

管理員指南

Sun™ ONE Message Queue

版本 3.0.1

817-5021-10
2002 年 10 月

Copyright © 2002 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303, U.S.A.。版權所有。

Sun Microsystems, Inc. 對本產品所採用的技術擁有相關智慧產權。特別是 (但不僅限於)，這些智慧產權可能包括一項或多項在 <http://www.sun.com/patents> 上列出的美國專利，以及一項或多項美國和其他國家/地區的其他專利或待批專利。

本產品在限制其使用、複製分發以及反編譯的授權下進行銷售。未經 Sun 及其授權者 (如果有) 的事先書面授權，不得以任何方式、任何形式複製本產品的任何部分。

協力廠商軟體 (包括字型技術)，由 Sun 供應商提供版權和授權。

Sun、Sun Microsystems、Sun 標誌、Java、Solaris、iPlanet、JDK、Java Naming and Directory Interface 和 Java 咖啡杯標誌是 Sun Microsystems, Inc. 在美國和其他國家/地區的商標或註冊商標。

所有 SPARC 商標均在授權下使用，它們是 SPARC International, Inc. 在美國和其他國家/地區的商標或註冊商標。帶有 SPARC 商標的產品均基於 Sun Microsystems, Inc. 開發的架構。

UNIX 是在美國和其他國家/地區的註冊商標，由 X/Open Company, Ltd. 獨家授權。

聯邦政府購用：商業軟體 — 政府使用者均應遵守標準軟體授權協議與條款。

目錄

圖錄	11
表錄	13
程序清單	15
前言	17
本指南的使用者	17
本指南的組織架構	18
慣例	19
文字慣例	19
目錄變數慣例	20
其他說明文件資源	21
MQ 說明文件集	21
線上說明	22
JavaDoc	22
範例用戶端應用程式	22
Java 訊息服務 (JMS) 規格	22
第 1 章 簡介	23
何謂 Sun ONE Message Queue ?	23
產品版本	24
平台版	24
企業版	25
企業訊息傳送系統	25
企業訊息傳送系統的要求	25
集中式訊息傳送和點對點訊息傳送	26
訊息傳送系統概念	27
訊息	27
訊息服務架構	27
訊息遞送模型	28

JMS 規格	28
JMS 訊息結構	28
JMS 程式設計模型	29
受管理物件	30
JMS/J2EE 程式設計：訊息驅動 Bean	30
訊息驅動 Bean	31
應用程式伺服器支援	32
JMS 訊息傳送問題	32
JMS 提供者獨立性	32
程式設計網域	33
用戶端識別碼	34
可靠的訊息傳送	34
回應/異動	35
永久性儲存	36
效能平衡	36
訊息選取	36
訊息順序和優先順序	37
 第 2 章 MQ 訊息傳送系統	 39
MQ 訊息伺服器	40
代理程式	40
連線服務	42
訊息路由器	45
持續性管理程式	48
安全性管理程式	51
記錄程式	54
實體目標	58
佇列目標	58
主題目標	59
自動建立的 (對應管理員建立的) 目標	59
暫存目標	60
多重代理程式叢集 (企業版)	60
多重代理程式架構	61
使用開發環境中的叢集	63
叢集配置屬性	63
MQ 用戶端運行時間	64
訊息生產	65
訊息使用	65
MQ 受管理物件	66
連線工廠受管理物件	67
目標受管理物件	68
啟動用戶端時置換屬性值	68

第 3 章 MQ 管理	71
MQ 管理工作	71
開發環境	71
生產環境	71
設定作業	72
維護作業	72
MQ 管理工具	73
管理主控台	73
指令行公用程式摘要	74
指令行語法	75
共用指令行選項	76
 第 4 章 管理主控台指導	 77
準備工作	78
啟動管理主控台	78
取得說明	80
使用代理程式	81
啟動代理程式	82
新增代理程式	83
變更管理員密碼	84
連線至代理程式	85
檢視連線服務	86
將實體目標新增至代理程式	87
使用實體目標	89
取得有關主題目標的資訊	90
使用物件儲存區	91
新增物件儲存區	91
檢查物件儲存區屬性	94
連線至物件儲存區	94
新增連線工廠受管理物件	94
新增目標受管理物件	96
受管理物件屬性	98
更新主控台資訊	99
執行範例應用程式	99
 第 5 章 啟動與配置代理程式	 101
配置檔案	102
合併屬性值	103
屬性命名語法	103
編輯實例配置檔案	104
啟動代理程式	108

使用叢集（企業版）	112
叢集配置屬性	112
連線代理程式	114
方法 1：無叢集配置檔案	114
方法 2：使用叢集配置檔案	114
將代理程式新增至叢集	114
重新啟動叢集中的代理程式	115
從叢集中移除代理程式	115
備份主代理程式的配置變更記錄	116
復原主代理程式的配置變更記錄	116
記錄	117
預設記錄配置	117
日誌訊息格式	117
變更記錄程式配置	118
變更輸出通道	119
變更捲動條件	119
記錄代理程式效能標準資訊	120
 第 6 章 代理程式和應用程式管理	123
指令公用程式	124
指令的語法	124
imqcmd 子指令	124
imqcmd 選項摘要	126
要使用 imqcmd 的必要條件	127
範例	128
控制代理程式的狀態	128
查詢與更新代理程式屬性	130
查詢代理程式	131
更新代理程式	132
管理連線服務	133
列出連線服務	134
查詢與更新服務屬性	135
暫停與繼續服務	136
管理目標	136
建立目標	137
取得有關目標的資訊	138
更新目標	139
清除目標	139
銷毀目標	139
管理持久訂閱	140
管理異動	141

第 7 章 管理受管理物件	145
關於物件儲存區	146
受管理物件	146
物件管理員公用程式 (imqobjmgr)	147
指令語法	147
imqobjmgr 子指令	147
imqobjmgr 指令選項摘要	148
需要的資訊	149
受管理物件屬性	150
連線工廠受管理物件	150
目標受管理物件	152
物件儲存區屬性	153
初始環境和位置資訊	153
安全性資訊 (僅限於 LDAP)	153
使用輸入檔案	154
新增和刪除受管理物件	157
新增連線工廠	157
新增主題或佇列	158
刪除受管理物件	160
取得資訊	160
列出受管理物件	160
有關單一物件的資訊	162
更新受管理物件	162
 第 8 章 安全性管理	 163
認證使用者	164
使用文字檔案使用者儲存庫	164
使用者管理員公用程式 (imqusermgr)	165
群組	167
狀態	167
使用者名稱與密碼的格式	168
移入和管理使用者儲存庫	168
變更預設管理員密碼	169
將 LDAP 伺服器用於使用者儲存庫	170
授權使用者：存取控制屬性檔案	172
存取規則語法	173
許可權運算	174
連線存取控制	175
目標存取控制	176
目標自動建立存取控制	177

加密：使用 SSL 服務（企業版）	177
設定透過 TCP/IP 的 SSL 服務	178
步驟 1. 產生自身簽名的證書	178
步驟 2. 啟用代理程式中基於 SSL 的服務	180
步驟 3. 啟動代理程式	180
步驟 4. 配置並執行基於 SSL 的用戶端	181
設定透過 HTTP 的 SSL 服務	182
使用密碼檔案	182
 附錄 A 設定外掛持續性	185
介紹	185
插入 JDBC 可存取的資料儲存區	186
與 JDBC 相關的代理程式配置屬性	187
資料庫管理員公用程式 (imqdbmgr)	189
指令語法	189
imqdbmgr 子指令	190
imqdbmgr 指令選項摘要	190
 附錄 B HTTP/HTTPS 支援（企業版）	191
HTTP/HTTPS 支援架構	191
實施 HTTP 支援	193
步驟 1. 在 Web Server 上佈署 HTTP 通道 Servlet	193
作為 JAR 檔案佈署	193
作為 Web 歸檔檔案佈署	194
步驟 2. 配置 httpjms 連線服務	194
步驟 3. 配置 HTTP 連線	195
設定連線工廠屬性	195
使用單一 Servlet 存取多重代理程式	196
使用 HTTP 代理	196
範例：佈署 HTTP 通道 Servlet	197
作為 JAR 檔案佈署	197
作為 WAR 檔案佈署	199
實施 HTTPS 支援	200
步驟 1. 為 HTTPS 通道 Servlet 產生自身簽名的證書	200
步驟 2. 在 Web Server 上佈署 HTTPS 通道 Servlet	201
作為 JAR 檔案佈署	201
作為 Web 歸檔檔案佈署	202
步驟 3. 配置 httpsjms 連線服務	202

步驟 4. 配置 HTTPS 連線	203
配置 JSSE	204
匯入超級使用者證書	204
設定連線工廠屬性	205
使用單一 Servlet 存取多重代理程式	205
使用 HTTP 代理	206
範例：佈署 HTTPS 通道 Servlet	206
作為 JAR 檔案佈署	206
作為 WAR 檔案佈署	208
附錄 C 將代理程式用作 Windows 服務	211
將代理程式作為 Windows 服務執行	211
服務管理員公用程式 (imqsvcadmin)	212
指令的語法	212
imqsvcadmin 子指令	212
imqsvcadmin 選項摘要	213
移除代理程式服務	214
重新配置代理程式服務	214
使用替代 Java 運行時間	214
查詢代理程式服務	214
疑難排解	215
附錄 D MQ 資料的位置	217
附錄 E MQ 介面的穩定性	219
詞彙表	223
索引	227

圖錄

圖 1-1	集中式訊息傳送和點對點訊息傳送	26
圖 1-2	訊息服務架構	27
圖 1-3	JMS 程式設計物件	29
圖 1-4	與 MDB 進行訊息傳送	31
圖 2-1	MQ 系統架構	39
圖 2-2	代理程式元件	41
圖 2-3	連線服務支援	43
圖 2-4	持續性管理程式支援	49
圖 2-5	安全性管理程式支援	52
圖 2-6	記錄機制	55
圖 2-7	多重代理程式 (叢集) 架構	61
圖 2-8	訊息傳送作業	65
圖 2-9	訊息遞送至 MQ 用戶端運行時間	66
圖 3-1	本機與遠端管理公用程式	74
圖 5-1	代理程式配置檔案	103
圖 B-1	HTTP/HTTPS 支援架構	192

表錄

表格 1	本書內容	18
表格 2	文件慣例	19
表格 3	MQ 目錄變數	20
表格 4	MQ 說明文件集	21
表格 1-1	JMS 程式設計物件	33
表格 2-1	主代理程式元件與功能	41
表格 2-2	代理程式支援的連線服務	42
表格 2-3	連線服務屬性	44
表格 2-4	訊息路由器屬性	47
表格 2-5	持續性屬性	50
表格 2-6	安全性屬性	53
表格 2-7	記錄種類	55
表格 2-8	記錄程式屬性	56
表格 2-9	自動建立配置屬性	60
表格 2-10	叢集配置屬性	63
表格 2-11	目標屬性	68
表格 3-1	共用 MQ 指令行選項	76
表格 5-1	代理程式實例配置屬性	104
表格 5-2	imqbrokerd 選項	109
表格 5-3	叢集配置屬性	112
表格 5-4	imqbrokerd 記錄程式選項及相應的屬性	118
表格 5-5	為連線服務收集的標準資訊	121
表格 5-6	為每個代理程式收集的標準資訊	121
表格 6-1	imqcmd 子指令	124
表格 6-2	imqcmd 選項	126
表格 6-3	用於控制代理程式的 imqcmd 子指令	129

表格 6-4	用於取得資訊及更新代理程式的 imqcmd 子指令	130
表格 6-5	代理程式屬性	132
表格 6-6	用於管理連線服務的 imqcmd 子指令	133
表格 6-7	代理程式支援的連線服務	134
表格 6-8	連線服務屬性	136
表格 6-9	用於管理目標的 imqcmd 子指令	137
表格 6-10	目標屬性	137
表格 6-11	用於管理持久訂閱的 imqcmd 子指令	140
表格 6-12	用於管理異動的 imqcmd 子指令	141
表格 7-1	imqobjmgr 子指令	147
表格 7-2	imqobjmgr 選項	148
表格 7-3	連線工廠屬性	151
表格 7-4	目標屬性	152
表格 7-5	物件儲存區的安全性屬性	154
表格 8-1	使用者儲存庫中的初始項目	164
表格 8-2	imqusermgr 子指令	166
表格 8-3	imqusermgr 選項	166
表格 8-4	無效的使用者名稱與密碼字元	168
表格 8-5	與 LDAP 相關的屬性	170
表格 8-6	存取規則的語法元素	173
表格 8-7	目標存取控制規則的元素	176
表格 8-8	鍵值儲存區屬性	179
表格 8-9	密碼檔案中的密碼	183
表格 A-1	與 JDBC 相關的屬性	187
表格 A-2	imqdbmgr 子指令	190
表格 A-3	imqdbmgr 選項	190
表格 B-1	httpjms 連線服務屬性	195
表格 B-2	用於佈署 HTTP 通道 Servlet JAR 檔案的 Servlet 引數	198
表格 B-3	httpsjms 連線服務屬性	203
表格 B-4	用於佈署 HTTPS 通道 Servlet JAR 檔案的 Servlet 引數	207
表格 C-1	imqsvcadmin 子指令	212
表格 C-2	imqsvcadmin 選項	213
表格 D-1	MQ 3.0 資料的位置	217
表格 E-1	MQ 3.0.1 介面的穩定性	219
表格 E-2	介面穩定性分類機制	221

程序清單

啓動管理主控台	78
顯示管理主控台說明資訊	80
啓動代理程式	82
將代理程式新增至管理主控台	83
變更管理員密碼	84
連線至代理程式	85
檢視可用的連線服務	86
將佇列目標新增至代理程式	88
檢視實體目標的屬性	89
從目標中清除訊息	90
刪除目標	90
新增檔案系統物件儲存區	91
顯示物件儲存區的屬性	94
連線至物件儲存區	94
將連線工廠新增至物件儲存區	95
將目標新增至物件儲存區	97
檢視或更新目標物件的屬性	98
執行 SimpleJNDIClient 應用程式	99
將代理程式連線至叢集	114
使用叢集配置檔案時將代理程式新增至叢集	115
在發生故障的情況下復原主代理程式	116
變更代理程式的記錄程式配置	118
編輯配置檔案以使用 LDAP 伺服器	170
設定 ssljms 連線服務	178
重新產生鍵對	179
插入 JDBC 可存取的資料儲存區	186

實施 HTTP 支援	193
啓用 httpjms 連線服務	194
新增通道 Servlet	197
爲通道 Servlet 配置虛擬路徑 (Servlet URL)	198
在啓動 Web Server 時載入通道 Servlet	198
停用伺服器存取記錄	199
將 HTTP 通道 Servlet 作爲 WAR 檔案佈署	199
實施 HTTPS 支援	200
啓用 httpsjms 連線服務	202
配置 JSSE	204
新增通道 Servlet	206
爲通道 Servlet 配置虛擬路徑 (Servlet URL)	207
在啓動 Web Server 時載入通道 Servlet	208
停用伺服器存取記錄	208
修改 HTTPS 通道 Servlet WAR 檔案	208
將 HTTPS 通道 Servlet 作爲 WAR 檔案佈署	209
查閱記錄的服務錯誤事件	215

前言

「Sun™ ONE Message Queue (MQ) 3.0.1 管理員指南」這本書提供執行 MQ 訊息傳送系統管理工作所需的背景和資訊。

本前言包含以下各節：

- 本指南的使用者
- 本指南的組織架構
- 慣例
- 其他說明文件資源

本指南的使用者

本指南可供需要執行 MQ 管理工作的管理員和應用程式開發人員使用。

MQ 管理員負責設定和管理 MQ 訊息傳送系統，尤其是位於本系統中心的 MQ 訊息伺服器。本書假設您不瞭解訊息傳送系統知識。

本指南還可供應用程式開發人員使用，以使他們更瞭解如何最適化他們的應用程式，從而充分利用 MQ 訊息傳送系統的功能和靈活性。

本指南的組織架構

本指南需要您完整閱讀。下表簡要描述了每章的內容：

表格 1 本書內容

章	描述
第 1 章 「簡介」	描述 MQ 訊息傳送系統以及術語的高級概念簡介。
第 2 章 「MQ 訊息傳送系統」	描述 MQ 訊息傳送系統，重點介紹可共同提供訊息傳送服務的 MQ 代理程式和 MQ 用戶端運行時間。
第 3 章 「MQ 管理」	描述 MQ 管理工作和工具，並介紹用於管理的命令行公用程式及其共用功能。
第 4 章 「管理主控台指導」	提供實用指導，以使您熟悉管理主控台 (即 MQ 訊息伺服器的圖形介面)。
第 5 章 「啓動與配置代理程式」	介紹如何啓動和配置 MQ 代理程式及代理程式叢集。
第 6 章 「代理程式和應用程式管理」	介紹如何執行與管理 MQ 代理程式相關的 (獨立於應用程式) 工作，以及用於管理訊息傳送應用程式的工作。
第 7 章 「管理受管理物件」	介紹如何執行與建立和管理 MQ 受管理物件相關的工作。
第 8 章 「安全性管理」	介紹如何執行與應用程式相關的安全性工作，如管理認證、授權以及加密。
附錄 A 「設定外掛持續性」	介紹如何設定 MQ 以使用 JDBC 相容資料庫來執行持續性功能。
附錄 B 「HTTP/HTTPS 支援 (企業版)」	介紹如何在訊息傳送用戶端與 MQ 訊息伺服器之間設定 HTTP 連線服務。
附錄 C 「將代理程式用作 Windows 服務」	介紹如何使用 MQ 服務管理公用程式 (imqsvcadmin) 來安裝、查詢以及移除代理程式 (作為 Windows 服務執行)。
附錄 D 「MQ 資料的位置」	描述不同種類 MQ 資料的位置。
附錄 E 「MQ 介面的穩定性」	描述不同 MQ 介面的穩定性。
「詞彙表」	定義 MQ 說明文件中使用的專有名詞。

慣例

本節提供了有關本文件中所使用慣例的資訊。

文字慣例

表格 2 文件慣例

格式	描述
<i>斜體字型</i>	斜體文字表示萬用字元。請在看到斜體文字時將其取代為適當的子句或值。斜體文字也用於指定需要強調的文件標題，或用於引入的單字或片語。
固定間距字型	固定間距字型文字表示範例程式碼、在指令行上所輸入的指令、目錄名稱、檔案名稱或路徑名稱、錯誤訊息文字、類別名稱、方法名稱 (包括簽名中的所有元素)、套裝軟體名稱、保留字以及 URL。
[]	指示指令行語法敘述中選用值的方括號。
全部大寫	所有大寫文字表示檔案系統類型 (GIF、TXT、HTML 等等)、環境變數 (IMQ_HOME) 或縮寫字 (MQ、JSP)。
鍵 + 鍵	透過加號連接在一處並同時按下的按鍵：Ctrl+A 表示同時按下這兩個鍵。
鍵 - 鍵	透過連字號連接在一處的連續按下的按鍵：Esc-S 表示按下 Esc 鍵，然後釋放它，接著按下 S 鍵。

目錄變數慣例

MQ 使用三個目錄變數，其設定方式根據平台的不同而不同。[表格 3](#) 描述了這些變數，並概述了它們在 Solaris、Windows 以及 Linux 平台上的使用方式。

表格 3 MQ 目錄變數

變數	描述
IMQ_HOME	它通常在 MQ 說明文件中使用，指的是 MQ 根安裝目錄： - 在 Solaris 上，沒有 MQ 根安裝目錄。因此，未在 MQ 說明文件中使用 IMQ_HOME 來指代 Solaris 上的檔案位置。 - 在 Solaris 上，Application Server Evaluation Edition 的 MQ 根安裝目錄為：<i>Application Server</i> 根安裝目錄/imq。 - 在 Windows 上，MQ 根安裝目錄由 MQ 安裝程式設定 (依預設，此目錄為 C:\Program Files\Sun Microsystems\Message Queue 3.0.1)。 - 在 Windows 上，Sun ONE Application Server 的 MQ 根安裝目錄為：<i>Application Server</i> 根安裝目錄/imq。 - 在 Linux 上，MQ 根安裝目錄預設為：/opt/imq。
IMQ_VARHOME	這是儲存 MQ 暫存檔或動態建立的配置檔與資料檔的 /var 目錄。可以將它設定為指向任何目錄的環境變數。 - 在 Solaris 上，IMQ_VARHOME 預設為 /var/imq 目錄。 - 在 Solaris 上，針對 Sun ONE Application Server Evaluation Edition，IMQ_VARHOME 預設為 IMQ_HOME/var。 - 在 Windows 上，IMQ_VARHOME 預設為 IMQ_HOME/var。 - 在 Windows 上，針對 Sun ONE Application Server，IMQ_VARHOME 預設為 IMQ_HOME/var。 - 在 Linux 上，IMQ_VARHOME 預設為 IMQ_HOME/var。

表格 3 MQ 目錄變數 (續)

變數	描述
IMQ_JAVAHOME	這是指向 MQ 可執行檔所需的 Java 運行時間 (JRE 1.4) 位置的环境變數： - 在 Solaris 上，IMQ_JAVAHOME 預設為 /usr/j2se/jre 目錄，但是，使用者可以選擇性地將值設定為 JRE 1.4 常駐的任意位置。 - 在 Windows 上，IMQ_JAVAHOME 預設為 IMQ_HOME/jre，但是，使用者可以選擇性地將值設定為 JRE 1.4 常駐的任意位置。 - 在 Linux 上，IMQ_JAVAHOME 預設為 /usr/java/j2sdk1.0/jre 目錄，但是，使用者可以選擇性地將值設定為 JRE 1.4 常駐的任意位置。

在本指南中，IMQ_HOME、IMQ_VARHOME 以及 IMQ_JAVAHOME 的顯示不附帶平台特定的環境變數表示法或語法 (例如 UNIX 上的 \$IMQ_HOME)。所有路徑名稱均使用 UNIX 檔案分隔符號表示法 (/)。

其他說明文件資源

除了本指南以外，MQ 還提供了其他說明文件資源。

MQ 說明文件集

構成 MQ 說明文件集的說明文件依照一般的使用順序列示在**表格 4**中。

表格 4 MQ 說明文件集

文件	使用者	描述
MQ 安裝指南	開發人員與管理員	介紹如何在 Solaris、Linux 以及 Windows 平台上安裝 MQ 軟體。
版次注意事項	開發人員與管理員	包括對新功能、限制、已知錯誤以及技術說明的描述。
MQ 開發人員指南	開發人員	提供快速啟動指導以及與 JMS MQ 實施相關的程式設計資訊。

表格 4 MQ 說明文件集 (續)

文件	使用者	描述
MQ 管理員指南	管理員，也建議開發人員使用	提供使用 MQ 管理工具執行管理工作所需的背景與資訊。

線上說明

MQ 3.0.1 包括用於執行 MQ 訊息服務管理工作的指令行公用程式。若要存取這些公用程式的線上說明，請參閱第 76 頁的「共用指令行選項」。

MQ 3.0.1 還包括圖形使用者介面 (GUI) 管理工具，即管理主控台 (imqadmin)。上下文關聯的線上說明包括在管理主控台中。

JavaDoc

JavaDoc 格式的 JMS 和 MQ API 說明文件可在以下位置提供：

IMQ_HOME/javadoc/index.html
(在 Solaris 上為 /usr/share/javadoc/imq/index.html)

本說明文件可在任何 HTML 瀏覽器 (如 Netscape 或 Internet Explorer) 中檢視。它包括標準的 JMS API 說明文件，以及用於 MQ 受管理物件的 MQ 特定的 API (請參閱「MQ 開發人員指南」的第 3 章)，對訊息傳送應用程式的開發人員很有幫助。

範例用戶端應用程式

以下位置中包含一些提供範例用戶端應用程式碼的範例應用程式：

IMQ_HOME/demo (在 Solaris 上為 /usr/demo/imq)

請參閱此目錄及其子目錄中的 README 檔案。

Java 訊息服務 (JMS) 規格

JMS 規格可在以下位置找到：

<http://java.sun.com/products/jms/docs.html>

規格包括範例用戶端程式碼。

本章提供 Sun™ ONE Message Queue (MQ) 的簡介，適用於管理員和程式設計人員閱讀。

何謂 Sun ONE Message Queue ？

MQ 產品為應用程式間的通訊和可靠訊息遞送提供了基於標準的解決方案。MQ 是實施 Java Message Service (JMS) 開放式標準的企業訊息傳送系統，即 JMS 提供者。

JMS 規格描述了一組程式設計介面 (請參閱第 29 頁的「**JMS 程式設計模型**」)，這些介面為 Java 應用程式提供了在分散式環境中建立、傳送、接收和讀取訊息的常用方法。

使用 Sun ONE Message Queue 軟體，在不同平台和作業系統上執行的程序可連線至一個共用 MQ 訊息服務 (請參閱第 27 頁的「**訊息服務架構**」)，以傳送和接收資訊。應用程式開發人員便可以將精力集中在應用程式的業務邏輯上，而不是集中在應用程式應如何在網路中進行通訊這樣的低層細節上。

MQ 的功能已超出 JMS 規格的最低要求。這些功能如下：

集中式管理 提供指令行和 GUI 工具，用於管理 MQ 訊息服務和訊息傳送中應用程式特定的方面，如目標和安全性。

可延伸式訊息服務 可讓您在以串聯方式 (多重代理程式叢集) 工作的大量 MQ 訊息服務元件 (代理程式) 之間平衡負載，以服務於不斷增多的 JMS 用戶端 (元件或應用程式)。

可調整的效能 在遞送可靠性要求較低的情況下，則可讓您提高 MQ 訊息服務的效能。

多重傳輸 支援 JMS 用戶端使用多種不同的傳輸方式進行相互通訊的功能，包括 TCP 和 HTTP 以及使用安全 (SSL) 連線。

JNDI 支援 支援將 Java Naming and Directory Interface (JNDI) 的基於檔案的實施和 LDAP 實施用作物件儲存區和使用者儲存庫。

SOAP 訊息傳送支援 支援經由 JMS 訊息傳送來建立和遞送 SOAP 訊息 (符合簡單物件存取協定 [SOAP] 規格的訊息)。SOAP 允許在分散式環境中的各點之間交換結構化 XML 資料。請參閱「MQ 開發人員指南」，以取得更多資訊。

請參閱 MQ 3.0.1 版次注意事項，以取得有關 JMS 相容性問題的說明文件。

產品版本

Sun ONE Message Queue 產品有兩個版本：平台版與企業版 — 每個版本對應不同的授權，如下所述。(若要將 MQ 從一個版本升級至另一個版本，請參閱「MQ 安裝指南」中的說明。)

平台版

該版本可以從 Sun 網站免費下載，並且還隨附有最新的 Sun ONE Application Server 平台。平台版未對每項 MQ 訊息服務所支援的 JMS 用戶端連線做出數量限制。該版本附帶有兩種授權，如下所述：

- **基本授權** 該授權提供基本的 JMS 支援 (它是一個完整的 JMS 提供者)，但不包括下列企業功能，如負載平衡 (多重代理程式訊息服務)、HTTP/HTTPS 連線、安全連線服務、可延伸式連線功能以及多重佇列遞送策略。該授權的有效期沒有限定，因此可以用於要求較低的生產環境。
- **具有 90 天試用期的企業授權** 該授權包括基本授權中未提供的所有企業功能 (如支援多重代理程式訊息服務、HTTP/HTTPS 連線、安全連線服務、可延伸式連線功能以及多重佇列遞送策略)。然而，軟體強制規定該授權只有 90 天的有效期，所以，它適合於評估企業版產品提供的企業功能 (請參閱第 25 頁的「企業版」)。

註 可以藉由以下方法啓用為期 90 天的試用授權：透過 `-license` 指令行選項啓動 MQ 訊息服務 (MQ 代理程式實例) (請參閱第 109 頁的表格 5-2)，然後將「try」作為要使用的授權進行傳送：

```
imqbrokerd -license try
```

每次啓動代理程式實例時都必須使用此選項，否則它會預設回基本的平台版授權。

企業版

該版本用於在生產環境中佈署與執行訊息傳送應用程式。它包括對多重代理程式訊息服務、HTTP/HTTPS 連線、安全連線服務、可延伸式連線功能以及多重佇列遞送策略的支援。您還可以使用企業版來進行開發和除錯，並可以載入測試訊息傳送應用程式和元件。企業版具有無限制的授權有效期，並且該授權對多重代理程式訊息服務中的代理程式數量也沒有限制，但是指定了受支援的 CPU 數量。

企業訊息傳送系統

企業訊息傳送系統可讓獨立的分散式元件或應用程式透過訊息互動。這些元件（無論是在同一系統、同一網路上執行，還是透過網際網路鬆散地連接在一起）均使用訊息傳送來傳輸資料以及協調各自的功能。

企業訊息傳送系統的要求

企業應用程式系統一般包含大量的分散式元件，這些元件在全天候的關鍵任務作業中交換數以萬計的訊息。若要支援此類系統，企業訊息傳送系統通常必須滿足以下要求：

可靠的遞送 從一個元件傳送至另一個元件的訊息不能由於網路或系統故障而遺失。這意味著系統必須能夠保證成功地遞送訊息。

非同步遞送 為使大量元件能夠同時交換訊息，並支援高密度流量，訊息傳送不能取決於使用者是否可以立即接收訊息。如果使用者忙碌或離線，系統必須允許先將訊息傳送出去，然後在使用者準備就緒時接收訊息。這就是所謂的非同步訊息遞送，通常稱為儲存與轉寄訊息傳送。

安全性 訊息傳送系統必須支援基本的安全性功能：使用者認證、訊息和資源的授權存取以及線上加密。

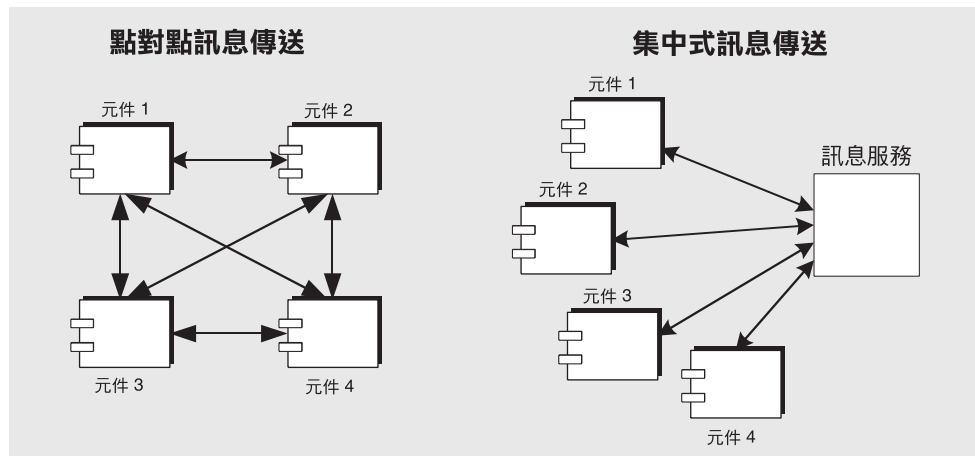
延伸性 訊息傳送系統必須能夠在不降低效能或訊息流量的情況下，適應不斷增加的負載，即不斷增多的使用者和訊息數量。隨著商業和應用程式的延伸，這將成為一項很重要的要求。

可管理性 訊息傳送系統必須提供監視和管理訊息遞送以及優化系統資源的工具。這些工具有助於衡量和維護系統的可靠性、安全性和效能。

集中式訊息傳送和點對點訊息傳送

傳統的點對點訊息傳送系統很難滿足企業訊息傳送系統的要求，如圖 1-1 中所示。

圖 1-1 集中式訊息傳送和點對點訊息傳送



在此系統中，每個訊息傳送元件與其他所有元件均保持連線。這些連線可以實現快速、安全且可靠的遞送，但是支援可靠性和安全性的程式碼必須駐留在每個元件中。隨著將元件不斷地加入系統，連線數目將成指數級增加。這使得非同步訊息遞送和可延伸性很難實現。集中式管理也會出現問題。

企業訊息傳送的首選方法是集中式訊息傳送系統，在圖 1-1 中也有說明。在此方法中，每個訊息傳送元件只需與一個中央訊息服務保持連線。訊息服務提供元件之間的訊息遞送和路由，並負責可靠遞送及安全性。元件透過定義完善的程式設計介面與訊息服務進行互動。隨著將元件不斷地加入系統，連線的數目只會線性增長，這樣就可以透過調整訊息服務來輕鬆地調整系統。此外，中央訊息服務還為系統提供集中式管理。

訊息傳送系統概念

以下幾個基本概念構成企業訊息傳送系統的基礎，包括：訊息、訊息服務架構和訊息遞送模型。

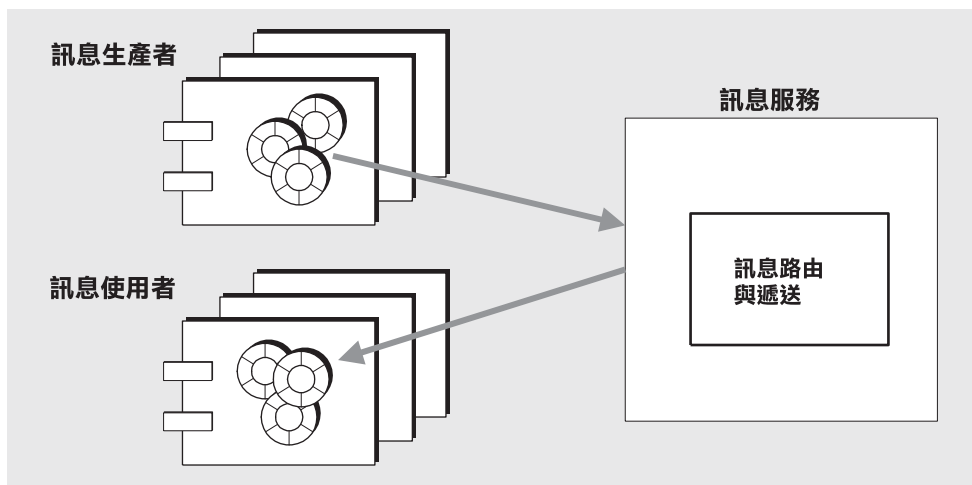
訊息

訊息包含某種格式的資料 (訊息內文) 以及描述訊息特徵或屬性的元資料 (訊息標頭)，如目標、有效期或由訊息傳送系統決定的其他特徵。

訊息服務架構

訊息傳送系統的基本架構如第 27 頁的圖 1-2 所示。它包含訊息生產者和訊息使用者，二者透過共用訊息服務來交換訊息。任何數目的訊息生產者和使用者均可駐留在同一個訊息傳送元件（或應用程式）中。訊息生產者將訊息傳送至訊息服務。然後，訊息服務使用訊息路由與遞送元件，將訊息遞送至一個或多個已註冊接收此訊息的訊息使用者。訊息路由與遞送元件負責保證將該訊息遞送至所有相應的使用者。

圖 1-2 訊息服務架構



訊息遞送模型

生產者與使用者之間存在多種關係：一對一、一對多和多對多。例如，您遞送訊息的方式可能是：

- 從一個生產者至一個使用者
- 從一個生產者至多個使用者
- 從多個生產者至一個使用者
- 從多個生產者至多個使用者

這些關係通常被簡化為兩個訊息遞送模型：點對點與出版/訂閱訊息傳送。點對點遞送模型主要處理由特定生產者創建並由特定使用者接收的訊息。出版/訂閱遞送模型主要處理由任意數目的生產者創建並由任意數目的使用者接收的訊息。這兩個訊息遞送模型可以同時使用。

過去訊息傳送系統支援這兩個訊息遞送模型的各種組合。Java Message Service (JMS) API 旨在為 Java 訊息傳送建立一種共用的程式設計方法。它同時支援點對點訊息遞送模型和出版/訂閱訊息遞送模型 (請參閱第 33 頁的「程式設計網域」)。

JMS 規格

JMS 指定訊息結構、程式設計模型和一組控制訊息傳送作業的規則和語義。因為 MQ 提供了 JMS 的實施方法，所以 JMS 概念對於理解 MQ 訊息傳送系統的工作原理很重要。本簡介說明了理解本書其餘各章所需的概念和術語。

JMS 訊息結構

依據 JMS 規格，訊息由以下三部分組成：標頭、屬性和內文。

標頭 標頭指定訊息的 JMS 特徵：目標、永久與否、存在時間以及優先順序。這些特徵控制訊息傳送系統遞送訊息的方式。

屬性 屬性 (可被認為標頭的延伸) 是選擇性的 — 應用程式可以使用屬性提供的值根據各種選取條件來過濾訊息。屬性為選擇性的。

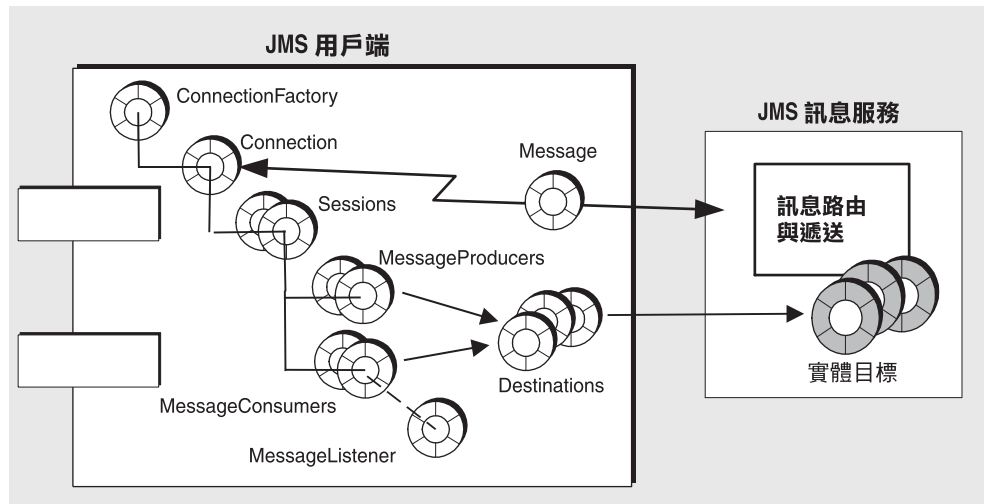
訊息內文 訊息內文包含要交換的實際資料。JMS 支援六種內文類型。

JMS 程式設計模型

在 JMS 程式設計模型中，JMS 用戶端（元件或應用程式）透過 JMS 訊息服務交換訊息。訊息生產者將訊息傳送至訊息服務，訊息接收者則從訊息服務接收這些訊息。這些訊息傳送作業是使用一組實施 JMS 應用程式設計介面 (API) 的物件（由 JMS 提供者提供）來執行的。

本節介紹實施 JMS API 的物件，以及用於為訊息遞送設定 JMS 用戶端的物件（如需更多資訊，請參閱「MQ 開發人員指南」）。第 29 頁的圖 1-3 顯示用於設計訊息遞送程式的 JMS 物件。

圖 1-3 JMS 程式設計物件



在 JMS 程式設計模型中，JMS 用戶端使用 `ConnectionFactory` 物件建立一個連線，將訊息傳送至 JMS 訊息服務以及從 JMS 訊息服務接收訊息均透過此連線進行。`Connection` 是 JMS 用戶端與訊息服務的作用中連線。通訊資源的分配以及用戶端的認證均在建立連線時進行。相對而言，這是一個十分重要的物件，大多數用戶端均使用單一連線來進行所有的訊息傳送。

連線用於建立階段作業。`Session` 是用於生產和使用訊息的單執行緒環境。它用於建立傳送和接收訊息的訊息生產者和使用者，並為所遞送的訊息定義遞送順序。階段作業透過大量的回應選項或異動（可由分散式異動管理程式管理）支援可靠的遞送。

JMS 用戶端使用 `MessageProducer` 將訊息傳送至指定實體目標（在 API 中表示為目標物件）。訊息生產者可以指定預設遞送模式（永久性與非永久性訊息）、優先順序和存在時間值，以控制由生產者傳送至實體目標的所有訊息。

同樣，JMS 用戶端使用 `MessageConsumer` 從指定實體目標 (在 API 中表示為目標物件) 接收訊息。訊息使用者可以使用訊息選擇器，該選擇器可讓訊息服務僅將符合選取條件的訊息遞送至訊息使用者。

訊息使用者可以支援同步或非同步的訊息使用 (請參閱「MQ 開發人員指南」)。透過向使用者註冊 `MessageListener` 可實現非同步使用。當階段作業執行緒呼叫 `MessageListener` 物件的 `onMessage()` 方法時，用戶端即可使用訊息。

受管理物件

第 29 頁的「JMS 程式設計模型」中描述的兩個物件取決於 JMS 提供者實施 JMS 訊息服務的方法。連線工廠物件取決於提供者用於遞送訊息的基本協定和機制，目標物件取決於提供者使用的特定命名慣例和實體目標功能。

通常，這些提供者特定的特徵使得 JMS 用戶端程式碼取決於特定的 JMS 實施。但是，要使 JMS 用戶端程式碼獨立於提供者，JMS 規格要求將提供者特定的實施和配置資訊封裝在名為受管理物件的物件中。這些物件可以使用標準的非提供者特定的方式來存取。

受管理物件由管理員建立和配置，並存儲在名稱服務中，由 JMS 用戶端透過標準 Java Naming and Directory Service (JNDI) 查找代碼來存取。以這種方式使用受管理物件可使 JMS 用戶端程式碼獨立於提供者。

JMS 提供兩種一般類型的受管理物件：連線工廠和目標。雖然二者都用於封裝提供者特定的資訊，但是它們在 JMS 用戶端中的用途有很大差異。連線工廠用於建立至訊息伺服器的連線，而目標物件用於識別 JMS 訊息服務使用的實體目標。

如需有關受管理物件的更多資訊，請參閱第 66 頁的「MQ 受管理物件」。

JMS/J2EE 程式設計：訊息驅動 Bean

除第 29 頁的「JMS 程式設計模型」中介紹的一般 JMS 用戶端程式設計模型之外，還有專用於 Java 2 企業版 (J2EE) 應用程式環境的 JMS。此專用的 JMS 用戶端稱為訊息驅動 Bean，是 EJB 2.0 規格 (<http://java.sun.com/products/ejb/docs.html>) 中指定的企業 JavaBeans (EJB) 元件系列之一。

因為其他 EJB 元件 (階段作業 Bean 和實體 Bean) 僅可被同步呼叫，所以就產生了對訊息驅動 Bean 的需求。這些 EJB 元件只能透過標準 EJB 介面來存取，因此它們沒有非同步接收訊息的機制。

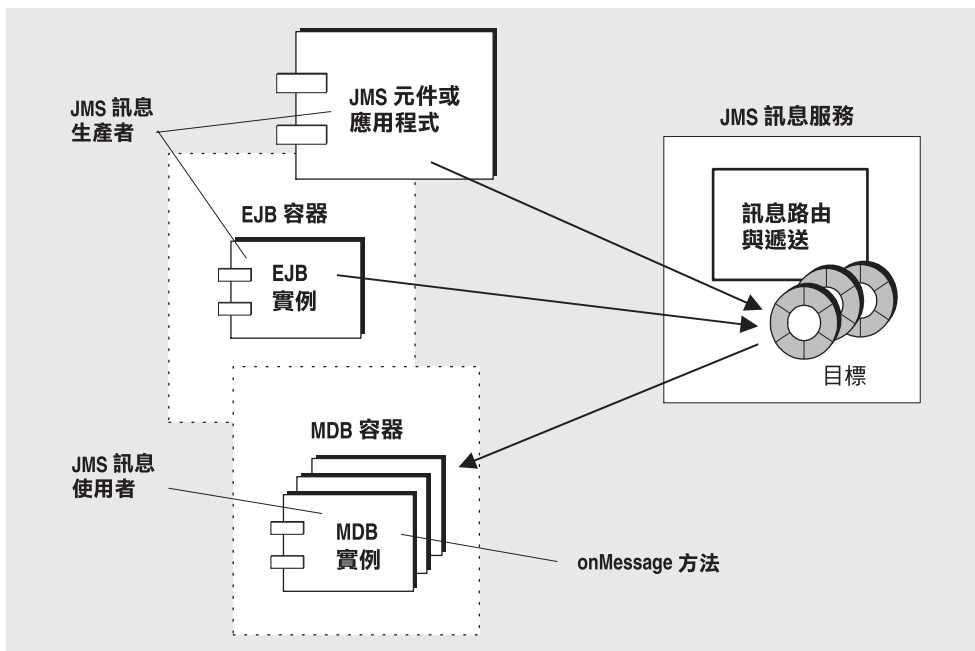
但是，非同步訊息傳送是許多企業應用程式中必不可少的要求。這些應用程式大都需要伺服器端元件在不佔用伺服器資源的情況下，能夠互相通訊和回應。因此就出現了對 EJB 元件的需求，該元件不需要緊密耦合到訊息生產者即可接收和使用訊息。對於伺服器端元件必須回應應用程式事件的任何應用程式，均必須具備此功能。在企業應用程式中，此功能還必須能夠依據不斷增加的負載來進行延伸。

訊息驅動 Bean

訊息驅動 Bean (MDB) 是由專用 EJB 容器 (為所支援的元件提供分散式服務的軟體環境) 支援的專用 EJB 元件。

訊息驅動 Bean MDB 是實施 JMS `MessageListener` 介面的 JMS 訊息使用者。在 MDB 容器接收訊息時，會呼叫 `onMessage` 方法 (由 MDB 開發人員編寫)。`onMessage()` 方法使用該訊息的方式與標準 `MessageListener` 物件的 `onMessage()` 方法使用訊息的方式一樣。您不能以呼叫其他 EJB 元件的方法來遠端呼叫 MDB，因此不存在與之關聯的主介面或遠端介面。MDB 可以使用來自單一目標的訊息。獨立式 JMS 應用程式、JMS 元件、EJB 元件或 Web 元件均可以生產訊息，如第 31 頁的圖 1-4 中所示。

圖 1-4 與 MDB 進行訊息傳送



MDB 容器 MDB 由專用 EJB 容器支援，負責建立和設定 MDB 實例，以用於非同步訊息使用。這包括設定與訊息服務 (包括認證) 的連線，建立與給定目標關聯的階段作業儲存區，以及管理在階段作業儲存區和關聯的 MDB 實例之間分配收到的訊息。由於容器控制 MDB 實例的生命週期，因此它透過管理 MDB 實例儲存區來容納進來的訊息負載。

與 MDB 關聯的是一個佈署描述元，此描述元為設定訊息使用時所用的受管理物件指定了 JNDI 查找名稱：連線工廠和目標。佈署描述元可能還包括佈署工具用於配置容器的其他資訊。每個容器僅支援單一 MDB 的實例。

應用程式伺服器支援

在 J2EE 架構中 (請參閱

<http://java.sun.com/j2ee/download.html#platformspec> 上的 J2EE 平台規格)，EJB 容器由應用程式伺服器託管。應用程式伺服器提供各種容器所需的資源：異動管理程式、永久性管理程式、名稱服務以及 JMS 提供者 (用於訊息傳送和 MDB)。

在 Sun ONE Application Server 中，訊息傳送資源由 Sun ONE Message Queue 提供。這意味著已將 MQ 訊息傳送系統 (請參閱第 2 章「MQ 訊息傳送系統」) 整合到 Sun ONE Application Server 中，提供將 JMS 訊息傳送至應用程式伺服器環境中執行的 MDB 和其他 JMS 訊息傳送元件所需的支援。

JMS 訊息傳送問題

本節描述許多影響 MQ 訊息服務管理的 JMS 程式設計問題。主要討論 MQ 管理員需要的概念和術語。

JMS 提供者獨立性

JMS 指定受管理物件的用法 (請參閱第 30 頁的「受管理物件」)，以支援可以移植到其他 JMS 提供者的用戶端應用程式的開發。受管理物件可讓 JMS 用戶端使用邏輯名稱查找和參考提供者特定的物件。這樣，用戶端程式碼無需瞭解提供者使用的特定命名語法、尋址語法或配置屬性，從而使程式碼獨立於提供者。

受管理物件為 MQ 管理員建立和配置的 MQ 系統物件。這些物件放在 JNDI 目錄服務中，JMS 用戶端可以使用 JNDI 查找存取它們。

MQ 受管理物件也可以由用戶端創設，而不是在 JNDI 目錄服務中查找。這樣做的缺點是，要求應用程式開發人員使用提供者特定的 API，並降低了 MQ 管理員成功控制和管理 MQ 訊息伺服器的能力。

如需有關受管理物件的更多資訊，請參閱第 66 頁的「MQ 受管理物件」。

程式設計網域

JMS 支援兩種完全不同的訊息遞送模型：點對點與出版/訂閱。

點對點 (佇列目標) 將訊息從生產者遞送至一個使用者。在此遞送模型中，目標為佇列。首先將訊息遞送至佇列目標，然後根據佇列的遞送策略 (請參閱第 58 頁的「佇列目標」) 將訊息從該佇列遞送至已向佇列註冊的一個使用者，每次遞送一封訊息。可以向佇列目標傳送訊息的生產者沒有限制，但每封訊息只能遞送至一個使用者，並由一個使用者成功接收。如果沒有向佇列目標註冊的使用者，佇列將保留其接收到的訊息，並在使用者向該佇列註冊時將訊息遞送給他們。

出版/訂閱 (主題目標) 將訊息從生產者遞送至任意數目的使用者。在此遞送模型中，目標為主題。訊息被首先遞送至主題目標，然後遞送至已訂閱該主題的所有作用中使用者。可以向主題目標傳送訊息的生產者沒有限制，並且每封訊息可以被遞送至任意數目的訂閱使用者。主題目標還支援持久訂閱。持久訂閱表示使用者已向主題目標進行註冊，但在遞送訊息時此使用者可以處於非作用狀態。然後，當使用者再處於作用中時，將接收此訊息。如果沒有使用者向主題目標進行註冊，則主題將不保留其收到的訊息，除非有非作用中使用者註冊了持久訂閱。

這兩種訊息遞送模型使用表示不同程式設計網域的不同 API 物件 (語義稍有差異) 進行處理，如表格 1-1 中所示。

表格 1-1 JMS 程式設計物件

基本類型 (統一網域)	點對點網域	出版/訂閱網域
Destination (Queue 或 Topic) ¹	Queue	Topic
ConnectionFactory	QueueConnectionFactory	TopicConnectionFactory
Connection	QueueConnection	TopicConnection
Session	QueueSession	TopicSession
MessageProducer	QueueSender	TopicPublisher
MessageConsumer	QueueReceiver	TopicSubscriber

¹ 取決於程式設計的方法，您可以指定特定的目標類型。

您可以使用**表格 1-1** 第一欄中所示的統一網域物件設計點對點和出版/訂閱訊息傳送的程式。這是通常採用的方法。但是，爲了符合早期的 JMS 1.02b 規格，可以使用點對點網域物件設計點對點訊息傳送的程式，使用出版/訂閱網域物件設計出版/訂閱訊息傳送的程式。

用戶端識別碼

JMS 提供者必須支援**用戶端識別碼**概念，用戶端識別碼將 JMS 用戶端至訊息服務的連線與代表用戶端的訊息服務所維護的狀態資訊相關聯。根據定義，用戶端識別碼是唯一的，並且每次僅套用至一個使用者。用戶端識別碼與持久訂閱名稱 (請參閱**第 33 頁的「出版/訂閱 (主題目標)」**) 結合使用，可以確保每個持久訂閱僅對應一個使用者。

JMS 規格允許用戶端透過 API 方法呼叫來設定用戶端識別碼，但是建議您使用連線工廠受管理物件 (請參閱**第 30 頁的「受管理物件」**) 從管理級別來設定它。但是如果硬連線至連線工廠，則每個使用者均需要個別的連線工廠 以擁有唯一身份。

使用可在 `ConnectionFactory` 物件中配置的特殊變數替換語法，MQ 提供了一種方法，可讓用戶端識別碼同時特定於 `ConnectionFactory` 和使用者。使用此方法時，單一 `ConnectionFactory` 物件可由建立持久訂閱的多個使用者使用，而不必擔心命名衝突或缺乏安全性。這樣就保護了使用者的持久訂閱，不會因爲其他使用者設定了錯誤的用戶端識別碼而導致意外刪除或不可用。

如需有關如何使用此 MQ 功能的詳細資訊，請參閱「MQ 開發人員指南」中有關連線工廠屬性的說明。

無論如何，若要建立持久訂閱，用戶端識別碼必須由用戶端使用 JMS API 透過程式設定，或在用戶端使用的 `ConnectionFactory` 物件中進行管理級別的配置。

可靠的訊息傳送

JMS 定義了兩種遞送模式：

永久性訊息 保證這些訊息僅被遞送一次並成功使用一次。對於這些訊息，可靠性是首要因素。

非永久性訊息 保證這些訊息最多被遞送一次。對於這些訊息，可靠性並非主要因素。

對於**永久性訊息**，確保可靠性有兩個方面。一個是要確保至訊息服務和來自訊息服務的訊息遞送成功。另一個是要確保訊息服務在將永久性訊息遞送至使用者之前不會遺失訊息。

回應/異動

可靠的訊息傳送取決於確保至目標和來自目標的永久性訊息的成功遞送。可使用 MQ 階段作業支援的兩個一般機制中的任何一個來實現此目的，這兩個機制是：回應和異動。異動（可以是本地異動或分散式異動）由分散式異動管理程式控制。

回應

可以將階段作業配置為使用回應來保證可靠遞送。

對於生產者，這意味著在生產者的 `send()` 方法返回之前，訊息服務將向其目標回應永久性訊息的遞送。對於使用者，這意味著在訊息服務從該目標刪除永久性訊息之前，用戶端將回應來自該目標的永久性訊息的遞送與使用。

本地異動

也可以將階段作業配置為異動的，這樣就可以將一個或多個訊息的生產和/或使用群組為原子單元，即異動。JMS API 提供了啟動、確定或回轉異動的方法。

在異動中生產或使用訊息時，代理程式會追蹤各種傳送和接收過程，並在用戶端發出確定異動的呼叫時完成這些作業。如果異動中特定的傳送或接收作業失敗，則會出現異常。用戶端程式碼可以透過忽略異常、重試作業或回轉整個異動來處理異常。在異動確定後，將完成所有成功的作業。在異動回轉後，將取消所有成功的作業。

本地異動的範圍始終是單一階段作業。即，可以將單一階段作業的環境中執行的一個或多個生產者或使用者作業群組為一個本地異動。

由於異動僅能是單一階段作業，因此不能有同時包含訊息生產和訊息使用的端對端異動。（換言之，至目標的訊息遞送和後續的至用戶端的遞送不能放在同一個異動中。）

分散式異動

MQ 還支援分散式異動。即，訊息的生產和使用可作為較大的分散式異動的一部分，其中包括涉及其他資源管理程式（如資料庫系統）的作業。在分散式異動中，分散式異動管理程式使用 Java Transaction API (JTA)、XA 資源 API 規格中定義的兩階段確定協定，追蹤和管理多個資源管理程式（如訊息服務和資料庫管理程式）執行的作業。在 Java 中，JTA 規格描述了資源管理程式和分散式異動管理程式之間的互動。

支援分散式異動表示，訊息傳送用戶端可以透過 JTA 定義的 XA 資源介面參與分散式異動。此介面定義了實施兩階段確定的許多方法。當用戶端進行 API 呼叫時，MQ 代理程式僅與分散式異動管理程式（由 Java Transaction Service [JTS] 提供）協作，來追蹤分散式異動中的各種傳送和接收作業以及異動狀態，並完成訊息傳送作業。

處理本地異動時，用戶端可以透過忽略異常、重試作業或回轉整個分散式異動來處理異常。

MQ 透過 XA 連線工廠 實施對分散式異動的支援，這可讓您建立 XA 連線，而 XA 連線又可讓您建立 XA 階段作業 (請參閱第 29 頁的「**JMS 程式設計模型**」)。此外，對分散式異動的支援還要求協力廠商 JTS 或與 J2EE 相容的 Application Server (提供 JTS)。

永久性儲存

可靠性的另一個重要方面是，確保將永久性訊息遞送至其目標後，訊息服務在將其遞送至使用者之前不會遺失這些訊息。這意味著將永久性訊息遞送至其目標時，訊息服務必須將其放在永久性資料存儲區中 (請參閱第 48 頁的「**持續性管理程式**」)。如果訊息服務由於某種原因失敗，它可以回復該訊息並將其遞送至相應的使用者。雖然這增加了訊息遞送的耗用時間，但卻提昇了可靠性。

訊息服務還必須儲存持久訂閱。這是因為要確保主題目標下的遞送，因此僅回復永久性訊息是不夠的。訊息服務還必須回復有關主題持久訂閱的資訊，否則它將無法將訊息遞送至在訊息到達時處於非作用中狀態、隨後又處於作用中狀態的訂閱者。

需要可靠訊息遞送的訊息傳送應用程式必須將訊息指定為永久性訊息，並使用佇列目標或主題目標的持久訂閱。

效能平衡

訊息遞送的可靠性越高，需要的耗用時間和頻寬就越多。可靠性與效能之間的平衡是設計時要考慮的重點。您可以透過選擇生產和使用非永久性訊息來最大化**效能**。另一方面，您也可以透過生產和使用永久性訊息並使用異動階段作業來最大化**可靠性**。在這兩種極端之間有許多選擇，這取決於應用程式的需要，包括使用 MQ 特定的連接和回應屬性 (請參閱「**MQ 開發人員指南**」)。

訊息選取

JMS 提供了一種機制，透過它，訊息服務可以根據訊息選擇器中的條件執行訊息過濾和路由。生產型用戶端可在訊息中放入應用程式特定的屬性，使用型用戶端可以使用基於這些屬性的選取條件來表明它對訊息的興趣。這樣可以簡化用戶端的工作，並消除了向不需要這些訊息的用戶端遞送訊息的耗用時間。但是，它也使得處理選取條件的訊息服務增加了一些額外耗用時間。JMS 規格中對訊息選擇器的語法和語義進行了概述。

訊息順序和優先順序

通常，可以確保將單一階段作業向目標傳送的所有訊息按其傳送順序遞送至使用者。但是，如果為它們分配不同的優先順序，訊息傳送系統將嘗試先遞送優先級較高的訊息。

此外，用戶端應用程式使用訊息的順序與訊息的生產順序僅具有一個粗略的關係。這是因為向目標遞送訊息以及從這些目標遞送訊息，可以取決於多個影響時間的問題，如訊息的傳送順序、傳送訊息的階段作業 (連線)、是否為永久性訊息、訊息的存在時間、訊息的優先順序、佇列目標的訊息遞送策略 (請參閱第 58 頁的「佇列目標」) 以及訊息服務的可用性。

對於使用多個互連代理程式 (請參閱第 60 頁的「多重代理程式叢集 (企業版)」) 的 MQ 訊息伺服器，用戶端使用訊息的排序更為複雜，因為從不同代理程式的目標遞送訊息的順序是不確定的。因此，由一個代理程式遞送的訊息可能會在另一個代理程式所遞送的訊息之前，即使後者可能會先收到訊息。

無論如何，對於給定使用者，具有較高優先順序的訊息總是在具有較低優先順序的訊息之前。

MQ 訊息傳送系統

本章描述 Sun™ ONE Message Queue (MQ) 訊息傳送系統，並介紹此系統的主要部分如何共同工作以提供可靠的訊息遞送，這些詳細說明的主要部分如圖 2-1 中所示。

圖 2-1 MQ 系統架構

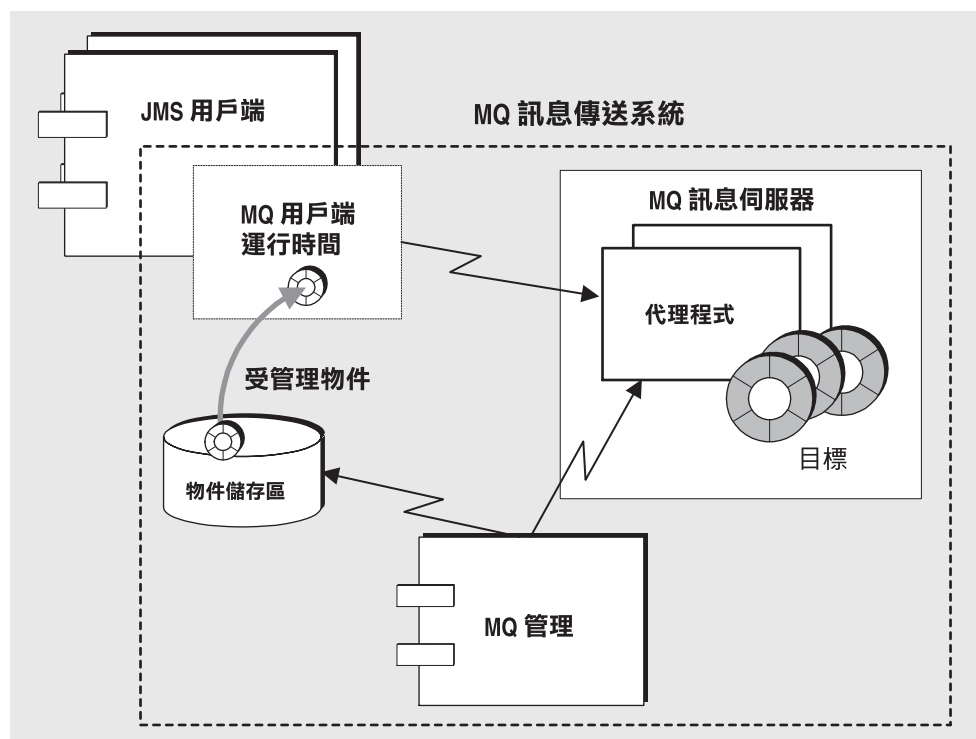


圖 2-1 中所示的 MQ 訊息傳送系統的主要部分如下：

- MQ 訊息伺服器
- MQ 用戶端運行時間
- MQ 受管理物件
- MQ 管理

前三個部分將在以下各節中詳細說明。最後一個部分將在第 3 章「MQ 管理」中介紹。

MQ 訊息伺服器

本節描述第 39 頁的圖 2-1 中所示的 MQ 訊息伺服器的不同部分。這些部分包括：

代理程式 MQ 代理程式為 MQ 訊息傳送系統提供遞送服務。訊息遞送依賴一些支援元件，這些元件可處理連線服務、訊息路由與遞送、持續性、安全性以及記錄 (請參閱第 40 頁的「代理程式」以獲得更多資訊)。訊息伺服器可使用一個或多個代理程式實例 (請參閱第 60 頁的「多重代理程式叢集 (企業版)」)。

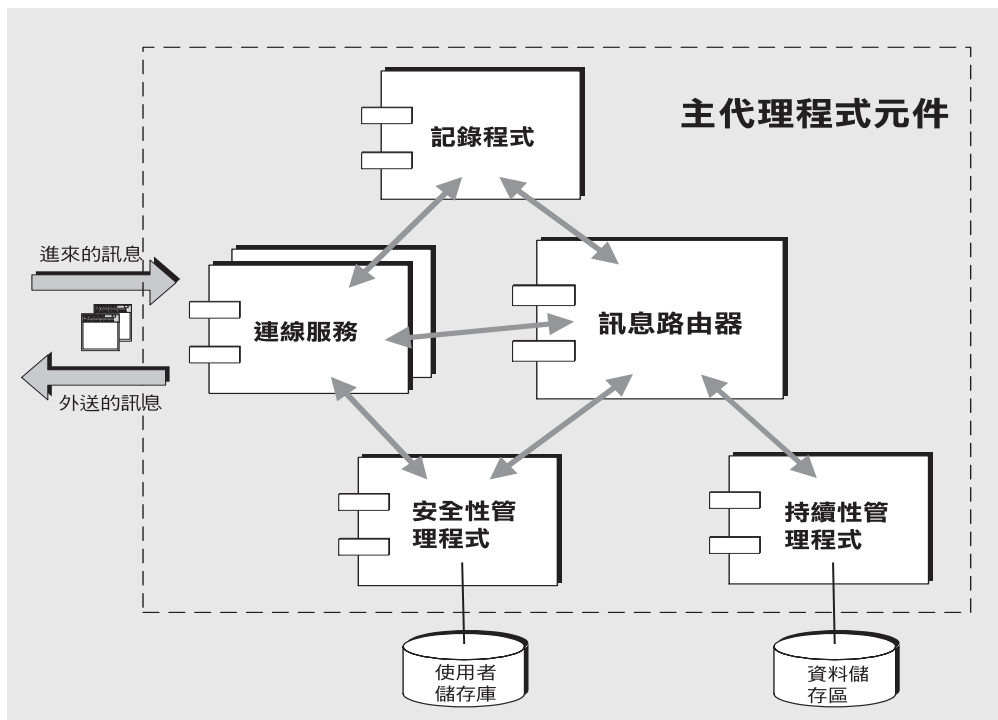
實體目標 訊息遞送過程分為兩個階段 — 從生產用戶端遞送至由代理程式維護的實體目標，然後從目標遞送至一個或多個使用用戶端。實體目標表示在代理程式的實體記憶體和/或永久性儲存體中的位置 (請參閱第 58 頁的「實體目標」以獲得更多資訊)。

代理程式

MQ 訊息傳送系統中的訊息遞送 — 從生產用戶端至目標，然後從目標至一個或多個使用用戶端 — 由代理程式 (或以串聯方式工作的代理程式實例叢集) 執行。若要執行訊息遞送，代理程式必須設定與用戶端通訊的通道、執行認證與授權、適當路由訊息、確保可靠遞送，並為監視系統效能提供資料。

若要執行這套複雜的功能，代理程式將使用一些不同的元件，並且每個元件在遞送過程中都有特定的作用。根據載入條件、應用程式的複雜性等，您可以配置這些內部元件，以最適化代理程式的效能。圖 2-2 中說明主代理程式元件，表格 2-1 中則簡單描述這些元件。

圖 2-2 代理程式元件



表格 2-1 主代理程式元件與功能

元件	描述/功能
連線服務	管理代理程式與用戶端之間的實體連線，為進來和外送的訊息提供傳輸。
訊息路由器	管理訊息的路由與遞送：這些訊息包括 JMS 訊息以及 MQ 訊息傳送系統為支援 JMS 訊息遞送而使用的控制訊息。
持續性管理程式	管理向永久性儲存體的資料寫入，以便系統故障不會導致遞送 JMS 訊息的故障。
安全性管理程式	為請求連線至代理程式的使用者提供認證服務，為已認證的使用者提供授權服務 (存取控制)。
記錄程式	將監視與診斷資訊寫入日誌檔或主控台，以便管理員可監視與管理代理程式。

以下各節更全面地探究不同的代理程式元件所執行的功能，以及可配置為影響元件運作方式的屬性。

連線服務

MQ 代理程式支援與 JMS 用戶端和 MQ 管理用戶端的通訊 (請參閱第 73 頁的「MQ 管理工具」)。每種服務均由其服務類型和協定類型指定。

服務類型 指定此服務提供 JMS 訊息遞送 (NORMAL) 或提供 MQ 管理 (ADMIN) 服務。

協定類型 指定支援此服務的基本傳輸協定層。

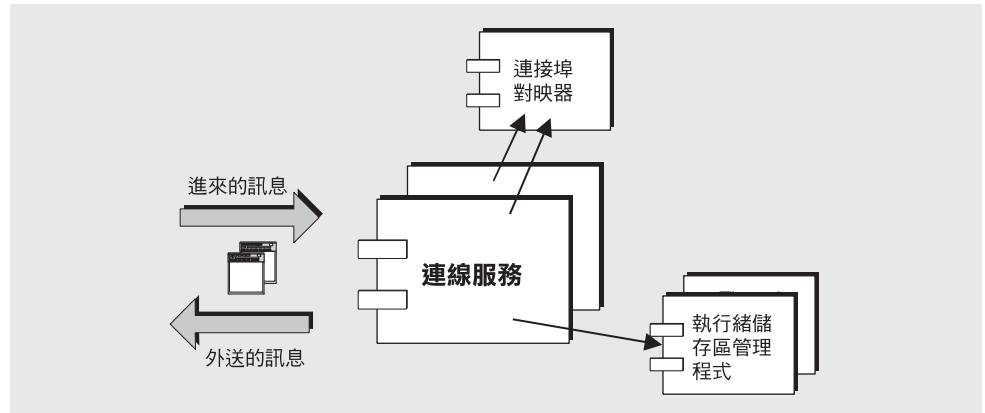
目前，MQ 代理程式可提供的連線服務如表格 2-2 中所示：

表格 2-2 代理程式支援的連線服務

服務名稱	服務類型	協定類型
jms	NORMAL (JMS 訊息遞送)	tcp
ssljms (企業版)	NORMAL (JMS 訊息遞送)	tls (基於 SSL 的安全性)
httpjms (企業版)	NORMAL (JMS 訊息遞送)	http
httpsjms (企業版)	NORMAL (JMS 訊息遞送)	https (基於 SSL 的安全性)
admin	ADMIN	tcp
ssladmin (企業版)	ADMIN	tls (基於 SSL 的安全性)

您可以將代理程式配置為執行這些連線服務的任何一種或全部。每種服務均有執行緒儲存區管理程式，並使用共用連接埠對映器服務註冊自己，如圖 2-3 中所示。

圖 2-3 連線服務支援



每種連線服務可用於特定的連接埠，此連接埠由代理程式的主機名稱和連接埠號指定。連接埠可靜態或動態配置。MQ 提供的**連接埠對映器**可將已動態配置的連接埠對映至不同的連線服務。連接埠對映器本身常駐在標準的連接埠號 7676 上。當用戶端設定與代理程式的連線時，它先聯絡連接埠對映器，以請求其所需的連線服務的連接埠號。

當配置 `jms`、`ssljms`、`admin` 和 `ssladmin` 連線服務時，您還可以為它們指定靜態連接埠號，但不建議這樣做。使用附錄 B「**HTTP/HTTPS 支援 (企業版)**」的**第 195 頁的表格 B-1**和**第 203 頁的表格 B-3**中描述的屬性，可分別配置 `httpjms` 和 `httpsjms` 服務。

每種連線服務均為多重執行緒，支援多重連線。這些連線所需的執行緒，在**執行緒儲存區管理程式**元件所管理的執行緒儲存區中維護。您可以配置此執行緒儲存區管理程式，以設定此執行緒儲存區中所維護的執行緒的最小數量和最大數量。由於連線需要執行緒，因此可將執行緒新增至執行緒儲存區。當超過最小數量時，系統將關閉執行緒（因為這些執行緒將成為自由執行緒，直至達到最小數量的臨界值），從而節省記憶體資源。您應將此數量變得足夠大，以便不必繼續建立新的執行緒。載入大量連線後，執行緒的數量會增加，直至達到執行緒儲存區的最大數量，然後連線必須等待，直至某執行緒變為可用。

執行緒儲存區中的執行緒可專屬於單一連線（**專屬模型**），或依需要指定給多重連線（**共用模型**）。

專屬模型 在專屬模型中，與代理程式的每條連線均需要兩個執行緒：一個專屬於處理進來的訊息，另一個專屬於處理外送的訊息。這可將連線數量限制為執行緒儲存區中最大執行緒數量的一半，但它可提供高效能。

共用模型 (企業版) 在共用執行緒模型中，僅當傳送或接收訊息時才將連線指定給執行緒。這種連線共用執行緒的模型會增加連線服務 (和代理程式) 可支援的連線數量，但也會增加效能耗用時間。執行緒儲存區管理程式使用一套分散執行緒，可監視連線狀態並依需要將連線指定給執行緒。您可以透過限制由每個此類分散執行緒所監視的連線數量來提昇效能。

每種連線服務可支援特定認證和授權 (存取控制) 功能 (請參閱第 51 頁的「安全性管理程式」)。

表格 2-3 中顯示與連線服務相關的可配置屬性。(如需有關配置這些屬性的說明，請參閱第 5 章「啟動與配置代理程式」。)

表格 2-3 連線服務屬性

屬性名稱	描述
<code>imq.service.activelist</code>	要在啟動代理程式時變為作用中的連線服務的清單，這些連線服務依名稱列出並以逗號分隔。支援的服務為： <code>jms</code> 、 <code>ssljms</code> 、 <code>httpjms</code> 、 <code>httpsjms</code> 、 <code>admin</code> 和 <code>ssladmin</code> 。預設： <code>jms</code> 和 <code>admin</code>
<code>imq.service_name.min_threads</code>	指定執行緒數量，一旦達到指定數量，執行緒便會在執行緒儲存區中維護，供已命名的連線服務使用。預設：取決於連線服務 (請參閱第 104 頁的表格 5-1)。
<code>imq.service_name.max_threads</code>	指定執行緒數量，一旦超過指定數量，系統便不會將新的執行緒新增至執行緒儲存區，供已命名的連線服務使用。此數量必須大於零，並且其值必須大於 <code>min_threads</code> 的值。預設：取決於連線服務 (請參閱第 104 頁的表格 5-1)。
<code>imq.service_name.threadpool_model</code>	指定執行緒是專屬於連線 (<code>dedicated</code>) 或依需要由連線共用 (<code>shared</code>)，以用於已命名的連線服務。共用模型 (執行緒儲存區管理) 會增加代理程式所支援的連線數量，但僅實施用於 <code>jms</code> 和 <code>admin</code> 連線服務。預設：取決於連線服務 (請參閱第 104 頁的表格 5-1)。
<code>imq.shared.connectionMonitor_limit</code>	僅用於共用執行緒儲存區模型，指定可由分散執行緒監視的最大連線數量。(系統配置了足夠的分散執行緒以監視所有連線。) 此值越小，系統將作用中的連線指定給執行緒的速度就越快。值 0 表示沒有限制。預設：取決於作業系統 (請參閱第 104 頁的表格 5-1)。
<code>imq.portmapper.port</code>	代理程式的主要連接埠，即連接埠對映器常駐的連接埠。如果您要在主機上執行多個代理程式實例，則必須為每個實例指定唯一的連接埠對映器連接埠。預設：7676

表格 2-3 連線服務屬性 (續)

屬性名稱	描述
<code>imq.service_name.protocol_type¹.port</code>	僅用於 <code>jms</code> 、 <code>ssljms</code> 、 <code>admin</code> 和 <code>ssladmin</code> 服務，為已命名的連線服務指定連接埠號。預設：0 (連接埠由連接埠對映器動態配置) 若要配置 <code>httpjms</code> 和 <code>httpsjms</code> 連線服務，請參閱附錄 B「HTTP/HTTPS 支援 (企業版)」
<code>imq.service_name.protocol_type¹.hostname</code>	僅用於 <code>jms</code> 、 <code>ssljms</code> 、 <code>admin</code> 和 <code>ssladmin</code> 服務，如果有多台主機可用 (例如，如果一台電腦中有多个網路介面卡)，指定已命名的連線服務所連結的主機 (主機名稱或 IP 位址)。預設：null (任何主機)

¹ 協定類型在表格 2-2 中指定。

訊息路由器

一旦使用支援的連線服務在用戶端和代理程式之間建立了連線，則可進行訊息路由與遞送。

基本的遞送機制

概括地講，代理程式所處理的訊息分為兩類：由生產者用戶端傳送且目標為使用者用戶端的有效載入 JMS 訊息，以及為了支援 JMS 訊息遞送而向用戶端傳送及從其中發出的一些控制訊息。

如果進來的訊息為 JMS 訊息，則代理程式將基於此訊息的目標類型 (佇列或主題)，將其路由至使用者用戶端：

- 如果目標為主題，則 JMS 訊息將立即路由至此主題的所有作用中訂閱者。如果訂閱者為非作用中的持久訂閱者，則訊息路由器將保留此訊息，直至此訂閱者成為作用中訂閱者，然後將此訊息遞送至此訂閱者。
- 如果目標為佇列，則 JMS 訊息將放置在相應的佇列中，然後在此訊息到達佇列前面時被遞送至適當的使用者。訊息到達佇列前面的順序，取決於它們到達的順序及優先順序。

一旦訊息路由器已將訊息遞送至所有預定的使用者，它便會將此訊息從記憶體中清除，如果此訊息為永久性的 (請參閱第 34 頁的「可靠的訊息傳送」)，它會將其從代理程式的永久性資料儲存區中移除。

可靠遞送：回應與異動

當為可靠遞送 (請參閱第 34 頁的「可靠的訊息傳送」) 新增需求時，剛才描述的遞送機制會變得更加複雜。可靠遞送包含兩個方面：確保向代理程式傳送及從其中發出的訊息遞送成功，並確保實際遞送訊息之前，代理程式不會遺失訊息或遞送資訊。

為確保訊息成功遞送至代理程式及從其中成功發出，MQ 使用了一些名為回應的控制訊息。

例如，當生產者將 JMS 訊息 (收費載入訊息而不是控制訊息) 傳送至目標時，代理程式會傳送回控制訊息 (即代理程式回應)，表明它已接收到此 JMS 訊息。(實際上，僅當生產者指定 JMS 訊息為永久性時，MQ 才執行此作業。) 生產用戶端使用代理程式回應，以確保遞送至目標 (請參閱第 65 頁的「訊息生產」)。

同樣，當代理程式將 JMS 訊息遞送至使用者時，此使用用戶端會傳送回它已接收到並已處理此訊息的回應。建立階段作業物件時，用戶端會指定如何自動或頻繁地傳送這些回應，但原則是，如果訊息路由器沒有從它已向其遞送訊息的每個訊息使用者 (例如，從主題的多重訂閱者的每個訂閱者) 接收到回應，則它不會從記憶體中刪除 JMS 訊息。

如果訂閱者為主題的持久訂閱者，則訊息路由器會將每個 JMS 訊息保留在目標中，並當每個持久訂閱者成為作用中訂閱者時遞送此訊息。訊息路由器接收用戶端回應時會記錄這些回應，並僅在已接收所有回應後才刪除 JMS 訊息 (除非 JMS 訊息在此之前已過期)。

而且，透過將代理程式回應傳送回用戶端，訊息路由器確認用戶端回應的接收。使用用戶端使用代理程式回應，以確保代理程式不會多次遞送同一 JMS 訊息 (請參閱第 65 頁的「訊息使用」)。如果由於某種原因而使代理程式無法接收到用戶端回應，則可能發生上述情況。

如果代理程式沒有接收到用戶端回應並將 JMS 訊息再次遞送，則此訊息會標記有重新遞送旗標。如果用戶端連線在代理程式接收到用戶端回應之前關閉，並隨後開啓新的連線，則此代理程式通常會重新遞送 JMS 訊息。例如，如果佇列的訊息使用者在回應訊息之前離線，而且另一個使用者隨後註冊此佇列，則代理程式將未回應的訊息重新遞送至此新的使用者。

以上描述的用戶端與代理程式回應過程，還適用於群組至異動中的 JMS 訊息遞送。在這些情況下，用戶端與代理程式回應可對異動作業，亦可對傳送或接收的個別 JMS 訊息作業。確定異動時，代理程式回應會被自動傳送。

代理程式追蹤異動，並可在發生故障時對異動進行確定或回轉。本異動管理還支援本機異動，此為較大型分散式異動 (請參閱第 35 頁的「分散式異動」) 的一部分。代理程式追蹤這些異動的狀態，直至它們被確定。代理程式啟動時，它會檢查所有未確定的異動，並依預設，會回轉所有異動 (那些處於 PREPARED 狀態的異動除外)。

可靠遞送：持續性

可靠遞送的另一個方面是，確保實際遞送訊息之前，代理程式不會遺失訊息或遞送資訊。一般來說，訊息會保留在記憶體中，直至它們已被遞送或過期。但是，如果代理程式出現故障，則這些訊息會遺失。

生產者用戶端可指定訊息為永久性的，在此情況下，訊息路由器會將此訊息傳送至將此訊息儲存在資料庫或檔案系統中的**持續性管理程式**（請參閱第 48 頁的「**持續性管理程式**」），以便在代理程式出現故障時回復此訊息。

管理系統資源

代理程式的效能取決於可用的系統資源，以及如何有效利用記憶體之類的資源。例如，訊息路由器具有的記憶體管理機制可監視系統上的記憶體。當記憶體資源不足時，系統會啟動用於回收記憶體以及減慢進來訊息流量的機制。

記憶體管理機制取決於記憶體資源的狀態：`green`（大量記憶體可用）、`yellow`（代理程式記憶體正在減少）、`orange`（代理程式記憶體不足）以及 `red`（代理程式無記憶體）。在記憶體資源的狀態從 `green` 變為 `yellow` 和 `orange`，然後變為 `red` 的過程中，代理程式會採取越來越重要的動作，以回收記憶體並減低訊息生產者的速度，最終使訊息停止流入代理程式。

您可以使用一些屬性配置代理程式的記憶體管理功能，這些屬性可設定記憶體中訊息的總數量和總容量限制，並可在記憶體資源變更至新狀態時調整利用臨界值。

表格 2-4 中詳細介紹了這些屬性。（如需有關設定這些屬性的說明，請參閱第 5 章「**啟動與配置代理程式**」。）

表格 2-4 訊息路由器屬性

屬性名稱	描述
<code>imq.message.expiration.interval</code>	指定回收已過期訊息發生的頻率（以秒為單位）。預設：60
<code>imq.system.max_count</code>	指定記憶體和磁碟（歸因於交換）中的最大訊息數量。其他訊息將被拒絕。值 0 表示沒有限制。預設：0
<code>imq.system.max_size</code>	指定記憶體和磁碟（歸因於交換）中的最大訊息容量（以位元組、千位元組或百萬位元組為單位）。其他訊息將被拒絕。值 0 表示沒有限制。預設：0
<code>imq.message.max_size</code>	指定訊息內文的最大允許容量（以位元組、千位元組或百萬位元組為單位）。大於此容量的任何訊息均將被拒絕。值 0 表示沒有限制。預設：70m（百萬位元組）
<code>imq.resource_state.threshold</code>	指定啟動每種記憶體狀態時的記憶體利用百分比。資源狀態的值可以為 <code>green</code> 、 <code>yellow</code> 、 <code>orange</code> 和 <code>red</code> 。預設：分別為 0、60、75 和 90

表格 2-4 訊息路由器屬性 (續)

屬性名稱	描述
imq.redelivered. optimization	每當重新遞送訊息 (true) 時，或僅當邏輯上需要如此操作 (false) 時，指定訊息路由器是否可透過設定重新遞送旗標最適化效能 (true/false)。預設：true
imq.transaction. autorollback	指定處於 PREPARED 狀態的分散式異動是否可在啟動代理程式時自動回轉 (true/false)。如果為 false，則您必須使用 imqcmd 手動確定或回轉異動 (請參閱第 141 頁的「管理異動」)。預設：false

持續性管理程式

如果要回復的代理程式出現故障，則此代理程式需要重新建立其訊息遞送作業的狀態。此操作要求它將所有永久性訊息以及重要的路由與遞送資訊儲存至資料儲存區。持續性管理程式元件管理此資訊的寫入和擷取。

若要回復發生故障的代理程式，不僅需要簡單地復原未遞送的訊息。代理程式還必須可執行以下作業：

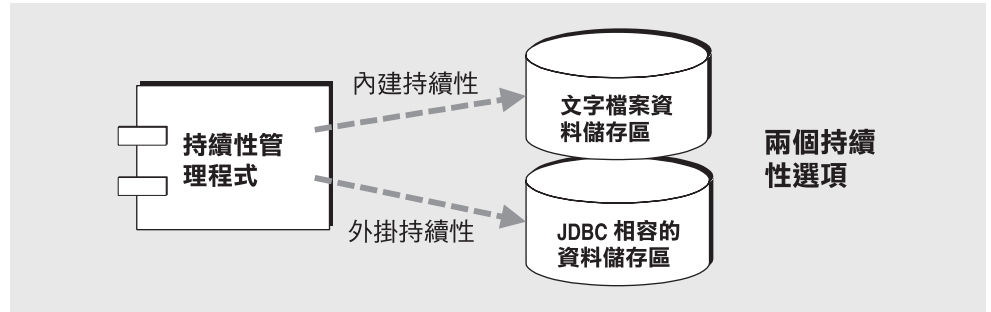
- 重新建立目標
- 復原每個主題的持久訂閱清單
- 復原每個訊息的回應清單
- 重新生產所有已確定異動的狀態

持續性管理程式管理所有此狀態資訊的儲存和擷取。

重新啟動代理程式時，它會重新建立目標和持久訂閱、回復永久性訊息、復原所有異動的狀態，並為未遞送訊息重新建立其路由表格。然後，它可以繼續訊息遞送。

MQ 支援內建和外掛持續性模組 (請參閱圖 2-4)。內建持續性基於文字檔案資料儲存區。外掛持續性使用 Java 資料庫連線 (JDBC) 介面，並需要 JDBC 相容的資料儲存區。內建持續性通常快於外掛持續性；但是某些使用者更喜歡使用了 JDBC 相容資料庫系統的容錯和管理功能。

圖 2-4 持續性管理程式支援



內建持續性

預設 MQ 永久性儲存體解決方案為文字檔案儲存區。這種方法使用個別檔案儲存永久性資料，如訊息、目標、持久訂閱以及異動。

文字檔案資料儲存區位於：

```
IMQ_VARHOME/instances/brokerName/filestore/
(在 Solaris 上為 /var/imq/instances/brokerName/filestore/)
```

其中代理程式名稱是識別代理程式實例的名稱。

建立基於檔案的資料儲存區，以便每個永久性訊息均儲存在它們各自的檔案中，即每個檔案儲存一個訊息。但是，目標、持久訂閱和異動則分別儲存在單獨檔案中，即所有目標儲存在一個檔案中，所有持久訂閱儲存在另一個檔案中，等等。

若要建立和刪除檔案，則需要昂貴的檔案系統作業，如同在資料儲存區中新增和刪除訊息。因此，MQ 實施重新使用這些訊息檔案：當不再需要檔案時，系統不是刪除它，而是將它加入自由檔案儲存區以便重新使用。您可以配置此檔案儲存區的容量。您還可以指定此檔案儲存區中被清除的自由檔案的百分比（截斷至零），而不是僅標記這些檔案以重新使用（未截斷）。已清除檔案的百分比越高，維護此檔案儲存區所需的磁碟空間就越少，但耗用時間就越多。您還可以指定已標記檔案是否在關機時將被清除。如果這些檔案被清除，則它們將佔用很少的磁碟空間，但關閉代理程式則需較長時間。

文字檔案儲存區中儲存訊息的速度，受可供資料儲存區使用的檔案描述元數量的影響；大量描述元可讓系統更快地處理大量永久性訊息。如需有關增加描述元數量的資訊，請參閱 MQ 版次注意事項中的「技術說明」一節。

此外，在目標檔案儲存區中，將目標新增至固定容量檔案比新增目標時增加檔案容量更有效。因此，根據您預計目標檔案最終儲存的目標數量（每個目標約使用 500 位元組），您可以透過設定此目標檔案的原來容量來提昇效能。

由於資料儲存區可包含專用資訊的訊息，因此建議您保護代理程式名稱/filestore/目錄，以防止未經授權的存取。如需說明，請參閱 MQ 版次注意事項中的「技術說明」一節。

外掛持續性

您可以設定代理程式，以透過 JDBC 驅動程式存取任何可存取的資料儲存區。這包含設定一些與 JDBC 相關的代理程式配置屬性，以及使用資料庫管理員公用程式 (imgdbmgr) 來建立具有適當綱目的資料儲存區。附錄 A 「設定外掛持續性」中詳細介紹了這些程序和相關的配置屬性

第 50 頁的表格 2-5 中詳細介紹了與持續性相關的配置屬性。(如需有關設定這些屬性的說明，請參閱第 5 章 「啟動與配置代理程式」。)

表格 2-5 持續性屬性

屬性名稱	描述
imq.persist.store	指定代理程式使用內建、基於檔案 (file) 的持續性，或使用外掛 JDBC 相容 (jdbc) 的持續性。預設：file
imq.persist.file.destination.file.size	用於內建、基於檔案的持續性，指定用於儲存目標的檔案的初始容量。預設：1m (百萬位元組)
imq.persist.file.message.filepool.limit	用於內建、基於檔案的持續性，指定檔案儲存區中可供重新使用的自由檔案的最大數量。此數量越大，代理程式處理永久性資料的速度就越快。超過此數量的自由檔案將被刪除。代理程式將依需要建立與刪除超過此限制的其他檔案。預設：10000
imq.persist.file.message.filepool.cleanratio	用於內建、基於檔案的持續性，指定檔案儲存區中被維護處於清除狀態的自由檔案的百分比 (截斷至零)。此值越高，則在作業期間清除檔案所需的耗用時間就越多，但檔案儲存區所需的磁碟空間就越少。預設：0
imq.persist.file.message.cleanup	用於內建、基於檔案的持續性，指定代理程式關閉時是否會清除檔案儲存區中的自由檔案。錯誤的值會加速代理程式的關閉，但檔案儲存區需要更多的磁碟空間。預設：false

表格 2-5 持續性屬性 (續)

屬性名稱	描述
imq.persist.file.message. fdpool.limit	用於內建、基於檔案的持續性，指定要保持開啓的資料檔案的最大數量 (即檔案描述元儲存區的容量)。此數量越大，便會提昇持續性作業的效能，但卻對需要檔案描述元的其他代理程式作業 (如建立用戶端連線) 產生不利影響。預設：25 (Solaris 和 Linux) 與 1024 (Windows)
imq.persist.file.sync. enabled	指定持續性作業是否與實體儲存裝置以記憶體狀態同步化。如果為 true，則會結束由系統崩潰引起的資料遺失，但會降低持續性作業的效能。預設：false

安全性管理程式

MQ 提供認證與授權 (存取控制) 功能，還支援加密功能。

認證與授權功能依賴使用者儲存庫 (請參閱第 52 頁的圖 2-5)：包含有關訊息傳送系統使用者的資訊 (如使用者名稱、密碼和群組成員身份) 的檔案、目錄或資料庫。當請求與代理程式連線時，這些名稱和密碼用於認證使用者。使用者名稱和群組成員身份，連同存取控制檔案一起用於授權作業 (如針對目標生產或使用訊息)。

MQ 管理員可移入 MQ 所提供的使用者儲存庫 (請參閱第 164 頁的「使用文字檔案使用者儲存庫」)，或將預先存在的 LDAP 使用者儲存庫插入安全性管理程式元件中。文字檔案使用者儲存庫易於使用，但其安全性易受侵害，因此應僅將其用於評估與開發，而 LDAP 使用者儲存庫很安全，因此最適合用於生產。

認證

MQ 安全性支援基於密碼的認證。當用戶端請求連線至代理程式時，它必須提交使用者名稱和密碼。安全性管理程式將此用戶端提交的名稱和密碼與儲存在使用者儲存庫中的名稱和密碼進行比較。將此密碼從用戶端傳送至代理程式時，系統會使用基本 64 編碼或訊息摘要 (MD5) 對密碼進行編碼。如需更加安全的傳送，請參閱第 53 頁的「加密 (企業版)」。您可以分別配置每種連線服務所使用的編碼類型，或在代理程式範圍基礎上設定編碼。

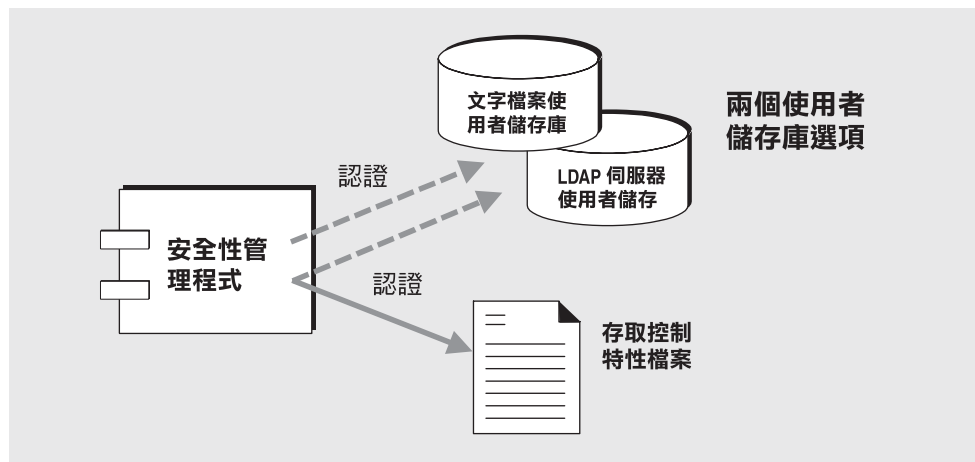
授權

一旦認證用戶端應用程式的使用者，此使用者便可被授權執行各種與 MQ 相關的作業。安全性管理程式支援基於使用者和基於群組的存取控制：根據使用者儲存庫中的使用者名稱或指定給此使用者的群組，此使用者可擁有執行某些 MQ 作業的許可權。您可在存取控制屬性檔案 (請參閱圖 2-5) 中指定這些存取控制。

當使用者嘗試執行作業時，安全性管理程式會檢查使用者名稱和群組成員身份 (從使用者儲存庫) 是否與為存取此作業指定的那些名稱和成員身份 (在存取控制屬性檔案中) 相符。存取控制屬性檔案可指定以下作業的許可權：

- 建立與代理程式的連線
- 存取目標：為任何給定目標或所有目標建立使用者、生產者或佇列瀏覽器
- 自動建立目標

圖 2-5 安全性管理程式支援



對於 MQ 3.0.1，預設存取控制屬性檔案僅明確參考一個群組：*admin* (請參閱第 167 頁的「群組」)。*admin* 群組中的使用者擁有管理服務連線許可權。管理服務可讓使用者執行管理功能，如建立目標以及監視與控制代理程式。依預設，您定義的任何其他群組中的使用者均無法取得管理服務連線。

作為 MQ 管理員，您可以定義群組，並將使用者與使用者儲存庫中的那些群組相關聯 (儘管群組在文字檔案使用者儲存庫中不能完全得到支援)。然後，透過編輯存取控制屬性檔案，您可以指定使用者和群組對目標的存取權，以用於生產和使用訊息，或在佇列目標中瀏覽訊息。您可以使個別目標或所有目標僅可由特定使用者或群組進行存取。

此外，如果代理程式已配置為允許自動建立目標 (請參閱第 59 頁的「自動建立的 (對應管理員建立的) 目標」)，則您可以透過編輯存取控制屬性檔案來控制代理程式可為何人自動建立目標。

加密 (企業版)

若要加密在用戶端與代理程式之間傳送的訊息，您需要使用基於安全套接層 (SSL) 標準的連線服務。透過在已啓用 SSL 的代理程式與已啓用 SSL 的用戶端之間建立已加密連線，SSL 可提供連線級別的安全性。

若要使用 MQ 基於 SSL 的連線服務，您將使用鍵值工具公用程式 (imqkeytool) 產生私密密鑰/公開密鑰對。此公用程式將公開密鑰內嵌在自身簽名的證書中，並放置在 MQ 鍵值儲存區中。MQ 鍵值儲存區自身具有密碼保護；若要對其解除鎖定，您必須在啟動時提供鍵值儲存區密碼。請參閱第 177 頁的「加密：使用 SSL 服務 (企業版)」。

一旦鍵值儲存區解除鎖定，則代理程式可將證書傳送至請求連線的任何用戶端。然後，用戶端使用此證書設定與代理程式的已加密連線。

表格 2-6 中顯示用於認證、授權、加密以及其他安全通訊的可配置屬性。(如需有關配置這些屬性的說明，請參閱第 5 章「啟動與配置代理程式」。)

表格 2-6 安全性屬性

屬性名稱	描述
imq.authentication.type	指定密碼應以基本 64 編碼 (basic) 傳送，或作為 MD5 摘要 (digest) 傳送。為代理程式支援的所有連線服務設定編碼。預設：digest
imq.service_name.authentication.type	指定密碼應以基本 64 程式碼 (basic) 傳送，或作為 MD5 摘要 (digest) 傳送。為已命名的連線服務設定編碼，並置換任何代理程式範圍的設定。預設：繼承 imq.authentication.type 設定的值。
imq.authentication.basic.user_repository	(用於基本 64 程式碼) 指定用於認證的使用者儲存庫的類型，即基於檔案 (file) 或 LDAP (ldap)。如需其他 LDAP 屬性，請參閱第 170 頁的表格 8-5。預設：file
imq.authentication.client.response.timeout	指定系統等待用戶端回應來自代理程式的認證請求的時間 (以秒為單位)。預設：180 (秒)
imq.accesscontrol.enabled	為代理程式支援的所有連線服務設定存取控制 (true/false)。表示系統是否將檢查已認證的使用者擁有使用連線服務的許可權，或擁有執行與特定目標相關的特定 MQ 作業的許可權 (如存取控制屬性檔案中所指定)。預設：true

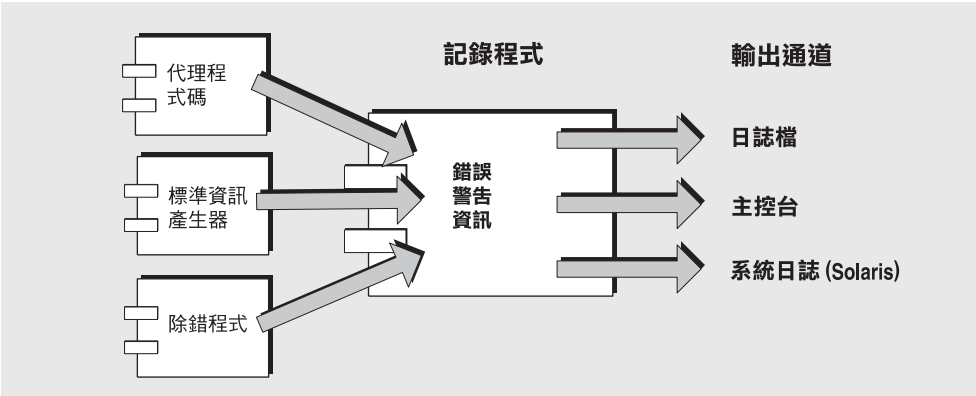
表格 2-6 安全性屬性 (續)

屬性名稱	描述
<code>imq.service_name.accesscontrol.enabled</code>	為已命名的連線服務設定存取控制 (<code>true/false</code>)，並置換代理程式範圍的設定。表示系統是否將檢查已認證的使用者擁有使用已命名連線服務的許可權，或擁有執行與特定目標相關的特定 MQ 作業的許可權 (如存取控制屬性檔案中所指定)。預設：繼承屬性 <code>imq.accesscontrol.enabled</code> 的設定
<code>imq.accesscontrol.file.filename</code>	為代理程式支援的所有連線指定存取控制屬性檔案的名稱。此檔案名稱指定目錄 <code>IMQ_HOME/etc</code> (在 Solaris 上為 <code>/etc/imq</code>) 的相對檔案路徑。預設： <code>accesscontrol.properties</code>
<code>imq.service_name.accesscontrol.file.filename</code>	為已命名的連線服務指定存取控制屬性檔案的名稱。此檔案名稱指定目錄 <code>IMQ_HOME/etc</code> (在 Solaris 上為 <code>/etc/imq</code>) 的相對檔案路徑。預設：繼承由 <code>imq.accesscontrol.file.filename</code> 指定的設定。
<code>imq.passfile.enabled</code>	指定是否已在密碼檔案中指定用於安全通訊的使用者密碼 (用於 SSL、LDAP、JDBC) (<code>true/false</code>)。預設： <code>false</code>
<code>imq.passfile.dirpath</code>	指定包含密碼檔案之目錄的路徑。 預設： <code>IMQ_HOME/etc</code> (在 Solaris 上為 <code>/etc/imq</code>)
<code>imq.passfile.name</code>	指定密碼檔案的名稱。預設： <code>passfile</code>
<code>imq.keystore.property_name</code>	用於基於 SSL 的服務：指定與 SSL 鍵值儲存區相關的安全性屬性。請參閱第 179 頁的表格 8-8。

記錄程式

代理程式包括一些用於監視和診斷其作業的元件。其中有產生資料 (代理程式碼、標準資訊產生器以及除錯程式) 的元件，還有可透過一些輸出通道 (日誌檔、主控台以及 Solaris 系統日誌) 寫出資訊的記錄程式元件。圖 2-6 中說明了此機制。

圖 2-6 記錄機制



您可以開啓或關閉標準資訊資料的產生，並指定產生標準資訊報告的頻率。

您還可以指定記錄程式級別 — 範圍從最嚴重和最重要的資訊 (錯誤) 至不太重要的資訊 (標準資訊資料)。表格 2-7 中以重要性的降序方式顯示了資訊的種類：

表格 2-7 記錄種類

種類	描述
ERROR	表示可導致系統故障問題的訊息
WARNING	應加以注意但不會導致系統故障的警示
INFO	標準資訊和其他資訊性訊息的報告

若要設定記錄程式級別，您可指定這些種類之一。記錄程式將寫出指定種類和所有更高級種類的資料。例如，如果您指定 `WARNING` 級別的記錄，則記錄程式將寫出警告資訊和錯誤資訊。

記錄程式可將資料寫入至一些輸出通道：標準輸出 (主控台)、日誌檔以及系統日誌常駐程式 (在 `Solaris` 平台上)。

對於每個輸出通道，您均可以指定為記錄程式設定的哪些種類將寫入至此通道。例如，如果記錄程式級別設定為 `ERROR`，則可以指定您僅將錯誤和警告寫入至主控台，以及僅將資訊 (標準資訊資料) 寫入至日誌檔。如需有關配置和使用 `Solaris` 系統日誌的資訊，請參閱 `syslog(1M)`、`syslog.conf(4)` 和 `syslog(3C)` 線上援助頁。

在日誌檔中，您可以指定何時關閉日誌檔並將輸出捲動至新檔案。一旦日誌檔達到指定容量或存在時間，系統將儲存此日誌檔並建立新的日誌檔。日誌檔儲存在以下位置：

```
IMQ_VARHOME/instances/brokerName/log/  
(在 Solaris 上為 /var/imq/instances/brokerName/log/)
```

當建立新的捲動日誌檔時，則 9 個最新日誌檔的歸檔檔案將被保留。日誌檔為文字檔案，它們相繼命名如下：

```
log.txt  
log_1.txt  
log_2.txt  
...  
log_9.txt
```

log.txt 為最新檔案，編號最高的檔案為最舊檔案。

表格 2-8 中顯示代理程式的可配置屬性，它們用於設定資訊的產生和記錄。(如需有關配置這些屬性的說明，請參閱第 5 章「啟動與配置代理程式」。)

表格 2-8 記錄程式屬性

屬性名稱	描述
imq.metrics.enabled	指定是否要收集標準資訊 (true/false)。預設：true
imq.metrics.interval	指定報告標準資訊的時間間隔 (以秒為單位)。值 0 表示永不報告。預設：0
imq.log.level	指定記錄程式級別：可寫入至輸出通道的輸出種類。包括指定種類以及所有更高級的種類。從高至低的值為：ERROR、WARNING 和 INFO。預設：INFO
imq.log.file.output	指定寫入至日誌檔的記錄資訊的種類。允許的值為：由垂直列 () 分隔的任何一組記錄種類，或者 ALL 或 NONE。預設：ALL
imq.log.file.dirpath	指定包含日誌檔之目錄的路徑。預設：IMQ_VARHOME/instances/ <i>brokerName</i> /log/ (在 Solaris 上為 /var/imq/instances/ <i>brokerName</i> /log/)
imq.log.file.filename	指定日誌檔的名稱。預設：log.txt

表格 2-8 記錄程式屬性 (續)

屬性名稱	描述
<code>imq.log.file.rolloverbytes</code>	指定日誌檔輸出捲動至新日誌檔時，此日誌檔的容量 (以位元組為單位)。值 0 表示沒有基於檔案容量的捲動。預設：0
<code>imq.log.file.rolloversecs</code>	指定日誌檔輸出捲動至新日誌檔時，此日誌檔的存在時間 (以秒為單位)。值 0 表示沒有基於檔案存在時間的捲動。預設：604800 (一週)
<code>imq.log.console.output</code>	指定寫入至主控台的記錄資訊的種類。允許的值為由垂直列 () 分隔的任何一組記錄種類，或者 ALL 或 NONE。預設：ERROR WARNING
<code>imq.log.console.stream</code>	指定主控台輸出寫入至 <code>stdout</code> (OUT)，或寫入至 <code>stderr</code> (ERR)。預設：ERR
<code>imq.log.syslog.facility</code>	(僅用於 Solaris) 指定 MQ 代理程式應記錄為的系統日誌設備。值鏡射 <code>syslog(3C)</code> 線上援助頁中列出的那些值。用於 MQ 的適當值為：LOG_USER、LOG_DAEMON 以及 LOG_LOCAL0 至 LOG_LOCAL7。預設：LOG_DAEMON
<code>imq.log.syslog.logpid</code>	(僅用於 Solaris) 指定是否同時記錄訊息與代理程式 ID (true/false)。預設：true
<code>imq.log.syslog.console</code>	(僅用於 Solaris) 指定如果訊息無法傳送至系統日誌，是否將它們寫入至系統主控台 (true/false)。預設：false
<code>imq.log.syslog.identity</code>	(僅用於 Solaris) 指定應前置於每個訊息的身份字串是否記錄至系統日誌。預設：imqbrokerd_ 以及代理程式實例名稱。
<code>imq.log.syslog.output</code>	(僅用於 Solaris) 指定寫入至 <code>syslogd(1M)</code> 的記錄資訊的種類。允許的值為由垂直列 () 分隔的任何記錄種類，或者 ALL 或 NONE。預設：ERROR

實體目標

MQ 訊息傳送以兩個階段的訊息遞送為前提：首先，從生產者用戶端至代理程式上的目標的訊息遞送，然後，從代理程式上的目標至一個或多個使用者用戶端的訊息遞送。有兩種類型的目標 (請參閱第 33 頁的「程式設計網域」)：佇列 (點對點遞送模型) 和主題 (出版/訂閱遞送模型)。這些目標表示在代理程式實體記憶體中的位置，進來的訊息在路由至使用者用戶端之前在這些位置進行整理。

您可以使用 MQ 管理工具建立實體目標 (請參閱第 136 頁的「管理目標」)。目標還可自動建立，如第 59 頁的「自動建立的 (對應管理員建立的) 目標」中所述。

本節描述兩種類型實體目標的屬性與運作方式：佇列與主題。

佇列目標

佇列目標用於點對點訊息傳送，其中訊息表示最終僅遞送至已在目標中註冊其興趣的一些使用者之一。當訊息從生產者用戶端到達時，它們會形成佇列並遞送至使用者用戶端。

已形成佇列的訊息的路由取決於佇列的遞送策略。MQ 實施三種佇列遞送策略：

- **單一** 此佇列僅可將訊息路由至一個訊息使用者。如果第二個訊息使用者嘗試註冊此佇列，則將被拒絕。如果已註冊訊息使用者離線，則不再發生訊息路由，並且訊息將被儲存，直至新的使用者註冊。
- **故障轉移 (企業版)** 此佇列可將訊息路由至多個訊息使用者，但僅在其主訊息使用者 (第一個註冊代理程式的使用者) 離線時才執行此作業。在此情況下，訊息將轉至下一個要註冊的使用者，並繼續路由至此使用者，直至此使用者發生故障，等等。如果訊息使用者未註冊，則訊息將被儲存，直至使用者註冊。
- **循環 (企業版)** 此佇列可將訊息路由至多個訊息使用者。假設一些使用者註冊一個佇列，則進入此佇列的第一個訊息將路由至已註冊的第一個訊息使用者，第二個訊息路由至已註冊的第二個使用者，等等。其他訊息以相同的順序路由至同組使用者。如果在使用者註冊佇列之前一些訊息已形成佇列，則這些訊息將被批次路由，以免湧向任一使用者。如果訊息使用者離線，則路由至此使用者的訊息將在剩餘的作用中使用者之間進行重新分布。由於進行重新分布，因此無法保證訊息遞送至使用者的順序與佇列接收它們的順序相同。

由於訊息可在佇列中保留較長一段時間，因此會出現記憶體資源問題。您不應為佇列配置太多的記憶體 (記憶體未完全利用)，也不應配置太少 (訊息將被拒絕)。考慮到基於每個佇列載入需求的靈活性，您可在建立佇列時設定實體屬性：佇列中訊息的最大數量、為佇列中訊息配置的最大記憶體，以及佇列中任何訊息的最大容量 (請參閱第 138 頁的表格 6-10)。

主題目標

主題目標用於出版/訂閱訊息傳送，其中訊息表示最終遞送至已在目標中註冊其興趣的所有使用者。當訊息從生產者到達時，它們會被路由至訂閱此主題的所有使用者。如果使用者已註冊此主題的持久訂閱，則他們不會在訊息遞送至此主題時成為作用中使用者 — 代理程式將儲存訊息直至使用者再次成為作用中使用者，然後遞送訊息。

訊息通常不會在主題目標中保留較長一段時間，因此通常不會出現大的記憶體資源問題。但是，您可以配置目標 (請參閱第 138 頁的表格 6-10) 接收的任何訊息所允許的最大容量。

自動建立的 (對應管理員建立的) 目標

由於 JMS 訊息伺服器是訊息傳送系統中的中央集線器，因此其效能和可靠性對企業應用程式的成功相當重要。由於目標可使用大量資源 (取決於它們處理的訊息數量和容量，以及註冊的訊息使用者的數量和持久性)，因此它們需要嚴密管理，以確保訊息伺服器的效能和可靠性。因此，MQ 管理員的標準慣例即為建立作為應用程式的目標、監視目標，以及必要時重新配置它們的資源需求。

然而，也可能有希望動態建立目標的情況。例如，在開發及測試循環期間，您可能希望代理程式依需要自動建立目標，而不需要管理員的介入。

MQ 支援這種自動建立功能。啟用自動建立後，每當 MessageConsumer 或 MessageProducer 嘗試存取不存在的目標時，代理程式便會自動建立目標。(用戶端應用程式的使用者必須擁有自動建立的權限 — 請參閱第 177 頁的「目標自動建立存取控制」)。

但是，當自動建立而不是明確建立目標時，可能導致不同的用戶端應用程式 (使用相同的目標名稱) 之間的衝突，或降低系統效能 (歸因於支援目標所需的資源)。由於此原因，當不再使用 MQ 自動建立的目標時，此目標將被代理程式自動銷毀：即，當此目標不再有訊息使用者用戶端且不再包含任何訊息時。如果代理程式重新啟動，則它將僅重新建立自動建立的目標 (如果這些目標它們包含永久性訊息)。

您可以使用表格 2-9 中所示的屬性配置 MQ 訊息伺服器，以啟用或停用自動建立功能。(如需有關配置這些屬性的說明，請參閱第 5 章「啟動與配置代理程式」。)

表格 2-9 自動建立配置屬性

屬性名稱	描述
<code>imq.autocreate.topic</code>	指定是否允許代理程式自動建立主題目標 (true/false)。預設：true
<code>imq.autocreate.queue</code>	指定是否允許代理程式自動建立佇列目標 (true/false)。預設：true
<code>imq.queue.deliverypolicy</code>	指定自動建立的佇列之預設遞送策略。值為：single、round-robin 或 failover。預設：single

暫存目標

暫存目標由用戶端應用程式明確建立和銷毀 (使用 JMS API)，這些應用程式需要目標以接收傳送至其他用戶端的訊息回覆。這些目標由代理程式維護，僅用於為建立目標所需連線的持續時間。只要暫存目標處於使用中，它無法由管理員銷毀，也無法由用戶端應用程式銷毀；即，如果它有作用中的訊息使用者。暫存目標不同於管理員建立的或自動建立的目标 (包含永久性訊息)，它們無法永久儲存，並當重新啟動代理程式時從不重新建立。它們對於 MQ 管理工具而言也不可視。

多重代理程式叢集 (企業版)

MQ 企業版支援使用多重互連的代理程式實例 (代理程式叢集) 的訊息伺服器的實施。叢集支援提供訊息伺服器的延展性。

隨著連線至代理程式的用戶端數量的增加，以及要遞送的訊息數量的增加，代理程式將最終超過資源限制 (如檔案描述元和記憶體限制)。提供增加的載入的一個方法為將多個代理程式 (即多個代理程式實例) 新增至 MQ 訊息伺服器，從而在多重代理程式間分布用戶端連線和訊息遞送。

您還可以使用多重代理程式最適化網路頻寬。例如，您可以在一組遠端代理程式間使用較慢、距離較長的網路連結，而使用較高速度的連結將用戶端連線至其各自的代理程式。

然而，使用代理程式叢集還有其他原因 (例如，提供具有不同使用者儲存庫的工作群組，或處理防火牆限制)，故障轉移不是這些原因之一。叢集中的代理程式無法用作出現故障的其他代理程式的自動備份。MQ 版本 3.0.1 不支援代理程式的自動故障轉移保護功能。(但是，應用程式可設計為使用多重代理程式，以實施自訂的故障轉移機制。)

第 112 頁的「使用叢集 (企業版)」中提供有關配置和管理代理程式叢集的資訊。

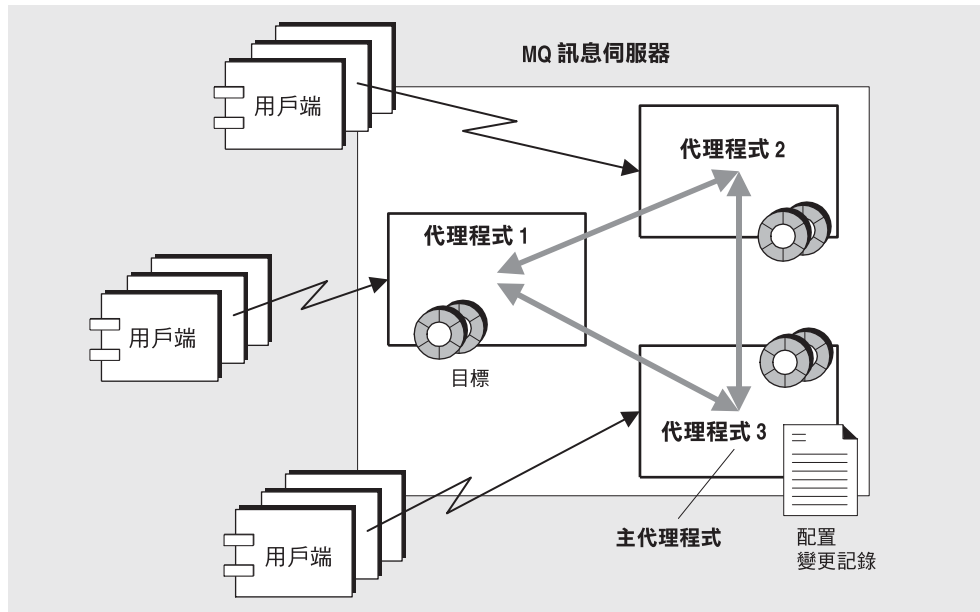
以下各節介紹 MQ 代理程式叢集的架構和內部功能。

多重代理程式架構

多重代理程式訊息伺服器允許用戶端連線分布在一些代理程式實例中，如圖 2-7 中所示。從用戶端的角度看，每個用戶端均連線至個別代理程式 (它的主代理程式)，並傳送和接收訊息，猶如此主代理程式為叢集中唯一的代理程式。但是，從訊息伺服器的角度看，主代理程式與叢集中的其他代理程式以串聯方式工作，為訊息生產者以及與其直接連線的使用者提供遞送服務。

一般來說，叢集中的代理程式可用任一隨機拓撲連線。但是，MQ 版本 3.0.1 僅支援完全連線的叢集 (即拓撲)，在拓撲中每個代理程式均直接連線至叢集中所有其他代理程式，如第 61 頁的圖 2-7 中所示。

圖 2-7 多重代理程式 (叢集) 架構



在多重代理程式配置中，每個目標的實例均常駐在叢集中所有代理程式上。此外，每個代理程式均瞭解已註冊所有其他代理程式的訊息使用者。因此，每個代理程式均可將訊息從其直接連線的訊息生產者路由至遠端訊息使用者，並將訊息從遠端生產者遞送至其直接連線的使用者。

在叢集配置中，與每個訊息生產者直接連線的代理程式為此生產者傳送至它的訊息執行路由。因此，永久性訊息由其主代理程式儲存和路由。

每當管理員在代理程式上建立或銷毀目標時，此資訊會自動傳遞至叢集中所有其他代理程式。同樣，每當訊息使用者註冊其主代理程式，或使用者與其主代理程式中斷連線（明確中斷連線，或者因為用戶端或網路故障，或者因為其主代理程式出現故障）時，有關使用者的相關資訊會在叢集中傳遞。在類似方式下，有關持久訂閱的資訊也會傳遞至叢集中的所有代理程式。

註 大量的網路流量和/或大量訊息可阻礙內部叢集連線。增加的等待時間有時可導致鎖定協定逾時錯誤。結果，用戶端可能在嘗試建立持久訂閱者或佇列訊息使用者時出現異常。使用較高速度的連線通常可避免這些問題。

將有關目標和訊息使用者的資訊傳遞至特定代理程式，通常會要求當共用資源發生變更時此代理程式應在線上。如果當進行此類變更時代理程式離線（例如，如果代理程式崩潰且隨後重新啟動，或者如果將新代理程式動態新增至叢集），將會如何？

為了提供已離線的代理程式（或已新增的新代理程式），MQ 將保留對叢集中所有永久性實體所作變更的記錄。即，已建立或銷毀的所有目標與所有持久訂閱的記錄。當代理程式動態新增至叢集時，它先從此**配置變更記錄**中讀取目標和持久訂閱者資訊。當它上線後，它將與其他代理程式交換有關目前作用中的使用者的資訊。新代理程式可使用此資訊完全整合至叢集。

配置變更記錄由叢集中的代理程式之一（即指定為**主代理程式**的代理程式）管理。由於主代理程式對於將代理程式動態新增至叢集極為關鍵，因此您應始終先啟動此代理程式。如果主代理程式不在線上，則叢集中的其他代理程式將無法完成它們的初始化。

如果主代理程式離線，則其他代理程式則無法存取配置變更記錄，而且 MQ 將不允許目標和持久訂閱在叢集中傳遞。在這些情況下，如果您嘗試建立或銷毀目標或持久訂閱（或嘗試執行一些重新啟動持久訂閱之類的相關作業），則會出現異常。

在任務重要的應用程式環境中，應定期備份配置變更記錄，以防止意外損毀記錄並避免主代理程式發生故障。您可以使用 `imqbrokerd` 指令的 `-backup` 選項（請參閱**第 109 頁的表格 5-2**）執行此作業，此選項可提供建立包含配置變更記錄的備份檔案的方法。隨後，您可以使用 `-restore` 選項復原此配置變更記錄。

如有必要，您可以透過備份配置變更記錄及修改適當的叢集配置屬性（請參閱**第 63 頁的表格 2-10**）來變更作為主代理程式服務的代理程式，以指定新的主代理程式，並使用 `-restore` 選項來重新啟動此新的主代理程式。

使用開發環境中的叢集

在開發環境中，叢集用於測試，而延展性與代理程式回復並非重要注意事項，因此不需要主代理程式。在未配置主代理程式的環境中，MQ 不需要主代理程式處於執行狀態以啟動其他代理程式，並且允許對目標與持久訂閱進行變更並將它們傳遞至叢集中所有執行中的代理程式。但是，如果代理程式離線並隨後復原，則它將不會與其離線時所作的變更同步。

在測試情況下，目標通常是自動建立的（請參閱第 59 頁的「自動建立的（對應管理員建立的）目標」），這些目標的持久訂閱由要進行測試的應用程式建立和銷毀。對目標和持久訂閱所作的這些變更將在叢集中傳遞。但是，如果您重新配置環境以使用主代理程式，則 MQ 將重新強加需求，即主代理程式處於執行狀態，以便對目標與持久訂閱進行變更，並使這些變更在叢集中傳遞。

叢集配置屬性

叢集中的每個代理程式必須在啟動時傳送有關叢集中其他代理程式的資訊（主機名稱與連接埠號）。此資訊用於在叢集中的代理程式之間建立連線。每個代理程式還必須知道主代理程式（如果已使用）的主機名稱和連接埠號。

叢集中的所有代理程式均應使用相同的叢集配置屬性。透過將叢集配置屬性放在啟動時每個代理程式均參考的一個中央叢集配置檔案中，您便可以達到此目的。

（您還可以複製這些配置屬性，並分別將它們提供給每個代理程式。但是，不建議這樣做，因為它可能導致叢集配置的不一致性。僅保留叢集配置屬性的副本可確保所有代理程式均參閱相同的資訊。）

表格 2-10 中顯示 MQ 叢集配置屬性。（如需有關設定這些屬性的說明，請參閱第 112 頁的「使用叢集（企業版）」。）

表格 2-10 叢集配置屬性

屬性名稱	描述
<code>imq.cluster.brokerlist</code>	指定叢集中的所有代理程式。由以逗號分隔的主機：連接埠項目清單組成，其中主機為每個代理程式的主機名稱，連接埠為它的連接埠對映器連接埠號。
<code>imq.cluster.masterbroker</code>	指定叢集中的哪個代理程式（如果有）為記錄狀態變更的主代理程式。屬性由主機：連接埠組成，其中主機為主代理程式的主機名稱，連接埠為它的連接埠對映器連接埠號。
<code>imq.cluster.url</code>	指定叢集配置檔案的位置。用於代理程式參考中央配置檔案，而不是用於叢集屬性值分別隨附的情況。由 URL 字串組成：如果保留在 Web 伺服器上，則可使用一般的 <code>http:URL</code> 存取它。如果保留在共用磁碟機上，則可使用 <code>file:URL</code> 存取它

表格 2-10 叢集配置屬性 (續)

屬性名稱	描述
imq.cluster.port	對於叢集中的每個代理程式，均可用於為叢集連線服務指定連接埠號。叢集連線服務用於叢集中代理程式之間的內部通訊。預設：0 (已動態配置連接埠)
imq.cluster.hostname	對於叢集中的每個代理程式，如果有多台主機可用 (例如，如果一台電腦中有多個網路介面卡)，則可用於指定叢集連線服務所連結的主機 (主機名稱或 IP 位址)。叢集連線服務用於叢集中代理程式之間的內部通訊。預設：null (所有可用的主機)

叢集配置檔案可用於儲存一組代理程式共用的所有代理程式配置屬性。儘管它原先專用於配置叢集，但是它還可用於儲存叢集中所有代理程式共用的其他代理程式屬性。

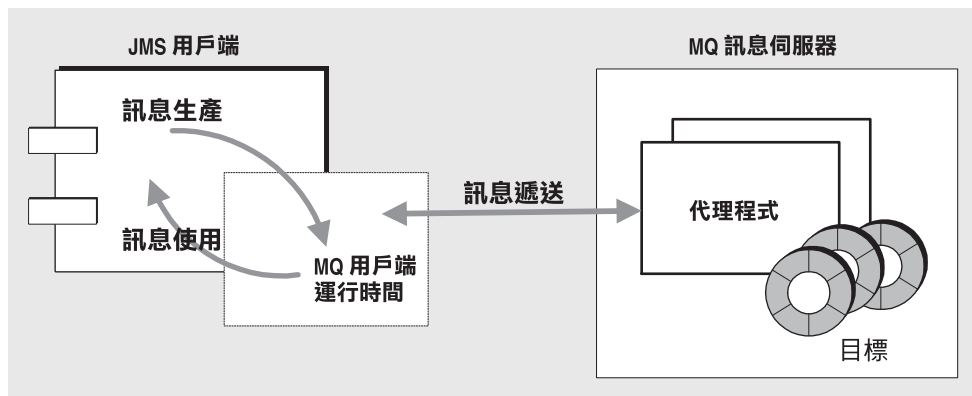
MQ 用戶端運行時間

MQ 用戶端運行時間為用戶端應用程式提供一個至 MQ 訊息伺服器的介面，它為用戶端應用程式提供第 29 頁的「JMS 程式設計模型」中介紹的所有 JMS 程式設計物件。它支援用戶端將訊息傳送至目標和從此類目標接收訊息所需的所有作業。

本節提供 MQ 用戶端運行時間如何工作的詳細描述。MQ 開發人員指南中說明了影響其效能的因素，因為這些因素影響用戶端應用程式的設計和效能。

第 65 頁的圖 2-8 說明訊息生產與使用如何在用戶端應用程式與 MQ 用戶端運行時間之間進行互動，而訊息遞送如何在 MQ 用戶端運行時間與 MQ 訊息伺服器之間進行互動。

圖 2-8 訊息傳送作業



訊息生產

在訊息生產中，訊息由用戶端建立，並透過連線傳送至代理程式上的目標。如果 MessageProducer 物件的訊息遞送模式已設定為永久性 (可確保的遞送，一次且僅有一次)，則用戶端執行緒會封鎖，直至此代理程式回應訊息已遞送至其目標並儲存在代理程式的永久性資料儲存區中。如果訊息不是永久性的，則代理程式不會傳回代理程式回應訊息 (在屬性名稱中稱為「Ack」)，並且用戶端執行緒不會封鎖。

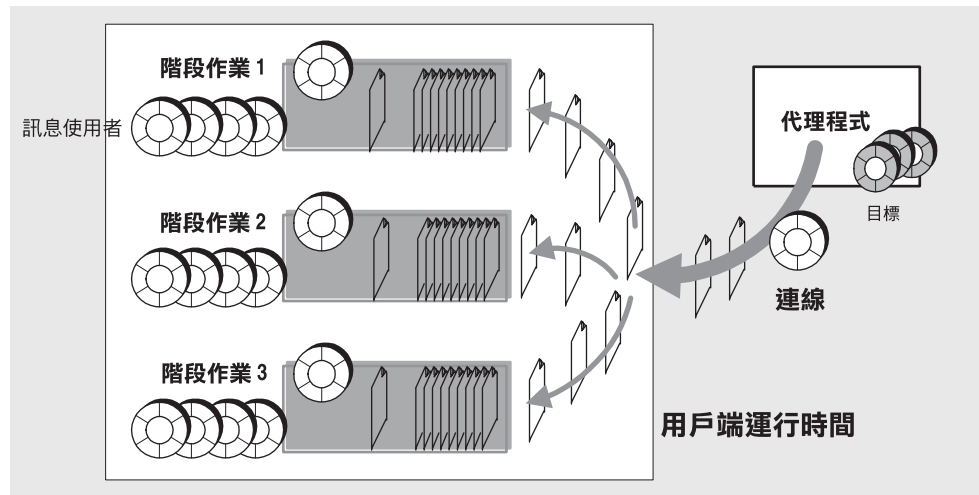
訊息使用

訊息使用比生產更為複雜。在以下條件下，到達代理程式上目標的訊息會透過連線遞送至 MQ 用戶端運行時間：

- 用戶端已為給定的目標設定使用者
- 使用者的選取條件 (如果有) 符合到達給定目標的訊息的條件
- 系統已告知連線啟動訊息遞送

透過連線遞送的訊息分布至適當的 MQ 階段作業，在此階段作業中，這些訊息形成佇列以供適當的 MessageConsumer 物件使用，如圖 2-9 中所示。訊息從每個階段作業佇列取回，每次一個 (階段作業為單一執行緒的)，並同步使用 (透過呼叫 receive 方法的用戶端執行緒) 或非同步使用 (透過呼叫 MessageListener 物件的 onMessage 方法的階段作業執行緒)。

圖 2-9 訊息遞送至 MQ 用戶端運行時間



當代理程式將訊息遞送至用戶端運行時間時，它會相應地標記訊息，但無法真正知道它們是否已被接收或使用。因此，在將訊息從代理程式目標刪除之前，代理程式會等待用戶端回應訊息的接收。

MQ 受管理物件

受管理物件允許用戶端應用程式碼獨立於供應程式。透過將供應程式特定的實施與配置資訊封裝在由用戶端應用程式以獨立於供應程式的方式使用的物件中，它們可達到此目的。受管理物件由管理員建立和配置，儲存在名稱服務中，並由用戶端應用程式透過標準的 JNDI 查找程式碼存取。

MQ 提供兩種類型的受管理物件：ConnectionFactory 與目標。當這兩種類型的受管理物件封裝供應程式特定的資訊時，它們會使用完全不同的用戶端應用程式。ConnectionFactory 物件用於建立與訊息伺服器的連線，而目標物件則用於識別實體目標。

受管理物件使控制和管理 MQ 訊息伺服器非常容易：

- 透過要求用戶端應用程式存取預先配置的 ConnectionFactory 物件 (請參閱第 150 頁的「受管理物件屬性」)，您可以控制連線的運作方式。
- 透過要求用戶端應用程式存取與現有實體目標相符的預先配置的目標物件，您可以控制實體目標的激增。(您還必須停用代理程式的自動建立功能 — 請參閱第 59 頁的「自動建立的 (對應管理員建立的) 目標」)。

- 透過置換用戶端應用程式設定的訊息標頭值 (請參閱第 150 頁的「受管理物件屬性」)，您可以控制 MQ 訊息伺服器資源。

因此，此排序可讓您 (作為 MQ 管理員) 控制訊息伺服器配置的詳細資訊，同時還允許用戶端應用程式獨立於供應程式：它們不必瞭解供應程式特定的語法和物件命名慣例 (請參閱第 32 頁的「JMS 提供者獨立性」)，或供應程式特定的配置屬性。

您可以使用 MQ 管理工具建立受管理物件，如第 7 章「管理受管理物件」中所述。建立受管理物件時，您可以指定其為唯讀 — 即防止用戶端應用程式變更建立物件時已設定的 MQ 特定的配置值。換言之，用戶端程式碼無法設定有關唯讀受管理物件的屬性值，您也無法使用用戶端應用程式啟動選項來置換這些值，如第 68 頁的「啟動用戶端時置換屬性值」中所述。

而用戶端應用程式可在其上創設 `ConnectionFactory` 和目標受管理物件，此做法會逐漸損壞受管理物件的基本用途 — 可讓您 (作為 MQ 管理員) 控制應用程式所需的代理程式資源，並調整其效能。此外，直接創設受管理物件可使用用戶端應用程式為供應程式特定的，而不是獨立於供應程式的。

連線工廠受管理物件

`ConnectionFactory` 物件用於在用戶端應用程式與 MQ 訊息伺服器之間建立實體連線。它還用於指定連線與用戶端運行時間的運作方式，此用戶端運行時間使用連線存取代理程式。

如果您希望支援分散式異動 (請參閱第 35 頁的「本地異動」)，則需要使用可支援分散式異動的特殊 `XAConnectionFactory` 物件。

若要建立 `ConnectionFactory` 受管理物件，請參閱第 157 頁的「新增連線工廠」。

透過配置 `ConnectionFactory` 受管理物件，您可以指定此受管理物件生產的所有連線共用的屬性值 (屬性)。`ConnectionFactory` 與 `XAConnectionFactory` 物件共用同一組屬性。根據這些屬性影響的運作方式，可將它們群組為一些種類：

- 連線規格
- 自動重新連線運作方式
- 用戶端識別
- 訊息標頭置換
- 可靠性與流量控制
- 佇列瀏覽器運作方式

- 應用程式伺服器支援
- JMS 定義的屬性支援

MQ 開發人員指南中詳細說明了這些種類中的每種及其相應的屬性。而系統可能要求您 (作為 MQ 管理員) 調整這些屬性的值，通常由應用程式開發人員決定需要調整哪些屬性以調整用戶端應用程式的效能。[第 151 頁的表格 7-3](#) 描述了按字母順序列出的屬性摘要。

目標受管理物件

目標受管理物件表示代理程式中的實體目標 (佇列或主題)，共用名稱為目標的物件與此實體目標相符。[表格 2-11](#) 中描述了它的兩個屬性。透過建立目標物件，您可允許用戶端應用程式的 MessageConsumer 和/或 MessageProducer 物件來存取相應的實體目標。

若要建立目標受管理物件，請參閱[第 158 頁的「新增主題或佇列」](#)。

表格 2-11 目標屬性

屬性/屬性名稱	描述
imqDestinationName	指定供應程式特定的實體目標名稱。您可在建立實體目標時指定此名稱。目標名稱必須僅包含字母數字字元 (無空格)，並且以字母字元或字元「_」和「\$」開頭。預設：Untitled_Destination_Object
imqDestinationDescription	指定對管理物件有用的資訊。預設：A Description for the Destination Object

啟動用戶端時置換屬性值

如同任何 Java 應用程式，您可以使用指令行啟動訊息傳送應用程式，以指定系統屬性。此機制還可用於置換受管理物件的屬性值，這些屬性值用於用戶端應用程式碼。例如，您可以置換受管理物件的配置，此受管理物件可在用戶端應用程式碼中透過 JNDI 查找存取。

若要在啟動用戶端應用程式時置換受管理物件的設定，您可以使用以下指令行語法：

```
java [-D attribute=value] ... clientAppName
```

其中，屬性與第 150 頁的「連線工廠受管理物件」中提供的任何 `ConnectionFactory` 受管理物件屬性相符。

例如，如果您要將用戶端應用程式連線至不同的代理程式，而不是在用戶端程式碼中存取的 `ConnectionFactory` 受管理物件中所指定的代理程式，您可以使用指令行置換來啟動用戶端應用程式，以設定其他代理程式的 `imqBrokerHostName` 和 `imqBrokerHostPort`。

但是，如果受管理物件已設定為唯讀，則使用指令行置換無法變更其屬性值。任何此類置換均被完全忽略。

MQ 管理

Sun™ ONE Message Queue (MQ) 管理由一些工作以及一些用於執行這些工作的工具組成。

本章首先提供了管理工作簡介，然後描述了管理工具，並重點介紹指令行管理公用程式的共用功能。

MQ 管理工作

您需要執行的特定工作取決於您是處於開發環境還是生產環境。

開發環境

在開發環境中，工作重點為 MQ 用戶端應用程式的程式設計。MQ 訊息伺服器主要用於測試。在開發環境中強調靈活性，而管理並不重要，它主要體現在為開發人員啟動代理程式以用於測試。資料儲存區的預設實施、使用者儲存庫、存取控制屬性檔案以及物件儲存區，對於開發測試通常已經足夠。如果您要執行多重代理程式測試，則可能不使用主代理程式。此外，要測試的應用程式通常可以使用自動建立的目標，您可能不需要使用中央管理的受管理物件。

生產環境

在生產環境中，由於必須可靠地佈署和執行應用程式，因此管理會變得更加重要。您必須執行的管理工作取決於訊息傳送系統的複雜性及其必須支援的應用程式的複雜性。但是一般來說，這些工作可以群組為設定作業與維護作業。

設定作業

通常，您必須至少執行以下設定作業中的一些作業（如果不是全部）：

- 安全性（請參閱第 8 章「安全性管理」）：
 - 將項目放入基於檔案的使用者儲存庫，或將代理程式配置為使用現有 LDAP 使用者儲存庫
(至少，您需要密碼保護管理功能。)
 - 修改存取控制屬性檔案中的存取設定
 - 設定基於 SSL 的連接服務
- 受管理物件（請參閱第 7 章「管理受管理物件」）：
 - 配置或設定 LDAP 物件儲存區
 - 建立 ConnectionFactory 和目標受管理物件
- 代理程式叢集（請參閱第 112 頁的「使用叢集（企業版）」）：
 - 建立中央配置檔案
 - 使用主代理程式
- 持續性：將代理程式配置為使用外掛持續性，而不是使用內建持續性（請參閱附錄 A「設定外掛持續性」）

維護作業

此外，在生產環境中，需要密切監視和控制 MQ 訊息伺服器資源。應用程式效能、可靠性以及安全性是必需的，您必須使用 MQ 管理工具執行一些不間斷的工作，如下所述：

- 應用程式管理：
 - 停用代理程式的自動建立功能（請參閱第 60 頁的表格 2-9）
 - 建立作為應用程式的實體目標（請參閱第 137 頁的「建立目標」）
 - 設定使用者對目標的存取權（請參閱第 172 頁的「授權使用者：存取控制屬性檔案」）
 - 監視與管理目標（請參閱第 136 頁的「管理目標」）
 - 監視與管理持久訂閱（請參閱第 140 頁的「管理持久訂閱」）
 - 監視與管理異動（請參閱第 141 頁的「管理異動」）

- 代理程式管理與調整：
 - 使用代理程式標準資訊調整和重新配置代理程式
 - 管理代理程式記憶體資源
 - 將代理程式新增至叢集以平衡負載
 - 回復失敗的代理程式
- 管理受管理物件
 - 依需要建立其他 `ConnectionFactory` 和目標受管理物件
 - 調整 `ConnectionFactory` 屬性值以提昇效能與流量 (請參閱第 67 頁的「[連線工廠受管理物件](#)」)

MQ 管理工具

MQ 管理工具分為兩類：指令行公用程式與圖形使用者介面 (GUI) 管理主控台 (`imqadmin`)。主控台合併了兩種指令行公用程式的功能：指令公用程式 (`imqcmd`) 和物件管理員公用程式 (`imqobjmgr`)。您可以使用主控台 (和這兩種指令行公用程式) 來遠端管理代理程式以及管理 MQ 受管理物件。其他指令行公用程式 (`imqbrokerd`、`imqusermgr`、`imqdbmgr` 和 `imqkeytool`) 必須與它們相關的代理程式在相同主機上執行，如圖 3-1 中所示。

有關管理主控台的資訊可從線上說明獲得。通常用於執行專用工作的指令行公用程式，在「[指令行公用程式摘要](#)」中有描述。

管理主控台

您可以使用管理主控台執行以下作業：

- 連線至代理程式並管理它。
- 在代理程式上建立實體目標
- 連線至物件儲存區
- 將受管理物件新增至物件儲存區。

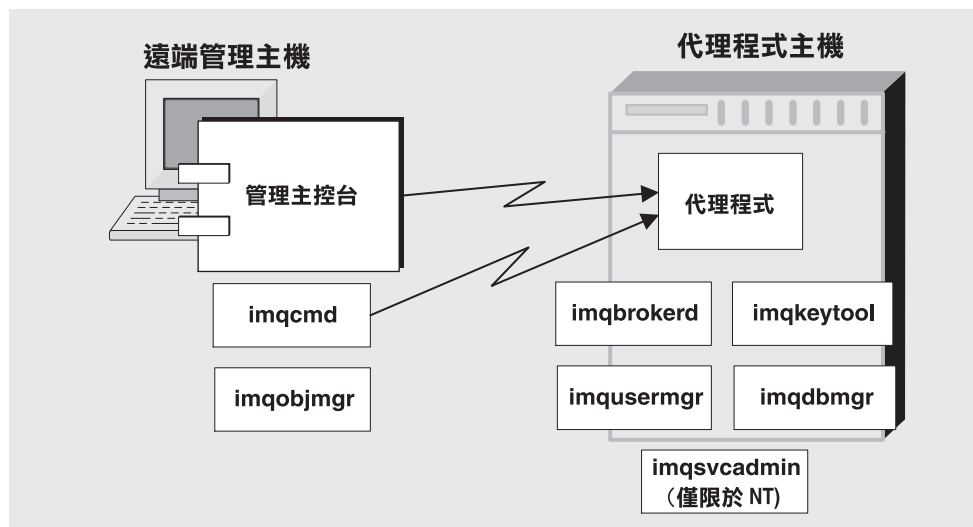
您無法使用管理主控台執行某些工作；其中主要包括啟動代理程式、建立代理程式叢集、配置代理程式的更加專用的屬性以及管理使用者資料庫。

[第 4 章「管理主控台指導」](#) 提供了簡短的實用指導，可讓您熟悉主控台並說明如何使用主控台完成基本的工作。

指令行公用程式摘要

本節介紹用於執行 MQ 管理工作的指令行公用程式。您可以使用 MQ 公用程式啟動與管理代理程式，並執行其他更加專用的管理工作。

圖 3-1 本機與遠端管理公用程式



所有 MQ 公用程式均可透過指令行介面 (CLI) 存取。公用程式指令共用常用格式、語法慣例和選項，如本章後續小節中所述。在後續各章中，您可以找到有關使用指令行公用程式的更加詳細的資訊。

代理程式 (imqbrokerd) 您可以使用代理程式公用程式啟動代理程式。您可以使用 imqbrokerd 指令的選項指定是否應在叢集中連線代理程式，並指定其他配置資訊。[第 5 章「啟動與配置代理程式」](#)中描述了此公用程式。

指令 (imqcmd) 啟動代理程式之後，您可以使用指令公用程式建立、更新和刪除實體目標；控制代理程式及其連接服務；管理代理程式的資源。您可以使用 imqcmd 指令執行此公用程式。[第 6 章「代理程式和應用程式管理」](#)中描述了此公用程式。

物件管理員 (imqobjmgr) 您可以使用物件管理員公用程式在透過 JNDI 可存取的物件儲存區中新增、列示、更新和刪除受管理物件。透過將 JMS 用戶端與 JMS 供應程式特定的命名和配置格式分離，受管理物件可讓 JMS 用戶端獨立於供應程式。您可以使用 imqobjmgr 指令執行此公用程式。[第 7 章「管理受管理物件」](#)中描述了此公用程式。

使用者管理員 (imqusermgr) 您可以使用使用者管理員公用程式移入用於認證和授權使用者的基於檔案的使用者儲存庫。您可以使用 `imqusermgr` 指令執行此公用程式。第 8 章「安全性管理」中描述了此公用程式。

鍵值工具 (imqkeytool) 您可以使用鍵值工具公用程式產生用於 SSL 認證的自身簽名證書。您可以使用 `imqkeytool` 指令執行此公用程式，如第 8 章「安全性管理」和附錄 B「HTTP/HTTPS 支援 (企業版)」中所述。

資料庫管理員 (imqdbmgr) 您可以使用資料庫管理員公用程式建立和管理用於永久性儲存的 JDBC 相容資料庫。您可以使用 `imqdbmgr` 指令執行此公用程式。如需更多資訊，請參閱附錄 A「設定外掛持續性」。

服務管理員 (imqsvcadm) 您可以使用服務管理員公用程式安裝、查詢和移除作為 Windows 服務的代理程式。如需更多資訊，請參閱附錄 C「將代理程式用作 Windows 服務」。

指令行語法

MQ 指令行介面公用程式為簡單的 shell 指令。即，從輸入這些公用程式的 Windows、Linux 或 Solaris 指令 shell 觀點看，公用程式本身的名稱即為一個指令，其子指令或選項僅為傳送至此指令的引數。因此，沒有指令可啟動或退出公用程式，而且就公用程式本身而言也無需此類指令。

所有指令行公用程式共用以下指令語法：

```
Utility_Name [subcommand] [argument] [[-option_name
[-option_argument]]...]
```

`Utility_Name` 指定 MQ 公用程式的名稱，例如 `imqcmd`、`imqobjmgr` 以及 `imqusermgr` 等。

必須記住以下四項重要內容：

- 在子指令 (和引數，如果公用程式接受兩種類型的運算元) 之後指定選項。
- 如果引數含有空格，請將整個引數括在引號內。通常將屬性值對括在引號內是最安全的。
- 如果您在指令行上指定 `-v` (版本) 或 `-h/-H` (說明) 選項，則系統不會執行此指令行上的其他任何選項。請參閱第 76 頁的表格 3-1，以獲得共用選項的描述。
- 使用空格將子指令、引數、選項和選項引數分隔。

以下為沒有子指令子句的指令行範例。此指令啟動預設代理程式。

```
imqbrokerd
```

以下指令略有些複雜：此指令為名為 `admin` 的管理員 (使用者) (其相應密碼為 `admin`) 銷毀名為 `myQueue` 的 `queue` 類型目標，並且未經確認，主控台上亦無輸出顯示。

```
imqcmd destroy dst -t q -n myQueue -u admin -p admin -f -s
```

共用指令行選項

表格 3-1 描述了所有 MQ 管理公用程式共用的選項。除了在指令行上指定子指令之後指定這些選項的要求之外，下述選項 (或任何其他傳送至公用程式的選項) 不必以特殊順序輸入。

表格 3-1 共用 MQ 指令行選項

選項	描述
-h	顯示指定公用程式的用法說明。
-H	顯示延伸用法說明，包括屬性清單和範例 (僅支援 <code>imqcmd</code> 和 <code>imqobjmgr</code>)。
-s	開啓靜謐模式：不顯示輸出。為 <code>imqbrokerd</code> 指定 <code>-silent</code> 。
-v	顯示版本資訊。
-f	執行給定動作而不提示使用者確認。
-pre	(僅用於 <code>imqobjmgr</code>) 開啓預覽模式，可讓使用者不用實際執行指令即可看到指令行剩餘部分的執行效果。這對檢查預設屬性的值很有用。
-javahome <i>path</i>	指定要使用的替代 Java 2 運行時間的位置。

管理主控台指導

本指導重點介紹管理主控台這種管理 Sun™ ONE Message Queue (MQ) 訊息伺服器的圖形介面的使用。遵循本指導，您將學習如何執行以下作業：

- 啟動代理程式以及使用主控台連線至代理程式並管理它
- 在代理程式上建立實體目標
- 建立物件儲存區並使用主控台連線至此物件儲存區
- 將受管理物件新增至物件儲存區

本指導旨在設定執行簡單 JMS 相容應用程式 SimpleJNDIClient 所需的目標和受管理物件。在本指導的最後部分，您將執行此應用程式。

提供的本指導主要是爲了指導您使用管理主控台執行基本的管理工作。本指導不能替代整個「MQ 開發人員指南」或此**管理員指南**的其他章。

使用圖形工具無法完成某些 MQ 管理工作；您將需要使用指令行公用程式執行以下此類作業：

- 配置某些代理程式屬性

某些代理程式屬性無法使用管理主控台來配置。可以如第 101 頁的**第 5 章「啟動與配置代理程式」**或第 132 頁的**「更新代理程式」**中所述來配置這些代理程式屬性。

- 建立代理程式叢集

請參閱第 112 頁的**「使用叢集 (企業版)」**，以獲得更多資訊。

- 管理使用者資料庫

請參閱第 164 頁的**「認證使用者」**，以獲得更多資訊。

準備工作

您必須安裝 MQ 產品之後才可以啟動本指導。如需更多資訊，請參閱「MQ 安裝指南」。請注意，本指導用於 Windows，並針對 Unix 使用者加入備註。

在本指導中，選擇 [Item1] > [Item2] > [Item3] 表示您應下拉名為 Item1 的功能表，從此功能表中選擇 Item2，然後從 Item2 提供的選取中選擇 Item3。

啟動管理主控台

管理主控台是您用於執行以下作業的圖形工具：

- 建立代理程式的參考並連線至代理程式
- 管理代理程式
- 在代理程式上建立實體目標，代理程式可使用這些實體目標遞送訊息
- 連線至您可將 MQ 受管理物件放在其中的物件儲存區

受管理物件可讓您管理 JMS 相容應用程式的訊息傳送需要。如需更多資訊，請參閱第 66 頁的「MQ 受管理物件」。

► 啟動管理主控台

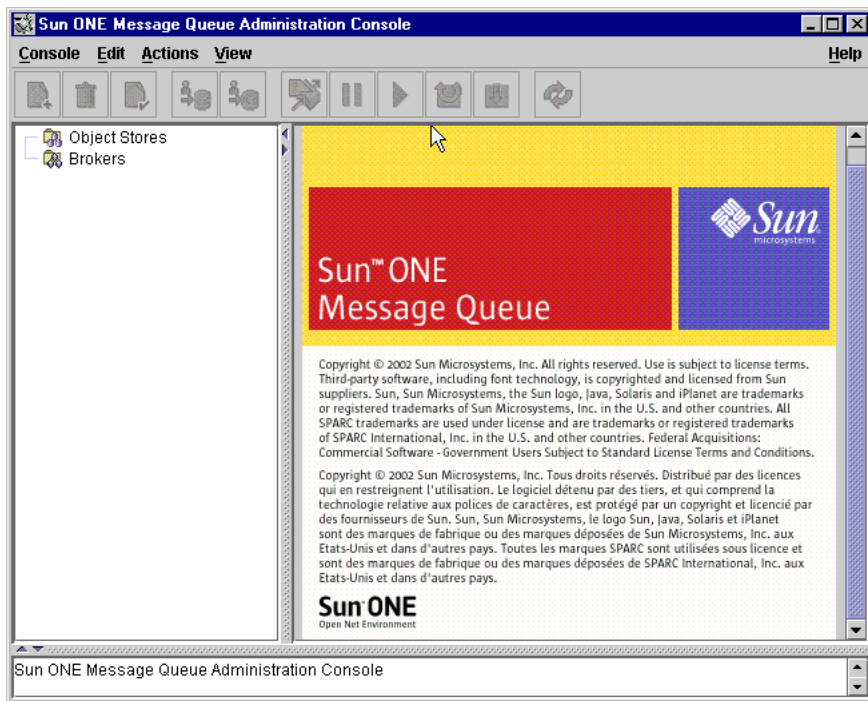
1. 選擇 [開始] > [程式集] > [Sun ONE Message Queue 3.0] > [Administration]。

可能需要等待幾秒鐘之後才會顯示主控台視窗。

非 Windows 使用者：在指令提示符下輸入以下指令：

```
$IMQ_HOME/bin/imqadmin (在 Solaris 上為 /usr/bin/imqadmin)
```

2. 花費幾秒鐘查看主控台視窗。



主控台視窗特徵為，頂部有一個功能表，緊挨功能表下面為工具列，視窗左側為導覽窗格，右側為較大的窗格（此時正顯示識別 Sun ONE Message Queue 產品的圖形），視窗底部為狀態窗格。

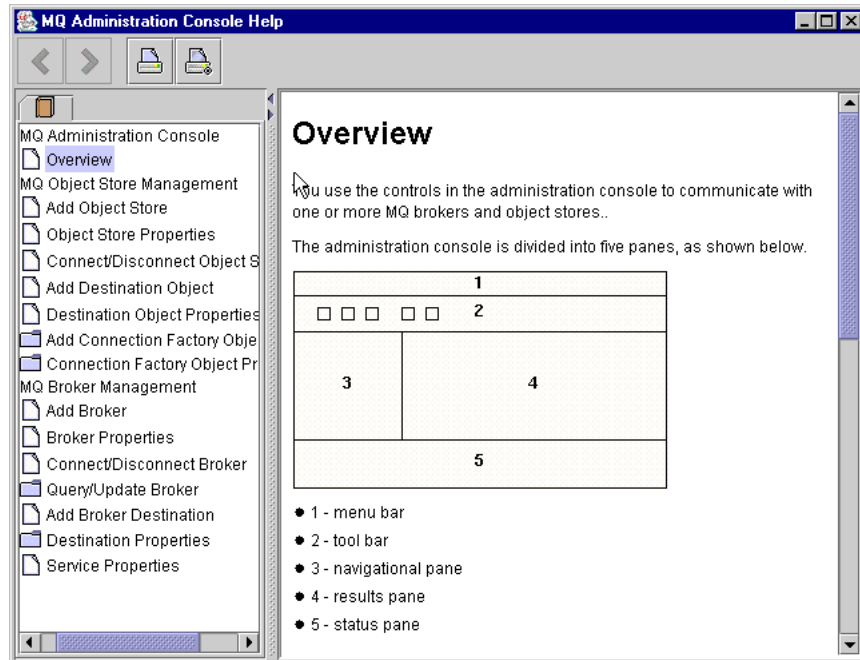
沒有指導可以提供完全的資訊，因此讓我們首先找到取得管理主控台說明資訊的方法。

取得說明

在功能表列的最右端可以找到 [Help] 功能表。

► 顯示管理主控台說明資訊

1. 下拉 [Help] 功能表並選擇 [Overview]。畫面上將會顯示說明視窗。



請注意說明資訊的組織方式。左側窗格顯示目錄表；右側窗格顯示您在左側所選取項目的內容。

查看 [Help] 視窗的右側窗格。它顯示管理主控台的框架視區，可識別每個主控台窗格的使用。

2. 查看 [Help] 視窗的目錄窗格。它以三個區域管理主題：簡介、物件儲存區管理和代理程式管理。每個區域均包含檔案與資料夾。每個資料夾為包含多個標籤的對話方塊提供說明；每個檔案為單個對話方塊或標籤提供說明。

您的第一項主控台管理工作（第 83 頁的「新增代理程式」）將是建立可透過主控台管理的代理程式的參考。但是開始之前，請先檢查線上說明以獲得資訊。

3. 在 [Help] 視窗的目錄窗格中，按一下 [Add Broker] 項目。

請注意內容窗格已經變更。它此時包含介紹新增代理程式涵義的文字，還包含描述 [Add Broker] 對話方塊中每個欄位之使用的文字。欄位名稱以粗體文字顯示。

4. 閱讀此說明文字。
5. 關閉 [Help] 視窗。

使用代理程式

代理程式為 MQ 訊息傳送系統提供遞送服務。訊息遞送過程分為兩個階段：首先將訊息遞送至代理程式上的實體目標，然後再將其遞送至一個或多個使用用戶端。

使用代理程式包括以下工作：

- 啟動並配置代理程式

您可以從 Windows 上的 [開始] > [程式集] 功能表啟動代理程式，或透過使用 `imqbrokerd` 指令啟動。如果您使用 `imqbrokerd` 指令，則可以使用指令行選項來指定代理程式配置資訊。如果您使用 [程式集] 功能表，則可以使用主控台以及第 5 章「[啟動與配置代理程式](#)」中所述的其他方法來指定配置資訊。

註 您無法使用管理主控台啟動代理程式實例。

- 管理代理程式及其服務，方法為透過使用管理主控台或透過使用指令指令行公用程式 (`imqcmd`)。
- 建立用戶端應用程式所需的實體目標
- 監視資源使用以提昇流量與可靠性

代理程式支援與應用程式用戶端以及管理用戶端的通訊。它透過不同的連線服務進行通訊，因此您可以將代理程式配置為執行這些服務的任何一項或者全部。如需有關連線服務的更多資訊，請參閱第 42 頁的「[連線服務](#)」。

啟動代理程式

您無法使用管理主控台啟動代理程式。啟動代理程式的步驟如下所述 (也可參閱第 5 章「啟動與配置代理程式」)。

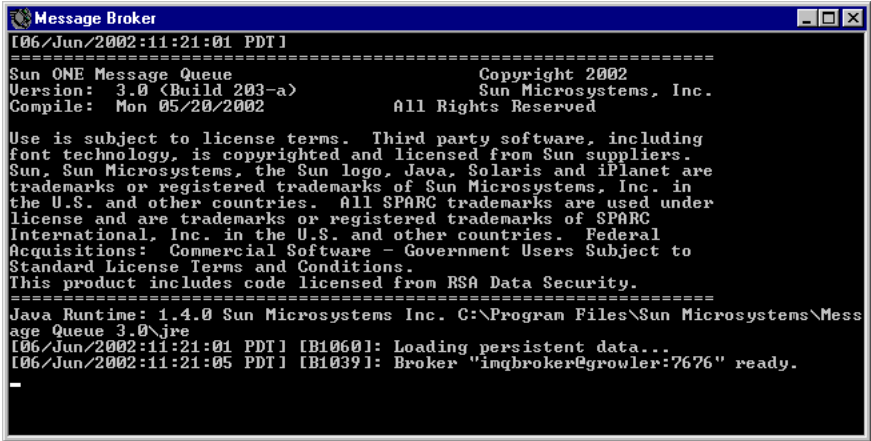
► 啟動代理程式

1. 選擇 [開始] > [程式集] > [Sun ONE Message Queue 3.0] > [Broker]。

非 Windows：輸入以下指令啟動代理程式。

`;%$IMQ_HOME/bin/imqbrokerd` (在 Solaris 上為 `/usr/bin/imqbrokerd`)

畫面上將會顯示代理程式程序視窗。代理程式的名稱已指定，表明此代理程式已準備就緒。



```

Message Broker
[06/Jun/2002:11:21:01 PDT]
=====
Sun ONE Message Queue                      Copyright 2002
Version: 3.0 <Build 203-a>                 Sun Microsystems, Inc.
Compile: Mon 05/20/2002                    All Rights Reserved

Use is subject to license terms. Third party software, including
font technology, is copyrighted and licensed from Sun suppliers.
Sun, Sun Microsystems, the Sun logo, Java, Solaris and iPlanet are
trademarks or registered trademarks of Sun Microsystems, Inc. in
the U.S. and other countries. All SPARC trademarks are used under
license and are trademarks or registered trademarks of SPARC
International, Inc. in the U.S. and other countries. Federal
Acquisitions: Commercial Software - Government Users Subject to
Standard License Terms and Conditions.
This product includes code licensed from RSA Data Security.
=====
Java Runtime: 1.4.0 Sun Microsystems Inc. C:\Program Files\Sun Microsystems\Mess
age Queue 3.0\jre
[06/Jun/2002:11:21:01 PDT] [B1060]: Loading persistent data...
[06/Jun/2002:11:21:05 PDT] [B1039]: Broker "imqbroker@growler:7676" ready.
  
```

2. 返回至管理主控台視窗。此時，您可以將代理程式新增至主控台並連線至此代理程式。

在向管理主控台新增代理程式參考之前，您不必啟動此代理程式，但是您必須啟動代理程式之後才可以與其連線。

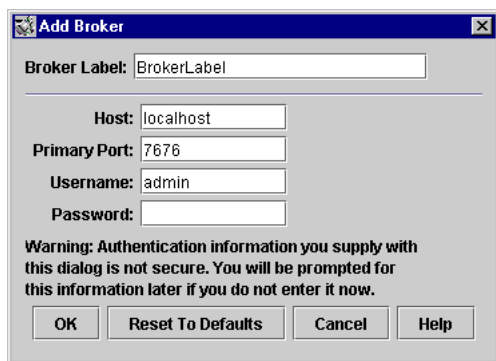
新增代理程式

新增代理程式可建立管理主控台中代理程式的參考。新增代理程式之後，您可以連線至此代理程式。

► 將代理程式新增至管理主控台

1. 在導覽窗格中的 [Brokers] 上按一下滑鼠右鍵，然後選擇 [Add Broker]。
2. 在 [Broker Label] 欄位中輸入 MyBroker。

這可提供識別管理主控台中代理程式的標籤。

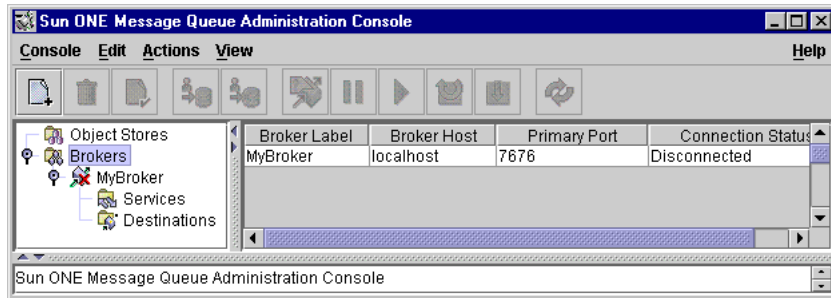


請注意對話方塊中指定的預設主機名稱 (localhost) 和主要連接埠 (7676)。稍後當您配置連線工廠 (用戶端將用來設定與此代理程式的連線) 時，您將需要指定這些值。

保留 [Password] 欄位為空白。如果您在連線時指定密碼將更安全。

3. 按一下 [OK] 以新增此代理程式。

查看導覽窗格。您剛新增的代理程式應已在 [Brokers] 下列出。代理程式圖示上的紅色 X 表示此代理程式目前尚未連線至主控台。



4. 在 [MyBroker] 上按一下滑鼠右鍵，然後從蹦現式功能表中選擇 [Properties]。
畫面上將會顯示代理程式屬性對話方塊。您可以使用此對話方塊更新您在新增此代理程式時指定的任何屬性。

變更管理員密碼

當您連線至代理程式時，如果您未在新增代理程式時指定密碼，則系統會提示您提供密碼。為了更高的安全性，最好在連線之前變更預設管理員密碼 (admin)。

► 變更管理員密碼

1. 開啓指令提示符號視窗，或者如果已開啓，請將其置於前面。
2. 輸入以下指令，以用 `abracadabra` 替代您自己的密碼。這樣您指定的密碼便可取代預設密碼 `admin`。

```
imqusermgr update -u admin -p abracadabra
```

(在 Solaris 和 Linux 上，您必須為超級使用者才能執行此作業。)

變更會立即生效。每當您使用 MQ 指令行公用程式或管理主控台之一時，您必須指定新的密碼。

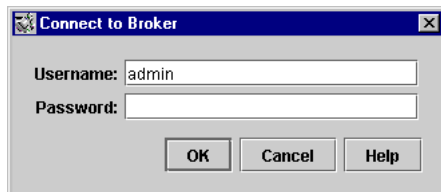
雖然用戶端與管理員使用不同的連線服務，但是系統還為用戶端指定了預設使用者名稱和密碼，以便您可以測試 MQ 而不必進行大量的管理設定。依預設，用戶端可以用使用者 `guest`，密碼 `guest` 連線至代理程式。但是，您應儘快為用戶端建立安全的使用者名稱和密碼。請參閱第 164 頁的「認證使用者」，以獲得更多資訊。

連線至代理程式

► 連線至代理程式

1. 在 MyBroker 上按一下滑鼠右鍵，然後選擇 [Connect to Broker]。

畫面上將會顯示對話方塊，可讓您指定您的名稱和密碼。



2. 在 [Password] 欄位中輸入 admin，或輸入您在第 84 頁的「變更管理員密碼」中為密碼指定的任何值。

指定使用者名稱 admin 並提供正確的密碼可讓您連線至代理程式，並擁有管理權限。

3. 按一下 [OK] 以連線至此代理程式。

連線至此代理程式之後，您可以從 [Actions] 功能表中選擇以獲得有關代理程式的資訊、暫停並繼續代理程式、關閉並重新啟動代理程式以及與代理程式中斷連線。

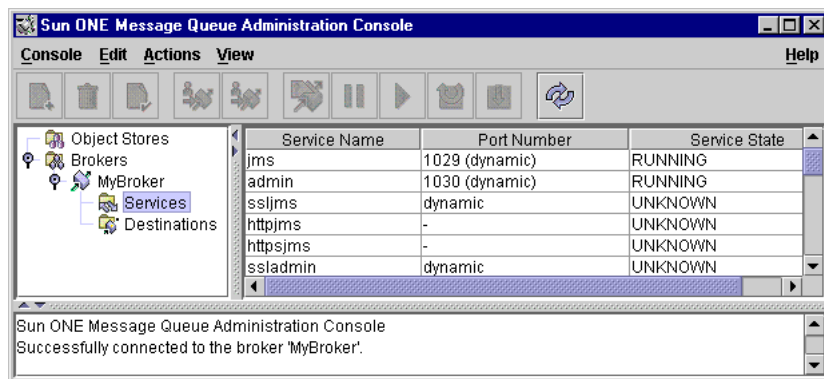
檢視連線服務

透過代理程式提供的連線服務和它支援的實體目標可以識別此代理程式。

► 檢視可用的連線服務

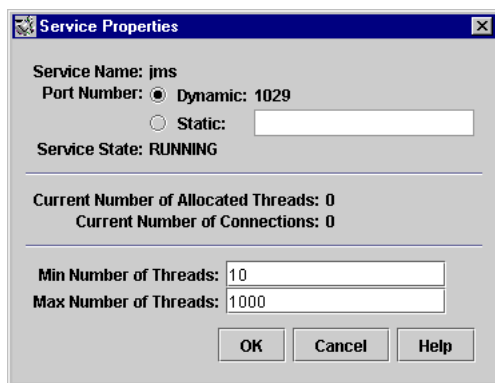
1. 在導覽窗格中選取 [Services]。

可用的服務會在結果窗格中列出。對於每種服務，均會提供其名稱、連接埠號與狀態。



2. 按一下結果窗格中的 **jms** 服務選取此服務。
3. 下拉 [Actions] 功能表並注意反白顯示的項目。
您便可以暫停 **jms** 服務或檢視與更新其屬性。
4. 從 [Actions] 功能表中選擇 [Properties]。

請注意，透過使用 [Service Properties] 對話方塊，您可以為服務指定靜態連接埠號，還可以變更配置給此服務的最小和最大執行緒數量。



5. 按一下 [OK] 或 [Cancel] 以關閉 [Properties] 對話方塊。
6. 在結果窗格中選取 admin 服務。
7. 下拉 [Actions] 功能表。

請注意您無法暫停此服務 (暫停項目已停用)。admin 服務是管理員與代理程式的連結。如果暫停此服務，您將無法再存取代理程式。

8. 選擇 [Actions] > [Properties] 以檢視 admin 服務的屬性。
9. 當您完成之後，按一下 [OK] 或 [Cancel]。

將實體目標新增至代理程式

您必須在代理程式上明確建立實體目標，以便 JMS 相容應用程式可以正常執行。如果代理程式已啟用目標自動建立功能 (此功能可讓代理程式動態地建立實體目標)，則您無需明確建立實體目標。

在開發環境中，可以使用目標自動建立功能。但是在生產設定中，最好應關閉目標自動建立功能並讓代理程式使用您已明確建立的實體目標。這樣可讓您 (管理員) 充分地瞭解在代理程式上使用的目標。

透過設定 `imq.autocreate.topic` 或 `imq.autocreate.queue` 屬性，您可以控制代理程式是否可以新增自動建立的目標。如需更多資訊，請參閱第 59 頁的「自動建立的 (對應管理員建立的) 目標」。

在本節指導中，您可將實體目標新增至代理程式。您應注意指定給此目標的名稱；稍後當您建立與此實體目標相對應的受管理物件時將需要此名稱。

► 將佇列目標新增至代理程式

1. 在 MyBroker 的 [Destinations] 節點上按一下滑鼠右鍵，然後選擇 [Add Broker Destination]。

畫面上將會顯示以下對話方塊：

A screenshot of the 'Add Broker Destination' dialog box. The dialog has a title bar with a close button. Inside, there is a text field for 'Destination Name' containing 'MyQueueDest'. Below it, 'Destination Type' has two radio buttons: 'Queue' (selected) and 'Topic'. Under 'Queue Delivery Policy', there are three radio buttons: 'Single' (selected), 'Round Robin', and 'Fail Over'. The 'Max Total Size of Messages' section has a radio button for 'Unlimited' (selected) and a text field with '0' and a 'bytes' dropdown. The 'Max Number of Messages' section has a radio button for 'Unlimited' (selected) and a text field with '0'. The 'Max Size per Message' section has a radio button for 'Unlimited' (selected) and a text field with '0' and a 'bytes' dropdown. At the bottom are four buttons: 'OK', 'Reset To Defaults', 'Cancel', and 'Help'.

2. 在 [Destination Name] 欄位中輸入 MyQueueDest。
3. 如果尚未選取 [Queue] 單選按鈕，請選取此按鈕。
4. 確定將 [Queue Delivery Policy] 選取為 [Single]。
5. 按一下 [OK] 以新增此實體目標。

此時此目標將會在結果窗格中出現。

使用實體目標

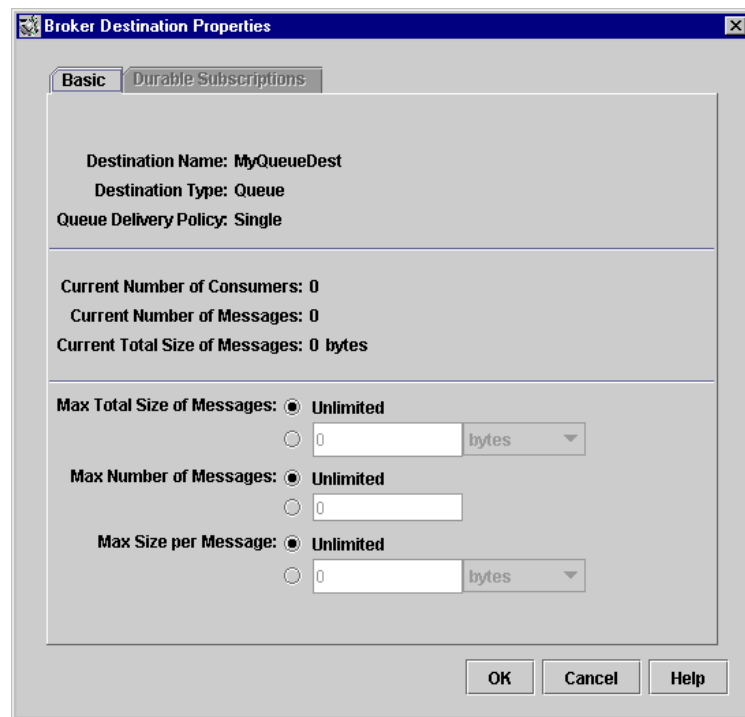
一旦您已在代理程式上新增實體目標，您便可以進行以下任何工作，如以下程序中所述：

- 檢視與更新實體目標的屬性
- 清除目標上的訊息
- 刪除目標

► 檢視實體目標的屬性

1. 選取 MyBroker 的 [Destinations] 節點。
2. 在結果窗格中選取 [MyQueueDest]。
3. 選擇 [Actions] > [Properties]。

畫面上將會顯示以下對話方塊：



請注意，您僅可變更與遞送至佇列的訊息大小和數量有關的佇列屬性。

4. 按一下 [Cancel] 以關閉此對話方塊。

► 從目標中清除訊息

1. 在結果窗格中選取實體目標。
2. 選擇 [Actions] > [Purge Messages]。

畫面上將會顯示確認對話方塊。

清除訊息可移除訊息並保留空白目標。

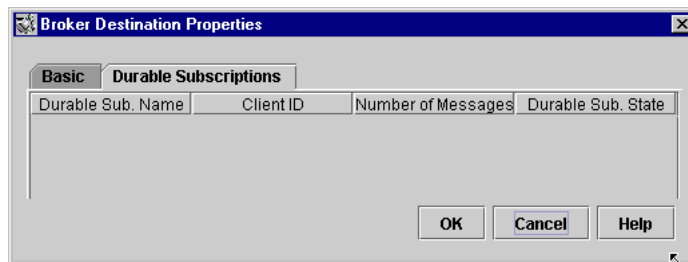
► 刪除目標

1. 在結果窗格中選取實體目標。
2. 選擇 [Edit] > [Delete]。

刪除目標可清除此目標上的訊息並移除此目標。

取得有關主題目標的資訊

有關主題目標的對話方塊包括列示持久訂閱資訊的附加標籤。



您可以使用此對話方塊：

- 清除持久訂閱以及移除與持久訂閱相關聯的所有訊息
- 刪除持久訂閱、清除與持久訂閱相關聯的所有訊息以及移除持久訂閱

使用物件儲存區

物件儲存區為 LDAP 目錄伺服器或檔案系統儲存區 (檔案系統中的目錄)，用於儲存 MQ 受管理物件，這些受管理物件封裝 MQ 特定的有關用戶端應用程式所用物件的實施與配置資訊。

雖然可以在用戶端節點中創設與配置受管理物件，但是您 (管理員) 最好在物件儲存區中建立、配置並儲存這些物件，用戶端應用程式可透過標準 JNDI 查找程式碼存取此物件儲存區。這樣可讓用戶端程式碼獨立於供應程式。

如需有關受管理物件的更多資訊，請參閱第 66 頁的「MQ 受管理物件」。

您無法使用管理主控台建立物件儲存區。您必須提前進行此作業，如以下一節中所述。

新增物件儲存區

新增物件儲存區可在管理主控台中建立現有物件儲存區的參考。即使您退出並重新啟動主控台，此參考仍可保留。

► 新增檔案系統物件儲存區

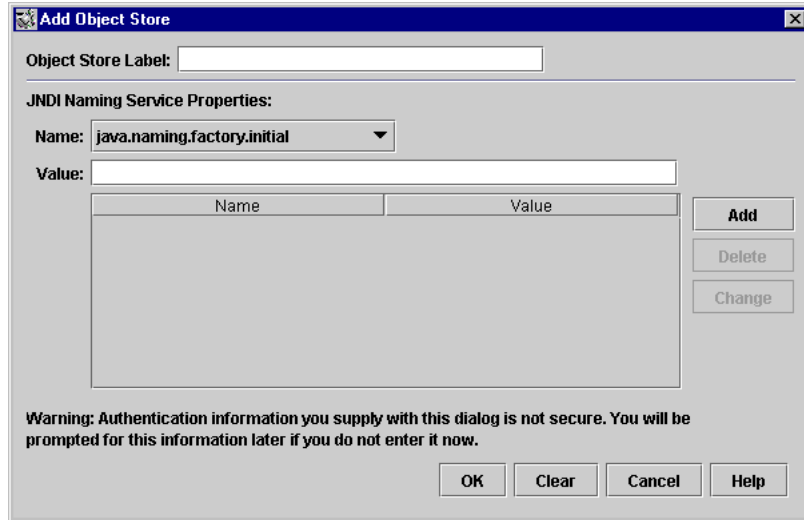
1. 如果您的 C 磁碟機上還沒有名為 Temp 的資料夾，請現在建立一個。

本指導中所用範例應用程式假設物件儲存區為 C 磁碟機上名為 Temp 的資料夾。一般來說，檔案系統物件儲存區可以為任何磁碟機上的任何目錄。

非 Windows：您可以使用 /tmp 目錄，此目錄應已存在。

2. 在 [Object Stores] 上按一下滑鼠右鍵，然後選擇 [Add Object Store]。

畫面上將會顯示以下對話方塊：



3. 在名為 [ObjectStoreLabel] 的欄位中輸入 MyObjectStore。

此項僅可為管理主控台中的物件儲存區顯示提供標籤。

在以下步驟中，您將需要輸入 JNDI 名稱/值對。JMS 相容應用程式將使用這些名稱/值對來查找受管理物件。

4. 從 [Name] 下拉式功能表中選擇 java.naming.factory.initial。

此屬性可讓您指定要使用何種 JNDI 服務供應程式。例如，檔案系統服務供應程式或 LDAP 服務供應程式。

5. 在 [Value] 欄位中輸入以下內容

```
com.sun.jndi.fscontext.RefFSContextFactory
```

這表示您將使用檔案系統儲存區。(對於 LDAP 儲存區，您將指定 com.sun.jndi.ldap.LdapCtxFactory。)

在生產環境中，您可能要使用 LDAP 目錄伺服器作為物件儲存區。如需有關設定伺服器以及進行 JNDI 查找的資訊，請參閱第 153 頁的「物件儲存區屬性」。

6. 按一下 [Add] 按鈕。

請注意此時屬性摘要窗格中列出的屬性及其值。

7. 從 [Name] 下拉式功能表中選擇 `java.naming.provider.url`。

此屬性可讓您指定物件儲存區的確切位置。對於檔案系統類型的物件儲存區，此項將為現有目錄的名稱。

8. 在 [Value] 欄位中輸入以下內容

```
file:///C:/Temp
```

(在 Solaris 和 Linux 上為 `file:///tmp`)

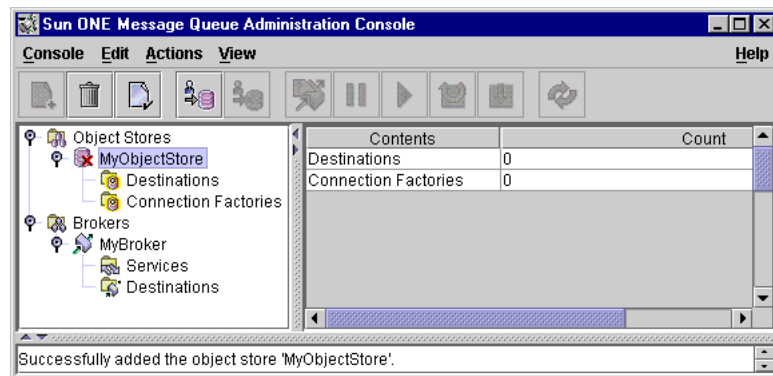
9. 按一下 [Add] 按鈕。

請注意此時屬性摘要窗格中列出的屬性及其值。如果您要使用 LDAP 伺服器，您可能還必須指定認證資訊；而檔案系統儲存區庫則無需此項。

10. 按一下 [OK] 以新增此物件儲存區。

11. 如果尚未在導覽窗格中選取節點 [MyObjectStore]，請現在選取它。

管理主控台此時顯示如下：



此物件儲存區已在導覽窗格中列出，儲存區內容、目標以及連線工廠已在結果窗格中列出。我們尚未將任何受管理物件新增至此物件儲存區，這可在結果窗格的 [Count] 欄中顯示。

導覽窗格中的物件儲存區圖示上畫有紅色的 X。這表示此物件儲存區已中斷連線。在使用物件儲存區之前，您將需要與其連線。

檢查物件儲存區屬性

當中斷管理主控台與物件儲存區的連線時，您可以檢查與變更此物件儲存區的某些屬性。

► 顯示物件儲存區的屬性

1. 在導覽窗格中的 [MyObjectStore] 上按一下滑鼠右鍵。
2. 從蹦現式功能表中選擇 [Properties]。

畫面上將會顯示對話方塊，此對話方塊顯示您新增此物件儲存區時指定的所有屬性。您可以變更這些屬性中的任何一項，然後按一下 [OK] 以更新舊的資訊。

3. 按一下 [OK] 或 [Cancel] 以關閉此對話方塊。

連線至物件儲存區

您必須連線至物件儲存區之後才可以將物件新增至此儲存區。

► 連線至物件儲存區

1. 在導覽窗格中的 [MyObjectStore] 上按一下滑鼠右鍵。
2. 從蹦現式功能表中選擇 [Connect to Object Store]。

請注意物件儲存區的圖示已不再有叉號。此時您可以將物件、連線工廠和目標新增至此物件儲存區。

新增連線工廠受管理物件

您可以使用管理主控台建立和配置連線工廠。用戶端程式碼使用連線工廠連線至代理程式。透過配置連線工廠，您可以控制連線工廠通常建立的連線運作方式。

如需有關配置連線工廠的資訊，請參閱線上說明與「MQ 開發人員指南」。

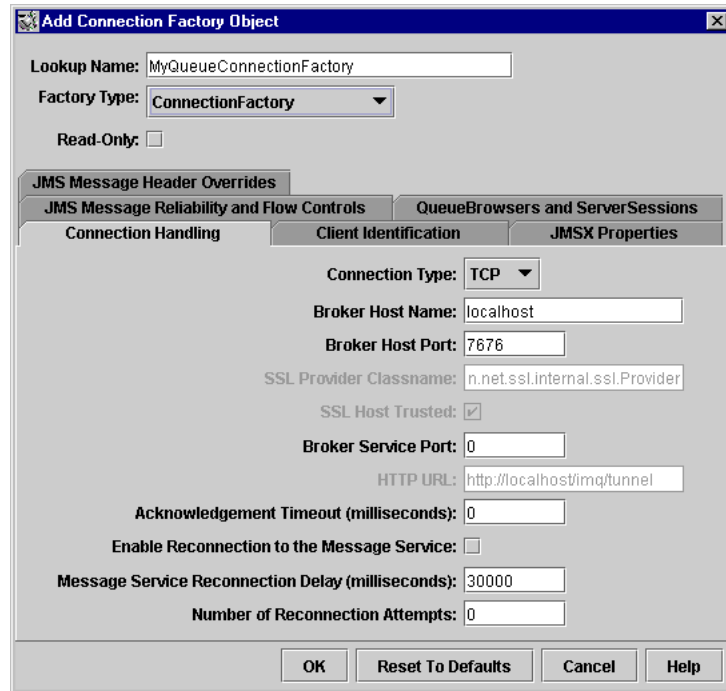
註	管理主控台僅列出與顯示 MQ 受管理物件。如果物件儲存區應包含的非 MQ 物件與您想新增的受管理物件具有相同的查找名稱，則當您嘗試新增作業時會出現錯誤。
----------	--

► 將連線工廠新增至物件儲存區

1. 如果尚未連線，請連線至 MyObjectStore (請參閱第 94 頁的「[連線至物件儲存區](#)」)

2. 在 [Connection Factories] 節點上按一下滑鼠右鍵，然後選擇 [Add Connection Factory Object]。

畫面上將會顯示 [Add Connection Factory Object] 對話方塊。



3. 在 [LookupName] 欄位中輸入名稱「MyQueueConnectionFactory」。

此名稱為用戶端程式碼查找連線工廠時使用的名稱，如 SimpleAdmin.java 的以下行中所示：

```
qcf=(javax.jms.QueueConnectionFactory)
    ctx.lookup("MyQueueConnectionFactory")
```

4. 從下拉式功能表中選取 [QueueConnectionFactory]，以指定連線工廠的類型。
5. 在 [Broker Host Name] 與 [Broker Host Port] 欄位中，輸入用戶端要與其連線的代理程式的主機名稱和連接埠。

在本指導中，用戶端連線至預設代理程式，即使用連接埠 7676 的 localhost 上的代理程式，因此您不必變更這些欄位。

6. 按一下此對話方塊的所有標籤，以查閱您可以為連線工廠配置的資訊種類。使用 [Add Connection Factory Object] 對話方塊右下角的 [Help] 按鈕，以獲得有關個別標籤的資訊。此時請勿變更任何預設值。
7. 按一下 [OK] 以建立此佇列連線工廠。
8. 查看結果窗格：已列出新建立的連線工廠的查找名稱與類型。

新增目標受管理物件

目標受管理物件與代理程式上的實體目標相關聯；它們指向那些目標，也就是說可讓用戶端查找並找到實體目標，而不依賴於供應程式特定的命名與配置那些目標的方法。

當 JMS 用戶端傳送訊息時，它會查找 (或創設) 目標受管理物件，並以 JMS API 的 `send()` 方法參考此物件。然後，代理程式負責將訊息遞送至與受管理物件相關聯的實體目標：

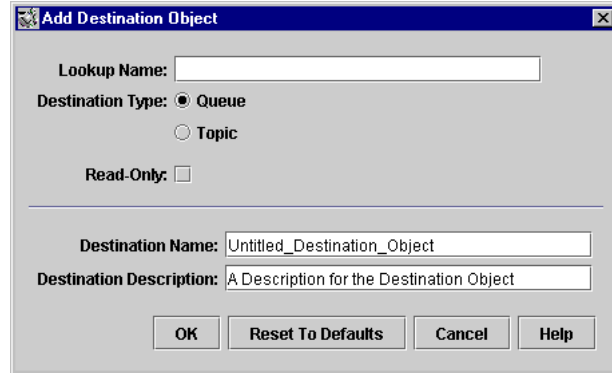
- 如果您已建立與此受管理物件相關聯的實體目標，則代理程式會將訊息遞送至此實體目標。
- 如果您尚未建立實體目標且已啟用實體目標的自動建立功能，則代理程式自身會建立實體目標並將訊息遞送至此目標。
- 如果您尚未建立實體目標且已停用實體目標的自動建立功能，則代理程式無法建立實體目標，亦無法遞送訊息。

在本指導的下一部分，您將要新增與您先前新增的實體目標相對應的受管理物件。

► 將目標新增至物件儲存區

1. 在導覽窗格的中的 [Destinations] 節點上按一下滑鼠右鍵。
2. 選擇 [Add Destination Object]。

管理主控台會顯示 [Add Destination Object] 對話方塊，您可用來指定有關此物件的資訊。



The image shows a Java Swing dialog box titled "Add Destination Object". It contains the following fields and controls:

- Lookup Name:** A text input field.
- Destination Type:** Two radio buttons, "Queue" (selected) and "Topic".
- Read-Only:** A checkbox.
- Destination Name:** A text input field with the value "Untitled_Destination_Object".
- Destination Description:** A text input field with the value "A Description for the Destination Object".
- Buttons:** "OK", "Reset To Defaults", "Cancel", and "Help".

3. 在 [Lookup Name] 欄位中輸入「MyQueue」。

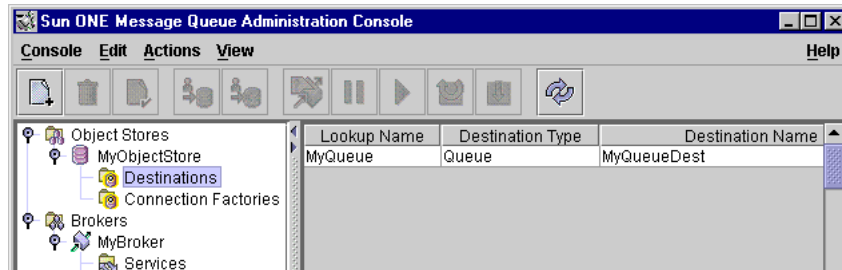
使用此查找名稱可以找到正在使用 JNDI 查找呼叫的物件。在範例應用程式中，此呼叫如下：

```
queue= (javax.jms.Queue) ctx.lookup("MyQueue");
```

4. 為 [Destination Type] 選擇 [Queue] 單選按鈕。
5. 在 [Destination Name] 欄位中輸入 MyQueueDest。
6. 按一下 [OK]。

此名稱爲您在代理程式上新增實體目標時指定的名稱。

- 在導覽窗格中選取 [Destinations]，並注意有關剛新增的佇列目標受管理物件的資訊在結果窗格中的顯示方式。

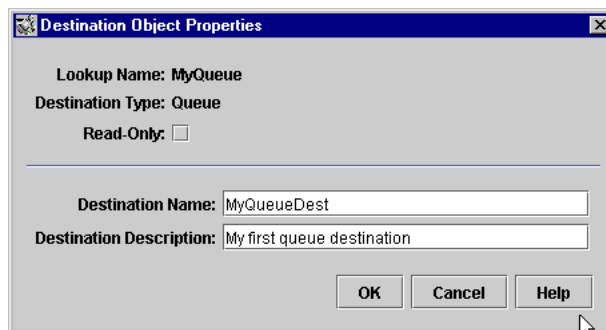


受管理物件屬性

若要檢視或更新受管理物件的屬性，您需要在導覽窗格中選取 [Destinations] 或 [Connection Factories]，在結果窗格中選取特定的物件然後選擇 [Actions] > [Properties]。

- 檢視或更新目標物件的屬性
 - 在導覽窗格中選取 [Destinations]。
 - 在結果窗格中選取 [MyQueue]。
 - 選擇 [Actions] > [Properties] 以檢視 [Destination Object Properties] 對話方塊。

請注意您僅可以變更目標名稱與描述的值。若要變更查找名稱，您必須刪除此物件，然後新增具有所需查找名稱之新的佇列受管理物件。



更新主控台資訊

無論您使用物件儲存區還是代理程式，均可以透過選擇 [View] > [Refresh] 來更新任何元素或元素群組的可視顯示。

執行範例應用程式

提供的範例應用程式 `SimpleJNDIClient` 可用於本指導。它使用您在上述指導中建立的受管理物件與目標：名為 `MyQueueDest` 的佇列目標、名為 `MyQueueConnectionFactory` 的佇列連線工廠受管理物件和名為 `MyQueue` 的佇列受管理物件。

此程式碼建立簡單的佇列傳送者和接收者，並傳送和接收「Hello World」訊息。

► 執行 `SimpleJNDIClient` 應用程式

1. 確定 `JAVA_HOME` 環境變數指向安裝 JDK 的目錄。
2. 使包含 `SimpleJNDIClient` 應用程式的目錄成為目前目錄；例如：

```
cd IMQ_HOME/demo/jms (在 Solaris 上為 /usr/demo/imq/jms)
```

您應發現 `SimpleJNDIClient.class` 檔案已存在。(如果您要變更此應用程式，將需要使用「MQ 開發人員指南快速入門指導」中的用於編譯用戶端應用程式的指令重新編譯它。)

3. 將 `CLASSPATH` 變數設定為包括目前目錄，此目錄包含 `SimpleJNDIClient.java` 和以下 jar 檔案：`jms.jar`、`imq.jar` 與 `fscontext.jar`。
4. 執行此應用程式之前，請開啓來源檔 `SimpleJNDIClient.java` 並閱讀全部來源。

它很簡短但已充分進行說明，並且有關您使用本指導已建立的受管理物件和目標的使用方法亦相當清楚。

5. 執行 SimpleJNDIClient 應用程式。

C:> java SimpleJNDIClient (在 Windows 上)

% java SimpleJNDIClient file:///tmp (在 Solaris 和 Linux 上)

如果此應用程式執行成功，您應看到以下輸出：

```
=====
$JAVA SimpleJNDIClient file:///tmp
Usage: java SimpleJNDIClinet [Context.PROVIDER_URL]

On Unix:
    java SimpleJNDIClient file:///tmp
On Windows:
    java SimpleJNDIClient file:///C:Temp

Using Windows default file:///C/Temp for Context.PROVIDER_URL

Publishing a message to Queue: MyQueueDest
Received the following message: Hello World.
=====
```

啟動與配置代理程式

安裝 Sun™ ONE Message Queue (MQ) 之後，您可以使用 `imqbrokerd` 指令啟動代理程式。代理程式實例的配置由一組配置檔案和使用 `imqbrokerd` 指令傳送的選項管理，這些選項置換配置檔案中的相應屬性。

本章介紹 `imqbrokerd` 指令的語法，以及如何使用指令行選項和配置檔案來配置代理程式實例。此外，還描述如何執行以下作業：

- 編輯代理程式實例配置檔案
- 使用代理程式叢集
- 控制代理程式的記錄

如需如何啟動與使用作為 Windows 服務的代理程式的描述，請參閱第 211 頁的「將代理程式用作 Windows 服務」。

配置檔案

已安裝的配置檔案 (用於配置代理程式) 位於以下目錄。

`IMQ_HOME/lib/props/broker`
(在 Solaris 上為 `/usr/share/lib/imq/props/broker`)

此目錄儲存以下檔案：

- 啟動時載入的預設配置檔案。此檔案名為 `default.properties`，並且不可編輯。您可能需要閱讀此檔案，以確定預設設定並找到您要變更之屬性的確切名稱。
- 包含安裝 MQ 時指定的所有屬性的安裝配置檔案。此檔案名為 `install.properties`；安裝之後無法編輯。

此外，當您首次執行代理程式時，會建立實例配置檔案，您可以用來指定代理程式實例的配置屬性。此檔案由代理程式實例維護以回應管理指令，如果您很仔細也可以直接編輯它。此實例配置檔案儲存在以下位置：

`IMQ_VARHOME/instances/brokerName/props/config.properties`
(在 Solaris 上為
`/var/imq/instances/brokerName/props/config.properties`)

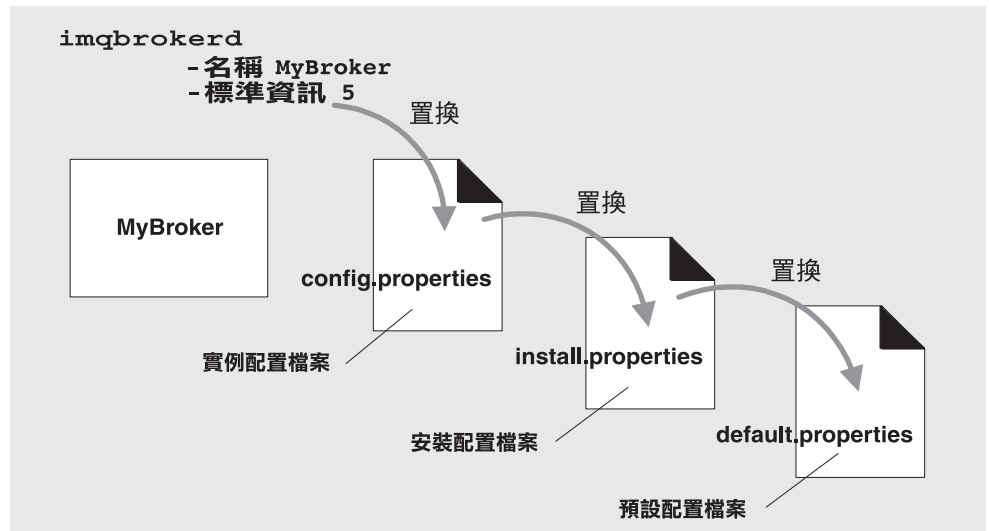
其中代理程式名稱為代理程式實例的名稱 (依預設為 `imqbroker`)。您可以編輯實例配置檔案以變更配置 (請參閱第 104 頁的「編輯實例配置檔案」)。

如果您連線叢集中的代理程式實例 (請參閱第 60 頁的「多重代理程式叢集 (企業版)」)，您可能還需要使用叢集配置檔案以指定叢集配置資訊。如需更多資訊，請參閱第 112 頁的「叢集配置屬性」。

合併屬性值

啟動時，系統會合併不同配置檔案中的屬性值。它使用在安裝和實例配置檔案中設定的值置換在預設配置檔案中指定的值。您可以透過使用 `imqbrokerd` 指令選項置換結果值。此機制在[第 103 頁的圖 5-1](#) 中有說明。

圖 5-1 代理程式配置檔案



屬性命名語法

配置檔案中的任何 MQ 屬性定義均使用以下命名語法：

```
propertyName= 值 [[, 值 1] ...]
```

例如，以下項目為自動建立佇列定義佇列類型：

```
imq.queue.default=single
```

以下項目定義訊息的逾期值：

```
imq.message.expiration.timeout=90
```

[第 104 頁的表格 5-1](#) 以字母順序列出代理程式配置屬性（及其預設值）。

編輯實例配置檔案

首次執行代理程式實例時，會自動建立 `config.properties` 檔案。您可以編輯此實例配置檔案，以自訂相應代理程式實例的運作方式和資源使用。

代理程式實例僅在啟動時讀取 `config.properties` 檔案。若要永久性變更 `config.properties` 檔案，您可以

- 使用管理工具。如需有關可使用 `imqcmd` 設定之屬性的資訊，請參閱第 132 頁的表格 6-5。
- 當代理程式實例關閉時編輯 `config.properties` 檔案；然後重新啟動此實例。(在 Solaris 和 Linux 平台上，僅首次啟動此代理程式實例的使用者有權編輯 `config.properties` 檔案。)

表格 5-1 以字母順序列出代理程式實例配置屬性 (及其預設值)。如需有關每種屬性的涵義和使用的更多資訊，請參閱指定的交叉參照部分。

表格 5-1 代理程式實例配置屬性

屬性名稱	類型	預設值	參考
<code>imq.accesscontrol.enabled</code>	布林	<code>true</code>	第 53 頁的表格 2-6
<code>imq.accesscontrol.file.filename</code>	字串	<code>accesscontrol.properties</code>	第 53 頁的表格 2-6
<code>imq.authentication.basic.user_repository</code>	字串	<code>file</code>	第 53 頁的表格 2-6
<code>imq.authentication.client.response.timeout</code>	整數 (秒)	<code>180</code>	第 53 頁的表格 2-6
<code>imq.authentication.type</code>	字串	<code>digest</code>	第 53 頁的表格 2-6
<code>imq.autocreate.queue</code>	布林	<code>true</code>	第 60 頁的表格 2-9
<code>imq.autocreate.topic</code>	布林	<code>true</code>	第 60 頁的表格 2-9
<code>imq.cluster.url</code>	字串	<code>null</code>	第 63 頁的表格 2-10
<code>imq.keystore.property_name</code>			第 179 頁的表格 8-8
<code>imq.log.console.output</code>	字串	<code>ERROR WARNING</code>	第 56 頁的表格 2-8
<code>imq.log.console.stream</code>	字串	<code>ERR</code>	第 56 頁的表格 2-8

1 作為位元組字串鍵入的值，可以用位元組、千位元組和百萬位元組表示：例如：1000 表示 1000 位元組；7500b 表示 7500 位元組；77k 表示 77 千位元組 (77 x 1024 = 78848 位元組)；17m 表示 17 百萬位元組 (17 x 1024 x 1024 = 17825792 位元組)

表格 5-1 代理程式實例配置屬性 (續)

屬性名稱	類型	預設值	參考
imq.log.file.dirpath	字串	IMQ_VARHOME/ instances/ brokerName/log (在 Solaris 上為 /var/imq/...)	第 56 頁的表格 2-8
imq.log.file.name	字串	log.txt	第 56 頁的表格 2-8
imq.log.file.output	字串	ALL	第 56 頁的表格 2-8
imq.log.file.rolloverbytes	整數 (位元組)	0	第 56 頁的表格 2-8
imq.log.file.rolloversecs	整數 (秒)	604800	第 56 頁的表格 2-8
imq.log.level	字串	INFO	第 56 頁的表格 2-8
imq.log.syslog.facility	字串	LOG_DAEMON	第 56 頁的表格 2-8
imq.log.syslog.logpid	布林	true	第 56 頁的表格 2-8
imq.log.syslog.logconsole	布林	false	第 56 頁的表格 2-8
imq.log.syslog.identity	字串	imqbrokerd_\${imq. 代理程式名稱}	第 56 頁的表格 2-8
imq.log.syslog.output	字串	ERROR	第 56 頁的表格 2-8
imq.message.expiration. interval	整數 (秒)	60	第 47 頁的表格 2-4
imq.message.max_size	位元組字串 ¹ 0 (無限制)	70m	第 47 頁的表格 2-4
imq.metrics.enabled	布林	true	第 56 頁的表格 2-8
imq.metrics.interval	整數 (秒)	0	第 56 頁的表格 2-8
imq.passfile.enabled	布林	false	第 53 頁的表格 2-6
imq.passfile.dirpath	字串	IMQ_HOME/etc (在 Solaris 上為 /etc/imq)	第 53 頁的表格 2-6
imq.passfile.name	字串	passfile	第 53 頁的表格 2-6
imq.persist.file. destination.file.size	位元組字串 ¹	1m	第 50 頁的表格 2-5
imq.persist.file.message. cleanup	布林	false	第 50 頁的表格 2-5

¹ 作為位元組字串鍵入的值，可以用位元組、千位元組和百萬位元組表示：例如：1000 表示 1000 位元組；7500b 表示 7500 位元組；77k 表示 77 千位元組 (77 x 1024 = 78848 位元組)；17m 表示 17 百萬位元組 (17 x 1024 x 1024 = 17825792 位元組)

表格 5-1 代理程式實例配置屬性 (續)

屬性名稱	類型	預設值	參考
<code>imq.persist.file.message.fdpool.limit</code>	整數	25 (Solaris 與 Linux) 1024 (Windows)	第 50 頁的表格 2-5
<code>imq.persist.file.message.filepool.cleanratio</code>	整數	0	第 50 頁的表格 2-5
<code>imq.persist.file.message.filepool.limit</code>	整數	10000	第 50 頁的表格 2-5
<code>imq.persist.file.sync.enabled</code>	布林	false	第 50 頁的表格 2-5
<code>imq.persist.jdbc.property_name</code>			第 187 頁的表格 A-1
<code>imq.persist.store</code>	字串	file	第 50 頁的表格 2-5
<code>imq.portmapper.port</code>	整數	7676	第 44 頁的表格 2-3
<code>imq.queue.deliverypolicy</code>	字串	single	第 60 頁的表格 2-9
<code>imq.redelivered.optimization</code>	布林	true	第 47 頁的表格 2-4
<code>imq.resource_state.threshold</code>	整數 (百分比)	0 (green) 60 (yellow) 75 (orange) 90 (red)	第 47 頁的表格 2-4
<code>imq.service.activelist</code>	清單	jms、admin	第 44 頁的表格 2-3
<code>imq.service_name.accesscontrol.enabled</code>	布林	繼承系統範圍的屬性的值	第 53 頁的表格 2-6
<code>imq.service_name.accesscontrol.file.filename</code>	字串	繼承系統範圍的屬性的值	第 53 頁的表格 2-6
<code>imq.service_name.authentication.type</code>	字串	繼承系統範圍的屬性的值	第 53 頁的表格 2-6
<code>imq.service_name.max_threads</code>	整數	1000 (jms) 500 (ssljms) 500 (httpjms) 500 (httpsjms) 50 (admin)	第 44 頁的表格 2-3

1 作為位元組字串鍵入的值，可以用位元組、千位元組和百萬位元組表示：例如：1000 表示 1000 位元組；7500b 表示 7500 位元組；77k 表示 77 千位元組 (77 x 1024 = 78848 位元組)；17m 表示 17 百萬位元組 (17 x 1024 x 1024 = 17825792 位元組)

表格 5-1 代理程式實例配置屬性 (續)

屬性名稱	類型	預設值	參考
<code>imq.service_name.min_threads</code>	整數	10 (jms) 10 (ssljms) 10 (httpjms) 10 (httpsjms) 4 (admin)	第 44 頁的表格 2-3
<code>imq.service_name.protocol type.hostname</code>	字串	null	第 44 頁的表格 2-3
<code>imq.service_name.protocol type.port</code>	整數	0	第 44 頁的表格 2-3
<code>imq.service_name.threadpool_model</code>	字串	dedicated (jms) dedicated (ssljms) dedicated (httpjms) dedicated (httpsjms) dedicated (admin)	第 44 頁的表格 2-3
<code>imq.shared.connectionMonitor_limit</code>	整數	512 (Solaris 與 Linux) 64 (Windows)	第 44 頁的表格 2-3
<code>imq.system.max_count</code>	整數， 0 (無限制)	0	第 47 頁的表格 2-4
<code>imq.system.max_size</code>	位元組字串 ¹ 0 (無限制)	0	第 47 頁的表格 2-4
<code>imq.transaction.autorollback</code>	布林	false	第 47 頁的表格 2-4
<code>imq.user_repository.ldap.property_name</code>			第 170 頁的表格 8-5

¹ 作為位元組字串鍵入的值，可以用位元組、千位元組和百萬位元組表示：例如：1000 表示 1000 位元組；7500b 表示 7500 位元組；77k 表示 77 千位元組 (77 x 1024 = 78848 位元組)；17m 表示 17 百萬位元組 (17 x 1024 x 1024 = 17825792 位元組)

啟動代理程式

啟動代理程式實例可使用 `imqbrokerd` 指令。

註 您無法使用管理主控台 (`imqadmin`) 或指令公用程式 (`imqcmd`) 啟動代理程式實例。代理程式實例必須已處於執行狀態，才能使用這些 MQ 管理工具。

若要置換一個或多個屬性值，請指定有效的 `imqbrokerd` 指令行選項。指令行選項可置換代理程式配置檔案中的值，但僅用於目前的代理程式階段作業：指令行選項不能寫入至實例配置檔案。

`imqbrokerd` 指令的語法如下 (選項和引數用空格分隔)：

```
imqbrokerd [[ -Dproperty=value] ...]
[ -backup fileName]
[ -cluster "[broker] [[,broker]...]"
[ -dbuser userName] [ -dbpassword password]
[ -force]
[ -h]
[ -javahome path | -jrehome path]
[ -ldappassword password]
[ -license name]
[ -loglevel level]
[ -metrics number]
[ -name brokerName ] [ -port number]
[ -shared]
[ -password keypassword] [ -passfile fileName]
[ -remove instance]
[ -reset data]
[ -restore fileName]
[ -shared]
[ -silent] [ -tty]
[ -version] [ -vmargs arg [[arg]...]
```

例如，若要啟動使用預設代理程式名稱與配置的代理程式，請使用以下指令：

```
imqbrokerd
```

此指令可在本地機器上啟動代理程式的預設實例 (名為 `imqbroker`)，連接埠對映器位於連接埠 7676。

註	在 Solaris 上，您可以將代理程式配置為異常結束後自動重新啟動，方法為透過將 <code>/etc/imq/imqborkerd.conf</code> 配置檔案中的 <code>RESTART</code> 屬性設定為 <code>YES</code> 。
----------	--

註	在 Solaris 和 Linux 平台上，包含配置資訊和永久性資料的目錄的許可權，取決於首次啟動代理程式實例的使用者的身份。因此，若要使代理程式實例運行正常，以後也必須僅由原來的使用者啟動它。
----------	---

表格 5-2 描述了 `imqbrokerd` 指令的選項，還描述了每個選項影響的配置屬性 (如果有)。

表格 5-2 `imqbrokerd` 選項

選項	影響的屬性	描述
<code>-backup fileName</code>	無影響的屬性。	僅適用於代理程式叢集。將主代理程式的配置變更記錄備份至指定檔案。請參閱第 116 頁的「備份主代理程式的配置變更記錄」。
<code>-cluster" [broker] [[,broker] ...]"</code> 代理程式為以下之一 <code>host[:port]</code> <code>[host]:port</code>	將 <code>imq.cluster.brokerlist</code> 設定為與其連線的代理程式清單。	僅適用於代理程式叢集。連線至指定主機和連接埠上的所有代理程式。此清單已與 <code>imq.cluster.brokerlist</code> 屬性中的清單合併。如果您沒有為主機指定值，則使用 <code>localhost</code> 。如果您沒有為連接埠指定值，則使用值 <code>7676</code> 。請參閱第 112 頁的「使用叢集 (企業版)」，以獲得有關如何使用此選項連線多重代理程式的更多資訊。
<code>-dbpassword password</code>	將 <code>imq.persist.jdbc.password</code> 設定為指定的密碼	為外掛 JDBC 相容資料儲存區指定密碼。請參閱附錄 A 「設定外掛持續性」。
<code>-dbuser fileName</code>	將 <code>imq.persist.jdbc.user</code> 設定為指定的使用者名稱	為外掛 JDBC 相容資料庫指定使用者名稱。請參閱附錄 A 「設定外掛持續性」。
<code>-D=value</code>	設定系統屬性。置換實例配置檔案中的相應屬性值。	將指定的屬性設定為指定的值。請參閱第 104 頁的表格 5-1，以獲得代理程式配置屬性。 小心：請仔細檢查使用 <code>D</code> 選項設定的屬性的拼寫與格式。如果您傳送了錯誤的值，系統將不警告您，並且 MQ 亦無法設定這些值。
<code>-force</code>	無影響的屬性。	未經使用者確認即執行動作。此選項僅適用於 <code>-remove instance</code> 選項，此選項通常要求確認。
<code>-h</code>	無影響的屬性。	顯示說明。不執行指令行上的其他任何指令。

表格 5-2 imqbrokerd 選項 (續)

選項	影響的屬性	描述
-javahome <i>path</i>	無影響的屬性。	指定替代 Java 2 相容 JDK 的路徑。預設為使用隨附的運行時間。
-jrehome <i>path</i>	無影響的屬性。	指定 Java 2 JRE 的路徑。
-ldappassword <i>password</i>	將 imq.user_repository.ldap.password 設定為指定的密碼	為存取 LDAP 使用者儲存庫指定密碼。請參閱第 170 頁的「將 LDAP 伺服器用於使用者儲存庫」。
-license [<i>name</i>]	無影響的屬性。	如果要載入的授權與 MQ 產品版本預設的授權不同，則指定要載入的授權。如果您尚未指定授權名稱，則此項會列出系統上安裝的所有授權。依據已安裝的 MQ 版本，名稱的值為 pe (平台版 — 基本功能)、try (平台版 — 90 天試用企業功能) 和 unl (企業版)。請參閱第 24 頁的「產品版本」。
-loglevel <i>level</i>	將 imq.broker.log.level 設定為指定的級別。	將記錄級別指定為以下之一：NONE、ERROR、WARNING 或 INFO。預設值為 INFO。如需更多資訊，請參閱第 54 頁的「記錄程式」。
-metrics <i>int</i>	將 imq.metrics.report.interval 設定為指定的秒數。	指定報告標準資訊的間隔時間 (以秒為單位)。
-name <i>brokerName</i>	將 imq.instancename 設定為指定的名稱。	指定此代理程式的實例名稱，並使用相應的實例配置檔案。如果您沒有指定代理程式名稱，則將此檔案的名稱設定為 imqbroker。 註：如果您在同一主機上執行代理程式的多個實例，則每個實例必須具有唯一的名稱。
-passfile <i>fileName</i>	將 imq.passfile.enabled 設定為 true。 將 jmq.passfile.dirpath 設定為包含此檔案的路徑。 將 imq.passfile.name 設定為此檔案的名稱。	指定檔案的名稱，您可從此檔案讀取 SSL 鍵值儲存區、LDAP 使用者儲存庫或 JDBC 相容資料庫的密碼。如需更多資訊，請參閱第 182 頁的「使用密碼檔案」。
-password <i>keypassword</i>	將 imq.keystore.password 設定為指定的密碼。	為 SSL 證書鍵值儲存區指定密碼。如需更多資訊，請參閱第 51 頁的「安全性管理程式」。
-port <i>number</i>	將 imq.portmapper.port 設定為指定的號碼。	指定代理程式的連接埠對映器連接埠號。依預設，此項設定為 7676。若要在同一伺服器上執行代理程式的兩個實例，則每個代理程式的連接埠對映器必須具有不同的連接埠號。JMS 用戶端使用此連接埠號連線至代理程式實例。

表格 5-2 imqbrokerd 選項 (續)

選項	影響的屬性	描述
-remove instance	無影響的屬性。	移除代理程式實例：刪除此實例配置檔案、日誌當、永久性儲存區以及與此實例相關聯的其他檔案和目錄。要求使用者確認，除非也指定了 -force 選項。
-reset store messages durables props	無影響的屬性。	依據給定的引數，重設資料儲存區 (或資料儲存區的子集) 或代理程式實例的配置屬性。 重設資料儲存區，清除所有永久性資料，包括永久性訊息、持久訂閱和異動資訊。此項可讓您啟動沒有記錄的代理程式實例。您還可以僅清除所有永久性訊息或僅清除所有持久訂閱。(如果您不想在後續的啟動中重設永久性儲存區，請重新啟動未使用 -reset 選項的代理程式實例。) 如需更多資訊，請參閱第 48 頁的「持續性管理程式」。 重設代理程式的屬性，用空白檔案取代現有實例配置檔案 (config.properties)：所有屬性均假設為預設值。
-restore fileName	無影響的屬性。	僅適用於代理程式叢集。用指定的備份檔案取代主代理程式的配置變更記錄。此檔案必須已使用 -backup 選項事先建立。請參閱第 116 頁的「復原主代理程式的配置變更記錄」。
-shared	將 imq.jms.threadpool_model 設定為 shared。	指定要使用共用執行緒儲存區模型實施的 jms 連線服務，在此模型中，執行緒在各連線之間共用以增加代理程式實例支援的連線數量。如需更多資訊，請參閱第 42 頁的「連線服務」。
-silent	將 imq.log.console.output 設定為 NONE。	關閉對主控台的記錄。
-tty	將 imq.log.console.output 設定為 ALL	指定所有訊息均在主控台顯示。依預設，僅顯示 WARNING 和 ERROR 級別的訊息。
-version	無影響的屬性。	顯示安裝產品的版本編號。
-vmargs arg [arg]...	無影響的屬性	指定要傳送至 Java VM 的引數。用空格分隔各引數。如果您要傳送多個引數，或者如果引數包含空格，請使用括起的引號。例如： <pre>imqbrokerd -tty -vmargs "-Xmx128m -Xincgc"</pre>

使用叢集 (企業版)

本節描述用於配置多重代理程式叢集的屬性以及連線代理程式的兩種方法，並介紹如何管理叢集。如需叢集的介紹，請參閱第 60 頁的「[多重代理程式叢集 \(企業版\)](#)」。

當使用叢集時，請確保叢集中所有代理程式的主機時鐘同步。

叢集配置屬性

當您將代理程式連線至叢集時，所有連線的代理程式必須為叢集配置屬性指定相同的值。這些屬性描述了叢集中代理程式的參與情況。[表格 5-3](#) 總結了與叢集相關的配置屬性。

表格 5-3 叢集配置屬性

屬性	描述
<code>imq.cluster.brokerlist</code>	用逗號分隔的清單指定叢集中的所有代理程式；每個項目指定代理程式的主機和連接埠。 例如： <code>host1:3000、host2:8000、ctrhost</code>
<code>imq.cluster.masterbroker</code>	指定主代理程式的主機和連接埠。 為生產環境設定此值。 例如， <code>ctrhost:7676</code>
<code>imq.cluster.url</code>	指定叢集配置檔案的位置。 例如： <code>http://webserver/imq/cluster.properties</code> <code>file:/net/mfsserver/imq/cluster.properties</code>
<code>imq.cluster.port</code>	對於叢集中的 每個 代理程式，均可用於為叢集連線服務指定連接埠號。叢集連線服務用於在叢集的各代理程式之間進行內部通訊。預設：0 (已動態配置連接埠)
<code>imq.cluster.hostname</code>	對於叢集中的 每個 代理程式，如果有多台主機可用 (例如，如果一台電腦中有多個網路介面卡)，則可用於指定叢集連線服務所連結的主機 (主機名稱或 IP 位址)。叢集連線服務用於在叢集的各代理程式之間進行內部通訊。 預設： <code>null</code> (所有可用的主機)

您可以使用以下兩種方法之一設定叢集屬性：

- 您可以在每個代理程式的實例配置檔案中 (或在啟動每個代理程式的指令行中)，設定與叢集相關的配置屬性。例如，若要連線代理程式 A (在 `host1` 上，連接埠為 7676)、代理程式 B (在 `host2` 上，連接埠為 5000) 和代理程式 C (在 `ctrlhost` 上，連接埠為 7676)，則用於代理程式 A、B 和 C 的實例配置檔案需要設定以下屬性。

```
imq.cluster.brokerlist=host1, host2:5000, ctrlhost
```

如果您決定變更叢集配置，則此方法要求更新所有代理程式中與叢集相關的屬性

- 您可以在一個中央叢集配置檔案中設定叢集配置屬性。這些屬性可能包括要連線的代理程式清單 (`imq.cluster.brokerlist`)，以及選擇性地包括主代理程式 (`imq.cluster.masterbroker`) 的位址。

如果您使用此方法，您還必須將 `imq.cluster.url` 屬性 (對於叢集中的每個代理程式) 設定為指向叢集配置檔案的位置。從易於維護的角度看，此方法為叢集配置的建議方法。

以下程式碼範例顯示叢集配置檔案的內容。`host1` 與 `ctrlhost` 均正在預設連接埠上執行。這些屬性指定 `host1` 與 `ctrlhost` 均在叢集中連線，並且 `ctrlhost` 為主代理程式。

```
imq.cluster.brokerlist=host1,host2:5000,ctrlhost
imq.cluster.masterbroker=ctrlhost
```

用於在叢集中連線的每個代理程式的實例配置檔案，必須還包含此叢集配置檔案的 URL；例如：

```
imq.cluster.url=file:/home/cluster.properties
```

連線代理程式

本節描述將代理程式連線至叢集的兩種方法。無論您使用哪種方法，您啟動的每個代理程式每 5 秒鐘將嘗試連線至其他代理程式；一旦此叢集中的其他代理程式啟動，嘗試即可成功。

如果將代理程式連線至叢集，則不必先啟動主代理程式。如果叢集中有代理程式先於主代理程式啟動，則此代理程式將保持暫停狀態，從而拒絕用戶端連線。當主代理程式啟動時，暫停的代理程式的全部功能將自動可用。

方法 1：無叢集配置檔案

► 將代理程式連線至叢集

1. 使用啟動代理程式的 `imqbrokerd` 指令的 `-cluster` 選項，並將代理程式 (要連線的) 的完整清單指定為 `-cluster` 選項的引數。
2. 對於每個要連線至叢集的代理程式，在啟動它時執行此作業。

例如，以下指令啟動新的代理程式，並將其連線至在 `host1` 的預設連接埠上執行的代理程式、在 `host2` 的連接埠 `7677` 上執行的代理程式以及在本地主機的連接埠 `7678` 上執行的代理程式。

```
imqbrokerd -cluster host1,host2:7677,:7678
```

方法 2：使用叢集配置檔案

還可以建立叢集配置檔案，此檔案指定要連線的代理程式清單 (和選擇性地指定主代理程式的位址)。這種定義叢集的方法更適合用於生產系統。請記住，叢集中的每個代理程式均必須將 `imq.cluster.url` 屬性的值設定為指向此叢集配置檔案。

將代理程式新增至叢集

一旦您已設定代理程式叢集，您可能需要新增新的代理程式或重新啟動此叢集中已存在的代理程式。

若要將新的代理程式新增至現有叢集，您可以執行以下作業之一：

如果您未使用叢集配置檔案，則啟動新代理程式時，使用 `-D` 選項在指令行上指定 `imq.cluster.brokerlist` 屬性和 (如有必要) `imq.cluster.masterbroker` 屬性。

► 使用叢集配置檔案時將代理程式新增至叢集

1. 將此新代理程式新增至叢集配置檔案中的 `imq.cluster.brokerlist` 屬性。
2. 向叢集中的所有代理程式發出以下指令。

```
imqcmd reload cls
```

此指令強制所有代理程式重新載入 `imq.cluster.brokerlist` 屬性，並確保此叢集中代理程式的所有永久性資訊均為最新的。

重新啟動叢集中的代理程式

若要重新啟動已是叢集成員的代理程式，您可以執行以下作業之一：

- 如果已使用叢集配置檔案定義了叢集，則使用 `-D` 選項在用於啟動代理程式的指令行上指定 `imq.cluster.url` 屬性。
- 如果未使用叢集配置檔案定義叢集，則啟動新代理程式時，使用 `-D` 選項在指令行上指定 `imq.cluster.brokerlist` 屬性 (如有必要指定 `imq.cluster.masterbroker` 屬性)。如果叢集不包括主代理程式，則啟動新代理程式時，您可以僅使用 `-cluster` 選項指定叢集中的代理程式清單。

從叢集中移除代理程式

從叢集中移除代理程式時，請注意以下內容：

- 如果代理程式 A、B 和 C 均使用以下指令行啟動，則僅重新啟動 A 將不會從叢集中移除 A。

```
imqbrokerd -cluster A,B,C
```

然而，您需要使用以下指令行重新啟動所有其他代理程式：

```
imqbrokerd -cluster B,C
```

然後，您需要啟動代理程式 A 而不指定 `-cluster` 選項。

- 如果已使用叢集配置檔案指定代理程式清單，則您將需要執行以下作業：
 - 從配置檔案中移除提及的代理程式。
 - 為要移除的代理程式變更或移除 `imq.cluster.url` 屬性，以便它不再使用共用屬性。
 - 使用 `imqcmd reload cls` 指令強制所有代理程式重新載入它們的叢集配置，從而重新配置此叢集。

備份主代理程式的配置變更記錄

每個叢集可以具有一個主代理程式，以記錄此叢集之永久性狀態的所有變更：其中包括持久訂閱和管理員建立的實體目標。所有代理程式在啟動期間參考主代理程式，以使有關這些永久性物件的資訊同步。因此，主代理程式發生故障可使整個叢集損壞。所以，透過使用 `imqbrokerd` 指令的 `-backup` 選項，定期備份主代理程式的變更記錄很重要。例如，

```
imqbrokerd -backup mybackuplog
```

適時地進行備份很重要。復原很舊的備份可能會導致資訊遺失：在最新的備份之前建立的所有永久性物件都將遺失。

復原主代理程式的配置變更記錄

► 在發生故障的情況下復原主代理程式

1. 關閉叢集中的所有代理程式。
2. 使用以下指令復原主代理程式的配置變更記錄：

```
imqbrokerd -restore mybackuplog
```

3. 如果您為主代理程式指定了新的名稱或連接埠號，您必須更新叢集配置檔案，以指定此主代理程式是此叢集的一部分並指定其新的名稱 (使用屬性 `imq.cluster.masterbroker`)。
4. 重新啟動所有代理程式。

復原代理程式必將導致某些舊資料重新載入代理程式的配置變更記錄；但是，經常定期備份 (如上節所述) 應可儘量避免此問題。

由於主代理程式記錄永久性物件變更的整個歷程，因此其資料庫可能會在一段時期內迅速增長。備份與復原作業可以有效地壓縮和最適化此資料庫。

記錄

本節描述代理程式的預設記錄配置，並介紹如何變更此配置以便將日誌資訊重新導向至替代輸出通道，以變更捲動條件並報告代理程式標準資訊。如需記錄的介紹，請參閱第 54 頁的「記錄程式」。

預設記錄配置

當您啟動代理程式時，它會自動配置為將日誌輸出儲存至一組捲動的日誌檔中，這些日誌檔位於

```
IMQ_VARHOME/instances/brokerName/log/
(在 Solaris 上為 /var/imq/instances/brokerName/log/)
```

日誌檔為簡單的文字檔案。它們命名如下，從最舊的到最新的：

```
log.txt
log_1.txt
log_2.txt
...
log_9.txt
```

依預設，日誌檔每週捲動一次；系統保留九個備份檔案。

- 若要變更保留日誌檔的目錄，請將屬性 `imq.log.file.dirpath` 設定為所需路徑。
- 若要將日誌檔的根名稱從 `log` 變更為其他名稱，請設定 `imq.log.file.filename` 屬性。

代理程式支援三種日誌種類：ERROR、WARNING 和 INFO (請參閱第 55 頁的表格 2-7)。設定記錄級別可收集此級別以下 (含此級別) 的所有訊息。預設日誌級別為 INFO。這表示記錄 ERROR、WARNING 和 INFO 的訊息。

日誌訊息格式

記錄的訊息包含時間標記、訊息程式碼和訊息本身。資訊容量因您所設定的日誌級別而有所不同。以下為 INFO 訊息的範例。

```
[13/Sep/2000:16:13:36 PDT] B1004 Starting the broker service
using tcp [ 25374,100] with min threads 50 and max threads of 500
```

變更記錄程式配置

第 56 頁的表格 2-8 中描述了所有記錄程式屬性。

► 變更代理程式的記錄程式配置

- 1. 設定日誌級別。
- 2. 為一個或多個記錄種類設定輸出通道 (檔案、主控台或二者)。
- 3. 如果您將輸出記錄至檔案，請為此檔案配置捲動條件。

您可以透過設定記錄程式屬性完成這些步驟。您可以用以下兩種方法之一執行此作業：

- 在啟動代理程式之前，為其變更或新增 `config.properties` 檔案中的記錄程式屬性。
- 在啟動代理程式的 `imqbrokerd` 指令中，指定記錄程式指令行選項。您還可以使用代理程式選項 `-D` 變更記錄程式屬性 (或任何代理程式屬性)。

在指令行上傳送的選項置換在代理程式實例配置檔案中指定的屬性。表格 5-4 列出了影響記錄的 `imqbrokerd` 選項。

表格 5-4 `imqbrokerd` 項及相應的屬性

imqbrokerd 選項	描述
<code>-metrics number</code>	指定收集標準資訊的間隔時間 (以秒為單位)。
<code>-loglevel level</code>	將日誌級別設定為以下之一：ERROR、WARNING 和 INFO。
<code>-silent</code>	關閉對主控台的記錄
<code>-tty</code>	將所有訊息傳送至主控台。依預設，僅顯示 WARNING 和 ERROR 級別的訊息。

以下各節描述如何變更預設配置以便執行以下作業：

- 變更輸出通道 (日誌訊息的目標)
- 變更捲動條件
- 日誌代理程式標準資訊

變更輸出通道

依預設，錯誤和警告訊息既可在終端機上顯示，也可記錄至日誌檔。(在 Solaris 上，錯誤訊息還可寫入至系統的系統日誌常駐程式。)

您可以以下方法變更日誌訊息的輸出通道：

- 若要使所有日誌種類 (對於給定的級別) 輸出顯示在畫面上，請使用 `imqbrokerd` 指令的 `-tty` 選項。
- 若要避免日誌輸出顯示在畫面上，請使用 `imqbrokerd` 指令的 `-silent` 選項。
- 使用 `imq.log.file.output` 屬性指定應寫入至日誌檔的記錄資訊的種類。例如，

```
imq.log.file.output=ERROR
```

- 使用 `imq.log.console.output` 屬性指定應寫入至主控台的記錄資訊的種類。例如，

```
imq.log.console.output=INFO
```

- 在 Solaris 上，使用 `imq.log.syslog.output` 屬性指定應寫入至 Solaris 系統日誌的記錄資訊的種類。例如，

```
imq.log.syslog.output=NONE
```

註

變更日誌訊息的目標之前，您必須確保已將記錄設定為與您要對映至輸出通道的日誌種類相對應的級別。例如，如果您將日誌級別設定為 `ERROR`，並將 `imq.log.console.output` 屬性設定為 `WARNING`，則不會記錄任何訊息，因為您未啟用這些級別訊息的記錄。

變更捲動條件

有兩個條件用於捲動日誌檔：時間和容量。預設為使用時間條件，每七天捲動一次檔案。

- 若要變更時間間隔，您需要變更屬性 `imq.log.file.rolloversecs`。例如，以下屬性定義將時間間隔變更為十天：

```
imq.log.file.rolloversecs=864000
```

- 若要將捲動條件變更為取決於檔案容量，您需要設定 `imq.log.file.rolloverbytes` 屬性。例如，以下定義使代理程式在檔案達到 500,000 位元組的限制時，捲動檔案。

```
imq.log.file.rolloverbytes=500000
```

如果您同時設定了與時間相關的和與容量相關的屬性，則先達到的限制將啟動捲動。如上所述，代理程式最多可保留九個捲動檔案。

記錄代理程式效能標準資訊

代理程式的預設配置包括以下設定：

- `imq.metrics.enabled=true`
- `imq.metrics.interval=0`
- `imq.log.level=INFO`

由於這些設定，代理程式可為代理程式和作用中的連線服務收集效能標準資訊，但是它不會產生標準資訊報告。

您可以以下兩種方法之一讓代理程式產生標準資訊報告：

- 使用 `imqbrokerd` 指令的 `-metrics` 選項，並指定代理程式產生報告的間隔時間 (以秒為單位)。
- 將 `imq.metrics.interval` 屬性設定為您想讓代理程式產生報告的間隔時間 (以秒為單位)。

因為 `INFO` 種類中包括標準資訊報告，所以依預設，標準資訊報告寫入至日誌檔輸出通道。

以下顯示標準資訊範例：

```
[31/Jan/2001:15:00:50 PST]
Connections: 0 JVM Heap: 6291456 bytes (5186320 free)
  In: 0 msgs (0bytes) 0 pkts (0 bytes)
  Out: 0 msgs (0bytes) 0 pkts (0 bytes)
Rate In: 0 msgs/sec (0 bytes/sec) 0 pkts/sec (0 bytes/sec)
Rate Out: 0 msgs/sec (0 bytes/sec) 0 pkts/sec (0 bytes/sec)
```

表格 5-5 描述了為每種連線服務產生的標準資訊的涵義。

表格 5-5 為連線服務收集的標準資訊

標準資訊	描述
封包輸入 (總計)	從上一次重設至今，代理程式讀取的封包總數。此項包括 MQ 協定封包，不僅僅是 JMS 訊息。
封包傳出 (總計)	從上一次重設至今，代理程式寫入的封包總數。
JMS 訊息傳入 (總計)	從上一次重設至今，代理程式讀取的 JMS 訊息總數。
JMS 訊息傳出 (總計)	從上一次重設至今，代理程式寫入的 JMS 訊息總數。
訊息位元組傳出 (總計)	從上一次重設至今，代理程式讀取的訊息位元組總數。
訊息位元組傳出 (總計)	寫入的訊息位元組總數。
目前 # 連線	開啓連線的目前數量。

表格 5-6 描述了為每個代理程式收集和報告的標準資訊。

表格 5-6 為每個代理程式收集的標準資訊

標準資訊	描述
VM 堆疊容量 (位元組)	Java VM 堆疊的最大容量。
VM 堆疊自由空間 (位元組)	Java VM 堆疊中剩餘的自由空間容量。

註 透過 `imgcmd metrics` 指令也可取得此資訊。

代理程式和應用程式管理

本章介紹如何執行與管理代理程式及其提供的服務相關的工作。其中的某些工作獨立於任何特定的用戶端應用程式。這些工作包括：

- 控制代理程式的狀態：您可以暫停、繼續、關閉以及重新啓動代理程式。
- 查詢與更新代理程式屬性
- 查詢與更新連線服務
- 配置與管理資源
- 管理連線服務

其他代理程式工作作為特定應用程式執行；這些工作包括管理實體目標、持久訂閱以及異動：

- MQ 訊息經由代理程式目標路由至它們的接收者或訂閱者。您負責在代理程式上建立這些目標。
- MQ 為持久訂閱者配置並維護資源，即使擁有持久訂閱的用戶端成為非作用中的用戶端。使用 MQ 指令工具可取得有關持久訂閱的資訊，並可銷毀持久訂閱或清除它們的訊息以便儲存 MQ 資源。
- MQ 異動與分散式異動由代理程式追蹤。如果發生故障，您可能需要手動確定或回轉異動。

本章介紹如何使用指令公用程式 (imqcmd) 以執行所有這些工作。使用管理主控台 (即 MQ 訊息伺服器的圖形介面)，您也可以完成這些相同工作中的許多項。如需更多資訊，請參閱第 4 章「**管理主控台指導**」。

指令公用程式

指令公用程式可讓您管理代理程式及其提供的服務。本節描述基本的 `imqcmd` 指令語法、提供子指令的清單，並總結 `imqcmd` 選項。後續各節介紹如何使用這些指令完成特定的工作。

指令的語法

`imqcmd` 指令的一般語法如下：

```
imqcmd subcommand argument [options]
imqcmd -h|H
imqcmd -v
```

請注意，如果您指定 `-v`、`-h` 或 `-H` 選項，則系統不會執行指令行上指定的子指令。例如，如果您輸入以下指令，則系統會顯示版本資訊但不會執行 `restart` 子指令。

```
imqcmd restart bkr -v
```

imqcmd 子指令

指令公用程式 (`imqcmd`) 包括表格 6-1 中列出的子指令：

表格 6-1 `imqcmd` 子指令

子指令與引數	描述
<code>create dst</code>	建立目標。
<code>commit txn</code>	確定異動。
<code>destroy dst</code>	銷毀目標。
<code>destroy dur</code>	銷毀持久訂閱。
<code>list dst</code>	列出代理程式上的目標。
<code>list dur</code>	列出有關主題的持久訂閱。
<code>list svc</code>	列出代理程式上的服務。
<code>list txn</code>	列出代理程式上的異動。
<code>metrics bkr</code>	顯示代理程式標準資訊。

表格 6-1 imqcmd 子指令 (續)

子指令與引數	描述
metrics svc	顯示服務標準資訊。
pause bkr	暫停代理程式上的所有服務。
pause svc	暫停服務。
purge dst	清除目標上的所有訊息而不銷毀目標。
purge dur	清除持久訂閱上的所有訊息而不銷毀持久訂閱。
query bkr	查詢並顯示有關代理程式的資訊。
query dst	查詢並顯示有關目標的資訊。
query svc	查詢並顯示有關服務的資訊。
query txn	查詢並顯示有關異動的資訊。
reload cls	重新載入代理程式叢集配置。
restart bkr	重新啟動目前執行的代理程式實例。不能用於啟動新的代理程式實例。
resume bkr	繼續代理程式上的所有服務。
resume svc	繼續服務。
rollback txn	回轉異動。
shutdown bkr	關閉代理程式實例。隨後可使用 imqbrokerd 指令啟動，而不是 imqcmd 的 restart bkr 子指令。
update bkr	更新代理程式的屬性。
update dst	更新目標的屬性。
update svc	更新服務的屬性。

imqcmd 選項摘要

表格 6-2 列出了 imqcmd 指令的選項。如需它們用法的說明，請參閱以下基於工作的各節。

表格 6-2 imqcmd 選項

選項	描述
-b <i>hostName:port</i>	指定代理程式主機名稱及其連接埠號。預設值為 localhost:7676。 僅指定連接埠：-b :7878 僅指定名稱：-b somehost
-c <i>clientID</i>	將持久訂閱者的 ID 指定至主題。如需更多資訊，請參閱第 140 頁的「管理持久訂閱」。
-d <i>topicName</i>	指定主題名稱。與 list dur 和 destroy dur 子指令配合使用。請參閱第 140 頁的「管理持久訂閱」。
-f	未經使用者確認即執行動作。
-h	顯示用法說明。不執行指令行上的其他任何指令。
-H	顯示用法說明、屬性清單和範例。不執行指令行上的其他任何指令。
-int <i>interval</i>	指定 imqcmd 顯示代理程式標準資訊的間隔時間 (以秒為單位)。(與 metrics 子指令配合使用。)
-javahome	指定要使用的替代 Java 2 相容運行時間 (預設為使用產品隨附的運行時間)。
-m <i>metricType</i>	指定要顯示的標準資訊的類型。可能是以下類型之一 ttl 代理程式傳入和傳出的訊息總數 (預設)。 rts 提供與 ttl 相同的資訊，但指定每秒的訊息數量。 cxn 連線、虛擬記憶體堆疊、執行緒 將此選項與 metrics bkr 或 metrics svc 子指令配合使用。以下指令顯示每隔五秒的預設代理程式的 cxn 類型標準資訊。 imqcmd metrics bkr -m cxn -int 5
-n <i>argumentName</i>	指定子指令引數的名稱。此選項可能為服務名稱、實體目標名稱、持久訂閱名稱或異動 ID 名稱，這取決於子指令。

表格 6-2 imqcmd 選項 (續)

選項	描述
-o <i>attribute=value</i>	指定屬性的值。此選項可以為代理程式 (請參閱第 130 頁的「查詢與更新代理程式屬性」) 的屬性、服務 (請參閱第 133 頁的「管理連線服務」) 的屬性或目標 (請參閱第 136 頁的「管理目標」) 的屬性，這取決於子指令引數。
-secure	使用 ssladmin 連線服務指定與代理程式的安全管理連線 (請參閱第 181 頁的「指令公用程式 (imqcmd)」中的「步驟 4. 配置並執行基於 SSL 的用戶端」)。
-p <i>password</i>	指定您的 (管理員的) 密碼。如果您遺漏此值，系統將提示您提供此密碼。
-s	靜謐模式。不顯示輸出。
-t <i>destinationType</i>	指定目標的類型：t (主題) 或 q (佇列)。
-tmp	顯示暫存目標。
-u <i>name</i>	指定您的 (管理員的) 名稱。如果您遺漏此值，系統將提示您提供此名稱。
-v	顯示版本資訊。不執行指令行上的其他任何指令。

每次發出 imqcmd 子指令時，您必須為主機名稱和連接埠號 (-b)、使用者名稱 (-u)、密碼 (-p) 和安全連線 (-secure) 指定選項。如果未指定主機名稱和連接埠號，則使用預設值。如果未指定使用者名稱和密碼資訊，則系統將提示您提供名稱和密碼。如果未指定 -secure，則連線可能會不安全。

要使用 imqcmd 的必要條件

為了使用 imqcmd 指令以管理代理程式，您必須執行以下作業：

- 使用 imqbrokerd 指令啟動代理程式。
請參閱第 108 頁的「啟動代理程式」。您僅可以使用指令公用程式管理執行中的代理程式；您不能使用它啟動代理程式。
- 使用 -b 選項指定目標代理程式，除非代理程式正在使用連接埠 7676 的本地主機上執行。
- 指定適當的管理員使用者名稱和密碼。如果您未指定，系統將提示您提供名稱和密碼。而且，請注意您使用 imqcmd 執行的每項作業都將由使用者儲存庫認證。

當您安裝 MQ 時，同時也安裝了預設文字檔案使用者儲存庫。檔案名為 `IMQ_HOME/etc/passwd` (在 Solaris 上為 `/etc/imq/passwd`)。此儲存庫出廠時隨附兩個項目：一個用於擁有管理員權限的使用者，一個用於客人使用者。這些項目可讓您連線至代理程式而不執行任何其他工作。例如，如果您僅要測試 MQ，則可以使用您的預設使用者名稱和密碼 (`admin/admin`) 執行公用程式。

如果您要設定生產系統，則將需要執行一些其他工作以認證和授權使用者。您還可以選擇使用您的使用者儲存庫的現有 LDAP 目錄伺服器。如需更多資訊，請參閱第 164 頁的「認證使用者」。

範例

以下指令列出在使用連接埠 7676 的 localhost 上執行的代理程式的屬性：

```
imqcmd query bkr -u admin -p admin
```

以下指令列出使用連接埠 1564 的 myserver 上執行的代理程式的屬性；使用者名稱為 alladin，使用者密碼為 abracadabra。

```
imqcmd query bkr -b myserver:1564 -u alladin -p abracadabra
```

假設使用者名稱 alladin 已指定給 admin 群組，您將作為管理用戶端連線至指定的代理程式。

控制代理程式的狀態

啟動代理程式之後，您可以使用以下 `imqcmd` 子指令控制代理程式的狀態。

- 暫停代理程式

暫停代理程式會暫停代理程式服務執行緒，這將導致代理程式停止偵聽連接埠。然後，您可以執行管理代理程式的訊息流量所需的任何管理工作。例如，如果某特定目標受到訊息的攻擊，則您可以暫停代理程式，並執行以下可能幫助您解決問題的任何動作：追蹤訊息來源、限制目標大小或銷毀目標。

以下指令暫停在使用連接埠 1588 的 myhost 上執行的代理程式。

```
imqcmd pause bkr -b myhost:1588
```

- 繼續代理程式

繼續代理程式會重新啟動代理程式的服務執行緒，並且此代理程式會繼續偵聽連接埠。以下指令繼續在使用連接埠 7676 的 localhost 上執行的代理程式。

```
imqcmd resume bkr
```

- 關閉代理程式
關閉代理程式會終止代理程式程序。這是適當的終止：代理程式停止接受新的連線和訊息，它將完成現有訊息的遞送並終止代理程式程序。以下指令關閉在使用連接埠 1572 的 ctrlsrv 上執行的代理程式

```
imgcmd shutdown bkr -b ctrlsrv:1572
```
- 重新啟動代理程式
關閉並重新啟動代理程式。以下指令重新啟動在使用連接埠 7676 的 localhost 上執行的代理程式：

```
imgcmd restart bkr
```

表格 6-3 總結了用於控制代理程式的 `imgcmd` 子指令。請記住，您必須指定代理程式主機名稱和連接埠號，除非您的目標為在使用連接埠 7676 的 localhost 上執行的代理程式。

表格 6-3 用於控制代理程式的 `imgcmd` 子指令

子指令	描述
<code>pause bkr [-b hostName:port]</code>	暫停預設代理程式或指定主機和連接埠上的代理程式。
<code>resume bkr [-b hostName:port]</code>	繼續預設代理程式或指定主機和連接埠上的代理程式。
<code>shutdown bkr [-b hostName:port]</code>	關閉預設代理程式或指定主機和連接埠上的代理程式。
<code>restart bkr [-b hostName:port]</code>	關閉並重新啟動預設代理程式或指定主機和連接埠上的代理程式。 請注意，此指令使用首次啟動代理程式時指定的選項重新啟動此代理程式。如果您要使用不同的選項，則必須關閉代理程式然後再次啟動它，從而指定您需要的選項。

查詢與更新代理程式屬性

指令公用程式包括子指令，您可以使用這些子指令取得有關代理程式的資訊及更新代理程式屬性。[表格 6-4](#) 列出了這些子指令。

表格 6-4 用於取得資訊及更新代理程式的 `imqcmd` 子指令

子指令語法	描述
<code>query bkr -b hostName:port</code>	列出預設代理程式或指定主機和連接埠上的代理程式屬性的目前設定。還顯示連線至指定代理程式的執行中的代理程式 (在多重代理程式叢集中) 之清單。
<code>reload cls</code>	僅適用於代理程式叢集。強制叢集中的所有代理程式重新載入 <code>imq.cluster.brokerlist</code> 屬性並更新叢集資訊。請參閱第 114 頁的「將代理程式新增至叢集」，以獲得更多資訊。
<code>update bkr [-b hostName:port] -o attribute=value [-o attribute=value] . . .</code>	變更預設代理程式或指定主機和連接埠上的代理程式的指定屬性。
<code>metrics bkr [-b hostName:port] [-m metricType] [-int interval]</code>	顯示預設代理程式或指定主機和連接埠上的代理程式的代理程式標準資訊。 使用 <code>-m</code> 選項以指定要顯示的標準資訊的類型： <code>ttl</code> 代理程式傳入和傳出的訊息總數 (預設)。 <code>rts</code> 提供與 <code>ttl</code> 相同的資訊，但指定每秒的訊息數量。 <code>cxn</code> 連線、虛擬記憶體堆疊、執行緒 使用 <code>-int</code> 選項以指定顯示標準資訊的間隔時間 (以秒為單位)。預設為 5 秒鐘

請記住，當使用[表格 6-4](#) 中列出的任何子指令時，您必須指定代理程式主機名稱和連接埠號，除非您的目標為在使用連接埠 7676 的 `localhost` 上執行的代理程式。

查詢代理程式

若要查詢並顯示有關單一代理程式的資訊，請使用查詢子指令。例如，

```
imgcmd query bkr
```

此指令的輸出如下：

```
%imgcmd query bkr

Querying the broker specified by:

-----
Host          Primary Port
-----
localhost     7676

Auto Create Queues                true
Auto Create Topics                true
Auto Created Queue Delivery Policy Round Robin
Cluster Broker List (active)      myhost/192.18.116.221:7676
Cluster Broker List (configured)
Cluster Master Broker
Cluster URL
Current Number of Messages in System 0
Current Size of Messages in System 0
Instance Name                      imgbroker
Log Level                          INFO
Log Rollover Interval (seconds)     604800
Log Rollover Size (bytes)           0 (unlimited)
Max Message Size                    70m
Max Number of Messages in System    0 (unlimited)
Max Size of Messages in System      0 (unlimited)
Primary Port                        7676
Version                             3.0

Successfully queried the broker.
```

更新代理程式

您可以使用 `update` 子指令更新**表格 6-5** 中列出的任何代理程式屬性。請注意，對代理程式的更新會自動寫入代理程式的實例配置檔案。

表格 6-5 代理程式屬性

屬性	描述
<code>imq.autocreate.queue</code>	指定是否允許代理程式自動建立佇列目標。預設為 <code>True</code> 。
<code>imq.autocreate.topic</code>	指定是否允許代理程式自動建立主題目標。預設為 <code>True</code> 。
<code>imq.queue.deliverypolicy</code>	指定自動建立的佇列之預設遞送策略。值為： <code>s</code> (單一)、 <code>r</code> (循環) 或 <code>f</code> (故障轉移)。預設為 <code>s</code> 。
<code>imq.cluster.url</code>	指定叢集配置檔案的位置。如需更多資訊，請參閱 第 112 頁 的「叢集配置屬性」。
<code>imq.log.level</code>	指定日誌級別為以下之一： <code>NONE</code> 、 <code>ERROR</code> 、 <code>WARNING</code> 和 <code>INFO</code> 。預設為 <code>INFO</code> 。
<code>imq.log.file.rolloversecs</code>	日誌檔被捲動之前的存在時間 (以秒為單位)。值 <code>0</code> 表示沒有基於檔案存在時間的捲動。預設為 <code>604800</code> (7 天)。
<code>imq.log.file.rolloverbytes</code>	指定日誌檔被捲動之前的最大大小。值 <code>0</code> 表示沒有基於檔案大小的捲動。預設為 <code>0</code> 。
<code>imq.message.max_size</code>	指定最大訊息大小 (以位元組為單位)。預設為 <code>70m</code> 。
<code>imq.system.max_count</code>	指定記憶體與磁碟中的最大訊息數量。值 <code>0</code> 表示沒有限制。預設為 <code>0</code> 。
<code>imq.system.max_size</code>	指定記憶體與磁碟中的最大訊息總量。值 <code>0</code> 表示沒有限制。預設為 <code>0</code> 。
<code>imq.portmapper.port</code>	指定連接埠對映器連接埠的數量。預設為 <code>7676</code> 。

例如，以下指令將佇列的預設遞送策略從單一變更至循環。

```
imqcmd update bkr -o "imq.queue.deliverypolicy=r"
```

管理連線服務

指令公用程式包括一些可讓您執行以下作業的子指令

- 列出可用的連線服務
- 顯示有關特定服務的資訊
- 更新服務的屬性
- 暫停與繼續服務

如需 MQ 連線服務的簡介，請參閱第 42 頁的「連線服務」。

表格 6-6 列出控制連線服務的 `imqcmd` 子指令。如果未指定主機名稱或連接埠，則假設它們為 `localhost` 和 `7676`。

表格 6-6 用於管理連線服務的 `imqcmd` 子指令

子指令語法	描述
<code>list svc [-b hostName:port]</code>	列出預設代理程式或指定主機和連接埠上的代理程式上的所有連線服務。
<code>metrics svc -n serviceName</code> <code>[-b hostName:port]</code> <code>[-m metricType] [-int interval]</code>	列出預設代理程式或指定主機和連接埠上的代理程式上的指定服務之標準資訊。 使用 <code>-m</code> 選項以指定要顯示的標準資訊的類型： <code>ttl</code> 代理程式傳入和傳出的訊息總數 (預設)。 <code>rts</code> 提供與 <code>ttl</code> 相同的資訊，但指定每秒的訊息數量。 <code>cxn</code> 連線、虛擬記憶體堆疊、執行緒 使用 <code>-int</code> 選項以指定顯示標準資訊的間隔時間 (以秒為單位)。預設為 5 秒鐘。
<code>query svc -n serviceName</code> <code>[-b hostName:port]</code>	顯示有關在預設代理程式或指定主機和連接埠上的代理程式上執行的指定服務的資訊。
<code>pause svc -n serviceName</code> <code>[-b hostName:port]</code>	暫停預設代理程式或指定主機和連接埠上的代理程式上執行的指定服務。您無法暫停管理服務。
<code>resume svc -n serviceName</code> <code>[-b hostName:port]</code>	繼續在預設代理程式或指定主機和連接埠上的代理程式上執行的指定服務。
<code>update svc -n serviceName</code> <code>[-b hostName:port]</code> <code>-o attribute=value</code> <code>[-o attribute=value 1] ...</code>	更新在預設代理程式或指定主機和連接埠上的代理程式上執行的指定服務的指定屬性。如需服務屬性的描述，請參閱第 136 頁的表格 6-8。

代理程式支援與應用程式用戶端和管理用戶端的通訊。目前，MQ 代理程式可提供
的連線服務如第 134 頁的表格 6-7 中所示。[服務名稱] 欄中的值是指用於為 -n 選項
指定服務名稱的值。(如表格中所示，每種服務均由它使用的服務類型和基本傳輸層
指定，所使用的服務類型為 NORMAL (JMS) 或 ADMIN。)

表格 6-7 代理程式支援的連線服務

服務名稱	服務類型	協定類型
jms	NORMAL (JMS 訊息遞送)	tcp
ssljms (企業版)	NORMAL (JMS 訊息遞送)	tls (基於 SSL 的安全性)
httpjms (企業版)	NORMAL (JMS 訊息遞送)	http
httpsjms (企業版)	NORMAL (JMS 訊息遞送)	https (基於 SSL 的安全性)
admin	ADMIN	tcp
ssladmin (企業版)	ADMIN	tls (基於 SSL 的安全性)

列出連線服務

若要列出代理程式上可用的連線服務，請使用以下指令：

```
imqcmd list svc [-b hostName:portNumber]
```

例如，以下指令列出在使用連接埠 6565 的主機 MyServer 上執行的代理程式上的可用服務。

```
imqcmd list svc -b MyServer:6565
```

以下指令列出在使用連接埠 7676 的 localhost 上執行的代理程式上的所有服務：

```
imqcmd list svc
```

此指令將輸出以下資訊：

Listing all the services on the broker specified by:

Host	Primary Port
localhost	7676

Service Name	Port Number	Service State
admin	33984 (dynamic)	RUNNING
httpjms		UNKNOWN
httpsjms	-	UNKNOWN
jms	33983 (dynamic)	RUNNING
ssladmin	dynamic	UNKNOWN
ssljms	dynamic	UNKNOWN

Successfully listed services.

查詢與更新服務屬性

若要查詢並顯示有關單一服務的資訊，請使用查詢子指令。例如，

```
imqcmd query svc -n jms
```

此指令的輸出如下：

Querying the service where:

Service Name
jms

On the broker specified by:

Host	Primary Port
localhost	7676

Current Number of Allocated Threads	120
Current Number of Connections	20
Max Number of Threads	1000
Min Number of Threads	50
Port Number	42019 (dynamic)
Service Name	jms
Service State	RUNNING

Successfully queried the service.

您可以使用 `update` 子指令變更表格 6-8 中列出的一個或多個服務屬性的值。

表格 6-8 連線服務屬性

屬性	描述
port	指定給要更新的服務之連接埠 (不適用於 <code>httpjms</code> 或 <code>httpsjms</code>) 。
minThreads	指定給服務的執行緒之最小數量。
maxThreads	指定給服務的最大執行緒數量。

以下指令將指定給 `jms` 服務之執行緒的最小數量變更至 20 。

```
imqcmd update svc -n jms -o "minThreads=20"
```

暫停與繼續服務

若要暫停管理服務 (無法暫停) 之外的任何服務，請使用以下指令：

```
imqcmd pause svc -n serviceName
```

若要繼續服務，請使用以下指令：

```
imqcmd resume svc -n serviceName
```

管理目標

所有 MQ 訊息均可經由在特定代理程式上建立的目標、佇列和主題路由至它們的使用者用戶端。您負責在代理程式上管理這些目標。這包括使用指令公用程式建立和銷毀目標、列出目標、顯示有關目標的資訊以及清除訊息。如需目標的介紹，請參閱第 58 頁的「實體目標」。

表格 6-9 提供了 `imqcmd` 目標子指令的摘要。如果代理程式不是預設 (`localhost:7676`) 代理程式，請記住指定它的主機名稱和連接埠。

表格 6-9 用於管理目標的 `imgcmd` 子指令

子指令	描述
<code>list dst [-tmp]</code>	列出所有目標以及暫存目標 (請參閱第 60 頁的「 暫存目標 」) 的選項。
<code>create dst -t type</code> <code> -n destName</code> <code> [-o attribute=value]</code> <code> [-o attribute=value1] ...</code>	建立指定類型、指定名稱以及指定屬性的目標。目標名稱必須僅包含字母數字字元 (無空格)，並且以字母字元或字元「 <code>_</code> 」開頭
<code>destroy dst -t type</code> <code> -n destName</code>	銷毀指定類型和名稱的目標。
<code>purge dst -t type</code> <code> -n destName</code>	清除具有指定類型和名稱的目標上的訊息。
<code>query dst -t type</code> <code> -n destName</code>	列出有關指定類型和名稱之目標的資訊。
<code>update dst -t type</code> <code> -n destName</code> <code> -o attribute=value</code> <code> [-o attribute=value1] ...</code>	更新指定目標上的指定屬性的值。 屬性名稱可以為 表格 6-10 中所述的任何屬性。

建立目標

建立目標時，您必須指定它的類型 (主題或佇列)，並指定目標屬性的值 (如果需要)。這些屬性的預設值在代理程式的配置檔案 (請參閱第 102 頁的「[配置檔案](#)」) 中設定。

銷毀目標會清除此目標上的所有資訊，並從代理程式中移除此目標；此作業不可逆。

表格 6-10 描述了可在建立目標時為每種類型的目標設定的屬性。

表格 6-10 目標屬性

目標類型	屬性	預設值	描述
佇列	queueDeliveryPolicy	單一	描述用於路由訊息的演算法。 值為 f = 故障轉移 r = 循環 s = 單一
佇列	maxTotalMsgBytes	0 (無限制)	佇列中允許的最大訊息總量 (以位元組為單位)。
佇列	maxNumMsgs	0 (無限制)	佇列中允許的最大訊息數量
佇列	maxBytesPerMsg	0 (無限制)	佇列中允許的任何單一訊息的 最大大小。
主題	maxBytesPerMsg	0 (無限制)	發佈至主題的任何單一訊息的 最大大小。

- 若要建立佇列目標，請輸入以下指令：

```
imqcmd create dst -n myQueue -t q -o "queueDeliveryPolicy=f"
```

請注意，目標名稱必須為有效的 Java 識別碼。
- 若要建立主題目標，請輸入以下指令：

```
imqcmd create dst -n myTopic -t t -o "maxBytesPerMsg=5000"
```

取得有關目標的資訊

若要取得有關目前目標屬性值的資訊，請使用以下指令：

```
imqcmd query dst -t q -n XQueue
```

然後，您可以使用 `update imqcmd` 子指令變更一個或多個屬性的值。

若要列出特定代理程式 (如在使用連接埠 4545 的 `myHost` 上執行的代理程式) 上的所有目標，請使用以下指令：

```
imqcmd list dst -b myHost:4545
```

`list` 指令可選擇性地包括暫存目標 (使用 `-tmp` 選項)。這些是由用戶端應用程式建立的目標，用戶端應用程式需要可在其上接收傳送至其他用戶端的訊息的回覆之目標 (請參閱第 60 頁的「暫存目標」)。您無法銷毀這些目標；當不再有作用中的訊息使用者時，只有用戶端應用程式執行的 API 呼叫才能銷毀這些目標。

更新目標

透過使用 `update dst` 子指令 and `-o` 選項指定要更新的屬性，您可以變更目標的屬性。如果要更新多個屬性，您可以多次使用 `-o` 選項。例如，以下指令將 `maxBytesPerMsg` 屬性變更至 1000，並將 `MaxNumMsgs` 屬性變更至 2000：

```
imqcmd update dst -t q -n myQueue -o "maxBytesPerMsg=1000"
-o maxNumMsgs=2000
```

請參閱第 138 頁的表格 6-10，以獲得您可以更新的屬性清單。

您無法使用 `update dst` 子指令更新目標的類型，或更新佇列的佇列遞送策略。

清除目標

您可以清除目前在目標上形成佇列的所有訊息。清除目標意味著刪除在實體目標上形成佇列的所有訊息。當目標上累積的訊息佔用太多的系統資源時，您可能需要清除訊息。當佇列沒有任何註冊的使用者用戶端並要接收許多訊息時，可能需要清除訊息。如果非作用中的主題持久訂閱者未成為作用中的訂閱者，也可能需要清除訊息。在這兩種情況下，訊息均會被不必要地保留。

若要清除目標上的訊息，請輸入以下指令：

```
imqcmd purge dst -n myQueue -t q
imqcmd purge dst -n myTopic -t t
```

如果您已關閉代理程式，並且不想在重新啟動它時遞送舊的訊息，請使用 `imqbrokerd` 指令的 `reset` 子指令清除舊的訊息；例如：

```
imqbrokerd -reset messages
```

這可避免重新啟動代理程式後清除目標的麻煩。

銷毀目標

若要銷毀目標，請輸入以下指令：

```
imqcmd destroy dst -t q -n myQueue
```

管理持久訂閱

您可能需要使用 `imqcmd` 子指令管理代理程式的持久訂閱。持久訂閱是指由用戶端註冊為持久的主題訂閱；它具有唯一的身份，並要求代理程式保留此訂閱的訊息 (即使它的使用者成為非作用中的使用者)。通常，代理程式僅在為持久訂閱者保留的訊息過期時，才刪除該訊息。

表格 6-12 提供了 `imqcmd` 持久訂閱子指令的摘要。如果代理程式不是預設 (`localhost:7676`) 代理程式，請記住指定它的主機名稱和連接埠。

表格 6-11 用於管理持久訂閱的 `imqcmd` 子指令

子指令	描述
<code>list dur -d destination</code>	列出指定目標的所有持久訂閱。
<code>destroy dur -n subscrName -c client_id</code>	銷毀指定用戶端識別碼 (請參閱第 34 頁的「用戶端識別碼」) 的指定持久訂閱。
<code>purge dur -n subscrName -c client_id</code>	清除指定用戶端識別碼 (請參閱第 34 頁的「用戶端識別碼」) 的所有訊息。

例如，以下指令列出主題 `SPQuotes` 的所有持久訂閱

```
imqcmd list dur -d SPQuotes
```

對於主題的每個持久訂閱，`list` 子指令會傳回持久訂閱的名稱、使用者的用戶端 ID、在此主題上形成佇列的訊息數量以及持久訂閱的狀態 (作用中/非作用中)。例如：

```
Listing all the durable subscriptions on the topic myTopic
on the broker specified by:
-----
Host           Primary Port
-----
localhost      7676

Name           Client ID           Number of      Durable Sub
                Messages         State
-----
myDurable      myClientID           1              INACTIVE

Successfully listed durable subscriptions.
```

您可以使用從 `list` 指令傳回的資訊識別您可能要銷毀或要為其清除訊息的持久訂閱。使用訂閱名稱和用戶端 ID 識別此訂閱。例如：

```
imqcmd destroy dur -n myDurable -c myClientID
```

管理異動

由用戶端應用程式啟動的所有異動可由代理程式追蹤。這些異動可以是簡單的 MQ 異動或由 XA 資源管理員管理的分散式異動 (請參閱第 35 頁的「本地異動」)。所有異動都有 MQ 異動 ID，即在代理程式上唯一識別異動的 64 位元的號碼。分散式異動還有由分散式異動管理員指定的分散式異動 ID (XID)，此 ID 長達 128 位元組。MQ 可維護 MQ 異動 ID 與 XID 的關聯。

如果發生故障，分散式異動很可能未經確定便處於 PREPARED 狀態。因此，作為管理員，您可能需要監視然後回轉或確定處於準備狀態的異動。

表格 6-12 提供了 `imqcmd` 異動子指令的摘要。如果代理程式不是預設 (`localhost:7676`) 代理程式，請記住指定它的主機名稱和連接埠。

表格 6-12 用於管理異動的 `imqcmd` 子指令

子指令	描述
<code>list txn</code>	列出要由代理程式追蹤的所有異動。
<code>query txn -n transaction_id</code>	列出有關指定異動的資訊。
<code>commit txn -n transaction_id</code>	確定指定的異動。
<code>rollback txn -n transaction_id</code>	回轉指定的異動。

例如，以下指令列出代理程式中的所有異動。

```
imqcmd list txn
```

對於每個異動，list 子指令會傳回異動 ID、狀態、使用者名稱、訊息或回應的數量以及建立時間。例如：

```
Listing all the transactions on the broker specified by:
-----
Host          Primary Port
-----
localhost     7676

-----
Transaction ID  State      User name  # Msgs/  Creation time
                # Acks
-----
64248349708800 PREPARED   guest     4/0      1/30/02 10:08:31 AM
64248371287808 PREPARED   guest     0/4      1/30/02 10:09:55 AM

Successfully listed transactions.
```

此指令顯示代理程式中的所有本機異動和分散式異動。您僅能確定或回轉處於 PREPARED 狀態的異動。如果您知道由於故障使異動處於此狀態，並且此異動未處於由分散式異動管理員確定的過程中，您應確定或回轉此異動。

例如，如果代理程式的自動回轉屬性設定為錯誤 (請參閱第 47 頁的表格 2-4)，則您必須在啟動代理程式時手動確定或回轉處於 PREPARED 狀態的異動。

list 子指令還顯示在異動中產生的訊息數量，以及在異動 (#Msgs/#Acks) 中回應的訊息數量。這些訊息不會被遞送，並且回應也不會被處理，直至異動被確定。

query 子指令可讓您查閱相同的資訊以及一些其他值：用戶端 ID、連線識別以及分散式異動 ID (XID)。例如，

```
imqcmd query txn -n 64248349708800
```

產生以下輸出：

```

Querying the transaction where:
-----
Transaction ID
-----
64248349708800

On the broker specified by:

-----
Host          Primary Port
-----
localhost     7676

Client ID
Connection          guest@192.18.116.219:62209->jms:62195
Creation time        1/30/02 10:08:31 AM
Number of acknowledgements 0
Number of messages   4
State                PREPARED
Transaction ID        64248349708800
User name            guest
XID
6469706F6C7369646577696E6465723130313234313431313030373230

Successfully queried the transaction.

```

commit 和 rollback 子指令可用於確定或回轉分散式異動。如前面所提及，僅可確定或回轉處於 PREPARED 狀態的異動。例如：

```
imqcmd commit txn -n 64248349708800
```

還可配置代理程式，以在啓動代理程式時自動回轉處於 PREPARED 狀態的異動。請參閱第 47 頁的表格 2-4 中的 imq.transaction.autorollback 屬性，以獲得更多資訊。

管理受管理物件

使用受管理物件可允許對移植到其他 JMS 提供者的用戶端應用程式進行開發。受管理物件是指封裝提供者特定的配置和命名資訊的物件。這些物件通常由 Sun™ ONE Message Queue (MQ) 管理員建立並由用戶端應用程式使用，以便與代理程式建立連線，這些連線用來向實體目標傳送訊息並從實體目標接收訊息。

如需受管理物件的簡介，請參閱第 66 頁的「MQ 受管理物件」。

MQ 提供兩種管理工具，用於建立和管理受管理物件：指令行物件管理員公用程式 (imqobjmgr) 和 GUI 管理主控台。這些工具可讓您執行以下工作：

- 新增或刪除物件儲存區中的受管理物件。
- 列出現有受管理物件。
- 查詢並顯示有關受管理物件的資訊。
- 修改物件儲存區中的現有受管理物件。

本章介紹如何使用物件管理員公用程式 (imqobjmgr) 以執行這些工作。如需有關管理主控台的資訊，請參閱第 4 章「管理主控台指導」。

關於物件儲存區

受管理物件放在即刻可用的物件儲存區中，用戶端應用程式可透過 JNDI 查找來存取此處的受管理物件。您可以使用兩種類型的物件儲存區：標準的 LDAP 目錄伺服器或檔案系統物件儲存區。

LDAP 伺服器 LDAP 伺服器是用於生產訊息傳送系統的建議物件儲存區。LDAP 實施可從一些供應商處獲得，並設計為在分散式系統中使用。LDAP 伺服器還提供在生產環境中很有用的安全性功能。MQ 管理工具設計為與 LDAP 伺服器配合使用。

檔案系統儲存區 MQ 還支援檔案系統物件儲存區實施。當檔案系統物件儲存區未經過完全測試並因此不建議用於生產系統時，它的優點是易於在開發環境中使用。您所要做的是在您的本機檔案系統上建立目錄，而不是設定 LDAP 伺服器。任何對此目錄擁有存取權的使用者均可使用 MQ 管理工具，以建立並管理受管理物件。

受管理物件

如需受管理物件的簡介，請參閱第 66 頁的「MQ 受管理物件」。

MQ 受管理物件有兩種基本類型：連線工廠和目標。**連線工廠**受管理物件可由用戶端應用程式使用，以建立與代理程式的連線。**目標**受管理物件可由用戶端應用程式使用，以識別生產者傳送訊息或使用者擷取訊息的目標。(特殊的 SOAP 終點受管理物件用於 SOAP 訊息傳送 — 請參閱「MQ 開發人員指南」，以獲得更多資訊。)

依據訊息遞送模型 (點對點或出版/訂閱) 來使用特定類型的連線工廠和目標。例如，在點對點程式設計中，可使用 `queueConnectionFactory` 和佇列目標。同樣，在出版和訂閱程式設計中，可使用 `topicConnectionFactory` 和主題目標。還有非特定的連線工廠和目標受管理物件類型可用，如支援分散式異動的連線工廠類型 (請參閱第 33 頁的表格 1-1 以獲得所有支援的類型)。

物件管理員公用程式 (imqobjmgr)

物件管理員公用程式可讓您建立並管理 MQ 受管理物件。本節描述基本的 `imqobjmgr` 指令語法、提供子指令的清單，並總結 `imqobjmgr` 指令選項。後續各節介紹如何使用 `imqobjmgr` 子指令完成特定的工作。

指令語法

`imqcmd` 指令的一般語法如下：

```
imqobjmgr subcommand [options]
imqobjmgr -h|H
imqobjmgr -v
```

請注意，如果您指定 `-v`、`-h` 或 `-H` 選項，則系統不會執行指令行上指定的子指令。例如，如果您輸入以下指令，則系統會顯示版本資訊但不會執行 `list` 子指令。

```
imqobjmgr list -v
```

imqobjmgr 子指令

物件管理員公用程式 (`imqobjmgr`) 包括表格 7-1 中列出的子指令：

表格 7-1 `imqobjmgr` 子指令

子指令	描述
<code>add</code>	將受管理物件新增至物件儲存區。
<code>delete</code>	從物件儲存區刪除受管理物件。
<code>list</code>	列出物件儲存區中的受管理物件。
<code>query</code>	顯示有關指定的受管理物件的資訊。
<code>update</code>	修改物件儲存區中的現有受管理物件。

imqobjmgr 指令選項摘要

表格 7-2 列出 imqobjmgr 指令的選項。如需它們用法的說明，請參閱以下基於工作的各節。

表格 7-2 imqobjmgr 選項

選項	描述
-f	未經使用者確認即執行動作。
-h	顯示用法說明。不執行指令行上的其他任何指令。
-H	顯示用法說明、屬性清單和範例。不執行指令行上的其他任何指令。
-i <i>fileName</i>	指定輸入檔案的名稱，此輸入檔案包含所有或部分子指令子句，指定 (物件類型、查找名稱、物件屬性、物件儲存區屬性或其他選項)。通常用於重複資訊，如物件儲存區屬性。
-j <i>attribute=value</i>	指定識別和存取 JNDI 物件儲存區必需的屬性。
-javahome	指定要使用的替代 Java 2 相容運行時間 (預設為使用產品隨附的運行時間)。
-l <i>lookupName</i>	指定受管理物件的 JNDI 查找名稱。此名稱在物件儲存區的環境中必須唯一。
-o <i>attribute=value</i>	指定受管理物件的屬性。
-pre	預覽模式。表示在不執行指令的情況下將進行的動作。
-r <i>read-only_state</i>	指定受管理物件是否為唯讀物件。值 <code>true</code> 表示受管理物件為唯讀物件。JMS 用戶端無法修改唯讀受管理物件的屬性。依預設，唯讀狀態設定為 <code>false</code> 。
-s	靜謐模式。不顯示輸出。

表格 7-2 imqobjmgr 選項 (續)

選項	描述
-t <i>type</i>	指定 MQ 受管理物件的類型： q = 佇列 t = 主題 cf = ConnectionFactory qf = queueConnectionFactory tf = topicConnectionFactory xcf = XA ConnectionFactory (分散式異動) xqf = XA queueConnectionFactory (分散式異動) xtf = XA topicConnectionFactory (分散式異動) e = SOAP 終點 ¹
-v	顯示版本資訊。不執行指令行上的其他任何指令。

1 此受管理物件類型用於支援 SOAP 訊息 (請參閱「MQ 開發人員指南」)。

下節描述使用任何 imqobjmgr 子指令時需要提供的資訊。

需要的資訊

當執行與受管理物件相關的大部分工作時，管理員必須將以下資訊指定為 imqobjmgr 子指令的選項：

- 受管理物件的類型：
表格 7-2 中顯示了允許的類型。
- 受管理物件的 JNDI 查找名稱：
這是將在用戶端程式碼中使用的邏輯名稱，以參考物件儲存區中的受管理物件 (使用 JNDI)。
- 受管理物件的屬性：

- 對於佇列和主題：代理程式上實體目標的名稱。這是使用 `imqcmd create` 子指令的 `-n` 選項指定的名稱。如果您沒有指定此名稱，則將使用 `Untitled_Destination_Object` 的預設名稱。
- 對於連線工廠：將與用戶端連線的代理程式的主機名稱和連接埠號。如果您沒有指定此資訊，則使用本地主機和預設連接埠號 (7676)。第 150 頁的「受管理物件屬性」一節介紹如何指定物件屬性。

如需其他屬性，請參閱第 150 頁的「受管理物件屬性」。

- JNDI 物件儲存區的屬性：

此資訊取決於您要使用檔案系統儲存區還是 LDAP 伺服器，但必須包括以下屬性：

- JNDI 實施的類型 (初始環境屬性)。例如，檔案系統或 LDAP。
- 物件儲存區中受管理物件的位置 (提供者 URL 屬性)，即它所在的「資料夾」。
- 存取物件儲存區所需的使用者名稱、密碼和授權類型 (如果有)。

如需有關物件儲存區屬性的更多資訊，請參閱第 153 頁的「物件儲存區屬性」。

受管理物件屬性

使用屬性 - 值對指定的受管理物件的屬性。以下各節將描述這些屬性。

連線工廠受管理物件

連線工廠 (和 XA 連線工廠) 受管理物件具有表格 7-3 中列出的屬性。與您有主要關係的兩個屬性為 `imqBrokerHostPort` 和 `imqBrokerHostName`，它們可用於指定用戶端應用程式將與其建立連線的代理程式。第 157 頁的「新增連線工廠」一節介紹當您將連線工廠受管理物件新增至您的物件儲存區時，如何指定這些屬性。

如需連線工廠屬性的更多描述以及有關如何使用它們的資訊，請參閱「MQ 開發人員指南」和用於 MQ 類別 `com.sun.messaging.ConnectionConfiguration` 的 JavaDoc API 說明文件。

表格 7-3 連線工廠屬性

屬性/屬性名稱	類型	預設值
imqAckOnAcknowledge	字串	未指定
imqAckOnProduce	字串	未指定
imqAckTimeout	字串	0 毫秒
imqBrokerHostName	字串	localhost
imqBrokerHostPort	字串	7676
imqBrokerServicePort	字串	0
imqConfiguredClientID	字串	未指定
imqConnectionType	字串	TCP
imqConnectionURL	字串	http://localhost/imq/ tunnel
imqDefaultPassword	字串	guest
imqDefaultUsername	字串	guest
imqDisableSetClientID	字串	false
imqFlowControlCount	字串	100
imqFlowControlIsLimited	字串	false
imqFlowControlLimit	字串	1000
imqLoadMaxToServerSession	字串	false
imqQueueBrowserMax MessagesPerRetrieve	字串	1000
imqQueueBrowserRetrieve Timeout	字串	60,000 毫秒
imqReconnect	布林	false
imqReconnectDelay	字串	30,000 毫秒
imqReconnectRetries	字串	0
imqSetJMSXAppID	字串	false
imqSetJMSXConsumerTXID	字串	false
imqSetJMSXProducerTXID	字串	false
imqSetJMSXRcvTimestamp	字串	false
imqSetJMSXUserID	字串	false

表格 7-3 連線工廠屬性 (續)

屬性/屬性名稱	類型	預設值
imqSSLIsHostTrusted	字串	true
imqJMSDeliveryMode	整數	2 (永久性的)
imqJMSExpiration	整數	0 (未過期)
imqJMSPriority	整數	4 (一般優先順序)
imqOverrideJMSDeliveryMode	布林	false
imqOverrideJMSExpiration	布林	false
imqOverrideJMSPriority	布林	false
imqOverrideJMSHeadersToTemporaryDestinations	布林	false

目標受管理物件

目標受管理物件識別實體主題或佇列目標，其具有表格 7-4 中列出的屬性。第 158 頁的「新增主題或佇列」一節介紹當您將目標受管理物件新增至您的物件儲存區時，如何指定這些屬性。

與您有主要關係的屬性為 imqDestinationName。這是您指定給符合主題或佇列受管理物件的實體目標的名稱。您還可以提供目標描述，這會幫助您將此目標與您可能建立以支援許多應用程式的其他目標區分開來。

如需更多資訊，請參閱用於 MQ 類別 com.sun.messaging.DestinationConfiguration 的 JavaDoc API 說明文件。

表格 7-4 目標屬性

屬性/屬性名稱	類型	預設
imqDestinationDescription	字串	A Description for the destination Object
imqDestinationName	字串 ¹	Untitled_Destination_Object

1 目標名稱僅能包含字母數字字元 (無空格)，並且必須以字母字元或字元「_」和/或「\$」開頭。

物件儲存區屬性

物件儲存區的屬性使用 `-j` 選項指定，並由屬性 - 值對組成。一般來說，您必須指定以下屬性：

初始環境和位置資訊

這些項目的不同格式取決於您要使用檔案系統儲存區還是 LDAP 伺服器。

檔案系統儲存區 作為使用檔案系統儲存區的範例，請在 C 磁碟機上建立一個名為 *MyObjstore* 的資料夾，並分別為初始環境和位置屬性指定以下值：

```
-j "java.naming.factory.initial=
    com.sun.jndi.fscontext.RefFSContextFactory"

-j "java.naming.provider.url=file:///C:/MyObjStore"
```

LDAP 伺服器 作為使用 LDAP 伺服器的範例，請分別為初始環境和位置屬性指定以下值：

```
-j "java.naming.factory.initial=
    com.sun.jndi.ldap.LdapCtxFactory"

-j "java.naming.provider.url=ldap://mydomain.com:389/o=imq"
```

安全性資訊 (僅限於 LDAP)

這些項目的不同格式取決於 LDAP 提供者。您還應查閱 LDAP 實施隨附的說明文件，以確定是否在所有作業上還是僅在變更已儲存資料的作業上需要安全性資訊。

安全性屬性類似於：

```
-j "java.naming.security.principal=
    uid=fooUser, ou=People, o=imq"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"
```

表格 7-5 描述這些項目：

表格 7-5 物件儲存區的安全性屬性

屬性	描述
...principal	認證服務之呼叫者的負責人的身份。此項目的格式取決於認證機制。如果沒有指定此屬性，則運作方式將由服務提供者決定。
...credentials	認證服務之呼叫者的負責人的憑證。此屬性的值取決於認證機制。例如，它可以是隨機密碼、清除文字密碼、密鑰以及證書等。如果沒有指定此屬性，則運作方式將由服務提供者決定。
...authentication	要使用的安全性級別。它的值為以下關鍵字之一：none、simple 和 strong。如果沒有指定此屬性，則運作方式將由服務提供者決定。 如果您指定 simple，則 imqobjmgr 將提示任何遺漏的負責人或憑證值。這可讓您以更安全的方式提供識別資訊。

使用輸入檔案

imqobjmgr 指令可讓您指定使用 Java 屬性檔案語法的輸入檔案的名稱，以表示所有或部分 imqobjmgr 子指令子句。

將輸入檔案與物件管理員公用程式 (imqobjmgr) 配合使用，對指定物件儲存區屬性極為有用，這可能與跨 imqobjmgr 的多重呼叫一樣，並且一般需要大量的鍵入。使用輸入檔案還可讓您避免鍵入可能超出指令行所允許的最大字元數的情況。

imqobjmgr 輸入檔案的一般語法如下 (此版本屬性僅適用於輸入檔案 — 它不是指令行選項，並且它的值必須設定為 2.0)：

```
version=2.0
cmdtype=[ add | delete | list | query | update ]
obj.type=[ q | t | qf | tf | cf | xqf | xtf | xcf | e ]
obj.lookupName=lookup name
obj.attrs.objAttrName1=value1
obj.attrs.objAttrName2=value2
obj.attrs.objAttrNameN=valueN
...
objstore.attrs.objStoreAttrName1=value1
objstore.attrs.objStoreAttrName2=value2
objstore.attrs.objStoreAttrNameN=valueN
...
```

作為如何使用輸入檔案的範例，請考量以下 imqobjmgr 指令：

```
imqobjmgr add
-t qf
-l "cn=myQCF"
-o "imqBrokerHostName=foo"
-o "imqBrokerHostPort=777"
-j "java.naming.factory.initial=
    com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=
    ldap://mydomain.com:389/o=imq"
-j "java.naming.security.principal=
    uid=fooUser, ou=People, o=imq"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"
```

此指令可封裝在檔案 (如 MyCmdFile) 中，此檔案具有以下內容：

```
version=2.0
cmdtype=add
obj.type=qf
obj.lookupName=cn=myQCF
obj.attrs.imqBrokerHostName=foo
obj.attrs.imqBrokerHostPort=777
objstore.attrs.java.naming.factory.initial=\
    com.sun.jndi.ldap.LdapCtxFactory
objstore.attrs.java.naming.provider.url=\
    ldap://mydomain.com:389/o=imq
objstore.attrs.java.naming.security.principal=\
    uid=fooUser, ou=People, o=imq
objstore.attrs.java.naming.security.credentials=fooPasswd
objstore.attrs.java.naming.security.authentication=simple
```

然後，您可以使用 **-i** 選項將此檔案傳送至物件管理員公用程式 (imgobjmgr)：

```
imgobjmgr -i MyCmdFile
```

您還可以使用輸入檔案指定某些選項，同時使用指令行指定其他選項。這可讓您使用輸入檔案以指定部分子指令子句，這與跨公用程式的許多呼叫一樣。例如，以下指令指定新增連線工廠受管理物件所需的所有選項 (指定要在何處儲存受管理物件的那些選項除外)。

```
imgobjmgr add
  -t qf
  -l "cn=myQCF"
  -o "imqBrokerHostName=foo"
  -o "imqBrokerHostPort=777"
  -i MyCmdFile
```

在此例中，檔案 `MyCmdFile` 將包含以下定義：

```
version=2.0
objstore.attrs.java.naming.factory.initial=\
  com.sun.jndi.ldap.LdapCtxFactory
objstore.attrs.java.naming.provider.url=\
  ldap://mydomain.com:389/o=imq
objstore.attrs.java.naming.security.principal=\
  uid=fooUser, ou=People, o=imq
objstore.attrs.java.naming.security.credentials=fooPasswd
objstore.attrs.java.naming.security.authentication=simple
```

輸入檔案的其他範例可在以下位置找到：

```
IMQ_HOME/demo/imgobjmgr
```

新增和刪除受管理物件

本節介紹如何將連線工廠和主題或佇列目標的受管理物件新增至物件儲存區。

註 物件管理員公用程式 (imqobjmgr) 僅列出並顯示 MQ 受管理物件。如果物件儲存區應包含的非 MQ 物件與您想新增的受管理物件具有相同的查找名稱，則當您嘗試新增作業時會出現錯誤。

新增連線工廠

若要啓用用戶端應用程式以便與代理程式建立連線，您需要新增表示用戶端應用程式所需連線類型的受管理物件：主題連線工廠或佇列連線工廠

若要新增佇列連線工廠，請使用以下指令：

```
imqobjmgr add
-t qf
-l "cn=myQCF"
-o "imqBrokerHostName=myHost"
-o "imqBrokerHostPort=7272"
-j "java.naming.factoryinitial=
    com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=ldap://mydomain.com:389/o=imq"
-j "java.naming.security.principal=
    uid=fooUser, ou=People, o=imq"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"
```

上述指令建立查找名稱爲 cn=myQCF 的受管理物件，此物件將連線至在 myHost 上執行的代理程式，並偵聽連接埠 7272。此受管理物件儲存在 LDAP 伺服器中。

註 如果您要使用 LDAP 伺服器儲存受管理物件，則指定字首爲「cn=」的查找名稱很重要 (如上述範例所示)。您可以使用 -l 選項指定查找名稱。如果您要使用檔案系統物件儲存區，則無需使用此字首。

您可以透過將輸入檔案指定爲 imqobjmgr 指令的引數來完成相同的作業。如需更多資訊，請參閱第 154 頁的「使用輸入檔案」。

新增主題或佇列

若要啓用用戶端應用程式以存取代理程式上的實體目標，您需要將可識別這些目標的受管理物件新增至物件儲存區。

在將相應的受管理物件新增至物件儲存區之前，最好先建立實體目標。使用指令公用程式 (imqcmd) 以在代理程式上建立實體目標，這些實體目標可由物件儲存區中的目標受管理物件識別。如需有關建立實體目標的資訊，請參閱第 136 頁的「管理目標」。

以下指令新增可識別主題目標的受管理物件，此主題目標的查找名稱爲 myTopic，實體目標名稱爲 TestTopic。受管理物件儲存在 LDAP 伺服器中。

```
imqobjmgr add
-t t
-l "cn=myTopic"
-o "imqDestinationName=TestTopic"
-j "java.naming.factory.initial=
    com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=
    ldap://mydomain.com:389/o=imq"
-j "java.naming.security.principal=
    uid=fooUser, ou=People, o=imq"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"
```

這是相同的指令，只是受管理物件儲存在 Solaris 檔案系統中：

```
imqobjmgr add
-t t
-l "cn=myTopic"
-o "imqDestinationName=TestTopic"
-j "java.naming.factory.initial=
    com.sun.jndi.fscontext.RefFSContextFactory"
-j "java.naming.provider.url=
    file:///home/foo/imq_admin_objects"
```

例如，在 LDAP 伺服器一例中，您可以使用輸入檔案 `MyCmdFile` 以指定子指令子句。此檔案將包含以下文字：

```
version=2.0
cmdtype=add
obj.type=t
obj.lookupName=cn=myTopic
obj.attrs.imqDestinationName=TestTopic
objstore.attrs.java.naming.factory.initial=
    com.sun.jndi.fscontext.RefFSContextFactory
objstore.attrs.java.naming.provider.url=
    file:///home/foo/imq_admin_objects
objstore.attrs.java.naming.security.principal=
    uid=fooUser, ou=People, o=imq
objstore.attrs.java.naming.security.credentials=fooPasswd
objstore.attrs.java.naming.security.authentication=simple
```

使用 `-i` 選項將檔案傳送至 `imqobjmgr` 指令：

```
imqobjmgr -i MyCmdFile
```

註

如果您要使用 LDAP 伺服器儲存受管理物件，則指定字首為「`cn=`」的查找名稱很重要（如上述範例所示）。您可以使用 `-l` 選項指定查找名稱。如果您要使用檔案系統物件儲存區，則無需使用此字首。

除了為 `-t` 選項指定 `q`，新增佇列物件的方法完全相同。

刪除受管理物件

使用 `delete` 子指令以刪除受管理物件。您必須指定物件的查找名稱、類型和位置。

以下指令刪除某主題的受管理物件，此主題的查找名稱爲 `cn=myTopic`，並儲存在 LDAP 伺服器上。

```
imqobjmgr delete
-t t
-l "cn=myTopic"
-j "java.naming.factory.initial=
    com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=
    ldap://mydomain.com:389/o=imq"
-j "java.naming.security.principal=
    uid=fooUser, ou=People, o=imq"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"
```

取得資訊

使用 `list` 和 `query` 子指令以列出物件儲存區中的受管理物件，並顯示有關個別物件的資訊。

列出受管理物件

使用 `list` 子指令以取得所有受管理物件的清單，或取得特定類型的所有受管理物件的清單。以下範例程式碼假設受管理物件儲存在 LDAP 伺服器中。

以下指令列出所有物件。

```
imgobjmgr list
-j "java.naming.factory.initial=
    com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=
    ldap://mydomain.com:389/o=imq"
-j "java.naming.security.principal=
    uid=fooUser, ou=People, o=imq"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"
```

以下指令列出類型 queue 的所有物件。

```
imgobjmgr list
-t q
-j "java.naming.factory.initial=
    com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=
    ldap://mydomain.com:389/o=imq"
-j "java.naming.security.principal=
    uid=fooUser, ou=People, o=imq"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"
```

有關單一物件的資訊

使用 `query` 子指令，以取得有關受管理物件的資訊。您必須指定物件的查找名稱和包含受管理物件的物件儲存區的屬性（如初始環境和位置）。

在以下範例中，`query` 子指令用來顯示有關查找名稱爲 `myTopic` 之物件的資訊。

```
imgobjmgr query
-l "cn=myTopic"
-j "java.naming.factory.initial=
    com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=
    ldap://mydomain.com:389/o=img"
-j "java.naming.security.principal=
    uid=fooUser, ou=People, o=img"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"
```

更新受管理物件

您可以使用 `update` 指令修改受管理物件的屬性。您必須指定物件的查找名稱和位置。您可以使用 `-o` 選項修改屬性值。

此指令變更表示主題連線工廠受管理物件的屬性：

```
imgobjmgr update
-t tf
-l "cn=MyTCF"
-o imgReconnectRetries=3
-j "java.naming.factory.initial=
    com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=
    ldap://mydomain.com:389/o=img"
-j "java.naming.security.principal=
    uid=fooUser, ou=People, o=img"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"
```

安全性管理

本章介紹如何執行與安全性相關的工作，其中包括認證、授權和加密。

認證使用者 您負責維護使用者儲存庫中的使用者、使用者群組以及密碼清單。本章的第一部分介紹如何建立、移入和管理使用者儲存庫。如需 Sun™ ONE Message Queue (MQ) 安全性的介紹，請參閱第 51 頁的「安全性管理程式」。

授權使用者 您負責編輯屬性檔案，此屬性檔案將使用者對代理程式作業的存取權對映至使用者的名稱或群組。本章的第二部分介紹如何自訂此屬性檔案。

加密：設定 SSL 服務 使用基於安全套接層 (SSL) 標準的連線服務，可讓您加密在用戶端與代理程式之間傳送的訊息。如需 MQ 如何處理加密的介紹，請參閱第 53 頁的「加密 (企業版)」。本章的最後一部分介紹如何設定基於 SSL 的連線服務，並提供有關使用 SSL 的其他資訊。

對於代理程式安全存取 SSL 鍵值儲存區、LDAP 使用者儲存庫或 JDBC 相容永久性儲存區所需的密碼，有三種方法可提供此類密碼：

- 透過啓動代理程式時讓系統提示您輸入
- 透過啓動代理程式時將密碼作為指令行選項輸入 (請參閱第 108 頁的「啓動代理程式」與第 109 頁的表格 5-2)
- 透過啓動代理程式時將密碼儲存在系統可存取的密碼檔案中 (請參閱第 182 頁的「使用密碼檔案」)

認證使用者

當使用者嘗試連線至代理程式時，此代理程式將透過檢查提供的名稱和密碼認證此使用者，如果名稱和密碼與系統配置此代理程式要參考的使用者儲存庫中的名稱和密碼相符，則代理程式承認此連線。此儲存庫有兩種類型：

- MQ 隨附的文字檔案儲存庫

這種類型的使用者儲存庫非常易於使用；但是其安全性易受侵害，因此應僅將其用於評估與開發。您可以使用使用者管理員公用程式 (imqusermgr) 移入和管理此儲存庫。若要啓用認證，您應移入包含每個使用者名稱、密碼以及使用者群組名稱的使用者儲存庫。

如需有關設定與管理使用者儲存庫的更多資訊，請參閱第 164 頁的「使用文字檔案使用者儲存庫」。

- LDAP 伺服器

這可以是現有或新的 LDAP 目錄伺服器，它將 LDAP v2 或 v3 協定用於使用者儲存庫。它不如文字檔案儲存庫易於使用，但是它很安全，因此更適用於生產環境。

如果您要使用 LDAP 使用者儲存庫，則將需要使用 LDAP 供應商提供的工具來移入和管理此使用者儲存庫。如需更多資訊，請參閱第 170 頁的「將 LDAP 伺服器用於使用者儲存庫」。

使用文字檔案使用者儲存庫

MQ 提供了文字檔案使用者儲存庫和指令行工具，即 MQ 使用者管理員 (imqusermgr)，您可以使用此工具移入和管理文字檔案使用者儲存庫。以下各節描述了文字檔案使用者儲存庫、其初始項目以及如何移入和管理此儲存庫。

預設文字檔案儲存庫位於：

IMQ_HOME/etc/passwd (在 Solaris 上為 /etc/imq/passwd)

此儲存庫隨附兩個已定義的項目 (列)，如下表中所述。

表格 8-1 使用者儲存庫中的初始項目

使用者名稱	密碼	群組	狀態
admin	admin	admin	active
guest	guest	anonymous	active

這些初始項目可讓 MQ 代理程式在安裝之後立即使用，而無需管理員的介入。換言之，要使用的 MQ 代理程式無需初始使用者/密碼設定。

初始 guest 使用者項目可讓 JMS 用戶端使用預設 guest 使用者名稱和密碼連線至代理程式（例如用於測試）。

初始 admin 使用者項目可讓您使用 imqcmd 指令並使用預設 admin 使用者名稱和密碼來管理代理程式。我們建議您更新此初始項目以變更密碼。

您可以使用使用者管理員公用程式編輯或移入文字檔案使用者儲存庫，而不必先配置或啟動代理程式。使用使用者管理員公用程式的唯一要求是，要在已安裝代理程式的主機上執行它，並且如果您要向儲存庫中寫入，則必須擁有適當的許可權：

- 在 Solaris 上，使用者管理員公用程式可由超級使用者執行，或者由透過基於存取控制的 Solaris 角色授予存取權的其他非超級使用者執行。若要授予此類存取權，超級使用者必須將 /etc/user_attr 的項目變成以下形式：

```
使用者名稱 :: :type=normal;profiles=Message Queue Management
```

這可將指定的使用者新增至 MQ 權限設定檔。若要使用此功能，您必須先執行設定檔 shell (pfsh、pfksh 和 pfcsh)，例如，

```
% /usr/bin/pfsh
```

然後執行需要的使用者管理員 (imqusermgr) 指令。

如需有關基於存取控制的 Solaris 角色的更多資訊，請參閱：

<http://docs.sun.com/>

?q=Rights+Profile&p=/doc/806-4078/6jd6cjrsl&a=view)。

- 在 Windows 上，安裝之後，使用者儲存庫檔案可由任何使用者寫入，因為作業系統不控制對使用基於使用者名稱的許可權屬性之檔案的存取。

以下各節介紹如何移入和管理文字檔案使用者儲存庫。

使用者管理員公用程式 (imqusermgr)

使用者管理員公用程式可讓您管理基於檔案的使用者儲存庫。本節描述基本的 imqusermgr 指令語法、提供子指令的清單，並總結 imqusermgr 指令選項。後續各節介紹如何使用 imqobjmgr 子指令完成特定的工作。

指令的語法

imqusermgr 指令的一般語法如下：

```
imqusermgr subcommand [options]
imqusermgr -h
imqusermgr -v
```

請注意，如果您指定 `-v` 或 `-h` 選項，則系統不會執行指令行上指定的子指令。例如，如果您輸入以下指令，則系統會顯示版本資訊但不會執行 `list` 子指令。

```
imqusermgr list -v
```

imqusermgr 子指令

表格 8-2 列出了 imqusermgr 子指令。

表格 8-2 imqusermgr 子指令

子指令	描述
add -u <i>name</i> -p <i>passwd</i> [-g <i>group</i>] [-s]	將使用者和相關密碼新增至儲存庫，並選擇性地指定使用者的群組。
delete -u <i>name</i> [-s] [-f]	從儲存庫中刪除指定的使用者。
list [-u <i>name</i>]	顯示有關指定使用者或所有使用者的資訊。
update -u <i>name</i> -p <i>passwd</i> [-a <i>state</i>] [-s] [-f]	更新指定使用者的密碼和/或狀態。
update -u <i>name</i> -a <i>state</i> [-p <i>passwd</i>] [-s] [-f]	

imqusermgr 指令選項摘要

表格 8-3 列出了 imqusermgr 指令的選項。

表格 8-3 imqusermgr 選項

選項	描述
-a <i>active_state</i>	指定使用者的狀態是否應為作用中 (<code>true/false</code>)。值 <code>true</code> 表示狀態為作用中。這是預設值。
-f	執行動作而無需使用者確認
-h	顯示用法說明。不執行指令行上的其他任何指令。
-p <i>passwd</i>	指定使用者的密碼。
-g <i>group</i>	指定使用者群組。有效值為 <code>admin</code> 、 <code>user</code> 和 <code>anonymous</code> 。
-s	設定靜謐模式。
-u <i>name</i>	指定使用者名稱。
-v	顯示版本資訊。不執行指令行上的其他任何指令。

群組

當將使用者項目新增至儲存庫時，管理員可以為此使用者選擇指定預先定義的三個群組之一：`admin`、`user` 或 `anonymous`。如果未指定群組，則指定預設群組 `user`。

- *admin* 群組用於代理程式管理員。依預設，已指定此群組的使用者可以配置和管理代理程式。管理員可以將多個使用者指定給 `admin` 群組。
- *user* 群組用於普通 (非管理) JMS 用戶端應用程式。大多數 MQ 用戶端應用程式將存取已在 `user` 群組中認證的代理程式。依預設，用戶端應用程式本身可以向所有主題與佇列生產訊息，以及使用來自所有主題與佇列的訊息，或者瀏覽任何佇列中的訊息。
- *anonymous* 群組用於 JMS 用戶端應用程式，這些應用程式不想使用代理程式知道的使用者名稱 (可能因為應用程式不知道要使用的實際使用者名稱)。這對於大多數 FTP 伺服器中存在的匿名帳號均類似。每次管理員僅可以將一個使用者指定給 `anonymous` 群組。預計您將透過存取控制限制此群組的存取權限 (與 `user` 群組相比較)，或者您將在佈署時從此群組中移除使用者。

爲了變更使用者的群組，管理員必須刪除使用者項目並爲此使用者新增其他項目，從而指定新的群組。

您可以指定定義群組成員可執行之作業的存取規則。如需更多資訊，請參閱第 172 頁的「授權使用者：存取控制屬性檔案」。

狀態

當管理員將使用者新增至儲存庫之後，此使用者的狀態依預設爲作用中。若要讓此使用者變成非作用中，管理員必須使用更新指令。例如，以下指令讓使用者 `JoeD` 狀態變成非作用中：

```
imqusermgr update -u JoeD -a false
```

已變成非作用中使用者的項目保留在儲存庫中；但是，非作用中使用者無法開啓新的連線。如果使用者爲非作用中，並且管理員新增具有相同名稱的其他使用者，則作業將失敗。管理員必須刪除非作用中使用者項目或變更新使用者的名稱，或者爲新使用者使用其他名稱。這樣可避免管理員新增重複名稱或密碼。

使用者名稱與密碼的格式

使用者名稱與密碼必須遵循以下原則：

- 使用者名稱與密碼不能包含**表格 8-4** 中所列字元。

表格 8-4 無效的使用者名稱與密碼字元

字元	描述
*	星號
,	逗號
:	冒號

- 使用者名稱與密碼不能包含新行或換行鍵作為字元。
- 如果名稱或密碼包含空格，則必須將整個名稱或密碼括在引號內。
- 名稱或密碼必須至少包含一個字元。
- 密碼或使用者名稱的長度沒有限制，除非指令 **shell** 對指令行上可輸入字元的最大數量強加限制。

移入和管理使用者儲存庫

使用 **add** 子指令可將使用者新增至儲存庫。例如，以下指令新增使用者 **Katharine** 和密碼 **sesame**。

```
imusermgr add -u Katharine -p sesame -g user
```

使用 **delete** 子指令可從儲存庫中刪除使用者。例如，以下指令刪除使用者 **Bob**：

```
imusermgr delete -u Bob
```

使用 **update** 子指令可變更使用者的密碼或狀態。例如，以下指令將 **Katharine** 的密碼變更為 **alladin**：

```
imusermgr update -u Katharine -p alladin
```

若要列出有關一個或多個使用者的資訊，請使用 `list` 指令。以下指令顯示有關名為 `isa` 的使用者之資訊：

```
imqusermgr list -u isa
```

```
-----
User Name      Group          Active State
-----
isa            admin          true
```

以下指令列出有關所有使用者的資訊：

```
imqusermgr list
```

```
-----
User Name      Group          Active State
-----
testuser3      user           true
testuser2      user           true
testuser1      user           true
isa            admin          true
admin          admin          true
guest          anonymous      true
testuser5      user           false
testuser4      user           false
```

變更預設管理員密碼

為了安全性，您必須將 `admin` 的預設密碼變更爲只有您自己知道的密碼。您需要使用 `imqusermgr` 工具來完成此作業。

以下指令將預設密碼變更爲 `grandpoobah`。

```
imqusermgr update -u admin -p grandpoobah
```

當代理程式正在執行時，您可以透過執行任一指令行工具來快速確認此變更生效。例如，以下指令應生效，

```
imqcmd list svc -u admin -p grandpoobah
```

而使用舊密碼應失敗。

變更密碼之後，使用任一管理工具（包括管理主控台）時，您均應提供新密碼。

將 LDAP 伺服器用於使用者儲存庫

如果您要將 LDAP 伺服器用於使用者儲存庫，則必須在實例配置檔案中設定某些代理程式屬性。當使用者嘗試連線至代理程式或執行某些作業時，這些屬性可讓代理程式查詢 LDAP 伺服器以獲得有關使用者和群組的資訊。實例配置檔案位於

```
IMQ_VARHOME/instances/brokerName/props/config.properties
(在 Solaris 上為
/var/imq/instances/brokerName/props/config.properties)
```

► 編輯配置檔案以使用 LDAP 伺服器

- 1. 透過設定以下屬性指定您要使用 LDAP 使用者儲存庫：

```
imq.authentication.basic.user_repository=ldap
```

- 2. 設定 `imq.authentication.type` 屬性，以確定密碼應以基本 64 編碼 (basic) 還是以 MD5 摘要 (digest) 從用戶端傳送至代理程式。當將 LDAP 目錄伺服器用於使用者儲存庫時，您必須將認證類型設定為 basic。例如，

```
imq.authentication.type=basic
```

- 3. 您還必須設定可控制 LDAP 存取的代理程式屬性。這些屬性儲存在代理程式的實例配置文件中，在表格 8-5 中有描述。MQ 使用 JNDI API 與 LDAP 目錄伺服器通訊。請參閱 JNDI 說明文件，以獲得有關這些屬性中所參考之語法與術語的更多資訊。MQ 3.0.1 使用 Sun JNDI LDAP 供應程式並使用簡單認證。

表格 8-5 與 LDAP 相關的屬性

屬性	描述
imq.user_repository. ldap.server	用於 LDAP 伺服器的主機：連接埠。主機指定執行目錄伺服器之主機的完全合格的 DNS 名稱。連接埠指定目錄伺服器用於通訊的連接埠號。
imq.user_repository. ldap.principal	識別名稱，代理程式將使用此名稱連結至目錄伺服器以進行搜尋。如果目錄伺服器允許匿名搜尋，則無需為此屬性指定值。

表格 8-5 與 LDAP 相關的屬性 (續)

屬性	描述
imq.user_repository. ldap.password	與代理程式所用識別名稱相關聯的密碼。僅能在密碼檔案中指定 (請參閱第 182 頁的「使用密碼檔案」)。 如需更高安全性，請讓代理程式提示您提供密碼，或使用以下指令行選項指定密碼：imqbrokerd -ldappassword。 如果目錄伺服器允許匿名搜尋，則無需密碼。
imq.user_repository. ldap.base	用於使用者項目的目錄庫。
imq.user_repository. ldap.uidattr	供應程式特定的屬性識別碼，其值可唯一識別使用者。例如：uid 和 cn 等。
imq.user_repository. ldap.usrfilter	JNDI 搜尋過濾器 (表示為邏輯表示式的搜尋查詢)。透過為使用者指定搜尋過濾器，代理程式可以縮小搜尋範圍以便提高效率。如需更多資訊，請參閱位於以下位置的 JNDI 指導： http://java.sun.com/products/jndi/tutorial。 此屬性可不設定。
imq.user_repository. ldap.grpsearch	指定您是否要啟用群組搜尋的布林。請參閱您的 LDAP 供應商提供的說明文件，以確定您是否可以將使用者與群組相關聯。 請注意，MQ 3.0.1 中不支援巢式群組。 預設：false
imq.user_repository. ldap.grpbase	用於群組項目的目錄庫。
imq.user_repository. ldap.gidattr	供應程式特定的屬性識別碼，其值為群組名稱。
imq.user_repository. ldap.memattr	群組項目中的屬性識別碼，其值為群組成員的識別名稱。
imq.user_repository. ldap.grpfiltler	JNDI 搜尋過濾器 (表示為邏輯表示式的搜尋查詢)。透過為群組指定搜尋過濾器，代理程式可以縮小搜尋範圍以便提高效率。如需更多資訊，請參閱位於以下位置的 JNDI 指導。 http://java.sun.com/products/jndi/tutorial 此屬性可不設定。

表格 8-5 與 LDAP 相關的屬性 (續)

屬性	描述
<code>imq.user_repository.ldap.timeout</code>	指定搜尋時間限制的整數 (以秒為單位)。依預設可設定為 180 秒。
<code>imq.user_repository.ldap.ssl.enabled</code>	指定代理程式與 LDAP 通訊時是否應使用 SSL 協定的布林。依預設可設定為 <code>false</code> 。

請參閱代理程式的 `default.properties` 檔案，以獲得與 LDAP 使用者儲存庫相關的屬性設定範例 (預設)。

4. 如有必要，您需要編輯存取控制屬性檔案中的使用者/群組和規則。如需有關使用存取控制屬性檔案的更多資訊，請參閱第 172 頁的「[授權使用者：存取控制屬性檔案](#)」。
5. 如果在連線認證和群組搜尋期間，您要讓代理程式透過 SSL 與 LDAP 目錄伺服器通訊，則需要啟動 LDAP 伺服器中的 SSL，然後在代理程式配置檔案中設定以下屬性：
 - 為 LDAP 使用者儲存庫屬性指定安全連接埠。例如：


```
imq.user_repository.ldap.server=myhost:7878
```
 - 將代理程式的 property `imq.user_repository.ldap.ssl.enabled` 設定為 `true`。

授權使用者：存取控制屬性檔案

連線至代理程式之後，使用者可能要生產訊息、使用目標上的訊息，或瀏覽佇列目標上的訊息。當使用者嘗試上述作業時，代理程式將檢查存取控制屬性檔案 (ACL 檔案) 以查看此使用者是否已被授權執行此作業。ACL 檔案包含的規則可指定特定使用者 (或使用者群組) 已被授權執行哪些作業。依預設，允許所有認證使用者在任何目標上生產和使用訊息。您可以編輯存取控制屬性檔案，以限制某些使用者和群組執行這些作業。

無論使用者資訊放在文字檔案儲存庫中還是放在 LDAP 儲存庫中，均可使用 ACL 檔案。預設 ACL 屬性檔案將與代理程式一同安裝。此檔案名稱為 `accesscontrol.properties`，並由安裝程式放在以下目錄中：

`IMQ_HOME/etc` (在 Solaris 上為 `/etc/imq`)

ACL 檔案的格式與 Java 屬性檔案的格式類似。它的開始部分定義檔案的版本，然後分三部分指定存取控制規則：

- 連線存取控制
- 目標存取控制
- 目標自動建立存取控制

version 屬性定義 ACL 屬性檔案的版本；您不能變更此項目。

```
version=JMQFileAccessControlModel/100
```

以下描述存取規則的基本語法並介紹如何計算許可權，並且還描述指定存取控制的 ACL 檔案的三個部分。

存取規則語法

在 ACL 屬性檔案中，存取控制定義了特定使用者或群組對受保護的資源（如目標和連線服務）擁有何種存取權。存取控制由一條規則或一組規則表示，每條規則用一個 Java 屬性表示：

這些規則的基本語法如下：

```
resourceType.resourceVariant.operation.access.principalType = principals
```

表格 8-6 描述了語法規則的元素。

表格 8-6 存取規則的語法元素

元素	描述
resourceType	為以下之一：connection、queue 或 topic。
resourceVariant	resourceType 指定的類型實例。例如，myQueue。萬用字元符號 (*) 可用於表示所有連線服務類型或所有目標。
operation	值取決於要制定的存取規則種類。
access	為以下之一：allow 或 deny。
principalType	為以下之一：user 或 group。如需更多資訊，請參閱第 167 頁的「群組」。
principals	可以擁有在規則左側指定的存取權的使用者。如果 principalType 為 user，則此項可以為個別使用者或使用者清單（以逗號分隔）；如果 principalType 為 group，則此項可以為單一群組或群組清單（以逗號分隔的清單）。萬用字元符號 (*) 可用於表示所有使用者或所有群組。

以下為存取規則的一些範例：

- 以下規則表示所有使用者均可以將訊息傳送至名為 **q1** 的佇列。

```
queue.q1.produce.allow.user=*
```

- 以下規則表示任何使用者均可以將訊息傳送至任何佇列。

```
queue.*.produce.allow.user=*
```

註

若要指定非 ASCII 的使用者、群組或目標名稱，您必須使用統一碼退出符號 (\uXXXX) 表示法。如果您已經編輯並儲存具有非 ASCII 編碼名稱的 ACL 檔案，則可以使用 `Java native2ascii` 工具將其轉換為 ASCII。如需更加詳細的資訊，請參閱 <http://java.sun.com/j2se/1.3/docs/guide/intl/faq.html>

許可權運算

當運算一系列規則所包含的許可權時，可套用以下原則：

- 特定存取規則置換一般存取規則。套用以下兩條規則之後，所有使用者均可以將訊息傳送至所有佇列，但是 **Bob** 卻無法將訊息傳送至 **tq1**。

```
queue.*.produce.allow.user=*
```

```
queue.tq1.produce.deny.user=Bob
```

- 賦予明確 *principal* 的存取權置換賦予 ** principal* 的存取權。以下規則拒絕 **Bob** 向 **tq1** 生產訊息，但允許其他所有使用者向 **tq1** 生產訊息。

```
queue.tq1.produce.allow.user=*
```

```
queue.tq1.produce.deny.user=Bob
```

- 用於使用者的 ** principal* 規則置換用於群組的相應 ** principal* 規則。例如，以下兩條規則允許所有認證使用者將訊息傳送至 **tq1**。

```
queue.tq1.produce.allow.user=*
```

```
queue.tq1.produce.deny.group=*
```

- 授予使用者的存取權置換授予使用者群組的存取權。在以下範例中，如果 **Bob** 為 **User** 成員，則他將被拒絕向 **tq1** 生產訊息，但是 **User** 的所有其他成員均可以向 **tq1** 生產訊息。

```
queue.tq1.produce.allow.group=User
```

```
queue.tq1.produce.deny.user=Bob
```

- 任何未透過存取規則明確授予的存取許可權，都會被默認拒絕。例如，如果 ACL 檔案不包含存取規則，則所有使用者均被拒絕所有作業。
- 相同使用者或群組的拒絕和允許許可權會相互抵銷。例如，以下兩條規則導致 Bob 無法瀏覽 t1：

```
queue.q1.browse.allow.user=Bob
```

```
queue.q1.browse.deny.user=Bob
```

以下兩條規則導致群組 User 無法使用 q5 上的訊息。

```
queue.q5.consume.allow.group=User
```

```
queue.q5.consume.deny.group=User
```

- 當存在多條相同的左側規則時，只有最後一項生效。

連線存取控制

ACL 屬性檔案中的連線存取控制部分，包含用於代理程式連線服務的存取控制規則。連線存取控制規則的語法如下：

```
connection.resourceVariant.access.principalType = principals
```

resourceVariant 定義有兩個值：NORMAL 與 ADMIN。依預設，所有使用者對 NORMAL 類型連線服務擁有存取權，但是只有那些群組為 admin 的使用者才對 ADMIN 類型連線服務擁有存取權。

您可以編輯連線存取控制規則，以限制使用者的連線存取權限。例如，以下規則拒絕 Bob 存取 NORMAL，但是允許其他所有使用者進行存取：

```
connection.NORMAL.deny.user=Bob
```

```
connection.NORMAL.allow.user=*
```

您可以使用星號 (*) 字元來指定所有認證使用者或群組。

您不能建立自己的服務類型；您必須將自己限制於由常數 NORMAL 與 ADMIN 所指定的預先定義類型之內。

目標存取控制

存取控制屬性檔案的目標存取控制部分，包含基於目標的存取控制規則。這些規則確定何人 (使用者/群組) 可以在何處 (目標) 做什麼 (作業)。這些規則所規定的存取權類型包括：將訊息傳送至佇列、將訊息出版至主題、接收來自佇列的訊息、訂閱主題以及瀏覽佇列中的訊息。

依預設，任何使用者或群組對任何目標擁有所有類型的存取權。您可以新增多條特定的目標存取規則，或編輯預設規則。本節的其餘部分介紹目標存取規則的語法，您必須理解此語法以編寫自己的規則。

目標規則的語法如下：

```
resourceType.resourceVariant.operation.access.principalType = principals
```

表格 8-7 描述了這些元素：

表格 8-7 目標存取控制規則的元素

元件	描述
resourceType	必須為 queue 或 topic 之一。
resourceVariant	目標名稱或所有目標 (*)，表示所有佇列或所有主題。
operation	必須為 produce、consume 或 browse 之一。
access	必須為 allow 或 deny 之一。
principalType	必須為 user 或 group 之一。

存取權可以賦予一個或多個使用者和/或一個或多個群組。

以下範例說明了不同種類的目標存取控制規則：

- 允許所有使用者將訊息傳送至任何佇列目標。
queue.*.produce.allow.user=*
- 拒絕群組 user 的任何成員訂閱主題 Admissions。
topic.Admissions.consume.deny.group=user

目標自動建立存取控制

ACL 屬性檔案的最後部分，包括指定代理程式將為哪些使用者和群組自動建立目標的存取規則。

當使用者在並不存在的目標上建立生產者或使用者時，如果代理程式的自動建立屬性已啟用且實體目標尚未存在，則代理程式將建立此目標。

依預設，任何使用者或群組均有權讓代理程式自動建立目標。此權限由以下規則指定：

```
queue.create.allow.user=*
```

```
topic.create.allow.user=*
```

您可以編輯 ACL 檔案以限制此類型的存取權。

目標自動建立存取規則的一般語法如下：

```
resourceType.create.access.principalType = principals
```

其中 *resourceType* 為 *queue* 或 *topic*。

例如，以下規則可讓代理程式為除 Snoopy 之外的所有使用者自動建立主題目標。

```
topic.create.allow.user=*
```

```
topic.create.deny.user=Snoopy
```

請注意，目標自動建立規則的效果必須與目標存取規則的效果一致。例如，如果您 1) 變更目標存取規則，以禁止任何使用者將訊息傳送至目標，但是 2) 啟用目標的自動建立功能，如果此目標不存在，則代理程式**將會**建立此目標，但**不會**將訊息遞送至此目標。

加密：使用 SSL 服務 (企業版)

MQ 企業版支援基於安全套接層 (SSL) 標準的連線服務：透過 TCP/IP (ssljms 與 ssladmin) 和透過 HTTP (httpsjms)。這些基於 SSL 的連線服務允許加密在用戶端與代理程式之間傳送的訊息。目前的 MQ 版次支援基於自身簽名伺服器證書的 SSL 加密。

若要使用基於 SSL 的連線服務，您需要使用鍵值工具公用程式 (imqkeytool) 產生私密密鑰/公開密鑰對。此公用程式將公開密鑰內嵌於自身簽名的證書中，此證書被傳送至請求與代理程式連線的用戶端，此用戶端使用此證書設定加密連線。

雖然 MQ 的基於 SSL 的連線服務在概念上相似，但是在設定方法上卻有些差異。因此，以下各節分別說明透過 TCP/IP 和 HTTP 的安全連線。

設定透過 TCP/IP 的 SSL 服務

兩種基於 SSL 的連線服務可提供透過 TCP/IP 的直接而安全的連線。

ssljms 此連線服務用於在用戶端與代理程式之間，透過安全而加密的連線遞送 JMS 訊息。

ssladmin 此連線服務用於在指令公用程式 (imqcmd) 這種指令行管理工具與代理程式之間建立安全而加密的連線。管理主控台 (imqadmin) 不支援安全連線。

► 設定 ssljms 連線服務

1. 產生自身簽名的證書。
2. 啟用代理程式中的 ssljms 連線服務。
3. 啟動代理程式。
4. 配置並執行用戶端。

除步驟 4 (配置並執行用戶端) 之外，設定 ssljms 連線服務的程序與設定 ssladmin 連線服務的程序相同。

以下各節較詳細地說明每個步驟。

步驟 1. 產生自身簽名的證書

在假設用戶端正與已知且可信任的伺服器通訊的情況下，MQ 3.0.1 中的 SSL 支援可保護線上資料。因此在 MQ 3.0.1 中，僅使用自身簽名的證書即可實施 SSL。

執行 imkeytool 指令，以便為代理程式產生自身簽名的證書。ssljms 和 ssladmin 連線服務可使用相同的證書。在指令提示符號下輸入以下內容：

```
imkeytool -broker
```

此公用程式將提示您提供它所需的資訊。(在 Unix 系統上，為了擁有建立鍵值儲存區的許可權，您可能需要以超級使用者 [root] 的身份執行 imkeytool。)

首先，imkeytool 提示您提供鍵值儲存區密碼以及某些組織資訊，然後提示您確認。接收到確認之後，它會在產生鍵對時暫停。然後它會要求您提供密碼，以鎖定特定的鍵對 (鍵密碼)；您應輸入 [Return] 以回應此提示：這會使鍵密碼與鍵值儲存區密碼相同。

註

請記住您所提供的密碼，稍後您需要將此密碼提供給代理程式 (當您啟動它時)，以便代理程式可開啓鍵值儲存區。您還可以將鍵值儲存區密碼儲存在密碼檔案中 (請參閱 [第 182 頁的「使用密碼檔案」](#))。

執行 `imqkeytool` 可執行 JDK `keytool` 公用程式，以產生自身簽名的證書並將其放在 MQ 的鍵值儲存區中，此儲存區位於

`IMQ_HOME/etc/keystore` (在 Solaris 上為 `/etc/imq/keystore`)

此鍵值儲存區的格式與 JDK1.2 `keytool` 公用程式支援的格式相同。

MQ 鍵值儲存區的可配置屬性如**表格 8-8** 中所示。(如需有關配置這些屬性的說明，請參閱**第 5 章「啟動與配置代理程式」**。)

表格 8-8 鍵值儲存區屬性

屬性名稱	描述
<code>imq.keystore.file.dirpath</code>	用於基於 SSL 的服務：指定包含鍵值儲存區檔案之目錄的路徑。 預設： <code>IMQ_HOME/etc</code> (在 Solaris 上為 <code>/etc/imq/</code>)
<code>imq.keystore.file.name</code>	用於基於 SSL 的服務：指定鍵值儲存區檔案的名稱。 預設： <code>keystore</code>
<code>imq.keystore.password</code>	用於基於 SSL 的服務：指定鍵值儲存區密碼。僅能儲存在密碼檔案中 (請參閱 第 182 頁 的「 使用密碼檔案 」)。如需更高安全性，請讓代理程式提示您提供密碼，或使用以下指令行選項指定密碼： <code>imqbrokerd -password</code> 。

您可能需要重新產生鍵對以解決某些問題；例如：

- 您忘記了鍵值儲存區密碼。
- 啟動代理程式時 SSL 服務無法初始化，並且您收到以下異常：
`java.security.UnrecoverableKeyException: Cannot recover key.`

可能導致此異常的原因為，您提供的鍵密碼與您在**第 178 頁**的「**步驟 1. 產生自身簽名的證書**」中產生自身簽名證書時的鍵值儲存區密碼不同。

► 重新產生鍵對

1. 移除位於以下位置的代理程式的鍵值儲存區：

`IMQ_HOME/etc/keystore` (在 Solaris 上為 `/etc/imq/keystore`)

2. 重新執行 `imqkeytool` 以產生鍵對，如**第 178 頁**的「**步驟 1. 產生自身簽名的證書**」中所述。

步驟 2. 啟用代理程式中基於 SSL 的服務

若要啟用代理程式中的 SSL 服務，您需要將 `ssljms` (`ssladmin`) 新增至 `imq.service.activelist` 屬性。

1. 開啓代理程式的實例配置檔案。您可以在以下位置找到它：

```
IMQ_VARHOME/instances/brokerName/props/config.properties  
(在 Solaris 上為  
/var/imq/instances/brokerName/props/config.properties)
```

其中代理程式名稱爲代理程式實例的名稱。

2. 將 `ssljms` 或 `ssladmin` 值或二者的值 (取決於您需要的服務) 新增至 `imq.service.activelist` 屬性：

```
imq.service.activelist=jms,admin,httpjms,ssljms,ssladmin
```

步驟 3. 啟動代理程式

啟動代理程式，提供鍵值儲存區密碼。您可以用以下任何一種方法提供密碼：

- 當代理程式啟動時讓它提示您提供密碼

```
imqbrokerd  
Please enter Keystore password: 我的密碼
```

- 使用 `imqbrokerd` 指令的 `-password` 選項：

```
imqbrokerd -password 我的密碼
```

- 將密碼放在密碼檔案中 (請參閱第 182 頁的「使用密碼檔案」)，代理程式啟動時將存取此檔案。您必須先設定以下代理程式配置屬性 (請參閱第 104 頁的「編輯實例配置檔案」)：

```
imq.passfile.enabled=true
```

一旦設定此屬性，您便可以用以下兩種方法之一存取密碼檔案：

- 將密碼檔案的位置傳送給 `imqbrokerd` 指令：

```
imqbrokerd -passfile /tmp/mypassfile
```

- 啟動不帶 `-passfile` 選項的代理程式，但使用以下兩個代理程式配置屬性指定密碼檔案的位置：

```
imq.passfile.dirpath=/tmp
```

```
imq.passfile.name=mypassfile
```

如需與密碼檔案相關的代理程式屬性的清單，請參閱第 53 頁的表格 2-6。

當您啟動具有 SSL 的代理程式或用戶端時，您可能會注意到它消耗了很多 CPU 週期，長達幾秒鐘。這是因為 MQ 使用 JSSE (Java 安全套接延伸) 以實施 SSL。JSSE 使用 `java.security.SecureRandom()` 產生隨機數字。此方法建立初始隨機數種子要佔用大量時間，這就是您所看到 CPU 使用率增加的原因。建立種子之後，CPU 使用率會降至正常。

步驟 4. 配置並執行基於 SSL 的用戶端

最後，您需要將用戶端配置為使用安全連線服務。依據您使用的連線服務，有兩種類型的用戶端：使用 `ssljms` 的 JMS 用戶端和使用 `ssladmin` 的 MQ 管理指令公用程式 (`imqcmd`)。這兩種用戶端將在以下各節中分別描述。

JMS 用戶端

您必須確保用戶端在其類別路徑中擁有必要的安全套接延伸 (JSSE) JAR 檔案，而且您需要告知用戶端使用 `ssljms` 連線服務。

1. 如果您的用戶端未使用 J2SDK1.4 (具有內建的 JSSE 與 JNDI 支援)，請確保用戶端在其類別路徑中擁有以下 JAR 檔案：

`jsse.jar`、`jnet.jar`、`jcrt.jar` 和 `jndi.jar`

2. 確保用戶端在其類別路徑中擁有以下 MQ JAR 檔案：

`imq.jar`、`jms.jar`

3. 啟動用戶端並將其連線至代理程式的 `ssljms` 服務。完成此作業的方法之一是輸入以下指令：

```
java -DimqConnectionType=TLS clientAppName
```

設定 `imqConnectionType` 可告知連線使用 SSL。

如需有關在用戶端應用程式中使用 `ssljms` 連線服務的更多資訊，請參閱「MQ 開發人員指南」中有關使用受管理物件的章節。

指令公用程式 (`imqcmd`)

您可以透過使用 `imqcmd` (請參閱第 126 頁的表格 6-2) 時包括 `-secure` 選項，建立安全管理連線，例如：

```
imqcmd list svc -b hostName:port -u adminName -p adminPassword -secure
```

其中 `adminName` 和 `adminPassword` 為 MQ 使用者儲存庫中的有效項目 (如果使用文字檔案儲存庫，請參閱第 169 頁的「變更預設管理員密碼」)。

列出連線服務 (如在此範例中) 是顯示 `ssladmin` 服務正在執行的一種方法，也是您可以成功地建立安全 `admin` 連線的一種方法，如以下輸出中所示：

```
Listing all the services on the broker specified by:
```

Host	Primary Port
localhost	7676

Service Name	Port Number	Service State
admin	33984 (dynamic)	RUNNING
httpjms	-	UNKNOWN
httpsjms	-	UNKNOWN
jms	33983 (dynamic)	RUNNING
ssladmin	35988 (dynamic)	RUNNING
ssljms	dynamic	UNKNOWN

```
Successfully listed services.
```

設定透過 HTTP 的 SSL 服務

在此基於 SSL 的連線服務 (`httpsjms`) 中，用戶端和代理程式藉由 HTTPS 通道 Servlet 建立安全連線。第 191 頁上的附錄 B 「HTTP/HTTPS 支援 (企業版)」中描述了 HTTPS 支援的架構與實施。

使用密碼檔案

當您想讓代理程式啟動而不提示您提供所需密碼，或不要求您將這些密碼作為 `imqbrokerd` 指令的選項提供時，您可以將所需密碼放在密碼檔案中。

密碼檔案為包含密碼的簡單文字檔。此檔案未加密，因此不如手動提供密碼安全。然而，您可以設定此檔案的許可權，限制何人有權檢視它。密碼檔案的許可權需要賦予啟動代理程式的使用者讀取它的許可權。

密碼檔案可以包含表格 8-9 中所示密碼：

表格 8-9 密碼檔案中的密碼

密碼	描述
imq.keystore.password	為基於 SSL 的服務指定鍵值儲存區密碼。
imq.user_repository.ldap.password	指定與識別名稱相關聯的密碼，此識別名稱指定給代理程式以連結至已配置的 LDAP 使用者儲存庫。
imq.persist.jdbc.password	如果必需，則指定用於開啓資料庫連線的密碼。

密碼檔案範例可在以下位置找到：

IMQ_HOME/etc/passfile.sample (在 Solaris 上為 /etc/imq/passfile.sample)

設定外掛持續性

本附錄介紹如何設定代理程式，以使用外掛持續性來存取 JDBC 可存取的資料儲存區。

介紹

Sun™ ONE Message Queue (MQ) 代理程式包括持續性管理程式元件，可管理永久性資訊的寫入與擷取 (請參閱第 48 頁的「持續性管理程式」)。依預設，此持續性管理程式已配置，可存取內建、基於檔案的資料儲存區，但是您可以重新配置它，以便透過 JDBC 相容的驅動程式插入任何可存取的資料儲存區。

若要配置代理程式以使用外掛持續性，您需要在代理程式實例配置檔案中設定一些與 JDBC 相關的屬性。您還需要建立適當的資料庫綱目，以執行 MQ 持續性作業。MQ 提供有公用程式，即資料庫管理員 (imqdbmgr)，此程式使用您的 JDBC 驅動程式和代理程式配置屬性以建立並管理外掛資料庫。

作為範例，本附錄中描述的程序使用 Java 2 SDK Enterprise Edition 隨附的 Cloudscape DBMS 加以描述 (J2EE SDK 可從 java.sun.com 下載獲得)。此範例使用 Cloudscape 的內嵌式版本 (並非用戶端/伺服器版本)。在程序中，說明均使用 Cloudscape 範例中的路徑名稱和屬性名稱加以描述。它們均用「範例」一詞識別：

其他範例可在以下位置找到：

`IMQ_HOME/demo/jdbc` (在 Solaris 上為 `/usr/demo/imq/jdbc`)

插入 JDBC 可存取的資料儲存區

插入 JDBC 可存取的資料儲存區僅需要幾個步驟。

► 插入 JDBC 可存取的資料儲存區

1. 在代理程式的配置檔案中設定與 JDBC 相關的屬性。

請參閱第 187 頁的表格 A-1 中提供的屬性。

2. 將您的 JDBC 驅動程式 JAR 檔案的副本或符號式連結放在以下路徑中：

IMQ_VARHOME/lib/ext (在 Solaris 上為 /usr/share/lib/imq/ext/)

副本範例 (Solaris)：

```
% cp j2sdk 安裝目錄 /lib/cloudscape/cloudscape.jar  
IMQ_VARHOME/lib/ext
```

符號式連結範例 (Solaris)：

```
% ln -s j2sdk 安裝目錄 /lib/cloudscape/cloudscape.jar  
IMQ_VARHOME/lib/ext
```

3. 建立 MQ 持續性所需的資料庫綱目。

使用 `imqdbmgr create all` 指令 (對於內嵌式資料庫) 或 `imqdbmgr create tbl` 指令 (對於外部資料庫)。請參閱第 189 頁的「資料庫管理員公用程式 (imqdbmgr)」。

範例：

```
% cd IMQ_HOME/bin (在 Solaris 上為 /usr/bin)  
% imqdbmgr create all
```

註 如果已使用內嵌式資料庫，則建議在以下目錄下建立此資料庫：

IMQ_VARHOME/instances/*brokerName*/dbstore/*databaseName*。
(在 Solaris 上為 /var/imq/instances/*brokerName*/dbstore/)

如果內嵌式資料庫未由使用者名稱和密碼保護，則可能由檔案系統許可權保護。若要確保代理程式可讀取和寫入資料庫，則執行此代理程式的使用者應與使用 `imqdbmgr` 指令 (請參閱第 189 頁的「資料庫管理員公用程式 (imqdbmgr)」) 建立內嵌式資料庫的使用者相同。

與 JDBC 相關的代理程式配置屬性

代理程式的實例配置檔案位於以下位置

```
IMQ_VARHOME/instances/brokerName/props/config.properties
(在 Solaris 上為
/var/imq/instances/brokerName/props/config.properties)
```

如果此檔案尚未存在，則您必須使用 `-name brokerName` 選項啟動代理程式，為 MQ 建立此檔案。

表格 A-1 描述了當您插入 JDBC 可存取的資料儲存區時需要設定的配置屬性。您可以使用外掛持續性在每個代理程式實例的實例配置檔案 (`config.properties`) 中設定這些屬性。此表格包括您為 Cloudscape DBMS 範例指定的值。

表格 A-1 與 JDBC 相關的屬性

屬性名稱	描述
<code>imq.persist.store</code>	指定基於檔案或基於 JDBC 的資料儲存區。 範例： <code>jdbc</code>
<code>imq.persist.jdbc.brokerid</code> (選擇性)	指定附加至資料庫表格名稱的代理程式實例識別碼，以便在多個代理程式實例使用相同的資料庫作為永久性資料儲存區的情況下，這些資料庫表格名稱均是唯一的。(由於內嵌式資料庫僅為一個代理程式實例儲存資料，因此通常不需要指定識別碼。) 識別碼必須為字母數字字串，其長度不超過資料庫所允許的最長表格名稱長度減去 12。 範例：Cloudscape 不需要
<code>imq.persist.jdbc.driver</code>	指定 JDBC 驅動程式的 Java 類別名稱，以連線至資料庫。 範例： <code>COM.cloudscape.core.JDBCdriver</code>
<code>imq.persist.jdbc.opendburl</code>	指定資料庫 URL，以開啓與現有資料庫的連線。 範例： <code>jdbc:cloudscape:IMQ_VARHOME/instances/brokerName/dbstore/imqdb</code> (在 Solaris 上為 <code>jdbc:cloudscape:var/imq...</code>)

表格 A-1 與 JDBC 相關的屬性 (續)

屬性名稱	描述
imq.persist.jdbc.createdburl (選擇性)	指定資料庫 URL，以開啓連線建立資料庫。(僅在使用 imqdbmgr 建立資料庫時指定。) 範例： jdbc:cloudscape:IMQ_VARHOME/ instances/ <i>brokerName</i> /dbstore/ imqdb;create=true (在 Solaris 上為 jdbc:cloudscape:var/imq...)
imq.persist.jdbc.closedburl (選擇性)	指定資料庫 URL，以在關閉代理程式時關閉目前資料庫連線。 範例 (Cloudscape 所必需的)： jdbc:cloudscape;;shutdown=true
imq.persist.jdbc.user (選擇性)	如果必需，則指定用於開啓資料庫連線的使用者名稱。考慮到安全性因素，可以指定此值，而不是使用指令行選項： imqbrokerd -dbuser 和 imqdbmgr -u
imq.persist.jdbc.needpassword (選擇性)	指定資料庫是否需要密碼以存取代理程式。值 true 表示需要密碼。可以使用以下指令行選項指定密碼： imqbrokerd -dbpassword imqdbmgr -p 如果未使用指令行選項或密碼檔案提供密碼 (請參閱第 182 頁的「使用密碼檔案」)，則代理程式將提示您提供密碼。
imq.persist.jdbc.password (選擇性)	如果必需，則指定用於開啓資料庫連線的密碼。僅能在密碼檔案中指定 (請參閱第 182 頁的「使用密碼檔案」)。如需更多安全性，請讓代理程式提示您提供密碼，或使用以下指令行選項指定密碼： imqbrokerd -dbpassword imqdbmgr -p

如同所有的代理程式配置屬性，可以使用 -D 指令行選項設定值。如果資料庫需要設定某些資料庫特定的屬性，則當啟動代理程式 (imqbrokerd) 或資料庫管理員公用程式 (imqdbmgr) 時，還可以使用 -D 指令行選項設定這些值。

範例：

對於 Cloudscape 內嵌式資料庫範例，並非在資料庫連線 URL (如表格 A-1 範例中所示的那些 URL) 中指定絕對路徑，`-D` 指令行選項可用於定義 Cloudscape 系統目錄：

```
-Dcloudscape.system.home=IMQ_VARHOME/instances/brokerName/dbstore
```

在這種情況下，要建立和開啓資料庫的 URL 僅可分別指定為：

```
imq.persist.jdbc.createdburl=jdbc:cloudscape:imqdb;create=true
```

和

```
imq.persist.jdbc.opendburl=jdbc:cloudscape:imqdb。
```

資料庫管理員公用程式 (imqdbmgr)

MQ 提供有資料庫管理員公用程式 (imqdbmgr)，以設定持續性所需的綱目。如果 MQ 資料庫表格損毀或者您想使用其他資料庫作為資料儲存區，此公用程式還可用於刪除這些表格。

註	如果代理程式實例異常崩潰，則未參考過的永久性資訊可能保留在資料儲存區中。啓動然後關閉代理程式將正常清除資料儲存區。
----------	---

本節描述基本的 imqdbmgr 指令語法、提供子指令的清單，並總結 imqdbmgr 指令選項。

指令語法

imqdbmgr 指令的一般語法如下：

```
imqdbmgr subcommand argument [options]
imqdbmgr -h|-help
imqdbmgr -v|-version
```

請注意，如果您指定 `-v` 或 `-h` 選項，則系統不會執行指令行上指定的子指令。例如，如果您輸入以下指令，則系統會顯示版本資訊但不會執行 `create` 子指令。

```
imqdbmgr create all -v
```

imqdbmgr 子指令

資料庫管理員公用程式 (imqdbmgr) 包括表格 A-2 中列出的子指令：

表格 A-2 imqdbmgr 子指令

子指令和引數	描述
create all	建立新資料庫和 MQ 永久性儲存綱目。此指令可在內嵌式資料庫系統上使用，當使用此指令時，需要指定屬性 <code>imq.persist.jdbc.createdburl</code> 。
create tbl	在現有資料庫系統中建立 MQ 永久性儲存綱目。此指令可在外部資料庫系統上使用。
delete tbl	在目前永久性儲存資料庫中刪除現有 MQ 資料庫表格。
recreate tbl	在目前永久性儲存資料庫中刪除現有 MQ 資料庫表格，然後重新建立 MQ 永久性儲存綱目。

imqdbmgr 指令選項摘要

表格 A-3 列出了 imqdbmgr 指令的選項。

表格 A-3 imqdbmgr 選項

選項	描述
-D <i>property=value</i>	將指定屬性設定為指定值。
-b <i>brokerName</i>	指定代理程式實例名稱並使用相應的實例配置檔案。
-h	顯示用法說明。不執行指令行上的其他任何指令。
-p <i>password</i>	指定資料庫密碼。
-u <i>name</i>	指定資料庫使用者名稱。
-v	顯示版本資訊。不執行指令行上的其他任何指令。

HTTP/HTTPS 支援 (企業版)

Sun™ ONE Message Queue (MQ) 企業版 (請參閱第 24 頁的「產品版本」) 包括對 HTTP 連線和 HTTPS 連線的支援。(HTTPS 是透過 HTTP 的安全連線，並使用安全套接層標準。) 此支援可讓用戶端應用程式使用 HTTP 協定 (並非直接的 TCP 連線) 與代理程式通訊。本附錄描述用於實施此支援的架構，並介紹允許用戶端使用基於 HTTP 的連線進行 MQ 訊息傳送所需的設定工作。

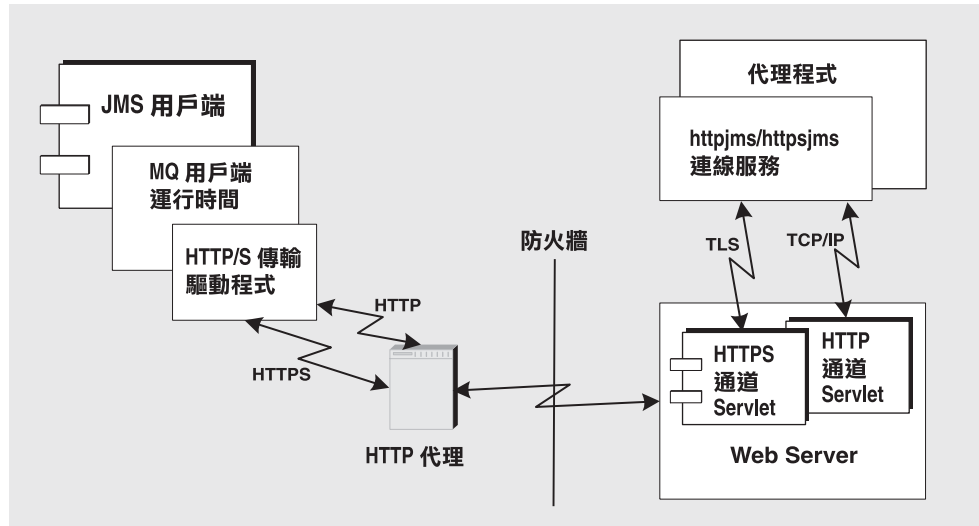
HTTP/HTTPS 支援架構

MQ 訊息傳送可在 HTTP/HTTPS 連線的頂層執行。由於通常可允許 HTTP/HTTPS 連線通過防火牆，因此可允許透過防火牆將用戶端應用程式與代理程式分隔。

第 192 頁的圖 B-1 顯示提供 HTTP/HTTPS 支援所需的主要元件。

- 在用戶端，HTTP 傳輸驅動程式將 MQ 訊息封裝在 HTTP 請求中，並確保這些請求以正確的序列傳送至 Web Server。
- 如有必要，JMS 用戶端可使用 HTTP 代理伺服器與代理程式通訊。代理的位址在啟動用戶端時使用指令行選項指定。請參閱第 196 頁的「使用 HTTP 代理」，以獲得更多資訊。
- HTTP 或 HTTPS 通道 Servlet (均隨附於 MQ) 被載入 Web Server 中，並用於在將 JMS 訊息轉寄至代理程式之前，將它們從用戶端 HTTP 請求中移除。HTTP/HTTPS 通道 Servlet 還將代理程式訊息傳送回用戶端，以回應用戶端發出的 HTTP 請求。單一 HTTP/HTTPS 通道 Servlet 可用於存取多重代理程式。

圖 B-1 HTTP/HTTPS 支援架構



- 在代理程式端，httpjms 或 httpsjms 連線服務展開並多路傳輸從相應通道 Servlet 進來的訊息。
- 如果 Web Server 發生故障並重新啓動，則所有連線均會復原且不影響用戶端。如果代理程式發生故障並重新啓動，則系統會拋出異常，且用戶端必須重新建立它們的連線。如果 Web Server 與代理程式均發生故障，且未重新啓動代理程式（這種情況不太可能發生），則 Web Server 會復原用戶端連線並繼續等待代理程式連線，而不通知用戶端。若要避免這種情況，請務必重新啓動代理程式。

如圖 B-1 中所示，HTTP 和 HTTPS 支援的架構非常相似。主要區別在於，對於 HTTPS (httpsjms 連線服務)，通道 Servlet 具有與用戶端應用程式和代理程式的安全連線。

可透過已啓用 SSL 的通道 Servlet (MQ 的 HTTPS 通道 Servlet) 提供與代理程式的安全連線，此通道 Servlet 將自身簽名的證書傳送至請求連線的任何代理程式。代理程式使用此證書來設定與 HTTPS 通道 Servlet 的加密連線。一旦建立此連線，用戶端應用程式與通道 Servlet 之間的安全連線可由用戶端應用程式和 Web Server 協商。

實施 HTTP 支援

以下各節描述實施 HTTP 支援所需採取的步驟。

► 實施 HTTP 支援

1. 在 Web Server 上佈署 HTTP 通道 Servlet。
2. 配置代理程式的 `httpjms` 連線服務並啟動代理程式。
3. 配置 HTTP 連線。

步驟 1. 在 Web Server 上佈署 HTTP 通道 Servlet

在 Web Server 上佈署 HTTP 通道 Servlet 有兩種常用方法：

- 將其作為 JAR 檔案佈署 — 用於支援 Servlet 2.1 或較低版本的 Web Server
- 將其作為 Web 歸檔 (WAR) 檔案佈署 — 用於支援 Servlet 2.2 或較高版本的 Web Server

作為 JAR 檔案佈署

佈署 MQ 通道 Servlet 由兩部分組成：使主機 Web Server 存取適當的 JAR 檔案，然後配置此 Web Server 以在啟動時載入 Servlet。

通道 Servlet JAR 檔案 (`imqservlet.jar`) 包含 HTTP 通道 Servlet 所需的所有類別，並且此檔案位於以下目錄：

`IMQ_HOME/lib` (在 Solaris 上為 `/usr/share/lib/imq`)

支援 Servlet 2.x 的任何 Web Server 均可用於載入此 Servlet。Servlet 類別名稱為：

```
com.sun.messaging.jmq.transport.  
httptunnel.servlet.HttpTunnelServlet
```

Web Server 必須可參閱 `imqservlet.jar` 檔案。如果您在不同的主機上執行 Web Server 和代理程式，則應將 `imqservlet.jar` 檔案的副本放在 Web Server 可存取的位置。

您還需要配置 Web Server，以在啟動時載入此 Servlet (請參閱第 197 頁的「範例：佈署 HTTP 通道 Servlet」)。

為了提昇效能，我們還建議您停用 Web Server 的存取記錄功能。

作為 Web 歸檔檔案佈署

將 HTTP 通道 Servlet 作為 WAR 檔案佈署需要使用 Web Server 提供的佈署機制。HTTP 通道 Servlet WAR 檔案 (imqhttp.war) 位於以下目錄：

IMQ_HOME/lib (在 Solaris 上為 /usr/share/lib/imq)

WAR 檔案包括佈署描述元，此描述元包含 Web Server 載入和執行 Servlet 所需的基本配置資訊。

步驟 2. 配置 httpjms 連線服務

依預設，HTTP 支援並未為 MQ 3.0.1 代理程式而啟用，因此您需要重新配置此代理程式以啟用 httpjms 連線服務。一旦重新配置，此代理程式便可啟動，如第 108 頁的「啟動代理程式」中所述。

► 啟用 httpjms 連線服務

1. 開啟位於以下位置的代理程式實例配置檔案：

```
IMQ_VARHOME/instances/brokerName/props/config.properties  
(在 Solaris 上為  
/var/imq/instances/brokerName/props/config.properties)
```

其中 *brokerName* 為代理程式實例的名稱。

2. 將 httpjms 值新增至 imq.service.activelist 屬性：

```
imq.service.activelist=jms,admin,httpjms
```

啟動時，代理程式會查找在其主機上執行的 Web Server 和 HTTP 通道 Servlet。但是，若要存取遠端通道 Servlet，您可以重新配置 servletHost 和 servletPort 連線服務屬性。

您還可以重新配置 pullPeriod 屬性，以提昇效能。第 195 頁的表格 B-1 中詳細介紹了 httpjms 連線服務配置屬性。

表格 B-1 httpjms 連線服務屬性

屬性名稱	描述
imq.httpjms.http.servletHost	如有必要，請變更此值，以指定 HTTP 通道 Servlet 正在其上執行的主機 (主機名稱或 IP 位址) 的名稱。(這可以為遠端主機，或本地主機上的特定主機名稱。) 預設：localhost
imq.httpjms.http.servletPort	變更此值，以指定代理程式用於存取 HTTP 通道 Servlet 的連接埠號。(如果 Web Server 上的預設連接埠已變更，則您必須相應地變更此屬性。) 預設：7675
imq.httpjms.http.pullPeriod	指定每個用戶端發出可從代理程式中移除訊息的 HTTP 請求的間隔時間 (以秒為單位)。如果值為零或負數，則用戶端將始終擱置一個 HTTP 請求，以準備儘快地移除訊息。由於用戶端數量較大，這可能會導致 Web Server 資源大量外流，並且伺服器可能沒有回應。在此類情況下，您應將 pullPeriod 屬性設定為正的秒數。此作業將設定用戶端的 HTTP 傳輸驅動程式在發出後續移除請求之前等待的時間。將此值設定為正數可保存 Web Server 資源，但會影響用戶端所監視的回應次數。預設：-1

步驟 3. 配置 HTTP 連線

用戶端應用程式必須使用已適當配置的連線工廠受管理物件，以建立與代理程式的 HTTP 連線。本節說明 HTTP 連線配置問題。

設定連線工廠屬性

若要實施 HTTP 支援，您應設定以下連線工廠屬性 (請參閱第 150 頁的「連線工廠受管理物件」)：

- 將 imqConnectionType 屬性設定為 HTTP
- 將 imqConnectionURL 設定為 HTTP 通道 Servlet URL

`http://主機名稱:連接埠/imq/tunnel`

您可以透過以下方法之一設定連線工廠屬性：

- 使用可建立連線工廠受管理物件 (請參閱第 157 頁的「新增連線工廠」) 之 imqobjmgr 指令的 -o 選項，或在使用管理主控台 (imqadmin) 建立連線工廠受管理物件時設定屬性。
- 使用可啟動用戶端應用程式之指令的 -D 選項 (請參閱「MQ 開發人員指南」)。

- 當您有計劃地藉由用戶端應用程式碼建立連線工廠後，請使用 JMS API 呼叫設定它的屬性 (請參閱「MQ 開發人員指南」)。

使用單一 Servlet 存取多重代理程式

如果您要執行多重代理程式，則無需配置多重 Web Server 和 Servlet 實例。您可以在同時執行的代理程式中共用單一 Web Server 和 HTTP 通道 Servlet 實例。為了執行此作業，您必須配置 `imqConnectionURL` 連線工廠屬性，如下所示：

```
http://hostName:port/imq/tunnel?ServerName=hostName:brokerName
```

其中 *hostName* 為代理程式主機名稱，*brokerName* 為您希望用戶端存取的特定代理程式實例的名稱。

若要檢查您是否已為 *hostName* 和 *brokerName* 輸入正確的字串，則應透過從瀏覽器存取 Servlet URL 產生 HTTP 通道 Servlet 的狀態報告。此報告將列出 Servlet 要存取的所有代理程式：

```
HTTP tunnel servlet ready.
Servlet Start Time : Thu May 30 01:08:18 PDT 2002
Accepting TCP connections from brokers on port : 7675
Total available brokers = 2
Broker List :
    jpgserv:broker2
    cochin:broker1
```

使用 HTTP 代理

如果您要使用 HTTP 代理存取 HTTP 通道 Servlet：

- 將 `http.proxyHost` 系統屬性設定為代理伺服器主機名稱。
- 將 `http.proxyPort` 系統屬性設定為代理伺服器連接埠號。

您可以使用可啟動用戶端應用程式之指令的 `-D` 選項來設定這些屬性。

範例：佈署 HTTP 通道 Servlet

本節描述如何在 Sun ONE WEB Server 上將 HTTP 通道 Servlet 作為 JAR 檔案和 WAR 檔案進行佈署。您使用的方法取決於 Sun ONE Web Server 的版本：如果它不支援 Servlet 2.2 或較高版本，它將無法處理 WAR 檔案佈署。

作為 JAR 檔案佈署

以下說明參考了使用基於瀏覽器的管理 GUI 之 Sun ONE Web Server FastTrack Edition 4.1 上的佈署。此程序由以下常用步驟組成：

1. 新增 Servlet
2. 配置 Servlet 虛擬路徑
3. 載入 Servlet
4. 停用 Servlet 存取記錄

這些步驟將在以下各小節中描述。您可以透過使用 Web 瀏覽器存取 Servlet URL 來驗證成功的 HTTP 通道 Servlet 佈署。它應顯示狀態資訊。

新增 Servlet

► 新增通道 Servlet

1. 選取 [Servlets] 標籤。
2. 選擇 [配置 Servlet 屬性]。
3. 在 [Servlet 名稱] 欄位中，為通道 Servlet 指定名稱。
4. 將 [Servlet 程式碼 (類別名稱)] 欄位設定為以下值：

```
com.sun.messaging.jmq.transport.  
httpunnel.servlet.HttpTunnelServlet
```

5. 在 [Servlet 類別路徑] 欄位中輸入 imqervlet.jar 的完整路徑。例如：

```
IMQ_HOME/lib/imqervlet.jar  
(在 Solaris 上為 /usr/share/lib/imq/imqervlet.jar)
```

6. 在 [Servlet 引數] 欄位中，輸入任何選擇性引數，如**表格 B-2** 中所示：

表格 B-2 用於佈署 HTTP 通道 Servlet JAR 檔案的 Servlet 引數

引數	預設值	參考
servletHost	所有主機	請參閱 第 195 頁的表格 B-1
servletPort	7675	請參閱 第 195 頁的表格 B-1

如果要使用這兩個引數，請使用逗號將它們分隔：

servletPort= 連接埠號 , servletHost=...

serverHost 和 serverPort 引數僅適用於 Web Server 和代理程式之間的通訊，並且僅在預設值有問題時才設定。但是，在此情況下，您還必須相應地設定代理程式配置屬性 (請參閱**第 195 頁的表格 B-1**)，例如：

imq.httpjms.http.servletPort

配置 Servlet 虛擬路徑 (Servlet URL)

► 為通道 Servlet 配置虛擬路徑 (Servlet URL)

1. 選取 [Servlets] 標籤。
2. 選擇 [配置 Servlet 虛擬路徑轉換]。
3. 設定 [虛擬路徑] 欄位。

例如，如果您希望 URL 為 `http://hostName:port/imq/tunnel`，請在 [虛擬路徑] 欄位中輸入以下字串。

`/imq/tunnel`

4. 將 [Servlet 名稱] 欄位設定為與**第 197 頁的「新增 Servlet」**的步驟 3 中相同的值。

載入 Servlet

► 在啟動 Web Server 時載入通道 Servlet

1. 選取 [Servlets] 標籤。
2. 選擇 [配置全域屬性]。
3. 在 [啟動 Servlets] 欄位中，輸入與**第 197 頁的「新增 Servlet」**的步驟 3 中相同的 Servlet 名稱值。

停用伺服器存取記錄

您不必停用伺服器存取記錄，但是如果您停用記錄，將會獲得更好的效能。

► 停用伺服器存取記錄

1. 選取 [狀態] 標籤。
2. 選擇 [記錄個人喜好頁面]。
3. 使用記錄用戶端存取控制來停用記錄

作為 WAR 檔案佈署

以下說明參考了 Sun ONE Web Server 6.0 Service Pack 2 上的佈署。您可以透過使用 Web 瀏覽器存取 Servlet URL 來驗證成功的 HTTP 通道 Servlet 佈署。它應顯示狀態資訊。

► 將 HTTP 通道 Servlet 作為 WAR 檔案佈署

1. 在基於瀏覽器的管理 GUI 中，選取 [虛擬伺服器類別] 標籤並選取 [管理類別]。
2. 選取適當的虛擬伺服器類別名稱 (如預設類別)，然後按一下 [管理] 按鈕。
3. 選取 [管理虛擬伺服器]。
4. 選取適當的虛擬伺服器名稱，然後按一下 [管理] 按鈕。
5. 選取 [Web 應用程式] 標籤。
6. 按一下 [佈署 Web 應用程式]。
7. 為 [WAR 檔案開啓] 和 [WAR 檔案路徑] 欄位選取適當的值，以便指向 imqhttp.war 檔案。此檔案可在以下目錄中找到：
IMQ_HOME/lib (在 Solaris 上為 /usr/share/lib/imq)
8. 在 [應用程式 URI] 欄位中，輸入 「/imq」 (無引號)。
9. 輸入應佈署 Servlet 的安裝目錄路徑 (通常位於 Sun ONE Web Server 安裝根目錄下)。
10. 按一下 OK。
11. 重新啟動 Web Server 實例。

此時，Servlet 可從以下位址獲得：

```
http://hostName:port/imq/tunnel
```

此時，用戶端可使用此 URL，以使用 HTTP 連線而連線至訊息服務。

實施 HTTPS 支援

以下各節描述實施 HTTPS 支援所需採取的步驟。它們與第 193 頁的「實施 HTTP 支援」中的那些步驟，以及產生和存取 SSL 證書所需的步驟相似。

► 實施 HTTPS 支援

1. 為 HTTPS 通道 Servlet 產生自身簽名的證書。
2. 在 Web Server 上佈署 HTTPS 通道 Servlet。
3. 配置代理程式的 httpsjms 連線服務並啟動代理程式。
4. 配置 HTTPS 連線。

以下各節中將更加詳細地說明這些步驟中的每一步。

步驟 1. 為 HTTPS 通道 Servlet 產生自身簽名的證書

在假設用戶端正與已知且可信任的伺服器通訊的情況下，MQ 3.0.1 中的 SSL 支援可確保線上資料的安全。因此在 MQ 3.0.1 中，僅使用自身簽名的伺服器證書即可實施 SSL。在 httpsjms 連線服務架構中，HTTPS 通道 Servlet 是代理程式與應用程式用戶端的伺服器。

執行 `imkeytool` 公用程式，以便為通道 Servlet 產生自身簽名的證書。在指令提示符號下輸入以下內容：

```
imkeytool -servlet 鍵值儲存區位置
```

此公用程式將提示您提供它所需的資訊。(在 Unix 系統上，為了擁有建立鍵值儲存區的許可權，您可能需要以超級使用者 [root] 的身份執行 `imkeytool`。)

首先，`imkeytool` 提示您提供鍵值儲存區密碼以及某些組織資訊，然後提示您確認。接收到確認之後，它會在產生鍵對時暫停。然後它會要求您提供密碼，以鎖定特定的鍵對 (鍵密碼)；您應輸入 [Return] 以回應此提示：這會使鍵密碼與鍵值儲存區密碼相同。

註	請記住您所提供的密碼，稍後您需要將此密碼提供給通道 Servlet，以便它可開啓鍵值儲存區。
----------	--

執行 `imkeytool` 可執行 `JDK keytool` 公用程式，以產生自身簽名的證書，並將其放在位於**鍵值儲存區位置**引數所指定的 MQ 鍵值儲存區檔案中。(此鍵值儲存區的格式與 `JDK1.2 keytool` 支援的格式相同。)

註 HTTPS 通道 Servlet 必須可參閱此鍵值儲存區。確保您將位於**鍵值儲存區位置**中已產生的鍵值儲存區移動/複製到 HTTPS 通道 Servlet 可存取的位置 (請參閱第 201 頁的「**步驟 2. 在 Web Server 上佈署 HTTPS 通道 Servlet**」)。

步驟 2. 在 Web Server 上佈署 HTTPS 通道 Servlet

在 Web Server 上佈署 HTTPS 通道 Servlet 有兩種常用方法：

- 將其作為 JAR 檔案佈署 — 用於支援 Servlet 2.1 或較低版本的 Web Server
- 將其作為 Web 歸檔 (WAR) 檔案佈署 — 用於支援 Servlet 2.2 或較高版本的 Web Server

在這兩種情況下，您都應確保為 Web Server 啟動加密，從而讓用戶端與代理程式之間進行端對端安全通訊。

作為 JAR 檔案佈署

佈署 MQ 通道 Servlet 由兩部分組成：使主機 Web Server 存取適當的 JAR 檔案，然後配置此 Web Server 以在啟動時載入 Servlet。

通道 Servlet JAR 檔案 (`imqservlet.jar`) 包含 HTTPS 通道 Servlet 所需的所有類別，並且此檔案位於以下目錄：

`IMQ_HOME/lib` (在 Solaris 上為 `/usr/share/lib/imq`)

支援 Servlet 2.x 的任何 Web Server 均可用於載入此 Servlet。Servlet 類別名稱為：

```
com.sun.messaging.jmq.transport.  
httpunnel.servlet.HttpsTunnelServlet。
```

Web Server 必須可參閱 `imqservlet.jar` 檔案。如果您想在不同的主機上執行 Web Server 和代理程式，則應將 `imqservlet.jar` 檔案的副本放在 Web Server 可存取的位置。

您還需要配置 Web Server，以在啟動時載入此 Servlet (請參閱第 206 頁的「**範例：佈署 HTTPS 通道 Servlet**」)。

確保 JSSE JAR 檔案在類別路徑中，以便在 Web Server 中執行 Servlet。檢查 Web Server 的說明文件，以便瞭解如何執行此作業。

配置 Web Server 的一個重要方面是指定自身簽名證書的位置和密碼，以便 HTTPS 通道 Servlet 使用此證書來建立與代理程式的安全連線。您必須將第 200 頁的「步驟 1. 為 HTTPS 通道 Servlet 產生自身簽名的證書」中建立的鍵值儲存區放在 HTTPS 通道 Servlet 可存取的位置。

為了提昇效能，我們還建議您停用 Web Server 的存取記錄功能。

作為 Web 歸檔檔案佈署

將 HTTPS 通道 Servlet 作為 WAR 檔案佈署需要使用 Web Server 提供的佈署機制。HTTPS 通道 Servlet WAR 檔案 (imqhttps.war) 位於以下目錄：

IMQ_HOME/lib (在 Solaris 上為 /usr/share/lib/imq)

WAR 檔案包括佈署描述元，此描述元包含 Web Server 載入和執行 Servlet 所需的基本配置資訊。

但是，imqhttps.war 檔案的佈署描述元無法知道您已將通道 Servlet 所需的鍵值儲存區檔案放在何處 (請參閱第 200 頁的「步驟 1. 為 HTTPS 通道 Servlet 產生自身簽名的證書」)。這要求您編輯通道 Servlet 的佈署描述元 (XML 檔案)，以在佈署 imqhttps.war 檔案之前指定鍵值儲存區的位置。

步驟 3. 配置 httpsjms 連線服務

依預設，HTTPS 支援並未為 MQ 3.0.1 代理程式而啟用，因此您需要重新配置此代理程式以啟用 httpsjms 連線服務。一旦重新配置，此代理程式便可啟動，如第 108 頁的「啟動代理程式」中所述。

► 啟用 httpsjms 連線服務

1. 開啓位於以下位置的代理程式實例配置檔案：

IMQ_VARHOME/instances/*brokerName*/props/config.properties
(在 Solaris 上為
/var/imq/instances/*brokerName*/props/config.properties)

其中 *brokerName* 為代理程式實例的名稱。

2. 將 httpsjms 值新增至 imq.service.activelist 屬性：

imq.service.activelist=jms,admin,httpsjms

啓動時，代理程式會查找在其主機上執行的 Web Server 和 HTTPS 通道 Servlet。但是，若要存取遠端通道 Servlet，您可以重新配置 `servletHost` 和 `servletPort` 連線服務屬性。

您還可以重新配置 `pullPeriod` 屬性，以提昇效能。[表格 B-3](#) 中詳細介紹了 `httpsjms` 連線服務配置屬性。

表格 B-3 `httpsjms` 連線服務屬性

屬性名稱	描述
<code>imq.httpsjms.https.servletHost</code>	如有必要，請變更此值，以指定 HTTPS 通道 Servlet 正在其上執行的主機（主機名稱或 IP 位址）的名稱。（這可以為遠端主機，或本地主機上的特定主機名稱。）預設： <code>localhost</code>
<code>imq.httpsjms.https.servletPort</code>	變更此值，以指定代理程式用於存取 HTTPS 通道 Servlet 的連接埠號。（如果 Web Server 上的預設連接埠已變更，則您必須相應地變更此屬性。）預設： <code>7674</code>
<code>imq.httpsjms.https.pullPeriod</code>	指定每個用戶端發出可從代理程式中移除訊息的 HTTP 請求的間隔時間（以秒為單位）。如果值為零或負數，則用戶端將始終擱置一個 HTTP 請求，以準備儘快地移除訊息。由於用戶端數量較大，這可能會導致 Web Server 資源大量外流，並且伺服器可能沒有回應。在此類情況下，您應將 <code>pullPeriod</code> 屬性設定為正的秒數。此作業將設定用戶端的 HTTP 傳輸驅動程式在發出後續移除請求之前等待的時間。將此值設定為正數可保存 Web Server 資源，但會影響用戶端所監視的回應次數。預設： <code>-1</code>

步驟 4. 配置 HTTPS 連線

用戶端應用程式必須使用已適當配置的連線工廠受管理物件，以建立與代理程式的 HTTPS 連線。

但是，用戶端還必須對 Java 安全套接延伸 (JSSE) 提供的 SSL 程式庫擁有存取權，並且還必須具有超級使用者證書。SSL 程式庫隨附於 JDK 1.4。如果您具有較低的 JDK 版本，請參閱「[配置 JSSE](#)」，否則請繼續執行「[匯入超級使用者證書](#)」。

一旦解決這些問題，您可以繼續配置 HTTPS 連線。

配置 JSSE

► 配置 JSSE

1. 將 JSSE JAR 檔案複製到 JRE_HOME/lib/ext 目錄。

jsse.jar、jnet.jar 和 jcert.jar

2. 靜態新增 JSSE 安全性供應程式，方法為透過將

`security.provider.n=com.sun.net.ssl.internal.ssl.Provider`

新增至 JRE_HOME/lib/security/java.security 檔案 (其中 *n* 為安全性供應程式套裝軟體的下一個可用優先順序號)。

3. 如果未使用 JDK1.4，則您需要使用可啟動用戶端應用程式之指令的 -D 選項，來設定以下 JSSE 屬性：

`java.protocol.handler.pkgs=com.sun.net.ssl.internal.www.protocol`

匯入超級使用者證書

如果 CA (為您的 Web Server 證書簽名) 的超級使用者證書依預設不在信任資料庫中，或者您使用的是專用 Web Server 證書，則您必須將此證書新增至信任資料庫。如果出現此情況，請遵循以下說明，否則請前往「[設定連線工廠屬性](#)」。

假設此證書儲存在 *cert_file* 中，並且 *trust_store_file* 為您的鍵值儲存區，請執行以下指令：

```
JRE_HOME/bin/keytool -import -trustcacerts
-alias alias_for_certificate -file cert_file
-keystore trust_store_file
```

對以下問題回答 YES：信任此證書嗎？

您還需要使用可啟動用戶端應用程式之指令的 -D 選項，來指定以下 JSSE 屬性：

```
javax.net.ssl.trustStore=trust_store_file
javax.net.ssl.trustStorePassword=trust_store_passwd
```

設定連線工廠屬性

若要實施 HTTPS 支援，您應設定以下連線工廠屬性 (請參閱第 150 頁的「連線工廠受管理物件」)：

- 將 `imqConnectionType` 屬性設定為 HTTP
透過 HTTPS 通道 Servlet (而不是 HTTP 通道 Servlet) 進行佈署和連線，可建立與代理程式的安全連線。但是，兩者均使用相同的連線類型。

- 將 `imqConnectionURL` 設定為 HTTPS 通道 Servlet URL

```
https://hostName:port/imq/tunnel
```

您可以透過以下方法之一設定連線工廠屬性：

- 使用可建立連線工廠受管理物件 (請參閱第 157 頁的「新增連線工廠」) 之 `imqobjmgr` 指令的 `-o` 選項，或在使用管理主控台 (`imqadmin`) 建立連線工廠受管理物件時設定屬性。
- 使用可啟動用戶端應用程式的指令的 `-D` 選項 (請參閱「MQ 開發人員指南」)。
- 當您有計劃地藉由用戶端應用程式碼建立連線工廠後，請使用 JMS API 呼叫設定它的屬性 (請參閱「MQ 開發人員指南」)。

使用單一 Servlet 存取多重代理程式

如果您要執行多重代理程式，則無需配置多重 Web Server 和 Servlet 實例。您可以在同時執行的代理程式中共用單一 Web Server 和 HTTPS 通道 Servlet 實例。為了執行此作業，您必須配置 `imqConnectionURL` 連線工廠屬性，如下所示：

```
https://hostName:port/imp/tunnel?ServerName=hostName:brokerName
```

其中 `hostName` 為代理程式主機名稱，`brokerName` 為您希望用戶端存取的特定代理程式實例的名稱。

若要檢查您是否已為 `hostName` 和 `brokerName` 輸入正確的字串，則應透過從瀏覽器存取 Servlet URL 產生 HTTPS 通道 Servlet 的狀態報告。此報告將列出 Servlet 要存取的所有代理程式：

```
HTTPS tunnel servlet ready.
Servlet Start Time : Thu May 30 01:08:18 PDT 2002
Accepting TCP connections from brokers on port : 7674
Total available brokers = 2
Broker List :
  jpgserv:broker2
  cochinchin:broker1
```

使用 HTTP 代理

如果您要使用 HTTP 代理存取 HTTPS 通道 Servlet：

- 將 `http.proxyHost` 系統屬性設定為代理伺服器主機名稱。
- 將 `http.proxyPort` 系統屬性設定為代理伺服器連接埠號。

您可以使用可啟動用戶端應用程式之指令的 `-D` 選項來設定這些屬性。

範例：佈署 HTTPS 通道 Servlet

本節描述如何在 Sun ONE Web Server 上將 HTTPS 通道 Servlet 作為 JAR 檔案和 WAR 檔案進行佈署。您使用的方法取決於 Sun ONE Web Server 的版本：如果它不支援 Servlet 2.2 或較高版本，它將無法處理 WAR 檔案佈署。

作為 JAR 檔案佈署

以下說明參考了使用基於瀏覽器的管理 GUI 之 Sun ONE Web Server FastTrack Edition 4.1 上的佈署。此程序由以下常用步驟組成：

1. 新增 Servlet
2. 配置 Servlet 虛擬路徑
3. 載入 Servlet
4. 停用 Servlet 存取記錄

這些步驟將在以下各小節中描述。您可以透過使用 Web 瀏覽器存取 Servlet URL 來驗證成功的 HTTP 通道 Servlet 佈署。它應顯示狀態資訊。

新增 Servlet

► 新增通道 Servlet

1. 選取 [Servlets] 標籤。
2. 選擇 [配置 Servlet 屬性]。
3. 在 [Servlet 名稱] 欄位中，為通道 Servlet 指定名稱。
4. 將 [Servlet 程式碼 (類別名稱)] 欄位設定為以下值：

```
com.sun.messaging.jmq.transport.  
httptunnel.servlet.HttpsTunnelServlet
```

5. 在 [Servlet 類別路徑] 欄位中輸入 `imqervlet.jar` 的完整路徑。例如：
- `IMQ_HOME/lib/imqervlet.jar`
(在 Solaris 上為 `/usr/share/lib/imq/imqervlet.jar`)
6. 在 [Servlet 引數] 欄位中，輸入所需引數和選擇性引數，如**表格 B-4** 中所示。

表格 B-4 用於佈署 HTTPS 通道 Servlet JAR 檔案的 Servlet 引數

引數	預設值	需要嗎？	另請參閱
<code>keystoreLocation</code>	無	需要	第 179 頁的表格 8-8
<code>keystorePassword</code>	無	需要	第 179 頁的表格 8-8
<code>serverHost</code>	所有主機	不需要	第 203 頁的表格 B-3
<code>serverPort</code>	7674	不需要	第 203 頁的表格 B-3

使用逗號將引數分隔，例如：

`keystoreLocation=鍵值儲存區位置,keystorePassword=鍵值儲存區密碼,`
`serverPort=連接埠號`

`serverHost` 和 `serverPort` 引數僅適用於 Web Server 和代理程式之間的通訊，並且僅在預設值有問題時才設定。但是，在此情況下，您還必須相應地設定代理程式配置屬性 (請參閱**第 203 頁的表格 B-3**)，例如：

`imq.httpsjms.https.servletPort`

配置 Servlet 虛擬路徑 (Servlet URL)

► 為通道 Servlet 配置虛擬路徑 (Servlet URL)

1. 選取 [Servlets] 標籤。
2. 選擇 [配置 Servlet 虛擬路徑轉換]。
3. 設定 [虛擬路徑] 欄位。

例如，如果您希望 URL 為 `http://hostName:port/imq/tunnel`，請在 [虛擬路徑] 欄位中輸入以下字串。

`/imq/tunnel`

4. 將 [Servlet 名稱] 欄位設定為與**第 206 頁的「新增 Servlet」**的步驟 3 中相同的值。

載入 Servlet

► 在啟動 Web Server 時載入通道 Servlet

1. 選取 [Servlets] 標籤。
2. 選擇 [配置全域屬性]。
3. 在 [啟動 Servlets] 欄位中，輸入與第 206 頁的「新增 Servlet」的步驟 3 中相同的 Servlet 名稱值。

停用伺服器存取記錄

您不必停用伺服器存取記錄，但是如果您停用記錄，將會獲得更好的效能。

► 停用伺服器存取記錄

1. 選取 [狀態] 標籤。
2. 選擇 [記錄個人喜好頁面]。
3. 使用記錄用戶端存取控制來停用記錄。

作為 WAR 檔案佈署

以下說明參考了 Sun ONE Web Server 6.0 Service Pack 2 上的佈署。您可以透過使用 Web 瀏覽器存取 Servlet URL 來驗證成功的 HTTPS 通道 Servlet 佈署。它應顯示狀態資訊。

在佈署 HTTPS 通道 Servlet 之前，請確保 JSSE JAR 檔案包含在 Web Server 的類別路徑中。執行此作業的最簡單方法為將 jsse.jar、jnet.jar 和 jcert.jar 複製到 IWS60_TOPDIR/bin/https/jre/lib/ext。

此外，在佈署 HTTPS 通道 Servlet 之前，您必須修改其佈署描述元，以指向您放置鍵值儲存區檔案的位置，並指定鍵值儲存區密碼。

► 修改 HTTPS 通道 Servlet WAR 檔案

1. 將 WAR 檔案複製到暫存目錄。

```
$ cp IMQ_HOME/lib/imaghttps.war /tmp  
(在 Solaris 上為 $ cp /usr/share/lib/imag/imaghttps.war /tmp)
```

2. 將此暫存目錄變為您的目前目錄。

```
$ cd /tmp
```

3. 擷取 WAR 檔案的內容。

```
$ jar xvf imaghttps.war
```

4. 列出 WAR 檔案的佈署描述元。

```
$ ls -l WEB-INF/web.xml
```

5. 編輯 web.xml 檔案，以便為 keystoreLocation 和 keystorePassword 引數 (如有必要，還包括 serverPort 和 serverHost 引數) 提供正確的值。
6. 重新組合 WAR 檔案的內容。

```
$ jar uvf imqhttps.war WEB-INF/web.xml
```

此時，您可以使用已修改的 imqhttps.war 檔案來佈署 HTTPS 通道 Servlet。(如果您擔心鍵值儲存區密碼洩漏，則可以使用檔案系統許可權來限制他人對 imqhttps.war 檔案的存取。)

► 將 HTTPS 通道 Servlet 作為 WAR 檔案佈署

1. 在基於瀏覽器的管理 GUI 中，選取 [虛擬伺服器類別] 標籤。按一下 [管理類別]。
2. 選取適當的虛擬伺服器類別名稱 (如預設類別)，然後按一下 [管理] 按鈕。
3. 選取 [管理虛擬伺服器]。
4. 選取適當的虛擬伺服器名稱，然後按一下 [管理] 按鈕。
5. 選取 [Web 應用程式] 標籤。
6. 按一下 [佈署 Web 應用程式]。
7. 為 [WAR 檔案開啓] 和 [WAR 檔案路徑] 欄位選取適當的值，以便指向已修改的 imqhttps.war 檔案 (請參閱第 208 頁的「[修改 HTTPS 通道 Servlet WAR 檔案](#)」)。
8. 在 [應用程式 URI] 欄位中，輸入「/imq」(無引號)。
9. 輸入應佈署 Servlet 的安裝目錄路徑 (通常位於 Sun ONE Web Server 安裝根目錄下)。
10. 按一下 OK。
11. 重新啓動 Web Server 實例。

此時，Servlet 可從以下位址獲得：

```
https://hostName:port/imq/tunnel
```

此時，用戶端可使用此 URL，以使用安全 HTTPS 連線而連線至訊息服務。

將代理程式用作 Windows 服務

本附錄介紹如何使用服務管理員 (imqsvcadmin) 公用程式來安裝、查詢和移除作為 Windows 服務執行的代理程式。

將代理程式作為 Windows 服務執行

當您安裝 Sun™ ONE Message Queue (MQ) 3.0.1 時，您可以選擇將代理程式作為 Windows 服務安裝。在安裝 MQ 3.0.1 之後，您還可以使用 imqsvcadmin 將代理程式作為 Windows 服務安裝。

將代理程式作為 Windows 服務安裝，意味著此代理程式將在系統啟動時啟動，並在後台執行直至關機。因此，您不必使用 imqbrokerd 指令啟動代理程式，除非您要啟動其他實例。若要將所有啟動選項傳送至代理程式，您可以使用 imqsvcadmin 指令 (請參閱第 213 頁的表格 C-2) 的 -args 引數，並確切地指定您已為 imqbrokerd 指令 (請參閱第 108 頁的「啟動代理程式」) 使用的相同選項。使用 imqcmd 指令以控制代理程式照常作業。

當代理程式作為 Windows 服務執行時，[工作管理員] 會將代理程式作為兩個可執行程序列出。第一個程序是 imqbrokersvc.exe，這是原生 Windows 服務包裝。第二個程序是實際執行代理程式的 Java 運行時間。

每次只能將一個代理程式作為 Windows 服務安裝並執行。

服務管理員公用程式 (imqsvcadmin)

服務管理員公用程式 (imqsvcadmin) 可讓您安裝、查詢和移除代理程式 (作為 Windows 服務執行)。本節描述 imqsvcadmin 指令的基本語法、提供子指令的清單、總結 imqsvcadmin 指令選項，並介紹如何使用這些指令來完成特定的工作。

指令的語法

imqsvcadmin 指令的一般語法如下：

```
imqsvcadmin subcommand [options]
imqsvcadmin -h
```

請注意，如果您指定 -v、-h、或 -H 選項，則系統不會執行指令行上指定的其他子指令。例如，如果您輸入以下指令，則系統會顯示說明資訊但不會執行 query 子指令。

```
imqsvcadmin query -h
```

imqsvcadmin 子指令

MQ 服務管理員公用程式 (imqsvcadmin) 包括**表格 C-1** 中列出的子指令：

表格 C-1 imqsvcadmin 子指令	
子指令	描述
install	安裝服務並指定啓動選項。
query	顯示 imqsvcadmin 指令的啓動選項。這包括服務是手動啓動還是自動啓動、服務的位置、Java 運行時間的位置以及啓動時傳送至代理程式的引數的值。
remove	移除服務。

imqsvcadmin 選項摘要

表格 C-2 列出 imqsvcadmin 指令的選項。如需它們用法的說明，請參閱以下基於工作的各節。

表格 C-2 imqsvcadmin 選項

選項	描述
-h	顯示用法說明。不執行指令行上的其他任何指令。
-javahome <i>path</i>	指定替代 Java 2 相容的 JDK 的路徑。預設為使用隨附的運行時間。 範例：imqsvcadmin -install -javahome d:\jdk1.4
-jrehome <i>path</i>	指定 Java 2 相容的 JRE 的路徑。 範例：imqsvcadmin -install -jrehome d:\jre\1.4
-vmargs <i>arg</i> [[<i>arg</i>] ...]	指定要傳送至正在執行代理程式服務的 Java VM 的其他引數。(您還可以在 [Windows 服務控制台] 的 [啟動參數] 欄位中指定這些引數。) 範例：-vmargs "-Xms16m -Xmx128m"
-args <i>arg</i> [[<i>arg</i>] ...]	指定要傳送至代理程式服務的其他指令行引數。如需 imqbrokerd 選項的描述，請參閱第 108 頁的「啟動代理程式」。 (您還可以在 [Windows 服務控制台] 的 [啟動參數] 欄位中指定這些引數。) 例如， imqsvcadmin -install -args "-passfile d:\imqpassfile"

您使用 -javahome、-vmargs 和 -args 選項指定的資訊，儲存在 Window 登入檔中的關鍵字 JavaHome、JVMArgs 和 ServiceArgs 下，路徑為

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet
  \Services\iMQ_Broker\Parameters
```

移除代理程式服務

在移除代理程式服務之前，您應使用 `imqcmd shutdown bkr` 指令關閉代理程式。然後使用 `imqsvcadmin remove` 指令移除服務，並重新啟動電腦。

重新配置代理程式服務

若要重新配置服務，請先移除此服務，然後將其重新安裝，並指定帶有 `-args` 引數的不同啟動選項。

使用替代 Java 運行時間

您可以使用 `-javahome` 或 `-jrehome` 選項來指定替代 Java 運行時間的位置。您還可以在 [Windows 服務控制台] 的 [啟動參數] 欄位中指定這些選項。請注意，[啟動參數] 欄位將反斜線 (\) 看作一個退出字元，因此當將反斜線用作路徑分割元時，您必須將其鍵入兩次；例如，`-javahome d:\\jdk1.3`。

查詢代理程式服務

若要確定代理程式服務的啟動選項，請使用 `imqsvcadmin` 指令的 `-q` 選項。

```
imqsvcadmin -query

Service iMQ_Broker is installed.
Display Name: iMQ_Broker
Start Type: Manual
Binary location: c:\Program Files\Sun Microsystems\
                  Message Queue 3.0\bin\imqbrokersvc
JavaHome: c:\j2sdk1.4.0
Broker Args: -passfile d:\imqpassfile
```

疑難排解

如果嘗試啟動服務時出現錯誤，您可以查閱透過執行以下作業記錄的錯誤事件。

► **查閱記錄的服務錯誤事件**

1. 啟動 [事件檢視器]
2. 查看 [日誌] > [應用程式]。
3. 選取 [檢視] > [重新顯示]，以查閱所有錯誤事件。

MQ 資料的位置

Sun™ ONE Message Queue (MQ) 使用多種資料，每種資料都儲存在不同的位置，如表格 D-1 中所示。

表格 D-1 MQ 3.0 資料的位置

資料種類	資料的位置
代理程式實例配置屬性	IMQ_VARHOME/instances/brokerName/props/config.properties (在 Solaris 上為 /var/imq/instances/brokerName/props/config.properties)
永久性儲存區 (訊息、目標、持久訂閱和異動)	IMQ_VARHOME/instances/brokerName/filestore/ (在 Solaris 上為 /var/imq/instances/brokerName/filestore/) 或 JDBC 可存取的資料儲存區
代理程式實例日誌檔	IMQ_VARHOME/instances/brokerName/log/ (在 Solaris 上為 /var/imq/instances/brokerName/log/)
受管理物件(物件儲存區)	您選擇的本機目錄 或 LDAP 伺服器
安全性：使用者儲存庫	IMQ_HOME/etc/passwd (在 Solaris 上為 /etc/imq/passwd) 或 LDAP 伺服器
安全性：存取控制檔案	IMQ_HOME/etc/accesscontrol.properties (在 Solaris 上為 /etc/imq/accesscontrol.properties)

表格 D-1 MQ 3.0 資料的位置 (續)

資料種類	資料的位置
安全性：密碼檔案	IMQ_HOME/etc/passfileName (在 Solaris 上為 /etc/imq/passfileName)
安全性：代理程式的鍵值 儲存區	IMQ_HOME/etc/keystore (在 Solaris 上為 /etc/imq/keystore)

MQ 介面的穩定性

Sun™ ONE Message Queue (MQ) 使用的許多介面可能對管理員自動管理工作很有用。表格 E-1 根據這些介面的穩定程度 (即它們在以後的產品版本中不進行變更的可能性) 對其進行分類。第 221 頁的表格 E-2 中描述了分類機制。

表格 E-1 MQ 3.0.1 介面的穩定性

介面	分類
imqbrokerd 指令行介面	形成中
imqadmin 指令行介面	不穩定
imqcmd 指令行介面	形成中
imqdbmgr 指令行介面	不穩定
imqkeytool 指令行介面	形成中
imqobjmgr 指令行介面	形成中
imqusermgr 指令行介面	不穩定
imqobjmgr 指令檔案	形成中
imqbrokerd 指令	穩定
imqadmin 指令	不穩定
imqcmd 指令	穩定
imqdbmgr 指令	不穩定
imqkeytool 指令	穩定
imqobjmgr 指令	穩定
imqusermgr 指令	不穩定

表格 E-1 MQ 3.0.1 介面的穩定性 (續)

介面	分類
JMS API (javax.jms)	標準
JAXM API (javax.xml)	標準
受管理物件 API (com.sun.messaging)	形成中
imq.jar 位置和名稱	穩定
jms.jar 位置和名稱	形成中
imqbroker.jar 位置和名稱	私有
imqutil.jar 位置和名稱	私有
imqadmin.jar 位置和名稱	私有
imqservlet.jar 位置和名稱	形成中
imqhttp.war 位置和名稱	形成中
imqhttps.war 位置和名稱	形成中
imqxm.jar 位置和名稱	形成中
jaxm-api.jar 位置和名稱	形成中
saaj-api.jar 位置和名稱	形成中
saaj-impl.jar 位置和名稱	形成中
activation.jar 位置和名稱	形成中
mail.jar 位置和名稱	形成中
dom4j.jar 位置和名稱	私有
fscontext.jar 位置和名稱	不穩定
從 imqbrokerd、imqadmin、imqcmd、imqdbmgr、 imqkeytool、imqobjmgr 和 imqusermgr 的輸出	不穩定
代理程式日誌檔的位置和內容格式	不穩定
密碼檔案	不穩定
accesscontrol.properties	不穩定

表格 E-2 介面穩定性分類機制

分類	描述
私有	不由客戶直接使用。可在任何版次中變更或移除。
形成中	由客戶使用。用於主要版次 (如 3.0、4.0) 或次要版次 (如 3.1、3.2) 的不相容變更。應謹慎而緩慢地進行變更。應進行適當的操作以確保所有的變更均相容，但是這點卻無法保證。
穩定	由客戶使用。僅用於主要版次 (如 3.0、4.0) 的不相容變更。
標準	由客戶使用。這些介面由正式標準定義，並由標準組織控制。幾乎沒有與這些介面不相容的變更。
不穩定	由客戶使用。用於主要版次 (如 3.0、4.0) 或次要版次 (如 3.1、3.2) 的不相容變更。通知客戶在未來版次中，這些介面可能會以不相容的方式進行大幅度移除或變更。建議客戶不要在不穩定的介面上建立明確的相依性。

詞彙表

本詞彙表提供有關使用 Sun™ ONE Message Queue (MQ) 時可能遇到的專有名詞和概念的資訊。

JMS (Java 訊息服務) 定義 JMS 用戶端如何存取訊息服務設備的標準的一組介面和語義。這些介面為 Java 程式提供標準的方法，以建立、傳送、接收和讀取訊息。

JMS 用戶端 與使用訊息服務交換訊息的其他 JMS 用戶端進行互動式操作的應用程式 (或軟體元件)。

JMS 供應程式 為訊息傳送系統實施 JMS 介面，並新增完整產品所需的管理和控制功能的產品。

MQ 用戶端運行時間 為 JMS 用戶端提供 MQ 訊息伺服器介面的軟體。用戶端運行時間支援用戶端將訊息傳送至目標和從此類目標接收訊息所需的所有作業。

MQ 訊息伺服器 為 MQ 訊息傳送系統提供遞送服務的軟體，其中包括與 JMS 用戶端的連線、訊息路由與遞送、持續性、安全性以及記錄。訊息伺服器維護實體目標，JMS 用戶端將訊息傳送至此目標，並且訊息從此目標遞送至使用用戶端。

主題 由管理員建立的可實施出版/訂閱遞送模型的物件。主題可作為內容階層 (負責收集和分布傳送給它的訊息) 中的節點來檢視。透過將主題用作媒介，訊息出版者可與訊息訂閱者保持獨立。

代理程式 MQ 實體，可管理訊息路由、遞送、持續性、安全性以及記錄，並提供可讓管理員監視和調整效能與資源使用的介面。

出版/訂閱遞送模型 出版者和訂閱者通常為匿名，並可動態出版或訂閱主題。系統將來自主題的多個出版者的訊息分布至其多個訂閱者。

生產 將訊息傳送至用戶端運行時間，以便遞送至目標。

生產者 由階段作業建立的物件 (MessageProducer)，用於將訊息傳送至目標。在點對點遞送模型中，生產者為傳送者 (QueueSender)；在出版/訂閱遞送模型中，生產者為出版者 (TopicPublisher)。

用戶端運行時間 請參閱 MQ 用戶端運行時間。

用戶端識別碼 將連線及其物件與作為 JMS 用戶端的 MQ 訊息伺服器所維護的狀態相關聯的識別碼。

目標 MQ 訊息伺服器中的實體目標，已生產的訊息遞送至此目標，以便路由並隨後遞送至使用者。此實體目標由受管理物件識別和封裝，JMS 用戶端使用此受管理物件指定它要生產訊息和/或使用訊息的目標。

佇列 由管理員建立的可實施點對點遞送模型的物件。佇列始終用於保留訊息，即使使用其訊息的用戶端處於非作用中。佇列被用作生產者與使用者之間的媒介保留位置。

使用 訊息使用者從目標接收訊息。

使用者 階段作業建立的用於從目標接收訊息的物件 (MessageConsumer)。在點對點遞送模型中，使用者為接收者或瀏覽者 (QueueReceiver or QueueBrowser)；在出版/訂閱遞送模型中，使用者為訂閱者 (TopicSubscriber)。

使用者群組 JMS 用戶端使用者所屬的群組，旨在授權對 MQ 訊息伺服器資源 (如連線和目標) 的存取權。

受管理物件 預先配置的 MQ 物件 (連線工廠或目標)，由管理員建立以供一個或多個 JMS 用戶端使用。

使用受管理物件可讓 JMS 用戶端獨立於供應程式；即，使用戶端脫離供應程式的專用方面。這些物件由管理員放置在 JNDI 名稱空間中，並由使用 JNDI 查找的 JMS 用戶端存取。

非同步通訊 一種通訊模式，在此模式中，訊息傳送者在繼續其他工作之前無需等待傳送方法的傳回。

訊息 JMS 用戶端使用的非同步請求、報告或事件。訊息包含標頭 (其他欄位可新增至此) 和內文。訊息標頭指定標準欄位和選擇性屬性。訊息內文包含要傳送的資料。

訊息服務 請參閱 MQ 訊息伺服器。

訊息傳送 企業應用程式使用的非同步請求、報告或事件的系統，可讓無密切關聯的應用程式可靠且安全地傳送資訊。

訊息選擇器 使用者選擇基於 JMS 訊息標頭中屬性值 (選擇器) 的訊息之方法。訊息服務執行基於訊息選擇器中所存在條件的訊息過濾與路由。

配置檔案 包含用於配置代理程式的 MQ 設定的一個或多個文字檔案。屬性為實例特定的或與叢集相關的。

授權 訊息服務決定使用者是否可存取訊息服務資源 (如連線服務或目標) 的過程。

異動 必須完成或完全回轉的原子單位的工作。

連線 1) 與 MQ 訊息伺服器的作用中連線。可以為佇列連線或主題連線。2) 階段作業的 Factory，使用構成 MQ 訊息伺服器基礎的連線以生產和使用訊息。

連線工廠 用戶端用於建立與 MQ 訊息伺服器連線的受管理物件。可以為 QueueConnectionFactory 物件或 TopicConnectionFactory 物件。

階段作業 用於傳送和接收訊息的單一執行緒環境。可以為佇列階段作業或主題階段作業。

資料儲存區 永久儲存代理程式所需資訊 (持久訂閱、有關目標的資料、永久性訊息以及稽核資料) 的資料庫。

網域 JMS 用戶端用於程式化 JMS 訊息傳送作業的一組物件。有兩種程式設計網域：一種用於點對點遞送模型，另一種用於出版/訂閱遞送模型。

遞送策略 當多個訊息使用者註冊時，佇列如何路由訊息的規格。這些策略為：單一、故障轉移和循環。

遞送模式 訊息傳送可靠性的指標：保證訊息遞送並成功使用一次且僅為一次 (永久性遞送模式)，或保證訊息至多遞送一次 (非永久性遞送模式)。

遞送模型 訊息遞送的模型：點對點或出版/訂閱。在 JMS 中，每種遞送模型都有單獨的程式設計網域，並且使用特定用戶端運行時間物件和特定目標類型 (佇列或主題)，以及統一的程式設計網域。

點對點遞送模型 生產者將訊息傳送至特定佇列；使用者從為保留其訊息而建立的佇列擷取訊息。一個訊息僅遞送至一個訊息使用者。

叢集 以串聯方式工作可提供訊息傳送服務的兩個或多個互連代理程式。

字母

admin 連線服務 42, 134

config.properties 檔案 102

HTTP 42, 134

HTTP 支援架構 191

HTTP 代理 191

HTTP 通道 Servlet 191, 197

HTTP 連線

支援 191

多重代理程式，用於 196

通道 Servlet，請參閱 HTTP 通道 Servlet

請求間隔時間 195

HTTP 傳輸驅動程式 191

httpjms 連線服務 42, 134

HTTPS 支援架構 191

HTTPS 通道 Servlet 182, 191

HTTPS 連線

支援 191

多重代理程式，用於 205

通道 Servlet，請參閱 HTTPS 通道 Servlet

請求間隔時間 203

httpsjms 連線服務 42, 134

imq.accesscontrol.enabled 屬性 53, 104

imq.accesscontrol.file.filename 屬性 54, 104

imq.authentication.basic.user_repository 屬性 53, 104

imq.authentication.client.response.timeout

屬性 53, 104

imq.authentication.type 屬性 53, 104

imq.autocreate.queue 屬性 60, 104, 132

imq.autocreate.topic 屬性 60, 104, 132

imq.cluster.brokerlist 屬性 63, 112

imq.cluster.hostname 屬性 64, 112

imq.cluster.masterbroker 屬性 63, 112

imq.cluster.port 屬性 64, 112

imq.cluster.url 屬性 63, 104, 112, 132

imq.httpjms.http.pullPeriod 屬性 195

imq.httpjms.http.servletHost 屬性 195

imq.httpjms.http.servletPort 屬性 195

imq.httpsjms.https.pullPeriod 屬性 203

imq.httpsjms.https.servletHost 屬性 203

imq.httpsjms.https.servletPort 屬性 203

imq.keystore.file.dirpath 屬性 179

imq.keystore.file.name 屬性 179

imq.keystore.password 屬性 179, 183

imq.log.console.output 屬性 57, 104

imq.log.console.output.stream 屬性 57, 104

imq.log.console.syslog.facility 屬性 57, 105

imq.log.console.syslog.identity 屬性 57, 105

imq.log.console.syslog.logconsole 屬性 57, 105

imq.log.console.syslog.logpid 屬性 57, 105

imq.log.console.syslog.output 屬性 57, 105

imq.log.file.dirpath 屬性 56, 105

imq.log.file.filename 屬性 56, 105

- imq.log.file.output 屬性 56, 105
- imq.log.file.rolloverbytes 屬性 57, 105, 132
- imq.log.file.rolloversecs 屬性 57, 105, 132
- imq.log.level 屬性 56, 105, 132
- imq.message.expiration.interval 屬性 47, 105
- imq.message.max_size 屬性 47, 105, 132
- imq.metrics.enabled 屬性 56, 105
- imq.metrics.interval 屬性 56, 105
- imq.passfile.dirpath 屬性 54, 105
- imq.passfile.enabled 屬性 54, 105
- imq.passfile.name 屬性 54, 105
- imq.persist.file.destination.file.size 屬性 50, 105
- imq.persist.file.message.cleanup 屬性 50, 105
- imq.persist.file.message.fdpool.limit 屬性 51, 106
- imq.persist.file.message.filepool.cleanratio 屬性 50, 106
- imq.persist.file.message.filepool.limit 屬性 50, 106
- imq.persist.file.sync.enabled 屬性 51, 106
- imq.persist.jdbc.brokerid 屬性 187
- imq.persist.jdbc.closedburl 屬性 188
- imq.persist.jdbc.createdburl 屬性 188
- imq.persist.jdbc.driver 屬性 187
- imq.persist.jdbc.needpassword 屬性 188
- imq.persist.jdbc.opendburl 屬性 187
- imq.persist.jdbc.password 屬性 183, 188
- imq.persist.jdbc.user 屬性 188
- imq.persist.store 屬性 50, 106, 187
- imq.portmapper.port 屬性 44, 106, 132
- imq.queue.deliverypolicy 屬性 60, 106, 132
- imq.redelivered.optimization 屬性 48, 106
- imq.service.activelist 屬性 44, 106
- imq.shared.connectionMonitor_limit 屬性 44, 107
- imq.system.max_count 屬性 47, 107, 132
- imq.system.max_size 屬性 47, 107, 132
- imq.transaction.autorollback 屬性 48, 107, 143
- imq.user_repository.ldap.base 屬性 171
- imq.user_repository.ldap.gidattr 屬性 171
- imq.user_repository.ldap.grpbase 屬性 171
- imq.user_repository.ldap.grpfilter 屬性 171
- imq.user_repository.ldap.grpsearch 屬性 171
- imq.user_repository.ldap.memattr 屬性 171
- imq.user_repository.ldap.password 屬性 171, 183
- imq.user_repository.ldap.principal 屬性 170
- imq.user_repository.ldap.server 屬性 170
- imq.user_repository.ldap.ssl.enabled 屬性 172
- imq.user_repository.ldap.timeout 屬性 172
- imq.user_repository.ldap.uidattr 屬性 171
- imq.user_repository.ldap.usrfilter 屬性 171
- imq. 服務名稱 .accesscontrol.enabled 屬性 106
- imq. 服務名稱 .accesscontrol.enabled 屬性 54
- imq. 服務名稱 .accesscontrol.file.filename 屬性 54, 106
- imq. 服務名稱 .authentication.type 屬性 53, 106
- imq. 服務名稱 .max_threads 屬性 44, 106
- imq. 服務名稱 .min_threads 屬性 44, 107
- imq. 服務名稱 .threadpool_model 屬性 44, 107
- imq. 服務名稱 .協定類型 .hostname 屬性 45, 107
- imq. 服務名稱 .協定類型 .port 屬性 45, 107
- imq. 資源狀態 .threshold 屬性 47, 106
- imqAckOnAcknowledge 屬性 151
- imqAckOnProduce 屬性 151
- imqAckTimeout 屬性 151
- imqbrokerd 指令
 - 使用 108
 - 指令語法 108
 - 選項 109
 - 關於 74
- imqBrokerHostName 屬性 151
- imqBrokerHostPort 屬性 151
- imqBrokerServicePort 屬性 151
- imqcmd 指令
 - 子指令 124
 - 用於 124, 147, 165
 - 目標管理 136
 - 指令語法 124

- 異動管理 141
- 連線至代理程式 127
- 與代理程式的安全連線 127, 181
- 選項 126
- 關於 74
- imqConfiguredClientID 屬性 151
- imqConnectionType 屬性 151
- imqConnectionURL 屬性 151
- imqdbmgr 指令
 - 子指令 190
 - 指令語法 189
 - 選項 190
 - 關於 75
- imqDefaultPassword 屬性 151
- imqDefaultUsername 屬性 151
- imqDestinationDescription 屬性 68, 152
- imqDestinationName 屬性 68, 152
- imqDisableSetClientID 屬性 151
- imqFlowControlCount 屬性 151
- imqFlowControlIsLimited 屬性 151
- imqFlowControlLimit 屬性 151
- imqJMSDeliveryMode 屬性 152
- imqJMSExpiration 屬性 152
- imqJMSPriority 屬性 152
- imqkeytool 指令
 - 使用 178, 200
 - 指令語法 178, 200
 - 關於 75
- imqLoadMaxToServerSession 屬性 151
- imqobjmgr 指令
 - 子指令 147
 - 指令語法 147
 - 選項 148
 - 關於 74
- imqOverrideJMSDeliveryMode 屬性 152
- imqOverrideJMSExpiration 屬性 152
- imqOverrideJMSHeadersToTemporaryDestinations 屬性 152
- imqOverrideJMSPriority 屬性 152
- imqQueueBrowserMaxMessagesPerRetrieve 屬性 151
- imqQueueBrowserRetrieveTimeout 屬性 151
- imqReconnect 屬性 151
- imqReconnectDelay 屬性 151
- imqReconnectRetries 屬性 151
- imqSetJMSXAppID 屬性 151
- imqSetJMSXConsumerTXID 屬性 151
- imqSetJMSXProducerTXID 屬性 151
- imqSetJMSXRcvTimestamp 屬性 151
- imqSetJMSXUserID 屬性 151
- imqSSLIsHostTrusted 屬性 152
- imqsvcadm 指令
 - 子指令 212
 - 用於 212
 - 指令語法 212
 - 選項 213
 - 關於 75
- imqusermgr 指令
 - 子指令 166
 - 使用者名稱 168
 - 指令語法 165
 - 密碼 168
 - 選項 166
 - 關於 75
- J2EE 應用程式
 - EJB 規格 30
 - JMS, 和 30
- JDBC 支援
 - 設定 185
 - 關於 50
 - 驅動程式 185, 187
- JDK
 - 指定路徑 110
 - 要指定路徑的選項 126, 148, 213

三畫

JMS

- 訊息結構 28
- 規格 28
- 程式設計模型 29

JMS 規格 22, 23

jms 連線服務 42, 134

JNDI

- MQ 支援 24
- 位置 (提供者 URL) 153
- 受管理物件, 和 30, 32
- 物件儲存區 74, 146
- 物件儲存區屬性 150, 153
- 初始環境 153
- 查找 66, 68, 91, 149
- 訊息驅動 Bean, 和 32

JRE, 指定路徑 110

LDAP 伺服器

- 使用者儲存庫存取 170
- 物件儲存區屬性 153

MDB 請參閱訊息驅動 Bean

SOAP 24

SSL

- 加密, 與 177
- 服務, 設定 163
- 連線服務, 與 42, 134
- 透過 HTTP 182
- 透過 TCP/IP 178
- 關於 53

ssladmin 連線服務 42, 134

- 配置 180
- 設定 178

ssljms 連線服務 42, 134

- 配置 180
- 設定 178

TCP 42, 134

TLS 42, 134

Windows 服務, 代理程式執行為 211

XA 連線工廠

- 另請參閱連線工廠受管理物件
- 關於 36

XA 資源管理程式, 請參閱分散式異動

三畫

工具, 管理, 請參閱管理工具

四畫

元件

- EJB 30
- MDB 31

內建持續性 49

分散式異動

- XA 資源管理員 141
- XA 資源管理程式 35
- 另請參閱 XA 連線工廠
- 關於 35

日誌檔

- 捲動條件 57
- 預設位置 56, 217

五畫

主代理程式 62, 63

主題

- 如實體目標 59
- 自動建立的 60, 104
- 新增受管理物件 158
- 關於 33
- 屬性 137

主題目標, 請參閱主題

代理程式

- HTTP 支援 193
- httpjms 連線服務屬性 195
- httpsjms 連線服務屬性 203
- HTTPS, 支援 200

- JDBC 支援，請參閱 JDBC 支援
- Windows 服務，執行為 211
- 工作 40
- 互連，請參閱代理程式叢集
- 主代理程式 62
- 列出服務 134
- 回應 (Ack) 46
- 多重代理程式叢集，請參閱代理程式叢集
- 存取控制，請參閱授權
- 安全性管理程式，請參閱安全性管理程式
- 自動建立目標屬性 60
- 更新 130
- 系統資源 47
- 持續性管理程式，請參閱持續性管理程式
- 查詢 130，131
- 重新啟動 48，129
- 屬性 132
- 記錄，請參閱記錄程式
- 訊息容量 47
- 訊息路由，請參閱訊息路由器
- 配置檔案，請參閱配置檔案
- 從故障回復 48
- 控制 129
- 啟動 108
- 啟動基於 SSL 的服務 180
- 連線在一起 114
- 連線至 127
- 連線服務，請參閱連線服務
- 實例名稱 110
- 實例配置屬性 104
- 暫停 129
- 標準資訊，請參閱標準資訊
- 叢集，請參閱代理程式叢集
- 關於 40
- 關閉 129
- 繼續 129
- 代理程式實例，請參閱代理程式
- 代理程式叢集
 - 主代理程式 62，63
 - 在僅開發環境中 63
 - 使用的原因 60
 - 架構 61
 - 要指定的選項 109
 - 重新啟動代理程式 115
 - 配置屬性 63，112
 - 配置變更記錄 62
 - 設定屬性 113
 - 連線代理程式 114
 - 新增代理程式 114
 - 資訊傳遞 62
 - 叢集配置檔案 63，112
- 出版/訂閱遞送 33
- 加密
 - 基於 SSL 的服務，與 177
 - 鍵值工具，與 53
 - 關於 53
- 可靠的遞送 34
- 可攜型，請參閱供應程式獨立性
- 外掛持續性
 - 設定 185
 - 關於 50
- 平台版 24
- 永久性
 - 遞送模式，請參閱遞送模式
- 永久性訊息 34
- 生產者 29
- 用戶端
 - 程式設計模型 29
 - 運行時間 64
 - 應用程式，請參閱用戶端應用程式
 - 識別碼 (ClientID) 34
- 用戶端應用程式
 - 系統屬性，與 68
 - 提供者獨立性 32
- 目標
 - 介紹的 40
 - 主題，請參閱主題
 - 列出 137
 - 存取控制 176
 - 自動建立的 59，177
 - 佇列，請參閱佇列
 - 更新屬性 137，139
 - 建立 137

六畫

- 清除訊息 137，139
- 資訊 137，138
- 實體 58
- 管理 136
- 暫存 60，138
- 銷毀 137
- 屬性 137
- 目標受管理物件 68
 - 已介紹 30
 - 屬性 152
- 目錄變數
 - IMQ_HOME 20
 - IMQ_JAVAHOME 21
 - IMQ_VARHOME 20
- 目錄變數 IMQ_HOME 20
- 目錄變數 IMQ_JAVAHOME 21
- 目錄變數 IMQ_VARHOME 20

六畫

- 企業版 25
- 回應
 - 代理程式 46，151
 - 用戶端 46
 - 異動，與 46
 - 等待期間 151
 - 遞送，屬於 46
 - 關於 35，46
- 存取控制屬性檔案，請參閱存取控制檔案
- 存取控制檔案
 - 用於 172
 - 存取規則 174
 - 位置 172，218
 - 版本 173
 - 格式 173
- 存取規則 174
- 安全性
 - 加密，請參閱加密
 - 物件儲存區，用於 153

- 授權，請參閱授權
- 管理程式，請參閱安全性管理程式
- 認證，請參閱認證
- 安全性管理程式
 - 作為代理程式元件 41
 - 屬性 53
 - 關於 51
- 安全套接層標準，請參閱 SSL
- 自身簽名的證書 178，200
- 自動建立目標
 - 屬性 60
 - 關於 59

七畫

- 佇列 58
 - 自動建立的 60，104
 - 新增受管理物件 158
 - 遞送策略，請參閱佇列遞送策略
 - 屬性 137
- 佇列目標，請參閱佇列
- 佇列遞送策略
 - 故障轉移 58
 - 單一 58
 - 循環 58
 - 關於 58
 - 屬性 137
- 系統日誌 55，119
- 系統屬性，設定 68
- 防火牆 191

八畫

- 使用者 30
- 使用者名稱
 - 格式 168
 - 預設 151，164

使用者群組

- 刪除指定 167
- 預先定義的 167
- 預設 52
- 關於 52

使用者儲存庫

- LDAP 伺服器 170
- 文字檔案 164
- 平台相依性 165
- 位置 164, 217
- 使用者狀態 167
- 使用者群組 167
- 移入 168
- 管理 168
- 關於 51
- 類型 53

供應程式獨立性

- 受管理物件 67

協定，請參閱傳輸協定

協定類型

- HTTP 42, 134
- TCP 42, 134
- TLS 42, 134

受管理物件

- XA 連線工廠，請參閱連線工廠受管理物件
- 主題，請參閱主題
- 目標，請參閱目標受管理物件
- 列出 161
- 佇列，請參閱佇列
- 刪除 160
- 更新 162
- 供應程式獨立性 67
- 物件儲存區，請參閱物件儲存區
- 查找名稱 148
- 查詢 162
- 連線工廠，請參閱連線工廠受管理物件
- 需要的資訊 149
- 關於 30, 66
- 類型 30, 66, 146
- 屬性 150

服務類型

- ADMIN 42
- NORMAL 42

版本，產品

- 平台 24
- 企業 25
- 關於 24

物件儲存區

- LDAP 伺服器 146
- LDAP 伺服器屬性 153
- 位置 217
- 檔案系統儲存區 146
- 檔案系統儲存區屬性 153
- 關於 146

九畫

持久訂閱

- ClientID，和 34
- 列出 140
- 清除訊息 140
- 銷毀 140
- 關於 33

持久訂閱，請參閱持久訂閱

持續性

- JDBC，請參閱 JDBC 持續性
- 內建 49
- 外掛，請參閱外掛持續性
- 持續性管理程式，請參閱持續性管理程式
- 資料儲存區請參閱資料儲存區

持續性管理程式

- JDBC 資料儲存區 187
- 外掛持續性，與 185
- 作為代理程式元件 41
- 屬性 50
- 資料儲存區請參閱資料儲存區
- 關於 48

指令行公用程式

- imqcmd 74, 124
- imqdbmgr 75, 190

- imqkeytool 75, 178, 200
- imqobjmgr 74, 147
- imqsvcadmin 75, 212
- imqusermgr 75, 166
- 基本語法 75
- 選項共用於 76
- 關於 74
- 指令行語法 75
- 指令選項 76
- 訂閱
 - 持久 ID 126
 - 管理持久 140
 - 銷毀持久 141
- 重新遞送旗標 46

十畫

- 容器
 - EJB 32
 - MDB 32
- 效能，可靠性，和 36
- 屬性
 - httpjms 連線服務 195
 - httpsjms 連線服務 203
 - 代理程式，更新 132
 - 代理程式實例配置 104
 - 安全性 53
 - 自動建立 60
 - 持續性 50
 - 記錄程式 56
 - 訊息路由器 47
 - 連線服務 44
 - 與 JDBC 相關 187
 - 與 LDAP 相關的 170
 - 鍵值儲存區 179
 - 叢集配置 63
- 記憶體管理 47
- 記錄，請參閱記錄程式

- 記錄程式
 - 作為代理程式元件 41
 - 重新導向日誌訊息 119
 - 屬性 56
 - 級別 55, 56, 110
 - 訊息格式 117
 - 捲動條件 119
 - 預設配置 117
 - 種類 55
 - 寫入至主控台 57, 111
 - 標準資訊 56, 120
 - 輸出通道 55, 119
 - 歸檔檔案 56
 - 關於 54
 - 變更配置 118

- 訊息
 - SOAP 24
 - 已介紹 28
 - 代理程式限制 47
 - 出版/訂閱遞送 33
 - 可靠遞送 34
 - 永久性 34
 - 生產 65
 - 回收已過期的 47
 - 回應 46, 151
 - 在目標上清除 137
 - 使用 65
 - 持續性 47, 48
 - 重新遞送 46
 - 限制 138
 - 偵聽程式 30, 65
 - 控制 46
 - 排序 37
 - 結構 28
 - 路由與遞送 45
 - 過濾 36
 - 遞送模式，請參閱遞送模式
 - 遞送模型 28, 33
 - 優先順序排序 37
 - 點對點遞送 33

訊息生產者，請參閱生產者

訊息伺服器

多重代理程式，請參閱代理程式叢集 60

關於 40

訊息使用者，請參閱使用者

訊息服務 27

訊息偵聽程式，請參閱偵聽程式

訊息傳送系統

MQ 架構 40

架構 27

訊息服務 27

訊息路由器

作為代理程式元件 41

屬性 47

關於 45

訊息遞送模型 28，33

訊息驅動 Bean

MDB 容器 32

佈署描述元 32

應用程式伺服器支援 32

配置檔案

config.properties 102

安裝 102

位置 102，217

置換的值設定在 102

預設 102

實例 102，113，132，217

編輯 104

配置變更記錄 62

十一畫

偵聽程式 30，31

基於 SSL 的服務

啓動 180

基於 SSL 的連線服務

設定 178

執行緒儲存區管理程式

共用執行緒 44

專屬執行緒 43

關於 43

密碼

JDBC 183

LDAP 183

SSL 鍵值儲存區 110，179，183

命名慣例 168

密碼檔案，請參閱密碼檔案

預設 151

編碼 53

密碼檔案

代理程式配置屬性 54

位置 183，218

使用 182

指令行選項 110

密碼檔案，請參閱密碼檔案

控制訊息 46

授權

MQ 版本 24

另請參閱存取控制檔案

使用者群組 52

載入 110

管理 172

關於 52

異動

分散式，請參閱分散式異動

回應，與 46

回轉 141

資訊 141

管理 141

確定 141

關於 35

許可權

MQ 作業 52

內嵌式資料庫 186

存取控制屬性檔案 52，173

使用者儲存庫 165

密碼檔案 182

十二畫

- 資料儲存區 50
- 運算 174
- 管理服務 52
- 鍵值儲存區 200
- 通訊協定，請參閱傳輸協定
- 連接埠，動態配置 43
- 連接埠對映器
 - 連接埠指定 44，110
 - 關於 43
- 連線
 - 已介紹 29
 - 重新連線 151
 - 重新連線延遲 151
 - 重新連線嘗試 151
- 連線工廠受管理物件
 - ClientID，和 34
 - JNDI 查找 30
 - 已介紹 29
 - 新增 157
 - 置換 68
 - 關於 67
 - 屬性 67，150，151
- 連線服務
 - admin 42，134
 - httpjms 42，134
 - httpsjms 42，134
 - HTTPS，請參閱 HTTPS 連線
 - HTTP，請參閱 HTTP 連線
 - jms 42，134
 - ssladmin 42，134，178
 - ssljms 42，134，178
 - 在啟動時啟動 44
 - 存取控制 53
 - 更新 133，135
 - 服務類型 42
 - 指令影響 133
 - 查詢 133，135
 - 屬性 44

- 基於 SSL 的 179
- 執行緒分配 136
- 執行緒儲存區管理程式 43
- 連接埠對映器，請參閱連接埠對映器
- 連線類型 42
- 暫停 133，136
- 靜態連接埠 44
- 關於 41
- 繼續 133，136

十二畫

- 提供者獨立性
 - 關於 32
- 程式設計網域 33
- 階段作業
 - 已介紹 29
 - 回應選項 35
 - 異動的 35

十三畫

- 傳輸協定
 - HTTP 42，134
 - TCP 42，134
 - TLS 42，134
- 資料，MQ，位置 217
- 資料儲存區
 - JDBC 可存取的 50
 - 文字檔案 49
 - 位置 217
 - 重設 111
 - 關於 48
- 路由，請參閱訊息路由器

十四畫

圖表顯示：訊息生產者將訊息傳送至訊息服務，
然後訊息服務將這些訊息轉送至訊息
使用者 27

實例配置檔案，請參閱配置檔案

管理工作

生產環境 71

開發環境 71

管理工具

指令行公用程式 74

管理主控台 73

關於 73

網域 33

認證

管理 164

關於 51

遞送，可靠的 34

遞送模式

永久性 34

非永久性 34

十五畫

暫存目標 60，138

標準資訊

報告間隔時間 110

摘要 121

關於 55

十六畫

輸入檔案 154

十七畫

應用程式伺服器 32

環境變數，請參閱目錄變數

鍵值工具 53

鍵值儲存區

屬性 179

檔案 179，201

鍵對

重新產生 179

產生 178

點對點遞送 33

十八畫

叢集，請參閱代理程式叢集

叢集配置檔案 63

簡單物件存取協定請參閱 SOAP

十九畫

證書 178，200

