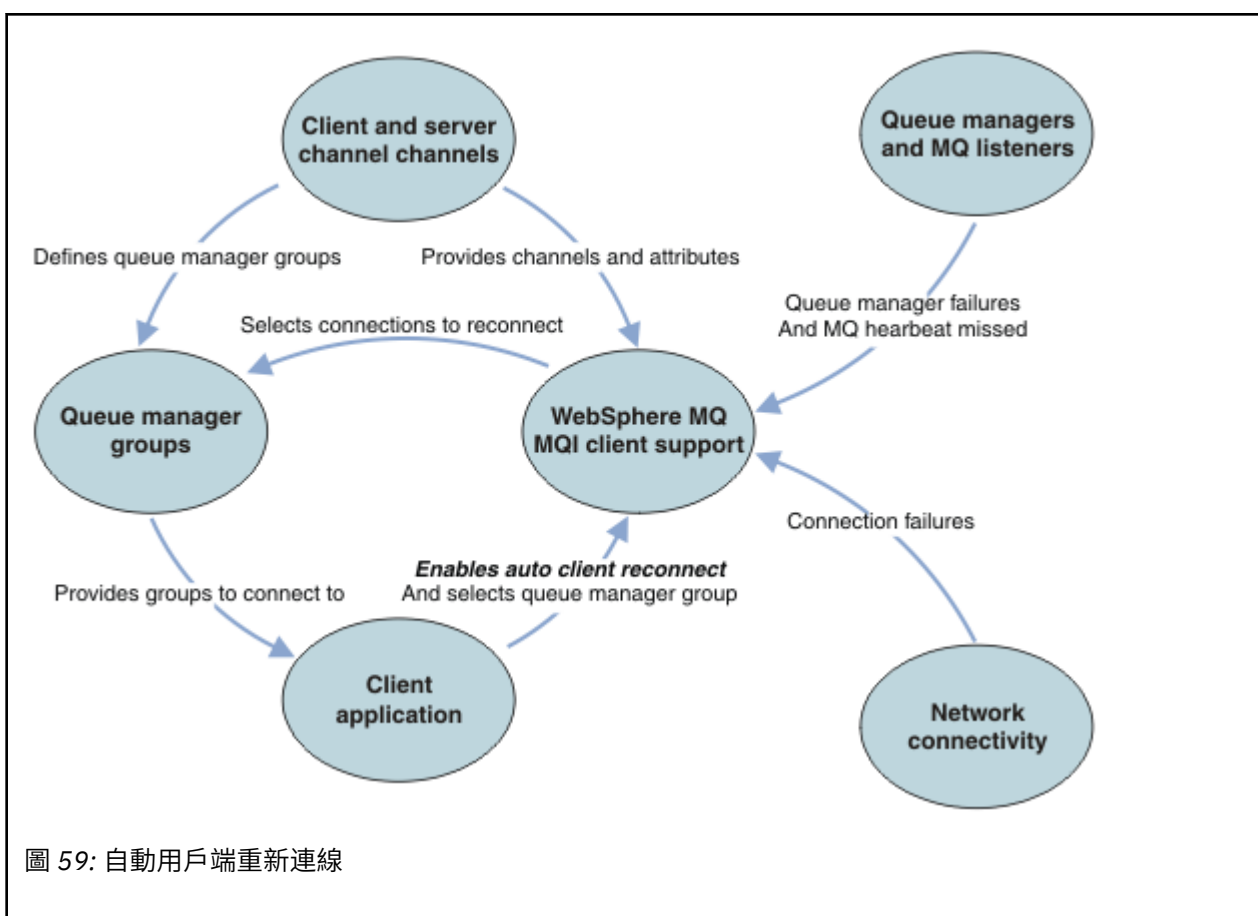
自動重新連線具有下列配置需求：

元件	需求	不符合需求的效果
WebSphere MQ MQI 用戶端安裝架構	請參閱 第 261 頁的表 28	MQRC_OPTIONS_ERROR
WebSphere MQ 伺服器安裝	層次 7.0.1	MQRC_OPTIONS_ERROR
通道	SHARECNV > 0	MQRC_ENVIRONMENT_ERROR
應用程式環境	必須是執行緒	MQRC_ENVIRONMENT_ERROR

表 29: 自動重新連線配置需求 (繼續)

元件	需求	不符合需求的效果
MQI	下列其中一項： <ul style="list-style-type: none"> MQCONN with MQCNO 選項 設為 MQCNO_RECONNECT 或 MQCNO_RECONNECT_Q_MGR。 Defrecon=YES QMGR 於 mqclient.ini 在 JMS 中，設定 Connection Factory 的 CLIENTRECONNECTOPTIONS 內容。 	MQCC_FAILED (當連線中斷或佇列管理程式結束或失敗時)。

第 262 頁的圖 59 顯示用戶端重新連線所涉及元件之間的主要互動。



用戶端應用程式

用戶端應用程式是 IBM WebSphere MQ MQI 用戶端。

- 依預設，不會自動重新連接用戶端。透過設定 MQCONN MQCNO 選項 MQCNO_RECONNECT 或 MQCNO_RECONNECT_Q_MGR，啟用自動用戶端重新連線。
- 許多應用程式以這樣的方式撰寫，使得它們能夠利用自動重新連線，而不需要額外的編碼。透過在 mqclient.ini 配置檔的通道段落中設定 DefRecon 屬性，啟用現有程式的自動重新連線，而不進行任何編碼變更。
- 請使用下列三個選項之一：

1. 請修改程式，使邏輯不受重新連線的影響。例如，您可能必須在同步點內發出 MQI 呼叫，然後重新提交已取消的交易。
2. 新增事件處理程式以偵測重新連線，並在重新建立連線時還原用戶端應用程式的狀態。
3. 不要啟用自動重新連線：請改為中斷用戶端連線，並發出新的 MQCONN 或 MQCONNX MQI 呼叫，尋找在相同佇列管理程式群組中執行的另一個佇列管理程式實例。

如需這三個選項的進一步詳細資料，請參閱第 329 頁的『應用程式回復』。

- 重新連接至同名的佇列管理程式，並不保證您已重新連接至佇列管理程式的相同實例。

使用 MQCNO 選項 MQCNO_RECONNECT_Q_MGR，可重新連接至相同佇列管理程式的實例。

- 用戶端可以登錄事件處理程式，以便通知它重新連線的狀態。無法使用傳入事件處理程式中的 MQHCONN。提供下列原因碼：

MQRC_RECONNECTING

連線失敗，系統正在嘗試重新連接。如果進行多次重新連接嘗試，您會收到多個 MQRC_RECONNECTING 事件。

MQRC_RECONNECTED

已進行重新連線，且已順利重新建立所有控點。

MQRC_RECONNECT_FAILED

重新連線失敗。

MQRC_RECONNECT_QMID_MISMATCH

可重新連接的連線已指定 MQCNO_RECONNECT_Q_MGR，且連線已嘗試重新連接至不同的佇列管理程式。

MQRC_RECONNECT_Q_MGR_REQD

用戶端程式中指定的選項 (例如 MQGET 呼叫中的 MQMO_MATCH_MSG_TOKEN) 需要重新連線至相同的佇列管理程式。

- 可重新連接的用戶端只能在連接之後自動重新連接。也就是說，如果 MQCONNX 呼叫本身失敗，則不會重試。例如，如果您收到來自 MQCONNX 的回覆碼 2543 - MQRC_STANDBY_Q_MGR，請在短暫延遲之後重新發出呼叫。

MQRC_RECONNECT_不相容

當應用程式嘗試使用 MQPMO_LOGICAL_ORDER (具有 MQPUT 及 MQPUT1) 或 MQGMO_LOGICAL_ORDER (具有 MQGET) 時，如果已設定重新連接選項，則會傳回此原因碼。傳回原因碼的原因是確保應用程式在這類情況下永不使用重新連接。

MQRC_CALL_INTERRUPTED

在執行「確定」呼叫期間，當連線中斷且用戶端重新連接時，會傳回此原因碼。同步點外部持續訊息的 MQPUT 也會導致將相同的原因碼傳回應用程式。

多重實例佇列管理程式

在多重實例佇列管理程式已使用自動用戶端重新連線來啟動其待命實例之後，簡化重新啟動 WebSphere MQ MQI 用戶端應用程式的作業。

多重實例佇列管理程式的待命實例通常位於與作用中實例不同的網址。在用戶端連線定義表 (CCDT) 中包含這兩個實例的網址。請為 **CONNAME** 參數提供網址清單，或為 CCDT 中的佇列管理程式定義多列。

通常，WebSphere MQ MQI 用戶端會重新連接至佇列管理程式群組中的任何佇列管理程式。有時您想要 WebSphere MQ MQI 用戶端只重新連接至相同的佇列管理程式。它可能與佇列管理程式有親緣性。您可以防止用戶端重新連接至不同的佇列管理程式。設定 MQCNO 選項 MQCNO_RECONNECT_Q_MGR。如果 WebSphere MQ MQI 用戶端重新連接至不同的佇列管理程式，則會失敗。如果您設定 MQCNO 選項 MQCNO_RECONNECT_Q_MGR，請不要在相同的佇列管理程式群組中包含其他佇列管理程式。如果用戶端所重新連接的佇列管理程式與它所連接的佇列管理程式不同，則用戶端會傳回錯誤。

佇列管理程式群組

您可以選取用戶端應用程式是否一律連接及重新連接至相同名稱的佇列管理程式、相同佇列管理程式，或用戶端連線表格中以相同 QMNAME 值定義的任何一組佇列管理程式。

- 用戶端通道定義中的佇列管理程式名稱屬性 QMNAME 是佇列管理程式群組的名稱。
- 在用戶端應用程式中，如果您將 MQCONN 或 MQCONNX QmgrName 參數的值設為佇列管理程式名稱，用戶端只會連接至具有該名稱的佇列管理程式。如果您在佇列管理程式名稱前面加上星號 (*)，用戶端會以相同的 QMNAME 值連接至佇列管理程式群組中的任何佇列管理程式。如需完整說明，請參閱 [CCDT 中的佇列管理程式群組](#)。

佇列共用群組

自動用戶端重新連線至 z/OS 佇列共用群組，會使用與任何其他環境相同的機制來重新連線。用戶端將重新連接至針對原始連線所配置的相同佇列管理程式選項。例如，當使用用戶端通道定義表時，管理者應該確定表格中的所有項目都解析為相同的 z/OS 佇列共用群組。

用戶端和伺服器通道定義

用戶端及伺服器通道定義會定義用戶端應用程式可以重新連接的佇列管理程式群組。這些定義會控管重新連線的選擇和計時，以及其他因素 (例如安全); 請參閱相關主題。考量重新連線的最相關通道屬性會列在兩個群組中:

用戶端連線屬性

連線親緣性 (AFFINITY)AFFINITY

連線親緣性。

用戶端通道加權 (CLNTWGHT)CLNTWGHT

用戶端通道加權。

連線名稱 (CONNAME)CONNAME

連線資訊。

活動訊號間隔 (HBINT)HBINT

活動訊號間隔。設定伺服器連線通道上的活動訊號間隔。

保持作用中間隔 (KAINT)KAINT

保持作用中間隔。在伺服器連線通道上設定保留作用中間隔。

請注意，KAINT 僅適用於 z/OS。

佇列管理程式名稱 (QMNAME)QMNAME

佇列管理程式名稱。

伺服器連線屬性

活動訊號間隔 (HBINT)HBINT

活動訊號間隔。設定用戶端連線通道上的活動訊號間隔。

保持作用中間隔 (KAINT)KAINT

保持作用中間隔。在用戶端連線通道上設定保留作用中間隔。

請注意，KAINT 僅適用於 z/OS。

KAINT 是網路層活動訊號，HBINT 是用戶端與佇列管理程式之間的 WebSphere MQ 活動訊號。將這些活動訊號設為較短的時間有兩個用途:

1. 透過模擬連線上的活動，負責關閉非作用中連線的網路層軟體不太可能關閉您的連線。
2. 如果連線關閉，則會縮短偵測到中斷連線之前的延遲。

預設 TCP/IP 保持作用中間隔為兩小時。請考量將 KAINT 和 HBINT 屬性設為較短的時間。請勿假設網路的正常行為符合自動重新連線的需求。例如，部分防火牆可以在 10 分鐘內關閉非作用中 TCP/IP 連線。

網路連線功能

用戶端的自動重新連線功能只會處理網路傳遞至 WebSphere MQ MQI 用戶端的網路失敗。

- IBM WebSphere MQ 看不見傳輸自動執行的重新連線。
- 設定 HBINT 有助於處理 WebSphere MQ 看不到的網路失敗。

佇列管理程式及 WebSphere MQ 接聽器

伺服器失敗、佇列管理程式失敗、網路連線功能失敗，以及管理者切換至另一個佇列管理程式實例，都會觸發用戶端重新連線。

- 如果您使用多重實例佇列管理程式，當您將控制從作用中佇列管理程式實例切換至待命實例時，會發生用戶端重新連線的其他原因。
- 使用預設 `endmqm` 指令結束佇列管理程式不會觸發自動用戶端重新連線。在 `endmqm` 指令上新增 `-r` 選項以要求自動用戶端重新連線，或新增 `-s` 選項以在關閉之後傳送至待命佇列管理程式實例。

WebSphere MQ MQI 用戶端自動重新連線支援

如果您在 WebSphere MQ MQI 用戶端中使用自動用戶端重新連線支援，用戶端應用程式會自動重新連接並繼續處理，而不會發出 MQCONN 或 MQCONNX MQI 呼叫來重新連接佇列管理程式。

- 下列其中一個出現項目會觸發自動用戶端重新連線：
 - 佇列管理程式失敗
 - 結束佇列管理程式並指定 `-r`，重新連接，`endmqm` 指令上的選項
- MQCONNX MQCNO 選項可控制您是否已啟用自動用戶端重新連線。這些選項在 [重新連接選項](#) 中說明。
- 自動用戶端重新連線會代表您的應用程式發出 MQI 呼叫，以還原連線控點及其他開啟物件的控點，以便您的程式在處理因連線中斷所產生的任何 MQI 錯誤之後，可以回復正常處理。請參閱第 330 頁的『[回復自動重新連接的用戶端](#)』。
- 如果您已為連線撰寫通道結束程式，則結束程式會收到這些額外的 MQI 呼叫。
- 您可以登錄重新連線事件處理程式，它會在重新連線開始及完成時觸發。

雖然重新連線不會超過一分鐘，但重新連線可能需要更長的時間，因為佇列管理程式可能有許多要管理的資源。在此期間，用戶端應用程式可能正在保留不屬於 WebSphere MQ 資源的鎖定。您可以配置逾時值來限制用戶端等待重新連線的時間。該值 (以秒為單位) 設定在 `mqclient.ini` 檔案中。

```
Channels:  
MQReconnectTimeout = 1800
```

在逾時過期之後，不會進行任何重新連線嘗試。當系統偵測到逾時已過期時，會傳回 MQRC_RECONNECT_FAILED 錯誤。

主控台訊息監視

有一些由佇列管理程式或通道起始程式發出的參考訊息，應該視為特別重要。這些訊息本身並不指出問題，但可能有助於追蹤，因為它們確實指出可能需要處理的潛在問題。

出現此訊息也可能指出使用者應用程式正在將大量訊息放入頁集，這可能是更大問題的症狀：

- PUT 的訊息 (例如不受控制的迴圈) 之使用者應用程式的問題。
- 來自佇列的 GET 訊息的使用者應用程式不再運作。

搭配使用 WebSphere MQ 與高可用性配置

如果您要在高可用性 (HA) 配置中操作 WebSphere MQ 佇列管理程式，您可以設定佇列管理程式來使用高可用性管理程式，例如 PowerHA for AIX (舊稱為 HACMP) 或 Microsoft Cluster Service (MSCS)，或使用 WebSphere MQ 多重實例佇列管理程式。

您需要注意下列配置定義：

佇列管理程式叢集

一個以上電腦上兩個以上佇列管理程式的群組，提供自動互連，並容許在它們之間共用佇列以進行負載平衡及備援。

HA 叢集

HA 叢集是由兩部以上電腦及資源 (例如磁碟及網路) 所組成的群組，它們會連接在一起，並以這樣的方式進行配置：如果其中一部電腦失敗，則高可用性管理程式 (例如 HACMP (UNIX) 或 MSCS (Windows))

會執行失效接手。失效接手會將應用程式的狀態資料從失敗的電腦傳送至叢集中的另一部電腦，並在該處重新起始其作業。這提供在 HA 叢集內執行之服務的高可用性。第 266 頁的『HA 叢集與佇列管理程式叢集的關係』中說明 IBM WebSphere MQ 叢集與 HA 叢集之間的關係。

多重實例佇列管理程式

在兩部以上電腦上配置相同佇列管理程式的實例。透過啟動多個實例，一個實例變成作用中實例，而其他實例變成待命實例。如果作用中實例失敗，則在不同電腦上執行的待命實例會自動接管。您可以使用多重實例佇列管理程式，根據 WebSphere MQ 來配置您自己的高可用性傳訊系統，而不需要叢集技術，例如 HACMP 或 MSCS。HA 叢集和多重實例佇列管理程式是讓佇列管理程式具有高可用性的替代方式。請勿透過將多重實例佇列管理程式放入 HA 叢集來結合它們。

多重實例佇列管理程式與 HA 叢集之間的差異

多重實例佇列管理程式及 HA 叢集是實現佇列管理程式高可用性的替代方式。以下是強調這兩種方法之間的差異的一些要點。

多重實例佇列管理程式包括下列特性：

- 整合至 WebSphere MQ 的基本失效接手支援
- 比 HA 叢集更快的失效接手
- 簡式配置及作業
- 與 WebSphere MQ 探險家整合

多重實例佇列管理程式的限制包括：

- 需要高可用性、高效能網路儲存體
- 更複雜的網路配置，因為佇列管理程式會在失效接手時變更 IP 位址

HA 叢集包括下列特性：

- 協調多個資源 (例如應用程式伺服器或資料庫) 的能力
- 更靈活的配置選項，包括包含兩個以上節點的叢集
- 可以在沒有操作員介入的情況下多次失效接手
- 在失效接手過程中接管佇列管理程式的 IP 位址

HA 叢集的限制包括：

- 需要額外的產品購買和技能
- 需要可在叢集節點之間切換的磁碟
- HA 叢集的配置相對複雜
- 失效接手在歷史上相當緩慢，但最近 HA 叢集產品正在改善此狀況
- 如果用來監視資源 (例如佇列管理程式) 的 Script 中有缺點，則可能會發生不必要的失效接手

HA 叢集與佇列管理程式叢集的關係

佇列管理程式叢集可減少管理，並在佇列管理程式叢集佇列的實例之間提供訊息負載平衡。它們也提供高於單一佇列管理程式的可用性，因為在佇列管理程式失敗之後，傳訊應用程式仍然可以存取佇列管理程式叢集佇列的現存實例。不過，僅佇列管理程式叢集並不會自動偵測佇列管理程式失敗，以及自動觸發佇列管理程式重新啟動或失效接手。HA 叢集提供這些特性。這兩種類型的叢集可以一起使用，以產生良好的效果。

在 UNIX and Linux 上搭配使用 WebSphere MQ 與高可用性叢集

您可以在 UNIX and Linux 平台上搭配使用 WebSphere MQ 與高可用性 (HA) 叢集：例如，PowerHA for AIX (舊稱為 HACMP)、Veritas Cluster Server、HP Serviceguard 或具有 Red Hat Cluster Suite 的 Red Hat Enterprise Linux 叢集。

在 WebSphere MQ 7.0.1 之前，已提供 SupportPac MC91 來協助配置 HA 叢集。WebSphere MQ 7.0.1 版對佇列管理程式儲存其資料的控制程度高於舊版。這可讓您更容易在 HA 叢集中配置佇列管理程式。SupportPac MC91 隨附的大部分 Script 已不再需要，且 SupportPac 已撤銷。

本節介紹第 267 頁的『HA 叢集配置』、HA 叢集與佇列管理程式叢集的關係、第 267 頁的『WebSphere MQ 用戶端』及第 267 頁的『在 HA 叢集中運作的 WebSphere MQ』，並引導您完成步驟並提供範例 Script，您可以調整這些 Script 來配置具有 HA 叢集的佇列管理程式。

請參閱您環境的特定 HA 叢集文件，以取得本節中所說明之配置步驟的協助。

HA 叢集配置

在本節中，術語節點用來指執行作業系統及 HA 軟體的實體；在此用法中，"computer"、"system" 或 "machine" 或 "partition" 或 "blade" 可能被視為同義字。您可以使用 WebSphere MQ 來協助設定待命或接管配置，包括所有叢集節點在其中執行 WebSphere MQ 工作量的相互接管。

待命配置是最基本的 HA 叢集配置，其中一個節點執行工作，而另一個節點僅充當待命節點。待命節點不執行工作，稱為閒置；此配置有時稱為冷待命。這類配置需要高度的硬體備援。為了節省硬體，可以延伸此配置，讓多個工作者節點具有單一待命節點。其要點是待命節點可以接管任何其他工作者節點的工作。此配置仍稱為待命配置，有時稱為 "N+1" 配置。

接管配置是更進階的配置，其中所有節點都會執行部分工作，而且在節點失敗時可以接管重要工作。

單向接管配置是待命節點執行一些額外、非重要及不可移動工作的配置。此配置類似於待命配置，但待命節點正在執行 (非重要) 工作。

交互接管配置是指所有節點都在其中執行高可用性 (可移動) 工作的配置。這種類型的 HA 叢集配置有時也稱為「主動/主動」，以指出所有節點都主動處理重要工作量。

使用延伸待命配置或任一接管配置時，請務必考量可能放置在可接管其他節點工作之節點上的尖峰負載。這類節點必須具備足夠容量來維持可接受的效能層次。

HA 叢集與佇列管理程式叢集的關係

佇列管理程式叢集可減少管理，並在佇列管理程式叢集佇列的實例之間提供訊息負載平衡。它們也提供高於單一佇列管理程式的可用性，因為在佇列管理程式失敗之後，傳訊應用程式仍然可以存取佇列管理程式叢集佇列的現存實例。不過，僅佇列管理程式叢集並不會自動偵測佇列管理程式失敗，以及自動觸發佇列管理程式重新啟動或失效接手。HA 叢集提供這些特性。這兩種類型的叢集可以一起使用，以產生良好的效果。

WebSphere MQ 用戶端

與佇列管理程式通訊且可能需要重新啟動或接管的 WebSphere MQ 用戶端必須寫入，才能容忍連線中斷，且必須反覆地嘗試重新連接。WebSphere MQ 第 7 版在處理「用戶端通道定義表 (CCDT)」時引進了有助於連線可用性及工作量平衡的特性；不過，當使用失效接手系統時，這些特性並不直接相關。

容許 WebSphere MQ MQI 用戶端參與兩階段交易的「延伸交易式用戶端 (ETC)」必須一律連接至相同的佇列管理程式。ETC 無法使用 IP 負載平衡器之類的技術，從佇列管理程式清單中選取。當您使用 HA 產品時，佇列管理程式會維護其身分 (名稱及位址)，不論它在哪一個節點上執行，因此 ETC 可以與受 HA 控制的佇列管理程式搭配使用。

在 HA 叢集中運作的 WebSphere MQ

所有 HA 叢集都具有失效接手單元的概念。這是一組定義，其中包含組成高可用性服務的所有資源。失效接手單元包括服務本身及其相依的所有其他資源。

HA 解決方案針對失效接手單元使用不同的術語：

- 在 PowerHA for AIX 上，失效接手單元稱為資源群組。
- 在 Veritas Cluster Server 上，它稱為服務群組。
- 在 Serviceguard 上，它稱為套件。

本主題使用術語資源群組來表示失效接手單元。

WebSphere MQ 的最小失效接手單元是佇列管理程式。一般而言，包含佇列管理程式的資源群組也包含磁區群組或磁碟群組中的共用磁碟，該磁區群組或磁碟群組專門保留供資源群組使用，以及用來連接佇列管理程式的 IP 位址。也可以併入其他 WebSphere MQ 資源，例如相同資源群組中的接聽器或觸發監視器，作為個別資源，或在佇列管理程式本身的控制下。

要在 HA 叢集中使用的佇列管理程式必須在叢集中節點之間共用的磁碟上具有其資料及日誌。HA 叢集可確保叢集中一次只有一個節點可以寫入磁碟。HA 叢集可以使用監視 Script 來監視佇列管理程式的狀態。

對於與佇列管理程式相關的資料及日誌，可以使用單一共用磁碟。不過，正常作法是使用個別共用檔案系統，以便它們可以獨立調整大小。

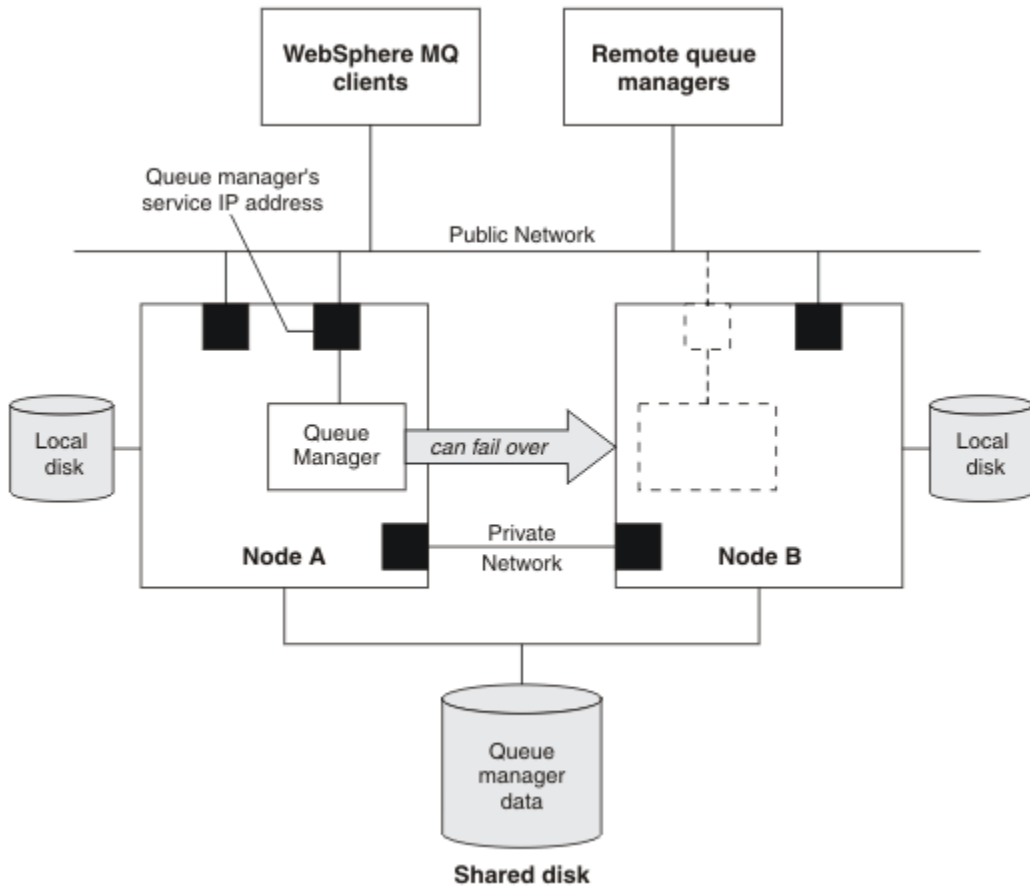


圖 60: HA 叢集

圖 1 說明具有兩個節點的 HA 叢集。高可用性叢集正在管理已在資源群組中定義之佇列管理程式的可用性。這是主動/被動或冷待命配置，因為目前只有一個節點（節點 A）正在執行佇列管理程式。已在共用磁碟上使用佇列管理程式的資料及日誌檔來建立佇列管理程式。佇列管理程式具有也由 HA 叢集管理的服務 IP 位址。佇列管理程式取決於共用磁碟及其服務 IP 位址。當 HA 叢集使佇列管理程式從節點 A 失效接手至節點 B 時，它會先將佇列管理程式的相依資源移至節點 B，然後啟動佇列管理程式。

如果 HA 叢集包含多個佇列管理程式，則在失效接手之後，HA 叢集配置可能會導致兩個以上佇列管理程式在相同節點上執行。HA 叢集中的每一個佇列管理程式都必須指派其自己的埠號，它會在任何特定時間剛好處於作用中的叢集節點上使用該埠號。

通常，HA 叢集會以 root 使用者身分執行。WebSphere MQ 以 mqm 使用者身分執行。WebSphere MQ 的管理已授與 mqm 群組的成員。請確保 mqm 使用者和群組都存在於所有 HA 叢集節點上。叢集中的使用者 ID 和群組 ID 必須一致。不容許 root 使用者管理 WebSphere MQ；啟動、停止或監視 Script 的 Script 必須切換至 mqm 使用者。

註: WebSphere MQ 必須正確安裝在所有節點上; 您無法共用產品執行檔。

配置共用磁碟

HA 叢集中的 WebSphere MQ 佇列管理程式需要共用磁碟上的資料檔及日誌檔位於共同指名的遠端檔案系統中。

若要配置共用磁碟，請完成下列步驟:

1. 決定佇列管理程式檔案系統的裝載點名稱。例如， /MQHA/qmgrname/data 代表佇列管理程式的資料檔， /MQHA/qmgrname/log 代表其日誌檔。
2. 建立磁區群組 (或磁碟群組)，以包含佇列管理程式的資料及日誌檔。此磁區群組由與佇列管理程式相同的資源群組中的高可用性 (HA) 叢集所管理。
3. 建立磁區群組中佇列管理程式資料及日誌檔的檔案系統。
4. 依序針對每一個節點，建立檔案系統的裝載點，並確定可以裝載檔案系統。mqm 使用者必須擁有裝載點。

圖 1 顯示 HA 叢集中佇列管理程式的可能佈置。佇列管理程式的資料及日誌目錄都位於裝載於 /MQHA/QM1 的共用磁碟上。當發生失效接手時，此磁碟會在 HA 叢集的節點之間切換，以便在每次重新啟動佇列管理程式時都可以使用資料。mqs.ini 檔案具有 QM1 佇列管理程式的段落。qm.ini 檔案中的 Log 段落具有 LogPath 的值。

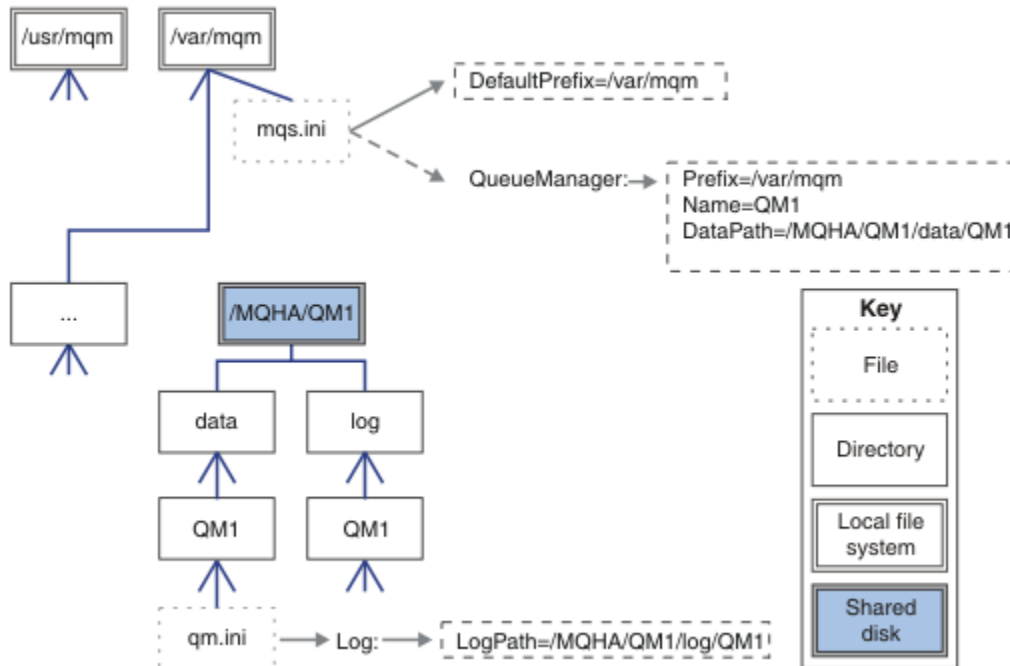


圖 61: 共用具名 data 及 log 目錄

建立佇列管理程式以在高可用性 (HA) 叢集中使用

在高可用性叢集中使用佇列管理程式的第一步是在其中一個節點上建立佇列管理程式。

若要建立要在 HA 叢集中使用的佇列管理程式，請選取叢集中要在其上建立佇列管理程式的其中一個節點。在此節點上，完成下列步驟：

1. 在節點上裝載佇列管理程式的檔案系統。
2. 使用 **crtmqm** 指令來建立佇列管理程式。例如：


```
crtmqm -md /MQHA/qmgrname/data -ld /MQHA/qmgrname/log qmgrname
```
3. 使用 **stmqm** 指令手動啟動佇列管理程式。
4. 完成佇列管理程式的任何起始配置，例如建立佇列及通道，以及將佇列管理程式設定為在佇列管理程式啟動時自動啟動接聽器。
5. 使用 **endmqm** 指令停止佇列管理程式。
6. 使用 **dspmqlnf** 指令來顯示您可以在稍後作業中使用的 **addmqinf** 指令，如第 270 頁的『將佇列管理程式配置資訊新增至高可用性 (HA) 叢集中的其他節點』中所記載：

```
dspmqlnf -o command qmgrname
```

其中 qmgrname 是佇列管理程式的名稱。

7. 顯示的 **addmqinf** 指令將類似於下列範例:

```
addmqinf -sQueueManager -vName=qmgrname -vDirectory=qmgrname \  
-vPrefix=/var/mqm -vDataPath=/MQHA/qmgrname/data/qmgrname
```

請仔細記下所顯示的指令。

8. 卸載佇列管理程式的檔案系統。

現在, 您已準備好完成 第 270 頁的『將佇列管理程式配置資訊新增至高可用性 (HA) 叢集中的其他節點』中說明的步驟。

將佇列管理程式配置資訊新增至高可用性 (HA) 叢集中的其他節點

您必須將佇列管理程式配置新增至 HA 叢集中的其他節點。

在完成此作業之前, 您必須已完成 第 269 頁的『建立佇列管理程式以在高可用性 (HA) 叢集中使用』中的步驟。

若要將佇列管理程式的配置資訊新增至 HA 叢集中的每一個其他節點, 請在每一個其他節點上完成下列步驟:

1. 裝載佇列管理程式檔案系統。
2. 透過直接編輯 `/var/mqm/mqs.ini`, 或透過發出 第 269 頁的『建立佇列管理程式以在高可用性 (HA) 叢集中使用』中步驟 6 和 7 中 **dspmqlinf** 指令所顯示的 **addmqinf** 指令, 將佇列管理程式配置資訊新增至節點。
3. 啟動和停止佇列管理程式以驗證配置。
用來啟動及停止佇列管理程式的指令, 必須從與 **addmqinf** 指令相同的 IBM WebSphere MQ 安裝中發出。若要從不同的安裝啟動及停止佇列管理程式, 您必須先使用 **setmqm** 指令設定與佇列管理程式相關聯的安裝。如需相關資訊, 請參閱 [setmqm](#)。
4. 卸載佇列管理程式檔案系統。

在高可用性 (HA) 叢集控制下啟動佇列管理程式

在 HA 叢集中, 佇列管理程式會呈現為資源。HA 叢集必須能夠啟動和停止佇列管理程式。在大部分情況下, 您可以使用 Shell Script 來啟動佇列管理程式。您必須使用網路檔案系統或將這些 Script 複製到每一個本端磁碟, 在叢集中所有節點上的相同位置提供這些 Script。

註: 在重新啟動失敗的佇列管理程式之前, 您必須先切斷應用程式與該佇列管理程式實例的連線。如果沒有, 佇列管理程式可能無法正確重新啟動。

這裡提供適當 Shell Script 的範例。您可以根據需要自訂這些, 並在 HA 叢集的控制下使用它們來啟動佇列管理程式。

下列 Shell Script 是如何從 HA 叢集使用者切換至 mqm 使用者以順利啟動佇列管理程式的範例:

```
#!/bin/ksh  
# A simple wrapper script to switch to the mqm user.  
su mqm -c name_of_your_script $*
```

下列 Shell Script 是如何啟動佇列管理程式, 而不需對佇列管理程式的現行狀態進行任何假設的範例。請注意, 它使用極其突然的方法來結束屬於佇列管理程式的任何程序:

```
#!/bin/ksh  
#  
# This script robustly starts the queue manager.  
#  
# The script must be run by the mqm user.  
  
# The only argument is the queue manager name. Save it as QM variable  
QM=$1  
  
if [ -z "$QM" ]  
then  
    echo "ERROR! No queue manager name supplied"  
    exit 1  
fi
```

```

# End any queue manager processes which might be running.

srchstr="( |-m)$QM *.*$"
for process in amqzmuc0 amqzma0 amqfcxba amqfcpub amqpcsea amqzlaa0 \
               amqzlsa0 runmqchi runmqlsr amqcrsta amqirmfa amqrmppa \
               amqzfuma amqzdmaa amqzmuf0 amqzmur0 amqzmgr0
do
  ps -ef | tr "\t" " " | grep $process | grep -v grep | \
  egrep "$srchstr" | awk '{print $2}' | \
  xargs kill -9 > /dev/null 2>&1
done

# It is now safe to start the queue manager.
# The stmqm command does not use the -x flag.
stmqm ${QM}

```

您可以修改 Script 來啟動其他相關程式。

在高可用性 (HA) 叢集的控制下停止佇列管理程式

在大部分情況下，您可以使用 Shell Script 來停止佇列管理程式。這裡提供適當 Shell Script 的範例。您可以根據需要自訂這些，並使用它們來停止高可用性叢集控制下的佇列管理程式。

下列 Script 是如何在不假設佇列管理程式現行狀態的情況下立即停止的範例。Script 必須由 mqm 使用者執行；因此可能需要在 Shell Script 中覆蓋此 Script，以將使用者從 HA 叢集使用者切換至 mqm (第 270 頁的『在高可用性 (HA) 叢集控制下啟動佇列管理程式』中提供範例 Shell Script)：

```

#!/bin/ksh
#
# The script ends the QM by using two phases, initially trying an immediate
# end with a time-out and escalating to a forced stop of remaining
# processes.
#
# The script must be run by the mqm user.
#
# There are two arguments: the queue manager name and a timeout value.
QM=$1
TIMEOUT=$2

if [ -z "$QM" ]
then
  echo "ERROR! No queue manager name supplied"
  exit 1
fi

if [ -z "$TIMEOUT" ]
then
  echo "ERROR! No timeout specified"
  exit 1
fi

for severity in immediate brutal
do
  # End the queue manager in the background to avoid
  # it blocking indefinitely. Run the TIMEOUT timer
  # at the same time to interrupt the attempt, and try a
  # more forceful version. If the brutal version fails,
  # nothing more can be done here.

  echo "Attempting ${severity} end of queue manager '${QM}'"
  case $severity in
    immediate)
      # Minimum severity of endmqm is immediate which severs connections.
      # HA cluster should not be delayed by clients
      endmqm -i ${QM} &
      ;;
    brutal)
      # This is a forced means of stopping queue manager processes.

      srchstr="( |-m)$QM *.*$"
      for process in amqzmuc0 amqzma0 amqfcxba amqfcpub amqpcsea amqzlaa0 \
                   amqzlsa0 runmqchi runmqlsr amqcrsta amqirmfa amqrmppa \
                   amqzfuma amqzdmaa amqzmuf0 amqzmur0 amqzmgr0
      do
        ps -ef | tr "\t" " " | grep $process | grep -v grep | \
        egrep "$srchstr" | awk '{print $2}' | \

```

```

        xargs kill -9 > /dev/null 2>&1
    done
esac

TIMED_OUT=yes
SECONDS=0
while (( $SECONDS < ${TIMEOUT} ))
do
    TIMED_OUT=yes
    i=0
    while [ $i -lt 5 ]
    do
        # Check for execution controller termination
        srchstr="( |-m)$QM *.*$"
        cnt=`ps -ef | tr "\t" " " | grep amqzma0 | grep -v grep | \
            egrep "$srchstr" | awk '{print $2}' | wc -l`
        i=`expr $i + 1`
        sleep 1
        if [ $cnt -eq 0 ]
        then
            TIMED_OUT=no
            break
        fi
    done

    if [ ${TIMED_OUT} = "no" ]
    then
        break
    fi

    echo "Waiting for ${severity} end of queue manager '${QM}'"
    sleep 1
done # timeout loop

if [ ${TIMED_OUT} = "yes" ]
then
    continue      # to next level of urgency
else
    break         # queue manager is ended, job is done
fi

done # next phase

```

監視佇列管理程式

通常會提供一種方法，讓高可用性 (HA) 叢集定期監視佇列管理程式的狀態。在大部分情況下，您可以使用 Shell Script 來執行此動作。這裡提供適當 Shell Script 的範例。您可以根據需要自訂這些 Script，並使用它們來進行您環境特有的其他監視檢查。

從 WebSphere MQ 7.1 版開始，系統上可以同時存在多個 WebSphere MQ 安裝。如需多個安裝的相關資訊，請參閱 [多個安裝](#)。如果您想要在多個安裝 (包括 7.1 版或更高版本的安裝) 之間使用監視 Script，您可能需要執行一些額外步驟。如果您具有主要安裝，或使用的 Script 版本早於 7.1 版，則不需要指定 `MQ_INSTALLATION_PATH` 即可使用該 Script。否則，下列步驟可確保正確識別 `MQ_INSTALLATION_PATH`：

1. 從 7.1 版安裝架構使用 `crtmqenv` 指令，以識別佇列管理程式的正確 `MQ_INSTALLATION_PATH`：

```
crtmqenv -m qmname
```

此指令會針對 `qmname` 指定的佇列管理程式傳回正確的 `MQ_INSTALLATION_PATH` 值。

2. 使用適當的 `qmname` 及 `MQ_INSTALLATION_PATH` 參數執行監視 Script。

註：PowerHA for AIX 不提供將參數提供給佇列管理程式的監視程式。您必須為每一個佇列管理程式建立個別監視程式，以封裝佇列管理程式名稱。以下是在 AIX 上用來封裝佇列管理程式名稱的 Script 範例：

```
#!/bin/ksh
su mqm -c name_of_monitoring_script qmname MQ_INSTALLATION_PATH
```

其中 `MQ_INSTALLATION_PATH` 是選用參數，指定與佇列管理程式 `qmname` 相關聯的 IBM WebSphere MQ 安裝路徑。

下列 Script 不健全，可能導致 `runmqsc` 當掉。一般而言，HA 叢集會將當掉的監視 Script 視為失敗，且本身就健全，無法因應這種可能性。

不過，Script 會容忍佇列管理程式處於啟動中狀態。這是因為 HA 叢集通常會在啟動佇列管理程式之後立即開始監視它。部分 HA 叢集會區分資源的起始階段與執行中階段，但必須配置起始階段的持續時間。因為啟動佇列管理程式所花費的時間取決於它必須執行的工作量，所以很難選擇啟動佇列管理程式所花費的時間上限。如果您選擇太低的值，HA 叢集會錯誤地假設佇列管理程式在未完成啟動時失敗。這可能會導致無休止的失效接手。

此 Script 必須由 mqm 使用者執行；因此可能需要將此 Script 包裝在 Shell Script 中，以將使用者從 HA 叢集使用者切換至 mqm (第 270 頁的『在高可用性 (HA) 叢集控制下啟動佇列管理程式』中提供範例 Shell Script):

```
#!/bin/ksh
#
# This script tests the operation of the queue manager.
#
# An exit code is generated by the runmqsc command:
# 0 => Either the queue manager is starting or the queue manager is running and responds.
#     Either is OK.
# >0 => The queue manager is not responding and not starting.
#
# This script must be run by the mqm user.
QM=$1
MQ_INSTALLATION_PATH=$2

if [ -z "$QM" ]
then
    echo "ERROR! No queue manager name supplied"
    exit 1
fi

if [ -z "$MQ_INSTALLATION_PATH" ]
then
    # No path specified, assume system primary install or MQ level < 7.1.0.0
    echo "INFO: Using shell default value for MQ_INSTALLATION_PATH"
else
    echo "INFO: Prefixing shell PATH variable with $MQ_INSTALLATION_PATH/bin"
    PATH=$MQ_INSTALLATION_PATH/bin:$PATH
fi

# Test the operation of the queue manager. Result is 0 on success, non-zero on error.
echo "ping qmgr" | runmqsc ${QM} > /dev/null 2>&1
pingresult=$?

if [ $pingresult -eq 0 ]
then # ping succeeded

    echo "Queue manager '${QM}' is responsive"
    result=0

else # ping failed

    # Don't condemn the queue manager immediately, it might be starting.
    srchstr="(|-m)$QM *.*$"
    cnt=`ps -ef | tr "\t" " " | grep strmqm | grep "$srchstr" | grep -v grep \
        | awk '{print $2}' | wc -l`
    if [ $cnt -gt 0 ]
    then
        # It appears that the queue manager is still starting up, tolerate
        echo "Queue manager '${QM}' is starting"
        result=0
    else
        # There is no sign of the queue manager starting
        echo "Queue manager '${QM}' is not responsive"
        result=$pingresult
    fi
fi

exit $result
```

將佇列管理程式置於高可用性 (HA) 叢集的控制之下

您必須在 HA 叢集的控制下，使用佇列管理程式的 IP 位址及共用磁碟來配置佇列管理程式。

若要定義資源群組以包含佇列管理程式及其所有相關聯資源，請完成下列步驟：

1. 建立資源群組，其中包含佇列管理程式、佇列管理程式的磁區或磁碟群組，以及佇列管理程式的 IP 位址。IP 位址是虛擬 IP 位址，而不是電腦的 IP 位址。
2. 請驗證 HA 叢集在叢集節點之間正確切換資源，且已備妥可控制佇列管理程式。

從高可用性 (HA) 叢集節點刪除佇列管理程式

您可能想要從不再需要執行佇列管理程式的節點中移除佇列管理程式。

若要從 HA 叢集中的節點移除佇列管理程式，請完成下列步驟：

1. 從 HA 叢集中移除節點，以便 HA 叢集不再嘗試在此節點上啟動佇列管理程式。
2. 使用下列 **rmvmqinf** 指令來移除佇列管理程式的配置資訊：

```
rmvmqinf qmgrname
```

若要完全刪除佇列管理程式，請使用 **dltmqm** 指令。不過，請注意，這會完全刪除佇列管理程式的資料及日誌檔。當您已刪除佇列管理程式時，您可以使用 **rmvmqinf** 指令，從其他節點移除其餘配置資訊。

支援 Microsoft Cluster Service (MSCS)

引進及設定 MSCS 以支援虛擬伺服器的失效接手。

此資訊僅適用於 **WebSphere MQ for Windows**。

Microsoft Cluster Service (MSCS) 可讓您將伺服器連接至叢集，提供更高的資料及應用程式可用性，並讓您更容易管理系統。MSCS 可以自動偵測及回復伺服器或應用程式的故障情形。

MSCS 支援虛擬伺服器的失效接手，這些虛擬伺服器對應於應用程式、網站、列印佇列或檔案共用 (例如，包括其磁碟轉軸、檔案及 IP 位址)。

失效接手是 MSCS 用來偵測叢集中某部電腦上應用程式的失敗，並依序關閉已中斷的應用程式、將其狀態資料傳送至另一部電腦，然後在該處重新起始應用程式的處理程序。

本節介紹 MSCS 叢集，並在下列各節中說明設定 MSCS 支援：

- [第 274 頁的『簡介 MSCS 叢集』](#)
- [第 275 頁的『為 MSCS 叢集作業設定 IBM WebSphere MQ』](#)

然後，在下列各節中告訴您如何配置 WebSphere MQ 以進行 MSCS 叢集作業：

- [第 277 頁的『建立佇列管理程式以與 MSCS 搭配使用』](#)
- [第 277 頁的『將佇列管理程式移至 MSCS 儲存體』](#)
- [第 278 頁的『將佇列管理程式置於 MSCS 控制下』](#)
- [第 283 頁的『從 MSCS 控制項移除佇列管理程式』](#)

然後在下列各節中提供有關搭配使用 MSCS 與 WebSphere MQ 的一些有用提示，並詳細說明 WebSphere MQ MSCS 支援公用程式：

- [第 284 頁的『使用 MSCS 的提示和要訣』](#)
- [第 287 頁的『IBM WebSphere MQ MSCS 支援公用程式』](#)

簡介 MSCS 叢集

MSCS 叢集是由兩部以上電腦組成的群組，它們連接在一起並配置成如果其中一部電腦失敗，MSCS 會執行失效接手，將應用程式的狀態資料從失敗電腦傳送至叢集中的另一部電腦，並在該處重新起始其作業。

[第 265 頁的『搭配使用 WebSphere MQ 與高可用性配置』](#) 包含 MSCS 叢集、多重實例佇列管理程式及 WebSphere MQ 叢集之間的比較。

在本節及其子層主題中，術語叢集本身使用時，**一律** 表示 MSCS 叢集。這不同於本手冊中其他位置所說明的 WebSphere MQ 叢集。

雙機器叢集包含兩台電腦 (例如，A 和 B)，它們共同連接至網路，以使用虛擬 IP 位址進行用戶端存取。它們也可能透過一個以上的私密網路相互連接。A 和 B 至少共用一個磁碟，供各自的伺服器應用程式使用。另外還有另一個共用磁碟，必須是獨立磁碟 (RAID) 層次 1 的備援陣列，供 MSCS 專用；這稱為仲裁磁碟。MSCS 會監視這兩台電腦，檢查軟硬體是否正確地執行。

在這類簡單的設定中，兩台電腦都已安裝所有應用程式，但只有電腦 A 會執行即時應用程式；電腦 B 只是在執行中及等待中。如果電腦 A 遇到一系列問題中的任何一個，MSCS 會依序關閉已中斷的應用程式，將其狀態資料傳送至另一部電腦，然後在那裡重新起始應用程式。這稱為失效接手。應用程式可以設為可察覺叢集，以便它們與 MSCS 完全互動，並循序進行失效接手。

第 275 頁的圖 62 顯示兩台電腦的叢集的一般設定。

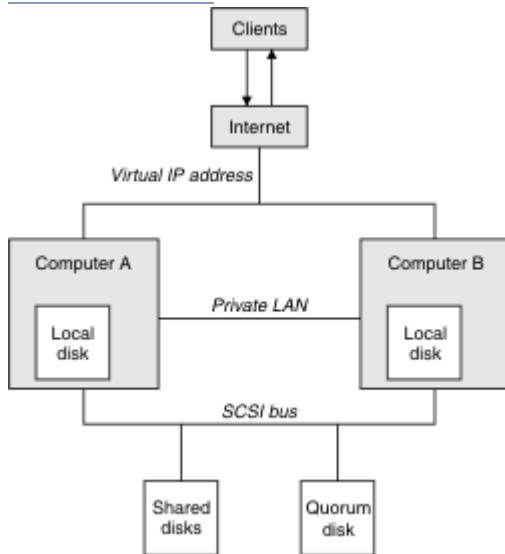


圖 62: 雙電腦 MSCS 叢集

在 MSCS 的控制下，每一台電腦都可以存取共用磁碟，但一次僅限於一台電腦。如果發生失效接手，MSCS 會轉換為存取另一台電腦。共用磁碟本身通常是 RAID，但不一定。

每一台電腦都連接到外部網路以進行用戶端存取，且每一台電腦都各有一個 IP 位址。不過，與此叢集通訊的外部用戶端只知道一個虛擬 IP 位址，而 MSCS 會適當地遞送叢集內的 IP 資料流量。

MSCS 也會在兩台電腦之間執行自己的通訊，例如透過一或多個專用連線或透過公用網路，以使用活動訊號來監視它們的狀態，以及同步化它們的資料庫。

為 MSCS 叢集作業設定 IBM WebSphere MQ

您可以將佇列管理程式設為 MSCS 的失效接手單元，以配置 IBM WebSphere MQ 進行叢集作業。您將佇列管理程式定義為 MSCS 的資源，然後可以監視它，並在發生問題時將它傳送至叢集中的另一部電腦。

若要針對此設定系統，您可以從在叢集中的每一部電腦上安裝 IBM WebSphere MQ 開始。

因為佇列管理程式與 IBM WebSphere MQ 安裝名稱相關聯，所以叢集中所有電腦上的 IBM WebSphere MQ 安裝名稱應該相同。請參閱 [安裝及解除安裝](#)。

佇列管理程式本身只需要存在於您建立它們的電腦上。如果發生失效接手，MSCS 會在另一部電腦上起始佇列管理程式。不過，佇列管理程式必須將其日誌和資料檔放在叢集共用磁碟上，而不是在本端磁碟機上。如果您已在本端磁碟機上安裝佇列管理程式，則可以使用 IBM WebSphere MQ 隨附的工具來移轉它；請參閱第 277 頁的『將佇列管理程式移至 MSCS 儲存體』。如果您要建立新的佇列管理程式以與 MSCS 搭配使用，請參閱第 277 頁的『建立佇列管理程式以與 MSCS 搭配使用』。

安裝及移轉之後，請使用 MSCS 叢集管理者讓 MSCS 知道您的佇列管理程式；請參閱第 278 頁的『將佇列管理程式置於 MSCS 控制下』。

如果您決定從 MSCS 控制項移除佇列管理程式，請使用第 283 頁的『從 MSCS 控制項移除佇列管理程式』中說明的程序。

設定對稱

當應用程式從一個節點切換至另一個節點時，不論節點為何，它都必須以相同方式運作。確保這一點的最佳方式是讓環境完全相同。

如果可以，請在每一部電腦上設定具有相同硬體、作業系統軟體、產品軟體及配置的叢集。具體而言，請確保兩台電腦上安裝的所有必要軟體在版本、維護層次、SupportPacs、路徑及結束程式等方面都相同，並確保保存在一般名稱空間(安全環境)，如第 276 頁的『MSCS 安全』中所述。

MSCS 安全

若要順利完成 MSCS 安全，請遵循下列準則。

指引如下：

- 請確定您在叢集中的每一部電腦上都有相同的軟體安裝。
- 跨叢集建立一般名稱空間(安全環境)。
- 建立網域中 MSCS 叢集成員的節點，其中作為叢集擁有者的使用者帳戶是網域帳戶。
- 將叢集上的其他使用者帳戶也設為網域帳戶，以便在兩個節點上都可以使用它們。如果您已有網域，且與 WebSphere MQ 相關的帳戶是網域帳戶，則會自動這樣做。如果您目前沒有網域，請考慮設定迷你網域，以滿足叢集節點和相關帳戶的需求。您的目標是讓包含兩部電腦的叢集看起來像是單一運算資源。

請記住，一部電腦的本端帳戶不存在於另一部電腦上。即使您在另一部電腦上建立具有相同名稱的帳戶，其安全 ID (SID) 也會不同，因此當您將應用程式移至另一個節點時，該節點上不存在許可權。

在失效接手或移動期間，WebSphere MQ MSCS 支援可確保包含佇列管理程式物件的所有檔案在目的地節點上都具有同等許可權。程式碼會明確檢查 Administrators 和 mqm 群組以及 SYSTEM 帳戶是否具有完全控制，並且如果 Everyone 具有舊節點的讀取權，則會在目的地節點上新增該許可權。

您可以使用網域帳戶來執行 WebSphere MQ 服務。請確定它存在於叢集中每一部電腦上的本端 mqm 群組中。

搭配使用多個佇列管理程式與 MSCS

如果您在電腦上執行多個佇列管理程式，則可以選擇其中一個設定。

設定如下：

- 單一群組中的所有佇列管理程式。在此配置中，如果任何佇列管理程式發生問題，則群組中的所有佇列管理程式會以群組方式失效接手至其他電腦。
- 每一個群組中的單一佇列管理程式。在此配置中，如果佇列管理程式發生問題，則它會單獨失效接手至其他電腦，而不會影響其他佇列管理程式。
- 前兩個組合的混合。

叢集模式

有兩種模式可讓您使用 WebSphere MQ 來執行叢集系統：「主動/被動」或「主動/主動」。

註：如果您將 MSCS 與 Microsoft Transaction Server (COM+) 一起使用，則無法使用「主動/主動」模式。

主動/被動模式

在「主動/被動」模式中，電腦 A 上具有執行中的應用程式，且電腦 B 是備份，只有在 MSCS 偵測到問題時才會使用。

您只能將此模式與一個共用磁碟搭配使用，但如果任何應用程式導致失效接手，則**所有**應用程式必須以群組方式傳送(因為一次只有一部電腦可以存取共用磁碟)。

您可以將具有 A 的 MSCS 配置為偏好的電腦。然後，當電腦 A 已修復或更換且再次正常運作時，MSCS 會偵測到此情況，並自動將應用程式切換回電腦 A。

如果您執行多個佇列管理程式，請考量每一個各有一個共用磁碟。然後將每一個佇列管理程式放入 MSCS 中的個別群組。以此方式，任何佇列管理程式都可以失效接手至其他電腦，而不會影響其他佇列管理程式。

主動/主動模式

在主動/主動模式中，電腦 A 和 B 都有執行中的應用程式，且每一部電腦上的群組都設為使用另一部電腦作為備份。如果在電腦 A 上偵測到失敗，MSCS 會將狀態資料傳送至電腦 B，並在該處重新起始應用程式。然後電腦 B 執行其自己的應用程式及 A 的應用程式。

對於此設定，您至少需要兩個共用磁碟。您可以將 MSCS 配置為 A 應用程式的偏好電腦，並將 B 配置為 B 應用程式的偏好電腦。在失效接手及修復之後，每一個應用程式都會自動回到自己的電腦上。

例如，對於 WebSphere MQ，這表示您可以執行兩個佇列管理程式，A 和 B 各一個佇列管理程式，各利用自己電腦的完整功能。在電腦 A 上失敗之後，兩個佇列管理程式都會在電腦 B 上執行。這將意味著共用一部電腦的功能，並減少以速度處理大量資料的能力。不過，當您在 A 上尋找並修復錯誤時，您的重要應用程式仍可使用。

建立佇列管理程式以與 MSCS 搭配使用

此程序確保以適合在 MSCS 控制下準備及放置的方式建立新的佇列管理程式。

首先，請在本端磁碟機上建立佇列管理程式及其所有資源，然後將日誌檔及資料檔移轉至共用磁碟。(您可以反轉此作業。)請 **不要** 嘗試使用共用磁碟機上的資源來建立佇列管理程式。

您可以從命令提示字元或在「WebSphere MQ 探險家」中，以兩種方式建立佇列管理程式來與 MSCS 搭配使用。使用命令提示字元的優點是建立佇列管理程式已停止，並設為手動啟動，這已備妥可供 MSCS 使用。(「IBM WebSphere MQ 探險家」會自動啟動新的佇列管理程式，並將它設為建立之後自動啟動。您必須變更此項。)

從命令提示字元建立佇列管理程式

請遵循下列步驟，從命令提示字元建立佇列管理程式，以與 MSCS 搭配使用：

1. 請確定您已設定環境變數 MQSPREFIX 來參照本端磁碟機，例如 C:\WebSphere MQ。如果您變更此項，請將機器重新開機，讓「系統」帳戶取得變更。如果您未設定變數，則會在佇列管理程式的 WebSphere MQ 預設目錄中建立佇列管理程式。
2. 使用 `crtmqm` 指令建立佇列管理程式。例如，若要在預設目錄中建立稱為 `mscs_test` 的佇列管理程式，請使用：

```
crtmqm mscs_test
```

3. 繼續進行 [第 277 頁的『將佇列管理程式移至 MSCS 儲存體』](#)。

使用「WebSphere MQ 探險家」建立佇列管理程式

請遵循下列步驟，使用「IBM WebSphere MQ 探險家」來建立佇列管理程式，以與 MSCS 搭配使用：

1. 從「開始」功能表啟動 IBM WebSphere MQ Explorer。
2. 在「Navigator」視圖中，展開樹狀結構節點以尋找 Queue Managers 樹狀結構節點。
3. 用滑鼠右鍵按一下 Queue Managers 樹狀結構節點，然後選取 New->Queue Manager。即會顯示「建立佇列管理程式」畫面。
4. 完成對話框 (步驟 1)，然後按一下 Next>。
5. 完成對話框 (步驟 2)，然後按一下 Next>。
6. 完成對話框 (步驟 3)，確定未選取 Start Queue Manager 和 Create Server Connection Channel，然後按一下 Next>。
7. 完成對話框 (步驟 4)，然後按一下 Finish。
8. 繼續進行 [第 277 頁的『將佇列管理程式移至 MSCS 儲存體』](#)。

將佇列管理程式移至 MSCS 儲存體

此程序會配置現存的佇列管理程式，使其適合置於 MSCS 控制下。

若要達到此目的，您可以將日誌檔及資料檔移至共用磁碟，以在失敗時讓其他電腦可以使用它們。例如，現有的佇列管理程式可能有 C:\WebSphere MQ\log\不要 嘗試手動移動檔案；請使用提供的公用程式作為 WebSphere MQ MSCS 支援的一部分，如本主題所述。

如果要移動的佇列管理程式使用 SSL 連線，且 SSL 金鑰儲存庫位於本端機器上的佇列管理程式資料目錄中，則金鑰儲存庫將與其餘佇列管理程式一起移至共用磁碟。依預設，指定 SSL 金鑰儲存庫位置 SSLKEYR 的佇

列管理程式屬性會設為 `MQ_INSTALLATION_PATH\qmgrs\QMGRNAME\ssl\key`，其位於佇列管理程式資料目錄下。`MQ_INSTALLATION_PATH` 代表 WebSphere MQ 安裝所在的高階目錄。`hamvmqm` 指令不會修改這個佇列管理程式屬性。在此狀況下，您必須使用 IBM WebSphere MQ 探險家或 MQSC 指令 `ALTER QMGR` 來修改佇列管理程式屬性 `SSLKEYR`，以指向新的 SSL 金鑰儲存庫檔案。

程序如下：

1. 關閉佇列管理程式，並檢查是否沒有錯誤。
2. 如果佇列管理程式的日誌檔或佇列檔已儲存在共用磁碟上，請跳過此程序的其餘部分，並直接前往 [第 278 頁的『將佇列管理程式置於 MSCS 控制下』](#)。
3. 對佇列檔及日誌檔進行完整媒體備份，並將備份儲存在安全的地方 (請參閱 [第 285 頁的『佇列管理程式日誌檔』](#)，以瞭解為何這很重要)。
4. 如果您已有適當的共用磁碟資源，請繼續步驟 6。否則，請使用 MSCS 叢集管理者來建立 共用磁碟類型的資源，其容量足以儲存佇列管理程式日誌檔及資料 (佇列) 檔。
5. 使用「MSCS 叢集管理者」將共用磁碟從一個叢集節點移至另一個叢集節點，然後重新移回，以測試共用磁碟。
6. 請確定在本端儲存佇列管理程式日誌及資料檔的叢集節點上，共用磁碟在線上。
7. 執行公用程式來移動佇列管理程式，如下所示：

```
hamvmqm /m qmname /dd "e:\
WebSphere MQ" /ld "e:\
WebSphere MQ\log"
```

將 `qmname` 替換為您的佇列管理程式名稱，將 `e` 替換為您的共用磁碟機代號，並將 `WebSphere MQ` 替換為您選擇的目錄。如果目錄尚未存在，則會建立它們。

8. 使用「IBM WebSphere MQ 探險家」來測試佇列管理程式，以確定它可以運作。例如：
 - a. 用滑鼠右鍵按一下佇列管理程式樹狀結構節點，然後選取 `Start`。即會啟動佇列管理程式。
 - b. 用滑鼠右鍵按一下 `Queues` 樹狀結構節點，然後選取 `New->Local Queue...`，並為佇列提供名稱。
 - c. 按一下 `Finish`。
 - d. 用滑鼠右鍵按一下佇列，然後選取 `Put Test Message...`。即會顯示「放置測試訊息」畫面。
 - e. 鍵入部分訊息文字，然後按一下 `Put Test Message`，並關閉畫面。
 - f. 用滑鼠右鍵按一下佇列，然後選取 `Browse Messages...`。即會顯示「訊息瀏覽器」畫面。
 - g. 確定您的訊息在佇列上，然後按一下 `Close`。即會關閉「訊息瀏覽器」畫面。
 - h. 用滑鼠右鍵按一下佇列，然後選取 `Clear Messages...`。清除佇列上的訊息。
 - i. 用滑鼠右鍵按一下佇列，然後選取 `Delete...`。即會顯示確認畫面，按一下 `OK`。已刪除佇列。
 - j. 用滑鼠右鍵按一下佇列管理程式樹狀結構節點，然後選取 `Stop...`。即會顯示「結束佇列管理程式」畫面。
 - k. 按一下 `OK`。佇列管理程式停止。
9. 以 WebSphere MQ 管理者身分，確定佇列管理程式的啟動屬性設為手動。在「IBM WebSphere MQ 探險家」的佇列管理程式內容畫面中，將啟動欄位設為 `manual`。
10. 繼續進行 [第 278 頁的『將佇列管理程式置於 MSCS 控制下』](#)。

將佇列管理程式置於 MSCS 控制下

將佇列管理程式置於 MSCS 控制下所涉及的作業，包括必要作業。

在 MSCS 控制下放置佇列管理程式之前

在將佇列管理程式置於 MSCS 控制下之前，請執行下列作業：

1. 確保叢集中的兩部機器上都已安裝 IBM WebSphere MQ 及其 MSCS 支援，且每一部電腦上的軟體都相同，如 [第 275 頁的『為 MSCS 叢集作業設定 IBM WebSphere MQ』](#) 中所述。

2. 使用 **haregtyp** 公用程式，將 WebSphere MQ 登錄為所有叢集節點上的 MSCS 資源類型。如需相關資訊，請參閱第 287 頁的『IBM WebSphere MQ MSCS 支援公用程式』。
3. 如果您尚未建立佇列管理程式，請參閱第 277 頁的『建立佇列管理程式以與 MSCS 搭配使用』。
4. 如果您已建立佇列管理程式，或它已存在，請確定您已在「第 277 頁的『將佇列管理程式移至 MSCS 儲存體』」中執行此程序。
5. 使用命令提示字元或「IBM WebSphere MQ 探險家」來停止佇列管理程式 (如果它正在執行中)。
6. 在繼續本主題中的下列任一 Windows 程序之前，請先測試共用磁碟機的 MSCS 作業。

Windows Server 2012



小心: MSCS 支援在 WebSphere MQ 7.5 中使用 32 位元 DLL。由於 Windows 2012 中的限制，IBM WebSphere MQ 佇列管理程式在重新啟動之後不會進行失效接手。

Microsoft 已淘汰在 Windows 2012 中使用 32 位元 DLL，因此這個問題目前沒有可用的作業系統修正程式。IBM 不提供 IBM WebSphere MQ 7.5 的 64 位元程式庫。

從 IBM MQ 8.0 開始，有 64 位元程式庫可供使用，因此您必須使用此版本的產品，以取得 Windows 2012 以及更新版本的完整 MSCS 功能。

若要在 Windows Server 2012 上將佇列管理程式置於 MSCS 控制下，請使用下列程序：

1. 登入管理佇列管理程式的叢集節點電腦，或以具有叢集管理許可權的使用者身分登入遠端工作站，並連接管理佇列管理程式的叢集節點。
2. 啟動「失效接手叢集管理」工具。
3. 用滑鼠右鍵按一下 **失效接手叢集管理 > 連接叢集 ...** 以開啟與叢集的連線。
4. 與舊版 Windows 上 MSCS 叢集管理者中使用的群組方法相反，「失效接手叢集管理」工具會使用服務及應用程式的概念。已配置的服務或應用程式包含要叢集一個應用程式所需的所有資源。您可以在 MSCS 下配置佇列管理程式，如下所示：
 - a. 用滑鼠右鍵按一下叢集，然後選取 **配置角色** 以啟動配置精靈。
 - b. 在「選取服務或應用程式」畫面上選取 **其他伺服器**。
 - c. 選取適當的 IP 位址作為用戶端存取點。

此位址應該是未用的 IP 位址，供用戶端及其他佇列管理程式用來連接至 虛擬 佇列管理程式。此 IP 位址不是任一節點的正常 (靜態) 位址; 它是它們之間 浮動 的額外位址。雖然 MSCS 會處理此位址的遞送，但它 **不會** 驗證是否可以呼叫到地址。
 - d. 指派儲存裝置供佇列管理程式專用。此裝置需要先建立為資源實例，然後才能指派。

您可以使用一個磁碟機來同時儲存日誌和佇列檔案，也可以將它們分散在各磁碟機中。在任一情況下，如果每一個佇列管理程式都有自己的共用磁碟，請確保此佇列管理程式所使用的所有磁碟機都是此佇列管理程式專用的，亦即，沒有其他任何磁碟機依賴這些磁碟機。此外，請確保為佇列管理程式使用的每個磁碟機建立資源實例。

磁帶機的資源類型視您使用的 SCSI 支援而定; 請參閱 SCSI 配接卡指示。每一個共用磁碟機可能已有群組和資源。如果是這樣，您不需要為每一個磁碟機建立資源實例。將它從現行群組移至針對佇列管理程式所建立的群組。

針對每一個磁碟機資源，將可能的擁有者設為兩個節點。將相依資源設為無。
 - e. 在「選取資源類型」畫面上選取 **IBM MQSeries MSCS** 資源。
 - f. 完成精靈中的其餘步驟。
5. 在使資源上線之前，IBM MQSeries MSCS 資源需要其他配置：
 - a. 選取新定義的服務，其中包含稱為「新建 IBM MQSeries MSCS」的資源。
 - b. 用滑鼠右鍵按一下 MQ 資源上的 **內容**。
 - c. 配置資源：
 - Name; 選擇可讓您輕鬆識別其適用的佇列管理程式的名稱。
 - Run in a separate Resource Monitor; 用於更好的隔離

- Possible owners; 設定兩個節點
- Dependencies; 新增此佇列管理程式的磁碟機及 IP 位址。

警告: 無法新增這些相依關係表示在失效接手期間, IBM WebSphere MQ 會嘗試將佇列管理程式狀態寫入錯誤的叢集磁碟。因為許多處理程序可能正在嘗試同步寫入此磁碟, 所以部分 IBM WebSphere MQ 處理程序可能會被封鎖而無法執行。

- Parameters; 如下所示:

- QueueManagerName (必要); 此資源要控制的佇列管理程式名稱。此佇列管理程式必須存在於本端電腦上。
- PostOnlineCommand (選用); 您可以指定每當佇列管理程式資源將其狀態從離線變更為線上時要執行的程式。如需詳細資料, 請參閱第 286 頁的『PostOnline 指令和 PreOffline 指令』。
- PreOfflineCommand (選用); 您可以指定每當佇列管理程式資源將其狀態從連線變更為離線時要執行的程式。如需詳細資料, 請參閱第 286 頁的『PostOnline 指令和 PreOffline 指令』。

註: looksAlive 輪詢間隔設為預設值 5000 毫秒。isAlive 輪詢間隔設為預設值 60000 毫秒。只有在完成資源定義之後, 才能修改這些預設值。如需進一步詳細資料, 請參閱第 283 頁的『looksAlive 和 isAlive 輪詢的摘要』。

- 選擇性地設定偏好節點 (但請注意第 286 頁的『使用偏好的節點』中的註解)
 - 依預設, 失效接手原則會設為可感應值, 但您可以調整臨界值及期間, 以控制資源失效接手及群組失效接手, 以符合佇列管理程式上的負載。
- 在「MSCS 叢集管理者」中讓佇列管理程式連線, 並讓它接受測試工作量, 以測試佇列管理程式。如果您正在實驗測試佇列管理程式, 請使用「IBM WebSphere MQ 探險家」。例如:
 - 用滑鼠右鍵按一下 Queues 樹狀結構節點, 然後選取 New->Local Queue..., 並為佇列提供名稱。
 - 按一下 Finish。即會建立佇列, 並顯示在內容視圖中。
 - 用滑鼠右鍵按一下佇列, 然後選取 Put Test Message...。即會顯示「放置測試訊息」畫面。
 - 鍵入部分訊息文字, 然後按一下 Put Test Message, 並關閉畫面。
 - 用滑鼠右鍵按一下佇列, 然後選取 Browse Messages...。即會顯示「訊息瀏覽器」畫面。
 - 確定您的訊息在佇列上, 然後按一下 Close。即會關閉「訊息瀏覽器」畫面。
 - 用滑鼠右鍵按一下佇列, 然後選取 Clear Messages...。清除佇列上的訊息。
 - 用滑鼠右鍵按一下佇列, 然後選取 Delete...。即會顯示確認畫面, 按一下 OK。已刪除佇列。
 - 使用「MSCS 叢集管理者」測試佇列管理程式是否可以離線並回到線上。
 - 模擬失效接手。

在「MSCS 叢集管理者」中, 用滑鼠右鍵按一下包含佇列管理程式的群組, 然後選取 Move Group。這可能需要幾分鐘才能完成。(如果在其他時間您想要快速將佇列管理程式移至另一個節點, 請遵循第 277 頁的『將佇列管理程式移至 MSCS 儲存體』中的程序。) 您也可以按一下滑鼠右鍵選取 Initiate Failure; 動作 (本端重新啟動或失效接手) 取決於現行狀態及配置設定。

Windows Server 2008

若要在 Windows Server 2008 上將佇列管理程式置於 MSCS 控制下, 請使用下列程序:

- 登入管理佇列管理程式的叢集節點電腦, 或以具有叢集管理許可權的使用者身分登入遠端工作站, 並連接管理佇列管理程式的叢集節點。
- 啟動「失效接手叢集管理」工具。
- 用滑鼠右鍵按一下 **失效接手叢集管理 > 管理叢集 ...** 以開啟與叢集的連線。
- 與舊版 Windows 上 MSCS 叢集管理者中使用的群組方法相反, 「失效接手叢集管理」工具會使用服務及應用程式的概念。已配置的服務或應用程式包含要叢集一個應用程式所需的所有資源。您可以在 MSCS 下配置佇列管理程式, 如下所示:
 - 用滑鼠右鍵按一下 **服務及應用程式 > 配置服務或應用程式 ...** 以啟動配置精靈。
 - 在「選取服務或應用程式」畫面上選取 **其他伺服器**。

- c. 選取適當的 IP 位址作為用戶端存取點。

此位址應該是未用的 IP 位址，供用戶端及其他佇列管理程式用來連接至 虛擬 佇列管理程式。此 IP 位址不是任一節點的正常 (靜態) 位址; 它是它們之間 浮動 的額外位址。雖然 MSCS 會處理此位址的遞送，但它 **不會** 驗證是否可以呼叫到位址。

- d. 指派儲存裝置供佇列管理程式專用。此裝置需要先建立為資源實例，然後才能指派。

您可以使用一個磁碟機來同時儲存日誌和佇列檔案，也可以將它們分散在各磁碟機中。在任一情況下，如果每一個佇列管理程式都有自己的共用磁碟，請確保此佇列管理程式所使用的所有磁碟機都是此佇列管理程式專用的，亦即，沒有其他任何磁碟機依賴這些磁碟機。此外，請確保為佇列管理程式使用的每個磁碟機建立資源實例。

磁帶機的資源類型視您使用的 SCSI 支援而定; 請參閱 SCSI 配接卡指示。每一個共用磁碟機可能已有群組和資源。如果是這樣，您不需要為每一個磁碟機建立資源實例。將它從現行群組移至針對佇列管理程式所建立的群組。

針對每一個磁碟機資源，將可能的擁有者設為兩個節點。將相依資源設為無。

- e. 在「選取資源類型」畫面上選取 **IBM MQSeries MSCS** 資源。

- f. 完成精靈中的其餘步驟。

5. 在使資源上線之前，IBM MQSeries MSCS 資源需要其他配置:

- a. 選取新定義的服務，其中包含稱為「新建 IBM MQSeries MSCS」的資源。

- b. 用滑鼠右鍵按一下 MQ 資源上的 **內容**。

- c. 配置資源:

- Name; 選擇可讓您輕鬆識別其適用的佇列管理程式的名稱。
- Run in a separate Resource Monitor; 用於更好的隔離
- Possible owners; 設定兩個節點
- Dependencies; 新增此佇列管理程式的磁碟機及 IP 位址。

警告: 無法新增這些相依關係表示 WebSphere MQ 會在失效接手期間嘗試將佇列管理程式狀態寫入錯誤叢集磁碟。因為許多處理程序可能正在嘗試同步寫入此磁碟，所以部分 IBM WebSphere MQ 處理程序可能會被封鎖而無法執行。

- Parameters; 如下所示:

- QueueManagerName (必要); 此資源要控制的佇列管理程式名稱。此佇列管理程式必須存在於本端電腦上。
- PostOnlineCommand (選用); 您可以指定每當佇列管理程式資源將其狀態從離線變更為線上時要執行的程式。如需詳細資料，請參閱 [第 286 頁的『PostOnline 指令和 PreOffline 指令』](#)。
- PreOfflineCommand (選用); 您可以指定每當佇列管理程式資源將其狀態從連線變更為離線時要執行的程式。如需詳細資料，請參閱 [第 286 頁的『PostOnline 指令和 PreOffline 指令』](#)。

註: looksAlive 輪詢間隔設為預設值 5000 毫秒。isAlive 輪詢間隔設為預設值 60000 毫秒。只有在完成資源定義之後，才能修改這些預設值。如需進一步詳細資料，請參閱 [第 283 頁的『looksAlive 和 isAlive 輪詢的摘要』](#)。

- d. 選擇性地設定偏好節點 (但請注意 [第 286 頁的『使用偏好的節點』](#) 中的註解)

- e. 依預設，失效接手原則 會設為可感應值，但您可以調整臨界值及期間，以控制 資源失效接手 及 群組失效接手，以符合佇列管理程式上的負載。

6. 在「MSCS 叢集管理者」中讓佇列管理程式連線，並讓它接受測試工作量，以測試佇列管理程式。如果您正在實驗測試佇列管理程式，請使用「IBM WebSphere MQ 探險家」。例如:

- a. 用滑鼠右鍵按一下 Queues 樹狀結構節點，然後選取 New->Local Queue...，並為佇列提供名稱。

- b. 按一下 Finish。即會建立佇列，並顯示在內容視圖中。

- c. 用滑鼠右鍵按一下佇列，然後選取 Put Test Message...。即會顯示「放置測試訊息」畫面。

- d. 鍵入部分訊息文字，然後按一下 Put Test Message，並關閉畫面。

- e. 用滑鼠右鍵按一下佇列，然後選取 Browse Messages...。即會顯示「訊息瀏覽器」畫面。

- f. 確定您的訊息在佇列上，然後按一下 Close 。即會關閉「訊息瀏覽器」畫面。
 - g. 用滑鼠右鍵按一下佇列，然後選取 Clear Messages...。清除佇列上的訊息。
 - h. 用滑鼠右鍵按一下佇列，然後選取 Delete...。即會顯示確認畫面，按一下 OK。已刪除佇列。
7. 使用「MSCS 叢集管理者」測試佇列管理程式是否可以離線並回到線上。
 8. 模擬失效接手。

在「MSCS 叢集管理者」中，用滑鼠右鍵按一下包含佇列管理程式的群組，然後選取 Move Group。這可能需要幾分鐘才能完成。(如果在其他時間您想要快速將佇列管理程式移至另一個節點，請遵循第 277 頁的『將佇列管理程式移至 MSCS 儲存體』中的程序。)您也可以按一下滑鼠右鍵選取 Initiate Failure; 動作(本端重新啟動或失效接手)取決於現行狀態及配置設定。

Windows 2003

若要在 Windows 2003 上將佇列管理程式置於 MSCS 控制下，請使用下列程序：

1. 登入管理佇列管理程式的叢集節點電腦，或以具有叢集管理許可權的使用者身分登入遠端工作站，並連接管理佇列管理程式的叢集節點。
2. 啟動 MSCS 叢集管理者。
3. 開啟與叢集的連線。
4. 建立 MSCS 群組，以用來包含佇列管理程式的資源。請以很明顯與哪個佇列管理程式相關的方式來命名群組。每個群組都可以包含多個佇列管理程式，如第 276 頁的『搭配使用多個佇列管理程式與 MSCS』中所述。

將群組用於所有剩餘步驟。

5. 為佇列管理程式使用的每一個 SCSI 邏輯磁碟機建立資源實例。

您可以使用一個磁碟機來同時儲存日誌和佇列檔案，也可以將它們分散在各磁碟機中。在任一情況下，如果每一個佇列管理程式都有自己的共用磁碟，請確保此佇列管理程式所使用的所有磁碟機都是此佇列管理程式專用的，亦即，沒有其他任何磁碟機依賴這些磁碟機。此外，請確保為佇列管理程式使用的每個磁碟機建立資源實例。

磁帶機的資源類型視您使用的 SCSI 支援而定；請參閱 SCSI 配接卡指示。每一個共用磁碟機可能已有群組和資源。如果是這樣，您不需要為每一個磁碟機建立資源實例。將它從現行群組移至針對佇列管理程式所建立的群組。

針對每一個磁碟機資源，將可能的擁有者設為兩個節點。將相依資源設為無。

6. 建立 IP 位址的資源實例。

建立 IP 位址資源(資源類型 *IP address*)。此位址應該是未用的 IP 位址，供用戶端及其他佇列管理程式用來連接至 虛擬 佇列管理程式。此 IP 位址不是任一節點的正常(靜態)位址；它是它們之間 浮動的 額外位址。雖然 MSCS 會處理此位址的遞送，但它 **不會** 驗證是否可以呼叫到位址。

7. 建立佇列管理程式的資源實例。

建立類型為 *IBM WebSphere MQ MSCS* 的資源。精靈會提示您輸入各種項目，包括下列項目：

- Name; 選擇可讓您輕鬆識別其適用的佇列管理程式的名稱。
- Add to group; 使用您建立的群組
- Run in a separate Resource Monitor; 用於更好的隔離
- Possible owners; 設定兩個節點
- Dependencies; 新增此佇列管理程式的磁碟機及 IP 位址。

警告: 無法新增這些相依關係表示 WebSphere MQ 會在失效接手期間嘗試將佇列管理程式狀態寫入錯誤叢集磁碟。因為許多處理程序可能正在嘗試同步寫入此磁碟，所以部分 IBM WebSphere MQ 處理程序可能會被封鎖而無法執行。

- Parameters; 如下所示：

- QueueManagerName (必要); 此資源要控制的佇列管理程式名稱。此佇列管理程式必須存在於本端電腦上。

- `PostOnlineCommand` (選用); 您可以指定每當佇列管理程式資源將其狀態從離線變更為線上時要執行的程式。如需詳細資料, 請參閱第 286 頁的『`PostOnline` 指令和 `PreOffline` 指令』。
- `PreOfflineCommand` (選用); 您可以指定每當佇列管理程式資源將其狀態從連線變更為離線時要執行的程式。如需詳細資料, 請參閱第 286 頁的『`PostOnline` 指令和 `PreOffline` 指令』。

註: `looksAlive` 輪詢間隔設為預設值 5000 毫秒。 `isAlive` 輪詢間隔設為預設值 30000 毫秒。只有在完成資源定義之後, 才能修改這些預設值。如需進一步詳細資料, 請參閱第 283 頁的『`looksAlive` 和 `isAlive` 輪詢的摘要』。

8. 選擇性地設定偏好節點 (但請注意第 286 頁的『使用偏好的節點』中的註解)
9. 依預設會將 失效接手原則 (如群組內容中所定義) 設為可感應值, 但您可以調整臨界值及期間, 以控制資源失效接手及群組失效接手, 以符合佇列管理程式上放置的負載。
10. 在「MSCS 叢集管理者」中讓佇列管理程式連線, 並讓它接受測試工作量, 以測試佇列管理程式。如果您正在實驗測試佇列管理程式, 請使用「IBM WebSphere MQ 探險家」。例如:
 - a. 用滑鼠右鍵按一下 `Queues` 樹狀結構節點, 然後選取 `New->Local Queue...`, 並為佇列提供名稱。
 - b. 按一下 `Finish`。即會建立佇列, 並顯示在內容視圖中。
 - c. 用滑鼠右鍵按一下佇列, 然後選取 `Put Test Message...`。即會顯示「放置測試訊息」畫面。
 - d. 鍵入部分訊息文字, 然後按一下 `Put Test Message`, 並關閉畫面。
 - e. 用滑鼠右鍵按一下佇列, 然後選取 `Browse Messages...`。即會顯示「訊息瀏覽器」畫面。
 - f. 確定您的訊息在佇列上, 然後按一下 `Close`。即會關閉「訊息瀏覽器」畫面。
 - g. 用滑鼠右鍵按一下佇列, 然後選取 `Clear Messages...`。清除佇列上的訊息。
 - h. 用滑鼠右鍵按一下佇列, 然後選取 `Delete...`。即會顯示確認畫面, 按一下 `OK`。已刪除佇列。
11. 使用「MSCS 叢集管理者」測試佇列管理程式是否可以離線並回到線上。
12. 模擬失效接手。

在「MSCS 叢集管理者」中, 用滑鼠右鍵按一下包含佇列管理程式的群組, 然後選取 `Move Group`。這可能需要幾分鐘才能完成。(如果在其他時間您想要快速將佇列管理程式移至另一個節點, 請遵循第 277 頁的『將佇列管理程式移至 MSCS 儲存體』中的程序。) 您也可以按一下滑鼠右鍵選取 `Initiate Failure`; 動作 (本端重新啟動或失效接手) 取決於現行狀態及配置設定。

looksAlive 和 isAlive 輪詢的摘要

`looksAlive` 及 `isAlive` 是 MSCS 回呼資源類型所提供程式庫程式碼及要求資源執行檢查以判定其本身工作狀態的間隔。這最終決定 MSCS 是否嘗試對資源進行失效接手。

每次經歷 `looksAlive` 間隔 (預設為 5000 毫秒) 時, 都會呼叫佇列管理程式資源來執行自己的檢查, 以判斷其狀態是否令人滿意。

每次經歷 `isAlive` 間隔 (預設為 30000 毫秒) 時, 都會再次呼叫佇列管理程式資源, 讓它執行另一項檢查, 以判斷資源是否正常運作。這會啟用兩個層次的資源類型檢查。

1. `looksAlive` 狀態檢查, 以確定資源是否似乎正常運作。
2. 更重要的 `isAlive` 檢查, 可判定佇列管理程式資源是否處於作用中。

如果判定佇列管理程式資源不在作用中, 則 MSCS 會根據其他進階 MSCS 選項, 對叢集中的另一個節點觸發資源及相關相依資源的失效接手。如需進一步資訊, 請參閱 [MSCS 文件](#)。

從 MSCS 控制項移除佇列管理程式

您可以從 MSCS 控制項中移除佇列管理程式, 並將它們傳回給手動管理。

您不需要從 MSCS 控制中移除佇列管理程式, 即可進行維護作業。您可以使用「MSCS 叢集管理者」, 讓佇列管理程式暫時離線。從 MSCS 控制中移除佇列管理程式是更永久的變更; 只有在您決定不再想要 MSCS 對佇列管理程式有任何進一步的控制時, 才會這麼做。

如果要移除的佇列管理程式使用 SSL 連線, 您必須使用「WebSphere MQ 探險家」或 MQSC 指令 `ALTER QMGR` 來修改佇列管理程式屬性 `SSLKEYR`, 以指向本端目錄上的 SSL 金鑰儲存庫檔案。

程序如下:

1. 使用「MSCS 叢集管理者」讓佇列管理程式資源離線，如 [第 284 頁的『從 MSCS 使佇列管理程式離線』](#) 中所述。
2. 毀損資源實例。這不會毀損佇列管理程式。
3. 選擇性地將佇列管理程式檔案從共用磁碟機移轉回本端磁碟機。若要執行此動作，請參閱 [第 284 頁的『從 MSCS 儲存體傳回佇列管理程式』](#)。
4. 測試佇列管理程式。

從 MSCS 使佇列管理程式離線

若要讓佇列管理程式從 MSCS 離線，請執行下列步驟：

1. 啟動 MSCS 叢集管理者。
2. 開啟與叢集的連線。
3. 選取 Groups，然後開啟包含要移動之佇列管理程式的群組。
4. 選取佇列管理程式資源。
5. 用滑鼠右鍵按一下它並選取 Offline。
6. 等待完成。

從 MSCS 儲存體傳回佇列管理程式

此程序會將佇列管理程式配置回其電腦的本端磁碟機，亦即，它會變成一般 WebSphere MQ 佇列管理程式。為了達到此目的，您可以從共用磁碟中移動日誌檔及資料檔案。例如，現有的佇列管理程式可能有 E:\WebSphere MQ\log\hamvmqm 公用程式：

1. 關閉佇列管理程式，並檢查是否沒有錯誤。
2. 對佇列檔及日誌檔進行完整媒體備份，並將備份儲存在安全的地方（請參閱 [第 285 頁的『佇列管理程式日誌檔』](#)，以瞭解為何這很重要）。
3. 決定要使用哪一個本端磁碟機，並確定它有足夠容量來儲存佇列管理程式日誌檔及資料（佇列）檔案。
4. 請確定檔案目前所在的共用磁碟在要將佇列管理程式日誌及資料檔移至其中的叢集節點上處於線上狀態。
5. 執行公用程式來移動佇列管理程式，如下所示：

```
hamvmqm /m qmname /dd "c:\
WebSphere MQ" /ld "c:\
WebSphere MQ\log"
```

將佇列管理程式名稱替換為 *qmname*，將本端磁碟機代號替換為 *c*，並將您選擇的目錄替換為 *WebSphere MQ*（如果目錄尚未存在，則會建立）。

6. 測試佇列管理程式以確保其運作（如 [第 277 頁的『將佇列管理程式移至 MSCS 儲存體』](#) 中所述）。

使用 MSCS 的提示和要訣

本節包含一些一般資訊，可協助您有效地使用 MSCS 的 WebSphere MQ 支援。

本節包含一些一般資訊，可協助您有效地使用 MSCS 的 WebSphere MQ 支援。

將佇列管理程式從一部機器失效接手至另一部機器需要多久時間？這在很大程度上取決於佇列管理程式上的工作量，以及資料流量的混合，例如，在同步點內持續的工作量，以及在失敗之前確定的工作量。IBM 測試已提供大約一分鐘的失效接手及失效回復時間。這是在非常輕負載的佇列管理程式上，實際時間會因負載而有很大的不同。

驗證 MSCS 是否在運作中

請遵循下列步驟，以確保您具有執行中 MSCS 叢集。

以 [第 277 頁的『建立佇列管理程式以與 MSCS 搭配使用』](#) 開頭的作業說明假設您有一個執行中的 MSCS 叢集，您可以在其中建立、移轉及毀損資源。如果您想要確定您有這類叢集：

1. 使用 MSCS 叢集管理者來建立群組。
2. 在該群組內，建立同屬應用程式資源的實例，並指定系統時鐘 (路徑名稱 C:\winnt\system32\clock.exe 及 C:\ 的工作目錄)。
3. 請確定您可以讓資源上線、將包含它的群組移至另一個節點，以及讓資源離線。

手動啟動

對於 MSCS 所管理的佇列管理程式，您 **必須** 將啟動屬性設為手動。這可確保 WebSphere MQ MSCS 支援可以重新啟動 IBM MQSeries 服務，而無需立即啟動佇列管理程式。

WebSphere MQ MSCS 支援必須能夠重新啟動服務，它才能執行監視和控制，但本身必須仍控制哪些佇列管理程式在執行中，以及在哪些機器上。如需相關資訊，請參閱第 277 頁的『將佇列管理程式移至 MSCS 儲存體』。

MSCS 及佇列管理程式

使用 MSCS 時有關佇列管理程式的考量。

在另一個節點上建立相符的佇列管理程式

為了讓叢集作業與 WebSphere MQ 搭配使用，節點 B 上的每一個節點 A 都需要相同的佇列管理程式。不過，您不需要明確建立第二個。您可以在一個節點上建立或準備佇列管理程式，將它移至另一個節點，如第 277 頁的『將佇列管理程式移至 MSCS 儲存體』中所述，且在該節點上完全重複。

預設佇列管理程式

請勿在 MSCS 控制下使用預設佇列管理程式。佇列管理程式沒有使它成為預設值的內容; WebSphere MQ 會保留它自己的個別記錄。如果您在失效接手時將佇列管理程式集移至另一部電腦，則它不會變成預設值。讓您的所有應用程式依名稱參照特定的佇列管理程式。

刪除佇列管理程式

一旦佇列管理程式已移動節點，其詳細資料即會存在於兩台電腦上的登錄中。當您想要刪除它時，請在一部電腦上正常執行，然後執行第 287 頁的『IBM WebSphere MQ MSCS 支援公用程式』中說明的公用程式，以清除另一部電腦上的登錄。

支援現有的佇列管理程式

您可以將現存的佇列管理程式置於 MSCS 控制之下，前提是您可以將佇列管理程式日誌檔及佇列檔放置在兩部機器之間共用 SCSI 匯流排的磁碟上 (請參閱第 275 頁的圖 62)。建立 MSCS 資源時，您需要讓佇列管理程式短暫離線。

如果您要建立新的佇列管理程式，請獨立於 MSCS 建立它，測試它，然後將它置於 MSCS 控制之下。請參閱：

- 第 277 頁的『建立佇列管理程式以與 MSCS 搭配使用』
- 第 277 頁的『將佇列管理程式移至 MSCS 儲存體』
- 第 278 頁的『將佇列管理程式置於 MSCS 控制下』

告知 MSCS 要管理哪些佇列管理程式

您可以使用「MSCS 叢集管理者」為每一個這類佇列管理程式建立資源實例，以選擇哪些佇列管理程式置於 MSCS 控制之下。此處理程序會呈現資源清單，您可以從中選取要讓該實例管理的佇列管理程式。

佇列管理程式日誌檔

當您將佇列管理程式移至 MSCS 儲存體時，會將其日誌及資料檔移至共用磁碟 (如需範例，請參閱第 277 頁的『將佇列管理程式移至 MSCS 儲存體』)。

在移動之前，建議您完全關閉佇列管理程式，並對資料檔案和日誌檔進行完整備份。

多個佇列管理程式

WebSphere MQ MSCS 支援可讓您在每一部機器上執行多個佇列管理程式，並將個別佇列管理程式置於 MSCS 控制之下。

一律使用 MSCS 來管理叢集

請勿嘗試使用控制指令或「IBM WebSphere MQ 探險家」，直接在 MSCS 控制下的任何佇列管理程式上執行啟動及停止作業。相反地，請使用「MSCS 叢集管理者」讓佇列管理程式上線或離線。

使用 MSCS 叢集管理者部分是為了避免 MSCS 報告佇列管理程式已離線 (事實上您已在 MSCS 控制之外啟動它) 可能造成的混淆。更嚴重的是，MSCS 偵測到未使用 MSCS 停止佇列管理程式失敗，會起始失效接手至另一個節點。

在主動/主動模式下工作

MSCS 叢集中的兩台電腦都可以在「主動/主動」模式下執行佇列管理程式。您不需要讓完全閒置的機器作為待命機器 (但如果您想要的話，可以在「主動/被動模式」中)。

如果您計劃使用兩部機器來執行工作量，請為每一部機器提供足夠容量 (處理器、記憶體、次要儲存體)，以令人滿意的效能層次來執行整個叢集工作量。

註: 如果您將 MSCS 與 Microsoft Transaction Server (COM+) 一起使用，則無法使用「主動/主動」模式。這是因為，若要搭配使用 WebSphere MQ 與 MSCS 及 COM+:

- 使用 WebSphere MQ 的 COM+ 支援的應用程式元件必須在與分散式交易協調程式 (DTC) (COM+ 的一部分) 相同的電腦上執行。
- 佇列管理程式也必須在相同電腦上執行。
- DTC 必須配置為 MSCS 資源，因此隨時只能在叢集中的其中一部電腦上執行。

PostOnline 指令和 PreOffline 指令

使用這些指令來整合 WebSphere MQ MSCS 支援與其他系統。您可以使用它們來發出 WebSphere MQ 指令，但有一些限制。

請在「參數」中指定 IBM WebSphere MQ MSCS 類型的資源。您可以使用它們來整合 WebSphere MQ MSCS 支援與其他系統或程序。例如，您可以指定傳送郵件訊息的程式名稱、啟動呼叫器，或產生要由另一個監視系統擷取的其他警示形式。

PostOnline 當資源從離線變更為線上時，會呼叫指令; 如果資源從線上變更為離線，則會呼叫 PreOffline 指令。依預設，當呼叫這些指令時，會從 Windows 系統目錄執行。因為 WebSphere MQ 使用 32 位元資源監視器處理程序，所以在 Windows 64 位元系統上，這是 \Windows\System64 目錄，而不是 \Windows\system32。如需相關資訊，請參閱 Windows x64 環境中檔案重新導向的相關 Microsoft 文件。這兩個指令都在用來執行「MSCS 叢集服務」的使用者帳戶下執行; 並且會非同步呼叫; WebSphere MQ MSCS 支援不會在繼續之前等待它們完成。這可消除它們可能封鎖或延遲進一步叢集作業的任何風險。

您也可以使用這些指令來發出 WebSphere MQ 指令，例如重新啟動要求端通道。不過，這些指令會在佇列管理程式狀態變更的時間點執行，因此它們不會預期執行長時間執行的功能，且不得對佇列管理程式的現行狀態進行假設; 很可能在佇列管理程式上線之後，管理者立即發出離線指令。

如果您要執行相對於佇列管理程式狀態的程式，請考量建立 MSCS Generic Application 資源類型的實例，將它們放置在與佇列管理程式資源相同的 MSCS 群組中，並使它們相對於佇列管理程式資源。

使用偏好的節點

當使用「主動/主動」模式來配置每一個佇列管理程式的偏好節點時，它會很有用。不過，一般而言，最好不要設定偏好的節點，而是依賴手動失效回復。

與其他一些相對無狀態的資源不同，佇列管理程式可能需要一段時間才能從一個節點失效接手 (或回復) 到另一個節點。為了避免不必要的中斷，請先測試已回復的節點，然後再讓佇列管理程式回到該節點。這會排除使用 immediate 失效回復設定。您可以將失效回復配置為在一天中的特定時間之間發生。

當您確定節點已完全回復時，最安全的路徑可能是手動將佇列管理程式移回所需的節點。這會阻止使用 preferred node 選項。

如果應用程式事件日誌中發生 *COM +* 錯誤

當您在新安裝的 MSCS 叢集上安裝 WebSphere MQ 時，您可能會在「應用程式事件」日誌中發現來源 COM + 及事件 ID 4691 所報告的錯誤。

這表示當 Microsoft Distributed Transaction Coordinator (MSDTC) 尚未配置成在 Microsoft Cluster Server (MSCS) 環境中執行時，您正在嘗試執行 WebSphere MQ。如需在叢集環境中配置 MSDTC 的相關資訊，請參閱 Microsoft 文件。

IBM WebSphere MQ MSCS 支援公用程式

您可以在命令提示字元下執行的 MSCS 公用程式的 IBM WebSphere MQ 支援清單。

IBM WebSphere MQ 支援 MSCS 包括下列公用程式：

登錄/取消登錄資源類型

haregtyp.exe

在您取消登錄 IBM WebSphere MQ MSCS 資源類型之後，就無法再建立該類型的任何資源。如果叢集內仍有該類型的實例，MSCS 不會讓您取消登錄資源類型：

1. 使用 MSCS 叢集管理者，停止在 MSCS 控制下執行的任何佇列管理程式，方法是讓它們離線，如第 284 頁的『從 MSCS 使佇列管理程式離線』中所述。
2. 使用 MSCS 叢集管理者，刪除資源實例。
3. 在命令提示字元中，輸入下列指令來取消登錄資源類型：

```
haregtyp /u
```

如果您要登錄類型(或稍後重新登錄它)，請在命令提示字元中輸入下列指令：

```
haregtyp /r
```

順利登錄 MSCS 檔案庫之後，如果您在安裝 IBM WebSphere MQ 之後尚未這樣做，則必須將系統重新開機。

將佇列管理程式移至 MSCS 儲存體

hamvmqm.exe

請參閱第 277 頁的『將佇列管理程式移至 MSCS 儲存體』。

從節點刪除佇列管理程式

hadltmqm.exe

假設您在叢集中有一個佇列管理程式，它已從一個節點移至另一個節點，現在您想要毀損它。在目前所在的節點上使用 IBM WebSphere MQ Explorer 來刪除它。它的登錄項目仍存在於另一部電腦上。若要刪除這些項目，請在該電腦上的提示處輸入下列指令：

```
hadltmqm /m qmname
```

其中 qmname 是要移除的佇列管理程式名稱。

檢查並儲存設定詳細資料

amqmsysn.exe

此公用程式會呈現一個對話框，顯示「IBM WebSphere MQ MSCS 支援中心」設定的完整詳細資料，例如您致電 IBM 支援中心時可能要求的詳細資料。有一個選項可將詳細資料儲存至檔案。

多重實例佇列管理程式

多重實例佇列管理程式是指在不同伺服器上配置的不同佇列管理程式的實例。佇列管理程式的其中一個實例會定義為作用中實例，另一個實例會定義為待命實例。如果作用中實例失敗，多重實例佇列管理程式會在待命伺服器上自動重新啟動。

第 288 頁的圖 63 顯示 QM1 的多重實例配置。IBM WebSphere MQ 安裝在兩部伺服器上，其中一部是備用伺服器。已建立一個佇列管理程式 QM1。QM1 的一個實例處於作用中，且在一部伺服器上執行。QM1 的另一個實例正在另一部伺服器上待命執行，未執行作用中處理程序，但已準備好在作用中實例失敗時從 QM1 的作用中實例接管。

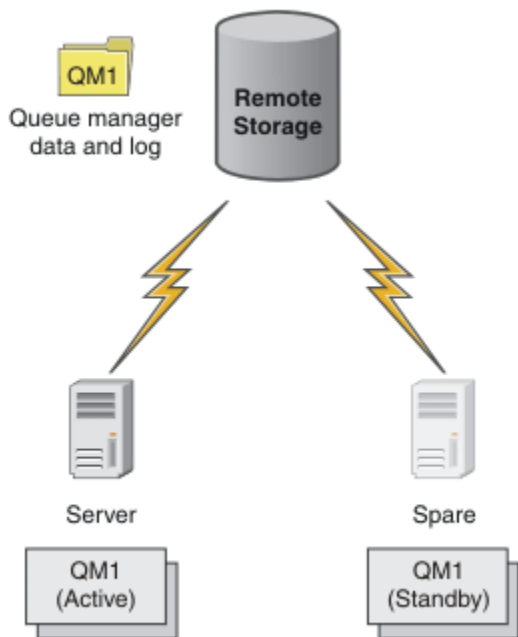


圖 63: 多重實例佇列管理程式

當您想要使用佇列管理程式作為多重實例佇列管理程式時，請使用 **crtmqm** 指令在其中一部伺服器上建立單一佇列管理程式，並將其佇列管理程式資料及日誌放置在共用網路儲存體中。在其他伺服器上，請使用 **addmqinf** 指令來建立對網路儲存體上佇列管理程式資料及日誌的參照，而不要再次建立佇列管理程式。

您現在可以從任一伺服器執行佇列管理程式。每一部伺服器都會參照相同的佇列管理程式資料和日誌；只有一個佇列管理程式，且一次只能在一部伺服器上作用。

佇列管理程式可以作為單一實例佇列管理程式或多重實例佇列管理程式來執行。在這兩種情況下，只有一個佇列管理程式實例在執行中，會處理要求。差別在於作為多重實例佇列管理程式執行時，未執行佇列管理程式作用中實例的伺服器會作為待命實例執行，並準備在作用中伺服器失敗時自動從作用中實例接管。

您對哪些實例最先變成作用中的唯一控制項，是您在兩部伺服器上啟動佇列管理程式的順序。第一個獲得佇列管理程式資料讀寫鎖定的實例會變成作用中實例。

您可以在作用中實例啟動之後，使用切換選項來停止作用中實例，以將控制傳送至待命伺服器，即可將作用中實例交換至其他伺服器。

QM1 的作用中實例在執行時，對共用佇列管理程式資料及日誌資料夾具有專用存取權。QM1 的待命實例會偵測作用中實例何時失敗，並變成作用中實例。它會接管處於作用中實例所保留狀態的 QM1 資料及日誌，並接受來自用戶端及通道的重新連線。

作用中實例可能因各種導致待命接管的原因而失敗：

- 管理作用中佇列管理程式實例的伺服器失敗。
- 管理作用中佇列管理程式實例的伺服器與檔案系統之間的連線功能失敗。
- WebSphere MQ 偵測到佇列管理程式處理程序無回應性，然後會關閉佇列管理程式。

您可以將佇列管理程式配置資訊新增至多部伺服器，並選擇任兩部伺服器作為作用中/待命配對來執行。總共有兩個實例的限制。您不能有兩個待命實例及一個作用中實例。

多重實例佇列管理程式是高可用性解決方案的一部分。您需要一些其他元件來建置有用的高可用性解決方案。

- 用戶端及通道重新連線，以將 WebSphere MQ 連線傳送至接管執行作用中佇列管理程式實例的電腦。

- 高效能共用網路檔案系統 (NFS)，可正確管理鎖定，並提供媒體及檔案伺服器故障的保護。
重要：您必須先停止在環境中執行的所有多重實例佇列管理程式實例，然後才能在 NFS 磁碟機上執行維護。如果 NFS 失敗，請確定您有要回復的佇列管理程式配置備份。
- 具復原力的網路和電源供應器，以消除基本基礎架構中的單一故障點。
- 容許失效接手的應用程式。特別是您需要密切注意交易式應用程式的行為，以及瀏覽 WebSphere MQ 佇列的應用程式。
- 監視及管理作用中及待命實例，以確保它們在執行中，並重新啟動失敗的作用中實例。雖然多重實例佇列管理程式會自動重新啟動，但您需要確定待命實例正在執行中，已準備好接管，且失敗的實例會重新回到線上作為新的待命實例。

當待命佇列管理程式變成作用中時，WebSphere MQ MQI 用戶端及通道會自動重新連接至待命佇列管理程式。如需重新連線及高可用性解決方案中其他元件的相關資訊，請參閱相關主題。Java 的 IBM WebSphere MQ 類別不支援自動用戶端重新連接。

支援的平台

您可以從 7.0.1 版開始，在任何非 z/OS 平台上建立多重實例佇列管理程式。

從 7.0.1 版開始，MQI 用戶端支援自動用戶端重新連線。

建立多重實例佇列管理程式

建立多重實例佇列管理程式，在一部伺服器上建立佇列管理程式，並在另一部伺服器上配置 IBM WebSphere MQ。多重實例佇列管理程式共用佇列管理程式資料及日誌。

建立多重實例佇列管理程式所涉及的大部分工作是設定共用佇列管理程式資料及日誌檔的作業。您必須在網路儲存體上建立共用目錄，並使用網路共用讓其他伺服器可以使用這些目錄。這些作業需要由具有管理權限的人員執行，例如 UNIX and Linux 系統上的 *root*。步驟如下所示：

1. 建立資料及日誌檔的共用。
2. 在一部伺服器上建立佇列管理程式。
3. 在第一部伺服器上執行指令 **dspmqinf**，以收集佇列管理程式配置資料，並將它複製到剪貼簿。
4. 使用所複製的資料執行指令 **addmqinf**，以在第二部伺服器上建立佇列管理程式配置。

您不會執行 **crtmqm** 在第二部伺服器上再次建立佇列管理程式。

檔案存取控制

您需要注意所有其他伺服器上的使用者和群組 *mqm* 具有存取共用的許可權。

在 UNIX and Linux 上，您需要使 *mqm* 的 *uid* 和 *gid* 在所有系統上都相同。您可能需要在每一個系統上編輯 */etc/passwd*，以針對 *mqm* 設定一般 *uid* 和 *gid*，然後將系統重新開機。

在 Microsoft Windows 上，執行佇列管理程式處理程序的使用者 ID 必須對包含佇列管理程式資料及日誌檔的目錄具有完全控制許可權。您可以使用兩種方式來配置許可權：

1. 建立以廣域群組作為替代安全主體的佇列管理程式。授權廣域群組對包含佇列管理程式資料及日誌檔的目錄具有完整控制存取權；請參閱第 312 頁的『[在 Windows 上保護共用佇列管理程式資料及日誌目錄和檔案的安全](#)』。使執行佇列管理程式的使用者 ID 成為廣域群組的成員。您無法讓本端使用者成為廣域群組的成員，因此佇列管理程式程序必須以網域使用者 ID 來執行。網域使用者 ID 必須是本端群組 *mqm* 的成員。作業第 291 頁的『[在網域工作站或伺服器上建立多重實例佇列管理程式](#)』示範如何使用以這種方式保護的檔案來設定多重實例佇列管理程式。
2. 在網域控制站上建立佇列管理程式，以便本端 *mqm* 群組具有網域範圍 "domain local"。保護與網域本端 *mqm* 的檔案共用，並在相同網域本端 *mqm* 群組下佇列管理程式的所有實例上執行佇列管理程式處理程序。作業第 303 頁的『[在網域控制站上建立多重實例佇列管理程式](#)』示範如何使用以這種方式保護的檔案來設定多重實例佇列管理程式。

配置資訊

修改每一部伺服器的相關 IBM WebSphere MQ 佇列管理程式配置資訊，以配置所需數目的佇列管理程式實例。每一部伺服器都必須安裝相容修正層次的相同 IBM WebSphere MQ 版本。指令 **dspmqlinf** 及 **addmqinf** 可協助您配置其他佇列管理程式實例。或者，您可以直接編輯 **mq.s.ini** 和 **qm.ini** 檔案。主題第 322 頁的『在 Linux 上建立多重實例佇列管理程式』、第 291 頁的『在網域工作站或伺服器上建立多重實例佇列管理程式』及第 303 頁的『在網域控制站上建立多重實例佇列管理程式』是顯示如何配置多重實例佇列管理程式的範例。

在 Windows、UNIX and Linux 系統上，您可以透過將單一 **mq.s.ini** 檔案放置在網路共用上，並將 **AMQ_MQS_INI_LOCATION** 環境變數設定為指向該檔案，來共用該檔案。

限制

1. 僅在具有相同作業系統、架構及排列法的伺服器上配置相同佇列管理程式的多個實例。例如，兩部機器都必須是 32 位元或 64 位元。
2. 所有 IBM WebSphere MQ 安裝都必須是版次 7.0.1 或更新版本。
3. 通常，作用中及待命安裝會維護在相同的維護層次。請參閱每一項升級的維護指示，以檢查您是否必須一起升級所有安裝。
請注意，主動和被動佇列管理程式的維護層次必須相同。
4. 僅在使用相同 IBM WebSphere MQ 使用者、群組及存取控制機制配置的佇列管理程式之間共用佇列管理程式資料及日誌。
5. 在 UNIX and Linux 系統上，使用 **硬性**、可岔斷、裝載而非 **軟性** 裝載來配置網路儲存體上的共用檔案系統。強迫岔斷裝載會強制佇列管理程式當掉，直到系統呼叫岔斷為止。在伺服器失敗之後，軟裝載不保證資料一致性。
6. 共用日誌及資料目錄無法儲存在 FAT 或 NFSv3 檔案系統上。對於 Windows 上的多重實例佇列管理程式，必須透過 Windows 網路所使用的「共用網際網路檔案系統 (CIFS)」通訊協定來存取網路儲存體。

Windows 網域及多重實例佇列管理程式

Windows 上的多重實例佇列管理程式需要共用其資料及日誌。共用必須可供在不同伺服器或工作站上執行的所有佇列管理程式實例存取。配置佇列管理程式並共用作為 Windows 網域的一部分。佇列管理程式可以在網域工作站或伺服器上執行，或在網域控制站上執行。

在配置多重實例佇列管理程式之前，請先閱讀第 315 頁的『在 Windows 上保護未共用佇列管理程式資料及日誌目錄和檔案的安全』及第 312 頁的『在 Windows 上保護共用佇列管理程式資料及日誌目錄和檔案的安全』，以檢閱如何控制對佇列管理程式資料及日誌檔的存取權。主題是教育性的；如果您想要直接前往 Windows 網域中設定多重實例佇列管理程式的共用目錄；請參閱第 291 頁的『在網域工作站或伺服器上建立多重實例佇列管理程式』。

在網域工作站或伺服器上執行多重實例佇列管理程式

從 Version 7.1 開始，多重實例佇列管理程式會在屬於網域成員的工作站或伺服器上執行。在 Version 7.1 之前，多重實例佇列管理程式只會在網域控制站上執行；請參閱第 291 頁的『在網域控制站上執行多重實例佇列管理程式』。若要在 Windows 上執行多重實例佇列管理程式，您需要網域控制站、檔案伺服器，以及兩個執行相同佇列管理程式的工作站或伺服器連接至相同網域。

可以在網域中任何伺服器或工作站上執行多重實例佇列管理程式的變更，是您現在可以建立具有其他安全群組的佇列管理程式。其他安全群組會在 **crtmqm** 指令的 **-a** 參數中傳遞。您可以使用群組來保護包含佇列管理程式資料及日誌的目錄。執行佇列管理程式處理程序的使用者 ID 必須是此群組的成員。當佇列管理程式存取目錄時，Windows 會檢查使用者 ID 必須存取目錄的許可權。透過同時提供群組及使用者 ID 網域範圍，執行佇列管理程式處理程序的使用者 ID 具有來自廣域群組的認證。當佇列管理程式在不同伺服器上執行時，執行佇列管理程式處理程序的使用者 ID 可以具有相同的認證。使用者 ID 不必相同。它必須是替代安全群組的成員，以及本端 **mqm** 群組的成員。

建立多重實例佇列管理程式的作業與 Version 7.0.1 中只有一項變更的作業相同。您必須將其他安全群組名稱新增至 **crtmqm** 指令的參數。該作業在第 291 頁的『在網域工作站或伺服器上建立多重實例佇列管理程式』中說明。

需要多個步驟來配置網域，以及網域伺服器和工作站。您必須瞭解 Windows 如何授權佇列管理程式存取其資料及日誌目錄。如果您不確定佇列管理程式處理程序如何獲授權存取其日誌及資料檔案，請閱讀第 315 頁的『在 Windows 上保護未共用佇列管理程式資料及日誌目錄和檔案的安全』主題。本主題包含兩項作業，可協助您瞭解所需步驟。作業有第 316 頁的『讀取及寫入本端 mqm 群組授權的資料及日誌檔』和第 319 頁的『讀取及寫入替代本端安全群組授權的資料及日誌檔』。另一個主題第 312 頁的『在 Windows 上保護共用佇列管理程式資料及日誌目錄和檔案的安全』說明如何使用替代安全群組來保護包含佇列管理程式資料及日誌檔的共用目錄。本主題包括四項作業：設定 Windows 網域、建立檔案共用、安裝 IBM WebSphere MQ for Windows，以及配置佇列管理程式以使用共用。作業如下：

1. 第 294 頁的『建立 IBM WebSphere MQ 的 Active Directory 及 DNS 網域』。
2. 第 296 頁的『在 Windows 網域中的伺服器或工作站上安裝 IBM WebSphere MQ』。
3. 第 299 頁的『建立佇列管理程式資料及日誌檔的共用目錄』。
4. 第 301 頁的『讀取及寫入由替代廣域安全群組授權的共用資料及日誌檔』。

然後，您可以使用網域來執行作業第 291 頁的『在網域工作站或伺服器上建立多重實例佇列管理程式』。在將知識傳送至正式作業網域之前，請先執行下列作業來探索設定多重實例佇列管理程式。

在網域控制站上執行多重實例佇列管理程式

在 Version 7.0.1 中，多重實例佇列管理程式僅在網域控制站上執行。可以使用網域 mqm 群組來保護佇列管理程式資料的安全。如第 312 頁的『在 Windows 上保護共用佇列管理程式資料及日誌目錄和檔案的安全』主題所說明，您無法與工作站或伺服器上的本端 mqm 群組共用受保護的目錄。不過，在網域控制站上，所有群組和主體都有網域範圍。如果您在網域控制站上安裝 IBM WebSphere MQ for Windows，則佇列管理程式資料及日誌檔會受到網域 mqm 群組（可共用）的保護。遵循作業第 303 頁的『在網域控制站上建立多重實例佇列管理程式』中的步驟，在網域控制站上配置多重實例佇列管理程式。

相關資訊

[Windows 2000、Windows Server 2003 及 Windows Server 2008 叢集節點作為網域控制站](#)

在網域工作站或伺服器上建立多重實例佇列管理程式

範例顯示如何在工作站或屬於 Windows 網域的伺服器上的 Windows 上設定多重實例佇列管理程式。伺服器不必是網域控制站。設定會示範所涉及的概念，而不是正式作業規模。此範例以 Windows Server 2008 為基礎。步驟在其他版本的 Windows Server 上可能有所不同。

在正式作業規模配置中，您可能需要根據現有網域來修改配置。例如，您可以定義不同的網域群組來授權不同的共用，以及將執行佇列管理程式的使用者 ID 分組。

範例配置由三部伺服器組成：

sun

Windows Server 2008 網域控制站。它擁有包含 *Sun*、*mars* 及 *venus* 的 *wmq.example.com* 網域。為了說明起見，它也用作檔案伺服器。

mars

用作第一個 IBM WebSphere MQ 伺服器的 Windows Server 2008。它包含一個多重實例佇列管理程式實例，稱為 *QMGR*。

venus

用作第二部 IBM WebSphere MQ 伺服器的 Windows Server 2008。它包含稱為 *QMGR* 之多重實例佇列管理程式的第二個實例。

將範例中的斜體名稱取代為您選擇的名稱。

開始之前

在 Windows 上，您不需要驗證計劃儲存佇列管理程式資料及日誌檔的檔案系統。檢查程序 [驗證共用檔案系統行為](#) 適用於 UNIX and Linux。在 Windows 上，檢查一律成功。

執行下列作業中的步驟。這些作業會建立網域控制站及網域，在一部伺服器上安裝 IBM WebSphere MQ for Windows，並建立資料及日誌檔的檔案共用。如果您要配置現有的網域控制站，可能會發現在新的 Windows Server 2008 上試用這些步驟很有用。您可以根據您的網域調整步驟。

1. 第 294 頁的『建立 IBM WebSphere MQ 的 Active Directory 及 DNS 網域』。

2. 第 296 頁的『在 Windows 網域中的伺服器或工作站上安裝 IBM WebSphere MQ』。
3. 第 299 頁的『建立佇列管理程式資料及日誌檔的共用目錄』。
4. 第 301 頁的『讀取及寫入由替代廣域安全群組授權的共用資料及日誌檔』。

關於這項作業

此作業是一系列作業之一，用來配置網域中的網域控制站及兩部伺服器，以執行佇列管理程式的實例。在這項作業中，您將配置第二部伺服器 *venus*，以執行另一個佇列管理程式實例 *QMGR*。遵循此作業中的步驟，以建立佇列管理程式 *QMGR* 的第二個實例，並測試它是否運作。

此作業與前一節中的四個作業不同。它包含將單一實例佇列管理程式轉換為多重實例佇列管理程式的步驟。所有其他步驟都適用於單一或多重實例佇列管理程式。

程序

1. 配置第二部伺服器以執行 IBM WebSphere MQ for Windows.
 - a) 執行作業 第 296 頁的『在 Windows 網域中的伺服器或工作站上安裝 IBM WebSphere MQ』中的步驟，以建立第二個網域伺服器。在此作業序列中，第二部伺服器稱為 *venus*。

提示: 在兩部伺服器上，使用相同的 IBM WebSphere MQ 安裝預設值來建立第二個安裝。如果預設值不同，您可能需要在 IBM WebSphere MQ 配置檔 *mqs.ini* 的 **QMGR QueueManager** 段落中，修改字首和 *InstallationName* 變數。變數是指每一個伺服器上每一個安裝及佇列管理程式可能不同的路徑。如果每個伺服器上的路徑都保持相同，則配置多重實例佇列管理程式會更簡單。
2. 在 *venus* 上建立第二個 *QMGR* 實例。
 - a) 如果 *mars* 上的 *QMGR* 不存在，請執行作業 第 301 頁的『讀取及寫入由替代廣域安全群組授權的共用資料及日誌檔』，以建立它
 - b) 檢查 *venus* 的字首和 *InstallationName* 參數值是否正確。

在 *mars* 上，執行 **dspmqlinf** 指令：

```
dspmqlinf QMGR
```

系統回應：

```
QueueManager:  
  Name=QMGR  
  Directory=QMGR  
  Prefix=C:\Program Files\IBM\WebSphere MQ  
  DataPath=\\sun\wmq\data\QMGR  
  InstallationName=Installation1
```

- c) 將 **QueueManager** 段落的機器可讀形式複製到剪貼簿。

在 *mars* 上，使用 **-o command** 參數再次執行 **dspmqlinf** 指令。

```
dspmqlinf -o command QMGR
```

系統回應：

```
addmqinf -s QueueManager -v Name=QMGR  
-v Directory=QMGR -v Prefix="C:\Program Files\IBM\WebSphere MQ"  
-v DataPath=\\sun\wmq\data\QMGR
```

- d) 在 *venus* 上，從剪貼簿執行 **addmqinf** 指令，以在 *venus* 上建立佇列管理程式的實例。

必要的話，請調整指令，以容納 *Prefix* 或 *InstallationName* 參數中的差異。

```
addmqinf -s QueueManager -v Name=QMGR  
-v Directory=QMGR -v Prefix="C:\Program Files\IBM\WebSphere MQ"  
-v DataPath=\\sun\wmq\data\QMGR
```

WebSphere MQ configuration information added.

3. 在 *venus* 上啟動佇列管理程式 *QMGR*，允許待命實例。

a) 檢查 *mars* 上的 *QMGR* 是否已停止。

在 *mars* 上，執行 **dspmq** 指令：

```
dspmq -m QMGR
```

系統回應視佇列管理程式的停止方式而定；例如：

```
C:\Users\Administrator>dspmq -m QMGR
QMNAME(QMGR) STATUS(Ended immediately)
```

b) 在 *venus* 上，執行 **strmqm** 指令以啟動 *QMGR* 允許 standbys：

```
strmqm -x QMGR
```

系統回應：

```
WebSphere MQ queue manager 'QMGR' starting.
The queue manager is associated with installation 'Installation1'.
5 log records accessed on queue manager 'QMGR' during the log
replay phase.
Log replay for queue manager 'QMGR' complete.
Transaction manager state recovered for queue manager 'QMGR'.
WebSphere MQ queue manager 'QMGR' started using V7.1.0.0.
```

結果

若要測試多重實例佇列管理程式切換，請執行下列步驟：

1. 在 *mars* 上，執行 **strmqm** 指令以啟動 *QMGR* 允許 standbys：

```
strmqm -x QMGR
```

系統回應：

```
WebSphere MQ queue manager 'QMGR' starting.
The queue manager is associated with installation 'Installation1'.
A standby instance of queue manager 'QMGR' has been started.
The active instance is running elsewhere.
```

2. 在 *venus* 上，執行 **endmqm** 指令：

```
endmqm -r -s -i QMGR
```

venus 上的系統回應：

```
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ended, permitting switchover to
a standby instance.
```

以及在 *mars* 上：

```
dspmq
QMNAME(QMGR) STATUS(Running as standby)
C:\Users\wmquser2>dspmq
QMNAME(QMGR) STATUS(Running as standby)
```

```
C:\Users\wmquser2>dspmqr
QMNAME(QMGR) STATUS(Running)
```

下一步

若要使用範例程式來驗證多重實例佇列管理程式，請參閱 [第 310 頁的『在 Windows 上驗證多重實例佇列管理程式』](#)。

建立 *IBM WebSphere MQ* 的 *Active Directory* 及 *DNS* 網域

此作業會在稱為 *sun* 的 Windows 2008 網域控制站上建立網域 *wmq.example.com*。它會將網域中的 Domain *mqm* 廣域群組配置成具有正確的權限，並具有一個使用者。

在正式作業規模配置中，您可能需要根據現有網域來修改配置。例如，您可以定義不同的網域群組來授權不同的共用，以及將執行佇列管理程式的使用者 ID 分組。

範例配置由三部伺服器組成：

sun

Windows Server 2008 網域控制站。它擁有包含 *Sun*、*mars* 及 *venus* 的 *wmq.example.com* 網域。為了說明起見，它也用作檔案伺服器。

mars

用作第一個 *IBM WebSphere MQ* 伺服器的 Windows Server 2008。它包含一個多重實例佇列管理程式實例，稱為 *QMGR*。

venus

用作第二部 *IBM WebSphere MQ* 伺服器的 Windows Server 2008。它包含稱為 *QMGR* 之多重實例佇列管理程式的第二個實例。

將範例中的斜體名稱取代為您選擇的名稱。

開始之前

1. 作業步驟與已安裝但未配置任何角色的 Windows Server 2008 一致。如果您要配置現有的網域控制站，可能會發現在新的 Windows Server 2008 上試用這些步驟很有用。您可以根據您的網域調整步驟。

關於這項作業

在此作業中，您可以在新的網域控制站上建立 *Active Directory* 及 *DNS* 網域。然後，您可以配置它，以在加入網域的其他伺服器及工作站上安裝 *IBM WebSphere MQ*。如果您不熟悉安裝及配置 *Active Directory* 以建立 Windows 網域，請遵循此作業。您必須建立 Windows 網域，才能建立多重實例佇列管理程式配置。此作業並非以最佳方式引導您配置 Windows 網域。若要在正式作業環境中部署多重實例佇列管理程式，您必須參閱 Windows 文件。

在作業期間，您會執行下列步驟：

1. 安裝 *Active Directory*。
2. 新增網域。
3. 將網域新增至 *DNS*。
4. 建立廣域群組 Domain *mqm*，並為其提供正確的權限。
5. 新增使用者，並使其成為廣域群組 Domain *mqm* 的成員。

此作業是一組說明存取佇列管理程式資料及日誌檔的相關作業之一。這些作業顯示如何建立佇列管理程式，以授權讀取及寫入儲存在您選擇的目錄中的資料和日誌檔。它們隨附作業 [第 290 頁的『Windows 網域及多重實例佇列管理程式』](#)。

基於作業的目的，網域控制站主機名稱為 *sun*，且兩部 *IBM WebSphere MQ* 伺服器稱為 *mars* 及 *venus*。網域稱為 *wmq.example.com*。您可以將作業中所有斜體名稱取代為您自己選擇的名稱。

程序

1. 以本端或 *Workgroup* 管理者身分登入網域控制站 *sun*。

如果伺服器已配置為網域控制站，您必須以網域管理者身分登入。

2. 執行 Active Directory 網域服務精靈。

a) 按一下 **開始 > 執行 ...** 鍵入 `dcpromo`，然後按一下 **確定**。

如果尚未安裝 Active Directory 二進位檔，Windows 會自動安裝這些檔案。

3. 在精靈的第一個視窗中，保持不勾選 **使用進階模式安裝** 勾選框。按 **下一步 > 下一步**，然後按一下 **在新樹系中建立新網域 > 下一步**。

4. 在 **樹系根網域的 FQDN** 欄位中鍵入 `wmq.example.com`。按 **下一步**。

5. 在「設定樹系功能層次」視窗中，從 **樹系功能層次 > 下一步** 清單中選取 **Windows Server 2003** 或更新版本。

IBM WebSphere MQ 支援的 Windows Server 最舊層次是 Windows Server 2003。

6. 選擇性的：在「設定網域功能層次」視窗中，從 **網域功能層次 > 下一步** 清單中選取 **Windows Server 2003** 或更新版本。

只有在您將「樹系功能層次」設為 **Windows Server 2003** 時，才需要此步驟。

7. 即會開啟「其他網域控制站選項」視窗，並選取 **DNS 伺服器** 作為其他選項。按 **下一步 及 是**，以清除警告視窗。

提示：如果已安裝 DNS 伺服器，則不會向您呈現此選項。如果您想要精確遵循此作業，請從此網域控制站移除所有角色，然後重新啟動。

8. 維持 Database、Log Files 和 SYSVOL 目錄不變；按 **下一步**。

9. 在「目錄服務還原模式管理者密碼」視窗的 **密碼** 及 **確認密碼** 欄位中鍵入密碼。按 **下一步 > 下一步**。在最終精靈視窗中選取 **完成時重新開機**。

10. 當網域控制站重新開機時，請以 `wmq\Administrator` 身分登入。

伺服器管理程式會自動啟動。

11. 開啟 `wmq.example.com\Users` 資料夾

a) 開啟 **伺服器管理程式 > 角色 > Active Directory 網域服務 > wmq.example.com > 使用者**。

12. 用滑鼠右鍵按一下 **使用者 > 新建 > 群組**。

a) 在 **群組名稱** 欄位中鍵入群組名稱。

註：偏好的群組名稱為 Domain mqm。請按原樣鍵入內容。

- 呼叫群組 Domain mqm 會修改網域工作站或伺服器上「"準備 IBM WebSphere MQ"」精靈的行為。它會導致「"準備 IBM WebSphere MQ"」精靈自動將 Domain mqm 群組新增至網域中每一個新的 IBM WebSphere MQ 安裝上的本端 mqm 群組。
- 您可以在沒有 Domain mqm 廣域群組的網域中安裝工作站或伺服器。如果您這樣做，則必須定義內容和 Domain mqm 群組相同的群組。必須使該群組或者隸屬於該群組的使用者、本端 mqm 群組的成員（不管是安裝到網域中的哪個 IBM WebSphere MQ）。您可以將網域使用者放入多個群組中。建立多個網域群組，每個群組對應於您想要單獨管理的安裝集。根據網域使用者管理的安裝，將網域使用者分割成不同的網域群組。將每一個網域群組新增至不同 IBM WebSphere MQ 安裝的本端 mqm 群組。網域群組中僅隸屬於特定本端 mqm 群組的網域使用者可以針對該安裝建立、管理及執行佇列管理程式。
- 在網域中的工作站或伺服器上安裝 IBM WebSphere MQ 時指定的網域使用者必須是 Domain mqm 群組的成員，或與 Domain mqm 群組具有相同內容所定義的替代群組的成員。

b) 點選 **廣域** 作為 **群組範圍**，或者將其變更為 **通用**。點選 **安全** 作為 **群組類型**。按一下 **確定**。

13. 將權限 **容許 讀取群組成員資格** 及 **容許 讀取 groupMembershipSAM** 新增至 Domain mqm 廣域群組的權限。

a) 在「伺服器管理程式」動作列中，按一下 **檢視 > 進階特性**

b) 在「伺服器管理程式」導覽樹狀結構中，按一下 **使用者**

c) 在「使用者」視窗中，用滑鼠右鍵按一下 **Domain mqm > 內容**

d) 按一下 **安全 > 進階 > 新增 ...**。鍵入 Domain mqm，然後按一下 **檢查名稱 > 確定**。

名稱 欄位會預先填入字串 Domain mqm (`domain name\Domain mqm`)。

- e) 按一下「屬性」。在套用至清單中，選取清單底端的後代使用者物件。
 - f) 從許可權清單中，選取讀取群組成員資格和讀取 groupMembershipSAM 容許勾選框；按一下確定 > 套用 > 確定 > 確定。
14. 將兩個以上使用者新增至 Domain mqm 廣域群組。
- 範例中的一個使用者 *wmquser1* 執行 IBM WebSphere MQ 服務，而另一個使用者 *wmquser2* 以互動方式使用。
- 需要網域使用者來建立佇列管理程式，以使用網域配置中的替代安全群組。雖然管理者具有執行 **crtmqm** 指令的權限，但使用者 ID 不足以成為管理者。網域使用者 (可以是管理者) 必須是本端 mqm 群組及替代安全群組的成員。
- 在此範例中，您建立 Domain mqm 廣域群組的 *wmquser1* 和 *wmquser2* 成員。"準備 IBM WebSphere MQ" 精靈會自動將 Domain mqm 配置為執行精靈的本端 mqm 群組成員。
- 您必須提供不同的使用者，才能針對單一電腦上的每一個 IBM WebSphere MQ 安裝執行 IBM WebSphere MQ 服務。您可以在不同的電腦上重複使用相同的使用者。
- a) 在「伺服器管理程式」導覽樹狀結構中，按一下使用者 > 新建 > 使用者
 - b) 在「新建物件-使用者」視窗中，在使用者登入名稱欄位中鍵入 *wmquser1*。在名字欄位中鍵入 *WebSphere*，在姓氏欄位中鍵入 *MQ1*。按下一步。
 - c) 在密碼和確認密碼欄位中鍵入密碼，並清除使用者在下次登入時必須變更密碼勾選框。按下一步 > 完成。
 - d) 在「使用者」視窗中，用滑鼠右鍵按一下 **WebSphere MQ > 新增至群組 ...**。鍵入 Domain mqm，然後按一下檢查名稱 > 確定 > 確定。
 - e) 重複步驟 a 至 d，以將 *WebSphere MQ2* 新增為 *wmquser2*。

15. 以服務方式執行 IBM WebSphere MQ。

如果您需要以服務方式執行 IBM WebSphere MQ，然後向網域使用者 (從網域管理者取得) 提供以服務方式執行的權限，請執行下列程序：

- a) 按一下開始 > 執行 ...。
鍵入指令 `secpol.msc` 並按一下確定。
- b) 開啟安全設定 > 本機原則 > 使用者權限指派。
在原則清單中，用滑鼠右鍵按一下以服務方式登入 > 內容。
- c) 按一下新增使用者或群組 ...
鍵入您從網域管理者取得的使用者名稱，然後按一下檢查名稱
- d) 如果由「Windows 安全」視窗提示，請鍵入具有足夠權限之帳戶使用者或管理者的使用者名稱及密碼，然後按一下確定 > 套用 > 確定。
關閉「本機安全性原則」視窗。

註：在 Windows Vista 及 Windows Server 2008 上，依預設會啟用「使用者帳戶控制 (UAC)」。

UAC 特性會限制使用者可以對某些作業系統機能執行的動作，即使這些使用者隸屬於「管理者」群組。您必須採取適當的步驟才能克服此限制。

下一步

繼續進行下一項作業 [第 296 頁的『在 Windows 網域中的伺服器或工作站上安裝 IBM WebSphere MQ』](#)。

在 Windows 網域中的伺服器或工作站上安裝 IBM WebSphere MQ

在此作業中，您將在 *wmq.example.com* Windows 網域中的伺服器或工作站上安裝並配置 IBM WebSphere MQ。

在正式作業規模配置中，您可能需要根據現有網域來修改配置。例如，您可以定義不同的網域群組來授權不同的共用，以及將執行佇列管理程式的使用者 ID 分組。

範例配置由三部伺服器組成：

sun

Windows Server 2008 網域控制站。它擁有包含 *Sun*、*mars* 及 *venus* 的 *wmq.example.com* 網域。為了說明起見，它也用作檔案伺服器。

mars

用作第一個 IBM WebSphere MQ 伺服器的 Windows Server 2008。它包含一個多重實例佇列管理程式實例，稱為 *QMGR*。

venus

用作第二部 IBM WebSphere MQ 伺服器的 Windows Server 2008。它包含稱為 *QMGR* 之多重實例佇列管理程式的第二個實例。

將範例中的斜體名稱取代為您選擇的名稱。

開始之前

1. 執行第 294 頁的『建立 IBM WebSphere MQ 的 Active Directory 及 DNS 網域』中的步驟，以建立網域 *wmq.example.com* 的網域控制站 *sun*。變更斜體名稱以符合您的配置。
2. 如需您可以執行 IBM WebSphere MQ 的其他 Windows 版本，請參閱 [Windows 系統上的軟硬體需求](#)。

關於這項作業

在此作業中，您可以將 Windows Server 2008 (稱為 *mars*) 配置為 *wmq.example.com* 網域的成員。您安裝 IBM WebSphere MQ，並將安裝配置成以 *wmq.example.com* 網域的成員身分執行。

此作業是一組說明存取佇列管理程式資料及日誌檔的相關作業之一。這些作業顯示如何建立佇列管理程式，以授權讀取及寫入儲存在您選擇的目錄中的資料和日誌檔。它們隨附作業第 290 頁的『Windows 網域及多重實例佇列管理程式』。

基於作業的目的，網域控制站主機名稱為 *sun*，且兩部 IBM WebSphere MQ 伺服器稱為 *mars* 及 *venus*。網域稱為 *wmq.example.com*。您可以將作業中所有斜體名稱取代為您自己選擇的名稱。

程序

1. 將網域控制站 *sun.wmq.example.com* 新增至 *mars* 作為 DNS 伺服器。
 - a) 在 *mars* 上，以 *mars\Administrator* 身分登入，然後按一下 **開始**。
 - b) 用滑鼠右鍵按一下 **網路 > 內容 > 管理網路連線**。
 - c) 用滑鼠右鍵按一下網路配接卡，然後按一下 **內容**。
系統會回應「區域連線內容」視窗，列出連線使用的項目。
 - d) 從「本端區域連線內容」視窗的項目清單中，選取 **Internet Protocol 第 4 版** 或 **Internet Protocol 第 6 版**。按一下 **內容 > 進階 ...** 然後按一下 **DNS** 標籤。
 - e) 在 DNS 伺服器位址下，按一下 **新增 ...**。
 - f) 鍵入網域控制站 (也是 DNS 伺服器) 的 IP 位址，然後按一下 **新增**。
 - g) 按一下 **附加這些 DNS 字尾 > 新增 ...**。
 - h) 鍵入 *wmq.example.com*，然後按一下 **新增**。
 - i) 在 **此連線的 DNS 字尾** 欄位中鍵入 *wmq.example.com*。
 - j) 選取在 **DNS 中登錄此連線的位址** 和在 **DNS 登錄中使用此連線的字尾**。按一下 **確定 > 確定 > 關閉**
 - k) 開啟指令視窗，並鍵入指令 **ipconfig /all**，以檢閱 TCP/IP 設定。
2. 在 *mars* 上，將電腦新增至 *wmq.example.com* 網域。
 - a) 按一下 **開始**
 - b) 用滑鼠右鍵按一下 **電腦 > 內容**。在電腦名稱、網域及工作群組設定部門中，按一下 **變更設定**。
 - c) 在「系統內容」視窗中，按一下 **變更 ...**。
 - d) 按一下「網域」，鍵入 *wmq.example.com*，然後按一下 **確定**。
 - e) 鍵入網域控制站管理者的 **使用者名稱** 及 **密碼**，該管理者有權允許電腦加入網域，然後按一下 **確定**。

- f) 按一下 **確定** > **確定** > **關閉** > **立即重新啟動**，以回應 "歡迎使用 *wmq.example.com* 網域" 訊息。
3. 檢查電腦是否為 *wmq.example.com* 網域的成員
- 在 *sun* 上，以 *wmq\Administrator* 身分登入網域控制站。
 - 開啟 **伺服器管理程式** > **Active Directory 網域服務** > **wmq.example.com** > **電腦**，並在「電腦」視窗中正確地列出 *mars*。
4. 在 *mars* 上安裝 IBM WebSphere MQ for Windows。
- 如需執行 IBM WebSphere MQ for Windows 安裝精靈的進一步資訊；請參閱 [在 Windows 上安裝 IBM WebSphere MQ 伺服器](#)。
- 在 *mars* 上，以本端管理者 *mars\Administrator* 身分登入。
 - 在 IBM WebSphere MQ for Windows 安裝媒體上執行 **Setup** 指令。
即會啟動「IBM WebSphere MQ 啟動程式」應用程式。
 - 按一下 **軟體需求**，以檢查是否已安裝必備軟體。
 - 按一下 **網路配置** > **是**，以配置網域使用者 ID。
作業第 294 頁的『[建立 IBM WebSphere MQ 的 Active Directory 及 DNS 網域](#)』會配置這組作業的網域使用者 ID。
 - 按一下 **WebSphere MQ 安裝**，選取安裝語言，然後按一下「啟動 IBM IBM WebSphere MQ 安裝程式」。
 - 確認授權合約，然後按 **下一步** > **下一步** > **安裝**，以接受預設配置。等待安裝完成，然後按一下 **完成**。
您可能想要變更安裝名稱、安裝不同的元件、為佇列管理程式資料及日誌配置不同的目錄，或安裝至不同的目錄。如果是的話，請按一下 **自訂**，而不是一般。
已安裝 IBM WebSphere MQ，且安裝程式會啟動 "準備 IBM WebSphere MQ" 精靈。
重要: 尚未執行精靈。
5. 使用 **以服務方式執行** 權限來配置將要執行 IBM IBM WebSphere MQ 服務的使用者。
- 選擇是要配置本端 *mqm* 群組、Domain *mqm* 群組，還是要使用權限來執行 IBM IBM WebSphere MQ 服務的使用者。在此範例中，您為使用者提供權限。
- 按一下 **開始** > **執行 ...**，鍵入指令 **secpol.msc**，然後按一下 **確定**。
 - 開啟 **安全設定** > **本機原則** > **使用者權限指派**。在原則清單中，用滑鼠右鍵按一下 **以服務方式登入** > **內容**。
 - 按一下 **新增使用者或群組 ...** 並鍵入 *wmquser1*，然後按一下 **檢查名稱**
 - 鍵入網域管理者 *wmq\Administrator* 的使用者名稱及密碼，然後按一下 **確定** > **套用** > **確定**。關閉「本機安全性原則」視窗。
6. 執行 "準備 IBM WebSphere MQ" 精靈。
- 如需執行「[準備 IBM WebSphere MQ](#)」精靈的進一步資訊；請參閱 [使用「準備 WebSphere MQ」精靈來配置 WebSphere MQ](#)。
- IBM IBM WebSphere MQ 安裝程式會自動執行 "準備 IBM WebSphere MQ"。
若要手動啟動精靈，請在 **啟動** > **所有程式** > **IBM WebSphere MQ** 資料夾中尋找 "準備 IBM WebSphere MQ" 的捷徑。選取對應於在多重安裝配置中安裝 IBM WebSphere MQ 的捷徑。
 - 按 **下一步** 並按一下 **是**，以回應 "識別網路中是否有 Windows 2000 或更新版本的網域控制站"問題。
 - 在第一個「為 Windows 網域使用者配置 IBM WebSphere MQ for Windows」視窗中，按 **是** > **下一步**。
 - 在第二個「為 Windows 網域使用者配置 IBM WebSphere MQ for Windows」視窗中，在 **網域** 欄位中鍵入 *wmq*。在 **使用者名稱** 欄位中鍵入 *wmquser1*，並在 **密碼** 欄位中鍵入密碼 (若有設定的話)。按 **下一步**。
精靈會使用 *wmquser1* 來配置並啟動 IBM IBM WebSphere MQ。

e) 在精靈的最後一頁中，視需要選取或清除勾選框，然後按一下 **完成**。

下一步

1. 執行作業 [第 316 頁的『讀取及寫入本端 mqm 群組授權的資料及日誌檔』](#)，以驗證安裝與配置是否正確運作。
2. 執行作業 [第 299 頁的『建立佇列管理程式資料及日誌檔的共用目錄』](#)，以配置檔案共用來儲存多重實例佇列管理程式的資料及日誌檔。

相關概念

WebSphere MQ Windows 服務所需的使用者權限

建立佇列管理程式資料及日誌檔的共用目錄

此作業是一組說明存取佇列管理程式資料及日誌檔的相關作業之一。這些作業顯示如何建立佇列管理程式，以授權讀取及寫入儲存在您選擇的目錄中的資料和日誌檔。

在正式作業規模配置中，您可能需要根據現有網域來修改配置。例如，您可以定義不同的網域群組來授權不同的共用，以及將執行佇列管理程式的使用者 ID 分組。

範例配置由三部伺服器組成：

sun

Windows Server 2008 網域控制站。它擁有包含 *Sun*、*mars* 及 *venus* 的 *wmq.example.com* 網域。為了說明起見，它也用作檔案伺服器。

mars

用作第一個 IBM WebSphere MQ 伺服器的 Windows Server 2008。它包含一個多重實例佇列管理程式實例，稱為 *QMGR*。

venus

用作第二部 IBM WebSphere MQ 伺服器的 Windows Server 2008。它包含稱為 *QMGR* 之多重實例佇列管理程式的第二個實例。

將範例中的斜體名稱取代為您選擇的名稱。

開始之前

1. 若要完全按照記載的方式執行此作業，請執行作業 [第 294 頁的『建立 IBM WebSphere MQ 的 Active Directory 及 DNS 網域』](#) 中的步驟，以 *sun.wmq.example.com* 在網域控制站上 *sun* 建立網域。變更斜體名稱以符合您的配置。

關於這項作業

此作業是一組說明存取佇列管理程式資料及日誌檔的相關作業之一。這些作業顯示如何建立佇列管理程式，以授權讀取及寫入儲存在您選擇的目錄中的資料和日誌檔。它們隨附作業 [第 290 頁的『Windows 網域及多重實例佇列管理程式』](#)。

在作業中，您可以建立包含資料及日誌目錄的共用，以及授權存取共用的廣域群組。您可以將授權共用的廣域群組名稱傳遞給其 *-a* 參數中的 *crtmqm* 指令。廣域群組可讓您彈性區隔此共用的使用者與其他共用的使用者。如果您不需要此彈性，請授權與 Domain *mqm* 群組共用，而不是建立新的廣域群組。

用於在此作業中共用的廣域群組稱為 *wmqha*，而共用稱為 *wmq*。它們定義在 Windows 網域 *wmq.example.com* 中的網域控制站 *sun* 上。共用具有廣域群組 *wmqha* 的完整控制許可權。將作業中的斜體名稱取代為您選擇的名稱。

基於此作業的目的，網域控制站與檔案伺服器是相同的伺服器。在實際應用程式中，請在不同伺服器之間分割目錄和檔案服務，以取得效能和可用性。

您必須將執行佇列管理程式的使用者 ID 配置成兩個群組的成員。它必須是 IBM WebSphere MQ 伺服器上本端 *mqm* 群組的成員，以及 *wmqha* 廣域群組的成員。

在這組作業中，當佇列管理程式以服務方式執行時，它會以使用者 ID *wmquser1* 執行，因此 *wmquser1* 必須是 *wmqha* 的成員。當佇列管理程式以互動方式執行時，它會以使用者 ID *wmquser2* 來執行，因此 *wmquser2* 必須是 *wmqha* 的成員。*wmquser1* 和 *wmquser2* 都是廣域群組 Domain *mqm* 的成員。Domain

mqm 是 *mars* 和 *venus* IBM WebSphere MQ 伺服器上本端 mqm 群組的成員。因此，*wmquser1* 和 *wmquser2* 都是這兩部 IBM WebSphere MQ 伺服器上本端 mqm 群組的成員。

程序

1. 以網域管理者身分登入網域控制站 *sun.wmq.example.com*。
2. 建立廣域群組 *wmqha*。
 - a) 開啟 **伺服器管理程式 > 角色 > Active Directory 網域服務 > *wmq.example.com* > 使用者**。
 - b) 開啟 *wmq.example.com\Users* 資料夾
 - c) 用滑鼠右鍵按一下 **使用者 > 新建 > 群組**。
 - d) 在 **群組名稱** 欄位中鍵入 *wmqha*。
 - e) 保持按一下 **廣域** 作為 **群組範圍**，並按一下 **安全** 作為 **群組類型**。按一下 **確定**。
3. 將網域使用者 *wmquser1* 及 *wmquser2* 新增至廣域群組 *wmqha*。
 - a) 在「伺服器管理程式」導覽樹狀結構中，按一下 **使用者**，然後在使用者清單中，用滑鼠右鍵按一下 ***wmqha* > 內容**。
 - b) 按一下「*wmqha* 內容」視窗中的「成員」標籤。
 - c) 按一下 **新增 ...; *wmquser1*; *wmquser2***，然後按一下 **檢查名稱 > 確定 > 套用 > 確定**。
4. 建立目錄樹狀結構以包含佇列管理程式資料及日誌檔。
 - a) 開啟一個命令提示字元。
 - b) 鍵入指令：

```
md c:\wmq\data , c:\wmq\logs
```
5. 授權廣域群組 *wmqha* 具有 *c:\wmq* 目錄及共用的完整控制權。
 - a) 在 Windows Explorer 中，用滑鼠右鍵按一下 **c: \wmq > 內容**。
 - b) 按一下 **安全** 標籤，然後按一下 **進階 > 編輯 ...**。
 - c) 清除 **包括來自此物件擁有者的可繼承權限** 的勾選框。在「Windows 安全」視窗中按一下 **複製**。
 - d) 在 **許可權項目** 清單中選取「使用者」的行，然後按一下 **移除**。在 **許可權項目** 清單中保留 SYSTEM、Administrators 及 CREATOR OWNER 的行。
 - e) 按一下 **新增 ...**，並鍵入廣域群組 *wmqha* 的名稱。按一下 **檢查名稱 > 確定**。
 - f) 在「*wmq* 的許可權項目」視窗中，選取 **許可權** 清單中的 **完整控制**。
 - g) 按一下 **確定 > 套用 > 確定 > 確定 > 確定**
 - h) 在 Windows Explorer 中，用滑鼠右鍵按一下 **c: \wmq > 共用 ...**
 - i) 按一下 **進階共用 ...** 並選取 **共用此資料夾** 勾選框。將共用名稱保留為 *wmq*。
 - j) 按一下 **許可權 > 新增 ...**，並鍵入廣域群組 *wmqha* 的名稱。按一下 **檢查名稱 > 確定**。
 - k) 在 **群組或使用者名稱** 清單中選取 *wmqha*。選取 ***wmqha*** 的許可權清單中的 **完整控制** 勾選框; 按一下 **套用**。
 - l) 在 **群組或使用者名稱** 清單中選取 *Administrators*。選取 **管理者的許可權** 清單中的 **完整控制** 勾選框; 按一下 **套用 > 確定 > 確定 > 關閉**。

下一步

確認您可以從每一個 IBM WebSphere MQ 伺服器讀取及寫入檔案至共用目錄。檢查 IBM WebSphere MQ 服務使用者 ID *wmquser1* 及互動式使用者 ID *wmquser2*。

1. 如果您是使用遠端桌面，則必須將 *wmq\wmquser1* 及 *wmquser2* 新增至 *mars* 上的本端群組 Remote Desktop Users。
 - a. 以 *wmq\Administrator* 身分登入 *mars*
 - b. 執行 **lusrmgr.msc** 指令，以開啟「本端使用者和群組」視窗。

- c. 按一下群組。用滑鼠右鍵按一下 遠端桌面使用者 > 內容 > 新增...。鍵入 *wmuser1*; *wmuser2* , 然後按一下 檢查名稱。
 - d. 鍵入網域管理者 *wmq\Administrator* 的使用者名稱及密碼, 然後按一下 確定 > 套用 > 確定。
 - e. 關閉「本端使用者和群組」視窗。
2. 以 *wmq\wmuser1* 身分登入 *mars* 。
 - a. 開啟「Windows 檔案總管」視窗, 並鍵入 `\\sun\wmq`。
系統會開啟 *sun.wmq.example.com* 上的 *wmq* 共用來回應, 並列出資料和日誌目錄。
 - b. 檢查 *wmuser1* 的許可權, 方法是在 *data* 子目錄中建立檔案, 新增部分內容, 讀取它, 然後刪除它。
 3. 以 *wmq\wmuser2* 身分登入 *mars* , 然後重複檢查。
 4. 執行下一項作業, 以建立佇列管理程式來使用共用資料及日誌目錄; 請參閱 第 301 頁的『讀取及寫入由替代廣域安全群組授權的共用資料及日誌檔』。

讀取及寫入由替代廣域安全群組授權的共用資料及日誌檔

此作業顯示如何在 `crtmqm` 指令上使用 `-a` 旗標。 `-a` 旗標可讓佇列管理程式使用替代安全群組來存取其在遠端檔案共用上的日誌及資料檔案。

在正式作業規模配置中, 您可能需要根據現有網域來修改配置。例如, 您可以定義不同的網域群組來授權不同的共用, 以及將執行佇列管理程式的使用者 ID 分組。

範例配置由三部伺服器組成:

sun

Windows Server 2008 網域控制站。它擁有包含 *Sun*、*mars* 及 *venus* 的 *wmq.example.com* 網域。為了說明起見, 它也用作檔案伺服器。

mars

用作第一個 IBM WebSphere MQ 伺服器的 Windows Server 2008。它包含一個多重實例佇列管理程式實例, 稱為 *QMGR*。

venus

用作第二部 IBM WebSphere MQ 伺服器的 Windows Server 2008。它包含稱為 *QMGR* 之多重實例佇列管理程式的第二個實例。

將範例中的斜體名稱取代為您選擇的名稱。

開始之前

執行下列作業中的步驟。這些作業會建立網域控制站及網域, 在一部伺服器上安裝 IBM WebSphere MQ for Windows, 並建立資料及日誌檔的檔案共用。如果您要配置現有的網域控制站, 可能會發現在新的 Windows Server 2008 上試用這些步驟很有用。您可以根據您的網域調整步驟。

1. 第 294 頁的『建立 IBM WebSphere MQ 的 Active Directory 及 DNS 網域』。
2. 第 296 頁的『在 Windows 網域中的伺服器或工作站上安裝 IBM WebSphere MQ』。
3. 第 299 頁的『建立佇列管理程式資料及日誌檔的共用目錄』。

關於這項作業

此作業是一組說明存取佇列管理程式資料及日誌檔的相關作業之一。這些作業顯示如何建立佇列管理程式, 以授權讀取及寫入儲存在您選擇的目錄中的資料和日誌檔。它們隨附作業 第 290 頁的『Windows 網域及多重實例佇列管理程式』。

在此作業中, 您可以建立佇列管理程式, 以將其資料及日誌儲存在檔案伺服器上的遠端目錄中。基於此範例的目的, 檔案伺服器與網域控制站是相同的伺服器。包含資料和日誌資料夾的目錄會以提供給廣域群組 *wmqha* 的完整控制許可權來共用。

程序

1. 以本端管理者 *mars\Administrator* 身分登入網域伺服器 *mars*。

- 開啟指令視窗。
- 重新啟動 IBM WebSphere MQ 服務。

您必須重新啟動服務，讓它在其下執行的使用者 ID 獲得您為它配置的其他安全認證。

鍵入指令：

```
endmqsvc  
strmqsvc
```

系統回應：

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
The MQ service for installation 'Installation1' ended successfully.  
及：  
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
The MQ service for installation 'Installation1' started successfully.
```

- 建立佇列管理程式。

```
crtmqm -a wmq\wmqha -sax -u SYSTEM.DEAD.LETTER.QUEUE -md \\sun\wmq\data -ld \\sun\wmq\logs  
QMGR
```

您必須 *wmqha* 透過指定廣域群組 "*wmq\wmqha*" 的完整網域名稱，來指定替代安全群組的網域 *wmq*。

您必須拼出共用 `\\sun\wmq` 的「一般命名慣例 (UNC)」名稱，且不要使用對映磁碟機參照。

系統回應：

```
WebSphere MQ queue manager created.  
Directory '\\sun\wmq\data\QMGR' created.  
The queue manager is associated with installation '1'  
Creating or replacing default objects for queue manager 'QMGR'  
Default objects statistics : 74 created. 0 replaced.  
Completing setup.  
Setup completed.
```

下一步

將訊息放入佇列並取得訊息，以測試佇列管理程式。

- 啟動佇列管理程式。

```
strmqm QMGR
```

系統回應：

```
WebSphere MQ queue manager 'QMGR' starting.  
The queue manager is associated with installation '1'.  
5 log records accessed on queue manager 'QMGR' during the log  
replay phase.  
Log replay for queue manager 'QMGR' complete.  
Transaction manager state recovered for queue manager 'QMGR'.  
WebSphere MQ queue manager 'QMGR' started using V7.1.0.0.
```

- 建立測試佇列。

```
echo define qlocal(QTEST) | runmqsc QMGR
```

系統回應：

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
Starting MQSC for queue manager QMGR.
```

```
1 : define qlocal(QTEST)
AMQ8006: WebSphere MQ queue created.
One MQSC command read.
No commands have a syntax error.
All valid MQSC commands were processed.
```

3. 使用範例程式 **amqsput** 來放置測試訊息。

```
echo 'A test message' | amqsput QTEST QMGR
```

系統回應:

```
Sample AMQSPUT0 start
target queue is QTEST
Sample AMQSPUT0 end
```

4. 使用範例程式 **amqsget** 取得測試訊息。

```
amqsget QTEST QMGR
```

系統回應:

```
Sample AMQSGET0 start
message <A test message>
Wait 15 seconds ...
no more messages
Sample AMQSGET0 end
```

5. 停止佇列管理程式。

```
endmqm -i QMGR
```

系統回應:

```
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ended.
```

6. 刪除佇列管理程式。

```
dltmqm QMGR
```

系統回應:

```
WebSphere MQ queue manager 'QMGR' deleted.
```

7. 刪除您建立的目錄。

提示: 將 /Q 選項新增至指令，以防止指令提示刪除每一個檔案或目錄。

```
del /F /S C:\wmq\*.*
rmdir /S C:\wmq
```

在網域控制站上建立多重實例佇列管理程式

範例顯示如何在網域控制站上的 Windows 上設定多重實例佇列管理程式。設定會示範所涉及的概念，而不是正式作業規模。此範例以 Windows Server 2008 為基礎。步驟在其他版本的 Windows Server 上可能有所不同。

配置使用迷你網域或 "domainlet" 的概念; 請參閱 Windows 2000、Windows Server 2003 及 Windows Server 2008 叢集節點作為網域控制站。若要將多重實例佇列管理程式新增至現有網域，請參閱 [第 291 頁的『在網域工作站或伺服器上建立多重實例佇列管理程式』](#)。

範例配置由三部伺服器組成:

sun

用作第一個網域控制站的 Windows Server 2008 伺服器。它定義包含 *sun*、*earth* 及 *mars* 的 *wmq.example.com* 網域。它包含一個多重實例佇列管理程式實例，稱為 *QMGR*。

earth

用作第二個網域控制站 IBM WebSphere MQ 伺服器的 Windows Server 2008。它包含稱為 *QMGR* 之多重實例佇列管理程式的第二個實例。

mars

用作檔案伺服器的 Windows Server 2008。

將範例中的斜體名稱取代為您選擇的名稱。

開始之前

1. 在 Windows 上，您不需要驗證計劃儲存佇列管理程式資料及日誌檔的檔案系統。檢查程序 [驗證共用檔案系統行為](#) 適用於 UNIX and Linux。在 Windows 上，檢查一律成功。
2. 執行第 294 頁的『[建立 IBM WebSphere MQ 的 Active Directory 及 DNS 網域](#)』中的步驟，以建立第一個網域控制站。
3. 執行第 306 頁的『[將第二個網域控制站新增至 wmq.example.com 網域](#)』中的步驟以新增第二個網域控制站，在兩個網域控制站上安裝 IBM WebSphere MQ for Windows，並驗證安裝。
4. 執行第 308 頁的『[在 wmq.example.com 網域中的網域控制站上安裝 IBM WebSphere MQ](#)』中的步驟，以在兩個網域控制站上安裝 IBM WebSphere MQ。

關於這項作業

在相同網域中的檔案伺服器上，建立佇列管理程式日誌及資料目錄的共用。接下來，在其中一個網域控制站上建立使用檔案共用的多重實例佇列管理程式的第一個實例。在另一個網域控制站上建立另一個實例，最後驗證配置。您可以在網域控制站上建立檔案共用。

在範例中，*sun* 是第一個網域控制站，*earth* 是第二個網域控制站，*mars* 是檔案伺服器。

程序

1. 建立要包含佇列管理程式資料及日誌檔的目錄。

a) 在 *mars* 上，鍵入下列指令：

```
md c:\wmq\data , c:\wmq\logs
```

2. 共用要包含佇列管理程式資料及日誌檔的目錄。

您必須允許完整控制存取網域本端群組 *mqm*，以及您用來建立佇列管理程式的使用者 ID。在此範例中，屬於 Domain Administrators 成員的使用者 ID 具有建立佇列管理程式的權限。

檔案共用必須位於與網域控制站位於相同網域的伺服器上。在此範例中，伺服器 *mars* 與網域控制站位於相同網域中。

- a) 在 Windows Explorer 中，用滑鼠右鍵按一下 **c: \wmq > 內容**。
- b) 按一下 **安全** 標籤，然後按一下 **進階 > 編輯 ...**。
- c) 清除 **包括來自此物件擁有者的可繼承權限** 的勾選框。在「Windows 安全」視窗中按一下 **複製**。
- d) 在 **許可權項目** 清單中選取「使用者」的行，然後按一下 **移除**。在 **許可權項目** 清單中保留 SYSTEM、Administrators 及 CREATOR OWNER 的行。
- e) 按一下 **新增 ...**，並鍵入網域本端群組 *mqm* 的名稱。按一下 **檢查名稱**。
- f) 為了回應「Windows 安全」視窗，請輸入 Domain Administrator 的名稱和密碼，然後按一下 **確定 > 確定**。
- g) 在「wmq 的許可權項目」視窗中，選取 **許可權** 清單中的 **完整控制**。
- h) 按一下 **確定 > 套用 > 確定 > 確定 > 確定**。
- i) 重複步驟 e 至 h，以新增 Domain Administrators。
- j) 在 Windows Explorer 中，用滑鼠右鍵按一下 **c: \wmq > 共用 ...**。
- k) 按一下 **進階共用 ...** 並選取 **共用此資料夾** 勾選框。將共用名稱保留為 *wmq*。

- l) 按一下 **許可權 > 新增 ...**，並鍵入網域本端群組 *mqm*; Domain Administrators 的名稱。按一下 **檢查名稱**。
 - m) 為了回應「Windows 安全」視窗，請輸入 Domain Administrator 的名稱和密碼，然後按一下 **確定 > 確定**。
3. 在第一個網域控制站上建立佇列管理程式 *QMGR* (*sun*)。

```
crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE -md \\mars\wmq\data -ld \\mars\wmq\logs QMGR
```

系統回應：

```
WebSphere MQ queue manager created.
Directory '\\mars\wmq\data\QMGR' created.
The queue manager is associated with installation 'Installation1'.
Creating or replacing default objects for queue manager 'QMGR'.
Default objects statistics : 74 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
```

4. 在 *sun* 上啟動佇列管理程式，允許待命實例。

```
strmqm -x QMGR
```

系統回應：

```
WebSphere MQ queue manager 'QMGR' starting.
The queue manager is associated with installation 'Installation1'.
5 log records accessed on queue manager 'QMGR' during the log
replay phase.
Log replay for queue manager 'QMGR' complete.
Transaction manager state recovered for queue manager 'QMGR'.
WebSphere MQ queue manager 'QMGR' started using V7.1.0.0.
```

5. 在 *earth* 上建立第二個 *QMGR* 實例。
- a) 檢查 *earth* 的字首和 `InstallationName` 參數值是否正確。

在 *sun* 上，執行 **dspmqinf** 指令：

```
dspmqinf QMGR
```

系統回應：

```
QueueManager:
  Name=QMGR
  Directory=QMGR
  Prefix=C:\Program Files\IBM\WebSphere MQ
  DataPath=\\mars\wmq\data\QMGR
  InstallationName=Installation1
```

- b) 將 **QueueManager** 段落的機器可讀形式複製到剪貼簿。
- 在 *sun* 上，使用 `-o command` 參數再次執行 **dspmqinf** 指令。

```
dspmqinf -o command QMGR
```

系統回應：

```
addmqinf -s QueueManager -v Name=QMGR
-v Directory=QMGR -v Prefix="C:\Program Files\IBM\WebSphere MQ"
-v DataPath=\\mars\wmq\data\QMGR
```

- c) 在 *earth* 上，從剪貼簿執行 **addmqinf** 指令，以在 *earth* 上建立佇列管理程式的實例。

必要的話，請調整指令，以容納 Prefix 或 InstallationName 參數中的差異。

```
addmqinf -s QueueManager -v Name=QMGR
-v Directory=QMGR -v Prefix="C:\Program Files\IBM\WebSphere MQ"
-v DataPath=\\mars\wmq\data\QMGR
```

WebSphere MQ configuration information added.

6. 在 *earth* 上啟動佇列管理程式的待命實例。

```
strmqm -x QMGR
```

系統回應：

```
WebSphere MQ queue manager 'QMGR' starting.
The queue manager is associated with installation 'Installation1'.
A standby instance of queue manager 'QMGR' has been started. The active
instance is running elsewhere.
```

結果

驗證佇列管理程式是否從 *sun* 切換至 *earth*：

1. 在 *sun* 上，執行下列指令：

```
endmqm -i -r -s QMGR
```

sun 上的系統回應：

```
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ended, permitting switchover to
a standby instance.
```

2. 在 *earth* 上反覆地鍵入指令：

```
dspmq
```

系統回應：

```
QMNAME(QMGR) STATUS(Running as standby)
QMNAME(QMGR) STATUS(Running as standby)
QMNAME(QMGR) STATUS(Running)
```

下一步

若要用範例程式來驗證多重實例佇列管理程式，請參閱第 310 頁的『[在 Windows 上驗證多重實例佇列管理程式](#)』。

相關工作

第 306 頁的『[將第二個網域控制站新增至 *wmq.example.com* 網域](#)』

第 308 頁的『[在 *wmq.example.com* 網域中的網域控制站上安裝 IBM WebSphere MQ](#)』

相關資訊

[Windows 2000、Windows Server 2003 及 Windows Server 2008 叢集節點作為網域控制站](#)

將第二個網域控制站新增至 *wmq.example.com* 網域

將第二個網域控制站新增至 *wmq.example.com* 網域，以建構 Windows 網域，在其中執行網域控制站及檔案伺服器上的多重實例佇列管理程式。

範例配置由三部伺服器組成：

sun

用作第一個網域控制站的 Windows Server 2008 伺服器。它定義包含 *sun*、*earth* 及 *mars* 的 *wmq.example.com* 網域。它包含一個多重實例佇列管理程式實例，稱為 *QMGR*。

earth

用作第二個網域控制站 IBM WebSphere MQ 伺服器的 Windows Server 2008。它包含稱為 *QMGR* 之多重實例佇列管理程式的第二個實例。

mars

用作檔案伺服器的 Windows Server 2008。

將範例中的斜體名稱取代為您選擇的名稱。

開始之前

1. 執行第 294 頁的『建立 IBM WebSphere MQ 的 Active Directory 及 DNS 網域』中的步驟，以建立網域 *wmq.example.com* 的網域控制站 *sun*。變更斜體名稱以符合您的配置。
2. 在預設工作群組 WORKGROUP 中的伺服器上安裝 Windows Server 2008。例如，伺服器名稱為 *earth*。

關於這項作業

在這項作業中，您將 Windows Server 2008 (稱為 *earth*) 配置成 *wmq.example.com* 網域中的第二個網域控制站。

此作業是一組說明存取佇列管理程式資料及日誌檔的相關作業之一。這些作業顯示如何建立佇列管理程式，以授權讀取及寫入儲存在您選擇的目錄中的資料和日誌檔。它們隨附作業第 290 頁的『Windows 網域及多重實例佇列管理程式』。

程序

1. 將網域控制站 *sun.wmq.example.com* 新增至 *earth* 作為 DNS 伺服器。
 - a) 在 *earth* 上，以 *earth\Administrator* 身分登入，然後按一下 **開始**。
 - b) 用滑鼠右鍵按一下 **網路 > 內容 > 管理網路連線**。
 - c) 用滑鼠右鍵按一下網路配接卡，然後按一下 **內容**。
系統會回應「區域連線內容」視窗，列出連線使用的項目。
 - d) 從「本端區域連線內容」視窗的項目清單中，選取 **Internet Protocol 第 4 版** 或 **Internet Protocol 第 6 版**。按一下 **內容 > 進階 ...** 然後按一下 **DNS** 標籤。
 - e) 在 DNS 伺服器位址下，按一下 **新增 ...**。
 - f) 鍵入網域控制站 (也是 DNS 伺服器) 的 IP 位址，然後按一下 **新增**。
 - g) 按一下 **附加這些 DNS 字尾 > 新增 ...**。
 - h) 鍵入 *wmq.example.com*，然後按一下 **新增**。
 - i) 在 **此連線的 DNS 字尾** 欄位中鍵入 *wmq.example.com*。
 - j) 選取在 **DNS 中登錄此連線的位址** 和在 **DNS 登錄中使用此連線的字尾**。按一下 **確定 > 確定 > 關閉**。
 - k) 開啟指令視窗，並鍵入指令 **ipconfig /all**，以檢閱 TCP/IP 設定。
2. 以本端或 Workgroup 管理者身分登入網域控制站 *sun*。
如果伺服器已配置為網域控制站，您必須以網域管理者身分登入。
3. 執行 Active Directory 網域服務精靈。
 - a) 按一下 **開始 > 執行 ...** 鍵入 *dcpromo*，然後按一下 **確定**。
如果尚未安裝 Active Directory 二進位檔，Windows 會自動安裝這些檔案。
4. 將 *earth* 配置為 *wmq.example.com* 網域中的第二個網域控制站。
 - a) 在精靈的第一個視窗中，保持不勾選 **使用進階模式安裝** 勾選框。按一下 **下一步 > 下一步**，然後按一下 **建立將網域控制站新增至現有網域 > 下一步**。

- b) 在 **鍵入此樹系中任何網域的名稱 ...** 中鍵入 *wmq* 欄位。按一下 **替代認證** 圓鈕，然後按一下 **設定 ...**。鍵入網域管理者的名稱及密碼，然後按 **確定 > 下一步 > 下一步 > 下一步**。
- c) 在「其他網域控制站選項」視窗中，接受已選取的 **DNS 伺服器** 及 **廣域型錄** 選項；按 **下一步 > 下一步**。
- d) 在「目錄服務還原模式管理者密碼」上，鍵入 **密碼** 及 **確認密碼**，然後按 **下一步 > 下一步**。
- e) 當系統提示輸入 **Network Credentials** 時，請鍵入網域管理者的密碼。在最終精靈視窗中選取 **完成時重新開機**。
- f) 在一段時間之後，可能會開啟一個視窗，其中含有關於 DNS 委派的 **DCPromo** 錯誤；請按一下 **確定**。伺服器會重新開機。

結果

當 *earth* 重新開機時，以網域管理者身分登入。檢查 *wmq.example.com* 網域是否已抄寫至 *earth*。

下一步

繼續安裝 IBM WebSphere MQ；請參閱 [第 308 頁的『在 *wmq.example.com* 網域中的網域控制站上安裝 IBM WebSphere MQ』](#)。

相關工作

[第 294 頁的『建立 IBM WebSphere MQ 的 Active Directory 及 DNS 網域』](#)

在 *wmq.example.com* 網域中的網域控制站上安裝 *IBM WebSphere MQ*

在 *wmq.example.com* 網域中的兩個網域控制站上安裝及配置 IBM WebSphere MQ。

此處放置簡要說明；用於第一段及摘要。

範例配置由三部伺服器組成：

sun

用作第一個網域控制站的 Windows Server 2008 伺服器。它定義包含 *sun*、*earth* 及 *mars* 的 *wmq.example.com* 網域。它包含一個多重實例佇列管理程式實例，稱為 *QMGR*。

earth

用作第二個網域控制站 IBM WebSphere MQ 伺服器的 Windows Server 2008。它包含稱為 *QMGR* 之多重實例佇列管理程式的第二個實例。

mars

用作檔案伺服器的 Windows Server 2008。

將範例中的斜體名稱取代為您選擇的名稱。

開始之前

1. 執行 [第 294 頁的『建立 IBM WebSphere MQ 的 Active Directory 及 DNS 網域』](#) 中的步驟，以建立網域 *wmq.example.com* 的網域控制站 *sun*。變更斜體名稱以符合您的配置。
2. 執行 [第 306 頁的『將第二個網域控制站新增至 *wmq.example.com* 網域』](#) 中的步驟，為網域 *wmq.example.com* 建立第二個網域控制站 *earth*。變更斜體名稱以符合您的配置。
3. 如需您可以執行 IBM WebSphere MQ 的其他 Windows 版本，請參閱 [Windows 系統上的軟硬體需求](#)。

關於這項作業

在 *wmq.example.com* 網域中的兩個網域控制站上安裝及配置 IBM WebSphere MQ。

程序

1. 在 *sun* 和 *earth* 上安裝 IBM WebSphere MQ。

如需執行 IBM WebSphere MQ for Windows 安裝精靈的進一步資訊；請參閱 [在 Windows 上安裝 IBM WebSphere MQ 伺服器](#)。

a) 在 *sun* 及 *earth* 上，以網域管理者 *wmq\Administrator* 身分登入。

b) 在 IBM WebSphere MQ for Windows 安裝媒體上執行 **Setup** 指令。

即會啟動「IBM WebSphere MQ 啟動程式」應用程式。

c) 按一下 **軟體需求**，以檢查是否已安裝必備軟體。

d) 按一下 **網路配置** > 否。

您可以配置網域使用者 ID 或不配置此安裝。所建立的使用者 ID 是網域本端使用者 ID。

e) 按一下 **WebSphere MQ 安裝**，選取安裝語言，然後按一下「啟動 IBM WebSphere MQ 安裝程式」。

f) 確認授權合約，然後按 **下一步** > **下一步** > **安裝**，以接受預設配置。等待安裝完成，然後按一下 **完成**。

如果您想要變更安裝名稱、安裝不同的元件、配置不同的佇列管理程式資料和日誌目錄，或安裝至不同的目錄，請按一下 **自訂**，而不是按一下 **一般**。

已安裝 IBM WebSphere MQ，且安裝程式會啟動 "準備 IBM WebSphere MQ" 精靈。

IBM WebSphere MQ for Windows 安裝會配置網域本端群組 *mqm* 及網域群組 *Domain mqm*。它會使 *Domain mqm* 成為 *mqm* 的成員。相同網域中的後續網域控制站會共用 *mqm* 及 *Domain mqm* 群組。

2. 在 *earth* 和 *sun* 兩者上，執行 "準備 IBM WebSphere MQ" 精靈。

如需執行「"準備 IBM WebSphere MQ"」精靈的進一步資訊，請參閱 [使用「準備 WebSphere MQ」精靈來配置 WebSphere MQ](#)。

a) IBM WebSphere MQ 安裝程式會自動執行 "準備 IBM WebSphere MQ"。

若要手動啟動精靈，請在 **啟動** > **所有程式** > **IBM WebSphere MQ** 資料夾中尋找 "準備 IBM WebSphere MQ" 的捷徑。選取對應於在多重安裝配置中安裝 IBM WebSphere MQ 的捷徑。

b) 按 **下一步**，並保持按一下 **否**，以回應問題 "識別網路中是否有 Windows 2000 或更新版本的網域控制站"¹。

c) 在精靈的最後一頁中，視需要選取或清除勾選框，然後按一下 **完成**。

"準備 IBM WebSphere MQ" 精靈會在第一個網域控制站上建立網域本端使用者 *MUSR_MQADMIN*，在第二個網域控制站上建立另一個網域本端使用者 *MUSR_MQADMIN1*。精靈會在每一個控制器上建立 IBM WebSphere MQ 服務，並以 *MUSR_MQADMIN* 或 *MUSR_MQADMIN1* 作為登入服務的使用者。

3. 定義有權建立佇列管理程式的使用者。

使用者必須有權在本端登入，並且是網域本端 *mqm* 群組的成員。在網域控制站上，網域使用者無權在本端登入，但管理者有權登入。依預設，沒有使用者同時具有這些屬性。在此作業中，將網域管理者新增至網域本端 *mqm* 群組。

a) 開啟 **伺服器管理程式** > **角色** > **Active Directory 網域服務** > *wmq.example.com* > **使用者**。

b) 用滑鼠右鍵按一下 **網域管理者** > **新增至群組 ...** 並鍵入 *mqm*; 按一下 **檢查名稱** > **確定** > **確定**

結果

1. 檢查 "準備 IBM WebSphere MQ" 是否已建立網域使用者 *MUSR_MQADMIN*:

a. 開啟 **伺服器管理程式** > **角色** > **Active Directory 網域服務** > *wmq.example.com* > **使用者**。

b. 用滑鼠右鍵按一下 **MUSR_MQADMIN** > **內容 ...** > **成員屬於**，並查看它是否為 *Domain users* 及 *mqm* 的成員。

2. 檢查 *MUSR_MQADMIN* 是否有權以服務方式執行:

a. 按一下 **開始** > **執行 ...**，鍵入指令 **secpol.msc**，然後按一下 **確定**。

b. 開啟 **安全設定** > **本機原則** > **使用者權限指派**。在原則清單中，用滑鼠右鍵按一下 **以服務方式登入** > **內容**，且請參閱 *MUSR_MQADMIN* 列為具有以服務方式登入的權利。按一下 **確定**。

¹ 您可以配置網域的安裝。因為網域控制站上的所有使用者和群組都有網域範圍，所以不會有任何差異。安裝 IBM WebSphere MQ 更簡單，就像它不在網域中一樣。

下一步

1. 執行作業 第 316 頁的『讀取及寫入本端 mqm 群組授權的資料及日誌檔』，以驗證安裝與配置是否正確運作。
2. 回到作業 第 303 頁的『在網域控制站上建立多重實例佇列管理程式』，以完成在網域控制站上配置多重實例佇列管理程式的作業。

相關概念

WebSphere MQ Windows 服務所需的使用者權限

在 Windows 上驗證多重實例佇列管理程式

使用範例程式 **amqsgbac**、**amqspbac** 及 **amqsmbac** 來驗證多重實例佇列管理程式配置。本主題提供在 Windows Server 2003 上驗證多重實例佇列管理程式配置的範例配置。

高可用性範例程式使用自動用戶端重新連線。當連接的佇列管理程式失敗時，用戶端會嘗試重新連接至相同佇列管理程式群組中的佇列管理程式。範例的說明 (高可用性範例程式) 示範使用單一實例佇列管理程式的用戶端重新連線，以簡化。您可以將相同的範例與多重實例佇列管理程式搭配使用，以驗證多重實例佇列管理程式配置。

此範例使用 第 303 頁的『在網域控制站上建立多重實例佇列管理程式』中說明的多重實例配置。使用配置來驗證多重實例佇列管理程式是否切換至待命實例。使用 **endmqm** 指令停止佇列管理程式，並使用 **-s**，切換，選項。用戶端程式會重新連接至新的佇列管理程式實例，並在稍微延遲之後繼續使用新的實例。

用戶端安裝在執行 Windows XP Service Pack 2 的 400 MB VMware 映像檔中。基於安全理由，它與執行多重實例佇列管理程式的網域伺服器在相同的 VMware 僅限主機網路上連接。它正在共用包含用戶端連線表格的 /MQHA 資料夾，以簡化配置。

使用 WebSphere MQ 探險家驗證失效接手

在使用範例應用程式來驗證失效接手之前，請在每一部伺服器上執行「WebSphere MQ 探險家」。使用「**新增遠端佇列管理程式 > 直接連接多重實例佇列管理程式**」精靈，將這兩個佇列管理程式實例新增至每一個瀏覽器。確保兩個實例都在執行中，允許待命。關閉使用作用中實例執行 VMware 映像檔的視窗，虛擬地關閉伺服器電源，或停止作用中實例，容許切換至待命實例並可重新連接用戶端重新連接。

註: 如果您關閉伺服器電源，請確定它不是管理 MQHA 資料夾的伺服器!

註: 在「**停止佇列管理程式**」對話框上可能無法使用 **容許切換至待命實例** 選項。遺漏選項，因為佇列管理程式作為單一實例佇列管理程式執行。您必須已在沒有 **允許待命實例** 選項的情況下啟動它。如果您拒絕停止佇列管理程式的要求，請查看「**詳細資料**」視窗，可能沒有待命實例在執行中。

使用範例程式驗證失效接手

選擇伺服器以執行作用中實例

您可能已選擇其中一部伺服器來管理 MQHA 目錄或檔案系統。如果您計劃透過關閉執行作用中伺服器的 VMware 視窗來測試失效接手，請確定它不是管理 MQHA!

在執行作用中佇列管理程式實例的伺服器上

1. 修改 *ipaddr1* 及 *ipaddr2*，並將下列指令儲存在 N:\hasample.tst. 中

```
DEFINE QLOCAL(SOURCE) REPLACE
DEFINE QLOCAL(TARGET) REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
MCAUSER(' ') REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNAME('ipaddr1(1414),ipaddr2(1414)') QMNAME(QM1) REPLACE
START CHANNEL(CHANNEL1)
DEFINE LISTENER(LISTENER.TCP) TRPTYPE(TCP) CONTROL(QMGR)
DISPLAY LISTENER(LISTENER.TCP) CONTROL
DISPLAY LSSTATUS(LISTENER.TCP) STATUS
```

註: 將 **MCAUSER** 參數保留空白，會將用戶端使用者 ID 傳送至伺服器。用戶端使用者 ID 必須在伺服器上具有正確的許可權。替代方案是將 SVRCONN 通道中的 **MCAUSER** 參數設為您為您在伺服器上配置的使用者 ID。

2. 以路徑 N:\ 開啟命令提示字元，並執行下列指令：

```
runmqsc -m QM1 < hasample.tst
```

3. 透過檢查 **runmqsc** 指令的輸出，驗證接聽器是否在執行中且具有佇列管理程式控制。

```
LISTENER(LISTENER.TCP)CONTROL(QMGR)  
LISTENER(LISTENER.TCP)STATUS(RUNNING)
```

或者，使用 TCPIP 接聽器正在執行且具有 Control = Queue Manager 的「WebSphere MQ 探險家」。

在用戶端上

1. 將伺服器上的共用目錄 C:\MQHA 對映至用戶端上的 N:\ 。
2. 開啟具有路徑 N:\ 的命令提示字元。設定環境變數 MQCHLLIB，以指向伺服器上的用戶端通道定義表 (CCDT)：

```
SET MQCHLLIB=N:\data\QM1\@ipcc
```

3. 在命令提示字元中鍵入下列指令：

```
start amqsghac TARGET QM1  
start amqsmhac -s SOURCE -t TARGET -m QM1  
start amqsphac SOURCE QM1
```

註：如果您有問題，請在命令提示字元下啟動應用程式，以便在主控台上印出原因碼，或查看 N:\data\QM1\errors 資料夾中的 AMQERR01.LOG 檔案。

在執行作用中佇列管理程式實例的伺服器上

1. 您可以：
 - 關閉使用作用中伺服器實例執行 VMware 映像檔的視窗。
 - 使用「WebSphere MQ 探險家」，停止作用中佇列管理程式實例，容許切換至待命實例，並指示可重新連接的用戶端重新連接。
2. 這三個用戶端最終會偵測到連線已中斷，然後重新連接。在此配置中，如果您關閉伺服器視窗，則重建這三個連線大約需要 7 分鐘。有些連線在其他連線之前就已重新建立。

結果

```
N:\>amqsphac SOURCE QM1  
Sample AMQSPHAC start  
target queue is SOURCE  
message <Message 1>  
message <Message 2>  
message <Message 3>  
message <Message 4>  
message <Message 5>  
17:05:25 : EVENT : Connection Reconnecting (Delay: 0ms)  
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)  
17:05:52 : EVENT : Connection Reconnected  
message <Message 6>  
message <Message 7>  
message <Message 8>  
message <Message 9>
```

```
N:\>amqsmhac -s SOURCE -t TARGET -m QM1  
Sample AMQSMHA0 start  
  
17:05:25 : EVENT : Connection Reconnecting (Delay: 97ms)  
17:05:48 : EVENT : Connection Reconnecting (Delay: 0ms)  
17:05:53 : EVENT : Connection Reconnected
```

```

N:\>amqsgnac TARGET QM1
Sample AMQSGHAC start
message <Message 1>
message <Message 2>
message <Message 3>
message <Message 4>
message <Message 5>
17:05:25 : EVENT : Connection Reconnecting (Delay: 156ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message <Message 6>
message <Message 7>
message <Message 8>
message <Message 9>

```

在 Windows 上保護共用佇列管理程式資料及日誌目錄和檔案的安全

本主題說明如何使用廣域替代安全群組來保護佇列管理程式資料及日誌檔的共用位置。您可以在不同伺服器上執行之佇列管理程式的不同實例之間共用該位置。

通常您不會設定佇列管理程式資料及日誌檔的共用位置。當您安裝 IBM WebSphere MQ for Windows 時，安裝程式會為該伺服器上建立的任何佇列管理程式建立您選擇的起始目錄。它使用本端 `mqm` 群組來保護目錄，並配置 IBM WebSphere MQ 服務的使用者 ID 以存取目錄。

當您使用安全群組來保護共用資料夾時，允許存取資料夾的使用者必須具有該群組的認證。假設遠端檔案伺服器上的資料夾受到稱為 *mars* 的伺服器上本端 `mqm` 群組的保護。讓執行佇列管理程式的使用者處理 *mars* 上本端 `mqm` 群組的成員。使用者具有的認證符合遠端檔案伺服器上資料夾的認證。使用這些認證，佇列管理程式可以存取其資料夾中的資料及日誌檔。在不同伺服器上執行佇列管理程式處理程序的使用者是不同本端 `mqm` 群組的成員，該群組沒有相符的認證。當佇列管理程式在與 *mars* 不同的伺服器上執行時，它無法存取在 *mars* 上執行時所建立的資料及日誌檔。即使您將使用者設為網域使用者，它也會有不同的認證，因為它必須從 *mars* 上的本端 `mqm` 群組獲得認證，且無法從不同的伺服器執行該動作。

為佇列管理程式提供廣域替代安全群組可解決此問題；請參閱第 313 頁的圖 64。使用廣域群組來保護遠端資料夾的安全。當您在 *mars* 上建立廣域群組名稱時，請將它傳遞給佇列管理程式。在 `crtmqm` 指令上使用 `-a[r]` 參數，傳遞廣域群組名稱作為替代安全群組。如果您傳送佇列管理程式以在不同的伺服器上執行，則安全群組的名稱會隨它一起傳送。名稱會以 `SecurityGroup` 形式在 `qm.ini` 檔案的 **AccessMode** 段落中傳送；例如：

```

AccessMode:
  SecurityGroup=wmq\wmq

```

`qm.ini` 中的 **AccessMode** 段落也包含 `RemoveMQMAccess`；例如：

```

AccessMode:
  RemoveMQMAccess=<true\false>

```

如果使用值 `true` 指定此屬性，並且也提供了存取群組，則不會將佇列管理程式資料檔案的存取權授與本端 `mqm` 群組。

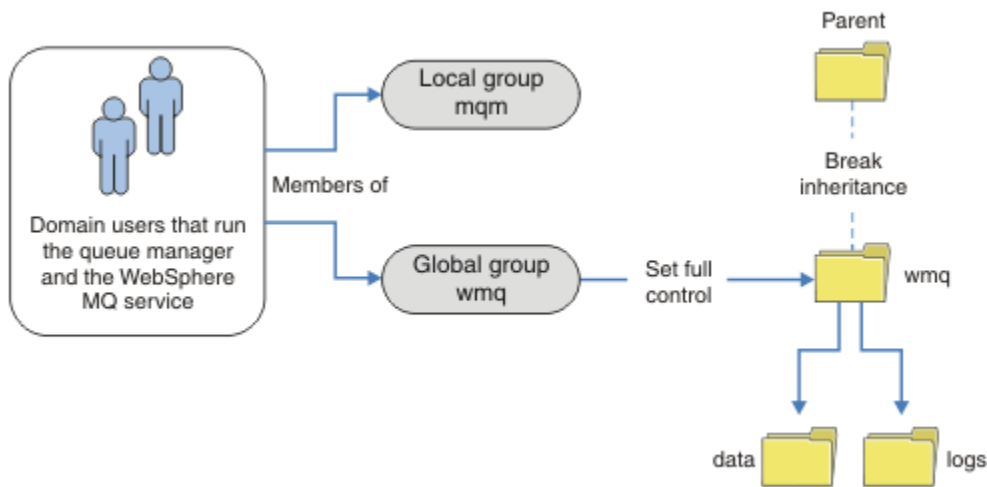


圖 64: 使用替代廣域安全群組來保護佇列管理程式資料及日誌的安全 (1)

對於用來執行佇列管理程式處理程序的使用者 ID，若要具有廣域安全群組的相符認證，該使用者 ID 也必須具有廣域範圍。您無法使本端群組或主體成為廣域群組的成員。在 [第 313 頁的圖 64](#) 中，執行佇列管理程式處理程序的使用者會顯示為網域使用者。

如果您要部署許多 IBM WebSphere MQ 伺服器，則 [第 313 頁的圖 64](#) 中的使用者分組並不方便。您需要針對每一部 IBM WebSphere MQ 伺服器，重複將使用者新增至本端群組的程序。請改為在網域控制站上建立 Domain mqm 廣域群組，並讓執行 IBM WebSphere MQ 群組成員的使用者成為 Domain mqm 群組的使用者；請參閱 [第 313 頁的圖 65](#)。當您將 IBM WebSphere MQ 安裝為網域安裝時，"準備 IBM WebSphere MQ" 精靈會自動使 Domain mqm 群組成為本端 mqm 群組的成員。相同的使用者同時在廣域群組 Domain mqm 和 wmq 中。

提示: 相同的使用者可以在不同的伺服器上執行 IBM WebSphere MQ，但在個別伺服器上，您必須具有不同的使用者，才能以服務方式執行 IBM WebSphere MQ，並以互動方式執行。對於伺服器上的每一個安裝，您也必須有不同的使用者。因此，Domain mqm 通常包含一些使用者。

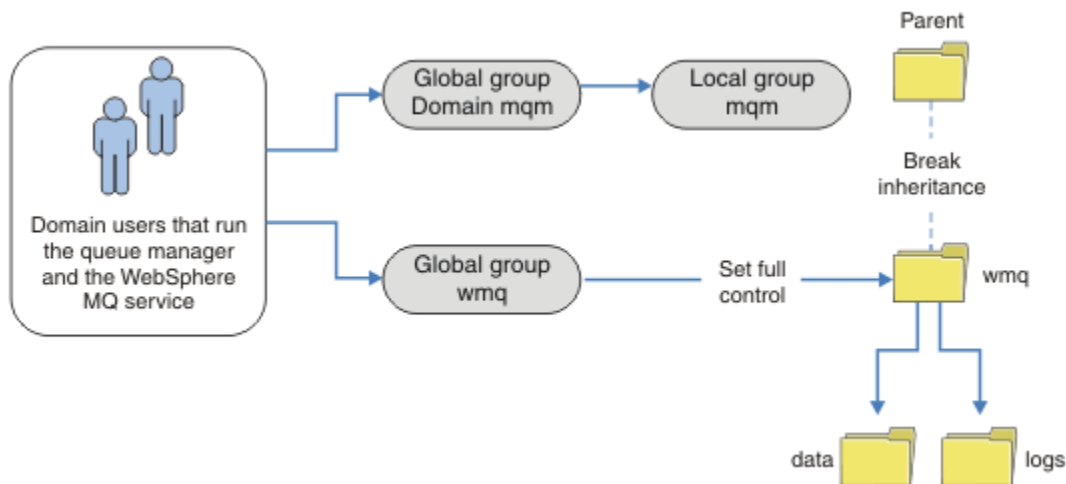


圖 65: 使用替代廣域安全群組來保護佇列管理程式資料及日誌的安全 (2)

[第 313 頁的圖 65](#) 中的組織是不必要的複雜。這一安排有兩個成員相同的全球集團。您可以簡化組織，並且只定義一個廣域群組；請參閱 [第 314 頁的圖 66](#)。

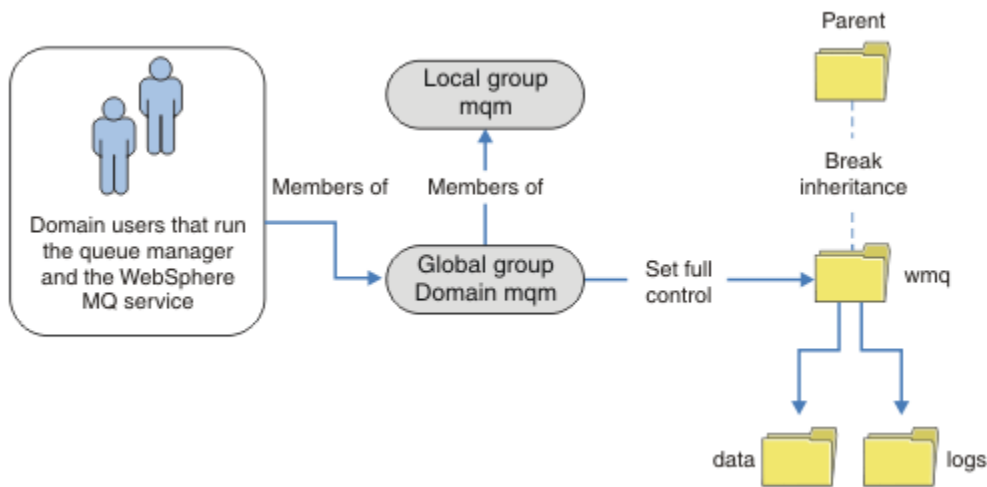


圖 66: 使用替代廣域安全群組來保護佇列管理程式資料及日誌的安全 (3)

或者，您可能需要更精細的存取控制，其中不同的佇列管理程式會限制為能夠存取不同的資料夾；請參閱第 315 頁的圖 67。在第 315 頁的圖 67 中，會在個別廣域群組中定義兩個網域使用者群組，以保護不同佇列管理程式日誌及資料檔的安全。會顯示兩個不同的本端 mqm 群組，它們必須位於不同的 IBM WebSphere MQ 伺服器上。在此範例中，佇列管理程式會分割成兩個集合，並將不同的使用者配置給這兩個集合。這兩組可能是測試和正式作業佇列管理程式。替代安全群組稱為 wmq1 和 wmq2。您必須根據廣域群組 wmq1 和 wmq2 是否在測試或正式作業部門中，手動將它們新增至正確的佇列管理程式。此配置無法利用 IBM WebSphere MQ 安裝像第 314 頁的圖 66 一樣將 Domain mqm 傳播至本端 mqm 群組，因為有兩個使用者群組。

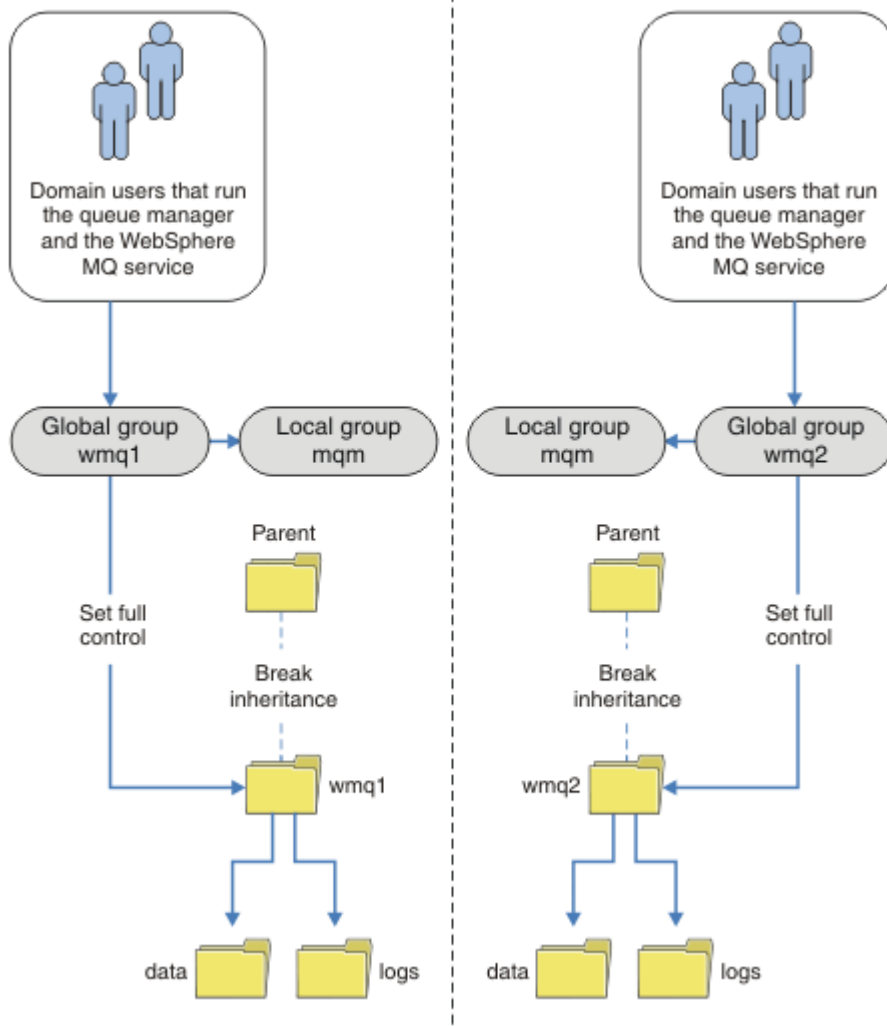


圖 67: 使用替代廣域安全主體來保護佇列管理程式資料及日誌的安全 (4)

分割兩個部門的另一種方式是將它們放在兩個視窗網域中。在該情況下，您可以回到使用 [第 314 頁的圖 66](#) 中顯示的較簡單模型。

在 Windows 上保護未共用佇列管理程式資料及日誌目錄和檔案的安全

本主題說明如何使用本端 mqm 群組及替代安全群組，來保護佇列管理程式資料及日誌檔的替代位置。

一般而言，您不會設定佇列管理程式資料及日誌檔的替代位置。當您安裝 IBM WebSphere MQ for Windows 時，安裝程式會為您所建立的任何佇列管理程式建立您選擇的起始目錄。它使用本端 mqm 群組來保護目錄，並配置 IBM WebSphere MQ 服務的使用者 ID 以存取目錄。

兩個範例示範如何配置 IBM WebSphere MQ 的存取控制。這些範例顯示如何使用佇列管理程式的資料及日誌來建立佇列管理程式，這些目錄不在安裝所建立的資料及日誌路徑上。在第一個範例 [第 316 頁的『讀取及寫入本端 mqm 群組授權的資料及日誌檔』](#) 中，您透過本端 mqm 群組授權來允許存取佇列及日誌目錄。第二個範例 [第 319 頁的『讀取及寫入替代本端安全群組授權的資料及日誌檔』](#) 不同，因為對目錄的存取權是由替代安全群組授權。當只在一部伺服器上執行的佇列管理程式存取目錄時，使用替代安全群組來保護資料和日誌檔的安全，可讓您選擇使用不同的本端群組或主體來保護不同的佇列管理程式。當在不同伺服器 (例如使用多重實例佇列管理程式) 上執行的佇列管理程式存取目錄時，使用替代安全群組來保護資料及日誌檔的安全是唯一選項; 請參閱 [第 312 頁的『在 Windows 上保護共用佇列管理程式資料及日誌目錄和檔案的安全』](#)。

在 Windows 上配置佇列管理程式資料及日誌檔的安全許可權不是一般作業。安裝 IBM WebSphere MQ for Windows 時，您可以指定佇列管理程式資料及日誌的目錄，或接受預設目錄。安裝程式會使用本端 mqm 群

組自動保護這些目錄的安全，並提供其完整控制許可權。安裝程序會確定執行佇列管理程式的使用者 ID 是本端 `mqm` 群組的成員。您可以修改目錄上的其他存取權，以符合您的存取需求。

如果您將資料及日誌檔目錄移至新位置，則必須配置新位置的安全。如果您備份佇列管理程式並將它還原至不同的電腦，或者如果您將佇列管理程式變更為多重實例佇列管理程式，則可以變更目錄的位置。您可以選擇兩種方法來保護新位置中佇列管理程式資料及日誌目錄的安全。您可以透過限制對本端 `mqm` 群組的存取權來保護目錄的安全，也可以限制對您選擇的任何安全群組的存取權。

使用本端 `mqm` 群組來保護目錄的步驟數最少。設定資料及日誌目錄的許可權，以容許本端 `mqm` 群組完全控制。一般方法是複製現有的許可權集，並從母項移除繼承。然後，您可以移除或限制其他主體的許可權。

如果您使用與「準備 IBM WebSphere MQ」精靈所設定服務不同的使用者 ID 來執行佇列管理程式，則該使用者 ID 必須是本端 `mqm` 群組的成員。作業第 316 頁的『讀取及寫入本端 `mqm` 群組授權的資料及日誌檔』會引導您完成步驟。

您也可以使用替代安全群組來保護佇列管理程式資料及日誌檔的安全。使用替代安全群組保護佇列管理程式資料及日誌檔的處理程序具有一些步驟，請參閱第 316 頁的圖 68。本端群組 `wmq` 是替代安全群組的範例。

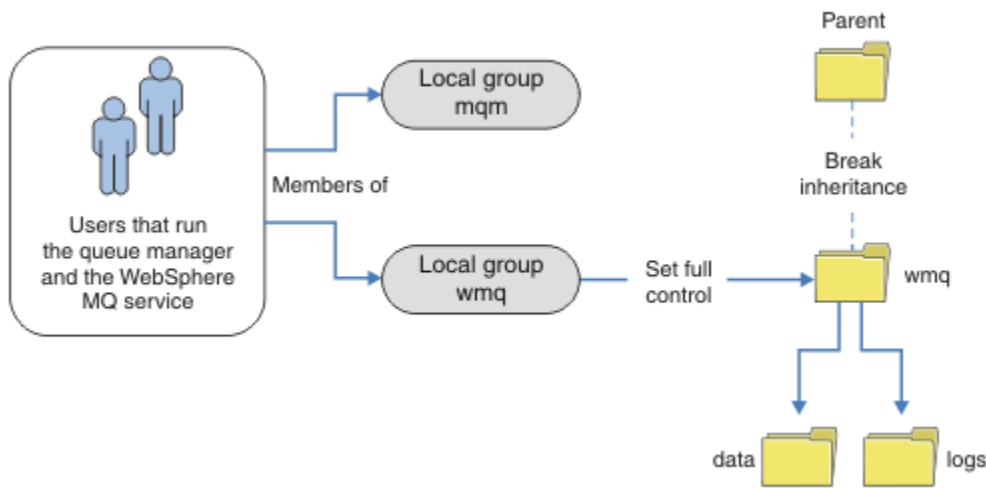


圖 68: 使用替代本端安全群組 `wmq` 來保護佇列管理程式資料及日誌的安全

1. 為佇列管理程式資料及日誌建立個別目錄、共用目錄或共用上層目錄。
2. 複製目錄或上層目錄的現有繼承許可權集，並根據您的需要來修改它們。
3. 授與替代群組 `wmq` 對目錄的完整控制權，以保護將包含佇列管理程式及日誌的目錄安全。
4. 提供執行佇列管理程式的所有使用者 ID，以處理替代安全群組或主體的認證：
 - a. 如果您將使用者定義為替代安全主體，則該使用者必須與將要在下執行佇列管理程式的使用者相同。使用者必須是本端 `mqm` 群組的成員。
 - b. 如果您將本端群組定義為替代安全群組，請將佇列管理程式將在其下執行的使用者新增至替代群組。使用者也必須是本端 `mqm` 群組的成員。
 - c. 如果您將廣域群組定義為替代安全群組，請參閱第 312 頁的『在 Windows 上保護共用佇列管理程式資料及日誌目錄和檔案的安全』。
5. 在 `crtmqm` 指令上使用 `-a` 參數指定替代安全群組或主體，以建立佇列管理程式。

讀取及寫入本端 `mqm` 群組授權的資料及日誌檔

此作業說明如何建立佇列管理程式，其資料及日誌檔儲存在您選擇的任何目錄中。檔案的存取權受到本端 `mqm` 群組的保護。目錄不共用。

開始之前

1. 安裝 IBM WebSphere MQ for Windows 作為主要安裝。
2. 執行 "準備 IBM WebSphere MQ" 精靈。對於此作業，請將安裝配置為以本端使用者 ID 或網域使用者 ID 執行。最後，若要完成 [第 290 頁的『Windows 網域及多重實例佇列管理程式』](#) 中的所有作業，必須為網域配置安裝。
3. 以管理者權限登入以執行作業的第一部分。

關於這項作業

此作業是一組說明存取佇列管理程式資料及日誌檔的相關作業之一。這些作業顯示如何建立佇列管理程式，以授權讀取及寫入儲存在您選擇的目錄中的資料和日誌檔。它們隨附作業 [第 290 頁的『Windows 網域及多重實例佇列管理程式』](#)。

在 Windows 上，您可以在您選擇的任何目錄中建立 IBM WebSphere MQ for Windows 的預設資料及日誌路徑。安裝與配置精靈會自動提供目錄存取權給本端 mqm 群組，以及執行佇列管理程式處理程序的使用者 ID。如果您建立佇列管理程式，並為佇列管理程式資料及日誌檔指定不同的目錄，則必須配置目錄的完整控制許可權。

在此範例中，您將 `c:\wmq` 目錄的許可權授與本端 mqm 群組，讓佇列管理程式完全控制其資料及日誌檔。

`crtmqm` 指令會建立在 workstation 使用 IBM WebSphere MQ 服務啟動時自動啟動的佇列管理程式。

作業是說明性的；它使用您可以變更的特定值。您可以變更的值是斜體。在作業結束時，請遵循指示來移除您所做的所有變更。

程序

1. 開啟一個命令提示字元。
2. 鍵入指令：

```
md c:\wmq\data , c:\wmq\logs
```

3. 設定目錄的許可權，以容許本端 mqm 群組讀取及寫入權。

```
cacls c:\wmq /T /E /G mqm:F
```

系統回應：

```
processed dir: c:\wmq
processed dir: c:\wmq\data
processed dir: c:\wmq\logs
```

4. 選擇性的：切換至本端 mqm 群組成員的使用者 ID。

您可以繼續以「管理者」身分，但如需實際的正式作業配置，請繼續使用具有更受限權限的使用者 ID。使用者 ID 必須至少是本端 mqm 群組的成員。

如果 IBM WebSphere MQ 安裝配置為網域的一部分，請讓使用者 ID 成為 Domain mqm 群組的成員。「準備 IBM WebSphere MQ」精靈會使 Domain mqm 廣域群組成為本端 mqm 群組的成員，因此您不需要讓使用者 ID 直接成為本端 mqm 群組的成員。

5. 建立佇列管理程式。

```
crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE -md c:\wmq\data -ld c:\wmq\logs QMGR
```

系統回應：

```
WebSphere MQ queue manager created.
Directory 'c:\wmq\data\QMGR' created.
The queue manager is associated with installation '1'
Creating or replacing default objects for queue manager 'QMGR'
Default objects statistics : 74 created. 0 replaced.
```

Completing setup.

Setup completed.

6. 檢查佇列管理程式所建立的目錄是否位於 `c:\wmq` 目錄中。

```
dir c:\wmq /D /B /S
```

7. 檢查檔案是否具有本端 `mqm` 群組的讀取及寫入，或完整控制許可權。

```
cacls c:\wmq\*.*
```

下一步

將訊息放入佇列並取得訊息，以測試佇列管理程式。

1. 啟動佇列管理程式。

```
strmqm QMGR
```

系統回應：

```
WebSphere MQ queue manager 'QMGR' starting.  
The queue manager is associated with installation '1'.  
5 log records accessed on queue manager 'QMGR' during the log  
replay phase.  
Log replay for queue manager 'QMGR' complete.  
Transaction manager state recovered for queue manager 'QMGR'.  
WebSphere MQ queue manager 'QMGR' started using V7.1.0.0.
```

2. 建立測試佇列。

```
echo define qlocal(QTEST) | runmqsc QMGR
```

系統回應：

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
Starting MQSC for queue manager QMGR.
```

```
1 : define qlocal(QTEST)  
AMQ8006: WebSphere MQ queue created.  
One MQSC command read.  
No commands have a syntax error.  
All valid MQSC commands were processed.
```

3. 使用範例程式 **amqsput** 來放置測試訊息。

```
echo 'A test message' | amqsput QTEST QMGR
```

系統回應：

```
Sample AMQSPUT0 start  
target queue is QTEST  
Sample AMQSPUT0 end
```

4. 使用範例程式 **amqsget** 取得測試訊息。

```
amqsget QTEST QMGR
```

系統回應：

```
Sample AMQSGET0 start  
message <A test message>  
Wait 15 seconds ...
```

```
no more messages
Sample AMQSGETO end
```

5. 停止佇列管理程式。

```
endmqm -i QMGR
```

系統回應：

```
WebSphere MQ queue manager 'QMGR' ending.
WebSphere MQ queue manager 'QMGR' ended.
```

6. 刪除佇列管理程式。

```
dltmqm QMGR
```

系統回應：

```
WebSphere MQ queue manager 'QMGR' deleted.
```

7. 刪除您建立的目錄。

提示：將 /Q 選項新增至指令，以防止指令提示刪除每一個檔案或目錄。

```
del /F /S C:\wmq\*. *
rmdir /S C:\wmq
```

相關概念

[第 290 頁的『Windows 網域及多重實例佇列管理程式』](#)

Windows 上的多重實例佇列管理程式需要共用其資料及日誌。共用必須可供在不同伺服器或工作站上執行。所有佇列管理程式實例存取。配置佇列管理程式並共用作為 Windows 網域的一部分。佇列管理程式可以在網域工作站或伺服器上執行，或在網域控制站上執行。

相關工作

[第 319 頁的『讀取及寫入替代本端安全群組授權的資料及日誌檔』](#)

此作業顯示如何在 **crtmqm** 指令上使用 **-a** 旗標。此旗標為佇列管理程式提供替代本端安全群組，讓它可以存取其日誌及資料檔案。

[第 301 頁的『讀取及寫入由替代廣域安全群組授權的共用資料及日誌檔』](#)

[第 291 頁的『在網域工作站或伺服器上建立多重實例佇列管理程式』](#)

[讀取及寫入替代本端安全群組授權的資料及日誌檔](#)

此作業顯示如何在 **crtmqm** 指令上使用 **-a** 旗標。此旗標為佇列管理程式提供替代本端安全群組，讓它可以存取其日誌及資料檔案。

開始之前

1. 安裝 IBM WebSphere MQ for Windows 作為主要安裝。
2. 執行 "準備 IBM WebSphere MQ" 精靈。對於此作業，請將安裝配置為以本端使用者 ID 或網域使用者 ID 執行。最後，若要完成 [第 290 頁的『Windows 網域及多重實例佇列管理程式』](#) 中的所有作業，必須為網域配置安裝。
3. 以管理者權限登入以執行作業的第一部分。

關於這項作業

此作業是一組說明存取佇列管理程式資料及日誌檔的相關作業之一。這些作業顯示如何建立佇列管理程式，以授權讀取及寫入儲存在您選擇的目錄中的資料和日誌檔。它們隨附作業 [第 290 頁的『Windows 網域及多重實例佇列管理程式』](#)。

在 Windows 上，您可以在您選擇的任何目錄中建立 IBM WebSphere MQ for Windows 的預設資料及日誌路徑。安裝與配置精靈會自動提供目錄存取權給本端 mqm 群組，以及執行佇列管理程式處理程序的使用者 ID。如果您建立佇列管理程式，並為佇列管理程式資料及日誌檔指定不同的目錄，則必須配置目錄的完整控制許可權。

在此範例中，您為佇列管理程式提供具有目錄完整控制權的替代安全本端群組。替代安全群組提供佇列管理程式許可權來管理目錄中的檔案。替代安全群組的主要目的是授權替代安全廣域群組。使用替代安全廣域群組來設定多重實例佇列管理程式。在此範例中，您將配置本端群組以熟悉使用替代安全群組，而無需在網域中安裝 IBM WebSphere MQ。將本端群組配置為替代安全群組並不常見。

`crtmqm` 指令會建立在工作站使用 IBM IBM WebSphere MQ 服務啟動時自動啟動的佇列管理程式。

作業是說明性的；它使用您可以變更的特定值。您可以變更的值是斜體。在作業結束時，請遵循指示來移除您所做的所有變更。

程序

1. 設定替代安全群組。

替代安全群組通常是網域群組。在此範例中，您建立使用本端替代安全群組的佇列管理程式。使用本端替代安全群組，您可以使用不屬於網域的 IBM WebSphere MQ 安裝來執行作業。

- a) 執行 `lusrmgr.msc` 指令，以開啟「本端使用者和群組」視窗。
- b) 用滑鼠右鍵按一下 **群組 > 新建群組 ...**
- c) 在 **群組名稱** 欄位中輸入 `altmqm`，然後按一下 **建立 > 關閉**。
- d) 識別執行 IBM IBM WebSphere MQ 服務的使用者 ID。
 - i) 按一下 **開始 > 執行 ...**，鍵入 `services.msc`，然後按一下 **確定**。
 - ii) 按一下服務清單中的 IBM IBM WebSphere MQ 服務，然後按一下「登入」標籤。
 - iii) 請記住使用者 ID，並關閉「服務瀏覽器」。
- e) 將執行 IBM IBM WebSphere MQ 服務的使用者 ID 新增至 `altmqm` 群組。同時新增您用來登入以建立佇列管理程式的使用者 ID，並以互動方式執行它。

Windows 會檢查執行佇列管理程式處理程序之使用者 ID 的權限，以檢查佇列管理程式存取資料及日誌目錄的權限。使用者 ID 必須是授權目錄之 `altmqm` 群組的成員（直接或間接透過廣域群組）。如果您安裝 IBM WebSphere MQ 作為網域的一部分，並且要在第 291 頁的『在網域工作站或伺服器上建立多重實例佇列管理程式』中執行作業，則在第 294 頁的『建立 IBM WebSphere MQ 的 Active Directory 及 DNS 網域』中建立的網域使用者 ID 是 `wmquser1` 及 `wmquser2`。

如果您未安裝佇列管理程式作為網域的一部分，則執行 IBM IBM WebSphere MQ 服務的預設本端使用者 ID 為 `MUSR_MQADMIN`。如果您想要在沒有管理者權限的情況下執行作業，請建立屬於本端 `mqm` 群組成員的使用者。

請遵循下列步驟，將 `wmquser1` 和 `wmquser2` 新增至 `altmqm`。如果您的配置不同，請以您的名稱來取代使用者 ID 和群組。

- i) 在群組清單中，用滑鼠右鍵按一下 **altmqm > 內容 > 新增 ...**
 - ii) 在「選取使用者、電腦或群組」視窗中鍵入 `wmquser1`；`wmquser2`，然後按一下 **檢查名稱**。
 - iii) 在「Windows 安全」視窗中鍵入網域管理者的名稱及密碼，然後按一下 **確定 > 確定 > 套用 > 確定**。
2. 開啟一個命令提示字元。
3. 重新啟動 IBM IBM WebSphere MQ 服務。

您必須重新啟動服務，讓它在其下執行的使用者 ID 獲得您為它配置的其他安全認證。

鍵入指令：

```
endmqsvc  
strmqsvc
```

系統回應：

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
The MQ service for installation 'Installation1' ended successfully.  
及：
```


5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.
The MQ service for installation 'Installation1' started successfully.

4. 鍵入指令:

```
md c:\wmq\data , c:\wmq\logs
```

5. 設定目錄的許可權，以容許本端使用者 *user* 讀取及寫入權。

```
cacls c:\wmq /T /E /G almqm:F
```

系統回應:

```
processed dir: c:\wmq
processed dir: c:\wmq\data
processed dir: c:\wmq\logs
```

6. 選擇性的: 切換至本端 *mqm* 群組成員的使用者 ID。

您可以繼續以「管理者」身分，但如需實際的正式作業配置，請繼續使用具有更受限權限的使用者 ID。使用者 ID 必須至少是本端 *mqm* 群組的成員。

如果 IBM WebSphere MQ 安裝配置為網域的一部分，請讓使用者 ID 成為 Domain *mqm* 群組的成員。「準備 IBM WebSphere MQ」精靈會使 Domain *mqm* 廣域群組成為本端 *mqm* 群組的成員，因此您不需要讓使用者 ID 直接成為本端 *mqm* 群組的成員。

7. 建立佇列管理程式。

```
crtmqm -a almqm -sax -u SYSTEM.DEAD.LETTER.QUEUE -md c:\wmq\data -ld c:\wmq\logs QMGR
```

系統回應:

```
WebSphere MQ queue manager created.
Directory 'c:\wmq1\data\QMGR' created.
The queue manager is associated with installation '1'
Creating or replacing default objects for queue manager 'QMGR'
Default objects statistics : 74 created. 0 replaced.
Completing setup.
Setup completed.
```

8. 檢查佇列管理程式所建立的目錄是否位於 *c:\wmq* 目錄中。

```
dir c:\wmq /D /B /S
```

9. 檢查檔案是否具有本端 *mqm* 群組的讀取及寫入，或完整控制許可權。

```
cacls c:\wmq\*.*
```

下一步

將訊息放入佇列並取得訊息，以測試佇列管理程式。

1. 啟動佇列管理程式。

```
strmqm QMGR
```

系統回應:

```
WebSphere MQ queue manager 'QMGR' starting.
The queue manager is associated with installation '1'.
5 log records accessed on queue manager 'QMGR' during the log
replay phase.
Log replay for queue manager 'QMGR' complete.
Transaction manager state recovered for queue manager 'QMGR'.
WebSphere MQ queue manager 'QMGR' started using V7.1.0.0.
```

2. 建立測試佇列。

```
echo define qlocal(QTEST) | runmqsc QMGR
```

系統回應:

```
5724-H72 (C) Copyright IBM Corp. 1994, 2024. ALL RIGHTS RESERVED.  
Starting MQSC for queue manager QMGR.
```

```
1 : define qlocal(QTEST)  
AMQ8006: WebSphere MQ queue created.  
One MQSC command read.  
No commands have a syntax error.  
All valid MQSC commands were processed.
```

3. 使用範例程式 **amqsput** 來放置測試訊息。

```
echo 'A test message' | amqsput QTEST QMGR
```

系統回應:

```
Sample AMQSPUT0 start  
target queue is QTEST  
Sample AMQSPUT0 end
```

4. 使用範例程式 **amqsget** 取得測試訊息。

```
amqsget QTEST QMGR
```

系統回應:

```
Sample AMQSGET0 start  
message <A test message>  
Wait 15 seconds ...  
no more messages  
Sample AMQSGET0 end
```

5. 停止佇列管理程式。

```
endmqm -i QMGR
```

系統回應:

```
WebSphere MQ queue manager 'QMGR' ending.  
WebSphere MQ queue manager 'QMGR' ended.
```

6. 刪除佇列管理程式。

```
dltmqm QMGR
```

系統回應:

```
WebSphere MQ queue manager 'QMGR' deleted.
```

7. 刪除您建立的目錄。

提示: 將 /Q 選項新增至指令，以防止指令提示刪除每一個檔案或目錄。

```
del /F /S C:\wmq\*.*  
rmdir /S C:\wmq
```

在 Linux 上建立多重實例佇列管理程式

範例顯示如何在 Linux 上設定多重實例佇列管理程式。此設定很小，以說明所涉及的概念。此範例以 Linux Red Hat Enterprise 5 為基礎。其他 UNIX 平台上的步驟有所不同。

此範例設定在 2 GHz 筆記型電腦上，具有執行 Windows XP Service Pack 2 的 3 GB RAM。兩個 VMware 虛擬機器 Server1 和 Server2，在 640 MB 映像檔中執行 Linux Red Hat Enterprise 5。Server1 會管理網路檔

案系統 (NFS)、佇列管理程式日誌及 HA 實例。NFS 伺服器通常不會同時管理其中一個佇列管理程式實例；這是為了簡化範例。Server2 會使用待命實例裝載 Server1 的佇列管理程式日誌。WebSphere MQ MQI 用戶端安裝在額外 400 MB VMware 映像檔上，且會執行 Windows XP Service Pack 2 並執行範例高可用性應用程式。出於安全原因，所有虛擬機器都配置為 VMware 僅限主機網路的一部分。

註：您應該只將佇列管理程式資料放置在 NFS 伺服器上。在 NFS 上，搭配使用下列三個選項與裝載指令，以確保系統安全：

noexec

透過使用此選項，您可以停止在 NFS 上執行二進位檔，以防止遠端使用者在系統上執行不想要的程式碼。

諾蘇伊德

透過使用此選項，您可以防止使用 set-user-identifier 及 set-group-identifier 位元，這會防止遠端使用者取得更高的專用權。

諾傑夫

透過使用此選項，您可以停止使用或定義字元及封鎖特殊裝置，這會防止遠端使用者從 chroot 監獄中越獄。

範例

表 30: Linux 上的說明性多重實例佇列管理程式配置	
Server1	Server2
以 root 身分登入	
遵循 安裝 IBM WebSphere MQ 中的指示來安裝 WebSphere MQ，建立 mqm 使用者和群組 (如果不存在的話)，並定義 /var/mqm。 檢查第一部機器上 /etc/passwd 中針對 mqm 顯示的 uid 和 gid；例如， mqm:x:501:100:MQ User:/var/mqm:/bin/bash 針對第二部機器上 /etc/passwd 中的 mqm，比對 uid 和 gid，以確保這些值相同。如果您必須變更值，請將此機器重新開機。	
完成 驗證共用檔案系統行為 作業，以檢查檔案系統是否支援多重實例佇列管理程式。	
在要共用的共用資料夾 /MQHA 中建立日誌和資料目錄。 例如： 1. mkdir /MQHA 2. mkdir /MQHA/logs 3. mkdir /MQHA/qmgrs	建立資料夾 /MQHA，以裝載共用檔案系統。將路徑與 Server1 上的路徑保持相同，例如： 1. mkdir /MQHA
確保使用者和群組 mqm 擁有 MQHA 目錄，並將使用者和群組的存取權設為 rwX ；例如 ls -al 顯示： drwxrwxr-x mqm mqm 4096 Nov 27 14:38 MQDATA 1. chown -R mqm:mqm /MQHA 2. chmod -R ug+rwx /MQHA	
建立佇列管理程式： crtmqm -ld /MQHA/logs -md /MQHA/qmgrs QM1	
新增 ² /MQHA *(rw, sync, no_wdelay, fsid=0) 至 /etc/exports	

² '*' 容許所有可以達到這個裝載 /MQHA 進行讀寫的機器。限制正式作業機器上的存取權。

表 30: Linux 上的說明性多重實例佇列管理程式配置 (繼續)	
Server1	Server2
啟動 NFS 常駐程式: <code>/etc/init.d/nfs start</code>	裝載匯出的檔案系統 /MQHA: <code>mount -t nfs4 -o hard,intr Server1:/ /MQHA</code>
從 Server1: 複製佇列管理程式配置詳細資料 <code>dspmqinf -o command QM1</code> 並將結果複製到剪貼簿: <code>addmqinf -s QueueManager -v Name=QM1 -v Directory=QM1 -v Prefix=/var/mqm -v DataPath=/MQHA/qmgrs/QM1</code>	將佇列管理程式配置指令貼到 Server2: <code>addmqinf -s QueueManager -v Name=QM1 -v Directory=QM1 -v Prefix=/var/mqm -v DataPath=/MQHA/qmgrs/QM1</code>
使用 -x 參數, 以任一順序啟動佇列管理程式實例: <code>strmqm -x QM1</code> 用來啟動佇列管理程式實例的指令必須從與 <code>addmqinf</code> 指令相同的 IBM WebSphere MQ 安裝中發出。若要從不同的安裝啟動及停止佇列管理程式, 您必須先使用 <code>setmqm</code> 指令設定與佇列管理程式相關聯的安裝。如需相關資訊, 請參閱 setmqm 。	

在 Linux 上驗證多重實例佇列管理程式

使用範例程式 `amqsgshac`、`amqspshac` 及 `amqsmshac` 來驗證多重實例佇列管理程式配置。本主題提供在 Linux Red Hat Enterprise 5 上驗證多重實例佇列管理程式配置的範例配置。

高可用性範例程式使用自動用戶端重新連線。當連接的佇列管理程式失敗時, 用戶端會嘗試重新連接至相同佇列管理程式群組中的佇列管理程式。範例的說明 (高可用性範例程式) 示範使用單一實例佇列管理程式的用戶端重新連線, 以簡化。您可以將相同的範例與多重實例佇列管理程式搭配使用, 以驗證多重實例佇列管理程式配置。

此範例使用第 322 頁的『在 Linux 上建立多重實例佇列管理程式』中說明的多重實例配置。使用配置來驗證多重實例佇列管理程式是否切換至待命實例。使用 `endmqm` 指令停止佇列管理程式, 並使用 `-s`, 切換, 選項。用戶端程式會重新連接至新的佇列管理程式實例, 並在稍微延遲之後繼續使用新的實例。

在此範例中, 用戶端是在 Windows XP Service Pack 2 系統上執行。系統正在管理兩個執行多重實例佇列管理程式的 VMware Linux 伺服器。

使用 WebSphere MQ 探險家驗證失效接手

在使用範例應用程式來驗證失效接手之前, 請在每一部伺服器上執行「WebSphere MQ 探險家」。使用「**新增遠端佇列管理程式 > 直接連接多重實例佇列管理程式**」精靈, 將這兩個佇列管理程式實例新增至每一個瀏覽器。確保兩個實例都在執行中, 允許待命。關閉使用作用中實例執行 VMware 映像檔的視窗, 虛擬關閉伺服器電源, 或停止作用中實例, 容許切換至待命實例。

註: 如果您關閉伺服器電源, 請確定它不是管理 /MQHA 的伺服器!

註: 在「**停止佇列管理程式**」對話框上可能無法使用 **容許切換至待命實例** 選項。遺漏選項, 因為佇列管理程式作為單一實例佇列管理程式執行。您必須已在沒有 **允許待命實例** 選項的情況下啟動它。如果您拒絕停止佇列管理程式的要求, 請查看「**詳細資料**」視窗, 這可能是因為沒有待命實例在執行中。

使用範例程式驗證失效接手

選擇要執行作用中實例的伺服器

您可能已選擇其中一部伺服器來管理 MQHA 目錄或檔案系統。如果您計劃透過關閉執行作用中伺服器的 VMware 視窗來測試失效接手, 請確定它不是管理 MQHA!

在執行作用中佇列管理程式實例的伺服器上

註: 在 MCAUSER 設為 mqm 的情況下執行 SVRCONN 通道, 可方便減少範例中配置步驟的數目。如果選擇另一個使用者 ID, 且您的系統設定不同於範例中使用的使用者 ID, 則您可能會遇到存取權問題。請勿在公開的系統上使用 mqm 作為 MCAUSER; 它可能會嚴重危及安全。

1. 修改 `ipaddr1` 及 `ipaddr2`, 並將下列指令儲存在 `/MQHA/hasamples.tst` 中

```
DEFINE QLOCAL(SOURCE) REPLACE
DEFINE QLOCAL(TARGET) REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
MCAUSER('mqm') REPLACE
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNNAME('ipaddr1(1414),ipaddr2
(1414)') QMNAME(QM1) REPLACE
START CHANNEL(CHANNEL1)
DEFINE LISTENER(LISTENER.TCP) TRPTYPE(TCP) CONTROL(QMGR)
DISPLAY LISTENER(LISTENER.TCP) CONTROL
START LISTENER(LISTENER.TCP)
DISPLAY LSSTATUS(LISTENER.TCP) STATUS
```

2. 開啟具有路徑 `/MQHA` 的終端機視窗, 然後執行下列指令:

```
runmqsc -m QM1 < hasamples.tst
```

3. 透過檢查 `runmqsc` 指令的輸出, 驗證接聽器是否在執行中且具有佇列管理程式控制。

```
LISTENER(LISTENER.TCP)CONTROL(QMGR)
LISTENER(LISTENER.TCP)STATUS(RUNNING)
```

或者, 使用 TCPIP 接聽器正在執行且具有 `Control = Queue Manager` 的「WebSphere MQ 探險家」。

在用戶端上

1. 將用戶端連線表格 `AMQCLCHL.TAB` 從伺服器上的 `/MQHA/qmgrs/QM1.000/@ipcc` 複製到用戶端上的 `C:\`。
2. 以路徑 `C:\` 開啟命令提示字元, 並將環境變數 `MQCHLLIB` 設為指向用戶端通道定義表 (CCDT)

```
SET MQCHLLIB=C:\
```

3. 在命令提示字元中鍵入下列指令:

```
start amqsghac TARGET QM1
start amqsmhac -s SOURCE -t TARGET -m QM1
start amqsphac SOURCE QM1
```

在執行作用中佇列管理程式實例的伺服器上

1. 您可以:
 - 關閉使用作用中伺服器實例執行 VMware 映像檔的視窗。
 - 使用「WebSphere MQ 探險家」, 停止作用中佇列管理程式實例, 容許切換至待命實例, 並指示可重新連接的用戶端重新連接。
2. 這三個用戶端最終會偵測到連線已中斷, 然後重新連接。在此配置中, 如果您關閉伺服器視窗, 則重建這三個連線大約需要 7 分鐘。有些連線在其他連線之前就已重新建立。

結果

```
N:\>amqsphac SOURCE QM1
Sample AMQSPHAC start
target queue is SOURCE
message <Message 1>
message <Message 2>
message <Message 3>
message <Message 4>
message <Message 5>
17:05:25 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message <Message 6>
message <Message 7>
message <Message 8>
message <Message 9>
```

```
N:\>amqsmhac -s SOURCE -t TARGET -m QM1
Sample AMQSMHA0 start
```

```
17:05:25 : EVENT : Connection Reconnecting (Delay: 97ms)
17:05:48 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:53 : EVENT : Connection Reconnected
```

```
N:\>amqsgnac TARGET QM1
```

```
Sample AMQSGHAC start
message <Message 1>
message <Message 2>
message <Message 3>
message <Message 4>
message <Message 5>
17:05:25 : EVENT : Connection Reconnecting (Delay: 156ms)
17:05:47 : EVENT : Connection Reconnecting (Delay: 0ms)
17:05:52 : EVENT : Connection Reconnected
message <Message 6>
message <Message 7>
message <Message 8>
message <Message 9>
```

刪除多重實例佇列管理程式

若要完全刪除多重實例佇列管理程式，您需要使用 **dltmqm** 指令來刪除佇列管理程式，然後使用 **rmvmqinf** 或 **dltmqm** 指令從其他伺服器中移除實例。

執行 **dltmqm** 指令，以在定義該佇列管理程式的任何伺服器上，刪除已在其他伺服器上定義實例的佇列管理程式。您不需要在建立 **dltmqm** 指令的相同伺服器上執行該指令。然後在具有佇列管理程式定義的所有其他伺服器上執行 **rmvmqinf** 或 **dltmqm** 指令。

您只能刪除已停止的佇列管理程式。在您刪除它時，沒有執行中的實例，嚴格來說，佇列管理程式不是單一或多重實例佇列管理程式；它只是一個具有其佇列管理程式資料並登入遠端共用的佇列管理程式。當您刪除佇列管理程式時，會刪除其佇列管理程式資料及日誌，並從您發出 **dltmqm** 指令之伺服器上的 **mqs.ini** 檔案中移除佇列管理程式段落。刪除佇列管理程式時，您需要對包含佇列管理程式資料及日誌的網路共用具有存取權。

在您先前已建立佇列管理程式實例的其他伺服器上，這些伺服器上的 **mqs.ini** 檔案中也有項目。您需要依序造訪每一部伺服器，並執行指令 **rmvmqinf** 佇列管理程式段落名稱來移除佇列管理程式段落。

在 UNIX and Linux 系統上，如果您已在網路儲存體中放置一般 **mqs.ini** 檔案，並透過在每一部伺服器上設定 **AMQ_MQS_INI_LOCATION** 環境變數，從所有伺服器中參照該檔案，則需要只從其中一部伺服器中刪除佇列管理程式，因為只有一個 **mqs.ini** 檔案要更新。

範例

第一個伺服器

```
dltmqm QM1
```

定義實例的其他伺服器

```
rmvmqinf QM1, 或
```

啟動和停止多重實例佇列管理程式

啟動和停止配置為單一實例或多重實例佇列管理程式的佇列管理程式。

當您在一對伺服器上定義多重實例佇列管理程式時，您可以在任一伺服器上執行佇列管理程式，作為單一實例佇列管理程式或多重實例佇列管理程式。

若要執行多重實例佇列管理程式，請使用 **strmqm -x QM1** 指令，在其中一部伺服器上啟動佇列管理程式；**-x** 選項允許實例進行失效接手。它會變成作用中實例。使用相同的 **strmqm -x QM1** 指令，在另一部伺服器上啟動待命實例；**-x** 選項允許實例作為待命實例啟動。

現在，佇列管理程式正在執行中，其中一個作用中實例正在處理所有要求，另一個待命實例已準備好在作用中實例失敗時接管。作用中實例會被授與佇列管理程式資料及日誌的專用存取權。待命資料庫會等待獲得佇列管理程式資料及日誌的專用存取權。當授與待命資料庫互斥存取權時，它會變成作用中實例。

您也可以作用中實例上發出 **endmqm -s** 指令，以手動將控制切換至待命實例。**endmqm -s** 指令會關閉作用中實例，而不關閉待命實例。佇列管理程式資料及日誌上的專用存取鎖定已釋放，待命資料庫會接管。

您也可以啟動及停止在不同伺服器上配置多個實例作為單一實例佇列管理程式的佇列管理程式。如果您未在 **strmqm** 指令上使用 **-x** 選項來啟動佇列管理程式，則會阻止在其他機器上配置的佇列管理程式實例啟動為待命實例。如果您嘗試啟動另一個實例，則會收到不允許佇列管理程式實例作為待命實例執行的回應。

如果您使用不含 **-s** 選項的 **endmqm** 指令來停止多重實例佇列管理程式的作用中實例，則作用中及待命實例都會停止。如果您搭配使用 **endmqm** 指令與 **-x** 選項來停止待命實例，則它會停止成為待命實例，且作用中實例會繼續執行。如果備用資料庫上沒有 **-x** 選項，則無法發出 **endmqm**。

只能同時執行兩個佇列管理程式實例；一個是作用中實例，另一個是待命實例。如果您同時啟動兩個實例，則 WebSphere MQ 無法控制哪個實例變成作用中實例；它由網路檔案系統決定。第一個獲得佇列管理程式資料互斥存取權的實例會變成作用中實例。

註：在重新啟動失敗的佇列管理程式之前，您必須先切斷應用程式與該佇列管理程式實例的連線。如果沒有，佇列管理程式可能無法正確重新啟動。

共用檔案系統

多重實例佇列管理程式使用網路檔案系統來管理佇列管理程式實例。

多重實例佇列管理程式會使用檔案系統鎖定與共用佇列管理程式資料及日誌的組合，來自動化失效接手。只有一個佇列管理程式實例可以具有共用佇列管理程式資料及日誌的獨佔性存取權。當它取得存取權時，它會變成作用中實例。在佇列管理程式資料及日誌變成可用之前，未成功取得專用存取權的另一個實例會等待成為待命實例。

網路檔案系統負責釋放它對作用中佇列管理程式實例所保留的鎖定。如果作用中實例以某種方式失敗，則網路檔案系統會釋放它為作用中實例所保留的鎖定。只要釋放專用鎖定，等待鎖定的待命佇列管理程式就會嘗試獲得它。如果成功，它會變成作用中實例，且對共用檔案系統上的佇列管理程式資料及日誌具有專用存取權。然後它會繼續啟動。

規劃檔案系統支援 相關主題說明如何設定及檢查您的檔案系統是否支援多重實例佇列管理程式。

多重實例佇列管理程式不會保護您免於在檔案系統中發生失敗。有許多方法可以保護您的資料。

- 投資可靠的儲存體 (例如備援磁碟陣列 (RAID))，並將它們包含在具有網路備援的網路檔案系統中。
- 將 WebSphere MQ 線性日誌備份至替代媒體，如果主要日誌媒體失敗，請使用替代媒體上的日誌來回復。您可以使用備份佇列管理程式來管理此處理程序。

多個佇列管理程式實例

多重實例佇列管理程式具有復原力，因為它在失敗之後使用待命佇列管理程式實例來還原佇列管理程式可用性。

抄寫佇列管理程式實例是一種非常有效的方法，可增進佇列管理程式處理程序的可用性。使用簡式可用性模型，純粹為圖解：如果佇列管理程式的一個實例的可靠性為 99% (超過一年，累積關閉時間為 3.65 天)，則新增佇列管理程式的另一個實例會將可用性增加至 99.99% (超過一年，累積關閉時間大約一小時)。

這個模型太簡單，無法為您提供可用性的實際數值估計值。若要現實地建立可用性模型，您需要收集故障之間的平均時間 (MTBF) 與平均修復時間 (MTTR) 的統計資料，以及故障與修復時間之間的時間機率分佈。

「多重實例佇列管理程式」一詞是指共用佇列管理程式資料及日誌之佇列管理程式的作用中及待命實例組合。多重實例佇列管理程式可讓一個佇列管理程式實例在一部伺服器上處於作用中狀態，另一個佇列管理程式實例在另一部伺服器上處於待命狀態，並在作用中實例失敗時準備自動接管，以防止佇列管理程式處理程序失敗。

失效接手或切換

待命佇列管理程式實例會根據要求(切換)，或在作用中實例失敗(失效接手)時，從作用中實例接管。

- 當待命實例啟動以回應向作用中佇列管理程式實例發出的 **endmqm -s** 指令時，即會進行切換。您可以指定 **endmqm** 參數 **-c**、**-i** 或 **-p**，以控制佇列管理程式突然停止的程度。

註：只有在待命佇列管理程式實例已啟動時，才會進行切換。**endmqm -s** 指令會釋放作用中佇列管理程式鎖定並允許切換：它不會啟動待命佇列管理程式實例。

- 當因為實例似乎非預期地停止(亦即，未發出 **endmqm** 指令)，而釋放作用中實例所保留佇列管理程式資料的鎖定時，即會發生失效接手。

當待命實例接管作為作用中實例時，它會將訊息寫入佇列管理程式錯誤日誌。

當佇列管理程式失敗或切換時，可重新連接的用戶端會自動重新連接。您不需要在 **endmqm** 指令上包括 **-r** 旗標，即可要求用戶端重新連線。Java 的 WebSphere MQ 類別不支援自動用戶端重新連接。

如果您發現無法重新啟動失敗的實例，即使發生失效接手且待命實例已變成作用中，請檢查本端連接至失敗實例的應用程式未與失敗實例中斷連線。本端連接的應用程式會結束或中斷與失敗佇列管理程式實例的連線，以確保失敗實例可以重新啟動。任何使用共用連結(這是預設值)的本端連接應用程式，會保留失敗實例的連線，以防止實例重新啟動。如果無法結束本端連接的應用程式，或確保它們在本端佇列管理程式實例失敗時中斷連線，請考量使用隔離的連結。使用隔離連結的本端連接應用程式不會阻止重新啟動本端佇列管理程式實例，即使它們未中斷連線也一樣。

通道及用戶端重新連線

在待命佇列管理程式實例變成作用中之後，通道及用戶端重新連線是還原訊息處理程序的重要部分。

多重實例佇列管理程式實例安裝在具有不同網址的伺服器上。您需要使用所有佇列管理程式實例的連線資訊來配置 IBM WebSphere MQ 通道及用戶端。當待命接管時，用戶端及通道會自動重新連接至位於新網址的新作用中佇列管理程式實例。Java 的 WebSphere MQ 類別不支援自動用戶端重新連接。

此設計不同於高可用性環境(例如 HA-CMP 工作)的方式。HA-CMP 提供叢集的虛擬 IP 位址，並將位址傳送至作用中伺服器。WebSphere MQ 重新連線不會變更或重新遞送 IP 位址。其運作方式是使用您在通道定義及用戶端連線中定義的網址重新連接。作為管理者，您需要定義通道定義中的網址，以及與任何多重實例佇列管理程式之所有實例的用戶端連線。配置多重實例佇列管理程式的網址的最佳方式取決於連線：

佇列管理程式通道

通道的 **CONNAME** 屬性是以逗點區隔的連線名稱清單；例如，**CONNAME('127.0.0.1(1234), 192.0.2.0(4321)')**。會以連線清單中指定的順序來嘗試連線，直到順利建立連線為止。如果沒有連線成功，通道會嘗試重新連接。

叢集通道

通常不需要其他配置，即可讓多重實例佇列管理程式在叢集中運作。

如果佇列管理程式連接至儲存庫佇列管理程式，則儲存庫會探索佇列管理程式的網址。它是指佇列管理程式中 **CLUSRCVR** 通道的 **CONNAME**。在 **TCPIP** 上，如果您省略 **CONNAME** 或將它配置為空白，則佇列管理程式會自動將它設為空白。當待命實例接管時，其 IP 位址會將前一個作用中實例的 IP 位址取代為 **CONNAME**。

必要的話，您可以使用佇列管理程式實例的網址清單來手動配置 **CONNAME**。

用戶端連線

用戶端連線可以使用連線清單或佇列管理程式群組來選取替代連線。如需用戶端重新連線至多重實例佇列管理程式的相關資訊，請參閱第 260 頁的『自動用戶端重新連線』。用戶端需要編譯，才能與 WebSphere MQ 7.0.1 用戶端程式庫或更新版本一起執行。它們必須至少連接至 7.0.1 版佇列管理程式。

發生失效接手時，重新連線需要一些時間。待命佇列管理程式必須完成其啟動。連接至失敗佇列管理程式的用戶端必須偵測連線失敗，並啟動新的用戶端連線。如果新的用戶端連線選取已變成新作用中的待命佇列管理程式，則用戶端會重新連接至相同的佇列管理程式。

如果用戶端在重新連線期間正在進行 MQI 呼叫，則必須容許在呼叫完成之前進行延伸等待。

如果在訊息通道上批次傳送期間發生失敗，則會回復並重新啟動批次。

切換比失效接手更快，且只需要停止佇列管理程式的一個實例並啟動另一個實例。對於只有少數日誌記錄要重播的佇列管理程式，最多可能需要幾秒鐘的切換時間。若要預估失效接手花費的時間，您需要新增偵測失敗所花費的時間。根據網路及檔案系統而定，最多偵測需要 10 秒的時間，而且可能需要數分鐘。

應用程式回復

應用程式回復是在失效接手之後應用程式處理的自動延續。失效接手之後的應用程式回復需要仔細設計。部分應用程式需要注意已發生失效接手。

應用程式回復的目標是讓應用程式僅在短暫延遲的情況下繼續處理。在繼續新的處理程序之前，應用程式必須先回復並重新提交失敗期間正在處理的工作單元。

應用程式回復的問題是失去 WebSphere MQ MQI 用戶端與佇列管理程式之間所共用，且儲存在佇列管理程式中的環境定義。WebSphere MQ MQI 用戶端會還原大部分環境定義，但環境定義的某些部分無法可靠地還原。下列各節說明應用程式回復的部分內容，以及它們如何影響連接至多重實例佇列管理程式之應用程式的回復。

交易式傳訊

從遞送訊息的角度來看，失效接手不會變更 WebSphere MQ 傳訊的持續性內容。如果訊息持續且在工作單元內正確管理，則在失效接手期間不會遺失訊息。

從交易處理的角度來看，在失效接手之後會取消或確定交易。

已回復未確定的交易。失效接手之後，可重新連接的應用程式會收到 MQRC_BACKED_OUT 原因碼，指出交易失敗，且需要發出 MQBACK 來回復交易。然後，它需要重新重新啟動交易。

已確定的交易是已達到兩階段確定之第二階段的交易，或已開始 MQCMIT 的單一階段（僅訊息）交易。

如果佇列管理程式是交易協調程式，且 MQCMIT 已在失敗之前開始其兩階段確定的第二階段，則交易會順利完成。完成作業由佇列管理程式控制，並在佇列管理程式重新執行時繼續執行。在可重新連接的應用程式中，MQCMIT 呼叫正常完成。

在只涉及訊息的單一階段確定中，一旦重新執行，已啟動確定處理的交易會在佇列管理程式的控制下正常完成。在可重新連接的應用程式中，MQCMIT 正常完成。

可重新連接的用戶端可以在佇列管理程式的控制下使用單一階段交易作為交易協調程式。延伸交易式用戶端不支援重新連線。如果在交易式用戶端連接時要求重新連線，則連線會成功，但無法重新連接。連線的行為如同無法重新連接一樣。

應用程式重新啟動或回復

失效接手會岔斷應用程式。失敗之後，應用程式可以從頭重新啟動，也可以在岔斷之後回復處理。後者稱為自動用戶端重新連線。Java 的 WebSphere MQ 類別不支援自動用戶端重新連線。

使用 WebSphere MQ MQI 用戶端應用程式，您可以設定連線選項來自動重新連接用戶端。選項為 MQCNO_RECONNECT 或 MQCNO_RECONNECT_Q_MGR。如果未設定任何選項，則用戶端不會嘗試自動重新連接，且佇列管理程式失敗會將 MQRC_CONNECTION_BROKEN 傳回用戶端。您可以透過發出新的 MQCONN 或 MQCONNX 呼叫，來設計用戶端嘗試並啟動新的連線。

伺服器程式必須重新啟動；當佇列管理程式或伺服器失敗時，它們無法在處理時由佇列管理程式自動重新連接。當多重實例佇列管理程式實例失敗時，通常不會在待命佇列管理程式實例上重新啟動 WebSphere MQ 伺服器程式。

您可以使用兩種方式來自動化 WebSphere MQ 伺服器程式，以在待命伺服器上重新啟動：

1. 將伺服器應用程式包裝成佇列管理程式服務。當待命佇列管理程式重新啟動時，它會重新啟動。

2. 撰寫您自己的失效接手邏輯，例如由待命佇列管理程式實例在啟動時所寫入的失效接手日誌訊息所觸發。然後，應用程式實例需要在啟動之後呼叫 MQCONN 或 MQCONNX，以建立與佇列管理程式的連線。

偵測失效接手

有些應用程式需要知道失效接手，有些則不需要。請考慮這兩個範例。

1. 透過傳訊通道取得或接收訊息的傳訊應用程式通常不需要通道另一端的佇列管理程式在執行中：如果通道另一端的佇列管理程式在待命實例上重新啟動，則不太可能受到影響。
2. WebSphere MQ MQI 用戶端應用程式會處理來自某個佇列的持續訊息輸入，並將持續訊息回應作為單一工作單元的一部分放置在另一個佇列上：如果它透過發出 MQBACK 並重新啟動工作單元，在同步點內處理來自 MQPUT、MQGET 或 MQCMIT 的 MQRC_BACKED_OUT 原因碼，不會遺失任何訊息。此外，應用程式不需要執行任何特殊處理來處理連線失敗。

不過，假設在第二個範例中，應用程式正在瀏覽佇列，以使用 MQGET 選項 MQGMO_MSG_UNDER_CURSOR 來選取要處理的訊息。重新連線會重設瀏覽游標，且 MQGET 呼叫不會傳回正確的訊息。在此範例中，應用程式必須注意已發生失效接手。此外，在針對游標下的訊息發出另一個 MQGET 之前，應用程式必須還原瀏覽游標。

遺失瀏覽游標是應用程式環境定義在重新連線之後如何變更的範例之一。其他案例記錄在 [第 330 頁的『回復自動重新連接的用戶端』](#) 中。

在失效接手之後，您有三個適用於 WebSphere MQ MQI 用戶端應用程式的替代設計型樣。其中只有一個不需要偵測失效接手。

不重新連線

在此型樣中，當連線中斷時，應用程式會停止現行連線上的所有處理程序。若要讓應用程式繼續處理，它必須建立與佇列管理程式的新連線。應用程式完全負責傳送在新連線上繼續處理所需的任何狀態資訊。在失去連線之後重新連接佇列管理程式的現有用戶端應用程式會以這種方式寫入。

失去連線之後，用戶端會從下一個 MQI 呼叫收到原因碼 (例如 MQRC_CONNECTION_BROKEN 或 MQRC_Q_MGR_NOT_AVAILABLE)。應用程式必須捨棄其所有 WebSphere MQ 狀態資訊 (例如佇列控點)，並發出新的 MQCONN 或 MQCONNX 呼叫來建立新的連線，然後重新開啟它需要處理的 WebSphere MQ 物件。

預設 MQI 行為是在失去與佇列管理程式的連線之後，讓佇列管理程式連線控點變成無法使用。預設值相當於在 MQCONNX 上設定 MQCNO_RECONNECT_DISABLED 選項，以防止在失效接手之後重新連線應用程式。

容錯移轉

撰寫應用程式，使它不受失效接手影響。有時謹慎的錯誤處理足以處理失效接手。

重新連線察覺

向佇列管理程式登錄 MQCBT_EVENT_HANDLER 事件處理程式。當用戶端開始嘗試重新連接伺服器時，會使用 MQRC_RECONNECTING 來公佈事件處理程式，在成功重新連線之後，會使用 MQRC_RECONNECTED 來公佈事件處理程式。然後，您可以執行常式來重新建立可預測的狀態，以便用戶端應用程式能夠繼續處理。

回復自動重新連接的用戶端

失效接手是非預期的事件，若要讓自動重新連接的用戶端依設計運作，重新連線的結果必須可預測。

使用交易是將非預期的失敗變成可預測且可靠的回復的主要元素。

在前一節中，提供了 WebSphere MQ MQI 用戶端的範例 [第 330 頁的『2』](#)，使用區域交易來協調 MQGET 和 MQPUT。用戶端發出 MQCMIT 或 MQBACK 呼叫以回應 MQRC_BACKED_OUT 錯誤，然後重新提交已取消的交易。佇列管理程式失敗會導致取消交易，且用戶端應用程式的行為可確保不會遺失任何交易，也不會遺失任何訊息。

並非所有程式狀態都作為交易的一部分進行管理，因此重新連線的結果變得更難以理解。您需要瞭解重新連線如何變更 WebSphere MQ MQI 用戶端的狀態，以便在佇列管理程式失效接手之後，能夠設計您的用戶端應用程式。

您可以決定在設計應用程式時不使用任何特殊失效接手程式碼，以與其他錯誤相同的邏輯來處理重新連線錯誤。或者，您可以選擇辨識重新連線需要特殊錯誤處理，並向 WebSphere MQ 登錄事件處理程式，以執行常式來處理失效接手。常式可以自行處理重新連線處理程序，或設定一個旗標，向主程式執行緒指出當它回復處理程序時，需要執行回復處理程序。

WebSphere MQ MQI 用戶端環境瞭解失效接手本身，並在重新連線之後，透過在用戶端中儲存部分狀態資訊，並代表用戶端應用程式發出其他 MQI 呼叫來還原其 WebSphere MQ 狀態，來盡可能地還原環境定義。例如，會還原在失敗點開啟之物件的控點，並以相同名稱開啟暫時動態佇列。但有些變更是不可避免的，您需要設計來處理這些變更。變更可分為五種：

1. 在應用程式還原一致的新環境定義狀態之前，MQI 呼叫會傳回新的或先前未診斷的錯誤。

例如，在重新連線之前儲存環境定義之後嘗試傳遞環境定義時，收到新錯誤的回覆碼 MQRC_CONTEXT_NOT_AVAILABLE。重新連線之後無法還原環境定義，因為安全環境定義未傳遞至未獲授權的用戶端程式。這樣做會讓惡意應用程式取得安全環境定義。

通常，應用程式會以仔細設計的方式處理一般及可預測的錯誤，並將不常見的錯誤降級至一般錯誤處理程式。錯誤處理程式可能會中斷與 WebSphere MQ 的連線並重新連接，甚至完全停止程式。為了改善連續性，可能需要以不同方式處理一些錯誤。

2. 非持續訊息可能會遺失。
3. 交易已回復。
4. 在同步點之外使用的 MQGET 或 MQPUT 呼叫可能會因訊息遺失而岔斷。
5. 由於 MQI 呼叫中的長時間等待，導致計時導致錯誤。

下列區段列出遺失環境定義的部分詳細資料。

- 除非將非持續訊息放入具有 NPMCLASS (HIGH) 選項的佇列，且佇列管理程式失敗不會岔斷關閉時儲存非持續訊息的選項，否則會捨棄非持續訊息。
- 當連線中斷時，會失去不可延續的訂閱。重新連線時，會重新建立它。請考量使用可延續訂閱。
- get-wait 間隔會重新計算；如果超出其限制，則會傳回 MQRC_NO_MSG_AVAILABLE。同樣地，會重新計算訂閱期限，以提供相同的整體到期時間。
- 瀏覽游標在佇列中的位置會遺失；通常會在第一則訊息之前重新建立。
 - 指定 MQGMO_BROWSE_MSG_UNDER_CURSOR 或 MQGMO_MSG_UNDER_CURSOR 的 MQGET 呼叫失敗，原因碼為 MQRC_NO_MSG_AVAILABLE。
 - 已解除鎖定因瀏覽而鎖定的訊息。
 - 具有控點範圍的瀏覽標示訊息會取消標示，且可以重新瀏覽。
 - 在大部分情況下，會取消標示協同瀏覽標示的訊息。
- 安全環境定義已遺失。嘗試使用已儲存的訊息環境定義，例如放置具有 MQPMO_PASS_ALL_CONTEXT 的訊息會失敗，並產生 MQRC_CONTEXT_NOT_AVAILABLE。
- 遺失訊息記號。MQGET 使用訊息記號會傳回原因碼 MQRC_NO_MSG_AVAILABLE。
註：MsgId 和 CorrelId(因為它們是訊息的一部分) 會在失效接手期間隨訊息一起保留，因此 MQGET 使用 MsgId 或 CorrelId 如預期運作。
- 在未確定交易中的同步點下放置在佇列上的訊息不再可用。
- 以邏輯順序處理訊息或以訊息群組處理訊息，會在重新連線之後產生回覆碼 MQRC_RECONNECT_INCOMPATIBLE。
- MQI 呼叫可能會傳回 MQRC_RECONNECT_FAILED，而不是用戶端今天通常會收到的較一般的 MQRC_CONNECTION_BROKEN。
- 如果 WebSphere MQ MQI 用戶端不知道訊息是否已順利遞送至佇列管理程式，則在 MQPUT 呼叫外部同步點期間會傳回 MQRC_CALL_INTERRUPTED。在 MQCMIT 期間重新連線的行為類似。
- 在成功重新連接之後，如果 WebSphere MQ MQI 用戶端未收到來自佇列管理程式的任何回應，會傳回 MQRC_CALL_INTERRUPTED。
 - 在同步點外部使用 MQPUT 呼叫來遞送持續訊息。
 - 使用同步點外部的 MQPUT1 呼叫來遞送持續訊息或具有預設持續性的訊息。

- 使用 MQCMIT 呼叫來確定交易。只有在順利重新連接之後，才會傳回回應。
- 通道會重新啟動為新的實例 (它們也可能是不同的通道)，因此不會保留通道結束狀態。
- 在回復已開啟暫時動態佇列的可重新連接用戶端的過程中，會還原暫時動態佇列。不會還原暫時動態佇列上的訊息，但已開啟佇列或已記住佇列名稱的應用程式可以繼續處理。

如果佇列正由建立它的應用程式以外的應用程式使用，則還原它的速度可能不夠快，無法在下次參照它時出現。例如，如果用戶端將暫時動態佇列建立為回覆目的地佇列，且通道要將回覆訊息放置在佇列上，則可能無法及時回復佇列。在此情況下，通道通常會將回覆目的地訊息放置在無法傳送的郵件佇列上。

如果可重新連接的用戶端應用程式依名稱開啟暫時動態佇列 (因為另一個應用程式已建立它)，則當重新連線發生時，WebSphere MQ MQI 用戶端無法重建暫時動態佇列，因為它沒有可從中建立它的模型。在 MQI 中，只有一個應用程式可以依模型開啟暫時動態佇列。其他想要使用暫時動態佇列的應用程式必須使用 MQPUT1 或伺服器連結，或能夠在失敗時重試重新連線。

只能將非持續訊息放置在暫時動態佇列中，在失效接手期間會遺失這些訊息；在重新連線期間使用 MQPUT1 將訊息放置在暫時動態佇列中，則會遺失這些訊息。如果在 MQPUT1 期間發生失效接手，則雖然 MQPUT1 成功，但可能不會放置訊息。此問題的暫行解決方法之一是使用永久動態佇列。任何伺服器連結應用程式都可以依名稱開啟暫時動態佇列，因為它無法重新連接。

資料回復及高可用性

使用多重實例佇列管理程式的高可用性解決方案必須包含在儲存體故障之後回復資料的機制。

多重實例佇列管理程式會增加佇列管理程式處理程序的可用性，但不會增加佇列管理程式用來儲存訊息及其他資訊之其他元件 (例如檔案系統) 的可用性。

讓資料具有高可用性的方法之一，就是使用網路上具復原力的資料儲存體。您可以使用網路檔案系統及具復原力的資料儲存體來建置自己的解決方案，也可以購買整合式解決方案。如果您想要結合備援與災難回復，則可以使用非同步磁碟抄寫 (允許磁碟抄寫超過數十或數百公里)。

您可以配置不同的 WebSphere MQ 目錄對映至儲存媒體的方式，以充分利用媒體。對於多重實例佇列管理程式，有兩種類型的 WebSphere MQ 目錄和檔案之間的重要差異。

必須在佇列管理程式實例之間共用的目錄。

必須在不同佇列管理程式實例之間共用的資訊位於兩個目錄中: qmgrs 和 logs 目錄。目錄必須位於共用網路檔案系統上。建議您使用可提供連續高可用性且卓越效能的儲存媒體，因為在建立及刪除訊息時，資料會持續變更。

非必須在佇列管理程式實例之間共用的目錄及檔案。

部分其他目錄不必在佇列管理程式的不同實例之間共用，而且除了使用鏡映檔案系統之外，還可以透過其他方式快速還原。

- WebSphere MQ 執行檔和 tools 目錄。透過重新安裝或從已備份的檔案保存檔備份及還原來取代。
- 針對整體安裝所修改的配置資訊。配置資訊是由 WebSphere MQ (例如 Windows、UNIX and Linux 系統上的 mqsc.ini 檔案) 或您自己配置管理的一部分 (例如 MQSC 配置 Script) 所管理。使用檔案保存檔來備份及還原。
- 安裝層面輸出，例如追蹤資料、錯誤日誌及 FFDC 檔案。這些檔案儲存在預設資料目錄的 errors 和 trace 子目錄中。UNIX and Linux 系統上的預設資料目錄是 /var/mqm。在 Windows 上，預設資料目錄是 WebSphere MQ 安裝目錄。

您也可以使用備份佇列管理程式，利用線性記載來定期進行多重實例佇列管理程式的媒體備份。備份佇列管理程式不會提供與鏡映檔案系統一樣快的回復，且不會回復自前次備份以來所做的變更。相較於在本地化儲存體失敗之後回復佇列管理程式，備份佇列管理程式機制更適合在離站災難回復實務範例中使用。

結合 IBM WebSphere MQ 可用性解決方案

應用程式正在使用其他 IBM WebSphere MQ 功能來改善可用性。多重實例佇列管理程式補充其他高可用性功能。

IBM WebSphere MQ 叢集會增加佇列可用性

您可以透過建立叢集佇列的多個定義來增加佇列可用性；叢集中每一個管理程式最多可以有一個佇列。

假設叢集的成員失敗，然後將新訊息傳送至叢集佇列。除非訊息必須移至失敗的佇列管理程式，否則訊息會傳送至叢集中具有佇列定義的另一個執行中佇列管理程式。

雖然叢集會大幅增加可用性，但有兩個相關的失敗實務範例會導致訊息延遲。使用多重實例佇列管理程式來建置叢集，可減少訊息延遲的機會。

孤立訊息

如果叢集中的佇列管理程式失敗，則無法再將可遞送至叢集中其他佇列管理程式的訊息遞送至失敗的佇列管理程式。在重新啟動失敗的佇列管理程式之前，會保留已傳送的訊息。

親緣性

親緣性是用來說明在兩個不同的計算之間共用的資訊的術語。例如，傳送要求訊息至伺服器的應用程式與預期處理回覆的相同應用程式之間存在親緣性。另一個範例是一連串訊息，每一個訊息的處理取決於先前的訊息。

如果您將訊息傳送至叢集佇列，則需要考量親緣性。您需要將連續訊息傳送至相同的佇列管理程式，還是每一則訊息都可以傳送至叢集的任何成員？

如果您確實需要將訊息傳送至叢集中的相同佇列管理程式，但失敗，則訊息會在傳送端的傳輸佇列中等待，直到失敗的叢集佇列管理程式再次執行為止。

如果叢集已配置多重實例佇列管理程式，則在待命接管期間，等待失敗佇列管理程式重新啟動的延遲限制為一分鐘左右的順序。當待命資料庫在執行中，孤立的訊息會回復處理，啟動新啟動佇列管理程式實例的通道，且在傳輸佇列中等待的訊息會開始流動。

若要配置叢集以克服失敗佇列管理程式所延遲的訊息，一種可能的方法是將兩個不同的佇列管理程式部署至叢集中的每一部伺服器，並將其中一部部署為作用中，而另一部部署為不同佇列管理程式的待命實例。這是主動-待命配置，它會增加叢集的可用性。

除了具有減少管理及增加可調整性的好處之外，叢集還會繼續提供額外的可用性元素，以補充多重實例佇列管理程式。叢集可防範其他類型的失敗，這些失敗會同時影響佇列管理程式的作用中及待命實例。

不中斷服務

叢集提供不中斷的服務。叢集收到的新訊息會傳送至作用中佇列管理程式進行處理。請勿依賴多重實例佇列管理程式來提供不中斷的服務，因為待命佇列管理程式需要一些時間來偵測失敗並完成其啟動、重新連接其通道，以及重新提交失敗的訊息批次。

本地化中斷

作用中、待命及檔案系統伺服器之間的距離有實際限制，因為它們需要以毫秒速度進行互動，以提供可接受的效能。

叢集佇列管理程式需要許多秒的互動速度，而且在地理上可以分散在世界任何地方。

作業錯誤

透過使用兩種不同的機制來增加可用性，您可以減少作業錯誤 (例如人為錯誤) 危及可用性工作的機會。

佇列共用群組會增加訊息處理可用性

佇列共用群組 (僅在 z/OS 上提供) 容許佇列管理程式群組共用服務佇列。如果一個佇列管理程式失敗，其他佇列管理程式會繼續處理佇列上的所有訊息。在 z/OS 上不支援多重實例佇列管理程式，只會在更廣泛的傳訊架構中補充佇列共用群組。

WebSphere MQ 用戶端增加應用程式可用性

WebSphere MQ MQI 用戶端程式可以根據佇列管理程式可用性、連線加權及親緣性，來連接至佇列管理程式群組中的不同佇列管理程式。透過在與執行佇列管理程式的機器不同的機器上執行應用程式，您可以改善解決方案的整體可用性，只要有方法在佇列管理程式實例所連接的失敗時重新連接應用程式。

佇列管理程式群組用來增加用戶端可用性，方法是從已停止的佇列管理程式中取消連結用戶端，並在佇列管理程式群組 (例如 IP Sprayer) 之間平衡用戶端連線的負載。用戶端應用程式必須與失敗的佇列管理程式沒有親緣性 (例如與特定佇列的相依關係)，否則無法回復處理。

自動用戶端重新連線及多重實例佇列管理程式會解決一些親緣性問題，以增加用戶端可用性。Java 的 WebSphere MQ 類別不支援自動用戶端重新連接。

您可以設定 MQCNO 選項 MQCNO_RECONNECT_Q_MGR，以強制用戶端重新連接至相同的佇列管理程式：

1. 如果先前連接的單一實例佇列管理程式不在執行中，則會重試連線，直到佇列管理程式再次執行為止。
2. 如果佇列管理程式配置為多重實例佇列管理程式，則用戶端會重新連接至作用中的任何實例。

透過自動重新連接至相同的佇列管理程式，會還原佇列管理程式代表用戶端保留的許多狀態資訊，例如已開啟的佇列及訂閱的主題。如果用戶端已開啟動態回覆目的地佇列來接收要求的回覆，則也會還原回覆目的地佇列的連線。

確定訊息未遺失 (記載)

WebSphere MQ 會記載從佇列管理程式失敗回復所需的所有資訊。

WebSphere MQ 會將佇列管理程式所控制資料的所有重要變更記錄在回復日誌中。

這包括建立及刪除物件、持續訊息更新、交易狀態、物件屬性的變更，以及通道活動。日誌包含您需要的資訊，可透過下列方式回復訊息佇列的所有更新項目：

- 保留佇列管理程式變更的記錄
- 保留佇列更新記錄以供重新啟動處理程序使用
- 可讓您在硬體或軟體故障之後還原資料

不過，WebSphere MQ 也依賴管理其檔案的磁碟系統。如果磁碟系統本身不可靠，則資訊 (包括日誌資訊) 仍可能遺失。

日誌的外觀

日誌由主要和次要檔案以及控制檔組成。您可以定義日誌檔的數目和大小，以及它們在檔案系統中的儲存位置。

WebSphere MQ 日誌由兩個元件組成：

1. 一或多個日誌資料檔案。
2. 日誌控制檔

日誌資料的檔案也稱為日誌範圍。

有一些日誌檔包含正在記錄的資料。您可以定義數目和大小 (如第 350 頁的『[變更 IBM WebSphere MQ 及佇列管理程式配置資訊](#)』中所說明)，或採用三個檔案的系統預設值。

在 WebSphere MQ for Windows 中，三個檔案各預設為 1 MB。在 WebSphere MQ for UNIX and Linux 系統中，三個檔案各預設為 4 MB。

當您建立佇列管理程式時，您定義的日誌檔數目是配置的主要日誌檔數目。如果您未指定數字，則會使用預設值。

在 WebSphere MQ for Windows 中，如果您未變更日誌路徑，則會在下列目錄下建立日誌檔：

```
C:\Program Files\IBM\WebSphere MQ\log\<QMgrName>
```

在 WebSphere MQ for UNIX and Linux 系統中，如果您未變更日誌路徑，則會在下列目錄下建立日誌檔：

```
/var/mqm/log/<QMgrName>
```

WebSphere MQ 會從這些主要日誌檔開始，但如果主要日誌空間不足，它會配置次要日誌檔。它會動態執行此動作，並在日誌空間需求減少時移除它們。依預設，最多可以配置兩個次要日誌檔。您可以變更此預設配置，如第 350 頁的『[變更 IBM WebSphere MQ 及佇列管理程式配置資訊](#)』中所述。

日誌控制檔

日誌控制檔包含控制日誌檔使用所需的資訊，例如其大小及位置，以及下一個可用檔案的名稱。

日誌控制檔僅供內部佇列管理程式使用。

佇列管理程式會在日誌控制檔中保留與回復日誌狀態相關聯的控制資料，且您不得修改日誌控制檔的內容。

註: 請確定啟動佇列管理程式時所建立的日誌夠大，足以容納應用程式將處理的訊息大小和數量。您可能需要變更預設日誌號碼及大小，以符合您的需求。如需相關資訊，請參閱第 337 頁的『計算日誌的大小』。

記載類型

在 WebSphere MQ 中，記載所需的檔案數目取決於檔案大小、您收到的訊息數目及訊息長度。有兩種方式可維護佇列管理程式活動的記錄：循環式記載和線性記載。

循環式記載

如果您只想要重新啟動回復，請使用循環記載，並使用日誌來回復系統停止時進行中的交易。

循環式記載會將所有重新啟動資料保留在日誌檔的循環中。記載會填入循環中的第一個檔案，然後移至下一個，以此類推，直到所有檔案都已滿為止。然後會回到循環中的第一個檔案，重新開始。只要產品在使用中，這個作業就會繼續執行，好處是您永遠都不會在日誌檔以外的檔案中執行記載。

WebSphere MQ 會保留重新啟動佇列管理程式所需的日誌項目，而不會遺失資料，直到不再需要它們來確保佇列管理程式資料回復為止。第 335 頁的『使用檢查點檢查作業來確保完整回復』中說明釋放日誌檔以供重複使用的機制。

線性記載 (linear logging)

如果您要重新啟動回復及媒體回復 (透過重播日誌內容來重新建立遺失或損壞的資料)，請使用線性記載。線性記載會以連續的檔案順序來保留日誌資料。空間不會重複使用，因此您可以一直擷取記載在尚未刪除之任何日誌範圍中的任何記錄。

由於磁碟空間有限，您可能需要考慮某種保存形式。這是一項管理作業，用來管理日誌的磁碟空間，必要時可重複使用或延伸現有空間。

與線性記載搭配使用的日誌檔數目可能非常大，視您的訊息流程及佇列管理程式的經歷時間而定。不過，有一些檔案被認為作用中。作用中檔案包含重新啟動佇列管理程式所需的日誌項目。作用中日誌檔統稱為作用中日誌。現行日誌檔數目通常小於配置檔中定義的主要日誌檔數目。(如需定義數字的相關資訊，請參閱第 337 頁的『計算日誌的大小』。)

控制日誌檔是否稱為作用中的關鍵事件是檢查點。WebSphere MQ 檢查點是回復日誌與物件檔之間的一致點。檢查點決定執行重新啟動回復所需的日誌檔集。重新啟動回復不需要非作用中的日誌檔，稱為非作用中。在某些情況下，媒體回復需要非作用中日誌檔。(如需檢查點檢查作業的進一步相關資訊，請參閱第 335 頁的『使用檢查點檢查作業來確保完整回復』。)

可以保存非作用中日誌檔，因為重新啟動回復不需要它們。媒體回復不需要的非作用中日誌檔可以視為多餘的日誌檔。如果您的作業不再感興趣，您可以刪除多餘的日誌檔。如需日誌檔處置的進一步相關資訊，請參閱第 338 頁的『管理日誌』。

如果在第二個或更新版本主要日誌檔中記錄新的檢查點，則第一個檔案可能會變成非作用中，且新的主要檔案會格式化並新增至主要儲存區的結尾，以還原可用於記載的主要檔案數目。以此方式，可以將主要日誌檔儲存區視為日誌檔延伸清單中的現行檔案集。同樣地，根據作業需求來管理非作用中檔案是一項管理作業。

雖然已針對線性記載定義次要日誌檔，但在一般作業中不會使用它們。如果發生狀況，可能是因為長期存在的交易，則無法從作用中儲存區釋放檔案，因為重新啟動仍可能需要該檔案，次要檔案會格式化並新增至作用中日誌檔儲存區。

如果可用的次要檔案數已用完，則會拒絕需要日誌活動之大部分進一步作業的要求，並傳回 MQRC_RESOURCE_PROBLEM 回覆碼給應用程式。

假設沒有硬體故障，這兩種類型的記載都可以應付非預期的電源流失。

使用檢查點檢查作業來確保完整回復

檢查點會同步化佇列管理程式資料和日誌檔，並標示可從中捨棄日誌記錄的一致性點。頻繁檢查點檢查作業可讓回復更快速。

訊息佇列的持續更新分兩個階段進行。首先，會將代表更新的記錄寫入日誌，然後更新佇列檔。因此，日誌檔會變成比佇列檔更最新的。為了確保從一致點開始重新啟動處理程序，WebSphere MQ 會使用檢查

點。檢查點是指日誌中所說明的記錄與佇列中的記錄相同的時間點。檢查點本身包含重新啟動佇列管理程式所需的一系列日誌記錄; 例如, 檢查點時所有作用中交易 (工作單元) 的狀態。

WebSphere MQ 會自動產生檢查點。當佇列管理程式啟動時、關閉時、記載空間不足時, 以及記載每 10000 個作業之後, 都會採用它們。

當佇列處理進一步訊息時, 檢查點記錄會變成與佇列的現行狀態不一致。

當 WebSphere MQ 重新啟動時, 它會在日誌中尋找最新的檢查點記錄。此資訊保留在每個檢查點結束時更新的檢查點檔案中。檢查點記錄代表日誌與資料之間的一致點。會向前重播自檢查點以來所執行的所有作業。這稱為重播階段。重播階段會使佇列回到系統失效或關機之前的邏輯狀態。在重播階段期間, 會建立系統失效或關機時進行中的交易清單。發出 AMQ7229 及 AMQ7230 訊息, 以指出重播階段的進度。

為了瞭解要取消或確定哪些作業, WebSphere MQ 會存取與進行中交易相關聯的每一個作用中日誌記錄。這稱為回復階段。發出 AMQ7231、AMQ7232 及 AMQ7234 訊息, 以指出回復階段的進度。

在回復階段期間存取所有必要的日誌記錄之後, 會依序解析每一個作用中的交易, 且會取消或確定與交易相關聯的每一個作業。這稱為解決階段。發出 AMQ7233 訊息以指出解決階段的進度。

WebSphere MQ 會維護指向日誌標頭和尾端的內部指標。它會將標頭指標移至與回復訊息資料一致的最近檢查點。

檢查點是用來使回復更有效率, 以及控制主要和次要日誌檔的重複使用。

在第 336 頁的圖 69 中, WebSphere MQ 不再需要最新檢查點 (檢查點 2) 之前的所有記錄。可以從檢查點資訊及任何後續日誌項目回復佇列。對於循環式記載, 可在檢查點之前重複使用任何釋放的檔案。對於線性日誌, 已釋放的日誌檔不再需要針對正常作業進行存取並變成非作用中。在此範例中, 佇列標頭指標會移至最新檢查點 Checkpoint 2 的點, 然後它會變成新的佇列標頭 Head 2。現在可以重複使用日誌檔 1。

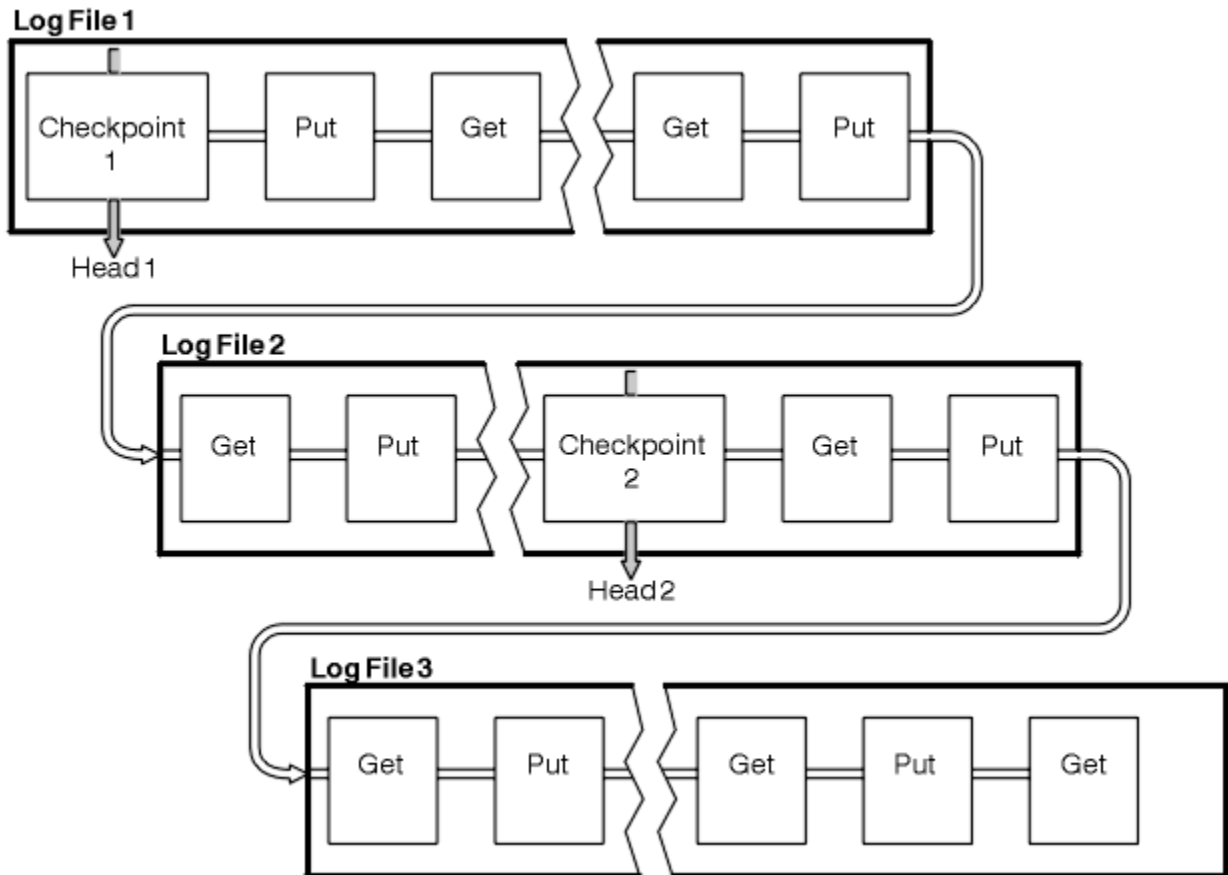


圖 69: 正在進行檢查點檢查作業

使用長時間執行的交易進行檢查點檢查作業

長時間執行的交易如何影響日誌檔的重複使用。

第 337 頁的圖 70 顯示長時間執行的交易如何影響日誌檔的重複使用。在此範例中，長時間執行的交易在顯示第一個檢查點之後進入日誌，顯示為 LR 1。直到第三個檢查點之後，交易才會完成 (在 LR 2 點)。從 LR 1 開始的所有日誌資訊都會保留下來，以便在必要時回復該交易，直到完成為止。

在長時間執行的交易完成之後，在 LR 2，日誌的標頭會移至最新記載的檢查點第 3 點。不再需要在 Checkpoint 3 (Head 2) 之前包含日誌記錄的檔案。如果您使用循環式記載，則可以重複使用空間。

如果在長時間執行交易完成之前，主要日誌檔已完全已滿，則會使用次要日誌檔來避免日誌已滿。

當移動日誌標頭且您使用循環式記載時，主要日誌檔可能變成適合重複使用，且日誌程式在填入現行檔案之後，會重複使用它可用的第一個主要檔案。如果您使用線性記載，則日誌標頭仍會向下移動作用中儲存區，且第一個檔案會變成非作用中。新的主要檔案會格式化並新增至儲存區底端，以準備未來記載活動。

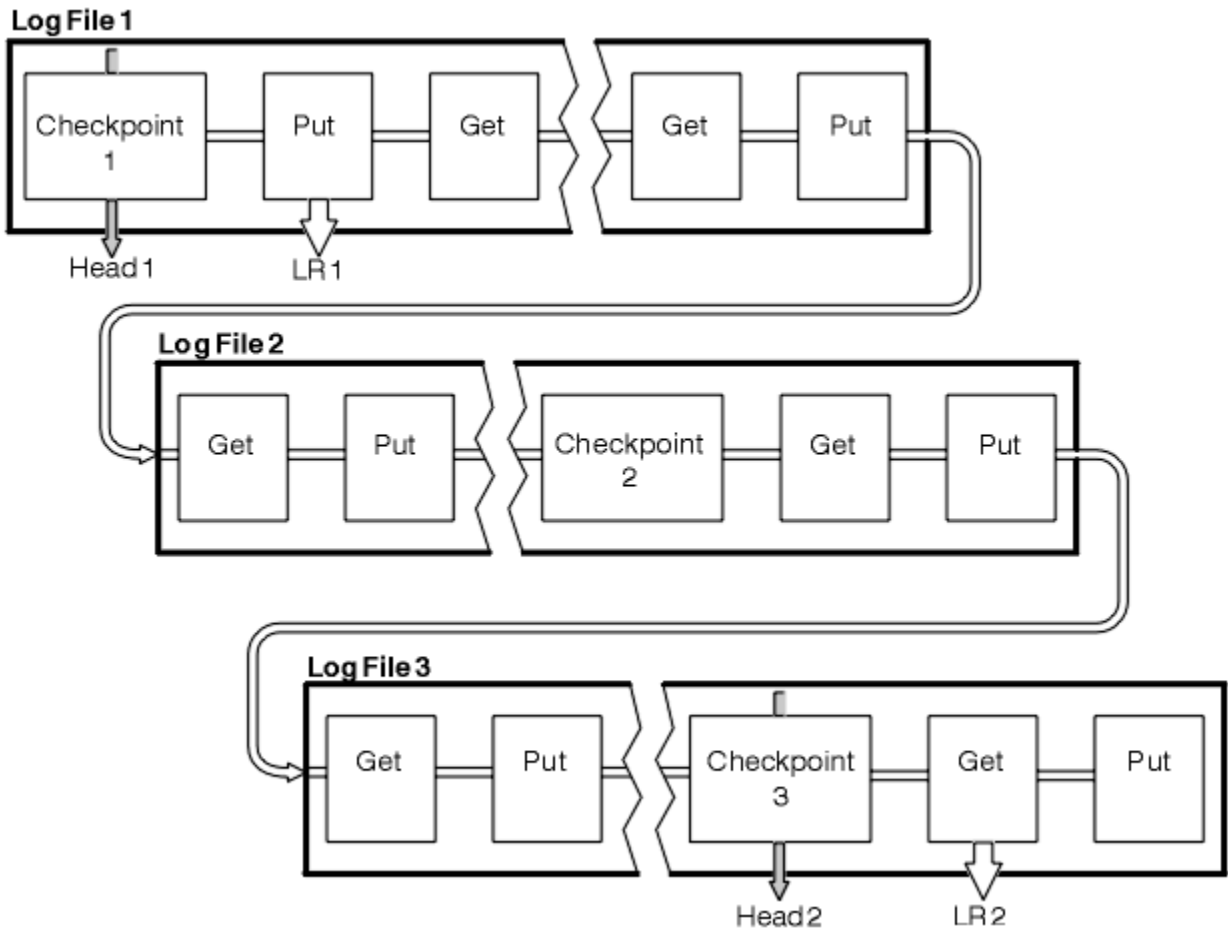


圖 70: 使用長時間執行的交易進行檢查點檢查作業

計算日誌的大小

預估佇列管理程式所需的日誌大小。

在決定佇列管理程式使用循環或線性記載之後，您需要預估佇列管理程式所需的日誌大小。日誌的大小由下列日誌配置參數決定：

LogFilePages

每一個主要及次要日誌檔的大小 (以 4K 頁為單位)

LogPrimaryFiles

預先配置的主要日誌檔數目

LogSecondaryFiles

主要日誌檔已滿時可以建立使用的次要日誌檔數目

第 338 頁的表 31 顯示佇列管理程式針對各種作業所記載的資料量。大部分佇列管理程式作業都需要最少的日誌空間量。不過，當持續訊息放入佇列時，**所有** 訊息資料必須寫入日誌中，才能回復訊息。日誌的大小通常取決於佇列管理程式需要處理的持續訊息數目及大小。

表 31: 日誌項目大小 (所有值都是近似值)	
作業	大小
放置持續訊息	750 位元組 + 訊息長度 如果訊息很大，則會分成 261844 個位元組的區段，每一個區段會額外新增 300 個位元組。
取得訊息	260 位元組
同步點，確定	750 位元組
同步點，回復	1000 個位元組 + 每一個要回復的取得或放置 12 個位元組
建立物件	1500 位元組
刪除物件	300 位元組
變更屬性	1024 位元組
記錄媒體影像	800 位元組 + 影像 映像檔分成 260 000 個位元組的區段，每一個區段會額外新增 300 個位元組。
檢查站	750 位元組 + 每一個作用中工作單元的 200 位元組 + 每一個叢集傳送端通道的 380 位元組 (如果每個佇列管理程式使用多個叢集傳輸佇列)。 基於效能原因，可能會針對任何緩衝的未確定的放置或取得項目記載其他資料。 如果您有叢集傳送端通道，則在每一個檢查點，會在每個叢集傳送端通道的日誌中額外寫入 380 個位元組。

註:

1. 您可以在每次佇列管理程式啟動時變更主要及次要日誌檔的數目。
2. 您無法變更日誌檔大小; 您必須在建立佇列管理程式 **之前** 決定它。
3. 主要日誌檔數目及日誌檔大小決定建立佇列管理程式時預先配置的日誌空間量。
4. 在 UNIX and Linux 系統上，主要及次要日誌檔的總數不能超過 511，在 Windows 上，則不能超過 255，在長時間執行的交易存在時，這會限制佇列管理程式可用來重新啟動回復的日誌空間量上限。佇列管理程式在進行媒體回復時可能需要的日誌空間量未共用此限制。
5. 當使用循環記載時，佇列管理程式會重複使用主要日誌空間。這表示佇列管理程式的日誌可以小於您預估佇列管理程式需要記載的資料量。當日誌檔已滿，且順序中的下一個主要日誌檔無法使用時，佇列管理程式將最多配置一個次要日誌檔。
6. 在檢查點期間可重複使用主要日誌檔。在採取檢查點之前，佇列管理程式會考量主要及次要日誌空間，因為日誌空間量偏低。

如果您未定義比次要日誌檔更多的主要日誌檔，則在取得檢查點之前，佇列管理程式可能會配置次要日誌檔。這會使主要日誌檔可供重複使用。

管理日誌

日誌幾乎是自我管理，但有時需要設法解決空間問題。

一段時間後，部分寫入的日誌記錄會變成不需要重新啟動佇列管理程式。如果您使用循環式記載，佇列管理程式會收回日誌檔中的釋放空間。使用者不會看到此活動，而且您通常不會看到已使用的磁碟空間量減少，因為配置的空間會快速重複使用。

在日誌記錄中，只有那些自前次完成檢查點啟動以來寫入的記錄，以及那些由任何作用中交易寫入的記錄，才需要重新啟動佇列管理程式。因此，如果長時間未使用檢查點，或長時間執行的交易會在很久之前寫入日誌記錄，則日誌可能會填滿。佇列管理程式通常會嘗試採取足夠的檢查點，以避免第一個問題。

當長時間執行的交易填滿日誌時，嘗試寫入日誌記錄會失敗，且部分 MQI 呼叫會傳回 MQRC_RESOURCE_PROBLEM。(保留空間以確定或回復所有進行中交易，因此 MQCMIT 或 MQBACK 不應失敗。)

佇列管理程式會回復耗用太多日誌空間的交易。以這種方式回復具有交易的應用程式無法在相同交易下指定同步點來執行後續的 MQPUT 或 MQGET 作業。嘗試在此狀態下放置或取得同步點下的訊息會傳回 MQRC_BACKED_OUT。然後，應用程式可以發出 MQCMIT(會傳回 MQRC_BACKED_OUT) 或 MQBACK，並啟動新的交易。當回復耗用太多日誌空間的交易時，會釋放其日誌空間，且佇列管理程式會繼續正常運作。

如果日誌填滿，則會發出 AMQ7463 訊息。此外，如果日誌因長時間執行的交易導致無法釋放空間而填滿，則會發出 AMQ7465 訊息。

最後，如果將記錄寫入日誌的速度超過日誌可以處理的速度，則會發出 AMQ7466 訊息。如果您看到此訊息，請增加日誌檔數目，或減少佇列管理程式正在處理的資料量。

磁碟已滿時會發生什麼情況

佇列管理程式記載元件可以處理已滿的磁碟，以及已滿的日誌檔。如果包含日誌的磁碟已滿，佇列管理程式會發出 AMQ6708 訊息，並取得錯誤記錄。

日誌檔會以其大小上限建立，而不是在日誌記錄寫入時延伸。這表示只有在 WebSphere MQ 建立新檔案時，它才會用盡磁碟空間；在日誌中寫入記錄時，它不會用盡空間。WebSphere MQ 一律知道現有日誌檔中可用的空間量，並相應地管理檔案內的空間。

如果您填入包含日誌檔的磁碟機，則可能可以釋放部分磁碟空間。如果您使用線性日誌，日誌目錄中可能有一些非作用中的日誌檔，您可以將這些檔案複製到另一個磁碟機或裝置。如果您仍用盡空間，請檢查佇列管理程式配置檔中的日誌配置是否正確。您可以減少主要或次要日誌檔的數目，使日誌不會超出可用空間。您無法變更現有佇列管理程式的日誌檔大小。佇列管理程式會假設所有日誌檔的大小都相同。

管理日誌檔

請為日誌檔配置足夠的空間。對於線性記載，您可以刪除不再需要的舊日誌檔。

如果您使用循環式記載，當您配置系統時，請確定有足夠的空間來保留日誌檔(請參閱第 357 頁的『IBM WebSphere MQ 的日誌預設值』和第 363 頁的『佇列管理程式日誌』)。日誌所使用的磁碟空間量不會增加超過已配置的大小，包括必要時要建立的次要檔案空間。

如果您使用線性日誌，則會在記載資料時持續新增日誌檔，且使用的磁碟空間量會隨著時間而增加。如果記載資料的速率偏高，則新日誌檔會快速使用磁碟空間。

一段時間後，線性日誌的較舊日誌檔不再需要重新啟動佇列管理程式或執行任何損壞物件的媒體回復。下列是判斷仍需要哪些日誌檔的方法：

日誌程式事件訊息

啟用時，當佇列管理程式開始將日誌記錄寫入新的日誌檔時，會產生日誌程式事件訊息。日誌程式事件訊息的內容指定佇列管理程式重新啟動及媒體回復仍需要的日誌檔。如需日誌程式事件訊息的相關資訊，請參閱 [日誌程式事件](#)

佇列管理程式狀態

執行 MQSC 指令 DISPLAY QMSTATUS 或 PCF 指令 Inquire Queue Manager Status 會傳回佇列管理程式資訊，包括必要日誌檔的詳細資料。如需 MQSC 指令的相關資訊，請參閱 [Script \(MQSC\) 指令](#)，以及 PCF 指令的相關資訊，請參閱 [自動化管理作業](#)。

佇列管理程式訊息

佇列管理程式會定期發出一對訊息，以指出需要哪些日誌檔：

- 訊息 AMQ7467 會提供重新啟動佇列管理程式所需的最舊日誌檔名稱。在佇列管理程式重新啟動期間，必須提供此日誌檔及所有較新的日誌檔。
- 訊息 AMQ7468 會提供媒體回復所需的最舊日誌檔名稱。

只有佇列管理程式重新啟動所需的日誌檔 (現行日誌檔) 才需要在線上。非作用中日誌檔可以複製到保存媒體 (例如用於災難回復的磁帶)，並從日誌目錄中移除。媒體回復不需要的非作用中日誌檔可以視為多餘的日誌檔。如果您的作業不再感興趣，您可以刪除多餘的日誌檔。

若要判定「較舊」及「較新」日誌檔，請使用日誌檔號碼，而非檔案系統所套用的修改時間。

如果找不到所需的任何日誌檔，則會發出操作員訊息 AMQ6767。請讓佇列管理程式可以使用日誌檔及所有後續日誌檔，然後重試作業。

註: 執行媒體回復時，所有必要的日誌檔必須同時存在於日誌檔目錄中。請確定您取得您可能想要回復之任何物件的一般媒體映像檔，以避免用盡磁碟空間來保留所有必要的日誌檔。若要取得佇列管理程式中所有物件的媒體影像，請執行 **rcdmqimg** 指令，如下列範例所示：

在 Windows 上

```
rcdmqimg -m QMNAME -t all *
```

在 UNIX and Linux 上

```
rcdmqimg -m QMNAME -t all "*"
```

執行 **rcdmqimg** 會向前移動媒體日誌序號 (LSN)。如需日誌序號的進一步詳細資料，請參閱第 342 頁的『使用 **dmpmqlog** 指令傾出日誌內容』。**rcdmqimg** 不會自動執行，因此必須手動執行，或從您所建立的自動作業執行。如需此指令的相關資訊，請參閱 **rcdmqimg** 及 **dmpmqlog**。

註: 在執行 **rcdmqimg** 指令時，也可以發出 AMQ7467 及 AMQ7468 訊息。

判斷多餘的日誌檔

管理線性日誌檔時，請務必確定可以刪除或保存哪些檔案。此資訊將協助您做出此決策。

請勿使用檔案系統的修改時間來判斷「較舊」的日誌檔。僅使用日誌檔號碼。佇列管理程式使用日誌檔遵循複雜的規則，包括在需要日誌檔之前預先配置及格式化日誌檔。如果您嘗試使用這些時間來決定相對經歷時間，則可能會看到日誌檔具有會誤導的修改時間。

若要判斷重新啟動佇列管理程式所需的最舊日誌檔，請發出指令 **DISPLAY QMSTATUS RECLOG**。

若要判斷執行媒體回復所需的最舊日誌檔，請發出指令 **DISPLAY QMSTATUS MEDIALOG**。

一般而言，較低的日誌檔數目意味著較舊的日誌。除非您有非常高的日誌檔週轉率，即 10 年每天 3000 個日誌檔的訂單，否則您不需要滿足 9 999 999 的數字折返。在此情況下，您可以保存數字小於 RECLOG 值的任何日誌檔，也可以刪除數字小於 RECLOG 及 MEDIALOG 值的任何日誌檔。

不過，如果您有非常高的日誌檔週轉率，或想要有信心應付一般情況，則通常可以使用下列演算法：

Let S == restart log file number

(來自 **DISPLAY QMSTATUS RECLOG**)。

Let M == 媒體回復日誌檔號碼

(從 **DISPLAY QMSTATUS MEDIALOG**)。

讓 L == 具有刪除或保存資格的日誌檔號碼
需要確定

```
function minlog (a , b) {  
  if (abs (a - b) < 5000000)  
    return min (a , b); # Not wrapped.  
  else  
    return max (a , b); # Wrapped.}
```

如果發生下列情況，則可以刪除日誌檔 L:

(L != S && L != M && minlog (L, minlog (S, M)) == L).

如果發生下列情況，則可以保存日誌檔 L:

(L != S && minlog (L, S) == L).

日誌檔位置

選擇日誌檔的位置時，請記住，如果 WebSphere MQ 由於缺少磁碟空間而無法格式化新日誌，則作業會受到嚴重影響。

如果您使用循環日誌，請確定磁碟機有足夠空間至少容納已配置的主要日誌檔。此外，請為至少一個次要日誌檔保留空間，如果日誌必須成長，則需要此空間。

如果您使用線性日誌，請容許大量的空間；日誌所耗用的空間會隨著資料的記載而持續增加。

最好將日誌檔放在與佇列管理程式資料不同的磁碟機上。這在效能方面有好處。也可以以鏡映排列方式將日誌檔放置在多個磁碟機上。這可防止包含日誌的磁碟機故障。在沒有鏡映的情況下，可能會強制您回到 WebSphere MQ 系統的前次備份。

使用日誌進行回復

使用日誌從失敗中回復。

有數種方式可讓您的資料受損。WebSphere MQ 可協助您從下列回復：

- 損壞的資料物件
- 系統中的電源流失
- 通訊失敗

本節查看如何使用日誌從這些問題回復。

從斷電或通訊故障中回復

WebSphere MQ 可以從通訊失敗和斷電兩者中回復。此外，有時也可以從其他類型的問題中回復，例如無意刪除檔案。

在通訊失敗的情況下，訊息會保留在佇列上，直到接收端應用程式移除它們為止。如果正在傳輸訊息，則訊息會保留在傳輸佇列中，直到可以順利傳輸為止。若要從通訊失敗中回復，您通常可以使用失敗的鏈結來重新啟動通道。

如果您失去電源，當佇列管理程式重新啟動時 WebSphere MQ 會將佇列還原至失敗時的已確定狀態。這可確保不會遺失持續訊息。非持續訊息會被捨棄；當 WebSphere MQ 突然停止時，它們無法存活。

回復損壞物件

IBM WebSphere MQ 物件有一些可能變成無法使用的方式，例如，因為無意造成損壞。然後，您必須回復完整系統或部分系統。所需的動作取決於偵測到損壞的時間、選取的日誌方法是否支援媒體回復，以及哪些物件損壞。

媒體回復

媒體回復會從線性日誌中記錄的資訊重建物件。例如，如果不小心刪除物件檔，或因為其他原因而無法使用，則媒體回復可以重建它。物件的媒體回復所需的日誌資訊稱為媒體映像檔。

媒體映像檔是一系列日誌記錄，其中包含可從中重建物件本身的物件映像檔。

重建物件所需的第一個日誌記錄稱為其媒體回復記錄；它是物件最新媒體映像檔的開頭。每一個物件的媒體回復記錄是檢查點期間所記錄的其中一個資訊片段。

從物件的媒體映像檔重建物件時，也需要重播任何日誌記錄，說明自前次取得映像檔以來對物件執行的更新。

例如，假設本端佇列具有在將持續訊息放入佇列之前取得的佇列物件影像。為了重建物件的最新影像，除了重播影像本身之外，還必須重播記錄將訊息放入佇列的日誌項目。

建立物件時，寫入的日誌記錄包含足夠資訊，可完全重建物件。這些記錄構成物件的第一個媒體影像。然後，在每次關閉時，佇列管理程式會自動記錄媒體影像，如下所示：

- 非本端處理程序物件及佇列的影像
- 空本端佇列的映像檔

也可以使用 `rcdmqing` 指令手動記錄媒體映像檔，如 `rcdmqing` 中所述。此指令會寫入 IBM WebSphere MQ 物件的媒體影像。當已寫入媒體映像檔時，只有保留媒體映像檔的日誌，以及在此時間之後建立的所有

日誌，才需要重建損壞的物件。建立媒體映像檔的好處取決於下列因素：可用的儲存體數量，以及建立日誌檔的速度。

從媒體映像檔回復

在佇列管理程式啟動期間，佇列管理程式會自動從其媒體映像檔回復部分物件。當佇列管理程式前次關閉時，如果它涉及任何不完整的交易，且在重新啟動處理程序期間發現毀損或損壞，則它會自動回復佇列。

您必須使用 `rcrmqobj` 指令手動回復其他物件，該指令會重播日誌中的記錄，以重建 IBM WebSphere MQ 物件。物件會從其在日誌中找到的最新映像檔重建，以及在儲存映像檔的時間與發出重建指令的時間之間所有適用的日誌事件。如果 IBM WebSphere MQ 物件變成已損壞，則唯一可以執行的有效動作是刪除它或透過此方法重建它。無法以這種方式回復非持續訊息。

如需 `rcrmqobj` 指令的進一步詳細資料，請參閱 `rcrmqobj`。

嘗試物件的媒體回復時，包含媒體回復記錄及所有後續日誌檔的日誌檔必須在日誌檔目錄中可用。如果找不到所需的檔案，則會發出操作員訊息 AMQ6767，且媒體回復作業失敗。如果您未取得要重建之物件的一般媒體映像檔，則可能沒有足夠的磁碟空間來保留重建物件所需的所有日誌檔。

在啟動期間回復損壞物件

如果佇列管理程式在啟動期間探索到損壞的物件，則它所採取的動作取決於物件類型，以及佇列管理程式是否配置為支援媒體回復。

如果佇列管理程式物件已損壞，除非佇列管理程式可以回復物件，否則無法啟動。如果佇列管理程式已配置線性日誌，因此支援媒體回復，則 IBM WebSphere MQ 會自動嘗試從其媒體映像檔重建佇列管理程式物件。如果選取的日誌方法不支援媒體回復，您可以還原佇列管理程式的備份或刪除佇列管理程式。

當佇列管理程式停止時，如果有任何交易在作用中，則也需要包含持續、未確定訊息的本端佇列，才能順利啟動佇列管理程式。如果發現其中任何本端佇列已損壞，且佇列管理程式支援媒體回復，則會自動嘗試從其媒體映像檔重建它們。如果有任何佇列無法回復，則 IBM WebSphere MQ 無法啟動。

在不支援媒體回復的佇列管理程式上，如果在啟動處理期間探索到任何包含未確定的訊息的已損壞本端佇列，則會將佇列標示為已損壞物件，並忽略其中未確定的訊息。此狀況是因為無法在這類佇列管理程式上執行損壞物件的媒體回復，而剩下的唯一動作是刪除它們。發出 AMQ7472 訊息以報告任何損壞。

在其他時間回復損壞物件

物件的媒體回復僅在啟動期間自動執行。在其他時間，當偵測到物件損壞時，會發出操作員訊息 AMQ7472，且大部分使用該物件的作業都會失敗。在佇列管理程式啟動之後，如果佇列管理程式物件隨時損壞，佇列管理程式會執行強制關機。當物件已損壞時，您可以刪除它，或者如果佇列管理程式正在使用線性日誌，請嘗試使用 `rcrmqobj` 指令從其媒體映像檔回復它 (如需進一步詳細資料，請參閱 `rcrmqobj`)。

保護 IBM WebSphere MQ 日誌檔

當佇列管理程式正在執行時，請勿碰觸日誌檔，可能無法進行回復。使用 `super` 使用者或 `mqm` 權限來保護日誌檔免遭無意修改。

當 IBM WebSphere MQ 佇列管理程式在執行中，請勿手動移除現行日誌檔。如果使用者不小心刪除佇列管理程式需要重新啟動的日誌檔，IBM WebSphere MQ 不會發出任何錯誤並繼續處理資料包括持續訊息。佇列管理程式會正常關閉，但可能無法重新啟動。因此無法回復訊息。

有權移除作用中佇列管理程式正在使用之日誌的使用者，也有權刪除其他重要佇列管理程式資源 (例如佇列檔、物件型錄及 IBM WebSphere MQ 執行檔)。因此，它們可能因缺乏經驗而損壞執行中或休眠佇列管理程式，而 IBM WebSphere MQ 無法保護本身。

授與超級使用者或 `mqm` 權限時請小心。

使用 `dmpmqlog` 指令傾出日誌內容

如何使用 `dmpmqlog` 指令來傾出佇列管理程式日誌的內容。

使用 `dmpmqlog` 指令來傾出佇列管理程式日誌的內容。依預設會傾出所有作用中日誌記錄，亦即，指令會從日誌標頭開始傾出 (通常是前次完成檢查點的開始)。

通常只有在佇列管理程式不在執行中時，才能傾出日誌。因為佇列管理程式會在關閉期間取得檢查點，所以日誌的作用中部分通常會包含少量日誌記錄。不過，您可以使用 `dmpmqlog` 指令，利用下列其中一個選項來傾出更多日誌記錄，以變更傾出的開始位置：

- 從日誌的基本開始傾出。日誌的基礎是日誌檔中包含日誌標頭的第一個日誌記錄。在此情況下傾出的其他資料量取決於日誌標頭在日誌檔中的位置。如果它接近日誌檔的開頭，則只會傾出少量其他資料。如果標頭接近日誌檔結尾，則會大量傾出更多資料。
- 指定傾出的起始位置作為個別日誌記錄。每一個日誌記錄都由唯一的日誌序號 (*LSN*) 識別。在循環式記載的情況下，此開始日誌記錄不能早於日誌的基礎；此限制不適用於線性日誌。在執行指令之前，您可能需要恢復非作用中日誌檔。您必須指定從先前 `dmpmqlog` 輸出取得的有效 *LSN* 作為起始位置。

例如，使用線性記載，您可以指定最後一個 `dmpmqlog` 輸出中的 `nextlsn`。`nextlsn` 會出現在 Log File Header 中，並指出要寫入的下一筆日誌記錄的 *LSN*。使用此作為起始位置，以格式化自前次傾出日誌以來寫入的所有日誌記錄。

- 僅限線性日誌，您可以指示 `dmpmqlog` 從任何給定的日誌檔範圍開始格式化日誌記錄。在此情況下，`dmpmqlog` 預期在與作用中日誌檔相同的目錄中找到此日誌檔，以及每一個連續的日誌檔。此選項不適用於循環日誌，其中 `dmpmqlog` 無法存取日誌基礎之前的日誌記錄。

`dmpmqlog` 指令的輸出是 Log File Header 及一系列格式化日誌記錄。佇列管理程式會使用數個日誌記錄來記錄其資料的變更。

部分已格式化的資訊僅供內部使用。下列清單包括最有用的日誌記錄：

日誌檔標頭

每一個日誌都有單一日誌檔標頭，一律是 `dmpmqlog` 指令所格式化的第一件事。它包含下列欄位：

<i>logactive</i>	主要日誌範圍的數目。
登入非作用中	次要日誌範圍數。
日誌大小	每個延伸範圍的 4 KB 頁面數。
<i>baselsn</i>	日誌範圍中包含日誌標頭的第一個 <i>LSN</i> 。
<i>nextlsn</i>	要寫入的下一個日誌記錄的 <i>LSN</i> 。
<i>headlsn</i>	日誌標頭上日誌記錄的 <i>LSN</i> 。
尾端	識別日誌尾端位置的 <i>LSN</i> 。
<i>hflag1</i>	日誌是 CIRCULAR 還是 LOG RETAIN (線性)。
<i>HeadExtentID</i>	包含日誌標頭の日誌範圍。

日誌記錄標頭

日誌內的每一個日誌記錄都有一個包含下列資訊的固定標頭：

<i>LSN</i>	日誌序號。
<i>LogRecd</i> 類型	日誌記錄的類型。
<i>XTranid</i>	與此日誌記錄相關聯的交易 ID (如果有的話)。 MQI 的 <i>TranType</i> 表示 WebSphere MQ 交易。XA 的 <i>TranType</i> 涉及其他資源管理程式。相同工作單元內涉及的更新項目具有相同的 <i>XTranid</i> 。
<i>QueueName</i>	與此日誌記錄相關聯的佇列 (如果有的話)。
<i>Qid</i>	佇列的唯一內部 ID。
<i>PrevLSN</i>	相同交易內前一個日誌記錄的 <i>LSN</i> (如果有的話)。

啟動佇列管理程式

這會記載佇列管理程式已啟動。

<i>StartDate</i>	佇列管理程式啟動的日期。
<i>StartTime</i>	佇列管理程式啟動的時間。

停止佇列管理程式

這會記載佇列管理程式已停止。

<i>StopDate</i>	佇列管理程式停止的日期。
<i>StopTime</i>	佇列管理程式停止的時間。
<i>ForceFlag</i>	使用的關機類型。

啟動檢查點

這表示佇列管理程式檢查點的開始。

結束檢查點

這表示佇列管理程式檢查點的結束。

<i>ChkPtLSN</i>	啟動此檢查點之日誌記錄的 LSN。
-----------------	-------------------

放置訊息

這會記載放置到佇列的持續訊息。如果訊息置於同步點之下，日誌記錄標頭會包含非空值 *XTranid*。記錄的其餘部分包含：

<i>MapIndex</i>	佇列上訊息的 ID。它可以用來比對用來從佇列取得此訊息的對應 MQGET。在此情況下，可以找到包含相同 <i>QueueName</i> 和 <i>MapIndex</i> 的後續 <i>Get Message</i> 日誌記錄。此時，可以針對該佇列的後續放置訊息重複使用 <i>MapIndex</i> ID。
資料	此日誌記錄的十六進位傾出中包含各種內部資料，後面接著「訊息描述子 (eyecatcher MD)」及訊息資料本身。

放置組件

對於單一日誌記錄而言太大的持續訊息會記載為多個 *Put Part* 日誌記錄，後面接著單一 *Put Message* 記錄。如果有 *Put Part* 記錄，則 *PrevLSN* 欄位會將 *Put Part* 記錄與最終 *Put Message* 記錄鏈結在一起。

資料	繼續前一個日誌記錄停止的訊息資料。
----	-------------------

取得訊息

只會記載持續訊息的取得。如果在同步點下取得訊息，則日誌記錄標頭包含非空值 *XTranid*。記錄的其餘部分包含：

<i>MapIndex</i>	識別從佇列擷取的訊息。包含相同 <i>QueueName</i> 及 <i>MapIndex</i> 的最新 <i>Put Message</i> 日誌記錄會識別所擷取的訊息。
<i>QPriority</i>	從佇列擷取的訊息優先順序。

啟動交易

指出新交易的開始。MQI 的 *TranType* 指出 WebSphere MQ 交易。XA 的 *TranType* 指出涉及其他資源管理程式。此交易所做的所有更新都將具有相同的 *XTranid*。

準備交易

指出佇列管理程式已準備好確定與指定 *XTranid* 相關聯的更新項目。在涉及其他資源管理程式的兩段式確定中，會寫入此日誌記錄。

確定交易

指出佇列管理程式已確定交易所做的所有更新。

回復交易

這表示佇列管理程式打算回復交易。

結束交易

這表示回復交易結束。

交易表格

此記錄在同步點期間寫入。它會記錄已進行持續更新之每一個交易的狀態。對於每一個交易，會記錄下列資訊：

<i>XTranid</i>	交易 ID。
<i>FirstLSN</i>	與交易相關聯的第一個日誌記錄的 LSN。
<i>LastLSN</i>	與交易相關聯的最後一個日誌記錄的 LSN。

交易參與者

此日誌記錄由佇列管理程式的 XA 交易管理程式元件寫入。它會記錄參與交易的外部資源管理程式。對於每一個參與者，會記錄下列各項：

<i>RMName</i>	資源管理程式的名稱。
<i>RMID</i>	資源管理程式 ID。這也會記載在後續的 <i>Transaction Prepared</i> 日誌記錄中，這些記錄會記錄資源管理程式所參與的廣域交易。
<i>SwitchFile</i>	這個資源管理程式的交換器載入檔。
<i>XAOpenString</i>	這個資源管理程式的 XA 開放式字串。
<i>XACloseString</i>	這個資源管理程式的 XA 關閉字串。

交易已備妥

此日誌記錄由佇列管理程式的 XA 交易管理程式元件寫入。它指出已順利準備指定的廣域交易。將指示每一個參與的資源管理程式確定。每一個備妥資源管理程式的 *RMID* 會記錄在日誌記錄中。如果佇列管理程式本身正在參與交易，則會呈現 *RMID* 為零的 *Participant Entry*。

交易忘記

此日誌記錄由佇列管理程式的 XA 交易管理程式元件寫入。當確定決策已遞送至每一個參與者時，它會遵循 *Transaction Prepared* 日誌記錄。

清除佇列

這會記載已清除佇列上所有訊息的事實，例如，使用 MQSC 指令 *CLEAR QUEUE*。

佇列屬性

這會記載佇列屬性的起始設定或變更。

建立物件

這會記載 WebSphere MQ 物件的建立。

<i>ObjName</i>	所建立物件的名稱。
<i>UserId</i>	執行建立的使用者 ID。

刪除物件

這會記載 WebSphere MQ 物件的刪除。

<i>ObjName</i>	已刪除的物件名稱。
----------------	-----------

備份及還原 IBM WebSphere MQ 佇列管理程式資料

備份佇列管理程式及佇列管理程式資料。

您可以定期採取措施，保護佇列管理程式免於因硬體故障而可能造成的毀損。有三種方法可保護佇列管理程式：

備份佇列管理程式資料

如果硬體失敗，可能會強制停止佇列管理程式。如果有任何佇列管理程式日誌資料因硬體故障而遺失，則佇列管理程式可能無法重新啟動。如果您備份佇列管理程式資料，則可能可以回復部分或全部遺失的佇列管理程式資料。

一般而言，您備份佇列管理程式資料的頻率越高，在硬體故障導致回復日誌失去完整性的情況下遺失的資料越少。

若要備份佇列管理程式資料，佇列管理程式不得在執行中。

若要備份及還原佇列管理程式資料，請參閱：

- [第 346 頁的『備份佇列管理程式資料』](#)。
- [第 347 頁的『還原佇列管理程式資料』](#)。

使用備份佇列管理程式

如果硬體故障嚴重，佇列管理程式可能無法復原。在此狀況下，如果無法復原的佇列管理程式具有專用的備份佇列管理程式，則可以啟動備份佇列管理程式來取代無法復原的佇列管理程式。如果定期更新，備份佇列管理程式日誌可以包含日誌資料，其中包括無法復原的佇列管理程式中最後一個完整日誌。

當現有佇列管理程式仍在執行中時，可以更新備份佇列管理程式。

若要建立及啟動備份佇列管理程式，請參閱：

- [第 348 頁的『建立備份佇列管理程式』](#)。
- [第 348 頁的『啟動備份佇列管理程式』](#)。

僅備份佇列管理程式配置

如果硬體失敗，可能會強制停止佇列管理程式。如果由於硬體故障而遺失佇列管理程式配置及日誌資料，則佇列管理程式將無法重新啟動或從日誌回復。如果您備份佇列管理程式配置，則可以從已儲存的定義重建佇列管理程式及其所有物件。

若要備份佇列管理程式配置，佇列管理程式必須在執行中。

若要備份及還原佇列管理程式配置，請參閱：

- [第 349 頁的『備份佇列管理程式配置』](#)。
- [第 349 頁的『還原佇列管理程式配置』](#)。

備份佇列管理程式資料

備份佇列管理程式資料可協助您防止因硬體錯誤而可能遺失資料。

開始之前

請確定佇列管理程式不在執行中。如果您嘗試備份執行中的佇列管理程式，則備份可能不一致，因為在複製檔案時正在進行更新。可能的話，請執行 `endmqm -w` 指令 (等待關閉) 來停止佇列管理程式，只有在失敗時，才使用 `endmqm -i` 指令 (立即關閉)。

關於這項作業

若要取得佇列管理程式資料的備份副本，請完成下列作業：

1. 使用配置檔中的資訊，搜尋佇列管理程式放置其資料及其日誌檔的目錄。如需相關資訊，請參閱 [第 350 頁的『變更 IBM WebSphere MQ 及佇列管理程式配置資訊』](#)。

註：您可能難以瞭解目錄中出現的名稱。會轉換名稱，以確保它們與您使用 WebSphere MQ 的平台相容。如需名稱轉換的相關資訊，請參閱 [瞭解 WebSphere MQ 檔名](#)。

2. 取得所有佇列管理程式的資料及日誌檔目錄 (包括所有子目錄) 的副本。

請確定您未遺失任何檔案，特別是 [第 334 頁的『日誌的外觀』](#) 中說明的日誌控制檔，以及 [第 60 頁的『起始設定和配置檔』](#) 中說明的配置檔。部分目錄可能是空的，但您需要它們全部，以便日後還原備份。

3. 保留檔案的所有權。若為 WebSphere MQ for UNIX and Linux 系統，您可以使用 `tar` 指令來執行此動作。(如果佇列大於 2 GB，則無法使用 `tar` 指令。如需相關資訊，請參閱 [啟用大型佇列](#)。

註：當您升級至 WebSphere MQ 7.5 版以及更新版本時，請確保備份 `.ini` 檔案及登錄項目。佇列管理程式資訊儲存在 `.ini` 檔案中，可用來回復至舊版 WebSphere MQ。

還原佇列管理程式資料

請遵循下列步驟來還原佇列管理程式資料的備份。

開始之前

請確定佇列管理程式不在執行中。

關於這項作業

如果要還原佇列管理程式資料的備份，請執行下列動作：

1. 使用配置檔中的資訊，尋找佇列管理程式放置其資料及其日誌檔的目錄。
2. 清空您要在其中放置備份資料的目錄。
3. 將已備份的佇列管理程式資料及日誌檔複製到正確的位置。
4. 更新配置資訊檔案。

請檢查產生的目錄結構，以確定您具有所有必要的目錄。

如需 IBM WebSphere MQ 目錄及子目錄的相關資訊，請參閱 [Windows 系統上的目錄結構](#) 及 [UNIX and Linux 系統上的目錄內容](#)。

請確定您有日誌控制檔以及日誌檔。另請檢查 IBM WebSphere MQ 與佇列管理程式配置檔是否一致，以便 WebSphere MQ 可以在正確的位置尋找還原的資料。

對於循環式記載，請同時備份佇列管理程式資料及日誌檔目錄，以便您可以還原一組一致的佇列管理程式資料及日誌。

對於線性記載，請同時備份佇列管理程式資料及日誌檔目錄。如果有對應的完整日誌檔序列可用，則只能還原佇列管理程式資料檔案。

註：當您升級至 WebSphere MQ 7.5 版以及更新版本時，請確保備份 `.ini` 檔案及登錄項目。佇列管理程式資訊儲存在 `.ini` 檔案中，可用來回復至舊版 WebSphere MQ。

結果

如果已正確備份及還原資料，則現在會啟動佇列管理程式。

使用備份佇列管理程式

現有的佇列管理程式可以有專用的備份佇列管理程式。

備份佇列管理程式是現有佇列管理程式的非作用中副本。如果現有佇列管理程式由於嚴重硬體故障而變成無法復原，則備份佇列管理程式可以連線，以取代無法復原的佇列管理程式。

必須定期將現有的佇列管理程式日誌檔複製到備份佇列管理程式，以確保備份佇列管理程式仍是災難回復的有效方法。對於要複製的日誌檔，不需要停止現有的佇列管理程式，不過，只有在佇列管理程式已完成寫入時，您才應該複製日誌檔。因為現有佇列管理程式日誌會持續更新，所以現有佇列管理程式日誌與複製到備份佇列管理程式日誌的日誌資料之間一律會有些微差異。備份佇列管理程式的定期更新會將兩個日誌之間的不相符程度降至最低。

如果需要使備份佇列管理程式連線，則必須先啟動，然後再啟動。在啟動備份佇列管理程式之前啟動備份佇列管理程式的需求，是防範意外啟動備份佇列管理程式的預防措施。一旦啟動備份佇列管理程式，就無法再更新它。

如需如何建立、更新及啟動備份佇列管理程式的相關資訊，請參閱下列主題：

- [第 348 頁的『建立備份佇列管理程式』](#)

- [第 348 頁的『更新備份佇列管理程式』](#)
- [第 348 頁的『啟動備份佇列管理程式』](#)

建立備份佇列管理程式

使用線性記載時，您只能使用備份佇列管理程式。

若要為現有佇列管理程式建立備份佇列管理程式，請執行下列動作：

1. 使用控制指令 `crtmqm` 為現有佇列管理程式建立備份佇列管理程式。備份佇列管理程式需要下列項目：
 - 具有與現有佇列管理程式相同的屬性，例如佇列管理程式名稱、記載類型及日誌檔大小。
 - 與現有佇列管理程式位於相同平台上。
 - 達到等於或高於現有佇列管理程式的程式碼層次。
2. 取得所有現有佇列管理程式的資料及日誌檔目錄 (包括所有子目錄) 的副本，如 [第 346 頁的『備份佇列管理程式資料』](#) 中所述。
3. 將備份佇列管理程式的資料和日誌檔目錄 (包括所有子目錄) 改寫為從現有佇列管理程式取得的副本。
4. 在備份佇列管理程式上執行下列控制指令：

```
stmqm -r BackupQMName
```

這會將佇列管理程式標示為 WebSphere MQ 內的備份佇列管理程式，並重播所有複製的日誌範圍，使備份佇列管理程式與現有的佇列管理程式一致。

更新備份佇列管理程式

為了確保備份佇列管理程式仍是災難回復的有效方法，必須定期更新。

定期更新可減少備份佇列管理程式日誌與現行佇列管理程式日誌之間的不相符。不需要停止要備份的佇列管理程式。

若要更新備份佇列管理程式，請執行下列動作：

1. 在要備份的佇列管理程式上發出下列 Script (MQSC) 指令：

```
RESET QMGR TYPE(ADVANCELOG)
```

這會停止對現行日誌的任何寫入，然後將佇列管理程式記載前進到下一個日誌範圍。這可確保您將所有記載的資訊備份至現行時間。

2. 在要備份的佇列管理程式上發出下列 Script (MQSC) 指令，以取得 (新的) 現行作用中日誌延伸範圍號碼：

```
DIS QMSTATUS CURRLOG
```

3. 將已更新的日誌範圍檔案從現行佇列管理程式日誌目錄複製到備份佇列管理程式日誌目錄-複製自前次更新以來的所有日誌範圍，並直到 (但不包括) 步驟 2 中指出的現行範圍。只複製以 "S.." 開頭的日誌延伸範圍檔案。
4. 在備份佇列管理程式上發出下列控制指令：

```
stmqm -r BackupQMName
```

這會重播所有複製的日誌範圍，並使備份佇列管理程式與佇列管理程式同步。當重播完成時，您會收到一則訊息，指出重新啟動回復所需的所有日誌範圍，以及媒體回復所需的所有日誌範圍。

警告：如果您將 non-contiguous 日誌集複製到備份佇列管理程式日誌目錄，則只會重播直到找到第一個遺漏日誌的日誌。

啟動備份佇列管理程式

您可以用備份佇列管理程式替代無法復原的佇列管理程式。

若要這麼做，請執行下列步驟：

1. 執行下列控制指令，以啟動備份佇列管理程式：

```
strmqm -a BackupQMName
```

已啟動備份佇列管理程式。現在作用中，無法再更新備份佇列管理程式。

2. 執行下列控制指令，以啟動備份佇列管理程式：

```
strmqm BackupQMName
```

WebSphere MQ 會將此視為重新啟動回復，並利用備份佇列管理程式中的日誌。在前次更新備份佇列管理程式重播期間，將會發生，因此只會回復前次記錄檢查點的作用中交易。

當替代無法復原的佇列管理程式來替代備份佇列管理程式時，可能會遺失無法復原的佇列管理程式中的部分佇列管理程式資料。遺失的資料量取決於前次更新備份佇列管理程式的時間。最近一次更新時，佇列管理程式資料流失越少。

3. 重新啟動所有通道。

請檢查產生的目錄結構，以確定您具有所有必要的目錄。

如需 WebSphere MQ 目錄和子目錄的相關資訊，請參閱 [規劃檔案系統支援](#)。

請確定您有日誌控制檔以及日誌檔。另請檢查 WebSphere MQ 與佇列管理程式配置檔是否一致，以便 WebSphere MQ 可以在正確的位置尋找還原的資料。

如果已正確備份及還原資料，則現在會啟動佇列管理程式。

註：即使佇列管理程式資料及日誌檔保留在不同的目錄中，也請同時備份及還原目錄。如果佇列管理程式資料及日誌檔具有不同的經歷時間，則佇列管理程式不是處於有效狀態，且可能不會啟動。如果它確實啟動，則您的資料可能會毀損。

備份佇列管理程式配置

備份佇列管理程式配置可協助您從其定義重建佇列管理程式。

如果要取得佇列管理程式配置的備份副本，請執行下列動作：

1. 請確定佇列管理程式正在執行。
2. a. 在 AIX、HP-UX、Linux、Solaris 或 Windows: 使用預設格式化選項 (-f mqsc) MQSC 及所有屬性 (-a) 來執行「傾出 MQ 配置」指令 (dmpmqcfg)，例如：

```
dmpmqcfg -m MYQMGR -a > /mq/backups/MYQMGR.mqsc
```

還原佇列管理程式配置

請遵循下列步驟來還原佇列管理程式配置的備份。

如果要還原佇列管理程式配置的備份，請執行下列動作：

1. 請確定佇列管理程式正在執行。請注意，如果無法使用其他方法來復原資料和日誌的損壞，則可能已重建佇列管理程式。
2. 視您的平台而定，執行下列其中一個指令：
 - a. 在 AIX、HP-UX、Linux、Solaris 或 Windows: 針對佇列管理程式執行 runmqsc，使用標準輸入重新導向，從 Dump MQ Configuration (dmpmqcfg) 指令所產生的 Script 檔中還原定義，例如：

```
runmqsc MYQMGR < /mq/backups/MYQMGR.mqsc
```

相關參考

[dmpmqcfg](#)

變更 IBM WebSphere MQ 及佇列管理程式配置資訊

變更 IBM WebSphere MQ 或個別佇列管理程式的行為，以符合您安裝的需求。

您可以透過變更在控管 IBM WebSphere MQ 的一組配置屬性 (或參數) 上指定的值，來變更 IBM WebSphere MQ 配置資訊。

透過編輯 IBM WebSphere MQ 配置檔來變更屬性資訊。在 IBM WebSphere MQ for Windows 和 Linux (x86 和 x86-64 平台) 上，可以使用 IBM WebSphere MQ Explorer 來編輯 IBM WebSphere MQ 配置檔。

在 Windows 系統上，您也可以使用 amqmdain 來變更配置資訊，如 [amqmdain](#) 中所述

若要進一步瞭解如何為您的平台配置 IBM WebSphere MQ 及佇列管理程式，請參閱下列子主題：

相關概念

[第 5 頁的『配置』](#)

在一或多部電腦上建立一或多個佇列管理程式，並在開發、測試及正式作業系統上配置它們，以處理包含商業資料的訊息。

相關工作

[規劃](#)

[管理 WebSphere MQ](#)

變更 UNIX, Linux, and Windows 系統上的配置資訊

配置屬性保留在配置檔中，位於節點及佇列管理程式的層次。

在 Windows、UNIX and Linux 平台上，您可以在下列範圍內變更 IBM WebSphere MQ 配置屬性：

- IBM WebSphere MQ 配置檔 (**mqs.ini**)，用於影響整個節點上 IBM WebSphere MQ 的變更。每一個節點都有一個 mqs.ini 檔案。
- 佇列管理程式配置檔 (**qm.ini**)，以影響特定佇列管理程式的變更。節點上的每一個佇列管理程式都有一個 qm.ini 檔案。

用戶端配置選項分別保留在用戶端配置檔中。

配置檔 (或 **段落** 檔案) 包含一或多個段落，這些段落是 .ini 檔案中的字行群組，它們一起具有共同功能或定義部分系統 (例如日誌功能、通道功能及可安裝的服務)。

因為使用 IBM WebSphere MQ 配置檔來尋找與佇列管理程式相關聯的資料，所以不存在或不正確的配置檔可能會導致部分或所有 MQSC 指令失敗。此外，應用程式也無法連接至 IBM WebSphere MQ 配置檔中未定義的佇列管理程式。

您對配置檔所做的任何變更通常要等到下次啟動佇列管理程式時才會生效。

在 Windows 和 Linux (x86 和 x86-64 平台) 系統上，您可以從 IBM WebSphere MQ Explorer 編輯配置資訊。

在 Windows 系統上，您也可以使用 amqmdain 指令來編輯配置檔。

如需 Windows、UNIX and Linux 系統上配置選項的相關資訊，請參閱下列子主題：

相關概念

[第 5 頁的『配置』](#)

在一或多部電腦上建立一或多個佇列管理程式，並在開發、測試及正式作業系統上配置它們，以處理包含商業資料的訊息。

[第 350 頁的『變更 IBM WebSphere MQ 及佇列管理程式配置資訊』](#)

變更 IBM WebSphere MQ 或個別佇列管理程式的行為，以符合您安裝的需求。

相關工作

[規劃](#)

[管理 WebSphere MQ](#)

相關參考

[第 355 頁的『用於變更 IBM WebSphere MQ 配置資訊的屬性』](#)

在 IBM WebSphere MQ for Windows 系統及 IBM WebSphere MQ for Linux (x86 及 x86-64 平台) 系統上, 使用 IBM WebSphere MQ Explorer 來修改配置資訊。在其他系統上, 透過編輯 mqs.ini 配置檔來修改資訊。

第 360 頁的『變更佇列管理程式配置資訊』

這裡說明的屬性會修改個別佇列管理程式的配置。它們會置換 WebSphere MQ 的任何設定。

編輯配置檔

使用指令或標準文字編輯器來編輯配置檔。

在編輯配置檔之前, 請先備份它, 以便在有需要時可以回復成您的副本。

您可以編輯配置檔:

- 自動, 使用變更節點上佇列管理程式配置的指令
- 手動, 使用標準文字編輯器

安裝之後, 您可以在 WebSphere MQ 配置檔中編輯預設值。

如果您在配置檔屬性上設定不正確的值, 則會忽略該值, 並發出操作員訊息以指出問題。(效果與完全遺漏屬性相同。)

當您建立新的佇列管理程式時:

- 備份 WebSphere MQ 配置檔
- 備份新的佇列管理程式配置檔

註解可以包含在配置檔中, 方法是在註解文字之前新增 ";" 或 "#" 字元。如果您要使用 ";" 或 "#" 字元, 但不包含代表註解的字元, 則可以在該字元前面加上 "\" 字元, 它將用作配置資料的一部分。

您何時需要編輯配置檔?

編輯配置檔以從備份回復、移動佇列管理程式、變更預設佇列管理程式或協助 IBM 支援。

例如, 在下列情況下, 您可能需要編輯配置檔:

- 您會遺失配置檔。(如果可以的話, 請從備份回復。)
- 您需要將一或多個佇列管理程式移至新目錄。
- 您需要變更預設佇列管理程式; 如果您不小心刪除現有的佇列管理程式, 可能會發生這種情況。
- IBM 支援中心建議您這麼做。

配置檔優先順序

屬性的值定義在多個位置。指令中設定的屬性優先於配置檔中的屬性。

配置檔的屬性值是根據下列優先順序來設定:

- 在指令行上輸入的參數優先於配置檔中定義的值
- qm.ini 檔案中定義的值優先於 mqs.ini 檔案中定義的值。

IBM WebSphere MQ 配置檔 mqs.ini

IBM WebSphere MQ 配置檔 mqs.ini 包含與節點上所有佇列管理程式相關的資訊。在安裝期間會自動建立它。

在 IBM WebSphere MQ for UNIX and Linux 產品上, 資料目錄和日誌目錄一律分別為 /var/mqm 和 /var/mqm/log。

在 Windows 系統上, 資料目錄 mqs.ini 的位置及日誌目錄的位置會儲存在登錄中, 因為它們的位置可能有所不同。

此外, 在 Windows 系統上, 安裝配置資訊 (包含在 IBM WebSphere MQ for UNIX and Linux 系統上的 mqinst.ini 中) 位於登錄中, 因為 Windows 上沒有 mqinst.ini 檔案。

Windows 系統的 mqs.ini 檔案由 HKLM\SOFTWARE\IBM\WebSphere MQ 金鑰中指定的 WorkPath 提供。它包含：

- 佇列管理程式的名稱
- 預設佇列管理程式的名稱
- 與每一個檔案相關聯的檔案位置

為新的 IBM WebSphere MQ 安裝提供的 LogDefaults 段落未包含屬性的任何明確值。缺少屬性表示在建立新的佇列管理程式時使用此值的預設值。第 352 頁的圖 71 中的 LogDefaults 段落會顯示預設值。LogBufferPages 屬性的零值表示 512。

如果您需要非預設值，則必須在 LogDefaults 段落中明確指定該值。

```
#*****#
#* Module Name: mqs.ini                               **#
#* Type       : WebSphere MQ Machine-wide Configuration File      **#
#* Function   : Define WebSphere MQ resources for an entire machine **#
#*****#
#* Notes     :                                               **#
#* 1) This is the installation time default configuration          **#
#*                                                  **#
#*****#
AllQueueManagers:
#*****#
#* The path to the qmgrs directory, below which queue manager data **#
#* is stored                                                       **#
#*****#
DefaultPrefix=/var/mqm

LogDefaults:
  LogPrimaryFiles=3
  LogSecondaryFiles=2
  LogFilePages=4096
  LogType=CIRCULAR
  LogBufferPages=0
  LogDefaultPath=/var/mqm/log

QueueManager:
  Name=saturn.queue.manager
  Prefix=/var/mqm
  Directory=saturn!queue!manager
  InstallationName=Installation1

QueueManager:
  Name=pluto.queue.manager
  Prefix=/var/mqm
  Directory=pluto!queue!manager
  InstallationName=Installation2

DefaultQueueManager:
  Name=saturn.queue.manager

ApiExitTemplate:
  Name=OurPayrollQueueAuditor
  Sequence=2
  Function=EntryPoint
  Module=/usr/ABC/auditor
  Data=123

ApiExitCommon:
  Name=MQPoliceman
  Sequence=1
  Function=EntryPoint
  Module=/usr/MQPolice/tmqp
  Data=CheckEverything
```

圖 71: UNIX 系統的 IBM WebSphere MQ 配置檔範例

佇列管理程式配置檔 qm.ini

佇列管理程式配置檔 qm.ini 包含與特定佇列管理程式相關的資訊。

每一個佇列管理程式都有一個佇列管理程式配置檔。建立與 qm.ini 檔案相關聯的佇列管理程式時，會自動建立該檔案。

V7.5.0.9 從 IBM WebSphere MQ Version 7.5.0Fix Pack 9 開始，在完全啟動佇列管理程式之前，**strmqm** 指令會檢查 qm.ini 檔案中 CHANNELS 及 SSL 段落的語法，這可讓您更容易查看錯誤的內容，並在 **strmqm** 發現 qm.ini 檔案包含任何錯誤時快速更正。如需相關資訊，請參閱 [strmqm](#)。

qm.ini 檔案的位置



在 UNIX and Linux 系統上，qm.ini 檔會保留在佇列管理程式所佔用的目錄樹狀結構根目錄中。例如，佇列管理程式 QMNAME 的配置檔路徑及名稱為：

```
/var/mqm/qmgrs/QMNAME/qm.ini
```

在 Windows 系統上，qm.ini 檔的位置由 HKLM\SOFTWARE\IBM\WebSphere MQ 金鑰中指定的 WorkPath 提供。例如，佇列管理程式 QMNAME 的配置檔路徑及名稱為：

```
C:\Program Files\IBM\WebSphere MQ\qmgrs\QMNAME\qm.ini
```

佇列管理程式名稱長度最多可為 48 個字元。不過，這並不保證名稱有效或唯一。因此，會根據佇列管理程式名稱來產生目錄名稱。此程序稱為名稱轉換。如需說明，請參閱 [瞭解 WebSphere MQ 檔名](#)。

範例 qm.ini 檔案



下列範例顯示如何在 IBM WebSphere MQ for UNIX and Linux 系統的佇列管理程式配置檔中排列屬性群組。

```
## Module Name: qm.ini                                ##
## Type       : WebSphere MQ queue manager configuration file ##
## Function   : Define the configuration of a single queue manager ##
##           ##                                         ##
##*****##
## Notes     :                                         ##
## 1) This file defines the configuration of the queue manager ##
##           ##                                         ##
##*****##

ExitPath:
  ExitsDefaultPath=/var/mqm/exits
  ExitsDefaultPath64=/var/mqm/exits64

Service:
  Name=AuthorizationService
  EntryPoints=13

ServiceComponent:
  Service=AuthorizationService
  Name=MQSeries.UNIX.auth.service
  Module=opt/mqm/bin/amqzfu
  ComponentDataSize=0

Log:
  LogPrimaryFiles=3
  LogSecondaryFiles=2
  LogFilePages=4096
  LogType=CIRCULAR
  LogBufferPages=01
  LogPath=/var/mqm/log/saturn!queue!manager/

AccessMode:
  SecurityGroup=wmq\wmq
```

```

XAResourceManager:
  Name=DB2 Resource Manager Bank
  SwitchFile=/usr/bin/db2swit
  XAOpenString=MQBankDB
  XACloseString=
  ThreadOfControl=THREAD

Channels: 2
  MaxChannels=200
  MaxActiveChannels=100
  MQIBindType=STANDARD

AccessMode:
  SecurityGroup=wmq\wmq
TCP:
  KeepAlive = Yes
  SvrSndBuffSize=32768
  SvrRcvBuffSize=32768
  Connect_Timeout=0

QMErrorLog:
  ErrorLogSize=262144
  ExcludeMessage=7234
  SuppressMessage=9001,9002,9202
  SuppressInterval=30

ApiExitLocal:
  Name=ClientApplicationAPIchecker
  Sequence=3
  Function=EntryPoint
  Module=/usr/Dev/ClientAppChecker
  Data=9.20.176.20

```

附註:

1. LogBufferPages 的零值會提供 512 值。
2. 如需通道段落的相關資訊，請參閱第 60 頁的『起始設定和配置檔』。
3. XAResourceManager 段落數上限為 255。不過，您應該只使用少量段落，以避免交易效能降低。

Unix 中 WebSphere MQ 使用副檔名為 .ini 的配置檔，例如 qm.ini。WebSphereMQ 中有一些公用程式 (例如 **setmqm**) 會建立檔案的暫時備份副本。例如，檔案 qm.ini 將建立名為 qm.ini.bak 的備份副本。公用程式會修改 qm.ini 檔，儲存已更新的檔案，然後刪除 qm.ini.bak 檔。如果公用程式無法儲存 qm.ini 檔，則會從備份檔 qm.ini.bak 還原 qm.ini 的內容，然後刪除 qm.ini.bak 檔。

如果有現有的 qm.ini.bak 檔，公用程式會以 qm.ini.bak 的內容來回復 qm.ini 檔，並刪除 qm.ini.bak 檔。因此，您不應使用副檔名 .bak 來建立 *.ini 檔的備份副本，因為 WebSphere MQ 公用程式可能會刪除這些備份檔。

如需變更生效時間的相關資訊，請參閱第 350 頁的『變更 UNIX, Linux, and Windows 系統上的配置資訊』。

安裝配置檔 mqinst.ini

UNIX and Linux 系統

安裝配置檔 mqinst.ini 包含 UNIX 或 Linux 系統上所有 IBM WebSphere MQ 安裝的相關資訊。

mqinst.ini 檔案位於 UNIX and Linux 系統上的 /etc/opt/mqm 目錄中。它包含哪個安裝 (如果有的話) 是主要安裝的相關資訊，以及每個安裝的下列資訊:

- 安裝名稱
- 安裝說明
- 安裝 ID
- 安裝路徑

此檔案不可直接編輯或參照，因為其格式未修正且可能變更。請改為使用下列指令來建立、刪除、查詢及修改 mqinst.ini 檔案中的值:

`crtmqinst` 以建立項目。

[dlmqinst](#) 以刪除項目。
[dspmqinst](#) 以顯示項目。
[setmqinst](#) 以設定項目。

安裝 ID (僅供內部使用) 會自動設定, 且不得變更。

Windows 系統

在 Windows 系統上, 安裝配置資訊保留在下列金鑰中:

```
HKLM\SOFTWARE\IBM\WebSphere MQ\Installation\
```

不得直接編輯或參照此索引鍵, 因為其格式未修正且可能變更。請改用下列指令來查詢及修改登錄中的值:

[dspmqinst](#) 以顯示項目。
[setmqinst](#) 以設定項目。

在 Windows 上, 無法使用 **crtmqinst** 和 **dlmqinst** 指令。安裝及解除安裝程序會處理必要登錄項目的建立及刪除。

用於變更 IBM WebSphere MQ 配置資訊的屬性

在 IBM WebSphere MQ for Windows 系統及 IBM WebSphere MQ for Linux (x86 及 x86-64 平台) 系統上, 使用 IBM WebSphere MQ Explorer 來修改配置資訊。在其他系統上, 透過編輯 mqs.ini 配置檔來修改資訊。

如需特定元件的屬性, 請參閱下列子主題:

相關概念

[第 5 頁的『配置』](#)

在一或多部電腦上建立一或多個佇列管理程式, 並在開發、測試及正式作業系統上配置它們, 以處理包含商業資料的訊息。

[第 350 頁的『變更 IBM WebSphere MQ 及佇列管理程式配置資訊』](#)

變更 IBM WebSphere MQ 或個別佇列管理程式的行為, 以符合您安裝的需求。

相關工作

[規劃](#)

[管理 WebSphere MQ](#)

相關參考

[第 360 頁的『變更佇列管理程式配置資訊』](#)

這裡說明的屬性會修改個別佇列管理程式的配置。它們會置換 WebSphere MQ 的任何設定。

所有佇列管理程式

使用「IBM WebSphere MQ 探險家」中的 General 及 Extended WebSphere MQ 內容頁面, 或 mqs.ini 檔案中的 AllQueueManagers 段落, 以指定所有佇列管理程式的下列相關資訊。

DefaultPrefix=directory_name

這個屬性指定佇列管理程式資料保留在其中的 qmgrs 目錄路徑。

如果您變更佇列管理程式的預設字首, 請抄寫安裝時所建立的目錄結構。

尤其是您必須建立 qmgrs 結構。在變更預設字首之前停止 WebSphere MQ, 並僅在您將結構移至新位置並變更預設字首之後重新啟動 WebSphere MQ。

註: 請勿刪除 UNIX and Linux 系統上的 /var/mqm/errors 目錄, 或 Windows 系統上的 \errors 目錄。

除了變更預設字首, 您也可以使用環境變數 MQSPREFIX 來置換 crtmqm 指令的 DefaultPrefix。

由於作業系統限制, 請將提供的路徑保持足夠短, 使路徑長度和任何佇列管理程式名稱的總和最長為 70 個字元。

ConvEBCDICNewline= NL_TO_LF | TABLE | ISO

EBCDIC 字碼頁包含 ASCII 字碼頁不支援的換行 (NL) 字元 (雖然部分 ASCII 的 ISO 變式包含對等項目)。使用 ConvEBCDICNewline 屬性來指定 WebSphere MQ 如何將 EBCDIC NL 字元轉換成 ASCII 格式。

NL_TO_LF

針對所有 EBCDIC 至 ASCII 轉換，將 EBCDIC NL 字元 (X'15 ') 轉換為 ASCII 換行字元 LF (X'0A')。

NL_TO_LF 是預設值。

表格

根據平台上用於所有 EBCDIC 至 ASCII 轉換的轉換表來轉換 EBCDIC NL 字元。

這種類型轉換的效果可能因平台而不同，也可能因語言而不同；即使在相同平台上，如果您使用不同的 CCSID，行為也可能不同。

ISO

轉換：

- 使用 TABLE 方法的 ISO CCSID
- 使用 NL_TO_CF 方法的所有其他 CCSID

可能的 ISO CCSID 顯示在 第 356 頁的表 32 中。

CCSID	代碼集
819	ISO8859-1
912	ISO8859-2
915	ISO8859-5
1089	ISO8859-6
813	ISO8859-7
916	ISO8859-8
920	ISO8859-9
1051	roman8

如果 ASCII CCSID 不是 ISO 子集，則 ConvEBCDICNewline 預設為 NL_TO_LF。

預設佇列管理程式

使用「IBM WebSphere MQ 探險家」中的 General WebSphere MQ 內容頁面，或 mqs.ini 檔案中的 DefaultQueueManager 段落來指定預設佇列管理程式。

名稱 =default_queue_manager

預設佇列管理程式會處理未明確指定佇列管理程式名稱的任何指令。如果您建立新的預設佇列管理程式，則會自動更新 DefaultQueueManager 屬性。如果您不小心建立新的預設佇列管理程式，然後想要回復原始佇列管理程式，請手動變更 DefaultQueueManager 屬性。

結束內容

使用「IBM WebSphere MQ 探險家」中的 Extended IBM WebSphere MQ 內容頁面或 mqs.ini 檔案中的 ExitProperties 段落，以指定佇列管理程式結束程式所使用的配置選項。

CLWLMode=SAFE|FAST

叢集工作量 (CLWL) 結束程式可讓您指定要在叢集中開啟哪個叢集佇列，以回應 MQI 呼叫 (例如，MQOPEN、MQPUT)。CLWL 結束程式會根據您在 CLWLMode 屬性上指定的值，以 FAST 模式或 SAFE 模式執行。如果您省略 CLWLMode 屬性，則叢集工作量結束程式會以 SAFE 模式執行。

SAFE

在與佇列管理程式不同的處理程序中執行 CLWL 結束程式。這是預設值。

以 SAFE 模式執行時，如果使用者撰寫的 CLWL 結束程式發生問題，則會發生下列情況：

- CLWL 伺服器處理程序 (amqzlw0) 失敗。
- 佇列管理程式會重新啟動 CLWL 伺服器處理程序。
- 錯誤日誌中會向您報告此錯誤。如果 MQI 呼叫正在進行中，您會收到回覆碼形式的通知。

佇列管理程式的完整性會保留下來。

註：在個別處理程序中執行 CLWL 結束程式可能會影響效能。

FAST

在佇列管理程式處理程序中行內執行叢集結束程式。

指定此選項可避免與在 SAFE 模式中執行相關聯的處理程序切換成本，從而增進效能，但會以犧牲佇列管理程式完整性為代價。只有在您確信 CLWL 結束程式 **沒有** 問題，且您特別關心效能時，才應該以 FAST 模式執行 CLWL 結束程式。

當 CLWL 結束程式以 FAST 模式執行時，如果發生問題，佇列管理程式會失敗，且您會面臨佇列管理程式完整性受損的風險。

IBM WebSphere MQ 的日誌預設值

使用 IBM WebSphere MQ Explorer 中的 Default log settings IBM WebSphere MQ 內容頁面，或 mqs.ini 檔案中的 LogDefaults 段落，來指定所有佇列管理程式的日誌預設值相關資訊。

如果段落不存在，則會使用 MQ 預設值。當您建立佇列管理程式時，會使用日誌屬性作為預設值，但如果您在 `crtmqm` 指令中指定日誌屬性，則可以置換日誌屬性。如需此指令的詳細資料，請參閱 [crtmqm](#)。

建立佇列管理程式之後，會從 [第 363 頁的『佇列管理程式日誌』](#) 中說明的設定取得該佇列管理程式的日誌屬性。

指定給特定佇列管理程式 (指定在 [第 363 頁的『佇列管理程式日誌』](#) 中) 的預設字首 (指定在 [第 355 頁的『所有佇列管理程式』](#) 中) 及日誌路徑，容許佇列管理程式及其日誌位於不同的實體磁碟機上。這是建議的方法，雖然依預設它們位於相同的磁碟機上。

如需計算日誌大小的相關資訊，請參閱 [第 337 頁的『計算日誌的大小』](#)。

註：下列參數清單中提供的限制是 WebSphere MQ 所設定的限制。作業系統限制可能會減少可能的日誌大小上限。

LogPrimaryFiles=3|2-254 (Windows) |2-510 (UNIX and Linux 系統)

建立佇列管理程式時所配置的日誌檔。

在 Windows 上，您可以擁有的主要日誌檔數目下限為 2，上限為 254，在 UNIX and Linux 系統上，則為 510。預設是 3。

在 Windows 上，主要及次要日誌檔的總數不得超過 255，在 UNIX and Linux 系統上，則不得超過 511，且不得小於 3。

建立或啟動佇列管理程式時會檢查這個值。您可以在建立佇列管理程式之後變更它。不過，除非重新啟動佇列管理程式，否則值中的變更不會生效，且效果可能不會立即生效。

LogSecondaryFiles=2|1-253 (Windows) |1-509 (UNIX and Linux 系統)

當主要檔耗盡時所配置的日誌檔。

在 Windows 上，次要日誌檔數目下限為 1，上限為 253，在 UNIX and Linux 系統上，則為 509。預設數字為 2。

在 Windows 上，主要及次要日誌檔的總數不得超過 255，在 UNIX and Linux 系統上，則不得超過 511，且不得小於 3。

當佇列管理程式啟動時，會檢查該值。您可以變更此值，但除非重新啟動佇列管理程式，否則變更不會生效，甚至可能不會立即生效。

LogFilePages =number

日誌資料保留在稱為日誌檔的一系列檔案中。日誌檔大小以 4 KB 頁面為單位指定。

預設日誌檔頁數是 4096，提供 16 MB 的日誌檔大小。

在 UNIX and Linux 系統上，日誌檔頁數下限為 64，在 Windows 上，日誌檔頁數下限為 32；在這兩種情況下，日誌檔頁數上限為 65 535。

註：無法針對佇列管理程式變更在佇列管理程式建立期間指定的日誌檔大小。

LogType=CIRCULAR| LINEAR

要使用的日誌類型。預設值為 CIRCULAR。

CIRCULAR

使用日誌來啟動重新啟動回復，以回復系統停止時進行中的交易。

如需循環式記載的完整說明，請參閱第 335 頁的『[記載類型](#)』。

線性

對於重新啟動回復及媒體或正向回復 (透過重播日誌內容來建立遺失或損壞的資料)。

如需線性記載的更完整說明，請參閱第 335 頁的『[記載類型](#)』。

如果您要變更預設值，可以編輯 LogType 屬性，或使用 `crtmqm` 指令指定線性記載。在建立佇列管理程式之後，您無法變更記載方法。

LogBufferPages=0|0-4096

配置給緩衝區記錄以進行寫入的記憶體數量，指定緩衝區大小 (以 4 KB 分頁為單位)。

緩衝區頁數下限為 18，上限為 4096。較大的緩衝區會導致更高的傳輸量，特別是對於較大的訊息。

如果您指定 0 (預設值)，佇列管理程式會選取大小。在 WebSphere MQ 7.0 版中，這是 512 (2048 KB)。

如果您指定 1 到 17 範圍內的數字，則佇列管理程式預設為 18 (72 KB)。如果您指定範圍 18 到 4096 之間的數字，佇列管理程式會使用指定的數字來設定配置的記憶體。

LogDefault 路徑 =目錄名稱

佇列管理程式的日誌檔所在的目錄。目錄位於佇列管理程式可以寫入的本端裝置上，且最好是位於與訊息佇列不同的磁碟機上。指定不同的磁碟機會在系統失效時提供額外的保護。

預設值為：

- `<DefaultPrefix>\log` 適用於 WebSphere MQ for Windows，其中 `<DefaultPrefix>` 是在 All Queue Managers WebSphere MQ 內容頁上的 `DefaultPrefix` 屬性上指定的值。此值是在安裝時設定。
- `/var/mqm/log` for WebSphere MQ for UNIX and Linux 系統

或者，您可以使用 `-ld` 旗標在 `crtmqm` 指令上指定目錄名稱。當建立佇列管理程式時，也會在佇列管理程式目錄下建立一個目錄，用來保留日誌檔。此目錄的名稱基於佇列管理程式名稱。這可確保日誌檔路徑是唯一的，而且符合目錄名稱長度的任何限制。

如果您未在 `crtmqm` 指令上指定 `-ld`，則會使用 `mqs.ini` 檔案中 `LogDefaultPath` 屬性的值。

佇列管理程式名稱會附加至目錄名稱，以確保多個佇列管理程式使用不同的日誌目錄。

建立佇列管理程式時，會在配置資訊的日誌屬性中建立 `LogPath` 值，並提供佇列管理程式日誌的完整目錄名稱。此值用於在啟動或刪除佇列管理程式時尋找日誌。

LogWriteIntegrity =SingleWrite|DoubleWrite| TripleWrite

日誌程式用來可靠地寫入日誌記錄的方法。

TripleWrite

這是預設方法。

請注意，可以選取 **DoubleWrite**，但如果您這樣做，系統會將此解譯為 **TripleWrite**。

SingleWrite

只有在託管 WebSphere MQ 回復日誌的檔案系統或裝置明確地保證 4KB 寫入這一最小單位時，才應該使用 **SingleWrite**。

也就是說，如果寫入 4KB 頁面失敗，不論原因為何，僅有的兩種可能狀態為前像或後像。不會有中間狀態。

進階配置與電源介面 (ACPI)

使用「IBM WebSphere MQ 探險家」中的 ACPI WebSphere MQ 內容頁面，來指定系統接收暫停要求時 WebSphere MQ 的行為方式。

Windows 支援「進階配置及電源介面 (ACPI)」標準。這可讓已啟用 ACPI 硬體的 Windows 使用者在系統進入並從暫停模式回復時停止並重新啟動通道。

請注意，只有在「警示監視器」執行時，才會套用 ACPI WebSphere MQ 內容頁中指定的設定。如果「警示監視器」正在執行中，則「警示監視器」圖示會出現在工作列上。

DoDialog=Y | N

顯示暫停要求時的對話框。

DenySuspend= Y | N

拒絕暫停要求。例如，如果 DoDialog= N，或如果 DoDialog= Y 且無法顯示對話框，則會使用此選項，因為您的 Notebook 蓋子已關閉。

CheckChannelsRunning=Y | N

檢查是否有任何通道在執行中。結果可以決定其他設定的結果。

下表概述這些參數的每一個組合的效果：

DoDialog	DenySuspend	CheckChannels 執行中	動作
N	N	N	接受暫停要求。
N	N	Y	接受暫停要求。
N	Y	N	拒絕暫停要求。
N	Y	Y	如果有任何通道正在執行中，則拒絕暫停要求；如果不接受要求，則拒絕要求。
Y	N	N	顯示對話框 (請參閱 附註 ；接受暫停要求)。這是預設值。
Y	N	Y	如果沒有通道在執行中，請接受暫停要求；如果它們顯示對話框 (請參閱 附註 ；接受要求)。
Y	Y	N	顯示對話框 (附註 ；拒絕暫停要求)。
Y	Y	Y	如果沒有通道在執行中，請接受暫停要求；如果它們顯示對話框 (附註 ；拒絕要求)。

註：在動作是要顯示對話的情況下，如果無法顯示對話 (例如，因為您的筆記本蓋子已關閉)，則會使用 DenySuspend 選項來判定是否接受或拒絕暫停要求。

API 結束程式

使用 IBM WebSphere MQ Explorer 或 amqmdain 指令來變更 API 結束程式的項目。

使用 IBM WebSphere MQ Explorer 中的 Exits IBM WebSphere MQ 內容頁面或 mqs.ini 檔案中的 ApiExitTemplate 及 ApiExitCommon 段落，以識別所有佇列管理程式的 API 結束常式。在 Windows 系統上，您也可以使用 amqmdain 指令來變更 API 結束程式的項目。(若要識別個別佇列管理程式的 API 結束常式，請使用 ApiExitLocal 段落，如 [第 371 頁的『API 結束程式』](#) 中所述。)

如需這些段落的屬性完整說明，請參閱 [配置 API 結束程式](#)。

佇列管理程式

每個佇列管理程式都有一個 QueueManager 段落。請使用段落來指定佇列管理程式目錄的位置。

在 Windows、UNIX and Linux 系統上，每個佇列管理程式都有一個 QueueManager 段落。這些屬性指定佇列管理程式名稱，以及包含與該佇列管理程式相關聯之檔案的目錄名稱。目錄名稱是以佇列管理程式名稱為基礎，但如果佇列管理程式名稱不是有效的檔名，則會轉換。如需名稱轉換的相關資訊，請參閱 [瞭解 WebSphere MQ 檔名](#)。

名稱 =queue_manager_name

佇列管理程式的名稱。

字首 =prefix

儲存佇列管理程式檔案的位置。依預設，此值與「所有佇列管理程式」資訊的 DefaultPrefix 屬性中指定的值相同。

目錄 =名稱

儲存佇列管理程式檔案之 <prefix>\QMGRS 目錄下的子目錄名稱。此名稱是以佇列管理程式名稱為基礎，但如果名稱重複，或佇列管理程式名稱不是有效的檔名，則可以轉換。

DataPath=path

建立佇列管理程式時所提供的明確資料路徑，這會置換字首和目錄作為佇列管理程式資料的路徑。

InstallationName=name

與此佇列管理程式相關聯的 WebSphere MQ 安裝名稱。與此佇列管理程式互動時，必須使用此安裝中的指令。如果沒有 InstallationName 值，則佇列管理程式會與 7.1 版之前的 WebSphere MQ 安裝相關聯。

相關概念

第 14 頁的『[將佇列管理程式與安裝相關聯](#)』

當您建立佇列管理程式時，它會自動與發出 **crtmqm** 指令的安裝相關聯。在 UNIX, Linux, and Windows 上，您可以使用 **setmqm** 指令來變更與佇列管理程式相關聯的安裝。

安全

使用 qm.ini 檔案中的 Security 段落來指定「物件權限管理程式 (OAM)」的選項。

ClusterQueueAccessControl=RQMName | Xmitq

設定此屬性，以檢查叢集佇列或叢集佇列管理程式上管理的完整佇列的存取控制。

RQMNAME

針對遠端管理佇列的存取控制所檢查的設定檔是具名佇列或具名佇列管理程式設定檔。

XMITQ

針對遠端管理佇列的存取控制所檢查的設定檔會解析至 SYSTEM.CLUSTER.TRANSMIT.QUEUE。

Xmitq 是預設值。

GroupModel=GlobalGroups

此屬性決定在判定 Windows 上使用者的群組成員資格時，OAM 是否檢查廣域群組。

預設值是不檢查廣域群組。

GlobalGroups

OAM 會檢查廣域群組。

設定 GlobalGroups 之後，授權指令 **setmqaut**、**dspmqaut** 及 **dmpmqaut** 會接受廣域群組名稱；請參閱 [setmqaut -g](#) 參數。

註：設定 ClusterQueueAccessControl=RQMName 並讓「授權服務」的自訂實作小於 MQZAS_VERSION_6 會導致佇列管理程式無法啟動。在此實例中，請設定 ClusterQueueAccessControl=Xmitq，或將自訂授權服務升級至 MQZAS_VERSION_6 或更高版本。

變更佇列管理程式配置資訊

這裡說明的屬性會修改個別佇列管理程式的配置。它們會置換 WebSphere MQ 的任何設定。

在 UNIX and Linux 系統上，您可以透過編輯 `qm.ini` 配置檔來修改佇列管理程式配置資訊。當您在 `qm.ini` 中定義段落時，不需要從新行開始每一個項目。您可以使用分號 (;) 或雜湊字元 (#) 來指出註解。

在 Windows 和 Linux (x86 和 x86-64 平台) 系統上，您可以使用 IBM WebSphere MQ Explorer 來修改部分配置資訊。不過，由於變更可安裝服務及其元件有重大含意，因此可安裝服務在 IBM WebSphere MQ Explorer 中是唯讀的。因此，您必須使用 Windows 上的 **regedit**，以及編輯 UNIX and Linux 上的 `qm.ini` 檔案，對可安裝服務進行任何變更。

如需變更佇列管理程式配置資訊的詳細資料，請參閱下列子主題：

相關概念

[第 5 頁的『配置』](#)

在一或多部電腦上建立一或多個佇列管理程式，並在開發、測試及正式作業系統上配置它們，以處理包含商業資料的訊息。

[第 350 頁的『變更 IBM WebSphere MQ 及佇列管理程式配置資訊』](#)

變更 IBM WebSphere MQ 或個別佇列管理程式的行為，以符合您安裝的需求。

相關工作

[規劃](#)

[管理 WebSphere MQ](#)

相關參考

[第 355 頁的『用於變更 IBM WebSphere MQ 配置資訊的屬性』](#)

在 IBM WebSphere MQ for Windows 系統及 IBM WebSphere MQ for Linux (x86 及 x86-64 平台) 系統上，使用 IBM WebSphere MQ Explorer 來修改配置資訊。在其他系統上，透過編輯 `mqs.ini` 配置檔來修改資訊。

存取模式

Access Mode 僅適用於 Windows 伺服器。AccessMode 段落由 `crtmqm` 指令上的 `-a [r]` 選項設定。在建立佇列管理程式之後，請勿變更 AccessMode 段落。

使用存取群組 (`-a [r]`) `crtmqm` 指令的選項，以指定 Windows 安全群組，其成員將獲授與所有佇列管理程式資料檔案的完整存取權。視使用的語法而定，群組可以是區域或廣域群組。群組名稱的有效語法如下：

LocalGroup

網域名稱 \ *GlobalGroup* 名稱

GlobalGroup name@網域名稱

在搭配執行 `crtmqm` 指令與 `-a [r]` 選項之前，您必須先定義其他存取群組。

如果您使用 `-ar` 而非 `-a` 來指定群組，則不會授與本端 `mqm` 群組對佇列管理程式資料檔案的存取權。如果管理佇列管理程式資料檔案的檔案系統不支援本端定義群組的存取控制項目，請使用此選項。

該群組通常是廣域安全群組，用來為多重實例佇列管理程式提供共用佇列管理程式資料和日誌資料夾的存取權。使用額外的安全存取群組可在資料夾上設定讀寫權，或共用包含的佇列管理程式資料和日誌檔。

除了使用名為 `mqm` 的本端群組來設定包含佇列管理程式資料及日誌之資料夾的許可權之外，也可以使用其他安全存取群組。與本端群組 `mqm` 不同，您可以將其他安全存取群組設為本端或廣域群組。它必須是廣域群組，才能對包含多重實例佇列管理程式所使用的資料和日誌檔的共用資料夾設定許可權。

Windows 作業系統會檢查是否有存取許可權可讀取及寫入佇列管理程式資料和日誌檔。它將檢查執行佇列管理程式程序的使用者 ID 的許可權。所檢查的使用者 ID 取決於您是將佇列管理程式作為服務啟動，還是以互動方式將其啟動。如果您以服務方式啟動佇列管理程式，則 Windows 系統所檢查的使用者 ID 是您使用準備 IBM WebSphere MQ 精靈所配置的使用者 ID。如果是互動方式啟動佇列管理程式，則 Windows 系統檢查的使用者 ID 是執行 `strmqm` 指令的使用者 ID。

使用者 ID 必須是本端 `mqm` 群組的成員，才能啟動佇列管理程式。如果該使用者 ID 是額外的安全存取群組的成員，則佇列管理程式可以使用該群組來讀取及寫入已提供許可權的檔案。

限制：您只能在 Windows 作業系統上指定額外的安全存取群組。如果在其他作業系統上指定額外的安全存取群組，則 `crtmqm` 指令會傳回錯誤。

相關概念

第 315 頁的『在 Windows 上保護未共用佇列管理程式資料及日誌目錄和檔案的安全』

第 312 頁的『在 Windows 上保護共用佇列管理程式資料及日誌目錄和檔案的安全』

相關工作

第 291 頁的『在網域工作站或伺服器上建立多重實例佇列管理程式』

相關參考

[crtmqm](#)

可安裝的服務

您可以使用 **regedit** 在 Windows 上變更可安裝的服務，以及使用 `qm.ini` 檔案中的 `Service` 段落
在 UNIX and Linux 上變更可安裝的服務。

註：變更可安裝服務及其元件有重大意義。因此，在「WebSphere MQ 探險家」中，可安裝的服務是唯讀的。

若要變更 Windows 系統上的可安裝服務，請使用 **regedit**，或在 UNIX and Linux 系統上，請使用 `qm.ini`
檔案中的 `Service` 段落。對於服務內的每一個元件，您也必須指定包含該元件之程式碼的模組名稱及路
徑。在 UNIX and Linux 系統上，請為此使用 `ServiceComponent` 段落。

名稱 =AuthorizationService|NameService

必要服務的名稱。

AuthorizationService

對於 WebSphere MQ，「授權服務」元件稱為物件權限管理程式或 OAM。

建立佇列管理程式時，會自動新增 `AuthorizationService` 段落及其相關聯的
`ServiceComponent` 段落。手動新增其他 `ServiceComponent` 段落。

NameService

依預設，不會提供名稱服務。如果您需要名稱服務，則必須手動新增 `NameService` 段落。

EntryPoints=number-of-entries

為服務定義的進入點數目。這包括起始設定和終止進入點。

SecurityPolicy=Default|NTSIDsRequired (僅限 WebSphere MQ for Windows)

只有在指定的服務是預設授權服務 (即 OAM) 時，`SecurityPolicy` 屬性才適用。`SecurityPolicy` 屬性可讓
您指定每一個佇列管理程式的安全原則。可能值包括：

Default

請使用預設安全原則來生效。如果未將特定使用者 ID 的 Windows 安全 ID (NT SID) 傳遞至 OAM，
則會嘗試搜尋相關安全資料庫來取得適當的 SID。

NTSIDsRequired

執行安全檢查時，將 NT SID 傳遞至 OAM。

如需相關資訊，請參閱 [Windows 安全 ID \(SID\)](#)。

SharedBindingsUserId=使用者類型

只有在指定的服務是預設授權服務 (即 OAM) 時，才會套用 `SharedBindingsUserId` 屬性。
`SharedBindingsUserId` 屬性僅與共用連結相關使用。此值可讓您從 `MQZ_AUTHENTICATE_USER` 函數
指定 `IdentityContext` 結構中的 `UserIdentifier` 欄位是有效使用者 ID 還是實際使用者 ID。如需
`MQZ_AUTHENTICATE_USER` 函數的相關資訊，請參閱 [MQZ_AUTHENTICATE_USER-Authenticate
user](#)。可能值包括：

Default

`UserIdentifier` 欄位的值會設為實際使用者 ID。

實數

`UserIdentifier` 欄位的值會設為實際使用者 ID。

有效

`UserIdentifier` 欄位的值設為有效使用者 ID。

FastpathBindingsUserId=使用者類型

只有在指定的服務是預設授權服務 (即 OAM) 時，FastpathBindingsUserId 屬性才適用。FastpathBindingsUserId 屬性僅與捷徑連結相關使用。此值可讓您從 MQZ_AUTHENTICATE_USER 函數指定 IdentityContext 結構中的 UserIdentifier 欄位是有效使用者 ID 還是實際使用者 ID。如需 MQZ_AUTHENTICATE_USER 函數的相關資訊，請參閱 [MQZ_AUTHENTICATE_USER-Authenticate user](#)。可能值包括：

Default

UserIdentifier 欄位的值會設為實際使用者 ID。

實數

UserIdentifier 欄位的值會設為實際使用者 ID。

有效

UserIdentifier 欄位的值設為有效使用者 ID。

IsolatedBindingsUserId =使用者類型

只有在指定的服務是預設授權服務 (即 OAM) 時，才會套用 IsolatedBindingsUserId 屬性。IsolatedBindingsUserId 屬性僅與隔離連結相關使用。此值可讓您從 MQZ_AUTHENTICATE_USER 函數指定 IdentityContext 結構中的 UserIdentifier 欄位是有效使用者 ID 還是實際使用者 ID。如需 MQZ_AUTHENTICATE_USER 函數的相關資訊，請參閱 [MQZ_AUTHENTICATE_USER-Authenticate user](#)。可能值包括：

Default

UserIdentifier 欄位的值設為有效使用者 ID。

實數

UserIdentifier 欄位的值會設為實際使用者 ID。

有效

UserIdentifier 欄位的值設為有效使用者 ID。

如需可安裝的服務和元件的相關資訊，請參閱 [UNIX、Linux 和 Windows 的可安裝服務和元件](#)。

如需一般安全服務的相關資訊，請參閱 [在 Windows 上設定安全，UNIX and Linux 系統](#)。

相關參考

[可安裝的服務參照資訊](#)

服務元件

當您新增可安裝的服務時，必須指定服務元件資訊。在 Windows 系統上使用 regedit，在 UNIX and Linux 系統上使用 qm.ini 檔案中的 ServiceComponent 段落。依預設會呈現授權服務段落，且相關聯的元件 (OAM) 處於作用中。

指定服務元件，如下所示：

服務 =service_name

必要服務的名稱。這必須符合在服務配置資訊的 Name 屬性上指定的值。

名稱 =component_name

服務元件的敘述性名稱。這必須是唯一的，且只包含對 WebSphere MQ 物件名稱有效的字元 (例如，佇列名稱)。此名稱出現在服務所產生的操作員訊息中。建議此名稱以公司商標或類似識別字串開頭。

模組 =module_name

要包含此元件之程式碼的模組名稱。這必須是完整路徑名稱。

ComponentData 大小 =大小

每次呼叫時傳遞給元件的元件資料區大小 (以位元組為單位)。如果不需要元件資料，請指定零。

如需可安裝的服務和元件的相關資訊，請參閱 [UNIX、Linux 和 Windows 的可安裝服務和元件](#)。

佇列管理程式日誌

使用「IBM WebSphere MQ 探險家」中的 Log 佇列管理程式內容頁面，或 qm.ini 檔中的 Log 段落，來指定佇列管理程式上記載的相關資訊。

依預設，這些設定繼承自指定給佇列管理程式預設日誌設定的設定 (如第 357 頁的『IBM WebSphere MQ 的日誌預設值』中所述)。僅當您想要以不同方式配置此佇列管理程式時，才變更這些設定。

如需計算日誌大小的相關資訊，請參閱第 337 頁的『計算日誌的大小』。

註：下列參數清單中提供的限制由 WebSphere MQ 設定。作業系統限制可能會減少可能的日誌大小上限。

LogPrimaryFiles = 3 | 2-254 (Windows) | 2-510 (UNIX and Linux 系統)

建立佇列管理程式時所配置的日誌檔。

在 Windows 上，您可以擁有的主要日誌檔數目下限為 2，上限為 254，在 UNIX and Linux 系統上，則為 510。預設是 3。

在 Windows 上，主要及次要日誌檔的總數不得超過 255，在 UNIX and Linux 系統上，則不得超過 511，且不得小於 3。

建立或啟動佇列管理程式時會檢查這個值。您可以在建立佇列管理程式之後變更它。不過，除非重新啟動佇列管理程式，否則值中的變更不會生效，且效果可能不會立即生效。

LogSecondary 檔案 = 2 | 1-253 (Windows) | 1-509 (UNIX and Linux 系統)

當主要檔耗盡時所配置的日誌檔。

在 Windows 上，次要日誌檔數目下限為 1，上限為 253，在 UNIX and Linux 系統上，則為 509。預設數字為 2。

在 Windows 上，主要及次要日誌檔的總數不得超過 255，在 UNIX and Linux 系統上，則不得超過 511，且不得小於 3。

當佇列管理程式啟動時，會檢查該值。您可以變更此值，但除非重新啟動佇列管理程式，否則變更不會生效，甚至可能不會立即生效。

LogFilePages = number

日誌資料保留在稱為日誌檔的一系列檔案中。日誌檔大小以 4 KB 頁面為單位指定。

預設日誌檔頁數是 4096，提供 16 MB 的日誌檔大小。

在 UNIX and Linux 系統上，日誌檔頁數下限為 64，在 Windows 上，日誌檔頁數下限為 32；在這兩種情況下，日誌檔頁數上限為 65 535。

註：無法針對佇列管理程式變更在佇列管理程式建立期間指定的日誌檔大小。

LogType=CIRCULAR | LINEAR

佇列管理程式要使用的記載類型。建立佇列管理程式之後，您無法變更要使用的記載類型。如需使用所需記載類型建立佇列管理程式的相關資訊，請參閱第 357 頁的『IBM WebSphere MQ 的日誌預設值』中 LogType 屬性的說明。

CIRCULAR

使用日誌來啟動重新啟動回復，以回復系統停止時進行中的交易。

如需循環式記載的完整說明，請參閱第 335 頁的『記載類型』。

線性

對於重新啟動回復及媒體或正向回復 (透過重播日誌內容來建立遺失或損壞的資料)。

如需線性記載的更完整說明，請參閱第 335 頁的『記載類型』。

LogBufferPages = 0 | 0-4096

配置給緩衝區記錄以進行寫入的記憶體數量，指定緩衝區大小 (以 4 KB 分頁為單位)。

緩衝區頁數下限為 18，上限為 4096。較大的緩衝區會導致更高的傳輸量，特別是對於較大的訊息。

如果您指定 0 (預設值)，佇列管理程式會選取大小。在 WebSphere MQ 7.0 版中，這是 512 (2048 KB)。

如果您指定 1 到 17 範圍內的數字，則佇列管理程式預設為 18 (72 KB)。如果您指定範圍 18 到 4096 之間的數字，佇列管理程式會使用指定的數字來設定配置的記憶體。

當佇列管理程式啟動時，會檢查該值。該值可以在所述限制內增加或減少。不過，除非下次啟動佇列管理程式，否則該值的變更不會生效。

LogPath=directory_name

佇列管理程式的日誌檔所在的目錄。這必須存在於佇列管理程式可以寫入的本端裝置上，且最好是存在於與訊息佇列不同的磁碟機上。指定不同的磁碟機會在系統失效時提供額外的保護。

預設值為：

- WebSphere MQ for Windows 中的 C:\Program Files\IBM\WebSphere MQ\log。
- WebSphere MQ for UNIX and Linux 系統中的 /var/mqm/log。

您可以使用 `-ld` 旗標在 `crtmqm` 指令上指定目錄的名稱。當建立佇列管理程式時，也會在佇列管理程式目錄下建立一個目錄，用來保留日誌檔。此目錄的名稱基於佇列管理程式名稱。這可確保日誌檔路徑是唯一的，而且符合目錄名稱長度的任何限制。

如果您未在 `crtmqm` 指令上指定 `-ld`，則會使用 `LogDefaultPath` 屬性的值。

在 WebSphere MQ for UNIX and Linux 系統中，使用者 ID `mqm` 及群組 `mqm` 必須對日誌檔具有完整權限。如果您變更這些檔案的位置，則必須自行提供這些權限。如果日誌檔位於產品隨附的預設位置中，則不需要這麼做。

LogWriteIntegrity = SingleWrite|DoubleWrite| TripleWrite

日誌程式用來可靠地寫入日誌記錄的方法。

TripleWrite

這是預設方法。

請注意，可以選取 **DoubleWrite**，但如果您這樣做，系統會將此解譯為 **TripleWrite**。

SingleWrite

只有在託管 WebSphere MQ 回復日誌的檔案系統或裝置明確地保證 4KB 寫入這一最小單位時，才應該使用 **SingleWrite**。

也就是說，如果寫入 4KB 頁面失敗，不論原因為何，僅有的兩種可能狀態為前像或後像。不會有中間狀態。

受限模式

此選項僅適用於 UNIX and Linux 系統。RestrictedMode 段落由 `crtmqm` 指令上的 `-g` 選項設定。在建立佇列管理程式之後，請勿變更此段落。如果您不使用 `-g` 選項，則不會在 `qm.ini` 檔案中建立段落。

在某些目錄下，當 IBM WebSphere MQ 應用程式連接至佇列管理程式資料目錄內的佇列管理程式時，會在這些目錄下建立檔案。為了讓應用程式在這些目錄中建立檔案，會授與它們全球寫入權：

- /var/mqm/sockets/QMgrName/@ipcc/ssem/hostname/
- /var/mqm/sockets/QMgrName/@app/ssem/hostname/
- /var/mqm/sockets/QMgrName/zsocketapp/hostname/

其中 `<QMGRNAME>` 是佇列管理程式的名稱，而 `<hostname>` 是主機名稱。

在部分系統上，無法接受授與所有使用者對這些目錄的寫入權。例如，那些不需要存取佇列管理程式的使用者。受限模式會修改儲存佇列管理程式資料之目錄的許可權。然後只有指定應用程式群組的成員才能存取目錄。用於與佇列管理程式通訊的 System V IPC 共用記憶體上的許可權也會以相同方式修改。

應用程式群組是其成員有權執行下列事項的群組名稱：

- 執行 MQI 應用程式
- 更新所有 IPCC 資源
- 變更部分佇列管理程式目錄的內容

如果要對佇列管理程式使用受限模式，請執行下列動作：

- 佇列管理程式的建立者必須在 `mqm` 群組及應用程式群組中。
- `mqm` 使用者 ID 必須在應用程式群組中。
- 所有想要管理佇列管理程式的使用者都必須位於 `mqm` 群組及應用程式群組中。
- 所有想要執行 IBM WebSphere MQ 應用程式的使用者都必須在應用程式群組中。

由不在應用程式群組中的使用者所發出的任何 `MQCONN` 或 `MQCONN` 呼叫失敗，原因碼為 `MQRC_Q_MGR_NOT_AVAILABLE`。

受限模式與 IBM WebSphere MQ 授權服務一起運作。因此，您也必須授與使用者連接至 IBM WebSphere MQ 的權限，並使用 IBM WebSphere MQ 授權服務來存取他們需要的資源。

Windows **UNIX** **Linux** 如需配置 IBM WebSphere MQ 授權服務的進一步資訊，請參閱在 Windows、UNIX and Linux 系統上設定安全。

只有在授權服務提供的控制項未提供足夠的佇列管理程式資源隔離時，才會使用 IBM WebSphere MQ 受限模式。

XA 資源管理程式

請使用「IBM WebSphere MQ 探險家」中的 XA resource manager 佇列管理程式內容頁面，或 qm.ini 檔中的 XAResourceManager 段落，來指定由佇列管理程式協調之廣域工作單元所涉及的下列資源管理程式相關資訊。

針對參與廣域工作單元的每一個資源管理程式實例，手動新增 XA 資源管理程式配置資訊；不提供預設值。

如需資源管理程式屬性的相關資訊，請參閱 [資料庫協調](#)。

名稱 =name (必要)

這個屬性識別資源管理程式實例。

Name 值的長度最多可以為 31 個字元。您可以使用資源管理程式的名稱，如其 XA 交換器結構中所定義。不過，如果您使用相同資源管理程式的多個實例，則必須為每一個實例建構唯一名稱。例如，您可以透過在 Name 字串中包括資料庫名稱來確保唯一性。

WebSphere MQ 會在訊息及 dspmqtrn 指令的輸出中使用 Name 值。

一旦相關聯的佇列管理程式已啟動且資源管理程式名稱生效，請勿變更資源管理程式實例的名稱，或從配置資訊中刪除其項目。

SwitchFile=name (必要)

包含資源管理程式的 XA 交換器結構之載入檔的完整名稱。

如果您將 64 位元佇列管理程式與 32 位元應用程式搭配使用，則 name 值應該只包含載入檔案的基本名稱，該載入檔案包含資源管理程式的 XA 交換器結構。

32 位元檔案將從 ExitsDefaultPath 指定的路徑載入至應用程式。

64 位元檔案將從 ExitsDefaultPath64 指定的路徑載入至佇列管理程式。

XAOpenString=string (選用)

要傳遞至資源管理程式 xa_open 進入點的資料字串。字串的內容取決於資源管理程式本身。例如，字串可以識別這個資源管理程式實例要存取的資料庫。如需定義此屬性的相關資訊，請參閱：

- [新增 DB2 的資源管理程式配置資訊](#)
- [新增 Oracle 的資源管理程式配置資訊](#)
- [新增 Sybase](#)
- [新增 Informix 的資源管理程式配置資訊](#)

並請參閱您的資源管理程式文件，以取得適當的字串。

XACloseString=string (選用)

要傳遞至資源管理程式 xa_close 進入點的資料字串。字串的內容取決於資源管理程式本身。如需定義此屬性的相關資訊，請參閱：

- [新增 DB2 的資源管理程式配置資訊](#)
- [新增 Oracle 的資源管理程式配置資訊](#)
- [新增 Sybase](#)
- [新增 Informix 的資源管理程式配置資訊](#)

並請參閱資料庫文件，以取得適當的字串。

ThreadOfControl=THREAD | PROCESS

此屬性是 WebSphere MQ for Windows 的必要屬性。當佇列管理程式需要從它自己的多執行緒處理程序之一呼叫資源管理程式時，它會使用這個值來進行序列化。

執行緒

資源管理程式完全可察覺執行緒。在多執行緒 WebSphere MQ 程序中，可以同時從多個執行緒對外部資源管理程式發出 XA 函數呼叫。

PROCESS

資源管理程式不是安全執行緒。在多執行緒 WebSphere MQ 程序中，一次只能對資源管理程式進行一次 XA 函數呼叫。

ThreadOfControl 項目不適用於多執行緒應用程式程序中佇列管理程式所發出的 XA 函數呼叫。一般而言，在不同執行緒上具有並行工作單元的應用程式需要每一個資源管理程式都支援此作業模式。

通道段落的屬性

這些屬性決定通道的配置。

此資訊不適用於 WebSphere MQ for the z/OS 平台。

使用「WebSphere MQ 探險家」中的 Channels 佇列管理程式內容頁面，或 qm.ini 檔案中的 CHANNELS 段落，來指定通道的相關資訊。

MaxChannels=100 | number

容許的現行通道數上限。

值必須在 1-65535 範圍內。預設值為 100。

MaxActive 通道 =MaxChannels_value

容許隨時作用中的通道數上限。預設值是指定給 MaxChannels 屬性的值。

MaxInitiators=3 | number

起始器數目上限。預設及上限值為 3。

MQIBindType= FASTPATH | STANDARD

應用程式的連結：

Fastpath

通道使用 MQCONN FASTPATH 進行連接；沒有代理程式處理程序。

標準

使用 STANDARD 連接通道。

PipeLine 長度 =1 | number

通道將使用的並行執行緒數目上限。預設值是 1。任何大於 1 的值都會被視為 2。

當您使用管線化時，請配置通道兩端的佇列管理程式，使 PipeLine 長度大於 1。

註：管線化只對 TCP/IP 通道有效。

AdoptNewMCA=N0| SVR | SDR | RCVR | CLUSRCVR | ALL | FASTPATH

如果 WebSphere MQ 收到啟動通道的要求，但發現通道實例已在執行中，在某些情況下，必須先停止現有的通道實例，才能啟動新的通道實例。AdoptNewMCA 屬性可讓您控制以這種方式結束哪些類型的通道。

如果您指定特定通道類型的 AdoptNewMCA 屬性，但由於相符通道實例已在執行中，新通道無法啟動：

1. 新通道會要求結束，以嘗試停止前一個通道。
2. 如果在 AdoptNewMCATimeout 等待間隔到期之前，前一個通道伺服器未回應此要求，則會結束前一個通道伺服器的執行緒或處理程序。
3. 如果前一個通道伺服器在步驟 2 之後未結束，且在 AdoptNewMCATimeout 等待間隔第二次到期之後，WebSphere MQ 會結束通道並發生 CHANNEL IN USE 錯誤。

AdoptNewMCA 功能適用於伺服器、傳送端、接收端及叢集接收端通道。如果是傳送端或伺服器通道，在接收端佇列管理程式中只能執行一個具有特定名稱的通道實例。如果是接收端或叢集接收端通道，則具有特定名稱之通道的多個實例可能正在接收端佇列管理程式中執行，但從特定遠端佇列管理程式一次只能執行一個實例。

註: AdoptNewMCA 在要求端或伺服器連線通道上不受支援。

從下列清單中指定一或多個值 (以逗點或空白區隔):

NO

不需要 AdoptNewMCA 特性。這是預設值。

SVR

採用伺服器通道。

SDR

採用傳送端通道。

RCVR

採用接收端通道。

CLUSRCVR

採用叢集接收端通道。

ALL

採用 FASTPATH 通道以外的所有通道類型。

Fastpath

如果通道是 FASTPATH 通道，請採用該通道。只有在同時指定適當的通道類型時，才會發生這種情況，例如: AdoptNewMCA=RCVR,SVR,FASTPATH。

注意! : AdoptNewMCA 屬性可能會使用 FASTPATH 通道以無法預期的方式運作。對 FASTPATH 通道啟用 AdoptNewMCA 屬性時，請格外小心。

AdoptNewMCATimeout=60 | 1-3600

新通道實例等待舊通道實例結束的時間量 (秒)。請指定範圍在 1 - 3600 的值。預設值是 60。

AdoptNewMCACheck = QM | ADDRESS | NAME | ALL

啟用 AdoptNewMCA 屬性時所需的檢查類型。可能的話，請執行完整檢查，以保護您的通道不會被意外或惡意關閉。至少檢查通道名稱是否相符。

如果是 QM、NAME 或 ALL，請指定下列一或多個值 (以逗點或空白區隔):

QM

請檢查佇列管理程式名稱是否相符。

請注意，佇列管理程式名稱本身是相符的，而不是 QMID。

ADDRESS

請檢查通訊來源 IP 位址。例如，TCP/IP 位址。

註: 以逗點區隔的 CONNAME 值適用於目標位址，因此與此選項無關。

如果多重實例佇列管理程式從 hosta 失效接手至 hostb，則來自該佇列管理程式的任何出埠通道都會使用來源 IP 位址 hostb。如果這與 hosta 不同，則 AdoptNewMCACheck=ADDRESS 無法符合。

您可以使用 SSL 或 TLS 搭配交互鑑別，以防止攻擊者中斷現有的執行中通道。或者，使用具有 IP 接管的 HACMP 類型解決方案，而非多重實例佇列管理程式，或使用網路負載平衡器來遮罩來源 IP 位址。

名稱

請檢查通道名稱是否相符。

ALL

請檢查是否有相符的佇列管理程式名稱、通訊位址，以及是否有相符的通道名稱。

預設值為 AdoptNewMCACheck=NAME, ADDRESS, QM。

相關概念

第 48 頁的『通道狀態』

通道隨時可以處於許多狀態之一。部分狀態也有子狀態。從給定狀態，通道可以移至其他狀態。

TCP、LU62、NETBIOS 及 SPX

請使用這些佇列管理程式內容頁面或 qm.ini 檔案中的段落，來指定網路通訊協定配置參數。它們會置換通道的預設屬性。

TCP

使用「IBM WebSphere MQ 探險家」的 TCP 佇列管理程式內容頁面，或 qm.ini 檔中的 TCP 段落，來指定傳輸控制通訊協定/Internet Protocol (TCP/IP) 配置參數。

埠 =1414|port_number

TCP/IP 階段作業的預設埠號 (以十進位表示法表示)。WebSphere MQ 的已知埠號是 1414。

Library1 =DLLName1 (僅限 WebSphere MQ for Windows)

TCP/IP Socket DLL 的名稱。

預設值是 WSOCK32。

KeepAlive=NO|YES

開啟或關閉 KeepAlive 函數。KeepAlive=YES 會使 TCP/IP 定期檢查連線的另一端是否仍然可用。如果不是，則會關閉通道。

ListenerBacklog= number

置換 TCP/IP 接聽器的預設未完成要求數。

在 TCP/IP 上接收時，會設定未完成的連線要求數目上限。這可以視為在 TCP/IP 埠上等待接聽器接受要求之要求的待辦事項。預設接聽器待辦事項值顯示在 [第 369 頁的表 33](#) 中。

平台	預設 ListenerBacklog 值
Windows 伺服器	100
Windows 工作站	5
Linux	100
Solaris	100
HP-UX	20
AIX V4.2 或更新版本	100
AIX V4.1 或更舊版本	10

註: 部分作業系統支援大於所顯示預設值的值。使用此選項可避免達到連線限制。

相反地，部分作業系統可能會限制 TCP 待辦事項的大小，因此有效的 TCP 待辦事項可能小於這裡所要求的。

如果待辦事項達到 [第 369 頁的表 33](#) 中顯示的值，則會拒絕 TCP/IP 連線，且通道無法啟動。對於訊息通道，這會導致通道進入 RETRY 狀態，並稍後重試連線。對於用戶端連線，用戶端會從 MQCONN 收到 MQRC_Q_MGR_NOT_AVAILABLE 原因碼，並稍後重試連線。

SvrSndBuffSize=32768|number

用戶端連線伺服器連線通道的伺服器端所使用的 TCP/IP 傳送緩衝區大小 (以位元組為單位)。

SvrRcvBuffSize=32768|number

用戶端連線伺服器連線通道的伺服器端所使用的 TCP/IP 接收緩衝區大小 (以位元組為單位)。

連線逾時 =0|number

嘗試連接 Socket 逾時之前的秒數。預設值零指定沒有連接逾時。

LU62 (僅限 WebSphere MQ for Windows)

使用「IBM WebSphere MQ 探險家」中的 LU6.2 佇列管理程式內容頁面，或 qm.ini 檔中的 LU62 段落，來指定 SNA LU 6.2 通訊協定配置參數。

TPName

要在遠端站台上啟動的 TP 名稱。

Library1 =DLLName 1

APPC DLL 的名稱。

預設值為 WCPIC32。

Library2 =DLLName2

與 Library1 相同，用於將程式碼儲存在兩個個別的程式庫中。

預設值為 WCPIC32。

NETBIOS (僅限 WebSphere MQ for Windows)

使用「IBM WebSphere MQ 探險家」中的 Netbios 佇列管理程式內容頁面，或 qm.ini 檔中的 NETBIOS 段落，來指定 NetBIOS 通訊協定配置參數。

LocalName=name

LAN 上用來識別此機器的名稱。

AdapterNum=@|adapter_number

LAN 配接卡的號碼。預設值為配接卡 0。

NumSess=1|number_of_sessions

要配置的階段作業數目。預設值是 1。

NumCmds=1|number_of_commands

要配置的指令數。預設值是 1。

NumNames=1|number_of_names

要配置的名稱數目。預設值是 1。

Library1 =DLLName1

NetBIOS DLL 的名稱。

預設值為 NETAPI32。

SPX (僅限 WebSphere MQ for Windows)

使用「IBM WebSphere MQ 探險家」中的 SPX 佇列管理程式內容頁面，或 qm.ini 檔中的 SPX 段落，來指定 SPX 通訊協定配置參數。

Socket =5E86|socket_number

以十六進位表示法表示的 SPX Socket 號碼。預設值為 X'5E86'。

BoardNum=@|adapter_number

LAN 配接卡號碼。預設值為配接卡 0。

KeepAlive= NO | YES

開啟或關閉 KeepAlive 函數。

KeepAlive=YES 會導致 SPX 定期檢查連線的另一端是否仍然可用。如果不是，則會關閉通道。

Library1 =DLLName1

SPX DLL 的名稱。

預設值為 WSOCK32.DLL。

Library2 =DLLName2

與 LibraryName1 相同，用於將程式碼儲存在兩個個別的程式庫中。

預設值為 WSOCK32.DLL。

ListenerBacklog= number

置換 SPX 接聽器的預設未完成要求數。

在 SPX 上接收時，會設定未完成的連線要求數目上限。這可以視為在 SPX Socket 上等待接聽器接受要求的要求待辦事項。預設接聽器待辦事項值顯示在 [第 370 頁的表 34](#) 中。

表 34: 預設未完成的連線要求 (SPX)	
平台	預設 ListenerBacklog 值
Windows 伺服器	100

表 34: 預設未完成的連線要求 (SPX) (繼續)	
平台	預設 ListenerBacklog 值
Windows 工作站	5

註: 部分作業系統支援大於所顯示預設值的值。使用此選項可避免達到連線限制。

相反地, 部分作業系統可能會限制 SPX 待辦事項的大小, 因此有效的 SPX 待辦事項可能小於這裡所要求的。

如果待辦事項達到第 370 頁的表 34 中顯示的值, 則會拒絕 SPX 連線, 且通道無法啟動。對於訊息通道, 這會導致通道進入 RETRY 狀態, 並稍後重試連線。對於用戶端連線, 用戶端會收到來自 MQCONN 的 MQRC_Q_MGR_NOT_AVAILABLE 原因碼, 應該稍後重試連線。

結束程式路徑

使用「IBM WebSphere MQ 探險家」中的 Exits 佇列管理程式內容頁面, 或 qm.ini 檔中的 ExitPath 段落, 來指定佇列管理程式系統上使用者結束程式的路徑。

ExitsDefault 路徑 =字串

ExitsDefault 路徑屬性指定下列項目的位置:

- 用戶端的 32 位元通道結束程式
- 伺服器的 32 位元通道結束程式及資料轉換結束程式
- 不完整的 XA 交換器載入檔案

ExitsDefaultPath64 =string

ExitsDefaultPath64 屬性指定下列項目的位置:

- 用戶端的 64 位元通道結束程式
- 伺服器的 64 位元通道結束程式及資料轉換結束程式
- 不完整的 XA 交換器載入檔案

API 結束程式

若為伺服器, 請使用 IBM WebSphere MQ Explorer 中的 Exits 佇列管理程式內容頁面, 或 qm.ini 檔案中的 ApiExitLocal 段落, 來識別佇列管理程式的 API 結束常式。若為用戶端, 請修改 mqclient.ini 檔中的 ApiExitLocal 段落, 以識別佇列管理程式的 API 結束常式。

在 Windows 系統上, 您也可以使用 amqmdain 指令來變更 API 結束程式的項目。(若要識別所有佇列管理程式的 API 結束常式, 請使用 ApiExitCommon 及 ApiExitTemplate 段落, 如第 359 頁的『API 結束程式』中所述。)

請注意, 為了讓 API 結束程式正確運作, 來自伺服器的訊息必須傳送至未轉換的用戶端。在 API 結束程式已處理訊息之後, 必須在用戶端上轉換訊息。因此, 這需要您已在用戶端上安裝所有轉換結束程式。

如需這些段落的屬性完整說明, 請參閱 [配置 API 結束程式](#)。

UNIX, Linux, and Windows 上的 QMErrorLog 段落

使用「WebSphere MQ 探險家」中的 Extended 佇列管理程式內容頁面, 或 qm.ini 檔中的 QMErrorLog 段落, 來自訂佇列管理程式錯誤日誌的作業及內容。



小心: 只有在 Windows 平台上使用本端佇列管理程式時, 才能使用「WebSphere MQ 探險家」來進行變更。

ErrorLog 大小 =maxsize

指定複製到備份的佇列管理程式錯誤日誌大小。maxsize 必須在 32768 到 2147483648 位元組的範圍內。如果未指定 ErrorLogSize, 則會使用預設值 2097152 個位元組 (2 MB)。

ExcludeMessage=msgIds

指定不要寫入佇列管理程式錯誤日誌的訊息。如果大量使用 WebSphere MQ 系統, 且有許多通道在停止及啟動中, 則會將大量參考訊息傳送至 z/OS 主控台和印副本日誌。WebSphere MQ-IMS 橋接器及緩衝

區管理程式也可能會產生大量參考訊息，因此排除訊息可防止您在需要時接收大量訊息。 *msgIds* 包含下列訊息 ID 的逗點區隔清單：

- 5211-已超出內容名稱長度上限。
- 5973-禁止分散式發佈/訂閱訂閱
- 5974-禁止分散式發佈/訂閱發佈
- 6254-系統無法動態載入共用程式庫
- 7234 - 載入的訊息數
- 9001 - 通道程式正常結束
- 9002 - 通道程式已啟動
- 9202 - 遠端主機無法使用
- 9208-從主機接收時發生錯誤
- 9209-連線已關閉
- 9228-無法啟動通道回應者
- 9489-已超出 SVRCONN 上限實例限制
- 9490-已超出每個用戶端的 SVRCONN 實例數上限
- 9508-無法連接佇列管理程式
- 9524 - 遠端佇列管理程式無法使用
- 9528 - 使用者要求結束通道
- 9558-遠端通道無法使用
- 9637-通道缺少憑證
- 9776-使用者 ID 已封鎖通道
- 9777-NOACCESS 對映已封鎖通道
- 9782-位址已封鎖連線
- 9999 - 通道程式異常結束

SuppressMessage=msgIds

指定在指定時間間隔內只寫入佇列管理程式錯誤日誌一次的訊息。如果大量使用 WebSphere MQ 系統，且有許多通道在停止及啟動中，則會將大量參考訊息傳送至 z/OS 主控台和印副本日誌。WebSphere MQ-IMS 橋接器及緩衝區管理程式也可能產生大量參考訊息，因此抑制訊息可防止您在需要時接收許多重複訊息。時間間隔由 *SuppressInterval* 指定。 *msgIds* 包含下列訊息 ID 的逗點區隔清單：

- 5211-已超出內容名稱長度上限。
- 5973-禁止分散式發佈/訂閱訂閱
- 5974-禁止分散式發佈/訂閱發佈
- 6254-系統無法動態載入共用程式庫
- 7234 - 載入的訊息數
- 9001 - 通道程式正常結束
- 9002 - 通道程式已啟動
- 9202 - 遠端主機無法使用
- 9208-從主機接收時發生錯誤
- 9209-連線已關閉
- 9228-無法啟動通道回應者
- 9489-已超出 SVRCONN 上限實例限制
- 9490-已超出每個用戶端的 SVRCONN 實例數上限
- 9508-無法連接佇列管理程式
- 9524 - 遠端佇列管理程式無法使用
- 9528 - 使用者要求結束通道
- 9558-遠端通道無法使用
- 9637-通道缺少憑證
- 9776-使用者 ID 已封鎖通道
- 9777-NOACCESS 對映已封鎖通道
- 9782-位址已封鎖連線
- 9999 - 通道程式異常結束

如果在 SuppressMessage 及 ExcludeMessage 中指定相同的訊息 ID，則會排除該訊息。

SuppressInterval=length

指定 SuppressMessage 中指定的訊息只寫入佇列管理程式錯誤日誌一次的時間間隔 (秒)。length 必須在 1 到 86400 秒的範圍內。如果未指定 SuppressInterval，則會使用預設值 30 秒。

佇列管理程式預設連結類型

請使用「IBM WebSphere MQ 探險家」中的 Extended 佇列管理程式內容頁面，或 qm.ini 檔中的 Connection 段落，來指定預設連結類型。

DefaultBindType =SHARED|ISOLATED

如果 DefaultBind 類型設為 ISOLATED，則應用程式及佇列管理程式會在個別處理程序中執行，且不會在它們之間共用任何資源。

如果 DefaultBind 類型設為 SHARED，則應用程式及佇列管理程式會在個別處理程序中執行，但在它們之間共用部分資源。

預設值為 shared。

佇列管理程式配置檔的 SSL 及 TLS 段落

請使用佇列管理程式配置檔的 SSL 段落，在佇列管理程式上配置 SSL 或 TLS 通道。

線上憑證狀態通訊協定 (OCSP)

憑證可以包含 AuthorityInfoAccess 延伸。此延伸規格指定要透過「線上憑證狀態通訊協定 (OCSP)」來聯絡的伺服器。若要容許佇列管理程式上的 SSL 或 TLS 通道使用 AuthorityInfoAccess 延伸，請確保其中命名的 OCSP 伺服器可用、配置正確且可透過網路存取。如需相關資訊，請參閱 [使用已撤銷的憑證](#)。

Crldistribution 點 (CDP)

憑證可以包含 Crldistribution 點延伸。此延伸包含 URL，可識別用來下載憑證撤銷清冊 (CRL) 的通訊協定，以及要聯絡的伺服器。

如果您要容許佇列管理程式上的 SSL 或 TLS 通道使用 Crldistribution 點延伸，請確保其中所指名的 CDP 伺服器可用、正確配置且可透過網路存取。

SSL 段落

使用 qm.ini 檔案中的 SSL 段落來配置佇列管理程式上 SSL 或 TLS 通道如何嘗試使用下列機能，以及在使用它們時發生問題時如何反應。

在下列每一種情況下，如果提供的值不是列出的其中一個有效值，則會採用預設值。未寫入任何錯誤訊息，指出指定無效值。

CDPCheckExtensions=YES|NO

CDPCheckExtensions 指定此佇列管理程式上的 SSL 或 TLS 通道是否嘗試檢查在 Crldistribution 點憑證延伸中命名的 CDP 伺服器。

- YES: SSL 或 TLS 通道嘗試檢查 CDP 伺服器，以判定是否撤銷數位憑證。
- NO: SSL 或 TLS 通道不會嘗試檢查 CDP 伺服器。此值是預設值。

OCSPAAuthentication=REQUIRED|WARN|OPTIONAL

OCSPAAuthentication 指定當無法從 OCSP 伺服器判定撤銷狀態時要採取的動作。

如果已啟用 OCSP 檢查，則 SSL 或 TLS 通道程式會嘗試聯絡 OCSP 伺服器。

如果通道程式無法聯絡任何 OCSP 伺服器，或如果沒有伺服器可以提供憑證的撤銷狀態，則會使用 OCSPAAuthentication 參數的值。

- REQUIRED: 無法判斷撤銷狀態會導致連線因錯誤而關閉。此值是預設值。
- 警告: 無法判定撤銷狀態會導致在佇列管理程式錯誤日誌中寫入警告訊息，但容許繼續進行連線。

- OPTIONAL: 無法判定撤銷狀態，容許以無聲自動方式繼續連線。未提供任何警告或錯誤。

OCSPCheckExtensions=YES|NO

OCSPCheckExtensions 指定此佇列管理程式上的 SSL 及 TLS 通道是否嘗試檢查在 AuthorityInfoAccess 憑證延伸中命名的 OCSP 伺服器。

- YES: SSL 和 TLS 通道會嘗試檢查 OCSP 伺服器，以判斷數位憑證是否已撤銷。此值是預設值。
- NO: SSL 和 TLS 通道不會嘗試檢查 OCSP 伺服器。

SSLHTTPProxyName=string

字串是 GSKit 為 OCSP 檢查要使用之 HTTP Proxy 伺服器的主機名稱或網址。此位址後面可以接著以括弧括住的選用埠號。如果您未指定埠號，則會使用預設 HTTP 埠 80。在 HP-UX PA-RISC 和 Sun Solaris SPARC 平台上，對於 AIX 上的 32 位元用戶端，網址只能是 IPv4 位址；在其他平台上，它可以是 IPv4 或 IPv6 位址。

例如，如果防火牆阻止存取 OCSP 回應端的 URL，則可能需要此屬性。

結束內容

使用 IBM WebSphere MQ Explorer 中的「叢集佇列管理程式內容」頁面，或 qm.ini 檔中的 ExitPropertiesLocal 段落，來指定佇列管理程式上結束內容的相關資訊。或者，您可以使用 **amqmdain** 指令來設定它。

依預設，此設定繼承自機器層面配置（在第 356 頁的『結束內容』中說明）的 ExitProperties 段落中的 CLWLMode 屬性。只有在您想要以不同方式配置此佇列管理程式時，才變更此設定。對於個別佇列管理程式，可以使用「叢集佇列管理程式內容」頁面上的叢集工作量模式屬性來置換此值。

CLWLMode=SAFE|FAST

叢集工作量 (CLWL) 結束程式可讓您指定要在叢集中開啟哪個叢集佇列，以回應 MQI 呼叫（例如，MQOPEN、MQPUT）。CLWL 結束程式會根據您在 CLWLMode 屬性上指定的值，以 FAST 模式或 SAFE 模式執行。如果您省略 CLWLMode 屬性，則叢集工作量結束程式會以 SAFE 模式執行。

SAFE

在與佇列管理程式不同的處理程序中執行 CLWL 結束程式。這是預設值。

以 SAFE 模式執行時，如果使用者撰寫的 CLWL 結束程式發生問題，則會發生下列情況：

- CLWL 伺服器處理程序 (amqzlw0) 失敗。
- 佇列管理程式會重新啟動 CLWL 伺服器處理程序。
- 錯誤日誌中會向您報告此錯誤。如果 MQI 呼叫正在進行中，您會收到回覆碼形式的通知。

佇列管理程式的完整性會保留下來。

註：在個別處理程序中執行 CLWL 結束程式可能會影響效能。

FAST

在佇列管理程式處理程序中行內執行叢集結束程式。

指定此選項可避免與在 SAFE 模式中執行相關聯的處理程序切換成本，從而增進效能，但會以犧牲佇列管理程式完整性為代價。只有在您確信 CLWL 結束程式 **沒有** 問題，且您特別關心效能時，才應該以 FAST 模式執行 CLWL 結束程式。

當 CLWL 結束程式以 FAST 模式執行時，如果發生問題，佇列管理程式會失敗，且您會面臨佇列管理程式完整性受損的風險。

子儲存區

此段落由 WebSphere MQ 建立。請不要變更它。

當您建立佇列管理程式時，WebSphere MQ 會自動寫入段落子儲存區及該段落內的屬性 ShortSubpool 名稱。WebSphere MQ 會選擇 ShortSubpool 名稱的值。請勿變更此值。

該名稱對應於在 /var/mqm/sockets 目錄內建立的目錄及符號鏈結，WebSphere MQ 會使用該目錄及符號鏈結，在其執行中處理程序之間進行內部通訊。

配置 HP Integrity NonStop Server

使用此資訊可協助您配置 IBM WebSphere MQ 用戶端以進行 HP Integrity NonStop Server 安裝。

如需使用配置檔來配置用戶端的詳細資料，請參閱 [第 107 頁的『使用配置檔來配置用戶端』](#)。

如需使用環境變數來配置用戶端的詳細資料，請參閱 [第 121 頁的『使用 WebSphere MQ 環境變數』](#)。

如果您在 TMF/ 閘道下執行 HP Integrity NonStop Server 作業的 IBM WebSphere MQ 用戶端，請參閱子主題，以取得如何配置 TMF/ 閘道的相關資訊。包括「閘道」處理程序的概觀、將「閘道」配置為在 Pathway 下執行，以及配置用戶端起始設定檔以讓 IBM WebSphere MQ Client for HP Integrity NonStop Server 能夠呼叫到「TMF 閘道」。

本節也包含 IBM WebSphere MQ 用戶端，以取得 HP Integrity NonStop Server 關於授與通道許可權的特定資訊。

閘道處理程序概觀

HP NonStop 交易管理機能 (TMF) 提供服務，可讓閘道程序登錄為資源管理程式。IBM WebSphere MQ 提供的 TMF/ 閘道處理程序在 Pathway 下執行。

IBM WebSphere MQ 會針對每一個由 TMF 協調的佇列管理程式登錄單一閘道處理程序，因此您必須為每一個要參與 TMF 協調工作單元的佇列管理程式配置個別的 TMF/ 閘道。這項登錄是為了讓每一個佇列管理程式都是獨立的資源管理程式，且為了管理目的，在 HP NonStop TMF 中登錄每一個佇列管理程式一次，會產生易於瞭解的對映。

對於多個 IBM WebSphere MQ 安裝，您必須從其中一個安裝中指定單一閘道處理程序，以讓每一個佇列管理程式由 TMF 協調。

「閘道」處理程序的介面支援相同版本或更早版本的任何用戶端。

如需管理閘道程序的相關資訊，請參閱 [管理 HP Integrity NonStop Server](#)。

配置閘道以在 Pathway 下執行

TMF/ 閘道是 HP NonStop 交易管理機能 (TMF) 與 IBM WebSphere MQ 之間的介面，可讓 TMF 成為 IBM WebSphere MQ 交易的交易協調程式。

IBM WebSphere MQ 提供的 TMF/ 閘道會將交易從 TMF 協調轉換為 eXtended Architecture (XA) 交易協調，以與遠端佇列管理程式進行通訊。

每個佇列管理程式必須有一個需要協調的 TMF/ 閘道，且需要用戶端配置，用戶端才能連接至正確的閘道。

TMF/ 閘道可以使用用戶端可用的所有機制來與佇列管理程式進行通訊。以您針對其他應用程式的方式來配置 TMF/ 閘道。

TMF/ 閘道不是 HP Integrity NonStop Server 處理程序配對，而且設計為在 Pathway 環境中執行。「TMF/ 閘道」會在 TMF 內建立永久資源，並在後續執行時重複使用，因此「TMF/ 閘道」必須一律在相同的使用者權限下執行。

定義 serverclass

TMF/ 閘道作為 Pathway 環境內的 serverclass 進行管理。如果要定義 serverclass，您必須設定下列伺服器屬性：

PROCESSTYPE=OSS

指定 serverclass 中的伺服器類型。「閘道」處理程序是多執行緒 OSS 程式。此屬性是必要的，且必須設為 OSS。

MAXSERVERS=1

指定這個 serverclass 中可同時執行的伺服器處理程序數目上限。任何佇列管理程式只能有單一「閘道」處理程序。此屬性是必要的，且必須設為 1。

NUMSTATIC=1

指定這個 serverclass 內的靜態伺服器數目上限。「閘道」程序必須作為靜態伺服器執行。此屬性是必要的，且必須設為 1。

TMF=ON

指定這個 serverclass 中的伺服器是否可以鎖定及更新 TMF 子系統所審核的資料檔。「閘道」處理程序會參與 IBM WebSphere MQ 用戶端應用程式的 TMF 交易，因此此屬性必須設為 ON。

PROGRAM=<OSS installation path>/opt/mqm/bin/runmqtmf

對於 IBM WebSphere MQ 的 IBM WebSphere MQ 用戶端，此屬性必須是 runmqtmf。此屬性必須是絕對 OSS 路徑名稱。觀察值很重要。

ARGLIST=-m<QMgr name>[-c<channel name>][,-p<port>][,-h<host name>][,-n<max threads>]

這些屬性提供參數給「閘道」處理程序，其中：

- QMgrName 是此「閘道」處理程序的佇列管理程式名稱。如果您使用佇列共用群組（或其他埠配送技術），則此參數必須以特定佇列管理程式為目標。此為必要參數。
- 通道名稱 是佇列管理程式上由「閘道」處理程序使用的伺服器通道名稱。這是選用的參數。
- port 是佇列管理程式的 TCP/IP 埠。這是選用的參數。
- host name 是佇列管理程式的主機名稱。這是選用的參數。
- max threads 是「閘道」處理程序所建立的工作程式執行緒數目上限。此參數的值可以為 10 或更大。即使指定低於 10 的值，也會使用最低值 10。如果未提供任何值，則「閘道」處理程序最多會建立 50 個執行緒。

除了第 376 頁的『使用環境變數配置 TMF/ 閘道』中說明的資訊之外，請使用 -c、-p 及 -h 屬性作為替代方法，以提供「閘道」的連線資訊。如果您指定一個以上但並非所有 -c、-p 及 -h 屬性，則您未指定的那些屬性會預設為下列值：

- 通道名稱 預設為 SYSTEM.DEF.SVRCONN
- 主機名稱 預設為 localhost
- port 預設為 1414

如果您提供的任何參數無效，則 TMF/ 閘道會向錯誤日誌發出診斷訊息 [AMQ5379](#) 並終止。

OWNER=ID

用來執行「閘道」且必須授與佇列管理程式連接權限的使用者 ID。

SECURITY="value"

指定可以從 IBM WebSphere MQ 用戶端應用程式存取 Gateway 的使用者（與 Owner 屬性相關）。

LINKDEPTH 和 MAXLINKS 必須配置適當的值，以符合可能想要與閘道同時通訊的預期 IBM WebSphere MQ 用戶端應用程式數目。如果這些值設定太低，您可能看到從用戶端應用程式發出的錯誤訊息 [AMQ5399](#)。

如需這些伺服器屬性的相關資訊，請參閱 *HP NonStop TS/MP 2.5 系統管理手冊*。

使用環境變數配置 TMF/ 閘道

定義 TMF/ 閘道最常用的方法之一是設定 MQSERVER 環境變數，例如：

```
ENV MQSERVER=<channel name>/<transport>/<host name>(<listener port>)
```

指令開頭的 ENV 是 Pathway 表示法。

配置用戶端起始設定檔

如果您使用 HP NonStop 交易管理機能 (TMF)，則必須具有 IBM WebSphere MQ 用戶端起始設定檔，才能讓 IBM WebSphere MQ 用戶端讓 HP Integrity NonStop Server 呼叫到「TMF 閘道」。

HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端起始設定檔可以保留在許多位置中，如需相關資訊，請參閱第 109 頁的『用戶端配置檔的位置』。

如需配置檔內容的詳細資料，以及範例，請參閱第 107 頁的『使用配置檔來配置用戶端』。使用 TMF 段落來指定 TMF 佇列管理程式及伺服器詳細資料，如需相關資訊，請參閱第 120 頁的『TMF 及 TMF/ 閘道段落』。

HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端項目範例如下:

```
TMF:
  PathMon=$PSD1P

TmfGateway:
  QManager=MQ5B
  Server=MQ-MQ5B

TmfGateway:
  QManager=MQ5C
  Server=MQ-MQ5C
```

如需使用環境變數來配置用戶端的相關資訊，請參閱 [第 121 頁的『使用 WebSphere MQ 環境變數』](#)。

將許可權授與通道

將許可權授與 HP Integrity NonStop Server 的 IBM WebSphere MQ 用戶端上的通道與其他作業系統相同，不過您必須知道閘道執行所在之擁有者的識別。

然後，您可以使用閘道擁有者的識別來授與適當的許可權。重要的差異是授與佇列管理程式通道的許可權不在任何應用程式的權限之下。

請使用 **setmqaut** 指令來授與授權，亦即，授與 IBM WebSphere MQ 主體或使用者群組執行作業的許可權，以及撤銷授權，亦即移除執行作業的許可權。

注意事項

本資訊係針對 IBM 在美國所提供之產品與服務所開發。

在其他國家中，IBM 可能不會提供本書中所提的各項產品、服務或功能。請洽當地 IBM 業務代表，以取得當地目前提供的產品和服務之相關資訊。這份文件在提及 IBM 的產品、程式或服務時，不表示或暗示只能使用 IBM 的產品、程式或服務。只要未侵犯 IBM 的智慧財產權，任何功能相當的產品、程式或服務都可以取代 IBM 的產品、程式或服務。不過，任何非 IBM 的產品、程式或服務，使用者必須自行負責作業的評估和驗證責任。

本文件所說明之主題內容，IBM 可能擁有其專利或專利申請案。提供本文件不代表提供這些專利的授權。您可以書面提出授權查詢，來函請寄到：

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

如果是有關雙位元組 (DBCS) 資訊的授權查詢，請洽詢所在國的 IBM 智慧財產部門，或書面提出授權查詢，來函請寄到：

智慧財產權授權
法務部與智慧財產權法律
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

下列段落不適用於英國，若與任何其他國家之法律條款抵觸，亦不適用於該國： International Business Machines Corporation 只依 "現況" 提供本出版品，不提供任何明示或默示之保證，其中包括且不限於不侵權、可商用性或特定目的之適用性的隱含保證。有些地區在特定交易上，不允許排除明示或暗示的保證，因此，這項聲明不一定適合您。

這項資訊中可能有技術上或排版印刷上的訛誤。因此，IBM 會定期修訂；並將修訂後的內容納入新版中。IBM 隨時會改進及/或變更本出版品所提及的產品及/或程式，不另行通知。

本資訊中任何對非 IBM 網站的敘述僅供參考，IBM 對該網站並不提供任何保證。這些網站所提供的資料不是 IBM 本產品的資料內容，如果要使用這些網站的資料，您必須自行承擔風險。

IBM 得以各種適當的方式使用或散布由您提供的任何資訊，無需對您負責。

如果本程式的獲授權人為了 (i) 在個別建立的程式和其他程式（包括本程式）之間交換資訊，以及 (ii) 相互使用所交換的資訊，因而需要相關的資訊，請洽詢：

IBM Corporation
軟體交互作業能力協調程式，部門 49XA
3605 公路 52 N
Rochester, MN 55901
U.S.A.

在適當條款與條件之下，包括某些情況下（支付費用），或可使用此類資訊。

IBM 基於雙方之 IBM 客戶合約、IBM 國際程式授權合約或任何同等合約之條款，提供本資訊所提及的授權程式與其所有適用的授權資料。

本文件中所含的任何效能資料都是在受管制的環境下判定。因此不同作業環境之下所得的結果，可能會有很大的差異。有些測定已在開發階段系統上做過，不過這並不保證在一般系統上會出現相同結果。甚至有部分的測量，是利用插補法而得的估計值，實際結果可能有所不同。本文件的使用者應驗證其特定環境適用的資料。

本文件所提及之非 IBM 產品資訊，取自產品的供應商，或其發佈的聲明或其他公開管道。IBM 並未測試過這些產品，也無法確認這些非 IBM 產品的執行效能、相容性或任何對產品的其他主張是否完全無誤。有關非 IBM 產品的性能問題應直接洽詢該產品供應商。

有關 IBM 未來方針或目的之所有聲明，僅代表 IBM 的目標與主旨，隨時可能變更或撤銷，不必另行通知。

這份資訊含有日常商業運作所用的資料和報告範例。為了要使它們儘可能完整，範例包括個人、公司、品牌和產品的名稱。這些名稱全屬虛構，如與實際公司的名稱和住址雷同，純屬巧合。

著作權授權：

本資訊含有原始語言之範例應用程式，用以說明各作業平台中之程式設計技術。您可以基於研發、使用、銷售或散布符合作業平台（撰寫範例程式的作業平台）之應用程式介面的應用程式等目的，以任何形式複製、修改及散布這些範例程式，而不必向 IBM 付費。這些範例並未在所有情況下完整測試。因此，IBM 不保證或暗示這些程式的可靠性、有用性或功能。

若貴客戶正在閱讀本項資訊的電子檔，可能不會有照片和彩色說明。

程式設計介面資訊

程式設計介面資訊 (如果有提供的話) 旨在協助您建立與此程式搭配使用的應用軟體。

本書包含預期程式設計介面的相關資訊，可讓客戶撰寫程式以取得 IBM WebSphere MQ 的服務。

不過，本資訊也可能包含診斷、修正和調整資訊。提供診斷、修正和調整資訊，是要協助您進行應用軟體的除錯。

重要：請勿使用此診斷、修改及調整資訊作為程式設計介面，因為它可能會變更。

商標

IBM、IBM 標誌 [ibm.com](http://www.ibm.com) 是 IBM Corporation 在全球許多適用範圍的商標。IBM 商標的最新清單可在 Web 的 "Copyright and trademark information" www.ibm.com/legal/copytrade.shtml 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: