

9.1

配置 *IBM MQ*

IBM

附註

使用本資訊及其支援的產品之前，請先閱讀第 851 頁的『[注意事項](#)』中的資訊。

除非新版中另有指示，否則此版本適用於 IBM® MQ 9.1.1 版及所有後續版本與修訂版。

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

© Copyright International Business Machines Corporation 2007, 2024.

目錄

配置	7
在 Multiplatforms 上建立佇列管理程式.....	7
可配置的暫時目錄.....	10
Userdata 目錄.....	10
建立預設佇列管理程式.....	11
將現有的佇列管理程式設為預設值.....	12
在建立佇列管理程式之後備份配置檔.....	13
配置用戶端與伺服器之間的連線.....	13
要使用的通訊類型.....	14
配置延伸交易式用戶端.....	16
定義 MQI 通道.....	25
建立及使用 AMQP 通道.....	25
在不同平台上建立伺服器連線和用戶端連線定義.....	30
在伺服器上建立伺服器連線及用戶端連線定義.....	35
MQI 通道的通道結束程式.....	50
將用戶端連接至佇列共用群組.....	53
使用 IBM MQ 環境變數.....	54
環境變數說明.....	55
在 Multiplatforms 上的 .ini 檔中變更 IBM MQ 配置資訊.....	72
IBM MQ 配置檔 mqs.ini.....	74
佇列管理程式配置檔 qm.ini.....	84
安裝配置檔 mqinst.ini.....	125
IBM MQ MQI client 配置檔 mqclient.ini.....	125
活動追蹤配置檔 mqat.ini.....	145
配置分散式佇列.....	147
IBM MQ 分散式佇列技術.....	148
分散式佇列管理簡介.....	165
在 UNIX, Linux, and Windows 上監視及控制通道.....	192
在 IBM i 上監視及控制通道.....	213
配置佇列管理程式叢集.....	232
配置發佈/訂閱傳訊.....	335
設定排入佇列的發佈/訂閱訊息屬性.....	336
正在啟動排入佇列的發佈/訂閱.....	337
正在停止排入佇列的發佈/訂閱.....	337
新增串流.....	338
刪除串流.....	339
新增訂閱點.....	339
配置分散式發佈/訂閱網路.....	340
配置多個安裝.....	356
在多重安裝環境中連接應用程式.....	356
變更主要安裝.....	363
將佇列管理程式與安裝相關聯.....	364
尋找系統上 IBM MQ 的安裝.....	366
配置高可用性、回復及重新啟動.....	367
自動用戶端重新連線.....	368
主控台訊息監視.....	373
高可用性配置.....	376
記載: 確定訊息不會遺失.....	502
備份及還原 IBM MQ 佇列管理程式資料.....	525
叢集錯誤回復的變更 (在 z/OS 以外的伺服器上).....	531
配置 JMS 資源.....	532
在 JNDI 名稱空間中配置 Connection Factory 和目的地.....	533

使用 IBM MQ Explorer 來配置 JMS 物件.....	537
使用管理工具來配置 JMS 物件.....	538
在 WebSphere Application Server 中配置 JMS 資源.....	545
將應用程式伺服器配置成使用最新的資源配接器維護層次.....	553
配置 JMS PROVIDERVERSION 內容.....	555
移除 WebSphere Application Server 可延續訂閱.....	562
正在配置 Managed File Transfer.....	564
多平台上的 MFT 配置選項.....	565
z/OS 上的 MFT 配置選項.....	566
配置 Redistributable Managed File Transfer Agent.....	566
建立 MFT 代理程式或日誌程式指令資料集.....	570
正在配置 Managed File Transfer for z/OS.....	571
配置 MFT on IBM i.....	597
初次使用時配置 MFT.....	598
配置 MFT 日誌程式.....	606
配置 Connect:Direct 橋接器.....	630
使用 MSCS 配置 MFT 代理程式.....	634
Managed File Transfer 中的高可用性代理程式.....	635
配置 IBM MQ Console 和 REST API.....	640
mqweb 伺服器的基本配置.....	640
配置安全.....	643
配置 HTTP 主機名稱.....	643
配置 HTTP 及 HTTP 埠.....	644
配置回應逾時.....	645
配置自動啟動.....	646
配置記載.....	647
配置 LTPA 記號.....	649
配置 administrative REST API 閘道.....	650
配置 messaging REST API.....	651
配置 REST API for MFT.....	653
調整 mqweb 伺服器 JVM.....	655
IBM MQ Console 和 REST API 安裝元件的檔案結構.....	656
在 z/OS 上配置 mqweb 伺服器使用情形記錄.....	658
在 Linux 上定義 Aspera gateway 連線.....	660
配置 IBM MQ 以與 IBM Cloud Private 計量服務搭配使用.....	663
在 IBM Cloud Private 上配置佇列管理程式以與計量服務實例搭配使用.....	664
透過 HTTP Proxy 連接至 IBM Cloud Private 計量服務.....	666
對計量服務連線進行疑難排解.....	667
配置 IBM MQ 以與 Salesforce 推送主題及平台事件搭配使用.....	667
配置 IBM MQ Bridge to Salesforce.....	669
IBM MQ Bridge to Salesforce 的其他配置選項.....	672
建立 Salesforce 平台事件的事件訊息.....	675
執行 IBM MQ Bridge to Salesforce.....	680
配置 IBM MQ 以與區塊鏈搭配使用.....	681
建立 IBM MQ Bridge to blockchain 的配置檔.....	683
Hyperledger Fabric 網路認證檔範例.....	685
執行 IBM MQ Bridge to blockchain.....	687
IBM MQ Bridge to blockchain 的其他配置選項.....	693
在 z/OS 上配置佇列管理程式.....	693
準備在 z/OS 上自訂佇列管理程式.....	694
設定 IBM MQ for z/OS.....	697
在 z/OS 上測試佇列管理程式.....	751
在 z/OS 上設定與其他佇列管理程式的通訊.....	759
將 IBM MQ 與 IMS 搭配使用.....	784
將 IBM MQ 與 CICS 搭配使用.....	791
升級及套用服務至 Language Environment 或 z/OS Callable Services.....	791
在 IMS 中使用 OTMA 結束程式.....	793
使用 IBM z/OSMF 來自動化 IBM MQ.....	797

正在配置 IBM MQ Advanced for z/OS VUE.....	806
啟用 MFT 代理程式與遠端 z/OS 佇列管理程式的連線功能.....	807
配置 IBM MQ Advanced for z/OS VUE 以與區塊鏈搭配使用.....	807
正在配置 IBM MQ Internet Pass-Thru.....	816
MQIPT 中的 HTTP 支援.....	816
MQIPT 中的 SOCKS 支援.....	817
MQIPT 中的 SSL/TLS 支援.....	818
Java security manager 輸入 MQIPT.....	841
MQIPT 中的安全結束程式.....	843
MQIPT 中的埠號控制.....	847
加密 MQIPT 中儲存的密碼.....	847
MQIPT 的其他安全考量.....	849
MQIPT 中的連線日誌.....	849
使用儲存器來配置 IBM MQ Internet Pass-Thru.....	850
注意事項.....	851
程式設計介面資訊.....	852
商標.....	852

正在配置 IBM MQ

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。

關於這項作業

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

您可以使用許多不同的方法，在「IBM 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 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 7 頁的『在 Multiplatforms 上建立佇列管理程式』](#)。

如需如何建立將 IBM 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 147 頁的『配置分散式佇列』](#)。

如需如何使用不同方法將用戶端連接至 IBM MQ 伺服器的指示，請參閱 [第 13 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 232 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 MQ 或佇列管理程式的行為。如需相關資訊，請參閱 [第 72 頁的『在 Multiplatforms 上的 .ini 檔中變更 IBM MQ 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。

 如需如何配置 IBM MQ for z/OS 的指示，請參閱 [第 693 頁的『在 z/OS 上配置佇列管理程式』](#)。

相關概念


[IBM MQ 技術概觀](#)

相關工作

[管理本端 IBM MQ 物件](#)

[管理遠端 IBM MQ 物件](#)

 [管理 IBM i](#)

 [管理 IBM MQ for z/OS](#)

規劃

 [在 z/OS 上規劃 IBM MQ 環境](#)

[第 693 頁的『在 z/OS 上配置佇列管理程式』](#)

使用這些指示，在「IBM MQ for z/OS」上配置佇列管理程式。

在 Multiplatforms 上建立佇列管理程式

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。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 MQ 物件。

開始之前

重要: IBM 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 MQ，則無法建立任何佇列管理程式。

在您可以建立佇列管理程式之前，您必須考量數個要點，特別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

若要建立佇列管理程式，請使用 IBM MQ 控制指令 `crtmqm`。`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®, and Windows](#) 上管理 IBM 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，請確定網路中沒有任何其他佇列管理程式具有相同的名稱。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。此外，如果您使用網路來進行發佈/訂閱傳訊，則訂閱會與建立它們的佇列管理程式名稱相關聯。因此，如果叢集或階層中的佇列管理程式具有相同的名稱，則可能會導致發佈無法到達它們。

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 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。這在範例中沒有顯示，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 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指應用程式所連接的佇列管理程式。當您呼叫 `runmqsc` 指令而未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。

將佇列管理程式指定為預設佇列管理程式時，會取代針對節點所指定的任何現有預設佇列管理程式。

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 11 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 [顯示及變更佇列管理程式屬性](#)。

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 367 頁的『配置高可用性、回復及重新啟動』](#)。

UNIX

僅適用於 IBM MQ for UNIX 系統

在使用 `crtmqm` 指令之前，您可以建立佇列管理程式目錄 `/var/mqm/qmgrs/qmgr`，即使在個別本端檔案系統上也一樣。當您使用 `crtmqm` 時，如果 `/var/mqm/qmgrs/qmgr` 目錄存在、空白且由 `mqm` 擁有，則會將它用於佇列管理程式資料。如果該目錄不是由 `mqm` 所擁有，則建立會失敗並產生 First Failure Support Technology (FFST) 訊息。如果目錄不是空的，則會建立新目錄。

關於這項作業

若要建立佇列管理程式，請使用 IBM MQ 控制指令 `crtmqm`。如需相關資訊，請參閱 `crtmqm`。`crtmqm` 指令會自動建立必要的預設物件及系統物件 (請參閱 [系統預設物件](#))。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。

Windows

在 Windows 系統上，您可以選擇使用 `crtmqm` 指令的 `sax` 選項來啟動佇列管理程式的多個實例。

當您已建立佇列管理程式及其物件時，可以使用 `strmqm` 指令來啟動佇列管理程式。

程序

- 如需協助您建立及管理佇列管理程式的相關資訊，請參閱下列子主題：
 - [第 11 頁的『建立預設佇列管理程式』](#)
 - [第 12 頁的『將現有的佇列管理程式設為預設值』](#)
 - [第 13 頁的『在建立佇列管理程式之後備份配置檔』](#)

相關概念

[使用佇列管理程式](#)

相關工作

[建立稱為 QM1 的佇列管理程式](#)

[第 72 頁的『在 Multiplatforms 上的 .ini 檔中變更 IBM MQ 配置資訊』](#)

您可以編輯配置檔 (.ini) 中的資訊，來變更 IBM MQ 或個別佇列管理程式的行為，以符合安裝的需求。您也可以變更 IBM MQ MQI clients 的配置選項。

[第 693 頁的『在 z/OS 上配置佇列管理程式』](#)

使用這些指示，在「IBM MQ for z/OS」上配置佇列管理程式。

相關參考

系統及預設物件

[crtmqm](#)

V 9.1.3 Linux UNIX 可配置的暫時目錄

從 IBM MQ 9.1.3 開始，包括可配置暫時目錄的概念，定義資料暫時至佇列管理程式的位置。這可用來容許將 UNIX and Linux 網域 Socket 放置在 Red Hat® OpenShift® 環境中未裝載的檔案系統上。

在 IBM MQ 9.1.3 之前，在 UNIX and Linux 平台上，當佇列管理程式執行時，會在 `/var/mqm/sockets` 目錄下建立 UNIX and Linux 網域 Socket。

在儲存器內執行佇列管理程式時(以 `/var/mqm` 作為已裝載檔案系統)，部分 Linux 平台可以阻止建立這些網域 Socket，因為它們容許儲存器外部的部分處理程序干擾儲存器內的作業。

此問題會阻止 IBM MQ 在預設安全環境定義下的 Red Hat OpenShift 儲存器平台中執行。

從 IBM MQ 9.1.3 開始，**EphemeralPrefix** 屬性可用來配置暫時目錄的位置。如果您不使用此屬性，則會看到行為沒有變更。

在 `mqs.ini` 中建立佇列管理程式項目(使用 **crtmqm** 或 **addmqinf** 指令)時，如果您執行下列動作，則會新增 **EphemeralPrefix** 屬性：

- 在 [AllQueue 管理程式](#) 段落中設定 **DefaultEphemeralPrefix** 屬性。
- 設定 `MQ_EPHEMERAL_PREFIX` 環境變數。
- 僅針對 **addmqinf** 指令指定 **-v EphemeralPrefix**。

您也可以現在有佇列管理程式停止時，明確地將 **EphemeralPrefix** 屬性新增至現有佇列管理程式，並在佇列管理程式重新啟動時，明確地新增此屬性。

如果您指定 **EphemeralPrefix**，當佇列管理程式啟動時，會導致在該字首(而非其一般位置)下建立佇列管理程式的資料暫時存在。亦即：

- 通常存在於 `/var/mqm/sockets/<QM>` 下的 Socket 檔案現在將位於 `/<EphemeralPrefix>/sockets/<QM>` 下
- 通常存在於 `/<Prefix>/qmgrs/<QM>/@<Subpool>` 下的子儲存區檔案現在將位於 `/<EphemeralPrefix>/qmgrs/<QM>/@<Subpool>` 下

附註：

- `/var/mqm/sockets/@SYSTEM` 會保留在其固定位置，且不是 **EphemeralPrefix** 的一部分。
- `AMQCLCHL.TAB` 會保留在 `/<Prefix>/qmgrs/<QM>/@ipcc` 之下，且不是 **EphemeralPrefix** 的一部分。

在 UNIX and Linux 平台上，**EphemeralPrefix** 限制為 12 個字元。

如果您指定的 **EphemeralPrefix** 太長或不存在，則會收到訊息 AMQ7001E: 指定給佇列管理程式的位置無效。

V 9.1.5 Multi Userdata 目錄

IBM MQ 9.1.5 以上版本有一個 `userdata` 目錄，可用來儲存持續應用程式狀態。

每一個 IBM MQ 佇列管理程式都有其持續狀態的專用檔案系統，其中包括其佇列資料及回復日誌。檔案系統包括 `userdata` 目錄，您可以用來儲存應用程式的持續狀態資訊。請參閱 [Unix 及 Linux 系統上的目錄內容](#) 及 [Windows 系統上的目錄內容](#)。

`userdata` 目錄在許多狀況下可能很有用，例如：

- 在 RDQM 配置中，當佇列管理程式失效接手至另一個節點時，也會移動應用程式資訊(請參閱 [第 462 頁的『儲存持續性應用程式狀態』](#))。
- 若為多重實例佇列管理程式，則其應用程式狀態會與其佇列管理程式資料一起位於共用網路檔案系統上。
- 通常，在其中配置應用程式的佇列管理程式服務。

如果您選擇將應用程式狀態儲存在 `userdata` 目錄中，則必須注意寫入此位置的資料可能會耗用配置給佇列管理程式的可用磁碟空間。您必須確保仍有足夠的磁碟空間可供佇列管理程式寫入佇列資料、日誌及其他持續性狀態資訊。

`userdata` 目錄具有 `mqm` 使用者和群組所有權，而且是全球可讀取的，因此使用者可以存取它，而不需要在 IBM MQ 管理者群組中 (即 `mqm`)。您無法修改 `userdata` 目錄的許可權，但可以使用您需要的任何所有權和許可權在其中建立內容。

Multi 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 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 7 頁的『在 Multiplatforms 上建立佇列管理程式』](#) 中說明的考量。

UNIX 當您使用 `crtmqm` 在 UNIX 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 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 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

[使用本端佇列](#)

相關工作

第 13 頁的『[在建立佇列管理程式之後備份配置檔](#)』

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

[顯示及變更佇列管理程式屬性](#)

相關參考

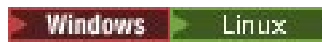
[系統及預設物件](#)

Multi 將現有的佇列管理程式設為預設值

您可以使用文字編輯器手動將現有佇列管理程式設為預設佇列管理程式，或在 Windows 及 Linux 上使用 IBM MQ Explorer，將現有佇列管理程式設為預設佇列管理程式。

關於這項作業

如果要使用文字編輯器，使現有的佇列管理程式成為預設佇列管理程式，請完成下列步驟。

 在 Windows 和 Linux (x86 和 x86-64 平台) 系統上，如果您偏好使用 IBM MQ Explorer 來進行這項變更，請參閱 [第 12 頁的『使用 IBM MQ Explorer 使佇列管理程式成為預設佇列管理程式』](#)。

當您建立預設佇列管理程式時，其名稱會插入 IBM 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，且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預設佇列管理程式。

相關工作

第 72 頁的『[在 Multiplatforms 上的 .ini 檔中變更 IBM MQ 配置資訊](#)』

您可以編輯配置檔 (.ini) 中的資訊，來變更 IBM MQ 或個別佇列管理程式的行為，以符合安裝的需求。您也可以變更 IBM MQ MQI clients 的配置選項。

Windows Linux 使用 IBM MQ Explorer 使佇列管理程式成為預設佇列管理程式

在 Windows 和 Linux (x86 和 x86-64 平台) 系統上，您可以使用 IBM MQ Explorer 來使現有的佇列管理程式成為預設佇列管理程式。

關於這項作業

若要使用 IBM MQ Explorer，使現有的佇列管理程式成為 Windows 及 Linux (x86 及 x86-64 平台) 系統上的預設佇列管理程式，請完成下列步驟。

如果您偏好使用文字編輯器來手動進行此變更，請參閱 [第 12 頁的『將現有的佇列管理程式設為預設值』](#)。

程序

1. 開啟 IBM MQ Explorer。
2. 用滑鼠右鍵按一下 **IBM MQ**，然後選取 **內容 ...**。即會顯示「**IBM MQ 的內容**」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**。

ULW 在建立佇列管理程式之後備份配置檔

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

關於這項作業

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。

配置檔有兩種類型：

- 當您安裝產品時，會建立 IBM 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。

如果您已安裝 AMQP 服務，則必須備份其他配置檔：

- **Windows** 在 Windows 系統上: amqp_win.properties
- **Linux** **UNIX** 在 UNIX 和 Linux 系統上: amqp_unix.properties

相關工作

第 72 頁的『在 Multiplatforms 上的 .ini 檔中變更 IBM MQ 配置資訊』

您可以編輯配置檔 (.ini) 中的資訊，來變更 IBM MQ 或個別佇列管理程式的行為，以符合安裝的需求。您也可以變更 IBM MQ MQI clients 的配置選項。

第 525 頁的『備份及還原 IBM MQ 佇列管理程式資料』

您可以透過備份佇列管理程式及佇列管理程式資料、僅備份佇列管理程式配置，以及使用備份佇列管理程式，來保護佇列管理程式免受硬體故障所導致的可能毀損。

配置用戶端與伺服器之間的連線

若要配置 IBM MQ MQI clients 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。

關於這項作業

在 IBM MQ 中，物件之間的邏輯通訊鏈結稱為 **通道**。用來將 IBM MQ MQI clients 連接至伺服器的通道稱為 **MQI 通道**。您可以在鏈結的每一端設定通道定義，以便 IBM MQ MQI client 上的 IBM MQ 應用程式可以與伺服器上的佇列管理程式進行通訊。

在定義 MQI 通道之前，您必須決定要使用哪種通訊形式，並在通道每一端定義連線。

如果您要在 IBM MQ MQI client 與位於不同實體網路上的佇列管理程式之間定義 MQI 通道，或透過防火牆進行通訊，則使用 IBM MQ Internet Pass-Thru 可能會簡化配置。如需相關資訊，請參閱 [IBM MQ Internet Pass-Thru](#)。

程序

1. 決定您要使用的通訊形式。
請參閱 第 14 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：

- a) 配置連線。
- b) 記錄通道定義所需的參數值。
- c) 啟動 接聽器，讓伺服器能夠偵測來自 IBM MQ MQI client 的送入網路要求。

相關概念

第 125 頁的『[IBM MQ MQI client 配置檔 mqclient.ini](#)』

您可以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

第 54 頁的『[使用 IBM MQ 環境變數](#)』

您可以使用指令來顯示現行設定或重設 IBM MQ 環境變數的值。

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 IBM MQ MQI client 與伺服器平台的組合。

MQI 通道的傳輸通訊協定類型

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：

- TCP/IP
- LU 6.2
- NetBIOS
- SPX

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 IBM MQ MQI clients，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 IBM 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	IBM MQ MQI client	IBM MQ 伺服器
TCP/IP 第 15 頁的『1』	 IBM i  UNIX  Windows	 IBM i  UNIX  Windows  z/OS
LU 6.2	 UNIX 第 15 頁的『2』  Windows	 IBM i  UNIX 第 15 頁的『2』  Windows  z/OS

表 1: 傳輸通訊協定- IBM MQ MQI client 與伺服器平台的組合 (繼續)

傳輸通訊協定	IBM MQ MQI client	IBM MQ 伺服器
NetBIOS	 Windows	 Windows
SPX	 Windows	 Windows

附註:

1.    使用 TCP/IP 的訊息通道可以指向 IBM Aspera fasp.io Gateway，它提供快速 TCP/IP 通道，可大幅增加網路傳輸量。請參閱 [在 Linux 上定義 Aspera gateway 連線](#)。
2. 除了 Linux (POWER 平台)

相關概念

第 202 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 208 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 226 頁的『[在 IBM i 上定義 TCP 連線](#)』

您可以使用連線名稱欄位，在通道定義內定義 TCP 連線。

第 777 頁的『[在 z/OS 上定義 TCP 連線](#)』

若要定義 TCP 連線，需要配置一些設定。

第 203 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 211 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 228 頁的『[在 IBM i 上定義 LU 6.2 連線](#)』

使用完整 LU 6.2 連線的模式名稱、TP 名稱及連線名稱，來定義 LU 6.2 通訊詳細資料。

第 205 頁的『[在 Windows 上定義 NetBIOS 連線](#)』

NetBIOS 連線僅適用於執行 Windows 的用戶端及伺服器。當建立與另一個 IBM MQ 產品的 NetBIOS 連線時，IBM MQ 會使用三種類型的 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。

相關工作

第 660 頁的『[在 Linux 上定義 Aspera gateway 連線](#)』

IBM Aspera fasp.io Gateway 提供快速 TCP/IP 通道，可大幅增加 IBM MQ 的網路傳輸量。在任何授與的 CD 平台上執行的佇列管理程式可以透過 Aspera gateway 進行連接。閘道本身部署在 Red Hat 上 或 Ubuntu Linux。

相關參考

第 15 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。

第 779 頁的『[使用 APPC/MVS 為 z/OS 定義 LU6.2 連線](#)』

若要定義 LU6.2 連線，則需要配置一些設定。

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 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:

表 2: 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
	20
 Linux	100
 IBM i	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 (在 UNIX and Linux 系統上為 /var/mqm/errors/AMQERR0n.LOG，在 Windows 上 IBM MQ 用戶端安裝架構的錯誤子目錄中為 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。

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 IBM 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 IBM WebSphere MQ 支援 WebSphere Application Server 4 或 5，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 6 以及更新版本包含 IBM WebSphere MQ 或 IBM MQ 傳訊提供者，因此您不需要使用延伸交易式用戶端。

配置符合 XA 標準的交易管理程式

首先配置 IBM 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，您必須先依照下列說明來配置 IBM MQ 基本用戶端：

-  在 AIX 上安裝 IBM MQ 用戶端
-  在 Linux 上安裝 IBM MQ 用戶端
-  在 Solaris 上安裝 IBM MQ 用戶端
-  在 Windows 上安裝 IBM MQ 用戶端

- **IBM i** 在 IBM i 上安裝 IBM MQ 用戶端

然後，您可以使用平台的資訊，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；使用 IBM MQ 基本用戶端的用戶端應用程式可以同時連接至單一執行緒內的多個佇列管理程式。
- 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 18 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 21 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 23 頁的『配置 CICS 的延伸交易式用戶端』](#)
- [第 24 頁的『配置 Tuxedo 的延伸交易式用戶端』](#)

相關概念

[第 20 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 21 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 22 頁的『[搭配使用延伸交易式用戶端與 TLS 通道](#)』

您無法使用 `xa_open` 字串來設定 TLS 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 21 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 21 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。

延伸交易式用戶端連線的 IBM MQ for z/OS 考量

部分 XA 交易管理程式使用的交易協調呼叫序列與連接至 IBM MQ for z/OS 的用戶端通常可用的特性不相容。

當偵測到不相容的序列時，IBM MQ for z/OS 可能會發出連線異常終止，並傳回錯誤回應給用戶端。

例如，`xa_prepare` 會接收異常終止 5C6-00D4007D，並傳回回覆碼 -3 (XAER_RMERR) 給用戶端。

另一個範例是 `xa_end` 接收異常終止 5C6-00D40079。

請確定您已在 IBM MQ for z/OS 上啟用 XA 用戶端連線的變更，可讓交易管理程式在不同的連線上準備交易。

附註：

- 依預設不會啟用變更。若要使用變更，您必須在 XA 用戶端所使用 SVRCONN 通道的說明欄位中的任何位置指定關鍵字 CSQSERVICE1 (大寫)。
- 具有 CSQSERVICE1 關鍵字的通道具有下列限制：
 - 不允許 GROUP 回復單元處置。僅容許 QMGR 回復處置單元。處置方式由 `xa_open` 呼叫上給定的名稱決定。如果使用佇列共用群組名稱，則 XA 連線會要求群組回復單元。
在 `xa_info` 參數中指定佇列共用群組名稱的 `xa_open` 呼叫失敗，出現 `xaer_inval`。
 - 不允許 `MQGMO_LOCK` 和 `MQGMO_UNLOCK` 選項。具有 `MQGMO_LOCK` 或 `MQGMO_UNLOCK` 的 `MQGET` 呼叫失敗，並發生 `MQRC_ENVIRONMENT_ERROR`。

相關概念

第 16 頁的『[配置符合 XA 標準的交易管理程式](#)』

首先配置 IBM 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。

`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 CONNAME 參數。

TRPTYPE

MQI 通道的通訊協定。下列通訊協定是有效值：

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。可以使用分號分隔字元來指定佇列管理程式的多個主機及埠，例如：

```
host1(1415);host2(1416);host3(1417)
```

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佇列共用群組名稱，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。

 如需使用 GROUP 回復單元的進一步資訊，請參閱 [佇列共用群組中的回復單元處置](#)。

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。

UID

提供給佇列管理程式進行鑑別的使用者 ID。如果提供此參數，則也必須提供 **PWD** 參數。如果已鑑別提供的使用者 ID 和密碼，則會使用使用者 ID 來識別交易管理程式的連線。使用者 ID 和密碼會在 MQCONN 呼叫上移入 MQCSP 物件。

UID 和 **PWD** 參數同時適用於用戶端和伺服器連結。

pwd

提供給佇列管理程式以進行鑑別的密碼。如果提供此參數，則也必須提供 **UID** 參數。

警告: 在某些情況下，用戶端應用程式的 MQCSP 結構中的密碼將透過網路以純文字傳送。若要確保用戶端應用程式密碼受到適當保護，請參閱 [IBM MQCSP 密碼保護](#)。

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** 參數所識別的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** 參數的值是否為位於 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。

如果應用程式連接至比 IBM WebSphere MQ 7.0.1 更早期版本的佇列管理程式，則 xa_open 呼叫會成功，但交易具有 QMGR 回復處置單元。  確保需要 GROUP 回復單元處置的應用程式僅連接至 IBM WebSphere MQ 7.0.1 或更新版本的佇列管理程式。

 如果應用程式在 **QMNAME** 參數欄位中使用佇列共用群組名稱，且它所連接的佇列管理程式上已停用 GROUPPUR 內容，則 xa_open 呼叫會失敗。

 如果應用程式用戶端連接至 IBM WebSphere MQ 7.0.1 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。如需 GROUP 回復單元處置的相關資訊，請參閱 [回復單元處置](#)。

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 字串中的 **QMNAME** 參數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 呼叫中的 **QMGrName** 參數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 21 頁的表 3。

TPM 的值	平台	AXLIB 的假設值
CICS	 AIX	/usr/lpp/encina/lib/libEncServer.a(EncServer_shr.o)
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	 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。

因此，在變更任何配置資訊之前，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。

MQRMIASwitchDynamic

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 22 頁的表 4](#) 所顯示的檔案庫中。

平台	包含 XA 交換器結構的程式庫
 AIX	MQ_INSTALLATION_PATH/lib/libmqcxa
 Linux	
 Solaris	
 Windows 系統	MQ_INSTALLATION_PATH\bin\mqcxa.dll ¹

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

每一個交換器結構中的 IBM 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 22 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。

搭配使用延伸交易式用戶端與 TLS 通道

您無法使用 xa_open 字串來設定 TLS 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 字串方法連接至佇列管理程式來傳遞設定 TLS 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 TLS 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 上的作用中目錄中，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

這會將通道定義提供給具有正確鑑別所需之 TLS 屬性的適當佇列管理程式，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。

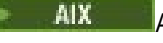


使用 CICS 線上資源定義 (RDO) 指令 **cicsadd** 來新增 XAD 資源定義。XAD 資源定義指定下列資訊：

- 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
-  Solaris
-  Windows

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 MQRMIXASwitchDynamic。如需每一個交換器載入檔案的完整路徑名稱，請參閱第 23 頁的表 5。

平台	切換載入檔案
 AIX  Linux  Solaris	MQ_INSTALLATION_PATH/lib/amqcsc
Windows	MQ_INSTALLATION_PATH\bin\mqcc4swi.dll ¹

MQ_INSTALLATION_PATH 代表 IBM 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="IBM MQ queue manager MARS" \  
XAOpen="channel=MARS.SVR,trptype=tcp,connname=MARS(1415),qmname=MARS,tpm=cics" \  
SwitchLoadFile="C:\Program Files\IBM\MQ\bin\mqcc4swi.dll"
```

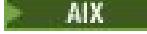



如需將 XAD 資源定義新增至 CICS 區域的相關資訊，請參閱適用於您平台的 *CICS Administration Reference* 及 *CICS Administration Guide*。

請注意搭配使用 CICS 與延伸交易式用戶端的下列相關資訊：

- 您只能將 IBM MQ 的一個 XAD 資源定義新增至 CICS 區域。這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，則必須執行不同區域中的應用程式。
- 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 IBM MQ MQI client 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 會是 cics。在某些情況下，當伺服器連線 MCA 隨後代表用戶端應用程式嘗試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 [保護安全](#)。

- 第 24 頁的表 6 中列出提供在 IBM MQ 用戶端系統上使用的 CICS 作業終止結束程式。配置這些結束程式的方式與配置 IBM 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](#)。


平台	來源	檔案庫
<ul style="list-style-type: none">  AIX  Linux  Solaris 	amqzscgx.c	amqczscg
 Windows 系統	amqzscgn.c	mqqc1415.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。

 在下列平台上，檔案名稱為 ubbstxcx.cfg:

- AIX
- Solaris

 Windows，檔案名稱為 ubbstxcn.cfg。

- 在 Tuxedo 資源管理程式表格中的佇列管理程式項目中，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：

-  在 AIX 及 Solaris 上，指定 udataobj/RM。

-  在 Windows 上，指定 udataobj\rm。

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 MQRMIXASwitch 或 MQRMIXASwitchDynamic。

Microsoft 交易伺服器

在您可以使用 Microsoft Transaction Server (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，並讓 IBM MQ MQI client 機器上的 IBM MQ 應用程式能夠存取通道。

這兩種方法詳細說明：

1. 在 IBM 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 IBM MQ MQI client 和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 30 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 IBM MQ MQI client 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 35 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 以外平台上的 IBM 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，IBM 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。

相關概念

第 173 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。

ULW 建立及使用 AMQP 通道

當您將 MQ Light API 的 IBM MQ 支援安裝至 IBM MQ 安裝時，可以執行 IBM MQ MQSC 指令 (`runmqsc`) 來定義、變更、刪除、啟動及停止通道。您也可以檢視通道的狀態。

開始之前

此作業假設您已安裝 AMQP 通道。您可以在安裝 IBM MQ 時選取 AMQP 服務元件來執行此動作。如需相關資訊，請遵循平台的鏈結，然後尋找 "AMQP Service" 的表格列：

- ▶ **AIX** [AIX 系統的 IBM MQ 元件](#)
- ▶ **Linux** [Linux 系統的 IBM MQ rpm 元件](#)
- ▶ **Linux** [適用於 Linux Ubuntu 系統的 IBM MQ Debian 元件](#)
- ▶ **Solaris** [適用於 Solaris 系統的 IBM MQ 元件](#)
- ▶ **Windows** [適用於 Windows 系統的 IBM MQ 特性](#)

若要建立與佇列管理程式的測試連線，您必須具有 MQ Light 用戶端。MQ Light 用戶端適用於 Node.js、Ruby、Java 及 Python。如需可用用戶端的相關資訊，請參閱 [IBM MQ Light 社群網站](#)。

這項作業是以 MQ Light Node.js 用戶端為基礎。不過，任何用戶端的 IBM MQ 佇列管理程式相關步驟都相同。

關於這項作業

下列程序假設您有現有的佇列管理程式。

如果您需要新的佇列管理程式，則會包含範例 Script，位於 `mqinstall/amqp/samples` 目錄中。該 Script 會建立新的佇列管理程式，啟動 AMQP 服務，建立稱為 `SAMPLE.AMQP.CHANNEL` 的新通道，並啟動該通道。

註：AMQP 通道不支援使用者定義的 AMQP 服務。AMQP 通道僅支援系統預設 `SYSTEM.AMQP.SERVICE` 服務。

Windows ▶ **Linux** 如果您執行範例 Script (`SampleMQM.sh` on Linux 或 `SampleMQM.bat` on Windows)，您可以在 [第 27 頁的『6』](#) 中啟動下列程序。

您可以使用預設通道 `SYSTEM.DEF.AMQP`，用於測試與佇列管理程式的 MQ Light 連線，或者您可以建立新通道。

下列程序使用預設通道。

程序

1. 從 `mqinstall/bin/` 目錄啟動 `runmqsc`：

```
runmqsc QMNAME
```

2. **V 9.1.0**
(只有在佇列管理程式為 IBM MQ 9.0.4 或更早版本時才需要。) 請檢查 AMQP 功能是否已安裝且正常運作。

使用 **START SERVICE** 指令來啟動 IBM MQ 服務，以控制 JVM：

```
START SERVICE(SYSTEM.AMQP.SERVICE)
```

註：從 IBM MQ 9.1 `SYSTEM.AMQP.SERVICE` 將其 **CONTROL** 屬性設為 `QMGR`。這會導致在啟動佇列管理程式時自動啟動服務。透過將 **CONTROL** 屬性設為 `MANUAL`，您可以防止在啟動佇列管理程式時啟動服務。

啟動佇列管理程式時，會自動啟動 AMQP 服務及 AMQP 通道 (如果已定義)。

3. 設定 MCAUSER 使用者 ID。

當 AMQP 用戶端連接至通道時，通道會指定用於佇列管理程式連線的 MCAUSER 使用者 ID。MCAUSER 的預設值為空白。在任何 AMQP 用戶端可以連接至佇列管理程式之前，您必須先指定 MCAUSER 值，該值必須是獲授權發佈及訂閱 IBM MQ 主題的有效 IBM MQ 使用者。

註: **Windows** 在 Windows(之前 IBM MQ 9.1.1) 上，MCAUSER 使用者 ID 設定僅支援長度最多 12 個字元的使用者 ID。

V 9.1.1 從 IBM MQ 9.1.1 開始，移除了 12 字元限制。

- a) 使用 **ALTER CHANNEL** 指令來設定 MCAUSER 使用者 ID:

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) MCAUSER(User ID)
```

- b) 使用下列兩個 **setmqaut** 指令來授權 MCAUSER 使用者 ID 發佈及訂閱主題:

```
setmqaut -m QMNAME -t topic -n SYSTEM.BASE.TOPIC -p MCAUSER  
-all +pub +sub
```

和

```
setmqaut -m QMNAME -t qmgr -p MCAUSER -all +connect
```

如果在新增或變更 MCAUSER 使用者 ID 時，通道正在執行中，您必須停止並重新啟動通道。

註: 如果未設定 MCAUSER 使用者 ID，或 MCAUSER 使用者 ID 未獲授權發佈或訂閱 IBM MQ 主題，則您會在 AMQP 用戶端中收到錯誤訊息。

4. 使用 **START CHANNEL** 指令來啟動預設 SYSTEM.DEF.AMQP 通道:

```
START CHANNEL(SYSTEM.DEF.AMQP)
```

5. 如果您要檢查通道狀態，請使用 **DISPLAY CHSTATUS** 指令:

```
DISPLAY CHSTATUS(SYSTEM.DEF.AMQP) CHLTYPE(AMQP)
```

當通道正確執行時，指令輸出中會顯示 STATUS(RUNNING)。

6. 變更預設埠。

AMQP 1.0 連線的預設埠為 5672。如果您已使用埠 5672 (如果先前已安裝 MQ Light)，則需要變更 AMQP 通道使用的埠。使用 **ALTER CHANNEL** 指令來變更埠:

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) PORT(NEW PORT NUMBER)
```

7. 如果您不想使用通道鑑別 (CHLAUTH) 規則來封鎖或過濾與 AMQP 通道的連線，請在佇列管理程式上停用通道鑑別，如下所示:

```
alter qmgr chlauth(disabled)
```

不建議您在正式作業佇列管理程式上停用連線鑑別。您只應該在開發環境中停用連線鑑別。

或者，配置佇列管理程式通道鑑別規則，以容許與 AMQP 通道的特定連線。

8. 選擇性的: 如果您想要在通道上啟用 SSL/TLS 加密，請使用佇列管理程式的已配置金鑰儲存庫，您必須將通道的 SSLCIPH 屬性設為適當的密碼規格。依預設，密碼規格為空白，表示通道上未使用 SSL/TLS 加密。使用 **ALTER CHANNEL** 指令來設定密碼規格。例如:

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) SSLCIPH(CIPHER SPECIFICATION)
```

此外，還有一些與 SSL/TLS 加密相關聯的其他通道配置選項，您可以設定如下:

- 依預設，佇列管理程式金鑰儲存庫中標籤對應於佇列管理程式 **CERTLABL** 屬性的憑證，是通道 SSL/TLS 加密所使用的名稱。您可以透過設定 **CERTLABL** 來選取不同的憑證。使用 **ALTER CHANNEL** 指令來指定所需憑證的標籤：

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) CERTLABL(CERTIFICATE LABEL)
```

- 您可以將通道設為需要 SSL/TLS 用戶端連線的憑證。您可以設定 **SSLCAUTH** 屬性，以選取是否需要 SSL/TLS 用戶端連線中的憑證。使用 **ALTER CHANNEL** 指令來設定是否需要 SSL/TLS 用戶端連線的憑證。例如：

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) SSLCAUTH(REQUIRED or OPTIONAL)
```

- **V 9.1.5** **V 9.1.0.5** 如果您將 **SSLCAUTH** 屬性設為 **REQUIRED**，則可以檢查來自用戶端之憑證的「識別名稱 (DN)」。若要從用戶端檢查憑證的「識別名稱」，請設定 **SSLPEER** 屬性。使用 **ALTER CHANNEL** 指令，從用戶端檢查憑證的「識別名稱」。例如：

```
ALTER CHANNEL(SYSTEM.DEF.AMQP) CHLTYPE(AMQP) SSLPEER (DN SPECIFICATION)
```

或者，您也可以使用通道鑑別記錄來容許或封鎖連線，因為與使用 **SSLPEER** 屬性相比，此方法提供更大的精度。如需設定 **SSLPEER** 及使用通道鑑別記錄作為替代方案的相關資訊，請參閱 [SSL 對等節點](#)。

9. 執行下列指令，以安裝 MQ Light Node.js 用戶端：

```
npm install mqlight
```

10. 導覽至 `node_modules/mqlight/samples` 目錄，並執行範例接收端應用程式：

- 如果您使用預設埠號，則可以執行範例接收端應用程式：

```
node recv.js
```

- 如果您將 **AMQP** 通道配置為使用不同的埠號，則可以使用參數來執行範例接收端應用程式，以指定新的埠號：

```
node recv.js -s amqp://localhost:6789
```

成功連線至預設通道會顯示下列訊息：

```
Connected to amqp://localhost:5672 using client-id recv_e79c55d
Subscribed to pattern: public
```

應用程式現在已連接至佇列管理程式，且正在等待接收訊息。它已訂閱主題 **public**。

註：除非您使用 **-i** 參數指定 **client-id**，否則會自動產生。

11. 在新的指令視窗中，導覽至 `node_modules/mqlight/samples` 目錄，並執行下列指令來執行範例傳送端應用程式：

```
node send.js
```

在接收端應用程式的指令視窗中，會顯示 **Hello World** 訊息。

12. 使用 **AMQSSUB** IBM MQ 範例來接收 MQ Light 範例訊息。

在 Linux 和 Windows 上，可以在下列位置找到範例：

- **Linux** Linux 上的 `mqinstall/samp/bin` 目錄。
- **Windows** Windows 上的 `mqinstall/Tools\c\Samples\Bin` 目錄。

a) 執行下列指令來執行範例:

```
amqssub public QM-name.
```

b) 重新執行下列指令，將訊息傳送至 IBM MQ 應用程式:

```
node send.js
```

13. 使用 **DEFINE CHANNEL** 指令來建立更多 AMQP 通道:

```
DEFINE CHANNEL(MY.AMQP.CHANNEL) CHLTYPE(AMQP) PORT(2345)
```

當您定義通道時，必須使用 **START CHANNEL** 指令手動啟動通道:

```
START CHANNEL(MY.AMQP.CHANNEL)
```

若要檢查通道是否正確執行，您可以執行範例接收端應用程式，並指定新通道的埠:

```
node recv.js -s amqp://localhost:2345
```

下一步

您可以使用下列指令來顯示 IBM MQ 連線、停止通道，以及刪除通道:

DISPLAY CONN(*) TYPE(CONN) WHERE (CHANNEL EQ SYSTEM.DEF.AMQP)

顯示 AMQP 通道在佇列管理程式上建立的 IBM MQ 連線。

DISPLAY CHSTATUS(*) CHLTYPE(AMQP) CLIENTID(*) ALL

顯示連接至指定通道的 AMQP 用戶端清單。

STOP CHANNEL (MY.AMQP.CHANNEL)

停止 AMQP 通道，並關閉它正在接聽的埠。

DELETE CHANNEL (MY.AMQP.CHANNEL)

刪除您建立的任何通道。

註: 請勿刪除預設通道 SYSTEM.DEF.AMQP。

您可以使用 **runmqsc** 或 PCF 來判斷 AMQP 功能是否已安裝至 IBM MQ 安裝，以及是否有與其相關聯的佇列管理程式:

- 使用 **runmqsc**，顯示佇列管理程式的屬性，並檢查 AMQPCAP (YES)。
- 使用 PCF，使用 **MQCMD_INQUIRE_Q_MGR** 指令，並確認 MQIA_AMQP_CAPABILITY 的值。

相關工作

[開發 AMQP 用戶端應用程式](#)

[保護 AMQP 用戶端安全](#)

相關參考

[strmqm](#)

ULW 從佇列管理程式中移除 AMQP 通道

您可以從安裝目錄中移除資料夾，以從佇列管理程式中移除 AMQP 通道。

程序

1. 停止佇列管理程式。
2. 移除 MQ Light API 的 IBM MQ 支援:

- **AIX** 在 AIX 上，執行下列指令：

```
installp -u mqm.amqp.rte
```

- **Linux** 在 Linux 上，移除 AMQP RPM。如果在安裝 RPM 之前重新包裝 RPM，請指定重新包裝 RPM 的名稱。

```
rpm -e MQSeriesAMQP
```

- **Windows** 在 Windows 上，從 IBM MQ 安裝中移除 amqp 資料夾。確保未移除 IBM MQ 安裝路徑中的其他檔案或資料夾。

3. 重新啟動佇列管理程式。

相關工作

[開發 AMQP 用戶端應用程式](#)

[保護 AMQP 用戶端安全](#)

ULW AMQP 通道日誌檔

AMQP 通道的日誌檔儲存在與 IBM MQ 日誌檔相同的 IBM MQ 資料目錄中。

Windows 上的預設資料目錄是 C:\ProgramData\IBM\MQ。

Linux 上的預設資料目錄是 /var/mqm。

AMQP 通道會將日誌資訊寫入 IBM MQ 資料目錄中的下列日誌檔：

- amqp.stdout, 寫入 qmgrs/*QM-name* 資料夾。
- amqp.stderr, 寫入 qmgrs/*QM-name* 資料夾。
- amqp_*.log, 寫入 qmgrs/*QM-name*/errors 資料夾。

如果 MQ Light 用戶端收到鑑別或授權錯誤，您的管理者可以在 amqp_0.log 檔和 MQ AMQERR*.log 檔中找到安全失敗原因的詳細資訊。

任何 FDC 檔案都會建立為 AMQP*.FDC 檔案，這些檔案會寫入 *data-directory/errors* 資料夾。

部分配置檔會寫入 qmgrs/*QM-name*/amqp 目錄。您不需要編輯此目錄中的任何檔案。

相關概念

[UNIX, Linux, and Windows 上的錯誤日誌](#)

相關工作

[開發 AMQP 用戶端應用程式](#)

[保護 AMQP 用戶端安全](#)

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。不過，在用戶端電腦上建立通道定義的方式有一些限制。

關於這項作業

在所有平台上，您可以使用 IBM MQ Script (MQSC) 指令、可程式化指令格式 (PCF) 指令或 IBM MQ Explorer，在伺服器機器上定義伺服器連線通道。

z/OS 在 z/OS 上，您也可以使用「作業」和「控制」畫面。

IBM i 在 IBM i 上，您也可以使用畫面介面。

因為 MQSC 指令僅在 IBM MQ 已安裝為 IBM MQ MQI client 的機器上無法使用，所以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。

當 **runmqsc** 時，適用下列考量：

- 您可以指定 **-c** 參數，並選擇性地指定 **-u** 參數，以將 **runmqsc** 作為用戶端連接至您要管理的佇列管理程式。
- 如果您使用 **-u** 參數來提供使用者 ID，則會提示您輸入相符的密碼。
- 如果您已使用 **CHCKLOCL (REQUIRED)** 或 **CHCKLOCL (REQDADM)** 來配置 **CONNAUTH AUTHINFO** 記錄，則必須使用 **-u** 參數，否則您將無法使用「**runmqsc**」來管理佇列管理程式。

程序

- 若要在伺服器上定義伺服器連線通道，請參閱第 31 頁的『[在伺服器上定義伺服器連線通道](#)』。
- 若要在 IBM MQ MQI client 上建立用戶端連線通道，請參閱第 31 頁的『[使用 MQSERVER 在 IBM MQ MQI client 上建立用戶端連線通道](#)』。

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：

```
critmqm QM1
```

- b) 啟動佇列管理程式：

```
strmqm 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 TLS 主體識別名稱。如需相關資訊，請參閱 [通道鑑別記錄](#)

使用 MQSERVER 在 IBM MQ MQI client 上建立用戶端連線通道

您可以使用 **MQSERVER** 環境變數，在用戶端工作站上定義用戶端連線通道。

關於這項作業

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。

如果您使用 **MQSERVER** 環境變數來定義 IBM MQ MQI client 機器與伺服器機器之間的通道，則這是應用程式唯一可用的通道，且不會參照用戶端通道定義表 (CCDT)。

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，回覆碼為 MQRC_Q_MGR_NAME_ERROR。

Linux

UNIX

在 UNIX and Linux 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** 環境變數所指向的任何用戶端通道定義。

程序

- 視您的平台而定，使用下列其中一個指令，以 **MQSERVER** 來指定通道定義。

- Windows 在 Windows 上，指定簡式通道定義，如下所示：

```
SET MQSERVER=ChannelName/TransportType/ConnectionName
```

例如：

```
export MQSERVER='SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)'
```

- Linux UNIX 在 UNIX and Linux 上，指定簡式通道定義，如下所示：

```
export MQSERVER=ChannelName/TransportType/ConnectionName
```

例如：

```
SET MQSERVER=SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)
```

- IBM i 在 IBM i 上，指定簡式通道定義，如下所示：

```
ADDENVVAR ENVVAR(MQSERVER) VALUE('ChannelName/TransportType/ConnectionName')
```

例如：

```
ADDENVVAR ENVVAR(MQSERVER) VALUE('SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)')
```

附註：

- ChannelName* 必須與伺服器上定義的名稱相同。它不能包含正斜線 (/) 字元，因為這個字元是用來區隔通道名稱、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。

- *TransportType* 可以是下列其中一項: LU62、TCP、NETBIOS、SPX, 視您的 IBM MQ 用戶端平台而定。
- **Linux** **UNIX** 在 UNIX and Linux 上, *TransportType* 區分大小寫且必須大寫。如果無法辨識傳輸類型, 則 MQCONN 或 MQCONNX 呼叫會傳回 2058
- *ConnectionName* 是定義給通訊協定的伺服器名稱 (*TransportType*)。它必須是完整網路名稱, 例如 AMACHINE.ACOMPANY.COM(1414)。
- *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。連線名稱清單可用作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, 則可以使用連線名稱清單來指定不同的佇列管理程式實例。
- 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:
 - **Linux** **UNIX** 在 UNIX and Linux 上:

```
unset MQSERVER
```

- **Windows** 在 Windows 上:

```
SET MQSERVER=
```

範例

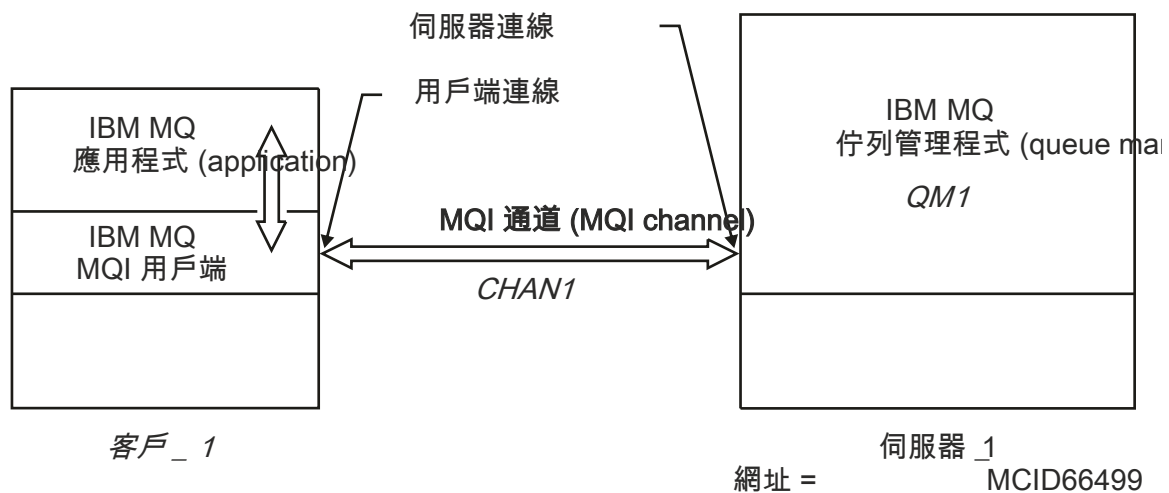


圖 1: 簡式通道定義的範例

若要建立第 33 頁的圖 1 中顯示的簡式通道定義, 請使用下列指令:

- **Linux** **UNIX** 在 UNIX and Linux 上:

```
export MQSERVER=CHANNEL1/TCP/'MCID66499'
```

- **Windows** 在 Windows 上:

```
SET MQSERVER=CHANNEL1/TCP/MCID66499
```

註: 如需如何變更 TCP/IP 埠號的相關資訊, 請參閱第 34 頁的『TCP/IP 預設埠』。

其他簡式通道定義範例如下:

- **Windows** 在 Windows 上:

```
SET MQSERVER=CHANNEL1/TCP/9.20.4.56
SET MQSERVER=CHANNEL1/NETBIOS/BOX643
```

- **Linux** **UNIX** 在 UNIX and Linux 上:

```
export MQSERVER=CHANNEL1/TCP/'9.20.4.56'
export MQSERVER=CHANNEL1/LU62/BOX99
```

其中 BOX99 是 LU 6.2 ConnectionName。

- **IBM i** 在 IBM i 上:

```
ADDENVVAR ENVVAR(MQSERVER) VALUE('CHANNEL1/TCP/9.20.4.56(1416)')
```

然後，在 IBM MQ MQI client 上，所有 **MQCONN** 或 **MQCONNX** 要求都會嘗試使用您已定義的通道，除非在提供給 **MQCONNX** 的 MQCNO 結構所參照的 MQCD 結構中置換該通道。

相關工作

第 54 頁的『[使用 IBM MQ 環境變數](#)』

您可以使用指令來顯示現行設定或重設 IBM MQ 環境變數的值。

第 35 頁的『[使用 MQCNO 在 IBM MQ MQI client 上建立用戶端連線通道](#)』

您可以在 MQCONNX 呼叫中使用 MQCNO 結構，在用戶端工作站上定義用戶端連線通道。

TCP/IP 預設埠

依預設，對於 TCP/IP，IBM 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：

- **Windows** 在 Windows 上:

```
SET MQSERVER=ChannelName/TransportType/ConnectionName(PortNumber)
```

- **Linux** **UNIX** 在 UNIX and Linux 上:

```
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 IBM MQ 新增至 services 檔，如第 209 頁的『[在 UNIX and Linux 上使用 TCP/IP 接聽器](#)』中所述。

SPX 預設 Socket

依預設，對於 SPX，IBM 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 IBM MQ 用戶端和伺服器位於相同的網路上，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 MQCNO 在 IBM MQ MQI client 上建立用戶端連線通道

您可以在 MQCONN 呼叫中使用 MQCNO 結構，在用戶端工作站上定義用戶端連線通道。

關於這項作業

IBM MQ MQI client 應用程式可以在 **MQCONN** 呼叫上使用連接選項結構 MQCNO，以參照包含用戶端連線通道定義的通道定義結構 MQCD。

透過這種方式，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，讓用戶端應用程式能夠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「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 TLS 相關的資訊。與 TLS 相關的其他資訊可以在 TLS 配置選項結構 MQSCO 中提供，該結構也由 **MQCONN** 呼叫上的 MQCNO 結構參照。

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。

相關工作





第 31 頁的『[使用 MQSERVER 在 IBM MQ MQI client 上建立用戶端連線通道](#)』
您可以使用 **MQSERVER** 環境變數，在用戶端工作站上定義用戶端連線通道。

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。

關於這項作業

您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 IBM MQ Script (MQSC) 指令、可程式化指令格式 (PCF) 指令，在伺服器機器上定義伺服器連線通道。
-   在 Linux 和 Windows 上，您也可以使用 IBM MQ Explorer。
-  在 z/OS 上，您也可以使用「作業」和「控制」畫面。
-  在 IBM i 上，您也可以使用畫面介面。

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。

程序

1. 若要定義伺服器連線通道，請參閱 [第 47 頁的『在伺服器上定義伺服器連線通道』](#)。
2. 若要定義用戶端連線通道，請參閱 [第 48 頁的『在伺服器上定義用戶端連線通道』](#)。

相關工作

第 36 頁的『[配置二進位格式 CCDT](#)』

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 Multiplatforms 上，當建立佇列管理程式時，會自動建立包含預設值的二進位 CCDT。您可以使用 **runmqsc** 指令來更新二進位 CCDT。

第 47 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 48 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 49 頁的『[存取用戶端連線通道定義](#)』

您可以複製或共用用戶端應用程式，使用用戶端通道定義表 (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和名稱。從 IBM MQ 9.0，IBM MQ 也可讓您透過 URL 來尋找用戶端通道定義表 (CCDT)。

配置用戶端通道定義表

用戶端通道定義表 (CCDT) 定義用戶端連線通道及其屬性。用戶端會讀取此檔案，以判定要連接的佇列管理程式。CCDT 檔案可以是 JSON 或二進位格式。

關於這項作業

佇列管理程式不會讀取 CCDT 檔案。它只用來提供通道定義和鑑別資訊給用戶端。

V 9.1.2 在 IBM MQ 9.1.2 之前，CCDT 僅以二進位格式提供。從 IBM MQ 9.1.2，您也可以建立 JavaScript Object Notation (JSON) 格式的 CCDT。

建立佇列管理程式時，會自動建立二進位格式 CCDT。您只使用 **runmqsc** 指令來更新儲存在此表格中的用戶端通道定義。

V 9.1.2 JSON 格式 CCDT 是副檔名為 .json 的純文字檔。您可以手動建立並更新此表格，其限制比使用 **runmqsc** 指令少。

z/OS 在應用程式伺服器內執行的 z/OS JMS 用戶端會使用 CCDT 來參照遠端佇列管理程式連線詳細資料。從 IBM MQ 9.1 開始，IBM MQ Advanced for z/OS Value Unit Edition 可讓 JMS 用戶端從遠端連接至其他 z/OS LPAR 上的佇列管理程式。因此，這些用戶端也可以使用 CCDT。

若要協助您配置 CCDT 以使用您的用戶端，請從下列作業中選擇：

程序

- 第 36 頁的『[配置二進位格式 CCDT](#)』
- **V 9.1.2**
第 38 頁的『[配置 JSON 格式 CCDT](#)』
- 第 45 頁的『[CCDT 的位置](#)』
- 第 46 頁的『[CCDT 的 URL 存取權](#)』

相關概念

統一叢集

[MQI 用戶端: 用戶端通道定義表 \(CCDT\)](#)

配置二進位格式 CCDT

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 Multiplatforms 上，當建立佇列管理程式時，會自動建立包含預設值的二進位 CCDT。您可以使用 **runmqsc** 指令來更新二進位 CCDT。

開始之前

V 9.1.2 從 IBM MQ 9.1.2 開始，您也可以建立採用 JavaScript Object Notation (JSON) 格式的 CCDT，使用此替代格式比使用二進位 CCDT 有一些優點。請參閱 [第 38 頁的『配置 JSON 格式 CCDT』](#)。

所有平台上的用戶端都可以檢視及使用 CCDT。不過，只能在 IBM MQ for Multiplatforms 下建立及修改二進位 CCDT。

關於這項作業

Multi 在多平台上：

- 在佇列管理程式資料目錄下的 @ipcc 目錄中，會自動建立二進位 CCDT。
- 除了自動建立外，與佇列管理程式相關聯的二進位 CCDT 也會與物件定義保持同步。當您定義、變更或刪除用戶端通道物件時，會在相同作業中同時更新佇列管理程式物件定義和 CCDT 中的項目。

附註：

- IBM MQ CCDT 檔案的設計是，只有在實際定義使用者所定義的所有用戶端連線通道之後，CCDT 檔案才會收縮。當刪除用戶端連線通道時，它只是在 CCDT 檔案中標示為已刪除，但實際上並未移除。
- 若要強制 CCDT 檔案縮小，在刪除一個以上用戶端連線通道之後，請發出下列指令：

```
rcrmqobj -m QM80 -t clchltab
```

- 您可以使用 **runmqsc** 指令來變更二進位 CCDT 的位置和內容。

所有平台上的用戶端都可以檢視及使用二進位 CCDT。

程序

- **Multi**
建立預設二進位 CCDT。
在 多平台上，當您建立佇列管理程式時，會建立稱為 AMQCLCHL.TAB 的預設二進位 CCDT。
依預設，AMQCLCHL.TAB 位於伺服器上的下列目錄中：

- **IBM i** 在 IBM i 上，在整合檔案系統中：

```
/QIBM/UserData/mqm/qmgrs/QUEEMANAGERNAME/&ipcc
```

- **Linux** **UNIX** 在 UNIX and Linux 系統上：

```
/prefix/qmgrs/QUEEMANAGERNAME/@ipcc
```

在 UNIX and Linux 系統上，*QUEEMANAGERNAME* 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。

- **Windows** 在 Windows 上：

```
MQ_INSTALLATION_PATH\data\qmgrs\QUEEMANAGERNAME\@ipcc
```

其中 *MQ_INSTALLATION_PATH* 代表 IBM 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。

- 找出 CCDT：
 - 在用戶端電腦上
 - 在由多個用戶端共用的位置
 - 在伺服器上作為共用檔案

請參閱第 45 頁的『CCDT 的位置』。

a) 直接在用戶端機器上建立二進位 CCDT。

- 搭配使用 `runmqsc` 指令與 `-n` 參數。
- CCDT 是建立在 `MQCHLLIB` 所指示的位置，以及 `MQCHLTAB` 所指示的檔名 (依預設為 `AMQCLCHL.TAB`) 中。
- **重要事項:** 如果您指定 `-n` 參數，則不得指定任何其他參數。

b) 變更位置。

您可以設定 `MQCHLLIB` 來變更 CCDT 的路徑。請記住，如果您在相同伺服器上有多個佇列管理程式，它們會共用相同的 CCDT 位置。

• 存取 CCDT

您可以存取 CCDT:

- 透過定義 `MQCCDTURL` 環境變數，從遠端從檔案、ftp 或 http URL。
- 在本端設定 `MQCHLLIB` 和 `MQCHLTAB` 環境變數。
- 在本端定義用戶端配置檔中 CHANNELS 段落的 `ChannelDefinitionDirectory` 及 `ChannelDefinitionFile` 屬性。

如需各種範例，請參閱第 45 頁的『CCDT 的位置』。

• 檢視或編輯 CCDT 內容。

您可以使用 `runmqsc` 指令來檢視 CCDT 內容:

1. 將環境變數設為 存取 CCDT
2. 執行指令 `runmqsc -n`
3. 執行指令 `DISPLAY CHANNEL (*)`，例如

Multi 在多平台上，您也可以使用 `runmqsc` 指令來編輯二進位 CCDT 內容。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。如需使用指令的其他範例，請參閱 `runmqsc`。

• 提供鑑別資訊給用戶端，以檢查 TLS 憑證撤銷。

- a) 定義包含鑑別資訊物件的名單。
- b) 在 CCDT 中，將佇列管理程式屬性 `SSLCRLNL` 設為名稱清單的名稱。

相關概念

使用已撤銷的憑證

相關工作

第 38 頁的『配置 JSON 格式 CCDT』

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。您可以使用文字編輯器來建立及更新 JavaScript 物件表示法 (JSON) CCDT。

V 9.1.2 配置 JSON 格式 CCDT

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。您可以使用文字編輯器來建立及更新 JavaScript 物件表示法 (JSON) CCDT。

開始之前

Multi 如果您是使用 IBM MQ for Multiplatforms，則可以改為使用建立佇列管理程式時自動建立的二進位 CCDT。請參閱第 36 頁的『配置二進位格式 CCDT』。

關於這項作業

JSON 格式的 CCDT 綱目檔名為:

Linux

/opt/mqm/lib/ccdt_schema.json

Windows



C:\Program Files\IBM\MQ\bin\ccdt_schema.json

沒有預設 JSON CCDT，且 IBM MQ 未提供任何工具來建立或編輯 JSON 格式的 CCDT。不過，當您手動開發 JSON CCDT 時，比使用 **runmqsc** 指令來使用二進位 CCDT 時，您有更多配置選項：

- 您不需要使用 IBM MQ for Multiplatforms 來建立及編輯 JSON CCDT 檔案。
- 使用 JSON 格式，您可以定義相同名稱的重複通道定義。當您在雲端上部署 IBM MQ 時，可以使用它來讓部署可擴充且高可用性。
- JSON 檔案是人類可讀的，可簡化佇列管理程式配置。
- 純文字檔格式可以與下列項目整合：
 - 用來追蹤 CCDT 歷程的版本控制工具
 - 持續交付中的自動化工具
- 您不需要專家工具即可維護 CCDT 檔案。
- 檔案較小。
- 此格式提供舊版及轉遞相容性。

附註：

1. JSON 標準會將重複索引鍵視為有效，不過，在指派屬性時，JSON 剖析器只會採用重複索引鍵的前次讀取值。因此，定義重複通道時，每一個通道都必須是指派給 'channel' 索引鍵之陣列值的元素。
2. JSON CCDT 不支援儲存「憑證撤銷清冊 (CRL)」及「線上憑證狀態通訊協定 (OCSP)」回應端位置資訊的「輕量型目錄存取通訊協定 (LDAP)」伺服器位置。

平台	JMS 用戶端編碼	C 用戶端編碼
 UNIX 平台、Linux 及 Windows	ASCII	ASCII
 z/OS	ASCII 或 EBCDIC	不適用



小心：當您透過 JSON CCDT 提供通道的任何定義 (包括不包括所有屬性的稀疏定義) 時，會使用 JSON 中未指定的任何項目的預設值，以定義的所有屬性來建構完整通道定義。

因此，您必須為您不想要預設值的每一個屬性提供特定值。

程序

- 建立 JSON CCDT
 - a) 使用一般文字編輯器來建立具有 .json 副檔名的純文字檔。
 - b) 定義 CCDT。

請參閱第 42 頁的『JSON CCDT 範例』及第 40 頁的『JSON CCDT 支援的通道屬性』。
- 找出 CCDT：
 - 在用戶端電腦上
 - 在由多個用戶端共用的位置
 - 在伺服器上作為共用檔案

請參閱第 45 頁的『CCDT 的位置』。
- 驗證 JSON CCDT
根據具有 JSON 陣容的綱目來驗證 CCDT。

如需如何建立具有兩個通道的 CCDT 檔案並驗證其運作的相關資訊，請參閱 [如何根據綱目驗證 IBM MQ CCDT JSON 檔案](#)。

CCDT 綱目隨附於產品和用戶端套件：

- **UNIX** 在 UNIX 系統上：
產品和用戶端套件中的 `$MQ_INSTALLATION_PATH/lib` 和 `/lib`。
- **Windows** 在 Windows 上：
產品和用戶端套件中的 `%MQ_INSTALLATION_PATH%\bin` 和 `\bin`。

附註：

- JSON Linters 在線上可用。
 - 綱目定義具有 'required' 索引鍵的必要屬性。
 - 綱目使用 'type' 索引鍵來定義屬性資料類型。
- 存取 CCDT
您可以存取 CCDT：
 - 透過定義 `MQCCDTURL` 環境變數，從遠端從檔案、ftp 或 http URL。
 - 在本端設定 `MQCHLLIB` 和 `MQCHLTAB` 環境變數。
 - 在本端定義用戶端配置檔中 CHANNELS 段落的 **ChannelDefinitionDirectory** 及 **ChannelDefinitionFile** 屬性。如需各種範例，請參閱 [第 45 頁的『CCDT 的位置』](#)。
 - 檢視或編輯 CCDT 內容
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。您可以使用文字編輯器來檢視或編輯 CCDT 內容。
如果您只想檢視 CCDT，也可以使用 `runmqsc` 指令來執行此動作，如下所示：
 1. 設定環境變數以提供您對 CCDT 的存取權，如前一個步驟中所述。
 2. 執行指令 `runmqsc -n`。如需相關資訊，請參閱 `runmqsc`。
 3. 執行 **DISPLAY CHANNEL** 指令。例如，執行 `DISPLAY CHANNEL(*)`。

相關概念

[統一叢集](#)

[使用已撤銷的憑證](#)

相關工作

[第 36 頁的『配置二進位格式 CCDT』](#)

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 Multiplatforms 上，當建立佇列管理程式時，會自動建立包含預設值的二進位 CCDT。您可以使用 `runmqsc` 指令來更新二進位 CCDT。

V 9.1.2 JSON CCDT 支援的通道屬性

JSON CCDT 支援的用戶端連線通道屬性清單。此清單是二進位 CCDT 所支援的屬性子集。

屬性對映

這些屬性會插入下列通道物件中：

```
{ "channel": [ { $CHANNEL_1_KEY_VALUE_LIST }, ..., { $CHANNEL_N_KEY_VALUE_LIST } ] }
```

其中 `$CHANNEL_X_KEY_VALUE_LIST` 是下表中所列出的屬性清單 (以逗點區隔)。

如需基本使用案例，請參閱 [第 42 頁的『JSON CCDT 範例』](#)。

如需可用屬性及其可能值的完整清單，請參閱 [按字母順序的通道屬性](#)。

下表列出 JSON 物件、索引鍵及資料類型，以及對應的二進位通道屬性定義。



小心: 必要屬性為通道 **name** 及通道 **type**。如果您也定義 **portRange**，則也需要 **low** 及 **high** 屬性。

JSON 物件	JSON 金鑰	JSON 資料類型	二進位屬性定義
通道 (陣列)	名	STRING	CHANNEL
通道 (陣列)	類型	STRING	CHLTYPE
channel.clientConnection	queueManager	STRING	QMNAME
channel.clientConnection.connection (陣列)	host	STRING	CONNNAME
channel.clientConnection.connection	port	INT	CONNNAME
channel.compression.header (陣列)	標頭 (header)	STRING	COMPHDR
channel.compression.message (陣列)	訊息	STRING	COMPMSG
channel.connectionManagement	affinity	STRING	AFFINITY
channel.connectionManagement	clientWeight	INT	CLNTWGHT
channel.connectionManagement	defaultReconnect	STRING	DEFRECON
channel.connectionManagement	disconnectInterval	INT	DISCINT
channel.connectionManagement	heartInterval	INT	HBINT
channel.connectionManagement	KeepAliveInterval	INT	KAINT
channel.connectionManagement	sharingConversations	INT	SHARECNV
channel.connectionManagement.localAddress (陣列)	host	STRING	LOCLADDR
channel.connectionManagement.localAddress (陣列)	port	INT	LOCLADDR
channel.connectionManagement.localAddress.portRange	high	INT	LOCLADDR
channel.connectionManagement.localAddress.portRange	低	INT	LOCLADDR
channel.exits.receive (陣列)	名	STRING	RCVEXIT
channel.exits.receive (陣列)	userData	STRING	RCVDATA
channel.exits.security	名	STRING	SCYEXIT
channel.exits.security	userData	STRING	SCYDATA
channel.exits.send (陣列)	名	STRING	SENDEXIT
channel.exits.send (陣列)	userData	STRING	SENDDATA
channel.general	說明	STRING	DESCR
channel.general	maximumMessageLength	INT	MAXMSGL
channel.timestamps	變更時間	STRING	ALTDATE 及 ALTTIME
channel.transmissionSecurity	certificateLabel	STRING	CERTLABL

JSON 物件	JSON 金鑰	JSON 資料 類型	二進位屬性定義
channel.transmissionSecurity	certificatePeer 名稱	STRING	SSLPEER
channel.transmissionSecurity	cipherSpecification	STRING	SSLCIPH

附註:

- channel.connectionManagement.localAddress 可以定義為下列其中一個按鍵組合:
 - 主機及埠:
 - host 和 portRange
 - port
 - portRange
- channel.timestamps.altered JSON 索引鍵是選用的, 如果未定義, 則值預設為 JSON CCDT 檔案前次修改時間。不過, 如果環境配置為從 URL 提取 CCDT, 則預設值是前次下載檔案的時間。
- channel.clientConnection.connection 必須同時包含主機和埠金鑰。
- 已變更的金鑰是封裝 ALTDATA 及 ALTTIME 屬性的單一字串。
- 傳輸類型只能是 TCP, 因此綱目中未定義下列屬性:
 - TRPTYPE
 - USERID
 - PASSWORD
 - MODENAME
 - TPNAME

相關參考

[通道類型的通道屬性](#)

V 9.1.2 JSON CCDT 範例

請使用本主題中列出的範例作為您需求的基礎。

開啟通用文字編輯器, 並複製下列其中一個範例:

- [第 42 頁的『定義簡式用戶端連線』](#)
- [第 43 頁的『使用 TLS 定義一個通道及一個佇列管理程式』](#)
- [第 43 頁的『定義一個通道及一個不使用 TLS 的佇列管理程式』](#)
- [第 43 頁的『定義兩個同名通道』](#)
- [第 44 頁的『CCDT 通道屬性定義的完整清單』](#)

定義簡式用戶端連線

```
{
  "channel":
  [
    {
      "general":
      {
        "description": "a channel"
      },
      "name": "channel",
      "clientConnection":
      {
        "connection":
        [
          {
            "host": "localhost",
            "port": 1414
          }
        ]
      }
    }
  ]
}
```

```

    },
    "queueManager": "QM1"
  },
  "type": "clientConnection"
}
]
}

```

使用 TLS 定義一個通道及一個佇列管理程式

```

{
  "channel": [
    {
      "name": "SSL.SVRCONN",
      "clientConnection": {
        "connection": [
          {
            "host": "aztlan1.fyre.ibm.com",
            "port": 1419
          }
        ],
        "queueManager": "QM92TLS"
      },
      "transmissionSecurity": {
        "cipherSpecification": "TLS_AES_128_GCM_SHA256",
        "certificateLabel": "ibmwebspheremqadministrator"
      },
      "type": "clientConnection"
    }
  ]
}

```

定義一個通道及一個不使用 TLS 的佇列管理程式

```

{
  "channel": [
    {
      "name": "SYSTEM.DEF.SVRCONN",
      "clientConnection": {
        "connection": [
          {
            "host": "aztlan1.fyre.ibm.com",
            "port": 1414
          }
        ],
        "queueManager": "QM92"
      },
      "type": "clientConnection"
    }
  ]
}

```

定義兩個同名通道

每一個通道都會連接至兩個不同的佇列管理程式:

```

{
  "channel": [
    {
      "general": {
        "description": "First channel"
      },
      "name": "channel",
      "clientConnection": {
        "connection": [
          {
            "host": "localhost",

```

```

        "port": 1414
      }
    ],
    "queueManager": "QM1"
  },
  "type": "clientConnection"
},
{
  "general":
  {
    "description": "Second channel"
  },
  "name": "channel",
  "clientConnection":
  {
    "connection":
    [
      {
        "host": "localhost",
        "port": 1415
      }
    ],
    "queueManager": "QM2"
  },
  "type": "clientConnection"
}
]
}

```

CCDT 通道屬性定義的完整清單

```

{
  "channel":
  [
    {
      "compression":
      {
        "header": [ "system" ],
        "message": [ "zlibfast" ]
      },
      "connectionManagement":
      {
        "sharingConversations": 10,
        "clientWeight": 1,
        "affinity": "none",
        "defaultReconnect": "yes",
        "disconnectInterval": 6000,
        "heartbeatInterval": 600,
        "keepAliveInterval": -1,
        "localAddress":
        [
          {
            "portRange":
            {
              "low": 2020,
              "high": 3030
            }
          }
        ]
      },
      "exits":
      {
        "receive":
        [
          {
            "name": "",
            "userData": ""
          }
        ],
        "security":
        {
          "name": "",
          "userData": ""
        },
        "send":
        [
          {
            "name": "",
            "userData": ""
          }
        ]
      }
    }
  ]
}

```

```

    }
  ]
},
"general":
{
  "description": "First channel",
  "maximumMessageLength": 4194304
},
"name": "the_channel",
"clientConnection":
{
  "connection":
  [
    {
      "host": "localhost",
      "port": 1414
    }
  ],
  "queueManager": "QM1"
},
"timestamps":
{
  "altered": "2018-12-04T15:37:22.000Z"
},
"transmissionSecurity":
{
  "cipherSpecification": "",
  "certificateLabel": "",
  "certificatePeerName": ""
},
"type": "clientConnection"
}
]
}

```

相關參考

[通道類型的通道屬性](#)

[按字母順序的通道屬性](#)

CCDT 的位置

IBM MQ 支援從檔案、FTP 或 HTTP URL 擷取 CCDT。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。或者，您可以將 CCDT 複製到個別用戶端電腦，或將 CCDT 複製到多個用戶端共用的位置，來配送 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。

從 IBM MQ 9.0 開始，可以在可透過 URL 存取的集中位置管理 CCDT，而不需要個別更新每一個已部署用戶端的 CCDT。IBM MQ 9.0 已新增原生 (C/C++、COBOL 及 RPG) 及未受管理 .NET 應用程式的功能，可從 URL 取回 CCDT，不論該 URL 是本端檔案、FTP 或 HTTP 資源。

IBM MQ 用戶端的預設快取行為是只有在檔案修改時間不同於前次擷取時，才會取回 CCDT 檔案。與大部分用戶端配置選項一樣，有多種方式可以提供 URL 位置：

- 透過傳入 MQCONNX MQI 呼叫的 MQCNO 結構 [CCDTUrlPtr](#) 及 [CCDTUrlOffset](#)
- [MQCCDTURL](#) 環境變數
- mqclient.ini 的「通道」段落中的 [ChannelDefinitionDirectory](#) 屬性

同時支援已鑑別和未經鑑別的 URL。這裡是一些範例：

```
export MQCCDTURL=ftp://myuser:password@myhost.sample.com//var/mqm/qmgrs/QMGR/@ipcc/AMQCLCHL.TAB
```

```
export MQCCDTURL=http://myhost.sample.com/var/mqm/qmgrs/QMGR/@ipcc/AMQCLCHL.TAB
```

如果您想要將此支援與 FTP 或 HTTP 搭配使用，則仍然需要在伺服器上管理 CCDT 檔案，但在 IBM MQ 9.0 中新增支援，則所有用戶端應用程式都可以自動挑選通道定義的變更，而無需手動推送更新項目或需要在每一個用戶端上裝載網路檔案系統。如需相關資訊，請參閱 [第 46 頁的『CCDT 的 URL 存取權』](#)。

如何指定 CCDT 在用戶端上的位置

在用戶端系統上，您可以使用下列方式來指定 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。
- 依照先前的說明，提供 CCDT 的 URL (檔案、FTP 或 HTTP)。

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。

如果您使用 URL 來提供 CCDT 的位置，則原生用戶端應用程式尋找用戶端通道定義的優先順序，如 [第 46 頁的『CCDT 的 URL 存取權』](#) 中所述。

CCDT 的 URL 存取權

您可以在可透過 URL 來存取的集中位置管理用戶端通道定義表 (CCDT)，而不需要針對每一個已部署的用戶端個別更新 CCDT。

從 IBM MQ 9.0 開始，可以使用下列任何方式透過 URL 找到用戶端通道定義表：

- 使用 MQCNO 進行程式設計
- 使用環境變數



小心：您只能對作為用戶端連接的原生程式 (即 C、COBOL 或 C++ 應用程式) 使用環境變數選項。環境變數對 Java、JMS 或受管理 .NET 應用程式沒有影響。

IBM MQ 支援從檔案、ftp 或 http URL 擷取 CCDT。

- 使用 mqclient.ini 檔案 CHANNELS 段落。

環境變數 **MQCCDTURL** 可讓您提供檔案、ftp 或 http URL 作為單一值，以便從中取得用戶端通道定義表。

除了現有的本端檔案系統目錄 (即 /var/mqm) 之外，您也可以使用 **MQCHLLIB** 環境變數指定的目錄路徑 (或 [第 134 頁的『用戶端配置檔的 CHANNELS 段落』](#) 中 **ChannelDefinitionDirectory** 屬性指定的路徑)，透過檔案、ftp 或 http URL 來尋找 CCDT 檔案。請注意，**MQCHLLIB** 值是目錄詞幹，並與 **MQCHLTAB** 一起運作，以衍生完整 URL。

透過正在 URL 中編碼的認證，支援連線上的基本鑑別：

已鑑別連線

```
export MQCHLLIB=ftp://myuser:password@myhost.sample.com/var/mqm/qmgrs/QMGR/@ipcc
export MQCHLLIB=http://myuser:password@myhost.sample.com/var/mqm/qmgrs/QMGR/@ipcc
```

未經鑑別的連線

```
export MQCHLLIB=ftp://myhost.sample.com/var/mqm/qmgrs/QMGR/@ipcc
export MQCHLLIB=http://myhost.sample.com/var/mqm/qmgrs/QMGR/@ipcc
export MQCHLLIB=file:///var/mqm/qmgrs/QMGR/@ipcc
```

註：如果您想要使用已鑑別連線，則必須與 JMS 一樣，提供 URL 中編碼的使用者名稱及密碼。

現在，原生用戶端應用程式尋找用戶端通道定義的優先順序如下：

1. MQCNO 中由 **ClientConnOffset** 及 **ClientConnPtr** 提供的 MQCD。
2. MQCNO 中 **CCDTUrlOffset** 和 **CCDTUrlPtr** 提供的 URL。
3. **MQSERVER** 環境變數。
4. 如果 mqclient.ini 檔已定義且包含 ServerConnection 參數，則會使用它所定義的通道。如需相關資訊，請參閱 [第 125 頁的『IBM MQ MQI client 配置檔 mqclient.ini』](#) 及 [第 134 頁的『用戶端配置檔的 CHANNELS 段落』](#)。
5. **MQCCDTURL** 環境變數。
6. **MQCHLLIB** 和 **MQCHLTAB** 環境變數。

7. 第 134 頁的『用戶端配置檔的 CHANNELS 段落』中的 **ChannelDefinitionDirectory** 和 **ChannelDefinitionFile**。

重要: 使用 URL 存取 CCDT 檔案一律會開啟檔案的唯讀副本，即使使用 `file://` 通訊協定也一樣。

嘗試開啟 CCDT 檔案以進行寫入存取 (例如，從用戶端使用 `runmqsc DEFINE CHANNEL` 指令時)，會傳回一則錯誤訊息，指出無法開啟檔案以進行寫入存取。

不過，可以使用 `runmqsc` 來讀取通道及鑑別資訊定義。

相關工作

第 49 頁的『存取用戶端連線通道定義』

您可以複製或共用用戶端應用程式，使用用戶端通道定義表 (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和名稱。從 IBM MQ 9.0，IBM MQ 也可讓您透過 URL 來尋找用戶端通道定義表 (CCDT)。

第 36 頁的『配置二進位格式 CCDT』

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 Multiplatforms 上，當建立佇列管理程式時，會自動建立包含預設值的二進位 CCDT。您可以使用 `runmqsc` 指令來更新二進位 CCDT。

搭配使用 CCDT 與 [IBM MQ classes for JMS](#)

相關參考

[CCDTURL](#)

[MQCNO-連接選項](#)

[XMSC_WMQ_CCDTURL](#)

Windows Active Directory 中的用戶端連線通道

在支援 Active Directory 的 Windows 系統上，IBM 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，會將它們寫入稱為 AMQCLCHL.TAB。如果用戶端連線通道使用 TCP/IP 通訊協定，則 IBM MQ 伺服器也會在 Active Directory 中發佈它們。當 IBM 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. MQCONNX 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 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 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。

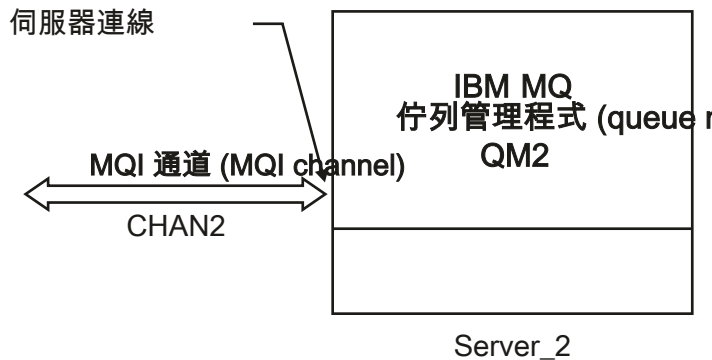


圖 2: 定義伺服器連線通道

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 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 TLS 主體識別名稱。如需相關資訊，請參閱 [通道鑑別記錄](#)

結果

Multi 在多平台上，此通道定義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，以及如何在 z/OS 上儲存用戶端連線通道定義的對應相關資訊，請參閱 [第 36 頁的『配置二進位格式 CCDT』](#)。

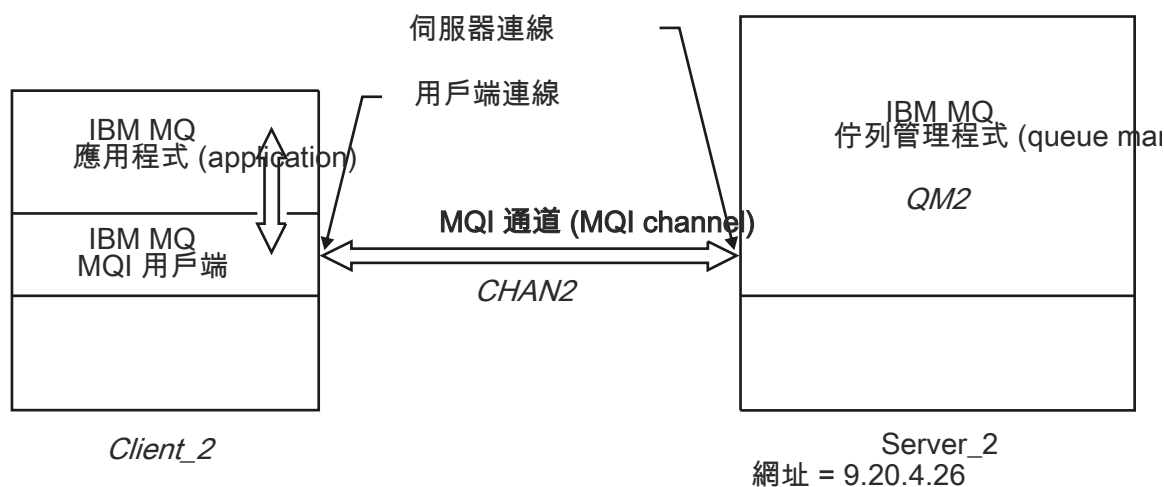


圖 3: 定義用戶端連線通道

存取用戶端連線通道定義

您可以複製或共用用戶端應用程式，使用用戶端通道定義表 (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和名稱。從 IBM MQ 9.0，IBM MQ 也可讓您透過 URL 來尋找用戶端通道定義表 (CCDT)。

開始之前

這項作業假設您已在 CCDT 中定義您需要的用戶端連線通道。請參閱 [第 36 頁的『配置用戶端通道定義表』](#)。

關於這項作業

如果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。有數種方法可執行此動作：

- 您可以將 CCDT 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。

從 IBM MQ 9.0、IBM MQ、原生 (C/C++、COBOL 及 RPG) 及未受管理的 .NET 應用程式，可以從 URL 取回集中管理的 CCDT，不論該 URL 是本端檔案、ftp 或 http 資源。

程序

1. 使用下列其中一種方式，讓 CCDT 可供用戶端應用程式使用：
 - a) 選擇性的: 將 CCDT 複製到用戶端電腦。
 - b) 選擇性的: 將 CCDT 複製到多個用戶端共用的位置。
 - c) 選擇性的: 將 CCDT 保留在伺服器上，但讓用戶端可共用它。
 - d) 選擇性的: 針對集中位置所管理的 CCDT，定義本端檔案、ftp 或 http URL，以便原生 (C/C++、COBOL 和 RPG) 和未受管理的 .NET 應用程式可以從這個 URL 取回 CCDT。

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：
 - a) 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 134 頁的『用戶端配置檔的 CHANNELS 段落』](#)。
 - b) 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 UNIX and Linux 系統上：

```
export MQCHLLIB= MQ_INSTALLATION_PATH/qmgrs/ QUEUEMANAGERNAME /@ipcc
export MQCHLTAB=AMQCLCHL.TAB
```

- **IBM i** 在 IBM i 上：

```
ADDENVVAR ENVVAR(MQCHLLIB) VALUE('/QIBM/UserData/mqm/qmgrs/QUEUEMANAGERNAME/@ipcc')
ADDENVVAR ENVVAR(MQCHLTAB) VALUE(AMQCLCHL.TAB)
```

其中 `MQ_INSTALLATION_PATH` 代表 IBM 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。
- 透過 URL，利用 MQCNO、環境變數或 `mqclient.ini` 檔案段落來進行程式設計，以提供集中管理之 CCDT 的位置。如需相關資訊，請參閱 [第 45 頁的『CCDT 的位置』](#) 及 [第 46 頁的『CCDT 的 URL 存取權』](#)。

如果設定 MQSERVER 環境變數，IBM MQ 用戶端會優先使用 MQSERVER 指定的用戶端連線通道定義，而不使用用戶端通道定義表中的任何定義。

相關概念

[第 46 頁的『CCDT 的 URL 存取權』](#)

您可以在可透過 URL 來存取的集中位置管理用戶端通道定義表 (CCDT)，而不需要針對每一個已部署的用戶端個別更新 CCDT。

[MQI 用戶端：用戶端通道定義表 \(CCDT\)](#)

相關工作

[第 36 頁的『配置二進位格式 CCDT』](#)

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 Multiplatforms 上，當建立佇列管理程式時，會自動建立包含預設值的二進位 CCDT。您可以使用 **runmqsc** 指令來更新二進位 CCDT。

ULW MQI 通道的通道結束程式

UNIX, Linux, and Windows 上的 IBM MQ MQI client 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 IBM 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 IBM 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地識別 IBM MQ 用戶端及伺服器，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 51 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 IBM 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 51 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 52 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)。

ULW 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, and Windows 系統上，在安裝 IBM MQ MQI client 期間，會將用戶端配置檔新增至系統。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

起始設定通道時，會在 MQCONN 或 MQCONNX 呼叫之後搜尋用戶端配置檔。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。

ULW 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 52 頁的圖 4 說明在用戶端連線中使用安全結束程式，並使用 IBM 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遞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。

MQCXP SecurityParms 欄位傳回的記憶體位址必須保持可定址且不變，直到 MQXR_TERM 為止。在對 MQXR_TERM 呼叫結束程式之前，結束程式不得失效或將記憶體釋放回系統。

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 4: 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 IBM WebSphere MQ 7.1 版本之前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](#)。

ULW 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 53 頁的『1』 ， 第 53 頁的『2』	X'81 '	X' 91 '

表 8: 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
MQDISC 第 53 頁的『1』	X'82'	X'92'
MQOPEN 第 53 頁的『3』	X'83'	X'93'
MQCLOSE	X'84'	X'94'
MQGET 第 53 頁的『4』	X'85'	X'95'
MQPUT 第 53 頁的『4』	X'86'	X'96'
MQPUT1 要求 第 53 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，將用戶端連接至佇列共用群組。

關於這項作業

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。如需共用佇列的相關資訊，請參閱 [共用佇列及佇列共用群組](#)。

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會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。當您不想連接至佇列共用群組內的特定佇列管理程式時，您可以使用通用介面來連接至群組。

同屬介面可以是 Sysplex Distributor VIPA 位址或 VTAM 同屬資源名稱，或是佇列共用群組的另一個共用介面。如需設定通用介面的詳細資料，請參閱 [使用佇列共用群組來設定 IBM MQ for z/OS 的通訊](#)。

程序

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。

1. 定義 SVRCONN 通道，如下列範例所示：

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(SVRCONN) TRPTYPE(TCP) +
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，以確保在您發出 MQCONN 或 MQCONNX 呼叫時一律連接至正確的伺服器連線通道。

2. 定義 CLNTCONN 通道，如下列範例所示：

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +
CONNNAME( VIPA address ) QMNAME(QSG1) +
DESCR('Client-connection to Queue Sharing Group QSG1') QSGDISP(GROUP)
```

結果

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，並將放置在該群組所擁有的共用佇列中。

使用 IBM MQ 環境變數

您可以使用指令來顯示現行設定或重設 IBM MQ 環境變數的值。

關於這項作業

您可以使用下列方式來使用環境變數：

- 設定系統設定檔中的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

對於每一個環境變數，您可以使用指令來顯示現行設定或重設變數值。除非另有說明，否則可在所有 IBM MQ MQI client 平台上使用這些指令。指令的格式視您的平台而定。例如：

-   在 UNIX and Linux 上：

```
export [environment variable]=value
```

- **Windows** 在 Windows 上:

```
Set [environment variable]=value
```

- **IBM i** 在 IBM i 上:

```
ADDENVVAR ENVVAR(environment variable) VALUE(xx)
```

適用的話，IBM MQ 會對您尚未設定的那些變數使用預設值。

註: **z/OS** IBM MQ for z/OS 不支援任何 IBM MQ 環境變數。如果您使用此平台作為伺服器，請參閱用戶端通道定義表，以取得如何在 z/OS 上產生用戶端通道定義表的相關資訊。您仍然可以在用戶端平台上使用 IBM MQ 環境變數。

程序

- **Windows**

在 Windows 上，針對每一個環境變數，使用下列指令來顯示現行設定或重設變數值:

- 若要移除環境變數的值，請使用下列指令:

```
SET MQSERVER=
```

- 若要顯示環境變數的現行設定，請使用下列指令:

```
SET MQSERVER
```

- 若要顯示階段作業的所有環境變數，請使用下列指令:

```
set
```

- **Linux** **UNIX**

在 UNIX and Linux 上，針對每一個環境變數，使用下列指令來顯示現行設定或重設變數值:

- 若要移除環境變數的值，請使用下列指令:

```
unset MQSERVER
```

- 若要顯示環境變數的現行設定，請使用下列指令:

```
echo $MQSERVER
```

- 若要顯示階段作業的所有環境變數，請使用下列指令:

```
set
```

相關工作

設定 IBM MQ classes for JMS 的環境變數
與 IBM MQ classes for Java 相關的環境變數
在 service.env 檔中定義其他環境變數

第 72 頁的『在 Multiplatforms 上的 .ini 檔中變更 IBM MQ 配置資訊』

您可以編輯配置檔 (.ini) 中的資訊，來變更 IBM MQ 或個別佇列管理程式的行為，以符合安裝的需求。您也可以變更 IBM MQ MQI clients 的配置選項。

相關參考

在 MFT 內容中使用環境變數

環境變數說明

供客戶使用之伺服器及用戶端環境變數的說明。

使用範例

-   在 UNIX and Linux 系統上，請使用下列格式: `export [environment variable]=value`。
-  在 Windows 系統上，請使用下列格式: `Set [environment variable]=value`。
-  在 IBM i 系統上，請使用下列格式: `ADDENVVAR ENVVAR(environment variable) VALUE(xx)`。
-  若為 IBM MQ Appliance，請參閱 IBM MQ Appliance 說明文件中的 [在 IBM MQ Appliance 上配置環境變數](#)。

AMQ_ALLOWED_CIPHERS

從 IBM MQ 9.1.1 開始，您可以使用 **AMQ_ALLOWED_CIPHERS** 環境變數來指定已啟用與 Multiplatforms 上 IBM MQ 通道搭配使用的 CipherSpecs 自訂清單。環境變數採用與 `.ini` 檔案的 **AllowedCipherSpecs** SSL 段落屬性相同的值：

- 單一 CipherSpec 名稱，或
- 要重新啟用的 IBM MQ CipherSpec 名稱清單 (以逗點區隔)，或
- ALL 的特殊值，代表所有 CipherSpecs (不建議)。

註：不建議啟用 **ALL** CipherSpecs，因為這將會啟用 SSL 3.0 和 TLS 1.0 通訊協定，以及大量低保護性加密演算法。

如需相關資訊，請參閱 [CipherSpec 順序中的提供 Multiplatforms 上已啟用 CipherSpecs 的自訂清單](#)。

AMQ_BAD_COMMS_DATA_FDCS

當設為任何值時，**AMQ_BAD_COMMS_DATA_FDCS** 環境變數會生效。

如果 IBM MQ 透過 TCP/IP 從主機接收的資料格式不正確 (例如，因為網路用戶端已連接至 IBM MQ 接聽器埠，並嘗試與不受支援的應用程式通訊協定進行通訊)，則佇列管理程式會將 **AMQ9207E** 錯誤訊息寫入佇列管理程式錯誤日誌中。IBM MQ 接聽器支援來自佇列管理程式訊息通道代理程式 (MCA) 以及來自 MQI、JMS 及 XMS 用戶端應用程式的 TCP/IP 連線。

註：IBM MQ 接聽器不支援 AMQP 及 MQTT 用戶端所使用的應用程式通訊協定，這些用戶端應該改為連接至在適用的 AMQP 通道或 MQXR 遙測服務中配置的網路埠。

也可能會寫入包含 IBM MQ 已收到之無效資料的失敗資料擷取 (FDC) 記錄。不過，如果這是與遠端交談的開始，且格式是簡式已知格式 (例如來自 HTTP Web 瀏覽器的 GET 要求)，則不會產生 FFST 檔案。如果您要置換此選項，以針對任何不正確的資料 (包括簡式已知格式) 寫入 FFST 檔案，則可以將

AMQ_BAD_COMMS_DATA_FDCS 環境變數設為任何值 (例如，TRUE)，然後重新啟動佇列管理程式。

AMQ_CONVEBCDICNEWLINE

從 IBM MQ 9.1.0 Fix Pack 2 和 IBM MQ 9.1.2 開始，您可以使用 **AMQ_CONVEBCDICNEWLINE** 環境變數來指定 IBM MQ 如何將 EBCDIC NL 字元轉換成 ASCII 格式。環境變數採用與 `mqs.ini` 的 **ConvEBCDICNewline** 屬性相同的值，即 `NL_TO_LF`、`TABLE` 或 `ISO` (請參閱第 78 頁的『`mqs.ini` 檔案的 AllQueue 管理程式段落』)。例如，在無法使用 `mqs.ini` 檔案的情況下，您可以使用 **AMQ_CONVEBCDICNEWLINE** 環境變數而非 **ConvEBCDICNewline** 段落屬性，在用戶端提供 **ConvEBCDICNewline** 功能。如果同時設定段落屬性及環境變數，則段落屬性優先。

如需相關資訊，請參閱 [編碼字集之間的資料轉換](#)。

AMQ_DIAGNOSTIC_MSG_SEVERITY

V 9.1.0

從 IBM MQ 9.1 開始，如果針對 IBM MQ 處理程序將 **AMQ_DIAGNOSTIC_MSG_SEVERITY** 環境變數設為 1，則當 IBM MQ 處理程序將訊息寫入錯誤日誌或主控台時，會將訊息嚴重性附加至訊息號碼作為單一大寫英文字母。

依預設會設定 **AMQ_DIAGNOSTIC_MSG_SEVERITY** 啟用的行為。您可以透過將環境變數設為 0 來關閉此行為。

如需相關資訊，請參閱 [使用錯誤日誌](#)。

AMQ_DISABLE_CLIENT_AMS

當您嘗試從舊版產品連接至佇列管理程式且使用下列其中一個用戶端時，如果報告 2085 (MQRC_UNKNOWN_OBJECT_NAME) 錯誤，則可以使用 **AMQ_DISABLE_CLIENT_AMS** 環境變數在用戶端停用 IBM MQ Advanced Message Security (AMS)：

- IBM Java runtime environment (JRE) 以外的 Java runtime environment (JRE)
- IBM MQ IBM MQ classes for JMS 或 IBM MQ classes for Java 用戶端。

註：您無法將 **AMQ_DISABLE_CLIENT_AMS** 環境變數用於 C 用戶端。您需要改用 **MQS_DISABLE_ALL_INTERCEPT** 環境變數。

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

AMQ_DMPMQCFG_QSGDISP_DEFAULT

V 9.1.5

V 9.1.0.5

從 IBM MQ 9.1.0 Fix Pack 5 和 IBM MQ 9.1.5 開始，依預設，查詢 **dmpmqcfg** 指令所使用的佇列管理程式處置方式只會查詢 QSGDISP (QMGR) 定義。您可以使用 **AMQ_DMPMQCFG_QSGDISP_DEFAULT** 環境變數 (可以設為下列其中一個值) 來查詢其他定義：

即時

僅併入以 QSGDISP (QMGR) 或 QSGDISP (COPY) 定義的物件。

ALL

併入以 QSGDISP (QMGR) 及 QSGDISP (COPY) 定義的物件。如果佇列管理程式是佇列共用群組的成員，則也會包括 QSGDISP (GROUP) 及 QSGDISP (SHARED)。

COPY

僅包括以 QSGDISP (COPY) 定義的物件

GROUP

僅包括以 QSGDISP (GROUP) 定義的物件；目標佇列管理程式必須是佇列共用群組的成員。

QMGR

僅併入以 QSGDISP (QMGR) 定義的物件。如果您使用此環境變數來符合 **dmpmqcfg** 的現有行為，則這是預設行為。

PRIVATE

僅併入以 QSGDISP (QMGR) 或 QSGDISP (COPY) 定義的物件。

SHARED

僅包括以 QSGDISP (SHARED) 定義的物件。

AMQ_LICENSING_METRIC

V 9.1.1

Multi

從 IBM MQ 9.1.1 開始，設定 **AMQ_LICENSING_METRIC=VPCMonthlyPeak** 環境變數會導致佇列管理程式上傳與每月 VPC 授權類型相關的資料，而不是上傳與每小時儲存器型授權相關的資料的預設行為。

如需配置 IBM MQ 以與 IBM Cloud Private 計量服務搭配使用的相關資訊，請參閱 IBM Cloud Private 說明文件中的 [IBM Cloud Private 計量服務](#)。

AMQ_LDAP_TRACE

V 9.1.0.4 V 9.1.4

從 IBM MQ 9.1.0 Fix Pack 4 及 IBM MQ 9.1.4 開始，如果 **AMQ_LDAP_TRACE** 環境變數設為非空值，則可以開啟及關閉 LDAP 用戶端追蹤，而無需同時停止或啟動佇列管理程式。

如需相關資訊，請參閱 [啟用 LDAP 用戶端程式庫程式碼的動態追蹤](#)。

AMQ_MQS_INI_LOCATION

Linux UNIX

在 UNIX and Linux 系統上，您可以透過在 **AMQ_MQS_INI_LOCATION** 環境變數中設定 `mqs.ini` 檔案的位置，來變更為用於 `mqs.ini` 檔案的位置。必須在系統層次設定此環境變數。

如需 `mqs.ini` 檔案 (包括目錄位置) 的相關資訊，請參閱 [第 74 頁的『IBM MQ 配置檔 mqs.ini』](#)。

AMQ_NO_BAD_COMMS_DATA_FDCS

V 9.1.5 V 9.1.0.5

當設為任何值時，**AMQ_NO_BAD_COMMS_DATA_FDCS** 環境變數會生效。

當嘗試將非 IBM MQ 用戶端連接至 IBM MQ TCP/IP 接聽器時，如果 IBM MQ 無法辨識起始資料傳輸，這會導致佇列管理程式將 `AMQ9207E` 錯誤訊息寫入佇列管理程式錯誤日誌中。也會寫入失敗資料擷取 (FDC) 記錄。您可以使用 **AMQ_NO_BAD_COMMS_DATA_FDCS** 環境變數來暫停產生這些診斷檔案。當 **AMQ_NO_BAD_COMMS_DATA_FDCS** 設為任何值 (例如 TRUE) 時，這會指示 IBM MQ 在起始通訊流程上報告 `AMQ9207E` 錯誤訊息時不要產生 FFST。若要有效，應該在啟動佇列管理程式及接聽器處理程序之前設定環境變數。

如果用戶端將有效 IBM MQ 通訊協定流程傳送至佇列管理程式，然後傳送無效資料，則會繼續產生 FDC，因為這表示用戶端問題需要進一步調查。

AMQ_NO_IPV6

當設為任何值時，**AMQ_NO_IPV6** 環境變數會生效。當設定此環境變數時，它會在嘗試連線時停用 IPv6。

AMQ_REVERSE_COMMIT_ORDER

AMQ_REVERSE_COMMIT_ORDER 環境變數會配置佇列管理程式，以便在 XA 交易中，在對應的資料庫更新完成之後，確定 IBM MQ 佇列管理程式變更。只有在完成對應的資料庫更新之後，從佇列讀取訊息的應用程式才會看到訊息。

註: 如果沒有閱讀並瞭解 [隔離層次](#) 中說明的實務範例，請勿設定 **AMQ_REVERSE_COMMIT_ORDER**。

AMQ_SSL_ALLOW_DEFAULT_CERT

從 IBM MQ 9.0.0 Fix Pack 1 和 IBM MQ 9.0.2 開始，如果未設定 **AMQ_SSL_ALLOW_DEFAULT_CERT** 環境變數，則只有在憑證包含 `ibmwebspheremquserid` 標籤名稱時，應用程式才能使用用戶端金鑰儲存庫中的個人憑證連接至佇列管理程式。當設定 **AMQ_SSL_ALLOW_DEFAULT_CERT** 環境變數時，憑證不需要標籤名稱 `ibmwebspheremquserid`。亦即，用來連接至佇列管理程式的憑證可以是預設憑證，前提是金鑰儲存庫中存在預設憑證，且金鑰儲存庫不包含字首為 `ibmwebspheremquserid` 的個人憑證。

值 1 會啟用預設憑證。

應用程式可以使用 `mqclient.ini` 檔案中 SSL 段落的 **CertificateLabel** 設定，而不使用 **AMQ_SSL_ALLOW_DEFAULT_CERT** 環境變數。如需相關資訊，請參閱 [數位憑證標籤](#)，瞭解需求和 [第 141 頁的『用戶端配置檔的 SSL 段落』](#)。

AMQ_SSL_LDAP_SERVER_VERSION

在 CRL 伺服器需要使用特定版本的 LDAP 通訊協定時，可以使用 **AMQ_SSL_LDAP_SERVER_VERSION** 環境變數來確保 IBM MQ 加密元件使用 LDAP v2 或 LDAP v3。

在用來啟動佇列管理程式或通道的環境中，將環境變數設為適當的值：

- 若要要求使用 LDAP v2，請設定 **AMQ_SSL_LDAP_SERVER_VERSION=2**。
- 若要要求使用 LDAP v3，請設定 **AMQ_SSL_LDAP_SERVER_VERSION=3**。

此環境變數不會影響 IBM MQ 佇列管理程式針對使用者鑑別或使用者授權所建立的 LDAP 連線。

GMQ_MQ_LIB

當系統上同時安裝 IBM MQ MQI client 和 IBM MQ 伺服器時，依預設會針對伺服器執行 ActiveX (MQAX) 應用程式的 IBM MQ 自動化類別。若要對用戶端執行 MQAX，必須在 **GMQ_MQ_LIB** 環境變數中指定用戶端連結程式庫，例如，設定 **GMQ_MQ_LIB=mqic.dll**。對於僅限用戶端安裝，不需要設定 **GMQ_MQ_LIB** 環境變數。未設定此環境變數時，IBM MQ 會嘗試載入 amqzst.dll。如果此 DLL 不存在 (如同僅在用戶端安裝中的情況)，IBM MQ 會嘗試載入 mqic.dll。

首頁



在 UNIX、Linux 和 IBM i 上，**HOME** 環境變數指定搜尋 mqclient.ini 檔案的目錄名稱。此檔案包含 IBM MQ MQI clients 使用的配置資訊。

如需相關資訊，請參閱第 125 頁的『IBM MQ MQI client 配置檔 mqclient.ini』和第 128 頁的『用戶端配置檔的位置』。

HOMEDRIVE 和 HOMEPATH



若要使用，必須同時設定 **HOMEDRIVE** 和 **HOMEPATH** 環境變數。它們在 Windows 系統上用來指定搜尋 mqclient.ini 檔案的目錄名稱。此檔案包含 IBM MQ MQI clients 使用的配置資訊。

如需相關資訊，請參閱第 125 頁的『IBM MQ MQI client 配置檔 mqclient.ini』和第 128 頁的『用戶端配置檔的位置』。

LDAP_BASEDN

LDAP_BASEDN 是執行 LDAP 範例程式所需的環境變數。它指定目錄搜尋的基本「識別名稱」。

LDAP_HOST

LDAP_HOST 是用於執行 LDAP 範例程式的選用環境變數。它指定 LDAP 伺服器執行所在的主機名稱；如果未指定，則會預設為本端主機。

LDAP_VERSION

LDAP_VERSION 是用於執行 LDAP 範例程式的選用環境變數。它指定要使用的 LDAP 通訊協定版本，可以是 2 或 3。大部分 LDAP 伺服器現在都支援第 3 版通訊協定；它們都支援第 2 版之前的版本。此範例同樣適用於任一版本的通訊協定，如果未指定，則預設為第 2 版。

MQ_CHANNEL_SUPPRESS_INTERVAL

MQ_CHANNEL_SUPPRESS_INTERVAL 環境變數指定時間間隔 (以秒為單位)，在此期間將抑制將以 **MQ_CHANNEL_SUPPRESS_MSGS** 定義的訊息寫入錯誤日誌，以及在抑制之前指定時間間隔期間容許訊息發生的次數。預設值為 60,5，表示在 60 秒間隔內，給定訊息的前 5 次出現之後，會抑制任何進一步出現的訊息。如需相關資訊，請參閱 [在 Multiplatforms 上抑制來自錯誤日誌的通道錯誤訊息](#)。

MQ_CHANNEL_SUPPRESS_INTERVAL 環境變數與 [第 84 頁的『佇列管理程式配置檔 qm.ini』](#) 檔案中的 `SuppressInterval` 類似。

MQ_CHANNEL_SUPPRESS_MSGS

MQ_CHANNEL_SUPPRESS_MSGS 環境變數會抑制錯誤日誌中的通道錯誤訊息。您可以指定抑制的訊息清單。**MQ_CHANNEL_SUPPRESS_MSGS** 與 **MQ_CHANNEL_SUPPRESS_INTERVAL** 一起使用，可指定每則訊息在暫停之前出現的次數，以及暫停訊息的時間長度。如需相關資訊，請參閱 [在 Multiplatforms 上抑制來自錯誤日誌的通道錯誤訊息](#)。

MQ_CHANNEL_SUPPRESS_MSGS 環境變數與 [第 84 頁的『佇列管理程式配置檔 qm.ini』](#) 檔案中的 `SuppressMessage` 相當，但您可以使用環境變數來暫停任何通道訊息，而 `qm.ini` 方法則有限制清單。

MQ_CONNECT_TYPE

Multi

在 Multiplatforms 上，您可以搭配使用 **MQ_CONNECT_TYPE** 環境變數與 MQCONNX 呼叫所用 MQCNO 結構的「選項」欄位中指定的連結類型。**MQ_CONNECT_TYPE** 只對 STANDARD 連結有任何作用。對於其他連結，會忽略 **MQ_CONNECT_TYPE**。

如需相關資訊，請參閱 [搭配使用 MQCONNX 呼叫選項與 MQ_CONNECT_TYPE](#)。

MQ_CROSS_QUEUE_ORDER_ALL

當您將 **MQ_CROSS_QUEUE_ORDER_ALL** 環境變數設為非零值時，會在工作單元中維護訊息放置順序。這表示，如果將工作單元 (UoW) 中的訊息放入多個佇列 (例如，Q1，然後 Q2)，則在發出 MQCMIT 時，會以訊息被 PUT 的相同佇列順序來遞送及提供訊息。

在多佇列管理程式環境中，在每一個佇列管理程式啟動之前，**MQ_CROSS_QUEUE_ORDER_ALL** 必須存在且在傳送端和接收端都有非空白值。

MQ_EPHEMERAL_PREFIX

V 9.1.3

ULW

MQ_EPHEMERAL_PREFIX 環境變數指定佇列管理程式的暫時目錄路徑，當佇列管理程式正在執行時，會在其中保留暫時佇列管理程式資料。

除了透過變更 `mq.s.ini` 檔 `AllQueueManager` 段落的 `DefaultEphemeralPrefix` 屬性中的 `EphemeralPrefix` 屬性來變更暫時字首之外，您也可以使用環境變數 **MQ_EPHEMERAL_PREFIX** 來置換 `crtmqm` 指令的 `EphemeralPrefix`。如需相關資訊，請參閱 [第 10 頁的『可配置的暫時目錄』](#)。

MQ 檔案路徑

Windows

在 Windows 平台上安裝執行時期套件期間，會配置 **MQ_FILE_PATH** 環境變數。此環境變數包含與 Windows 登錄中下列機碼相同的資料：

```
HKEY_LOCAL_MACHINE\SOFTWARE\IBM\WebSphere MQ\Installation\InstallationName\FilePath
```

如需相關資訊，請參閱 [setmqenv \(設定 IBM MQ 環境\)](#) 及 [crtmqenv \(建立 IBM MQ 環境\)](#)。

MQ_JAVA_DATA_PATH

MQ_JAVA_DATA_PATH 環境變數指定 IBM MQ classes for JMS 和 IBM MQ classes for Java 的日誌和追蹤輸出的目錄。它由 IBM MQ classes for JMS 和 IBM MQ classes for Java 隨附的 Script 使用。

如需相關資訊，請參閱 [設定 IBM MQ classes for JMS](#) 及 [與 IBM MQ classes for Java 相關的環境變數](#)。

MQ_JAVA_INSTALL_PATH

MQ_JAVA_INSTALL_PATH 環境變數指定 IBM MQ classes for JMS 的安裝目錄，如 [IBM MQ classes for JMS](#) 中所安裝的項目，以及 [IBM MQ classes for Java 安裝目錄](#) 中所顯示的 IBM MQ classes for Java。

如需相關資訊，請參閱 [設定 IBM MQ classes for JMS](#) 及 [與 IBM MQ classes for Java 相關的環境變數](#)。

MQ_JAVA_LIB_PATH

MQ_JAVA_LIB_PATH 環境變數指定儲存 IBM MQ classes for JMS 及 IBM MQ classes for Java 程式庫的目錄。IBM MQ classes for JMS 或 IBM MQ classes for Java 隨附的部分 Script (例如 IVTRun) 使用此環境變數。

如需相關資訊，請參閱 [設定 IBM MQ classes for JMS](#) 及 [與 IBM MQ classes for Java 相關的環境變數](#)。

MQ_SET_NODELAYACK

AIX

MQ_SET_NODELAYACK 環境變數會在 AIX 上關閉 TCP 延遲確認。

當您設定此環境變數時，此設定會透過使用 TCP_NODELAYACK 選項呼叫作業系統的 `setsockopt` 呼叫來關閉 TCP 延遲確認通知。只有 AIX 支援此函數，因此 **MQ_SET_NODELAYACK** 環境變數只會影響 AIX。

MQAPI_TRACE_LOGFILE

範例 API 結束程式會對具有 **MQAPI_TRACE_LOGFILE** 環境變數中所定義字首的使用者指定檔案產生 MQI 追蹤。

如需相關資訊，請參閱 [API 結束程式範例程式](#)。

MQAPPLNAME

ULW

V 9.1.2

如果尚未選擇應用程式名稱，您可以使用 **MQAPPLNAME** 環境變數作為名稱，以用來識別佇列管理程式的連線。只會使用前 28 個字元，且它們不能全為空白或空值。

如需相關資訊，請參閱 [在支援的程式設計語言中使用應用程式名稱](#)。

MQCCSID

MQCCSID 環境變數指定要使用的編碼字集號碼，並置換已配置伺服器的 CCSID 值。**MQCCSID** 可用來置換應用程式的原生 CCSID，並指定要使用的編碼字集號碼，例如，如果原生 CCSID 是不受支援的 CCSID 或不是必要的 CCSID。

若要設定 **MQCCSID**，請使用下列其中一個指令：

- Linux UNIX 在 UNIX and Linux 上：

```
export MQCCSID=number
```

- Windows 在 Windows 上：

```
SET MQCCSID=number
```

- IBM i 在 IBM i 上：

```
ADDENVVAR ENVVAR(MQCCSID) VALUE(number)
```

如需相關資訊，請參閱 [選擇用戶端或伺服器 CCSID](#)。

MQCCDTURL

MQCCDTURL 環境變數提供相等的功能來設定 **MQCHLLIB** 與 **MQCHLTAB** 環境變數的組合。它可讓您提供檔案、ftp 或 http URL 作為單一值，您可以從中取得用戶端通道定義表，以作為用戶端連接的原生程式，亦即 C、COBOL 或 C++ 應用程式。

註：使用環境變數來提供 URL 對 Java、JMS 或受管理 .NET 應用程式沒有影響。

IBM MQ 支援從檔案、ftp 或 http URL 擷取 CCDT。不過，**MQCCDTURL** 只接受 URL 值。它不接受現有的本端檔案系統目錄格式。

若要使用 **MQCCDTURL** 來取代 **MQCHLLIB** 和 **MQCHLTAB** 與本端檔案搭配，您可以使用 'file://' 通訊協定。因此，如下列 AIX 和 Linux 範例所示：

```
export MQCCDTURL=file:///var/mqm/qmgrs/QMGR/@ipcc/MYCHL.TAB
```

相當於：

```
export MQCHLLIB=/var/mqm/qmgrs/QMGR/@ipcc
export MQCHLTAB=MYCHL.TAB
```

您也可以指定 JSON 檔案，如下列 Windows 範例所示：

```
set MQCCDTURL=file:/c:/mq-channels/CCDT-QMGR1.json
```

相當於：

```
set MQCHLLIB=C:\mq-channels
set MQCHLTAB=CCDT-QMGR1.json
```

如需相關資訊，請參閱 [第 46 頁的『CCDT 的 URL 存取權』](#)。

MQCERTLABL

MQCERTLABL 環境變數定義通道定義的憑證標籤，供 IBM MQ 用來尋找在 TLS 信號交換期間傳送的個人憑證。

如需相關資訊，請參閱 [數位憑證標籤](#)，瞭解需求。

MQCERTVPOL

MQCERTVPOL 環境變數指定要使用的憑證驗證原則類型。此環境變數會置換用戶端配置檔 SSL 段落中的 **CertificateValPolicy** 屬性。

MQCERTVPOL 可以設為下列兩個值之一：

ANY

請使用基礎 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。

若要設定 **MQCERTVPOL**，請使用下列其中一個指令：

- Linux 或 UNIX 若為 UNIX and Linux 系統：

```
export MQCERTVPOL= value
```

- Windows 若為 Windows 系統：

```
SET MQCERTVPOL= value
```

- **IBM i** 若為 IBM i 系統:

```
ADDENVVAR ENVVAR(MQCERTVPOL) VALUE(value)
```

如需相關資訊，請參閱 [IBM MQ 中的憑證驗證原則](#) 及 [在 IBM MQ 中配置憑證驗證原則](#)。

MQCHLLIB

MQCHLLIB 環境變數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 IBM MQ MQI client 工作站。

若要設定 **MQCHLLIB**，請使用下列其中一個指令：

- **Windows** 在 Windows 上:

```
SET MQCHLLIB=pathname
```

例如：

```
SET MQCHLLIB=C:\wmqtest
```

- **Linux** **UNIX** 若為 UNIX and Linux 系統:

```
export MQCHLLIB=pathname
```

- **IBM i** 若為 IBM i:

```
ADDENVVAR ENVVAR(MQCHLLIB) VALUE(pathname)
```

如果未設定 **MQCHLLIB**，用戶端的路徑預設為：

- **Linux** **UNIX** 在 UNIX and Linux 上: `/var/mqm/`
- **Windows** 在 Windows 上: `MQ_INSTALLATION_PATH`
- **IBM i** 在 IBM i 上: `/QIBM/UserData/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。

- **Linux** **UNIX** 在 UNIX and Linux 上: `datapath/@ipcc`
- **Windows** 在 Windows 上: `datapath\@ipcc`
- **IBM i** 在 IBM i 上: `datapath/&ipcc`

或：


- **Linux** **UNIX** 在 UNIX and Linux 上: `/prefix/qmgrs/qmgrname/@ipcc`
- **Windows** 在 Windows 上: `MQ_INSTALLATION_PATH\data\qmgrs\qmgrname\@ipcc`
- **IBM i** 在 IBM i 上: `/prefix/qmgrs/qmgrname/&ipcc`

其中：

- `MQ_INSTALLATION_PATH` 代表 IBM 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` 是佇列管理程式段落中定義的「字首」值。字首通常是下列其中一個值：

-   UNIX and Linux 系統上的 /var/mqm。
-  IBM i 上的 /QIBM/UserData/mqm/。
- *mqgrname* 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。
- 定義佇列管理程式段落的位置視平台而定：
 -    在 IBM i, UNIX and Linux 上的 mq.s.ini 檔案中。
 -  在 Windows 上的登錄中。

附註：

1.  如果您使用 IBM MQ for z/OS 作為伺服器，則檔案必須保留在 IBM MQ 用戶端工作站上。
2. 如果設定，MQCHLLIB 會置換用來尋找 CCDT 的路徑。
3. MQCHLLIB 可以包含與 MQCHLTAB 環境變數組合使用的 URL (請參閱 第 46 頁的『CCDT 的 URL 存取權』)。
4. 環境變數 (例如 **MQCHLLIB**) 可以採用平台專用方式，將範圍限定為處理程序或工作，或全系統。
5. 如果您在伺服器上設定 **MQCHLLIB** 全系統，它會針對伺服器上的所有佇列管理程式，設定相同的 CCDT 檔案路徑。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)。
6. 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，則會遺失您的修改。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。
7. 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同名的用戶端連線通道定義。如果定義先前是由相同的佇列管理程式所建立，則不會取代該定義。
8. 指令 **rcrmqobj -t clchltab** 會刪除並重建 CCDT 檔案。僅使用在執行指令所針對的佇列管理程式上建立的用戶端連線定義來重建檔案。
9. 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。
10. **MQCHLLIB** 的路徑不需要引號。

如需相關資訊，請參閱 第 45 頁的『CCDT 的位置』、第 46 頁的『CCDT 的 URL 存取權』及 使用環境變數將用戶端應用程式連接至佇列管理程式。

MQCHLTAB

MQCHLTAB 環境變數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。

若要設定 **MQCHLTAB**，請使用下列其中一個指令：

-   在 UNIX and Linux 上：

```
export MQCHLTAB=filename
```

-  在 Windows 上：

```
SET MQCHLTAB=filename
```


- ▶ **IBM i** 在 IBM i 上:

```
ADDENVVAR ENVVAR(MQCHLTAB) VALUE(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。

如需相關資訊，請參閱第 45 頁的『[CCDT 的位置](#)』、第 46 頁的『[CCDT 的 URL 存取權](#)』及 [使用環境變數將用戶端應用程式連接至佇列管理程式](#)。

MQCLNTCF

MQCLNTCF 環境變數指定 IBM MQ MQI client 配置檔的位置。此檔案包含 IBM MQ MQI clients 使用的配置資訊。

您可以使用 **MQCLNTCF** 環境變數來修改 `mqclient.ini` 檔案的檔案路徑。

此環境變數的格式為完整 URL。這表示檔名不一定是 `mqclient.ini`，這有助於將檔案放置在網路連接的檔案系統上。如需相關資訊，請參閱第 125 頁的『[IBM MQ MQI client 配置檔 mqclient.ini](#)』及第 128 頁的『[用戶端配置檔的位置](#)』。

MQDOTNET_TRACE_ON

MQDOTNET_TRACE_ON 環境變數用來啟用 IBM MQ .NET 可重新配送用戶端的追蹤。等於及小於 0 的值不會啟用追蹤，1 會啟用預設追蹤，而大於 1 的值會啟用詳細資料追蹤。

如需相關資訊，請參閱 [安裝 IBM MQ classes for .NET Standard](#)。

MQIPADDRV

MQIPADDRV 環境變數指定用於通道連線的 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 和 IPv6 以及用戶端配置檔 TCP 段落的 **IPAddressVersion** 屬性相同。如果未設定環境變數，則會採用 "MQIPADDR_IPV4"。

若要設定 **MQIPADDRV**，請使用下列其中一個指令：

- ▶ **Linux** ▶ **UNIX** 在 UNIX and Linux 上:

```
export MQIPADDRV=MQIPADDR_IPV4|MQIPADDR_IPV6" />
```

- ▶ **Windows** 在 Windows 上:

```
SET MQIPADDRV=MQIPADDR_IPV4|MQIPADDR_IPV6
```

- ▶ **IBM i** 在 IBM i 上:

```
ADDENVVAR ENVVAR(MQIPADDRV) VALUE(MQIPADDR_IPV4|MQIPADDR_IPV6)
```

MQLICENSE

▶ **Linux** ▶ **V 9.1.5**

在 Linux 系統上，安裝產品之後，您可以使用 **MQLICENSE** 環境變數來接受或檢視 IBM MQ 授權。

如需您可能想要或需要執行此動作之原因的相關資訊，請參閱 [IBM MQ for Linux 上的授權接受](#)。

MQLICENSE 環境變數可以設為下列兩個值之一：

接受

接受授權後置安裝。

檢視

如果已接受授權，則顯示授權。

若要接受授權後置安裝，請使用下列指令：

```
export MQLICENSE=accept
```

若要檢視授權，請使用下列指令：

```
export MQLICENSE=view
```

註：您也可以使用下列指令來接受及顯示授權：

- [mqlicense](#) (安裝後接受授權)
- [dspmqlic](#) (顯示 IBM MQ 授權)

MQMAXERRORLOGSIZE

Multi

MQMAXERRORLOGSIZE 環境變數指定複製到備份的佇列管理程式錯誤日誌大小。

如需相關資訊，請參閱 [使用錯誤日誌](#)。

MQNAME

Windows

MQNAME 環境變數指定 IBM MQ 處理程序可以使用的本端 NetBIOS 名稱。NetBIOS 連線僅適用於執行 Windows 的用戶端及伺服器。

若要設定 **MQNAME**，請使用下列指令：

```
SET MQNAME=Your_env_Name
```

例如：

```
SET MQNAME=CLIENT1
```

如果您在 IBM MQ MQI client 上同時執行多個 IBM MQ 應用程式，則部分 NetBIOS 實作需要由 **MQNAME** 設定的唯一名稱。

如需相關資訊，請參閱 [第 205 頁的『定義 IBM MQ 本端 NetBIOS 名稱』](#)。

MQNOREMPOOL

當您設定 **MQNOREMPOOL** 環境變數時，它會關閉通道儲存區，並導致通道作為接聽器的執行緒執行。

如需相關資訊，請參閱 [MCATYPE](#) (訊息通道代理程式類型)。

MQPSE_TRACE_LOGFILE

當您執行「發佈結束程式範例程式」AMQSPSE0 時，您可以使用 **MQPSE_TRACE_LOGFILE** 環境變數，這是結束程式的範例 C 程式，用來在發佈遞送給訂閱者之前截取它。在要追蹤的應用程式程序中，此環境變數說明必須寫入追蹤檔的位置。

如需相關資訊，請參閱 [發佈結束程式範例程式](#)。

MQS_AMSCRED_KEYFILE

您可以使用 **MQS_AMSCRED_KEYFILE** 環境變數來置換或提供要在 IBM MQ Advanced Message Security (AMS) 應用程式執行時期使用的起始金鑰檔，或當您使用 **runamscred** 指令來保護金鑰儲存庫配置檔時使用。

如需相關資訊，請參閱 [搭配使用金鑰儲存庫和憑證與 AMS](#)，以及 [保護 IBM MQ 元件配置檔中的密碼](#)。

MQS_DISABLE_ALL_INTERCEPT

當您嘗試從舊版產品連接至佇列管理程式，且將 IBM MQ 與原生 C 用戶端搭配使用時，如果報告 2085 (MQRC_UNKNOWN_OBJECT_NAME) 錯誤，則可以使用 **MQS_DISABLE_ALL_INTERCEPT** 環境變數來停用 IBM MQ Advanced Message Security (AMS)。

註：您只能將 **MQS_DISABLE_ALL_INTERCEPT** 環境變數用於 C 用戶端。對於 Java 用戶端，您需要改用 **AMQ_DISABLE_CLIENT_AMS** 環境變數。

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

MQS_IPC_HOST

因為 IPC 檔案系統物件必須由系統識別，所以會將佇列管理程式執行所在之每一個系統的子目錄新增至目錄路徑。如果產生的主機名稱值產生問題，您可以使用 **MQS_IPC_HOST** 環境變數來設定主機名稱。

如需相關資訊，請參閱 [在 Multiplatforms 上共用 IBM MQ 檔案](#)。

MQS_KEYSTORE_CONF

MQS_KEYSTORE_CONF 環境變數指定 IBM MQ Advanced Message Security (AMS) 的金鑰儲存庫配置檔位置 (如果該檔案不在預設位置 `home_directory/.mq/keystore.conf` 中)。

如需相關資訊，請參閱 [搭配使用金鑰儲存庫和憑證與 AMS](#)。

如果您在 Managed File Transfer 上遇到問題，請參閱 [如果 MFT 未從 AMS 中的金鑰儲存庫配置檔讀取金鑰儲存庫內容，怎麼辦](#)。

MQS_TRACE_OPTIONS

AIX

對於 AIX 上的選擇性元件追蹤，請使用 **MQS_TRACE_OPTIONS** 環境變數來個別啟動高詳細資料及參數追蹤功能。

註：只有在「IBM 支援中心」指示您這樣做時，才會設定 **MQS_TRACE_OPTIONS** 環境變數。

如需相關資訊，請參閱 [在 UNIX and Linux 上追蹤](#)。

MQSERVER

MQSERVER 環境變數用來定義最小通道。 **MQSERVER** 指定 IBM MQ 伺服器的位置，以及要使用的通訊方法。

註：您無法使用 **MQSERVER** 來定義 TLS 通道或具有通道結束程式的通道。如需如何定義 TLS 通道的相關資訊，請參閱 [使用 TLS 保護通道](#)。

下列範例顯示如何設定 **MQSERVER**：

- Linux UNIX 在 UNIX and Linux 上：

```
export MQSERVER='SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)'
```

- **Windows** 在 Windows 上:

```
SET MQSERVER=SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)
```

- **IBM i** 在 IBM i 上:

```
ADDENVVAR ENVVAR(MQSERVER) VALUE('SYSTEM.DEF.SVRCONN/TCP/AMACHINE.ACOMPANY.COM(1414)')
```

註:

- 通道名稱不能包含正斜線 (/) 字元，因為這個字元是用來區隔通道名稱、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。
- 傳輸類型可以是 LU62、TCP、NETBIOS、SPX 之一，視您的 IBM MQ 用戶端平台而定。
- 連線名稱必須是完整網路名稱。例如 AMACHINE.ACOMPANY.COM(1414)。
- 連線名稱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。連線名稱清單可用作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，則可以使用連線名稱清單來指定不同的佇列管理程式實例。

如果您使用 **MQSERVER** 環境變數來定義 IBM MQ MQI client 機器與伺服器機器之間的通道，則這是應用程式唯一可用的通道，且不會參照用戶端通道定義表 (CCDT)。

如需相關資訊，請參閱 [第 31 頁的『使用 MQSERVER 在 IBM MQ MQI client 上建立用戶端連線通道』](#)。

MQSENAUT

當您將 **MQSNOAUT** 環境變數設為任何值時，它會停用物件權限管理程式 (OAM) 並防止任何安全檢查。這可能適用於測試環境。

只有在建立佇列管理程式時，**MQSNOAUT** 環境變數才會生效。



警告: 若要啟用 OAM，您必須刪除佇列管理程式、刪除環境變數，然後重建佇列管理程式，而不指定 **MQSNOAUT**。

如需相關資訊，請參閱 [防止 AIX、Linux 及 Windows 系統上的安全存取檢查](#)。

MQSPREFIX

除了變更預設字首，您也可以使用 **MQSPREFIX** 環境變數來置換 **crtmqm** 指令的 **DefaultPrefix**。

如需相關資訊，請參閱 [IBM MQ 檔名](#) 及 [mqsc.ini 檔的 AllQueueManagers 段落](#)。

MQSSLCRYP



MQSSLCRYP 環境變數包含參數字串，可用來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 **MQSSLCRYP**，請使用下列其中一個指令:

- **Linux** **UNIX** 在 UNIX and Linux 系統上:

```
export MQSSLCRYP=string
```

- **Windows** 在 Windows 系統上:

```
SET MQSSLCRYP=string
```

如需相關資訊，請參閱 [配置 UNIX, Linux, and Windows 上的加密硬體](#)。

MQSSLFIPS

MQSSLFIPS 環境變數指定在 IBM MQ 中執行加密法時是否僅使用 FIPS 認證的演算法。您可以將此環境變數設為 YES 或 NO。預設值是 NO。這些值與 **ALTER QMGR** 指令的 **SSLFIPS** 參數相同。

若要設定 **MQSSLFIPS**，請使用下列其中一個指令：

- Linux UNIX 在 UNIX and Linux 系統上：

```
export MQSSLFIPS=YES|NO
```

- Windows 在 Windows 系統上：

```
SET MQSSLFIPS=YES|NO
```

- IBM i 在 IBM i 上：

```
ADDENVVAR ENVVAR(MQSSLFIPS) VALUE(YES|NO)
```

使用加密硬體會影響 FIPS 認證演算法的使用。如需相關資訊，請參閱 [指定在執行時期只在 MQI 用戶端上使用 FIPS 認證的 CipherSpecs](#)。

MQSSLKEYR

MQSSLKEYR 環境變數以詞幹格式指定金鑰儲存庫的位置，該金鑰儲存庫用於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。

若要設定 **MQSSLKEYR**，請使用下列其中一個指令：

- Linux UNIX 在 UNIX and Linux 系統上：

```
export MQSSLKEYR=pathname
```

- Windows 在 Windows 系統上：

```
SET MQSSLKEYR=pathname
```

- IBM i 在 IBM i 上：

```
ADDENVVAR ENVVAR(MQSSLKEYR) VALUE(pathname)
```

此環境變數沒有預設值。

如需相關資訊，請參閱 [ALTER QMGR](#) 指令的 **SSLKEYR** 參數。

MQSSLPROXY

MQSSLPROXY 環境變數指定 GSKit for OCSP 檢查要使用之 HTTP Proxy 伺服器的主機名稱及埠號。

若要設定 **MQSSLPROXY**，請使用下列其中一個指令：

- Linux UNIX 在 UNIX and Linux 系統上：

```
export MQSSLPROXY="string"
```

- **Windows** 在 Windows 系統上：

```
SET MQSSLPROXY= string
```

您使用 **MQSSLPROXY** 指定的字串可以是 GSKit 用於 OCSP 檢查之 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。

Linux

UNIX

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"
```

如需相關資訊，請參閱 [使用線上憑證狀態通訊協定 \(OCSP\)](#)。

MQSSLRESET

MQSSLRESET 環境變數指定在重新協議 TLS 秘密金鑰之前，在 TLS 通道上傳送及接收的未加密位元組數。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 1 位元組到 32 KB 範圍內的 TLS 秘密金鑰重設計數，則 TLS 通道會使用 32 KB 的秘密金鑰重設計數。此秘密重設計數是為了避免對小型 TLS 秘密金鑰重設值進行過多的金鑰重設。

若要設定 **MQSSLRESET**，請使用下列其中一個指令：

- **Linux** **UNIX** 在 UNIX and Linux 系統上：

```
export MQSSLRESET=integer
```

- **Windows** 在 Windows 系統上：

```
SET MQSSLRESET=integer
```

- **IBM i** 在 IBM i 上：

```
ADDENVVAR ENVVAR(MQSSLRESET) VALUE(integer)
```

如需相關資訊，請參閱 [重設 SSL 和 TLS 秘密金鑰](#)。

MQSUITEB

ULW

您可以在 UNIX, Linux, and Windows 平台上配置 IBM MQ 以符合 NSA Suite B 標準。

MQSUITEB 環境變數指定是否使用 Suite B 相容加密法。如果要使用套組 B 加密法，您可以將 **MQSUITEB** 設為下列其中一項，以指定加密法的強度：

- 無
- 128_BIT, 192_BIT
- 128_BIT
- 192_BIT

您可以使用逗點區隔清單來指定多個值。將值 NONE 與任何其他值搭配使用是無效的。

如需相關資訊，請參閱 [配置 IBM MQ for Suite B](#)。

MQTCPTIMEOUT

MQTCPTIMEOUT 環境變數指定 IBM MQ 等待 TCP 連接呼叫的時間長度。

ODQ_MSG

如果您使用與 **runmqdlq** 不同的無法傳送郵件的佇列處理程式，則可以使用範例來源 **amqsdlq** 作為基本程式。範例類似於產品內提供的無法傳送郵件的處理程式，但追蹤和錯誤報告不同。使用 **ODQ_MSG** 環境變數來設定包含錯誤及參考訊息的檔案名稱。所提供的檔案稱為 **amqsdlq.msg**。

如需相關資訊，請參閱 [無法傳送郵件的佇列處理程式範例](#)。

ODQ_TRACE

如果您使用與 **runmqdlq** 不同的無法傳送郵件的佇列處理程式，則可以使用範例來源 **amqsdlq** 作為基本程式。範例類似於產品內提供的無法傳送郵件的處理程式，但追蹤和錯誤報告不同。若要啟用追蹤，請將 **ODQ_TRACE** 環境變數設為 **YES** 或 **yes**。

如需相關資訊，請參閱 [無法傳送郵件的佇列處理程式範例](#)。

OMQ_PATH

OMQ_PATH 環境變數指定當 ActiveX Script 的 IBM MQ 自動化類別失敗時，您可以在其中找到「第一個失敗症狀」報告的位置。

OMQ_TRACE

ActiveX (MQAX) 的 IBM MQ 自動化類別包括追蹤機能，可協助「IBM 支援中心」識別發生問題時的狀況。它會顯示您執行 MQAX Script 時所採用的路徑。除非您有問題，否則請在追蹤設為關閉的情況下執行，以避免不必要的使用系統資源。**OMQ_TRACE** 是您設定用來控制追蹤的三個環境變數之一。為 **OMQ_TRACE** 指定任何值會開啟追蹤機能。即使您將 **OMQ_TRACE** 設為 **OFF**，追蹤仍會在作用中。

如需相關資訊，請參閱 [控制 IBM MQ ActiveX](#)。

OMQ_TRACE_LEVEL

OMQ_TRACE_LEVEL 是您設定用來控制 ActiveX 之 IBM MQ 自動化類別追蹤的三個環境變數之一。它會設定必要的追蹤層次。大於 9 的值不會在追蹤檔中產生任何其他資訊。

如需相關資訊，請參閱 [控制 IBM MQ ActiveX](#)。

OMQ_TRACE_PATH

OMQ_TRACE_PATH 是您設定用來控制 ActiveX 之 IBM MQ 自動化類別追蹤的三個環境變數之一。它會設定寫入追蹤檔的追蹤目錄。

如需相關資訊，請參閱 [控制 IBM MQ ActiveX](#)。

ONCONFIG

ONCONFIG 環境變數指定 Informix 伺服器配置檔的名稱。例如，在 UNIX and Linux 系統上，請使用下列指令：

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

在 Windows 系統上，請使用下列指令：

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

如需相關資訊，請參閱 [配置 Informix](#)。

WCF_TRACE_ON

WCF 自訂通道有兩種不同的追蹤方法可用。這兩種追蹤方法會獨立或一起啟動。每一個方法都會產生它自己的追蹤檔，因此當兩個追蹤方法都已啟動時，會產生兩個追蹤輸出檔。啟用和停用兩種不同的追蹤方法有四個組合。除了這些組合以啟用 WCF 追蹤外，還可以使用 **WCF_TRACE_ON** 環境變數來啟用 XMS .NET 追蹤。

如需相關資訊，請參閱 [追蹤 IBM MQ 的 WCF 自訂通道](#)。

WMQSOAP_HOME

在 IBM MQ 中正確安裝並配置 .NET SOAP over JMS 服務管理環境之後，完成其他配置步驟時，會使用 **WMQSOAP_HOME** 環境變數。它可從本端佇列管理程式存取。

如需相關資訊，請參閱 [WCF 用戶端至 IBM MQ 範例所管理的 .NET 服務](#) 及 [WCF 用戶端至 IBM MQ 範例所管理的 Axis Java 服務](#)。

XMS_TRACE_ON、XMS_TRACE_FILE_PATH、XMS_TRACE_FORMAT 及 XMS_TRACE_SPECIFICATION

如果您使用 IBM MQ classes for XMS .NET Framework，則可以從應用程式配置檔以及從 XMS 環境變數配置追蹤。

V 9.1.1 如果您使用 IBM MQ classes for XMS .NET Standard，則必須從 XMS 環境變數配置追蹤。追蹤通常是在「IBM 支援中心」的指引下使用。

若要啟用及配置 XMS .NET 應用程式的追蹤，請在執行應用程式之前設定下列環境變數：

XMS_TRACE_ON

如果設定 **XMS_TRACE_ON** 環境變數，依預設會啟用所有追蹤。

XMS_TRACE_FILE_PATH

如果您要將追蹤及 FFDC 記錄寫入現行工作目錄中的替代位置，則 **XMS_TRACE_FILE_PATH** 環境變數會指定該目錄的完整路徑名稱。

XMS_TRACE_FORMAT

XMS_TRACE_FORMAT 環境變數指定必要的追蹤格式，可以是 BASIC 或 ADVANCED。

XMS_ 追蹤_ 規格

XMS_TRACE_SPECIFICATION 環境變數會置換 應用程式配置檔的追蹤區段中所定義的追蹤設定。

XMS_TRACE_SPECIFICATION 僅適用於 IBM MQ classes for XMS .NET Framework。

如需相關資訊，請參閱 [追蹤 XMS .NET 應用程式](#) 及 [使用 XMS 環境變數來追蹤 XMS .NET 應用程式](#)。

相關工作

第 54 頁的『使用 IBM MQ 環境變數』

您可以使用指令來顯示現行設定或重設 IBM MQ 環境變數的值。

Multi 在 Multiplatforms 上的 .ini 檔中變更 IBM MQ 配置資訊

您可以編輯配置檔 (.ini) 中的資訊，來變更 IBM MQ 或個別佇列管理程式的行為，以符合安裝的需求。您也可以變更 IBM MQ MQI clients 的配置選項。

關於這項作業

您可以透過變更在控管 IBM MQ 的一組配置屬性 (或參數) 上指定的值，來變更節點或佇列管理程式層次的 IBM MQ 配置資訊。

配置檔 (或段落檔) 包含一或多個段落，這些段落是 .ini 檔案中的行群組，它們一起具有一般功能或定義部分系統 (例如日誌功能、通道功能及可安裝服務)。您可以在下列配置檔內修改 IBM MQ 配置屬性：

IBM MQ 配置檔 mqs.ini

mqs.ini 檔案會影響整個節點的變更。每一個 IBM MQ 安裝架構都有一個 mqs.ini 檔。

因為使用 IBM 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 MQ 配置檔中未定義的佇列管理程式。

佇列管理程式配置檔 `qm.ini`

`qm.ini` 檔案會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。

IBM MQ MQI client 配置檔 `mqclient.ini`

IBM MQ MQI clients 的配置選項分別保留在用戶端配置檔中，通常稱為 `mqclient.ini`。

活動追蹤配置檔 `mqat.ini`


`mqat.ini` 檔用來配置活動追蹤行為。


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，請從備份回復它。)
- 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。

重要： 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。

關於編輯配置檔的注意事項：

- 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在配置檔中定義的值。
 - `qm.ini` 檔案中定義的值優先於 `mqs.ini` 檔案中定義的值。
- 安裝之後，您可以編輯 IBM MQ 配置檔中的預設值。
- 備份佇列管理程式時，請記得同時併入其配置檔 (`qm.ini`) 及中央 IBM MQ 配置檔 (`mqs.ini`)。
- 如果您在配置檔屬性上設定不正確的值，則效果與完全遺漏屬性相同。系統不處理此值，並發出操作員訊息以指出問題。
-  在 IBM i 上，`.ini` 檔案是常駐於 IFS 中的串流檔。
- `mqat.ini` 檔案的格式有許多語法規則。如需相關資訊，請參閱應用程式活動追蹤 [使用 `mqat.ini` 配置活動追蹤行為](#)。

 在 UNIX 或 Linux 上，安裝配置檔 `mqinst.ini` 包含所有 IBM MQ 安裝的相關資訊。`mqinst.ini` 檔案不能直接編輯或參照，因為其格式不是固定的，而且可能會變更。相反地，您必須使用指令來編輯它。如需相關資訊，請參閱 [第 125 頁的『安裝配置檔 `mqinst.ini`』](#)。

程序

1. 在編輯配置檔之前，請先備份它，以便在有需要時您可以回復至其中的副本。
2. 使用下列其中一種方式來編輯 `.ini` 配置檔：
 - 使用標準文字編輯器手動執行。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，則可以使用 "\" 字元作為該字元的字首。然後會使用該字元作為配置資料的一部分。
 - 使用指令來自動變更節點上佇列管理程式的配置。如需相關資訊，請參閱 [指令參照](#)。

 例如，Windows 特定指令 `amqmdain` 會自動更新 `qm.ini` 內容的子集。如需相關資訊，請參閱 `amqmdain`。

-   在 Linux (x86 和 x86-64) 及 Windows 上，您可以使用 IBM MQ Explorer 來更新 `qm.ini` 內容的子集。如需相關資訊，請參閱 [使用 MQ 探險家配置 IBM MQ](#)。

註： 因為變更可安裝服務及其元件有重大含意，所以可安裝服務在 IBM MQ Explorer 中是唯讀的。因此，您必須透過編輯 `qm.ini` 檔案，對可安裝服務進行任何變更。如需相關資訊，請參閱 [第 113 頁的『`qm.ini` 檔案的服務段落](#)』。

Multi IBM MQ 配置檔 mqs.ini

IBM 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。

註: 如需如何及何時編輯 `mqs.ini` 檔案以及您對檔案所做的任何變更生效的相關資訊, 請參閱 [第 72 頁的『在 Multiplatforms 上的 .ini 檔中變更 IBM MQ 配置資訊』](#)。

目錄位置

Linux **UNIX** 在 UNIX and Linux 上, 資料目錄和日誌目錄一律分別為 `/var/mqm` 和 `/var/mqm/log`。

Windows 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。UNIX and Linux 系統上 `mqinst.ini` 中包含的安裝配置資訊也位於登錄中, 因為 Windows 上沒有 `mqinst.ini` 檔案 (請參閱 [第 125 頁的『安裝配置檔 mqinst.ini』](#))。

Windows Windows 系統的 `mqs.ini` 檔案由 `HKLM\SOFTWARE\IBM\IBM 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

IBM i 在 IBM i 上, `mqs.ini` 檔案儲存在 `/QIBM/UserData/mqm` 中。檔案包含:

- 佇列管理程式的名稱。
- 預設佇列管理程式的名稱。
- 與每一個佇列管理程式相關聯的檔案位置。
- 識別任何 API 結束程式的資訊 (如需相關資訊, 請參閱 [配置 API 結束程式](#))。

特別是使用 `mqs.ini` 檔案來尋找與每一個佇列管理程式相關聯的資料。

UNIX and Linux 的 mqs.ini 檔範例

```
Linux UNIX

#*****#
#* Module Name: mqs.ini                               **#
#* Type       : IBM MQ Machine-wide Configuration File **#
#* Function   : Define IBM 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

```

Windows 的 mqs.ini 檔範例

Windows

```

#*****#
#* Module Name: mqs.ini                                     #*
#* Type       : IBM MQ Machine-wide Configuration File    #*
#* Function   : Define IBM 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=C:\ProgramData\IBM\MQ

LogDefaults:
LogPrimaryFiles=3
LogSecondaryFiles=2
LogFilePages=4096
LogType=CIRCULAR
LogBufferPages=0
LogDefaultPath=C:\ProgramData\IBM\MQ\log

QueueManager:
Name=saturn.queue.manager
Prefix=C:\ProgramData\IBM\MQ
Directory=saturn!queue!manager
InstallationName=Installation1

QueueManager:
Name=pluto.queue.manager
Prefix=C:\ProgramData\IBM\MQ
Directory=pluto!queue!manager
InstallationName=Installation2

DefaultQueueManager:
Name=saturn.queue.manager

ApiExitTemplate:
Name=OurPayrollQueueAuditor
Sequence=2
Function=EntryPoint
Module=C:\usr\ABC\auditor
Data=123

ApiExitCommon:

```

```
Name=MQPoliceman
Sequence=1
Function=EntryPoint
Module=C:\usr\MQPolice\tmpq
Data=CheckEverything
```

IBM i 的 mqs.ini 檔範例

IBM i

```
*****#
#* Module Name: mqs.ini                                     #*
#* Type       : IBM MQ Configuration File                 #*
#* Function   : Define IBM MQ resources for the node     #*
#*           :                                           #*
#* Notes     :                                           #*
#* 1) This is an example IBM MQ configuration file       #*
#*           :                                           #*
#* AllQueueManagers:                                     #*
#* The path to the qmgrs directory, within which queue manager data #*
#* is stored                                           #*
#* DefaultPrefix=/QIBM/UserData/mqm                     #*
#*
QueueManager:
Name=saturn.queue.manager
Prefix=/QIBM/UserData/mqm
Library=QMSATURN.Q
Directory=saturn!queue!manager

QueueManager:
Name=pluto.queue.manager
Prefix=/QIBM/UserData/mqm
Library=QMPLUTO.QU
Directory=pluto!queue!manager

DefaultQueueManager:
Name=saturn.queue.manager
```

附註:

1. 節點上的 IBM MQ 會使用佇列管理程式及日誌登載的預設位置。
2. 佇列管理程式 saturn.queue.manager 是節點的預設佇列管理程式。與此佇列管理程式相關聯的檔案目錄已自動轉換為檔案系統的有效檔名。
3. 因為使用 IBM 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 IBM MQ 指令失敗。此外，應用程式也無法連接至 IBM MQ 配置檔中未定義的佇列管理程式。

mqs.ini 段落



小心: 本主題鏈結至 mqs.ini 檔案中各段落的相關資訊。每一個段落都包含該段落中參數的相關資訊。

Multi

mqs.ini 檔案段落及屬性的摘要

IBM MQ 配置檔 mqs.ini 的段落屬性摘要，以及指向相關資訊的鏈結。

表 9: mqs.ini 檔案的段落	
段落及屬性	屬性的說明
AllQueue 管理程式段落	
DefaultPrefix	qmgrs 目錄的路徑，佇列管理程式資料保留在其中。

表 9: mqs.ini 檔案的段落 (繼續)





段落及屬性	屬性的說明
  DefaultEphemeral 字首	在其中保留佇列管理程式暫時資料的目錄路徑。
 ConvEBCDICNewline	IBM MQ 如何將 EBCDIC NL 字元轉換成 ASCII 格式
ApiExit 共用段落及 ApiExit 範本段落	
<u>名稱</u>	在 MQAXP 結構的 ExitInfo 名稱欄位中傳遞給它的 API 結束程式敘述性名稱。
<u>函數</u>	包含 API 結束碼之模組的函數進入點名稱。
<u>模組</u>	包含 API 結束碼的模組。
<u>資料</u>	要在 MQAXP 結構的 ExitData 欄位中傳遞至 API 結束程式的資料。
<u>順序</u>	相對於其他 API 結束程式，呼叫此 API 結束程式的順序。
DefaultQueue 管理程式段落	
<u>名稱</u>	處理未明確指定佇列管理程式名稱之任何指令的佇列管理程式名稱。
ExitProperties 段落	
<u>CLWLMode</u>	叢集 workloac (CLWL) 結束程式是以 FAST 模式還是 SAFE 模式執行。
LogDefaults 段落	
<u>LogPrimaryFiles</u>	建立佇列管理程式時所配置的日誌檔。
<u>LogSecondaryFiles</u>	當主要檔耗盡時所配置的日誌檔。
<u>LogFilePages</u>	日誌檔頁數。(日誌檔大小以 4 KB 大小分頁為單位來指定。)
<u>LogType</u>	佇列管理程式要使用的記載類型 (循環或線性)。
<u>LogBufferPages</u>	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)。
<u>LogDefaultPath</u>	佇列管理程式的日誌檔所在的目錄。
<u>LogWriteIntegrity</u>	日誌程式用來可靠地寫入日誌記錄的方法。
QueueManager 段落	
<u>名稱</u>	佇列管理程式的名稱。
<u>字首</u>	儲存佇列管理程式檔案的位置。
<u>目錄</u>	儲存佇列管理程式檔案之 prefix\QMGRS 目錄下的子目錄名稱。
<u>DataPath</u>	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。
<u>InstallationName</u>	與此佇列管理程式相關聯的 IBM MQ 安裝架構名稱。
 EphemeralPrefix	儲存佇列管理程式暫時資料的位置。

表 9: *mqs.ini* 檔案的段落 (繼續)

段落及屬性	屬性的說明
HA	若為 HA 及 DR/HA 抄寫的資料佇列管理程式，此值為 Replicated。 此屬性會在建立佇列管理程式時新增，且不應該變更。
DR	若為 DR/HA 抄寫的資料佇列管理程式，此值為 Replicated。 此屬性會在建立佇列管理程式時新增，且不應該變更。
DRRole	針對 DR 抄寫資料佇列管理程式而存在，指出佇列管理程式的現行 DR 角色。 當 IBM MQ 建立並更新佇列管理程式時，會新增屬性。不應手動變更。

Multi **mqs.ini 檔案的 AllQueue 管理程式段落**

AllQueue 段落可以指定 qmgrs 目錄的路徑，其中儲存與佇列管理程式相關聯的檔案，可執行程式庫的路徑，以及將 EBCDIC 格式資料轉換成 ASCII 格式的方法。

使用 *mqs.ini* 檔中的 AllQueue 段落，以指定所有佇列管理程式的相關資訊。

Windows **Linux** 或者，在 Linux (x86 和 x86-64) 和 Windows 上，請使用 IBM MQ Explorer General 和 Extended IBM MQ 內容頁。

DefaultPrefix= directory_name

此屬性指定 qmgrs 目錄的路徑，在該目錄中保留佇列管理程式資料。

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 IBM 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 IBM MQ。

註: **ULW** 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。

V 9.1.3 **ULW** **DefaultEphemeralPrefix = directory_name**

此屬性指定在其中保留佇列管理程式暫時資料的目錄路徑 (例如 IPC Socket)，並且在建立佇列管理程式時僅用來設定佇列管理程式的 EphemeralPrefix。此外，如果您變更預設值，則必須自行建立目錄。

您必須建立具有許可權的暫時資料目錄，以容許 IBM MQ 群組存取權寫入該目錄。

作為變更 *mqs.ini* 檔案的替代方案，您可以使用環境變數 **MQ_EPHEMERAL_PREFIX** 來置換 **crtmqm** 指令的 **DefaultEphemeralPrefix**。

由於作業系統限制，預設暫時字首在 UNIX and Linux 平台上限制為 24 個字元。

DefaultEphemeralPrefix 在 IBM MQ Appliance 或 IBM i 上不受支援。

Multi **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 IBM MQ 如何將 EBCDIC NL 字元轉換成 ASCII 格式。

IBM i 在 IBM MQ for IBM i 上，CCSID 1253 被視為 ISO CCSID，NL_TO_LF 會影響 ISO 及 ASCII 轉換。

z/OS **ConvEBCDICNewline** 屬性在 z/OS 上無法使用。z/OS 上的行為相當於 ConvEBCDICNewline=TABLE。請注意，其他平台上的預設值可能不同。

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 79 頁的表 10 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。

V 9.1.2 **V 9.1.0.2** 從 IBM MQ 9.1.0 Fix Pack 2 和 IBM MQ 9.1.2 開始，您可以使用 **AMQ_CONVEBDICNEWLINE** 環境變數而非 **ConvEBCDICNewline** 段落屬性，例如，在無法使用 mqs.ini 檔案的情況下，在用戶端提供 **ConvEBCDICNewline** 功能。環境變數採用與 **ConvEBCDICNewline** 屬性相同的值 (NL_TO_LF、TABLE 或 ISO)。如果同時設定屬性及環境變數，則段落屬性優先。

Multi **ApiExitCommon** 及 **ApiExit mqs.ini** 檔案的範本段落

ApiExit 範本及 ApiExit 共用段落會識別所有佇列管理程式的 API 結束常式。

使用 mqs.ini 檔案中的 ApiExit 範本及 ApiExit 共用段落，以識別所有佇列管理程式的 API 結束常式。(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 93 頁的『ApiExit qm.ini 檔案的本端段落』中所述。)

Windows **Linux** 或者，在 Linux (x86 和 x86-64) 和 Windows 上，請使用 IBM MQ Explorer Exits IBM MQ 內容頁面。

Windows 在 Windows 上，您也可以使用 **amqmdain** 指令來變更 API 結束程式的項目。

如需使用這些屬性的相關資訊，請參閱 [配置 API 結束程式](#)。

Name=ApiExit_name

在 MQAXP 結構的 ExitInfo 名稱欄位中傳遞給它的 API 結束程式敘述性名稱。

此名稱必須是唯一的，長度不得超過 48 個字元，且只包含 IBM MQ 物件名稱的有效字元 (例如，佇列名稱)。

Function=function_name

包含 API 結束碼之模組的函數進入點名稱。此進入點是 MQ_INIT_EXIT 函數。

這個欄位的長度限制為 MQ_EXIT_NAME_LENGTH。

模組 = 模組名稱

包含 API 結束碼的模組。

如果這個欄位含有模組的完整路徑名稱，則會依其現狀使用。如果此欄位只包含模組名稱，則會使用 qm.ini 檔案的 ExitPath 段落中的 **ExitsDefaultPath** 屬性來尋找模組。

在支援個別執行緒程式庫的平台上，您必須提供 API 結束程式模組的非執行緒及執行緒版本。執行緒版本必須具有 **_r** 字尾。在載入之前，IBM MQ 應用程式 Stub 的執行緒版本會隱含地將 **_r** 附加至給定的模組名稱。

此欄位的長度限制為平台支援的路徑長度上限。

Data=data_name

要在 MQAXP 結構的 ExitData 欄位中傳遞至 API 結束程式的資料。

如果您包括此屬性，則會移除前導及尾端空白，將剩餘字串截斷為 32 個字元，並將結果傳遞至結束程式。如果您省略此屬性，則會將預設值 32 個空白傳遞給結束程式。

此欄位的長度上限為 32 個字元。

Sequence=sequence_number

相對於其他 API 結束程式，呼叫此 API 結束程式的順序。在具有較高序號的結束程式之前，會先呼叫具有較低序號的結束程式。結束程式的序號不需要連續。序列 1、2、3 與序列 7、42、1096 具有相同的結果。如果兩個結束程式具有相同的序號，則佇列管理程式會決定要先呼叫哪個結束程式。您可以在 MQAXP 中的 ExitChainAreaPtr 所指出的 ExitChain 區域中放置時間或標記，或寫入您自己的日誌檔，以識別事件之後呼叫的項目。

此屬性是不帶正負號的數值。

Multi DefaultQueue mqs.ini 檔案的管理程式段落

DefaultQueue 管理程式段落指定節點的預設佇列管理程式。

使用 mqs.ini 檔中的 DefaultQueue 管理程式段落來指定預設佇列管理程式。

Windows **Linux** 或者，在 Linux (x86 和 x86-64) 和 Windows 上，請使用 IBM MQ Explorer General IBM MQ 內容頁面。

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。

Multi mqs.ini 檔案的 ExitProperties 段落

ExitProperties 段落指定佇列管理程式結束程式所使用的配置選項。

使用 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。

Windows **Linux** 或者，在 Linux (x86 和 x86-64) 和 Windows 上，請使用 IBM MQ Explorer Extended IBM MQ 內容頁面。

CLWLMode= SAFE (預設值) | FAST

叢集工作量 (CLWL) 結束程式可讓您指定叢集中要開啟的叢集佇列，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。如果省略 **CLWLMode** 屬性，則叢集工作量結束程式會以 SAFE 模式執行。

安全

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。如果您確信 CLWL 結束程式沒有問題，且您特別關心效能，則只應該在 FAST 模式下執行 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。

Multi **mq5.ini 檔案的 LogDefaults 段落**

LogDefaults 段落指定所有佇列管理程式的日誌預設值相關資訊。

使用 mq5.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。

Windows **Linux** 或者，在 Linux (x86 和 x86-64) 和 Windows 上，請使用 IBM MQ Explorer Default log settings IBM MQ 內容頁面。

如果段落不存在，則會使用 IBM 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。如需此指令的相關資訊，請參閱 **crtmqm**。

建立佇列管理程式之後，會從第 108 頁的『qm.ini 檔案的日誌段落』中說明的設定取得該佇列管理程式的日誌屬性。

註：為新的 IBM MQ 安裝提供的 LogDefaults 段落不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。LogDefaults 段落的預設值顯示在第 74 頁的『UNIX and Linux 的 mq5.ini 檔範例』和第 75 頁的『Windows 的 mq5.ini 檔範例』中。LogBufferPages 屬性的零值表示 512。

在第 78 頁的『mq5.ini 檔案的 AllQueue 管理程式段落』中指定的預設字首，以及在第 108 頁的『qm.ini 檔案的日誌段落』中指定給特定佇列管理程式的日誌路徑，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 507 頁的『計算日誌的大小』。

註：下列參數清單中給定的限制是 IBM 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。

LogPrimary 檔案 = 3 (預設值) |2-254 (Windows) |2-510 (UNIX and Linux)

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 上則為 510。預設是 3。

主要及次要日誌檔的總數在 Windows 上不得超過 255，在 UNIX and Linux 上不得超過 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 (預設值) | 1-253 (Windows) | 1-509 (UNIX and Linux)

當主要檔耗盡時所配置的日誌檔。

在 Windows 上，次要日誌檔的數目下限為 1，而在 UNIX and Linux 上，次要日誌檔的數目上限為 253。預設數字為 2。

主要及次要日誌檔的總數在 Windows 上不得超過 255，在 UNIX and Linux 上不得超過 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 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，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 (預設值) | LINEAR

要使用的日誌類型。預設值為 CIRCULAR。

循環式

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 503 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 503 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。

 從 IBM MQ 9.1.0 開始，您可以在建立佇列管理程式之後變更記載方法。如需相關資訊，請參閱 [migmqlog](#)。

LogBufferPages = 0 (預設值) | 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 512 (2048 KB) 的大小。

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 = directory_name

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` 代表 IBM MQ for Windows，其中 `DefaultPrefix` 是在 All Queue Managers IBM MQ 內容頁的 `DefaultPrefix` 屬性上指定的值。此值是在安裝時設定。
-   `/var/mqm/log` 適用於 UNIX 和 Linux 系統。

或者，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 `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 (預設值)

日誌程式用來可靠地寫入日誌記錄的方法。

TripleWrite (預設值)

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

只有在管理 IBM MQ 回復日誌的檔案系統及裝置明確保證 4KB 寫入的最小單位性時，您才應該使用 **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。

註: 如果持續性工作量中有足夠的並行性，則除了預設值 **TripleWrite** 之外，設定任何其他值的潛在好處都是最小的。

如需相關資訊，請參閱第 110 頁的『[LogWriteIntegrity - 使用 SingleWrite 或 TripleWrite](#)』。

Multi mqs.ini 檔案的 QueueManager 段落

QueueManager 段落指定佇列管理程式目錄的位置。

每個佇列管理程式都有一個 QueueManager 段落。此段落的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 IBM 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 mqs.ini 檔中 [所有佇列管理程式](#) 段落的 **DefaultPrefix** 屬性上指定的值相同。

目錄 = name

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 **Prefix** 和 **Directory** 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 IBM MQ 安裝架構名稱。與此佇列管理程式互動時，必須使用此安裝中的指令。

IBM i 媒體庫 = name

儲存與此佇列管理程式相關之 IBM i 物件 (例如，異動日誌及異動日誌接收器) 的檔案庫名稱。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案庫名稱，則可以轉換。

ULW **V 9.1.3** EphemeralPrefix = name

儲存佇列管理程式暫時資料的位置。

依預設，此值不存在，表示資料儲存在「字首」位置下。

此值是在建立佇列管理程式時，從 **MQ_EPHEMERAL_PREFIX** 環境變數的值或 mqs.ini 檔中 [AllQueue 管理程式](#) 段落的 **DefaultEphemeralPrefix** 屬性設定。

相關工作

第 364 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。

Windows 進階配置與電源介面 (ACPI)

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。

使用 IBM MQ Explorer 中的 ACPI IBM MQ 內容頁面，以指定系統接收暫停要求時 IBM MQ 的行為方式。

請注意，只有在「警示監視器」執行時，才會套用 ACPI IBM MQ 內容頁面中指定的設定。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。

Multi 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。當建立與 qm.ini 檔案相關聯的佇列管理程式時，會自動建立該檔案。

註：如需如何及何時編輯 qm.ini 檔案以及您對檔案所做的任何變更何時生效的相關資訊，請參閱 [第 72 頁的『在 Multiplatforms 上的 .ini 檔中變更 IBM MQ 配置資訊』](#)。

從 IBM MQ 9.0.4 及 IBM MQ 9.0.0 Fix Pack 2 開始，**strmqm** 指令會在完全啟動佇列管理程式之前檢查 qm.ini 檔案中 CHANNELS 及 SSL 段落的語法，這可讓您更容易查看錯誤所在，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

Linux **UNIX** 在 UNIX and Linux 系統上，qm.ini 檔案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
```

Windows 在 Windows 系統上，qm.ini 檔案的位置由 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如下：

```
C:\ProgramData\IBM\MQ\qmgrs\QMNAME\qm.ini
```

IBM i qm.ini 檔會保留在 *mqmdata directory/QMNAME/qm.ini* 中，其中 *mqmdata directory* 依預設是 */QIBM/UserData/mqm*，而 *QMNAME* 是套用起始設定檔的佇列管理程式名稱。

註：您可以在 *mqm.ini* 檔案中變更 *mqmdata directory*。

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 [IBM MQ 檔名](#) 和 [IBM i 上的物件名稱](#)。

qm.ini 段落



小心：

- 本主題鏈結至 qm.ini 檔案中各段落的相關資訊。每一個段落都包含該段落中參數的相關資訊，包括適當的範例。
- 每一個段落都會顯示該段落所適用之 IBM MQ for Multiplatforms 的平台。

V 9.1.4

Multi

啟動時自動配置 qm.ini

從 IBM MQ 9.1.4 開始，您可以配置佇列管理程式，在每次啟動佇列管理程式時自動套用檔案或檔案集 (包含 qm.ini 置換) 的內容。

您可以使用此配置來進行修改，並在下一次重新啟動佇列管理程式時自動重播。例如，如果 qm.ini 置換位於裝載磁碟機上，則可以具有集中式配置，其中在啟動時，會將最新版本套用至每個佇列管理程式。

您可以使用此功能，透過使用自動叢集功能來簡化建立統一叢集的作業。如需此範例，請參閱 [從 IBM MQ 9.1.4 建立統一叢集](#)。

註：這些置換僅在佇列管理程式啟動時套用，無法影響佇列管理程式的建立。例如，您無法使用此功能來設定主要日誌檔的數目。

開始之前

您可以使用：

1. 單一檔案，並建立包含 qm.ini 檔案變更的文字檔。
2. 一組 qm.ini 格式檔案：
 - 識別將存在配置的目錄，以及
 - 在該目錄中，建立檔案，每一個檔案都具有副檔名 .ini，例如 *qminisettings.ini*。

一或多個檔案只需要包含變更項目的段落及 **attribute=value** 設定。例如，若要更新「通道」段落中的 **MaxChannels** 屬性，檔案可能包含：

```
Channels:  
MaxChannels=1234
```

請注意，在 qm.ini 置換檔案中，任何以 # 作為字首的行都會被視為註解。

啟用 qm.ini 檔案屬性的自動配置

您可以使用 `crtmqm` 指令的 `-ii` 旗標，並指向特定檔案或目錄，來配置新的佇列管理程式。所提供的值儲存在 `qm.ini` 檔案中的 `AutoConfig` 段落下，作為屬性 `IniConfig`。

您可以透過新增指向有效檔案或目錄的 `AutoConfig` 段落屬性 `IniConfig`，來配置現有佇列管理程式以啟用自動 MQSC 配置。例如：

```
AutoConfig:
  IniConfig=C:\MQ_Configuration\uniclus.ini
```

自動配置如何運作？

在佇列管理程式啟動期間，會驗證 `AutoConfig` 段落屬性 `IniConfig` 所識別的配置，以確保語法有效，然後將佇列管理程式資料樹狀結構以單一 `cached.ini` 檔案儲存在 `autocfg` 目錄中。

在中處理目錄中的多個檔案時，會按字母順序處理它們。

在第一次啟動佇列管理程式期間，由於無法讀取檔案或目錄，導致佇列管理程式無法啟動，且主控台及佇列管理程式錯誤日誌都有適當的錯誤訊息。

在後續重新啟動時，如果無法讀取所指向的檔案或目錄，則會使用先前快取的檔案，且寫入佇列管理程式錯誤日誌的訊息會強調顯示此情況。

當您使用 `strmqm` 指令時，在呼叫佇列管理程式之前，`cached.ini` 檔的內容會套用至 `qm.ini` 檔作為置換。

這確實表示對於待命佇列管理程式，會在處理 `strmqm` 指令時讀取設定，而不是在佇列管理程式變成作用中時讀取設定。

如何建置取代 qm.ini 檔案？

第一次配置自動起始設定配置並啟動佇列管理程式時，會將現行 `qm.ini` 檔案的副本複製到佇列管理程式資料目錄內的 `autoconfig` 子目錄中，作為 `base_qm.ini`。從這裡開始，這被視為基準線。

每次啟動佇列管理程式時（即 `strmqm` 時間），都會捨棄目前作用中的 `qm.ini` 檔案，並取代為 `base_qm.ini` 的副本。然後將 `cached.ini` 檔案中的配置套用至此檔案。

一旦佇列管理程式受到自動配置控制，對 `qm.ini` 檔所做的所有變更都應該透過 `AutoConfig` 段落中使用 `IniConfig` 屬性所指向的一或多個檔案來執行。

因為會在佇列管理程式啟動時移除現有的 `qm.ini` 檔，所以只會將所提供之 `qm.ini` 檔中使用 `IniConfig` 屬性的配置套用至佇列管理程式基線。

如果在先前啟動佇列管理程式時，已透過自動起始設定配置變更段落或屬性，則除非仍在 `IniConfig` 屬性所識別的一或多個檔案中識別這些變更，否則會移除這些變更。

由於在佇列管理程式啟動時重建 `qm.ini` 檔案，這表示對 `qm.ini` 檔案所做的任何手動變更都會遺失。如果您確實需要使變更持續，且無法使用 `IniConfig` 屬性來進行該變更，您可以執行下列其中一項：

- 對 `base_qm.ini` 檔案本身進行變更。
- 刪除 `base_qm.ini` 檔案。

如果您刪除此檔案，則會根據 `qm.ini` 檔案的現行內容，在下次啟動佇列管理程式時重建 `base_qm.ini`。這會在未來的新基準線開始時強迫所有現行變更。

Multi qm.ini 檔案段落及屬性的摘要

佇列管理程式配置檔 `qmi.ini` 的段落屬性摘要，以及相關資訊的鏈結。

表 11: qm.ini 檔案的段落

段落及屬性	屬性的說明
Windows AccessMode 段落	
Windows 存取群組 ¹	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。
ApiExit 本端段落	
名稱	在 MQAXP 結構的 ExitInfo 名稱欄位中傳遞給它的 API 結束程式敘述性名稱。
函數	包含 API 結束碼之模組的函數進入點名稱。
模組	包含 API 結束碼的模組。
資料	要在 MQAXP 結構的 ExitData 欄位中傳遞至 API 結束程式的資料。
順序	相對於其他 API 結束程式，呼叫此 API 結束程式的順序。
V 9.1.4 AutoCluster 段落	
V 9.1.4 類型	自動叢集的類型。唯一有效的選項是 Uniform，它代表統一叢集。
V 9.1.4 ClusterName	自動叢集的名稱。
V 9.1.4 RepositoryName1	自動叢集中第一個完整儲存庫的佇列管理程式名稱。
V 9.1.4 Repository1Conname	自動叢集成員應該如何連接至佇列管理程式的連線名稱 (CONNAME) 值。
V 9.1.4 RepositoryName2	自動叢集中第二個完整儲存庫的佇列管理程式名稱。
V 9.1.4 Repository2Conname	自動叢集成員應該如何連接至佇列管理程式的連線名稱 (CONNAME) 值。
V 9.1.4 AutoConfig 段落	
V 9.1.4 MQSCConfig	完整檔案路徑或目錄路徑，其中所有 *.mqsc 檔案都會套用至每個佇列管理程式啟動時的佇列管理程式。
V 9.1.4 IniConfig	完整檔案路徑或目錄路徑，其中所有 *.ini 檔案都會在每個佇列管理程式啟動時套用至 qm.ini 檔案。
通道段落	
MaxChannels	容許的現行通道數上限。
MaxActiveChannels	容許隨時處於作用中的通道數上限。
MaxInitiators	起始器數目上限。
MQIBindType	應用程式的連結。
PipeLineLength	通道將使用的並行執行緒數目上限。
AdoptNewMCA	哪些類型的通道可以停止現有通道實例，以便在 IBM MQ 收到啟動通道的要求時可以啟動新的通道實例，但發現通道實例已在執行中。
AdoptNewMCATimeout	新通道實例等待舊通道實例結束的時間量 (秒)。

表 11: *qm.ini* 檔案的段落 (繼續)











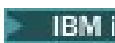
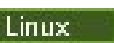

段落及屬性	屬性的說明
AdoptNewMCACheck	啟用 AdoptNewMCA 屬性時所需的檢查類型。
 ChlauthEarly 採用	處理連線鑑別和通道鑑別規則的順序。
PasswordProtection	在 MQCSP 結構中設定受保護的密碼，而不是使用 TLS。
 IgnoreSeqNumberMismatch	控制佇列管理程式在通道啟動期間如何處理序號不符。
連線段落	
DefaultBind 類型	應用程式與在個別處理程序中執行的佇列管理程式之間是否共用部分資源或沒有資源。
DiagnosticMessages 段落	
NAME	段落的名稱。
服務	此段落正在啟用的服務。
ExcludeMessage	不要寫入佇列管理程式錯誤日誌的訊息。
SuppressMessage	在指定時間間隔內，要寫入佇列管理程式錯誤日誌一次的訊息。
 SuppressInterval	在 SuppressMessage 中指定的訊息僅寫入佇列管理程式錯誤日誌一次的時間間隔 (秒)。
嚴重性	以逗點區隔的嚴重性層次清單。
FilePath	寫入日誌檔的路徑。(只有在「服務」屬性設為「檔案」時才支援。)
FilePrefix	日誌檔的字首。(只有在「服務」屬性設為「檔案」時才支援。)
FileSize	日誌輪替的大小。(只有在「服務」屬性設為「檔案」時才支援。)
格式	檔案的格式。(只有在「服務」屬性設為「檔案」時才支援。)
  系統日誌	Syslog 服務會使用 JSON 格式 診斷訊息規格，將任何未過濾的訊息傳送至 syslog。
  ID	與 syslog 項目相關聯的 ident 值。(僅當「服務」屬性設為 Syslog 時才受支援。)
ExitPath 段落	
ExitsDefaultPath	佇列管理程式系統上使用者結束程式的路徑 (32 位元)。
ExitsDefaultPath64	佇列管理程式系統上使用者結束程式的路徑 (64 位元)。
ExitProperties 本端段落	
CLWLMode	叢集 workloac (CLWL) 結束程式是以 FAST 模式還是 SAFE 模式執行。
   檔案系統段落	
   ValidateAuth	容許非 mqm 群組成員的使用者存取錯誤目錄和檔案。

表 11: qm.ini 檔案的段落 (繼續)

段落及屬性	屬性的說明
日誌段落	
LogPrimaryFiles	建立佇列管理程式時所配置的日誌檔。
LogSecondaryFiles	當主要檔耗盡時所配置的日誌檔。
LogFilePages	日誌檔頁數。(日誌檔大小以 4 KB 大小分頁為單位來指定。)
LogType	佇列管理程式要使用的記載類型 (循環或線性)。
LogBufferPages	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)。
LogPath	佇列管理程式的日誌檔所在的目錄。
LogWriteIntegrity	日誌程式用來可靠地寫入日誌記錄的方法。
 LogManagement	用於手動或由佇列管理程式管理日誌範圍的方法。
 LU62 段落	
 TPName	要在遠端站台上啟動的 TP 名稱。
 Library1	APPC DLL 的名稱。
 Library2	與 Library1 相同, 用於將程式碼儲存在兩個個別的程式庫中。
 NETBIOS 段落	
 LocalName	LAN 上用來識別此機器的名稱。
 AdapterNum	LAN 配接卡的號碼。
 NumSess	要配置的階段作業數目。
 NumCmds	要配置的指令數。
 NumNames	要配置的名稱數目。
 Library1	NetBIOS DLL 的名稱。
QMErrorLog 段落	
ErrorLog 大小	指定要複製到備份的佇列管理程式錯誤日誌大小。
ExcludeMessage	指定不要寫入佇列管理程式錯誤日誌的訊息。
SuppressMessage	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。
SuppressInterval	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 (以秒為單位)。
  「受限模式」段落 ²	
  ApplicationGroup	如果未明確定義其目的地的傳輸佇列, 則為放置遠端訊息的本端傳輸佇列名稱。

表 11: *qm.ini* 檔案的段落 (繼續)

段落及屬性	屬性的說明
安全段落	
<u>ClusterQueueAccessControl</u>	請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。
 <u>GroupModel</u>	在決定 Windows 上使用者的群組成員資格時，「物件權限管理程式 (OAM)」是否檢查廣域群組。
服務段落	
<u>名稱</u>	必要服務的名稱。
<u>EntryPoints</u>	為服務定義的進入點數目。
 <u>SecurityPolicy</u>	在 Windows 上，每一個佇列管理程式的安全原則
  <u>SecurityPolicy</u>	在 UNIX and Linux 上，佇列管理程式是使用使用者型或群組型授權。
<u>SharedBindingsUserId</u>	僅適用於共用連結，不論 IdentityContext 結構中的 UserIdentifier 欄位 (來自 MQZ_AUTHENTICATE_USER 函數) 是有效使用者 ID 還是實際使用者 ID。
<u>FastpathBindingsUserId</u>	僅限捷徑連結，不論 IdentityContext 結構中的 UserIdentifier 欄位 (來自 MQZ_AUTHENTICATE_USER 函數) 是有效使用者 ID 還是實際使用者 ID。
<u>IsolatedBindingsUserId</u>	僅限隔離連結，不論 IdentityContext 結構中來自 MQZ_AUTHENTICATE_USER 函數的 UserIdentifier 欄位是有效使用者 ID 還是實際使用者 ID。
ServiceComponent 段落	
<u>服務</u>	必要服務的名稱。
<u>名稱</u>	服務元件的敘述性名稱。
<u>模組</u>	要包含此元件之程式碼的模組名稱。
<u>ComponentData 大小</u>	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)。
 SPX 段落	
 <u>Socket</u>	以十六進位表示法表示的 SPX Socket 號碼。
 <u>BoardNum</u>	LAN 配接卡號碼。
 <u>KeepAlive</u>	開啟或關閉 KeepAlive 函數。
 <u>Library1</u>	SPX DLL 的名稱。
 <u>Library2</u>	與 LibraryName1 相同，用於將程式碼儲存在兩個個別的程式庫中。
 <u>ListenerBacklog</u>	置換 SPX 接聽器的預設未完成要求數。
SSL 段落	
<u>AllowOutboundSNI</u>	指定具有 SNI 功能的用戶端在起始 TLS 連線時，是否將 SNI 設為遠端系統的目標 IBM MQ 通道名稱。

表 11: qm.ini 檔案的段落 (繼續)

段落及屬性	屬性的說明
V 9.1.1 AllowedCipher 規格	指定已啟用與 Multiplatforms 上 IBM MQ 通道搭配使用的 CipherSpecs 自訂清單。
V 9.1.4 AllowTLSV13	佇列管理程式是否能夠使用 TLS 1.3 CipherSpecs。
CDPCheckExtensions	此佇列管理程式上的 TLS 通道是否嘗試檢查 CrlDistribution 點憑證延伸中指定的 CDP 伺服器。
V 9.1.4 MinimumRSAKey 大小	指定 RSA 憑證必須具備才能被接受的金鑰大小下限。
OCSPAAuthentication	無法從 OCSP 伺服器判定撤銷狀態時要採取的動作。
OCSPCheckExtensions	此佇列管理程式上的 TLS 通道是否嘗試檢查在 AuthorityInfoAccess 憑證延伸中命名的 OCSP 伺服器。
V 9.1.5 V 9.1.5 OCSPTimeout	執行撤銷檢查時等待 OCSP 回應者的秒數。
SSLHTTPProxyName	GSKit for OCSP 檢查要使用之 HTTP Proxy 伺服器的主機名稱或網址。
V 9.1.5 V 9.1.5 SSLHTTPConnectTimeout	執行撤銷檢查時，等待順利建立與 HTTP 伺服器的網路連線的秒數。
子儲存區段落 <small>第 92 頁的『3』</small>	此段落由 IBM MQ 建立。不要變更它。
ShortSubpool 名稱 <small>第 92 頁的『3』</small>	對應於在 /var/mqm/sockets 目錄內所建立之目錄和符號鏈結的名稱，IBM MQ 會使用該名稱來進行其執行中處理程序之間的內部通訊。
Windows TCP 段落	
埠號	TCP/IP 階段作業的預設埠號 (以十進位表示法表示)。
Windows Library1	TCP/IP Socket DLL 的名稱。
KeepAlive	開啟或關閉 KeepAlive 函數。
ListenerBacklog	置換 TCP/IP 接聽器的預設未完成要求數。
連接逾時	嘗試連接 Socket 逾時之前的秒數。
SndBuff 大小	通道傳送端所使用的 TCP/IP 傳送緩衝區大小 (以位元組為單位)。
RcvBuff 大小	通道接收端所使用的 TCP/IP 接收緩衝區大小 (以位元組為單位)。
RcvSndBuffSize	接收端通道傳送端所使用的 TCP/IP 傳送緩衝區大小 (以位元組為單位)。
RcvRcvBuffSize	接收端通道的接收端所使用的 TCP/IP 接收緩衝區大小 (以位元組為單位)。
SvrSndBuffSize	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	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)。
V 9.1.0 調整參數段落	

表 11: *qm.ini* 檔案的段落 (繼續)

段落及屬性	屬性的說明
SuppressDspAuthFail	如果連線缺少物件的 + dsp 權限，當授權檢查失敗時，佇列管理程式是否會暫停產生授權事件，並將 AMQ8077 錯誤訊息寫入錯誤日誌中。
ImplSyncOpenOutput	在對持續放置啟用隱含同步點之前，在同步點之外，已開啟佇列以供放置的應用程式數目下限。
V 9.1.2 UniformCluster 名稱	您用作統一叢集的 IBM MQ 叢集名稱。
V 9.1.0.9 OAMLdapConnect 逾時	LDAP 用戶端等待建立伺服器 TCP 連線的時間上限 (以秒為單位)。
V 9.1.0.9 OAMLdapQueryTimeLimit	LDAP 用戶端等待從伺服器接收 LDAP 要求回應的時間上限 (以秒為單位)。
V 9.1.0.15 OAMLdapResponseWarningTime	如果 LDAP 伺服器連線花費的時間超過 OAMLdapResponseWarningTime 參數指定的臨界值秒數，則會將 AMQ5544W 訊息寫入錯誤日誌中。
ExpiryInterval	指出佇列管理程式掃描佇列的頻率，以尋找其他佇列活動尚未清除的過期訊息。這是以秒為單位的時間間隔。
V 9.1.4 變數段落	
V 9.1.4 屬性 = 值	在 MQSC 定義期間用來作為插入項目的名稱及關聯值。
XAResourceManager 段落	
名稱	資源管理程式實例。
SwitchFile	包含資源管理程式的 XA 交換器結構之載入檔的完整名稱。
XAOpenString	要傳遞至資源管理程式 <code>xa_open</code> 進入點的資料字串。
XACloseString	要傳遞至資源管理程式 <code>xa_close</code> 進入點的資料字串。
控制執行緒	當佇列管理程式需要從其自己的多執行緒處理程序之一呼叫資源管理程式時，用於序列化的值。Windows 的必要項目。

附註:

1. 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。
2. 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。
3. 當您建立佇列管理程式時，IBM MQ 會自動寫入子儲存區段落，以及該段落內的屬性 ShortSubpool 名稱。IBM MQ 選擇 ShortSubpool 名稱的值。請勿變更此值。

Windows *qm.ini* 檔案的 AccessMode 段落

「存取模式」僅適用於 Windows 伺服器。*qm.ini* 檔案的 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

在使用 `-a [r]` 選項執行 `crtmqm` 指令之前，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。如果已啟動佇列管理程式作為服務，則 Windows 系統所檢查的使用者 ID 是使用準備 IBM MQ 精靈配置的使用者 ID。如果是以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。

範例段落

```
AccessMode:  
SecurityGroup=wmq\wmq
```

相關概念

[第 429 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 426 頁的『保護 Windows 上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 404 頁的『在 Windows 上的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 \(建立佇列管理程式\)](#)

ApiExit qm.ini 檔案的本端段落

若為伺服器，請使用 IBM MQ Explorer 中的 Exits 佇列管理程式內容頁面，或 `qm.ini` 檔案的 ApiExit 本端段落，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 ApiExit 本端段落，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 ApiExit 範本段落，如 [第 79 頁的『ApiExitCommon 及 ApiExit mqs.ini 檔案的範本段落』](#) 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。

如需使用這些屬性的相關資訊，請參閱 [配置 API 結束程式](#)。

Name=ApiExit_name

在 MQAXP 結構的 ExitInfo 名稱欄位中傳遞給它的 API 結束程式敘述性名稱。

此名稱必須是唯一的，長度不得超過 48 個字元，且只包含 IBM MQ 物件名稱的有效字元 (例如，佇列名稱)。

Function=function_name

包含 API 結束碼之模組的函數進入點名稱。此進入點是 MQ_INIT_EXIT 函數。

這個欄位的長度限制為 MQ_EXIT_NAME_LENGTH。

模組 = 模組名稱

包含 API 結束碼的模組。

如果這個欄位含有模組的完整路徑名稱，則會依其現狀使用。如果此欄位只包含模組名稱，則會使用 `qm.ini` 檔案的 `ExitPath` 段落中的 `ExitsDefaultPath` 屬性來尋找模組。

在支援個別執行緒程式庫的平台上，您必須提供 API 結束程式模組的非執行緒及執行緒版本。執行緒版本必須具有 `_r` 字尾。在載入之前，IBM MQ 應用程式 Stub 的執行緒版本會隱含地將 `_r` 附加至給定的模組名稱。

此欄位的長度限制為平台支援的路徑長度上限。

Data=data_name

要在 MQAXP 結構的 `ExitData` 欄位中傳遞至 API 結束程式的資料。

如果您包括此屬性，則會移除前導及尾端空白，將剩餘字串截斷為 32 個字元，並將結果傳遞至結束程式。如果您省略此屬性，則會將預設值 32 個空白傳遞給結束程式。

此欄位的長度上限為 32 個字元。

Sequence=sequence_number

相對於其他 API 結束程式，呼叫此 API 結束程式的順序。在具有較高序號的結束程式之前，會先呼叫具有較低序號的結束程式。結束程式的序號不需要連續。序列 1、2、3 與序列 7、42、1096 具有相同的結果。如果兩個結束程式具有相同的序號，則佇列管理程式會決定要先呼叫哪個結束程式。您可以在 MQAXP 中的 `ExitChainAreaPtr` 所指出的 `ExitChain` 區域中放置時間或標記，或寫入您自己的日誌檔，以識別事件之後呼叫的項目。

此屬性是不帶正負號的數值。

V 9.1.4

Multi

qm.ini 檔案的 AutoCluster 段落

當佇列管理程式開始識別叢集是否為自動叢集的成員，並可識別叢集的完整儲存庫時，會使用 `AutoCluster` 段落。

以下是 `AutoCluster` 段落的必要屬性：

Type =Uniform

指定自動叢集的類型，唯一有效的選項是 `統一`，代表統一叢集。

ClusterName=<String>

叢集的名稱，即自動叢集名稱。

下列屬性是 `AutoCluster` 段落的選用屬性，但您必須成對提供它們：

RepositoryName1 =<String>

這是自動叢集中第一個完整儲存庫的佇列管理程式名稱。這可以是此佇列管理程式的名稱，也可以是另一個佇列管理程式的名稱。

Repository1Conname=<Connection Name String>

這是自動叢集成員應該如何連接至此佇列管理程式的連線名稱 (CONNAME) 值。

Repository2Name=<String>

這是自動叢集中第二個完整儲存庫的佇列管理程式名稱。這可以是此佇列管理程式的名稱，也可以是另一個佇列管理程式的名稱。

Repository2Conname=<Connection name string>

這是自動叢集成員應該如何連接至此佇列管理程式的連線名稱 (CONNAME) 值。

範例段落

```
AutoCluster:
  Repository1Name=QM1
  Repository2Name=QM2
  Repository1Conname=127.0.0.1(1414)
  Repository2Conname=127.0.0.1(1415)
  ClusterName=UNIFORMCLUSTER1
  Type=Uniform
```

相關概念

[自動應用程式平衡](#)

相關工作

[建立新的統一叢集](#)

[使用自動叢集配置](#)

V 9.1.4

Multi

qm.ini 檔案的 AutoConfig 段落

在設定統一叢集時，經常會使用 AutoConfig 段落的屬性。

註：您只能對統一叢集使用 AutoCluster 段落。

MQSCConfig=<Path>

路徑是完整檔案路徑或目錄的路徑，在每次啟動佇列管理程式時，都會將所有 *.mqsc 檔案套用至佇列管理程式。

如需相關資訊，請參閱 [啟動時從 MQSC Script 自動配置](#)。

IniConfig=<Path>

該路徑是完整檔案路徑或目錄的路徑，其中所有 *.ini 檔案都會套用至每個佇列管理程式啟動時的 qm.ini 檔案。

如需相關資訊，請參閱 [第 85 頁的『啟動時自動配置 qm.ini』](#)。

範例段落

```
AutoConfig:
  MQSCConfig=/tmp/auto.mqsc
  IniConfig=/tmp/auto.ini
```

相關概念

[自動應用程式平衡](#)

相關工作

[建立新的統一叢集](#)

[使用自動叢集配置](#)

Multi

qm.ini 檔案的通道段落

「通道」段落的屬性會決定通道的配置。

z/OS

此資訊不適用於 IBM MQ for z/OS。

使用 qm.ini 檔案中的 CHANNELS 段落來指定通道的相關資訊。

Windows

Linux

或者，在 Linux (x86 及 x86-64) 和 Windows 上，請使用 IBM MQ Explorer Channels 佇列管理程式內容頁面。

MaxChannels= 100 (預設值) |number

容許的 現行 通道數上限。

預設值為 100。

必要的話，您可以將 **MaxChannels** 設為不同的值，以限制現行通道數目上限。對於 IBM MQ Appliance，預設值為 999 999 999 999，不應變更。

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 (預設值) |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

使用 MQCONNX FASTPATH 來連接通道; 沒有代理程式處理程序。

STANDARD

使用 STANDARD 連接通道。

PipeLineLength = 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, 請將通道兩端的佇列管理程式配置為 **PipeLineLength** 大於 1。

註: 管線化只對 TCP/IP 通道有效。

AdoptNewMCA= NO (預設值) | SVR | SDR | RCVR | CLUSRCVR | ALL | FASTPATH

如果 IBM 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. 如果前一個通道伺服器在 **AdoptNewMCATimeout** 等待間隔到期之前未回應此要求, 則會結束前一個通道伺服器的執行緒或處理程序。
3. 如果在步驟 2 之後未結束前一個通道伺服器, 且在 **AdoptNewMCATimeout** 等待間隔第二次到期之後, IBM MQ 會以 CHANNEL IN USE 錯誤結束通道。

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 (預設值) | 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 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。

V9.1.0 ChlauthEarlyAdopt = Y (預設值) | N

在決定 IBM MQ 用戶端應用程式連線的安全環境定義時，連線鑑別和通道鑑別規則的處理順序是一個重要因素。



小心：如果 **ChlauthEarlyAdopt** 不在 qm.ini 檔中，則預設值為 N，不過，從 IBM MQ 9.0.4 建立所有佇列管理程式時，都會將 **ChlauthEarlyAdopt=Y** 自動新增至 qm.ini 檔。

如果在佇列管理程式上的連線鑑別 AUTHINFO 物件上設定 ADOPTCTX (YES)，則 **ChlauthEarlyAdopt** 只會採用已提供給佇列管理程式進行連線鑑別的使用者 ID。

ChlauthEarlyAdopt 的有效值如下：

Y

在套用通道鑑別規則之前，通道會先使用佇列管理程式連線鑑別來驗證並採用應用程式所提供的使用者 ID 和密碼認證。在此作業模式中，通道鑑別規則會比對連線鑑別檢查所產生的使用者 ID。

N

通道會延遲應用程式所提供之使用者 ID 和密碼認證的連線鑑別驗證，直到套用通道鑑別規則為止。請注意，在此作業模式中，通道鑑別封鎖及對映規則無法考量使用者 ID 及密碼驗證的結果。

例如，預設鑑別資訊物件設為 **ADOPTCTX(YES)**，且使用者 fred 已登入。已配置下列兩個 CHLAUTH 規則：

```
SET CHLAUTH('MY.CHLAUTH') TYPE(ADDRESSMAP) DESCR('Block all access by
default') ADDRESS('*') USERSRC(NOACCESS) ACTION(REPLACE)
SET CHLAUTH('MY.CHLAUTH') TYPE(USERMAP) DESCR('Allow user bob and force
CONNAUTH') CLNTUSER('bob') CHCKCLNT(REQUIRED) USERSRC(CHANNEL)
```

發出下列指令，目的是將指令鑑別為使用者 bob 採用的安全環境定義：

```
runmqsc -c -u bob QMGR
```

事實上，佇列管理程式會使用 fred 而非 bob 的安全環境定義，且連線會失敗。

若要使用 bob 的安全環境定義，**ChlauthEarlyAdopt** 必須設為 Y。

PasswordProtection = compatible|always|選用|warn

從 IBM MQ 8.0，在 MQCSP 結構中設定受保護的密碼，而不是使用 TLS。

MQCSP 密碼保護適用於測試及開發目的，因為使用 MQCSP 密碼保護比設定 TLS 加密更簡單，但沒有那麼安全。

如需相關資訊，請參閱 [MQCSP 密碼保護](#)。

V 9.1.0 IgnoreSeqNumberMismatch = NO (預設值) | YES

通道兩端的「訊息通道代理程式 (MCA)」會保留透過通道傳送的訊息數目計數，以維護同步化。例如，如果刪除位於一端的通道定義，然後重建，則可能會失去同步化。在這些情況下，可能需要 RESET CHANNEL 來確認同步化資料已遺失，並允許通道繼續啟動。

IgnoreSeqNumberMismatch 屬性必須設定在接收端佇列管理程式上。

實際上，這個屬性會在接收端通道上執行重設通道指令。

此屬性使用下列值來控制佇列管理程式在通道啟動期間如何處理序號不符：

NO

通道重新同步期間會檢查通道序號，如果兩個 MCA 不同意相同的序號，則會報告錯誤訊息 AMQ9526，且通道將無法啟動。

YES

通道重新同步期間會檢查通道序號，但如果兩個 MCA 不同意相同的序號，則會報告警告訊息 AMQ9703，且通道啟動會繼續進行。在正常情況下，不應該需要此屬性值。當知道同步化資料已遺失時 (例如在災難回復期間)，此選項可避免需要手動確認每一個序號不符。指定此值的效果與管理者自動發出 **RESET CHANNEL** 以回應每個序號不符的結果類似。

V 9.1.5 V 9.1.0.5 ChlauthIgnoreUserCase = N (預設值) | Y

啟用佇列管理程式，使 CHLAUTH 規則內的使用者名稱比對不區分大小寫。此選項容許：

- CHLAUTH TYPE (USERMAP) 規則中要不區分大小寫相符的 CLNTUSER
- CHLAUTH TYPE (BLOCKUSER) 規則中要不區分大小寫相符的 USERLIST

ChlauthIgnoreUserCase 的有效值如下：

N

通道鑑別規則嘗試比對區分大小寫的用戶端使用者識別，例如指定 CLNTUSER ('Fred') 的規則將不符合 'fred' 或 'FRED'，它只會符合使用者 ID 'Fred'。這是預設值。

Y

通道鑑別規則嘗試以不區分大小寫的方式比對用戶端使用者識別，例如指定 CLNTUSER ('Fred') 的 TYPE (USERMAP) 或 TYPE (USERBLOCK) 的通道鑑別規則將符合任何大小寫變化，例如使用者 ID 'Fred'、'FRED' 及 'fred' 全部相符。

請注意，在比對通道鑑別規則時忽略使用者 ID 的大小寫時，可以有多個規則相符。如果發生此情況，則未定義符合的規則。例如，使用下列規則，如果使用者 'fred' 透過 CLIENT 通道連接至佇列管理程式，則他們可能對映至 'mquser1' 或 'mquser2'：

```
SET CHLAUTH('CLIENT') TYPE(USERMAP) CLNTUSER('fred') USERSRC(MAP) MCAUSER('mquser1')
SET CHLAUTH('CLIENT') TYPE(USERMAP) CLNTUSER('FRED') USERSRC(MAP) MCAUSER('mquser2')
```

若要在使用 ChlauthIgnoreUserCase=Y 時避免任何不確定性，請避免定義 CHLAUTH 規則將重疊，並在使用不區分大小寫相符項時導致不同的行為。

ChlauthIssue 警告 = y

如果您想要在 **SET CHLAUTH** 指令上設定 [WARN = YES](#) 屬性時產生訊息 AMQ9787，請設定此屬性。

範例段落

```
Channels:
  MaxChannels=200
  MaxActiveChannels=100
  MQIBindType=STANDARD
  PipelineLength=2
```

相關概念

第 174 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。

Multi **qm.ini** 檔案的連線段落

連線段落定義預設連結類型。

使用 `qm.ini` 檔案中的連線段落來指定預設連結類型。

Windows **Linux** 或者，在 Linux (x86 及 x86-64) 和 Windows 上，請使用 IBM MQ Explorer Extended 佇列管理程式內容頁面。

註：如果您需要連線段落，則必須建立連線段落。

DefaultBindType = SHARED (預設值) | 隔離

如果 **DefaultBindType** 設為 隔離，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 **DefaultBindType** 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，但部分資源會在它們之間共用。

預設值為 SHARED。



小心: DefaultBindType 適用於所有 MQCONN 呼叫，以及使用 MQCONNX 搭配 `MQCNO_STANDARD_BINDING` 的任何 MQCONNX。

變更 **DefaultBindType** 可能會導致部分應用程式效能降低。

範例段落

```
Connection:  
DefaultBindType=SHARED
```

診斷訊息記載

IBM MQ 的診斷訊息日誌是一種機制，可讓 IBM MQ 系統的各種元件報告與 IBM MQ 配置及執行時期狀態變更和問題相關的診斷訊息。

這些日誌有時稱為 IBM MQ 錯誤日誌，但一律包含 IBM MQ 資訊及警告訊息，以及錯誤訊息。向這些日誌報告的 IBM MQ 的三個主要元件如下：

- 佇列管理程式
- IBM MQ 用戶端
- IBM MQ 系統的其餘部分

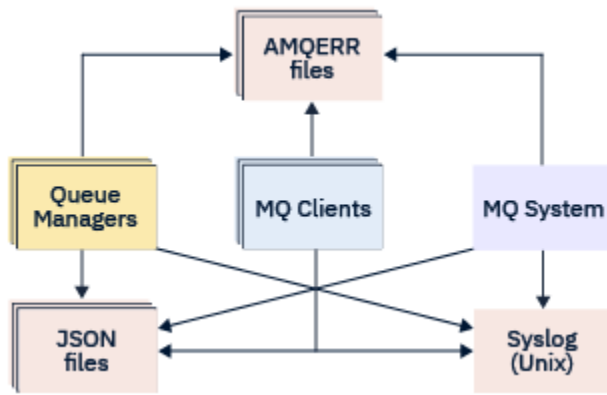
IBM MQ 支援透過許多不同的方法 (稱為 診斷訊息服務) 來報告診斷訊息，容許自訂方法來記錄及耗用此資訊：

- AMQERRnn 日誌檔
- JSON 格式化日誌檔
- **UNIX** JSON 格式的 Syslog

IBM MQ 的 JSON 輸出會格式化為單行 JSON 物件，因此 JSON 日誌或 Syslog 記錄的每一個別行都代表有效的 JSON 物件。整個日誌不會封裝成單一 JSON 物件。

下圖顯示佇列管理程式、IBM MQ 用戶端及 IBM MQ 系統可以所有使用說明的方法來報告診斷訊息。

圖 5: IBM MQ 的不同部分如何報告診斷訊息



如何配置 IBM MQ 診斷日誌:

診斷日誌是使用 `qm.ini` 檔案內的段落來定義及自訂，這些段落是特定於需要它們的 IBM MQ 元件。每一個唯一的記載端點都會定義在 `ini` 檔案內其自己的段落標題下，以及其中所定義的任何自訂作業。自訂作業可以包括:

- 發生輪替之前要覆蓋的日誌檔大小; 不適用於 Syslog
- 基於日誌訊息嚴重性的任何過濾，以及
- 要暫停的任何特定訊息碼。

IBM MQ 可以配置為寫入三種類型的記載端點中的任何一種或全部，以容許特定日誌段落履行特定角色。同樣地，可以定義多個檔案服務。例如:

- JSON 格式可讓您透過本端及雲端環境中的自動化工具更容易剖析。
- Syslog 輸出可讓 IBM MQ 元件將診斷資訊整合至與系統上其他產品一致的一般 OS 記載位置。
- 根據嚴重性過濾的日誌端點，容許記錄特定日誌檔，例如，系統中只有嚴重錯誤。

不論配置的診斷記載樣式為何，除了使用任何其他記載配置之外，也一律會寫入保留在 IBM MQ 系統日誌目錄 (`/var/mqm/errors/AMQERRnn.log`) 及特定佇列管理程式日誌目錄 (`/var/mqm/qmgrs/<qmgr_name>/errors/AMQERRnn.log`) 下的傳統診斷檔案。

僅對於佇列管理程式，可以透過指定 [第 102 頁的『診斷訊息服務段落』](#) 的屬性來執行這些必要日誌的選用配置。

不同的段落區域

其他段落可以套用至 IBM MQ 的不同區域。

佇列管理程式 (`qm.ini`)

適用於佇列管理程式所產生的日誌訊息

系統 (`mqs.ini`)

適用於系統所產生的日誌訊息。此選項不是佇列管理程式特有的，除非佇列管理程式無法存取或寫入其自己的日誌。

範本 (`mqs.ini`)

作為範本的一或多個段落，會在建立佇列管理程式時複製到 `qm.ini`。

用戶端 (`mqclient.ini`)

適用於用戶端作業，例如適用於遠端佇列管理程式的用戶端模式中的 `runmqsc`。

在 JSON 格式化日誌與傳統格式化日誌之間轉換

已加強 `mqrc` 指令，以容許在 JSON 與傳統格式化日誌之間以及不同語言之間進行一些轉換。

相關參考

[第 102 頁的『診斷訊息服務段落』](#)

可用的診斷訊息服務選項會啟用 IBM MQ 診斷記載的自訂作業，因此日誌輸出可以從 IBM MQ 的不同元件導向至不同的日誌端點。

第 101 頁的『QMErrorLog 段落』

您可以使用 `qm.ini` 檔案中的佇列管理程式錯誤日誌段落 `QMErrorLog`，來修改 IBM MQ 錯誤日誌的作業及內容。

第 104 頁的『診斷訊息服務』

可以定義下列診斷訊息服務及其服務特定屬性，這些屬性指定在配置檔的 `DiagnosticSystem` 訊息、`DiagnosticMessages` 及 `DiagnosticMessages` 範本段落中：

Multi **QMErrorLog 段落**

您可以使用 `qm.ini` 檔案中的佇列管理程式錯誤日誌段落 `QMErrorLog`，來修改 IBM MQ 錯誤日誌的作業及內容。

`QMErrorLog` 服務是傳統的 IBM MQ 診斷記載服務，用來輸出與佇列管理程式相關的診斷訊息。`QMErrorLog` 服務會持續執行，無法關閉，但可以自訂到某種程度。

您可以使用 `qm.ini` 檔案中的 `QMErrorLog` 段落，以排除某些訊息寫入佇列管理程式錯誤日誌。您也可以給定的時間間隔內，暫停將訊息寫入錯誤日誌。

Windows **Linux** 或者，您可以使用 IBM MQ Explorer 中的 [延伸佇列管理程式內容頁面](#) 來排除及抑制具有 **已排除訊息**、**已抑制訊息** 及 **已抑制訊息間隔** 屬性的訊息，而不直接編輯 `qm.ini` 檔案。



小心：

- **Windows** 只有在 Windows 平台上使用本端佇列管理程式時，才能使用 IBM MQ Explorer 來進行變更。
- `QMErrorLog` 段落不適用於 IBM MQ 系統配置檔 `mqs.ini` 或用戶端配置檔 (通常稱為 `mclient.ini`)。

`QMErrorLog` 段落中可以包含下列屬性：

ErrorLog 大小 = *maxsize*

指定要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 33554432 位元組 (32 MB)。

必要的話，您可以使用這個屬性，將大小上限縮減回先前的上限 2 MB。

您可以使用 **MQMAXERRORLOGSIZE** 環境變數來設定日誌的大小。

ExcludeMessage= *msgIds*

指定不要寫入佇列管理程式錯誤日誌的訊息。

如需相關資訊，請參閱第 102 頁的『診斷訊息服務段落』中的 [ExcludeMessage](#)。

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。如果同時在 `SuppressMessage` 和 `ExcludeMessage` 中指定相同的訊息 ID，則會排除該訊息。

此選項不適用於 `mclient.ini` 中定義的診斷訊息服務。如需相關資訊，請參閱第 102 頁的『診斷訊息服務段落』中的 [SuppressMessage](#)。

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。

範例段落

```
QMErrorLog:
  ErrorLogSize=262144
  ExcludeMessage=7234
  SuppressMessage=9001,9002,9202
  SuppressInterval=30
```

相關概念

第 84 頁的『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。

相關參考

第 102 頁的『診斷訊息服務段落』

可用的診斷訊息服務選項會啟用 IBM MQ 診斷記載的自訂作業，因此日誌輸出可以從 IBM MQ 的不同元件導向至不同的日誌端點。

Multi 診斷訊息服務段落

可用的診斷訊息服務選項會啟用 IBM MQ 診斷記載的自訂作業，因此日誌輸出可以從 IBM MQ 的不同元件導向至不同的日誌端點。

您可以使用具有下列其中一個名稱的段落來啟用其他診斷訊息服務：

• DiagnosticSystemMessages

定義當產生進入系統錯誤日誌的診斷訊息時所使用的服務。在 mqs.ini 或 mqclient.ini 檔案中有效。

用戶端應用程式在 mqclient.ini 檔及 mqs.ini 中使用 **DiagnosticSystemMessages** 段落，**DiagnosticSystemMessages** 段落控制沒有佇列管理程式環境定義之伺服器應用程式的訊息。

您可以配置佇列管理程式及應用程式，以額外將所有訊息寫入 syslog 服務。

• DiagnosticMessages

定義當產生進入佇列管理程式錯誤日誌的診斷訊息時所使用的服務。僅在 qm.ini 檔案中有效。

• DiagnosticMessagesTemplate

建立佇列管理程式時，從 mqs.ini 檔案複製到 qm.ini 檔案中的 **DiagnosticMessages** 的段落。

若要顯示診斷訊息，請使用 [mqrc](#) 指令。

段落的屬性



小心： 服務及段落的名稱是必要的。

name=<stanzaname>

段落的名稱。此值在 ini 檔案中必須是唯一的。

服務 = 服務類型

此屬性定義由此段落啟用的服務 (其中服務名稱不區分大小寫)。

例如，若要啟用 syslog 作為其他服務，請輸入下列指令：

```
Service=syslog
```

請參閱第 104 頁的『[診斷訊息服務](#)』及其可與診斷訊息服務段落搭配使用的特定屬性。

您可以將下列選用屬性新增至段落：

- [ExcludeMessage](#)
- [SuppressMessage](#)
- [SuppressInterval](#)
- 第 104 頁的『[嚴重性](#)』

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 IBM 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 主控台及印刷本日誌。IBM 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
IBM i 7163 - 工作啟動訊息 (僅限 IBM i)
7234 - 載入的訊息數
8245-實體沒有足夠的權限來顯示物件
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
9545-斷線間隔過期
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 IBM 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 主控台及印副本日誌。IBM 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，因此抑制訊息會阻止您在需要時接收一些重複訊息。時間間隔由 SuppressInterval 指定。msgIds 包含下列以逗點區隔的訊息 ID 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
IBM i 7163 - 工作啟動訊息 (僅限 IBM i)
7234 - 載入的訊息數
8245-實體沒有足夠的權限來顯示物件
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
9545-斷線間隔過期
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

如果同時在 SuppressMessage 和 ExcludeMessage 中指定相同的訊息 ID，則會排除該訊息。
此選項不適用於 MQ client.ini 中定義的診斷訊息服務。

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 必須在範圍 1-86400 秒內。如果未指定 SuppressInterval，則會使用預設值 30 秒。

嚴重性

以逗點區隔的嚴重性層次清單，其中嚴重性層次的名稱不區分大小寫。容許值為：

- I (或資訊或 0)
- W (或警告或 10)
- E (或錯誤或 20 和 30)
- S (或「停止」或 40)
- T (或系統或 50)

附註：

1. 預設值為 all
2. 只有所選嚴重性層次中的訊息才會呈現給服務。

或者，您可以使用加號字元 (+)，以顯示指定的錯誤層次及所有更高層次。例如，若要顯示所有錯誤：

```
Severities=E+
```

相關參考

第 101 頁的『QLErrorLog 段落』

您可以使用 qm.ini 檔案中的佇列管理程式錯誤日誌段落 QLErrorLog，來修改 IBM MQ 錯誤日誌的作業及內容。

第 104 頁的『診斷訊息服務』

可以定義下列診斷訊息服務及其服務特定屬性，這些屬性指定在配置檔的 DiagnosticSystem 訊息、DiagnosticMessages 及 DiagnosticMessages 範本段落中：

Multi 診斷訊息服務

可以定義下列診斷訊息服務及其服務特定屬性，這些屬性指定在配置檔的 DiagnosticSystem 訊息、DiagnosticMessages 及 DiagnosticMessages 範本段落中：

已定義下列診斷訊息服務：

檔案

此服務會以類似於 QLErrorLog 服務的方式，將任何未過濾的訊息傳送至檔案。視指定的 **Format** 而定，會使用現有的文字格式或指定的 JSON 格式。依預設，有三個檔案稱為 AMQERR01.LOG、AMQERR02.LOG 及 AMQERR03.LOG 或 AMQERR01.json、AMQERR02.json 及 AMQERR03.json，視 **Format** 內容而定，並根據所配置的大小來確定這些輪替。


下列屬性僅在「檔案」段落中受支援：

FilePath

寫入日誌檔的路徑。預設值是與 AMQERR01.log 檔案相同的位置，即系統或佇列管理程式。路徑必須是絕對路徑，但可以包含可更換的插入項目。例如：

+ MQ_Q_MGR_DATA_PATH +

佇列管理程式診斷訊息目錄母項的完整路徑。預設值為：

-  在 UNIX and Linux 平台上: /var/mqm/qmgrs/<QM_name>

- **Windows** 在 Windows 上, C:\Program Data\IBM\MQ\qmgrs\

+ MQ_DATA_PATH +

系統診斷訊息目錄母項的完整路徑。預設值為:

- **UNIX** 在 UNIX and Linux 平台上:/var/mqm
- **Windows** 在 Windows 上: C:\Program Data\IBM\MQ

如果沒有使用現有的錯誤目錄, 您必須以適當的許可權來建立這個路徑。

FilePrefix

日誌檔的字首。預設值為 AMQERR。

FileSize

日誌輪替的大小。預設值為 32MB, 與 [第 101 頁的『QLErrorLog 段落』](#) 的 **ErrorLogSize** 內容相同, 語意相同。

註: ErrorLogSize 內容僅適用於預設錯誤日誌服務, 不適用於自訂診斷服務。

您可以使用 **MQMAXERRORLOGSIZE** 環境變數來設定日誌的大小。

Format

檔案的格式。該值可以是 *text* (適用於其他 QLErrorLog 樣式服務) 或 *json* (預設值)。

根據此屬性的設定, 檔案的字尾是 .LOG 或 .json。

例如, 編輯佇列管理程式的 *qm.ini* 檔, 並新增下列段落:

```
DiagnosticMessages:
  Service = File
  Name = JSONLogs
  Format = json
  FilePrefix = AMQERR
```

重新啟動之後, 佇列管理程式會在其 **ERRORS** 目錄中有 **AMQERR0x.json** 檔。

您可以定義多個「檔案」服務。這容許配置, 如下列範例所示, 其中不同標籤的訊息會分割成不同的日誌集:

```
DiagnosticMessages:
  Name=ErrorsToFile
  Service=File
  Severities=E+
  FilePrefix=OnlyErrors

DiagnosticMessages:
  Name=NonErrorstoFile
  Service=File
  Severities=1 W
  FilePrefix=Information
```

Linux **UNIX** **Syslog**

Syslog 服務在 Windows 或 IBM i 上無法使用

您只能定義一個 Syslog 服務, 且 Syslog 服務會使用 [JSON 格式 診斷訊息規格](#), 將任何未過濾的訊息傳送至 syslog。該資訊會以表格中顯示的順序新增至 syslog, 從 msgID 開始並插入。

訊息的嚴重性會以下列方式對映至 syslog 層次:

表 12: 將訊息嚴重性對映至 syslog 層次	
嚴重性	層次
0	LOG_INFO
10	LOG_WARNING
20	LOG_ERR

表 12: 將訊息嚴重性對映至 syslog 層次 (繼續)	
嚴重性	層次
30	LOG_ERR
40	日誌警示
50	日誌警示

下列屬性僅在 syslog 段落中受支援:

Ident

定義與 syslog 項目相關聯的 **ident** 值。預設值為 *ibm-mq*。

下列範例顯示傳送至 Syslog 的錯誤訊息:

```
DiagnosticMessages:
  Name=ErrorsToSyslog
  Ident=mq
  Service=Syslog
  Severities=E+
```

如需通用段落屬性的相關資訊，請參閱 [第 102 頁的『診斷訊息服務段落』](#)。

附註:

1. 僅針對「檔案」服務，您可以有多個段落，每一個都有不同的名稱。只有在序列中使用最終名稱的定義才會生效。
2. 只有在重新啟動佇列管理程式時，段落值的變更才會生效。

Multi qm.ini 檔案的 ExitPath 段落

ExitPath 段落指定佇列管理程式系統上使用者結束程式的路徑。

使用 qm.ini 檔案中的 ExitPath 段落，以指定佇列管理程式系統上使用者結束程式的路徑。

Windows **Linux** 或者，在 Linux (x86 及 x86-64) 和 Windows 上，請使用 IBM MQ Explorer Exits 佇列管理程式內容頁面。

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

範例段落

```
ExitPath:
  ExitsDefaultPath=/var/mqm/exits
  ExitsDefaultPath64=/var/mqm/exits64
```

ExitProperties qm.ini 檔案的本端段落

ExitProperties 本端段落指定佇列管理程式上結束程式內容的相關資訊。

使用 qm.ini 檔案中的 ExitPropertiesLocal 段落，以指定佇列管理程式上結束程式內容的相關資訊。

Windows

Linux

或者，在 Linux (x86 和 x86-64) 及 Windows 上，使用 IBM MQ Explorer 叢集佇列管理程式內容頁。

Windows

或者，在 Windows 上，您可以使用 **amqmdain** 指令來指定此資訊。

依預設，此設定繼承自機器層面配置的 ExitProperties 段落中的 **CLWLMode** 屬性 (如第 80 頁的『mqs.ini 檔案的 ExitProperties 段落』中所述)。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。

使用 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。

Windows

Linux

或者，在 Linux (x86 和 x86-64) 和 Windows 上，請使用 IBM MQ Explorer Extended IBM MQ 內容頁面。

CLWLMode= SAFE (預設值) | FAST

叢集工作量 (CLWL) 結束程式可讓您指定叢集中要開放的叢集佇列，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。如果省略 **CLWLMode** 屬性，則叢集工作量結束程式會以 SAFE 模式執行。

安全

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。如果您確信 CLWL 結束程式沒有問題，且您特別關心效能，則只應該在 FAST 模式下執行 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。

範例段落

```
ExitPropertiesLocal:
  CLWLMode=SAFE
```

IBM i

Linux

UNIX

qm.ini 檔案的檔案系統段落

「檔案系統」段落指定在佇列管理程式錯誤日誌上設定的許可權是否保持不變或變更回其預設值。

在大部分情況下，預期在錯誤日誌檔上設定的預設許可權會很有用，因此大部分 IBM MQ 管理者都不需要變更它們。

不過，您的 IBM MQ 管理者可能想要變更其錯誤日誌檔的許可權，在此情況下，他們應該設定 Filesystem 段落選項 **ValidateAuth=No**，這會導致佇列管理程式保留之後的許可權不變。

預設行為 (沒有 `ValidateAuth=No`) 是佇列管理程式會檢查佇列管理程式錯誤日誌的檔案許可權，並將它們變更回其預設值。此檢查可以隨時發生，包括在佇列管理程式結束或啟動作業期間。

範例段落

```
Filesystem:  
  ValidateAuth=No
```

Multi **qm.ini** 檔案的日誌段落

「日誌」段落指定在佇列管理程式上記載的相關資訊。

使用 `qm.ini` 檔案中的「日誌」段落，以指定記載在佇列管理程式上的相關資訊。

Windows 或者，在 Linux (x86 及 x86-64) 和 Windows 上，請使用 IBM MQ Explorer Log 佇列管理程式內容頁面。

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 81 頁的『`mqs.ini` 檔案的 `LogDefaults` 段落』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 507 頁的『計算日誌的大小』。

註: 下列參數清單中給定的限制由 IBM MQ 設定。作業系統限制可能會減少可能的日誌大小上限。

LogPrimary 檔案 = 3 (預設值) | 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。

主要及次要日誌檔總數在 Windows 上不得超過 255，在 UNIX and Linux 系統上不得超過 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 (預設值) | 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，而在 UNIX and Linux 系統上，上限為 253。預設數字為 2。

主要及次要日誌檔總數在 Windows 上不得超過 255，在 UNIX and Linux 系統上不得超過 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，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 (預設值) | LINEAR

佇列管理程式要使用的記載類型。預設值為 CIRCULAR。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 81 頁的『`mqs.ini` 檔案的 `LogDefaults` 段落』中 **LogType** 屬性的說明。

循環式

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 503 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 503 頁的『記載類型』](#)。

註: 透過在 `qm.ini` 檔案中修改此屬性，無法變更佇列管理程式的 **LogType**。若要變更佇列管理程式的 **LogType**，您必須使用 **migmqlog** 指令。

LogBufferPages = 0 (預設值) | 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。




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：

-  Windows 中的 `C:\ProgramData\IBM\MQ\log`。
-   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 IBM MQ for UNIX 及 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 (預設值)

日誌程式用來可靠地寫入日誌記錄的方法。

TripleWrite (預設值)

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

只有在管理 IBM MQ 回復日誌的檔案系統及裝置明確保證 4KB 寫入的最小單位性時，您才應該使用 **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。

註: 如果持續性工作量中有足夠的並行性，則除了預設值 **TripleWrite** 之外，設定任何其他值的潛在好處都是最小的。

如需相關資訊，請參閱 [第 110 頁的『LogWriteIntegrity -使用 SingleWrite 或 TripleWrite』](#)。

LogManagement= 手動 (預設值) | 自動 | 保存

用於手動或由佇列管理程式管理日誌範圍的方法。預設值為 **手動**。

僅當 **LogType** 為 **LINEAR** 時，此屬性才適用。

如果您變更 **LogManagement** 值，則在重新啟動佇列管理程式之前，變更不會生效。

如果找到無法辨識的屬性值，則在更正該值之前不會啟動佇列管理程式。

LogManagement 內容在 IBM i 上無效。

手動 (預設值)

您可以手動管理日誌範圍。指定此選項表示佇列管理程式不會重複使用或刪除日誌範圍，即使回復不再需要日誌範圍亦是如此。

自動

日誌範圍由佇列管理程式自動管理。指定此選項表示佇列管理程式能夠重複使用日誌範圍，或在回復不再需要日誌範圍時將其刪除。不會考慮保存。

保存

日誌範圍由佇列管理程式管理，但您必須在每一個日誌範圍的保存完成時通知佇列管理程式。

指定此選項表示佇列管理程式可以任意重複使用或刪除日誌範圍（只要佇列管理程式收到回復不再需要的日誌範圍已保存的通知）。

您可以使用 **RESET QMGR MQSC** 指令或 **Reset Queue Manager PCF** 指令來執行此通知。

範例段落

```
Log:
LogPrimaryFiles=3
LogSecondaryFiles=2
LogFilePages=4096
LogType=CIRCULAR
LogBufferPages=0
LogPath=/var/mqm/log/saturn!queue!manager/
```

註: `LogBufferPages` 的零值會提供 512 值。

LogWriteIntegrity -使用 SingleWrite 或 TripleWrite

LogWriteIntegrity 選項的設定位於 `qm.ini` 檔案的 Log 段落中，會決定日誌程式在 IBM MQ 中用來將日誌記錄寫出至回復日誌的演算法。預設值是 *TripleWrite*，在幾乎每一種可能的情況下，此設定都是安全的。

只有在要寫入局部日誌頁面時，**LogWriteIntegrity** 的設定才會有任何作用。對於具有合理並行活動數量的佇列管理程式，此實務範例很少發生。

SingleWrite

SingleWrite 選取一個演算法，在非常特殊的情況下，它的執行效能會優於預設 *TripleWrite* 設定。*SingleWrite* 設定是安全的，只有在所有情況下基礎儲存體平台都可以絕對保證同步寫入 MQ 回復日誌的 4KB 頁面會自動寫入。

只有在管理 IBM MQ 回復日誌的檔案系統或裝置明確保證最小寫入 4KB 時，您才應該使用 *SingleWrite* 設定。也就是說，當寫入 4KB 頁面因任何原因而失敗時，只有兩個可能的狀態應該是前像或後像，且不可能有中間狀態。在所有其他情況下，您應該使用 *TripleWrite*。

在具有足夠並行性的系統上，佇列管理程式只會寫入日誌資料的完整頁面，如果達到完整頁面的高百分比，則 *SingleWrite* 與 *TripleWrite* 之間沒有顯著的效能差異。

在並行性很少的系統上，*SingleWrite* 具有顯著的效能優勢，不過偏好的解決方案通常是增加並行性，而不是使用 *SingleWrite*。

請注意，很難可靠地判定 4KB 寫入的原子性，基礎軟體或硬體的變更可能會使任何此類保證失效。

如果您不確定您的儲存空間基礎架構現在是否提供必要保證，而且在未來的任何情況下，您都應該使用 *TripleWrite*。

Windows qm.ini 檔的 LU62 段落 (僅限 Windows)

LU62 段落指定 SNA LU 6.2 通訊協定配置參數。這些參數會置換通道的預設屬性。

使用 `qm.ini` 檔案中的 LU62 段落來指定 SNA LU 6.2 通訊協定配置參數。它們會置換通道的預設屬性。

Windows **Linux** 或者，在 Linux (x86 及 x86-64) 和 Windows 上，請使用 IBM MQ Explorer LU6.2 佇列管理程式內容頁面。

TPName

要在遠端站台上啟動的 TP 名稱。

Library1= DLL 名稱 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。

Windows **qm.ini 檔案的 NETBIOS 段落 (僅限 Windows)**

`qm.ini` 檔案中的 NETBIOS 段落指定 NetBIOS 通訊協定配置參數。這些參數會置換通道的預設屬性。

使用 `qm.ini` 檔案中的 NETBIOS 段落，以指定 NetBIOS 通訊協定配置參數。它們會置換通道的預設屬性。

Windows **Linux** 或者，在 Linux (x86 和 x86-64) 及 Windows 上，請使用 IBM MQ Explorer Netbios 佇列管理程式內容頁。

LocalName= name

LAN 上用來識別此機器的名稱。

AdapterNum= 0 (預設值) | adapter_number

LAN 配接卡的號碼。預設值為配接卡 0。

NumSess= 1 (預設值) | number_of_sessions

要配置的階段作業數目。預設值是 1。

NumCmds= 1 (預設值) | number_of_commands

要配置的指令數。預設值是 1。

NumNames= 1 (預設值) | 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。

Linux **UNIX** **qm.ini 檔案的 RestrictedMode 段落**

RestrictedMode 段落指定群組的名稱，該群組包含允許執行 MQI 應用程式、更新所有 IPCC 資源，以及變更某些佇列管理程式目錄的內容的成員。此段落僅適用於 UNIX and Linux 系統。

RestrictedMode 段落由 `crtmqm` 指令上的 **-g** 選項設定。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 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 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 MQCONNX 呼叫都會失敗，原因碼為 MQRC_Q_MGR_NOT_AVAILABLE。

重要：在許多作業系統上，若要將使用者新增至要辨識的群組，有問題的使用者必須登出再重新登入。

受限模式與 IBM MQ 授權服務一起運作。因此，您也必須授與使用者連接至 IBM MQ 的權限，並使用 IBM MQ 授權服務來存取他們需要的資源。

 如需配置 IBM MQ 授權服務的進一步資訊，請參閱 [在 UNIX, Linux, and Windows 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 MQ 受限模式。

相關參考

[crtmqm](#) (建立佇列管理程式)

mqm.ini 檔案的安全段落

「安全」段落指定「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 並使「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。

範例段落

```
Security:
  ClusterQueueAccessControl=Xmitq
  GroupModel=GlobalGroups
```

Multi **qm.ini 檔案的服務段落**

「服務」段落是用來變更可安裝的服務。此段落包含服務的名稱，以及為服務定義的進入點數目。

註: **Windows** **Linux** 變更可安裝服務及其元件有重大意義。因此，可安裝的服務在 IBM MQ Explorer 中是唯讀的。

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。為此使用 [ServiceComponent](#) 段落。

Service 和 **ServiceComponent** 段落可以任意順序出現，其下的段落索引鍵也可以任意順序出現。對於其中任一段落，所有段落索引鍵都必須存在。如果段落索引鍵重複，則會使用最後一個段落索引鍵。

在啟動時，佇列管理程式會依序處理配置檔中的每一個服務元件項目。然後它會載入指定的元件模組，呼叫元件的進入點 (必須是元件起始設定的進入點)，並傳遞配置控點給它。

名稱 = **AuthorizationService** (預設值) | **NameService**

必要服務的名稱。

AuthorizationService

對於 IBM 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。

Linux **AIX** 佇列管理程式配置檔中的下列段落在 IBM MQ for AIX 上定義兩個授權服務元件。**MQ_INSTALLATION_PATH** 代表 IBM MQ 安裝所在的高階目錄。

```
Service:
  Name=AuthorizationService
  EntryPoints=13

ServiceComponent:
  Service=AuthorizationService
  Name=MQSeries.UNIX.auth.service
  Module= MQ_INSTALLATION_PATH/lib/amqzfu
  ComponentDataSize=0

ServiceComponent:
  Service=AuthorizationService
  Name=user.defined.authorization.service
  Module=/usr/bin/udas01
  ComponentDataSize=96
```

圖 6: UNIX and Linux qm.ini

Linux **AIX** 服務元件段落 (**MQSeries.UNIX.auth.service**) 定義預設授權服務元件 OAM。如果您移除此段落並重新啟動佇列管理程式，則會停用 OAM，且不會進行授權檢查。

Windows 您也可以使用 IBM MQ 服務來新增 **SecurityPolicy** 屬性。只有在 **Service** 段落中指定的服務是授權服務 (即預設 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。可能的值為：

Default

如果您要讓預設安全原則生效，請指定 **Default**。如果未針對特定使用者 ID 將 Windows 安全 ID (NT SID) 傳遞至 OAM，則會嘗試透過搜尋相關安全資料庫來取得適當的 SID。

NTSIDsRequired

執行安全檢查時，需要將 NT SID 傳遞至 OAM。

Windows 服務元件段落 MQSeries.WindowsNT.auth.service 定義預設授權服務元件 OAM。如果您移除此段落並重新啟動佇列管理程式，則會停用 OAM，且不會進行授權檢查。

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。

Linux **AIX** 下列名稱服務的 UNIX and Linux 配置檔段落範例指定 (虛構) ABC 公司所提供的名稱服務元件。

```
# Stanza for name service
Service:
  Name=NameService
  EntryPoints=5

# Stanza for name service component, provided by ABC
ServiceComponent:
  Service=NameService
  Name=ABC.Name.Service
  Module=/usr/lib/abcname
  ComponentDataSize=1024
```

圖 7: qm.ini 中的名稱服務段落 (適用於 UNIX and Linux 系統)

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。

Windows SecurityPolicy= 預設值 |NTSIDsRequired

在 Windows 系統上，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。

可能的值為：

預設值

請使用預設安全原則來生效。如果未針對特定使用者 ID 將 Windows 安全 ID (NT SID) 傳遞至 OAM，則會嘗試透過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 \(SID\)](#)。

另請參閱 [配置授權服務段落: Windows 系統](#)。

Linux **UNIX** SecurityPolicy= user|group|default

在 UNIX and Linux 系統上，此值指定佇列管理程式是使用使用者型或群組型授權。值不區分大小寫。如果您不包含 **SecurityPolicy** 屬性，則會使用 default，它會使用群組型授權。

重新啟動佇列管理程式，讓變更生效。另請參閱 [配置授權服務段落: Windows 系統](#)。

SharedBindingsUserId= 使用者類型

只有在指定的服務是預設授權服務 (即 OAM) 時，**SharedBindingsUserId** 屬性才適用。

SharedBindingsUserId 屬性只與共用連結的關係搭配使用。此值可讓您指定 *IdentityContext* 結構中的 *UserIdentifier* 欄位 (來自 MQZ_AUTHENTICATE_USER 函數) 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](#)。

可能的值為：

預設值

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 欄位的值會設為有效使用者 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 屬性只能搭配快速路徑連結使用。此值可讓您指定 *IdentityContext* 結構中的 *UserIdentifier* 欄位 (來自 MQZ_AUTHENTICATE_USER 函數) 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](#)。

可能的值為:

預設值

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 欄位的值會設為有效使用者 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, **IsolatedBindingsUserId** 屬性才適用。

IsolatedBindingsUserId 屬性僅與隔離連結的關係搭配使用。此值可讓您指定 *IdentityContext* 結構中的 *UserIdentifier* 欄位 (來自 MQZ_AUTHENTICATE_USER 函數) 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](#)。

可能的值為:

預設值

UserIdentifier 欄位的值會設為有效使用者 ID。

實數

UserIdentifier 欄位的值會設為實際使用者 ID。

有效

UserIdentifier 欄位的值會設為有效使用者 ID。

如需可安裝的服務和元件的相關資訊, 請參閱 [UNIX, Linux, and 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 UNIX and Linux 系統上設定安全](#)。

範例段落

```
Service:  
  Name=AuthorizationService  
  EntryPoints=14
```

相關概念

[AIX、Linux 及 Windows 的可安裝服務及元件](#)

相關參考

[IBM i 上可安裝的服務及元件](#)

[可安裝的服務參照資訊](#)



qm.ini 檔案的 ServiceComponent 段落

ServiceComponent 段落指定服務元件的資訊。當您新增可安裝的服務時, 必須指定服務元件資訊。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。

Service 和 **ServiceComponent** 段落可以任意順序出現，其下的段落索引鍵也可以任意順序出現。對於其中任一段落，所有段落索引鍵都必須存在。如果段落索引鍵重複，則會使用最後一個段落索引鍵。

在啟動時，佇列管理程式會依序處理配置檔中的每一個服務元件項目。然後它會載入指定的元件模組，呼叫元件的進入點 (必須是元件起始設定的進入點)，並傳遞配置控點給它。

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 IBM 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。

範例段落

```
ServiceComponent:  
  Service=AuthorizationService  
  Name=MQSeries.UNIX.auth.service  
  Module=amqzfu  
  ComponentDataSize=0
```

如需顯示 AuthorizationService 段落及其相關聯 ServiceComponent 段落和 NameService 段落及其相關聯 ServiceComponent 段落的進一步範例，請參閱 [第 113 頁的『qm.ini 檔案的服務段落』](#)。

相關概念

[AIX、Linux 及 Windows 的可安裝服務及元件](#)

相關參考

[第 113 頁的『qm.ini 檔案的服務段落』](#)

「服務」段落是用來變更可安裝的服務。此段落包含服務的名稱，以及為服務定義的進入點數目。

[IBM i 上可安裝的服務及元件](#)

[可安裝的服務參照資訊](#)

Windows qm.ini 檔案的 SPX 段落 (僅限 Windows)

SPX 段落指定 SPX 通訊協定配置參數。這些參數會置換通道的預設屬性。

使用 qm.ini 檔案中的 SPX 段落，以指定 SPX 通訊協定配置參數。

Windows **Linux** 或者，在 Linux (x86 及 x86-64) 和 Windows 上，請使用 IBM MQ Explorer SPX 佇列管理程式內容頁面。

Socket = 5E86 (預設值) | socket_number

以十六進位表示法表示的 SPX Socket 號碼。預設值為 X'5E86'。

BoardNum= 0 (預設值) | 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 117 頁的表 13](#) 中。

平台	預設 ListenerBacklog 值
Windows 伺服器	100
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 117 頁的表 13](#) 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，且應該稍後重試連線。

Multi qm.ini 檔案的 SSL 段落

SSL 段落用來配置佇列管理程式上的 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 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。

AllowOutboundSNI = YES (預設值) 否

如果已啟用，具有 SNI 功能的用戶端會在起始 TLS 連線時，將 SNI 設為遠端系統的目標 IBM MQ 通道名稱。如果此屬性設為 NO，具有 SNI 功能的用戶端將不會設定 SNI 標頭，導致出埠連線要求在 TLS 信號交換期間接收遠端佇列管理程式的預設憑證，因此無法使用每個通道憑證。

V 9.1.1 AllowedCipherSpecs =name|name list| ALL

指定已啟用與 Multiplatforms 上 IBM MQ 通道搭配使用的 CipherSpecs 自訂清單。

- 單一 CipherSpec 名稱，或

- 要重新啟用的 IBM MQ CipherSpec 名稱清單 (以逗點區隔) , 或
- ALL 的特殊值, 代表所有 CipherSpecs (不建議)。

註: 不建議啟用 **ALL** CipherSpecs , 因為這將會啟用 SSL 3.0 及 TLS 1.0 通訊協定, 以及大量低保護性加密演算法。

如需相關資訊, 請參閱 [啟用 CipherSpecs](#) 中的 [提供 Multiplatforms 上已啟用 CipherSpecs 的自訂清單](#)。

V 9.1.4 **ULW** **AllowTLSV13=Y | YES | T | TRUE|N|NO|F|FALSE**

指定佇列管理程式是否能夠使用 TLS 1.3 CipherSpecs。

- Y、YES、T 或 TRUE: 啟用 TLS 1.3 , 以容許佇列管理程式使用 TLS 1.3 CipherSpecs。
- N、NO、F 或 FALSE: 停用 TLS 1.3, 這表示佇列管理程式無法使用 TLS 1.3 CipherSpecs。

如需相關資訊, 請參閱 [啟用 CipherSpecs](#)。

CDPCheckExtensions= YES |NO (預設值)

指定此佇列管理程式上的 TLS 通道是否嘗試檢查 CrlDistribution 點憑證延伸中指定的 CDP 伺服器。

- YES (預設值): TLS 通道會嘗試檢查 CDP 伺服器, 以判定是否撤銷數位憑證。
- NO: TLS 通道不會嘗試檢查 CDP 伺服器。這個值是預設值。

V 9.1.4 **ULW** **MinimumRSAKeySize=int**

指定 RSA 憑證必須具備的金鑰大小下限, 以便在 TLS 信號交換期間接受。容許任何等於 0 或以上的值。如果未指定, 則預設為 1。

OCSPAAuthentication=REQUIRED (預設值) | WARN | OPTIONAL

指定當無法從 OCSP 伺服器判斷撤銷狀態時要採取的動作。

如果已啟用 OCSP 檢查, 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 (預設值): 無法判斷撤銷狀態會導致連線因錯誤而關閉。這個值是預設值。
- 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 (預設值) 否

指定此佇列管理程式上的 TLS 通道是否嘗試檢查 AuthorityInfoAccess 憑證延伸中所命名的 OCSP 伺服器。

- YES (預設值): TLS 通道會嘗試檢查 OCSP 伺服器, 以判定是否撤銷數位憑證。這個值是預設值。
- NO: TLS 通道不會嘗試檢查 OCSP 伺服器。

ULW **V 9.1.5** **OCSPTimeout= number**

執行撤銷檢查時等待 OCSP 回應者的秒數。

如果未設定任何值, 則會使用 IBM MQ 預設值 30 秒。

SSLHTTPProxyName= string

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。

Solaris **AIX** 對於 AIX、及 Solaris SPARC 平台、上的 32 位元用戶端, 網址只能是 IPv4 位址。

在其他平台上, 網址可以是 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。

ULW V 9.1.5 SSLHTTPConnectTimeout= *number*|0

執行撤銷檢查時，等待順利建立與 HTTP 伺服器的網路連線的秒數。

如果未設定任何值，則會使用 IBM MQ 預設值 0 (off)。

範例段落

```
SSL:
  OutboundSNI=CHANNEL
  AllowedCipherSpecs=TLS13 CipherSpec list
  AllowTLSV13=Y
  CDPCheckExtensions=NO
  MinimumRSAKeySize=1
  OCSPAuthentication=REQUIRED
  OCSPCheckExtensions=YES
  OCSPTimeout=30
  SSLHTTPConnectTimeout=0
```

附註:

- **OutboundSNI** 的預設值為 **通道**。
- **V 9.1.1** **TLS13 CipherSpec** 清單是特定 CipherSpecs 而非別名密碼的清單。如果您只需要 TLS1.3 密碼，則必須列出它們。例如：
 - TLS_CHACHA20_POLY1305_SHA256
 - TLS_AES_256_GCM_SHA384
 - TLS_AES_128_GCM_SHA256
 - TLS_AES_128_CCM_SHA256
 - TLS_AES_128_CCM_8_SHA256
- **V 9.1.4** 除非您已啟用弱密碼，否則 **AllowTLSV13** 的預設值為 Y，在這種情況下，它會關閉 (除非您明確開啟它)。
- **CDPCheckExtensions** 的值只能是 Yes 或 No。

Multi qm.ini 檔案的子儲存區段落

此段落由 IBM MQ 建立。請不要變更它。

當您建立佇列管理程式時，IBM MQ 會自動寫入子儲存區段落及該段落內的屬性 **ShortSubpoolName**。IBM MQ 會選擇 **ShortSubpoolName** 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，IBM MQ 用於其執行中處理程序之間的內部通訊。

Multi qm.ini 檔案的 TCP 段落

TCP 段落指定「傳輸控制通訊協定/Internet Protocol (TCP/IP)」配置參數。這些參數會置換通道的預設屬性。

使用 qm.ini 檔案中的 TCP 段落，以指定 TCP/IP 配置參數。

Windows **Linux** 或者，在 Linux (x86 和 x86-64) 及 Windows 上，請使用 IBM MQ Explorer SPX TCP 佇列管理程式內容頁。

埠 = 1414 (預設值) | *port_number*

TCP/IP 階段作業的預設埠號 (以十進位表示法表示)。IBM MQ 的已知埠號為 1414。

Windows **Library1= *DLLName1* (僅限 Windows)**

TCP/IP Socket DLL 的名稱。

預設值是 WSOCK32。



KeepAlive= NO (預設值) | 是

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。這可視為在 TCP/IP 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 120 頁的表 14](#) 中。

平台	預設 ListenerBacklog 值
 Windows 伺服器	100
 Linux	100
 Solaris	100
 AIX V5.3 或更新版本	100

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 120 頁的表 14](#) 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，稍後會重試連線。

下列內容群組可用來控制 TCP/IP 使用的緩衝區大小。這些值會直接傳遞至作業系統的 TCP/IP 層。使用這些內容時應非常小心。如果值設定不正確，可能會對 TCP/IP 效能產生不利影響。如需如何影響效能的進一步相關資訊，請參閱您環境的 TCP/IP 文件。零值指出作業系統將管理緩衝區大小，相對於 IBM MQ 所修正的緩衝區大小。

Connect_Timeout= 0 (預設值) | number

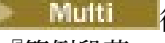
嘗試連接 Socket 逾時之前的秒數。預設值零指定沒有連接逾時。

IBM MQ 通道處理程序透過非阻斷式 Socket 進行連接。因此，如果 Socket 的另一端未備妥，則 connect () 會立即傳回 EINPROGRESS 或 EWOULDBLOCK。在此之後，當報告通訊錯誤時，將再次嘗試連接，最多總共 20 次這類嘗試。

如果 Connect_Timeout 設為非零值，IBM MQ 會在 select () 呼叫期間等待 Socket 備妥。這會增加後續 connect () 呼叫成功的機會。在由於網路上的高負載而導致連接需要一些等待期間的情況下，此選項可能很有用。

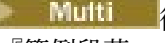
SndBufferSize = number | 0 (預設值)

通道傳送端所使用的 TCP/IP 傳送緩衝區大小 (以位元組為單位)。此段落值可以由通道類型更特定的段落置換，例如 RcvSndBufferSize。如果值設為零，則會使用作業系統預設值。如果未設定任何值，則會使用 IBM MQ 預設值 32768。

 從「IBM MQ 8.0」中，會以預設值 0 自動建立新的佇列管理程式 (請參閱 [第 121 頁的『範例段落』](#))。

RcvBufferSize = number | 0 (預設值)

通道接收端所使用的 TCP/IP 接收緩衝區大小 (以位元組為單位)。此段落值可以由通道類型更特定的段落置換，例如 RcvRcvBufferSize。如果值設為零，則會使用作業系統預設值。如果未設定任何值，則會使用 IBM MQ 預設值 32768。

 從「IBM MQ 8.0」中，會以預設值 0 自動建立新的佇列管理程式 (請參閱 [第 121 頁的『範例段落』](#))。

RcvSndBuffSize = number |0 (預設值)

接收端通道傳送端所使用的 TCP/IP 傳送緩衝區大小 (以位元組為單位)。如果值設為零，則會使用作業系統預設值。如果未設定任何值，則會使用 IBM MQ 預設值 32768。

Multi 從「IBM MQ 8.0」中，會以預設值 0 自動建立新的佇列管理程式 (請參閱 [第 121 頁的『範例段落』](#))。

RcvRcvBuffSize = number |0 (預設值)

接收端通道的接收端所使用的 TCP/IP 接收緩衝區大小 (以位元組為單位)。如果值設為零，則會使用作業系統預設值。如果未設定任何值，則會使用 IBM MQ 預設值 32768。

Multi 從「IBM MQ 8.0」中，會以預設值 0 自動建立新的佇列管理程式 (請參閱 [第 121 頁的『範例段落』](#))。

SvrSndBuffSize = number |0 (預設值)

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)。如果值設為零，則會使用作業系統預設值。如果未設定任何值，則會使用 IBM MQ 預設值 32768。

Multi 從「IBM MQ 8.0」中，會以預設值 0 自動建立新的佇列管理程式 (請參閱 [第 121 頁的『範例段落』](#))。

SvrRcvBuffSize = number |0 (預設值)

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)。如果值設為零，則會使用作業系統預設值。如果未設定任何值，則會使用 IBM MQ 預設值 32768。

Multi 從 IBM MQ 8.0 開始，會以預設值 0 自動建立新的佇列管理程式 (請參閱 [第 121 頁的『範例段落』](#))。

範例段落

```
TCP:
  SndBuffSize=0
  RcvBuffSize=0
  RcvSndBuffSize=0
  RcvRcvBuffSize=0
  ClntSndBuffSize=0
  ClntRcvBuffSize=0
  SvrSndBuffSize=0
  SvrRcvBuffSize=0
```

註: **Multi** 對於 Multiplatforms 上的新佇列管理程式，qm.ini file 的 TCP 段落中的預設 TCP 傳送及接收緩衝區大小會設為由作業系統管理。如前述範例所示，會以傳送及接收緩衝區的預設值 0 自動建立新的佇列管理程式。這僅適用於新的佇列管理程式。會保留從舊版 IBM MQ 移轉的佇列管理程式的 TCP 傳送及接收緩衝區設定。

如果從 qm.ini 檔移除 TCP 緩衝區大小內容，則預設緩衝區會設為 32K。使用此預設值時請小心，因為 32K 可能不是適用於所有傳訊實務的緩衝區。

如果 TCP 傳送和接收緩衝區內容設為零，則會使用 OS 預設值。選擇這些預設值的方法會因作業系統而異，但通常可在 "tcp" 或 get/setsockopt () OS 手動頁面中找到。

V 9.1.0 **Multi** qm.ini 檔案的 TuningParameters 段落

TuningParameters 段落指定用於調整佇列管理程式的選項。

SuppressDspAuthFail= YES |NO (預設值)

設為 YES 時，如果連線缺少對物件的 + dsp 權限，則佇列管理程式會暫停產生授權事件，並在授權檢查失敗時將 AMQ8077 錯誤訊息寫入錯誤日誌中。

ImplSyncOpenOutput=value

ImplSyncOpenOutput 是在針對持續放置啟用隱含同步點之前，在同步點之外，已開啟佇列以供放置的應用程式數目下限。**ImplSyncOpenOutput** 的預設值為 2。

這樣做的效果是如果只有一個應用程式針對放置作業開啟該佇列，則會關閉 **ImplSyncOpenOutput**。

指定 `ImplSyncOpenOutput=1` 表示一律考量隱含的同步點。您可以設定任何正整數值。如果您不想要新增隱含的同步點，請設定 `ImplSyncOpenOutput=OFF`。

V 9.1.2 UniformCluster 名稱 = 叢集的名稱

您用作統一叢集的 IBM MQ 叢集名稱。

V 9.1.0.9 OAMLdapConnectTimeout=time|0 (預設值)

LDAP 用戶端等待建立伺服器 TCP 連線的時間上限 (以秒為單位)。如果您透過連線名單提供多個 LDAP 伺服器，則逾時會套用至每一個個別連線嘗試，因此如果達到此逾時，則會嘗試連線至名稱清單中的下一個項目。

time 的最大值為 3600 秒，而 0 的值為最小值及預設值，表示等待無限制。

V 9.1.0.9 OAMLdapQueryTimeLimit=time|0 (預設值)

一旦建立連線並傳送 LDAP 要求，LDAP 用戶端將等待從伺服器接收 LDAP 要求回應的時間上限 (以秒為單位)。

time 的最大值為 3600 秒，而 0 的值為最小值及預設值，表示等待無限制。

V 9.1.0.15 OAMLdapResponseWarningTime=threshold

如果 LDAP 伺服器連線花費的時間超過 `OAMLdapResponseWarningTime` 參數指定的臨界值秒數，則會將 `AMQ5544W` 訊息寫入錯誤日誌中。預設臨界值為 10 秒。

ExpiryInterval

指出佇列管理程式掃描佇列的頻率，以尋找其他佇列活動尚未清除的過期訊息。這是以秒為單位的時間間隔。

依預設，在正式作業 IBM MQ 建置上，期限掃描器大約每 5 分鐘執行一次。



注意: 通常不需要變更 `ExpiryInterval` 值，您應該僅在 IBM 支援中心的指引下修改此值。

範例段落

V 9.1.0.15

```
TuningParameters:
  SuppressDspAuthFail=N0
  ImplSyncOpenOutput=2
  OAMLdapConnectTimeout=60
  OAMLdapQueryTimeLimit=60
  OAMLdapResponseWarningTime=10
  ExpiryInterval=300
```

相關概念

隱含的同步點

V 9.1.4 Multi qm.ini 檔案的 Variables 段落

Variables 段落指定用於自動統一叢集的配置變數。

在 CONNAME 的自動叢集配置期間，您可以使用 Variables 段落中列出的屬性，以及叢集接收端通道的通道名稱 MQSC 欄位。配置變數無法在 MQSC Script 的任何其他元素中使用。

attribute=value

指定名稱及關聯值，以在 MQSC 定義期間作為插入項目使用。

當您建立佇列管理程式時，可以使用 `crtmqm` 指令上的 `-iv` 指令行選項來提供 `attribute=value` 配對。

範例段落

```
Variables:
  CONNAME=127.0.0.1(1414)
```

相關概念

[自動應用程式平衡](#)

相關工作

[建立新的統一叢集](#)

相關參考

[使用自動叢集配置](#)

Multi **qm.ini 檔的 XAResourceManager 段落**

XAResourceManager 段落，指定在佇列管理程式協調的廣域工作單元中所涉及的資源管理程式相關資訊。

請使用 qm.ini 檔中的 XAResourceManager 段落，來指定在佇列管理程式協調的廣域工作單元中所涉及的資源管理程式相關資訊。

Windows **Linux** 或者，在 Linux (x86 及 x86-64) 和 Windows 上，使用 IBM MQ Explorer XA 資源管理程式佇列管理程式內容頁面。

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。

IBM 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

Windows 此屬性對於 Windows 是必要的。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。

THREAD

資源管理程式完全可察覺執行緒。在多執行緒 IBM MQ 處理程序中，可以同時從多個執行緒對外部資源管理程式進行 XA 函數呼叫。

PROCESS

資源管理程式不是安全執行緒。在多執行緒 IBM MQ 程序中，一次只能對資源管理程式進行一個 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。

範例段落

```
XAResourceManager:
  Name=DB2 Resource Manager Bank
  SwitchFile=/usr/bin/db2swit
  XAOpenString=MQBankDB
  XACloseString=
  ThreadOfControl=THREAD
```

註: 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。

IBM i 的 qm.ini 檔範例

顯示如何在 IBM i 的佇列管理程式配置檔中排列屬性群組的範例。

```
#####
#* Module Name: qm.ini                                     *#
#* Type       : IBM 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 *#
#*          :                                           *#
#####
Log:
LogPath=QMSATURN.Q
LogReceiverSize=65536

CHANNELS:
MaxChannels = 20          ; Maximum number of channels allowed.
                          ; Default is 100.
MaxActiveChannels = 10   ; Maximum number of channels allowed to be
                          ; active at any time. The default is the
                          ; value of MaxChannels.

TCP:
                          ; TCP/IP entries.
KeepAlive = Yes          ; Switch KeepAlive on.
SvrSndBuffSize=20000     ; Size in bytes of the TCP/IP send buffer for each
                          ; channel instance. Default is 32768.
SvrRcvBuffSize=20000     ; Size in bytes of the TCP/IP receive buffer for each
                          ; channel instance. Default is 32768.
Connect_Timeout=10000    ; Number of seconds before an attempt to connect the
                          ; channel instance times out. Default is zero (no timeout).

QMErrorLog:
ErrorLogSize = 262144
ExcludeMessage = 7234
SuppressMessage = 9001,9002,9202
SuppressInterval = 30
```

```
TuningParameters:
ImplSyncOpenOutput=2
```

ULW 安裝配置檔 mqinst.ini

在 UNIX 或 Linux 上，安裝配置檔 `mqinst.ini` 包含所有 IBM MQ 安裝的相關資訊。在 Windows 上，安裝配置資訊位於登錄中。

mqinst.ini 檔案的位置

Linux > UNIX

`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

重要: `mqinst.ini` 檔案不能直接編輯或參照，因為其格式不是固定的，而且可能會變更。

安裝 ID (僅供內部使用) 會自動設定，且不得變更。

您必須使用下列指令來建立、刪除、查詢及修改檔案中的值，而不是直接編輯 `mqinst.ini` 檔案：

`crtmqinst` 以建立項目。

`dltmqinst` 以刪除項目。

`dspmqinst` 以顯示項目。

`setmqinst` 以設定項目。

Windows 上的安裝配置資訊

Windows

Windows 上沒有 `mqinst.ini` 檔案。安裝配置資訊位於登錄中，並保留在下列機碼中：

```
HKLM\SOFTWARE\IBM\WebSphere MQ\Installation\InstallationName
```

重要: 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。

相反地，您必須使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 `dltmqinst` 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。

Multi IBM MQ MQI client 配置檔 mqclient.ini

您可以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。


您可以使用類似於佇列管理程式配置檔 `qm.ini` 的文字檔來配置 IBM MQ MQI clients。該檔案包含許多段落，每一個段落都包含 `attribute-name = value` 格式的多行。

IBM MQ MQI client 配置檔通常命名為 `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下列平台：

- **ULW** UNIX, Linux, and Windows
- **IBM i** IBM i

註: 在 IBM i 上, 沒有預設 mqclient.ini 檔案。不過, 您可以在 IBM i Integrated File System (IFS) 中建立檔案。

如需相關資訊, 請參閱 第 128 頁的『用戶端配置檔的位置』。

註:  z/OS 平台無法用來執行 IBM MQ 用戶端。因此, mqclient.ini 檔案不存在於 IBM MQ for z/OS 上。

IBM MQ MQI client 配置檔中的屬性適用於使用下列項目的用戶端:

- MQI
- IBM MQ classes for Java
- IBM MQ classes for JMS
- IBM MQ classes for .NET
- XMS

雖然 IBM MQ MQI client 配置檔中的屬性適用於大部分 IBM 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 MQ classes for Java 或 IBM MQ classes for JMS 的用戶端讀取。如需相關資訊, 請參閱第 129 頁的『哪些 IBM 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)」。

 以下是 IBM MQ 9.1.2 中 Continuous Delivery 的用戶端配置檔範例:

```
## Module Name: mqclient.ini                ##
## Type       : IBM 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=myMod
  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)

Connection:
  ApplName=ExampleApplName
```

以下是用戶端配置檔 適用於 IBM MQ 9.1.0 Long Term Support 版本及 Continuous Delivery 之前 IBM MQ 9.1.2 的範例:

```
## Module Name: mqclient.ini                ##
## Type       : IBM 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=myMod
  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)
```

您無法使用用戶端配置檔來設定多個通道連線。

在 IBM WebSphere MQ 7.0 之前的版本中支援的環境變數仍在更新版本中受支援，且在這類環境變數符合用戶端配置檔中的對等值時，環境變數會置換用戶端配置檔值。

對於使用 IBM 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：

- 透過在 JMS 配置檔中設定內容。
- 透過設定 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的環境變數。

配置檔中的註解



您可以使用分號 ';' 和雜湊 '#' 字元來標示配置檔內註解的開頭。這可以將整行標示為註解，或表示行尾的註解，不會包含在設定值中。

如果值需要其中一個字元，則您必須使用反斜線字元 '\' 來跳出該字元。

下列範例顯示配置檔內註解的用法：

```
# Example of an SSL stanza with comments
SSL:
  ClientRevocationChecks=REQUIRED ; Example of an end of line comment
  SSLCryptoHardware=GSK_PKCS11=/driver\;label\;password\;SYMMETRIC_CIPHER_ON # Example of
  escaped comment characters.
```

相關概念

第 129 頁的『[哪些 IBM MQ 用戶端可以讀取每一個屬性](#)』

IBM 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 MQ classes for Java 或 IBM MQ classes for JMS 的用戶端讀取。

相關參考

第 128 頁的『[用戶端配置檔的位置](#)』

IBM MQ MQI client 配置檔可以保留在許多位置中。

用戶端配置檔的位置

IBM MQ MQI client 配置檔可以保留在許多位置中。

用戶端應用程式使用下列搜尋路徑來尋找 IBM MQ MQI client 配置檔：

1. 環境變數 MQCLNTCF 指定的位置。

此環境變數的格式為完整 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。

附註：

- 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
 - 為了容許不支援讀取環境變數的 Java 1.4.2 JRE，MQCLNTCF 環境變數可以置換為 MQCLNTCF Java 系統內容。
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 MQ 資料目錄中，稱為 mqclient.ini 的檔案。

附註：

- IBM MQ 資料目錄不存在於特定平台上 (例如 IBM i 和 z/OS)，或在用戶端已隨附另一個產品的情況下。

IBM i 在 IBM i 上，沒有預設 mqclient.ini 檔案。不過，檔案可以建立在 IBM i Integrated File System (IFS) 的 /QIBM/UserData/mqm/ 目錄中，以及定義指向它的環境變數 **MQCLNTCF**。例如：

```
ADDENVVAR ENVVAR(MQCLNTCF) VALUE('/QIBM/UserData/mqm/mqclient.ini') REPLACE(*YES)
```

如需環境變數的其他範例，請參閱 [環境變數](#)。

z/OS z/OS 平台無法用來執行 IBM MQ 用戶端。因此，mqclient.ini 檔案不存在於 IBM MQ for z/OS 上。

- **Linux** **UNIX** 在 UNIX and Linux 系統上，目錄為 /var/mqm
- **Windows** 在 Windows 平台上，您可以在安裝期間配置環境變數 MQ_DATA_PATH，以指向資料目錄。它通常是 C:\ProgramData\IBM\MQ

註：如果您只安裝用戶端，環境變數可能是 MQ_FILE_PATH。

- 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_DATA_PATH 環境變數置換為 MQ_DATA_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。
 - **Linux** **UNIX** 對於 UNIX and Linux 平台上的 C 用戶端，這是 HOME 環境變數的值。
 - **Windows** 對於 Windows 上的 C 用戶端，這是 HOMEDRIVE 及 HOMEPATH 環境變數的連結值。

哪些 IBM MQ 用戶端可以讀取每一個屬性

IBM 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 MQ classes for Java 或 IBM MQ classes for JMS 的用戶端讀取。

表 15: 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						
CCSID	要使用的編碼字集號碼。	是	否	否	是	是
ChannelDefinition 名錄	包含用戶端通道定義表之檔案的目錄路徑。	是	否	否	是	是
ChannelDefinition 檔案	包含用戶端通道定義表的檔案名稱。	是	否	否	是	是
ReconDelay	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
DefRecon	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
MQReconnectTimeout	重新連接至用戶端的逾時 (以秒為單位)。	是	否	否	是	否
ServerConnection 參數	IBM MQ 伺服器的位置，以及要使用的通訊方法。	是	否	否	是	是
Put1DefaultAlwaysSync	使用選項 MQPMO_RESPONSE_AS_Q_DEF 控制 MQPUT1 函數呼叫的行為。	是	是	是	是	是

表 15: 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
PasswordProtection	可讓您在 MQCSP 結構中設定受保護的密碼，而不是使用 SSL 或 TLS。	是	是	是	是	是
ClientExit 路徑段落						
ExitsDefaultPath	指定用戶端 32 位元通道結束程式的位置。	是	是	是	是	是
ExitsDefaultPath64	指定用戶端 64 位元通道結束程式的位置。	是	是	是	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
▶ V 9.1.2 ▶ V 9.1.2 連線段落						
ApplName	用戶端配置檔中指定的應用程式名稱。	是	否	否	否	否
JMQI 段落						
useMQCSPAuthentication	控制在向佇列管理程式進行鑑別時，IBM MQ classes for Java 和 IBM MQ classes for JMS 應用程式是否應該使用相容模式或 MQCSP 鑑別模式。	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

表 15: 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
<u>PurgeTime</u>	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
<u>UpdatePercentage</u>	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
PreConnect 段落						
<u>資料</u>	儲存連線定義的儲存庫 URL。	是	否	否	否	否
<u>函數</u>	進入包含 PreConnect 結束碼之媒體庫的功能進入點名稱。	是	否	否	否	否
<u>模組</u>	包含 API 結束碼的模組名稱。	是	否	否	否	否
<u>順序</u>	相對於其他結束程式, 呼叫此結束程式的順序。	是	否	否	否	否
安全段落						
<u>DisableClientAMS</u>	停用或啟用 AMS, 以進行與佇列管理程式的用戶端連線。	是	是	是	否	否
SSL 段落						
<u>AllowOutboundSNI</u>	指定具有 SNI 功能的用戶端在起始 TLS 連線時, 是否將 SNI 設為遠端系統的目標 IBM MQ 通道名稱。	是	否	否	否	否


表 15: 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
V 9.1.4 AllowTLSV13	佇列管理程式是否能夠使用 TLS 1.3 CipherSpecs。	是 (C/C++ 用戶端)	否	否	否	否
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
ClientRevocationCheck	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
V 9.1.4 MinimumRSAKeySize	指定 RSA 憑證必須具備才能被接受的金鑰大小下限。	是 (C/C++ 用戶端)	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 MQ 的行為。	是	否	否	否	否
OCSPCheckExtensions	控制 IBM MQ 是否作用於 AuthorityInfo Access 憑證延伸。	是	否	否	否	否

表 15: 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
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 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
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

表 15: 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
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

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。

註: 這個段落的每一個屬性的說明, 指出哪些 IBM MQ 用戶端可以讀取該屬性。如需所有 IBM MQ MQI client 配置檔段落的摘要表格, 請參閱 [每一個用戶端可以讀取哪些 IBM MQ 屬性](#)。

在 CHANNELS 段落中可以包含下列屬性:

CCSID = number

要使用的編碼字集號碼。


此屬性可以由 C、未受管理 .NET、受管理 .NET 及受管理 XMS .NET 用戶端讀取。



CCSID 號碼相當於 [MQCCSID](#) 環境變數。

ChannelDefinitionDirectory = path

包含用戶端通道定義表之檔案的目錄路徑。

此屬性可以由 C、未受管理 .NET、受管理 .NET 及受管理 XMS .NET 用戶端讀取。

 在 Windows 系統上, 預設值是 IBM MQ 資料及日誌檔目錄, 通常是 C:\ProgramData\IBM\MQ。

  在 UNIX and Linux 系統上, 預設值為 /var/mqm。

ChannelDefinitionDirectory 可以包含與 ChannelDefinitionFile 屬性一起使用的 URL (請參閱 [第 46 頁的『CCDT 的 URL 存取權』](#))。

ChannelDefinition 名錄路徑相當於 [MQCHLLIB](#) 環境變數。

ChannelDefinitionFile = filename|AMQCLCHL.TAB

包含用戶端通道定義表的檔案名稱。

此屬性可以由 C、未受管理 .NET、受管理 .NET 及受管理 XMS .NET 用戶端讀取。

用戶端通道定義表相當於 [MQCHLTAB](#) 環境變數。

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。

此屬性可以由 C、未受管理 .NET、IBM MQ classes for JMS、受管理 .NET 及受管理 XMS .NET 用戶端讀取。

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 |DISABLED

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。

此屬性可以由 C、未受管理 .NET、IBM MQ classes for JMS、受管理 .NET 及受管理 XMS .NET 用戶端讀取。

IBM MQ classes for Java 不支援自動用戶端重新連線。

DefRecon 選項的解譯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。

否

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

表 16: 自動重新連線取決於在應用程式及 mqclient.ini 檔案中設定的值

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
否	YES	QMGR	NO	NO
YES	YES	QMGR	YES	NO
QMGR	YES	QMGR	QMGR	NO
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)。

此屬性可以由 C 及未受管理的 .NET 用戶端及受管理 .NET 用戶端讀取。

IBM MQ classes for JMS 用戶端可以使用 Connection Factory 內容 CLIENTRECONNECTTIMEOUT 來指定重新連接的逾時。此內容的預設值是 1800 秒 (30 分鐘)。

IBM MQ classes for XMS .NET 用戶端可以使用下列內容指定重新連接的逾時：

- Connection Factory 內容 `CLIENTRECONNECTTIMEOUT`。此內容的預設值是 1800 秒（30 分鐘）。此內容僅適用於「受管理」模式。
- 內容 `XMSC.WMQ_CLIENT_RECONNECT_TIMEOUT`。此內容的預設值是 1800 秒（30 分鐘）。此內容僅適用於「受管理」模式。

ServerConnection 參數

ServerConnection 參數相等於 `MQSERVER` 環境變數，並指定 IBM MQ 伺服器的位置及要使用的通訊方法。

此屬性可以由 C、未受管理 .NET、受管理 .NET 及受管理 XMS .NET 用戶端讀取。

ServerConnectionParms 屬性僅定義簡式通道；您無法使用它來定義 TLS 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，因為這個字元用來區隔通道名稱、傳輸類型和連線名稱。

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 (預設值) | YES

使用選項 `MQPMO_RESPONSE_AS_Q_DEF` 控制 `MQPUT1` 函數呼叫的行為。

此屬性可以由 C、未受管理 .NET、IBM MQ classes for Java、IBM MQ classes for JMS、受管理 .NET 及受管理 XMS .NET 用戶端讀取。

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。

PasswordProtection = 相容|一律|選用

從 IBM MQ 8.0 開始，可讓您在 `MQCSP` 結構中設定受保護的密碼，而不是使用 TLS。

此屬性可以由 C、未受管理 .NET、IBM MQ classes for Java、IBM MQ classes for JMS、受管理 .NET 及受管理 XMS .NET 用戶端讀取。

`MQCSP` 密碼保護適用於測試及開發目的，因為使用 `MQCSP` 密碼保護比設定 TLS 加密更簡單，但沒有那麼安全。

如需進一步資訊，請參閱 [MQCSP 密碼保護](#)。

相關工作

將 IBM MQ MQI 應用程式連接至佇列管理程式

Multi 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。

註: 這個段落的每一個屬性的說明, 指出哪些 IBM MQ 用戶端可以讀取該屬性。如需所有 IBM MQ MQI client 配置檔段落的摘要表格, 請參閱每一個用戶端可以讀取哪些 IBM MQ 屬性。

ClientExit 路徑段落中可以包含下列屬性:

ExitsDefault 路徑 = 字串

指定用戶端 32 位元通道結束程式的位置。

此屬性可以由 C、未受管理 .NET、受管理 .NET、受管理 XMS .NET、IBM MQ classes for Java 及 IBM MQ classes for JMS 用戶端讀取。IBM MQ classes for Java 及 IBM MQ classes for JMS 用戶端使用此屬性來尋找未在 Java 中寫入的 32 位元通道結束程式。

ExitsDefaultPath64 = string

指定用戶端 64 位元通道結束程式的位置。

此屬性可以由 C、未受管理 .NET、受管理 .NET、受管理 XMS .NET、IBM MQ classes for Java 及 IBM MQ classes for JMS 用戶端讀取。IBM MQ classes for Java 及 IBM MQ classes for JMS 用戶端使用此屬性來尋找未在 Java 中寫入的 64 位元通道結束程式。

JavaExitsClassPath = string

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。

IBM MQ classes for Java 及 IBM MQ classes for JMS 用戶端可以讀取此屬性。

在 JMS 配置檔中, JavaExitsClass 路徑名稱會提供標準 com.ibm.mq.cfg。字首和這個完整名稱也會用於 IBM WebSphere MQ 7.0 或更新版本的系統內容。在 IBM WebSphere MQ 6.0, 此屬性是使用系統內容 com.ibm.mq.exitClasspath 指定的, 該內容記錄在 IBM WebSphere MQ 6.0 Readme 檔中。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 IBM WebSphere MQ 7.0 或更新版本中仍允許使用它。

V 9.1.2

Multi

用戶端配置檔的連線段落

請使用連線段落來指定應用程式名稱。

註: 這個段落的每一個屬性的說明, 指出哪些 IBM MQ 用戶端可以讀取該屬性。如需所有 IBM MQ MQI client 配置檔段落的摘要表格, 請參閱每一個用戶端可以讀取哪些 IBM MQ 屬性。

「連線」段落中可以包含下列屬性:

ApplName = ExampleApplName

您可以在用戶端配置檔中指定應用程式名稱。

此屬性可供 C 及未受管理的 .NET 用戶端使用。

Multi

用戶端配置檔的 JMQUI 段落

使用 JMQUI 段落來指定 IBM MQ classes for Java 及 IBM MQ classes for JMS 所使用之 Java 訊息佇列作業介面 (JMQUI) 的配置參數。

註: 這個段落的每一個屬性的說明, 指出哪些 IBM MQ 用戶端可以讀取該屬性。如需所有 IBM MQ MQI client 配置檔段落的摘要表格, 請參閱每一個用戶端可以讀取哪些 IBM MQ 屬性。

JMQUI 段落中可以包含下列屬性:

useMQCSPauthentication = NO|YES

控制在向佇列管理程式進行鑑別時, IBM MQ classes for Java 和 IBM MQ classes for JMS 應用程式是否應該使用相容模式或 MQCSP 鑑別模式。

此屬性可由 IBM MQ classes for Java 及 IBM MQ classes for JMS 用戶端讀取。

此屬性可以有如下列值:

NO

向佇列管理程式進行鑑別時使用相容模式。這是預設值。

YES

向佇列管理程式進行鑑別時，請使用 MQCSP 鑑別模式。

如需相容模式及 MQCSP 鑑別模式的相關資訊，請參閱 [使用 Java 用戶端進行連線鑑別](#)。

Windows 用戶端配置檔的 LU62、NETBIOS 及 SPX 段落

僅在 Windows 系統上，使用這些段落來指定所指定網路通訊協定的配置參數。

LU62 段落

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，用於將程式碼儲存在兩個個別的程式庫中。

TPName

要在遠端站台上啟動的 TP 名稱。

NETBIOS 段落

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 環境變數。

NumCmds = *number*|1

要配置的指令數。

NumSess = *number*|1

要配置的階段作業數。

SPX 段落

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。

Multi 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。

註: 這個段落的每一個屬性的說明, 指出哪些 IBM MQ 用戶端可以讀取該屬性。如需所有 IBM MQ MQI client 配置檔段落的摘要表格, 請參閱每一個用戶端可以讀取哪些 IBM MQ 屬性。

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。

此屬性可以由 C、未受管理 .NET、IBM MQ classes for Java、IBM MQ classes for JMS、受管理 .NET 及受管理 XMS .NET 用戶端讀取。

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。

此屬性可以由 C、未受管理 .NET、IBM MQ classes for Java、IBM MQ classes for JMS、受管理 .NET 及受管理 XMS .NET 用戶端讀取。

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。

此屬性可以由 C、未受管理 .NET、IBM MQ classes for Java、IBM MQ classes for JMS、受管理 .NET 及受管理 XMS .NET 用戶端讀取。

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。

Multi 用戶端配置檔的 PreConnect 段落

使用 PreConnect 段落來配置 mqclient.ini 檔案中的 PreConnect 結束程式。

註: 這個段落的每一個屬性的說明，指出哪些 IBM MQ 用戶端可以讀取該屬性。如需所有 IBM MQ MQI client 配置檔段落的摘要表格，請參閱[每一個用戶端可以讀取哪些 IBM MQ 屬性](#)。

PreConnect 段落中可以包含下列屬性:

資料 = user_data

此屬性指定傳遞至預先連接結束程式的使用者資料。傳遞至預先連接結束程式的資料是您正在使用之預先連接結束程式的實作及其預期傳遞的資料所特有。

此屬性可由 C 及未受管理的 .NET 用戶端讀取。

例如，此屬性可用來指定儲存連線定義之儲存庫的 URL，例如使用 LDAP 伺服器時:

```
Data = ldap://myLDAPServer.com:389/cn=wmq,ou=ibm,ou=com
```

函數 = myFunc

進入包含 PreConnect 結束碼之媒體庫的功能進入點名稱。

此屬性可由 C 及未受管理的 .NET 用戶端讀取。

函數定義遵循 PreConnect 結束程式原型 [MQ_PRECONNECT_EXIT](#)。

此欄位的長度上限為 MQ_EXIT_NAME_LENGTH。

模組 = myMod

包含 API 結束碼的模組名稱。

此屬性可由 C 及未受管理的 .NET 用戶端讀取。

如果此欄位包含模組的完整路徑名稱，則會依現狀使用。

順序 = sequence_number

相對於其他結束程式，呼叫此結束程式的順序。在具有較高序號的結束程式之前，會先呼叫具有較低序號的結束程式。不需要結束程式的序列編號是連續的; 序列 1、2、3 的結果與序列 7、42、1096 相同。此屬性是不帶正負號的數值。

此屬性可由 C 及未受管理的 .NET 用戶端讀取。

在 mqclient.ini 檔案內可以定義多個 PreConnect 段落。每一個結束程式的處理順序由段落的 Sequence 屬性決定。

相關工作

使用儲存庫中的預先連接結束程式來參照連線定義

Multi 用戶端配置檔的安全段落

使用「安全」段落來停用或啟用 AMS，以進行用戶端與佇列管理程式的連線。

註: 這個段落的每一個屬性的說明，指出哪些 IBM MQ 用戶端可以讀取該屬性。如需所有 IBM MQ MQI client 配置檔段落的摘要表格，請參閱[每一個用戶端可以讀取哪些 IBM MQ 屬性](#)。

下列屬性可以包含在「安全」段落中:

DisableClientAMS = NO|YES

如果您使用 IBM WebSphere MQ 7.5 或更新版本用戶端從舊版產品連接至佇列管理程式，且報告 2085 (MQRC_UNKNOWN_OBJECT_NAME) 錯誤，則 DisableClientAMS 屬性可讓您停用 IBM MQ Advanced Message Security (AMS)。

從 IBM WebSphere MQ 7.5 開始，會自動在 IBM MQ 用戶端中啟用 IBM MQ Advanced Message Security (AMS)，因此依預設，用戶端會嘗試檢查佇列管理程式中物件的安全原則。不過，舊版產品 (例如 IBM WebSphere MQ 7.1) 上的伺服器未啟用 AMS，這會導致報告 2085 (MQRC_UNKNOWN_OBJECT_NAME) 錯誤。

下列範例顯示如何使用 DisableClientAMS 屬性:

- 若要停用 AMS，請執行下列動作:

```
Security:
DisableClientAMS=Yes
```

- 若要啟用 AMS，請執行下列動作:

```
Security:
DisableClientAMS=No
```

此屬性可由 C、IBM MQ classes for Java 及 IBM MQ classes for JMS 用戶端讀取。

相關工作

在用戶端停用 Advanced Message Security

Multi 用戶端配置檔的 SSL 段落

請使用 SSL 段落來指定使用 TLS 的相關資訊。

註: 這個段落的每一個屬性的說明，指出哪些 IBM MQ 用戶端可以讀取該屬性。如需所有 IBM MQ MQI client 配置檔段落的摘要表格，請參閱[每一個用戶端可以讀取哪些 IBM MQ 屬性](#)。

SSL 段落中可以包含下列屬性:

AllowOutboundSNI = YES (預設值) 否

如果已啟用，具有 SNI 功能的用戶端會在起始 TLS 連線時，將 SNI 設為遠端系統的目標 IBM MQ 通道名稱。如果此屬性設為 NO，具有 SNI 功能的用戶端將不會設定 SNI 標頭，導致出埠連線要求在 TLS 信號交換期間接收遠端佇列管理程式的預設憑證，因此無法使用每個通道憑證。

此屬性可由 C 及未受管理的 .NET 用戶端讀取。

V 9.1.4 ULW AllowTLSV13 = Y | YES | T | TRUE | N | NO | F | FALSE

指定佇列管理程式是否能夠使用 TLS 1.3 CipherSpecs (請參閱 [啟用 CipherSpecs](#))。

此屬性可由 C/C++ 用戶端讀取。

此屬性有下列可能值:

- Y (預設值)、YES (預設值)、T (預設值) 或 TRUE (預設值): 啟用 TLS 1.3，以容許佇列管理程式使用 TLS 1.3 CipherSpecs。
- N、NO、F 或 FALSE: 停用 TLS 1.3，這表示佇列管理程式無法使用 TLS 1.3 CipherSpecs。

註: 使用 MQI 用戶端時, 除非在應用程式所使用 `mqclient.ini` 檔案的 SSL 段落中明確指定 **AllowTLSV13** 值, 否則會推斷該值。如需相關資訊, 請參閱 [IBM MQ MQI 用戶端及 TLS 1.3](#)。

CDPCheckExtensions = YES|NO (預設值)

CDPCheckExtensions 指定此佇列管理程式上的 TLS 通道是否嘗試檢查在 CrlDistribution 點憑證延伸中命名的 CDP 伺服器。

此屬性可由 C 及未受管理的 .NET 用戶端讀取。

此屬性有下列可能值:

- YES (預設值): TLS 通道會嘗試檢查 CDP 伺服器, 以判定是否撤銷數位憑證。
- NO: TLS 通道不會嘗試檢查 CDP 伺服器。這個值是預設值。

CertificateLabel = string

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。

如需相關資訊, 請參閱 [憑證標籤 \(CERTLABL\)](#)。

CertificateVal 原則 = 字串

決定使用的憑證驗證類型。

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:

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。

ClientRevocation 檢查 = 必要|選用|已停用

判定如果用戶端連接呼叫使用 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。

如果您使用 MQSCO 或 CCDT, 則連線會成功。如果沒有 CCDT 檔案, 且也未提供 MQSCO, 則連線會失敗, 原因碼為 2059, 且錯誤日誌會報告 AMQ9518E: 檔案 '/var/mqm/AMQCLCHL.TAB'。

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 = string

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。

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:

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。

V 9.1.4

ULW

MinimumRSAKeySize=int

指定 RSA 憑證必須具備才能被接受的金鑰大小下限。容許任何等於 0 或以上的值。如果未指定，則預設為 1。

此屬性可由 C/C++ 用戶端讀取。

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，IBM MQ 的行為。另請參閱 **ClientRevocationChecks**。

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：

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 MQRQ_SSL_HANDSHAKE_ERROR 為 ReasonQualifier 的 MQRQ_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 MQRQ_SSL_UNKNOWN_REVOCATION 之 ReasonQualifier 的 MQRQ_CHANNEL_SSL_WARNING 訊息。容許連線繼續進行。

OCSPCheckExtensions = YES|NO

控制 IBM MQ 是否作用於 AuthorityInfoAccess 憑證延伸。

此屬性可由 C 及未受管理的 .NET 用戶端讀取。

如果值設為 NO，IBM 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。

ULW

V 9.1.5

OCSPTimeout = number

執行撤銷檢查時等待 OCSP 回應者的秒數。

此屬性可由 C 及未受管理的 .NET 用戶端讀取。

如果未設定任何值，則會使用 IBM MQ 預設值 30 秒。

SSLCryptoHardware = string

設定在系統上配置 PKCS #11 加密硬體所需的參數字串。

此屬性可由 C 及未受管理的 .NET 用戶端讀取。

以下列格式指定字串: GSK_PKCS11 = *driver path and filename;token label;token password;symmetric cipher setting;*

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。

  當提供字串的不同元件時，您必須使用反斜線字元來跳出分號字元，因為分號字元會被視為註解。例如: '\;'

SSLFipsRequired = YES|NO

指定在 IBM MQ 中執行加密法時是否只使用 FIPS 認證的演算法。



此屬性可由 C 及未受管理的 .NET 用戶端讀取。

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 = string

該字串是 GSKit 用於 OCSP 檢查之 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。

此屬性可由 C 及未受管理的 .NET 用戶端讀取。

  對於 AIX 上的 Sun Solaris SPARC 平台，以及適用於 32 位元用戶端，網址只能是 IPv4 位址。

在其他平台上，網址可以是 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。

SSLHTTPConnectTimeout = number|0

執行撤銷檢查時，等待順利建立與 HTTP 伺服器的網路連線的秒數。

此屬性可由 C 及未受管理的 .NET 用戶端讀取。

如果未設定任何值，則會使用 IBM MQ 預設值 0 (off)。

SSLKeyRepository = pathname

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。

此屬性可由 C 及未受管理的 .NET 用戶端讀取。

SSLKeyResetCount = integer|0

在重新協議秘密金鑰之前，在 TLS 通道上傳送及接收的未加密位元組數。

此屬性可由 C 及未受管理的 .NET 用戶端讀取。

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 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。

註: 這個段落的每一個屬性的說明，指出哪些 IBM MQ 用戶端可以讀取該屬性。如需所有 IBM MQ MQI client 配置檔段落的摘要表格，請參閱每一個用戶端可以讀取哪些 IBM MQ 屬性。

TCP 段落中可以包含下列屬性:

ClntRcvBuffSize = number|0

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)。

此屬性可以由 C、未受管理 .NET、IBM MQ classes for Java、IBM MQ classes for JMS、受管理 .NET 及受管理 XMS .NET 用戶端讀取。

零值指出作業系統將管理緩衝區大小，相對於 IBM MQ 所修正的緩衝區大小。如果值設為零，則會使用作業系統預設值。如果未設定任何值，則會使用 IBM MQ 預設值 32768。

ClntSndBuffSize = number|0

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)。

此屬性可以由 C、未受管理 .NET、IBM MQ classes for Java、IBM MQ classes for JMS、受管理 .NET 及受管理 XMS .NET 用戶端讀取。

零值指出作業系統將管理緩衝區大小，相對於 IBM MQ 所修正的緩衝區大小。如果值設為零，則會使用作業系統預設值。如果未設定任何值，則會使用 IBM MQ 預設值 32768。

Connect_Timeout = 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。

此屬性可由 C、未受管理的 .NET、IBM MQ classes for Java 及 IBM MQ classes for JMS 用戶端讀取。

IBM MQ 通道處理程序透過非阻斷式 Socket 進行連接。因此，如果 Socket 的另一端未備妥，則 connect () 會立即傳回 *EINPROGRESS* 或 *EWOULDBLOCK*。在此之後，當報告通訊錯誤時，將再次嘗試連接，最多總共 20 次這類嘗試。

如果 Connect_Timeout 設為非零值，IBM MQ 會在 select () 呼叫期間等待 Socket 備妥。這會增加後續 connect () 呼叫成功的機會。在由於網路上的高負載而導致連接需要一些等待期間的情況下，此選項可能很有用。

在 Connect_Timeout、ClntSndBuffSize 及 ClntRcvBuffSize 參數之間沒有關係。

IPAddressVersion = MQIPADDR_IPV4|MQIPADDR_IPV6

指定要用於通道連線的 IP 通訊協定。

此屬性可以由 C、未受管理 .NET、受管理 .NET 及受管理 XMS .NET 用戶端讀取。

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。

此屬性可以由 C、未受管理 .NET、IBM MQ classes for Java、IBM MQ classes for JMS、受管理 .NET 及受管理 XMS .NET 用戶端讀取。

Windows Library1 = DLLName|WSOCK32

(僅限 Windows) TCP/IP Socket DLL 的名稱。

此屬性可由 C 及未受管理的 .NET 用戶端讀取。

Multi 活動追蹤配置檔 mqat.ini

活動追蹤配置檔 mqat.ini 用來配置活動追蹤行為。此檔案用來定義報告活動追蹤資料的層次及頻率。此檔案也提供一種方法來定義規則，以根據應用程式名稱來啟用及停用活動追蹤。

mqat.ini 檔案遵循與 mqs.ini 及 qm.ini 檔案相同的段落索引鍵及參數值配對格式。此檔案由單一段落「AllActivity 追蹤」組成，依預設用於配置所有活動追蹤的活動追蹤資料報告層次及頻率。該檔案也可以包含多個 ApplicationTrace 段落。這些段落中的每一個都會根據連線的應用程式名稱與規則的比對，為一個以

上連線的追蹤行為定義規則。如需相關資訊，請參閱 [應用程式活動追蹤](#) 及 [使用 mqat.ini 配置活動追蹤行為](#)。

佇列管理程式會套用一些規則，以決定要用於連線的段落設定。對於符合 ApplicationTrace 段落的那些連線，您可以選擇性地在「AllActivity 追蹤」段落下置換廣域追蹤層次和頻率設定。如需相關資訊，請參閱 [使用 mqat.ini 來配置活動追蹤行為](#)。

目錄位置

IBM i **Linux** **UNIX** 在 UNIX and Linux 及 IBM i 系統上，mqat.ini 位於佇列管理程式資料目錄中，該目錄與 qm.ini 檔案位於相同位置。

Windows 在 Windows 系統上，mqat.ini 位於佇列管理程式資料目錄 C:\Program Files\IBM\WebSphere MQ\qmgrs\queue_manager_name 中。執行要追蹤之應用程式的使用者需要讀取此檔案的許可權。

Multi AllActivity mqat.ini 檔案的追蹤段落

mqat.ini 配置檔的 AllActivity 追蹤段落指定用來配置佇列管理程式追蹤層次的參數。

除非置換，否則單一 AllActivity 追蹤段落會定義套用到所有 IBM MQ 連線之活動追蹤的設定。

[ApplicationTrace 段落](#)中的更具體資訊可以置換 AllActivity 追蹤段落中的個別值。

如果指定多個 AllActivity 追蹤段落，則會使用最後一個段落中的值。所選 AllActivity 追蹤中遺漏的參數採用預設值。會忽略先前 AllActivity 追蹤段落中的參數及值。

ActivityInterval

追蹤訊息之間的時間間隔(秒)。活動追蹤不使用計時器執行緒，因此追蹤訊息不會在經歷時間的確切時刻寫入，而是在經歷時間間隔之後執行第一個 MQI 作業時寫入。如果此值為 0，則會在連線中斷時(或達到活動計數時)寫入追蹤訊息。預設值為 1。

ActivityCount

追蹤訊息之間的 MQI 作業數。如果此值為 0，則會在連線中斷(或活動間隔過了)時寫入追蹤訊息。預設值為 100。

TraceLevel

針對每一個作業所追蹤的參數詳細資料數量。個別作業的說明詳細說明針對每一個追蹤層次包括哪些參數。設為低、中或高。預設值為 MEDIAL。

TraceMessage 資料

針對 MQGET、MQPUT、MQPUT1 及回呼作業所追蹤的訊息資料量(以位元組為單位)。預設為 0。

StopOnGetTrace 訊息

可以設為 ON 或 OFF。預設為 ON。

SubscriptionDelivery

可以設為 BATCHED 或 IMMEDIATE。決定當有一或多個活動追蹤訂閱存在時，是否要使用 **ActivityInterval** 和 **ActivityCount** 參數。將此參數設為 IMMEDIATE 會導致在追蹤資料具有相符訂閱時，將 **ActivityInterval** 及 **ActivityCount** 值置換為有效值 1。每一個活動追蹤記錄不會與來自相同連線的其他記錄一起批次處理，而是立即遞送至訂閱而沒有延遲。IMMEDIATE 設定會增加收集活動追蹤資料的效能額外負擔。預設值是 BATCHED。

相關工作

[使用 mqat.ini 配置活動追蹤行為](#)

Multi mqat.ini 檔案的 ApplicationTrace 段落

mqat.ini 配置檔可以包含多個 ApplicationTrace 段落。每一個這些段落根據比對連線的應用程式名稱與規則，為一或多個連線定義追蹤行為的規則。

您可以為 ApplicationTrace 段落設定下列值：

追蹤

可設為 ON 或 OFF 的活動追蹤開關。 **Trace** 參數是沒有預設值的必要參數。 它可以在應用程式特定段落中使用，以判定現行應用程式段落範圍的活動追蹤是否為作用中。 請注意，此值會置換佇列管理程式的 **ACTVTRC** 及 **ACTVCONO** 設定。

ApplName

ApplName 參數指定為字串，是沒有預設值的必要參數。 此值用來決定 ApplicationTrace 段落適用於哪些應用程式。 它與 API 結束程式環境定義結構 (相當於 MQMD.PutApplName) 中的 **ApplName** 值相符。 **ApplName** 值的內容會根據應用程式環境而有所不同。

在 Multiplatforms 上，只有 MQAXC.ApplName 符合段落中的值。 進行比較時，會忽略最右邊路徑分隔字元左側的字元。

ApplName 值結尾可以使用單一萬用字元 (*)，以符合該點之後任何數目的字元。 如果 **ApplName** 值設為單一萬用字元 (*)，則 **ApplName** 值符合所有應用程式。

IBM i ApplFunction

ApplFunction 參數指定為字串。 預設值為 *。 此參數的值用來限定套用 ApplicationTrace 段落及 **ApplName** 值的應用程式。

該段落是選用的，僅適用於 IBM i 佇列管理程式。 **ApplName** 值結尾可以使用單一萬用字元 (*) 來符合任意數目的字元。 例如，指定 **ApplName = *** 及 **ApplFunction = AMQSPUTO** 的 ApplicationTrace 段落適用於任何工作中 AMQSPUTO 程式的所有呼叫。

ApplClass

ApplClass 參數定義應用程式的類別，且可以設為下列值：

- 使用者
- MCA
- ALL (這是預設值)

如需 **AppType** 值如何對應至 IBM MQ 連線的說明，請參閱 [使用 mqat.ini 來配置活動追蹤行為](#) 中的表 3。

選擇性地針對符合 ApplicationTrace 段落的那些連線，可以置換 AllActivity 追蹤段落下的廣域追蹤層次及頻率設定。

可以在 ApplicationTrace 段落下設定下列參數。 如果未設定，則會從 [AllActivity 追蹤段落](#) 設定繼承值：

ActivityInterval

追蹤訊息之間的時間間隔 (秒)。 活動追蹤不使用計時器執行緒，因此追蹤訊息不會在經歷時間的確切時刻寫入，而是在經歷時間間隔之後執行第一個 MQI 作業時寫入。 如果此值為 0，則會在連線中斷時 (或達到活動計數時) 寫入追蹤訊息。 預設值為 1。

ActivityCount

追蹤訊息之間的 MQI 作業數。 如果此值為 0，則會在連線中斷 (或活動間隔過了) 時寫入追蹤訊息。 預設值為 100。

TraceLevel

針對每一個作業所追蹤的參數詳細資料數量。 個別作業的說明詳細說明針對每一個追蹤層次包括哪些參數。 設為 低、中或 高。 預設值為 MEDIAL。

TraceMessage 資料

針對 MQGET、MQPUT、MQPUT1 及回呼作業所追蹤的訊息資料量 (以位元組為單位)。 預設為 0。

StopOnGetTrace 訊息

可以設為 ON 或 OFF。 預設為 ON。

相關工作

[使用 mqat.ini 配置活動追蹤行為](#)

配置分散式佇列



本節提供 IBM 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

開始之前

在閱讀本節之前，瞭解 [分散式佇列及叢集中引進的通道、佇列及其他概念](#) 會很有幫助。

如果您需要連接兩個位於不同實體網路上的佇列管理程式，或需要透過防火牆進行通訊，則使用 IBM MQ Internet Pass-Thru 可能會簡化配置。如需相關資訊，請參閱 [IBM MQ Internet Pass-Thru](#)。

程序

- 使用下列子主題中的資訊，以使用分散式佇列來連接應用程式：
 - [第 148 頁的『IBM MQ 分散式佇列技術』](#)
 - [第 165 頁的『分散式佇列管理簡介』](#)
 - [第 168 頁的『如何將訊息傳送至另一個佇列管理程式』](#)
 - [第 186 頁的『觸發通道』](#)
 - [第 184 頁的『訊息安全』](#)
 -  [第 192 頁的『在 UNIX, Linux, and Windows 上監視及控制通道』](#)
 -  [第 213 頁的『在 IBM i 上監視及控制通道』](#)

相關概念

[第 697 頁的『設定 IBM MQ for z/OS』](#)
請使用本主題作為逐步手冊，以自訂 IBM MQ for z/OS 系統。

相關工作

[第 13 頁的『配置用戶端與伺服器之間的連線』](#)
若要配置 IBM MQ MQI clients 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 232 頁的『配置佇列管理程式叢集』](#)
叢集提供一種機制，以簡化起始配置及進行中管理的方式來交互連接佇列管理程式。您可以定義叢集元件，以及建立和管理叢集。

[第 72 頁的『在 Multiplatforms 上的 .ini 檔中變更 IBM MQ 配置資訊』](#)
您可以編輯配置檔 (.ini) 中的資訊，來變更 IBM MQ 或個別佇列管理程式的行為，以符合安裝的需求。您也可以變更 IBM MQ MQI clients 的配置選項。





[第 693 頁的『在 z/OS 上配置佇列管理程式』](#)
使用這些指示，在「IBM MQ for z/OS」上配置佇列管理程式。

[第 759 頁的『在 z/OS 上設定與其他佇列管理程式的通訊』](#)
本節說明您在開始使用分散式佇列之前需要進行的 IBM MQ for z/OS 準備。

IBM MQ 分散式佇列技術

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：

-  [UNIX, Linux, and Windows 的訊息通道規劃範例](#)
-  [IBM i 的訊息通道規劃範例](#)
-  [z/OS 的訊息通道規劃範例](#)
-  [z/OS 使用佇列共用群組的訊息通道規劃範例](#)

相關概念

[通道](#)
[訊息佇列作業簡介](#)
[分散式佇列及叢集](#)

相關工作

第 147 頁的『[配置分散式佇列](#)』

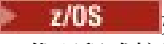
本節提供 IBM MQ 安裝之間交互通訊的更詳細資訊，包括佇列定義、通道定義、觸發及同步點程序

相關參考

[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 147 頁的『[配置分散式佇列](#)』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 149 頁的『[訊息流程控制 \(message flow control\)](#)』中所述。  如果您的佇列管理程式位於佇列共用群組中，且已啟用內部群組佇列作業 (IGQ)，則可以由 IGQ 代理程式控制訊息流程。這些代理程式在 [內部群組佇列作業](#) 中有說明。

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 [物件類型](#) 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 [分散式佇列及叢集](#)。

相關概念

第 149 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 150 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 150 頁的表 17 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	不適用
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	不適用
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	不適用
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 151 頁的圖 8 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。

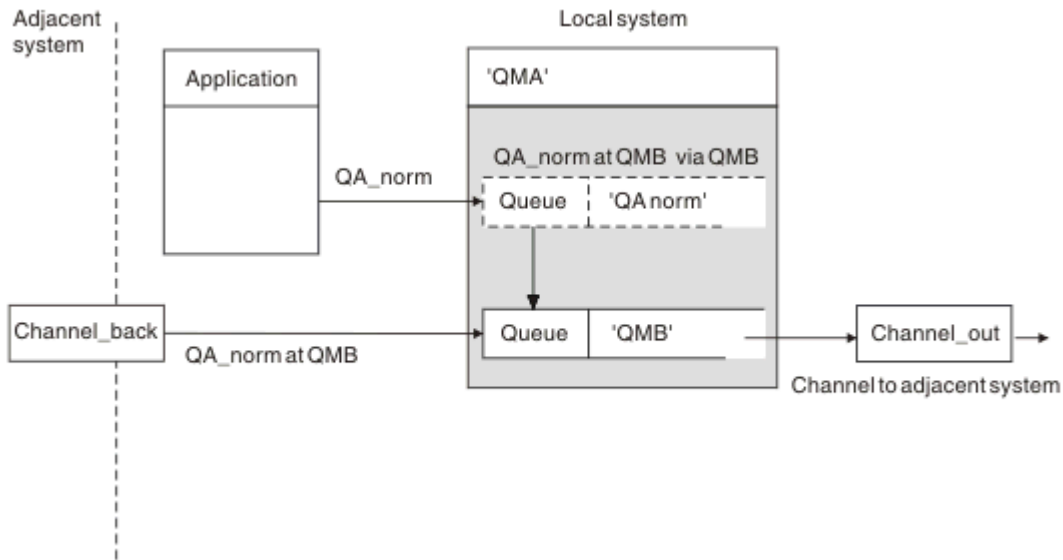


圖 8: 遠端佇列定義用來將佇列名稱解析成相鄰佇列管理程式的傳輸佇列

如果您是 IBM 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。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。

相關參考

[佇列名稱解析](#)

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。

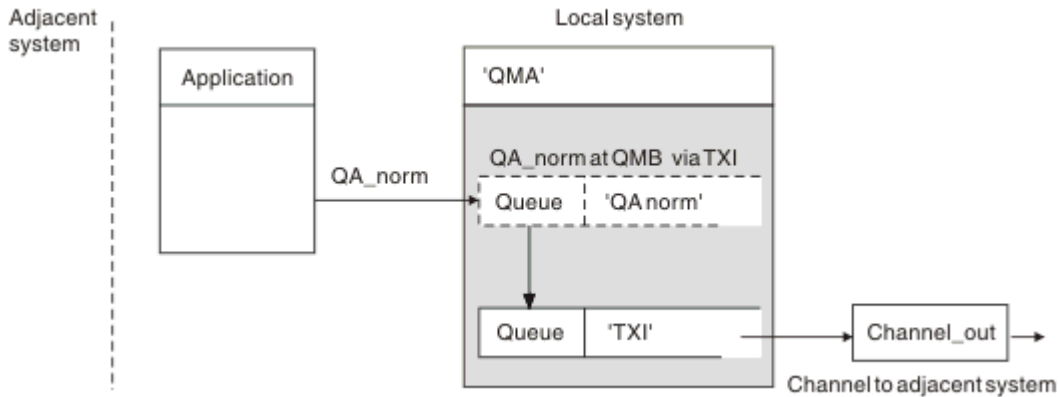


圖 9: 遠端佇列定義容許使用不同的傳輸佇列

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 151 頁的『將訊息放置在遠端佇列上』中的第 151 頁的圖 8 所示相同的系統配置。本主題中的第 152 頁的圖 9 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 152 頁的圖 9 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 233 頁的『定義叢集佇列』。

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。

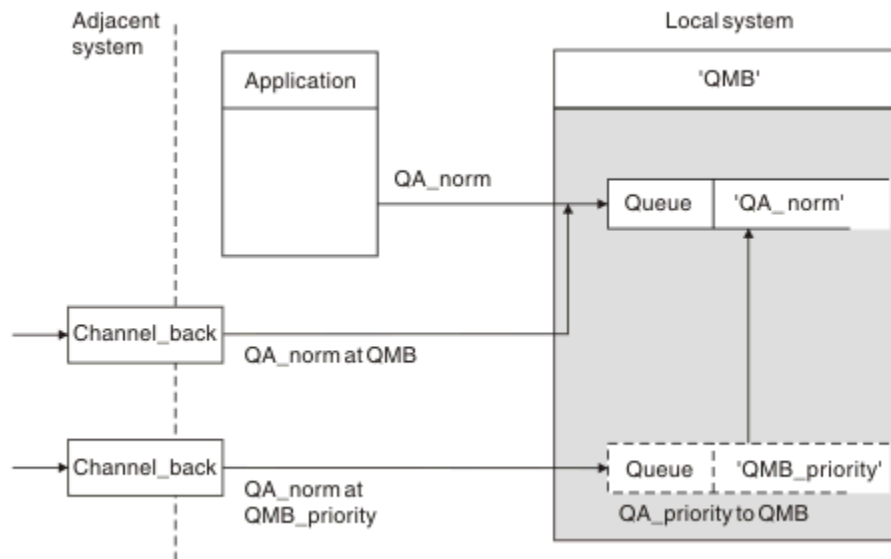


圖 10: 直接接收訊息，並解析別名佇列管理程式名稱

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 152 頁的圖 10。

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。

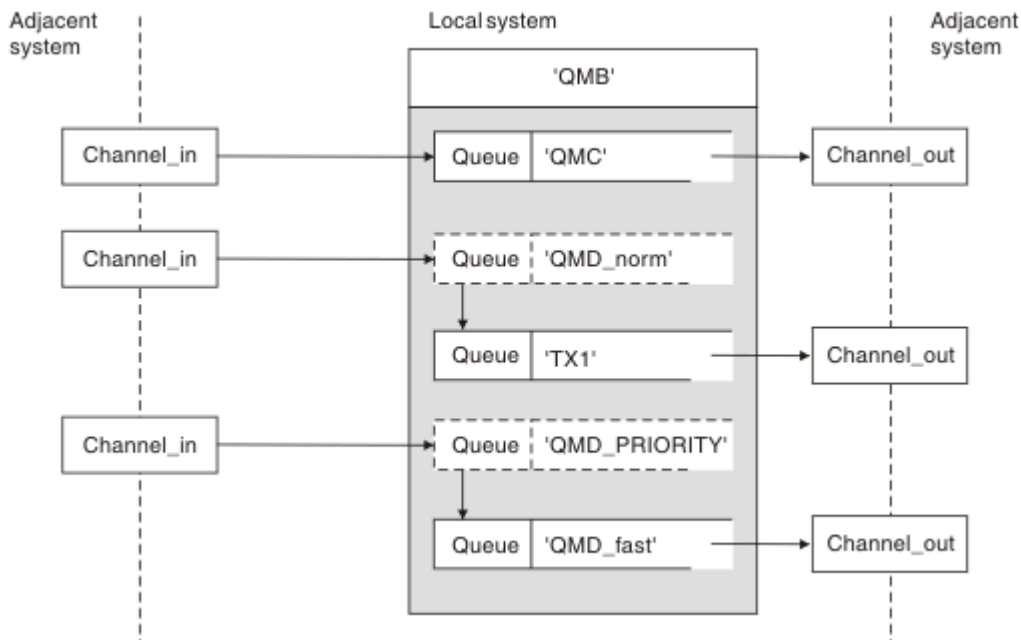


圖 11: 透過系統傳遞訊息的三種方法

第 152 頁的『接收訊息』中的第 152 頁的圖 10 技術顯示如何擷取別名流程。第 153 頁的圖 11 說明透過結合先前說明的技術來建置網路的方式。

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 所設定的別名訊息流程，即指名的服務類別 QMD_norm。
- 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。

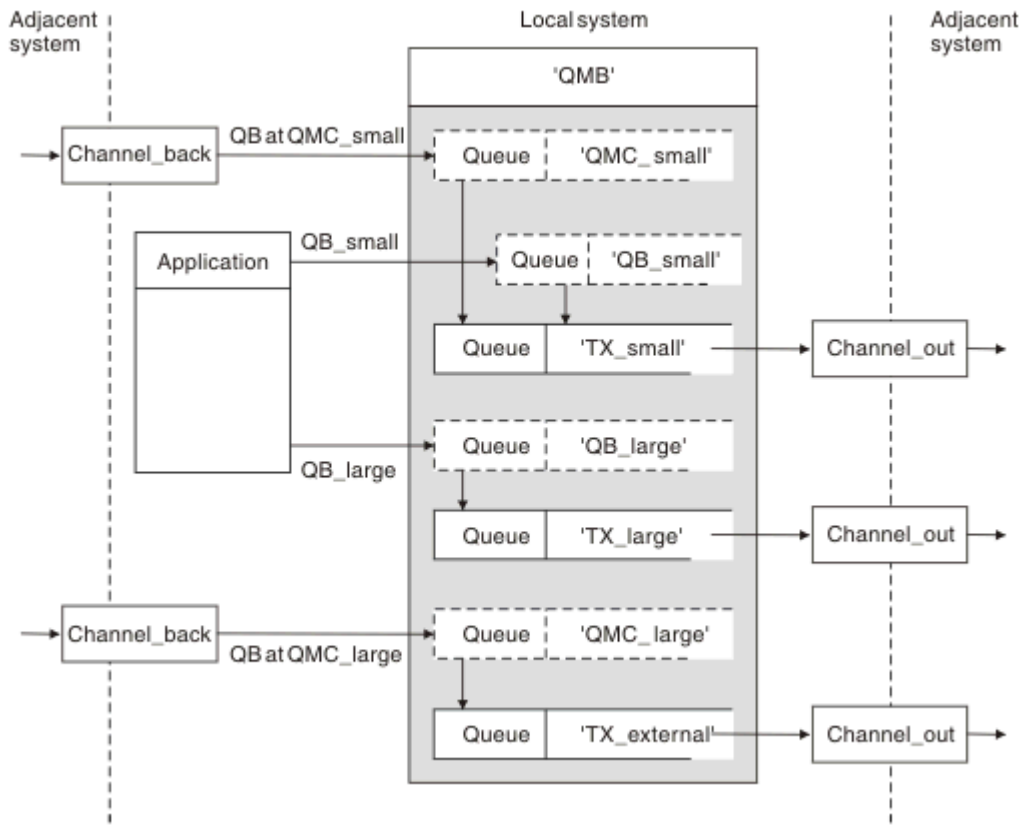


圖 12: 區隔訊息流程

在第 155 頁的圖 12 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。

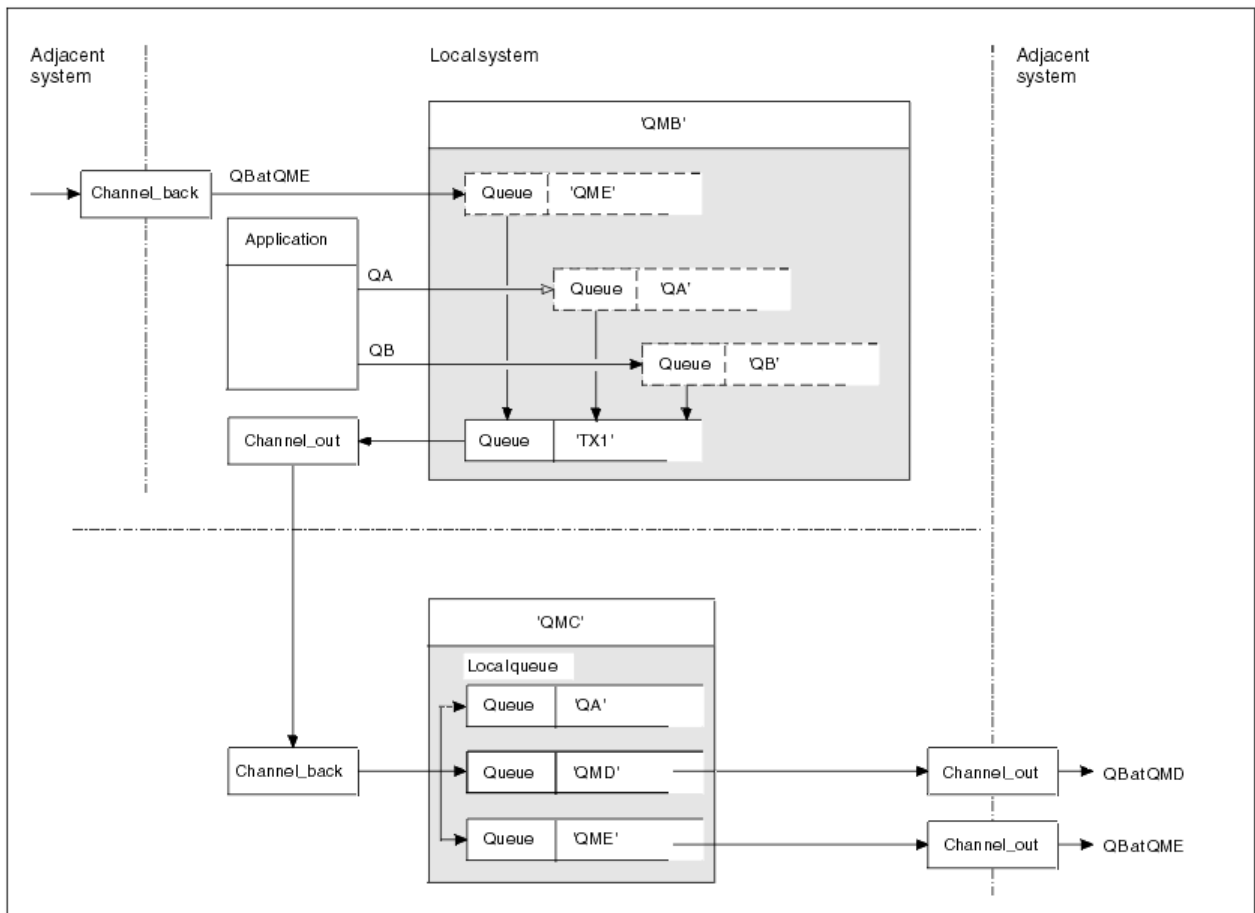


圖 13: 將訊息流程結合至通道

第 156 頁的圖 13 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。

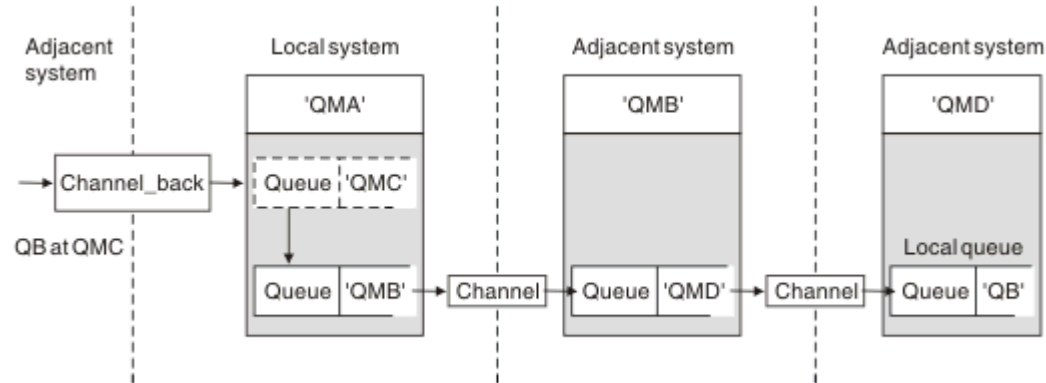


圖 14: 將訊息串流轉向另一個目的地

第 157 頁的圖 14 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 315 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 IBM 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。

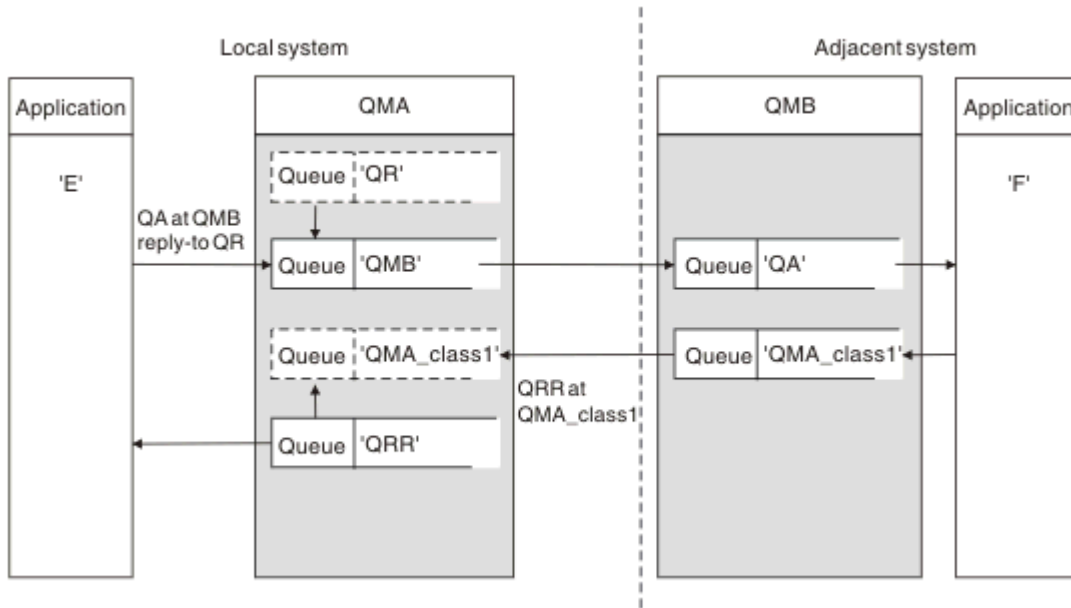


圖 15: PUT 呼叫期間的回覆目的地佇列名稱替代

第 158 頁的圖 15 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 164 頁的表 21 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 150 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 159 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 161 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 161 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 159 頁的圖 16 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。

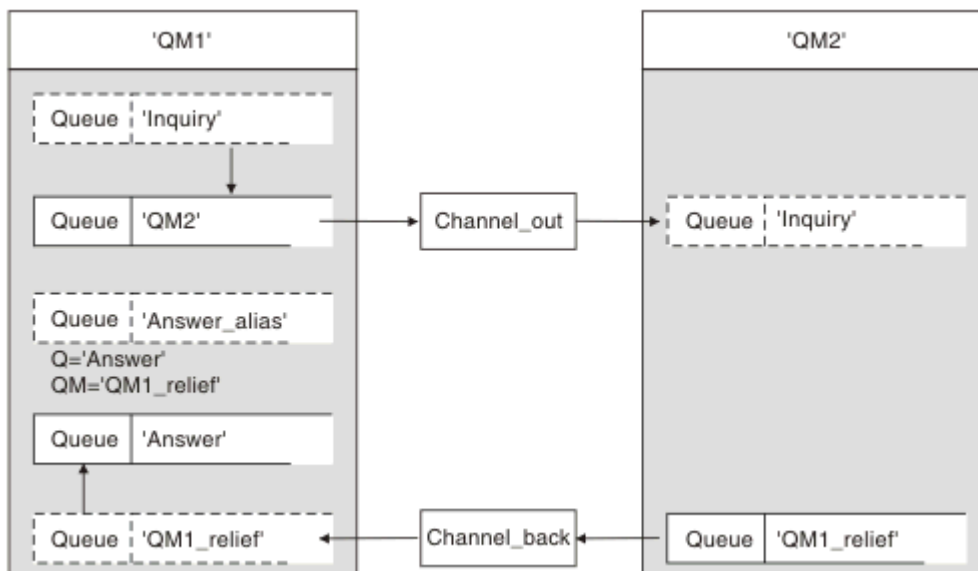


圖 16: 回覆目的地佇列別名範例

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

QM1 的 IBM 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 Inquiry
	遠端佇列管理程式名稱 QM2
	遠端佇列名稱 Inquiry
	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	Inquiry
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

位於 QM2 的 IBM MQ 系統管理者必須確保送入訊息的本端佇列存在，且正確命名的傳輸佇列可用於回覆訊息。

物件	定義
本端佇列	Inquiry
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 [叢集](#)。

 如果佇列管理程式正在佇列共用群組 (QSG) 中執行，且已啟用內部群組佇列作業 (IGQ)，您可以使用 SYSTEM.QSG.TRANSMIT.QUEUE。如需相關資訊，請參閱 [內部群組佇列作業](#)。

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 163 頁的表 19](#)、[第 163 頁的表 20](#) 和 [第 164 頁的表 21](#) 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)

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	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)
(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

通道會使用序號來檢查訊息的遞送順序是否與從傳輸佇列取得訊息的順序相同。

通道序號會在通道啟動時檢查，如果發生不符，表示通道任一端的持續同步化資料已遺失；例如，災難回復 (DR) 配置，或通道不確定時批次處理的結束已岔斷。

重設或忽略序號不符，請參閱 *qm.ini* 檔案的通道段落中的 **IgnoreSeqNumberMismatch**，不會有遺失或複製一批訊息的風險，也不會重設通道的不確定狀態。

可以使用 DISPLAY CHSTATUS 來顯示此資訊。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 "loopback" 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 IBM MQ 顯示路徑應用程式 (可透過控制指令 `dspmqrte` 取得)，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：

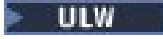


- 可讓您定義及控制佇列管理程式之間的通訊通道
- 提供訊息通道服務，可讓您將訊息從本端佇列類型 (稱為傳輸佇列) 移至本端系統上的通訊鏈結，以及從通訊鏈結移至目的地佇列管理程式上的本端佇列
- 提供您使用畫面、指令及程式來監視通道作業及診斷問題的機能

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。訊息傳送及接收是由稱為 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。

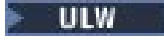



然後，MCA 由 DQM 本身控制。此結構與平台相關，但通常包括接聽器和觸發監視器，以及操作員指令和畫面。

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 192 頁的『[在 UNIX, Linux, and Windows 上監視及控制通道](#)』
-  第 761 頁的『[在 z/OS 上監視及控制通道](#)』
-  第 213 頁的『[在 IBM i 上監視及控制通道](#)』

如需訊息通道規劃範例，請參閱：

-  [UNIX, Linux, and Windows 的訊息通道規劃範例](#)
-  [IBM i 的訊息通道規劃範例](#)
-  [z/OS 的訊息通道規劃範例](#)
-  [z/OS 使用佇列共用群組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 166 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 173 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 184 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 188 頁的『起始設定和配置檔』](#)

通道起始設定資料的處理取決於 IBM MQ 平台。

[第 190 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時，IBM MQ 訊息可能需要資料轉換。

[第 190 頁的『撰寫您自己的訊息通道代理程式』](#)

IBM 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 190 頁的『分散式佇列管理要考量的其他事項』](#)

準備 IBM 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。

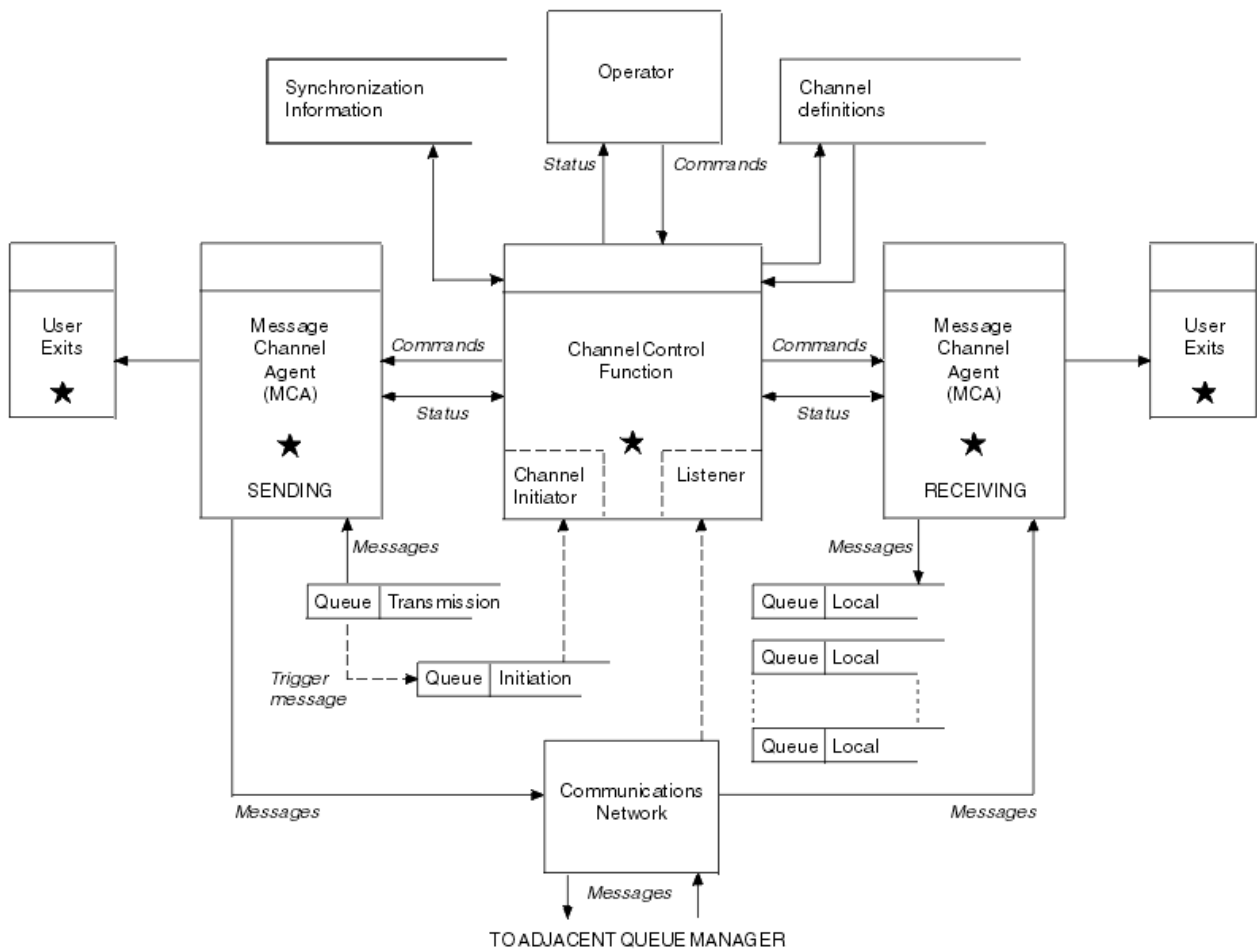


圖 17: 分散式佇列管理模型

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 IBM 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 ( 在 z/OS 上是必要項目，否則 是選用項目)
 - 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 IBM 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 `MQSC` 指令中說明的 IBM MQ Script 指令 (MQSC)、自動化管理作業中說明的可程式指令格式 (PCF) 指令，或「IBM MQ 探險家」。
-  在 z/OS 上，您也可以使用 管理 IBM MQ for z/OS 中說明的「作業」及「控制」畫面。
-  在 IBM i 上，您也可以使用畫面介面。

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 148 頁的『IBM MQ 分散式佇列技術』](#)

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 165 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 186 頁的『觸發通道』](#)

IBM MQ 提供在符合佇列上的特定條件時自動啟動應用程式的機能。此機能稱為觸發。

[第 184 頁的『訊息安全』](#)

除了 IBM 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。

相關工作

第 7 頁的『[在 Multiplatforms 上建立佇列管理程式](#)』

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。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 MQ 物件。

第 192 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 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 213 頁的『[在 IBM i 上監視及控制通道](#)』

使用 DQM 指令和畫面來建立、監視及控制遠端佇列管理程式的通道。每一個佇列管理程式都有一個 DQM 程式，用來控制與相容遠端佇列管理程式的交互連線。

第 13 頁的『[配置用戶端與伺服器之間的連線](#)』

若要配置 IBM MQ MQI clients 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 232 頁的『[配置佇列管理程式叢集](#)』

叢集提供一種機制，以簡化起始配置及進行中管理的方式來交互連接佇列管理程式。您可以定義叢集元件，以及建立和管理叢集。

第 759 頁的『[在 z/OS 上設定與其他佇列管理程式的通訊](#)』

本節說明您在開始使用分散式佇列之前需要進行的 IBM MQ for z/OS 準備。

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 IBM 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 IBM 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。

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。

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。

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

 此為必要項目。您必須使用稱為 SYSTEM.CHANNEL.INITQ。

 這是選用項目。請考量命名起始佇列 SYSTEM.CHANNEL.INITQ。

- 傳輸佇列定義

USAGE 屬性設為 XMITQ 的本端佇列。  如果您是使用 IBM MQ for IBM i 原生介面，則 USAGE 屬性為 *TMQ。

- 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 170 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。

[第 171 頁的『在 IBM i 上建立傳輸佇列』](#)

您可以使用「建立 MQM 佇列」畫面，在 IBM i 平台上建立傳輸佇列。

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 IBM 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。

在 IBM i 上建立傳輸佇列

您可以使用「建立 MQM 佇列」畫面，在 IBM i 平台上建立傳輸佇列。

您必須針對每一個傳送訊息通道，將「用法」欄位屬性設為 *TMQ 來定義本端佇列。

如果您要使用遠端佇列定義，請使用相同的指令來建立類型 *RMT 的佇列，並使用 *NORMAL。

若要建立傳輸佇列，請從指令行使用 CRTMQMQ 指令來呈現第一個佇列建立畫面；請參閱 [第 171 頁的圖 18](#)。

```
Create MQM Queue (CRTMQMQ)
Type choices, press Enter.
Queue name . . . . .
Queue type . . . . . ____ *ALS, *LCL, *MDL, *RMT
Message Queue Manager name . . . *DFT_____
-----

Bottom
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys
+
```

圖 18: 建立佇列 (1)

鍵入佇列名稱，並指定您要建立的佇列類型：本端、遠端或別名。若為傳輸佇列，請在此畫面上指定「本端 (*LCL)」，然後按 Enter 鍵。

即會呈現「建立 MQM 佇列」畫面的第二頁；請參閱 [第 171 頁的圖 19](#)。

```
Create MQM Queue (CRTMQMQ)
Type choices, press Enter.
Queue name . . . . . > HURS.2.HURS.PRIORIT
Queue type . . . . . > *LCL *ALS, *LCL, *MDL, *RMT
Message Queue Manager name . . . *DFT
Replace . . . . . *NO *NO, *YES
Text 'description' . . . . .
Put enabled . . . . . *YES *SYSDFTQ, *NO, *YES
Default message priority . . . . 0 0-9, *SYSDFTQ
Default message persistence . . *NO *SYSDFTQ, *NO, *YES
Process name . . . . .
Triggering enabled . . . . . *NO *SYSDFTQ, *NO, *YES
Get enabled . . . . . *YES *SYSDFTQ, *NO, *YES
Sharing enabled . . . . . *YES *SYSDFTQ, *NO, *YES

More...
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys
```

圖 19: 建立佇列 (2)

變更所顯示的任何預設值。向下按頁面以捲動至下一個畫面；請參閱 [第 172 頁的圖 20](#)。

```

Create MQM Queue (CRTMQMQ)

Type choices, press Enter.

Default share option . . . . . *YES      *SYSDFTQ, *NO, *YES
Message delivery sequence . . . *PTY      *SYSDFTQ, *PTY, *FIFO
Harden backout count . . . . . *NO      *SYSDFTQ, *NO, *YES
Trigger type . . . . . *FIRST      *SYSDFTQ, *FIRST, *ALL...
Trigger depth . . . . . 1          1-999999999, *SYSDFTQ
Trigger message priority . . . . 0          0-9, *SYSDFTQ
Trigger data . . . . . '          '
Retention interval . . . . . 999999999 0-999999999, *SYSDFTQ
Maximum queue depth . . . . . 5000     1-24000, *SYSDFTQ
Maximum message length . . . . . 4194304 0-4194304, *SYSDFTQ
Backout threshold . . . . . 0          0-999999999, *SYSDFTQ
Backout requeue queue . . . . . '          '
Initiation queue . . . . . '          '

More...
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

```

圖 20: 建立佇列 (3)

針對傳輸佇列，在此畫面的「使用情形」欄位中鍵入 *TMQ，並變更其他欄位中顯示的任何預設值。

```

Create MQM Queue (CRTMQMQ)

Type choices, press Enter.

Usage . . . . . *TMQ      *SYSDFTQ, *NORMAL, *TMQ
Queue depth high threshold . . . 80      0-100, *SYSDFTQ
Queue depth low threshold . . . 20      0-100, *SYSDFTQ
Queue full events enabled . . . *YES      *SYSDFTQ, *NO, *YES
Queue high events enabled . . . *YES      *SYSDFTQ, *NO, *YES
Queue low events enabled . . . *YES      *SYSDFTQ, *NO, *YES
Service interval . . . . . 999999999 0-999999999, *SYSDFTQ
Service interval events . . . . *NONE      *SYSDFTQ, *HIGH, *OK, *NONE
Distribution list support . . . *NO      *SYSDFTQ, *NO, *YES
Cluster Name . . . . . *SYSDFTQ
Cluster Name List . . . . . *SYSDFTQ
Default Binding . . . . . *SYSDFTQ *SYSDFTQ, *OPEN, *NOTFIXED

Bottom
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

```

圖 21: 建立佇列 (4)

當您對欄位包含正確資料感到滿意時，請按 Enter 鍵以建立佇列。

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 192 頁的『[在 UNIX, Linux, and Windows 上監視及控制通道](#)』中詳述的那些 IBM 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

相關概念

[通道的問題判斷](#)

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 IBM 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 IBM 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 147 頁的『配置分散式佇列』](#) 中說明的實務範例。

如需定義 MQI 通道的相關資訊，請參閱 [第 25 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 183 頁的『處理不確定的通道』](#)。

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 175 頁的圖 22](#) 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 176 頁的圖 23](#) 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。

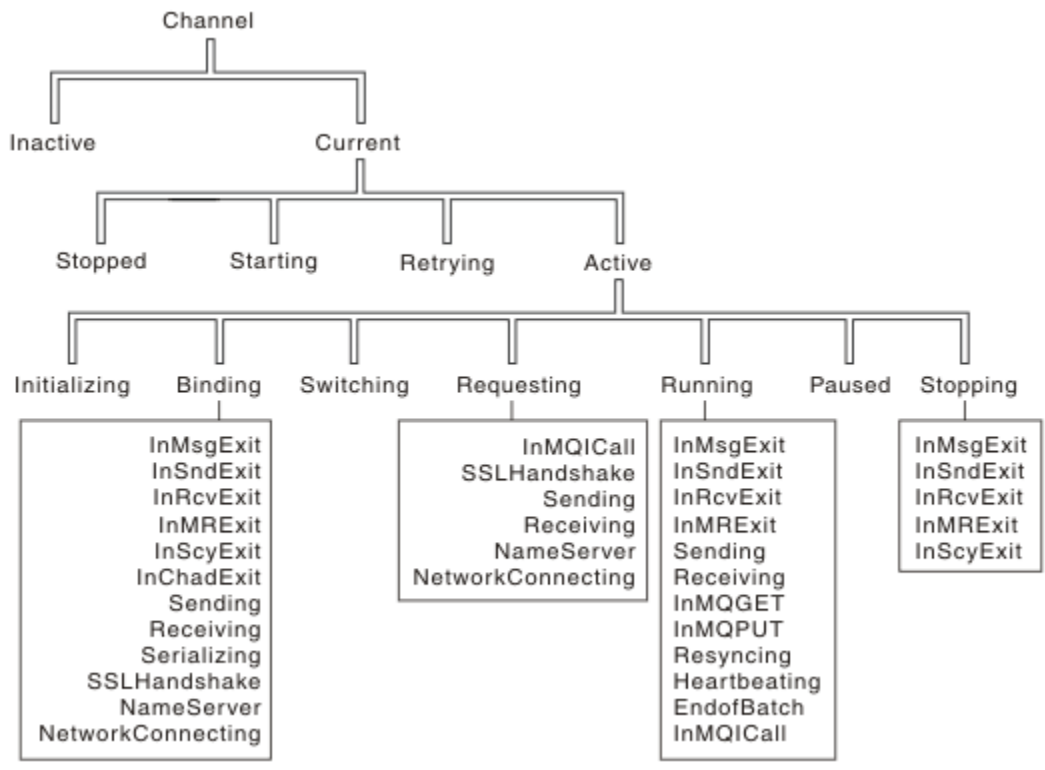


圖 22: 通道狀態及子狀態

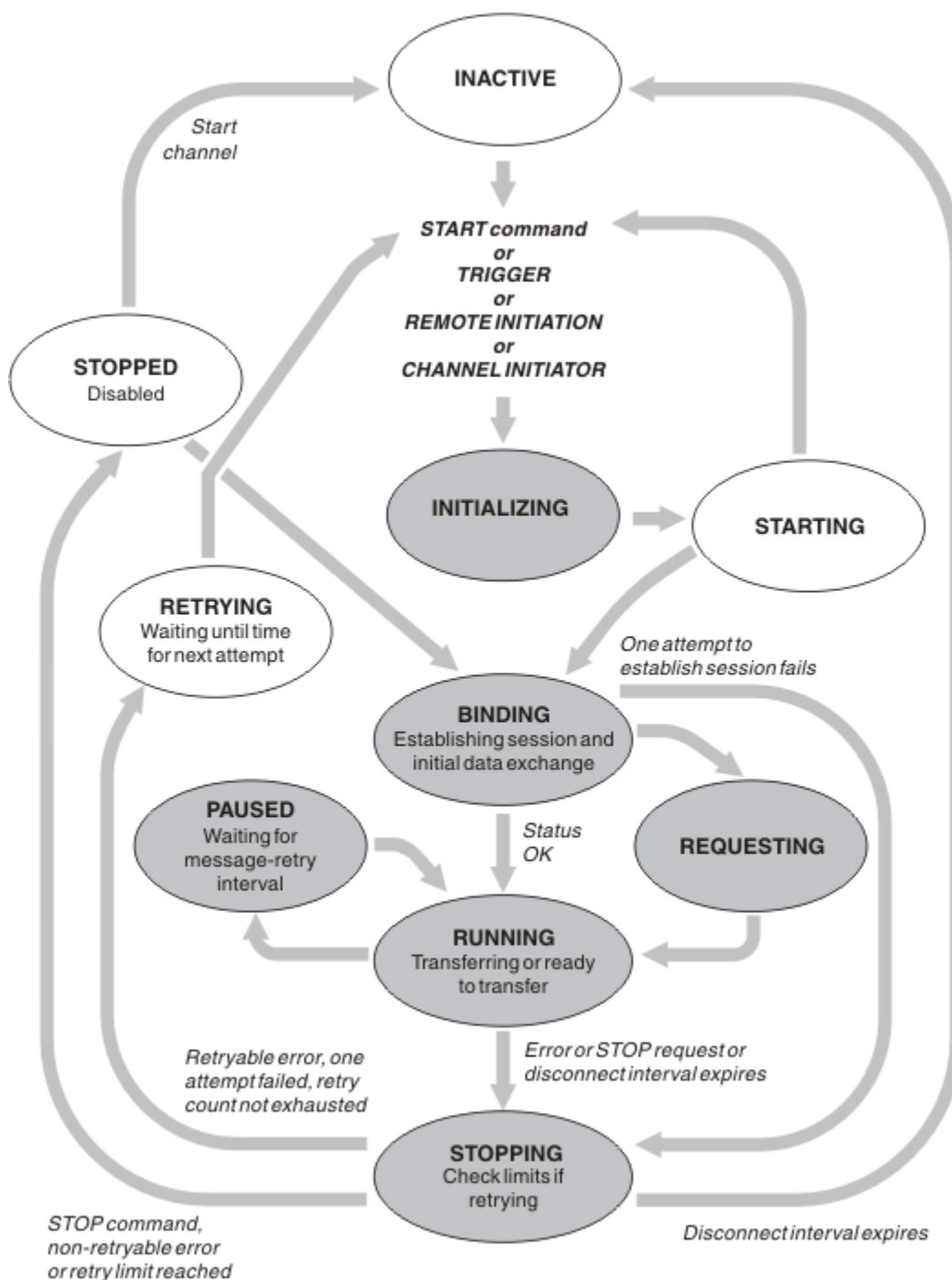


圖 23: 通道狀態之間的流程

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 176 頁的圖 23](#) 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 175 頁的圖 22](#) 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。

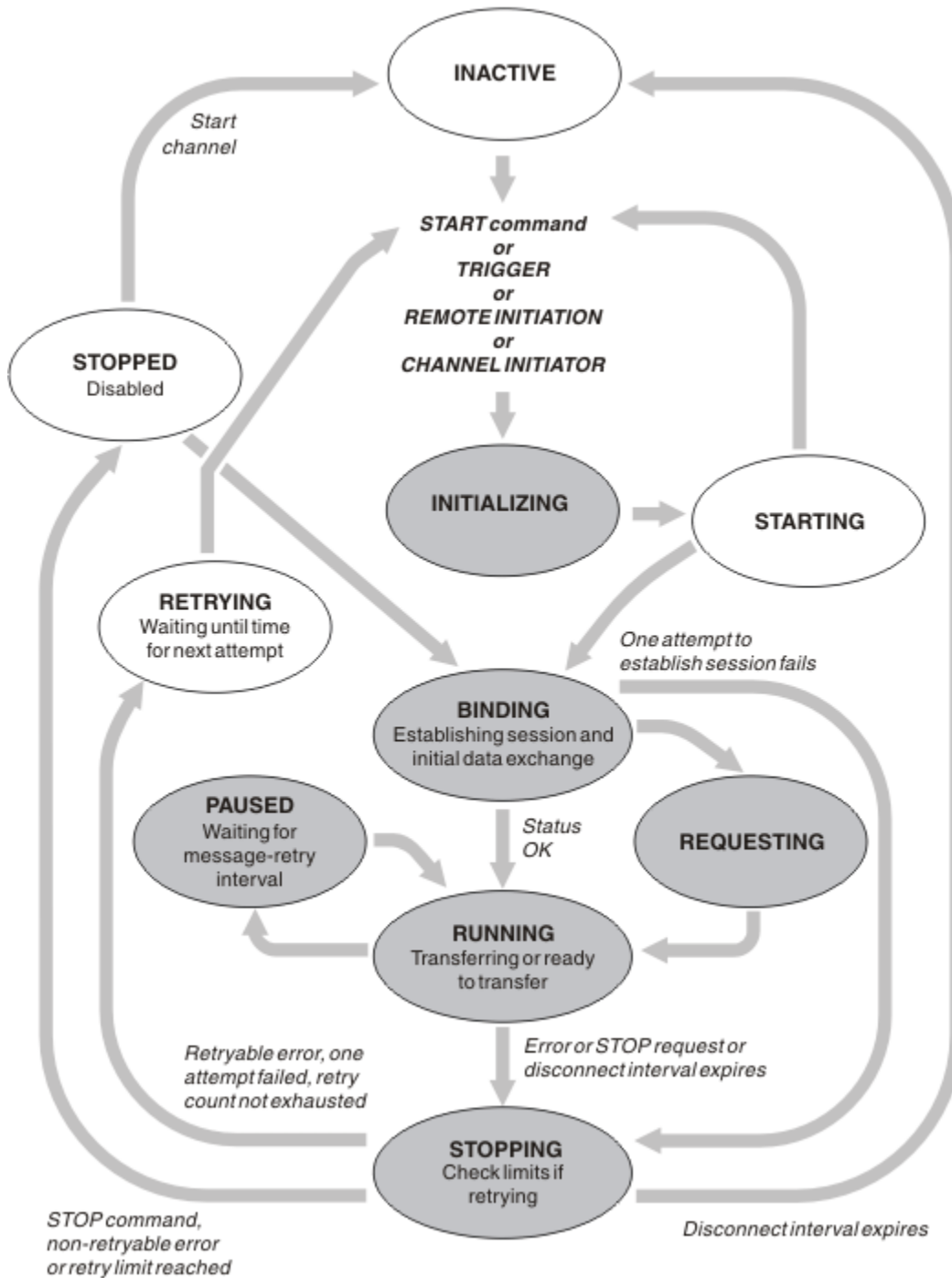


圖 24: 通道狀態之間的流程

註:

1. 當通道處於 第 177 頁的圖 24 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。針對您的平台指定此項目：

-  使用 ALTER QMGR MAXCHL 指令。
-  編輯佇列管理程式起始設定檔。
-   編輯佇列管理程式配置檔。
- 使用 IBM MQ Explorer。

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](#)。如需指定通道數上限的相關資訊，請參閱下列主題：

-  [管理 IBM MQ。](#)
-  [管理 IBM MQ for IBM i。](#)
-  [管理 IBM MQ for z/OS。](#)

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  或 z/OS 主控台，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上限的相關資訊，請參閱下列主題：

-  [管理 IBM MQ。](#)
-  [管理 IBM MQ for IBM i。](#)
-  [管理 IBM MQ for z/OS。](#)

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 182 頁的『重新啟動已停止的通道』](#))。

註: 對於 **IBM i**、IBM i、UNIX, Linux, and 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。

z/OS 如果通道起始程式在通道處於 RETRYING 或 STOPPED 狀態時停止，則會在通道起始程式重新啟動時記住通道狀態。不過，如果通道起始程式在通道處於 STOPPED 狀態時停止，則會重設 SVRCONN 通道類型的通道狀態。

Multi 如果在通道處於 RETRYING 或 STOPPED 狀態時停止佇列管理程式，則會在重新啟動佇列管理程式時記住通道狀態。從 IBM MQ 8.0 開始，這也適用於 SVRCONN 通道。先前，如果通道起始程式在通道處於 STOPPED 狀態時停止，則會重設 SVRCONN 通道類型的通道狀態。

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使用 **MAXINST** 參數耗盡佇列管理程式通道資源，並防止單一用戶端應用程式使用 **MAXINSTC** 參數耗盡伺服器連線通道容量。

您可以使用 **DEFINE CHANNEL** 指令來設定 **MAXINST** 和 **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。

相關參考

[通道屬性及通道類型](#)

[定義通道](#)

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 IBM MQ for z/OS 中，如果您使用 TCP/IP 作為傳輸通訊協定，也可以指定 **Keepalive** 間隔通道屬性 (KAINT) 的值。建議您為 **Keepalive** 間隔提供高於活動訊號間隔的值，以及小於斷線值的值。您可以使用這個屬性來指定每一個通道的逾時值，如 [保持作用中間隔 \(KAINT\)](#)中所述。

在 IBM MQ for IBM i UNIX, Linux, and 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 IBM MQ for IBM i, UNIX, Linux, and 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。

在 IBM MQ for z/OS 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. 如果設定 RCVTIME，則會將逾時設為下列其中一項：
 - 已協議的 HBINT 乘以常數
 - 協議的 HBINT 加上固定秒數
 - 固定秒數

視 RCVTTYPE 參數而定，並遵循 RCVTMIN 所強制的任何限制 (如果適用的話)。當配置 RCVTTYPE (EQUAL) 時，RCVTMIN 不適用。如果您使用常數值 RCVTIME，且使用活動訊號間隔，請不要指定小於活動訊號間隔的 RCVTIME。如需 RCVTIME、RCVTMIN 及 RCVTTYPE 屬性的詳細資料，請參閱 [ALTER QMGR](#) 指令。

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 IBM WebSphere MQ 7 用戶端通道中，活動訊號可以同時從伺服器及用戶端傳送。任一端的逾時基於 $2 * HBINT$ (對於少於 60 秒的 HBINT)，以及 $HBINT + 60$ (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。

建議設定

IBM MQ for z/OS

作為起始起點，您可以使用：

```
/cpl ALTER QMGR TCPKEEP(YES) RCVTTYPE(ADD) RCVTIME(60) ADOPTMCA(ALL) ADOPTCHK(ALL)
```

其中 cpl 是佇列管理程式子系統的指令字首。

如需各種參數的相關資訊，請參閱 [ALTER QMGR](#) 及 [IBM MQ 網路可用性](#)。

如果傳送端的 IP 位址可以轉換為多個位址，則您可能需要將 ADOPTCHK 設為 QMNAME 而非 ALL。

IBM MQ for Multiplatforms

在 qm.ini 中，新增下列資訊：

```
TCP:
KeepAlive=Yes
CHANNELS:
AdoptNewMCA=ALL
AdoptNewMCACheck=ALL
```

如需相關資訊，請參閱 [ALTER QMGR](#)、[分散式佇列的配置檔段落](#) 及 [第 95 頁的『qm.ini 檔案的通道段落』](#)。

如果寄件者的 IP 位址可以轉換為多個位址，則您可能需要將 **AdoptNewMCACheck** 設為 QMNAME，而不是 ALL。

採用 MCA

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

如果通道失去聯絡，則接收端通道可以保持「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，則不容許另一個版本的相同接收端通道啟動。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 MQ 取消接收端通道，並在其位置啟動新的通道。

相關工作

[管理 IBM MQ](#)

[管理 IBM MQ for z/OS](#)

[管理 IBM MQ for IBM i](#)

停止及靜止通道

您可以在斷線時間間隔到期之前停止並靜止通道。

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 MQ 檔案總管
-   其他平台專用機制，如下所示：

 若為 **z/OS**：
「停止通道」畫面

 若為 **IBM i**：
ENDMQMCHL CL 指令或 WRKMQMCHL 畫面上的 END 選項

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。

 在 IBM MQ for z/OS 上，FORCE 會岔斷任何進行中的訊息重新配置，這可能會導致 BIND_NOT_FIXED 訊息局部重新配置或不正常。

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。

 在 IBM MQ for z/OS 上，TERMINATE 會岔斷進行中的任何訊息重新配置，這可能會導致 BIND_NOT_FIXED 訊息局部重新配置或不正常。

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放以供專用輸入，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，因為伺服器仍開放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。

關於這項作業

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。

z/OS 如果通道起始程式在通道處於 RETRYING 或 STOPPED 狀態時停止，則會在通道起始程式重新啟動時記住通道狀態。不過，如果通道起始程式在通道處於 STOPPED 狀態時停止，則會重設 SVRCONN 通道類型的通道狀態。

Multi 如果在通道處於 RETRYING 或 STOPPED 狀態時停止佇列管理程式，則會在重新啟動佇列管理程式時記住通道狀態。從 IBM MQ 8.0 開始，這也適用於 SVRCONN 通道。先前，如果通道起始程式在通道處於 STOPPED 狀態時停止，則會重設 SVRCONN 通道類型的通道狀態。

程序

- 以下列其中一種方式重新啟動通道：
 - 使用 `START CHANNEL MQSC` 指令。
 - 使用 啟動通道 PCF 指令。
 - 使用 [IBM MQ Explorer](#)
 - **z/OS** 在 z/OS 上，使用 [啟動通道畫面](#)。
 - **IBM i** 在 IBM i 上，使用 `STRMQMCHL CL` 指令或 [WRKMQMCHL](#) 畫面上的 START 選項。

處理不確定的通道

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。術語 手動 包括使用包含 IBM MQ 系統管理指令的操作員或程式。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 CURLWID
```

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 IBM 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)
```

IBM i

在 IBM MQ for IBM i 上，您可以使用「解析 MQM 通道」指令 RSVMQMCHL。

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。

訊息安全

除了 IBM 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 184 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 185 頁的圖 25](#) 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。）

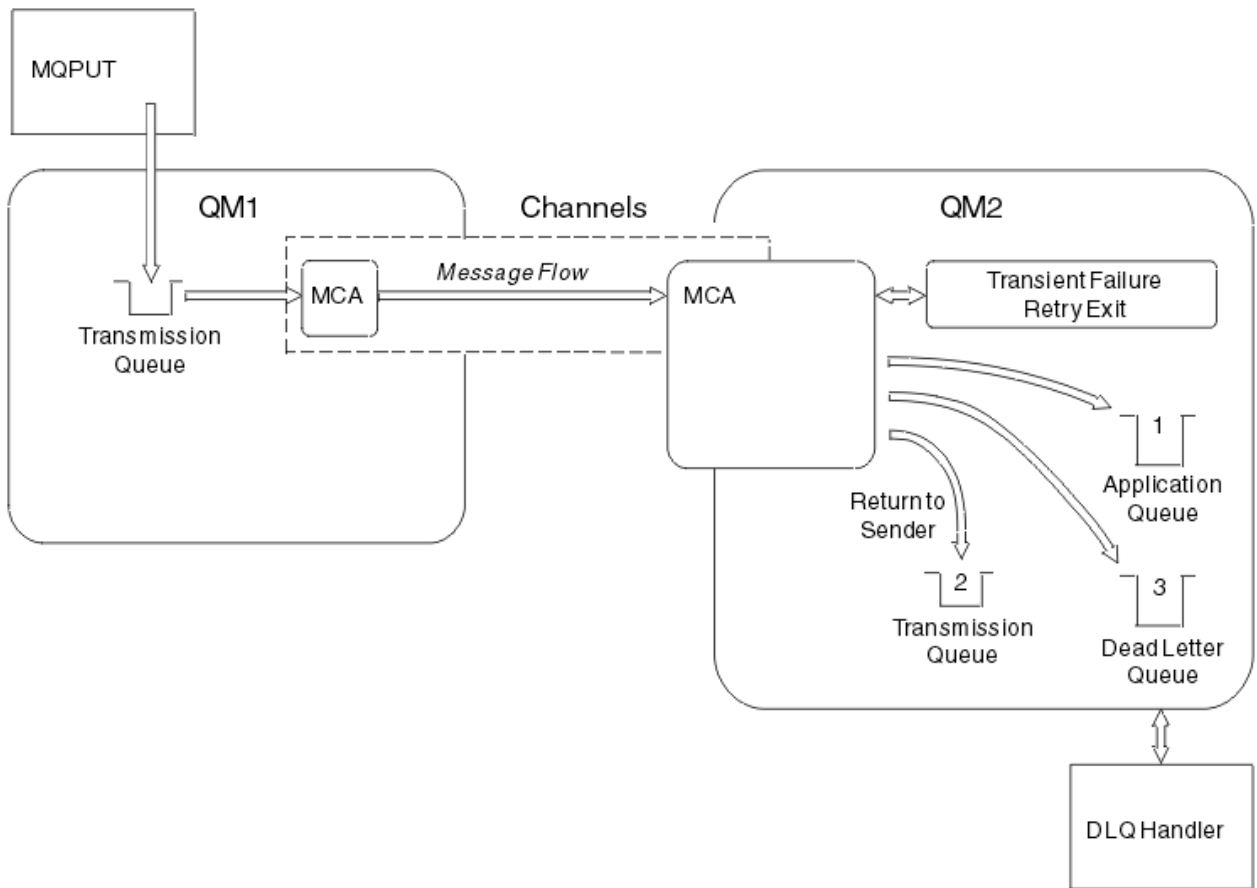


圖 25: 無法遞送訊息時會發生什麼情況

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。IBM MQ for UNIX、Linux 及 Windows 系統的處理無法傳送郵件的佇列上的訊息，以及 z/OS 系統的無法傳送郵件的佇列處理程式公用程式 (CSQDLQH) 中說明此處理程序。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 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

IBM MQ 提供在符合佇列上的特定條件時自動啟動應用程式的機能。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 IBM 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 187 頁的『在 UNIX, Linux, and Windows 上觸發通道。』](#)
- **IBM i** 若為 IBM i，請參閱 [第 187 頁的『IBM MQ for IBM i 中的觸發通道』](#)
- **z/OS** 若為 z/OS，請參閱 [第 760 頁的『傳輸佇列及觸發通道』](#)

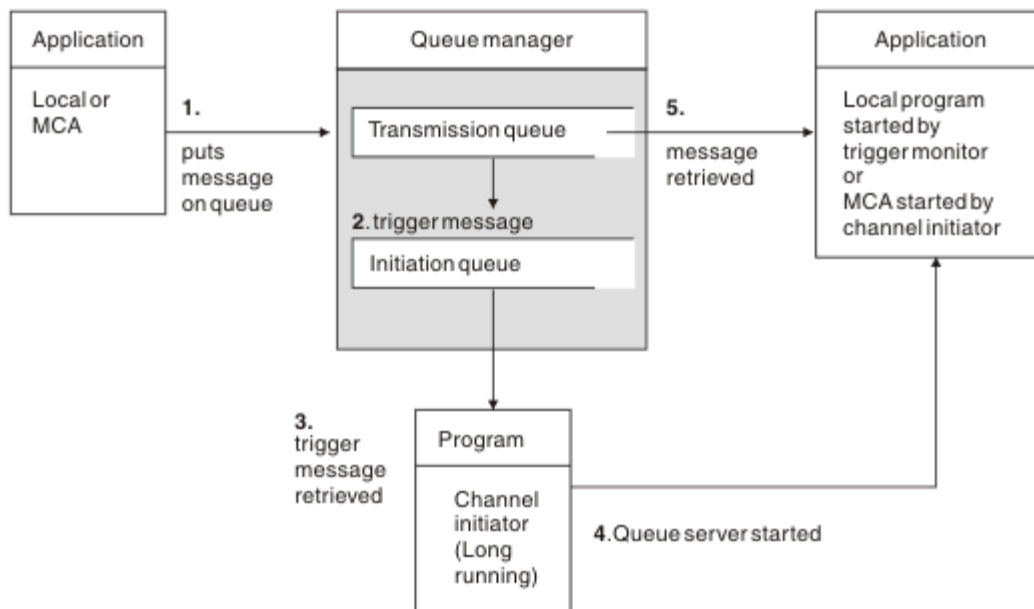


圖 26: 觸發的概念

第 186 頁的圖 26 中顯示觸發所需的物件。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。

z/OS 在 z/OS 上，起始佇列的名稱是固定的，因此不會在啟動指令上使用。

- 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。IBM MQ 若為 **IBM i**、UNIX, Linux, and Windows 系統，容許將通道名稱指定為空白，在此情況下，會使用具有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啟用觸發。

註:

1. 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。
2. 起始佇列及觸發程序可用來觸發任意數目的通道。
3. 可以定義任何數目的起始佇列及觸發程序。
4. 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, and Windows 上觸發通道。



您可以在 IBM 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 IBM 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，不需要程序定義。

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。

IBM MQ for IBM i 中的觸發通道



在 IBM MQ for IBM i 中觸發通道是使用通道起始程式處理程序來實作。起始佇列 SYSTEM.CHANNEL.INITQ，否則它會隨佇列管理程式自動啟動。

指定 SYSTEM.CHANNEL.INITQ 作為起始佇列，並啟用佇列的觸發。通道起始程式會啟動指定此傳輸佇列的第一個可用通道。

```
CRTMQMQ QNAME(MYXMITQ1) QTYPE(*LCL) MQMNAME(MYQMGR)
TRGENBL(*YES) INITQNAME(SYSTEM.CHANNEL.INITQ)
USAGE(*TMQ)
```

您可以使用 STRMQMCHLI 指令手動啟動最多三個通道起始程式處理程序，並指定不同的起始佇列。您也可以指定多個能夠處理傳輸佇列的通道，並選擇要啟動的通道。此功能仍提供為與舊版相容。其用法已淘汰。

註：一次只能有一個通道可以處理傳輸佇列。

```
STRMQMCHLI QNAME(MYINITQ)
```

設定通道的傳輸佇列，指定 TRGENBL(*YES)，並在 TRIGDATA 欄位中指定通道名稱，以選擇要嘗試啟動的通道。例如：

```
CRTMQMQ QNAME(MYXMITQ2) QTYPE(*LCL) MQMNAME(MYQMGR)  
TRGENBL(*YES) INITQNAME(MYINITQ)  
USAGE(*TMQ) TRIGDATA(MYCHANNEL)
```

相關概念

第 188 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。

相關工作

第 147 頁的『[配置分散式佇列](#)』

本節提供 IBM 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, and Windows 上的通道程式](#)

 [IBM i 上的交互通訊工作](#)

 [IBM i 上的通道狀態](#)

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 IBM MQ for Multiplatforms 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 來變更此項目。

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 [IBM 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 IBM MQ 平台。

z/OS 系統

z/OS

在 IBM MQ for z/OS 中，使用 **ALTER QMGR MQSC** 指令指定起始設定及配置資訊。如果您將 **ALTER QMGR** 指令放置在 CSQINP2 起始設定輸入資料集中，則每次啟動佇列管理程式時都會處理這些指令。

若要在每次啟動通道起始程式時執行 MQSC 指令 (例如 **START LISTENER**)，請將它們放入 CSQINPX 起始設定輸入資料集，並在通道起始程式啟動作業程序中指定選用的 DD 陳述式 CSQINPX。

如需 CSQINP2 及 CSQINPX 的相關資訊，請參閱 [自訂起始設定輸入資料集](#) 及 [ALTER QMGR](#)。

Windows、IBM i、UNIX and Linux 系統

在 IBM MQ for Windows、IBM i、UNIX and Linux 系統中，有一些配置檔用來保留 IBM MQ 安裝架構的基本配置資訊。

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。

IBM MQ 配置檔

此檔案保留與 IBM MQ 系統上所有佇列管理程式相關的資訊。該檔案稱為 `mqs.ini`。它在 [第 74 頁的『IBM MQ 配置檔 mqs.ini』](#) 中說明。

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 IBM 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。

IBM i

若為 IBM i：

```
/QIBM/UserData/mqm/qmgrs/QMNAME/qm.ini
```

若為 Windows 系統:

```
C:\ProgramData\IBM\MQ\mqgrs\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時，IBM MQ 訊息可能需要資料轉換。

IBM 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

IBM 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。

您可能想要撰寫自己的 MCA 程式，讓 IBM MQ 透過您自己的專有通訊協定交互作業，或透過 IBM 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 IBM MQ 提供的 MCA 交互作業。)

如果您決定使用 IBM 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 IBM 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。如果是這樣，您可以使用 IBM 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 IBM 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。

Linux **UNIX** 在 UNIX and Linux 系統上，隨 IBM MQ 提供 DLQ 處理程式；如需相關資訊，請參閱 [範例 DLQ 處理程式 amqsdlq](#)。

IBM i 如需 IBM MQ for IBM i 的相關資訊，請參閱 [IBM MQ for IBM i 無法傳送郵件的佇列處理程式](#)。

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

IBM i 在 IBM MQ for IBM i 上，您可以指定系統中容許的通道數上限，以及一次可以處於作用中的通道數上限。您可以在 `qm.ini` 檔案的 `QIBM/UserData/mqm/qmgrs/queue_manager_name` 目錄中指定這些號碼。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 IBM MQ 一起提供，但可以由每一個安裝根據本端需求提供。

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
STANDARD	未定義	STANDARD
Fastpath	未定義	Fastpath
STANDARD	STANDARD	STANDARD
Fastpath	STANDARD	STANDARD
STANDARD	Fastpath	STANDARD
Fastpath	Fastpath	Fastpath
STANDARD	用戶端	用戶端
Fastpath	用戶端	STANDARD
STANDARD	LOCAL	STANDARD
Fastpath	LOCAL	STANDARD

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 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。

關於這項作業

您可以使用下列類型的指令來控制通道：

IBM MQ 指令 (MQSC)

在 UNIX, Linux, and Windows 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 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

您可以在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 [使用控制指令進行管理](#)。

可程式化指令格式指令

如需詳細資料，請參閱 [PCF 指令](#)。

IBM MQ Explorer

在 Linux 和 Windows 系統上，您可以使用 IBM 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 193 頁的表 22](#)。

程序


- [第 193 頁的『設定及控制通道所需的功能』](#)
- [第 195 頁的『開始使用物件』](#)
- [第 201 頁的『在 Windows 上設定通訊』](#)
- [第 207 頁的『在 UNIX and Linux 上設定通訊』](#)

相關工作

[第 213 頁的『在 IBM i 上監視及控制通道』](#)

使用 DQM 指令和畫面來建立、監視及控制遠端佇列管理程式的通道。每一個佇列管理程式都有一個 DQM 程式，用來控制與相容遠端佇列管理程式的交互連線。

相關參考

 [UNIX, Linux, and Windows 上的通道程式](#)

 [UNIX, Linux, and Windows 的訊息通道規劃範例](#)

[配置資訊範例](#)

[通道屬性](#)

ULW 設定及控制通道所需的功能

可能需要一些 IBM 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 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 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 195 頁的『開始使用物件』](#)
- [第 195 頁的『建立相關聯的物件』](#)
- [第 195 頁的『建立預設物件』](#)
- [第 196 頁的『建立頻道』](#)
- [第 196 頁的『顯示通道』](#)
- [第 197 頁的『顯示通道狀態』](#)
- [第 197 頁的『使用連線測試檢查鏈結』](#)
- [第 197 頁的『啟動通道』](#)
- [第 199 頁的『停止通道』](#)
- [第 200 頁的『重新命名通道』](#)
- [第 200 頁的『重設通道』](#)
- [第 200 頁的『解決通道上不確定的訊息』](#)

第 193 頁的表 22 顯示您可能需要的 IBM MQ 函數完整清單。

表 22: UNIX, Linux, and Windows 系統中所需的功能			
函數	控制指令	MQSC	IBM MQ Explorer 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			

表 22: UNIX, Linux, and Windows 系統中所需的機能 (繼續)

函數	控制指令	MQSC	IBM MQ Explorer 對等項目?
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
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

表 22: UNIX, Linux, and Windows 系統中所需的機能 (繼續)			
函數	控制指令	MQSC	IBM MQ Explorer 對等項目?
結束接聽器	endmqsr, 僅在下列平台上: <ul style="list-style-type: none"> •  AIX •  Solaris •  Windows 系統 		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 IBM MQ 指令 (MQSC) 或 IBM MQ Explorer 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 [UNIX, Linux, and Windows 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 [使用控制指令進行管理](#)。

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。

ULW 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。

必須使用相同的通道名稱來定義兩端。兩端必須具有相容的通道類型，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，指令會如出現在指令檔中，以及在 `UNIX`, `Linux`, and `Windows` 中所鍵入的一樣顯示。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。使用連結字元 `+`，如所示，以在多行上繼續：

- **Windows** 在 `Windows` 上，請使用 `Ctrl-z` 來結束指令行的項目。
- **Linux** **UNIX** 在 `UNIX` and `Linux` 上，使用 `Ctrl-d`。
- 或者，在 `UNIX`, `Linux`, and `Windows` 上，使用 `end` 指令。

ULW 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
```

ULW 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。

ULW 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、伺服器及叢集傳送端通道使用。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，則在儲存之前會先由 IBM MQ 加密該密碼。然後會在流經交談之前將它解密。

ULW 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：

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

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

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

`MQ_DATA_PATH` 代表 IBM MQ 安裝所在的高階目錄。

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

▶ **Linux** ▶ **UNIX** **UNIX and Linux**

`/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 UNIX, Linux, and Windows 上，使用 **runmqclsr** 指令來啟動 IBM 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。

```
runmqclsr -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。

ULW 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 182 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。

ULW 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，兩端各一個。請確定您同時在兩端重新命名通道。

ULW 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 195 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 173 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。

相關工作

[第 147 頁的『配置分散式佇列』](#)

本節提供 IBM 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](#)

ULW 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、伺服器或叢集傳送端
- 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 195 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 173 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。

相關工作

第 147 頁的『[配置分散式佇列](#)』

本節提供 IBM 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](#)

Windows 在 Windows 上設定通訊

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。若要成功，連線必須已定義且可用。本節說明如何使用 IBM MQ for Windows 系統可用的通訊形式來執行此動作。

開始之前

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

 使用 TCP/IP 的訊息通道可以指向 IBM Aspera fasp.io Gateway，它提供快速 TCP/IP 通道，可大幅增加網路傳輸量。在任何授與的 CD 平台上執行的佇列管理程式可以透過 Aspera gateway 進行連接。閘道本身部署在 Red Hat 上或 Ubuntu Linux。請參閱 [在 Linux 上定義 Aspera gateway 連線](#)。

關於這項作業

在 Windows 上設定 IBM MQ 的通訊時，您可以選擇下列通訊類型：

- TCP/IP
- LU 6.2
- NetBIOS

程序

- 如需為 Windows 系統設定通訊的相關資訊，請參閱所選通訊類型的子主題：
 - 第 202 頁的『[在 Windows 上定義 TCP 連線](#)』
 - 第 203 頁的『[在 Windows 上定義 LU 6.2 連線](#)』
 - 第 205 頁的『[在 Windows 上定義 NetBIOS 連線](#)』

在使用 TCP/IP 以外的通訊協定的環境中，並非 IBM MQ for Windows 的所有功能及機能都可用。無法使用的項目是 IBM MQ Explorer。

相關工作

第 192 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 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 13 頁的『[配置用戶端與伺服器之間的連線](#)』

若要配置 IBM MQ MQI clients 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 207 頁的『[在 UNIX and Linux 上設定通訊](#)』

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。若要成功，連線必須已定義且可用。本節說明如何使用 IBM MQ for UNIX or Linux 系統可用的通訊形式來執行此動作。

相關參考

第 14 頁的『[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 IBM MQ MQI client 與伺服器平台的組合。

Windows 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。

開始之前

V 9.1.4 **MQ Adv.** **CD** 使用 TCP/IP 的訊息通道可以指向 IBM Aspera fasp.io Gateway，它提供快速 TCP/IP 通道，可大幅增加網路傳輸量。在任何授與的 CD 平台上執行的佇列管理程式可以透過 Aspera gateway 進行連接。閘道本身部署在 Red Hat 上或 Ubuntu Linux。請參閱 [在 Linux 上定義 Aspera gateway 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 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 MQ for Windows 的 .ini 檔案中指定預設埠號，以變更預設埠號：

```
TCP:  
Port=1822
```

註：若要選取要使用的 TCP/IP 埠號，IBM 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 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 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 MQ 接聽器。

若要執行隨 IBM MQ 提供的接聽器（以執行緒方式啟動新通道），請使用 `runmqclsr` 指令。

使用 `runmqclsr` 指令的基本範例：

```
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。

註: 若要選取要使用的 TCP/IP 埠號, IBM 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 MQ 的號碼。

若要取得最佳效能, 請如第 191 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 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 180 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。

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。

如需相關資訊, 以及 Windows 的特定值, 請參閱第 210 頁的『在 IBM MQ for Multiplatforms 上使用 TCP 接聽器待辦事項選項』。

Windows 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 MVS/ESA, 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 IBM 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。

如需配置 AnyNet SNA over TCP/IP 的最新資訊，請參閱下列線上 IBM 文件: [AnyNet SNA over TCP/IP 及 SNA Node Operations](#)。

相關概念

第 204 頁的『在 Windows 上的 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 204 頁的『在 Windows 上接收 LU 6.2』

接收通道程式會啟動，以回應來自傳送通道的啟動要求。

Windows 在 Windows 上的 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
```

Windows 在 Windows 上接收 LU 6.2

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，請依照執行通道和接聽器作為授信應用程式中的說明，將 IBM 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 IBM 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 *Microsoft SNA 伺服器 APPC 程式設計師手冊* 或 *Microsoft 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。

Windows 在 Windows 上定義 NetBIOS 連線

NetBIOS 連線僅適用於執行 Windows 的用戶端及伺服器。當建立與另一個 IBM MQ 產品的 NetBIOS 連線時，IBM MQ 會使用三種類型的 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 IBM 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 IBM 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 205 頁的『[定義 IBM MQ 本端 NetBIOS 名稱](#)』

IBM MQ 通道處理程序所使用的本端 NetBIOS 名稱可以用三種方式來指定。

第 206 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 206 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 MQ 可讓您使用 `qm.ini` 檔案的 NETBIOS 段落中的 `AdapterNum` 值，並在 `runmqlsr` 指令上指定 `-a` 參數，來控制 LAN 配接卡 (LANA) 號碼的選擇。

第 206 頁的『[起始 NetBIOS 連線](#)』

定義起始連線所需的步驟。

第 207 頁的『[定義 NetBIOS 連線的目標接聽器](#)』

定義要在 NetBIOS 連線接收端執行的步驟。

Windows 定義 IBM MQ 本端 NetBIOS 名稱

IBM MQ 通道處理程序所使用的本端 NetBIOS 名稱可以用三種方式來指定。

依優先順序，有三種方式：

1. `RUNMQLSR` 指令的 `-l` 參數中指定的值，例如：

```
RUNMQLSR -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，則必須在啟動 IBM MQ 處理程序的每一個視窗中發出 `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。

Windows 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. RUNMQLSR 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
```

Windows 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 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 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。

Windows 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 通道必須以執行緒方式執行。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
```

Windows 定義 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 IBM MQ 接聽器程式以建立工作站，並讓它可以聯絡它。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。

若要取得最佳效能，請如第 191 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 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](#)。

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 IBM 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。

Linux UNIX 在 UNIX and Linux 上設定通訊

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。若要成功，連線必須已定義且可用。本節說明如何使用 IBM MQ for UNIX or Linux 系統可用的通訊形式來執行此動作。

開始之前

您可能會發現參閱下列各節會很有用：

- **AIX** [配置範例- IBM MQ for AIX](#)
- **Solaris** [配置範例- IBM MQ for Solaris](#)
- **Linux** [配置範例- IBM MQ for Linux](#)

V 9.1.4 **MQ Adv.** **CD** 使用 TCP/IP 的訊息通道可以指向 IBM Aspera fasp.io Gateway，它提供快速 TCP/IP 通道，可大幅增加網路傳輸量。在任何授與的 CD 平台上執行的佇列管理程式可以透過 Aspera gateway 進行連接。閘道本身部署在 Red Hat 上或 Ubuntu Linux。請參閱 [在 Linux 上定義 Aspera gateway 連線](#)。

關於這項作業

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。

在 UNIX and Linux 上設定 IBM MQ 的通訊時，您可以選擇下列通訊類型：

- TCP/IP

- LU 6.2

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 MQ MQI clients, 具有使用不同傳輸通訊協定的替代通道可能很有用。如需 IBM MQ MQI clients 的相關資訊, 請參閱 [IBM MQ MQI clients 概觀](#)。

程序

如需為 UNIX and Linux 系統設定通訊的相關資訊, 請參閱所選通訊類型的子主題:

- [第 208 頁的『在 UNIX and Linux 上定義 TCP 連線』](#)
- [第 211 頁的『在 UNIX and Linux 上定義 LU 6.2 連線』](#)
- [V 9.1.4](#) [MQ Adv.](#) [Linux](#) [CD](#) [第 660 頁的『在 Linux 上定義 Aspera gateway 連線』](#)

相關工作

[第 192 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 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 13 頁的『配置用戶端與伺服器之間的連線』](#)

若要配置 IBM MQ MQI clients 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 201 頁的『在 Windows 上設定通訊』](#)

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。若要成功, 連線必須已定義且可用。本節說明如何使用 IBM MQ for Windows 系統可用的通訊形式來執行此動作。

相關參考

[第 14 頁的『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 IBM MQ MQI client 與伺服器平台的組合。

[Linux](#) [UNIX](#) **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。

開始之前

[V 9.1.4](#) [MQ Adv.](#) [CD](#) 使用 TCP/IP 的訊息通道可以指向 IBM Aspera fasp.io Gateway, 它提供快速 TCP/IP 通道, 可大幅增加網路傳輸量。在任何授與的 CD 平台上執行的佇列管理程式可以透過 Aspera gateway 進行連接。閘道本身部署在 Red Hat 上或 Ubuntu Linux。請參閱 [在 Linux 上定義 Aspera gateway 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 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 IBM 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 209 頁的『在 UNIX and Linux 上使用 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 210 頁的『在 IBM MQ for Multiplatforms 上使用 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。

第 211 頁的『使用 IBM MQ 接聽器』

若要執行隨 IBM 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 211 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。

Linux > UNIX 在 UNIX and Linux 上使用 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 IBM 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 IBM 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：

• Linux > AIX 在 AIX 上：

```
refresh -s inetd
```

• Linux > Solaris 在 Solaris 10 或更新版本上：

```
inetconv
```

• Linux > UNIX 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 設定的語言環境可能不符合 IBM MQ 處理程序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 IBM 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 IBM 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 210 頁的『在 IBM MQ for Multiplatforms 上使用 TCP 接聽器待辦事項選項』。

Multi 在 IBM MQ for Multiplatforms 上使用 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 210 頁的表 24 中。

伺服器平台	連線要求數上限
AIX AIX	100
Linux Linux	100
IBM i IBM i	255
Solaris Solaris	100
Windows Windows 伺服器	100

如果待辦事項達到 第 210 頁的表 24 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
```

這會置換未完成要求的預設數目上限 (請參閱 第 210 頁的表 24) TCP/IP 接聽器。

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。

如果要在啟用 backlog 選項的情況下執行接聽器，請執行下列任一動作：

- 使用 runmqtsr -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](#)。

相關概念

第 779 頁的『[使用 TCP 接聽器待辦事項選項](#)』

在 TCP/IP 上接收時，會設定未完成的連線要求數目上限。這些未完成的的要求可視為在 TCP/IP 埠上等待接聽器接受要求之要求的待辦事項。

Linux → UNIX 使用 IBM MQ 接聽器

若要執行隨 IBM 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。

若要取得最佳效能，請如第 191 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 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](#)。

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 IBM 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。

Linux → UNIX 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 180 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。

Linux → UNIX 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 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。

表 25: 遠端佇列管理程式平台的本端 UNIX and Linux 系統上的設定		
遠端平台	TPNAME	TPPATH
z/OS 沒有 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。	-

表 25: 遠端佇列管理程式平台的本端 UNIX and Linux 系統上的設定 (繼續)

遠端平台	TPNAME	TPPATH
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。	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 IBM 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 212 頁的『在 UNIX and Linux 上的 LU 6.2 上傳送結束』

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 212 頁的『在 UNIX and Linux 上接收 LU 6.2』

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。

Linux **UNIX** 在 UNIX and Linux 上的 LU 6.2 上傳送結束

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
```

Solaris 在 Solaris 上, 將 APPC_LOCAL_LU 環境變數設為本端 LU 名稱。

當 IBM 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。

Linux **UNIX** 在 UNIX and Linux 上接收 LU 6.2

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 IBM 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。

Solaris **AIX** 在 AIX 及 Solaris 上, 設定 APPCTPN (交易名稱) 及 APPCLLU (本端 LU 名稱) 環境變數 (您可以將配置畫面用於所呼叫的交易程式)。

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。

使用 DQM 指令和畫面來建立、監視及控制遠端佇列管理程式的通道。每一個佇列管理程式都有一個 DQM 程式，用來控制與相容遠端佇列管理程式的交互連線。

關於這項作業

下列清單是通道控制功能元件的簡要說明：

- 通道定義會保留為佇列管理程式物件。
- 通道指令是 IBM MQ for IBM i 指令集的子集。

使用指令 GO CMDMQM 來顯示完整的 IBM MQ for IBM i 指令集。

- 您可以使用通道定義畫面或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
- 也可以使用 MQSC 來管理通道
- 也可以使用 IBM MQ Explorer 來管理通道
- 序號及邏輯工作單元 (LUW) ID 儲存在同步化檔案中，並用於通道同步化目的。

您可以使用指令和畫面來：定義訊息通道和相關聯的物件，以及監視和控制訊息通道。透過使用 F4=Prompt 鍵，您可以指定相關的佇列管理程式。如果您不使用提示，則會採用預設佇列管理程式。使用 F4=Prompt，會顯示另一個畫面，您可以在其中輸入相關佇列管理程式名稱，有時也可以輸入其他資料。

您需要使用畫面來定義的物件如下：

- 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
- 訊息通道定義

如需使用這些物件所涉及之概念的相關資訊，請參閱第 147 頁的『[配置分散式佇列](#)』。

通道必須完整定義，且其相關聯物件必須存在且可供使用，才能啟動通道。

此外，每個通道的特定通訊鏈結必須先定義且可用，才能執行通道。如需如何定義 LU 6.2 及 TCP/IP 鏈結的說明，請參閱安裝的特定通訊手冊。

程序

- 如需建立及使用物件的相關資訊，請參閱：
 - [第 214 頁的『在 IBM i 上建立物件』](#)
 - [第 214 頁的『在 IBM i 上建立通道』](#)
 - [第 216 頁的『在 IBM i 上啟動通道』](#)
 - [第 217 頁的『在 IBM i 上選取通道』](#)
 - [第 217 頁的『在 IBM i 上瀏覽通道』](#)
 - [第 219 頁的『在 IBM i 上重新命名通道』](#)
 - [第 219 頁的『在 IBM i 上使用通道狀態』](#)
 - [第 220 頁的『IBM i 上的使用通道選項』](#)

相關概念

[第 225 頁的『設定 IBM i 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。若要成功，必須定義連線並使其可用。

相關工作

第 13 頁的『配置用戶端與伺服器之間的連線』

若要配置 IBM MQ MQI clients 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。

相關參考

[配置範例- IBM MQ for IBM i](#)

[IBM MQ for IBM i 的訊息通道規劃範例](#)

[IBM MQ for IBM i CL 指令](#)

IBM i 在 IBM i 上建立物件

您可以使用 CRTMQMQ 指令來建立佇列及別名物件。

您可以建立佇列及別名物件，例如：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。

如需預設物件的清單，請參閱 [系統及預設物件](#)。

IBM i 在 IBM i 上建立通道

您可以從「建立通道」畫面或在指令行上使用 CRTMQMCHL 指令來建立通道。

若要建立通道，請執行下列動作：

1. 從「使用 MQM 通道 (WRKMQMCHL)」畫面中使用 F6。

或者，從指令行使用 CRTMQMCHL 指令。

不論任何一種方式，都會顯示「建立通道」畫面。類型：

- 所提供欄位中的通道名稱
- 鏈結這一端的通道類型

2. 按 Enter 鍵。

註：您必須唯一命名網路中的所有通道。如 [顯示所有通道的網路圖](#) 中所示，在通道名稱中包括來源和目標佇列管理程式名稱是很好的作法。

即會驗證您的項目，並立即報告錯誤。更正任何錯誤並繼續。

即會呈現您所選擇通道類型的適當通道設定畫面。使用您先前收集的資訊來完成欄位。按 Enter 鍵以建立通道。

在說明畫面的通道定義畫面說明中，以及在 [通道屬性](#) 中，系統會提供協助來決定各種欄位的內容。

```

Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

Channel name . . . . . > CHANNAME_____
Channel type . . . . . > *SDR__  *RCVR, *SDR, *SVR, *RQSTR...
Message Queue Manager name *DFT_____

-----
Replace . . . . . *NO   *NO, *YES
Transport type . . . . . *TCP___ *LU62, *TCP, *SYSDFTCHL
Text 'description' . . . . . > 'Example Channel Definition'_____

-----
Connection name . . . . . *SYSDFTCHL_____
-----
-----
-----
-----
-----

More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

圖 27: 建立通道 (1)

```

Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

Transmission queue . . . . . 'TRANSMISSION_QUEUE_NAME'_____

-----
Message channel agent . . . . . *NONE_____ Name, *SYSDFTCHL, *NONE
Library . . . . . _____ Name
Message channel agent user ID . *SYSDFTCHL__ Character value...
Coded Character Set Identifier *SYSDFTCHL__ 0-9999, *SYSDFTCHL
Batch size . . . . . 50_____ 1-9999, *SYSDFTCHL
Disconnect interval . . . . . 6000_____ 1-999999, *SYSDFTCHL
Short retry interval . . . . . 60_____ 0-999999999, *SYSDFTCHL
Short retry count . . . . . 10_____ 0-999999999, *SYSDFTCHL
Long retry interval . . . . . 1200_____ 0-999999999, *SYSDFTCHL
Long retry count . . . . . 999999999__ 0-999999999, *SYSDFTCHL
Security exit . . . . . *NONE_____ Name, *SYSDFTCHL, *NONE
Library . . . . . _____ Name
Security exit user data . . . . *SYSDFTCHL_____

More...
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

圖 28: 建立通道 (2)

```

Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

Send exit . . . . . *NONE_____ Name, *SYSDFTCHL, *NONE
Library . . . . . _____ Name
+ for more values
Send exit user data . . . . . _____
+ for more values
Receive exit . . . . . *NONE_____ Name, *SYSDFTCHL, *NONE
Library . . . . . _____ Name
+ for more values
-----
Receive exit user data . . . . . _____
+ for more values
Message exit . . . . . *NONE_____ Name, *SYSDFTCHL, *NONE
Library . . . . . _____ Name
+ for more values
-----
More...
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

```

圖 29: 建立通道 (3)

```

Create MQM Channel (CRTMQMCHL)

Type choices, press Enter.

Message exit user data . . . . . _____
+ for more values
Convert message . . . . . *SYSDFTCHL_ *YES, *NO, *SYSDFTCHL
Sequence number wrap . . . . . 99999999__ 100-99999999, *SYSDFTCHL
Maximum message length . . . . . 4194304___ 0-4194304, *SYSDFTCHL
Heartbeat interval . . . . . 300_____ 0-999999999, *SYSDFTCHL
Non Persistent Message Speed . . *FAST_____ *FAST, *NORMAL, *SYSDFTCHL
Password . . . . . *SYSDFTCHL_ Character value, *BLANK...
Task User Profile . . . . . *SYSDFTCHL_ Character value, *BLANK...
Transaction Program Name . . . . *SYSDFTCHL

```

Bottom
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

圖 30: 建立通道 (4)

IBM i 在 IBM i 上啟動通道

您可以從「使用通道」畫面或在指令行上使用 STRMQCHL 指令來啟動通道。

接聽器僅適用於 TCP。對於 SNA 接聽器，您必須配置通訊子系統。

若要讓應用程式能夠交換訊息，您必須使用 STRMQLSR 指令啟動入埠連線的接聽器程式。

對於出埠連線，您必須使用下列其中一種方式來啟動通道：

1. 視 MCATYPE 參數而定，使用 CL 指令 STRMQMCHL 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。)

```
STRMQMCHL CHLNAME(QM1.TO.QM2) MQNAME(MYQMGR)
```

2. 使用通道起始程式來觸發通道。當佇列管理程式啟動時，會自動啟動一個通道起始程式。透過變更該佇列管理程式之 qm.ini 檔案中的 chinit 段落，可以刪除此自動啟動。

3. 使用 WRKMQMCHL 指令來開始「使用通道」畫面，並選擇選項 14 來啟動通道。

IBM i 在 IBM i 上選取通道

您可以從「使用通道」畫面中選取通道。

若要選取通道，請使用 WRKMQMCHL 指令，從「使用通道」畫面開始：

1. 將游標移至與所需通道名稱相關聯的選項欄位。
2. 鍵入選項號碼。
3. 請按 Enter 鍵以啟動您的選擇。

如果您選取多個通道，則會依序啟動選項。

```
Work with MQM Channels

Queue Manager Name . . : CNX

Type options, press Enter.
2=Change 3=Copy 4=Delete 5=Display 8=Work with Status 13=Ping
14=Start 15=End 16=Reset 17=Resolve

Opt  Name                Type      Transport  Status
CHLNIC          *RCVR    *TCP      INACTIVE
CORSAIR.TO.MUSTANG *SDR     *LU62     INACTIVE
FV.CHANNEL.MC.DJE1 *RCVR    *TCP      INACTIVE
FV.CHANNEL.MC.DJE2 *SDR     *TCP      INACTIVE
FV.CHANNEL.MC.DJE3 *RQSTR   *TCP      INACTIVE
FV.CHANNEL.MC.DJE4 *SVR     *TCP      INACTIVE
FV.CHANNEL.PETER  *RCVR    *TCP      INACTIVE
FV.CHANNEL.PETER.LU *RCVR    *LU62     INACTIVE
FV.CHANNEL.PETER.LU1 *RCVR    *LU62     INACTIVE
More...
Parameters or command
===>
F3=Exit F4=Prompt F5=Refresh F6=Create F9=Retrieve F12=Cancel
F21=Print
```

圖 31: 使用通道

IBM i 在 IBM i 上瀏覽通道

您可以從「顯示通道」畫面或在指令行上使用 DSPMQMCHL 指令來瀏覽通道。

若要瀏覽通道的設定，請使用 WRKMQMCHL 指令，從「顯示通道」畫面開始：

1. 針對必要的通道名稱鍵入選項 5 (顯示)。
2. 請按 Enter 鍵以啟動您的選擇。

如果您選取多個頻道，則會依序呈現它們。

或者，您可以從指令行使用 DSPMQMCHL 指令。

這會導致顯示適當的「顯示通道」畫面，以及通道現行設定的詳細資料。這些欄位在 [通道屬性](#) 中說明。

```

Display MQM Channel

Channel name . . . . . : ST.JST.2T01
Queue Manager Name . . . . . : QMREL
Channel type . . . . . : *SDR
Transport type . . . . . : *TCP
Text 'description' . . . . . : John's sender to WINSDOA1

Connection name . . . . . : MUSTANG

Transmission queue . . . . . : WINSDOA1

Message channel agent . . . . . :
Library . . . . . :
Message channel agent user ID : *NONE
Batch interval . . . . . : 0
Batch size . . . . . : 50
Disconnect interval . . . . . : 6000

F3=Exit F12=Cancel F21=Print

```

圖 32: 顯示 TCP/IP 通道 (1)

```

Display MQM Channel

Short retry interval . . . . . : 60
Short retry count . . . . . : 10
Long retry interval . . . . . : 6000
Long retry count . . . . . : 10
Security exit . . . . . :
Library . . . . . :
Security exit user data . . . . . :
Send exit . . . . . :
Library . . . . . :
Send exit user data . . . . . :
Receive exit . . . . . :
Library . . . . . :
Receive exit user data . . . . . :
Message exit . . . . . :
Library . . . . . :
Message exit user data . . . . . :
More...

F3=Exit F12=Cancel F21=Print

```

圖 33: 顯示 TCP/IP 通道 (2)

```
Display MQM Channel
Sequence number wrap . . . . . : 999999999
Maximum message length . . . . : 10000
Convert message . . . . . : *NO
Heartbeat interval . . . . . : 300
Nonpersistent message speed . . *FAST
```

Bottom

F3=Exit F12=Cancel F21=Print

圖 34: 顯示 TCP/IP 通道 (3)

IBM i 在 IBM i 上重新命名通道

您可以從「使用通道」畫面中重新命名通道。

如果要重新命名訊息通道，請從「使用通道」畫面開始：

1. 結束通道。
2. 使用選項 3 (複製) 來建立具有新名稱的重複項。
3. 請使用選項 5 (顯示畫面) 來檢查它是否已正確建立。
4. 請使用選項 4 (刪除) 來刪除原始通道。

如果您決定重新命名訊息通道，請確保同時重新命名兩個通道端點。

IBM i 在 IBM i 上使用通道狀態

您可以從「使用通道狀態」畫面中使用通道狀態。

使用 WRKMQMCHST 指令，顯示顯示通道狀態的一組畫面中的第一個畫面。當您選取「變更視圖」(F11) 時，您可以依序檢視狀態畫面。

或者，從「使用 MQM 通道」畫面中選取選項 8 (使用狀態) 也會顯示第一個狀態畫面。

MQSeries Work with Channel Status

Type options, press Enter.

5=Display 13=Ping 14=Start 15=End 16=Reset 17=Resolve

Opt Name	Connection	Indoubt	Last Seq
CARTS_CORSAIR_CHAN	GBIBMIYA.WINSDOA1	NO	1
CHLNIC	9.20.2.213	NO	3
FV.CHANNEL.PETER2	9.20.2.213	NO	6225
JST.1.2	9.20.2.201	NO	28
MP_MUST_TO_CORS	9.20.2.213	NO	100
MUSTANG.TO.CORSAIR	GBIBMIYA.WINSDOA1	NO	10
MP_CORS_TO_MUST	9.20.2.213	NO	101
JST.2.3	9.5.7.126	NO	32
PF_WINSDOA1_LU62	GBIBMIYA.IYA80020	NO	54
PF_WINSDOA1_LU62	GBIBMIYA.WINSDOA1	NO	500
ST.JCW.EXIT.2T01.CHL	9.20.2.213	NO	216

Bottom

Parameters or command

==>

F3=Exit F4=Prompt F5=Refresh F6=Create F9=Retrieve F11=Change view

F12=Cancel F21=Print

圖 35: 通道狀態畫面集的第一個

「使用通道狀態」畫面中可用的選項如下:

功能表選項	說明
5=Display	顯示通道設定。
13=Ping	在適當的情況下起始連線測試動作。
14=Start	啟動通道。
15=End	停止通道。
16=Reset	重設通道序號。
17=Resolve	手動解決不確定的通道狀況。

IBM i 上的使用通道選項

使用指令 WRKMQMCHL 可到達「使用通道」畫面，它可讓您監視列出之所有通道的狀態，並針對選取的通道發出指令。

「使用通道」畫面中可用的選項如下:

功能表選項	說明
第 221 頁的 『2=Change』	變更通道的屬性。
第 221 頁的『3=Copy』	將通道的屬性複製到新通道。
第 221 頁的 『4=Delete』	刪除通道。
第 221 頁的 『5=Display』	顯示通道的現行設定。
第 221 頁的 『6=Create』	顯示「建立通道」畫面
第 222 頁的『8=Work 狀 態』	顯示通道狀態畫面。

功能表選項	說明
第 223 頁的『13=Ping』	執行連線測試機能，透過與遠端系統交換固定資料訊息來測試與相鄰系統的連線。
第 223 頁的『14=Start』	啟動選取的通道，或重設已停用的接收端通道。
第 224 頁的『15=End』	要求通道關閉。
第 224 頁的『16=Reset』	要求通道重設鏈結這一端的序號。數字必須在兩端相等，才能啟動通道。
第 224 頁的『17=Resolve』	要求通道解決不確定的訊息，而不建立與另一端的連線。
第 225 頁的『18=Display 權限』	顯示 IBM MQ 物件權限
第 225 頁的『19=Grant 權限』	授與 IBM MQ 物件權限
第 225 頁的『20=Revoke 權限』	撤銷 IBM MQ 物件權限
第 225 頁的『21=Recover 物件』	回復 IBM MQ 物件
第 225 頁的『22=Record 影像』	記錄 IBM MQ 物件影像

IBM i **2=Change**

請使用「變更」選項來變更現有的通道定義。

「變更」選項或 CHGMQMCHL 指令會變更現有的通道定義，但通道名稱除外。在通道定義畫面中鍵入要變更的欄位，然後按 Enter 鍵儲存更新的定義。

IBM i **3=Copy**

使用「複製」選項來複製現有的通道。

「複製」選項使用 CPYMQMCHL 指令來複製現有的通道。「複製」畫面可讓您定義新的通道名稱。不過，您必須將使用的字元限制為適用於 IBM i 物件名稱的那些字元；請參閱 [管理 IBM MQ for IBM i](#)。

在「複製」畫面上按 Enter 鍵以顯示現行設定的詳細資料。您可以變更任何新的通道設定。按 Enter 鍵儲存新的通道定義。

IBM i **4=Delete**

使用「刪除」選項來刪除選取的通道。

即會顯示一個畫面，以確認或取消您的要求。

IBM i **5=Display**

使用「顯示」選項來顯示通道的現行定義。

此選項會顯示畫面，其中的欄位會顯示參數的現行值，並防止使用者輸入。

IBM i **6=Create**

使用「建立」選項來顯示「建立通道」畫面。

使用「建立」選項，或從指令行輸入 CRTMQMCHL 指令，以取得「建立通道」畫面。從 [第 215 頁的圖 27](#) 開始，有「建立通道」畫面的範例。

使用此畫面，您可以從填入 IBM MQ for IBM i 所提供預設值的欄位畫面建立通道定義。輸入通道的名稱，選取您要建立的通道類型，以及要使用的通訊方法。

當您按 Enter 鍵時，即會顯示畫面。在此畫面及其餘畫面的所有必要欄位中鍵入資訊，然後按 Enter 鍵儲存定義。

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 MQ for IBM i 物件名稱的那些字元。

所有畫面都具有部分欄位的 IBM MQ for IBM i 所提供的預設值。您可以自訂這些值，也可以在建立或複製通道時變更它們。若要自訂值，請參閱 *IBM MQ for IBM i* 系統管理。

您可以透過針對每一種通道類型設定具有必要預設值的虛擬通道，並在每次想要建立新的通道定義時複製它們，來建立您自己的通道預設值集。

相關參考

通道屬性

8=Work 狀態

請使用「使用狀態」來查看詳細通道狀態資訊。

狀態直欄會告訴您通道是作用中還是非作用中，且會連續顯示在「使用 MQM 通道」畫面中。請使用選項 8 (使用狀態) 來查看顯示的更多狀態資訊。或者，可以使用 WRKMQMCHST 指令從指令行顯示此資訊。請參閱第 219 頁的『在 IBM i 上使用通道狀態』。

- 通道名稱
- 通道類型
- 通道狀態
- 通道實例
- 遠端佇列管理程式
- 傳輸佇列名稱
- 通訊連線名稱
- 通道的不確定狀態
- 最後序號
- 不確定訊息數
- 不確定的序號
- 傳輸佇列上的訊息數
- 邏輯工作單元 ID
- 不確定的邏輯工作單元 ID
- 通道次狀態
- 監視通道
- 標頭壓縮
- 訊息壓縮
- 壓縮時間指示器
- 壓縮率指示器
- 傳輸佇列時間指示器
- 網路時間指示器
- 結束時間指示器
- 批次大小指示器
- 現行共用交談
- 共用交談數上限

IBM i 13=Ping

使用 Ping 選項與遠端系統交換固定資料訊息。

成功的 IBM MQ Ping 可讓系統監督者有通道可用及運作的信心。

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。

它只能從傳送端和伺服器通道使用。對應的通道在鏈路的遠端啟動，並執行啟動引數協商。正常通知錯誤。

訊息交換的結果會呈現在您的「連線測試」畫面中，並且是傳回的訊息文字，以及傳送訊息的時間和收到回覆的時間。

使用 LU 6.2 進行連線測試

在 IBM MQ for IBM i 中呼叫 Ping 時，會以要求函數之使用者的使用者 ID 來執行它，而通道程式的正常執行方式則是針對通道程式採用 QMQM 使用者 ID。使用者 ID 會流向接收端，且在接收端必須有效，才能配置 LU 6.2 交談。

IBM i 14=Start

使用啟動選項來手動啟動通道。

「啟動」選項適用於傳送端、伺服器及要求端通道。如果已設定佇列管理程式觸發的通道，則不需要這樣做。

「啟動」選項也用於接收端、伺服器連線、叢集傳送端及叢集接收端通道。啟動處於 STOPPED 狀態的接收端通道表示可以從遠端通道啟動它。

啟動時，傳送端 MCA 會讀取通道定義檔，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。

當您使用觸發時，必須啟動持續執行的觸發程序來監視起始佇列。STRMQMCHLI 指令可用來啟動處理程序。

在通道的遠端，接收處理程序可能會從傳送端開始回應通道啟動。針對 LU 6.2 及 TCP/IP 連接通道執行此動作的方法不同：

- LU 6.2 連接通道在通道接收端不需要任何明確動作。
- TCP 連接的通道需要接聽器處理程序持續執行。此處理程序會等待來自鏈結遠端的通道啟動要求，並啟動該連線的通道定義中定義的處理程序。

當遠端系統為 IBM i 時，您可以使用 STRMLSR 指令。

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、針對 GET 啟用，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。
- 訊息通道必須是非作用中。

若要傳送訊息，遠端佇列和遠端佇列定義必須存在。

會傳回一則訊息給畫面，確認已接受啟動通道的要求。如需確認「啟動」處理程序已成功，請檢查系統日誌，或按 F5 鍵 (重新整理畫面)。

IBM i 15=End

使用「結束」來停止通道活動

使用「結束」選項來要求通道停止活動。通道不再傳送任何其他訊息。

在按 Enter 鍵之前選取 F4，以選擇通道是否變成「已停止」或「非作用中」，以及是否使用「受控制」或「立即停止」來停止通道。操作員必須重新啟動已停止的通道，才能重新變成作用中。可以觸發非作用中通道。

stop immediate

請使用「立即停止」來停止通道，而不完成任何工作單元。

此選項會終止通道處理程序。因此，通道不會完成處理現行訊息批次，因此無法讓通道處於不確定的狀態。一般而言，操作員最好使用受控制的停止選項。

受控制的停止

使用「受控制停止」，在現行工作單元結束時停止通道。


此選項會要求通道依序關閉；完成現行訊息批次，並與通道另一端一起執行同步點程序。


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。您可以使用下列方式來重新啟動通道：

- 使用 **START CHANNEL MQSC** 指令。
- 使用 **Start Channel PCF** 指令。
- 使用 IBM MQ Explorer。
-  在 z/OS 上，使用「啟動通道」畫面。
-  在 IBM i 上，使用 **STRMQMCHL CL** 指令或 **WRKMQMCHL** 畫面上的 **START** 選項。

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。

 如果通道起始程式在通道處於 RETRYING 或 STOPPED 狀態時停止，則會在通道起始程式重新啟動時記住通道狀態。不過，如果通道起始程式在通道處於 STOPPED 狀態時停止，則會重設 SVRCONN 通道類型的通道狀態。

 如果在通道處於 RETRYING 或 STOPPED 狀態時停止佇列管理程式，則會在重新啟動佇列管理程式時記住通道狀態。從 IBM MQ 8.0 開始，這也適用於 SVRCONN 通道。先前，如果通道起始程式在通道處於 STOPPED 狀態時停止，則會重設 SVRCONN 通道類型的通道狀態。

IBM i 16=Reset

使用「重設」選項來強制新的訊息順序。

「重設」選項會變更訊息序號。請小心使用它，且只有在您已使用「解決」選項來解決任何不確定的狀況之後。這個選項只適用於傳送端或伺服器通道。下次啟動通道時，第一則訊息會啟動新序列。

IBM i 17=Resolve

使用「解決」選項來強制本端確定或取消保留在傳輸佇列中的不確定訊息。

當傳送端或伺服器保留不確定的訊息時 (例如，因為鏈結的一端已終止，且沒有回復的可能性)，請使用「解決」選項。「解決」選項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
- 通道定義 (本端) 必須存在
- 佇列管理程式必須在執行中 (本端)

▶ IBM i 18=Display 權限

使用「顯示權限」選項，以顯示使用者獲授權對特定 IBM MQ 物件執行的動作。

對於選擇的物件及使用者，DSPMQAUT 指令會顯示使用者必須對 IBM MQ 物件執行動作的授權。如果使用者是多個群組的成員，則指令會顯示所有群組對物件的合併授權。

▶ IBM i 19=Grant 權限

使用「授與權限」選項，將對 IBM MQ 物件執行動作的權限授與另一個使用者或使用者群組。

只有 QMQADM 群組中的使用者才能使用 GRMQMAUT 指令。QMQADM 中的使用者會透過依名稱識別使用者，或將權限授與 *PUBLIC 中的所有使用者，來授與其他使用者對指令中指定的 IBM MQ 物件執行動作的權限。

▶ IBM i 20=Revoke 權限

使用「取消」權限可移除對使用者物件執行動作的權限。

RVKMQMAUT 指令僅適用於 QMQADM 群組中的使用者。QMQADM 群組中的使用者透過依名稱識別使用者，或撤銷 *PUBLIC 中所有使用者的權限，移除其他使用者對指令中所指名之 IBM MQ 物件執行動作的權限。

▶ IBM i 21=Recover 物件

使用「回復」物件，從儲存在 IBM MQ 異動日誌中的資訊還原損壞物件。

回復物件使用「重建 MQ 物件」指令 (RCRMQMOBJ) 來回復指令中所指名已損壞的所有物件。如果物件未損壞，則不會對該物件執行任何動作。

▶ IBM i 22=Record 影像

使用「記錄影像」來減少回復一組物件所需的異動日誌接收器數目，並將回復時間縮至最小。

RCDMQMIMG 指令會取得指令中所選取所有物件的檢查點。它會將整合檔案系統 (IFS) 中物件的現行值與稍後的物件相關資訊 (例如記錄在異動日誌接收器中的 MQPUT 及 MQGET) 同步。

當指令完成時，IFS 中的物件是最新的，且那些異動日誌接收器不再需要存在才能回復物件。任何已切斷的異動日誌接收器都可以分離 (只要它們不需要存在以回復其他物件)。

▶ IBM i 設定 IBM i 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。若要成功，必須定義連線並使其可用。

DQM 是 IBM MQ for IBM i 的遠端佇列作業機能。它為 IBM MQ for IBM i 佇列管理程式提供通道控制程式，這些佇列管理程式會形成通訊鏈結的介面，可由系統操作員控制。

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，必須定義連線並使其可用。本節說明如何確保連線已定義且可用。

在啟動通道之前，必須依照本節的說明來定義傳輸佇列，且必須包含在訊息通道定義中。

您可以在 IBM MQ for IBM i 系統之間選擇下列兩種通訊形式的乙太網路：

- [第 226 頁的『在 IBM i 上定義 TCP 連線』](#)

若為 TCP，可以使用主機位址，並依照 *IBM i Communication Configuration Reference* 中的說明來設定這些連線。

在 TCP 環境中，每個分散式服務都會配置一個唯一的 TCP 位址，可供遠端機器用來存取服務。TCP 位址由主機名稱/號碼及埠號組成。所有佇列管理程式都會使用這類數字，以 TCP 方式彼此通訊。

- [第 227 頁的『在 TCP 上接收』](#)

這種通訊形式需要定義 IBM i SNA 邏輯單元類型 6.2 (LU 6.2)，以提供負責本端佇列管理程式的 IBM i 系統與負責遠端佇列管理程式的系統之間的實體鏈結。如需在 IBM i 中配置通訊的詳細資料，請參閱 *IBM i 通訊配置參考手冊*。

在必要時，觸發安排也必須根據必要程序和佇列的定義來準備。

V 9.1.4 **MQ Adv.** **CD** 使用 TCP/IP 的訊息通道可以指向 IBM Aspera fasp.io Gateway，它提供快速 TCP/IP 通道，可大幅增加網路傳輸量。在任何授與的 CD 平台上執行的佇列管理程式可以透過 Aspera gateway 進行連接。閘道本身部署在 Red Hat 上或 Ubuntu Linux。請參閱 [在 Linux 上定義 Aspera gateway 連線](#)。

相關工作

[第 213 頁的『在 IBM i 上監視及控制通道』](#)

使用 DQM 指令和畫面來建立、監視及控制遠端佇列管理程式的通道。每一個佇列管理程式都有一個 DQM 程式，用來控制與相容遠端佇列管理程式的交互連線。

相關參考

[配置範例- IBM MQ for IBM i](#)

[IBM MQ for IBM i 的訊息通道規劃範例](#)

[IBM i 上的交互通訊工作](#)

[IBM i 上的通道狀態](#)

IBM i 在 *IBM i* 上定義 TCP 連線

您可以使用連線名稱欄位，在通道定義內定義 TCP 連線。

通道定義包含一個欄位 CONNECTION NAME，其中包含目標或主機名稱的 TCP 網路位址 (例如 ABCHOST)。TCP 網址可以是 IPv4 帶點十進位格式 (例如 127.0.0.1) 或 IPv6 十六進位格式 (例如 2001:DB8:0:0:0:0:0:0)。如果 CONNECTION NAME 是主機名稱或名稱伺服器，則會使用 IBM i 主機表格將主機名稱轉換為 TCP 主機位址。

完整 TCP 位址需要埠號；如果未提供此號碼，則會使用預設埠號 1414。在連線的起始端 (傳送端、要求端及伺服器通道類型)，可以提供連線的選用埠號，例如：

```
Connection name 127.0.0.1 (1555)
```

在此情況下，起始端會嘗試連接至埠 1555 的接收端程式。

V 9.1.4 **MQ Adv.** **CD** 使用 TCP/IP 的訊息通道可以指向 IBM Aspera fasp.io Gateway，它提供快速 TCP/IP 通道，可大幅增加網路傳輸量。在任何授與的 CD 平台上執行的佇列管理程式可以透過 Aspera gateway 進行連接。閘道本身部署在 Red Hat 上或 Ubuntu Linux。請參閱 [在 Linux 上定義 Aspera gateway 連線](#)。

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。

如需相關資訊，以及 IBM i 的特定值，請參閱 [第 210 頁的『在 IBM MQ for Multiplatforms 上使用 TCP 接聽器待辦事項選項』](#)。

相關概念

第 227 頁的『在 TCP 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。若要回應啟動要求，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 STRMQMLSR 指令啟動此接聽器程式。

IBM i 在 TCP 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。若要回應啟動要求，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 STRMQMLSR 指令啟動此接聽器程式。

您可以針對每一個佇列管理程式啟動多個接聽器。依預設，STRMQMLSR 指令會使用埠 1414，但您可以置換此值。若要置換預設值，請將下列陳述式新增至所選取佇列管理程式的 qm.ini 檔案。在此範例中，需要接聽器才能使用埠 2500:

```
TCP:
Port=2500
```

qm.ini 檔位於此 IFS 目錄 :/QIBM/UserData/mqm/qmgrs/ 佇列管理程式名稱。

當 TCP 接聽器啟動時，這個新值是唯讀的。如果您已在執行接聽器，則該程式不會看到此變更。若要使用新值，請停止接聽器，然後重新發出 STRMQMLSR 指令。現在，每當您使用 STRMQMLSR 指令時，接聽器都會預設為新埠。

或者，您可以在 STRMLSR 指令上指定不同的埠號。例如：

```
STRMQMLSR MQMNAME( queue manager name ) PORT(2500)
```

此變更會使接聽器在接聽器工作期間預設為新埠。

使用 TCP SO_KEEPALIVE 選項

如果您想要使用 SO_KEEPALIVE 選項 (如需相關資訊，請參閱第 180 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 (IFS 目錄 /QIBM/UserData/mqm/qmgrs/ 佇列管理程式名稱 中的 qm.ini):

```
TCP:
KeepAlive=yes
```

然後，您必須發出下列指令：

```
CFGTCP
```

選取選項 3 (變更 TCP 屬性)。您現在可以指定時間間隔 (分鐘)。您可以指定 1 到 40320 分鐘範圍內的值；預設值是 120。

使用 TCP 接聽器待辦事項選項

在 TCP 上接收時，會設定未完成的連線要求數目上限。這個數目可視為在 TCP 埠上等待接聽器接受要求之要求的待辦事項。

IBM i 上的預設接聽器待辦事項值是 255。如果待辦事項達到此值，則會拒絕 TCP 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，且可以稍後重試連線。

不過，若要避免此錯誤，您可以在 qm.ini 檔案中新增項目：

```
ListenerBacklog = n
```

這會置換 TCP 接聽器的預設未完成要求數上限 (255)。

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。

IBM i 在 IBM i 上定義 LU 6.2 連線

使用完整 LU 6.2 連線的模式名稱、TP 名稱及連線名稱, 來定義 LU 6.2 通訊詳細資料。

鏈結的起始端必須具有遞送登錄定義, 以補充此 CSI 物件。IBM i Programming: Work Management Guide 中提供從遠端 LU 6.2 系統管理工作要求的進一步相關資訊。

如需相關資訊, 請參閱 *Multiplatform APPC Configuration Guide* 及下表。

遠端平台	TPNAME
z/OS 或 MVS	與遠端佇列管理程式的對應端資訊相同。
IBM i	與 IBM i 系統上遞送登錄中的比較值相同。
UNIX and Linux 系統	遠端 LU 6.2 配置中定義的可呼叫交易程式。
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。

如果相同電腦上有多個佇列管理程式, 請確定通道定義中的 TP 名稱是唯一的。

相關概念

第 228 頁的『起始結束 (傳送端)』

使用 CRTMQMCHL 指令定義傳輸類型 *LU62 的通道。

第 230 頁的『已起始結束 (接收端)』

使用 CRTMQMCHL 指令定義傳輸類型為 *LU62 之訊息通道鏈結的接收端。

IBM i 起始結束 (傳送端)

使用 CRTMQMCHL 指令定義傳輸類型 *LU62 的通道。

在 IBM MQ for IBM i V5.3 或更新版本中, CSI 物件是選用的。

起始端畫面顯示在圖 LU 6.2 通訊設定畫面-起始端中。若要取得所顯示的完整畫面, 請從第一個畫面按 F10。

```
Create Comm Side Information (CRTCSI)

Type choices, press Enter.

Side information . . . . . > WINSDOA1   Name
Library . . . . . > QSYS           Name, *CURLIB
Remote location . . . . . > WINSDOA1   Name
Transaction program . . . . . > MQSERIES

Text 'description' . . . . . *BLANK

Additional Parameters

Device . . . . . *LOC           Name, *LOC
Local location . . . . . *LOC           Name, *LOC, *NETATR
Mode . . . . . JSTMOD92        Name, *NETATR
Remote network identifier . . . *LOC           Name, *LOC, *NETATR, *NONE
Authority . . . . . *LIBCRTAUT   Name, *LIBCRTAUT, *CHANGE...

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

圖 36: LU 6.2 通訊設定畫面-起始端

完成起始結束欄位，如下所示：

邊資訊

為此定義提供一個名稱，用來儲存要建立的端資訊物件，例如 WINSDOA1。

註：對於 LU 6.2，訊息通道定義與通訊連線之間的鏈結是傳送端訊息通道定義的 **連線名稱** 欄位。此欄位包含 CSI 物件的名稱。

檔案庫

儲存此定義的程式庫名稱。

CSI 物件必須可在負責處理訊息通道的程式 (例如 QSYS、QMQM 及 QGPL) 可存取的檔案庫中使用。

如果名稱不正確、遺漏或找不到，則會在通道啟動時發生錯誤。

遠端位置

指定與程式通訊的遠端位置名稱。

簡言之，此必要參數包含遠端系統上友機的邏輯單元名稱，如用於兩個系統之間通訊鏈結的裝置說明中所定義。

在遠端系統上發出指令 DSPNETA 並查看預設本端位置名稱，即可找到 **遠端位置** 名稱。

交易程式

指定遠端系統上要啟動的交易程式名稱 (最多 64 個字元)。它可以是交易程序名稱、程式名稱、通道名稱或符合遞送登錄中 **比較值** 的字串。

此為必要參數。

註：若要指定 SNA 服務異動程式名稱，請輸入服務異動程式名稱的十六進位表示法。例如，若要以十六進位表示法 21F0F0F1 指定服務交易程式名稱，您可以輸入 X'21F0F0F1'。

如需 SNA 服務異動程式名稱的相關資訊，請參閱 *SNA Transaction Programmer's Reference manual for LU Type 6.2*。

如果接收端是另一個 IBM i 系統，則會使用 **交易程式** 名稱來比對傳送端的 CSI 物件與接收端的遞送登錄。對於目標 IBM i 系統上的每一個佇列管理程式，此名稱必須是唯一的。請參閱 起始結束 (接收端) 下的 **呼叫程式** 參數。另請參閱「新增遞送項目」畫面中的 **比較資料: 比較值** 參數。

文字說明

說明 (最多 50 個字元)，提醒您此連線的預期用途。

裝置

指定用於遠端系統的裝置說明名稱。可能值包括：

*LOC

裝置由系統決定。

裝置名稱

指定與遠端位置相關聯的裝置名稱。

本端位置

指定本端位置名稱。可能值包括：

*LOC

本端位置名稱由系統決定。

*NETATR

使用系統網路屬性中指定的 LCLLOCNAME 值。

本端位置名稱

指定您的位置名稱。如果您要指出遠端位置的特定位置名稱，請指定本端位置。使用 DSPNETA 指令可以找到位置名稱。

Mode

指定用來控制階段作業的模式。此名稱與「一般程式設計介面 (CPI)-通訊模式名稱」相同。可能值包括：

*NETATR

使用網路屬性中的模式。

空白

使用八個空白字元。

模式名稱

指定遠端位置的模式名稱。

註: 因為模式決定通訊階段作業的傳輸優先順序，所以根據所傳送訊息的優先順序來定義不同模式可能很有用；例如 MQMODE_HI、MQMODE_MED 及 MQMODE_LOW。(您可以有多個 CSI 指向相同位置。)

遠端網路 ID

指定與遠端位置一起使用的遠端網路 ID。可能值包括：

*LOC

使用遠端位置的遠端網路 ID。

*NETATR

使用網路屬性中指定的遠端網路 ID。

*NONE

遠端網路沒有名稱。

遠端網路 ID

指定遠端網路 ID。請在遠端位置使用 DSPNETA 指令來尋找此網路 ID 的名稱。它是遠端位置的「本端網路 ID」。

權限管理中心

指定您提供給使用者的權限，這些使用者對物件沒有特定權限，不在授權清單上，且具有對物件沒有特定權限的群組設定檔。可能值包括：

*LIBCRTAUT

物件的公用權限取自指定檔案庫的 CRTAUT 參數。此值是在建立時決定。如果檔案庫的 CRTAUT 值在建立物件之後變更，則新值不會影響現有物件。

*CHANGE

變更權限可讓使用者對物件執行基本功能，然而使用者無法變更物件。變更權限提供物件作業權限及所有資料權限。

*ALL

使用者可以執行所有作業，但那些僅限擁有者執行或由授權清單管理權限控制的作業除外。使用者可以控制物件的存在，並指定物件的安全、變更物件，以及對物件執行基本功能。使用者可以變更物件的所有權。

*USE

使用權限提供物件作業權限及讀取權限。

*EXCLUDE

排除權限會阻止使用者存取物件。

授權清單

指定授權清單的名稱，該授權清單具有用於端資訊的權限。

IBM i 已起始結束 (接收端)

使用 CRTMQMCHL 指令定義傳輸類型為 *LU62 之訊息通道鏈結的接收端。

將 CONNECTION NAME 欄位保留空白，並確定對應的詳細資料符合通道的傳送端。如需詳細資料，請參閱 [建立通道](#)。

若要讓起始端啟動接收端通道，請在起始端將遞送登錄新增至子系統。子系統必須是配置 LU 6.2 階段作業中使用之 APPC 裝置的子系統。因此，它必須具有該裝置的有效通訊登錄。遞送登錄會呼叫啟動訊息通道接收端的程式。

使用 IBM i 指令 (例如，ADDRTGE) 來定義通訊階段作業所起始的鏈結結尾。

起始的結束畫面顯示在 [LU 6.2 通訊設定畫面-新增遞送登錄](#)中。

```

Add Routing Entry (ADDRTE)

Type choices, press Enter.

Subsystem description . . . . . QCMN      Name
Library . . . . . *LIBL      Name, *LIBL, *CURLIB
Routing entry sequence number . 1      1-9999
Comparison data:
Compare value . . . . . MQSERIES

Starting position . . . . . 37      1-80
Program to call . . . . . AMQCRC6B   Name, *RTGDTA
Library . . . . . QMAS400      Name, *LIBL, *CURLIB
Class . . . . . *SBSD        Name, *SBSD
Library . . . . . *LIBL      Name, *LIBL, *CURLIB
Maximum active routing steps . . *NOMAX 0-1000, *NOMAX
Storage pool identifier . . . . . 1      1-10

Bottom
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

```

圖 37: LU 6.2 通訊設定畫面起始結束

子系統說明

此定義所在子系統的名稱。請使用 IBM i WRKSBSD 指令來檢視及更新遞送登錄的適當子系統說明。

遞送登錄序號

子系統中用來識別此通訊定義的唯一號碼。您可以使用 1-9999 範圍內的值。

比較資料: 比較值

用來與 **交易程式** 參數啟動階段作業時所收到的字串進行比較的字串，如 [圖 1](#) 所示。字串衍生自傳送者 CSI 的「交易程式」欄位。

比較資料: 開始位置

字串中要開始比較的字元位置。

註: 起始位置欄位是字串中用於比較的字元位置，且此位置一律為 37。

要呼叫的程式

執行要呼叫以啟動階段作業之入埠訊息程式的程式名稱。

會針對預設佇列管理程式呼叫程式 AMQCRC6A。此程式隨 IBM MQ for IBM i 一起提供，並設定環境，然後呼叫 AMQCRS6A。

若為其他佇列管理程式:

- 每一個佇列管理程式在其程式庫中都有一個可呼叫的特定 LU 6.2 程式。此程式稱為 AMQCRC6B，建立佇列管理程式時會自動產生。
- 每一個佇列管理程式都需要新增具有唯一遞送資料的特定遞送登錄。此遞送資料必須符合發出要求的系統所提供的 **交易程式** 名稱 (請參閱 [起始結束 \(傳送端\)](#))。

LU 6.2 通訊設定畫面-顯示遞送登錄中顯示範例:

```

Display Routing Entries
System: MY400
Subsystem description: QCMN      Status: ACTIVE

Type options, press Enter.
5=Display details

Start
Opt  Seq Nbr  Program      Library      Compare Value  Pos
10   *RTGDTA           'QZSCSRVR'    37
20   *RTGDTA           'QZRCRVR'    37
30   *RTGDTA           'QZHQTRG'    37
50   *RTGDTA           'QVPPRINT'   37
60   *RTGDTA           'QNPSRVR'    37
70   *RTGDTA           'QNMAPINGD'  37
80   QNMAREXECD  QSYS      'AREXECD'    37
90   AMQCRC6A   QMQMBW    'MQSERIES'   37
100  *RTGDTA           'QTFDWNLD'   37
150  *RTGDTA           'QMFRCVR'    37

F3=Exit  F9=Display all detailed descriptions  F12=Cancel

```

圖 38: LU 6.2 通訊設定畫面起始結束

在 LU 6.2 通訊設定畫面-顯示遞送登錄中，序號 90 代表預設佇列管理程式，並提供與 IBM MQ for IBM i 舊版 (即 V3R2、V3R6、V3R7 及 V4R2) 的配置的相容性。這些版次只容許一個佇列管理程式。序號 92 及 94 代表兩個額外的佇列管理程式，稱為 ALPHA 及 BETA (使用檔案庫 QMALPHA 及 QMBETA 建立)。

註: 您可以使用不同的遞送資料，為每一個佇列管理程式提供多個遞送登錄。這些登錄根據使用的類別提供不同工作優先順序的選項。

類別

用於透過此遞送登錄啟動之步驟的類別名稱及檔案庫。類別定義遞送步驟執行環境的屬性，並指定工作優先順序。必須指定適當的類別項目。例如，使用 WRKCLS 指令來顯示現有類別或建立類別。*IBM i Programming: Work Management Guide* 中提供從遠端 LU 6.2 系統管理工作要求的進一步相關資訊。

關於工作管理的附註

AMQCRC6A 工作無法利用 [工作管理](#) 中所記載的一般 IBM i 工作管理特性，因為其啟動方式與其他 IBM MQ 工作不同。若要變更 LU62 接收端工作的執行時期內容，您可以進行下列其中一項變更：

- 變更 AMQCRC6A 工作在遞送登錄上指定的類別說明
- 變更通訊登錄上的工作說明

如需配置「通訊工作」的相關資訊，請參閱 *IBM i Programming: Work Management Guide*。

配置佇列管理程式叢集

叢集提供一種機制，以簡化起始配置及進行中管理的方式來交互連接佇列管理程式。您可以定義叢集元件，以及建立和管理叢集。

開始之前

如需叢集作業概念的簡介，請參閱 [叢集](#)。

當您設計佇列管理程式叢集時，必須做出一些決策。請參閱 [範例叢集](#) 及 [設計叢集](#)。

相關工作

[第 342 頁的『將叢集主題定義移至不同的佇列管理程式』](#)

對於主題主機遞送或直接遞送叢集，您可能需要在解除佇列管理程式任務時移動叢集主題定義，或者因為叢集佇列管理程式已失敗或長時間無法使用。

相關參考

[刪除主題](#)

定義叢集的元件

叢集由佇列管理程式、叢集通道及叢集佇列組成。您可以定義叢集佇列，並修改預設叢集物件的部分層面。您可以取得自動定義通道的配置及狀態資訊，以及個別叢集傳送端通道與傳輸佇列之間的關係。

如需定義每一個叢集元件的相關資訊，請參閱下列子主題：

相關概念

[叢集的元件](#)

[叢集通道](#)

相關工作

[定義叢集主題](#)

[第 242 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。

[第 252 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。

定義叢集佇列


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。

 在 IBM MQ for z/OS 中，叢集佇列可以是佇列共用群組的成員共用的佇列。

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

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** 設為 CHANNEL，則會針對每一個叢集傳送端通道，從 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE 自動建立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：如果您搭配使用專用 SYSTEM.CLUSTER.TRANSMIT.QUEUES 與從早於 IBM WebSphere MQ 7.5 的產品版本升級的佇列管理程式，請確保 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE 將 **SHARE/NOSHARES** 選項設為 **SHARE**。

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 上，可以使用提供的範例 `CSQ4INSX` 來定義。

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。IBM 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
```

相關概念

[叢集佇列數](#)

相關參考

[ClusterChannel 名稱 \(MQCHAR20\)](#)

使用自動定義的叢集傳送端通道

在您建立起始 CLUSSDR 及 CLUSRCVR 定義以將佇列管理程式引進叢集之後，當需要將訊息移至叢集中的另一個佇列管理程式時，IBM MQ 會自動建立其他叢集傳送端通道定義。您可以檢視自動定義叢集傳送端通道的相關資訊，但無法修改它們。若要修改其行為，您可以使用通道自動定義結束程式。

開始之前

如需自動定義通道的簡介，請參閱 [自動定義叢集傳送端通道](#)。

關於這項作業

視需要由叢集建立自動定義的叢集傳送端通道，且它們會保持作用中狀態，直到使用正常斷線間隔規則將它們關閉為止。

叢集傳送端通道 (CLUSSDR) 可以自動定義，以移動應用程式訊息及內部叢集管理訊息。例如，在發佈/訂閱叢集 (其中已定義叢集主題) 中，可以在局部儲存庫之間定義通道，以允許交換「Proxy 訂閱」狀態。如果長時間不需要 (非作用中) 自動定義的 CLUSSDR 會從局部儲存庫的叢集資訊快取中移除，且在該佇列管理程式上不再可見。

Multi 在多平台上，OAM (物件權限管理程式) 不知道自動定義的叢集傳送端通道是否存在。如果您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相符叢集接收端通道上執行相同的動作。

z/OS 在 z/OS 上，您可以使用與任何其他通道相同的方式來保護自動定義叢集傳送端通道的安全。

程序

- 顯示給定叢集佇列管理程式之自動定義通道的相關資訊。

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)
```

- 顯示給定 CLUSRCVR 的自動定義通道狀態。

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)
```

- 使用通道自動定義結束程式來修改自動定義通道的行為。

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 IBM MQ 通道自動定義結束程式。例如，您可以在叢集環境中使用通道自動定義結束程式來進行下列任何修改：

- 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。

CLUSSDR 通道結束程式的名稱是從 CLUSRCVR 通道定義自動產生，因此可能不適合您的需求-尤其是通道兩端位於不同平台時。

在不同平台上，結束程式名稱的格式不同。例如：

- **z/OS** 在 z/OS 平台上，SCYEXIT (安全結束程式名稱) 參數的格式為 SCYEXIT('SECEXIT')
- **Windows** 在 Windows 平台上，SCYEXIT (安全結束程式名稱) 參數的格式為 SCYEXIT('drive:\path\library (secexit)')

註: **z/OS** 如果沒有通道自動定義結束程式，z/OS 佇列管理程式會從通道另一端的 CLUSRCVR 通道定義衍生 CLUSSDR 通道結束程式名稱。若要從非 z/OS 名稱衍生 z/OS 結束程式名稱，請使用下列演算法：

- 多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，最多使用 8 個字元的磁帶庫。

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

- 對於早於 IBM WebSphere MQ 7 的佇列管理程式，請將 **PROPCTL** 屬性設為 NONE 值。

每一個自動定義的叢集傳送端通道都以對應的叢集接收端通道為基礎。在 IBM WebSphere MQ 7 之前，叢集接收端通道沒有 **PROPCTL** 屬性，因此在自動定義的叢集傳送端通道中，這個屬性會設為 COMPAT。

如果叢集需要使用 **PROPCTL** 來移除從 IBM WebSphere MQ 7 或更新版本佇列管理程式傳送至舊版 IBM MQ 上佇列管理程式的訊息中的應用程式標頭 (例如 RFH2)，則您必須撰寫通道自動定義結束程式，將 **PROPCTL** 設為 NONE 值。

- 使用通道屬性 LOCLADDR 來控制定址的各個層面。
 - 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進行出埠通訊，則這會很有用。
 - 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，您可以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。您也必須在手動定義的 CLUSSDR 通道中指定它。
 - 如果您想要叢集中的所有佇列管理程式針對其所有出埠通訊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。

註: 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。

Multi 在多平台上，您可以設定預設本端位址值，以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。該值的格式符合 MQSC 屬性 LOCLADDR 的格式。

相關參考

本端位址 (LOCLADDR)

使用預設叢集物件

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。

如需這些物件的完整清單，請參閱 [預設叢集物件](#)。下列清單只包含您可以變更的那些物件。

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.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 (z/OS 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。

相關概念

[預設叢集物件](#)

使用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。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。此交換器的處理是自動化及交易式。

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 237 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 238 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 239 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 IBM WebSphere MQ 7.5 之前，叢集連線會共用單一叢集傳輸佇列 SYSTEM.CLUSTER.TRANSMIT.QUEUE。從 IBM WebSphere MQ 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 [叢集作業：規劃如何配置叢集傳輸佇列](#) 及第 284 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 238 頁的圖 39 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`。

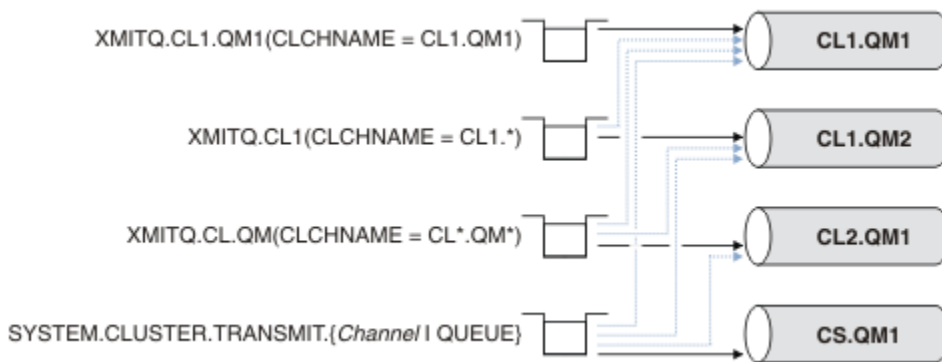


圖 39: 傳輸佇列/叢集傳送端通道優先順序

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 238 頁的圖 39 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 238 頁的圖 39 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。使用一個佇列時，您有一組要設定及查詢的叢集傳輸佇列屬性；使用多個佇列時，您有多組。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 [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。如果可以選擇連接至目的地的叢集傳送端通道，則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 241 頁的圖 40 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。

如果您想要分隔出訊息資料流量，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 241 頁的圖 40 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。

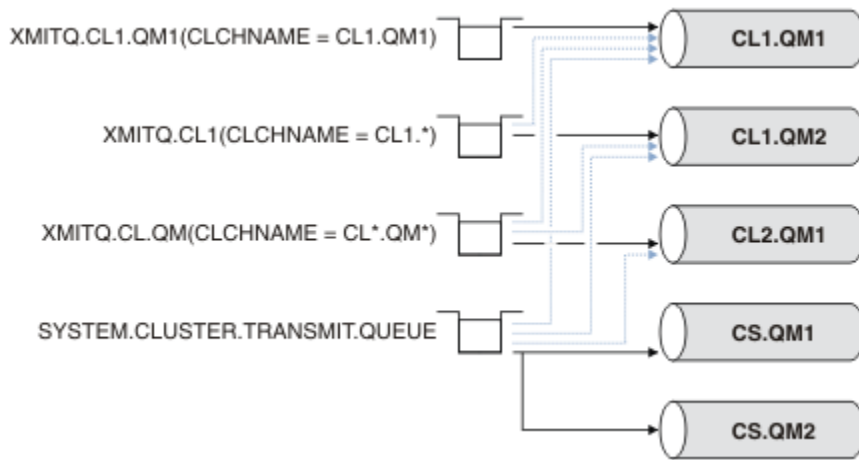


圖 40: 多個叢集傳送端通道

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 239 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

叢集通道

第 507 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

叢集作業: 使用多個叢集傳輸佇列的應用程式隔離

叢集作業: 規劃如何配置叢集傳輸佇列

第 274 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 254 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 281 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 284 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。

設定新的叢集

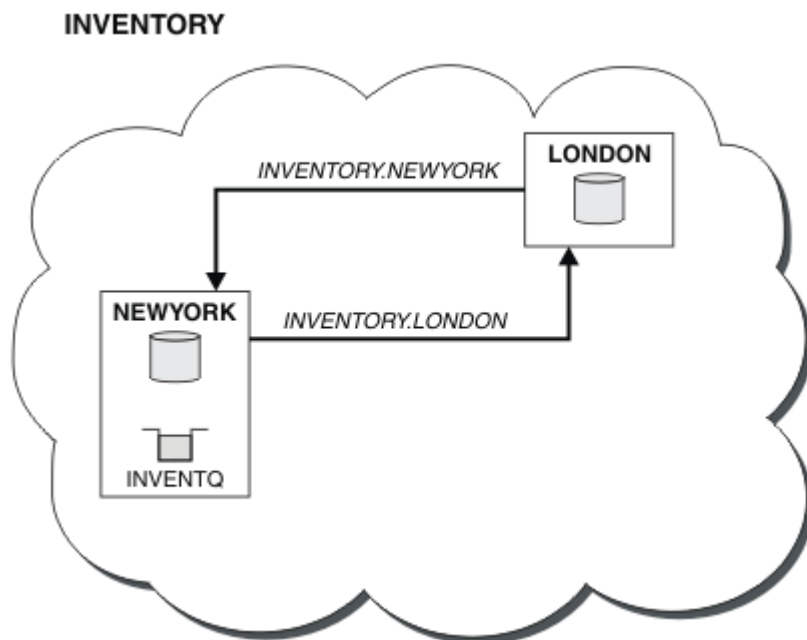
請遵循下列指示來設定範例叢集。個別指示說明在 TCP/IP、LU 6.2 上設定叢集，以及使用單一傳輸佇列或多個傳輸佇列。測試叢集的運作方式是將訊息從一個佇列管理程式傳送至另一個佇列管理程式。

開始之前

- 無需遵循這些指示，您可以使用 IBM 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，以協助您瞭解設定叢集所採取的步驟，請參閱第 233 頁的『定義叢集佇列』、叢集通道及 接聽器。

關於這項作業

您正在為鏈商店設定新的 IBM 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。



此叢集看起來如下：

您可以配置叢集中的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。

INVENTORY 是小型叢集。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。

程序

- [第 243 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 245 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 248 頁的『在 z/OS 上使用 LU 6.2 設定叢集』](#)
- [第 250 頁的『驗證叢集』](#)

相關概念

叢集

[叢集作業與分散式佇列作業的比較](#)

[叢集的元件](#)

相關工作

[第 232 頁的『配置佇列管理程式叢集』](#)

叢集提供一種機制，以簡化起始配置及進行中管理的方式來交互連接佇列管理程式。您可以定義叢集元件，以及建立和管理叢集。

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


這是說明簡式叢集的不同配置的三個主題之一。

開始之前

如需所建立叢集的概觀，請參閱 [第 242 頁的『設定新的叢集』](#)。

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

請遵循下列步驟，使用傳輸通訊協定 TCP/IP 在多平台上設定叢集。  在 z/OS 上，您必須遵循 [第 777 頁的『在 z/OS 上定義 TCP 連線』](#) 中的指示來設定 TCP/IP 連線，而不是在步驟 [第 244 頁的『4』](#) 中定義接聽器。否則，z/OS 的步驟會相同，但錯誤訊息會寫入主控台，而不是佇列管理程式錯誤日誌。

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 252 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 IBM 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: IBM 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)
```

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，您可以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。請參閱 [叢集接收端通道: CLUSRCVR](#)。CLUSRCVR 通道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，也可以輸入為 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)，且叢集不包含 z/OS 上的佇列管理程式，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。

您手動定義從每個完整儲存庫佇列管理程式到叢集中所有其他完整儲存庫佇列管理程式的 CLUSSDR 通道。請參閱 [叢集傳送端通道: CLUSSDR](#)。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。它們都需要手動定義的 CLUSSDR 通道，其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。當佇列管理程式在相同叢集中同時具有叢集接收端通道和叢集傳送端通道的定義時，會啟動叢集傳送端通道。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(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)
CONNAME(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)
CONNAME(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。

下一步

現在，您已準備好 [驗證叢集](#)。

相關工作

第 245 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](#)』
這是說明簡式叢集的不同配置的三個主題之一。

第 248 頁的『[在 z/OS 上使用 LU 6.2 設定叢集](#)』
這是其中一個樹狀結構主題，說明簡式叢集的不同配置。

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

這是說明簡式叢集的不同配置的三個主題之一。

開始之前

如需所建立叢集的概觀，請參閱 [第 242 頁的『設定新的叢集』](#)。

關於這項作業

請遵循下列步驟，使用傳輸通訊協定 TCP/IP 在多平台上設定叢集。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 254 頁的『將佇列管理程式新增至叢集: 個別傳輸佇列』](#)。

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 252 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 IBM 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: IBM 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 [第 252 頁的『將佇列管理程式加入叢集中』](#) 及 [第 254 頁的『將佇列管理程式新增至叢集: 個別傳輸佇列』](#) 主題。

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)
```

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，您可以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。請參閱 [叢集接收端通道: CLUSRCVR](#)。CLUSRCVR 通道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，也可以輸入為 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'
```

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)，且叢集不包含 z/OS 上的佇列管理程式，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')
```

8. 在 LONDON 佇列管理程式上定義 CLUSSDR 通道。

您手動定義從每個完整儲存庫佇列管理程式到叢集中所有其他完整儲存庫佇列管理程式的 CLUSSDR 通道。請參閱 [叢集傳送端通道: CLUSSDR](#)。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。它們都需要手動定義的 CLUSSDR 通道，其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。當佇列管理程式在相同叢集中同時具有叢集接收端通道和叢集傳送端通道的定義時，會啟動叢集傳送端通道。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(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)
CONNAME(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。

下一步

現在，您已準備好 [驗證叢集](#)。

相關工作

第 243 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](#)』
這是說明簡式叢集的不同配置的三個主題之一。

第 248 頁的『[在 z/OS 上使用 LU 6.2 設定叢集](#)』
這是其中一個樹狀結構主題，說明簡式叢集的不同配置。

在 z/OS 上使用 LU 6.2 設定叢集

這是其中一個樹狀結構主題，說明簡式叢集的不同配置。

開始之前

如需所建立叢集的概觀，請參閱 [第 242 頁的『設定新的叢集』](#)。

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 252 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。

- 您可以依任何順序執行其餘步驟。
- 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。
- 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: IBM 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。

 請參閱 z/OS 及 第 780 頁的『在 LU 6.2 上接收』上的通道起始程式。

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，您可以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。請參閱 [叢集接收端通道: CLUSRCVR](#)。CLUSRCVR 通道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)
CONNNAME(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)
CONNNAME(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)
CONNNAME(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。

您手動定義從每個完整儲存庫佇列管理程式到叢集中所有其他完整儲存庫佇列管理程式的 CLUSSDR 通道。請參閱 [叢集傳送端通道: CLUSSDR](#)。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。它們都需要手動定義的 CLUSSDR 通道，其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。當佇列管理程式在相同叢集中同時具有叢集接收端通道和叢集傳送端通道的定義時，會啟動叢集傳送端通道。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(LU62)
CONNNAME(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。

下一步

現在，您已準備好 [驗證叢集](#)。

相關工作

第 243 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](#)』
這是說明簡式叢集的不同配置的三個主題之一。

第 245 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](#)』
這是說明簡式叢集的不同配置的三個主題之一。

驗證叢集

對等節點主題說明簡式叢集的三種不同配置。本主題說明如何驗證叢集。

開始之前

本主題假設您正在驗證您透過下列其中一項作業所建立的叢集：

- 第 243 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 245 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 248 頁的『[在 z/OS 上使用 LU 6.2 設定叢集](#)』。

如需已建立之叢集的概觀，請參閱 [第 242 頁的『設定新的叢集』](#)。

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 **amqspout** 在兩個佇列管理程式之間傳送訊息。

3. 在 LONDON 上，執行 **amqspout 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 242 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。請勿在其他佇列管理程式上定義叢集接收端通道 INVENTORY.PARIS 的傳送端。其他定義會在需要時自動建立。請參閱 [叢集通道](#)。

3. **z/OS**

在 IBM MQ for z/OS 上啟動通道起始程式。

4. 在佇列管理程式 PARIS 上定義 CLUSSDR 通道。

當您將不是完整儲存庫的佇列管理程式新增至叢集時，您只會定義一個叢集傳送端通道，以建立與完整儲存庫的起始連線。請參閱 [叢集傳送端通道: CLUSSDR](#)。

在 PARIS 上，對名為 INVENTORY.LONDON 且具有網址 LONDON.CHSTORE.COM 的佇列管理程式進行 CLUSSDR 通道的下列定義。

```
DEFINE CHANNEL(INVENTORY.LONDON) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from PARIS to repository at LONDON')
```

5. 選擇性的: 如果您要將先前已從相同叢集中移除的佇列管理程式新增至叢集，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:

a) 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** 指令可能會對正在運行的叢集造成干擾，此後每隔 27 天在叢集物件向所有相關佇列管理程式自動傳送狀態更新項目時也可能造成干擾。請參閱在 [大型叢集中重新整理可能影響叢集的效能及可用性](#)。

b) 重新啟動 CLUSSDR 通道

(例如，使用 [START CHANNEL](#) 指令)。

c) 重新啟動 CLUSRCVR 通道。

結果

下圖顯示此作業所設定的叢集。

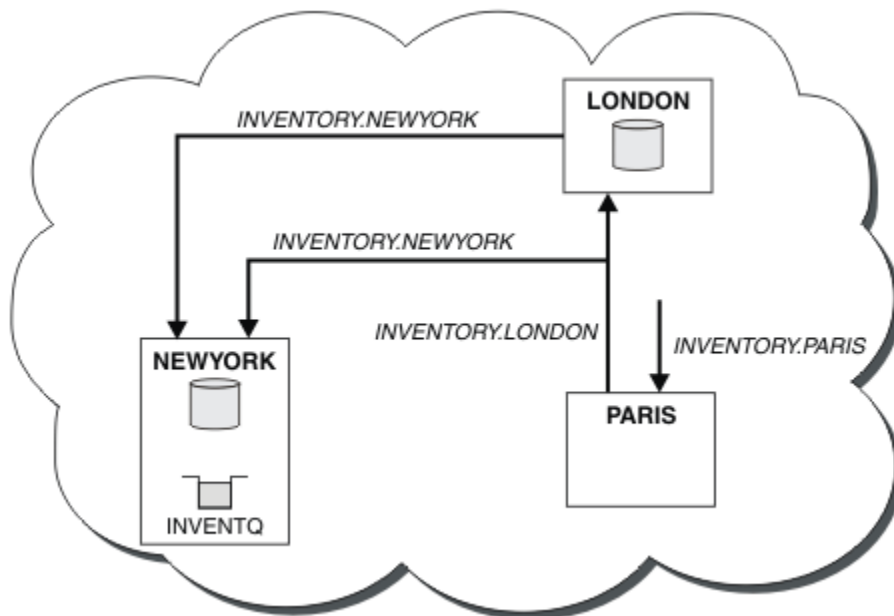


圖 41: 具有三個佇列管理程式的 INVENTORY 叢集

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 242 頁的『設定新的叢集』。

關於這項作業

此作業是第 252 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: 如果您搭配使用專用 SYSTEM.CLUSTER.TRANSMIT.QUEUES 與從早於 IBM WebSphere MQ 7.5 的產品版本升級的佇列管理程式，請確保 SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE 將 SHARE/NOSHARES 選項設為 **SHARE**。

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)  
CONNNAME(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。請勿在其他佇列管理程式上定義叢集接收端通道 INVENTORY.PARIS 的傳送端。其他定義會在需要時自動建立。請參閱 叢集通道。

4. 在佇列管理程式 PARIS 上定義 CLUSSDR 通道。

當您將不是完整儲存庫的佇列管理程式新增至叢集時，您只會定義一個叢集傳送端通道，以建立與完整儲存庫的起始連線。請參閱 [叢集傳送端通道: CLUSSDR](#)。

在 PARIS 上，對名為 INVENTORY.LONDON 且具有網址 LONDON.CHSTORE.COM 的佇列管理程式進行 CLUSSDR 通道的下列定義。

```
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。

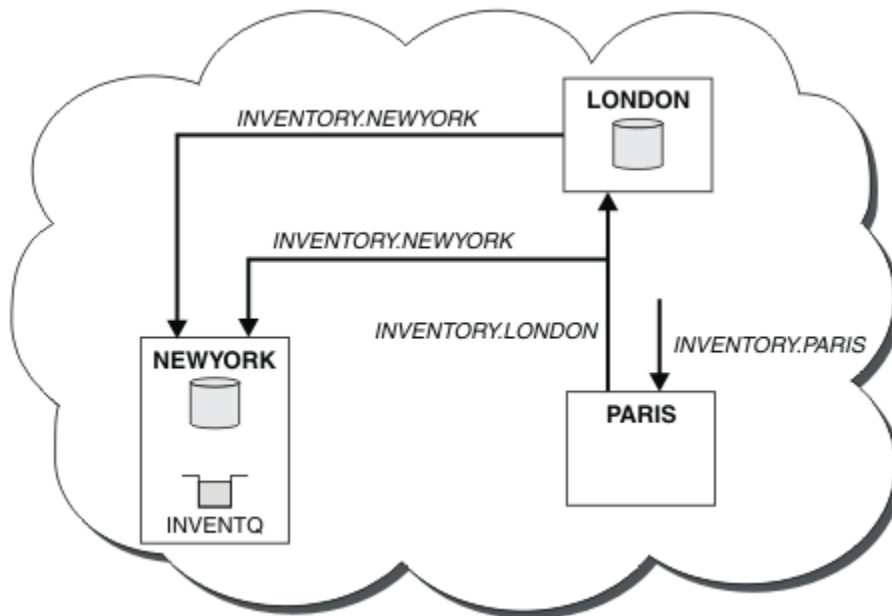


圖 42: 具有三個佇列管理程式的 INVENTORY 叢集

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。

相關工作

[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。

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 242 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 MQ 從系統中找出連線名稱。IBM 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)
CONNNAME(LONDON.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from PARIS to repository at LONDON')
```

4. 選擇性的: 如果您要將先前已從相同叢集中移除的佇列管理程式新增至叢集，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:

- a) 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** 指令可能會對正在運行的叢集造成干擾，此後每隔 27 天在叢集物件向所有相關佇列管理程式自動傳送狀態更新項目時也可能造成干擾。請參閱在大型叢集中重新整理可能影響叢集的效能及可用性。

- b) 重新啟動 CLUSSDR 通道
(例如，使用 **START CHANNEL** 指令)。
c) 重新啟動 CLUSRCVR 通道。

結果

此作業所設定的叢集與第 252 頁的『將佇列管理程式加入叢集中』相同:

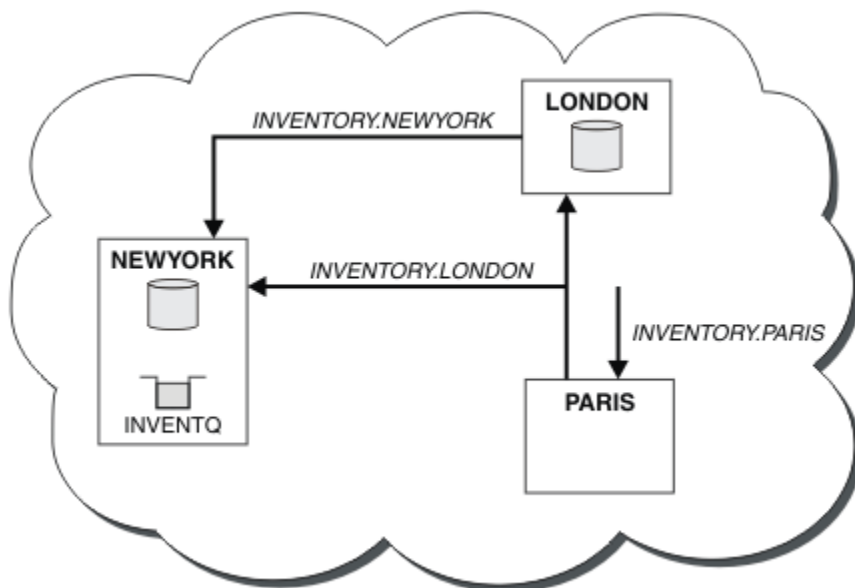


圖 43: 具有三個佇列管理程式的 INVENTORY 叢集

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 MQ 會將 +QMNAME+ 解析為佇列管理程式名稱 (LONDON)。然後，IBM MQ 會比對稱為 INVENTORY.LONDON 之通道的定義與對應的 CLUSRCVR 定義。

IBM 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。

相關工作

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。

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 252 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 262 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)
```

```
CONNNAME(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: 如果您要將先前已從相同叢集中移除的佇列管理程式新增至叢集，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：

- a) 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** 指令可能會對正在運行的叢集造成干擾，此後每隔 27 天在叢集物件向所有相關佇列管理程式自動傳送狀態更新項目時也可能造成干擾。請參閱在大型叢集中重新整理可能影響叢集的效能及可用性。

- b) 重新啟動 CLUSSDR 通道

(例如，使用 [START CHANNEL](#) 指令)。

- c) 重新啟動 CLUSRCVR 通道。

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 260 頁的圖 44 顯示此作業所設定的 INVENTORY 叢集。

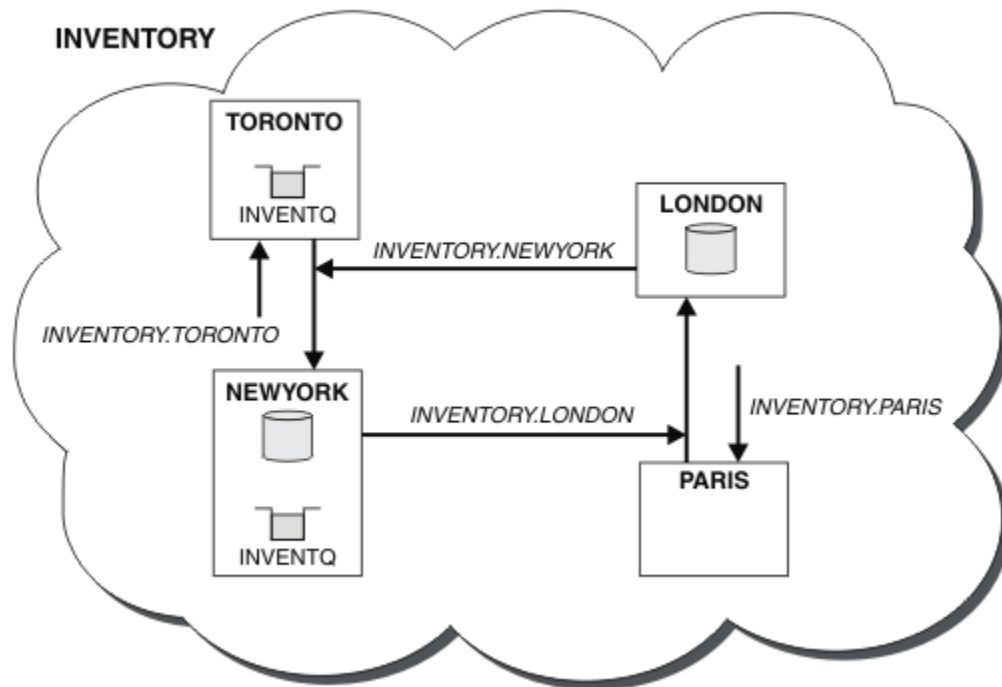


圖 44: 具有四個佇列管理程式的 INVENTORY 叢集

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

現在，您已完成所有定義，如果尚未完成，請在 IBM 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。

▶ z/OS 將佇列共用群組新增至現有叢集

將 z/OS 上的佇列共用群組新增至現有叢集。

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. 佇列共用群組僅在 IBM MQ for z/OS 上受支援。此作業不適用於其他平台。

測試情境：

- 已依照第 242 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 及 NEWYORK。
- 您想要將佇列共用群組新增至此叢集。群組 QSGP 包含三個佇列管理程式：P1、P2 及 P3。它們共用 INVENTQ 佇列的實例，由 P1 定義。

關於這項作業

請遵循下列步驟來新增管理共用佇列的新佇列管理程式。

程序

1. 請先決定佇列管理程式所參照的完整儲存庫。

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。您選擇的完整儲存庫沒有特別重要的意義。在此範例中，選擇 NEWYORK。在佇列共用群組加入叢集之後，它會與這兩個完整儲存庫進行通訊。

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 P1、P2 和 P3 上，定義：

```
DEFINE CHANNEL(INVENTORY.Pn) CHLTYPE(CLUSRCVR) TRPTYPE(TCP)
CONNAME(Pn.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-receiver channel for sharing queue manager')
```

叢集接收端通道會通告每一個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。

3. 定義佇列共用群組的 CLUSSDR 通道。

叢集的每個成員都需要定義一個叢集傳送端通道，它可以在其上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。佇列共用群組中的其中一個佇列管理程式需要下列群組定義。此定義可確保每個佇列管理程式都有叢集傳送端通道定義。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY) QSGDISP(GROUP)
DESCR('Cluster-sender channel to repository at NEWYORK')
```

4. 定義共用佇列。

在 P1 上定義佇列 INVENTQ，如下所示：

```
DEFINE QLOCAL(INVENTQ) CLUSTER(INVENTORY) QSGDISP(SHARED) CFSTRUCT(STRUCTURE)
```

在新的佇列管理程式上啟動通道起始程式及接聽器程式。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 262 頁的圖 45 顯示此作業所設定的叢集。

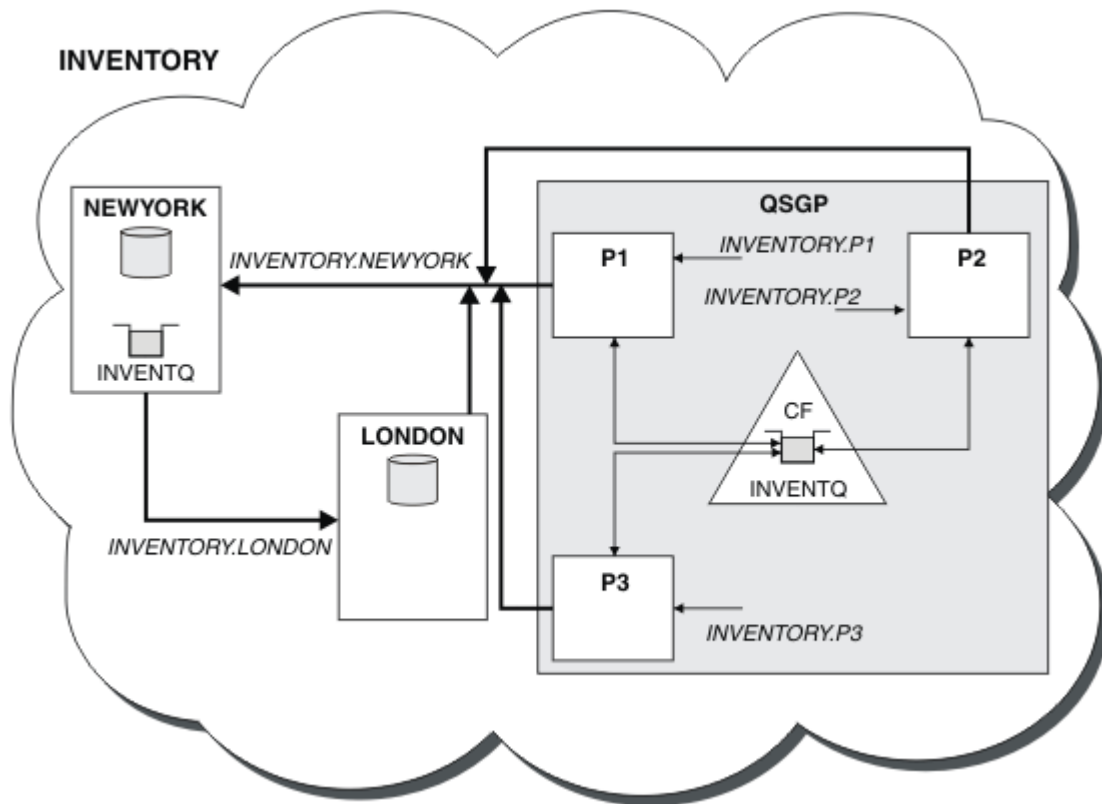


圖 45: 叢集和佇列共用群組

現在，LONDON 放置在 INVENTQ 佇列上的訊息，會輪流在通告管理該佇列的四個佇列管理程式周圍遞送。

下一步

讓佇列共用群組的成員管理叢集佇列的好處是群組的任何成員都可以回覆要求。在此情況下，在共用佇列上接收訊息之後，P1 可能變成無法使用。佇列共用群組的另一個成員可以改為回覆。

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 252 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)
```

- 將叢集中所有佇列管理程式上指向 LONDON 的所有其他手動定義叢集傳送端通道，取代為指向 NEWYORK 或 PARIS 的通道。刪除通道之後，請一律發出 **start channel** 指令，讓叢集可以再次使用自動通道。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 264 頁的圖 46 顯示此作業所設定的叢集。

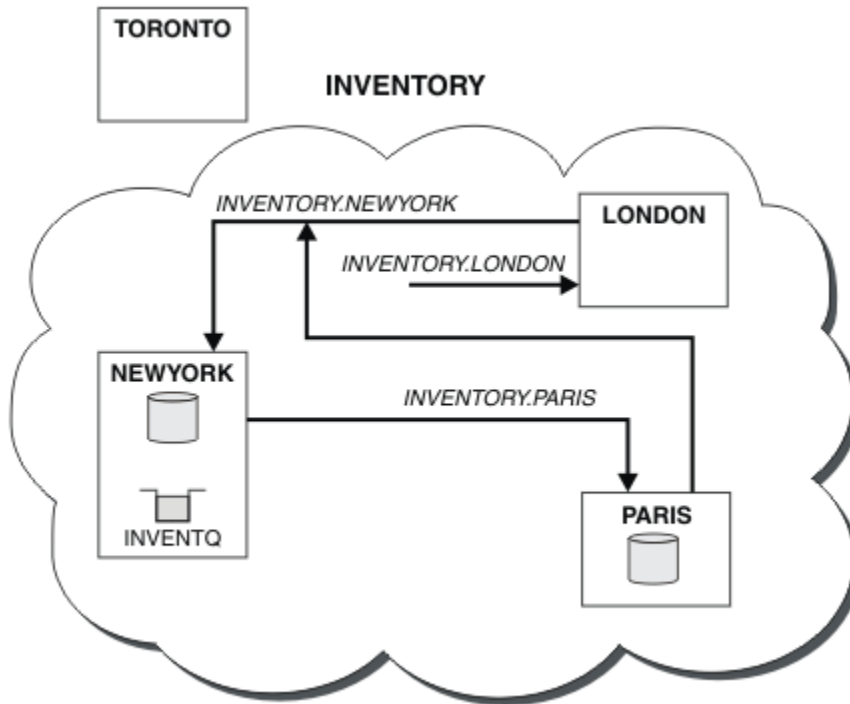


圖 46: 具有完整儲存庫的 INVENTORY 叢集已移至 PARIS

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。支援的通訊協定如下：

- 任何平台上的 TCP 或 LU 6.2
- **Windows** NetBIOS 或 Windows 系統上的 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 IBM MQ 平台上，可以使用 **START LISTENER** 指令來啟動接聽器。

Multi 在 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。

z/OS 非共用接聽器埠 (INDISP (QMGR)) 必須用於 z/OS 上的 CLUSRCVR 通道，以及用於 z/OS 的 CLUSSDR 通道。

程序

1. 啟動通道起始程式。

- **z/OS** 在 z/OS 上，每一個佇列管理程式都有一個通道起始程式，且會以個別位址空間來執行。在佇列管理程式啟動期間，您可以使用 **MQSC START CHINIT** 指令來啟動它。
- **ULW** 在 UNIX, Linux, and Windows 上，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。
- **IBM i** 在 IBM i 上，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。

- **z/OS** 在 z/OS 上，使用 IBM MQ 提供的通道接聽器程式。若要啟動 IBM MQ 通道接聽器，請使用您在啟動通道起始程式時發出的 **MQSC** 指令 **START LISTENER**。例如：

```
START LISTENER PORT(1414) TRPTYPE(TCP)
```

或：

```
START LISTENER LUNAME(LONDON.LUNAME) TRPTYPE(LU62)
```

佇列共用群組的成員可以使用共用接聽器，而非每一個佇列管理程式的接聽器。請勿將共用接聽器與叢集搭配使用。具體而言，請勿讓 CLUSRCVR 通道的 CONNAME 成為佇列共用群組的共用接聽器位址。如果您這麼做，佇列管理程式可能會收到其沒有定義之佇列的訊息。

- **IBM i** 在 IBM i 上，使用 IBM MQ 提供的通道接聽器程式。如果要啟動 IBM MQ 通道接聽器，請使用 **CL** 指令 **STRMLSR**。例如：

```
STRMQMLSR MQMNAME(QM1) PORT(1414)
```

- **Windows** 在 Windows 上，請使用 IBM MQ 提供的通道接聽器程式，或作業系統提供的機能。

若要啟動 IBM MQ 通道接聽器，請使用 `RUNMQLSR` 指令。例如：

```
RUNMQLSR -t tcp -p 1414 -m QM1
```

- **Linux** **UNIX** 在 UNIX and Linux 上，請使用 IBM MQ 所提供的通道接聽器程式，或作業系統所提供的機能；例如，用於 TCP 通訊的 `inetd`。

如果要啟動 IBM MQ 通道接聽器，請使用 `runmqlsr` 指令。例如：

```
runmqlsr -t tcp -p 1414 -m QM1
```

若要使用 `inetd` 來啟動通道，請配置兩個檔案：

- a. 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 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。

- b. 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 IBM 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：

AIX 在 AIX 上：

```
refresh -s inetd
```

Solaris **Linux** 在 Solaris 或 Linux 上：

- a. 使用下列指令尋找 `inetd` 的程序 ID：

```
ps -ef | grep inetd
```

- b. 執行適當的指令。

若為 Solaris 9 和 Linux：

```
kill -1 inetd processid
```

若為 Solaris 10 或更新版本：

```
inetconv
```

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 242 頁的『設定新的叢集』中，透過第 262 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 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 267 頁的圖 47 中說明此網路。

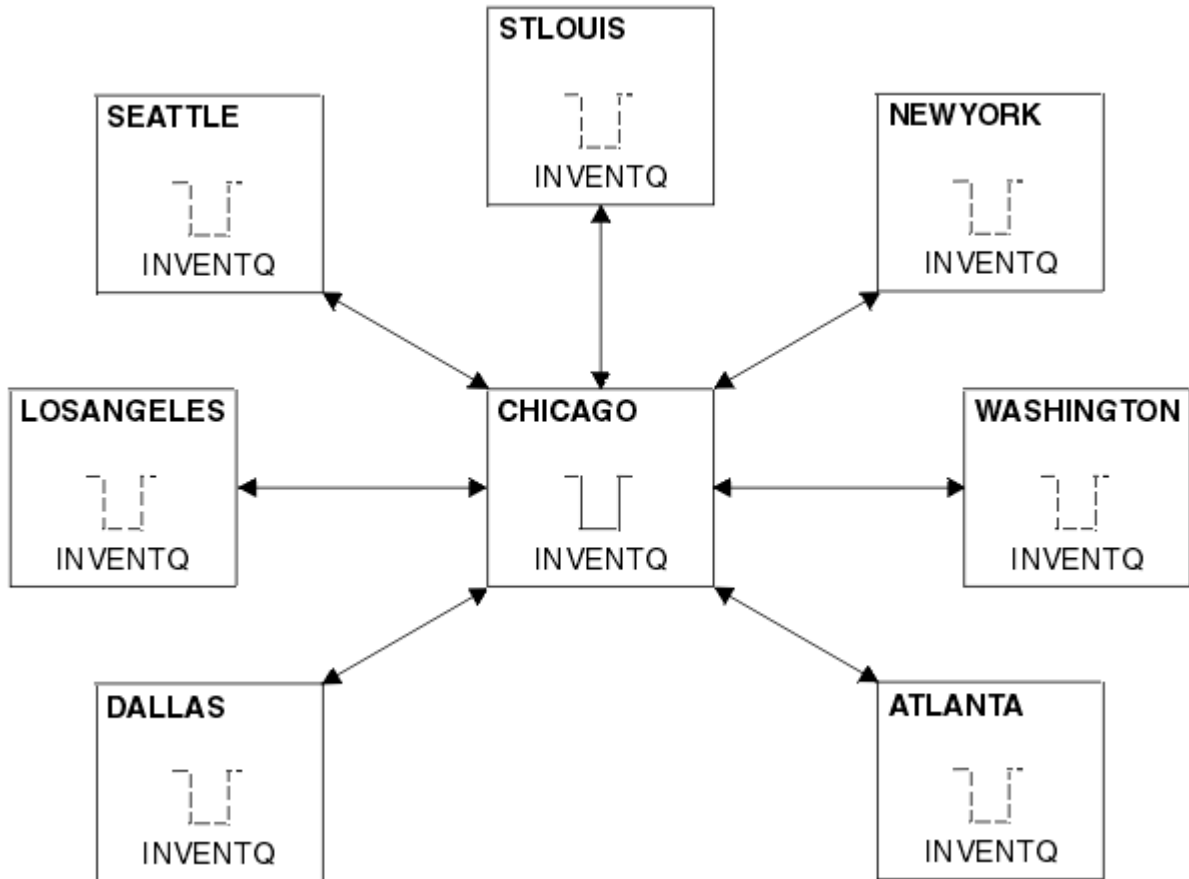


圖 47: 中心及分支網路

- 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_name` 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 334 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)  
CONNAME(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)  
CONNAME(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 266 頁的『將現有網路轉換為叢集』中的說明來設定 IBM MQ 叢集。
- 將實作稱為 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，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 CLUSQMR
```

11.

結果

此作業所設定的叢集顯示在 [第 272 頁的圖 48](#) 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。

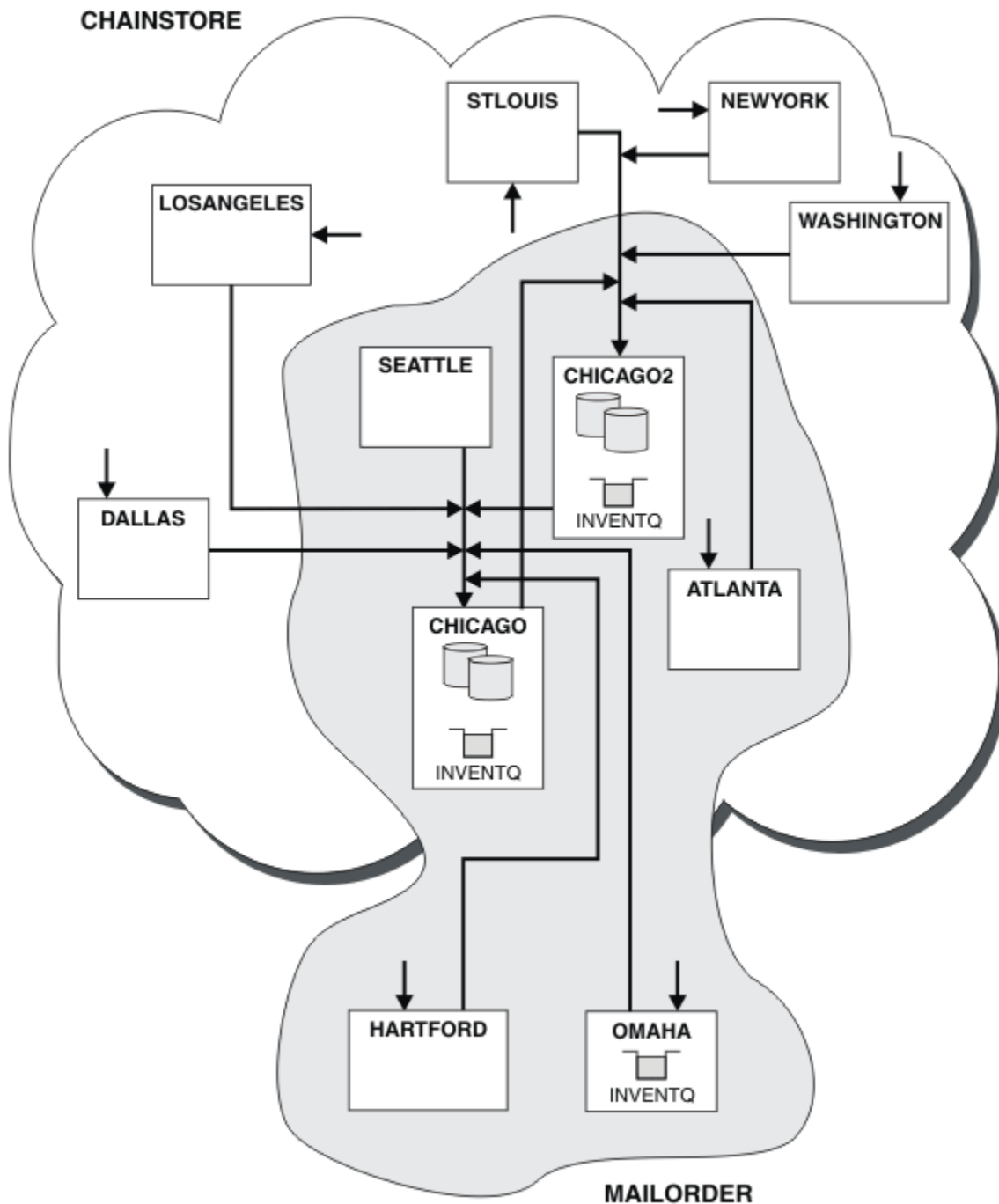


圖 48: 交互連接叢集

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。

相關工作

移除叢集網路

從網路中移除叢集，並還原分散式佇列配置。

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 266 頁的『將現有網路轉換為叢集』中的說明來設定 IBM 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_UNKOWNN_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 298 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。

相關工作

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。

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 275 頁的圖 49 中顯示用來說明隔離叢集訊息資料流量的範例叢集配置。[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。

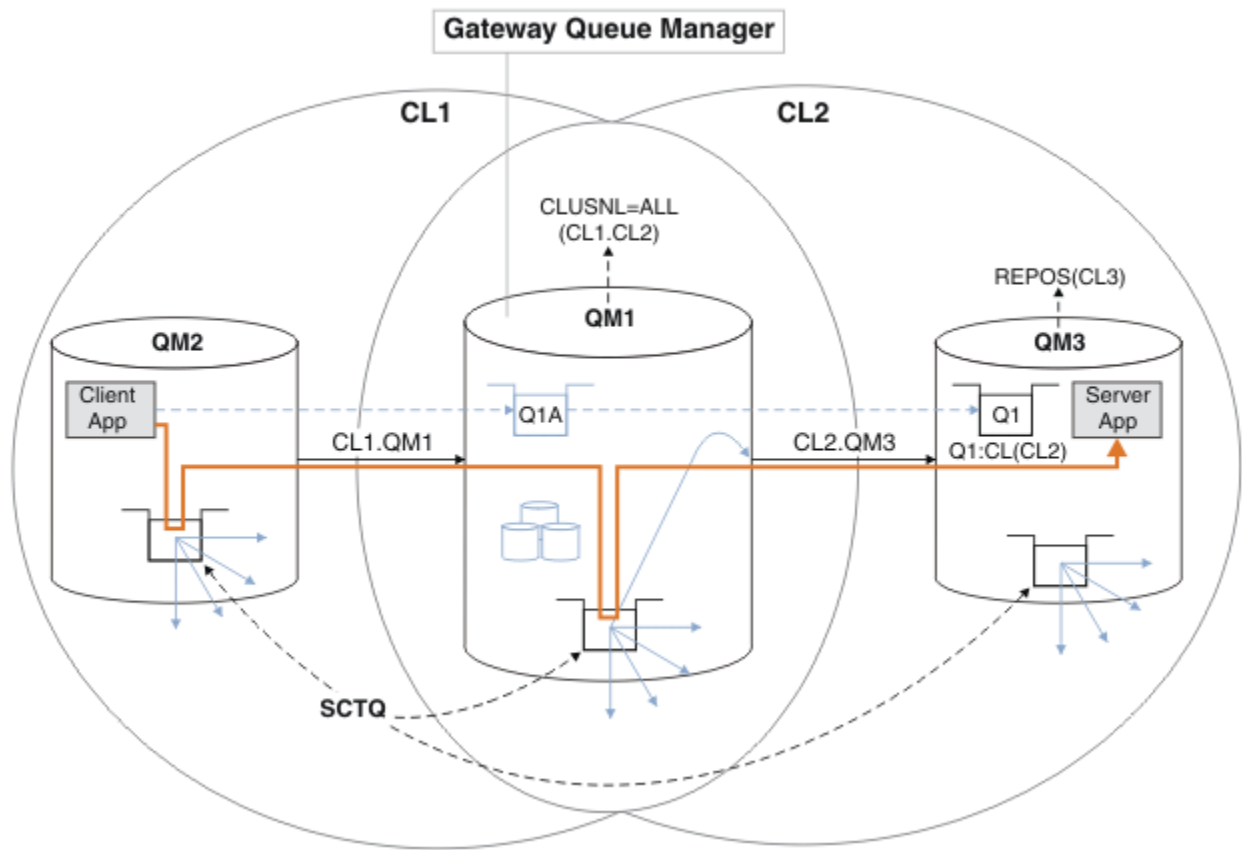


圖 49: 使用 IBM MQ 叢集部署至中心及分支架構的主從架構應用程式

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 [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 QM n
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 QM n
DEFINE LISTENER(TCP141 n) TRPTYPE(TCP) IPADDR(hostname) PORT(141 n) CONTROL(QMGR) REPLACE
START LISTENER(TCP141 n)
```

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 276 頁的表 27 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 QM m
DEFINE CHANNEL(CL c.QM n) CHLTYPE(CLUSSDR) CONNAME('localhost(141 n)') CLUSTER(CL c) REPLACE
DEFINE CHANNEL(CL c.QM m) CHLTYPE(CLUSRCVR) CONNAME('localhost(141 m)') CLUSTER(CL c) 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 **amqspu**t，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: 關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。

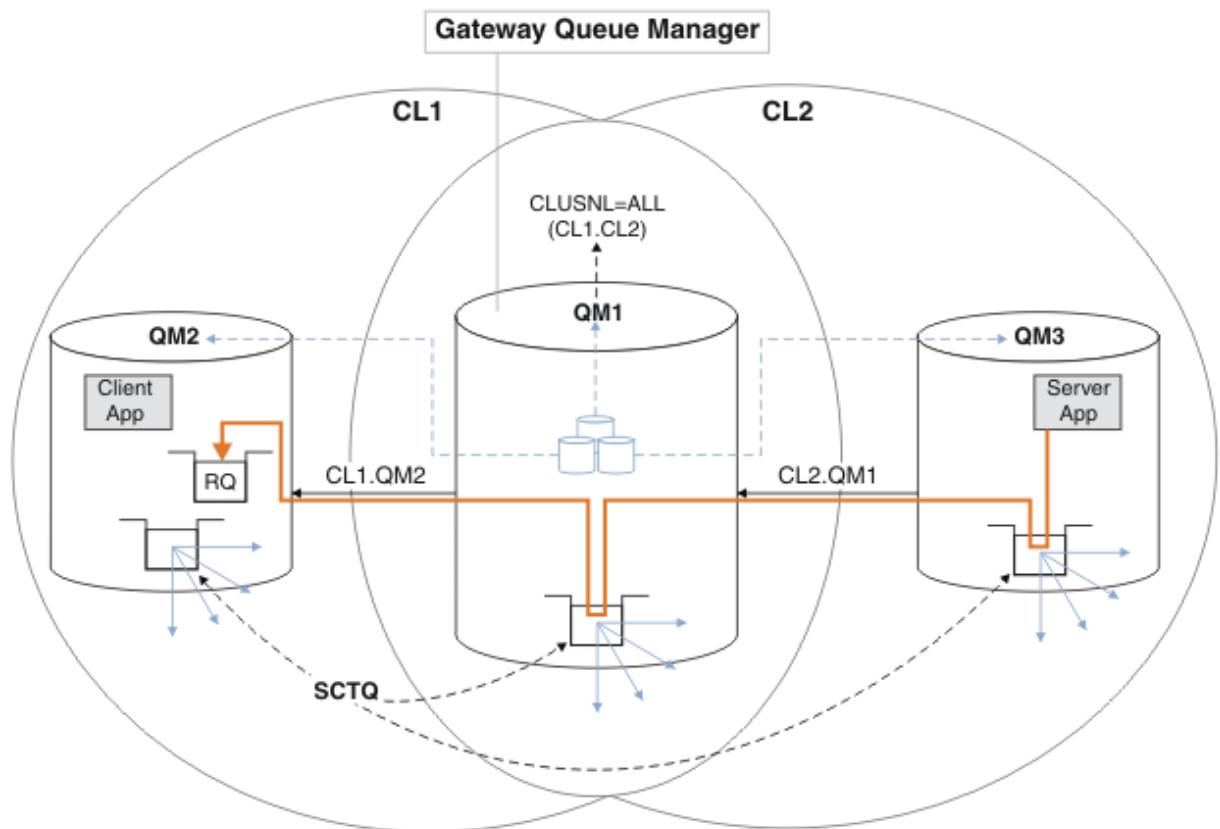


圖 50: 使用佇列管理程式別名將回覆訊息傳回至不同的叢集

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: IBM 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

[存取控制及多個叢集傳輸佇列](#)

相關工作

[叢集作業: 使用多個叢集傳輸佇列的應用程式隔離](#)

[叢集作業: 規劃如何配置叢集傳輸佇列](#)

[第 254 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。

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

Construct the overlapping clusters shown in [使用 IBM MQ 叢集部署至中心及分支架構的主從架構應用程式](#) in [第 274 頁的『使用閘道佇列管理程式建立兩個重疊的叢集』](#) by following the steps in that task.

關於這項作業

解決方案使用分散式佇列作業，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 280 頁的圖 51](#) 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 281 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。

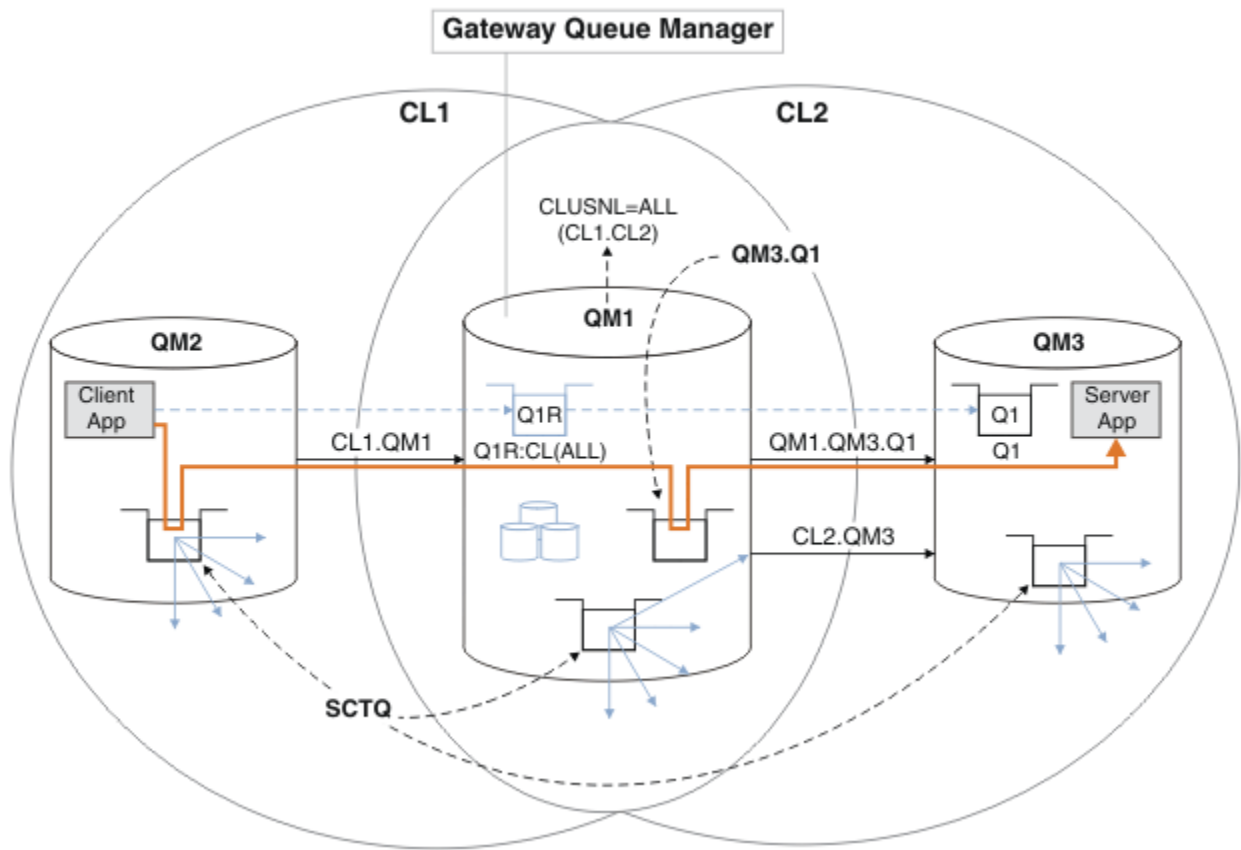


圖 51: 使用遠端佇列定義部署至中心及分支叢集架構的主從架構應用程式

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。

```
DEFINE QREMOTE CLUSNL(ALL) RNAME(Q1) RQMNAME(QM3) XMITQ(QM3.Q1) REPLACE
```

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 **amqspout**，以放置訊息。

```
C:\IBM\MQ>amqspout 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

[存取控制及多個叢集傳輸佇列](#)

相關工作

[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。

[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。

[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。

[叢集作業: 使用多個叢集傳輸佇列的應用程式隔離](#)

[叢集作業: 規劃如何配置叢集傳輸佇列](#)

[第 254 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。

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 IBM WebSphere MQ 7.5 或更新版本上。
2. Construct the overlapping clusters shown in [使用 IBM MQ 叢集部署至中心及分支架構的主從架構應用程式 in 第 274 頁的『使用閘道佇列管理程式建立兩個重疊的叢集』](#) by following the steps in that task.

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 282 頁的圖 52](#)。

此解決方案比 [第 279 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 282 頁的圖 52 中隔離叢集訊息資料流量的配置不會導致與在第 279 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一樣大的資料流量隔離。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 282 頁的圖 52 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 284 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』。

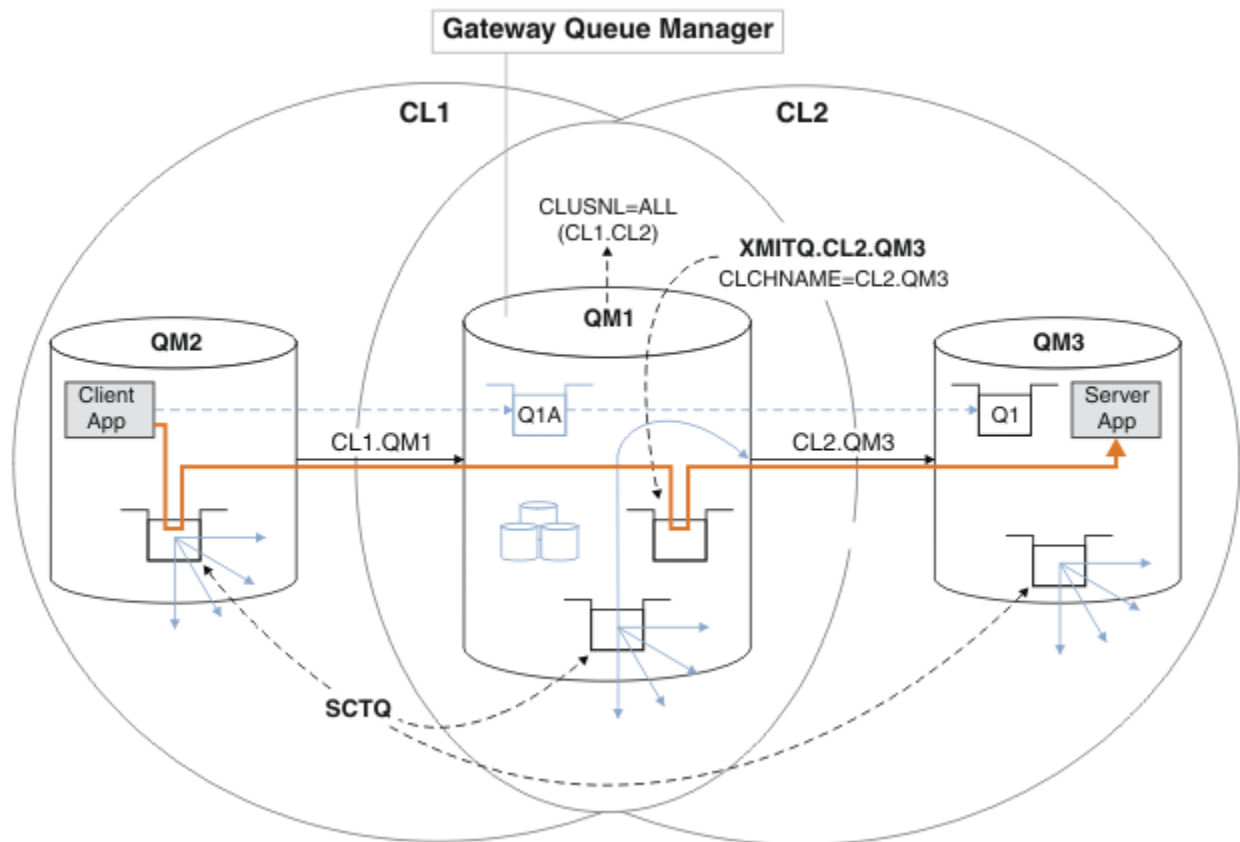


圖 52: 使用其他叢集傳輸佇列部署至中心及分支架構的主從架構應用程式。

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 IBM 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)
CONNNAME(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 IBM 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

存取控制及多個叢集傳輸佇列

[第 237 頁的『使用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。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。此交換器的處理是自動化及交易式。

相關工作

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。

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。

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開道佇列管理程式上的叢集訊息。

叢集作業: 使用多個叢集傳輸佇列的應用程式隔離

叢集作業: 規劃如何配置叢集傳輸佇列

[第 254 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。

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 282 頁的圖 52](#) 中說明的配置。

1. 閘道佇列管理程式必須位於 IBM WebSphere MQ 7.5 或更新版本上。
2. Construct the overlapping clusters shown in [使用 IBM MQ 叢集部署至中心及分支架構的主從架構應用程式 in 第 274 頁的『使用閘道佇列管理程式建立兩個重疊的叢集』](#) by following the steps in that task.
3. 在 [第 281 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 282 頁的圖 52](#) 中的步驟，以建立解決方案而不需要其他叢集。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 281 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。在這項作業中，您將在 QM3 上新增叢集 CL3 來隔離 Q1，以變更第 281 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 285 頁的圖 53 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。修改閘道佇列管理程式上的叢集名單，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 MQ 無法自動將訊息從您在第 281 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。

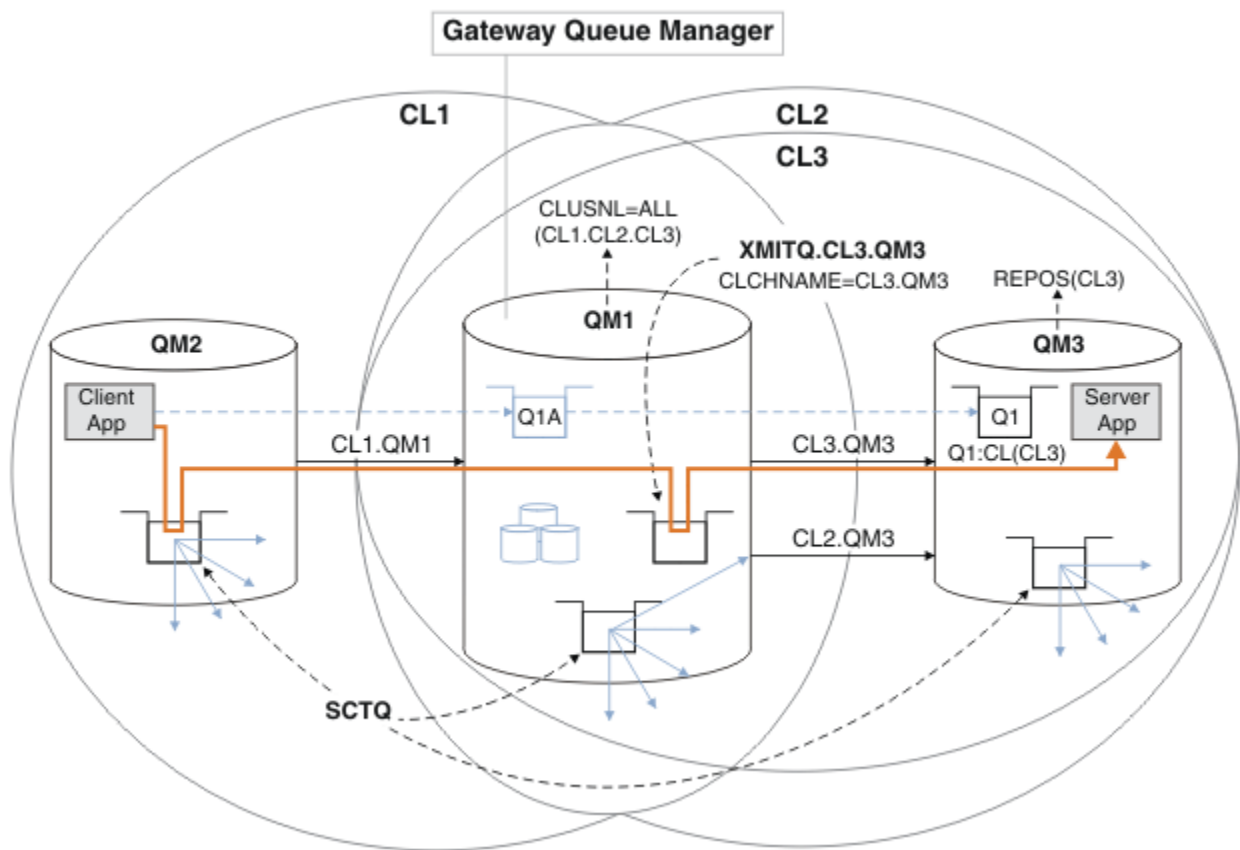


圖 53: 使用額外叢集來區隔閘道佇列管理程式中的訊息資料流量，該閘道佇列管理程式會進入相同佇列管理程式上的許多叢集佇列之一

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)
```

- 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
```

- 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)
```

- 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
```

- 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](#)。

- 將 QM3 上的目標佇列 Q1 從 CL2 變更為 CL3。

```
*... On QM3
ALTER QLOCAL(Q1) CLUSTER(CL3)
```

- 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
```

- 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
```

- 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)
```

- 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。作為回應，當下次啟動叢集傳送端通道時，IBM 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 IBM 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 IBM 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 **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

存取控制及多個叢集傳輸佇列

第 237 頁的『[使用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。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。此交換器的處理是自動化及交易式。

相關工作

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。

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。

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。

叢集作業: [使用多個叢集傳輸佇列的應用程式隔離](#)

叢集作業: [規劃如何配置叢集傳輸佇列](#)

第 254 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。

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 IBM WebSphere MQ 7.5 或更新版本上。
2. Construct the overlapping clusters shown in 使用 IBM MQ 叢集部署至中心及分支架構的主從架構應用程式 in 第 274 頁的『使用閘道佇列管理程式建立兩個重疊的叢集』 by following the steps in that task.

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 IBM WebSphere MQ 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 289 頁的圖 54。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 254 頁的『將佇列管理程式新增至叢集: 個別傳輸佇列』。

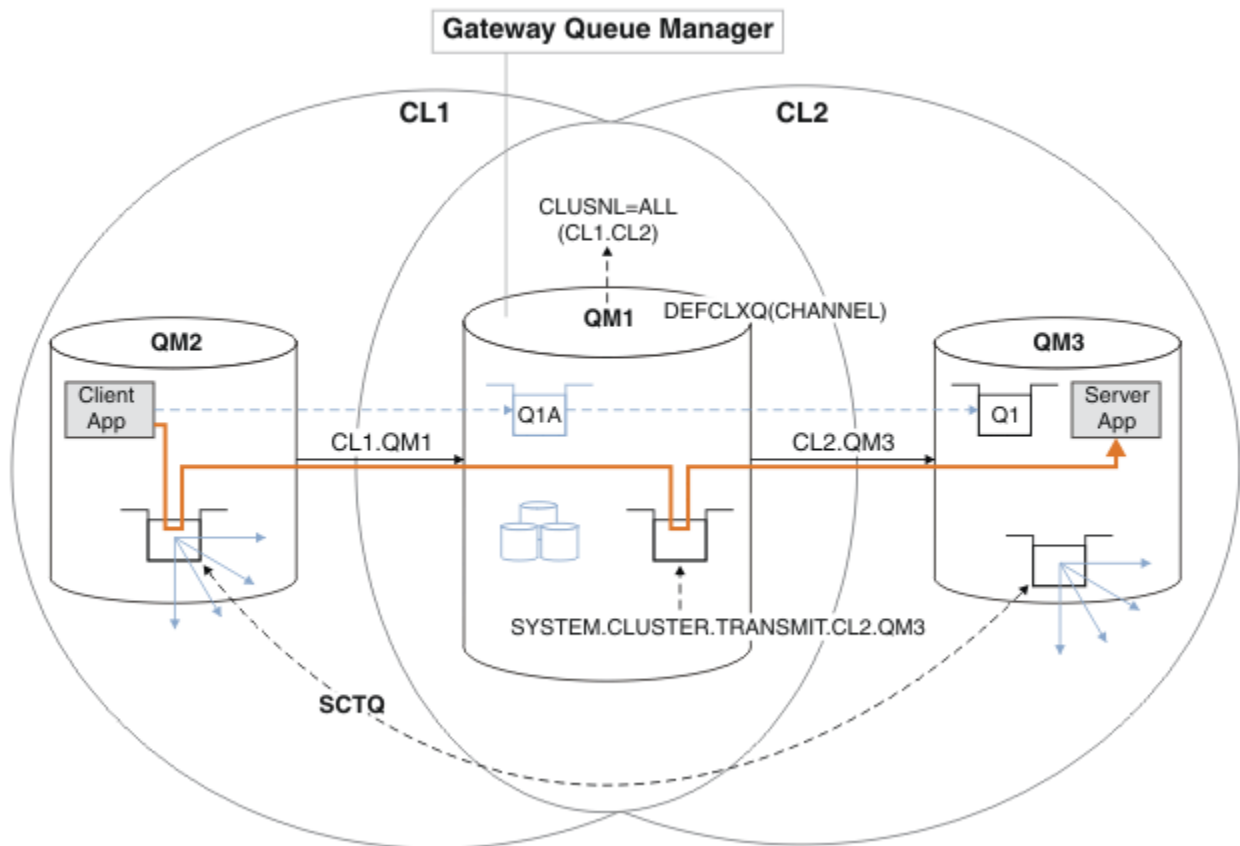


圖 54: 將主從架構應用程式部署至閘道佇列管理程式上具有個別叢集傳輸佇列的 hub 及輪輻架構。

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)
CONNNAME(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)
CONNNAME(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 IBM 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 IBM 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)
CONNNAME(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)
CONNNAME(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)
CONNNAME(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)
CONNNAME(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

[存取控制及多個叢集傳輸佇列](#)

相關工作

[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。

[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。

[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。

叢集作業: 使用多個叢集傳輸佇列的應用程式隔離

叢集作業: 規劃如何配置叢集傳輸佇列

第 254 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。

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 258 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 NAMLIST(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)
```

從叢集中移除佇列管理程式: 最佳作法

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 297 頁的『從叢集中移除佇列管理程式: 替代方法』。

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 252 頁的『將佇列管理程式加入叢集中』中所述，並如第 293 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. 在從叢集中移除佇列管理程式之前，請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 262 頁的『將完整儲存庫移至另一個佇列管理程式』完成步驟 1-6。如果不要將要移除之佇列管理程式的完整儲存庫功能移至不同的佇列管理程式，則只需要完成步驟 5 及 6。
- 如果佇列管理程式管理叢集佇列，請從第 293 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，如第 342 頁的『將叢集主題定義移至不同的佇列管理程式』中所述。

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。

2. 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('')
```

3. 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(CLUSDR) 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。

- #### 4. 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.PARIS) 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)
```

如果此通道正在使用手動定義的傳輸佇列，且 CLCHNAME 型樣不符合任何其他現有或計劃的通道，則您可能想要刪除傳輸佇列。例如：

```
DELETE QLOCAL(PARIS.CUSTOM.XMITQ)
```

註：如果自動定義傳輸佇列或共用 SYSTEM.CLUSTER.TRANSMIT.QUEUE 正在使用中，不需要此步驟。

移除的佇列管理程式可以稍後再新增回叢集中，如第 252 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 298 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。

相關工作

[從叢集中移除佇列管理程式 \(使用 IBM MQ Explorer\)](#)

相關參考

[ALTER CHANNEL \(變更通道設定\)](#)

[DISPLAY CHANNEL \(顯示通道定義\)](#)

[DISPLAY CHSTATUS \(顯示通道狀態\)](#)

[DISPLAY CLUSQMGR \(顯示叢集佇列管理程式的通道資訊\)](#)

[停止通道 \(停止通道\)](#)

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 295 頁的『從叢集中移除佇列管理程式: 最佳作法』](#)。

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 252 頁的『將佇列管理程式加入叢集中』](#) 中所述，並如 [第 293 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. 在從叢集中移除佇列管理程式之前，請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 262 頁的『將完整儲存庫移至另一個佇列管理程式』](#) 完成步驟 1-6。如果不要將要移除之佇列管理程式的完整儲存庫功能移至不同的佇列管理程式，則只需要完成步驟 5 及 6。
 - 如果佇列管理程式管理叢集佇列，請從 [第 293 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，如 [第 342 頁的『將叢集主題定義移至不同的佇列管理程式』](#) 中所述。
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。
2. 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)
```

3. 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。

4. 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)
```

5. 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 252 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 298 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。

相關參考

[DELETE CHANNEL \(刪除通道\)](#)

[DISPLAY CHANNEL \(顯示通道定義\)](#)

[DISPLAY CHSTATUS \(顯示通道狀態\)](#)

[DISPLAY CLUSQMGR \(顯示叢集佇列管理程式的通道資訊\)](#)

[停止通道 \(停止通道\)](#)

[RESET CLUSTER \(重設叢集\)](#)

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。

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 295 頁的『從叢集中移除佇列管理程式: 最佳作法』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](#)

[叢集作業與分散式佇列作業的比較](#)

[叢集元件](#)

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。

IBM MQ 會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。  在 z/OS 上，將叢集傳輸佇列的 INDXTYPE 設為 CORRELID。

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」。
- 請確定叢集傳輸佇列儲存在  (例如 z/OS 頁集) 上的媒體不會變滿。

關於這項作業



下列程序僅適用於 z/OS。

程序

將叢集傳輸佇列的 INDXTYPE 設為 CORRELID

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 [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 MQ 可確保不會從佇列中遺失任何資料。

相關資訊

叢集作業：使用 REFRESH CLUSTER 最佳作法

回復叢集佇列管理程式

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** 指令可能會對正在運行的叢集造成干擾，此後每隔 27 天在叢集物件向所有相關佇列管理程式自動傳送狀態更新項目時也可能造成干擾。請參閱[在大型叢集中重新整理可能影響叢集的效能及可用性](#)。

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。

相關資訊

叢集作業：使用 REFRESH CLUSTER 最佳作法

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 KAINTE 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。

z/OS 開啟 IBM MQ for z/OS

使用 ALTER QMGR 上的 ADOPTMCA 及 ADOPTCHK 參數控制通道的叢集接收端問題。

Multi 開啟多平台

使用 qm.ini 檔案或 Windows NT Registry 中的 AdoptNewMCA、AdoptNewMCATimeout 及 AdoptNewMCACheck 屬性，控制仍在作用中通道的叢集接收端問題。

檢查分散式網路的非同步指令是否已完成

在分散式網路中使用時，許多指令都是非同步的。視指令及發出時的網路狀態而定，可能需要大量時間才能完成。佇列管理程式在完成時不會發出訊息，因此您需要其他方法來檢查指令是否已完成。

關於這項作業

幾乎您對叢集所做的任何配置變更都可能非同步完成。這是因為在叢集內運作的內部管理和更新週期。對於發佈/訂閱階層，任何會影響訂閱的配置變更可能會非同步完成。從指令名稱來看，這並不總是顯而易見的。

下列 MQSC 指令可能全部非同步完成。其中每一個指令都有 PCF 對等項目，且大部分也可以從 IBM MQ Explorer 內取得。在沒有工作量的小型網路上執行時，這些指令通常會在幾秒內完成。然而，大型且忙碌的網路卻不是如此。此外，**REFRESH CLUSTER** 指令可能需要更長的時間，尤其是同時在多個佇列管理程式上發出時。

若要確信這些指令已完成，請檢查預期的物件是否存在於遠端佇列管理程式上。

程序

• ALTER QMGR

對於 ALTER QMGR PARENT 指令，請使用 DISPLAY PUBSUB TYPE(PARENT) ALL 來追蹤所要求母項關係的狀態。

若為 ALTER QMGR REPOS 及 ALTER QMGR REPOSNL 指令，請使用 DISPLAY CLUSQMGR QMTYPE 來確認完成。

• DEFINE CHANNEL、ALTER CHANNEL 及 DELETE CHANNEL

對於表格 ALTER CHANNEL 參數中列出的所有參數，請使用 DISPLAY CLUSQMGR 指令來監視變更何時延伸到叢集。

• DEFINE NAMELIST、ALTER NAMELIST 及 DELETE NAMELIST。

如果您在 QMgr 物件的 CLUSNL 屬性上使用 NAMELIST，則佇列或叢集通道可能會影響該物件。適當監視受影響的物件。

SYSTEM.QPUBSUB.QUEUE.NAMELIST 的變更可能會影響在發佈/訂閱階層中建立或取消 Proxy 訂閱。請使用 DISPLAY SUB SUBTYPE(PROXY) 指令來監視此情況。

• DEFINE 佇列、ALTER 佇列及 DELETE 佇列。

對於表格 DISPLAY QUEUE 指令可傳回的參數中列出的所有參數，請使用 DISPLAY QCLUSTER 指令來監視變更何時延伸到叢集。

• DEFINE SUB 和 DELETE SUB

當您在主題字串上定義第一個訂閱時，您可以在發佈/訂閱階層或發佈/訂閱叢集中建立 Proxy 訂閱。同樣地，當您刪除主題字串的前次訂閱時，您可以取消發佈/訂閱階層或發佈/訂閱叢集中的 Proxy 訂閱。

若要檢查定義或刪除訂閱的指令是否已完成，請檢查預期的 Proxy 訂閱是否存在於分散式網路中的其他佇列管理程式上。如果您在叢集中使用直接遞送，請檢查叢集中其他局部儲存庫上是否存在預期的 Proxy 訂閱。如果您在叢集中使用主題主機遞送，請檢查相符主題主機上是否存在預期的 Proxy 訂閱。請使用下列 MQSC 指令：

```
DISPLAY SUB(*) SUBTYPE(PROXY)
```

當在叢集或階層中發出 MQI 呼叫時，對下列對等訂閱及取消訂閱 MQI 呼叫使用相同的檢查：

- 使用 [MQSUB](#) 進行訂閱。
- 搭配使用 [MQCLOSE](#) 與 [MQCO_REMOVE_SUB](#) 來取消訂閱。
- [DEFINE TOPIC](#)、[ALTER TOPIC](#) 及 [DELETE TOPIC](#)

如果要檢查定義、變更或刪除叢集主題的指令已完成，請顯示叢集中其他局部儲存庫中的主題 (如果您使用直接遞送) 或在其他主題主機上 (如果您使用主題主機遞送)。

對於表格 [DISPLAY TOPIC](#) 指令可傳回的參數中列出的所有參數，請使用 [DISPLAY TCLUSTER](#) 指令來監視變更何時延伸到叢集。

註：

- **CLUSTER** 參數可能會影響發佈/訂閱叢集中 Proxy 訂閱的建立或取消。
- **PROXYSUB** 和 **SUBSCOPE** 參數可能會影響在發佈/訂閱階層或發佈/訂閱叢集中建立或取消 Proxy 訂閱。
- 請使用 [DISPLAY SUB SUBTYPE\(PROXYSUB\)](#) 指令來監視此情況。
- [重新整理叢集](#)

如果您正在執行 **REFRESH CLUSTER** 指令，請輪詢叢集指令佇列深度。在尋找物件之前，請等待它達到零，並保持為零。

1. 請使用下列 MQSC 指令來檢查叢集指令佇列深度是否為零。

```
DISPLAY QL(SYSTEM.CLUSTER.COMMAND.QUEUE) CURDEPTH
```

2. 重複檢查，直到佇列深度達到零，並在後續檢查中保持零。

REFRESH CLUSTER 指令會移除並重建物件，在大型配置中可能需要很長時間才能完成。請參閱[發佈/訂閱叢集的 REFRESH CLUSTER 注意事項](#)。

- [REFRESH QMGR 類型 \(PROXYSUB\)](#)

若要檢查 **REFRESH QMGR TYPE(PROXYSUB)** 指令是否已完成，請檢查 Proxy 訂閱是否已在分散式網路中的其他佇列管理程式上更正。如果您在叢集中使用直接遞送，請檢查叢集中其他局部儲存庫上的 Proxy 訂閱是否已更正。如果您在叢集中使用主題主機遞送，請檢查是否已在相符主題主機上更正預期的 Proxy 訂閱。請使用下列 MQSC 指令：

```
DISPLAY SUB(*) SUBTYPE(PROXYSUB)
```

- [重設叢集](#)

若要檢查 **RESET CLUSTER** 指令是否已完成，請使用 [DISPLAY CLUSQMGR](#)。

- [RESET QMGR 類型 \(PUBSUB\)](#)

若要檢查 **RESET QMGR** 指令是否已完成，請使用 [DISPLAY PUBSUB TYPE\(PARENT|CHILD\)](#)。

註：**RESET QMGR** 指令可能會導致取消發佈/訂閱階層或發佈/訂閱叢集中的 Proxy 訂閱。請使用 [DISPLAY SUB SUBTYPE\(PROXYSUB\)](#) 指令來監視此情況。

- 您也可能想要監視在指令完成時趨向零佇列深度的其他系統佇列。

例如，您可能想要監視 SYSTEM.INTER.QMGR.CONTROL 佇列及 SYSTEM.INTER.QMGR.FANREQ 佇列。請參閱 [監視叢集裡的 Proxy 訂閱資料流量](#)，以及 [在發佈/訂閱網路中平衡生產者和消費者](#)。

下一步

如果這些檢查未確認非同步指令已完成，則可能發生錯誤。若要調查，請先檢查發出指令之佇列管理程式的日誌，然後 (針對叢集) 檢查叢集完整儲存庫日誌。

相關參考

 [z/OS 上 CLUSTER 指令的非同步行為](#)

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

[叢集作業與分散式佇列作業的比較](#)

叢集的元件

[第 315 頁的『佇列管理程式別名及叢集』](#)

在傳送訊息至叢集或從叢集傳送訊息時，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 318 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 317 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 232 頁的『配置佇列管理程式叢集』](#)

叢集提供一種機制，以簡化起始配置及進行中管理的方式來交互連接佇列管理程式。您可以定義叢集元件，以及建立和管理叢集。

[第 242 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 305 頁的圖 55 顯示稱為 QM3 的佇列管理程式，它位於稱為 DEMO 的叢集外部。QM3 可以是 IBM 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)
```

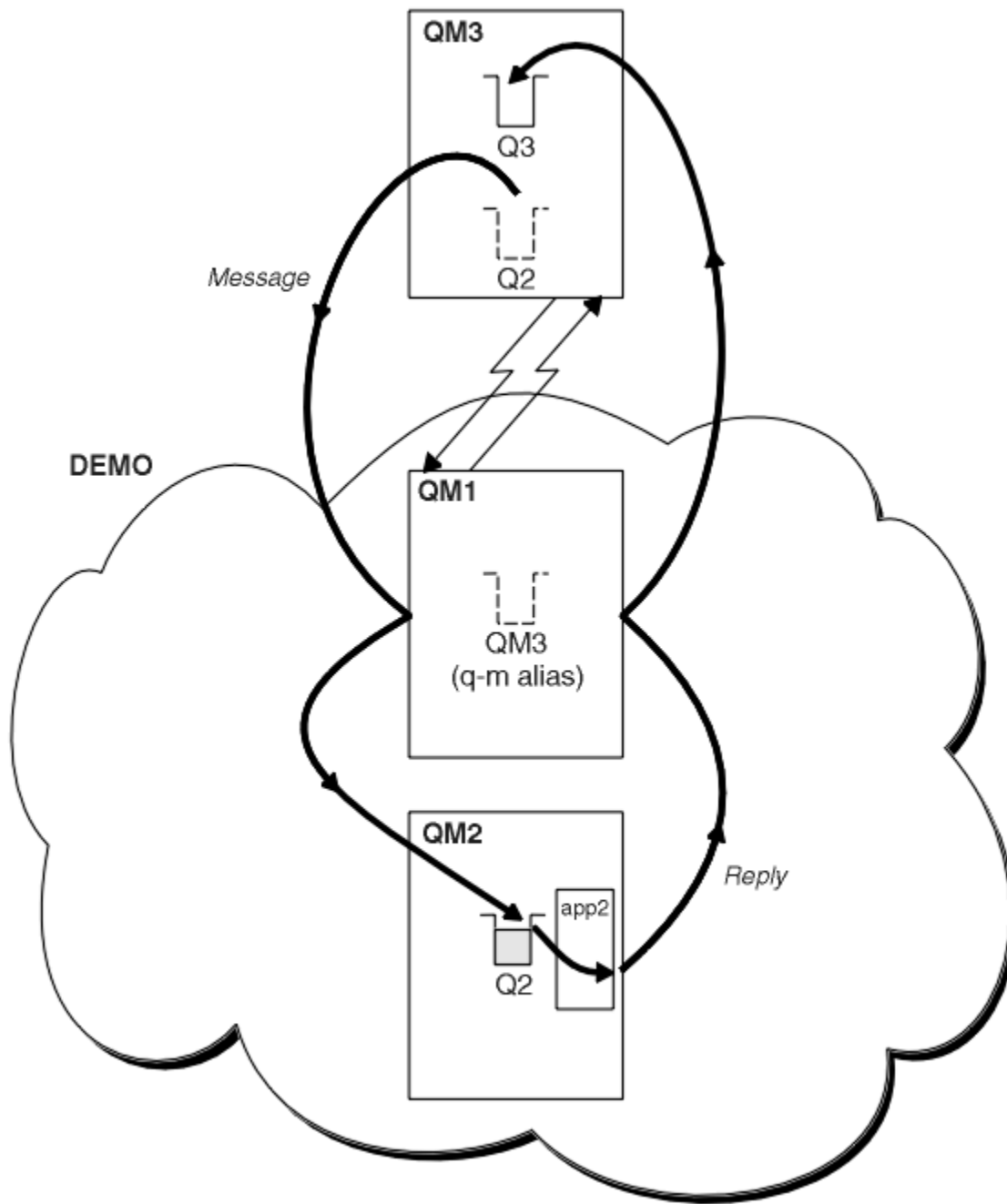



圖 55: 從叢集外部的佇列管理程式放置

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 305 頁的圖 55](#) 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。閘道 QM1 會通告叢集外部佇列管理程式 QM3 的佇列管理程式別名。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。IBM MQ 不容許您為傳輸佇列及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。閘道佇列管理程式 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。

相關概念

佇列管理程式別名及叢集

在傳送訊息至叢集或從叢集傳送訊息時，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。

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。

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。

相關工作

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。

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。

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 307 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 307 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 304 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 308 頁的圖 56](#)，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(' ')
```

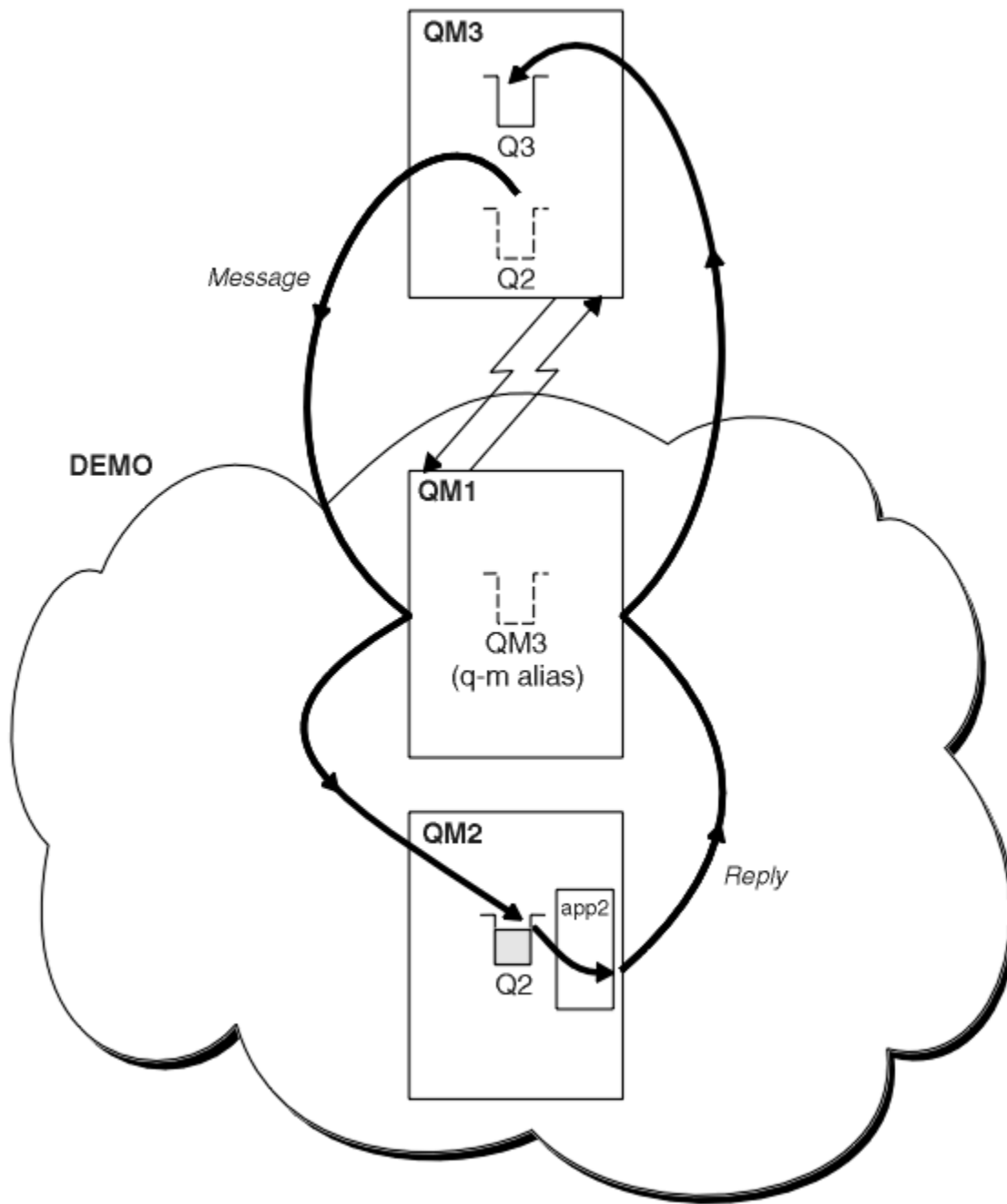


圖 56: 從叢集外部的佇列管理程式放置

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) XMIT(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 309 頁的圖 57 顯示叢集內的佇列管理程式 QM 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。

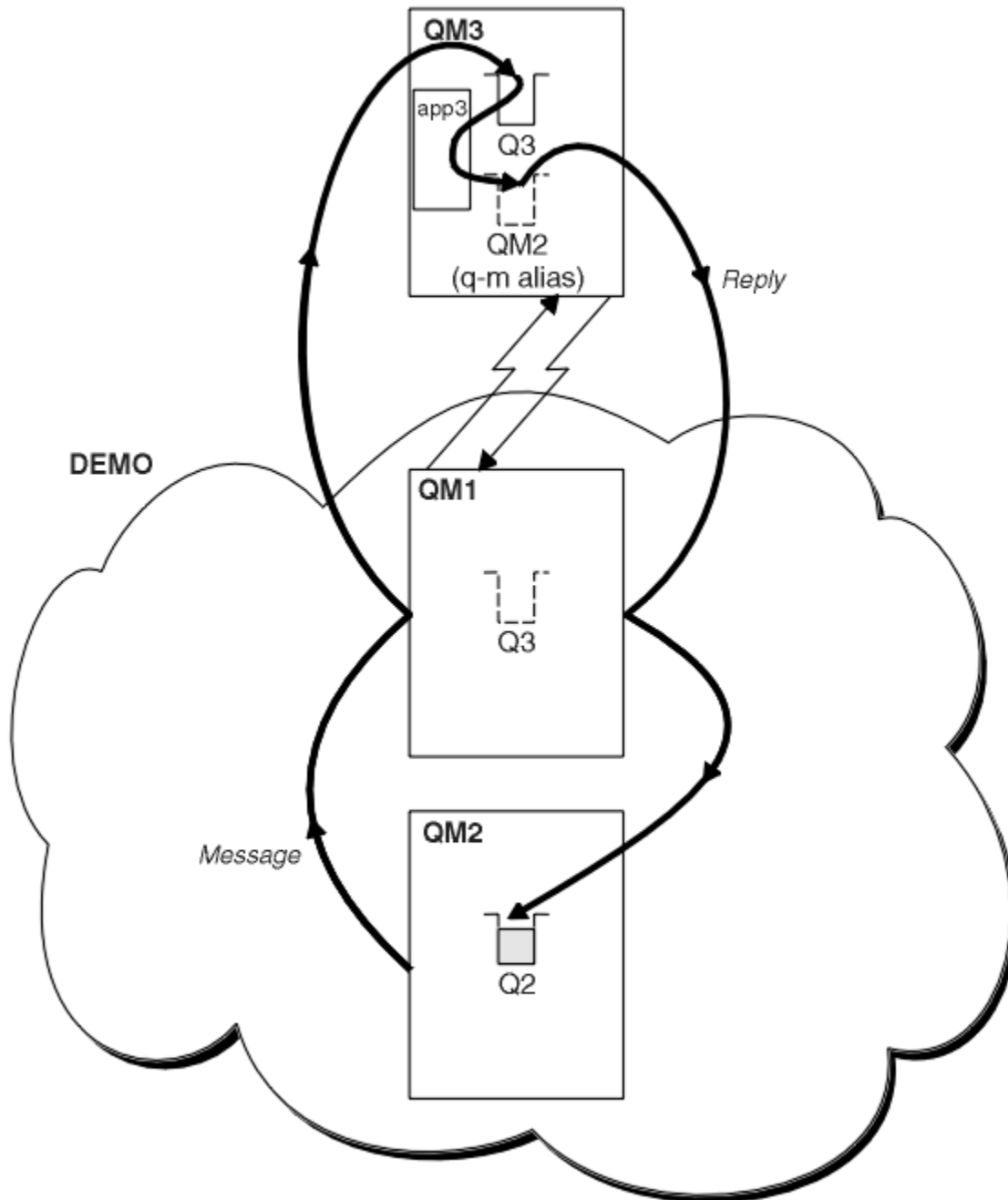


圖 57: 放入叢集外部的佇列管理程式

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。通道未顯示在 第 309 頁的圖 57 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。

相關概念

佇列管理程式別名及叢集

在傳送訊息至叢集或從叢集傳送訊息時，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。

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。

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。

相關工作

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。

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。

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。

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 304 頁的『配置叢集的要求/回覆』中的第 305 頁的圖 55 所示。

關於這項作業

在此實務範例中，叢集外部的佇列管理程式 QM3 (在第 312 頁的圖 58 中) 將要求傳送至佇列 Q2。Q2 在兩個佇列管理程式上進行管理: QM2 及 QM4 在叢集內 DEMO。這兩個佇列管理程式都配置了預設連結選項 NOTfixed，以便使用工作量平衡。來自 QM3(叢集外的佇列管理程式) 的要求會透過 QM1 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 312 頁的圖 58 中。

此程序會延伸第 304 頁的『配置叢集的要求/回覆』中第 305 頁的圖 55 的範例。

程序

1. 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。

2. 在 QM1 上建立佇列管理程式別名 Q3。

```
DEFINE QREMOTE(Q3) RNAME(' ') RQMNAME(' ')
```

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

3. 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)
```

4. QM1(開道佇列管理程式) 沒有特殊定義。

結果

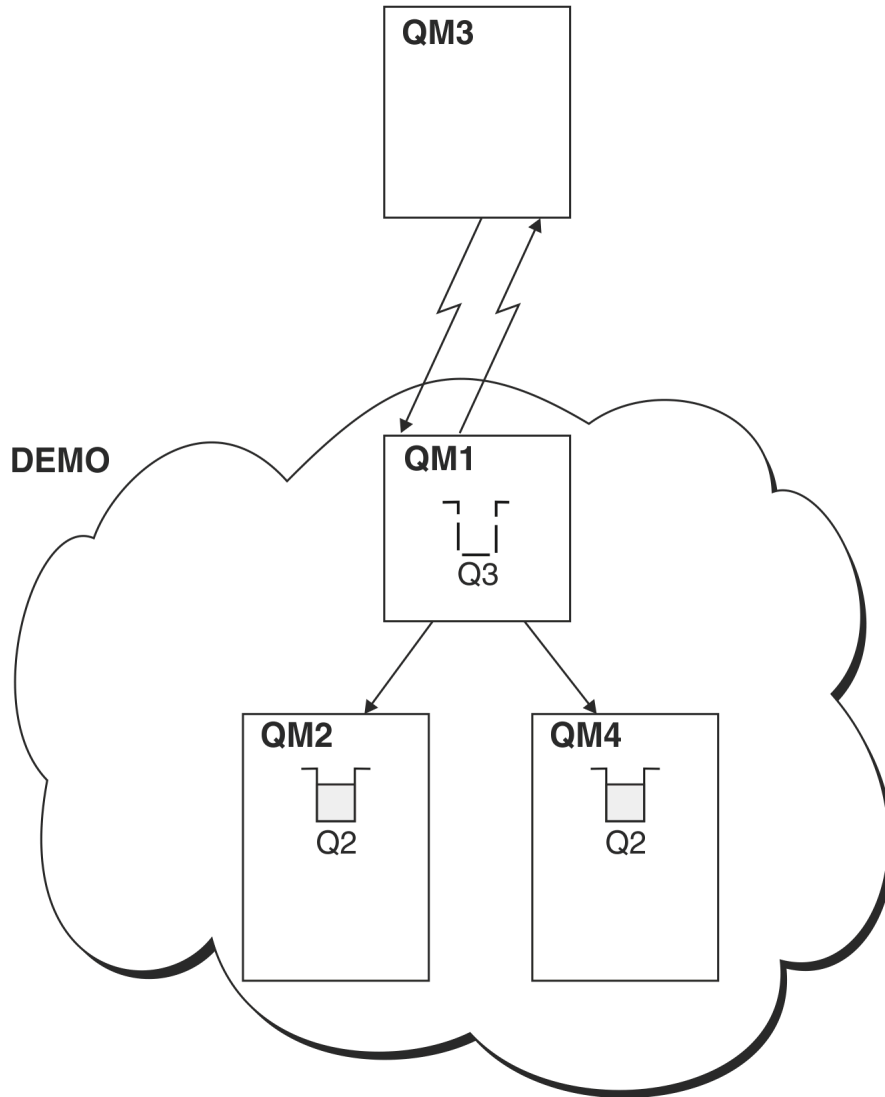


圖 58: 從叢集外部的佇列管理程式放置

當位於 QM3 的應用程式發出 MQPUT 呼叫，將訊息放置到 Q2 時，QM3 上的 QREMOTE 定義會導致訊息透過閘道佇列管理程式 QM1 遞送。當 QM1 接收訊息時，它會知道該訊息仍適用於名為 Q2 的佇列，並執行名稱解析。QM1 會檢查其本端定義，且找不到 Q2 的任何定義。然後 QM1 會檢查其叢集配置，並發現它知道叢集 DEMO 中有兩個 Q2 實例。QM1 現在可以利用工作量平衡，在 QM2 和 QM4 上的 Q2 實例之間配送訊息。

相關概念

佇列管理程式別名及叢集

在傳送訊息至叢集或從叢集傳送訊息時，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。

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。

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。

相關工作

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。

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。

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。

相關資訊

[佇列名稱解析](#)

[名稱解析](#)

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 [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。

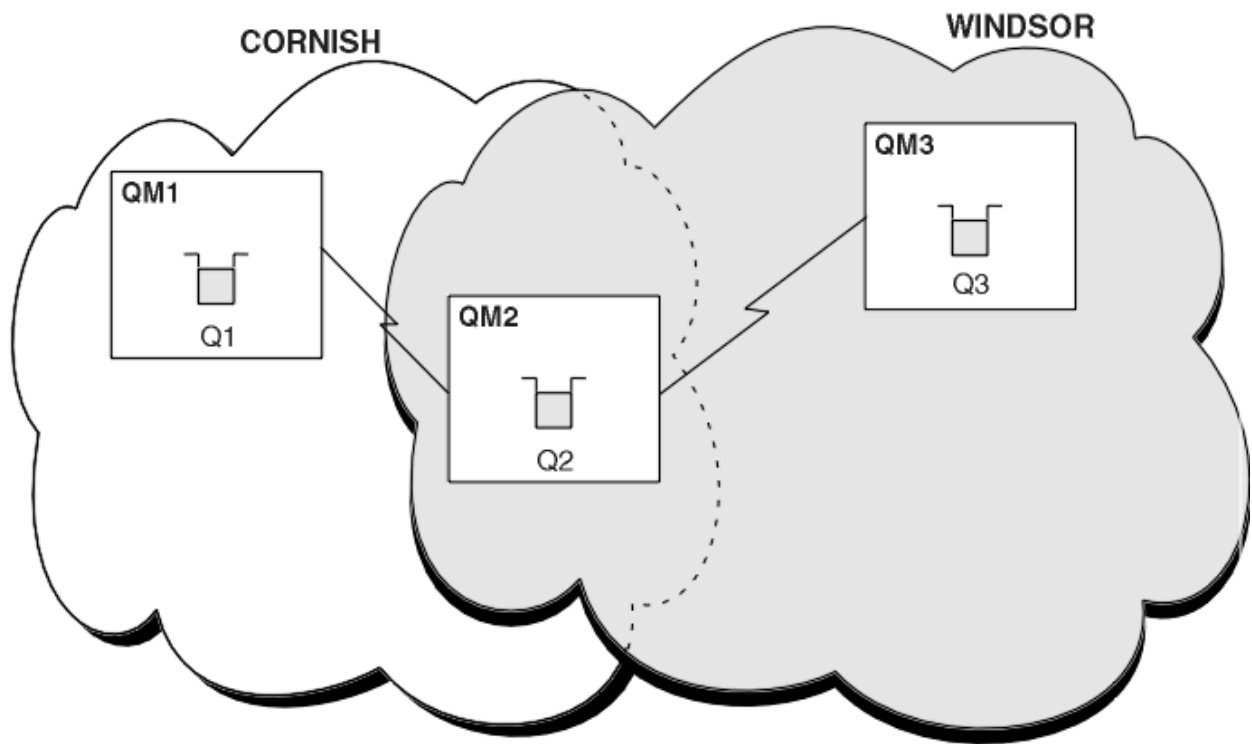


圖 59: 跨叢集橋接

第 313 頁的圖 59 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 313 頁的圖 59](#) 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。

相關概念

佇列管理程式別名及叢集

在傳送訊息至叢集或從叢集傳送訊息時，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。

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。

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。

相關工作

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。

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。

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。

佇列管理程式別名及叢集

在傳送訊息至叢集或從叢集傳送訊息時，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。

使用遠端佇列定義 (含空白 RNAME) 建立的佇列管理程式別名有五種用法：

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，則會在傳輸標頭中以佇列管理程式別名定義中的 RQMNAME 替代佇列管理程式名稱。

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。如需配置的範例，請參閱第 274 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。

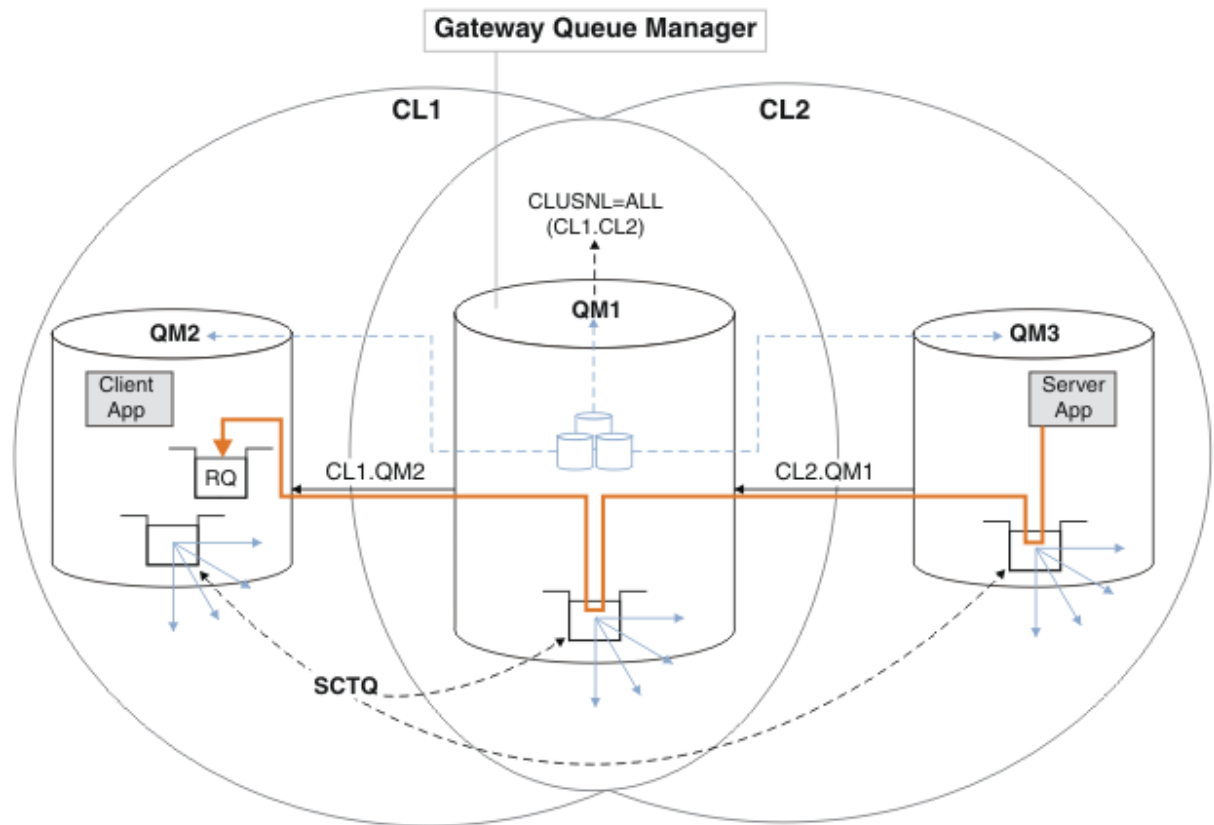


圖 60: 使用佇列管理程式別名將回覆訊息傳回至不同的叢集

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 310 頁的『從叢集外部配置工作量平衡』

相關概念

[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。

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。

相關工作

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。

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。

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。

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。

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。

相關概念

佇列管理程式別名及叢集

在傳送訊息至叢集或從叢集傳送訊息時，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。

相關工作

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。

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。

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。

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。

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。

相關概念

佇列管理程式別名及叢集

在傳送訊息至叢集或從叢集傳送訊息時，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。

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

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。

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。

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。


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。

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。  例如，在 z/OS 平行 Sysplex 中，複製的應用程式可能會存取共用

Db2 或「虛擬儲存體存取方法 (VSAM)」資料庫中的資料。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，並獨立於其他實例執行。

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地分配工作負載。

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，且應用程式在傳送訊息時不會為佇列管理程式命名。如果叢集包含相同佇列的多個實例，則 IBM MQ 會選取要將訊息遞送至其中的佇列管理程式。根據佇列管理程式及佇列的可用性，以及與佇列管理程式、佇列及通道相關聯的許多叢集工作量特定屬性，來選擇適當的目的地。請參閱 [叢集中的工作量平衡](#)。

 在「IBM MQ for z/OS」中，佇列共用群組中的佇列管理程式可以將叢集佇列當作共用佇列來管理。共用叢集佇列可供相同佇列共用群組中的所有佇列管理程式使用。例如，在具有多個相同佇列實例的叢集中，佇列管理程式 QM2 及/或 QM4 可以是共用佇列管理程式。每一個都有佇列 Q3 的定義。與 QM4 位於相同佇列共用群組中的任何佇列管理程式，都可以讀取放入共用佇列 Q3 的訊息。每一個佇列共用群組最多可以包含 32 個佇列管理程式，每一個佇列管理程式都具有相同資料的存取權。佇列共用會大幅增加訊息的傳輸量。

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

[叢集作業與分散式佇列作業的比較](#)

[分散式佇列及叢集](#)

[叢集的元件](#)

[叢集通道](#)

[針對 MQPUT 停用叢集佇列時會發生什麼情況](#)

[第 304 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

[撰寫及編譯叢集工作量結束程式](#)

[第 232 頁的『配置佇列管理程式叢集』](#)

[叢集提供一種機制，以簡化起始配置及進行中管理的方式來交互連接佇列管理程式。您可以定義叢集元件，以及建立和管理叢集。](#)

[第 242 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。](#)

[第 310 頁的『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。](#)

相關參考

[叢集傳送端通道上設定的工作量平衡無法運作](#)

[「叢集佇列監視」範例程式 \(AMQSCLM\)](#)

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 320 頁的圖 61 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。

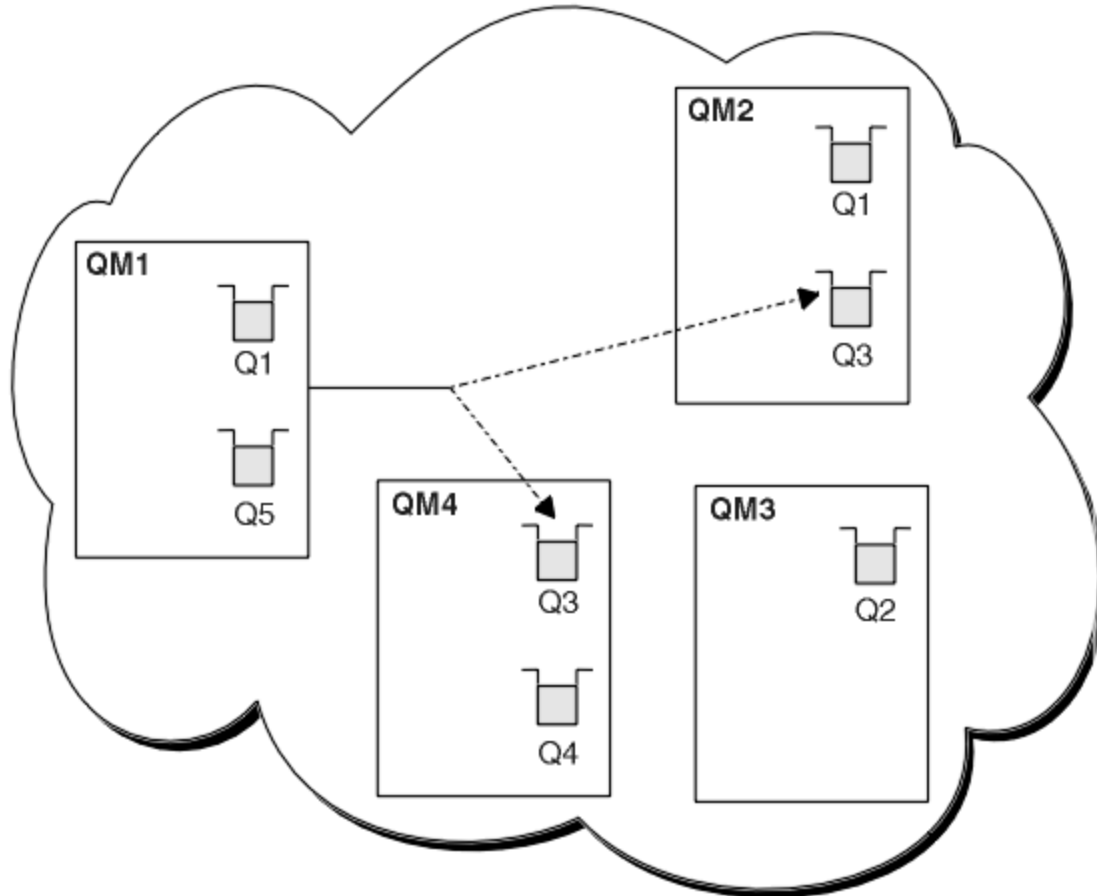


圖 61: 具有相同佇列的多個實例的叢集

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。

z/OS 在 IBM MQ for z/OS 上，在訊息重新配置處理程序完成之前，通道不會完全停止。停止模式設為 FORCE 或 TERMINATE 的通道會岔斷處理程序，因此如果您這樣做，部分 BIND_NOT_FIXED 訊息可能已重新配置給另一個通道，或訊息可能不正常。

註: **z/OS**

1. 在設定具有相同佇列的多個實例的叢集之前，請確定您的訊息彼此沒有相依關係。例如，需要以特定順序或由相同的佇列管理程式來處理。
2. 使相同佇列之不同實例的定義相同。否則，您會從不同的 MQINQ 呼叫取得不同的結果。

相關概念

[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。

相關工作

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。

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。

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。

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。

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。

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。

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

- 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)
```

- 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 [處理訊息親緣性](#)。

- 在巴黎的系統上安裝庫存應用程式。
- 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，請在 IBM 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 322 頁的圖 62 顯示此作業所設定的叢集。

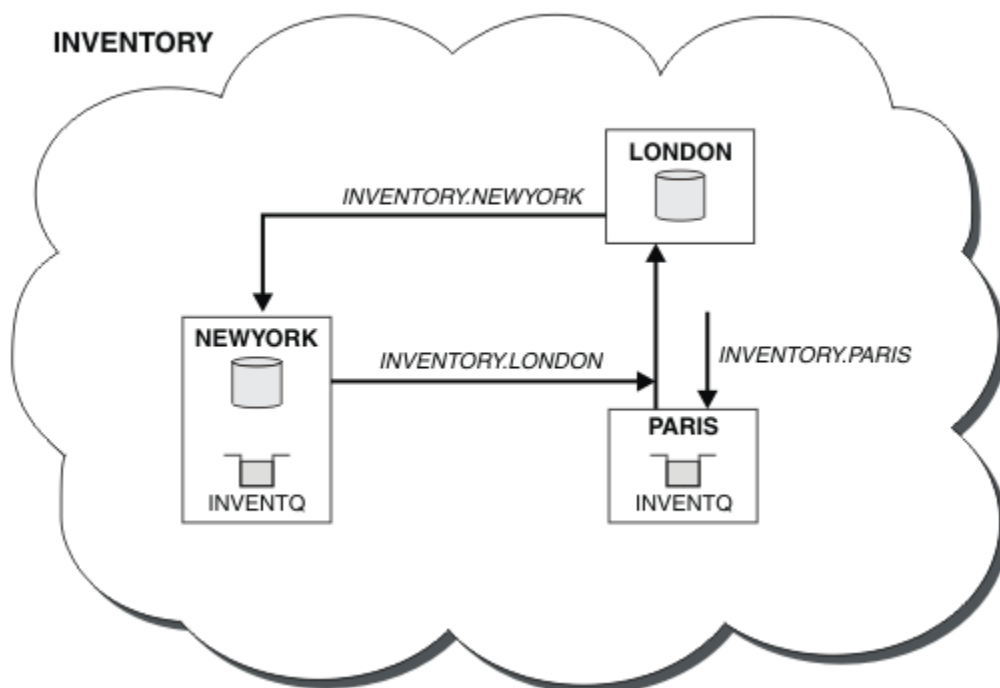


圖 62: *INVENTORY* 叢集，具有三個佇列管理程式

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。

相關概念

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。

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。

相關工作

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。

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。

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。

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。

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。

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 IBM 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 324 頁的圖 63 顯示此作業所設定的叢集。

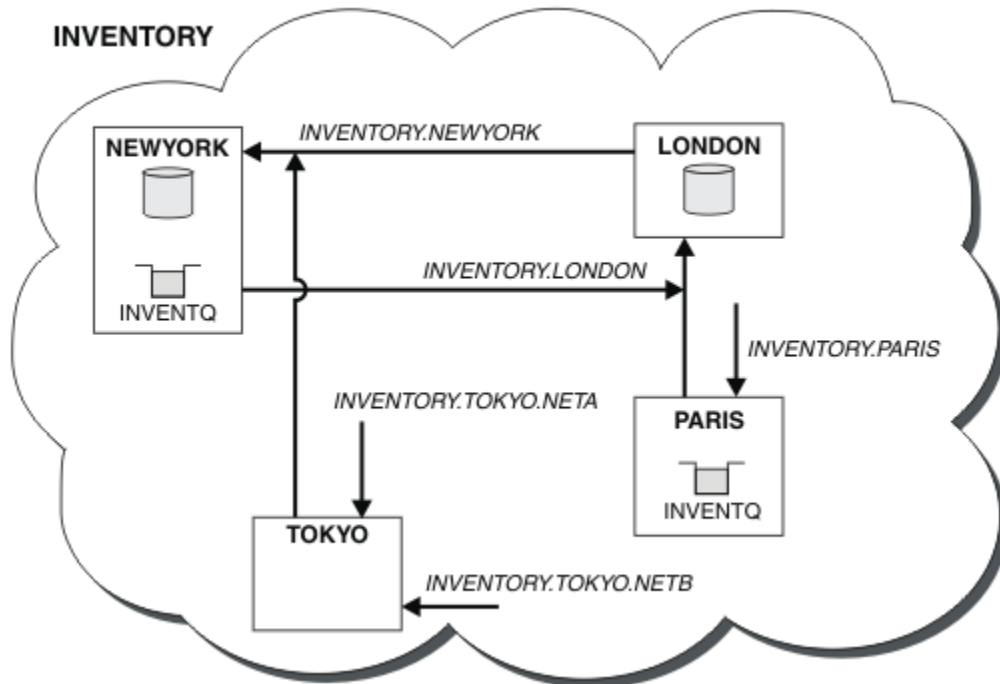


圖 63: INVENTORY 叢集，具有四個佇列管理程式

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。

相關概念

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。

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。

相關工作

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。

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。

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。

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。

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。

第 252 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 323 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：

- ALTER CHANNEL(INVENTORY.TOKYO.NETA) CHLTYPE(CLUSRCVR) NETPRTY(2) DESCR('Main cluster-receiver channel for TOKYO')
- 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。

相關概念

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。

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。

相關工作

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。

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。

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。

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。

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。

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 252 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 IBM 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 327 頁的圖 64 顯示此作業所設定的叢集。

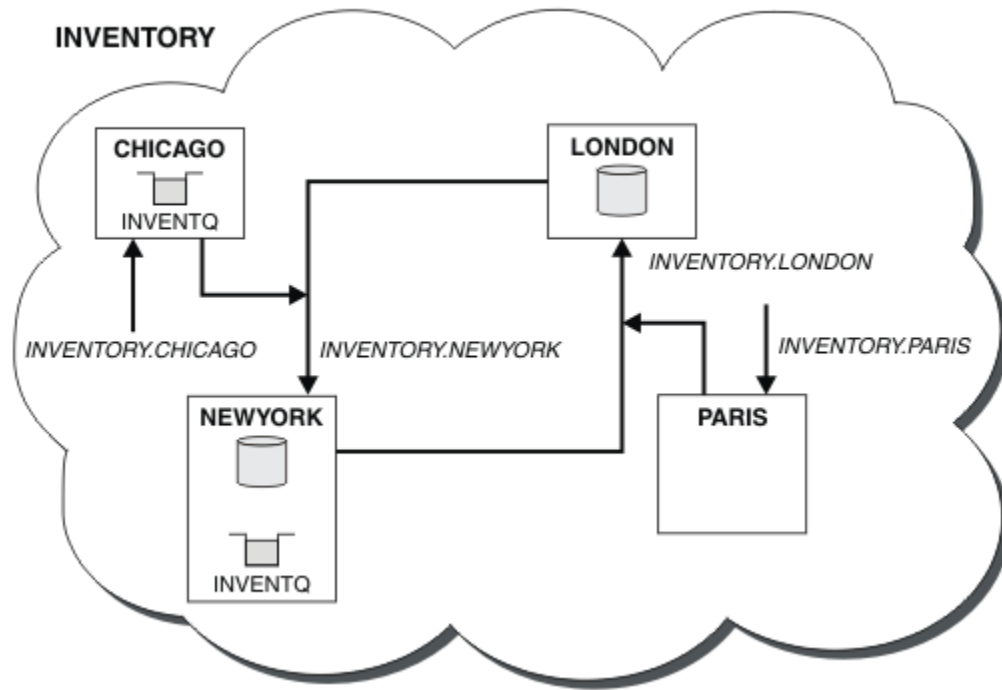


圖 64: INVENTORY 叢集，具有四個佇列管理程式

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，其狀態會變更數次，部分狀態比叢集工作量管理演算法更不理想。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 CLWLRANK 從主要到備份的手動切換。

相關概念

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。

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。

相關工作

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。

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。

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。

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。

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。

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)  
CONNAME(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)  
CONNAME(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 IBM 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 329 頁的圖 65 顯示此作業所設定的叢集。

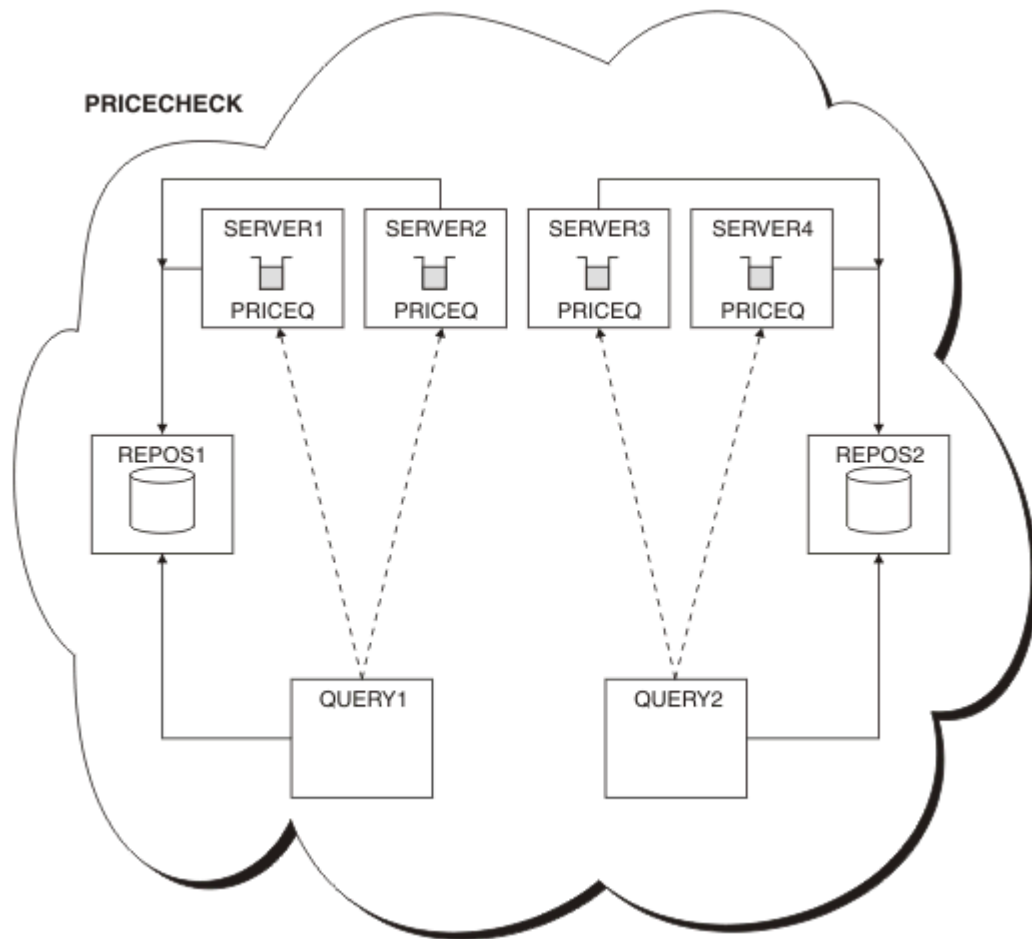


圖 65: PRICECHECK 叢集，具有四個伺服器佇列管理程式、兩個儲存庫及兩個查詢佇列管理程式

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。

相關概念

[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。

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。

相關工作

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。

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。

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。

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。

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。

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 252 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

- 決定哪一個完整儲存庫 LOSANGELES 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。

```
DEFINE CHANNEL(INVENTORY.NEWYORK) CHLTYPE(CLUSSDR) TRPTYPE(TCP)
CONNAME(NEWYORK.CHSTORE.COM) CLUSTER(INVENTORY)
DESCR('Cluster-sender channel from LOSANGELES to repository at NEWYORK')
```

- 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 IBM 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 330 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。

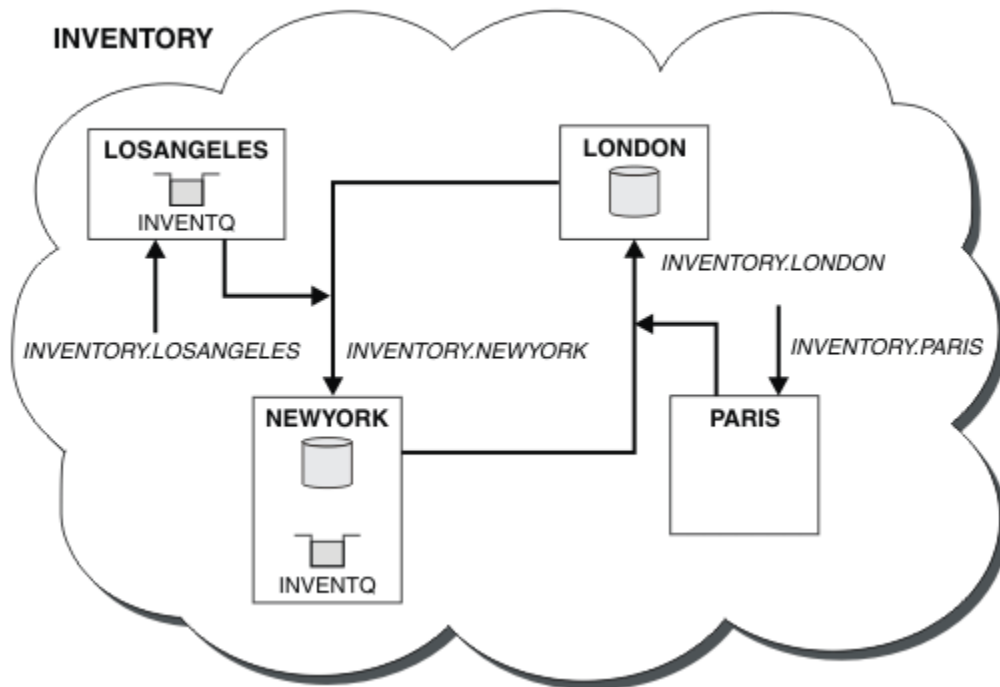


圖 66: 具有四個佇列管理程式的 *INVENTORY* 叢集

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*。

相關概念

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。

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。

相關工作

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。

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。

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。

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。

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。

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)。

相關概念

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。

相關工作

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。

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。

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。

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。

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。

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。

處理訊息親緣性

訊息親緣性很少是良好程式設計設計的一部分。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状态資訊，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。

配置發佈/訂閱傳訊

您可以啟動、停止及顯示佇列發佈/訂閱的狀態。您也可以在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。

程序

- 如需控制佇列發佈/訂閱的相關資訊，請參閱下列子主題：
 - [第 336 頁的『設定排入佇列的發佈/訂閱訊息屬性』](#)
 - [第 337 頁的『正在啟動排入佇列的發佈/訂閱』](#)
 - [第 337 頁的『正在停止排入佇列的發佈/訂閱』](#)

- 第 338 頁的『新增串流』
- 第 339 頁的『刪除串流』
- 第 339 頁的『新增訂閱點』
- 第 346 頁的『在發佈/訂閱網路中結合主題空間』

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 28: 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。

- `PublishBatch 間隔 =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「IBM MQ 探險家」、可程式化指令或 **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

您可以透過設定佇列管理程式的 PSMODE 屬性來啟動已排入佇列的發佈/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

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 IBM MQ Explorer 或 **runmqsc** 指令來啟用該介面。

範例

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

下一步

IBM 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

您可以透過設定佇列管理程式的 PSMODE 屬性來停止已排入佇列的發佈/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
- 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 IBM MQ Explorer 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，以容許應用程式之間的資料隔離，或容許與 IBM WebSphere MQ 6 發佈/訂閱階層進行交互作業。

開始之前

熟悉發佈/訂閱串流的運作方式。請參閱 [串流和主題](#)。

關於這項作業

請使用 PCF 指令、**runmqsc** 或 IBM 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 IBM WebSphere MQ 6 串流同名的本端佇列。
2. 定義與 IBM WebSphere MQ 6 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 IBM WebSphere MQ 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，IBM WebSphere MQ 6 和 IBM WebSphere MQ 7.1 佇列管理程式在相同的發佈/訂閱階層中運作。IBM WebSphere MQ 6 佇列管理程式會共用稱為 'Sport' 的串流。此範例顯示如何在稱為 'Sport' 的 IBM WebSphere MQ 7.1 佇列管理程式上建立佇列及主題，並使用與 IBM WebSphere MQ 6 串流 'Sport' 共用的主題字串 'Sport'。

使用主題字串 'Soccer/Results' 發佈至主題 'Sport' 的 IBM WebSphere MQ 7.1 發佈應用程式會建立產生的主題字串 'Sport/Soccer/Results'。在 IBM WebSphere MQ 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 IBM WebSphere MQ 6 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: IBM MQ queue created.
define topic('Sport') topicstr('Sport')
  2 : define topic('Sport') topicstr('Sport')
AMQ8690: IBM 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: IBM 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第 6 版，請以 IBM WebSphere MQ 6 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第 71 版，您必須在階層中手動配置每一個 IBM WebSphere MQ 7.1 佇列管理程式。

刪除串流

您可以從 IBM WebSphere MQ 7.1 或更新版本的佇列管理程式中刪除串流。

開始之前

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。

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

在管理串流的所有其他已連接 IBM WebSphere MQ 7.1 或更新版本佇列管理程式上，重複步驟 3 至 5。

新增訂閱點

如何使用新的訂閱點來延伸您已從舊版 IBM Integration Bus 移轉的現有已排入佇列的發佈/訂閱應用程式。

開始之前

1. 請檢查是否尚未在 SYSTEM.QPUBSUB.SUBPOINT.NAMELIST 中定義訂閱點。
2. 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

IBM WebSphere MQ 7.1 或更新版本的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。

重要: 已從 IBM MQ 8.0 中移除訂閱點移轉機制。如果您需要移轉現有的應用程式，在移轉至最新版本之前，您必須先執行產品版本說明文件中說明的程序。

如果已排入佇列的發佈/訂閱程式使用已從 IBM WebSphere MQ 6 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 7.1 或更新版本所撰寫的整合發佈/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。

配置分散式發佈/訂閱網路

連接至分散式發佈/訂閱拓撲的佇列管理程式會共用一般聯合主題空間。在一個佇列管理程式上建立的訂閱可以接收連接至拓撲中另一個佇列管理程式的應用程式所發佈的訊息。

您可以控制透過將叢集或階層中的佇列管理程式連接在一起所建立的主題空間範圍。在發佈/訂閱叢集中，主題物件必須針對要跨越叢集之主題空間的每一個分支「叢集化」。在階層中，每一個佇列管理程式都必須配置成在階層中識別其「母項」。

您可以選擇每一個發佈和訂閱是本端或廣域，以進一步控制拓撲內發佈和訂閱的流程。本端發佈和訂閱不會延伸到發佈者或訂閱者所連接的佇列管理程式之外。

相關概念

[分散式發佈/訂閱網路](#)

[發佈範圍](#)

[訂閱範圍](#)

[主題空間](#)

相關工作

[定義叢集主題](#)

配置發佈/訂閱叢集

在佇列管理程式上定義主題。若要使主題成為叢集主題，請設定 **CLUSTER** 內容。若要選擇用於本主題之發佈和訂閱的遞送，請設定 **CLROUTE** 內容。

開始之前

部分叢集配置無法容納直接遞送發佈/訂閱的額外需要。在使用此配置之前，請先探索 [設計發佈/訂閱叢集中詳細說明的考量和選項](#)。

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。

另請參閱 [發佈/訂閱叢集的遞送: 行為注意事項](#)。

測試情境:

- 已依照第 252 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。

關於這項作業

當您在叢集中的佇列管理程式上定義主題時, 您需要指定主題是否為叢集主題, 以及 (如果是的話) 在叢集內遞送此主題的發佈和訂閱。若要使主題成為叢集主題, 您可以使用叢集名稱在 TOPIC 物件上配置 **CLUSTER** 內容。藉由在叢集中的佇列管理程式上定義叢集主題, 可讓整個叢集都可以使用該主題。若要選擇要在叢集內使用的訊息遞送, 請將 TOPIC 物件上的 **CLROUTE** 內容設為下列其中一個值:

- **DIRECT**
- **TOPICHOST**

依預設, 主題遞送是 **DIRECT**。這是 IBM MQ 8.0 之前的唯一選項。當您在佇列管理程式上配置直接遞送的叢集主題時, 叢集裡的所有佇列管理程式便可察覺叢集裡的所有其他佇列管理程式。當執行發佈及訂閱作業時, 每一個佇列管理程式都可以直接連接到叢集中的任何其他佇列管理程式。請參閱 [直接遞送的發佈/訂閱叢集](#)。

從 IBM MQ 8.0 開始, 您可以改為將主題遞送配置成 **TOPICHOST**。當您使用主題主機遞送時, 叢集中的所有佇列管理程式會感知到管理遞送主題定義的叢集佇列管理程式 (亦即, 您定義主題物件所在的佇列管理程式)。執行發佈和訂閱作業時, 叢集中的佇列管理程式只會連接到這些主題主機佇列管理程式, 而不會彼此直接連接。主題主機佇列管理程式負責從已發佈這些發佈資訊的佇列管理程式中, 將發佈資訊遞送到具有相符訂閱的佇列管理程式。請參閱 [主題主機遞送的發佈/訂閱叢集](#)。

註: 在已叢集化主題物件 (透過設定 **CLUSTER** 內容) 之後, 您無法變更 **CLROUTE** 內容的值。必須先將該物件取消叢集 (將 **CLUSTER** 設定為 ' '), 然後您才能變更此值。將主題取消叢集會將主題定義轉換成本端主題, 這將導致在某個期間內, 不會將發佈資訊傳遞到遠端佇列管理程式上的訂閱; 在執行此變更時應將這一點考慮在內。請參閱定義與另一個佇列管理程式中的叢集主題具有相同名稱的非叢集主題的作用。如果您嘗試變更叢集 **CLROUTE** 內容的值, 系統會產生 MQRCCF_CLROUTE_NOT_ALTERABLE 異常狀況。

程序

1. 選擇佇列管理程式來管理主題。

任何叢集佇列管理程式都可以管理主題。從三個佇列管理程式 (LONDON、NEWYORK 或 PARIS) 中選擇一個, 並配置 TOPIC 物件的內容。如果您計劃使用直接遞送, 則不會造成您選擇哪個佇列管理程式的作業差異。如果您計劃使用主題主機遞送, 所選擇的佇列管理程式在遞送發佈資訊方面有額外的責任。因此, 對於主題主機遞送, 請選擇在其中一個功能更強大的系統上管理且具有良好網路連線功能的佇列管理程式。

2. 在佇列管理程式上定義主題。

若要使主題成為叢集主題, 請在定義主題時包括叢集名稱, 並設定您要用於此主題的發佈和訂閱的遞送。例如, 若要在 LONDON 佇列管理程式上建立直接遞送叢集主題, 請如下所示建立主題:

```
DEFINE TOPIC(INVENTORY) TOPICSTR('/INVENTORY') CLUSTER(INVENTORY) CLROUTE(DIRECT)
```

藉由在叢集中的佇列管理程式上定義叢集主題, 可讓整個叢集都可以使用該主題。

如需使用 **CLROUTE** 的相關資訊, 請參閱 [DEFINE TOPIC \(CLROUTE\)](#) 及 [發佈/訂閱叢集的遞送: 行為注意事項](#)。

結果

叢集已準備好接收主題的發佈和訂閱。

下一步

如果您已配置主題主機遞送發佈/訂閱叢集, 則可能想要為此主題新增第二個主題主機。請參閱第 343 頁的『將額外主題主機新增至主題主機遞送叢集』。

如果您有數個個別發佈/訂閱叢集，例如，因為您的組織地理位置分散，您可能想要將部分叢集主題延伸至所有叢集。您可以透過連接階層中的叢集來執行此動作。請參閱第 348 頁的『結合多個叢集的主題空間』。您也可以控制哪些發佈從一個叢集流向另一個叢集。請參閱第 350 頁的『結合及隔離多個叢集中的主題空間』。

相關概念

結合發佈及訂閱範圍

從 IBM WebSphere MQ 7.0 開始，發佈和訂閱範圍會獨立運作，以判斷佇列管理程式之間的發佈流程。

在發佈/訂閱網路中結合主題空間

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

相關工作

將叢集主題定義移至不同的佇列管理程式

對於主題主機遞送或直接遞送叢集，您可能需要在解除佇列管理程式任務時移動叢集主題定義，或者因為叢集佇列管理程式已失敗或長時間無法使用。

將額外主題主機新增至主題主機遞送叢集

在主題主機遞送發佈/訂閱叢集中，可以使用多個佇列管理程式，透過在那些佇列管理程式上定義相同的叢集主題物件，將發佈遞送至訂閱。這可用來改善可用性及工作量平衡。當您為相同的叢集主題物件新增額外的主題主機時，您可以使用 **PUB** 參數來控制何時開始透過新的主題主機遞送發佈資訊。

將佇列管理程式連接至發佈/訂閱階層

您可以將子項佇列管理程式連接至階層中的母項佇列管理程式。如果子項佇列管理程式已經是另一個階層或叢集的成員，則此連線會將階層結合在一起，或將叢集結合至階層。

切斷佇列管理程式與發佈/訂閱階層的連線

在發佈/訂閱階層中，切斷子項佇列管理程式與母項佇列管理程式的連線。

設計發佈/訂閱叢集

分散式發佈/訂閱疑難排解

禁止叢集發佈/訂閱

將叢集主題定義移至不同的佇列管理程式

對於主題主機遞送或直接遞送叢集，您可能需要在解除佇列管理程式任務時移動叢集主題定義，或者因為叢集佇列管理程式已失敗或長時間無法使用。

關於這項作業

您可以在叢集中具有相同叢集主題物件的多個定義。這是主題主機遞送叢集的正常狀態，以及直接遞送叢集的異常狀態。如需相關資訊，請參閱 [相同名稱的多個叢集主題定義](#)。

若要將叢集主題定義移至叢集中的不同佇列管理程式，而不中斷發佈資訊的流程，請完成下列步驟。此程序會將定義從佇列管理程式 QM1 移至佇列管理程式 QM2。

程序

1. 在 QM2 上建立重複的叢集主題定義。

對於直接遞送，請設定所有屬性以符合 QM1 的定義。

對於主題主機遞送，起始將新的主題主機定義為 PUB(DISABLED)。這可讓 QM2 瞭解叢集中的訂閱，但不開始遞送發佈。

2. 等待資訊透過叢集傳播。

等待完整儲存庫佇列管理程式將新的叢集主題定義延伸到叢集中的所有佇列管理程式。使用 **DISPLAY CLUSTER** 指令來顯示每一個叢集成員的叢集主題，並檢查是否有源自 QM2 的定義。

對於主題主機遞送，請等待 QM2 上的新主題主機瞭解所有訂閱。比較 QM2 已知的 Proxy 訂閱與 QM1 已知的 Proxy 訂閱。在佇列管理程式上檢視 Proxy 訂閱的一種方法是發出下列 **runmqsc** 指令：

```
DISPLAY SUB(*) SUBTYPE(PROXY)
```

3. 對於主題主機遞送，請將 QM2 上的主題主機重新定義為 PUB(ENABLED)，然後將 QM1 上的主題主機重新定義為 PUB(DISABLED)。

既然 QM2 上的新主題主機已瞭解其他佇列管理程式上的所有訂閱，主題主機可以開始遞送發佈。

透過使用 PUB(DISABLED) 設定來靜止通過 QM1 的訊息資料流量，當您刪除叢集主題定義時，可確保沒有任何發佈資訊透過 QM1 進行訓練。

4. 從 QM1 中刪除叢集主題定義。

如果佇列管理程式可供使用，則只能從 QM1 中刪除定義。否則，您必須同時執行這兩個定義，直到重新啟動或強制移除 QM1 為止。

如果 QM1 長時間保持無法使用，且在該期間您需要在 QM2 上修改叢集主題定義，則 QM2 定義會比 QM1 定義新，因此通常會優先使用。

在此期間，如果 QM1 與 QM2 上的定義之間存在差異，則會將錯誤寫入兩個佇列管理程式的錯誤日誌中，以警示您叢集主題定義發生衝突。

如果 QM1 永不會回到叢集，例如，由於硬體故障後非預期的解除任務，您可以最後使用 **RESET CLUSTER** 指令強制退出佇列管理程式。**RESET CLUSTER** 會自動刪除目標佇列管理程式上管理的所有主題物件。

相關概念

結合發佈及訂閱範圍

從 IBM WebSphere MQ 7.0 開始，發佈和訂閱範圍會獨立運作，以判斷佇列管理程式之間的發佈流程。

在發佈/訂閱網路中結合主題空間

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

相關工作

配置發佈/訂閱叢集

在佇列管理程式上定義主題。若要使主題成為叢集主題，請設定 **CLUSTER** 內容。若要選擇用於本主題之發佈和訂閱的遞送，請設定 **CLROUTE** 內容。

將額外主題主機新增至主題主機遞送叢集

在主題主機遞送發佈/訂閱叢集中，可以使用多個佇列管理程式，透過在那些佇列管理程式上定義相同的叢集主題物件，將發佈遞送至訂閱。這可用來改善可用性與工作量平衡。當您為相同的叢集主題物件新增額外的主題主機時，您可以使用 **PUB** 參數來控制何時開始透過新的主題主機遞送發佈資訊。

將佇列管理程式連接至發佈/訂閱階層

您可以將子項佇列管理程式連接至階層中的母項佇列管理程式。如果子項佇列管理程式已經是另一個階層或叢集的成員，則此連線會將階層結合在一起，或將叢集結合至階層。

切斷佇列管理程式與發佈/訂閱階層的連線

在發佈/訂閱階層中，切斷子項佇列管理程式與母項佇列管理程式的連線。

將額外主題主機新增至主題主機遞送叢集

在主題主機遞送發佈/訂閱叢集中，可以使用多個佇列管理程式，透過在那些佇列管理程式上定義相同的叢集主題物件，將發佈遞送至訂閱。這可用來改善可用性與工作量平衡。當您為相同的叢集主題物件新增額外的主題主機時，您可以使用 **PUB** 參數來控制何時開始透過新的主題主機遞送發佈資訊。

開始之前

在數個佇列管理程式上定義相同的叢集主題物件，只對主題主機遞送叢集有用。在直接遞送叢集中定義多個相符主題不會變更其行為。此作業僅適用於主題主機遞送叢集。

此作業假設您已閱讀文章 [相同名稱的多個叢集主題定義](#)，特別是下列各節：

- [主題主機遞送叢集中有多個叢集主題定義](#)

- PUB 參數的特殊處理

關於這項作業

當佇列管理程式成為遞送主題主機時，它必須先瞭解叢集中已訂閱的所有相關主題是否存在。如果在新增其他主題主機時將發佈發佈至那些主題，且在該主機得知叢集中其他佇列管理程式上存在訂閱之前將發佈遞送至新主機，則新主機不會將該發佈轉遞至那些訂閱。這會導致訂閱遺失發佈。

發佈資訊不會透過已明確將叢集主題物件 **PUB** 參數設為 **DISABLED** 的主題主機佇列管理程式遞送，因此您可以使用此設定來確保在新增額外主題主機的過程中，沒有訂閱遺失發佈資訊。

註：當佇列管理程式管理已定義為 **PUB (DISABLED)** 的叢集主題時，連接至該佇列管理程式的發佈者無法發佈訊息，且該佇列管理程式上的相符訂閱不會接收在叢集中其他佇列管理程式上發佈的發佈。因此，必須仔細考量在存在訂閱及發佈應用程式連接的佇列管理程式上定義主題主機遞送主題。

程序

1. 配置新的主題主機，並起始將新的主題主機定義為 **PUB (DISABLED)**。

這可讓新主題主機瞭解叢集中的訂閱，但不開始遞送發佈。

如需配置主題主機的相關資訊，請參閱 [第 340 頁的『配置發佈/訂閱叢集』](#)。

2. 判斷新主題主機何時已瞭解所有訂閱。

如果要這麼做，請比較新主題主機已知的 **Proxy** 訂閱與現有主題主機已知的 **Proxy** 訂閱。檢視 **Proxy** 訂閱的一種方法是發出下列 **runmqsc** 指令：**DISPLAY SUB(*) SUBTYPE (PROXY)**

3. 將新的主題主機重新定義為 **PUB (ENABLED)**。

在新的主題主機瞭解其他佇列管理程式上的所有訂閱之後，主題可以開始遞送發佈。

相關概念

結合發佈及訂閱範圍

從 IBM WebSphere MQ 7.0 開始，發佈和訂閱範圍會獨立運作，以判斷佇列管理程式之間的發佈流程。

在發佈/訂閱網路中結合主題空間

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

相關工作

配置發佈/訂閱叢集

在佇列管理程式上定義主題。若要使主題成為叢集主題，請設定 **CLUSTER** 內容。若要選擇用於本主題之發佈和訂閱的遞送，請設定 **CLROUTE** 內容。

將叢集主題定義移至不同的佇列管理程式

對於主題主機遞送或直接遞送叢集，您可能需要在解除佇列管理程式任務時移動叢集主題定義，或者因為叢集佇列管理程式已失敗或長時間無法使用。

將佇列管理程式連接至發佈/訂閱階層

您可以將子項佇列管理程式連接至階層中的母項佇列管理程式。如果子項佇列管理程式已經是另一個階層或叢集的成員，則此連線會將階層結合在一起，或將叢集結合至階層。

切斷佇列管理程式與發佈/訂閱階層的連線

在發佈/訂閱階層中，切斷子項佇列管理程式與母項佇列管理程式的連線。

結合發佈及訂閱範圍

從 IBM WebSphere MQ 7.0 開始，發佈和訂閱範圍會獨立運作，以判斷佇列管理程式之間的發佈流程。

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

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

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

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

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

範例 :football 結果服務

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

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

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

這兩個主題都會發佈至整個叢集。

聯盟已設定下列訂閱，因此任何球隊的球迷都可以使用三種有趣的方式訂閱結果。

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

接收所有結果

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

接收所有首頁結果

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

因為訂閱具有 QMGR 範圍，所以只會比對在家中發佈的結果。

接收我的所有團隊結果

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

因為訂閱具有 QMGR 範圍，所以只會比對本端重新發佈的本端團隊結果。

相關概念

在發佈/訂閱網路中結合主題空間

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

[分散式發佈/訂閱網路](#)

[發佈範圍](#)

[訂閱範圍](#)

相關工作

[配置發佈/訂閱叢集](#)

在佇列管理程式上定義主題。若要使主題成為叢集主題，請設定 **CLUSTER** 內容。若要選擇用於本主題之發佈和訂閱的遞送，請設定 **CLROUTE** 內容。

[將叢集主題定義移至不同的佇列管理程式](#)

對於主題主機遞送或直接遞送叢集，您可能需要在解除佇列管理程式任務時移動叢集主題定義，或者因為叢集佇列管理程式已失敗或長時間無法使用。

[將額外主題主機新增至主題主機遞送叢集](#)

在主題主機遞送發佈/訂閱叢集中，可以使用多個佇列管理程式，透過在那些佇列管理程式上定義相同的叢集主題物件，將發佈遞送至訂閱。這可用來改善可用性及工作量平衡。當您為相同的叢集主題物件新增額外的主題主機時，您可以使用 **PUB** 參數來控制何時開始透過新的主題主機遞送發佈資訊。

[將佇列管理程式連接至發佈/訂閱階層](#)

您可以將子項佇列管理程式連接至階層中的母項佇列管理程式。如果子項佇列管理程式已經是另一個階層或叢集的成員，則此連線會將階層結合在一起，或將叢集結合至階層。

切斷佇列管理程式與發佈/訂閱階層的連線

在發佈/訂閱階層中，切斷子項佇列管理程式與母項佇列管理程式的連線。

在發佈/訂閱網路中結合主題空間

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

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

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

相關概念

結合發佈及訂閱範圍

從 IBM WebSphere MQ 7.0 開始，發佈和訂閱範圍會獨立運作，以判斷佇列管理程式之間的發佈流程。

分散式發佈/訂閱網路

主題空間

相關工作

配置發佈/訂閱叢集

在佇列管理程式上定義主題。若要使主題成為叢集主題，請設定 **CLUSTER** 內容。若要選擇用於本主題之發佈和訂閱的遞送，請設定 **CLROUTE** 內容。

將叢集主題定義移至不同的佇列管理程式

對於主題主機遞送或直接遞送叢集，您可能需要在解除佇列管理程式任務時移動叢集主題定義，或者因為叢集佇列管理程式已失敗或長時間無法使用。

將額外主題主機新增至主題主機遞送叢集

在主題主機遞送發佈/訂閱叢集中，可以使用多個佇列管理程式，透過在那些佇列管理程式上定義相同的叢集主題物件，將發佈遞送至訂閱。這可用來改善可用性及工作量平衡。當您為相同的叢集主題物件新增額外的主題主機時，您可以使用 **PUB** 參數來控制何時開始透過新的主題主機遞送發佈資訊。

將佇列管理程式連接至發佈/訂閱階層

您可以將子項佇列管理程式連接至階層中的母項佇列管理程式。如果子項佇列管理程式已經是另一個階層或叢集的成員，則此連線會將階層結合在一起，或將叢集結合至階層。

切斷佇列管理程式與發佈/訂閱階層的連線

在發佈/訂閱階層中，切斷子項佇列管理程式與母項佇列管理程式的連線。

定義叢集主題

在發佈/訂閱叢集中建立單一主題空間

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

開始之前

您已在單一第 7 版佇列管理程式上實作發佈/訂閱系統。

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

關於這項作業

您現在想要擴增系統以支援更多發佈者和訂閱者，並在整個叢集中顯示每個主題。

程序

1. 建立叢集以與發佈/訂閱系統搭配使用。

如果您有現有的傳統叢集，基於效能考量，最好為新的發佈訂閱系統設定新的叢集。您可以將相同的伺服器用於兩個叢集的叢集儲存庫。

2. 選擇一個佇列管理程式 (可能是其中一個儲存庫) 作為叢集主題主機。
3. 請確定在整個發佈/訂閱叢集中可見的每一個主題都解析為管理主題物件。
設定命名發佈/訂閱叢集的 **CLUSTER** 屬性。

下一步

將發佈者及訂閱者應用程式連接至叢集中的任何佇列管理程式。

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

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

相關概念

[分散式發佈/訂閱網路](#)

[主題空間](#)

相關工作

將 [IBM WebSphere MQ 7 或更新版本佇列管理程式新增至現有的 IBM WebSphere MQ 6 主題空間](#) 延伸現有的 IBM WebSphere MQ 6 發佈/訂閱系統，以與 IBM WebSphere MQ 7 或更新版本的佇列管理程式交互作業，並共用相同的主題空間。

[結合多個叢集的主題空間](#)

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

[結合及隔離多個叢集中的主題空間](#)

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

[發佈及訂閱多個叢集中的主題空間](#)

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

[定義叢集主題](#)

將 *IBM WebSphere MQ 7* 或更新版本佇列管理程式新增至現有的 *IBM WebSphere MQ 6* 主題空間

延伸現有的 IBM WebSphere MQ 6 發佈/訂閱系統，以與 IBM WebSphere MQ 7 或更新版本的佇列管理程式交互作業，並共用相同的主題空間。

開始之前

您具有現有的 IBM WebSphere MQ 6 發佈/訂閱系統。

您已在新的伺服器上安裝 IBM WebSphere MQ 7 或更新版本，並配置佇列管理程式。

關於這項作業

您想要延伸現有的 IBM WebSphere MQ 6 發佈/訂閱系統，以使用 IBM WebSphere MQ 7 或更新版本的佇列管理程式。

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

您打算在未來將 IBM WebSphere MQ 6 佇列管理程式升級至 IBM WebSphere MQ 7 或更新版本。目前，您正在 IBM WebSphere MQ 7 或更新版本佇列管理程式上繼續執行現有已排入佇列的發佈/訂閱應用程式。

程序

1. 建立一組傳送端-接收端通道，以雙向連接第 7 版或更新版本佇列管理程式與其中一個 IBM WebSphere MQ 6 佇列管理程式。

2. 建立兩個具有目標佇列管理程式名稱的傳輸佇列。如果您因某些原因而無法使用目標佇列管理程式的名稱作為傳輸佇列名稱，請使用佇列管理程式別名。
3. 配置傳輸佇列以觸發傳送端通道。
4. 如果 IBM WebSphere MQ 6 發佈/訂閱系統使用串流，請依照 [第 338 頁的『新增串流』](#) 中的說明，將串流新增至第 7 版或更新版本的佇列管理程式。
5. 檢查第 7 版或更新版本佇列管理程式 **PSMODE** 是否設為 ENABLE。
6. 變更其 **PARENT** 屬性，以參照其中一個 IBM WebSphere MQ 6 佇列管理程式。
7. 請檢查佇列管理程式之間的上下代關係狀態是雙向作用中。

下一步

完成作業之後，IBM WebSphere MQ 6 及 IBM WebSphere MQ 7 或更新版本的佇列管理程式會共用相同的主題空間。例如，您可以執行下列所有作業。

- 在 IBM WebSphere MQ 6 與 IBM WebSphere MQ 7 或更新版本的佇列管理程式之間交換發佈及訂閱。
- 在 IBM WebSphere MQ 7 或更新版本佇列管理程式上執行現有的 IBM WebSphere MQ 6 發佈/訂閱程式。
- 檢視及修改 IBM WebSphere MQ 6 或 IBM WebSphere MQ 7 或更新版本佇列管理程式上的主題空間。
- 撰寫 IBM WebSphere MQ 7 或更新版本的發佈/訂閱應用程式，並在 IBM WebSphere MQ 7 或更新版本的佇列管理程式上執行它們。
- 使用 IBM WebSphere MQ 7 或更新版本應用程式建立新的發佈和訂閱，並與 IBM WebSphere MQ 6 應用程式交換它們。

相關概念

[分散式發佈/訂閱網路](#)

[主題空間](#)

相關工作

[在發佈/訂閱叢集中建立單一主題空間](#)

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

[結合多個叢集的主題空間](#)

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

[結合及隔離多個叢集中的主題空間](#)

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

[發佈及訂閱多個叢集中的主題空間](#)

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

[定義叢集主題](#)

結合多個叢集的主題空間

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

開始之前

這項作業假設您有現有的直接遞送發佈/訂閱叢集，且您想要將一些叢集主題延伸到所有叢集中。

註：您無法對主題主機遞送的發佈/訂閱叢集執行此動作。

關於這項作業

若要將發佈從一個叢集延伸到另一個叢集，您需要在階層中將叢集結合在一起；請參閱 [第 349 頁的圖 67](#)。階層式連線會在連接的佇列管理程式之間傳送訂閱和發佈，而叢集會在每一個叢集內傳送叢集主題，但不會在叢集之間。

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

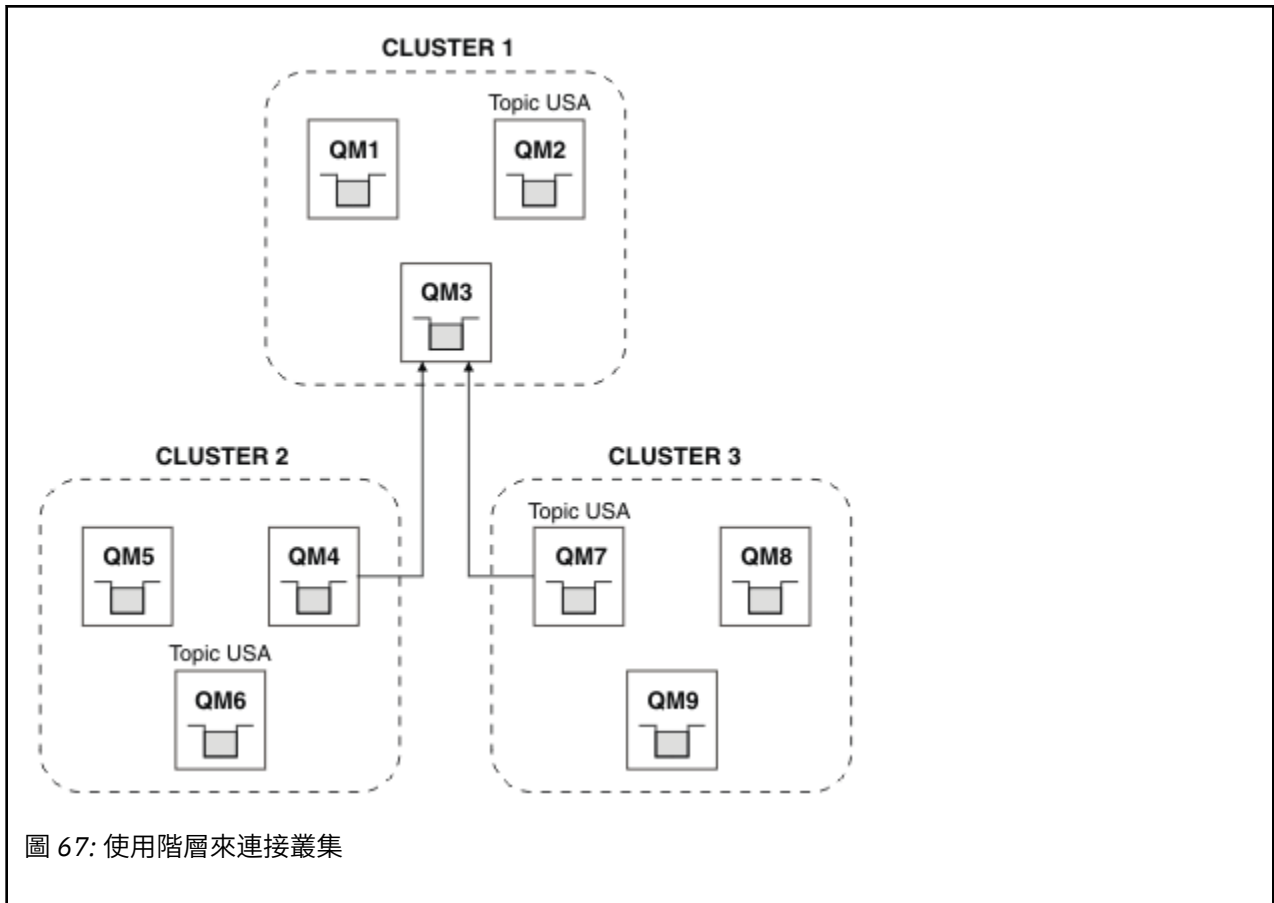


圖 67: 使用階層來連接叢集

下列步驟會將叢集連接至階層。

程序

1. 建立兩組傳送端-接收端通道，以雙向連接 QM3 和 QM4，以及 QM3 和 QM7。您必須使用傳統傳送端-接收端通道及傳輸佇列 (而非叢集) 來連接階層。
2. 使用目標佇列管理程式的名稱建立三個傳輸佇列。如果您因某些原因而無法使用目標佇列管理程式的名稱作為傳輸佇列名稱，請使用佇列管理程式別名。
3. 配置傳輸佇列以觸發傳送端通道。
4. 檢查 QM3、QM4 及 QM7 的 **PSMODE** 是否設為 **ENABLE**。
5. 將 QM4 和 QM7 的 **PARENT** 屬性變更為 QM3。
6. 請檢查佇列管理程式之間的上下代關係狀態是雙向作用中。
7. 使用屬性 **CLUSTER** ('CLUSTER 1')、**CLUSTER** ('CLUSTER 2') 及 **CLUSTER** ('CLUSTER 3') 來建立管理主題 USA 叢集 1、2 和 3 中的三個叢集主題主機佇列管理程式中的每一個。叢集主題主機不需要是階層式連接的佇列管理程式。

下一步

您現在可以發佈或訂閱 [第 349 頁的圖 67](#) 中的叢集主題 USA。發佈訂閱會流向所有三個叢集中的發佈者及訂閱者。

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

另請參閱 [第 350 頁的『結合及隔離多個叢集中的主題空間』](#)。

相關概念

分散式發佈/訂閱網路

主題空間

相關工作

在發佈/訂閱叢集中建立單一主題空間

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

將 IBM WebSphere MQ 7 或更新版本佇列管理程式新增至現有的 IBM WebSphere MQ 6 主題空間

延伸現有的 IBM WebSphere MQ 6 發佈/訂閱系統，以與 IBM WebSphere MQ 7 或更新版本的佇列管理程式交互作業，並共用相同的主題空間。

結合及隔離多個叢集中的主題空間

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

發佈及訂閱多個叢集中的主題空間

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

定義叢集主題

結合及隔離多個叢集中的主題空間

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

開始之前

請檢查主題 [第 348 頁的『結合多個叢集的主題空間』](#)。它可能足以滿足您的需求，而不需要新增其他佇列管理程式作為橋接器。

註: 您只能使用直接遞送的發佈/訂閱叢集來完成此作業。您無法使用主題主機遞送叢集來執行此動作。

關於這項作業

[第 348 頁的『結合多個叢集的主題空間』](#) 中 [第 349 頁的圖 67](#) 所示拓撲的潛在改進是隔離並非在所有叢集之間共用的叢集主題。透過建立不在任何叢集中的橋接佇列管理程式來隔離叢集; 請參閱 [第 351 頁的圖 68](#)。使用橋接佇列管理程式來過濾哪些發佈和訂閱可以從一個叢集流向另一個叢集。

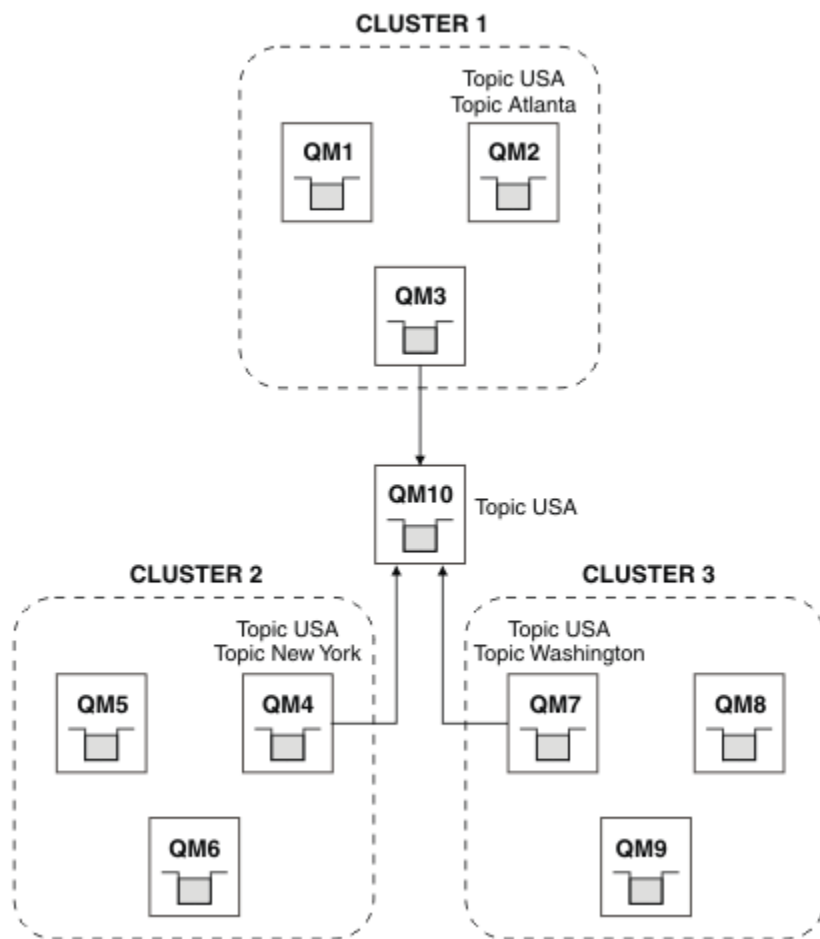


圖 68: 橋接叢集

使用橋接器來隔離您不想在其他叢集上透過橋接器公開的叢集主題。在 [第 351 頁的圖 68](#) 中，USA 是在所有叢集中共用的叢集主題，而 Atlanta、New York 及 Washington 是在每個叢集中僅共用的叢集主題。

使用下列程序來建立配置模型：

程序

1. 修改所有 `SYSTEM.BASE.TOPIC` 主題物件以具有 **SUBSCOPE (QMGR)** 及 **PUBSCOPE (QMGR)** 所有佇列管理程式上。
除非您明確設定 **SUBSCOPE (ALL)**，否則不會將任何主題 (甚至叢集主題) 延伸到其他佇列管理程式及 **PUBSCOPE (ALL)** 叢集主題的根主題。
2. 使用屬性 **CLUSTER (clustername) SUBSCOPE (ALL)**，在每一個叢集中定義要共用的三個叢集主題主機佇列管理程式上的主題。及 **PUBSCOPE (ALL)**。
如果您想要在所有叢集之間共用部分叢集主題，請在每一個叢集中定義相同的主題。使用每一個叢集的叢集名稱作為叢集屬性。
3. 對於您想要在所有叢集之間共用的叢集主題，請使用屬性 **SUBSCOPE (ALL)** 及 **PUBSCOPE (ALL)**，在橋接器佇列管理程式 (QM10) 上再次定義主題。

範例

在 [第 351 頁的圖 68](#) 的範例中，只有繼承自 USA 的主題會在所有三個叢集之間傳播。

下一步

使用 **SUBSCOPE (ALL)** 在橋接器佇列管理程式上定義主題的訂閱及 **PUBSCOPE (ALL)** 在叢集之間延伸。

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

任何其他訂閱都是佇列管理程式的本端訂閱。

相關概念

[分散式發佈/訂閱網路](#)

[主題空間](#)

[發佈範圍](#)

[訂閱範圍](#)

相關工作

[在發佈/訂閱叢集中建立單一主題空間](#)

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

[將 IBM WebSphere MQ 7 或更新版本佇列管理程式新增至現有的 IBM WebSphere MQ 6 主題空間](#)

延伸現有的 IBM WebSphere MQ 6 發佈/訂閱系統，以與 IBM WebSphere MQ 7 或更新版本的佇列管理程式交互作業，並共用相同的主題空間。

[結合多個叢集的主題空間](#)

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

[發佈及訂閱多個叢集中的主題空間](#)

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

[定義叢集主題](#)

發佈及訂閱多個叢集中的主題空間

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

開始之前

在叢集之間的交集內建立具有部分佇列管理程式的多個傳統叢集。

關於這項作業

基於各種不同的原因，您可能已選擇重疊叢集。

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

透過確保在非重疊主題空間中定義主題，您可以將主題部署至重疊發佈/訂閱叢集，請參閱 [第 353 頁的圖 69](#)。如果主題空間重疊，則部署至重疊叢集會導致問題。

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

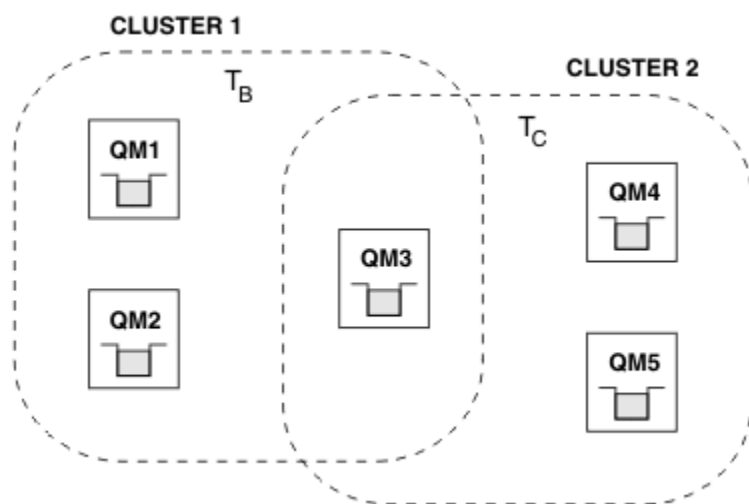


圖 69: 重疊叢集，非重疊主題空間

程序

建立方法來確保主題空間不會重疊。

例如，定義每一個主題空間的唯一根主題。建立根主題叢集主題。

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

範例

在 [第 353 頁的圖 69](#) 發佈者及連接至 QM3 的訂閱者中，可以發佈或訂閱 T_B 或 T_C

下一步

將使用這兩個叢集中的主題的發佈者和訂閱者連接至重疊中的佇列管理程式。

將必須僅使用特定叢集中主題的發佈者和訂閱者連接至不在重疊中的佇列管理程式。

相關概念

[分散式發佈/訂閱網路](#)

[主題空間](#)

相關工作

[在發佈/訂閱叢集中建立單一主題空間](#)

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

[將 IBM WebSphere MQ 7 或更新版本佇列管理程式新增至現有的 IBM WebSphere MQ 6 主題空間](#)

延伸現有的 IBM WebSphere MQ 6 發佈/訂閱系統，以與 IBM WebSphere MQ 7 或更新版本的佇列管理程式交互作業，並共用相同的主題空間。

[結合多個叢集的主題空間](#)

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

[結合及隔離多個叢集中的主題空間](#)

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

[定義叢集主題](#)

將佇列管理程式連接至發佈/訂閱階層

您可以將子項佇列管理程式連接至階層中的母項佇列管理程式。如果子項佇列管理程式已經是另一個階層或叢集的成員，則此連線會將階層結合在一起，或將叢集結合至階層。

開始之前

1. 發佈/訂閱階層中的佇列管理程式必須具有唯一的佇列管理程式名稱。
2. 發佈/訂閱階層依賴 "已排入佇列的發佈/訂閱" 佇列管理程式特性。這必須同時在母項和子項佇列管理程式上啟用。請參閱第 337 頁的『正在啟動排入佇列的發佈/訂閱』。
3. 發佈/訂閱關係依賴佇列管理程式傳送端和接收端通道。建立通道有兩種方式：
 - 同時將母項和子項佇列管理程式新增至 IBM MQ 叢集。請參閱第 252 頁的『將佇列管理程式加入叢集中』。
 - 建立從子佇列管理程式到母項的傳送端/接收端通道配對，以及從母項到子項的傳送端/接收端通道配對。每一個通道都需要使用與目標佇列管理程式同名的傳輸佇列，或使用與目標佇列管理程式同名的佇列管理程式別名。如需如何建立點對點通道連線的相關資訊，請參閱第 148 頁的『IBM MQ 分散式佇列技術』。

如需在每一種通道配置類型上配置階層的範例，請參閱下列一組發佈/訂閱階層實務範例：

- [實務範例 1: 使用具有佇列管理程式別名的點對點通道](#)
- [實務範例 2: 對傳輸佇列及遠端佇列管理程式使用同名點對點通道](#)
- [實務範例 3: 使用叢集通道來新增佇列管理程式](#)

關於這項作業

使用 ALTER QMGR PARENT (*PARENT_NAME*) **runmqsc** 指令，將子項連接至母項。這項配置是在子項佇列管理程式上執行，其中 *PARENT_NAME* 是母項佇列管理程式的名稱。

程序

```
ALTER QMGR PARENT (PARENT_NAME)
```

範例

第一個範例顯示如何連接佇列管理程式 QM2 作為 QM1 的子項，然後查詢 QM2 以確認它已順利成為 **STATUS** 為 ACTIVE 的子項：

```
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: IBM 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)
```

如果 **STATUS** 未顯示為 **ACTIVE**，請檢查子項與母項之間的通道是否已正確配置且在執行中。請檢查這兩個佇列管理程式錯誤日誌，以取得可能的錯誤。

下一步

依預設，發佈者和某個佇列管理程式上的訂閱者所使用的主題，會與階層中其他佇列管理程式上的發佈者和訂閱者共用。可以配置受管理主題，以透過使用 **SUBSCOPE** 和 **PUBSCOPE** 主題內容來控制共用層次。請參閱第 340 頁的『配置分散式發佈/訂閱網路』。

相關概念

[結合發佈及訂閱範圍](#)

從 IBM WebSphere MQ 7.0 開始，發佈和訂閱範圍會獨立運作，以判斷佇列管理程式之間的發佈流程。

[在發佈/訂閱網路中結合主題空間](#)

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

[串流和主題](#)

[發佈/訂閱傳訊](#)

相關工作

[配置發佈/訂閱叢集](#)

在佇列管理程式上定義主題。若要使主題成為叢集主題，請設定 **CLUSTER** 內容。若要選擇用於本主題之發佈和訂閱的遞送，請設定 **CLROUTE** 內容。

[將叢集主題定義移至不同的佇列管理程式](#)

對於主題主機遞送或直接遞送叢集，您可能需要在解除佇列管理程式任務時移動叢集主題定義，或者因為叢集佇列管理程式已失敗或長時間無法使用。

[將額外主題主機新增至主題主機遞送叢集](#)

在主題主機遞送發佈/訂閱叢集中，可以使用多個佇列管理程式，透過在那些佇列管理程式上定義相同的叢集主題物件，將發佈遞送至訂閱。這可用來改善可用性及工作量平衡。當您為相同的叢集主題物件新增額外的主題主機時，您可以使用 **PUB** 參數來控制何時開始透過新的主題主機遞送發佈資訊。

[切斷佇列管理程式與發佈/訂閱階層的連線](#)

在發佈/訂閱階層中，切斷子項佇列管理程式與母項佇列管理程式的連線。

相關參考

[DISPLAY PUBSUB](#)

切斷佇列管理程式與發佈/訂閱階層的連線

在發佈/訂閱階層中，切斷子項佇列管理程式與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: IBM MQ queue manager changed.  
  2 : display pubsub type(child)  
AMQ8147: IBM MQ object not found.  
display pubsub type(parent)  
  3 : display pubsub type(parent)  
AMQ8147: IBM 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。

相關概念

結合發佈及訂閱範圍

從 IBM WebSphere MQ 7.0 開始，發佈和訂閱範圍會獨立運作，以判斷佇列管理程式之間的發佈流程。

在發佈/訂閱網路中結合主題空間

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

相關工作

配置發佈/訂閱叢集

在佇列管理程式上定義主題。若要使主題成為叢集主題，請設定 **CLUSTER** 內容。若要選擇用於本主題之發佈和訂閱的遞送，請設定 **CLROUTE** 內容。

將叢集主題定義移至不同的佇列管理程式

對於主題主機遞送或直接遞送叢集，您可能需要在解除佇列管理程式任務時移動叢集主題定義，或者因為叢集佇列管理程式已失敗或長時間無法使用。

將額外主題主機新增至主題主機遞送叢集

在主題主機遞送發佈/訂閱叢集中，可以使用多個佇列管理程式，透過在那些佇列管理程式上定義相同的叢集主題物件，將發佈遞送至訂閱。這可用來改善可用性及工作量平衡。當您為相同的叢集主題物件新增額外的主題主機時，您可以使用 **PUB** 參數來控制何時開始透過新的主題主機遞送發佈資訊。

將佇列管理程式連接至發佈/訂閱階層

您可以將子項佇列管理程式連接至階層中的母項佇列管理程式。如果子項佇列管理程式已經是另一個階層或叢集的成員，則此連線會將階層結合在一起，或將叢集結合至階層。

ULW 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。

關於這項作業

此資訊適用於 UNIX, Linux, and Windows。

程序

- 使用下列鏈結中的資訊來配置安裝:
 - [第 363 頁的『變更主要安裝』](#)
 - [第 364 頁的『將佇列管理程式與安裝相關聯』](#)
 - [第 356 頁的『在多重安裝環境中連接應用程式』](#)

ULW 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 7.1 或更新版本的程式庫，IBM 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 MQ 會使用與應用程式所連接之佇列管理程式相關聯的安裝架構中的程式庫。

下列概念用來說明應用程式連接至 IBM MQ 的方式：

鏈結

編譯應用程式時，應用程式會鏈結至 IBM 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 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 358 頁的『載入 IBM MQ 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 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 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 7.1 或更新版本的程式庫，則 IBM 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 MQ 會從與佇列管理程式相關聯的安裝中載入程式庫。

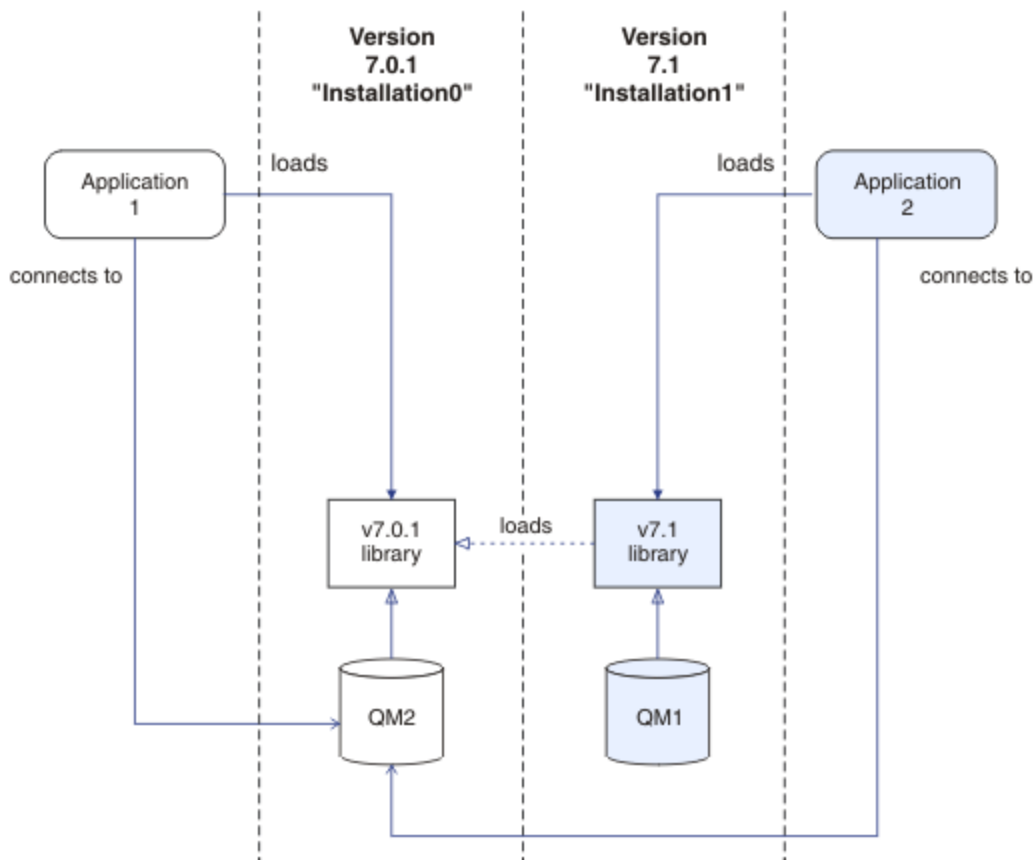


圖 70: 在多重安裝環境中連接應用程式

例如，[第 357 頁的圖 70](#) 顯示具有 IBM WebSphere MQ 7.0.1 安裝 (Installation0) 及 IBM WebSphere MQ 7.1 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 IBM WebSphere MQ 7.0.1 程式庫。當 application 1 連接至 QM2 時，會使用 IBM WebSphere MQ 7.0.1 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 IBM WebSphere MQ 7.0.1 程式庫無法載入其他程式庫版本。也就是說，如果直接載入 IBM WebSphere MQ 7.0.1 程式庫，則無法使用與 IBM 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 IBM WebSphere MQ 7.1 程式庫。當 application 2 連接至 QM2 時，IBM WebSphere MQ 7.1 程式庫會載入並使用 IBM WebSphere MQ 7.0.1 程式庫。如果 application 2 連接至 QM1，或 QM2 與 Installation1 相關聯，則會載入 IBM WebSphere MQ 7.1 程式庫，且應用程式會如預期般運作。

在 [UNIX, Linux, and Windows 上的多重安裝佇列管理程式共存性](#) 中，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 7.1 程式庫的相關資訊，請參閱 [第 358 頁的『載入 IBM MQ 程式庫』](#)。

支援及限制

如果找到並載入下列任何 IBM WebSphere MQ 7.1 或更新版本的程式庫，IBM 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
- .NET 處於未受管理模式

IBM 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 361 頁的『使用多個安裝架構的應用程式限制』](#)。

相關概念

[第 361 頁的『使用多個安裝架構的應用程式限制』](#)

在多個安裝環境中使用 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 358 頁的『載入 IBM MQ 程式庫』](#)

在決定如何載入 IBM 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 MQ 的安裝位置，以及 IBM MQ 的位置是否可能變更。

相關工作

選擇主要安裝

[第 363 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。

[第 364 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。

載入 IBM MQ 程式庫

在決定如何載入 IBM 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 MQ 的安裝位置，以及 IBM MQ 的位置是否可能變更。

此資訊適用於 IBM WebSphere MQ 7.1 或更新版本的程式庫。


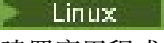

IBM MQ 程式庫的定位和載入方式視您的安裝環境而定：

- 在 UNIX and Linux 系統上，如果 IBM WebSphere MQ 7.1 或更新版本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 IBM WebSphere MQ 7.1 或更新版本的安裝作為主要安裝，或手動建立符號鏈結。
- 如果 IBM WebSphere MQ 7.1 或更新版本安裝在非預設位置中 (如果也安裝了 IBM WebSphere MQ 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 MQ 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 360 頁的『作業系統程式庫載入機制』](#)。

最理想的情況是，您應該確定作業系統所載入的 IBM MQ 程式庫是與佇列管理程式相關聯的程式庫。

載入 IBM 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	設定或變更應用程式的內嵌式執行時期搜尋路徑 (RPath)。 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 建置程序化應用程式 。	<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 MQ 的位置變更，您必須變更 RPath。
UNIX and Linux 系統	使用 setmqenv 或 crtmqenv 搭配 -k 或 -l 選項來設定 <code>LD_LIBRARY_PATH</code> 環境變數。( 在 AIX 上，此環境變數為 <code>LIBPATH</code>)	<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 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。  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	使用 setmqenv 或 crtmqenv 設定 <code>PATH</code> 變數。	<ul style="list-style-type: none"> 現有應用程式不需要變更。 如果 IBM 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。
 UNIX, Linux, and Windows 系統	將主要安裝設為 IBM WebSphere MQ 7.1 或更新版本安裝。請參閱第 363 頁的『 變更主要安裝 』。 如需主要安裝的相關資訊，請參閱 選擇主要安裝 。	<ul style="list-style-type: none"> 現有應用程式不需要變更。 如果 IBM 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 MQ。 	<ul style="list-style-type: none"> 安裝 IBM WebSphere MQ 7.0.1 時，您無法將主要安裝設為 IBM WebSphere MQ 7.1 或更新版本。   UNIX and Linux: 如果 <code>/usr/lib</code> 不在預設搜尋路徑中，則無法運作。

Linux 的程式庫載入考量

 Linux

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

Solaris

舊版 IBM 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 MQ 現在會自動載入這些程式庫的適當版本。如果 IBM 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檔案庫：

- 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。

Linux

UNIX

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 (也包括 AIX 上的 `LIBPATH`)。如果設定此變數，則它會定義一組目錄，以搜尋所需的 IBM 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 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。如果現有應用程式使用 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 7.1，或使用其他方法來尋找程式庫。
- 使用預設程式庫路徑。如果在搜尋 `LD_LIBRARY_PATH` 變數及 RPath 位置之後找不到 IBM 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** AIX: `dump`
- **Linux** Linux: `readelf`
- **Solaris** Solaris: `elfdump`

相關概念

[第 361 頁的『使用多個安裝架構的應用程式限制』](#)

在多個安裝環境中使用 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 356 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 7.1 或更新版本的程式庫，IBM 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 MQ 會使用與應用程式所連接之佇列管理程式相關聯的安裝架構中的程式庫。

相關工作

[選擇主要安裝](#)

[第 363 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。

[第 364 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。

在多個安裝環境中使用 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 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

在多重安裝環境中，必須更新現有的結束程式，以便與 IBM WebSphere MQ 7.1 或更新版本的安裝搭配使用。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 結構撰寫，無法使用內嵌的 RPATH 來尋找 IBM MQ 程式庫，也無法鏈結至 IBM MQ 程式庫。如需相關資訊，請參閱 [在 UNIX, Linux, and Windows 上撰寫結束程式及可安裝服務](#)。

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 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從應用程式載入 IBM MQ 執行時期程式庫的安裝相關聯。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 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. 將非捷徑連接至與 IBM WebSphere MQ 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連接快速路徑沒有影響。
4. 您無法結合連接至與 IBM WebSphere MQ 7.0.1 安裝相關聯的佇列管理程式，以及連接至與 IBM WebSphere MQ 7.1 或更新版本安裝相關聯的佇列管理程式的捷徑。

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：

- 安裝必須與載入 IBM MQ 執行時期程式庫的安裝相同。
- 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_INSTALLATION_MISMATCH](#)

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

ULW 在多個安裝環境中連接 .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)。所有其他安裝架構的 .NET 組件可在每一個安裝架構的安裝路徑中找到，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 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 362 頁的『使用應用程式配置檔來連接 .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 363 頁的『使用 DEVPATH 連接 .NET 應用程式』](#)。

使用應用程式配置檔來連接 .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：

	使用舊版 IBM MQ 編譯的應用程式	使用較新 IBM MQ 版本編譯的應用程式
使用更新版本 IBM MQ 主要安裝來執行應用程式。(GAC 中的較新版本組件):	不需要變更	不需要變更
使用舊版 IBM MQ 主要安裝來執行應用程式。(GAC 中的舊版組件):	不需要變更	在應用程式配置檔中: <ul style="list-style-type: none"> • 使用 <i>bindingRedirect</i> 標籤，指出使用 GAC 中的舊版組件
使用更新版本的 IBM MQ 非主要安裝來執行應用程式。(安裝資料夾中的更新版本組件):	在應用程式配置檔中: <ul style="list-style-type: none"> • 使用 <i>codebase</i> 標籤來指向更新版本組件的位置 • 使用 <i>bindingRedirect</i> 標籤來指出使用更新版本組件 	在應用程式配置檔中: <ul style="list-style-type: none"> • 使用 <i>codebase</i> 標籤來指向更新版本組件的位置
使用舊版 IBM MQ 非主要安裝來執行應用程式。(安裝資料夾中的舊版組件):	在應用程式配置檔中: <ul style="list-style-type: none"> • 使用 <i>codebase</i> 標籤來指向舊版組件的位置 • 包括標籤 <i>publisherpolicy Apply=no</i> 	在應用程式配置檔中: <ul style="list-style-type: none"> • 使用 <i>codebase</i> 標籤來指向舊版組件的位置 • 使用 <i>bindingRedirect</i> 標籤指出使用舊版組件 • 包括標籤 <i>publisherpolicy Apply=no</i>

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 MQ 安裝路徑來修改此檔案。該

檔案也可以使用 *linkedConfiguration* 標籤直接併入其他配置檔中。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，請參閱適當的 Microsoft *DEVPATH* 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 356 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 7.1 或更新版本的程式庫，IBM 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 MQ 會使用與應用程式所連接之佇列管理程式相關聯的安裝架構中的程式庫。

多個安裝

相關工作

選擇主要安裝

使用 .NET

ULW 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 7.1 或更新版本的安裝與 IBM WebSphere MQ 7.0.1 的安裝共存，則 IBM WebSphere MQ 7.0.1 安裝必須是主要安裝。當已安裝 IBM WebSphere MQ 7.1 或更新版本，且 IBM WebSphere MQ 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。

Windows 在 Windows 上安裝程序期間，您可以指定安裝是主要安裝。

Linux **UNIX** 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。

程序

- 若要將安裝設為主要安裝，請完成下列步驟：
 - 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 7.1 或更新版本安裝的安裝路徑。

- 如果現有 IBM WebSphere MQ 7.1 或更新版本安裝設為主要安裝，請 [取消設定它](#)，然後再繼續下一步。

如果系統上已安裝 IBM WebSphere MQ 7.0.1，則無法變更主要安裝。

c) 請確定您以適當的權限登入：

- **UNIX** 在 UNIX and Linux 上作為 root。
- **Linux** 作為 Windows 系統上「管理者」群組的成員。

d) 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
```

e) **Windows**

在 Windows 系統上，重新啟動系統。

- 若要取消將安裝設為主要安裝，請完成下列步驟：

a) 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 7.1 或更新版本安裝的安裝路徑。

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

b) 請確定您以適當的權限登入：

- **UNIX** 在 UNIX and Linux 上作為 root。
- **Linux** 作為 Windows 系統上「管理者」群組的成員。

c) 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特性](#)

[外部媒體庫及控制指令鏈結至 UNIX and Linux 上的主要安裝](#)

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](#)

ULW 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。

關於這項作業

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 IBM WebSphere MQ 7.1 或更新版本安裝上啟動 IBM WebSphere MQ 7.0.1 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，並將其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，而不只是那些與 **dspmq** 指令相同安裝相關聯的佇列管理程式。**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。

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。

當您想要將佇列管理程式與安裝相關聯時，可以採用下列方式來使用 **setmqm** 指令：

- 在同等版本的 IBM 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 IBM MQ 移轉至新版 IBM 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 [維護及移轉](#)。

程序

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。

下一步

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 `mq5.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 366 頁的『尋找系統上 IBM MQ 的安裝』](#)

如果系統上有多個 IBM MQ 安裝，您可以檢查哪些版本已安裝及其位置。

[第 74 頁的『IBM MQ 配置檔 `mq5.ini`』](#)

IBM MQ 配置檔 `mq5.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

[addmqinf](#)

[dspmq](#)

[dspmqinst](#)
[endmqm](#)
[setmqm](#)
[strmqm](#)

ULW 尋找系統上 IBM MQ 的安裝

如果系統上有多個 IBM 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 MQ 安裝架構：

- 使用 **dspmqver** 指令。如果是從 IBM WebSphere MQ 7.0.1 安裝架構發出，這個指令不會提供系統上所有安裝的詳細資料。
- 使用平台安裝工具來查詢已安裝 IBM MQ 的位置。然後從 IBM WebSphere MQ 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 MQ 的指令範例：

- **AIX** 在 AIX 系統上，您可以使用 **lsllpp** 指令：

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

- **Linux** 在 Linux 系統上，您可以使用 **rpm** 指令：

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

- **Solaris** 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
```

- **Windows** 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
```

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

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

然後從 IBM WebSphere MQ 7.1 或更新版本安裝中使用 **dspmqver** 指令。

- **Windows** 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
```

- **Windows** 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
```

reg.exe 指令只會顯示 IBM WebSphere MQ 7.1 或更新版本安裝的相關資訊。

相關概念

多個安裝

相關參考

[dspmqver](#)

[dspmqinst](#)

配置高可用性、回復及重新啟動

您可以透過在佇列管理程式失敗時維護佇列可用性，以及在伺服器或儲存體失敗之後回復訊息，讓應用程式具有高可用性。

關於這項作業

z/OS 在 z/OS 上，平台內建高可用性。您也可以使用佇列共用群組來改善伺服器應用程式可用性。請參閱 [共用佇列及佇列共用群組](#)。

Multi 在多平台上，您可以使用用戶端重新連線來自動在佇列管理程式群組之間切換用戶端，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，以增進用戶端應用程式可用性。IBM MQ classes for Java 不支援自動重新連接用戶端。多重實例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。您將伺服器應用程式部署至這個佇列管理程式。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。

在 Multiplatforms 上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。從 IBM WebSphere MQ 7.1 開始，叢集錯誤回復會重新執行導致問題的作業，直到解決問題為止。請參閱 [變更 z/OS 以外的伺服器上的叢集錯誤回復](#)。您也可以將 IBM MQ for Multiplatforms 配置為平台專用叢集作業解決方案的一部分，例如：

- Microsoft Cluster Server
- **IBM i** IBM i 上的 HA 叢集
- **Linux** **UNIX** PowerHA for AIX (舊稱為 AIX 上的 HACMP) 及其他 UNIX and Linux 叢集作業解決方案

Linux 在 Linux 系統上，您可以配置抄寫的資料佇列管理程式 (RDQ)，以實作高可用性或災難回復解決方案。為了高可用性，在由三部 Linux 伺服器組成的群組中，每一個節點上會配置相同佇列管理程式的實例。三個實例的其中一個是作用中實例。來自作用中佇列管理程式的資料會同步抄寫至其他兩個實例，因此其中一個實例可以在發生某個失敗時接管。對於災難回復，佇列管理程式會在一個站台的主要節點上執行，該佇列管理程式的次要實例位於另一個站台的回復節點上。在主要實例與次要實例之間抄寫資料，如果主要節點因故遺失，則可以將次要實例變成主要實例並啟動。

MQ Appliance 高可用性或災難回復解決方案的另一個選項是部署一對 IBM MQ 應用裝置。請參閱 IBM MQ Appliance 說明文件中的 [高可用性](#) 及 [災難回復](#)。

傳訊系統可確保輸入系統的訊息會遞送至其目的地。當訊息從一個佇列管理程式移至另一個佇列管理程式時，IBM MQ 可以使用 **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。IBM MQ 會維護處理訊息接收、傳輸及遞送之佇列管理程式活動的回復日誌。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 IBM MQ 時，請重新啟動回復。
2. 失敗回復，當失敗停止時 IBM MQ。
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。



注意：您無法將回復日誌移至不同的作業系統。

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。

相較之下，手動重新連線需要用戶端應用程式使用 MQCONN 或 MQCONNX 重新建立連線，以及重新開啟物件。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 368 頁的表 31 列出必須安裝在用戶端工作站上的最早 IBM 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 369 頁的表 32 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有權存取重新連線選項，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。對於抄寫的資料佇列管理程式或「IBM MQ 應用裝置」上的 HA 佇列管理程式，您可以指定作用中及待命佇列管理程式所使用的浮動 IP 位址，以簡化配置。
- 若為 JMS 用戶端，請在 Connection Factory 配置中設定重新連線選項。在 Java EE 伺服器的 EJB 儲存器內執行時，MDB 可以利用 IBM MQ 資源配接器 (或接聽器埠，如果在 WebSphere Application Server 中執行) 的啟動規格所提供的重新連接機制來重新連接至 IBM MQ。不過，如果應用程式不是 MDB (或在 Web 儲存器中執行)，則應用程式必須實作自己的重新連接邏輯，因為在此情況下不支援自動用戶端重新連接。IBM MQ 資源配接器提供將訊息遞送至訊息驅動 Bean 的這項重新連接能力，但其他 Java EE 元素 (例如 Servlet) 必須實作自己的重新連線。

註: IBM MQ classes for Java 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
	IBM MQ classes for Java	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

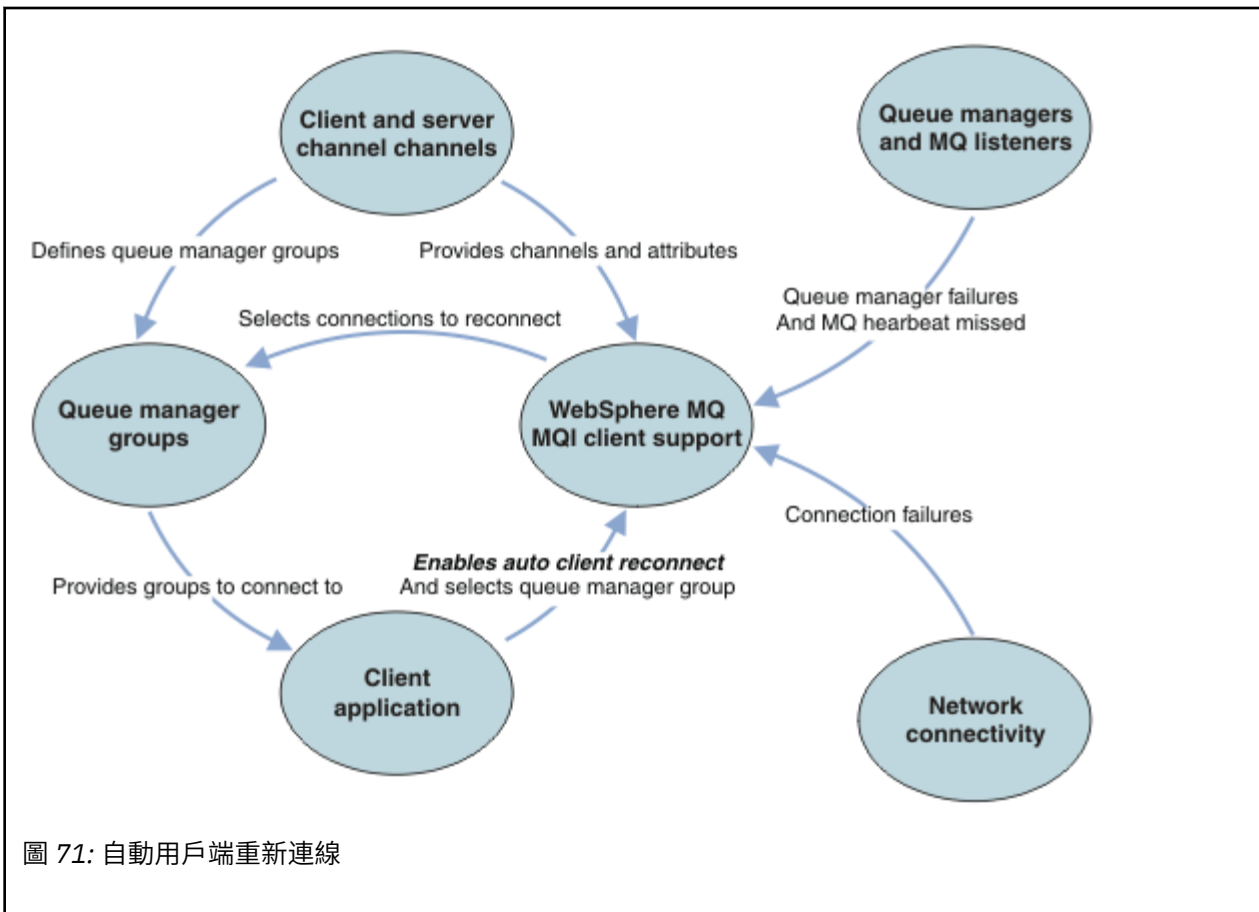
1. 在 WCF 連結配置中設定受管理或未受管理模式。

自動重新連線具有下列配置需求：

表 32: 自動重新連線配置需求

元件	需求	不符合需求的效果
IBM MQ MQI client 版安裝	請參閱 第 368 頁的表 31	MQRC_OPTIONS_ERROR
IBM 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
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 369 頁的圖 71 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 MQ MQI client。如需 JMS 用戶端自動用戶端重新連線的詳細資料，請參閱 [使用自動 JMS 用戶端重新連線](#)。

- 依預設，不會自動重新連接用戶端。透過設定 MQCONNX 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 443 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，會傳回此原因碼。當設定重新連接選項時。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。

高可用性佇列管理程式

高可用性佇列管理程式具有一個作用中實例，以及一個以上佇列管理程式待命實例。作用中佇列管理程式會與待命佇列管理程式同步，以便在作用中實例失敗時待命佇列管理程式可以自動接管。有許多不同的解決方案可提供高可用性佇列管理程式，請參閱 [第 376 頁的『高可用性配置』](#)。

在高可用性佇列管理程式啟動其待命實例之後，您可以使用自動用戶端重新連線來簡化重新啟動 IBM MQ MQI client 應用程式的作業。

高可用性佇列管理程式的待命實例通常與作用中實例位於不同的網址。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。抄寫的資

料佇列管理程式及 IBM MQ 應用裝置高可用性佇列管理程式支援浮動 IP 位址，您可以在其中指定單一地址，以與作用中或待命佇列管理程式搭配使用。

佇列管理程式群組


通常，IBM MQ MQI clients 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「IBM MQ MQI client」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。

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 IBM MQ MQI client 重新連接至不同的佇列管理程式，則它會失敗。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。

佇列共用群組

 z/OS 佇列共用群組的自動用戶端重新連線，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 IBM 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 KAINTE 和 HBINTE 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 IBM MQ MQI client 的網路失敗。

- IBM MQ 看不見傳輸自動執行的重新連線。
- 設定 HBINTE 有助於處理 IBM MQ 看不見的網路失敗。

佇列管理程式及 IBM 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。

IBM MQ MQI client 自動重新連線支援

如果您在 IBM MQ MQI client 中使用自動用戶端重新連線支援，則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在 `endmqm` 指令上指定 `-r`，重新連接選項
- 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 445 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。

雖然預期的重新連線時間不超過一分鐘，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保留不屬於 IBM 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。

相關概念

[可重新連接的用戶端](#)

相關工作

[停止佇列管理程式](#)

在 IBM MQ for z/OS 上，有一些由佇列管理程式或通道起始程式發出的參考訊息，應視為特別重要。這些訊息本身並不表示有問題，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。

這些主控台訊息的存在也可能指出使用者應用程式正在將大量訊息放入頁面集，這可能是更大問題的症狀：

- PUT 訊息 (例如不受控制的迴圈) 的使用者應用程式發生問題。
- 從佇列取得訊息的使用者應用程式不再運作。

要監視的主控台訊息

下列清單概述可能指出較大問題的訊息。請判斷是否需要使用系統自動化來追蹤這些訊息，並提供適當的文件，以便有效地追蹤任何潛在問題。

CSQI004I: csect-name 考量索引 佇列名稱 BY index-type FOR connection-type CONNECTION connection-name, num-msgs 跳過訊息

- 佇列管理程式偵測到應用程式從未定義索引的佇列中，依訊息 ID 或相關性 ID 來接收訊息。
- 請考量透過將本端佇列物件 *queue-name*、INDXTYPE 屬性變更為具有值 *index-type*，來建立所識別佇列的索引。

CSQI031I: csect-name 頁面集的新延伸範圍 psid 已順利格式化

- 請檢查配置給此頁集之佇列的現行深度。
- 請調查無法處理訊息的原因。

CSQI041I: csect-name JOB jobname USER userid HAD ERROR ACCESSING PAGE SET psid

- 判斷是否將頁集配置給佇列管理程式。
- 發出 **DISPLAY USAGE** 指令，以判定頁集的狀態。
- 請檢查佇列管理程式工作日誌，以取得其他錯誤訊息。

CSQI045I: csect-name 日誌 RBA 已達到 rba。規劃日誌重設

- 請計劃在方便的時間停止佇列管理程式，並重設日誌。
- 如果佇列管理程式使用 6 位元組日誌 RBA，請考量將佇列管理程式轉換為使用 8 位元組日誌 RBA。

CSQI046E: csect-name 日誌 RBA 已達到 rba。執行日誌重設

- 請計劃在方便的時間停止佇列管理程式，並重設日誌。
- 如果佇列管理程式使用 6 位元組日誌 RBA，請考量將佇列管理程式轉換為使用 8 位元組日誌 RBA。

CSQI047E: csect-name 日誌 RBA 已達到 rba。停止佇列管理程式並重設日誌

- 立即停止佇列管理程式並重設日誌。
- 如果佇列管理程式使用 6 位元組日誌 RBA，請考量將佇列管理程式轉換為使用 8 位元組日誌 RBA。

CSQJ004I: ACTIVE LOG COPY n INACTIVE, LOG IN SINGLE MODE, ENDRBA= ttt

- 佇列管理程式已啟動「單一」記載模式。這通常表示日誌卸載問題。
- 發出 **DISPLAY LOG** 指令，以判定您的作用中日誌和保存日誌雙工設定。此顯示畫面也會顯示有多少作用中日誌需要卸載處理。
- 請檢查佇列管理程式工作日誌，以取得其他錯誤訊息

CSQJ031D: csect-name, 日誌 RBA 範圍必須重設。回覆 'Y' 以繼續啟動或 'N' 以關閉

- 請停止佇列管理程式並儘快重設日誌，然後重設日誌。
- 如果佇列管理程式使用 6 位元組日誌 RBA，請考量將佇列管理程式轉換為使用 8 位元組日誌 RBA。

CSQJ032E: csect-name alert-lvl -ATING END OF THE LOG RBA RANGE of max-rba。 CURRENT LOG RBA IS current-rba。

- 計劃儘快停止佇列管理程式並重設日誌。
- 如果佇列管理程式使用 6 位元組日誌 RBA，請考量將佇列管理程式轉換為使用 8 位元組日誌 RBA。

CSQJ110E: LAST COPYn 作用中日誌資料集為 nnn 百分比已滿

- 執行顯示要求來判斷與日誌卸載處理程序相關的未完成要求，以採取步驟來完成其他等待中卸載作業。請採取必要動作來滿足任何要求，並允許卸載繼續。
- 請考量是否有足夠的作用中日誌資料集。必要的話，您可以使用 DEFINE LOG 指令來動態新增其他日誌資料集。

CSQJ111A: ACTIVE LOG DATA SETS 中的空間用盡

- 執行顯示要求，以確保沒有與日誌卸載處理程序相關的未完成要求。請採取必要動作來滿足任何要求，並允許卸載繼續。
- 請考量是否有足夠的作用中日誌資料集。必要的話，您可以使用 DEFINE LOG 指令來動態新增其他日誌資料集。
- 如果由於缺少卸載所需的資源而導致延遲，則必須提供必要的資源以容許卸載完成，從而允許記載繼續進行。如需從此狀況回復的相關資訊，請參閱 保存日誌問題。

CSQJ114I: 保存資料集上發生錯誤，卸載只繼續產生一個保存資料集

- 請檢查佇列管理程式工作日誌，以取得其他錯誤訊息。
- 建立保存日誌的第二個副本，並手動更新您的 BSDS。

CSQJ115E: 卸載失敗，無法配置保存資料集

檢閱訊息 CSQJ103E 或 CSQJ073E 的錯誤狀態資訊。請更正導致資料集配置錯誤的狀況，以便在重試時發生卸載。

CSQJ136I: 無法配置磁帶機以進行連線-ID= xxxx CORRELATION-ID= yyyyyy, m ALLOCATED n ALLOWED

- 請檢查佇列管理程式工作日誌，以取得其他錯誤訊息。

CSQJ151I: csect-name ERROR READING RBA rrr, CONNECTION-ID= xxxx 相關性-ID= yyyyyy REASON CODE= ccc

- 請檢查佇列管理程式工作日誌，以取得其他訊息。
- 發出 **DISPLAY CONN** 指令，以判定哪個連線未確定其活動。
- 請確定應用程式可以確定其更新項目。

CSQJ160I: LONG UOW FOUND, URID= urid CONNECTION NAME= name

- 請檢查佇列管理程式工作日誌，以取得其他訊息。
- 發出 **DISPLAY CONN** 指令，以判定哪個連線未確定其活動。
- 請確定應用程式可以確定其更新項目。

CSQJ161I: UOW 在 n OFFLOADS, URID= urid CONNECTION NAME= name

- 判斷是否將頁集配置給佇列管理程式。
- 發出 **DISPLAY USAGE** 指令，以判定頁集的狀態。
- 請檢查佇列管理程式工作日誌，以取得其他訊息。

CSQP011E: CONNECT ERROR STATUS ret-code for PAGE SET psid

- 請檢查配置給此頁集之佇列的現行深度。
- 請調查處理訊息失敗的原因。

CSQP013I: csect-name NEW EXTENT CREATED FOR PAGE SET psid. 現在將格式化新的延伸範圍

- 請檢查配置給此頁集之佇列的現行深度。
- 請調查處理訊息失敗的原因。
- 判斷佇列是否需要重新定位至另一個頁集。
- 如果磁區已滿，請判斷您是否需要使頁集成為多磁區資料集。如果頁集已經是磁區，請考慮將更多磁區新增至正在使用的儲存體群組。一旦有更多空間可用，請將頁集 **EXPAND** 方法設為 **SYSTEM**，以重試擴充。如果需要重試，請將 **EXPAND** 切換至 **SYSTEM**，然後回到正常設定。

CSQP014E: csect-name 針對頁集 psid 擴充失敗。未來延伸要求將被拒絕

- 請檢查配置給此頁集之佇列的現行深度。
- 請調查處理訊息失敗的原因。
- 判斷佇列是否需要重新定位至另一個頁集。

CSQP016E: csect-name PAGE SET psid 已達到延伸範圍數目上限。無法重新延伸

- 請檢查配置給此頁集之佇列的現行深度。
- 請調查處理訊息失敗的原因。

CSQP017I: csect-name EXPANSION STARTED FOR PAGE SET psid

發出 DISPLAY THREAD 指令，以判定 IBM MQ 中「工作單元」的狀態。

CSQP047E: 無法使用的頁集可能導致問題-請採取動作以更正此狀況

- 遵循系統程式設計師回應。

CSQQ008I: nn 回復單元在佇列管理程式 qqqq 中仍不確定。

- 調查無法傳送郵件的佇列狀態。請確定無法傳送的郵件佇列未停用 PUT。
- 請確定無法傳送的郵件佇列未達到 MAXMSG 限制。

CSQQ113I: psb-name region-id 無法處理此訊息

- 請檢查 CSQOUTX 資料集，以判斷 CSQINPX 失敗的原因。
- 部分指令可能未處理。

CSQX035I: csect-name 連線至佇列管理程式 qmgr-name 停止或中斷，MQCC= mqcc MQRC= mqrc (mqrc-text)

- 請檢查 MQRC 以判斷失敗的原因。
- 這些代碼記錄在 IBM MQ for z/OS 訊息、完成及原因碼中。

CSQX032I: csect-name 起始設定指令處理程式已終止

- 請檢查 MQRC 以判斷失敗的原因。
- 這些代碼記錄在 IBM MQ for z/OS 訊息、完成及原因碼中。

CSQX048I: csect-name 無法轉換名稱的訊息，MQCC= mqcc MQRC= mqrc (mqrc-text)

- 請檢查工作日誌，以判斷 TCP/IP 失敗的原因。
- 請檢查 TCP/IP 位址空間是否有錯誤。

CSQX234I: csect-name 接聽器已停止，TRPTYPE= trptype INDISP= disposition

- 如果接聽器未停止，請在 **STOP** 指令之後，檢查 TCP/IP 位址空間是否有錯誤。
- 遵循系統程式設計師回應。

CSQX407I: csect-name 叢集佇列 q-name 定義不一致

- 叢集內多個叢集佇列的值不一致。調查並解決差異。

CSQX411I: csect-name 儲存庫管理程式已停止

- 如果儲存庫管理程式因錯誤而停止，請檢查工作日誌中的訊息。

CSQX417I: csect-名稱 叢集-移除的佇列管理程式仍有傳送端 qmgr-name

- 遵循系統程式設計師回應。

CSQX418I: csect-name 只有叢集 cluster_name 的一個儲存庫

- 為了增加高可用性，叢集應該配置兩個完整儲存庫。

CSQX419I: csect-name 沒有叢集接收端 cluster_name

- 遵循系統程式設計師回應。

CSQX420I: csect-name 沒有叢集 cluster_name 的儲存庫

- 遵循系統程式設計師回應。

CSQX448E: csect-name 儲存庫管理程式因錯誤而停止。在 n 秒內重新啟動

- 遵循系統程式設計師回應。

此訊息每 600 秒 (10 分鐘) 就會輸出一次，直到 SYSTEM.CLUSTER.COMMAND.QUEUE :

```
ALTER QLOCAL (SYSTEM.CLUSTER.COMMAND.QUEUE) GET (ENABLED)
```

在啟用佇列之前，在發出第一個 CSQX448E 訊息之前，可能需要人為介入來解決導致儲存庫管理程式結束的問題。

CSQX548E: csect-name 傳送至本端無法傳送郵件的佇列、通道 channel-name reason=mqrc (mqrc-text)

- 遵循系統程式設計師回應。

CSQX788I: csect-name 使用函數 'func' 查閱位址 address 的 DNS 花費 n 秒

- 遵循系統程式設計師回應。


CSQY225E: csect-name 佇列管理程式嚴重缺乏高於限制採取動作的本端儲存體


- 佇列管理程式正在執行高於限制的虛擬儲存體嚴重不足。應該採取動作來緩解狀況，並避免佇列管理程式可能異常終止。

CSQ5038I: csect-name 自 hh.mm.ss.nnnnnn 以來，服務作業 service-task 一直沒有回應。檢查 Db2 的問題

- 遵循系統程式設計師回應。

高可用性配置

如果您想要在高可用性 (HA) 配置中操作 IBM 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)，或具有 IBM MQ 多重實例佇列管理程式。  在 Linux 系統上，您也可以部署抄寫的資料佇列管理程式 (RDQM)，其使用仲裁型群組來提供高可用性。

 高可用性或災難回復解決方案的另一個選項是部署一對 IBM MQ 應用裝置。請參閱 IBM MQ Appliance 說明文件中的 [高可用性](#) 及 [災難回復](#)。

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。從 IBM WebSphere MQ 7.1 開始，叢集錯誤回復會重新執行導致問題的作業，直到解決問題為止。

HA 叢集

HA 叢集是由兩部以上電腦及資源 (例如磁碟及網路) 組成的群組，它們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，則為高可用性管理程式 (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 377 頁的『[HA 叢集與佇列管理程式叢集的關係](#)』中說明 IBM 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 IBM MQ 來配置您自己的高可用性傳訊系統，而不需要 HACMP 或 MSCS 之類的叢集技術。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。

V 9.1.0 高可用性抄寫資料佇列管理程式 (HA RDQM)

在由三部 Linux 伺服器組成的群組中，每一個節點上所配置的可相同佇列管理程式實例。三個實例的其中一個是作用中實例。來自作用中佇列管理程式的資料會同步抄寫至其他兩個實例，因此其中一個實例可以在發生某個失敗時接管。伺服器的分組由 Pacemaker 控制，並由 DRBD 進行抄寫。

V 9.1.0 災難回復抄寫的資料佇列管理程式 (DR RDQM)

佇列管理程式在一個站台的主要節點上執行，該佇列管理程式的次要實例位於不同站台的回復節點上。在主要實例與次要實例之間抄寫資料，如果主要節點因故遺失，則可以將次要實例變成主要實例並啟動。兩個節點都必須是 Linux 伺服器。抄寫由 DRBD 控制。

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 IBM 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 IBM MQ Explorer 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

佇列管理程式叢集在佇列管理程式叢集佇列的可用實例之間提供訊息負載平衡。這提供比單一佇列管理程式更高的可用性，因為在佇列管理程式失敗之後，傳訊應用程式仍然可以將訊息傳送至佇列管理程式叢集佇列的現存實例，以及存取這些實例。不過，雖然佇列管理程式叢集會自動將新訊息遞送至叢集中的可用佇列管理程式，但在該佇列管理程式重新啟動之前，目前在無法使用的佇列管理程式上佇列的訊息無法使用。因此，僅佇列管理程式叢集並不會提供所有訊息資料的高可用性，或提供佇列管理程式失敗的自動偵測，以及佇列管理程式重新啟動或失效接手的自動觸發。高可用性 (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。如需佇列管理程式叢集的簡介，請參閱 [設計叢集](#)。

Linux UNIX UNIX and Linux 上的 HA 叢集

您可以在 UNIX and Linux 平台上搭配使用 IBM MQ 與高可用性 (HA) 叢集: 例如, PowerHA for AIX (舊稱為 HACMP)、Veritas Cluster Server、HP Serviceguard 或 Red Hat Enterprise Linux 叢集與 Red Hat Cluster Suite。

在 IBM WebSphere MQ 7.0.1 之前, 已提供 SupportPac MC91 來協助配置 HA 叢集。對於佇列管理程式儲存其資料的位置, IBM WebSphere MQ 7.0.1 提供比舊版更大的控制程度。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 378 頁的『HA 叢集配置』、HA 叢集與佇列管理程式叢集的關係、第 378 頁的『IBM MQ 用戶端』及第 379 頁的『在 HA 叢集中運作的 IBM 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 IBM MQ 來協助設定待命或接管配置, 包括所有叢集節點執行 IBM MQ 工作量的交互接管。

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。

IBM MQ 用戶端

必須寫入與佇列管理程式通訊且可能需要重新啟動或接管的 IBM MQ 用戶端, 以容忍連線中斷, 且必須反覆地嘗試重新連接。IBM WebSphere MQ 7 在處理「用戶端通道定義表 (CCDT)」時引進了一些特性, 可協助連線可用性及工作量平衡; 不過, 這些特性在使用失效接手系統時並不直接相關。

只要用戶端連接至相同的佇列管理程式, 交易式功能可讓 IBM MQ MQI client 參與兩階段交易。交易式功能無法使用技術 (例如 IP 負載平衡器) 從佇列管理程式清單中進行選取。當您使用 HA 產品時, 佇列管理程式會維護其身分 (名稱和位址), 不論它在哪一個節點上執行, 因此交易式功能可以與受到 HA 控制的佇列管理程式搭配使用。

在 HA 叢集中運作的 IBM 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。

IBM 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 IBM 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。

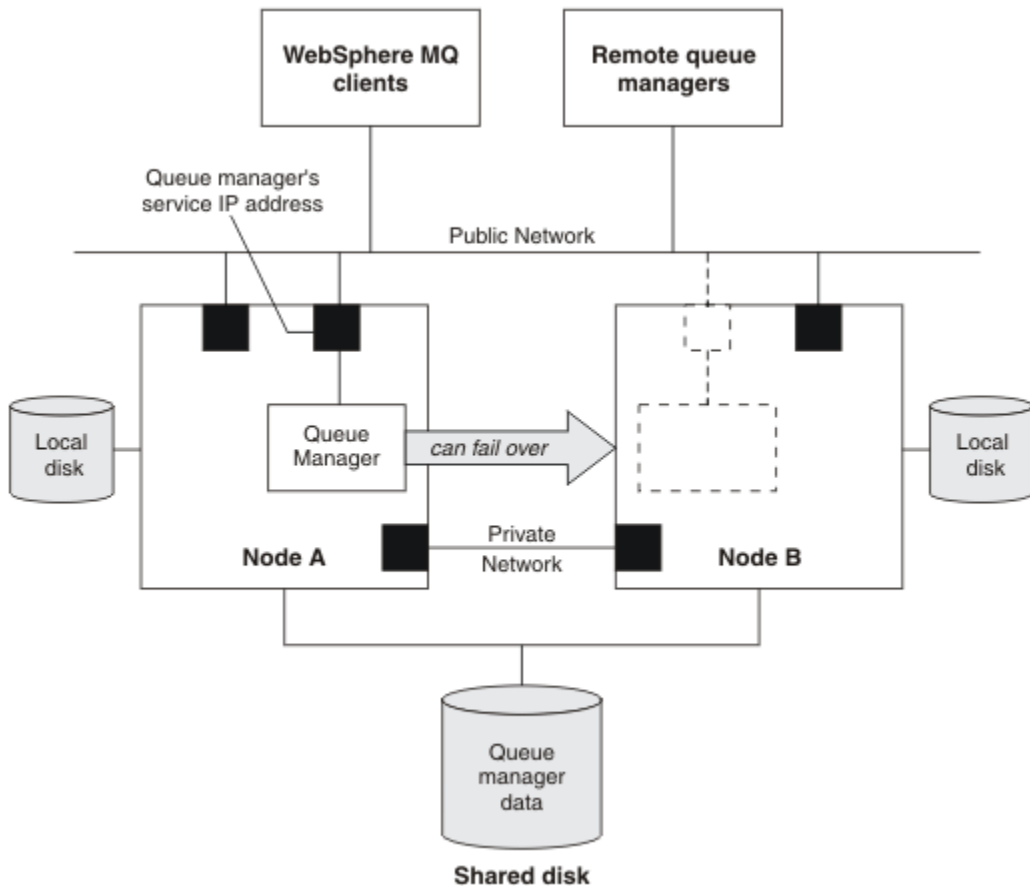


圖 72: HA 叢集

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。IBM MQ 以 mqm 使用者身分執行。IBM 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 IBM MQ；啟動、停止或監視 Script 的 Script 必須切換至 mqm 使用者。

註：IBM MQ 必須正確安裝在所有節點上；您無法共用產品執行檔。

Linux UNIX 在 UNIX and Linux 上配置共用磁碟

HA 叢集中的 IBM MQ 佇列管理程式需要共用磁碟上共用具名遠端檔案系統中的資料檔及日誌檔。

關於這項作業

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。

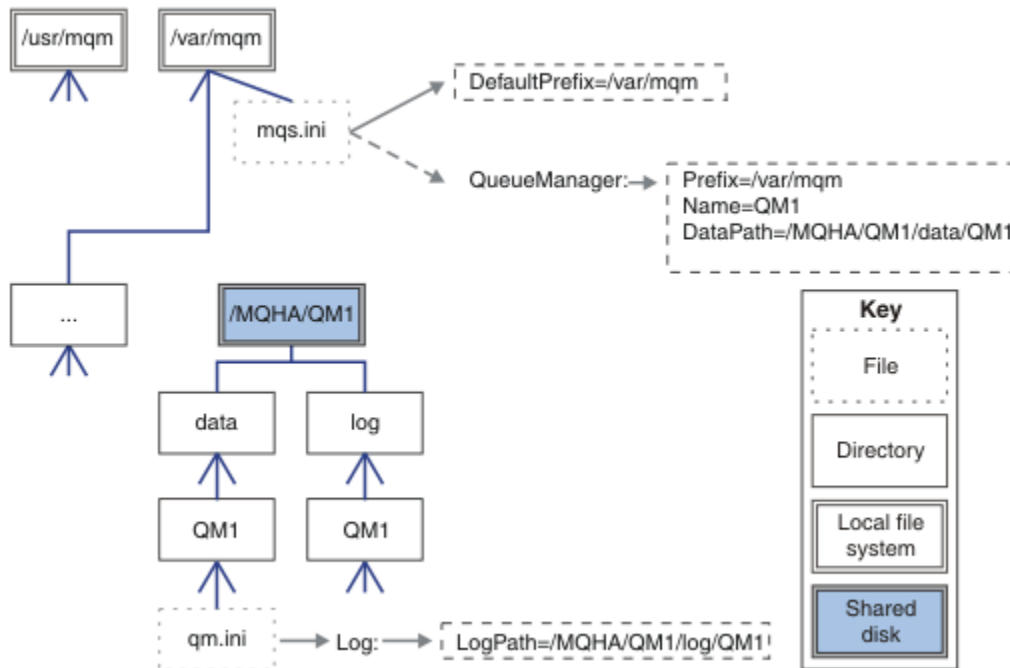


圖 73: 共用具名 data 及 log 目錄

程序

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。

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。

關於這項作業

若要建立佇列管理程式以在 HA 叢集中使用，您必須先選取叢集中要建立佇列管理程式的其中一個節點，然後在此節點上完成下列步驟。

程序

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 **strmqm** 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 **dspmqrinf** 指令來顯示 **addmqinf** 指令：

```
dspmqrinf -o command qmgrname
```

其中 **qmgrname** 是佇列管理程式的名稱。

如需使用 **addmqinf** 指令的相關資訊，請參閱 [第 381 頁的『將佇列管理程式配置新增至 UNIX and Linux 上的其他 HA 叢集節點』](#)。

addmqinf 指令的顯示方式與下列範例類似：

```
addmqinf -sQueueManager -vName=qmgrname -vDirectory=qmgrname \  
-vPrefix=/var/mqm -vDataPath=/MQHA/qmgrname/data/qmgrname
```

7. 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。

下一步

現在，您已準備好完成 [第 381 頁的『將佇列管理程式配置新增至 UNIX and Linux 上的其他 HA 叢集節點』](#) 中說明的步驟。

點

您必須將佇列管理程式配置資訊新增至 HA 叢集中的其他節點。

開始之前

在完成此作業之前，您必須已完成 [第 381 頁的『在 UNIX and Linux 上建立 HA 叢集佇列管理程式』](#) 中的步驟。建立佇列管理程式之後，您必須在其他每一個節點上完成下列步驟，將佇列管理程式的配置資訊新增至 HA 叢集中的其他每一個節點。

關於這項作業

當您建立佇列管理程式以在 HA 叢集中使用時，必須先選取叢集中要在其上建立佇列管理程式的其中一個節點，如 [第 381 頁的『在 UNIX and Linux 上建立 HA 叢集佇列管理程式』](#) 中所述。

程序

1. 裝載佇列管理程式檔案系統。
2. 將佇列管理程式配置資訊新增至節點。

有兩種方式可新增配置資訊：

- 直接編輯 `/var/mqm/mqs.ini`。
- 發出 **addmqinf** 指令，該指令由第 381 頁的『在 UNIX and Linux 上建立 HA 叢集佇列管理程式』中步驟 6 中的 **dspmqinf** 指令所顯示。

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 MQ 安裝中發出。若要從不同於目前與佇列管理程式相關聯的安裝來啟動及停止佇列管理程式，您必須先使用 **setmqm** 指令來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。

Linux UNIX 在 UNIX and Linux 上啟動 HA 叢集佇列管理程式的 Shell Script 範例

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 amqcrista amqrrmfa amqrmppa \
amqzfuma amqmuf0 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 stirmqm command does not use the -x flag.
stirmqm ${QM}
```

您可以修改 Script 來啟動其他相關程式。

Linux

UNIX

在 UNIX and Linux 上停止 HA 叢集佇列管理程式的範例 Shell

Script

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佇列管理程式的範例。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 382 頁的『在 UNIX and Linux 上啟動 HA 叢集佇列管理程式的 Shell Script 範例』中提供 Shell Script 範例。）

```
#!/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 amqzxma0 amqfcxba amqfqpud amqpcsea amqzlaa0 \
            amqzlsa0 runmqchi runmqlsr amqcrsta amqrrmfa amqzmpa \
            amqzfuma 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

```

註: 視針對特定佇列管理程式執行的處理程序而定，此 Script 中包含的佇列管理程式處理程序清單可能不是完整清單，或可能包含比針對該佇列管理程式執行的處理程序更多的處理程序：

```

for process in amqzmuc0 amqzma0 amqfcxba amqfpub amqpcsea amqzlaa0 \
    amqzlsa0 runmqchi runmqslr amqcrista amqirmfa amqimppa \
    amqzfuma amqmuf0 amqzmur0 amqzmgr0

```

可以根據所配置的特性，以及針對特定佇列管理程式執行的處理程序，在此清單中併入或排除處理程序。如需處理程序的完整清單，以及依特定順序停止處理程序的相關資訊，請參閱 [在 UNIX 及 Linux 上手動停止佇列管理程式](#)。

Linux → UNIX 在 UNIX and Linux 上監視 HA 叢集佇列管理程式

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 IBM WebSphere MQ 7.1 開始，可以在系統上同時安裝多個 IBM MQ。如需多個安裝的相關資訊，請參閱 [多個安裝](#)。如果您想要跨多個安裝 (包括位於 IBM WebSphere MQ 7.1 或更高版本的安裝) 使用監視 Script，則可能需要執行一些其他步驟。如果您具有主要安裝，或使用的 Script 版本早於 IBM WebSphere MQ 7.1，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地識別 `MQ_INSTALLATION_PATH`：

1. 從 IBM WebSphere MQ 7.1 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 不提供為佇列管理程式的監視程式提供參數的方法。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 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 382 頁的『在 UNIX and Linux 上啟動 HA 叢集佇列管理程式的 Shell Script 範例』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 stirmqm | 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
```

```
fi
fi
exit $result
```

Linux **UNIX** 在 *UNIX and Linux* 上將佇列管理程式置於高可用性叢集控制下
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。

關於這項作業

若要將佇列管理程式置於高可用性叢集的控制之下，您必須定義資源群組來包含佇列管理程式及其所有相關聯資源。

程序

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。

Linux **UNIX** 在 *UNIX and Linux* 上刪除 HA 叢集佇列管理程式
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。

關於這項作業

若要從 HA 叢集中的節點移除佇列管理程式，您必須移除其配置資訊。

程序

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`
3. 選擇性的: 若要完全刪除佇列管理程式，請使用 **dltmqm** 指令。
重要: 請注意，使用 **dltmqm** 指令刪除佇列管理程式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。

Windows 支援 Microsoft 叢集服務 (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 IBM MQ for Windows。

「Microsoft 叢集服務 (MSCS)」可讓您將伺服器連接至叢集，以提供更高的資料及應用程式可用性，並更容易管理系統。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 387 頁的『簡介 MSCS 叢集』](#)
- [第 388 頁的『為 MSCS 叢集作業設定 IBM MQ』](#)

然後，在下列各節中告訴您如何配置 IBM MQ 以進行 MSCS 叢集作業:

- [第 389 頁的『建立佇列管理程式以與 MSCS 搭配使用』](#)
- [第 390 頁的『將佇列管理程式移至 MSCS 儲存體』](#)
- [第 391 頁的『將佇列管理程式置於 MSCS 控制下』](#)
- [第 396 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 IBM MQ 的一些有用提示，並詳細說明 IBM MQ MSCS 支援公用程式：

- [第 397 頁的『使用 MSCS 的提示和要訣』](#)
- [第 399 頁的『支援 MSCS 公用程式』](#)

Windows 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 376 頁的『高可用性配置』](#) 包含 MSCS 叢集、多重實例佇列管理程式及 IBM 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。這與本手冊中其他位置說明的 IBM MQ 叢集不同。

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。還有另一個共用磁碟，必須是獨立磁碟的備援陣列 (RAID) 層次 1，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 387 頁的圖 74](#) 顯示兩台電腦的叢集的一般設定。

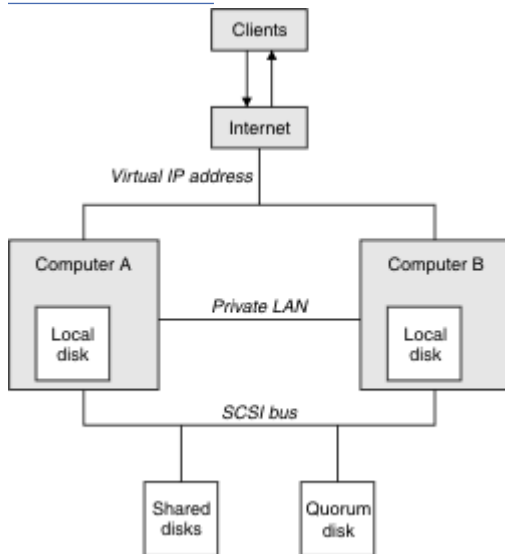


圖 74: 雙電腦 MSCS 叢集

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。

Windows 為 MSCS 叢集作業設定 IBM 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 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 MQ 開始。

因為佇列管理程式與 IBM MQ 安裝名稱相關聯，所以叢集中所有電腦上的 IBM 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 MQ 隨附的工具來移轉它；請參閱 [第 390 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 389 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 391 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 396 頁的『從 MSCS 控制項移除佇列管理程式』](#) 中說明的程序。

Windows 設定對稱及 MSCS

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，請確保在兩台電腦上安裝的所有必要軟體在版本、維護層次、SupportPacs、路徑及結束程式方面都相同，並確保存在一般名稱空間(安全環境)，如 [第 388 頁的『MSCS 安全』](#) 中所述。

Windows 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。如果您已具有網域，且與 IBM 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，IBM 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 IBM 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。

Windows 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。

Windows 叢集模式和 MSCS

有兩種模式可讓您在 Windows 上使用 IBM 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 IBM MQ, 這表示您可以執行兩個佇列管理程式, 分別在 A 和 B 上執行一個佇列管理程式, 每個佇列管理程式都利用其專屬電腦的完整功能。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。

Windows 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 IBM MQ Explorer 中以兩種方式建立佇列管理程式, 以與 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 MQ Explorer 會自動啟動新的佇列管理程式, 並將它設為在建立之後自動啟動。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:\IBM 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。如果您未設定此變數, 則會在佇列管理程式的 IBM 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 390 頁的『將佇列管理程式移至 MSCS 儲存體』。

使用 IBM MQ Explorer 建立佇列管理程式

請遵循下列步驟, 以使用 IBM MQ Explorer 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 MQ Explorer。
2. 在「Navigator」視圖中，展開樹狀結構節點以尋找 佇列管理程式 樹狀結構節點。
3. 用滑鼠右鍵按一下 佇列管理程式 樹狀結構節點，然後選取 **新建 > 佇列管理程式**。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 **下一步>**。
5. 完成對話框 (步驟 2)，然後按 **下一步>**。
6. 完成對話框 (步驟 3)，確定未選取 **啟動佇列管理程式** 和 **建立伺服器連線通道**，然後按 **下一步>**。
7. 完成對話框 (步驟 4)，然後按一下 **完成**。
8. 繼續進行 [第 390 頁的『將佇列管理程式移至 MSCS 儲存體』](#)。

Windows 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\QMname 和 C:\WebSphere MQ\qmgrs\QMname 之類的路徑。



小心: 請勿嘗試手動移動檔案; 請依照本主題的說明，使用提供作為「IBM MQ MSCS 支援」一部分的公用程式。

如果要移動的佇列管理程式使用 TLS 連線，且 TLS 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他佇列管理程式一起移至共用磁碟。依預設，指定 TLS 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 IBM 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 MQ Explorer 或 MQSC 指令 ALTER QMGR 來修改佇列管理程式屬性 SSLKEYR，以指向新的 TLS 金鑰儲存庫檔。

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 391 頁的『將佇列管理程式置於 MSCS 控制下』](#)。
3. 對佇列檔和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 398 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: \
IBM MQ " /ld " e: \
IBM MQ \log"
```

將您的佇列管理程式名稱替換為 *qmname*，將您的共用磁碟機代號替換為 *e*，並將您選擇的目錄替換為 *IBM MQ*。如果目錄尚未存在，則會建立它們。

8. 使用「IBM MQ Explorer」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 **啟動**。即會啟動佇列管理程式。
 - b. 用滑鼠右鍵按一下 佇列 樹狀結構節點，然後選取 **新建 > 本端佇列 ...**，並提供佇列名稱。
 - c. 按一下 **完成**。
 - d. 用滑鼠右鍵按一下佇列，然後選取 **放置測試訊息 ...**。即會顯示「放置測試訊息」畫面。
 - e. 輸入一些訊息文字，然後按一下 **放置測試訊息**，並關閉畫面。

- f. 用滑鼠右鍵按一下佇列，然後選取 **瀏覽訊息 ...**。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 **關閉**。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 **清除訊息 ...**。清除佇列上的訊息。
 - i. 用滑鼠右鍵按一下佇列，然後選取 **刪除 ...**。即會顯示確認畫面，按一下 **確定**。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 **停止 ...**。即會顯示「結束佇列管理程式」畫面。
 - k. 按一下 **確定**。佇列管理程式停止。
9. 身為 IBM MQ 管理者，請確定佇列管理程式的啟動屬性設為手動。在 IBM MQ Explorer 中，將佇列管理程式內容畫面中的啟動欄位設為 manual。
 10. 繼續進行 [第 391 頁的『將佇列管理程式置於 MSCS 控制下』](#)。

Windows 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 MQ 及其 MSCS 支援，且每一部電腦上的軟體都相同，如 [第 388 頁的『為 MSCS 叢集作業設定 IBM MQ』](#) 中所述。
2. 使用 **haregtyp** 公用程式，將 IBM 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 399 頁的『支援 MSCS 公用程式』](#)。
3. 如果您尚未建立佇列管理程式，請參閱 [第 389 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，請確定您已執行「[第 390 頁的『將佇列管理程式移至 MSCS 儲存體』](#)」中的程序。
5. 使用命令提示字元或「IBM 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

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 **MQSeries MSCS** 資源。
 - f. 完成精靈中的其餘步驟。
5. 在讓資源上線之前，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 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；为了更好的隔離
 - 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 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 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 399 頁的『MSCS 中的 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 399 頁的『MSCS 中的 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 396 頁的『在 MSCS 上輪詢 looksAlive 及 isAlive』。
 - d. 選擇性地設定偏好節點 (但請注意第 399 頁的『在 MSCS 中使用偏好的節點』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 MQ 探險家」。例如：
- a. 用滑鼠右鍵按一下佇列樹狀結構節點，然後選取 **新建 > 本端佇列 ...**，並提供佇列名稱。
 - b. 按一下 **完成**。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 **放置測試訊息 ...**。即會顯示「放置測試訊息」畫面。
 - d. 輸入一些訊息文字，然後按一下 **放置測試訊息**，並關閉畫面。
 - e. 用滑鼠右鍵按一下佇列，然後選取 **瀏覽訊息 ...**。即會顯示「訊息瀏覽器」畫面。
 - f. 確定您的訊息位於佇列上，然後按一下 **關閉**。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 **清除訊息 ...**。清除佇列上的訊息。
 - h. 用滑鼠右鍵按一下佇列，然後選取 **刪除 ...**。即會顯示確認畫面，按一下 **確定**。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 390

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：

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 > 管理叢集 ...** 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 **服務及應用程式 > 配置服務或應用程式 ...** 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 **MQSeries MSCS** 資源。
 - f. 完成精靈中的其餘步驟。
5. 在讓資源上線之前，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 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**；為了更好的隔離
- **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 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 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 399 頁的『MSCS 中的 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 399 頁的『MSCS 中的 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 396 頁的『在 MSCS 上輪詢 looksAlive 及 isAlive』](#)。

- d. 選擇性地設定偏好節點 (但請注意 [第 399 頁的『在 MSCS 中使用偏好的節點』](#) 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 MQ 探險家」。例如:
- a. 用滑鼠右鍵按一下 佇列 樹狀結構節點, 然後選取 **新建 > 本端佇列 ...**, 並提供佇列名稱。
 - b. 按一下**完成**。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 **放置測試訊息 ...**。即會顯示「**放置測試訊息**」畫面。
 - d. 輸入一些訊息文字, 然後按一下 **放置測試訊息**, 並關閉畫面。
 - e. 用滑鼠右鍵按一下佇列, 然後選取 **瀏覽訊息 ...**。即會顯示「**訊息瀏覽器**」畫面。
 - f. 確定您的訊息位於佇列上, 然後按一下 **關閉**。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 **清除訊息 ...**。清除佇列上的訊息。
 - h. 用滑鼠右鍵按一下佇列, 然後選取 **刪除 ...**。即會顯示確認畫面, 按一下 **確定**。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 390 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 388 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地址。

7. 建立佇列管理程式的資源實例。

建立類型 *IBM 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 ; 为了更好的隔離
- 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 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 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 399 頁的『MSCS 中的 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 399 頁的『MSCS 中的 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 396 頁的『在 MSCS 上輪詢 looksAlive 及 isAlive』。

8. 選擇性地設定偏好節點 (但請注意第 399 頁的『在 MSCS 中使用偏好的節點』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 MQ 探險家」。例如：
 - a. 用滑鼠右鍵按一下 佇列 樹狀結構節點，然後選取 **新建 > 本端佇列 ...**，並提供佇列名稱。
 - b. 按一下**完成**。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 **放置測試訊息 ...**。即會顯示「**放置測試訊息**」畫面。
 - d. 輸入一些訊息文字，然後按一下 **放置測試訊息**，並關閉畫面。
 - e. 用滑鼠右鍵按一下佇列，然後選取 **瀏覽訊息 ...**。即會顯示「**訊息瀏覽器**」畫面。
 - f. 確定您的訊息位於佇列上，然後按一下 **關閉**。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 **清除訊息 ...**。清除佇列上的訊息。
 - h. 用滑鼠右鍵按一下佇列，然後選取 **刪除 ...**。即會顯示確認畫面，按一下 **確定**。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 390 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 在 MSCS 上輪詢 looksAlive 及 isAlive

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](#)。

Windows 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。

如果要移除佇列管理程式使用 TSL 連線, 您必須使用 IBM 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 TLS 金鑰儲存庫檔。

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 396 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 396 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 或 Role (如果您是使用 Windows 2012), 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 IBM 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\QMname 和 E:\WebSphere MQ\qmgrs\QMname 之類的路徑。請勿嘗試手動移動檔案; 請使用作為 IBM MQ MSCS 支援的一部分提供的 **hamvmqm** 公用程式:

1. 對佇列檔和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 398 頁的『佇列管理程式日誌檔』](#), 以瞭解為何這很重要)。
2. 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。
3. 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。

4. 執行公用程式來移動佇列管理程式，如下所示：

```
hamvmqm /m qmname /dd " c:\
IBM MQ " /ld "c:\
IBM MQ \log"
```

將您的佇列管理程式名稱替換為 *qmname*，將您的本端磁碟機代號替換為 *c*，並將您選擇的目錄替換為 *IBM MQ* (如果目錄尚未存在，則會建立目錄)。

5. 測試佇列管理程式以確保其運作 (如 [第 390 頁的『將佇列管理程式移至 MSCS 儲存體』](#) 中所述)。

Windows 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 IBM MQ 支援。

本節包含一些一般資訊，可協助您有效地使用 MSCS 的 IBM 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，以及在失效之前確定的工作量。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。

Windows 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 389 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。

Windows 手動啟動及 MSCS

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 IBM MQ MSCS 支援可以重新啟動「MQSeries 服務」，而無需立即啟動佇列管理程式。

IBM MQ MSCS 支援必須能夠重新啟動服務，它才能執行監視及控制，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 390 頁的『將佇列管理程式移至 MSCS 儲存體』](#)。

Windows 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

若要使用叢集作業來使用 IBM MQ，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 390 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；IBM 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 399 頁的『支援 MSCS 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

您可以將現存佇列管理程式置於 MSCS 控制下，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 387 頁的圖 74](#))。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 389 頁的『建立佇列管理程式以與 MSCS 搭配使用』](#)
- [第 390 頁的『將佇列管理程式移至 MSCS 儲存體』](#)
- [第 391 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案移至共用磁碟 (如需範例，請參閱 [第 390 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

IBM 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 控制下。

Windows 一律使用 MSCS 來管理叢集

請勿嘗試使用控制指令或 IBM MQ Explorer，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。

Windows 在 MSCS 中以主動/主動模式工作

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 IBM MQ 與 MSCS 及 COM+：

- 使用 IBM MQ 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。

Windows MSCS 中的 *PostOnline* 指令和 *PreOffline* 指令

使用這些指令來整合 IBM MQ MSCS 支援與其他系統。您可以使用它們來發出 IBM MQ 指令，但會有一些限制。

請在「參數」中指定 IBM MQ MSCS 類型的資源。您可以使用它們來整合 IBM 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 IBM 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\SysWOW64` 目錄，而不是 `\Windows\system32` 目錄。如需相關資訊，請參閱 Microsoft Windows x64 環境中檔案重新導向的相關文件。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；並以非同步方式呼叫；IBM 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 IBM 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。

如果您要執行相依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，並使它們相依於佇列管理程式資源。

Windows 在 MSCS 中使用偏好的節點

在 MSCS 中使用「主動/主動」模式為每一個佇列管理程式配置偏好節點時，它可能很有用。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。

Windows 在 MSCS 上安裝時發生 COM + 錯誤

當您在新安裝的 MSCS 叢集上安裝 IBM MQ 時，可能會發現「應用程式事件」日誌中報告「來源 COM +」及「事件 ID 4691」的錯誤。

這表示當「Microsoft 分散式交易協調程式 (MSDTC)」尚未配置成在 Microsoft Cluster Server (MSCS) 環境中執行時，您正在嘗試執行 IBM MQ。如需在叢集環境中配置 MSDTC 的相關資訊，請參閱 Microsoft 文件。

Windows 支援 MSCS 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 MQ 支援清單。

IBM 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 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 396 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 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 390 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。使用「IBM MQ 檔案總管」，在目前所在的節點上刪除它。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 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。

Multi

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相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。

多重實例佇列管理程式配置範例

第 401 頁的圖 75 顯示佇列管理程式 QM1 的多重實例配置範例。IBM 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。

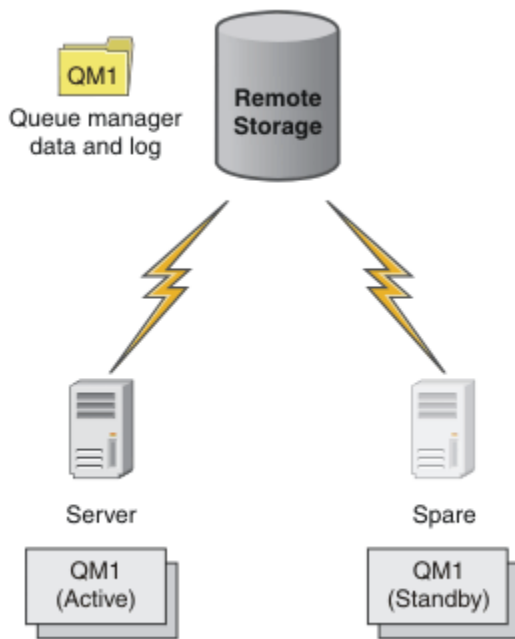


圖 75: 多重實例佇列管理程式

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 **admqinf** 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。
- IBM MQ 偵測到佇列管理程式處理程序的無回應性，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。

建置高可用性解決方案所需的其他元件

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。

- 用戶端及通道會重新連線，以將 IBM 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。尤其是您需要密切注意交易式應用程式的行為，以及瀏覽 IBM 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，IBM MQ MQI clients 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。IBM MQ classes for Java 不支援自動重新連接用戶端。

支援的平台

您可以在 IBM WebSphere MQ 7.0.1 以及更新版本所支援的任何非 z/OS 平台上建立多重實例佇列管理程式。

IBM WebSphere MQ 7.0.1 以及更新版本支援 MQI 用戶端的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 MQ。多重實例佇列管理程式會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，您需要讓 *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 426 頁的『保護 Windows 上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 404 頁的『在 Windows 上的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 418 頁的『在 Windows 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 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以相容的修正程式層次安裝相同版本的 IBM MQ。指令 **dspmqlnf** 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 `mqs.ini` 和 `qm.ini` 檔案。主題第 437 頁的『在 Linux 上建立多重實例佇列管理程式』、第 404 頁的『在 Windows 上的網域工作站或伺服器上建立多重實例佇列管理程式』及第 418 頁的『在 Windows 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，您可以將單一 `mqs.ini` 檔案放在網路共用上，並將 **AMQ_MQS_INI_LOCATION** 環境變數設為指向它，以共用單一檔案。


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 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 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。 例如，在 Linux 伺服器上設定的網路共用可以包含 UNIX and Linux 佇列管理程式的個別佇列管理程式資料及日誌，但不能包含 IBM i 使用的佇列管理程式資料。

 只要 IBM i 和 UNIX 系統的共用不同，您可以在相同的網路儲存體上建立多個共用。您可以為不同的擁有者提供不同的共用。此限制是 UNIX 與 IBM i 之間用於 IBM MQ 使用者和群組的不同名稱的結果。使用者和群組可以具有相同的 `uid` 和 `gid` 這一事實並不會放寬限制。

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。
7.  z/OS 不支援多重實例佇列管理程式。使用佇列共用群組。

可重新連接的用戶端會使用 z/OS 佇列管理程式。

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 429 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 426 頁的『保護 Windows 上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；如果您想要直接在 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 404 頁的『在 Windows 上的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 IBM WebSphere MQ 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。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。在 `crtmqm` 指令中，會在 `-a` 參數中傳遞其他安全群組。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。

如需建立多重實例佇列管理程式的詳細資料，請參閱第 404 頁的『在 Windows 上的網域工作站或伺服器上建立多重實例佇列管理程式』。

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和日誌目錄。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和資料檔案，請閱讀第 429 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。作業為第 430 頁的『讀取及寫入本端 mqm 群組授權的資料及日誌檔』和第 433 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 426 頁的『保護 Windows 上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 MQ for Windows，以及配置佇列管理程式以使用共用。作業如下：

1. 第 407 頁的『在 Windows 上建立 Active Directory 及 DNS 網域』。
2. 第 410 頁的『在 Windows 網域中的伺服器或工作站上安裝 IBM MQ』。
3. 第 412 頁的『在 Windows 上建立佇列管理程式資料及日誌檔的共用目錄』。
4. 第 415 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 404 頁的『在 Windows 上的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

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 426 頁的『保護 Windows 上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 MQ for Windows，則佇列管理程式資料及日誌檔會受到網域 mqm 群組的安全保護，該群組可以共用。遵循作業第 418 頁的『在 Windows 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

管理授權和存取控制

如何使用 Windows Server 叢集節點作為網域控制站

Windows 在 Windows 上的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 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 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 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 407 頁的『在 Windows 上建立 Active Directory 及 DNS 網域』](#)。
2. [第 410 頁的『在 Windows 網域中的伺服器或工作站上安裝 IBM MQ』](#)。
3. [第 412 頁的『在 Windows 上建立佇列管理程式資料及日誌檔的共用目錄』](#)。
4. [第 415 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 MQ for Windows。
 - a) 執行作業 [第 410 頁的『在 Windows 網域中的伺服器或工作站上安裝 IBM 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 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 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 415 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:\ProgramData\IBM\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** 參數重新執行 **dspmqlinf** 指令。

```
dspmqlinf -o command QMGR
```

系統回應：

```
addmqinf -s QueueManager -v Name=QMGR
-v Directory=QMGR -v Prefix="C:\ProgramData\IBM\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:\ProgramData\IBM\MQ"
-v DataPath=\\sun\wmq\data\QMGR
```

IBM 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 **dspmqr** 指令：

```
dspmqr -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>dspmqr -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：

```
IBM 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'.
IBM 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：

```
IBM 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：

```
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ending.  
IBM 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>dspmq  
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 424 頁的『在 Windows 上驗證多重實例佇列管理程式』](#)。

Windows 在 *Windows* 上建立 *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 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 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 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 403 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 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 MQ 支援的最舊 Windows Server 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 MQ"」精靈的行為。它會導致 "準備 IBM MQ" 精靈自動將群組 Domain mqm 新增至網域中 IBM MQ 的每一個新安裝上的本端 mqm 群組。
 - 可以在不含任何 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, 則必須使用與 Domain mqm 群組相同的內容來定義群組。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 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 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。
 - 您在網域中的工作站或伺服器上安裝 IBM MQ 時指定的網域使用者必須隸屬於 Domain mqm 群組, 或者隸屬於使用與 Domain mqm 群組相同的內容來定義的替代群組。
- 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**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* 會執行 IBM MQ 服務, 以互動方式使用另一個使用者 *wmquser2*。

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 MQ"」精靈會自動將 Domain mqm 配置成本端 mqm 群組的成員, 只要在其中執行精靈一次。

您必須提供不同的使用者, 才能在單一電腦上針對每一個 IBM MQ 安裝執行 IBM 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 MQ。

如果您需要以服務方式執行 IBM MQ, 然後為網域使用者 (從網域管理者取得) 提供以服務方式執行的存取權, 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 Server 2008 及 Windows Server 2012 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 410 頁的『在 Windows 網域中的伺服器或工作站上安裝 IBM MQ』](#)。

相關工作

Windows [在 Windows 網域中的伺服器或工作站上安裝 IBM MQ](#)

Windows [在 Windows 上建立佇列管理程式資料及日誌檔的共用目錄](#)

Windows [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](#)

Windows [在 Windows 網域中的伺服器或工作站上安裝 IBM 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 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 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 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 407 頁的『在 Windows 上建立 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 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，您將 Windows Server 2008 (稱為 *mars*) 配置為 *wmq.example.com* 網域的成員。您安裝 IBM 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 403 頁的『Windows 網域及多重實例佇列管理程式』](#)。

基於作業的目的，網域控制站主機名為 *sun*，且兩部 IBM 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

- 將網域控制站 *sun.wmq.example.com* 新增至 *mars* 作為 DNS 伺服器。
 - 在 *mars* 上，以 *mars\Administrator* 身分登入，然後按一下 **開始**。
 - 用滑鼠右鍵按一下 **網路 > 內容 > 管理網路連線**。
 - 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。
 - 從「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。
 - 在 DNS 伺服器位址下，按一下 **新增 ...**。
 - 鍵入網域控制站 (也是 DNS 伺服器) 的 IP 位址，然後按一下 **新增**。
 - 按一下 **附加這些 DNS 字尾 > 新增 ...**。
 - 鍵入 *wmq.example.com*，然後按一下 **新增**。
 - 在 **此連線的 DNS 字尾** 欄位中鍵入 *wmq.example.com*。
 - 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**。
 - 開啟指令視窗，並鍵入指令 **ipconfig /all**，以檢閱 TCP/IP 設定。
- 在 *mars* 上，將電腦新增至 *wmq.example.com* 網域。
 - 按一下 **開始**
 - 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**。
 - 在「系統內容」視窗中，按一下 **變更 ...**。
 - 按一下「網域」，鍵入 *wmq.example.com*，然後按一下 **確定**。
 - 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**。
 - 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。
- 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*。
- 在 *mars* 上安裝 IBM MQ for Windows。

如需執行 IBM MQ for Windows 安裝精靈的進一步資訊; 請參閱 [在 Windows 上安裝 IBM MQ 伺服器](#)。

 - 在 *mars* 上，以本端管理者 *mars\Administrator* 身分登入。
 - 在 IBM MQ for Windows 安裝媒體上執行 **Setup** 指令。
即會啟動「IBM 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 407 頁的『在 Windows 上建立 Active Directory 及 DNS 網域』](#) 會配置這組作業的網域使用者 ID。
 - 按一下 **IBM MQ 安裝**，選取安裝語言，然後按一下「**啟動 IBM 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 MQ，且安裝程式會啟動 "準備 IBM MQ" 精靈。
重要: 尚未執行精靈。
- 配置將使用 **以服務方式執行** 權限來執行 IBM 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 MQ 服務的使用者。在此範例中，您為使用者提供權限。

- 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**。
 - c) 按一下 **新增使用者或群組 ...** 並鍵入 *wmquser1*，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**。關閉「本機安全性原則」視窗。
6. 執行 "準備 IBM MQ" 精靈。

如需執行「準備 IBM MQ」精靈的進一步資訊；請參閱 [使用「準備 IBM MQ」精靈來配置 IBM MQ](#)。

- a) IBM MQ Installer 會自動執行 "Prepare IBM MQ"。

若要手動啟動精靈，請在 **啟動 > 所有程式 > IBM MQ** 資料夾中尋找 "準備 IBM MQ" 的捷徑。選取對應於在多重安裝配置中安裝 IBM MQ 的捷徑。

- b) 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。
- c) 在第一個「為 Windows 網域使用者配置 IBM MQ for Windows」視窗中，按 **是 > 下一步**。
- d) 在第二個「為 Windows 網域使用者配置 IBM 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 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 430 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 412 頁的『在 Windows 上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。

相關工作

Windows [在 Windows 上建立 Active Directory 及 DNS 網域](#)

Windows [在 Windows 上建立佇列管理程式資料及日誌檔的共用目錄](#)

Windows [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](#)

相關參考

[IBM MQ Windows 服務所需的使用者權限](#)

Windows [在 Windows 上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 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 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 407 頁的『在 Windows 上建立 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 403 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至其 *-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 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 MQ 伺服器上本端 *mqm* 群組的成員。因此，*wmquser1* 和 *wmquser2* 都是兩個 IBM 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。按一下「Windows 安全」視窗中的 **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 MQ 伺服器讀取及寫入檔案至共用目錄。檢查 IBM 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 *wmquser1* ; *wmquser2*，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\wmquser1* 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 *wmquser1* 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\wmquser2* 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 415 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』](#)。

相關工作

Windows 在 Windows 上建立 Active Directory 及 DNS 網域

Windows 在 Windows 網域中的伺服器或工作站上安裝 IBM MQ

Windows 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

Windows 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和資料檔案。

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 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 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 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 407 頁的『在 Windows 上建立 Active Directory 及 DNS 網域』](#)。
2. [第 410 頁的『在 Windows 網域中的伺服器或工作站上安裝 IBM MQ』](#)。
3. [第 412 頁的『在 Windows 上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 403 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`。
2. 開啟指令視窗。
3. 重新啟動 IBM 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. 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：

```
IBM 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。

1. 啟動佇列管理程式。

```
strmqm QMGR
```

系統回應：

```
IBM 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'.  
IBM 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: IBM 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:

```
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ended.
```

6. 刪除佇列管理程式。

```
dltmqm QMGR
```

系統回應:

```
IBM 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
```

相關工作

Windows [在 Windows 上建立 Active Directory 及 DNS 網域](#)

Windows [在 Windows 網域中的伺服器或工作站上安裝 IBM MQ](#)

Windows [在 Windows 上建立佇列管理程式資料及日誌檔的共用目錄](#)

Windows 在 Windows 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 404 頁的『在 Windows 上的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 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 407 頁的『在 Windows 上建立 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 421 頁的『將第二個 Windows 網域控制站新增至範例網域』](#) 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 MQ for Windows，並驗證安裝。
4. 執行 [第 422 頁的『在網域範例中的 Windows 網域控制站上安裝 IBM MQ』](#) 中的步驟，以在兩個網域控制站上安裝 IBM 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。按一下「Windows 安全」視窗中的 **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：

```
IBM 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：

```
IBM 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'.
IBM 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 **dspmqlinf** 指令：

```
dspmqlinf QMGR
```

系統回應：

```
QueueManager:
Name=QMGR
```

```
Directory=QMGR
Prefix=C:\ProgramData\IBM\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` 參數重新執行 **dspmqlinf** 指令。

```
dspmqlinf -o command QMGR
```

系統回應：

```
addmqinf -s QueueManager -v Name=QMGR
-v Directory=QMGR -v Prefix="C:\ProgramData\IBM\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
```

IBM 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：

```
IBM 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：

```
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ending.
IBM 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

若要使用範例程式來驗證多重實例佇列管理程式，請參閱 [第 424 頁的『在 Windows 上驗證多重實例佇列管理程式』](#)。

相關工作

[第 421 頁的『將第二個 Windows 網域控制站新增至範例網域』](#)

[第 422 頁的『在網域範例中的 Windows 網域控制站上安裝 IBM 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](#)

Windows 將第二個 Windows 網域控制站新增至範例網域

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 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 407 頁的『在 Windows 上建立 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 403 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「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 **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 MQ；請參閱第 422 頁的『在網域範例中的 Windows 網域控制站上安裝 IBM MQ』。

相關工作

Windows 在網域範例中的 Windows 網域控制站上安裝 IBM MQ
第 407 頁的『在 Windows 上建立 Active Directory 及 DNS 網域』

Windows 在網域範例中的 Windows 網域控制站上安裝 IBM MQ
在 *wmq.example.com* 網域中的兩個網域控制站上安裝及配置 IBM MQ 的安裝。
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 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 407 頁的『在 Windows 上建立 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 421 頁的『將第二個 Windows 網域控制站新增至範例網域』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 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 MQ 的安裝。

程序

1. 在 *sun* 和 *earth* 上安裝 IBM MQ。

如需執行 IBM MQ for Windows 安裝精靈的進一步資訊，請參閱 [在 Windows 上安裝 IBM 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 MQ for Windows 安裝媒體上執行 **Setup** 指令。

即會啟動「IBM 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 **IBM MQ 安裝**，選取安裝語言，然後按一下「啟動 IBM 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 MQ，且安裝程式會啟動 "準備 IBM MQ" 精靈。

IBM 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 MQ" 精靈。

如需執行「"準備 IBM MQ"」精靈的進一步相關資訊，請參閱 [使用「準備 IBM MQ」精靈來配置 IBM MQ](#)。

- a) IBM MQ 安裝程式會自動執行 "準備 IBM MQ"。

若要手動啟動精靈，請在 **啟動 > 所有程式 > IBM MQ** 資料夾中尋找 "準備 IBM MQ" 的捷徑。選取對應於在多重安裝配置中安裝 IBM MQ 的捷徑。

- b) 按 **下一步** 並按一下 **否**，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。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**。

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 MQ 更簡單，就像它不在網域中一樣。

"準備 IBM 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 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 ...** 及類型 mqm；按一下 **檢查名稱 > 確定 > 確定**

結果

1. 檢查 "準備 IBM 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. 按一下按 **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**。

下一步

1. 執行作業 [第 430 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 418 頁的『在 Windows 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。

相關工作

Windows 將第二個 Windows 網域控制站新增至範例網域

相關參考

IBM MQ Windows 服務所需的使用者權限

Windows 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。本主題提供範例配置，以驗證 Windows Server 2003 上的多重實例佇列管理程式配置。

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 418 頁的『在 Windows 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 7 Service Pack 1 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 IBM MQ Explorer 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 IBM MQ Explorer。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。



小心: 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) +
CONNNAME(' 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** 參數設為您已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 IBM MQ Explorer。

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 amqsgnac TARGET QM1
start amqsmnac -s SOURCE -t TARGET -m QM1
start amqspnac 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 IBM MQ Explorer，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，則需要大約 7 分鐘才能重新建立所有三個連線。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
```

Windows 保護 Windows 上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 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。它會使用本端 mqm 群組來保護目錄，並配置 IBM 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 427 頁的圖 76。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。名稱在 `qm.ini` 檔案的 **AccessMode** 段落中以 `SecurityGroup` 形式傳送；例如：

```
AccessMode:
SecurityGroup=wmq\wmq
```

`qm.ini` 中的 **AccessMode** 段落也包括 `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。

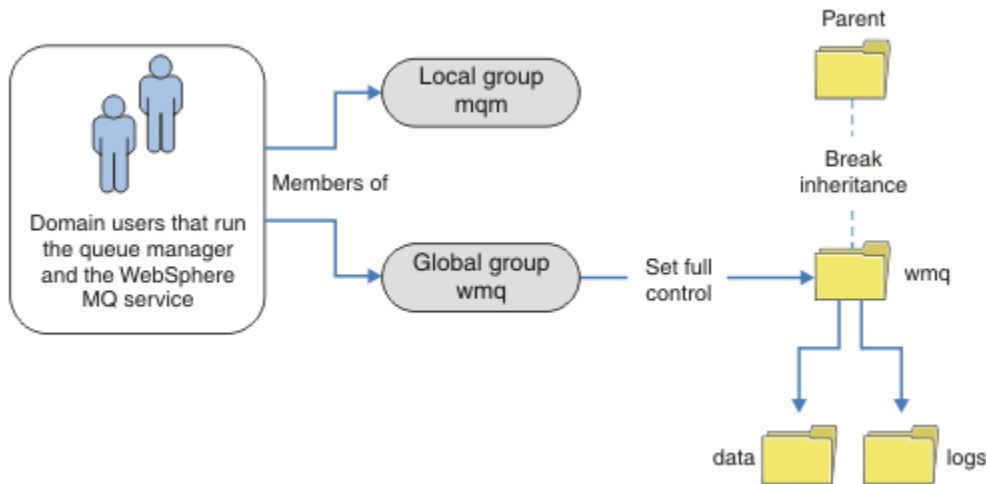


圖 76: 使用替代廣域安全群組來保護佇列管理程式資料及日誌的安全 (1)

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 427 頁的圖 76 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 MQ 伺服器，則第 427 頁的圖 76 中的使用者分組並不方便。您需要針對每一部 IBM 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 MQ 群組成員的使用者成為 `Domain mqm` 群組的使用者；請參閱第 428 頁的圖 77。當您將 IBM MQ 安裝為網域安裝時，"準備 IBM 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 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 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。

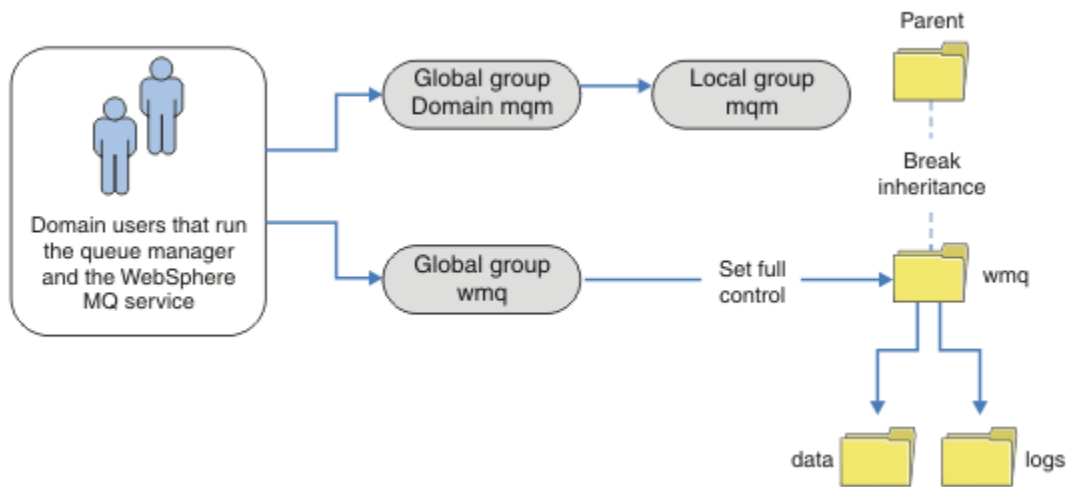


圖 77: 使用替代廣域安全群組來保護佇列管理程式資料及日誌的安全 (2)

第 428 頁的圖 77 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 428 頁的圖 78。

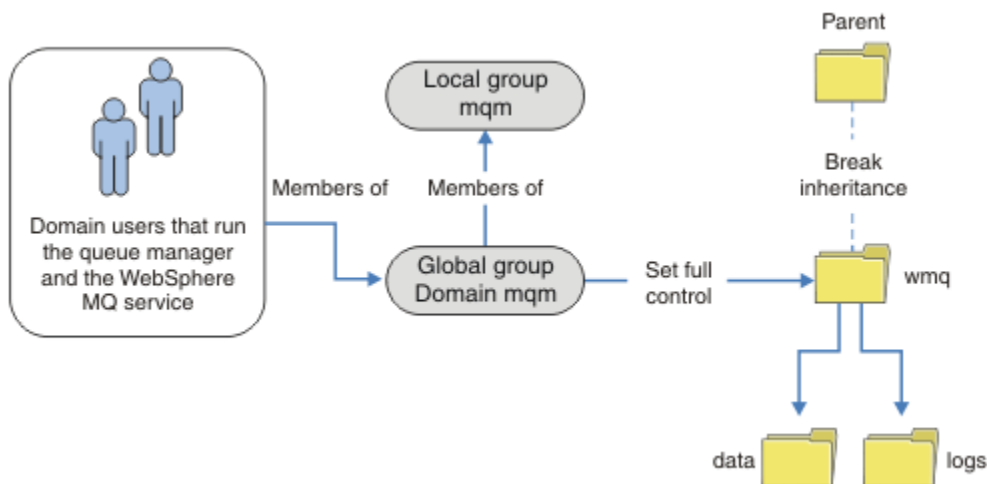


圖 78: 使用替代廣域安全群組來保護佇列管理程式資料及日誌的安全 (3)

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 429 頁的圖 79。在第 429 頁的圖 79 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 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。配置無法利用 IBM MQ 安裝架構將 Domain mqm 傳播至本端 mqm 群組 (如 第 428 頁的圖 78 中所示)，因為有兩個使用者群組。

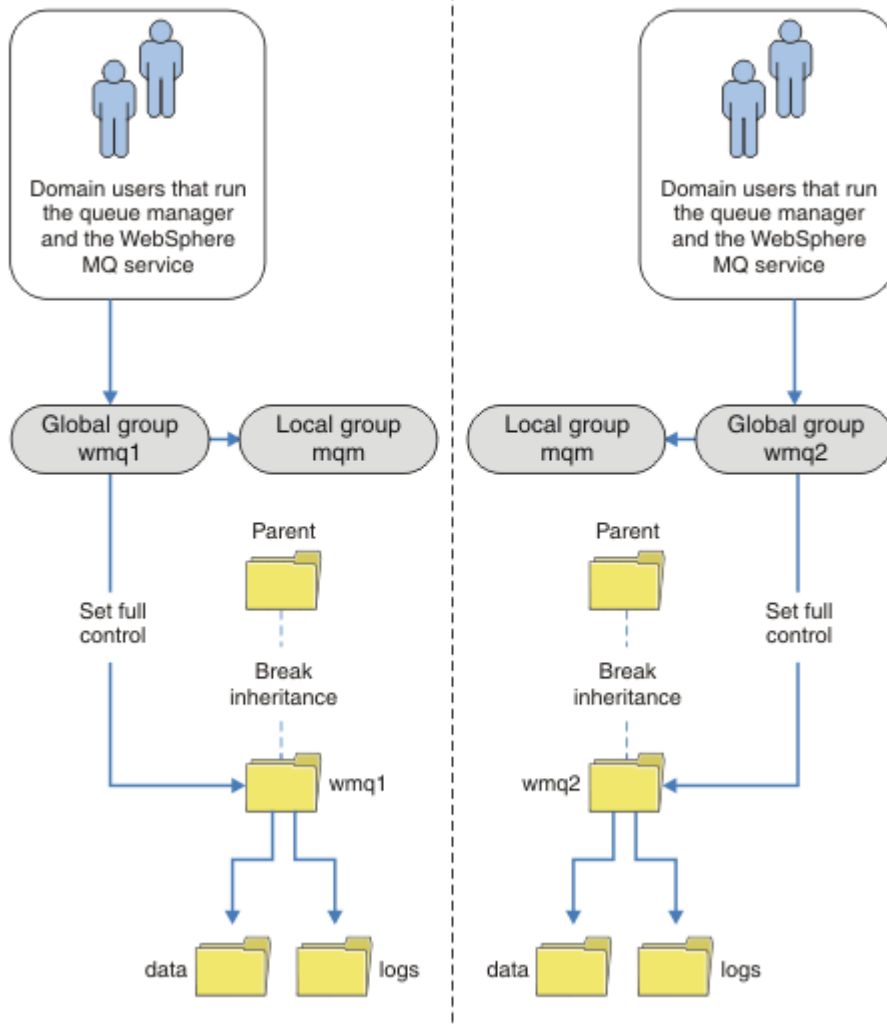


圖 79: 使用替代廣域安全主體來保護佇列管理程式資料及日誌的安全 (4)

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 428 頁的圖 78](#) 中顯示的較簡單模型。

Windows 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 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。它會使用本端 mqm 群組來保護目錄，並配置 IBM MQ 服務的使用者 ID 以存取目錄。

兩個範例示範如何配置 IBM 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 430 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 433 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。當目錄由在不同伺服器 (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 426 頁的『保護 Windows 上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 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 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 430 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 430 頁的圖 80。本端群組 `wmq` 是替代安全群組的範例。

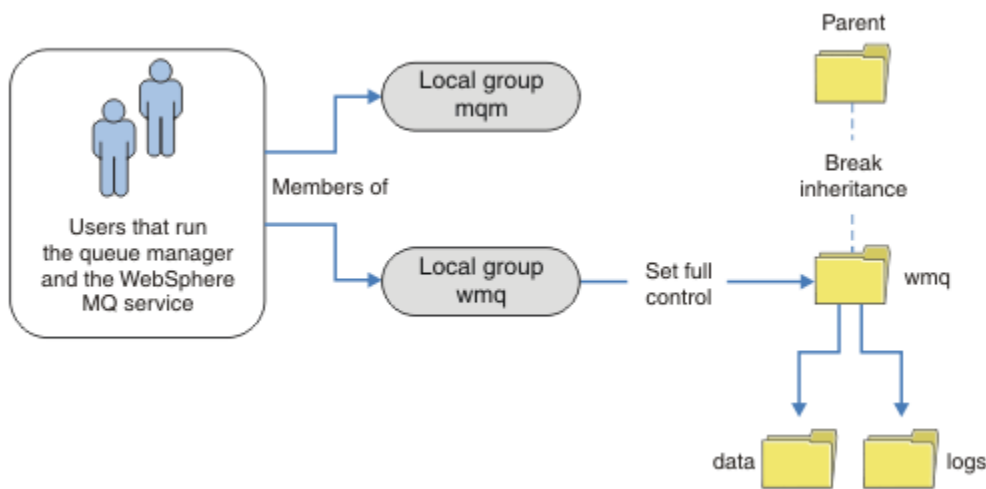


圖 80: 使用替代本端安全群組 `wmq` 來保護佇列管理程式資料及日誌的安全

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 426 頁的『保護 Windows 上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。

Windows 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 MQ for Windows 作為主要安裝。
2. 執行 "準備 IBM 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 403 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 403 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 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 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 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 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：

```
IBM 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:\umq` 目錄中。

```
dir c:\umq/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:\umq\*.*
```

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：

```
IBM 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'.  
IBM 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: IBM 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 AMQSGETO 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：

```
IBM MQ queue manager 'QMGR' ending.  
IBM MQ queue manager 'QMGR' ended.
```

6. 刪除佇列管理程式。

```
dltmqm QMGR
```

系統回應：

```
IBM 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 403 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。

相關工作

Windows 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 415 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 404 頁的『在 Windows 上的網域工作站或伺服器上建立多重實例佇列管理程式』](#)

Windows 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 MQ for Windows 作為主要安裝。
2. 執行 "準備 IBM 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 403 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 403 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 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，您配置本端群組以熟悉替代安全群組的使用，而無需安裝 IBM 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 workstation 使用 IBM 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 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 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 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 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 MQ 作為網域的一部分，並且要在第 404 頁的『在 Windows 上的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 407 頁的『在 Windows 上建立 Active Directory 及 DNS 網域』中建立的網域使用者 ID 是 `wmuser1` 及 `wmuser2`。

如果您未安裝佇列管理程式作為網域的一部分，則執行 IBM MQ 服務的預設本端使用者 ID 是 `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 `wmuser1` 和 `wmuser2` 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 `wmuser1 ; wmuser2`，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 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 altmqm: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 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 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 altmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE -md c:\wmq\data -ld c:\wmq\logs QMGR
```

系統回應：

```
IBM 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：

```
IBM 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'.  
IBM 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: IBM 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 AMQSGETO 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：

```
IBM MQ queue manager 'QMGR' ending.
IBM MQ queue manager 'QMGR' ended.
```

6. 刪除佇列管理程式。

```
dltmqm QMGR
```

系統回應：

```
IBM 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
```

相關工作

Windows 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。

Linux 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。

關於這項作業

範例設定在 2 GHz 筆記型電腦上，具有執行 Windows 7 Service Pack 1 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 7 Service Pack 1 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。

程序

1. 以 `root` 身分登入。
2. 閱讀 [安裝 IBM MQ -概觀](#)，並遵循適當的鏈結以安裝 IBM MQ、建立 `mqm` 使用者和群組，以及定義 `/var/mqm`。
3. 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。
4. 對於 `Server1`，請完成下列步驟：
 - a. 在要共用的共用資料夾 `/MQHA` 中建立日誌和資料目錄。例如：
 - i) `mkdir /MQHA`
 - ii) `mkdir /MQHA/logs`
 - iii) `mkdir /MQHA/qmgrs`
5. 對於 `Server2`，請完成下列步驟：
 - a. 建立資料夾 `/MQHA`，以裝載共用檔案系統。保持路徑與 `Server1` 上的路徑相同。例如：
 - i) `mkdir /MQHA`
6. 確保使用者和群組 `mqm` 擁有 `MQHA` 目錄，並且使用者和群組的存取權設為 `rw`。例如，`ls -al` 會顯示 `drwxrwxr-x mqm mqm 4096 Nov 27 14:38 MQDATA`。
 - a. `chown -R mqm:mqm /MQHA`
 - b. `chmod -R ug+rw /MQHA`
7. 輸入下列指令來建立佇列管理程式：`crtmqm -ld /MQHA/logs -md /MQHA/qmgrs QM1`
8. 新增 `^/MQHA *(rw, sync, no_wdelay, fsid=0)` 至 `/etc/exports`
9. 對於 `Server1`，請完成下列步驟：
 - a. 啟動 NFS 常駐程式：`/etc/init.d/ nfs start`
 - b. 從 `Server1`: 複製佇列管理程式配置詳細資料

```
dspmqlnf -o command QM1
```

並將結果複製到剪貼簿:

```
addmqinf -s QueueManager
-v Name=QM1
-v Directory=QM1
-v Prefix=/var/mqm
-v DataPath=/MQHA/qmgrs/QM1
```

10. 對於 `Server2`，請完成下列步驟：
 - a. 輸入下列指令來裝載匯出的檔案系統 `/MQHA:mount -t nfs4 -o hard,intr Server1:/ /MQHA`
 - b. 將佇列管理程式配置指令貼到 `Server2`:

```
addmqinf -s QueueManager
-v Name=QM1
-v Directory=QM1
```

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。

```
-v Prefix=/var/mqm
-v DataPath=/MQHA/qmgrs/QM1
```

11. 使用 **-x** 參數，以任一順序啟動佇列管理程式實例: **strmqm -x QM1**。

用來啟動佇列管理程式實例的指令必須從與 **addmqinf** 指令相同的 IBM 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](#)。

Linux 在 Linux 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 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 437 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 7 Service Pack 1 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 IBM MQ Explorer 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 IBM MQ Explorer。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 IBM MQ Explorer。

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 IBM MQ Explorer，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，則需要大約 7 分鐘才能重新建立所有三個連線。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
```

Multi 刪除多重實例佇列管理程式

在 Multiplatforms 上，若要完全刪除多重實例佇列管理程式，您可以使用 **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。

Linux UNIX

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
```

```
dltmqm QM1
```

Multi 啟動和停止多重實例佇列管理程式

啟動及停止在 Multiplatforms 上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 IBM 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。

Multi 共用檔案系統

在 Multitplatforms 上，多重實例佇列管理程式會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 IBM 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進行回復。您可以使用備份佇列管理程式來管理此處理程序。

Multi 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。

Multi 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。IBM MQ classes for Java 不支援自動重新連接用戶端。

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是否已中斷與失敗實例的連線。

本端連接的應用程式必須從失敗的佇列管理程式實例結束或中斷連線，才能重新啟動失敗的實例。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。

Multi 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 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。IBM MQ classes for Java 不支援自動重新連接用戶端。

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。IBM 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 TCP/IP 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。用戶端需要編譯才能與 IBM WebSphere MQ 7.0.1 用戶端程式庫一起執行。它們必須至少連接至一個 IBM WebSphere MQ 7.0.1 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。

Multi 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 IBM MQ MQI client 與佇列管理程式之間共用且儲存在佇列管理程式中的環境定義。IBM MQ MQI client 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 IBM 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。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。IBM MQ classes for Java 不支援自動重新連接用戶端。

使用 IBM MQ MQI client 應用程式，您可以設定連線選項，以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 IBM MQ 伺服器程式。

您可以使用兩種方式來自動執行 IBM 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. IBM MQ MQI client 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入另一個佇列：如果它透過重新啟動工作單元，在同步點內處理來自 MQPUT、MQGET 或 MQCMIT 的 MQRC_BACKED_OUT 原因碼，則不會遺失任何訊息。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 445 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 IBM MQ MQI client 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 IBM 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 IBM 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 IBM MQ MQI client 的範例 [第 444 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。您需要知道重新連線如何變更 IBM MQ MQI client 的狀態，以便設計用戶端應用程式以在佇列管理程式失效接手之後存活。

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 IBM 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。

IBM MQ MQI client 環境知道失效接手本身，並在重新連線之後盡可能還原環境定義，方法是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 IBM MQ 狀態。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 IBM MQ 的連線並重新連接，甚至完全停止程式。若要改善連續性，您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 IBM MQ MQI client 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 重新連線。在 MQCMIT 期間重新連線的行為類似。
- 如果 IBM MQ MQI client 未從佇列管理程式收到任何回應來指出成功或失敗，則會傳回 MQRC_CALL_INTERRUPTED (在成功重新連接之後)。
 - 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，IBM MQ MQI client 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。

Multi 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 IBM 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，兩種類型的 IBM MQ 目錄和檔案之間有重要的區別。

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及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。

- IBM 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 IBM MQ (例如 Windows、UNIX and Linux 系統上的 mq.ini 檔) 或您自己的配置管理 (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 IBM 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 MQ 可用性解決方案

應用程式正在使用其他 IBM 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 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

叢集提供不間斷的服務。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

z/OS 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，且只能在更廣泛的傳訊架構中補充佇列共用群組。

IBM MQ 用戶端增加應用程式可用性

根據佇列管理程式可用性、連線加權及親緣性，IBM MQ MQI client 程式可以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。IBM MQ classes for Java 不支援自動重新連接用戶端。

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。

MQ Adv. Linux V9.1.0 RDQM 高可用性

RDQM (抄寫的資料佇列管理程式) 是可在 Linux 平台上使用的高可用性解決方案。

RDQM 配置由高可用性 (HA) 群組中所配置的三部伺服器組成，每一部伺服器都有一個佇列管理程式實例。其中一個實例是執行中的佇列管理程式，它會同步將其資料抄寫至另外兩個實例。如果執行此佇列管理程式的伺服器失敗，則另一個佇列管理程式實例會啟動並具有現行資料可操作。佇列管理程式的三個實例會共用浮動 IP 位址，因此用戶端只需要配置單一 IP 位址。一次只能執行佇列管理程式的一個實例，即使高可用性群組因網路問題而分割也一樣。執行佇列管理程式的伺服器稱為「主要」，其他兩部伺服器則稱為「次要」。

有三個節點可用來大幅減少出現核心分裂狀況的可能性。在雙節點高可用性系統中，當兩個節點之間的連線功能中斷時，可能會發生核心分裂。由於沒有連線功能，兩個節點可以同時執行佇列管理程式，累計不同的資料。當連線還原時，資料有兩個不同版本 ('spit-brain')，需要人為介入來決定要保留哪些資料集，以及要捨棄哪些資料集。

RDQM 使用具有額定的三節點系統來避免核心分裂狀況。可與至少一個其他節點通訊的節點會形成仲裁。佇列管理程式只能在具有仲裁的節點上執行。佇列管理程式無法在未連接至少一個其他節點的節點上執行，因此永遠無法同時在兩個節點上執行：

- 如果單一節點失敗，佇列管理程式可以在其他兩個節點的其中一個節點上執行。如果兩個節點失敗，則佇列管理程式無法在其餘節點上執行，因為節點沒有仲裁 (其餘節點無法指出其他兩個節點是否失敗，或它們仍在執行中且失去連線功能)。
- 如果單一節點失去連線功能，則佇列管理程式無法在此節點上執行，因為節點沒有仲裁。佇列管理程式可以在其餘兩個節點的其中一個節點上執行，這些節點確實具有仲裁。如果所有節點都失去連線功能，則佇列管理程式無法在任何節點上執行，因為沒有任何節點具有仲裁。

註: IBM MQ Console 不支援抄寫的資料佇列管理程式。您可以搭配使用 IBM MQ Explorer 與抄寫的資料佇列管理程式，但這不會顯示 RDQM 特性特定的資訊。

這三個節點的群組配置由 Pacemaker 處理。三個節點之間的抄寫由 DRBD 處理。(如需 Pacemaker 的相關資訊，請參閱 <https://clusterlabs.org/pacemaker/>; 如需 DRBD 的相關資訊，請參閱 <https://docs.linbit.com/docs/users-guide-9.0/>。)

您可以使用第 525 頁的『備份佇列管理程式資料』中說明的處理程序來備份已抄寫的資料佇列管理程式。停止佇列管理程式並備份它不會影響 RDQM 配置所執行的節點監視。

下圖顯示在 HA 群組中三個節點的每一個節點上執行 RDQM 的一般部署。

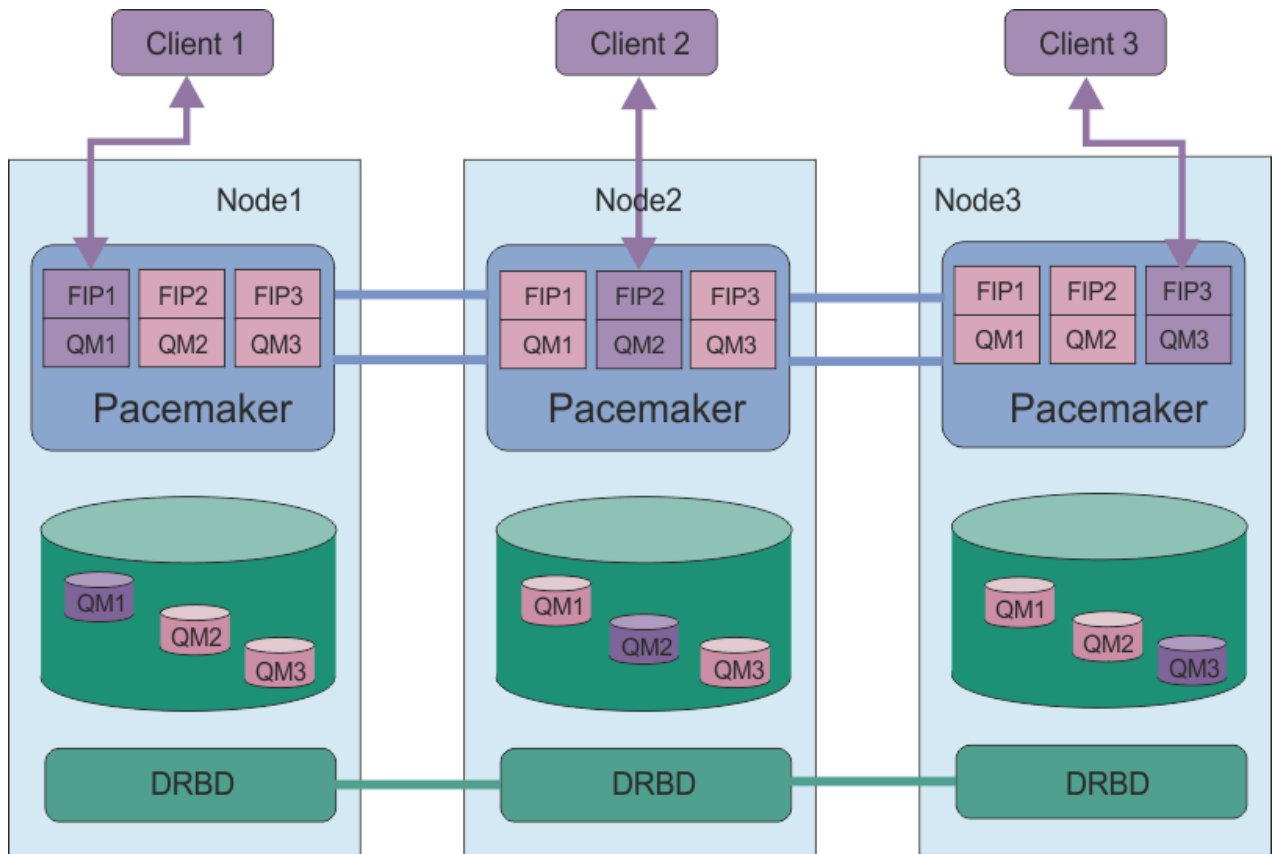


圖 81: 具有三個 RDQM 的 HA 群組範例

在下一個圖中，Node3 已失敗，Pacemaker 鏈結已遺失，而佇列管理程式 QM3 會改為在 Node2 上執行。

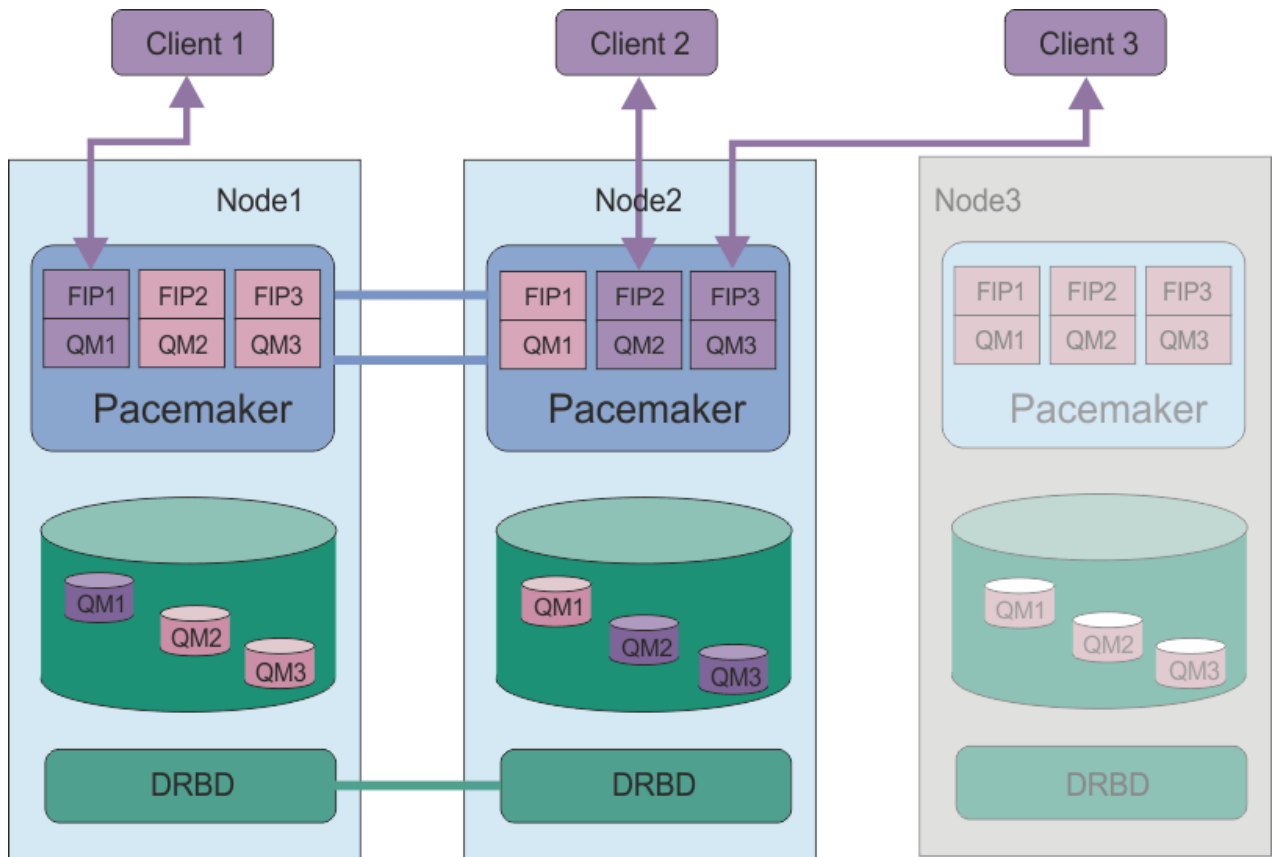


圖 82: node3 失敗之後的範例

註: 當佇列管理程式失效接手至另一個節點時，它們會保留失效接手時所處的狀態。已啟動執行中的佇列管理程式，已停止的佇列管理程式仍會停止。

相關工作

[安裝 RDQM \(抄寫的資料佇列管理程式\)](#)

[升級 RDQM \(抄寫的資料佇列管理程式\)](#)

[移轉抄寫的資料佇列管理程式](#)

Linux V 9.1.0 RDQM HA 解決方案的需求

在配置 RDQM 高可用性 (HA) 群組之前，您必須符合一些需求。

系統需求

在配置 RDQM HA 群組之前，您必須在要成為 HA 群組一部分的三部伺服器中的每一部伺服器上完成部分配置。

- 每一個節點都需要一個名為 drbdpool 的磁區群組。每一個抄寫的資料佇列管理程式的儲存體，會從這個磁區群組配置成每個佇列管理程式的個別邏輯磁區。為了取得最佳效能，此磁區群組應該由一或多個對應於內部磁碟機 (最好是 SSD) 的實體磁區組成。您可以在安裝 RDQM HA 解決方案之前或之後建立 drbdpool，但必須在實際建立任何 RDQM 之前先建立 drbdpool。使用 **vgs** 指令來檢查磁區群組配置。輸出應該與下列項目類似：

```
VG          #PV #LV #SN Attr   VSize  VFree
drbdpool   1   9   0 wz--n- <16.00g <7.00g
rhe1       1   2   0 wz--n- <15.00g  0
```

特別是，請檢查屬性的第六欄 (即 wz--nc) 中沒有 c 字元。c 指出已啟用叢集作業，如果已啟用，則您必須刪除磁區群組，並重建它而不使用叢集作業。

- 建立 drbdpool 磁區群組之後，對其不執行任何其他動作。IBM MQ 會管理在 drbdpool 中建立的邏輯磁區，以及它們的裝載方式和裝載位置。
- 每一個節點最多需要三個用於配置 RDQM 支援的介面：
 - Pacemaker 的主要介面，用來監視 HA 群組。
 - Pacemaker 用來監視 HA 群組的替代介面。
 - 同步資料抄寫的介面，稱為抄寫介面。考慮到 HA 群組中執行的所有抄寫資料佇列管理程式的預期工作量，這應該有足夠的頻寬來支援抄寫需求。

您可以配置 HA 群組，以便所有三個介面都使用相同的 IP 位址，每個介面使用個別的 IP 位址，或使用相同的 IP 位址作為抄寫介面的主要及替代及個別 IP 位址。

為了達到容錯上限，這些介面應該是獨立的「網路介面卡 (NIC)」。

- DRBD 要求 HA 群組中的每一個節點都具有有效的網際網路主機名稱 (uname -n 所傳回的值)，如 RFC 1123 修正的 RFC 952 所定義。
- 如果 HA 群組中的節點之間有防火牆，則防火牆必須容許埠範圍上節點之間的資料流量。提供範例 Script /opt/mqm/samp/rdqm/firewalld/configure.sh，如果您在 RHEL 中執行標準防火牆，則會開啟必要的埠。您必須以 root 身分執行 Script。如果您使用其他防火牆，請檢查服務定義 /usr/lib/firewalld/services/rdqm*，以查看需要開啟哪些埠。
- 如果系統在「強制」模式下使用 SELinux，您必須執行下列指令：

```
semanage permissive -a drbd_t
```

網路需求

建議您在相同資料中心的 RDQM HA 群組中找到三個節點。

如果您選擇尋找不同資料中心的節點，請注意下列限制：

- 效能會隨著資料中心之間的延遲增加而快速降低。雖然 IBM 將支援最多 5 毫秒的延遲，但您可能會發現應用程式效能無法容忍超過 1 到 2 毫秒的延遲。
- 透過抄寫鏈結傳送的資料不受限於任何其他加密，除了使用 IBM MQ AMS 可能已備妥的加密之外。

您可以配置浮動 IP 位址，讓用戶端能夠對抄寫的資料佇列管理程式 (RDQM) 使用相同的 IP 位址，而不論其執行所在 HA 群組中的哪個節點。浮動位址會連結至 RDQM 之主要節點上的具名實體介面。如果 RDQM 失效接手，且不同的節點變成主要節點，則浮動 IP 會連結至新主要節點上同名的介面。三個節點上的實體介面必須都具有相同的名稱，且屬於與浮動 IP 位址相同的子網路。

配置叢集的使用者需求

您可以 root 使用者身分來配置 RDQM HA 群組。如果您不想要配置為 root，請改為配置為 mqm 群組中的使用者。如果 mqm 使用者要配置 RDQM 叢集，您必須符合下列需求：

- mqm 使用者必須能夠使用 sudo，在組成 RDQM HA 群組的三部伺服器中的每一部伺服器上執行指令。
- 如果 mqm 使用者可以使用不含密碼的 SSH，在構成 RDQM HA 群組的三部伺服器中的每一部伺服器上執行指令，則使用者只需要在其中一部伺服器上執行指令。
- 如果您為 mqm 使用者配置無密碼 SSH，則該使用者在所有三部伺服器上必須具有相同的 UID。

您必須配置 sudo，以便 mqm 使用者可以使用 root 權限執行下列指令：

```
/opt/mqm/bin/crtmqm
/opt/mqm/bin/dltmqm
/opt/mqm/bin/rdqmadm
/opt/mqm/bin/rdqmstatus
```

使用佇列管理程式的使用者需求

若要建立、刪除或配置抄寫的資料佇列管理程式 (RDQ)，您必須使用同時屬於 mqm 及 haclient 群組 (haclient 群組是在 Pacemaker 安裝期間建立) 的使用者 ID。

您可以設定無密碼 SSH，以便只需要在 HA 群組中的一個節點上發出配置指令。

關於這項作業

若要設定無密碼 SSH，您必須在每一個節點上配置 mqm ID，然後在每一個節點上為該使用者產生金鑰。然後，將金鑰配送至其他節點，並測試連線以將每一個節點新增至已知主機清單。最後，您會鎖定 mqm ID。

註：這些指示假設您使用個別的主要、替代及抄寫介面來定義 HA 群組，因此您透過主要及替代介面來定義無密碼 SSH 存取。如果您計劃配置具有單一 IP 位址的系統，則透過該單一介面定義無密碼 SSH 存取。

程序

1. 在三個節點中的每一個節點上，完成下列步驟來設定 mqm 使用者並產生 SSH 金鑰：

- a) 將 mqm 起始目錄切換至 /home/mqm：

```
usermod -d /home/mqm mqm
```

- b) 建立 /home/mqm 目錄：

```
mkhomedir_helper mqm
```

- c) 新增 mqm 密碼：

```
passwd mqm
```

- d) 以 mqm 身分執行互動式 Shell：

```
su mqm
```

- e) 產生 mqm 鑑別金鑰：

```
ssh-keygen -t rsa -f /home/mqm/.ssh/id_rsa -N ''
```

2. 在三個節點中的每一個節點上，完成下列步驟，以將該節點的金鑰新增至其他兩個節點，並測試每一個節點主要及 (如果使用的話) 替代位址的連線：

- a) 將金鑰新增至遠端節點

```
ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node1_primary_address
ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node1_alternate_address
ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node2_primary_address
ssh-copy-id -i /home/mqm/.ssh/id_rsa.pub remote_node2_alternate_address
```

- b) 檢查遠端節點的無密碼 ssh 及更新 known_hosts：

```
ssh remote_node1_primary_address uname -n
ssh remote_node1_alternate_address uname -n
ssh remote_node2_primary_address uname -n
ssh remote_node2_alternate_address uname -n
```

對於每一個連線，系統會提示您確認要繼續進行。確認每一個都要更新 known_hosts。您必須先完成此作業，然後再嘗試使用無密碼 SSH 來配置 HA 群組。

- c) 以 mqm 身分結束互動式 Shell：

```
exit
```

3. 在每一個節點上，以 root 身分完成下列步驟，以移除 mqm 密碼並鎖定 ID：

- a) 移除 mqm 密碼：

```
passwd -d mqm
```

- b) 鎖定 mqm：

```
passwd -l mqm
```

4. 在每一個節點上，以 root 身分完成下列步驟，以設定 mqm 使用者的 sudo 存取權：

a) 使用 **visudo** 指令編輯 sudoers 檔案：

```
visudo
```

b) 搜尋 "### Allows people in group wheel to run all commands" 行，並在該行下方新增下列文字：

```
##mqm    ALL=(ALL)    ALL
```

c) 搜尋 "### Same thing without a password" 行，並在該行下方新增下列文字：

```
%mqm    ALL=(ALL)    NOPASSWD: ALL
```

Linux V 9.1.0 定義 Pacemaker 叢集 (HA 群組)

HA 群組是 Pacemaker 叢集。您可以透過編輯 `/var/mqm/rdqm.ini` 檔案並執行 **rdqmadm** 指令，來定義 Pacemaker 叢集。

關於這項作業

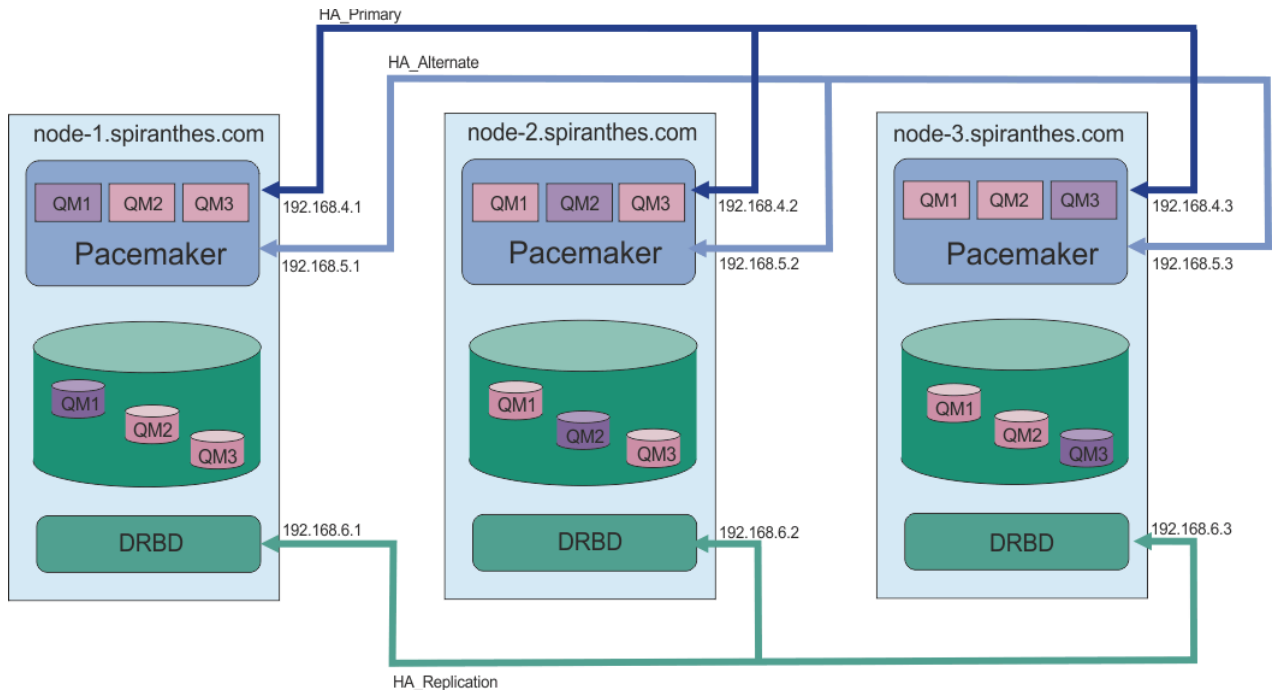
如需 Pacemaker 的相關資訊，請參閱 <https://clusterlabs.org/pacemaker/>。如果 mqm 使用者可以使用 sudo，您可以將 Pacemaker 叢集建立為 mqm 群組中的使用者。如果使用者也可以在沒有密碼的情況下透過 SSH 連接至每一部伺服器，則您只需要編輯 `rdqm.ini` 檔案，並在其中一部伺服器上執行 **rdqmadm**，以建立 Pacemaker 叢集。否則，您必須建立檔案，並在每一部要成為節點的伺服器上，以 root 身分執行指令。

`rdqm.ini` 檔案提供 Pacemaker 叢集中所有節點的 IP 位址。您可以定義用於 Pacemaker 的 `HA_Primary` 及 `HA_Secondary` 介面，以監視系統，但 Pacemaker 可以改為將名為 `HA_Replication` 的抄寫介面用於此用途 (必要的話)。在 HA 群組中執行的所有 RDQM 預期工作量的情況下，`HA_Replication` 介面必須具有足夠的頻寬來支援抄寫需求。

下列範例檔顯示範例 Pacemaker 叢集的配置，該叢集針對每一個介面使用個別 IP 位址：

```
Node:
  HA_Primary=192.168.4.1
  HA_Alternate=192.168.5.1
  HA_Replication=192.168.6.1
Node:
  HA_Primary=192.168.4.2
  HA_Alternate=192.168.5.2
  HA_Replication=192.168.6.2
Node:
  HA_Primary=192.168.4.3
  HA_Alternate=192.168.5.3
  HA_Replication=192.168.6.3
```

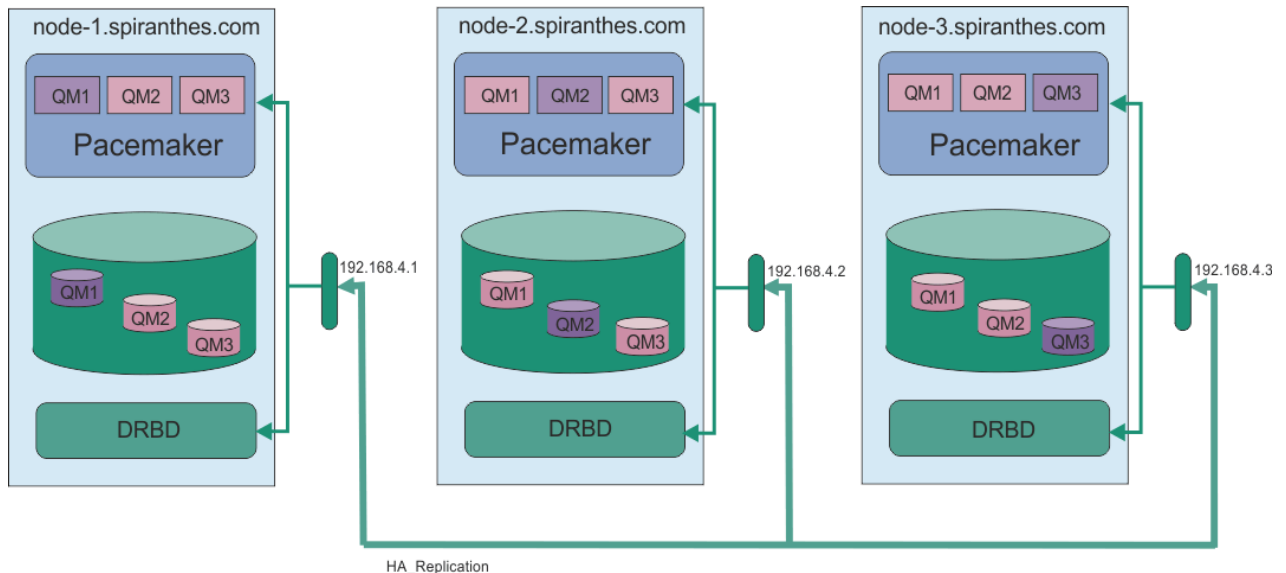
下圖說明此配置：



下列範例檔案顯示使用 HA_Replication 介面進行監視之範例 Pacemaker 叢集的配置。在此情況下，您只指定 HA_Replication 介面：

```
Node:
  HA_Replication=192.168.4.1
Node:
  HA_Replication=192.168.4.2
Node:
  HA_Replication=192.168.4.3
```

下圖說明此配置：

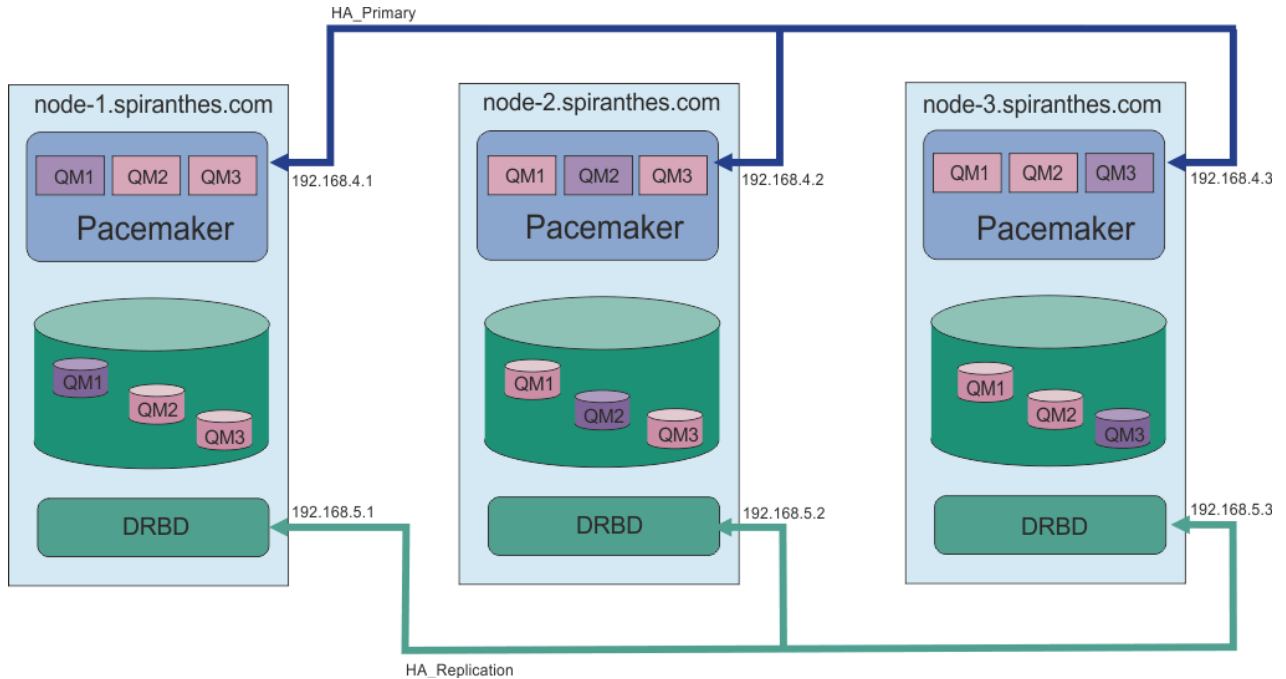


如果您想要使用兩個 IP 位址，則 `rdqm.ini` 檔案具有每一個節點的 `HA_Primary` 和 `HA_Replication` 欄位，但沒有 `HA_Alternate` 欄位：

```
Node:
  HA_Primary=192.168.4.1
  HA_Replication=192.168.5.1
Node:
  HA_Primary=192.168.4.2
```

```
HA_Replication=192.168.5.2
Node:
HA_Primary=192.168.4.3
HA_Replication=192.168.5.3
```

下圖說明此配置:



在配置中的所有 `rdqm.ini` 檔案中，您指定節點的順序必須相同。您的三個節點必須具有共用視圖，其中哪個是 Node1，哪個是 Node2 等等。

程序

- 若要將 Pacemaker 叢集定義為使用者 `root`，請執行下列動作：
 - a) 編輯三部伺服器之一上的 `/var/mqm/rdqm.ini` 檔案，以便該檔案定義叢集。
 - b) 將檔案複製到將成為 Pacemaker 叢集中節點的其他兩部伺服器。
 - c) 在三部伺服器中的每一部伺服器上，以 `root` 身分執行下列指令：

```
rdqmadm -c
```

- 若要將 Pacemaker 叢集定義為每一個節點上 `mqm` 群組中的使用者，請執行下列動作：
 - a) 請確定使用者 `mqm` 可以使用 `sudo` 來執行指令。
 - b) 編輯三部伺服器之一上的 `/var/mqm/rdqm.ini` 檔案，以便該檔案定義 Pacemaker 叢集。
 - c) 將 `/var/mqm/rdqm.ini` 複製到將成為 Pacemaker 叢集中節點的其他兩部伺服器。
 - d) 在每一部伺服器上執行下列指令：

```
rdqmadm -c
```

- 若要從一個節點將 Pacemaker 叢集定義為 `mqm` 群組中的使用者，請執行下列動作：
 - a) 確保使用者 `mqm` 可以使用 `sudo` 來執行指令，並且可以選擇性地使用 SSH 連接至每一個伺服器，而無需密碼。
 - b) 編輯三部伺服器之一上的 `/var/mqm/rdqm.ini` 檔案，以便該檔案定義 Pacemaker 叢集。
 - c) 請執行下列指令：

```
rdqmadm -c
```

相關參考

[rdqmadm \(管理抄寫的資料佇列管理程式叢集\)](#)

Linux V 9.1.0 刪除 Pacemaker 叢集 (HA 群組)

HA 群組是 Pacemaker 叢集。您可以執行 **rdqmadm** 指令並指定 **-u** 選項，以刪除 Pacemaker 叢集配置。

關於這項作業

如果任何抄寫的資料佇列管理程式仍存在於任何節點上，則無法刪除 Pacemaker 叢集配置。

程序

- 若要刪除 Pacemaker 叢集配置，請從任何節點輸入下列指令：

```
rdqmadm -u
```

相關參考

[rdqmadm \(管理抄寫的資料佇列管理程式叢集\)](#)

Linux V 9.1.0 建立 HA RDQM

您可以使用 **crtmqm** 指令來建立高可用性抄寫資料佇列管理程式 (RDQM)。

關於這項作業

如果 **mqm** 使用者可以使用 **sudo**，您可以在 **mqm** 群組中以使用者身分建立高可用性抄寫資料佇列管理程式 (RDQM)。如果使用者也可以在沒有密碼的情況下透過 **SSH** 連接至每一個節點，則您只需要在一個節點上執行 **create RDQM** 指令，即可在所有三個節點上建立 RDQM。否則，您必須是 **root** 才能建立 RDQM，且必須在所有三個節點上執行指令。

程序

- 如果要以 **mqm** 群組中的使用者身分來建立 RDQM，請執行下列動作：
 - 確保 **mqm** 使用者可以使用 **sudo** 來執行指令，並且可以使用 **SSH** 連接至每一個伺服器，而無需密碼。
 - 執行下列指令：

```
crtmqm -sx [-fs FilesystemSize] qmname
```

其中 **qmname** 是所抄寫資料佇列管理程式的名稱。您可以選擇性地指定佇列管理程式的檔案系統大小 (亦即，在 **drbdpool** 磁區群組中建立的邏輯磁區大小)。

該指令會嘗試使用 **SSH**，以 **mqm** 使用者身分連接至叢集中的其他節點。如果連線成功，則會在節點上建立佇列管理程式的次要實例。否則，您必須建立次要實例，然後執行 **crtmqm -sx** 指令 (如使用者 **root** 所述)。

- 如果要以使用者 **root** 身分建立 RDQM，請執行下列動作：
 - 在每一個節點上輸入下列指令，以管理 RDQM 的次要實例：

```
crtmqm -sxs [-fs FilesystemSize] qmname
```

其中 **qmname** 是所抄寫資料佇列管理程式的名稱。您可以選擇性地指定佇列管理程式的檔案系統大小 (亦即，在 **drbdpool** 磁區群組中建立的邏輯磁區大小)。您必須對 **HA** 群組中所有三個節點上的 RDQM 指定相同的檔案系統大小。

這個指令會建立 RDQM 的次要實例。

- 在其餘節點上，輸入下列指令：

```
crtmqm -sx [-fs FilesystemSize] qmname
```


其中 *qmname* 是所抄寫資料佇列管理程式的名稱。您可以選擇性地指定佇列管理程式的檔案系統大小。

該指令會判斷佇列管理程式的次要實例是否存在於其他兩個節點上。如果次要檔存在，指令會建立並啟動主要佇列管理程式。如果次要伺服器不存在，則會指示您在每一個節點上執行 **crtmqm -sxs** 指令。

除了 DataPath (**-md**) 和 LogPath (**-ld**) 引數之外，適用於建立標準 Linux 佇列管理程式的所有引數也適用於主要抄寫資料佇列管理程式。

註: 當您建立 RDQM 時，會為抄寫鏈結配置下一個高於 7000 的可用埠號。如果探索到另一個應用程式正在使用所選擇的埠，則 **crtmqm** 指令會失敗並產生 AMQ6543 錯誤，且該埠會新增至排除清單。您必須刪除佇列管理程式的次要實例，然後重新執行 **crtmqm** 指令。

相關參考

[crtmqm](#)

Linux V 9.1.0 刪除 HA RDQM

您可以使用 **dltmqm** 指令來刪除高可用性抄寫資料佇列管理程式 (RDQM)。

關於這項作業

您必須執行指令來刪除 RDQM 主要節點上的 RDQM。RDQM 必須先結束。如果 mqm 使用者具有必要的 sudo 專用權，則可以該使用者身分執行指令。否則，您必須以 root 身分執行指令。刪除與主要佇列管理程式相關聯的資源之後，指令會嘗試使用 ssh 來刪除次要佇列管理程式，以連接至其他節點。如果此刪除失敗，您必須在其他節點上手動執行 **dltmqm**，以完成處理程序。在次要節點上，如果尚未刪除主要佇列管理程式，則指令會失敗。

程序

- 如果要刪除 RDQM，請輸入下列指令：

```
dltmqm RDQM_name
```

相關參考

[dltmqm](#)

MQ Adv. Linux 移轉佇列管理程式以變成 HA RDQM 佇列管理程式

您可以透過備份其持續資料，然後將資料還原至新建立且具有相同名稱的 RDQM 佇列管理程式，將現有佇列管理程式移轉為高可用性 (HA) 抄寫的資料佇列管理程式 (RDQM)。

關於這項作業

HA 抄寫資料佇列管理程式需要專用邏輯磁區 (檔案系統)，以及磁碟抄寫和 HA 控制的配置。只有在建立新的佇列管理程式時，才會配置這些元件。可以移轉現有佇列管理程式以使用 RDQM，方法是備份其持續資料，然後將資料還原至新建立且具有相同名稱的 RDQM 佇列管理程式。此程序會保留建立備份時的佇列管理程式配置、狀態及持續訊息。

註: 您只能從與 RDQM 安裝版本相同或更低的 IBM MQ 版本移轉佇列管理程式。作業系統和架構也必須相同。否則，您必須在目標平台上建立新的佇列管理程式，請參閱 [將佇列管理程式移至不同的作業系統](#)。

在移轉佇列管理程式之前，您應該滿足下列條件：

- 評估您的高可用性需求，並參閱 [第 448 頁的『RDQM 高可用性』](#)。
- 檢閱連接至佇列管理程式的應用程式及佇列管理程式。請考量將連線遞送至佇列管理程式執行所在之 RDQM 節點所需的變更。比方說，如果您配置 RDQM 高可用性，您可以考慮使用浮動 IP 位址，請參閱 [第 463 頁的『建立及刪除浮動 IP 位址』](#)。
- 為您選擇的配置佈建或識別現有的 RDQM 節點。如需 RDQM 系統需求的相關資訊，請參閱 [第 450 頁的『RDQM HA 解決方案的需求』](#)。
- 在每一個節點上安裝 IBM MQ Advanced，其中包括 RDQM 特性。

- 配置 RDQM HA 群組配置，請參閱 [第 453 頁的『定義 Pacemaker 叢集 \(HA 群組\)』](#)。
- 選擇性地使用測試佇列管理程式來驗證 RDQM 配置，然後可以刪除該配置。建議測試配置，以在移轉佇列管理程式之前識別並解決任何問題。
- 請檢閱佇列管理程式的安全配置，然後在每一個 RDQM 節點上抄寫必要的本端使用者和群組。
- 請檢閱佇列管理程式及通道配置，以判斷是否使用 API 結束程式、通道結束程式或資料轉換結束程式。在每一個 RDQM 節點上安裝必要的結束程式。
- 請檢閱已定義的任何佇列管理程式服務，然後在每一個 RDQM 節點上安裝並配置必要的程序。

程序

1. 備份現有的佇列管理程式:

- 透過發出等待關閉指令 `endmqm -w` 或立即關閉指令 `endmqm -i` 來停止現有佇列管理程式。此步驟很重要，可確保備份中的資料一致。
- 透過檢視 IBM MQ 配置檔 `mq.s.ini`，判定佇列管理程式資料目錄的位置。在 Linux 上，此檔案位於 `/var/mqm` 目錄中。如需 `mq.s.ini` 的相關資訊，請參閱 [第 74 頁的『IBM MQ 配置檔 mq.s.ini』](#)。
在檔案中找出佇列管理程式的 `QueueManager` 段落。如果段落包含名為 `DataPath` 的索引鍵，則其值為佇列管理程式資料目錄。如果金鑰不存在，則可以使用 `Prefix` 及 `Directory` 金鑰的值來判定佇列管理程式資料目錄。佇列管理程式資料目錄是這些值的連結，格式為 `prefix/qmgrs/directory`。如需 `QueueManager` 段落的相關資訊，請參閱 [第 83 頁的『mq.s.ini 檔案的 QueueManager 段落』](#)。
- 建立佇列管理程式資料目錄的備份。在 Linux 上，您可以使用 `tar` 指令來執行此動作。例如，若要備份佇列管理程式的資料目錄，您可以使用下列指令。請注意指令的最後一個參數，它是單一句點 (點):

```
tar -cvzf qm-data.tar.gz -C queue_manager_data_dir .
```

- 檢視 IBM MQ 佇列管理程式配置檔 `qm.ini`，以判斷佇列管理程式日誌目錄的位置。此檔案位於佇列管理程式資料目錄中。如需檔案的相關資訊，請參閱 [第 84 頁的『佇列管理程式配置檔 qm.ini』](#)。
佇列管理程式日誌目錄定義為 `Log` 段落中 `LogPath` 索引鍵的值。如需段落的相關資訊，請參閱 [第 108 頁的『qm.ini 檔案的日誌段落』](#)。
- 建立佇列管理程式日誌目錄的備份。在 Linux 上，您可以使用 `tar` 指令來執行此動作。例如，若要備份佇列管理程式的日誌目錄，您可以使用下列指令。請注意指令的最後一個參數，它是單一句點 (點):

```
tar -cvzf qm-log.tar.gz -C queue_manager_log_dir .
```

- 如果佇列管理程式使用的任何憑證儲存庫不在佇列管理程式資料目錄中，請建立它們的備份。請確定已備份金鑰資料庫檔和密碼隱藏檔。如需佇列管理程式金鑰儲存庫的相關資訊，請參閱 [SSL/TLS 金鑰儲存庫及尋找佇列管理程式的金鑰儲存庫](#)。如需在佇列管理程式配置為使用「AMS 訊息通道代理程式 (MCA)」截取時尋找 AMS 金鑰儲存庫的相關資訊，請參閱 [訊息通道代理程式 \(MCA\) 截取](#)。
- 不再需要現有的佇列管理程式，因此可以刪除它。不過，可能的話，在目標系統上順利還原現有的佇列管理程式之後，您應該只刪除它。延遲刪除可確保在移轉處理程序未順利完成時，佇列管理程式可以重新啟動。

註: 如果您延遲刪除現有的佇列管理程式，請不要重新啟動它。佇列管理程式必須保留結束狀態，因為在移轉期間會遺失對其配置或狀態所做的進一步變更。

2. 準備主要 RDQM 節點:

- 建立新的 RDQM 佇列管理程式，其名稱與您所備份的佇列管理程式相同。請確定 `crtmqm` 配置給 RDQM 佇列管理程式的檔案系統夠大，足以包含現有佇列管理程式的資料、主要日誌和次要日誌，以及一些未來擴充的額外空間。如需如何建立 RDQM 佇列管理程式的相關資訊，請參閱 [第 456 頁的『建立 HA RDQM』](#)。
- 判斷佇列管理程式的主要 RDQM 節點。如需如何判斷主要節點的相關資訊，請參閱 [rdqmstatus \(顯示 RDQM 狀態\)](#)。
- 在主要 RDQM 節點上，如果 RDQM 佇列管理程式已啟動，請使用 `endmqm -w` 或 `endmqm -i` 指令停止它。

- d) 在主要 RDQM 節點上，判斷 RDQM 佇列管理程式的資料和日誌目錄位置 (使用步驟 1b 和 1d 中說明的方法)。
- e) 在主要 RDQM 節點上，刪除 RDQM 佇列管理程式資料和日誌目錄的內容，但不刪除目錄本身。
3. 還原主要 RDQM 節點上的佇列管理程式：
- a) 將佇列管理程式資料及日誌目錄的備份複製到主要 RDQM 節點，以及佇列管理程式所使用憑證儲存庫的任何個別備份。
- b) 將佇列管理程式資料目錄的備份還原至新的 RDQM 佇列管理程式的空資料目錄，以確保保留檔案所有權及許可權。如果是使用步驟 1c 中的範例 tar 指令來建立備份，則 root 使用者可以使用下列指令來還原它：

```
tar -xvzpf qm-data.tar.gz -C queue_manager_data_dir
```

- c) 將佇列管理程式日誌目錄的備份還原至新的 RDQM 佇列管理程式的空白日誌目錄，以確保保留檔案所有權和許可權。如果備份是使用步驟 1e 中的範例 tar 指令來建立，則 root 使用者可以使用下列指令來還原它：

```
tar -xvzpf qm-log.tar.gz -C queue_manager_log_dir
```

- d) 編輯 RDQM 佇列管理程式之資料目錄中的已還原佇列管理程式配置檔 `qm.ini`。更新 Log 段落中 LogPath 索引鍵的值，以指定 RDQM 佇列管理程式的日誌目錄。

請檢閱配置檔中定義的其他檔案路徑，並在必要時更新它們。例如，您可能需要更新下列路徑：

- 診斷訊息服務所產生錯誤日誌檔的路徑。
- 佇列管理程式所需的結束程式路徑。
- 如果佇列管理程式是 XA 交易協調程式，則為交換器載入檔的路徑。

- e) 如果佇列管理程式配置為使用「AMS 訊息通道代理程式 (MCA)」截取，請將 AMS 金鑰儲存庫複製到新的 RDQM 安裝架構，然後檢閱並更新配置。金鑰儲存庫必須在每一個 RDQM 節點上可用，因此如果它不在佇列管理程式的已抄寫檔案系統中，則必須改為複製到每一個節點。如需相關資訊，請參閱 [訊息通道代理程式 \(MCA\) 截取](#)。

- f) 驗證 `dspmq` 指令是否顯示佇列管理程式，且其狀態報告為已結束。下列範例顯示 RDQM HA 佇列管理程式的輸出範例：

```
$ dspmq -o status -o ha
QMNAME(QM1) STATUS(Ended normally) HA(Replicated)
```

- g) 使用 `rdqmstatus` 指令來顯示佇列管理程式的狀態，以驗證已還原的佇列管理程式資料已抄寫至次要 RDQM 節點。在每一個節點上，HA 狀態應該報告為 Normal。下列範例顯示 RDQM HA 佇列管理程式的輸出範例：

```
$ rdqmstatus -m QM1
Node:                               mqhvm10-adm
Queue manager status:                Ended normally
Queue manager file system:           50MB used, 0.2GB allocated [42%]
HA role:                              Primary
HA status:                            Normal
HA control:                           Disabled
HA current location:                  This node
HA preferred location:                 This node
HA floating IP interface:              None
HA floating IP address:                None

Node:                               mqhvm11-adm
HA status:                            Normal

Node:                               mqhvm12-adm
HA status:                            Normal
```

- h) 在主要 RDQM 節點上啟動佇列管理程式。
- i) 連接至佇列管理程式，並更新 SSLKEYR 佇列管理程式屬性的值，以指定佇列管理程式憑證儲存庫的新位置。依預設，此屬性的值設為 `queue_manager_data_directory/ssl/key`。憑證儲存庫必

須位於每一個 RDQM 節點上的相同位置。如果儲存庫不在佇列管理程式的抄寫檔案系統中，則必須改為將它複製到每一個節點。

- j) 檢閱佇列管理程式的 IBM MQ 物件定義，並更新參照已變更網路設定、IBM MQ 安裝目錄或佇列管理程式資料目錄 (包括下列物件) 的物件屬性值：
 - 接聽器使用的本端 IP 位址 (IPADDR 屬性)。
 - 通道使用的本端 IP 位址 (LOCLADDR 屬性)。
 - 定義給叢集接收端通道的本端 IP 位址 (CONNAME 屬性)。
 - 為通訊資訊物件定義的本端 IP 位址 (GRPADDR 屬性)。
 - 針對處理程序及服務物件定義定義的系統路徑。
- k) 停止並重新啟動佇列管理程式，以確保變更生效。
- l) 針對連接至已移轉佇列管理程式的遠端佇列管理程式，重複步驟 3j，以及應用程式的對等設定，包括：
 - 通道連線名稱 (CONNAME 屬性)。
 - 通道鑑別規則，根據佇列管理程式的 IP 位址或主機名稱來限制來自佇列管理程式的入埠連線。
 - 用戶端通道定義表 (CCDT)、網域名稱設定 (DNS)、網路遞送或對等連線資訊。
- m) 對每一個 RDQM 節點執行佇列管理程式的受管理失效接手，以確保已順利建立必要的配置，請參閱 [第 462 頁的『設定 RDQM 的偏好位置』](#)。

調整 HA RDQM 佇列管理程式的檔案系統大小

若要調整現有高可用性 (HA) 抄寫資料佇列管理程式 (RDQM) 的檔案系統大小，請備份其持續資料，然後將資料還原至新建立且名稱相同但檔案系統大小不同的 RDQM 佇列管理程式。

關於這項作業

HA 抄寫的資料佇列管理程式需要專用邏輯磁區 (檔案系統)，以及磁碟抄寫和 HA 控制的配置。只有在建立新的佇列管理程式時，才會配置這些元件。檔案系統在建立之後無法調整大小，因為它在每一個節點上必須具有相同的大小。若要調整現有抄寫資料佇列管理程式 (RDQM) 的檔案系統大小，您可以備份其持續資料，然後將資料還原至名稱相同但檔案系統大小不同的新建立 RDQM 佇列管理程式。此程序會保留建立備份時的佇列管理程式配置、狀態及持續訊息。

程序

1. 備份主要 RDQM 節點上的現有 RDQM 佇列管理程式：
 - a) 判斷佇列管理程式的主要 RDQM 節點。如需如何判斷主要節點的相關資訊，請參閱 [rdqmstatus \(顯示 RDQM 狀態\)](#)。
 - b) 在主要 RDQM 節點上，如果 RDQM 佇列管理程式已啟動，請使用 `endmqm -w` 或 `endmqm -i` 指令停止它。
 - c) 透過檢視 IBM MQ 配置檔 `mqs.ini`，判定佇列管理程式資料目錄的位置。在 Linux 上，此檔案位於 `/var/mqm` 目錄中。如需 `mqs.ini` 的相關資訊，請參閱 [第 74 頁的『IBM MQ 配置檔 mqs.ini』](#)。

在檔案中找出佇列管理程式的 `QueueManager` 段落。佇列管理程式資料目錄是名為 `DataPath` 的索引鍵值。如需 `QueueManager` 段落的相關資訊，請參閱 [第 83 頁的『mqs.ini 檔案的 QueueManager 段落』](#)。

- d) 建立佇列管理程式資料目錄的備份。在 Linux 上，您可以使用 `tar` 指令來執行此動作。例如，若要備份佇列管理程式的資料目錄，您可以使用下列指令。請注意指令的最後一個參數，它是單一點 (.) 字元：

```
tar -cvzf qm-data.tar.gz -C queue_manager_data_dir .
```

- e) 檢視 IBM MQ 佇列管理程式配置檔 `qm.ini`，以判斷佇列管理程式日誌目錄的位置。此檔案位於佇列管理程式資料目錄中。如需檔案的相關資訊，請參閱 [第 84 頁的『佇列管理程式配置檔 qm.ini』](#)。

佇列管理程式日誌目錄定義為日誌段落中 `LogPath` 索引鍵的值。如需段落的相關資訊，請參閱第 108 頁的『`qm.ini` 檔案的日誌段落』。

- f) 建立佇列管理程式日誌目錄的備份。在 Linux 上，您可以使用 `tar` 指令來執行此動作。例如，若要備份佇列管理程式的日誌目錄，您可以使用下列指令。請注意指令的最後一個參數，它是單一句點 (.) 字元：

```
tar -cvzf qm-log.tar.gz -C queue_manager_log_dir .
```

- g) 刪除現有的 RDQM 佇列管理程式。

2. 使用所需大小的檔案系統還原佇列管理程式：

- a) 建立新的 RDQM 佇列管理程式，其名稱與您所備份的佇列管理程式相同。請確定透過 `crtmqm` 配置給 RDQM 佇列管理程式的檔案系統是您需要的大小，而且它已夠大，足以包含現有佇列管理程式的資料、主要日誌及次要日誌，以及一些未來擴充的額外空間。如需如何建立 RDQM 佇列管理程式的相關資訊，請參閱第 456 頁的『建立 HA RDQM』。

- b) 判斷佇列管理程式的主要 RDQM 節點。如需如何判斷主要節點的相關資訊，請參閱 `rdqmstatus` (顯示 RDQM 狀態)。

- c) 在主要 RDQM 節點上，如果 RDQM 佇列管理程式已啟動，請使用 `endmqm -w` 或 `endmqm -i` 指令停止它。

- d) 在主要 RDQM 節點上，判斷 RDQM 佇列管理程式之資料和日誌目錄的新位置 (使用步驟 1c 和 1e 中說明的方法)。

- e) 在主要 RDQM 節點上，刪除 RDQM 佇列管理程式資料和日誌目錄的內容，但不刪除目錄本身。

- f) 在主要 RDQM 節點上，將佇列管理程式資料目錄的備份還原至新的 RDQM 佇列管理程式的空資料目錄，以確保保留檔案所有權及許可權。如果是使用步驟 1d 中的範例 `tar` 指令來建立備份，則 `root` 使用者可以使用下列指令來還原它：

```
tar -xvzpf qm-data.tar.gz -C queue_manager_data_dir
```

- g) 在主要 RDQM 節點上，將佇列管理程式日誌目錄的備份還原至新的 RDQM 佇列管理程式的空白日誌目錄，以確保保留檔案所有權和許可權。如果是使用步驟 1f 中的範例 `tar` 指令來建立備份，則 `root` 使用者可以使用下列指令來還原它：

```
tar -xvzpf qm-log.tar.gz -C queue_manager_log_dir
```

- h) 在主要 RDQM 節點上，編輯新 RDQM 佇列管理程式之資料目錄中的已還原佇列管理程式配置檔 `qm.ini`。更新 `Log` 段落中 `LogPath` 索引鍵的值，以指定您在步驟 2d 中判定的新 RDQM 佇列管理程式的日誌目錄。請檢閱配置檔中定義的其他檔案路徑，並在必要時更新它們。例如，您可能需要更新下列路徑：

- 診斷訊息服務所產生錯誤日誌檔的路徑。
- 佇列管理程式所需的結束程式路徑。
- 如果佇列管理程式是 XA 交易協調程式，則為交換器載入檔的路徑。

- i) 驗證 `dspmq` 指令是否顯示佇列管理程式，且其狀態報告為已結束。下列範例顯示 RDQM HA 佇列管理程式的輸出範例：

```
$ dspmq -o status -o ha
QMNAME(QM1) STATUS(Ended normally) HA(Replicated)
```

- j) 使用 `rdqmstatus` 指令來顯示佇列管理程式的狀態，以驗證已還原的佇列管理程式資料已抄寫至次要 RDQM 節點。在每一個節點上，HA 狀態應該報告為 `Normal`。下列範例顯示 RDQM HA 佇列管理程式的輸出範例：

```
$ rdqmstatus -m QM1
Node: mqhavm10-adm
Queue manager status:      Ended normally
Queue manager file system: 50MB used, 0.2GB
allocated [42%]
HA role:                   Primary
HA status:                 Normal
HA control:                Disabled
HA current location:       This node
```

```

HA preferred location:      This node
HA floating IP interface:  None
HA floating IP address:    None
Node:                      mqhavam11-adm
HA status:                 Normal
Node:                      mqhavam12-adm
HA status:                 Normal

```

k) 在主要 RDQM 節點上啟動佇列管理程式。

l) 對每一個 RDQM 節點執行佇列管理程式的受管理失效接手，以確保已順利建立必要的配置，請參閱 [第 462 頁的『設定 RDQM 的偏好位置』](#)。

V 9.1.5 儲存持續性應用程式狀態

您可以儲存與應用程式相關的持續狀態資訊，以及其他佇列管理程式資料。

每一個 IBM MQ 佇列管理程式都有其持續狀態的專用檔案系統，其中包括其佇列資料及回復日誌。在 RDQM 配置中，檔案系統由在 Linux 系統 (節點) 之間抄寫的邏輯磁區所支援。檔案系統包括 `userdata` 目錄，您可以用來儲存應用程式的持續狀態資訊。因此，當抄寫的資料佇列管理程式移至 RDQM 配置中的另一個節點上執行時，您有可用的應用程式環境定義及佇列管理程式環境定義。請參閱 [Unix 及 Linux 系統上的目錄內容](#)。

如果您選擇將應用程式狀態儲存在 `userdata` 目錄中，則必須注意寫入此位置的資料可能會耗用配置給佇列管理程式的可用磁碟空間。您必須確保仍有足夠的磁碟空間可供佇列管理程式寫入佇列資料、日誌及其他持續性狀態資訊。

`userdata` 目錄具有 `mqm` 使用者和群組所有權，而且是全球可讀取的，因此使用者可以存取它，而不需要在 IBM MQ 管理者群組中 (即 `mqm`)。您無法修改 `userdata` 目錄的許可權，但可以使用您需要的任何所有權和許可權在其中建立內容。

在 RDQM 佇列管理程式失效接手期間，佇列管理程式會結束，且其現行 RDQM 節點上的檔案系統會卸載。然後會裝載檔案系統，並在 RDQM 配置中的另一個節點上重新啟動佇列管理程式。如果處理程序的其中一個檔案具有開啟控點，則無法卸載檔案系統。為了確保佇列管理程式失效接手可以完成，如果無法卸載佇列管理程式檔案系統，則會傳送 `SIGTERM` 信號給具有開啟檔案控點的處理程序，如果未釋放開啟控點，則會接著 `SIGKILL`。您的應用程式必須設計成正確回應 `SIGTERM`。如果應用程式或處理程序配置為佇列管理程式服務，則在受管理失效接手期間，可以在解除裝載檔案系統之前，在佇列管理程式關閉期間結束它們。如果應用程式或處理程序未配置為佇列管理程式服務，或發生未受管理的失效接手 (例如失去額定)，則可能會傳送信號以釋放檔案系統。

Linux V 9.1.0 設定 RDQM 的偏好位置

抄寫資料佇列管理程式 (RDQM) 的「偏好位置」會識別 RDQM 應該執行的節點 (如果該節點可用的話)。

關於這項作業

「偏好的位置」是當 HA 群組處於正常狀態 (所有可用的節點和連線) 時，Pacemaker 應該在其上執行佇列管理程式的節點名稱。當建立佇列管理程式時，「偏好的位置」會起始設定為主要節點的名稱。您可以執行指令，在三個節點中的任何一個節點上設定「偏好位置」。您必須是同時屬於 `mqm` 和 `haclient` 群組的使用者。

程序

- 如果要將本端或指定節點指派為指名佇列管理程式的偏好位置，請輸入下列指令：

```
rdqmadm -p -m qmname [ -n nodename[,nodename] ]
```

其中 `qmname` 是您要指定偏好位置的 RDQM 名稱，而 `nodename` 選擇性地是偏好節點的名稱。

如果 HA 群組處於正常狀態，且「偏好位置」不是現行主要節點，則會在新的「偏好位置」上停止並重新啟動佇列管理程式。您可以指定以逗點區隔的兩個節點名稱清單，以指派偏好位置的第二個喜好設定。

- 如果要清除偏好的位置，使佇列管理程式在還原時不會自動回到節點，請輸入下列指令：

```
rdqmadm -p -m qmname -d
```

相關參考

[rdqmadm \(管理抄寫的資料佇列管理程式叢集\)](#)

Linux

V 9.1.0

建立及刪除浮動 IP 位址

浮動 IP 位址可讓用戶端對抄寫的資料佇列管理程式 (RDQM) 使用相同的 IP 位址，而不論其執行所在 HA 群組中的哪個節點。

關於這項作業

您可以使用 **rdqmint** 指令來建立或刪除浮動 IP 位址。浮動位址會連結至 RDQM 之主要節點上的具名實體介面。如果 RDQM 失效接手，且不同的節點變成主要節點，則浮動 IP 會連結至新主要節點上同名的介面。三個節點上的實體介面必須屬於與浮動 IP 位址相同的子網路。下圖說明浮動 IP 位址的用法。

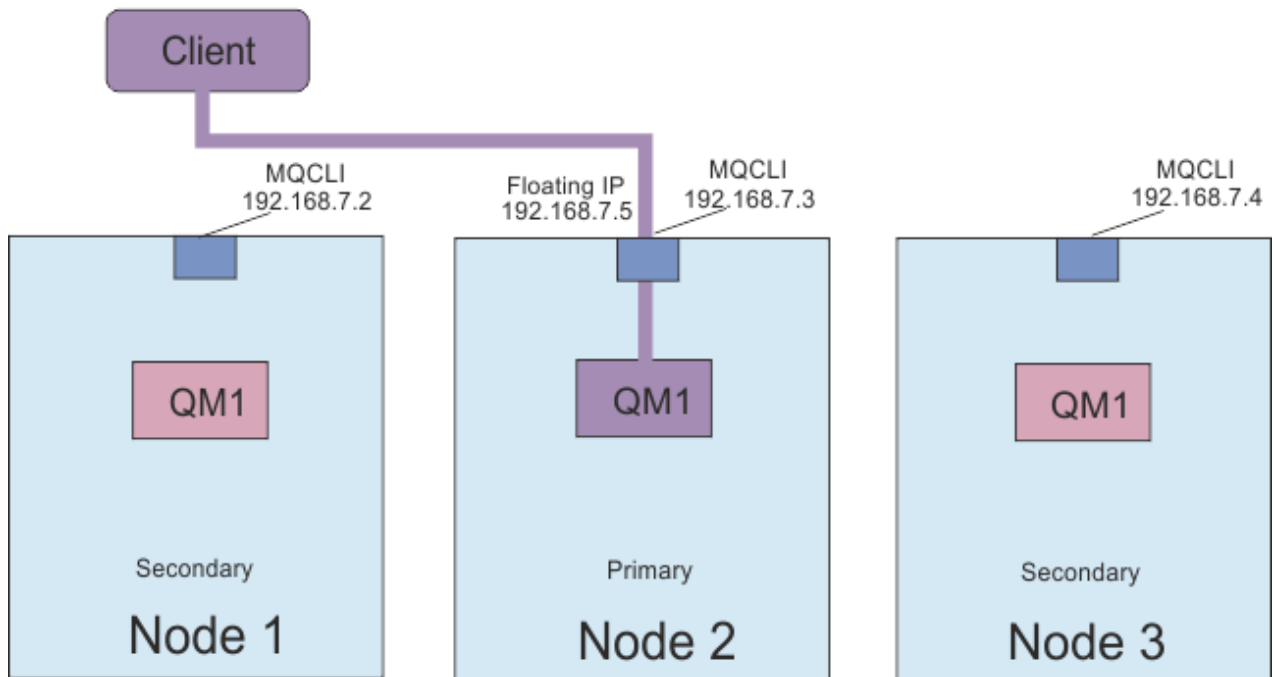
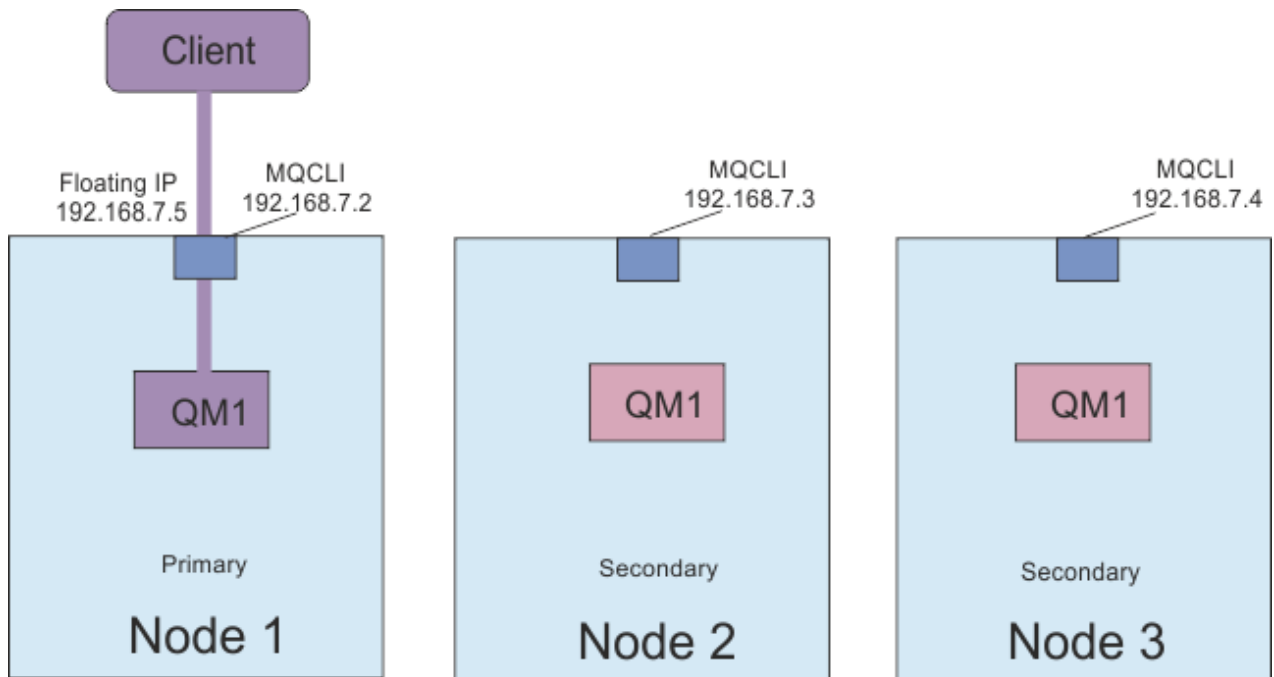


圖 83: 浮動 IP 位址

您必須同時是 `mqm` 和 `haclient` 群組中的使用者，才能執行 `rdqmint` 指令。您可以在 RDQM 的主要節點或其中一個次要節點上建立或刪除浮動 IP 位址。

註: 您不能對多個 RDQM 使用相同的浮動 IP 位址，每一個 RDQM 的浮動 IP 位址必須是唯一的。

程序

- 如果要建立 RDQM 的浮動 IP 位址，請輸入下列指令:

```
rdqmint -m qmname -a -f ipv4address -l interfacename
```


其中：

QMNAME

是您要為其建立浮動 IP 位址的 RDQM 名稱。

ipv4address

ipv4 格式的浮動 IP 位址。

浮動 IP 位址必須是尚未在任一軟體驅動裝置上定義的有效 IPv4 位址，並且它必須與為本端介面定義的靜態 IP 位址所屬的子網路相同。

interfaceName

主要節點上要連結的實體介面名稱。

例如：

```
rdqmint -m QM1 -a -f 192.168.7.5 -l MQCLI
```

- 若要刪除現有的浮動 IP 位址，請輸入下列指令：

```
rdqmint -m qmname -d
```

相關參考

[rdqmint \(新增或刪除 RDQM 的浮動 IP 位址\)](#)

Linux

V 9.1.0

啟動、停止及顯示 HA RDQM 的狀態

您可以使用標準 IBM MQ 控制指令的變式，來啟動、停止及檢視所抄寫資料佇列管理程式 (RDQM) 的現行狀態。

關於這項作業

您必須以同時屬於 `mqm` 和 `haclient` 群組的使用者身分執行指令，以啟動、停止及檢視已抄寫資料佇列管理程式 (RDQM) 的現行狀態。

您必須執行指令，以在該佇列管理程式的主要節點上啟動及停止佇列管理程式。

程序

- 如果要啟動 RDQM，請在 RDQM 的主要節點上輸入下列指令：

```
strmqm qmname
```

其中 `qmname` 是您要啟動之 RDQM 的名稱。

RDQM 已啟動，且 Pacemaker 開始管理 RDQM。如果您想要指定任何其他 `strmqm` 選項，則必須將 `-ns` 選項與 `strmqm` 一起指定。

- 若要停止 RDQM，請在 RDQM 的主要節點上輸入下列指令：

```
endmqm qmname
```

其中 `qmname` 是您要停止之 RDQM 的名稱。

Pacemaker 會停止管理 RDQM，然後結束 RDQM。停止 RDQM 時，可以使用所有其他 `endmqm` 參數。

- 如果要檢視 RDQM 的狀態，請輸入下列指令：

```
dspmq
```

輸出的狀態資訊取決於您是在 RDQM 的主要或次要節點上執行指令。如果在主要節點上執行，則會顯示 **dspmq** 所傳回的其中一個正常狀態訊息。如果您在次要節點上執行指令，則會顯示狀態 `running elsewhere`。例如，如果在節點 RDQM7 上執行 **dspmq**，則可能會傳回下列資訊：

```
QMNAME(RDQM8)          STATUS(Running elsewhere)
QMNAME(RDQM9)          STATUS(Running elsewhere)
QMNAME(RDQM7)          STATUS(Running)
```

如果主要節點無法使用，或者 **dspmq** 由非 root 的使用者或 haclient 群組成員執行，則會報告 Unavailable 狀態。例如：

```
QMNAME(RDQM8)          STATUS(Unavailable)
QMNAME(RDQM9)          STATUS(Unavailable)
QMNAME(RDQM7)          STATUS(Unavailable)
```

您可以輸入指令 **dspmq -o ha** (或 **dspmq -o HA**) 來檢視節點已知的佇列管理程式清單，以及它們是否為 RDQM，例如：

```
dspmq -o ha

QMNAME(RDQM8)          HA(Replicated)
QMNAME(RDQM9)          HA(Replicated)
QMNAME(RDQM7)          HA(Replicated)
QMNAME(QM7)            HA()
```

相關參考

[dspmq \(顯示佇列管理程式\)](#)

[endmqm \(結束佇列管理程式\)](#)

[strmqm \(啟動佇列管理程式\)](#)

Linux V 9.1.0 檢視 RDQM 和 HA 群組狀態

您可以檢視 HA 群組及個別抄寫資料佇列管理程式 (RDQM) 的狀態。

關於這項作業

您可以使用 **rdqmstatus** 指令來檢視個別 RDQM 及整體 HA 群組的狀態。

您必須是 mqm 和 haclient 群組中的使用者，才能執行 **rdqmstatus** 指令。您可以在三個節點中的任何一個節點上執行該指令。

程序

- 如果要檢視屬於 HA 配置之節點和 RDQ 的狀態，請執行下列動作：

```
rdqmstatus
```

即會顯示您執行指令所在節點的識別，以及 HA 配置中 RDQM 的狀態，例如：

```
Node:                  mqhvm07.exampleco.com
Queue manager name:   RDQM8
Queue manager status: Running elsewhere
HA current location:  mqhvm08.exampleco.com

Queue manager name:   RDQM9
Queue manager status: Running elsewhere
HA current location:  mqhvm09.exampleco.com

Queue manager name:   RDQM7
Queue manager status: Running
HA current location:  This node
```

- 若要檢視 HA 群組中三個節點的狀態，請輸入下列指令：

```
rdqmstatus -n
```

會報告每一個節點的線上或離線狀態。例如：

```
Node mqha04(mqhvm04.example.com) is online
Node mqha05(mqhvm05.example.com) is offline
Node mqha06(mqhvm06.example.com) is online
```

- 若要檢視 HA 群組中所有節點上特定佇列管理程式的狀態，請輸入下列指令：

```
rdqmstatus -m qmname
```

其中 *qmname* 是您要檢視其狀態的 RDQM 名稱。會顯示現行節點上 RDQM 的狀態，後面接著現行節點視景中其他兩個節點的狀態摘要。

下表彙總 **rdqmstatus** 指令可針對 RDQM 傳回之現行節點的相關資訊。

表 33: 現行節點狀態		
「狀態」屬性	可能值	顯示時
節點名稱	<i>nodeName</i>	一律顯示
佇列管理程式狀態	執行中 在別處執行 已結束 無法使用	一律顯示
CPU	<i>n.nn%</i>	只有在現行節點具有主要角色 (亦即, RDQM 正在此節點上執行) 時才會顯示
記憶體	<i>nnn</i> MB 已使用, <i>y.y</i> 已配置 GB	只有在現行節點具有主要角色 (亦即, RDQM 正在此節點上執行) 時才會顯示
佇列管理程式檔案系統	<i>nnn</i> 已使用 MB, <i>y.y</i> 已配置 GB [<i>z%</i>]	只有在現行節點具有主要角色 (亦即, RDQM 正在此節點上執行) 時才會顯示
HA 角色	主要次要不明	一律顯示
HA 狀態	所有節點處於待命狀態 此節點處於待命狀態 遠端節點待命中 混合 遠端節點的狀態	所有節點處於待命狀態 現行節點處於待命狀態 兩個遠端節點都處於待命狀態 每一個遠端節點的不同狀態 (請參閱下表以取得個別狀態) 兩個遠端節點的狀態相同 (請參閱下表以取得所有值)
HA 控制	已啟用 已停用 不明	一律顯示。顯示 RDQM 是否在 Pacemaker 控制下
HA 偏好位置	無 此節點 不明 <i>nodeName</i>	一律顯示
HA 浮動 IP 介面	<i>interface_name</i>	一律顯示
HA 浮動 IP 位址	<i>IPV4_address</i>	一律顯示

下表彙總 **rdqmstatus** 指令針對 HA 群組中其他節點所傳回的資訊。

表 34: 其他節點狀態		
「狀態」屬性	可能值	顯示時
節點名稱	<code>nodename</code>	一律顯示
HA 狀態	正常 正在進行同步化 遠端節點無法使用 不一致 已暫停 遠端節點處於待命狀態 不明	節點彼此同步 正在與遠端節點同步化 無法與遠端節點通訊 與遠端節點不同步且不同步 抄寫已暫停 遠端節點處於待命狀態
HA 同步化進行中	<code>n.n%</code>	在同步化進行中，且指令以 <code>root</code> 身分執行時顯示
HA 預估同步化時間	<code>yyyy-mm-dd hh:mm:ss.nnn</code>	正在進行同步化時顯示
HA 不同步資料	<code>nKB</code>	當遠端節點無法使用或不一致時顯示

範例

主要節點上正常狀態的範例:

```
Node:                               mqhvm07.exampleco.com
Queue manager status:              Running
CPU:                               0.00
Memory:                             123MB
Queue manager file system:         606MB used, 1.0GB allocated [60%]
HA role:                             Primary
HA status:                           Normal
HA control:                           Enabled
HA current location:                 This node
HA preferred location:                 This node
HA floating IP interface:             Eth4
HA floating IP address:               192.0.2.4
```

```
Node:                               mqhvm08.exampleco.com
HA status:                           Normal
```

```
Node:                               mqhvm09.exampleco.com
HA status:                           Normal
```

次要節點上正常狀態的範例:

```
Node:                               mqhvm08.exampleco.com
Queue manager status:              Running elsewhere
HA role:                             Secondary
HA status:                           Normal
HA control:                           Enabled
HA current location:                 mqhvm07.exampleco.com
HA preferred location:                 mqhvm07.exampleco.com
HA floating IP interface:             Eth4
HA floating IP address:               192.0.2.4
```

```
Node:                               mqhvm07.exampleco.com
HA status:                           Normal
```

```
Node:                               mqhvm09.exampleco.com
HA status:                           Normal
```

正在進行同步化時主要節點上的狀態範例:

```
Node:                               mqhvm07.exampleco.com
Queue manager status:              Running
CPU:                               0.53
Memory:                             124MB
```

```

Queue manager file system:      51MB used, 1.0GB allocated [5%]
HA role:                        Primary
HA status:                      Synchronization in progress
HA control:                     Enabled
HA current location:           This node
HA preferred location:         This node
HA floating IP interface:      Eth4
HA floating IP address:       192.0.2.4

Node:                            mqhvm08.exampleco.com
HA status:                      Synchronization in progress
HA synchronization progress:   11.0%
HA estimated time to completion: 2017-09-06 14:55:05

Node:                            mqhvm09.exampleco.com
HA status:                      Synchronization in progress
HA synchronization progress:   11.0%
HA estimated time to completion: 2017-09-06 14:55:06

```

顯示多個狀態的主要節點範例:

```

Node:                            mqhvm07.exampleco.com
Queue manager status:          Running
CPU:                          0.02
Memory:                       124MB
Queue manager file system:     51MB used, 1.0GB allocated [5%]
HA role:                        Primary
HA status:                      Mixed
HA control:                     Enabled
HA current location:           This node
HA preferred location:         This node
HA floating IP interface:      Eth4
HA floating IP address:       192.0.2.4

Node:                            mqhvm08.exampleco.com
HA status:                      Normal

Node:                            mqhvm09.exampleco.com
HA status:                      Inconsistent

```

相關參考

 [rdqmstatus](#)

  [在高可用性配置中更換故障節點](#)

如果 HA 群組中的其中一個節點失敗，您可以更換它。

關於這項作業

更換節點所採取的步驟取決於實務範例:

- 如果您要將故障節點更換為具有相同配置的節點，則可以在不中斷 HA 群組的情況下更換節點。
- 如果新節點具有不同的配置，則您必須刪除然後重建 HA 群組。您可以先從其執行所在的節點備份佇列管理程式，然後在重建 HA 群組之後還原它們。

程序

- 如果將替換用節點配置成看起來像故障節點 (相同主機名稱、相同 IP 位址等)，請在新節點上完成下列步驟:
 - a) 建立與其他節點上的檔案相符的 `rdqm.ini` 檔案，然後執行 `rdqmadm -c` 指令 (請參閱 [第 453 頁的『定義 Pacemaker 叢集 \(HA 群組\)』](#))。
 - b) 執行 `crtmqm -sxs qmanager` 指令，以重建每一個抄寫的資料佇列管理程式 (請參閱 [第 456 頁的『建立 HA RDQM』](#))。
- 如果更換節點與故障節點具有不同的配置:
 - a) 必要的話，請備份佇列管理程式 (請參閱 [第 525 頁的『備份及還原 IBM MQ 佇列管理程式資料』](#))。
 - b) 使用 `dlmqm` 指令，從 HA 群組中的其他節點刪除抄寫的資料佇列管理程式 (請參閱 [第 457 頁的『刪除 HA RDQM』](#))。

- c) 使用 `rdqmadm -u` 指令取消配置 Pacemaker 叢集 (請參閱 [第 456 頁的『刪除 Pacemaker 叢集 \(HA 群組\)』](#))。
- d) 使用 `rdqmadm -c` 指令來重新配置 Pacemaker 叢集，包括新節點的資訊 (請參閱 [第 453 頁的『定義 Pacemaker 叢集 \(HA 群組\)』](#))。
- e) 必要的話 (亦即，如果您沒有其他節點的 SSH 存取權)，請執行 `crtmqm -sxs qmanager` 指令，在其他節點上重建每一個抄寫的資料佇列管理程式 (請參閱 [第 456 頁的『建立 HA RDQM』](#))。
- f) 執行 `crtmqm -sx qmanager` 指令，以在取代節點上建立佇列管理程式。
- g) 必要的話，請將資料及配置還原至佇列管理程式 (請參閱 [第 525 頁的『備份及還原 IBM MQ 佇列管理程式資料』](#))。

MQ Adv. Linux V9.1.0 **RDQM 災難回復**

RDQM (抄寫的資料佇列管理程式) 可在 Linux 平台子集上使用，而且可以提供災難回復解決方案。

如需完整資料，請參閱 軟體產品相容性報告。

您可以建立在一部伺服器上執行的災難回復佇列管理程式主要實例，以及在另一部伺服器上作為回復節點的佇列管理程式次要實例。在佇列管理程式實例之間抄寫資料。如果您失去主要佇列管理程式，則可以手動將次要實例設為主要實例並啟動佇列管理程式，然後從相同位置回復工作。當佇列管理程式處於次要角色時，您無法啟動它。兩個節點之間的資料抄寫由 DRBD 處理。

您可以選擇在主要和次要佇列管理程式之間同步或非同步抄寫資料。如果您選取非同步選項，則作業 (例如 IBM MQ PUT 或 GET) 會完成並回到應用程式，然後再將事件抄寫至次要佇列管理程式。非同步抄寫表示在回復狀況之後，部分傳訊資料可能會遺失。但是次要佇列管理程式將處於一致狀態，且能夠立即開始執行，即使它在訊息串流的稍早期部分啟動也一樣。

雖然您可以移轉現有的佇列管理程式，以變成 RDQM 佇列管理程式，但無法將災難回復新增至現有的佇列管理程式 (請參閱 [第 475 頁的『移轉佇列管理程式以變成 DR RDQM 佇列管理程式』](#))。無法以 RDQM 災難回復及 RDQM 高可用性來配置佇列管理程式。

您可以在許多不同的伺服器上執行數個 RDQM 佇列管理程式配對。例如，您可以讓主要災難回復佇列管理程式在不同節點上執行，而其所有次要災難回復佇列管理程式則在相同節點上執行。下列圖表說明部分範例配置。

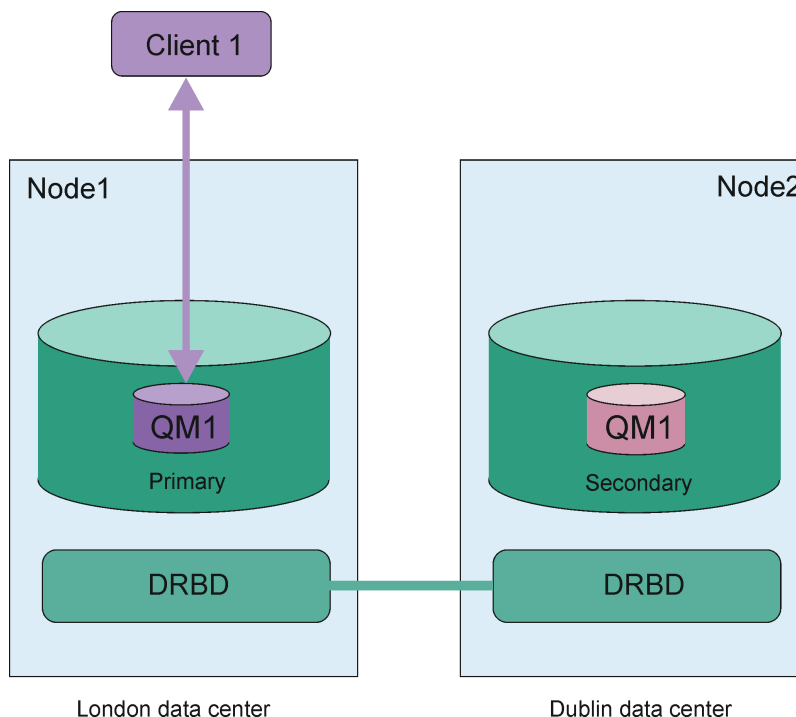


圖 84: 單一 RDQM 配對

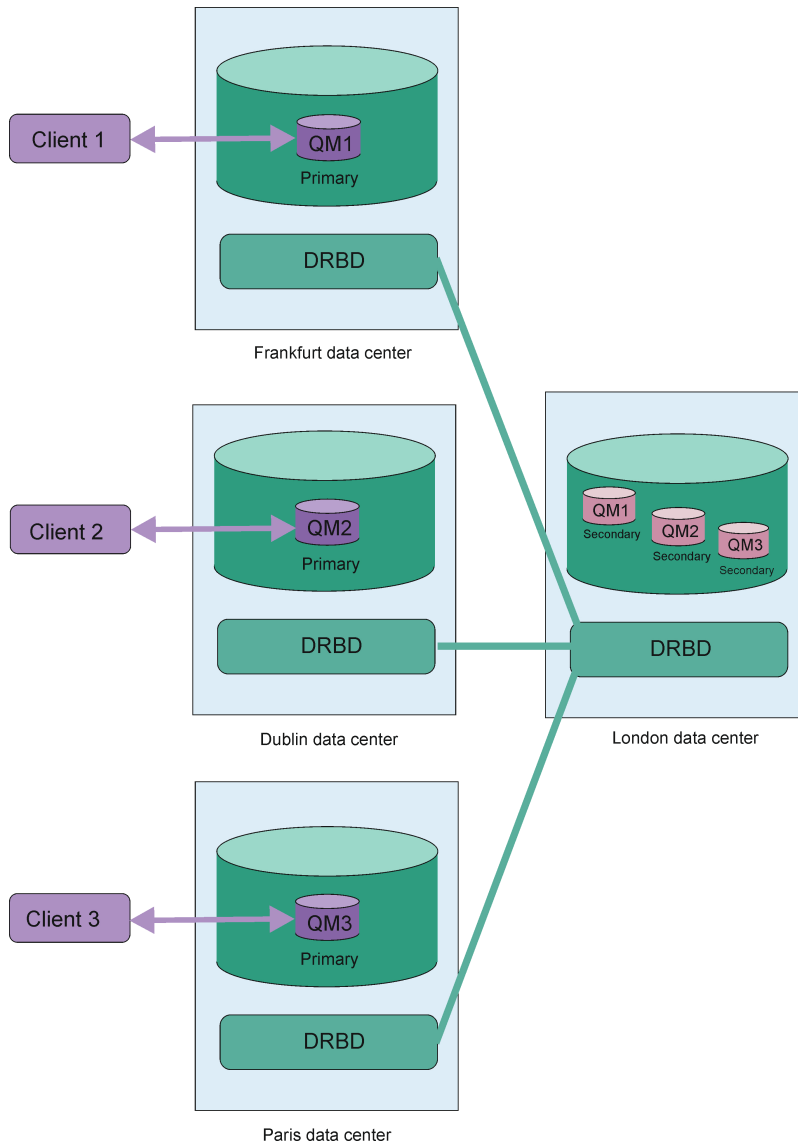


圖 85: 相同節點中的次要佇列管理程式

抄寫、同步化及 Snapshot

當災難回復配置中的兩個節點連接時，災難回復佇列管理程式的持續資料的任何更新都會從佇列管理程式的主要實例傳送至次要實例。這稱為 **抄寫**。

如果兩個節點之間的網路連線中斷，則會追蹤佇列管理程式主要實例的持續資料變更。還原網路連線時，會使用不同的處理程序，讓次要實例盡快達到速度。這稱為 **同步化**。

進行同步化時，次要實例上的資料處於不一致狀態。會取得次要佇列管理程式資料狀態的 **Snapshot**。如果在同步化期間發生主要節點或網路連線失敗，則次要實例會回復為這個 Snapshot，且可以啟動佇列管理程式。不過，自原始網路故障以來發生的任何更新都會遺失。

分割資料 (核心分割)

在失去佇列管理程式的主要實例之後，DR RDQM 配置需要使用者動作，才能在回復節點上升級及執行次要實例。不論誰 (或任何) 提升次要實例，都有責任確保先前的主要佇列管理程式已停止。如果原始主要資料庫保持執行中，它可能會處理訊息，當還原正常作業時，佇列管理程式的兩個實例會有不同的資料視圖。這稱為分割或核心分裂狀態。

請考量下列狀況：

- 執行主要佇列管理程式的節點完全失敗。您將次要實例升級成主要實例; 無法採取動作來停止原始主要實例, 因為它不在執行中。修復或取代原始節點時, 該節點上的佇列管理程式最初會成為次要佇列管理程式, 並與回復節點上的主要佇列管理程式同步。然後會反轉兩個佇列管理程式的角色, 並重新建議一般作業。在此狀況下, 唯一可能的資料流失是在節點失敗之前, 主要節點尚未完成抄寫至次要節點的任何資料。
- 網路故障會影響執行佇列管理程式主要及次要實例的節點之間的抄寫鏈結。在此狀況下, 您必須確保先停止原始主要資料庫, 然後再升級次要資料庫。如果原始主要資料庫仍然具有其他網路連線功能, 則實際上會同時有兩個主要實例在執行中, 且分割的資料可以累積。(如果抄寫鏈結正在運作中, 則如果主要實例仍在執行中, 則無法提升次要佇列管理程式, 指令會失敗。)
- 執行佇列管理程式主要實例的節點上發生完整網路失敗。同樣地, 您必須確保先停止主要實例, 然後再升級次要實例。如果在還原網路時前一個主要實例仍在執行中, 則會有兩個主要實例, 且會產生重新分割的資料。

當您執行受管理失效接手時, 應該不會看到佇列管理程式實例的 `partitioned DR` 狀態。受管理失效接手會結束主要節點上的佇列管理程式, 然後在完全抄寫資料之後, 啟動回復節點上的佇列管理程式。未預期分割狀態, 因為佇列管理程式已結束, 且在回復節點上啟動之前, 資料已在節點之間同步。如果在節點之間失去連線功能時在回復節點上啟動佇列管理程式, 則在失去連線功能時, 如果佇列管理程式在主要節點上處於作用中狀態, 則可能會出現資料分歧。在此實務範例中, 因為佇列管理程式資料未同步, 所以在還原連線功能之後, 預期會報告分割狀態。如果發生分割狀態, 您可能必須檢查這兩個資料集, 並針對要保留的資料集做出明智的決策。請參閱第 486 頁的『解決 DR RDQM 中的分割 (核心分割) 問題』。

Linux V 9.1.0 RDQM DR 解決方案的需求

在配置 RDQM 災難回復 (DR) 佇列管理程式配對之前, 您必須符合一些需求。

系統需求

在配置 RDQM DR 之前, 您必須在要管理 RDQM DR 佇列管理程式的每一部伺服器上完成部分配置。

- 每一個節點都需要一個名為 `drbdpool` 的磁區群組。每個災難回復抄寫資料佇列管理程式 (DR RDQM) 的儲存體, 會從這個磁區群組配置成每個佇列管理程式兩個個別的邏輯磁區。(每一個佇列管理程式都需要兩個邏輯磁區來支援回復至 Snapshot 作業, 因此每一個 DR RDQM 所配置的儲存體僅是您建立它時所指定之儲存體的兩倍。) 為了取得最佳效能, 此磁區群組應該由一或多個對應於內部磁碟機 (最好是 SSD) 的實體磁區組成。
- 建立 `drbdpool` 磁區群組之後, 對其不執行任何其他動作。IBM MQ 會管理在 `drbdpool` 中建立的邏輯磁區, 以及它們的裝載方式和裝載位置。
- 每一個節點都需要用於資料抄寫的介面。這應該有足夠的頻寬來支援抄寫需求, 因為所有抄寫的資料佇列管理程式都有預期的工作量。

為了達到容錯上限, 此介面應該是獨立「網路介面卡 (NIC)」。

- DRBD 要求用於 RDQM 的每一個節點都具有有效的網際網路主機名稱 (`uname -n` 所傳回的值), 如 RFC 1123 修正的 RFC 952 所定義。
- 如果用於 DR RDQM 的節點之間有防火牆, 則防火牆必須容許埠上用於抄寫的節點之間的資料流量。
- 如果系統在非 `permissive` 模式下使用 SELinux, 則您必須執行下列指令:

```
semanage permissive -a drbd_t
```

網路需求

建議您在不同的資料中心尋找用於災難回復的節點。

您應該瞭解下列限制:

- 效能會隨著資料中心之間的延遲增加而快速降低。IBM 將支援同步抄寫最多 5 毫秒的延遲, 以及非同步抄寫最多 50 毫秒的延遲。
- 透過抄寫鏈結傳送的資料不受限於任何其他加密, 除了使用 IBM MQ AMS 可能已備妥的加密之外。

- 由於需要在兩個 RDQM 節點之間抄寫資料，因此配置 RDQM 佇列管理程式以進行災難回復會產生額外負擔。同步抄寫所產生的額外負擔大於非同步抄寫。使用同步抄寫時，會封鎖磁碟 I/O 作業，直到將資料寫入兩個節點為止。使用非同步抄寫時，必須只將資料寫入主要節點，然後才能繼續處理。

使用佇列管理程式的使用者需求

若要建立、刪除或配置抄寫的資料佇列管理程式 (RDQM)，您必須是 root 使用者，或具有屬於 mqm 群組的使用者 ID，該群組獲授與下列指令的 sudo 權限：

- **crtmqm**
- **dltmqm**
- **rdqmdir**

屬於 mqm 群組的使用者可以使用下列指令來檢視 DR RDQM 的狀態：

- **dspm**
- **rdqmstatus**

Linux V 9.1.0 建立災難回復 RDQM

您可以使用 **crtmqm** 指令來建立抄寫的資料佇列管理程式 (RDQM)，以充當災難回復配置中的主要或次要。

關於這項作業

如果使用者可以使用 **sudo**，您可以在 mqm 群組中以使用者身分建立抄寫的資料佇列管理程式 (RDQM)。否則，您必須以 root 身分建立 RDQM。

您必須在一個節點上建立主要 RDQM DR 佇列管理程式。然後，您必須在另一個節點上建立相同佇列管理程式的次要實例。主要及次要實例必須具有相同的名稱，並配置相同的儲存體數量。

程序

- 如果要建立主要 DR RDQM，請執行下列動作：
 - a) 執行下列指令：

```
crtmqm -rr p [-rt (a | s)] -rl Local_IP -ri Recovery_IP -rn Recovery_Name -ip Port
[other_crtmqm_options] [-fs size] QMname
```

其中：

-rr p

指定您要建立佇列管理程式的主要實例。

-rt a | s

-rt s 指定 DR 配置使用同步抄寫，**-rt a** 指定 DR 配置使用非同步抄寫。非同步抄寫是預設值。

-rl Local_IP

指定要用於此佇列管理程式的 DR 抄寫的本端 IP 位址。

-ri Recovery_IP

指定在管理佇列管理程式次要實例的伺服器上，用於抄寫之介面的 IP 位址。

-rn 回復名稱

指定管理佇列管理程式次要實例的系統名稱。該名稱是您在該伺服器上執行 `uname -n` 時所傳回的值。您必須在該伺服器上明確建立次要佇列管理程式。

-rp 埠

指定用於 DR 抄寫的埠。

other_crtmqm_options

您可以選擇性地指定下列一或多個一般 **crtmqm** 選項：

- z
- q

- -c 文字
- -d *DefaultTransmission* 佇列
- -h *MaxHandles*
- -g *ApplicationGroup*
- -oa 使用者|群組
- -t *TrigInt*
- -u *DeadQ*
- -x *MaxUMsgs*
- -lp *LogPri*
- -ls *LogSec*
- -lc | -l
- -lla | -lln
- _lf *LogFile* 大小
- -p 埠

-fs size

選擇性地指定要為佇列管理程式建立的檔案系統大小，即在 *drbdpool* 磁區群組中建立的邏輯磁區大小。也會建立該大小的另一個邏輯磁區，以支援回復至 Snapshot 作業，因此 DR RDQM 的儲存體總計僅為這裡指定的兩倍。

QMNAME

指定抄寫資料佇列管理程式的名稱。此名稱區分大小寫。

指令完成之後，它會輸出您在次要節點上需要 **tp** 輸入的指令，以建立佇列管理程式的次要實例。您也可以主要節點上使用 **rdqmdr** 指令，來擷取您需要在次要節點上執行才能建立次要佇列管理程式的 **crtmqm** 指令，請參閱第 480 頁的『管理 DR RDQM 的主要及次要性質』。

- 如果要建立次要 DR RDQM，請執行下列動作：

a) 在要管理 RDQM 次要實例的節點上輸入下列指令：

```
crtmqm -rr s [-rt (a | s)] -rl Local_IP -ri Primary_IP -rn Primary_Name -rp Port
[other_crtmqm_options] [-fs size] QMname
```

其中：

-rr s

指定您要建立佇列管理程式的次要實例。

-rt a | s

-rt s 指定 DR 配置使用同步抄寫，**-rt a** 指定 DR 配置使用非同步抄寫。

-rl Local_IP

指定要用於此佇列管理程式的 DR 抄寫的本端 IP 位址。

-ri Primary_IP

在管理佇列管理程式主要實例的伺服器上，指定用於抄寫的介面 IP 位址。

-rn Primary_Name

指定管理佇列管理程式主要實例的系統名稱。該名稱是您在該伺服器上執行 `uname -n` 時所傳回的值。

-rp 埠

指定用於 DR 抄寫的埠。

other_crtmqm_options

您可以選擇性地指定下列一或多個一般 **crtmqm** 選項：

- -z

-fs size

指定要為佇列管理程式建立的檔案系統大小，即在 *drbdpool* 磁區群組中建立的邏輯磁區大小。如果您在建立主要佇列管理程式時已指定非預設大小，則必須在這裡指定相同的值。

QMNAME

指定抄寫資料佇列管理程式的名稱。這必須與您指定給佇列管理程式主要實例的名稱相同。請注意，名稱區分大小寫。

下一步

建立佇列管理程式的主要及次要實例之後，您必須檢查兩個節點上的狀態，以檢查兩者是否正確。在兩個節點上使用 **rdqmstatus** 指令。節點應該會顯示正常狀態，如第 481 頁的『[檢視 DR RDQM 狀態](#)』中所述。如果他們未顯示此狀態，請刪除次要實例並重建它，小心使用正確的引數。

相關參考

[crtmqm](#)

Linux V 9.1.0 刪除 DR RDQM

您可以使用 **dltmqm** 指令來刪除災難回復抄寫的資料佇列管理程式 (RDQM)。

關於這項作業

您必須執行指令來刪除 RDQM 的主要和次要節點上的 RDQM。RDQM 必須先結束。如果 mqm 使用者具有必要的 sudo 專用權，則可以該使用者身分執行指令。否則，您必須以 root 身分執行指令。

程序

- 若要刪除 DR RDQM，請輸入下列指令：

```
dltmqm RDQM_name
```

相關參考

[dltmqm](#)

MQ Adv. Linux 移轉佇列管理程式以變成 DR RDQM 佇列管理程式

您可以透過備份現有佇列管理程式的持續資料，然後將資料還原至新建立且具有相同名稱的 RDQM 佇列管理程式，將現有佇列管理程式移轉為災難回復 (DR) 抄寫的資料佇列管理程式 (RDQM)。

關於這項作業

「DR 抄寫資料佇列管理程式」需要專用邏輯磁區 (檔案系統) 及磁碟抄寫的配置。只有在建立新的佇列管理程式時，才會配置這些元件。可以移轉現有佇列管理程式以使用 RDQM，方法是備份其持續資料，然後將資料還原至新建立且具有相同名稱的 RDQM 佇列管理程式。此程序會保留建立備份時的佇列管理程式配置、狀態及持續訊息。

註：您只能從與 RDQM 安裝版本相同或更低的 IBM MQ 版本移轉佇列管理程式。作業系統和架構也必須相同。否則，您必須在目標平台上建立新的佇列管理程式，請參閱 [將佇列管理程式移至不同的作業系統](#)。

在移轉佇列管理程式之前，您應該滿足下列條件：

- 評估您的災難回復需求，並參閱第 470 頁的『[RDQM 災難回復](#)』。
- 檢閱連接至佇列管理程式的應用程式及佇列管理程式。請考量將連線遞送至佇列管理程式執行所在之 RDQM 節點所需的變更。
- 為您選擇的配置佈建或識別現有的 RDQM 節點。如需 RDQM 系統需求的相關資訊，請參閱第 472 頁的『[RDQM DR 解決方案的需求](#)』。
- 在每一個節點上安裝 IBM MQ Advanced，其中包括 RDQM 特性。
- 選擇性地使用測試佇列管理程式來驗證 RDQM 配置，然後可以刪除該配置。建議測試配置，以在移轉佇列管理程式之前識別並解決任何問題。
- 請檢閱佇列管理程式的安全配置，然後在每一個 RDQM 節點上抄寫必要的本端使用者和群組。
- 請檢閱佇列管理程式及通道配置，以判斷是否使用 API 結束程式、通道結束程式或資料轉換結束程式。在每一個 RDQM 節點上安裝必要的結束程式。
- 請檢閱已定義的任何佇列管理程式服務，然後在每一個 RDQM 節點上安裝並配置必要的程序。

程序

1. 備份現有的佇列管理程式:

- a) 透過發出等待關閉指令 `endmqm -w` 或立即關閉指令 `endmqm -i` 來停止現有佇列管理程式。此步驟很重要，可確保備份中的資料一致。
- b) 透過檢視 IBM MQ 配置檔 `mqs.ini`，判定佇列管理程式資料目錄的位置。在 Linux 上，此檔案位於 `/var/mqm` 目錄中。如需 `mqs.ini` 的相關資訊，請參閱第 74 頁的『IBM MQ 配置檔 `mqs.ini`』。
在檔案中找出佇列管理程式的 `QueueManager` 段落。如果段落包含名為 `DataPath` 的索引鍵，則其值為佇列管理程式資料目錄。如果金鑰不存在，則可以使用 `Prefix` 及 `Directory` 金鑰的值來判定佇列管理程式資料目錄。佇列管理程式資料目錄是這些值的連結，格式為 `prefix/qmgrs/directory`。如需 `QueueManager` 段落的相關資訊，請參閱第 83 頁的『`mqs.ini` 檔案的 `QueueManager` 段落』。
- c) 建立佇列管理程式資料目錄的備份。在 Linux 上，您可以使用 `tar` 指令來執行此動作。例如，若要備份佇列管理程式的資料目錄，您可以使用下列指令。請注意指令的最後一個參數，它是單一句點 (點):

```
tar -cvzf qm-data.tar.gz -C queue_manager_data_dir .
```

- d) 檢視 IBM MQ 佇列管理程式配置檔 `qm.ini`，以判斷佇列管理程式日誌目錄的位置。此檔案位於佇列管理程式資料目錄中。如需檔案的相關資訊，請參閱第 84 頁的『佇列管理程式配置檔 `qm.ini`』。
佇列管理程式日誌目錄定義為 `Log` 段落中 `LogPath` 索引鍵的值。如需段落的相關資訊，請參閱第 108 頁的『`qm.ini` 檔案的日誌段落』。
- e) 建立佇列管理程式日誌目錄的備份。在 Linux 上，您可以使用 `tar` 指令來執行此動作。例如，若要備份佇列管理程式的日誌目錄，您可以使用下列指令。請注意指令的最後一個參數，它是單一句點 (點):

```
tar -cvzf qm-log.tar.gz -C queue_manager_log_dir .
```

- f) 如果佇列管理程式使用的任何憑證儲存庫不在佇列管理程式資料目錄中，請建立它們的備份。請確定已備份金鑰資料庫檔和密碼隱藏檔。如需佇列管理程式金鑰儲存庫的相關資訊，請參閱 [SSL/TLS 金鑰儲存庫](#) 及 [尋找佇列管理程式的金鑰儲存庫](#)。如需在佇列管理程式配置為使用「AMS 訊息通道代理程式 (MCA)」截取時尋找 AMS 金鑰儲存庫的相關資訊，請參閱 [訊息通道代理程式 \(MCA\) 截取](#)。
- g) 不再需要現有的佇列管理程式，因此可以刪除它。不過，可能的話，在目標系統上順利還原現有的佇列管理程式之後，您應該只刪除它。延遲刪除可確保在移轉處理程序未順利完成時，佇列管理程式可以重新啟動。

註: 如果您延遲刪除現有的佇列管理程式，請不要重新啟動它。佇列管理程式必須保留結束狀態，因為在移轉期間會遺失對其配置或狀態所做的進一步變更。

2. 準備主要 RDQM 節點:

- a) 建立新的 RDQM 佇列管理程式，其名稱與您所備份的佇列管理程式相同。請確定 `crtmqm` 配置給 RDQM 佇列管理程式的檔案系統夠大，足以包含現有佇列管理程式的資料、主要日誌和次要日誌，以及一些未來擴充的額外空間。如需如何建立 RDQM 佇列管理程式的相關資訊，請參閱第 473 頁的『[建立災難回復 RDQM](#)』。
- b) 判斷佇列管理程式的主要 RDQM 節點。如需如何判斷主要節點的相關資訊，請參閱 [rdqmstatus](#) (顯示 RDQM 狀態)。
- c) 在主要 RDQM 節點上，如果 RDQM 佇列管理程式已啟動，請使用 `endmqm -w` 或 `endmqm -i` 指令停止它。
- d) 判斷 RDQM 佇列管理程式的資料和日誌目錄位置 (請使用步驟 1b 和 1d 中說明的方法)。
- e) 刪除 RDQM 佇列管理程式資料和日誌目錄的內容，但不刪除目錄本身。

3. 還原主要 RDQM 節點上的佇列管理程式:

- a) 將佇列管理程式資料及日誌目錄的備份複製到主要 RDQM 節點，以及佇列管理程式所使用憑證儲存庫的任何個別備份。
- b) 將佇列管理程式資料目錄的備份還原至新的 RDQM 佇列管理程式的空資料目錄，以確保保留檔案所有權及許可權。如果是使用步驟 1c 中的範例 `tar` 指令來建立備份，則 `root` 使用者可以使用下列指令來還原它:

```
tar -xvzpf qm-data.tar.gz -C queue_manager_data_dir
```

- c) 將佇列管理程式日誌目錄的備份還原至新的 RDQM 佇列管理程式的空白日誌目錄，以確保保留檔案所有權和許可權。如果備份是使用步驟 1e 中的範例 tar 指令來建立，則 root 使用者可以使用下列指令來還原它：

```
tar -xvzpf qm-log.tar.gz -C queue_manager_log_dir
```

- d) 編輯 RDQM 佇列管理程式之資料目錄中的已還原佇列管理程式配置檔 `qm.ini`。更新 Log 段落中 LogPath 索引鍵的值，以指定 RDQM 佇列管理程式的日誌目錄。

請檢閱配置檔中定義的其他檔案路徑，並在必要時更新它們。例如，您可能需要更新下列路徑：

- 診斷訊息服務所產生錯誤日誌檔的路徑。
 - 佇列管理程式所需的結束程式路徑。
 - 如果佇列管理程式是 XA 交易協調程式，則為交換器載入檔的路徑。
- e) 如果佇列管理程式配置為使用「AMS 訊息通道代理程式 (MCA)」截取，請將 AMS 金鑰儲存庫複製到新的 RDQM 安裝架構，然後檢閱並更新配置。金鑰儲存庫必須在每一個 RDQM 節點上可用，因此如果它不在佇列管理程式的已抄寫檔案系統中，則必須改為複製到每一個節點。如需相關資訊，請參閱訊息通道代理程式 (MCA) 截取。
- f) 驗證 `dspmq` 指令是否顯示佇列管理程式，且其狀態報告為已結束。下列範例顯示 RDQM DR 佇列管理程式的範例輸出：

```
$ dspmq -o status -o dr
QMNAME(QM1) STATUS(Ended normally) DRROLE(Primary)
```

- g) 使用 `rdqmstatus` 指令來顯示佇列管理程式的狀態，以驗證已還原的佇列管理程式資料已抄寫至次要 RDQM 節點。在每一個節點上，DR 狀態應該報告為 Normal。下列範例顯示 RDQM DR 佇列管理程式的範例輸出：

```
$ rdqmstatus -m QM1
Queue manager status:           Ended normally
Queue manager file system:      51MB used, 1.0GB allocated [5%]
DR role:                         Primary
DR status:                       Normal
DR type:                         Synchronous
DR port:                         3000
DR local IP address:             192.168.20.1
DR remote IP address:           192.168.20.2
```

- h) 在主要 RDQM 節點上啟動佇列管理程式。

- i) 連接至佇列管理程式，並更新 SSLKEYR 佇列管理程式屬性的值，以指定佇列管理程式憑證儲存庫的新位置。依預設，此屬性的值設為 `queue_manager_data_directory/ssl/key`。憑證儲存庫必須位於每一個 RDQM 節點上的相同位置。如果儲存庫不在佇列管理程式的抄寫檔案系統中，則必須改為將它複製到每一個節點。

- j) 檢閱佇列管理程式的 IBM MQ 物件定義，並更新參照已變更網路設定、IBM MQ 安裝目錄或佇列管理程式資料目錄 (包括下列物件) 的物件屬性值：

- 接聽器使用的本端 IP 位址 (IPADDR 屬性)。
- 通道使用的本端 IP 位址 (LOCLADDR 屬性)。
- 定義給叢集接收端通道的本端 IP 位址 (CONNAME 屬性)。
- 為通訊資訊物件定義的本端 IP 位址 (GRPADDR 屬性)。
- 針對處理程序及服務物件定義定義的系統路徑。

- k) 停止並重新啟動佇列管理程式，以確保變更生效。

- l) 針對連接至已移轉佇列管理程式的遠端佇列管理程式，重複步驟 3j，以及應用程式的對等設定，包括：

- 通道連線名稱 (CONNAME 屬性)。
- 通道鑑別規則，根據佇列管理程式的 IP 位址或主機名稱來限制來自佇列管理程式的入埠連線。

- 用戶端通道定義表 (CCDT)、網域名稱設定 (DNS)、網路遞送或對等連線資訊。
- m) 對每一個 RDQM 節點執行佇列管理程式的受管理失效接手，以確保已順利建立必要的配置，請參閱 [第 484 頁的『切換至回復節點』](#)。

調整 DR RDQM 佇列管理程式的檔案系統大小

若要調整現有災難回復 (DR) 抄寫資料佇列管理程式 (RDQM) 的檔案系統大小，您可以備份其持續資料，然後將資料還原至名稱相同但檔案系統大小不同的新建立 RDQM 佇列管理程式。

關於這項作業

DR 抄寫資料佇列管理程式需要專用邏輯磁區 (檔案系統) 及磁碟抄寫的配置。只有在建立新的佇列管理程式時，才會配置這些元件。檔案系統在建立之後無法調整大小，因為它在每一個節點上必須具有相同的大小。若要調整現有抄寫資料佇列管理程式 (RDQM) 的檔案系統大小，您可以備份其持續資料，然後將資料還原至名稱相同但檔案系統大小不同的新建立 RDQM 佇列管理程式。此程序會保留建立備份時的佇列管理程式配置、狀態及持續訊息。

程序

1. 備份主要 RDQM 節點上的現有 RDQM 佇列管理程式:

- a) 判斷佇列管理程式的主要 RDQM 節點。如需如何判斷主要節點的相關資訊，請參閱 [rdqmstatus \(顯示 RDQM 狀態\)](#)。
- b) 在主要 RDQM 節點上，如果 RDQM 佇列管理程式已啟動，請使用 `endmqm -w` 或 `endmqm -i` 指令停止它。
- c) 透過檢視 IBM MQ 配置檔 `mqs.ini`，判定佇列管理程式資料目錄的位置。在 Linux 上，此檔案位於 `/var/mqm` 目錄中。如需 `mqs.ini` 的相關資訊，請參閱 [第 74 頁的『IBM MQ 配置檔 mqs.ini』](#)。

在檔案中找出佇列管理程式的 `QueueManager` 段落。佇列管理程式資料目錄是名為 `DataPath` 的索引鍵值。如需 `QueueManager` 段落的相關資訊，請參閱 [第 83 頁的『mqs.ini 檔案的 QueueManager 段落』](#)。

- d) 建立佇列管理程式資料目錄的備份。在 Linux 上，您可以使用 `tar` 指令來執行此動作。例如，若要備份佇列管理程式的資料目錄，您可以使用下列指令。請注意指令的最後一個參數，它是單一句點 (.) 字元:

```
tar -cvzf qm-data.tar.gz -C queue_manager_data_dir .
```

- e) 檢視 IBM MQ 佇列管理程式配置檔 `qm.ini`，以判斷佇列管理程式日誌目錄的位置。此檔案位於佇列管理程式資料目錄中。如需檔案的相關資訊，請參閱 [第 84 頁的『佇列管理程式配置檔 qm.ini』](#)。

佇列管理程式日誌目錄定義為日誌段落中 `LogPath` 索引鍵的值。如需段落的相關資訊，請參閱 [第 108 頁的『qm.ini 檔案的日誌段落』](#)。

- f) 建立佇列管理程式日誌目錄的備份。在 Linux 上，您可以使用 `tar` 指令來執行此動作。例如，若要備份佇列管理程式的日誌目錄，您可以使用下列指令。請注意指令的最後一個參數，它是單一句點 (.) 字元:

```
tar -cvzf qm-log.tar.gz -C queue_manager_log_dir .
```

- g) 刪除現有的 RDQM 佇列管理程式。

2. 使用所需大小的檔案系統還原佇列管理程式:

- a) 建立新的 RDQM 佇列管理程式，其名稱與您所備份的佇列管理程式相同。請確定透過 `crtmqm` 配置給 RDQM 佇列管理程式的檔案系統是您需要的大小，而且它已夠大，足以包含現有佇列管理程式的資料、主要日誌及次要日誌，以及一些未來擴充的額外空間。如需如何建立 RDQM 佇列管理程式的相關資訊，請參閱 [第 473 頁的『建立災難回復 RDQM』](#)。
- b) 判斷佇列管理程式的主要 RDQM 節點。如需如何判斷主要節點的相關資訊，請參閱 [rdqmstatus \(顯示 RDQM 狀態\)](#)。
- c) 在主要 RDQM 節點上，如果 RDQM 佇列管理程式已啟動，請使用 `endmqm -w` 或 `endmqm -i` 指令停止它。

- d) 在主要 RDQM 節點上，判斷 RDQM 佇列管理程式之資料和日誌目錄的新位置 (使用步驟 1c 和 1e 中說明的方法)。
- e) 在主要 RDQM 節點上，刪除 RDQM 佇列管理程式資料和日誌目錄的內容，但不刪除目錄本身。
- f) 在主要 RDQM 節點上，將佇列管理程式資料目錄的備份還原至新的 RDQM 佇列管理程式的空資料目錄，以確保保留檔案所有權及許可權。如果是使用步驟 1d 中的範例 **tar** 指令來建立備份，則 root 使用者可以使用下列指令來還原它：

```
tar -xvzpf qm-data.tar.gz -C queue_manager_data_dir
```

- g) 在主要 RDQM 節點上，將佇列管理程式日誌目錄的備份還原至新的 RDQM 佇列管理程式的空白日誌目錄，以確保保留檔案所有權和許可權。如果是使用步驟 1f 中的範例 **tar** 指令來建立備份，則 root 使用者可以使用下列指令來還原它：

```
tar -xvzpf qm-log.tar.gz -C queue_manager_log_dir
```

- h) 在主要 RDQM 節點上，編輯新 RDQM 佇列管理程式之資料目錄中的已還原佇列管理程式配置檔 **qm.ini**。更新 Log 段落中 LogPath 索引鍵的值，以指定您在步驟 2d 中判定的新 RDQM 佇列管理程式的日誌目錄。請檢閱配置檔中定義的其他檔案路徑，並在必要時更新它們。例如，您可能需要更新下列路徑：

- 診斷訊息服務所產生錯誤日誌檔的路徑。
- 佇列管理程式所需的結束程式路徑。
- 如果佇列管理程式是 XA 交易協調程式，則為交換器載入檔的路徑。

- i) 驗證 **dspmqr** 指令是否顯示佇列管理程式，以及其狀態是否報告為 **ended**。下列範例顯示 RDQM DR 佇列管理程式的範例輸出：

```
$ dspmqr -o status -o dr
QMNAME(QM1) STATUS(Ended normally) DR(Primary)
```

- j) 使用 **rdqmstatus** 指令來顯示佇列管理程式的狀態，以驗證還原的佇列管理程式資料已抄寫至次要 RDQM 節點。在每一個節點上，DR 狀態應該報告為 **Normal**。下列範例顯示主要節點上 RDQM DR 佇列管理程式的範例輸出：

```
$ rdqmstatus -m QM1
Queue manager status:      Running
CPU:                       0.00
Memory:                    123MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
DR role:                   Primary
DR status:                 Normal
DR type:                   Synchronous
DR port:                   3000
DR local IP address:       192.168.20.1
DR remote IP address:     192.168.20.2
```

下列範例顯示回復節點上 RDQM DR 佇列管理程式的輸出範例：

```
Queue manager status:      Ended immediately
DR role:                   Secondary
DR status:                 Normal
DR port:                   3000
DR local IP address:       192.168.20.2
DR remote IP address:     192.168.20.1
```

- k) 在主要 RDQM 節點上啟動佇列管理程式。
- l) 執行將佇列管理程式切換至回復節點，以確定已順利建立必要的配置，請參閱 [第 484 頁的『切換至回復節點』](#)。

V 9.1.5 儲存持續性應用程式狀態

您可以儲存與應用程式相關的持續狀態資訊，以及其他佇列管理程式資料。

每一個 IBM MQ 佇列管理程式都有其持續狀態的專用檔案系統，其中包括其佇列資料及回復日誌。在 RDQM 配置中，檔案系統由在 Linux 系統 (節點) 之間抄寫的邏輯磁區所支援。檔案系統包括 **userdata** 目錄，您可以用來儲存應用程式的持續狀態資訊。因此，當抄寫的資料佇列管理程式移至 RDQM 配置中的另一個節

點上執行時，您有可用的應用程式環境定義及佇列管理程式環境定義。請參閱 [Unix 及 Linux 系統上的目錄內容](#)。

如果您選擇將應用程式狀態儲存在 `userdata` 目錄中，則必須注意寫入此位置的資料可能會耗用配置給佇列管理程式的可用磁碟空間。您必須確保仍有足夠的磁碟空間可供佇列管理程式寫入佇列資料、日誌及其他持續性狀態資訊。

`userdata` 目錄具有 `mqm` 使用者和群組所有權，而且是全球可讀取的，因此使用者可以存取它，而不需要在 IBM MQ 管理者群組中 (即 `mqm`)。您無法修改 `userdata` 目錄的許可權，但可以使用您需要的任何所有權和許可權在其中建立內容。

在 RDQM 佇列管理程式失效接手期間，佇列管理程式會結束，且其現行 RDQM 節點上的檔案系統會卸載。然後會裝載檔案系統，並在 RDQM 配置中的另一個節點上重新啟動佇列管理程式。如果處理程序的其中一個檔案具有開啟控點，則無法卸載檔案系統。為了確保佇列管理程式失效接手可以完成，如果無法卸載佇列管理程式檔案系統，則會傳送 `SIGTERM` 信號給具有開啟檔案控點的處理程序，如果未釋放開啟控點，則會接著 `SIGKILL`。您的應用程式必須設計成正確回應 `SIGTERM`。如果應用程式或處理程序配置為佇列管理程式服務，則在受管理失效接手期間，可以在解除裝載檔案系統之前，在佇列管理程式關閉期間結束它們。如果應用程式或處理程序未配置為佇列管理程式服務，或發生未受管理的失效接手 (例如失去額定)，則可能會傳送信號以釋放檔案系統。

Linux V 9.1.0 管理 DR RDQM 的主要及次要性質

您可以將次要災難回復抄寫資料佇列管理程式 (DR RDQM) 變更為主要 DR RDQM。您也可以將主要實例變更為次要實例。

關於這項作業

您可以使用 `rdqmdr` 指令，將 RDQM 的次要實例變更為主要實例。如果您因某些原因而失去主要實例，則可能需要完成此動作。然後，您可以啟動佇列管理程式，並在回復節點上繼續執行它。

您也可以使用 `rdqmdr` 指令，將 RDQM 的主要實例變更為次要實例。您可能需要完成此動作，例如，如果您正在重新配置系統。

您也可以在主要佇列管理程式上使用 `rdqmdr`，以擷取在回復節點上建立該佇列管理程式的次要實例所需的確切指令。

如果使用者可以使用 `sudo`，您可以在 `mqm` 群組中以使用者身分使用 `rdqmdr` 指令。否則，您必須以 `root` 身分登入。

程序

- 若要將 DR RDQM 的次要實例變更為主要實例，請輸入下列指令：

```
rdqmdr -m QMname -p
```

如果佇列管理程式的主要實例仍在執行中，且 DR 抄寫鏈結仍在運作中，則此指令會失敗。

- 若要將佇列管理程式的主要實例變更為次要實例，請輸入下列指令：

```
rdqmdr -m QMname -s
```

- 若要顯示配置佇列管理程式的次要實例所需的 `crtmqm` 指令，請在主要節點上輸入下列指令：

```
rdqmdr -d -m QMname
```

您可以在次要節點上輸入傳回的 `crtmqm` 指令，以建立 RD RDQM 的次要實例。

Linux V 9.1.0 啟動、停止及顯示 DR RDQM 的狀態

您可以使用標準 IBM MQ 控制指令的變式，來啟動、停止及檢視災難回復抄寫資料佇列管理程式 (DR RDQM) 的現行狀態。

關於這項作業

您必須以屬於 mqm 群組的使用者身分執行指令，以啟動、停止及檢視已抄寫資料佇列管理程式 (RDQM) 的現行狀態。

您必須執行指令來啟動和停止該佇列管理程式之主要節點上的佇列管理程式 (亦即，佇列管理程式目前執行所在的節點)。

程序

- 若要啟動 DR RDQM，請在 RDQM 的主要節點上輸入下列指令：

```
strmqm qmname
```

其中 *qmname* 是您要啟動之 RDQM 的名稱。

- 若要停止 RDQM，請在 RDQM 的主要節點上輸入下列指令：

```
endmqm qmname
```

其中 *qmname* 是您要停止之 RDQM 的名稱。

- 如果要檢視 RDQM 的狀態，請輸入下列指令：

```
dspmq -m QMname
```

輸出的狀態資訊取決於您是在 RDQM 的主要或次要節點上執行指令。如果在主要節點上執行，則會顯示 **dspmq** 所傳回的其中一個正常狀態訊息。如果您在次要節點上執行指令，則會顯示狀態 Ended immediately。例如，如果在節點 RDQM7 上執行 **dspmq**，則可能會傳回下列資訊：

```
QMNAME(DRQM8)          STATUS(Ended immediately)
QMNAME(DRQM7)          STATUS(Running)
```

您可以搭配使用引數與 **dspmq**，以建立 RDQM 是否配置用於災難回復，以及它目前是主要或次要實例：

```
dspmq -m QMname -o (dr | DR)
```

會顯示下列其中一個回應：

DRROLE()

指出佇列管理程式未配置災難回復。

DRROLE(Primary)

指出佇列管理程式已配置為 DR 主要。

DRROLE(Secondary)

指出佇列管理程式已配置為 DR 次要。

相關參考

[dspmq](#)

[endmqm](#)

[strmqm](#)

Linux V 9.1.0 檢視 DR RDQM 狀態

您可以檢視節點上所有災難回復抄寫資料佇列管理程式 (DR RDQ) 的狀態，或指定 DR RDQM 的詳細資訊。

關於這項作業

您可以使用 **rdqmstatus** 指令來檢視所有 DR RDQM 或個別 RDQM 的狀態。

您必須是 mqm 群組中的使用者，才能執行 **rdqmstatus** 指令。您可以在 DR RDQM 配對的任一節點上執行指令。

程序

- 若要檢視節點上所有 DR RDQM 的狀態，請在該節點上執行下列指令：

```
rdqmstatus
```

即會顯示節點上 DR RDQM 的狀態，例如：

```
Queue manager name:      DRQM8
Queue manager status:   Ended immediately
DR role:                Secondary

Queue manager name:      DRQM7
Queue manager status:   Running
DR role:                Primary
```

- 如果要檢視特定 RDQM 的狀態，請輸入下列指令：

```
rdqmstatus -m qmname
```

下表彙總所傳回的資訊。

表 35: 狀態屬性		
「狀態」屬性	可能的值	顯示時
佇列管理程式狀態	狀態 (如 dspmq 所顯示)	一律顯示
CPU	<i>n.nn%</i>	僅當現行節點上的 RDQM 具有主要角色時才顯示
記憶體	<i>nnn</i> MB	僅當現行節點上的 RDQM 具有主要角色時才顯示
佇列管理程式檔案系統	<i>nnn</i> 已使用 MB, <i>n</i> 。 <i>n</i> 已配置 GB [<i>n%</i>]	僅當現行節點上的 RDQM 具有主要角色時才顯示
DR 角色	主要 次要 不明	一律顯示
DR 狀態	正常	一般作業
	正在進行同步化	正在進行同步化
	已分割	當 DR 抄寫網路無法使用時，已在兩個節點上啟動佇列管理程式
	遠端系統無法使用	已失去與另一個節點的連線
	不一致	正在進行同步化，但已岔斷
	正在回復至 Snapshot	使用者已選擇回復佇列管理程式進入不一致狀態時所產生的 Snapshot。
	未配置遠端系統	已配置 RDQM 的主要實例，但未配置次要實例
	協議失敗	其中一個節點已設為同步抄寫，另一個已設為非同步抄寫
DR 類型	同步或非同步	一律顯示
DR 埠	<i>port_number</i> (用來抄寫此佇列管理程式資料的 TCP/IP 埠)	一律顯示

表 35: 狀態屬性 (繼續)		
「狀態」屬性	可能的值	顯示時
DR 本端 IP 位址	此佇列管理程式正在從中抄寫 DR 的本端 IP 位址	一律顯示
DR 遠端 IP 位址	針對 DR, 此佇列管理程式正在抄寫的遠端 IP 位址	一律顯示
DR 不同步資料	nKB	當遠端節點無法使用或不一致時顯示
DR 同步化進行中	n%	同步化進行中時顯示
DR 預估完成時間	YYYY-MM-DD HH:MM:SS	同步化進行中時顯示
Snapshot 回復進度	n%	當 DR 狀態為 Reverting to snapshot 時顯示。狀態計數已關閉, 因此 0% 顯示完成

範例

主要節點上正常狀態的範例:

```
Queue manager status:      Running
CPU:                      0.00
Memory:                   123MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
DR role:                   Primary
DR status:                 Normal
DR type:                   Synchronous
DR port:                   3000
DR local IP address:       192.168.20.1
DR remote IP address:      192.168.20.2
```

次要節點上正常狀態的範例:

```
Queue manager status:      Ended immediately
DR role:                   Secondary
DR status:                 Normal
DR port:                   3000
DR local IP address:       192.168.20.2
DR remote IP address:      192.168.20.1
```

正在進行同步化時主要節點上的狀態範例:

```
Queue manager status:      Running
CPU:                      0.53
Memory:                   124MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
DR role:                   Primary
DR status:                 Synchronization in progress
DR type:                   Synchronous
DR port:                   3000
DR local IP address:       192.168.20.1
DR remote IP address:      192.168.20.2
DR synchronization progress: 11.0%
DR estimated time to completion: 2017-09-06 14:55:05
```

主要節點的範例, 顯示它已分割:

```
Queue manager status:      Running
CPU:                      0.02
Memory:                   124MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
DR role:                   Primary
DR status:                 Partitioned
DR type:                   Synchronous
DR port:                   3000
```

```
DR local IP address: 192.168.20.1
DR remote IP address: 192.168.20.2
```

相關參考

Linux `rdqmstatus`

Linux V 9.1.0 在災難回復環境中運作

在許多狀況下，您可能想要在災難回復配置中切換至次要佇列管理程式。

災難回復

在主要網站上的主要佇列管理程式完全遺失之後，您會在回復網站上啟動次要佇列管理程式。應用程式會重新連接至回復網站上的佇列管理程式，而次要佇列管理程式會處理應用程式訊息。回復至先前配置所採取的步驟，視失敗原因而定。例如，完全遺失主要節點與暫時遺失。

如需在暫時失去主要網站之後採取的步驟，請參閱第 484 頁的『[切換至回復節點](#)』。如需採取下列永久失敗的步驟，請參閱第 485 頁的『[更換災難回復配置中的故障節點](#)』。

災難回復測試支援

您可以暫時切換至次要實例，並檢查應用程式是否可以順利連接，以測試災難回復配置。您遵循與在主要節點暫時故障之後切換時相同的程序，請參閱第 484 頁的『[切換至回復節點](#)』。

正在回復至 Snapshot

在進行同步化時，如果您在主要節點中遇到失敗，您可以回復到在啟動同步化之前取得的次要佇列管理程式資料 Snapshot。然後，次要伺服器會還原至一致狀態，且可以作為主要伺服器執行。若要回復至 Snapshot，請將次要變成主要，如第 484 頁的『[切換至回復節點](#)』中所述。在啟動佇列管理程式之前，您必須先檢查回復至 Snapshot 是否已完成 (使用 `rdqmstatus` 指令)。

Linux V 9.1.0 切換至回復節點

如果主要站台中發生災難，您可以採取步驟來切換至回復站台。

關於這項作業

在失去主要站台上的主要佇列管理程式之後，您會將回復站台上的次要佇列管理程式設為主要佇列管理程式並啟動它。應用程式會重新連接至回復網站上的佇列管理程式，且佇列管理程式會處理應用程式訊息。您也可以使用此程序來測試回復節點。

重要: 在提升原始次要實例之前，您必須確定佇列管理程式的主要實例已停止並變成次要實例。否則，分割的資料可能會增加。

您必須以 root 身分登入，或以屬於 mqm 群組且具有必要 sudo 配置的使用者身分登入。

程序

1. 如果您使用此程序來測試次要佇列管理程式 (亦即，主要實例仍在執行中)，則必須停止主要實例，並將它重新指定為次要實例:

```
endmqm qmname
rdqmdr -m qmname -s
```

2. 在回復節點上輸入下列指令，使次要佇列管理程式成為主要佇列管理程式:

```
rdqmdr -m qmname -p
```

3. 輸入下列指令，以啟動佇列管理程式:

```
strmqm qmname
```

4. 請確定您的應用程式重新連接回復佇列管理程式上的佇列管理程式。如果您已使用替代連線名稱清單來定義通道，並指定主要及次要佇列管理程式，則您的應用程式會自動連接至新的主要佇列管理程式。

相關參考

[strmqm](#)

[rdqmdr](#)

如果您遺失災難回復配置中的其中一個節點，您可以遵循此程序來更換節點並還原災難回復配置。

關於這項作業

如果發生災難，導致主要站台中的節點無法修復，您可以在佇列管理程式在回復節點上執行時取代失敗節點，然後還原原始災難回復配置。替換用節點必須採用失敗節點的身分：名稱和 IP 位址必須相同。

您必須以 root 身分登入，或以屬於 mqm 群組且具有必要 sudo 配置的使用者身分登入。

程序

在失去主要網站上的佇列管理程式之後，請採取下列步驟：

1. 在回復節點上，執行下列指令，讓次要佇列管理程式擔任主要角色：

```
rdqmdr -m QMname -p
```

其中 *qmname* 是佇列管理程式的名稱。

2. 擷取您需要在替換用主要節點上執行的指令，以重新配置災難回復：

```
rdqmdr -m QMname -d
```

複製此指令的輸出。

3. 執行下列指令，以啟動佇列管理程式：

```
strmqm QMname
```

4. 請確定您的應用程式重新連接回復節點上的佇列管理程式。如果您已使用替代連線名稱清單來定義通道，並指定主要及次要佇列管理程式，則您的應用程式會自動連接至新的主要佇列管理程式。
5. 更換主要站台上的故障節點，並將其配置為具有原始節點上用於災難回復的相同名稱及 IP 位址。然後執行您在步驟 2 中複製的 **crtmqm** 指令來配置災難回復。現在您有佇列管理程式的次要實例，主要實例會將其資料與次要實例同步化。
6. 結束現行主要實例。
7. 同步化完成之後，讓在回復節點上執行的主要實例再次進入次要實例：

```
rdqmdr -m QMname -s
```

8. 在取代主要節點上，將佇列管理程式的次要實例設為主要實例：

```
rdqmdr -m QMname -p
```

9. 在取代主要節點上，啟動佇列管理程式：

```
strmqm QMname
```

您現在已還原配置，如同主要網站失敗之前一樣。

相關參考

[strmqm](#)

[rdqmdr](#)

[endmqm](#)

解決 DR RDQM 中的不一致問題

如果佇列管理程式的主要和次要實例之間的同步化失敗，則可以報告 inconsistent 的 DR 狀態。

關於這項作業

佇列管理程式的次要實例會報告不一致狀態，因為在同步化作業期間，主要實例的抄寫連線會遺失。您可能必須採取動作來解決此狀況。請考量下列事件順序：

1. DR 主要佇列管理程式與 DR 次要佇列管理程式同步
2. 主要與次要之間的抄寫鏈結遺失
3. 已在主要與次要之間還原抄寫鏈結
4. 當 DR 次要佇列管理程式捕捉到 DR 主要佇列管理程式時，即會進行重新同步。在此期間，會針對兩個佇列管理程式報告 `synchronization in progress` 的 DR 狀態。
5. 如果在重新同步期間再次遺失抄寫，則 DR 次要上的狀態會報告為 `Inconsistent`。

如果管理主要佇列管理程式的節點仍可運作，且可以還原抄寫鏈結，則會自動重新同步。在您不採取任何動作的情況下，會解決不一致狀態。

如果管理主要佇列管理程式的節點不再運作，您可以在次要佇列管理程式上實作回復 Snapshot，以解決不一致狀態。此作業會將資料回復至前次已知良好狀態。

程序

如果要解決不一致狀態，請執行下列動作：

1. 在回復節點上，將次要實例設為主要實例：

```
rdqmdr -m qmname -p
```

即會啟動回復至 Snapshot 作業。

2. 在回復節點上，檢查佇列管理程式的狀態，以查看回復至 Snapshot 作業何時完成：

```
rdqmstatus -m qmname
```

3. 當佇列管理程式狀態為 `Normal` 時，啟動佇列管理程式：

```
stimqm qmname
```

解決 DR RDQM 中的分割 (核心分割) 問題

如果災難回復配對中的兩個佇列管理程式同時以主要角色執行，則可能會發生分割問題。

關於這項作業

如果您在回復節點上升級佇列管理程式的次要實例，而原始主要實例繼續在主要節點上執行，則實際上會有兩個版本的相同佇列管理程式在執行中，每一個都有自己的佇列管理程式資料視圖。每一個節點上佇列管理程式的 DR 狀態報告為 `Partitioned`。

您必須決定哪一個佇列管理程式具有最正確的資料視圖，並在捨棄另一個佇列管理程式時保留該集合。您可以使用 `rdqmdr` 指令來完成這項作業。

程序

- 如果要保留回復節點上佇列管理程式中的資料，請執行下列動作：

- a) 請確定這兩個佇列管理程式實例都已停止。
- b) 指定主要節點上的佇列管理程式是次要：

```
rdqmdr -m qmname -s
```

- c) 指定回復節點上的佇列管理程式是主要：

```
rdqmdr -m qmname -p
```

同步化開始，從回復節點上佇列管理程式的資料複製到主要節點。

d) 檢查同步化的狀態:

```
rdqmstatus -m qmname
```

e) 同步化完成時，將回復節點上的佇列管理程式降級:

```
rdqmdr -m qmname -s
```

f) 提升主要節點上的佇列管理程式，並啟動它:

```
rdqmdr -m qmname -p  
stimqm qmname
```

• 若要保留主要節點上佇列管理程式中的資料，請執行下列動作:

a) 請確定這兩個佇列管理程式實例都已停止。

b) 指定回復節點上的佇列管理程式是次要:

```
rdqmdr -m qmname -s
```

c) 指定主要節點上的佇列管理程式是主要:

```
rdqmdr -m qmname -p
```

同步化開始，主要節點上的佇列管理程式資料會複製到回復節點。

d) 檢查同步化的狀態:

```
rdqmstatus -m qmname
```

e) 同步化完成時，請在主要節點上啟動佇列管理程式:

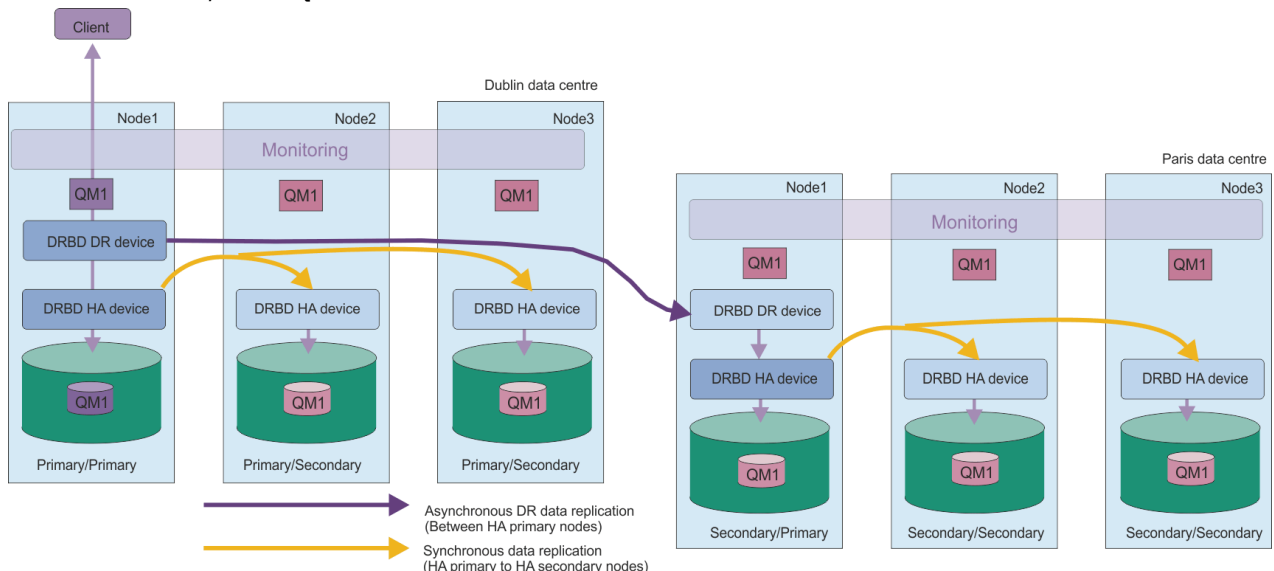
```
stimqm qmname
```

Linux V 9.1.5 RDQM 災難回復及高可用性

您可以配置抄寫的資料佇列管理程式 (RDQM)，以在一個站台的高可用性群組上執行，但如果發生災難，導致第一個群組無法使用，則可以由另一個站台的另一個高可用性群組進行失效接手。這稱為 DR/HA RDQM。

DR/HA RDQM 結合高可用性 RDQM 的特性 (請參閱 第 448 頁的『RDQM 高可用性』) 和災難回復 RDQM (請參閱 第 470 頁的『RDQM 災難回復』)。

下圖顯示範例 DR/HA RDQM。



主要站台上的 DR/HA RDQM 與災難回復站台之間的抄寫一律為非同步。使用非同步抄寫，IBM MQ PUT 或 GET 之類的作業會完成並回到應用程式，然後再將事件抄寫至次要佇列管理程式。

您可以有兩個作用中站台，而不是「主要」及「回復」站台(必要的話)，因此在正常作業期間，部分 DR/HA RDQM 會在一個站台上執行，而部分則在另一個站台上執行。如果發生災難，且一個站台變成無法使用，則所有 DR/HA RDQM 都會在相同站台的相同 HA 群組上執行。

每一個 HA 群組的配置方式與一般 HA 群組相同。您可以在每一個 HA 群組中定義 DR/HA RDQM 的浮動 IP 位址。每個 HA 群組的浮動 IP 位址可以相同或不同。

您無法將現有的 RDQM 升級成 DR/HA RDQM，您必須建立 DR/HA RDQM。(必要的話，您可以備份現有 RDQM 的資料，刪除它，將它重建為 DR/HA RDQM，然後還原資料，請參閱 [第 525 頁的『備份及還原 IBM MQ 佇列管理程式資料』](#)。)

若要配置 DR/HA RDQM，您必須完成下列主要步驟：

1. 在 'main' 網站上配置 HA 群組。
2. 在「回復」網站上配置高可用性群組。
3. 在 'main' 站台中 HA 群組的一個節點上建立主要/主要 DR/HA RDQM。
4. 在「主要」站台中其他兩個節點上建立主要/次要 DR/HA RDQM。
5. 定義應用程式的浮動 IP 位址，以在 'main' 站台上 HA 群組的任何節點上執行時存取 DR/HA RDQM。
6. 在「回復」站台上 HA 群組的一個節點上建立次要/主要 DR/HA RDQM。
7. 在「回復」站台中其他兩個節點上建立次要/次要 DR/HA RDQM。
8. 當應用程式在「回復」站台上 HA 群組的任何節點上執行時，定義應用程式的浮動 IP 位址以存取 DR/HA RDQM。

下列主題提供每一個步驟的詳細資料。

Linux V 9.1.5 DR/HA RDQM 解決方案的需求

DR/HA RDQM 解決方案的需求與 HA RDQM 解決方案和 DR RDQM 解決方案的需求相同。

如需配置的 HA 部分需求的詳細資料，請參閱 [第 450 頁的『RDQM HA 解決方案的需求』](#)。

如需配置中 DR 組件的詳細資料，請參閱 [第 472 頁的『RDQM DR 解決方案的需求』](#)。

Linux V 9.1.5 配置 DR/HA RDQM 的 HA 群組

您必須在主要及回復站台上建立 HA 群組。如果任一站台上現有的 HA 群組，您可以在該 HA 群組中建立 DR/HA RDQM。(現有的 RDQ 機制將繼續如常運作。)

此程序與 RDQM 高可用性的說明相同，請參閱 [第 453 頁的『定義 Pacemaker 叢集 \(HA 群組\)』](#)。

定義高可用性群組時，您可以在 `rdqm.ini` 檔案中指定每個節點用於監視及抄寫的 IP 位址。建立 HA 群組以支援 DR/HA RDM 時，您也可以指定您所定義的 HA 群組用於 DR 抄寫的 IP 位址，以及 DR 配對的其他 HA 群組中節點用於 DR 抄寫的 IP 位址。(如果您未在 `rdqm.ini` 檔中指定 DR 抄寫 IP 位址，則可以在建立 DR/HA RDQM 時在指令行中指定它們。)

如果您要配置現有的 HA 群組，則可以將 DR 抄寫 IP 位址新增至現有的 `rdqm.ini` 檔案。您不需要在更新 `rdqm.ini` 之後重新執行 `rdqmadm`，但必須先更新 `rdqm.ini`，然後再建立任何 DR/HA RDQM。

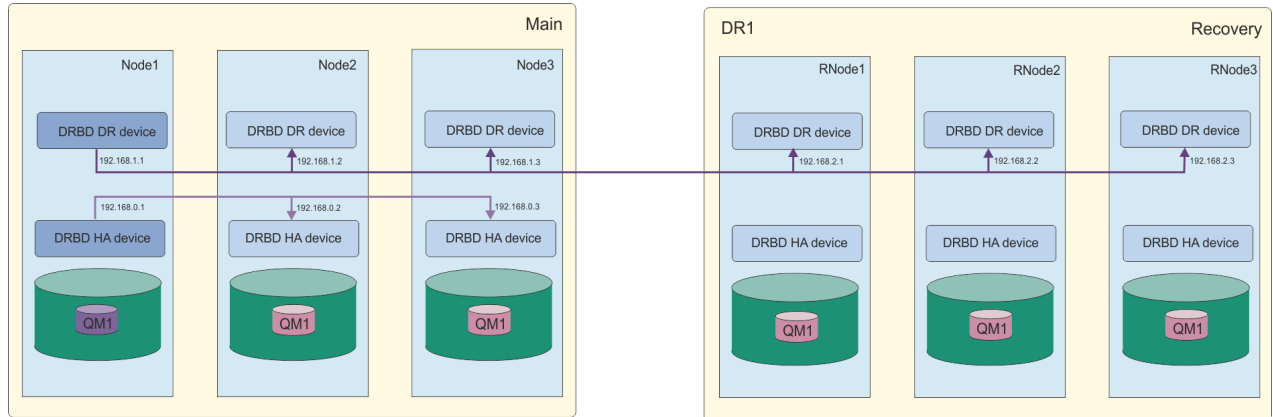
使用 Node 段落中的 `DR_Replication` 屬性，在您定義的 HA 群組上指定 DR 抄寫介面，例如：

```
Node:
  Name=Node1
  HA_Replication=192.168.0.1
  DR_Replication=192.168.1.1
Node:
  Name=Node2
  HA_Replication=192.168.0.2
  DR_Replication=192.168.1.2
Node:
  Name=Node3
  HA_Replication=192.168.0.3
  DR_Replication=192.168.1.3
```


使用 DRGroup 段落來指定遠端 HA 群組的 DR 抄寫位址，例如：

```
DRGroup:  
Name=DR1  
DR_Replication=192.168.2.1  
DR_Replication=192.168.2.2  
DR_Replication=192.168.2.3
```

下圖說明此配置：



如果您在建立 DR/HA RDQM 時，未在 `rdqm.ini` 檔案中或在指令行上為本端 HA 群組中的節點指定 DR 抄寫 IP 位址，則針對每一個節點定義的 `HA_Replication` 介面會用於 DR 抄寫。您必須在 `rdqm.ini` 檔案中或在 `crtmqm` 指令行上指定遠端 HA 群組 DR 抄寫位址。

Linux V 9.1.5 建立 DR/HA RDQM

您可以使用 `crtmqm` 指令，在 DR/HA 配置中建立抄寫的資料佇列管理程式 (RDQM)。

關於這項作業

如果使用者可以使用 `sudo`，您可以在 `mqm` 群組中以使用者身分建立 DR/HA RDQM。否則，您必須以 `root` 身分建立 RDQM。

您必須建立一些 DR/HA RDQM：

- 在「主要」網站上的 HA 群組上：
 - 在您要佇列管理程式在正常條件下執行的節點上，建立主要/主要 DR/HA RDQM。
 - 在 HA 群組中其他兩個節點的每一個上，建立主要/次要 DR/HA RDQM。
- 在「回復」網站上的 HA 群組上：
 - 在佇列管理程式將執行的節點上，如果它失效接手至回復站台，請建立次要/主要 DR/HA RDQM。當您在「主要」網站上建立主要/主要佇列管理程式時，可以使用指令輸出。
 - 在 HA 群組中其他兩個節點的每一個上，建立次要/次要 DR/HA RDQM。

所有佇列管理程式實例都必須具有相同的名稱，且必須配置相同的儲存體數量。

程序

- 如果要建立主要/主要 DR/HA RDQM，請執行下列動作：

a) 執行下列指令：

```
crtmqm -sx -rr p  
[-rl DRLocalIP1,DRLocalIP2,DRLocalIP3]  
(-ri DRRemoteIP1,DRRemoteIP2,DRRemoteIP3 | -in GroupName)  
-ip DRPort  
[-z] [-q] [-c Text] [-d DefXmitQ] [-h MaxHandles]  
[-g ApplicationGroup] [-oa user|group]  
[-t TrigInt] [-u DeadQ] [-x MaxUMsgs]  
[-lp LogPri] [-ls LogSec]
```

```
[-lc | -ll | -lla | -lln] [-lf LogFileSize]
[-p Port] [-fs FilesystemSize] QMgrName
```

其中：

-sx

指出起始 HA 角色是主要角色。

-rr p

指出起始 DR 角色是主要。

-rl DRLocalIP1、DRLocalIP2、DRLocalIP3

選擇性地指定本端站台 (即「主要」站台) 上三個節點上 DR 介面的 IP 位址。如果未指定，則會使用 `rdqm.ini` 檔案中指定的 IP 位址。

-ri DRRemoteIP1、DRRemoteIP2、DRRemoteIP3

指定遠端站台 (即「回復」站台) 上三個節點上 DR 介面的 IP 位址。您必須指定此參數或 `-rn` 參數。

-rn GroupName

指定 `rdqm.ini` 檔案中指定的遠端 HA 群組名稱。您必須指定 `-ri` 或 `-rn`。

-rp 埠

指定用於 DR 抄寫的埠。

other_crtmqm_options

您可以選擇性地指定下列一或多個一般 `crtmqm` 選項：

- z
- q
- c 文字
- d *DefaultTransmission* 佇列
- h *MaxHandles*
- g *ApplicationGroup*
- oa user | group
- t *TrigInt*
- u *DeadQ*
- x *MaxUMsgs*
- lp *LogPri*
- ls *LogSec*
- lc | -l
- lla | -lln
- lf *LogFile* 大小
- p 埠

-fs size

選擇性地指定要為佇列管理程式建立的檔案系統大小，即 `drbdpool` 磁區群組中建立的邏輯磁區大小。也會建立該大小的另一個邏輯磁區，以支援回復至 Snapshot 作業，因此 DR RDQM 的儲存體總計僅為這裡指定的兩倍。

QMNAME

指定抄寫資料佇列管理程式的名稱。此名稱區分大小寫。

指令完成之後，它會輸出您可以在回復站台上輸入的指令，以建立佇列管理程式的次要/主要實例。

- 如果要在 HA 群組中的其他兩個節點上建立主要/次要 DR/HA RDQM，請執行下列動作：

a) 在每一個節點上輸入下列指令：

```
crtmqm -sxs -rr p
      [-rl DRLocalIP1,DRLocalIP2,DRLocalIP3]
      (-ri DRRemoteIP1,DRRemoteIP2,DRRemoteIP3 | -rn GroupName)
```

```
-rp DRPort  
[-fs FilesystemSize] QMgrName
```

其中：

-sxs

指出起始 HA 角色是次要角色。

-rr p

指出起始 DR 角色是主要。

-rl DRLocalIP1、DRLocalIP2、DRLocalIP3

選擇性地指定本端站台 (即「主要」站台) 上三個節點上 DR 介面的 IP 位址。如果未指定，則會使用 `rdqm.ini` 檔案中指定的 IP 位址。

-ri DRRemoteIP1、DRRemoteIP2、DRRemoteIP3

指定遠端站台 (即「回復」站台) 上三個節點上 DR 介面的 IP 位址。您必須指定此參數或 `-rn` 參數。

-rn GroupName

指定 `rdqm.ini` 檔案中指定的遠端 HA 群組名稱。您必須指定 `-ri` 或 `-rn`。

-rp 埠

指定用於 DR 抄寫的埠。

-fs size

指定要為佇列管理程式建立的檔案系統大小，即在 `drbdpool` 磁區群組中建立的邏輯磁區大小。如果您在建立主要/主要 RDQM 時已指定非預設大小，則必須在這裡指定相同的值。

QMNAME

指定主要/次要 RDQM 的名稱。這必須與您為 RDQM 的主要/主要實例指定的名稱相同。請注意，名稱區分大小寫。

- 如果要在佇列管理程式失效接手至回復網站時將執行的節點上建立次要/主要 DR/HA RDQM，請執行下列動作：

a) 當您在主要站台上建立主要/主要 DR/HA 時，請使用指令輸出，或輸入下列指令：

```
critmqm -sx -rr s  
          [-rl DRLocalIP1,DRLocalIP2,DRLocalIP3]  
          (-ri DRRemoteIP1,DRRemoteIP2,DRRemoteIP3 | -rn GroupName)  
          -rp DRPort  
          [-fs FilesystemSize] QMgrName
```

-sx

指出起始 HA 角色是主要角色。

-rr s

指出起始 DR 角色是次要的。

-rl DRLocalIP1、DRLocalIP2、DRLocalIP3

選擇性地指定本端站台 (即「回復」站台) 上三個節點上 DR 介面的 IP 位址。如果未指定，則會使用 `rdqm.ini` 檔案中指定的 IP 位址。

-ri DRRemoteIP1、DRRemoteIP2、DRRemoteIP3

指定遠端站台 (即「主要」站台) 上三個節點上 DR 介面的 IP 位址。您必須指定此參數或 `-rn` 參數。

-rn GroupName

指定 `rdqm.ini` 檔案中指定的遠端 HA 群組名稱。您必須指定 `-ri` 或 `-rn`。

-rp 埠

指定用於 DR 抄寫的埠。

-fs size

選擇性地指定要為佇列管理程式建立的檔案系統大小，即 `drbdpool` 磁區群組中建立的邏輯磁區大小。也會建立該大小的另一個邏輯磁區，以支援回復至 Snapshot 作業，因此 DR RDQM 的儲存體總計僅為這裡指定的兩倍。

QMNAME

指定抄寫資料佇列管理程式的名稱。此名稱區分大小寫。

- 如果要在回復站台的其他兩個節點上建立次要/次要 HA/DR RDQM，請執行下列動作：

a) 在每一個節點上輸入下列指令：

```
crtmqm -sxs -rr s
      [-rl DRLocalIP1,DRLocalIP2,DRLocalIP3]
      (-ri DRRemoteIP1,DRRemoteIP2,DRRemoteIP3 | -rn GroupName)
      -rp DRPort
      [-fs FilesystemSize] QMgrName
```

-sxs

指出起始 HA 角色是主要角色。

-rr s

指出起始 DR 角色是次要的。

-rl DRLocalIP1、DRLocalIP2、DRLocalIP3

選擇性地指定本端站台上三個節點上 DR 介面的 IP 位址。如果未指定，則會使用 rdqm.ini 檔案中指定的 IP 位址。

-ri DRRemoteIP1、DRRemoteIP2、DRRemoteIP3

指定遠端站台上三個節點上 DR 介面的 IP 位址。您必須指定此參數或 -rn 參數。

-rn GroupName

指定 rdqm.ini 檔案中指定的遠端 HA 群組名稱。您必須指定 -ri 或 -rn。

-rp 埠

指定用於 DR 抄寫的埠。

-fs size

選擇性地指定要為佇列管理程式建立的檔案系統大小，即 drbdpool 磁區群組中建立的邏輯磁區大小。也會建立該大小的另一個邏輯磁區，以支援回復至 Snapshot 作業，因此 DR RDQM 的儲存體總計僅為這裡指定的兩倍。

QMNAME

指定抄寫資料佇列管理程式的名稱。此名稱區分大小寫。

下一步

建立所有 DR/HA RDQM 之後，您必須檢查主要/主要及次要/主要實例上的狀態，才能檢查所有實例是否正確。在節點上使用 **rdqmstatus** 指令。節點應該會顯示正常狀態，如第 494 頁的『檢視 DR/HA RDQM 和 HA 群組狀態』中所述。如果他們未顯示此狀態，請刪除次要/主要實例並重建它，小心使用正確的引數。

相關工作

第 489 頁的『建立 DR/HA RDQM』

您可以使用 **crtmqm** 指令，在 DR/HA 配置中建立抄寫的資料佇列管理程式 (RDQM)。

相關參考

[crtmqm](#)

Linux V 9.1.5 刪除 DR/HA RDQM

您可以使用 **dltmqm** 指令來刪除 DR/HA 抄寫的資料佇列管理程式 (RDQM)。

關於這項作業

您必須執行指令來刪除主要/主要節點和次要/主要節點上的 RDQM。RDQM 必須先結束。如果 mqm 使用者具有必要的 sudo 專用權，則可以該使用者身分執行指令。否則，您必須以 root 身分執行指令。

程序

- 若要刪除 DR/HA RDQM，請輸入下列指令：

```
dltmqm RDQM_name
```

相關參考

[dltmqm](#)

Linux V 9.1.5 建立浮動 IP 位址

您可以在 DR/HA RDQM 配置中為每一個 HA 群組建立浮動 IP 位址。

浮動 IP 位址可讓用戶端使用 DR/HA RDQM 的相同 IP 位址，而不論它在 HA 群組中的哪個節點上執行。如果您的兩個 HA 群組具有應用程式連線功能的專用/隔離網路，則可以為這兩個群組定義相同的浮動 IP 位址。您仍必須在每一個 HA 群組上定義該浮動 IP 位址兩次。

您可以使用與 HA RDQM 相同的方法來建立及刪除浮動 IP 位址。請參閱 [第 463 頁的『建立及刪除浮動 IP 位址』](#)。

Linux V 9.1.5 啟動、停止及顯示 DR/HA RDQM 的狀態

您可以使用標準 IBM MQ 控制指令的變式來啟動、停止及檢視 DR/HA RDQM 的現行狀態。

關於這項作業

您必須以屬於 mqm 及 haclient 群組的使用者身分，執行啟動、停止及檢視 DR/HA RDQM 現行狀態的指令。

您必須執行指令，以在該佇列管理程式的主要節點上啟動及停止佇列管理程式。

程序

- 如果要啟動 RDQM，請在 RDQM 的主要節點上輸入下列指令：

```
strmqm qmname
```

其中 *qmname* 是您要啟動之 DR/HA RDQM 的名稱。

RDQM 已啟動，且 Pacemaker 開始管理 RDQM。如果您想要指定任何其他 `strmqm` 選項，則必須將 `-ns` 選項與 `strmqm` 一起指定。

- 若要停止 RDQM，請在 DR/HA RDQM 的主要節點上輸入下列指令：

```
endmqm qmname
```

其中 *qmname* 是您要停止之 RDQM 的名稱。

Pacemaker 會停止管理 RDQM，然後結束 RDQM。停止 RDQM 時，可以使用所有其他 `endmqm` 參數。

- 如果要檢視 RDQM 的狀態，請輸入下列指令：

```
dspmq -m QMname
```

輸出的狀態資訊取決於您是在 RDQM 的主要或次要節點上執行指令。如果在主要節點上執行，則會顯示 `dspmq` 所傳回的其中一個正常狀態訊息。如果您在次要節點上執行指令，則會顯示狀態 `Ended immediately`。例如，如果在節點 RDQM7 上執行 `dspmq`，則可能會傳回下列資訊：

```
QMNAME(DRQM8)          STATUS(Ended immediately)
QMNAME(DRQM7)          STATUS(Running)
```

您可以搭配使用引數與 `dspmq`，以建立 RDQM 是否配置用於災難回復，以及它目前是主要或次要實例：

```
dspmq -m QMname -o (dr | DR)
```

會顯示下列其中一個回應：

DRROLE()

指出佇列管理程式未配置災難回復。

DRROLE(Primary)

指出佇列管理程式已配置為 DR 主要。

DRROLE(Secondary)

指出佇列管理程式已配置為 DR 次要。

使用 **dspmq -o all** 指令來檢視 DR/HA RDQM 的災難回復及高可用性資訊。比方說，如果您在 DR/HA RDQM 執行所在的節點上執行 **dspmq -o all**，則會看到下列狀態資訊：

```
QMNAME(TESTQM1)                                STATUS(Running) HA(Replicated)
DRROLE(Primary)
```

相關參考

- [dspmq \(顯示佇列管理程式\)](#)
- [endmqm \(結束佇列管理程式\)](#)
- [strmqm \(啟動佇列管理程式\)](#)

Linux V 9.1.5 檢視 DR/HA RDQM 和 HA 群組狀態

您可以檢視 DR/HA 抄寫資料佇列管理程式 (RDQM) 的 HA 狀態及 DR 角色。

關於這項作業

您可以使用 **rdqmstatus** 指令來檢視個別 RDQM 的狀態，或取得 HA 群組已知所有 RDQM 的狀態概觀。

您必須是 mqm 和 haclient 群組中的使用者，才能執行 **rdqmstatus** 指令。您可以在任一 HA 群組中的任何節點上執行指令。

程序

- 如果要檢視屬於 HA 配置之節點和 RDQ 的狀態，請執行下列動作：

```
rdqmstatus
```

即會顯示您執行指令所在節點的身分，以及 HA 配置中 RDQ 的狀態，以及其現行 DR 角色，例如：

```
Node: main-alice
Queue manager name: RDQM1
Queue manager status: Running elsewhere
HA current location: main-charlie

Queue manager name: RDQM9
Queue manager status: Running elsewhere
HA current location: main-bob
DR role: Primary

Queue manager name: RDQM7
Queue manager status: Running
HA current location: This node
DR role: Primary
```

在此範例中，RDQM7 及 RDQM8 都是 DR/HA RDQM，而 RDQM1 是 HA RDQM，它未配置成能夠切換至災難回復站台。

- 若要檢視 HA 群組中所有節點上特定佇列管理程式的狀態，請輸入下列指令：

```
rdqmstatus -m qmname
```

其中 *qmname* 是您要檢視其狀態的 RDQM 名稱。會顯示現行節點上 RDQM 的狀態，後面接著現行節點視景中其他兩個節點的狀態摘要。

下表彙總 **rdqmstatus** 指令可針對 RDQM 傳回之現行節點的相關資訊。

「狀態」屬性	可能值	顯示時
節點名稱	<i>NodeName</i>	一律顯示
佇列管理程式狀態	佇列管理程式的狀態 (其中一個適用於 dspmq 指令的狀態)	一律顯示

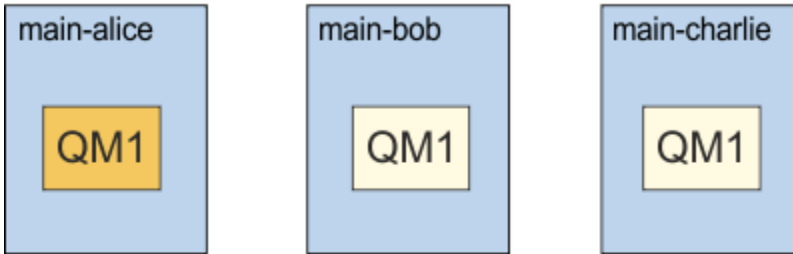
表 36: 現行節點狀態 (繼續)		
「狀態」屬性	可能值	顯示時
CPU	<i>n.nn%</i>	只有在此節點上執行 RDQM 時才會顯示
記憶體	已使用 <i>nnn</i> MB	只有在此節點上執行 RDQM 時才會顯示
佇列管理程式檔案系統	<i>nnn</i> 已使用 MB , <i>y</i> 。 <i>y</i> 已配置 GB [z%]	只有在此節點上執行 RDQM 時才會顯示
HA 角色	主要 次要 V9.1.5 次要擱置 不明	一律顯示
HA 狀態	所有節點處於待命狀態 此節點處於待命狀態 遠端節點待命中 混合	所有節點處於待命狀態 現行節點處於待命狀態 兩個遠端節點都處於待命狀態 每一個遠端節點的不同狀態
HA 控制	已啟用 已停用 不明	一律顯示。顯示 RDQM 是否在 Pacemaker 控制下
HA 偏好位置	無 此節點 不明 <i>nodeName</i>	一律顯示
HA 浮動 IP 介面	<i>interface_name</i>	一律顯示
HA 浮動 IP 位址	<i>IPV4_address</i>	一律顯示
DR 角色	主要 次要 不明	一律顯示

表 36: 現行節點狀態 (繼續)		
「狀態」屬性	可能值	顯示時
DR 狀態	正常 同步化進行中 已分割 遠端系統無法使用 不一致 正在回復至 Snapshot 未配置遠端系統 協議失敗	一切都好 同步化進行中。 使用者已啟動佇列 manager on each node while the DR 抄寫網路已 unavailable. 與另一個節點的連線已遺失。 同步化正在進行中，但已岔斷。 使用者已選擇回復至擷取的 Snapshot 佇列管理程式所輸入的不一致狀態。 已配置主要伺服器但次生卻沒有 主要節點與次要節點之間的起始協議失敗。原因可能是抄寫類型不相容，或次要節點配置了較小的檔案系統大小。
DR 狀態 (在 HA 次要節點上)	請參閱 <i>HA_Primary_Node</i>	在 HA 次要節點上顯示，因為 DR 狀態僅在 HA 主要節點上已知。
DR 埠	用來抄寫此佇列管理程式資料的 TCP/IP 埠。	一律顯示。
DR 本端 IP 位址	此佇列管理程式將用於 DR 抄寫的本端 IP 位址	一律顯示。
DR 遠端 IP 位址清單	此佇列管理程式將用於 DR 抄寫的遠端 IP 位址。以逗點區隔的三個 IP 位址清單。	一律顯示。
DR 現行遠端 IP 位址	此佇列管理程式連接至以進行 DR 抄寫的現行遠端 IP。	適用於具有作用中 DR 連線的 HA 主要。
DR 現行遠端 IP 位址 (在 HA 次要節點上)	請參閱 <i>HA_Primary_Node</i>	在 HA 次要節點上顯示，因為 DR 連線僅在 HA 主要節點上
DR 不同步資料	xKB	當遠端節點無法使用或不一致時顯示。
DR 同步化進行中	y%	同步化進行中時顯示。
DR 預估完成時間	yyyy-MM-dd HH:mm:ss	同步化進行中時顯示。
Snapshot 回復進度	y%	當 DR 狀態為「回復至 Snapshot」時顯示

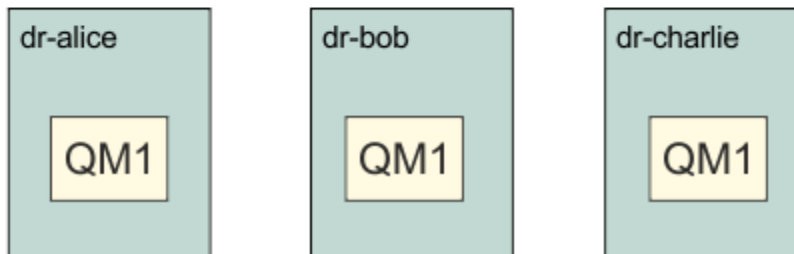
範例

這些範例說明 **rdqmstatus -m qm1** 指令在下列 DR/HA 配置的各種節點上執行:

main site



dr site



在作為 DR 主要及 HA 主要節點的節點上正常狀態的範例:

```
Node: main-alice
Queue manager status: Running
CPU: 0.00%
Memory: 123MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
HA role: Primary
HA status: Normal
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA floating IP interface: None
HA floating IP address: None
DR role: Primary
DR status: Normal
DR port: 3000
DR local IP address: 192.168.1.1
DR remote IP address list: 192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address: 192.168.2.1

Node: main-bob
HA status: Normal

Node: main-charlie
HA status: Normal
```

DR 主要及 HA 次要節點上正常狀態的範例:

```
Node: main-bob
Queue manager status: Running elsewhere
HA role: Secondary
HA status: Normal
HA control: Enabled
HA current location: main-alice
HA preferred location: main-alice
HA floating IP interface: None
HA floating IP address: None
DR role: Primary
DR status: See main-alice
DR port: 3000
DR local IP address: 192.168.1.2
DR remote IP address list: 192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address: See main-alice

Node: main-alice
```

```
HA status: Normal
Node: main-charlie
HA status: Normal
```

在 DR 次要及 HA 主要節點上的正常狀態範例:

```
Node: dr-alice
Queue manager status: Ended immediately
HA role: Primary
HA status: Normal
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA floating IP interface: None
HA floating IP address: None
DR role: Secondary
DR status: Normal
DR port: 3000
DR local IP address: 192.168.2.1
DR remote IP address list: 192.168.1.1,192.168.1.2,192.168.1.3
DR current remote IP address: 192.168.1.1

Node: dr-bob
HA status: Normal

Node: dr-charlie
HA status: Normal
```

作為 DR 次要及 HA 次要的節點上正常狀態的範例:

```
Node: dr-bob
Queue manager status: Ended immediately
HA role: Secondary
HA status: Normal
HA control: Enabled
HA current location: dr-alice
HA preferred location: dr-alice
HA floating IP interface: None
HA floating IP address: None
DR role: Secondary
DR status: See dr-alice
DR port: 3000
DR local IP address: 192.168.2.2
DR remote IP address list: 192.168.1.1,192.168.1.2,192.168.1.3
DR current remote IP address: See dr-alice

Node: dr-alice
HA status: Normal

Node: dr-charlie
HA status: Normal
```

在作為 DR 主要及 HA 主要節點的節點上進行 DR 同步化的範例:

```
Node: main-alice
Queue manager status: Running
CPU: 0.00%
Memory: 123MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
HA role: Primary
HA status: Normal
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA floating IP interface: None
HA floating IP address: None
DR role: Primary
DR status: Normal
DR port: 3000
DR local IP address: 192.168.1.1
DR remote IP address list: 192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address: 192.168.2.1
DR synchronization progress: 11.0%
DR estimated time to completion: 2018-09-06 14:55:05

Node: main-bob
HA status: Normal
```

```
Node: main-charlie
HA status: Normal
```

在 DR 主要及 HA 主要節點上分割的 DR 範例:

```
Node: main-alice
Queue manager status: Running
CPU: 0.00%
Memory: 123MB
Queue manager file system: 51MB used, 1.0GB allocated [5%]
HA role: Primary
HA status: Normal
HA control: Enabled
HA current location: This node
HA preferred location: This node
HA floating IP interface: None
HA floating IP address: None
DR role: Primary
DR status: Partitioned
DR port: 3000
DR local IP address: 192.168.1.1
DR remote IP address list: 192.168.2.1,192.168.2.2,192.168.2.3
DR current remote IP address: 192.168.2.1
DR out of sync data: 372KB

Node: main-bob
HA status: Normal

Node: main-charlie
HA status: Normal
```

相關參考

 [rdqmstatus](#)

  [在 DR/HA 環境中運作](#)

在 DR/HA 環境中運作時，對於高可用性及災難回復有個別考量。

如果執行 DR/HA RDQM 的節點失敗，RDQM 會自動失效接手至該 HA 群組中的另一個節點。如果整個站台失敗，則您必須在回復站台上 HA 群組中的偏好節點上手動啟動 RDQM。這裡的考量與一般 DR RDQM 的考量相同，如需相關資訊，請參閱第 484 頁的『在災難回復環境中運作』。

如果其中一個節點完全故障，且需要更換，請參閱第 485 頁的『更換災難回復配置中的故障節點』和 第 469 頁的『在高可用性配置中更換故障節點』以取得指引。

  [更換 DR/HA 配置中的故障節點](#)

如果任一 HA 群組中的其中一個節點失敗，您可以取代它。

關於這項作業

程序會根據您要更換的節點是 DR 配置中的主要或次要節點而有所不同。在任一情況下，新節點必須與您要取代的節點具有相同的配置，亦即，它必須具有相同的主機名稱、相同的 IP 位址等。

您也可能會遇到下列狀況：您在主要或回復網站上完全失去 HA 群組，且必須取代整個 HA 群組。

程序

- 對於 DR 配置中作為主要節點的替換用節點，請在新節點上完成下列步驟：
 - 建立與其他節點上的檔案相符的 `rdqm.ini` 檔案，然後執行 `rdqmadm -c` 指令 (請參閱第 453 頁的『定義 Pacemaker 叢集 (HA 群組)』)。
 - 執行 `crtmqm -sxs -rr p qmanager` 指令，以重建每一個 DR/HA RDQM (請參閱第 489 頁的『建立 DR/HA RDQM』)。
- 對於 DR 配置中為次要的替換用節點，請在新節點上完成下列步驟：
 - 建立與其他節點上的檔案相符的 `rdqm.ini` 檔案，然後執行 `rdqmadm -c` 指令 (請參閱第 453 頁的『定義 Pacemaker 叢集 (HA 群組)』)。

- b) 執行 `crtmqm -sx -rr sqmanager` 指令，以重建每一個 DR/HA RDQM (請參閱 [第 489 頁的『建立 DR/HA RDQM』](#))。
- 若要取代整個 HA 群組，請完成下列步驟：
 - a) 如果您在 DR 主要網站 (即主要網站) 失去整個 HA 群組，則必須遵循步驟來執行 DR 次要網站的受管理失效接手，以繼續執行 DR/HA RDQM (請參閱 [第 484 頁的『在災難回復環境中運作』](#))。(如果您在回復站台失去整個 HA 群組，則 DR/HA RDQM 會繼續在主要站台上執行，而無需人為介入。)
 - b) 在三個替換用節點上重建 HA 群組，如 [第 488 頁的『配置 DR/HA RDQM 的 HA 群組』](#) 中所述。
 - c) 在新的 HA 群組上重建 DR/HA RDQM，如 [第 489 頁的『建立 DR/HA RDQM』](#) 中所述。
 - d) 必要的話，請執行從回復網站回到主要網站的受管理失效接手。

Linux V 9.1.5 DR/HA RDQM 工作範例

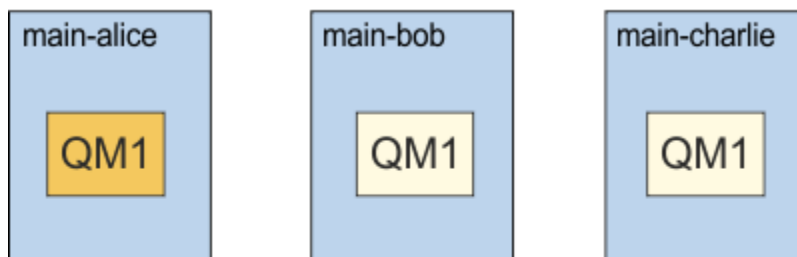
此範例顯示如何建立及刪除 DR/HA RDQM。

建立 DR/HA RDQM

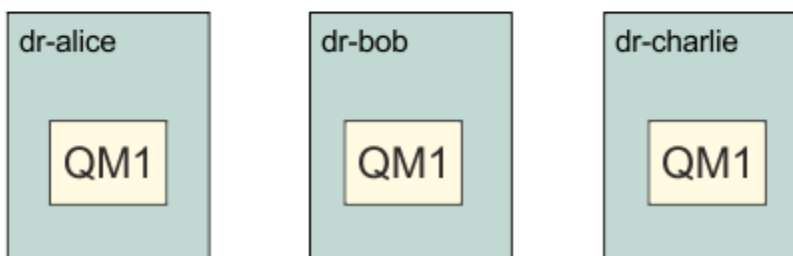
範例配置有兩個站台，名稱為 'main' 和 'dr'。每一個站台都有三個節點，名為 'alice'、'bob' 及 'charlie'。節點具有由其站台名稱及名稱組成的完整名稱，因此 'main-alice'、'dr-alice' 等。

下列步驟會建立在 main-alice 上執行且名為 QM1 的 DR/HA RDQM。主要 alice 節點是 HA 及 DR 主要節點。

main site



dr site



如果在 `rdqm.ini` 檔案中指定本端及遠端 DR IP 位址，則不需要在指令行上指定任何 IP 位址，而且可以透過在 main-alice 上執行下列指令來建立名為 QM1 的 DR/HA RDQM：

```
crtmqm -sx -rr p -rn DR1 -rp 7001 QM1
```

如果在 `rdqm.ini` 檔案中指定本端 DR IP 位址，則可以在指令行上指定遠端 DR IP 位址：

```
crtmqm -sx -rr p -ri 192.168.2.1,192.168.2.2,192.168.2.3 -rp 7001 QM1
```

如果在 `rdqm.ini` 檔案中未指定任何 DR IP 位址，則可以在指令行上同時指定遠端及本端 DR IP 位址：

```
crtmqm -sx -rr p -rl 192.168.1.1,192.168.1.2,192.168.1.3 -ri  
192.168.2.1,192.168.2.2,192.168.2.3 -rp 7001 QM1
```

回應建立 QM1 的輸出如下列範例所示:

```
Creating replicated data queue manager configuration.
Secondary queue manager created on 'main-bob'.
Secondary queue manager created on 'main-charlie'.
IBM MQ queue manager created.
Directory '/var/mqm/vols/qm1/qmgr/qm1' created.
The queue manager is associated with installation 'Installation1'.
Creating or replacing default objects for queue manager 'QM1'.
Default objects statistics : 83 created. 0 replaced. 0 failed.
Completing setup.
Setup completed.
Enabling replicated data queue manager.
Replicated data queue manager enabled.
Issue the following command on the remote HA group to create the DR/HA secondary queue manager:
crtmqm -sx -rr s -rl 192.168.2.1,192.168.2.2,192.168.2.3 -ri
192.168.1.1,192.168.1.2,192.168.1.3 -rp 7001 -fs 3072M QM1
```

從訊息複製指令，以在 dr-alice 上建立 QM1 的 DR 次要實例:

```
crtmqm -sx -rr s -rl 192.168.2.1,192.168.2.2,192.168.2.3 -ri
192.168.1.1,192.168.1.2,192.168.1.3 -rp 7001 -fs 3072M QM1
```

下列訊息是 dr-alice 上的輸出:

```
Creating replicated data queue manager configuration.
Secondary queue manager created on 'dr-bob'.
Secondary queue manager created on 'dr-charlie'.
IBM MQ secondary queue manager created.
Enabling replicated data queue manager.
```

測試 DR 次要

若要測試 QM1 的災難回復特性，請在 main-alice 上執行下列指令，以讓 QM1 成為 DR 次要實例:

```
rdqmdr -m QM1 -s
Queue manager 'QM1' has been made the DR secondary on this node.
```

在 dr-alice 上執行下列指令，以讓 QM1 成為該節點上的 DR 主要實例:

```
rdqmdr -m QM1 -p
Queue manager 'QM1' has been made the DR primary on this node.
```

刪除 DR/HA RDQM

若要刪除名為 QM1 的 DR/HA RDQM，請先結束 main-alice 上的佇列管理程式:

```
endmqm -w QM1
Replicated data queue manager disabled.
Waiting for queue manager 'QM1' to end.
IBM MQ queue manager 'QM1' ended.
```

然後在 main-alice 上執行下列指令，以刪除 QM1:

```
dltmqm QM1
Removing replicated data queue manager configuration.
Secondary queue manager deleted on 'main-bob'.
Secondary queue manager deleted on 'main-charlie'.
IBM MQ queue manager 'QM1' deleted.
```

最後，您必須在 dr-alice 上刪除 QM1:

```
dltmqm QM1
Removing replicated data queue manager configuration.
Secondary queue manager deleted on 'dr-bob'.
Secondary queue manager deleted on 'dr-charlie'.
IBM MQ queue manager 'QM1' deleted.
```

相關概念

第 499 頁的『在 DR/HA 環境中運作』

在 DR/HA 環境中運作時，對於高可用性與災難回復有個別考量。

相關工作

第 489 頁的『建立 DR/HA RDQM』

您可以使用 `crtmqm` 指令，在 DR/HA 配置中建立抄寫的資料佇列管理程式 (RDQM)。

第 492 頁的『刪除 DR/HA RDQM』

您可以使用 `dltmqm` 指令來刪除 DR/HA 抄寫的資料佇列管理程式 (RDQM)。

記載: 確定訊息不會遺失

IBM MQ 會將佇列管理程式所控制之持續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，IBM MQ 也依賴於管理其檔案 (包括日誌檔) 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。



注意: 您無法將回復日誌移至不同的作業系統。

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。

IBM 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。

有許多日誌範圍包含要記錄的資料。您可以定義數目及大小 (如第 81 頁的『[mqs.ini 檔案的 LogDefaults 段落](#)』中所說明)，或採用三個主要及兩個次要延伸範圍的系統預設值。

三個主要及兩個次要延伸範圍各預設為 16 MB。

當您建立佇列管理程式時，預先配置の日誌範圍數是已配置的主要日誌範圍數。如果您未指定數字，則會使用預設值。

IBM MQ 使用兩種類型的記載：

- 循環
- 線性

與線性記載搭配使用的日誌範圍數目可能非常大，視媒體映像檔記錄的頻率而定。

如需相關資訊，請參閱第 503 頁的『[記載類型](#)』。

Windows

在 IBM MQ for Windows 中，如果您尚未變更日誌路徑，則會在下列目錄下建立日誌範圍：

```
C:\ProgramData\IBM\MQ\log\QMgrName
```

ULW 在 IBM MQ for UNIX and Linux 系統中，如果您尚未變更日誌路徑，則會在下列目錄下建立日誌範圍：

```
/var/mqm/log/QMgrName
```

IBM MQ 從這些主要日誌範圍開始，但如果主要日誌空間不足，則會配置次要日誌範圍。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範圍。您可以變更此預設配置，如第 72 頁的『在 Multiplatforms 上的 .ini 檔中變更 IBM MQ 配置資訊』中所述。

日誌範圍會以字母 S 或字母 R 作為字首。作用中、非作用中及多餘延伸範圍的字首為 S，而重複使用延伸範圍的字首為 R。

備份或還原佇列管理程式時，請備份及還原所有作用中、非作用中及多餘的延伸範圍，以及日誌控制檔。

註：您不需要備份及還原重複使用延伸範圍。

日誌控制檔

日誌控制檔包含說明日誌範圍狀態所需的資訊，例如其大小及位置，以及下一個可用範圍的名稱。

重要：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。

日誌控制檔位於日誌路徑中，稱為 `amqhlctl.lfh`。備份或還原佇列管理程式時，請確定已備份及還原日誌控制檔，以及日誌範圍。

記載類型

在 IBM MQ 中，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。

IBM 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 504 頁的『使用檢查點檢查作業來確保完整回復』中說明了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。線性記載會將日誌資料保留在連續的日誌檔序列中。

日誌檔可以選擇性地為：

- 重複使用，但只有在重新啟動回復或媒體回復不再需要它們時才會重複使用。
- 手動保存以進行長期儲存及分析。

媒體影像的頻率會決定線性日誌檔可以重複使用的時間，並且是線性日誌檔必須可用的磁碟空間量的主要因素。

您可以根據時間或日誌使用情形，將佇列管理程式配置成自動取得定期媒體映像檔，也可以手動排定媒體映像檔。

您的管理者會決定要實作的原則，以及磁碟空間使用情形的含意。重新啟動回復所需的日誌檔必須一律可用，而只有媒體回復所需的日誌檔可以保存至長期儲存體 (例如磁帶)。

如果管理者啟用自動日誌管理及自動媒體映像檔，則線性記載的行為方式與非常大的循環日誌類似，但媒體回復已啟用針對媒體故障的改良備援。

V 9.1.0 從 IBM MQ 9.1.0，您可以使用 `migmqlog` 指令，將佇列管理程式的現有日誌類型從線性變更為循環，或從循環變更為線性。

非作用中的線性日誌範圍

V 9.1.0 **Multi**

從 IBM MQ 9.1.0 開始，如果您使用自動日誌管理 (包括保存)，日誌程式會追蹤非作用中的線性日誌範圍。



小心: 如果您是使用自動日誌管理而不保存，則此處理程序不支援使用備份佇列管理程式。

ULW

當回復不再需要日誌延伸範圍時，必要的話，日誌程式會在方便時刪除日誌延伸範圍或重複使用它。

重複使用的日誌範圍會重新命名為日誌序列中的下一個。會定期寫入訊息 AMQ7490，指出已建立、刪除或重複使用多少延伸範圍。

日誌程式會選擇要保留多少延伸範圍以供重複使用，以及何時刪除那些延伸範圍。

作用中日誌

在線性和循環式記載中，都有許多被認為作用中的檔案。作用中日誌是重新啟動回復可能參照的日誌空間量上限 (不論您是使用循環式或線性記載)。

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 507 頁的『計算日誌的大小』](#)。)

請注意，作用中日誌空間不包括媒體回復所需的空間，且與線性記載搭配使用的日誌檔數目可能非常大，視您的訊息流程及媒體映像檔的頻率而定。

非作用中日誌

當重新啟動回復不再需要日誌檔時，它會變成非作用中。重新啟動回復或媒體回復不需要的日誌檔可以視為多餘的日誌檔。

使用自動日誌管理時，佇列管理程式會控制這些多餘日誌檔的處理。如果您已選取手動日誌管理，則當您的作業不再感興趣時，管理者會負責管理 (例如，刪除及保存) 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 512 頁的『管理日誌』](#)。

次要日誌檔

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，則需要日誌活動之大部分進一步作業的要求將被拒絕，並傳回 MQRC_RESOURCE_PROBLEM 回覆碼給應用程式，且任何長時間執行的交易都將被視為非同步回復。



小心: 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

循環式記載和線性記載佇列管理程式都支援重新啟動回復。不論先前的佇列管理程式實例在重新啟動時如何突然終止 (例如停電)，佇列管理程式都會在終止時將其持續狀態還原為正確的交易式狀態。

重新啟動回復取決於所維護的磁碟完整性。同樣地，作業系統應該確保磁碟完整性，不論作業系統終止的突然程度如何。

在磁碟完整性未維護的高度異常事件中，線性記載 (及媒體回復) 會提供一些進一步的備援及可回復性選項。隨著越來越常見的技术 (例如 RAID)，磁碟完整性問題越來越罕見，許多企業會配置循環式記載並僅使用重新啟動回復。

IBM MQ 設計成典型的「預寫記載」資源管理程式。訊息佇列的持續更新分兩個階段進行：

1. 代表更新的日誌記錄會可靠地寫入回復日誌
2. 佇列檔或緩衝區的更新方式對您的系統而言最有效率，但不一定一致。

因此，日誌檔可能比基礎佇列緩衝區及檔案狀態更最新。

如果容許此狀況持續下去，則需要大量日誌重播，才能在損毀回復之後使佇列狀態一致。

IBM MQ 會使用 checkpoints，以限制在損毀回復之後所需的日誌重播量。控制日誌檔是否稱為作用中的關鍵事件是 checkpoint。

IBM MQ 檢查點是一個點：

- 回復日誌與物件檔之間的一致性。
- 識別日誌中的一個位置，可保證後續日誌記錄的轉遞重播在佇列管理程式結束時，會將佇列還原至正確的邏輯狀態。

在檢查點期間，IBM MQ 會視需要清除佇列檔的較舊更新項目，以限制需要重播的日誌記錄數量，讓佇列在損毀回復之後回到一致狀態。



最近的完整檢查點會標示日誌中的一個點，在損毀回復期間必須從中執行重播。因此，檢查點的頻率是在記錄檢查點的額外負荷與那些檢查點所隱含的潛在回復時間的改善之間的取捨。

在最近完成檢查點開始的日誌中，位置是判斷日誌檔是作用中還是非作用中的關鍵因素之一。另一個關鍵因素是與現行作用中交易所進行的第一個持續更新相關的第一個日誌記錄的日誌中的位置。

如果在第二個或更新版本的日誌檔中記錄新的檢查點，且沒有現行交易參照第一個日誌檔中的日誌記錄，則第一個日誌檔會變成非作用中。在循環式記載的情況下，第一個日誌檔現在已備妥可供重複使用。在線性記載的情況下，媒體回復通常仍需要第一個日誌檔。

如果您配置循環式記載或自動日誌管理，佇列管理程式會管理非作用中日誌檔。如果您使用手動日誌管理來配置線性記載，則它會變成管理作業，以根據作業需求來管理非作用中檔案。

IBM MQ 會自動產生檢查點。它們會在下列時間取得：


- 當佇列管理程式啟動時
- 關機時
- 當記載空間不足時
-  Multi 自從取得前一個檢查點之後，已記載 50,000 個作業之後
-  z/OS 自前次建立檢查點以來，已記載 *number_of_operations* 之後，其中 *number_of_operations* 是在 LOGLOAD 內容中設定的作業數。

當 IBM 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。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。


 Multi 發出 AMQ7229 及 AMQ7230 訊息，以指出重播階段的進度。

為了要知道要取消或確定哪些作業，IBM 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。

 Multi 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。

 Multi 發出 AMQ7233 訊息以指出解決階段的進度。

 z/OS 在 z/OS 上，重新啟動處理程序由各種階段組成。

1. 根據頁集所需的媒體回復，以及取消工作單元及取得不確定工作單元的鎖定所需的最舊日誌記錄，來建立回復日誌範圍。
2. 一旦決定日誌範圍，即會執行轉遞日誌讀取，以讓頁面設定進入最新狀態，並鎖定與不確定或進行中工作單元相關的任何訊息。
3. 當完成轉遞日誌讀取時，會往回讀取日誌，以取消失敗時進行中或進行中取消的任何工作單元。

z/OS 您可能會看到的訊息範例:

```

CSQR001I +MQOX RESTART INITIATED
CSQR003I +MQOX RESTART - PRIOR CHECKPOINT RBA=00000001E48C0A5E
CSQR004I +MQOX RESTART - UR COUNTS - 806
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN BACKOUT=0
CSQR030I +MQOX Forward recovery log range 815
from RBA=00000001E45FF7AD to RBA=00000001E48C1882
CSQR005I +MQOX RESTART - FORWARD RECOVERY COMPLETE - 816
IN COMMIT=0, INDOUBT=0
CSQR032I +MQOX Backward recovery log range 817
from RBA=00000001E48C1882 to RBA=00000001E48C1882
CSQR006I +MQOX RESTART - BACKWARD RECOVERY COMPLETE - 818
INFLIGHT=0, IN BACKOUT=0
CSQR002I +MQOX RESTART COMPLETED

```

註: 如果要讀取大量日誌，則會定期發出訊息 CSQR031I (正向回復) 及 CSQR033I (反向回復) 以顯示進度。

在第 506 頁的圖 86 中，IBM 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。

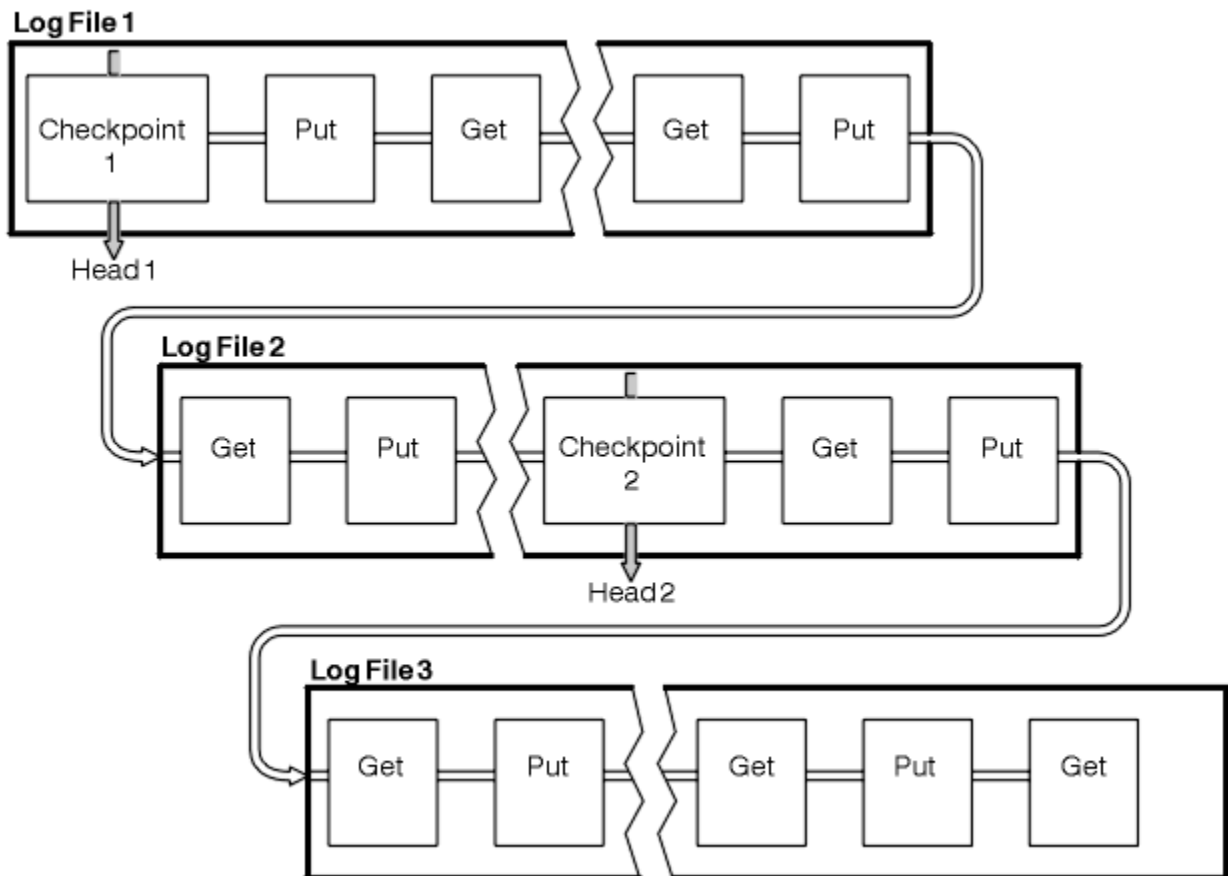


圖 86: 正在進行檢查點檢查作業

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 507 頁的圖 87 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，日誌標頭會邏輯移至最新記載的檢查點 Checkpoint 3。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，主要日誌檔完全已滿，則可以使用次要日誌檔來避免日誌已滿。

完全受佇列管理程式控制的活動 (例如檢查點檢查作業) 會排定為嘗試將活動保留在主要日誌內。

不過，當需要次要日誌空間來支援佇列管理程式控制之外的行為 (例如其中一個交易的持續時間) 時，佇列管理程式會嘗試使用任何已定義的次要日誌空間，以容許該活動完成。

如果該活動在使用日誌空間總計的 80% 之前未完成，則佇列管理程式會起始動作以收回日誌空間，而不論這對應用程式有何影響。

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。

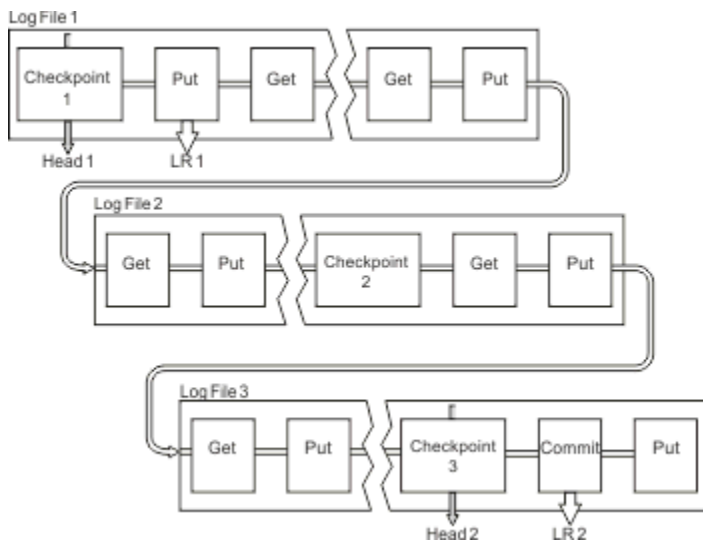


圖 87: 使用長時間執行的交易進行檢查點檢查作業

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是使用循環式記載還是線性記載之後，您需要預估佇列管理程式所需的 作用中日誌 大小。作用中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

附註:

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，但可能不會立即注意到您對次要日誌所做的變更所造成的影響。
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 系統上則不能超過 255，這會在長時間執行的交易存在時，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. 當使用循環式記載時，佇列管理程式會重複使用主要及次要日誌空間。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。

如需您需要配置的日誌數的相關資訊，請參閱第 508 頁的『我應該做多大的作用中日誌?』。主要日誌範圍依序使用，且該順序不會變更。

例如，如果您有三個主要日誌 0、1 及 2，則使用順序為 0、1、2 後接 1、2、0、2、0、1、回到 0、1、2 等。您已配置的任何次要日誌都會根據需要散置。

6. 在檢查點期間可重複使用主要日誌檔。在取得檢查點之前，佇列管理程式會同時考量主要及次要日誌空間，因為日誌空間量偏低。

V9.1.0 佇列管理程式會嘗試以在主要範圍內保留日誌使用情形的方式來排定檢查點。

如需相關資訊，請參閱第 81 頁的『mqsc.ini 檔案的 LogDefaults 段落』。

我應該做多大的作用中日誌?

預估佇列管理程式所需的作用中日誌大小。

作用中日誌的大小受到下列限制:

```
logsize = (primaryfiles + secondaryfiles) * logfilepages * 4096
```

當佇列管理程式將每秒資料數量上限寫入磁碟時，日誌應該夠大，足以應付執行最久的交易。

如果執行時間最長的交易執行 N 秒，且佇列管理程式每秒寫入磁碟的資料量上限為日誌中每秒 B 個位元組數，則您的日誌應該至少為:

```
logsize >= 2 * (N+1) * B
```

當您以尖峰工作量執行時，或當您記錄媒體映像檔時，佇列管理程式可能會將每秒資料數量上限寫入磁碟。

如果交易執行時間太長，以致於包含其第一個日誌記錄的日誌延伸範圍未包含在作用中日誌內，則佇列管理程式會從具有最舊日誌記錄的交易開始，一次回復一個作用中交易。

在使用主要及次要檔案數目上限之前，佇列管理程式需要將舊日誌範圍設為非作用中，且佇列管理程式需要配置另一個日誌範圍。

決定在容許佇列管理程式回復之前，您要執行最長執行交易的時間。您最長的執行中交易可能正在等待慢速網路資料流量，或者在設計欠佳的交易情況下，等待使用者輸入。

您可以發出下列 **runmqsc** 指令，來調查執行時間最長的交易執行時間:

```
DISPLAY CONN(*) UOWLOGDA UOWLOGTI
```

發出 **dspmqrtn -a** 指令時，會顯示處於所有狀態的所有 XA 及非 XA 指令。

發出此指令會列出針對所有現行交易寫入第一個日誌記錄的日期和時間。



小心: 基於計算日誌大小的目的，自寫入第一個日誌記錄以來的時間很重要，而不是自應用程式或交易啟動以來的時間。將執行時間最長的交易長度四捨五入至最接近的秒。這是因為佇列管理程式中的最佳化。

如果應用程式開始 (例如，發出 MQGET 呼叫，且在實際取得訊息之前等待一段時間)，則可以在應用程式啟動之後很久才寫入第一個日誌記錄。

透過檢閱來自的觀察日期和時間輸出上限

```
DISPLAY CONN(*) UOWLOGDA UOWLOGTI
```

您原先發出的指令，從現行日期和時間開始，您可以預估執行時間最長的交易執行時間。

請確保在最長執行交易以尖峰工作量執行時反覆地執行此 **runmqsc** 指令，以便您不會低估最長執行交易的長度。

在 IBM MQ 8.0 中，使用作業系統工具，例如 UNIX 平台上的 **iostat**。

從 IBM MQ 9.0 開始，您可以發出下列指令來探索佇列管理程式每秒寫入日誌的位元組數：

```
amqsrua -m qmgr -c DISK -t Log
```

寫入的邏輯位元組數，顯示佇列管理程式每秒寫入日誌的位元組數。例如：

```
$ amqsrua -m mark -c DISK -t Log
Publication received PutDate:20160920 PutTime:15383157 Interval:4 minutes,39.579 seconds
Log - bytes in use 37748736
Log - bytes max 50331648
Log file system - bytes in use 316243968
Log file system - bytes max 5368709120
Log - physical bytes written 4334030848 15501948/sec
Log - logical bytes written 3567624710 12760669/sec
Log - write latency 411 uSec
```

在此範例中，每秒寫入日誌的邏輯位元組數為 12760669/sec 或大約每秒 12 個 MiB。

使用

```
DISPLAY CONN(*) UOWLOGDA UOWLOGTI
```

顯示執行時間最長的交易是：

```
CONN(57E14F6820700069)
EXTCONN(414D51436D61726B2020202020202020)
TYPE(CONN)
APPLTAG(msginteg_r) UOWLOGDA(2016-09-20)
UOWLOGTI(16.44.14)
```

由於現行日期和時間是 2016-09-20 16.44.19，此交易已執行 5 秒。不過，在佇列管理程式回復交易之前，您需要容許交易執行 10 秒。因此您的日誌大小應該為：

```
2 * (10 + 1) * 12 = 264 MiB
```

日誌檔數目必須能夠包含最大預期日誌大小 (在前述文字中計算)。這將是：

日誌檔數目下限 = (必要日誌大小)/(LogFilePages * 日誌檔頁面大小 (4096))

使用在前述文字中計算的預設 LogFilePages(即 4096) 及日誌大小估計值 264MiB，日誌檔數目下限應該為：

```
264MiB / (4096 x 4096) = 16.5
```

即 17 個日誌檔。

如果您調整日誌大小，以便在主要檔內執行預期的工作量：

- 次要檔案會在需要額外日誌空間時提供一些緊急情況。
- 循環式記載一律使用預先配置的主要檔，這比配置及取消配置次要檔略快。
- 佇列管理程式只會使用主要檔中剩餘的空間來計算下一個檢查點的時間。

因此，在前述範例中，請設定下列值，讓工作量在主要日誌檔內執行：

- **LogFilePages** = 4096
- **LogPrimaryFiles** = 17
- **LogSecondaryFiles** = 5

請注意下列項目：

- 在此範例中，5 次要日誌超過作用中日誌空間的 20%。

V 9.1.0 從 IBM MQ 9.1.0 開始，日誌程式會嘗試單獨保留主要檔案中的工作量。因此，日誌程式會在僅部分主要檔案已滿時排定檢查點。

V 9.1.0 如果有任何非預期的長時間執行交易，則具有次要檔案是應急的。

您應該知道當日誌空間總計超過 80% 使用中時，佇列管理程式會採取動作來減少日誌空間使用情形。

- 不論您是使用線性或循環式記載，都執行相同的計算。

不論您是在計算線性作用中日誌或循環作用中日誌的大小，因為作用中日誌的概念在線性記載和循環記載中都是一樣的。

- 僅媒體回復所需的日誌範圍不在作用中日誌內，因此不會計入主要及次要檔案的數目。

- **V 9.1.0** 從 IBM MQ 9.1.0 中，[DISPLAY QMSTATUS LOG](#) 的 `LOGUTIL` 欄位可協助您計算所需的作用中日誌大小(大約)。

此欄位設計為可讓您對所需日誌大小進行合理預估，而無需持續取樣，以判定最長執行交易的持續時間，或佇列管理程式的尖峰傳輸量。

我應該做多大的 LogFile 頁面？

通常會讓 LogFile 頁面夠大，以便您可以輕鬆地增加作用中日誌的大小，而不會達到主要檔案數目上限。少數幾個大型日誌檔比許多小型日誌檔更適合，因為少數幾個大型日誌檔可讓您更有彈性地增加您需要的日誌大小。

對於線性記載，非常大的日誌檔可能會產生效能變數。使用非常大的日誌檔，有一個更大的步驟可建立及格式化新的日誌檔，或保存舊的日誌檔。這更多是手動及保存日誌管理的問題，因為使用自動日誌管理時，很少會建立新的日誌檔。

如果我讓我的日誌太小怎麼辦？

估計日誌大小下限時需要考量的點。

如果您讓日誌太小：

- 長時間執行的交易將會取消。
- 下一個檢查點想要在前一個檢查點結束之前開始。

重要：無論您如何不精確地估計日誌的大小，都會維護資料完整性。

如需檢查點的說明，請參閱第 504 頁的『[使用檢查點檢查作業來確保完整回復](#)』。如果作用中日誌範圍中剩餘的日誌空間量變得很短，佇列管理程式會更頻繁地排定檢查點。

檢查點需要一些時間量；它不是即時的。需要在檢查點中記錄的資料越多，檢查點所花費的時間越長。如果日誌是小型檢查點，則可能重疊，這表示在前一個檢查點結束之前，會要求下一個檢查點。如果發生此情況，則會寫入錯誤訊息。

如果長時間執行的交易取消，或檢查點重疊，佇列管理程式會繼續處理工作量。短期交易繼續正常執行。

不過，佇列管理程式未以最佳方式執行，效能可能會降低。您應該以足夠的日誌空間重新啟動佇列管理程式。

如果我讓我的日誌太大，會發生什麼？

估計日誌大小上限時需要考量的點。

如果您讓日誌太大：

- 您可能會增加緊急重新啟動所花費的時間，但這不太可能。
- 您正在使用不必要的磁碟空間。
- 可容忍長時間執行的交易。

重要：無論您如何不精確地估計日誌的大小，都會維護資料完整性。

V 9.1.0 為了協助您預估日誌的大小上限，您可以使用日誌使用率統計資料。如需進一步資訊，請參閱第 515 頁的『[決定如何設定 IMGLOGLN 和 IMGINTVL](#)』及 [ALTER QMGR](#)。

如需佇列管理程式在重新啟動時如何讀取日誌的說明，請參閱第 504 頁的『使用檢查點檢查作業來確保完整回復』。佇列管理程式會重播前次檢查點的日誌，然後解析佇列管理程式結束時處於作用中的所有交易。

為了解決交易，佇列管理程式會讀取回與該交易相關聯的所有日誌記錄。這些日誌記錄可能早於前次檢查點。

透過配置佇列管理程式一個非常大的日誌，您將在重新啟動時授與佇列管理程式讀取日誌中每一筆日誌記錄的許可權，不過通常佇列管理程式不需要這麼做。在發生這種情況的可能性不大的情況下，該程序可能需要很長時間。

如果檢查點檢查作業在佇列管理程式結束之前非預期地停止，則會大幅增加具有大型日誌之佇列管理程式的重新啟動時間。限制日誌大小會限制緊急重新啟動時間。

若要避免這些問題，您應該確保：

- 您的工作負載可以輕鬆地放入不是過大的日誌中。
- 您可以避免長時間執行交易。

V9.1.0 我應該建立多大的日誌檔案系統？

預估佇列管理程式所需的日誌檔案系統大小。

請務必讓日誌檔案系統足夠大，讓佇列管理程式有足夠空間來寫入其日誌。如果佇列管理程式完全填滿日誌檔案系統，它會寫入 FFDC、回復交易，且可能會突然終止佇列管理程式。

您為日誌保留的磁碟空間量必須至少與作用中日誌一樣大。確切的大小取決於：

- 您選擇的對數類型 (線性或循環)
- 作用中日誌 (主要檔案、次要檔案、日誌檔頁面) 的大小
- 您選擇的日誌管理 (手動、自動或保存)
- 你的應急計劃在一個受損物體的情況下。

如果您選擇循環日誌，則日誌檔案系統應該

```
LogFilesystemSize >= (PrimaryFiles + SecondaryFiles + 1) * LogFileSize
```

這可讓佇列管理程式寫入所有主要及次要檔案。在異常情況下，佇列管理程式可能會寫入超出次要數目的額外延伸範圍。之前的演算法會將此納入考量。

如果您選擇線性日誌，則日誌檔案系統應該比作用中日誌大很多。

如果您選擇手動日誌管理，佇列管理程式會在需要時繼續寫入新的日誌範圍，當不再需要時，您必須負責刪除它們 (並保存它們)。

日誌檔案系統需要大多少，很大程度上取決於您刪除多餘或非作用中延伸範圍的策略。

一旦延伸範圍變成非作用中 (重新啟動回復不需要)，您可能決定立即保存及刪除這些延伸範圍，或者您可能決定只保存及刪除多餘的延伸範圍 (媒體或重新啟動回復不需要)。

如果您只保存及刪除多餘的延伸範圍，且如果您有損壞的物件，**MEDIALOG** 將不會向前移動，因此不再有多餘的延伸範圍。在解決問題之前，您將停止保存及刪除延伸範圍，可能是透過回復物件。

除非您停止工作量，否則您必須解決此問題的時間取決於日誌檔案系統的大小。因此，在使用線性記載時，最佳作法是擁有慷慨的日誌檔案系統。

如果您選擇線性日誌及自動或保存日誌管理，則佇列管理程式會重複使用日誌範圍。

可重複使用的日誌範圍會以字母 R 作為字首。當記錄媒體映像檔時，由於保存多餘的延伸範圍，佇列管理程式隨後可以重複使用那些延伸範圍。

因此重複使用範圍小於在媒體映像檔之間寫入日誌的資料長度：

```
ReuseExtents <= LogDataLengthBetweenMediaImages
```

自動錄製媒體影像並設定 **IMGLOGLN** 時， `LogDataLengthBetweenMediaImages` 最多可以達到兩次 **IMGLOGLN**，因為 **IMGLOGLN** 不是固定上限的目標。

手動記錄媒體影像或依間隔自動記錄媒體影像時， `LogDataLengthBetweenMediaImages` 視您的工作量及取得影像之間的間隔而定。

除了作用中延伸範圍和重複使用延伸範圍之外，還有非作用中延伸範圍 (僅媒體回復需要) 和多餘延伸範圍 (重新啟動或媒體回復不需要)。

使用自動或保存日誌管理時，佇列管理程式不會重複使用媒體回復所需的延伸範圍。因此，非作用中延伸範圍的數目取決於您取得媒體影像的頻率，以及您是手動還是自動取得媒體影像。

IMGINTVL 和 **IMGLOGLN** 是目標，而不是媒體映像檔之間的固定下限或上限。不過，在估計您可能需要的日誌檔案系統大小上限時，自動媒體映像檔不可能分開記錄兩次以上 **IMGINTVL** 或 **IMGLOGLN**。

使用自動或保存日誌管理來調整日誌檔案系統大小時，您也應該考量佇列或其他物件損壞時可能發生的情況。在該情況下，佇列管理程式無法取得損壞物件的媒體影像，且 **MEDIALOG** 不會向前移動。

如果您的工作量繼續，您的非作用中日誌將無限制成長，因為仍需要媒體回復所需的最舊範圍，且無法重複使用。如果工作量繼續，在佇列管理程式開始回復交易甚至可能突然結束之前，您必須等到日誌檔案系統完全填滿以修正問題。

因此，對於自動及保存日誌管理：

```
LogFilesystemSize > (PrimaryFiles + SecondaryFiles +  
(((TimeBetweenMediaImages * 2) + TimeNeededToResolveDamagedObject) * ExtentsUsedPerHour))  
* LogFilePages
```

註：上述演算法假設只要媒體回復不再需要 **SET LOG ARCHIVED** 來進行保存日誌管理，就會針對每一個延伸範圍呼叫它。

管理日誌

V 9.1.0 從 IBM MQ 9.1.0，IBM MQ 支援線性日誌的自動日誌管理及自動媒體回復。循環日誌幾乎是自我管理，但有時需要人為介入來解決空間問題。

註：**IBM i** 自動及保存日誌管理在 IBM i 上無效。

在循環式記載上，佇列管理程式會收回日誌檔中釋放的空間。使用者不會看到此活動，而且您通常不會看到所使用的磁碟空間量減少，因為配置的空間會快速重複使用。

V 9.1.0 從 IBM MQ 9.1.0 開始，您可以在使用循環式記載時刪除次要檔。如需相關資訊，請參閱 **RESET QMGR TYPE (REDUCELOG)**。

在線性記載上，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。

Multi 如果日誌已滿，則會發出 AMQ7463 訊息。此外，如果由於長時間執行的交易導致無法釋放空間而導致日誌填滿，則會發出 AMQ7465 訊息。

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。

一段時間後，寫入的最舊日誌記錄會變成不需要重新啟動佇列管理程式。

當偵測到長時間執行的交易時，會排定活動以非同步回復該交易。如果由於某個非預期的原因，非同步回復失敗，則在該狀況下，部分 MQI 呼叫會傳回 `MQRC_RESOURCE_PROBLEM`。

請注意，已保留空間來確定或回復所有進行中交易，因此 **MQCMIT** 或 **MQBACK** 不應失敗。

以這種方式回復交易的應用程式無法執行在相同交易下指定同步點的後續 **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，會釋放日誌空間，且佇列管理程式會繼續正常運作。

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 AMQ6709 訊息，並取得錯誤記錄。

日誌檔會以固定大小建立，而不是在日誌記錄寫入時延伸。這表示只有在建立新檔案時，IBM MQ 才能用盡磁碟空間；在將記錄寫入日誌時，無法用盡空間。IBM 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。

V 9.1.0 從 IBM MQ 9.1.0 開始，當您使用線性記載時，您可以選擇使用：

- 自動管理日誌範圍。

如需新日誌屬性的相關資訊，請參閱 [DISPLAY QMSTATUS](#)。

另請參閱下列指令或其 PCF 對等項目：

- [重設佇列管理程式](#)
- 分散式平台的 [SET LOG](#)

- 控制媒體映像檔使用的選項。

如需下列相關資訊，請參閱 [ALTER QMGR](#) 指令及 [ALTER QUEUES](#)：

- IMGINTVL
- IMGLOGLN
- IMGRCOVO
- IMGRCOVQ
- IMGSCHEM

循環式記載會傳回資源問題。

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。佇列管理程式要求所有日誌範圍的大小都相同。

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。

循環式記載的特定資訊

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 81 頁的『mqsc.ini 檔案的 LogDefaults 段落』和第 108 頁的『qm.ini 檔案的日誌段落』）。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。

線性記載特定的資訊

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。下列方法決定仍需要哪些日誌檔：

日誌程式事件訊息

當發生重要事件（例如記錄媒體映像檔）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 [使用 MQSC 指令管理](#)，以及 PCF 指令的相關資訊，請參閱 [自動化作業](#)。

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 AMQ7467I 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 AMQ7468I 提供媒體回復所需的最舊日誌檔名稱。

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。

適用於這兩種記載類型的資訊

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。

如果找不到所需的任何日誌檔，則會發出操作員訊息 AMQ6767E。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。

自動清除日誌範圍-僅限線性記載

V 9.1.0 Multi

從 IBM MQ 9.1.0 開始，您可以選擇使用自動管理回復不再需要的線性日誌範圍。

您可以在 `qm.ini` 檔案的「日誌」段落中使用 **LogManagement** 屬性，或使用 IBM MQ Explorer 來設定自動管理。如需相關資訊，請參閱第 108 頁的『`qm.ini` 檔案的日誌段落』。

如需日誌作業的詳細資料，以及下列使用日誌的指令，請參閱 **DISPLAY QMSTATUS** 的 [LOG](#) 參數：

- [重設佇列管理程式](#)
- [設定日誌](#)

自動取得媒體影像-僅限線性記載

V 9.1.0

從 IBM MQ 9.1.0 開始，有一個整體交換器可控制佇列管理程式是否自動寫入媒體映像檔，預設值是尚未設定交換器。

您可以使用下列佇列管理程式屬性，來控制是否進行自動媒體影像處理，以及處理程序的頻率：

IMGSCHED

佇列管理程式是否自動寫入媒體影像

IMGINTVL

寫入媒體影像的頻率 (分鐘)

IMGLOGLN

自物件的前一個媒體映像檔以來寫入的日誌 MB 數。

如果您在當天的重要時間，工作量非常繁重，且您想要確定系統傳輸量不會受到自動媒體映像檔的影響，您可能想要設定 **IMGSCHED(MANUAL)** 來暫時關閉自動媒體映像檔。

您可以在工作量期間隨時切換 **IMGSCHED**。



小心：如果您沒有取得媒體映像檔，則不會向前移動 **MEDIALOG**，因此您需要保存延伸範圍，或確定您有足夠的磁碟空間。

您也可以控制其他使用者定義物件的自動及手動媒體影像：

- 鑑別資訊
- 通道
- 用戶端連線
- 接聽器
- 名稱清單

- 處理程序
- 別名佇列
- 本端佇列
- 服務
- 主題

對於內部系統物件 (例如物件型錄及佇列管理程式物件)，佇列管理程式會適當地自動寫入媒體影像。

如需屬性的相關資訊，請參閱 [ALTER QMGR](#)。

您也只能啟用或停用本端及永久動態佇列的自動及手動媒體映像檔。您可以使用 **IMGRCOVQ** 佇列屬性來執行此動作。

如需 **IMGRCOVQ** 屬性的相關資訊，請參閱 [ALTER QUEUES](#)。

附註:

1. 只有在使用線性記載時，才支援媒體映像檔。如果您已啟用自動媒體映像檔，但正在使用循環式記載，則會發出錯誤訊息，並停用佇列管理程式的自動媒體映像檔屬性。
2. 如果您已啟用自動媒體映像檔，但未指定日誌的頻率 (分鐘或 MB)，則會發出錯誤訊息，且不會寫入自動媒體映像檔。
3. 如果需要，您可以在設定 **IMGSCHEM(AUTO)** 時使用 `rcdmqimg` 來手動記錄媒體映像檔。
這可讓您在適合您企業的時間取得媒體映像檔，例如，當系統安靜時。自動媒體成像會考量這些手動媒體影像，因為採用手動媒體影像會重設間隔和日誌長度，在此之前會取得下一個自動媒體影像。
4. 從 IBM MQ 9.1.0 開始，佇列管理程式只會寫入媒體映像檔中的持續訊息，而不會寫入非持續訊息。這可以在移轉至 IBM MQ 9.1.0 或更新版本時減少媒體映像檔的大小

決定如何設定 **IMGLOGLN** 和 **IMGINTVL**

V 9.1.0

讓 **IMGLOGLN** 和 **IMGINTVL** 夠大，因此佇列管理程式只會花費一小部分時間來記錄媒體影像，但會足夠小，以便：

- 損壞的物件可以在合理的時間量內回復，以及
- 足夠小，讓您的日誌適合您的磁碟，而不會耗盡空間。

如果您設定 **IMGLOGLN**，良好的作法是讓 **IMGLOGLN** 成為佇列上資料量的數倍，以及工作量資料速率的數倍。您建立的 **IMGLOGLN** 越大，佇列管理程式記錄媒體影像所花費的時間就越短。

同樣地，如果您設定 **IMGINTVL**，則良好的作法是讓 **IMGINTVL** 乘以佇列管理程式記錄媒體映像檔所花費的時間量。您可以手動錄製媒體影像，以瞭解錄製媒體影像所需的時間。

如果您使 **IMGLOGLN** 和 **IMGINTVL** 太大，回復損壞物件可能需要很長的時間，因為必須重播自前次媒體映像檔以來的所有延伸範圍。

使 **IMGLOGLN** 和 **IMGINTVL** 足夠小，以便您可以接受回復損壞物件所花費的時間上限。

使 **IMGLOGLN** 和 **IMGINTVL** 變得非常大，表示日誌會變得非常大，因為媒體影像很少被記錄。



小心: 確保此大小的日誌適合您的日誌檔案系統，因為如果日誌檔案系統完全填滿，您的工作量將會取消。

您可以同時設定 **IMGINTVL** 和 **IMGLOGLN**。這可能有助於確保在繁重工作量 (由 **IMGLOGLN** 控制) 期間定期取得自動媒體映像檔，但在工作量非常輕 (由 **IMGINTVL** 控制) 時仍會偶爾取得自動媒體映像檔。

IMGINTVL 及 **IMGLOGLN** 是間隔及日誌資料長度的目標，在此間隔及日誌資料長度之間會取得自動媒體映像檔。

這些屬性不應視為固定上限或下限。事實上，如果佇列管理程式認為這是一個非常好的時間，則佇列管理程式可能會決定更快地排定自動媒體映像檔：

- 因為佇列是空的，所以從效能而言，取得媒體映像檔是最有效率的，而且

- 媒體映像檔已有一段時間未錄製

有時，自動媒體映像檔之間的時間間隔可能比 **IMGINTVL** 及/或 **IMGLOGLN** 的時間間隔稍長。

如果佇列上的資料量接近 **IMGLOGLN**，則媒體映像檔之間的時間間隔可能大於 **IMGLOGLN**。如果記錄媒體映像檔所花費的時間幾乎等於 **IMGINTVL**，則媒體映像檔之間的時間間隔可能大於 **IMGINTVL**。

這是不好的作法，因為佇列管理程式會花費大量時間來記錄媒體影像。

使用自動媒體映像檔記錄時，佇列管理程式會個別記錄每一個物件及佇列的媒體映像檔，因此佇列管理程式會針對每一個物件分別追蹤映像檔之間的時間間隔及日誌長度。

隨著時間的推移，媒體影像的記錄變得交錯，而不是同時為所有物件記錄媒體影像。這會驚人地展開記錄媒體影像的效能影響，並且是使用自動記錄媒體影像比手動記錄的另一個優點。

手動取得媒體映像檔-僅限線性記載

V 9.1.0

記錄佇列的媒體影像涉及將所有持續訊息從該佇列寫入日誌。對於包含大量訊息資料的佇列，這涉及將大量資料寫入日誌，而且此處理程序在發生時可能會影響系統效能。

記錄其他物件的媒體影像可能比較快速，因為其他物件的媒體影像不包含使用者資料。

您需要仔細考量何時記錄佇列的媒體映像檔，以便處理程序不會干擾尖峰工作量。

您必須定期記錄所有物件的媒體影像，以便更新媒體回復所需的最舊日誌範圍。

記錄佇列媒體影像的良好時機是在它是空的時，因為此時不會將任何訊息資料寫入日誌。相反地，當佇列非常深或有非常大的訊息時，則是不正確的時間。

記錄佇列媒體映像檔的良好時機是在系統安靜時；而不正確的時間是在尖峰工作量期間。例如，如果您的工作負載一律在午夜安靜，則您可能會決定在每晚午夜錄製媒體影像。

交錯每一個佇列的記錄可能會分散效能影響，因此減少其影響。自您前次記錄媒體映像檔以來，記錄媒體映像檔的時間越長，就越重要，因為媒體回復所需的日誌範圍數目越來越多。

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。

例如，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：

Windows 在 Windows

```
rcdmqimg -m QMNAME -t all *
```

Linux **UNIX** 在 UNIX and Linux

```
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 522 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**。

如果您選擇對佇列管理程式控制的自動媒體映像檔使用線性記載，則不需要使用 **rcdmqimg** 手動記錄媒體映像檔來管理日誌空間。

註：執行 **rcdmqimg** 指令時也可以發出 AMQ7467 及 AMQ7468 訊息。

局部媒體影像

V 9.1.0

較好的做法是只將 IBM MQ 訊息用於預期在不久的將來會耗用的資料，以便每則訊息在佇列上的時間量相對較短。

相反地，使用 IBM MQ 訊息來長期儲存資料 (例如資料庫) 是不理想的作法。

此外，最好是確保佇列相對淺層，而不理想的作法是具有深度佇列，其訊息已在佇列上很長時間。

遵循這些準則，可讓佇列管理程式將自動記錄媒體影像的效能最佳化。

記錄空佇列的媒體影像非常有效率 (從效能觀點來看)，而取得含有大量資料之佇列的媒體影像則非常沒有效率，因為所有這些資料都必須寫入媒體影像中的日誌。

對於具有最近放置訊息的淺層佇列，佇列管理程式可以進行進一步最佳化。

如果目前在佇列上的所有訊息都是最近放置的，則佇列管理程式可能能夠代表放置所有訊息之前的時間 (回復點) 來記錄媒體映像檔，因此能夠記錄空佇列的映像檔。此處理程序在效能方面是非常低的成本。

如果隨後已取得回復點在佇列上的所有訊息，則這些訊息不需要記錄在媒體影像中，因為它們已不在佇列上。

這稱為局部媒體映像檔。然後，在需要回復佇列的不太可能事件中，將會重播自最後一個媒體映像檔以來與此佇列相關的所有日誌記錄，因此會還原所有最近放置的訊息。

即使在回復點佇列中有一些訊息，目前在佇列中 (因此必須記錄在部分媒體映像檔中)，記錄這個較小的部分媒體映像檔仍比記錄所有訊息的完整媒體映像檔更有效率。

確保訊息在佇列上停留短暫的時間量，可能會增進自動記錄媒體影像的效能。

判斷多餘日誌檔-僅限線性記載

若為循環式記載，請永不從日誌目錄中刪除資料。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。

若要判定所需的最舊日誌檔，您可以使用三個位置：

- DISPLAY QMSTATUS 指令
- 日誌程式事件訊息及，最後
- 錯誤日誌訊息

對於 DISPLAY QMSTATUS 指令，若要判斷需要的最舊日誌範圍，請執行下列動作：

- 重新啟動佇列管理程式，並發出指令 DISPLAY QMSTATUS RECLG。
- 執行媒體回復，發出指令 DISPLAY QMSTATUS MEDIALOG。
- **V9.1.0** 判斷保存通知的名稱，發出指令 DISPLAY QMSTATUS ARCHLOG。

V9.1.0 使用循環式記載時，您可以透過發出指令 **RESET QMGR TYPE (REDUCELOG)** 來減少次要日誌範圍的數目。

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，否則在 10 年每天 3000 個日誌檔的訂單中，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 RECLG 值的任何日誌檔，也可以刪除數字小於 RECLG 及 MEDIALOG 值的任何日誌檔。



小心: 日誌檔折返，因此 9 999 999 之後的下一個數字是零。

日誌檔位置

選擇日誌檔的位置時，請記住，如果 IBM 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。

您應該將日誌檔放在與佇列管理程式資料不同的磁碟機上。

此裝置上的資料完整性至關重要-您應該容許內建備援。

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 IBM MQ 系統的前次備份。

V 9.1.3 Coldstart: 日誌範圍遺漏或毀損時要採取的動作

如果您的企業遺失重新啟動回復所需的部分或所有日誌範圍，佇列管理程式將無法重播回復日誌，因此無法重新啟動。如果您需要在回復日誌以任何方式毀損時重新啟動佇列管理程式，但代價是維護資料完整性，則可以這麼做，但強烈建議您不要這麼做。此處理程序稱為 *coldstarting* 佇列管理程式。

重要: Coldstarting a queue manager should only in exceptional exceptional. and hage data integrity ri 風險 as describing on this page. IBM 建議您重建佇列管理程式，最好是冷啟動，以回應毀損的資料檔案。

如果基於作業原因而需要冷啟動，請聯絡 IBM 支援代表，以檢閱問題的主要原因。您應該儘早以重建的佇列管理程式取代冷啟動的佇列管理程式。

coldstart 的效果

在 coldstart 上，佇列管理程式會建立空的回復日誌，並依賴佇列檔中的資料，以及處於其現有狀態的其他物件檔。因為佇列檔中的資料可能不一致，所以訊息可能遺失、重複、毀損或不一致。

佇列管理程式會將所有其他持續保存物件的配置儲存在回復日誌及物件檔中。其他內部狀態資料也會記錄在回復日誌中，因此在 coldstart 時，會重設內部狀態資料，且所有其他配置資料可能不正確。

冷啟動的影響無法預期且範圍廣泛，因此除非絕對必要，否則您應該避免冷啟動。在 coldstarting 之後，佇列和物件檔中的資訊可能不一致，以致於佇列管理程式完全不會重新啟動。

如果佇列管理程式確實重新啟動，則沒有簡單的方法來探索可以依賴哪些訊息資料或配置，以及哪些無法。此外，在冷啟動之後，佇列可能會損壞，因此完全無法使用。

此外，如果您可以從特定佇列取得或放入特定佇列，則其中的訊息可能已毀損、遺漏或重複。交易和通道可能無法確定。即使您的佇列管理程式已順利啟動，且佇列看起來完整，直到稍後才會實現 coldstart 的無法預期效果。

如果您需要 coldstart ，該怎麼做

執行冷啟動不應視為標準作業實務，IBM 強烈建議您不要這麼做。不過，如果您處於絕對需要冷啟動佇列管理程式的位置，請聯絡 [IBM MQ 支援](#)。

對於線性佇列管理程式而言，冷啟動佇列管理程式的處理程序比循環佇列管理程式要複雜得多。在 IBM MQ 9.1.3 中，coldstart 處理程序已大為簡化，不再需要複製或重新命名日誌範圍。

從 IBM MQ 9.1.3 中，聯絡「IBM 支援中心」，該支援中心會為您提供金鑰，您可以將該金鑰傳遞至 `strmqm` 指令，以冷啟動佇列管理程式。



小心: IBM MQ 9.1.3 coldstart 指令仍會面臨與手動 coldstart 相同的遺失資料完整性風險，而且 IBM 強烈建議您不這麼做。

消除未來冷啟動: 要求

`strmqm` 指令需要用於 coldstart 的索引鍵，因為 IBM MQ 希望您在需要 coldstart 時與 IBM MQ 支援中心聯絡，因為 IBM MQ 渴望瞭解您如何進入此狀況。

顯然，冷冰冰的做法是最好避免的。IBM MQ 已付出相當大的努力來確保您不需要對佇列管理程式進行冷啟動，而且 IBM 渴望探索產品是否還有其他可做的事，以減輕對冷啟動的需求。

避免冷啟動的預防措施

建立佇列管理程式時的預設記載方法是循環式記載。使用循環式記載，您容許佇列管理程式具有給定大小的特定數目的主要及次要日誌範圍。建立足夠大的日誌檔案系統，以包含所有主要及次要日誌範圍，而且您應該永不需要管理它們。

或者，您可以使用相對於循環的線性記載。線性記載可讓您在不太可能發生佇列及其他物件損壞的事件中，新增回復佇列及其他物件的能力。但依預設，線性記載需要您刪除重新啟動或媒體回復不再需要的日誌範圍。這稱為手動日誌管理。

以這種方式管理日誌範圍時，可能會不小心刪除太多日誌範圍，因此最終必須冷啟動。若要降低此風險，請使用自動日誌管理，以便佇列管理程式代表您管理日誌範圍。

最佳作法是將回復日誌放置在個別日誌檔案系統中，該檔案系統只包含回復日誌。如果您將回復日誌放置在與佇列管理程式其餘部分相同的檔案系統中，有時會發現檔案系統意外填滿，可能是因為佇列檔太大。將佇列管理程式的日誌目錄設為個別檔案系統，或在 `crtmqm` 指令上使用 `-ld` 指令行選項指定不同的日誌檔案系統。

如果保存佇列檔的檔案系統已滿，您可能無法放入那些佇列，但佇列管理程式會繼續執行。如果包含回復日誌的檔案系統已滿，則佇列管理程式會突然結束，且在您釋放部分空間之前不會重新啟動。

請小心不要刪除重新啟動回復所需的日誌範圍，否則您可能會發現自己需要冷啟動。有時您可能會發現需要冷啟動，因為包含其回復日誌的磁碟失敗。最佳作法是將回復日誌放置在抄寫的磁碟上，以降低磁碟損毀的風險。

將訊息及配置移至新的取代佇列管理程式，可避免先前已冷啟動的佇列管理程式持續發生問題。

請記下哪些佇列管理程式先前已冷啟動，即使它們已在很久之前冷啟動，且在此期間已停止、重新啟動及移轉。當您聯絡「IBM 支援中心」時，假設佇列管理程式先前已啟動 `coldstarted`，如果已啟動，請提供盡可能多的資訊，以瞭解導致 `coldstart` 需求的原因。

使用日誌進行回復

您可以使用日誌中的資訊來協助您從失敗中回復。

有數種方式可讓您的資料受損。IBM 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

IBM MQ 可以從通訊失敗及失去電源兩者中回復。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，持續訊息會保留在佇列上，直到接收端應用程式移除為止。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，IBM 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 IBM MQ 突然停止時，它們無法存活。

回復損壞物件

IBM 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

V 9.1.0 從 IBM MQ 9.1.0，在線性記載佇列管理程式上，只能記錄可回復物件的媒體映像檔。例如，您需要考量 `IMGRCOV0` 和 `IMGRCOVQ` 選項。

V 9.1.0 同樣地，您只能從其在線性記載佇列管理程式上的媒體映像檔回復物件子集 (定義為媒體可回復)。如果未定義為可回復媒體的物件損壞，則該物件的選項與循環式記載佇列管理程式的選項相同。

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其為媒體回復記錄；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 `rcdmqimg` 指令手動記錄媒體映像檔，如 `rcdmqimg` 中所述。此指令會寫入 IBM MQ 物件的媒體影像。

V 9.1.0 如果設定 `IMGSCHED(AUTO)`，佇列管理程式會自動記錄媒體映像檔。如需相關資訊，請參閱 `ALTER QMGR`，以取得 `IMGINTVL` 及 `INGLOGLN` 的相關資訊。

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 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 MQ 物件損壞，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。

存在哪些物件檔

V 9.1.0

佇列管理程式會將 `runmqsc` 中定義的物件屬性儲存在磁碟上的檔案中。這些物件檔位於佇列管理程式資料目錄下的子目錄中。

Linux **UNIX** 例如，在 UNIX 和 Linux 平台上，通道儲存在 `/var/mqm/qmgrs/qmgr/channel` 中。

這些物件檔中的資料是物件的媒體影像。如果這些物件檔被刪除或毀損，則儲存在該檔案中的物件會損壞。使用線性記載佇列管理程式，可以使用 `rcrmqobj` 指令從日誌中回復損壞物件。

大部分物件檔只包含物件的屬性，因此通道檔包含通道的屬性。異常狀況如下：

- 型錄
物件型錄會編目所有類型的所有物件，並儲存在 `qmanager/QMQMOBJCAT` 中。
- 同步檔
同步檔包含與所有通道相關聯的內部狀態資料。
- 佇列

佇列檔包含該佇列上的訊息以及該佇列的屬性。

請注意，**runmqsc** 或 IBM MQ Explorer 中沒有公開任何型錄或 **syncfile** 物件。

可以記錄型錄及佇列管理程式，但無法回復。如果這些物件損壞，則佇列管理程式會先結束，並在重新啟動時自動回復這些物件。

訂閱未列在要記錄或回復的物件中，因為可延續訂閱儲存在系統佇列上。若要記錄或回復可延續訂閱，請記錄或回復 **SYSTEM.DURABLE.SUBSCRIBER.QUEUE**。

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 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 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](#))。

V 9.1.0 如果佇列 (或其他物件) 損壞，**MEDIALOG** 將不會向前移動。這是因為 **MEDIALOG** 是媒體回復所需的最舊延伸範圍。如果您繼續執行工作量，**CURRLOG** 仍會繼續向前移動，因此會寫入新的延伸範圍。視您的配置 (包括 **LogManagement** 設定) 而定，這可能會開始填入日誌檔案系統。如果日誌檔案系統完全填滿，則會回復交易，且佇列管理程式可能會突然結束。因此當佇列損壞時，在佇列管理程式結束之前，您可能只有有限的時間可以採取行動。您有多少時間，取決於工作量導致佇列管理程式寫入新延伸範圍的速率，以及日誌檔案系統中的可用空間量。

V 9.1.0 如果您使用手動日誌管理，則可能正在保存重新啟動回復不需要的延伸範圍，然後從日誌檔案系統中刪除這些延伸範圍，即使媒體回復仍需要它們。只要您可以在需要時從保存檔中還原它們，即可接受此作業。當佇列損壞且 **MEDIALOG** 停止向前移動時，此原則不會導致日誌檔案系統填滿。不過，如果您只保存及刪除重新啟動或媒體回復不需要的延伸範圍，則在佇列損壞時，您的日誌檔案系統會開始填滿。

V 9.1.0 如果您使用自動或保存日誌管理，佇列管理程式將不會重複使用媒體回復仍需要的延伸範圍，即使您可能已保存它們並使用 **SET LOG** 保存來通知佇列管理程式。因此，如果佇列損壞，日誌檔案系統將會開始填入。

V 9.1.0 如果佇列損壞，您將寫入 **OBJECT DAMAGED FFDC** 且 **MEDIALOG** 停止向前移動。可以從 **FFDC** 中識別損壞的物件，或者因為當您在 **runmqsc** 中顯示其狀態時，它是具有最舊 **MEDIALOG** 的物件。

V 9.1.0 如果日誌檔案系統已填滿，且您擔心因為日誌檔案系統已滿而導致工作量取消，則回復物件或靜止工作量可能會停止發生此情況。

保護 IBM 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 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 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 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 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，使用下列其中一個選項來傾出更多日誌記錄，以變更傾出的起始位置：

- 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，您可以從最後一個 dmpmqlog 輸出指定 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 從任何給定日誌檔延伸範圍開始格式化日誌記錄。在此情況下，dmpmqlog 預期會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 指令格式化的第一件事。它包含下列欄位：

日誌作用中	主要日誌範圍的數目。
登入非作用中	次要日誌範圍數。
logsize	每個延伸範圍的 4 KB 頁面數。
baselsn	日誌範圍中包含日誌標頭的第一個 LSN。
下一個	要寫入的下一個日誌記錄的 LSN。
headlsn	日誌標頭上日誌記錄的 LSN。
尾	識別日誌尾端位置的 LSN。
hflag1	日誌是 CIRCULAR 還是 LOG RETAIN (線性)。
HeadExtentID	包含日誌標頭之日誌範圍。

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：

LSN	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> 表示僅限 IBM MQ 的交易。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 LSN (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) 的表示法，然後是訊息資料本身。

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。
-----------------	--

啟動交易

指出新交易的開始。MQI 的 TranType 指出僅限 IBM MQ 的交易。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 IBM 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 IBM MQ 物件的刪除。

ObjName 已刪除的物件名稱。

備份及還原 IBM MQ 佇列管理程式資料

您可以透過備份佇列管理程式及佇列管理程式資料、僅備份佇列管理程式配置，以及使用備份佇列管理程式，來保護佇列管理程式免受硬體故障所導致的可能毀損。

關於這項作業



注意: 如果您將佇列管理程式移至不同的作業系統，則必須非常小心。如需相關資訊，請參閱 [將佇列管理程式移至不同的作業系統](#)。

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。

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。

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，則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的最後一個完整日誌。

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。

程序

- 若要備份及還原佇列管理程式資料，請參閱：
 - 第 525 頁的『[備份佇列管理程式資料](#)』。
 - 第 526 頁的『[還原佇列管理程式資料](#)』。
- 若要備份及還原佇列管理程式配置，請參閱：
 - 第 527 頁的『[備份佇列管理程式配置](#)』
 - 第 527 頁的『[還原佇列管理程式配置](#)』
- 若要建立、更新及啟動備份佇列管理程式，請參閱 [第 528 頁的『使用備份佇列管理程式』](#)。

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


在開始備份佇列管理程式之前，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：

程序

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 72 頁的『在 Multiplatforms 上的 .ini 檔中變更 IBM MQ 配置資訊』](#)。
註：會轉換目錄中出現的名稱，以確保它們與您使用 IBM MQ 的平台相容。如需名稱轉換的相關資訊，請參閱 [瞭解 IBM 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 502 頁的『日誌的外觀』](#) 中說明的日誌控制檔，以及 [第 188 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。
對於循環性記載，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。
3. 保留檔案的所有權。

 若為 IBM MQ for UNIX 及 Linux 系統，您可以使用 **tar** 指令來執行此動作。(如果您有大於 2 GB 的佇列，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 IBM WebSphere MQ 7.5 以及更新版本時，請確保備份 **qm.ini** 檔案及登錄項目。佇列管理程式資訊儲存在 **qm.ini** 檔案中，可用來回復至舊版 IBM MQ。

相關工作

停止佇列管理程式

[第 13 頁的『在建立佇列管理程式之後備份配置檔』](#)

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

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

開始備份之前，請確定佇列管理程式不在執行中。

在叢集中還原佇列管理程式的備份時，如需相關資訊，請參閱 [第 301 頁的『回復叢集佇列管理程式』](#) 及 [叢集作業：可用性、多重實例及災難回復](#)。

註：當您升級至更新版本的 IBM MQ 時，請確定您已備份 **.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 IBM MQ。

程序

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。

請確定您有日誌控制檔以及日誌檔。

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。

4. 更新配置資訊檔案。

檢查 IBM MQ 與佇列管理程式配置檔是否一致，以便 IBM MQ 可以在正確的位置尋找還原的資料。

5. 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 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](#)。

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。

Multi 備份佇列管理程式配置

如果佇列管理程式配置及日誌資料因硬體故障而遺失，且佇列管理程式無法重新啟動或從日誌回復，則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。

關於這項作業

ULW 在 UNIX, Linux, and Windows 上，您可以使用 **dmpmqcfcg** 指令來傾出 IBM MQ 佇列管理程式的配置。

IBM i 在 IBM i 上，您可以使用「傾出 MQ 配置 (DMPMQMCFG)」指令來傾出佇列管理程式的配置物件及權限。

程序

1. 請確定佇列管理程式在執行中。
2. 視您的平台而定，請使用下列其中一個指令來備份佇列管理程式配置：

- **ULW** 在 UNIX, Linux, and 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 **dmpmqcfcg**，使用標準輸出重新導向將定義儲存至檔案。例如：

```
dmpmqcfcg -m MYQMGR -a > /mq/backups/MYQMGR.mqsc
```

- **IBM i** 在 IBM i 上: 使用 OUTPUT (*MQSC) 及 EXPATTR (*ALL) 的預設格式化選項執行「傾出 MQ 配置」指令 (**DMPMQMCFG**)，並使用 TOFILE 及 TOMBR 將定義儲存至實體檔案成員。例如：

```
DMPMQMCFG MQMNAME(MYQMGR) OUTPUT(*MQSC) EXPATTR(*ALL) TOFILE(QMQMSAMP/QMQSC)
TOMBR(MYQMGRDEF)
```

相關工作

第 527 頁的『還原佇列管理程式配置』

您可以先確定佇列管理程式正在執行中，然後執行適合您平台的指令，以從備份還原佇列管理程式的配置。

相關參考

[dmpmqcfcg \(傾出佇列管理程式配置\)](#)

[傾出 MQ 配置 \(DMPMQMCFG\)](#)

Multi 還原佇列管理程式配置

您可以先確定佇列管理程式正在執行中，然後執行適合您平台的指令，以從備份還原佇列管理程式的配置。

關於這項作業

ULW 在 UNIX, Linux, and Windows 上, 您可以使用 **runmqsc** 指令來還原 IBM MQ 佇列管理程式的配置。

IBM i 在 IBM i 上, 您可以使用 **STRMQMMQSC** 指令來還原佇列管理程式的配置物件及權限。

程序

1. 請確定佇列管理程式在執行中。

請注意, 如果透過其他方法無法復原資料及日誌的損壞, 則可能已重建佇列管理程式。

2. 視您的平台而定, 使用下列其中一個指令來還原佇列管理程式配置:

- **ULW** 在 UNIX, Linux, and Windows 上, 針對佇列管理程式執行 **runmqsc**, 使用標準輸入重新導向, 從「傾出 MQ 配置 (**dmpmqcfcg**)」指令所產生的 Script 檔還原定義 (請參閱 [第 527 頁的『備份佇列管理程式配置』](#))。例如:

```
runmqsc MYQMGR < /mq/backups/MYQMGR.mqsc
```

- **IBM i** 在 IBM i 上: 針對佇列管理程式執行 **STRMQMMQSC**, 並使用 **SRCMBR** 及 **SRCFILE** 參數, 從 Dump MQ Configuration (**DMPMQMCFG**) 指令所產生的實體檔成員中還原定義 (請參閱 [第 527 頁的『備份佇列管理程式配置』](#))。例如:

```
STRMQMMQSC MQMNAME(MYQMGR) SRCFILE(QMQMSAMP/QMQSC) SRCMBR(MYQMGR)
```

相關工作

[第 527 頁的『備份佇列管理程式配置』](#)

如果佇列管理程式配置及日誌資料因硬體故障而遺失, 且佇列管理程式無法重新啟動或從日誌回復, 則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。

相關參考

[dmpmqcfcg \(傾出佇列管理程式配置\)](#)

[runmqsc \(執行 MQSC 指令\)](#)

[傾出 MQ 配置 \(DMPMQMCFG\)](#)

[啟動 IBM MQ 指令 \(STRMQMMQSC\)](#)

使用備份佇列管理程式

現有佇列管理程式可以具有專用的備份佇列管理程式, 以進行災難回復。

關於這項作業

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, 才應該複製日誌檔; 請參閱 [第 529 頁的『更新備份佇列管理程式』](#), 以取得如何確保不再寫入特定日誌檔的相關資訊, 以便安全地複製它。

註: 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。啟動備份佇列管理程式之後, 便無法再更新它。

重要: 一旦舊備份佇列管理程式變成新的作用中佇列管理程式，不論任何原因，都不再有備份佇列管理程式。這實際上是非同步抄寫的一種形式，因此新的作用中佇列管理程式預期在邏輯上落後舊的作用中佇列管理程式一段時間。因此，舊的作用中佇列管理程式不再充當新作用中佇列管理程式的備份。

程序

- 如需使用備份佇列管理程式的相關資訊，請參閱下列主題：
 - [第 529 頁的『建立備份佇列管理程式』](#)
 - [第 529 頁的『更新備份佇列管理程式』](#)
 - [第 530 頁的『啟動備份佇列管理程式』](#)

相關概念

[第 502 頁的『記載: 確定訊息不會遺失』](#)

IBM MQ 會將佇列管理程式所控制之持續資料的所有重要變更記錄在回復日誌中。

建立備份佇列管理程式

您可以建立備份佇列管理程式作為現有佇列管理程式的非作用中副本。

關於這項作業

重要: 使用線性記載時，您只能使用備份佇列管理程式。

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。

程序

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。
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 525 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 **strmqm** 控制指令，如下列範例所示：

```
strmqm -r BackupQMName
```

這個指令會將佇列管理程式標示為「IBM 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。

相關參考

[crtmqm \(建立佇列管理程式\)](#)

[strmqm \(啟動佇列管理程式\)](#)

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。

關於這項作業

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。在備份之前，不需要停止佇列管理程式。



警告: 如果您將一組非連續的日誌複製到備份佇列管理程式日誌目錄，則只會重播截至找到第一個遺漏日誌的點為止的日誌。

程序

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) [第 530 頁的『2』](#) 中指出的現行範圍。只複製以 "S .." 開頭の日誌延伸範圍檔案。

4. 在備份佇列管理程式上執行 **strmqm** 控制指令，如下列範例所示:

```
strmqm -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。

相關參考

[RESET QMGR](#)

[DISPLAY QMSTATUS](#)

[strmqm \(啟動佇列管理程式\)](#)

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。

關於這項作業

在叢集中還原佇列管理程式的備份時，如需相關資訊，請參閱 [第 301 頁的『回復叢集佇列管理程式』](#) 及 [叢集作業: 可用性、多重實例及災難回復](#)。

如果無法復原的佇列管理程式有專用的備份佇列管理程式，您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。

當以備份佇列管理程式取代無法復原的佇列管理程式時，來自無法復原之佇列管理程式的部分佇列管理程式資料可能會遺失。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。

註: 即使佇列管理程式資料及日誌檔保留在不同的目錄中，請確定您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。即使啟動，您的資料也可能毀損。

程序

1. 執行 **strmqm** 控制指令，以啟動備份佇列管理程式，如下列範例所示:

```
strmqm -a BackupQMName
```

已啟動備份佇列管理程式。既然它已在作用中，就無法再更新備份佇列管理程式。

2. 執行 **strmqm** 控制指令以啟動備份佇列管理程式，如下列範例所示:

```
strmqm BackupQMName
```

IBM MQ 會將此視為重新啟動回復，並使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期間，將會進行重播，因此只會回復前次記錄檢查點的作用中交易。

3. 重新啟動所有通道。

4. 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 MQ 目錄和子目錄的相關資訊，請參閱 [規劃檔案系統支援](#)。

5. 請確定您有日誌控制檔以及日誌檔。另請檢查 IBM MQ 與佇列管理程式配置檔是否一致，以便 IBM MQ 可以在正確的位置尋找還原的資料。

結果

如果已正確備份及還原資料，則佇列管理程式現在會啟動。

相關工作

第 182 頁的『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。

相關參考

[strmqm \(啟動佇列管理程式\)](#)

叢集錯誤回復的變更 (在 z/OS 以外的伺服器上)

從 IBM WebSphere MQ 7.1 開始，佇列管理程式會重新執行導致問題的作業，直到解決問題為止。如果在五天之後仍無法解決問題，佇列管理程式會關閉，以防止快取變得更過時。

在 IBM WebSphere MQ 7.1 之前，如果佇列管理程式偵測到管理叢集的本端儲存庫管理程式有問題，則會更新錯誤日誌。在某些情況下，它會停止管理叢集。佇列管理程式繼續與叢集交換應用程式訊息，並依賴叢集定義的快取越來越過時。從 IBM WebSphere MQ 7.1 開始，佇列管理程式會重新執行導致問題的作業，直到解決問題為止。如果在五天之後仍無法解決問題，佇列管理程式會關閉，以防止快取變得更過時。隨著快取變得越來越過時，它會導致更多問題。7.1 或更新版本中關於叢集錯誤的變更行為不適用於 z/OS。

本端儲存庫管理程式處理程序 `amqrrmfa` 會針對佇列管理程式處理叢集管理的每個層面。即使沒有叢集定義，處理程序也會在所有佇列管理程式上執行。

在 IBM WebSphere MQ 7.1 之前，如果佇列管理程式在本端儲存庫管理程式中偵測到問題，則會在短暫間隔之後停止儲存庫管理程式。佇列管理程式會持續執行、處理應用程式訊息及開啟佇列的要求，以及發佈或訂閱主題。

當儲存庫管理程式停止時，佇列管理程式可用的叢集定義快取會變得更過時。一段時間後，訊息遞送至錯誤目的地，且應用程式失敗。應用程式無法嘗試開啟尚未延伸到本端佇列管理程式的叢集佇列或發佈主題。

除非管理者檢查錯誤日誌中的儲存庫訊息，否則管理者可能不會發現叢集配置有問題。如果在更長的時間內無法辨識失敗，且佇列管理程式未更新其叢集成員資格，則會發生更多問題。不穩定會影響叢集中的所有佇列管理程式，且叢集似乎不穩定。

從 IBM WebSphere MQ 7.1 開始，IBM MQ 採用不同的方法來處理叢集錯誤。儲存庫管理程式會重新執行失敗的作業，而不是停止儲存庫管理程式而繼續執行。如果佇列管理程式偵測到儲存庫管理程式的問題，它會遵循兩個動作的其中一個。

1. 如果錯誤不會危及佇列管理程式的作業，佇列管理程式會將訊息寫入錯誤日誌。它會每 10 分鐘重新執行一次失敗的作業，直到作業成功為止。依預設，您有五天時間處理錯誤；如果失敗，佇列管理程式會將訊息寫入錯誤日誌，並關閉。你可以推遲五天休班。
2. 如果錯誤危及佇列管理程式的作業，佇列管理程式會將訊息寫入錯誤日誌，並立即關閉。

危及佇列管理程式作業的錯誤是佇列管理程式無法診斷的錯誤，或是可能有無法預期結果的錯誤。這種類型的錯誤通常會導致佇列管理程式寫入 FFST 檔案。危及佇列管理程式作業的錯誤可能是由 IBM MQ 中的錯誤、管理者或程式執行非預期的動作 (例如結束 IBM MQ 處理程序) 所造成。

錯誤回復行為中的變更點是限制佇列管理程式繼續執行的時間，且不一致叢集定義的數目會增加。隨著叢集定義中不一致的數目增加，異常應用程式行為的機會也隨之增加。

在五天之後關閉佇列管理程式的預設選項，是在限制不一致數目與在偵測並解決問題之前保持佇列管理程式可用之間的折衷方案。

您可以在修正問題或等待計劃的佇列管理程式關閉時，延長佇列管理程式無限期關閉之前的時間。五天停留可讓佇列管理程式持續執行到長週末，讓您有時間在重新啟動佇列管理程式之前回應任何問題或延長時間。

更正動作

您可以選擇動作來處理叢集錯誤回復的問題。第一個選擇是監視並修正問題，第二個選擇是監視並延遲修正問題，最後一個選擇是繼續管理叢集錯誤回復，如 IBM WebSphere MQ 7.1 之前的版本。

1. 監視佇列管理程式錯誤日誌，以取得錯誤訊息 [AMQ9448](#) 及 [AMQ5008](#)，並修正問題。

[AMQ9448](#) 指出在執行指令之後，儲存庫管理程式已傳回錯誤。此錯誤會標示每 10 分鐘重試一次指令的開始，最後會在五天之後停止佇列管理程式，除非您延遲關閉。

[AMQ5008](#) 指出佇列管理程式已停止，因為遺漏 IBM MQ 處理程序。[AMQ5008](#) 儲存庫管理程式在五天之後停止所產生的結果。如果儲存庫管理程式停止，則佇列管理程式會停止。

2. 請監視佇列管理程式錯誤日誌中的錯誤訊息 [AMQ9448](#)，並延遲修正問題。

如果您停用從 `SYSTEM.CLUSTER.COMMAND.QUEUE` 取得訊息，儲存庫管理程式會停止嘗試執行指令，並無限期地繼續進行，而不處理任何工作。不過，會釋放儲存庫管理程式對佇列所保留的任何控點。因為儲存庫管理程式未停止，所以佇列管理程式在五天之後不會停止。

執行 `MQSC` 指令以停用從 `SYSTEM.CLUSTER.COMMAND.QUEUE` 取得訊息：

```
ALTER QLOCAL(SYSTEM.CLUSTER.COMMAND.QUEUE) GET(DISABLED)
```

若要回復從 `SYSTEM.CLUSTER.COMMAND.QUEUE` 接收訊息，請執行 `MQSC` 指令：

```
ALTER QLOCAL(SYSTEM.CLUSTER.COMMAND.QUEUE) GET(ENABLED)
```

3. 將佇列管理程式回復至 IBM WebSphere MQ 7.1 之前的相同叢集錯誤回復行為。

您可以設定佇列管理程式調整參數，以在儲存庫管理程式停止時保持佇列管理程式執行。

調整參數是 `TolerateRepositoryFailure`，位於 `qm.ini` 檔案的 `TuningParameters` 段落中。

若要防止佇列管理程式停止，如果儲存庫管理程式停止，請將 `TolerateRepositoryFailure` 設為 `TRUE`；請參閱第 532 頁的圖 88。

重新啟動佇列管理程式以啟用 `TolerateRepositoryFailure` 選項。

如果發生叢集錯誤，導致儲存庫管理程式無法順利啟動，因此佇列管理程式無法啟動，請將 `TolerateRepositoryFailure` 設為 `TRUE`，以在沒有儲存庫管理程式的情況下啟動佇列管理程式。

特殊考量

在 IBM WebSphere MQ 7.1 之前，部分管理不屬於叢集的佇列管理程式的管理者會停止 `amqrrmfa` 處理程序。停止 `amqrrmfa` 不會影響佇列管理程式。

在 IBM WebSphere MQ 7.1 或更新版本中停止 `amqrrmfa` 會導致佇列管理程式停止，因為它被視為佇列管理程式失敗。除非您設定佇列管理程式調整參數 `TolerateRepositoryFailure`，否則您不得在 7.1 或更新版本中停止 `amqrrmfa` 程序。

範例

```
TuningParameters:  
TolerateRepositoryFailure=TRUE
```

圖 88: 在 `qm.ini` 中，將 `TolerateRepositoryFailure` 設為 `TRUE`

相關概念

[佇列管理程式配置檔 `qm.ini`](#)

配置 JMS 資源

JMS 應用程式可以建立及配置連接至 IBM MQ 所需的資源，以及存取傳送或接收訊息的目的地的方法之一，是使用「Java 命名和目錄介面 (JNDI)」，從命名和目錄服務 (稱為 JNDI 名稱空間) 內的位置擷取受管理物件。在 JMS 應用程式可以從 JNDI 名稱空間擷取受管理物件之前，您必須先建立並配置受管理物件。

關於這項作業

您可以使用下列任一工具，在 IBM MQ 中建立及配置受管理物件：

IBM MQ Explorer

您可以使用 IBM MQ Explorer 來建立及管理儲存在 LDAP、本端檔案系統或其他位置中的 JMS 物件定義。

IBM MQ JMS 管理工具

IBM MQ JMS 管理工具是指令行工具，可用來建立及配置儲存在 LDAP、本端檔案系統或其他位置中的 IBM MQ JMS 物件。JMS 管理工具使用類似於 `runmqsc` 的語法，並且支援 Scripting。

管理工具使用配置檔來設定特定內容的值。提供了配置檔範例，您可以在使用該工具來配置 JMS 資源之前，先編輯該配置檔以符合您的系統。如需配置檔的相關資訊，請參閱第 539 頁的『[配置 JMS 管理工具](#)』。

部署至 WebSphere Application Server 的 IBM MQ JMS 應用程式需要從應用程式伺服器 JNDI 儲存庫存取 JMS 物件。因此，如果您在 WebSphere Application Server 與 IBM MQ 之間使用 JMS 傳訊，則必須在 WebSphere Application Server 中建立對應於您在 IBM MQ 中建立之物件的物件。

IBM MQ Explorer 及 IBM MQ JMS 管理工具無法用來管理儲存在 WebSphere Application Server 中的 IBM MQ JMS 物件。相反地，您可以使用下列任一工具，在「WebSphere Application Server」中建立及配置受管理物件：

WebSphere Application Server 管理主控台

WebSphere Application Server 管理主控台是 Web 型工具，可用來管理 WebSphere Application Server 中的 IBM MQ JMS 物件。

WebSphere Application Server wsadmin Scripting 用戶端

WebSphere Application Server wsadmin Scripting 用戶端提供特殊化指令來管理 WebSphere Application Server 中的 IBM MQ JMS 物件。

如果您想要使用 JMS 應用程式，從 WebSphere Application Server 內存取 IBM MQ 佇列管理程式的資源，您必須使用 WebSphere Application Server 中的 IBM MQ 傳訊提供者，其中包含 IBM MQ classes for JMS 版本。所有利用 IBM MQ 傳訊提供者來執行 JMS 傳訊的應用程式，都會使用 WebSphere Application Server 所提供的 IBM MQ 資源配接器。當您套用 WebSphere Application Server 修正套件時，通常會自動更新 IBM MQ 資源配接器，但如果您先前已手動更新資源配接器，則必須手動更新配置，以確保正確套用維護。

相關工作

[撰寫 IBM MQ classes for JMS 應用程式](#)

相關參考

[runmqsc](#)

在 JNDI 名稱空間中配置 Connection Factory 和目的地

JMS 應用程式會透過「Java 命名和目錄介面 (JNDI)」來存取命名和目錄服務中的受管理物件。JMS 受管理物件儲存在命名和目錄服務 (稱為 JNDI 名稱空間) 內的位置。JMS 應用程式可以查閱受管理物件，以連接至 IBM MQ 並存取用於傳送或接收訊息的目的地。

關於這項作業

JMS 應用程式會使用環境定義，在命名及目錄服務中查閱 JMS 物件的名稱：

起始環境定義

起始環境定義定義 JNDI 名稱空間的根。對於命名和目錄服務中的每一個位置，您需要指定起始環境定義，以提供起始點，讓 JMS 應用程式可以在命名和目錄服務的該位置中解析受管理物件的名稱。

子環境定義

環境定義可以有一或多個子環境定義。子環境定義是 JNDI 名稱空間的一部分，可以包含 Connection Factory 和目的地等受管理物件，以及其他子環境定義。子環境定義本身不是一個物件；它只是子環境定義中物件命名慣例的一種延伸。

您可以使用 IBM MQ Explorer 或 IBM MQ JMS 管理工具來建立環境定義。

在 IBM MQ classes for JMS 應用程式可以從 JNDI 名稱空間擷取受管理物件之前，您必須先使用 IBM MQ Explorer 或 IBM MQ JMS 管理工具來建立受管理物件。您可以建立及配置下列類型的 JMS 物件：

Connection Factory

JMS Connection Factory 物件會定義一組連線的標準配置內容。JMS 應用程式使用 Connection Factory 來建立與 IBM MQ 的連線。您可以建立其中一個傳訊網域 (點對點傳訊網域和發佈/訂閱傳訊網域) 專用的 Connection Factory。或者，從 JMS 1.1，您可以建立與網域無關的 Connection Factory，以同時用於點對點和發佈/訂閱傳訊。

目的地

JMS 目的地是一個物件，代表用戶端產生的訊息目標，以及 JMS 應用程式耗用的訊息來源。JMS 應用程式可以使用單一目的地物件來放置訊息及從中取得訊息，或者應用程式可以使用個別目的地物件。目的地物件有兩種類型：

- 點對點傳訊所用的 JMS 佇列目的地
- 發佈/訂閱傳訊所用的 JMS 主題目的地

下圖顯示在 IBM MQ JNDI 名稱空間中建立的 JMS 物件範例。

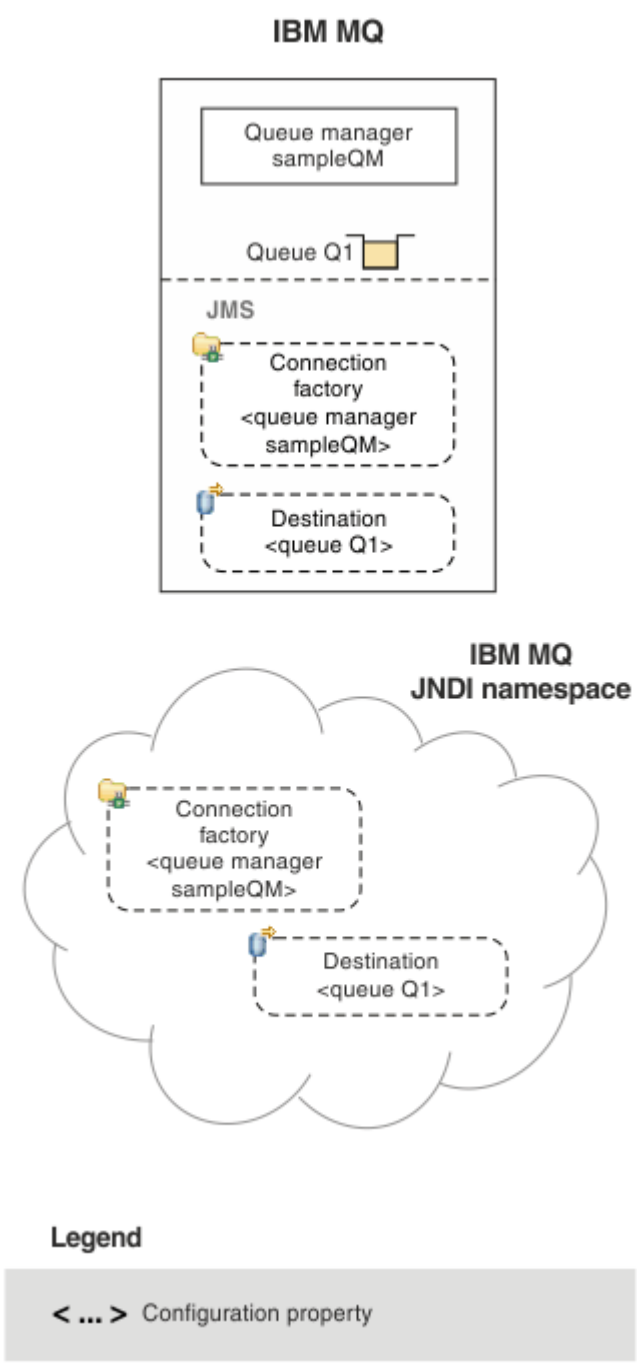


圖 89: IBM MQ 中建立的 JMS 物件

如果您在 WebSphere Application Server 與 IBM MQ 之間使用 JMS 傳訊，則必須在 WebSphere Application Server 中建立對應的物件，以用來與 IBM MQ 通訊。當您在 WebSphere Application Server 中建立其中一個物件時，它會儲存在 WebSphere Application Server JNDI 名稱空間中，如下圖所示。

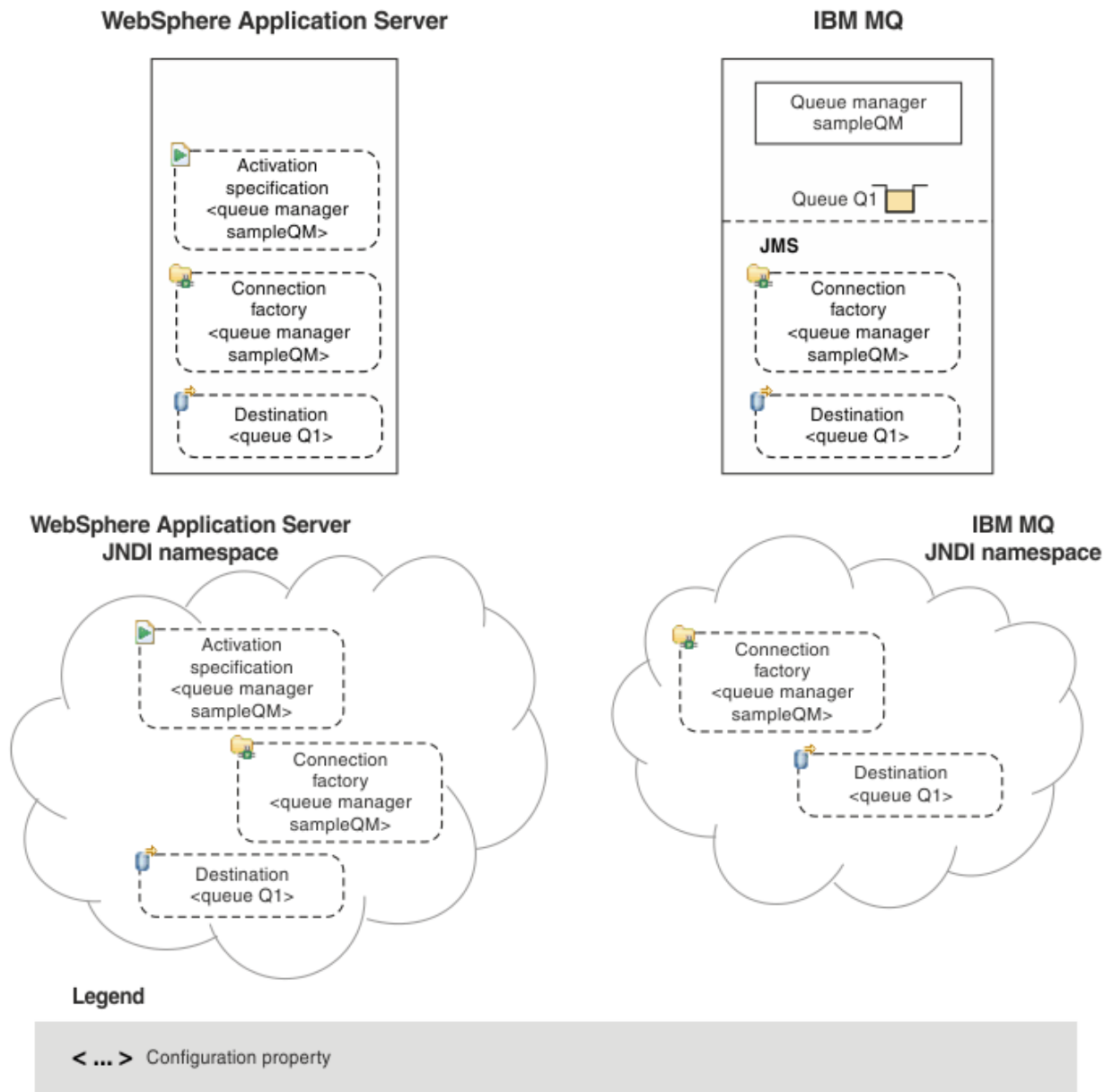


圖 90: 在 *WebSphere Application Server* 中建立的物件，以及在 *IBM MQ* 中建立的對應物件

如果您的應用程式使用訊息驅動 Bean (MDB)，則 Connection Factory 只會用於出埠訊息，且啟動規格會接收入埠訊息。啟動規格是 Java EE Connector Architecture 1.5 (JCA 1.5) 標準的一部分。JCA 1.5 提供標準方法來整合 JMS 提供者 (例如 IBM MQ) 與 Java EE 應用程式伺服器 (例如 WebSphere Application Server)。JMS 啟動規格可以與一或多個訊息驅動 Bean (MDB) 相關聯，並提供這些 MDB 接聽抵達目的地之訊息所需的配置。

您可以使用 WebSphere Application Server 管理主控台或 wsadmin Scripting 指令，來建立及配置您需要的 JMS 資源。

程序

- 若要使用 IBM MQ Explorer 為 IBM MQ 配置 JMS 物件，請參閱第 537 頁的『[使用 IBM MQ Explorer 來配置 JMS 物件](#)』。
- 若要使用 IBM MQ JMS 管理工具來配置 IBM MQ 的 JMS 物件，請參閱第 538 頁的『[使用管理工具來配置 JMS 物件](#)』。

- 若要為 WebSphere Application Server 配置 JMS 物件，請參閱 [第 545 頁的『在 WebSphere Application Server 中配置 JMS 資源』](#)。

結果

IBM MQ classes for JMS 應用程式可以從 JNDI 名稱空間擷取受管理物件，並在必要時使用 IBM JMS 延伸或 IBM MQ JMS 延伸來設定或變更其一或多個內容。

相關工作

使用 [JNDI 來擷取 JMS 應用程式中的受管理物件](#)

在 [IBM MQ classes for JMS 應用程式中建立及配置 Connection Factory 和目的地](#)

使用 IBM MQ Explorer 來配置 JMS 物件

使用 IBM MQ Explorer 圖形使用者介面，從 IBM MQ 物件建立 JMS 物件，從 JMS 物件建立 IBM MQ 物件，以及管理及監視其他 IBM MQ 物件。

關於這項作業

IBM MQ Explorer 是圖形使用者介面，您可以在其中管理及監視 IBM MQ 物件，不論它們是由本端電腦管理還是在遠端系統上管理。「IBM MQ Explorer」可在 Windows 及 Linux x86-64 上執行。它可以從遠端連接至在任何受支援平台 (包括 z/OS) 上執行的佇列管理程式，讓您可以從主控台檢視、探索及變更整個傳訊骨幹。

在 IBM MQ Explorer 中，所有 Connection Factory 都會儲存在適當環境定義和子環境定義的 Connection Factory 資料夾中。

您可以使用 IBM MQ Explorer 執行下列類型的作業，從 IBM MQ Explorer 中的現有物件或從建立新物件精靈內進行環境定義：

- 從下列任何 IBM MQ 物件建立 JMS Connection Factory：
 - 不論在本端電腦或遠端系統上，都是 IBM MQ 佇列管理程式。
 - IBM MQ 通道。
 - IBM MQ 接聽器。
- 使用 JMS Connection Factory，將 IBM MQ 佇列管理程式新增至 IBM MQ Explorer。
- 從 IBM MQ 佇列建立 JMS 佇列。
- 從 JMS 佇列建立 IBM MQ 佇列。
- 從 IBM MQ 主題建立 JMS 主題，它可以是 IBM MQ 物件或動態主題。
- 從 JMS 主題建立 IBM MQ 主題。

程序

- 啟動 IBM MQ Explorer (如果它尚未在執行中)。如果 IBM MQ Explorer 正在執行並顯示「歡迎使用」頁面，請關閉「歡迎使用」頁面，以開始管理 IBM MQ 物件。
- 如果您尚未這樣做，請建立起始環境定義來定義 JNDI 名稱空間的根目錄，其中 JMS 物件會儲存在命名和目錄服務中。

將起始環境定義新增至「IBM MQ Explorer」後，您就可以在 JNDI 名稱空間中建立 Connection Factory 物件、目的地物件和子環境定義。

起始環境定義會顯示在「Navigator」視圖的「JMS 受管理物件」資料夾中。請注意，雖然會顯示 JNDI 名稱空間的完整內容，但在 IBM MQ Explorer 中，您只能編輯儲存在該處的 IBM MQ classes for JMS 物件。如需相關資訊，請參閱 [新增起始環境定義](#)。
- 建立及配置您需要的子環境定義和 JMS 受管理物件。

如需相關資訊，請參閱 [建立及配置 JMS 受管理物件](#)。
- 配置 IBM MQ。

如需相關資訊，請參閱 [使用 IBM MQ Explorer 來配置 IBM MQ](#)。

相關概念

[IBM MQ Explorer 簡介](#)

相關工作

[在 IBM MQ classes for JMS 應用程式中建立及配置 Connection Factory 和目的地](#)

使用管理工具來配置 JMS 物件

您可以使用 IBM MQ JMS 管理工具來定義八種 IBM MQ classes for JMS 物件類型的內容，並將它們儲存在 JNDI 名稱空間內。然後，應用程式可以使用 JNDI 從名稱空間擷取這些受管理物件。

關於這項作業

下表顯示您可以使用動詞來建立、配置及操作的八種受管理物件類型。「關鍵字」直欄顯示您可以在 [第 538 頁的表 37](#) 顯示的指令中替代 *TYPE* 的字串。

物件類型	關鍵字	說明
MQConnectionFactory	CF	JMS ConnectionFactory 介面的 IBM MQ 實作。這代表在點對點和發佈/訂閱網域中建立連線的 Factory 物件。
MQQueueConnectionFactory	QCF	JMS QueueConnectionFactory 介面的 IBM MQ 實作。這代表在點對點網域中建立連線的 Factory 物件。
MQTopicConnectionFactory	TCF	JMS TopicConnectionFactory 介面的 IBM MQ 實作。這代表在發佈/訂閱網域中建立連線的 Factory 物件。
MQQUEUE	Q	JMS 佇列介面的 IBM MQ 實作。這代表點對點網域中訊息的目的地。
MQTopic	T	JMS 主題介面的 IBM MQ 實作。這代表發佈/訂閱網域中訊息的目的地。
MQXAConnectionFactory 第 538 頁的『1』	XACF	JMS XAConnectionFactory 介面的 IBM MQ 實作。這代表在點對點和發佈/訂閱網域中建立連線的 Factory 物件，以及連線使用 JMS 類別 XA 版本的位置。
MQXAQueueConnectionFactory 第 538 頁的『1』	XAQCF	JMS XAQueueConnectionFactory 介面的 IBM MQ 實作。這代表一個 Factory 物件，用來在使用 JMS 類別 XA 版本的點對點網域中建立連線。
MQXATopicConnectionFactory 第 538 頁的『1』	XATCF	JMS XATopicConnectionFactory 介面的 IBM MQ 實作。這代表一個 Factory 物件，用來在使用 JMS 類別 XA 版本的發佈/訂閱網域中建立連線。

註:

1. 這些類別是提供給應用程式伺服器供應商使用。它們不太可能直接對應用程式設計師有用。

如需如何配置這些物件的相關資訊，請參閱 [第 544 頁的『配置 JMS 物件』](#)。

您需要使用此工具的內容類型及值列在 [IBM MQ classes for JMS 物件的內容](#)中。

您也可以使用此工具來操作 JNDI 內的目錄名稱空間子環境定義，如 [第 542 頁的『配置子環境定義』](#) 中所述。

您也可以使用 IBM MQ Explorer 來建立及配置 JMS 受管理物件。

相關工作

在 [IBM MQ classes for JMS 應用程式中建立及配置 Connection Factory 和目的地](#)
使用 JNDI 來擷取 JMS 應用程式中的受管理物件

配置 JMS 管理工具

IBM MQ JMS 管理工具使用配置檔來設定特定內容的值。提供範例配置檔，您可以編輯它以符合您的系統。

關於這項作業

配置檔是一個純文字檔，由一組鍵值組組成，以等號 (=) 區隔。您可以透過設定配置檔中定義的三個內容的值來配置管理工具。下列範例顯示這三個內容：

```
#Set the service provider
INITIAL_CONTEXT_FACTORY=com.sun.jndi.ldap.LdapCtxFactory
#Set the initial context
PROVIDER_URL=ldap://polaris/o=ibm_us,c=us
#Set the authentication type
SECURITY_AUTHENTICATION=none
```

(在此範例中，行第一個直欄中的雜湊符號 (#) 指出註解，或未使用的行。)

IBM MQ 隨附範例配置檔 (用作預設配置檔)。範例檔稱為 `JMSAdmin.config`，位於 `MQ_JAVA_INSTALL_PATH/bin` 目錄中。您可以編輯此範例檔來定義系統所需的設定，或建立您自己的配置檔。

當您啟動管理工具時，可以使用 `-cfg` 指令行參數來指定您要使用的配置檔，如第 540 頁的『[啟動管理工具](#)』中所述。如果您在呼叫工具時未指定配置檔名稱，則工具會嘗試載入預設配置檔 (`JMSAdmin.config`)。它會先在現行目錄中搜尋此檔案，然後在 `MQ_JAVA_INSTALL_PATH/bin` 目錄中搜尋，其中 `MQ_JAVA_INSTALL_PATH` 是 IBM MQ classes for JMS 安裝的路徑。


儲存在 LDAP 環境中的 JMS 物件名稱必須符合 LDAP 命名慣例。其中一個慣例是物件和環境定義名稱必須包含字首，例如 `cn=` (通用名稱) 或 `ou=` (組織單位)。管理工具可讓您參照不含字首的物件和環境定義名稱，以簡化 LDAP 服務提供者的使用。如果您未提供字首，工具會自動將預設字首新增至您提供的名稱。若為 LDAP，這是 `cn=`。必要的話，您可以在配置檔中設定 **NAME_PREFIX** 內容來變更預設字首。

註：您可能需要配置 LDAP 伺服器來儲存 Java 物件。如需相關資訊，請參閱 LDAP 伺服器的說明文件。

程序

1. 透過配置 **INITIAL_CONTEXT_FACTORY** 內容，定義工具使用的服務提供者。

此內容支援的值如下：

- `com.sun.jndi.ldap.LdapCtxFactory` (適用於 LDAP)
- `com.sun.jndi.fscontext.RefFSContextFactory` (適用於檔案系統環境定義)
-  `com.ibm.jndi.LDAPCtxFactory` 僅在 z/OS 上受支援，並提供 LDAP 伺服器的存取權。不過，這個類別與 `com.sun.jndi.ldap.LdapCtxFactory` 不相容，因為使用一個 `InitialContextFactory` 建立的物件無法使用另一個來讀取或修改。

您也可以使用管理工具，利用 `JMSAdmin` 配置檔中定義的三個參數來連接其他 JNDI 環境定義。如果要使用不同的 `InitialContextFactory`，請執行下列動作：

- a) 將 **INITIAL_CONTEXT_FACTORY** 內容設為必要的類別名稱。
- b) 使用 **USE_INITIAL_DIR_CONTEXT**、**NAME_PREFIX** 和 **NAME_READABILITY_MARKER** 內容來定義 `InitialContextFactory` 的行為。
這些內容的設定在範例配置檔註解中說明。

如果您使用其中一個支援的 **INITIAL_CONTEXT_FACTORY** 值，則不需要定義 **USE_INITIAL_DIR_CONTEXT**、**NAME_PREFIX** 和 **NAME_READABILITY_MARKER** 內容。不過，如果

您想要置換系統預設值，您可以提供值給這些內容。比方說，如果您的物件儲存在 LDAP 環境中，您可以將 **NAME_PREFIX** 內容設為必要的字首，來變更工具新增至物件和環境定義名稱的預設字首。

如果您省略三個 InitialContextFactory 內容中的一或多個，則管理工具會根據其他內容的值提供適當的預設值。

2. 透過配置 **PROVIDER_URL** 內容，定義階段作業起始環境定義的 URL。

這個 URL 是工具所執行之所有 JNDI 作業的根目錄。支援兩種形式的此內容：

- ldap://hostname/contextname
- 檔案: [drive:] /pathname

LDAP URL 的格式可能有所不同，視您的 LDAP 提供者而定。如需相關資訊，請參閱 LDAP 文件。

3. 透過配置 **SECURITY_AUTHENTICATION** 內容，定義 JNDI 是否將安全認證傳遞至服務提供者。

只有在使用 LDAP 服務提供者時，才會使用這個內容，且可以採用下列三個值之一：

none (匿名鑑別)

如果您將這個參數設為 none，JNDI 不會將任何安全認證傳遞給服務提供者，且會執行 *anonymous authentication*。

simple (簡式鑑別)

如果您將參數設為 simple，安全認證會透過 JNDI 傳遞至基礎服務提供者。這些安全認證採用使用者識別名稱 (使用者 DN) 和密碼的形式。

CRAM-MD5 (CRAM-MD5 鑑別機制)

如果您將參數設為 CRAM-MD5，安全認證會透過 JNDI 傳遞至基礎服務提供者。這些安全認證採用使用者識別名稱 (使用者 DN) 和密碼的形式。

如果您未提供有效的 **SECURITY_AUTHENTICATION** 內容值，內容會預設為 none。

如果需要安全認證，當工具起始設定時，系統會提示您提供安全認證。若要避免此問題，您可以在 JMSAdmin 配置檔中設定 **PROVIDER_USERDN** 和 **PROVIDER_PASSWORD** 內容。

註: 如果您不使用這些內容，則輸入的文字包括密碼會回應至畫面。這可能有安全含意。

工具本身不會鑑別；鑑別作業會委派給 LDAP 伺服器。LDAP 伺服器管理者必須設定並維護對目錄不同部分的存取權。如需相關資訊，請參閱 LDAP 文件。如果鑑別失敗，工具會顯示適當的錯誤訊息並終止。

如需安全和 JNDI 的詳細資訊，請參閱 Oracle 的 Java 網站 ([Oracle Technology Network for Java Developers](http://www.oracle.com/technetwork/java/javadevelopers/))。

啟動管理工具

管理工具具有指令行介面，可讓您以互動方式使用，或用來啟動批次程序。

關於這項作業

互動模式提供命令提示字元，您可以在其中輸入管理指令。在批次模式中，啟動工具的指令包括包含管理指令 Script 的檔案名稱。

程序

互動模式

- 若要以互動模式啟動工具，請輸入下列指令：

```
JMSAdmin [-t] [-v] [-cfg config_filename]
```

其中：

-t

啟用追蹤 (預設值為關閉追蹤)

追蹤檔在 "%MQ_JAVA_DATA_PATH%\errors 中產生 (Windows) 或 /var/mqm/trace (UNIX)。追蹤檔的名稱格式如下：

```
mqjms_PID.trc
```

其中 *PID* 是 JVM 的程序 ID。

-v

產生詳細輸出 (預設值為 Terse 輸出)

-cfg config_filename

命名替代配置檔。如果省略此參數，則會使用預設配置檔 `JMSAdmin.config`。如需配置檔的相關資訊，請參閱第 539 頁的『配置 JMS 管理工具』。

即會顯示命令提示字元，指出工具已準備好接受管理指令。此提示最初顯示為：

```
InitCtx>
```

指出現行環境定義 (亦即，所有命名和目錄作業目前所參照的 JNDI 環境定義) 是 **PROVIDER_URL** 配置參數中所定義的起始環境定義。如需此參數的相關資訊，請參閱第 539 頁的『配置 JMS 管理工具』。

當您遍訪目錄名稱空間時，提示會變更以反映此情況，因此提示一律會顯示現行環境定義。

批次模式

- 若要以批次模式啟動工具，請輸入下列指令：

```
JMSAdmin test.scp
```

其中 *test.scp* 是包含管理指令的 Script 檔。如需相關資訊，請參閱第 541 頁的『使用管理指令』。檔案中的最後一個指令必須是 END 指令。

使用管理指令

管理工具接受由管理動詞及其適當參數組成的指令。

關於這項作業

下表列出在使用管理工具輸入指令時可以使用的管理動詞。

動詞	簡短形式	說明
ALTER	ALT	至少變更受管理物件的其中一個內容
定義	DEF	建立及儲存受管理物件，或建立子環境定義
顯示畫面	DIS	顯示一或多個儲存的受管理物件的內容，或現行環境定義的內容
DELETE	DEL	從名稱空間中移除一或多個受管理物件，或移除空的子環境定義
CHANGE	chg	變更現行環境定義，容許使用者遍訪起始環境定義下方任何位置的目錄名稱空間 (擱置安全許可範圍)
COPY	CP	製作已儲存受管理物件的副本，並以替代名稱儲存它
移動	MV	變更儲存受管理物件的名稱
END		關閉管理工具

程序

- 如果管理工具尚未啟動，請依照第 540 頁的『啟動管理工具』中的說明來啟動它。即會顯示命令提示字元，指出工具已準備好接受管理指令。此提示最初顯示為：

```
InitCtx>
```

若要變更現行環境定義，請使用 **CHANGE** 動詞，如 [第 542 頁的『配置子環境定義』](#) 中所述。

- 以下列格式輸入指令：

```
verb [param]*
```

其中 **verb** 是 [第 541 頁的表 38](#) 中列出的其中一個管理動詞。所有有效的指令都包含一個動詞，它以標準或簡短格式出現在指令開頭。動詞名稱不區分大小寫。

- 若要終止指令，請按 Enter 鍵，除非您要一起輸入數個指令，在此情況下，請在按 Enter 鍵之前直接鍵入加號 (+)。

通常，若要終止指令，請按 Enter 鍵。不過，您可以在按 Enter 鍵之前直接鍵入加號 (+) 來置換此值。這可讓您輸入多行指令，如下列範例所示：

```
DEFINE Q(BookingsInputQueue) +
QMGR(QM.POLARIS.TEST) +
QUEUE(BOOKINGS.INPUT.QUEUE) +
PORT(1415) +
CCSID(437)
```

- 若要關閉管理工具，請使用 **END** 動詞。
這個動詞不能採用任何參數。

配置子環境定義

您可以使用動詞 **CHANGE**、**DEFINE**、**DISPLAY** 和 **DELETE** 來配置目錄名稱空間子環境定義。

關於這項作業

下表說明這些動詞的用法。

指令語法	說明
DEFINE CTX (ctxName)	嘗試建立現行環境定義的子項子環境定義，名稱為 ctxName。如果存在安全違規、子環境定義已存在或提供的名稱無效，則失敗。
DISPLAY CTX	顯示現行環境定義的內容。受管理物件以 a 標註，子環境定義以 [D] 標註。也會顯示每一個物件的 Java 類型。
DELETE CTX (ctxName)	嘗試刪除名稱為 ctxName 的現行環境定義子項環境定義。如果找不到環境定義、不是空的，或有安全違規，則失敗。
CHANGE CTX (ctxName)	變更現行環境定義，以便它現在參照名稱為 ctxName 的子項環境定義。可以提供 ctxName 的兩個特殊值之一： = UP 移至現行環境定義的母項 = INIT 直接移至起始環境定義 如果指定的環境定義不存在或存在安全違規，則失敗。

儲存在 LDAP 環境中的 JMS 物件名稱必須符合 LDAP 命名慣例。其中一個慣例是物件和環境定義名稱必須包含字首，例如 cn= (通用名稱) 或 ou= (組織單位)。管理工具可讓您參照不含字首的物件和環境定義名稱，以簡化 LDAP 服務提供者的使用。如果您未提供字首，工具會自動將預設字首新增至您提供的名稱。若為 LDAP，這是 cn=。必要的話，您可以在配置檔中設定 **NAME_PREFIX** 內容來變更預設字首。如需相關資訊，請參閱 [第 539 頁的『配置 JMS 管理工具』](#)。

註：您可能需要配置 LDAP 伺服器來儲存 Java 物件。如需相關資訊，請參閱 LDAP 伺服器的說明文件。

建立 JMS 物件

如果要建立 JMS Connection Factory 和目的地物件，並將它們儲存在 JNDI 名稱空間中，請使用 DEFINE 動詞。若要将物件儲存在 LDAP 環境中，您必須提供符合特定使用慣例的名稱。管理工具可透過將預設字首新增至物件名稱，來協助您遵循 LDAP 命名慣例。

關於這項作業

DEFINE 動詞會以您指定的類型、名稱和內容來建立受管理物件。新物件會儲存在現行環境定義中。

儲存在 LDAP 環境中的 JMS 物件名稱必須符合 LDAP 命名慣例。其中一個慣例是物件和環境定義名稱必須包含字首，例如 cn= (通用名稱) 或 ou= (組織單位)。管理工具可讓您參照不含字首的物件和環境定義名稱，以簡化 LDAP 服務提供者的使用。如果您未提供字首，工具會自動將預設字首新增至您提供的名稱。若為 LDAP，這是 cn=。必要的話，您可以在配置檔中設定 **NAME_PREFIX** 內容來變更預設字首。如需相關資訊，請參閱第 539 頁的『配置 JMS 管理工具』。

註：您可能需要配置 LDAP 伺服器來儲存 Java 物件。如需相關資訊，請參閱 LDAP 伺服器的說明文件。

程序

1. 如果管理工具尚未啟動，請依照第 540 頁的『啟動管理工具』中的說明來啟動它。
即會顯示命令提示字元，指出工具已準備好接受管理指令。
2. 請確定命令提示字元顯示您要在其中建立新物件的環境定義。
當您啟動管理工具時，提示一開始會顯示為：

```
InitCtx>
```

若要變更現行環境定義，請使用 CHANGE 動詞，如第 542 頁的『配置子環境定義』中所述。

3. 如果要建立 Connection Factory、佇列目的地或主題目的地，請使用下列指令語法：

```
DEFINE TYPE (name) [property]*
```

也就是說，輸入 DEFINE 動詞，後面接著 TYPE (name) 受管理物件參照，後面接著零個以上內容 (請參閱 [IBM MQ classes for JMS 物件的內容](#))。

4. 如果要建立 Connection Factory、佇列目的地或主題目的地，請使用下列指令語法：

```
DEFINE TYPE (name) [property]*
```

5. 若要顯示新建立的物件，請使用具有下列指令語法的 DISPLAY 動詞：

```
DISPLAY TYPE (name)
```

範例

下列範例顯示使用 DEFINE 動詞在起始環境定義中建立的佇列 testQueue。因為此物件儲存在 LDAP 環境中，所以雖然未以字首輸入物件名稱 testQueue，但工具會自動新增一個，以確保符合 LDAP 命名慣例。提交指令 DISPLAY Q(testQueue) 也會導致新增此字首。

```
InitCtx> DEFINE Q(testQueue)
InitCtx> DISPLAY CTX
Contents of InitCtx
a cn=testQueue          com.ibm.mq.jms.MQQueue
1 Object(s)
0 Context(s)
1 Binding(s), 1 Administered
```

建立 JMS 物件的錯誤條件範例

當您建立物件時，可能會發生一些常見錯誤狀況。

以下是這些錯誤狀況的範例：

CipherSpec 對映至 CipherSuite

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) SSLCIPHERSUITE(RC4_MD5_US)
WARNING: Converting CipherSpec RC4_MD5_US to
CipherSuite SSL_RSA_WITH_RC4_128_MD5
```

物件的內容無效

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) PRIORITY(4)
Unable to create a valid object, please check the parameters supplied
Invalid property for a QCF: PRI
```

內容值的類型無效

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) CCSID(english)
Unable to create a valid object, please check the parameters supplied
Invalid value for CCS property: English
```

內容衝突-用戶端/bin 指定

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) HOSTNAME(polaris.hursley.ibm.com)
Unable to create a valid object, please check the parameters supplied
Invalid property in this context: Client-bindings attribute clash
```

內容衝突-結束起始設定

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) SECEXITINIT(initStr)
Unable to create a valid object, please check the parameters supplied
Invalid property in this context: ExitInit string supplied
without Exit string
```

內容值超出有效範圍

```
InitCtx/cn=Trash> DEFINE Q(testQ) PRIORITY(12)
Unable to create a valid object, please check the parameters supplied
Invalid value for PRI property: 12
```

不明內容

```
InitCtx/cn=Trash> DEFINE QCF(testQCF) PIZZA(ham and mushroom)
Unable to create a valid object, please check the parameters supplied
Unknown property: PIZZA
```

以下是從 JMS 應用程式查閱 JNDI 管理物件時，在 Windows 上可能發生的錯誤狀況範例。

1. 如果您使用 WebSphere JNDI 提供者 `com.ibm.websphere.naming.WsnInitialContextFactory`，則必須使用正斜線 (/) 來存取子環境定義中所定義的受管理物件；例如，`jms/MyQueue` 名稱。如果您使用反斜線 (\)，則會擲出 `InvalidName` 異常狀況。
2. 如果您使用 Oracle JNDI 提供者 `com.sun.jndi.fscontext.RefFSContextFactory`，則必須使用反斜線 (\) 來存取子環境定義中定義的受管理物件；例如 `ctx1\\fred`。如果您使用正斜線 (/)，則會擲出 `NameNotFoundException`。

配置 JMS 物件

您可以使用動詞 ALTER、DEFINE、DISPLAY、DELETE、COPY 及 MOVE 來操作目錄名稱空間中的受管理物件。

關於這項作業

第 545 頁的表 40 彙總這些動詞的使用。將 *TYPE* 替換為代表所需受管理物件的關鍵字，如第 538 頁的『使用管理工具來配置 JMS 物件』中所述。

指令語法	說明
ALTER <i>TYPE</i> (name) [property] *	嘗試以提供的內容來更新受管理物件的內容。如果有安全違規、找不到指定的物件，或提供的新內容無效，則失敗。
DEFINE <i>TYPE</i> (名稱) [內容] *	嘗試使用提供的內容建立 <i>TYPE</i> 類型的受管理物件，並將它儲存在現行環境定義中的名稱 name 下。如果有安全違規、所提供的名稱無效或該名稱的物件存在，或所提供的內容無效，則失敗。
DISPLAY <i>TYPE</i> (名稱)	顯示 <i>TYPE</i> 類型的受管理物件的內容，在現行環境定義中以 name 名稱連結。如果物件不存在或存在安全違規，則失敗。
DELETE <i>TYPE</i> (名稱)	嘗試從現行環境定義中移除類型為 <i>TYPE</i> 且名為 name 的受管理物件。如果物件不存在或存在安全違規，則失敗。
COPY <i>TYPE</i> (nameA) <i>TYPE</i> (nameB)	製作 <i>TYPE</i> 類型之受管理物件的副本，名為 nameA，並命名副本 nameB。這全都在現行環境定義的範圍內發生。如果要複製的物件不存在、名為 nameB 的物件存在或存在安全違規，則失敗。
MOVE <i>TYPE</i> (nameA) <i>TYPE</i> (nameB)	將類型為 <i>TYPE</i> 且名為 nameA 的受管理物件移至 (重新命名) nameB。這全都在現行環境定義的範圍內發生。如果要移動的物件不存在、名為 nameB 的物件存在或存在安全違規，則失敗。

在 WebSphere Application Server 中配置 JMS 資源

如果要在 WebSphere Application Server 中配置 JMS 資源，您可以使用管理主控台或 wsadmin 指令。

關於這項作業

Java Message Service (JMS) 應用程式通常依賴外部配置的物件，這些物件說明應用程式如何連接至其 JMS 提供者及其存取的目的地。JMS 應用程式在執行時期使用 Java Naming Directory Interface (JNDI) 來存取下列類型的物件：

- 啟動規格 (由 Java EE 應用程式伺服器使用)
- 統一 Connection Factory (使用 JMS 1.1，相較於網域特定的 Queue Connection Factory 和 Topic Connection Factory，偏好獨立於網域的 (統一) Connection Factory)
- Topic Connection Factory (由 JMS 1.0 應用程式使用)
- Queue Connection Factory (供 JMS 1.0 應用程式使用)
- 佇列
- 主題

透過 WebSphere Application Server 中的 IBM MQ 傳訊提供者，Java Message Service (JMS) 傳訊應用程式可以使用 IBM MQ 系統作為 JMS 傳訊資源的外部提供者。若要啟用此方法，請在 WebSphere Application Server 中配置 IBM MQ 傳訊提供者，以定義 JMS 資源來連接至 IBM MQ 網路上的任何佇列管理程式。

您可以使用 WebSphere Application Server 來配置應用程式 (例如 Queue Connection Factory) 的 IBM MQ 資源，以及管理與 JMS 目的地相關聯的訊息和訂閱。您可以透過 IBM MQ 來管理安全。

相關工作

[同時使用 IBM MQ 和 WebSphere Application Server](#)

WebSphere Application Server 主題

[使用 IBM MQ 傳訊提供者的交互作業](#)

[使用 IBM MQ 傳訊提供者管理傳訊](#)

[管理主控台畫面名稱至指令名稱和 IBM MQ 名稱的對映](#)

使用管理主控台來配置 JMS 資源

您可以使用 WebSphere Application Server 管理主控台來配置 IBM MQ JMS 提供者的啟動規格、Connection Factory 和目的地。

關於這項作業

您可以使用 WebSphere Application Server 管理主控台來建立、檢視或修改下列任何資源：

- 啟動規格
- 與網域無關的 Connection Factory (JMS 1.1 或更新版本)
- Queue Connection Factory
- Topic Connection Factory
- 佇列
- 主題

下列步驟提供您可以使用管理主控台來配置 JMS 資源，以與 IBM MQ 傳訊提供者搭配使用之方式的概觀。每一個步驟都包括 WebSphere Application Server 產品說明文件中的主題名稱，如需相關資訊，請參閱。如需 WebSphere Application Server 產品說明文件中這些主題的鏈結，請參閱 相關鏈結。

在混合版本 WebSphere Application Server Cell 中，您可以在所有版本的節點上管理 IBM MQ 資源。不過，部分內容並非在所有版本上都可用。在此狀況下，管理主控台只會顯示該特定節點的內容。

程序

如果要建立或配置啟動規格，以便與 IBM MQ 傳訊提供者搭配使用，請執行下列動作：

- 如果要建立啟動規格，請使用「建立 IBM MQ JMS」資源精靈。
您可以使用精靈來指定啟動規格的所有詳細資料，也可以選擇使用用戶端通道定義表 (CCDT) 來指定 IBM MQ 的連線詳細資料。當您使用精靈指定連線詳細資料時，您可以選擇個別輸入主機及埠資訊，或者如果您使用多重實例佇列管理程式，則可以選擇以連線名稱清單形式輸入主機及埠資訊。如需相關資訊，請參閱 建立 IBM MQ 傳訊提供者的啟動規格。
- 如果要檢視或變更啟動規格的配置內容，請使用管理主控台 IBM MQ 傳訊提供者 Connection Factory 設定畫面。
這些配置內容控制如何建立與相關聯佇列及主題的連線。如需相關資訊，請參閱 配置 IBM MQ 傳訊提供者的啟動規格。

如果要建立或配置統一 Connection Factory、Queue Connection Factory 或 Topic Connection Factory，以便與 IBM MQ 傳訊提供者搭配使用，請執行下列動作：

- 如果要建立 Connection Factory，請先選取您要建立的 Connection Factory 類型，然後使用「建立 IBM MQ JMS 資源」精靈來指定詳細資料。
 - 如果您的 JMS 應用程式預期只使用點對點傳訊，請為點對點傳訊網域建立網域特定的 Connection Factory，以用來建立專門用於點對點傳訊的連線。
 - 如果您的 JMS 應用程式只預期使用發佈/訂閱傳訊，請為發佈/訂閱傳訊網域建立網域特定的 Connection Factory，以用來建立專門用於發佈/訂閱傳訊的連線。
 - 若為 JMS 1.1 或更新版本，請建立可同時用於點對點傳訊和發佈/訂閱傳訊的獨立網域 Connection Factory，讓您的應用程式在相同交易下同時執行點對點和發佈/訂閱工作。

您可以選擇是否使用精靈來指定 Connection Factory 的所有詳細資料，也可以選擇使用用戶端通道定義表 (CCDT) 來指定 IBM MQ 的連線詳細資料。當您使用精靈指定連線詳細資料時，您可以選擇個別輸入主機及埠資訊，或者如果您使用多重實例佇列管理程式，則可以選擇以連線名稱清單形式輸入主機及埠資訊。如需相關資訊，請參閱 建立 IBM MQ 傳訊提供者的 Connection Factory。

如果要檢視或變更 Connection Factory 的配置內容，請執行下列動作：

- 針對您要配置的 Connection Factory 類型，使用管理主控台 Connection Factory 設定畫面。
配置內容控制如何建立與相關聯佇列及主題的連線。如需相關資訊，請參閱 [配置 IBM MQ 傳訊提供者的集合 Factory](#)，或 [配置 IBM MQ 傳訊提供者的佇列集合 Factory](#)，或 [配置 IBM MQ 傳訊提供者的主題集合 Factory](#)。

如果要配置 JMS 佇列目的地，以使用 IBM MQ 傳訊提供者來進行點對點傳訊，請執行下列動作：

- 請利用管理主控台 IBM MQ 傳訊提供者佇列設定畫面來定義下列類型的內容：
 - 一般內容，包括管理及 IBM MQ 佇列內容。
 - 連線內容，指定如何連接至管理佇列的佇列管理程式。
 - 進階內容，用來控制對 IBM MQ 傳訊提供者目的地所建立之連線的行為。
 - 佇列目的地的任何自訂內容。

如需相關資訊，請參閱 [配置 IBM MQ 傳訊提供者的佇列](#)。

如果要使用 IBM MQ 傳訊提供者來建立或配置發佈/訂閱傳訊的 JMS 主題目的地，請執行下列動作：

- 使用 IBM MQ 傳訊提供者主題設定畫面來定義下列類型的內容：
 - 一般內容，包括管理及 IBM MQ 主題內容。
 - 進階內容，用來控制對 IBM MQ 傳訊提供者目的地所建立之連線的行為。
 - 佇列目的地的任何自訂內容。

如需相關資訊，請參閱 [配置 IBM MQ 傳訊提供者的主題](#)。

相關概念

[第 400 頁的『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相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。

相關工作

[第 36 頁的『配置二進位格式 CCDT』](#)

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 Multiplatforms 上，當建立佇列管理程式時，會自動建立包含預設值的二進位 CCDT。您可以使用 `runmqsc` 指令來更新二進位 CCDT。

[第 335 頁的『配置發佈/訂閱傳訊』](#)

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。

WebSphere Application Server 主題

[IBM MQ 傳訊提供者啟動規格](#)

[建立 IBM MQ 傳訊提供者的啟動規格](#)

[配置 IBM MQ 傳訊提供者的啟動規格](#)

[建立 IBM MQ 傳訊提供者的 Connection Factory](#)

[配置 IBM MQ 傳訊提供者的統一 Connection Factory](#)

[配置 IBM MQ 傳訊提供者的 Queue Connection Factory](#)

[配置 IBM MQ 傳訊提供者的 Topic Connection Factory](#)

[配置 IBM MQ 傳訊提供者的佇列](#)

[配置 IBM MQ 傳訊提供者的主題](#)

使用 wsadmin Scripting 指令來配置 JMS 資源

您可以使用 WebSphere Application Server wsadmin Scripting 指令來建立、修改、刪除或顯示 JMS 啟動規格、Connection Factory、佇列和主題的相關資訊。您也可以顯示和管理 IBM MQ 資源配接器的設定。

關於這項作業

下列步驟提供您可以使用 WebSphere Application Server `wsadmin` 指令來配置 JMS 資源以與 IBM MQ 傳訊提供者搭配使用之方式的概觀。如需如何使用這些指令的相關資訊，請參閱 相關鏈結，以取得 WebSphere Application Server 產品說明文件的鏈結。

如果要執行指令，請使用 `wsadmin` Scripting 用戶端的 `AdminTask` 物件。

使用指令來建立新物件或進行變更之後，請將變更儲存至主要配置。例如，使用下列指令：

```
AdminConfig.save()
```

如果要查看可用的 IBM MQ 傳訊提供者管理指令清單，以及每一個指令的簡要說明，請在 `wsadmin` 提示下輸入下列指令：

```
print AdminTask.help('WMQAdminCommands')
```

若要查看給定指令的概觀說明，請在 `wsadmin` 提示下輸入下列指令：

```
print AdminTask.help('command_name')
```

程序

如果要列出在發出指令的範圍中所定義的所有 IBM MQ 傳訊提供者資源，請使用下列指令。

- 如果要列出啟動規格，請使用 **listWMQActivationSpecs** 指令。
- 若要列出 Connection Factory，請使用 **listWMQConnectionFactory** 指令。
- 如果要列出佇列類型目的地，請使用 **listWMQQueues** 指令。
- 如果要列出主題類型目的地，請使用 **listWMQTopics** 指令。

如果要在特定範圍為 IBM MQ 傳訊提供者建立 JMS 資源，請使用下列指令。

- 如果要建立啟動規格，請使用 **createWMQActivationSpec** 指令。
您可以指定用來建立連線的所有參數來建立啟動規格，也可以建立啟動規格，以便它使用用戶端通道定義表 (CCDT) 來尋找要連接的佇列管理程式。
- 如果要建立 Connection Factory，請使用 **createWMQConnectionFactory** 指令，並使用 **-type** 參數來指定您要建立的 Connection Factory 類型：
 - 如果您的 JMS 應用程式預期只使用點對點傳訊，請為點對點傳訊網域建立網域特定的 Connection Factory，以用來建立專門用於點對點傳訊的連線。
 - 如果您的 JMS 應用程式只預期使用發佈/訂閱傳訊，請為發佈/訂閱傳訊網域建立網域特定的 Connection Factory，以用來建立專門用於發佈/訂閱傳訊的連線。
 - 若為 JMS 1.1 或更新版本，請建立可同時用於點對點傳訊和發佈/訂閱傳訊的獨立網域 Connection Factory，讓您的應用程式在相同交易下同時執行點對點和發佈/訂閱工作。

預設類型是與網域無關的 Connection Factory。

- 如果要建立佇列類型目的地，請使用 **createWMQQueue** 指令。
- 如果要建立主題類型目的地，請使用 **createWMQTopic** 指令。

如果要修改特定範圍之 IBM MQ 傳訊提供者的 JMS 資源，請使用下列指令。

- 如果要修改啟動規格，請使用 **modifyWMQActivationSpec** 指令。
您無法變更啟動規格的類型。例如，您無法建立啟動規格，在其中手動輸入所有配置資訊，然後修改它使用 CCDT。
- 如果要修改 Connection Factory，請使用 **modifyWMQConnectionFactory** 指令。
- 如果要修改佇列類型目的地，請使用 **modifyWMQQueue** 指令。
- 如果要修改主題類型目的地，請使用 **modifyWMQTopic** 指令。

如果要刪除特定範圍之 IBM MQ 傳訊提供者的 JMS 資源，請使用下列指令。

- 如果要刪除啟動規格，請使用 **deleteWMQActivationSpec** 指令。
- 若要刪除 Connection Factory，請使用 **deleteWMQConnectionFactory** 指令。
- 如果要刪除佇列類型目的地，請使用 **deleteWMQQueue** 指令。
- 如果要刪除主題類型目的地，請使用 **deleteWMQTopic** 指令。

如果要顯示特定 IBM MQ 傳訊提供者資源的相關資訊，請使用下列指令。

- 如果要顯示與特定啟動規格相關聯的所有參數及其值，請使用 **showWMQActivationSpec** 指令。
- 如果要顯示與特定 Connection Factory 相關聯的所有參數及其值，請使用 **showWMQConnectionFactory** 指令。
- 若要顯示與特定佇列類型目的地相關聯的所有參數及其值，請使用 **showWMQQueue** 指令。
- 若要顯示與主題類型目的地相關聯的所有參數及其值，請使用 **deleteWMQTopic** 指令。

如果要管理 IBM MQ 資源配接器或 IBM MQ 傳訊提供者的設定，請使用下列指令。

- 如果要管理安裝在特定範圍內之 IBM MQ 資源配接器的設定，請使用 **manageWMQ** 指令。
- 若要顯示 **manageWMQ** 指令可以設定的所有參數及其值，請使用 **showWMQ** 指令。這些設定與 IBM MQ 資源配接器或 IBM MQ 傳訊提供者相關。**showWMQ** 指令也會顯示 IBM MQ 資源配接器上所設定的任何自訂內容。

相關概念

第 400 頁的『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相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。

相關工作

第 36 頁的『配置二進位格式 CCDT』

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 Multiplatforms 上，當建立佇列管理程式時，會自動建立包含預設值的二進位 CCDT。您可以使用 **runmqsc** 指令來更新二進位 CCDT。

第 335 頁的『配置發佈/訂閱傳訊』

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。

WebSphere Application Server 主題

[**createWMQActivationSpec** 指令](#)

[**createWMQConnectionFactory** 指令](#)

[**createWMQQueue** 指令](#)

[**createWMQTopic** 指令](#)

[**deleteWMQActivationSpec** 指令](#)

[**deleteWMQConnectionFactory** 指令](#)

[**deleteWMQQueue** 指令](#)

[**deleteWMQTopic** 指令](#)

[**listWMQActivationSpecs** 指令](#)

[**listWMQConnectionFactories** 指令](#)

[**listWMQQueues** 指令](#)

[**listWMQTopics** 指令](#)

[**modifyWMQActivationSpec** 指令](#)

[**modifyWMQConnectionFactory** 指令](#)

[**modifyWMQQueue** 指令](#)

[**modifyWMQTopic** 指令](#)

[**showWMQActivationSpec** 指令](#)

[**showWMQConnectionFactory** 指令](#)

[showMQQueue 指令](#)

[showMQTopic 指令](#)

[showMQ 指令](#)

[manageMQ 指令](#)

使用 JMS 2.0 共用訂閱

在 WebSphere Application Server traditional 9.0 中，您可以搭配 IBM MQ 9.0 來配置及使用 JMS 2.0 共用訂閱。

關於這項作業

JMS 2.0 規格引進了共用訂閱的概念，可讓一或多個消費者開啟單一訂閱。這些訊息會在所有這些消費者之間共用。只要這些消費者連接至相同的佇列管理程式，就沒有任何限制。

「共用訂閱」可以是可延續或不可延續，其語意與現在稱為非共用訂閱的語意相同。

為了讓消費者能夠識別要使用的訂閱，它需要提供訂閱名稱。這類似於未共用的可延續訂閱，但在所有需要共用訂閱的情況下，都需要訂閱名稱。不過，如果是可延續共用訂閱，則不需要 `clientId`；可以提供一個，但並非必要。

雖然共用訂閱可以視為負載平衡機制，但無論是在 IBM MQ 或 JMS 2.0 規格中，都沒有任何關於如何在消費者之間配送訊息的承諾。

在 WebSphere Application Server traditional 9.0 中，預先安裝 IBM MQ 9.0 資源配接器。

下列步驟顯示如何利用 WebSphere Application Server traditional 管理主控台，將啟動規格配置成使用共用可延續或共用不可延續訂閱。

程序

首先在 JNDI 中建立物件。

1. 正常在 JNDI 中建立主題目的地 (請參閱 [第 546 頁的『使用管理主控台來配置 JMS 資源』](#))。
2. 建立啟動規格 (請參閱 [第 546 頁的『使用管理主控台來配置 JMS 資源』](#))。

您可以使用確切需要的內容來建立啟動規格。如果您想要使用可延續訂閱，您可以在建立時選取它並指定名稱。如果您想要使用不可延續訂閱，此時無法指定名稱。相反地，您需要為訂閱名稱建立自訂內容。

更新您以必要的自訂內容所建立的啟動規格。您可能需要指定兩個自訂內容：

- 無論如何，您都必須建立自訂內容，以指定這個啟動規格應該使用共用訂閱。
- 如果訂閱建立為不可延續，則需要將訂閱名稱內容設為自訂內容。

下表顯示您可以為每一個自訂內容指定的有效值：

內容名稱	類型	有效值
sharedSubscription	字串	true、false
subscriptionName	字串	非零長度 Java 字串

3. 從顯示在 **啟動規格集合** 表單中的清單中選取啟動規格。
啟動規格的詳細資料會顯示在 **IBM MQ 傳訊提供者啟動規格設定** 表單中。
4. 在 **IBM MQ 傳訊提供者啟動規格設定** 表單上，按一下 **自訂內容**。
即會顯示 **自訂內容** 表單。
5. 如果您使用不可延續訂閱，請建立 `subscriptionName` 自訂內容。
在啟動規格的「**自訂內容**」畫面上，按一下 **新建**，然後輸入下列詳細資料：

姓名

自訂內容的名稱，在本例中是 `subscriptionName`。

值

自訂內容的值。您可以在 **值** 欄位中使用 JNDI 名稱，例如 WASSharedSubOne。

類型

自訂內容的類型。從清單中選取自訂內容類型，在此情況下必須是 `java.lang.String`。

6. 對於共用可延續及共用不可延續訂閱，請建立 `sharedSubscription` 自訂內容。

在啟動規格的「**自訂內容**」畫面上，按一下 **新建**，然後輸入下列詳細資料：

姓名

自訂內容的名稱，在本例中是 `sharedSubscription`。

值

自訂內容的值。如果要指定啟動規格使用共用訂閱，請將值設為 `true`。如果您稍後想要停止使用此啟動規格的共用訂閱，您可以將此自訂內容的值設為 `false` 來執行此動作。

類型

自訂內容的類型。從清單中選取自訂內容類型，在此情況下必須是 `java.lang.String`。

7. 設定內容時，請重新啟動應用程式伺服器。

然後當訊息送達時，會驅動啟動規格的訊息驅動 Bean (MDB)，但只有 MDB 會共用所傳送的訊息。

相關概念

[複製及共用訂閱](#)

[訂閱延續性](#)

相關工作

[配置入埠通訊的資源配接器](#)

WebSphere Application Server traditional 9.0 的相關資訊

[配置 IBM MQ 傳訊提供者的主題](#)

[IBM MQ 傳訊提供者啟動規格](#)

[建立 IBM MQ 傳訊提供者的啟動規格](#)

[配置 IBM MQ 傳訊提供者的啟動規格](#)

[配置 IBM MQ 傳訊提供者 JMS 資源的自訂內容](#)

使用 JMS 2.0 ConnectionFactory 和「目的地查閱」內容

在 WebSphere Application Server traditional 9.0 中，啟動規格的 ConnectionFactory 查閱和 DestinationLookup 內容可以與受管理物件的 JNDI 名稱一起提供，以便優先於其他啟動規格內容使用。

關於這項作業

JMS 2.0 規格在用來驅動訊息驅動 Bean (MDB) 的啟動規格上指定兩個其他內容。先前，每一個供應商必須在啟動規格上指定自訂內容，以提供連接至傳訊系統及定義要從哪個目的地取得訊息所需的詳細資料。

現在標準 connectionFactory 查閱及 destinationLookup 內容可用來提供要查閱及使用之相關物件的 JNDI 名稱。在 WebSphere Application Server traditional 9.0 內，已預先安裝 IBM MQ 9.0 資源配接器。

下列步驟顯示如何使用 WebSphere Application Server traditional 管理主控台來自訂及使用這兩個內容。

程序

首先在 JNDI 中建立物件。

1. 如常在 JNDI 中建立 ConnectionFactory (請參閱 [第 546 頁的『使用管理主控台來配置 JMS 資源』](#))。
2. 正常在 JNDI 中建立目的地 (請參閱 [第 546 頁的『使用管理主控台來配置 JMS 資源』](#))。
Destination 物件必須具有正確的值。
3. 使用任何需要的值來建立啟動規格 (請參閱 [第 546 頁的『使用管理主控台來配置 JMS 資源』](#))。
您可以使用確切需要的內容來建立啟動規格。不過，您應該記住下列考量：

- 如果您想要 IBM MQ 資源配接器使用 Java EE Connection Factory 和目的地查閱內容，當您建立啟動規格時，所使用的內容會較不相關（請參閱 [ActivationSpec ConnectionFactory 查閱](#) 和 [DestinationLookup 內容](#)）。
- 不過，任何尚未定義在 Connection Factory 或目的地上的內容，仍必須在啟動規格中指定。因此，您必須定義連線消費者內容和其他內容，以及實際建立連線時所使用的鑑別資訊。
- 在 Connection Factory 上定義的內容中，ClientID 內容具有特殊處理。這是因為一般實務是使用具有多個啟動規格的單一 Connection Factory。這可簡化管理，不過 JMS 規格確實需要唯一的用戶端 ID，因此啟動規格必須能夠置換 ConnectionFactory 上所設定的任何值。如果啟動規格上未設定 ClientID，則會使用 Connection Factory 上的任何值。

請依照步驟 第 552 頁的『4』的說明，利用 WebSphere Application Server 管理主控台來更新您利用兩個新自訂內容所建立的啟動規格，或依照步驟 第 552 頁的『5』的說明來改用註釋。

4. 在 WebSphere Application Server 管理主控台中更新啟動規格。

這兩個內容需要在啟動規格的自訂內容畫面上設定。這些內容不會出現在主要啟動規格畫面或「啟動規格建立」精靈中。

- 從顯示在 **啟動規格集合** 表單中的清單中選取啟動規格。
啟動規格的詳細資料會顯示在 **IBM MQ 傳訊提供者啟動規格設定** 表單中。
- 在 **IBM MQ 傳訊提供者啟動規格設定** 表單上，按一下 **自訂內容**。
即會顯示 **自訂內容** 表單。
- 在 **自訂內容** 表單上，建立兩個新的自訂內容，兩者都是 java.lang.String 類型。
在每一種情況下，按一下 **新建**，然後輸入自訂內容的下列詳細資料：

姓名

自訂內容的名稱，可以是 connectionFactoryLookup 或 destinationLookup。

值

自訂內容的值。您可以在 **值** 欄位中使用 JNDI 名稱，例如 QuoteCF 和 QuoteQ。

類型

自訂內容的類型。從清單中選取自訂內容類型，在此情況下必須是 java.lang.String。

現在，已部署的 MDB 將使用這些值來建立 Connection Factory 和目的地。部署 MDB 時，不需要設定 JNDI 值配置。

5. 請使用註釋來取代啟動規格。

也可以在 MDB 程式碼中使用註釋來指定值。例如，使用 JNDI 名稱 QuoteCF 及 QuoteQ，程式碼看起來如下：

```
@MessageDriven(activationConfig = {
    @ActivationConfigProperty(propertyName = "destinationType" , propertyValue =
"javax.jms.Topic" ),
    @ActivationConfigProperty(propertyName = "destinationLookup" , propertyValue =
"QuoteQ" ),
    @ActivationConfigProperty(propertyName = "connectionFactoryLookup" , propertyValue
= "QuoteCF" )}, mappedName = "LookupMDB" )
@TransactionAttribute(TransactionAttributeType.REQUIRED)
@TransactionManagement(TransactionManagementType.CONTAINER)
publicclass LookupMDB implements MessageListener {
```

相關工作

[配置入埠通訊的資源配接器](#)

WebSphere Application Server traditional 9.0 的相關資訊

[配置 IBM MQ 傳訊提供者的統一 Connection Factory](#)

[配置 IBM MQ 傳訊提供者的主題](#)

[IBM MQ 傳訊提供者啟動規格](#)

[建立 IBM MQ 傳訊提供者的啟動規格](#)

[配置 IBM MQ 傳訊提供者的啟動規格](#)

[配置 IBM MQ 傳訊提供者 JMS 資源的自訂內容](#)

將應用程式伺服器配置成使用最新的資源配接器維護層次

當您套用 WebSphere Application Server 修正套件時，為了確保 IBM MQ 資源配接器會自動更新至最新可用的維護層次，您可以將環境中的所有伺服器配置成使用已套用至每個節點安裝的 WebSphere Application Server 修正套件所包含的資源配接器最新版本。

開始之前

重要: 如果您在任何平台上使用 WebSphere Application Server 8.5 或更早版本，請勿將 IBM MQ 8.0 或更新版本的資源配接器安裝到應用程式伺服器中。IBM MQ 8.0 或更新版本的資源配接器只能部署至支援 JMS 2.0 的應用程式伺服器。不過，WebSphere Application Server 8.5 或更早版本只支援 JMS 1.1。這些 WebSphere Application Server 版本隨附 IBM WebSphere MQ 7.0 資源配接器，可用來使用 BINDINGS 或 CLIENT 傳輸連接至 IBM MQ 8.0 佇列管理程式。

關於這項作業

如果下列任何情況適用於您的配置，且您想要將環境中的所有伺服器配置成使用最新版本的 IBM MQ 資源配接器，請使用這項作業：

- 在套用 WebSphere Application Server 7.0 Fix Pack 1 或更新版本之後，您環境中任何應用程式伺服器的 JVM 日誌都會顯示下列 IBM MQ 資源配接器版本資訊：

```
WMSG1703I:RAR 實作版本 7.0.0.0-k700-L080820
```

- 您環境中任何應用程式伺服器的 JVM 日誌包含下列項目：

```
WMSG1625E: 無法偵測  
the IBM MQ messaging provider code at the specified path <null>
```

- 先前已手動更新一或多個節點來使用 IBM MQ 資源配接器的特定維護層次，現在已被現行 WebSphere Application Server 維護層次所包含的資源配接器最新版本所取代。

範例所參照的 *profile_root* 目錄是 WebSphere Application Server 設定檔的起始目錄，例如 C:\Program Files\IBM\WebSphere\AppServer1。

當您對環境中的所有 Cell 和單一伺服器安裝執行下列步驟時，當套用新的 WebSphere Application Server 修正套件時，您的伺服器會自動收到 IBM MQ 資源配接器的維護。

程序

- 啟動應用程式伺服器。如果設定檔是網路部署配置的一部分，請啟動部署管理程式及所有節點代理程式。如果設定檔包含管理代理程式，請啟動管理代理程式。
- 檢查 IBM MQ 資源配接器的維護層次。
 - 開啟命令提示字元視窗，並切換至 *profile_root*\bin 目錄。
例如，輸入 `cd C:\Program Files\IBM\WebSphere\AppServer1\bin`。
 - 輸入 `wsadmin.bat -lang jython` 來啟動 wsadmin 工具，如果提示您這樣做，請輸入您的使用者名稱和密碼。
 - 鍵入下列指令，然後按「返回」兩次：

```
wmqInfoMBeansUnsplit = AdminControl.queryNames("WebSphere:type=WMQInfo,*")  
wmqInfoMBeansSplit = AdminUtilities.convertToList(wmqInfoMBeansUnsplit)  
for wmqInfoMBean in wmqInfoMBeansSplit: print wmqInfoMBean; print AdminControl.invoke(wmqInfoMBean,  
'getInfo', '')
```

您也可以 Jacl 中執行此指令。如需如何執行此動作的進一步相關資訊，請參閱 WebSphere Application Server 產品說明文件中的確保伺服器使用最新可用的 *IBM MQ* 資源配接器維護層次。

- 在指令顯示的輸出中尋找 WMSG1703I 訊息，並檢查資源配接器層次。

例如，對於 WebSphere Application Server 7.0.1 Fix Pack 5，訊息應該為：

```
WMSG1703I: RAR 實作版本 7.0.1.3-k701-103-100812
```

此訊息顯示版本是 7.0.1.3-k701-103-100812，這是此修正套件的正確資源配接器層次。不過，如果改為顯示下列訊息，則表示您需要將資源配接器調整為 Fix Pack 15 的正確維護層次。

- 將下列 Jython Script 複製到稱為 `convertWMQRA.py` 的檔案中，然後將它儲存至設定檔根目錄，例如 `C:\Program Files\IBM\WebSphere\AppServer1\bin`。

```

ras = AdminUtilities.convertToList(AdminConfig.list('J2CResourceAdapter'))

for ra in ras :
    desc = AdminConfig.showAttribute(ra, "description")
    if (desc == "WAS 7.0 Built In IBM MQ Resource Adapter") or (desc == "WAS 7.0.0.1 Built In IBM MQ
Resource Adapter"):
        print "Updating archivePath and classpath of " + ra
        AdminConfig.modify(ra, [['archivePath', "${WAS_INSTALL_ROOT}/installedConnectors/wmq.jmsra.rar"]])
        AdminConfig.unsetAttributes(ra, ['classpath'])
        AdminConfig.modify(ra, [['classpath', "${WAS_INSTALL_ROOT}/installedConnectors/wmq.jmsra.rar"]])
        AdminConfig.save()
    #end if
#end for

```

提示: 儲存檔案時，請確定它儲存為 Python 檔案，而不是文字檔。

- 使用 WebSphere Application Server `wsadmin` 工具來執行您剛建立的 Jython Script。

開啟命令提示字元，並導覽至 WebSphere Application Server 起始目錄中的 `\bin` 目錄 (例如 `C:\Program Files\IBM\WebSphere\AppServer1\bin` 目錄)，然後鍵入下列指令並按 Return:

```
wsadmin -lang jython -f convertWMQRA.py
```

如果提示您這麼做，請輸入您的使用者名稱和密碼。

註: 如果您對屬於網路部署配置的設定檔執行 Script，該 Script 會更新該配置中所有需要更新的設定檔。如果您有預先存在的配置檔不一致，則可能需要完整重新同步。

- 如果您在網路部署配置中執行，請確定節點代理程式已完全重新同步化。如需相關資訊，請參閱使用 `wsadmin Scripting` 工具同步化節點或新增、管理及移除節點。
- 停止設定檔中的所有伺服器。如果設定檔是網路部署配置的一部分，也請停止配置中的任何叢集成員，停止配置中的所有節點代理程式，然後停止部署管理程式。如果設定檔包含管理代理程式，請停止管理代理程式。
- 從 `profile_root/bin` 目錄執行 `osgiCfgInit` 指令。
`osgiCfgInit` 指令會重設 OSGi 執行時期環境所使用的類別快取。如果設定檔是網路部署配置的一部分，請從配置中每個設定檔的 `profile_root/bin` 目錄執行 `osgiCfgInit` 指令。
- 重新啟動設定檔中的所有伺服器。如果設定檔是網路部署配置的一部分，也請重新啟動配置中的任何叢集成員，重新啟動配置中的所有節點代理程式，然後重新啟動部署管理程式。如果設定檔包含管理代理程式，請重新啟動管理代理程式。
- 重複步驟 2，以檢查資源配接器現在是否處於正確的層次。

下一步

在執行本主題中說明的步驟之後，如果您繼續遇到問題，且先前已使用 WebSphere Application Server 管理主控台中「JMS 提供者設定」畫面上的 **更新資源配接器** 按鈕，來更新環境中任何節點上的 IBM MQ 資源配接器，則您可能遇到 [APAR PM10308](#) 中說明的問題。

相關工作

[使用 IBM MQ 資源配接器](#)

WebSphere Application Server 8.5.5 的相關資訊

[確定伺服器使用最新可用的 IBM MQ 資源配接器維護層次](#)

[使用 wsadmin Scripting 工具同步化節點](#)

[新增、管理及移除節點](#)

[JMS 提供者設定值](#)

配置 JMS PROVIDERVERSION 內容

IBM MQ 傳訊提供者具有三種作業模式：標準模式、有限制的標準模式及移轉模式。您可以設定 JMS **PROVIDERVERSION** 內容，以選取 JMS 應用程式使用哪些模式來發佈及訂閱。

關於這項作業

主要可以透過設定 **PROVIDERVERSION** Connection Factory 內容來控制 IBM MQ 傳訊提供者作業模式的選擇。如果尚未指定模式，也可以自動選取作業模式。

PROVIDERVERSION 內容會區分三個 IBM MQ 傳訊提供者作業模式：

IBM MQ 傳訊提供者標準模式

一般模式使用 IBM MQ 佇列管理程式的所有特性來實作 JMS。此模式已最佳化為使用 JMS 2.0 API 和功能。

有限制的 IBM MQ 傳訊提供者標準模式

有限制的標準模式使用 JMS 2.0 API，但不使用新特性，即共用訂閱、延遲遞送及非同步傳送。

IBM MQ 傳訊提供者移轉模式

在移轉模式下，您可以連接至 IBM MQ 8.0 或更新版本的佇列管理程式，但不會使用 IBM WebSphere MQ 7.0 或更新版本佇列管理程式的任何特性 (例如先讀及串流)。

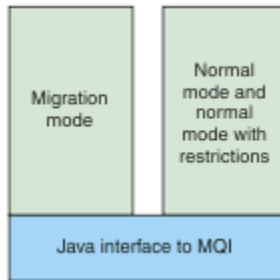


圖 91: 傳訊提供者模式

程序

如果要配置特定 Connection Factory 的 **PROVIDERVERSION** 內容，請執行下列動作：

- 如果要使用 IBM MQ Explorer 來配置 **PROVIDERVERSION** 內容，請參閱 [配置佇列管理程式和物件](#)。
- 若要使用 JMS 管理工具來配置 **PROVIDERVERSION** 內容，請參閱 [配置佇列管理程式及物件](#)。
- 如果要在 JMS 應用程式中使用 IBM JMS 延伸或 IBM MQ JMS 延伸來配置 **PROVIDERVERSION** 內容，請參閱在 [IBM MQ classes for JMS 應用程式中建立及配置 Connection Factory](#) 和目的地。

如果要置換 JVM 中所有 Connection Factory 的 Connection Factory 提供者模式設定，請執行下列動作：

- 如果要置換 Connection Factory 提供者模式設定，請使用 `com.ibm.msg.client.wmq.overrideProviderVersion` 內容
如果無法變更正在使用的 Connection Factory，則可以使用 `com.ibm.msg.client.wmq.overrideProviderVersion` 內容來置換 Connection Factory 上的任何設定。此置換會套用至 JVM 中的所有 Connection Factory，但不會修改實際的 Connection Factory 物件。

相關概念

[Connection Factory 內容](#)

相關工作

[JMS 提供者版本疑難排解](#)

相關參考

[PROVIDERVERSION](#)

[IBM MQ classes for JMS 物件的內容之間的相依關係](#)

IBM MQ 傳訊提供者作業模式

您可以將 Connection Factory 的 PROVIDERVERSION 內容設為適當的值，以選取 JMS 應用程式用來發佈和訂閱的 IBM MQ 傳訊提供者作業模式。在某些情況下，PROVIDERVERSION 內容會設為未指定，在這種情況下，JMS 用戶端會使用演算法來決定要使用的作業模式。

PROVIDERVERSION 內容值

您可以將 Connection Factory **PROVIDERVERSION** 內容設為下列任何值：

8 - 標準模式

JMS 應用程式使用標準模式。此模式會使用 IBM MQ 佇列管理程式的所有特性來實作 JMS。

7 - 有限制的標準模式

JMS 應用程式使用有限制的標準模式。此模式使用 JMS 2.0 API，但不使用新特性，例如共用訂閱、延遲傳遞或非同步傳送。

6 - 移轉模式

JMS 應用程式使用移轉模式。在移轉模式中，IBM MQ classes for JMS 會使用與 IBM WebSphere MQ 6.0 隨附的特性和演算法類似的特性和演算法。

unspecified (預設值)

JMS 用戶端會使用演算法來決定使用哪一種作業模式。

為 **PROVIDERVERSION** 內容指定的值必須是字串。若要指定選項 8、7 或 6，則可以採用下列任一格式來執行此動作：

- V.R.M.F
- V.R.M
- V.R
- V

其中 V、R、M 和 F 是大於或等於零的整數值。額外的 R、M 和 F 值是選用項目，可供您在需要微調控制時使用。例如，如果您想要使用 **PROVIDERVERSION** 層次為 7，則可以設定 **PROVIDERVERSION = 7、7.0、7.0.0 或 7.0.0.0**。

Connection Factory 物件的類型

您可以為下列類型的 Connection Factory 物件設定 **PROVIDERVERSION** 內容：

- MQConnectionFactory
- MQQueueConnectionFactory
- MQTopicConnectionFactory
- MQXAConnectionFactory
- MQXAQueueConnectionFactory
- MQXAQueueConnectionFactory
- MQXAQueueConnectionFactory
- MQXATopicConnectionFactory

如需這些不同 Connection Factory 類型的相關資訊，請參閱 [第 538 頁的『使用管理工具來配置 JMS 物件』](#)。

相關概念

[IBM MQ 傳訊提供者](#)

PROVIDERVERSION 標準模式

一般模式使用 IBM MQ 佇列管理程式的所有特性來實作 JMS。此模式已最佳化為使用 JMS 2.0 API 和功能。

下列流程圖顯示 JMS 用戶端進行的檢查，以判斷是否可以建立標準模式連線。

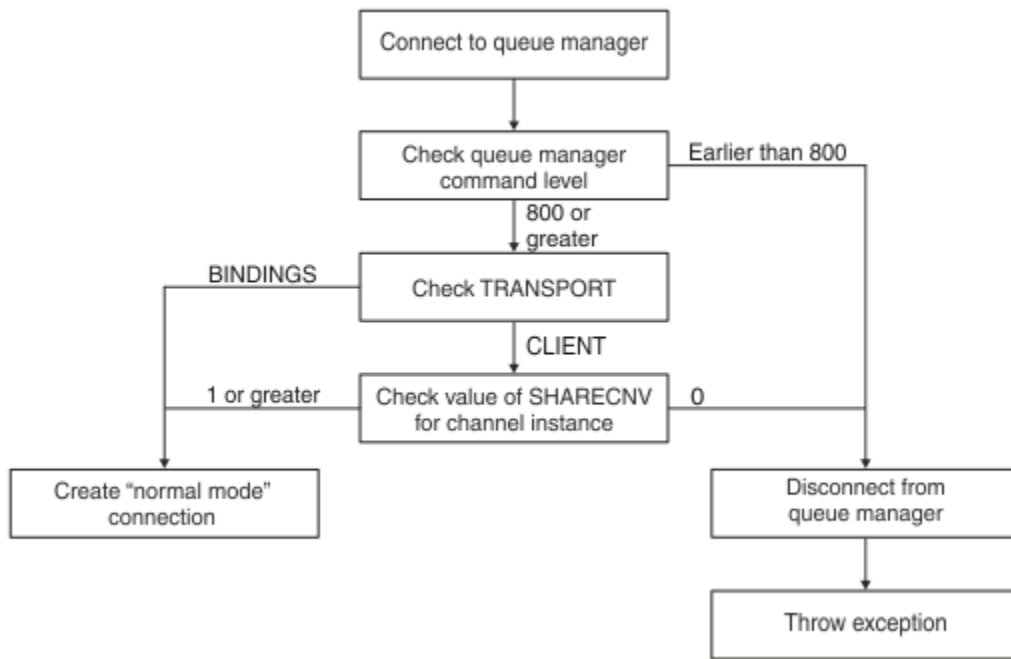


圖 92: *PROVIDERVERSION* 標準模式

如果 Connection Factory 設定中指定的佇列管理程式的指令層次是 800 以上，且 Connection Factory 的 **TRANSPORT** 內容設定為 BINDINGS，則會建立標準模式連線，而不檢查任何進一步內容。

如果 Connection Factory 設定中指定的佇列管理程式的指令層次是 800 以上，且 **TRANSPORT** 內容設定為 CLIENT，則亦會檢查伺服器連線通道上的 **SHARECNV** 內容。需要此檢查，因為 IBM MQ 傳訊提供者標準模式使用共用交談特性。因此，為順利嘗試標準模式連線，用於控制可共用交談數的 **SHARECNV** 內容的值必須是 1 以上。

如果流程圖中顯示的所有檢查都成功，則會建立與佇列管理程式的標準模式連線，然後可以使用所有 JMS 2.0 API 和特性 (即非同步傳送、延遲遞送及共用訂閱)。

由於下列一種原因，嘗試建立標準模式連線失敗：

- Connection Factory 設定中指定的佇列管理程式的指令層次低於 800。在此情況下，`createConnection` 方法會失敗，並出現異常狀況 JMSFMQ0003。
- 伺服器連線通道上的 **SHARECNV** 內容設為 0。如果此內容沒有 1 或更大的值，則 `createConnection` 方法會失敗，並發生異常狀況 JM5CC5007。

相關資訊

[IBM MQ classes for JMS 物件的內容之間的相依關係](#)

[DEFINE CHANNEL \(SHARECNV 內容\)](#)

[TRANSPORT](#)

PROVIDERVERSION 有限制的標準模式

有限制的標準模式會使用 JMS 2.0 API，但不會使用新的 IBM MQ 8.0 或更新版本特性，例如共用訂閱、延遲遞送或非同步傳送。

下列流程圖顯示 JMS 用戶端進行的檢查，以判定是否可以建立具有限制連線的標準模式。

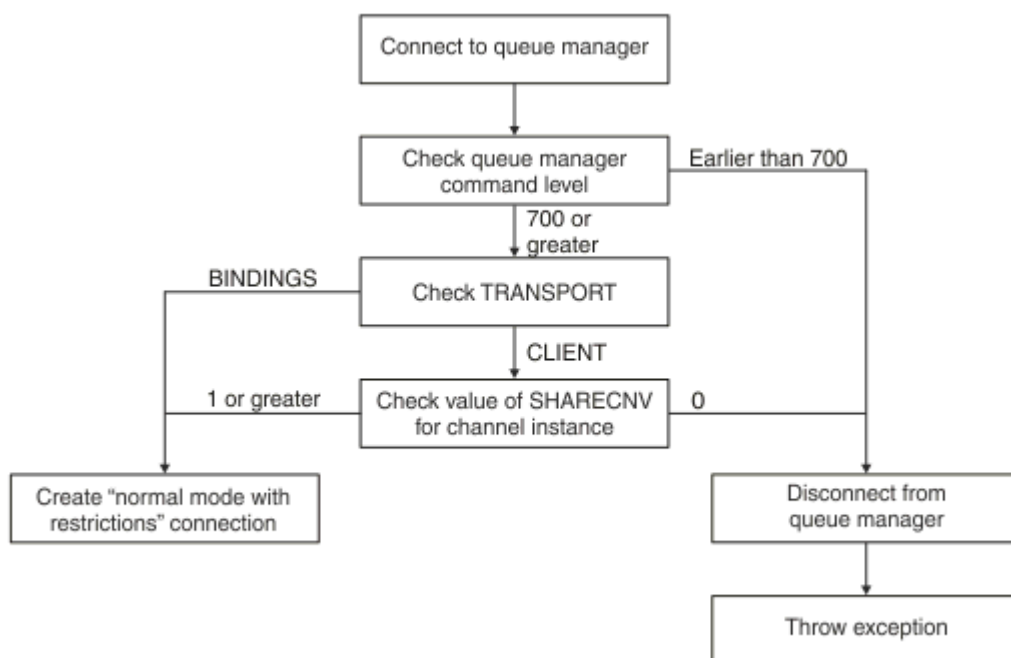


圖 93: PROVIDERVERSION 標準模式有限制

如果在 Connection Factory 設定中指定的佇列管理程式具有指令層次 700 或以上，且 Connection Factory 的 **TRANSPORT** 內容設為 BINDINGS，則會建立標準模式連線，而不會檢查任何進一步的內容。

如果在 Connection Factory 設定中指定的佇列管理程式具有指令層次 700 或以上，且 **TRANSPORT** 內容設為 CLIENT，則也會檢查伺服器連線通道上的 **SHARECNV** 內容。需要這項檢查，因為 IBM MQ 傳訊提供者標準模式 (有限制) 使用共用交談特性。因此，若要讓具有限制連線的標準模式嘗試成功，**SHARECNV** 內容 (用於控制可以共用的交談數) 必須具有 1 或更大的值。

如果流程圖中顯示的所有檢查都成功，則會建立與佇列管理程式具有限制連線的標準模式，然後您可以使用 JMS 2.0 API，但不能使用非同步傳送、延遲遞送或共用訂閱特性。

嘗試建立具有限制連線的標準模式失敗，原因如下：

- Connection Factory 設定中指定的佇列管理程式具有早於 700 的指令層次。在此情況下，createConnection 方法失敗，異常狀況為 JMSFCC5008。
- 伺服器連線通道上的 **SHARECNV** 內容設為 0。如果此內容沒有 1 或更大的值，則 createConnection 方法會失敗，並發生異常狀況 JMSSC5007。

相關資訊

[IBM MQ classes for JMS 物件的內容之間的相依關係](#)

[DEFINE CHANNEL \(SHARECNV 內容\)](#)

[TRANSPORT](#)

PROVIDERVERSION 移轉模式

對於移轉模式，IBM MQ classes for JMS 會使用類似於 IBM WebSphere MQ 6.0 隨附的特性和演算法，例如排入佇列的發佈/訂閱、在用戶端實作的選項、非多工通道，以及用來實作接聽器的輪詢。

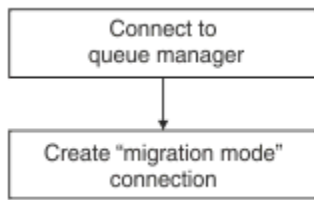


圖 94: PROVIDERVERSION 移轉模式

如果您想要使用 IBM MQ Enterprise Transport 6.0 連接至 WebSphere Message Broker 6.0 或 6.1，則必須使用移轉模式。

您可以使用移轉模式連接至 IBM MQ 8.0 佇列管理程式，但不會使用 IBM MQ classes for JMS 佇列管理程式的任何新增特性，例如先讀或串流。如果您有 IBM MQ 8.0 或更新版本的用戶端連接至分散式平台上的 IBM MQ 8.0 或更新版本的佇列管理程式，[z/OS](#) 或 z/OS 上的 IBM MQ 8.0 或更新版本的佇列管理程式，則訊息選擇是由佇列管理程式而非用戶端系統完成。

如果指定 IBM MQ 傳訊提供者移轉模式，且 IBM MQ classes for JMS 嘗試使用任何 JMS 2.0 API，API 方法呼叫會失敗，異常狀況為 JMSCC5007。

相關資訊

[IBM MQ classes for JMS 物件的內容之間的相依關係](#)
[TRANSPORT](#)

PROVIDERVERSION 未指定

當未指定 Connection Factory 的 **PROVIDERVERSION** 內容時，JMS 用戶端會使用演算法來決定要使用哪一種作業模式來連接佇列管理程式。當 Connection Factory 與新版 IBM MQ classes for JMS 搭配使用時，在 JNDI 名稱空間中使用舊版 IBM MQ classes for JMS 所建立的 Connection Factory 會採用未指定的值。

如果未指定 **PROVIDERVERSION** 內容，則會在呼叫 `createConnection` 方法時使用演算法。演算法會檢查一些 Connection Factory 內容，以判斷是否需要 IBM MQ 傳訊提供者標準模式、有限制的標準模式或 IBM MQ 傳訊提供者移轉模式。一律先嘗試標準模式，然後再嘗試有限制的標準模式。如果這兩種連線都無法建立，JMS 用戶端會切斷與佇列管理程式的連線，然後重新連接佇列管理程式，以嘗試移轉模式連線。

檢查 BROKERVER、BROKERQMGR、PSMODE 及 BROKERCONQ 內容

內容值的檢查開始於 **BROKERVER** 內容，如 [圖 1](#) 所示。

如果 **BROKERVER** 內容設為 V1，則接下來會勾選 **TRANSPORT** 內容，如 [圖 2](#) 所示。不過，如果 **BROKERVER** 內容設為 V2，則在檢查 **TRANSPORT** 內容之前，會先完成 [圖 1](#) 中顯示的其他檢查。

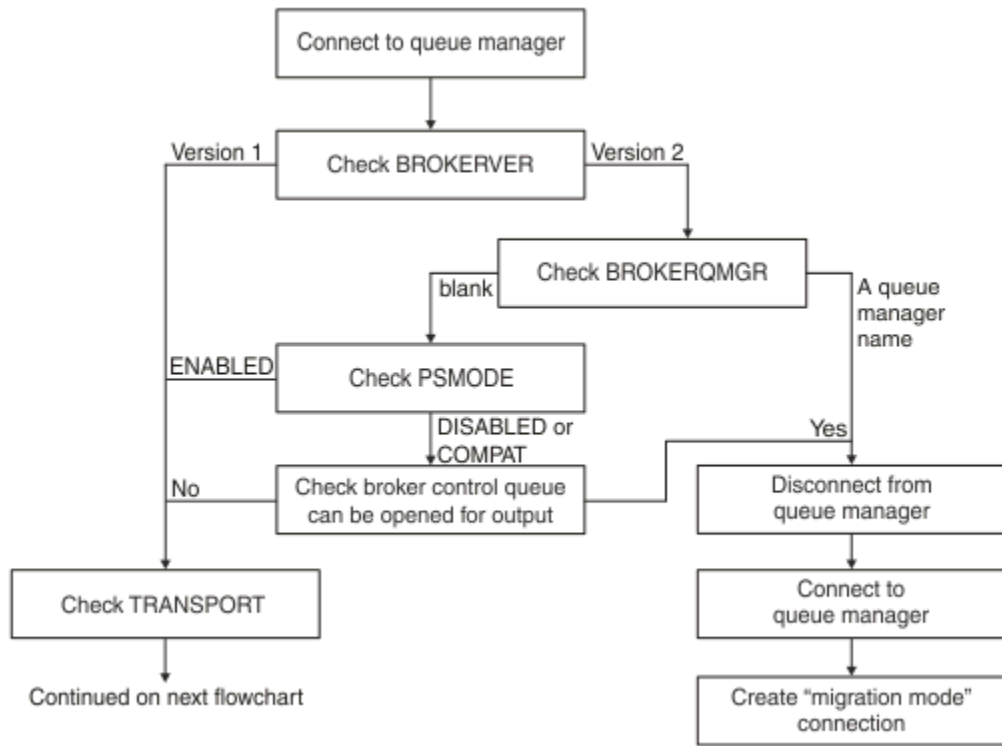


圖 95: PROVIDERVERSION 未指定

如果 **BROKERVER** 內容設為 V2，則為了能夠進行標準模式連線，**BROKERQMGR** 內容必須為空白。此外，佇列管理程式上的 **PSMODE** 屬性必須設為 **ENABLED**，否則 **BROKERCONQ** 內容指定的分配管理系統控制佇列無法開啟以供輸出。

如果根據標準模式連線的需要設定內容值，則檢查下一步會移至 **TRANSPORT** 內容，如圖 2 所示。

如果未將內容值設為標準模式連線所需的值，則 JMS 用戶端會中斷與佇列管理程式的連線，然後重新連接並建立移轉模式連線。在下列情況下會發生這種情況：

- 如果 **BROKERQMGR** 內容為空白，且佇列管理程式上的 **PSMODE** 屬性設為 **COMPAT** 或 **DISABLED**，則可以開啟 **BROKERCONQ** 內容所指定的分配管理系統控制佇列以進行輸出 (亦即，**MQOPEN** 表示輸出成功)。
- 如果 **BROKERQMGR** 內容指定佇列名稱。

檢查 TRANSPORT 內容及指令層次

圖 2 顯示針對佇列管理程式的 **TRANSPORT** 內容及指令層次所進行的檢查。

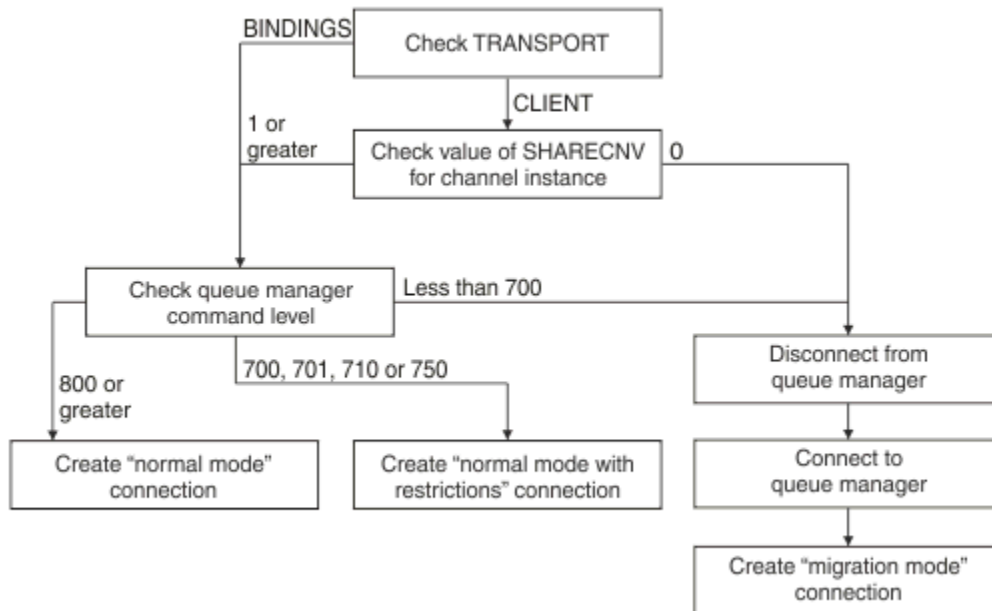


圖 96: PROVIDERVERSION 未指定 (續)

在下列任一情況下，會建立標準模式連線：

- Connection Factory 的 **TRANSPORT** 內容設為 BINDINGS，且佇列管理程式的指令層次為 800 或以上。
- **TRANSPORT** 內容設為 CLIENT，伺服器連線通道上的 **SHARECNV** 內容值為 1 或以上，且佇列管理程式的指令層次為 800 或以上。

如果佇列管理程式的指令層次為 710 或 750，則會建立與佇列管理程式具有限制連線的標準模式。

如果 **TRANSPORT** 內容設為 CLIENT，且伺服器連線通道上的 **SHARECNV** 內容值為 0，則也會建立移轉模式連線。

相關資訊

[IBM MQ classes for JMS 物件的內容之間的相依關係](#)

[ALTER QMGR \(PSMODE 屬性\)](#)

[BROKERCONQ](#)

[BROKERQMGR](#)

[BROKERVER](#)

[DEFINE CHANNEL \(SHARECNV 內容\)](#)

[TRANSPORT](#)

何時置換 PROVIDERVERSION 預設值

如果使用舊版 IBM MQ classes for JMS 在 JNDI 名稱空間中建立的 Connection Factory 與新版 IBM MQ classes for JMS 搭配使用，則 Connection Factory 的 **PROVIDERVERSION** 內容會設為預設值 `unspecified`，並使用演算法來決定使用作業的 IBM MQ 傳訊提供者模式。不過，在兩種情況下，您必須置換 **PROVIDERVERSION** 內容的預設選項，IBM MQ classes for JMS 才能正確運作。

註：本主題中說明的移轉模式適用於從 IBM WebSphere MQ 6.0 移轉至 IBM WebSphere MQ 7.0。它不適用於從更新版本移轉。

IBM WebSphere MQ 6.0、WebSphere Application Server 6.0.x 及 WebSphere Message Broker 6 不支援，因此本主題僅作為參考之用。

當 **PROVIDERVERSION** 內容設為預設值 `unspecified` 時，會使用演算法來決定要使用的作業模式，如第 559 頁的『[PROVIDERVERSION 未指定](#)』中所述。不過，在下列兩種情況下，您無法使用此演算法。

1. 如果 WebSphere Message Broker 和 WebSphere Event Broker 處於相容模式，您必須指定 **PROVIDERVERSION** 內容的值， WebSphere Message Broker 和 WebSphere Event Broker 才能正確運作。
2. 如果您使用 WebSphere Application Server 6.0.1、 WebSphere Application Server 6.0.2 或 WebSphere Application Server 6.1， Connection Factory 是利用 WebSphere Application Server 管理主控台來定義。

在 WebSphere Application Server 中， Connection Factory 上 **BROKERVER** 內容的預設值是 V2。對於使用 JMS 管理工具 **JMSAdmin** 或 IBM MQ Explorer 所建立的 Connection Factory， **BROKERVER** 內容的預設值是 V1。此內容現在在 IBM MQ 中 未指定。

如果 **BROKERVER** 內容設為 V2，因為它是由 WebSphere Application Server 所建立，或 Connection Factory 之前已用於發佈/訂閱，且具有已定義 **BROKERCONQ** 內容 (因為之前已用於發佈/訂閱傳訊) 的現有佇列管理程式，則會使用 IBM MQ 傳訊提供者移轉模式。

不過，如果您想要應用程式使用對等式通訊，且應用程式正在使用曾經用於發佈/訂閱的現有佇列管理程式，且 Connection Factory **BROKERVER** 設為 2(如果在 WebSphere Application Server 中建立 Connection Factory，則此為預設值)，則會使用 IBM MQ 傳訊提供者移轉模式。在此情況下，不需要使用 IBM MQ 傳訊提供者移轉模式; 請改用 IBM MQ 傳訊提供者標準模式。您可以使用下列其中一種方法來解決此問題:

- 將 **BROKERVER** 設為 1 或未指定。您選擇的選項視您的應用程式而定。
- 將 **PROVIDERVERSION** 設為 8 或 7，這是 WebSphere Application Server 6.1 中的自訂內容。

或者，使用用戶端配置內容，或修改連接的佇列管理程式，使其未設定 **BROKERCONQ** 內容，或使佇列無法使用。

在 WebSphere Application Server 中配置提供者版本資訊

如果要在 WebSphere Application Server 中配置提供者版本資訊，您可以使用管理主控台或 wsadmin 指令。

程序

如果要在 WebSphere Application Server 中配置 IBM MQ Connection Factory 或啟動規格物件的提供者版本資訊，請參閱 相關資訊，以取得 WebSphere Application Server 產品說明文件中進一步資訊的鏈結。

WebSphere Application Server 8.5.5 的相關資訊

[IBM MQ 傳訊提供者 Connection Factory 設定](#)

[createWMQConnectionFactory 指令](#)

[IBM MQ 傳訊提供者啟動規格設定](#)

[createWMQActivationSpec 指令](#)

WebSphere Application Server 8.0.0 的相關資訊

[IBM MQ 傳訊提供者 Connection Factory 設定](#)

[createWMQConnectionFactory 指令](#)

[IBM MQ 啟動規格設定](#)

[createWMQActivationSpec 指令](#)

WebSphere Application Server 7.0.0 的相關資訊

[IBM MQ 傳訊提供者 Connection Factory 設定](#)

[createWMQConnectionFactory 指令](#)

[IBM MQ 啟動規格設定](#)

[createWMQActivationSpec 指令](#)

移除 WebSphere Application Server 可延續訂閱

將 IBM MQ 傳訊提供者與 WebSphere Application Server 7.0 及 IBM MQ 8.0 搭配使用時，不會移除由連結至啟動規格的訊息驅動 Bean 應用程式建立的可延續訂閱。您可以使用 IBM MQ Explorer 或 IBM MQ 指令行公用程式來移除可延續訂閱。

關於此作業

移除可延續訂閱的訊息驅動 Bean 應用程式可以配置成使用接聽器埠或啟動規格，前提是應用程式是在使用 [IBM MQ 傳訊提供者標準模式](#) 連接至 IBM MQ 的 WebSphere Application Server 7.0 或 IBM MQ 8.0 實例內執行。

如果訊息驅動 Bean 應用程式連結至接聽器埠，則在第一次啟動應用程式時，IBM MQ 傳訊提供者會為應用程式建立可延續訂閱。從應用程式伺服器解除安裝訊息驅動 Bean 應用程式並重新啟動應用程式伺服器時，將會移除可延續訂閱。

連結至啟動規格的訊息驅動 Bean 應用程式會以略微不同的方式運作。第一次啟動應用程式時，將會為應用程式建立可延續訂閱。但是，在解除安裝應用程式並重新啟動應用程式伺服器時，將不會移除可延續訂閱。

這可能會導致不再安裝在 WebSphere Application Server 系統中的應用程式在「IBM MQ 發佈/訂閱」引擎上剩餘一些可延續訂閱。這些訂閱稱為「孤立訂閱」，在執行「發佈/訂閱」引擎時可能會導致佇列管理程式中發生問題。

在主題中發佈訊息時，「IBM MQ 發佈/訂閱」引擎會為在該主題中登錄的每個可延續訂閱建立該訊息的副本，並將其放置在內部佇列中。使用該可延續訂閱的應用程式，隨後會從此內部佇列中挑選並消費訊息。

如果不再安裝使用該可延續訂閱的訊息驅動 Bean 應用程式，則會繼續為應用程式建立已發佈訊息的副本。但是，絕不會處理這些訊息，這表示可能有大量訊息保留在內部佇列中，絕不會被移除。

開始之前

向「IBM MQ 發佈/訂閱」引擎登錄的訂閱，將具有與其相關聯的「訂閱名稱」。

WebSphere Application Server IBM MQ 傳訊提供者為連結至啟動規格的訊息驅動 Bean 建立的可延續訂閱，將具有下列格式的「訂閱名稱」：

```
JMS:queue manager name:client identifier:subscription name
```

其中：

queue manager name

這是執行「發佈/訂閱」引擎的 IBM MQ 佇列管理程式的名稱。

client identifier

這是訊息驅動 Bean 所連結的啟動規格的「用戶端 ID」內容值。

subscription name

這是訊息驅動 Bean 應用程式已配置為要使用的啟動規格之啟動規格「訂閱名稱」內容值。

例如，假設我們擁有的啟動規格已設定為連接至佇列管理程式 testQM。該啟動規格已設定下列內容：

- 用戶端 ID = testClientID
- 訂閱名稱 = durableSubscription1

如果取出可延續訂閱的訊息驅動 Bean 連結至此啟動規格，則會在具有下列「訂閱名稱」的佇列管理程式 testQM 上的 IBM MQ 發佈/訂閱引擎中建立訂閱：

- JMS:testQM:testClientID:durableSubscription1

使用下列兩種方法其中之一，即可檢視已向給定佇列管理程式的 IBM MQ 發佈/訂閱引擎登錄的訂閱：

- 第一個選項是使用「MQ 探險家」。當「MQ 探險家」已連接至用於發佈/訂閱工作的佇列管理程式時，按一下導覽窗格中的 IBM WebSphere MQ ->queue manager name-> Subscriptions 項目，即可檢視目前向發佈/訂閱引擎登錄的訂閱者清單。
- 檢視已向發佈/訂閱引擎登錄之訂閱的另一種方法，是使用 IBM MQ 指令行公用程式 **runmqsc**，並執行 **display sub** 指令。若要執行此作業，請啟動命令提示字元，切換至 *WebSphere MQ\bin* 目錄，然後輸入下列指令以啟動 **runmqsc**：

```
- runmqsc queue manager name
```

在 **runmqsc** 公用程式已啟動時輸入下列指令，以列出目前已向在 **runmqsc** 所連接佇列管理程式中執行的發佈/訂閱引擎登錄的所有可延續訂閱：

- display sub(*) durable

若要檢查已向發佈/訂閱引擎登錄的可延續訂閱是否仍在作用中：

1. 產生已向「發佈/訂閱」引擎登錄的可延續訂閱清單。
2. 對於每個可延續訂閱：
 - 檢查可延續訂閱者的訂閱名稱，並記下 *client identifier* 及 *subscription name* 值。
 - 檢查正在連接至此「發佈/訂閱」引擎的 WebSphere Application Server 系統。查看是否有「用戶端 ID」內容符合 *client identifier* 值，且訂閱名稱內容符合 *subscription name* 的任何已定義啟動規格。
 - 如果找不到用戶端 ID 和訂閱名稱內容符合 IBM MQ 訂閱名稱中用戶端 ID 和訂閱名稱欄位的啟動規格，則沒有啟動規格使用這個可延續訂閱。因此可以刪除此可延續訂閱。
 - 如果有已定義的啟動規格符合可延續訂閱名稱，則需要進行的最後檢查是查看是否有訊息驅動 Bean 應用程式正在使用此啟動規格。若要執行此作業：
 - 記下已取出您目前所檢查可延續訂閱的啟動規格之 JNDI 名稱。
 - 在 WebSphere Application Server 管理主控台中，為已安裝的每個訊息驅動 Bean 應用程式啟動「配置」窗格。
 - 在「配置」窗格中按一下「訊息驅動 Bean 接聽器連結」鏈結。
 - 即會顯示包含訊息驅動 Bean 應用程式相關資訊的表格。如果在「連結」直欄中已選取啟動規格圓鈕，且「目標資源 JNDI 名稱」欄位包含已取出此可延續訂閱的啟動規格之 JNDI 名稱，則表示此訂閱仍在使用中而無法刪除。
 - 如果找不到使用啟動規格的任何訊息驅動 Bean 應用程式，則可刪除此可延續訂閱。

程序

一旦已識別「孤立」的可延續訂閱，則可使用「IBM MQ Explorer」或 IBM MQ 指令行公用程式 **runmqsc** 來刪除它。

若要使用「IBM MQ Explorer」來刪除「孤立」的可延續訂閱，請執行下列動作：

1. 強調顯示此訂閱的項目
2. 用滑鼠右鍵按一下項目，然後選取 **刪除 ...** 功能表。即會顯示確認視窗。
3. 檢查確認視窗中顯示的訂閱名稱是否正確，然後按一下**是**。

「IBM MQ Explorer」現在即會從「發佈/訂閱」引擎中刪除此訂閱，並清理與其相關聯的任何內部資源（例如為已在其中登錄可延續訂閱的主題所發佈的未處理訊息）。

如果要使用 IBM MQ 指令行公用程式 **runmqsc** 來刪除「孤立」可延續訂閱，必須執行指令 **delete sub**：

1. 開啟命令提示字元階段作業
2. 導覽至 *IBM MQ\bin* 目錄
3. 輸入下列指令以啟動 **runmqsc**：

```
runmqsc queue manager name
```

4. 在 **runmqsc** 公用程式已啟動時，輸入：

```
delete sub(Subscription name)
```

其中 *Subscriptionname* 為可延續訂閱的訂閱名稱，它採用下列格式：

- *JMS:queue manager name:client identifier:subscription name*

正在配置 Managed File Transfer

您可以在安裝之後配置 Managed File Transfer 的特性。

您可以利用 IBM MQ 高可用性解決方案來改善 Managed File Transfer 配置的備援。如果您的代理程式使用抄寫的資料佇列管理程式 (RDQM)，則必須將它們配置為使用浮動 IP 位址特性。這表示代理程式會使用相同的 IP 位址來與三個 RDQM 實例中的任何一個進行通訊，並在失效接手時自動重新連接 (請參閱 RDQM 高可用性 及 建立及刪除浮動 IP 位址)。如果您使用多重實例佇列管理程式解決方案，則應用程式會使用不同的 IP 位址來與每一個實例進行通訊，失效接手時由用戶端重新連接處理 (請參閱 [多重實例佇列管理程式](#) 及 [通道與用戶端重新連線](#))。

相關概念

[使用 Managed File Transfer 的提示及要訣](#)

相關工作

[監視 MFT 資源](#)

[利用使用者結束程式自訂 MFT](#)

[配置 MQMFTCredentials.xml](#)

[保護 Managed File Transfer](#)

[指定要使用 MFT 執行的程式](#)

[疑難排解 Managed File Transfer](#)

[管理 Managed File Transfer](#)

相關參考

[MFT 指令](#)

[MFT agent.properties 檔案](#)

[MFT 回復及重新啟動](#)

多平台上的 MFT 配置選項

Managed File Transfer 提供了一組內容檔，這些檔案包含有關您的設定的金鑰資訊，是作業所必要的。這些內容檔位於安裝產品時所定義的配置目錄中。

您可以具有多組配置選項，每一組配置選項包含一組目錄及內容檔。這些內容檔中定義的值用作所有 Managed File Transfer 指令的預設參數，除非您明確在指令行上指定不同的值。

若要變更您正在使用的預設配置選項集，您可以使用 **fteChangeDefaultConfigurationOptions** 指令。若要變更您正在用於個別指令的配置選項集，您可以將 **-p** 參數與任何 Managed File Transfer 指令搭配使用。

配置選項集的名稱即是協調佇列管理程式的名稱，建議不要變更。您仍可變更配置選項集的名稱，但必須變更 **config** 和 **logs** 目錄的名稱。在下列範例中，配置選項集的名稱會以 **coordination_qmgr_name** 顯示。

配置選項目錄結構

當您配置產品時，會在配置目錄的下列結構中建立目錄及內容檔。您也可以使用下列指令來變更這些目錄及內容檔：**fteSetupCoordination**、**fteSetupCommands**、**fteChangeDefaultConfiguration** 及 **fteCreateAgent**。

```
MQ_DATA_PATH/mqft/  
  config/  
    coordination_qmgr_name/  
      coordination.properties  
      command.properties  
      agents/  
        agent_name/  
          agent.properties  
          exits  
      loggers/  
        logger_name  
          logger.properties  
  installations/  
    installation_name/  
      installation.properties
```

coordination_qmgr_name 目錄是配置選項目錄。配置目錄中可能存在多個配置選項目錄。**agent_name** 目錄是代理程式目錄。除了包含 **agent.properties** 檔案外，此目錄包含 **exits** 目錄 (使用者結束程式常

式及 **fteCreateBridgeAgent** 與 **fteCreateCDAgent** 指令產生的各種 XML 檔案的預設位置)。配置選項集的 agents 目錄中可能具有多個代理程式目錄。

內容檔

installation.properties

installation.properties 檔案指定預設配置選項集的名稱。此項目將 Managed File Transfer 指向一組結構化目錄以及包含要使用之配置的內容檔。通常一組配置選項的名稱是相關聯協調佇列管理程式的名稱。如需 installation.properties 檔案的相關資訊，請參閱 [MFT installation.properties 檔案](#)。

coordination.properties

coordination.properties 檔案指定協調佇列管理程式的連線詳細資料。因為數個 Managed File Transfer 安裝可能共用相同的協調佇列管理程式，您可以使用符號鏈結來連接共用磁碟機上的通用 coordination.properties 檔。如需 coordination.properties 檔案的相關資訊，請參閱 [MFT coordination.properties 檔案](#)。

command.properties

MFT command.properties 檔指定當您發出指令時要連接的指令佇列管理程式，以及 Managed File Transfer 聯絡該佇列管理程式所需的資訊。如需 command.properties 檔案的相關資訊，請參閱 [MFT command.properties 檔案](#)。

agent.properties

每一個 Managed File Transfer Agent 都有自己的內容檔 agent.properties，其中必須包含代理程式用來連接至其佇列管理程式的資訊。agent.properties 檔也包含變更代理程式行為的內容。如需 agent.properties 檔案的相關資訊，請參閱 [MFT agent.properties 檔案](#)。

logger.properties

logger.properties 檔案指定日誌程式的配置內容。如需 logger.properties 檔案的相關資訊，請參閱 [MFT 日誌程式配置內容](#)。

內容檔及字碼頁

由於 Java 的限制，所有 Managed File Transfer 內容檔的內容必須保留美式英文。如果在非美式英文系統上編輯內容檔，您必須使用 Unicode ESC 序列。

z/OS 上的 MFT 配置選項

z/OS 上的 Managed File Transfer 配置選項與分散式平台的選項相同。

如需多平台上配置選項的相關資訊，請參閱第 565 頁的『多平台上的 MFT 配置選項』。

在 z/OS 上，配置位置由環境變數 BFG_DATA 定義。如果 BFG_DATA 所參照的「UNIX 系統服務」目錄下尚無配置存在，則會由 MFT 指令 PDSE 程式庫資料集的 BFGCUSTOM JCL Script 產生建立配置所需的工作。然後，配置將會在您執行這些產生的工作時建立。建立配置時，需使用對可存取的現有目錄進行參照的 BFG_DATA。

您也可以使用 Multiplatforms 及 z/OS 上可用的相同 **fte** 指令來建立及維護配置。如需 **fte** 指令的清單，請參閱 [MFT 指令](#)。

Windows Linux V9.1.0 配置 Redistributable Managed File Transfer

Agent

您可以配置 Redistributable Managed File Transfer Agent 以連接至現有 IBM MQ 基礎架構，並可讓使用者傳送檔案，而無需安裝 IBM MQ 以取得 Managed File Transfer 功能。

開始之前

如需 Redistributable Managed File Transfer Agent 可重新配送的授權條款相關資訊，請參閱 [IBM MQ 可重新配送元件](#)。

Redistributable Managed File Transfer Agent 提供 Managed File Transfer 的功能，但下列情況例外：

- 不支援將模式連線連結到協調、指令及代理程式佇列管理程式，您必須使用用戶端模式連線。當您發出指令時，當您使用安裝為 IBM MQ 一部分的 Managed File Transfer 時，必須提供選用參數：佇列管理程式主機、埠、名稱及通道名稱。
- 不包括下列指令：

- fteCreateCDAgent.cmd
- fteCreateLogger.cmd
- fteDeleteLogger.cmd
- fteMigrateLogger.cmd
- fteSetLoggerTraceLevel.cmd
- fteShowLoggerDetails.cmd
- fteStartLogger.cmd
- fteStopLogger.cmd

如需可用指令的完整清單，請參閱 [已安裝的 MFT 指令集](#)。

- 不支援 Managed File Transfer Connect:Direct。
- 不包括「IBM MQ Explorer」。

Windows 您必須在系統上安裝下列 Microsoft 程式庫，才能使用 Redistributable Managed File Transfer Agent：

- Microsoft Visual C++ Redistributable 2008
- Microsoft Visual C++ Redistributable 2012

可從 Microsoft 取得這些程式庫。請參閱 [最新支援的 Visual C++ 下載](#)。

註：Redistributable Managed File Transfer package 不支援 Advanced Message Security。

關於這項作業

您可以選擇性地下載並配置 Redistributable Managed File Transfer Agent 以連接至現有 IBM MQ 基礎架構，以容許使用者在其本端環境與現有 IBM MQ 基礎架構之間傳送檔案，而無需他們安裝 IBM MQ。請遵循下列步驟來下載 Redistributable Managed File Transfer Agent，並進行解壓縮：

程序

1. 從 Fix Central 下載 [IBM MQ 可重新配送的 Managed File Transfer 代理程式套件](#)。

- a) 為您的作業系統選擇套件：

保存檔名或 .zip 檔名說明檔案內容及對等維護層次。

V 9.1.0 例如，對於 IBM MQ 9.1.0，檔名如下：

- **Windows** 9.1.0.0-IBM-MQFA-Redist-Win64
- **Linux** 9.1.0.0-IBM-MQFA-Redist-LinuxX64
- **Linux** 9.1.0.0-IBM-MQFA-Redist-LinuxS390X
- **Linux** 9.1.0.0-IBM-MQFA-Redist-LinuxPPC64LE

- b) 識別您要在其中解壓縮套件的目錄，例如：

- **Windows** C:\MFTZ
- **Linux** /home/MFTZ

2. 將所下載套件的內容解壓縮：

- **Windows** 在 Windows 上，使用 Windows Explorer 工具來擷取。

- **Linux** 在 Linux 上，解壓縮並 untar，如下所示：

```
gunzip 9.0.1.0-IBM-MQFA-Redist-LinuxX64.tar.gz
```

接著，

```
tar xvf 9.0.1.0-IBM-MQFA-Redist-LinuxX64.tar
```

會建立下列目錄：

- **Windows** **Linux** bin: 包含所有必要的 MFT 指令
- **Windows** bin64: 包含 Windows 64 位元 OS 支援所需的必要程式庫
- **Windows** **Linux** java: 包含 IBM JRE 和 IBM MQ 程式庫
- **Windows** **Linux** licenses: 包含授權檔
- **Windows** **Linux** mqft: 包含 Ant 支援及核心 MFT 功能支援所需的 ant 及 lib 目錄
- **Windows** **Linux** swtag: 包含 swidtag 檔案，授權管理程式需要此檔案來識別機器上的安裝

下一步

您已準備好配置 MFT 代理程式。如需後續步驟，請參閱 [第 568 頁的『建立 Redistributable Managed File Transfer Agent 的起始配置』](#)。

相關概念

[配置可重新配送的 MFT 代理程式時可能發生的錯誤](#)

Windows **Linux** **V 9.1.0** 建立 Redistributable Managed File Transfer Agent 的起始配置

您可以配置 Managed File Transfer Agent 以連接至現有 IBM MQ 配置。

開始之前

請務必下載 Redistributable Managed File Transfer Agent 套件的內容，並進行解壓縮。如需相關資訊，請參閱 [第 566 頁的『配置 Redistributable Managed File Transfer Agent』](#)。

關於這項作業

您先建立 Redistributable Managed File Transfer Agent 需要的環境。然後，您可以設定與在 IBM MQ 伺服器上執行之佇列管理程式的連線功能，然後在啟動及驗證代理程式之前配置代理程式及代理程式佇列管理程式。

程序

1. 為 Redistributable Managed File Transfer Agent 建立環境。

當您執行 **fteCreateEnvironment** 指令時，會建立 MFT 資料目錄，其中含有 MFT 代理程式配置資訊。確保您所在的 bin 目錄是在您將所下載的 MFT 代理程式可重新配送元件解壓縮時建立的目錄。請執行下列指令：

- **Windows**

```
fteCreateEnvironment.cmd -d datapath location
```

- **Linux**

```
./fteCreateEnvironment -d datapath location
```


此指令採用下列選用參數:

-d

此參數指定在其中建立、儲存及維護 MFT 配置的資料路徑位置。如果您在未指定資料位置的情況下執行 **fteCreateEnvironment**，則會在解壓縮 Redistributable Managed File Transfer Agent 的位置建立 mftdata 目錄。

註: 如果可重新配送的代理程式將以 Windows 服務方式執行，則需要在系統環境中設定 **BFG_DATA** 環境變數，服務才能運作。

V 9.1.2 **-n** 安裝名稱

此參數用於指定 IBM MQ 安裝的名稱或唯一名稱。

您可能想要使用此參數的狀況範例如下:

- 如果您想要快速測試新功能或特性，請搭配使用可重新配送的套件與現有的配置，其中代理程式已配置為僅以用戶端模式連接至佇列管理程式。(請注意，此參數不適用於以連結模式配置連接至佇列管理程式的任何代理程式。)
- 如果您要從標準 Managed File Transfer 安裝架構移轉至 Redistributable Managed File Transfer Agent 套件，且您想要使用標準安裝架構所建立的相同配置。在這種情況下，已安裝標準 Managed File Transfer，但正在連接另一部機器上執行的代理程式佇列管理程式。

預設安裝名稱變數為 **BFG_INSTALLATION_NAME**。

如需 **fteCreateEnvironment** 指令的相關資訊，請參閱 [fteCreate 環境 \(設定 Redistributable Managed File Transfer Agent 的環境\)](#)。

您還可以設定環境變數 **BFG_DATA** 來搭配資料路徑位置:

```
BFG_DATA=Datapath location
```

在建立、啟動及停止代理程式或其他任何指令之前，您必須確保 **BFG_DATA** 變數是設為正確的資料路徑位置。

2. 設定 IBM MQ 連線功能。

a) 使用 **fteSetupCoordination** 指令來設定協調佇列管理程式。

fteSetupCoordination 指令會建立協調佇列管理程式所需的設定，以及進一步配置所需的目錄。Redistributable Managed File Transfer Agent 是以用戶端模式運作，因此您必須在此指令中提供額外參數來避免錯誤，因為連結模式不受支援。

```
fteSetupCoordination -coordinationQMgr PRMFTDEM02  
-coordinationQMgrHost 9.121.59.233 -coordinationQMgrPort 3002  
-coordinationQMgrChannel SYSTEM.DEF.SVRCONN
```

如需使用 **fteSetupCoordination** 指令的詳細資料及步驟，請參閱 [fteSetupCoordination](#)。如需如何配置協調佇列管理程式的相關資訊，請參閱第 600 頁的『配置 MFT 的協調佇列管理程式』。

b) 建立及設定指令佇列管理程式:

```
fteSetupCommands -p PRMFTDEM02 -connectionQMgrHost 9.121.59.233  
-connectionQMgrPort 3002 -connectionQMgrChannel SYSTEM.DEF.SVRCONN  
-connectionQMgr PRMFTDEM02 -f
```

如需使用 **fteSetupCommands** 指令的詳細資料及步驟，請參閱 [fteSetup 指令: 建立 MFT command.properties 檔案](#)。

3. 為端點建立 MFT 代理程式定義。

```
fteCreateAgent -p PRMFTDEM02 -agentQMgrHost 9.121.59.233  
-agentQMgrPort 3002 -agentQMgrChannel SYSTEM.DEF.SVRCONN  
-agentName AGENT.TRI.BANK -agentQMgr PRMFTDEM02 -f
```

如需使用 **fteCreateAgent** 指令來配置代理程式及代理程式佇列管理程式的相關資訊，請參閱 [fteCreateAgent](#)。

在每一個代理程式的第 569 頁的『2』和第 569 頁的『3』步驟中，您會在代理程式佇列管理程式上建立佇列和主題定義。

4. 啟動代理程式，然後您就可以開始傳送檔案。

```
fteStartAgent -p PRMFTDEM02 AGENT.TRI.BANK
```

您可以透過執行下列指令來驗證代理程式的狀態：

```
fteListAgents
```

如需使用 **fteListAgents** 指令的詳細資料，請參閱 [fteList 代理程式](#)。

相關概念

第 564 頁的『正在配置 Managed File Transfer』

您可以在安裝之後配置 Managed File Transfer 的特性。

第 565 頁的『多平台上的 MFT 配置選項』

Managed File Transfer 提供了一組內容檔，這些檔案包含有關您的設定的金鑰資訊，是作業所必要的。這些內容檔位於安裝產品時所定義的配置目錄中。

相關參考

fteCreateTransfer: 啟動新的檔案傳送

z/OS 建立 MFT 代理程式或日誌程式指令資料集

您可以從 Managed File Transfer 指令範本資料集，為特定的 Managed File Transfer Agent 或 Managed File Transfer Logger 建立指令的 PDSE 資料集，以進行特定協調。

關於這項作業

請完成下列步驟：

程序

1. 複製 MFT 指令範本 PDSE 程式庫資料集 SBFGCMD5。

z/OS SBFGCMD5 必須複製到新的程式庫，例如 *prefix.agent.JCL_*。進行下列取代作業時，您可以使用經過更新的 SBFGCMD5(BFGCOPY) 成員：

- 將 *++supplied-library++* 取代為 SBFGCMD5 PDSE 的完整名稱。
- **z/OS** 將 *++service-library++* 取代為新的 MFT 指令 PDSE 程式庫資料集的完整名稱。+ *+service-library++* 是建立的代理程式或日誌程式服務的輸出資料集。

2. 針對新的 MFT 指令 PDSE 程式庫資料集，編輯成員 BFGCUSTM，這是用來為代理程式或日誌程式自訂指令的 JCL Script。每一個變數都會以 *++variable name++* 的格式指定，您必須將其取代為必要值。如需各個 JCL 變數的說明，請參閱第 581 頁的『z/OS JCL 變數』。BFGSTDIN DD 陳述式會定義三個種類的變數：Variables、Properties 及 Environment。此陳述式採用下列格式：

```
[Variables]
variable1=value1
variable2=value2
....
variableN=valueN
[Properties]
property1=property value1
property2=property value2
...
propertyN=property valueN
[Environment]
custom_variable1=value1
custom_variable2=value2
....
custom_variableN=valueN
```

變數會定義每一個指令所需的一組設定及環境變數。

內容定義會置換 MFT 配置內容。您可以視需要新增代理程式或日誌程式內容，為您的環境自訂代理程式或日誌程式。如需所有內容的清單，請參閱第 590 頁的『配置內容檔』。提供這項機制，是為了免去您存取 MFT 配置內容檔（以「UNIX 系統服務」檔案形式進行維護）的需要。

Environment 會定義所有額外需要的自訂環境變數。

3. 提交新的 MFT 指令 PDSE 程式庫資料集的工作 BFGCUSTM。此工作會產生適用於代理程式或日誌程式的 JCL 指令集，作為 PDSE 的新成員。如需指令的完整清單，請參閱第 584 頁的『z/OS 代理程式及日誌程式指令 JCL Script』。

工作 BFGCUSTM 會對包括 DISP=OLD 之 DD 陳述式的 JCL 所在的程式庫進行更新。您必須在提交之後結束編輯器，讓工作執行。

請檢查輸出工作日誌，以確認 JCL Script 已順利執行。如果執行失敗，請予以更正，然後重新提交 BFGCUSTM 工作。

BFGCUSTM JCL Script 也會在必要時更新「UNIX 系統服務」MFT 配置內容檔，使檔案保持一致。如果 CoordinationQMGr 內容定義的配置不存在，則會輸出警告訊息，而且您必須執行產生的 BFGCFGR 和 BFGCMCR 工作，才能建立配置內容檔。必須針對代理程式執行 BFGAGCR，而針對日誌程式編輯執行 BFGLGCRS。如果指定的配置已存在，該配置將會以 BFTCUSTM JCL Script 中定義的任何內容進行更新。

相關概念

第 566 頁的『z/OS 上的 MFT 配置選項』

z/OS 上的 Managed File Transfer 配置選項與分散式平台的選項相同。

相關工作

第 580 頁的『在 z/OS 上更新現有 MFT 代理程式或日誌程式指令資料集』

您可以更新從 Managed File Transfer 指令範本資料集建立的 Managed File Transfer 指令 PDSE 程式庫資料集。

z/OS 正在配置 Managed File Transfer for z/OS

Managed File Transfer for z/OS 需要經過自訂，元件才能正常運作。

關於這項作業

您需要：

1. 編輯 PDSE 成員，以指定配置資料
2. 定義協調佇列管理程式。
3. 定義指令佇列管理程式
4. 配置一或多個代理程式
5. 選用項目：配置日誌程式作業，以將資料儲存在 Db2 中

下列主題詳細說明所需執行之作業的順序。

相關概念

第 571 頁的『檢閱 MFT 配置』

在開始之前，您需要先檢閱您的系統配置。

相關工作

[正在安裝 Managed File Transfer for z/OS](#)

z/OS 檢閱 MFT 配置

在開始之前，您需要先檢閱您的系統配置。

Managed File Transfer (MFT) 需要一或多個佇列管理程式，才能為每個已定義的 MFT 配置扮演下列角色：

- 協調佇列管理程式，用於維護配置中各個代理程式狀態的相關資訊，這些資訊發佈至協調程式的相關主題中。

- 一或多個指令或連線佇列管理程式，作為 MFT 指令的 IBM MQ 網路進入點。
- 一或多個代理程式佇列管理程式，其負責在 MFT 代理程式與 IBM MQ 網路之間提供通訊。

上述每一個角色可由不同的佇列管理程式執行，您也可以將這些角色結合在一個最簡單的配置中，讓單一佇列管理程式執行所有角色。

如果要在現有的 MFT 環境中新增 z/OS 佇列管理程式，您需要在 z/OS 佇列管理程式與此配置的其他佇列管理程式之間定義連線功能。您可以透過手動定義傳輸佇列或利用叢集作業來達到此目的。

每個 MFT 代理程式與單一佇列管理程式進行通訊。如果有多個代理程式與同一個佇列管理程式通訊，則代理程式佇列管理程式將為每個代理程式定義多個佇列：

- SYSTEM.FTE.COMMAND.agent_name
- SYSTEM.FTE.DATA.agent_name
- SYSTEM.FTE.REPLY.agent_name
- SYSTEM.FTE.STATE.agent_name
- SYSTEM.FTE.EVENT.agent_name
- SYSTEM.FTE.AUTHAGT1.agent_name
- SYSTEM.FTE.AUTHTRN1.agent_name
- SYSTEM.FTE.AUTHOPS1.agent_name
- SYSTEM.FTE.AUTHSCH1.agent_name
- SYSTEM.FTE.AUTHMON1.agent_name
- SYSTEM.FTE.AUTHADM1.agent_name

請注意，您可以定義通用安全設定檔，在其中使用諸如 SYSTEM.FTE.COMMAND.* 之類的設定檔，也可以為每個代理程式定義特定的設定檔。

相關概念

第 572 頁的『開始之前』

Managed File Transfer (MFT) 配置會使用 UNIX 系統服務 (USS) 和 PDSE 資料集中的檔案。

z/OS 開始之前

Managed File Transfer (MFT) 配置會使用 UNIX 系統服務 (USS) 和 PDSE 資料集中的檔案。

大部分的配置和作業都可使用 PDSE 中的 JCL 來完成，而您必須熟悉 USS 環境中的運作方式。

您可以透過 ISPF 或在您的工作站上透過指令使用 Telnet 類型階段作業（例如 Telnet Putty 或 SSH）來存取 OMVS。

如果您透過 ISPF 使用 OMVS，則可以使用標準 ISPF 編輯器，並瀏覽指令 **oedit** 和 **obrowse**。

您必須熟悉下列 USS 指令

指令	函數
chmod xxx path	變更檔案存取權。
df -k path	報告檔案系統中剩餘的可用空間。 -k 會以 KB 為單位報告可用空間。
du -kt path	報告路徑下的目錄大小。報告的大小 (以 KB 為單位)。
find path -name xxx	在路徑目錄中搜尋名為 xxx 的檔案。xxx 區分大小寫，且可以類似於 *zzz。
ls -ltrd directory	列出指定目錄（而不是目錄中的檔案）的相關資訊。
ls -ltr path	列出路徑中檔案的相關資訊。

表 41: 一般 UNIX 系統服務指令 (繼續)	
指令	函數
obrowse filename	瀏覽檔名。
oedit filename	在 OMVS 中編輯檔案。

檢閱下表中的項目，並以您的企業適用的項目完成表格。您在編輯成員 `BFGCUSTM` 時將需要這些值。

表 42: 成員 <code>BFGCUSTM</code> 所需的參數		
姓名	範例資料	Comments
ADMIN_JOB1		工作卡。所有工作皆以相同的 JCL 卡產生。
armELEMENT	如果使用 ARM，請為此代理程式或日誌程式使用 ARM 原則中指定的 ARM ELEMENT 值。如果未使用 ARM，請將此參數設為空白；例如 armELEMENT=	
armELEMENTYPE	如果使用 ARM，請使用 ARM 原則中指定的 ARM ELEMENTYPE。例如，為代理程式使用 armELEMENTYPE=SYSBFGAG，或為日誌程式使用 armELEMENTYPE=SYSBFGLG。如果未使用 ARM，請將此參數設為空白；例如 armELEMENTYPE=	
BFG_DATA		視需要填寫
BFG_GROUP_NAME	MQM	
BFG_JAVA_HOME	/java/java71_bit64_GA/J7.1_64/	
BFG_JVM_PROPERTIES		視需要填寫
BFG_PROD	/var/ibm/wmqmft	
BFG_WTO	YES	用以取得 syslog 上的 MFT 訊息。
CLEAN_AGENT_PROPS	-trs	此參數指定在執行 BFGAGCL 成員時將用來清除代理程式的選項。如需此參數的有效值相關資訊，請參閱 fteClean 代理程式: 清除 MFT 代理程式 。
coordinationQMgr	MQPV	必要配置
CREDENTIAL_PATH		用於移轉
Db2_HLQ	SYS2.Db2。第 10 版	
DB_PROPS_PATH		用於移轉
FTE_CONFIG		用於移轉
JOBCARD1		這是長時間執行作業、代理程式和日誌程式的工作卡。
LIBRARY	SCEN.FTE.JCL	MFT PDSE 的名稱。您需要每個代理程式或日誌程式作業的副本。

表 42: 成員 BFGCUSTM 所需的參數 (繼續)		
姓名	範例資料	Comments
MQ_HLQ	IBM MQ 資料集的高階限定元。例如 MQM.V915	
MQ_LANG	E	
MQ_PATH	/mqm/V9R1M5	
名稱	AGENT1	
OUTPUT_CLASS	*	
PATH	bin:/usr/bin:/usr/sbin	
productId	MFT	此參數用來設定要記錄其 Managed File Transfer 使用情形的產品類型。如需此參數的有效值相關資訊，請參閱 fteSetProductId: 設定 z/OS SCRT 記錄產品 ID。
QMGR	MQPV	
SERVICE_TYPE	AGENT 或 LOGGER	
TMPDIR	/tmp	讀取和寫入暫存檔可存取的 USS 路徑。

此外，您必須檢閱下列變數並提供必要的值：

- coordinationQMGrHost=
- coordinationQMGrPort=
- coordinationQMGrChannel=
- connectionQMGr=
- connectionQMGrHost=
- connectionQMGrPort=
- connectionQMGrChannel=

這些內容是 AGENT 或 LOGGER 通用的。

註：「主機」、「埠」和「通道」是用戶端連線的必要項目，且對於本端機器上的連結連線不應保留為空白。

相關概念

第 574 頁的『[要檢查的項目](#)』

確定您有足夠的磁碟空間、用來儲存資料的目錄，且必要的檔案皆存在。

第 577 頁的『[編輯成員 BFGCUSTM](#)』

您必須先編輯成員 BFGCUSTM，並為您的企業所使用的參數輸入值，才能執行工作。

要檢查的項目

確定您有足夠的磁碟空間、用來儲存資料的目錄，且必要的檔案皆存在。

確認您有足夠的磁碟空間

確認您要用來儲存配置專用檔案的檔案系統上有足夠的磁碟空間可供使用。

如果代理程式追蹤已啟用，依預設將可使用 100 MB 的磁碟空間。

配置檔本身並不大，只有幾 KB 而已。

如果您預計要使用兩個代理程式和一個日誌程式，則至少需要 300 MB。您可以使用指令 **df -k path**，其中，**path** 是安裝專用檔案的位置。這會指出以 KB 為單位的可用空間和空間總計。

300 MB 是 307,200 KB，因此您至少應提供 310,000 KB

建立並檢查用來儲存 Managed File Transfer 資料的目錄

您必須有用來儲存 Managed File Transfer (MFT) 資料的目錄。

確認您在檔案系統 **df -k /var** 中有足夠的空間。此檔案系統至少應有 310,000 KB 可供使用。

如果您尚未建立此檔案系統，請使用 **mkdir** 指令；例如 **mkdir /var/mft**。

使用指令 **ls -ltrd /var/mft** 顯示使用者對此目錄具有的許可權。

如果擁有者或群組不正確，請使用指令 **chown owner:group /var/mft**。

如果群組的權限不正確，請使用下列指令為擁有者和群組提供讀取、寫入和執行權限。請注意，下列指令也會為所有使用者提供讀取和執行權限：**chmod 775 /var/mft**。

確認檔案存取，且您具有其存取權。

對您在自訂期間所將使用的檔案使用 **ls -ltr** 指令。例如：

```
ls -ltrd /java/java71_bit64_GA/J7.1_64/bin
```

會產生

```
drwxr-xr-x 4 SYSTASK TSOUSER 8192 Nov 15 2013 /java/java71_bit64_GA/J7.1_64/bin
```

其中，**drwxr-xr-x** 表示

d

這是目錄。

rwX

擁有者 **SYSTASK** 具有目錄的讀取、寫入和執行權限。

r-x

群組 **TSOUSER** 中的人員可讀取和執行目錄中的檔案。

r-x

通用存取，即任何人皆可讀取或執行目錄中的檔案。

檢查以下指定的檔案：

表 43: 使用者對特定檔案所需的存取權	
路徑	使用者執行配置所需的存取權
BFG_JAVA_HOME	讀取和執行
/tmp	讀取和寫入
BFG_PROD	讀取
BFG_DATA	寫入
MQ_PATH	讀取

相關概念

[第 572 頁的『開始之前』](#)

Managed File Transfer (MFT) 配置會使用 UNIX 系統服務 (USS) 和 PDSE 資料集中的檔案。

[第 576 頁的『一般 MFT for z/OS 配置』](#)

不同 Managed File Transfer 配置的概觀

不同 Managed File Transfer 配置的概觀

Managed File Transfer 會使用連結至佇列管理程式的代理程式來傳送資料。

MFT 可以使用多個佇列管理程式：

- 一個以上用於傳送資料的佇列管理程式。
- 一個用於發出要求的指令佇列管理程式。例如，啟動傳送的要求傳送到此佇列管理程式之後，相關聯的指令將遞送至 MFT 代理程式。
- 一個用於管理工作的協調佇列管理程式。

有三個一般 Managed File Transfer (MFT) 配置：

1. 單一佇列管理程式與一個以上使用本端連線的代理程式。這可能是用來將資料集的內容放置到 IBM MQ 佇列中。
2. 單一佇列管理程式與一個在分散式機器上使用用戶端連結的 MFT 用戶端。
3. 由通道連接的兩個佇列管理程式，且在每部機器上各有一個以上的代理程式。這些代理程式可以是用戶端或本端連結。

請注意下列幾點：

1. MFT 是以 Java 撰寫，其中包含用於配置及操作 MFT 的部分 Shell Script 及 JCL。
2. Db2 狀態及活動可以記載並儲存在 Db2 表格中。
3. 配置 MFT 的人員必須熟悉「UNIX 系統服務 (USS)」。

- 含有名為 /u/userID/myfile.txt2 的檔案的目錄結構
- USS 指令，例如：

cd (切換目錄)

ls (列出)

chmod (變更檔案許可權)

chown (變更檔案所有權或可以存取檔案或目錄的群組)

4. 在 USS 中需要下列產品，才能配置及執行 MFT：

- Java；例如 /java/java71_bit64_GA/J7.1_64/
- IBM MQ 第 800 版，例如 /mqm/V8R0M03。
- Db2 JDBC 程式庫（如果要針對狀態和歷程使用 Db2）；例如 /db2/db2v10/jdbc/lib

您需要協調佇列管理程式。不過，您可以使用相同的佇列管理程式來執行代理程式、處理指令和進行協調。如果您使用多個佇列管理程式，您必須擇一作為協調程式。

檢查您的 IBM MQ 連線功能

如果您具有現有的 MFT 協調佇列管理程式，您在執行配置的佇列管理程式與協調和指令佇列管理程式之間必須有連線。

您需要為每一個代理程式及日誌程式建立 JCL 程式庫。JCL 包含用來建立及執行代理程式或日誌程式的配置及工作。

編輯並執行 BFGCOPY 成員，為每個代理程式和日誌程式建立 IBM 提供的 SBFGCMD5 程式庫的副本。

此程式庫會用來定義代理程式或日誌程式的配置，而在自訂之後，會包含可用來建立必要 Managed File Transfer 配置和代理程式或日誌程式的工作。

您可以在此程序中建立成員 BFGCUSTM。

註：如果您熟悉 USS 指令，您可以用您在其他平台上使用的相同指令來配置 z/OS。

相關概念

第 576 頁的『[一般 MFT for z/OS 配置](#)』
不同 Managed File Transfer 配置的概觀

第 577 頁的『[編輯成員 BFGCUSTM](#)』

您必須先編輯成員 BFGCUSTM，並為您的企業所使用的參數輸入值，才能執行工作。

z/OS 編輯成員 BFGCUSTM

您必須先編輯成員 BFGCUSTM，並為您的企業所使用的參數輸入值，才能執行工作。

請參閱成員 BFGCUSTM 所需的參數，以取得需要特定值的參數清單。

此外，您必須檢閱下列變數並提供必要的值：

- coordinationQMGrHost=
- coordinationQMGrPort=
- coordinationQMGrChannel=
- connectionQMGr=
- connectionQMGrHost=
- connectionQMGrPort=
- connectionQMGrChannel=

這些內容是 AGENT 或 LOGGER 通用的。

註：「主機」、「埠」和「通道」是用戶端連線的必要項目，且對於本端機器上的連結連線不應保留為空白。

如果這是您 Managed File Transfer 環境中的第一個佇列管理程式，且您要讓協調、指令和執行中的代理程式使用相同的佇列管理程式，請將值設定為本端佇列管理程式名稱。

```
coordinationQMGr=MQPV  
connectionQMGr=MQPV
```

其中，MQPV 是您的本端佇列管理程式名稱。

提交工作，以更新 PDSE，並在指定的路徑下建立目錄結構。

請注意，這項工作需要專用權，因此在工作執行時請停止使用 PSDE。

提示：當您提交工作 BFGCUSTM 時，此工作一律會取代所有的 JCL 檔案。您應將您變更的每個成員重新命名。

相關概念

第 572 頁的『[開始之前](#)』

Managed File Transfer (MFT) 配置會使用 UNIX 系統服務 (USS) 和 PDSE 資料集中的檔案。

第 579 頁的『[建立代理程式](#)』

您必須複製 PDSE 以建立代理程式專用的 PDSE，例如 *user.MFT.AGENT1*。如果先前的代理程式或日誌程式配置中存在 PDSE，請加以複製。如果這是您的第一個配置，請複製隨附於 MFT 的 PDSE。

z/OS 定義協調佇列管理程式

Managed File Transfer 要求將佇列管理程式建立為協調佇列管理程式。

根據您所選擇的配置，此佇列管理程式會位於本端 MVS 系統或另一部機器上。如果是前者，其連線將會是連結連線；如果是後者，則是用戶端連線。

在您順利執行配置步驟後，PDSE 中會有經配置的成員。

成員 BFGCFR 會定義協調佇列管理程式，而這項工作會：

1. 在 Managed File Transfer (MFT) 目錄中建立目錄結構，並建立配置檔。

2. 執行 CSQUTIL 以定義 IBM MQ 資源。

如果協調佇列管理程式位於遠端機器上，則此工作會失敗。

成員 BCFCFCR 會在 USS 中建立檔案，並建立 MQ 定義。這項工作會：

1. 建立 MFT 主題、
2. 建立 MFT 佇列
3. 將 `NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)` 變更為
`NAMES(SYSTEM.BROKER.DEFAULT.STREAM, SYSTEM.BROKER.ADMIN.STREAM, SYSTEM.FTE)`
4. 執行 `ALTER QMGR PSMODE(ENABLED)`

在執行變更之前，會發出 `DISPLAY NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)` 指令。如果您的 `NAMLIST` 不是預設的，您應變更名稱清單，將 `SYSTEM.FTE` 新增至您的名稱清單

以您自己的字首（例如 `CCPCFCR`）重新命名成員 `BCFCFCR`，因為重新自訂此檔案會加以取代。

插入認證檔的名稱，以編輯這個重新命名過的成員。例如：

```
%BFGCMD CMD=fteSetupCoordination +  
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>'
```

儲存並提交工作。請注意，如果您需要重新提交工作，您必須新增 `-f` 選項。

這項工作執行時，會列出它所建立的 IBM MQ 資源。您必須保護這些資源。

```
DEFINE TOPIC('SYSTEM.FTE') TOPICSTR('SYSTEM.FTE') REPLACE  
ALTER TOPIC('SYSTEM.FTE') NPMSGDLV(ALLAVAIL) PMSGDLV(ALLAVAIL)  
DEFINE QLOCAL(SYSTEM.FTE) LIKE(SYSTEM.BROKER.DEFAULT.STREAM) REPLACE  
ALTER QLOCAL(SYSTEM.FTE) DESCR('Stream for MFT Pub/Sub interface')  
* Altering namelist: SYSTEM.QPUBSUB.QUEUE.NAMELIST  
* Value prior to alteration:  
DISPLAY NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST)  
ALTER NAMELIST(SYSTEM.QPUBSUB.QUEUE.NAMELIST) +  
NAMES(SYSTEM.BROKER.DEFAULT.STREAM+  
,SYSTEM.BROKER.ADMIN.STREAM,SYSTEM.FTE)  
* Altering PSMODE. Value prior to alteration:  
DISPLAY QMGR PSMODE  
ALTER QMGR PSMODE(ENABLED)
```

相關工作

第 578 頁的『定義指令佇列管理程式』

您可以使用與協調和指令佇列管理程式相同的佇列管理程式，或者建立新的指令佇列管理程式。

定義指令佇列管理程式

您可以使用與協調和指令佇列管理程式相同的佇列管理程式，或者建立新的指令佇列管理程式。

關於這項作業

您必須具有指令佇列管理程式，但您的協調和指令佇列管理程式可以使用相同的佇列管理程式。否則，您必須建立新的指令佇列管理程式。它可以位於與協調佇列管理程式相同的機器上，但不一定要如此。

程序

1. 使用專屬字首（例如 `CCPCMCR`）來重新命名成員 `BFGCMCR`。
您必須重新命名 `BFGCMCR`，因為重新自訂此檔案會取代此檔案。
2. 插入認證檔的名稱，以編輯已重新命名的成員。

例如：

```
%BFGCMD CMD=fteSetupCommands +  
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>' +
```

3. 儲存並提交工作。

請注意，如果您需要重新提交工作，您必須新增 `-f` 選項。

此佇列管理程式會用於 **ftePingAgent** 之類的指令。

4. 檢閱此成員、加以提交，並檢閱輸出。

下一步

如需如何建立代理程式的相關資訊，請參閱第 579 頁的『[建立代理程式](#)』。

相關概念

第 577 頁的『[定義協調佇列管理程式](#)』

Managed File Transfer 要求將佇列管理程式建立為協調佇列管理程式。

相關工作

[配置 MQMFTCredentials.xml](#)

相關參考

[MFT 認證檔案格式](#)

建立代理程式

您必須複製 PDSE 以建立代理程式專用的 PDSE，例如 `user.MFT.AGENT1`。如果先前的代理程式或日誌程式配置中存在 PDSE，請加以複製。如果這是您的第一個配置，請複製隨附於 MFT 的 PDSE。

檢閱成員 `BFGCUSTM`，如果您需要使用不同的認證檔，請加以建立。

有許多內容仍與第 577 頁的『[編輯成員 BFGCUSTM](#)』中詳述的自訂作業相同。

您必須變更：

- `//SYSEXEC DD DSN=SCEN.FTE.JCL.AGENT1`
- `LIBRARY`，以符合代理程式 PDSE
- `SERVICE_TYPE=AGENT`
- `NAME`，使其成為代理程式（符合 PDSE）JOB CARD 的名稱
- 變更 `BFG_JVM_PROPERTIES="-Xmx1024M"`

提交此工作；請切記，此工作需要資料集的互斥存取權。

代理程式的工作全都會有 `BFGAG*` 格式的名稱

重新命名成員 `BFGAGCR`。此工作會更新 Managed File Transfer 目錄中的檔案，並使用 `CSQUTIL` 在本端佇列管理程式中建立代理程式專用佇列。指定認證檔的名稱，例如 `-credentialsFile //'SCEN.FTE.JCL.VB(CREDOLD)`。如果未指定此名稱，啟動代理程式的工作就不會使用認證檔。

檢查輸出以確定程序順利執行。

提示：將 `agent.properties` 檔案的路徑名稱從工作的輸出複製到代理程式在 PDSE 中的成員。

例如，將 `/u/userid/fte/wmqmft/mqft/config/MQPA/agents/AGENT1/agent.properties` 複製到成員 `AGENT` 中。

如果您需要顯示內容檔，並新增 `/u/userid/fte/wmqmft/mqft/logs/MQPA/agents/AGENT1/logs` 一行，這將會有幫助。

這是追蹤檔案的存放之處。

相關概念

第 577 頁的『[定義協調佇列管理程式](#)』

Managed File Transfer 要求將佇列管理程式建立為協調佇列管理程式。

第 580 頁的『[使用代理程式](#)』

如何使用各種指令確定代理程式正常運作。

相關工作

第 578 頁的『定義指令佇列管理程式』

您可以使用與協調和指令佇列管理程式相同的佇列管理程式，或者建立新的指令佇列管理程式。

使用代理程式

如何使用各種指令確定代理程式正常運作。

啟動代理程式

重新命名成員 BFGAGST、檢閱成員，然後提交工作。

如果作業成功，您會收到訊息 BFGAG0059I：代理程式已順利啟動。

顯示作用中的代理程式

重新命名成員 BFGAGLI、檢閱成員，然後提交會使用協調佇列管理程式的工作。

您必須解決任何連線問題

Ping 代理程式以確認它可運作

重新命名成員 BFGAGPI、檢閱成員，然後提交會使用指令佇列管理程式的工作。

您必須解決任何連線問題

執行測試傳送

如需進一步資訊，請參閱第 585 頁的『執行驗證傳送』。

停止代理程式

重新命名成員 BFGAGSP、檢閱成員，然後提交工作。

使用成員 BFGAGST 重新啟動代理程式。

相關概念

第 579 頁的『建立代理程式』

您必須複製 PDSE 以建立代理程式專用的 PDSE，例如 *user.MFT.AGENT1*。如果先前的代理程式或日誌程式配置中存在 PDSE，請加以複製。如果這是您的第一個配置，請複製隨附於 MFT 的 PDSE。

在 z/OS 上更新現有 MFT 代理程式或日誌程式指令資料集

您可以更新從 Managed File Transfer 指令範本資料集建立的 Managed File Transfer 指令 PDSE 程式庫資料集。

程序

1. 編輯 BFGCUSTM JCL Script 成員，並更新 BFGSTDIN DD 陳述式中的變數及內容。

如果您要移除先前定義的內容，請將其值設為空值，而非移除項目。當 BFGCUSTM JCL Script 執行時，指定的內容會以更新形式套用至實際的代理程式及日誌程式「UNIX 系統服務」內容檔；將內容設為空值，表示要移除該內容

2. 提交工作 BFGCUSTM。此工作會重新產生適用於代理程式或日誌程式的 JCL 指令集。如需指令的完整清單，請參閱第 584 頁的『z/OS 代理程式及日誌程式指令 JCL Script』。請檢查輸出工作日誌，以確認 JCL Script 已順利執行。如果執行失敗，請予以更正，然後重新提交 BFGCUSTM 工作。

結果

您可以修改產生的 JCL Script，並新增您自己的邏輯。不過，在重新執行 BFGCUSTM 時請多加留意，因為您可能會改寫自訂邏輯。

相關概念

第 566 頁的『z/OS 上的 MFT 配置選項』

z/OS 上的 Managed File Transfer 配置選項與分散式平台的選項相同。

相關工作

第 570 頁的『建立 MFT 代理程式或日誌程式指令資料集』

您可以從 Managed File Transfer 指令範本資料集，為特定的 Managed File Transfer Agent 或 Managed File Transfer Logger 建立指令的 PDSE 資料集，以進行特定協調。

z/OS z/OS JCL 變數

您可以在 BFGCUSTM Script 中使用替代值、JCL 變數及配置內容。

下表列出 MFT 指令 PDSE 程式庫資料集中的 BFGCUSTM JCL Script 所適用的替代值。在提交 BFGCUSTM 工作之前，您必須將這些替代值取代為適當的值。

替代變數	值
++library++	包含 MFT 指令 PDSE 程式庫的資料集名稱。
++bfg_java_home++	Java 安裝的位置。
++bfg_prod++	MFT 產品安裝「UNIX 系統服務」根目錄的位置。

下表說明 MFT 指令 PDSE 程式庫資料集中，BFGCUSTM JCL Script 的 BFGSTDIN DD 陳述式所適用的環境變數（位於 [Variables] 區段中）。在提交 BFGCUSTM 工作之前，您必須將所有以替代值（亦即，以兩個加號 ++ 括住的值）指定的變數取代為適當的值。

環境變數	值
LIBRARY	包含 MFT 指令 PDSE 程式庫的資料集名稱。
TMPDIR	暫存檔的「UNIX 系統服務」目錄。
BFG_PROD	MFT 產品安裝「UNIX 系統服務」根目錄的位置。
BFG_DATA	Managed File Transfer for z/OS 的資料目錄位置，此為 DATA_DIR 的路徑。
BFG_JAVA_HOME	Java 安裝的位置。
BFG_JVM_PROPERTIES	選用項目。設定 BFG_JVM_PROPERTIES 環境變數的值。這些內容會傳遞至 Java 虛擬機器。

表 45: 環境變數 (繼續)

環境變數	值
BFG_GROUP_NAME	<p>mqm 檔案群組通常會與 MFT 配置資料檔案和指令相關聯。因此，mqm 群組的所有使用者成員皆可存取及變更 MFT 配置。如需相關資訊，請參閱 IBM MQ 中 MFT 的檔案系統許可權。</p> <p>對於 z/OS 系統，檔案群組是 USS 檔案系統實體，因此不一定需要定義 mqm 檔案群組。您可以使用 BFG_GROUP_NAME 環境變數，將 z/OS USS 檔案系統群組與 MFT 配置資料檔案產生關聯。例如，在 USS Shell 提示中，使用：</p> <pre>export BFG_GROUP_NAME=FTEGB</pre> <p>可定義要與後續針對現行 USS 階段作業建立的配置檔相關聯的群組 <i>FTEGB</i>。</p> <p>您可以將 BFG_GROUP_NAME 設為空白值，或將它移除。</p> <p>註: 初次執行 BFGCUSTM 時，若 MFT 配置要供多個使用者 ID 使用，請務必將 BFG_GROUP_NAME 群組設定為所有需要的使用者 ID 皆可存取。第二次執行 BFGCUSTM 時，不得變更 BFG_GROUP_NAME (否則，必須同時變更目錄中 BFG_DATA 參照之所有檔案及目錄的 USS 群組檔案權限，以反映新的 BFG_GROUP_NAME 設定)。</p> <p>如果您在 BFG_GROUP_NAME 環境變數設為非空白值的 z/OS 系統上執行 fteMigrateAgent 指令，則該指令會檢查使用者是否為 BFG_GROUP_NAME 變數所命名群組的成員。如果使用者不在具名群組中，指令可能會報告錯誤訊息 BFGCL0502E: 您未獲授權執行所要求的作業。無法執行。如需使用者必須符合才能順利執行該指令之準則的詳細資料，請參閱 fteMigrateAgent。</p>
BFG_WTO	當 BFG_WTO 設為 YES、ON 或 TRUE 時，會啟用 z/OS 記載。這可控制寫入代理程式事件日誌的訊息是否也會寫入 z/OS 操作員日誌功能，它可讓您在從 JCL 執行代理程式時更容易存取自動化產品。遞送程式碼是 Programmer Information (11)，描述子碼是 Informational (12)。
SERVICE_TYPE	指定 MFT 指令程式庫是用於代理程式還是日誌程式。有效值為 AGENT 或 LOGGER。
名稱	SERVICE_TYPE 值的代理程式或日誌程式的名稱。
QMGR	與 SERVICE_TYPE 值的代理程式或日誌程式相關聯的本端佇列管理程式名稱。
OUTPUT_CLASS	SYSOUT 資料集的輸出類別。預設為 *，以要求與工作陳述式中的 MSGCLASS 參數相同的輸出類別。
MQ_PATH	用於 BFGPROF 中建立 LIBPATH 環境變數。
MQ_HLQ	IBM MQ 資料集的高階限定元。
MQ_LANG	必要的語言。

表 45: 環境變數 (繼續)	
環境變數	值
DB2_HLQ	選用項目。Db2 資料集的高階限定元。
JOBCARD1	JCL 指令工作的標頭行 1。
JOBCARD2	JCL 指令工作的標頭行 2。
JOBCARD3	JCL 指令工作的標頭行 3。
ADMIN_JOB1	管理工作的標頭行 1。
ADMIN_JOB2	管理工作的標頭行 2。
ADMIN_JOB3	管理工作的標頭行 3。
FTE_CONFIG	用於移轉的現有 WMQFTE 配置。如果不需要移轉，請設為空白值。
CREDENTIAL_PATH	用於移轉的認證檔的路徑，例如 /u/user1/agent3。Managed File Transfer 移轉的認證檔必須位於 IBM WebSphere MQ File Transfer Edition 7.0.4.4 上配置資訊及配置檔的個別檔案中。只有移轉指令 BFGAGMG 和 BFGLGMG JCL Script 才需要此項目。如果不需要移轉，請設為空白值。
DB_PROPS_PATH	指定移轉的資料庫日誌程式內容檔。只有在內容檔未使用下列預設名稱及路徑時，才需使用此選項： <code>config_directory/coordination_qmgr/databaselogger.properties</code> 。如果不需要移轉，請設為空白值。

註: MFT 隨附的 IBM MQ Jar 檔位於目錄 `MQMFT product root/java/lib` 中，一律使用且不可配置。

下表說明 MFT 指令 PDSE 程式庫資料集中，BFGCUSTM JCL Script 之 BFGSTDIN DD 陳述式的必要 MFT 配置內容。在提交 BFGCUSTM 工作之前，您必須將以替代值（亦即，以兩個加號 ++ 括住的值）指定的內容取代為適當的非空白值。這些內容會定義 MFT 配置內容的置換。您可以新增代理程式或日誌程式內容，為您的環境自訂代理程式或日誌程式。如需所有內容的清單，請參閱第 590 頁的『配置內容檔』。

表 46: BFGSTDIN DD 陳述式的必要配置內容	
內容	值
coordinationQMGr	與代理程式或日誌程式相關聯配置的協調佇列管理程式名稱。
coordinationQMGrHost	選用項目。協調佇列管理程式執行所在系統的主機名稱。如果將此內容值保留為空白，則會採用連結模式連線。
coordinationQMGrPort	選用項目。協調佇列管理程式用來接聽的埠號。只有在同時為 coordinationQMGrHost 內容指定非空白值時，才使用此參數。
coordinationQMGrChannel	選用項目。用來連接至協調佇列管理程式的通道。只有在同時為 coordinationQMGrHost 內容指定非空白值時，才使用此參數。
connectionQMGr	與代理程式或日誌程式相關聯配置的指令佇列管理程式名稱。

表 46: BFGSTDIN DD 陳述式的必要配置內容 (繼續)

內容	值
connectionQMGrHost	選用項目。執行指令佇列管理程式之系統的主機名稱。如果將此內容值保留為空白，則會採用連結模式連線。
connectionQMGrPort	選用項目。指令佇列管理程式接聽的埠號。只有在同時為 connectionQMGrHost 內容指定非空白值時，才使用此參數。
connectionQMGrChannel	選用項目。用來連接至指令佇列管理程式的通道。只有在同時為 connectionQMGrHost 內容指定非空白值時，才使用此參數。

z/OS z/OS 代理程式及日誌程式指令 JCL Script

MFT 指令 PDSE 程式庫資料集中的可用 JCL 指令集。

表 47: MFT 指令 PDSE 程式庫資料集中可用的 JCL 指令

成員	fte 指令行指令的說明
BFGCOPY	為此程式庫建立副本的工作
BFGCUSTM	為代理程式或日誌程式自訂此程式庫的工作
BFGCFCR	fteSetupCoordination
BFGCMCR	fteSetup 指令: 建立 MFT command.properties 檔案
BFGAGCR	fteCreateAgent 。只有在 SERVICE_TYPE 變數設為 AGENT 時才會建立。
BFGLGCRS	fteCreateLogger 。只有在 SERVICE_TYPE 變數設為 LOGGER 時才會建立。
BFGAGST	fteStartAgent 。只有在 SERVICE_TYPE 變數設為 AGENT 時才會建立。
BFGAGSTP	fteStartAgent 程序。只有在 SERVICE_TYPE 變數設為 AGENT 時才會建立。
BFGAGPI	ftePingAgent 。只有在 SERVICE_TYPE 變數設為 AGENT 時才會建立。
BFGAGSP	fteStopAgent 。只有在 SERVICE_TYPE 變數設為 AGENT 時才會建立。
BFGLGST	fteStartLogger 。只有在 SERVICE_TYPE 變數設為 LOGGER 時才會建立。
BFGLGSTP	fteStartLogger 程序。只有在 SERVICE_TYPE 變數設為 LOGGER 時才會建立。
BFGLGSP	fteStopLogger 。只有在 SERVICE_TYPE 變數設為 LOGGER 時才會建立。
BFGAGSH	fteShowAgentDetails 。只有在 SERVICE_TYPE 變數設為 AGENT 時才會建立。
BFGLGSH	fteShowLoggerDetails 。只有在 SERVICE_TYPE 變數設為 LOGGER 時才會建立。

表 47: MFT 指令 PDSE 程式庫資料集中可用的 JCL 指令 (繼續)

成員	fte 指令行指令的說明
BFGCFDF	fteChangeDefaultConfigurationOptions
BFGAGCL	fteCleanAgent 。只有在 SERVICE_TYPE 變數設為 AGENT 時才會建立。
BFGAGDE	fteDeleteAgent 。只有在 SERVICE_TYPE 變數設為 AGENT 時才會建立。
BFGLGDE	fteDeleteLogger 。只有在 SERVICE_TYPE 變數設為 LOGGER 時才會建立。
BFGPRSH	fteDisplayVersion
BFGAGLI	fteListAgents 。只有在 SERVICE_TYPE 變數設為 AGENT 時才會建立。
BFGMNL	fteListMonitors
BFGSTLI	fteListScheduledTransfers
BFGTMLI	fteListTemplates
BFGAGMG	fteMigrateAgent 。只有在 SERVICE_TYPE 變數設為 AGENT 時才會建立。
BFGLGMG	fteMigrateLogger 。只有在 SERVICE_TYPE 變數設為 LOGGER 時才會建立。
BFGCROBS	fteObfuscate 範例
BFGZRAS	fteRAS
BFGAGTC	fteSetAgentTraceLevel 。只有在 SERVICE_TYPE 變數設為 AGENT 時才會建立。
BFGLGTC	fteSetLoggerTrace 層次。只有在 SERVICE_TYPE 變數設為 LOGGER 時才會建立。
BFGPRANS	fteAnt 範例
BFGTRCAS	fteCancelTransfer 範例
BFGMNCRS	fteCreateMonitor 範例
BFGTMCRS	fteCreateTemplate 範例
BFGTRCRS	fteCreateTransfer 範例
BFGMNDES	fteDeleteMonitor 範例
BFGSTDES	fteDeleteScheduledTransfer 範例
BFGTMDES	fteDeleteTemplates 範例

附註:

- 對於建立 MQSC 或參照刪除 Script 的指令，JCL 將要求您執行一個 Script，但該 Script 已由工作執行。
- 執行 BGCUSTOM 工作時，BFGZRAS 會建立 BFGRAS 成員。

z/OS 執行驗證傳送

如何執行傳送以確認產品可正常運作。

重新命名並編輯成員 BFGTRCRS。

1. 在 %BFGCMD CMD=fteCreateTransfer -h 前面加上 /*
2. 移除成員中的其他註解。
3. 為 -sa 和 -da 指定現行代理程式名稱
4. 儲存 JCL
5. 提交 JCL

此 JCL 會連接到指令佇列管理程式。

配置記載作業

記載作業需要在與協調佇列管理程式相同的映像上執行。您可以記載到 Db2。

建立記載作業

複製 PDSE 以建立日誌程式專用的 PDSE。例如，user.MFT.LOGGER。

如果您需要使用不同的認證檔，請加以建立。如需相關資訊，請參閱 [在 z/OS 上配置 MQMFTCredentials.xml](#)。

檢閱成員 [BFGCUSTM](#)。請注意，有許多內容仍與先前的自訂相同。

但是，您必須：

- 變更 //SYSEXEC DD DSN=SCEN.FTE.JCL....
- 變更 LIBRARY 以符合代理程式 PDSE
- 將 QMGR 變更為協調佇列管理程式的名稱
- 使 SERVICE_TYPE=LOGGER
- 將 NAME 變更為日誌程式的名稱（符合 PDSE）
- 檢查 JOBCARD 並變更工作名稱，使該名稱與代理程式的工作名稱不同。
- 檢閱 BFG_JVM_properties = "-Xmx1024M"

如果您使用 Db2 日誌程式，建議您建立一個檔案，以便擷取 Db2 追蹤資料，來識別 Db2 問題。

檔案的名稱指定在 JVM 內容中，其中，JDBC 追蹤內容檔會有如下的內容

```
db2.jcc.traceDirectory=/u/johndoe/fte
db2.jcc.traceFile=jccTrace1
db2.jcc.traceFileAppend=false
# turn on all traces
# db2.jcc.traceLevel=-1
# turn off all traces
db2.jcc.traceLevel=0
```

設定兩個 JVM 內容

```
BFG_JVM_PROPERTIES=-Ddb2.jcc.propertiesFile=/u/.../sql.properties
-Ddb2.jcc.ssid=DBC
```

其中，/u/.../sql.properties 是您 Db2 追蹤內容檔的名稱，DBC 則是您 Db2 子系統的名稱。

提交此工作；請注意，此工作需要資料集的互斥存取權。代理程式的工作全都會有 BFGLG* 之類的名稱。

記載到檔案

如需如何記載至 Db2 的相關資訊，請參閱第 587 頁的『[記載到 Db2 時建立記載作業](#)』

重新命名成員 BFGLGCRS。此工作會更新 Managed File Transfer (MFT) 目錄中的檔案，並使用 CSQUTIL 在本端佇列管理程式中建立代理程式專用佇列。

原始檔案具有會列出指令語法的 %BFGCMD CMD=fteCreateLogger -h 指令。

若要建立日誌程式作業，請在陳述式前面放置 /* 以註銷 %BFGCMD CMD=fteCreateLogger -h，確保第一個直欄是空白的。

從第二個指令中移除註解，並配置陳述式。例如：

```
%BFGCMD CMD=fteCreateLogger +
-p MQPH +
-loggerQMgr MQPH +
-loggerType FILE +
-fileLoggerMode circular +
-fileSize 5MB +
-fileCount 5 +
-p MQPH +
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>'
LOGGER
```

檢查輸出，以確認是否已順利處理。

提示：將 `logger.properties` 檔案的路徑名稱從工作的輸出複製到代理程式在 PDSE 中的成員。

例如，複製到成員 APATH 中

```
/u/user_ID/fte/wmqmft/mqft/config/MQPH/loggers/LOGGER/logger.properties
```

如果您需要顯示內容檔，這將會有幫助。

將目錄新增至此檔案：

```
/u/user_ID/fte/wmqmft/mqft/logs/MQPH/loggers/LOGGER/
```

如果您記載到此檔案，日誌檔將會儲存在此目錄中，例如 `LOGGER0-20140522123654897.log`。

追蹤檔案會位於日誌子目錄中，例如

```
/u/user_ID/fte/wmqmft/mqft/logs/MQPH/loggers/LOGGER/logs
```

現在，您可以開始記載作業。

記載到 Db2 時建立記載作業

重新命名成員 BFGLGCRS。

此工作會更新 MFT 目錄中的檔案，並使用 CSQUTIL 在本端佇列管理程式中建立代理程式專用佇列。

您必須知道：

Db2 名稱	範例
-dbName databaseName	您可以從 Db2 子系統的訊息 DSNL004I 中的位置值取得此訊息
-dbDriver filePath	例如 /db2/db2v10/jdbc/classes/db2jcc.jar
-dbLib filePath	例如 /db2/db2v10/jdbc/lib/libdb2jcct2zos_64.so

編輯檔案。原始檔案具有會列出指令語法的 %BFGCMD CMD=fteCreateLogger -h 指令。

從第二個指令中移除註解，並配置陳述式。例如

```
%BFGCMD CMD=fteCreateLogger +
-p MQPH +
-loggerQMgr MQPH +
-loggerType DATABASE +
-dbType DB2 +
-databaseName DSNDBCP +
```

```
-dbDriver /db2/db2v10/jdbc/classes/db2jcc.jar +
-dbLib /db2/db2v10/jdbc/lib/ +
-credentialsFile //'<MFTCredentialsDataSet(MemberName)>' +
LOGGER
```

若要建立日誌程式作業，請在陳述式前面放置 /* 以註銷 %BFGCMD CMD=fteCreateLogger -h，確保第一個直欄是空白的。

提交工作並檢查輸出，以確認是否已順利處理。

提示: 將 logger.properties 檔案的路徑名稱從工作的輸出複製到代理程式在 PDSE 中的成員。

例如，複製到成員 APATH 中：

```
/u/user_ID/fte/wmqmft/mqft/config/MQPH/loggers/LOGGER/logger.properties into member USS
```

如果您需要顯示內容檔，這將會有幫助

追蹤檔案會位於日誌子目錄中，例如：

```
/u/user_ID/fte/wmqmft/mqft/logs/MQPH/loggers/LOGGER/logs
```

建立 Db2 表格

您必須建立 Db2 表格。這些定義是在 USS 檔案 mqft/sql/ftelog_tables_zos.sql 中。

在您的 PDSE 中建立成員 Db2。編輯此成員，並在指令行上使用 COPY 指令。從 USS 定義檔中複製。

由於網站特定需求可能有很大不同，這個檔案只會指定表格的基本結構，以及它們將位於的表格空間。

SQL Script 會指定表格空間，以確保使用具有足夠分頁大小的緩衝池來建立表格空間，以保留可能的最大表格列。請注意，此時並不會指定 LOB 位置之類的屬性。

您的資料庫管理者可能會想修改此檔案的副本，以定義與效能相關的屬性。

此檔案也會採用 FTELOG 的預設綱目名稱、FTELOGTS 的預設表格空間名稱，以及 FTELOGDB 的資料庫名稱。如有需要，您可以變更這些名稱，以符合現有的資料庫和任何本端命名慣例；請遵循檔案開頭處的註解中說明的程序。

重要: 請使用 **SPUFI** 之類的線上工具執行指令，因為檔案中有註解，而 **DSNTINAD** 之類的批次程式並不接受註解。

開始日誌程式作業

重新命名、檢查並提交成員 BFGLGST。您應會收到訊息 BFGDB0023I：日誌程式已完成啟動活動，目前正在執行中。

日誌程式作業

若要顯示日誌程式狀態，請重新命名、檢查並提交成員 BFGLGSH

若要停止日誌程式，請重新命名、檢查並提交成員 BFGLGSP。

z/OS 上 MFT 的環境變數

如果您要直接從 USS 環境執行指令，或是執行您自己的 JCL Script，在完成自訂及配置之後，您必須在執行 Managed File Transfer 所提供的配置及管理 Script 之前，設定一些環境變數。您必須為每一個使用者以及在要呼叫 Script 的每一個環境中設定這些變數。

為避免與其他產品發生衝突，您可以選擇在起始目錄中建立 .wmqfterc Script。然後，每一個 Managed File Transfer Script 都會呼叫 .wmqfterc Script，您可以使用此 Script 來提供 Managed File Transfer 的自訂環境設定。

您還可以設定另外一個選用的環境變數 BFG_WTO，這樣當您從 JCL 執行代理程式時可將訊息傳送至操作員日誌。

表 49: 必要的 z/OS 環境變數	
環境變數	值
BFG_JAVA_HOME	Java 安裝的位置。如需受支援 Java 層次的相關資訊，請參閱 IBM MQ 的系統需求 。
BFG_DATA	Managed File Transfer for z/OS 的資料目錄位置。這是 DATA_DIR 的路徑。
STEPLIB	<p>必須包括下列 IBM MQ 資料集：</p> <ul style="list-style-type: none"> • SCSQAUTH • SCSQANLE • SCSQLOAD <p>如果您想要在 z/OS 系統上執行資料庫日誌程式元件，STEPLIB 必須同時依顯示的順序包括下列 Db2 資料集：</p> <ul style="list-style-type: none"> • SDSNEXIT • SDSNLOD2 • SDSNLOAD
LIBPATH	必須包括「z/OS UNIX 系統服務」空間中 IBM MQJava 程式庫的位置 (對於 IBM MQ 8.0，預設值為 /mqm/V8R0M0/java/lib)。

下列是 .profile 範例，它正確配置 Managed File Transfer 的環境變數：

```
LIBPATH=/mqm/V8R0M0/java/lib:$LIBPATH
STEPLIB=MQM.V800.SCSQAUTH:MQM.V800.SCSQANLE:MQM.V800.SCSQLOAD
PATH=/u/fteuser/bin:/u/fteuser/J7.0/bin:/bin:/usr/bin:/u/fteuser/extras/bin:/bin:$PATH
BFG_JAVA_HOME=/u/fteuser/J7.0
BFG_DATA=/u/fteuser/DATA_DIR
export PATH LIBPATH STEPLIB BFG_JAVA_HOME BFG_DATA
```

您也可以選擇性地設定下列環境變數：

表 50: 選用的 z/OS 環境變數

環境變數	值
BFG_WTO	<p>下列其中一個值將啟用 BFG_WTO:</p> <ul style="list-style-type: none"> • YES • 開啟 • TRUE <p>下列其中一值將停用 BFG_WTO。這些值不區分大小寫。</p> <ul style="list-style-type: none"> • NULL • NO • OFF • FALSE <p>啟用 z/OS 記載。依預設，會停用此環境變數。</p> <p>寫入代理程式事件日誌的訊息也會寫入 z/OS 操作員日誌機能，它可讓您在從 JCL 執行代理程式時更容易存取自動化產品。遞送程式碼是 Programmer Information (11)，描述子碼是 Informational (12)。</p>
BFG_GROUP_NAME	<p>mqm 檔案群組通常會與 Managed File Transfer 配置資料檔案和指令相關聯。因此，mqm 群組成員的所有使用者皆可存取及變更 Managed File Transfer 配置。如需相關資訊，請參閱 IBM MQ 中 MFT 的檔案系統許可權。</p> <p>對於 z/OS 系統，檔案群組是 USS 檔案系統實體，因此不一定需要定義 mqm 檔案群組。您可以使用 BFG_GROUP_NAME 環境變數，為 Managed File Transfer 配置資料檔案定義替代的現有 z/OS USS 檔案系統群組。例如，在 USS Shell 提示中：</p> <pre style="background-color: #f0f0f0; padding: 5px;">export BFG_GROUP_NAME=FTEGB</pre> <p>可定義要與後續針對現行 USS 階段作業建立的配置檔相關聯的群組 FTEGB。</p> <p>您可以將 BFG_GROUP_NAME 設為空白值，或將它移除。</p> <p>如果您在 BFG_GROUP_NAME 環境變數設為非空白值的 z/OS 系統上執行 fteMigrateAgent 指令，則該指令會檢查使用者是否為 BFG_GROUP_NAME 變數所命名群組的成員。如果使用者不在具名群組中，指令可能會報告錯誤訊息 BFGCL0502E：您未獲授權執行所要求的作業。無法執行。如需使用者必須符合才能順利執行該指令之準則的詳細資料，請參閱 fteMigrateAgent。</p>

▶ z/OS 配置內容檔

Managed File Transfer 中所使用內容的摘要。

- [MFT coordination.properties 檔案](#)

- [MFT command.properties 檔案](#)
- [MFT agent.properties 檔](#)
- [日誌程式配置內容檔](#)

為 z/OS Automatic Restart Manager (ARM) 配置 MFT

Managed File Transfer 是已啟用 ARM 的應用程式。

開始之前

如需為您的系統啟用 ARM 以及定義 ARM 原則的相關資訊，請參閱[使用 z/OS Automatic Restart Manager \(ARM\)](#)。

如果想要使用 MFT DB 日誌程式功能自動重新啟動並重新連接至 Db2 資料庫，ARM 是唯一可用的受支援重新啟動管理程式。

關於這項作業

使用 ARM，可以透過設定代理程式/日誌程式內容 armELEMTYPE 及 armELEMENT，將代理程式及日誌程式配置為重新啟動。內容 armELEMTYPE 定義 ARM 元素的類型，內容 armELEMENT 則是 ARM 要登錄的元素名稱：

- 您可以將代理程式 ELEMTYPE 設為 SYSBFGAG，而 armELEMENT 可以設為對應於代理程式名稱。
- 您可以將日誌程式 ELEMTYPE 設為 SYSBFGLG，而 armELEMENT 可以設為對應於日誌程式名稱。

註：配置為由 ARM 重新啟動的代理程式及日誌程式，只能透過批次工作或已啟動的作業順利執行。嘗試從 USS 指令行直接啟動代理程式或日誌程式將會失敗，並產生 ARM 錯誤原因碼。

範例

下列重新啟動原則的範例將代理程式 BFGFT7CAG1 定義為與佇列管理程式 FT7C 相依：

```
RESTART_ORDER
  LEVEL(3)
  ELEMENT_TYPE(SYSBFGAG,SYSBFGLG)

RESTART_GROUP(GROUP7C)
  ELEMENT(SYSMQMGRFT7C)
  ELEMENT(BFGFT7CAG1)
  RESTART_ATTEMPTS(3,300)
```

範例: 在 z/OS 上為 Managed File Transfer 代理程式建立 JCL

使用此資訊來產生一些 JCL，可用來在 IBM MQ for z/OS 上建立及啟動代理程式。

複製範例程式庫

執行下列程序：

1. 透過開啟程式庫來建立程式庫 SBFGCMD5 的副本 (請參閱 [第 576 頁的『複製 SBFGCMD5 以建立 JCL 程式庫』](#))。
 - 大部分成員 (以 BFGX、BFGY 或 BFGZ 開頭的成員) 都是您稍後用來為代理程式產生自訂 JCL 的範本。
 - 重要成員是 BFGCOPY。
2. 開啟 BFGCOPY 並取代：
 - ++ supplied_library ++**
作為產品一部分安裝的 SBFGCMD5 程式庫名稱。
 - ++ 服務程式庫 ++**
您要用於代理程式的程式庫名稱 (目標程式庫)。

3. 提交工作，並且您具有可以使用的新檔案庫。

編輯 BFGCUSTM

執行下列程序：

1. 開啟新的程式庫，以便您可以編輯 BFGCUSTM 成員 (請參閱 [第 577 頁的『編輯成員 BFGCUSTM』](#))
2. 修改成員中以 ++ 字元括住的所有參數，並將它們取代為適當的值。例如，將：

++bfg_prod++

指向已安裝 IBM MQ Managed File Transfer for z/OS 的 USS 目錄。

++ bfg_data ++

作為儲存 IBM MQ Managed File Transfer for z/OS 配置的 USS 目錄。

++ service_type ++

至單字 AGENT

++ agent_name ++

成為代理程式的名稱

附註：

1. 部分項目 (例如 CLEAN_AGENT_PROPS 所需的 ++options++) 是不需要的，因此您應該移除這些項目。
2. 如需 BFGCUSTM 成員中所有參數的完整清單，以及它們應該具有哪些值的說明，請參閱 [第 572 頁的『開始之前』](#)。

提交 BFGCUSTM JCL

執行下列程序：

1. 提交工作。
2. 在 ISPF 中結束程式庫。

這是必要的，因為 BFGCUSTM 工作正在更新程式庫，且在程式庫開啟時無法執行此動作。

3. 當工作完成時，請查看工作日誌。

您將會看到一些訊息，指出已在檔案庫中建立新成員。

其中每一個成員都包含 JCL，可用來執行代理程式的特定作業。如需這些成員及其對應之 IBM MQ Managed File Transfer 指令的清單，請參閱 [第 584 頁的『z/OS 代理程式及日誌程式指令 JCL Script』](#)。

提交 BFGAGCR 以建立代理程式

新成員 BFGAGCR 包含一些 JCL，可透過呼叫 **fteCreateAgent** 指令來 [建立代理程式](#)。

執行下列程序：

1. 開啟成員 BFGAGCR。

您應該會看到 BFGAGCR 已移入您的名稱：

- 代理程式
- 代理程式佇列管理程式
- MFT 拓撲的協調佇列管理程式

2. 提交成員 BFGAGCR。

當成員執行時，它會：

- 為代理程式建立必要的配置檔。
- 連接至代理程式佇列管理程式，並使用 CSQUTIL 建立代理程式所需的系統佇列。
- 向協調佇列管理程式登錄代理程式。

提交 BFGAGST 以啟動代理程式

執行下列程序：

1. 提交 BFGAGST 成員。如需顯示代理程式正常運作的各種指令，請參閱 [使用代理程式](#)。
2. 當工作完成時，請檢查工作日誌是否包含下列訊息：

```
BFGAG0058I: The agent has successfully initialized.  
BFGAG0059I: The agent has been successfully started.
```

這表示您的代理程式已啟動、執行中，並準備好執行受管理傳送。

將 MFT 代理程式移至新的 z/OS LPAR

有時需要將 IBM MQ Managed File Transfer for z/OS 代理程式從一個 LPAR 移至另一個，同時將代理程式保留在具有相同協調及指令佇列管理程式的相同 IBM MQ Managed File Transfer 拓撲中。執行此動作所需的步驟取決於最初建立所移轉代理程式的方式。

關於這項作業

以下列其中一種方式移動 IBM MQ Managed File Transfer for z/OS 代理程式：

- 如果最初使用 SBFGCMD5 程式庫的自訂版本建立代理程式，請使用程式庫在新的 LPAR 上重建它。
- 如果最初是透過執行 USS 指令來建立代理程式，請使用指令在新的 LPAR 上重建代理程式。

註：

排定的傳送及傳送範本儲存在 IBM MQ Managed File Transfer 拓撲的協調佇列管理程式上。此作業假設協調佇列管理程式不是移動工作的一部分。在此情況下，在移動完成之後，與正在移動的代理程式相關聯的任何排定傳送及傳送範本都會保留在現有協調佇列管理程式上。

程序

- 移動使用 SBFGCMD5 程式庫自訂版本所建立的代理程式。

如果代理程式是使用自訂版本的 SBFGCMD5 程式庫來建立，您可以使用該程式庫來重建 IBM MQ Managed File Transfer for z/OS 環境，以及在新 LPAR 上的代理程式配置。若要執行此作業，請完成下列步驟：

1. 將程式庫的自訂版本從原始 LPAR 複製到新的 LPAR。
2. 在新 LPAR 上的程式庫自訂版本中編輯 BFGCUSTM 成員，並確定參數值仍然有效。
3. 在新的 LPAR 上執行 BFGCUSTM 成員，以建立配置環境及建立代理程式所需的所有 JCL。
4. 執行 BFGCFR 成員以定義新 LPAR 上代理程式要使用的協調佇列管理程式，並建立儲存 IBM MQ Managed File Transfer 配置所需的目錄結構。
5. 接下來，執行 BFGCMCR 成員，以定義要由新 LPAR 上代理程式使用的指令佇列管理程式。
6. 執行 BFGAGCR 成員以重建代理程式及其配置。
7. 請確定代理程式所使用的系統佇列存在於該代理程式的佇列管理程式上。

如果要移動的代理程式具有與其相關聯的資源監視器，則您需要在新的代理程式上重建監視器。若要執行此作業，請完成下列步驟：

1. 在原始 LPAR 上，執行 BFGMNL1 成員，將與原始代理程式相關聯之資源監視器的定義匯出至 XML 檔。
 2. 將包含資源監視器定義的 XML 檔案複製到新的 LPAR。
 3. 使用新 LPAR 上 SBFGCMD5 程式庫中的 BFGMNCRS 成員，來匯入儲存在 XML 檔案中的資源監視器定義。這會導致在新的代理程式上建立監視器。
- 移動在 USS 中執行指令所建立的代理程式。

如果最初是透過執行 USS 指令來建立代理程式，您可以使用指令在新的 LPAR 上重建代理程式。若要執行此作業，請完成下列步驟：

1. 在新的 LPAR 上執行 `fteSetupCoordination` 指令，以定義代理程式要使用的協調佇列管理程式，並建立儲存 IBM MQ Managed File Transfer 配置所需的目錄結構。
2. 執行 `fteSetup` 指令，可定義新 LPAR 上代理程式要使用的指令佇列管理程式。
3. 執行 `fteCreateAgent` 指令，以重建代理程式及其配置。
4. 請確定代理程式所使用的系統佇列存在於該代理程式的佇列管理程式上。

如果要移動的代理程式具有與其相關聯的資源監視器，則您需要在新的代理程式上重建監視器。若要執行此作業，請完成下列步驟：

1. 在原始 LPAR 上，執行 `fteListMonitors` 指令並指定 `-ox` 參數，以將與原始代理程式相關聯之資源監視器的定義匯出至 XML 檔案。
2. 將包含資源監視器定義的 XML 檔案複製到新的 LPAR。
3. 在新的 LPAR 上執行 `fteCreateMonitor` 指令，並指定 `-ix` 參數，以匯入儲存在 XML 檔案中的資源監視器定義。這會導致在新的代理程式上建立監視器。

搭配使用 Managed File Transfer for z/OS 與 JZOS Java 啟動程式

在 IBM MQ for z/OS 系統上，您可以套用本主題中的指示，作為在企業中使用 Managed File Transfer 的替代方法。

概觀

Managed File Transfer for z/OS (MFT) 使用標準 z/OS 安裝程序。執行 MFT 指令的另一種方式是使用 JCL 及 JZOS Java Launcher。

如需進一步詳細資料，請參閱 [JZOS 批次啟動程式和工具箱](#)。

如果 JCL 無法正確處理，請參閱 [JZOS 的一般 MFT 問題](#)。

IBM MQ 8.0 以及更新版本的 JCL 範例



小心：若為 IBM WebSphere MQ File Transfer Edition 7.0，參數是以 `FTE_` 開頭，而不是以 `BFG_` 開頭。

```
//JOHNDOEA JOB 1,MSGCLASS=H
// JCLLIB ORDER=(SCEN.MFT.JCL)      (1)
// INCLUDE MEMBER=BFGJCL8           (2)
// DD * (2A)
. ${BFG_PROD}/bin/fteBatch createAgent (3)
export IBM_JAVA_OPTIONS="${BFG_JAVA_OPTIONS} ${BFG_LANG}" (4)
export JZOS_MAIN_ARGS="${BFG_MAIN_ARGS}" (4)
//MAINARGS DD *
-agentName MYAGENT (5)
-f
-agentQMgr MQPD
-p MQPD
/*
```

其中：

- (1) 是所包含的 JCL 陳述式的位置
- (2) 包括來自 1) 的位置的指定 JCL 成員
- (2A) 這會延伸 //STDENV - 請參閱下方說明
- (3) 這是要執行的指令，不含前導 `fte` 字首
- (4) 這幾行是必要項目，它們可設定 JZOS 的資訊
- (5) 指令的參數

- BFGJCL8 成員（您可以選取自己的名稱）會呼叫 JZOS。此成員有執行 MFT 所需的 STEPLIB 和其他 JCL。

您需要包含的其他 JCL

您應該包括 IBM MQ for z/OS 程式庫的 JCL，如果您使用的是 Db2 日誌程式，則還要包括 Db2 程式庫的 JCL。

例如：

```
//WMQFTE EXEC PGM=JVMLDM86,REGION=0M PARM='+T' (1)
//STEPLIB DD DSN=SYS1.SIEALNKE,DISP=SHR (2)
//* MQ libraries
// DD DSN=MQM.V800.SCSQAUTH,DISP=SHR MQ Bindings
// DD DSN=MQM.V800.SCSQANLE,DISP=SHR MQ Bindings
// DD DSN=MQM.V800.SCSQLOAD,DISP=SHR MQ Bindings

//* DB2 libraries
// DD DISP=SHR,DSN=SYS2.DB2.V10.SDSNEXIT.DBCP
// DD DISP=SHR,DSN=SYS2.DB2.V10.SDSNLOAD
// DD DISP=SHR,DSN=SYS2.DB2.V10.SDSNLOAD2
//SYSOUT DD SYSOUT=H
//SYSPRINT DD SYSOUT=H
//STDOUT DD SYSOUT=H
//STDERR DD SYSOUT=H

//STDENV DD DSN=SCEN.MFT.JCL(BFGZENV8),DISP=SHR (3)
```

其中：

- (1) 是 JZOS 程式的名稱。請查看 SYS1.SIEALNKE 以瞭解您系統上的版本。請新增 ,PARM='+T' 以提供其他診斷資訊。
- (2) 這是含有 JZOS 程式的資料集。
- (3) 這是 Shell Script 的成員名稱。它會定義 MFT 所需的參數。請參閱 [第 595 頁的『用於定義 MFT 的 Shell Script』](#)。

它可以是任何資料集及成員。它必須位於檔案末尾，因為 JCL 工作會加以延伸。請參閱 [第 594 頁的『IBM MQ 8.0 以及更新版本的 JCL 範例』](#) 中的 2A。

用於定義 MFT 的 Shell Script

在 [第 595 頁的『您需要包含的其他 JCL』](#) 範例中，使用了成員 BFGZENV8。這是基於 JZOS 設定檔。

您可以對 MFT 第 8 版和 IBM WebSphere MQ File Transfer Edition 7.0 使用相同的配置檔，只進行一些小小的變更。請記住，在 MFT 第 8 版之前，參數是以 FTE 開頭。請參閱 [第 595 頁的『範例檔』](#)。

您必須知道：

- 安裝 Java 的位置
- IBM MQ for z/OS Java 程式庫的位置
- MFT 檔案的位置
- 使用者 ID 需要位於特定群組內，才能視為 IBM MQ for z/OS 管理者。您需要此群組的名稱
- 如果您不要使用英文訊息，則需要知道要指定哪一種語言。

範例檔

```
# This is a shell script that configures
# any environment variables for the Java JVM.
# Variables must be exported to be seen by the launcher.
# Use PARM='+T' and set -x to debug environment script problems
set -x
# . /etc/profile
#
# Java configuration (including MQ Java interface)
#
```

```

export _BPXK_AUTOCVT="ON"
export JAVA_HOME="/java/java71_bit64_sr3_fp30/J7.1_64/"
export PATH="/bin:${JAVA_HOME}/bin/classic/"
LIBPATH="/lib:/usr/lib:${JAVA_HOME}/bin"
LIBPATH="$LIBPATH:${JAVA_HOME}/bin/classic"
LIBPATH=$LIBPATH:"/mqm/V8R0M0/java/lib/"
export LIBPATH

export BFG_JAVA_HOME="${JAVA_HOME}"
export BFG_WTO="YES"
export BFG_GROUP_NAME=MQADM
export BFG_PROD="/HMF8800/"
export BFG_CONFIG="/u/johndoe/fteconfig"
# export BFG_LANG=" -Duser.language=de "
export BFG_LANG=" "

```

其中：

export _BPXK_AUTOCVT="ON"

對於 Unicode 轉換是必要項目

export JAVA_HOME="/java/java71_bit64/J7.1_64/"

是 Java 目錄的位置。請指定 Java 的路徑名稱。此目錄包含 bin 和其他目錄。

export PATH="/bin:\${JAVA_HOME}/bin/classic/"

設定 Java 可執行陳述式的路徑陳述式

LIBPATH="/lib:/usr/lib:\${JAVA_HOME}/bin"

設定 Java 可執行陳述式的程式庫路徑

LIBPATH="\$LIBPATH:\${JAVA_HOME}/bin/classic"

在 LIBPATH 陳述式中新增更多 Java 程式庫。

LIBPATH=\$LIBPATH:"/mqm/V8R0M0/java/lib/"

在程式庫路徑中新增 IBM MQ for z/OS 程式庫。請指定 USS 中的 IBM MQ for z/OS 程式庫的名稱。

export LIBPATH

使 LIBPATH 可用於 JZOS

export BFG_JAVA_HOME="\${JAVA_HOME}"

將 BFG_JAVA_HOME 設為上方指定的 JAVA_HOME 的值

export BFG_WTO="YES"

將 BFG_WTO 設為 YES 會使用 WTO 將訊息顯示在工作日誌中

export BFG_GROUP_NAME=MQADM

屬於所指定群組之成員的使用者 ID，將視為 IBM MQ for z/OS 管理者

export BFG_PROD="/HMF8800/"

是 MFT 程式碼所在位置的路徑

export BFG_DATA="/u/johndoe/fteconfig"

是儲存 MFT 配置資訊的位置

export BFG_LANG=" -Duser.language=de "

是將語言定義為德文的已註銷陳述式

export BFG_LANG=" "

將語言指定為預設值（英文）。

/lib/messages/BFGNVMessages_*.properties 中的 MFT 產品內容列出可用的語言。預設值是將值保留為空白，表示將使用英文。

若為第 7 版，請指定：

```

export FTE_JAVA_HOME="${JAVA_HOME}"
export FTE_WTO="YES"
export FTE_GROUP_NAME=SCENU
export FTE_PROD=""/HMF7100/"
export FTE_CONFIG="/u/johndoe/fteconfig"
export BFG_LANG=" "

```

請注意， /u/johndoe/fteconfig 與 BFG_DATA 中的路徑不同

相關工作

第 571 頁的『正在配置 Managed File Transfer for z/OS』

Managed File Transfer for z/OS 需要經過自訂，元件才能正常運作。

[規劃 Managed File Transfer](#)

IBM i 配置 MFT on IBM i

在安裝 Managed File Transfer 之後，您必須完成協調佇列管理程式及代理程式的某些配置，才能開始使用它。

關於這項作業

安裝之後，您必須對新的協調佇列管理程式及新的代理程式執行 Managed File Transfer 所提供的配置 Script，才能使用協調佇列管理程式及代理程式來傳送檔案。然後，您必須啟動已建立的代理程式。

程序

- 對於所有新的協調佇列管理程式：針對協調佇列管理程式執行 `coordination_qmgr_name.mqsc` 檔案中的 MQSC 指令。如果協調佇列管理程式與安裝不在相同的電腦上，請將 MQSC Script 檔複製到佇列管理程式所在的電腦上，然後再執行 Script。
 - 從 IBM i 指令行使用下列指令啟動 qshell: CALL QSHELL
 - 切換至下列目錄： /QIBM/UserData/mqm/mqft/config/coordination_qmgr_name
 - 發出下列指令，將 `coordination_qmgr_name` 取代為佇列管理程式的名稱：

```
/QSYS.LIB/QMQM.LIB/RUNMQSC.PGM coordination_qmgr_name < coordination_qmgr_name.mqsc
```

您可以改用手動方式配置協調佇列管理程式。如需相關資訊，請參閱第 600 頁的『配置 MFT 的協調佇列管理程式』。

- 針對所有新的代理程式：對代理程式佇列管理程式執行 `agent_name_create.mqsc` 檔中的 MQSC 指令。

如果代理程式佇列管理程式與代理程式不在相同的電腦上，請將 MQSC Script 檔複製到佇列管理程式所在的電腦上，然後再執行 Script。

 - 從 IBM i 指令行使用下列指令啟動 qshell: CALL QSHELL
 - 切換至下列目錄： /QIBM/UserData/mqm/mqft/config/agent_qmgr_name/agents
 - 發出下列指令，將 `agent_qmgr_name` 取代為代理程式佇列管理程式的名稱，並將 `agent_name` 取代為代理程式的名稱：

```
/QSYS.LIB/QMQM.LIB/RUNMQSC.PGM agent_qmgr_name < agent_name_create.mqsc
```

您可以改用手動方式配置代理程式佇列管理程式。如需相關資訊，請參閱第 600 頁的『配置 MFT 代理程式佇列管理程式』。

- 如果您在安裝期間尚未啟動 QMFT 子系統，請從 IBM i 指令行使用下列指令來啟動 QMFT 子系統：
STRSBS SBSDB(QMQMMFT/QMFT) 或 STRSBS QMQMMFT/QMFT
- 使用 **fteStartAgent** 指令來啟動新的代理程式。
 - 從 IBM i 指令行使用下列指令啟動 qshell: CALL QSHELL
 - 切換至下列目錄： /QIBM/ProdData/mqm/bin
 - 發出下列指令，將 AGENT 取代為代理程式的名稱：

```
./fteStartAgent AGENT
```

下一步

建議您設定沙盤推演，以限定代理程式可存取的檔案系統區域。此特性在 [使用 MFT 代理程式沙盤推演](#) 中有說明。

相關概念

第 598 頁的『[初次使用時配置 MFT](#)』

初次使用 Managed File Transfer 代理程式及佇列管理程式時，必須先執行某些配置作業。

初次使用時配置 MFT

初次使用 Managed File Transfer 代理程式及佇列管理程式時，必須先執行某些配置作業。

連接至 IBM MQ

使用 IBM MQ 佇列管理程式的所有網路通訊（包括與 Managed File Transfer 相關的通訊）都涉及 IBM MQ 通道。IBM MQ 通道代表網路鏈結的一端。通道分類為訊息通道或 MQI 通道。

Managed File Transfer 及通道

Managed File Transfer 使用 MQI 通道在用戶端模式下將代理程式連接至其代理程式佇列管理程式，以及將指令應用程式（例如，**fteCreateTransfer**）連接至其指令及協調佇列管理程式。在預設配置中，使用稱為 SYSTEM.DEF.SVRCONN 的 SVRCONN 通道（依預設，存在於所有佇列管理程式上）進行這些連線。由於這些預設值，您無需變更基本 Managed File Transfer 安裝的任何 MQI 通道。

訊息通道端點有六種類型，但本主題只會說明傳送端至接收端通道配對。請參閱 [分散式佇列元件](#)，以取得其他通道組合的相關資訊。

所需的訊息路徑

IBM MQ 訊息只能透過訊息通道傳送，因此，您必須確保通道可用於 Managed File Transfer 所需的所有訊息路徑。這些路徑不必是直接的；必要時，可以透過中間佇列管理程式傳送訊息。此主題僅涵蓋直接點對點通訊。如需這些選項的相關資訊，請參閱 [如何進入遠端佇列管理程式](#)。

Managed File Transfer 使用的通訊路徑如下：

代理程式至代理程式

在之間傳送檔案的任何兩個代理程式，需要在其關聯的佇列管理程式之間進行雙向通訊。由於此路徑會傳送大量資料，因此，請根據您的需要考量儘可能讓路徑簡短、快速或便宜。

代理程式至協調

參與傳送的代理程式中的日誌訊息，必須能夠抵達協調佇列管理程式。

指令至代理程式

指令應用程式或「IBM MQ Explorer」（使用指令佇列管理程式）所連接的任何佇列管理程式，必須能夠將訊息傳送至使用這些指令應用程式控制的代理程式的佇列管理程式。若要啟用指令顯示的回饋訊息，請使用雙向連線。

如需相關資訊，請參閱適用於您企業所使用之一個以上平台的驗證 *IBM MQ* 安裝。

相關概念

第 604 頁的『[配置多重實例佇列管理程式以使用 MFT](#)』

IBM WebSphere MQ 7.0.1 及更新版本支援建立多重實例佇列管理程式。多重實例佇列管理程式可自動在待命伺服器上重新啟動。Managed File Transfer 支援與多重實例代理程式佇列管理程式、多重實例協調佇列管理程式，以及多重實例指令佇列管理程式的連線。

相關工作

第 599 頁的『[配置 MFT 網路佇列管理程式](#)』

如果您的 Managed File Transfer 網路包含多個 IBM MQ 佇列管理程式，這些 IBM MQ 佇列管理程式必須能夠在遠端彼此進行通訊。

第 600 頁的『[配置 MFT 的協調佇列管理程式](#)』

執行 **fteSetupCoordination** 指令之後，請執行 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` 目錄中的 `coordination_qmgr_name.mqsc` Script，以執行協調佇列管理程式所需的配置。但如果要手動執行此配置，請在協調佇列管理程式上執行下列步驟。

配置 MFT 網路佇列管理程式

如果您的 Managed File Transfer 網路包含多個 IBM MQ 佇列管理程式，這些 IBM MQ 佇列管理程式必須能夠在遠端彼此進行通訊。

關於這項作業

有以下兩種方式來將佇列管理程式配置為能夠彼此進行通訊：

- 藉由設定 IBM MQ 佇列管理程式叢集。

如需 IBM MQ 佇列管理程式叢集及如何配置它們的相關資訊，請參閱 [第 232 頁的『配置佇列管理程式叢集』](#)。

- 藉由設定佇列管理程式之間的通道（如下所述）：

設定佇列管理程式之間的通道

在佇列管理程式之間設定下列訊息通道：

- 從代理程式佇列管理程式至協調佇列管理程式
- 從指令佇列管理程式至代理程式佇列管理程式。
- 從代理程式佇列管理程式至指令佇列管理程式（使回饋訊息能夠以指令顯示）。
- 從指令佇列管理程式至協調佇列管理程式
- 從代理程式佇列管理程式至 Managed File Transfer 網路中的任何其他代理程式佇列管理程式

如果您需要如何設定此通訊的進一步相關資訊，請從以下資訊開始：[使用 MQSC 管理遠端 IBM MQ 物件](#)。

部分建議的範例步驟如下：

程序

1. 使用與協調佇列管理程式相同的名稱，在 IBM MQ 佇列管理程式上建立傳輸佇列。

您可以使用下列 MQSC 指令：

```
DEFINE QLOCAL(coordination-qmgr-name) USAGE(XMITQ)
```

2. 在 IBM MQ 佇列管理程式上，建立 Managed File Transfer 協調佇列管理程式的傳送端通道。在上一步中建立的傳輸佇列名稱是此通道所需的參數。如果需要與 Managed File Transfer for IBM WebSphere MQ 7.5 或 Managed File Transfer 代理程式通訊，請確定傳送端通道的 CONVERT 參數設為 no。（舊版 IBM WebSphere MQ File Transfer Edition 一律以 UTF-8 格式發佈訊息，這表示任何資料轉換都會毀損訊息。對於 Managed File Transfer for IBM MQ 8.0 或更新版本上的代理程式而言，這不是必要項目，因為以空白格式發佈訊息。）

您可以使用下列 MQSC 指令：

```
DEFINE CHANNEL(channel-name) CHLTYPE(SDR) CONNAME('coordination-qmgr-host(coordination-qmgr-port)')  
XMITQ(coordination-qmgr-name) CONVERT(NO)
```

註：只有在必要時設定 CONVERT(NO)。

3. 在 Managed File Transfer 協調佇列管理程式上，建立 IBM MQ 佇列管理程式的接收端通道。為此接收端通道指定與 IBM MQ 佇列管理程式的傳送端通道相同的名稱。

您可以使用下列 MQSC 指令：

```
DEFINE CHANNEL(channel-name) CHLTYPE(RCVR)
```

下一步

接下來，請遵循協調佇列管理程式的配置步驟：[第 600 頁的『配置 MFT 的協調佇列管理程式』](#)。

配置 MFT 的協調佇列管理程式

執行 `fteSetupCoordination` 指令之後，請執行 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` 目錄中的 `coordination_qmgr_name.mqsc` Script，以執行協調佇列管理程式所需的配置。但如果要手動執行此配置，請在協調佇列管理程式上執行下列步驟。

關於這項作業

程序

1. 建立稱為 SYSTEM.FTE 的本端佇列。
2. 將 SYSTEM.FTE 佇列新增至 SYSTEM.QPUBSUB.QUEUE.NAMELIST 名稱清單。
3. 使用 SYSTEM.FTE 的主題字串建立稱為 SYSTEM.FTE 的主題。
4. 確保 SYSTEM.FTE 主題的非持續性訊息遞送 (NPMGDLV) 及持續性訊息遞送 (PMSGDLV) 屬性設定為 ALLAVAIL。
5. 確保協調佇列管理程式的「發佈/訂閱」模式 (PSMODE) 屬性設定為 ENABLED。

下一步

如果您在已配置為協調佇列管理程式的佇列管理程式上執行 `strmqm -c` 指令，則該指令會刪除 [步驟 2](#) 中所做的變更 (將 SYSTEM.FTE 佇列新增至 SYSTEM.QPUBSUB.QUEUE.NAMELIST 名單)。這是因為 `strmqm -c` 會重建預設的 IBM MQ 物件，並反轉 Managed File Transfer 變更。因此，如果您已使用 `strmqm -c` 啟動佇列管理程式，請完成下列其中一個步驟：

- 再次於佇列管理程式上執行 `coordination_qmgr_name.mqsc` Script。
- 重複 [步驟 2](#)。

相關概念

[第 598 頁的『連接至 IBM MQ』](#)

使用 IBM MQ 佇列管理程式的所有網路通訊 (包括與 Managed File Transfer 相關的通訊) 都涉及 IBM MQ 通道。IBM MQ 通道代表網路鏈結的一端。通道分類為訊息通道或 MQI 通道。

[第 604 頁的『配置多重實例佇列管理程式以使用 MFT』](#)

IBM WebSphere MQ 7.0.1 及更新版本支援建立多重實例佇列管理程式。多重實例佇列管理程式可自動在待命伺服器上重新啟動。Managed File Transfer 支援與多重實例代理程式佇列管理程式、多重實例協調佇列管理程式，以及多重實例指令佇列管理程式的連線。

相關工作

[第 599 頁的『配置 MFT 網路佇列管理程式』](#)

如果您的 Managed File Transfer 網路包含多個 IBM MQ 佇列管理程式，這些 IBM MQ 佇列管理程式必須能夠在遠端彼此進行通訊。

相關參考

[fteSetupCoordination](#)

配置 MFT 代理程式佇列管理程式

安裝之後，請執行 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name` 目錄中的 `agent_name_create.mqsc` Script，為代理程式佇列管理程式執行所需的配置。但是，如果您要手動執行此配置，請在代理程式佇列管理程式上完成以下步驟。

程序

1. 建立代理程式作業佇列。
請命名以下佇列：

- SYSTEM.FTE.COMMAND.agent_name
- SYSTEM.FTE.DATA.agent_name
- SYSTEM.FTE.EVENT.agent_name
- SYSTEM.FTE.REPLY.agent_name
- SYSTEM.FTE.STATE.agent_name

如需佇列參數的相關資訊，請參閱 [MFT 代理程式佇列設定](#)。

2. 建立代理程式權限佇列。

請命名以下佇列：

- SYSTEM.FTE.AUTHADM1.agent_name
- SYSTEM.FTE.AUTHAGT1.agent_name
- SYSTEM.FTE.AUTHMON1.agent_name
- SYSTEM.FTE.AUTHOPS1.agent_name
- SYSTEM.FTE.AUTHSCH1.agent_name
- SYSTEM.FTE.AUTHTRN1.agent_name

如需佇列參數的相關資訊，請參閱 [MFT 代理程式佇列設定](#)。

下一步

如需建立及配置通訊協定橋接器代理程式的相關資訊，請參閱 [fteCreateBridgeAgent \(建立及配置 MFT 通訊協定橋接器代理程式\)](#) 及為 [FTPS 伺服器配置通訊協定橋接器](#)。

建立 IBM MQ File Transfer 結構

您可以透過同一部機器上，連線到佇列管理程式的單一代理程式配置 Managed File Transfer 結構。

關於這項作業

MFT 配置會儲存在代理程式所在機器之 IBM MQ DataPath 下的檔案結構中。

下列範例配置適用於名為 SAMPLECOORD (已停用安全) 的 MFT on IBM MQ 8.0 佇列管理程式，以及名為 SAMPLEAGENT 的單一 MFT 代理程式：

```
+--- config
      +--- SAMPLECOORD
            +--- command.properties
            +--- coordination.properties
            +--- SAMPLECOORD.mqsc
            +--- agents
                  +--- SAMPLEAGENT
                        +--- agent.properties
                        +--- SAMPLEAGENT_create.mqsc
                        +--- SAMPLEAGENT_delete.mqsc

+---logs
      +--- SAMPLECOORD
            +--- agents
                  +--- SAMPLEAGENT
                        +--- logs
```

此範例假設已停用佇列管理程式的安全功能。下列指令（在 `runmqsc` 中執行）會在重新啟動佇列管理程式之後停用安全功能：

```
runmqsc queue manager
alter qmgr CONNAUTH(NONE);
alter qmgr CHLAUTH(DISABLED);
end;
```

對於在 MFT for IBM MQ 8.0 或更新版本中啟用安全的配置，**CONNAUTH** 需要所有與佇列管理程式連接的 MFT 指令提供使用者 ID 和密碼認證。您可以為每個指令套用額外的參數 **-mquserid** 和 **-mqpassword**，或是定義 MQMFTCredentials.xml 檔案。下列範例認證檔會定義使用者 ID fteuser；在連接到佇列管理程式 SAMPLECOORD 時，會對此 ID 使用 MyPassword 密碼：

```
<tns:mqmftCredentials xmlns:tns="http://wmqfte.ibm.com/MQMFTCredentials"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/MQMFTCredentials MQMFTCredentials.xsd">
  <tns:qmgr mqPassword="MyPassword" MyUserId="fteuser" name="SAMPLECOORD"/>
</tns:mqmftCredentials>
```

如需相關資訊，請參閱 [MFT 和 IBM MQ 連線鑑別](#)。

附註：

- 若要尋找 MFT 配置目錄，請使用 **fteDisplayVersion -v** 指令。
- z/OS 使用者的 MQMFTCredential.xml 檔案可在具有變數記錄格式 (RECFM=V) 或未定義記錄格式 (RECFM=U) 之分割資料集中的成員中找到。
- 對於已啟用安全的配置，請將下列參數新增至下列步驟，以將認證與相關佇列管理程式相關聯：
-credentialsFile full credential file path。
- MQMFTCredential.xml 中的明碼密碼可以使用下列指令加以模糊：

```
fteObfuscate -credentialsFile full file path to MQMFTCredentials.xml
```

程序

1. 建立協調佇列管理程式。

協調佇列管理程式是單一佇列管理程式，用於接收來自其代理程式的所有傳輸日誌及狀態資訊。請執行下列指令：

```
fteSetupCoordination -coordinationQMgr coordination_qmgr_name
```

這會建立基本最上層配置，並建立 IBM MQ Script 檔來呼叫 *coordination_qmgr_name.mqsc*。接著需要執行下列 IBM MQ 指令，將配置載入佇列管理程式：

```
runmqsc queue manager name < coordination_qmgr_name.mqsc
```

註：您可以使用如下，將 TCP 用戶端連線到佇列管理程式：

```
fteSetupCoordination -coordinationQMgr coordination_qmgr_name
-coordinationQMgrHost coordination_qmgr_host -coordinationQMgrPort coordination_qmgr_port
-coordinationQMgrChannel coordination_qmgr_channel
```

您必須為所建立的 *coordination_qmgr_name.mqsc*，在協調佇列管理程式執行所在的機器上，執行 **runmqsc** 指令。

2. 建立指令佇列管理程式。

指令佇列管理程式是單一佇列管理程式，此佇列管理程式已預先配置，讓 IBM MQ 基礎架構可以將 MFT 要求路由到相關代理程式。請執行下列指令：

```
fteSetupCommands -connectionQMgr Command QM Name -p Coordination QM Name
```

這會在協調目錄中建立 *command.properties* 檔案。請註意，**-p** 為選用項目；對於針對預設協調而設定的指令並非必要。

註：您可以使用如下，將 TCP 用戶端連線到佇列管理程式：

```
fteSetupCommands -p coordination_qmgr_name -commandQMgr connection_qmgr_name
-commandQMgrHost connection_qmgr_host -commandQMgrPort connection_qmgr_port
-commandQMgrChannel connection_qmgr_channel
```

3. 建立代理程式。

代理程式是可以用於傳送及接收檔案的應用程式。請執行下列指令：

```
fteCreateAgent -p coordination_qmgr_name -agentName agent_name -agentQMgr agent_qmgr_name
```

這會在協調下建立代理程式配置，並建立一個 IBM MQ Script 檔，以呼叫代理程式配置目錄中的 `agent_name.mqsc`。

執行下列 IBM MQ 指令，將 IBM MQ Script 檔載入佇列管理程式中：

```
runmqsc agent_qmgr_name < agent_name_create.mqsc file
```

註：您可以使用如下，將 TCP 用戶端連線到佇列管理程式：

```
fteCreateAgent -p coordination_qmgr_name -agentName agent_name -agentQMgr agent_qmgr_name
-agentQMgrHost agent_qmgr_host -agentQMgrPort agent_qmgr_port -agentQMgrChannel
agent_qmgr_channel
```

4. 啟動代理程式。

請執行下列指令：

```
fteStartAgent -p coordination_qmgr_name agentName
```

代理程式會在背景中啟動，並傳回命令提示字元。若要檢查代理程式確在執行，請執行下列指令：

```
fteListAgents -p coordination_qmgr_name
```

這會顯示代理程式的狀態。如果代理程式順利執行，則會報告其處於「備妥」狀態。

結果

基本 MFT 基礎架構已經就緒，您已可使用 **fteCreateTransfer** 指令要求傳輸。或者，如果可以使用「IBM MQ Explorer」，則可使用 MFT 外掛程式來建立及監視傳輸。

您可以重複「步驟 3：建立代理程式」，為配置新增更多的代理程式。若是使用 TCP 用戶端連線，這些代理程式可以位在不同的機器上。若為不同的機器，每台機器都必須重複 **fteSetupCoordination** 及 **fteSetupCommands** 指令，但不需要再執行 mqsc Script。

比較複雜之配置的協調及各個代理程式可有個別的佇列管理程式。在這些情況下，必須連接各個佇列管理程式。

相關工作

如果 **fteListAgents** 指令未列出您的 MFT 代理程式，怎麼辦？

相關參考

[fteSetupCoordination](#)

[fteSetupCommands: 建立 MFT command.properties 檔](#)

[fteCreateAgent](#)

[fteObfuscate: 加密機密資料](#)

[MFT 認證檔案格式](#)

[MFT agent.properties 檔案](#)

配置多重實例佇列管理程式以使用 MFT

IBM WebSphere MQ 7.0.1 及更新版本支援建立多重實例佇列管理程式。多重實例佇列管理程式可自動在待命伺服器上重新啟動。Managed File Transfer 支援與多重實例代理程式佇列管理程式、多重實例協調佇列管理程式，以及多重實例指令佇列管理程式的連線。

配置多重實例佇列管理程式

重要: 如需配置 IBM MQ 多重實例佇列管理程式的相關資訊，請參閱第 400 頁的『[多重實例佇列管理程式](#)』。在嘗試配置多重實例佇列管理程式以使用 Managed File Transfer 之前，請確定您已閱讀此資訊。

使用多重實例佇列管理程式作為代理程式佇列管理程式

若要讓代理程式能夠同時連接至多重實例佇列管理程式的作用中及待用實例，請將 `agentQMGrStandby` 內容新增至代理程式的 `agent.properties` 檔。`agentQMGrStandby` 內容可定義待命佇列管理程式實例的用戶端連線所使用的主機名稱及埠號。此內容的值必須以 MQ CONNAME 格式指定，亦即 `host_name(port_number)`。

`agentQMGr` 內容可指定多重實例佇列管理程式的名稱。`agentQMGrHost` 內容可指定作用中佇列管理程式實例的主機名稱，`agentQMGrPort` 內容則會指定作用中佇列管理程式實例的埠號。代理程式必須以用戶端模式連接至多重實例佇列管理程式的作用中及待命實例。

如需相關資訊，請參閱 [MFT agent.properties](#) 檔案。

此範例顯示連接至多重實例佇列管理程式 QM_JUPITER 之 AGENT1 的 `agent.properties` 檔案內容。QM_JUPITER 的作用中實例位於系統 `host1` 上，並使用埠號 1414 進行用戶端連線。QM_JUPITER 的待命實例位於系統 `host2` 上，並使用埠號 1414 進行用戶端連線。

```
agentName=AGENT1
agentDesc=
agentQMGr=QM_JUPITER
agentQMGrPort=1414
agentQMGrHost=host1
agentQMGrChannel=SYSTEM.DEF.SVRCONN
agentQMGrStandby=host2(1414)
```

使用多重實例佇列管理程式作為協調佇列管理程式

若要啟用與多重實例協調佇列管理程式的作用中及待命實例的連線，請將 `coordinationQMGrStandby` 內容新增至 Managed File Transfer 拓撲中的所有 `coordination.properties` 檔案。

如需相關資訊，請參閱 [MFT coordination.properties](#) 檔。

此範例顯示指定多重實例協調佇列管理程式 QM_SATURN 之連線詳細資料的 `coordination.properties` 檔案內容。QM_SATURN 的作用中實例位於系統 `coordination_host1` 上，並使用埠號 1420 進行用戶端連線。QM_SATURN 的待命實例位於系統 `coordination_host2` 上，並使用埠號 1420 進行用戶端連線。

```
coordinationQMGr=QM_SATURN
coordinationQMGrHost=coordination_host1
coordinationQMGrPort=1420
coordinationQMGrChannel=SYSTEM.DEF.SVRCONN
coordinationQMGrStandby=coordination_host2(1420)
```

Managed File Transfer 獨立式日誌程式必須一律以連結模式連接至其佇列管理程式。搭配使用獨立式日誌程式與多重實例協調佇列管理程式時，請以連結模式將獨立式日誌程式連接至不同的佇列管理程式。此作業步驟詳述於第 616 頁的『[MFT 獨立式日誌程式的替代配置](#)』中。您必須使用多重實例協調佇列管理程式的兩個實例的主機名稱及埠號，定義獨立式日誌程式的佇列管理程式與協調佇列管理程式之間的通道。如需如何執行此動作的相關資訊，請參閱第 400 頁的『[多重實例佇列管理程式](#)』。

IBM MQ Explorer 的 Managed File Transfer 外掛程式以用戶端模式連接至協調佇列管理程式。如果多重實例協調佇列管理程式的作用中實例失敗，則協調佇列管理程式的待命實例即會成為作用中實例，讓外掛程式重新連接。

Managed File Transfer 指令 **fteList*** 及 **fteShowAgentDetails** 可直接連接至協調佇列管理程式。如果多重實例協調沒有可用的作用中實例，這些指令將會嘗試連接至協調佇列管理程式的待用實例。

使用多重實例佇列管理程式作為指令佇列管理程式

若要啟用與多重實例指令佇列管理程式的作用中及待命實例的連線，請將 `connectionQMGrStandby` 內容新增至 Managed File Transfer 拓撲中的所有 `command.properties` 檔案。

如需相關資訊，請參閱 `MFT command.properties` 檔案。

此範例顯示指定多重實例指令佇列管理程式 `QM_MARS` 之連線詳細資料的 `command.properties` 檔案內容。`QM_MARS` 的作用中實例位於系統 `command_host1` 上，並使用埠號 1424 進行用戶端連線。`QM_MARS` 的待命實例位於系統 `command_host2` 上，並使用埠號 1424 進行用戶端連線。

```
connectionQMGr=QM_SATURN
connectionQMGrHost=command_host1
connectionQMGrPort=1424
connectionQMGrChannel=SYSTEM.DEF.SVRCONN
connectionQMGrStandby=command_host2(1424)
```

相關概念

第 598 頁的『[連接至 IBM MQ](#)』

使用 IBM MQ 佇列管理程式的所有網路通訊（包括與 Managed File Transfer 相關的通訊）都涉及 IBM MQ 通道。IBM MQ 通道代表網路鏈結的一端。通道分類為訊息通道或 MQI 通道。

相關工作

第 599 頁的『[配置 MFT 網路佇列管理程式](#)』

如果您的 Managed File Transfer 網路包含多個 IBM MQ 佇列管理程式，這些 IBM MQ 佇列管理程式必須能夠在遠端彼此進行通訊。

第 600 頁的『[配置 MFT 的協調佇列管理程式](#)』

執行 **fteSetupCoordination** 指令之後，請執行 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name` 目錄中的 `coordination_qmgr_name.mqsc` Script，以執行協調佇列管理程式所需的配置。但如果要手動執行此配置，請在協調佇列管理程式上執行下列步驟。

保留 MFT 日誌訊息

Managed File Transfer 會將檔案傳送進度及日誌資訊，傳送至協調佇列管理程式。協調佇列管理程式會將此資訊發佈至 `SYSTEM.FTE` 主題的任何相符訂閱。如果沒有訂閱，則不保留此資訊。

確保保留資訊的方法

如果傳送進度或日誌資訊對您的業務很重要，您必須採取下列其中一個步驟，以確保將資訊保留下來：

- 使用 Managed File Transfer 資料庫日誌程式，將已發佈的訊息複製到 `SYSTEM.FTE/Log` 主題至 Oracle 或 Db2 資料庫。
- 定義 `SYSTEM.FTE` 主題，將發佈資訊儲存在 IBM MQ 佇列上。在傳送任何檔案傳送之前定義此訂閱，以確保所有進度及日誌訊息都保留在佇列上。
- 撰寫應用程式，以使用訊息佇列介面 (MQI) 或 IBM MQ JMS 來建立可延續訂閱，並處理分送至訂閱的發佈。在傳送任何檔案之前，此應用程式必須在作業中，以確保應用程式能夠接收所有進度及日誌訊息。

下列幾節會詳述以上每一種方法。

請不要依賴「IBM MQ Explorer」外掛程式來保留日誌資訊。

使用 Managed File Transfer 資料庫日誌程式來保留日誌訊息

資料庫日誌程式是 Managed File Transfer 的選用元件，可用來將日誌資訊複製到資料庫中進行分析及審核。資料庫日誌程式是一個獨立式 Java 應用程式，可安裝在管理協調佇列管理程式及資料庫的系統上。如需資料庫日誌程式的相關資訊，請參閱第 606 頁的『[配置 MFT 日誌程式](#)』。

使用 IBM MQ Explorer 外掛程式來保留進度和日誌訊息

第一次啟動「IBM MQ Explorer」外掛程式的實例時，此實例會在協調佇列管理程式上建立可延續訂閱。這個可延續訂閱用來收集「傳送日誌」及「現行傳送進度」視圖中顯示的資訊。

可延續訂閱的名稱會加上字首，以顯示訂閱是由 IBM MQ Explorer MFT 外掛程式、主機名稱及使用者名稱所建立，例如 MQExplorer_MFT_Plugin_HOST_TJWatson。

加上此字首是為了方便管理者刪除「IBM MQ Explorer」外掛程式實例目前不再使用的可延續訂閱。

在協調佇列管理程式上使用可延續訂閱會造成在 SYSTEM.MANAGED.DURABLE 佇列上建置訊息。如果您有流量高的 Managed File Transfer 網路，而且（或者）又不常使用「IBM MQ Explorer」外掛程式，則此訊息資料可能會填入本端檔案系統中。

若要停止發生此情況，可以指定 IBM MQ Explorer 外掛程式對協調佇列管理程式使用不可延續的訂閱。在 IBM MQ Explorer 中執行下列步驟：

1. 選取視窗 > 喜好設定 > MQ Explorer > Managed File Transfer
2. 從傳送日誌訂閱類型清單中，選擇 NON_DURABLE。

將發佈儲存在 IBM MQ 佇列

若要將日誌或進度訊息儲存在 IBM MQ 佇列，請在協調佇列管理程式上配置訂閱，這會將訊息轉遞至此佇列。例如，若要將所有日誌訊息轉遞至名為 LOG.QUEUE 的佇列，請提交下列 MQSC 指令：

```
define sub(MY.SUB) TOPICSTR('Log/#') TOPICOBJ(SYSTEM.FTE) DEST(LOG.QUEUE)WSHEMA(TOPIC)
```

在日誌訊息轉遞至 IBM MQ 佇列之後，它們會持續保存在佇列上，直到使用該佇列的 IBM MQ 應用程式處理這些訊息為止。

撰寫應用程式來管理 SYSTEM.FTE 主題的可延續訂閱

您可以使用 IBM MQ 支援的其中一個應用程式設計介面，撰寫應用程式來管理他們自己對 SYSTEM.FTE 主題的可延續訂閱。這些應用程式可以接收 IBM MQ 佇列或日誌訊息，並採取符合業務需求的方式進行處理。

如需可用應用程式設計介面的相關資訊，請參閱[開發應用程式](#)。

配置 MFT 日誌程式

當 Managed File Transfer 傳輸檔案時，它會將其動作的相關資訊發佈至協調佇列管理程式上的主題。資料庫日誌程式是 Managed File Transfer 的選用元件，可用於將此資訊複製到資料庫中進行分析及審核。

日誌程式有三個版本：

-  獨立式檔案日誌程式
- 獨立式資料庫日誌程式
- Java Platform, Enterprise Edition (Java EE) 日誌程式

IBM i 上的日誌程式



IBM i 平台不支援 Managed File Transfer 日誌程式。

獨立式檔案日誌程式



獨立式檔案日誌程式是一種 Java 程序，可以在管理協調佇列管理程式的系統上執行，或是在管理與協調佇列管理程式連接之佇列管理程式的系統上執行。獨立式檔案日誌程式使用 IBM MQ 連結來連接至其相關聯的佇列管理程式。獨立式日誌程式是使用 **fteCreateLogger** 指令建立的。

Windows 您可以 Windows 服務方式執行獨立式檔案日誌程式，以確保檔案日誌程式在您從 Windows 階段作業登出時繼續執行，且可以配置為在系統重新啟動時自動啟動。如需相關資訊，請參閱第 607 頁的『安裝 MFT 獨立式檔案日誌程式』。

下列平台不支援獨立式檔案日誌程式：

- **z/OS** z/OS
- **IBM i** IBM i

獨立式資料庫日誌程式

獨立式資料庫日誌程式是一種 Java 應用程式，可安裝在管理佇列管理程式及資料庫的系統上。獨立式資料庫日誌程式通常會與協調佇列管理程式安裝在相同的系統上，但也可安裝在與任何可連線至協調佇列管理程式的佇列管理程式相同的系統上。獨立式資料庫日誌程式使用 IBM MQ 連結來連接到其相關聯的佇列管理程式，並使用第 2 類或第 4 類 JDBC 驅動程式連接至 Db2 或 Oracle 資料庫。需要這些連線類型的原因在於，獨立式資料庫日誌程式使用佇列管理程式的 XA 支援來協調佇列管理程式及資料庫上的廣域交易，進而保護資料。

Windows 如果您使用的是 Windows 系統，則可以將獨立式日誌程式作為 Windows 服務來執行，以確保日誌程式在您從 Windows 階段作業登出後能夠繼續執行。如需相關資訊，請參閱獨立式資料庫日誌程式的第 613 頁的『安裝 MFT 獨立式資料庫日誌程式』。

Java EE 資料庫日誌程式

Java EE 資料庫日誌程式以 EAR 檔形式提供，可以將其安裝到應用程式伺服器中。如果具備現有 Java EE 應用程式伺服器環境可供使用，則與使用獨立式資料庫日誌程式相比，Java EE 資料庫日誌程式可能更為方便，因為可以在管理 JEE 資料庫日誌程式的同時管理其他企業應用程式。您也可以將 Java EE 資料庫日誌程式安裝在與管理 IBM MQ 伺服器及資料庫的系統不相同的系統上。Java EE 資料庫日誌程式支援與 Db2 及 Oracle 資料庫搭配使用。當安裝在 WebSphere Application Server 7.0 上時，Java EE 資料庫日誌程式也支援 Oracle Real Application Clusters。

如需如何配置日誌程式的相關指示，請參閱下列主題：

- 第 607 頁的『安裝 MFT 獨立式檔案日誌程式』
- 第 613 頁的『安裝 MFT 獨立式資料庫日誌程式』
- 第 617 頁的『安裝 MFT 的 Java EE 資料庫日誌程式』

ULW 安裝 MFT 獨立式檔案日誌程式

獨立式檔案日誌程式是必須使用 IBM MQ 連結連接至協調佇列管理程式的 Java 處理程序。若要定義獨立式檔案日誌程式，請使用 **fteCreateLogger** 指令，並遵循本主題中的步驟。

關於這項作業

如需獨立式檔案日誌程式的相關資訊，請參閱第 606 頁的『配置 MFT 日誌程式』。本主題中的步驟會配置日誌程式，以連接至協調佇列管理程式。如需替代日誌程式配置，請參閱第 616 頁的『MFT 獨立式日誌程式的替代配置』。

下列平台不支援獨立式檔案日誌程式：

- **z/OS** z/OS
- **IBM i** IBM i

程序

1. 確定您已安裝 Managed File Transfer Logger 元件。如需相關資訊，請參閱 [Managed File Transfer 產品選項](#)

- 執行 **fteCreateLogger** 指令，以指定協調佇列管理程式，並將參數 `-loggerType` 設為 `FILE`，以建立您的獨立式檔案日誌程式。如需相關資訊，請參閱 [fteCreateLogger](#)。
 - 選擇性的: 如果您想要使用自訂格式，請修改 **fteCreateLogger** 指令所建立的 XML 檔。日誌格式定義位於 `FileLoggerFormat.xml` 檔中。如需相關資訊，請參閱第 608 頁的『MFT 獨立式檔案日誌程式格式』。
 - 針對協調佇列管理程式執行 **fteCreateLogger** 指令所提供的 `MQSC` 指令，以建立日誌程式佇列。
 - 找出要執行日誌程式程序的使用者，並配置該使用者的權限。如需相關資訊，請參閱第 613 頁的『配置 MFT 獨立式檔案日誌程式的使用者存取權』。
 - 選擇性的: 您可以編輯您在執行 **fteCreateLogger** 指令時所建立的 `logger.properties` 檔，進一步配置獨立式檔案日誌程式。此檔案是由鍵值組所組成的 Java 內容檔。`logger.properties` 檔案位於 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` 目錄中。如需可用內容及其影響的相關資訊，請參閱 [MFT 日誌程式配置內容](#)。
- 7. Windows**
- 選擇性的: 如果您使用的是 Windows 系統，則可以將獨立式檔案日誌程式當作 Windows 服務來執行。請搭配執行 **fteModifyLogger** 指令與 `-s` 參數。如需相關資訊，請參閱 [fteModifyLogger](#)。
- 使用 **fteStartLogger** 指令來啟動獨立式檔案日誌程式。如需相關資訊，請參閱 [fteStartLogger](#)。
- 如果您已執行前一個步驟，並在 Windows 上搭配使用 **fteModifyLogger** 指令與 `-s` 參數，則獨立式檔案日誌程式會以 Windows 服務方式啟動。
- 檢查日誌程式輸出。獨立式檔案日誌程式會產生兩種類型的輸出：檔案傳送審核資料及日誌程式診斷資料。檔案傳送審核資料位於 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name/logs` 中。日誌程式診斷資料位於 `MQ_DATA_PATH/mqft/logs/coordination_qmgr_name/loggers/logger_name` 中
- 您可以使用 **fteStopLogger** 指令來停止日誌程式。如需相關資訊，請參閱 [fteStopLogger](#)。

結果

ULW MFT 獨立式檔案日誌程式格式

檔案日誌程式寫入的訊息資訊格式可定義於 `FileLoggerFormat.xml` 檔中。

日誌程式的配置目錄位於 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` 中。在建立新的檔案日誌程式時會建立此檔案，其中包含檔案日誌程式所使用的預設定義集。如需預設日誌格式定義的相關資訊，請參閱 [MFT 獨立式檔案日誌程式預設日誌格式](#)。

如果您要指定自己的自訂日誌格式，請編輯 `FileLoggerFormat.xml` 檔案。

自訂日誌格式定義

日誌格式定義由一組訊息類型組成，各個訊息類型分別有其格式定義。訊息類型的格式定義包含 XPATH 格式中提供的一組插入項目，以及用來區隔各個插入項目的分隔字元。插入項目的排序可決定在對日誌檔產生的輸出字行中放置內容的順序。例如，以下是 `callStarted` 訊息類型的定義：

```
<callStarted>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/transaction/action/
        @time</insert>
      <insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/agent/
        @agent</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/agent/@QMGr</insert>
      <insert type="user" width="0" ignoreNull="false">/transaction/job/name</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/transferSet/
        call/command/@type</insert>
      <insert type="user" width="0" ignoreNull="true">/transaction/transferSet/
        call/command/@name</insert>
      <insert type="system" width="0" ignoreNull="true">callArguments</insert>
    </inserts>
  </format>
</callStarted>
```



```
</format>
</callStarted>
```

此格式會在日誌檔中產生如下的字行：

```
2011-11-25T10:53:04;414d5120514d5f67627468696e6b20206466cf4e20004f02;[CSTR];
AGENT1;AGENT_QM;Managed Call;executable;echo;call test;
```

在格式定義中提供的插入項目所使用的順序，即為資訊出現在日誌檔中字行的順序。如需定義 `FileLoggerFormat.xml` 檔案格式之 XML 綱目的相關資訊，請參閱 [獨立式檔案日誌程式格式 XSD](#)。

訊息類型

FTE 代理程式會將一系列不同訊息類型寫入 `SYSTEM.FTE/Log` 子主題。如需相關資訊，請參閱 [SYSTEM.FTE topic](#)。日誌檔定義可包含下列訊息類型的格式定義：

```
callCompleted
callStarted
monitorAction
monitorCreate
monitorFired
notAuthorized
scheduleDelete
scheduleExpire
scheduleSkipped
scheduleSubmitInfo
scheduleSubmitTransfer
scheduleSubmitTransferSet
transferStarted
transferCancelled
transferComplete
transferDelete
transferProgress
```

訊息格式可能不同。大部分的訊息類型會在日誌檔中為每一個取自 `SYSTEM.FTE/Log` 子主題的日誌訊息寫入單一行。這會導致日誌格式定義中提供的 XPATH 位址與訊息根目錄產生關聯。以下是使用此方法寫入輸出的訊息類型：

```
callCompleted
callStarted
monitorAction
monitorCreate
monitorFired
notAuthorized
scheduleDelete
scheduleExpire
scheduleSkipped
scheduleSubmitInfo
scheduleSubmitTransfer
transferStarted
transferCancelled
transferComplete
transferDelete
```

另一種用來寫入日誌訊息的方法，是使用多行來表示日誌訊息內傳送集中的項目。在此情況下，會將提供的格式套用至日誌訊息內傳送集中的每一個項目。如果您想要併入傳送集內每一個項目的特定資訊，則提供的 XPATH 就必須以項目作為其 XPATH 根目錄。以下是使用此方法寫入輸出的訊息類型：

```
scheduleSubmitTransferSet
transferProgress
```

傳送集中的每一個項目會分別寫入成一行輸出。您想要對傳送集中所有項目進行修正的資訊，仍可使用與日誌訊息的根目錄相對的 XPATH 位址。在下列簡化的 `transferProgress` 格式定義範例中，會對時間戳記

及傳送 ID 進行修正。任何以某項目作為其根目錄而相對的資訊，都會隨著寫入的每一個字行而有所不同。在此範例中，會寫入每一個項目的來源及目的地檔案資訊。

```
<transferProgress>
  <format>
    <inserts>
      <insert type="user" width="19" ignoreNull="false">/transaction/action/
        @time</insert>
      <insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
      <insert type="system" width="6" ignoreNull="false">type</insert>
      <insert type="user" width="3" ignoreNull="true">status/@resultCode</insert>
      <insert type="user" width="0" ignoreNull="false">source/file |
        source/queue</insert>
      <insert type="user" width="0" ignoreNull="false">source/file/@size |
        source/queue/@size</insert>
      <insert type="user" width="5" ignoreNull="true">source/@type</insert>
      <insert type="user" width="6" ignoreNull="true">source/@disposition</insert>
      <insert type="user" width="0" ignoreNull="false">destination/file |
        destination/queue</insert>
      <insert type="user" width="0" ignoreNull="false">destination/file/@size |
        destination/queue/@size</insert>
      <insert type="user" width="5" ignoreNull="true">destination/@type</insert>
      <insert type="user" width="9" ignoreNull="true">destination/@exist</insert>
      <insert type="user" width="0" ignoreNull="true">status/supplement</insert>
    </inserts>
    <separator></separator>
  </format>
</transferProgress>
```

這會產生一行以上的日誌檔項目，格式如下：

```
2011-11-25T13:45:16;414d5120514d5f67627468696e6b20206466cf4e20033702;[TPR0];0
;/src/test1.file;3575;file;leave ;/dest/test1.file;3575;file;overwrite;;
2011-11-25T13:45:16;414d5120514d5f67627468696e6b20206466cf4e20033702;[TPR0];0
;/src/test2.file;3575;file;leave ;/dest/test2.file;3575;file;overwrite;;
```

插入項目格式

定義訊息類型的格式時，有兩種類型的插入項目可供使用：**使用者**及**系統**。插入項目的類型定義於插入元素的 **type** 屬性中。這兩種類型的插入項目也可透過插入元素的 **width** 及 **ignoreNull** 屬性來自訂其佈置。例如：

```
<insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
```

在此範例中，插入項目會採用在 `/transaction/@ID` 的日誌訊息中找到的資訊，並將其修除或填補為 48 個字元，然後再寫入日誌中。如果 `/transaction/@ID` 的內容是空值，則會在資訊填補為 48 個字元後寫入字串 `null`，因為 `ignoreNull` 屬性設為 `false`。如果 `ignoreNull` 設為 `true`，則空字串會填補為 48 個字元，並加以寫入。設定 `width="0"` 表示不修整欄寬，但不表示修整寬度為 0。當發現非預期的空值時，可以透過此方式使用 `ignoreNull` 屬性來偵測日誌中的空值。這在對新的日誌檔定義進行除錯時，可能會很有幫助。

使用者定義插入項目

使用者插入項目包含要寫入該插入項目的資訊的 XPATH 位址。此位址參照在 FTE 日誌訊息中找到的部分資訊。如需日誌訊息格式的相關資訊，請參閱：

- [檔案傳送日誌訊息格式](#)
- [排定的檔案傳送日誌訊息格式](#)
- [MFT 監視器日誌訊息格式](#)

系統定義插入項目


系統定義插入項目包含一個關鍵字，其參照無法在日誌訊息中找到或難以使用 XPATH 語言定義的部分資訊。

支援的系統插入項目是：

- `type` - 以簡短格式寫入日誌訊息類型。
- `callArguments` - 以空格區隔格式寫入提供給受管理呼叫的引數集。
- `transferMetaData` - 以逗點區隔的 `key=value` 格式寫入為傳送定義的 `meta` 資料項目集。

下表列出每一個訊息類型的系統定義插入項目的 "type" 值。

訊息類型	"type" 系統插入項目的值
callCompleted	[CCOM]
callStarted	[CSTR]
monitorAction	[MACT]
monitorCreate	[MCRT]
monitorFired	[MFIR]
notAuthorized	[AUTH]
scheduleDelete	[SDEL]
scheduleExpire	[SEXP]
scheduleSkipped	[SSKP]
scheduleSubmitInfo	[SSIN]
scheduleSubmitTransfer	[SSTR]
scheduleSubmitTransferSet	[SSTS]
transferStarted	[TSTR]
transferCancelled	[TCAN]
transferComplete	[TCOM]
transferDelete	[TDEL]
transferProgress	[TPRO]

 在 *MFT* 獨立式檔案日誌程式中排除訊息類型
如果您想要將特定訊息類型排除在檔案日誌程式輸出之外，可以使用空的訊息類型元素。

範例

例如，下列格式定義將會阻止檔案日誌程式輸出 `transferProgress` 訊息。

```
<?xml version="1.0" encoding="UTF-8"?>
<logFormatDefinition xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="1.00"
  xsi:noNamespaceSchemaLocation="FileLoggerFormat.xsd">
  <messageTypes>
    <transferProgress></transferProgress>
  </messageTypes>
</logFormatDefinition>
```

```
</messageTypes>
</logFormatDefinition>
```

ULW 定義 MFT 獨立式檔案日誌程式的自訂格式

您可以在日誌格式定義中定義自訂訊息類型的子集，以減少自訂日誌檔格式時需要的配置量。

關於這項作業

如果 `FileLoggerFormat.xml` 檔中不包括 `messageTypes` 元素，則該訊息類型格式將會使用預設格式。您只需要指定想要與預設值不同的格式。

範例

在此範例中，格式定義會將 `transferStarted` 訊息類型的預設格式取代為此精簡版本，只輸出啟動傳送的使用者。其他訊息類型未併入此日誌格式定義中，因此全都使用預設格式：

```
<?xml version="1.0" encoding="UTF-8"?>
<logFormatDefinition xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="1.00"
  xsi:noNamespaceSchemaLocation="FileLoggerFormat.xsd">
  <messageTypes>
    <transferStarted>
      <format>
        <inserts>
          <insert type="user" width="19" ignoreNull="false">/transaction/action/
            @time</insert>
          <insert type="user" width="48" ignoreNull="false">/transaction/@ID</insert>
          <insert type="system" width="6" ignoreNull="false">type</insert>
          <insert type="user" width="0" ignoreNull="true">/transaction/originator/
            userID</insert>
        </inserts>
        <separator>;</separator>
      </format>
    </transferStarted>
  </messageTypes>
</logFormatDefinition>
```

相關參考

[MFT 獨立式檔案日誌程式的預設日誌格式](#)
[獨立式檔案日誌程式格式 XSD](#)

ULW 減少 MFT 獨立式檔案日誌程式中重複的訊息

獨立式檔案日誌程式的日誌中可能會出現重複的日誌訊息。您可以使用 `logger.properties` 檔案來調整獨立式檔案日誌程式及減少重複數。

檔案日誌程式日誌中的重複訊息

在失敗的情況下，可能會將日誌訊息寫入獨立式檔案日誌程式的日誌中，而不會耗用來自 `SYSTEM.FTE/Log# topic committed to IBM MQ`。若發生此情況，當獨立式檔案日誌程式重新啟動時，它會再次擷取相同的訊息，並重新將它寫入至日誌檔中。請在手動查看日誌檔時或自動處理日誌檔時，計劃處理這些重複的可能性。為了協助偵測重複，獨立式檔案日誌程式在啟動時，會將下列訊息輸出至日誌檔：

```
BFGDB0054I: The file logger has successfully started
```

重複一律會在獨立式檔案日誌程式的啟動時間發生，因為此時會處理前一個實例失敗之前所讀取的最後一個訊息。知道啟動新實例的時機後，您就可以偵測是否預期會有重複，以及是否需要處理它們。

減少重複數

獨立式檔案日誌程式會將它處理的日誌訊息群組成交易，以改進效能。此批次大小是失敗時您會看到的重複訊息數上限。若要減少重複數，您可以在 `logger.properties` 檔案中調整下列內容：

```
wmqfte.max.transaction.messages
```

例如，將此項設定為 1，則複製的訊息數上限會減少為 1。請注意，修改此值會影響獨立式檔案日誌程式的效能，因此需要進行徹底測試，以確保這不會對系統產生不利影響。

`logger.properties` 檔案位於 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` 目錄中。如需可用內容及其效果的相關資訊，請參閱 [MFT 日誌程式配置內容](#)

配置 MFT 獨立式檔案日誌程式的使用者存取權

在測試環境中，您可以為一般使用者帳戶新增任何所需的新專用權。在正式作業環境中，建議您以執行工作所需的最小權限來建立新使用者。

關於這項作業

您必須將獨立式檔案日誌程式及 IBM MQ 安裝在單一系統上。配置使用者的權限，如下所示：

程序

1. 確保使用者具有讀取權限及執行權限（必要時），並且已在安裝 Managed File Transfer 的過程中安裝檔案。
2. 確定使用者有權限在位於配置目錄內的 `logs` 目錄中建立及寫入任何檔案。此目錄用於事件日誌，必要時也會用於診斷追蹤及「首次失敗資料擷取 (FFDC)」檔案。
3. 確定使用者具有專屬的群組，且不屬於任何對協調佇列管理程式具備多項權限的群組。使用者不得屬於 `mqm` 群組。在特定平台上，也會對工作人員群組自動指定佇列管理程式存取權；因此獨立式檔案日誌程式使用者不應屬於工作人員群組。您可以使用「IBM MQ Explorer」，檢視佇列管理程式本身及其內含物件的權限記錄。用滑鼠右鍵按一下物件，然後選取 **物件權限 > 管理權限記錄**。在指令行上，您可以使用 `dspmqaout`（顯示權限）或 `dmpmqaut`（傾出權限）指令。
4. 使用 IBM MQ Explorer 或 `setmqaut`（授與或撤銷權限）指令中的 **管理權限記錄** 視窗，以新增使用者專屬群組的權限（在 UNIX 上，IBM MQ 權限僅與群組相關聯，而不與個別使用者相關聯）。必要權限如下所示：
 - 佇列管理程式上的 `CONNECT` 及 `INQUIRE`（IBM MQ Java 程式庫需要 `INQUIRE` 權限才能運作）。
 - 對 `SYSTEM.FTE` 主題的 `SUBSCRIBE` 權限。
 - 對 `SYSTEM.FTE.LOG.RJCT.logger_name` 佇列的 `PUT` 權限。
 - 對 `SYSTEM.FTE.LOG.CMD.logger_name` 佇列的 `GET` 權限。

指定的拒絕及指令佇列名稱為預設名稱。如果您在配置獨立式檔案日誌程式佇列時選擇了不同的佇列名稱，請改為將權限新增至那些佇列名稱。

安裝 MFT 獨立式資料庫日誌程式

完成下列步驟，以安裝並配置獨立式資料庫日誌程式。

關於這項作業

重要：IBM i 平台不支援 Managed File Transfer 日誌程式。

如需獨立式資料庫日誌程式的相關資訊，請參閱第 606 頁的『[配置 MFT 日誌程式](#)』。

註：您無法同時對資料庫中的相同綱目執行多個資料庫日誌程式（獨立式或 Java EE）。如果嘗試這麼做，在嘗試將傳送日誌資料寫入資料庫時將會當機。

程序

1. 使用資料庫的適用文件安裝資料庫軟體。

如果 JDBC 支援是資料庫的選用元件，則必須安裝此元件。

2. 執行 **fteCreateLogger** 指令將參數 **-loggerType** 設為 DATABASE，以建立您的獨立式資料庫日誌程式。如需相關資訊，請參閱 [fteCreateLogger](#)。

預設的綱目名稱是 FTELOG。如果使用 FTELOG 以外的綱目名稱，則必須在進行下一步驟前，先編輯所提供且適用於您資料庫的 SQL 檔案 `ftelog_tables_db2.sql` 或 `ftelog_tables_oracle.sql`，以反映此綱目名稱。如需相關資訊，請參閱 [MFT 日誌程式配置內容](#) 中的 `wmqfte.database.schema`。

3. 使用資料庫的工具，建立所需的資料庫表格。

Multi 在多平台上，`ftelog_tables_db2.sql` 及 `ftelog_tables_oracle.sql` 檔案包含您可執行以建立表格的 SQL 指令。

z/OS 在 z/OS 上，您需要執行的檔案視您所使用的 Db2 for z/OS 版本而定：

- 若為 Db2 for z/OS 9.0 以及更舊版本，則執行 `ftelog_tables_zos.sql` 檔案來建立表格。此檔案建立表格時，對欄位使用 INTEGER 資料類型來表示傳送的檔案大小，以及與每一項傳送相關聯的表格 ID。
 - 若為 Db2 for z/OS 9.1 以及更新版本，則執行 `ftelog_tables_zos_bigint.sql` 檔案來建立表格。此檔案建立表格時，對欄位使用 BIGINT 資料類型來表示傳送的檔案大小，以及與每一項傳送相關聯的表格 ID。
4. 對日誌程式指令佇列管理程式執行 **fteCreateLogger** 指令所提供的 MQSC 指令，以建立日誌程式佇列。獨立式資料庫日誌程式在協調佇列管理程式上使用兩個佇列。第一個佇列是指令佇列，其中存放用於控制獨立式資料庫日誌程式作業的訊息。此指令佇列的預設名稱是 `SYSTEM.FTE.LOG.CMD.logger_name`。第二個佇列是拒絕佇列。由於獨立式資料庫日誌程式不會捨棄日誌訊息，因此如果日誌程式遇到它無法處理的訊息，它便會將該訊息放在拒絕佇列中以供檢查，有可能進行重新處理。不建議針對此用途使用佇列管理程式的無法傳送的郵件佇列，因為被拒絕的訊息沒有 DLH 標頭，並且不應因其他原因將被拒絕的訊息與無法傳送的郵件佇列中存在的訊息結合在一起。拒絕佇列的預設名稱是 `SYSTEM.FTE.LOG.RJCT.logger_name`。這兩個佇列定義於 **fteCreateLogger** 指令所產生的 MQSC Script 檔中。
 5. 選擇使用者及配置權限
 6. 選擇性的: 您可以編輯步驟 [第 614 頁](#) 的『2』中 **fteCreateLogger** 指令所建立的 `logger.properties` 檔，以進一步配置獨立式資料庫日誌程式。此檔案是由鍵值組所組成的 Java 內容檔。`logger.properties` 檔案位於 `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/loggers/logger_name` 目錄中。如需可用內容及其效果的相關資訊，請參閱 [MFT 日誌程式配置內容](#)。

7. **Windows**

- 選擇性的: 如果您使用的是 Windows 系統，則可以將獨立式資料庫日誌程式作為 Windows 服務來執行。請搭配執行 **fteModifyLogger** 指令與 **-s** 參數。如需相關資訊，請參閱 [fteModifyLogger](#)。
8. 選擇性的: 如果使用的資料庫是 Oracle，或是您要從遠端連接至 Db2 資料庫，則必須指定日誌程式用來對資料庫伺服器進行鑑別的使用者名稱及密碼。這個使用者名稱及密碼會指定在與 `MQMFTCredentials.xsd` 綱目所定義的格式相符的認證檔案中。如需相關資訊，請參閱 [MFT 認證檔案格式](#)。建立認證檔案後，您必須使用 `wmqfte.database.credentials.file` 內容，在 `logger.properties` 檔案中指定認證檔案的位置。
9. 使用 **fteStartLogger** 指令來啟動獨立式資料庫日誌程式。依預設，獨立式資料庫日誌程式會在背景執行，並且獨立式資料庫日誌程式會將輸出放在 `logs` 目錄的某個檔案中。如果您想要在前景中執行獨立式資料庫日誌程式，並且同時對主控台以及日誌檔產生輸出，請在 **fteStartLogger** 指令中加入 **-F** 參數。

如果您已執行前一個步驟，並在 Windows 上搭配使用 **fteModifyLogger** 指令與 **-s** 參數，則獨立式資料庫日誌程式會以 Windows 服務方式啟動。

搭配使用 MFT 與遠端資料庫

您可以使用 Managed File Transfer 日誌程式，與遠端系統上的資料庫進行通訊。

關於這項作業

如果安裝資料庫的機器不同於安裝有 Managed File Transfer 的機器，則請完成下列步驟。除非另有說明，否則這些步驟同時適用於 Db2 及 Oracle。

程序

1. 在安裝有 Managed File Transfer 的系統上安裝資料庫用戶端。
2. 將遠端資料庫伺服器新增至本端資料庫用戶端配置。Managed File Transfer 及 IBM MQ 必須更新此配置，才能正確存取資料庫。
3. 在 `logger.properties` 檔案中指定新內容，以使用認證檔案連接至資料庫：
`wmfte.database.credentials.file`。
註：舊版的 Managed File Transfer 使用的是內容 **`wmqfte.oracle.user`** 或 **`wmqfte.database.user`**，以及 **`wmqfte.oracle.password`** 或 **`wmqfte.database.password`**。這些內容現已淘汰。請改用 **`wmfte.database.credentials.file`**。
4. 僅限 Oracle：若要容許遠端連接至資料庫，請將協調佇列管理程式的 `qm.ini` 檔案中的 XAResourceManager 段落變更為以下內容（確保變更資料庫名稱、使用者名稱及使用者密碼，以符合自己的資訊）：
`Oracle_XA+Acc=P/ftelog/
qgw783jhT+SesTm=35+DB=FTEAUDIT1+SqlNet=FTEAUDIT1+threads=false`，以粗體強調顯示變更。
5. 僅限 Oracle：使用 **`wmqfte.oracle.host`** 及 **`wmqfte.oracle.port`** 內容，在 `logger.properties` 檔中指定主機及埠。主機及埠的預設值可讓您使用本端資料庫用戶端，因此如果先前使用了本端資料庫，您可能並未設定這些值。

相關參考

[MFT 日誌程式配置內容](#)

配置 MFT 獨立式資料庫日誌程式的使用者存取權

在測試環境中，您可以為一般使用者帳戶新增任何所需的新專用權。在正式作業環境中，建議您以執行工作所需的最小權限來建立新使用者。

關於這項作業

執行獨立式資料庫日誌程式所需的使用者帳戶的數目及類型，取決於您所使用的系統數目。您可以將獨立式資料庫日誌程式、IBM MQ 及您的資料庫安裝在單一系統上，或安裝在兩個系統上。獨立式資料庫日誌程式必須與 IBM MQ 位於相同的系統上。元件可安裝在下列拓撲中：

獨立式資料庫日誌程式、IBM MQ 及資料庫，全都在相同的系統上

您可以定義單一作業系統使用者，以與這三項元件搭配使用。此配置適用於獨立式資料庫日誌程式。獨立式資料庫日誌程式使用「連結」模式連接至 IBM MQ，連接至資料庫時則使用原生連線。

獨立式資料庫日誌程式及 IBM MQ 位於一個系統上，資料庫則位於個別系統

您可以為此配置建立兩個使用者：在執行獨立式資料庫日誌程式的系統上配置一個作業系統使用者，以及在資料庫伺服器上配置一個具有資料庫遠端存取權的作業系統使用者。此配置適用於使用遠端資料庫的獨立式資料庫日誌程式。獨立式資料庫日誌程式使用「連結」模式連接至 IBM MQ，存取資料庫時則使用用戶端連線。

為方便解說，本文的其餘指示均假設使用者的名稱為 `ftelog`，但您可以使用任何使用者名稱。配置使用者的權限，如下所示：

程序

1. 確保使用者具有讀取權限及執行權限（必要時），並且已將檔案作為 Managed File Transfer Remote Tools and Documentation 安裝的一部分進行安裝。
2. 確保使用者具有在 `logs` 目錄（位於配置目錄）中建立及寫入任何檔案的權限。此目錄用於事件日誌，必要的話還用於診斷追蹤及 FFDC 檔案。

3. 確定使用者具有專屬的群組，且不屬於任何對協調佇列管理程式具備多項權限的群組。使用者不得屬於 mqm 群組。在特定平台上，也會對工作人員群組自動指定佇列管理程式存取權；因此獨立式資料庫日誌程式使用者不應屬於工作人員群組。您可以使用「IBM MQ Explorer」，檢視佇列管理程式本身及其內含物件的權限記錄。用滑鼠右鍵按一下物件，然後選取物件權限 > 管理權限記錄。在指令行上，您可以使用 `dspmqaout` (顯示權限) 或 `dmpmqaut` (傾出權限) 指令。
4. 使用 IBM MQ Explorer 或 `setmqaut` (授與或撤銷權限) 指令中的 **管理權限記錄** 視窗，以新增使用者專屬群組的權限 (在 UNIX 上，IBM MQ 權限僅與群組相關聯，而不與個別使用者相關聯)。必要權限如下所示：
 - 佇列管理程式上的 CONNECT 及 INQUIRE (IBM MQ Java 程式庫需要 INQUIRE 權限才能運作)。
 - 對 SYSTEM.FTE 主題的 SUBSCRIBE 權限。
 - 對 SYSTEM.FTE.LOG.RJCT.*logger_name* 佇列的 PUT 權限。
 - 對 SYSTEM.FTE.LOG.CMD.*logger_name* 佇列的 GET 權限。

指定的拒絕及指令佇列名稱為預設名稱。如果您在配置獨立式資料庫日誌程式佇列時選擇了不同的佇列名稱，請改為將權限新增至那些佇列名稱。
5. 執行特定於所用資料庫的使用者配置。
 - 如果您的資料庫是 Db2，請執行下列步驟：

使用 Db2 時可透過數種機制來管理資料庫使用者。下列指示適用於以作業系統使用者為基礎的預設機制。

 - 確保 `fteolog` 使用者不屬於任何 Db2 管理群組 (例如 `db2iadm1`、`db2fadm1` 或 `dasadm1`)。
 - 授與使用者連接至資料庫權限，以及對步驟 2: 建立所需資料庫表格 中所建立表格的選取、插入及更新權限
 - 如果您的資料庫是 Oracle，請執行下列步驟：
 - 確保 `fteolog` 使用者不屬於任何 Oracle 管理群組 (例如，Windows 上的 `ora_dba`，或 UNIX 上的 `dba`)
 - 授與使用者連接至資料庫權限，以及對步驟 2: 建立所需資料庫表格 中所建立表格的選取、插入及更新權限

MFT 獨立式日誌程式的替代配置

一般而言，Managed File Transfer 獨立式日誌程式 (無論是檔案或是資料庫類型) 會與協調佇列管理程式位於相同的系統上，且會以 IBM MQ 連結模式連接至協調佇列管理程式。不過，也可以與任何可連線至協調佇列管理程式的佇列管理程式，安裝在相同的系統上。獨立式日誌程式可透過本身自動建立的訂閱來接收訊息。這是安裝指示中所述的配置。

但是，如果有網站特定的考量，您可以將獨立式日誌程式配置成以兩種其他方式接收訊息；此配置由 `wmqfte.message.source.type` 內容控制。此內容在 MFT 日誌程式配置內容 中說明。

管理訂閱

依預設，獨立式日誌程式會使用預設的可延續訂閱選項及受管理訂閱，就 SYSTEM.FTE/Log/# 主題建立其專屬訂閱 (也就是說，佇列管理程式會控制在將訊息傳遞至應用程式之前用來保留這些訊息的備份佇列)。如果訂閱或佇列上需要使用其他選項，您可以自行建立訂閱、設定您需要的選項，並將獨立式日誌程式配置成改用該訂閱。請記得要新增獨立式日誌程式的權限，才能使用您所建立的訂閱。

舉例來說，使用兩個萬用字元訂閱對日誌空間進行分割、將日誌從名稱以 FINANCE 開頭的代理程式傳送至某個資料庫中，以及將日誌從名稱以 ACCOUNTING 開頭的代理程式傳送至另一個資料庫中，皆屬於此配置的運用。使用此類型配置需要兩個獨立式日誌程式實例，且兩者分別要有其專屬 `logger.properties` 檔來參照必要的訂閱及其專屬的指令佇列及拒絕佇列。

若要僅收集來自名稱以 ACCOUNTING 開頭的代理程式的日誌訊息，請在協調佇列管理程式上以主題字串 `SYSTEM.FTE/Log/ACCOUNTING*` 建立訂閱物件。請將萬用字元用法值設為字元層次萬用字元。您還必須

將項目新增至日誌程式的 `logger.properties` 檔。比方說，例如您以這些設定建立稱為 `ACCOUNTING.LOGS` 的訂閱物件，請將下列項目新增至 `logger.properties` 檔：

```
wmqfte.message.source.type=administrative subscription
wmqfte.message.source.name=ACCOUNTING.LOGS
```

獨立式日誌程式只會處理以 `SYSTEM.FTE/Log/` 的主題字串開頭的日誌訊息。您可以指定更為嚴格的主題字串，但不能指定限制較少的字串。如果您在錯誤中指定了較鬆散的限定字串，則所有與 `SYSTEM.FTE/Log/` 以外的主題字串相關的發佈都會移至拒絕佇列，且獨立式日誌程式將會產生錯誤訊息 `BFGDB0002E`。此錯誤訊息意味著獨立式日誌程式配置有問題。

佇列

一般拓撲是指獨立式日誌程式與協調佇列管理程式執行於相同系統的架構。如果這種情況不可行，您可以在協調佇列管理程式上使用其他佇列管理程式上的佇列建立訂閱，作為訂閱目的地（使用遠端佇列定義，或者使用訂閱的 `DESTQMGR` 內容）。然後，日誌程式將可在管理第二個佇列管理程式的系統上執行，並讀取來自佇列的訊息。為確保交易完整性，獨立式日誌程式必須一律以連結模式連接至其佇列管理程式。您必須在獨立式日誌程式所連接的相同佇列管理程式上，定義拒絕佇列及指令佇列。佇列管理程式必須使用 `IBM WebSphere MQ 7.5` 或更新版本。

例如，若要收集訂閱放置在佇列 `USER.QUEUE` 上的日誌訊息，請將這些項目新增至 `logger.properties` 檔：

```
wmqfte.message.source.type=queue
wmqfte.message.source.name=USER.QUEUE
```

安裝 MFT 的 Java EE 資料庫日誌程式

請遵循下列指示安裝及配置 JEE 資料庫日誌程式，以與 `Managed File Transfer` 搭配使用。

關於這項作業

如需 `Java EE` 資料庫日誌程式的相關資訊，請參閱第 606 頁的『[配置 MFT 日誌程式](#)』主題。

註：`Java EE` 資料庫日誌程式無法與獨立式日誌程式同時執行，除非這些日誌程式使用的是不同的資料庫實例。

程序

1. 在安裝 `Java EE` 資料庫日誌程式之前，必須先準備環境。請遵循第 618 頁的『[準備安裝 MFT 的 Java EE 資料庫日誌程式](#)』主題中的指示作業。
2. 您可以在符合 `Java Platform, Enterprise Edition (Java EE)` 標準的應用程式伺服器中安裝 `Java EE` 資料庫日誌程式。如需相關指示，請參閱下列主題：
 - 第 620 頁的『[使用 WebSphere Application Server 7.0 安裝 MFT 的 Java EE 資料庫日誌程式](#)』
 - 第 624 頁的『[使用 WebSphere Application Server Community Edition 安裝 MFT 的 Java EE 資料庫日誌程式](#)』

相關工作

第 618 頁的『[準備安裝 MFT 的 Java EE 資料庫日誌程式](#)』

在安裝 `Java EE` 資料庫日誌程式之前，請遵循下列指示來準備 `Managed File Transfer` 環境。

第 620 頁的『[使用 WebSphere Application Server 7.0 安裝 MFT 的 Java EE 資料庫日誌程式](#)』

遵循下列指示，使用 `WebSphere Application Server 7.0` 來安裝及配置 `Managed File Transfer` 的 `Java Platform, Enterprise Edition (Java EE)` 資料庫日誌程式。

第 624 頁的『[使用 WebSphere Application Server Community Edition 安裝 MFT 的 Java EE 資料庫日誌程式](#)』

遵循下列指示，使用 `WebSphere Application Server Community Edition` 來安裝及配置 `Managed File Transfer` 的 `Java Platform, Enterprise Edition (Java EE)` 資料庫日誌程式。

第 627 頁的『[配置 MFT 的 Java EE 資料庫日誌程式的使用者存取權](#)』

當您配置 Managed File Transfer 的 Java Platform, Enterprise Edition (Java EE) 資料庫日誌程式時，您需要使用者帳戶才能存取 IBM MQ、資料庫及作業系統。所需的作業系統使用者數目，取決於您用來管理這些元件的系統數目。

第 629 頁的『[從獨立式資料庫日誌程式移轉至 MFT 的 Java EE 資料庫日誌程式](#)』

您可以從獨立式資料庫日誌程式移轉至 Java EE 資料庫日誌程式。必須停止獨立式資料庫日誌程式，然後再安裝 JEE 資料庫日誌程式。若要避免遺失或複製日誌項目，您必須停止將訊息發佈至 SYSTEM.FTE 主題，並在安裝 Java EE 資料庫日誌程式之後重新啟動它。移轉之前，請先備份資料庫。

相關參考

[MFT 日誌程式的權限](#)

準備安裝 MFT 的 Java EE 資料庫日誌程式

在安裝 Java EE 資料庫日誌程式之前，請遵循下列指示來準備 Managed File Transfer 環境。

關於這項作業

如需 Java EE 資料庫日誌程式的相關資訊，請參閱第 606 頁的『[配置 MFT 日誌程式](#)』主題。

程序

1. 使用資料庫的適用文件安裝資料庫軟體。

如果 JDBC 支援是資料庫的選用元件，則必須安裝此元件。

2. 使用您的資料庫提供的工具來建立資料庫。資料庫必須至少具有 8K 的表格空間及緩衝池頁面大小。

預設的綱目名稱是 FTELOG。如果您使用 FTELOG 以外的綱目名稱，必須先編輯您的資料庫所適用的 SQL 檔 `ftelog_tables_db2.sql` 或 `ftelog_tables_oracle.sql` 予以反映，才能繼續下一步。

3. 使用資料庫的工具，建立所需的資料庫表格。

Multi 在多平台上，`ftelog_tables_db2.sql` 及 `ftelog_tables_oracle.sql` 檔案包含您可執行以建立表格的 SQL 指令。

z/OS 在 z/OS 上，您需要執行的檔案視您所使用的 Db2 for z/OS 版本而定：

- 若為 Db2 for z/OS 9.0 以及更舊版本，則執行 `ftelog_tables_zos.sql` 檔案來建立表格。此檔案建立表格時，對欄位使用 INTEGER 資料類型來表示傳送的檔案大小，以及與每一項傳送相關聯的表格 ID。
 - 若為 Db2 for z/OS 9.1 以及更新版本，則執行 `ftelog_tables_zos_bigint.sql` 檔案來建立表格。此檔案建立表格時，對欄位使用 BIGINT 資料類型來表示傳送的檔案大小，以及與每一項傳送相關聯的表格 ID。
4. 如果您將綱目名稱變更為 FTELOG 以外的名稱，則必須在 EAR 檔中變更綱目名稱。如需相關資訊，請參閱第 619 頁的『[變更 MFT 的 Java EE 資料庫日誌程式中的綱目名稱](#)』。
 5. 在 IBM MQ 中建立拒絕佇列。
因為日誌程式一律不捨棄日誌訊息，所以日誌程式在遇到無法處理的訊息時，會將該訊息放置在拒絕佇列上，以進行檢查且可能會重新處理。請不要將佇列管理程式的「無法傳送的郵件」佇列用於此用途，因為拒絕的訊息並沒有 DLH 標頭，且不可與因為其他原因而放置到「無法傳送的郵件」佇列的訊息結合。**fteCreateLogger** 指令會建立拒絕佇列。此拒絕佇列的預設名稱是 `SYSTEM.FTE.LOG.RJCT.logger_name`
 6. 請遵循第 627 頁的『[配置 MFT 的 Java EE 資料庫日誌程式的使用者存取權](#)』主題中的指示。

下一步

現在您可以將 Java EE 資料庫日誌程式安裝在符合 Java EE 標準的應用程式伺服器中。請根據您所使用的應用程式伺服器，遵循下列主題中的指示作業：

- 第 620 頁的『[使用 WebSphere Application Server 7.0 安裝 MFT 的 Java EE 資料庫日誌程式](#)』
- 第 624 頁的『[使用 WebSphere Application Server Community Edition 安裝 MFT 的 Java EE 資料庫日誌程式](#)』

變更 MFT 的 Java EE 資料庫日誌程式中的綱目名稱

Java Platform, Enterprise Edition (Java EE) 資料庫日誌程式可使用具有非預設綱目名稱的資料庫。您必須變更 Java EE 資料庫日誌程式 EAR 檔中的綱目名稱。

關於這項作業

若要變更 Java EE 資料庫日誌程式所使用的綱目名稱，請完成下列步驟：

程序

1. 使用下列指令，從 EAR 檔中解壓縮 JPA JAR 檔：

```
jar -xvf ear_file lib/jpa_file
```

其中：

- *ear_file* 是 `com.ibm.wmqfte.databaselogger.jee.oracle.ear` 或 `com.ibm.wmqfte.databaselogger.jee.ear`，視您使用的是 Db2 還是 Oracle 而定。
- *jpa_file* 是 `com.ibm.wmqfte.web.jpa.oracle.jar` 或 `com.ibm.wmqfte.web.jpa.jar`，視您使用的是 Db2 還是 Oracle 而定。

2. 使用下列指令，從 JPA JAR 檔解壓縮 `persistence.xml` 檔：

```
jar -xvf lib/jpa_file META_INF/persistence.xml
```

其中：

- *jpa_file* 是 `com.ibm.wmqfte.web.jpa.oracle.jar` 或 `com.ibm.wmqfte.web.jpa.jar`，視您使用的是 Db2 還是 Oracle 而定。

3. 編輯 `persistence.xml` 檔，以變更下列字行：

```
<property name="openjpa.jdbc.Schema" value="schema_name" />
```

其中

- *schema_name* 是您想要使用的綱目名稱。

4. 使用下列指令，以修改過的 `persistence.xml` 檔更新 JPA JAR：

```
jar -uvf lib/jpa_file META_INF/persistence.xml
```

其中：

- *jpa_file* 是 `com.ibm.wmqfte.web.jpa.oracle.jar` 或 `com.ibm.wmqfte.web.jpa.jar`，視您使用的是 Db2 還是 Oracle 而定。

5. 使用下列指令，以修改過的 JPA JAR 檔更新 EAR 檔：

```
jar -uvf ear_file lib/jpa_file
```

其中：

- *ear_file* 是 `com.ibm.wmqfte.databaselogger.jee.oracle.ear` 或 `com.ibm.wmqfte.databaselogger.jee.ear`，視您使用的是 Db2 還是 Oracle 而定。
- *jpa_file* 是 `com.ibm.wmqfte.web.jpa.oracle.jar` 或 `com.ibm.wmqfte.web.jpa.jar`，視您使用的是 Db2 還是 Oracle 而定。

下一步

使用修改過的 EAR 檔來安裝 Java EE 資料庫日誌程式。

相關工作

第 620 頁的『[使用 WebSphere Application Server 7.0 安裝 MFT 的 Java EE 資料庫日誌程式](#)』
遵循下列指示，使用 WebSphere Application Server 7.0 來安裝及配置 Managed File Transfer 的 Java Platform, Enterprise Edition (Java EE) 資料庫日誌程式。

第 624 頁的『[使用 WebSphere Application Server Community Edition 安裝 MFT 的 Java EE 資料庫日誌程式](#)』

遵循下列指示，使用 WebSphere Application Server Community Edition 來安裝及配置 Managed File Transfer 的 Java Platform, Enterprise Edition (Java EE) 資料庫日誌程式。

在 WebSphere Application Server 7.0 中設定原生程式庫路徑

如果您在 WebSphere Application Server 7.0 上部署 Java Platform, Enterprise Edition (Java EE) 資料庫日誌程式應用程式，並且想要在應用程式與 IBM MQ 之間使用連結模式連線，則必須使用系統上 IBM MQ 原生程式庫的位置來配置 IBM MQ 傳訊提供者。

關於這項作業

如果您未在應用程式伺服器中設定原生程式庫路徑，則可能會在 WebSphere Application Server 7.0 系統輸出日誌中收到下列錯誤訊息：

```
A connection could not be made to WebSphere MQ for the following reason:  
CC=2;RC=2495;AMQ8568: The native JNI library 'mqjbnj' was not found. [3=mqjbnj]
```

請使用 WebSphere Application Server 7.0 管理主控台來完成下列步驟：

程序

1. 在導覽窗格中，展開**資源 > JMS > JMS 提供者**。
2. 選取位於 Connection Factory 之正確範圍的 IBM MQ 傳訊提供者，或建立連結模式連線的啟動規格。
註：使用 Server 範圍的原生路徑資訊勝過更高範圍的原生路徑資訊，而使用 Node 範圍的原生路徑資訊則勝過 Cell 範圍的原生路徑資訊。
3. 在「一般內容」下，於**原生程式庫路徑**欄位中輸入包含 IBM MQ 原生程式庫的目錄完整名稱。
例如，在 Linux 上，輸入 /opt/mqm/java/lib。只輸入一個目錄名稱。
4. 按一下**確定**。
5. 重新啟動應用程式伺服器以，重新整理配置。
6. 必要性的：再次重新啟動應用程式伺服器，以載入程式庫。

使用 WebSphere Application Server 7.0 安裝 MFT 的 Java EE 資料庫日誌程式

遵循下列指示，使用 WebSphere Application Server 7.0 來安裝及配置 Managed File Transfer 的 Java Platform, Enterprise Edition (Java EE) 資料庫日誌程式。

開始之前

安裝 JEE 資料庫日誌程式應用程式之前，請遵循第 618 頁的『[準備安裝 MFT 的 Java EE 資料庫日誌程式](#)』及第 620 頁的『[在 WebSphere Application Server 7.0 中設定原生程式庫路徑](#)』主題中的指示。

關於這項作業

如需 Java EE 資料庫日誌程式的相關資訊，請參閱第 606 頁的『[配置 MFT 日誌程式](#)』。

程序

1. 設定 XA JDBC 提供者：
 - a) 從 WebSphere Application Server 7.0 管理主控台導覽中選取 **資源 > JDBC > JDBC 提供者**。
 - b) 使用主控台精靈，按一下**新建**，以建立 JDBC 提供者。
 - c) 在精靈的步驟 1 中，從**資料庫類型**清單中選取您所使用的資料庫，然後從**提供者類型**清單中選取關聯的提供者類型。從**實作類型**清單中，選取 **XA 資料來源**。按下一步。

- d) 在精靈的步驟 2，確定所需的資料庫 Jar 檔的目錄位置已正確設定。按下一步。
 - e) 在摘要頁面上按一下**完成**，以建立 JDBC 提供者。
2. 建立鑑別別名。您可以分別為資料來源及 IBM MQ 建立不同的別名：
- a) 從 WebSphere Application Server 7.0 管理主控台導覽中選取 **安全 > 廣域安全**。
 - b) 在**鑑別**標題下，展開 **Java 鑑別和授權服務**。
 - c) 按一下 **J2C 鑑別資料**。鑑別別名頁面即會開啟。
 - d) 建立資料來源的鑑別別名：
 - i) 按一下**新建**。
 - ii) 輸入**別名、使用者 ID、密碼及說明**的詳細資料。在**使用者 ID 及密碼**欄位中輸入的詳細資料，必須符合您建立資料庫使用者時所輸入的詳細資料。如需相關資訊，請參閱第 627 頁的『[配置 MFT 的 Java EE 資料庫日誌程式的使用者存取權](#)』。
 - iii) 按一下**確定**。
 - e) 建立 IBM MQ 的鑑別別名：
 - i) 按一下**新建**。
 - ii) 輸入**別名、使用者 ID、密碼及說明**的詳細資料。在**使用者 ID 及密碼**欄位中輸入的詳細資料，必須符合 IBM MQ 安裝的使用者及密碼設定。
 - iii) 按一下**確定**。
3. 建立資料來源：
- a) 從 WebSphere Application Server 7.0 管理主控台導覽中選取 **資源 > JDBC > 資料來源**。
 - b) 選取**範圍**下拉清單，然後將範圍變更為適當的值。例如，Node=yourNode，Server=yourServer。
 - c) 使用主控台精靈，按一下**新建**，以建立資料來源。
 - d) 在精靈的步驟 1，在**資料來源名稱**欄位中輸入 wmqfte-database，然後在 **JNDI 名稱**欄位中輸入 jdbc/wmqfte-database。按下一步。
 - e) 在精靈的步驟 2 中，使用**選取現有的 JDBC 提供者**下拉清單，以選取上述步驟所建立的 JDBC 提供者。按下一步。
 - f) **Db2**：在精靈的步驟 3，在**驅動程式類型**欄位中輸入 4。
 - g) **Db2**：在**資料庫名稱、伺服器名稱及埠號**欄位中輸入詳細資料，然後按下一步。
 - Oracle**：在 **URL** 欄位中輸入連線 URL，然後在**資料儲存庫 Helper 類別名稱**欄位中選擇正確的資料儲存庫 Helper。
 - Oracle RAC**：在連接至 Oracle Real Application Cluster 時，連線 URL 必須包括連接至所有可用資料庫實例所需的主機資訊。
 - h) 在精靈的步驟 4 中，從 **XA 回復的鑑別別名**清單中選取您在步驟 2d 中定義的資料來源鑑別別名的名稱。請在**元件管理的鑑別別名及儲存器管理的鑑別別名**清單中，選取相同的名稱。
 - i) 在摘要頁面上按一下**完成**，以建立資料來源。
4. 選擇性的：驗證資料來源的配置：
- a) 從 WebSphere Application Server 7.0 管理主控台導覽中選取 **資源 > JDBC > 資料來源**。
 - b) 按一下**測試連線**按鈕。
5. 建立主題。
- a) 從 WebSphere Application Server 7.0 管理主控台導覽中，按一下**資源 > JMS > 主題**。
 - b) 選取**範圍**下拉清單，然後將範圍變更為適當的值。例如，Node=yourNode，Server=yourServer。
 - c) 按一下**新建**。
 - d) 按一下 **IBM MQ 傳訊提供者**。
 - e) 在主題中「內容」頁的「管理」畫面上，為**名稱及 JNDI 名稱**欄位選擇唯一值；您在後續配置時將參考這些資訊。

- f) 在「**IBM MQ 主題**」畫面的**主題名稱**欄位中，輸入 `SYSTEM.FTE/Log/#`。
6. 建立啟動規格：
- 從 WebSphere Application Server 7.0 管理主控台導覽中，按一下**資源 > JMS > 啟動規格**。
 - 選取**範圍**下拉清單，然後將範圍變更為適當的值。例如，`Node=yourNode, Server=yourServer`。
 - 按一下**新建**。
 - 按一下 **IBM MQ 傳訊提供者**。
 - 在精靈的步驟 1 中，為**名稱**及 **JNDI 名稱**欄位選擇唯一值；您在後續配置時將再次參考這些資訊。
 - 在步驟 1.1 中，將您在步驟 5 中所設定主題的 JNDI 名稱輸入目的地 **JNDI 名稱**欄位中。
 - 從目的地類型清單中，選取**主題**。
 - 在精靈的步驟 1.2 中，選取**可延續訂閱**。在**訂閱名稱**欄位中，輸入 `SYSTEM.FTE.DATABASELOGGER.AUTO`。
 - 在精靈的步驟 2 中，選取**將所有必要資訊輸入此精靈中**。
 - 在步驟 2.1 中，將您的佇列管理程式名稱輸入**佇列管理程式或佇列共用群組名稱**欄位中。
 - 在步驟 2.2 中，從**傳輸**清單中選取您所選擇的傳輸方法。如果您選取**連結**，則不需要提供其他資訊。如果您選取**用戶端**或**先連結後用戶端**，請輸入主機名稱、埠及伺服器連線通道的詳細資料。
 - 選擇性的：按一下**測試連線**以確認佇列管理程式是否存在。不過，在步驟 6n 中參照鑑別別名之前，您應該會收到 `NOT_AUTHORIZED`。
 - 按一下**儲存**。
 - 按一下您建立的「啟動規格」名稱。在**配置**標籤的**一般內容**區段中，向下捲動至「**進階**」畫面，並在**用戶端 ID**欄位中輸入唯一名稱以識別您的 IBM MQ 連線。您必須完成此步驟，否則 IBM MQ 會拒絕您的連線，並顯示 `JMSCC0101` 錯誤碼。
 - 如果您選擇**用戶端**作為傳輸方法，請向下捲動至「**安全設定**」畫面，然後從**鑑別別名**清單中選取您在步驟 8 中定義的鑑別別名。
 - 按一下**套用**。
 - 在**配置**標籤的「**其他內容**」區段中，按一下**進階內容**。在「**進階內容**」畫面的「**連線消費者**」區段中，在**伺服器階段作業數上限**欄位中輸入 1。
註：請確保完成此步驟，然後再繼續。若未完成此步驟，日誌程式將無法正確運作。
 - 在**配置**標籤的「**其他內容**」區段中，按一下**進階內容**。將**如果訊息遞送失敗即停止端點**的值設為最小值 1。

如果 `_numberOfFailedAttemptsBeforeReject` 內容的值設為大於 1 的值（如需相關資訊，請參閱 9j），則**如果訊息遞送失敗即停止端點**至少應設為 `_numberOfFailedAttemptsBeforeReject` 內容的值。這可以在接收到無法處理的訊息（例如，形態異常的傳送日誌訊息）時防止端點停止。如需相關資訊，請參閱 [MFT 日誌程式錯誤處理及拒絕](#)。
7. 建立 Queue Connection Factory。
- 從 WebSphere Application Server 7.0 管理主控台導覽中，按一下**資源 > JMS > Queue Connection Factory**。
 - 選取**範圍**下拉清單，然後將範圍變更為適當的值。例如，`Node=yourNode, Server=yourServer`。
 - 按一下**新建**。
 - 按一下 **IBM MQ 傳訊提供者**。
 - 在精靈的步驟 1 中，為**名稱**及 **JNDI 名稱**欄位選擇唯一值；您在後續配置時將再次參考這些資訊。
 - 在步驟 2 中，選取**將所有必要資訊輸入此精靈中**。
 - 在步驟 2.1 中，將您的佇列管理程式名稱輸入**佇列管理程式或佇列共用群組名稱**欄位中。
 - 在步驟 2.2 中，從**傳輸**清單中選取您所選擇的傳輸方法。如果您選取**連結**，則不需要提供其他資訊。如果您選取**用戶端**或**先連結後用戶端**，請輸入主機名稱、埠及伺服器連線通道的詳細資料。

- i) 選擇性的: 按一下**測試連線**以確認佇列管理程式是否存在。不過, 在步驟 7h 中參照鑑別別名之前, 您應該會收到 NOT_AUTHORIZED。
 - j) 如果您選取**用戶端**或**先連結後用戶端**作為傳輸方法, 請按一下您剛才建立的 Queue Connection Factory 名稱。向下捲動至**配置**標籤的「安全設定」畫面, 然後從 **XA** 回復的鑑別別名及儲存器管理的鑑別別名清單中, 選取您在步驟 2e 中定義的鑑別別名。
8. 在 WebSphere Application Server 中建立拒絕佇列:
- a) 從 WebSphere Application Server 7.0 管理主控台導覽中, 按一下**資源 > JMS > 佇列**。
 - b) 選取**範圍**下拉清單, 然後將範圍變更為適當的值。例如, Node=yourNode, Server=yourServer。
 - c) 按一下**新建**。
 - d) 按一下 **IBM MQ 傳訊提供者**。
 - e) 為**名稱**及 **JNDI 名稱**欄位選擇唯一值; 您在後續配置時將再次參考這些資訊。
 - f) 在 **佇列名稱** 欄位中輸入 SYSTEM.FTE.LOG.RJCT.logger_name。請確定您已在協調佇列管理程式上建立此佇列。
 - g) 在**佇列管理程式名稱**欄位中, 輸入您的佇列管理程式名稱。
 - h) 按一下**確定**。
9. 安裝 JEE 資料庫日誌程式應用程式:
- a) 從 WebSphere Application Server 7.0 管理主控台, 選取**應用程式 > 新建應用程式**。
 - b) 選取**範圍**下拉清單, 然後將範圍變更為適當的值。例如, Node=yourNode, Server=yourServer。
 - c) 從選項清單中, 選取**新建企業應用程式**。
 - d) 在「**準備安裝應用程式**」頁面上, 從 Managed File Transfer Service 安裝架構的 MQ_INSTALLATION_PATH/mqft/web 目錄中選取 com.ibm.wmqfte.databaselogger.jee.ear 檔或 com.ibm.wmqfte.databaselogger.jee.oracle.ear 檔, 然後按**下一步**。
 - e) 在下列畫面上, 選取**詳細**以顯示所有安裝選項及參數, 然後按**下一步**。
 - f) 在精靈步驟 1 至 4 中按**下一步**, 以接受預設值。
 - g) 在精靈的步驟 5 **連結訊息驅動 Bean 的接聽器**中, 捲動至**接聽器連結**區段。按一下**啟動規格**。
輸入下列欄位的必要值:
目標資源 JNDI 名稱
您在步驟 6d 中建立啟動規格時所指定的 JNDI 名稱。
目的地 JNDI 名稱
您在步驟 5d 中建立主題時所指定的 JNDI 名稱。
按**下一步**。
 - h) 在精靈的步驟 6 **將資源參照對映至資源**中, 在**目標資源 JNDI 名稱**欄位中輸入詳細資料。此名稱是您在步驟 7c 中為拒絕 Queue Connection Factory 指定的 JNDI 名稱。按**下一步**。
 - i) 在精靈的步驟 7 **將資源環境項目參照對映至資源**中, 在**目標資源 JNDI 名稱**欄位中輸入詳細資料。此名稱是您在步驟 8d 中所建立的拒絕佇列的 JNDI 名稱。按**下一步**。
 - j) 在精靈的步驟 8 **對映 EJB 模組的環境項目**中, 接受預設值 1。按**下一步**。
Oracle RAC: 連接至 Oracle Real Application Cluster 時, 您必須將 _numberOfFailedAttemptsBeforeReject 內容的值至少設為 2。此內容決定日誌程式在失敗之後嘗試處理審核訊息的次數。在資料庫失效接手的情況下, 通常至少發生一次失敗。為避免無謂地將訊息移至拒絕佇列, 您可以增加此值以容許第二次嘗試, 如此通常即可在新資料庫實例的連線建立時成功處理。如果您在測試期間發現訊息在資料庫實例的失效接手期間仍移至拒絕佇列, 請進一步提高此值: 實例切換的時機若不湊巧, 相同訊息就可能失敗多次。不過, 請注意, 提高此值將會對所有失敗案例造成影響 (例如, 形態異常的訊息), 而非只影響資料庫失效接手, 因此請謹慎地提高此值, 以避免發生不必要的重試。
 - k) 在精靈的步驟 9 **模組的 meta 資料**中, 按**下一步**。

- l) 在精靈的步驟 10 **摘要**中，按一下**完成**。
10. 您現在可以從 WebSphere Application Server 7.0 管理主控台中啟動該應用程式：
- 從主控台導覽中，選取**應用程式 > 應用程式類型 > WebSphere 企業應用程式**。
 - 從收集表格中選取**日誌程式企業應用程式**的勾選框，然後按一下**啟動**。

使用 *WebSphere Application Server Community Edition* 安裝 *MFT* 的 *Java EE* 資料庫日誌程式

遵循下列指示，使用 WebSphere Application Server Community Edition 來安裝及配置 Managed File Transfer 的 Java Platform, Enterprise Edition (Java EE) 資料庫日誌程式。

開始之前

安裝 Java EE 資料庫日誌程式應用程式之前，請遵循第 618 頁的『準備安裝 MFT 的 Java EE 資料庫日誌程式』主題中的指示。

關於這項作業

如需 Java EE 資料庫日誌程式的相關資訊，請參閱第 606 頁的『配置 MFT 日誌程式』主題。

程序

1. 部署 IBM MQ 資源配接器 `wmq.jmsra.rar`。

- 若要使用協調佇列管理程式 `QM_JUPITER` 為 Java EE 資料庫日誌程式部署 IBM MQ 資源配接器，請執行下列步驟。如果 WebSphere Application Server Community Edition 實例與您要連接的 IBM MQ 佇列管理程式皆在相同系統上執行，則適用此範例。
 - 建立計劃檔案以定義 MFT 協調佇列管理程式的連線。下列範例計劃檔案會定義稱為 `QM_JUPITER` 的佇列管理程式連線，以及該佇列管理程式上稱為 `SYSTEM.FTE.LOG.RJCT.LOGGER1` 的佇列參照。

```
<?xml version="1.0" encoding="UTF-8"?>
<connector xmlns="http://geronimo.apache.org/xml/ns/j2ee/connector">
  <resourceadapter>
    <resourceadapter-instance>
      <resourceadapter-name>WMQ</resourceadapter-name>
      <workmanager>
        <gbean-link>DefaultWorkManager</gbean-link>
      </workmanager>
    </resourceadapter-instance>
  </resourceadapter>
  <outbound-resourceadapter>
    <connection-definition>
      <connectionfactory-interface>javax.jms.ConnectionFactory</connectionfactory-interface>
      <connectiondefinition-instance>
        <name>jms/WMQFTEJEEEDBLoggerRejectQueueCF</name>
        <config-property-setting name="queueManager">QM_JUPITER</config-property-setting>
        <config-property-setting name="transportType">BINDINGS</config-property-setting>
        <connectionmanager>
          <xa-transaction>
            <transaction-caching/>
          </xa-transaction>
          <single-pool>
            <max-size>10</max-size>
            <min-size>1</min-size>
            <blocking-timeout-milliseconds>5000</blocking-timeout-milliseconds>
            <idle-timeout-minutes>2</idle-timeout-minutes>
            <match-all />
          </single-pool>
        </connectionmanager>
      </connectiondefinition-instance>
    </connection-definition>
  </outbound-resourceadapter>
</resourceadapter>
<adminobject>
  <adminobject-interface>javax.jms.Queue</adminobject-interface>
  <adminobject-class>com.ibm.mq.connector.outbound.MQQueueProxy</adminobject-class>
  <adminobject-instance>
    <message-destination-name>jms/WMQFTEJEEEDBLoggerRejectQueue</message-destination-name>
    <config-property-setting name="baseQueueManagerName">QM_JUPITER</config-property-setting>
    <config-property-setting name="baseQueueName">SYSTEM.FTE.LOG.RJCT.LOGGER1</config-property-setting>
  </adminobject-instance>
</adminobject>
</connector>
```



```
</adminobject>  
</connector>
```

若要在您的環境中使用此計劃檔案，請將 QM_JUPITER 變更為您的協調佇列管理程式的名稱。

- b. 開啟 WebSphere Application Server CE 管理主控台。
 - c. 從「歡迎使用頁面」上的共用主控台動作清單中，按一下**部署新的應用程式 > 新部署**。
 - d. 在**保存**欄位中，輸入 `mq_install_root/java/lib/jca/wmq.jmsra.rar`
 - e. 在**計劃**欄位中，鍵入您在步驟 1a 中建立的計劃檔案的路徑。
- 如果 WebSphere Application Server Community Edition 實例與您要連接的 IBM MQ 佇列管理程式不在相同系統上執行，請執行下列步驟以部署 IBM MQ 資源配接器。
 - a. 建立一個計劃檔案，以定義 WMQFTE 協調佇列管理程式的連線。下列範例計劃檔案定義與佇列管理程式 QM_SATURN 的連線，該佇列管理程式位於與 WebSphere Application Server Community Edition 安裝不同的系統上，並定義稱為 SYSTEM.FTE.LOG.RJCT.LOGGER1。QM_SATURN 的主機名稱是 saturn.example.com。QM_SATURN 的埠是 1415。QM_SATURN 的通道是 SYSTEM.DEF.SVRCONN。

因為應用程式伺服器與佇列管理程式位於不同的系統上，所以您必須使用用戶端模式的佇列管理程式連線。下列計劃檔案會將具有名稱 `transportType` 的 `<config-property-setting>` 元素值設為 `CLIENT`。

```
<?xml version="1.0" encoding="UTF-8"?>  
<connector xmlns="http://geronimo.apache.org/xml/ns/j2ee/connector">  
  <resourceadapter>  
    <resourceadapter-instance>  
      <resourceadapter-name>WMQ</resourceadapter-name>  
      <workmanager>  
        <gbean-link>DefaultWorkManager</gbean-link>  
      </workmanager>  
    </resourceadapter-instance>  
    <outbound-resourceadapter>  
      <connection-definition>  
        <connectionfactory-interface>javax.jms.ConnectionFactory</connectionfactory-interface>  
        <connectiondefinition-instance>  
          <name>jms/WMQFTEJEEEDBLoggerRejectQueueCF</name>  
          <config-property-setting name="queueManager">QM_SATURN</config-property-setting>  
          <config-property-setting name="transportType">CLIENT</config-property-setting>  
          <config-property-setting name="channel">SYSTEM.DEF.SVRCONN</config-property-setting>  
          <config-property-setting name="hostName">saturn.example.com</config-property-setting>  
          <config-property-setting name="port">1415</config-property-setting>  
          <connectionmanager>  
            <xa-transaction>  
              <transaction-caching/>  
            </xa-transaction>  
            <single-pool>  
              <max-size>10</max-size>  
              <min-size>1</min-size>  
              <blocking-timeout-milliseconds>5000</blocking-timeout-milliseconds>  
              <idle-timeout-minutes>2</idle-timeout-minutes>  
              <match-all />  
            </single-pool>  
          </connectionmanager>  
        </connectiondefinition-instance>  
      </connection-definition>  
    </outbound-resourceadapter>  
  </resourceadapter>  
</adminobject>  
<adminobject>  
  <adminobject-interface>javax.jms.Queue</adminobject-interface>  
  <adminobject-class>com.ibm.mq.connector.outbound.MQQueueProxy</adminobject-class>  
  <adminobject-instance>  
    <message-destination-name>jms/WMQFTEJEEEDBLoggerRejectQueue</message-destination-name>  
    <config-property-setting name="baseQueueManagerName">QM_SATURN</config-property-setting>  
    <config-property-setting name="baseQueueName">SYSTEM.FTE.LOG.RJCT.LOGGER1</config-property-setting>  
  </adminobject-instance>  
</adminobject>  
</connector>
```

若要在您的環境中使用此計劃檔案，請將 QM_SATURN 變更為您的協調佇列管理程式的名稱。請將主機名稱、埠及通道的值，變更為您的協調佇列管理程式的值。

- b. 將 `mq_install_root/java/lib/jca/wmq.jmsra.rar` 檔案從 IBM MQ 安裝所在的系統，複製到 WebSphere Application Server CE 安裝所在的系統。
 - c. 開啟 WebSphere Application Server CE 管理主控台。
 - d. 從「歡迎使用頁面」上的共用主控台動作清單中，按一下部署新的應用程式 > 新部署。
 - e. 在保存欄位中，鍵入您所取得的 `wmq.jmsra.rar` 檔案副本的路徑。
 - f. 在計劃欄位中，鍵入您所建立的計劃檔案的路徑。
2. 您必須定義資料庫連接器，讓 Java EE 資料庫日誌程式應用程式可以從 WebSphere Application Server Community Edition 環境內存取所需的資料庫。
- 從 WebSphere Application Server Community Edition 管理主控台中執行下列步驟：
- a) 視您使用的 WebSphere Application Server Community Edition 層次而定，從主控台導覽中，選取服務 > 資料庫儲存區，或選取資源 > 資料來源。
 - b) 使用 Geronimo 資料庫儲存區精靈，建立資料庫儲存區。在資料庫儲存區的名稱欄位中，鍵入 `jdbc/wmqfte-database`。
 - c) 根據您的資料庫，將 DB2 XA 或 Oracle Thin 選取為資料庫類型。
 - d) 按下一步。
 - e) 在驅動程式 jar 欄位中，為您的資料庫選取適當的 jar。
 - f) 在資料庫名稱欄位中，鍵入您所連接以取得傳送狀態資訊的資料庫名稱。
 - g) 在使用者名稱欄位中，鍵入您在連接資料庫以及對其進行鑑別時所使用的使用者名稱。
 - h) 在密碼及確認密碼欄位中，鍵入對您的資料庫進行鑑別時所使用的密碼。
 - i) 在埠號欄位中，鍵入您所使用的埠號（如果不是預設埠）。
 - j) 確認驅動程式類型的值是 4。
 - k) 從交易類型清單中選取 XA。
 - l) 按一下部署。
3. 針對您的環境更新 Managed File Transfer Java EE 資料庫日誌程式應用程式 `openejb-jar.xml` 檔案。請使用 Java SDK Jar 公用程式來完成下列步驟：
- a) 執行下列指令，從提供的 EAR 檔解壓縮 EJB JAR 檔：

```
jar -xf ear_file_name com.ibm.wmqfte.databaselogger.jee.ejb.jar
```

其中 `ear_file_name` 是 `com.ibm.wmqfte.databaselogger.jee.ear` 或 `com.ibm.wmqfte.databaselogger.jee.oracle.ear`，視您使用的是 Db2 還是 Oracle 而定。EAR 檔位於 IBM WebSphere MQ File Transfer Edition Server 安裝架構的 `MQ_INSTALLATION_PATH/mqft/web` 目錄中。

- b) 執行下列指令，從先前解壓縮的 EJB JAR 檔 `com.ibm.wmqfte.databaselogger.jee.ejb.jar` 中解壓縮 `META-INF/openejb-jar.xml` 檔：

```
jar -xf com.ibm.wmqfte.databaselogger.jee.ejb.jar META-INF/openejb-jar.xml
```

- c) 使用文字編輯器，對解壓縮的 `META-INF/openejb-jar.xml` 檔進行編輯。將下列 `activation-config-property` 值變更為您的環境所適用的值：

queueManager

Java EE 資料庫日誌程式使用的 IBM MQ 佇列管理程式名稱。

hostName

用來連接至指定的 IBM MQ 佇列管理程式的主機名稱。如果您已在連結模式下連接至佇列管理程式，則不需要此值。

transportType

以用戶端模式還是連結模式連接至指定的 IBM MQ 佇列管理程式。

port

如果您已將 **transportType** 指定為連結，則不需要此項目。用來連接至指定的 IBM MQ 佇列管理程式的埠。

channel

如果您已將 **transportType** 指定為連結，則不需要此項目。用來連接至指定的 IBM MQ 佇列管理程式的伺服器通道。

- d) 執行下列指令，以修改過的 META-INF/openejb-jar.xml 檔更新 EJB JAR 檔：

```
jar -uf com.ibm.wmqfte.databaselogger.jee.ejb.jar META-INF/openejb-jar.xml
```

- e) 執行下列指令，以更新過的 EJB JAR 檔更新提供的 EAR 檔：

```
jar -uf ear_file_name com.ibm.wmqfte.databaselogger.jee.ejb.jar
```

其中 *ear_file_name* 是 `com.ibm.wmqfte.databaselogger.jee.ear` 或 `com.ibm.wmqfte.databaselogger.jee.oracle.ear`，視您的資料庫而定。

4. 若要將 EAR 檔部署至應用程式伺服器，請從 WebSphere Application Server Community Edition 管理主控台完成下列步驟。
- 從主控台導覽功能表中，選取應用程式 > 新部署。
 - 在保存欄位中指定 EAR 檔：`com.ibm.wmqfte.databaselogger.jee.ear` 或 `com.ibm.wmqfte.databaselogger.jee.oracle.ear`，視您的資料庫而定。
 - 將計劃欄位保留為空白。
 - 確定已選取在安裝後啟動應用程式方框。
 - 按一下安裝。JEE 資料庫日誌程式應用程式即會安裝並且啟動。

配置 MFT 的 Java EE 資料庫日誌程式的使用者存取權

當您配置 Managed File Transfer 的 Java Platform, Enterprise Edition (Java EE) 資料庫日誌程式時，您需要使用者帳戶才能存取 IBM MQ、資料庫及作業系統。所需的作業系統使用者數目，取決於您用來管理這些元件的系統數目。

關於這項作業

執行 Java EE 資料庫日誌程式所需的使用者帳戶的數目及類型，取決於您使用的系統數目。使用者帳戶需要存取下列三種環境：

- 本端作業系統
- IBM MQ
- 資料庫

您可以將 JEE 資料庫日誌程式、IBM MQ 及您的資料庫安裝在單一系統上，或安裝在數個系統上。元件可安裝在下列範例拓撲中：

Java EE 資料庫日誌程式、IBM MQ 及資料庫全都在同一系統上

您可以定義單一作業系統使用者，以與這三項元件搭配使用。日誌程式使用「連結」模式連接至 IBM MQ，連接至資料庫時則使用原生連線。

Java EE 資料庫日誌程式和 IBM MQ 位於一個系統上，資料庫則位於另一個系統上

您可以為此配置建立兩個使用者：在執行日誌程式的系統上配置一個作業系統使用者，以及在資料庫伺服器上配置一個具有資料庫遠端存取權的作業系統使用者。日誌程式會使用「連結」模式連接至 IBM MQ，存取資料庫時則使用用戶端連線。

Java EE 資料庫日誌程式位於一個系統上，IBM MQ 在另一個系統上，資料庫又在另一個系統上

您可以為此配置建立三個使用者：一個作業系統使用者負責啟動應用程式伺服器、一個 IBM MQ 使用者負責存取要使用的佇列及主題，一個資料庫伺服器使用者負責存取及插入至資料庫表格。日誌程式會使用「用戶端」模式存取 IBM MQ，並使用用戶端連線來存取資料庫。

為方便解說，本文的其餘指示均假設使用者的名稱為 `ftelog`，但您可以使用任何使用者名稱，新建或現有的都可以。請如下所示配置使用者的權限：

程序

1. 確定作業系統使用者具有專屬的群組，且不屬於任何對協調佇列管理程式具備多項權限的群組。使用者不得屬於 `mqm` 群組。在特定平台上，也會對工作人員群組自動指定佇列管理程式存取權；因此日誌程式使用者不應屬於工作人員群組。您可以使用「IBM MQ Explorer」，檢視佇列管理程式本身及其內含物件的權限記錄。用滑鼠右鍵按一下物件，然後選取**物件權限 > 管理權限記錄**。在指令行上，您可以使用 `dspmqaout`（顯示權限）或 `dmpmqaut`（傾出權限）指令。
2. 使用 IBM MQ Explorer 中的「**管理權限記錄**」視窗或 `setmqaut`（授與或撤銷權限）指令來新增 IBM MQ 使用者專屬群組的權限（在 UNIX 上，IBM MQ 權限僅與群組相關聯，而與個別使用者無關）。必要權限如下所示：
 - 對佇列管理程式的 `CONNECT` 及 `INQUIRE`（IBM MQ Java 程式庫需要 `INQUIRE` 權限才能運作）。
 - 對 `SYSTEM.FTE` 主題的 `SUBSCRIBE` 權限。
 - 對 `SYSTEM.FTE.LOG.RJCT.logger_name` 佇列的 `PUT` 權限。

指定的拒絕及指令佇列名稱為預設名稱。如果您在配置日誌程式佇列時選擇了不同的佇列名稱，請改為將權限新增至那些佇列名稱。

3. 執行所使用的資料庫專用的資料庫使用者配置。
 - 如果您的資料庫是 Db2，請執行下列步驟：

註：使用 Db2 時可透過數種機制來管理資料庫使用者。下列指示適用於以作業系統使用者為基礎的預設機制。

 - 確保 `ftelog` 使用者不屬於任何 Db2 管理群組（例如，`db2iadm1`、`db2fadm1` 或 `dasadm1`）。
 - 授與使用者連接至資料庫的許可權，以及對您在 **步驟 2: 建立必要的資料庫表格** 中建立的表格進行選取、插入及更新的許可權。
 - 如果您的資料庫是 Oracle，請執行下列步驟：
 - 確保 `ftelog` 使用者不在任何 Oracle 管理群組中（例如，Windows 上的 `ora_dba` 或 UNIX 上的 `dba`）。
 - 授與使用者連接至資料庫權限，以及對 **步驟 2: 建立所需資料庫表格** 中所建立表格的選取、插入及更新權限。

移轉 Java EE 資料庫日誌程式

若要將 WebSphere Application Server 7.0 上的 Java EE 資料庫日誌程式從 IBM WebSphere MQ File Transfer Edition 7.0 移轉至 IBM WebSphere MQ 7.5 或更新版本，請完成下列步驟：

程序

1. 開啟 WebSphere Application Server 主控台。
2. 按一下 **應用程式 > 應用程式類型 > 企業應用程式**。從應用程式清單中找出 IBM WebSphere MQ File Transfer Edition 資料庫日誌程式應用程式。如果資料庫日誌程式應用程式尚未停止，請選取該應用程式，然後按一下 **停止**。
3. 記下您先前對 JEE 資料庫日誌程式設定的配置設定。稍後在步驟 [第 629 頁的『7』](#) 中您會需要這些設定。
 - a) 若您最初在安裝資料庫日誌程式時，變更了 EJB 模組的預設值（如需相關資訊，請參閱步驟 9），請按一下 **企業應用程式 > WebSphere MQ File Transfer Edition 資料庫日誌程式 > EJB 模組的環境項目**，並記下該窗格中的設定。
 - b) 按一下 **企業應用程式 > WebSphere MQ File Transfer Edition 資料庫日誌程式 > 訊息驅動 Bean 接聽器連結**，並記下所使用的啟動規格，即目標資源 **JNDI** 名稱和目的地 **JNDI** 名稱。
 - c) 按一下 **企業應用程式 > WebSphere MQ File Transfer Edition 資料庫日誌程式 > 資源參照**，並記下拒絕 Queue Connection Factory 詳細資料。


- d) 按一下**企業應用程式 > WebSphere MQ File Transfer Edition 資料庫日誌程式 > 資源環境項目參照**，並記下拒絕佇列詳細資料。
- 按一下 **應用程式 > 應用程式類型 > 企業應用程式**，以解除安裝 IBM WebSphere MQ File Transfer Edition 資料庫日誌程式應用程式。選取資料庫日誌程式應用程式，並按一下**解除安裝**。
 - 選用項目：如果是使用多重安裝以移轉至 IBM WebSphere MQ 7.5 或更新版本，且原生程式庫路徑不相同，請按一下**資源 > JMS 提供者 > WebSphere MQ 傳訊提供者**以變更路徑。
例如，如果原生程式庫路徑為：C:\Program Files\IBM\WebSphere MQ\java\lib，請將路徑變更為：C:\Program Files\IBM\New MQ Installation Location\java\lib
 - 選用項目：如果是使用多重安裝以移轉至 IBM WebSphere MQ 7.5 或更新版本，則必須使用 `setmqm` 指令將佇列管理程式與新安裝相關聯。
 - 使用第 620 頁的『[使用 WebSphere Application Server 7.0 安裝 MFT 的 Java EE 資料庫日誌程式](#)』中的資訊及您先前在步驟 第 628 頁的『3』中記錄的資訊，重新安裝資料庫日誌程式應用程式。
 - 按一下**應用程式 > 應用程式類型 > 企業應用程式**，來啟動新的資料庫日誌程式。選取資料庫日誌程式應用程式，並按一下**啟動**。
 - 如果要驗證移轉，請檢查資料庫，以確定寫入項目。

從獨立式資料庫日誌程式移轉至 MFT 的 Java EE 資料庫日誌程式

您可以從獨立式資料庫日誌程式移轉至 Java EE 資料庫日誌程式。必須停止獨立式資料庫日誌程式，然後再安裝 JEE 資料庫日誌程式。若要避免遺失或複製日誌項目，您必須停止將訊息發佈至 SYSTEM.FTE 主題，並在安裝 Java EE 資料庫日誌程式之後重新啟動它。移轉之前，請先備份資料庫。

關於這項作業

程序

- 在停止資料庫之前，請先針對協調佇列管理程式執行以下 MQSC 指令：ALTER QM PSMODE(COMPAT)
這會停止將訊息發佈至 SYSTEM.FTE/Log 主題。靜待日誌程式處理其訂閱的所有訊息。依預設，此訂閱稱為 SYSTEM.FTE.LOGGER.AUTO。
- 使用 `fteStopLogger` 指令停止資料庫日誌程式。
- 使用資料庫軟體隨附的工具備份資料庫。
- 刪除屬於獨立式資料庫日誌程式的訂閱。
依預設，此訂閱稱為 SYSTEM.FTE.LOGGER.AUTO。
- 如果您的資料庫綱目為舊版，則必須按次序將綱目移轉至每個後續層次。比方說，例如資料庫綱目為 7.0.1 版且要移轉至 7.0.4 版，則必須先將綱目從 7.0.1 版移轉至 7.0.2 版，然後從 7.0.2 版移轉至 7.0.3 版，最後從 7.0.3 版移轉至 7.0.4 版。透過針對必須移轉的每個資料庫綱目版本執行下列其中一項動作，可以將該綱目從 *old* 版本移轉至 *new* 版本（其中 *old* 及 *new* 為說明綱目版本的變數）：
 -  **z/OS** 如果資料庫是 z/OS 上的 Db2，並且您要在 V7.0.2 與 V7.0.3 綱目之間或在 V7.0.3 與 V7.0.4 綱目之間進行移轉，則必須建立新的資料庫綱目並將現有資料複製到其中。如需相關資訊，請參閱 [將 z/OS 上 Db2 上的資料庫表格移轉至 MQ V8.0 或更新版本](#)。
 - 如果資料庫不是 Db2，或建立資料庫時的頁面大小超過 8K，您可以透過完成下列步驟，採用與其他版本相同的方式移轉綱目。
 - 在任何其他情況下，如果要在資料庫表格之間進行移轉，請完成下列步驟：
 - 選擇適用於資料庫平台並且名稱中包含字串 *old-new* 的檔案。此檔案位於 Remote Tools and Documentation 安裝的 `MQ_INSTALLATION_PATH/mqft/sql` 目錄中。
 - 如果已對起始綱目進行修改，請檢閱移轉檔案，以確定該檔案與修改後的資料庫相容。
 - 針對您的資料庫執行 SQL 檔案。
- 安裝 Java EE 資料庫日誌程式 EAR 檔。
- 部署 Java EE 資料庫日誌程式。如需相關資訊，請參閱 第 617 頁的『[安裝 MFT 的 Java EE 資料庫日誌程式](#)』。

8. 針對協調佇列管理程式執行以下 MQSC 指令：ALTER QMGR PSMODE(ENABLED)
這會將訊息發佈至 SYSTEM.FTE/Log 主題。

結果

配置 Connect:Direct 橋接器

配置 Connect:Direct 橋接器，以在 Managed File Transfer 網路與 Connect:Direct 網路之間傳送檔案。Connect:Direct 橋接器的元件為 Connect:Direct 節點，以及專門用來與該節點通訊的 Managed File Transfer 代理程式。此代理程式稱為 Connect:Direct 橋接器代理程式。

開始之前

組成 Connect:Direct 橋接器的代理程式及節點必須在同一系統上，或可以存取同一檔案系統（例如透過共用 NFS 裝載）。此檔案系統用於在涉及 Connect:Direct 橋接器的檔案傳送期間，將檔案暫時儲存在 **cdTmpDir** 參數所定義的目錄中。Connect:Direct 橋接器代理程式及 Connect:Direct 橋接器節點，必須能使用相同路徑名稱，對此目錄進行定址。例如，如果代理程式和節點是在不同的 Windows 系統上，則這些系統必須使用相同的磁碟機代號來裝載共用檔案系統。下列配置容許代理程式及節點使用相同路徑名稱：

- 代理程式和節點是在同一個系統上，它可能執行 Windows 或 Linux for x86-64
- 代理程式在 Linux for x86-64 上，節點在 UNIX 上
- 代理程式是在一個 Windows 系統上，而節點是在另一個 Windows 系統上

下列配置不容許代理程式及節點使用相同路徑名稱：

- 代理程式在 Linux for x86-64 上，節點在 Windows 上
- 代理程式是在 Windows 上，而節點是在 UNIX 上

規劃 Connect:Direct 橋接器安裝時，請考量此限制。

如需 Connect:Direct 橋接器支援的作業系統版本的詳細資料，請參閱 [IBM MQ 系統需求網頁](#)。

關於這項作業

Connect:Direct 橋接器代理程式是一種 Managed File Transfer 代理程式，專門用來與 Connect:Direct 節點通訊。


依預設，Connect:Direct 橋接器代理程式會使用 TCP/IP 通訊協定連接至 Connect:Direct 節點。如果要在 Connect:Direct 橋接器代理程式與 Connect:Direct 節點之間建立安全連線，則可使用 SSL 通訊協定或 TLS 通訊協定。

程序

1. 選擇 Connect:Direct 橋接器代理程式及節點的作業系統:
 - a) 選擇執行 Windows 或 Linux on x86-64 的系統，以在其中安裝 Connect:Direct 橋接器代理程式。
 - b) 選擇 Connect:Direct for Windows 或 Connect:Direct for UNIX 支援的作業系統，以在其中安裝 Connect:Direct 橋接器節點。
2. 選擇並配置 Connect:Direct 節點:

在遵循下列指示之前，您必須已安裝 Connect:Direct 節點。


 - a) 選擇要與 Managed File Transfer 代理程式通訊的 Connect:Direct 節點。
 - b) 檢查所選 Connect:Direct 節點的網路對映。如果該網路對映包含在 Windows 作業系統上執行的遠端節點的任何項目，您必須確保這些項目指定節點在 Windows 上執行。

 如果您為 Connect:Direct 橋接器選取的 Connect:Direct 節點是在 Windows 上執行，請使用「Connect:Direct 要求端」來編輯網路對映。確保 Windows 上執行之任何遠端節點的 **作業系統** 欄位設為 **Windows**。
3. 建立並配置 Connect:Direct 橋接器代理程式:

- a) 使用 **fteCreateCDAgent** 指令建立 Connect:Direct 橋接器代理程式。
 - 您必須為 **cdNode** 參數提供值。此參數會指定代理程式用於 Connect:Direct 節點的名稱，而該節點是 Connect:Direct 橋接器的一部分。請使用您在上一節中選擇的 Connect:Direct 節點名稱。
 - 為 **cdNodeHost** 及 **cdNodePort** 參數提供值，這兩個參數可定義與代理程式通訊的 Connect:Direct 節點。

如果您不為 **cdNodeHost** 參數提供值，則會使用本端系統的主機名稱或 IP 位址。如果您不為 **cdNodePort** 參數提供值，則會使用值 1363。
 - 選擇性地使用 [fteCreate](#) 代理程式中的資訊，以判定您是否需要指定 **cdTmpDir** 參數的值。
- b) 將 Managed File Transfer 使用的使用者認證，對映至 Connect:Direct 節點上的使用者認證。您可以使用下列其中一種方法來對映認證：
 - 建立用來定義認證對映資訊的 `ConnectDirectCredentials.xml` 檔案。如需相關資訊，請參閱第 631 頁的『[使用 ConnectDirectCredentials.xml 檔案對映 Connect:Direct 的認證](#)』。
 - 撰寫使用者結束程式，以執行 Connect:Direct 橋接器的認證對映。如需相關資訊，請參閱第 633 頁的『[使用結束程式類別對映 Connect:Direct 的認證](#)』。
4. 配置 `ConnectDirectNodeProperties.xml` 檔案，以包含遠端 Connect:Direct 節點的相關資訊：在遵循下列指示之前，您必須已建立 Connect:Direct 橋接器代理程式。

編輯 Connect:Direct 橋接器代理程式配置目錄中的範本 `ConnectDirectNodeProperties.xml`。針對您要定義其相關資訊的每個 Connect:Direct 節點或節點群組，執行下列步驟：

 - a) 在 `nodeProperties` 元素內，建立 `node` 元素。
 - b) 將 `name` 屬性新增至 `node` 元素。請將此屬性的值，指定為符合一個以上遠端 Connect:Direct 節點名稱的型樣。
 - c) 選擇性的：將 `pattern` 屬性新增至 `node` 元素，以指定 `name` 屬性值中的型樣類型。有效值為 `regex` 及 `wildcard`。預設選項為 `wildcard`。
 - d) 將 `type` 屬性新增至 `node` 元素，以指定執行 `name` 屬性所指定遠端 Connect:Direct 節點的作業系統。下列為有效值：
 - Windows - 節點在 Windows 上執行
 - UNIX - 節點在 UNIX 或 Linux 上執行
 -  z/OS、zos、os/390 或 os390 - 節點在 z/OS 上執行此屬性的值不區分大小寫。Connect:Direct 橋接器不支援傳送至其他作業系統上的遠端節點。
5. 配置 Connect:Direct 橋接器代理程式與 Connect:Direct 節點之間的安全連線。

如需如何執行此動作的範例，請參閱 [在 Connect:Direct 橋接器代理程式與 Connect:Direct 節點之間配置 SSL 或 TLS](#)。

對映 Connect:Direct 的認證

使用 Connect:Direct 橋接器代理程式的預設認證對映功能，或撰寫您自己的使用者結束程式，將 Managed File Transfer 中的使用者認證對映至 Connect:Direct 節點上的使用者認證。Managed File Transfer 提供一個使用者結束程式範例，可執行使用者認證對映。

使用 `ConnectDirectCredentials.xml` 檔案對映 Connect:Direct 的認證

使用 Connect:Direct 橋接器代理程式的預設認證對映功能，將 Managed File Transfer 中的使用者認證對映至 Connect:Direct 節點上的使用者認證。Managed File Transfer 提供 XML 檔案，您可以編輯此檔案以包括自己的認證資訊。

關於這項作業

使用 **fteCreateCDAgent** 指令建立 Connect:Direct 橋接器代理程式之後，需要手動建立 `ConnectDirectCredentials.xml` 檔案。您必須先編輯此檔案以併入主機、使用者及認證資訊，才能使用 Connect:Direct 橋接器代理程式。如需相關資訊，請參閱 [Connect:Direct 認證檔案格式](#)。依預設，此檔案會從現行使用者的起始目錄載入，例如 `/home/fteuser/ConnectDirectCredentials.xml`。若要

使用其他位置，請使用 `ConnectDirectNodeProperties.xml` 檔案中的 `<credentialsFile>` 元素指定該位置。

程序

1. 確保元素 `<tns:pnode name="Connect:Direct node host" pattern="wildcard">` 中的 `name` 屬性包含 `Connect:Direct` 橋接器代理程式所連接之 `Connect:Direct` 節點的名稱值。此值必須是您為 `fteCreateCDAgent -cdNode` 參數指定的同一值。
`pattern` 屬性的值可以是 `wildcard` 或 `regex`。如果未指定此屬性，預設值將為 `wildcard`。
2. 將使用者 ID 及認證資訊，作為 `<tns:pnode>` 的子元素插入檔案中。
您可以將下列 `<tns:user>` 元素的一個以上實例插入檔案中：

```
<tns:user name="name"
          pattern="pattern"
          ignorecase="ignorecase"
          cdUserId="cdUserId"
          cdPassword="cdPassword"
          pnodeUserId="pnodeUserId"
          pnodePassword="pnodePassword">
</tns:user>
```

其中：

- `name` 是符合與 MFT 傳送要求相關聯的 MQMD 使用者 ID 的型樣。
- `pattern` 可指定為 `name` 屬性所指定的型樣，是萬用字元表示式還是 Java 正規表示式。`pattern` 屬性的值可以是 `wildcard` 或 `regex`。如果未指定此屬性，預設值將為 `wildcard`。
- `ignorecase` 可指定，是否區分 `name` 屬性所指定型樣的大小寫。如果未指定此屬性，預設值將為 `true`。
- `cdUserId` 是 `Connect:Direct` 橋接器代理程式用來連接至 `<tns:pnode>` 元素的 `name` 屬性所指定的 `Connect:Direct` 節點的使用者 ID。如有可能，請確保 `cdUserId` 為 `Connect:Direct` 管理者使用者 ID。如果 `cdUserId` 不得為 `Connect:Direct` 管理者，請確保該使用者 ID 在 `Connect:Direct` 橋接器節點上具有下列職能權限：
 - 若為 Windows 節點，請設定下列權限。為了提高可讀性，以下範例使用換行進行格式化：

```
View Processes in the TCQ      value: yes
Issue the copy receive, copy send, run job, and run task
Process statements
Issue the submit Process statement value: yes
Monitor, submit, change, and delete all Processes value: all
Access Process statistics value: all
Use the trace tool or issue traceon and traceoff commands value: yes
Override Process options such as file attributes and remote node ID value: yes
```

- 若為 UNIX 節點，請在 `userfile.cfg` 檔案中設定下列參數：

```
pstmt.copy value: y
pstmt.upload value: y
pstmt.download value: y
pstmt.runjob value: y
pstmt.runtask value: y
cmd.submit value: y
pstmt.submit value: y
```



```

cmd.chgproc          value: y
cmd.delproc          value: y
cmd.flspoc           value: y
cmd.selproc          value: a
cmd.selstats         value: a
cmd.trace            value: y
snode.ovrd           value: y

```

- *cdPassword* 是與 *cdUserId* 屬性所指定使用者 ID 相關聯的密碼。
- 您可以選擇性地指定 *pnodeUserId* 屬性。此屬性的值是由 `<tns:pnode>` 元素的 *name* 屬性所指定的 `Connect:Direct` 節點用來提交 `Connect:Direct` 處理程序的使用者 ID。如果您沒有指定 *pnodeUserId* 屬性，`Connect:Direct` 節點會使用 *cdUserId* 屬性所指定的使用者 ID，來提交 `Connect:Direct` 程序。
- 您可以選擇性地指定 *pnodePassword* 屬性。此屬性的值是與 *pnodeUserId* 屬性所指定使用者 ID 相關聯的密碼。

如果沒有任何使用者元素符合 MQMD 使用者 ID，傳送將會失敗。

3. 選擇性的: 您可以併入一個以上的 `<tns:snode>` 元素，作為 `<tns:user>` 元素的子元素。`<tns:snode>` 元素指定屬於 `Connect:Direct` 橋接器的 `Connect:Direct` 節點所使用的認證。這些認證皆為使用者 ID 及密碼，可供 `Connect:Direct` 橋接器節點，用來連接至作為檔案傳送來源或目的地的 `Connect:Direct` 節點。

請將下列一個或多個元素插入檔案中：

```

<tns:snode name="name"
  pattern="pattern"
  userId="userId"
  password="password" />

```

其中：

- *name* 是符合作為檔案傳送來源或目的地的 `Connect:Direct` 節點名稱的型樣。
- *pattern* 可指定為 *name* 屬性所指定的型樣，是萬用字元表示式還是 Java 正規表示式。*pattern* 屬性的值可以是 *wildcard* 或 *regex*。如果未指定此屬性，預設值將為 *wildcard*。
- *userId* 是由 `<tns:pnode>` 元素的 *name* 屬性所指定的 `Connect:Direct` 節點用來連接至 `Connect:Direct` 節點的使用者 ID，該節點符合 `<tns:snode>` 的 *name* 屬性所指定的型樣。
- *password* 是與 *userId* 屬性所指定使用者 ID 相關聯的密碼。

如果沒有任何 `<tns:snode>` 元素符合檔案傳送的次要節點，這並不會導致傳送失敗。傳送仍會啟動，但不會指定與 *snode* 搭配使用的使用者 ID 及密碼。

結果

在搜尋使用者名稱或 `Connect:Direct` 節點名稱的型樣相符項時，`Connect:Direct` 橋接器代理程式會從檔案開頭開始搜尋到檔案結尾。找到的第一個相符項，即為所用的相符項。

相關工作

第 630 頁的『[配置 Connect:Direct 橋接器](#)』

配置 `Connect:Direct` 橋接器，以在 Managed File Transfer 網路與 `Connect:Direct` 網路之間傳送檔案。

`Connect:Direct` 橋接器的元件為 `Connect:Direct` 節點，以及專門用來與該節點通訊的 Managed File Transfer 代理程式。此代理程式稱為 `Connect:Direct` 橋接器代理程式。

相關參考

[Connect:Direct 認證檔案格式](#)

[fteCreateCDAgent: 建立 Connect:Direct 橋接器代理程式](#)

使用結束程式類別對映 *Connect:Direct* 的認證

如果您不想使用 `Connect:Direct` 橋接器代理程式的預設認證對映功能，則可透過撰寫專屬的使用者結束程式，將 Managed File Transfer 中的使用者認證，對映至 `Connect:Direct` 節點上的使用者認證。配置專屬的認證對映使用者結束程式，將會停用預設認證對映功能。

關於這項作業

為對映 Connect:Direct 認證而建立的使用者結束程式，必須實作 `com.ibm.wmqfte.exitroutine.api.ConnectDirectCredentialExit` 介面。如需相關資訊，請參閱 [CDCredentialExit.java](#) 介面。

Windows 使用 MSCS 配置 MFT 代理程式

如果平台受 MFT 支援且執行其中一個 Windows 版本，則支援 Managed File Transfer (MFT) 代理程式 Microsoft Cluster Service (MSCS) 設定。

關於這項作業

此作業說明兩個實務範例，您可以遵循這些實務範例來實現 MFT 代理程式的失效接手：

- 實務範例 1: 將代理程式配置為 MSCS 資源。
- 實務範例 2: 將代理程式佇列管理程式及代理程式配置為 MSCS 資源。

程序

實務範例 1: 將代理程式配置為 MSCS 資源

- 若要將代理程式配置為 MSCS 資源，請完成下列步驟：
 - a) 在叢集中的每個機器本端安裝 Managed File Transfer。
請參閱 [安裝 Managed File Transfer](#)。
 - b) 在叢集中的主要機器上建立代理程式。
代理程式應該配置為使用 CLIENT 傳輸連接至代理程式佇列管理程式。請確定您在佇列管理程式上建立此代理程式的所有物件。如需如何執行此動作的相關資訊，請參閱 [設定代理程式](#)。
 - c) 透過將 Windows 服務工具中代理程式服務的 **啟動類型** 欄位設為 **手動**，修改代理程式以作為 Windows 服務執行，並將其配置為在 Windows 重新啟動時不會自動啟動。
如需相關資訊，請參閱 [以 Windows 服務方式啟動 MFT 代理程式](#)。
 - d) 在次要機器上重複實務範例 1 的步驟 [第 634 頁的『2』](#) 及步驟 [第 634 頁的『3』](#)。
這可確保日誌、內容等的檔案結構存在於叢集中的其他機器上。請注意，不需要如步驟 [第 634 頁的『2』](#) 一樣建立佇列管理程式物件。
 - e) 在主要機器上，將代理程式新增為 MSCS 控制下的「一般服務」。
若要執行此作業：
 - a. 用滑鼠右鍵按一下叢集，然後選取 **角色->新增資源->「一般服務」**。
 - b. 從 Windows 服務清單中，選取代理程式服務，然後按 **下一步** 來完成配置精靈。代理程式服務現在已新增為 MSCS 資源。如果發生失效接手，則會在另一部機器上啟動代理程式服務。

實務範例 2: 將代理程式佇列管理程式及代理程式配置為 MSCS 資源

- 若要將代理程式佇列管理程式及代理程式配置為 MSCS 資源，請完成下列步驟：
 - a) 將代理程式佇列管理程式配置成作為 MSCS 資源執行。
如需如何執行此動作的相關資訊，請參閱 [第 391 頁的『將佇列管理程式置於 MSCS 控制下』](#)。
 - b) 在叢集中的主要機器上建立代理程式。
代理程式應該配置為使用 BINDINGS 傳輸連接至代理程式佇列管理程式。請確定您在佇列管理程式上建立此代理程式的所有物件。如需如何執行此動作的相關資訊，請參閱 [設定代理程式](#)。
 - c) 透過將 Windows 服務工具中代理程式服務的 **啟動類型** 欄位設為 **手動**，修改代理程式以作為 Windows 服務執行，並將其配置為在 Windows 重新啟動時不會自動啟動。
如需相關資訊，請參閱 [以 Windows 服務方式啟動 MFT 代理程式](#)。
 - d) 請確定代理程式佇列管理程式 (在 MSCS 控制下) 在次要機器上執行。

在此機器上建立的代理程式將使用 BINDINGS 傳輸來連接至佇列管理程式，因此在建立代理程式時需要可用。

- e) 在次要機器上重複實務 2 的步驟 [第 634 頁的『2』](#) 和步驟 [第 634 頁的『3』](#)。
這可確保日誌、內容等的檔案結構存在於叢集中的其他機器上。請注意，不需要如步驟 [第 634 頁的『2』](#) 一樣建立佇列管理程式物件。
- f) 將代理程式新增為 MSCS 控制下的「通用服務」。
若要執行此作業：
 - a. 用滑鼠右鍵按一下叢集，然後選取 **角色-> 新增資源-> 「一般服務」**。
 - b. 從 Windows 服務清單中，選取代理程式服務，然後按 **下一步** 來完成配置精靈。
- g) 修改代理程式服務的資源內容，以在相依關係清單中新增佇列管理程式資源。
這可確保在啟動代理程式之前啟動佇列管理程式資源。
- h) 讓佇列管理程式資源離線，然後讓代理程式資源連線。請驗證佇列管理程式資源和代理程式是否都已啟動。
如果發生失效接手，則會在次要機器上啟動代理程式服務及代理程式佇列管理程式。

V 9.1.4 Managed File Transfer 中的高可用性代理程式

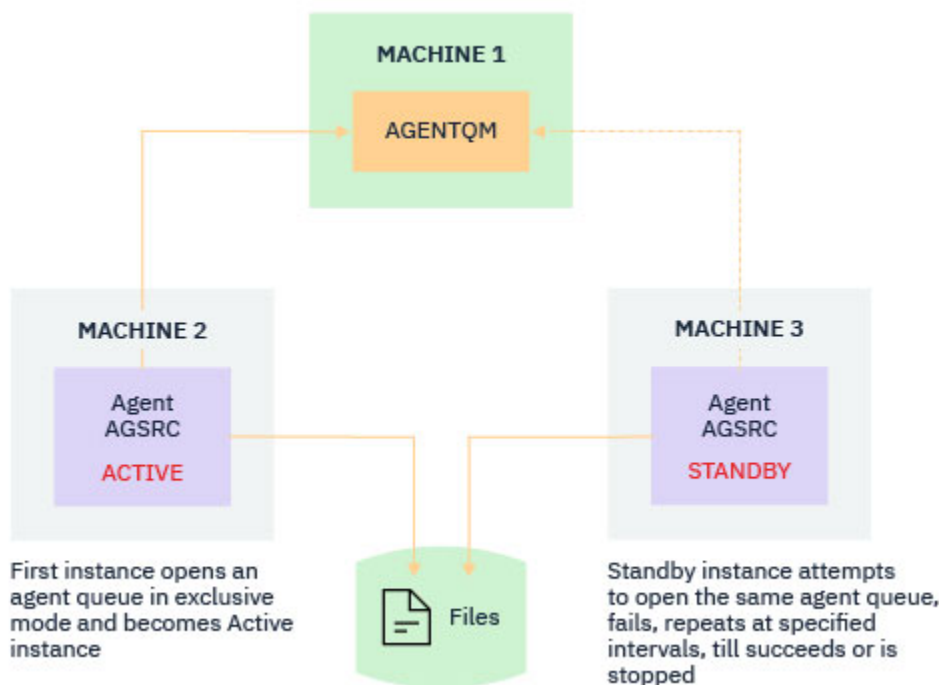
從 IBM MQ 9.1.4，您可以在 MFT 中配置標準或橋接器代理程式，以在高可用性 (HA) 配置中執行。HA 設定中涉及一對具有相同配置的代理程式實例，其中一個實例在一部機器上執行，另一個實例在另一部機器上執行。這兩個實例都配置為連接至相同的代理程式佇列管理程式。

概觀

在兩個實例中，只有一個 (稱為 作用中實例) 正在處理檔案傳送，而另一個 (稱為 待命實例) 則處於可完成其起始設定並接管執行檔案傳送的狀態，但不會處理任何檔案傳送。

當作用中實例失敗或失去與佇列管理程式的連線功能時，待命實例會完成其起始設定，變成作用中，並開始處理檔案傳送。作用中實例失敗時正在進行的任何進行中傳送都會從前次已知檢查點回復。

下圖顯示作用中及待命代理程式的一般配置：



附註：

1. 代理程式的一個實例正在兩個不同的機器上執行，其中一個實例作為作用中實例，另一個實例作為待命實例。
2. 僅作用中實例處理程序傳送。待命實例處於閒置狀態，等待作用中實例關閉。
3. 在代理程式的兩個實例之間共用同一組代理程式佇列。
4. 代理程式的兩個實例都需要存取相同的共用檔案系統，才能執行受管理傳送

作用中-待命代理程式實例機制的運作方式是鎖定共用資源，例如 IBM MQ 多重實例佇列管理程式。在共用資源上取得鎖定的代理程式實例會變成作用中實例，而另一個實例 (無法取得鎖定) 會變成待命實例。

這裡的共用資源是新的佇列 `SYSTEM.FTE.HA.<agent name>`。配置 IBM MQ 9.1.4 代理程式時，會自動建立此佇列。

處理程序運作方式

若要建立 HA 代理程式，您可以在兩部機器上建立具有相同配置參數的代理程式，方法是使用其他 `-x` 參數來執行 `fteCreateAgent` 或 `fteCreateBridgeAgent` 指令，並將 `agent.properties` 檔案中的 `highlyAvailable` 代理程式內容設為 `true`。

附註:

- 這兩個配置必須指向相同的代理程式佇列管理程式。
- 在代理程式佇列管理程式上，必須只建立必要的代理程式佇列一次。

如需 `-x` 參數及 `agent.properties` 檔案的相關資訊，請參閱 `fteCreateAgent` 指令，以取得 `highlyAvailable` 代理程式內容的相關資訊。

請注意，不論您是否指定 `-x` 參數，執行任一指令都會建立一個 MQSC 檔，其中包含在代理程式佇列管理程式及 `<SYSTEM.FTE.HA.agent name` 佇列中建立 IBM MQ 物件所需的 Script。

建立高可用性代理程式配置時，`fteCreateAgent` 或 `fteCreateBridgeAgent` 指令會訂閱 `SYSTEM.FTE/Agents/agent name` 主題，以檢查其他位置是否存在相同代理程式的實例。如果找到相同代理程式的實例，則任一指令會在檔案系統上建立必要的配置，但不會重新發佈代理程式的建立。

當代理程式以 HA 模式啟動時:

1. 代理程式會嘗試以專用 GET 模式開啟 `SYSTEM.FTE.HA.agent name` 佇列。
2. 如果代理程式順利開啟 `SYSTEM.FTE.HA.agent name` 佇列，則這會變成代理程式的作用中實例，並繼續進一步啟動處理程序。
3. 如果嘗試以專用 GET 模式開啟 `SYSTEM.FTE.HA.agent name` 佇列失敗，並出現 `MQRC_OBJECT_IN_USE` 原因碼，則表示已在其他位置執行代理程式的作用中實例。因此，此實例會變成代理程式的待命實例。

待命實例會以指定的間隔嘗試開啟 `SYSTEM.FTE.HA.<agent name>` 佇列。針對此目的，在 `agent.properties` 檔案中提供了其他代理程式內容 `standbyPollInterval`。

使用預設值，待命實例會每五秒嘗試開啟一次 `SYSTEM.FTE.HA.agent name` 佇列。這會一直重複，直到實例成功開啟 `SYSTEM.FTE.HA.agent name` 佇列，或使用 `fteStopAgent` 指令停止。

多個待命實例

所有待命實例都會嘗試以專用 GET 模式取得 `SYSTEM.FTE.HA.agent name` 佇列，且在作用中實例失敗之後成功的實例會變成作用中實例。

作用中實例會維護所有已知待命實例的資訊，並將資訊發佈為代理程式狀態發佈的一部分。從 IBM MQ 9.1.4，`fteShowAgentDetails` 指令、代理程式 GET REST API 回應及 IBM MQ Explorer MFT 外掛程式的輸出會顯示所有待命實例的相關資訊。

如需相關資訊，請參閱 `fteShowAgentDetails` 指令及代理程式 `GET REST API` 回應的輸出範例。

如需 XML 格式的代理程式狀態資訊範例，請參閱 `MFT 代理程式狀態訊息`。

版本需求

作用中及待命代理程式必須是 IBM MQ 9.1.4 或更高版本。



小心:

- 您無法在高可用性模式下配置或啟動 IBM MQ 9.1.4 之前的 IBM MQ 版本。
- 作用中及待命實例都必須執行相同版本的程式碼。

會驗證作用中及待命實例的版本，以確保兩個實例的版本相同。暫時動態佇列用於實例之間的通訊。在 `agent.properties` 檔案中定義的兩個代理程式內容 `dynamicQueuePrefix` 和 `modelQueueName` 會產生暫時動態佇列的名稱。

V 9.1.4 Managed File Transfer 中高可用性代理程式的必要資訊

對於在高可用性配置中執行的標準或橋接器 MFT 代理程式，您需要瞭解各種類型的資訊。此資訊包括代理程式啟動的不同方法、如何在日誌檔中識別代理程式實例，以及代理程式的狀態資訊。

啟動代理程式

在其他位置以非 HA 模式執行代理程式的實例

如果嘗試啟動未配置為 HA 代理程式之代理程式的另一個實例，則會先檢查是否可以獲得 `SYSTEM.FTE.HA.agent name` 佇列上的鎖定。

因為其他實例以非 HA 模式啟動，所以此實例將獲得 `SYSTEM.FTE.HA.agent name` 佇列上的鎖定。代理程式會繼續起始設定，但稍後會失敗，因為指令佇列由另一個實例專用開啟。

在此情況下，下列範例中顯示的訊息會記載至代理程式的 `output0.log` 檔案，且代理程式會每 30 秒繼續嘗試開啟指令佇列一次：

```
BFGMQ1045I: 代理程式的系統佇列 'SYSTEM.FTE.COMMAND.SRC' 配置為 NOSHARE 或 DEFSOPT (SHARED)。
```

```
BFGAG0035W: 代理程式在嘗試開啟佇列時收到 MQI 原因碼 2042 'SYSTEM.FTE.COMMAND.SRC'，連線名稱為 'localhost (1414) ' 及通道 'MFT_HA_CHN'。代理程式將每 30 秒重試一次作業。
```

在其他位置以 HA 模式執行代理程式的實例

如果嘗試啟動未配置為 HA 代理程式之代理程式的另一個實例，則會先檢查是否可以獲得 `SYSTEM.FTE.HA.agent name` 佇列上的鎖定。

因為其他實例已作為作用中實例執行，所以嘗試獲得鎖定失敗。實例無法啟動，且下列錯誤訊息會記載至代理程式的 `output0.log` 檔案：

```
BFGAG0194E: 此代理程式的實例已在其他位置執行。因此，這個實例無法繼續並將停止。
```

Windows 以 Windows 服務方式啟動代理程式

在 Windows 上，您可以將代理程式作為 Windows 服務啟動。

在啟動期間，Windows 會以標準或 HA 模式啟動 MFT 代理程式。如果代理程式配置為以 HA 模式執行，則服務會作為作用中或待命實例執行，視先獲得鎖定的實例而定。

在日誌檔中識別代理程式的實例類型

參考訊息會寫入代理程式的 `output0.log` 檔案，以指出實例類型。當代理程式實例啟動為作用中實例時，會寫入下列訊息：

```
BFGAG0193I: 代理程式已順利起始設定為作用中實例。
```

當代理程式實例啟動為待命實例時，會寫入下列訊息：

```
BFGAG0193I: 代理程式已順利起始設定為待命實例。
```

代理程式狀態更新

因為有兩個相同代理程式實例在執行中，您需要在代理程式狀態發佈中同時具有這兩個實例的相關資訊。請注意，作用中實例是發佈兩個實例的狀態的實例。

待命實例

在發佈代理程式狀態時，作用中實例會檢查待命實例發佈的經歷時間。

基於此目的，`agent.properties` 檔案中還有兩個其他內容：

- **standbyStatusExpiry** 是要放置至代理程式指令佇列的待命狀態訊息到期時間。如果代理程式的作用中實例在該期間內未處理此訊息，則訊息會到期。

依預設，**standbyStatusExpiry** 的值為 30 秒。訊息也是低優先順序 9 訊息，容許傳送要求的優先順序處理高於待命狀態訊息。

- **standbyStatusPublishInterval** 設定待命實例發佈其狀態的頻率。

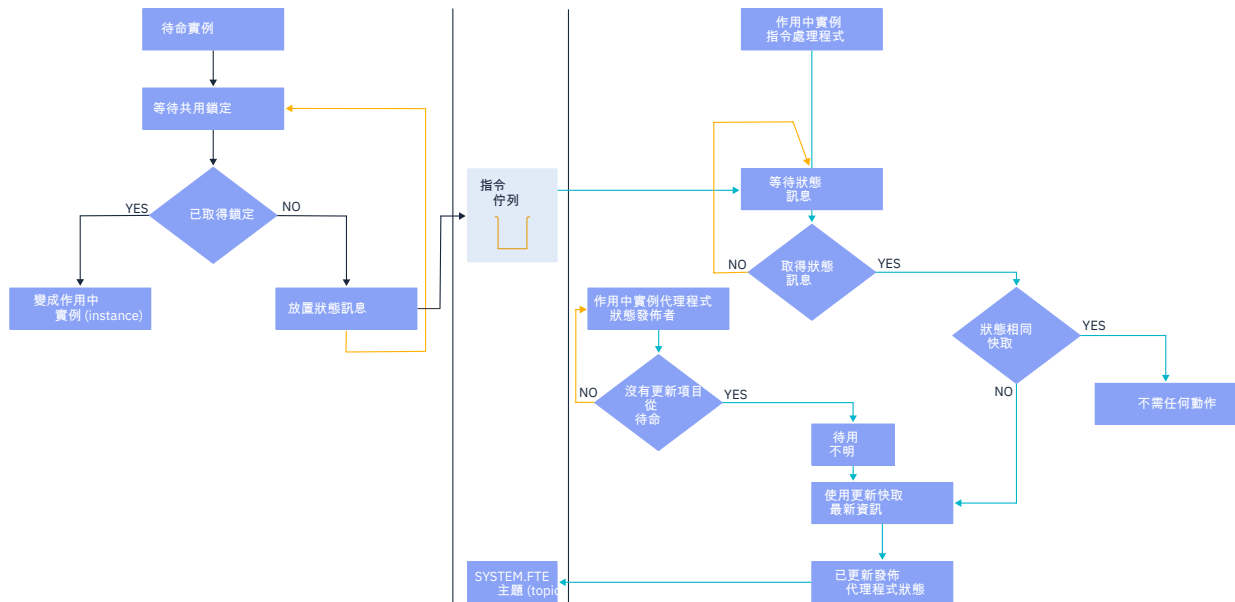
作用中實例

作用中實例會執行下列動作，以處理來自待命實例的狀態更新：

1. 從 `SYSTEM.FTE.COMMAND.<agent name>` 佇列取得訊息，並將訊息處理委派給工作程式執行緒。
2. 工作者執行緒會從訊息內文中擷取內容，以待命實例資訊更新代理程式狀態物件，並通知代理程式狀態發佈者發佈狀態。
3. 代理程式狀態發佈者會發佈狀態。

請注意，這裡已完成最佳化，以快取待命狀態資訊。提出要求時，代理程式狀態發佈者會檢查具有快取狀態的新狀態，並僅在有差異時發佈。

下圖說明作用中或待命實例遵循的流程，以發佈代理程式的狀態：



V 9.1.4 捨棄高可用性代理程式中的實例、失效接手及維護

高可用性 Managed File Transfer 實例可以捨棄、以各種不同的方式失敗，且可能需要維護。

捨棄待命實例狀態

在某些情況下，作用中實例會忙於傳送，且無法處理待命實例狀態訊息，或待命實例失敗，或因任何原因而未發佈狀態訊息。

在這種情況下，知道待命實例存在的作用中代理程式會等待 `agent.properties` 檔案中 `standbyStatusDiscardTime` 內容所指定的值，然後再從其清單中移除待命實例。此內容的預設值為 600 秒，是 `standbyStatusPublishInterval` 內容的兩倍。

正常失效接手實例

您必須搭配使用 `fteStopAgent` 指令與 `-i` 選項，以執行一般失效接手。

這可確保作用中實例立即停止。如果您在沒有 `-i` 選項的情況下停止代理程式，則代理程式會繼續執行，直到作用中實例完成所有進行中的傳送為止，因此失效接手可能需要很長時間。

任何進行中的傳送都會從最後一個已知檢查點回復。

在其他狀況下失效接手實例

如果作用中實例以不正常的方式結束，或整個機器失敗，則代理程式佇列的連線會中斷，且佇列管理程式會關閉所有開啟的佇列 (包括 `SYSTEM.FTE.HA.<agent name>` 佇列及連線)。

因此，待命實例會獲得專用 GET，並完成代理程式起始設定的其餘部分。

同樣地，任何進行中傳送都會從最後一個已知檢查點回復。

如果佇列管理程式的連線中斷

CLIENT 模式

代理程式處理程序由數個執行緒組成。除了預設執行緒之外 (例如，定期發佈代理程式狀態的執行緒)，每個傳送要求都會以一組在傳送完成之後結束的執行緒來處理。

其中許多執行緒會連接至代理程式佇列管理程式，並放置及取得訊息。任何這些連線可能因網路問題或佇列管理程式失敗而中斷。當任何執行緒偵測到連線中斷問題時，執行緒會通知主要執行緒起始回復，並結束。

然後主要執行緒會啟動另一個執行緒，以等待建立與佇列管理程式的連線。重新連接之後，會嘗試獲得代理程式的專用 GET。如果成功，代理程式會繼續完成回復，並變成作用中實例。如果嘗試獲得專用 GET 失敗，實例會變成待命實例。

BINDINGS 模式

以連結模式連接時，如果代理程式中斷連線，則代理程式程序會結束。程序控制器會處理代理程式的重新啟動。當代理程式重新啟動時，它會經歷嘗試取得其本身專用 GET 的處理程序。

如果代理程式成功，則它會變成作用中實例；否則代理程式會變成待命實例。

套用維護層次升級

將維護套用至高可用性代理程式的步驟與針對多重實例佇列管理程式所記載的步驟類似。如需相關資訊，請參閱 [將維護層次更新套用至 Windows 上的多重實例佇列管理程式](#)、[將維護層次更新套用至 AIX 上的多重實例佇列管理程式](#)、[將維護層次更新套用至 Solaris 上的多重實例佇列管理程式](#) 或 [將維護層次更新套用至 Linux 上的多重實例佇列管理程式](#)。

在套用維護層次之前，您必須停止在要套用維護層次的機器上執行的代理程式。如果您要更新作用中實例，為了傳送的連續性，您必須將作用中實例失效接手至待命實例。

升級完成之後，您必須啟動代理程式實例，將現行作用中實例失效接手至已升級的實例，然後升級待命實例。

從舊版產品移轉代理程式

從 IBM MQ 9.1.4 之前的 IBM MQ 版本移轉的代理程式以非高可用性執行。您可以遵循 [從舊版移轉 Managed File Transfer 代理程式](#) 中的程序，讓它們以高可用性模式執行。

管理 IBM MQ Console 和 REST API 的 mqweb 伺服器隨附了預設配置。為了使用其中任一元件，需要完成一些配置作業，例如配置安全以容許使用者登入。本主題說明所有可用的配置選項。

程序

- [第 643 頁的『配置安全』](#)
- [第 643 頁的『配置 HTTP 主機名稱』](#)
- [第 644 頁的『配置 HTTP 及 HTTP 埠』](#)
- [第 645 頁的『配置回應逾時』](#)
- [第 646 頁的『配置自動啟動』](#)
- [第 647 頁的『配置記載』](#)
- [第 649 頁的『配置 LTPA 記號』](#)
- [第 650 頁的『配置 administrative REST API 閘道』](#)
- [第 651 頁的『配置 messaging REST API』](#)
- [第 653 頁的『配置 REST API for MFT』](#)
- [第 655 頁的『調整 mqweb 伺服器 JVM』](#)
- [第 656 頁的『IBM MQ Console 和 REST API 安裝元件的檔案結構』](#)
- [第 658 頁的『在 z/OS 上配置 mqweb 伺服器使用情形記錄』](#)



mqweb 伺服器的基本配置

您必須先安裝正確的元件並配置管理 REST API 或 IBM MQ Console 的 mqweb 伺服器，然後才能開始使用 REST API 或 IBM MQ Console。

關於這項作業





此作業的程序著重於 mqweb 伺服器的基本配置，以便您可以快速開始使用 REST API 和 IBM MQ Console。配置安全大綱的步驟如何設定基本使用者登錄，但配置使用者和角色的其他選項仍存在。如需配置 mqweb 伺服器安全的相關資訊，請參閱 [IBM MQ 主控台](#) 及 [REST API 安全](#)。

註：您必須具有 mqwebuser.xml 檔案的存取權，才能完成此程序：

-  **z/OS** 在 z/OS 上，您必須是具有 mqwebuser.xml 檔案寫入權的使用者。
-  **Multi** 在所有其他作業系統上，您必須是 [特許使用者](#)，才能存取 mqwebuser.xml 檔案。

程序

1. 安裝 IBM MQ Console 和 REST API 元件：

-  **AIX** 在 AIX 上，安裝 mqm.web.rte 檔案集。如需在 AIX 上安裝檔案集的相關資訊，請參閱 [AIX 安裝作業](#)。
-  **IBM i** 在 IBM i 上，安裝 WEB 元件。如果要使用這項特性，您也必須安裝 5724L26 IBM MQ Java 傳訊和 Web 服務，以及 5770JV1 Java SE 8 必備項目。如需在 IBM i 上安裝特性的相關資訊，請參閱 [IBM i 安裝作業](#)。
-  **Linux** 在 Linux 上，安裝 MQSeriesWeb 元件。如需在 Linux 上安裝元件的相關資訊，請參閱 [Linux 安裝作業](#)。
-  **Solaris** 在 Solaris 上，安裝 web 元件。如需在 Solaris 上安裝元件的相關資訊，請參閱 [Solaris 安裝作業](#)。

- **Windows** 在 Windows 上，安裝 Web Administration 特性。如需在 Windows 上安裝特性的相關資訊，請參閱 [Windows 安裝作業](#)。
 - **z/OS** 在 z/OS 上，安裝 IBM MQ for z/OS Unix System Services Web Components 特性。如需在 z/OS 上安裝元件及特性的相關資訊，請參閱 [z/OS 安裝作業](#)。
2. **z/OS**
在 z/OS 上，透過執行 **crtmqweb** Script，建立管理 IBM MQ Console 和 REST API 的 mqweb 伺服器。此 Script 會建立包含 mqweb 伺服器配置及日誌檔的 WebSphere Liberty 使用者目錄。如需執行 **crtmqweb** Script 的相關資訊，請參閱 [第 749 頁的『建立 mqweb 伺服器』](#)。
 3. **z/OS**
在 z/OS 上，建立已編目程序以啟動 mqweb 伺服器。
如需相關資訊，請參閱 [第 750 頁的『建立 mqweb 伺服器的程序』](#)。
 4. 將現有配置檔 `mqwebuser.xml` 取代為配置為提供基本安全的基本登錄範例檔。將 `basic_registry.xml` 檔案從 `MQ_INSTALLATION_PATH/web/mq/samp/configuration` 目錄複製到您系統的適當目錄，並將檔案重新命名為 `mqwebuser.xml`：

- **Linux** **UNIX** 在 UNIX 和 Linux 上：`var/mqm/web/installations/installationName/servers/mqweb`
- **Windows** 在 Windows 上：
`MQ_DATA_PATH\web\installations\installationName\servers\mqweb`
其中 `MQ_DATA_PATH` 是 IBM MQ 資料路徑，此路徑是在安裝 IBM MQ 期間選取的資料路徑。依預設，此路徑為 `C:\ProgramData\IBM\MQ`。
- **z/OS** 在 z/OS 上：`WLP_user_directory/servers/mqweb`
其中 `WLP_user_directory` 是在執行 **crtmqweb** Script 以建立 mqweb 伺服器定義時指定的目錄。

`basic_registry.xml` 範例檔會配置四個使用者：

MQADMIN

屬於 MQWebAdmin 角色的管理使用者。

mqreader

唯讀管理使用者，是 MQWebAdminRO 角色的成員。

mftadmin

屬於 MFTWebAdmin 角色成員的管理使用者。

mftreader

唯讀管理使用者，是 MFTWebAdminRO 角色的成員。

所有使用者也都是 MQWebUser 角色的成員。

如需可用角色的相關資訊，請參閱 [IBM MQ Console 和 REST API 上的角色](#)

5. 選擇性的：編輯 `mqwebuser.xml` 檔案，以新增更多使用者和群組。將適當的角色指派給那些使用者和群組，以獲授權使用 REST API 或 IBM MQ Console。您也可以變更依預設定義之使用者的密碼，並將新密碼編碼。如需相關資訊，請參閱 [配置使用者和角色](#)。

註：

- **z/OS** 在 z/OS 上，如果您將使用者新增至 MQWebUser 角色，則也必須授與 mqweb 已啟動作業使用者 ID 替代使用者對具有 MQWebUser 角色之使用者 ID 的存取權。例如：

```
RDEFINE MQADMIN hlq.ALTERNATE.USER.userId UACC(NONE)
PERMIT hlq.ALTERNATE.USER.userId CLASS(MQADMIN) ACCESS(UPDATE) ID(mqwebUserId)
```

- **z/OS** **Multi** 若要完成開始使用 messaging REST API 的步驟，您必須將使用者新增至 `mqwebuser.xml` 檔案。此使用者必須與系統上現有的 IBM MQ 使用者具有相同的名稱。遵循與 `xml`

檔案中其他使用者相同的格式，在 xml 檔案中的下列行之後新增使用者 ID 和密碼: <user name="mftreader" password="mftreader"/>。

6. z/OS

在 z/OS 上，輸入下列指令，以設定 WLP_USER_DIR 環境變數，讓變數指向 mqweb 伺服器配置：

```
export WLP_USER_DIR=WLP_user_directory
```

其中，*WLP_user_directory* 是傳遞至 *crtmqweb* 的目錄名稱。例如：

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

如需相關資訊，請參閱第 749 頁的『建立 mqweb 伺服器』。

7. 依預設，REST API 和 IBM MQ Console 只能從與 mqweb 伺服器相同的主機中使用。透過輸入下列指令，啟用與 mqweb 伺服器的遠端連線：

```
setmqweb properties -k httpHost -v hostname
```

其中 *hostname* 指定 IP 位址、具有網域名稱字尾的網域名稱伺服器 (DNS) 主機名稱，或 IBM MQ 安裝所在伺服器的 DNS 主機名稱。使用雙引號中的星號 * 來指定所有可用的網路介面，如下列範例所示：

```
setmqweb properties -k httpHost -v "*"
```

8. 選擇性的：依預設，不會啟用 MFT 的 administrative REST API。如果您想要使用此特性，則必須啟用它並配置協調佇列管理程式：

- a) 輸入下列指令，以啟用 administrative REST API for MFT：

```
setmqweb properties -k mqRestMftEnabled -v true
```

- b) 輸入下列指令，以配置哪個佇列管理程式是協調佇列管理程式：

```
setmqweb properties -k mqRestMftCoordinationQmgr -v qmgrName
```

其中 *qmgrName* 是協調佇列管理程式的名稱。

c) CD

若要啟用 POST 呼叫，請輸入下列指令來配置哪個佇列管理程式是指令佇列管理程式：

```
setmqweb properties -k mqRestMftCommandQmgr -v qmgrName
```

其中 *qmgrName* 是指令佇列管理程式的名稱。

註：此步驟適用於 IBM MQ 9.1.2。

9. 啟動支援 REST API 和 IBM MQ Console 的 mqweb 伺服器：

- **ULW** 在 UNIX, Linux, and Windows 上，以特許使用者身分輸入下列指令：

```
stirmqweb
```

- **IBM i** 在 IBM i 上，以特許使用者身分，在 Qshell 中輸入下列指令：

```
/QIBM/ProdData/mqm/bin/stirmqweb
```

- **z/OS** 在 z/OS 上，啟動您在第 750 頁的『建立 mqweb 伺服器的程序』中建立的程序。

下列訊息會發出至 STDOUT DD，以指出 mqweb 伺服器已順利啟動。

```
[AUDIT ] MQWB2019I: MQ Console level: 9.2.4 - V924-CD924-L211028
[AUDIT ] MQWB0023I: MQ REST API level: 9.2.4 - V924-CD924-L211028
[AUDIT ] CWWKZ0001I: Application com.ibm.mq.rest started in 1.763 seconds.
[AUDIT ] CWWKZ0001I: Application com.ibm.mq.console started in 2.615 seconds.
[AUDIT ] CWWKF0011I: The mqweb server is ready to run a smarter planet. The mqweb
server started in 10.016 seconds.
```

您隨時可以停止 z/OS 上的 mqweb 伺服器啟動作業，或使用 **endmqweb** 指令來停止 mqweb 伺服器。不過，如果 mqweb 伺服器不在執行中，則無法使用 REST API 或 IBM MQ Console。

10. **z/OS**

選擇性的：在 z/OS 上，如果您想要容許系統自動化產品設陷 MQ Console 及 REST API 啟動時發出的 MQWB2019I 及 MQWB0023I 訊息，請配置 mqweb 伺服器以將這些訊息寫入 MVS 主控台。若要將 mqweb 伺服器配置為將 MQWB2019I 及 MQWB0023I 訊息寫入 MVS 主控台，請編輯您在步驟 [第 641 頁的『4』](#) 中建立的 mqwebuser.xml 檔案，並將下列行新增至該檔案：

```
<zosLogging enableLogToMVS="true" wtoMessage="MQWB2019I,MQWB0023I"/>
```

如需在 mqweb 伺服器中配置 z/OS Logging 的相關資訊，請參閱 [z/OS Logging \(zosLogging\)](#)。

下一步

1. 配置 mqweb 伺服器設定，包括啟用 HTTP 連線及變更埠號。如需相關資訊，請參閱 [第 640 頁的『配置 IBM MQ Console 和 REST API』](#)。
2. 選擇性地配置 REST API：
 - a. 配置 REST API 的「跨原點資源共用」。依預設，您無法從與 REST API 不在相同網域上管理的 Web 資源存取 REST API。也就是說，未啟用跨原點要求。您可以配置「跨原點資源共用 (CORS)」，以容許來自指定 URL 的跨原點要求。如需相關資訊，請參閱 [為 REST API 配置 CORS](#)。
 - b. 針對 MFT 配置 REST API。如需相關資訊，請參閱 [第 653 頁的『配置 REST API for MFT』](#)。
3. 使用 REST API 或 IBM MQ Console：
 - [開始使用 administrative REST API](#)
 - [開始使用 messaging REST API](#)
 - [開始使用 IBM MQ Console](#)

V 9.1.0 配置安全

您可以透過編輯 mqwebuser.xml 檔案，來配置 IBM MQ Console 和 REST API 的安全。您可以透過配置基本使用者登錄或 LDAP 登錄，或 WebSphere Liberty 隨附的任何其他登錄類型，來配置及鑑別使用者。然後，您可以透過向使用者和群組指派角色來授權這些使用者。

關於這項作業

若要配置 IBM MQ Console、及 REST API 的安全，您必須配置使用者和群組。然後，這些使用者和群組可以獲授權使用 IBM MQ Console 及/或 REST API。如需配置使用者和群組以及鑑別和授權使用者的相關資訊，請參閱 [IBM MQ 主控台 和 REST API 安全](#)。

當使用者向 IBM MQ Console 進行鑑別時，會產生 LTPA 記號。此記號可讓使用者使用 IBM MQ Console，而無需重新鑑別，直到記號到期為止。

如果您對 REST API 使用記號型鑑別，當使用者使用 HTTP POST 方法來登入 /login REST API 資源時，會產生不同的 LTPA 記號。您可以配置這個記號的到期時間，以及這個記號是否可以同時用於 HTTP 和 HTTP 連線。如需相關資訊，請參閱 [第 649 頁的『配置 LTPA 記號』](#)。

程序

- [IBM MQ 主控台 和 REST API 安全](#)
- [第 649 頁的『配置 LTPA 記號』](#)

V 9.1.0 配置 HTTP 主機名稱

依預設，管理 IBM MQ Console 和 REST API 的 mqweb 伺服器配置為僅容許本端連線。也就是說，只能在安裝 IBM MQ Console 和 REST API 的系統上存取 IBM MQ Console 和 REST API。您可以使用 **setmqweb** 指令來配置主機名稱，以容許遠端連線。

開始之前

若要完成此作業，必須是具有特定專用權的使用者，讓您可以使用 **dspmqweb** 及 **setmqweb** 指令：

- ▶ **z/OS** 在 z/OS 上，您必須具有執行 **dspmqweb** 和 **setmqweb** 指令的權限，以及對 `mqwebuser.xml` 檔案的寫入權。
- ▶ **Multi** 在所有其他作業系統上，必須是 特許使用者。



小心: **V 9.1.0** ▶ **z/OS**

在 z/OS 上發出 **setmqweb** 或 **dspmqweb** 指令之前，您必須先設定 `WLP_USER_DIR` 環境變數，讓變數指向您的 mqweb 伺服器配置。

若要這樣做，請發出下列指令：

```
export WLP_USER_DIR=WLP_user_directory
```

其中，`WLP_user_directory` 是傳遞至 `crtmqweb` 的目錄名稱。例如：

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

如需相關資訊，請參閱 [建立 mqweb 伺服器](#)。

程序

- 使用下列指令來檢視 HTTP 主機名稱的現行配置：

```
dspmqweb properties -a
```

`httpHost` 欄位會顯示 HTTP 主機名稱。如需相關資訊，請參閱 [dspmqweb](#)。

- 使用下列指令來設定 HTTP 主機名稱：

```
setmqweb properties -k httpHost -v hostName
```

其中 `hostName` 指定 IP 位址、具有網域名稱字尾的網域名稱伺服器 (DNS) 主機名稱，或 IBM MQ 安裝所在伺服器的 DNS 主機名稱。請使用雙引號中的星號來指定所有可用的網路介面。使用值 `localhost`，以僅容許本端連線。

- 使用下列指令取消設定 HTTP 主機名稱：

```
setmqweb properties -k httpHost -d
```

V 9.1.0 配置 HTTP 及 HTTP 埠

依預設，管理 IBM MQ Console 和 REST API 的 mqweb 伺服器會使用 HTTPS 埠 9443。已停用與 HTTP 連線相關聯的埠。您可以啟用 HTTP 埠、配置不同的 HTTP 埠，或停用 HTTP 或 HTTPS 埠。您可以使用 **setmqweb** 指令來配置埠。

開始之前

如果您啟用 HTTP 埠，且使用記號型鑑別，則必須啟用相同的 LTPA 記號，以同時用於 HTTP 和 HTTPS 連線。如需相關資訊，請參閱 第 649 頁的『[配置 LTPA 記號](#)』。

若要完成此作業，必須是具有特定專用權的使用者，讓您可以使用 **dspmqweb** 及 **setmqweb** 指令：

- ▶ **z/OS** 在 z/OS 上，您必須具有執行 **dspmqweb** 和 **setmqweb** 指令的權限，以及對 `mqwebuser.xml` 檔案的寫入權。
- ▶ **Multi** 在所有其他作業系統上，必須是 特許使用者。



小心: **V 9.1.0** ▶ **z/OS**

在 z/OS 上發出 **setmqweb** 或 **dspmqweb** 指令之前，您必須先設定 WLP_USER_DIR 環境變數，讓變數指向您的 mqweb 伺服器配置。

若要這樣做，請發出下列指令：

```
export WLP_USER_DIR=WLP_user_directory
```

其中，*WLP_user_directory* 是傳遞至 **crtmqweb** 的目錄名稱。例如：

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

如需相關資訊，請參閱[建立 mqweb 伺服器](#)。

程序

- 使用下列指令來檢視 HTTP 及 HTTPS 埠的現行配置：

```
dspmqweb properties -a
```

`httpPort` 欄位會顯示 HTTP 埠，而 `httpsPort` 欄位會顯示 HTTPS 埠。如需相關資訊，請參閱 **dspmqweb**。

- 使用下列指令來啟用或配置 HTTP 埠：

- 使用下列指令啟用或設定 HTTP 埠：

```
setmqweb properties -k httpPort -v portNumber
```

其中 *portNumber* 指定您要用於 HTTP 連線的埠。您可以使用 `-1` 值來停用埠。

- 使用下列指令，將 HTTP 埠值重設為預設值 `-1`：

```
setmqweb properties -k httpPort -d
```

- 配置 HTTPS 埠：

- 使用下列指令來設定 HTTPS 埠號：

```
setmqweb properties -k httpsPort -v portNumber
```

其中 *portNumber* 指定您要用於 HTTPS 連線的埠。您可以使用 `-1` 值來停用埠。

- 使用下列指令，將 HTTPS 埠號重設為預設值 `9443`：

```
setmqweb properties -k httpsPort -d
```

V 9.1.0 配置回應逾時

依預設，如果將回應傳回給用戶端所花費的時間超過 30 秒，則 IBM MQ Console 和 REST API 會逾時。您可以使用 **setmqweb** 指令，將 IBM MQ Console 和 REST API 配置成使用不同的逾時值。

開始之前

若要完成此作業，必須是具有特定專用權的使用者，讓您可以使用 **dspmqweb** 及 **setmqweb** 指令：

- **z/OS** 在 z/OS 上，您必須具有執行 **dspmqweb** 和 **setmqweb** 指令的權限，以及對 `mqwebuser.xml` 檔案的寫入權。
- **Multi** 在所有其他作業系統上，必須是 [特許使用者](#)。



小心： **V 9.1.0** **z/OS**

在 z/OS 上發出 **setmqweb** 或 **dspmqweb** 指令之前，您必須先設定 WLP_USER_DIR 環境變數，讓變數指向您的 mqweb 伺服器配置。

若要這樣做，請發出下列指令：

```
export WLP_USER_DIR=WLP_user_directory
```

其中，*WLP_user_directory* 是傳遞至 `crtmqweb` 的目錄名稱。例如：

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

如需相關資訊，請參閱[建立 mqweb 伺服器](#)。

程序

- 使用下列指令來檢視要求逾時的現行配置：

```
dspmqweb properties -a
```

`mqRestRequestTimeout` 欄位顯示回應逾時的現行值。如需相關資訊，請參閱 [dspmqweb](#)。

- 使用下列指令來設定要求逾時：

```
setmqweb properties -k mqRestRequestTimeout -v timeout
```

其中 *timeout* 指定逾時之前的時間 (以秒為單位)。

- 使用下列指令，將要求逾時值重設為預設值 30 秒：



```
setmqweb properties -k mqRestRequestTimeout -d
```

V 9.1.0 配置自動啟動

依預設，當 `mqweb` 伺服器啟動時，會自動啟動 IBM MQ Console。您可以使用 `setmqweb` 指令來配置 IBM MQ Console 和 REST API 是否自動啟動。

開始之前

若要完成此作業，必須是具有特定專用權的使用者，讓您可以使用 `dspmqweb` 及 `setmqweb` 指令：

-  在 z/OS 上，您必須具有執行 `dspmqweb` 和 `setmqweb` 指令的權限，以及對 `mqwebuser.xml` 檔案的寫入權。
-  在所有其他作業系統上，必須是 [特許使用者](#)。



小心:  

在 z/OS 上發出 `setmqweb` 或 `dspmqweb` 指令之前，您必須先設定 `WLP_USER_DIR` 環境變數，讓變數指向您的 `mqweb` 伺服器配置。

若要這樣做，請發出下列指令：

```
export WLP_USER_DIR=WLP_user_directory
```

其中，*WLP_user_directory* 是傳遞至 `crtmqweb` 的目錄名稱。例如：

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

如需相關資訊，請參閱[建立 mqweb 伺服器](#)。

程序

- 使用下列指令來檢視自動啟動的現行配置：

```
dspmqweb properties -a
```

`mqRestAutostart` 欄位會顯示 REST API 是否自動啟動，而 `mqConsoleAutostart` 欄位會顯示 IBM MQ Console 是否自動啟動。如需相關資訊，請參閱 [dspmqweb](#)。

- 使用下列指令來配置 IBM MQ Console 是否自動啟動：

```
setmqweb properties -k mqConsoleAutostart -v start
```

其中 *start* 是您要 IBM MQ Console 自動啟動的值 *true*，否則為 *false*。

- 使用下列指令來配置 REST API 是否自動啟動:

```
setmqweb properties -k mqRestAutostart -v start
```

其中 *start* 是您要 REST API 自動啟動的值 *true*，否則為 *false*。

V 9.1.0 配置記載

您可以配置記載層次、日誌檔大小上限，以及管理 IBM MQ Console 和 REST API 的 mqweb 伺服器所使用的日誌檔數目上限。您可以使用 **setmqweb** 指令來配置記載。

開始之前

若要完成此作業，必須是具有特定專用權的使用者，讓您可以使用 **dspmqweb** 及 **setmqweb** 指令：

- **z/OS** 在 z/OS 上，您必須具有執行 **dspmqweb** 和 **setmqweb** 指令的權限，以及對 `mqwebuser.xml` 檔案的寫入權。
- **Multi** 在所有其他作業系統上，必須是 特許使用者。

關於這項作業

mqweb 伺服器的日誌檔可以在下列其中一個目錄中找到：

- **ULW** 在 UNIX, Linux, and Windows 上: `MQ_DATA_PATH/web/installations/installationName/servers/mqweb/logs`
- **z/OS** 在 z/OS 上: `WLP_user_directory/servers/mqweb/logs`

其中 `WLP_user_directory` 是在執行 **V 9.1.0 crtmqweb** Script 以建立 mqweb 伺服器定義時指定的目錄。

mqweb 伺服器的傳訊追蹤檔案可以在下列其中一個目錄中找到：

- **ULW** 在 UNIX, Linux, and Windows 上: `MQ_DATA_PATH/web/installations/installationName/servers/mqweb`
- **z/OS** 在 z/OS 上: `WLP_user_directory/servers/mqweb`

其中 `WLP_user_directory` 是在執行 **V 9.1.0 crtmqweb** Script 以建立 mqweb 伺服器定義時指定的目錄。

如需啟用 IBM MQ Console 和 REST API 追蹤的相關資訊，請參閱 [追蹤 IBM MQ Console 和 REST API](#)。

程序

- 使用下列指令來檢視 REST API 記載的現行配置：

```
dspmqweb properties -a
```

`maxTraceFileSize` 欄位顯示追蹤檔大小上限，`maxTraceFiles` 欄位顯示追蹤檔數目上限，`traceSpec` 欄位顯示使用的追蹤層次。此外，`maxMsgTraceFileSize` 欄位會顯示傳訊追蹤檔大小上限，而 `maxMsgTraceFiles` 欄位會顯示傳訊追蹤檔數目上限。如需相關資訊，請參閱 [dspmqweb](#)。

- 配置日誌檔大小上限：

- 使用下列指令來設定日誌檔大小上限：

```
setmqweb properties -k maxTraceFileSize -v size
```

其中 *size* 指定 IBM MQ Console 的每一個日誌檔可以達到的大小 (以 MB 為單位)。

- 使用下列指令，將日誌檔大小上限重設為預設值 20 MB：
`setmqweb properties -k maxTraceFileSize -d`
- 配置用於記載的檔案數目上限：
 - 使用下列指令來設定要用於記載的檔案數目上限：
`setmqweb properties -k maxTraceFiles -v max`
 其中 *max* 指定 IBM MQ Console 的檔案數目上限。
 - 使用下列指令，將用於記載的檔案數目上限重設為預設值 2：
`setmqweb properties -k maxTraceFiles -d`
- 配置傳訊追蹤大小上限：
 - 使用下列指令來設定訊息追蹤大小上限：
`setmqweb properties -k maxMsgTraceFileSize -v size`
 其中 *size* 指定每一個日誌檔可以達到的大小 (MB)。
 - 使用下列指令，將日誌檔大小上限重設為預設值 200 MB：
`setmqweb properties -k maxMsgTraceFileSize -d`
- 配置要使用的傳訊追蹤檔數目上限：
 - 使用下列指令，設定用於傳訊追蹤的檔案數目上限：
`setmqweb properties -k maxMsgTraceFiles -v max`
 其中 *max* 指定檔案數目上限。
 - 使用下列指令，將用於傳訊追蹤的檔案數目上限重設為預設值 5：
`setmqweb properties -k maxMsgTraceFiles -d`
- 配置使用的記載層次：
 - 使用下列指令來設定所使用的記載層次：
`setmqweb properties -k traceSpec -v level`
 其中 *level* 是第 648 頁的表 52 中列出的其中一個值。下表概述詳細程度增加的記載層次。當您啟用記載層次時，也會在它之前啟用每一個層次。例如，如果您啟用 ***=warning** 記載層次，則也會啟用 ***=severe** 及 ***=fatal** 記載層次。
 當 IBM 服務要求時，請變更此值。
 - 使用下列指令，將使用的記載層次重設為預設值 ***=info**：
`setmqweb properties -k traceSpec -d`

值	已套用記載層次
*=off	已關閉記載。
* = 嚴重	作業無法繼續，且元件、應用程式及伺服器無法運作。
* = 嚴重	作業無法繼續，但元件、應用程式及伺服器仍可運作。此層次也會指出即將發生之無法復原的錯誤。
* = 警告	潛在錯誤或即將發生錯誤。此層次也可以指出漸進失敗 (例如，潛在的資源洩漏)。
*=audit	影響伺服器狀態或資源的重大事件
*=info	概述整體作業進度的一般資訊

值	已套用記載層次
* =config	配置變更或狀態
* = detail	詳述子作業進度的一般資訊
* = fine	追蹤資訊-一般追蹤 + 方法進入、結束及回覆值
* = finer	追蹤資訊-詳細追蹤
* = 最細微	追蹤資訊-更詳細的追蹤，包括除錯問題所需的所有詳細資料
* = 全部	記載所有事件

配置 LTPA 記號

LTPA 記號可用來避免需要使用者對 mqweb 伺服器的每一項要求提供使用者名稱及密碼認證。您可以使用 **setmqweb** 指令來配置 LTPA 記號 Cookie 的名稱、LTPA 鑑別記號的期限間隔，以及配置 HTTP 連線是否可以使用 LTPA 記號。

開始之前

若要完成此作業，必須是具有特定專用權的使用者，讓您可以使用 **dspmqweb** 及 **setmqweb** 指令：

- ▶ **z/OS** 在 z/OS 上，您必須具有執行 **dspmqweb** 和 **setmqweb** 指令的權限，以及對 `mqwebuser.xml` 檔案的寫入權。
- ▶ **Multi** 在所有其他作業系統上，必須是 [特許使用者](#)。

註：如果您同時使用 IBM MQ Console 及記號鑑別與 REST API，則會共用到期間隔。



小心： **V 9.1.0** **z/OS**

在 z/OS 上發出 **setmqweb** 或 **dspmqweb** 指令之前，您必須先設定 `WLP_USER_DIR` 環境變數，讓變數指向您的 mqweb 伺服器配置。

若要這樣做，請發出下列指令：

```
export WLP_USER_DIR=WLP_user_directory
```

其中，`WLP_user_directory` 是傳遞至 `crtmqweb` 的目錄名稱。例如：

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

如需相關資訊，請參閱 [建立 mqweb 伺服器](#)。

關於這項作業

當使用者登入 IBM MQ Console 時，會產生 LTPA 記號。如果您對 REST API 使用記號型鑑別，當使用者使用 HTTP POST 方法來登入 `/login` REST API 資源時，會產生 LTPA 記號。此記號會在 Cookie 中傳回。在記號到期之前，會使用記號來鑑別使用者，而不需要使用者使用其使用者 ID 及密碼重新登入。預設期限間隔為 120 分鐘。

包含 LTPA 記號的 Cookie 名稱會因平台而異：

- ▶ **MQ Appliance** 在 IBM MQ Appliance 上，LTPA 記號是 `LtpaToken2`。這個值無法變更。
- ▶ **ULW** **z/OS** 依預設，在所有其他平台上，包含 LTPA 記號的 Cookie 名稱會以 `LtpaToken2` 開頭，並包含可在 mqweb 伺服器重新啟動時變更的字尾。此隨機化 Cookie 名稱容許多個 mqweb 伺服器在相同系統上執行。不過，如果您想要 Cookie 名稱保持一致值，您可以使用 **setmqweb** 指令來指定 Cookie 所擁有的名稱。

ULW **z/OS** **IBM i** 如果您同時啟用 HTTP 及 HTTP 埠，則可以針對 HTTP 要求重複使用針對 HTTP 要求發出的 LTPA 記號。依預設會停用此行為，但您可以使用 **setmqweb** 指令來啟用此行為。

程序

- 使用下列指令，檢視 LTPA 記號的現行期限、LTPA 記號 Cookie 的名稱，以及 LTPA 記號是否可用於 HTTP 要求：

```
dspmqweb properties -a
```

- `ltpaCookieName` 欄位顯示 LTPA 記號 Cookie 的名稱。如果您沒有設定 Cookie 名稱，這個內容的值是 `LtpaToken2_${env.MQWEB_LTPA_SUFFIX}` (在 UNIX, Linux, and Windows 上) 或 `LtpaToken2_${httpsPort}` (在 z/OS 上)。mqweb 伺服器會使用 `LtpaToken2_` 字首後面的變數來產生 Cookie 的唯一名稱。您無法設定此變數，但可以將 `ltpaCookieName` 變更為您選擇的值。
- `ltpaExpiration` 欄位顯示 LTPA 記號到期時間。
- 如果 HTTP 要求可以使用 LTPA 記號，`secureLtpa` 欄位會設為 `false`。

如需相關資訊，請參閱 [dspmqweb](#)。

- 配置 LTPA 記號期限：

- 輸入下列指令來設定 LTPA 記號期限：

```
setmqweb properties -k ltpaExpiration -v time
```

其中 `time` 指定 LTPA 記號到期及使用者登出之前的時間 (分鐘)。

- 輸入下列指令，將 LTPA 記號期限重設為預設值 120 分鐘：

```
setmqweb properties -k ltpaExpiration -d
```

- **ULW** **z/OS**
配置 LTPA 記號 Cookie 名稱：

- 輸入下列指令來設定 LTPA 記號 Cookie 名稱：

```
setmqweb properties -k ltpaCookieName -v name
```

其中 `name` 指定 LTPA 記號 Cookie 的唯一名稱。

- 輸入下列指令，將 LTPA 記號 Cookie 名稱重設為預設值，其中 `LtpaToken2_` 字首後面接著隨機字元：

```
setmqweb properties -k ltpaCookieName -d
```

- **ULW** **z/OS**
輸入下列指令，以配置 HTTP 連線是否可以使用 LTPA 記號：

```
setmqweb properties -k secureLtpa -v secure
```

其中 `secure` 指定 LTPA 記號是否可同時供不安全 HTTP 連線及安全 HTTP 連線使用。`false` 值可讓 HTTP 和 HTTP 連線使用相同的 LTPA 記號。

V 9.1.0 配置 administrative REST API 閘道

依預設，會啟用 administrative REST API 閘道。當啟用 administrative REST API 閘道時，您可以使用閘道佇列管理程式來執行 REST API 的遠端管理。您可以配置用作預設閘道佇列管理程式的佇列管理程式，也可以透過使用 **setmqweb** 指令停用 administrative REST API 閘道來防止遠端管理。

關於這項作業

若要完成此作業，必須是具有特定專用權的使用者，讓您可以使用 **dspmqweb** 及 **setmqweb** 指令：

- **z/OS** 在 z/OS 上，您必須具有執行 **dspmqweb** 和 **setmqweb** 指令的權限，以及對 `mqwebuser.xml` 檔案的寫入權。

- ▶ **Multi** 在所有其他作業系統上，必須是特許使用者。

當下列兩個陳述式都成立時，會使用預設閘道佇列管理程式：

- REST 要求的 `ibm-mq-rest-gateway-qmgr` 標頭中未指定佇列管理程式。
- 在 REST API 資源 URL 中指定的佇列管理程式不是本端佇列管理程式。

如需使用 REST API 進行遠端管理的相關資訊，請參閱 [使用 REST API 進行遠端管理](#)。



小心： **V 9.1.0** **z/OS**

在 z/OS 上發出 `setmqweb` 或 `dspmqweb` 指令之前，您必須先設定 `WLP_USER_DIR` 環境變數，讓變數指向您的 mqweb 伺服器配置。

若要這樣做，請發出下列指令：

```
export WLP_USER_DIR=WLP_user_directory
```

其中，`WLP_user_directory` 是傳遞至 `crtmqweb` 的目錄名稱。例如：

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

如需相關資訊，請參閱 [建立 mqweb 伺服器](#)。

程序

- 使用下列指令來檢視 administrative REST API 閘道的現行配置：

```
dspmqweb properties -a
```

`mqRestGatewayEnabled` 欄位會顯示是否已啟用閘道，而 `mqRestGatewayQmgr` 欄位會顯示預設閘道佇列管理程式的名稱。如需相關資訊，請參閱 [dspmqweb](#)。

- 使用下列指令來配置是否啟用 administrative REST API 閘道：

```
setmqweb properties -k mqRestGatewayEnabled -v enabled
```

其中 `enabled` 是用來啟用 administrative REST API 閘道的值 **true**，否則為 **false**。

- 配置哪個佇列管理程式用來作為預設閘道佇列管理程式：

– 使用下列指令來設定預設閘道佇列管理程式：

```
setmqweb properties -k mqRestGatewayQmgr -v qmgrName
```

其中 `qmgrName` 是與 mqweb 伺服器安裝相同之佇列管理程式的名稱。

– 使用下列指令取消設定預設閘道佇列管理程式：

```
setmqweb properties -k mqRestGatewayQmgr -d
```

V 9.1.0 配置 messaging REST API

依預設，管理 IBM MQ Console 及 REST API 的 mqweb 伺服器已啟用 messaging REST API。

V 9.1.2 從 messaging REST API 到 IBM MQ 的連線會儲存在儲存區中，每一個佇列管理程式都有 20 個連線可用。當所有連線都在使用中時，messaging REST API 會建立新的非儲存區連線來用於要求。您可以使用 `setmqweb properties` 指令來變更儲存連線數目上限，以及 messaging REST API 在所有連線都在使用中時的行為。您也可以使用 `setmqweb properties` 指令來配置是否啟用傳訊。

開始之前

若要完成此作業，必須是具有特定專用權的使用者，讓您可以使用 `dspmqweb` 及 `setmqweb` 指令：

- ▶ **z/OS** 在 z/OS 上，您必須具有執行 `dspmqweb` 和 `setmqweb` 指令的權限，以及對 `mqwebuser.xml` 檔案的寫入權。

- **Multi** 在所有其他作業系統上，必須是特許使用者。

若要使用 messaging REST API，必須向 mqweb 伺服器鑑別呼叫者，且必須是 MQWebUser 角色的成員。MQWebAdmin 和 MQWebAdminRO 角色不適用於 messaging REST API。呼叫者也必須獲得授權來存取用於透過 OAM 或 RACF 進行傳訊的佇列。如需 REST API 安全的相關資訊，請參閱 [IBM MQ Console 和 REST API 安全](#)。



小心: **V 9.1.0** **z/OS**

在 z/OS 上發出 **setmqweb** 或 **dspmqweb** 指令之前，您必須先設定 WLP_USER_DIR 環境變數，讓變數指向您的 mqweb 伺服器配置。

若要這樣做，請發出下列指令：

```
export WLP_USER_DIR=WLP_user_directory
```

其中，*WLP_user_directory* 是傳遞至 crtmqweb 的目錄名稱。例如：

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

如需相關資訊，請參閱 [建立 mqweb 伺服器](#)。

關於這項作業

您可以使用 **setmqweb properties** 指令來配置是否啟用傳訊。

V 9.1.2 為了最佳化 messaging REST API 的效能，會儲存 IBM MQ 佇列管理程式的連線。也就是說，每一個 REST 要求會使用來自連線儲存區的連線，而不是每一個 REST 要求建立、使用及毀損其自己的連線。依預設，每一個佇列管理程式儲存區都有 20 個連線，當所有連線都在使用中時，有三個選項可用來處理要求：

- messaging REST API 可以建立新的非儲存區連線，以用於要求。此行為是預設行為。
- messaging REST API 可能會傳回錯誤。
- messaging REST API 可以等待儲存連線變成可用。此等待是無限期等待。

當所有連線都在使用中，您可以使用 **setmqweb properties** 指令來變更儲存連線數目上限，以及 messaging REST API 的預設行為。

程序

- 使用下列指令來檢視 HTTP messaging REST API 的現行配置：

```
dspmqweb properties -a
```

mqRestMessagingEnabled 欄位顯示是否啟用 messaging REST API。值為 True 表示已啟用 messaging REST API，否則為 False。如需相關資訊，請參閱 [dspmqweb](#)。

- 使用下列指令來配置 messaging REST API：

```
setmqweb properties -k mqRestMessagingEnabled -v enabled
```

其中 *enabled* 是 true 值 (如果您想要啟用 messaging REST API)，否則為 false。

- **V 9.1.2**

配置 messaging REST API 的連線儲存區：

- 使用下列指令來配置每一個佇列管理程式儲存區的連線儲存區大小上限：

```
setmqweb properties -k mqRestMessagingMaxPoolSize -v size
```

其中 *size* 指定儲存區的大小。

註：如果已設定 *mqRestMessagingMaxPoolSize* 的值較大，且正在連接大量佇列管理程式，您應該考慮增加 mqweb 伺服器資料堆的大小上限。

如需相關資訊，請參閱 [調整 mqweb 伺服器 JVM](#)。

- 使用下列指令來配置當儲存區內所有連線都在使用中時，messaging REST API 的行為：
`setmqweb properties -k mqRestMessagingFullPoolBehavior -v action`
其中 *action* 指定要採取的動作。*action* 可以是下列其中一個值：

區塊

當儲存區中的所有連線都在使用中時，請等待連線變成可用。

error

當儲存區中的所有連線都在使用中時，傳回錯誤。

溢位

當儲存區中的所有連線都在使用中時，請建立要使用的非儲存區連線，並在使用之後刪除連線。

V 9.1.0 配置 REST API for MFT

LTS

依預設，管理 IBM MQ Console 和 REST 服務的 mqweb 伺服器未啟用 MFT。您可以使用 `setmqweb properties` 指令來配置是否啟用 REST API for MFT、設定協調佇列管理程式，以及指定 MFT 重新連接逾時。

V 9.1.2

對於 **Create transfer** 之類的建立 REST API 指令，您必須新增

`mqRestMftCommandQmgr`。

V 9.1.4

CD

依預設，管理 IBM MQ Console 和 REST 服務的

mqweb 伺服器未啟用 MFT。

開始之前

V 9.1.4

您需要：

- 若要練習 MFT REST 服務，您需要將 `mqRestMftEnabled` 設為 `true`，並將指令佇列管理程式名稱設為 `mqRestMftCoordinationQmgr`，以啟用 MFT REST 服務：

```
setmqweb properties -k mqRestMftEnabled -v true
setmqweb properties -k mqRestMftCoordinationQmgr -v <coordinationQmgrName>
```

- 若要提交任何、建立要求 (例如，建立傳送或資源監視器)，您需要設定 `mqRestMftCommandQmgr`

```
setmqweb properties -k mqRestMftCommandQmgr -v <commandQmgrName>
```

- 透過發出 `endmqweb` 指令後接 `strmqweb` 指令，重新啟動 mqweb 伺服器以強制執行您剛剛設定的值。
- 發出 `dspmqweb` 指令來檢查 Web 伺服器的狀態。

若要完成此作業，必須是具有特定專用權的使用者，讓您可以使用 `dspmqweb` 及 `setmqweb` 指令：

- **z/OS** 在 z/OS 上，您必須具有執行 `dspmqweb` 和 `setmqweb` 指令的權限，以及對 `mqwebuser.xml` 檔案的寫入權。
- **Multi** 在所有其他作業系統上，必須是 特許使用者。

若要使用 REST API for MFT，呼叫者必須向 mqweb 伺服器進行鑑別，且必須是一個以上 MFTWebAdmin 或 MFTWebAdminRO 角色的成員。



小心: **V 9.1.0** **z/OS**

在 z/OS 上發出 `setmqweb` 或 `dspmqweb` 指令之前，您必須先設定 `WLP_USER_DIR` 環境變數，讓變數指向您的 mqweb 伺服器配置。

若要這樣做，請發出下列指令：

```
export WLP_USER_DIR=WLP_user_directory
```

其中, `WLP_user_directory` 是傳遞至 `crtmqweb` 的目錄名稱。例如:

```
export WLP_USER_DIR=/var/mqm/web/installation1
```

如需相關資訊, 請參閱[建立 mqweb 伺服器](#)。

關於這項作業

當您配置 REST API for MFT 逾時時, 請注意, 第一次嘗試重新建立連線是在與協調佇列管理程式的連線中斷之後立即進行。如果失敗, 則每次嘗試重新連線之間的時間為 5 分鐘。因此, 設定 0-5 之間的值只會導致一次嘗試重新連接。

在重新連線逾時之後, 會在呼叫任何 REST API for MFT 資源時進行下一次嘗試重新連接。如果此重新連線嘗試失敗, MFT 會每五分鐘再次嘗試重新連接一次, 直到重新連接逾時為止。

程序

- 使用下列指令來檢視 REST API for MFT 的現行配置:

```
dspmweb properties -a
```

`mqRestMftEnabled` 欄位顯示是否啟用 REST API for MFT。如果已啟用 messaging REST API, 則此值為 `True`, 否則為 `False`。

`mqRestMftCoordinationQmgr` 欄位顯示協調佇列管理程式的名稱。

V 9.1.2 `mqRestMftCommandQmgr` 欄位顯示指令佇列管理程式的名稱。

`mqRestMftReconnectTimeoutInMinutes` 欄位會顯示重新連線逾時值, 直到「MFT 傳送 REST」服務停止嘗試連接至協調佇列管理程式為止。如需相關資訊, 請參閱[dspmweb](#)。

- 配置是否啟用 REST API for MFT:

- a) 輸入下列指令, 以配置是否啟用 REST API for MFT:

```
setmqweb properties -k mqRestMftEnabled -v value
```

其中, 如果您想要啟用 REST API for MFT, 則值為 `true`, 否則為 `false`。

- b) 透過輸入下列指令, 重新啟動 mqweb 伺服器:

```
endmqweb  
startmqweb
```

- 配置從中擷取傳送詳細資料的協調佇列管理程式:

- a) 使用下列指令來配置協調佇列管理程式:

```
setmqweb properties -k mqRestMftCoordinationQmgr -v qmgrName
```

其中 `qmgrName` 是協調佇列管理程式的名稱。協調佇列管理程式必須位於 mqweb 伺服器執行所在的機器上。依預設, 此佇列管理程式名稱為空白。如果未設定值, 則 MFT 的 REST API 無法運作。

- b) 透過輸入下列指令, 重新啟動 mqweb 伺服器:

```
endmqweb  
startmqweb
```

- 配置逾時 (分鐘), 在此之後, REST API for MFT 會停止嘗試連接至協調佇列管理程式:

- a) 使用下列其中一個指令來配置逾時:

- 設定逾時:

```
setmqweb properties -k mqRestMftReconnectTimeoutInMinutes -v time
```

其中 `time` 指定發生逾時之前的時間 (分鐘)。

如果此值設定在 0-5 之間, 則 REST API for MFT 只會嘗試重新連接至協調佇列管理程式一次。如果連線失敗, 則在呼叫 REST API 之前不會嘗試重新建立連線。

如果此值設為 -1，則 REST API for MFT 會嘗試重新連接，直到連線成功為止。

- 將逾時重設為預設值 30 分鐘：

```
setmqweb properties -k mqRestMftReconnectTimeoutInMinutes -d
```

- b) 透過輸入下列指令，重新啟動 mqweb 伺服器：

```
endmqweb  
startmqweb
```

- **V 9.1.2** 配置建立要求的指令佇列管理程式：

- a) 使用下列指令來配置指令佇列管理程式：

```
setmqweb properties -k mqRestMftCommandQmgr -v qmgrName
```

其中 *qmgrName* 是指令佇列管理程式的名稱。指令佇列管理程式必須位於 mqweb 伺服器執行所在的機器上。依預設，此佇列管理程式名稱為空白。如果未設定值，則 create 指令的 MFT REST API 無法運作。

- b) 透過輸入下列指令，重新啟動 mqweb 伺服器：

```
endmqweb  
startmqweb
```

V 9.1.0 調整 mqweb 伺服器 JVM

依預設，mqweb 伺服器 Java 虛擬機器 (JVM) 會針對配置參數使用平台專用預設值，例如資料堆大小下限及上限，以及類別快取大小。

關於這項作業

您可能需要變更預設值，以增進效能或解決問題。例如，如果 mqweb 伺服器擲出 `java.lang.OutOfMemoryError`，則您必須增加資料堆大小上限。如果您嘗試載入大量佇列物件，也應該增加資料堆的大小。

如果在 IBM MQ Console 中顯示儀表板配置資訊時遇到問題，您必須設定一個變數來決定配置的檔案編碼。您可以變更 `jvm.options` 檔案中的預設值。

程序

1. 開啟 `jvm.options` 檔案。

您可以在下列其中一個目錄中找到 `jvm.options` 檔：

- **ULW** 在 UNIX, Linux, and Windows 上: `MQ_DATA_PATH/web/installations/installationName/servers/mqweb`
- **z/OS** 在 z/OS 上: `WLP_user_directory/servers/mqweb`

其中 `WLP_user_directory` 是在執行 `crtmqweb` Script 以建立 mqweb 伺服器定義時指定的目錄。

2. 選擇性的: 將下列行新增至檔案，以設定資料堆大小上限：

```
-XmxMaxSizem
```

其中 `MaxSize` 指定資料堆的大小上限 (MB)。

例如，下列行將資料堆大小上限設為 1GB：

```
-Xmx1024m
```

3. 選擇性的: 將下列行新增至檔案，以設定資料堆大小下限：

```
-XmsMinSizem
```

其中 *MinSize* 指定資料堆的大小下限 (MB)。從預設值增加資料堆大小下限可以減少啟動 mqweb 伺服器所花費的時間。

例如，下列行將資料堆大小下限設為 512MB:

```
-Xms512m
```

4. 選擇性的: 將下列行新增至檔案，以設定類別快取的大小:

```
-XscmxSize
```

其中 *Size* 指定類別快取大小 (以 MB 為單位)。

例如，下列行將類別快取大小設為 100MB:

```
-Xscmx100m
```

Java 共用類別快取用來儲存資料，例如載入的類別和 AOT (AOT) 編譯程式碼。

類別快取可大幅減少啟動 mqweb 伺服器所花費的時間。第一次啟動 mqweb 伺服器時，會建立類別快取，且伺服器可能花費大量時間來啟動。後續重新啟動伺服器會更快速，因為可以從共用類別快取載入類別。

從預設值增加類別快取的大小可以減少啟動 mqweb 伺服器所花費的時間。

z/OS 當在不同的 z/OS 系統上啟動 mqweb 伺服器時，會重建類別快取。因此，在 sysplex 中不同 z/OS 系統上啟動 mqweb 伺服器所花費的時間比在相同系統上重新啟動伺服器所花費的時間要長得多。

請注意，只有在建立類別快取時，此值的變更才會生效。類別快取是在第一次啟動 mqweb 伺服器時，或在使用 Java 類別快取公用程式毀損類別快取之後建立。

5. 選擇性的: 將下列行新增至檔案，以設定 IBM MQ Console 中用於使用者儀表板配置資訊的檔案編碼:

```
-Dfile.encoding=UTF-8
```

6. 重新啟動 mqweb 伺服器。

z/OS 在 z/OS 上，停止並重新啟動 mqweb 伺服器已啟動作業。

Multi 在所有其他平台上，在指令行上輸入下列指令:

```
endmqweb  
strmqweb
```

IBM MQ Console 和 REST API 安裝元件的檔案結構


有兩組目錄結構與 IBM MQ Console 和 REST API 安裝元件相關聯。一個目錄結構包含可編輯的檔案。另一個目錄結構包含無法編輯的檔案。

可編輯檔案

使用者可編輯檔案會作為 IBM MQ Console 和 REST API 安裝元件起始安裝的一部分來放置。因為可以編輯這些檔案，所以在套用維護時不會變更這些檔案。

使用者可編輯檔案的位置視作業系統而定:

- **ULW** 在 UNIX, Linux, and Windows 上: `MQ_DATA_PATH/web/installations/installationName/`
- **IBM i** 在 IBM i 上: `MQ_DATA_PATH/web/installations/Installation1/`
- **z/OS** 在 z/OS 上: `WLP_user_directory`

其中 `WLP_user_directory` 是在執行  `crtmqweb` Script 以建立 mqweb 伺服器定義時指定的目錄。




在這個最上層目錄之下，會呈現下列目錄和檔案：

目錄和檔案	說明
<code>angular.persistence/</code>	儲存 IBM MQ Console 儀表板配置的目錄。
<code>servers/</code>	WebSphere Liberty 設定檔伺服器目錄。
<code>servers/mqweb</code>	包含 mqweb 伺服器目錄結構的目錄。
<code>servers/mqweb/logs</code>	包含 mqweb 伺服器日誌的目錄。
<code>servers/mqweb/logs/console.log</code>	基本伺服器狀態及作業訊息的日誌。
<code>servers/mqweb/logs/ffdc</code>	首次失敗資料擷取 (FFDC) 輸出目錄。
<code>servers/mqweb/logs/messages.log</code>	mqweb 伺服器的執行時期訊息日誌，包括 IBM MQ Console 和 REST API。較舊的訊息儲存在稱為 <code>messages_timestamp.log</code> 的檔案中。
<code>servers/mqweb/logs/trace.log</code>	mqweb 伺服器的追蹤日誌，包括 IBM MQ Console 和 REST API。較舊的追蹤儲存在稱為 <code>trace_timestamp.log</code> 的檔案中。只有在啟用追蹤時，這些檔案才會存在。
<code>servers/mqweb/logs/state</code>	伺服器特定狀態。
<code>servers/mqweb/server.xml</code>	主要伺服器配置檔。 此檔案是唯讀的。編輯 <code>mqwebuser.xml</code> 檔以置換預設配置。
<code>servers/mqweb/mqwebuser.xml</code>	IBM MQ Console 和 REST API 的配置檔。在此檔案中配置的設定會置換預設配置。 您必須是 <u>特許使用者</u> 才能編輯此檔案。
<code>servers/mqweb/resources</code>	包含各種伺服器資源 (例如金鑰儲存庫) 的目錄。
<code>servers/mqweb/workarea</code>	伺服器在運作時所建立的目錄。此目錄是在第一次執行伺服器之後建立。

不可編輯的檔案

不可編輯的檔案會作為 IBM MQ Console 和 REST API 安裝元件起始安裝的一部分而放置。當套用維護時，會更新這些檔案。

使用者可編輯檔案的位置視作業系統而定：

-  在 UNIX, Linux, and Windows 上: `MQ_INSTALLATION_PATH/web`
-  在 IBM i 上: `MQ_INSTALLATION_PATH/web`
-  在 z/OS 上: `installation_directory/web/`

其中 `installation_directory` 是 IBM MQ UNIX 系統服務元件安裝路徑。

此位置中存在下列目錄結構及檔案：

目錄和檔案	說明
<code>bin/</code>	包含 Liberty 指令的目錄。

目錄和檔案	說明
	您必須是 特許使用者 ，才能在此目錄中執行 Script。
mq/	包含各種 IBM MQ 資源的目錄結構。
mq/apps/	包含 IBM MQ Console 和 REST API 應用程式的目錄。
mq/etc/	
mq/etc/mqweb.xml	mqweb 伺服器的唯讀配置檔。 編輯 mqwebuser.xml 檔案以進行配置變更。
mq/libs	包含供 IBM MQ Console 和 REST API 使用的共用程式庫的目錄。
mq/samp	包含範例的目錄。
mq/samp/configuration	包含範例配置檔的目錄，這些配置檔可以複製到 mqwebuser.xml 檔。

z/OS V9.1.0 在 z/OS 上配置 mqweb 伺服器使用情形記錄

為了判定產品使用情形，z/OS 系統會記錄管理 IBM MQ Console 和 REST API 的 mqweb 伺服器所使用的處理器時間量。mqweb 伺服器可能需要配置為根據正確的產品 ID 來記錄使用情形。

開始之前

您必須具有卸載及裝載檔案系統的權限，以及停止並重新啟動 mqweb 伺服器已啟動作業，才能完成此程序。

關於這項作業

依預設，mqweb 伺服器使用情形記錄為獨立式 IBM MQ for z/OS 產品，產品 ID 為 5655-MQ9。如果您已獲授權使用 IBM MQ for z/OS Value Unit Edition (VUE) 或 IBM MQ Advanced for z/OS Value Unit Edition，則必須遵循此程序來變更針對其記錄 mqweb 伺服器使用情形的產品 ID。

在下列所有步驟中，請替換下列各項：

- WLP_USER_DIR，具有最初在 mqweb 伺服器 JCL 程序中傳遞至 **crtmqweb** (也稱為 'USERDIR') 的 Liberty 使用者目錄名稱。
- PathPrefix，具有 IBM MQ for z/OS UNIX 系統服務 (USS) 元件安裝路徑的名稱。

程序

1. 在 USS 中，於 WLP_USER_DIR 目錄中建立新目錄 'mqweb_extension'。

例如：

```
mkdir /var/mqm/web/installation1/mqweb_extension
```

2. 在 WLP_USER_DIR/mqweb_extension 目錄中建立稱為 mqweb.properties 的 ASCII 檔案。
3. 編輯 mqweb.properties 檔案，以併入下列這一行：

```
com.ibm.websphere.productInstall= WLP_USER_DIR/mqweb_extension
```

此外，請新增下列三行中的其中一行，以參照所記錄產品的 PID：

```
com.ibm.websphere.productId=com.ibm.mq, 或
com.ibm.websphere.productId=com.ibm.mqvue, 或
com.ibm.websphere.productId=com.ibm.mqadv
```

- 在 USS 中，建立 WLP_USER_DIR/mqweb_extension/lib/features/目錄。
例如，使用 USS 指令

```
mkdir -p WLP_USER_DIR/mqweb_extension/lib/features/
```

- 在 USS 中，建立 WLP_USER_DIR/mqweb_extension/lib/versions/目錄。
例如，使用 USS 指令

```
mkdir -p WLP_USER_DIR/mqweb_extension/lib/versions/
```

- 將下列其中一個檔案從 PathPrefix/web/mq/etc 複製到 WLP_USER_DIR/mqweb_extension/lib/versions。

您需要複製的檔案取決於您要配置使用情形記錄的產品類型。

mq.properties

mqweb 伺服器使用情形記錄為獨立式 IBM MQ for z/OS 產品，產品 ID 為 5655-MQ9。

mqVue.properties

mqweb 伺服器使用情形記錄為獨立式 IBM MQ for z/OS Value Unit Edition (VUE) 產品，產品 ID 為 5655-VU9。

mqAdvancedVue.properties

mqweb 伺服器使用情形記錄為 IBM MQ Advanced for z/OS Value Unit Edition 產品的一部分，產品 ID 為 5655-AV1。

- 如果要記錄的 mqweb 伺服器用法是針對產品 ID 5655-VU9 或 ID 5655-AV1，請編輯 WLP_USER_DIR/mqweb_extension/lib/versions 目錄中的內容檔。
透過取代行來執行此動作

```
com.ibm.websphere.productReplaces=com.ibm.websphere.appserver.zos
```

具有線條

```
com.ibm.websphere.productReplaces=com.ibm.mq
```

- 在 WLP_USER_DIR/servers/mqweb 目錄中，編輯檔案 server.env 並新增下列行：

```
WLP_PRODUCT_EXT_DIR=WLP_USER_DIR/mqweb_extension
```

- 請變更 Liberty 使用者目錄中目錄及檔案的所有權及許可權，讓它們屬於 mqweb 伺服器執行所在的使用者 ID 及群組。
使用下列指令：

```
chown -R userid:group WLP_USER_DIR  
chmod -R 770 WLP_USER_DIR
```

- 如果 mqweb 伺服器正在執行中，請在 mqweb 伺服器啟動型作業上使用 MVS **STOP** 指令來停止伺服器。
- 搭配使用 TSO/E **MOUNT** 指令與 **MODE (RDWR)** 參數，以裝載 IBM MQ for z/OS UNIX 系統服務元件檔案系統以取得讀寫權。
- 從 PathPrefix/web/lib/versions 目錄中刪除下列所有檔案：
 - mq.properties
 - mqVue.properties
 - mqAdvancedVue.properties
- 將 PathPrefix/web/mq/etc/mq.properties 複製到 PathPrefix/web/lib/versions 目錄。
視所記錄的產品而定，可能會置換這個預設內容檔，但它是必要的。
- 搭配使用 TSO/E **MOUNT** 指令與 **MODE (READ)** 參數，以裝載 IBM MQ for z/OS UNIX 系統服務元件檔案系統以進行唯讀存取。
- 使用 MVS **START** *procname* 指令啟動 mqweb 伺服器，其中 *procname* 是 mqweb 伺服器啟動作業程序的名稱。

16. 在 `messages.log` 的 `WLP_USER_DIR/servers/mqweb/logs` 目錄中，`CWWKB0108I` 訊息會驗證正在記錄的產品。

例如：

```
CWWKB0108I: IBM CORP product MQM MVS/ESA version V9 R1.0 successfully registered with z/OS.
```

V 9.1.4 MQ Adv. Linux MQ Adv. VUE 在 Linux 上定義 Aspera gateway 連線

IBM Aspera fasp.io Gateway 提供快速 TCP/IP 通道，可大幅增加 IBM MQ 的網路傳輸量。在任何授與的 CD 平台上執行的佇列管理程式可以透過 Aspera gateway 進行連接。閘道本身部署在 Red Hat 上或 Ubuntu Linux。

關於這項作業

Aspera gateway 可用來改善佇列管理程式通道的效能。如果網路具有高延遲或容易遺失封包，且通常用來加速不同資料中心的佇列管理程式之間的連線，則特別有效。

註：對於不會遺失封包的快速網路，使用 Aspera gateway 時效能會降低，因此在定義 Aspera gateway 連線之前及之後檢查網路效能是很重要的。

您在 IP 網路連線的每一端定義一個 Aspera gateway，然後使用 TCP/IP 將佇列管理程式通道連接至每一個閘道。佇列管理程式不需要在與其使用的 Aspera gateway 相同的機器上執行，且多個佇列管理程式可以使用相同的閘道。唯一的限制如下：

若要使用 Aspera gateway，您必須具有下列一或多個授權：

- IBM MQ Advanced for Multiplatforms
- IBM MQ Appliance
- **V 9.1.5** IBM MQ Advanced for z/OS VUE

您可以在下列任何 Linux (Red Hat 或 Ubuntu) 平台上部署 Aspera gateway：

- Linux for x86-64
- **V 9.1.5** Linux on POWER Systems - Little Endian
- **V 9.1.5** Linux for IBM Z

除非閘道個別獲得授權，否則 Aspera gateway 的使用限制為 IBM MQ 訊息。

使用 Aspera gateway 的佇列管理程式可以在 Continuous Delivery (CD) 版本支援的任何授權平台上執行。如需 CD 平台的完整清單，請參閱 [產品說明文件中的版本和平台圖示](#)。

對於不是位於與其使用之 Aspera gateway 相同的機器上的每一個佇列管理程式，請檢查佇列管理程式與 Aspera gateway 之間是否有快速網路連線。

您可以使用 `toml` 檔案來建立閘道定義，以定義閘道使用的入埠及出埠。Aspera gateway 隨附範例 `toml` 檔。出埠閘道定義定義從本端佇列管理程式到閘道的連線，以及從本端閘道到遠端閘道的連線。入埠閘道定義定義從遠端閘道到本端閘道的連線，以及從本端閘道到本端佇列管理程式的連線。

下列步驟提供入門及執行的基本手冊。如需詳細資訊，請參閱 [IBM Aspera fasp.io Gateway V1.0.0 說明文件](#)。

程序

1. 取得 Aspera gateway 安裝映像檔。

如果您的企業具有 IBM MQ Advanced for Multiplatforms 或 IBM MQ Appliance 授權，則可以從 Passport Advantage 下載 Aspera gateway，例如 "IBM MQ V9.1.x Continuous Delivery Release for IBM Aspera fasp.io eAssembly"。若要下載此 eAssembly，請跳至 [下載 IBM MQ 9.1](#)，然後按一下最新持續交付版本的標籤。eAssembly 包含閘道可用之所有 Linux 平台的安裝映像檔。

V9.1.5 如果您的企業具有 IBM MQ Advanced for z/OS VUE 授權，您可以從 SMP/E 安裝作業的「連接器套件」元件中取得 Aspera gateway。如需相關資訊，請參閱 IBM MQ Advanced for z/OS VUE 程式目錄 (IBM MQ for z/OS 程式目錄 PDF 檔案)。安裝「連接器套件」時，它會在 Unix 系統服務中建立 fasp 目錄，其中包含 .zip 檔案。 .zip 檔包含可使用閘道之所有 Linux 平台的安裝映像檔。請注意，Aspera gateway 無法原生在 z/OS 上執行。

2. 將 Aspera gateway 安裝映像檔複製到將執行閘道的兩部機器上，然後解壓縮並安裝閘道。

使用 RPM Package Manager (RPM) 安裝：

```
rpm -ivh ibm-fasp.io-gateway-1.0.0_qa_48-1.x86_64.rpm
```

若要在 Ubuntu 上使用 RPM 安裝，您有兩個選項：

- 將 --force-debian 屬性新增至 Aspera gateway 安裝指令，如 [使用 rpm 在 Linux 上安裝 IBM MQ 用戶端](#) 中所述。
- 使用 apt-get 指令：

```
sudo apt-get install ./ibm-fasp.io-gateway_1.0.0_amd64.deb
```

3. 配置每一個閘道。

修改安裝所建立之 /etc/fasp.io 目錄中的 gateway.toml 及 logging.toml 檔案。使用 gateway.toml 檔案來定義閘道使用的入埠及出埠埠，並使用 logging.toml 檔案來定義您需要的記載層次。本主題稍後提供編輯 gateway.toml 檔案的範例。

4. 在網路連線的每一端，變更通道定義以連接至本端閘道所接聽的埠。
5. 啟動每一個閘道服務。
從命令提示字元，執行下列指令：

```
systemctl start fasp.io-gateway
```

6. 重新啟動通道。

您的佇列管理程式現在正在透過 Aspera gateway 連線進行通訊。

範例

此範例定義執行 Linux 的兩部機器上的 Aspera gateway 連線。配置如下所示：

- 本端閘道機器的 IP 位址為 9.20.193.107。遠端閘道機器的 IP 位址為 9.20.192.115。
- 本端佇列管理程式正在 IP 位址為 9.20.121.5 的機器上執行。遠端佇列管理程式正在 IP 位址為 9.20.121.25 的機器上執行。兩個佇列管理程式都在埠 1414 上接聽。
- 本端佇列管理程式上的佇列管理程式通道會變更為使用 **conname 9.20.193.107(1500)** 連接至本端 Aspera gateway。遠端佇列管理程式上的佇列管理程式通道會變更為使用 **conname 9.20.192.115(1500)** 連接至遠端 Aspera gateway。

1. 在本端閘道機器上定義 Aspera gateway 連線：

- 安裝 Aspera gateway：

```
rpm -ivh ibm-fasp.io-gateway-1.0.0_qa_48-1.x86_64.rpm
```

- 修改安裝所建立 /etc/fasp.io 目錄中的 gateway.toml 檔案。編輯它以設定本端閘道定義。

```
[[bridge]]
  name = "Outbound"
  [bridge.local]
    protocol = "tcp"
    host = "9.20.193.107"
    port = 1500

  [bridge.forward]
    protocol = "fasp"
    host = "9.20.192.115"
    port = 1600
```

```
[[bridge]]
  name = "Inbound"
  [bridge.local]
    protocol = "fasp"
    host = "9.20.193.107"
    port = 1600

  [bridge.forward]
    protocol = "tcp"
    host = "9.20.121.5"
    port = 1414
```

2. 重複前一個步驟，以在遠端閘道機器上定義 Aspera gateway 連線。修改安裝所建立 /etc/fasp.io 目錄中的 gateway.toml 檔案。編輯它以設定遠端閘道定義：

```
[[bridge]]
  name = "Outbound"
  [bridge.local]
    protocol = "tcp"
    host = "9.20.192.115"
    port = 1500

  [bridge.forward]
    protocol = "fasp"
    host = "9.20.193.107"
    port = 1600

[[bridge]]
  name = "Inbound"
  [bridge.local]
    protocol = "fasp"
    host = "9.20.192.115"
    port = 1600

  [bridge.forward]
    protocol = "tcp"
    host = "9.20.121.25"
    port = 1414
```

3. 在連線的每一端，變更通道定義，以連接至本端閘道所接聽的埠。
 - 使用 **conname** 9.20.193.107(1500)，變更本端佇列管理程式上的佇列管理程式通道，以連接至本端 Aspera gateway。
 - 使用 **conname** 9.20.192.115(1500) 變更遠端佇列管理程式上的佇列管理程式通道，以連接至遠端 Aspera gateway。
4. 在本端閘道機器上執行下列指令，以啟動本端閘道：

```
systemctl start fasp.io-gateway
```

5. 在遠端閘道機器上執行下列指令，以啟動遠端閘道：

```
systemctl start fasp.io-gateway
```

6. 重新啟動通道。

下一步

Aspera gateway 會傳遞它所接收的資料，而不會以任何方式解譯它。這表示您可以在使用 Aspera gateway 的佇列管理程式通道之間配置 TLS，因為閘道連線不知道 TLS 信號交換。這也表示任何受支援 IBM MQ 平台上的佇列管理程式都可以使用 Aspera gateway。

若要將多重實例佇列管理程式與閘道搭配使用，請為佇列管理程式的每一個實例配置閘道定義。

註：Aspera gateway 僅已使用佇列管理程式通道進行測試。它尚未經過用戶端通道的測試。這是因為 Aspera gateway 的預期用途是用於透過慢速網路來連接遠端佇列管理程式，而用戶端應用程式一般是透過快速網路連接至本端資料中心中的佇列管理程式。

相關參考

[Aspera gateway 導覽圖](#)

[第 14 頁的『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 IBM MQ MQI client 與伺服器平台的組合。
[IBM Aspera fasp.io Gateway V1.0.0 說明文件](#)

V 9.1.0 Multi 配置 IBM MQ 以與 IBM Cloud Private 計量服務搭配使用

配置 IBM MQ 以與 IBM Cloud Private 計量服務搭配使用，以報告及檢視佇列管理程式啟動及使用資訊。

開始之前

在配置 IBM MQ 佇列管理程式以使用 IBM Cloud Private 服務之前，您必須具有 IBM Cloud (formerly Bluemix) 帳戶。若要建立帳戶，請參閱 [註冊 IBM Cloud](#)。

關於這項作業

透過使用 IBM Cloud Private 計量服務，您可以將內部部署 IBM 產品連接至 IBM Cloud Private 中的服務實例，並在單一儀表板中查看組織中的所有已登錄產品。

您可以配置 AIX、Linux 及 Windows 佇列管理程式，並將其連接至計量服務實例，然後查看其啟動及使用情形資訊。不過，在 Linux Container 環境以外的平台上，無法使用資料來支援「每小時儲存器型」計價授權。

V 9.1.1 若要記錄每月 VPC 授權類型的用量資料，而不是預設每小時授權度量值，請設定環境變數 `AMQ_LICENSEING_METRIC=VPCMonthlyPeak`。這會導致佇列管理程式上傳與每月 VPC 授權類型相關的資料，而不是上傳與每小時儲存器型授權相關的資料的預設行為。

將下列屬性與 `qm.ini` 檔案中的 `ReportingService` (舊稱為 `BluemixRegistration`) 段落搭配使用：

APIKeyFile

具有計量服務實例 **APIKey** 值的文字檔位置。

CapacityReporting

以下列格式將錯誤日誌訊息定期寫入 AMQERR 日誌：

```
4/22/2018 01:44:29 PM - Process(1274.1) User(bld-adm) Program(amqmgr0)
Host(8b3b83f2bc7d) Installation(Docker)
VRMF(9.1.0.0)
Time(2018-04-22T13:44:29.295Z)
ArithInsert1(300)
CommentInsert1(8.5)
CommentInsert2(IBM MQ Advanced)
```

CapacityReporting 屬性所產生的資訊會插入訊息 AMQ5064 中，這可讓您更瞭解您企業使用的 IBM MQ 程度：

AMQ5064

此佇列管理程式已執行 300 秒。目前正在使用 8.5 核心執行。授權類型為 IBM MQ Advanced。

嚴重性

0：資訊

說明

這是使用情形追蹤的參考訊息。

回應

無。

V 9.1.1 LicensingGroup

佇列管理程式所屬的計費群組。這會影響在計量服務產生的報告中對資料進行分組的方式。

ServiceURL

IBM Cloud Private 服務位址。

ServiceProxy

當佇列管理程式無法直接存取計量服務執行所在的網路時，可以使用的 HTTP Proxy 的 URL 和埠。

您可以查看產品安裝所在的主機、您正在使用的產品版本，以及它們執行所在的平台。從針對每一個產品顯示的高階使用情形度量值中，您可以取得工作量有多大的概觀。對於 IBM MQ，您可以查看哪些佇列管理程式使用較多，以及哪些佇列管理程式具有較少的工作量。

當佇列管理程式配置為連接至計量服務的實例時，會向 IBM Cloud Private 報告下列資訊：

- IBM MQ 佇列管理程式名稱
- IBM MQ 佇列管理程式 ID
- IBM MQ 安裝根目錄
- IBM MQ 已安裝元件 (名稱及版本)
- 主機名稱
- 主機作業系統名稱
- 主機作業系統版本
- IBM MQ 佇列管理程式的虛擬處理器核心 (VPC) 使用情形資訊

您可以在計量服務實例儀表中監視佇列管理程式 VPC 使用情形度量。

程序

- 配置佇列管理程式，以與 IBM Cloud Private 上的計量服務實例搭配使用。
- 透過 HTTP Proxy 連接至 IBM Cloud Private 計量服務。
- 對 IBM Cloud Private 計量服務的連線進行疑難排解。

相關概念

[虛擬處理器核心 \(VPC\) 的定價標準](#)

V 9.1.1 Multi 在 IBM Cloud Private 上配置佇列管理程式以與計量服務實例搭配使用

設定佇列管理程式的安全及 IBM Cloud 登錄資訊，然後連接至您已建立的計量服務實例。

關於這項作業

您的 IBM Cloud Private 計量服務 實例儀表板只會顯示配置為包含安全及 IBM Cloud Private 登錄資訊之佇列管理程式的資料。

程序

1. 請遵循 ICP 所記載的步驟，在下列位置建立服務 ID：
使用 [IBM Cloud Private CLI](#) 來建立服務 ID。
2. 遵循 ICP 所記載的步驟，以在下列位置建立 API 金鑰：
[API 金鑰管理 API](#)。
3. 從 ICP 叢集下載 TLS 憑證。
請記下您將憑證下載至其中的位置。您可以在步驟 [第 665 頁的『9』](#) 中，將下載的憑證新增至佇列管理程式的金鑰儲存庫。
4. 建立文字檔 `apikeyfile.txt`，並新增您在前一項作業中所複製的 **API key** 值。
記下 `apikeyfile.txt` 的位置，以便您可以在 [步驟 8](#) 中包括它的路徑。此檔案必須可供佇列管理程式使用者 (UNIX 系統上的 `mqm`) 讀取。該檔案必須僅包含 **API key** 本身，而不是 JSON 有效負載，例如 `d9c11b45-4dda-4de4-c0b2-2e4e1004dc64`。
5. 建立佇列管理程式，例如 `QM1`。
如需相關資訊，請參閱 [在 Multiplatforms 上建立及管理佇列管理程式](#)。
6. 啟動佇列管理程式 `QM1`。
如需相關資訊，請參閱 [啟動佇列管理程式](#)。

7. 在執行 IBM MQ 指令之前，請記得先設定 IBM MQ 指令行環境。

執行 **setmqenv** 指令。

AIX 在 AIX 上:

```
. /usr/mqm/bin/setmqenv -s
```

Linux 在 Linux 上:

```
. /opt/mqm/bin/setmqenv -s
```

Windows 在 Windows 上:

```
"C:\Program Files\IBM\MQ\bin\setmqenv.cmd" -n installation name
```

8. 為佇列管理程式 QM1 建立 SSL 信任儲存庫。

AIX 在 AIX:Amend for AIX 上開始建立信任儲存庫

```
runmqckm -keydb -create -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms  
-expire 30 -stash
```

Linux 在 Linux 上:

```
runmqckm -keydb -create -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms  
-expire 30 -stash
```

Windows 在 Windows 上:

```
runmqckm -keydb -create -db "MQ data directory\qmgrs\QM1\ssl\key.kdb" -pw password -type  
cms -expire 30 -stash
```

9. 將您在步驟 [第 664 頁](#) 的『3』中下載的數位憑證新增至佇列管理程式的信任儲存庫。

AIX 在 AIX 上:

```
runmqckm -cert -add -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms  
-label RootCA  
-file Download_location/RootCA.crt -format ascii -trust enable  
  
runmqckm -cert -add -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms  
-label ServerCert  
-file Download_location/CERT.crt -format ascii -trust enable
```

Linux 在 Linux 上:

```
runmqckm -cert -add -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms  
-label RootCA  
-file Download_location/RootCA.crt -format ascii -trust enable  
  
runmqckm -cert -add -db MQ data directory/qmgrs/QM1/ssl/key.kdb -pw password -type cms  
-label ServerCert  
-file Download_location/CERT.crt -format ascii -trust enable
```

Windows 在 Windows 上:

```
runmqckm -cert -add -db "MQ data directory\qmgrs\QM1\ssl\key.kdb" -pw password -type cms  
-label RootCA  
-file "Download_location\RootCA.crt" -format ascii -trust enable  
  
runmqckm -cert -add -db "C:\ProgramData\IBM\MQ\qmgrs\QM1\ssl\key.kdb" -pw password -type  
cms -label ServerCert  
-file "Download_location\CERT.crt" -format ascii -trust enable
```

10. 將含有 apikeyfile 路徑的新 ReportingService 段落新增至佇列管理程式的 qm.ini 檔案:

```
ReportingService:
  APIKeyFile=APIKey file location/apikeyfile.txt
```

11. 將 **API host** 值新增至 `qm.ini` 檔案。

ReportingService 段落區段現在包含 `apikeyfile` 及 **API host (ServiceURL)** 值的路徑:

```
ReportingService:
  APIKeyFile=APIKey file location/apikeyfile.txt
  ServiceURL=https://productinsights-api.ng.bluemix.net
```

儲存並結束 `qm.ini` 檔案。

12. 重新啟動佇列管理程式，讓變更生效。

系統可能會要求您授與佇列管理程式處理程序 `amqzmu10` 存取網路的許可權。需要存取權才能讓佇列管理程式聯絡計量服務。

13. 檢視計量服務實例中佇列管理程式 `QM1` 的相關資訊。

當報告狀態為作用中時，會向計量服務報告所指定整合節點上所有整合伺服器的啟動及使用情形資訊。使用情形資訊每 15 分鐘更新一次。

14. 選擇性的: 從佇列管理程式的 `qm.ini` 檔中移除 ReportingService 段落，並重新啟動佇列管理程式，以停止佇列管理程式向計量服務產生報告。

15. 選擇性的: 如果佇列管理程式無法向計量服務報告啟動或使用資訊，請檢查佇列管理程式日誌檔中的診斷資訊。

AIX 修正

AIX

在 AIX 上:

```
/var/mqm/qmgrs/QM1/errors/AMQERR0*.log
```

Linux

在 Linux 上:

```
/var/mqm/qmgrs/QM1/errors/AMQERR0*.log
```

Windows

在 Windows 上:

```
C:\ProgramData\IBM\MQ\errors\AMQERR0*.log
```

結果

您已建立計量服務實例，並將佇列管理程式配置成連接至實例。您可以在計量服務實例儀表板中查看佇列管理程式的相關資訊。

V 9.1.1

Multi

透過 HTTP Proxy 連接至 IBM Cloud Private 計量服務

如果佇列管理程式正在無法直接存取 ICP 叢集的系統上執行，您可以使用組織提供的 HTTP Proxy 來連接至 IBM Cloud Private 中的計量服務實例。

開始之前

您已配置安全，並將 **API key** 及服務 URL 新增至佇列管理程式的 `qm.ini` 檔。

關於這項作業

使用此作業來配置佇列管理程式，以透過組織提供的 HTTP Proxy 來連接至 IBM Cloud Private 中的 計量服務實例。

程序

- 將服務 Proxy 屬性新增至 `qm.ini` 檔的 IBM Cloud Private 登錄段落。

您可以設定 **ServiceProxy** 屬性，如下所示：

- 包含 `http://` 字首及選擇性地包含埠的 URL。如果未指定埠，則會使用 `1080`。

```
ReportingService:  
ServiceProxy=http://myorgproxy.net:1080
```

註：**ServiceProxy** 參數必須設為有效的 `http://` URL。不支援其他 Proxy 通訊協定，例如 HTTP 和 SOCKS。

- 請在變更生效之前重新啟動佇列管理程式。

V 9.1.1 Multi 對計量服務連線進行疑難排解

將佇列管理程式連接至計量服務實例時可能遇到的錯誤的疑難排解建議。

佇列管理程式無法向已配置的計量服務登錄或上傳用量度量值至已配置的計量服務

請檢查佇列管理程式是否具有網路的存取權。API 金鑰檔中的 **APIKey** 值不正確。確定已安裝 GSKit 元件。

qm.ini 段落無效

找到無效的 `qm.ini` 段落。如需相關資訊，請檢查錯誤日誌。

HTTP 服務 Proxy 參數無效

佇列管理程式 `ReportingService` 段落的 **ServiceProxy** 屬性值未正確配置。佇列管理程式不會向服務登錄。**ServiceProxy** 參數必須設為有效的 `http://` URL。不支援其他 Proxy 通訊協定，例如 HTTP 和 SOCKS。

Linux V 9.1.0 配置 IBM MQ 以與 Salesforce 推送主題及平台事件搭配使用

透過配置然後執行 IBM MQ Bridge to Salesforce，使用此資訊來設定 Salesforce 及 IBM MQ 網路的安全及連線。

開始之前

註：從 2022 年 11 月 22 日開始的所有版本都已淘汰 IBM MQ Bridge to Salesforce (請參閱 [美國公告信 222-431](#))。

- IBM MQ Bridge to Salesforce 在 Linux for x86-64 (64 位元) 上可用。橋接器不支援連接至在 IBM WebSphere MQ 6.0 及更早版本上執行的佇列管理程式。
- **V 9.1.2** IBM MQ 9.1.2 引進進一步的配置選項。主要變更是佇列管理程式現在可以支援多個橋接器實例，其中已適當地配置它們。如需進一步資訊，請參閱第 672 頁的『[IBM MQ Bridge to Salesforce 的其他配置選項](#)』。
- 安裝 **MQSeriesSFBridge** 套件。如需相關資訊，請參閱 [在 Linux 上安裝 IBM MQ 伺服器](#)。

關於這項作業

Salesforce 是雲端型客戶關係管理平台。如果您是使用 Salesforce 來管理客戶資料及互動，則可以使用 IBM MQ Bridge to Salesforce 來訂閱 Salesforce 推送主題及平台事件，然後可以將這些事件發佈至 IBM MQ 佇列管理程式。連接至該佇列管理程式的應用程式可以有用的方式耗用推送主題及平台事件資料。您也可以使用橋接器，在 Salesforce 中建立平台事件的事件訊息。

如需 IBM MQ Bridge to Salesforce 的概觀，請參閱 [圖 1](#) 中的圖表。

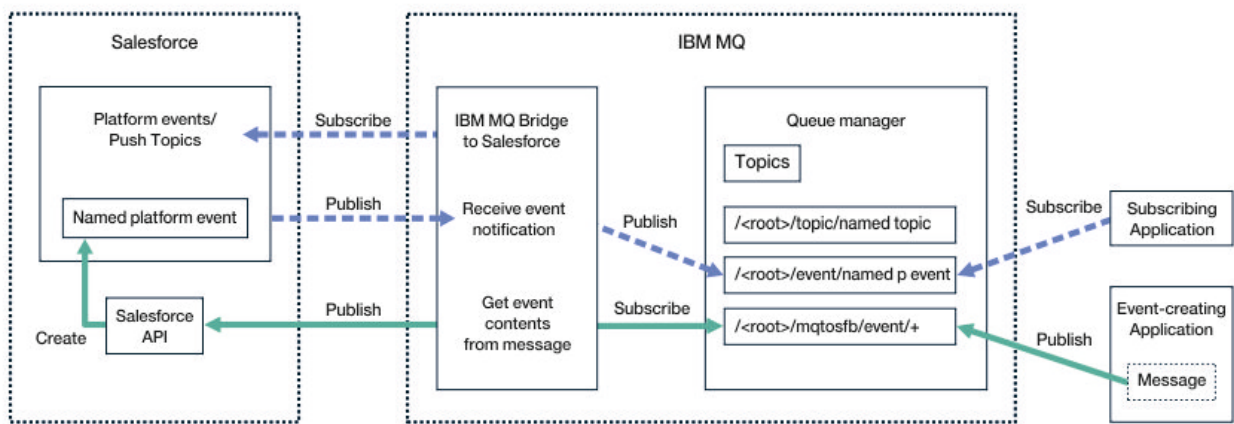


圖 97: IBM MQ Bridge to Salesforce

推送主題是您定義的查詢，用於使用 Force.com Streaming API 來接收 Salesforce 中記錄變更的通知。如需配置推送主題及如何使用 Streaming API 的相關資訊，請參閱 [Streaming API 簡介](#) 及 [使用 PushTopics](#)。

平台事件是可自訂的事件訊息，可定義來決定 Force.com 平台產生或耗用的事件資料。如需平台事件及 Salesforce 事件之間差異的相關資訊，請參閱 [企業傳訊平台事件](#) 及 [Salesforce 事件之間的差異](#)。

- 若要建立用於訂閱推送主題及平台事件的配置，請參閱第 669 頁的『[配置 IBM MQ Bridge to Salesforce](#)』。
- 若要建立配置以建立 Salesforce 平台事件的事件訊息，請參閱第 675 頁的『[建立 Salesforce 平台事件的事件訊息](#)』。

您可以透過 IBM MQ Console 及搭配使用 **-p** 參數與 **amqsrua** 指令，以兩種方式從橋接器監視資料。針對整體橋接器狀態發佈一組資料：

- 以間隔處理的推送主題訊息總數 (在 STATUS/PUSHTOPIC 樹狀結構下)。
- 在此間隔中看到的推送主題數。
- 以間隔處理的平台事件總數 (在 STATUS/PLATFORM 樹狀結構下)。
- 在此間隔中看到的平台事件數。
- 在一個間隔內 (在 STATUS/MQPE 樹狀結構下) 處理的 IBM MQ 已建立平台事件總數。
- 在此間隔中看到唯一數目的 IBM MQ 已建立平台事件。
- 在此間隔中看到的 IBM MQ 已建立平台事件發佈數失敗。

對於每一個已配置的 Salesforce 主題，會發佈進一步訊息。IBM MQ 主題使用完整 Salesforce 主題名稱，以及物件名稱中的 /event 或 /topic：

- 間隔內處理的訊息數。

若要配置 IBM MQ Console 以監視橋接器資料，請參閱 [配置 IBM MQ Bridge to Salesforce](#) 中的步驟 9 及 10。如需使用 **amqsrua** 指令的相關資訊，請參閱 [監視 IBM MQ Bridge to Salesforce](#)。

請遵循下列作業中的步驟來配置及執行 IBM MQ Bridge to Salesforce：

程序

1. 配置 IBM MQ Bridge to Salesforce。
2. **V9.1.0**
建立 Salesforce 平台事件的事件訊息。
3. 執行 IBM MQ Bridge to Salesforce。

相關工作

[追蹤 IBM MQ Bridge to Salesforce](#)

相關參考

[runmqsfb \(執行 IBM MQ Bridge to Salesforce\)](#)

Linux V 9.1.0 配置 IBM MQ Bridge to Salesforce

您可以配置 IBM MQ，並輸入 IBM MQ Bridge to Salesforce 參數來建立配置檔，以及將 Salesforce 推送主題及平台事件連接至 IBM MQ 佇列管理程式。

開始之前

在啟動此作業之前，請確定您已在 x86-64 Linux 平台上的 IBM MQ 安裝中安裝 MQSeriesSFBridge 套件。

關於這項作業

此作業會引導您完成建立 IBM MQ Bridge to Salesforce 配置檔所需的最小設定，並順利連接至 Salesforce 及 IBM MQ，以便您可以訂閱 Salesforce 推送主題及平台事件。如需所有參數的意義及選項的相關資訊，請參閱 [runmqsfb](#) 指令。您必須考量自己的安全需求，並自訂適合您部署的參數。

若要建立配置以建立 Salesforce 平台事件的事件訊息，請參閱 [第 675 頁的『建立 Salesforce 平台事件的事件訊息』](#)。

訂閱 Salesforce 推送主題及平台事件

當 IBM MQ Bridge to Salesforce 建立與 Salesforce 及 IBM MQ 的連線時，它會建立 Salesforce 推送主題及平台事件的訂閱。橋接器想要訂閱的推送主題或平台事件名稱必須包含在配置檔中，或在建立連線之前新增在指令行中。

其中一個配置屬性是 IBM MQ 主題樹狀結構的根目錄，並在此根目錄下發佈事件。橋接器會存取此根目錄，並新增完整 Salesforce 主題名稱，例如 /MQ/SF/ROOT/topic/EscalatedCases。監視主題及連接至 IBM MQ 的應用程式可能會在 /topic/EscalatedCases 下尋找推送主題，並在 /event/NewCustomer__e 下尋找平台事件。

已發佈的訊息包含控制資訊，以及包含所要求資料欄位的資料結構。對於推送主題，資料結構是 **subject**；對於平台事件，結構是 **payload**。如果未在 Salesforce 中定義主題或事件，則橋接器無法訂閱這些主題或事件。如果橋接器在嘗試訂閱主題時發生錯誤，則橋接器會停止。

主題物件不需要在 IBM MQ 中定義，但根據樹狀結構中最接近的母元素，必須存在適當的權限。依預設，重新發佈的訊息僅包含原始訊息中的相關資料結構。移除控制資訊。對於平台事件，發佈具有有效負載結構。**橋接器程式行為** 配置參數集中的 **Publish control data with the payload** 配置選項可讓您重新發佈整個訊息，包括控制資料。如需相關資訊，請參閱 [配置參數](#)。

從 Salesforce 發佈時，每一個推送主題及平台事件都有相關聯的 *ReplayID*。當建立與伺服器的連線時，可以使用 *ReplayID* 來要求發佈的起始點。Salesforce 會維護最多 24 小時的歷程，並容許橋接器不會遺失最近的推送主題及平台事件，即使它在產生時未啟動也一樣。橋接器支援兩種服務品質模式：

最多一次

橋接器不會使用 *ReplayId* 進行重新啟動。重新啟動橋接器之後，只會處理新產生的推送主題及平台事件。申請必須準備好處理遺失的出版品。橋接器仍會追蹤 *ReplayId* 並強化至佇列，因此橋接器可以使用其他服務品質重新啟動，並知道現行狀態。

至少一次

橋接器會追蹤 *ReplayId*，並強化至佇列。在重新啟動橋接器時，會使用持續保存的 *ReplayId* 向伺服器要求發佈的起始點。如果差距不超過 24 小時，則會傳送較舊的出版品。主題的 *ReplayId* 不會在每一則訊息上強化。它會定期寫入持續訊息中，並在橋接器關閉時寫入。應用程式必須準備好查看重複的出版品。

ReplayId 會以訊息形式寫入新定義的佇列。在啟動橋接器之前，您必須先定義此佇列

SYSTEM.SALESFORCE.SYNCQ。如果 **SYSTEM.SALESFORCE.SYNCQ** 不存在，則不論服務品質模式為何，橋接器都不會繼續。提供 MQSC Script 來建立具有相關屬性的佇列。必須使用 DEFSOPT (EXCL) NOSHARE 選項來配置佇列，以確保橋接器程式只有一個實例可以更新 **SYSTEM.SALESFORCE.SYNCQ** 佇列。

若要建立配置以建立平台事件的事件訊息，請參閱 [第 675 頁的『建立 Salesforce 平台事件的事件訊息』](#)。

程序

1. 建立並啟動佇列管理程式。

- a) 建立佇列管理程式，例如 SQM1。

```
crtmqm SQM1
```

- b) 啟動佇列管理程式。

```
strmqm SQM1
```

2. 註: 若要使用現有的登入及安全 Salesforce 認證和自簽憑證，請跳至步驟 [第 670 頁的『3』](#)。

選擇性的: 建立 Salesforce 帳戶的安全記號。

- a) 登入您的 Salesforce 帳戶。
- b) 遵循說明文章 [Salesforce 說明中的步驟建立或重設安全記號: 重設安全記號](#)。

3. 在 Salesforce 中建立 CA 簽章安全憑證。

- a) 從 **Force.com** 首頁的 **管理** 功能表中選取 **安全控制項**，然後選取 **憑證及金鑰管理**。即會開啟「憑證及金鑰管理」頁面。

- b) 按一下 **建立 CA 簽署憑證**。

即會開啟「憑證」頁面。

- c) 在 **標籤** 欄位中輸入憑證的名稱，按 Tab，然後按一下 **儲存**。

即會顯示「憑證及金鑰詳細資料」資訊。

- d) 按一下 **回到清單: 憑證和金鑰**。

- e) 按一下 **匯出至金鑰儲存庫**。

- f) 輸入金鑰儲存庫的密碼，然後按一下 **匯出**。

- g) 將匯出的金鑰儲存庫儲存至本端檔案系統。

4. 使用 IBM 金鑰管理 GUI 來開啟您從 Salesforce 匯出的金鑰儲存庫，並移入簽章者憑證。

- a) 執行 **strmqikm** 指令，以開啟 IBM 金鑰管理 GUI。

如需相關資訊，請參閱 [使用 runmqckm、runmqakm 及 strmqikm 來管理數位憑證](#)。

- b) 按一下 **開啟金鑰資料庫檔**，並瀏覽至 Salesforce 金鑰儲存庫的位置。

- c) 按一下 **開啟**，確定從 **金鑰資料庫類型** 選項中選取 **JKS**，然後按一下 **確定**。

- d) 輸入您在步驟 3f 中為金鑰儲存庫建立的密碼，然後按一下 **確定**。

- e) 從 **金鑰資料庫內容** 選項中選取 **簽章者憑證**。

- f) 按一下 **移入**。

- g) 從 **新增 CA 憑證** 清單中選取 **Verisign Inc.** 勾選框，然後按一下 **確定**。

5. 選擇性的: 在 Salesforce 帳戶中建立 IBM MQ Bridge to Salesforce 的應用程式連線，以產生 OAuth 消費者金鑰和密碼。

在正式作業環境中使用 IBM MQ Bridge to Salesforce 時，您需要 **消費者金鑰** 及 **消費者密碼** 代碼。

- a) 從 **Force.com** 首頁的 **建置** 功能表中選取 **建立**，然後選取 **應用程式**。

即會開啟「應用程式」頁面。

- b) 從 **已連接的應用程式** 區段中按一下 **新建**。

即會開啟「新建連接的應用程式」頁面。

- c) 在 **已連接的應用程式名稱** 中輸入 IBM MQ Bridge to Salesforce 的名稱，例如 **MQBridgeToSalesforce**。

- d) 輸入 **API 名稱**。

如果您按 Tab 鍵跳到下一個欄位，則會將 **已連接的應用程式名稱** 複製到 **API 名稱** 名稱欄位。

- e) 輸入 **聯絡人電子郵件**。

- f) 在 **API (啟用 OAuth 設定)** 區段中選取 **啟用 OAuth 設定** 選項。

然後會呈現該區段中的其他選項。

- g) 新增 **回呼 URL**，例如 `https://www.ibm.com`。
 - h) 在 **選取的 OAuth 範圍** 子區段中，從 **可用的 OAuth 範圍** 清單中選取 **完整存取權 (完整)** 選項，然後按一下 **新增**，將完整存取權新增至 **選取的 OAuth 範圍** 清單。
 - i) 按一下 **儲存**。
 - j) 按一下 **繼續**。
 - k) 記下 **消費者金鑰** 及 **消費者密碼** 代碼。
6. 在佇列管理程式上建立必要的同步化佇列。

```
cat /opt/mqm/mqsf/samp/mqsfbSyncQ.mqsc | runmqsc SQM1
```

同步化佇列會在應用程式或佇列管理程式重新啟動之間維護事件狀態。佇列深度可以較小，因為佇列上預期只有單一訊息。一次只能針對此佇列執行橋接器的一個實例，因此預設選項設定為專用存取。

7. 使用 IBM MQ、Salesforce 和 IBM MQ Bridge to Salesforce 行為的連線和安全參數來建立配置檔。

```
runmqsfb -o new_config.cfg
```

現有的值會顯示在方括弧內。按 Enter 以接受現有值，按 Space then Enter 以清除值，然後鍵入 Enter 以新增值。

- a) 輸入佇列管理程式 SQM1: 的連線值:

連線所需的最小值為佇列管理程式名稱、IBM MQ 基本主題根目錄及通道名稱。

```
Connection to Queue Manager
-----
Queue Manager or JNDI CF      : []SQM1
MQ Base Topic                 : []/sf
MQ Channel                   : []A channel you have defined or for example
SYSTEM.DEF.SVRCONN
MQ Conname                   : []
MQ Publication Error Queue   : [SYSTEM.SALESFORCE.ERRORQ]
MQ CCDT URL                   : []
JNDI implementation class    : [com.sun.jndi.fscontext.RefFSContextFactory]
JNDI provider URL            : []
MQ Userid                     : []
MQ Password                   : []
```

註: 如果您在本端連接，則不需要通道名稱。您不需要在配置檔中提供佇列管理程式名稱及基本主題，因為稍後當您執行橋接器時，可以將它們併入指令行。

- b) 輸入 Salesforce 連線的值:

連線所需的最小值為 Salesforce 使用者 ID、密碼、安全記號及登入端點。在正式作業環境中，您可以為 OAuth 安全新增消費者金鑰和密碼。

```
Connection to Salesforce
-----
Salesforce Userid (reqd)     : []salesforce_login_email
Salesforce Password (reqd)   : []salesforce_login_password
Security Token (reqd)       : []Security_Token
Login Endpoint               : [https://login.salesforce.com]
Consumer ID                  : []
Consumer Secret Key          : []
```

- c) 輸入 TLS 連線的憑證儲存庫值:

TLS 連線所需的最小值是 TLS 憑證及金鑰儲存庫密碼的金鑰儲存庫路徑。如果未提供信任儲存庫路徑或密碼，則金鑰儲存庫和密碼參數會用於信任儲存庫和密碼。如果您將 TLS 用於 IBM MQ 佇列管理程式連線，則可以使用相同的金鑰儲存庫。

```
Certificate stores for TLS connections
-----
Personal keystore for TLS certificates : []path_to_keystore, for example: /var/mqm/qmgrs/
SQM1/ssl/key.jks
Keystore password                   : []keystore_password
Trusted store for signer certificates : []
```

```
Trusted store password      : []
Use TLS for MQ connection  : [N]
```

d) 輸入值以配置 IBM MQ Bridge to Salesforce 的行為:

您不需要變更或提供任何這些值，但如果您知道推送主題或平台事件名稱，請在這裡新增它們。當您準備好執行橋接器時，也可以稍後在指令行中新增它們。您必須在配置檔或指令行中指定日誌檔。

```
Behaviour of bridge program
-----
PushTopic Names           : []
Platform Event Names      : []
MQ Monitoring Frequency   : [30]
At-least-once delivery? (Y/N) : [Y]
Subscribe to MQ publications for platform events? (Y/N) : [N]
Publish control data with the payload? (Y/N) : [N]
Delay before starting to process events : [0]
Runtime logfile for copy of stdout/stderr : []
```

8. 選擇性的: 建立 IBM MQ 服務以控制程式的執行。編輯範例 `mqsfbService.mqsc` 檔案，以指向新建立的配置檔，並對指令參數進行任何其他變更。

```
cat modified mqsfbService.mqsc | runmqsc SQM1
```

9. 選擇性的: 遵循 [開始使用 IBM MQ 主控台](#) 中的指示，以設定 IBM MQ Console。
10. 選擇性的: 將 IBM MQ Bridge to Salesforce 配置成以無根使用者身分執行。

若要能夠以無根使用者身分 (例如在無根儲存器內) 執行 IBM MQ Bridge to Salesforce，必須正確地設定 `Java userRoot` 及 `systemRoot` 目錄，以確保執行橋接器處理程序的使用者具有讀寫存取權。如果要這麼做，請設定下列 JVM 內容:

```
export MQSFB_EXTRA_JAVA_OPTIONS="-
Djava.util.prefs.userRoot=directory_with_read_write_access"
```

```
export MQSFB_EXTRA_JAVA_OPTIONS="-
Djava.util.prefs.systemRoot=directory_with_read_write_access"
```

結果

您已建立配置檔，供 IBM MQ Bridge to Salesforce 用來訂閱 Salesforce 推送主題及平台事件，並將它們發佈至 IBM MQ 網路。

下一步

完成第 680 頁的『[執行 IBM MQ Bridge to Salesforce](#)』的步驟。

相關工作

[追蹤 IBM MQ Bridge to Salesforce](#)

[監視 IBM MQ Bridge to Salesforce](#)

相關參考

[runmqsfb \(執行 IBM MQ Bridge to Salesforce\)](#)

Linux

V 9.1.2

IBM MQ Bridge to Salesforce 的其他配置選項

IBM MQ 9.1.2 引進其他配置選項，允許兩個主要類別的其他拓撲，處理「入埠」(從 Salesforce 產生，發佈至 IBM MQ 應用程式的事件)及「出埠」(IBM MQ 應用程式發佈傳送至 Salesforce 的事件)工作。此外，追蹤和記載的工作方式也有所變更。

IBM MQ 9.1.0 IBM MQ Bridge to Salesforce 中的變更

依預設，除了日誌檔現在開始輪換之外，IBM MQ 9.1.0 橋接器對行為沒有任何變更。如需相關資訊，請參閱第 673 頁的『[循環日誌](#)』。

主要變更是佇列管理程式現在支援多個橋接器實例。如果要啟用這項特性，以及其他拓撲的其餘部分，您必須進行一些手動配置變更。

如需其他配置選項的相關資訊，請參閱 [runmqfsb](#)；如需已修訂配置資訊的範例，請參閱 [第 674 頁的『配置輸出範例』](#)。

分隔入埠工作

橋接器的多個實例可以處理從 Salesforce 至 IBM MQ 的入埠工作，但這些實例必須處理獨立的 Salesforce 推送主題及事件集。否則，IBM MQ 應用程式可能會看到重複的事件，因為沒有跨橋接器通訊協定可停止複製事件。每一個實例都使用自己的可配置同步化佇列來保留 **ReplayId**。

在下列情況下，這可能很有用：

- 不同的 Salesforce 主題具有不同的安全授權。每一個橋接器實例都有一組不同的認證，用於存取 Salesforce。
- 您擔心來自 Salesforce 的工作量太多，單一橋接器無法處理。因此，您可以安排將主題分割成 "A-M" 透過一個橋接器，而 "N-Z" 透過另一個橋接器。

共用出埠工作

橋接器支援多個實例，以支援從 IBM MQ 傳送至 Salesforce 的出埠工作。如果橋接器的一個實例失敗，則訂閱相同佇列管理程式上相同主題的其他實例可以繼續處理發佈。

註：這不需要變更 IBM MQ 主題配置。

必須設定這些協同作業實例，以便最多其中一個實例處理來自 Salesforce 的入埠工作，因為該實例必須具有同步化佇列的互斥存取權。

當您對下列項目有疑慮時，這可能很有用：

- 來自 IBM MQ 的工作量。由於對 Salesforce 的要求是同步的，因此橋接器無法在仍處理一則訊息的情況下處理新工作。擁有多個消費者可以緩解這種情況。
- 可用性架構。例如，現在可以在個別資料中心執行多個實例，具有更好的失效接手及災難回復選項。作為 IBM MQ 用戶端執行也會區隔橋接器與佇列管理程式位置。

追蹤及除錯互動

除錯旗標會繼續像之前一樣運作。也就是說，`-d1` 會提供橋接器除錯資訊，而 `-d2` 會開啟必備元件的除錯記載。不過，如果您在啟動橋接器時已啟用 IBM MQ 追蹤，則會自動開啟 `-d2` 層次報告。

循環日誌

日誌檔的預設行為會變更為有三個日誌檔，每一個大小為 2 MB。您可以使用其他配置內容來置換這些值。日誌檔的現有配置屬性或指令行參數會作為日誌的基本名稱，並新增索引。

如果配置の日誌檔具有：

- 沒有檔案類型，索引會新增至檔名結尾。
將日誌檔設為 `abc` 會產生稱為 `abc.0`、`abc.1` 等的日誌。
- 檔案類型，索引會插入在檔案類型之前。
將日誌檔設為 `abc.log` 會產生稱為 `abc.0.log`、`abc.1.log` 等的日誌。

附註：

1. 由於橋接器可以使用任意使用者許可權來執行，因此無法強制日誌的特定目錄 (例如，`/var/mqm/qmgrs/<qm>/errors`)。
2. 相同的資訊會繼續寫入 `stdout` 和 `stderr` 串流。
3. 每當個別日誌檔重新開啟時，即會重新列印基本配置資訊。資訊將一律可用，而不是在程式開始時只列印一次。

保留日誌

新的拓撲更有可能針對特定佇列管理程式執行橋接器的多個實例。

為了避免實例彼此干擾，並避免改寫先前執行的橋接器，如果 .0 日誌已存在，橋接器將不會啟動。

您需要啟動程序，以在啟動橋接器之前刪除先前的日誌副本，或將類似時間戳記的內容新增至名稱。

相關工作

第 667 頁的『[配置 IBM MQ 以與 Salesforce 推送主題及平台事件搭配使用](#)』

透過配置然後執行 IBM MQ Bridge to Salesforce，使用此資訊來設定 Salesforce 及 IBM MQ 網路的安全及連線。

相關參考

[runmqfsb](#)

Linux V 9.1.2 配置輸出範例

配置輸出範例，顯示來自 IBM MQ 9.1.0 IBM MQ Bridge to Salesforce 的變更。

```
IBM MQ Bridge to Salesforce
5724-H72 (C) Copyright IBM Corp. 2017, 2024.
Level : <<unknown>>

Enter new values for the configuration attributes. The
current settings are shown.
Press ENTER to accept current values; use SPACE+ENTER
to clear values.

Connection to Queue Manager
-----
Queue Manager or JNDI CF      : [V9000_A]
MQ Base Topic                 : [/sf]
MQ Channel                    : []
MQ Conname                     : []
MQ Publication Error Queue    : [SYSTEM.SALESFORCE.DEADQ]
MQ Replay Status Queue       : [SYSTEM.SALESFORCE.SYNCQ]
MQ CCDT URL                   : []
JNDI implementation class     : [com.sun.jndi.fscontext.ReffSContextFactory]
JNDI provider URL            : []
MQ Userid                     : []
MQ Password                   : []

Connection to Salesforce
-----
Salesforce Userid (reqd)      : [johndoe@<yourenterprise>.com]
Salesforce Password (reqd)   : [*****]
Security Token                : [*****]
Login Endpoint                : [https://login.salesforce.com]
Consumer Key                  : [3MVG9HxRZv05HarQhSy89qSKYNr1gDcv1wE3zN5kyFAa4Wxt]
Consumer Secret              : [*****]

Certificate stores for TLS connections
-----
Personal keystore for TLS certificates : [/var/mqm/ssl/key.jks]
Keystore password                   : [*****]
Trusted store for signer certificates : []
Trusted store password              : []
Use TLS for MQ connection           : [N]

Event processing
-----
PushTopic Names                   : []
Platform Event Names              : []
At-least-once delivery for Salesforce events? (Y/N) : [N]
At-least-once delivery for MQ publications? (Y/N) : [N]
Subscribe to MQ publications for platform events? (Y/N) : [Y]
Publish control data with the payload? (Y/N) : [Y]
Treat unknown Salesforce topic as warning (Y/N) : [N]

Behaviour of bridge program
-----
Bridge unique identifier          : []
MQ Monitoring Frequency          : [30]
Delay before starting to process events : [0]
Continue to retry after maximum reconnection attempts (Y/N) : [N]
```

```
Runtime logfile for copy of stdout/stderr : [/tmp/runmqsfb.log]
Number of logfiles      : [3]
Maximum size of each logfile : [2097152]
Done.
```

相關參考

[runmqsfb](#)

Linux V 9.1.0 建立 Salesforce 平台事件的事件訊息

您可以配置 IBM MQ，並輸入 IBM MQ Bridge to Salesforce parameters 來建立配置檔，以及使用橋接器來建立 Salesforce 平台事件的事件訊息。

開始之前

- 您已在 x86-64 Linux 平台上的 IBM MQ 安裝中安裝 **MQSeriesSFBridge** 套件。

關於這項作業

此作業會引導您完成建立 IBM MQ Bridge to Salesforce 配置檔所需的最小設定，並順利連接至 Salesforce 及 IBM MQ，以便您可以建立 Salesforce 平台事件的事件訊息。如需所有參數的意義及選項的相關資訊，請參閱 [runmqsfb](#) 指令。您必須考量自己的安全需求，並自訂適合您部署的參數。

若要建立用於訂閱推送主題及平台事件的配置，請參閱 [第 669 頁的『配置 IBM MQ Bridge to Salesforce』](#)。

建立 Salesforce 平台事件的事件訊息

您可以使用 IBM MQ 應用程式來建立放置在佇列管理程式主題 `/root/mqtosfb/event/+` 上的訊息。橋接器會訂閱主題，從訊息取得內容，並使用它來發佈 Salesforce 平台事件的事件訊息。如需平台事件的相關資訊，請參閱 Salesforce 開發人員說明文件中的 [使用平台事件來遞送自訂通知](#)。

若要啟用橋接器來建立事件訊息，除了用於訂閱推送主題及平台事件的屬性之外，您還必須提供下列兩個屬性：

- 在 [佇列管理程式連線](#) 的橋接器配置屬性中，建立並新增 **MQ Publication Error Queue** 的名稱。
- 在橋接器配置屬性中，將 **Subscribe to MQ publications for platform events** 選項設為 Y，以定義 [橋接器程式的行為](#)。

您需要在 Salesforce 中建立平台事件並定義內容欄位，然後才能使用橋接器來建立該平台事件的事件訊息。平台事件名稱及其內容決定您需要如何格式化橋接器所處理的 IBM MQ 訊息。例如，如果 Salesforce 平台事件 **Object name** 是 `MQPlatformEvent1`，且您的兩個自訂定義欄位是具有 **API name** `MyText__c` 及 `Name__c` 的文字欄位，則在 `/root/mqtosfb/event/MQPlatformEvent1__e` 主題上發佈的 IBM MQ 訊息必須是正確格式化的 JSON，如下所示：

```
{ "MyText__c" : "Some text here", "Name__c" : "Bob Smith" }
```

訊息必須格式化，以便 IBM MQ Bridge to Salesforce 可以將它辨識為 MQFMT_STRING 格式化訊息內文。

請參閱步驟 [第 677 頁的『7』](#)，以在 Salesforce 中建立平台事件，或者如果您已有想要為其建立事件訊息的平台事件，請跳過此步驟。您必須格式化 IBM MQ 訊息，以符合 Salesforce 平台事件中設定的欄位。Salesforce 平台事件內的欄位可以指定為選用或必要。如需相關資訊，請參閱 Salesforce 開發人員說明文件中的 [平台事件欄位](#)。

當橋接器執行時，它會訂閱指定的 IBM MQ 主題。

- 如果您在橋接器配置中指定 **At-most-once** 服務品質，則橋接器所建立的訂閱不可延續。不會處理 IBM MQ 應用程式在橋接器未執行時所做的任何發佈。
- 如果您在橋接器配置中指定 **At-least-once** 服務品質，則橋接器所建立的訂閱是可延續的。這表示當橋接器不在執行中時，橋接器可以處理 IBM MQ 應用程式所建立的發佈。可延續訂閱需要已知的訂閱和用戶端 ID。橋接器使用 `D_SUB_RUNMQSFB` 作為訂閱名稱，並使用 `runmqsfb_1` 作為用戶端 ID。

如果橋接器用於訂閱 Salesforce 推送主題及平台事件，而不是用於建立事件訊息，則它會嘗試刪除可延續訂閱 (如果配置已變更，且訂閱現在是孤立的)。

您可以移除橋接器建立的可延續訂閱，如下所示：

使用 IBM MQ Explorer。

開啟橋接器所使用佇列管理程式的 **訂閱資料夾**，並尋找以 `:D_SUB_RUNMQSFB` 結尾的訂閱名稱，其中主題字串是 `/sf/mqtosfb/event+`。用滑鼠右鍵按一下訂閱名稱，然後按一下刪除。如果您收到指出訂閱正在使用中的錯誤，則您的橋接器可能仍在執行中。請停止橋接器，然後重試刪除訂閱。

使用 `runmqsc` 來尋找並刪除訂閱。

啟動 `runmqsc` 介面並執行 `DISPLAY SUB (*)`。尋找以 `:D_SUB_RUNMQSFB` 結尾的訂閱名稱 **SUB**。發出 `delete sub` 指令，並包含您要刪除之訂閱的 **SUBID**。例如：`DELETE SUB SUBID(414D5120514D3120202020202020205C589459987E8620)`

停止，然後使用 **At-most-once** 服務品質來啟動橋接器。

如果您以 **At-least-once** 服務品質 `At-least-once delivery? (Y/N) : [Y]` 啟動橋接器，則所建立的訂閱是可延續的。若要刪除訂閱，請在配置檔中將服務品質變更為 `At-least-once delivery? (Y/N) : [N]`，然後重新啟動橋接器。會刪除可延續訂閱，並建立不可延續訂閱。

程序

1. 建立並啟動佇列管理程式。

- a) 建立佇列管理程式，例如 PEQM1。

```
crtmqm PEQM1
```

- b) 啟動佇列管理程式。

```
strmqm PEQM1
```

2. **註：**若要使用現有的登入及安全 Salesforce 認證和自簽憑證，請跳至步驟 4。

選擇性的：建立 Salesforce 帳戶的安全記號。

- a) 登入您的 Salesforce 帳戶。
- b) 遵循說明文章 [Salesforce 說明中的步驟建立或重設安全記號](#)：重設安全記號。

3. 在 Salesforce 中建立自簽安全憑證。

- a) 從 **Force.com** 首頁的 **管理** 功能表中選取 **安全控制項**，然後選取 **憑證及金鑰管理**。即會開啟「憑證及金鑰管理」頁面。
- b) 按一下 **建立自簽憑證**。即會開啟「憑證」頁面。
- c) 在 **標籤** 欄位中輸入憑證的名稱，按 Tab，然後按一下 **儲存**。即會顯示「憑證及金鑰詳細資料」資訊。
- d) 按一下 **回到清單：憑證和金鑰**。
- e) 按一下 **匯出至金鑰儲存庫**。
- f) 輸入金鑰儲存庫的密碼，然後按一下 **匯出**。
- g) 將匯出的金鑰儲存庫儲存至本端檔案系統。

4. 使用 IBM 金鑰管理 GUI 來開啟您從 Salesforce 匯出的金鑰儲存庫，並移入簽章者憑證。

- a) 執行 `strmqikm` 指令，以開啟 IBM 金鑰管理 GUI。如需相關資訊，請參閱 [使用 runmqckm、runmqakm 及 strmqikm 來管理數位憑證](#)。
- b) 按一下 **開啟金鑰資料庫檔**，並瀏覽至 Salesforce 金鑰儲存庫的位置。
- c) 按一下 **開啟**，確定從 **金鑰資料庫類型** 選項中選取 **JKS**，然後按一下 **確定**。
- d) 輸入您在步驟 3f 中為金鑰儲存庫建立的密碼，然後按一下 **確定**。
- e) 從 **金鑰資料庫內容** 選項中選取 **簽章者憑證**。
- f) 按一下 **移入**。

- g) 從 **新增 CA 憑證** 清單中選取 **Verisign Inc.** 勾選框，然後按一下 **確定**。
5. 選擇性的: 在 Salesforce 帳戶中建立 IBM MQ Bridge to Salesforce 的應用程式連線，以產生 OAuth 消費者金鑰和密碼。
在正式作業環境中使用 IBM MQ Bridge to Salesforce 時，您需要 **消費者金鑰** 及 **消費者密碼** 代碼。
- a) 從 **Force.com 首頁** 的 **建置** 功能表中選取 **建立**，然後選取 **應用程式**。
即會開啟「**應用程式**」頁面。
 - b) 從 **已連接的應用程式** 區段中按一下 **新建**。
即會開啟「**新建連接的應用程式**」頁面。
 - c) 在 **已連接的應用程式名稱** 中輸入 IBM MQ Bridge to Salesforce 的名稱，例如 **MQBridgeToSalesforce**。
 - d) 輸入 **API 名稱**。
如果您按 Tab 鍵跳到下一個欄位，則會將 **已連接的應用程式名稱** 複製到 **API 名稱** 名稱欄位。
 - e) 輸入 **聯絡人電子郵件**。
 - f) 在 **API (啟用 OAuth 設定)** 區段中選取 **啟用 OAuth 設定** 選項。
然後會呈現該區段中的其他選項。
 - g) 新增 **回呼 URL**，例如 `https://www.ibm.com`。
 - h) 在 **選取的 OAuth 範圍** 子區段中，從 **可用的 OAuth 範圍** 清單中選取 **完整存取權 (完整)** 選項，然後按一下 **新增**，將完整存取權新增至 **選取的 OAuth 範圍** 清單。
 - i) 按一下 **儲存**。
 - j) 按一下 **繼續**。
 - k) 記下 **消費者金鑰** 及 **消費者密碼** 代碼。
6. 在佇列管理程式上建立必要的同步化及錯誤佇列。

```
cat /opt/mqm/mqsfb/samp/mqsfbSyncQ.mqsc | runmqsc PEQM1
```

同步化佇列會在應用程式或佇列管理程式重新啟動之間維護事件狀態。佇列深度可以較小，因為佇列上預期只有單一訊息。一次只能針對此佇列執行橋接器的一個實例，因此預設選項設定為專用存取。必須先建立錯誤佇列，然後才能使用橋接器來建立平台事件的事件訊息。錯誤佇列用於 Salesforce 無法順利處理的訊息。您必須在橋接器配置參數區段 **Connection to Queue Manager** 中新增錯誤佇列名稱，如步驟第 678 頁的『8.a』所示。

7. 選擇性的: 在 Salesforce 帳戶中建立平台事件物件。

- a) 從 **Force.com 首頁** 的 **開發** 功能表中選取 **平台事件**，然後按一下 **新建平台事件**。
即會開啟「**新建平台事件**」頁面。
- b) 完成 **標籤** 和 **多個標籤** 欄位。
- c) 按一下 **儲存**。
即會開啟「**平台事件定義明細**」頁面。
- d) 定義 **自訂欄位和關係**。
例如，您可以新增兩個具有標籤 *MyText* 和 *Name* 的文字欄位，並將 **資料類型** 欄位長度分別設為 *Text (64)* 和 *Text (32)*。

您已建立平台事件並為其定義 **Custom Fields and Relationships**。使用平台事件 平台物件名稱 或 **API 名稱** 作為 IBM MQ 主題，您可以將想要橋接器處理的訊息放置到該主題。例如，您可以使用 **AMQSPUBA** 範例，將下列 JSON 格式的訊息新增至 `/sf/mqtosfb/event/Salesforce Platform Object Name/API name` 主題：

```
{ "MyText__c" : "Some text here", "Name__c" : "Bob Smith" }
```

在橋接器啟動之後，您可以執行 **AMQSPUBA** 範例來建立訊息。從 `MQ installation location/samp/bin` 目錄，發出下列指令：

```
./amqspub /sf/mqtosfb/event/Salesforce Platform Object Name/API name PEQM1
```

在提示中，以 JSON 格式輸入訊息。

8. 使用 IBM MQ、Salesforce 和 IBM MQ Bridge to Salesforce 行為的連線和安全參數來建立配置檔。

```
runmqsfb -o new_config.cfg
```

現有的值會顯示在方括弧內。按 Enter 以接受現有值，按 Space then Enter 以清除值，然後鍵入 Enter 以新增值。

a) 輸入佇列管理程式 PEQM1: 的連線值:

連線所需的最小值為佇列管理程式名稱、IBM MQ 基本主題根、錯誤佇列名稱及通道名稱。

```
Connection to Queue Manager
-----
Queue Manager or JNDI CF      : []PEQM1
MQ Base Topic                 : []/sf
MQ Channel                    : []A channel you have defined or for example
SYSTEM.DEF.SVRCONN
MQ Conname                    : []
MQ Publication Error Queue   : [SYSTEM.SALESFORCE.ERRORQ]
MQ CCDT URL                   : []
JNDI implementation class    : [com.sun.jndi.fscontext.RefFSContextFactory]
JNDI provider URL            : []
MQ Userid                     : []
MQ Password                   : []
```

註: 如果您在本端連接，則不需要通道名稱。您不需要在配置檔中提供佇列管理程式名稱及基本主題，因為稍後當您執行橋接器時，可以將它們併入指令行。

b) 輸入 Salesforce 連線的值:

連線所需的最小值為 Salesforce 使用者 ID、密碼、安全記號及登入端點。在正式作業環境中，您可以為 OAuth 安全新增消費者金鑰和密碼。

```
Connection to Salesforce
-----
Salesforce Userid (reqd)     : []salesforce_login_email
Salesforce Password (reqd)  : []salesforce_login_password
Security Token (reqd)       : []Security_Token
Login Endpoint               : [https://login.salesforce.com]
Consumer ID                  : []
Consumer Secret Key         : []
```

c) 輸入 TLS 連線的憑證儲存庫值:

TLS 連線所需的最小值是 TLS 憑證及金鑰儲存庫密碼的金鑰儲存庫路徑。如果未提供信任儲存庫路徑或密碼，則金鑰儲存庫和密碼參數會用於信任儲存庫和密碼。如果您將 TLS 用於 IBM MQ 佇列管理程式連線，則可以使用相同的金鑰儲存庫。

```
Certificate stores for TLS connections
-----
Personal keystore for TLS certificates : []path_to_keystore, for example: /var/mqm/qmgrs/
PEQM1/ssl/key.jks
Keystore password                  : []keystore_password
Trusted store for signer certificates : []
Trusted store password             : []
Use TLS for MQ connection         : [N]
```

d) 輸入值以配置 IBM MQ Bridge to Salesforce 的行為:

您必須將 **Subscribe to MQ publications for platform events** 選項從預設 N 變更為 Y，以使用橋接器來建立事件訊息。您也必須在配置檔或指令行中指定日誌檔。

```
Behaviour of bridge program
-----
PushTopic Names                : []
Platform Event Names           : []
MQ Monitoring Frequency        : [30]
At-least-once delivery? (Y/N) : [Y]
Subscribe to MQ publications for platform events? (Y/N) : [Y]
Publish control data with the payload? (Y/N) : [N]
Delay before starting to process events : [0]
Runtime logfile for copy of stdout/stderr : []
```

9. 選擇性的: 建立 IBM MQ 服務以控制程式的執行。編輯範例 `mqsfbService.mqsc` 檔案, 以指向新建立的配置檔, 並對指令參數進行任何其他變更。

```
cat modified mqsfbService.mqsc | runmqsc PEQM1
```

10. 選擇性的: 遵循 [開始使用 IBM MQ 主控台](#) 中的指示, 以設定 IBM MQ Console。
11. 選擇性的: 在 IBM MQ Console 實例中新增及配置小組件, 以檢視 Salesforce 資料。
 - a) 按一下 **新增小組件**。
即會開啟新的小組件。
 - b) 選取 **圖表**
 - c) 按一下新小組件標題列中的 **配置小組件** 圖示。
 - d) 選擇性的: 輸入 **小組件標題**。
 - e) 從 **要監視的資源來源** 下拉功能表中選取 **Salesforce Bridge** 。
 - f) 從 **資源類別** 下拉功能表中, 選取 **橋接器狀態**。
 - g) 從 **資源類型** 下拉功能表中, 選取 **MQ 建立的平台事件**。
 - h) 從 **資源元素** 下拉功能表中, 選取 **總計 MQ 建立的平台事件**。
 - i) 按一下 **儲存**。

您已配置 IBM MQ Console, 以顯示 IBM MQ 建立的平台事件總數。當橋接器正在執行中, 且您開始在 `/sf/mqtosfb/event/Salesforce Platform Object Name/API name` 主題上放置訊息時, 小組件會顯示橋接器所建立的訊息事件總數。

V 9.1.0 IBM MQ Bridge to Salesforce 的訊息格式及錯誤訊息

IBM MQ Bridge to Salesforce 所處理訊息的格式化相關資訊。

應用程式會將訊息放入特定的佇列管理程式主題, 例如 `/root/mqtosfb/event/MQPlatformEvent1__e`。橋接器會訂閱主題, 從訊息取得內容, 並使用它來發佈 Salesforce 平台事件的事件訊息。

您需要在 Salesforce 中建立平台事件並定義內容欄位, 然後才能使用橋接器來建立該平台事件的事件訊息。平台事件名稱及其內容決定您需要如何格式化橋接器所處理的 IBM MQ 訊息。例如, 如果 Salesforce 平台事件 **Object name** 是 `MQPlatformEvent1`, 且您的兩個自訂定義欄位是具有 **API name** `MyText__c` 及 `Name__c` 的文字欄位, 則在 `/root/mqtosfb/event/MQPlatformEvent1__e` 主題上發佈的 IBM MQ 訊息必須是正確格式化的 JSON, 如下所示:

```
{ "MyText__c" : "Some text here", "Name__c" : "Bob Smith" }
```

橋接器所耗用及產生的訊息是 JSON 格式的文字 (MQSTR) 訊息。輸入訊息是簡式 JSON, 程式可以使用字串連結來產生它。

錯誤訊息

橋接器可以偵測到錯誤, 例如, 如果訊息不是文字格式, 或 Salesforce, 例如, 如果平台事件名稱不存在。如果在處理輸入訊息時發生錯誤, 則會將訊息連同說明錯誤的內容移至橋接器錯誤佇列。該錯誤也會寫入橋接器的 `stderr` 串流。

Salesforce 所產生的錯誤是 JSON。下列是錯誤格式化訊息所導致的一些錯誤:

不當的平台事件內容, 狀態 400 文字

```
{ "message": "No such column 'Name__c' on subject of type MQPlatformEvent2__e", "errorCode": "INVALID_FIELD" }
```

平台事件名稱無效, 狀態 404 文字

```
{ "errorCode": "NOT_FOUND", "message": "The requested resource does not exist" }
```

JSON 不正確，狀態 400 文字

```
{"errorCode":"NOT_FOUND","message":"The requested resource does not exist"}
```

訊息不是 JSON，狀態 400 文字

```
{  
  "message":  
    "Unexpected character ('h' (code 104)): expected a valid value (number, String, array,  
    object, 'true', 'false' or 'null') at [line:1, column:2]",  
  "errorCode":"JSON_PARSER_ERROR"}  
}
```

不是文字訊息 (未傳送至 Salesforce)

```
Error: Publication on topic ' /sf/mqtosfb/event/MQPlatformEvent1' does not contain a text  
formatted message
```

Linux V 9.1.0 執行 IBM MQ Bridge to Salesforce

執行 IBM MQ Bridge to Salesforce 以連接至 Salesforce 和 IBM MQ。連接時，橋接器可以建立 Salesforce 主題的訂閱，並將訊息重新發佈至 IBM MQ 主題。橋接器也可以建立 Salesforce 平台事件的事件訊息。

開始之前

您已完成 作業中的配置步驟:

- 第 669 頁的『配置 IBM MQ Bridge to Salesforce』
- 第 675 頁的『建立 Salesforce 平台事件的事件訊息』

關於這項作業

使用您在上一項作業中建立的配置檔，來執行 IBM MQ Bridge to Salesforce。如果您未在配置檔中併入所有必要的參數，請務必在指令行中併入它們。

程序

1. 在 Salesforce 中定義您要訂閱的推送主題或平台事件，或定義您要為其建立事件訊息的平台事件。
2. 啟動 IBM MQ Bridge to Salesforce，以連接至 Salesforce 及佇列管理程式。如果您正在執行橋接器以訂閱 Salesforce 事件，請包括您在步驟 1 中定義的推送主題或平台事件的名稱。

```
runmqsfb -f new_config.cfg -r logFile -p PushtopicName -e eventName
```

當橋接器已連接時，會傳回下列訊息:

- 如果您使用橋接器來訂閱 Salesforce 推送主題及平台事件:

```
Successful connection to queue manager QM1  
Warning: Subscribing to MQ-created platform events is not enabled.  
Successful login to Salesforce at https://eu11.salesforce.com  
Ready to process events.
```

- 如果您使用橋接器來建立 Salesforce 平台事件的事件訊息:

```
Successful connection to queue manager QM1  
Successful login to Salesforce at https://eu11.salesforce.com  
Successful subscription to '/sf/mqtosfb/event/' for MQ-created platform events  
Ready to process events.
```

3. 選擇性的: 如果在執行橋接器之後傳回的訊息指出連線不成功，請對佇列管理程式及 Salesforce 的連線進行疑難排解。
 - a) 以除錯模式發出指令，並搭配除錯選項 1。


```
runmqsfb -f new_config.cfg -r logFile -p PushtopicName -e eventName -d 1
```

橋接器會逐步執行連線設定，並以簡潔模式顯示處理訊息。

b) 在除錯模式下使用除錯選項 2 發出指令。

```
runmqsfb -f new_config.cfg -r logFile -p PushtopicName -e eventName -d 2
```

橋接器會逐步執行連線設定，並以詳細模式顯示處理訊息。完整輸出會寫入日誌檔。

4. 使用 Salesforce 介面來修改資料庫中的記錄，以產生事件。

5. 跳至 IBM MQ Console，以查看推送主題的變更出現在您在第一個作業中配置的小組件中。

下一步

例如，使用 `MQSFB_EXTRA_JAVA_OPTIONS` 變數來傳入 JVM 內容，以啟用 IBM MQ 追蹤。如需相關資訊，請參閱 [追蹤 IBM MQ Bridge to Salesforce](#)。

相關工作

[監視 IBM MQ Bridge to Salesforce](#)

相關參考

[runmqsfb \(執行 IBM MQ Bridge to Salesforce\)](#)

z/OS

MQ Adv.

Linux

V 9.1.0

配置 IBM MQ 以與區塊鏈搭配使用

設定並執行 IBM MQ Bridge to blockchain，以安全地連接 [Linux](#) IBM MQ Advanced 或

[z/OS](#) IBM MQ Advanced for z/OS Value Unit Edition 佇列管理程式與 IBM Blockchain。使用橋接器，透過使用連接至 IBM MQ Advanced 或 IBM MQ Advanced for z/OS VUE 佇列管理程式的傳訊應用程式，以非同步方式連接至、查閱及更新區塊鏈中資源的狀態。

開始之前

註：從 2022 年 11 月 22 日開始的所有版本都已淘汰 IBM MQ Bridge to blockchain (請參閱 [美國公告信 222-431](#))。



小心： [V 9.1.4](#) 不再支援以 Hyperledger Composer 為建置基礎的 IBM MQ Bridge to blockchain。

您必須執行 IBM MQ 9.1.4，才能使用 Hyperledger Fabric 上所建置的 IBM MQ Bridge to blockchain。

• IBM MQ Bridge to blockchain 可用於連接至：

- [Linux](#) IBM MQ Advanced，或
- [z/OS](#) IBM MQ Advanced for z/OS VUE

僅限佇列管理程式。

- [V 9.1.4](#) 佇列管理程式必須與橋接器或更高版本的指令層次相同；例如 9.1.4。
- [V 9.1.4](#) 支援 IBM MQ Bridge to blockchain 與以 Hyperledger Fabric 1.4 架構為基礎的區塊鏈網路搭配使用。

關於這項作業

Blockchain 是共用的分散式分帳，由區塊鏈組成，這些區塊鏈代表網路中對等節點之間的協議交易。鏈結中的每一個區塊會鏈結至前一個區塊，依此類推，回到第一個交易。

IBM Blockchain 以 Hyperledger Fabric 為建置基礎，您可以在本端使用 Docker 進行開發，或在 IBM Cloud (formerly Bluemix) 的容器叢集中進行開發。您也可以正式作業中啟動及使用 IBM Blockchain 網路，以建置及控管具有高階安全、隱私權及效能的商業網路。如需相關資訊，請參閱 [IBM Blockchain Platform](#)。

Hyperledger Fabric 是一個開放程式碼企業區塊鏈架構，由 Hyperledger Project 成員 (包括 IBM 作為起始程式碼貢獻者) 協同開發。Hyperledger Project，或 Hyperledger，是 Linux Foundation 開放程式碼、全球協同行動，以推動跨產業區塊鏈技術。如需相關資訊，請參閱 [IBM Blockchain](#)、[Hyperledger 專案](#) 及 [Hyperledger Fabric](#)。

如果您已使用 IBM MQ Advanced 或 IBM MQ Advanced for z/OS VUE 及 IBM Blockchain，則可以使用 IBM MQ Bridge to blockchain 來傳送簡式查詢、更新項目，以及從區塊鏈網路接收回覆。透過此方式，您可以整合內部部署 IBM 軟體與雲端區塊鏈服務。

您可以在 圖 1 中查看橋接器操作程序的簡要概觀。使用者應用程式會將 JSON 格式的訊息放置在 IBM MQ Advanced 或 IBM MQ Advanced for z/OS VUE 佇列管理程式的輸入/要求佇列上。橋接器會連接至佇列管理程式、從輸入/要求佇列取得訊息、檢查 JSON 是否正確格式化，然後對區塊鏈發出查詢或更新。區塊鏈傳回的資料由橋接器剖析，並放置在回覆佇列上，如原始 IBM MQ 要求訊息中所定義。使用者應用程式可以連接至佇列管理程式，從回覆佇列取得回應訊息，並使用該資訊。

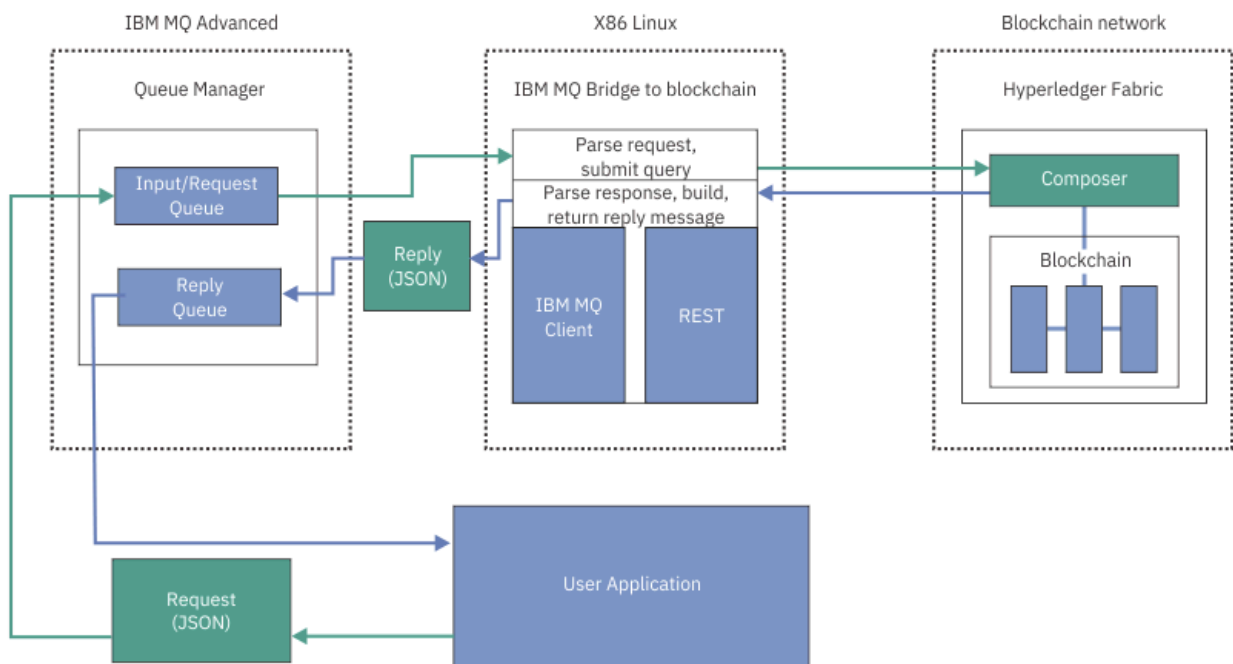


圖 98: IBM MQ Bridge to blockchain

您可以將 IBM MQ Bridge to blockchain 配置為以參與者或對等節點身分連接至區塊鏈網路。當橋接器執行時，傳訊應用程式會要求橋接器驅動鏈碼常式，以查詢或更新資源狀態，並將結果作為回應傳回給傳訊應用程式。

程序

1. 建立並啟動佇列管理程式，或啟動您要與 IBM MQ Bridge to blockchain 搭配使用的現有佇列管理程式。
建立佇列管理程式：

```
crtmqm adv_qmgr_name
```

啟動佇列管理程式：

```
strmqm adv_qmgr_name
```

2. 為 **DefineQ.mqsc** Script 中定義的橋接器建立佇列。
針對用於下列項目的預設具名佇列提供範例橋接器佇列定義：

- 使用者認證，例如 SYSTEM.BLOCKCHAIN.IDENTITY.QUEUE
- 橋接器的訊息輸入，例如 APPL1.BLOCKCHAIN.INPUT.QUEUE
- 來自區塊鏈的回覆，例如 APPL1.BLOCKCHAIN.REPLY.QUEUE

從 /opt/mqm/mqbc/samp 目錄，發出下列指令：

```
runmqsc adv_qmgr_name < ./DefineQ.mqsc
```

不同的應用程式可以使用相同的輸入佇列，但您可以指定多個回覆佇列，每一個應用程式一個回覆佇列。您不需要使用已定義的回覆佇列。如果您要對回覆使用動態佇列，則必須考量其安全配置。

結果

您已建立橋接器處理來自 IBM MQ 及區塊鏈網路的訊息所需的佇列。

下一步

使用 IBM MQ Advanced 或 IBM MQ Advanced for z/OS VUE 佇列管理程式資訊及來自區塊鏈網路的認證，來建立 IBM MQ Bridge to blockchain 的配置檔。

V9.1.0 建立 IBM MQ Bridge to blockchain 的配置檔

輸入佇列管理程式及區塊鏈網路參數，以建立 IBM MQ Bridge to blockchain 的配置檔，以連接至 IBM MQ 及 IBM Blockchain 網路。

開始之前

- 您已建立並配置區塊鏈網路。
- 您有來自區塊鏈網路的認證檔。
- 您已在 x86 Linux 環境上安裝 IBM MQ Bridge to blockchain。
- 您已啟動 IBM MQ Advanced 佇列管理程式。

關於這項作業

此作業會引導您完成建立 IBM MQ Bridge to blockchain 配置檔及順利連接至 IBM Blockchain 和 IBM MQ 網路所需的最小設定。

您可以使用橋接器來連接至以 Hyperledger Fabric 1.4 architecture 為基礎的區塊鏈網路。若要使用橋接器，您需要來自區塊鏈網路的配置資訊。在此作業的每一個步驟中，您可以找到基於兩個不同配置區塊鏈網路的配置詳細資料範例：

- 在 Docker 中執行的 Hyperledger Fabric 網路。如需相關資訊，請參閱 [開始使用 Hyperledger Fabric](#)、[撰寫第一個應用程式](#) 及 [第 685 頁的『Hyperledger Fabric 網路認證檔範例』](#)。
- 在 IBM Cloud (formerly Bluemix) 的 Kubernetes 叢集中執行的 Hyperledger Fabric 網路。如需相關資訊，請參閱 [在 IBM Blockchain Platform 上開發雲端沙盤推演](#)。

如需所有 IBM MQ Bridge to blockchain 參數的意義及選項的相關資訊，請參閱 [runmqbcb](#) 指令。您必須考量自己的安全需求，並自訂適合您部署的參數。

程序

1. 執行橋接器以建立配置檔。

您需要來自區塊鏈網路認證檔及 IBM MQ Advanced 佇列管理程式的參數。

```
runmqbcb -o config_file_name.cfg
```

如下列範例所示，現有的值會顯示在方括弧內。按 Enter 以接受現有值，按 Space then Enter 以清除值，並在方括弧內鍵入，然後按 Enter 以新增值。您可以用逗點來區隔值清單 (例如同層級)，或在新行上輸入每一個值。空白行結束清單。

註: 您無法編輯現有值。您可以保留、取代或清除它們。

2. 輸入 IBM MQ Advanced 佇列管理程式的連線值。

連線所需的最小值是佇列管理程式名稱、橋接器輸入的名稱，以及您定義的身分佇列。對於遠端佇列管理程式的連線，您也需要 **MQ Channel** 及 **MQ Conname** (佇列管理程式執行所在的主機位址及埠)。若要在步驟 第 684 頁的『4』中使用 TLS 來連接至 IBM MQ，您必須使用 JNDI 或 CCDT，並相應地指定 **MQ CCDT URL** 或 **JNDI implementation class** 及 **JNDI provider URL**。

```
Connection to Queue Manager
-----
Queue Manager                : [adv_qmgr_name]
Bridge Input Queue           : [APP[1].BLOCKCHAIN.INPUT.QUEUE]
Bridge User Identity Queue   : [SYSTEM.BLOCKCHAIN.IDENTITY.QUEUE]
MQ Channel                   : []
MQ Conname                   : []
MQ CCDT URL                  : []
JNDI implementation class    :
    [com.sun.jndi.fscontext.RefFSContextFactory]
JNDI provider URL           : []
MQ Userid                    : []
MQ Password                  : []
```

3. 13. 輸入網路的 Hyperledger Fabric 伺服器認證。

下列程式碼顯示您應該預期的內容範例:

```
Fabric Server
-----
Network configuration file   : []connection-tls.json
Wallet                       : []
User Name                    : []User1
Certificate                  : []<path_to_user_certificate>
Private Key                  : []<path_to_private_key>/private_key.pem
Organisation                 : []Org1MSP
```

4. 輸入 TLS 連線的憑證儲存庫值。

如果您沒有任何區域，請將此區域保留空白。

```
Certificate stores for MQ TLS connections
-----
Personal keystore            : []
Keystore password            : []
Trusted store for signer certs : []
Trusted store password       : []
```

5. 輸入要寫入橋接器日誌的日誌檔路徑。

```
Behaviour of bridge program
-----
Runtime logfile for copy of stdout/stderr : []bridgelog.log
Number of logfiles                       : [3]
Maximum size of each logfile (bytes)     : [2097152]
```



小心: 先前，有關對等節點、排序節點及憑證管理中心的詳細資料儲存在此橋接器設定中。不過，此資訊現在儲存在 網路配置檔中，該檔案在設定的 Hyperledger Fabric 伺服器區段中鏈結。

結果

您已建立配置檔，供 IBM MQ Bridge to blockchain 用來連接至 IBM Blockchain 網路及 IBM MQ Advanced 佇列管理程式。


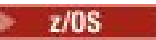
下一步

完成 第 687 頁的『[執行 IBM MQ Bridge to blockchain](#)』的步驟。

V 9.1.4 Hyperledger Fabric 網路認證檔範例

在 Docker 中執行之本端實例化 Hyperledger Fabric 區塊鏈網路中的 .yaml 檔案內容，可用來配置 IBM MQ Bridge to blockchain。

IBM MQ Bridge to blockchain 可用於連接至：

-  IBM MQ Advanced, 或
-  IBM MQ Advanced for z/OS VUE

僅限佇列管理程式。

在您完成 [開始使用 Hyperledger Fabric 指導教學](#)、[瞭解幕後發生的情況](#) 並使用其中一個 [Hyperledger Fabric 範例啟動網路](#) 之後，您應該在 `/blockchain/fabric-samples/basic-network` 資料夾中具有下列配置檔。

如果您想要連接至區塊鏈網路，當您是 [第 683 頁的『建立 IBM MQ Bridge to blockchain 的配置檔』](#) 時，必須使用此檔案中的配置詳細資料。

```
{
  "name": "basic-network",
  "version": "1.0.0",
  "client": {
    "organization": "Org1",
    "connection": {
      "timeout": {
        "peer": {
          "endorser": "300"
        },
        "orderer": "300"
      }
    }
  },
  "channels": {
    "mychannel": {
      "orderers": [
        "orderer.example.com"
      ],
      "peers": {
        "peer0.org1.example.com": {
          "endorsingPeer": true,
          "chaincodeQuery": true,
          "ledgerQuery": true,
          "eventSource": true
        },
        "peer0.org2.example.com": {
          "endorsingPeer": true,
          "chaincodeQuery": false,
          "ledgerQuery": true,
          "eventSource": false
        }
      }
    }
  },
  "organizations": {
    "Org1": {
      "mspid": "Org1MSP",
      "peers": [
        "peer0.org1.example.com"
      ],
      "certificateAuthorities": [
        "ca-org1"
      ],
      "adminPrivateKeyPEM": {
        "path": "$<path_to_private_key>/admin_private_key"
      },
      "signedCertPEM": {
        "path": "<path_to_org_signed_cert>/Admin@org1.example.com-cert.pem"
      }
    },
    "Org2": {
      "mspid": "Org2MSP",
      "peers": [
        "peer0.org2.example.com"
      ]
    }
  }
}
```

```

    "certificateAuthorities": [
      "ca-org2"
    ]
  },
  "orderers": {
    "orderer.example.com": {
      "url": "grpc://localhost:7050",
      "mspid": "OrdererMSP",
      "grpcOptions": {
        "ssl-target-name-override": "orderer.example.com",
        "hostnameOverride": "orderer.example.com"
      },
      "tlsCACerts": {
        "path": "<path_to_orderer_cert>/ca.crt"
      },
      "adminPrivateKeyPEM": {
        "path": "<path_to_orderers_private_key>/<private_key>"
      },
      "signedCertPEM": {
        "path": "<path_to_orderer_signed_cert>/Admin@example.com-cert.pem"
      }
    }
  },
  "peers": {
    "peer0.org1.example.com": {
      "url": "grpc://localhost:7051",
      "grpcOptions": {
        "ssl-target-name-override": "peer0.org1.example.com",
        "hostnameOverride": "peer0.org1.example.com",
        "request-timeout": 120001
      },
      "tlsCACerts": {
        "path": "<path_to_peer_cert>/ca.crt"
      }
    },
    "peer0.org2.example.com": {
      "url": "grpc://localhost:9051",
      "grpcOptions": {
        "ssl-target-name-override": "peer0.org2.example.com",
        "hostnameOverride": "peer0.org2.example.com",
        "request-timeout": 120001
      },
      "tlsCACerts": {
        "path": "<path_to_peer_cert>/ca.crt"
      }
    }
  },
  "certificateAuthorities": {
    "ca-org1": {
      "url": "https://localhost:7054",
      "grpcOptions": {
        "verify": true
      },
      "tlsCACerts": {
        "path": "<path_to_ca_cert>/ca.org1.example.com-cert.pem"
      },
      "registrar": [
        {
          "enrollId": "admin",
          "enrollSecret": "adminpw"
        }
      ]
    },
    "ca-org2": {
      "url": "https://localhost:8054",
      "grpcOptions": {
        "verify": true
      },
      "tlsCACerts": {
        "path": "<path_to_ca_cert>/ca.org2.example.com-cert.pem"
      },
      "registrar": [
        {
          "enrollId": "admin",
          "enrollSecret": "adminpw"
        }
      ]
    }
  }
}

```

V 9.1.0 執行 IBM MQ Bridge to blockchain

執行 IBM MQ Bridge to blockchain 以連接至 IBM Blockchain 和 IBM MQ。連接時，橋接器已準備好處理要求訊息，將它們傳送至 Hyperledger Composer 區塊鏈網路，並接收及處理回覆。

關於這項作業

使用您在上一項作業中建立的配置檔，來執行 IBM MQ Bridge to blockchain。

程序

1. 啟動您要與橋接器搭配使用的 IBM MQ Advanced 佇列管理程式。
2. 啟動 IBM MQ Bridge to blockchain，以連接至 Hyperledger Composer REST 伺服器及 IBM MQ Advanced 佇列管理程式。

執行橋接器指令。

```
runmqbc -f /config_file_location/config_file_name.cfg -r /log_file_location/logFile.log
```

當橋接器已連接時，會傳回類似下列的輸出：

```
2018-05-17 14:28:16.866 BST IBM MQ Bridge to Blockchain
5724-H72 (C) Copyright IBM Corp. 2017, 2024.

2018-05-17 14:28:19.331 BST Ready to process input messages.
```

3. 選擇性的：如果在執行橋接器之後傳回的訊息指出連線不成功，則對 IBM MQ Advanced 佇列管理程式及區塊鏈網路的連線進行疑難排解。

- a) 以除錯模式發出指令，並搭配除錯選項 1。

```
runmqbc -f /config_file_location/config_file_name.cfg -r /log_file_location/logFile.log
-d 1
```

橋接器會逐步執行連線設定，並以簡潔模式顯示處理訊息。

- b) 在除錯模式下使用除錯選項 2 發出指令。

```
runmqbc -f /config_file_location/config_file_name.cfg -r /log_file_location/logFile.log
-d 2
```

橋接器會逐步執行連線設定，並以詳細模式顯示處理訊息。完整輸出會寫入日誌檔。

結果

您已啟動 IBM MQ Bridge to blockchain，並使用 Hyperledger Composer REST 伺服器連接至佇列管理程式及區塊鏈網路。

下一步

- 遵循第 690 頁的『執行 IBM MQ Bridge to blockchain 用戶端範例』中的步驟，將查詢或更新訊息格式化並傳送至區塊鏈網路。
- 使用 `MQBCB_EXTRA_JAVA_OPTIONS` 變數來傳入 JVM 內容，例如啟用 IBM MQ 追蹤。如需相關資訊，請參閱 [追蹤 IBM MQ Bridge to blockchain](#)。

z/OS V 9.1.0 IBM MQ 9.1.4 之前 IBM MQ Bridge to blockchain 的訊息格式

IBM MQ Bridge to blockchain 所傳送及接收之訊息的格式化相關資訊。

LTS



小心: 訊息格式的現有格式已作廢。從 IBM MQ 9.1.4, 如果您有 Hyperledger Fabric 網路, 請使用第 689 頁的『IBM MQ 9.1.4 中 IBM MQ Bridge to blockchain 的訊息格式』中說明的訊息格式。

應用程式要求 IBM MQ Bridge to blockchain 驅動 Hyperledger Composer 定義的 REST API, 以處理區塊鏈上保留的資訊。應用程式透過在橋接器要求佇列上放置要求訊息來執行此動作。REST 要求的結果由橋接器格式化為回覆訊息。橋接器會使用要求訊息 MQMD 的 **ReplyToQ** 及 **ReplyToQMGR** 欄位中包含的資訊, 作為回覆訊息的目的地。

要求及回覆訊息是 JSON 格式的文字 (MQSTR) 訊息。

要求訊息格式

要求訊息包含三個屬性:

方法 (method)

用來呼叫 Hyperledger Composer REST API 的 REST 動詞, 例如 POST、DELETE 或 GET

路徑

Hyperledger Composer REST API 的路徑。這會新增至基本伺服器 URL。路徑應該以 "api/" 開頭。

body

方法特定內容。這通常是 JSON 結構。

下列範例使用 POST 方法, 以路徑 `api/Trader` 來建立新的 Trader 物件。內文會指定使用者的 Hyperledger Composer 模型所定義的「交易商」類別, 也會指定在區塊鏈網路內建立新 Trader 物件所需的其他值。

```
{
  "method": "POST",
  "path": "api/Trader",
  "body": {
    "$class": "org.example.trading",
    "tradeId": "Trader2",
    "firstName": "Jane",
    "lastName": "Doe"
  }
}
```

回覆訊息格式

回覆訊息會將其相關性 ID 設為入埠訊息的訊息 ID。任何使用者定義內容都會從要求訊息複製到回覆訊息。回覆中的使用者 ID 會設為發送端的使用者 ID。

statusCode 是 HTTP 狀態碼。如果錯誤來自 IBM MQ 或橋接器, 則會使用適當的 **statusCode**。

statusType 是字串, 可以是 *SUCCESS* 或 *FAILURE*。

對於成功要求, 回覆訊息中的 **"data"** 元素包含來自所呼叫 Hyperledger Composer REST API 的回應。

成功處理的範例:

```
{
  "statusCode": 200,
  "statusType": "SUCCESS",
  "data": [
    {
      "$class": "org.example.trading",
      "firstName": "John",
      "lastName": "Doe",
      "tradeId": "Trader1"
    },
    {
      "$class": "org.example.trading",
      "firstName": "Jane",
      "lastName": "Doe",
      "tradeId": "Trader2"
    }
  ]
}
```

所有錯誤回應都具有相同的欄位, 不論它們是否由橋接器本身所產生, 都是從呼叫 Hyperledger Composer REST 伺服器、區塊鏈或從鏈碼呼叫產生。例如:

- JSON 輸入訊息不正確

```
{
  "statusCode": 400,
  "statusType": "FAILURE",
  "message": "[AMQBC021E] Error: Cannot parse input message or there are
  missing fields in the message. Missing fields appear to be: "method"."
}
```

- Hyperledger Composer REST 伺服器無法處理的要求

```
{
  "statusCode": 500,
  "statusType": "FAILURE",
  "message": "Error trying to invoke business network. Error: No valid responses
  from any peers.\nResponse from attempted peer comms was an error: Error: chaincode
  error (status: 500, message: Error: Failed to add object with ID 'Trader1'
  as the object already exists)"
}
```

應用程式可以透過查看 **statusType** 字串，或從資料欄位的存在來判斷要求是成功還是失敗。當處理輸入訊息時發生錯誤，且橋接器未將它傳送至區塊鏈時，從橋接器傳回的值是 MQRC 值，通常是 **MQRC_FORMAT_ERROR**。

V 9.1.4 Linux IBM MQ 9.1.4 中 IBM MQ Bridge to blockchain 的訊息格式

IBM MQ Bridge to blockchain 所傳送及接收之訊息的格式化相關資訊。

應用程式要求 IBM MQ Bridge to blockchain 驅動 Hyperledger Fabric 伺服器處理區塊鏈上保留的資訊。應用程式透過在橋接器要求佇列上放置要求訊息來執行此動作。橋接器會將要求的結果格式化為回覆訊息。橋接器會使用要求訊息 MQMD 的 **ReplyToQ** 及 **ReplyToQMGR** 欄位中包含的資訊，作為回覆訊息的目的地。

要求及回覆訊息是 JSON 格式的文字 (MQSTR) 訊息，且包含四個元素。

要求訊息格式

要求訊息包含下列屬性：

operation

字串-不區分大小寫

submit 代表更新項目或 evaluate 代表查詢

網路

字串-有時在 Hyperledger Fabric 中稱為 channel

合約

字串-要呼叫的智慧型合約或鏈碼套件

args

陣列-通常是字串，但部分元素可以是巢狀 JSON 物件。

contract 的實際引數，包括方法名稱。

例如：

```
{
  "operation" : "Evaluate",
  "network"   : "mychannel",
  "contract"  : "marbles0",
  "args"     : [ "readMarble" , "marble1" ]
}
```

註：除了確保這些元素存在，且訊息是有效的 JSON 之外，橋接器不會對內容進行驗證。橋接器依賴 Hyperledger Fabric 來處理要求或傳回錯誤。

回覆訊息格式

回覆訊息會將其相關性 ID 設為入埠訊息的訊息 ID。任何使用者定義內容都會從要求訊息複製到回覆訊息。回覆中的使用者 ID 會設為發送端的使用者 ID。

statusCode 是 HTTP 狀態碼。如果錯誤來自 IBM MQ 或橋接器，則會使用適當的 **statusCode**。

statusType 是字串，可以是 *SUCCESS* 或 *FAILURE*。

對於成功要求，回覆訊息中的 **"data"** 元素包含來自所呼叫 Hyperledger Composer REST API 的回應。

成功處理的範例：

```
{
  "statusCode": 200,
  "statusType": "SUCCESS",
  "data": [
    {
      "$class": "org.example.trading",
      "firstName": "John",
      "lastName": "Doe",
      "tradeId": "Trader1"
    },
    {
      "$class": "org.example.trading",
      "firstName": "Jane",
      "lastName": "Doe",
      "tradeId": "Trader2"
    }
  ]
}
```

所有錯誤回應都具有相同的欄位，不論它們是否由橋接器本身所產生，都是從呼叫 Hyperledger Composer REST 伺服器、區塊鏈或從鏈碼呼叫產生。例如：

- JSON 輸入訊息不正確

```
{
  "statusCode": 400,
  "statusType": "FAILURE",
  "message": "[AMQBC021E] Error: Cannot parse input message or there are missing fields in the message. Missing fields appear to be: \"method\"."
}
```

- Hyperledger Composer REST 伺服器無法處理的要求

```
{
  "statusCode": 500,
  "statusType": "FAILURE",
  "message": "Error trying to invoke business network. Error: No valid responses from any peers.\nResponse from attempted peer comms was an error: Error: chaincode error (status: 500, message: Error: Failed to add object with ID 'Trader1' as the object already exists)"
}
```

應用程式可以透過查看 **statusType** 字串，或從資料欄位的存在來判斷要求是成功還是失敗。當處理輸入訊息時發生錯誤，且橋接器未將它傳送至區塊鏈時，從橋接器傳回的值是 MQRC 值，通常是 **MQRC_FORMAT_ERROR**。

z/OS 執行 IBM MQ Bridge to blockchain 用戶端範例

您可以使用 IBM MQ Bridge to blockchain 隨附的 JMS 用戶端範例，將訊息放置在區塊鏈橋接器正在檢查的輸入佇列中，並查看收到的回覆。此範例基於使用 IBM MQ Bridge to blockchain 與 Hyperledger Composer Trader 網路範例整合。

開始之前



如需相關資訊，請參閱 [/trade_network](#)

您的 IBM MQ Bridge to blockchain 正在執行中，且已連接至 IBM MQ Advanced 或 IBM MQ Advanced for z/OS VUE 佇列管理程式及區塊鏈網路。

關於這項作業

在 IBM MQ Bridge to blockchain 的 samp 目錄中尋找 JMS 範例應用程式 (ComposerBCBSamp.java)。

例如: <MQ_INSTALL_ROOT>/mqbc/samp/ComposerBCBSamp.java，其中 <MQ_INSTALL_ROOT> 是:

-  IBM MQ 的安裝目錄
-  IBM MQ 的 USS 元件安裝所在的 USS 目錄

程序

1. 編輯用戶端範例 Java 原始檔。

遵循範例中的指示來配置它，以符合您的 IBM MQ 環境及區塊鏈網路。

範例中的下列程式碼定義三個要傳送至橋接器的 JSON 要求訊息:

- 首先，移除現有的 'commodity'
- 其次，若要建立新的 'commodity'、'owner' 及相關聯的值，
- 最後，在前兩則要求訊息之後，顯示 'commodity' 的新相關資訊

```
private static JSONObject[] createMessageBodies() {
    JSONObject[] msgs = new JSONObject[3]; // This method creates 3 messages
    JSONObject m, m2;
    String commodityName = "BC";

    // Clean out the commodity in case it's already there. If
    // it's not there, there will be an error returned from Composer.
    m = new JSONObject();
    m.put("method", "DELETE");
    m.put("path", "api/Commodity/" + commodityName);
    msgs[0] = m;

    // To add the item to the table, the
    // operation looks like this:
    //
    // { "method": "POST",
    //   "path": "api/Commodity",
    //   "body" : {
    //     "$class": "org.example.trading.Commodity",
    //     "tradingSymbol" : "BC",
    //     "description" : "BC",
    //     "mainExchange" : "HERE",
    //     "owner" : "Me",
    //     "quantity" : 100
    //   }
    // }
    // You can see this structure in the API Explorer
    m = new JSONObject();
    m.put("method", "POST");
    m.put("path", "api/Commodity");
    m2 = new JSONObject();
    m2.put("$class", " org.example.trading.Commodity");
    m2.put("tradingSymbol", commodityName);
    m2.put("description", "Blockchain Sample Description");
    m2.put("mainExchange", "My Exchange");
    m2.put("owner", "Me");
    m2.put("quantity", 100);
    m.put("body", m2);
    msgs[1] = m;

    // And list all items that have been created
```

```

    m = new JSONObject();
    m.put("method", "GET");
    m.put("path", "api/Commodity");
    msgs[2] = m;

    return msgs;
}

```

2. 編譯範例。

指向橋接器目錄中隨附的 IBM MQ 用戶端類別及 JSON4J.jar 檔案。

```

javac -cp <MQ_INSTALL_ROOT>/java/lib/*:<MQ_INSTALL_ROOT>/mqbc/prereqs/JSON4J.jar
ComposerBCClient.java

```

3. 執行已編譯的類別。

```

java -cp <MQ_INSTALL_ROOT>/java/lib/*:<MQ_INSTALL_ROOT>/mqbc/prereqs/JSON4J.jar .
ComposerBCClient

```

```

Starting Simple MQ Blockchain Bridge Client
Starting the connection.
Sent message:
{"method":"DELETE"," path ":"api/Commodity/BC"}
Response text:
{
  "statusCode": 204,
  "statusType": "SUCCESS",
  "message": "OK",
  "data": ""
}
SUCCESS
Sent message:
{"body":
{"$class":"org.example.trading.Commodity","owner":"Me","quantity":100,"description":"Blockcha
in Sample Description","mainExchange":"My
Exchange","tradingSymbol":"BC"},"operation":"POST","url":"Commodity"}
Response text:
{
  "statusCode": 200,
  "statusType": "SUCCESS",
  "message": "OK",
  "data": {
    "$class": "org.example.trading.Commodity",
    "description": "Blockchain Sample Description",
    "mainExchange": "My Exchange",
    "owner": "Me",
    "quantity": 100,
    "tradingSymbol": "BC"
  }
}
SUCCESS
Sent message:
{"method":"GET","path":"api/Commodity"}
Response text:
{
  "statusCode": 200,
  "statusType": "SUCCESS",
  "message": "OK",
  "data": [
    {
      "$class": "org.example.trading.Commodity",
      "description": "Blockchain Sample Description",
      "mainExchange": "My Exchange",
      "owner": "resource:org.example.trading.Trader#Me",
      "quantity": 100,
      "tradingSymbol": "BC"
    }
  ]
}
SUCCESS

```

message 欄位包含已順利處理訊息的 "OK"，或如果要求失敗，則包含失敗原因的相關資訊。

如果用戶端在等待回應時收到逾時錯誤，請檢查橋接器是否在執行中。

IBM MQ 9.1.2 對 IBM MQ Bridge to blockchain 上的追蹤和記載工作方式引進了一項變更。

IBM MQ 9.1.0 IBM MQ Bridge to blockchain 中的變更

依預設，除了日誌檔現在開始輪換之外，IBM MQ 9.1.0 橋接器對行為沒有任何變更。如需相關資訊，請參閱 [第 693 頁的『循環日誌』](#)。

追蹤及除錯互動

除錯旗標會繼續像之前一樣運作。也就是說，`-d1` 會提供橋接器除錯資訊，而 `-d2` 會開啟必備元件的除錯記載。不過，如果您在啟動橋接器時已啟用 IBM MQ 追蹤，則會自動開啟 `-d2` 層次報告。

循環日誌

日誌檔的預設行為會變更為有三個日誌檔，每一個大小為 2 MB。您可以使用其他配置內容來置換這些值。日誌檔的現有配置屬性或指令行參數會作為日誌的基本名稱，並新增索引。

如果配置の日誌檔具有：

- 沒有檔案類型，索引會新增至檔名結尾。
將日誌檔設為 `abc` 會產生稱為 `abc.0`、`abc.1` 等的日誌。
- 檔案類型，索引會插入在檔案類型之前。
將日誌檔設為 `abc.log` 會產生稱為 `abc.0.log`、`abc.1.log` 等的日誌。

附註：

1. 由於橋接器可以使用任意使用者許可權來執行，因此無法強制日誌的特定目錄 (例如，`/var/mqm/qmgrs/<qm>/errors`)。
2. 相同的資訊會繼續寫入 `stdout` 和 `stderr` 串流。
3. 每當個別日誌檔重新開啟時，即會重新列印基本配置資訊。資訊將一律可用，而不是在程式開始時只列印一次。

在 z/OS 上配置佇列管理程式

使用這些指示，在「IBM MQ for z/OS」上配置佇列管理程式。

開始之前

在配置 IBM MQ 之前，請先閱讀 [IBM MQ for z/OS 概念中的 IBM MQ for z/OS 概念](#)。

在 [規劃 z/OS 上的 IBM MQ 環境](#) 中，閱讀如何規劃 IBM MQ for z/OS 環境。

關於這項作業

安裝 IBM MQ 之後，您必須先執行一些作業，才能將它提供給使用者使用。

程序

- 如需如何在「IBM MQ for z/OS」上配置佇列管理程式的相關資訊，請參閱下列子主題。

相關概念

[IBM MQ for z/OS 概念](#)

相關工作

[第 7 頁的『在 Multiplatforms 上建立佇列管理程式』](#)

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。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 MQ 物件。


保護安全

第 147 頁的『[配置分散式佇列](#)』

本節提供 IBM 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 13 頁的『[配置用戶端與伺服器之間的連線](#)』

若要配置 IBM MQ MQI clients 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。

 [管理 IBM MQ for z/OS](#)

規劃

 [發出指令](#)

相關參考

 [IBM MQ for z/OS 公用程式](#)

準備在 z/OS 上自訂佇列管理程式

當使用可安裝特性、國家語言特性的詳細資料，以及測試和設定安全的相關資訊來自訂佇列管理程式時，請使用這個主題。

準備自訂作業

「程式目錄」列出 IBM MQ 安裝磁帶的內容、IBM MQ 的程式及服務層次資訊，並說明如何使用「系統修改程式延伸 (SMP/E)」來安裝 IBM MQ for z/OS。IBM MQ for z/OS 的程式目錄可以從 [IBM 出版品中心](#) 下載 (請參閱 IBM MQ for z/OS 程式目錄 PDF 檔案)。

當您已安裝 IBM MQ 時，必須先執行一些作業，然後才能讓使用者使用它。如需這些作業的說明，請參閱下列各節：

- [第 697 頁的『設定 IBM MQ for z/OS』](#)
- [第 751 頁的『在 z/OS 上測試佇列管理程式』](#)
- [在 z/OS 上設定安全](#)

如果您是從舊版 IBM MQ for z/OS 移轉，則不需要執行大部分自訂作業。如需您必須執行之作業的相關資訊，請參閱 [維護及移轉](#)。

IBM MQ for z/OS 的可安裝特性

IBM MQ for z/OS 包含下列特性：

基本程式

這是必要的；它包含所有主要功能，包括

- 管理及公用程式
- 支援使用 IBM MQ 應用程式設計介面或 C++ 的 CICS、IMS 及批次類型應用程式
- 分散式佇列機能 (同時支援 TCP/IP 及 APPC 通訊)

國家語言特性

這些包含所有受支援國家語言的錯誤訊息及畫面。每一種語言都有一個相關聯的語言字母。語言及字母如下：

C

簡體中文

E

U.S. 英文 (大小寫混合格式)

F

法文

K

日文

U

U.S. 英文 (大寫)

您必須安裝美式英文 (大小寫混合格式) 選項。您也可以安裝一或多種其他語言。(其他語言的安裝程序需要安裝美式英文 (大小寫混合格式)，即使您不會使用美式英文 (大小寫混合格式)。)

IBM MQ for z/OS Unix 系統服務元件

此特性是選用的。如果您想要建置並執行使用 Java Message Service (JMS) 來連接至 IBM MQ for z/OS 的 Java 應用程式，或如果您想要建置並執行使用 HTTP 來連接至 IBM MQ for z/OS 的 HTTP 應用程式，請選取此特性。

V 9.1.0 IBM MQ for z/OS Unix 系統服務 Web 元件

此特性是選用的。

如果您想要使用 IBM MQ Console 或 REST API，請選取此特性。

您必須安裝 IBM MQ for z/OS Unix 系統服務元件特性，才能安裝此特性。

安裝之後存在的程式庫

IBM MQ 隨附數個個別載入程式庫。第 695 頁的表 53 顯示安裝 IBM MQ 之後可能存在的程式庫。

姓名	說明
thlqual.SCSQANLC	包含 IBM MQ 簡體中文版的載入模組。
thlqual.SCSQANLE	包含 U.S 的載入模組。IBM MQ 的英文 (大小寫混合格式) 版本。
thlqual.SCSQANLF	包含 IBM MQ 法文版本的載入模組。
thlqual.SCSQANLK	包含日文版 IBM MQ 的載入模組。
thlqual.SCSQANLU	包含 U.S 的載入模組。IBM MQ 的英文 (大寫) 版本。
thlqual.SCSQASMS	包含組譯器範例程式的原始檔。
thlqual.SCSQAUTH	所有 IBM MQ 產品載入模組的主要儲存庫; 它也包含預設參數模組 CSQZPARM。此檔案庫必須是 APF 授權及 PDS-E 格式。
thlqual.SCSQCICS	包含必須包含在 CICS DFHRPL 連結中的額外載入模組。此檔案庫必須是 APF 授權及 PDS-E 格式。
thlqual.SCSQCLST	包含範例程式所使用的 CLIST。
thlqual.SCSQCOBC	包含 COBOL 記錄定義檔，包括範例程式所需的記錄定義檔。
thlqual.SCSQCOBS	包含 COBOL 範例程式的原始檔。
thlqual.SCSQCPPS	包含 C++ 範例程式的原始檔。
thlqual.SCSQC37S	包含 C 範例程式的原始檔。
thlqual.SCSQC37O	包含 C 標頭，包括範例程式所需的標頭。
thlqual.SCSQDEFS	包含 C++ 及共用佇列作業之 Db2 DBRM 的端定義。
thlqual.SCSQEXEC	如果您使用 IBM MQ 作業和控制台，則包含要併入 SYSEXEC 或 SYSPROC 連結中的 REXX 執行檔。
thlqual.SCSQHPPS	包含 C++ 的標頭檔。
thlqual.SCSQINST	包含安裝工作的 JCL。

表 53: 安裝之後存在的 IBM MQ 程式庫 (繼續)	
姓名	說明
thlqual.SCSQLINK	早期程式碼庫。包含在系統起始程式載入 (IPL) 時載入的載入模組。磁帶庫必須經過 APF 授權。
thlqual.SCSQLOAD	載入程式庫。包含非 APF 程式碼、使用者結束程式、公用程式、範例、安裝驗證程式及配接器 Stub 的載入模組。程式庫不需要經過 APF 授權，也不需要在此清單中。此檔案庫必須為 PDS-E 格式。
thlqual.SCSQMACS	包含 Assembler 巨集，包括：範例巨集、產品巨集及系統參數巨集。
thlqual.SCSQMAPS	包含範例程式所使用的 CICS 對映集。
thlqual.SCSQMSGC	如果您對 IBM MQ 作業和控制台使用簡體中文語言特性，則包含要併入 ISPMLIB 連結中的 ISPF 訊息。
thlqual.SCSQMSGE	如果您使用 U.S，則包含要併入 ISPMLIB 連結中的 ISPF 訊息。IBM MQ 作業及控制面板的英文 (大小寫混合格式) 語言特性。
thlqual.SCSQMSGF	如果您對 IBM MQ 作業和控制台使用法文特性，則包含要併入 ISPMLIB 連結中的 ISPF 訊息。
thlqual.SCSQMSGK	如果您將日文語言特性用於 IBM MQ 作業和控制台，則包含要併入 ISPMLIB 連結中的 ISPF 訊息。
thlqual.SCSQMSGU	如果您使用 U.S，則包含要併入 ISPMLIB 連結中的 ISPF 訊息。IBM MQ 作業及控制面板的英文 (大寫) 語言特性。
thlqual.SCSQMVR1	包含分散式佇列作業的載入模組。此檔案庫必須是 APF 授權及 PDS-E 格式。
thlqual.SCSQPLIC	包含 PL/I 併入檔。
thlqual.SCSQPLIS	包含 PL/I 範例程式的原始檔。
thlqual.SCSQPDLA	包含要併入 ISPLIB 連結中之傾出格式製作程式的 IPCS 畫面。也包含 IBM MQ 範例程式的畫面。
thlqual.SCSQPDLN	如果您將簡體中文語言特性用於 IBM MQ 作業和控制面板，則包含要包含在 ISPLIB 連結中的 ISPF 畫面。
thlqual.SCSQPDLN	如果您使用 U.S，則包含要包含在 ISPLIB 連結中的 ISPF 畫面。IBM MQ 作業及控制面板的英文 (大小寫混合格式) 語言特性。
thlqual.SCSQPDLF	如果您對 IBM MQ 作業和控制台使用法文特性，則包含要併入 ISPLIB 連結中的 ISPF 畫面。
thlqual.SCSQPDLK	如果您對 IBM MQ 作業和控制台使用日文語言特性，則包含要併入 ISPLIB 連結中的 ISPF 畫面。
thlqual.SCSQPDLU	如果您使用 U.S，則包含要包含在 ISPLIB 連結中的 ISPF 畫面。IBM MQ 作業及控制面板的英文 (大寫) 語言特性。
thlqual.SCSQPROC	包含範例 JCL 及預設系統起始設定資料集。
thlqual.SCSQSNLC	包含特殊用途函數 (例如早期程式碼) 所需之 IBM MQ 模組簡體中文版本的載入模組。
thlqual.SCSQSNLE	包含 U.S 的載入模組。特殊用途函數 (例如早期程式碼) 所需之 IBM MQ 模組的英文 (大小寫混合格式) 版本。
thlqual.SCSQSNLF	包含特殊用途函數 (例如早期程式碼) 所需之 IBM MQ 模組法文版本的載入模組。

表 53: 安裝之後存在的 IBM MQ 程式庫 (繼續)	
姓名	說明
thlqual.SCSQSNLK	包含特殊用途函數 (例如早期程式碼) 所需之 IBM MQ 模組日文版的載入模組。
thlqual.SCSQSNLU	包含 U.S 的載入模組。特殊用途函數 (例如早期程式碼) 所需的 IBM MQ 模組英文 (大寫) 版本。
thlqual.SCSQTBLC	如果您對 IBM MQ 作業和控制台使用簡體中文語言特性, 則包含要包含在 ISPTLIB 連結中的 ISPF 表格。
thlqual.SCSQTBLE	如果您使用 U.S, 則包含要併入 ISPTLIB 連結中的 ISPF 表格。IBM MQ 作業及控制面板的英文 (大小寫混合格式) 語言特性。
thlqual.SCSQTBLE	如果您對 IBM MQ 作業和控制台使用法文特性, 則包含要併入 ISPTLIB 連結中的 ISPF 表格。
thlqual.SCSQTBLE	如果您將日文語言特性用於 IBM MQ 作業和控制台, 則包含要併入 ISPTLIB 連結中的 ISPF 表格。
thlqual.SCSQTBLE	如果您使用 U.S, 則包含要併入 ISPTLIB 連結中的 ISPF 表格。IBM MQ 作業及控制面板的英文 (大寫) 語言特性。

註: 請勿修改或自訂任何這些程式庫。如果您要進行變更, 請複製檔案庫並對副本進行變更。

相關概念

IBM MQ for z/OS 概念

第 784 頁的『將 IBM MQ 與 IMS 搭配使用』

IBM MQ -IMS 配接器和 IBM MQ -IMS 橋接器是兩個元件, 可讓 IBM MQ 與 IMS 互動。

第 791 頁的『將 IBM MQ 與 CICS 搭配使用』

如果要將 IBM MQ 與 CICS 搭配使用, 您必須配置 IBM MQ CICS 配接器, 並選擇性地配置 IBM MQ CICS bridge 元件。

第 793 頁的『在 IMS 中使用 OTMA 結束程式』

如果您想要使用 IMS Open Transaction Manager Access exits with IBM MQ for z/OS, 請使用這個主題。

相關工作

第 759 頁的『在 z/OS 上設定與其他佇列管理程式的通訊』

本節說明您在開始使用分散式佇列之前需要進行的 IBM MQ for z/OS 準備。

管理 IBM MQ for z/OS

相關參考

第 791 頁的『升級及套用服務至 Language Environment 或 z/OS Callable Services』

您必須採取的動作會因您使用 CALLLIBS 或 LINK, 以及您的 SMP/E 版本而有所不同。

z/OS 設定 IBM MQ for z/OS

請使用本主題作為逐步手冊, 以自訂 IBM MQ for z/OS 系統。

配置佇列管理程式的最佳方式是依所示順序執行下列步驟:

1. 配置基本佇列管理程式。
2. 配置通道起始程式, 它會執行佇列管理程式至佇列管理程式的通訊, 以及遠端用戶端應用程式通訊。
3. 如果您要加密或保護訊息, 請配置 Advanced Message Security for z/OS。
4. 如果您要使用 IBM MQ 來傳送檔案, 請配置 Managed File Transfer for z/OS。
5. 如果您要使用管理或傳訊 REST API 或 MQ Console 從 Web 瀏覽器管理 IBM MQ, 請配置 mqweb 伺服器。

在您順利安裝 IBM MQ 之後, 本主題會引導您完成設定的各種階段。安裝程序在「程式目錄」中說明。IBM MQ for z/OS 的程式目錄可以從 [IBM 出版品中心](#) 下載 (請參閱 [IBM MQ for z/OS 程式目錄 PDF 檔案](#))。

IBM MQ 會提供範例來協助您進行自訂作業。範例資料集成員的名稱以四個字元 CSQ4 開頭，且位於程式庫 thlqual.SCSQPROC 中。

在執行本主題中說明的自訂作業之前，您必須考量一些配置選項，因為它們會影響 IBM MQ for z/OS 的效能和資源需求。例如，您必須決定要使用哪些全球化程式庫。

如果您想要自動化部分自訂作業步驟，請參閱 [第 797 頁的『使用 IBM z/OSMF 來自動化 IBM MQ』](#)。

配置選項

如需這些選項的相關資訊，請參閱在 [z/OS 上規劃](#)。

本節中每一項作業的說明指出：

- 此作業是設定 IBM MQ 程序的一部分。也就是說，當您在 z/OS 系統上自訂 IBM MQ 時，您會執行一次作業。(在平行 Sysplex 中，您必須對 Sysplex 中的每一個 z/OS 系統執行作業，並確保每一個 z/OS 系統的設定相同。)
- 此作業是新增佇列管理程式的一部分。也就是說，當您新增每一個佇列管理程式時，您會針對該佇列管理程式執行一次作業。

如果您使用指令來變更各種 z/OS 系統參數，並依照建議來執行 [第 710 頁的『更新 SYS1.PARMLIB 成員』](#)，則沒有任何作業需要您對 z/OS 系統執行 IPL。

為了簡化作業並協助進行問題判斷，請確保 Sysplex 中的所有 z/OS 系統都以相同方式設定，以便在緊急情況下可以在任何系統上快速建立佇列管理程式。

為了易於維護，請考慮定義別名來參照 IBM MQ 程式庫；如需相關資訊，請參閱 [使用別名來參照 IBM MQ 程式庫](#)。

相關概念

[IBM MQ for z/OS 概念](#)

[第 784 頁的『將 IBM MQ 與 IMS 搭配使用』](#)

IBM MQ -IMS 配接器和 IBM MQ - IMS 橋接器是兩個元件，可讓 IBM MQ 與 IMS 互動。

[第 791 頁的『將 IBM MQ 與 CICS 搭配使用』](#)

如果要將 IBM MQ 與 CICS 搭配使用，您必須配置 IBM MQ CICS 配接器，並選擇性地配置 IBM MQ CICS bridge 元件。

[第 793 頁的『在 IMS 中使用 OTMA 結束程式』](#)

如果您想要使用 IMS Open Transaction Manager Access exits with IBM MQ for z/OS，請使用這個主題。

相關工作

[第 759 頁的『在 z/OS 上設定與其他佇列管理程式的通訊』](#)

本節說明您在開始使用分散式佇列之前需要進行的 IBM MQ for z/OS 準備。

[管理 IBM MQ for z/OS](#)

相關參考

[第 791 頁的『升級及套用服務至 Language Environment 或 z/OS Callable Services』](#)

您必須採取的動作會因您使用 CALLLIBS 或 LINK，以及您的 SMP/E 版本而有所不同。

相關資訊

[IBM MQ for z/OS 的程式目錄](#)

配置 IBM MQ 的 z/OS 系統

請使用這些主題作為逐步手冊，以自訂 IBM MQ for z/OS 系統。

識別 z/OS 系統參數

部分作業涉及更新 z/OS 系統參數。您必須知道在執行系統 IPL 時指定了哪些。

- 您需要針對要執行 IBM MQ 的每一個 z/OS 系統執行一次此作業。
- 從舊版移轉時，您可能需要執行這項作業。

SYS1.PARMLIB(IEASYSpp) 包含指向 SYS1.PARMLIB (其中 pp 代表用來執行系統 IPL 的 z/OS 系統參數清單)。

您需要尋找的項目如下：

若為 [第 699 頁的『APF 授權 IBM MQ 載入程式庫』](#)：

PROG=xx 或 APF=aa 指向授權程式機能 (APF) 授權程式庫清單 (成員 PROGxx 或 IEFAPFaa)

若為 [第 700 頁的『更新 z/OS 鏈結清單和 LPA』](#)：

LNK=kk 指向鏈結清單 (成員 LNKLSTkk) LPA=mm 指向 LPA 清單 (成員 LPALSTmm)

若為 [第 703 頁的『更新 z/OS 程式內容表』](#)：

SCH=xx 指向「程式內容表 (PPT)」(成員 SCHEDxx)

若為 [第 703 頁的『將 IBM MQ 子系統定義為 z/OS』](#)：

SSN=ss 指向已定義的子系統清單 (成員 IEFSSNss)

APF 授權 IBM MQ 載入程式庫

APF-授權各種程式庫。部分載入模組可能已獲授權。

- 您需要針對要執行 *IBM MQ* 的每一個 z/OS 系統執行一次此作業。
- 如果您是使用佇列共用群組，則必須確保 *Sysplex* 中每一個 z/OS 系統上的 *IBM MQ* 設定都相同。
- 從舊版移轉時，您可能需要執行這項作業。
- 使用程式庫搜索 (LLA):
 - 部分 *IBM MQ* 用法可能導致高輸入/輸出 (IO) 從檔案庫載入模組。使用作業系統的 LLA 機能可以減少此 IO。
 - 在下列期間可能會發生此高 IO:
 - MQCONN/MQDISC 速率較高的應用程式，例如在 WLM 儲存程序中。
 - 正在載入通道結束程式。如果您具有經常啟動及停止的通道，並使用通道結束程式。
 - SYS1.PARMLIB 指定 LLA 設定。在 LIBRARIES 陳述式中併入程式庫名稱表示一律會從 VLF (Virtual Lookaside Facility) 取得程式副本，因此在大量使用時通常不需要 I/O。
併入 FREEZE 陳述式表示沒有 I/O 可取得相關 DD 陳述式連結目錄 (這通常會比程式載入本身更多 I/O)。
使用作業系統指令 "F LLA , REFRESH " 在對任何這些程式庫進行任何變更之後。

IBM MQ 載入程式庫 thlqual.SCSQAUTH 和 thlqual.SCSQLINK 必須經過 APF 授權。您也必須對國家語言特性 (thlqual.SCSQANLx 和 thlqual.SCSQSNLx) 及分散式佇列特性 (thlqual.SCSQMVR1) 的程式庫進行 APF 授權。如果您使用 Advanced Message Security，則也必須 APF 授權程式庫 thlqual.SDRQAUTH。

不過，LPA 中的所有載入模組都會自動獲得 APF 授權。如果 SYS1.PARMLIB 成員 IEASYSpp 包含陳述式：

```
LNKAUTH=LNKLST
```

如果未指定 LNKAUTH，則 LNKAUTH=LNKLST 是預設值。

視您選擇放置在 LPA 或鏈結清單中的內容而定 (請參閱 [第 700 頁的『更新 z/OS 鏈結清單和 LPA』](#))，您可能不需要將程式庫放置在 APF 鏈結清單中

註：您必須 APF-授權 IBM MQ STEPLIB 中包含的所有程式庫。如果您在 STEPLIB 中放置未經 APF 授權的程式庫，則整個程式庫連結會失去其 APF 授權。

APF 清單位於 SYS1.PARMLIB 成員 PROGxx 或 IEAAPFaa。清單包含 APF 授權 z/OS 程式庫的名稱。清單中的項目順序並不重要。如需 APF 清單的相關資訊，請參閱 [z/OS MVS Initialization and Tuning Reference](#) 手冊。

如需調整系統的相關資訊，請參閱 [SupportPac MP16](#)

如果您使用具有動態格式的 PROGxx 成員，則只需要發出 z/OS 指令 SETPROG APF,ADD,DSNAME=hlq.SCSQ XXXX,VOLUME= YYYYYY，變更就會生效：其中 XXXX 會因媒體庫名稱而異，YYYYY 是磁區。否則，如果您使用靜態格式或 IEAAPFa 成員，則必須在系統上執行 IPL。

請注意，您必須使用 APF 清單中程式庫的實際名稱。如果您嘗試使用程式庫的資料集別名，授權會失敗。

相關概念

第 700 頁的『更新 z/OS 鏈結清單和 LPA』

以新版本的早期程式碼庫來更新 LPA 程式庫。其他代碼可以在鏈結清單或 LPA 中。

第 694 頁的『準備在 z/OS 上自訂佇列管理程式』

當使用可安裝特性、國家語言特性的詳細資料，以及測試和設定安全的相關資訊來自訂佇列管理程式時，請使用這個主題。

z/OS 更新 z/OS 鏈結清單和 LPA

以新版本的早期程式碼庫來更新 LPA 程式庫。其他代碼可以在鏈結清單或 LPA 中。

- 您需要針對要執行 IBM MQ 的每一個 z/OS 系統執行一次此作業。
- 如果您使用佇列共用群組，在將任何佇列管理程式移轉至 IBM MQ 9.1.0 之前，您應該將 QSG 中每一個佇列管理程式的早期程式碼重新整理至 IBM MQ 9.1.0 層次。

在每一個 LPAR 上安裝最新的早期程式碼，然後在移轉之前的某個時間點逐一重新整理佇列管理程式。您不需要同時移轉所有佇列管理程式。

- 從舊版移轉時，您可能需要執行這項作業。如需進一步詳細資料，請參閱「程式目錄」。IBM MQ for z/OS 的程式目錄可以從 [IBM 出版品中心](#) 下載 (請參閱 [IBM MQ for z/OS 程式目錄 PDF 檔案](#))。

註：LPA 的資料集是特定版本專用。如果您在系統中使用現有的 LPA，請聯絡您的系統管理者，以決定要使用的 LPA。

早期程式碼

部分 IBM MQ 載入模組需要新增至 MVS，IBM MQ 才能作為子系統。這些模組稱為 Early 程式碼，即使佇列管理程式不在作用中，也可以執行它們。例如，在主控台上發出具有 IBM MQ 指令字首的操作員指令時，此早期程式碼將取得控制權，並檢查它是否需要啟動佇列管理程式，或將要求傳遞至執行中的佇列管理程式。此程式碼會載入至「鏈結壓縮區 (LPA)」。

有一組 Early 模組，用於所有佇列管理程式，這些模組必須位於 IBM MQ 的最高層次。來自較高版本 IBM MQ 的早期程式碼將使用較低版本 IBM MQ 的佇列管理程式，但不會使用相反版本。

早期程式碼包含下列載入模組：

- 媒體庫 thqual.SCSQLINK 中的 CSQ3INI 和 CSQ3EPX
- 程式庫 thqual.SCSQSNL x 中的 CSQ3ECMX，其中 x 是您的語言字母：
 - thlqual.SCSQSNLE，適用於美式英文大小寫混合格式
 - thlqual.SCSQSNLU，適用於美式英文大寫
 - thlqual.SCSQSNLK，適用於日文
 - thlqual.SCSQSNLF，代表法文
 - thlqual.SCSQSNLC，適用於中文

IBM MQ 包括使用者修改，可將 thqual.SCSQSNL i 程式庫的內容移至 thqual.SCSQLINK，並通知 SMP/E。此使用者修改稱為 CSQ8UURL，並在 Long Term Support 或 Continuous Delivery 的 *IBM MQ for z/OS* 的程式目錄中說明。IBM MQ for z/OS 的程式目錄可以從 [IBM 出版品中心](#) 下載 (請參閱 [IBM MQ for z/OS 程式目錄 PDF 檔案](#))。

當您已更新 LPA 程式庫中的早期程式碼時，在下一 z/OS IPL (使用 CLPA 選項) 時，它可供從 SYS1.PARMLIB 中 IEFSSNss 成員中定義的 IPL 期間所新增的所有佇列管理程式子系統使用。

對於稍後新增的任何新佇列管理程式子系統，您可以讓它立即可用，而不需要 IPL (如第 703 頁的『將 IBM MQ 子系統定義為 z/OS』中所述) 將它新增至 LPA，如下所示：

- 如果您未使用 CSQ8UERL，請發出下列 z/OS 指令：

```
SETPROG LPA,ADD,MODNAME=(CSQ3INI,CSQ3EPX),DSNAME=thqua1.SCSQLINK
SETPROG LPA,ADD,MODNAME=(CSQ3ECMX),DSNAME=thqua1.SCSQSNL x
```

- 如果您確實使用 CSQ8UERL，則可以使用下列 z/OS 指令，將早期程式碼載入至 LPA：

```
SETPROG LPA,ADD,MASK=*,DSNAME=thqua1.SCSQLINK
```

- 如果您是使用 Advanced Message Security，則還必須發出下列 z/OS 指令，以在 LPA 中包含其他模組：

```
SETPROG LPA,ADD,MODNAME=(CSQ0DRTM),DSNAME=thqua1.SCSQLINK
```

如果您已套用維護，或想要使用更新版本或版次的 IBM MQ 來重新啟動佇列管理程式，則可以使用下列步驟將早期程式碼提供給現有佇列管理程式。您未執行這些步驟的佇列管理程式會繼續使用它們已使用的早期程式碼版本。除非您特別嘗試將維護套用至所有佇列管理程式，或將它們全部更新至 IBM MQ 的最新版本或版次，否則不需要對 LPAR 上的所有佇列管理程式執行這些步驟。

1. 如本主題先前所述，使用 z/OS SETPROG 指令將它新增至 LPA。
2. 使用 IBM MQ 指令 STOP QMGR 停止佇列管理程式。
3. 請確定 qmgr.REFRESH.QMGR 安全設定檔。請參閱 MQSC 指令、設定檔及其存取層次。
4. 使用 IBM MQ 指令 REFRESH QMGR TYPE (早期) 來重新整理佇列管理程式的早期程式碼。
5. 使用 IBM MQ 指令 START QMGR 重新啟動佇列管理程式。

[MQSC 指令](#)中會說明 IBM MQ 指令 STOP QMGR、REFRESH QMGR 及 START QMGR。

其他代碼

下列程式庫中所有 IBM MQ 提供的載入模組都是重新進入的，可以放在 LPA 中：

- SCSQAUTH
- SCSQANL x，其中 x 是您的語言字母
- SCSQMVR1

重要：不過，如果您將程式庫放在 LPA 中，只要您套用維護，就必須手動將任何已變更的模組複製到 LPA 中。因此，最好將 IBM MQ 載入程式庫放在鏈結清單中，在維護之後可以發出 z/OS 指令 REFRESH LLA 來更新。

特別建議對 SCSQAUTH 使用此選項，以便您不必將它併入數個 STEPLIB 中。只有一個語言程式庫 SCSQANL x 應該放在 LPA 或鏈結清單中。鏈結清單程式庫指定在 SYS1.PARMLIB。

分散式佇列機能及 CICS bridge (但非佇列管理程式本身) 需要存取 Language Environment (LE) 執行時期程式庫 SCEERUN。如果您使用其中任一設施，則需要將 SCEERUN 併入鏈結清單中。

部分模組會在佇列管理程式啟動時載入至 ECSA。在 ECSA 受限環境中，可以改為將這些模組放入 LPA 中。如需相關資訊，請參閱第 701 頁的『將 IBM MQ 廣域模組放入 LPA 中』。

重要：若要使用位於 IBM MQ 9.1 的此機能，您必須套用 APAR PH52358。

相關概念

第 703 頁的『更新 z/OS 程式內容表』

IBM MQ 佇列管理程式需要一些額外的 PPT 項目。

 將 IBM MQ 廣域模組放入 LPA 中

當 IBM MQ for z/OS 佇列管理程式啟動時，它會將部分載入模組 (廣域模組) 載入延伸共用服務區 (ECSA)。在佇列管理程式關閉時，會釋放 ECSA。

重要：若要使用位於 IBM MQ 9.1 的此機能，您必須套用 APAR PH52358。

有 19 個廣域模組 (位於 IBM MQ 9.1)，每個執行中佇列管理程式大約耗用 1.2 MB 的 ECSA。在每個 LPAR 執行多個佇列管理程式且由於 ECSA 或高專用限制而需要減少 ECSA 耗用量的環境中，可以將廣域模組放入 LPA 中。

註: 雖然 CSQ7GPLM 是廣域模組，但不應新增至 LPA。

如果佇列管理程式在其 STEPLIB 中找不到廣域模組，且偵測到該模組位於 LPA 中，則會直接使用 LPA 副本，而不是將模組副本載入至 ECSA。或者，如果佇列管理程式程式碼通常是從鏈結清單載入，則會優先載入 LPA 中的任何廣域模組，而不會載入鏈結清單中的任何廣域模組。

z/OS 共用儲存體追蹤功能 (請參閱 使用共用儲存體追蹤功能) 會針對每一個佇列管理程式，在每一個佇列管理程式的 MSTR 位址空間下追蹤儲存體，並可用來偵測廣域模組正在使用多少空間。

依預設，廣域模組位於 SCSQAUTH 載入程式庫中。如果佇列管理程式的 MSTR 位址空間透過 STEPLIB 連結找到 SCSQAUTH，則會優先使用 LPA 中的任何廣域模組，並載入至 ECSA。

廣域模組如下：

CSQ0GPLM, CSQ3AMGP, CSQ3SSGP, CSQ9PREP,
CSQ9SCNB, CSQGGPLM, CSQMCGLM, CSQMGPLM, CSQRGLM1,
CSQSLD1, CSQVGEPL, CSQVSRX, CSQWDL2, CSQWDL3,
CSQWVZSA, CSQWZDGO, CSQWVZPS, CSQWVGTM, CSQZTDDM

附註:

- 在不同 IBM MQ 版本中，IBM MQ 的廣域模組名稱保持不變。因此，如果您將廣域模組載入至 LPA，則它們應該來自單一 IBM MQ 版本，且只能由以相同 IBM MQ 版本執行的佇列管理程式使用。
- 如果在相同 LPAR 上執行多個版本的 IBM MQ，則在任何給定時間，只有其中一個可以在 LPA 中具有其廣域模組。

如果維護套用至已載入廣域模組至 LPA 的 IBM MQ 安裝，且該維護會更新任何廣域模組，則您應該重新執行下列文字中說明的程序。

程序

若要將廣域模組從 IBM MQ 版本放入 LPA 中，請執行下列步驟：

1. 建立 thlqual.SCSQAUTH 載入程式庫及其內容的副本，例如：thlqual.LOCAL.SCSQAUTH。請確保使用外部安全管理程式 (ESM) 來保護此載入程式庫不受未獲授權的存取。
2. APF 授權 thlqual.LOCAL.SCSQAUTH 載入程式庫；請參閱 第 699 頁的『APF 授權 IBM MQ 載入程式庫』。
3. 使用與 thlqual.LOCAL.SCSQAUTH 相同的屬性來建立新的 h1qual.GLOBAL.SCSQAUTH 載入程式庫。

註: 此載入程式庫不需要 APF 授權。請確保此載入程式庫受到保護，不會使用 ESM 進行未獲授權的存取。

4. 將 19 個廣域模組從 thlqual.LOCAL.SCSQAUTH 複製到 thlqual.GLOBAL.SCSQAUTH。
5. 從 thlqual.LOCAL.SCSQAUTH 中刪除 19 個廣域模組。
6. 將來自 thlqual.GLOBAL.SCSQAUTH 的 19 個廣域模組放入 LPA 中，方法如下：
 - a. 將 thlqual.GLOBAL.SCSQAUTH 新增至 SYS1.PARMLIB 的 LPALSTxx 成員。然後，您必須使用 CLPA 選項來 IPL 系統，以確定檔案庫內容已載入 PLPA 中。
 - b. 使用下列指令將模組動態新增至 LPA：

```
SETPROG  
LPA,ADD,MODNAME=(CSQ0GPLM,CSQ3AMGP,CSQ3SSGP,CSQ9PREP,CSQ9SCNB,CSQGGPLM,  
CSQMCGLM,CSQMGPLM,CSQRGLM1,CSQSLD1,CSQVGEPL,CSQVSRX,CSQWDL2,CSQWDL3,  
CSQWVZSA,CSQWZDGO,CSQWVZPS,CSQWVGTM,CSQZTDDM),DSNAME= thlqual.GLOBAL.SCSQAUTH
```

註: LPALSTxx 是在 LPA 中放置模組的偏好長期方法。

7. 發出下列指令，驗證模組是否位於 LPA 中：

如果模組已順利載入 LPA 中，指令的輸出應該會指出模組的進入點和載入點。

對於需要使用來自 LPA 的廣域模組的每一個佇列管理程式，如果您通常會放置：

1. 鏈結清單中的 thlqual.SCSQAUTH，只要停止並啟動佇列管理程式即可。廣域模組是從 LPA 載入，本端模組是從鏈結清單載入。
2. thlqual.SCSQAUTH 在 MSTR JCL STEPLIB 中，變更 JCL，以便 STEPLIB 使用 thlqual.LOCAL.SCSQAUTH 而非 thlqual.SCSQAUTH。停止並啟動佇列管理程式；會從 LPA 載入廣域模組，並從 STEPLIB 載入本端模組。

CHIN 和 AMSM JCL 可以像任何 IBM MQ 應用程式一樣繼續使用 thlqual.SCSQAUTH。

若要回復佇列管理程式以將廣域模組載入至 ECSA，請執行下列步驟：

1. 停止佇列管理程式
2. 透過移除 LPA1STxx 定義或使用下列指令，在下次 IPL 時從 LPA 移除廣域模組：

```
SETPROG LPA,DELETE,MODNAME=(xxx) FORCE=YES
```

3. 如果 thlqual.LOCAL.SCSQAUTH 位於佇列管理程式的 STEPLIB 中，請將它取代為 thlqual.SCSQAUTH。
4. 重新啟動佇列管理程式。

相關概念

第 700 頁的『更新 z/OS 鏈結清單和 LPA』

以新版本的早期程式碼庫來更新 LPA 程式庫。其他代碼可以在鏈結清單或 LPA 中。

z/OS 更新 z/OS 程式內容表

IBM MQ 佇列管理程式需要一些額外的 PPT 項目。

- 您必須針對要執行 IBM MQ 的每一個 z/OS 系統執行一次此作業。
- 如果您是使用佇列共用群組，則必須確保 Sysplex 中每一個 z/OS 系統上的 IBM MQ 設定都相同。
- 從舊版移轉時，您不需要執行這項作業。
- 當您需要 Advanced Message Security 時，確實需要執行此作業的 CSQ0DSRV 部分。

thlqual.SCSQPROC(CSQ4SCHD) 中提供包含所有必要 PPT 項目的範例。確保將必要項目新增至 PPT，您可以在 SYS1.PARMLIB(SCHEDxx)。

在 z/OS 1.12 以及更新版本中，CSQYASCP 已定義給具有詳細屬性的作業系統，不再需要包含在 PARMLIB 的 SCHEDxx 成員中。

IBM MQ 佇列管理程式會控制交換本身。不過，如果您有大量載入的 IBM MQ 網路，且回應時間很重要，則最好透過新增 CSQXJST PPT 項目，讓 IBM MQ 通道起始程式不可交換，否則會有影響 z/OS 系統其餘部分效能的風險。

如果您需要 Advanced Message Security，請新增 CSQ0DSRV PPT 項目。

發出 z/OS 指令 SET SCH=xx，其中 xx 是 PARMLIB 的 SCHEDxx 成員字尾，讓這些變更生效。

相關概念

第 703 頁的『將 IBM MQ 子系統定義為 z/OS』

更新子系統名稱表格，並決定指令字首字串的慣例。

z/OS 配置佇列管理程式及通道起始程式

請使用這些主題作為逐步指引，以配置佇列管理程式及通道起始程式。

z/OS 將 IBM MQ 子系統定義為 z/OS

更新子系統名稱表格，並決定指令字首字串的慣例。

針對每一個 IBM MQ 佇列管理程式，重複這項作業。從舊版移轉時，您不需要執行這項作業。

相關概念

第 707 頁的『建立 IBM MQ 佇列管理程式的程序』

每一個 IBM MQ 子系統都需要已編目的程序，才能啟動佇列管理程式。您可以建立自己的或使用 IBM 提供的程序庫。

z/OS 更新子系統名稱表格

定義 IBM MQ 子系統時，您必須將登錄新增至子系統名稱表格。

z/OS 的子系統名稱表格，最初取自 SYS1.PARMLIB 成員 IEFSSNss 包含正式定義的 z/OS 子系統的定義。若要定義每一個 IBM MQ 子系統，您必須透過變更 SYS1.PARMLIB，最好是使用 z/OS 指令 SETSSI。

IBM MQ 子系統起始設定支援平行處理，因此 IBM MQ 子系統定義陳述式可以在 IEFSSNss 表格中的 BEGINPARALLEL 關鍵字上方及下方新增，該表格位於 z/OS V1.12 以及更新版本。

如果您使用 SETSSI 指令，則變更會立即生效，而且不需要對您的系統執行 IPL。請確定您已更新 SYS1.PARMLIB 也一樣，如第 710 頁的『更新 SYS1.PARMLIB 成員』中所述，以便變更為後續 IPL 之後仍然有效。

SETSSI 指令用來動態定義 IBM MQ 子系統：

```
SETSSI ADD,S=ssid,I=CSQ3INI,P='CSQ3EPX,cpf,scope'
```

IEFSSNss 中的對應資訊可以透過下列兩種方式之一來指定：

- IEFSSNss 中 IBM MQ 子系統定義的關鍵字參數格式。這是建議方法。

```
SUBSYS SUBNAME(ssid) INITRTN(CSQ3INI) INITPARM('CSQ3EPX,cpf,scope')
```

- IBM MQ 子系統定義的位置參數形式。

```
ssid,CSQ3INI,'CSQ3EPX,cpf,scope'
```

請勿在一個 IEFSSNss 成員中混合兩種格式。如果需要不同的表單，請針對每一種類型使用個別 IEFSSNss 成員，並將新成員的 SSN 運算元新增至 IEASYSpp SYS1.PARMLIB 成員。若要指定多個 SSN，請使用 SSN=(aa, bb, ...) 在 IEASYSpp 中。

在範例中，

ssid

子系統 ID。長度最多為四個字元。所有字元都必須是英數(大寫 A 到 Z、0 到 9)，且必須以英文字母開頭。佇列管理程式將具有與子系統相同的名稱，因此您只能使用 z/OS 子系統名稱及 IBM MQ 物件名稱兩者容許的字元。

cpf

指令字首字串(如需 CPF 的相關資訊，請參閱第 705 頁的『定義指令字首字串(CPF)』)。

scope

系統範圍，在 z/OS Sysplex 中執行時使用(如需系統範圍的相關資訊，請參閱第 706 頁的『Sysplex 環境中的 CPF』)。

第 705 頁的圖 99 顯示 IEFSSNss 陳述式的數個範例。


```
CSQ1,CSQ3INI,'CSQ3EPX,+mqs1cpf,S'
CSQ2,CSQ3INI,'CSQ3EPX,+mqs2cpf,S'
CSQ3,CSQ3INI,'CSQ3EPX,++,S'
```

圖 99: 定義子系統的 IEFSSNss 陳述式範例

註: 當您已在子系統中建立物件時, 無法變更子系統名稱或使用另一個子系統中某個子系統的頁集。若要執行上述任一項, 您必須從一個子系統卸載所有物件及訊息, 並將它們重新載入至另一個子系統。

第 705 頁的表 54 提供一些範例, 顯示子系統名稱與指令字首字串 (CPF) 的關聯, 如第 705 頁的圖 99 中的陳述式所定義。

IBM MQ 子系統名稱	CPF
CSQ1	+mqs1cpf
CSQ2	+mqs2cpf
CSQ3	++

註: IBM MQ 不支援 z/OS 指令 SETSSI 的 ACTIVATE 及 DEACTIVATE 功能。

若要檢查變更的狀態, 請在 SDSF 中發出下列指令: /D SSI,L。您會看到以 ACTIVE 狀態建立的新子系統。

z/OS 定義指令字首字串 (CPF)

IBM MQ 的每一個子系統實例都可以有指令字首字串來識別該子系統。

針對所有子系統採用全系統的 CPF 慣例, 以避免衝突。請遵循下列準則:

- 將 CPF 定義為最多 8 個字元的字串。
- 請勿使用任何其他子系統已在使用的 CPF, 並避免使用系統上定義的 JES 倒退字元作為字串的第一個字元。
- 使用第 706 頁的表 56 中列出的有效字元集中的字元來定義 CPF。
- 請勿使用已定義處理程序的縮寫或可能與指令語法混淆的 CPF。例如, CPF (例如 'D') 與 z/OS 指令 (例如 DISPLAY) 衝突。若要避免發生這種情況, 請使用其中一個特殊字元 (如第 706 頁的表 56 所示) 作為 CPF 字串中的第一個或唯一字元。
- 請勿定義 CPF, 它是現有 CPF 的子集或超集。如需範例, 請參閱第 705 頁的表 55。

子系統名稱	已定義 CPF	遞送至的指令
MQA	!A	MQA
MQB	!B	MQB
MQC1	!C1	MQC1
MQC2	!C2	MQC2
MQB1	!B1	MQB

預期用於子系統 MQB1 的指令 (使用 CPF!B1) 會遞送至子系統 MQB, 因為此子系統的 CPF 是!B, 一個子集!B1。例如, 如果您輸入下列指令:

```
!B1 START QMGR
```

子系統 MQB 接收下列指令:

```
1 START QMGR
```

(在此情況下，它無法處理)。

您可以發出 z/OS 指令 DISPLAY OPDATA 來查看存在哪些字首。

如果您是在 Sysplex 中執行，z/OS 會在 CPF 登錄時診斷此類型的任何衝突 (如需 CPF 登錄的相關資訊，請參閱第 706 頁的『Sysplex 環境中的 CPF』)。

第 706 頁的表 56 顯示您在定義 CPF 字串時可以使用的字元:

字集	內容
英文字母	大寫 A 到 Z，小寫 a 到 z
數字	0 到 9
國家 (見附註)	@ \$# (可表示為十六進位值的字元)
特殊	. □ () * & + - = ¢ < ! ; % _ ? : >

註:

系統會辨識下列國家字元的十六進位表示法: @ 為 X'7C'，\$ 為 X'5B'，# 為 X'7B'。在 U.S 以外的國家/地區。終端機鍵盤上所代表的 U.S 國字元可能會產生不同的十六進位表示法，並導致錯誤。例如，在某些國家/地區中，\$ 字元可能會產生 X'4A'。

分號 (;) 作為 CPF 有效，但在大部分系統上，此字元是指令定界字元。

z/OS Sysplex 環境中的 CPF

請利用這個主題來瞭解如何在 Sysplex 範圍內使用 CPF。

如果在 Sysplex 環境中使用，IBM MQ 會登錄您的 CPF，讓您能夠從 Sysplex 中的任何主控台輸入指令，並將該指令遞送至適當的系統來執行。指令回應會傳回至原始主控台。

定義 Sysplex 作業的範圍

範圍用來決定當您在 Sysplex 環境中執行 IBM MQ 時，IBM MQ 子系統所執行的 CPF 登錄類型。

範圍的可能值如下:

M

系統範圍。

IBM MQ 會在系統 IPL 時向 z/OS 登錄 CPF，並在 z/OS 系統處於作用中的整個時間保持登錄狀態。

IBM MQ 指令必須在連接至執行目標子系統之 z/OS 映像檔的主控台上輸入，或者您必須使用 ROUTE 指令將指令導向至該映像檔。

如果您不是在 Sysplex 中執行，請使用這個選項。

S

Sysplex 已啟動範圍。

當 IBM MQ 子系統啟動時，CPF 會向 z/OS 登錄，並保持作用中狀態，直到 IBM MQ 子系統終止為止。

您必須使用 ROUTE 指令將原始 START QMGR 指令引導至目標系統，但所有進一步的 IBM MQ 指令都可以在任何連接至 Sysplex 的主控台上輸入，並自動遞送至目標系統。

在 IBM MQ 終止之後，您必須使用 ROUTE 指令，將後續的 START 指令引導至目標 IBM MQ 子系統。

X

Sysplex IPL 範圍。

IBM MQ 會在系統 IPL 時向 z/OS 登錄 CPF，並在 z/OS 系統處於作用中的整個時間保持登錄狀態。

IBM MQ 指令可以在任何連接至 Sysplex 的主控台中輸入，並自動遞送至執行目標系統的映像檔。

CPF 範圍為 S 的 IBM MQ 子系統可以定義在 Sysplex 內的一或多個 z/OS 映像檔上，因此這些映像檔可以共用單一子系統名稱表格。不過，您必須確保在您要 IBM MQ 子系統執行所在的 z/OS 映像檔上發出起始 START 指令 (或遞送至該映像檔)。如果您使用此選項，則可以停止 IBM MQ 子系統，並在 Sysplex 內的不同 z/OS 映像檔上重新啟動它，而不需要變更子系統名稱表格或執行 z/OS 系統的 IPL。

CPF 範圍為 X 的 IBM MQ 子系統只能定義在 Sysplex 內的一個 z/OS 映像檔上。如果您使用此選項，則必須為每一個 z/OS 映像檔定義唯一的子系統名稱表格，而這些映像檔需要具有範圍 X 的 CPF 的 IBM MQ 子系統。

如果您要使用 z/OS 自動重新啟動管理程式 (ARM) 來自動重新啟動不同 z/OS 映像檔中的佇列管理程式，則必須在每一個可能重新啟動該佇列管理程式的 z/OS 映像檔中定義每一個佇列管理程式。每一個佇列管理程式都必須以 Sysplex 層面的唯一 4 個字元子系統名稱 (CPF 範圍為 S) 來定義。

z/OS 建立 IBM MQ 佇列管理程式的程序

每一個 IBM MQ 子系統都需要已編目的程序，才能啟動佇列管理程式。您可以建立自己的或使用 IBM 提供的程序庫。

- 針對每一個 IBM MQ 佇列管理程式，重複這項作業。
- 從舊版移轉時，您可能需要修改已編目程序。

針對子系統名稱表格中定義的每一個 IBM MQ 子系統，在程序檔案庫中建立編目程序，以啟動佇列管理程式。IBM 提供的程序庫稱為 SYS1.PROCLIB，但您的安裝可能使用自己的命名慣例。

佇列管理程式啟動型作業程序的名稱是由子系統名稱與字元 MSTR 連結而成。例如，子系統 CSQ1 具有程序名稱 CSQ1MSTR。您定義的每一個子系統都需要一個程序。

您需要併入以您選取的語言包含訊息的檔案庫：

- thlqual.SCSQSNLE，適用於美式英文大小寫混合格式
- thlqual.SCSQSNLU，適用於美式英文大寫
- thlqual.SCSQSNLK，適用於日文
- thlqual.SCSQSNLF，代表法文
- thlqual.SCSQSNLC，適用於中文

本產品說明文件中的許多範例和指示假設您有一個稱為 CSQ1 的子系統。如果最初建立稱為 CSQ1 的子系統以進行安裝驗證及測試，您可能會發現這些範例更容易使用。

thlqual.SCSQPROC 中提供兩個範例啟動型作業程序。成員 CSQ4MSTR 對每一個訊息類別使用一個頁面集，成員 CSQ4MSRR 對訊息主要類別使用多個頁面集。將其中一個程序複製到 SYS1.PROCLIB 的成員 xxxxMSTR (其中 xxxx 是 IBM MQ 子系統的名稱)，或者如果您不是使用 SYS1.PROCLIB，則複製到程序庫。針對您定義的每一個 IBM MQ 子系統，將範例程序複製到程序庫中的成員。

當您已複製成員時，您可以使用成員中的指示，根據每一個子系統的需求來修改它們。如需指定佇列管理程式所使用儲存體限制的相關資訊，請參閱 [建議的區域大小](#)。您也可以使用符號參數，以容許在啟動程序時修改程序。如果您有數個 IBM MQ 子系統，您可能會發現使用 JCL 併入群組適用於程序的共用部分，以簡化未來維護。

如果您使用佇列共用群組，STEPLIB 連結必須包含 Db2 執行時期目標程式庫 SDSNLOAD，且必須經過 APF 授權。只有在無法透過鏈結清單或 LPA 來存取 STEPLIB 連結時，才需要此程式庫。

如果您使用 Advanced Message Security，STEPLIB 連結必須包含 thlqual.SDRQAUTH，且必須獲得 APF 授權。

註：您可以記下引導資料集 (BSDS)、日誌和頁面集的名稱，以在 JCL 中使用，然後在程序的後續步驟中定義這些集。

相關概念

第 708 頁的『建立通道起始程式的程序』

針對每一個 IBM MQ 子系統，修改 CSQ4CHIN 的副本。視您使用的其他產品而定，您可能需要容許存取其他資料集。

建立通道起始程式的程序

針對每一個 IBM MQ 子系統，修改 CSQ4CHIN 的副本。視您使用的其他產品而定，您可能需要容許存取其他資料集。

- 針對每一個 IBM MQ 佇列管理程式，重複這項作業。
- 從舊版移轉時，您可能需要修改已編目程序。

您需要為每一個要使用分散式佇列的 IBM MQ 子系統建立通道起始程式啟動作業程序。

若要執行此作業：

1. 將範例啟動型作業程序 thlqual.SCSQPROC(CSQ4CHIN) 複製到您的程序庫。將程序命名為 *xxxxxx* CHIN，其中 *xxxxxx* 是 IBM MQ 子系統的名稱 (例如，CSQ1CHIN 會是佇列管理程式 CSQ1 的通道起始程式啟動作業程序)。
2. 為您要使用的每一個 IBM MQ 子系統建立副本。
3. 使用範例程序 CSQ4CHIN 中的指示來調整程序以符合您的需求。您也可以使用符號參數，以容許在啟動程序時修改程序。這會以 [管理 IBM MQ for z/OS](#) 中的啟動選項來說明。

連結分散式佇列程式庫 thlqual.SCSQMVR1。

需要存取 LE 執行時期程式庫 SCEERUN; 如果它不在鏈結清單中 (SYS1.PARMLIB(LNKLSTkk)，在 STEPLIB DD 陳述式中連結它。

4. 授權程序在外部安全管理程式下執行。
5. 您需要併入以您選取的語言包含訊息的檔案庫：
 - thlqual.SCSQSNLE，適用於美式英文大小寫混合格式
 - thlqual.SCSQSNLU，適用於美式英文大寫
 - thlqual.SCSQSNLK，適用於日文
 - thlqual.SCSQSNLF，代表法文
 - thlqual.SCSQSNLC，適用於中文

通道起始程式是一個長時間執行的位址空間。若要防止在耗用限制的 CPU 數量之後終止它，請確認下列任一項：

- z/OS 系統中已啟動作業的預設值是無限制 CPU; JOBCLASS (STC) 的 JES2 配置陳述式 (TIME = (1440,00)) 會達到此目的，或
- 明確地將 TIME=1440 或 TIME=NOLIMIT 參數新增至 CSQXJST 的 EXEC 陳述式。

如果您想要使用通道結束程式，稍後可以將結束程式庫 (CSQXLIB) 新增至此程序。您需要停止並重新啟動通道起始程式，才能執行此動作。

如果您使用 TLS，則需要存取系統 TLS 執行時期程式庫。此程式庫稱為 SIEALNKE。磁帶庫必須經過 APF 授權。

如果您使用 TCP/IP，通道起始程式位址空間必須能夠存取 TCPIP.DATA 資料集。必須設定資料集的方式，視您使用的 TCP/IP 產品及介面而定。其中包括：

- 環境變數，RESOLVER_CONFIG
- 檔案系統上的 /etc/resolv.conf
- // SYSTCPD DD 陳述式
- // SYSTCPDD DD 陳述式
- *jobname/userid*.TCPIP.DATA
- SYS1.TCPPARMS(TCPDATA)

- `zapname.TCPIP.DATA`

其中部分會影響您的已啟動作業程序 JCL。如需相關資訊，請參閱 [z/OS Communications Server: IP Configuration Guide](#)。

相關概念

第 709 頁的『將 IBM MQ 子系統定義給 z/OS WLM 服務類別』

若要在 z/OS 系統中提供 IBM MQ 適當的效能優先順序，您必須將佇列管理程式及通道起始程式位址空間指派給適當的 z/OS 工作量管理 (WLM) 服務類別。如果您未明確執行此動作，則可能會套用不適當的預設值。

z/OS 將 IBM MQ 子系統定義給 z/OS WLM 服務類別

若要在 z/OS 系統中提供 IBM MQ 適當的效能優先順序，您必須將佇列管理程式及通道起始程式位址空間指派給適當的 z/OS 工作量管理 (WLM) 服務類別。如果您未明確執行此動作，則可能會套用不適當的預設值。

- 針對每一個 IBM MQ 佇列管理程式，重複這項作業。
- 從舊版移轉時，您不需要執行這項作業。

使用 WLM 提供的 ISPF 對話框來執行下列作業：

- 從 WLM 連結資料集擷取 z/OS WLM 原則定義。
- 透過將佇列管理程式及通道起始程式已啟動作業程序名稱新增至選擇的服務類別，來更新此原則定義
- 在 WLM 連結資料集上安裝已變更的原則

然後使用 z/OS 指令啟動此原則

```
V WLM,POLICY=policyname,REFRESH
```

如需設定效能選項的相關資訊，請參閱 [在 z/OS 上規劃 IBM MQ 環境](#)。

相關概念

第 739 頁的『設定 Db2 環境』

如果您使用佇列共用群組，則必須透過自訂並執行一些範例工作來建立必要的 Db2 物件。

z/OS 實作 *ESM* 安全控制

實作佇列管理程式及通道起始程式的安全控制。

- 針對每一個 IBM MQ 佇列管理程式，重複這項作業。
- 從舊版移轉時，您可能需要執行這項作業。

如果您使用 RACF 作為外部安全管理程式，請參閱 [在 z/OS 上設定安全](#)，其中說明如何實作這些安全控制項。

如果您是使用通道起始程式，則還必須執行下列動作：

- 如果子系統具有作用中的連線安全，請對外部安全管理程式定義連線安全設定檔 `ssid.CHIN` (如需相關資訊，請參閱 [通道起始程式的連線安全設定檔](#))。
- 如果您是使用「傳輸層安全 (TLS)」或 Socket 介面，請確保通道起始程式執行所使用的權限所使用的使用者 ID 已配置為使用「UNIX 系統服務」，如 *OS/390 UNIX 系統服務規劃* 文件中所述。
- 如果您使用 TLS，請確定通道起始程式執行所使用的使用者 ID 已配置為存取 ALTER QMGR 指令的 `SSLKEYR` 參數中指定的金鑰環。

在啟動佇列管理程式之前，請透過下列方式設定 IBM MQ 資料集及系統安全：

- 授權佇列管理程式啟動型作業程序在外部安全管理程式下執行。
- 授權存取佇列管理程式資料集。
- **V 9.1.4** 必要的話，配置 z/OS 資料集加密。

請參閱 [IBM MQ for z/OS 上具有資料集加密之靜態資料的機密性](#) 一節。的文件以取得相關資訊。

如需如何執行此動作的詳細資料，請參閱 [z/OS 的安全安裝作業](#)。

如果您使用 RACF，只要您使用 RACF STARTED 類別，就不需要對系統執行 IPL (請參閱 [RACF 已啟動作業程序的授權](#))。

相關概念

第 710 頁的『[更新 SYS1.PARMLIB 成員](#)』

若要確保您的變更在 IPL 之後仍然有效，您必須更新 SYS1.PARMLIB

第 742 頁的『[實作佇列共用群組的 ESM 安全控制](#)』

實作佇列共用群組中所有佇列管理程式的安全控制，以存取 Db2 及連結機能清單結構。

z/OS 更新 **SYS1.PARMLIB** 成員

若要確保您的變更在 IPL 之後仍然有效，您必須更新 SYS1.PARMLIB

- 您需要針對要執行 *IBM MQ* 的每一個 *z/OS* 系統執行一次此作業。
- 如果您是使用佇列共用群組，則必須確保 *Sysplex* 中每一個 *z/OS* 系統上的 *IBM MQ* 設定都相同。
- 從舊版移轉時，您可能需要執行這項作業。

更新 SYS1.PARMLIB 成員如下：

1. 更新成員 IEFSSNss，如第 703 頁的『[將 IBM MQ 子系統定義為 z/OS](#)』中所述。
2. 變更 IEASYSpp，以便在執行 IPL 時使用下列成員：
 - 第 699 頁的『[APF 授權 IBM MQ 載入程式庫](#)』中使用的 PROGxx 或 IEAAPFa 成員
 - 第 700 頁的『[更新 z/OS 鏈結清單和 LPA](#)』中使用的 LNKLSTkk 及 LPALSTmm 成員
 - 第 703 頁的『[更新 z/OS 程式內容表](#)』中使用的 SCHEDxx 成員
 - 第 703 頁的『[將 IBM MQ 子系統定義為 z/OS](#)』中使用的 IEFSSNss 成員

相關概念

第 710 頁的『[自訂起始設定輸入資料集](#)』

製作範例起始設定輸入資料集的工作副本，並加以修改以符合您的系統需求。

z/OS 自訂起始設定輸入資料集

製作範例起始設定輸入資料集的工作副本，並加以修改以符合您的系統需求。

- 針對每一個 *IBM MQ* 佇列管理程式，重複這項作業。
- 從舊版移轉時，您需要執行這項作業。

每一個 *IBM MQ* 佇列管理程式都會從 *IBM MQ* 起始設定輸入資料集所包含的一系列指令中取得其起始定義。這些資料集由佇列管理程式啟動作業程序中定義的 DD 名稱 CSQINP1、CSQINP2 及 CSQINPT 參照。

這些指令的回應會寫入 DD 名稱 CSQOUT1、CSQOUT2 和 CSQOUTT 所參照的起始設定輸出資料集。

若要保留原件，請製作每一個範例的工作副本。然後，您可以自訂這些工作副本中的指令，以符合您的系統需求。

如果您使用多個 *IBM MQ* 子系統，如果您在起始設定輸入資料集名稱的高階限定元中包括子系統名稱，則可以更輕鬆地識別與每一個資料集相關聯的 *IBM MQ* 子系統。

如需範例的進一步相關資訊，請參閱下列主題：

- [起始設定資料集格式](#)
- [使用 CSQINP1 範例](#)
- [使用 CSQINP2 範例](#)
- [使用 CSQINPX 範例](#)
- [使用 CSQINPT 範例](#)

起始設定資料集格式

起始設定輸入資料集可以是分割的資料集 (PDS) 成員或循序資料集。它們可以是一連串的资料集。請以 80 個位元組的記錄長度來定義它們，其中：

- 只有直欄 1 到 72 是有意義的。系統不處理直欄 73 到 80。
- 直欄 1 中具有星號 (*) 的記錄會解譯為註解，並被忽略。
- 系統不處理空白記錄。
- 每一個指令必須在新記錄上啟動。
- Trailing-means 會從下一筆記錄的直欄 1 繼續。
- 尾端 + 表示從下一筆記錄的第一個非空白直欄繼續。
- 指令中允許的字元數上限為 32 762。

起始設定輸出資料集是循序資料集，記錄長度為 125，記錄格式為 VBA，區塊大小為 629。

使用 CSQINP1 範例

資料集 thlqual.SCSQPROC 保留兩個成員，其中包含緩衝池的定義、緩衝池關聯的頁集，以及 ALTER SECURITY 指令。

成員 CSQ4INP1 會針對每一個訊息類別使用一個頁面集。成員 CSQ4INPR 針對訊息的主要類別使用多個頁集。

在佇列管理程式啟動作業程序的 CSQINP1 連結中包含適當的範例。

附註：

1. IBM MQ 最多支援 100 個緩衝池，範圍從 0 到 99。只能從 CSQINP1 起始設定資料集發出 DEFINE BUFFPOOL 指令。範例中的定義指定四個緩衝池。
2. 佇列管理程式使用的每一個頁集都必須使用 DEFINE PSID 指令在 CSQINP1 起始設定資料集中定義。頁集定義會將緩衝池 ID 與頁集相關聯。如果未指定緩衝池，依預設會使用緩衝池零。
必須定義頁集零 (00)。它包含所有物件定義。您最多可以為每一個佇列管理程式定義 100 個頁集。
3. ALTER SECURITY 指令可用來變更安全屬性 TIMEOUT 及 INTERVAL。在 CSQ4INP1 中，TIMEOUT 的預設值定義為 54，INTERVAL 則定義為 12。

如需組織緩衝池及頁集的相關資訊，請參閱 [在 z/OS 上規劃](#)。

如果您在佇列管理程式執行時動態變更緩衝池及頁集定義，則也應該更新 CSQINP1 定義。除非緩衝池定義包括 REPLACE 屬性，否則只有在冷啟動 IBM MQ 時才會保留變更。

使用 CSQINP2 範例

此表格列出佇列管理程式啟動作業程序的 CSQINP2 連結中可包含的 thlqual.SCSQPROC 成員，以及其功能的說明。命名慣例為 CSQ4INS*。需要針對您的配置修改 CSQ4INY*。您應該避免變更 CSQINS* 成員，因為當您移轉至下一版時，將需要重新套用任何變更。您可以改為將 DEFINE 或 ALTER 指令放置在 CSQ4INY* 成員中。

成員名稱	說明
CSQ4INSG	系統物件定義。
CSQ4INSA	通道鑑別的系統物件及預設規則。
CSQ4INSX	系統物件定義。
CSQ4INSS	如果您使用佇列共用群組，請自訂並併入此成員。
CSQ4INSJ	如果您使用 JMS 來使用發佈/訂閱，請自訂並包含此成員。
CSQ4INSM	Advanced Message Security 的系統物件定義。

表 57: thlqual.SCSQPROC 的成員 (繼續)

成員名稱	說明
CSQ4INSR	如果您使用 WebSphere Application Server, 或 IBM MQ V7 或更新版本中已排入佇列的發佈/訂閱常駐程式所支援的排入佇列發佈/訂閱介面, 請自訂並包含此成員。
CSQ4DISP	CSQINP2 範例, 用於顯示物件定義。
CSQ4INYC	叢集作業定義。
CSQ4INYD	分散式佇列定義。
CSQ4INYG	一般定義。
CSQ4INYS	儲存類別定義, 對訊息的主要類別使用多個頁集。
CSQ4INYS	儲存類別定義, 針對每一個訊息類別使用一個頁集。

您只需要定義物件一次, 而不是每次啟動佇列管理程式時都要定義物件, 因此不需要每次都在 CSQINP2 中包含這些定義。如果您每次都包含它們, 則會嘗試定義已存在的物件, 且會收到類似下列的訊息:

```
CSQM095I +CSQ1 CSQMAQLC QLOCAL(SYSTEM.DEFAULT.LOCAL.QUEUE) ALREADY EXISTS
CSQM090E +CSQ1 CSQMAQLC FAILURE REASON CODE X'00D44003'
CSQ9023E +CSQ1 CSQMAQLC ' DEFINE QLOCAL ' ABNORMAL COMPLETION
```

此失敗不會損壞物件。如果您想要將 SYSTEM 定義資料集保留在 CSQINP2 連結中, 您可以針對每一個物件指定 REPLACE 屬性來避免失敗訊息。

使用 CSQINPX 範例

範例 thlqual.SCSQPROC(CSQ4INPX) 包含您每次啟動通道起始程式時可能想要執行的一組指令。這些通常是與通道相關的指令, 例如 START LISTENER, 每次通道起始程式啟動時都需要這些指令, 而不是每當佇列管理程式啟動時, 且在輸入資料集 CSQINP1 或 CSQINP2 中不容許這些指令。您必須先自訂此範例, 然後才能使用; 然後您可以將它併入通道起始程式的 CSQINPX 資料集。

資料集中包含的 IBM MQ 指令會在通道起始程式起始設定結束時執行, 並將輸出寫入 CSQOUTX DD 陳述式指定的資料集。輸出類似於 IBM MQ 公用程式 (CSQUTIL) 的 COMMAND 函數所產生的輸出。如需詳細資料, 請參閱 [CSQUTIL 公用程式](#)。

您可以指定可從 CSQUTIL 發出的任何 IBM MQ 指令, 而不只是通道指令。在處理 CSQINPX 時, 您可以從其他來源輸入指令。不論前一個指令是否成功, 都會依序發出所有指令。

若要指定指令回應時間, 您可以使用虛擬指令 COMMAND 作為資料集中的第一個指令。這會採用單一選用關鍵字 RESPTIME (nnn), 其中 nnn 是等待每一個指令回應的時間 (以秒為單位)。這是在 5 到 999 的範圍內; 預設值是 30。

如果 IBM MQ 偵測到對四個指令的回應花費太長時間, 則會停止處理 CSQINPX, 且不會發出進一步指令。通道起始程式未停止, 但訊息 CSQU052E 會寫入 CSQOUTX 資料集, 且訊息 CSQU013E 會傳送至主控台。

當 IBM MQ 順利完成 CSQINPX 的處理時, 會將訊息 CSQU012I 傳送至主控台。

使用 CSQINPT 範例

此表格列出可併入佇列管理程式啟動作業程序的 CSQINPT 連結中的 thlqual.SCSQPROC 成員, 以及其功能的說明。

表 58: thlqual.SCSQPROC 的成員	
成員名稱	說明
CSQ4INST	系統預設訂閱定義。
CSQ4INYT	發佈/訂閱定義。

當發佈/訂閱起始設定完成時，會執行資料集中包含的 IBM MQ 指令，並將輸出寫入 CSQOUTT DD 陳述式指定的資料集。輸出類似於 IBM MQ 公用程式 (CSQUTIL) 的 COMMAND 函數所產生的輸出。如需詳細資料，請參閱 [CSQUTIL 公用程式](#)。

相關概念

第 713 頁的『[建立引導和日誌資料集](#)』
使用提供的程式 CSQJU003 來準備引導資料集 (BSDS) 和日誌資料集。

z/OS 建立引導和日誌資料集

使用提供的程式 CSQJU003 來準備引導資料集 (BSDS) 和日誌資料集。

- 針對每一個 IBM MQ 佇列管理程式，重複這項作業。
- **V9.1.4** 如果您使用 z/OS 資料集加密來保護 BSDS 或作用中日誌資料集，則必須先配置此選項，然後在此步驟中配置資料集。
- 從舊版移轉時，您不需要執行這項作業。

V9.1.4 如果您要移轉佇列管理程式，並為作用中日誌資料集或 BSDS 新增 z/OS 資料集加密，則需要轉換資料集。

V9.1.4 請參閱 [IBM MQ for z/OS 上靜態資料的機密性，具有資料集加密](#) 一節 如需配置 z/OS 資料集加密及將現有 IBM MQ 資料集轉換為加密的相關資訊。

執行 CSQJU003 以建立單一或雙重記載環境的範例 JCL 及「存取方法服務 (AMS)」控制陳述式保留在 thlqual.SCSQPROC(CSQ4BSDS) 中。自訂並執行此工作，以建立 BSDS 及日誌，並預先格式化日誌。

重要: 您應該使用最新版本的 CSQ4BSDS，或手動更新 JCL 以使用 RECORDS (850 60)。

啟動型作業程序 CSQ4MSTR(如 [第 707 頁的『建立 IBM MQ 佇列管理程式的程序』](#) 中所述) 在下列格式的陳述式中參照 BSDS:

```
//BSDS1 DD DSN=++HLQ++.BSDS01,DISP=SHR  
//BSDS2 DD DSN=++HLQ++.BSDS02,DISP=SHR
```

BSDS 會參照日誌資料集。

註:

1. 必須在 LOGDEF 步驟中的 SYSPRINT DD 陳述式上指定 BLKSIZE。BLKSIZE 必須是 629。
2. 為了協助識別來自不同佇列管理程式的引導資料集和日誌資料集，請將子系統名稱包含在這些資料集的高階限定元中。
3. 如果您使用佇列共用群組，則必須使用 SHAREOPTIONS (2 3) 來定義引導及日誌資料集。

如需規劃引導和日誌資料集及其大小的相關資訊，請參閱 [在 z/OS 上規劃](#)。

從 IBM MQ 8.0 開始，8 個位元組的日誌 RBA 加強功能可改善佇列管理程式的可用性，如 [較大的日誌相對位元組位址](#) 中所述。在第一次啟動佇列管理程式之前，若要在佇列管理程式上啟用 8 個位元組的日誌 RBA，請在建立記載環境之後執行下列步驟。

1. 使用 **IDCAMS ALTER**，將第 1 版格式 BSDS (使用 CSQJU003 程式建立) 重新命名為類似於 ++HLQ+.V1.BSDS01 的格式。

註: 請確保重新命名資料和索引元件以及 VSAM 叢集。

2. 配置新的 BSDS，其屬性與已定義的屬性相同。這些將變成佇列管理程式在啟動時將使用的第 2 版格式 BSDS。
3. 執行 BSDS 轉換公用程式 (CSQJUCNV)，將第 1 版格式 BSDS 轉換為新的第 2 版格式 BSDS。
4. 順利完成轉換之後，請刪除第 1 版格式 BSDS。

註: **V 9.1.0** 如果佇列管理程式位於佇列共用群組中，則佇列共用群組中的所有佇列管理程式都必須如下所示啟動，才能啟用 8 個位元組的日誌 RBA:

- 如果佇列管理程式位於 IBM MQ 8.0.0，則必須已使用 **OPMODE** 來啟動 (*NEWFUNC,800*)
- 如果佇列管理程式位於 IBM MQ 9.0.0 LTS，則必須已使用 **OPMODE(NEWFUNC,900)** 或 **OPMODE(NEWFUNC,800)** 來啟動。
- 如果佇列管理程式位於 IBM MQ 9.0.n CD、IBM MQ 9.1.0 LTS 或更新版本，則只需要在該層次啟動它。

相關概念

第 714 頁的『定義頁面集』

使用其中一個提供的範例，為每一個佇列管理程式定義頁集。

z/OS 定義頁面集

使用其中一個提供的範例，為每一個佇列管理程式定義頁集。

- 針對每一個 *IBM MQ* 佇列管理程式，重複這項作業。

V 9.1.4 如果您使用 *z/OS* 資料集加密來保護頁面集，則必須先配置此選項，然後在此步驟中配置資料集。

- 從舊版移轉時，您不需要執行這項作業。

V 9.1.4 如果您要移轉佇列管理程式並新增頁面集的 *z/OS* 資料集加密，則需要轉換頁面集。

V 9.1.4 請參閱 *IBM MQ for z/OS* 上靜態資料的機密性，具有資料集加密。一節 如需配置 *z/OS* 資料集加密及將現有 *IBM MQ* 資料集轉換為加密的相關資訊。

為每一個 *IBM MQ* 佇列管理程式定義個別頁集。 `thlqual.SCSQPROC(CSQ4PAGE)` 及 `thlqual.SCSQPROC(CSQ4PAGR)` 包含 *JCL* 及 *z/OS* 存取方法服務 (*AMS*) 控制陳述式，以定義及格式化頁集。成員 *CSQ4PAGE* 針對每一個訊息類別使用一個頁面集，成員 *CSQ4PAGR* 針對主要訊息類別使用多個頁面集。 *JCL* 執行所提供的公用程式 *CSQUTIL*。請檢閱範例，並針對您想要的頁面集數目及要使用的大小自訂它們。如需頁集及如何計算適當大小的相關資訊，請參閱 [在 z/OS 上規劃](#)。

第 707 頁的『建立 *IBM MQ* 佇列管理程式的程序』中說明的起始作業程序 *CSQ4MSTR* 會在下列格式的陳述式中參照頁集:

```
//CSQP00nn DD DISP=OLD,DSN=xxxxxxxx
```

其中 *nn* 是 00 到 99 之間的頁集號碼，而 *xxxxxxxx* 是您定義的資料集。

註:

1. 如果您想要使用動態頁集擴充功能，請確保為每個頁集定義次要範圍。 `thlqual.SCSQPROC(CSQ4PAGE)` 顯示如何執行此動作。
2. 為了協助識別來自不同佇列管理程式的頁集，請在與每一個頁集相關聯之資料集的高階限定元中包含子系統名稱。
3. 如果您想要容許 *FORCE* 選項與公用程式 *CSQUTIL* 的 *FORMAT* 函數搭配使用，則必須在 *AMS DEFINE CLUSTER* 陳述式上新增 *REUSE* 屬性。這在 [管理 IBM MQ for z/OS](#) 中有說明。
4. 如果您的頁集大於 4 GB，則必須使用「儲存體管理系統 (*SMS*) 延伸定址能力」功能。

相關概念

第 741 頁的『將 *IBM MQ* 項目新增至 Db2 表格』

如果您使用佇列共用群組，請執行 *CSQ5PQSG* 公用程式，將佇列共用群組及佇列管理程式項目新增至 Db2 資料共用群組中的 *IBM MQ* 表格。

z/OS 自訂系統參數模組

IBM MQ 系統參數模組可控制 IBM MQ 在其作業中使用的記載、保存、追蹤及連線環境。提供預設模組。您應該建立自己的系統參數模組，因為部分參數 (例如資料集名稱) 通常是網站特定的。

- 視需要針對每一個 IBM MQ 佇列管理程式重複此作業。
- 從舊版移轉時，您可能需要執行這項作業。如需詳細資料，請參閱 [在 z/OS 上移轉 IBM MQ](#)。
- 若要在現有佇列管理程式上啟用 *Advanced Message Security for z/OS*，您只需要依照第 716 頁的『使用 CSQ6SYSP』中的說明，將 SPLCAP 設為 YES。如果您是第一次配置此佇列管理程式，請完成整個作業。

系統參數模組具有 **V 9.1.0** 四個巨集，如下所示：

巨集名稱	用途
CSQ6SYSP	指定連線和追蹤參數，請參閱第 716 頁的『使用 CSQ6SYSP』
CSQ6LOGP	控制日誌起始設定，請參閱第 724 頁的『使用 CSQ6LOGP』
CSQ6ARVP	控制保存檔起始設定，請參閱第 727 頁的『使用 CSQ6ARVP』
V 9.1.0 V 9.1.0 CSQ6USGP	控制使用情形記錄，請參閱第 733 頁的『使用 CSQ6USGP』

IBM MQ 提供預設系統參數模組 CSQZPARM，當您發出 START QMGR 指令 (不含 PARM 參數) 來啟動 IBM MQ 實例時，會自動呼叫此模組。CSQZPARM 位於 IBM MQ 也隨附的 APF 授權程式庫 thlqual.SCSQAUTH 中。當您啟動 IBM MQ 時，這些參數的值會顯示為一系列訊息。

如需如何使用此指令的相關資訊，請參閱 [START QMGR](#)。

建立您自己的系統參數模組

如果 CSQZPARM 不包含您想要的系統參數，您可以使用 thlqual.SCSQPROC(CSQ4ZPRM) 中提供的範例 JCL 來建立自己的系統參數模組。

如果要建立您自己的系統參數模組，請執行下列動作：

1. 建立 JCL 範例的工作副本。
2. 視需要編輯副本中每一個巨集的參數。如果您從巨集呼叫中移除任何參數，則會在執行時期自動挑選預設值。
3. 將位置保留元 ++NAME++ 取代為載入模組要採用的名稱 (這可以是 CSQZPARM)。
4. 如果您的組譯器不是高階組譯器，請依照組譯器的需要來變更 JCL。
5. 執行 JCL 來組合及鏈結編輯系統參數巨集的自訂版本，以產生載入模組。這是具有您指定名稱的新系統參數模組。
6. 將產生的載入模組放置在 APF 授權使用者檔案庫中。
7. 新增使用者對 APF 授權使用者檔案庫的 READ 存取權。
8. 在 IBM MQ 佇列管理程式啟動作業程序 STEPLIB 中包含此程式庫。此媒體庫名稱必須在 STEPLIB 中的媒體庫 thlqual.SCSQAUTH 之前。
9. 當您啟動佇列管理程式時，請呼叫新的系統參數模組。例如，如果新模組命名為 NEWMODS，請發出下列指令：

```
START QMGR PARM(NEWMODS)
```

10. 請檢查工作日誌，確定指令順利完成。日誌中應該有類似下列的項目：

您也可以佇列管理程式啟動 JCL 中指定參數模組名稱。如需相關資訊，請參閱 [啟動及停止佇列管理程式](#)。

註: 如果您選擇將模組命名為 CSQZPARM，則不需要在 START QMGR 指令上指定 PARM 參數。

微調系統參數模組

IBM MQ 也提供一組三個組譯器來源模組，可用來細部調整現有的系統參數模組。這些模組位於程式庫 thlqual.SCSQASMS 中。通常，您在測試環境中使用這些模組來變更系統參數巨集中的預設參數。每一個來源模組都會呼叫不同的系統參數巨集：

此組譯器來源模組 ...	呼叫此巨集 ...
CSQFSYSP	CSQ6SYSP (連線和追蹤參數)
CSQJLOGP	CSQ6LOGP (日誌起始設定)
CSQJARVP	CSQ6ARVP (保存起始設定)

以下是您使用這些模組的方式：

1. 製作使用者組譯器檔案庫中每一個組譯器來源模組的工作副本。
2. 視需要新增或變更任何參數的值，以編輯您的副本。
3. 組合任何已編輯模組的副本，以在使用者物件檔案庫中建立物件模組。
4. 鏈結編輯這些物件程式碼模組與現有的系統參數模組，以產生作為新系統參數模組的載入模組。
5. 請確定新的系統參數模組是使用者授權檔案庫的成員。
6. 將這個程式庫包含在佇列管理程式啟動作業程序 STEPLIB 中。此媒體庫必須在 STEPLIB 中的媒體庫 thlqual.SCSQAUTH 之前。
7. 發出 START QMGR 指令，並在 PARM 參數中指定新的模組名稱，以呼叫新的系統參數模組。

在 SCSQPROC 的成員 CSQ4UZPR 中提供範例 usermod，示範如何在 SMP/E 控制下管理自訂系統參數。

變更系統參數

您可以在佇列管理程式執行時變更部分系統參數；請參閱 [SET SYSTEM](#)、[SET LOG](#) 及 [SET ARCHIVE](#) 指令。

將 SET 指令放置在起始設定輸入資料集中，讓它們在您每次啟動佇列管理程式時生效。

相關概念

第 734 頁的『自訂通道起始程式參數』

使用 ALTER QMGR 來自訂通道起始程式，以符合您的需求。

使用 CSQ6SYSP

請參閱本主題，以瞭解如何使用 CSQ6SYSP 來設定系統參數。

CSQ6SYSP 的預設參數，以及您是否可以使用 SET SYSTEM 指令來變更每一個參數，會顯示在 [第 716 頁的表 59](#) 中。如果您要變更任何這些值，請參閱參數的詳細說明。

參數	說明	預設值	設定指令
ACELIM	ACE 儲存區的大小，以 1 KB 區塊為單位。	0 (無限制)	✓
CLCACHE	指定要使用的叢集快取類型。	靜態	-
CMDUSER	指令安全檢查的預設使用者 ID。	CSQOPR	-

表 59: CSQ6SYSP 參數的預設值 (繼續)

參數	說明	預設值	設定指令
EXCLMSG	指定要從任何日誌中排除的訊息清單。此清單中的訊息不會傳送至 z/OS 主控台及印副本日誌。因此，從 CPU 視景中使用 EXCLMSG 參數來排除訊息比使用第 738 頁的『暫停參考訊息』中說明的方法更有效率。	()	✓
EXITLIM	在每次呼叫期間可執行佇列管理程式結束程式的時間 (以秒為單位)。	30	-
EXITTCB	用來執行佇列管理程式結束程式的已啟動伺服器作業數。	8	-
LOGLOAD	IBM MQ 在一個檢查點開始與下一個檢查點之間寫入的日誌記錄數。	500 000	✓
MULCCAPT	決定「測量使用情形計價」內容，此內容控制用來收集「測量使用情形授權收費 (MULC)」所使用資料的演算法。	請參閱 參數說明	-
OTMACON	OTMA 連線參數。	請參閱 參數說明	-
QINDXBLD	決定佇列管理程式重新啟動是等待所有索引重建，還是在所有索引重建之前完成。	等待	-
QMCCSID	佇列管理程式的編碼字集 ID。	0	-
QSGDATA	佇列共用群組參數。	請參閱 參數說明	-
RESAUDIT	RESLEVEL 審核參數。	YES	-
ROUTCDE	指派給未從特定主控台請求之訊息的訊息遞送碼。	1	-
服務	保留供 IBM 使用。	0	✓
SMFACCT	指定在啟動佇列管理程式時是否要收集 SMF 帳戶資料。 請注意，只有在啟動通道起始程式時，才會收集類別 4 通道統計資料。	NO	-
SMFSTAT	指定啟動佇列管理程式時是否要收集 SMF 統計資料。 請注意，只有在啟動通道起始程式時，才會收集類別 4 通道起始程式統計資料。	NO	-
SPLCAP	指定是否在此佇列管理程式上啟用佇列安全原則功能。若為 Advanced Message Security for z/OS，請將此參數設為 YES。	NO	-
STATIME	每次收集統計資料之間的預設時間 (分鐘)。	30	✓
TRACSTR	指定是否自動啟動追蹤。	NO	-
TRACTBL	廣域追蹤機能要使用的追蹤表格大小 (以 4 KB 區塊為單位)。	99 (396 KB)	✓
WLMTIME	掃描 WLM 受管理佇列的佇列索引之間的時間。	30	-
WLMTIMU	WLMTIME 的單位 (分鐘或秒)。	分鐘	-

ACELIM

指定 ACE 儲存區的大小上限（以 1 KB 區塊為單位）。此數字必須在 0-999999 範圍內。預設值為零，表示除了系統中的可用空間以外，沒有任何強制的限制。

您僅應在已識別為使用過多 ECSA 儲存體的佇列管理程式中設定 ACELIM 的值。限制 ACE 儲存區將會限制系統中的連線數，從而限制佇列管理程式使用的 ECSA 儲存體數量。

一旦佇列管理程式達到此限制，應用程式便無法取得新的連線。缺少新連線會導致 MQCONN 處理失敗，透過 RRS 協調的應用程式，可能在任何 IBM MQ API 中發生失敗。

一個 ACE 代表連線的執行緒相關控制區塊所需要的 ECSA 總計大約 12.5%。因此，例如，如果指定 ACELIM=5120，則預期會將佇列管理程式所配置的 ECSA 總量（針對執行緒相關控制區塊）限制為大約 40960K；，即 5120 乘以 8。

為了限制佇列管理程式配置給執行緒相關控制區塊的 ECSA 總量為 5120K，需要 ACELIM 值 640。

您可以使用統計資料 CLASS(3) 追蹤所產生的 SMF 115 子類型 5 記錄，來監視 'ACE/PEB' 儲存區的大小，從而為 ACELIM 設定適當的值。

您可以從統計資料 CLASS(2) 追蹤所寫入的 SMF 115 子類型 7 記錄，取得佇列管理程式用於控制區塊的 ECSA 儲存體總數量；也就是將 QSRSPHBT 中的前兩個元素相加。

請注意，您應該將設定 ACELIM 視為一種機制，用來防止 z/OS 映像檔變成行為不當的佇列管理程式，而非將其視為一種手段，用來控制佇列管理程式的應用程式連線。

CLCACHE

指定要使用的叢集快取類型。

叢集快取是一個儲存體區域，用來儲存與叢集相關的資訊。

如果叢集快取是靜態的，則它具有在佇列管理程式啟動時配置的固定大小。如果快取已滿，則會發出 CSQM060E 訊息，且需要更多空間的應用程式要求會收到 MQRC_CLUSTER_RESOURCE_ERROR。

如果您將 CLCACHE 設為動態，則叢集快取可以視需要展開。不過，您首先需要確保任何已安裝的叢集工作量結束程式都可以使用動態快取來運作。

如果已安裝的叢集工作量結束程式無法運作，則會發出動態快取訊息 CSQM061E。

提供 MQXCLWLN 供叢集工作量結束程式以運作方式導覽叢集快取，而不論使用動態或靜態快取。

對於新的佇列管理程式，除非您要使用不支援動態快取的叢集工作量結束程式，否則請設定 CLCACHE=XX_ENCODE_CASE_ONE Dynamic。

對於已使用靜態快取且位於未新增許多新佇列及佇列管理程式的叢集中的現有佇列管理程式，繼續使用 CLCACHE=STATIC 是合理的。

對於已使用靜態快取且位於將要新增許多新佇列或佇列管理程式的叢集中的現有佇列管理程式，請開始使用 CLCACHE=DYNAMIC。

靜態

當叢集快取為靜態時，其大小會在佇列管理程式啟動時固定，足以容納目前的叢集資訊量加上一些擴充空間。當佇列管理程式處於作用中狀態時，無法增加大小。這是預設值。

動態

當叢集快取是動態時，如果在佇列管理程式作用中時需要，則可以自動增加佇列管理程式啟動時所配置的起始大小。

CMDUSER

指定用於指令安全檢查的預設使用者 ID。此使用者 ID 必須定義給 ESM (例如，RACF)。請指定 1 到 8 個英數字元的名稱。第一個字元必須是字母。

預設值為 CSQOPR。

EXCLMSG

指定要排除的錯誤訊息清單。

此清單是動態的，並使用 SET SYSTEM 指令更新。

預設值是空清單 ()。

所提供的訊息沒有 CSQ 字首，也沒有動作碼字尾 (I-D-E-A)。例如，若要排除訊息 CSQX500I，請將 X500 新增至此清單。此清單最多可以包含 16 個訊息 ID。

若要符合併入清單中的資格，必須在正常啟動 MSTR 或 CHIN 位址空間之後發出訊息，且開頭為下列其中一個字元：E、H、I、J、L、M、N、P、R、T、V、W、X、Y、2、3、5、9。

因為處理指令而發出的訊息 ID 可以新增至清單中，但不會排除。例如，由於 DISPLAY USAGE PSID (*) 指令而發出訊息 ID，不過無法抑制此訊息。

EXITLIM

指定每次呼叫佇列管理程式結束程式所容許的時間 (秒)。(此參數對通道結束程式沒有作用。)

請指定 5 到 9999 範圍內的值。

預設值為 30。佇列管理程式會每隔 30 秒輪詢一次執行中的結束程式。在每一個輪詢上，任何已執行超過 EXITLIM 指定的時間的都會強制終止。

EXITTCB

指定在佇列管理程式中用來執行結束程式的已啟動伺服器作業數。(此參數對通道結束程式沒有作用。) 您指定的數字必須至少等於佇列管理程式可能必須執行的結束程式數目上限 (通道結束程式除外)，否則會因 6c6 異常終止而失敗。

請指定 0 到 99 範圍內的值。零值表示無法執行任何結束程式。

預設值為 8。

LOGLOAD

指定 IBM MQ 在一個檢查點開始與下一個檢查點之間寫入的日誌記錄數。IBM MQ 會在已寫入您指定的記錄數之後，啟動新的檢查點。

請指定 200 到 16 000 000 範圍內的值。

預設值為 500 000。

值越大，IBM MQ 的效能越好；不過，如果參數設為較大的值，則重新啟動會花費較長的時間。

建議設定：

測試系統	10,000
正式作業系統	500 000

在正式作業系統中，提供的預設值可能會導致檢查點頻率太高。

LOGLOAD 的值會決定佇列管理程式檢查點的頻率。值太大表示在檢查點之間將大量資料寫入日誌，導致在失敗之後增加佇列管理程式正向回復重新啟動時間。值太小會導致檢查點在尖峰負載期間過於頻繁發生，對回應時間及處理器用量造成不利影響。

LOGLOAD 建議起始值為 500 000。對於每秒 100 個訊息的 1 KB 持續訊息速率 (即 100 個 MQPUT s with commit 及 100 個 MQGET s with commit)，檢查點之間的時間大約是 5 分鐘。

註：這僅作為準則，且此參數的最佳值取決於個別系統的性質。

MULCCAPT

指定用來收集「測量使用授權收費 (MULC)」所使用資料的演算法。

STANDARD

MULC 是根據從 IBM MQ API MQCONN 呼叫到 IBM MQ API MQDISC 呼叫的時間。

已修正

MULC 基於從 IBM MQ API 呼叫開始到 IBM MQ API 呼叫結束的時間。

預設值為 STANDARD

OTMACON

OTMA 參數。此關鍵字採用五個位置參數：

OTMACON = (Group, Member, Druexit, Age, Tpipepfx)

群組

這是此特定 IBM MQ 實例所屬的 XCF 群組名稱。

其長度可以為 1 到 8 個字元，且必須以大寫字元輸入。

預設值為空白，表示 IBM MQ 不得嘗試結合 XCF 群組。

成員

這是 XCF 群組內此特定 IBM MQ 實例的成員名稱。

它的長度可以是 1 到 16 個字元，且必須以大寫字元輸入。

預設值是 4 個字元的佇列管理程式名稱。

德魯 exit

這會指定 IMS 要執行之 OTMA 目的地解析使用者結束程式的名稱。

它的長度可以是 1 到 8 個字元。

預設值為 DFSYDRU0。

這是選用參數; 如果 IBM MQ 要從 IBM MQ 未啟動的 IMS 應用程式接收訊息，則這是必要參數。名稱必須對應於在 IMS 系統中編碼的目的地解析使用者結束程式。如需相關資訊，請參閱 [第 793 頁的『在 IMS 中使用 OTMA 結束程式』](#)。

經歷時間

這代表 IMS 將 IBM MQ 中的使用者 ID 視為先前已驗證的時間長度 (秒)。

它可以在 0 到 2 147 483 647 的範圍內。

預設值為 2 147 483 647。

建議您將此參數與 ALTER SECURITY 指令的 interval 參數一起設定，以維護大型主機中安全快取設定的一致性。

Tpipefx

這代表要用於 Tpipe 名稱的字首。

它包含三個字元; 第一個字元是在 A 到 Z 的範圍內，後續字元是 A 到 Z 或 0 到 9。預設值為 CSQ。

每次 IBM MQ 建立 Tpipe 時都會使用此選項; 其餘名稱由 IBM MQ 指派。您無法為 IBM MQ 所建立的任何 Tpipe 設定完整 Tpipe 名稱。

QINDEXBLD

決定佇列管理程式重新啟動是等待所有佇列索引重建，還是在所有索引重建之前完成。

等待

佇列管理程式重新啟動會等待所有佇列索引建置完成。這表示在建立索引時，在正常 IBM MQ API 處理期間不會延遲任何應用程式，因為在任何應用程式可以連接至佇列管理程式之前，都會先建立所有索引。

這是預設值。

不等待

在完成所有佇列索引建置之前，佇列管理程式可以重新啟動。

QMCCSID

指定佇列管理程式 (因此分散式佇列) 要使用的預設編碼字集 ID。

請指定 0 到 65535 範圍內的值。該值必須代表以 [國家語言](#) 所選擇語言的原生 z/OS 字碼頁列出的 EBCDIC 字碼頁。

零 (預設值) 表示使用目前設定的 CCSID，如果沒有設定，則使用 CCSID 500。這表示如果您已明確將 CCSID 設為任何非零值，則無法透過將 QMCCSID 設為零來重設它; 您現在必須使用正確的非零 CCSID。如果 QMCCSID 是零，您可以發出 DISPLAY QMGR CCSID 指令來檢查實際使用的 CCSID。

QSGDATA

佇列共用群組資料。此關鍵字採用五個位置參數：

QSGDATA = (Qsgname , Dsgname , Db2name , Db2serv , Db2blob)

QSGNAME

這是佇列管理程式所屬的佇列共用群組名稱。

請參閱 [IBM MQ 物件的命名規則](#)，以取得有效字元。名稱：

- 長度可以為 1 到 4 個字元
- 不得以數值開頭
- 不得以 @ 結尾。

這是因為基於實作原因，在內部以 @ 符號填補少於四個字元的名稱，預設值為空白，表示佇列管理程式不是任何佇列共用群組的成員。

茲尼亞梅

這是佇列管理程式要連接的 Db2 資料共用群組名稱。

其長度可以為 1 到 8 個字元，且必須以大寫字元輸入。

預設值為空白，表示您不使用佇列共用群組。

Db2name

這是佇列管理程式要連接的 Db2 子系統或群組附件名稱。

其長度可以為 1 到 4 個字元，且必須以大寫字元輸入。

預設值為空白，表示您不使用佇列共用群組。

註：Db2 子系統 (或群組連接) 必須位於 Dsgname 中指定的 Db2 資料共用群組中，且所有佇列管理程式都必須指定相同的 Db2 資料共用群組。

Db2serv

這是用於存取 Db2 的伺服器作業數。

它可以在 4 到 10 的範圍內。

預設值為 4。

Db2blob

這是用於存取「二進位大型物件 (BLOB)」的 Db2 作業數。

它可以在 4 到 10 的範圍內。

預設值為 4。

如果您指定其中一個名稱參數 (即 **Qsgname**、**Dsgname** 或 **Db2name**)，則必須輸入其他名稱的值，否則 IBM MQ 會失敗。

RESAUDIT

指定是否針對連線處理期間執行的 RESLEVEL 安全檢查寫入 RACF 審核記錄。

請指定下列其中一項：

NO

未執行 RESLEVEL 審核。

YES

RESLEVEL 審核已執行。

預設值為「是」。

ROUTCDE

指定指派給訊息的預設 z/OS 訊息遞送碼，這些訊息不是在 MQSC 指令的直接回應中傳送。

請指定下列其中一項：

1. 1 到 16 (含) 範圍內的值。

2. 以逗點區隔並以括弧括住的值清單。每一個值必須在 1 到 16 (含) 的範圍內。

預設值是 1。

如需 z/OS 遞送碼的相關資訊，請參閱 *z/OS MVS Routing and Descriptor Codes* 手冊其中一個磁區中的 [訊息說明](#)。

SERVICE

此欄位保留供 IBM 使用。

SMFACCT

指定當佇列管理程式啟動時，IBM MQ 是否自動將帳戶資料傳送至 SMF。

請指定下列其中一項：

NO

請勿自動開始收集統計資料。

YES

開始自動收集預設類別 1 的帳戶資料。

整數

自動收集其帳戶的類別清單，範圍為 1 到 4。

預設值為 NO。

SMFSTAT

指定當佇列管理程式啟動時是否自動收集 SMF 統計資料。

請指定下列其中一項：

NO

請勿自動開始收集統計資料。

YES

開始自動收集預設類別 1 的統計資料。

整數

自動收集其統計資料的類別清單，範圍為 1 到 4。若要收集類別 2 或 3 統計資料，也必須指定類別 1。

預設值為 NO。

SPLCAP

安全原則功能可透過原則來啟用更高層次的訊息安全，這些原則可控制在寫入及從佇列讀取訊息時，是否簽署或加密訊息。

透過使用下列其中一個值來配置 SPLCAP，啟用此佇列管理程式的安全原則處理：

NO

在佇列管理程式起始設定期間，未啟用為佇列實作訊息安全原則的功能。

YES

LTS 在佇列管理程式起始設定期間，會啟用訊息安全功能。

如果設定此控制項，佇列管理程式會在起始設定期間嘗試從 SDRQAUTH 載入授權啟用模組，並啟動其他位址空間 (AMSM)。

除非已授權 AMS，且訊息安全的必要配置已備妥，否則佇列管理程式不會啟動。

V9.1.3 如果佇列管理程式在 IBM MQ 9.1.2 或更早版本中執行，它會檢查 SDRQAUTH 程式庫是否屬於佇列管理程式的 STEPLIB，且包含 AMS 啟用模組。

如果在 IBM MQ 9.1.3 或更新版本中執行，佇列管理程式會檢查 AMSPROD 屬性是否設為 AMS、ADVANCED 或 ADVANCEDVUE 其中之一。

如果這些檢查通過，則在佇列管理程式起始設定期間會啟用訊息安全功能，並啟動 AMSM 位址空間。

如果這些檢查失敗，或訊息安全的必要配置已備妥，則佇列管理程式無法啟動。

預設值為 NO。

STATIME

指定連續聚集統計資料之間的預設時間 (分鐘)。

請指定 0 到 1440 範圍內的數字。

如果您指定一個零值，則在 SMF 資料收集廣播時會收集統計資料及帳戶資料。如需設定此項的相關資訊，請參閱 [使用系統管理機能](#)。

預設值為 30。

TRACSTR

指定是否要自動啟動廣域追蹤。

請指定下列其中一項：

NO

請勿自動啟動廣域追蹤。

YES

自動啟動預設類別類別 1 的廣域追蹤。

整數

要在 1 到 4 範圍內自動啟動廣域追蹤的類別清單。

自動啟動所有類別的廣域追蹤。

如果您未在巨集中指定關鍵字，則預設值為 NO。

註：提供的預設系統參數載入模組 (CSQZPARM) 具有 TRACSTR=YES (在組譯器模組 CSQFSYSP 中設定)。如果您不想自動啟動追蹤，請建立您自己的系統參數模組，或在佇列管理程式啟動之後發出 STOP TRACE 指令。

如需 STOP TRACE 指令的詳細資料，請參閱 [停止追蹤](#)。

TRACTBL

指定廣域追蹤機能儲存 IBM MQ 追蹤記錄的追蹤表格預設大小 (以 4 KB 區塊為單位)。

請指定 1 到 999 範圍內的值。

預設值為 99。這相當於 396 KB。

註：在 ECSA 中配置追蹤表格的儲存體。因此，您必須小心選取此值。

WLMTIME

指定每次掃描 WLM 受管理佇列的索引之間的時間 (分鐘或秒，視 WLMTIMU 的值而定)。

請指定 1 到 9999 範圍內的值。

預設值為 30。

WLMTIMU

與 WLMTIME 參數一起使用的時間單位。

請指定下列其中一項：

分鐘

WLMTIME 代表分鐘數。

秒鐘

WLMTIME 代表秒數。

預設值為 MINS。

相關參考

第 724 頁的『[使用 CSQ6LOGP](#)』

請參閱本主題，以瞭解如何使用 CSQ6LOGP 來指定記載選項。

第 727 頁的『[使用 CSQ6ARVP](#)』



請使用本主題作為參照，以瞭解如何使用 CSQ6ARVP 來指定保存環境

z/OS 使用 CSQ6LOGP

請參閱本主題，以瞭解如何使用 CSQ6LOGP 來指定記載選項。

使用 CSQ6LOGP 來建立記載選項。

CSQ6LOGP 的預設參數，以及您是否可以使用 SET LOG 指令來變更每一個參數，會顯示在 [CSQ6LOGP 參數](#) 的預設值中。如果您需要變更任何這些值，請參閱參數的詳細說明。

參數	說明	預設值	設定指令
COMPLOG	控制是否啟用日誌壓縮。	NONE	X
DEALLCT	保存磁帶機在取消配置之前保持未用的時間長度。	零	X
INBUFF	作用中和保存日誌資料集的輸入緩衝區儲存體大小。	60 KB	-
MAXARCH	可記錄的保存日誌磁區數目上限。	500	X
MAXCNOFF	可平行執行的 CSQJOFF7 卸載作業數上限。	31	-
MAXRTU	配置來同時讀取保存日誌磁帶磁區的專用磁帶機數目上限。	2	X
OFFLOAD	開啟或關閉保存。	YES (ON)	-
OUTBUFF	作用中和保存日誌資料集的輸出緩衝區儲存體大小。	4 000 KB	-
TWOACTV	單一或雙重作用中記載。	YES (雙重)	-
TWOARCH	單一或雙重保存記載。	YES (雙重)	-
TWOBSDS	單一或雙重 BSDS。	YES (雙重 BSDS)	-
WRTHRSH	將輸出緩衝區寫入作用中日誌資料集之前要填入的輸出緩衝區數目。	20	X
  ZHYWRITE	指定是否啟用 zHyper 寫入特性。	NO	X

COMPLOG

指定是否啟用日誌壓縮。

指定下列任一項：

NONE

未啟用日誌壓縮。

RLE

使用執行長度編碼來啟用日誌壓縮。

ANY

佇列管理程式會選取提供最大日誌記錄壓縮程度的壓縮演算法。此選項會導致 RLE 壓縮。

預設值為「無」。

如需日誌壓縮的詳細資料，請參閱 [日誌壓縮](#)。

DEALLCT

指定在取消配置保存讀取磁帶機之前，容許其保持未用的時間長度 (以分鐘為單位)。

指定下列其中一項：

- 在 0 到 1440 範圍內的時間 (分鐘)
- NOLIMIT

指定 1440 或 NOLIMIT 表示永不取消配置磁帶機。

預設值為零。

從磁帶讀取保存日誌資料時，建議您將此值設為足夠高，以容許 IBM MQ 針對多個讀取應用程式最佳化磁帶處理。

INBUFF

指定在回復期間讀取作用中及保存日誌的輸入緩衝區大小 (以 KB 為單位)。請使用 28 到 60 範圍內的十進位數。指定的值會四捨五入至 4 的倍數。

預設值為 60 KB。

建議設定:

測試系統 28 KB

正式作業系統 60 KB

將此項設為最大值，以取得最佳日誌讀取效能。

MAXARCH

指定可在 BSDS 中記錄的保存日誌磁區數目上限。當超出此數目時，會在 BSDS 開始時重新開始記錄。

請使用 10 到 1000 範圍內的十進位數。

預設值為 500。

建議設定:

測試系統 500 (預設值)

正式作業系統 1 000

將此值設為上限，以便 BSDS 可以記錄儘可能多的日誌。

如需日誌和 BSDS 的相關資訊，請參閱 [管理 IBM MQ 資源](#)。

MAXCNOFF

指定可平行執行的 CSQJOFF7 卸載作業數。

這容許調整佇列管理程式或佇列管理程式，使其不會使用所有可用的磁帶機。

相反地，在嘗試配置任何新的保存資料集之前，佇列管理程式會等到 CSQJOFF7 卸載作業完成為止。

如果佇列管理程式正在保存至磁帶，請設定此參數，讓並行磁帶要求數目不應等於或超出可用的磁帶機數目，否則系統可能會當掉。

請注意，如果使用雙重保存，則每一個卸載作業都會執行這兩個保存，因此需要相應地設定參數。例如，如果佇列管理程式是雙重保存至磁帶，則 MAXCNOFF=2 值將容許最多兩個作用中日誌同時保存至四個磁帶。

如果有數個佇列管理程式共用磁帶機，您應該相應地設定每一個佇列管理程式的 MAXCNOFF。

預設值為 31。

請指定 1 到 31 範圍內的值。

MAXRTU

指定可同時配置來讀取保存日誌磁帶磁區的專用磁帶機數目上限。

此參數及 DEALLCT 參數容許 IBM MQ 最佳化從磁帶機讀取的保存日誌。

請指定 1 到 99 範圍內的值。

預設值是 2。

建議您將此值設為至少比 IBM MQ 可用的磁帶機數目少 1。否則，卸載處理程序可能會延遲，這可能會影響系統效能。針對保存日誌處理期間的傳輸量上限，請指定此選項可能的最大值，並記住您至少需要一個磁帶機來進行卸載處理。

OFFLOAD

指定保存是開啟或是關閉。

指定下列任一項：

YES

正在進行保存

NO

保存已關閉

預設值為「是」。

請注意：除非您在測試環境中工作，否則請 **不要** 關閉保存。如果您關閉它，則無法保證在系統或交易失敗時將回復資料。

OUTBUFF

指定 IBM MQ 用於寫入作用中及保存日誌資料集之輸出緩衝區的儲存體大小總計 (以 KB 為單位)。每一個輸出緩衝區為 4 KB。

參數必須在 128 到 4000 的範圍內。指定的值會四捨五入至 4 的倍數。基於相容性理由，將接受介於 40 至 128 之間的值，並將其視為值 128。

預設值為 4000 KB。

建議設定：

測試系統	400 KB
正式作業系統	4 000 KB

將此值設為上限，以避免日誌輸出緩衝區耗盡。

TWOACTV

指定單一或雙重作用中記載。

指定下列任一項：

NO

單一作用中日誌

YES

雙重作用中日誌

預設值為「是」。

如需使用單一及雙重記載的相關資訊，請參閱 [管理 IBM MQ 資源](#)。

TWOARCH

指定當卸載作用中日誌時，IBM MQ 所產生的保存日誌數目。

指定下列任一項：

NO

單一保存日誌

YES

雙重保存日誌

預設值為「是」。

建議設定：

測試系統	NO
正式作業系統	是 (預設值)

如需使用單一及雙重記載的相關資訊，請參閱 [管理 IBM MQ 資源](#)。

TWOBSDS

指定引導資料集的數目。

指定下列任一項:

NO

單一 BSDS

YES

雙重 BSDS

預設值為「是」。

如需使用單一及雙重記載的相關資訊，請參閱 [管理 IBM MQ 資源](#)。

WRTHRSR

指定在寫入作用中日誌資料集之前要填入的 4 KB 輸出緩衝區數目。

緩衝區數目越大，發生寫入的頻率越低，這會增進 IBM MQ 的效能。如果發生重大事件 (例如確定點)，則可能會在達到此數目之前寫入緩衝區。

指定 1 到 256 範圍內的緩衝區數目。

預設值為 20。

V 9.1.2 ZHYWRITE

指定是否在啟用 zHyperWrite 的情況下寫入作用中日誌。作用中日誌資料集需要位於具備 zHyperWrite 功能的磁區上，才能啟用 zHyperWrite。

如需啟用 zHyperWrite 的作用中日誌的相關資訊，請參閱 [使用 zHyperWrite 與 IBM MQ 作用中日誌](#)。

值可以為:

NO

zHyperWrite 未啟用。

YES

已啟用 zHyperWrite。

相關參考

第 716 頁的『[使用 CSQ6SYSP](#)』
請參閱本主題，以瞭解如何使用 CSQ6SYSP 來設定系統參數。

第 727 頁的『[使用 CSQ6ARVP](#)』
請使用本主題作為參照，以瞭解如何使用 CSQ6ARVP 來指定保存環境

z/OS 使用 CSQ6ARVP

請使用本主題作為參照，以瞭解如何使用 CSQ6ARVP 來指定保存環境

使用 CSQ6ARVP 來建立保存環境。

第 727 頁的表 61 中顯示 CSQ6ARVP 的預設參數，以及您是否可以使用 SET ARCHIVE 指令來變更每一個參數。如果您需要變更任何這些值，請參閱參數的詳細說明。如需規劃儲存體的相關資訊，請參閱 [規劃 z/OS 上的儲存體及效能需求](#)。

參數	說明	預設值	設定指令
ALCUNIT	進行主要及次要空間配置的單位。	BLK (區塊)	X
ARCPFX1	第一個保存日誌資料集名稱的字首。	CSQARC1	X
ARCPFX2	第二個保存日誌資料集名稱的字首。	CSQARC2	X
ARCRETN	保存日誌資料集的保留期間 (以天為單位)。	9999	X
ARCWRTC	向操作員傳送關於保存日誌資料集的訊息的路由代碼清單。	1,3,4	X
ARCWTOR	在嘗試裝載保存日誌資料集之前，是否要傳送訊息給操作員並等待回覆。	YES	X

表 61: CSQ6ARVP 參數的預設值 (繼續)

參數	說明	預設值	設定指令
<u>blksize</u>	保存日誌資料集的區塊大小。	28 672	X
<u>Catalog</u>	是否在 ICF 中編目保存日誌資料集。	NO	X
<u>精簡</u>	是否應該壓縮保存日誌資料集。	NO	X
<u>PRIQTY</u>	DASD 資料集的主要空間配置。	25 715	X
<u>PROTECT</u>	建立資料集時是否由 ESM 設定檔保護保存日誌資料集。	NO	X
<u>QUIESCE</u>	指定 ARCHIVE LOG 與 MODE (QUIESCE) 時容許靜止的時間上限 (以秒為單位)。	5	X
<u>SECQTY</u>	DASD 資料集的次要空間配置。請參閱 ALCUNIT 參數，以取得要使用的單元。	540	X
<u>TSTAMP</u>	保存資料集名稱是否應該包括時間戳記。	NO	X
<u>單元</u>	儲存保存日誌資料集第一份副本的裝置類型或裝置名稱。	磁帶	X
<u>UNIT2</u>	儲存保存日誌資料集第二個副本的裝置類型或裝置名稱。	Blank	X

ALCUNIT

指定配置主要及次要空間的單位。

請指定下列其中一項：

CYL

磁柱

TRK

磁軌

BLK

區塊數目

建議您使用 BLK，因為它與裝置類型無關。

預設值是 BLK。

如果保存 DASD 磁區上的可用空間可能片段化，建議您指定較小的主要延伸範圍，並容許擴充至次要延伸範圍。如需作用中日誌空間配置的相關資訊，請參閱 [規劃日誌保存儲存體](#)。

ARCPFX1

指定第一個保存日誌資料集名稱的字首。

請參閱 TSTAMP 參數，以取得資料集命名方式的說明，以及 ARCPFX1 長度的限制。

此參數不能保留空白。

預設值為 CSQARC1。

您可能需要授權與 IBM MQ 佇列管理程式位址空間相關聯的使用者 ID，以建立具有此字首的保存日誌。

ARCPFX2

指定第二個保存日誌資料集名稱的字首。

如需如何命名資料集以及 ARCPFX2 長度限制的說明，請參閱 TSTAMP 參數。

即使 TWOARCH 參數指定為 NO，此參數也不能為空白。

預設值為 CSQARC2。

您可能需要授權與 IBM MQ 佇列管理程式位址空間相關聯的使用者 ID，以建立具有此字首的保存日誌。

ARCRETN

指定建立保存日誌資料集時要使用的保留期間 (以天為單位)。

參數必須在 0 到 9999 的範圍內。

預設值是 9999。

建議設定:

測試系統	3	在測試系統中，可能長時間不需要保存日誌。
正式作業系統	9 999 (預設值)	將此值設為高值可有效地關閉自動保存日誌刪除。

如需捨棄保存日誌資料集的相關資訊，請參閱 [捨棄保存日誌資料集](#)。

ARCWRTC

指定關於保存日誌資料集給操作員的訊息的 z/OS 遞送碼清單。如果 ARCWTOR 設為 NO，則會忽略此欄位。

最多指定 14 個遞送碼，每一個遞送碼的值在 1 到 16 的範圍內。您必須至少指定一個代碼。以逗點區隔清單中的代碼，而不是以空白區隔。

預設值是值清單: 1,3,4。

如需 z/OS 遞送碼的相關資訊，請參閱 *z/OS MVS 系統訊息* 手冊的其中一個磁區中 [訊息說明](#) 中的 遞送碼。

ARCWTOR

指定是否先傳送一則訊息給操作員，並接收回覆，再嘗試裝載保存日誌資料集。

其他 IBM MQ 使用者可能被迫等待資料集裝載，但在 IBM MQ 等待回覆訊息時，它們不受影響。

指定下列任一項:

YES

裝置需要很長時間來裝載保存日誌資料集。例如，磁帶機。

NO

裝置沒有長時間延遲。例如，DASD。

預設值為「是」。

建議設定:

測試系統	NO	
正式作業系統	是 (預設值)	這取決於作業程序。如果使用磁帶機器人，則 NO 可能更適合。

BLKSIZE

指定保存日誌資料集的區塊大小。您指定的區塊大小必須與您在 UNIT 參數中指定的裝置類型相容。

參數必須在 4 097 到 28 672 的範圍內。您指定的值會四捨五入至 4 096 的倍數。

預設值為 28 672。

儲存體管理子系統 (SMS) 資料類別區塊大小會置換此參數 (如果有提供的話)。

如果保存日誌資料集已寫入 DASD，建議您選擇容許每個磁軌有兩個區塊的區塊大小上限。例如，對於 3390 裝置，您應該使用 24 576 的區塊大小。

如果將保存日誌資料集寫入磁帶，則指定最大可能的區塊大小可提高讀取保存日誌的速度。您應該使用 28 672 的區塊大小。

建議設定:

測試系統	視用於保存日誌的媒體而定，使用區塊大小建議。 亦即，適用於磁碟 24 576 及磁帶 28 672。
正式作業系統	視用於保存日誌的媒體而定，使用區塊大小建議。 亦即，適用於磁碟 24 576 及磁帶 28 672。

CATALOG

指定是否在主要整合式型錄機能 (ICF) 型錄中編目保存日誌資料集。

指定下列任一項：

NO

未編目保存日誌資料集

YES

已編目保存日誌資料集

預設值為 NO。

必須編目 DASD 上配置的所有保存日誌資料集。如果您保存至 DASD，且 CATALOG 參數設為 NO，則每次配置保存日誌資料集時都會顯示訊息 [CSQJ072E](#)，且 IBM MQ 會編目資料集。

建議設定：

測試系統	YES
正式作業系統	YES，在 DASD 上配置保存檔時

COMPACT

指定是否對寫入至保存日誌的資料進行壓縮。這個選項只適用於有改進資料記錄功能 (IDRC) 特性的 3480 或 3490 裝置。打開這項特性時，磁帶控制單元中的硬體以遠高於正常情況的密度來撰寫資料，以便在每一個磁區中容納更多資料。如果您不使用具有 IDRC 特性或 3490 基本型號的 3480 裝置 (3490E 除外)，請指定 NO。如果您要壓縮資料，請指定 YES。

指定下列任一項：

NO

不壓縮資料集

YES

壓縮資料集

預設值為 NO。

指定 YES 會對效能產生不利影響。也請注意，壓縮至磁帶的資料可以使用支援 IDRC 特性的裝置來唯讀。如果您必須將保存磁帶傳送至另一個站台以進行遠端回復，則這可能是一個問題。

建議設定：

測試系統	不適用
正式作業系統	NO (預設值)

這僅適用於 3480 和 3490 IDR 壓縮。將此設為 YES 可能會降低回復及重新啟動期間的保存日誌讀取效能；不過，它不會影響寫入磁帶。

PRIQTY

指定 ALCUNITs 中 DASD 資料集的主要空間配置。

該值必須大於零。

預設值為 25 715。

對於日誌資料集或其對應 BSDS 的副本 (以較大者為準)，此值必須足夠。若要判定必要值，請遵循下列程序：

1. 決定已配置的作用中日誌記錄數目 (c)，如第 713 頁的『建立引導和日誌資料集』中所述。
2. 決定每一個保存日誌區塊中 4096 位元組區塊的數目：

$$d = \text{BLKSIZE} / 4096$$

其中 BLKSIZE 是四捨五入值。

3. 如果 ALCUNIT = BLK:

$$\text{PRIQTY} = \text{INT}(c / d) + 1$$

其中 INT 表示向下捨入為整數。

如果 ALCUNIT = TRK:

$$\text{PRIQTY} = \text{INT}(c / (d * \text{INT}(e/\text{BLKSIZE}))) + 1$$

其中 e 是每一個磁軌的位元組數 (若為 3390 裝置，則為 56664)，而 INT 表示向下捨入為整數。

如果 ALCUNIT = CYL:

$$\text{PRIQTY} = \text{INT}(c / (d * \text{INT}(e/\text{BLKSIZE}) * f)) + 1$$

其中 f 是每一個磁柱的磁軌數目 (15 代表 3390 裝置)，而 INT 表示向下捨入為整數。

如需建立日誌和保存資料集的大小的相關資訊，請參閱第 713 頁的『建立引導和日誌資料集』和第 714 頁的『定義頁面集』。

建議設定：

測試系統

1 680

足以保留整個作用中日誌，亦即：

$$10\ 080 / 6 = 1\ 680 \text{ blocks}$$

正式作業系統

保存至磁帶時不適用。

如果保存 DASD 磁區上的可用空間可能片段化，建議您指定較小的主要延伸範圍，並容許擴充至次要延伸範圍。如需作用中日誌空間配置的相關資訊，請參閱在 [z/OS 上規劃](#)。

PROTECT

指定建立保存日誌資料集時，是否要透過獨立的 ESM (外部安全管理程式) 設定檔來保護這些資料集。

指定下列任一項：

NO

未建立設定檔。

YES

卸載日誌時會建立離散資料集設定檔。如果您指定 YES:

- IBM MQ 的 ESM 保護必須處於作用中。
- 與 IBM MQ 佇列管理程式位址空間相關聯的使用者 ID 必須具有建立這些設定檔的權限。
- 如果您要保存至磁帶，則 TAPEVOL 類別必須處於作用中。

否則，卸載會失敗。

預設值為 NO。

QUIESCE

指定當發出 ARCHIVE LOG 指令並指定 MODE (QUIESCE) 時，靜止所容許的時間上限 (以秒為單位)。

參數必須在 1 到 999 的範圍內。

預設值為 5。

SECQTY

指定 ALCUNITs 中 DASD 資料集的次要空間配置。次要延伸範圍最多可以配置 15 次;如需詳細資料,請參閱 *z/OS MVS JCL Reference* 及 *z/OS MVS JCL User 's Guide*。

參數必須大於零。

預設值為 540。

TSTAMP

指定保存日誌資料集名稱中是否包含時間戳記。

指定下列任一項:

NO

名稱不包含時間戳記。保存日誌資料集的名稱如下:

```
arcpxi.A nnnnnn
```

其中 *arcpxi* 是 ARCPFX1 或 ARCPFX2 指定的資料集名稱字首。 *arcpxi* 最多可以有 35 個字元。

YES

名稱包括時間戳記。保存日誌資料集的名稱如下:

```
arcpxi.cyyddd.T hhmsst.A nnnnnn
```

其中 *c* 是 'D' (截至 1999 年且包括 1999 年) 或 'E' (2000 年及更新版本), 而 *arcpxi* 是由 ARCPFX1 或 ARCPFX2 指定的資料集名稱字首。 *arcpxi* 最多可以有 19 個字元。

EXT

名稱包括時間戳記。保存日誌資料集的名稱如下:

```
arcpxi.D yyyddd.T hhmsst.A nnnnnn
```

其中 *arcpxi* 是 ARCPFX1 或 ARCPFX2 指定的資料集名稱字首。 *arcpxi* 最多可以有 17 個字元。

預設值為 NO。

UNIT

指定用來儲存保存日誌資料集第一份副本之裝置的裝置類型或裝置名稱。

指定 1 到 8 個英數字元的裝置類型或單位名稱。第一個字元必須是字母。

此參數不能空白。

預設值為 TAPE。

如果您保存至 DASD, 則可以指定有限磁區範圍的通用裝置類型, 例如 UNIT=3390。

如果您保存至 DASD, 請確定:

- 主要空間配置足夠大, 可包含作用中日誌資料集中的所有資料。
- 保存日誌資料集型錄選項 (CATALOG) 設為 YES。
- 您已使用 BLKSIZE 的適當值。

如果您保存至 TAPE，IBM MQ 最多可以延伸至 20 個磁區。

建議設定：

測試系統	DASD
正式作業系統	磁帶

如需選擇保存日誌位置的相關資訊，請參閱 [在 z/OS 上規劃](#)。

UNIT2

指定用來儲存保存日誌資料集第二份副本之裝置的裝置類型或單元名稱。

指定 1 到 8 個英數字元的裝置類型或單位名稱。第一個字元必須是字母。如果此參數空白，則會使用為 UNIT 參數設定的值。

預設值為空白。

相關參考

第 716 頁的『使用 CSQ6SYSP』

請參閱本主題，以瞭解如何使用 CSQ6SYSP 來設定系統參數。

第 724 頁的『使用 CSQ6LOGP』

請參閱本主題，以瞭解如何使用 CSQ6LOGP 來指定記載選項。

 使用 CSQ6USGP

請使用本主題作為參考，以瞭解如何使用 CSQ6USGP 來設定系統參數。

使用 CSQ6USGP 來控制產品使用情形記錄。

CSQ6USGP 的預設參數顯示在 [第 733 頁的表 62](#) 中。如果您需要變更任何這些值，請參閱參數的詳細說明。



小心： 您無法使用 SET SYSTEM 指令來變更任何這些參數。

參數	說明	預設值
<u>QMGRPROD</u>	要針對其記錄佇列管理程式使用情形的產品	Blank
<u>AMSPROD</u>	要針對其記錄 Advanced Message Security (AMS) 使用情形的產品	Blank

QMGRPROD

指定要記錄佇列管理程式使用情形的產品。

請指定下列其中一項：

MQ

佇列管理程式使用情形記錄為獨立式 IBM MQ for z/OS 產品，產品 ID 為 5655-MQ9。


VUE

佇列管理程式使用情形記錄為獨立式 IBM MQ for z/OS Value Unit Edition (VUE) 產品，產品 ID 為 5655-VU9。

ADVANCEDVUE

佇列管理程式使用情形記錄為 IBM MQ Advanced for z/OS Value Unit Edition 產品的一部分，產品 ID 為 5655-AV1。

AMSPROD

 如果未設定此參數，則不會啟動 AMS 位址空間，且會輸出訊息 [CSQY024I](#)。

指定要記錄 Advanced Message Security 用法的產品 (如果使用的話)。

請指定下列其中一項：

AMS

AMS 用法記錄為獨立式 Advanced Message Security for z/OS 產品，產品 ID 為 5655-AM9。

ADVANCED

AMS 用法記錄為 IBM MQ Advanced for z/OS 產品的一部分，產品 ID 為 5655-AV9。

ADVANCEDVUE

AMS 用法記錄為 IBM MQ Advanced for z/OS Value Unit Edition 產品的一部分，產品 ID 為 5655-AV1。

如需產品使用情形記錄的相關資訊，請參閱 [報告產品資訊](#)。

相關參考

第 716 頁的『使用 CSQ6SYSP』

請參閱本主題，以瞭解如何使用 CSQ6SYSP 來設定系統參數。

第 724 頁的『使用 CSQ6LOGP』

請參閱本主題，以瞭解如何使用 CSQ6LOGP 來指定記載選項。

z/OS 自訂通道起始程式參數

使用 ALTER QMGR 來自訂通道起始程式，以符合您的需求。

- 視需要針對每一個 IBM MQ 佇列管理程式重複此作業。
- 從舊版移轉時，您必須執行這項作業。

許多佇列管理程式屬性可控制分散式佇列作業的運作方式。請使用 MQSC 指令 ALTER QMGR 來設定這些屬性。起始設定資料集範例 thlqual.SCSQPROC(CSQ4INYG) 包含您可以自訂的部分設定。如需相關資訊，請參閱 ALTER QMGR。

每次啟動通道起始程式時，這些參數的值都會顯示為一系列訊息。

配接器、分派器和通道數目上限之間的關係

ALTER QMGR 參數 CHIADAPS 及 CHIDISPS 定義通道起始程式所使用的作業控制區塊 (TCB) 數目。CHIADAPS (配接器) TCB 用來對佇列管理程式發出 IBM MQ API 呼叫。CHIDISPS (分派器) TCB 用來呼叫通訊網路。

ALTER QMGR 參數 MAXCHL 會影響通道在分派器 TCB 上的分佈。

CHIDISPS

如果您有少量通道，請使用預設值。

每一個處理器都有一項作業可最佳化系統效能。由於分派器作業是 CPU 密集的，因此原則是盡可能保持儘可能少的作業忙碌，以將尋找及啟動執行緒所花費的時間減至最少。

CHIDISPS (20) 適用於具有 100 個以上通道的系統。如果有 CHIDISPS (20)，其分派器 TCB 比必要多，則不可能有任何重大缺點。

作為準則，如果您有超過 1000 個通道，則容許每 50 個現行通道有一個分派器。例如，指定 CHIDISPS (40) 以處理最多 2000 個作用中通道。

如果您使用 TCP/IP，則用於 TCP/IP 通道的分派器數目上限為 100，即使您在 CHIDISPS 中指定較大的值也一樣。

CHIADAPS

佇列管理程式的每一個 IBM MQ API 呼叫都與任何其他無關，且可以在任何配接器 TCB 上進行。由於日誌 I/O，使用持續訊息的呼叫可能比使用非持續訊息的呼叫花費更長時間。因此，通道起始程式在許多通道上處理大量持續訊息時，可能需要超過預設 8 個配接卡 TCB，才能達到最佳效能。尤其是在達到的批次大小很小的情況下，因為批次處理結束也需要日誌 I/O，以及使用小型用戶端通道的位置。

正式作業環境的建議值是 CHIADAPS (30)。使用超過此值不可能帶來任何重大額外好處，而且如果這是比必要還要多的配接卡 TCB，則在具有 CHIADAPS (30) 時不可能有任何重大缺點。

MAXCHL

每一個通道在通道啟動時與特定分派器 TCB 相關聯，並保持與該 TCB 相關聯，直到通道停止為止。許多通道可以共用每一個 TCB。MAXCHL 用於將通道分散到可用的分派器 TCB。第一個 (MIN ((MAXCHL/CHIDISPS), 10)) 要啟動的通道會與第一個分派器 TCB 相關聯，依此類推，直到所有分派器 TCB 都在使用中為止。

這對少數通道及大型 MAXCHL 的影響是通道「不會」平均分佈在分派器之間。例如，如果您設定 CHIDISPS (10)，並將 MAXCHL 保留為預設值 200，但只有 50 個通道，則 5 個分派器將分別與 10 個通道相關聯，而 5 個未使用。我們建議將 MAXCHL 設定為實際使用的通道數 (如果這是一個小固定數目)。

如果您變更此佇列管理程式內容，則還必須檢閱 ACTCHL、LU62CHL 及 TCPCHL 佇列管理程式內容，以確保值相容。如需這些內容及其關係的完整說明，請參閱 [佇列管理程式參數](#)。

為通道起始程式設定 z/OS UNIX 系統服務環境

通道起始程式 (CHINIT) 使用 OMVS 執行緒。在建立新的 CHINIT 或修改分派器或 SSLTASKS 數目之前，請先檢閱 OMVS 配置參數。

每一個 CHINIT 都使用 3 + CHIDISP + SSLTASKS OMVS 執行緒。這些會提供 LPAR 中所使用的 OMVS 執行緒總數，以及 CHINIT 啟動作業使用者 ID 所使用的執行緒數目。

您可以使用 **D OMVS,L** 並檢閱 MAXPROCSYS 的現行使用情形、高水使用情形及系統限制 (系統容許的處理程序數目上限)。

如果您要新增 CHINIT 或增加 CHIDISPS 或 SSLTASKS 的值，則必須計算執行緒增加數，並檢閱對 MAXPROCSYS 值的影響。您可以使用 **SETOMVS** 指令來動態變更 MAXPROCSYS，或更新 BPXPRCxx parmlib 值或兩者。

OMVS 參數 MAXPROCUSER 是單一 OMVS 使用者 (具有相同 UID) 可以擁有的 OMVS 執行緒數。執行緒會計入此值。因此，如果您有 2 個 CHINITs 具有相同的已啟動作業使用者 ID，且每個具有 10 個分派器及 3 個 SSLTASKS，則 OMVS uid 有 $2 * (3 + 10 + 3) = 32$ 個執行緒。

您可以發出 **D OMVS,O** 指令來顯示預設 MAXPROCUSER，也可以使用 **SETOMVS** 指令來動態變更 MAXPROCUSER，或更新 BPXPRCxx parmlib 值或兩者。

您可以使用 RACF 指令 **ALTUSER userid OMVS(PROCUSERMAX(nnnn))** 或對等項目來置換每個使用者的此值。

若要啟動通道起始程式，請發出下列指令：

```
START CHINIT
```

若要確保通道起始程式已順利啟動，請檢查 xxxxCHIN(ssidCHIN) 工作日誌中是否沒有 ICH408I 錯誤。

相關概念

第 735 頁的『設定批次、TSO 和 RRS 配接器』

透過將程式庫新增至適當的 STEPLIB 連結，讓應用程式可以使用配接卡。若要滿足配接卡發出的 SNAP 傾出，請配置 CSQSNAP DDname。考慮使用 CSQBDEFV 來改善應用程式的可攜性

[通道起始程式統計資料記錄](#)

設定批次、TSO 和 RRS 配接器

透過將程式庫新增至適當的 STEPLIB 連結，讓應用程式可以使用配接卡。若要滿足配接卡發出的 SNAP 傾出，請配置 CSQSNAP DDname。考慮使用 CSQBDEFV 來改善應用程式的可攜性

- 視需要針對每一個 IBM MQ 佇列管理程式重複此作業。
- 從舊版移轉時，您可能需要執行這項作業。

若要讓配接卡可供使用批次連線的批次及其他應用程式使用，請將下列 IBM MQ 程式庫新增至批次應用程式的 STEPLIB 連結：

- thlqual.SCSQANL x
- thlqual.SCSQAUTH

其中 x 是您國家語言的語言字母。(如果程式庫位於 LPA 或鏈結清單中，則不需要執行此動作。)

若為 TSO 應用程式，請將程式庫新增至 TSO 登入程序中的 STEPLIB 連結，或使用 TSO 指令 TSOLIB 來啟動它們。

如果配接卡偵測到非預期的 IBM MQ 錯誤，它會對 DDname CSQSNAP 發出 z/OS SNAP 傾出，並向應用程式發出原因碼 MQRD_UNEXPECTED_ERROR。如果 CSQSNAP DD 陳述式不在應用程式 JCL 中，或 CSQSNAP 未配置給 TSO 下的資料集，則不會執行傾出。如果發生這種情況，您可以將 CSQSNAP DD 陳述式包含在應用程式 JCL 中，或將 CSQSNAP 配置給 TSO 下的資料集，然後重新執行應用程式。不過，由於部分問題是間歇性，建議您在應用程式 JCL 中包含 CSQSNAP 陳述式，或將 CSQSNAP 配置給 TSO 登入程序中的資料集，以擷取發生失敗的原因。

提供的程式 CSQBDEFV 可改善應用程式的可攜性。在 CSQBDEFV 中，您可以指定要連接的佇列管理程式或佇列共用群組的名稱，而不是在應用程式的 MQCONN 或 MQCONNx 呼叫中指定它。您可以為每一個佇列管理程式或佇列共用群組建立新版 CSQBDEFV。為達成此目的，請遵循下列步驟：

1. 將 IBM MQ 組合程式 CSQBDEFV 從 thlqual.SCSQASMS 複製到使用者程式庫。
2. 提供的程式包含預設子系統名稱 CSQ1。您可以保留此名稱，以進行測試及安裝驗證。若為正式作業子系統，您可以將 NAME=CSQ1 變更為一個至四個字元的子系統名稱，或使用 CSQ1。
如果您使用佇列共用群組，則可以指定佇列共用群組名稱而非 CSQ1。如果您這麼做，程式會對該群組內的作用中佇列管理程式發出連接要求。
3. 組合並鏈結編輯程式，以產生 CSQBDEFV 載入模組。對於組件，請在 SYSLIB 連結中包含程式庫 thlqual.SCSQMACS；使用鏈結編輯參數 RENT, AMODE=31, RMODE=ANY。這會顯示在 thlqual.SCSQPROC(CSQ4DEFV) 中的範例 JCL 中。然後將載入程式庫包含在 z/OS 批次或 TSO STEPLIB 中，在 thlqual.SCSQAUTH 之前。

相關概念

第 736 頁的『設定作業及控制面板』

若要設定作業及控制台，您必須先設定包含必要畫面、EXECs、訊息及表格的檔案庫。若要這樣做，您必須考量要用於畫面的國家語言特性。完成此動作之後，您可以選擇性地更新 IBM MQ 作業和控制台的主要 ISPF 功能表，並變更功能鍵設定。

z/OS 設定作業及控制面板

若要設定作業及控制台，您必須先設定包含必要畫面、EXECs、訊息及表格的檔案庫。若要這樣做，您必須考量要用於畫面的國家語言特性。完成此動作之後，您可以選擇性地更新 IBM MQ 作業和控制台的主要 ISPF 功能表，並變更功能鍵設定。

- 您需要針對要執行 IBM MQ 的每一個 z/OS 系統執行一次此作業。
- 從舊版移轉時，您可能需要執行這項作業。

設定程式庫

請遵循下列步驟來設定 IBM MQ 作業及控制台：

1. 請確定連結中包含的所有程式庫都採用相同的格式 (F、FB、V、VB)，且具有相同的區塊大小，或依減少區塊大小的順序。否則，您在嘗試使用這些畫面時可能會發生問題。
2. 將程式庫 thlqual.SCSQEXEC 併入 SYSEXEC 或 SYSPROC 連結中，或使用 TSO ALTLIB 指令來啟動它。此程式庫 (在安裝期間以固定區塊 80 記錄格式配置) 包含必要的 EXECs。
最好將程式庫放入 SYSEXEC 連結中。不過，如果您要將它放在 SYSPROC 中，則檔案庫必須具有 80 個位元組的記錄長度。
3. 將 thlqual.SCSQAUTH 及 thlqual.SCSQANLx 新增至 TSO 登入程序 STEPLIB，或使用 TSO TSOLIB 指令啟動它 (如果它不在鏈結清單或 LPA 中)。
4. 您可以將 IBM MQ 畫面程式庫永久新增至 ISPF 程式庫設定，或容許在使用畫面時動態設定它們。對於先前的選擇，您需要執行下列動作：

- a. 在 ISPLLIB 連結中包含包含作業和控制面板定義的程式庫。名稱是 thlqual.SCSQPNLx，其中 x 是國家語言的語言字母。
 - b. 在 ISPTLIB 連結中包含包含必要表格的程式庫。名稱是 thlqual.SCSQTBLx，其中 x 是國家語言的語言字母。
 - c. 在 ISPMLIB 連結中併入包含必要訊息的程式庫。名稱是 thlqual.SCSQMSGx，其中 x 是國家語言的語言字母。
 - d. 將包含必要載入模組的程式庫併入 ISPLLIB 連結中。此媒體庫的名稱為 thlqual.SCSQAUTH。
5. 測試您可以從「TSO 指令處理器」畫面存取 IBM MQ 畫面。這通常是 ISPF/PDF 主要選項功能表上的選項 6。您執行的 EXEC 名稱是 CSQOREXX。如果您像步驟 4 一樣將 IBM MQ 程式庫永久放置在 ISPF 設定中，則沒有參數可指定。如果沒有，請使用下列：

```
CSQOREXX thlqual langletter
```

其中 langletter 是識別要使用之國家語言的字母：

- C** 簡體中文
- E** U.S. 英文 (大小寫混合格式)
- F** 法文
- K** 日文
- U** U.S. 英文 (大寫)

更新 ISPF 功能表

您可以更新 ISPF 主功能表，以容許從 ISPF 存取 IBM MQ 作業和控制台。The required setting for &ZSEL is:

```
CMD(%CSQOREXX thlqual langletter)
```

如需 thlqual 和 langletter 的相關資訊，請參閱步驟 [第 737 頁的『5』](#)。

如需詳細資料，請參閱 *z/OS: ISPF Dialog Developer 's Guide and Reference* 手冊。

更新功能鍵及指令設定

您可以使用一般 ISPF 程序來變更畫面所使用的功能鍵及指令設定。應用程式 ID 是 CSQO。

不過，不建議這樣做，因為不會更新說明資訊來反映您所做的任何變更。

相關概念

[第 737 頁的『併入 IBM MQ 傾出格式化成員』](#)

若要能夠使用「互動式問題控制系統 (IPCS)」來格式化 IBM MQ 傾出，您必須更新部分系統檔案庫。

併入 IBM MQ 傾出格式化成員

若要能夠使用「互動式問題控制系統 (IPCS)」來格式化 IBM MQ 傾出，您必須更新部分系統檔案庫。

- 您需要針對要執行 IBM MQ 的每一個 z/OS 系統執行一次此作業。
- 從舊版移轉時，您需要執行這項作業。

若要能夠使用「互動式問題控制系統 (IPCS)」來格式化 IBM MQ 傾出，請將資料集 `thlqual.SCSQPROC(CSQ7IPCS)` 複製到 `SYS1.PARMLIB`。您應該不需要編輯此資料集。

如果您已自訂 IPCS 的 TSO 程序，`thlqual.SCSQPROC(CSQ7IPCS)` 可以複製到 `IPCSPARM` 定義中的任何程式庫。如需 `IPCSPARM` 的詳細資料，請參閱 [z/OS MVS IPCS 自訂作業 手冊](#)。

您也必須將程式庫 `thlqual.SCSQPNLA` 包含在 `ISPPLIB` 連結中。

若要讓傾出格式化程式可供 TSO 階段作業或 IPCS 工作使用，您也必須在 `STEPLIB` 連結中包含程式庫 `thlqual.SCSQAUTH`，或使用 TSO `TSOLIB` 指令來啟動它 (即使它已在鏈結清單或 LPA 中)。

相關概念

第 738 頁的『暫停參考訊息』

您的 IBM MQ 系統可能會產生大量參考訊息。您可以防止選取的訊息傳送至主控台或印刷日誌。

暫停參考訊息

您的 IBM MQ 系統可能會產生大量參考訊息。您可以防止選取的訊息傳送至主控台或印刷日誌。

- 您需要針對要執行 *IBM MQ* 的每一個 z/OS 系統執行一次此作業。
- 從舊版移轉時，您不需要執行這項作業。

如果大量使用 IBM 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 主控台及印副本日誌。IBM 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。

必要的話，您可以使用 `SYS1.PARMLIB` 的 `MPFLSTxx` 成員指定的 z/OS 訊息處理機能清單，來暫停其中一些主控台訊息。您指定的訊息仍會出現在印副本日誌上，但不會出現在主控台上。

範例 `thlqual.SCSQPROC(CSQ4MPFL)` 顯示 `MPFLSTxx` 的建議設定。如需 `MPFLSTxx` 的相關資訊，請參閱 [z/OS MVS Initialization and Tuning Reference](#) 手冊。

如果您想要在紙本日誌上暫停選取的參考訊息，您可以使用 z/OS 安裝結束程式 `IEAVMXIT`。您可以針對必要的訊息設定下列位元開關：

CTXTRDTM

刪除訊息。

訊息不會顯示在主控台上，也不會記載在印副本中。

CTXTESJL

抑制工作日誌。

訊息不會進入 JES 工作日誌。

CTXTNWTP

不執行 WTP 處理。

訊息不會傳送至 TSO 終端機或批次工作的系統訊息資料集。

註：

1. 如需其他參數的完整資料，請參閱 [MVS Installation Exits](#) 文件。
2. 不建議您暫停建議的抑制清單 `CSQ4MPFL` 中的其他訊息。

此外，您還可以指定額外參數：

EXCLMSG

指定要從任何日誌中排除的訊息清單。

此清單中的訊息不會傳送至 z/OS 主控台及印副本日誌。如需進一步資訊，請參閱 [第 716 頁的『使用 CSQ6SYSP』](#) 中的 [EXCLMSG](#)。

相關工作

第 751 頁的『在 z/OS 上測試佇列管理程式』

當您已自訂或移轉佇列管理程式時，您可以執行安裝驗證程式及 IBM MQ for z/OS 隨附的部分範例應用程式來測試它。

z/OS 配置佇列共用群組

如果您要使用共用佇列以取得高可用性，請使用這些主題作為逐步指引，以配置佇列共用群組。

當您完成 IBM MQ for z/OS 系統設定程序的這個部分中的步驟時，您應該第 715 頁的『自訂系統參數模組』新增佇列共用群組資料。您需要修改 `CSQ6SYSP` 以指定 `QSGDATA` 參數。

z/OS 設定 Db2 環境

如果您使用佇列共用群組，則必須透過自訂並執行一些範例工作來建立必要的 Db2 物件。

設定 Db2 環境

您必須自訂並執行一些範例工作，以建立及連結必要的 Db2 物件。

- 針對每一個 Db2 資料共用群組重複此作業。
- 從舊版移轉時，您需要執行 `bind` 和 `grant` 步驟。
- 如果您未使用佇列共用群組，請省略此作業。

如果您稍後想要使用佇列共用群組，請在該時間執行此作業。

IBM MQ 提供兩組相等的工作。具有 `CSQ45` 字首的那些版本適用於與舊版 IBM MQ 的相容性，以及與 Db2 第 11 版及更舊版本的相容性。如果您使用 Db2 V12 或更新版本來設定新的資料共用群組，則建議您使用具有 `CSQ4X` 字首的工作，因為這些工作會利用較新的 Db2 功能來動態調整大小及「通用表格空間 (UTS)」

您必須建立一個環境，其中 IBM MQ 可以存取及執行用於佇列共用群組的 Db2 計劃。

必須針對每一個新的 Db2 資料共用群組執行下列步驟。所有範例 JCL 都位於 `thlqual.SCSQPROC` 中。

1. 自訂並執行範例 JCL `CSQ45CSG V9.1.0` (或 `CSQ4XCSG`)，以建立要用於 IBM MQ 資料庫、表格空間及表格的儲存體群組。
2. 自訂並執行範例 JCL `CSQ45CDB V9.1.0` (或 `CSQ4XCDB`)，以建立要供連接至此 Db2 資料共用群組的所有佇列管理程式使用的資料庫。
3. 自訂並執行範例 JCL `CSQ45CTS V9.1.0` (或 `CSQ4XCTS`)，以建立表格空間，其中包含用於佇列共用群組的佇列管理程式及通道起始程式表格 (要在步驟 1 中建立)。
4. 自訂並執行範例 JCL `CSQ45CTB V9.1.0` (或 `CSQ4XCTB`)，以建立 12 個 Db2 表格及相關聯的索引。請勿變更任何列名稱或屬性。
5. 自訂並執行範例 JCL `CSQ45BPL`，以連結佇列管理程式、公用程式及通道起始程式的 Db2 計劃。
6. 自訂並執行範例 JCL `CSQ45GEX`，以將執行權限授與佇列管理程式、公用程式及通道起始程式所使用者 ID 的計劃。佇列管理程式及通道起始程式的使用者 ID 是用來執行其已啟動作業程序的使用者 ID。公用程式的使用者 ID 是可用來提交批次工作的使用者 ID。

下表顯示的適當方案名稱：

- **LTS** LTS 直欄中的 Long Term Support 版本。
- **CD** CD 直欄中的 Continuous Delivery 版本，其中 n 代表 CD 版次。

在每一個版本中，n 會以 1 為增量。例如，在 IBM MQ 9.0.3 中，`CSQ5A90n` 是 `CSQ5A903`。

使用者	計劃 (LTS)	計劃 (CD)
佇列管理程式	CSQ5A 910、 CSQ5C 910、 CSQ5D 910、 CSQ5K 910、 CSQ5L 910、 CSQ5M 910、 CSQ5P 910、 CSQ5R 910、 CSQ5S 910、 CSQ5T 910、 CSQ5U 910、 CSQ5W 910	CSQ5A 91n、 CSQ5C 91n、 CSQ5D 91n、 CSQ5K 91n、 CSQ5L 91n、 CSQ5M 91n、 CSQ5P 91n、 CSQ5R 91n、 CSQ5S 91n、 CSQ5T 91n、 CSQ5U 91n、 CSQ5W 91n
CSQUTIL 批次公用程式的 SDEFS 函數	CSQ52 910	CSQ52 91n
CSQ5PQSG 及 CSQJUCNV 批次公用程式	CSQ5B 910	CSQ5B 91n
CSQUZAP 服務公用程式	CSQ5Z 910	CSQ5Z 91n

如果在 Db2 設定期間失敗，則可以自訂並執行下列工作：

- CSQ45DTB，用來捨棄表格和索引。
- CSQ45DTS **V 9.1.0** (或 CSQ4XDTS)，以捨棄表格空間。
- CSQ45DDB **V 9.1.0** (或 CSQ4XDDB)，以捨棄資料庫。
- CSQ45DSG **V 9.1.0** (或 CSQ4XDSG)，以捨棄儲存體群組。

註：如果這些工作因 Db2 鎖定問題而失敗，則可能是因為競用 Db2 資源，特別是當系統大量使用時。稍後重新提交工作。當系統很少使用或靜止時，最好執行這些工作。

如需設定 Db2 的相關資訊，請參閱 *Db2 for z/OS 11.0.0* 中的 [Db2 管理](#)。

如需設定 Db2 的相關資訊，請參閱 *Db2 for z/OS 12.0.0* 中的 [Db2 管理](#)。

如需 Db2 表格大小的相關資訊，請參閱 [在 z/OS 上規劃](#)。

相關概念

第 740 頁的『設定連結機能』

如果您使用佇列共用群組，請使用 IXCMIAPU，在連結機能「資源管理 (CFRM)」原則資料集中，定義佇列共用群組 (QSG) 中的佇列管理程式所使用的連結機能結構。

z/OS 設定連結機能

如果您使用佇列共用群組，請使用 IXCMIAPU，在連結機能「資源管理 (CFRM)」原則資料集中，定義佇列共用群組 (QSG) 中的佇列管理程式所使用的連結機能結構。

- 針對每一個佇列共用群組，重複這項作業。
- 從舊版移轉時，您可能需要執行這項作業。
- 如果您未使用佇列共用群組，請省略此作業。

如果您稍後想要使用佇列共用群組，請在該時間執行此作業。

佇列共用群組的所有結構都以佇列共用群組的名稱開頭。定義下列結構：

- 稱為 *qsg-name* CSQ_ADMIN 的管理結構。此結構由 IBM MQ 本身使用，且不包含任何使用者資料。
- 稱為 *qsg-name* CSQSYSAPPL 的系統應用程式結構。IBM MQ 系統佇列會使用此結構來儲存狀態資訊。

- 一個以上用來保留共用佇列之訊息的結構。這些可以具有您選擇的任何名稱，長度最多為 16 個字元。
 - 前四個字元必須是佇列共用群組名稱。(如果佇列共用群組名稱長度少於四個字元，則必須以 @ 符號填補至四個字元。)
 - 第五個字元必須是英文字母，後續字元可以是英文字母或數值。此部分名稱(不含佇列共用群組名稱)是您在定義共用佇列或 CF 結構物件時指定的 CFSTRUCT 名稱。

在用來保留共用佇列之訊息的結構名稱中，您只能使用英文字母和數值字元，不能使用任何其他字元(例如，在管理結構名稱中使用的 _ 字元)。

IXCMIAPU 的範例控制陳述式位於資料集 thlqual.SCSQPROC(CSQ4CFRM)。自訂這些，並將它們新增至連結機能的 IXCMIAPU 工作，然後執行它。

順利定義結構之後，請啟動正在使用的 CFRM 原則。若要執行此動作，請發出下列 z/OS 指令：

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME= policy-name
```

如需規劃 CF 結構及其大小的相關資訊，請參閱 [定義連結機能資源](#)。

相關概念

第 709 頁的『實作 ESM 安全控制』
實作佇列管理程式及通道起始程式的安全控制。

z/OS 設定 SMDS 環境

如果您要使用 SMDS 來卸載共用佇列上的訊息，請設定 SMDS 卸載儲存體環境。

- 針對您要配置以將資料卸載至 SMDS 之佇列共用群組中的每一個佇列管理程式及結構執行此作業。
- 如果您稍後想要配置其他結構以將資料卸載至 SMDS，則可以在該時間再次執行此作業。
- 如果您不是使用佇列共用群組，請省略此作業。

如果您稍後想要使用佇列共用群組，請在該時間執行此作業。

設定 SMDS 環境

1. 預估結構及資料集空間需求。請參閱 [共用訊息資料集容量考量](#)。
2. 配置及預先格式化資料集。請參閱 [建立共用訊息資料集](#)。
3. 當您將 CF 結構定義至 IBM MQ 時，請確定您已使用 CFLEVEL (5) 及 OFFLOAD (SMDS) 定義 CFSTRUCT。

相關概念

第 740 頁的『設定連結機能』
如果您使用佇列共用群組，請使用 IXCMIAPU，在連結機能「資源管理 (CFRM)」原則資料集中，定義佇列共用群組 (QSG) 中的佇列管理程式所使用的連結機能結構。

z/OS 將 IBM MQ 項目新增至 Db2 表格

如果您使用佇列共用群組，請執行 CSQ5PQSG 公用程式，將佇列共用群組及佇列管理程式項目新增至 Db2 資料共用群組中的 IBM MQ 表格。

- 針對每一個 IBM MQ 佇列共用群組及每一個佇列管理程式，重複這項作業。
- 從舊版移轉時，您可能需要執行這項作業。
- 如果您不是使用佇列共用群組，請省略此作業。

如果您稍後想要使用佇列共用群組，請在該時間執行此作業。

針對每一個佇列共用群組，以及要成為佇列共用群組成員的每一個佇列管理程式，執行 CSQ5PQSG。(CSQ5PQSG 在 [管理 IBM MQ for z/OS](#) 中說明。)

以指定的順序執行下列動作：

1. 使用 CSQ5PQSG 程式的 ADD QSG 函數，將佇列共用群組項目新增至 IBM MQ Db2 表格中。
thlqual.SCSQPROC(CSQ45AQS) 中提供了範例。

針對 Db2 資料共用群組中定義的每一個佇列共用群組，執行一次此功能。在新增任何參照佇列共用群組的佇列管理程式項目之前，佇列共用群組項目必須存在。

2. 使用 CSQ5PQSG 程式的 ADD QMGR 函數，將佇列管理程式項目新增至 IBM MQ Db2 表格中。
thlqual.SCSQPROC(CSQ45AQM) 中提供範例。

針對要成為佇列共用群組成員的每一個佇列管理程式，執行此功能。

註：

- a. 佇列管理程式只能是一個佇列共用群組的成員。
- b. 您必須執行 RRS，才能使用佇列共用群組。

相關概念

第 715 頁的『[自訂系統參數模組](#)』

IBM MQ 系統參數模組可控制 IBM MQ 在其作業中使用的記載、保存、追蹤及連線環境。提供預設模組。您應該建立自己的系統參數模組，因為部分參數 (例如資料集名稱) 通常是網站特定的。

實作佇列共用群組的 *ESM* 安全控制

實作佇列共用群組中所有佇列管理程式的安全控制，以存取 Db2 及連結機能清單結構。

- 針對佇列共用群組中的每一個 *IBM MQ* 佇列管理程式，重複這項作業。
- 從舊版移轉時，您可能需要執行這項作業。

請確定與佇列管理程式、通道起始程式及公用程式相關聯的使用者 ID 有權建立與您要建立連線之每一個 Db2 子系統的 RRSF 連線。佇列管理程式及通道起始程式的使用者 ID 是用來執行其已啟動作業程序的使用者 ID。

公用程式的使用者 ID 是可用來提交批次工作的使用者 ID。使用者 ID 需要 READ 存取權的 RACF 設定檔是 DSNR 資源類別中的 Db2ssid.RRSF

需要授與與佇列共用群組中每一個佇列管理程式相關聯的使用者 ID 適當層次的連結機能清單結構存取權。RACF 類別是 FACILITY。

下列使用者 ID 需要 ALTER 存取權：

- IXLSTR.structure-name 設定檔的佇列管理程式 ID
- 執行 CSQ5PQSG 的使用者 ID

相關概念

第 709 頁的『[實作 ESM 安全控制](#)』


實作佇列管理程式及通道起始程式的安全控制。

正在配置 Advanced Message Security for z/OS

請使用這些主題作為配置 Advanced Message Security (AMS) 的逐步手冊。

開始之前

在開始配置 AMS 之前，請確定已執行下列佇列管理程式配置步驟：

1.  從 IBM MQ 9.1.3 開始，忽略此步驟。
若為 IBM MQ 9.1.3 之前的 IBM MQ for z/OS 版本，APF 會授權程式庫 thqual.SDRQAUTH，如第 699 頁的『[APF 授權 IBM MQ 載入程式庫](#)』中所述。
2. 將 CSQ0DRTM 模組新增至 LPA，如第 700 頁的『[更新 z/OS 鏈結清單和 LPA](#)』中所述。
3. 將 CSQ0DSRV 項目新增至 z/OS 程式內容表 (PPT)，如第 703 頁的『[更新 z/OS 程式內容表](#)』中所述。
4. 將 CSQ4INSM 成員包含在佇列管理程式啟動作業程序的 CSQINP2 連結中，如第 710 頁的『[自訂起始設定輸入資料集](#)』中所述。

5. **V 9.1.3** 若為 IBM MQ 9.1.3 之前的 IBM MQ for z/OS 版本，請將 thqual.SDRQAUTH 程式庫包含在佇列管理程式 STEPLIB 連結中，如第 707 頁的『[建立 IBM MQ 佇列管理程式的程序](#)』中所述。

從 IBM MQ 9.1.3 開始，您可以使用 AMSPROD 屬性來啟用 AMS。如需詳細資料，請參閱 [使用 IBM MQ for z/OS 產品進行產品使用記錄](#)。

下一步

配置受 AMS 保護之佇列的原則。安全原則在 [管理 Advanced Message Security 安全原則](#) 中有說明。

[z/OS 上的範例配置](#) 中有 AMS 配置的範例。

z/OS 建立 Advanced Message Security 的程序

要配置成使用 Advanced Message Security (AMS) 的每一個 IBM MQ 子系統都需要已編目程序，才能啟動 AMS 位址空間。您可以建立自己的或使用 IBM 提供的程序庫。

程序

1. 將範例啟動型作業程序 *thlqual.SCSQPROC* (CSQ4AMSM) 複製到 SYS1.PROCLIB 或 (如果未使用 SYS1.PROCLIB，您的程序媒體庫。將程序命名為 xxxxAMSM，其中 xxxx 是 IBM MQ 子系統的名稱。例如，CSQ1AMSM 會是佇列管理程式 CSQ1 的 AMS 啟動作業程序。
2. 為您要使用的每一個 IBM MQ 子系統建立副本。
3. 使用範例程序 CSQ4AMSM 中的指示來修改程序以符合您的需求。您也可以使用 JCL 中符號參數，以容許在啟動程序時修改程序。
4. 使用 Language Environment[®] _CEE_ENVFILE 檔案，檢閱並選擇性地變更傳遞至 AMS 作業的參數。範例 *thlqual.SCSQPROC(CSQ40ENV)* 列出支援的參數。
5. 針對每一個 IBM MQ 佇列管理程式，重複步驟 1 到 4。

下一步

第 743 頁的『[設定 Advanced Message Security 起始作業使用者 ID](#)』

z/OS 設定 Advanced Message Security 起始作業使用者 ID

Advanced Message Security (AMS) 作業需要一個使用者 ID，可讓它稱為「UNIX 系統服務 (USS)」程序。

關於這項作業

此外，作業所代表的使用者也必須具有 UNIX UID (使用者 ID) 及 GID (群組 ID) 的適當定義，因此這些使用者稱為「UNIX 系統服務」使用者。如需定義 UNIX 系統服務 UID 及 GID 的相關資訊，請參閱 *z/OS: Security Server RACF Security Administrator's Guide*。

z/OS: UNIX 系統服務規劃 會比較傳統 UNIX 安全與 z/OS 安全。傳統 UNIX 安全與 z/OS 安全之間的主要差異在於核心服務支援兩個適當的專用權層次：UNIX 層次及 z/OS UNIX 層次。

視您安裝的安全原則而定，Advanced Message Security 作業可以使用超級使用者權限 (uid (0)) 來執行，或使用 RACF FACILITY 類別 BPX.DAEMON 和 BPX.SERVER 設定檔允許的 RACF 身分來執行，因為這項作業必須能夠採用其使用者的 RACF 身分。

如果使用後者方法，或您已啟動 BPX.DAEMON 或 BPX.SERVER 設定檔，Advanced Message Security 作業程式 (*thlqual.SCSQAUTH(CSQ0DSRV)*) 必須位於 RACF 程式控制程式庫中。

檢閱 *z/OS: UNIX 系統服務規劃*，以確保您瞭解傳統 UNIX 安全與 z/OS UNIX 安全之間的安全差異。這可讓您根據安裝的安全原則來管理 Advanced Message Security 作業，以部署及執行特許「UNIX 系統服務」程序。

供參考，對這項審查有用的出版品如下：

- *z/OS: UNIX 系統服務規劃*
- *z/OS: Security Server RACF Security Administrator's Guide*

註: 請小心選擇此作業的使用者 ID, 因為 Advanced Message Security 收件者憑證會載入與此使用者 ID 相關聯的金鑰環中。在 [z/OS 上使用憑證](#) 中會討論此考量。

這裡顯示的步驟說明如何設定 Advanced Message Security 已啟動作業使用者。這些步驟使用 RACF 指令作為範例。如果您使用不同的安全管理程式, 則應該使用對等指令。

註: 本節中的範例假設您已針對 RACF STARTED、FACILITY 及 SURROGAT 類別啟動通用設定檔指令處理程序, 並進行通用設定檔檢查。如需 RACF 如何處理通用設定檔的相關資訊, 請參閱 *z/OS: Security Server RACF Command Language Reference*。

程序

1. 將 Advanced Message Security 起始作業使用者定義為 RACF。本節中的範例使用使用者 ID WMQAMSM。

```
ADDUSER WMQAMSM NAME('AMS user') OMVS (UID(0)) DFLTGRP(group)
```

視您的安裝標準而定, 選取適當的預設「群組」。

註: 如果您不想授與 USS 超級使用者權限 (UID (0)), 則必須允許 Advanced Message Security 使用者 ID 存取 BPX.DAEMON 和 BPX.SERVER 機能類別設定檔:

```
PERMIT BPX.DAEMON CLASS(FACILITY) ID(WMQAMSM) ACCESS(READ)
```

且 Advanced Message Security 作業程式 (*thlqual.SCSQAUTH* (CSQODSRV)) 必須位於 RACF 程式控制程式庫中。

若要使 SCSQAUTH 程式庫受到程式控制, 您可以使用下列指令:

```
RALTER PROGRAM * ADDMEM('thlqual.SCSQAUTH'//NOPADCHK) -or-  
RALTER PROGRAM ** ADDMEM('thlqual.SCSQAUTH'//NOPADCHK)  
SETROPTS WHEN(PROGRAM) REFRESH
```

您還必須針對 Advanced Message Security 作業使用的國家語言程式庫 (*thlqual.SCSQANLx*) 啟用程式控制。

2. 判斷 RACF STARTED 類別是否為作用中。如果不是, 請啟動 RACF STARTED 類別:

```
SETROPTS CLASSACT(STARTED)
```

3. 指定您在步驟 1 中選取或建立的使用者 ID, 以定義 Advanced Message Security 作業的已啟動類別設定檔:

```
RDEFINE STARTED qmgrAMSM.* STDATA(USER(WMQAMSM))
```

其中 *qmgr* 是已啟動作業名稱的字首。例如, 已啟動作業可能命名為 CSQ1AMSM。在此情況下, 您會將 *qmgrAMSM.** 替換為 CSQ1AMSM.*。

AMS 起始作業必須命名為 *qmgrAMSM*。

4. 使用 **SETROPTS** RACF 指令來重新整理儲存體內 RACLISTed STARTED 類別設定檔:

```
SETROPTS RACLIST(STARTED) REFRESH
```

5. 在 IBM MQ 訊息的保護處理期間, Advanced Message Security 作業會暫時採用要求者的主機使用者 ID 身分。因此, 必須在 SURROGAT 類別中為每一個可以提出要求的使用者 ID 定義設定檔。

如果 RACF SURROGAT 類別處於作用中, 則定義單一通用設定檔可讓 Advanced Message Security 作業假設任何使用者的身分。如果 SURROGAT 類別不在作用中, 則會忽略此檢查。*z/OS: UNIX* 系統服務規劃中說明了所需的 SURROGAT 設定檔。

若要在 SURROGAT 類別中定義設定檔, 請執行下列動作:

- a) 使用 RACF SETROPTS 指令啟動 RACF SURROGAT 類別:

```
SETROPTS CLASSACT(SURROGAT)
```

- b) 啟動 RACF SURROGAT 類別的通用設定檔處理:

```
SETROPTS GENERIC(SURROGAT)
```

- c) 啟動 RACF SURROGAT 類別的通用設定檔指令處理程序:

```
SETROPTS GENCMD(SURROGAT)
```

- d) 在 SURROGAT 類別中定義通用設定檔:

```
RDEFINE SURROGAT BPX.SRV.* UACC(NONE)
```

- e) 允許 Advanced Message Security 使用者 ID 存取通用 SURROGAT 類別設定檔:

```
PERMIT BPX.SRV.* CLASS(SURROGAT) ID(WMQAMSM) ACCESS(READ)
```

註: 如果您想要限制特定使用者由 Advanced Message Security 作業處理, 您可以定義更具體的設定檔, 如 z/OS: UNIX 系統服務規劃中所述。

例如, 稱為 BPX.SRV.MQUSER1 的設定檔會控制 AMS 作業是否可以採用使用者 ID MQUSER1 的身分。

- f) 允許 Advanced Message Security 使用者 ID 使用 BPX.SERVER 機能 (如果尚未在 [建立憑證和金鑰環](#) 中完成的話):

```
PERMIT BPX.SERVER CLASS(FACILITY) ID(WMQAMSM) ACCESS(READ)
```

- g) 使用 **SETROPTS RACF** 指令來重新整理儲存體內 RACLISTed 已啟動類別設定檔:

```
SETROPTS RACLIST(SURROGAT) REFRESH  
SETROPTS RACLIST(FACILITY) REFRESH
```

6. Advanced Message Security 作業使用 z/OS System SSL 服務所提供的機能, 來開啟 SAF 管理的金鑰環。存取金鑰環內容的基礎「系統授權機能 (SAF)」是由 RACF 或對等的安全管理程式所控制。

此服務是 IRRSDL00 (R_datalib) 可呼叫服務。這個可呼叫服務受到相同設定檔的保護, 這些設定檔用來保護定義給 RACF FACILITY 類別的 RACF RACDCERT 指令。因此, 必須允許 Advanced Message Security 使用者 ID 使用下列指令來存取設定檔:

- a) 如果您尚未這樣做, 請對保護 RACDCERT 指令及 IRRSDL00 可呼叫服務的 RACF FACILITY 類別定義 RACF 通用設定檔:

```
RDEFINE FACILITY IRR.DIGTCERT.* UACC(NONE)  
SETROPTS RACLIST(FACILITY) REFRESH
```

- b) 授與已啟動作業使用者 ID 對 RACF 通用設定檔的權限:

```
PERMIT IRR.DIGTCERT.* CLASS(FACILITY) ID(WMQAMSM) ACC(READ)
```

或者, 您可以授與 RDATA LIB 類別中資料服務作業使用者金鑰環的 READ 存取權, 如下所示:

```
PERMIT WMQASMD.DRQ.AMS.KEYRING.LST CLASS(RDATA LIB) ID(WMQAMSM) ACC(READ)
```

7. 配置資源安全:

- a) Advanced Message Security 已啟動作業使用者需要權限，才能以批次應用程式連接至佇列管理程式。

如果您的佇列管理程式已啟用連線安全，請使用下列指令授與 AMS 作業權限來連接至佇列管理程式：

```
PERMIT hlq.BATCH CLASS(MQCONN) ID(WMQAMSM) ACC(READ)
```

其中 *hlq* 可以是佇列管理程式名稱佇列共用群組名稱。

如需進一步資訊，請參閱 [批次連線的連線安全設定檔](#)。

- b) Advanced Message Security 啟動型作業使用者需要有瀏覽 SYSTEM.PROTECTION.POLICY.QUEUE。如果佇列管理程式上的佇列安全處於作用中狀態，請使用下列指令授與 AMS 使用者存取佇列的權限：

```
RDEFINE MQQUEUE hlq.SYSTEM.PROTECTION.POLICY.QUEUE UACC(NONE)  
PERMIT hlq.SYSTEM.PROTECTION.POLICY.QUEUE CLASS(MQQUEUE) ID(WMQAMSM) ACCESS(READ)
```

其中 *hlq* 可以是佇列管理程式名稱佇列共用群組名稱。

如果佇列管理程式使用大小寫混合的設定檔，請改為在 MXQUEUE 類別中定義設定檔。

若要使用 CSQOUTIL 公用程式來管理 AMS 安全原則，管理者需要存取權，才能將訊息放置到 SYSTEM.PROTECTION.POLICY.QUEUE。這是透過授與 UPDATE 存取權給保護佇列的設定檔來執行。

如需進一步資訊，請參閱 [佇列安全的設定檔](#)。

下一步

[第 746 頁的『將 RACDCERT 許可權授與 Advanced Message Security 的安全管理者』](#)

將 RACDCERT 許可權授與 Advanced Message Security 的安全管理者

您的 Advanced Message Security 安全管理者需要權限，才能使用 RACDCERT 指令來建立及管理數位憑證。

程序

- 識別這個角色的適當使用者 ID，並授與使用 RACDCERT 指令的許可權。例如：

```
PERMIT IRR.DIGTCERT.* CLASS(FACILITY) ID(admin) ACCESS(CONTROL)  
SETROPTS RACLIST(FACILITY) REFRESH
```

其中 *admin* 是 Advanced Message Security 安全管理者的使用者 ID。

下一步

[第 746 頁的『授與使用者 Advanced Message Security 資源許可權』](#)

授與使用者 Advanced Message Security 資源許可權

Advanced Message Security 使用者需要相關資源許可權。

關於這項作業

Advanced Message Security 使用者 (即放置或取得 Advanced Message Security 受保護訊息的使用者) 需要：

- 與其使用者 ID 相關聯的 OMVS 區段
- IRR.DIGTCERT.LISTRING 或 RDATA LIB
- ISF 類別 CSFSERV 和 CSFKEYS 設定檔的許可權
- 可放入 SYSTEM.PROTECTION.ERROR.QUEUE

Advanced Message Security 作業會暫時採用其用戶端的身分；亦即，在處理 Advanced Message Security 所保護佇列的 IBM MQ 訊息期間，該作業會充當 Advanced Message Security 之使用者的 z/OS 使用者 ID 的代理。

為了讓作業採用使用者的 z/OS 身分，用戶端 z/OS 使用者 ID 必須具有與其使用者設定檔相關聯的已定義 OMVS 區段。

作為管理輔助工具，RACF 可讓您定義可與 RACF 使用者和群組設定檔相關聯的預設 OMVS 區段。如果 z/OS 使用者 ID 或群組設定檔未明確定義 OMVS 區段，則會使用此預設值。如果您計劃讓大量使用者使用 Advanced Message Security，則可以選擇使用此預設值，而不是明確定義每一個使用者的 OMVS 區段。

z/OS: Security Server RACF Security Administrator 's Guide 包含定義預設 OMVS 區段的詳細程序。檢閱本出版物中概述的程序，以判定 RACF 使用者和群組設定檔中預設 OMVS 區段的定義是否適用於您的安裝。

程序

1. 授與 IRR.DIGTCERT.LISTRING 設定檔:

- 授與 IRR.DIGTCERT.LISTRING 設定檔，發出下列指令:

```
RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(READ)
```

- 授與 IRR.DIGTCERT.LISTRING 設定檔，發出下列指令:

```
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(userid) ACCESS(READ)
```

其中 `userid` 是 Advanced Message Security 使用者的名稱。

- 或者，使用 RDATALIB 類別來授與特定金鑰環的存取權。RDATALIB 許可權優先於 IRR.DIGTCERT.LISTRING 許可權。例如:

```
PERMIT user.DRQ.AMS.KEYRING.LST CLASS(RDATALIB) ID(user) ACC(READ)
```

2. 如果您使用 ICSF 管理的憑證和私密金鑰，Advanced Message Security 使用者需要存取某些類別 CSFSERV 和 CSFKEYS 設定檔。下表詳述此存取權:

類別	設定檔	許可權
CSFSERV	CSFDSG	READ
CSFSERV	CSFPKE	READ
CSFSERV	CSFPKD	READ
CSFSERV	CSFDSV	READ
CSFKEYS	ICSF PKDS 標籤	READ

3. 在已定義 AMS 原則的佇列上執行作業的應用程式需要存取權，才能將訊息放置到 SYSTEM.PROTECTION.ERROR.QUEUE。使用下列指令授與佇列的放置存取權:

```
RDEFINE MQQUEUE hlq.SYSTEM.PROTECTION.ERROR.QUEUE UACC(NONE)  
PERMIT hlq.SYSTEM.PROTECTION.ERROR.QUEUE CLASS(MQQUEUE) ID(userId) ACCESS(UPDATE)
```

其中 `hlq` 可以是佇列管理程式名稱佇列共用群組名稱，而 `userId` 是應用程式使用者 ID。

下一步

第 747 頁的『[建立 Advanced Message Security 的金鑰環](#)』

建立 Advanced Message Security 的金鑰環

Advanced Message Security (AMS) 用於簽署和加密的憑證儲存在 z/OS SAF 金鑰環中。您需要先建立這些金鑰環和憑證，才能使用 AMS。

關於這項作業

Advanced Message Security 會存取下列金鑰環中的憑證:

- AMS 位址空間使用者所擁有的單一金鑰環。
- 在已定義 AMS 原則的佇列上傳送或接收訊息的個別使用者所擁有的金鑰環。

這些金鑰環都必須命名為 `drq.ams.keyring`。

AMS 所使用的金鑰環和憑證的相關資訊，以及在 [z/OS 上使用憑證](#) 中的範例實務。

請遵循下列步驟來建立 AMS 所需的金鑰環，並將憑證連接至金鑰環。在啟動 AMS 之前，您必須先建立 AMS 位址空間使用者所擁有的金鑰環。您可以建立隨時傳送或接收訊息的使用者所擁有的金鑰環。

程序

1. 發出下列指令，以建立 AMS 位址空間使用者所擁有的金鑰環:

```
RACDCERT ID(amsUser) ADDRING(drq.ams.keyring)
```

其中 *amsUser* 是 AMS 位址空間的使用者 ID。

2. 針對每一個使用者 ID 發出步驟 1 中的指令，為傳送或接收 AMS 所保護的訊息的每一個使用者建立金鑰環。
3. 將使用者憑證發證者的憑證管理中心 (CA) 憑證連接至 AMS 位址空間使用者 ID 所擁有的金鑰環。發出下列指令:

```
RACDCERT ID(amsUser) CONNECT(CERTAUTH LABEL('caLabel') RING(drq.ams.keyring))
```

其中 *amsUser* 是 AMS 位址空間的使用者 ID，而 *caLabel* 是 CA 憑證的標籤。

如果您使用 RACF 作為 CA，且需要建立憑證管理中心憑證，請遵循 [定義本端憑證管理中心憑證](#) 中的範例。

4. 如果您使用隱私權或機密性安全原則來加密 AMS 所保護佇列上的訊息，請將訊息收件者的憑證連接至 AMS 位址空間使用者 ID 所擁有的金鑰環。發出下列指令:

```
RACDCERT ID(amsUser) CONNECT(ID(userId) LABEL('certLabel')  
RING(drq.ams.keyring) USAGE(SITE))
```

其中 *amsUser* 是 AMS 位址空間的使用者 ID，*userId* 是訊息收件者，而 *certLabel* 是使用者憑證的標籤。

USAGE(SITE) 屬性會阻止在金鑰環中存取私密金鑰。

如果您要使用 RACF 建立自己的憑證，請遵循 [使用私密金鑰建立數位憑證](#) 中的範例來建立憑證。

5. 將傳送或接收 AMS 所保護訊息的每一個使用者的憑證，連接至使用者所擁有的金鑰環。憑證必須連接成金鑰環中的預設憑證。發出下列指令:

```
RACDCERT ID(userId) CONNECT(ID(userId) LABEL('certLabel')  
RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

其中 *userId* 是傳送或接收訊息的使用者，而 *certLabel* 是使用者憑證的標籤。

下一步

第 748 頁的 [『啟用 Advanced Message Security』](#)

啟用 Advanced Message Security

佇列管理程式的安全原則功能由系統參數模組中的 SPLCAP 參數控制。

關於這項作業

請遵循下列步驟，針對單一佇列管理程式啟用 Advanced Message Security (AMS)。

這項作業需要您變更系統參數模組。如需建立及自訂系統參數模組的相關資訊，請參閱 [第 715 頁的『自訂系統參數模組』](#)。

程序

1. 在 CSQ6SYSP 中，將 **SPLCAP** 設為 YES。如需 CSQ6SYSP 巨集的相關資訊，請參閱 [第 716 頁的『使用 CSQ6SYSP』](#)。
2. **V9.1.3**
視您的授權而定，將 **AMSPROD** 設為 AMS、ADVANCED 或 ADVANCEDVUE。如需 CSQ6USGP 巨集的相關資訊，請參閱 [使用 CSQ6USGP](#)。
3. 重新編譯系統參數模組。
4. 以更新的系統參數模組重新啟動佇列管理程式。當佇列管理程式啟動時，會自動啟動 AMS 位址空間。

z/OS V9.1.0 配置 mqweb 伺服器

請使用這些主題作為逐步手冊，以配置 mqweb 伺服器。

相關工作

[第 640 頁的『配置 IBM MQ Console 和 REST API』](#)

管理 IBM MQ Console 和 REST API 的 mqweb 伺服器隨附了預設配置。為了使用其中任一元件，需要完成一些配置作業，例如配置安全以容許使用者登入。本主題說明所有可用的配置選項。

z/OS V9.1.0 建立 mqweb 伺服器

如果您已安裝 IBM MQ for z/OS UNIX 系統服務 Web 元件，並且想要使用 MQ Console 或 REST API，則需要建立並自訂 mqweb 伺服器。

開始之前

您需要建立 SYSTEM.REST.REPLY.QUEUE，以便使用 Liberty 伺服器。請使用 [第 710 頁的『自訂起始設定輸入資料集』](#) 中的最新 **CSQ4INSG** 範例來執行此動作。



小心：啟動 mqweb 伺服器時，如果您遇到錯誤訊息 CWWKG0014E，如下列輸出中所示：

```
Launching mqweb (MQM MVS/ESA V9 R2.0/wlp...) (en_US)
YAUDIT   " CWWKE0001I: The server mqweb has been
launched.
          YWARNING " CWWKF0009W: The server has not been configured to install any
features.
          YAUDIT   " CWWKF0011I: The mqweb server is ready to run a smarter planet.
The mqweb server started in 6.348 seconds.
          YERROR   " CWWKG0014E: The configuration parser detected an XML syntax
error while parsing the root of the configuration and the referenced configuration
documents.
                                     Error: An invalid XML character (Unicode: 0x4c) was found
in the prolog of the document.
                                     File: file:<your filepath>/servers/mqweb/server.xml Line:
1 Column: 1
```

您應該檢查 AUTOCVT 的 z/OS 設定 (自動將檔案從一個字碼集轉換成另一個字碼集)，並執行下列其中一項來根據需要調整值。

在 USS 終端機中：

發出指令：`echo $_BPXK_AUTOCVT` 以顯示此環境變數的值。如果未定義環境變數，則不會顯示任何值。

如果要設定環境變數，請參閱 [_BPXK 環境變數](#)。

全系統：

顯示 [z/OS UNIX System Services \(OMVS\) 的狀態](#) 中的範例 6 顯示如何在 BPXPRMxx 中顯示全系統 AUTOCVT 陳述式的值。

若要設定整個系統的環境變數，請在 BPXPRMxx 中使用 [AUTOCVT 陳述式](#)。

如果在 USS 終端機中設定環境變數 `_BPXX_AUTOCVT`，它會置換 BPXPRMxx 中陳述式 AUTOCVT 的系統範圍設定。

關於這項作業

- 您需要針對要執行 MQ Console 或 REST API 的每一個 z/OS 系統執行一次此作業。
- 執行中的每一個 IBM MQ 版本都需要 mqweb 伺服器。
- 從舊版移轉時，您可能需要重新整理或修改伺服器配置。

MQ Console 和 REST API 需要建立單一 Liberty 伺服器，稱為 mqweb。

伺服器配置和日誌檔全都儲存在 Liberty 使用者目錄下。



小心: 如需 RDEFINE 範例的部分範例，請參閱 RACF 設定檔，並在 [用來控制 IBM MQ 資源存取的設定檔](#) 一節中，參閱 [名稱清單的設定檔](#)，以取得在伺服器上設定必要安全的相關資訊。

請執行下列程序來建立 mqweb 伺服器：

程序

1. 選擇 Liberty 使用者目錄的適當位置。

mqweb 伺服器執行所在的使用者 ID，需要此使用者目錄及其內容的讀取及寫入權。因為此使用者目錄將包含日誌檔及伺服器配置，所以您應該在個別檔案系統中建立此目錄。

註: mqweb 伺服器啟動時有大量磁碟 I/O。若要減少啟動 mqweb 伺服器所花費的時間，請確保 IBM MQ 安裝 z/OS UNIX 檔案系統及 Liberty 使用者目錄檔案系統都是 Sysplex-aware 或本端裝載在 mqweb 伺服器執行所在的系統上。

2. 在 UNIX 系統服務中，發出下列指令，確定現行目錄是 PathPrefix/web/bin:

```
cd PathPrefix/web/bin
```

PathPrefix 是 IBM MQ UNIX 系統服務元件安裝路徑。

3. 透過執行 **crtmqweb** Script，建立包含範本 mqweb 伺服器定義的 Liberty 使用者目錄。例如，若要在 /var/mqm/mqweb 中建立 Liberty 使用者目錄，請發出下列指令：

```
./crtmqweb /var/mqm/mqweb
```

註: **crtmqweb** Script 接受一個選用參數- Liberty 使用者目錄的名稱。

如果您未提供 Liberty 使用者目錄的名稱，則會使用預設值 /var/mqm/web/installation1。

4. 使用下列指令，變更 Liberty 使用者目錄中目錄及檔案的所有權，讓它們屬於 mqweb 伺服器執行所在的使用者 ID 及群組：

```
chown -R userid:group path
```

若要授與群組對路徑的寫入權，請發出下列指令：

```
chmod -R 770 path
```

下一步

第 750 頁的『[建立 mqweb 伺服器的程序](#)』

相關工作

第 640 頁的『[配置 IBM MQ Console 和 REST API](#)』

管理 IBM MQ Console 和 REST API 的 mqweb 伺服器隨附了預設配置。為了使用其中任一元件，需要完成一些配置作業，例如配置安全以容許使用者登入。本主題說明所有可用的配置選項。

建立 mqweb 伺服器的程序

如果您已安裝 IBM MQ for z/OS Unix 系統服務 Web 元件，並且想要使用 MQ Console 或 REST API，則需要建立編目程序來啟動 mqweb 伺服器。mqweb 伺服器是管理 MQ Console 和 REST API 的 Liberty 伺服器。

- 您需要針對要執行 MQ Console 或 REST API 的每一個 z/OS 系統執行一次此作業。

- 執行中的每一個 IBM MQ 版本都需要 mqweb 伺服器。例如，對於位於 IBM MQ 9.1.0 的佇列管理程式，稱為 MQWB0910 的啟動作業，以及對於位於 IBM MQ 9.0.5 的佇列管理程式，稱為 MQWB0905 的啟動作業。

如果您在 z/OS 系統上只有一個佇列管理程式，則可以執行單一 Liberty 伺服器啟動作業，並在移轉佇列管理程式時變更它使用的檔案庫。

- 從舊版移轉時，您可能需要修改已編目程序。

請執行下列程序來建立已編目的程序：

1. 將範例啟動型作業程序 th1qua1.SCSQPROC (CSQ4WEBS) 複製到您的程序檔案庫。

根據您企業的標準來命名程序。

例如 MQWB0910，指出這是 IBM MQ 9.1.0 mqweb 伺服器的已編目程序。

2. 使用範例程序 CSQ4WEBS 中的指示，根據您的需求自訂程序。

請注意，Liberty 使用者目錄是執行 **crtmqweb** Script 以建立 mqweb 伺服器定義時指定的目錄。

如需詳細資料，請參閱第 749 頁的『建立 mqweb 伺服器』。

註：請確保在編輯成員時指定 **關閉大寫**，因為檔案具有小寫資料。

3. 授權程序在外部安全管理程式下執行。

4. 使用 IBM Workload Manager (WLM) 來分類此位址空間。

mqweb 伺服器是 IBM MQ 應用程式，使用者與此應用程式互動。應用程式在 WLM 中不需要具有高重要性，且 **STCUSER** 的服務類別可能適合。

下一步

遵循第 640 頁的『mqweb 伺服器的基本配置』中的步驟，以完成 mqweb 伺服器的配置。

相關工作

第 640 頁的『配置 IBM MQ Console 和 REST API』

管理 IBM MQ Console 和 REST API 的 mqweb 伺服器隨附了預設配置。為了使用其中任一元件，需要完成一些配置作業，例如配置安全以容許使用者登入。本主題說明所有可用的配置選項。

z/OS 在 z/OS 上測試佇列管理程式

當您已自訂或移轉佇列管理程式時，您可以執行安裝驗證程式及 IBM MQ for z/OS 隨附的部分範例應用程式來測試它。

關於這項作業

安裝並自訂 IBM MQ for z/OS 之後，您可以使用所提供的安裝驗證程式 CSQ4IVP1 來確認 IBM MQ for z/OS 可運作。

基本安裝驗證程式 CSQ4IVP1 會測試非共用佇列，並驗證基本 IBM MQ，而不使用 C、COBOL 或 CICS 範例。

執行基本安裝驗證之後，您可以使用具有不同佇列的 CSQ4IVP1 來測試共用佇列，也可以測試 Db2 及連結機能是否已正確設定。若要確認分散式佇列作業正常運作，您可以使用所提供的安裝驗證程式 CSQ4IVPX，

CSQ4IVP1 提供作為載入模組，並提供一組程序化範例應用程式作為來源模組，示範「訊息佇列介面 (MQI)」的一般用法。您可以使用這些來源模組來測試不同的程式設計語言環境。您可以使用所提供的範例 JCL，來編譯及鏈結編輯適用於您安裝的任何其他範例。

程序

- 如需如何在 z/OS 上測試佇列管理程式的相關資訊，請參閱下列子主題：

- 第 752 頁的『執行基本安裝驗證程式』
- 第 755 頁的『測試佇列共用群組』

- [第 756 頁的『分散式佇列測試』](#)
- [第 758 頁的『使用 IBM MQ for z/OS 來測試 C、C++、COBOL、PL/I 及 CICS 程式』](#)

相關概念

[IBM MQ for z/OS 概念](#)

相關工作

[在 z/OS 上規劃 IBM MQ 環境](#)

[第 693 頁的『在 z/OS 上配置佇列管理程式』](#)

使用這些指示，在「IBM MQ for z/OS」上配置佇列管理程式。

[管理 IBM MQ for z/OS](#)

執行基本安裝驗證程式

安裝並自訂 IBM MQ 之後，您可以使用所提供的安裝驗證程式 CSQ4IVP1 來確認 IBM MQ 可運作。

基本安裝驗證程式是批次組譯器 IVP，可驗證基本 IBM MQ，而不使用 C、COBOL 或 CICS 範例。

「批次組合器 IVP」是由 SMP/E 所鏈結編輯，載入模組隨附在程式庫 thlqual.SCSQLOAD 中。

完成 SMP/E APPLY 步驟和自訂步驟之後，請執行「批次組合器 IVP」。

如需進一步詳細資料，請參閱下列各節：

- [CSQ4IVP1 應用程式概觀](#)
- [準備執行 CSQ4IVP1](#)
- [執行中 CSQ4IVP1](#)
- [檢查 CSQ4IVP1 的結果](#)

CSQ4IVP1 應用程式概觀

CSQ4IVP1 是一個批次應用程式，可連接至 IBM MQ 子系統並執行下列基本功能：

- 問題 IBM MQ 呼叫
- 與指令伺服器通訊
- 驗證觸發是否處於作用中
- 產生並刪除動態佇列
- 驗證訊息期限處理
- 驗證訊息確定處理

準備執行 CSQ4IVP1

在執行 CSQ4IVP1 之前：

1. 檢查 IVP 項目是否位於佇列管理程式啟動程式中的 CSQINP2 資料集連結中。IVP 項目在成員 thlqual.SCSQPROC(CSQ4IVPQ) 中提供。否則，請將 thlqual.SCSQPROC(CSQ4IVPQ) 中提供的定義新增至 CSQINP2 連結。如果佇列管理程式目前正在執行中，您需要重新啟動它，這些定義才會生效。
2. 執行安裝驗證程式所需的範例 JCL CSQ4IVPR 位於程式庫 thlqual.SCSQPROC 中。
使用 IBM MQ 程式庫的高階限定元、您要使用的國家語言、四個字元的 IBM MQ 佇列管理程式名稱，以及工作輸出的目的地，來自訂 CSQ4IVPR JCL。
3. 如果 IBM MQ 安全處於作用中狀態，則更新 RACF 以容許 CSQ4IVP1 存取其資源。

若要在啟用 IBM MQ 安全時執行 CSQ4IVP1，您需要具有存取物件權限的 RACF 使用者 ID。如需將資源定義給 RACF 的詳細資料，請參閱 [在 z/OS 上設定安全](#)。執行 IVP 的使用者 ID 必須具有下列存取權：

權限管理中心	設定檔	類別
READ	ssid.DISPLAY.PROCESS	MQCMDS
UPDATE	ssid.SYSTEM.COMMAND.INPUT	MQQUEUE
UPDATE	ssid.SYSTEM.COMMAND.REPLY.MODEL	MQQUEUE
UPDATE	ssid.CSQ4IVP1.**	MQQUEUE
READ	ssid.BATCH	MQCONN

這些需求假設所有 IBM MQ 安全都在作用中。第 753 頁的圖 100 顯示用來啟動 IBM MQ 安全的 RACF 指令。此範例假設佇列管理程式名稱為 CSQ1，且執行範例 CSQ4IVP1 的人員使用者 ID 為 TS101。

```
RDEFINE MQCMDS CSQ1.DISPLAY.PROCESS
PERMIT CSQ1.DISPLAY.PROCESS CLASS(MQCMDS) ID(TS101) ACCESS(READ)

RDEFINE MQQUEUE CSQ1.SYSTEM.COMMAND.INPUT
PERMIT CSQ1.SYSTEM.COMMAND.INPUT CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.COMMAND.REPLY.MODEL
PERMIT CSQ1.SYSTEM.COMMAND.REPLY.MODEL CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.CSQ4IVP1.**
PERMIT CSQ1.CSQ4IVP1.** CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQCONN CSQ1.BATCH
PERMIT CSQ1.BATCH CLASS(MQCONN) ID(TS101) ACCESS(READ)
```

圖 100: CSQ4IVP1 的 RACF 指令

執行 CSQ4IVP1

完成這些步驟之後，請啟動佇列管理程式。如果佇列管理程式已在執行中，且您已變更 CSQINP2，則必須停止佇列管理程式並重新啟動它。

IVP 以批次工作方式執行。自訂工作卡以符合安裝的提交需求。

檢查 CSQ4IVP1 的結果

IVP 分成 10 個階段；在執行下一個階段之前，每一個階段都必須完成零完成碼。IVP 會產生報告，並列出：

- 正在連接的佇列管理程式名稱。
- 一行訊息，顯示每一個階段所傳回的完成碼和原因碼。
- 適當的一行參考訊息。

第 755 頁的圖 101 中提供了範例報告

z/OS 如需完成碼和原因碼的說明，請參閱 [IBM MQ for z/OS 訊息、完成碼和原因碼](#)。

部分階段有多個 IBM MQ 呼叫，如果失敗，則會發出一則訊息，指出傳回失敗的特定 IBM MQ 呼叫。此外，對於某些階段，IVP 會將說明及診斷資訊放入註解欄位中。

IVP 工作要求特定佇列管理程式物件的專用控制，因此應該透過系統進行單一執行緒。不過，IVP 可以針對佇列管理程式執行的次數沒有限制。

每個階段所執行的功能如下：

階段 1

發出 MQCONN API 呼叫來連接至佇列管理程式。

階段 2

決定指令伺服器用來擷取要求訊息的系統指令輸入佇列名稱。此佇列接收來自階段 5 的顯示要求。

若要這麼做，呼叫順序如下：

1. 發出 MQOPEN 呼叫並指定佇列管理程式名稱，以開啟佇列管理程式物件。
2. 發出 MQINQ 呼叫，以找出系統指令輸入佇列的名稱。
3. 發出 MQINQ 呼叫，以瞭解各種佇列管理程式事件切換的相關資訊。
4. 發出 MQCLOSE 呼叫以關閉佇列管理程式物件。

順利完成此階段時，系統指令輸入佇列的名稱會顯示在註解欄位中。

階段 3

使用 **MQOPEN** 呼叫開啟起始佇列。

在此階段會開啟此佇列，因為預期會有觸發訊息，而此訊息是由於指令伺服器回覆來自階段 5 的要求而到達。必須開啟佇列以供輸入，以符合觸發準則。

階段 4

使用 CSQ4IVP1.MODEL 佇列作為模型。動態佇列具有與從中建立它的模型相同的屬性。這表示當階段 5 中指令伺服器要求的回覆寫入此佇列時，觸發訊息會寫入階段 3 中開啟的起始佇列。

順利完成此階段時，會在註解欄位中指出永久動態佇列的名稱。

階段 5

向指令伺服器指令佇列發出 MQPUT1 要求。

類型為 MQMT_REQUEST 的訊息會寫入系統指令輸入佇列，要求顯示處理程序 CSQ4IVP1。訊息的訊息描述子指定在「階段 4」中建立的永久動態佇列，作為指令伺服器回應的回覆目的地佇列。

階段 6

從起始佇列發出 **MQGET** 要求。在此階段，會針對在「階段 3」中開啟的起始佇列發出 GET WAIT，間隔為 1 分鐘。傳回的訊息預期是由指令伺服器的回應訊息寫入回覆目的地佇列所產生的觸發訊息。

第 7 階段

刪除在階段 4 中建立的永久動態佇列。因為佇列上仍有訊息，所以會使用 MQCO_PURGE_DELETE 選項。

階段 8

1. 開啟動態佇列。
2. MQPUT 已設定期限間隔的訊息。
3. 等待訊息到期。
4. 嘗試 MQGET 過期訊息。
5. MQCLOSE 佇列。

第 9 階段

1. 開啟動態佇列。
2. MQPUT 訊息。
3. 發出 MQCMIT 以確定現行工作單元。
4. MQGET 訊息。
5. 發出 MQBACK 以取消訊息。
6. MQGET 相同的訊息，並確定取消計數設為 1。
7. 發出 MQCLOSE 以關閉佇列。

階段 10

使用「**MQDISC**」中斷與佇列管理程式的連線。

執行 IVP 之後，您可以刪除不再需要的任何物件。

如果 IVP 未順利執行，請手動嘗試每一個步驟，以找出哪個功能失敗。


```
RDEFINE MQQUEUE CSQ1.CSQ4IVPG.**
PERMIT CSQ1.CSQ4IVPG.** CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)
```

圖 102: 佇列共用群組 CSQ4IVP1 的 RACF 指令

針對佇列共用群組執行 CSQ4IVP1

完成這些步驟之後，請啟動佇列管理程式。如果佇列管理程式已在執行中，且您已變更 CSQINP2，則必須停止佇列管理程式並重新啟動它。

IVP 以批次工作方式執行。自訂工作卡以符合安裝的提交需求。

檢查佇列共用群組的 CSQ4IVP1 結果

佇列共用群組的 IVP 與基本 IVP 的運作方式相同，不同之處在於所建立的佇列稱為 CSQIVPG。xx。遵循第 753 頁的『檢查 CSQ4IVP1 的結果』中提供的指示，檢查佇列共用群組的 IVP 結果。

分散式佇列測試

您可以使用所提供的安裝驗證程式 CSQ4IVPX 來確認分散式佇列作業可運作。

CSQ4IVPX 工作概觀

CSQ4IVPX 是啟動通道起始程式並發出 IBM MQ DISPLAY CHINIT 指令的批次工作。這會驗證分散式佇列的所有主要層面都可運作，同時避免需要設定通道及網路定義。

準備執行 CSQ4IVPX

在執行 CSQ4IVPX: 之前:

- 執行安裝驗證程式所需的範例 JCL CSQ4IVPX 位於程式庫 thlqual.SCSQPROC 中。
使用 IBM MQ 程式庫的高階限定元、您要使用的國家語言、四個字元的佇列管理程式名稱及工作輸出的目的地，來自訂 CSQ4IVPX JCL。
- 更新 RACF，以容許 CSQ4IVPX 在 IBM MQ 安全處於作用中狀態時存取其資源。若要在啟用 IBM MQ 安全時執行 CSQ4IVPX，您需要具有存取物件權限的 RACF 使用者 ID。如需將資源定義給 RACF 的詳細資料，請參閱在 z/OS 上設定安全。執行 IVP 的使用者 ID 必須具有下列存取權：

權限管理中心	設定檔	類別
CONTROL	ssid.START.CHINIT 和 ssid.STOP.CHINIT	MQCMDS
UPDATE	ssid.SYSTEM.COMMAND.INPUT	MQQUEUE
UPDATE	ssid.SYSTEM.CSQUTIL.*	MQQUEUE
READ	ssid.BATCH	MQCONN
READ	ssid.DISPLAY.CHINIT	MQCMDS

這些需求假設已定義連線安全設定檔 ssid.CHIN (如 通道起始程式的連線安全設定檔 中所示)，且所有 IBM MQ 安全都在作用中。執行此動作的 RACF 指令顯示在 第 757 頁的圖 103 中。此範例假設：

- 佇列管理程式名稱為 CSQ1
 - 執行範例 CSQ4IVPX 之人員的使用者 ID 為 TS101
 - 通道起始程式位址空間正在使用者 ID CSQ1MSTR 下執行
- 更新 RACF，以容許通道起始程式位址空間具有下列存取權：

權限管理中心	設定檔	類別
READ	ssid.CHIN	MQCONN
UPDATE	ssid.SYSTEM.COMMAND.INPUT	MQQUEUE
UPDATE	ssid.SYSTEM.CHANNEL.INITQ	MQQUEUE
UPDATE	ssid.SYSTEM.CHANNEL.SYNCQ	MQQUEUE
ALTER	ssid.SYSTEM.CLUSTER.COMMAND.QUEUE	MQQUEUE
UPDATE	ssid.SYSTEM.CLUSTER.TRANSMIT.QUEUE	MQQUEUE
ALTER	ssid.SYSTEM.CLUSTER.REPOSITORY.QUEUE	MQQUEUE
CONTROL	ssid.CONTEXT.**	MQADMIN

要執行此動作的 RACF 指令也會顯示在 [第 757 頁的圖 103](#) 中。

```

RDEFINE MQCMDS CSQ1.DISPLAY.DQM
PERMIT CSQ1.DISPLAY.DQM CLASS(MQCMDS) ID(TS101) ACCESS(READ)

RDEFINE MQCMDS CSQ1.START.CHINIT
PERMIT CSQ1.START.CHINIT CLASS(MQCMDS) ID(TS101) ACCESS(CONTROL)

RDEFINE MQCMDS CSQ1.STOP.CHINIT
PERMIT CSQ1.STOP.CHINIT CLASS(MQCMDS) ID(TS101) ACCESS(CONTROL)

RDEFINE MQQUEUE CSQ1.SYSTEM.COMMAND.INPUT
PERMIT CSQ1.SYSTEM.COMMAND.INPUT CLASS(MQQUEUE) ID(TS101,CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.CSQUTIL.*
PERMIT CSQ1.SYSTEM.CSQUTIL.* CLASS(MQQUEUE) ID(TS101) ACCESS(UPDATE)

RDEFINE MQCONN CSQ1.BATCH
PERMIT CSQ1.BATCH CLASS(MQCONN) ID(TS101) ACCESS(READ)

RDEFINE MQCONN CSQ1.CHIN
PERMIT CSQ1.CHIN CLASS(MQCONN) ID(CSQ1MSTR) ACCESS(READ)

RDEFINE MQQUEUE CSQ1.SYSTEM.CHANNEL.SYNCQ
PERMIT CSQ1.SYSTEM.CHANNEL.SYNCQ CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.CLUSTER.COMMAND.QUEUE
PERMIT CSQ1.SYSTEM.CLUSTER.COMMAND.QUEUE CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(ALTER)

RDEFINE MQQUEUE CSQ1.SYSTEM.CLUSTER.TRANSMIT.QUEUE
PERMIT CSQ1.SYSTEM.CLUSTER.TRANSMIT.QUEUE CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQQUEUE CSQ1.SYSTEM.CLUSTER.REPOSITORY.QUEUE
PERMIT CSQ1.SYSTEM.CLUSTER.REPOSITORY.QUEUE CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(ALTER)

RDEFINE MQQUEUE CSQ1.SYSTEM.CHANNEL.INITQ
PERMIT CSQ1.SYSTEM.CHANNEL.INITQ CLASS(MQQUEUE) ID(CSQ1MSTR) ACCESS(UPDATE)

RDEFINE MQADMIN CSQ1.CONTEXT.**
PERMIT CSQ1.CONTEXT.** CLASS(MQADMIN) ID(CSQ1MSTR) ACCESS(CONTROL)

```

圖 103: RACF 適用於 CSQ4IVPX 的指令

執行 CSQ4IVPX

完成這些步驟之後，請啟動佇列管理程式。

IVP 以批次工作方式執行。自訂工作卡以符合安裝的提交需求。

檢查 CSQ4IVPX 的結果

CSQ4IVPX 會執行 CSQUTIL IBM MQ 公用程式，以發出三個 MQSC 指令。SYSPRINT 輸出資料集應該類似第 758 頁的圖 104，但根據您的佇列管理程式屬性，詳細資料可能會不同。

- 您應該會看到指令 **(1)**，每個指令後面接著數個訊息。
- 每一個指令的最後一則訊息應該是 "CSQ9022I ... 正常完成" **(2)**。
- 整個工作應該完成，回覆碼為零 **(3)**。

```
CSQU000I CSQUTIL IBM MQ for z/OS - V6
CSQU001I CSQUTIL Queue Manager Utility - 2005-05-09 09:06:48
COMMAND
CSQU127I CSQUTIL Executing COMMAND using input from CSQUCMD data set
CSQU120I CSQUTIL Connecting to queue manager CSQ1
CSQU121I CSQUTIL Connected to queue manager CSQ1
CSQU055I CSQUTIL Target queue manager is CSQ1
START CHINIT
(1)
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQM138I +CSQ1 CSQMSCHI CHANNEL INITIATOR STARTING
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQ9022I +CSQ1 CSQXCRPS ' START CHINIT' NORMAL COMPLETION
(2)
DISPLAY CHINIT
(1)
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQM137I +CSQ1 CSQMDQM DISPLAY CHINIT COMMAND ACCEPTED
CSQN205I COUNT= 12, RETURN=00000000, REASON=00000000
CSQX830I +CSQ1 CSQXRQM Channel initiator active
CSQX002I +CSQ1 CSQXRQM Queue sharing group is QSG1
CSQX831I +CSQ1 CSQXRQM 8 adapter subtasks started, 8 requested
CSQX832I +CSQ1 CSQXRQM 5 dispatchers started, 5 requested
CSQX833I +CSQ1 CSQXRQM 0 SSL server subtasks started, 0 requested
CSQX840I +CSQ1 CSQXRQM 0 channel connections current, maximum 200
CSQX841I +CSQ1 CSQXRQM 0 channel connections active, maximum 200,
including 0 paused
CSQX842I +CSQ1 CSQXRQM 0 channel connections starting,
0 stopped, 0 retrying
CSQX836I +CSQ1 Maximum channels - TCP/IP 200, LU 6.2 200
CSQX845I +CSQ1 CSQXRQM TCP/IP system name is TCPIP
CSQX848I +CSQ1 CSQXRQM TCP/IP listener INDISP=QMGR not started
CSQX848I +CSQ1 CSQXRQM TCP/IP listener INDISP=GROUP not started
CSQX849I +CSQ1 CSQXRQM LU 6.2 listener INDISP=QMGR not started
CSQX849I +CSQ1 CSQXRQM LU 6.2 listener INDISP=GROUP not started
CSQ9022I +CSQ1 CSQXCRPS ' DISPLAY CHINIT' NORMAL COMPLETION
(2)
STOP CHINIT
(1)
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQM137I +CSQ1 CSQMTCHI STOP CHINIT COMMAND ACCEPTED
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQ9022I +CSQ1 CSQXCRPS ' STOP CHINIT' NORMAL COMPLETION
(2)
CSQU057I CSQUCMDS 3 commands read
CSQU058I CSQUCMDS 3 commands issued and responses received, 0 failed
CSQU143I CSQUTIL 1 COMMAND statements attempted
CSQU144I CSQUTIL 1 COMMAND statements executed successfully
CSQU148I CSQUTIL Utility completed, return code=0
(3)
```

圖 104: CSQ4IVPX 的輸出範例

z/OS 使用 IBM MQ for z/OS 來測試 C、C++、COBOL、PL/I 及 CICS 程式

您可以使用 IBM MQ 隨附的範例應用程式來測試 C、C++、COBOL、PL/I 或 CICS。

IVP (CSQ4IVP1) 提供作為載入模組，並提供範例作為來源模組。您可以使用這些來源模組來測試不同的程式設計語言環境。

如需範例應用程式的相關資訊，請參閱 [IBM MQ for z/OS 的範例程式](#)。

z/OS 在 z/OS 上設定與其他佇列管理程式的通訊

本節說明您在開始使用分散式佇列之前需要進行的 IBM MQ for z/OS 準備。

關於這項作業

若要定義分散式佇列需求，您需要定義下列項目：

- 通道起始程式程序及資料集
- 通道定義
- 佇列及其他物件
- 存取安全

如果您使用佇列共用群組，請參閱 [分散式佇列及佇列共用群組](#)。

如需準備使用 IBM MQ for z/OS 設定分散式佇列作業時要考量的其他要點，請參閱 [第 759 頁的『在 z/OS 上使用分散式佇列的考量』](#)。

程序

若要啟用分散式佇列作業，請完成下列步驟：

- 自訂分散式佇列機能，並定義所需的 IBM MQ 物件，如 [定義系統物件](#) 及 [第 694 頁的『準備在 z/OS 上自訂佇列管理程式』](#) 中所述。
- 定義存取安全，如 [z/OS 上通道起始程式的安全考量](#) 中所述。
- 依照 [第 776 頁的『設定 z/OS 的通訊』](#) 中的說明來設定您的通訊。

相關概念

[第 697 頁的『設定 IBM MQ for z/OS』](#)

請使用本主題作為逐步手冊，以自訂 IBM MQ for z/OS 系統。

相關工作

[第 147 頁的『配置分散式佇列』](#)

本節提供 IBM 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

z/OS 在 z/OS 上使用分散式佇列的考量

準備在 z/OS 上使用分散式佇列時要考量的點。

如果您使用佇列共用群組，請參閱 [分散式佇列及佇列共用群組](#)。

操作員訊息

因為通道起始程式使用許多非同步作業分派器，所以在登出依時間順序時可能會出現操作員訊息。

通道作業指令

通道作業指令通常包含兩個階段。已檢查指令語法並驗證通道是否存在時，會將要求傳送至通道起始程式。訊息 CSQM134I 或 CSQM137I 會傳送至指令發出者，以指出第一個階段已完成。當通道起始程式已處理指令時，會將進一步訊息連同訊息 CSQ9022I 或 CSQ9023E 一起傳送至指令發出者，以指出它是否成功。產生的任何錯誤訊息也可以傳送至 z/OS 主控台。

不過，除了 **DISPLAY CLUSQMGR** 之外，所有叢集指令都以非同步方式運作。變更物件屬性的指令會更新物件，並將要求傳送至通道起始程式。系統會檢查使用叢集的指令語法，並將要求傳送至通道起始程式。在這兩種情況下，都會將訊息 CSQM130I 傳送至指令發出者，指出已傳送要求。此訊息後面接著訊息 CSQ9022I，指出指令已順利完成，因為已傳送要求。它不會指出叢集要求已順利完成。會非同步處理傳送至通道起始程式的要求，以及從叢集其他成員收到的叢集要求。在某些情況下，必須將這些要求傳送至整

個叢集，以判斷它們是否成功。在通道起始程式執行所在的系統上，會向 z/OS 報告任何錯誤。它們不會傳送至指令發出者。

無法遞送的訊息佇列

IBM MQ for z/OS 隨附「無法傳送的郵件」處理程式。如需相關資訊，請參閱 [無法傳送郵件的佇列處理程式公用程式 \(CSQUDLQH\)](#)。

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。此行為會導致佇列顯示為「使用中」。

安全變更

如果您變更使用者 ID 的安全存取權，變更可能不會立即生效。如需相關資訊，請參閱 [z/OS 上通道起始程式的安全考量](#)、[佇列安全的設定檔](#)及 [第 709 頁的『實作 ESM 安全控制』](#)。

已停止通訊-TCP

如果 TCP 因某些原因而停止然後重新啟動，則會停止等待 TCP 埠的 IBM MQ for z/OS TCP 接聽器。

自動通道-重新連接可讓通道起始程式偵測 TCP/IP 無法使用，並在 TCP/IP 傳回時自動重新啟動 TCP/IP 接聽器。此自動重新啟動可讓作業人員減輕注意 TCP/IP 問題的需求，並手動重新啟動接聽器。當接聽器無法運作時，也可以使用通道起始程式，以 LSTRTMR 指定的間隔重試接聽器。這些嘗試可以繼續進行，直到 TCP/IP 返回且接聽器順利自動重新啟動為止。如需 LSTRTMR 的相關資訊，請參閱 [ALTER QMGR](#) 及 [分散式佇列訊息 \(CSQX ...\)](#)。

通訊已停止- LU6.2

如果 APPC 已停止，則接聽器也會停止。同樣地，在此情況下，接聽器會在 LSTRTMR 間隔自動重試，因此如果 APPC 重新啟動，接聽器也可以重新啟動。

如果 Db2 失敗，已在執行中的共用通道會繼續執行，但任何新的通道啟動要求都會失敗。當 Db2 還原時，新要求能夠完成。

z/OS 自動重新啟動管理 (ARM)

自動重新啟動管理 (ARM) 是一種 z/OS 回復功能，可增進特定批次工作或已啟動作業 (例如，子系統) 的可用性。因此，可以加快恢復生產性工作。

若要使用 ARM，您必須以特定方式設定佇列管理程式及通道起始程式，讓它們自動重新啟動。如需相關資訊，請參閱 [使用 z/OS Automatic Restart Manager \(ARM\)](#)。

相關概念

[第 697 頁的『設定 IBM MQ for z/OS』](#)

請使用本主題作為逐步手冊，以自訂 IBM MQ for z/OS 系統。

相關工作

[第 147 頁的『配置分散式佇列』](#)

本節提供 IBM 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

定義 IBM MQ 物件

使用其中一個 IBM MQ 指令輸入方法來定義 IBM MQ 物件。如需定義這些物件的進一步詳細資料，請參閱本主題內的資訊。

如需定義物件的相關資訊，請參閱 [第 761 頁的『在 z/OS 上監視及控制通道』](#)。

傳輸佇列及觸發通道

定義下列項目：

- 每一個傳送訊息通道使用 XMITQ 的本端佇列。
- 遠端佇列定義。

遠端佇列物件有三種不同的用途，視名稱和內容的指定方式而定：

- 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

這三種方式顯示在 [三種使用遠端佇列定義物件的方式](#)中。

使用傳輸佇列上的 TRIGDATA 欄位來觸發指定的通道。例如：

```
DEFINE QLOCAL(MYXMITQ) USAGE(XMITQ) TRIGGER +
INITQ(SYSTEM.CHANNEL.INITQ) TRIGDATA(MYCHANNEL)
DEFINE CHL(MYCHANNEL) CHLTYPE(SDR) TRPTYPE(TCP) +
XMITQ(MYXMITQ) CONNAME('9.20.9.30(1555)')
```

所提供的範例 CSQ4INXD 提供必要定義的其他範例。

z/OS 失去與 CF 結構的連線功能 (其中已定義共用通道的同步化佇列) 或類似問題，可能會暫時阻止通道啟動。解決問題之後，如果您使用觸發類型 FIRST，且通道在觸發時無法啟動，則必須手動啟動通道。如果您想要在解決問題之後自動啟動觸發通道，請考量將佇列管理程式 TRIGINT 屬性設為預設值以外的值。將 TRIGINT 屬性設為預設值以外的值，會導致通道起始程式在傳輸佇列上有訊息時定期重試啟動通道。

同步化佇列

DQM 需要一個與序號及邏輯工作單元 ID (LUWID) 搭配使用的佇列。您必須確定佇列名稱為 SYSTEM.CHANNEL.SYNCQ (請參閱 [在 z/OS 上規劃](#))。此佇列必須可用，否則通道起始程式無法啟動。

請確定您使用 INDXTYPE (MSGID) 來定義此佇列。此屬性可改善可存取它們的速度。

通道指令佇列

您需要確定名稱為 SYSTEM.CHANNEL.INITQ。

如果通道起始程式偵測到 SYSTEM.CHANNEL.INITQ，除非更正問題，否則無法正常繼續。問題可能是下列其中一項：

- 佇列已滿
- 佇列未啟用放置
- 佇列所位於的頁集已滿
- 通道起始程式對佇列沒有正確的安全授權

如果在通道起始程式執行時，佇列定義變更為 GET (DISABLED)，則起始程式無法從佇列取得訊息，並會終止。

啟動通道起始程式

使用通道起始程式來實作觸發。在 IBM MQ for z/OS 上，使用 MQSC 指令 START CHINIT 啟動起始器。

停止通道起始程式

當您停止佇列管理程式時，通道起始程式會自動停止。如果您需要停止通道起始程式，但不需要停止佇列管理程式，則可以使用 MQSC 指令 STOP CHINIT。

z/OS 在 z/OS 上監視及控制通道

使用 DQM 指令和畫面來建立、監視及控制遠端佇列管理程式的通道。

每一個 z/OS 佇列管理程式都有一個 DQM 程式 (通道起始程式) 使用原生 z/OS 機能來控制與遠端佇列管理程式的交互連線。

在 z/OS 上實作這些畫面及指令會整合至作業及控制畫面及 MQSC 指令。這兩組畫面和指令的組織沒有區別。

您也可以使用「可程式化指令格式 (PCF)」指令來輸入指令。如需使用這些指令的相關資訊，請參閱 [自動化管理作業](#)。

本節中的資訊適用於所有使用通道起始程式進行分散式佇列作業的情況。不論您是使用佇列共用群組，還是內部群組佇列作業，都適用。

DQM 通道控制功能

如需分散式佇列管理模型的概觀，請參閱第 166 頁的『[訊息傳送及接收](#)』。

通道控制功能由畫面、指令和程式、兩個同步化佇列、通道指令佇列和通道定義組成。本主題是通道控制功能元件的簡要說明。

- 通道定義會保留為頁集零或 Db2 中的物件，就像 z/OS 中的其他 IBM MQ 物件一樣。
- 您可以使用作業及控制台、MQSC 指令或 PCF 指令來執行下列動作：
 - 建立、複製、顯示、變更及刪除通道定義
 - 啟動和停止通道起始程式及接聽器
 - 當無法重新建立鏈結時，啟動、停止及連線測試通道、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
 - 顯示 DQM 的相關資訊

尤其是您可以使用 CSQINPX 起始設定輸入資料集來發出 MQSC 指令。每次啟動通道起始程式時都可以處理此集。如需相關資訊，請參閱 [起始設定指令](#)。

- 有兩個佇列 (SYSTEM.CHANNEL.SYNCQ 和 SYSTEM.QSG.CHANNEL.SYNCQ) 用於通道重新同步化目的。基於效能考量，請使用 INDXTYPE (MSGID) 來定義這些佇列。
- 通道指令佇列 (SYSTEM.CHANNEL.INITQ) 用於保留通道起始程式、通道及接聽器的指令。
- 通道控制函數程式在其自己的位址空間中執行，與佇列管理程式分開，並包含通道起始程式、接聽器、MCA、觸發監視器及指令處理程式。
- 如需佇列共用群組及共用通道，請參閱 [共用佇列及佇列共用群組](#)。
- 對於內部群組佇列作業，請參閱 [內部群組佇列作業](#)

在 z/OS 上管理頻道

如需如何管理通道、通道起始程式及接聽器的相關資訊，請使用下表中的鏈結：

表 64: 通道作業	
要執行的作業	MQSC 指令
定義通道	定義通道
變更通道定義	ALTER CHANNEL
顯示通道定義	顯示通道
刪除通道定義	刪除通道
啟動通道起始程式	開始 CHINIT
停止通道起始程式	停止 CHINIT
顯示通道起始程式資訊	顯示 CHINIT
啟動通道接聽器	啟動接聽器

表 64: 通道作業 (繼續)	
要執行的作業	MQSC 指令
停止通道接聽器	停止接聽器
啟動通道	啟動通道
測試通道	Ping 通道
重設通道的訊息序號	重設通道
解決通道上不確定的訊息	解析通道
停止通道	停止通道
顯示通道狀態	DISPLAY CHSTATUS
顯示叢集通道	DISPLAY CLUSQMGR

相關概念

第 763 頁的『[使用畫面及指令](#)』

您可以使用 MQSC 指令、PCF 指令或作業及控制台來管理 DQM。

第 697 頁的『[設定 IBM MQ for z/OS](#)』

請使用本主題作為逐步手冊，以自訂 IBM MQ for z/OS 系統。

第 776 頁的『[設定 z/OS 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。若要成功，連線必須定義且可用。本節說明如何定義連線。

第 780 頁的『[為具有佇列共用群組的 DQM 準備 IBM MQ for z/OS](#)』

請使用本節中的指示，在 IBM MQ for z/OS 上配置具有佇列共用群組的分散式佇列。

第 783 頁的『[使用佇列共用群組來設定 IBM MQ for z/OS 的通訊](#)』

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。為了讓此嘗試成功，必須定義連線並使其可用。

相關工作

第 147 頁的『[配置分散式佇列](#)』

本節提供 IBM 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 759 頁的『[在 z/OS 上設定與其他佇列管理程式的通訊](#)』

本節說明您在開始使用分散式佇列之前需要進行的 IBM MQ for z/OS 準備。

使用畫面及指令

您可以使用 MQSC 指令、PCF 指令或作業及控制台來管理 DQM。

如需 MQSC 指令的相關資訊，請參閱 [使用 MQSC 指令管理](#)。如需 PCF 指令的相關資訊，請參閱 [使用可程式指令格式指令自動化](#)。

使用起始畫面

如需呼叫作業及控制台、使用功能鍵及取得說明的簡介，請參閱 [管理 IBM MQ for z/OS](#)。

註：若要使用作業及控制台，您必須具有正確的安全授權；如需相關資訊，請參閱 [管理 IBM MQ for z/OS](#) 及子主題。第 764 頁的圖 105 顯示當您啟動畫面階段作業時所顯示的畫面。畫面之後的文字會說明您在此畫面中執行的動作。

```

IBM MQ for z/OS - Main Menu

Complete fields. Then press Enter.

Action . . . . . 1 0. List with filter 4. Manage
1. List or Display 5. Perform
2. Define like 6. Start
3. Alter 7. Stop
8. Command
Object type . . . . . CHANNEL +
Name . . . . . *
Disposition . . . . . A Q=Qmgr, C=Copy, P=Private, G=Group,
S=Shared, A=All

Connect name . . . . . MQ25 - local queue manager or group
Target queue manager . . . MQ25
- connected or remote queue manager for command input
Action queue manager . . . MQ25 - command scope in group
Response wait time . . . . 10 5 - 999 seconds

(C) Copyright IBM Corporation 1993, 2024. All rights reserved.

Command ==> -----
F1=Help F2=Split F3=Exit F4=Prompt F9=SwapNext F10=Messages
F12=Cancel

```

圖 105: 作業及控制項起始畫面

從這個畫面中，您可以：

- 在 **動作** 欄位中鍵入適當的數字，以選取您要執行的動作。
- 指定您要使用的物件類型。如果您不確定物件類型是什麼，請按 F4 以取得物件類型的清單。
- 顯示指定類型的物件清單。在 **名稱** 欄位中鍵入星號 (*)，然後按 Enter 鍵以顯示已在此子系統上定義的物件 (指定類型) 清單。然後，您可以依序選取一或多個要使用的物件。第 765 頁的圖 106 顯示以這種方式產生的通道清單。
- 在 **處置** 欄位中，指定您要使用之物件的佇列共用群組中的處置。處置會決定物件的保留位置及物件的行為方式。
- 在 **連接名稱** 欄位中，選擇您要連接的本端佇列管理程式或佇列共用群組。如果您想要在遠端佇列管理程式上發出指令，請選擇 **目標佇列管理程式** 欄位或 **動作佇列管理程式** 欄位，視遠端佇列管理程式是否為佇列共用群組的成員而定。如果遠端佇列管理程式不是佇列共用群組的成員，請選擇 **目標佇列管理程式** 欄位。如果遠端佇列管理程式是佇列共用群組的成員，請選擇 **動作佇列管理程式** 欄位。
- 在 **回應等待時間** 欄位中選擇要接收回應的等待時間。

List Channels - MQ25 Row 1 of 8

Type action codes, then press Enter. Press F11 to display connection status.
1=Display 2=Define like 3=Alter 4=Manage 5=Perform
6=Start 7=Stop

```
Name          Type      Disposition Status
<> *          CHANNEL  ALL      MQ25
- SYSTEM.DEF.CLNTCONN CLNTCONN  QMGR  MQ25
- SYSTEM.DEF.CLUSRCVR CLUSRCVR  QMGR  MQ25 INACTIVE
- SYSTEM.DEF.CLUSSDR  CLUSSDR   QMGR  MQ25 INACTIVE
- SYSTEM.DEF.RECEIVER RECEIVER  QMGR  MQ25 INACTIVE
- SYSTEM.DEF.REQUESTER REQUESTER QMGR  MQ25 INACTIVE
- SYSTEM.DEF.SENDER   SENDER    QMGR  MQ25 INACTIVE
- SYSTEM.DEF.SERVER   SERVER     QMGR  MQ25 INACTIVE
- SYSTEM.DEF.SVRCONN  SVRCONN   QMGR  MQ25 INACTIVE
***** End of list *****
```

Command ==>

F1=Help F2=Split F3=Exit F4=Filter F5=Refresh F7=Bkwd
F8=Fwd F9=SwapNext F10=Messages F11=Status F12=Cancel

圖 106: 列出通道

z/OS 在 z/OS 上定義通道

在 z/OS 上，您可以使用 MQSC 指令或使用作業及控制台來定義通道。

若要使用 MQSC 指令定義通道，請使用 [DEFINE CHANNEL](#)。

使用作業及控制面板，從起始畫面開始，完成下列欄位，然後按 Enter 鍵：

欄位	值
動作	2 (定義相似)
物件類型	通道類型 (例如 SENDER) 或 CHANNEL
姓名	
處置	新物件的位置。

系統會向您呈現一些畫面，以完成您要定義之通道的名稱及屬性相關資訊。它們會以預設屬性值來起始設定。在按 Enter 鍵之前變更您想要的任何值。

註：如果您在 **object type** 欄位中輸入 CHANNEL，則會先呈現「選取有效的通道類型」畫面。

如果您要使用與現有通道相同的屬性來定義通道，請在起始畫面的 **Name** 欄位中放置您要複製的通道名稱。畫面會以現有物件的屬性來起始設定。

如需通道屬性的相關資訊，請參閱 [通道屬性](#)

註：

1. 唯一地命名網路中的所有通道。如 [顯示所有通道的網路圖](#) 所示，在通道名稱中包括來源及目標佇列管理程式名稱，是執行此命名的好方法。

在定義頻道之後，您必須保護頻道安全，請參閱 [第 767 頁的『保護通道安全』](#)

z/OS 變更通道定義

您可以使用 MQSC 指令或使用作業及控制台來變更通道定義。

若要使用 MQSC 指令變更通道定義，請使用 ALTER CHANNEL。

使用作業及控制面板，從起始畫面開始，完成下列欄位，然後按 Enter 鍵：

欄位	值
動作	3 (變更)
物件類型	通道類型 (例如 SENDER) 或 CHANNEL
姓名	CHANNEL.TO.ALTER
處置	儲存物件的位置。

您會看到一些畫面，其中包含通道現行屬性的相關資訊。請透過改寫新值來變更您想要的任何未保護欄位，然後按 Enter 鍵以變更通道定義。

如需通道屬性的相關資訊，請參閱 [通道屬性](#)。

顯示通道定義

您可以使用 MQSC 指令或使用作業及控制台來顯示通道定義。

若要使用 MQSC 指令顯示通道定義，請使用 DISPLAY CHANNEL。

使用作業及控制面板，從起始畫面開始，完成下列欄位，然後按 Enter 鍵：

欄位	值
動作	1 (清單或顯示畫面)
物件類型	通道類型 (例如 SENDER) 或 CHANNEL
姓名	CHANNEL.TO.DISPLAY
處置	物件的位置。

您會看到一些畫面，其中顯示通道現行屬性的相關資訊。

如需通道屬性的相關資訊，請參閱 [通道屬性](#)。

刪除通道定義

您可以使用 MQSC 指令或使用作業及控制台來刪除通道定義。

若要使用 MQSC 指令刪除通道定義，請使用 DELETE CHANNEL。

使用作業及控制面板，從起始畫面開始，完成下列欄位，然後按 Enter 鍵：

欄位	值
動作	4 (管理)
物件類型	通道類型 (例如 SENDER) 或 CHANNEL
姓名	CHANNEL.TO.DELETE
處置	物件的位置。

您會看到另一個畫面。在此畫面上選取功能類型 1。

請按 Enter 鍵以刪除通道定義；系統會要求您再次按 Enter 鍵以確認您要刪除通道定義。

註：通道起始程式必須在執行中，才能刪除通道定義 (用戶端連線通道除外)。

顯示通道起始程式的相關資訊

您可以使用 MQSC 指令或使用作業及控制台來顯示通道起始程式的相關資訊。

若要使用 MQSC 指令顯示通道起始程式的相關資訊，請使用 DISPLAY CHINIT。

使用作業及控制面板，從起始畫面開始，完成下列欄位，然後按 Enter 鍵：

欄位	值
動作	1 (顯示器)
物件類型	SYSTEM
姓名	Blank

您會看到另一個畫面。在此畫面上選取功能類型 1。

註:

1. 如果您有許多通道，則顯示分散式佇列資訊可能需要一些時間。
2. 通道起始程式必須在執行中，您才能顯示分散式佇列的相關資訊。

z/OS 保護通道安全

您可以使用 MQSC 指令或使用作業及控制台來保護通道的安全。

若要使用 MQSC 指令來保護通道安全，請使用 [SET CHLAUTH](#)。

使用作業及控制面板，從起始畫面開始，完成下列欄位，然後按 Enter 鍵：

欄位	值
動作	8

您會看到一個編輯器，您可以在其中提供 MQSC 指令 (在此情況下是 CHLAUTH 指令)，請參閱 [第 767 頁的圖 107](#)。當您完成在指令中鍵入時，需要加號 (+)。鍵入 PF3 以結束編輯器，並將指令提交至指令伺服器。

```
***** Top of Data *****
000001 SET CHLAUTH(SYSTEM.DEF.SVRCONN) +
000002 TYPE(SSLPEERMAP) +
000003 SSLPEER('CN="John Smith"') +
000004 MCAUSER('PUBLIC')
***** Bottom of Data *****

Command ==>                               Scroll ==> PAGE
F1=Help  F3=Exit  F4=LineEdit F12=Cancel
```

圖 107: 輸入指令

然後會向您呈現指令的輸出，請參閱 [第 767 頁的圖 108](#)

```
***** Top of Data *****
000001 CSQU000I CSQUTIL IBM MQ for z/OS V7.1.0
000002 CSQU001I CSQUTIL Queue Manager Utility - 2011-04-20 14:42:58
000003 COMMAND TGTQMGR(MQ23) RESPTIME(30)
000004 CSQU127I Executing COMMAND using input from CSQUCMD data set
000005 CSQU120I Connecting to MQ23
000006 CSQU121I Connected to queue manager MQ23
000007 CSQU055I Target queue manager is MQ23
000008 SET CHLAUTH(SYSTEM.DEF.SVRCONN) +
000009 TYPE(SSLPEERMAP) +
000010 SSLPEER('CN="John Smith"') +
000011 MCAUSER('PUBLIC')
000012 CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
000013 CSQ9022I !MQ23 CSQMCA ' SET CHLAUTH ' NORMAL COMPLETION
000014 CSQU057I 1 commands read
000015 CSQU058I 1 commands issued and responses received, 0 failed
000016 CSQU143I 1 COMMAND statements attempted
000017 CSQU144I 1 COMMAND statements executed successfully
000018 CSQU148I CSQUTIL Utility completed, return code=0
Command ==>                               Scroll ==> PAGE
F1=Help  F3=Exit  F5=Rfind  F6=Rchange  F9=SwapNext F12=Cancel
```

圖 108: 指令輸出

z/OS 啟動通道起始程式

您可以使用 MQSC 指令或使用作業及控制台來啟動通道起始程式。

若要使用 MQSC 指令來啟動通道起始程式，請使用 START CHINIT。

使用作業及控制面板，從起始畫面開始，完成下列欄位，然後按 Enter 鍵：

欄位	值
動作	6 (開始)
物件類型	SYSTEM
姓名	Blank

即會顯示「啟動系統功能」畫面。下列畫面後面的文字說明要採取的動作：

```
Start a System Function

Select function type, complete fields, then press Enter to start system
function.

Function type . . . . . _ 1. Channel initiator
2. Channel listener
Action queue manager . . . : MQ25

Channel initiator
JCL substitution . . . . . -----
-----

Channel listener
Inbound disposition . . . Q G=Group, Q=Qmgr
Transport type . . . . . _ L=LU6.2, T=TCP/IP
LU name (LU6.2) . . . . . -----
Port number (TCP/IP) . . . 1414
IP address (TCP/IP) . . . -----

Command ==>
F1=Help F2=Split F3=Exit F9=SwapNext F10=Messages F12=Cancel
```

圖 109: 啟動系統功能

選取函數類型 1 (通道起始程式)，然後按 Enter 鍵。

停止通道起始程式

您可以使用 MQSC 指令或使用作業及控制台來停止通道起始程式。

若要使用 MQSC 指令停止通道起始程式，請使用 STOP CHINIT。

使用作業及控制面板，從起始畫面開始，完成下列欄位，然後按 Enter 鍵：

欄位	值
動作	7 (停止)
物件類型	SYSTEM
姓名	Blank

即會顯示「停止系統功能」畫面。畫面後面的文字說明如何使用此畫面：


```

Stop a System Function

Select function type, complete fields, then press Enter to stop system
function.

Function type . . . . . _ 1. Channel initiator
2. Channel listener
Action queue manager . . . : MQ25

Channel initiator
Restart shared channels Y Y=Yes, N=No

Channel listener
Inbound disposition . . . Q G=Group, Q=Qmgr
Transport type . . . . . _ L=LU6.2, T=TCP/IP

Port number (TCP/IP) . . . -----
IP address (TCP/IP) . . . -----

Command ==> -----
F1=Help F2=Split F3=Exit F9=SwapNext F10=Messages F12=Cancel

```

圖 110: 停止函數控制項

選取函數類型 1 (通道起始程式)，然後按 Enter 鍵。

通道起始程式在停止之前，會等待所有執行中通道以靜止模式停止。

註: 如果部分通道是執行中但非作用中的接收端或要求端通道，則對接收端或傳送端通道起始程式發出的停止要求會導致它立即停止。

不過，如果訊息在流動，通道起始程式會等待現行訊息批次完成，然後再停止。

啟動通道接聽器

您可以使用 MQSC 指令或使用作業及控制台來啟動通道接聽器。

若要使用 MQSC 指令來啟動通道接聽器，請使用 START LISTENER。

使用作業及控制面板，從起始畫面開始，完成下列欄位，然後按 Enter 鍵：

欄位	值
動作	6 (開始)
物件類型	SYSTEM
姓名	Blank

即會顯示「啟動系統功能」畫面 (請參閱 [第 768 頁的圖 109](#))。

選取函數類型 2 (通道接聽器)。選取入埠處置。選取傳輸類型。如果傳輸類型是 L，請選取 LU 名稱。如果傳輸類型為 T，請選取埠號及 (選擇性地) IP 位址。按 Enter 鍵。

註: 對於 TCP/IP 接聽器，您可以啟動多個埠和 IP 位址組合。

停止通道接聽器

您可以使用 MQSC 指令或使用作業及控制台來停止通道接聽器。

若要使用 MQSC 指令停止通道接聽器，請使用 STOP LISTENER。

使用作業及控制面板，從起始畫面開始，完成下列欄位，然後按 Enter 鍵：

欄位	值
動作	7 (停止)

欄位	值
物件類型	SYSTEM
姓名	Blank

即會顯示「停止系統功能」畫面 (請參閱 第 769 頁的圖 110)。

選取函數類型 2 (通道接聽器)。選取入埠處置。選取傳輸類型。如果傳輸類型為 'T'，請選取埠號及 (選擇性地) IP 位址。按 Enter 鍵。

註: 對於 TCP/IP 接聽器，您可以停止埠和 IP 位址的特定組合，也可以停止所有組合。

啟動通道

您可以使用 MQSC 指令或使用作業及控制台來啟動通道。

若要使用 MQSC 指令啟動通道，請使用 START CHANNEL。

使用作業及控制面板，從起始畫面開始，完成下列欄位，然後按 Enter 鍵:

欄位	值
動作	6 (開始)
物件類型	通道類型 (例如 SENDER) 或 CHANNEL
姓名	CHANNEL.TO.USE
處置	物件的處置。

即會顯示「啟動通道」畫面。畫面後面的文字說明如何使用畫面:

```

Start a Channel

Select disposition, then press Enter to start channel.

Channel name . . . . . : CHANNEL.TO.USE
Channel type . . . . . : SENDER
Description . . . . . : Description of CHANNEL.TO.USE

Disposition . . . . . P   P=Private on MQ25
S=Shared on MQ25
A=Shared on any queue manager

Command ==>
F1=Help   F2=Split   F3=Exit   F9=SwapNext F10=Messages F12=Cancel

```

圖 111: 啟動通道

選取通道實例的處置，以及要啟動它的佇列管理程式。

按 Enter 鍵以啟動通道。

啟動共用通道

如果要啟動共用通道，並將它保留在指定的通道起始程式上，請使用 disposition = S (在 START CHANNEL 指令上，指定 CHLDISP (FIXSHARED))。

一次只能有一個共用通道實例在執行中。嘗試啟動通道的第二個實例失敗。

以這種方式啟動通道時，下列規則適用於該通道：

- 您可以從佇列共用群組中的任何佇列管理程式停止通道。即使在您發出停止通道要求時，其啟動所在的通道起始程式未在執行中，您也可以停止它。當通道已停止時，您可以在相同或另一個通道起始程式上指定 disposition = S (CHLDISP (FIXSHARED)) 來重新啟動它。您也可以指定 disposition = A (CHLDISP (SHARED)) 來啟動它。
- 如果通道處於啟動中或重試狀態，您可以在相同或不同的通道起始程式上指定 disposition = S (CHLDISP (FIXSHARED)) 來重新啟動它。您也可以指定 disposition = A (CHLDISP (SHARED)) 來啟動它。
- 當通道進入非作用中狀態時，即可啟動該通道。觸發程式啟動的共用通道一律具有共用處置 (CHLDISP (SHARED))。
- 當通道進入非作用中狀態時，可以在任何通道起始程式上使用 CHLDISP (FIXSHARED) 來啟動通道。您也可以指定 disposition = A (CHLDISP (SHARED)) 來啟動它。
- 當啟動通道的通道起始程式使用 SHARED (RESTART) 停止時，或當通道起始程式異常終止時，佇列共用群組中的任何其他作用中通道起始程式都無法回復通道。只有在下次重新啟動通道啟動所在的通道起始程式時，才會回復通道。這會停止失敗的通道回復嘗試傳遞至佇列共用群組中的其他通道起始程式，這會新增至其工作量。

測試通道

您可以使用 MQSC 指令或使用作業及控制台來測試通道。

若要使用 MQSC 指令測試通道，請使用 PING CHANNEL。

使用作業及控制面板，從起始畫面開始，完成下列欄位，然後按 Enter 鍵：

欄位	值
動作	5 (執行)
物件類型	SENDER、SERVER 或 CHANNEL
姓名	CHANNEL.TO.USE
處置	通道物件的處置。

即會顯示「執行通道功能」畫面。畫面後面的文字說明如何使用畫面：

```
Perform a Channel Function
Select function type, complete fields, then press Enter.

Function type . . . . . _ 1. Reset 3. Resolve with commit
2. Ping 4. Resolve with backout

Channel name . . . . . : CHANNEL.TO.USE
Channel type . . . . . : SENDER
Description . . . . . : Description of CHANNEL.TO.USE

Disposition . . . . . P P=Private on MQ25
S=Shared on MQ25
A=Shared on any queue manager

Sequence number for reset . . 1 1 - 999999999
Data length for ping . . . . 16 16 - 32768

Command ==>
F1=Help F2=Split F3=Exit F9=SwapNext F10=Messages F12=Cancel
```

圖 112: 測試通道

選取函數類型 2 (ping)。

選取要完成測試的通道處置，以及要測試的佇列管理程式。

資料長度最初設為 16。如果您想要的話，請變更它，然後按 Enter 鍵。

z/OS 重設通道的訊息序號

您可以使用 MQSC 指令或使用作業及控制台來重設通道的訊息序號。

若要使用 MQSC 指令重設通道序號，請使用 RESET CHANNEL。

使用作業及控制面板，從起始畫面開始，完成下列欄位，然後按 Enter 鍵：

欄位	值
動作	5 (執行)
物件類型	通道類型 (例如 SENDER) 或 CHANNEL
姓名	CHANNEL.TO.USE
處置	通道物件的處置。

即會顯示「執行通道功能」畫面 (請參閱 [第 771 頁的圖 112](#))。

選取功能類型 1 (重設)。

選取要對其執行重設的通道處置，以及要在哪個佇列管理程式上執行重設的通道處置。

序號 欄位最初設為 1。如果您想要的話，請變更此值，然後按 Enter 鍵。

z/OS 解決通道上不確定的訊息

您可以使用 MQSC 指令或使用作業及控制台來解決通道上不確定的訊息。

若要使用 MQSC 指令解決通道上不確定的訊息，請使用 RESOLVE CHANNEL。

使用作業及控制面板，從起始畫面開始，完成下列欄位，然後按 Enter 鍵：

欄位	值
動作	5 (執行)
物件類型	SENDER、SERVER 或 CHANNEL
姓名	CHANNEL.TO.USE
處置	物件的處置。

即會顯示「執行通道功能」畫面 (請參閱 [第 771 頁的圖 112](#))。

選取函數類型 3 或 4 (使用確定或取消來解析)。(如需相關資訊，請參閱 [第 183 頁的『處理不確定的通道』](#)。)

選取要完成解析的通道處置，以及要完成解析的佇列管理程式。按 Enter 鍵。

z/OS 停止通道

您可以使用 MQSC 指令或使用作業及控制台來停止通道。

若要使用 MQSC 指令停止通道，請使用 STOP CHANNEL。

使用作業及控制面板，從起始畫面開始，完成下列欄位，然後按 Enter 鍵：

欄位	值
動作	7 (停止)
物件類型	通道類型 (例如 SENDER) 或 CHANNEL
姓名	CHANNEL.TO.USE

欄位	值
處置	物件的處置。

即會顯示「停止通道」畫面。畫面後面的文字說明如何使用畫面:

```

Stop a Channel

Complete fields, then press Enter to stop channel.

Channel name . . . . . : CHANNEL.TO.USE
Channel type . . . . . : SENDER
Description . . . . . : Description of CHANNEL.TO.USE

Disposition . . . . . P   P=Private on MQ25
A=Shared on any queue manager

Stop mode . . . . . 1   1. Quiesce  2. Force
Stop status . . . . . 1   1. Stopped  2. Inactive

Queue manager . . . . . -----
Connection name . . . . . -----

Command ===>
F1=Help   F2=Split   F3=Exit   F9=SwapNext F10=Messages F12=Cancel

```

圖 113: 停止通道

選取要執行停止的通道處置，以及要在哪個佇列管理程式上停止它。

選擇您需要的停止模式:

靜止

當現行訊息已完成，然後批次結束時，通道會停止，即使尚未達到批次大小值，且有訊息已在傳輸佇列中等待。未啟動任何新批次。此模式是預設值。

強制

通道立即停止。如果正在進行批次訊息，則可能會產生「不確定」狀況。

選擇您要停止之通道的佇列管理程式及連線名稱。

選擇您需要的狀態:

已停止

通道不會自動重新啟動，且必須手動重新啟動。如果未指定佇列管理程式或連線名稱，則此模式是預設值。如果指定名稱，則不容許該名稱。

非作用中

必要時，會自動重新啟動通道。如果指定佇列管理程式或連線名稱，則此模式是預設值。

請按 Enter 鍵停止通道。

如需相關資訊，請參閱第 181 頁的『[停止及靜止通道](#)』。如需重新啟動已停止通道的相關資訊，請參閱第 182 頁的『[重新啟動已停止的通道](#)』。

註: 如果共用通道處於重試狀態，且其啟動所在的通道起始程式不在執行中，則會在輸入指令的佇列管理程式上發出通道的 STOP 要求。

顯示通道狀態

您可以使用 MQSC 指令或使用作業及控制台來顯示通道狀態。

若要使用 MQSC 指令顯示通道或一組通道的狀態，請使用 DISPLAY CHSTATUS。

註: 如果您有許多通道，則顯示通道狀態資訊可能需要一些時間。

使用「清單通道」畫面上的作業及控制面板(請參閱 第 765 頁的圖 106)，會顯示每一個通道的通道狀態摘要，如下所示：

INACTIVE	沒有作用中的連線
狀態	一個連線作用中
<i>nnn</i> 狀態	多個連線是現行連線，且所有現行連線都具有相同的狀態
<i>nnn</i> 現行	多個連線是現行連線，且現行連線並非全部具有相同的狀態
Blank	IBM MQ 無法判斷作用中的連線數(例如，因為通道起始程式不在執行中)

註：對於處置 GROUP 的通道物件，不會顯示任何狀態。

其中 *nnn* 是作用中連線數，而 *status* 是下列其中一項：

起始設定	INITIALIZING
BIND	BINDING
START	STARTING
RUN	RUNNING
停止	STOPPING 或 STOPPED
RETRY	RETRYING
REQST	REQUESTING

若要顯示通道狀態的相關資訊，請按「清單通道」或「顯示畫面」上的「狀態」鍵(F11)，或「變更通道」畫面，以顯示「清單通道-現行狀態」畫面(請參閱 第 774 頁的圖 114)。

```
List Channels - Current Status - MQ25      Row 1 of 16

Type action codes, then press Enter. Press F11 to display saved status.
1=Display current status

Channel name      Connection name      State
Start time      Messages Last message time Type Disposition
<> *
CHANNEL ALL MQ25

- RMA0.CIRCUIT.ACL.F RMA1
- 2005-03-21 10.22.36 557735 2005-03-24 09.51.11 SENDER PRIVATE MQ25
- RMA0.CIRCUIT.ACL.N RMA1
- 2005-03-21 10.23.09 378675 2005-03-24 09.51.10 SENDER PRIVATE MQ25
- RMA0.CIRCUIT.CL.F RMA2
- 2005-03-24 01.12.51 45544 2005-03-24 09.51.08 SENDER PRIVATE MQ25
- RMA0.CIRCUIT.CL.N RMA2
- 2005-03-24 01.13.55 45560 2005-03-24 09.51.11 SENDER PRIVATE MQ25
- RMA1.CIRCUIT.CL.F RMA1
- 2005-03-21 10.24.12 360757 2005-03-24 09.51.11 RECEIVER PRIVATE MQ25
- RMA1.CIRCUIT.CL.N RMA1
- 2005-03-21 10.23.40 302870 2005-03-24 09.51.09 RECEIVER PRIVATE MQ25
***** End of list *****
Command ==>
F1=Help F2=Split F3=Exit F4=Filter F5=Refresh F7=Bkwd
F8=Fwd F9=SwapNext F10=Messages F11=Saved F12=Cancel
```

圖 114: 列出通道連線

狀態的值如下：

起始設定	INITIALIZING
BIND	BINDING
START	STARTING

RUN	RUNNING
停止	STOPPING 或 STOPPED
RETRY	RETRYING
REQST	REQUESTING
不確定	STOPPED 和 INDOUBT (YES)

如需相關資訊，請參閱 [第 174 頁的『通道狀態』](#)。

您可以按 F11 以查看具有已儲存狀態的通道連線類似清單; 按 F11 以回到現行清單。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。

請使用動作碼 1 或斜線 (/) 來選取連線，然後按 Enter 鍵。即會顯示「顯示通道連線現行狀態」畫面。

顯示叢集通道

您可以使用 MQSC 指令或使用作業及控制台來顯示叢集通道。

若要顯示已定義 (明確地或使用自動定義) 的所有叢集通道，請使用 MQSC 指令 DISPLAY CLUSQMGR。

使用作業及控制面板，從起始畫面開始，完成下列欄位，然後按 Enter 鍵：

欄位	值
動作	1 (清單或顯示畫面)
物件類型	CLUSCHL
姓名	*

您會看到類似圖 [第 775 頁的圖 115](#) 的畫面，其中每一個叢集通道的資訊會佔用三行，並包括其通道、叢集及佇列管理程式名稱。對於叢集傳送端通道，會顯示整體狀態。

```
List Cluster queue manager Channels - MQ25      Row 1 of 9

Type action codes, then press Enter. Press F11 to display connection status.
1=Display 5=Perform 6=Start 7=Stop

Channel name      Connection name      State
Type      Cluster name      Suspended
Cluster queue manager name      Disposition
<> *          -          MQ25
- TO.MQ90.T      HURSLEY.MACH90.COM(1590)
- CLUSRCVR      VJH01T              N
- MQ90          -          MQ25
- TO.MQ95.T      HURSLEY.MACH95.COM(1595)      RUN
- CLUSSDRA      VJH01T              N
- MQ95          -          MQ25
- TO.MQ96.T      HURSLEY.MACH96.COM(1596)      RUN
- CLUSSDRB      VJH01T              N
- MQ96          -          MQ25
***** End of list *****

Command ==> -----
F1=Help  F2=Split  F3=Exit  F4=Filter  F5=Refresh  F7=Bkwd
F8=Fwd   F9=SwapNext  F10=Messages  F11=Status  F12=Cancel
```

圖 115: 列出叢集通道

若要顯示一或多個通道的完整資訊，請針對其名稱鍵入「動作碼 1」，然後按 Enter 鍵。使用動作碼 5、6 或 7 來執行功能 (例如 ping、resolve 及 reset)，以及啟動或停止叢集通道。

若要顯示通道狀態的相關資訊，請按「狀態」鍵 (F11)。

z/OS 準備 **IBM MQ for z/OS** 以使用 **zEnterprise Data Compression Express** 機能
zEnterprise 資料壓縮 (zEDC) Express 機能適用於某些型號的 IBM Z 機器，從 IBM zEC12 GA2 開始，最低 z/OS 層次為 z/OS 2.1。

如需進一步資訊，請參閱 [zEnterprise 資料壓縮 \(zEDC\)](#)。

必要條件

對於 IBM z15 以及更新版本，zEnterprise 資料壓縮 (zEDC) Express 機能已從硬體系統的 PCIe I/O 抽換匣中的選購配件移至晶片上，以作為 zEDC 的 Integrated Accelerator。有了這項變更，配置必備項目會更新且相依於您的硬體系統。

IBM z15 或更新版本

根據您的 z/OS 層次，套用下列其中一個 PTF：

- z/OS 2.4: UJ00636
- z/OS 2.3: UJ00635
- z/OS 2.2: UJ00638
- z/OS 2.1: UJ00639

z15 或更新版本系統沒有硬體需求。這些系統中的 Integrated Accelerator for zEDC 解決方案提供內建資料加速，因此不再需要個別的配接卡。

IBM zEC12 GA2 至 IBM z14

您的系統也必須具備下列需求：

- zEDC Express[®] 配接卡，安裝在硬體系統的 PCIe I/O 抽換匣中。
- 必須在 IFAPRDxx parmlib 成員中啟用 zEDC 軟體功能 (選用的 paid-for 特性)。

程序

IBM zEC12 GA2 至 IBM z14

請確定通道起始程式使用者 ID 對 RACF FACILITY CLASS 中的 FPZ.ACCELERATOR.COMPRESSION 設定檔具有 READ 權限，或您企業使用的外部安全管理程式 (ESM) 中的對等項目。



小心： IBM z15 或更新版本不需要。

IBM zEnterprise zEC12 GA2 或更新版本

在傳送端和接收端使用 COMPMSG (ZLIBFAST) 來配置通道。配置之後，會使用 zlib 壓縮來壓縮及解壓縮流經通道的訊息。

當要壓縮的資料大小高於臨界值下限時，會在 zEDC 中執行壓縮。臨界值視所使用的 IBM z 硬體而定

- IBM zEC12 GA2 至 IBM z14 的臨界值下限為 4KB
- IBM z15 或更新版本的臨界值下限為 1KB

對於低於臨界值大小的訊息，會在軟體中執行壓縮或膨脹。

z/OS 設定 z/OS 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。若要成功，連線必須定義且可用。本節說明如何定義連線。

DQM 是 IBM MQ 的遠端佇列作業機能。它為形成通訊鏈結介面的佇列管理程式提供通道控制程式。這些鏈結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。

從可用於 z/OS 的兩種通訊協定形式之一進行選擇：

- 第 777 頁的『在 z/OS 上定義 TCP 連線』

- [第 779 頁的『使用 APPC/MVS 為 z/OS 定義 LU6.2 連線』](#)

V 9.1.4 **MQ Adv.** **CD** 使用 TCP/IP 的訊息通道可以指向 IBM Aspera fasp.io Gateway，它提供快速 TCP/IP 通道，可大幅增加網路傳輸量。在任何授與的 CD 平台上執行的佇列管理程式可以透過 Aspera gateway 進行連接。閘道本身部署在 Red Hat 上或 Ubuntu Linux。請參閱 [在 Linux 上定義 Aspera gateway 連線](#)。

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多個通訊協定進行通訊。

您也可能會發現參照 [配置範例- IBM MQ for z/OS 會很有用](#)。如果您使用佇列共用群組，請參閱 [第 783 頁的『使用佇列共用群組來設定 IBM MQ for z/OS 的通訊』](#)。

相關概念

[第 763 頁的『使用畫面及指令』](#)

您可以使用 MQSC 指令、PCF 指令或作業及控制台來管理 DQM。

[第 697 頁的『設定 IBM MQ for z/OS』](#)

請使用本主題作為逐步手冊，以自訂 IBM MQ for z/OS 系統。

[第 761 頁的『在 z/OS 上監視及控制通道』](#)

使用 DQM 指令和畫面來建立、監視及控制遠端佇列管理程式的通道。

[第 780 頁的『為具有佇列共用群組的 DQM 準備 IBM MQ for z/OS』](#)

請使用本節中的指示，在 IBM MQ for z/OS 上配置具有佇列共用群組的分散式佇列。

[第 783 頁的『使用佇列共用群組來設定 IBM MQ for z/OS 的通訊』](#)

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。為了讓此嘗試成功，必須定義連線並使其可用。

相關工作

[第 147 頁的『配置分散式佇列』](#)

本節提供 IBM 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 759 頁的『在 z/OS 上設定與其他佇列管理程式的通訊』](#)

本節說明您在開始使用分散式佇列之前需要進行的 IBM MQ for z/OS 準備。

z/OS 在 z/OS 上定義 TCP 連線

若要定義 TCP 連線，需要配置一些設定。

TCP 位址空間名稱必須在 TCP 系統參數資料集 *tcipip.TCPIP.DATA*。在資料集中，必須併入 "TCPIPJOBNAME *TCPIP_proc*" 陳述式。

如果您使用防火牆，則需要配置從通道起始程式到通道中位址的 allow 連線，以及從遠端連線到佇列管理程式的連線。

一般而言，防火牆的定義會將傳送端 IP 位址及埠配置為目的地 IP 位址及埠：

- z/OS 映像檔可以有多个主機名稱，而且您可能需要配置防火牆，並以多個主機位址作為來源位址。

您可以使用 NETSTAT HOME 指令來顯示這些名稱和位址。

- 通道起始程式可以在不同埠上具有多個接聽器，因此您需要配置這些埠。
- 如果您使用佇列共用群組的共用埠，則也必須配置共用埠。

通道起始程式位址空間必須具有讀取資料集的權限。可以使用下列技術來存取 TCPIP.DATA 資料集，視您使用的 TCP/IP 產品及介面而定：

- 環境變數，RESOLVER_CONFIG
- 檔案系統上的 /etc/resolv.conf
- // SYSTCPD DD 陳述式
- // SYSTCPDD DD 陳述式
- *jobname/userid.TCPIP.DATA*

- SYS1.TCPPARMS(TCPDATA)
- *zapname*.TCPIP.DATA

您也必須小心正確地指定 TCP/IP 的高階限定元。

您需要適當配置的「網域名稱系統 (DNS)」伺服器，能夠同時進行「名稱至 IP」位址轉換及 IP 位址至「名稱」轉換。

註：解析器配置的部分變更需要回收使用它的應用程式，例如 IBM MQ。

如需相關資訊，請參閱下列內容：

- [基本 TCP/IP 系統](#)
- [z/OS UNIX System Services](#).

啟動時每一個 TCP 通道都會使用 TCP 資源；您可能需要在 PROFILE.TCPIP 配置資料集：

ACBPOOLSIZE

每個已啟動的 TCP 通道新增一個，外加一個

CCBPOOLSIZE

每個已啟動的 TCP 通道各新增一個，每個 DQM 分派器各新增一個，外加一個

DATABASUFFERPOOLSIZE

每個已啟動的 TCP 通道新增兩個，外加一個

MAXFILEPROC

控制通道起始程式中每個分派器可以處理的通道數。

此參數指定在 SYSI.PARMLIB。請確定您指定的值夠大，足以滿足您的需求。

依預設，通道起始程式只能連結至與 TCPNAME 佇列管理程式屬性中所指定堆疊相關聯的 IP 位址。若要容許通道起始程式使用系統上的其他 TCP/IP 堆疊進行通訊，請將 TCPSTACK 佇列管理程式屬性變更為 MULTIPLE。

V 9.1.4 **MQ Adv.** **CD** 使用 TCP/IP 的訊息通道可以指向 IBM Aspera fasp.io Gateway，它提供快速 TCP/IP 通道，可大幅增加網路傳輸量。在任何授與的 CD 平台上執行的佇列管理程式可以透過 Aspera gateway 進行連接。閘道本身部署在 Red Hat 上或 Ubuntu Linux。請參閱 [在 Linux 上定義 Aspera gateway 連線](#)。

相關概念

第 778 頁的『[傳送端](#)』

在 TCP/IP 連線的傳送端，有許多設定要配置。

第 779 頁的『[在 TCP 上接收](#)』

在 TCP/IP 連線的接收端，有許多設定要配置。

第 779 頁的『[使用 TCP 接聽器待辦事項選項](#)』

在 TCP/IP 上接收時，會設定未完成的連線要求數目上限。這些未完成的要求可視為在 TCP/IP 埠上等待接聽器接受要求之要求的待辦事項。

z/OS 傳送端

在 TCP/IP 連線的傳送端，有許多設定要配置。

通道定義中的連線名稱 (CONNAME) 欄位必須設為目標的主機名稱 (例如 MVSHUR1) 或 TCP 網址。TCP 網址可以是 IPv4 帶點十進位格式 (例如 127.0.0.1) 或 IPv6 十六進位格式 (例如 2001:DB8:0:0:0:0:0)。如果連線名稱是主機名稱，則需要 TCP 名稱伺服器才能將主機名稱轉換為 TCP 主機位址。(此需求是 TCP 而非 IBM MQ 的函數。)

在連線的起始端 (傳送端、要求端及伺服器通道類型)，可以提供連線的選用埠號，例如：

連線名稱

192.0.2.0(1555)

在此情況下，起始端會嘗試連接至在埠 1555 上接聽的接收程式。

註：如果未指定選用埠號，則會使用預設埠號 1414。

通道起始程式可以使用任何作用中且可用的 TCP/IP 堆疊。依預設，通道起始程式會將其出埠通道連結至 TCPNAME 佇列管理程式屬性中所指名之 TCP/IP 堆疊的預設 IP 位址。如果要透過不同的堆疊來連接，您需要在通道的 LOCLADDR 屬性中指定堆疊的主機名稱或 IP 位址。

z/OS 在 TCP 上接收

在 TCP/IP 連線的接收端，有許多設定要配置。

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。您可以使用 [START LISTENER](#) 指令或使用作業和控制台來啟動此接聽器程式。

依預設：

- 「TCP 接聽器」程式會使用埠 1414，並在 TCP 堆疊可用的所有位址上接聽。
- TCP/IP 接聽器只能連結至與 TCPNAME 佇列管理程式屬性中所指名之 TCP/IP 堆疊相關聯的位址。

若要啟動其他位址或所有可用 TCP 堆疊的接聽器，請將 TCPSTACK 佇列管理程式屬性設為 'MULTIPLE'。

您可以在 START LISTENER 指令中指定 IPADDR，以啟動 TCP 接聽器程式只在特定位址或主機名稱上接聽。如需相關資訊，請參閱[接聽器](#)。

z/OS 使用 TCP 接聽器待辦事項選項

在 TCP/IP 上接收時，會設定未完成的連線要求數目上限。這些未完成的要求可視為在 TCP/IP 埠上等待接聽器接受要求之要求的待辦事項。

z/OS 上的預設接聽器待辦事項值是 10000。如果待辦事項達到此值，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，且可以稍後重試連線。

z/OS 使用 APPC/MVS 為 z/OS 定義 LU6.2 連線

若要定義 LU6.2 連線，則需要配置一些設定。

APPC/MVS 設定

通道起始程式的每一個實例都必須具有要在 SYS1.PARMLIB，如下列範例所示：

```
LUADD ACBNAME( luname ) NOSCHED TPDATA(CSQ.APPCTP)
```

luname 是要使用的邏輯單元名稱。需要 NOSCHED；不使用 TPDATA。不需要新增至 ASCHPMxx 成員或 APPC/MVS TP 設定檔資料集。

必須延伸端資訊資料集，以定義 DQM 所使用的連線。如需如何使用 APPC 公用程式 ATBSDLFUMU 執行此動作的詳細資料，請參閱提供的範例 CSQ4SIDE。如需要使用之 TPNAME 值的詳細資料，請參閱下表以取得相關資訊：

遠端平台	TPNAME
z/OS 或 MVS	與遠端佇列管理程式對應端資訊中的 TPNAME 相同。
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對應端資訊中的 TPNAME 相同。
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。

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。

另請參閱 [多平台 APPC 配置手冊](#)，以取得可能需要之 VTAM 定義的相關資訊。

在佇列管理程式使用 APPC 與相同或其他 z/OS 系統上的佇列管理程式進行通訊的環境中，請確定通訊 LU 的 VTAM 定義指定 SECACPT (ALREADYV)，或有 LU 之間連線的 RACF APPCLU 設定檔指定 CONVSEC (ALREADYV)。

在嘗試啟動入埠或出埠通訊之前，必須針對基本及接聽器 LU 發出 z/OS 指令 VARY ACTIVE。



小心: 除了 APPC 設定之外，您還必須發出下列指令：

```
ALTER QMGR LUNAME(luname)
```

然後重新啟動通道起始程式。

如需進一步資訊，請參閱 [LUNAME](#)。

相關概念

第 780 頁的『[連接至 LU 6.2](#)』

若要連接至 LU 6.2，需要配置一些設定。

第 780 頁的『[在 LU 6.2 上接收](#)』

若要在 LU 6.2 上接收，需要配置一些設定。

z/OS 連接至 LU 6.2

若要連接至 LU 6.2，需要配置一些設定。

通道定義中的連線名稱 (CONNNAME) 欄位必須設為符號式目的地名稱，如 APPC/MVS 的端資訊資料集中所指定。

要使用的 LU 名稱 (如先前所述定義給 APPC/MVS) 也必須在通道起始程式參數中指定。它必須設為接聽器用來接收的相同 LU。

通道起始程式使用 "SECURITY (SAME)" APPC/MVS 選項，因此它是用於出埠傳輸的通道起始程式位址空間的使用者 ID，並呈現給接收端。

z/OS 在 LU 6.2 上接收

若要在 LU 6.2 上接收，需要配置一些設定。

接收 MCA 是為了回應來自傳送端通道的啟動要求而啟動。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。接聽器程式是 APPC/MVS 伺服器。您可以使用 START LISTENER 指令或使用作業和控制台來啟動它。您必須指定 LU 名稱，以與週邊資訊資料集中定義的符號式目的地名稱搭配使用。如此識別的本端 LU 必須與用於出埠傳輸的本端 LU 相同，如通道起始程式參數中所設定。

z/OS 為具有佇列共用群組的 DQM 準備 IBM MQ for z/OS

請使用本節中的指示，在 IBM MQ for z/OS 上配置具有佇列共用群組的分散式佇列。

如需使用佇列共用群組的範例配置，請參閱 [範例配置- IBM MQ for z/OS 使用佇列共用群組](#)。如需使用佇列共用群組的訊息通道規劃範例，請參閱 [z/OS 使用佇列共用群組的訊息通道規劃範例](#)。

您需要建立並配置下列元件，以啟用具有佇列共用群組的分散式佇列作業：

- [LU 6.2 和 TCP/IP 接聽器](#)
- [傳輸佇列及觸發](#)
- [訊息通道代理程式](#)
- [同步化佇列](#)

建立元件之後，您需要設定通訊，請參閱 [第 783 頁的『使用佇列共用群組來設定 IBM MQ for z/OS 的通訊』](#)。

如需如何在使用佇列共用群組時監視及控制通道的相關資訊，請參閱 [第 761 頁的『在 z/OS 上監視及控制通道』](#)。

如需佇列共用群組概念及好處，請參閱下列各節。

服務類別

共用佇列是提供不同服務類別的本端佇列類型。共用佇列上的訊息儲存在連結機能 (CF) 中，可讓佇列共用群組中的所有佇列管理程式存取它們。共用佇列上的訊息必須是長度不超過 100 MB 的訊息。

通用介面

佇列共用群組有一個通用介面，可讓網路將群組當作單一實體來檢視。此視圖是透過具有單一通用位址來達成，該位址可用來連接至群組內的任何佇列管理程式。

佇列共用群組中的每一個佇列管理程式都會在與一般位址邏輯相關的位址上接聽入埠階段作業要求。如需相關資訊，請參閱 [第 782 頁的『佇列共用群組的 LU 6.2 及 TCP/IP 接聽器』](#)。

負載平衡通道啟動

在佇列共用群組中的任何通道起始程式上執行的出埠通道可以處理共用傳輸佇列。負載平衡通道啟動決定啟動通道指令的目標位置。選擇具有必要通訊子系統存取權的適當通道起始程式。例如，以 TRPTYPE (LU6.2) 定義的通道無法在只能存取 TCP/IP 子系統的通道起始程式上啟動。

通道起始程式的選項取決於通道負載及通道起始程式的空餘空間。通道負載是作用中通道數佔通道起始程式參數中所定義容許的作用中通道數上限的百分比。空餘空間是作用中通道數與容許數目上限之間的差異。

透過使用一般位址，可以在佇列共用群組中對入埠共用通道進行負載平衡，如 [第 782 頁的『佇列共用群組的 LU 6.2 及 TCP/IP 接聽器』](#) 中所述。

共用通道回復

下表顯示共用通道失敗的類型，以及如何處理每一種類型。

失敗類型:	發生情況:
通道起始程式通訊子系統失敗	相依於通訊子系統的通道會進入通道重試，並由負載平衡啟動指令在適當的佇列共用群組通道起始程式上重新啟動。
通道起始程式失敗	通道起始程式失敗，但相關聯的佇列管理程式仍保持作用中。佇列管理程式會監視失敗並起始回復處理。
佇列管理程式失敗	佇列管理程式失敗 (使相關聯通道起始程式失敗)。佇列共用群組中的其他佇列管理程式會監視事件並起始同層級回復。
共用狀態失敗	通道狀態資訊儲存在 Db2 中，因此當通道狀態變更時，失去與 Db2 的連線會變成失敗。執行中通道可以繼續執行，而不需要存取這些資源。在無法存取 Db2 時，通道會進入重試。

代表失敗系統的共用通道回復處理需要在管理回復的系統上提供與 Db2 的連線功能，以擷取共用通道狀態。

用戶端通道

用戶端連線通道可以受益於連接至通用介面之佇列共用群組中的訊息高可用性，而不是連接至特定佇列管理程式。如需相關資訊，請參閱 [用戶端連線通道](#)。

相關概念

共用佇列及佇列共用群組

[第 697 頁的『設定 IBM MQ for z/OS』](#)

請使用本主題作為逐步手冊，以自訂 IBM MQ for z/OS 系統。

[第 783 頁的『叢集和佇列共用群組』](#)

您可以在單一定義中使共用佇列可供叢集使用。若要這樣做，您可以在定義共用佇列時指定叢集的名稱。

[第 783 頁的『通道和序列化』](#)

在共用佇列同層級回復期間，處理共用佇列上訊息的訊息通道代理程式會序列化其對佇列的存取權。

內部群組佇列

相關工作

第 147 頁的『[配置分散式佇列](#)』

本節提供 IBM 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 759 頁的『[在 z/OS 上設定與其他佇列管理程式的通訊](#)』

本節說明您在開始使用分散式佇列之前需要進行的 IBM MQ for z/OS 準備。

z/OS 佇列共用群組的 LU 6.2 及 TCP/IP 接聽器

群組 LU 6.2 及 TCP/IP 接聽器會接聽邏輯連接至一般位址的位址。

對於 LU 6.2 接聽器，指定的 LUGROUP 會對映至與佇列共用群組相關聯的 VTAM 通用資源。如需設定此技術的範例，請參閱第 779 頁的『[使用 APPC/MVS 為 z/OS 定義 LU6.2 連線](#)』。

對於 TCP/IP 接聽器，可以使用下列其中一種方式，將指定的埠連接至通用位址：

- 對於前端路由器 (例如 IBM Network Dispatcher)，入埠連接要求會從路由器轉遞至佇列共用群組的成員。
- 對於 TCP/IP Sysplex Distributor，每一個正在執行且正在接聽設定為分散式 DVIPA 之特定位址的接聽器，都會配置一定比例的送入要求。如需設定此技術的範例，請參閱 [使用 Sysplex Distributor](#)

z/OS 傳輸佇列及佇列共用群組的觸發

在訊息從佇列共用群組移至目的地之前，會使用共用傳輸佇列來儲存訊息。

它是一個共用佇列，可供佇列共用群組中的所有佇列管理程式存取。

觸發

已觸發的共用佇列可以針對滿足的觸發條件產生多個觸發訊息。在與觸發的共用佇列相關聯的佇列共用群組中，佇列管理程式上定義的每一個本端起始佇列都會產生一則觸發訊息。

對於分散式佇列，每個通道起始程式接收滿足的共享傳輸佇列觸發條件的觸發訊息。不過，實際上只有一個通道起始程式會處理觸發的啟動，而其他通道起始程式則會安全失敗。然後會以負載平衡啟動來啟動觸發通道 (請參閱第 780 頁的『[為具有佇列共用群組的 DQM 準備 IBM MQ for z/OS](#)』) 觸發以啟動通道 QSG.TO.QM2。若要建立共用傳輸佇列，請使用 IBM MQ 指令 (MQSC)，如下列範例所示：

```
DEFINE QLOCAL(QM2) DESCR('Transmission queue to QM2') +
USAGE(XMITQ) QSGDISP(SHARED) +
CFSTRUCT(APPLICATION1) INITQ(SYSTEM.CHANNEL.INITQ) +
TRIGGER TRIGDATA(QSG.TO.QM2)
```

註：如果已設定共用佇列來觸發，且遺失與管理共用佇列之「連結機能」的連線，則可能會產生觸發事件，並將訊息放入起始佇列。即使未將任何訊息放置到原始共用佇列設定來進行觸發，也可能會發生這種情況。這是由 IXLVECTR 巨集的字元過度指示所造成，如 [清單通知向量](#)中所記載。

z/OS 佇列共用群組的訊息通道代理程式

只有在通道起始程式可以存取具有該名稱之通道的通道定義時，才能在通道起始程式上啟動通道。

訊息通道代理程式是一個 IBM MQ 程式，可控制訊息的傳送及接收。訊息通道代理程式會將訊息從一個佇列管理程式移至另一個佇列管理程式；通道的每一端都有一個訊息通道代理程式。

通道定義可以定義為專用於佇列管理程式，或儲存在共用儲存庫上且可在任何地方使用 (群組定義)。這表示在佇列共用群組中的任何通道起始程式上都可以使用群組定義的通道。

註：可以變更或刪除群組定義的專用副本。

若要建立群組通道定義，請使用 IBM MQ 指令 (MQSC)，如下列範例所示：

```
DEFINE CHL(QSG.TO.QM2) CHLTYPE(SDR) +
TRPTYPE(TCP) CONNAME(QM2.MACH.IBM.COM) +
XMITQ(QM2) QSGDISP(GROUP)
```

```
DEFINE CHL(QM2.TO.QSG) CHLTYPE(RCVR) TRPTYPE(TCP) +
QSGDISP(GROUP)
```

您可以從兩個視景來查看佇列共用群組的分散式佇列所使用的訊息通道代理程式：

入埠

如果入埠通道透過群組接聽器連接至佇列管理程式，則它是共用通道。它會透過一般介面連接至佇列共用群組，然後導向至群組內的佇列管理程式，或以特定佇列管理程式的群組埠或群組接聽器所使用的 luname 為目標。

出埠

如果從共用傳輸佇列移動訊息，則出埠通道是共用通道。在範例指令中，傳送端通道 QSG.TO.QM2 是共用通道，因為其傳輸佇列 QM2 是以 QSGDISP (SHARED) 定義。

佇列共用群組的同步化佇列

共用通道有自己稱為 SYSTEM.QSG.CHANNEL.SYNCQ。

佇列共用群組的任何成員都可以存取此同步化佇列。(專用通道會繼續使用專用同步化佇列。請參閱第 760 頁的『定義 IBM MQ 物件』)。這表示在通訊子系統、通道起始程式或佇列管理程式失敗時，可以在佇列共用群組內的不同佇列管理程式及通道起始程式實例上重新啟動通道。如需進一步資訊，請參閱第 780 頁的『為具有佇列共用群組的 DQM 準備 IBM MQ for z/OS』。

具有佇列共用群組的 DQM 需要名為 SYSTEM.QSG.CHANNEL.SYNCQ。此佇列必須可用，群組接聽器才能順利啟動。

如果群組接聽器因為佇列無法使用而失敗，則可以定義佇列，且可以重新啟動接聽器，而無需重新啟動通道起始程式。非共用通道不受影響。

請確定您使用 INDXTYPE (MSGID) 來定義此佇列。此定義可提高存取佇列上訊息的速度。

叢集和佇列共用群組

您可以在單一定義中使共用佇列可供叢集使用。若要這樣做，您可以在定義共用佇列時指定叢集的名稱。

網路中的使用者會將共用佇列視為由佇列共用群組內的每一個佇列管理程式所管理。(共用佇列未通告為由佇列共用群組所管理)。用戶端可以啟動與佇列共用群組所有成員的階段作業，以將訊息放入相同的共用佇列。

如需相關資訊，請參閱第 232 頁的『配置佇列管理程式叢集』。

通道和序列化

在共用佇列同層級回復期間，處理共用佇列上訊息的訊息通道代理程式會序列化其對佇列的存取權。

當訊息通道代理程式處理一或多個共用佇列上未確定的訊息時，如果佇列共用群組中的佇列管理程式失敗，通道和相關聯的通道起始程式會結束，且佇列管理程式會進行共用佇列同層級回復。

因為共用佇列同層級回復是非同步活動，所以在共用佇列同層級回復完成之前，同層級通道回復可能會嘗試同步重新啟動佇列共用群組的另一個部分中的通道。如果發生此事件，可能會在仍在回復的訊息之前先處理已確定的訊息。為了確保不會以這種方式不按順序處理訊息，處理共用佇列上的訊息的訊息通道代理程式會序列化其對這些佇列的存取權。

嘗試啟動共用佇列同層級回復仍在進行中的通道，可能會導致失敗。會發出一則錯誤訊息，指出正在進行回復，且通道會進入重試狀態。一旦佇列管理程式同層級回復完成，即可在下一次重試時重新啟動通道。

嘗試 RESOLVE、PING 或 DELETE 通道可能會因相同原因而失敗。

使用佇列共用群組來設定 IBM MQ for z/OS 的通訊

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。為了讓此嘗試成功，必須定義連線並使其可用。

從可使用的兩種通訊協定形式之一進行選擇：

- [TCP](#)
- [LU 6.2 到 APPC/MVS](#)

您可能會發現參照 [範例配置- IBM MQ for z/OS 使用佇列共用群組](#) 會很有用。

z/OS 定義佇列共用群組的 TCP 連線

若要定義佇列共用群組的 TCP 連線，必須配置傳送及接收端上的特定屬性。

如需設定 TCP 的相關資訊，請參閱 [第 777 頁的『在 z/OS 上定義 TCP 連線』](#)。

傳送端

通道定義中用來連接至佇列共用群組的連線名稱 (CONNAME) 欄位必須設為佇列共用群組的通用介面 (請參閱 [佇列共用群組](#))。如需詳細資料，請參閱 [使用 Sysplex Distributor](#)。

使用佇列共用群組在 TCP 上接收

接收共用通道程式是為了回應來自傳送通道的啟動要求而啟動。若要這樣做，必須啟動接聽器，以偵測送入的網路要求並啟動相關聯的通道。您可以使用 START LISTENER 指令、使用群組的入埠處置或使用作業及控制台來啟動此接聽器程式。

佇列共用群組中的所有群組接聽器都必須在相同的埠上接聽。如果您在單一 MVS 映像檔上執行多個通道起始程式，則可以定義虛擬 IP 位址，並啟動 TCP 接聽器程式，透過在 START LISTENER 指令中指定 IPADDR，僅在特定位址或主機名稱上接聽。(如需相關資訊，請參閱 [START LISTENER](#)。)

z/OS 在 z/OS 上定義 LU 6.2 連線

若要定義佇列共用群組的 LU 6.2 連線，必須配置傳送及接收端上的特定屬性。

如需設定 APPC/MVS 的相關資訊，請參閱 [設定 z/OS 的通訊](#)。

連接至 APPC/MVS (LU 6.2)

連接至佇列共用群組的通道定義中的連線名稱 (CONNAME) 欄位必須設為符號式目的地名稱，如 APPC/MVS 的端資訊資料集中所指定。在這個符號式目的地中指定的友機 LU 必須是同屬資源名稱。如需詳細資料，請參閱 [使用一般資源向網路定義您自己](#)。

使用通用介面在 LU 6.2 上接收

啟動接收共用 MCA，以回應來自傳送端通道的啟動要求。若要這樣做，必須啟動群組接聽器程式，以偵測送入的網路要求並啟動相關聯的通道。接聽器程式是 APPC/MVS 伺服器。您可以使用 START LISTENER 指令、使用入埠處置群組或使用作業及控制台來啟動它。您必須指定 LU 名稱，才能使用週邊資訊資料集中定義的符號式目的地名稱。如需詳細資料，請參閱 [使用一般資源向網路定義您自己](#)。

z/OS 將 IBM MQ 與 IMS 搭配使用

IBM MQ -IMS 配接器和 IBM MQ - IMS 橋接器是兩個元件，可讓 IBM MQ 與 IMS 互動。

若要配置 IBM MQ 和 IMS 一起運作，您必須完成下列作業：

- [第 785 頁的『設定 IMS 配接器』](#)
- [第 790 頁的『設定 IMS 橋接器』](#)

相關概念

IBM MQ 及 IMS

[第 791 頁的『將 IBM MQ 與 CICS 搭配使用』](#)

如果要將 IBM MQ 與 CICS 搭配使用，您必須配置 IBM MQ CICS 配接器，並選擇性地配置 IBM MQ CICS bridge 元件。

[第 793 頁的『在 IMS 中使用 OTMA 結束程式』](#)

如果您想要使用 IMS Open Transaction Manager Access exits with IBM MQ for z/OS, 請使用這個主題。

IBM MQ for z/OS 上的 IMS 及 IMS 橋接器應用程式

相關工作

第 693 頁的『在 z/OS 上配置佇列管理程式』

使用這些指示, 在「IBM MQ for z/OS」上配置佇列管理程式。

相關參考

第 791 頁的『升級及套用服務至 Language Environment 或 z/OS Callable Services』

您必須採取的動作會因您使用 CALLLIBS 或 LINK, 以及您的 SMP/E 版本而有所不同。

z/OS 設定 IMS 配接器

若要在 IMS 內使用 IBM MQ, 需要 IBM MQ - IMS 配接器 (通常稱為 IMS 配接器)。

本主題告訴您如何使 IMS 配接卡可供 IMS 子系統使用。如果您不熟悉自訂 IMS 子系統, 請參閱 *IBM Documentation* 中的 *IMS* 資訊。

若要讓 IMS 配接器可供 IMS 應用程式使用, 請遵循下列步驟:

1. 使用 IMS 外部子系統連接機能 (ESAF), 將 IBM MQ 定義為 IMS 作為外部子系統。

請參閱 第 786 頁的『將 IBM MQ 定義為 IMS』。

2. 在 IMS 控制區域及連接至 IBM MQ 之任何相依區域 (如果不在 LPA 或鏈結清單中) 的 JCL 中, 將 IBM MQ 載入程式庫 thlqual.SCSQAUTH 併入 JOBLIB 或 STEPLIB 連結中。如果您的 JOBLIB 或 STEPLIB 未獲授權, 也請在包含 IMS 模組 (通常是 IMS RESLIB) 的程式庫之後, 將它包含在 DFSESL 連結中。

也包括 thlqual.SCSQANLx (其中 x 是語言字母)。

如果 DFSESL 存在, 則 SCSQAUTH 及 SCSQANLx 需要併入連結中, 或新增至 LNKLIST。在 JCL 中新增至 STEPLIB 或 JOBLIB 連結並不足夠。

3. 將 IBM MQ 組程式 CSQQDEFV 從 thlqual.SCSQASMS 複製到使用者程式庫。

4. 所提供的程式 CSQQDEFV 包含一個子系統名稱 CSQ1, 以 MQM1 的 IMS 語言介面記號 (LIT) 識別為預設值。您可以保留此名稱, 以進行測試及安裝驗證。

若為正式作業子系統, 請將 NAME=CSQ1 變更為您自己的子系統名稱, 或使用 CSQ1。您可以視需要新增進一步子系統定義。如需 LIT 的進一步資訊, 請參閱 第 789 頁的『將 IBM MQ 佇列管理程式定義至 IMS 配接器』。

5. 組合並鏈結編輯程式, 以產生 CSQQDEFV 載入模組。對於組合, 請在 SYSLIB 連結中包含程式庫 thlqual.SCSQMACS; 使用鏈結編輯參數 RENT。這會顯示在 thlqual.SCSQPROC(CSQ4DEFV) 中的範例 JCL 中。

6. 針對連接至 IBM MQ 的任何相依區域, 在 JCL 中包含您在 JOBLIB 或 STEPLIB 連結中建立的模組 CSQQDEFV 的使用者程式庫。將此檔案庫放在 SCSQAUTH 之前, 因為 SCSQAUTH 具有預設載入模組。如果您不這麼做, 則會收到來自 IMS 的使用者 3041 異常終止。

7. 如果 IMS 配接器偵測到非預期的 IBM MQ 錯誤, 它會對 DD 名稱 CSQSNAP 發出 z/OS SNAP 傾出, 並對應用程式發出原因碼 MQRC_UNEXPECTED_ERROR。如果 CSQSNAP DD 陳述式不在 IMS 相依區域 JCL 中, 則不會執行傾出。如果發生這種情況, 您可以在 JCL 中併入 CSQSNAP DD 陳述式, 然後重新執行應用程式。不過, 由於部分問題可能是間歇性, 建議您併入 CSQSNAP DD 陳述式, 以擷取發生失敗的原因。

8. 如果您想要使用動態 IBM MQ 呼叫 (如 [動態呼叫 IBM MQ Stub](#) 中所述), 請建置動態 Stub, 如 第 786 頁的圖 116 所示。

9. 如果您想要使用 IMS 觸發監視器, 請定義 IMS 觸發監視器應用程式 CSQQTRMN, 並執行 PSBGEN 及 ACBGEN。請參閱第 790 頁的『設定 IMS 觸發監視器』。

10. 如果您使用 RACF 來保護 OPERCMDS 類別中的資源, 請確定與 IBM MQ 佇列管理程式位址空間相關聯的使用者 ID 有權向它可能連接的任何 IMS 系統發出 MODIFY 指令。

```

//DYNSTUB EXEC PGM=IEWL,PARM='RENT,REUS,MAP,XREF'
//SYSPRINT DD SYSOUT=*
//ACSQMOD DD DISP=SHR,DSN=thlqual.SCSQLOAD
//IMSLIB DD DISP=SHR,DSN=ims.reslib
//SYSLMOD DD DISP=SHR,DSN=private.load1
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,1)
//SYSLIN DD *
INCLUDE ACSQMOD(CSQSTUB)
INCLUDE IMSLIB(DFSLI000)
ALIAS MQCONN,MQCONN,MQDISC MQI entry points
ALIAS MQGET,MQPUT,MQPUT1 MQI entry points
ALIAS MQOPEN,MQCLOSE MQI entry points
ALIAS MQBACK,MQCMIT MQI entry points
ALIAS CSQBBAK,CSQBCMT MQI entry points
ALIAS MQINQ,MQSET MQI entry points
ALIAS DFSPLI,PLITDLI IMS entry points
ALIAS DFSCOBOL,CBLTDLI IMS entry points
ALIAS DFSFOR,FORTDLI IMS entry points
ALIAS DFSASM,ASMTDLI IMS entry points
ALIAS DFSPASCL,PASTDLI IMS entry points
ALIAS DFHEI01,DFHEI1 IMS entry points
ALIAS DFSAIBLI,AIBTDLI IMS entry points
ALIAS DFSESS,DSNWLI,DSNHLI IMS entry points
ALIAS MQCRTMH,MQDLTMH,MQDLTMP IMS entry points
ALIAS MQINQMP,MQSETMP,MQMHBUF,MQBUFMH IMS entry points
MODE AMODE(31),RMODE(24) Note RMODE setting
NAME CSQDYNS(R)
/*

1Specify the name of a library accessible to IMS applications that
want to make dynamic calls to IBM MQ.

```

圖 116: 用來鏈結-編輯動態呼叫 Stub 的 JCL 範例

相關概念

IBM MQ 及 IMS

第 790 頁的『設定 IMS 橋接器』

IBM MQ - IMS 橋接器是選用元件，可讓 IBM MQ 在未啟用 IBM MQ 的現有程式及交易中輸入及輸出。

IBM MQ for z/OS 上的 IMS 及 IMS 橋接器應用程式

z/OS 將 IBM MQ 定義為 IMS

IBM MQ 必須定義至 IMS 控制區域，以及每一個存取該 IBM MQ 佇列管理程式的相依區域。若要這樣做，您必須在 IMS 中建立子系統成員 (SSM)。PROCLIB 程式庫，並向適用的 IMS 區域識別 SSM。

將子系統成員登錄置於 IMS 中。PROCLIB

IMS 中的每一個 SSM 項目。PROCLIB 定義從 IMS 區域到不同佇列管理程式的連線。

若要命名 SSM，請將 IMS IMSCTRL 巨集的 IMSID 欄位值 (1 到 4 個英數字元) 與您網站所定義的任何名稱 (1 到 4 個英數字元) 連結。

一個 SSM 可以由所有 IMS 區域共用，或者可以為每個區域定義特定成員。此成員包含與外部子系統連線數一樣多的項目。每一個項目都是 80 個字元的記錄。

位置參數

此項目中的欄位如下：

```
SSN,LIT,ESMT,RTT,REO,CRC
```

其中：

SSN

指定 IBM MQ 佇列管理程式名稱。它是必要項目，且必須包含 1 到 4 個字元。

lit

指定提供給 IMS 的語言介面記號 (LIT)。這是必要欄位，其值必須符合 CSQQDEFV 模組中的值。

ESMT

指定外部子系統模組表格 (ESMT)。此表格指定 IMS 必須載入哪些附件模組。CSQQESMT 是此欄位的必要值。

RTT

IBM MQ 不支援此選項。

REO

指定在 IMS 應用程式參照無法運作的外部子系統時，或在建立執行緒時無法使用資源時，要使用的區域錯誤選項 (REO)。此欄位是選用的，且包含單一字元，可以是：

R

將回覆碼傳遞至應用程式，指出 IBM MQ 服務的要求失敗。

Q

以異常終止碼 U3051 結束應用程式，將活動取消至最後一個確定點，執行交易的 PSTOP，並將輸入訊息重新排入佇列。只有在 IMS 應用程式嘗試參照無法運作的外部子系統時，或在建立執行緒時無法使用資源時，此選項才適用。

如果在 IBM MQ 處理要求時發生 IBM MQ 問題，則會將 IBM MQ 完成及原因碼傳回給應用程式；亦即，在配接器將要求傳遞至 IBM MQ 之後。

A

以異常終止碼 U3047 結束應用程式，並捨棄輸入訊息。只有在 IMS 應用程式參照無法運作的外部子系統時，或在建立執行緒時無法使用資源時，此選項才適用。

如果在 IBM MQ 處理要求時發生 IBM MQ 問題，則會將 IBM MQ 完成及原因碼傳回給應用程式；亦即，在配接器將要求傳遞至 IBM MQ 之後。

CRC

可以指定此選項，但 IBM MQ 不會使用此選項。

註：如需所有位置參數的完整資料，請參閱 [如何將外部子系統指定給 IMS](#)。

SSM 項目範例如下：

```
CSQ1, MQM1, CSQQESMT, , R,
```

其中：

CSQ1	隨 IBM MQ 提供的預設子系統名稱。您可以變更此項以符合您的安裝。
MQM1	CSQQDEFV 中提供的預設 LIT。
CSQQESMT	外部子系統模組名稱。您必須使用此值。
R	REO 選項。

關鍵字參數

IBM MQ 參數可以關鍵字格式指定。SST 參數的值可以是 DB2 或 MQ。在 IMS 14 中新增了對 MQ 值的支援。使用 MQ 可協助釐清，IMS 子系統指令現在包括 SST 值，但否則沒有任何顯著效果。必要的話，仍可以使用 DB2 值。其他參數如 [位置參數](#) 中所述，如下列範例所示：

```
SST=MQ, SSN=SYS3, LIT=MQM3, ESMT=CSQQESMT
```

其中：

SYS3	子系統名稱
MQM3	CSQQDEFV 中提供的 LIT
CSQESMT	外部子系統模組名稱

指定 SSM EXEC 參數

在 IMS 控制區域的啟動程序中指定 SSM EXEC 參數。此參數指定一個字元到四個字元的子系統成員名稱 (SSM)。

如果您為 IMS 控制區域指定 SSM，則在控制區域下執行的任何相依區域都可以連接至 IMS 中指定的 IBM MQ 佇列管理程式。SSM 參數指定的 PROCLIB 成員。IMS。PROCLIB 成員名稱是與 SSM EXEC 參數中指定的一到四個字元連結的 IMS ID (IMSID= *xxxx*)。IMS ID 是 IMSCTRL 產生巨集的 IMSID 參數。

IMS 可讓您定義所需數目的外部子系統連線。可以為不同的 IBM MQ 佇列管理程式定義多個連線。所有 IBM MQ 連線都必須位於相同的 z/OS 系統內。對於相依區域，您可以指定相依區域 SSM，或使用針對控制區域指定的相依區域 SSM。您可以在相依區域 SSM 及控制區域 SSM 中指定不同的區域錯誤選項 (REO)。第 788 頁的表 66 顯示 SSM 規格的不同可能性。

控制區域的 SSM	相依區域的 SSM	動作	Comments
否	否	無	無法連接外部子系統。
否	是	無	無法連接外部子系統。
是	否	使用控制區域 SSM	在區域中排程的應用程式可以存取在控制區域 SSM 中識別的外部子系統。每一個附件的結束程式及控制區塊會載入控制區域及相依區域位址空間中。
是	是 (空白)	沒有 SSM 用於相依區域	在此區域中排程的應用程式只能存取 DL/I 資料庫。每個附件的結束程式和控制區塊會載入控制區域位址空間中。
是	是 (不是空的)	使用控制區域 SSM 檢查相依區域 SSM	在此區域中排程的應用程式只能存取兩個 SSM 中所識別的外部子系統。每一個附件的結束程式及控制區塊會載入控制區域及相依區域位址空間中。

沒有特定參數可控制 SSM 規格可能性的數目上限。

預載 IMS 配接器

如果 IMS 已預載 IMS 配接器，則可以改善它的效能。Preloading is controlled by the DFSMPLxx member of IMS.PROCLIB: see "IMS Administration Guide: System" for more information. 要指定的 IBM MQ 模組名稱如下：

CSQACLST	CSQAMLST	CSQAPRH	CSQAVICM	CSQFSALM	CSQQDEFV
CSQQCONN	CSQQDISC	CSQQTERM	CSQQINIT	CSQQBACK	CSQQCMMT
CSQESMT	CSQQPREP	CSQQTTHD	CSQQWAIT	CSQQNORM	CSQQSSOF
CSQQSSON	CSQFSTAB	CSQQRESV	CSQQSNOP	CSQQCMND	CSQQCVER
CSQQTMID	CSQQTRGI	CSQQCON2	CSQBPAPI	CSQBCRMH	CSQBAPPL

如需使用 IBM MQ classes for JMS 的相關資訊，請參閱在 IMS 中使用 IBM MQ classes for JMS。

IMS 的現行版本僅支援從 MPP、BMP、IFP、JMP 及 JBP 區域中的 PDS-E 格式程式庫預先載入 IBM MQ 模組。任何其他類型的 IMS 區域都不支援從 PDS-E 程式庫預先載入。如果任何其他類型的區域需要預載，則提供的 IBM MQ 模組必須複製到 PDS 格式檔案庫。

z/OS 將 IBM MQ 佇列管理程式定義至 IMS 配接器

IBM MQ 佇列管理程式及其對應語言介面記號 (LIT) 的名稱必須定義在佇列管理程式定義表格中。

使用提供的 CSQQDEFFX 巨集來建立 CSQQDEFV 載入模組。第 789 頁的圖 117 顯示此組譯器巨集的語法。

```
CSQQDEFX TYPE=ENTRY | DEFAULT, NAME=qmgr-name, LIT=token  
or  
CSQQDEFX TYPE=END
```

圖 117: CSQQDEFX 巨集語法

參數

TYPE=ENTRY | DEFAULT

指定 TYPE=ENTRY 或 TYPE=DEFAULT，如下所示：

TYPE=ENTRY

指定要產生說明 IMS 應用程式可用的 IBM MQ 佇列管理程式的表格項目。如果這是第一個項目，則也會產生表格標頭，包括 CSQQDEFV CSECT 陳述式。

TYPE=DEFAULT

至於 TYPE=ENTRY。指定的佇列管理程式是當 MQCONN 或 MQCONNX 指定的名稱都是空白時要使用的預設佇列管理程式。表格中只能有一個這類項目。

NAME= qmgr-name

指定佇列管理程式的名稱，如 MQCONN 或 MQCONNX 所指定。

LIT = token

指定 IMS 用來識別佇列管理程式的語言介面記號 (LIT) 名稱。

MQCONN 或 MQCONNX 呼叫會將名稱輸入參數及 hconn 輸出參數與名稱標籤相關聯，因此會與 CSQQDEFV 項目中的 LIT 相關聯。傳遞 hconn 參數的進一步 IBM MQ 呼叫會使用 MQCONN 或 MQCONNX 呼叫中所識別的 CSQQDEFV 項目中的 LIT，來直接呼叫具有相同 LIT 的 IMS SSM PROCLIB 成員中所定義的 IBM MQ 佇列管理程式。

總而言之，MQCONN 或 MQCONNX 呼叫的 **name** 參數會識別 CSQQDEFV 中的 LIT，而 SSM 成員中的相同 LIT 會識別 IBM MQ 佇列管理程式。(如需 MQCONN 呼叫的相關資訊，請參閱 [MQCONN-Connect 佇列管理程式](#)。如需 MQCONNX 呼叫的相關資訊，請參閱 [MQCONNX-Connect 佇列管理程式 \(延伸\)](#)。)

TYPE=END

指定表格已完成。如果省略此參數，則會採用 TYPE=ENTRY。

使用 CSQQDEFX 巨集

第 789 頁的圖 118 顯示佇列管理程式定義表的一般佈置。

```
CSQQDEFX NAME=subsystem1, LIT=token1  
CSQQDEFX NAME=subsystem2, LIT=token2, TYPE=DEFAULT  
CSQQDEFX NAME=subsystem3, LIT=token3  
...  
CSQQDEFX NAME=subsystemN, LIT=tokenN  
CSQQDEFX TYPE=END  
END
```

圖 118: 佇列管理程式定義表格的佈置

z/OS 設定 IMS 觸發監視器

您可以設定 IMS 批次導向程式來監視 IBM MQ 起始佇列。

使用 thlqual.SCSQPROC 程式庫中的模型 CSQQTAPL，將應用程式定義至 IMS (請參閱 [CSQQTRMN 的交易定義範例](#))。

使用 thlqual.SCSQPROC 程式庫中的模型 CSQQTPSB 來產生 PSB 和 ACB (請參閱 [CSQQTRMN 的 PSB 定義範例](#))。

```
* This is the application definition *
* for the IMS Trigger Monitor BMP      *

APPLCTN PSB=CSQQTRMN,
PGMTYPE=BATCH,
SCHDTYP=PARALLEL
```

圖 119: CSQQTRMN 的交易定義範例

```
PCB TYPE=TP,           ALTPCB for transaction messages
MODIFY=YES,           To "triggered" IMS transaction
PCBNAME=CSQQTRMN
PCB TYPE=TP,           ALTPCB for diagnostic messages
MODIFY=YES,           To LTERM specified or "MASTER"
PCBNAME=CSQQTRMG,
EXPRESS=YES
PSBGEN LANG=ASSEM,
PSBNAME=CSQQTRMN,    Runs program CSQQTRMN
CMPAT=YES
```

圖 120: CSQQTRMN 的 PSB 定義範例

如需啟動及停止 IMS 觸發監視器的進一步資訊，請參閱 [控制 IMS 觸發監視器](#)。

z/OS 設定 IMS 橋接器

IBM MQ - IMS 橋接器是選用元件，可讓 IBM MQ 在未啟用 IBM MQ 的現有程式及交易中輸入及輸出。

本主題說明自訂 IBM MQ - IMS 橋接器必須執行的動作。

定義 IBM MQ 的 XCF 和 OTMA 參數。

此步驟定義 IBM MQ 系統的 XCF 群組和成員名稱，以及其他 OTMA 參數。IBM MQ 和 IMS 必須屬於相同的 XCF 群組。使用 CSQ6SYSP 巨集的 OTMACON 關鍵字來修改系統參數載入模組中的這些參數。

如需相關資訊，請參閱 [使用 CSQ6SYSP](#)。

定義 XCF 和 OTMA 參數給 IMS。

此步驟定義 IMS 系統的 XCF 群組及成員名稱。IMS 和 IBM MQ 必須屬於相同的 XCF 群組。

在 JCL 中，或在 IMS PROCLIB 中的成員 DFSPBxxx 中，將下列參數新增至 IMS 參數清單：

OTMA=Y

這會在 IMS 啟動時自動啟動 OTMA。(這是選用項目，如果您指定 OTMA=N，也可以發出 IMS 指令 /START OTMA 來啟動 OTMA。)

GRNAME=

此參數提供 XCF 群組名稱。

它與儲存類別定義中指定的群組名稱相同 (請參閱下一步)，也與 CSQ6SYSP 巨集之 OTMACON 關鍵字的 **Group** 參數相同。

OTMANM=

此參數提供 IMS 系統的 XCF 成員名稱。

這與儲存類別定義中指定的成員名稱相同 (請參閱下一步)。

告知 IBM MQ IMS 系統的 XCF 群組及成員名稱。

這是由佇列的儲存類別指定。如果您想要透過 IBM MQ - IMS 橋接器傳送訊息，則必須在定義佇列的儲存類別時指定此項。在儲存類別中，您必須定義 XCF 群組及目標 IMS 系統的成員名稱。若要執行此動作，請使用 IBM MQ 作業及控制台，或使用 IBM MQ 指令，如 [可程式指令格式簡介](#) 中所述。

設定您需要的安全。

/SECURE OTMA IMS 指令決定要套用至 **每一個** IBM MQ 佇列管理程式 (透過 OTMA 連接至 IMS) 的安全等級。如需相關資訊，請參閱 [搭配使用 IBM MQ 與 IMS 的安全考量](#)。

將其他 IMS 連線新增至相同的佇列管理程式

若要將 IMS 連線新增至相同的佇列管理程式，您必須：

- 定義第二個儲存類別 [STGCLASS](#) 以指向新的 IMS；如需相關資訊，請參閱 [DEFINE STGCLASS](#)。
- 新增本端佇列以指向第二個儲存類別。

重要：

- 一個本端佇列不能指向兩個儲存類別。
- 一個儲存類別不能指向兩個 IMS 橋接器。
- IBM MQ 和 IMS 必須屬於相同的 XCF 群組。使用 CSQ6SYSP 巨集的 OTMACON 關鍵字來修改系統參數載入模組中的這些參數。

如需相關資訊，請參閱 [使用 CSQ6SYSP](#)。

相關概念

IBM MQ 及 IMS

第 785 頁的『[設定 IMS 配接器](#)』

若要在 IMS 內使用 IBM MQ，需要 IBM MQ - IMS 配接器 (通常稱為 IMS 配接器)。

[IBM MQ for z/OS 上的 IMS 及 IMS 橋接器應用程式](#)

z/OS 將 IBM MQ 與 CICS 搭配使用

如果要將 IBM MQ 與 CICS 搭配使用，您必須配置 IBM MQ CICS 配接器，並選擇性地配置 IBM MQ CICS bridge 元件。

如需配置 IBM MQ CICS 配接器和 IBM MQ CICS bridge 元件的相關資訊，請參閱 CICS 說明文件的 [配置 MQ](#) 一節。

相關概念

IBM MQ 及 CICS

第 784 頁的『[將 IBM MQ 與 IMS 搭配使用](#)』

IBM MQ -IMS 配接器和 IBM MQ - IMS 橋接器是兩個元件，可讓 IBM MQ 與 IMS 互動。

相關參考

第 791 頁的『[升級及套用服務至 Language Environment 或 z/OS Callable Services](#)』

您必須採取的動作會因您使用 CALLLIBS 或 LINK，以及您的 SMP/E 版本而有所不同。

z/OS 升級及套用服務至 Language Environment 或 z/OS Callable Services

您必須採取的動作會因您使用 CALLLIBS 或 LINK，以及您的 SMP/E 版本而有所不同。

下表顯示如果您升級下列產品的層次，或將服務套用至下列產品，您需要對 IBM MQ for z/OS 執行的動作：

- 語言環境
- z/OS 可呼叫服務 (例如 APPC 和 RRS)

表 67: 已套用服務或產品已升級至新版次

產品	使用 CALLLIBS 和 SMP/E V3r2 或更新版本時的動作 註: 您不需要針對 Language Environment 及 Callable 服務執行個別工作。一份工作就夠了	使用 LINK 時的動作
語言環境	<ol style="list-style-type: none"> 將 SMP/E 工作上的界限設為「目標」區域。 在 SMP_CNTL 卡上指定 LINK LMODS CALLLIBS。您也可以指定其他參數, 例如 CHECK、RETRY (YES) 及 RC。如需進一步資訊, 請參閱 <i>SMP/E for z/OS</i>: 指令。 執行 SMP/E 工作。 	如果已設定 SMP/E 區域進行自動重新鏈結, 且已執行 CSQ8SLDQ 工作, 則不需要採取任何動作。
可呼叫服務	<ol style="list-style-type: none"> 將 SMP/E 工作上的界限設為「目標」區域。 在 SMP_CNTL 卡上指定 LINK LMODS CALLLIBS。您也可以指定其他參數, 例如 CHECK、RETRY (YES) 及 RC。如需進一步資訊, 請參閱 <i>SMP/E for z/OS</i>: 指令。 執行 SMP/E 工作。 	如果已設定 SMP/E 區域進行自動重新鏈結, 且已執行 CSQ8SLDQ 工作, 則不需要採取任何動作。

表 68: 其中一個產品已在新的 SMP/E 環境及檔案庫中更新為新版本

產品	使用 CALLLIBS 和 SMP/E V3r2 或更新版本時的動作 註: 您不需要針對 Language Environment 及 Callable 服務執行三個個別工作。一項工作就足以滿足兩種產品的需求。	使用 LINK 時的動作
語言環境	<ol style="list-style-type: none"> 變更 SCEELKED 及 SCEESPC 的 DDDEF, 以指向新的磁帶庫。 將 SMP/E 工作上的界限設為「目標」區域。 在 SMP_CNTL 卡上指定 LINK LMODS CALLLIBS。您也可以指定其他參數, 例如 CHECK、RETRY (YES) 及 RC。如需進一步資訊, 請參閱 <i>SMP/E for z/OS</i>: 指令。 執行 SMP/E 工作。 	<ol style="list-style-type: none"> 刪除 IBM MQ for z/OS 目標區域中下列 LMOD 項目的 XZMOD 子項目: CMQXDCST、CMQXRCTL、CMQXSUPR、CSQCBE00、CSQCBE30、CSQCBP00、CSQCBP10、CSQCBR00、CSQUCVX、CSQUDLQH、CSQVXPCB、CSQVXSPT、CSQXDCST、CSQXRCTL、CSQXSUPR、CSQXTCMI、CSQXTCP、CSQXTNSV、CSQ7DRPS、IMQB23IC、IMQB23IM、IMQB23IR、IMQS23IC、IMQS23IM、IMQS23IR 在 IBM MQ 區域與 Language Environment 區域之間設定適當的 ZONEINDEX。 修改 CSQ8SLDQ 以參照 LINK 指令的 FROMZONE 參數上的新區域。CSQ8SLDQ 可在 SCSQINST 程式庫中找到。 執行 CSQ8SLDQ。

表 68: 其中一個產品已在新的 SMP/E 環境及檔案庫中更新為新版本 (繼續)

產品	使用 CALLIBS 和 SMP/E V3r2 或更新版本時的動作 註: 您不需要針對 Language Environment 及 Callable 服務執行三個個別工作。一項工作就足以滿足兩種產品的需求。	使用 LINK 時的動作
可呼叫服務	<ol style="list-style-type: none"> 將 CSSLIB 的 DDDEF 變更為指向新的程式庫 將 SMP/E 工作上的界限設為「目標」區域。 在 SMP_CNTL 卡上指定 LINK LMODS CALLIBS。您也可以指定其他參數, 例如 CHECK、RETRY (YES) 及 RC。如需進一步資訊, 請參閱 <i>SMP/E for z/OS</i>: 指令。 執行 SMP/E 工作。 	<ol style="list-style-type: none"> 刪除 IBM MQ for z/OS 目標區域中下列 LMOD 項目的 XZMOD 子項目: CMQXRCTL、CMQXSUPR、CSQBSRV、CSQILPLM、CSQXJST、CSQXRCTL、CSQXSUPR、CSQ3AMGP、CSQ3EPX、CSQ3REPL 在 IBM MQ 區域與「可呼叫服務」區域之間設定適當的 ZONEINDEX。 修改 CSQ8SLDQ 以參照 LINK 指令的 FROMZONE 參數上的新區域。CSQ8SLDQ 可在 SCSQINST 程式庫中找到。 執行 CSQ8SLDQ。

如需使用 CALLIBS 時重新鏈結模組的工作範例, 請參閱 第 793 頁的『執行 LINK CALLIBS 工作』。

z/OS 執行 LINK CALLIBS 工作

使用 CALLIBS 時重新鏈結模組的範例工作。

以下是在 SMP/E V3r2 系統上使用 CALLIBS 時, 重新鏈結模組的工作範例。您必須提供 JOBCARD 以及包含 IBM MQ for z/OS 的 SMP/E CSI 的資料集名稱。

```

//*****
//* RUN LINK CALLIBS.
//*****
//CALLIBS EXEC PGM=GIMSMP,REGION=4096K
//SMPCSI DD DSN=your.csi
//      DISP=SHR
//SYSPRINT DD SYSOUT=*
//SMPCNTL DD *
SET BDY(TZONE).
LINK LMODS CALLIBS .
/*
    
```

圖 121: 範例 SMP/E LINK CALLIBS 工作

z/OS 在 IMS 中使用 OTMA 結束程式

如果您想要使用 IMS Open Transaction Manager Access exits with IBM MQ for z/OS, 請使用這個主題。

如果您要將輸出從 IMS 交易傳送至 IBM MQ, 且該交易並非源自 IBM MQ, 則需要撰寫一個以上 IMS OTMA 結束程式的程式碼。

同樣地, 如果您想要將輸出傳送至非 OTMA 目的地, 且交易確實源自 IBM MQ, 則您也需要撰寫一個以上 IMS OTMA 結束程式的程式碼。

IMS 中提供下列結束程式, 可讓您自訂 IMS 與 IBM MQ 之間的處理:

- OTMA 預先遞送結束程式
- 目的地解析使用者 (DRU) 結束程式

OTMA 結束程式名稱

您必須將預先遞送結束程式命名為 DFSYPRX0。您可以將 DRU 命名為任何結束程式，只要它與 IMS 中已存在的模組名稱不衝突即可。

指定目的地解析使用者結束程式名稱

您可以使用 CSQ6SYSP 巨集之 OTMACON 關鍵字的 *Druexit* 參數，指定 IMS 要執行的 OTMA DRU 結束程式名稱。

若要簡化物件識別，請考慮採用 DRU0xxxx 的命名慣例，其中 xxxx 是 IBM MQ 佇列管理程式的名稱。

如果您未在 OTMACON 參數中指定 DRU 結束程式的名稱，則預設值為 DFSYDRU0。如需相關資訊，請參閱 [DFSYDRU0](#)。

IMS 目的地的命名慣例

您需要從 IMS 程式傳送輸出至目的地的命名慣例。這是在 IMS 應用程式的 CHNG 呼叫中設定的目的地，或在 IMS PSB 中預設的目的地。

OTMA 結束程式的範例實務

如需 IMS 的預先遞送結束程式及目的地遞送結束程式的範例，請使用下列主題：

- [第 794 頁的『預先遞送結束程式 DFSYPRX0』](#)
- [第 795 頁的『目的地解析使用者結束程式』](#)

若要簡化識別，請讓 OTMA 目的地名稱與 IBM MQ 佇列管理程式名稱類似，例如 IBM MQ 佇列管理程式名稱重複。在此情況下，如果 IBM MQ 佇列管理程式名稱是 "VCPE"，CHNG 呼叫所設定的目的地是 "VCPEVCPE"。

相關概念

IBM MQ 及 IMS

[第 784 頁的『將 IBM MQ 與 IMS 搭配使用』](#)

IBM MQ -IMS 配接器和 IBM MQ - IMS 橋接器是兩個元件，可讓 IBM MQ 與 IMS 互動。

[IBM MQ for z/OS 上的 IMS 及 IMS 橋接器應用程式](#)

預先遞送結束程式 DFSYPRX0

本主題包含 IMS 中 OTMA 的樣本預先遞送結束程式。

您必須先撰寫預先遞送結束程式 DFSYPRX0 的程式碼。請參閱 [OTMA 目的地解析使用者結束程式 \(DFSYPRX0 及其他 OTMAYPRX 類型結束程式\)](#)，以取得 IMS 傳遞至此常式的參數。

此結束程式會測試訊息是否預期用於已知 OTMA 目的地 (在我們的範例 VCPEVCPE 中)。如果是的話，結束程式必須檢查傳送訊息的交易是否源自 OTMA。如果訊息源自 OTMA，它會有 OTMA 標頭，因此您應該結束 DFSYPRX0，並將暫存器 15 設為零。

- 如果傳送訊息的交易不是源自 OTMA，您必須將用戶端名稱設為有效的 OTMA 用戶端。這是您要將訊息傳送至其中的 IBM MQ 佇列管理程式的 XCF 成員名稱。您應該將用戶端名稱 (在 CSQ6SYSP 巨集的 OTMACON 參數中) 設為佇列管理程式名稱。這是預設值。然後，您應該結束 DFSYPRX0 設定暫存器 15 到 4。
- 如果傳送訊息的交易是在 OTMA 中產生，且目的地是非 OTMA，您應該將暫存器 15 設為 8 並結束。
- 在所有其他情況下，您應該將暫存器 15 設為零。

如果您將 OTMA 用戶端名稱設為 IMS 不知道的名稱，您的應用程式 CHNG 或 ISRT 呼叫會傳回 A1 狀態碼。

對於與多個 IBM MQ 佇列管理程式進行通訊的 IMS 系統，您應該針對每一個 IBM MQ 佇列管理程式重複該邏輯。

第 795 頁的圖 122 中顯示組譯器程式碼範例：

```
TITLE 'DFSYPX0: OTMA PRE-ROUTING USER EXIT'
DFSYPX0 CSECT
DFSYPX0 AMODE 31
DFSYPX0 RMODE ANY
*
SAVE (14,12),,DFSYPX0&SYSDATE&SYSTIME
SPACE 2
LR R12,R15          MODULE ADDRESSABILITY
USING DFSYPX0,R12
*
L R2,12(,R1)       R2 -> OTMA PREROUTE PARMS
*
LA R3,48(,R2)      R3 AT ORIGINAL OTMA CLIENT (IF ANY)
CLC 0(16,R3),=XL16'00' OTMA ORIG?
BNE OTMAIN        YES, GO TO THAT CODE
*
NOOTMAIN DS 0H          NOT OTMA INPUT
LA R5,8(,R2)          R5 IS AT THE DESTINATION NAME
CLC 0(8,R5),=C'VCPEVCPE' IS IT THE OTMA UNSOLICITED DEST?
BNE EXIT0           NO, NORMAL PROCESSING
*
L R4,80(,R2)        R4 AT ADDR OF OTMA CLIENT
MVC 0(16,R4),=CL16'VCPE' CLIENT OVERRIDE
B EXIT4            AND EXIT
*
OTMAIN DS 0H          OTMA INPUT
LA R5,8(,R2)        R5 IS AT THE DESTINATION NAME
CLC 0(8,R5),=C'VCPEVCPE' IS IT THE OTMA UNSOLICITED DEST?
BNE EXIT8           NO, NORMAL PROCESSING

*
EXIT0 DS 0H
LA R15,0            RC = 0
B BYEBYE
*
EXIT4 DS 0H
LA R15,4            RC = 4
B BYEBYE
*
EXIT8 DS 0H
LA R15,8            RC = 8
B BYEBYE
*
BYEBYE DS 0H
RETURN (14,12),,RC=(15) RETURN WITH RETURN CODE IN R15
SPACE 2
REQUATE
SPACE 2
END
```

圖 122: OTMA 預先遞送結束程式組合器範例

▶ z/OS 目的地解析使用者結束程式

本主題包含 IMS 的目的地解析使用者結束程式範例。

如果您已在 DFSYPX0 中設定暫存器 15 到 4，或如果交易來源是 OTMA 且您將「暫存器 15」設為零，則會呼叫您的 DRU 結束程式。在此範例中，DRU 結束程式名稱為 DRUOVCPPE。

DRU 結束程式會檢查目的地是否為 VCPEVCPE。如果是的話，它會設定 OTMA 使用者資料 (在 OTMA 字首中) 如下：

偏移

OTMA 使用者資料

(十進位)

0

OTMA 使用者資料長度 (在此範例中，334)

2

MQMD

326

回覆格式

這些偏移是 IBM MQ - IMS 橋接器預期找到此資訊的位置。

DRU 結束程式應該盡可能簡單。因此，在此範例中，特定 IBM MQ 佇列管理程式的所有源自 IMS 的訊息都會放入相同的 IBM MQ 佇列。

如果訊息需要持續保存，IMS 必須使用同步化交易管道。若要這樣做，DRU 結束程式必須設定 OUTPUT 旗標。如需相關資訊，請參閱 [指定 IBM MQ 的 synchronized tp 管道](#)。

撰寫 IBM MQ 應用程式以處理此佇列，並使用 MQMD 結構、MQIIH 結構 (如果有的話) 或使用者資料中的資訊，將每一則訊息遞送至其目的地。

第 796 頁的圖 123 中顯示組譯器 DRU 結束程式範例。

```
TITLE 'DRU0VCPE: OTMA DESTINATION RESOLUTION USER EXIT'
DRU0VCPE CSECT
DRU0VCPE AMODE 31
DRU0VCPE RMODE ANY
*
SAVE (14,12),,DRU0VCPE&SYSDATE&SYSTIME
SPACE 2
LR R12,R15          MODULE ADDRESSABILITY
USING DRU0VCPE,R12
*
L R2,12(,R1)        R2 -> OTMA DRU PARMS
*
L R5,88(,R2)        R5 ADDR OF OTMA USERDATA
LA R6,2(,R5)        R6 ADDR OF MQMD
USING MQMD,R6       AS A BASE
*
LA R4,MQMD_LENGTH+10 SET THE OTMA USERDATA LEN
STH R4,0(,R5)       = LL + MQMD + 8
*
MVI 0(R6),X'00'     ...NULL FIRST BYTE
MVC 1(255,R6),0(R6) ...AND PROPAGATE IT
MVC 256(MQMD_LENGTH-256+8,R6),255(R6) ...AND PROPAGATE IT
*
VCPE DS 0H
CLC 44(16,R2),=CL16'VCPE' IS DESTINATION VCPE?
BNE EXIT4          NO, THEN DEST IS NON-OTMA
MVC MQMD_REPLYTOQ,=CL48'IMS.BRIDGE.UNSOLICITED.QUEUE'
MVC MQMD_REPLYTOQMGR,=CL48'VCPE' SET QNAME AND QMGRNAME
MVC MQMD_FORMAT,MQFMT_IMS SET MQMD FORMAT NAME
MVC MQMD_LENGTH(8,R6),MQFMT_IMS_VAR_STRING
*
B EXIT0            SET REPLYTO FORMAT NAME
*
EXIT0 DS 0H
LA R15,0           SET RC TO OTMA PROCESS
B BYEBYE          AND EXIT
*
EXIT4 DS 0H
LA R15,4           SET RC TO NON-OTMA
B BYEBYE          AND EXIT
*
BYEBYE DS 0H
RETURN (14,12),,RC=(15) RETURN CODE IN R15
SPACE 2
REQUATE
SPACE 2
CMQA EQUONLY=NO
CMQMDA DSECT=YES
SPACE 2
END
```

圖 123: 範例組譯器 DRU 結束程式

IBM z/OS Management Facility (z/OSMF) 在作業導向的 Web 瀏覽器型使用者介面中提供系統管理功能，並提供整合的使用者協助，讓您可以更輕鬆地管理大型主機 z/OS 系統的日常作業及管理。

透過簡化部分傳統作業並自動化其他作業，z/OSMF 可以協助簡化 z/OS 系統管理的部分區域。

只要按一下按鈕，即可從使用者提供的入口網站佈建或取消佈建資源。z/OSMF 提供 REST API 來協助執行這項作業。

z/OSMF 隨附的範例 Marketplace 入口網站也可以用來佈建及取消佈建資源。或者，更有經驗的使用者可以使用 z/OSMF Web 使用者介面 (WUI)。

本節假設您瞭解 z/OSMF，但如果您不熟悉 z/OSMF，則應該閱讀 [開始使用 z/OSMF](#)。或者，您可以從 z/OSMF WUI 線上說明存取此區段。

您應該熟悉 z/OS Cloud 配置，亦即：

- 雲端供應-資源管理服務
- 工作量管理-如需相關資訊，請參閱 [IBM z/OS Management Facility Programming Guide](#)。
- 入門-請參閱 [入門指導教學-雲端](#)

z/OSMF 2.2 引進角色型活動及作業，因此您必須瞭解概念，例如：

網域
管理者
核准者
租戶
範本
實例
工作流程

以此類推。

已提供範例 IBM MQ z/OSMF 工作流程及相關聯檔案，且可以安裝為「IBM MQ for z/OS UNIX 系統服務元件」特性的一部分。「IBM MQ for z/OS 程式目錄 IBM MQ for z/OS 的程式目錄可以從 IBM 出版品中心下載 (請參閱 [IBM MQ for z/OS 程式目錄 PDF 檔案](#))。」中說明此特性的安裝程序，以及目錄和檔案結構。

範例工作流程以 XML 撰寫，並示範如何自動佈建 (建立) 或取消佈建 (毀損) IBM MQ 佇列管理程式、通道起始程式及本端佇列，以及如何對佈建的 IBM MQ 資源執行動作。工作流程內的步驟會提交工作 (JCL)、執行 REXX 執行程式、處理程序 Shell Script 或發出 REST API 呼叫。

範例設計用來說明可使用 z/OSMF 達成的函數類型。預期通常會使用 z/OSMF 工作流程來佈建資源，而且基本上會使用 IBM MQ 應用程式來執行放置或取得訊息之類的動作。

您可以依提供的方式執行範例工作流程，前提是已設定工作流程變數內容 (如下列各節所述)，或者您可以視需要自訂它們。您可能偏好撰寫自己的工作流程來執行其他功能。在執行範例工作流程之前，請參閱：

- [第 798 頁的『z/OSMF 的必備項目』](#)
- [第 799 頁的『安全設定』](#)
- [第 801 頁的『限制』](#)

範例工作流程應用程式提供給：

- [第 802 頁的『自動佈建或取消佈建 IBM MQ 佇列管理程式，並對佈建的佇列管理程式執行動作』](#)
- [第 803 頁的『自動佈建或取消佈建 IBM MQ 本端佇列，並對佈建的佇列執行動作』](#)

相關概念

[第 697 頁的『設定 IBM MQ for z/OS』](#)

請使用本主題作為逐步手冊，以自訂 IBM MQ for z/OS 系統。

使用 IBM MQ 執行 IBM z/OS Management Facility (z/OSMF) 所需的必要條件

IBM MQ 9.1.0 中隨附的 IBM MQ 工作流程會利用 z/OSMF 中的新功能，該功能透過 z/OS 2.1 及 2.2 上的 APAR 提供。下列文字提供更多詳細資料。

1. 您已正確安裝並配置 IBM z/OS Management Facility 2.2。如果您是在啟用安全的情況下執行，請確保已配置 z/OSMF 所記載的所有安全設定。

2. 您已安裝下列 APAR:

z/OS 2.1

- PI71068
- PI71079
- PI71082
- PI71084
- OA50130

z/OS 2.2

- PI70526
- PI70521
- PI70527
- PI67839
- PI70767
- PI46315
- OA49081
- OA49802
- OA50130

3. 已配置 z/OSMF Angel (必要的話) 和伺服器程序。
4. 已配置 z/OS Cloud 環境 (如上述簡要說明並由 z/OSMF 記錄)
5. 已安裝 IBM MQ for z/OS 9.0.1，且可以使用產品載入程式庫。
6. 已執行下列 IBM MQ 佇列管理程式自訂作業:

作業	說明
1	識別 z/OS 系統參數
2	APF 授權 IBM MQ 載入程式庫
3	更新 z/OS 鏈結清單和 LPA
4	更新 z/OS 程式內容表

7. 範例工作流程及相關聯檔案安裝在適合的 UNIX System Services for z/OS (USS) 目錄中。
8. **'/tmp'** USS 目錄可用，因為 provision.xml 工作流程可能在此目錄中建立暫存檔。如果建立檔案，工作流程通常會在使用之後刪除檔案。
9. deprovision.xml 檔案中有步驟可呼叫 CSQ4ZWS1.rexx 和 CSQ4ZWS2.rexx REXX 執行程式。這些執行程式會等待佇列管理程式及通道起始程式子系統停止; 執行程式會將 USS 'SLER' 指令當作系統呼叫來呼叫。

視您的 USS 配置而定，您可能會發現 'SLER' 指令未如編碼般運作。如果在處理期間發生錯誤，指出找不到 'SLER' 指令，您可以嘗試在執行程式 CSQ4ZWS1.rexx 和 CSQ4ZWS2.rexx: 中取代下列行:

```
CALL SYSCALLS('ON')          /* Enable USS calls */
ADDRESS SYSCALL
```

```
"SLEEP" 10 /* Sleep for 10 seconds */
CALL SYSCALLS 'OFF' /* Disable USS calls */
```

具有

```
'sleep' 10
```

然後，發出 Open MVS (OMVS) **env** 指令，以檢查 PATH 環境變數設定。請確定包含 **sleep** 指令的目錄已定義到 PATH 中。請注意，**sleep** 指令通常位於 /bin 目錄中。

10. 確定已啟動 z/OSMF。

Angel 和伺服器 z/OSMF 程序都必須啟動，且 z/OSMF Web 使用者介面 (WUI) 已啟動且在執行中。如需進一步詳細資料，請參閱 [Liberty 設定檔: z/OS](#)。

即使您想要使用 REST API 來驅動工作流程，也需要啟動 z/OSMF WUI。z/OSMF WUI 有助於監視工作流程的建立和執行。

相關概念


第 797 頁的『[使用 IBM z/OSMF 來自動化 IBM MQ](#)』

IBM z/OS Management Facility (z/OSMF) 在作業導向的 Web 瀏覽器型使用者介面中提供系統管理功能，並提供整合的使用者協助，讓您可以更輕鬆地管理大型主機 z/OS 系統的日常工作及管理。

z/OS V9.1.0 安全設定

執行 z/OSMF 所需的安全設定。

下列使用者 ID 變數內容定義在內容檔中。如需詳細資料，請參閱第 805 頁的『[執行工作流程](#)』。

使用者 ID 內容	說明
CSQ_USERID	用來執行工作流程步驟的使用者 ID。不過請注意，根據下列文字中所列出的 CSQ_ADMIN_* 使用者 ID 設定，所選取步驟 (通常需要更高層次的權限) 將以不同的使用者 ID 執行。使用中的使用者 ID 由工作流程中個別步驟上的 runAsUser 內容識別。
CSQ_ADMIN_APF_USERID	當 APF 授權包含佇列管理程式系統參數模組的載入程式庫時要使用的使用者 ID。
CSQ_APF_APPROVAL_ID	用來允許使用者以使用者 CSQ_ADMIN_APF_USERID 身分執行資料集 APF 授權步驟的核准 ID。
CSQ_ADMIN_CONSOLE_USERID	在執行下執行發出 z/OS 主控台指令的步驟時所使用的使用者 ID。  小心: 必須允許此使用者 ID 對已啟動作業設定檔 (MVS.START.STC.*) 進行 UPDATE 存取。在 OPERCMDS 類別中。如需相關資訊，請參閱 z/OS 文件中的 控制操作員指令的使用 。
CSQ_CONSOLE_APPROVAL_ID	用來允許使用者執行在以使用者 CSQ_ADMIN_CONSOLE_USERID 身分執行下發出 z/OS 主控台指令之步驟的核准 ID。
CSQ_ADMIN_SAF_USERID	發出 SAF 指令時要使用的使用者 ID。
CSQ_SAF_APPROVAL_ID	用來允許使用者在以使用者 CSQ_ADMIN_SAF_USERID 身分執行下執行 SAF 指令步驟的核准 ID。
CSQ_ADMIN_SSI_USERID	發出 SETSSI 指令以識別供應給 z/OS 的子系統時要使用的使用者 ID。
CSQ_SSI_APPROVAL_ID	用來允許使用者在以使用者 CSQ_ADMIN_SSI_USERID 身分執行的情況下執行 SETSSI 指令步驟的核准 ID。

註: 用來執行供應及取消供應工作流程的使用者 ID 需要具有足夠的權限，如下所示：

1. 「佇列管理程式」供應及取消供應工作流程會使用 SETPROG 指令來 APF 授權資料集。在內容 CSQ_ADMIN_APF_USERID 中設定使用者 ID，或需要允許用來執行工作流程的使用者 ID 發出此指令。您可以發出下列指令來達到此目的：

```
PERMIT MVS.SETPROG CLASS(OPERCMD5) ID(value of CSQ_ADMIN_APF_USERID) ACCESS(UPDATE)
```

註: SETPROG 指令可能不會在 z/OS 系統的 IPL 期間持續保存, 因此可能需要在 IPL 之後手動發出下列 SETPROG 指令:

```
SETPROG APF,ADD,DSN=value of CSQ_AUTH_LIB_HLQ.value of CSQ_SSID.APF.LOAD,SMS
```

如需 SETPROG 指令的詳細資料, 請參閱 [使用 RACF 來控制 APF 清單](#)。

此外, 您可能已啟用 FACILITY 類別來控制哪些程式庫可以獲得 APF 授權, 因此您可能需要發出下列指令:

```
PERMIT CSVAPF.libname CLASS(FACILITY) ID(value of CSQ_ADMIN_APF_USERID)  
ACCESS(UPDATE)
```

2. 「佇列管理程式」供應工作流程中的步驟會發出 SETSSI 指令, 以向 z/OS 識別 IBM MQ 子系統。需要允許內容 CSQ_ADMIN_SSI_USERID 中設定的使用者 ID 使用此指令。您可以發出下列指令來達到此目的:

```
PERMIT MVS.SETSSI.ADD CLASS(OPERCMD5) ID(value of CSQ_ADMIN_SSI_USERID)  
ACCESS(CONTROL)
```

註: 已透過 SETSSI 指令識別為 z/OS 的子系統, 不會在 z/OS 系統的 IPL 期間持續保存。因此, 可能需要在 IPL 之後手動發出下列 SETSSI 指令:

```
SETSSI ADD,S='value of CSQ_SSID',I=CSQ3INI,  
P='CSQ3EPX,value of CSQ_CMD_PFX,S'
```

如需 SETSSI 指令的詳細資料, 請參閱: [SETSSI 指令](#)。

3. 工作流程會發出佇列管理程式指令, 因此如果您打算啟用安全, 則必須將 MQADMIN 或 MXADMIN 類別 (視使用的類別而定) 的 CSQ_ADMIN_RACF_USERID 內容中設定的使用者 ID (或用來執行工作流程的使用者 ID) 授與 CLAUTH (用戶端鑑別) 權限。這是容許這個使用者 ID 定義這些類別的安全設定檔。您可以發出下列指令來達到此目的:

```
ALTUSR value of CSQ_ADMIN_RACF_USERID CLAUTH(MQADMIN)
```

如需 **CLAUTH** 的詳細資料, 請參閱 [CLAUTH \(類別權限\) 屬性](#)。

4. deprovision.xml 工作流程發出 z/OS 指令 (例如, DISPLAY ACTIVE 工作、CANCEL 或 FORCE 子系統), 因此內容 CSQ_ADMIN_CONSOLE_USERID 中設定的使用者 ID (或用來執行工作流程的使用者 ID) 必須具有適當的權限才能發出這類指令。
5. 使用「軟體服務」作業的範本表格來要求佇列管理程式實例的使用者, 必須有權存取 z/OSMF 所定義的 z/OSMF 和「配置輔助程式」。
6. 佇列管理程式供應消費者的使用者 ID 需要權限, 才能在使用變數 CSQ_PROC_LIB 所定義的 PROCLIB 資料集中新增及刪除成員。
7. 必須在佈建佇列之前佈建佇列管理程式。
8. 若要使用 queueLoad.xml 和 queueOffload.xml 工作流程, 必須提前定義使用的資料集。此外, 用來執行這些工作流程的使用者 ID 需要被授與對資料集的 UPDATE 權限。
9. 佇列管理程式 provision.xml 工作流程中的步驟目前會停用子系統安全。您可以新增適當的安全指令來保護 IBM MQ 資源, 以修改工作 csq4znse.jcl 來啟用子系統安全。不過, 請注意, 如果您確實新增其他指令, 則還需要新增指令以刪除 csq4dse.jcl 中由 deprovision.xml 工作流程提交的安全許可權。

註: 此步驟會發出 RACF 安全指令。如果您使用替代安全產品, 則需要修改此步驟, 以針對您的安全產品發出適當的指令。

網路需求

新增佇列管理程式範本及範本的資源時，您需要按一下 **建立網路資源儲存區**。這會使用此範本的網路資源來建立資源儲存區。

使用「配置輔助程式」，您的網路管理者需要透過定義要配置給此範本的埠數目限制，來完成此網路資源儲存區定義。

對於每一個範本實例，provision.xml 工作流程會配置範圍中的埠，並啟動接聽器以在該埠上接聽。

使用 IBM Workload Manager 分類

如果您想要使用 WLM 來分類佇列管理程式及通道起始程式位址空間，則在新增範本以供應佇列管理程式時，需要指定此項。

是否分類，由在檔案 workflow_variables.properties 中設定的旗標 **CSQ_DEFINE_MSTR_WLM_RULE** 和 **CSQ_DEFINE_CHIN_WLM_RULE** 所控制。

如需使用 WLM 分類的相關資訊，請參閱 z/OSMF 配置手冊。

相關概念

第 798 頁的『z/OSMF 的必備項目』

使用 IBM MQ 執行 IBM z/OS Management Facility (z/OSMF) 所需的必要條件

z/OS V9.1.0 限制

z/OSMF 與 IBM MQ 搭配使用時的限制。

1. provision.xml 工作流程目前會自動執行下列強調顯示的佇列管理程式自訂作業：

作業	說明
1	識別 z/OS 系統參數
2	APF 授權 IBM MQ 載入程式庫 (provision.xml 會執行 APF 授權部分程式庫)
3	更新 z/OS 鏈結清單和 LPA
4	更新 z/OS 程式內容表
5	將 IBM MQ 子系統定義至 z/OS
6	建立 IBM MQ 佇列管理程式的程序
7	建立通道起始程式的程序
8	將 IBM MQ 子系統定義至 z/OS WLM 服務類別
9	選取並設定連結機能卸載儲存體環境
10	設定連結機能
11	實作 ESM 安全控制
12	更新 SYS1.PARMLIB 成員
13	自訂起始設定輸入資料集
14	建立引導和日誌資料集
15	定義頁面集
16	將 IBM MQ 項目新增至 Db2 資料共用群組
17	自訂系統參數模組 (部分)
18	自訂通道起始程式參數 (部分)

作業	說明
19	設定批次、TSO 和 RRS 配接器
20	設定作業及控制面板
21	併入 IBM MQ 傾出格式化成員
22	暫停參考訊息
23	更新 Advanced Message Security 的系統 DIAG 成員
24	建立 Advanced Message Security 的程序
25	設定已啟動作業使用者 Advanced Message Security
26	將 RACDCERT 許可權授與 Advanced Message Security 的安全管理者
27	授與使用者 Advanced Message Security 資源許可權

2. 必要的話，需要手動執行未以粗體字強調顯示的自訂作業。
3. 目前依現狀使用範例 INP1 及 INP2 成員。必要的話，可以定義其他內容來控制這些成員所定義的資源。
4. 內容檔中所列出特定內容的相關註解指出使用那些內容的任何限制。如需詳細資料，請參閱第 805 頁的『執行工作流程』。

相關概念

第 799 頁的『安全設定』
執行 z/OSMF 所需的安全設定。

自動佈建 IBM MQ 物件

提供範例以自動佈建佇列管理程式及本端佇列。

自動佈建或取消佈建 IBM MQ 佇列管理程式，並對佈建的佇列管理程式執行動作

提供下列佇列管理程式特定範例 z/OSMF 工作流程：

工作流程名稱	說明
provision.xml	<p>佈建 IBM MQ for z/OS 佇列管理程式</p> <p>此範例工作流程：</p> <ul style="list-style-type: none"> • 供應佇列管理程式所需的系統資源。 • 供應通道起始程式所需的系統資源。 • 啟動佇列管理程式 (也會啟動通道起始程式及 TCP/IP 接聽器) • 執行範例佇列管理程式安裝驗證程式。 <p>可以設定環境內容，以控制具有不同性質之佇列管理程式的供應。如需相關資訊，請參閱第 805 頁的『執行工作流程』。</p> <p>註：提供資訊清單檔 (provision.mf) 以協助新增此工作流程的範本。此檔案包含 qaas_readme.pdf 檔案的參照，其中包含其他資訊。在新增範本之後，您可以透過鏈結存取檔案。</p>
deprovision.xml	<p>取消供應 IBM MQ for z/OS 佇列管理程式</p> <p>此範例工作流程：</p> <ul style="list-style-type: none"> • 停止通道起始程式 (也會停止 TCP/IP 接聽器) 及佇列管理程式。 • 等待子系統停止 • 取消供應所有通道起始程式和佇列管理程式系統資源。

工作流程名稱	說明
startQMGr.xml	啟動 IBM MQ for z/OS 佇列管理程式 此範例工作流程會啟動佇列管理程式 (也會啟動通道起始程式及 TCP/IP 接聽器)。
stopQMGr.xml	停止 IBM MQ for z/OS 佇列管理程式 此範例工作流程會停止通道起始程式 (也會停止 TCP/IP 接聽器) 及佇列管理程式。

每一個工作流程都會執行一或多個步驟。工作流程中的註解會說明每一個步驟所執行的功能。部分步驟只會要求資料輸入，而部分步驟則會提交 JCL、呼叫 REXX 執行程式、Shell Script 或發出 REST API 呼叫，以完成指定的函數。

請參閱每一個步驟，以取得 JCL 或 REXX 執行程式檔案的確切名稱。工作流程及相關聯的 JCL 或 REXX 執行檔會參照在一或多個變數 XML 檔中宣告的變數。如需詳細資料，請參閱第 805 頁的『[工作流程變數宣告檔](#)』。

deprovision、**startQMGr** 和 **stopQMGr** 可以作為已供應 IBM MQ for z/OS 佇列管理程式的動作來執行。

自動佈建或取消佈建 IBM MQ 本端佇列，並對佈建的佇列執行動作

提供下列佇列特定範例 z/OSMF 工作流程：

工作流程名稱	說明
defineQueue.xml	定義本端佇列 此範例工作流程示範如何使用 z/OSMF 工作流程，根據內容設定來定義小型、中型或大型佇列。 註： 提供資訊清單檔 (provision.mf) 以協助新增此工作流程的範本。此檔案包含 qaas_readme.pdf 檔案的參照，其中包含其他資訊。在新增範本之後，您可以透過鏈結存取檔案。
displayQueue.xml	顯示本端佇列的選定屬性 此範例工作流程顯示本端佇列的選定屬性。屬性會以 z/OSMF 變數傳回 (請參閱工作流程中的步驟，以取得變數名稱)，並隨後顯示。必要的話，可以使用 REST API 來存取變數的內容。 如需詳細資料，請參閱 雲端供應 REST API ，另請參閱 z/OSMF 工作流程服務 。
deleteQueue.xml	刪除本端佇列 此範例工作流程會刪除指定佇列管理程式上的本端佇列。
putQueue.xml	將一或多個訊息放入本端佇列。 此範例工作流程會將一或多個訊息放入本端佇列。可以指定訊息文字，但如果同時將多個訊息放入本端佇列，則會使用相同的訊息文字。
getQueue.xml	從本端佇列取得一或多個訊息。 此範例工作流程會從本端佇列取得一或多個訊息。訊息會以 z/OSMF 變數傳回 (請參閱工作流程中的步驟，以取得變數名稱)，並隨後顯示。必要的話，您可以使用 REST API 來存取變數的內容。 如需詳細資料，請參閱 雲端供應 REST API ，另請參閱 z/OSMF 工作流程服務 。

工作流程名稱	說明
loadQueue.xml	將訊息從資料集載入至本端佇列。 此範例工作流程會將訊息從資料集載入至本端佇列。資料集的預設名稱是透過設定內容來指定。如需詳細資料，請參閱第 805 頁的『執行工作流程』。
offloadQueue.xml	將訊息從本端佇列卸載至資料集。 此範例工作流程會將訊息從本端佇列卸載至資料集。資料集的預設名稱是透過設定內容來指定。如需詳細資料，請參閱第 805 頁的『執行工作流程』。
clearQueue.xml	清除本端佇列上的訊息。 此範例工作流程會清除 (刪除) 本端佇列上的所有訊息。

附註:

1. **放置佇列** 動作可讓您輸入部分訊息資料，並將一或多個訊息放置在佇列中。如果在給定要求期間將多個訊息放置在佇列上，則會使用相同的訊息資料。
2. loadQueue.xml 和 offloadQueue.xml 工作流程會呼叫 SCSQLOAD 程式庫中的執行檔模組 CSQUDMSG，別名為 QLOAD。這相當於 IBM MQ for Multiplatforms 提供的 **dmpmqmsg** 公用程式。因此，從資料集載入至佇列或從佇列載入至資料集的訊息預期為 **dmpmqmsg** 格式。

在 SCSQPROC 中也提供範例 JCL 作為成員 CSQ4QLOD。

嘗試 loadQueue 及 offloadQueue 動作最簡單的方法是執行下列動作:

- a. 發出 **putQueue** 幾次，將部分訊息放入佇列中。
- b. 使用 **offloadQueue**，將佇列中的訊息卸載至資料集。
- c. 必要的話，請發出 **clearQueue**，從佇列中移除所有訊息。
- d. 使用 **loadQueue**，將資料集的訊息載入至相同或不同的佇列。

如果您對 **dmpmqmsg** 格式感興趣，則可以在發出「卸載」要求之後瀏覽資料集的內容。

3. 您可以對佈建的 IBM MQ for z/OS 本端佇列執行 **displayQueue**、**deleteQueue**、**putQueue**、**getQueue**、**loadQueue**、**offloadQueue** 及 **clearQueue** 作為動作。如需動作和動作檔的進一步詳細資料，請參閱 z/OSMF 程式設計手冊。
4. 依預設會刪除所有與動作相關的工作流程。其原因是將使用者清理工作流程的需求降至最低。

不過，問題是動作會產生部分輸出。例如，**displayQueue** 和 **getQueue** 動作都會產生輸出。

無法看到輸出，因為一旦執行動作，即會刪除相關的工作流程。因此，如果您從 z/OS WUI 驅動工作流程動作，則需要針對您要查看其輸出的每一個動作，將 **<workflow>** 標籤上的 **cleanAfterComplete** 旗標設為假。

例如，若要查看 **displayQueue** 的輸出，請如下所示設定旗標:

```
<action name="displayQueue">
  <workflow cleanAfterComplete="false">
    ...
  </workflow>
</action>
```

不過，這表示您接著必須手動清除動作相關的工作流程。

每一個範例 z/OSMF 工作流程都會執行一或多個步驟。工作流程中的註解會說明每一個步驟所執行的功能。部分步驟只會要求資料輸入，而部分步驟會提交 JCL，而其他步驟則會呼叫 REXX 執行程式來完成所述的功能。

請參閱每一個步驟，以取得 JCL 或 REXX 執行程式檔案的確切名稱。工作流程及相關聯的 JCL 或 REXX 執行檔會參照在一或多個 [第 805 頁的『工作流程變數宣告檔』](#) 中宣告的變數。

相關概念

第 801 頁的『限制』

z/OSMF 與 IBM MQ 搭配使用時的限制。

z/OS V9.1.0 執行工作流程

範例「z/OSMF」工作流程所參照檔案的說明，以及您如何執行工作流程。

工作流程變數宣告檔

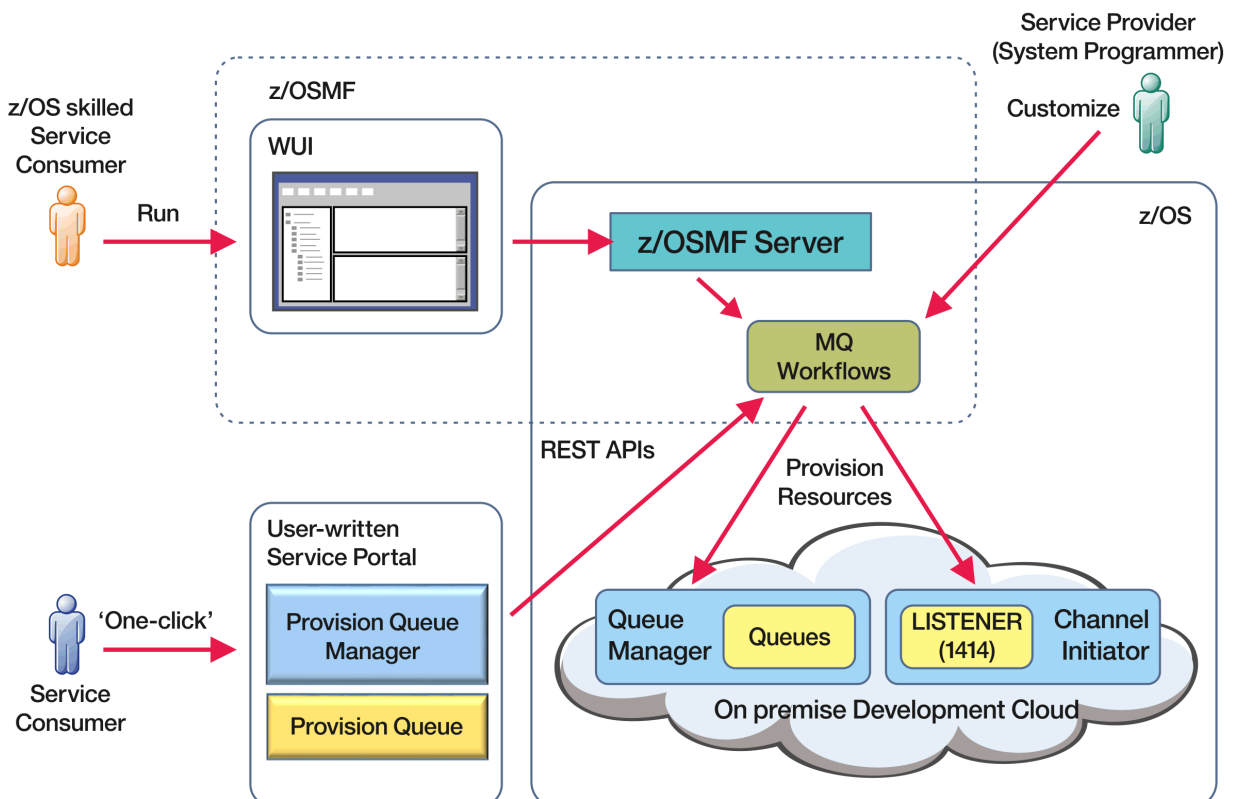
下列檔案宣告範例 z/OSMF 工作流程及相關聯 JCL 或 REXX 執行檔所參照的變數：

工作流程變數宣告檔名	說明
common_variables.xml	佇列管理程式 (加上通道起始程式) 及佇列工作流程共用的變數。
qmgr_variables.xml	特定於佇列管理程式 (加上通道起始程式) 工作流程的變數。
queue_variables.xml	特定於佇列工作流程的變數。
tcPIP_variables.xml	特定於佇列管理程式 (加上通道起始程式) 工作流程的變數，用於識別 TCP/IP 資源。

註：變數的預設可見性為 *private*。為了容許使用 z/OSMF REST API 來查詢變數，選取的變數已標示為 公用。不過，必要的話，您可以變更給定變數的可見性。

執行工作流程

圖 124: IBM MQ for z/OS 資源的「按一下」供應



在可以執行工作流程之前，需要在下列檔案中設定部分內容：

工作流程變數內容檔名稱	說明
workflow_variables.properties	<p>工作流程變數的起始內容。檔案中的註解指出每一個內容的用途。</p> <ul style="list-style-type: none">Properties within meta-brackets (< >) need to be set to user specific values.環境內容可以設為供應開發 (DEV)、測試 (TEST)、品質保證 (QA) 或正式作業 (PROD) 環境的佇列管理程式。 <p>其他內容設定會控制要為每個環境供應的佇列管理程式性質。例如，您可以針對每一種環境類型，改變作用中日誌數目或頁集數目。</p> <ul style="list-style-type: none">其他內容會設為 IBM MQ 預設值，但必要時可以修改以符合本端使用慣例。

一般而言，設定內容之後，工作流程可以依現狀執行。不過，必要的話，您可以自訂工作流程來修改或移除現有的步驟，或新增步驟。

工作流程可以執行：

- 從 z/OSMF WUI。

從 WUI 中的「雲端供應->軟體服務」，工作流程可以自動或手動模式執行。手動模式在測試時很有用，在這兩種模式中，都可以監視工作流程中每一個步驟的進度。

如需詳細資料，請參閱 [雲端佈建服務](#) 及 [建立工作流程](#)。

- 使用 z/OSMF REST Workflow Services。

「REST 工作流程服務」可用來透過 REST API 執行工作流程。此模式適用於從使用者撰寫的入口網站建立一鍵式作業。

如需詳細資料，請參閱 [雲端供應 REST API](#)，另請參閱 [z/OSMF 工作流程服務](#)。

- 使用 z/OSMF 隨附的範例 Marketplace 入口網站。

相關概念

第 802 頁的『自動佈建 IBM MQ 物件』

提供範例以自動佈建佇列管理程式及本端佇列。

z/OS MQ Adv. VUE V 9.1.0 正在配置 IBM MQ Advanced for z/OS VUE

使用此資訊來配置 IBM MQ Advanced for z/OS VUE 授權提供的特性。

關於這項作業

您可以啟用 Managed File Transfer 代理程式與 IBM MQ Advanced for z/OS Value Unit Edition 的遠端連線。

您可以使用從 IBM MQ Advanced for z/OS Value Unit Edition 佇列管理程式到 IBM Cloud (formerly Bluemix) 中 IBM Blockchain 服務的連線功能。

V 9.1.0 您可以使用用戶端連線，將 IBM MQ classes for JMS 或 IBM MQ classes for Java 應用程式連接至 z/OS 上具有 **ADVCAP (ENABLED)** 屬性的佇列管理程式。如需相關資訊，請參閱 [Java 及 JMS 用戶端與 z/OS 佇列管理程式的連線功能](#)。

程序

- 啟用 Managed File Transfer 代理程式與 IBM MQ Advanced for z/OS Value Unit Edition 的遠端連線。
- 配置 IBM MQ Advanced for z/OS VUE 以與 IBM Cloud 中的 IBM Blockchain 服務搭配使用。

線功能

z/OS 上在 IBM MQ Advanced for z/OS VUE 產品 ID (PID) 下執行的 Managed File Transfer 代理程式可以透過用戶端連線連接至 z/OS 上的遠端佇列管理程式。

當代理程式啟動時，它會將 BFGPR0137I 訊息寫入其事件日誌 (output0.log)，以顯示其執行所在的 PID。此訊息的範例如下：

```
BFGPR0137I: 已針對產品 'MQ z/OS MFT'、產品 ID '5655-MF9' 啟動產品使用情形資料記錄。
```

如需 IBM MQ 產品、其相關聯 PID 值及匯出分類的詳細資料，請參閱 [IBM MQ 產品 ID 及匯出資訊](#)。

z/OS 上以任何其他 PID 執行的 MFT 代理程式，只能使用連結連線來連接至本端佇列管理程式。

不論 MFT PID 為何，z/OS 上的 MFT 代理程式只能連接至佇列管理程式 (也可以在 z/OS 上執行)。

如果 IBM MQ Advanced for z/OS VUE 代理程式嘗試連接至不在 z/OS 上執行的佇列管理程式，則會發出訊息 BFGQM1044E: z/OS 上的代理程式用戶端連線必須通往 z/OS 上的佇列管理程式，並且代理程式啟動會結束。

相關工作

[在 z/OS 上啟動 MFT 代理程式](#)

與區塊鏈搭配使用

設定並執行 IBM MQ Bridge to blockchain，以安全地連接 z/OS 佇列管理程式及 IBM Blockchain 上的 IBM MQ。使用橋接器，透過使用連接至 IBM MQ Advanced for z/OS VUE 佇列管理程式的傳訊應用程式，非同步連接、查閱及更新區塊鏈中資源的狀態。

開始之前

- IBM MQ Bridge to blockchain 是 IBM MQ Advanced for z/OS Value Unit Edition 9.1.0 上連接器套件的一部分。您可以連接至在相同或更高指令層次上執行的 IBM MQ Advanced for z/OS VUE 佇列管理程式。
- 支援 IBM MQ Bridge to blockchain 與基於 Hyperledger Composer built on Hyperledger Fabric 的區塊鏈網路搭配使用。
- IBM MQ Bridge to blockchain 必須安裝在「UNIX 系統服務」環境中，且需要 IBM 中的 Java runtime environment 8。

關於這項作業

Blockchain 是共用的分散式分帳，由區塊鏈組成，這些區塊鏈代表網路中對等節點之間的協議交易。鏈結中的每一個區塊會鏈結至前一個區塊，依此類推，回到第一個交易。

IBM Blockchain 以 Hyperledger Fabric 和 Hyperledger Composer 為建置基礎。您可以在本端使用 Docker 進行開發，或在 IBM Cloud 的容器叢集中進行開發。您也可以正式作業中啟動及使用 IBM Blockchain 網路，以建置及控管具有高階安全、隱私權及效能的商業網路。如需相關資訊，請參閱 [IBM Blockchain Platform](#)。

Hyperledger Fabric 和 Hyperledger Composer 是一個開放程式碼的企業區塊鏈架構，由 Hyperledger Project 成員以協同方式開發，包括 IBM 作為起始程式碼貢獻者。Hyperledger Project，或 Hyperledger，是 Linux Foundation 開放程式碼、全球協同行動，以推動跨產業區塊鏈技術。如需相關資訊，請參閱 [IBM Blockchain](#)、[Hyperledger 專案](#)、[Hyperledger Fabric](#) 及 [Hyperledger Composer](#)。

如果您已使用 IBM MQ Advanced for z/OS VUE 和 IBM Blockchain，則可以使用 IBM MQ Bridge to blockchain 透過 Hyperledger Composer REST 介面來驅動 Hyperledger Composer 商業模型，以容許您更新或查詢區塊鏈中的狀態，並從區塊鏈網路接收回應。透過此方式，您可以整合內部部署 IBM 軟體與 (雲端區塊鏈服務或本端管理的內部部署解決方案)。

如需橋接器作業處理程序的簡要概觀，請參閱 [圖 1](#)。使用者應用程式會將 JSON 格式的訊息放置在 z/OS 佇列管理程式的輸入/要求佇列上。使用 Hyperledger Composer REST 伺服器，橋接器會連接至佇列管理程

式，從輸入/要求佇列取得訊息，檢查 JSON 是否正確格式化，然後向區塊鏈發出 REST 要求。區塊鏈傳回的資料由橋接器剖析，並放置在回覆佇列上，如原始 IBM MQ 要求訊息中所定義。使用者應用程式可以連接至佇列管理程式，從回覆佇列取得回應訊息，並使用該資訊。

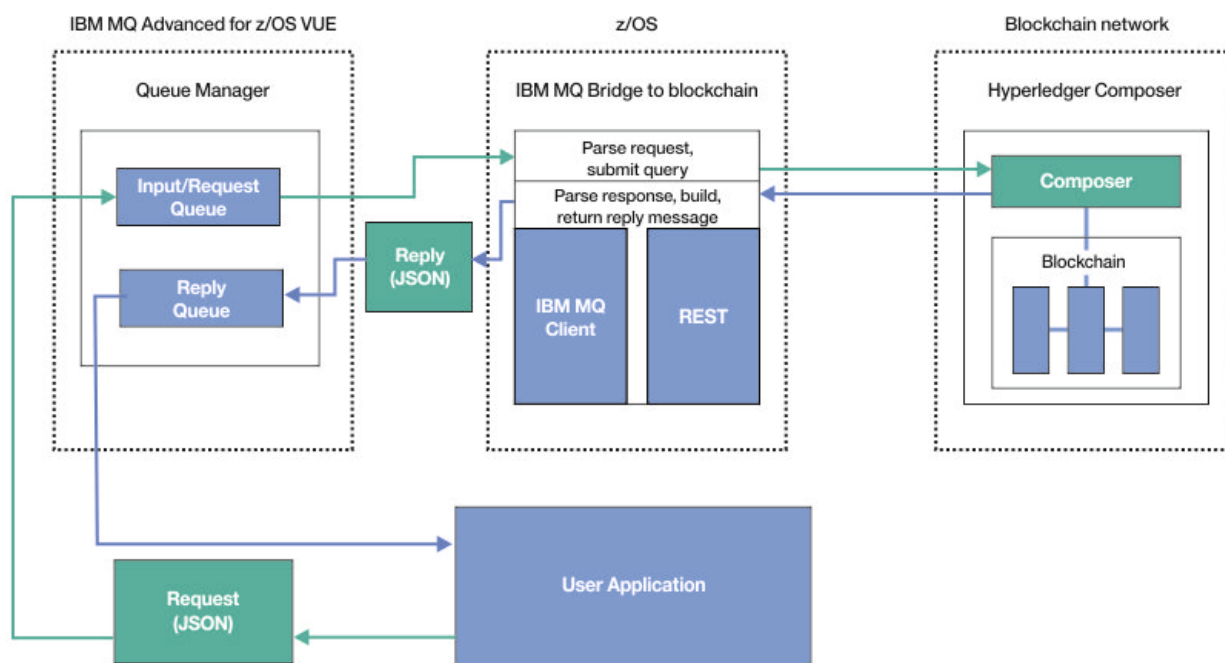


圖 125: IBM MQ Bridge to blockchain

您必須將 IBM MQ Bridge to blockchain 配置成連接至 Hyperledger Composer REST 伺服器，而不是直接連接至基礎 Hyperledger Fabric 層。當橋接器執行時，傳訊應用程式會根據使用者定義的商業網路模型，要求橋接器驅動 Hyperledger Composer REST API，進而驅動可查詢或更新資源狀態的基礎鏈碼常式，並使用 Hyperledger Composer REST 伺服器將結果作為回應傳回給傳訊應用程式。

程序

在 `th1qua1.SCSQPROC (CSQ4BCBQ)` 中自訂並提交範例 JCL，以建立橋接器的佇列。

針對用於下列項目的預設具名佇列提供範例橋接器佇列定義：

- 橋接器的訊息輸入: `SYSTEM.BLOCKCHAIN.INPUT.QUEUE` 及 `APPL1.BLOCKCHAIN.INPUT.QUEUE`
- 來自區塊鏈的回覆: `APPL1.BLOCKCHAIN.REPLY.QUEUE`

不同的應用程式可以使用相同的輸入佇列，但您可以指定多個回覆佇列，每一個應用程式一個。您不需要使用已定義的回覆佇列。如果您要對回覆使用動態佇列，則必須考量其安全配置。

結果

您已建立橋接器處理來自 IBM MQ 及區塊鏈網路的訊息所需的佇列。

下一步

使用佇列管理程式的資訊及來自區塊鏈網路的認證，來建立 IBM MQ Bridge to blockchain 的配置檔。

z/OS MQ Adv. VUE V 9.1.0 建立 IBM MQ Bridge to blockchain 的配置檔

輸入佇列管理程式及區塊鏈網路參數，以建立 IBM MQ Bridge to blockchain 的配置檔，以連接至 IBM MQ 及 IBM Blockchain 網路。

開始之前

- 您已建立並配置 Hyperledger Composer 區塊鏈網路。
- 您已在 z/OS 環境上安裝 IBM MQ Bridge to blockchain。
- 您已啟動 IBM MQ Advanced for z/OS VUE 佇列管理程式。

關於這項作業

此作業會引導您完成建立 IBM MQ Bridge to blockchain 配置檔及順利連接至 IBM Blockchain 和 IBM MQ 網路所需的最小設定。

您可以使用橋接器來連接至以 Hyperledger Composer 為基礎的區塊鏈網路。若要使用橋接器，您需要來自區塊鏈網路的配置資訊。在此作業的每一個步驟中，您可以找到基於兩個不同配置區塊鏈網路的配置詳細資料範例：

- 在 Docker 中執行的 Hyperledger Composer 網路。如需相關資訊，請參閱 [安裝 Hyperledger Composer 及產生 REST API](#)。
- 在 IBM Cloud 的 Kubernetes 叢集中執行的 Hyperledger Composer 網路。如需相關資訊，請參閱 [在 IBM Blockchain Platform 上開發雲端沙盤推演](#)。

如需所有 IBM MQ Bridge to blockchain 參數的意義及選項的相關資訊，請參閱 [runmqbcb](#) 指令。您必須考量自己的安全需求，並自訂適合您部署的參數。

程序

1. 在 UNIX 系統 Shell (USS) 環境中執行橋接器，以建立配置檔。
您需要來自 Hyperledger Composer 安全資訊以及來自 IBM MQ Advanced for z/OS VUE 佇列管理程式的參數。

從 USS 中安裝 IBM MQ 之位置的 mqbc/bin 目錄執行橋接器 Script。

```
./runmqbcb -o config_file_name.cfg
```

如下列範例所示，現有的值會顯示在方括弧內。按 Enter 以接受現有值，按 Space then Enter 以清除值，並在方括弧內鍵入，然後按 Enter 以新增值。您可以用逗點來區隔值清單 (例如同層級)，或在新行上輸入每一個值。空白行結束清單。

註: 您無法編輯現有值。您可以保留、取代或清除它們。

2. 輸入 IBM MQ Advanced for z/OS VUE 佇列管理程式的連線值。
連線所需的最小值為您定義的佇列管理程式名稱及橋接器輸入佇列名稱。對於遠端 IBM MQ Advanced for z/OS VUE 佇列管理程式的連線，您還需要 **MQ Channel** 及 **MQ Conname** (佇列管理程式執行所在的主機位址及埠)。

若要在步驟 [第 810 頁的『5』](#) 中使用 TLS 來連接至 IBM MQ，您必須使用 JNDI 或 CCDT，並相應地指定 **MQ CCDT URL** 或 **JNDI implementation class** 及 **JNDI provider URL**。

註: **MQ CCDT** 或 **JNDI** 值優先於值重疊的配置檔。

```
Connection to Queue Manager
-----
Queue Manager                : [z/OS_ADV_VUE_qmgr_name]
Bridge Input Queue           : [APPL1.BLOCKCHAIN.INPUT.QUEUE]
MQ Channel                   : []
MQ Conname                   : []
MQ CCDT URL                  : []
JNDI implementation class    : []
JNDI provider URL           : []
MQ Userid                    : []
MQ Password                  : []
```

3. 輸入與區塊鏈網路相關聯之 Hyperledger Composer REST 伺服器的認證 (如果已配置的話)。

在下列範例中，Hyperledger Composer REST 伺服器已使用 **passport-ldapauth NodeJS** 模組來配置 LDAP 認證儲存庫。請注意，您可以使用以這種方式提供基本使用者和密碼樣式認證的任何 **passport-*** 模組。如需相關資訊，請參閱 [啟用 REST 伺服器的鑑別](#)。

```
User Identification
-----
Userid                : []admin
Password              : []*****
API path for Login    : auth/ldap
```

4. 輸入 Hyperledger Composer REST 伺服器的位址。

請注意，此屬性不需要任何通訊協定 (即 http 或 https)，且埠號是必要的。使用 HTTP 或 HTTPS 通訊協定取決於 REST 伺服器的安全配置。如果提供憑證和私密金鑰組給 REST 伺服器，則會使用 HTTPS。使用 \$url.12。否則，會使用 HTTP。如需如何指定憑證和私密金鑰組的相關資訊，請參閱步驟 [第 810 頁的『5』](#)。

```
REST Server
-----
Address for Composer REST server : [composer-rest-server-ip-address:3000]
```

5. 輸入 TLS 連線的憑證儲存庫值。

橋接器充當連接至佇列管理程式的 IBM MQ JMS 用戶端，這表示可以將它配置為使用 TLS 安全，以與任何其他 IBM MQ JMS 用戶端相同的方式安全地連接。只有在步驟 [第 809 頁的『2』](#) 中指定 JNDI 或 CCDT 資訊之後，才會公開 TLS 連線詳細資料的配置。

憑證儲存庫用於 Hyperledger Composer 及 IBM MQ Advanced for z/OS VUE 佇列管理程式。如果指定憑證儲存庫，橋接器一律會嘗試使用 HTTP 連接至 Hyperledger REST 伺服器。不過，可以使用下列選項來停用 IBM MQ 連線的 TLS，同時仍將 TLS 用於 Hyperledger Composer。

```
Certificate stores for TLS connections
-----
Personal keystore          : []
Keystore password         : []
Trusted store for signer certs : []
Trusted store password    : []
Use TLS for MQ connection : [N]
Timeout for Blockchain operations : [12]
```

如需相關資訊，請參閱 [使用 HTTP 及 TLS 保護 REST 伺服器安全](#)。

6. 選擇性的: 輸入 IBM MQ Bridge to blockchain 的日誌檔位置。

您可以在配置檔或指令行中指定日誌檔名稱及位置。

```
Behavior of bridge program
-----
Runtime logfile for copy of stdout/stderr : [/var/mqm/errors/runmqbcb.log]
Done.
```

結果

您已建立配置檔，供 IBM MQ Bridge to blockchain 用來連接至 IBM Blockchain 網路及 IBM MQ Advanced for z/OS VUE 佇列管理程式。

下一步

完成 [第 811 頁的『執行 IBM MQ Bridge to blockchain』](#) 的步驟

相關參考

[runmqbcb \(執行 IBM MQ Bridge to blockchain\)](#)

IBM MQ Bridge to blockchain 的 IBM MQ 安全配置

使用 IBM MQ Bridge to blockchain 來設定 IBM MQ 安全的考量。

下列範例顯示 RACF 定義，可用來讓 IBM MQ Bridge to blockchain 存取它需要的佇列。這些定義假設橋接器是在 MQBCBUSR 使用者 ID 下執行。

此外，還需要授與「IBM MQ Bridge to blockchain」存取權，才能連接至佇列管理程式：

- 直接使用連結模式；請參閱 [批次連線的連線安全設定檔](#)，或
- 透過 CHINIT 使用用戶端模式；請參閱 [用戶端 MQI 要求](#)

IBM MQ Bridge to blockchain 要求佇列的授權

發出下列 RACF 指令，以授與 MQBCBUSR 使用者 ID 從預設 SYSTEM.BLOCKCHAIN.INPUT.QUEUE 要求佇列：

```
RDEFINE MQQUEUE SYSTEM.BLOCKCHAIN.INPUT.QUEUE UACC(NONE)
PERMIT SYSTEM.BLOCKCHAIN.INPUT.QUEUE CLASS(MQQUEUE) ID(MQBCBUSR) ACCESS(UPDATE)
```

IBM MQ Bridge to blockchain 回應佇列的授權

發出下列 RACF 指令，以授與 MQBCBUSR 使用者 ID 存取權，將訊息傳送至 APPL1.BLOCKCHAIN.REPLY.QUEUE。此佇列名稱指定在要求訊息上的回覆目的地佇列名稱上：

```
RDEFINE MQQUEUE APPL1.BLOCKCHAIN.REPLY.QUEUE UACC(NONE)
PERMIT APPL1.BLOCKCHAIN.REPLY.QUEUE CLASS(MQQUEUE) ID(MQBCBUSR) ACCESS(UPDATE)
PERMIT CONTEXT.APPL1.BLOCKCHAIN.REPLY.QUEUE CLASS(MQADMIN) ID(MQBCBUSR) ACCESS(UPDATE)
```

相關概念

[佇列安全的設定檔](#)

相關工作

第 690 頁的『[執行 IBM MQ Bridge to blockchain 用戶端範例](#)』

您可以使用 IBM MQ Bridge to blockchain 隨附的 JMS 用戶端範例，將訊息放置在區塊鏈橋接器正在檢查的輸入佇列中，並查看收到的回覆。此範例基於使用 IBM MQ Bridge to blockchain 與 Hyperledger Composer Trader 網路範例整合。

相關參考

[API-資源安全存取快速參照](#)

z/OS MQ Adv. VUE V 9.1.0 執行 IBM MQ Bridge to blockchain

執行 IBM MQ Bridge to blockchain 以連接至 IBM Blockchain 和 IBM MQ。連接時，橋接器已準備好處理要求訊息，將它們傳送至 Hyperledger Composer 區塊鏈網路，並接收及處理回覆。

關於這項作業

使用您在上一項作業中建立的配置檔，來執行 IBM MQ Bridge to blockchain。

程序

1. 啟動您要與橋接器搭配使用的 IBM MQ Advanced for z/OS VUE 佇列管理程式。
2. 啟動 IBM MQ Bridge to blockchain，以連接至區塊鏈網路及 IBM MQ Advanced for z/OS VUE 佇列管理程式。

您可以：

- a) 從 IBM MQ 安裝所在的 USS 位置中的 mqbc/bin 目錄，直接在「UNIX 系統服務 (USS)」中執行橋接器。

```
./runmqbc -f /config_file_location/config_file_name.cfg -r /log_file_location/logFile.log
```

or

- b) 使用 thlqual.SCSQPROC (CSQ4BCB) 中提供的範例 JCL，在 z/OS 系統上執行橋接器。

您必須對 JCL 進行一些特定於您環境的更新:

- 將 ++THLQUAL++ 取代為 IBM MQ 目標程式庫資料集的高階限定元。
- 將 ++LANGLETTER++ 取代為您要在其中顯示訊息之語言的字母。
- 將 ++PATHPREFIX++ 取代為 IBM MQ for z/OS USS 元件安裝路徑。
- 將 ++CONFIGFILE++ 取代為從 USS 使用 runmqbc -o <file> 指令所建立之配置檔的路徑。
- 將 ++JAVAHOME++ 取代為在 Java 8 或更新版本上執行之 64 位元 Java 虛擬機器 (JVM) 的位置。

當橋接器已連接時，會傳回類似下列的輸出:

```
2018-05-17 14:28:16.866 BST IBM MQ Bridge to Blockchain
5724-H72 (C) Copyright IBM Corp. 2017, 2024.
```

```
2018-05-17 14:28:19.331 BST Ready to process input messages.
```

3. 選擇性的: 如果在執行橋接器之後傳回的訊息指出連線不成功，則對 IBM MQ Advanced for z/OS VUE 佇列管理程式及區塊鏈網路的連線進行疑難排解。

- a) 以除錯模式發出指令，並搭配除錯選項 1。

```
./runmqbc -f /config_file_location/config_file_name.cfg -r /log_file_location/
logFile.log -d 1
```

橋接器會逐步執行連線設定，並以簡潔模式顯示處理訊息。

- b) 在除錯模式下使用除錯選項 2 發出指令。

```
./runmqbc -f /config_file_location/config_file_name.cfg -r /log_file_location/
logFile.log -d 2
```

橋接器會逐步執行連線設定，並以詳細模式顯示處理訊息。完整輸出會寫入日誌檔。

請注意，您也可以選擇性地指定 JCL 內的除錯模式選項，方法是將 '-d 0' 變更為 '-d 1' 或 '-d 2'。

結果

您已啟動 IBM MQ Bridge to blockchain 並連接至佇列管理程式及區塊鏈網路。

下一步

- 遵循第 690 頁的『[執行 IBM MQ Bridge to blockchain 用戶端範例](#)』中的步驟，將查詢或更新訊息格式化並傳送至區塊鏈網路。
- 使用 `MQBCB_EXTRA_JAVA_OPTIONS` 變數來傳入 JVM 內容，例如啟用 IBM MQ 追蹤。如需相關資訊，請參閱 [追蹤 IBM MQ Bridge to blockchain](#)。

z/OS V 9.1.0 IBM MQ 9.1.4 之前 IBM MQ Bridge to blockchain 的訊息格式

IBM MQ Bridge to blockchain 所傳送及接收之訊息的格式化相關資訊。

LTS



小心: 訊息格式的現有格式已作廢。從 IBM MQ 9.1.4，如果您有 Hyperledger Fabric 網路，請使用第 689 頁的『[IBM MQ 9.1.4 中 IBM MQ Bridge to blockchain 的訊息格式](#)』中說明的訊息格式。

應用程式要求 IBM MQ Bridge to blockchain 驅動 Hyperledger Composer 定義的 REST API，以處理區塊鏈上保留的資訊。應用程式透過在橋接器要求佇列上放置要求訊息來執行此動作。REST 要求的結果由橋接器格式化為回覆訊息。橋接器會使用要求訊息 MQMD 的 **ReplyToQ** 及 **ReplyToQMGr** 欄位中包含的資訊，作為回覆訊息的目的地。

要求及回覆訊息是 JSON 格式的文字 (MQSTR) 訊息。

要求訊息格式

要求訊息包含三個屬性:

方法 (method)

用來呼叫 Hyperledger Composer REST API 的 REST 動詞，例如 POST、DELETE 或 GET

路徑

Hyperledger Composer REST API 的路徑。這會新增至基本伺服器 URL。路徑應該以 "api/" 開頭。

body

方法特定內容。這通常是 JSON 結構。

下列範例使用 POST 方法，以路徑 `api/Trader` 來建立新的 Trader 物件。內文會指定使用者的 Hyperledger Composer 模型所定義的「交易商」類別，也會指定在區塊鏈網路內建立新 Trader 物件所需的其他值。

```
{ "method": "POST",
  "path": "api/Trader",
  "body": {
    "$class": "org.example.trading",
    "tradeId": "Trader2",
    "firstName": "Jane",
    "lastName": "Doe"
  }
}
```

回覆訊息格式

回覆訊息會將其相關性 ID 設為入埠訊息的訊息 ID。任何使用者定義內容都會從要求訊息複製到回覆訊息。回覆中的使用者 ID 會設為發送端的使用者 ID。

statusCode 是 HTTP 狀態碼。如果錯誤來自 IBM MQ 或橋接器，則會使用適當的 **statusCode**。

statusType 是字串，可以是 *SUCCESS* 或 *FAILURE*。

對於成功要求，回覆訊息中的 **"data"** 元素包含來自所呼叫 Hyperledger Composer REST API 的回應。

成功處理的範例：

```
{
  "statusCode": 200,
  "statusType": "SUCCESS",
  "data": [
    {
      "$class": "org.example.trading",
      "firstName": "John",
      "lastName": "Doe",
      "tradeId": "Trader1"
    },
    {
      "$class": "org.example.trading",
      "firstName": "Jane",
      "lastName": "Doe",
      "tradeId": "Trader2"
    }
  ]
}
```

所有錯誤回應都具有相同的欄位，不論它們是否由橋接器本身所產生，都是從呼叫 Hyperledger Composer REST 伺服器、區塊鏈或從鏈碼呼叫產生。例如：

- JSON 輸入訊息不正確

```
{
  "statusCode": 400,
  "statusType": "FAILURE",
  "message": "[AMQBC021E] Error: Cannot parse input message or there are missing fields in the message. Missing fields appear to be: \"method\"."
}
```

- Hyperledger Composer REST 伺服器無法處理的要求

```
{
```

```

"statusCode": 500,
"statusType": "FAILURE",
"message": "Error trying to invoke business network. Error: No valid responses
from any peers.\nResponse from attempted peer comms was an error: Error: chaincode
error (status: 500, message: Error: Failed to add object with ID 'Trader1'
as the object already exists)"
}

```

應用程式可以透過查看 **statusType** 字串，或從資料欄位的存在來判斷要求是成功還是失敗。當處理輸入訊息時發生錯誤，且橋接器未將它傳送至區塊鏈時，從橋接器傳回的值是 MQRC 值，通常是 **MQRC_FORMAT_ERROR**。

執行 **IBM MQ Bridge to blockchain** 用戶端範例

您可以使用 IBM MQ Bridge to blockchain 隨附的 JMS 用戶端範例，將訊息放置在區塊鏈橋接器正在檢查的輸入佇列中，並查看收到的回覆。此範例基於使用 IBM MQ Bridge to blockchain 與 Hyperledger Composer Trader 網路範例整合。

開始之前



如需相關資訊，請參閱 [/trade_network](#)

您的 IBM MQ Bridge to blockchain 正在執行中，且已連接至 IBM MQ Advanced 或 IBM MQ Advanced for z/OS VUE 佇列管理程式及區塊鏈網路。

關於這項作業

在 IBM MQ Bridge to blockchain 的 samp 目錄中尋找 JMS 範例應用程式 (ComposerBCBSamp.java)。

例如: <MQ_INSTALL_ROOT>/mqbc/samp/ComposerBCBSamp.java，其中 <MQ_INSTALL_ROOT> 是：

-  IBM MQ 的安裝目錄
-  IBM MQ 的 USS 元件安裝所在的 USS 目錄

程序

1. 編輯用戶端範例 Java 原始檔。

遵循範例中的指示來配置它，以符合您的 IBM MQ 環境及區塊鏈網路。

範例中的下列程式碼定義三個要傳送至橋接器的 JSON 要求訊息：

- 首先，移除現有的 'commodity'
- 其次，若要建立新的 'commodity'、'owner' 及相關聯的值，
- 最後，在前兩則要求訊息之後，顯示 'commodity' 的新相關資訊

```

private static JSONObject[] createMessageBodies() {
    JSONObject[] msgs = new JSONObject[3]; // This method creates 3 messages
    JSONObject m, m2;
    String commodityName = "BC";

    // Clean out the commodity in case it's already there. If
    // it's not there, there will be an error returned from Composer.
    m = new JSONObject();
    m.put("method", "DELETE");
    m.put("path", "api/Commodity/" + commodityName);
    msgs[0] = m;

    // To add the item to the table, the
    // operation looks like this:
    //
    // { "method": "POST",
    //   "path": "api/Commodity",
    //   "body" : {
    //     "$class": "org.example.trading.Commodity",

```

```

//      "tradingSymbol" : "BC",
//      "description" : "BC",
//      "mainExchange" : "HERE",
//      "owner" : "Me",
//      "quantity" : 100
//    }
//  }
// You can see this structure in the API Explorer
m = new JSONObject();
m.put("method", "POST");
m.put("path", "api/Commodity");
m2 = new JSONObject();
m2.put("$class", " org.example.trading.Commodity");
m2.put("tradingSymbol", commodityName);
m2.put("description", "Blockchain Sample Description");
m2.put("mainExchange", "My Exchange");
m2.put("owner", "Me");
m2.put("quantity", 100);
m.put("body", m2);
msgs[1] = m;

// And list all items that have been created
m = new JSONObject();
m.put("method", "GET");
m.put("path", "api/Commodity");
msgs[2] = m;

return msgs;
}

```

2. 編譯範例。

指向橋接器目錄中隨附的 IBM MQ 用戶端類別及 JSON4J.jar 檔案。

```

javac -cp <MQ_INSTALL_ROOT>/java/lib/*:<MQ_INSTALL_ROOT>/mqbc/prereqs/JSON4J.jar
ComposerBCClient.java

```

3. 執行已編譯的類別。

```

java -cp <MQ_INSTALL_ROOT>/java/lib/*:<MQ_INSTALL_ROOT>/mqbc/prereqs/JSON4J.jar:.
ComposerBCClient

```

```

Starting Simple MQ Blockchain Bridge Client
Starting the connection.
Sent message:
{"method":"DELETE","path":"api/Commodity/BC"}
Response text:
{
  "statusCode": 204,
  "statusType": "SUCCESS",
  "message": "OK",
  "data": ""
}
SUCCESS
Sent message:
{"body":
{"$class":"org.example.trading.Commodity","owner":"Me","quantity":100,"description":"Blockcha
in Sample Description","mainExchange":"My
Exchange","tradingSymbol":"BC"},"operation":"POST","url":"Commodity"}
Response text:
{
  "statusCode": 200,
  "statusType": "SUCCESS",
  "message": "OK",
  "data": {
    "$class": "org.example.trading.Commodity",
    "description": "Blockchain Sample Description",
    "mainExchange": "My Exchange",
    "owner": "Me",
    "quantity": 100,
    "tradingSymbol": "BC"
  }
}
SUCCESS
Sent message:
{"method":"GET","path":"api/Commodity"}

```

```

Response text:
{
  "statusCode": 200,
  "statusType": "SUCCESS",
  "message": "OK",
  "data": [
    {
      "$class": "org.example.trading.Commodity",
      "description": "Blockchain Sample Description",
      "mainExchange": "My Exchange",
      "owner": "resource:org.example.trading.Trader#Me",
      "quantity": 100,
      "tradingSymbol": "BC"
    }
  ]
}
SUCCESS

```

message 欄位包含已順利處理訊息的 "OK"，或如果要求失敗，則包含失敗原因的相關資訊。

如果用戶端在等待回應時收到逾時錯誤，請檢查橋接器是否在執行中。

正在配置 IBM MQ Internet Pass-Thru

本節說明 IBM MQ Internet Pass-Thru (MQIPT) 支援的各種特性，以及如何配置它們。

[IBM MQ Internet Pass-Thru 配置參照](#)中說明了可在 MQIPT 配置檔中指定的內容。

MQIPT 中的 HTTP 支援

MQIPT 支援 HTTP 通道作業。MQIPT 可以配置為將它轉遞的資料封包編碼為 HTTP 要求。

IBM MQ 通道不接受 HTTP 要求。因此，需要第二個 MQIPT 來接收 HTTP 要求，並將它們轉換回一般 IBM MQ 通訊協定封包。第二個 MQIPT 會除去 HTTP 標頭，將送入封包轉換回標準 IBM MQ 通訊協定封包，然後再將它傳遞至目的地佇列管理程式。

在兩個 MQIPT 實例之間使用 HTTP 時，HTTP 要求及回覆流程所在的 TCP/IP 連線會持續保存，並在訊息通道的生命期限內保持開啟。MQIPT 不會關閉要求/回覆配對之間的 TCP/IP 連線。

如果有兩個 MQIPT 實例透過 HTTP 進行通訊，則 HTTP 要求可能會長時間保持未完成的狀態。範例是在要求端/伺服器通道中，當伺服器端等待新訊息到達其傳輸佇列時。IBM MQ 通道通訊協定提供「活動訊號」機制，需要等待結束定期將活動訊號訊息傳送給其夥伴。預設通道活動訊號期間為 5 分鐘。MQIPT 使用此活動訊號作為 HTTP 回覆。請勿停用此通道活動訊號，或將其設為過高值，以避免在部分防火牆中造成逾時問題。

MQIPT 接受 HTTP Proxy 或伺服器所產生的區塊格式 HTTP 資料流量。

如需在 MQIPT 中使用 HTTP 的範例，請參閱 [配置 HTTP 通道作業](#)。

HTTP Proxy

HTTP Proxy 可以放置在 MQIPT 的兩個實例之間。HTTP Proxy 必須符合下列需求：

- Proxy 必須支援 HTTP 1.1 通訊協定。
- Proxy 必須允許使用 MQIPT 所設定的 **Connection** 或 **Proxy-Connection** HTTP 標頭。這可讓 MQIPT 的兩個實例之間的連線在訊息通道的生命期限內保持開啟。
- 必須跨 Proxy 維護持續性連線的一對一對映。這可確保從 Proxy 到目的地 MQIPT 的 TCP/IP 連線不會用來傳輸多個訊息通道的資料。

您可以設定內容來配置如何在部分 HTTP Proxy 上管理持續連線。例如，您可以設定在持續連線上可提出的要求數上限。應該設定下列內容：

- 應啟用持續連線。
- 應該停用多個 HTTP 階段作業重複使用從 Proxy 至 MQIPT 的 TCP/IP 連線，以維護 Proxy 之間持續連線的一對一對映。
- Proxy 要求的逾時值應該設為高值。例如，12 小時。

- 可以在持續連線提出的要求數上限應該設為高值。例如， 5000。

MQIPT 會使用 HTTP **POST** 要求，在 MQIPT 的兩個實例之間傳送資料。如果 MQIPT 配置使用 **HTTPProxy** 內容指定 Proxy 的主機名稱，則 MQIPT 會連接至 Proxy，並使用 HTTP **CONNECT** 方法要求 Proxy 建立通往目的地 MQIPT 的通道。這容許 HTTP 連線通過 Proxy，而不終止 Proxy 中的 TLS 階段作業。

如果負載平衡器放置在 MQIPT 實例之間，則必須配置為使用 **MQIPTSessionId** HTTP Cookie 的值，以確保每個階段作業的所有要求都轉遞至相同目的地。

MQIPT 中的 HTTP

在發出用戶端連線的 MQIPT 上啟用 **HTTPS** 和 **SSLClient** 路徑內容，即可在 HTTP 連線上使用 HTTP。

MQIPT 必須有權存取將用來鑑別目標 HTTP Proxy/伺服器的授信 CA 憑證。**SSLClientCAKeyring** 內容可用來定義包含授信 CA 憑證的金鑰環檔。

HTTP 的一般設定將使用本端 HTTP Proxy 來透過防火牆進行通道輸出，並連接至遠端 HTTP 伺服器 (或另一個 Proxy)，然後再連接至遠端 MQIPT。連線伺服器端上的這個 MQIPT 不需要任何特定的配置，因為連線要求會被視為任何一般 HTTP 連線。

MQIPT 使用 **HTTPProxy** 和 **HTTPServer** 內容來區分本端和遠端 Proxy。具有 **HTTPProxy** 內容集的 MQIPT 路徑被視為本端 HTTP Proxy，具有 **HTTPServer** 內容集的 MQIPT 路徑是遠端伺服器 (或 Proxy)。

通常會建立 HTTP Proxy/伺服器上接聽器埠位址 443 的 HTTP 連線，但可以使用 **HTTPProxyPort** 及 **HTTPServerPort** 內容來置換此預設值。

MQIPT 中的 SOCKS 支援

SOCKS Proxy 是一項網路服務，用來作為通過防火牆的控制點。啟用 SOCKS 的應用程式 (在防火牆內執行) 可以使用 SOCKS Proxy 來連接遠端應用程式。

MQIPT 可以透過啟用 **SocksServer** 內容來充當 SOCKS Proxy，從而容許啟用 SOCKS 的 IBM MQ 應用程式透過 MQIPT 連接至遠端 IBM MQ 佇列管理程式。使用此特性時，會在 SOCKS 信號交換處理程序期間取得目標目的地及目的地埠位址，因此會置換 **Destination** 及 **DestinationPort** 路由內容。這是支援 IBM MQ 叢集作業的主要特性。

MQIPT 也可以代表尚未啟用 SOCKS 的本端 IBM MQ 應用程式，作為 SOCKS 用戶端。當使用僅容許透過 SOCKS Proxy 進行出埠連線的防火牆時，這非常有用。每一個 MQIPT 路徑都可以配置成與不同的 SOCKS Proxy 進行通訊。

如需如何使用 SOCKS 的範例，請參閱 [配置 SOCKS Proxy](#)。

MQIPT 中的叢集作業

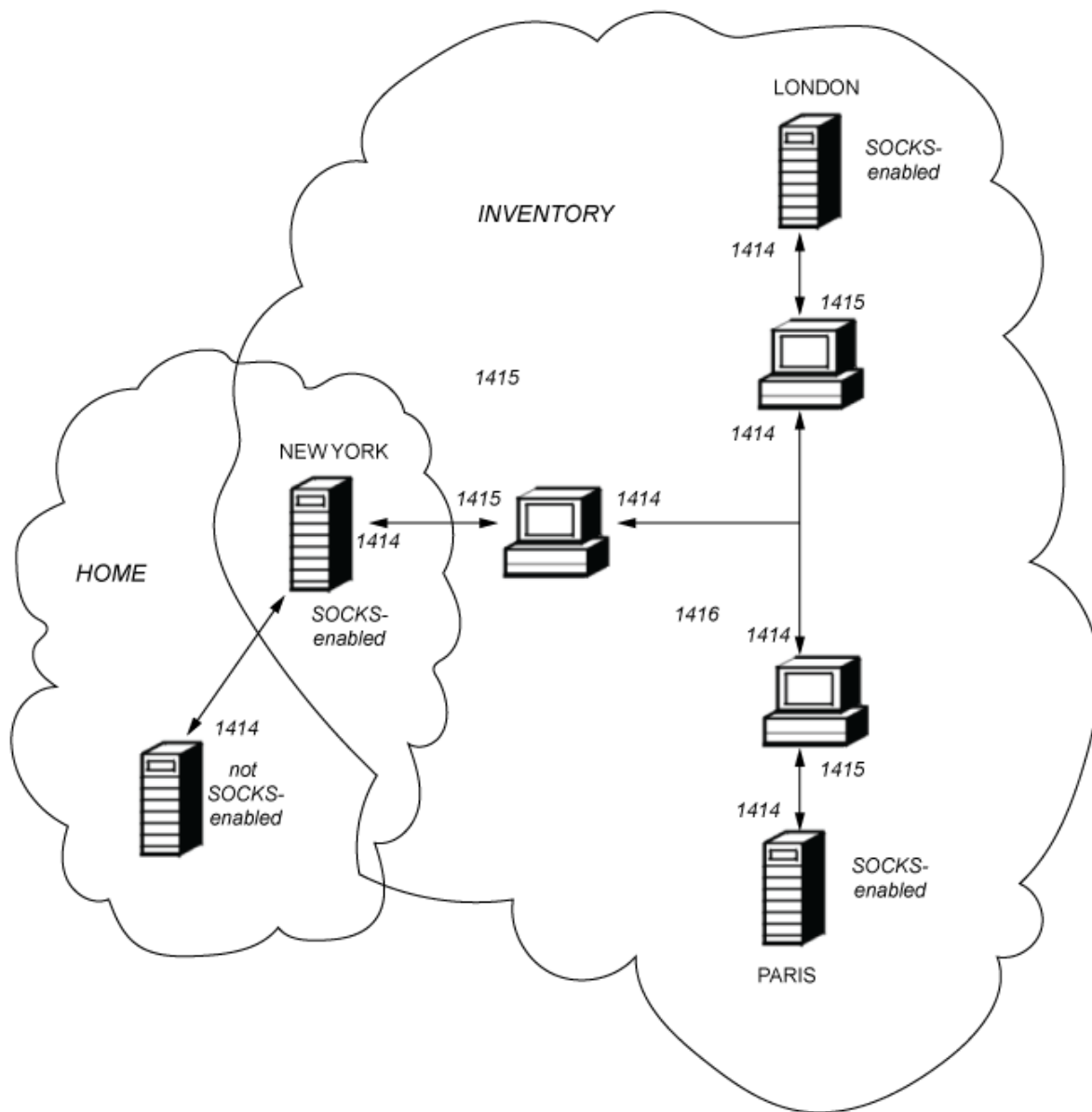
IBM MQ 叢集可以與 MQIPT 搭配使用，方法為 SOCKS-啟用叢集中跨越網際網路的每一個佇列管理程式，以及啟用 MQIPT 作為 SOCKS Proxy。

在下圖中，NEWNYK 和 CHICAGO 位於稱為 HOME 的叢集中，且兩者都擁有完整儲存庫。紐約、倫敦和巴黎都在另一個名為 "庫存" 的叢集裡。請注意，芝加哥不需要啟用 SOCKS，因為它位於不需要 MQIPT 的叢集中。

INVENTORY 叢集中的每一個佇列管理程式實際上在 MQIPT 後面「隱藏」。由於佇列管理程式已啟用 SOCKS，當啟動叢集傳送端通道時，會使用 MQIPT 作為 SOCKS Proxy，將要求送出至其目的地。通常會使用叢集接收端通道上的 CONNAME 來識別本端佇列管理程式，但與 MQIPT 搭配使用時，CONNAME 必須識別本端 MQIPT 及其送入的接聽器埠。在下圖中，所有送入的接聽器埠位址都是 1414，而送出的接聽器埠位址則是 1415。

有兩種方法可以執行啟用 SOCKS 的佇列管理程式。第一個是至 SOCKS-啟用佇列管理程式執行所在的整個電腦。第二個是 SOCKS-只啟用佇列管理程式。使用任一方法，您都必須配置 SOCKS 用戶端，以便它只使用 MQIPT 作為 SOCKS Proxy 來建立遠端連線，並停用使用者鑑別。市場上有許多產品可達成 SOCKS 支援。您必須選擇支援 SOCKS V5 通訊協定的通訊協定。

如需如何配置叢集網路的範例，請參閱 [配置 MQIPT 叢集作業支援](#)。



MQIPT 中的 SSL/TLS 支援

安全 Socket 可用來確保通訊隱私權、通訊完整性及鑑別。

通訊隱私權

連線可以成為專用。要在用戶端與伺服器之間交換的資料可以加密，且只有傳送端和接收端可以理解資料。這表示可以安全地傳送私人資訊，例如信用卡號碼。

通訊完整性

連線可靠。訊息傳輸包括基於安全雜湊函數的訊息完整性檢查。

鑑別

用戶端可以鑑別伺服器，且已鑑別伺服器可以鑑別用戶端。這表示保證只在預期的當事人之間交換資訊。鑑別機制基於數位憑證 (X.509v3 憑證) 的交換。

Secure Socket 通訊協定

在 MQIPT 中，使用「傳輸層安全 (TLS)」及 Secure Sockets Layer (SSL) 通訊協定來提供安全 Socket。這兩個安全 Socket 通訊協定是類似的，但不會交互作業。在本文件中，除非註明特定差異，否則可交換使用 SSL 和 TLS 術語。

MQIPT 支援所提供 Java runtime environment (JRE) 提供的 SSL 3.0、TLS 1.0、TLS 1.1 及 TLS 1.2。遠端通道的 IBM MQ CipherSpec 可判定 MQIPT 使用的通訊協定。

SSL 3.0 不安全，因此依預設會在 MQIPT 中停用。**V9.1.4** 依預設，在 IBM MQ 9.1.4 的 MQIPT 中也會停用 TLS 1.0 和 TLS 1.1。如果您需要使用任何這些已停用的通訊協定，可以遵循 [第 837 頁的『在 MQIPT 中啟用已淘汰的通訊協定及 CipherSuites』](#) 中的程序來重新啟用它們。

SSL/TLS 通訊協定可以使用不同的數位簽章演算法來鑑別通訊雙方。SSL/TLS 中使用的加密作業、資料機密性的加密，以及訊息完整性的安全雜湊，依賴用戶端與伺服器之間共用秘密金鑰。SSL/TLS 提供容許共用秘密金鑰的各種金鑰交換機制。SSL/TLS 可以利用各種演算法來加密和雜湊。

JRE 加密元件

JRE 的 SSL/TLS 加密元件包含 IBMJSSEFIPS 和 IBMJCEFIPS 安全提供者，它們已認證符合層次 1 的 FIPS 140-2 標準。這些安全提供者在 JRE 中具有最高優先順序，因此只要可用，就會使用 FIPS 認證的實作。支援各種加密演算法；請使用 SSL/TLS CipherSuites 來指定它們。並非所有 CipherSuites 都經過 FIPS 140-2 認證。

SSL/TLS 橋接模式

當路徑同時設定 SSLServer 和 SSLClient 時，MQIPT 會接受一個送入的 SSL/TLS 安全連線，並建立另一個 MQIPT 或目的地佇列管理程式的第二個 SSL/TLS 安全連線。在這兩個 SSL/TLS 連線之間，IBM MQ 通道資訊會解密並重新加密。SSL/TLS 橋接也稱為 SSL/TLS 終止 Proxy。

IBM MQ 支援使用 MQIPT 進行 SSL/TLS 橋接。如果 IBM MQ 的其他 SSL/TLS 終止 Proxy 結合或重新建構的 SSL/TLS 記錄大小不同於 IBM MQ 所傳送的記錄大小，則已觀察到這些 Proxy 會導致連線中斷。這是因為佇列管理程式配置及管理送入 IBM MQ 網路資料之記憶體的方式，以及 IBM MQ 網路資料包裝至 SSL/TLS 記錄的方式之間的互動。

MQIPT 會保留 SSL/TLS 記錄中 IBM MQ 網路資料的包裝，而不分割或結合它們。如果其他 SSL/TLS 橋接器未確切保留 SSL/TLS 記錄，則可能會導致 IBM MQ 通道失敗，並產生錯誤訊息：

```
AMQ9638: SSL communications error for channel  
AMQ9208: Error on receive from host
```

SSL/TLS Proxy 模式

MQIPT 路徑可以在 SSL/TLS Proxy 模式中配置，作為 SSL/TLS 橋接的替代方案。在此模式中，路徑只會在兩個 IBM MQ 端點之間轉遞 SSL/TLS 資料；它不參與 SSL/TLS 信號交換，且不需要任何數位憑證。

如果已針對 SSL/TLS 通訊配置透過 MQIPT 進行通訊的 IBM MQ 通道，且您想要將 MQIPT 用於其他用途 (例如透過防火牆遞送連線或透過安全結束程式限制容許的連線集)，則可以使用 SSL/TLS Proxy 模式。以 SSL/TLS Proxy 模式執行時，MQIPT 會在將封包轉遞至目的地之前，檢查從新連線收到的起始 SSL/TLS 封包是否有效。

IBM MQ 支援具有 MQIPT 或任何其他 SSL/TLS Proxy 的 SSL/TLS Proxy 模式

MQIPT 的 IBM MQ 多重憑證支援

IBM MQ 8.0 以及更新版本支援在相同佇列管理程式上使用多個憑證，使用通道定義上使用 **CERTLABL** 屬性指定的每個通道憑證標籤。佇列管理程式 (例如，伺服器連線或接收端) 的入埠通道依賴使用「TLS 伺服器名稱指示 (SNI)」來偵測通道名稱，以便從佇列管理程式提供正確的憑證。

如果通道透過 MQIPT 連接至目的地佇列管理程式，且 MQIPT 路徑同時設定 **SSLServer** 及 **SSLClient**，則端點之間會有兩個個別的 TLS 階段作業，且 SNI 資料不會在階段作業岔斷之間流動。這會防止在目的地佇列管理程式上使用個別通道憑證，以進行 MQIPT 與佇列管理程式之間的 TLS 連線。若要在目的地佇列管

理程式上使用每個通道的憑證，對於透過 MQIPT 傳遞的 TLS 連線，MQIPT 路徑必須使用「SSL/TLS Proxy 模式」，其會完整轉遞所有 TLS 控制流程，包括 SNI 名稱。

用於 MQIPT 所終止或起始之 TLS 連線的憑證可以針對每一個路徑個別配置，例如使用 **SSLServerSiteLabel** 及 **SSLClientSiteLabel** 路徑內容。

MQIPT 支援的 CipherSuites

下表顯示 MQIPT 支援哪些 CipherSuites，以及依預設啟用哪些 CipherSuite。

依預設，只會啟用 CipherSuites 的子集。JRE 會停用基於數個被視為不安全的演算法的 CipherSuites。如果您知道潛在危險，但仍需要使用其中一個 CipherSuites，則可以遵循第 837 頁的『在 MQIPT 中啟用已淘汰的通訊協定及 CipherSuites』中的程序來新增對已停用 CipherSuite 的支援。

CipherSuite	依預設啟用
SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA	
SSL_DH_anon_EXPORT_WITH_RC4_40_MD5	
SSL_DH_anon_WITH_3DES_EDE_CBC_SHA	
SSL_DH_anon_WITH_AES_128_CBC_SHA	
SSL_DH_anon_WITH_AES_128_CBC_SHA256	
SSL_DH_anon_WITH_AES_128_GCM_SHA256	
SSL_DH_anon_WITH_AES_256_CBC_SHA	
SSL_DH_anon_WITH_AES_256_CBC_SHA256	
SSL_DH_anon_WITH_AES_256_GCM_SHA384	
SSL_DH_anon_WITH_DES_CBC_SHA	
SSL_DH_anon_WITH_RC4_128_MD5	
SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA	
SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA	
SSL_DHE_DSS_WITH_AES_128_CBC_SHA	V 9.1.4 是
SSL_DHE_DSS_WITH_AES_128_CBC_SHA256	V 9.1.4 是
SSL_DHE_DSS_WITH_AES_128_GCM_SHA256	V 9.1.4 是
SSL_DHE_DSS_WITH_AES_256_CBC_SHA	V 9.1.4 是
SSL_DHE_DSS_WITH_AES_256_CBC_SHA256	V 9.1.4 是
SSL_DHE_DSS_WITH_AES_256_GCM_SHA384	V 9.1.4 是
SSL_DHE_DSS_WITH_DES_CBC_SHA	
SSL_DHE_DSS_WITH_RC4_128_SHA	
SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA	
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA	
SSL_DHE_RSA_WITH_AES_128_CBC_SHA	V 9.1.4 是
SSL_DHE_RSA_WITH_AES_128_CBC_SHA256	V 9.1.4 是

CipherSuite	依預設啟用
SSL_DHE_RSA_WITH_AES_128_GCM_SHA256	V 9.1.4 是
SSL_DHE_RSA_WITH_AES_256_CBC_SHA	V 9.1.4 是
SSL_DHE_RSA_WITH_AES_256_CBC_SHA256	V 9.1.4 是
SSL_DHE_RSA_WITH_AES_256_GCM_SHA384	V 9.1.4 是
SSL_DHE_RSA_WITH_DES_CBC_SHA	
SSL_ECDH_anon_WITH_3DES_EDE_CBC_SHA	
SSL_ECDH_anon_WITH_AES_128_CBC_SHA	
SSL_ECDH_anon_WITH_AES_256_CBC_SHA	
SSL_ECDH_anon_WITH_NULL_SHA	
SSL_ECDH_anon_WITH_RC4_128_SHA	
SSL_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA	
SSL_ECDH_ECDSA_WITH_AES_128_CBC_SHA	是
SSL_ECDH_ECDSA_WITH_AES_128_CBC_SHA256	是
SSL_ECDH_ECDSA_WITH_AES_128_GCM_SHA256	是
SSL_ECDH_ECDSA_WITH_AES_256_CBC_SHA	是
SSL_ECDH_ECDSA_WITH_AES_256_CBC_SHA384	是
SSL_ECDH_ECDSA_WITH_AES_256_GCM_SHA384	是
SSL_ECDH_ECDSA_WITH_NULL_SHA	
SSL_ECDH_ECDSA_WITH_RC4_128_SHA	
SSL_ECDH_RSA_WITH_3DES_EDE_CBC_SHA	
SSL_ECDH_RSA_WITH_AES_128_CBC_SHA	是
SSL_ECDH_RSA_WITH_AES_128_CBC_SHA256	是
SSL_ECDH_RSA_WITH_AES_128_GCM_SHA256	是
SSL_ECDH_RSA_WITH_AES_256_CBC_SHA	是
SSL_ECDH_RSA_WITH_AES_256_CBC_SHA384	是
SSL_ECDH_RSA_WITH_AES_256_GCM_SHA384	是
SSL_ECDH_RSA_WITH_NULL_SHA	
SSL_ECDH_RSA_WITH_RC4_128_SHA	
SSL_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	
SSL_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	是
SSL_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	是
SSL_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	是
SSL_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	是
SSL_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	是

CipherSuite	依預設啟用
SSL_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	是
SSL_ECDHE_ECDSA_WITH_NULL_SHA	
SSL_ECDHE_ECDSA_WITH_RC4_128_SHA	
SSL_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	
SSL_ECDHE_RSA_WITH_AES_128_CBC_SHA	是
SSL_ECDHE_RSA_WITH_AES_128_CBC_SHA256	是
SSL_ECDHE_RSA_WITH_AES_128_GCM_SHA256	是
SSL_ECDHE_RSA_WITH_AES_256_CBC_SHA	是
SSL_ECDHE_RSA_WITH_AES_256_CBC_SHA384	是
SSL_ECDHE_RSA_WITH_AES_256_GCM_SHA384	是
SSL_ECDHE_RSA_WITH_NULL_SHA	
SSL_ECDHE_RSA_WITH_RC4_128_SHA	
SSL_KRB5_EXPORT_WITH_DES_CBC_40_MD5	
SSL_KRB5_EXPORT_WITH_DES_CBC_40_SHA	
SSL_KRB5_EXPORT_WITH_RC4_40_MD5	
SSL_KRB5_EXPORT_WITH_RC4_40_SHA	
SSL_KRB5_WITH_3DES_EDE_CBC_MD5	
SSL_KRB5_WITH_3DES_EDE_CBC_SHA	
SSL_KRB5_WITH_DES_CBC_MD5	
SSL_KRB5_WITH_DES_CBC_SHA	
SSL_KRB5_WITH_RC4_128_MD5	
SSL_KRB5_WITH_RC4_128_SHA	
SSL_RSA_EXPORT_WITH_DES40_CBC_SHA	
SSL_RSA_EXPORT_WITH_RC4_40_MD5	
SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA (附註 1)	
SSL_RSA_FIPS_WITH_DES_CBC_SHA (附註 1)	
SSL_RSA_WITH_3DES_EDE_CBC_SHA	
SSL_RSA_WITH_AES_128_CBC_SHA	是
SSL_RSA_WITH_AES_128_CBC_SHA256	是
SSL_RSA_WITH_AES_128_GCM_SHA256	是
SSL_RSA_WITH_AES_256_CBC_SHA	是
SSL_RSA_WITH_AES_256_CBC_SHA256	是
SSL_RSA_WITH_AES_256_GCM_SHA384	是
SSL_RSA_WITH_DES_CBC_SHA	
SSL_RSA_WITH_NULL_MD5	

CipherSuite	依預設啟用
SSL_RSA_WITH_NULL_SHA	
SSL_RSA_WITH_NULL_SHA256	
SSL_RSA_WITH_RC4_128_MD5	是
SSL_RSA_WITH_RC4_128_SHA	

註:

1. 雖然支援此 CipherSuite 以與舊版相容，但它不再符合 FIPS 標準，應該避免使用它。

IBM MQ CipherSpecs 和 MQIPT CipherSuites

下表顯示 IBM MQ 所支援的 CipherSpecs 與 MQIPT 所支援的 CipherSuites 之間的關係。

此表格也顯示 IBM MQ 預期每一個 CipherSpec 使用的通訊協定版本。

IBM MQ CipherSpec 可唯一決定加密演算法以及要使用的安全 Socket 通訊協定版本。部分 IBM MQ CipherSpecs 僅因通訊協定版本而異，因此僅配置 CipherSuite 並不足夠。SSL/TLS 信號交換會協議雙方支援的最高安全 Socket 通訊協定版本，然後從一組相互啟用的密碼中選取 CipherSuite。

例如，具有 SSLClientCipherSuites=SSL_RSA_WITH_3DES_EDE_CBC_SHA 的 SSLClient 路徑可以與遠端佇列管理程式協議 TLS_RSA_WITH_3DES_EDE_CBC_SHA (TLS 1.0) 或 TRIPLE_DES_SHA_US (SSL 3.0)。事實上，可以協議此 CipherSuite over TLS 1.2，但 IBM MQ 不支援此 CipherSuite over TLS 1.2。因此，SSLClient 路徑特別可能導致佇列管理程式上發生 AMQ9616 或 AMQ9631 錯誤。

若要避免在 SSLClient 路徑上發生此類錯誤，請將 **SSLClientProtocols** 路徑內容設為預期的 CipherSpec 的適當值。在某些情況下，也可能需要使用 **SSLServerProtocols** route 內容來限制伺服器端通訊協定集。使用表格中顯示的通訊協定版本，以判定這些路由內容的正確設定。

此問題尤其會影響 SSLClient 路徑的下列 CipherSuites 及 CipherSpecs：

- SSL_RSA_WITH_3DES_EDE_CBC_SHA，對應於：
 - SSL 3.0: MQ CipherSpec TRIPLE_DES_SHA_US
 - TLS 1.0: MQ CipherSpec TLS_RSA_WITH_3DES_EDE_CBC_SHA
- SSL_RSA_WITH_DES_CBC_SHA，對應於：
 - SSL 3.0: MQ CipherSpec DES_SHA_EXPORT
 - TLS 1.0: MQ CipherSpec TLS_RSA_WITH_DES_CBC_SHA
- SSL_RSA_WITH_RC4_128_SHA，對應於：
 - SSL 3.0: MQ CipherSpec RC4_SHA_US
 - TLS 1.2: MQ CipherSpec TLS_RSA_WITH_RC4_128_SHA256

如果您要使用單一 MQIPT SSLClient 路徑來通道多個使用不同 CipherSpecs 的 IBM MQ 通道，請確保所有通道都具有彼此使用相同安全 Socket 通訊協定版本的 CipherSpecs，並將 **SSLClientProtocols** 設為使用此單一通訊協定版本。

如需 IBM MQ CipherSpecs 的相關資訊，請參閱 [啟用 CipherSpecs](#)。

IBM MQ CipherSpec	MQIPT CipherSuite	通訊協定版本
DES_SHA_EXPORT	SSL_RSA_WITH_DES_CBC_SHA	SSLv3
DES_SHA_EXPORT1024	N/A	N/A
ECDHE_ECDSA_3DES_EDE_CBC_SHA256	SSL_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	TLSv1.2
ECDHE_ECDSA_AES_128_CBC_SHA256	SSL_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	TLSv1.2

IBM MQ CipherSpec	MQIPT CipherSuite	通訊協定版本
ECDHE_ECDSA_AES_128_GCM_SHA256	SSL_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	TLSv1.2
ECDHE_ECDSA_AES_256_CBC_SHA384	SSL_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	TLSv1.2
ECDHE_ECDSA_AES_256_GCM_SHA384	SSL_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	TLSv1.2
ECDHE_ECDSA_NULL_SHA256	SSL_ECDHE_ECDSA_WITH_NULL_SHA	TLSv1.2
ECDHE_ECDSA_RC4_128_SHA256	SSL_ECDHE_ECDSA_WITH_RC4_128_SHA	TLSv1.2
ECDHE_RSA_3DES_EDE_CBC_SHA256	SSL_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	TLSv1.2
ECDHE_RSA_AES_128_CBC_SHA256	SSL_ECDHE_RSA_WITH_AES_128_CBC_SHA256	TLSv1.2
ECDHE_RSA_AES_128_GCM_SHA256	SSL_ECDHE_RSA_WITH_AES_128_GCM_SHA256	TLSv1.2
ECDHE_RSA_AES_256_CBC_SHA384	SSL_ECDHE_RSA_WITH_AES_256_CBC_SHA384	TLSv1.2
ECDHE_RSA_AES_256_GCM_SHA384	SSL_ECDHE_RSA_WITH_AES_256_GCM_SHA384	TLSv1.2
ECDHE_RSA_NULL_SHA256	SSL_ECDHE_RSA_WITH_NULL_SHA	TLSv1.2
ECDHE_RSA_RC4_128_SHA256	SSL_ECDHE_RSA_WITH_RC4_128_SHA	TLSv1.2
FIPS_WITH_3DES_EDE_CBC_SHA (附註 1)	SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA	SSLv3
FIPS_WITH_DES_CBC_SHA (附註 1)	SSL_RSA_FIPS_WITH_DES_CBC_SHA	SSLv3
NULL_MD5	SSL_RSA_WITH_NULL_MD5	SSLv3
NULL_SHA	SSL_RSA_WITH_NULL_SHA	SSLv3
RC2_MD5_EXPORT	N/A	N/A
RC4_56_SHA_EXPORT1024	N/A	N/A
RC4_MD5_EXPORT	SSL_RSA_EXPORT_WITH_RC4_40_MD5	SSLv3
RC4_MD5_US	SSL_RSA_WITH_RC4_128_MD5	SSLv3
RC4_SHA_US	SSL_RSA_WITH_RC4_128_SHA	SSLv3
TLS_RSA_WITH_3DES_EDE_CBC_SHA	SSL_RSA_WITH_3DES_EDE_CBC_SHA	TLSv1
TLS_RSA_WITH_AES_128_CBC_SHA	SSL_RSA_WITH_AES_128_CBC_SHA	TLSv1
TLS_RSA_WITH_AES_128_CBC_SHA256	SSL_RSA_WITH_AES_128_CBC_SHA256	TLSv1.2
TLS_RSA_WITH_AES_128_GCM_SHA256	SSL_RSA_WITH_AES_128_GCM_SHA256	TLSv1.2
TLS_RSA_WITH_AES_256_CBC_SHA	SSL_RSA_WITH_AES_256_CBC_SHA	TLSv1
TLS_RSA_WITH_AES_256_CBC_SHA256	SSL_RSA_WITH_AES_256_CBC_SHA256	TLSv1.2
TLS_RSA_WITH_AES_256_GCM_SHA384	SSL_RSA_WITH_AES_256_GCM_SHA384	TLSv1.2
TLS_RSA_WITH_DES_CBC_SHA	SSL_RSA_WITH_DES_CBC_SHA	TLSv1
TLS_RSA_WITH_NULL_NULL	N/A	N/A
TLS_RSA_WITH_NULL_SHA256	SSL_RSA_WITH_NULL_SHA256	TLSv1.2
TLS_RSA_WITH_RC4_128_SHA256	SSL_RSA_WITH_RC4_128_SHA	TLSv1.2
TRIPLE_DES_SHA_US	SSL_RSA_WITH_3DES_EDE_CBC_SHA	SSLv3

MQIPT 中的 SSL/TLS 信號交換

在 SSL/TLS 用戶端與伺服器之間的起始連線要求期間，當執行 CipherSuites 的鑑別及協議時，會執行 SSL/TLS 信號交換處理程序。

所有支援的 SSL/TLS CipherSuites (請參閱第 818 頁的『MQIPT 中的 SSL/TLS 支援』) (匿名 CipherSuites 除外) 都需要伺服器鑑別並容許用戶端鑑別; 伺服器可以配置為要求用戶端鑑別。您應該避免使用匿名 CipherSuites，因為它們無法保證遠端對等節點的身分。中間人攻擊可能在您不知情的情況下截取匿名 SSL/TLS 連線。僅在值得信任的內部網路上使用匿名 CipherSuites，且僅在您準備接受資料截取風險時使用。

SSL/TLS 中的通訊同層級鑑別是根據公開金鑰加密法及 X.509v3 數位憑證。應該在 SSL/TLS 通訊協定中鑑別的網站需要私密金鑰及數位憑證 (包含對應的公開金鑰以及網站身分的相關資訊)，憑證的有效時間。憑證由「憑證管理中心」簽署，此類憑證管理中心的憑證稱為簽章者憑證。後面接著一或多個簽章者憑證的憑證構成憑證鏈。憑證鏈的特點是，從第一個憑證 (網站憑證) 開始，可以使用下一個簽章者憑證中包含的公開金鑰來驗證鏈中每一個憑證的簽章。

當建立需要伺服器鑑別的安全連線時，伺服器會傳送憑證鏈給用戶端，以證明其身分。只有在伺服器可以鑑別時 (例如，驗證伺服器網站憑證的簽章)，SSL/TLS 用戶端才會繼續建立伺服器的連線。為了驗證簽章，SSL/TLS 用戶端需要信任伺服器網站本身，或伺服器所提供憑證鏈中至少一個簽章者。必須在用戶端維護授信網站及簽章者的憑證，才能執行此驗證。

SSL/TLS 用戶端會從網站憑證開始檢查伺服器的憑證鏈。在下列情況下，用戶端會將網站憑證的簽章視為有效：

- 網站憑證位於授信網站或簽章者憑證的儲存庫中
- 鏈中的簽章者憑證可以根據其授信簽章者憑證儲存庫來驗證

在後一種情況下，SSL/TLS 用戶端會檢查憑證鏈是否確實正確簽署 (從授信簽章者憑證到伺服器網站憑證)。也會檢查此處理程序所涉及的每一個憑證的格式及有效性日期的正確性。如果其中任何檢查失敗，則會拒絕與伺服器的連線。驗證伺服器憑證之後，用戶端會在 SSL/TLS 通訊協定的後續步驟中使用該憑證中內嵌的公開金鑰。只有在伺服器確實具有對應的私密金鑰時，才能建立 SSL/TLS 連線。

用戶端鑑別遵循相同的程序：如果 SSL/TLS 伺服器需要用戶端鑑別，用戶端會將憑證鏈傳送至伺服器以證明其身分。伺服器會根據其授信網站和簽章者憑證的儲存庫來驗證鏈結。在驗證用戶端的憑證之後，伺服器會在 SSL/TLS 通訊協定的後續步驟中使用該憑證中內嵌的公開金鑰。只有在用戶端確實具有對應的私密金鑰時，才能建立 SSL/TLS 連線。

最新版本的 TLS 通訊協定提供高安全通訊 (SSL 及較舊的 TLS 通訊協定被視為不安全)。不過，通訊協定會根據應用程式所提供的資訊來運作。只有在也安全維護該資訊庫時，才能達到安全通訊的整體目標。例如，如果信任網站和簽章者憑證的儲存庫受損，您可以建立與非常不安全的通訊夥伴的安全連線。

SSL/TLS 的 MQIPT 實作

SSL 3.0 及 TLS 1.0、1.1 及 1.2 是使用儲存在金鑰環檔 (檔案類型為 .p12 或 .pfx) 中的「公開金鑰密碼化標準 (PKCS) #12」記號來實作，其中包含 X509.V3 憑證。 **V9.1.4** MQIPT 也可以使用支援 PKCS#11 加密記號介面標準的加密硬體金鑰儲存庫。MQIPT 使用 IBM Java Secure Socket Extension (JSSE) 套件。

MQIPT 可以作為 SSL/TLS 用戶端或 SSL/TLS 伺服器，視起始連線的端而定。用戶端會啟動連線，且伺服器會接受連線要求。MQIPT 路徑可以同時作為用戶端和伺服器。在此情況下，使用「SSL/TLS Proxy 模式」特性通常會提供更好的效能。

當 MQIPT 配置為「SSL/TLS Proxy 模式」時，它只會在兩個端點之間轉遞 SSL/TLS 資料；它不參與 SSL/TLS 信號交換，也不需要任何數位憑證。

MQIPT 不會將入埠 TLS 連線收到的「TLS 伺服器名稱指示 (SNI)」資料傳遞至出埠 TLS 連線。這表示使用 **CERTLABL** 通道屬性指定的每個通道憑證無法用於 MQIPT 與目的地佇列管理程式之間的 TLS 連線。若要在目的地佇列管理程式上使用每個通道的憑證，對於透過 MQIPT 傳遞的 TLS 連線，MQIPT 路徑必須使用「SSL/TLS Proxy 模式」，其會完整轉遞所有 TLS 控制流程，包括 SNI 名稱。如需在佇列管理程式上搭配使用多個憑證與 MQIPT 的相關資訊，請參閱第 819 頁的『MQIPT 的 IBM MQ 多重憑證支援』。

每一個 MQIPT 路徑都可以獨立配置其自己的 SSL/TLS 內容集。如需詳細資料，請參閱 [MQIPT 路由內容](#)。

在 MQIPT 中加密金鑰環密碼

使用 `mqiPTPW` 指令來加密用來開啟金鑰環檔的密碼，或用來存取 MQIPT 所使用的加密硬體。下列任何內容都可以使用加密密碼：`SSLClientKeyRingPW`、`SSLClientCAKeyRingPW`、`SSLServerKeyRingPW` 和 `SSLServerCAKeyRingPW`。這個主題說明儲存金鑰環密碼供 MQIPT 使用的正確方式。

MQIPT 不支援 `mqiPTkeyman` (iKeyman) 隱藏檔機能。您必須使用 `mqiPTPW` 指令來儲存已加密密碼，而不是使用隱藏檔。

在早於 IBM MQ 9.1.5 的版本中，供 MQIPT 使用的金鑰環密碼儲存在任何 `SSL*KeyRingPW` 內容所參照的檔案中。

V 9.1.5 從 IBM MQ 9.1.5 中，使用 `mqiPTPW` 指令來加密供 MQIPT 使用的金鑰環密碼，並將 `SSL*KeyRingPW` 內容的值設為已加密密碼。為了與 IBM MQ 9.1.5 之前建立的配置相容，MQIPT 能夠區分內容值中的已加密密碼和檔名。

在 IBM MQ 9.1.5 之前的 MQIPT 版本中提供的金鑰儲存庫密碼加密方法已淘汰，但仍可使用。若要改善金鑰環密碼的保護，請使用最新的保護方法，重新加密先前已加密的任何金鑰環密碼。

若要加密金鑰環密碼以供 MQIPT 使用，請遵循 [加密儲存的密碼](#) 中的步驟。

您必須使用密碼 `mqiPTSample` 來開啟 MQIPT 安裝目錄的 `samples/ssl` 子目錄中提供的任一範例金鑰環檔案。

在 MQIPT 中從金鑰環檔選取憑證

可以將多個個人憑證儲存在相同的金鑰環檔案或加密硬體記號中。`SSLClientSite*` 內容可以在用戶端使用，以選取要傳送至伺服器進行鑑別的憑證，而 `SSLServerSite*` 內容可以在伺服器端使用，以選取要傳送至用戶端進行鑑別的憑證。

使用這些內容，可以根據憑證的「識別名稱 (DN)」來選取憑證。或者，憑證標籤可用來使用 `SSLServerSiteLabel` 和 `SSLClientSiteLabel` 內容來選取憑證。

MQIPT 中的信任設定

金鑰環包含個人憑證，其中包括簽章者憑證或簽章者憑證鏈。

MQIPT 使用兩種類型的金鑰環：

憑證管理中心 (CA) 金鑰環

此金鑰環包含用來驗證屬於遠端對等節點之憑證的授信 CA 憑證。這些 CA 憑證有助於判斷遠端對等節點是否值得信任。MQIPT 同時支援 PKCS #12 格式金鑰環檔案及支援 PKCS #11 介面的加密硬體金鑰儲存庫，以儲存 CA 憑證。MQIPT CA 金鑰環檔由 `SSLClientCAKeyRing` 和 `SSLServerCAKeyRing` 路由內容識別。透過設定 `SSLClientCAKeyRingUseCryptoHardware` 及 `SSLServerCAKeyRingUseCryptoHardware` 內容，啟用使用加密硬體來存取 CA 憑證。

SSL/TLS 用戶端上的 CA 金鑰環應該包含授信 CA 憑證清單，將用來鑑別從伺服器傳送的憑證。如果針對用戶端鑑別配置 SSL 伺服器路徑，則 SSL/TLS 伺服器端上的 CA 金鑰環應該包含信任 CA 憑證清單，將用來鑑別從用戶端傳送的憑證。

個人憑證金鑰環

此金鑰環包含 MQIPT 用來向遠端對等節點識別本身的個人憑證。當您產生自簽憑證或要求 CA 簽章憑證時，應該使用個人憑證金鑰環來執行此動作。MQIPT 同時支援 PKCS #12 格式金鑰環檔案及支援 PKCS #11 介面的加密硬體金鑰儲存庫，以儲存個人憑證。在 MQIPT 中，個人憑證金鑰環檔由 `SSLClientKeyRing` 和 `SSLServerKeyRing` 路徑內容識別。透過設定 `SSLClientKeyRingUseCryptoHardware` 及 `SSLServerKeyRingUseCryptoHardware` 內容，啟用使用加密硬體來存取個人憑證。

SSL/TLS 伺服器端上的金鑰環應該包含 MQIPT 伺服器的個人憑證。如果 SSL 用戶端路徑需要用戶端鑑別，SSL/TLS 用戶端上的金鑰環應該包含用戶端的個人憑證。

如果您需要用戶端鑑別，則必須在伺服器端啟用 `SSLServerAskClientAuth` 內容。用戶端上的金鑰環應該包含用戶端的個人憑證。伺服器端的 MQIPT 金鑰環 (由 `SSLServerCAKeyRing` 內容識別) 應該包含將用來鑑別用戶端的授信 CA 憑證清單。

如果您未配置路徑的 CA 金鑰環，則 MQIPT 會改為在個人憑證金鑰環中搜尋 CA 憑證 (如果已配置的話)。例如，如果未設定 **SSLServerCAKeyRing** 的值，MQIPT 會在 **SSLServerKeyRing** 所識別的金鑰環中搜尋 CA 憑證。

除了使用授信 CA 所簽署的憑證之外，您還可以使用自簽憑證。您可以在 `samples/ssl` 子目錄中的 MQIPT 隨附的 `sslSample.pfx` 範例金鑰環檔案中找到自簽憑證的範例。若要開啟範例 PKCS#12 金鑰環檔，您必須使用密碼 `mqiptSample`。

在測試情境中，如果您必須確保 SSL/TLS 連線功能，而不需向 CA 支付憑證費用，則自簽憑證可能很有用。不過，您不應在正式作業環境中使用自簽憑證。若要建立 CA 簽章憑證，請參閱 [建立金鑰環檔案](#)。

您可以使用稱為 **mqiptkeyman** 的公用程式 (隨 MQIPT 提供) 來管理數位憑證及金鑰儲存庫。如需安裝指示及進一步資訊，請參閱第 829 頁的『MQIPT 中的 `mqiptKeyman` 和 `mqiptKeycmd`』。

您必須使用作業系統的安全特性來保護任何金鑰環檔案及密碼檔，以防止未獲授權存取它們。

在 MQIPT 中測試 SSL/TLS

您可以使用本文件中提供的範例來測試 SSL/TLS 連線。

如需各種實務範例的說明，請參閱 [開始使用 IBM MQ Internet Pass-Thru](#)。特別是請參閱下列作業：

- [鑑別 SSL/TLS 伺服器](#)
- [鑑別 SSL/TLS 用戶端](#)
- [以 SSL/TLS Proxy 模式執行 MQIPT](#)
- [使用安全管理程式以 SSL/TLS Proxy 模式執行 MQIPT](#)

若要測試 SSL/TLS 配置是否正確運作，您可以使用自簽憑證。自簽憑證在測試情境中很有用，因此您可以確保 SSL/TLS 連線功能，而無需支付憑證管理中心 (CA) 的費用。如需詳細資料，請參閱 [建立測試憑證](#)。

您可以在 `samples/ssl` 子目錄中的 MQIPT 隨附的 `sslSample.pfx` 範例金鑰環檔案中找到自簽憑證的範例。若要開啟範例 PKCS #12 金鑰環檔案，您必須使用密碼 `mqiptSample`。提供範例憑證是為了方便您在測試期間使用。不過，所有 MQIPT 使用者都知道範例憑證的私密金鑰。這表示它不安全，應該只在測試環境中使用。

您不應在正式作業環境中使用任何自簽憑證，不論它們是否為範例憑證。而是從授信 CA 取得 CA 簽章憑證。若要建立 CA 簽章憑證，請參閱 [建立金鑰環檔案](#)。

建立或要求憑證時，您應該考量適合您安全需求的金鑰類型、金鑰大小及數位簽章演算法。如需進一步資訊，請參閱第 830 頁的『MQIPT 的數位憑證考量』。

一些第三方供應商提供憑證和憑證管理技術。

MQIPT 中的 SSL/TLS 錯誤訊息

信號交換失敗會以 JSSE 異常狀況的形式記載在 MQIPT 連線日誌中。

如需相關資訊，請參閱第 849 頁的『MQIPT 中的連線日誌』。下表說明不同的異常狀況、可能的原因，以及解決失敗的對應動作。

憑證異常狀況通常與連線遠端的憑證相關。

如果錯誤與 IBM MQ 用戶端或佇列管理程式的憑證相關，則術語 **金鑰環檔案** 包括遠端友機的 IBM MQ 金鑰儲存庫。

在 MQIPT 中，CA 憑證儲存在 CA 金鑰環檔中，該檔案由 **SSLClientCAKeyRing** 和 **SSLServerCAKeyRing** 路徑內容識別。如果未設定 CA 金鑰環路徑內容，則會改為在對應的個人金鑰環檔案 (由 **SSLClientKeyRing** 或 **SSLServerKeyRing** 內容參照) 中搜尋 CA 憑證。

異常狀況	原因	動作
CertificateException	憑證不受信任，因為它是由不在 CA 金鑰環中的 CA 所簽署。	請檢查 CA 金鑰環檔中是否存在所有必要的 CA 憑證。使用 MQIPT 隨附的 IBM Key Management 工具來新增任何遺漏的 CA 憑證，請小心從值得信任的來源取得每一個 CA 憑證的副本。
CertificateExpired 異常狀況	<ol style="list-style-type: none"> 憑證已過期: 其 notAfter 日期已過。 系統時鐘設定不正確。 	<ol style="list-style-type: none"> 取得新的憑證，並將它插入金鑰環檔中。如果憑證屬於「憑證管理中心」，請將新憑證放入 CA 金鑰環檔中。 請檢查 UTC 系統時鐘是否設為正確的時間。
CertificateNotYetValid 異常狀況	<ol style="list-style-type: none"> 過早使用憑證: 其 notBefore 日期尚未到達。 系統時鐘設定不正確。 	<ol style="list-style-type: none"> 請檢查是否已正確產生並簽署憑證。如果您的組織運作自己的 CA，則 CA 的 UTC 系統時鐘可能不正確。 請檢查 UTC 系統時鐘是否設為正確的時間。
CertificateParsing 異常狀況	<ol style="list-style-type: none"> 憑證包含無效的 DER 資料。 憑證使用不受支援的 DER 特性。 	請確定已正確產生憑證，且可以在 MQIPT 隨附的 IBM Key Management 工具中檢視。請考量取得具有較少憑證延伸的新憑證。
CertificateRevoked 異常狀況	已啟用憑證撤銷檢查，且發現憑證已撤銷。	不應信任有問題的憑證。請取得取代憑證，並確定新的憑證及其私密金鑰存在於金鑰環檔案中。
CertPathBuilderException	憑證鏈不是由已辨識的「憑證管理中心」所簽署。	<ol style="list-style-type: none"> 如果您使用 CA 簽章憑證，請檢查 CA 金鑰環檔中是否存在所有主要 CA 及中繼 CA 憑證。 如果您使用自簽憑證，請確定您已擷取遠端憑證公用部分的副本，並將它新增至 CA 金鑰環檔。避免在正式作業環境中使用自簽憑證。
CertStore 異常狀況 KeyStore 異常狀況	<p>由於下列其中一個原因，從金鑰環讀取憑證時發生錯誤:</p> <ol style="list-style-type: none"> 金鑰環檔案已損壞。 遺漏金鑰環檔。 儲存的密碼不符合金鑰環檔案密碼。 如果路徑配置為使用加密硬體，則 MQIPT 無法連接至加密硬體。 	<ol style="list-style-type: none"> 請確定可以讀取金鑰環檔，且可以使用 IBM Key Management 工具來檢視所有憑證。 請檢查所有金鑰環路徑內容是否都參照正確的檔名。 請檢查儲存的金鑰環檔密碼是否正確。使用 mqiptPW 工具來儲存正確的密碼。 如果路徑配置為使用加密硬體，請檢查下列項目: <ul style="list-style-type: none"> Java 安全內容檔指定已安裝 IBMPKCS11Impl 安全提供者。 Java 安全內容檔包含用來起始設定 IBMPKCS11Impl 安全提供者之配置檔的完整名稱。 用來起始設定 IBMPKCS11Impl 安全提供者的配置檔有效。

異常狀況	原因	動作
SSLException: 沒有可用的憑證或金鑰對應於已啟用的 SSL 密碼組合。	您必須具有個人憑證，且具有所使用 CipherSuites 的正確金鑰類型。例如，名稱以 SSL_ECDH_ECDSA_ 開頭的 CipherSuites 需要具有「橢圓曲線」公開金鑰的憑證。最常用的 CipherSuites 需要具有 RSA 公開金鑰的憑證。	使用 IBM Key Management 工具開啟金鑰環檔案。在「個人憑證」視圖下，依序選取每一個憑證並檢視它。按一下 檢視詳細資料 ，並導覽至「主旨公開金鑰」區段，以查看公開金鑰類型。然後檢查 MQIPT SSLClientCipherSuites 和 SSLServerCipherSuites 路徑內容，以確定已啟用適當的 CipherSuites。
SSLException: 沒有共同的密碼組合 SSLHandshakeException: 沒有共同的密碼組合	信號交換無法同意 CipherSuite，因為在連線兩端啟用的 CipherSuites 集之間沒有重疊。尤其是出埠 IBM MQ 連線只會啟用單一密碼，因此 SSLServer MQIPT 路徑特別可能會遇到此錯誤。 當符合下列三個條件時，也會發生此錯誤： <ul style="list-style-type: none"> 在路徑上未指定任何 CipherSuite 在配置給路徑的金鑰環中找不到適當的網站憑證 已停用匿名 CipherSuites 	檢查 MQIPT SSLClientCipherSuites 及 SSLServerCipherSuites 路徑內容中已啟用 CipherSuites 的清單。請考慮啟用其他 CipherSuites。請參閱提供的表格，以判定要針對每一個 IBM MQ channel CipherSpec 值啟用的正確 CipherSuites。 如果路徑上未指定 CipherSuite，請檢查金鑰環路徑內容是否參照正確的金鑰環檔案，以及金鑰環是否包含 MQIPT 可以使用的個人憑證。如果路徑配置為使用加密硬體，請檢查用來起始設定 IBMPKCS11Impl 安全提供者的配置檔中的 tokenlabel 屬性是否指定正確的加密裝置記號標籤。

MQIPT 中的 mqiptKeyman 和 mqiptKeycmd

mqiptKeyman (iKeyman) 是 IBM MQ 使用者已熟悉的憑證和金鑰管理應用程式。**mqiptKeyman** 及 **mqiptKeycmd** 指令可用來管理 IBM MQ Internet Pass-Thru 所使用金鑰環檔案中的對稱及非對稱金鑰、數位憑證及憑證要求。這些檔案也可以用來管理金鑰環檔案本身。

mqiptKeyman 使用術語 金鑰資料庫 來參照金鑰環檔; 這些術語是同義的。

mqiptKeyman 可以在兩種模式下執行: 圖形使用者介面 (GUI) 和指令行介面 (CLI)。使用 **mqiptKeyman** 指令來啟動 iKeyman GUI，並使用 **mqiptKeycmd** 指令來執行 iKeyman CLI。

用於在 IBM MQ 中管理憑證的對等指令是 **stmqikm** (用於啟動 iKeyman GUI)，以及 **runmqckm** (用於執行 iKeyman CLI)。IBM MQ 指令在 [使用 runmqckm、runmqakm 及 stmqikm 來管理數位憑證](#) 中有說明。

MQIPT 所需的金鑰環檔案格式

建立金鑰環檔以在 MQIPT 中使用時，您必須使用 PKCS #12 檔案格式：

- 在 GUI 中，建立金鑰環檔案時，在 **金鑰資料庫類型** 欄位中選取 PKCS#12。
- 在 CLI 中，包括 **mqiptKeycmd -keydb -create** 指令上的 **-type pkcs12** 參數。

V 9.1.4 MQIPT 也可以存取儲存在支援 PKCS #11 介面之加密硬體中的憑證。此介面也可以用來管理 PKCS #11 硬體上的憑證。如需相關資訊，請參閱 [第 838 頁的『在 MQIPT 中使用 PKCS #11 加密硬體』](#)。

加密 MQIPT 的金鑰環密碼

建立金鑰環檔案之後，您必須以 MQIPT 可用來存取檔案的格式來加密金鑰環密碼。如需此作業的相關資訊，請參閱第 826 頁的『在 MQIPT 中加密金鑰環密碼』。

請注意，MQIPT 不支援隱藏檔機能。您必須使用 `mqiPTPW` 指令來加密金鑰環密碼，而不是使用隱藏檔。

指令行範例

CLI 使用與 IBM MQ `runmqckm` 指令相同的語法。將必要參數附加至 `mqiPTKeycmd`，如下列範例所示：

- 若要建立 PKCS#12 檔案，請執行下列動作：

```
mqiPTKeycmd -keydb -create -db key.p12 -pw password -type pkcs12
```

- 若要建立自簽個人憑證以進行測試，請執行下列動作：

```
mqiPTKeycmd -cert -create -db key.p12 -pw password -type pkcs12  
-label mqiPT -dn "CN=Test Certificate,OU=Sales,O=Example,C=US"  
-sig_alg SHA256WithRSA -size 2048
```

此指令會建立具有 2048 位元 RSA 公開金鑰的數位憑證，以及使用 RSA 與 SHA-256 雜湊演算法的數位簽章。當您建立憑證時，請小心選擇適合您組織安全需求的公開金鑰加密演算法、金鑰大小及數位簽章演算法。如需相關資訊，請參閱第 830 頁的『MQIPT 的數位憑證考量』。

此範例使用適用於測試目的的自簽憑證。不過，在正式作業環境中，您應該改用「憑證管理中心」簽署的憑證。

請注意，MQIPT v2.0 及更舊版本不支援 SHA-2 數位簽章，因此此憑證不適用於建立與舊版 MQIPT 的安全 Socket 連線；需要較舊的簽章演算法 (例如 SHA1WithRSA)。

- 若要建立 CA 簽章憑證的憑證申請以用於正式作業，請執行下列動作：

```
mqiPTKeycmd -certreq -create -db key.p12 -pw password -type pkcs12 -file cert.req  
-label mqiPT -dn "CN=Test Certificate,OU=Sales,O=Example,C=US"  
-sig_alg SHA256WithRSA -size 2048
```

指令會建立具有 2048 位元 RSA 公開金鑰的數位憑證要求，以及使用 RSA 搭配 SHA-256 雜湊演算法的數位簽章。當您建立憑證時，請小心選擇適合您組織安全需求的公開金鑰加密演算法、金鑰大小及數位簽章演算法。如需相關資訊，請參閱第 830 頁的『MQIPT 的數位憑證考量』。

- 若要将 CA 簽署的個人憑證檔案 `cert.crt` 接收到金鑰環檔案中，請執行下列動作：

```
mqiPTKeycmd -cert -receive -db key.p12 -pw password -type pkcs12 -file cert.crt
```

您必須確保簽署個人憑證之 CA 的 CA 憑證存在於 CA 金鑰環檔案中，例如：

```
mqiPTKeycmd -cert -add -db key.p12 -pw password -type pkcs12 -file ca.crt -label rootCA
```

MQIPT 的數位憑證考量

要考量的點包括憑證金鑰大小、選取適當的憑證數位簽章演算法和數位憑證，以及 CipherSuite compatibilityDigital 憑證和 CipherSuite 相容性。

MQIPT 的憑證金鑰大小考量

公開金鑰大小取決於您組織的安全原則，且取決於所使用的加密演算法。一般而言，較大的金鑰大小會更安全。下表列出您應該使用的金鑰大小下限：

演算法	金鑰大小下限 (位元)
橢圓曲線	256
RSA	2048

建立憑證或憑證申請時，請指定憑證的金鑰大小。

- 使用 `mqiptykeycmd` CLI 指令時，`-size` 參數會指定金鑰大小。
- 使用 `mqiptykeyman` GUI 時，「憑證建立」視窗中的 **金鑰大小** 欄位會指定金鑰大小。

選取適當的憑證數位簽章演算法

若要防止偽造數位憑證，請務必使用強大的數位簽章演算法。當您建立或要求憑證時，請小心選取良好的演算法。

您應該避免使用以 MD5 或 SHA-1 為基礎的舊數位簽章演算法，因為這些演算法不再足夠安全，無法現代使用。可能的話，請使用其中一個較新的 SHA-2 型數位簽章演算法，例如 SHA-256 with RSA (SHA256WithRSA)。

不過，2.1 版之前的 MQIPT 版本不支援 SHA-2 數位簽章，因此為了與舊版 MQIPT 的交互作業能力，請使用 SHA1WithRSA 數位簽章演算法。不過，您應該計劃升級舊版 MQIPT，並逐步淘汰使用 MD5 及 SHA-1 數位簽章。

- 使用 `mqiptykeycmd` CLI 指令時，`-sig_alg` 參數會指定數位簽章演算法。
- 使用 `mqiptykeyman` GUI 時，「憑證建立」視窗的 **簽章演算法** 欄位會指定數位簽章演算法。

MQIPT 中的數位憑證及 CipherSuite 相容性

並非所有 CipherSuites 都可以與所有數位憑證搭配使用。有各種類型的 CipherSuite，依其 CipherSuite 名稱字首分組。每一種類型的 CipherSuite 對可使用的數位憑證類型施加不同的限制。這些限制適用於所有 MQIPT SSL/TLS 連線，但特別適用於橢圓曲線加密法的使用者。執行安全 Socket 信號交換時，MQIPT 會自動選取個人憑證，以識別適用於協議 CipherSuite 的個人憑證。在大部分情況下，MQIPT 會自動與遠端對等節點交互作業。不過，在某些情況下，您可能需要使用特定的 MQIPT CipherSuite 來與遠端 IBM MQ 系統交互作業。MQIPT 隨附的 `mqiptykeyman` 應用程式只能使用 DSA 及 RSA 公開金鑰來建立憑證及憑證要求。此外，IBM MQ `runmqakm` 公用程式可以使用「橢圓曲線」公開金鑰來建立憑證及憑證申請。請諮詢您的「憑證管理中心」，以取得建立其他憑證類型的建議。

要使用的數位憑證類型視您使用的 CipherSuite 類型而定：

- 名稱開頭為 `SSL_ECDH_ECDSA_` 及 `SSL_ECDHE_ECDSA_` 的 CipherSuites 需要具有「橢圓曲線」公開金鑰的數位憑證。
- 名稱包含 `anon` 的 CipherSuites 是匿名的；它們不需要數位憑證即可識別遠端對等節點。在使用替代鑑別方法的網路中，此類 CipherSuites 可以避免憑證生命週期管理的額外需要，但通常會避免因缺少鑑別而使用它們。
- 其他 CipherSuites 需要具有 RSA 公開金鑰的數位憑證。

註：`mqiptykeyman` 和 `mqiptykeycmd` 工具無法建立具有「橢圓曲線」公開金鑰的憑證或憑證申請。您可以使用 IBM MQ 隨附的 `runmqakm` 指令來達到此目的。`runmqakm` 指令在 [使用 `runmqckm`、`runmqakm` 及 `strmqikm` 來管理數位憑證](#) 中有說明。

MQIPT 中的憑證結束程式

憑證結束程式的目的是驗證 MQIPT 收到的 SSL/TLS 同層級憑證。

您可以配置 MQIPT 路徑，以在建立新連線時充當 SSL/TLS 用戶端，並在收到連線要求時充當 SSL/TLS 伺服器。在 SSL/TLS 信號交換處理程序期間，SSL/TLS 用戶端會從伺服器接收同層級憑證，且憑證可用來鑑別伺服器。SSL/TLS 伺服器也可以從用戶端接收同層級憑證，憑證可用來鑑別用戶端。

當 MQIPT 收到同層級憑證時，會呼叫憑證結束程式，可讓您執行進一步驗證。MQIPT 會捕捉結束程式所捕捉到的任何異常狀況，且連線要求會終止。因此，最好是讓結束程式捕捉所有異常狀況，並將適當的回覆碼傳回給 MQIPT。

提供範例以顯示可以實作憑證結束程式，如需相關資訊，請參閱 [使用憑證結束程式來鑑別 SSL/TLS 伺服器](#)。

註：MQIPT 在單一 Java virtual machine 中執行，因此使用者定義的憑證結束程式可能會以下列其中一種方式危及 MQIPT 的正常作業：

- 影響系統資源
- 產生瓶頸
- 降低效能

在正式作業環境中實作憑證結束程式之前，您應該廣泛測試憑證結束程式的效果。

MQIPT 中的 *com.ibm.mq.ipt.exit.CertificateExit* 類別

必須由使用 `SSLExitName` 內容定義的類別實作的抽象類別。

類別包含用於執行結束程式的預設實作，以及一些您可以根據需求選擇性地置換的公用方法。受支援方法的完整清單如下：

方法

public int init (IPTTrace)

當 MQIPT 載入結束程式時，MQIPT 會呼叫 `init` 方法，並且可以實作它來執行結束程式的任何起始設定；例如，載入在驗證處理程序期間使用的資料。預設實作不會執行任何動作。

public int refresh (IPTTrace)

會實作重新整理方法來執行任何資料的重新整理；例如，重新載入在驗證處理程序期間使用之磁碟的任何資料。當 MQIPT 管理者已發出重新整理指令時，會呼叫此方法。預設實作不會執行任何動作。

public void close (IPTTrace)

當路徑即將停止或 MQIPT 關閉時，會實作關閉方法來執行任何內部管理。預設實作不會執行任何動作。

public CertificateExitResponse validate (IPTTrace)

會呼叫 `validate` 方法來執行同層級憑證的驗證。傳回物件可用來將資訊傳回 MQIPT；例如，回覆碼及可新增至連線日誌的部分文字。預設實作會傳回含有 `CertificateExitResponse.OK` 的 `CertificateExit` 回應。

支援取得內容的方法：

public int getListenerPort ()

擷取路由接聽器埠-如 `ListenerPort` 內容所定義

public String getDestination()

擷取目的地位址-如「目的地」內容所定義

public int getDestinationPort ()

擷取目的地接聽器埠位址-如 `DestinationPort` 內容所定義

public String getClientIPAddress ()

擷取提出連線要求之用戶端的 IP 位址

public int getClientPortAddress()

擷取提出連線要求的用戶端所使用的埠位址

public boolean isSSLClient()

用來判斷是否以 SSL/TLS 用戶端或 SSL/TLS 伺服器來呼叫結束程式。如果傳回 `true`，則結束程式位於連線的用戶端，並驗證從伺服器取得的憑證。如果傳回 `false`，則結束程式位於連線的伺服器端，並驗證用戶端所傳送的憑證。對於同時充當 SSL/TLS 伺服器和 SSL/TLS 用戶端的路徑而言，它是有效的，可解密及重新加密資料流量。在此狀況下，雖然有單一結束程式類別，但會將類別的部分實例當作用戶端來呼叫，並將部分實例當作伺服器來呼叫。您可以使用 `isSSLClient` 來判斷給定實例的狀況。

public int getConnThreadID()

用來擷取處理連線要求之工作程式執行緒的 ID，這可能有助於除錯。

public String getChannelName ()

會擷取連線要求中使用的 IBM MQ 通道名稱。只有在送入要求未使用 SSL/TLS 且 MQIPT 充當 SSL/TLS 用戶端時，才能使用此選項。

public String getQMName()

擷取連線要求中使用的 IBM MQ 佇列管理程式名稱。只有在用戶端要求未使用 SSL/TLS 且 MQIPT 充當 SSL/TLS 用戶端時，才能使用此選項。

public boolean getTimedout()

結束程式用來判斷逾時是否已過期。

public IPTCertificate getCertificate()

擷取需要驗證的 SSL/TLS 憑證。

public String getExitData ()

擷取結束程式資料，如 SSLExitData 內容所定義。

public String getExitName ()

會擷取 SSLExitName 內容所定義的結束程式名稱。

MQIPT 中的 *com.ibm.mq.ipt.exit.CertificateExitResponse* 類別

此類別用於在驗證憑證之後將資訊傳回 MQIPT。

建構子

public CertificateExitResponse (int rc, string message)

此建構子可用來傳回回覆碼及一些訊息文字。可能的原因碼為

- ExitRc。好的
- ExitRc。有效的錯誤
- ExitRc。已拒絕 VALIDATE_REJECTED

public CertificateExitResponse (int rc)

這個建構子可以用來傳回回覆碼，不含訊息文字。可能的原因碼為

- ExitRc。好的
- ExitRc。有效的錯誤
- ExitRc。已拒絕 VALIDATE_REJECTED

public CertificateExitResponse ()

此建構子可用來傳回回覆碼 ExitRc。好，沒有訊息文字。

方法

public String getVersion()

此方法會傳回此類別的版本。

public String toString

此方法將傳回回應的字串表示法，例如， " Reason code: 4 , Message: Failed CRL check."

MQIPT 中的 *com.ibm.mq.ipt.exit.IPTCertificate* 類別

此類別包含要驗證的 SSL/TLS 憑證。

方法

public int getVersion()

此方法會傳回此類別的版本。

public byte [] getDerEncoding ()

此方法會傳回 X.509 憑證的 ASN.1/DER 編碼，如果發生錯誤，則會傳回 NULL。

public byte [] getPemEncoding ()

此方法會傳回 X.509 憑證的 PEM (BASE64) 編碼，如果發生錯誤，則會傳回 NULL。

public String getLabel()

此方法會傳回憑證標籤，如果發生錯誤，則會傳回空值。

public String getName()

此方法會傳回憑證的「識別名稱」，如果無法使用，則會傳回空值。例如：

```
CN=Test Queue Manager,OU=Sales,O=Example,L=London,C=GB
```

public String getIssuerName ()

此方法會傳回發證者的憑證「識別名稱」，如果無法使用，則會傳回空值。例如：

```
CN=Certificate Authority,OU=Security,O=Example,L=New York,C=US
```

public IPTCertificate getSigner()

此方法會傳回簽章者憑證，如果無法使用，則會傳回空值。對於自簽憑證，它會傳回本身的參照。

public String toString()

此方法會傳回憑證的字串表示法。

MQIPT 中的 *com.ibm.mq.ipc.exit.IPTTrace* 類別

MQIPT 追蹤函數提供進入及結束呼叫，可在進入及結束方法時使用。也有各種資料呼叫可追蹤有用資訊。

方法

public void entry (String fid)

其中 *fid* 用來識別進行呼叫的位置，例如類別和方法名稱。

此方法會將具有適當縮排層次的項目寫入追蹤輸出檔，以記錄控制流程進入方法的點。此呼叫是選用的，但如果使用它，也必須在相同方法內使用 "exit (String)" 的相符呼叫。

public void exit (String fid)

其中 *fid* 用來識別進行呼叫的位置，例如類別和方法名稱。

此方法會將結束程式寫入具有適當縮排層次的追蹤輸出檔，以記錄控制流程離開方法的點。只有在先前已在相同方法內使用對 "entry (String)" 的呼叫時，才會使用此方法。

public void exit (String fid, int rc)

其中 *fid* 用來識別進行呼叫的位置，例如類別及方法名稱，而 *rc* 是來自方法的數值回覆碼。這個追蹤方法應該用來記錄傳回整數之方法的結束程式。

此方法會將結束程式寫入具有適當縮排層次的追蹤輸出檔，以記錄控制流程離開該方法的點，以及該方法的數值回覆碼。只有在先前已在相同方法內使用對 "entry (String)" 的呼叫時，才會使用此方法。

public void exit (String fid, boolean rc)

其中 *fid* 是用來識別進行呼叫的位置，例如類別和方法名稱，而 *rc* 是來自方法的布林回覆碼。這個追蹤方法應該用來記錄從傳回布林的方法結束。

此方法會以適當的縮排層次將結束程式寫入追蹤輸出檔，以記錄控制流程離開方法的點，以及該方法的布林回覆碼。只有在先前已在相同方法內使用對 "entry (String)" 的呼叫時，才會使用此方法。

public void data (String fid, String data)

其中 *fid* 用來識別進行呼叫的位置，例如類別和方法名稱。

此方法會將部分字串資料寫入追蹤輸出檔。

public void data (String fid, int data)

其中 *fid* 用來識別進行呼叫的位置，例如類別和方法名稱。

此方法會將部分整數資料寫入追蹤輸出檔。

public void data (String fid, byte [])

其中 *fid* 用來識別進行呼叫的位置，例如類別和方法名稱。

此方法會將部分二進位資料寫入追蹤輸出檔。

範例追蹤

為了協助在結束程式中診斷問題，您可以使用與 MQIPT 相同的追蹤機能，或者您可以實作自己的追蹤功能。如果您決定使用 MQIPT 追蹤函數，則會有進入及結束呼叫，可在進入及結束方法時使用。還有各種資料呼叫可追蹤有用資訊，如下列範例所示。

```
/**
 * This method is called to initialize the exit (for example, for
 * loading validation information) and place itself in a ready
 * state to validate connection requests.
 */
public int init(IPTTrace t) {
    final String fid = "MyExit.init";

    // Trace entry into this method
    t.entry(fid);

    // Trace useful information
    t.data(fid, "Starting exit - MQIPT version " + getVersion());

    // Perform initialization and load any data
    t.data(fid, "Ready for work");

    // Trace exit from this method
    t.exit(fid);

    return ExitRc.OK;
}
```

此方法會以下列範例中所示的格式產生追蹤：

```
16:36:48.625    14    5000-1s    -----{ ConnectionThread.setCertificateExit()
16:36:48.625    14    5000-1s    Creating instance of certificate exit
16:36:48.625    14    5000-1s    Calling init() of certificate exit
16:36:48.625    14    5000-1s    -----} MyExit.init()
16:36:48.625    14    5000-1s    Starting exit - MQIPT version 2.1.0.0
16:36:48.625    14    5000-1s    Ready for work
16:36:48.625    14    5000-1s    -----} MyExit.init() rc=0
16:36:48.625    14    5000-1s    -----} ConnectionThread.setCertificateExit() rc=0
```

MQIPT 中的憑證結束程式回覆碼

在許多不同狀況下呼叫憑證結束程式時，MQIPT 可辨識的回覆碼。

在下列情況下，當呼叫憑證結束程式時，MQIPT 會辨識下列回覆碼：

回覆碼	說明	起始設定	驗證	重新整理
ExitRc. 好的	已順利完成要求。	yes	yes	yes
ExitRc. INIT_ERROR	起始設定要求失敗，將停用路徑。	yes		
ExitRc. REFRESH_ERROR	重新整理要求失敗，將停用路徑。			yes
ExitRc. 有效的錯誤	驗證程序失敗，連線要求被拒絕。		yes	
ExitRc. 已拒絕 VALIDATE_REJECTED	驗證要求被拒絕，連線要求被拒絕。		yes	

MQIPT 中的 LDAP 及 CRL

MQIPT 支援使用「輕量型目錄存取通訊協定 (LDAP)」伺服器，對數位憑證執行「憑證撤銷清冊 (CRL)」鑑別。

LDAP 支援的實作方式與 IBM MQ 中的實作方式類似，因為相同的 LDAP 伺服器可以同時用於 IBM MQ 和 MQIPT。

在 SSL/TLS 信號交換期間，通訊夥伴會使用數位憑證彼此鑑別。鑑別時可能會檢查收到的憑證是否仍可信任。憑證管理中心 (CA) 基於各種原因撤銷憑證，包括下列各項：

- 擁有者已移至不同的組織。
- 私密金鑰不再是秘密金鑰。

CA 在「憑證撤銷清冊 (CRL)」中發佈撤銷的個人憑證。已撤銷的 CA 憑證會發佈在「權限撤銷清單 (ARL)」中。請注意，後續對 CRL 的參照也適用於 ARL。

如需搭配使用 LDAP 伺服器與 IBM MQ 以及 CRL 及 ARL 管理的相關資訊，請參閱 [使用憑證撤銷清冊及權限撤銷清冊](#)。

MQIPT 在每一個路徑上最多可以支援兩部 LDAP 伺服器。第一個 LDAP 伺服器會被視為主要伺服器，而第二個 LDAP 伺服器會保留作為備份。只有在無法呼叫到主要伺服器時，才會使用第二部伺服器。備用伺服器應該是主要伺服器的鏡映映像檔。

使用 LDAP 使用者 ID 和密碼內容，可以利用使用者 ID 和密碼來保護 LDAP 伺服器上所儲存資訊的存取權。

V9.1.5 在 IBM MQ 9.1.5 的 MQIPT 配置中，LDAP 伺服器密碼可以加密。如需加密 MQIPT 所使用的密碼的相關資訊，請參閱 [加密儲存的密碼](#)。

當 MQIPT 從金鑰環檔載入 PKCS #12 記號時，會檢查任何 CA 憑證的 CRL 有效性。如果 CA 憑證具有附加 CRL，則會檢查它是否已過期，如果已過期，則會從 LDAP 伺服器擷取較新的 CRL。任何擷取的 CRL 都會載入現行記號中，並附加至其 CA 憑證。

當查詢傳送至主要 LDAP 伺服器時，如果沒有符合給定 CA 的項目，則會假設該 CA 沒有 CRL，且不會使用備用伺服器。不過，如果無法呼叫到主要 LDAP 伺服器，或未在給定的時間範圍內傳回，則會使用備用伺服器。來自備用伺服器的任何錯誤都會導致用戶端連線結束。可以透過將 **LDAPIgnoreErrors** 內容設為 true 來置換此動作。

MQIPT 所擷取的任何 CRL 都會保留在快取中，並由該路徑上的所有連線共用。如果快取的 CRL 已過期，則會從快取中移除 CRL，並從 LDAP 伺服器擷取新的 CRL。如果沒有新的 CRL 可用，仍會拒絕連線。

也會檢查從 LDAP 伺服器擷取的 CRL 是否到期，並顯示警告訊息 (MQCPW001)。過期 CRL 仍會載入系統中，且會拒絕任何參照該 CRL 的連線要求。您應該將 LDAP 伺服器中過期的 CRL 取代為現行 CRL。

LDAPCacheTimeout 內容可用來控制 CRL 快取清除的頻率。預設值為 1 天。將此值設為 0 表示在重新啟動路徑之前不會清除快取項目。

過期 CRL 可以儲存在金鑰環檔或 LDAP 伺服器上。如果尚未發出新的 CRL，則會拒絕進一步連線要求。您可以透過啟用 **IgnoreExpiredCRLs** 內容來忽略過期 CRL。

註：如果您啟用 **LDAPIgnoreErrors** 內容或 **IgnoreExpiredCRLs** 內容，則可能會使用已撤銷的憑證來建立 SSL/TLS 連線。

MQIPT 中的多值憑證識別名稱 OU 內容

您可以比對憑證識別名稱中的多個組織單位 (OU) 值。

下列路由內容現在支援多個 OU 值的比對：

- **SSLClientDN_OU**
- **SSLClientSiteDN_OU**
- **SSLServerDN_OU**
- **SSLServerSiteDN_OU**

若要符合多個 OU 值，請在路由內容值中使用逗點作為分隔字元。 例如：

```
SSLClientDN_OU=Sales, Europe
```

這會比對同時具有 OU=Sales 和 OU=Europe 的憑證。 OU 值與 IBM MQ SSLPEER 過濾器中的多個 OU 值以相同順序進行比對。

請勿在 [route] 區段中多次指定相同的路由內容。 符合多個 OU 值的正確方式是指定內容一次，如前述範例所示。 如果您在相同的 mqipt.conf 區段中多次輸入相同的屬性，則最後一個值會生效。 例如，下列項目只會導致符合 歐洲，因為第二行會置換第一行：

```
SSLClientDN_OU=Sales
SSLClientDN_OU=Europe
```

如果您必須符合 OU 值內的文字逗點，請緊接在逗點之前插入反斜線 (\) 作為跳出字元。 例如：

```
SSLClientDN_OU=Sales\, Europe
```

這符合單一值: OU=Sales, Europe。 後面未緊接逗點的反斜線符合文字反斜線。

如果您要從舊版 MQIPT 進行升級，並依賴在 OU 值中符合逗點的能力，則必須在 OU 路徑內容中插入反斜線跳出字元，以保留先前的行為。

在 MQIPT 中啟用已淘汰的通訊協定及 CipherSuites

依預設，MQIPT 隨附的 Java runtime environment (JRE) 中會停用安全 Socket 通訊協定及視為不安全的 CipherSuites。 必須先啟用這些已淘汰的通訊協定及 CipherSuites，然後才能使用它們。

關於這項作業

如果您知道潛在危險，但仍需要使用 MQIPT 中視為不安全的其中一個通訊協定或 CipherSuites，請遵循此程序來啟用您需要使用的通訊協定或 CipherSuite。

程序

1. 編輯 java.security 檔案，位於 *mqipt_path*/java/jre/lib/security 目錄中，其中 *mqipt_path* 是 MQIPT 的安裝位置。
2. 從 jdk.tls.disabledAlgorithms 內容中已停用的演算法清單移除對應項目，以將通訊協定或演算法的支援新增至 JRE。
 - **V 9.1.4** 若要新增通訊協定支援，請從已停用的演算法清單中移除通訊協定。 例如，若要新增 TLS 1.0 的支援，請從清單中移除 TLSv1。
 - 若要新增對 CipherSuite 的支援，請從已停用演算法清單中移除對應的演算法。 例如，若要新增對 SSL_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA 密碼組合的支援，請從清單中移除 3DES_EDE_CBC。
3. **V 9.1.4**
如果要在 JRE 中啟用 SSL 3.0，您也需要設定系統內容 com.ibm.jsse2.disableSSLv3=false。 您可以使用環境變數 **MQIPT_JVM_OPTIONS** 來設定內容。 例如：

```
set MQIPT_JVM_OPTIONS=-Dcom.ibm.jsse2.disableSSLv3=false
```

4. 若要在 MQIPT 路徑上啟用 SSL 3.0、TLS 1.0 或 TLS 1.1，請將對應的通訊協定新增至 **SSLServerProtocols** 或 **SSLClientProtocols** 路徑內容。
5. 重新啟動 MQIPT，使 JRE 內容的變更生效。

V 9.1.4 在 MQIPT 中使用 PKCS #11 加密硬體

MQIPT 可以存取儲存在支援 PKCS #11 介面之加密硬體中的數位憑證。

開始之前

在開始配置 MQIPT 以使用加密硬體之前，請確定加密卡、卡驅動程式及任何相關聯的支援軟體都已安裝且正常運作。

IBM Java PKCS11 加密提供者 (IBMPKCS11Impl 提供者) 提供 MQIPT 中的 PKCS #11 加密硬體支援。如需 IBMPKCS11Impl 提供者以及 Java 8 支援的加密卡清單的相關資訊，請參閱 [IBM PKCS11 加密提供者](#)。

關於這項作業

您可以將 MQIPT 所存取的個人憑證及 CA 憑證儲存在加密硬體金鑰儲存庫中。不過，由於 PKCS #11 裝置通常沒有足夠的可用空間來儲存大量簽章者憑證，因此您可能想要對 CA 憑證使用個別的檔案型金鑰儲存庫。

請遵循此程序，將 MQIPT 配置成使用加密硬體金鑰儲存庫中的憑證。

註： 搭配使用加密硬體與 MQIPT 是一項 IBM MQ Advanced 功能。若要使用此功能，也需要使用 MQIPT 路徑連接的本端佇列管理程式具有 IBM MQ Advanced、IBM MQ Appliance 或 IBM MQ Advanced for z/OS VUE 授權。

程序

1. 建立在起始設定 IBMPKCS11Impl 提供者時使用的配置檔。

下載 IBMPKCS11Impl 提供者所支援的每一個硬體加密卡的範例配置檔，並為您的系統配置範例。範例可以從 IBM Documentation 中的下列 Java 主題下載：[配置檔](#)。

配置檔是文字檔，應該至少包含下列屬性：

名稱

提供者實例的名稱字尾。

程式庫

加密硬體隨附的 PKCS #11 程式庫完整名稱。

tokenlabel

PKCS #11 加密裝置記號標籤。

例如，配置檔可能包含下列項目：

```
name = IPTPKCS11Provider
library = /usr/lib64/pkcs11/PKCS11_API.so
tokenlabel = icatoken
```

2. 編輯 Java 安全內容檔 `java.security`，位於 MQIPT 安裝目錄的 `java/jre/lib/security` 子目錄中。

- a) 如果檔案中還沒有它，請新增 IBMPKCS11Impl 安全提供者。

例如，新增下列這一行：

```
security.provider.12=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl
```

- b) 在提供者名稱之後新增配置檔的完整名稱。

例如，如果您在步驟 [第 838 頁](#) 的『1』中建立的配置檔稱為 `/opt/mqipt/pkcs11.cfg`，則應該將此路徑新增至與安全提供者相同的行：

```
security.provider.12=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl /opt/mqipt/
pkcs11.cfg
```

3. 如果您使用 CA 憑證的金鑰環檔，而不是將 CA 憑證儲存在加密硬體中，請建立 PKCS #12 格式的 CA 金鑰環檔。

您可以使用 **mqiPTKeyman** 圖形使用者介面 (GUI) 或 **mqiPTKeycmd** 指令行介面 (CLI) 來建立 CA 金鑰環檔案。

- 若要使用 CLI，請輸入下列指令：

```
mqiPTKeycmd -keydb -create -db filename -pw password -type pkcs12
```

其中 *filename* 是要建立的金鑰環檔案的名稱，而 *password* 是金鑰環密碼。

- 若要使用 GUI，請遵循下列步驟：
 - a. 透過輸入指令 **mqiPTKeyman** 來啟動 GUI。
 - b. 按一下 **金鑰資料庫檔 > 開啟**。
 - c. 按一下 **金鑰資料庫類型**，然後選取 **PKCS11Config**。
 - d. 按一下 **確定**。即會開啟「開啟加密記號」視窗。
 - e. 選取您要用來儲存憑證的加密裝置記號標籤。
 - f. 在 **加密記號密碼** 欄位中，輸入存取加密硬體所需的密碼。
 - g. 若要建立新的 CA 金鑰環檔，請選取 **建立新的次要金鑰資料庫檔**。
 - h. 按一下 **金鑰資料庫類型**，然後選取 **PKCS12**。
 - i. 在 **檔名** 欄位中，鍵入 CA 金鑰環的檔名。
 - j. 在 **位置** 欄位中，鍵入 CA 金鑰環檔的完整路徑。
 - k. 按一下 **確定**。這時會開啟「密碼提示」視窗。
 - l. 在 **密碼** 欄位中輸入 CA 金鑰環的密碼，然後在 **確認密碼** 欄位中再次輸入它。
 - m. 按一下 **確定**。
4. 使用 **mqiPTKeycmd** 或 **mqiPTKeyman**，要求加密硬體的個人憑證。
- 若要使用 CLI，請輸入下列指令：

```
mqiPTKeycmd -certreq -create -crypto module_name -tokenlabel hardware_token  
-pw password -label label -size key_size  
-sig_alg algorithm -dn distinguished_name -file filename
```

其中：

-crypto module_name

指定加密硬體隨附的 PKCS #11 檔案庫的完整名稱。

-tokenlabel token_label

指定 PKCS #11 加密裝置記號標籤。

-pw password

指定用於存取加密硬體的密碼。

-label label

指定憑證標籤。

-size key_size

指定金鑰大小。值可以是 512、1024、2048 或 4096。

-sig_alg algorithm

指定用於建立項目金鑰組的非對稱簽章演算法。值可以是 MD2_WITH_RSA、MD2WithRSA、MD5_WITH_RSA、MD5WithRSA、SHA1WithDSA、SHA1WithECDSA、SHA1WithRSA、SHA2/ECDSA、SHA224WithECDSA、SHA256_WITH_RSA、SHA256WithECDSA、SHA256WithRSA、SHA2WithECDSA、SHA3/ECDSA、SHA384_WITH_RSA、SHA384WithECDSA、SHA384WithRSA、SHA3WithECDSA、SHA5/ECDSA、SHA512_WITH_RSA、SHA512WithECDSA、SHA512WithRSA、SHA5WithECDSA、SHA_WITH_DSA、SHA_WITH_RSA 或 SHAWithDSA。預設值為 SHA256WithRSA。

-dn distinguished_name

指定以雙引號括住的 X.500 識別名稱。

-file filename

指定憑證申請的檔名。

- 若要使用 GUI，請遵循下列步驟：
 - a. 從 **建立** 功能表中，按一下 **新建憑證申請**。
 - b. 在 **金鑰標籤** 欄位中，輸入憑證標籤。
 - c. 選取您需要的 **金鑰大小** 和 **簽章演算法**。
 - d. 輸入 **通用名稱** 和 **組織** 的值，然後選取 **國家/地區**。對於其餘選用欄位，請接受預設值，或鍵入或選取新值。
 - e. 在 **輸入要在其中儲存憑證申請的檔案名稱** 欄位中，接受預設值 `certreq.arm`，或鍵入具有完整路徑的新值。
 - f. 按一下 **確定**。
 - g. **個人憑證申請** 清單會顯示您所建立之新個人憑證申請的標籤。憑證申請會儲存在您選擇的檔案中。
- 5. 在 CA 傳送個人憑證給您之後，請將 CA 憑證新增至加密金鑰儲存庫或 CA 金鑰環檔 (如果尚未存在的話)。

- 若要使用 CLI 將 CA 憑證新增至 CA 金鑰環檔，請輸入下列指令：

```
mqiptKeycmd -cert -add -db filename -pw password -type pkcs12  
            -label label -file cert_filename
```

其中 *filename* 是 CA 金鑰環檔案的名稱，*password* 是 CA 金鑰環密碼，*label* 是附加至憑證的標籤，而 *cert_filename* 是包含 CA 憑證的檔案名稱。

- 若要使用 CLI 將 CA 憑證新增至加密硬體，請輸入下列指令：

```
mqiptKeycmd -cert -add -crypto module_name -tokenlabel hardware_token  
            -pw password -label label -file cert_filename
```

其中 *module_name* 是隨加密硬體提供的 PKCS #11 程式庫的完整名稱，*hardware_token* 是 PKCS #11 加密裝置記號標籤，*password* 是用於存取加密硬體的密碼，*label* 是附加至憑證的標籤，*cert_filename* 是包含 CA 憑證的檔案名稱。

- 若要使用 GUI，請遵循下列步驟：
 - a. 在 **金鑰資料庫內容** 欄位中，選取 **簽章者憑證**。
 - b. 按一下 **新增**。這時會開啟「從檔案新增 CA 憑證」視窗。
 - c. 鍵入憑證檔名以及儲存憑證的位置，或按一下 **瀏覽** 以選取名稱及位置。
 - d. 按一下 **確定**。這時會開啟「輸入標籤」視窗。
 - e. 在「輸入標籤」視窗中，鍵入憑證的名稱。
 - f. 按一下 **確定**。憑證已新增至金鑰資料庫。
- 6. 將 CA 提供的個人憑證接收到加密硬體金鑰儲存庫中。

- 若要使用 CLI，請輸入下列指令：

```
mqiptKeycmd -cert -receive -file filename -crypto module_name  
            -tokenlabel hardware_token -pw password
```

其中 *filename* 是包含要接收的憑證的檔案名稱，*module_name* 是隨加密硬體提供的 PKCS #11 程式庫的完整名稱，*hardware_token* 是 PKCS #11 加密裝置記號標籤，而 *password* 是用於存取加密硬體的密碼。

如果 CA 憑證儲存在 CA 金鑰環中，而不是儲存在加密硬體中，則您會收到無法驗證憑證鏈的警告，因為 `mqiptKeycmd` 指令在接收加密金鑰儲存庫中的個人憑證時無法存取 CA 金鑰環。

- 若要使用 GUI，請遵循下列步驟：
 - a. 按一下 **接收**。即會開啟「從檔案接收憑證」視窗。
 - b. 鍵入新個人憑證的憑證檔名及位置，或按一下 **瀏覽** 以選取名稱及位置。

c. 按一下**確定**。**個人憑證** 欄位會顯示您新增之個人憑證的標籤。

7. 使用 **mqiPTPW** 指令加密密碼以存取加密硬體。

V 9.1.5 輸入下列指令：

```
mqiPTPW -sf encryption_key_file
```

其中 *encryption_key_file* 是包含 MQIPT 安裝之密碼加密金鑰的檔案名稱。如果您的 MQIPT 安裝使用預設密碼加密金鑰，則不需要指定 **-sf** 參數。鍵入密碼，以在提示時存取要加密的加密硬體。

如需加密金鑰儲存庫密碼的相關資訊，請參閱第 826 頁的『在 MQIPT 中加密金鑰環密碼』。

8. 如果您在步驟第 838 頁的『3』中建立 CA 金鑰環檔，請遵循步驟第 841 頁的『7』中的指示來加密 CA 金鑰環檔的密碼。

9. 編輯 *mqiPT.conf* 配置檔。

a) 將 **EnableAdvancedCapabilities** 廣域內容設為 **true**，以確認您具有適當的授權來使用此 IBM MQ Advanced 特性。

b) 將一或多個 **SSLServerKeyRingUseCryptoHardware**、**SSLServerCAKeyRingUseCryptoHardware**、**SSLServerKeyRingUseCryptoHardware** 或 **SSLServerKeyRingUseCryptoHardware** 內容設為 **true**，即可在路徑上使用加密硬體金鑰儲存庫。

如需在路徑上啟用加密硬體之內容的相關資訊，請參閱 [MQIPT 路徑內容](#)。

c) 如果您使用 CA 憑證的金鑰環檔，請設定一個以上 **SSLServerCAKeyRing** 或 **SSLServerCAKeyRing** 內容來指定 CA 金鑰環的位置。

如果您已配置路徑來使用網站憑證的加密硬體，且未指定 CA 金鑰環檔，則會使用加密硬體金鑰儲存庫作為 CA 金鑰儲存庫。

d) 使用 **SSLServerKeyRingPW**、**SSLServerCAKeyRingPW**、**SSLClientKeyRingPW** 或 **SSLClientCAKeyRingPW** 內容來指定加密密碼，以存取加密硬體及 CA 金鑰環。

V 9.1.5 將 **SSL*KeyRingPW** 內容的值設定為 **mqiPTPW** 指令所輸出的已加密密碼。

e) 如果加密硬體包含多個個人憑證，請指定 MQIPT 應該選取哪個憑證來傳送至 SSL/TLS 伺服器或用戶端進行鑑別。

您可以透過為 SSL/TLS 用戶端路徑設定一個以上 **SSLClientSite*** 內容，或為 SSL/TLS 伺服器路徑設定一個以上 **SSLServerSite*** 內容，來指定應該選取哪些憑證。

如需從金鑰環選取憑證的相關資訊，請參閱第 826 頁的『在 MQIPT 中從金鑰環檔選取憑證』。
[MQIPT 路徑內容](#)中說明從金鑰環選取憑證的內容。

V 9.1.5 例如，若要對 TLS 伺服器路徑上的網站憑證使用加密硬體金鑰儲存庫，以及使用金鑰環檔案來儲存相同路徑的 CA 憑證，請將下列內容新增至路徑定義：

```
SSLServerKeyRingUseCryptoHardware=true
SSLServerKeyRingPW=<mqiPTPW>1!g0RdM4wft5d1rCgNMDEGag==!dZxhgQD2A8Ea0yEqawQvPg==
SSLServerCAKeyRing=/opt/mqiPT/ssl/ca.pfx
SSLServerCAKeyRingPW=<mqiPTPW>1!3Vdripiu6kMwn0sWRCVgT5g==!LH1tGLEg30FvN8+02Re0YA==
SSLServerSiteLabel=mqiPTsite
```

10. 重新啟動 MQIPT。

Java security manager 輸入 MQIPT

Java security manager 可以與任何 MQIPT 特性搭配使用，以提供進一步的安全層次。

MQIPT 使用 `java.lang.SecurityManager` 類別中定義的預設 Java security manager。可以使用廣域內容 **SecurityManager** 來啟用或停用 MQIPT 中的 Java security manager 特性。如需相關資訊，請參閱 [MQIPT 廣域內容](#)。

Java security manager 使用兩個預設原則檔：

- 主機上虛擬機器的所有實例都會使用名為 `$MQIPT_PATH/java/jre/lib/security/java.policy` 的廣域系統原則檔 (其中 `$MQIPT_PATH` 是 MQIPT 的安裝目錄)。

- 稱為 `.java.policy` 的使用者特定原則檔案，可以存在於使用者的起始目錄中。

也可以使用其他 MQIPT 原則檔案。您應該使用 MQIPT 原則檔，而非先前說明的預設原則檔。如需相關資訊，請參閱 MQIPT 廣域內容中的 **SecurityManagerPolicy**。

原則檔的語法相當複雜，雖然可以使用文字編輯器來變更它，但通常更容易使用 Java 隨附的「原則工具」公用程式來進行任何變更。「原則工具」公用程式可以在 `$MQIPT_PATH/java/jre/bin` 目錄中找到，並在 Java 說明文件中完整記載。

MQIPT 已提供範例原則檔 (`mqiptSample.policy`)，以顯示必須設定哪些許可權來執行 MQIPT。

您必須編輯範例原則檔，以符合您的配置。請特別注意，MQIPT 起始目錄 (`mqipt.conf` 的位置) 可能與 MQIPT 安裝目錄不同，因此在安全原則中配置 `FilePermission` 項目時，請小心指定正確的目錄。

您必須變更下列項目：

- 授與 `errors` 目錄讀取和寫入權的 **java.io.FilePermission** 項目。此項目中的檔案路徑必須參照 MQIPT 起始目錄，因為這是 `errors` 目錄所在的位置。MQIPT 會在 `errors` 目錄中建立「FFST 失敗資料擷取」檔案 (`AMQ*.FDC`) 及追蹤檔案 (`AMQ*.TRC*`)。您必須確保 MQIPT 有權在 `errors` 目錄中建立追蹤和 FFST 檔案，以便可以進行疑難排解。
- **java.io.FilePermission** 項目，授與日誌目錄的讀取及寫入權。此項目中的檔案路徑必須參照 MQIPT 起始目錄，因為這是日誌目錄所在的位置。如果啟用 `ConnectionLog` 廣域內容，MQIPT 會在 `logs` 目錄中建立連線日誌檔 (`mqipt*.log`)。
- **java.io.FilePermission** 項目，授與對 MQIPT 安裝目錄中任何目錄 (例如 `bin`、`exits`、`lib` 及 `ssl` 目錄) 的讀取及執行存取權。這些項目中的檔案路徑必須變更為參照 MQIPT 安裝目錄。如果不需要這些項目，則可以省略其中的部分項目。
- 必須修改 **java.net.SocketPermission** 項目，以控制每一個接聽 MQIPT 路徑的連線。每個 MQIPT 路徑的接聽器埠和接聽器位址都需要接聽及接受許可權。
- 必須修改 **java.net.SocketPermission** 項目，以控制每一個 MQIPT 路徑之外的連線。MQIPT 路徑所連接的任何路徑目的地、Proxy 伺服器或 LDAP 伺服器都需要連接許可權。使用主機名稱指定位址時需要解析許可權。

視您的配置而定，您可能還需要新增下列項目：

- **java.io.FilePermission** 項目，用於授與 `mqipt.conf` 的讀取權，或包含 `mqipt.conf` 的 MQIPT 起始目錄。如果您需要使用「管理用戶端」從遠端配置 MQIPT，則 MQIPT 也需要 `mqipt.conf` 的寫入權，才能儲存配置變更。
- **java.io.FilePermission** 項目，用於授與對安全原則檔案本身的讀取權。如果 MQIPT 重新整理會導致重新讀取安全原則檔，這會很有用。
- 部分 **java.io.FilePermission** 項目，用於授與對任何 SSL/TLS 金鑰環檔案及密碼隱藏檔的讀取權。只有在使用已啟用 `SSLClient` 或 `SSLServer` 內容的路徑時，才需要這樣做。
- 部分 **java.io.FilePermission** 項目，用於授與對任何 MQIPT 結束程式類別的讀取或執行存取權。只有在啟用 MQIPT 結束程式時，才需要此項目。如果結束程式需要，您可能需要授與其他許可權。

註: Windows **java.io.FilePermission** 項目必須針對路徑中的每一個反斜線使用兩個反斜線字元 (`\\`)。這是因為使用單一反斜線作為跳出字元。

範例檔假設 MQIPT 已安裝在 `C:\Program Files\IBM\MQ Internet Pass-Thru` 中的 Windows 系統上。它也假設 MQIPT 起始目錄 (`mqipt.conf` 檔案的位置) 與 MQIPT 安裝目錄相同。

如果您已在另一個位置安裝 MQIPT，則必須變更 `codeBase` 定義中的目錄，以參照 MQIPT 安裝目錄。請小心包含正確的字首 (`file:/`) 和正確的檔案字尾 (`/lib/com.ibm.mq.ipt.jar`)。在 UNIX and Linux 系統上，假設 MQIPT 安裝在 `/opt/mqipt` 中，一般 `codeBase` URL 可能是 `file:/opt/mqipt/lib/com.ibm.mq.ipt.jar`。

通常使用三個屬性來定義許可權。若要控制 Socket 連線，其值為：

類別許可權

`java.net.SocketPermission`

要控制的名稱

這是由 `hostname:port` 格式所組成，其中名稱的每一個元件都可以透過萬用字元來指定。主機名稱可以是網域名稱或 IP 位址。主機名稱的最左側位置可由星號 (*) 指定。例如，`harry.company1.com` 將由下列每一個字串進行比對：

- `harry`
- `harry.company1.com`
- `*.company1.com`
- `*`
- `198.51.100.123` (假設這是 `harry.company1.com` 的 IP 位址)

名稱的埠元件可以指定為單一埠位址或埠位址範圍，例如：

1414

僅埠 1414

1414-

所有大於或等於 1414 的埠位址

-1414

所有埠位址都小於或等於 1414

1-1414

介於 1 和 1414 (含) 之間的所有埠位址

容許的動作

`java.net.SocketPermission` 使用的動作如下：

- 接受，這容許許可權接受來自指定目標的連線
- 連接，這容許連接至指定目標的許可權
- 接聽，這容許在一或多個指定埠上接聽連線要求的許可權
- 解析，這容許使用 DNS 名稱服務將網域名稱解析為 IP 位址的許可權

也可以透過 `java.security.manager` 和 `java.security.policy` Java 系統內容來控制 Java security manager，但建議您使用 `SecurityManager` 和 `SecurityManager` 原則內容來控制 MQIPT。

若要在追蹤和 FFST 記錄中包括診斷資訊，MQIPT 必須存取某些 MQIPT 系統內容和環境變數。您必須一律在 Java 安全原則中包含下列內容：

```
permission java.util.PropertyPermission "java.home", "read";
permission java.util.PropertyPermission "java.version", "read";
permission java.util.PropertyPermission "java.runtime.version", "read";
permission java.util.PropertyPermission "java.vm.info", "read";
permission java.util.PropertyPermission "java.vm.vendor", "read";
permission java.util.PropertyPermission "os.arch", "read";
permission java.util.PropertyPermission "os.name", "read";
permission java.util.PropertyPermission "os.version", "read";
permission java.lang.RuntimePermission "getenv.MQIPT_PATH";
permission java.lang.RuntimePermission "getStackTrace";
```

如果您未包含所有這些內容，MQIPT 將無法正常運作，且問題診斷將會受損。

MQIPT 中的安全結束程式

使用安全結束程式來控制對目標目的地的存取權，如 **Destination** route 內容所定義。當 MQIPT 收到來自用戶端的連線要求時，但在與目標目的地建立連線之前，會呼叫安全結束程式。

根據起始連線內容，安全結束程式會決定是否容許完成連線。

當啟動路徑時，會呼叫安全結束程式，以起始設定並準備好處理連線要求。應該使用起始設定處理程序來載入任何使用者資料，並準備此資料以快速且輕鬆地存取，從而最小化處理連線要求所花費的時間。

每一個路徑都可以有自己的安全結束程式。

- **SecurityExit** 內容用來啟用/停用使用者定義安全結束程式。

- **SecurityExitName** 內容用來定義使用者定義安全結束程式的類別名稱。
- **SecurityExitPath** 內容用來定義包含類別檔的目錄名稱。如果未設定此內容，則會假設在 `exits` 子目錄中找到類別檔。**SecurityExitPath** 也可以定義包含使用者定義安全結束程式之 JAR 檔的名稱。
- MQIPT 使用 **SecurityExitTimeout** 內容來決定在驗證連線要求時應等待安全結束程式回應的時間長度。

如需安全結束程式內容的詳細資料，請參閱 [MQIPT 路由內容](#)。

MQIPT 使用 `SecurityExit` 類別來呼叫使用者定義安全結束程式。此類別必須由使用者定義的安全結束程式延伸，且其大部分方法會置換，以提供所需的功能。`SecurityExitResponse` 物件用來將資料傳回給 MQIPT，而 MQIPT 會使用此資料來決定是否應接受或拒絕連線要求。`SecurityExitResponse` 物件也可以包含新的目的地和目的地埠位址，用來置換安全結束程式內容所定義的路徑。

提供三個範例安全結束程式，以顯示如何實作安全結束程式。

- `SampleSecurityExit` 顯示如何根據 IBM MQ 通道的名稱來控制對 IBM MQ 佇列管理程式的存取權。它只容許通道名稱以字串 "MQIPT" 開頭的連線。如需相關資訊，請參閱 [使用安全結束程式](#)。
- `SampleRoutingExit` 容許將用戶端連線要求動態遞送至已定義 IBM MQ 伺服器的儲存區，每一部伺服器管理具有相同名稱及相同屬性的佇列管理程式。範例包括包含伺服器名稱清單的配置檔。如需相關資訊，請參閱 [使用安全結束程式將用戶端連線要求遞送至 IBM MQ 佇列管理程式伺服器](#)。
- `SampleOneRouteExit` 容許動態遞送至 IBM MQ 佇列管理程式，此佇列管理程式衍生自連線要求中使用的 IBM MQ 通道名稱。此範例包括一個配置檔，其中包含佇列管理程式名稱至伺服器名稱的對映。如需相關資訊，請參閱 [動態遞送用戶端連線要求](#)。

註: MQIPT 在單一 JVM 中執行，因此使用者定義的安全結束程式可能會以下列其中一種方式危及 MQIPT 的正常作業：

- 影響系統資源
- 產生瓶頸
- 降低效能

在正式作業環境中實作安全結束程式之前，您應該廣泛測試安全結束程式的效果。

MQIPT 中的 `com.ibm.mq.ipt.exit.SecurityExit` 類別

此類別及其公用方法必須由使用者定義安全結束程式延伸，以取得部分一般資料的存取權，並容許進行部分 MQIPT 起始設定。

在 MQIPT 呼叫每一個方法之前，部分內容將可供方法使用。可以使用此類別中定義的適當 `get` 方法來擷取其值。

方法

`public int init (IPTTrace)`

可用的內容如下：

- 接聽器埠 (listener port)
- destination
- 目的地埠
- version

當路徑啟動時，MQIPT 會呼叫 `init` 方法。從這個方法返回時，安全結束程式必須準備好驗證連線要求。有效的可能回覆碼為 `ExitRc`。OK 或 `ExitRc`。INIT_ERROR。

`public int refresh (IPTTrace)`

可用的內容如下：

- 接聽器埠 (listener port)
- destination
- 目的地埠

當 IPT Administration Client 要求自行重新整理時，MQIPT 會呼叫重新整理方法。當配置檔中的內容已變更時，通常會呼叫此動作。MQIPT 會從配置檔載入所有內容，並判斷哪些內容已變更，以及是否需要立即重新啟動路徑，或是否可以等到 MQIPT 下次重新啟動。

此方法應該對它所使用的任何外部資料 (亦即，在 init 方法期間載入的資料) 執行重新載入。有效的可能回覆碼為 ExitRc。OK 或 ExitRc。REFRESH_ERROR。

public void close (IPTTrace)

可用的內容如下：

- 接聽器埠 (listener port)
- destination
- 目的地埠

當 MQIPT IPT Administration Client 要求停止時，MQIPT 會呼叫 close () 方法。它應該釋放在作業期間所獲得的任何系統資源。MQIPT 將在關閉之前等待此方法完成。

如果已啟用安全結束程式，但現在已在配置檔中停用，則也會呼叫此方法。

public SecurityExitResponse validate (IPTTrace)

可用的內容如下：

- 接聽器埠 (listener port)
- destination
- 目的地埠
- timeout
- 用戶端 IP 位址
- 用戶端埠位址
- 通道名稱
- 佇列管理程式名稱

當 MQIPT 收到要驗證的連線要求時，它會呼叫 validate 方法。如果已啟用 SSLProxyMode 內容，則無法使用通道名稱及佇列管理程式名稱，因為此特性僅用於通道 SSL/TLS 資料，因此通常從起始資料流程取得的資料將無法讀取。

安全結束程式必須傳回包含下列資訊的 SecurityExitResponse 物件：

- 原因碼 (必須設定)
- 新目的地位址 (選用)
- 新的目的地接聽器埠位址 (選用)
- 訊息 (選用)

原因碼將決定 MQIPT 是否接受或拒絕連線。可以選擇性地設定 newDestination 及 newDestination 埠欄位，以定義新的目標佇列管理程式。如果您未設定這些內容，則將使用配置檔中定義的路徑目的地及 DestinationPort 內容。任何訊息都會附加至連線日誌檔項目。

支援取得內容的方法：

public int getListenerPort ()

擷取路由接聽器埠-如 ListenerPort 內容所定義

public String getDestination()

擷取目的地位址-如「目的地」內容所定義

public int getDestinationPort ()

擷取目的地接聽器埠位址-如 DestinationPort 內容所定義

public String getClientIPAddress ()

擷取提出連線要求之用戶端的 IP 位址

public int getClientPortAddress()

擷取提出連線要求的用戶端所使用的埠位址

public int getTimeout()

會擷取逾時值。MQIPT 將等待安全結束程式驗證要求-如 SecurityExit 逾時內容所定義

public int getConnThreadID()

擷取處理連線要求的連線執行緒 ID，這對於除錯目的很有用

public String getChannelName ()

擷取連線要求中使用的 IBM MQ 通道名稱

public String getQMName()

擷取連線要求中使用的 IBM MQ 佇列管理程式名稱

public boolean getTimedout()

安全結束程式可用來判斷逾時是否已過期

com.ibm.mq.ipt.exit.SecurityExitResponse 類別

這個類別用來將使用者定義安全結束程式的回應傳回給 MQIPT，並用來判斷是否應接受或拒絕連線要求。此類型的物件僅在 validate 方法中建立 (請參閱第 844 頁的『MQIPT 中的 com.ibm.mq.ipt.exit.SecurityExit 類別』)。有一些便利建構子可用來建立這些物件，且每一個內容都有一些方法。如需相關資訊，請參閱範例安全結束程式。

建立預設 SecurityExit 回應物件會拒絕連線要求。

建構子

- **public SecurityExitResponse (String dest , int destPort, int rc , String msg)**

其中：

- dest 是新的目標目的地
- destPort 是新的目的地埠位址
- rc 是原因碼
- msg 是將新增至連線日誌項目的訊息

- **public SecurityExitResponse (String dest , int destPort, int rc)**

- 公用 SecurityExit 回應 (int rc , String msg)

- 公用 SecurityExit 回應 (int rc)

方法

public void setDestination(String dest)

設定連線要求的新目的地位址

public void setDestinationPort (int port) throws IPTException

為連線要求設定新的目的地接聽器埠位址-對無效埠位址擲出 IPTException

public void setMessage(String msg)

將訊息新增至連線日誌記錄

public void setReasonCode (int rc)

設定連線要求的原因碼。

MQIPT 中的安全結束程式回覆碼

在許多不同狀況下呼叫安全結束程式時，MQIPT 可辨識的回覆碼。

在下列狀況中呼叫安全結束程式時，MQIPT 可辨識下列回覆碼：

回覆碼	說明	起始設定	驗證	重新整理
ExitRc。好的	已順利完成要求。	yes	yes	yes

回覆碼	說明	起始設定	驗證	重新整理
ExitRc. INIT_ERROR	起始設定要求失敗，將停用路徑。	yes		
ExitRc. REFRESH_ERROR	重新整理要求失敗。			yes
ExitRc. 未獲授權	驗證程序失敗，連線要求被拒絕。		yes	
ExitRc. DISABLE_SSL	驗證要求成功，目標連線將不會使用 SSL 或 TLS。		yes	

MQIPT 中的埠號控制

使用 MQIPT 時，可以限制建立送出連線時所使用的本端埠號範圍。

設定路徑上的 **OutgoingPort** 內容以指定起始本端埠號，並設定 **MaxConnectionThreads** 以指定要使用的埠數目。例如，如果您將 **OutgoingPort** 設為 1600，並將 **MaxConnectionThreads** 設為 20，則該路徑的本端埠號範圍是 1600-1619。

MQIPT 管理者負責確保路徑之間沒有埠號衝突。

如果未定義 **OutgoingPort**，預設值 0 表示每一個連線都使用系統配置的埠號。

使用 HTTP 時，送出埠的數目是不使用 HTTP 時的兩倍。在前一個範例中，如果路徑使用 HTTP，則數字範圍會是 1600-1639。

如需相關資訊，請參閱 [配置埠號](#)。

多網卡系統

使用多網卡系統時，您可以使用 **LocalAddress** 內容來指定送出連線將連結的 IP 位址。此內容不支援主機名稱。

V 9.1.5 加密 MQIPT 中儲存的密碼

MQIPT 配置可能包含用來存取各種資源的密碼，以及利用指令埠來存取 MQIPT 的密碼。從 IBM MQ 9.1.5 開始，所有這些密碼都應該受到加密保護。

關於這項作業

在 IBM MQ 9.1.5 之前的版本中，只能加密 MQIPT 用來存取金鑰環或加密硬體金鑰儲存庫的密碼。加密密碼儲存在任何 **SSL*KeyRingPw** 內容所參照的檔案中。LDAP 伺服器的其他密碼及 MQIPT 存取密碼以純文字儲存在 `mqipt.conf` 配置檔中。

從 IBM MQ 9.1.5 開始，所有供 MQIPT 使用的儲存密碼都應該透過使用 **mqiptPw** 指令加密密碼來保護。已加密密碼會儲存為 `mqipt.conf` 配置檔中的內容值。MQIPT 能夠在內容值中區分已加密密碼、純文字密碼及檔名。您應該以此方式加密所有儲存供 MQIPT 使用的密碼，因為它是最安全的保護方法。

在 IBM MQ 9.1.5 之前的 MQIPT 中使用加密金鑰儲存庫密碼的方法已淘汰，但仍可使用。若要改善金鑰環密碼的保護，請使用最新的保護方法，重新加密先前已加密的任何金鑰環密碼。

如果 MQIPT 配置中存在純文字或弱保護密碼，則在 MQIPT 啟動或路徑啟動時，會發出警告訊息。

使用此程序，可以使用最新保護方法來加密要儲存以供 MQIPT 使用的密碼。若要在 IBM MQ 9.1.4 或更早版本的 MQIPT 中加密金鑰環密碼，請遵循 [第 848 頁的『在 IBM MQ 9.1.4 或更早版本中加密 MQIPT 中的金鑰環密碼』](#) 中的步驟。

程序

1. 選擇性的: 如果您還沒有密碼加密金鑰，請建立包含密碼加密金鑰的檔案。

MQIPT 使用加密金鑰來加密密碼。您可以在檔案中指定自己的加密金鑰。檔案必須至少包含一個字元，且只能包含一行文字。

使用相同的密碼加密金鑰來加密及解密 MQIPT 實例的所有已儲存密碼。因此，每一個 MQIPT 安裝架構只需要單一密碼加密金鑰檔。

如果您計劃以自動啟動的服務方式執行 MQIPT，則必須建立預設名稱為 `mqiPT_cred.key` 的密碼加密金鑰檔，並將它放置在 MQIPT 起始目錄中。

您不需要指定密碼加密金鑰，但這樣做更安全。如果您未指定自己的加密金鑰，則會使用預設加密金鑰。

註：您必須確保對密碼加密金鑰檔設定適當的檔案許可權，以防止任何未獲授權的使用者讀取加密金鑰。只有執行 `mqiPTPW` 指令的使用者及執行 MQIPT 的使用者才需要讀取密碼加密金鑰的權限。

2. 使用 `mqiPTPW` 指令加密密碼。

`mqiPTPW` (加密儲存密碼)中說明 `mqiPTPW` 指令的語法。

如果您已在步驟 第 847 頁的『1』中建立密碼加密金鑰檔，請使用 `mqiPTPW` 的 `-sf` 參數來指定檔名。例如，可以發出下列指令，使用 `-sf` 參數所指定檔案中的加密金鑰來加密密碼：

```
mqiPTPW -sf /opt/mqiPT/mqiPT_password.key
```

3. 輸入提示時要加密的密碼。

`mqiPTPW` 會輸出已加密的密碼。

4. 將加密密碼複製到 `mqiPT.conf` 配置檔中的適當內容。

例如，下行指定 MQIPT 存取密碼的加密密碼：

```
AccessPW=<mqiPTPW>!QL+2Jvj/tigKK1D7Nz80qw==!AMDBef0UrmPf5i10uqV5MA==
```

5. 啟動 MQIPT。如果您在步驟 第 847 頁的『1』中使用預設名稱以外的名稱建立密碼加密金鑰檔，請在啟動 MQIPT 時指定加密金鑰檔的名稱。

您可以在啟動 MQIPT 時使用 `-sf` 參數指定密碼加密金鑰檔的名稱。例如，發出下列指令，以在 `-sf` 參數指定的檔案中使用加密金鑰來啟動 MQIPT：

```
mqiPT /opt/mqiPT -sf /opt/mqiPT/mqiPT_password.key
```

如需啟動 MQIPT 時用來指定密碼加密金鑰檔名稱之其他方法的相關資訊，請參閱 [指定密碼加密金鑰](#)。

在 IBM MQ 9.1.4 或更早版本中加密 MQIPT 中的金鑰環密碼

在 IBM MQ 9.1.4 及更早版本中，用來存取 MQIPT 所使用金鑰環的已加密密碼會儲存在檔案中。請遵循此作業中的程序來加密金鑰環密碼，以供 MQIPT 在 IBM MQ 9.1.4 或更早版本中使用。

關於這項作業

從 IBM MQ 9.1.5 中的 MQIPT，改用 第 847 頁的『加密 MQIPT 中儲存的密碼』中說明的更安全保護方法。

程序

1. 使用 `mqiPTPW` 指令來加密金鑰環密碼。

輸入下列指令來加密密碼：

```
mqiPTPW password filename
```

其中

PASSWORD

是存取金鑰環所需的明碼密碼

filename

是要建立的密碼檔名稱

`mqiPTPW` (加密儲存密碼)中說明 `mqiPTPW` 指令的語法。

- 將適當的路徑內容設定為包含步驟 第 848 頁的『1』中所建立已加密密碼的檔案名稱。

例如，若要指定包含 MQIPT TLS 伺服器憑證之金鑰環的密碼檔，請將下列行新增至 `mqiPT.conf` 配置檔：

```
SSLServerKeyRingPW=filename
```

MQIPT 的其他安全考量

MQIPT 具有數個其他功能，可協助設計程式建置安全解決方案。

- 如果內部網路中有許多用戶端都嘗試建立送出連線，則它們都可以通過位於防火牆內的 MQIPT。然後，防火牆管理者必須僅授與外部存取權給 MQIPT 電腦。
- 除非 MQIPT 充當 SOCKS Proxy 或使用安全結束程式，否則 MQIPT 只能連接至已在其配置檔中明確配置它的佇列管理程式。
- MQIPT 會驗證它所接收及傳輸的訊息是否有效，並符合 IBM MQ 通訊協定。這有助於防止 MQIPT 用於 IBM MQ 通訊協定之外的安全攻擊。如果 MQIPT 充當 SSL/TLS Proxy，當所有 IBM MQ 資料和通訊協定都已加密時，MQIPT 只能保證起始 SSL/TLS 信號交換。在此狀況下，請使用 [Java security manager](#)。
- MQIPT 容許通道結束程式執行自己的端對端安全通訊協定。
- 您可以設定 `MaxConnectionThreads` 內容來限制送入連線總數。這有助於保護有漏洞的內部佇列管理程式免受阻斷服務攻擊。

配置檔

您必須保護 MQIPT 配置檔 `mqiPT.conf` 不受未獲授權使用者讀取，因為它可能包含機密性資訊，例如 `AccessPW` 密碼，可控制對 MQIPT 的遠端管理存取。遵循 第 847 頁的『加密 MQIPT 中儲存的密碼』中的程序來保護配置檔中指定的所有密碼。此外，請確保 `mqiPT.conf` 受到保護，不會遭到未獲授權的修改。設定 `mqiPT.conf` 的作業系統檔案許可權，以便只有執行 MQIPT 的使用者帳戶才能讀取或更新檔案。

指令埠

MQIPT 指令埠接受 `mqiPTAdmin` Script 或 IPT Administration Client 發出的控制指令。如果已啟用 MQIPT 指令埠，則必須防止對它進行未獲授權的存取。特別是如果啟用 `RemoteShutdown` 內容，則遠端使用者可以關閉 MQIPT。

您應該使用防火牆來限制可連接至 MQIPT 指令埠的電腦集。您也應該使用 `AccessPW` 內容來設定密碼，以控制對指令埠的存取權。

註: MQIPT 指令埠的連線未加密。透過網路傳送的資料 (包括密碼) 可能對網路上的其他使用者可見。

在啟用指令埠之前，您必須先評量容許從遠端管理 MQIPT 的風險。如果已啟用指令埠，請考量使用 `RemoteShutdown` 內容停用遠端關閉。

MQIPT 中的連線日誌

MQIPT 提供連線日誌機能，其中包含所有成功及失敗連線嘗試的清單。

它是使用 `ConnectionLog` 和 `MaxLogFileSize` 內容來控制。如需相關資訊，請參閱 [MQIPT 廣域內容](#)。

每次啟動 MQIPT 時，都會建立新的連線日誌。為了識別，檔名包括現行時間戳記，例如：

```
mqiPTYYYYMMDDHHmmSS.log
```

其中

YYYY 是年份
MM 是月份
DD 是日
HH 是小時

mm 是分鐘

SS 是秒

當連線日誌達到 **MaxLogFileSize** 內容所決定的大小上限時，會建立備份檔 `mqiPT001.log`。最多維護兩個備份檔 (`mqiPT001.log` 和 `mqiPT002.log`)。

連線日誌中的項目代表連線要求的每一部分。MQIPT 所接收的連線要求，以及 MQIPT 對目的地地址所產生的新連線，會顯示為兩個日誌項目，隨後在每一個連線結束時又會有兩個項目。

以下是成功連線要求的連線日誌：

```
Wed May 15 13:13:51 BST 2013 conn accept 127.0.0.1(3842) 127.0.0.1(5000) OK 5000-0
Wed May 15 13:13:51 BST 2013 conn conn 127.0.0.1(3843) localhost(3500) OK 5000-0
Wed May 15 13:13:52 BST 2013 conn close 127.0.0.1(3842) 127.0.0.1(5000) OK 5000-0
Wed May 15 13:13:52 BST 2013 conn close 127.0.0.1(3843) localhost(3500) OK 5000-0
```

以下是失敗連線要求的連線日誌：

```
Wed May 15 14:56:40 BST 2013 conn accept 127.0.0.1(4138) 127.0.0.1(7000) OK 7000-0
Wed May 15 14:56:40 BST 2013 conn close 127.0.0.1(4138) 127.0.0.1(7000) ERROR 7000-0
Unrecognized SSL handshake request '54'
```

使用儲存器來配置 IBM MQ Internet Pass-Thru

您可以在容器中執行 IBM MQ Internet Pass-Thru (MQIPT)。儲存器映像檔使用的基本映像檔必須使用受支援的 Linux 作業系統。

程序

- `mq-container` GitHub 儲存庫中提供範例 MQIPT Docker 映像檔。若要建置並執行儲存器，請遵循 [Docker 上的 IBM MQ Internet Pass-Thru](#) 中的指示。

下一步

您可以使用 `docker ps` 指令來檢視執行中容器。若要檢視在 Docker 儲存器中執行之 MQIPT 的主控制台輸出，請使用 `docker logs ${CONTAINER_ID}` 指令。

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。本書的使用者應依自己的特定環境，查證適用的資料。

本文件所提及之非 IBM 產品資訊，取自產品的供應商，或其發佈的聲明或其他公開管道。IBM 並未測試過這些產品，也無法確認這些非 IBM 產品的執行效能、相容性或任何對產品的其他主張是否完全無誤。有關非 IBM 產品的性能問題應直接洽詢該產品供應商。

有關 IBM 未來方針或目的之所有聲明，僅代表 IBM 的目標與主旨，隨時可能變更或撤銷，不必另行通知。

這份資訊含有日常商業運作所用的資料和報告範例。為了要使它們儘可能完整，範例包括個人、公司、品牌和產品的名稱。這些名稱全屬虛構，如與實際公司的名稱和住址雷同，純屬巧合。

著作權授權：

本資訊含有原始語言之範例應用程式，用以說明各作業平台中之程式設計技術。您可以基於研發、使用、銷售或散布符合作業平台（撰寫範例程式的作業平台）之應用程式介面的應用程式等目的，以任何形式複製、修改及散布這些範例程式，而不必向 IBM 付費。這些範例並未在所有情況下完整測試。因此，IBM 不保證或暗示這些程式的可靠性、有用性或功能。

若貴客戶正在閱讀本項資訊的電子檔，可能不會有照片和彩色說明。

程式設計介面資訊

程式設計介面資訊 (如果有提供的話) 旨在協助您建立與此程式搭配使用的應用軟體。

本書包含預期程式設計介面的相關資訊，可讓客戶撰寫程式以取得 WebSphere MQ 的服務。

不過，本資訊也可能包含診斷、修正和調整資訊。提供診斷、修正和調整資訊，是要協助您進行應用軟體的除錯。

重要：請勿使用此診斷、修改及調整資訊作為程式設計介面，因為它可能會變更。

商標

IBM、IBM 標誌 [ibm.com](http://www.ibm.com) 是 IBM Corporation 在全球許多適用範圍的商標。IBM 商標的最新清單可在 Web 的 "Copyright and trademark information" www.ibm.com/legal/copytrade.shtml 中找到。其他產品和服務名稱，可能是 IBM 或其他公司的商標。

Microsoft 及 Windows 是 Microsoft Corporation 在美國及/或其他國家或地區的商標。

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

Linux 是 Linus Torvalds 在美國及/或其他國家或地區的註冊商標。

本產品包含 Eclipse Project (<http://www.eclipse.org/>) 所開發的軟體。

Java 和所有以 Java 為基礎的商標及標誌是 Oracle 及/或其子公司的商標或註冊商標。



產品編號:

(1P) P/N: