



Sun Java™ System
Message Queue 3
管理指南

2005Q1

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

文件號碼 819-2220

Copyright © 2005 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. 版權所有。

Sun Microsystems, Inc. 對本文件中描述之產品所採用的技術擁有相關智慧財產權。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。

美國政府權利 - 商業軟體。政府使用者應遵循 Sun Microsystems, Inc. 之標準軟體授權合約，以及 FAR 及其增補文件的適用條款。應依照授權條款使用。本發行版本可能包括協力廠商開發的材料。

Sun、Sun Microsystems、Sun 標誌、Java、Solaris、Sun[tm] ONE、JDK、Java 命名與目錄介面、JavaMail、JavaHelp 以及 Javadoc 為 Sun Microsystems, Inc. 在美國與其他國家 / 地區的商標或註冊商標。

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

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

該產品受美國出口控制法的規管與控制，也可能需要遵守其他國家 / 地區的進出口法律。嚴禁直接或間接用於核武器、導彈、生化武器或核能海上最終用途。嚴禁出口或再出口至被美國列入禁運清單的國家 / 地區或美國出口排除清單上確定的實體，包括但不限於被拒絕的個人，以及特別指定的國家。

目錄

圖目錄	13
表目錄	15
程序目錄	19
前言	21
本書適用對象	22
閱讀本書之前	22
本書架構	22
本書採用的慣例	23
U 字慣例	24
目錄變數慣例	24
相關U件	27
Message Queue U件集	27
線上說明	28
JavaDoc	28
範例客戶端應用程式	28
Java 訊息服務 (JMS) 規格	28
協力廠商U件	29
Sun 歡迎您提出寶貴意見	29
第 I 部分 Message Queue 管理簡介	31
第 1 章 管理工具	33
開發環境中的管理工作	34
生產環境中的管理工作	34
設定作業	34
維護作業	35
管理工具	36

指令行公用程式	36
管理主控台	38

第 2 章 管理快速入門	39
準備工作	40
啟動管理主控台	40
取得群集	42
啟動代理程式	43
新增代理程式	44
連線至代理程式	45
檢視連線服務	46
將實體目標新增至代理程式	47
管理實體目標	48
取得相關主題的資訊	51
使用物件儲存	51
新增物件儲存	52
檢查物件儲存特性	54
連線至物件儲存	54
新增連線工廠受管理物件	55
新增目標物件	57
檢視受管理物件特性	58
更新主控台資訊	59
執行範例應付程式	59

第 II 部分 管理工作

61

第 3 章 啟動代理程式和客戶端	63
準備系統資源	64
巨步化系統時鐘	64
設定檔案描述元限制 (Solaris 或 Linux)	64
以互動方式啟動代理程式	64
自動啟動代理程式	66
在 Solaris 和 Linux 自動啟動	66
在 Windows 自動啟動	66
啟動 Message Queue 客戶端	68
移除代理程式實例	69
第 4 章 配置代理程式	71
關於可配置的代理程式元件	72
連線服務	73

訊息路由器	76
持續性管理程式	80
安全性管理程式	83
監視服務	87
關於配置檔表	90
實例配置檔表	91
合併特性值	91
特性命名語法	92
編輯實例配置檔表	92
在指令行上輸入配置選項	93
設定永久性儲存	93
配置檔表系統儲存	94
配置 JDBC 儲存	94
保護永久性資料	98
重建 (基於檔表的) 永久性儲存	98
外掛 (JDBC) 永久性儲存	98
第 5 章 管理代理程式	101
必要條件	102
使用 imqcmd 指令公用程式	102
指定使用者名稱和密碼	102
指定代理程式名稱和連接埠	103
範例	103
顯示說明	104
顯示產品版本	104
顯示代理程式資訊	104
更新代理程式特性	105
暫停與繼續代理程式	106
暫停代理程式	106
繼續代理程式	107
關閉和重新啟動代理程式	107
顯示代理程式度量	108
管理連線服務	109
列出連線服務	109
顯示連線服務資訊	110
更新連線服務特性	111
顯示連線服務度量	111
暫停與繼續連線服務	112
取得有關連線的資訊	113
管理長期訂閱	115
管理作業事件	116

第 6 章 管理實體目標	119
使用 imqcmd 指令公用程式	120
子指令	120
建立實體目標	121
列出實體目標	122
顯示實體目標資訊	123
更新實體目標特性	124
暫停和繼續實體目標	125
清除實體目標	125
刪除實體目標	126
壓縮實體目標	127
配置使用停用的訊息佇列	128
配置使用停用的訊息佇列	129
配置和管理停用的訊息佇列	129
啟用停用訊息記錄功能	130
第 7 章 管理安全性	131
認證使用者	132
使用子檔或使用者儲存庫	132
將 LDAP 伺服器註冊於使用者儲存庫	137
授權使用者：存取控制特性檔	140
建立存取控制特性檔	141
存取規則的語法	142
如何計算許可權	143
連線服務的存取控制	144
實體目標的存取控制	145
自動建立實體目標的存取控制	146
使用基於 SSL 的服務	146
TCP/IP 的完全連線服務	147
配置使用自身簽名的憑證	147
配置使用簽名的憑證	152
使用密碼檔	155
完全考量	156
密碼檔的內容	156
建立稽核記錄	157
第 8 章 管理管理物件	159
關於物件儲存	160
LDAP 伺服器物件儲存	160
檔或系統物件儲存	161
關於受管理物件屬性	162
連線工廠屬性	162

用戶端標識	165
目標受管理物件屬性	168
使用物件管理員公用程式 (imqobjmgr)	169
需要的資訊	169
使用指令檔	170
新增和刪除受管理物件	172
新增連線工廠	173
新增主題佇列	174
刪除受管理物件	175
列出受管理物件	176
取得有關單一物件的資訊	176
更新受管理物件	177
第 9 章 使用代理程式叢集	179
叢集配置特性	180
設定個別代理程式的叢集特性	180
使用叢集配置檔錄	181
管理叢集	181
連線代理程式	181
將代理程式新增至叢集	182
從叢集中移除代理程式	183
主代理程式	184
管理配置變更記錄	184
當主代理程式無法使用時	185
第 10 章 監視訊息傳呼器	187
監視工具簡介	188
配置及使用代理程式記錄功能	189
預設記錄配置	189
記錄訊息格式	190
變更記錄程式配置	191
互動顯示度量	194
imqcmd metrics	195
使用顯示度量資料的 metrics 子指令	196
度量輸出：imqcmd metrics	196
imqcmd query	198
撰寫應用程式來監視代理程式	199
設定基於訊息的監視	199
安全性與存取注意事項	200
度量輸出：度量訊息	201

第 11 章 分析與調整訊息佇列	203
關於效能	204
效能稽核程序	204
效能類型	204
效能測定	205
基本使用式樣	205
影響效能的因素	206
影響效能的應用程式設計因素	208
影響效能的訊息服務因素	214
調整配置以改善效能	219
系統調整	219
代理程式調整	223
用戶端執行階段訊息流量調整	225
第 12 章 排解疑難雜症	227
用戶端無法建立連線	228
連線流量過慢	232
用戶端無法建立訊息產生者	234
訊息產生延遲或過慢	235
儲存訊息	237
訊息伺服器流量不穩定	241
訊息未送達用戶	242
佇列的訊息佇列中包含訊息	246

第 III 部分 參照

第 13 章 指令參照	255
指令行語法	256
指令的輸入規則	256
指令行範例	256
共用指令選項	257
imqbrokerd	257
語法	257
指令選項	258
另譯參數	261
imqcmd	261
語法	261
子指令	262
指令選項	269
另譯參數	271
imqobjmgr	271

語法	271
子指令	271
指令選項	272
另請參閱	273
imqdbmgr	273
語法	273
子指令	274
指令選項	274
另請參閱	275
imqusermgr	275
語法	275
子指令	276
指令選項	276
另請參閱	277
imqsvcadm	277
語法	277
子指令	277
指令選項	278
另請參閱	278
imqkeytool	279
語法	279
另請參閱	279
第 14 章 管理程式特性參照	281
按字母順序排列的特性清單	282
連線服務特性	285
訊息路由 器特性	287
持續性管理程式特性	289
基於檔案的持續性	290
基於 JDBC 的持續性	291
安全性管理程式特性	293
監視和記錄特性	297
叢集配置特性	300
第 15 章 實體環境特性參照	303
第 16 章 管理物件屬性參照	307
目標特性	308
連線工廠屬性	308
連線處理	309
用戶端標識	312
訊息標頭標識	312

穩定性與流量控制	313
佇列瀏覽器運作方式和伺服器階段作業	314
JMS 定義的特性支援	315
SOAP 端點屬性	315

第 17 章 JMS 資源介面屬性參照	317
ResourceAdapter JavaBean	318
ManagedConnectionFactory JavaBean	319
ActivationSpec JavaBean	320

第 18 章 度量參照	323
JVM 度量	324
代理程式範疇度量	324
連線服務度量	326
目標度量	328

第 IV 部分 附錄	331
-------------------------	------------

附錄 A Message Queue 資料的作業系統特定位置	333
Solaris	334
Linux	335
Windows	336

附錄 B Message Queue 介面的穩定性	337
--	------------

附錄 C HTTP/HTTPS 支援	341
HTTP/HTTPS 支援架構	342
啓用 HTTP 支援	343
步驟 1. 在 Web Server 上部署 HTTP 通道 Servlet	343
步驟 2. 配置 httpjms 連線服務	344
步驟 3. 配置 HTTP 連線	346
範例 1：在 Sun Java System Web Server 上部署 HTTP 通道 Servlet	347
範例 2：在 Sun Java System Application Server 7.0 上部署 HTTP 通道 Servlet	350
啓用 HTTPS 支援	352
步驟 1. 為 HTTPS 通道 Servlet 產生自身簽名的憑證	353
步驟 2. 在 Web Server 上部署 HTTPS 通道 Servlet	353
步驟 3. 配置 httpsjms 連線服務	355
步驟 4. 配置 HTTPS 連線	356
範例 3：在 Sun Java System Web Server 上部署 HTTPS 通道 Servlet	359
範例 4：在 Sun Java System Application Server 7.0 上部署 HTTPS 通道 Servlet	363

環境排解	364
伺服器或代理程式發生故障	364
用戶端無法透過通道 Servlet 連線	365
字彙表	367
索引	369

圖目錄

圖 1-1	本機與遠端管理公用程式	37
圖 4-1	代理程式服務元件	72
圖 4-2	連線服務支援	74
圖 4-3	持續性管理程式支援	81
圖 4-4	安全性管理程式支援	85
圖 4-5	監視服務支援	87
圖 4-6	代理程式配置檔案	92
圖 11-1	透過 Message Queue 服務的訊息傳送	207
圖 11-2	傳送模式的效能影響	210
圖 11-3	訂閱類型的效能影響	212
圖 11-4	訊息容量的效能影響	213
圖 11-5	傳輸協定速度	216
圖 11-6	傳輸協定的效能影響	217
圖 11-7	在 1K (1,024 個位元組) 封包上變更 inbufsz 的效果	222
圖 11-8	在 1K (1,024 個位元組) 封包上變更 outbufsz 的效果	222
圖 12-1	QBrowser 視窗	244
圖 12-2	QBrowser 訊息的詳細資訊	245
圖 C-1	HTTP/HTTPS 支援架構	342

表目錄

表 1	本書內容	22
表 2	文件慣例	24
表 3	Message Queue 目錄變數	25
表 4	Message Queue 文件集	27
表 4-1	主代理程式服務元件和功能	73
表 4-2	代理程式支援的連線服務	73
表 4-3	度量主題目標	88
表 5-1	代理程式支援的連線服務	109
表 5-2	imqcmd 更新的連線服務特性	111
表 6-1	imqcmd 指令公用程式的實體目標子指令	120
表 6-2	實體目標磁碟使用度量	127
表 6-3	停用的訊息佇列處理標準實體目標特性的方式	129
表 7-1	使用者儲存庫中的初始項目	133
表 7-2	imqusermgr 選項	134
表 7-3	存取規則的語法元素	142
表 7-4	實體目標存取控制規則的元素	145
表 7-5	自身簽名的憑證需要的識別名稱資訊	148
表 7-6	使用密碼的指令	155
表 7-7	密碼檔案中的密碼	156
表 8-1	LDAP 物件儲存屬性	160
表 8-2	檔案系統物件儲存屬性	161
表 8-3	命名慣例範例	173
表 10-1	度量監視工具的優點與限制	188
表 10-2	記錄級別	190
表 10-3	imqbrokerd 記錄程式選項和對應的特性	191
表 10-4	imqcmd metrics 子指令語法	195
表 10-5	imqcmd metrics 子指令選項	195

表 10-6	imqcmd query 子指令語法	198
表 10-7	度量主題目標	199
表 11-1	高可靠性和高效能方案的比較	209
表 13-1	共用 Message Queue 指令行選項	257
表 13-2	imqbrokerd 選項	258
表 13-3	imqcmd 子指令	262
表 13-4	imqcmd 子指令用於管理代理程式	264
表 13-5	用於管理目標的 imqcmd 子指令	265
表 13-6	用於管理連線服務的 imqcmd 子指令	267
表 13-7	用於管理連線服務的 imqcmd 子指令	268
表 13-8	用於管理長期訂閱的 imqcmd 子指令	268
表 13-9	用於管理作業事件的 imqcmd 子指令	268
表 13-10	imqcmd 選項	269
表 13-11	imqobjmgr 子指令	271
表 13-12	imqobjmgr 選項	272
表 13-13	imqdbmgr 子指令	274
表 13-14	imqdbmgr 選項	274
表 13-15	imqusermgr 子指令	276
表 13-16	imqusermgr 選項	276
表 13-17	imqsvcadmin 子指令	277
表 13-18	imqsvcadmin 選項	278
表 14-1	代理程式實例配置特性	282
表 14-2	連線服務特性	285
表 14-3	訊息路由器特性	287
表 14-4	自動建立配置特性	288
表 14-5	必要的持續性管理程式特性	289
表 14-6	基於檔案的持續性的特性	290
表 14-7	基於 JDBC 的持續性的特性	291
表 14-8	安全性管理程式特性	293
表 14-9	密鑰儲存特性	297
表 14-10	監視服務特性	297
表 14-11	叢集配置特性	300
表 15-1	實體目標特性	303
表 16-1	目標受管理物件屬性	308
表 16-2	連線工廠屬性：連線處理	309
表 16-3	imqAddressList 屬性的定址方案	310
表 16-4	訊息伺服器位址範例	311

表 16-5	連線工廠屬性：用戶端標識	312
表 16-6	連線工廠屬性：訊息標頭覆寫	312
表 16-7	連線工廠屬性：穩定性與流量控制	313
表 16-8	連線工廠屬性：佇列瀏覽器運作方式	314
表 16-9	連線工廠屬性：JMS 定義的特性支援	315
表 16-10	SOAP 端點屬性	316
表 17-1	資源介面屬性	318
表 17-2	受管理連線工廠屬性	319
表 17-3	啟動規格屬性	321
表 18-1	JVM 度量	324
表 18-2	代理程式範圍度量	324
表 18-3	連線服務度量	326
表 18-4	目標度量	328
表 A-1	Solaris 上 Message Queue 資料的位置	334
表 A-2	Linux 上 Message Queue 資料的位置	335
表 A-3	Windows 上 Message Queue 資料的位置	336
表 B-1	介面穩定性分類方案	337
表 B-2	Message Queue 介面的穩定性	338
表 C-1	httpjms 連線服務特性	345
表 C-2	用於部署 HTTP 通道 Servlet JAR 檔案的 Servlet 引數	348
表 C-3	httpsjms 連線服務特性	355
表 C-4	用於部署 HTTPS 通道 Servlet JAR 檔案的 Servlet 引數	360

程序目錄

若要顯示管理主控台說明資訊	42
若要將代理程式新增至管理主控台	44
若要連線至代理程式	45
若要檢視可用的連線服務	46
若要將佇列目標新增至代理程式	48
若要檢視實體目標的特性	49
清除實體目標的訊息	50
若要刪除目標	51
新增檔案系統物件儲存	52
若要顯示物件儲存的特性	54
若要連線至物件儲存	55
若要將連線工廠新增至物件儲存	55
若要將目標新增至物件儲存	57
若要檢視或更新目標物件的特性	58
若要執行 HelloWorldMessageJNDI 應用程式	59
查閱記錄的服務錯誤事件	68
基本的傳送機制	76
插入 JDBC 可存取的資料儲存	95
建立實體目標	122
回收未使用的實體目標磁碟空間	128
編輯配置檔案以使用 LDAP 伺服器	138
設定管理使用者	140
設定基於 SSL 的連線服務	147
若要重新產生鍵對	150
若要啟用代理程式中基於 SSL 的服務	150
若要取得簽名的憑證	153
若要安裝簽名的憑證	153

配置 Java 用戶端執行階段	154
使用叢集配置檔案將新的代理程式新增至叢集	182
不使用叢集配置檔案將新的代理程式新增至叢集	183
使用指令行從叢集中移除代理程式	183
使用叢集配置檔案從叢集中移除代理程式	184
若要備份配置變更記錄	184
復原配置變更記錄	184
變更代理程式的記錄程式配置	191
若要使用記錄檔報告度量資訊	193
若要使用 <code>metrics</code> 子指令	196
若要設定基於訊息的監視	200
啓用 HTTP 支援	343
啓動 <code>httpjms</code> 連線服務	345
新增通道 Servlet	348
若要配置通道 Servlet 的虛擬路徑 (Servlet URL)	348
在啓動 Web Server 時載入通道 Servlet	349
停用伺服器存取記錄	349
將 HTTP 通道 Servlet 部署為 WAR 檔案	349
在 Application Server 7.0 環境中部署 HTTP 通道 Servlet	351
若要修改 Application Server 的伺服器策略檔案	352
啓用 HTTPS 支援	352
啓動 <code>httpsjms</code> 連線服務	355
配置 JSSE	356
新增通道 Servlet	359
若要配置通道 Servlet 的虛擬路徑 (Servlet URL)	360
在啓動 Web Server 時載入通道 Servlet	360
停用伺服器存取記錄	361
修改 HTTPS 通道 Servlet WAR 檔案	361
將 HTTPS 通道 Servlet 部署為 WAR 檔案	362
在 Application Server 7.0 環境中部署 HTTPS 通道 Servlet	363
若要修改 Application Server 的伺服器策略檔案	364

Sun Java™ System Message Queue 管理指南提供您管理 Message Queue 訊息傳送系統時所需的資訊。

本書介紹 Sun Java System Message Queue 3 2005Q1 (Message Queue 3.6)。

本前言包含以下各節：

- 第 22 頁 「本書適用對象」
- 第 22 頁 「閱讀本書之前」
- 第 22 頁 「本書架構」
- 第 23 頁 「本書採用的慣例」
- 第 27 頁 「相關文件」
- 第 29 頁 「協力廠商文件」
- 第 29 頁 「Sun 歡迎您提出寶貴意見」

本書讀者對象

本書可供需要執行 Message Queue 管理工作的管理員和應用程式開發者使用。

Message Queue 管理員負責設定和管理 Message Queue 訊息傳送系統，尤其是位於本系統中心的 Message Queue 訊息伺服器。

閱讀本書之前

您必須先閱讀 Message Queue 技術概述，熟悉 Message Queue 實作 Java 訊息規格、Message Queue 服務元件，以及開發、部署及管理 Message Queue 應用程式的基本程序。

本書架構

下表簡要描述了本書的內容。

表 1 本書內容

部分/章	說明
第 I 部分「Message Queue 管理簡介」	
第 1 章「管理工作和工具」	介紹 Message Queue 管理工作和工具。
第 2 章「管理快速入門」	提供實用指導，讓您熟悉管理主控台。
第 II 部分「管理工作」	
第 3 章「啓動代理程式和用戶端」	描述如何啓動 Message Queue 代理程式和用戶端。
第 4 章「配置代理程式」	描述如何設定及讀取配置特性，並介紹代理程式的可配置內容。此外還描述如何設定檔案或資料庫執行持續性功能。
第 5 章「管理代理程式」	描述代理程式管理工作。
第 6 章「管理實體目標」	描述主題和佇列的相關管理工作。
第 7 章「管理安全性」	解釋與安全性相關的工作，例如管理密碼檔案、認證、授權和加密。
第 8 章「管理受管理物件」	描述物件儲存，並解釋如何執行目標受管理物件和連線工廠受管理物件的相關工作。
第 9 章「使用代理程式叢集」	描述如何設定及管理 Message Queue 代理程式叢集。

表 1 本書內容 (續)

部分 / 章	說明
第 10 章「監視訊息伺服器」	描述如何設定及使用 Message Queue 監視設備。
第 11 章「分析與調校訊息服務」	描述分析訊息伺服器效能的技術，並解釋如何將訊息伺服器調校到最佳化效能。
第 12 章「排解疑難問題」	提供如何找出一般 Message Queue 問題成因和如何解決問題的建議。
第 III 部分「參照」	
第 13 章「指令參照」	提供 Message Queue 指令公用程式的語法和說明。
第 14 章「代理程式特性參照」	列出並描述可用來配置代理程式的特性。
第 15 章「實體目標特性參照」	列出並描述可用來配置主題和佇列的特性。
第 16 章「受管理物件屬性參照」	列出並描述可用來配置目標受管理物件和連線工廠受管理物件的特性。
第 17 章「JMS 資源介面屬性參照」	列出並描述可用來配置與應用程式伺服器配合使用的 Message Queue 資源介面的特性。
第 18 章「度量參照」	列出並描述 Message Queue 代理程式產生的度量。
第 IV 部分「附錄」	
附錄 A「Message Queue 資料的作業系統特定位置」	列出 Message Queue 檔案在每個支援平台上的位置。
附錄 B「Message Queue 介面的穩定性」	描述不同 Message Queue 介面的穩定性。
附錄 C「HTTP/HTTPS 支援」	描述如何設定使用 HTTP 與 Message Queue 通訊。

本書採用的慣例

此節提供了有關本文件中所使用慣例的資訊。

ㄷ字慣例

表 2 ㄷ字慣例

格式	說明
ABC123 <i>斜體</i>	斜體文字表示定位字元。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。
ABC123 固定間距字體	等寬字體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。
「書名」	書名
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 (Message Queue、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

Message Queue 使用三個目錄變數，其設定方式根據平台的不同而不同。表 3 描述了這些變數，並概述它們在 Solaris™、Windows 以及 Linux 平台上的使用方式。

表 3 Message Queue 目錄變數

變數	說明
IMQ_HOME	<p>它通常在 Message Queue 文件中使用，指的是 Message Queue 基底目錄（根安裝目錄）：</p> <ul style="list-style-type: none"> • 在 Solaris 和 Linux 上，沒有 Message Queue 根安裝目錄。因此，未在 Message Queue 文件中使用 IMQ_HOME 來指代 Solaris 上的檔案位置。 • 在 Solaris 和 Windows 上，Sun Java System Application Server 的 Message Queue 根安裝目錄位於 Application Server 基底目錄下的 /imq。 • 在 Windows 上，Message Queue 根安裝目錄由 Message Queue 安裝程式設定（依預設，該目錄為 C:\Program Files\Sun\MessageQueue3）。
IMQ_VARHOME	<p>這是儲存 Message Queue 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。</p> <ul style="list-style-type: none"> • 在 Solaris 上，IMQ_VARHOME 預設為 /var/imq 目錄。 • 在 Solaris 上，針對 Sun Java System Application Server Evaluation Edition，IMQ_VARHOME 預設為 IMQ_HOME/var 目錄。 • 在 Windows 上，IMQ_VARHOME 預設為 IMQ_HOME/var 目錄。 • 在 Windows 上，針對 Sun Java System Application Server，IMQ_VARHOME 預設為 IMQ_HOME/var 目錄。 • 在 Linux 上，IMQ_VARHOME 預設為 /var/opt/sun/mq 目錄。

表 3 Message Queue 目錄變數(續)

變數	說明
IMQ_JAVAHOME	<p>這是指向 Message Queue 可執行檔所需的 Java™ 執行環境 (JRE) 位置的環境變數：</p> <ul style="list-style-type: none"> 在 Solaris 上，IMQ_JAVAHOME 依下列順序尋找 Java 執行階段，但是使用者可以選擇性地將值設定為所需 JRE 駐留的任意位置。 Solaris 8 或 9： <ul style="list-style-type: none"> /usr/jdk/entsys-j2se /usr/jdk/jdk1.5.* /usr/jdk/j2sdk1.5.* /usr/j2se Solaris 10： <ul style="list-style-type: none"> /usr/jdk/entsys-j2se /usr/java /usr/j2se 在 Linux 上，Message Queue 先依下列順序尋找 Java 執行階段，但是使用者可以選擇性地將 IMQ_JAVAHOME 值設定為所需 JRE 駐留的任意位置。 <ul style="list-style-type: none"> /usr/jdk/entsys-j2se /usr/java/jre1.5.* /usr/java/jdk1.5.* /usr/java/jre1.4.2* /usr/java/j2sdk1.4.2* 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所需 JRE 駐留的任意位置。

在本書中，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)。路徑名稱通常使用 UNIX 目錄分隔符號表示法 (/)。

相關文件

除了本書以外，Message Queue 還提供了其他文件資源。

Message Queue 文件集

構成 Message Queue 文件集的文件依照一般的使用次序列示在表 4 中。

表 4 Message Queue 文件集

文件	使用者	說明
Message Queue Installation Guide	開發者與管理員	解釋如何在 Solaris、Linux 以及 Windows 平台上安裝 Message Queue 軟體。
Message Queue 版本說明	開發者與管理員	包括對新功能、限定、已知錯誤以及技術性注意事項的描述。
Message Queue 技術概述	開發者與管理員	描述 Message Queue 概念、功能和元件。
Message Queue 管理指南	管理員與開發者	提供使用 Message Queue 管理工具執行管理工作時所需的背景與資訊。
Message Queue Developer's Guide for Java Clients	開發者	提供如何開發 Java 用戶端程式的資訊，此程式以 Message Queue 實作 JMS 和 SOAP/JAXM 規格。
Message Queue Developer's Guide for C Clients	開發者	提供如何開發 C 用戶端程式的資訊，此程式將 C 介面 (C-API) 用於 Message Queue 訊息服務。

線上說明

Message Queue 包括用於執行 Message Queue 訊息服務管理工作的命令行公用程式。若要存取這些公用程式的線上說明，請參閱第 13 章「指令參照」。

Message Queue 還包括圖形化使用者介面 (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

下列位置提供 JavaDoc 格式的 JMS 和 Message Queue API 文件：

平台	位置
Solaris	/usr/share/javadoc/imq/index.html
Linux	/opt/sun/mq/javadoc/index.html/
Windows	IMQ_HOME/javadoc/index.html

本文件可在任何 HTML 瀏覽器 (如 Netscape 或 Internet Explorer) 中檢視。它包括標準的 JMS API 文件，以及用於 Message Queue 受管理物件的 Message Queue 特定的 API (請參閱 Message Queue Developer's Guide for Java Clients 的第 3 章)，對訊息傳送應用程式的開發者很有幫助。

範例用戶端應用程式

大量的範例應用程式提供了範例用戶端應用程式碼，它們在平台特定的目錄中 (請參閱附錄 A「Message Queue 資料的作業系統特定位置」)。

請參閱該目錄及其各個子目錄中的 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。

協力廠商文件

協力廠商 URL 為本文件的參照資料且提供了額外的相關資訊。

備註 Sun 對於本文件中所提及之協力廠商網站的使用不承擔任何責任。Sun 對於此類網站或資源中的 (或透過它們所取得的) 任何內容、廣告、產品或其他材料不做背書，也不承擔任何責任。對於因使用或依靠此類網站或資源中的 (或透過它們所取得的) 任何內容、產品或服務而造成的或連帶產生的實際或名義上之損壞或損失，Sun 概不負責，也不承擔任何責任。

Sun 歡迎您提出寶貴意見

Sun 致力於提高文件品質，因此誠心歡迎您提出意見與建議。

若要分享您的意見，請至 <http://docs.sun.com> 然後按一下 [傳送您的回饋意見]。請在線上表單中提供文件標題和文件號碼。在書的標題頁或文件的上方可找到文件號碼，通常是 7 或 9 個數字。例如，本書的標題為「Sun Java System Message Queue 3 2005Q1 管理指南」，而文件號碼為 819-2220。

提供意見與建議時，您可能需要在表單中提供文件的英文標題及文件號碼。此文件的英文文件號碼及標題為：819-0066，「Sun Java System Message Queue 3 2005Q1 Administration Guide」。

Sun 歡迎您提出寶貴意見

Message Queue 管理簡介

第 1 章 管理工作和工具

第 2 章 管理快速入門

管理工作和工具

Sun Java™ System Message Queue 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。本章包含以下各節：

- [第 34 頁](#) 「開發環境中的管理工作」
- [第 34 頁](#) 「生產環境中的管理工作」
- [第 36 頁](#) 「管理工具」

開發環境中的管理工作

在開發環境中，工作重點為 Message Queue 用戶端應用程式的程式設計，程式設計師會經常管理他們自己的系統。Message Queue 訊息伺服器主要用於測試。在開發環境中強調靈活性，且通常管理會包含以下作業：

- 最小管理，主要包含為開發者啟動代理程式以用於測試。
- 內建基於檔案的持續性、基於檔案的使用者儲存庫和檔案系統儲存的使用。這些簡單的配置通常足以用於開發測試。
- 多重代理程式測試不使用主代理程式。
- 使用自動建立的目標，而不是管理員建立的目標。
- 以用戶端程式碼而不是由管理員創設受管理物件。

生產環境中的管理工作

在生產環境中，由於必須可靠地部署和執行應用程式，因此管理會變得更加重要。您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。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）：

- **管理員安全性**（受保護的管理工具使用）：
 - 授權：允許特定個人或群組從停用的訊息佇列存取管理連線服務和使用訊息（請參閱第 144 頁「連線服務的存取控制」和第 145 頁「實體目標的存取控制」）。
 - 如果您使用的是預設管理使用者（admin）和基於檔案的使用者儲存庫，請變更使用者密碼（請參閱第 137 頁「變更預設管理員密碼」）。
 - 如果您正在授權一個群組，請確定每個管理員都屬於該群組。
- 基於檔案的使用者儲存庫

基於檔案的使用者儲存庫擁有一個管理員（admin）的單一群組。如果您建立了一個新的管理使用者，請確定新的使用者位於 admin 群組中。

- LDAP 使用者儲存庫

建立一個 LDAP 伺服器中的群組，或使用一個現有的群組。請確定您要給予管理權限的使用者為群組中的成員，然後將管理連線授權給群組的成員。

如需更多資訊，請參閱第 137 頁「將 LDAP 伺服器用於使用者儲存庫」。

- **一般安全性** (請參閱第 7 章「管理安全性」):
 - 認證：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。)
 - 授權：修改存取控制特性檔案中的存取設定。
 - 加密：設定基於 SSL 的連線服務 (請參閱第 146 頁「使用基於 SSL 的服務」)。
- **受管理物件** (請參閱第 8 章「管理受管理物件」):
 - 配置或設定 LDAP 物件儲存。
 - 建立 ConnectionFactory 和目標受管理物件。
- **代理程式叢集** (請參閱第 9 章「使用代理程式叢集」):
 - 建立中央配置檔案。
 - 使用主代理程式。
- **持續性**：決定代理程式使用的是外掛持續性，或是內建持續性，並設定所需的儲存 (請參閱第 93 頁「設定永久性儲存」)。
- **記憶體管理**：設定目標屬性，以便訊息數目和訊息配置的記憶體容量，符合可用的代理程式記憶體資源 (請參閱第 303 頁表 15-1)。

維護作業

在生產環境中，需要密切監視和控制 Message Queue 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 Message Queue 管理工具執行一些不間斷的工作，如下所述：

- **應用程式管理**
 - 設定 imq.autocreate.queue 和 imq.autocreate.topic 特性的值，以停用代理程式的自動建立功能 (請參閱第 287 頁「訊息路由器特性」)。

- 建立作為應用程式的實體目標 (請參閱第 119 頁第 6 章「管理實體目標」)。
- 設定使用者對目標的存取權 (請參閱第 140 頁「授權使用者：存取控制特性檔案」)。
- 監視與管理目標 (請參閱第 115 頁「管理長期訂閱」)。
- 監視與管理長期訂閱 (請參閱第 115 頁「管理長期訂閱」)。
- 監視與管理作業事件 (請參閱第 116 頁「管理作業事件」)。
- **代理程式管理與調校**
 - 使用代理程式度量調校和重新配置代理程式 (請參閱第 203 頁第 11 章「分析與調校訊息服務」)。
 - 管理代理程式記憶體資源 (請參閱第 203 頁第 11 章「分析與調校訊息服務」)。
 - 將代理程式新增至叢集以平衡負載 (請參閱第 9 章「使用代理程式叢集」)。
 - 回復失敗的代理程式 (請參閱第 64 頁「以互動方式啟動代理程式」)。
- **管理應用程式**
 - 依需要建立其他 `ConnectionFactory` 和目標受管理物件 (請參閱第 172 頁「新增和刪除受管理物件」)。
 - 調整 `ConnectionFactory` 屬性值以確保 Java 用戶端應用程式的正確運作方式 (請參閱第 8 章「管理受管理物件」)。

管理工具

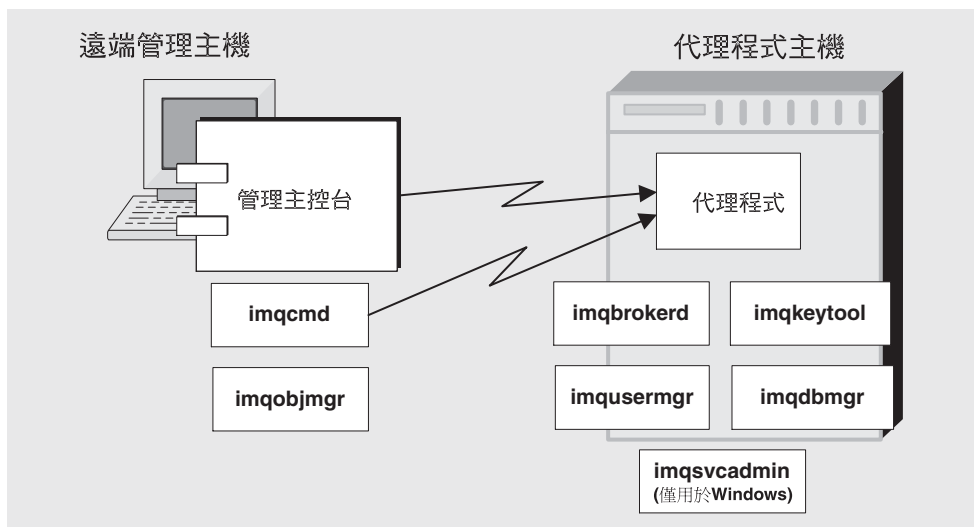
Message Queue 管理工具分為兩類：

- 命令行公用程式
- 圖形管理主控台 (imqadmin)

命令行公用程式

本節介紹用於執行 Message Queue 管理工作的命令行公用程式。您可以使用 Message Queue 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 1-1 本機與遠端管理公用程式



所有 Message Queue 公用程式均可透過命令行介面 (CLI) 存取。公用程式指令共用常用格式、語法慣例和選項，如本章後續中所述。在 [第 13 章「指令參照」](#) 中，您可以找到有關使用命令行公用程式的參照資訊。

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，並指定代理程式在啟動時使用的其他配置資訊。

指令 (imqcmd) 啟動代理程式後，您可以使用命令行公用程式建立、更新和刪除實體目標，控制代理程式及其連線服務，以及管理代理程式的資源。

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。

使用者管理員 (imqusermgr) 您可以使用使用者管理員公用程式，寫入用於認證和授權使用者的基於檔案的使用者儲存庫。

鍵值工具 (imqkeytool) 您可以使用鍵值工具公用程式，產生用於 SSL 認證的自身簽名憑證。

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。

服務管理員 (imqsvcadmin) 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。

管理主控台

管理主控台合併了兩種指令行公用程式的一些功能：指令公用程式 (imqcmd) 和物件管理員公用程式 (imqobjmgr)。

您可以使用管理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 Message Queue 受管理物件。其他指令行公用程式 (imqusermgr、imqdbmgr 和 imqkeytool) 必須與它們相關的代理程式在相同主機上執行，如第 37 頁圖 1-1 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](#)」中有描述

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。
- 在代理程式上建立並管理實體目標。
- 連線至物件儲存。
- 將受管理物件新增至物件儲存並管理這些物件。

您無法使用管理主控台執行某些工作，包括啟動代理程式、建立代理程式叢集、配置代理程式和實體目標的更加專用的特性以及管理使用者資料庫。

第 2 章「[管理快速入門](#)」提供了簡短的實用指導，可讓您熟悉管理主控台，並說明如何使用主控台完成基本的工作。

管理快速入門

本快速入門的重點放在基本管理工作，亦即使用圖形化界面的管理主控台來管理 **Message Queue** 代理程式與物件儲存。您只要遵循本章中的指示，將可學習如何執行以下作業：

- 啓動代理程式。
- 連線至代理程式，並且使用管理主控台來管理代理程式。
- 在代理程式上建立實體目標。
- 建立物件儲存並使用管理主控台連線至此物件儲存。
- 將目標物件新增到物件儲存中，並檢視其特性。

快速入門旨在設定當您執行簡單 **JMS** 相容應用程式 `HelloWorldMessageJNDI` 時所需的實體目標和受管理物件。此應用程式在範例應用程式目錄 (Solaris 和 Windows 平台的 `demo` 或 Linux 的 `examples`；請參閱附錄 A 「[Message Queue 資料的作業系統特定位置](#)」) 的 `helloworld` 子目錄中。您將在本快速入門的最後部分，執行此應用程式。

提供本快速入門主要是爲了指導您使用管理主控台執行基本的管理工作。這不能取代您閱讀和參照文件。

有些 **Message Queue** 管理工作不能靠管理主控台來完成。類似下列的工作，必須使用命令行公用程式來執行：

- 配置某些實體目標特性
- 建立代理程式叢集
- 管理使用者資料庫

如需如何完成這些工作的詳細資訊，請參閱第 6 章 「[管理實體目標](#)」、第 9 章 「[使用代理程式叢集](#)」和第 7 章 「[管理安全性](#)」。

準備工作

在您開始之前，請務必先安裝 Message Queue 產品。如需更多資訊，請參閱 Message Queue Installation Guide。請注意，本章主要以 Windows 為主，並針對 UNIX® 使用者加入備註。

在本章中，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

若要啟動管理主控台，請執行以下方法之一：

- 在 Windows 上，請選擇 [開始] > [程式集] > [Sun Microsystems] > [Sun Java System Message Queue 3.6] > [管理]。

- 在 Solaris 上，請輸入下列指令：

```
/usr/bin/imqadmin
```

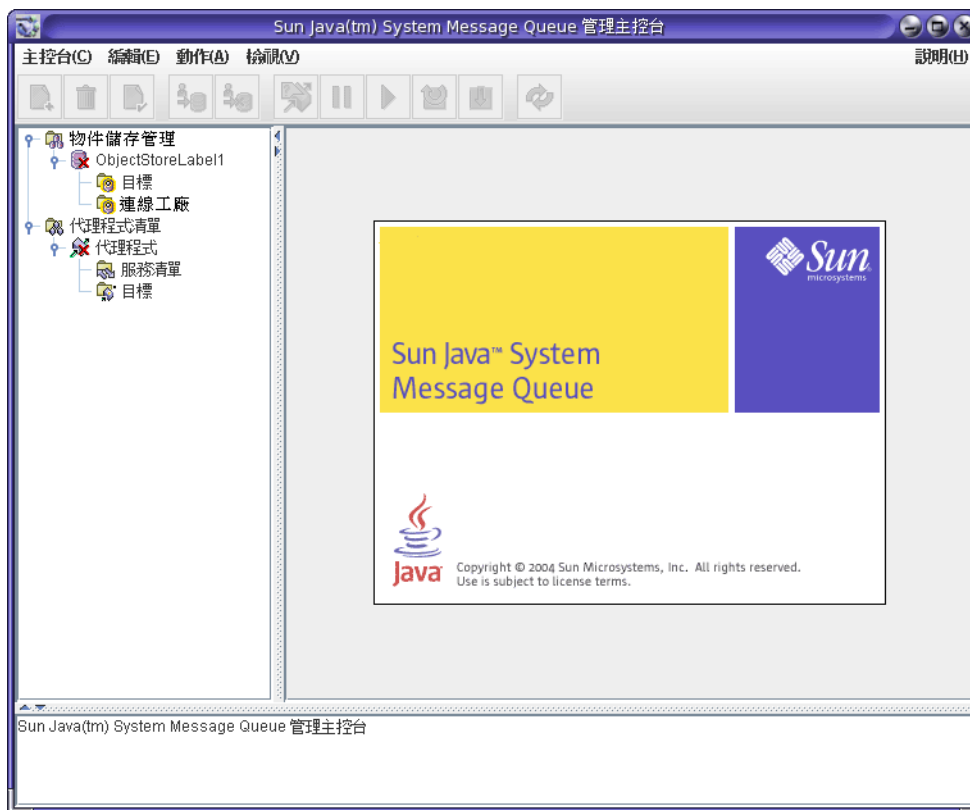
- 在 Linux 上，請輸入下列指令：

```
/opt/sun/mq/bin/imqadmin
```

可能需要等待幾秒鐘之後才會顯示主控台視窗。

花費幾秒鐘查看主控台視窗。

主控台視窗特徵為，頂部有一個功能表列，緊挨功能表列下面為工具列，視窗左側為瀏覽窗格，右側為結果窗格（此時正顯示識別 Sun Java System Message Queue 產品的圖形），視窗底部為狀態窗格。



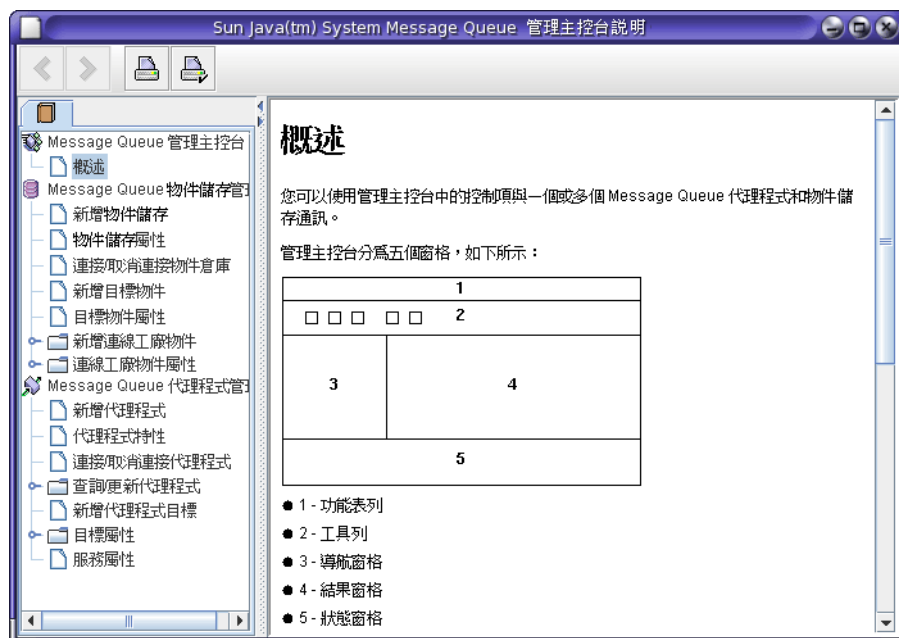
本章未提供完整的資訊，因此讓我們首先找到取得管理主控台說明資訊的方法。

取得說明

在功能表列的最右端可以找到 [說明] 功能表。

► 若要顯示管理主控台說明資訊

1. 下拉 [說明] 功能表並選擇 [簡介]。畫面上將會顯示說明視窗。



請注意說明資訊的組織方式。左側的瀏覽窗格顯示目錄表；右側的結果窗格顯示您在瀏覽窗格選取的任何項目的內容。

查看 [說明] 視窗的結果窗格。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 [說明] 視窗的瀏覽窗格。它以三個區域管理主題：簡介、物件儲存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 44 頁「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照。但是開始之前，請先檢查線上說明以獲得資訊。

3. 在 [說明] 視窗的目錄窗格中，按一下 [新增代理程式] 項目。

請注意結果窗格已經變更。它此時包含介紹新增代理程式涵義的文字，還包含描述 [新增代理程式] 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 [說明] 視窗。

啓動代理程式

您無法使用管理主控台啓動代理程式。請使用下列其中一種方法：

- 在 Windows 上，請選擇 [開始] > [程式集] > [Sun Microsystems] > [Sun Java System Message Queue 3.6] > [訊息代理程式]。

- 在 Solaris 上，請輸入下列指令：

```
/usr/bin/imqbrokerd
```

- 在 Linux 上，請輸入下列指令：

```
/opt/sun/mq/bin/imqbrokerd
```

如果您使用 Windows [開始] 功能表，會出現指令視窗。指令出現下列回應時，表示代理程式已經備妥，顯示如下行：

```
Loading persistent data...  
Broker "imqbroker@stan:7676 ready.
```

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之前，您不必啓動此代理程式，但是您必須啓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照。新增代理程式之後，您可以連線至此代理程式。

► 若要將代理程式新增至管理主控台

1. 在瀏覽窗格中的 [代理程式] 上按一下滑鼠右鍵，然後選擇 [新增代理程式]。
2. 在 [代理程式標籤] 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。

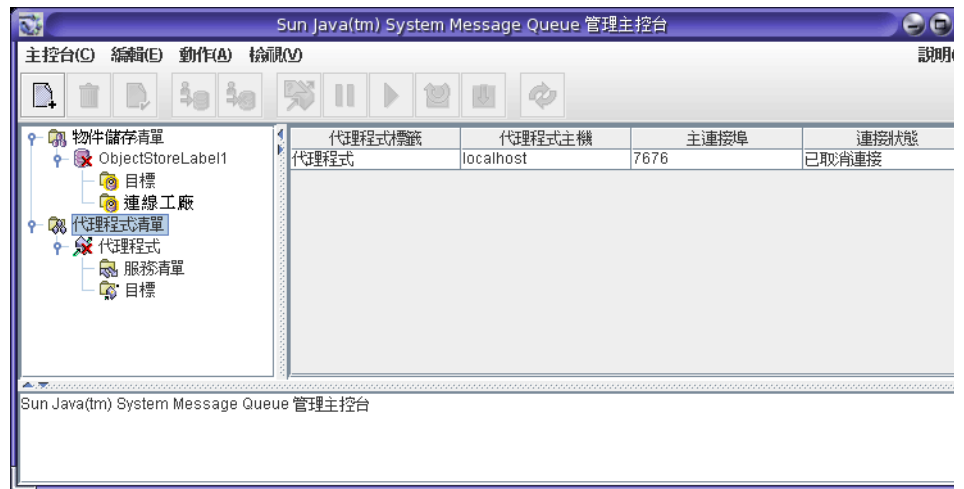


請注意對話方塊中指定的預設主機名稱 (localhost) 和主要連接埠 (7676)。稍後當您配置連線工廠 (用戶端將用來設定與此代理程式的連線) 時，您必須指定這些值。

保留 [密碼] 欄位為空白。如果您在連線時指定密碼將更安全。

3. 按一下 [確定] 以新增此代理程式。

查看瀏覽窗格。您剛新增的代理程式應已在 [代理程式] 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。

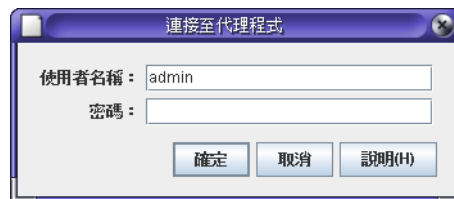


4. 在 [MyBroker] 上按一下滑鼠右鍵，然後從快顯功能表中選擇 [特性]。
畫面上將會顯示代理程式特性對話方塊。您可以使用此對話方塊更新您在新增此代理程式時指定的任何特性。
5. 按一下 [取消] 以關閉此對話方塊。

連線至代理程式

► 若要連線至代理程式

1. 在 MyBroker 上按一下滑鼠右鍵，然後選擇 [連線至代理程式]。
出現對話方塊，要求您輸入使用者名稱和密碼。



依預設，管理主控台可以用使用者 admin，密碼 admin 連線至代理程式。在本例，使用預設值。在實際情況中，您應儘快建立安全的使用者名稱和密碼。請參閱第 132 頁「認證使用者」，以獲得更多資訊。

2. 在 [密碼] 欄位中輸入 admin。

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 [確定] 以連線至此代理程式。

連線至此代理程式之後，您可以從 [動作] 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。

► 若要檢視可用的連線服務

1. 在瀏覽窗格中選取 [服務]。

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 [動作] 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特性。
4. 從 [動作] 功能表中選擇 [特性]。

請注意，透過使用 [服務特性] 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 [確定] 或 [取消] 以關閉 [特性] 對話方塊。
6. 在結果窗格中選取 **admin** 服務。
7. 下拉 [動作] 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 [動作] > [特性] 以檢視 **admin** 服務的特性。
9. 當您完成之後，按一下 [確定] 或 [取消]。

將實體目標新增至代理程式

依預設，代理程式啟用實體目標自動建立功能。自動建立功能可以讓代理程式動態建立實體目標。

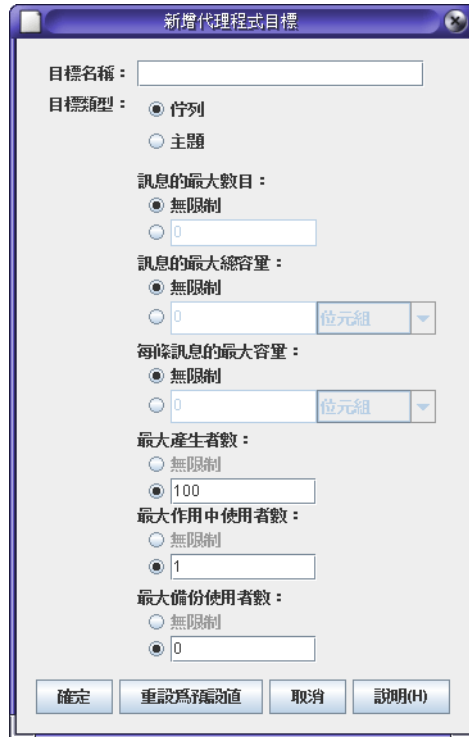
在開發環境中，您不需明確建立實體目標來測試用戶端程式碼。

在生產設定中，建議您明確建立實體目標。這樣可讓您 (管理員) 充分地瞭解在代理程式上使用的實體目標。

現在您要將實體目標新增至代理程式。請注意您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。

► 若要將佇列目標新增至代理程式

1. 在 MyBroker 的 [目標] 節點上按一下滑鼠右鍵，然後選擇 [新增代理程式目標]。
畫面上將會顯示以下對話方塊：



2. 在 [目標名稱] 欄位中輸入 MyQueueDest。
3. 如果尚未選取 [佇列] 單選按鈕，請選取此按鈕。
4. 按一下 [確定] 以新增此實體目標。
此時此實體目標將會在結果窗格中出現。

管理實體目標

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，如以下程序所述：

- 檢視與更新實體目標的特性
- 清除實體目標的訊息

- 刪除實體目標

► 若要檢視實體目標的特性

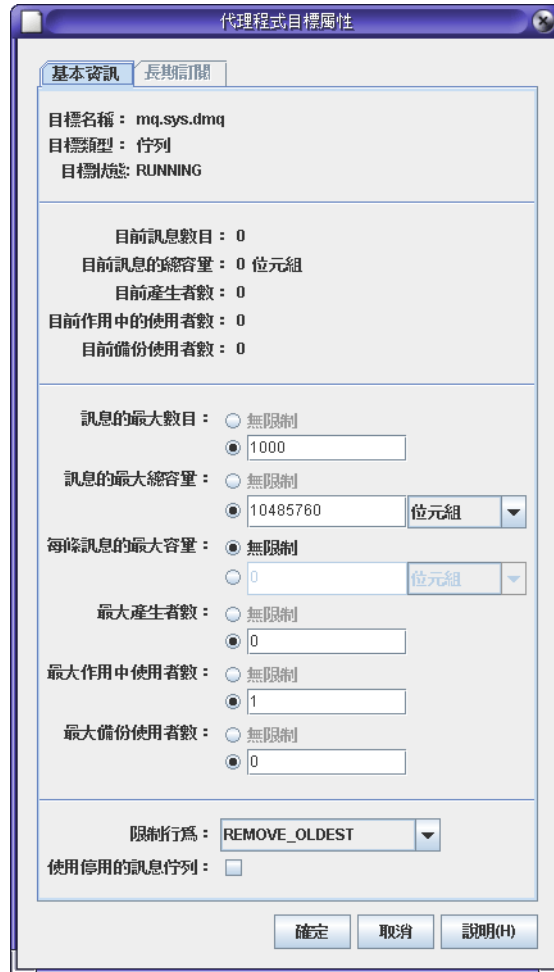
1. 選取 MyBroker 的 [目標] 節點。

結果面板中出現兩個實體目標，MyQueueDest 和 `mq.sys.dmq`。mq.sys.dmq 目標是系統建立的佇列，儲存代理程式的到期訊息和拒絕訊息。目前請暫時略過此停用的訊息佇列。

2. 在結果窗格中選取 [MyQueueDest]。

3. 選擇 [動作] > [特性]。

畫面上將會顯示以下對話方塊：



請注意，對話方塊會顯示目前有關佇列的狀態資訊，以及一些您可以變更的特性資訊。

4. 按一下 [取消] 以關閉此對話方塊。

► 清除實體目標的訊息

1. 在結果窗格中選取實體目標。
2. 選擇 [動作] > [清除訊息]。

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。

► 若要刪除目標

1. 在結果窗格中選取實體目標。
2. 選擇 [編輯] > [刪除]。

畫面上將會顯示確認對話方塊。

備註 請勿刪除 MyQueueDest 佇列目標。

刪除實體目標時會清除此目標上的訊息並移除此目標。

取得有關主題的資訊

代理程式主題目標的特性對話方塊包括一個列示長期訂閱資訊的附加標籤。此標籤在佇列為停用狀態。



您可以使用此對話方塊：

- 清除長期訂閱以及移除與長期訂閱相關聯的所有訊息
- 刪除長期訂閱、清除與長期訂閱相關聯的所有訊息以及移除長期訂閱

使用物件儲存

物件儲存可用來儲存 Message Queue 受管理物件。這些受管理物件封裝 Message Queue 特定的有關用戶端應用程式所用物件的實作與配置資訊。物件儲存可能是 LDAP 目錄伺服器或檔案系統儲存 (檔案系統中的目錄)。

受管理物件可以在用戶端程式碼中創設與配置。但是管理員最好還是在用戶端應用程式可以透過 JNDI 存取的物件儲存中，建立、配置並儲存這些物件。這樣可讓用戶端程式碼獨立於供應程式。

您無法使用管理主控台建立物件儲存。您必須提前進行此作業，如以下一節中所述。

新增物件儲存

新增物件儲存可在管理主控台中建立現有物件儲存的參照。即使您退出並重新啓動主控台，此參照仍可保留。

► 新增檔案系統物件儲存

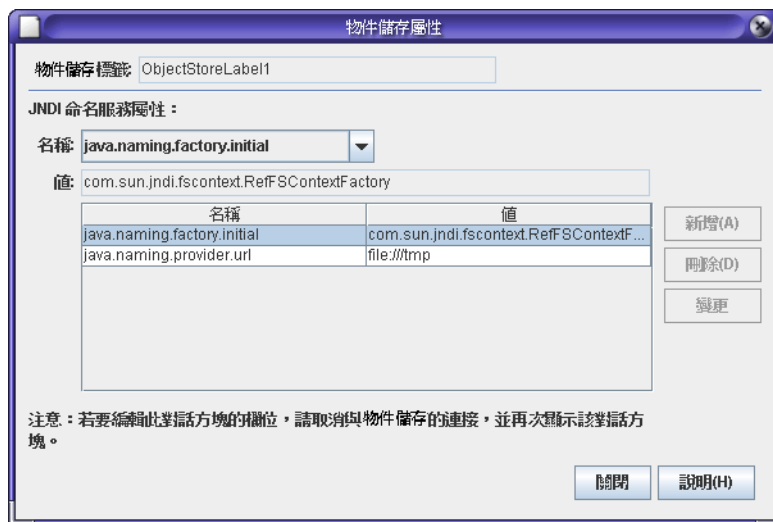
1. 如果您的 C 磁碟機上還沒有名為 Temp 的資料夾，請現在建立一個。

本章中所用範例應用程式假設物件儲存為 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 [物件儲存] 上按一下滑鼠右鍵，然後選擇 [新增物件儲存]。

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內的物件儲存顯示提供標籤。

在以下步驟中，您必須輸入 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 [名稱] 下拉式清單中選取 java.naming.factory.initial。

此特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 [值] 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。(對於 LDAP 儲存，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。如需有關設定伺服器以及進行 JNDI 查找的資訊，請參閱第 160 頁「LDAP 伺服器物件儲存」。

6. 按一下 [新增] 按鈕。

請注意此時特性摘要窗格中列出的特性及其值。

7. 從 [名稱] 下拉式清單中選取 java.naming.provider.url。

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

8. 在 [值] 欄位中輸入以下內容

file:///C:/Temp

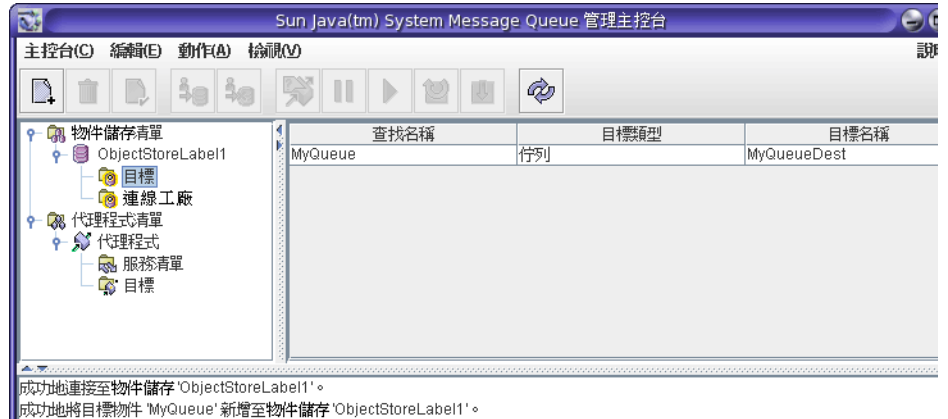
(在 Solaris 和 Linux 上為 file:///tmp)

9. 按一下 [新增] 按鈕。

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

10. 按一下 [確定] 以新增此物件儲存。

11. 如果尚未在瀏覽窗格中選取節點 [MyObjectStore]，請現在選取它。
管理主控台此時顯示如下：



此物件儲存已在瀏覽窗格中列出，其內容、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，這可在結果窗格的 [計數] 欄中顯示。

瀏覽窗格中的物件儲存圖示上畫有紅色的 X。這表示此物件儲存已中斷連線。在使用物件儲存之前，您必須先連線至物件儲存。

檢查物件儲存特性

當中斷管理主控台與物件儲存的連線時，您可以檢查與變更此物件儲存的某些特性。

► 若要顯示物件儲存的特性

1. 在瀏覽窗格中的 [MyObjectStore] 上按一下滑鼠右鍵。
2. 從快顯功能表中選擇 [特性]。

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時指定的所有特性。您可以變更這些特性中的任何一項，然後按一下 [確定] 以更新舊的資訊。

3. 按一下 [確定] 或 [取消] 以關閉此對話方塊。

連線至物件儲存

您必須連線至物件儲存之後才可以將物件新增至此儲存。

► 若要連線至物件儲存

1. 在瀏覽窗格中的 [MyObjectStore] 上按一下滑鼠右鍵。
2. 從快顯功能表中選擇 [連線至物件儲存]。

請注意物件儲存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。

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。

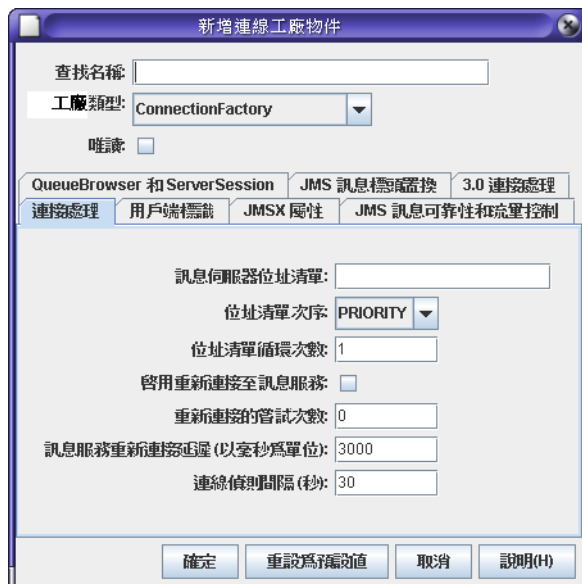
如需有關配置連線工廠的資訊，請參閱線上說明與 [Message Queue Developer's Guide for Java Clients](#)。

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

► 若要將連線工廠新增至物件儲存

1. 如果尚未連線，請連線至 MyObjectStore (請參閱第 55 頁「[連線至物件儲存](#)」)。

- 在 [連線工廠] 節點上按一下滑鼠右鍵，然後選擇 [新增連線工廠物件]。
畫面上將會顯示 [新增連線工廠物件] 對話方塊。



- 在 [查找名稱] 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 HelloWorldMessageJNDI.java 的以下行中所示：

```
qcf=(javax.jms.QueueConnectionFactory)
    ctx.lookup("MyQueueConnectionFactory")
```
- 從下拉式清單中選取 [QueueConnectionFactory]，以指定連線工廠的類型。
- 按一下 [連線處理] 標籤。
- 您通常可以在 [訊息伺服器位址清單] 欄位輸入用戶端連線之代理程式的位址。此欄位的範例如下所示：

```
mq://localhost:7676/jms
```

依預設，您不需輸入任何值，因為將連線工廠配置為在連接埠 7676 上的本地主機執行以連線至代理程式，此配置為本快速入門所預期的範例。
- 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 [新增連線工廠物件] 對話方塊右下角的 [說明] 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。
- 按一下 [確定] 以建立此佇列連線工廠。

9. 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。目標受管理物件可以讓用戶端查找並找到實體目標，並不需要提供者的特定目標名稱與配置。

當用戶端傳送訊息時，它會查找或創設目標受管理物件，並以 JMS API 的 `send()` 方法參照此物件。然後，代理程式負責將訊息傳送至與受管理物件相關聯的實體目標，如下所示：

- 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傳送至此實體目標。
- 如果您尚未建立實體目標且已啓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傳送至此目標。
- 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，亦無法傳送訊息。

在本快速入門的下一部分，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。

► 若要將目標新增至物件儲存

1. 在瀏覽窗格的中的 [目標] 節點 (於 [MyObjectStore] 節點下) 上按一下滑鼠右鍵。
2. 選擇 [新增目標物件]。

管理主控台會顯示 [新增目標物件] 對話方塊，您可用來指定有關此物件的資訊。



3. 在 [查找名稱] 欄位中輸入 MyQueue 。

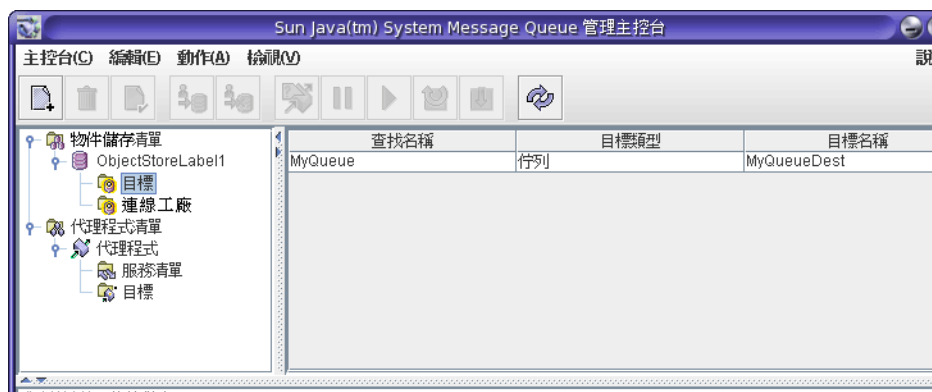
使用此查找名稱可以找到正在使用 JNDI 查找呼叫的物件。在範例應用程式中，此呼叫如下：

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

4. 為 [目標類型] 選擇 [佇列] 單選按鈕。
5. 在 [目標名稱] 欄位中輸入 MyQueueDest 。

此名稱為您在代理程式上新增實體目標時指定的名稱（請參閱第 47 頁「將實體目標新增至代理程式」）。

6. 按一下 [確定]。
7. 在瀏覽窗格中選取 [目標]，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。



檢視受管理物件特性

若要檢視或更新受管理物件的特性，您需要在瀏覽窗格中選取 [目標] 或 [連線工廠]，在結果窗格中選取特定的物件然後選擇 [動作] > [特性]。

► 若要檢視或更新目標物件的特性

1. 選取瀏覽窗格中 [MyObjectStore] 的 [目標] 節點。
2. 在結果窗格中選取 [MyQueue] 。
3. 選擇 [動作] > [特性] 以檢視 [目標物件特性] 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。

4. 按一下 [取消] 以關閉此對話方塊。



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 [檢視] > [重新顯示] 來更新任何元素或元素群組的可視顯示。

執行佇列應用程式

範例應用程式 HelloWorldMessageJNDI，可供您搭配快速入門使用。此範例應用程式使用您建立的實體目標和受管理物件：

- 佇列實體目標的名稱是 MyQueueDest
- 佇列連線工廠受管理物件和佇列受管理物件的 JNDI 查找名稱分別是 MyQueueConnectionFactory 和 MyQueue

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。

► 若要執行 HelloWorldMessageJNDI 應用程式

1. 使包含 HelloWorldmessageJNDI 應用程式的目錄成爲目前目錄；例如：

```
cd IMQ_HOME\demo\helloworld\helloworldmessagejndi (Windows)
```

```
cd /usr/demo/imq/helloworld/helloworldmessagejndi (Solaris)
```

```
cd /opt/sun/mq/examples/helloworld/helloworldmessagejndi (Linux)
```

您應會發現 `HelloWorldMessageJNDI.class` 檔案已存在 (如果您要變更此應用程式，就必須使用 `Message Queue Developer's Guide for C Clients` 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)。設定 `CLASSPATH` 變數，以納入目前包含檔案 `HelloWorldMessageJNDI.class` 的目錄與 `Message Queue` 產品的 JAR 檔案，JAR 檔案如下所示：`jms.jar`、`imq.jar` 和 `fscontext.jar`。請參閱 `Message Queue Developer's Guide for Java Clients`，以獲得設定 `CLASSPATH` 的說明。

JNDI jar 檔案 (`jndi.jar`) 隨附於 JDK 1.4。如果您在使用 JDK，您則不需新增 `jndi.jar` 至您的 `CLASSPATH` 設定。如果您是使用較早版本的 JDK，您則必須將 `jndi.jar` 納入您的 `CLASSPATH`。請參閱 `Message Queue Developer's Guide for Java Clients`，以取得其他資訊。

2. 執行此應用程式前，請開啓來源檔案 `HelloWorldMessageJNDI.java`，接著閱讀來源。它很簡短但已充分進行說明，並且有關您建立的受管理物件和目標的使用方法亦相當清楚。
3. 請執行以下其中一個指令，以執行 `HelloWorldMessageJNDI` 應用程式：

在 Windows 上為 `java HelloWorldMessageJNDI`

在 Solaris 和 Linux 上為 `% java HelloWorldMessageJNDI file:///tmp`

如果此應用程式執行成功，您應看到以下輸出：

```
java HelloWorldMessageJNDI
正在使用 file:///C:/Temp for Context.PROVIDER_URL

正以查找名稱查找佇列連線工廠物件：MyQueueConnectionFactory
找到佇列連線工廠物件。
正以查找名稱查找佇列物件：MyQueue
找到佇列物件。

正在建立與代理程式的連線。
已建立與代理程式的連線。

正在發佈訊息至佇列：MyQueueDest
已接受以下訊息：Hello World
```

執行範例應冊 程式

- 第 3 章 啓動代理程式和用戶端
- 第 4 章 配置代理程式
- 第 5 章 管理代理程式
- 第 6 章 管理實體目標
- 第 7 章 管理安全性
- 第 8 章 管理受管理物件
- 第 9 章 使用代理程式叢集
- 第 10 章 監視訊息伺服器
- 第 11 章 分析與調校訊息服務
- 第 12 章 排解疑難問題

啟動代理程式和用戶端

安裝 Sun Java™ System Message Queue 並執行一些基本準備步驟後，您即可以啟動代理程式和用戶端。

本章包含以下各節：

- [第 64 頁](#) 「準備系統資源」
- [第 64 頁](#) 「以互動方式啟動代理程式」
- [第 66 頁](#) 「自動啟動代理程式」
- [第 68 頁](#) 「啟動 Message Queue 用戶端」
- [第 69 頁](#) 「移除代理程式實例」

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，這些選項覆寫配置檔案中的相應特性。如需代理程式配置的詳細資訊，請參閱[第 71 頁第 4 章](#) 「配置代理程式」。

準備系統資源

啟動代理程式之前，必須先執行兩個系統級別的工作：同步化系統時鐘，以及設定 Solaris 或 Linux 的檔案描述元限制。後面幾節將描述這些工作。

同步化系統時鐘

啟動任何代理程式或用戶端之前，必須先同步化會與 Message Queue 系統互動的所有主機的時鐘。如果您使用訊息逾時 (TimeToLive)，則同步化尤為重要。不同步的時鐘時間戳記可能讓 TimeToLive 功能無法正常運作或無法傳送訊息。同步化對代理程式叢集也十分重要。

您應該配置系統以執行時間同步化協定，例如 Simple Network Time Protocol (SNTP)。Solaris 和 Linux 的 xntpd 常駐程式和 Windows 的 W32Time Time 服務通常支援時間同步化功能。如需配置此服務的詳細資訊，請參閱作業系統的文件。

代理程式執行後，請避免回頭設定系統時鐘。

設定檔案描述元限制 (Solaris 或 Linux)

Solaris 和 Linux 平台上，在其中執行用戶端或代理程式的 Shell 對程序可以使用的檔案描述元的數目指定了軟式限制。在 Message Queue 系統中，用戶端建立的每個連線或代理程式接受的每個連線，均使用這些檔案描述元的其中之一。每個擁有永久性訊息的實體目標皆使用一個檔案描述元。

因此，這些因素會限制連線數目。如果不變更檔案描述元限制，您的代理程式或用戶端便無法在 Solaris 上支援 256 個以上的連線，或在 Linux 上支援 1024 個以上的連線。(連線限制實際上低於上述數目，因為會持續使用檔案描述元。)

若要變更檔案描述元限制，請參閱 ulimit 線上手冊。您需要在用戶端或代理程式將在其中執行的每個 shell 中變更此限制。

以互動方式啟動代理程式

您可以在指令行使用 imqbrokerd 指令，以互動方式啟動代理程式。(或者，您可以從 Windows 的 [開始] 功能表，啟動代理程式。) 您不能使用管理主控台 (imqadmin) 或指令公用程式 (imqcmd) 來啟動代理程式，因為您必須先執行代理程式，才能使用這些工具。

Solaris 和 Linux 平台上的代理程式實例，必須由最初啟動平台的使用者來啟動。代理程式實例初次啟動時，Message Queue 使用這位使用者的 `umask` 來設定代理程式實例目錄 (包含配置資訊和永久性資料) 的許可權。每個代理程式實例都有自己的配置特性和基於檔案的訊息儲存組。

代理程式實例的預設名稱是 `imqbrokerd`。您只要使用下列指令，即可從指令行以這個名稱和預設配置來啟動代理程式

```
imqbrokerd
```

這個指令在本地機器上使用預設連接埠 7676 上的連接埠對映器，啟動代理程式實例 `imqbrokerd`。

若要指定非預設的實例名稱，請在 `imqbrokerd` 指令上使用 `-name` 選項。下列指令啟動的代理程式實例的名稱是 `myBroker`：

```
imqbrokerd -name myBroker
```

您還可以在 `imqbrokerd` 指令行上使用其他選項來控制代理程式作業的各種狀況。下列範例使用 `-tty` 選項，向指令視窗 (標準輸出) 傳送錯誤和警告：

```
imqbrokerd -name myBroker -tty
```

您也可以指令行上使用 `-D` 選項來覆寫代理程式實例配置檔案 (`config.properties`) 中指定的特性值。這個範例設定 `imq.jms.max_threads` 特性，將 `jms` 連線服務可用的最大執行緒數目，提高到 2000：

```
imqbrokerd -name myBroker -Dimq.jms.max_threads=2000
```

如需 `imqbrokerd` 指令的語法、子指令和選項的詳細資訊，請參閱第 13 章「指令參照」。如需這項資訊的快速摘要，請輸入下列指令：

```
imqbrokerd -help
```

備註 如果您有 Sun Java System Message Queue Platform Edition 授權，您可以使用 `imqbrokerd` 指令的 `-license` 選項，啟動 Enterprise Edition 試用授權，試用 Enterprise Edition 功能 90 天。請指定授權名稱 `try`：

```
imqbrokerd -license try
```

您每次啟動代理程式時都必須使用此選項，否則它會預設為標準 Platform Edition 授權。

自動啟動代理程式

除了明確的從指令行啟動代理程式之外，您還可以設定在系統啟動時自動啟動代理程式。執行方式因執行代理程式的平台 (Solaris、Linux 或 Windows) 而異。

在 Solaris 和 Linux 自動啟動

在 Solaris 和 Linux 系統上安裝 Message Queue 時，可以自動啟動的程序檔放在 `/etc/rc*` 樹狀目錄中。若要使用這些程序檔，您必須如下所示，編輯配置檔案 `/etc/imq/imqbrokerd.conf` (Solaris) 或 `/etc/opt/sun/mq/imqbrokerd.conf` (Linux)：

- 若要在系統啟動時自動啟動代理程式，請將 `AUTOSTART` 特性設定為 `YES`。
- 若要在異常結束後自動重新啟動代理程式，請將 `RESTART` 特性設定為 `YES`。
- 若要設定啟動代理程式的指令行引數，請指定一或多個 `ARGS` 特性值。

在 Windows 自動啟動

若要在 Windows 系統啟動時自動啟動代理程式，您必須將代理程式定義為 Windows 服務。在 Windows 系統上安裝 Message Queue 時，您可以將代理程式安裝為服務。安裝後，您可以使用服務管理員公用程式 `imqsvcadmin` 執行下列作業：

- 將代理程式新增為 Windows 服務。
- 決定代理程式服務的啟動選項。
- 移除以 Windows 服務身份執行的代理程式。

如需 `imqsvcadmin` 指令的語法、子指令和選項的參照資訊，請參閱第 13 章「[指令參照](#)」。

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。

若要將啟動選項傳給代理程式，您可以在 `imqsvcadmin` 指令中使用 `-args` 引數。這等同於第 64 頁「[以互動方式啟動代理程式](#)」中描述的 `imqbrokerd` 指令的 `-D` 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：

- 原生 Windows 服務包裝 `imqbrokersvc.exe`
- 執行代理程式的 Java 執行階段

系統只有一個代理程式作為 Windows 服務執行。

重新配置代理程式服務

重新配置 Windows 服務的順序如下：

1. 停止服務。
2. 移除服務。
3. 新增服務，使用 `-args` 選項指定不同的代理程式啟動選項，或使用 `-vmargs` 選項指定不同的 Java 版引數。

使用替代 Java 執行階段

您可以使用 `-javahome` 或 `-jrehome` 選項來指定替代 Java 執行階段的位置。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。

[啟動參數] 欄位將反斜線 (\) 視為退出字元，因此當您以反斜線來區隔路徑時，必須輸入兩次；例如，

```
-javahome d:\\jdk1.3。
```

顯示代理程式服務啟動選項

若要決定代理程式服務的啟動選項，請在 `imqsvcadmin` 指令中使用 `query` 選項。

```
imqsvcadmin query  
  
已安裝服務 iMQ_Broker。  
顯示名稱：iMQ_Broker  
啟動類型：手動  
二進位位置：c:\Program Files\Sun Microsystems\  
Message Queue 3.5\bin\imqbrokersvc  
JavaHome：c:\j2sdk1.4.0  
代理程式引數：-passfile d:\imqpassfile
```

服務啟動問題疑難排解

啟動服務時如果出現錯誤，您可以檢視已記錄的錯誤事件。

► 查閱記錄的服務錯誤事件

1. 啟動 [事件檢視器]。
2. 查看 [記錄] > [應用程式]。
3. 選取 [檢視] > [重新顯示]，以查閱所有錯誤事件。

移除當作 Windows 服務執行的代理程式

若要移除當作服務執行的代理程式，請執行下列其中一項：

- 使用指令。先使用 `imqcmd shutdown bkr` 指令關閉代理程式，然後使用 `imqsvcadmin remove` 指令移除服務。
- 使用 Windows 服務的控制台管理工具。這個功能在不同 Windows 版本中有不同的位置。

完成後，重新啟動電腦。

啟動 Message Queue 用戶端

啟動用戶端應用程式之前，您必須先向應用程式開發者取得系統的設定資訊。如果您啟動 Java 用戶端應用程式，您必須設定 `CLASSPATH` 變數，並且確定已安裝正確的 JAR 檔案。Message Queue Developer's Guide for Java Clients 中有一般的系統設定步驟資訊，但是您的開發者可能會提供其他資訊。

啟動 Java 用戶端應用程式時，請使用下列的指令行格式：

```
java clientAppName
```

啟動 C 用戶端應用程式時，請使用應用程式開發者提供的格式。

應用程式開發者或應用程式文件應該提供應用程式設定的屬性值資訊。您可能要覆寫應用程式設定的某些屬性。您可以在指令行中指定要覆寫的屬性。

您可能還要在指令行中為任何使用 JNDI 查找功能來尋找連線工廠的 Java 用戶端，指定屬性。如果查找功能傳回的連線工廠比應用程式還舊，這表示連線工廠不支援新的屬性。此時，Message Queue 會將這些屬性設定為預設值。您可以在指令行中指定屬性，將這些屬性設定為非預設值。

在指令行中提供屬性值時，請使用 Java 應用程式的下列指令行語法：

```
java [-Dattribute=value ]... clientAppName
```

attribute 值必須如第 16 章「受管理物件屬性參照」中所描述，是連線工廠受管理物件屬性。如果值中有空格，請將指令行的整個 *attribute=value* 部分括在引號內。

下列範例啟動用戶端應用程式 MyMQClient。此應用程式連線至連接埠 7677 上主機 OtherHost 的代理程式，覆寫應用程式設定的任何主機名稱和連接埠。

```
java -DimqAddressList=mq://OtherHost:7677/jms MyMQClient
```

有時您無法使用指令行來指定屬性值。管理員可以設定受管理物件只允許讀取，或應用程式開發者以程式碼讓用戶端如此操作。若要瞭解啟動用戶端程式的最佳方法，就必須與應用程式開發者溝通。

移除代理程式實例

本節提供在 Solaris 或 Linux 上移除代理程式實例的資訊。如需移除 Windows 服務的詳細資訊，請參閱第 68 頁「移除當作 Windows 服務執行的代理程式」。

若要移除代理程式實例，請在 `imqbrokerd` 指令中使用 `-remove` 選項。移除代理程式實例的指令格式如下：

```
imqbrokerd [options..] -remove instance
```

例如，如果代理程式的名稱是 `myBroker`，則指令如下：

```
imqbrokerd -name myBroker -remove instance
```

此指令移除該指定代理程式的整個實例目錄。

如需移除代理程式時可用的選項清單，請參閱第 255 頁「指令參照」中的 `imqbrokerd` 參照資訊。

如果 Solaris 或 Linux 上的代理程式設定為在系統啟動時自動啟動，請編輯配置檔案 `/etc/imq/imqbrokerd.conf` (Solaris) 或 `/etc/opt/sun/mq/imqbrokerd.conf` (Linux)，並將 `AUTOSTART` 特性設定為 `NO`。

移除代理程式實例

配置代理程式

代理程式實例啓動時，它的配置由一組配置檔案和傳給 `imqbrokerd` 指令的選項來管理。本章說明配置檔案和指令行選項如何互動配置代理程式實例、描述每個代理程式元件的功能及列出其配置特性，並解釋如何設定配置。

本章包含以下各節：

- [第 72 頁](#) 「關於可配置的代理程式元件」
- [第 90 頁](#) 「關於配置檔案」
- [第 92 頁](#) 「編輯實例配置檔案」
- [第 93 頁](#) 「在指令行上輸入配置選項」
- [第 93 頁](#) 「設定永久性儲存」
- [第 98 頁](#) 「保護永久性資料」

如需配置特性的完整參照資訊，請參閱[第 14 章](#) 「代理程式特性參照」。

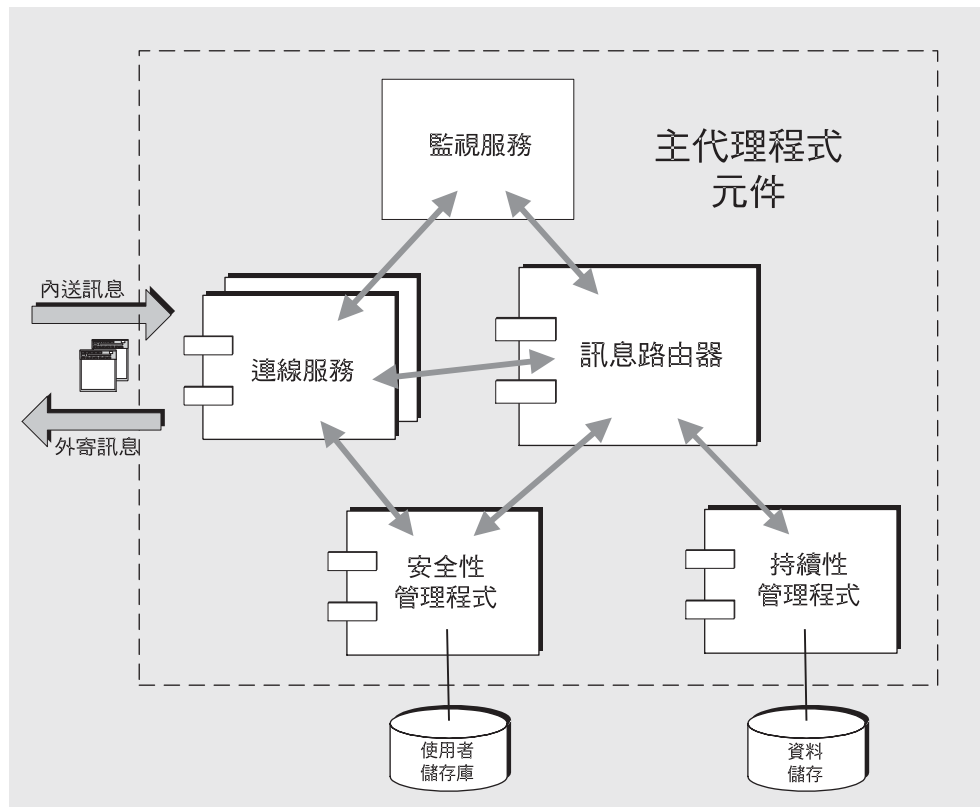
關於可配置的代理程式元件

Message Queue 訊息傳送系統中的訊息傳送 - 從生產型用戶端至目標, 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。

若要執行訊息傳送, 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、確保可靠傳送, 並為監視系統效能提供資料。

若要執行其功能, 代理程式將使用一些內部元件, 並且每個元件在遞送過程中都有特定的角色。圖 4-1 中說明了這些代理程式元件。

圖 4-1 代理程式服務元件



訊息路由器元件執行主要的訊息路由和傳送服務，其他元件則提供重要的支援服務。
表 4-1 中簡單描述每個元件。

表 4-1 主代理程式服務元件和功能

元件	說明 / 功能	特性說明 ...
連線服務	管理代理程式與用戶端之間的實體連線，為內送和外寄的訊息提供傳輸。	第 285 頁「連線服務特性」
訊息路由器	管理訊息的路由與傳送：這些訊息包括 JMS 訊息以及 Message Queue 訊息傳送系統為支援 JMS 訊息傳送而使用的控制訊息。	第 287 頁「訊息路由器特性」
持續性管理程式	管理永久性儲存體的資料寫入和擷取。	第 289 頁「持續性管理程式特性」
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）。	第 293 頁「安全性管理程式特性」
監視服務	產生可以寫入大量輸出通道的度量與診斷資訊，另外，管理員可以使用輸出通道來監視和管理代理程式。	第 297 頁「監視和記錄特性」

根據載入條件、應用程式的複雜性等，您可以配置這些元件，最佳化代理程式的效能。下面幾節介紹每個元件所執行的功能和您可以設定來影響元件運作方式的特性。

連線服務

Message Queue 代理程式支援與 Message Queue 應用程式用戶端和 Message Queue 管理用戶端的通訊。每種連線服務均由其服務類型和協定類型指定如下：

- *服務類型*指定此服務是否提供 JMS 訊息傳送 (NORMAL) 或是 Message Queue 管理 (ADMIN) 服務
- *協定類型*指定支援此服務的基本傳輸協定層。

表 4-2 列出 Message Queue 代理程式可以使用的連線服務：

表 4-2 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	tcp
ssljms (Enterprise Edition)	NORMAL	tls (基於 SSL 的安全性)

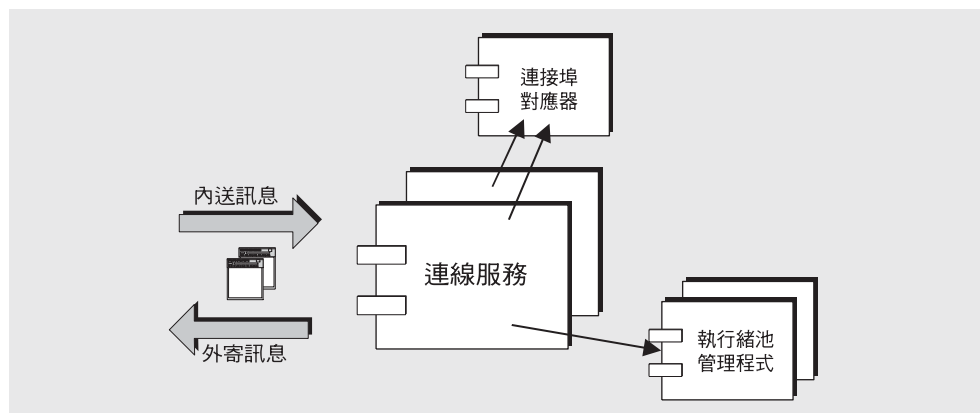
表 4-2 代理程式支援的連線服務 (續)

服務名稱	服務類型	協定類型
httpjms (Enterprise Edition)	NORMAL	http
httpsjms (Enterprise Edition)	NORMAL	https (基於 SSL 的安全性)
admin	ADMIN	tcp
ssladmin	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。依預設，會啟用 jms 和 admin 服務。

Message Queue 可以將連線服務動態對映到連接埠號，或者您可以明確的指派連接埠。每種服務皆使用共用連接埠對映器登錄，但它們均有自己的執行緒池管理程式，如圖 4-2 所示。

圖 4-2 連線服務支援



下面幾節描述連接埠對映器和執行緒池管理程式之間的關係。

連接埠對映器

Message Queue 會提供一個 *連接埠對映器*，以指定連線服務的連接埠。連接埠對映器常駐在標準的連接埠號 7676。當用戶端設定與代理程式連線時，它先聯絡連接埠對映器，要求提供所指定的連線服務的連接埠號。

jms、ssljms、admin 和 ssladmin 連線服務的可以是動態或靜態。依預設，連線服務啟動時，會動態配置其連接埠。或者，您可以為服務指定靜態連接埠，但通常不建議靜態連接埠號。靜態連接埠號通常只用於連線遍訪防火牆等特殊狀況。

使用附錄 C「HTTP/HTTPS 支援」的第 345 頁表 C-1 和第 355 頁表 C-3 中描述的特性，可分別配置 httpjms 和 httpsjms 服務。

執行緒池管理程式

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，在執行緒池管理程式元件所管理的執行緒池中維護。

您可以配置此執行緒池管理程式，以設定此執行緒池中所維護的執行緒的最小數目和最大數目。由於連線需要執行緒，因此可將執行緒新增至執行緒池。當執行緒超過最小數目時，系統將關閉執行緒（因為這些執行緒將成為自由執行緒，直至達到最小數目的臨界值），從而節省記憶體資源。此數目必須夠大，才不必繼續建立新的執行緒。在大量連線負載下，執行緒的數目會增加，直至達到執行緒池的最大數目，然後連線必須等待，直至某執行緒變為可用。

執行緒池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 每個到代理程式的連線均要求兩個專屬執行緒：一個處理連線內送的訊息，另一個處理連線外寄的訊息。這可將連線數目限制為執行緒池中最大執行緒數目的一半，但它可提供高性能。

共用模型 (Enterprise Edition) 傳送或接收訊息時，共用執行緒會負責處理連線。因為每個連線皆沒有要求專屬執行緒，所以此模型會增加連線服務（和代理程式）可支援的連線數目。但也會增加共用執行緒中效能耗用的時間。執行緒池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。透過限制每個此類分散執行緒監視的連線數目，在此活動中效能耗用的時間可以縮到最短。

安全性

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 83 頁「**安全性管理程式**」）。

連線服務特性

這些是連線服務相關的可配置特性：

- `imq.service.activelist`。代理程式啟動時啟動的連線服務清單。
- `imq.hostname`。如果有多台主機可用（例如，如果一台電腦中有多張網路介面卡），則指定所有連線服務所連結的主機。

- `imq.portmapper.port`。指定代理程式的主要連接埠 - 連接埠對映器常駐的連接埠。
- `imq.portmapper.hostname`。如果有多個可用主機，指定連接埠對映器連結到的主機。
- `imq.portmapper.backlog`。指定拒絕請求前，連接埠對映器可以處理運作請求的最大數目。此特性可以設定請求數目，這些請求可儲存在作業系統儲存區，等待連接埠對映器進行處理。
- `imq.service_name.protocol_type.port`。僅適用於 `jms`、`ssljms`、`admin` 和 `ssladmin` 服務，為已命名的連線服務指定連接埠號。
- `imq.service_name.protocol_type.hostname`。僅適用於 `jms`、`ssljms`、`admin` 和 `ssladmin` 服務，有多個可用主機時，指定已命名的連線服務連結到的主機。
- `imq.service_name.min_threads`。指定執行緒數目，一旦達到指定數目，執行緒便會在執行緒池中維護，供已命名的連線服務使用。
- `imq.service_name.max_threads`。指定執行緒數目，一旦超過指定數目，系統便不會將新的執行緒新增至執行緒池，供已命名的連線服務使用。
- `imq.service_name.threadpool_model`。指定當命名的連線服務有需要時，執行緒是專屬於連線或是由連線共用。
- `imq.shared.connectionMonitor_limit`。僅適用於共用執行緒池模型，指定可由分散執行緒監視的最大連線數目。

如需這些特性的完整說明，請參閱第 285 頁表 14-2。

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傳送。

基本的傳送機制

概括地講，代理程式所處理的訊息分為兩類：

- JMS 有效負載訊息是由生產者用戶端傳送且目標為使用者用戶端的訊息
- 控制訊息是為了支援 JMS 訊息傳送而與用戶端之間相互傳送的訊息

如果內送的訊息為 JMS 訊息，則代理程式將基於此訊息的目標是佇列或主題，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使用中用戶。如果長期用戶處於非使用中，則訊息路由器將保留此訊息，直至此用戶成為使用中用戶，再將訊息傳送至此用戶。
- 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傳送至適當的用戶。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。

一旦訊息路由器將訊息傳送至所有預定的用戶，它便會將此訊息從記憶體中清除。如果此訊息為永久性的，訊息路由器會將其從代理程式的永久性資料儲存中移除。

可靠傳送：確認和作業事件

當為可靠傳送新增需求時，剛才描述的傳送機制會變得更加複雜。可靠傳送包含兩個方面：

- 確保至代理程式和來自代理程式的訊息傳送成功
- 確保實際傳送訊息之前，代理程式不會遺失訊息或傳送資訊

為確保訊息成功遞送至代理程式及從其中成功發出，Message Queue 使用了一些回應控制訊息。

例如，生產者向目標傳送 JMS 訊息（有效負載訊息）時，代理程式會回應已經收到 JMS 訊息。（依預設，僅當產生者指定 JMS 訊息為永久性時，Message Queue 才會執行此作業）。生產型用戶端使用代理程式回應，以確保傳送至目標。

同樣，當代理程式將 JMS 訊息傳送至用戶時，此使用用戶端會傳送回它已接收到並已處理此訊息的確認。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確認，但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，從主題的多重用戶的每個用戶）接收到確認，則它不會從記憶體中刪除 JMS 訊息。

如果主題有多個長期訂閱，則訊息路由器會將每個 JMS 訊息保留在目標中，並當每個長期用戶成為使用中用戶時傳送此訊息。

訊息路由器接收用戶端確認時會記錄這些確認，並僅在已接收所有確認後才刪除 JMS 訊息，除非 JMS 訊息在此之前已過期。

而且，透過將代理程式確認傳送回用戶端，訊息路由器確認用戶端確認的接收。使用用戶端使用代理程式回應，以確保代理程式不會多次傳送同一 JMS 訊息。如果代理程式無法接收到用戶端確認，則可能發生上述情況。

如果代理程式沒有接收到用戶端確認並再次傳送 JMS 訊息，則此訊息會標記有重新傳送旗標。在下列情況下，代理程式通常會重新傳送 JMS 訊息：

- 用戶端連線在代理程式接收到用戶端確認之前關閉，並隨後開啓新的連線。

- 用戶端應用程式回復階段作業。
- 用戶端應用程式回復回轉作業事件。

例如，如果佇列的訊息用戶在確認訊息之前離線，而且另一個用戶隨後註冊此佇列，則代理程式將未確認的訊息重新傳送至此新的用戶。

以上描述的用戶端確認與代理程式回應，還套用於群組至作業事件中的 JMS 訊息傳送。在這些情況下，這些過程可對作業事件作業，亦可對傳送或接收的個別 JMS 訊息作業。確定作業事件時，代理程式回應會被自動傳送。

代理程式會追蹤作業事件，並可對作業事件進行確定或在發生故障時回轉。本作業事件管理還支援本機作業事件，此為較大型分散式作業事件的一部分。代理程式會追蹤這些作業事件的狀態，直至它們被確定。代理程式啟動時，它會檢查所有未確定的作業事件，並依預設回轉所有作業事件（那些處於 PREPARED 狀態的作業事件除外）。如果您有設定 `imq.transaction.autorollback` 特性，代理程式還會回轉處於 PREPARED 狀態的作業事件。

可靠傳送：持續性

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

如果生產者用戶端指定訊息為永久性的，訊息路由器會將此訊息傳送至**持續性管理程式**。持續性管理程式會將訊息儲存在資料庫或檔案系統（請參閱第 80 頁「[持續性管理程式](#)」）以便在代理程式出現故障時回復此訊息。

管理記憶體資源和訊息流量

代理程式的效能和穩定性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。要特別注意的是，訊息產生速度快於使用速度時，會使用到所有的記憶體資源，因而訊息路由器可能過度負載。若要預防發生此情形，訊息路由器會使用三個記憶體保護級別，以在資源不足時維持系統作業：

個別目標上的訊息限制 您可以設定實體目標特性，以指定訊息數目限制和訊息使用的總記憶體限制（請參閱第 15 章「[實體目標特性參照](#)」）。您也可以指定訊息路由器到達這些限制時的運作方式。四個限制運作方式為：

- 減緩訊息產生者速度 (FLOW_CONTROL)
- 拋出記憶體中最舊的訊息 (REMOVE_OLDEST)
- 根據訊息存在時間，拋出記憶體中具最低優先權的訊息 (REMOVE_LOW_PRIORITY)
- 拒絕最新的訊息 (REJECT_NEWEST)

系統範圍訊息限制 系統範圍訊息限制為第二道保護。您可以指定套用於系統上之所有目標的系統範圍限制：訊息總數目和所有訊息使用的記憶體（請參閱第 287 頁表 14-3）。如果到達任何系統範圍訊息限制，訊息路由器會拒絕新的訊息。

系統記憶體臨界值 系統記憶體臨界值為第三道保護。您可以指定代理程式採取越來越重要的動作時可用的系統記憶體臨界值，以防止記憶體超過負載。此動作取決於記憶體資源的下列狀態：

- 綠色（大量記憶體可用）
- 黃色（代理程式記憶體正在減少）
- 橙色（代理程式記憶體不足）
- 紅色（代理程式無記憶體可用）。

在代理程式的記憶體狀態從綠色變為黃色和橙色，然後變為紅色的過程中，代理程式會採取越來越重要的動作，動作類型如下所示：

- 將訊息移出使用中記憶體並放入永久性儲存體（請參閱第 80 頁「持續性管理程式」）。
- 減低非永久性訊息的產生者，最終使訊息停止流入代理程式。永久性訊息流量會自動受到代理程式確認每個訊息之要求的限制。

上述兩種方法皆會降低效能。

如果到達系統記憶體臨界值，那表示目標訊息限制和系統範圍訊息限制太小。在某些情況下，臨界值無法及時擷取潛在的記憶體負載。因此，您不應依賴此功能來控制記憶體資源，而是應個別且整體地配置目標，以最佳化記憶體資源。

訊息路由器特性

這些是管理記憶體資源的系統範圍限制和系統記憶體臨界值：

- `imq.destination.DMQ.truncateBody`。指定停用的訊息佇列只包含訊息的標頭和特性資料。捨棄訊息內文的內容。
- `imq.message.expiration.interval`。指定回收已過期訊息發生的頻率（以秒為單位）。
- `imq.system.max_count`。指定代理程式保留訊息的最大數目。
- `imq.system.max_size`。指定代理程式保留的最大訊息總量。
- `imq.message.max_size`。指定訊息內文的最大值。
- `imq.resource_state.threshold`。指定啟動每種記憶體狀態時的記憶體利用百分比。

- `imq.resource_state.count`。指定在啓動每個記憶體資源狀態時，批次中允許內送訊息的最大數目。
- `imq.transaction.autorollback`。指定處於 PREPARED 狀態的分散式作業事件是否可在啓動代理程式時自動回轉。

如需這些特性的完整說明，請參閱第 287 頁表 14-3。

持續性管理程式

若要回復代理程式（萬一代理程式發生故障），則此代理程式需要重新建立其訊息傳送作業的狀態。此操作要求它將所有永久性訊息以及基本的路由與傳送資訊儲存至資料儲存。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，不僅只要復原未傳送的訊息。代理程式還必須可執行以下作業：

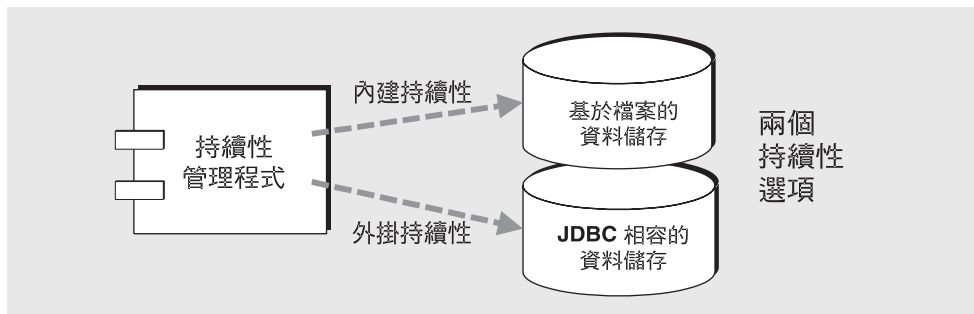
- 重新建立目標
- 復原每個主題的長期訂閱清單
- 復原每個訊息的確認清單
- 重新生產所有已確定作業事件的狀態

持續性管理程式管理所有此狀態資訊的儲存和擷取。

重新啓動代理程式時，它會重新建立目標和長期訂閱、回復永久性訊息、復原所有作業事件的狀態，並為未傳送訊息重新建立其路由表格。然後，它可以繼續訊息傳送。

Message Queue 支援內建和外掛持續性模組（請參閱圖 4-3）。內建持續性是基於檔案的資料儲存。外掛持續性使用 Java 資料庫連結 (JDBC™) 介面，並需要 JDBC 資料儲存。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 4-3 持續性管理程式支援



內建持續性

預設 Message Queue 永久性儲存體解決方案為基於檔案的資料儲存。這種方法使用個別檔案儲存永久性資料，如訊息、目標、長期訂閱以及作業事件。

基於檔案的資料儲存位於以代理程式實例 (*instanceName*) 作為辨別依據的目錄中，且代理程式實例與資料儲存相互關聯（請參閱附錄 A 「Message Queue 資料的作業系統特定位置」）：

```
.../instances/instanceName/fs350/
```

建立基於檔案的資料儲存，以便根據永久性訊息於目標的常駐位置，將永久性訊息儲存在目錄中。大部分訊息會儲存在由智慧型記錄所組成的單一檔案中。

若要減少因新增和移除訊息而產生的分段程序，您可以壓縮智慧型記錄檔案（請參閱第 127 頁「壓縮實體目標」）。除此之外，內建的持續性管理程式會將大小超過可配置臨界值的訊息 (`imq.persist.file.message.max_record_size`) 儲存在其個別檔案中，而非儲存於智慧型記錄檔案。針對這些個別檔案，要維護檔案池，以便檔案被重新使用。當不再需要訊息檔案時，系統不會刪除它，而是將它加入目標目錄中的可用檔案池中，用來儲存新的訊息。

您可以在目標檔案池中配置檔案的最大數目 (`imq.persist.file.destination.message.filepool.limit`)。您還可以指定此檔案池中被截斷至零而清除，而不是僅標記為重新使用的可用檔案的百分比 (`imq.persist.file.message.filepool.cleanratio`)。清除檔案的百分比越高，磁碟空間就越少，維護此檔案池所需的耗用時間就越多。

您可以指定是否要在關機時清除已標記檔案 (`imq.persist.file.message.cleanup`)。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。

所有其他永久性資料 (目標、長期訂閱和作業事件) 皆儲存在個別檔案中。所有目標在一個檔案中，所有長期訂閱在另一個檔案中，以此類推。

若要最大化可靠性，您可以使用 `imq.persist.file.sync.enabled` 屬性，指定持續性作業應該讓實體儲存裝置與內部記憶體狀態同步。這可以消除因系統當機而產生的資料遺失，但會影響效能。如果您在 **Sun Cluster** 環境中執行 **Message Queue**，必須將叢集中所有節點的此屬性設定為 `true`。

因為資料儲存可以包含敏感或專用性質的訊息，所以您應該保護 `instances/instanceName/fs350/` 目錄，以防未經授權的存取。如需說明，請參閱 [第 98 頁「保護永久性資料」](#)。

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。這包含設定一些與 **JDBC** 相關的代理程式配置特性，以及使用資料庫管理員公用程式 (`imqdbmgr`) 來建立具有適當模式的資料儲存。[第 93 頁「設定永久性儲存」](#) 中詳細介紹了這些程序和相關的配置特性。

持續性管理程式特性

這個特性指定您使用的持續性類型：

- `imq.persist.store`。指定代理程式使用內建、基於檔案 (檔案) 的持續性，或使用外掛 **JDBC** 相容 (`jdbc`) 的持續性。

這些特性屬於內建持續性：

- `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。
- `imq.persist.file.message.max_record_size`。指定訊息新增到訊息儲存檔案的最大容量。
- `imq.persist.file.destination.message.filepool.limit`。指定目標檔案池中，可供重新使用之可用檔案的最大數目。
- `imq.persist.file.message.filepool.cleanratio`。指定目標檔案池中被維護處於清除狀態之可用檔案的百分比 (截斷至零)。
- `imq.persist.file.message.cleanup`。指定代理程式關閉時，是否會清除目標檔案池中的可用檔案。

如需這些特性的完整說明，請參閱 [第 290 頁表 14-6](#)。

這些特性屬於基於 JDBC 的持續性：

- `imq.persist.jdbc.brokerid`。指定代理程式實例識別碼，以附加至資料庫中多個代理程式實例使用的表格名稱。
- `imq.persist.jdbc.driver`。指定 JDBC 驅動程式的 Java 類別名稱，以連線至資料庫。
- `imq.persist.jdbc.opendburl`。指定資料庫 URL，以開啓與現有資料庫的連線。
- `imq.persist.jdbc.createdburl`。指定資料庫 URL，以開啓連線建立資料庫。
- `imq.persist.jdbc.closedburl`。指定資料庫 URL，以在關閉代理程式時關閉目前資料庫連線。
- `imq.persist.jdbc.user`。如有必要，則指定用於開啓資料庫連線的使用者名稱。
- `imq.persist.jdbc.needpassword`。指定資料庫是否需要密碼以存取代理程式。
- `imq.persist.jdbc.password`。如有必要，則指定用於開啓資料庫連線的密碼。
- `imq.persist.jdbc.table.IMQSV35`。用來建立版本表格的 SQL 指令。
- `imq.persist.jdbc.table.IMQCCREC35`。用來建立配置變更記錄表格的 SQL 指令。
- `imq.persist.jdbc.table.IMQDEST35`。用來建立目標表格的 SQL 指令。
- `imq.persist.jdbc.table.IMQINT35`。用來建立偏好表格的 SQL 指令。
- `imq.persist.jdbc.table.IMQMSG35`。用來建立訊息表格的 SQL 指令。
- `imq.persist.jdbc.table.IMQPROPS35`。用來建立特性表格的 SQL 指令。
- `imq.persist.jdbc.table.IMQILIST35`。用來建立狀態表格的 SQL 指令。
- `imq.persist.jdbc.table.IMQTXN35`。用來建立作業事件表格的 SQL 指令。
- `imq.persist.jdbc.table.IMQTACK35`。用來建立作業事件確認表格的 SQL 指令。

如需這些特性的完整說明，請參閱第 291 頁表 14-7。

安全性管理程式

Message Queue 提供認證與授權 (存取控制) 功能，還支援加密功能。

認證與授權功能因使用者儲存庫而異 (請參閱第 85 頁圖 4-4)：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)。

Message Queue 管理員可寫入 Message Queue 所提供的使用者儲存庫 (請參閱第 132 頁「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 (請參閱第 137 頁「將 LDAP 伺服器用於使用者儲存庫」)。

認證

Message Queue 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 85 頁「加密」。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。

所有安全性管理程式特性都在第 86 頁「安全性管理程式特性」中列出，並在第 86 頁「安全性管理程式特性」中詳細描述。

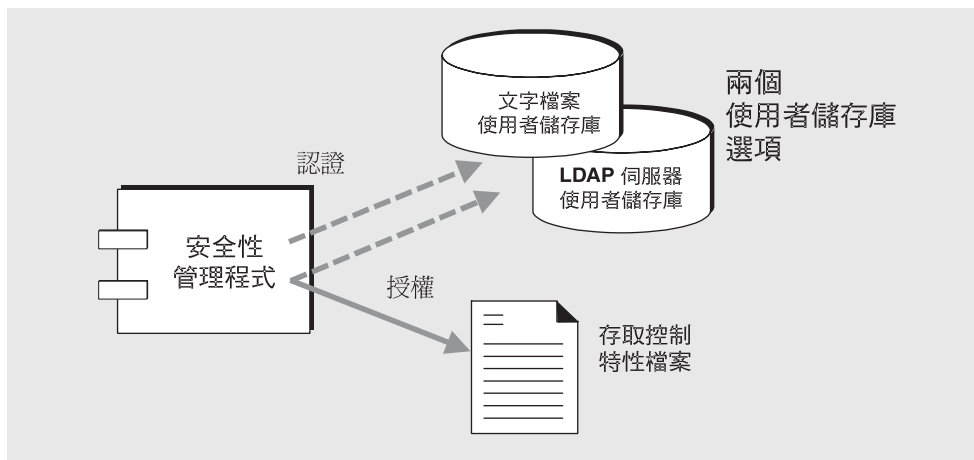
授權

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 Message Queue 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 Message Queue 作業的許可權。您可在存取控制特性檔案 (請參閱圖 4-4) 中指定這些存取控制。

當使用者嘗試執行作業時，安全性管理程式會從使用者儲存庫檢查使用者名稱和群組成員身份，是否與存取控制特性檔案中指定存取此作業的名稱和成員身份相符。存取控制特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用戶、產生者或佇列瀏覽器
- 自動建立目標

圖 4-4 安全性管理程式支援



預設存取控制特性檔案僅明確參照一個群組：*admin*（請參閱第 135 頁「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 Message Queue 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）。

透過編輯存取控制特性檔案，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。如果代理程式已配置為允許自動建立目標，則您可以透過編輯存取控制特性檔案來控制代理程式可為哪些使用者和群組自動建立目標。

所有安全性管理程式特性都在第 86 頁「安全性管理程式特性」中列出，並在第 86 頁「安全性管理程式特性」中詳細描述。

加密

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 Message Queue 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憑證中，並放置在 Message Queue 密鑰儲存區中。Message Queue 密鑰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密鑰儲存區密碼。請參閱第 146 頁「使用基於 SSL 的服務」。

一旦密鑰儲存區解除鎖定，則代理程式可將憑證傳送至請求連線的任何用戶端。然後，用戶端使用此憑證設定與代理程式的已加密連線。

所有安全性管理程式特性都在下一節中列出，並在第 86 頁「安全性管理程式特性」中詳細描述。

安全性管理程式特性

這些特性是用於認證、授權、加密以及其他安全通訊的可配置特性：

- `imq.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。
- `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。
- `imq.authentication.basic.user_repository`。指定基本 64 程式碼認證用的使用者儲存庫類型 (基於檔案或 LDAP)。
- `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)。
- `imq.accesscontrol.enabled`。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 Message Queue 作業的許可權 (如存取控制特性檔案中所指定)。
- `imq.service_name.accesscontrol.enabled`。為已命名的連線服務設定存取控制 (true/false)，並覆寫代理程式範圍的設定。
- `imq.accesscontrol.file.filename`。為代理程式實例支援的所有連線服務，指定存取控制特性檔案的名稱。
- `imq.service_name.accesscontrol.file.filename`。為代理程式實例的已命名連線服務，指定存取控制特性檔案的名稱。
- `imq.passfile.enabled`。指定是否已在檔案中指定用於安全通訊的使用者密碼 (用於 SSL、LDAP、JDBC™)。
- `imq.passfile.dirpath`。指定包含密碼檔案之目錄的路徑。
- `imq.passfile.name`。指定密碼檔案的名稱。

- `imq.keystore.property_name`。用於基於 SSL 的服務：指定與 SSL 密鑰儲存相關的安全性特性。請參閱第 297 頁表 14-9。

如需這些特性的完整說明，請參閱第 293 頁表 14-8。

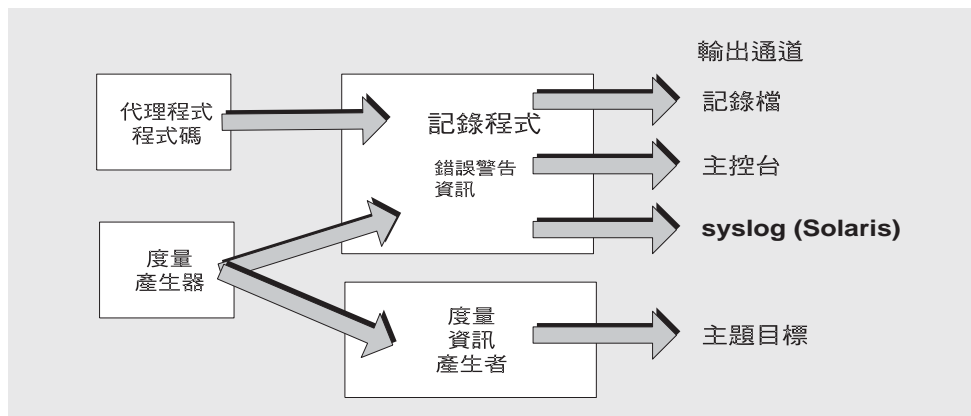
監視服務

代理程式包括一些用於監視和診斷其作業的元件。這些元件如下：

- 產生資料的元件（記錄事件與度量產生器的代理程式程式碼）
- 記錄程式元件（請參閱「記錄程式」），可透過一些輸出通道寫出資訊
- 訊息產生器，可將包含度量資訊的 JMS 訊息傳送到 JMS 監視用戶端使用的主題目標。

圖 4-5 中說明了一般方案。

圖 4-5 監視服務支援



度量產生器

度量產生器提供代理程式活動的相關資訊，例如流入和流出代理程式的訊息流量、代理程式記憶體中的訊息數目和使用的記憶體、開啓連線的數目，以及使用的執行緒數目。

您可以開啓或關閉度量資料的產生，並指定產生度量報告的頻率。

記錄程式

Message Queue 記錄程式會取得代理程式程式碼產生的資訊和度量產生器，並將資訊寫入一些輸出通道：標準輸出 (主控台)、記錄檔及 Solaris™ 作業系統上的 syslog 常駐程式。

您可以指定記錄程式收集的資訊類型，以及寫入每個輸出通道的類型。

例如，您可以指定記錄程式級別，判定記錄程式收集的資訊類型：錯誤 (ERROR)、錯誤和警告 (WARNING)，或錯誤、警告和資訊 (INFO)。

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 INFO，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) 寫入至記錄檔。

如果您使用記錄檔，您可以指定何時關閉記錄檔並將輸出自動重建至新檔案。當建立新的自動重建記錄檔時，將會保留 9 個最新記錄檔的歸檔檔案。

如需配置記錄程式的相關資訊，請參閱第 189 頁「配置及使用代理程式記錄功能」。如需有關配置和使用 Solaris syslog 的資訊，請參閱 `syslog(1M)`、`syslog.conf(4)` 和 `syslog(3C)` 線上手冊。

度量資訊產生者 (Enterprise Edition)

訊息產生者元件會在固定時間間隔，從度量產生器元件收到資訊。然後將資訊寫入訊息，再將訊息傳送到一些度量主題目標。度量訊息的傳送目標，視訊息中包含的資訊類型而定。

總共有 5 個度量主題目標。表 4-3 中顯示了這些目標的名稱，和傳送到每個目標的度量訊息類型。

表 4-3 度量主題目標

主題目標名稱	度量訊息的類型
<code>mq.metrics.broker</code>	代理程式度量
<code>mq.metrics.jvm</code>	Java 虛擬機器度量
<code>mq.metrics.destination_list</code>	目標及其類型的清單
<code>mq.metrics.destination.queue. monitoredDestinationName</code>	用於指定名稱之佇列的目標度量
<code>mq.metrics.destination.topic. monitoredDestinationName</code>	用於指定名稱之主題的目標度量

訂閱這些度量主題目標的 `Message Queue` 用戶端，會使用目標中的訊息並處理度量資訊。例如，用戶端可以指定到 `mq.metrics.broker` 目標，以接收和處理像代理程式中訊息總數目的這類資訊。

度量訊息產生者為可建立訊息 (`MapMessage` 類型，包含與度量資料對應的名稱值對) 的內部 `Message Queue` 用戶端。只有在有一個或多個對應度量主題目標的用戶時，才會產生這些訊息。

度量訊息產生者產生的訊息類型為 `MapMessage`。它們是根據訊息包含的度量類型，由一些名稱 / 值對而組成。每個名稱 / 值對會對應到一個度量數目和度量數目的值。

代理程式度量訊息包含一些訊息數目值，例如，流入和流出代理程式的訊息數目、這些訊息的大小，以及目前記憶體中的訊息數目和大小等。如需每個度量訊息類型中報告度量數目詳細資訊，請參閱 `Message Queue Developer's Guide for Java Clients`。手冊中解釋如何寫入使用度量訊息的 `Message Queue` 用戶端。

除了包含於度量訊息主體的度量資訊外，每個訊息標頭還有提供下列資訊的特性：

- 訊息類型
- 傳送訊息代理程式的主機、連接埠和位址
- 度量範例的製作時間

這些特性對處理不同類型或不同代理程式的度量訊息的 `Message Queue` 用戶端應用程式十分有用。

監視服務特性

這些特性是可配置特性，代理程式可以使用他們設定資訊的產生、記錄和度量訊息產生等資訊。

- `imq.metrics.enabled`。指定是否要將度量資訊寫入記錄程式。
- `imq.metrics.interval`。如果已啓用度量記錄，則請指定將度量資訊寫入記錄程式的時間間隔 (以秒為單位)。
- `imq.log.level`。指定記錄程式級別：可寫入至輸出通道的輸出種類。
- `imq.log.file.output`。指定寫入至記錄檔的記錄資訊的種類。
- `imq.log.file.dirpath`。指定包含記錄檔之目錄的路徑。
- `imq.log.file.filename`。指定記錄檔的名稱。
- `imq.log.file.rolloverbytes`。指定記錄檔輸出自動重建至新記錄檔時，此記錄檔的容量 (以位元組為單位)。

- `imq.log.file.rolloversecs`。指定記錄檔輸出自動重建至新記錄檔時，此記錄檔的存在時間 (以秒為單位)。
- `imq.log.console.output`。指定寫入至主控台的記錄資訊的種類。
- `imq.log.console.stream`。指定主控台輸出寫入至 `stdout` (OUT)，或寫入至 `stderr` (ERR)。
- `imq.log.syslog.facility`。(僅適用於 Solaris) 指定 Message Queue 代理程式應記錄為的 `syslog` 設備。
- `imq.log.syslog.logpid`。(僅適用於 Solaris) 指定是否同時記錄訊息與代理程式程序 ID。
- `imq.log.syslog.logconsole`。(僅適用於 Solaris) 指定如果訊息無法傳送至 `syslog`，是否將它們寫入至系統主控台。
- `imq.log.syslog.identity`。(僅適用於 Solaris) 指定應前置於每個訊息的身份字串是否記錄至 `syslog`。
- `imq.log.syslog.output`。(僅適用於 Solaris) 指定寫入至 `syslogd(1M)` 的記錄資訊的種類。
- `imq.log.timezone`。指定記錄時間戳記的時區。
- `imq.metrics.topic.enabled`。指定是否啟用度量訊息產生功能。
- `imq.metrics.topic.interval`。指定產生度量訊息的時間間隔 (以秒為單位)。
- `imq.metrics.topic.persist`。指定度量訊息是否為永久性訊息。
- `imq.metrics.topic.timetolive`。指定傳送到度量主題目標之度量訊息的有效期 (以秒為單位)。
- `imq.destination.logDeadMsgs`。指定每次代理程式捨棄停用的訊息或是將停用的訊息放入停用訊息佇列中時，是否在記錄檔中寫入訊息。

如需這些特性的完整參照資訊，請參閱第 297 頁表 14-10。

關於配置檔案

代理程式配置檔案可用來配置代理程式。附錄 A 「Message Queue 資料的作業系統特定位置」列出作業系統的這些檔案所在的目錄。

此目錄儲存以下檔案：

- 啟動時載入的預設配置檔案。此檔案名稱為 `default.properties`，並且不可編輯。您可以閱讀此檔案，以確定預設值並找到您要變更之特性的確切名稱。

- 包含安裝 Message Queue 時指定的所有特性的**安裝配置檔案**。此檔案名稱爲 `install.properties`；安裝之後無法編輯。

實例配置檔案

您首次執行代理程式時，會建立實例配置檔案。您可以使用實例配置檔案來指定代理程式實例的配置特性。

實例配置檔案儲存於以代理程式實例名稱 (*instanceName*) 作為辨別依據的目錄中，且代理程式實例與配置檔案相互關聯：

```
.../instances/instanceName/props/config.properties
```

有關 `instances` 目錄的位置，請參閱附錄 A 「[Message Queue 資料的作業系統特定位置](#)」。

備註 `.../instances/instanceName` 目錄和實例配置檔案，歸其對應的代理程式實例的建立者所有。代理程式實例必須始終由同一個使用者重新啓動。

代理程式實例會維護實例配置檔案。當您使用管理工具變更配置時，即會修改實例配置檔案。您也可以以手動方式編輯實例配置檔案來變更配置（請參閱第 92 頁「[編輯實例配置檔案](#)」）。若要執行此動作，您必須爲 `.../instances/instanceName` 目錄的擁有者，或以超級使用者身份登入，才可變更目錄上的權限。

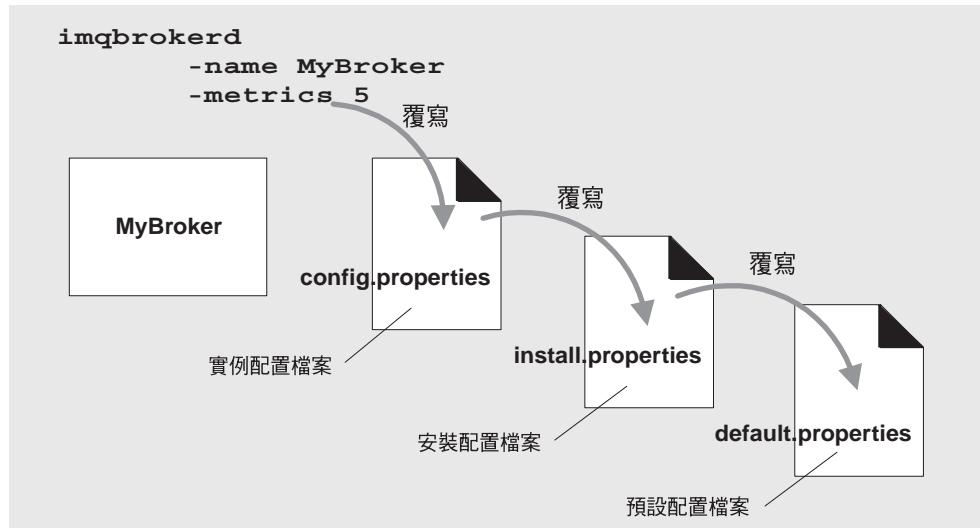
如果您連線叢集中的代理程式實例，您可能還需要使用 [叢集配置檔案](#) 以指定叢集配置資訊。如需更多資訊，請參閱第 300 頁「[叢集配置特性](#)」。

合併特性值

啓動時，代理程式會合併不同配置檔案中的特性值。它使用在安裝和實例配置檔案中的值，覆寫在預設配置檔案中指定的值。

您可以透過使用 `imqbrokerd` 指令選項覆寫結果值。此方案在圖 4-6 中有說明。

圖 4-6 代理程式配置檔案



特性命名語法

配置檔案中的任何 Message Queue 特性定義均使用以下命名語法：

```
propertyName=value [, value1] ...]
```

例如，以下項目指定拒絕其他訊息前，代理程式會在記憶體和永久性儲存中攔截 50,000 個訊息。

```
imq.system.max_count=50000
```

下列項目指定每天 (86400 秒) 會建立一個新的記錄檔：

```
imq.log.file.rolloversecs=86400
```

第 281 頁第 14 章「代理程式特性參照」列出代理程式配置特性及其預設值。

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，您可以執行以下作業之一：

- 使用管理工具。如需有關可使用 `imqcmd` 設定之特性的資訊，請參閱第 282 頁表 14-1。
- 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 作業系統上，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 14-1 以字母順序列出代理程式實例配置特性及其預設值。如需有關每種特性的涵義和使用的更多資訊，請參閱指定的交叉參照部分。

在指令行上輸入配置選項

您可以在啟動代理程式時或啟動之後，在指令行上輸入代理程式配置選項。

啟動時，您可以使用 `imqbrokerd` 指令來啟動代理程式實例。指令的 `-D` 選項，可用來指定任何代理程式特性及其值。如果您使用 `imqsvcadm` 指令將代理程式啟動為 Windows 服務，則 `-args` 選項可用來指定啟動配置特性。

執行代理程式實例時，您也可以設定一些代理程式特性。若要修改執行中代理程式的配置，您可以使用 `imqcmd update bkr` 指令。

如需啟動配置的詳細資訊，請參閱第 3 章「啟動代理程式和用戶端」，尤其是第 64 頁「以互動方式啟動代理程式」中的範例。

如需修改執行中代理程式的配置的詳細資訊，請參閱第 5 章「管理代理程式」和第 14 章「代理程式特性參照」。

設定永久性儲存

Message Queue 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。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，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。

Message Queue 資料儲存包含作業事件、訊息、長期訂閱和實體目標的相關資訊。此外還包含確認的訊息狀態資訊。

本章解釋如何設定代理程式使用永久性儲存。它包括以下主題：

- 第 94 頁「配置檔案系統儲存」

- [第 94 頁「配置 JDBC 儲存」](#)
- [第 98 頁「保護永久性資料」](#)

配置檔案系統儲存

您建立代理程式實例時，會自動建立檔案系統資料儲存。儲存在代理程式實例目錄中。這是作業系統特有的位置。有關永久性儲存的實際位置，請參閱[附錄 A「Message Queue 資料的作業系統特定位置」](#)。

依預設，Message Queue 會非同步寫入至磁碟。作業系統可以緩衝這些作業以提供更好的效能。但是，如果寫入作業期間發生意外的系統失敗，則訊息可能會遺失。若要改進可靠性，您可以讓 Message Queue 同步寫入磁碟，但請注意這個選項會降低效能。若要同步寫入磁碟，請設定代理程式特性 `imq.persist.file.sync`。如需這個特性的詳細資訊，請參閱[第 290 頁表 14-6](#)。

啓動代理程式實例時，您可以使用 `imqbrokerd -reset` 選項來清除檔案系統儲存。如需這個選項及其子選項的詳細資訊，請參閱[第 258 頁表 13-2](#)。

配置 JDBC 儲存

若要配置代理程式使用基於 JDBC 的持續性，請在代理程式配置檔案中設定 JDBC 的相關特性，並且建立適當的資料庫模式。Message Queue 資料庫管理員公用程式 (`imqdbmgr`) 使用 JDBC 驅動程式和代理程式配置特性來建立及管理資料庫。

例如，本章中描述的程序使用 Java 2 Platform, Enterprise Edition (J2EE) SDK 隨附的 PointBase DBMS。您可以從 `java.sun.com` 下載 1.4 版本。此範例使用 PointBase 的內嵌式版本（並非用戶端 / 伺服器版本）。在執行程序中，說明均使用 PointBase 範例中的路徑名稱和特性名稱來加以描述。它們均用「範例：」一詞識別。

您可以使用 Oracle 和 PointBase 配置範例。要尋找範例檔案，請參閱[附錄 A「Message Queue 資料的作業系統特定位置」](#)。請在列出作業系統資訊的表格中，尋找「範例應用程式與配置」。

此外，實例配置檔案 `config.properties` 中以注釋值的形式提供了 PointBase 內嵌版本、PointBase 伺服器版本和 Oracle 的範例。

插入 JDBC 可存取的资料儲存

插入 JDBC 可存取的资料儲存僅需要幾個步驟。

► 插入 JDBC 可存取的资料儲存

1. 在代理程式的配置檔案中設定與 JDBC 相關的特性。

請參閱第 291 頁「基於 JDBC 的持續性」中提供的特性。

2. 將您的 JDBC 驅動程式 JAR 檔案的副本或符號連結放在以下路徑中：

`/usr/share/lib/imq/ext/ (Solaris)`

`/opt/sun/mq/share/lib/ (Linux)`

`IMQ_VARHOME\lib\ext (Windows)`

副本範例 (Solaris)：

```
% cp j2eeSDK_install_directory/pointbase/lib/pointbase.jar
/usr/share/lib/imq/ext
```

符號連結範例 (Solaris)：

```
% ln -s j2eeSDK_install_directory/lib/pointbase/pointbase.jar
/usr/share/lib/imq/ext
```

3. 建立 Message Queue 持續性所需的資料庫模式。

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 98 頁「資料庫管理員公用程式 (`imqdbmgr`)」。

範例：

- a. 變更 `imqdbmgr` 常駐的目錄。

`cd /usr/bin (Solaris)`

`cd /opt/sun/mq/bin (Linux)`

`cd IMQ_HOME/bin (Windows)`

- b. 輸入 `imqdbmgr` 指令。

`imqdbmgr create all`

備註 若您使用內嵌式資料庫，最好建立在下列目錄中：

```
.../instances/instanceName/dbstore/databasename.
```

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` 指令建立內嵌式資料庫的使用者相同（請參閱第 98 頁「資料庫管理員公用程式 (`imqdbmgr`)」）。

與 JDBC 相關的代理程式特性

代理程式實例配置檔案儲存於以代理程式實例名稱作為辨別依據的目錄中，且代理程式實例與配置檔案相互關聯（請參閱附錄 A「[Message Queue 資料的作業系統特定位置](#)」）：

```
.../instances/instanceName/props/config.properties
```

如果此檔案不存在，那麼您必須使用 `-name instanceName` 選項啟動代理程式，讓 `Message Queue` 建立此檔案。

[第 291 頁「基於 JDBC 的持續性」](#) 描述了當您插入 JDBC 可存取的資料儲存時，需要設定的配置特性。本節的最後有這些特性的摘要。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`) 中設定這些特性。

這些實例配置特性可讓您自訂建立 `Message Queue` 資料庫綱要的 SQL 程式碼：有一個指定 SQL 程式碼的可配置特性，此 SQL 程式碼可建立資料庫表格。這些特性需要利用外掛資料庫來指定適當的資料類型。

因為與 SQL 語法相關的資料庫供應商之間會有不相容的情形，所以請確切檢查您資料庫供應商提供的相對文件，並隨之調整 [第 291 頁表 14-7](#) 中的特性。例如，使用 `PointBase` 資料庫時，您可能需要將 `IMQMSG35` 表格中的 `MSG` 欄寬調整到最大（請參閱 `imq.persist.jdbc.table.IMQMSG35` 特性）。

如同所有的代理程式配置特性，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特性，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：

在 `PointBase` 內嵌式資料庫範例中，您並不在資料庫連線 URL 中指定資料庫絕對路徑，而是使用 `-D` 指令行選項定義 `PointBase` 系統目錄：

```
-Ddatabase.home=IMQ_VARHOME/instances/instanceName/dbstore
```

在此情況下，您可以如下所示，指定 URL 來建立資料庫：

```
imq.persist.jdbc.createdburl=jdbc:pointbase:embedded:dbName;new
```

您可以如下所示，指定 URL 來開啓資料庫：

```
imq.persist.jdbc.opendburl=jdbc:pointbase:embedded:dbName
```

下面是 JDBC 相關特性的摘要：

- `imq.persist.store`。指定基於檔案或基於 JDBC 的資料儲存。
- `imq.persist.jdbc.brokerid`。指定附加至資料庫表格名稱的代理程式實例識別碼，以確保他們是唯一的。
- `imq.persist.jdbc.driver`。指定 JDBC 驅動程式的 Java 類別名稱，以連線至資料庫。
- `imq.persist.jdbc.opendburl`。指定資料庫 URL，以開啓與現有資料庫的連線。
- `imq.persist.jdbc.createdburl`。指定資料庫 URL，以開啓連線建立資料庫。
- `imq.persist.jdbc.closedburl`。指定資料庫 URL，以在關閉代理程式時關閉目前資料庫連線。
- `imq.persist.jdbc.user`。如有必要，則指定用於開啓資料庫連線的使用者名稱。
- `imq.persist.jdbc.needpassword`。指定資料庫是否需要密碼以存取代理程式。
- `imq.persist.jdbc.password`。如有必要，則指定用於開啓資料庫連線的密碼。
- `imq.persist.jdbc.table.IMQSV35`。用來建立版本表格的 SQL 指令。
- `imq.persist.jdbc.table.IMQCCREC35`。用來建立配置變更記錄表格的 SQL 指令。
- `imq.persist.jdbc.table.IMQDEST35`。用來建立目標表格的 SQL 指令。
- `imq.persist.jdbc.table.IMQINT35`。用來建立偏好表格的 SQL 指令。
- `imq.persist.jdbc.table.IMQMSG35`。用來建立訊息表格的 SQL 指令。
- `imq.persist.jdbc.table.IMQPROPS35`。用來建立特性表格的 SQL 指令。
- `imq.persist.jdbc.table.IMQILIST35`。用來建立偏好狀態表格的 SQL 指令。
- `imq.persist.jdbc.table.IMQTXN35`。用來建立作業事件表格的 SQL 指令。
- `imq.persist.jdbc.table.IMQTACK35`。用來建立作業事件確認表格的 SQL 指令。

如需這些特性的完整參照資訊，請參閱第 14 章「代理程式特性參照」。

資料庫管理員公用程式 (imqdbmgr)

Message Queue 提供資料庫管理員公用程式 (imqdbmgr)，以設定持續性所需的模式。如果 Message Queue 資料庫表格損毀或者您想使用其他資料庫作為資料儲存，您還可以使用此公用程式刪除這些表格。

如需 imqcmd 指令的語法、子指令和選項的詳細資訊，請參閱第 13 章「指令參照」。

保護永久性資料

永久性儲存可以包含暫時儲存的訊息檔案和其他資訊。由於這些訊息可能包含專用資訊，請務必保護資料儲存，以防止未經授權的存取。本節描述如何確保內建檔案儲存或 JDBC 儲存的資料安全。

內建 (基於檔案的) 永久性儲存

使用內建永久性的代理程式會將永久性資料寫入文字檔案資料儲存，該文字檔案資料儲存的位置是特定於作業系統的 (請參閱附錄 A「Message Queue 資料的作業系統特定位置」)：

```
.../instances/instanceName/fs350/
```

此處的 *instanceName* 是辨別代理程式實例的名稱。

instanceName/filestore/ 目錄在代理程式實例首次啟動時建立。保護此目錄的程序取決於在其上執行代理程式的作業系統。

Solaris 和 Linux `IMQ_VARHOME/instances/instanceName/filestore/` 目錄的許可權取決於啟動代理程式實例的使用者的 `umask`。因此，透過適當地設定 `umask`，您可以限制啟動代理程式實例和讀取其永久性檔案的許可權。此外，管理員 (超級使用者) 可以透過將 `IMQ_VARHOME/instances` 目錄的許可權設定為 700 來保護永久性資料。

Windows `IMQ_VARHOME/instances/instanceName/filestore/` 目錄的許可權可以使用您所使用的 Windows 作業系統提供的機制來設定。這通常需要開啓目錄的內容對話方塊。

外掛 (JDBC) 永久性儲存

使用外掛永久性儲存的代理程式將永久性資料寫入 JDBC 相容的資料庫中。

對於由資料庫伺服器管理的資料庫 (例如 Oracle 資料庫)，建議您建立使用者名稱和密碼，以存取 Message Queue 資料庫表格 (表格名稱以「IMQ」開頭)。如果資料庫不允許保護個別表格，請建立僅由 Message Queue 代理程式使用的專用資料庫。請諮詢資料庫供應商，以取得有關如何建立使用者名稱 / 密碼存取的文件。

代理程式開啓資料庫連線所需的使用者名稱和密碼可以作為代理程式配置特性提供。但是，在啓動代理程式時，將它們作為指令行選項提供會更加安全 (請參閱 Message Queue 管理指南的附錄 A 「設定外掛永久性儲存」)。

對於代理程式透過資料庫 JDBC™ 驅動程式直接存取的內嵌式資料庫，通常是以設定儲存永久性資料的目錄上檔案許可權的方式來確保此資料庫的安全，「[內建 \(基於檔案的\) 永久性儲存](#)」中有詳細的描述。但是為確保代理程式和 imqdbmgr 公用程式都能讀取和寫入資料庫，二者應該由同一位使用者執行。

保護永久性資料

管理代理程式

本章介紹如何執行與管理代理程式及其服務相關的基本工作。本章有下列幾節：

- 第 102 頁 「必要條件」
- 第 102 頁 「使用 `imqcmd` 指令公用程式」
- 第 104 頁 「顯示說明」
- 第 104 頁 「顯示產品版本」
- 第 104 頁 「顯示代理程式資訊」
- 第 105 頁 「更新代理程式特性」
- 第 106 頁 「暫停與繼續代理程式」
- 第 107 頁 「關閉和重新啓動代理程式」
- 第 108 頁 「顯示代理程式度量」
- 第 109 頁 「管理連線服務」
- 第 113 頁 「取得有關連線的資訊」
- 第 115 頁 「管理長期訂閱」
- 第 116 頁 「管理作業事件」

本章未包含有關管理代理程式的所有相關主題。下列幾章包含其他大型主題：

- 在代理程式上管理實體目標。如需有關如何建立、顯示、更新及銷毀實體目標和如何使用停用的訊息佇列等主題的詳細資訊，請參閱第 6 章 「管理實體目標」
- 設定代理程式的安全性。如需有關使用者認證、存取控制、加密、密碼檔案和稽核記錄等主題的詳細資訊，請參閱第 7 章 「管理安全性」

必要條件

您可以使用 `imqcmd` 和 `imqusermgr` 指令來管理代理程式。管理代理程式前，請先執行以下作業：

- 使用 `imqbrokerd` 指令啟動代理程式。代理程式執行後，您才能使用其他指令。
- 決定要設定 Message Queue 管理使用者或使用預設帳號。您必須指定使用者名稱和密碼，才能使用管理指令。

當您安裝 Message Queue 時，同時也安裝了預設文字檔案使用者儲存庫。此儲存庫出廠時隨附兩個預設項目：管理使用者和客人使用者。測試 Message Queue 時，您可以使用預設使用者名稱和密碼 (`admin/admin`) 來執行 `imqcmd` 公用程式。

設定生產系統時，必須設定認證及授權管理使用者。如需設定基於檔案使用者儲存庫或配置使用 LDAP 目錄伺服器的詳細資訊，請參閱第 7 章「管理安全性」。在生產環境中使用非預設使用者名稱和密碼，較為安全。

- 如果您要與代理程式使用安全連線，請在目標代理程式實例上設定並啟用 `ssladmin` 服務。如需更多資訊，請參閱第 146 頁「使用基於 SSL 的服務」。

使用 `imqcmd` 指令公用程式

`imqcmd` 指令公用程式能讓您管理代理程式及其服務。

如需 `imqcmd` 指令的語法、子指令和選項的詳細資訊，請參閱第 255 頁第 13 章「指令參照」。如需管理實體目標的詳細資訊，請參閱第 303 頁第 15 章「實體目標特性參照」。

指定使用者名稱和密碼

每個 `imqcmd` 指令都要經由使用者儲存庫進行認證，因此需要使用者名稱和密碼。但是下列情形例外：

- 使用 `-h` 或 `-H` 選項的指令顯示說明指令。
- 使用 `-v` 選項的指令顯示產品版本。

指定使用者名稱

使用 `-u` 選項指定管理使用者名稱。如果省略使用者名稱，指令會提示您輸入。

為了方便您閱讀，本章範例中 `-u` 選項的引數都顯示預設使用者名稱 `admin`。在生產環境中應該使用自訂使用者名稱。

指定密碼

您可以使用下列任一方法指定密碼：

- 建立密碼檔案 (passfile)，將密碼輸入檔案中。在指令行中使用 `-passfile` 選項，提供密碼檔案名稱。
- 讓指令提示您輸入密碼。只要別人看不到您輸入的內容，這就是指定密碼最安全的方法。

舊版 `Message Queue` 容許您在指令行上使用 `-p` 選項來指定密碼。這個選項已經被停用，將從未來版本中移除。

指定代理程式名稱和連接埠

`imqcmd` 的預設代理程式是本地主機上執行的代理程式，預設連接埠是 7676。

對遠端主機和 / 或非預設連接埠上執行的代理程式下達指令時，必須使用 `-b` 選項來指定代理程式的主機和連接埠。

範例

本節中的範例說明如何使用 `imqcmd`。

第一個範例列出在使用連接埠 7676 的 `localhost` 上執行的代理程式之特性，因此不需要 `-b` 選項。指令使用預設管理使用者名稱 (`admin`) 並且省略密碼，指令會提示您輸入密碼。

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

下列範例列出在使用連接埠 1564 的 `myserver` 上執行的代理程式的特性。使用者名稱是 `aladdin`。這個指令需要更新使用者儲存庫，以便將使用者名稱 `aladdin` 指定至 `admin` 群組。

```
imqcmd query bkr -b myserver:1564 -u aladdin
```

下列範例列出在使用連接埠 7676 的 `localhost` 上執行的代理程式的特性。該指令的初始逾時設定為 20 秒，重試 (逾時後) 次數設定為 7。使用者密碼在呼叫指令時之目前目錄的 `myPassfile` 密碼檔案中。

```
imqcmd query bkr -u admin -passfile myPassfile -rtm 20 -rtr 7
```

如果您要與代理程式使用安全連線，可以在範例中使用 `-secure` 選項。`-secure` 選項會讓 `imqcmd` 使用已經配置並啟動的 `ssladmin` 服務。

顯示說明

若要顯示 `imqcmd` 指令公用程式的說明，請使用 `-h` 或 `-H` 選項，不可以使用子指令。您無法取得特定子指令的說明。

例如，下列指令顯示 `imqcmd` 的說明：

```
imqcmd -H
```

如果您輸入的指令行除了子指令和其他選項之外還包含 `-h` 或 `-H` 選項，指令公用程式只處理 `-h` 或 `-H` 選項。忽略指令行的所有其他項目。

顯示產品版本

若要顯示 Message Queue 產品版本，您可以使用 `-v` 選項。例如：

```
imqcmd -v
```

如果您輸入的指令行除了子指令和其他選項之外還包含 `-v` 選項，指令公用程式只處理 `-v` 選項。忽略指令行的所有其他項目。

顯示代理程式資訊

若要查詢並顯示單一代理程式的相關資訊，請使用 `query bkr` 子指令。

`query bkr` 子指令的語法如下：

```
imqcmd query bkr -b hostName:port
```

這個子指令列出預設代理程式或指定主機和連接埠上的代理程式特性的目前設定。它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。

例如：

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

提示您輸入密碼後，指令會輸出類似下面的內容：

Version	3.6
實例名稱	imqbroker
主要連接埠	7676
系統中目前的訊息數目	0
系統中的目前訊息位元組	0
停用的訊息佇列中目前的訊息數目	0
停用的訊息佇列中目前訊息的總位元組	0
記錄停用的訊息	true
截斷停用的訊息佇列中的訊息內文	false
系統中的最大訊息數目	無限制 (-1)
系統中的訊息最大位元組	無限制 (-1)
最大訊息容量	70m
自動建立佇列	true
自動建立主題	true
自動建立最大使用中用戶數佇列	1
自動建立最大備份用戶數佇列	0
叢集代理程式清單 (使用中)	
叢集代理程式清單 (已配置)	
叢集主代理程式	
叢集 URL	
記錄層	
次	INFO
記錄自動重建的間隔 (以秒為單位)	604800
記錄自動重建的大小 (位元組)	無限制 (-1)

更新代理程式特性

您可以使用 `update bkr` 子指令更新下列代理程式特性：

- `imq.autocreate.queue`
- `imq.autocreate.topic`
- `imq.autocreate.queue.maxNumActiveConsumers`
- `imq.autocreate.queue.maxNumBackupConsumers`
- `imq.cluster.url`
- `imq.destination.DMQ.truncateBody`
- `imq.destination.logDeadMsgs`
- `imq.log.level`

- `imq.log.file.rolloversecs`
- `imq.log.file.rolloverbytes`
- `imq.system.max_count`
- `imq.system.max_size`
- `imq.message.max_size`
- `imq.portmapper.port`

`update bkr` 子指令的語法如下：

```
imqcmd update bkr [-b hostName:port] -o attribute=value [-o attribute=value1]...
```

這個子指令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。

第 14 章「代理程式特性參照」中描述了這些特性。

例如，以下指令會關閉佇列目標的自動建立功能：

```
imqcmd update bkr -o "imq.autocreate.queue=false" -u admin
```

暫停與繼續代理程式

啓動代理程式之後，您可以使用 `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的連線服務執行緒，這將導致代理程式停止偵聽連線連接埠。因此，代理程式將無法繼續接受新的連線、接收訊息和派送訊息。

但是，暫停代理程式不會暫停管理連線服務。此服務可讓您執行所需的管理工作，以管理代理程式的訊息流量。例如，如果某特定實體目標受到訊息的攻擊，則您可以暫停代理程式，並執行可能幫助您解決問題的動作，例如：

- 追蹤訊息來源
- 限制實體目標大小
- 銷毀實體目標

暫停代理程式也不會暫停 `cluster` 連線服務。但是，叢集中的訊息傳送會根據叢集中不同代理程式執行的傳送功能而有差異。

pause bkr 子指令的語法如下：

```
imqcmd pause bkr [-b hostName:port]
```

這個指令暫停預設代理程式或指定主機和連接埠上的代理程式。

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 -u admin
```

您也可以暫停個別連線服務和個別實體目標。如需更多資訊，請參閱第 112 頁「暫停與繼續連線服務」和第 125 頁「暫停和繼續實體目標」。

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。

resume bkr 子指令的語法如下：

```
imqcmd resume bkr [-b hostName:port]
```

這個子指令繼續預設代理程式或指定主機和連接埠上的代理程式。

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 -u admin
```

關閉和重新啟動代理程式

適當地關閉代理程式會終止代理程式程序。代理程式停止接受新的連線和訊息，它將完成現有訊息的傳送並終止代理程式程序。

shutdown bkr 子指令的語法如下：

```
imqcmd shutdown bkr [-b hostName:port]
```

這個子指令關閉預設代理程式或指定主機和連接埠上的代理程式。

以下指令關閉在使用連接埠 1572 的 ctrlsrv 上執行的代理程式：

```
imqcmd shutdown bkr -b ctrlsrv:1572 -u admin
```

您可以關閉代理程式後再重新啟動。restart bkr 子指令的語法如下：

```
imqcmd restart bkr [-b hostName:port]
```

這個子指令使用首次啟動代理程式時指定的選項，關閉預設代理程式或指定主機和連接埠上的代理程式後再重新啟動。若要選擇其他選項，請關閉代理程式後再重新啟動，指定您要的選項。

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：

```
imqcmd restart bkr -u admin
```

顯示代理程式度量

若要顯示有關代理程式的度量資訊，請使用 `metrics bkr` 子指令。

`metrics bkr` 子指令的語法如下：

```
imqcmd metrics bkr [-b hostName:port]
                  [-m metricType] [-int interval] [-msp numSamples]
```

這個子指令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度量。

您可以使用 `-m` 選項，指定下列一種要顯示的度量類型：

- **ttl** 顯示流入和流出代理程式的訊息和封包的度量 (預設度量類型)。
- **rts** 顯示 (每秒) 流入和流出代理程式的訊息和封包之流量速率的度量。
- **cxn** 顯示連線、虛擬記憶體堆疊和執行緒。

使用 `-int` 選項，指定顯示度量的間隔時間 (以秒為單位)。預設為 5 秒鐘。

使用 `-msp` 選項，指定輸出中顯示的範例數目。預設的數目為沒有限制 (無限)。

例如，若要取得每 10 秒時間間隔，代理程式流入與流出的訊息速率，請使用：

```
imqcmd metrics bkr -m rts -int 10 -u admin
```

此指令的輸出如下：

訊息 / 秒鐘		訊息位元組 / 秒鐘		Pkts / 秒鐘		Pkt 位元組 / 秒鐘	
輸入	輸出	輸入	輸出	輸入	輸出	輸入	輸出
0	0	27	56	0	0	38	66
10	0	7365	56	10	10	7457	1132
0	0	27	56	0	0	38	73
0	10	27	7402	10	20	1400	8459
0	0	27	56	0	0	38	73

如需使用 `imqcmd` 報告代理程式度量的更多詳細說明，請參閱第 324 頁「代理程式範圍度量」。

管理連線服務

指令公用程式包括允許您執行以下連線服務管理工作的子指令：

- 列出連線服務
- 顯示連線服務資訊
- 更新連線服務特性
- 顯示連線服務度量
- 暫停與繼續連線服務

代理程式支援來自應用程式用戶端和管理用戶端的連線。目前，Message Queue 代理程式可提供的連線服務如表 5-1 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，每種服務都與使用的服務類型（應用程式用戶端為 NORMAL，管理用戶端為 ADMIN）和基本傳輸協定相關。

表 5-1 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	tcp
ssljms (Enterprise Edition)	NORMAL	tls (基於 SSL 的安全性)
httpjms (Enterprise Edition)	NORMAL	http
httpsjms (Enterprise Edition)	NORMAL	https (基於 SSL 的安全性)
admin	ADMIN	tcp
ssladmin (Enterprise Edition)	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 `list svc` 子指令。

`list svc` 子指令的語法如下：

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

這個子指令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。

您可以在指令中使用子指令如下：

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

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

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

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 -u admin
```

此指令將輸出以下資訊：

服務名稱	連接埠號	服務狀態
admin	41844 (動態)	執行中
httpjms	-	未知
httpsjms	-	未知
jms	41843 (動態)	執行中
ssladmin	動態	未知
ssljms	動態	未知

顯示連線服務資訊

若要查詢並顯示單一服務的相關資訊，請使用查詢子指令。

query svc 子指令的語法如下：

```
imqcmd query svc -n serviceName [-b hostName:port]
```

這個子指令顯示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相關資訊。

例如：

```
imqcmd query svc -n jms -u admin
```


提示您輸入密碼後，指令會輸出類似下面的內容：

服務名稱	jms
服務狀態	執行中
連接埠號	60920 (動態)
目前配置的執行緒數目	0
目前的連線數	0
最小執行緒數目	10
最大執行緒數目	1000

更新連線服務特性

您可以使用 `update` 子指令變更表 5-2 中列出的一個或多個服務特性的值。

表 5-2 imqcmd 更新的連線服務特性

特性	說明
<code>port</code>	指定給要更新的服務之連接埠 (不套用於 <code>httpjms</code> 或 <code>httpsjms</code>)。0 值表示連接埠對映器會動態配置連接埠。
<code>minThreads</code>	指定給服務的執行緒之最小數目。
<code>maxThreads</code>	指定給服務的最大執行緒數目。

`update` 子指令的語法如下：

```
imqcmd update svc -n serviceName [-b hostName:port]
-o attribute=value [-o attribute=value1]...
```

這個子指令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。如需服務屬性的說明，請參閱第 285 頁「連線服務特性」。

以下指令將指定給 `jms` 服務之執行緒的最小數目變更至 20。

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

顯示連線服務度量

若要顯示有關單一服務的度量資訊，請使用 `metrics` 子指令。

metrics 子指令的語法如下：

```
imqcmd metrics svc -n serviceName [-b hostName:port] [-m metricType]
[-int interval] [-msp numSamples]
```

這個子指令顯示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度量。

使用 -m 選項指定要顯示的度量類型：

- **t11** 顯示經由指定連線服務，流入和流出代理程式的訊息和封包的度量。(預設度量類型)
- **rts** 顯示經由指定連線服務，(每秒)流入和流出代理程式的訊息和封包之流量速率的度量。
- **cxn** 顯示連線、虛擬記憶體堆疊和執行緒。

使用 -int 選項，指定顯示度量的間隔時間 (以秒為單位)。預設為 5 秒鐘。

使用 -msp 選項，指定輸出中顯示的範例數目。預設的數目為沒有限制 (無限)。

例如，若要取得由 jms 連線服務處理之訊息和封包的累計總數，請使用：

```
imqcmd metrics svc -n jms -m t11 -u admin
```

提示您輸入密碼後，指令會輸出類似下面的內容：

```
-----
  訊息          訊息位元組      Pkts          Pkt 位元組
  輸入 輸出    輸入 輸出    輸入 輸出    輸入 輸出
-----
 164  100 120704 73600  282  383 135967 102127
 657  100 483552 73600  775  876 498815 149948
-----
```

如需使用 imqcmd 報告連線服務度量的更多詳細說明，請參閱第 326 頁「[連線服務度量](#)」。

暫停與繼續連線服務

若要暫停管理服務 (無法暫停) 之外的任何服務，請使用 `pause svc` 和 `resume svc` 子指令。

`pause svc` 子指令的語法如下：

```
imqcmd pause svc -n serviceName [-b hostName:port]
```

這個子指令暫停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。您無法暫停管理服務。

您可以使用指令行如下：

```
imqcmd pause svc -n serviceName -u admin
```

暫停服務會造成以下影響：

- 代理程式會停止接受暫停服務上的新用戶端連線。如果 Message Queue 用戶端嘗試開啓新的連線，將會發生異常。
- 暫停服務上的所有現有連線會維持持續狀態，但在繼續服務前，代理程式會暫停此類連線上的所有訊息處理。(例如，如果用戶端嘗試傳送訊息，在繼續服務前，send() 方法會鎖定此訊息。)
- 將會維護任何已被代理程式接收之訊息的訊息傳送狀態。(例如，繼續服務時，不會中斷作業事件且將繼續訊息傳送。)

若要繼續服務，請使用 resume svc 子指令。

resume svc 子指令的語法如下：

```
imqcmd resume svc -n serviceName [-b hostName:port]
```

這個子指令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。

您可以使用指令行如下：

```
imqcmd resume svc -n serviceName -u admin
```

取得有關連線的資訊

指令公用程式包括允許您列出和取得連線相關資訊的子指令。

list cxn 子指令列出指定服務名稱的所有連線。list cxn 子指令的語法如下：

```
imqcmd list cxn [-svn serviceName] [-b hostName:port]
```

這個子指令列出預設代理程式上指定服務名稱的所有連線，或指定主機和連接埠上代理程式上的所有連線。如果未指定服務名稱，則會列出所有服務。

例如：

```
imqcmd list cxn -u admin
```

提示您輸入密碼後，指令會輸出類似下面的內容：

```

列出以下指定的代理程式上所有連線：
-----
主機                主要連接埠
-----
localhost           7676
-----
連線 ID            使用者  服務    產生者  用戶    主機
-----
1964412264455443200  guest  jms     0        1        127.0.0.1
1964412264493829311  admin  admin   1        1        127.0.0.1
-----
成功地列出連線。

```

若要查詢並顯示單一服務的相關資訊，請使用 `query` 子指令。

```
query cxn -n connectionID [-b hostName:port]
```

這個子指令顯示在預設代理程式或指定主機和連接埠上的代理程式上，指定服務的相關資訊。

例如：

```
imqcmd query cxn -n 421085509902214374 -u admin
```

提示您輸入密碼後，指令會輸出類似下面的內容：

```

連線 ID            421085509902214374
使用者            guest
服務              jms
產生者            0
用戶              1
主機              111.22.333.444
連接埠            60953
用戶端 ID
用戶端平台

```

管理長期訂閱

您可能需要使用 `imqcmd` 子指令管理代理程式的長期訂閱。*長期訂閱*是指由用戶端註冊為長期的主題訂閱；它具有唯一的身份，並要求代理程式保留此訂閱的訊息（即使它的用戶成為非使用中的用戶）。通常，代理程式僅在為長期用戶保留的訊息過期時，才刪除該訊息。

若要列出指定實體目標的長期訂閱，請使用 `list dur` 子指令。`list dur` 子指令的語法如下：

```
imqcmd list dur -d destName
```

例如，下列指令使用本地主機之預設連接埠上執行的代理程式，列出主題 `SPQuotes` 的所有長期訂閱：

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

對於主題的每個長期訂閱，`list dur` 子指令會傳回長期訂閱的名稱、使用者的用戶端 ID、在此主題上形成佇列的訊息數目以及長期訂閱的狀態（使用中 / 非使用中）。例如：

名稱	用戶端 ID	訊息數目	長期訂閱狀態
myDurable	myClientID	1	INACTIVE

您可以使用從 `list dur` 子指令傳回的資訊，識別您可能要銷毀或要為其清除訊息的長期訂閱。

`destroy dur` 子指令使用指定的用戶端識別碼，銷毀指定的長期訂閱。`destroy dur` 子指令的語法如下：

```
imqcmd destroy dur -n subscrName -c client_id
```

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
```

`purge dur` 子指令使用指定的用戶端識別碼，清除指定長期訂閱的所有訊息。`purge dur` 子指令的語法如下：

```
imqcmd purge dur -n subscrName -c client_id
```

管理作業事件

由用戶端應用程式啟動的所有作業事件可由代理程式追蹤。這些作業事件可以是簡單的 Message Queue 作業事件或由分散式作業事件 (XA 資源) 管理員管理的分散式作業事件。

每個作業事件都有 Message Queue 作業事件 ID，即在代理程式上唯一識別作業事件的 64 位元的號碼。分散式作業事件還有由分散式作業事件管理員指定的分散式作業事件 ID (XID)，此 ID 長達 128 個位元組。Message Queue 可維護 Message Queue 作業事件 ID 與 XID 的關聯。

如果發生故障，分散式作業事件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作業事件。

若要列出代理程式追蹤的所有作業事件，請使用 `list txn` 指令。`list tx` 子指令的語法如下：

```
imqcmd list txn
```

例如，以下指令列出代理程式中的所有作業事件。

```
imqcmd list txn
```

對於每個作業事件，`list` 子指令會傳回作業事件 ID、狀態、使用者名稱、訊息或確認的數目以及建立時間。例如：

作業事件 ID	狀態	使用者名稱	訊息數目 / 確認數目	建立時間
64248349708800	PREPARED	guest	4/0	1/30/02 10:08:31 AM
64248371287808	PREPARED	guest	0/4	1/30/02 10:09:55 AM

此指令顯示代理程式中的所有本機作業事件和分散式作業事件。您僅能確定或回轉處於 PREPARED 狀態的作業事件。如果您知道由於故障使作業事件處於此狀態，並且此作業事件未處於由分散式作業事件管理員確定的過程中，您應確定或回轉此作業事件。

例如，如果代理程式的自動回轉特性設定為 `false` (請參閱第 287 頁表 14-3)，那麼您必須在啟動代理程式時手動確定或回轉處於 PREPARED 狀態的作業事件。

`list` 子指令還顯示在作業事件中產生的訊息數目，以及在作業事件 (#Msgs/#Acks) 中確認的訊息數目。這些訊息不會被傳送，並且確認也不會被處理，直至作業事件被確定。

`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作業事件 ID (XID)。`query txn` 子指令的語法如下：

```
imqcmd query txn -n transaction_id
```

例如，下列範例的輸出內容應該如下：

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

指令的輸出內容如下：

```

用戶端 ID
連線      guest@192.18.116.219:62209->jms:62195
建立時間  1/30/02 10:08:31 AM
確認數目  0
訊息數目  4
狀態      PREPARED
作業事件 ID      64248349708800
使用者名稱      guest
XID
6469706F6C7369646577696E6465723130313234313431313030373230

```

`commit` 和 `rollback` 子指令可用於確定或回轉分散式作業事件。如前面所提及，僅可確定或回轉處於 `PREPARED` 狀態的作業事件。

`commit` 子指令的語法如下：

```
imqcmd commit txn -n transaction_id
```

例如：

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

還可配置代理程式，以在啟動代理程式時自動回轉處於 `PREPARED` 狀態的作業事件。

`rollback` 子指令的語法如下：

```
imqcmd rollback txn -n transaction_id
```

請參閱第 287 頁表 14-3 中的 `imq.transaction.autorollback` 特性，以獲得更多資訊。

管理實體目標

Message Queue 訊息可以經由代理程式上的實體目標，路由至其使用者用戶端。代理程式會管理實體目標的相關記憶體和永久性儲存體，並且設定其運作方式。

您只要在叢集中的一個代理程式上建立實體目標，叢集會將此實體目標傳遞至所有代理程式。應用程式用戶端可以訂閱或使用叢集中任何代理程式上的主題或佇列，因為代理程式會合作跨叢集路由訊息。但是唯有產生訊息的原始代理程式，可以管理訊息的持續性和確認。

本章說明如何執行下列工作：

- [第 120 頁](#) 「使用 `imqcmd` 指令公用程式」
- [第 121 頁](#) 「建立實體目標」
- [第 122 頁](#) 「列出實體目標」
- [第 123 頁](#) 「顯示實體目標資訊」
- [第 124 頁](#) 「更新實體目標特性」
- [第 125 頁](#) 「暫停和繼續實體目標」
- [第 125 頁](#) 「清除實體目標」
- [第 126 頁](#) 「銷毀實體目標」
- [第 127 頁](#) 「壓縮實體目標」
- [第 128 頁](#) 「配置使用停用的訊息佇列」

[表13-5](#) 提供有關使用 `imqcmd` 子指令來管理實體目標並且完成這些工作的完整參照資訊。

備註 用戶端應用程式與實體目標互動時，必定使用 Destination 物件。爲了供應程式獨立性和可移植性，用戶端通常使用管理員建立的目標物件，又稱目標受管理物件。您可以如第 8 章「管理受管理物件」中所描述，配置由用戶端應用程式使用受管理物件。

使用 imqcmd 指令公用程式

imqcmd 指令公用程式可讓您管理實體目標。imqcmd 指令的語法和您使用它來管理其他代理程式服務時相同。

如需有關 imqcmd、其子指令和選項的完整參照資訊，請參閱第 255 頁第 13 章「指令參照」。

子指令

表 6-1 列出本章中描述的 imqcmd 子指令。如需有關這些子指令的參照資訊，請參閱第 265 頁「實體目標管理子指令」。

表 6-1 imqcmd 指令公用程式的實體目標子指令

子指令與引數	說明
compact dst	壓縮用於一個或多個實體目標的內建且基於檔案的資料儲存。
create dst	建立實體目標。
destroy dst	銷毀實體目標。
list dst	列出代理程式上的實體目標。
metrics dst	顯示實體目標度量。
pause dst	暫停代理程式上的一個或多個實體目標。
purge dst	清除實體目標上的所有訊息而不銷毀實體目標。
query dst	查詢並顯示實體目標的相關資訊。
resume dst	繼續代理程式上一個或多個暫停的實體目標。
update dst	更新目標的特性。

建立實體目標

若要建立實體目標，您可以使用 `imqcmd create` 子指令。`create` 子指令的語法如下：

```
create dst -t destType -n destName [-o property=value] [-o property=value1]...
```

建立實體目標時，您必須指定以下項目：

- 實體目標類型 `t` (topic) 或 `q` (queue)。
- 實體目標名稱。命名規則如下：
 - 名稱只能使用字母數字字元。不能包含空格。
 - 名稱能以英文字母字元、底線字元 (`_`) 或錢幣符號 (`$`) 開頭。但不能以字元字串「`mq`」開頭。
- 實體目標特性的任何非預設值。

更新實體目標時，您也可以設定特性。

很多實體目標特性可以管理代理程式記憶體資源和訊息流量。例如，您可以指定能夠傳給實體目標的產生者數目、產生者可以傳送的訊息數量和容量，以及到達實體目標限制時代理程式的回應方式。這些限制類似代理程式配置特性控制的代理程式範圍限制。

下列特性可用於佇列目標和主題目標：

- `maxNumMsgs`。指定實體目標中允許給未使用訊息的最大數目。
- `maxTotalMsgBytes`。指定實體目標中允許給未使用訊息的最大總記憶體容量 (以位元組為單位)。
- `limitBehavior`。指定達到記憶體限制臨界值時，代理程式回應的方式。
- `maxBytesPerMsg`。指定實體目標中允許的任何單一訊息的最大容量 (以位元組為單位)。
- `maxNumProducers`。指定實體目標中產生者的最大數目。
- `consumerFlowLimit`。指定最大訊息數目，這些訊息將會傳送給單一批次中的某個用戶。
- `isLocalOnly`。僅套用於代理程式叢集。指定在其他代理程式上不重複實體目標，且限制目標僅傳送訊息給本地用戶 (連線到有建立實體目標代理程式的用戶)。
- `useDMQ`。指定捨棄實體目標的停用訊息或放到停用的訊息佇列中。

下列特性僅用於佇列目標：

- `maxNumActiveConsumers`。指定最大用戶數，此數值可作用於來自佇列目標的負載平衡傳送。)
- `maxNumBackupConsumers`。指定最大備份用戶數，如果無法從佇列目標進行負載平衡傳送，那麼這些用戶可以取代使用中的用戶。
- `localDeliveryPreferred`。僅套用於代理程式叢集中的負載平衡佇列傳送。如果本地代理程式沒有用戶，則指定訊息僅傳送給遠端用戶。

如需實體目標特性的完整參照資訊，請參閱第 303 頁第 15 章「實體目標特性參照」。

針對自動建立的目標，請在代理程式實例配置檔案中設定預設特性值。第 288 頁表 14-4 中有自動建立特性的參照資訊。

► 建立實體目標

- 若要建立佇列目標，請輸入以下指令：

```
mqcmd create dst -n myQueue -t q -o "maxNumActiveConsumers=5"
```

- 若要建立主題目標，請輸入以下指令：

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

列出實體目標

您可以取得關於實體目標目前特性值的相關資訊、關於與實體目標相關之產生者或用戶的數目以及關於訊息傳送度量的資訊，例如實體目標中訊息的數目和大小。

若要尋找您需要資訊的實體目標，請先列出代理程式的所有實體目標。您可以使用 `list dst` 子指令來列出。`dst` 子指令的語法如下：

```
list dst [-t destType] [-tmp]
```

這個指令可以列出所指定類型的實體目標。目標類型 (`-t`) 選項的值有 `q` (`queue`) 或 `t` (`topic`)。

如果省略目標類型，則會列出所有類型的實體目標。

`list dst` 子指令可選擇性指定要列出的目標類型，或選擇性納入暫存目標 (使用 `-tmp` 選項)。用戶端建立的暫存目標，通常用來接收傳送到其他用戶端的訊息回應。

例如，若要取得在連接埠 4545 的 myHost 上執行之代理程式的所有實體目標，請輸入以下指令：

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

除了其他實體目標之外，停用的訊息佇列 `mq.sys.dmq` 也一定會出現，除非您指定只包含主題的目標類型 `t`。

顯示實體目標資訊

若要取得實體目標目前的特性值的相關資訊，請使用 `query dst` 子指令。`query dst` 子指令的語法如下：

```
query dst -t destType -n destName
```

這個指令列出指定類型和名稱之目標的相關資訊。例如：

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

此指令的輸出如下：

目標名稱	目標類型
XQueue	佇列
在由以下所指定的代理程式上：	
主機	主要連接埠
localhost	7676
目標名稱	XQueue
目標類型	佇列
目標狀態	執行中
管理建立	true
目前訊息數目	0
目前訊息總位元組	0
目前產生者數目	0
目前使用中用戶數目	0
目前備份用戶數目	0
最大訊息數目	無限制 (-1)
訊息最大總位元組	無限制 (-1)
每條訊息的最大位元組	無限制 (-1)
最大產生者數目	100
最大使用中用戶數目	1
最大備份用戶數目	0

限制運作方式	REJECT_NEWEST
用戶流量限制	1000
是本機目標	false
偏好本機傳送	false
使用停用的訊息佇列	true

輸出也會顯示與目標相關之產生者與用戶的數目。佇列目標的這個數目包含使用中用戶和備份用戶。

您可以使用 `update dst` 子指令來變更一或多個特性值 (請參閱第 124 頁「更新實體目標特性」)。

更新實體目標特性

透過使用 `update dst` 子指令和 `-o` 選項指定要更新的特性，您可以變更實體目標的特性。`update dst` 子指令的語法如下：

```
update dst -t destType -n destName -o property=value [-o property=value1]...
```

這個指令會更新指定目標上的指定特性的值。特性名稱可能是表 15-1 中描述的任何特性。

您可以多次使用 `-o` 選項，更新多個特性。例如，以下指令將 `maxBytesPerMsg` 特性變更為 1000，並將 `MaxNumMsgs` 特性變更為 2000：

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

請參閱第 15 章「實體目標特性參照」，以獲得您可以更新的特性清單。

您無法使用 `update dst` 子指令更新實體目標的類型或更新 `isLocalOnly` 特性。

備註 停用的訊息佇列是一特殊的實體目標，它的特性與其他目標的特性不同。如需更多資訊，請參閱第 128 頁「配置使用停用的訊息佇列」。

暫停和繼續實體目標

您可以暫停實體目標，以便控制從產生者傳送訊息到目標，或從目標傳送訊息到用戶，或同時控制上述兩個動作。要特別注意的是，您可以暫停到目標的訊息流量，以避免目標在訊息產生速度高於使用速度時收到過多訊息。

若要暫停傳送到實體目標的訊息或暫停從實體目標傳送的訊息，請使用 `pause dst` 子指令。`pause dst` 子指令的語法如下：

```
pause dst [-t destType -n destName] [-pst pauseType]
```

針對指定類型和名稱的目標，這個子指令可以暫停至用戶的訊息傳送 (`-pst CONSUMERS`)，或暫停來自產生者的訊息傳送 (`-pst PRODUCERS`)，或者兩者皆暫停 (`-pst ALL`)。如果沒有指定目標類型與名稱，則會暫停所有實體目標。預設為 `ALL`。

範例：

```
imqcmd pause dst -n myQueue -t q -pst PRODUCERS -u admin
imqcmd pause dst -n myTopic -t t -pst CONSUMERS -u admin
```

若要繼續向暫停的目標傳送，請使用 `resume dst` 子指令。`resume dst` 子指令的語法如下：

```
resume dst [-t destType -n destName]
```

這個子指令繼續向指定類型和名稱之暫停目標傳送訊息。如果沒有指定目標類型與名稱，所有目標都會繼續。

範例：

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

在代理程式叢集中，實體目標實例會常駐在叢集中的每個代理程式。您必須個別暫停。

清除實體目標

您可以清除目前在實體目標上形成佇列的所有訊息。清除實體目標意味著刪除在實體目標上形成佇列的所有訊息。

當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。如果非使用中的主題長期用戶未成為使用中的用戶，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。

若要清除實體目標上的訊息，請使用 `purge dst` 子指令。`purge dst` 子指令的語法如下：

```
purge dst -t destType -n destName
```

這個子指令清除在指定類型與名稱的實體目標上的訊息。

範例：

```
imqcmd purge dst -n myQueue -t q -u admin
```

```
imqcmd purge dst -n myTopic -t t -u admin
```

如果您已關閉代理程式，並且不想在重新啓動它時傳送舊的訊息，請使用 `-reset messages` 選項清除舊的訊息，例如：

```
imqbrokerd -reset messages -u admin
```

這可避免重新啓動代理程式後清除目標的麻煩。

在代理程式叢集中，實體目標實例會常駐在叢集中的每個代理程式。您必須個別清除這些目標。

銷毀實體目標

若要銷毀實體目標，請使用 `destroy dst` 子指令。`destroy dst` 子指令的語法如下：

```
destroy dst -t destType -n destName
```

這個子指令會銷毀指定類型與名稱的實體目標。

範例：

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

銷毀實體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；此作業不可復原。

您不能銷毀停用的訊息佇列。

壓縮實體目標

如果您使用內建且基於檔案的資料儲存 (而非外掛式 JDBC 相容的資料儲存) 作為用於訊息的永久性儲存, 您可以監視磁碟使用和壓縮磁碟 (需要時)。

建立基於檔案的訊息儲存, 以便根據訊息於實體目標的位置, 將訊息儲存在目錄中。在各個實體目標的目錄中, 大部分的訊息會儲存在一個智慧型記錄 (即智慧型記錄檔案) 組成的檔案中。(若要減少分段程序, 大小超過可配置臨界值的訊息會儲存在其個別的檔案中。)

因為會存留不同大小的訊息, 接著從智慧型記錄檔案移除這些訊息, 所以檔案中可能會產生空白片段, 且將無法再重新使用這些可用記錄。

若要管理未使用的可用記錄, 使用率下降時, 指令公用程式包括每個實體目標監視磁碟使用及回收可用磁碟空間的子指令。

監視實體目標的磁碟使用

若要監視實體目標的磁碟使用, 請使用下列指令:

```
imqcmd metrics dst -t q -n myQueue -m dsk -u admin
```

此指令的輸出如下:

保留的	使用的	使用率
806400	804096	99
1793024	1793024	100
2544640	2518272	98

子指令輸出中的欄有以下意思:

表 6-2 實體目標磁碟使用度量

度量	說明
保留的	所有記錄使用的磁碟空間 (以位元組為單位), 包括保留使用中訊息和等待被重新使用之可用記錄的記錄
使用的	保留使用中訊息的記錄所使用的磁碟空間 (以位元組為單位)
使用率	使用的磁碟空間除以保留的磁碟空間所得之商數。比率越高表示有越多磁碟空間被用來保留使用中訊息。

回收未使用的實體目標磁碟空間

磁碟使用模式依使用特定實體目標之訊息傳送應用程式的特徵而異。根據流入和流出實體目標的相關訊息流量和相關的訊息大小，保留的磁碟空間可能會不斷增加。

如果訊息產生速率大於訊息使用速率，那麼通常會重新使用可用記錄，且使用率應較高。但是，如果訊息產生速率與訊息使用速率相似或小於訊息使用速率，那麼使用率會比較低。

一般而言，您會穩定保留的磁碟空間並保持高使用率。通常如果系統到達一個穩定狀態，即保留的磁碟空間大致固定且使用率高 (75% 以上)，那麼就沒有需要回收未使用的磁碟空間。如果系統到達穩定狀態且使用率低 (50% 以下)，您可以壓縮磁碟，回收可用記錄佔用的磁碟空間。

`compact dst` 子指令可用來壓縮資料儲存。`compact dst` 子指令的語法如下：

```
compact dst [-t destType -n destName]
```

這個子指令壓縮用於指定類型與名稱之實體目標的內建且基於檔案的資料儲存。如果沒有指定目標類型與名稱，所有目標都會壓縮。必須在壓縮目標前暫停實體目標。

如果保留的磁碟空間不斷增加，請藉由設定目標記憶體限制特性和限制運作方式，重新配置目標的記憶體管理 (請參閱第 303 頁表 15-1)。

► 回收未使用的實體目標磁碟空間

1. 暫停目標。

```
mqcmd pause dst -t q -n myQueue -u admin
```

2. 壓縮磁碟。

```
mqcmd compact dst -t q -n myQueue -u admin
```

3. 繼續實體目標。

```
mqcmd resume dst -t q -n myQueue -u admin
```

如果未指定目標類型和名稱，則會在所有實體目標上執行這些作業。

配置使用停用的訊息佇列

停用的訊息佇列，`mq.sys.dmq` 是系統建立的實體目標，保留代理程式的停用訊息和它的其他實體目標。停用的訊息佇列是用來監視、調校系統效率以及疑難排解的工具。如需「停用的訊息」一詞的定義和停用的訊息佇列的詳細資訊，請參閱 Message Queue 技術概述。

代理程式啟動時會自動建立停用的訊息佇列。代理程式將無法處理或到期的訊息放在此佇列中。此外，其他實體目標可以使用停用的訊息佇列來保留已捨棄的訊息。使用停用的訊息佇列而提供的資訊，有助於系統的疑難排解。

配置使用停用的訊息佇列

依預設，系統會配置實體目標使用停用的訊息佇列。您可以設定實體目標特性 `useDMQ`，讓實體目標使用或不使用停用的訊息佇列。

下列範例建立的佇列 `myDist`，依預設使用停用的訊息佇列：

```
imqcmd create dst -n myDist -t q
```

下列範例讓相同的佇列停用此停用的訊息佇列：

```
imqcmd update dst -n myDist -t q -o useDMQ=false
```

您可以設定 `imq.autocreate.destination.useDMQ` 代理程式特性，讓代理程式上所有自動建立的實體目標使用或不使用停用的訊息佇列。

配置及管理停用的訊息佇列

`imqcmd` 指令公用程式可以管理停用的訊息佇列。管理停用的訊息佇列的方式與管理其他佇列的方式，稍有不同。例如，停用的訊息佇列是系統建立的，因此您不能建立、暫停或銷毀。

停用的訊息佇列特性

您可以使用配置其他佇列的方式來配置停用的訊息佇列，但是有些實體目標特性可能不套用或是預設值不同。[表 6-3](#) 列出停用的訊息佇列另以特殊方式處理的佇列特性。

表 6-3 停用的訊息佇列處理標準實體目標特性的方式

特性	停用的訊息佇列的處理方式
<code>limitBehavior</code>	停用的訊息佇列的預設值是 <code>REMOVE_OLDEST</code> 。其他佇列的預設值是 <code>REJECT_NEWEST</code> 。停用的訊息佇列不支援流量控制。
<code>localDeliveryPreferred</code>	不套用於停用的訊息佇列。
<code>maxNumMsgs</code>	停用的訊息佇列的預設值是 1000。其他佇列的預設值是 -1 (無限制)。
<code>maxNumProducers</code>	不套用於停用的訊息佇列。

表 6-3 停用的訊息佇列處理標準實體目標特性的方式 (續)

特性	停用的訊息佇列的處理方式
maxTotalMsgBytes	停用的訊息佇列的預設值是 10 MB。其他佇列的預設值是 -1 (無限制)。
isLocalOnly	代理程式叢集中的停用的訊息佇列一定是本機實體目標，其特性一定設定為 true 。但是，本機代理程式的停用的訊息佇列可以包含叢集中其他代理程式的用戶端產生的訊息，前提是如果本機代理程式將訊息標示為停用。

訊息內容

代理程式可以將整個訊息放入停用的訊息佇列中，或是捨棄訊息內文內容，只保留標頭和特性資料。依預設，停用的訊息佇列儲存整個訊息。

如果您想減少佇列大小或不打算復原停用的訊息，可以考慮捨棄內文內容。

若要捨棄內文內容，只保留標頭和特性資料，請如下列範例所示，將 `imq.destination.DMQ.truncateBody` 代理程式特性設定為 `true`：

```
imqcmd update bkr -o imq.destination.DMQ.truncateBody=true
```

啟用停用訊息記錄功能

除了標準的佇列監視和記錄選項，您還可以記錄代理程式分類為停用的訊息。

如果啟用停用訊息記錄功能，代理程式會記錄下列類型的事件：

- 代理程式將訊息移動到停用的訊息佇列。
- 代理程式捨棄停用的訊息佇列中的訊息，或捨棄實體目標 (不使用停用的訊息佇列) 中的訊息。
- 實體目標已達上限。

依預設，不記錄停用的訊息。下列範例記錄停用的訊息：

```
imqcmd update bkr -o imq.destination.logDeadMsgs=true
```

停用訊息記錄功能套用於所有使用停用的訊息佇列的實體目標。您不能啟用或停用個別實體目標的記錄功能。

管理安全性

管理員負責配置用於認證使用者的使用者儲存庫、定義存取控制、配置用於加密用戶端與代理程式間之通訊的安全套接層 (SSL) 連線服務，以及設定用於啟動代理程式的密碼檔案。

本章包含以下各節：

- [第 132 頁](#) 「認證使用者」
- [第 140 頁](#) 「授權使用者：存取控制特性檔案」
- [第 146 頁](#) 「使用基於 SSL 的服務」
- [第 155 頁](#) 「使用密碼檔案」
- [第 157 頁](#) 「建立稽核記錄」

認證使用者

您負責維護使用者儲存庫中的使用者、使用者群組以及使用者密碼清單。您可以在每個代理程式實例使用不同的使用者儲存庫。本節介紹如何建立、寫入和管理儲存庫。

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。如果名稱和密碼與配置每個代理程式要參考的特定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。

儲存庫可能是下列任一類型：

- **Message Queue 隨附的文字檔案儲存庫**

這種類型的使用者儲存庫在使用上非常容易。您可以使用使用者管理員公用程式 (imqusermgr) 寫入和管理此儲存庫。若要啟用認證，您應寫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「[使用文字檔案使用者儲存庫](#)」。

- **LDAP 伺服器**

LDAP 目錄伺服器可以是現有或新的 LDAP 目錄伺服器，它將使用 LDAP v2 或 v3 協定。它不如文字檔案儲存庫易於使用，但是它可延伸，因此更適用於生產環境。

如果您要使用 LDAP 使用者儲存庫，您可以使用 LDAP 供應商提供的工具來寫入和管理此使用者儲存庫。如需更多資訊，請參閱第 137 頁「[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

Message Queue 提供了文字檔案使用者儲存庫和命令行工具，即 Message Queue 使用者管理員 (imqusermgr)，您可以使用此工具寫入和管理文字檔案使用者儲存庫。以下各節描述文字檔案使用者儲存庫，以及您如何使用 Message Queue 使用者管理員公用程式 (imqusermgr) 寫入和管理儲存庫。

建立使用者儲存庫

文字檔案使用者儲存庫適用於特定實例。建立的預設使用者儲存庫 (命名為 passwd)，用於您啟動的每個代理程式實例。此使用者儲存庫位於以代理程式實例名稱作為辨別依據的目錄中，且代理程式實例與儲存庫相互關聯 (請參閱附錄 A「[Message Queue 資料的作業系統特定位置](#)」)：

```
.../instances/instanceName/etc/passwd
```

此儲存庫隨附兩個項目(列)，如以下表 7-1 中所述。

表 7-1 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	作用中
guest	guest	anonymous	使用中

這些初始項目可讓 Message Queue 代理程式在安裝之後立即使用，而無須管理員的介入。Message Queue 代理程式無須初始使用者 / 密碼設定。

初始 guest 使用者項目可讓用戶端使用預設的 guest 使用者名稱和密碼連線到代理程式實例(例如用於測試)。

初始 admin 使用者項目可讓您使用 imqcmd 指令和預設的 admin 使用者名稱和密碼來管理代理程式實例。您應該更新此初始項目來變更密碼(請參閱第 137 頁「變更預設管理員密碼」)。

以下各節介紹如何寫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 (imqusermgr) 可讓您編輯或寫入文字檔案使用者儲存庫。本節介紹使用者管理員公用程式。後續各節介紹如何使用 imqusermgr 子指令完成特定的工作。

如需 imqusermgr 指令的完整參照資訊，請參閱第 13 章「指令參照」。

使用 imqusermgr 前，請注意以下事項：

- 如果代理程式特定的使用者儲存庫不存在，您必須啟動對應的代理程式實例來建立儲存庫。
- 必須在代理程式安裝的主機上執行 imqusermgr 指令。
- 您必須擁有寫入儲存庫的適當許可權：也就是說，在 Solaris 和 Linux 上，您必須為超級使用者或為第一個建立代理程式實例的使用者。

備註 以下各節中的範例均假設為預設的代理程式實例。

子指令

imqusermgr 指令有子指令 add、delete、list 和 update。

add 子指令 add 子指令可以將使用者和相關密碼新增到指定的 (或預設的) 代理程式實例儲存庫，並選擇性地指定使用者的群組。子指令語法如下：

```
add [-i instanceName] -u userName -p passwd [-g group] [-s]
```

delete 子指令 delete 子指令可以從指定的 (或預設的) 代理程式實例儲存庫刪除指定的使用者。子指令語法如下：

```
delete [-i instanceName] -u userName [-s] [-f]
```

list 子指令 list 子指令可以顯示有關指定的 (或預設的) 代理程式實例儲存庫中，指定的使用者或所有使用者的資訊。子指令語法如下：

```
list [-i instanceName] [-u userName]
```

update 子指令 update 子指令可以更新指定的 (或預設的) 代理程式實例儲存庫中，指定之使用者的密碼和 / 或狀態。子指令語法如下：

```
update [-i instanceName] -u userName -p passwd [-a state] [-s] [-f]
```

```
update [-i instanceName] -u userName -a state [-p passwd] [-s] [-f]
```

指令選項

表 7-2 列出了 imqusermgr 指令的選項。

表 7-2 imqusermgr 選項

選項	說明
-a <i>active_state</i>	指定使用者的狀態是否應為使用中 (true/false)。值 true 表示狀態為使用中。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。
-i <i>instanceName</i>	指定代理程式實例使用者儲存庫套用的指令。如果未指定，則假設為預設實例名稱值 imqbroker。
-p <i>passwd</i>	指定使用者密碼。
-g <i>group</i>	指定使用者群組。有效值為 admin、user 和 anonymous。
-s	設定無訊息模式。
-u <i>userName</i>	指定使用者名稱。

表 7-2 imqusermgr 選項 (續)

選項	說明
-v	顯示版本資訊。不執行指令行上的其他任何指令。

群組

為代理程式實例的使用者儲存庫新增使用者項目時，您可以指定三個預先定義群組中的任一項：`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。您可以將多個使用者指定給 `admin` 群組。
- **user 群組** 用於一般 (非管理) Message Queue 用戶端使用者。大多數用戶端使用者屬於 `user` 群組。依預設，此群組中的使用者可以產生訊息到所有主題與佇列、使用來自所有主題與佇列的訊息，以及瀏覽任何佇列中的訊息。
- **anonymous 群組**。用於 Message Queue 用戶端，這些用戶端不想使用代理程式知道的使用者名稱 (可能因為用戶端應用程式不知道要使用的實際使用者名稱)。這帳號對於大多數 FTP 伺服器中存在的匿名帳號均類似。您一次僅可將一個使用者指定給 `anonymous` 群組。您應當限制此群組的存取權限 (與 `user` 群組相比較)，或者在部署時移除此群組中的使用者。

若要變更使用者的群組，您必須刪除使用者項目並為此使用者新增其他項目，從而指定新的群組。

您不能將系統建立的群組重新命名或刪除，也不能建立新群組。但是，您可以指定用來定義群組成員可執行作業的存取規則。如需更多資訊，請參閱第 140 頁「[授權使用者：存取控制特性檔案](#)」。

註冊者狀態

依預設，新增使用者至儲存庫時，使用者的狀態為使用中。若要讓此使用者變成非使用中狀態，您必須使用更新指令。例如，以下指令可讓使用者 `JoeD` 的狀態變成非使用中：

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

已變成非使用中使用者的項目保留在儲存庫中；但是，非使用中使用者無法開啓新的連線。如果使用者為非使用中，且您新增具有相同名稱的其他使用者，則作業將失敗。您必須刪除非作用中使用者項目或變更新使用者的名稱，或者為新使用者命名其他名稱。這可防止您新增重複的使用者名稱。

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：

- 使用者名稱中不能包含星號 (*)、逗號 (,)、冒號 (:)、新行或換行。
- 使用者名稱或密碼必須至少包含一個字元。
- 如果使用者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。
- 密碼或使用者名稱的長度沒有限制，除非指令 `shell` 對指令行上可輸入的最大字元數強加限制。

加入和管理使用者儲存庫

使用 `add` 子指令可新增使用者至儲存庫。例如，以下指令可將使用者 `Katharine` 和密碼 `sesame`，新增到預設的代理程式實例使用者儲存庫。

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

使用 `delete` 子指令可從儲存庫刪除使用者。例如，以下指令可刪除使用者 `Bob`：

```
imqusermgr 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 `aladdin`：

```
imqusermgr update -u Katharine -p aladdin
```

若要列出有關一個或所有使用者的資訊，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
```

```
% imqusermgr list -u isa
用於代理實例的使用者儲存庫：imqbroker
-----
使用者名稱      群組      使用中狀態
-----
isa              admin     true
```

以下指令列出有關所有使用者的資訊：

```
imqusermgr list
```

```
% imqusermgr list

用於代理實例的使用者儲存庫：imqbroker
-----
使用者名稱    群組          使用中狀態
-----
admin         admin         true
guest        anonymous     true
isa          admin         true
testuser1    user          true
testuser2    user          true
testuser3    user          true
testuser4    user          false
testuser5    user          false
```

變更預設管理員密碼

為了安全性，您應將 `admin` 的預設密碼變更為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 `mybroker` 代理程式實例的預設管理員密碼 `admin` 變更為 `grandpoobah`。

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

當代理程式實例正在執行時，您可以透過執行任一指令行工具，來快速確認此變更是否生效。例如，下列指令會提示您提供密碼：

```
imqcmd list svc mybroker -u admin
```

輸入新密碼 (`grandpoobah`) 即生效，輸入舊密碼即失敗。

變更密碼後，使用任一 `Message Queue` 管理工具（包括管理主控台）時，您均應提供新的密碼。

將 LDAP 伺服器註冊於使用者儲存庫

若要將 LDAP 伺服器用於使用者儲存庫，請執行下列工作：

- 編輯實例配置檔案

- 設定管理員的存取控制

編輯實例配置檔案

若要代理程式使用目錄伺服器者，請在代理程式實例配置檔案 `config.properties` 中設定一些特性值。這些特性可讓代理程式實例查詢 LDAP 伺服器以獲得有關使用者和群組的資訊。每當使用者嘗試連線到代理程式實例或執行某些訊息傳送作業時，代理程式就會查詢 LDAP 伺服器。

實例配置檔案在代理程式實例目錄下的目錄中。路徑格式如下：

```
.../instances/instanceName/props/config.properties
```

如需作業系統特定的實例目錄位置之詳細資訊，請參閱附錄 A 「[Message Queue 資料的作業系統特定位置](#)」。

► 編輯配置檔案以使用 LDAP 伺服器

1. 透過設定以下特性指定您要使用 LDAP 使用者儲存庫：

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

2. 設定 `imq.authentication.type` 特性，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 存取的代理程式特性。這些特性儲存在代理程式的實例配置檔案中。下一節會提供這個特性的摘要說明。

Message Queue 使用 JNDI API 與 LDAP 目錄伺服器通訊。請參閱 JNDI 文件，以獲得這些特性中所參照之語法和術語的更多相關資訊。Message Queue 會 Sun JNDI LDAP 供應商和簡單認證。

Message Queue 支援 LDAP 認證防故障備用：您可以指定用於嘗試認證的 LDAP 目錄伺服器清單（請參閱 `imq.user.repos.ldap.server` 特性的參照資訊）。

如需 LDAP 使用者儲存庫相關特性的設定範例，請參閱代理程式的 `config.properties` 檔案。

4. 如有必要，您需要編輯存取控制特性檔案中的使用者 / 群組和規則。如需有關使用存取控制特性檔案的更多資訊，請參閱第 140 頁「[授權使用者：存取控制特性檔案](#)」。

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特性：
 - 指定 LDAP 伺服器用於 SSL 通訊的連接埠。例如：


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

這些是 LDAP 的相關特性：

- `imq.user_repository.ldap.server`。 *host:port* 用於 LDAP 伺服器
- `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。
- `imq.user_repository.ldap.password`。與代理程式所用識別名稱相關聯的密碼。
- `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`、`cn`。
- `imq.user_repository.ldap.usrfilter`。用於使用者的 JNDI 搜尋篩選器。
- `imq.user_repository.ldap.grpsearch`。指定您是否要啟用群組搜尋的布林。
- `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`。用於群組的 JNDI 搜尋篩選器。
- `imq.user_repository.ldap.timeout`。指定搜尋時間限制的整數 (以秒為單位)。
- `imq.user_repository.ldap.ssl.enabled`。指定代理程式與 LDAP 通訊時是否應使用 SSL 協定的布林。

如需這些特性的完整參照資訊，請參閱第 293 頁「安全性管理程式特性」。

設定管理員的存取控制

若要建立管理使用者，您可以使用存取控制特性檔案，指定可以建立 ADMIN 連線的使用者和群組。這些使用者和群組必須預先在 LDAP 目錄中定義。

任何使用者或群組只要能夠建立 ADMIN 連線，即可下達管理指令。

► 設定管理使用者

1. 將代理程式特性 `imq.accesscontrol.enabled` 設定為 `true` 預設值，啟用存取控制檔案。

`imq.accesscontrol.enabled` 特性可以啟用存取控制檔案。

2. 開啓存取控制檔案 `accesscontrol.properties`。附錄 A「[Message Queue 資料的作業系統特定位置](#)」中列出此檔的位置。

檔案中包含一個項目，大致如下：

```
service connection access control
#####
connection.NORMAL.allow.user=*
connection.ADMIN.allow.group=admin
```

列出的項目僅為範例。請注意基於檔案的使用者儲存庫中有 `admin` 群組，但是依預設，LDAP 目錄中沒有此群組。您必須將 LDAP 目錄中定義的群組名稱，換成要授與 Message Queue 管理員權限的群組名稱。

3. 若要將 Message Queue 管理員權限授與使用者，請如下列所示，輸入使用者名稱：

```
connection.ADMIN.allow.user=userName [, userName2, ..]
```

4. 若要將 Message Queue 管理員權限授與群組，請如下列所示，輸入群組名稱：

```
connection.ADMIN.allow.group=groupName [, groupName2, ..]
```

授權使用者：存取控制特性檔案

存取控制特性檔案 (ACL 檔案) 包含的規則，指定使用者和群組使用者可以執行的作業。您可以編輯 ACL 檔案，限制某些使用者和群組執行作業。您可以在每個代理程式實例使用不同的 ACL 檔案。

當用戶端應用程式執行以下作業之一時，代理程式會檢查其 ACL 檔案：

- 建立連線

- 建立生產者
- 建立用戶
- 瀏覽佇列

代理程式會檢查 ACL 檔案，判斷這位提出要求的使用者或使用者所屬的群組是否已被授權執行此作業。

如果您編輯 ACL 檔案，新的設定會在下次代理程式檢查檔案驗證授權時生效。您編輯此檔案後，不需要重新啟動代理程式。

無論使用者資訊是放置在文字檔案使用者儲存庫（請參閱第 132 頁「使用文字檔案使用者儲存庫」）或 LDAP 使用者儲存庫（請參閱第 137 頁「將 LDAP 伺服器用於使用者儲存庫」）中，均可使用 ACL 檔案。

建立存取控制特性檔案

ACL 檔案為特定實例。每次啟動代理程式實例時，會在實例目錄中建立預設檔案 `accesscontrol.properties`。此檔案的路徑格式如下（請參閱附錄 A「[Message Queue 資料的作業系統特定位置](#)」）：

```
.../instances/brokerInstanceName/etc/accesscontrol.properties
```

ACL 檔案的格式與 Java 特性檔案的格式類似。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
- 實體目標存取控制
- 實體目標自動建立存取控制

`version` 特性定義 ACL 特性檔案的版本；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。

存取規則的語法

在 ACL 特性檔案中，存取控制定義了特定使用者或群組對受保護的資源（如實體目標和連線服務）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 特性表示：

這些規則的基本語法如下：

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

表 7-3 描述了語法規則的元素。

表 7-3 存取規則的語法元素

元素	說明
<i>resourceType</i>	為以下之一：connection、queue 或 topic。
<i>resourceVariant</i>	<i>resourceType</i> 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實體目標。
<i>operation</i>	值取決於要制定的存取規則種類。
<i>access</i>	為以下之一：allow 或 deny。
<i>principalType</i>	為以下之一：user 或 group。如需更多資訊，請參閱第 135 頁「群組」。
<i>principals</i>	可以擁有在規則左側指定的存取權的使用者。如果 <i>principalType</i> 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 <i>principalType</i> 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=*
```

備註 若要指定非 ASCII 的使用者、群組或目標名稱，請使用 Unicode 退出 (\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/1.4/docs/guide/intl/faq.html>

如何計算許可權

檔案中有多個存取規則時，許可權計算如下：

- 特定存取規則覆寫一般存取規則。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* 的存取權覆寫賦予 * *principal* 的存取權。以下規則拒絕 Bob 向 tq1 產生訊息，但允許其他所有使用者產生訊息。

```
queue.tq1.produce.allow.user=*
queue.tq1.produce.deny.user=Bob
```

- 用於使用者的 * *principal* 規則覆寫用於群組的相應 * *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會覆寫授予使用者群組的存取權。在下列範例中，即使 Bob 是 User 成員，也無法產生訊息傳給 tq1。User 的所有成員可以執行這項作業。

```
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，則所有使用者無法執行所有作業。
- 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 q1：

```
queue.q1.browse.allow.user=Bob
queue.q1.browse.deny.user=Bob
```

以下兩條規則導致 User 群組無法使用 q5 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。

連線服務的存取控制

ACL 特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。您只能授權存取這些連線服務類型的預定值。

預設 ACL 特性檔案授權所有使用者存取 NORMAL 連線服務，並授權 admin 群組中的使用者存取 ADMIN 連線服務：

```
connection.NORMAL.allow.user=*
connection.ADMIN.allow.group=admin
```

如果您使用基於檔案的使用者儲存庫，`imqusermgr` 會建立預設群組 `admin`。如果您使用 LDAP 使用者儲存庫，您只要執行下列一項作業即可使用預設 ACL 特性檔案：

- 在 LDAP 目錄中定義 `admin` 群組。
- 將 ACL 特性檔案中的 `admin` 名稱，換成 LDAP 目錄中定義的一或多個群組名稱。

您可以限制連線存取權限。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。

您使用 ACL 特性檔案授權存取 ADMIN 連線的方式，與基於檔案的使用者儲存庫和 LDAP 使用者儲存庫有下列不同：

- **基於檔案的使用者儲存庫**
 - 如果存取控制停用，`admin` 群組中的使用者有 ADMIN 連線權限。
 - 如果存取控制已啟用，請編輯 ACL 檔案。請明確的授與使用者或群組存取 ADMIN 連線服務的權限。
- **LDAP 使用者儲存庫**。如果您使用 LDAP 使用者儲存庫，請執行下列所有作業：
 - 啟用存取控制。
 - 編輯 ACL 檔案，並提供可以建立 ADMIN 連線的使用者或群組名稱。指定 LDAP 目錄伺服器中定義的任何使用者或群組。

實體目標的存取控制

存取控制特性檔案的目標存取控制部分，包含基於實體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（實體目標）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實體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實體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 7-4 描述了這些元素：

表 7-4 實體目標存取控制規則的元素

元件	說明
<i>resourceType</i>	可能是 <code>queue</code> 或 <code>topic</code> 。
<i>resourceVariant</i>	實體目標名稱或所有實體目標 (*)，表示所有佇列或所有主題。
<i>operation</i>	可能是 <code>produce</code> 、 <code>consume</code> 或 <code>browse</code> 。
<i>access</i>	可能是 <code>allow</code> 或 <code>deny</code> 。
<i>principalType</i>	可能是 <code>user</code> 或 <code>group</code> 。

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實體目標存取控制規則：

- 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`

自動建立實體目標的存取控制

ACL 特性檔案的最後部分，包括指定代理程式將為哪些使用者和群組自動建立實體目標的存取規則。

當用戶在並不存在的實體目標上建立生產者或用戶時，如果代理程式的自動建立特性已啟用，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實體目標。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。

實體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，實體目標自動建立規則的效果必須與實體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實體目標，但不會將訊息傳送至此目標。

使用基於 SSL 的服務

基於安全套接層 (SSL) 標準的連線服務，可以在用戶端與代理程式之間傳送加密訊息。本節介紹如何設定基於 SSL 的連線服務。

Message Queue 支援基於安全套接層 (SSL) 標準的下列連線服務：

- `ssljms`、`ssladmin` 和 `cluster` 在 TCP/IP 上使用。
- `httpsjms` 在 HTTP 上使用。

這些連線服務允許在用戶端和代理程式之間傳送加密訊息。Message Queue 支援基於自身簽名伺服器憑證或簽名憑證的 SSL 加密。

若要使用基於 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於自身簽名的憑證中，此憑證被傳送至請求與代理程式連線的用戶端，此用戶端使用此憑證設定加密連線。

雖然 Message Queue 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。

本節還介紹透過 TCP/IP 的安全連線。

使用者透過 HTTP 基於 SSL 的連線服務 httpsjms，能讓用戶端和代理程式使用 HTTPS 通道 Servlet，建立安全連線。如需透過 HTTP 設定安全連線的詳細資訊，請參閱附錄 C 第 341 頁「HTTP/HTTPS 支援」。

TCP/IP 的安全連線服務

下列基於 SSL 的連線服務可提供透過 TCP/IP 的直接且安全的連線：

- ssljms 在用戶端與代理程式之間，透過安全且加密的連線傳送訊息。
- ssladmin 服務在 Message Queue 指令公用程式 (imqcmd) 和代理程式之間，建立安全且加密的連線。管理主控台 (imqadmin) 不支援安全連線。
- cluster 服務傳送訊息，並且在叢集中的代理程式之間，透過安全的加密連線，提供代理程式互通通訊 (請參閱第 182 頁「代理程式之間的安全連線」)。

配置使用自身簽名的憑證

本節描述如何使用自身簽名的憑證，設定基於 SSL 的服務。

若要提高認證層級，您可以使用認證機構驗證的簽名憑證。請先遵循本節中的步驟，再到第 152 頁「配置使用簽名的憑證」執行其他步驟。

► 設定基於 SSL 的連線服務

1. 產生自身簽名的憑證。
2. 啟用代理程式中的 ssljms、ssladmin 或 cluster 連線服務。
3. 啟動代理程式。
4. 設定並執行用戶端 (僅套用於 ssljms 連線服務)。

除步驟 4 (配置並執行用戶端) 之外，設定 ssljms 和 ssladmin 連線服務的程序是相同的。

以下各節較詳細地說明每個步驟。

步驟 1. 產生自身簽名的憑證

Message Queue 在假設用戶端正與已知且可信任的伺服器通訊的情況下，SSL 支援搭配自身簽名的憑證，可保護線上資料。

執行 `imqkeytool` 指令，以便為代理程式產生自身簽名的憑證。在 UNIX® 系統上，為了擁有建立密鑰儲存的許可權，您可能需要以超級使用者 (root) 的身份執行 `imqkeytool`。

同樣的憑證可以用於 `ssljms`、`ssladmin` 或 `cluster` 連線服務。

在命令提示字元下輸入以下內容：

```
imqkeytool -broker
```

公用程式會提示您輸入密鑰儲存密碼。

```
代理程式正在產生密鑰儲存 ...
輸入密鑰儲存密碼：
```

接著，公用程式會提示您輸入憑證所屬代理程式的識別資訊。您提供的資訊會製成 X.500 識別名稱。下表為每個提示列出提示及其說明和範例。這些值不區分大小寫，而且可以包含空格。

表 7-5 自身簽名的憑證需要的識別名稱資訊

提示	說明	範例
您的姓名是？	X.500 <code>commonName (CN)</code> 。請輸入執行代理程式伺服器完全合格的名稱。	<code>myhost.sun.com</code>
您的組織單位名稱是什麼？	X.500 <code>organizationalUnit (OU)</code> 。請輸入部門或事業單位名稱。	<code>purchasing</code>
您的組織名稱是什麼？	X.500 <code>organizationName (ON)</code> 。較大的組織名稱，例如公司或政府實體。	<code>My Company, Inc.</code>
您的城市或所在地名稱是什麼？	X.500 <code>localityName (L)</code> 。	<code>San Francisco</code>
您的州或省名稱是什麼？	X.500 <code>stateName (ST)</code> 。請輸入州或省的名稱，不使用縮寫。	<code>California</code>
這個單位的兩個字母的國碼是什麼？	X.500 <code>country (C)</code> 。	<code>US</code>

輸入完畢後，`imqkeytool` 會顯示您輸入的資訊，請您確認。例如：

```
Is CN=mqserver.sun.com, OU=purchasing, O=My Company, Inc., L=San  
Francisco, ST=California, C=US correct?
```

若要重新輸入這些值，請接受預設值或輸入 `no`；若要接受現行值並繼續執行，請輸入 `yes`。您確認後，`imqkeytool` 會在產生鍵對時暫停。

接著，`imqkeytool` 會要求您提供密碼，以鎖定特定的鍵對（鍵密碼）。請對此提示按 `Return` 鍵，表示鍵密碼和密鑰儲存密碼相同。

備註 請記住您提供的密碼。您啓動代理程式時必須提供這個密碼，代理程式才能開啓密鑰儲存。您可以將密鑰儲存密碼儲存在密碼檔案中（請參閱第 155 頁「使用密碼檔案」）。

執行 `imqkeytool` 可執行 `JDK keytool` 公用程式，以產生自身簽名的憑證，並將其放在 `Message Queue` 密鑰儲存中，且如附錄 A「[Message Queue 資料的作業系統特定位置](#)」中所示，其目錄位置因作業系統而異。

此密鑰儲存的格式與 `JDK1.2 keytool` 公用程式支援的格式相同。

這些是 `Message Queue` 密鑰儲存的配置特性：

- `imq.keystore.file.dirpath`。用於基於 SSL 的服務：指定包含密鑰儲存檔案之目錄的路徑。如需預設值，請參閱附錄 A「[Message Queue 資料的作業系統特定位置](#)」。
- `imq.keystore.file.name`。用於基於 SSL 的服務：指定密鑰儲存檔案的名稱。
- `imq.keystore.password`。用於基於 SSL 的服務：指定密鑰儲存密碼。

您可能需要重新產生鍵對以解決某些問題；例如：

- 您忘記了密鑰儲存密碼。
- 您啓動代理程式時，基於 SSL 的服務無法初始化，並且您收到以下異常：
`java.security.UnrecoverableKeyException: 無法回復恢復密鑰。`

可能導致此異常的原因為，您提供的鍵密碼與您在第 148 頁「[步驟 1. 產生自身簽名的憑證](#)」中產生自身簽名的憑證時的密鑰儲存密碼不同。

► 若要重新產生鍵對

1. 移除代理程式的密鍵儲存，如附錄 A 「[Message Queue 資料的作業系統特定位置](#)」中所示
2. 重新執行 `imqkeytool` 以產生鍵對，如第 148 頁「[步驟 1. 產生自身簽名的憑證](#)」中所述。

步驟 2. 啓用代理程式中基於 SSL 的服務

若要啓用代理程式中基於 SSL 的服務，您需要將 `ssljms` (或 `ssladmin`) 新增到 `imq.service.activelist` 特性。

備註 請使用 `imq.cluster.transport` 特性啓用基於 SSL 的 `cluster` (叢集) 連線服務，而非使用 `imq.service.activelist` 特性。請參閱第 182 頁「[代理程式之間的安全連線](#)」。

► 若要啓用代理程式中基於 SSL 的服務

1. 開啓代理程式的實例配置檔案。

實例配置檔案位於以代理程式實例名稱 (*instanceName*) 作為辨別依據的目錄中，且代理程式實例與配置檔案相互關聯 (請參閱附錄 A 「[Message Queue 資料的作業系統特定位置](#)」) :

```
.../instances/instanceName/props/config.properties
```

2. 新增用於 `imq.service.activelist` 特性的項目 (如果項目不存在的話)，並將基於 SSL 的服務納入清單。

依預設，特性包括 `jms` 和 `admin` 連線服務。您必須新增 `ssljms` 或 `ssladmin` 連線服務，或兩者皆新增 (因您要啓動的服務而異)。

```
imq.service.activelist=jms,admin,ssljms,ssladmin
```

步驟 3. 啓動代理程式

啓動代理程式，提供密鑰儲存密碼。您可以以下任何一種方法提供密碼：

- 當代理程式啓動時讓它提示您提供密碼

```
imqbrokerd
請輸入密鑰儲存密碼：mypassword
```

- 將密碼放在密碼檔案中，請參閱第 155 頁「[使用密碼檔案](#)」中的描述。您將密碼放在密碼檔案中並且設定特性為 `imq.passfile.enabled=true` 後，請執行下列其中一項操作：

- 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特性指定密碼檔案的位置：

```
imq.passfile.dirpath=/tmp
```

```
imq.passfile.name=mypassfile
```

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 Message Queue 使用 JSSE (Java 安全套接延伸) 以實作 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，請將用戶端配置為使用安全連線服務。TCP/IP 上有兩種安全連線情形：

- 應用程式用戶端使用 `ssljms`
- Message Queue 管理用戶端 (例如 `imqcmd`) 使用 `ssladmin`

這兩種用戶端將在以下各節中分別描述。

應用程式用戶端使用 `ssljms`

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、jcert.jar 和 jndi.jar
```

2. 確保用戶端在其類別路徑中擁有以下 Message Queue JAR 檔案：

```
imq.jar 和 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 *Message Queue Developer's Guide for Java Clients* 中有關使用受管理物件的章節。

管理用戶端 (`imqcmd`) 使用 `ssladmin`

您可以透過使用 `imqcmd` 時包括的 `-secure` 選項，建立安全管理連線。例如：

```
imqcmd list svc -b hostName:port -u adminName -secure
```

其中 `adminName` 是 Message Queue 使用者儲存庫中的有效項目，指令會提示您輸入密碼。(如果您是使用文字檔案儲存庫，請參閱第 137 頁「變更預設管理員密碼」)。

列出連線服務是顯示 `ssladmin` 服務正在執行的一種方法，也是您可以成功建立安全 `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:
```

主機	主要連接埠	
localhost	7676	
服務名稱	連接埠號	服務狀態
admin	33984 (動態)	執行中
httpjms	-	未知
httpsjms	-	未知
jms	33983 (動態)	執行中
ssladmin	35988 (動態)	執行中
ssljms	動態	未知

成功列出服務。

配置使用簽名的憑證

簽名憑證提供的伺服器認證層級，高於自身簽名憑證的伺服器認證層級。若要實作簽名的憑證，請在密鑰儲存中安裝簽名的憑證，然後配置 Message Queue 用戶端在與 `imqbrokerd` 建立 SSL 連線時要求簽名的憑證。

您只能在用戶端和代理程式之間（不能在叢集的多個代理程式之間）實作簽名的憑證。

以下的指示是假設您已完成在第 147 頁「配置使用自身簽名的憑證」文件中的步驟。當您遵循指示時，存取 <http://java.sun.com> 上有關 J2SE `keytool` 和 X.509 憑證的資訊可能有所幫助。

步驟 1：取得並安裝簽名的憑證

► 若要取得簽名的憑證

1. 使用 J2SE keytool，為您剛產生的自身簽名的憑證，產生憑證簽名請求 (Certificate Signing Request, CSR)。

範例：

```
keytool -certreq -keyalg RSA -alias imq -file certreq.csr
        -keystore /etc/imq/keystore -storepass myStorePassword
```

現在 CSR 將憑證封裝在 certreq.csr 檔案中。

2. 請使用下列其中一種方法，產生或請求簽名憑證：
 - 請 Thawte 或 Verisign 等知名的認證機構 (CA) 簽署憑證。如需這項程序的詳細資訊，請參閱 CA 文件。
 - 使用 SSL 簽名軟體套裝軟體，自行簽署憑證。

簽出的憑證是一串 ASCII 字元。如果您收到 CA 簽名的憑證，它可能是一個電子郵件附件或是訊息中的文字。

3. 得到簽名的憑證後，請將它存入檔案中。

這些指示使用 broker.cer 範例名稱來代表代理程式憑證。

► 若要安裝簽名的憑證

1. 檢查 \$JAVA_HOME/lib/security/cacerts，查看 J2SE 是否如下所示，預設支援 CA：

```
keytool -v -list -keystore $JAVA_HOME/lib/security/cacerts
```

指令列出系統密鑰儲存中的根 CA。

如果有列出您的 CA，請跳過下一個步驟。

2. 如果 J2SE 未支援您的 CA，請將認證機構的根憑證匯入 imqbrokerd 密鑰儲存中。

範例：

```
keytool -import -alias ca -file ca.cer -noprompt -trustcacerts
        -keystore /etc/imq/keystore -storepass myStorePassword
```

ca.cer 值是來自 CA 的 CA 根憑證。

如果您使用 CA 測試憑證，可能需要匯入「測試 CA 根」憑證。您的 CA 應該會指示如何取得「測試 CA 根」憑證。

3. 將簽名的憑證匯入密鑰儲存中，取代原來的自身簽名的憑證。

例如：

```
keytool -import -alias imq -file broker.cer -noprompt -trustcacerts  
-keystore /etc/imq/keystore -storepass myStorePassword
```

broker.cer 值是您取自 CA 簽名的憑證所在的檔案。

現在 imqbrokerd 密鑰儲存有簽名的憑證可用於 SSL 連線。

步驟 2：配置用戶端執行階段以請求簽名的憑證

► 配置 Java 用戶端執行階段

依預設，Message Queue 用戶端執行階段信任 imqbrokerd，並接受任何呈上的憑證。現在您必須配置用戶端執行階段以請求簽名的憑證，並確保用戶端信任簽署憑證的 CA。

1. 若要配置用戶端向 imqbrokerd 請求有效簽名的憑證，請將用戶端 ConnectionFactory 物件的 imqSSLIsHostTrusted 屬性設定為 false。
2. 請如第 151 頁「步驟 4. 配置並執行基於 SSL 的用戶端」所描述，嘗試與 imqbrokerd 建立 SSL 連線。

如果 broker 的憑證是由知名的 CA 簽署，可能會連線成功，您可以跳過下一個步驟。如果連線失敗出現憑證驗證錯誤，請執行下一個步驟。

3. 請如下面幾節所描述，在用戶端的信任儲存中安裝負責簽名的 CA 根憑證。

配置有信任儲存的用戶端時，有三個選項：

- 將根 CA 安裝到預設系統 cacerts 檔案中。
- 將根 CA 安裝到替代系統檔案 jssecacerts 中。建議您使用這個選項。
- 將根 CA 安裝到任何密鑰儲存檔案中，並且配置用戶端將此視為信任儲存。

下面幾節中有如何使用這些選項來安裝「Verisign 測試根 CA」的範例。根 CA 在 testrootca.cer 檔案中。這些範例假設 J2SE 安裝在 /usr/j2se 中。

安裝到預設系統 cacerts 檔案中

這個範例將根 CA 安裝到 \$JAVA_HOME/usr/jre/lib/security/cacerts 檔案中。

```
keytool -import -keystore /usr/j2se/jre/lib/security/cacerts  
-alias VerisignTestCA -file testrootca.cer -noprompt  
-trustcacerts -storepass myStorePassword
```

依預設，用戶端會搜尋這個密鑰儲存，因此不需要進一步配置用戶端。

安裝到 *jssecacerts* 中

這個範例將根 CA 安裝到 `$JAVA_HOME/usr/jre/lib/security/jssecacerts` 檔案中。

```
keytool -import -keystore /usr/j2se/jre/lib/security/jssecacerts
        -alias VerisignTestCA -file testrootca.cer -noprompt
        -trustcacerts -storepass myStorePassword
```

依預設，用戶端會搜尋這個密鑰儲存，因此不需要進一步配置用戶端。

安裝到其他檔案中

這個範例將根 CA 安裝到 `/home/smith/.keystore` 檔案中。

```
keytool -import -keystore /home/smith/.keystore
        -alias VerisignTestCA -file testrootca.cer -noprompt
        -trustcacerts -storepass myStorePassword
```

依預設，用戶端不搜尋這個密鑰儲存，因此您必須為用戶端提供信任儲存的位置。若要執行這項操作，請在用戶端執行時，設定 `javax.net.ssl.trustStore` Java 系統的特性。例如：

```
javax.net.ssl.trustStore=/home/smith/.keystore
```

使用密碼檔案

有些類型的指令需要密碼。表 7-6 中的第一欄列出需要密碼的指令，第二欄列出需要密碼的原因。

表 7-6 使用密碼的指令

指令	用途	密碼的用途
<code>imqbrokerd</code>	啓動代理程式	存取外掛持續性資料儲存、SSL 憑證密鑰儲存或 LDAP 使用者儲存庫
<code>imqcmd</code>	管理代理程式	認證已授權使用指令之管理使用者
<code>imqdbmgr</code>	管理外掛資料儲存	存取資料儲存

您可以在密碼檔案 (*passfile*) 中指定這些密碼，並且使用 `-passfile` 選項來指定檔案名稱。下面是 `-passfile` 選項的格式：

```
imqbrokerd -passfile myPassfile
```

備註 您可以在舊版的指令行中使用 `-p`、`-password`、`-dbpassword` 和 `-ldappassword` 選項來指定密碼。但是這些選項已經被停用，將從未來的版本中移除。本版指令行中這些選項的任一值，會停用密碼檔案中的相關值。

安全考量

除非別人可以看到您的監視器，否則遵照提示來指定密碼是最安全的指定密碼方法。您也可以從指令行指定密碼檔案。但若使用非互動式指令，就必須使用密碼檔案。

密碼檔案沒有加密，您必須設定許可權，防止無權限者存取。設定許可權時請限制使用者檢視檔案，但是允許啟動代理程式的使用者讀取此檔。

密碼檔案的內容

密碼檔案是純文字檔，包含一組特性和值。每一個值是指令使用的密碼。

密碼檔案可以包含表 7-7 中所示的密碼：

表 7-7 密碼檔案中的密碼

密碼	影響的指令	說明
<code>imq.imqcmd.password</code>	<code>imqcmd</code>	指定 <code>imqcmd</code> 指令行的管理員密碼。每個指令都會認證此密碼。
<code>imq.keystore.password</code>	<code>imqbrokerd</code>	為基於 SSL 的服務指定密鑰儲存密碼。
<code>imq.persist.jdbc.password</code>	<code>imqbrokerd</code> <code>imdbmgr</code>	如果必需，則指定用於開啓資料庫連線的密碼。
<code>imq.user_repository.ldap.password</code>	<code>imqbrokerd</code>	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。

Message Queue 產品有提供密碼檔案範例。有關此範例檔案的位置，請參閱附錄 A 「Message Queue 資料的作業系統特定位置」。

建立稽核記錄

Message Queue 僅 Enterprise Edition 支援稽核記錄。啓用稽核記錄後，Message Queue 會記錄下列類型的事件：

- 啓動、關機、重新啓動及移除代理程式實例
- 使用者認證和授權
- 重設永久性儲存
- 建立、清除及銷毀實體目標
- 管理長期用戶銷毀

若要將稽核記錄在 Message Queue 代理程式記錄檔中，請將 `imq.audit.enabled` 代理程式特性設定為 `true`。記錄檔中的所有稽核記錄都有關鍵字 `AUDIT`。

如需 `imq.audit.enabled` 特性的參照資訊，請參閱第 293 頁「[安全性管理程式特性](#)」。

建立稽核記錄

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。受管理物件可封裝提供者特定的配置和命名資訊。

Message Queue 管理員通常會為用戶端應用程式建立受管理物件，使其能夠與代理程式建立連線。用戶端應用程式會使用連線來向實體目標傳送訊息並從實體目標接收訊息。

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。因為這些工作涉及您使用的物件儲存與您建立的受管理物件之屬性理解，在描述如何使用 `imqobjmgr` 管理受管理物件前，本章會先提供這兩個主題的背景。

本章包含以下各節：

- [第 160 頁](#) 「關於物件儲存」
- [第 162 頁](#) 「關於受管理物件屬性」
- [第 169 頁](#) 「使用物件管理員公用程式 (`imqobjmgr`)」
- [第 172 頁](#) 「新增和刪除受管理物件」
- [第 176 頁](#) 「列出受管理物件」
- [第 176 頁](#) 「取得有關單一物件的資訊」
- [第 177 頁](#) 「更新受管理物件」

關於物件儲存

受管理物件放在即時可用的物件儲存中，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：標準的 LDAP 目錄伺服器或檔案系統物件儲存。

LDAP 伺服器物件儲存

LDAP 伺服器是用於生產訊息傳送系統的建議物件儲存。LDAP 實作可從一些供應商處獲得，並設計為在分散式系統中使用。LDAP 伺服器也在生產環境中提供有利的安全性功能。

Message Queue 管理工具可管理 LDAP 伺服器上的物件儲存。然而，依 LDAP 伺服器文件規定，您可能必須先配置 LDAP 伺服器，以便儲存 Java 物件與執行 JNDI 查找。

使用 LDAP 伺服器作為物件儲存時，您必須指定表 8-1 中顯示的屬性。這些屬性分為以下幾個種類：

- **初始環境**：此屬性的值針對 LDAP 伺服器物件儲存進行修正。
- **位置**：指定 LDAP 伺服器中，設定用來儲存您受管理物件的 URL 與目錄路徑。要特別注意的是，您必須檢查指定的路徑是否存在。
- **安全性資訊**：根據 LDAP 提供者而異。您應查閱 LDAP 實作隨附的文件，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。

表 8-1 LDAP 物件儲存屬性

屬性	說明
java.naming.factory.initial	LDAP 伺服器上 JNDI 查找的初始環境 com.sun.jndi.ldap.LdapCtxFactory
java.naming.provider.url	LDAP 伺服器 URL 與目錄路徑資訊。例如： ldap://mydomain.com:389/ou=mqobjs,o=myapp 受管理物件儲存於 /myapp/mqobjs 目錄中。
java.naming.security.principal	認證 LDAP 伺服器之呼叫者的負責人的身份。此項目的格式取決於認證方案。例如： uid=fooUser, ou=People, o=mq 如果沒有指定此特性，則運作方式將由 LDAP 服務提供者決定。

表 8-1 LDAP 物件儲存屬性 (續)

屬性	說明
java.naming.security.credentials	<p>認證 LDAP 伺服器之呼叫者的負責人的憑證。此特性的值取決於認證方案：它可以是雜湊密碼、清除文字密碼、密鑰以及憑證等。例如：</p> <pre>fooPasswd</pre> <p>如果沒有指定此特性，則運作方式將由 LDAP 服務提供者決定。</p>
java.naming.security.authentication	<p>要使用的安全性級別。它的值為以下其中一各關鍵字：none、simple 和 strong。</p> <p>舉例來說，如果您指定 simple，則會提示您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。</p> <p>如果沒有指定此特性，則運作方式將由 LDAP 服務提供者決定。</p>

檔案系統物件儲存

Message Queue 還支援檔案系統物件儲存實作。當檔案系統物件儲存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。

但是，檔案系統儲存不可作為部署在多重電腦節點之用戶端的集中物件儲存，除非這些用戶端可以存取物件儲存所在的目錄。除此之外，任何對此目錄擁有存取權的使用者均可使用 Message Queue 管理工具，以建立並管理受管理物件。

使用檔案系統物件儲存時，您必須指定表 8-2 中顯示的屬性。這些屬性分為以下幾個種類：

- **初始環境**：此屬性的值針對檔案系統物件儲存進行修正。
- **位置**：此屬性的值指定儲存您受管理物件的目錄路徑。該目錄必須存在，且 Message Queue 管理工具的使用者與用戶端應用程式的使用者皆擁有適當的存取許可權，以便存取儲存。

表 8-2 檔案系統物件儲存屬性

屬性	說明
java.naming.factory.initial	<p>檔案系統物件儲存上 JNDI 查找的初始環境：</p> <pre>com.sun.jndi.fscontext.RefFSContextFactory</pre>

表 8-2 檔案系統物件儲存屬性 (續)

屬性	說明
<code>java.naming.provider.url</code>	目錄路徑資訊例如： <code>file:///C:/myapp/mqobjs</code>

關於受管理物件屬性

Message Queue 受管理物件有兩種基本類型：

- **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用戶擷取訊息的目標。

Message Queue 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。本章只描述如何使用指令行。

使用屬性值對指定的受管理物件的屬性。

連線工廠屬性

連線工廠的配置會傳送至所有由連線工廠建立作為用戶端應用程式的連線。連線已配置成定義傳送或接收訊息的一方，可指定用戶端執行階段如何處理訊息流量，並為經由連線傳送的全部訊息自動設定某些資訊。

有兩種連線工廠物件：

- `ConnectionFactory` 支援一般訊息傳送和非分散式作業事件。
- `XAConnectionFactory` 支援分散式作業事件。

`ConnectionFactory` 與 `XAConnectionFactory` 物件共用同一組屬性。

連線工廠物件可由管理員或應用程式建立和配置 (以建立原型或測試為目的)。您可以使用 `imqobjmgr` 工具或管理主控台來設定連線工廠屬性。

本節描述以下各章節中的連線工廠屬性，這些章節是由屬性影響的運作方式所架構的。

- [第 163 頁「連線處理」](#)
- [第 165 頁「用戶端標識」](#)
- [第 166 頁「可靠性和流量控制」](#)

- 第 167 頁「佇列瀏覽器運作方式和伺服器階段作業」
- 第 168 頁「JMS 訊息標頭覆寫」
- 第 167 頁「JMS 定義的特性支援」

與您有主要關係的屬性為 `imqAddressList`，它可用於指定用戶端將與其建立連線的代理程式。第 173 頁「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時，如何指定屬性。

如需連線工廠屬性的參照資訊，請參閱第 16 章「受管理物件屬性參照」和 `Message Queue` 類別 `com.sun.messaging.ConnectionConfiguration` 的 `JavaDoc` API 文件。

連線處理

您可使用連線處理的屬性來指定要連線的訊息伺服器位址，如果需要重新連線，則要指定嘗試重新連線的次數，以及連線嘗試之間的間隔時間。

用戶端所連線的訊息伺服器位址，是您指定為 `imqAddressList` 屬性的數值。訊息伺服器位址包含代理程式主機名稱、連接埠號和連線服務類型。

連接埠的編號可以是代理程式的連接埠對映器常駐的連接埠，或是特定連線服務常駐的連接埠。如果您指定了連接埠對映器的連接埠，連接埠對映器會動態指定連線用的連接埠。如需指定訊息伺服器位址的完整資訊，請參閱第 310 頁「`imqAddressList` 屬性值的語法」。

自動重新連線

在單一代理程式環境或多重代理程式叢集環境中，您可以設定當連線失敗時，讓用戶端能夠自動重新連線至代理程式的連線處理屬性。您也可以配置重新連線的程序。

重新連線功能可提供連線防故障備用，但不提供資料防故障備用：當用戶端自動重新連線至不同的代理程式實例時，失敗或斷線代理程式所保留的永久性訊息和其他狀態資訊將會遺失。

如果自動重新連線啓用的話，`Message Queue` 會在連線失敗時保留暫存目標，以防用戶端可能會重新連線和再次存取。在讓用戶端有時間進行重新連線並使用這些目標之後，代理程式將刪除這些目標。

處理重新連線的方法取決於用戶端連線至單一代理程式，或是連線至叢集一部分的代理程式。以下各節描述了各種可能性。

重新連線至單一代理程式 若要讓用戶端能在連線失敗時自動重新連線至代理程式，您必須設定以下的連線工廠屬性：

- `imqReconnectEnabled`。啓用自動重新連線運作方式。

- `imqReconnectAttempts`。指定用戶端執行階段嘗試重新連線至用戶端的次數。
- `imqReconnectInterval`。指定嘗試重新連線至用戶端的間隔時間。

如需這些屬性的完整參照資訊，請參閱第 309 頁「[連線處理](#)」。

重新連線至叢集中的代理程式 在多重代理程式叢集的環境中，如果您指定 `imqAddressList` 屬性的多重位址，將重複自動重新連線清單中的代理程式。清單中的所有代理程式都必須以 **Message Queue Enterprise Edition** 安裝。

如果用戶端連線至清單中的第一個位址時失敗，用戶端執行階段會嘗試重新連線至清單中的另一個代理程式。如果嘗試連線失敗，用戶端執行階段會繼續在清單中逐一連線，直到能夠重新連線至用戶端為止。

如果嘗試連線不成功，用戶端執行階段會以指定的次數在清單中循環連線，直到找到可用的代理程式或無法找到任何的代理程式為止。`imqAddressListBehavior` 屬性的設定會決定，是要依照位址清單中的順序選擇重新連線的代理程式，還是在清單中隨機選擇。

若要讓用戶端能重新連線至叢集中的代理程式，請使用以下的屬性：

- `imqReconnectEnabled`。啟用自動重新連線運作方式。
- `imqReconnectAttempts`。指定在跳至下一個代理程式的位址之前，嘗試連線每個位址的次數。
- `imqReconnectInterval`。指定嘗試重新連線的間隔時間。
- `imqAddressListIterations`。指定在清單中重複連線的次數。
- `imqAddressListBehavior`。指定嘗試連線的次序為依據位址清單中的位址次序或是隨機次序。

如需這些屬性的完整參照資訊，請參閱第 309 頁「[連線處理](#)」。

連線偵測

`imqPingInterval` 屬性可指定從用戶端執行階段到代理程式的偵測作業頻率。用戶端執行階段會不定期測試連線，以預先偵測到失敗的連線。如果偵測作業失敗，用戶端執行階段會拋出異常至用戶端應用程式的異常偵聽程式物件中。如果應用程式中沒有異常偵聽程式，則下一個使用連線的嘗試會失敗。

偵測功能對於那些等待接收訊息而無法傳送訊息的使用者用戶端應用程式來說，特別的重要。否則這些應用程式將無法得知連線何時會失敗。不定期產生訊息的用戶端也可因此功能而受惠，因為它可以在需要傳送訊息之前處理失敗的連線。

依預設，偵測間隔設定在 30 秒。-1 的值會停用偵測作業。

對於連線中斷的回應會根據特定的作業系統而有所差異。例如，在一些作業系統上，偵測會立即報告失敗的情況。其他的作業系統可能會繼續嘗試建立與代理程式的連線，緩衝連續的偵測，直到偵測成功或緩衝區溢位。

如需 `imqPingInterval` 屬性的完整參照資訊，請參閱第 309 頁「連線處理」。

用戶端標識

`Message Queue` 會定義一組連線工廠屬性，以支援用戶端認證和唯一的用戶端 ID 設定，這是長期用戶所必需的。

嘗試連線至代理程式的用戶端必須經過認證。如果用戶端在建立連線時並未指定使用者名稱或密碼，會發生以下其中一種情況：

- 如果您未設定連線工廠屬性 `imqDefaultUsername` 和 `imqDefaultPassword`，用戶端執行階段會將數值 `guest/guest` 傳送至代理程式，而代理程式會使用這些數值來認證用戶端。

此使用者儲存庫出廠時隨附了 `guest/guest` 項目，所以用戶端將可取得連線。

- 如果您已設定連線工廠屬性 `imqDefaultUsername` 和 `imqDefaultPassword`，用戶端執行階段會將這些數值傳送至代理程式，而代理程式會使用這些數值來認證用戶端。

如果這對使用者 / 密碼位於使用者儲存庫中，則用戶端可取得連線。

此方案可允許任何使用者取得連線，這對於開發與測試來說非常的方便。在生產系統中，僅限於已新增至使用者儲存庫的使用者才能存取連線。

除了請求連線的用戶端之代理程式認證以外，JMS 規格會需要在維護用戶端狀態時，該連線能提供唯一的用戶端識別碼。`Message Queue` 使用用戶端 ID 來記錄長期用戶。如果長期用戶成為非使用中的用戶，代理程式會保留該用戶的訊息，等到用戶恢復作用時再傳送這些訊息。代理程式會藉由用戶端 ID 來識別用戶。

您可以使用管理級別設定 `ClientID`，或者用戶端可有計劃地設定 `ClientID`。如果多重用戶端從相同的連線工廠取得連線，請為連線工廠設定 `ClientID`。`Message Queue` 將能為每個從該工廠取得的連線提供唯一的 `ClientID`。

若要確保 `ClientID` 為唯一的值，請使用以下的格式來設定 `imqConfiguredClientID` 屬性：

```
imqConfiguredClientID=${u}string
```

`${u}` 必須為屬性值的前四個字元。如果不是出現 "u" 的字元，在建立連線時會發生 JMS 異常。

string 數值是您要與連線工廠所生產的連線發生關聯的數值，例如 `Xconn`。在使用者認證階段，`Message Queue` 會以 `u:userName` 替代 `u`。例如，如果與連線有關聯的使用者為 `Athena` 而指定給連線的字串為 `${u}Xconn`，則 `ClientID` 將為 `u:AthenaXconn`。

此方案將可確保每個由連線工廠所產生的連線都將包含 `ClientID`。

有一種方案無法運作的情況：如果兩個用戶端使用如 `guest` 的預設使用者名稱來取得連線，則都會有相同 `${u}` 元件的 `ClientID`。在執行階段中，第一個要求連線的用戶端可取得連線，第二個則不行，因為 `MQ` 無法以非唯一的 `ClientID` 來建立連線。

您可以設定 `imqDisableSetClientID` 屬性，以防止使用連線工廠的用戶端有計劃地變更已配置用戶端 ID。

您必須設定長期訂閱的 `imqConfiguredClientID` 屬性，除非應用程式使用的是 `setClientId()` 方法。

總而言之，這些就是影響用戶端識別的屬性：

- `imqDefaultUsername`。當用戶端未指定建立連線的使用者名稱時，可指定由代理程式認證的預設使用者名稱。
- `imqDefaultPassword`。當用戶端未指定建立連線的密碼時，可指定由代理程式認證的預設密碼。
- `imqConfiguredClientID`。指定以管理級別配置的用戶端 ID 值。
- `imqDisableSetClientID`。指定使用連線工廠的用戶端是否可以有計劃地變更用戶端 ID。

如需這些屬性的完整參照資訊，請參閱第 312 頁「用戶端標識」。

可靠性和流量控制

用戶端傳送和接收的訊息與 `Message Queue` 所使用的控制訊息，皆會忽略相同用戶端與代理程式間的連線。如果 `JMS` 訊息傳送阻擋控制訊息，則會造成控制訊息（例如代理程式確認）傳送的延遲。

您可以設定連線工廠屬性，讓您管理與用戶端訊息流量相關的控制訊息流量。控制這兩種訊息的流量涉及流量與可靠性之間的洩漏。如需如何使用這些屬性以管理流量控制與可靠性的說明，請參閱第 225 頁「用戶端執行階段訊息流量調整」。

以下屬性會影響用戶端和控制訊息的流量：

- `imqAckTimeout`。指定（以毫秒為單位）用戶端執行階段等待任何代理程式回應的最長時間。
- `imqConnectionFlowCount`。指定計數批次中的最大 `JMS` 訊息數。

- `imqConnectionFlowLimitEnabled`。限制連線層級的訊息流量。
- `imqConnectionFlowLimit`。指定可透過連線傳送的，以及可以在用戶端執行階段緩衝待用的訊息數量限制。
- `imqConsumerFlowLimit`。指定每個用戶可透過連線傳送的，以及可以在用戶端執行階段緩衝待用的訊息數量限制。
- `imqConsumerFlowThreshold`。指定 `imqConsumerFlowLimit` 的百分比，此為用戶端執行階段緩衝的每位用戶之訊息數，低於該值時，便會繼續傳送訊息給用戶。

如需這些屬性的完整參照資訊，請參閱第 313 頁「穩定性與流量控制」。

佇列瀏覽器運作方式和伺服器階段作業

這些屬性影響用戶端佇列瀏覽：

- `imqQueueBrowserMaxMessagesPerRetrieve`。指定瀏覽佇列目標內容時，用戶端一次將擷取的最大訊息數量。
- `imqQueueBrowserRetrieveTimeout`。指定瀏覽佇列目標內容時，用戶端擷取訊息所需等待的時間。
- `imqLoadMaxToServerSession`。JMS 應用程式伺服器設備，指定 `Message Queue ConnectionConsumer` 是否應該將 `maxMessages` 上載到 `ServerSession` 階段作業，或一次載入一個訊息。

如需這些屬性的完整參照資訊，請參閱第 314 頁「佇列瀏覽器運作方式和伺服器階段作業」。

JMS 定義的特性支援

您可以使用連線工廠屬性，在連線所產生的訊息上自動設定 JMS 定義的特性。JMS 特性是在 JMS 規格中所定義，位於

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

使用以下屬性來設定 JMS 定義的特性：

- `imqSetJMSXUserID`。對已產生的訊息，指定 `Message Queue` 是否要設定 JMS 定義的特性 `JMSXUserID` (傳送訊息的使用者識別)。
- `imqSetJMSXAppID`。對已產生的訊息，指定 `Message Queue` 是否要設定 JMS 定義的特性 `JMSXAppID` (傳送訊息的應用程式識別)。
- `imqSetJMSXProducerTXID`。對已產生的訊息，指定 `Message Queue` 是否要設定 JMS 定義的特性 `JMSXProducerTXID` (產生了此訊息的作業事件處理識別碼)。
- `imqSetJMSXConsumerTXID`。對已使用的訊息，指定 `Message Queue` 是否要設定 JMS 定義的特性 `JMSXConsumerTXID` (使用了此訊息的作業事件處理識別碼)。

- `imqSetJMSXrvcTimestamp`。對已使用的訊息，指定 `Message Queue` 是否要設定 JMS 定義的特性 `JMSXrvcTimestamp` (訊息傳送至用戶的時間)。

如需這些屬性的完整參照資訊，請參閱第 315 頁「JMS 定義的特性支援」。

JMS 訊息標頭覆寫

您可以設定連線工廠的屬性，以覆寫指定訊息的永久性、使用期限和優先權的 JMS 訊息標頭欄位。此設定是用於從連線工廠取得的連線所生產之訊息。

可覆寫以下 JMS 欄位中的數值：

- `JMSDeliveryMode` (訊息永久性 / 非永久性)
- `JMSExpiration` (訊息使用期限)
- `JMSPriority` (訊息優先順序 - 整數 0 到 9)

如需有關這些欄位的更多資訊，請參閱

<http://java.sun.com/products/jms/docs.html> 中的 JMS 規格。

由於覆寫這些訊息標頭可能會干擾應用程式需求，請先諮詢應用程式使用者或設計者再使用此功能。

下表包含了處理訊息覆寫的連線工廠屬性。大部分的屬性都是成對的。每對的第一個屬性可指定是否能覆寫已指定的標頭欄位，第二個屬性則指定覆寫值。

- `imqOverrideJMSDeliveryMode` 和 `imqJMSDeliveryMode`。第一個屬性指定是否可覆寫用戶端設定的 `JMSDeliveryMode` 欄位，第二個屬性則指定其覆寫值。
- `imqOverrideJMSExpiration` 和 `imqJMSExpiration`。第一個屬性指定是否可覆寫用戶端設定的 `JMSExpiration` 欄位，第二個屬性則指定其覆寫值。
- `imqOverrideJMSPriority` 和 `imqJMSPriority`。第一個屬性指定是否可覆寫用戶端設定的 `JMSPriority` 欄位，第二個屬性則指定其覆寫值。
- `imqOverrideJMSHeadersToTemporaryDestinations`。指定臨時目標是否套用覆寫。

如需這些屬性的完整參照資訊，請參閱第 312 頁「訊息標頭覆寫」。

目標受管理物件屬性

目標受管理物件識別實體主題或佇列目標，其具有第 308 頁表 16-1 中列出的屬性。第 174 頁「新增主題佇列」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 `Message Queue` 類別 `com.sun.messaging.DestinationConfiguration` 的 JavaDoc API 文件。

使用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 `Message Queue` 受管理物件。使用此公用程式，您可以執行以下工作：

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

如需 `imqobjmgr` 指令的語法、子指令和選項的詳細資訊，請參閱第 13 章「指令參照」。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，您必須將以下資訊指定為 `imqobjmgr` 子指令的選項：

- **受管理物件類型**
第 271 頁表 13-11 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，以參照物件儲存中的受管理物件 (使用 JNDI)。
- **受管理物件屬性 (add 與 update 子指令特別需要)：**
 - 對於目標：代理程式上實體目標的名稱。這是使用 `imqcmd create dst` 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：大部分的通用屬性為位址清單 (mqAddressList)，它可指定用戶端嘗試連線的訊息伺服器位址 (一個或多個)。如果您沒有指定此資訊，則會使用本地主機與預設連接埠號 (7676)，這表示用戶端將會連線到本地主機連接埠 7676 上的代理程式。第 173 頁「新增連線工廠」一節介紹如何指定物件屬性。

如需其他屬性，請參閱第 162 頁「連線工廠屬性」。

- **物件儲存屬性**

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

- JNDI 實作的類型 (初始環境屬性)。例如，檔案系統或 LDAP。
- 物件儲存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)。

如需有關物件儲存屬性的更多資訊，請參閱第 160 頁「LDAP 伺服器物件儲存」和第 161 頁「檔案系統物件儲存」。

使用指令檔

mqobjmgr 指令可讓您指定使用 Java 特性檔案語法的指令檔的名稱，以表示所有或部分 mqobjmgr 子指令子句。

將指令檔與物件管理員公用程式 (mqobjmgr) 配合使用，對指定物件儲存屬性極為有用，這可能與跨 mqobjmgr 的多重呼叫一樣，並且一般需要大量的鍵入。使用指令檔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。

mqobjmgr 指令檔的一般語法如下 (版本特性會反映指令檔版本，但不會反映 Message Queue 產品的版本 - 不是指令行選項 - 它的值必須設定成 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
...
```

作為如何使用指令檔的範例，請考量以下 imqobjmgr 指令：

```
imqobjmgr add
-t qf
-l "cn=myQCF"
-o "imqAddressList=mq://foo:777/jms"
-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.imqAddressList=mq://foo:777/jms
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 (imqobjmgr)：

```
imqobjmgr -i MyCmdFile
```

您還可以使用指令檔指定某些選項，同時使用指令行指定其他選項。這可讓您使用指令檔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 "imgAddressList=mq://foo:777/jms"
  -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=img
objstore.attrs.java.naming.security.principal=\
    uid=fooUser, ou=People, o=img
objstore.attrs.java.naming.security.credentials=fooPasswd
objstore.attrs.java.naming.security.authentication=simple
```

指令檔的其他範例可在以下位置找到：

```
/usr/demo/img/imgobjmgr (Solaris)
/opt/sun/mq/examples/imgobjmgr (Linux)
IMQ_HOME/demo/imgobjmgr (Windows)
```

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。

備註 物件管理員公用程式 (imgobjmgr) 僅列出並顯示 Message Queue 受管理物件。如果物件儲存應包含的非 Message Queue 物件與您想新增的受管理物件具有相同的查找名稱，那麼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 "imqAddressList=mq://myHost:7272/jms"
-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`。受管理物件儲存在 LDAP 伺服器中。您可以透過將指令檔指定爲 `imqobjmgr` 指令的引數來完成相同的作業。如需更多資訊，請參閱第 170 頁「使用指令檔」。

備註

命名慣例：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：`cn=myQCF`）。您可以使用 `-l` 選項指定查找名稱。如果您正在使用檔案系統物件儲存，那麼您不需使用 `cn` 字首，但請勿使用包含「`/`」的查找名稱。請參閱表 8-3。

表 8-3 命名慣例範例

物件儲存類型	正確名稱	錯誤名稱
LDAP 伺服器	<code>cn=myQCF</code>	<code>myQCF</code>
檔案系統	<code>myTopic</code>	<code>myObjects/myTopic</code>

新增主題佇列

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。

在將相應的受管理物件新增至物件儲存前，最好先建立實體目標。使用指令公用程式 (imgcmd) 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 113 頁「取得有關連線的資訊」。

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。

```
imgobjmgr add
-t t
-l "cn=myTopic"
-o "imgDestinationName=TestTopic"
-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"
```

這是相同的指令，只是受管理物件儲存在 Solaris 檔案系統中：

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

例如，在 LDAP 伺服器一例中，您可以使用指令檔 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
```

備註 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，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，此主題的查找名稱為 `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=img"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"

```

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

```

取得有關物件的資訊

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 cn=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 imgReconnectAttempts=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"
```

更新受管理物件

使用代理程式叢集

Message Queue Enterprise Edition 支援代理程式叢集的使用：代理程式的群組一起工作，為用戶端提供訊息遞送服務。叢集可使訊息伺服器透過分散用戶端的連線至數個代理程式，來調整其訊息流量的作業。如需叢集和其運作方式的一般說明，請參閱 Message Queue 技術概述。

本章描述如何管理代理程式叢集、將叢集與代理程式連線並進行配置。它包含下面幾節：

- [第 180 頁](#) 「叢集配置特性」
- [第 181 頁](#) 「管理叢集」
- [第 184 頁](#) 「主代理程式」

叢集配置特性

您可以為叢集的每個代理程式成員指定 *叢集配置特性* 以定義叢集。您可以為每個叢集中的代理程式個別地設定這些特性，但通常較方便的做法是將其集合至所有代理程式所參照的中央 *叢集配置檔案* 中。這樣可以防止違反合約的設定，並確保叢集中的所有代理程式都能共用相同且一致的配置資訊。

叢集配置特性詳述於第 300 頁表 14-11 中。這些部分包括：

- `imq.cluster.brokerlist` 會將主機名稱和連接埠號賦予所有屬於叢集的代理程式。
- `imq.cluster.masterbroker` 指定哪一個代理程式 (若有) 為記錄狀態變更的主代理程式。
- `imq.cluster.url` 指定叢集配置檔案 (若有) 的位置。
- `imq.cluster.hostname` 將主機名稱或 IP 位址賦予叢集連線服務，用於在叢集的各代理程式之間進行內部通訊。如果有多個主機，此設定才會有用：例如，如果一部電腦中有多個網路介面卡。
- `imq.cluster.port` 將連接埠號賦予叢集連線服務。
- `imq.cluster.transport` 指定叢集連線服務所使用的傳輸協定，例如 `tcp` 或 `ssl`。

您可以單獨設定個別代理程式的 `hostname` 和 `port` 特性，但是叢集中所有代理程式的 `brokerlist`、`masterbroker`、`url` 和 `transport` 值必須相同。

以下各節描述如何使用叢集配置檔案，以個別或集中的方式，設定每個叢集中代理程式的叢集配置特性。

設定個別代理程式的叢集特性

您可以在代理程式的實例配置檔案中 (或啟動代理程式時的指令行上) 設定叢集配置特性。例如，若要建立的叢集包含了 `host1` 連接埠 9876 上、`host2` 連接埠 5000 上和 `ctrlhost` 預設連接埠 (7676) 上的代理程式，三個代理程式的實例配置中都將包含以下的特性：

```
imq.cluster.brokerlist=host1:9876,host2:5000,ctrlhost
```

請注意，如果您需要變更叢集配置，則此方法要求更新叢集中所有代理程式的實例配置檔案。

使用叢集配置檔案

基於一致性以及便於維護的考量，建議您將所有的共用叢集配置特性集合至單一叢集配置檔案中，而不要為每個代理程式個別地進行設定。在此方法中，每個代理程式的實例配置檔案都必須設定 `imq.cluster.url` 特性指向叢集配置檔案的位置：例如，

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

叢集配置檔案接著會為叢集中的所有代理程式定義共用配置特性，例如要連線的代理程式清單 (`imq.cluster.brokerlist`)、用於傳輸協定的叢集連線服務 (`imq.cluster.transport`)，以及 (選擇性) 主代理程式的位址 (`imq.cluster.masterbroker`)。以下的程式碼定義與上一個範例中相同的叢集，以及在 `ctrlhost` 上執行，且作為主代理程式的代理程式：

```
imq.cluster.brokerlist=host1:9876,host2:5000,ctrlhost
imq.cluster.masterbroker=ctrlhost
```

管理叢集

本節說明如何連線至一組代理程式以形成一個叢集、將新的代理程式新增至現有叢集，以及從叢集中移除代理程式。

連線代理程式

有兩種通用方法可將代理程式連線到叢集：從指令行 (使用 `-cluster` 選項) 或在叢集配置檔案中設定 `imq.cluster.brokerlist` 特性。無論您使用哪種方法，您啟動的每個代理程式每 5 秒鐘會嘗試連線至其他代理程式；一旦啟動此叢集中的主代理程式 (如果已配置)，連線即可成功。如果叢集中有代理程式比主代理程式先啟動，則此代理程式將保持暫停狀態，從而拒絕用戶端連線，直到主代理程式啟動時；然後暫停的代理程式的全部功能將自動可用。

當您要啟動代理程式時，可以使用 `imqbrokerd` 指令的 `-cluster` 選項在完整的代理程式清單中指定，而不是使用叢集配置檔案。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 (7676) 上執行的代理程式、在 `host2` 的連接埠 5000 上執行的代理程式，以及在預設主機 (`localhost`) 的連接埠 9876 上執行的代理程式：

```
imqbrokerd -cluster host1,host2:5000,:9876
```

一個更適合用於生產系統的替代方法，就是建立一個使用 `imq.cluster.brokerlist` 特性的叢集配置檔案以指定要連線的代理程式清單。叢集中的每個代理程式均必須將自己的 `imq.cluster.url` 特性設定為指向此叢集配置檔案。

Linux 必要條件：設定 IP 位址

有一個將代理程式連線至 Linux 系統中叢集的特殊必要條件。一些 Linux 安裝程式會自動將 `localhost` 項目設定為網路迴路的 IP 位址 (127.0.0.1)。您必須設定系統的 IP 位址，這樣叢集中所有代理程式的位址才會正確。

請在參與叢集的所有 Linux 系統中檢查 `/etc/hosts` 檔案，完成一部分的叢集設定。如果系統使用的是靜態 IP 位址，請編輯 `/etc/hosts` 檔案，為 `localhost` 指定正確的位址。如果位址是以網域名稱服務 (DNS) 註冊，請編輯 `/etc/nsswitch.conf` 檔案以變更項目的順序，則系統可以在參考本機 `hosts` 檔案之前執行 DNS 查找。`/etc/nsswitch.conf` 檔案中的行應該如下所示讀取：

```
hosts: dns files
```

代理程式之間的完全連線

要求叢集中代理程式間的加密訊息傳送的安全性時，您必須配置叢集連線服務以使用基於 SSL 的傳輸協定。針對叢集中的每個代理程式，設定基於 SSL 的連線服務，如第 146 頁「[使用基於 SSL 的服務](#)」中所述。然後再從叢集配置檔案，或個別將每個代理程式的 `imq.cluster.transport` 特性設定為 `ssl`。

將代理程式新增至叢集

將新的代理程式新增至叢集的程序取決於叢集是否使用叢集配置檔案。

► 使用叢集配置檔案將新的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` 特性。
2. 向叢集中的每個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叢集配置，並確保此叢集中代理程式的所有永久性資訊均為最新的。

3. (選用) 在代理程式的 `config.properties` 檔案中設定 `imq.cluster.url` 特性的值，以指向叢集配置檔案。
4. 啟動新的代理程式。

如果您並未執行**步驟 3**，請使用 `imqbrokerd` 指令行上的 `-D` 選項以設定 `imq.cluster.url` 的值。

► 不使用叢集配置檔案將新的代理程式新增至叢集

您可以編輯 `config.properties` 檔案或使用 `imqbrokerd` 指令行上的 `-D` 選項來設定以下特性的值：

- `imq.cluster.brokerlist`
- `imq.cluster.masterbroker` (如有必要)
- `imq.cluster.transport` (如果您正在使用安全叢集連線服務)

從叢集中移除代理程式

您用來從叢集中移除代理程式的方法，取決於您原先是透過指令行或者是藉由中央叢集配置檔案來建立叢集。

使用指令行移除代理程式

如果您已使用指令行的 `imqbrokerd` 指令將代理程式連線至叢集，您必須停止各代理程式然後重新啟動，並在指令行上指定一組新的叢集成員。程序如下：

► 使用指令行從叢集中移除代理程式

1. 使用 `imqcmd` 指令停止叢集中的各代理程式。
2. 重新啟動將保留在叢集中的代理程式，使用 `imqbrokerd` 指令的 `-cluster` 選項，只指定那些剩餘的代理程式。

例如，假設您原本使用以下的指令來啟動 *A*、*B* 和 *C* 代理程式來建立包含這三者的叢集

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

若要從叢集中移除代理程式 *A*，請使用以下指令重新啟動代理程式 *B* 和 *C*

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

使用叢集配置檔案移除代理程式

如果您原本使用中央叢集配置檔案中的 `imq.cluster.brokerlist` 特性來指定叢集的成員代理程式以建立叢集，您就不需要為了移除其中一個代理程式而停止所有的代理程式。反之，您只要編輯配置檔案來排除要移除的代理程式，以強制剩餘的叢集成員重新載入叢集配置，並重新配置已排除的代理程式，這樣它就不再指向相同的叢集配置檔案。程序如下：

► 使用叢集配置檔案從叢集中移除代理程式

1. 編輯叢集配置檔案，從以 `imq.cluster.brokerlist` 特性指定的清單中移除已排除的代理程式。
2. 向叢集中每個剩餘的代理程式發出以下指令：

```
imqcmd reload cls
```

這會強制代理程式重新載入叢集配置。
3. 停止從叢集所移除的代理程式。
4. 編輯此代理程式的 `config.properties` 檔案，移除或指定其 `imq.cluster.url` 特性的不同值。

主代理程式

叢集可選擇性地擁有一個主代理程式，它可維護配置變更記錄以記錄叢集永久性狀態中的任何變更。無論在叢集配置檔案或個別代理程式的實例配置檔案中，主代理程式都是由 `imq.cluster.masterbroker` 配置特性所識別。

配置變更記錄包含了與叢集相關聯之永久性實體中的變更資訊，例如長期訂閱和管理員建立之實體目標。叢集中的所有代理程式會在啟動期間參考主代理程式，以更新這些永久性實體的資訊。關於主代理程式發生故障所導致的無法同步化，請參閱第 185 頁「當主代理程式無法使用時」以獲得更多資訊。

管理配置變更記錄

由於配置變更記錄包含了重要資訊，您必須定期進行備份，使其能夠在發生故障的情況下復原。雖然從備份中復原將會失去備份後叢集永久性狀態中的任何變更，但經常定期備份可儘量減低此潛在性的資訊遺失。備份與復原作業可以有效地壓縮和最佳化包含在配置變更記錄中的變更歷史，且配置變更記錄能夠隨著時間而迅速增長。

► 若要備份配置變更記錄

使用 `imqbrokerd` 指令的 `-backup` 選項來指定備份檔案的名稱。例如：

```
imqbrokerd -backup mybackuplog
```

► 復原配置變更記錄

1. 關閉叢集中的所有代理程式。

2. 使用該指令從備份檔案中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一個新的名稱或連接埠號，請相應地在叢集配置檔案中更新 `imq.cluster.brokerlist` 和 `imq.cluster.masterbroker` 特性。
4. 重新啟動叢集中的所有代理程式。

當主代理程式無法使用時

由於叢集中的所有代理程式都需要主代理程式來執行永久性作業，當主代理程式無法使用時，以下叢集中所有代理程式的 `imqcmd` 子指令都會傳回一個錯誤：

- `create dst`
- `destroy dst`
- `update dst`
- `destroy dur`

自動建立的實體目標和暫存目標不受影響。

在未配置主代理程式的環境中，任何嘗試建立或取消長期訂閱的用戶端應用程式都會收到一個錯誤。不過用戶端可以指定並與現有的長期訂閱進行互動。

主代理程式

監視訊息伺服器

本章描述可用來監視訊息伺服器的工具，以及如何取得度量資料。本章有下列幾節：

- [第 188 頁](#) 「監視工具簡介」
- [第 189 頁](#) 「配置及使用代理程式記錄功能」
- [第 194 頁](#) 「互動顯示度量」
- [第 199 頁](#) 「撰寫應用程式來監視代理程式」

有關特定度量的參照資訊，請參閱[第 18 章](#) 「度量參照」。

監視工具簡介

Message Queue 資訊有三個監視介面：記錄檔、互動式指令，和可以取得度量的用戶端 API。各有下列優點和缺點：

- 記錄檔提供長期的度量資料記錄，但是不易剖析。
- 指令可讓您快速取得符合您需求的資訊範例，但無法查看歷程資訊或有計劃的處理資料。
- 用戶端 API 可讓您擷取資訊、處理資訊、處理資料、呈現圖表或傳送警示。然而若要使用它，您必須撰寫自訂應用程式來擷取和分析資料。

表 10-1 比較不同的工具。

表 10-1 度量監視工具的優點與限制

標準資訊監視工具	優點	限制
imqcmd metrics	遠端監視 便於抽查 在指令選項中設定報告間隔時間；可即時變更 易於選取偏好的特定資料 以簡單表格格式呈現的資料	沒有取得所有資料的單一指令 難以有計劃分析資料 未建立歷程記錄 難以查看歷程趨勢
記錄檔	固定取樣 建立歷程記錄	需要配置代理程式特性；必須關閉並重新啟動代理程式才可生效 僅本地監視 資料格式難以讀取或剖析；沒有剖析工具 無法即時變更報告間隔時間；所有度量資料的報告間隔時間皆相同 不提供彈性選取資料 僅代理程式度量；不包括目標和連線服務度量 若間隔時間過短可能會影響效能

表 10-1 度量監視工具的優點與限制 (續)

標準資訊監視工具	優點	限制
用戶端 API	遠端監視 易於選取偏好的特定資料 可以有計劃的分析資料並以任何格式呈現	需要配置代理程式特性；必須關閉並重新啟動代理程式才可生效 您必須寫入您自己的度量監視用戶端 無法即時變更報告間隔時間；所有度量資料的報告間隔時間皆相同

除了表格中顯示的差異性之外，每個工具皆會收集代理程式產生的不同度量資訊子集。如需每個監視工具收集的度量資料的詳細資訊，請參閱第 323 頁第 18 章「度量參照」。

配置及使用代理程式記錄功能

Message Queue 記錄程式會取得代理程式程式碼產生的資訊、除錯程式和度量產生器，並將資訊寫入一些輸出通道：標準輸出 (主控台)、記錄檔及 Solaris™ 作業系統上的 syslog (系統記錄) 常駐程式程序。

您可以指定記錄程式收集的資訊類型，以及寫入每個輸出通道的類型。要特別注意的是，您可以指定您要寫入記錄檔的度量資訊。

本節描述代理程式的預設記錄配置，並介紹如何將記錄資訊重新導向至替代輸出通道、如何變更記錄檔自動重建條件，以及如何將度量資料傳送到記錄檔。

預設記錄配置

代理程式會自動配置將記錄輸出儲存到一組自動重建的記錄檔中。這些記錄檔位於以相關代理程式實例名稱辨別的目錄中 (請參閱附錄 A 「Message Queue 資料的作業系統特定位置」)：

```
.../instances/instanceName/log/
```

記錄檔為純文字檔案。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，記錄檔每週自動重建一次；系統保留九個備份檔案。

- 若要變更保留記錄檔的目錄，請將特性 `imq.log.file.dirpath` 設定為所需路徑。
- 若要將記錄檔的根名稱從 `log` 變更為其他名稱，請設定 `imq.log.file.filename` 特性。

代理程式支援三種記錄級別：`ERROR`、`WARNING`、`INFO`。表 10-2 解釋了每個級別的意義。

表 10-2 記錄級別

級別	說明
<code>ERROR</code>	表示可導致系統故障問題的訊息。
<code>WARNING</code> (警告)	應加以注意但不會導致系統故障的警示。
<code>INFO</code>	度量和其他資訊性訊息的報告。

設定記錄級別可收集此級別以上 (含此級別) 的訊息。預設級別是 `INFO`，因此依預設，`ERROR`、`WARNING` 和 `INFO` 訊息會全部記錄。

記錄訊息格式

記錄的訊息包含時間戳記、訊息程式碼和訊息本身。資訊容量因您所設定的記錄級別而有所不同。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
```

若要變更時區時間戳記，請參閱第 297 頁表 14-10 中描述的有關 `imq.log.timezone` 特性的資訊。

變更記錄程式配置

第 297 頁表 14-10 中描述了記錄的相關特性。

► 變更代理程式的記錄程式配置

1. 設定記錄級別。
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，請為此檔案配置自動重建條件。

您可以透過設定記錄程式特性完成這些步驟。您可以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` 檔案中的記錄程式特性。
- 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` 變更記錄程式特性 (或任何代理程式特性)。

在指令行上傳送的選項覆寫在代理程式實例配置檔案中指定的特性。表 10-3 列出了影響記錄的 `imqbrokerd` 選項。

表 10-3 `imqbrokerd` 記錄程式選項和對應的特性

imqbrokerd 選項	說明
<code>-metrics interval</code>	指定寫入記錄程式的度量資訊的間隔時間 (以秒為單位)。
<code>-loglevel level</code>	將記錄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 (記錄訊息的目標)
- 變更自動重建條件

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，也可記錄至記錄檔 (在 Solaris 上，錯誤訊息還可寫入至系統的 `syslog` 常駐程式)。

您可以用以下方法變更記錄訊息的輸出通道：

- 若要使所有記錄種類（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。
- 若要避免記錄輸出顯示在畫面上，請使用 `imqbrokerd` 指令的 `-silent` 選項。
- 使用 `imq.log.file.output` 特性指定應寫入至記錄檔的記錄資訊的種類。例如，
`imq.log.file.output=ERROR`
- 使用 `imq.log.console.output` 特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` 特性指定應寫入至 Solaris syslog 的記錄資訊的種類。例如，
`imq.log.syslog.output=NONE`

備註 變更記錄程式輸出通道前，您必須確定已將記錄設定為支援您要對映到輸出通道的資訊的級別。例如，如果您要將記錄級別設定為 `ERROR`，並將 `imq.log.console.output` 特性設定為 `WARNING`，則不會記錄任何訊息，因為您沒有啟用 `WARNING` 訊息的記錄。

變更記錄檔自動重建條件

有兩個條件用於自動重建記錄檔：時間和容量。預設為使用時間條件，每七天自動重建一次檔案。

- 若要變更時間間隔，您需要變更特性 `imq.log.file.rolloversecs`。例如，以下特性定義將時間間隔變更為十天：
`imq.log.file.rolloversecs=864000`
- 若要將自動重建條件變更為取決於檔案容量，您需要設定 `imq.log.file.rolloverbytes` 特性。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，自動重建檔案。
`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特性，則先達到的限制將啟動自動重建。如上所述，代理程式最多可保留九個自動重建檔案。

代理程式執行時，您可以設定或變更記錄檔自動重建特性。若要設定這些特性，請使用 `imqcmd update bkr` 指令。

將度量資料傳送到記錄檔

本節描述使用代理程式記錄檔報告度量資訊的程序。如需配置記錄程式的一般資訊，請參閱第 189 頁「配置及使用代理程式記錄功能」。

► 若要使用記錄檔報告度量資訊

1. 配置代理程式的度量產生功能：

- a. 確定 `imq.metrics.enabled=true`

依預設，會開啓用於記錄的度量產生功能。

- b. 設定度量產生的合適秒數間隔時間。

`imq.metrics.interval=interval`

您可以在啓動代理程式時，在 `config.properties` 檔案中或使用 `-metrics interval` 指令行選項設定此值。

2. 確定記錄程式已收集度量資訊：

`imq.log.level=INFO`

這是預設值。您可以在啓動代理程式時，在 `config.properties` 檔案中或使用 `-loglevel level` 指令行選項設定此值。

3. 確定已設定記錄程式，將度量資訊寫入記錄檔：

`imq.log.file.output=INFO`

這是預設值。您可以在 `config.properties` 檔案中進行設定。

4. 啓動代理程式。

下面是代理程式度量輸出到記錄檔的範例：

```
[21/Jul/2004:11:21:18 PDT]
連線：0    JVM 堆疊：8323072 位元組 (7226576 位元組可用) 執行緒：0 (14-1010)
  輸入：0 個訊息 (0 個位元組) 0 個封包 (0 個位元組)
  輸出：0 個訊息 (0 個位元組) 0 個封包 (0 個位元組)
輸入速率：0 個訊息 / 秒鐘 (0 個位元組 / 秒鐘) 0 個封包 / 秒鐘 (0 個位元組 / 秒鐘)
輸出速率：0 個訊息 / 秒鐘 (0 個位元組 / 秒鐘) 0 個封包 / 秒鐘 (0 個位元組 / 秒鐘)
```

如需度量資料的參照資訊，請參閱第 18 章「度量參照」。

記錄停用的訊息

您可以啟用代理程式的停用訊息記錄功能來監視實體目標。無論您有沒有使用此停用訊息佇列，都可以記錄這些停用的訊息。

啟用此停用訊息記錄功能後，代理程式會記錄下列類型的事件：

- 實體目標超出最大大小。
- 代理程式因為下列原因，移除實體目標的訊息：
 - 已達目標大小限制。
 - 訊息的時限已到。
 - 訊息太大。
 - 代理程式處理訊息時發生錯誤。

如果停用訊息佇列在使用中，還會記錄下列類型的事件：

- 代理程式將訊息移動到停用訊息佇列。
- 代理程式將停用訊息佇列中的訊息移除並且捨棄。

依預設，不記錄停用的訊息。若要記錄，請設定代理程式屬性 `imq.destination.logDeadMsgs`。

互動顯示度量

Message Queue 代理程式可以報告下列類型的度量：

- **Java 虛擬機器 (JVM) 度量**。有關 JVM 堆疊大小的資訊。
- **代理程式範圍度量**。有關儲存於代理程式中的訊息、輸入和輸出代理程式的訊息流量，以及記憶體用途的資訊。追蹤訊息的數量和位元組數。
- **連線服務度量**。有關連線和連線執行緒資源的資訊，以及有關特定連線服務之訊息流量的資訊。
- **目標度量**。有關流入和流出特定實體目標的訊息流量的資訊，以及有關實體目標用戶、記憶體和磁碟空間使用的資訊。

`imqcmd` 指令可以取得整個代理程式、個別連線服務和個別實體目標的度量資訊。若要取得度量資料，通常可使用 `imqcmd` 子指令 `metrics`。度量資料會依您指定的間隔時間或次數寫入主控台畫面。

您也可以使用 `query` 子指令，檢視也包含配置資訊的類似資料。請參閱第 198 頁「`imqcmd query`」，以獲得更多資訊。

imqcmd metrics

`imqcmd metrics` 的語法和選項個別顯示於表 10-4 和表 10-5 中。

表 10-4 `imqcmd metrics` 子指令語法

子指令語法	提供的度量資料
<pre>metrics bkr [-b <i>hostName:port</i>] [-m <i>metricType</i>] [-int <i>interval</i>] [-msp <i>numSamples</i>]</pre>	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度量。
或	
<pre>metrics svc -n <i>serviceName</i> [-b <i>hostName:port</i>] [-m <i>metricType</i>] [-int <i>interval</i>] [-msp <i>numSamples</i>]</pre>	顯示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。
或	
<pre>metrics dst -t <i>destType</i> -n <i>destName</i> [-b <i>hostName:port</i>] [-m <i>metricType</i>] [-int <i>interval</i>] [-msp <i>numSamples</i>]</pre>	顯示有關指定類型和名稱之實體目標的度量資訊。

表 10-5 `imqcmd metrics` 子指令選項

子指令選項	說明
<code>-b <i>hostName:port</i></code>	指定報告度量資訊之代理程式的主機名稱和連接埠。預設值為 <code>localhost:7676</code>
<code>-int <i>interval</i></code>	指定顯示度量的間隔時間 (以秒為單位)。預設為 5 秒鐘。

表 10-5 imqcmd metrics 子指令選項 (續)

子指令選項	說明
-m <i>metricType</i>	<p>指定要顯示的度量類型：</p> <p>ttl 顯示流入和流出代理程式、服務或目標的訊息與封包之度量 (預設度量類型)</p> <p>rts 顯示流入和流出代理程式、連線服務或目標的訊息和封包之流量速率的度量 (每秒)</p> <p>cm 顯示連線、虛擬記憶體堆疊和執行緒 (僅代理程式和連線服務)</p> <p>con 顯示用戶相關的度量 (僅目標)</p> <p>dsk 顯示磁碟使用度量 (僅目標)。</p>
-msp <i>numSamples</i>	指定顯示在輸出中的範例數目。預設的數目為沒有限制 (無限)。
-n <i>destName</i>	指定報告度量資料之實體目標的名稱 (如果有的話)。沒有預設值。
-n <i>serviceName</i>	指定報告度量的連線服務 (如果有的話)。沒有預設值。
-t <i>destTyp</i>	指定報告度量資料之實體目標 (如果有的話) 的類型 (佇列或主題)。沒有預設值。

使用顯示度量資料的 metrics 子指令

本節描述使用 metrics 子指令報告度量資訊的程序。

► 若要使用 metrics 子指令

1. 啟動需要度量資訊的代理程式。
請參閱第 64 頁「以互動方式啟動代理程式」。
2. 發出適當的 imqcmd metrics 指令和選項，其內容分別顯示於表 10-4 和表 10-5。

度量輸出：imqcmd metrics

本節包含 imqcmd metrics 子指令的輸出範例。這些範例顯示代理程式範圍、連線服務和實體目標度量。

代理程式範疇度量

若要取得訊息和封裝每隔 10 秒流入和流出代理程式的速率，請使用 `metrics bkr` 子指令：

```
imqcmd metrics bkr -m rts -int 10 -u admin
```

此指令會產生與以下相似的輸出 (請參閱第 324 頁表 18-2 中的資料說明)：

訊息 / 秒鐘		訊息位元組 / 秒鐘		Pkts / 秒鐘		Pkt 位元組 / 秒鐘	
輸入	輸出	輸入	輸出	輸入	輸出	輸入	輸出
0	0	27	56	0	0	38	66
10	0	7365	56	10	10	7457	1132
0	0	27	56	0	0	38	73
0	10	27	7402	10	20	1400	8459
0	0	27	56	0	0	38	73

連線服務度量

若要取得由 `jms` 連線服務處理之訊息和封包的累積總數，請使用 `metrics svc` 子指令：

```
imqcmd metrics svc -n jms -m ttl -u admin
```

此指令會產生與以下相似的輸出 (請參閱第 326 頁表 18-3 中的資料說明)：

訊息		訊息位元組		Pkts		Pkt 位元組	
輸入	輸出	輸入	輸出	輸入	輸出	輸入	輸出
164	100	120704	73600	282	383	135967	102127
657	100	483552	73600	775	876	498815	149948

實體目標度量

若要取得有關實體目標的度量資訊，請使用 `metrics dst` 子指令：

```
imqcmd metrics dst -t q -n XQueue -m ttl -u admin
```

此指令會產生與以下相似的輸出 (請參閱第 328 頁表 18-4 中的資料說明)：

訊息		訊息位元組		訊息計數		總訊息位元組 (k)			最大	
輸入	輸出	輸入	輸出	目前	尖峰	平均	目前	尖峰	平均	訊息 (k)
200	200	147200	147200	0	200	0	0	143	71	0
300	200	220800	147200	100	200	10	71	143	64	0
300	300	220800	220800	0	200	0	0	143	59	0

若要取得實體目標的用戶相關資訊，請使用以下 `metrics dst` 子指令：

```
imqcmd metrics dst -t q -n SimpleQueue -m con -u admin
```

此指令會產生與以下相似的輸出 (請參閱第 328 頁表 18-4 中的資料說明)：

使用中用戶			備份用戶			訊息計數		
目前	尖峰	平均	目前	尖峰	平均	目前	尖峰	平均
1	1	0	0	0	0	944	1000	525

imqcmd query

`imqcmd query` 的語法和選項顯示於表 10-6 中，並隨附該指令提供的度量資料說明。

表 10-6 `imqcmd query` 子指令語法

子指令語法	提供的度量資料
<code>query bkr</code> [-b <i>hostName:port</i>]	目前儲存於代理程式記憶體與永久性儲存之訊息數目和訊息位元組的資訊 (請參閱第 104 頁「顯示代理程式資訊」)
或	
<code>query svc -n <i>serviceName</i></code> [-b <i>hostName:port</i>]	目前已配置執行緒的數目和指定連線服務之連線數目的資訊 (請參閱第 110 頁「顯示連線服務資訊」)
或	

表 10-6 imqcmd query 子指令語法

子指令語法	提供的度量資料
<pre>query dst -t destType -n destName [-b hostName:port]</pre>	目前產生者、使用中和備份用戶，與用於指定目標中記憶體和永久性儲存中儲存的訊息和訊息位元組的資訊 (請參閱第 123 頁「顯示實體目標資訊」)
備註	因為 imqcmd query 提供有限的度量資料，所以未在第 323 頁第 18 章「度量參照」的表格中列出此工具。

撰寫應用程式來監視代理程式

Message Queue 提供度量監視功能，此功能可讓代理程式將度量資料寫入 JMS 訊息，接著並根據包含於訊息中的度量資訊類型，將其傳送到其中一個度量主題目標。

您可以藉由寫入訂閱度量主題目標、使用目標中的訊息並處理訊息中度量資訊的用戶端應用程式，來存取此度量資訊。

總共有 5 個度量主題目標。表 10-7 中顯示了這些目標的名稱，和傳送到每個目標的度量訊息類型。

表 10-7 度量主題目標

主題名稱	度量訊息的類型
mq.metrics.broker	代理程式度量
mq.metrics.jvm	Java 虛擬機器度量
mq.metrics.destination_list	目標及其類型的清單
mq.metrics.destination.queue. <i>monitoredDestinationName</i>	用於指定名稱之佇列的目標度量
mq.metrics.destination.topic. <i>monitoredDestinationName</i>	用於指定名稱之主題的目標度量

設定基於訊息的監視

本節描述使用基於訊息監視功能，收集度量資訊的程序。程序包括用戶端部署和管理工作。

► 若要設定基於訊息的監視

1. 寫入度量監視用戶端。

請參閱 *Message Queue Developer's Guide for Java Clients*，以取得設計用戶端訂閱度量主題目標、使用度量訊息並從這些訊息擷取度量資料的說明。

2. 藉由設定 `config.properties` 檔案中的代理程式特性值，配置代理程式的度量訊息產生者：

- a. 啟用度量訊息產生。

設定 `imq.metrics.topic.enabled=true`

預設為 `true`。

- b. 設定產生度量訊息的間隔時間 (以秒為單位)。

設定 `imq.metrics.topic.interval=interval`

預設為 60 秒鐘。

- c. 指定度量訊息是否為永久性訊息 (即它們是否為在代理程式發生故障後倖存)。

設定 `imq.metrics.topic.persist`

預設為 `false`。

- d. 指定度量訊息刪除前，保留在其個別目標中的時間長短。

設定 `imq.metrics.topic.timetolive`

預設值為 300 秒。

3. 設定任何您要在度量主題目標上的存取控制。

請參閱以下「[安全性與存取注意事項](#)」中的說明。

4. 啟動度量監視用戶端。

用戶訂閱度量主題時，即會自動建立度量主題目標。一旦建立度量主題，代理程式度量訊息產生者會開始傳送度量訊息到度量主題。

安全性與存取注意事項

有兩個限制度量主題目標存取的原因：

- 度量資料可能包括有關代理程式或其資源的關聯資訊
- 過量的度量主題目標訂閱可能會增加代理程式耗用時間，並對效能產生不良影響

由於這些考量，所以建議您限制度量主題目標的存取。

監視用戶端必須遵守與其他用戶端相同的認證和授權控制。只有 Message Queue 使用者儲存庫中的使用者可以連線到代理程式。

您可以藉由存取控制特性檔限制存取特定度量主題目標，藉以提供額外保護，如第 140 頁「[授權使用者：存取控制特性檔案](#)」中所述。

例如，以下 `accesscontrol.properties` 檔案中的項目會拒絕除了 `user1` 和 `user2` 以外的其他任何人存取 `mq.metrics.broker` 度量主題。

```
topic.mq.metrics.broker.consume.deny.user=*
topic.mq.metrics.broker.consume.allow.user=user1,user2
```

以下項目僅允許使用 `user3` 監視主題 `t1`。

```
topic.mq.metrics.destination.topic.t1.consume.deny.user=*
topic.mq.metrics.destination.topic.t1.consume.allow.user=user3
```

根據度量資料關聯，您也可以使用加密連線，將度量監視用戶端連線到代理程式。如需使用加密連線的資訊，請參閱第 146 頁「[使用基於 SSL 的服務](#)」。

度量輸出：度量訊息

您使用基於訊息監視的 API 取得的度量資料輸出，為您寫入度量監視用戶端的函數。您僅受限於代理程式中度量產生器提供的資料。如需此資料的完整清單，請參閱第 323 頁「[度量參照](#)」。

撰寫應辦程式來監視代理程式

分析與調校訊息傳送

本章涵蓋一些如何分析與調校 Message Queue 服務的主題，以最佳化您訊息傳送應用程式的效能。它包括以下主題：

- [第 204 頁](#) 「關於效能」
- [第 206 頁](#) 「影響效能的因素」
- [第 219 頁](#) 「調整配置以改善效能」

關於效能

本節提供效能調校的一些背景資訊。

效能調校程序

訊息傳送應用程式的效能取決於應用程式和 **Message Queue** 服務間的互動。因此，最佳化效能需要應用程式開發者和管理員共同的努力。

最佳化效能的程序以應用程式設計為起點，接著在部署應用程式後，調校訊息服務繼續。效能調校程序包括以下階段：

- 定義應用程式的效能需求
- 設計考慮影響效能因素的應用程式（特別是可靠性與效能間的平衡）
- 建立基本效能測量
- 調校或重新配置訊息服務以最佳化效能。

以上略述的程序通常是反覆的。應用程式部署期間，**Message Queue** 管理員會評估應用程式一般效能需求的訊息伺服器合適性。如果效能評定測試符合這些需求，管理員可以根據本章說明調校系統。但是，如果效能評定測試不符合效能需求，則可能需要重新設計應用程式或修改架構部署。

效能類型

一般而言，效能為訊息服務從產生者傳送訊息到用戶之速度和效率的測量。但是，依您的需要，您可能需要注意幾個不同方面的效能。

連線負載 可支援訊息產生者、訊息用戶或運作之連線系統的數目。

訊息流量 每秒導入訊息傳送系統的訊息數目或訊息位元組。

延時 從訊息產生者傳送特定訊息到訊息用戶所花的時間。

穩定性 訊息服務的整體可用性，或如何逐漸降低可用性以防大量負載或失敗。

效率 訊息傳送的效率；與計算已使用訊息相關的訊息流量測量。

通常這些不同方面的效能是相互關聯的。如果訊息流量高，則表示訊息較不可能儲存於訊息伺服器中，因此，延時時間短（單一訊息可迅速傳送）。但是，延時可取決於很多因素：通訊連結的速度、訊息伺服器處理速度和用戶端處理速度等。

無論如何，會有數個不同方面的效能。對您最重要的方面通常取決於特定應用程式的需求。

效能評定

效能評定為建立用於您訊息傳送應用程式與測量訊息流量之測試套件的程序，或用於此測試套件的其他方面效能。

例如，您可能會以數個生產型用戶端，使用數個連線、階段作業和訊息產生者來建立測試套件，並以某特定速率，將標準大小的永久性或非永久性訊息傳送到數個佇列或主題（所有皆取決於您訊息傳送應用程式設計）。同樣地，測試套件包括數個使用數個連線和階段作業的使用用戶端以及在測試套件實體目標中，以特定確認模式使用訊息的訊息用戶（特定類型）。

使用標準的測試套件，您可以測量訊息產生和使用間所需的時間或平均訊息流量速率，並且可以監視系統以觀察連線執行緒用法、訊息儲存資料、訊息流量資料和其他相關度量。接著，您可以在對效能產生不良影響前，急速增加訊息產生的速率、訊息產生者的數目或其他變數。您可以達到的最大流量為您訊息服務配置的效能評定。

使用此效能評定，您可以修改部分您測試套件的特徵。藉由小心控制所有可能影響效能的因素（請參閱第 208 頁「影響效能的應用程式設計因素」），您可以注意到變更部分因素會如何影響效能評定。例如，您可以將連線數目或訊息大小增加五倍或十倍，即可發現效能受到影響。

相反地，您可以將基於應用程式的因素維持一致，並以某種控制方法變更您的代理程式配置（例如，變更連線特性、執行緒池特性、JVM 記憶體限制、限制運作方式、內建與外掛掛續性等），接著您可以注意到這些變更如何影響效能。

您應用程式的效能評定會提供資訊，這些資訊在您要藉由調校訊息服務，以增加已部署應用程式的效能時會有所幫助。效能評定可以更加準確地預測一個變更或一組變更的效果。

一般來說，應在受控制的測試環境中，於一段足以穩定您訊息服務的長時間內執行效能評定。（啟動時，Just-In-Time 編譯會將 Java 程式碼轉為機器程式碼，這會對效能造成不良影響。）

基本使用式樣

一旦部署並執行訊息傳送應用程式，建立基本使用式樣是很重要的。您必須知道尖峰需求何時出現並可以將需求數目化。例如，需求通常會因一般使用者的數目、活動級別、天數或前述所有因素而有變化。

若要建立基本使用式樣，您必須長時間監視您的訊息伺服器，查看類似下面的資料：

- 連線數目
- 儲存於代理程式中 (或特定實體目標中) 的訊息數目
- 流入或和流出代理程式 (或特定實體目標中) 的訊息流量
- 使用中用戶數目

您還可以使用度量資料中提供的平均與尖峰值。

檢查這些出乎設計意料之外的基本度量是非常重要的。經由此動作，您會檢查用戶端程式碼是否正常運作：例如，連線未處於開啓狀態或使用的訊息未處於未確認狀態。這些程式碼錯誤會使用訊息伺服器資源，且可能會明顯影響效能。

基本使用式樣可幫助您判斷如何將系統調校到最佳化效能。例如：

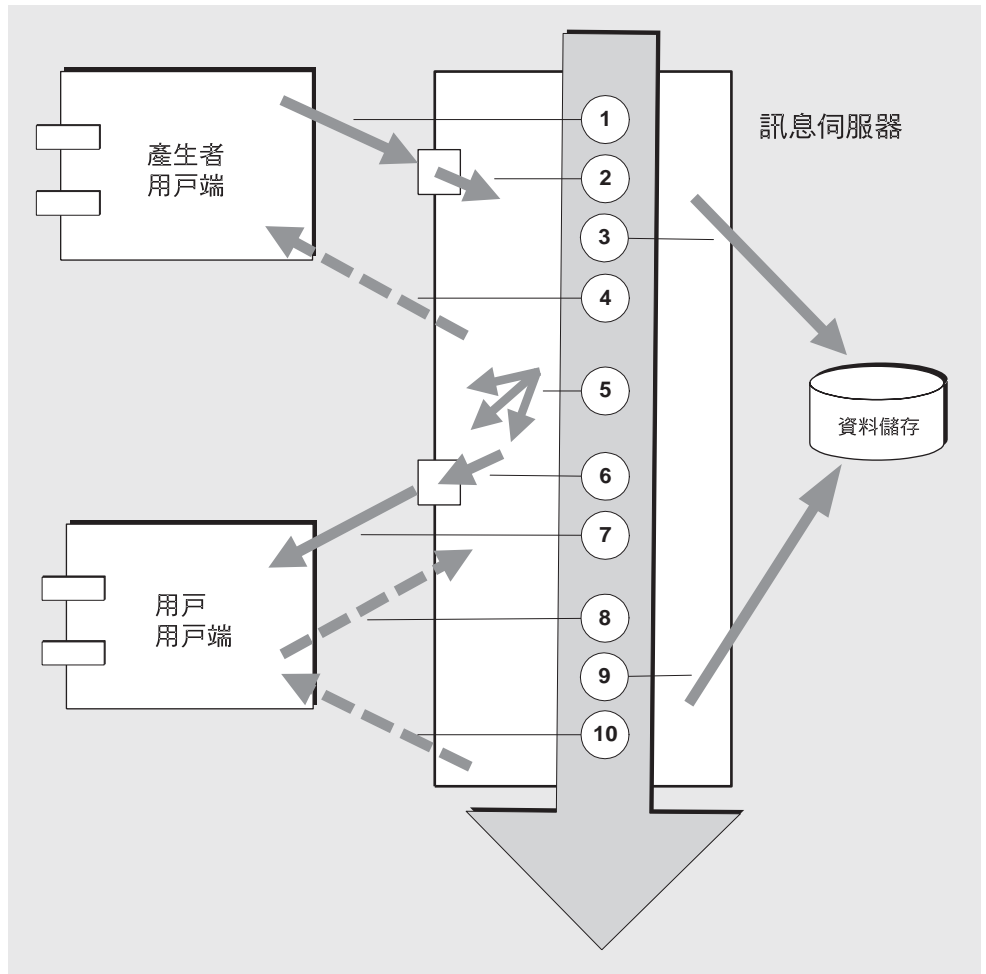
- 如果某個實體目標的使用度較其他實體目標高，那麼您可能要將此實體目標的訊息記憶體限制設定的比其他實體目標的限制高，或者要相應地調整限制運作方式。
- 如果所需的連線數目明顯大於最大執行緒池容量允許的大小，那麼您可能要增加執行緒池大小，或調整共用執行緒模型。
- 如果尖峰訊息流量遠遠大於平均流量，那麼在記憶體不足時，可能會影響您使用的限制運作方式。

一般而言，您對使用式樣的瞭解越多，您調校系統用於未來需求的式樣與計劃的能力越好。

影響效能的因素

訊息延時和訊息流量為兩個主要的效能指示器，它們的值通常取決於一般訊息完成訊息傳送程序中各個步驟所花費的時間。以下顯示的步驟用於永久性、可靠且已傳送之訊息的情況下。下圖描述這些步驟。

圖 11-1 透過 Message Queue 服務的訊息傳送



1. 這些訊息是由生產型用戶端傳送到訊息伺服器。
2. 訊息伺服器在訊息中進行讀取
3. 訊息置於永久性儲存體中，以確保其可靠性
4. 訊息伺服器會確認訊息的接收，以確保其可靠性
5. 訊息伺服器會判斷訊息的路由
6. 訊息伺服器會在訊息外等候

7. 這些訊息是由訊息伺服器傳送至使用用戶端
8. 使用用戶端會確認訊息的接收，以確保其可靠性
9. 訊息伺服器會處理用戶端確認，以確保其可靠性
10. 訊息伺服器會確認是否已處理用戶端確認

由於這些步驟是連續性的，所以在從生產型用戶端傳送訊息至使用用戶端時，這其中任何步驟都是可能遇到的瓶頸。大部分的步驟取決於訊息傳送系統的實體特徵：網路頻寬、電腦處理速度、訊息伺服器架構等。但是，還有些步驟是取決於訊息傳送應用程式的特徵和要求的可靠性級別。

以下小節說明應用程式設計因素與訊息傳送系統因素對效能的影響。當在訊息傳送中應用程式設計和訊息傳送系統因素相互操作時，將單獨的考慮每個種類。

影響效能的應用程式設計因素

應用程式設計決策會對整體訊息傳送效能產生顯著的效果。

影響效能最重要的因素為影響訊息傳送可靠性的因素。這些因素如下：

- 傳送模式 (永久性 / 非永久性訊息)
- 作業事件使用
- 確認模式
- 長期與非長期訂閱

其他影響效能的應用程式設計因素如下：

- 選擇器的使用 (訊息篩選)
- 訊息容量
- 訊息內文類型

以下各節描述影響訊息傳送效能的各個因素。一般來說，要在效能與可靠性間取得平衡：即提昇可靠性的因素往往會降低效能。

表 11-1 顯示各種應用程式設計因素通常如何影響訊息傳送效能。表格顯示兩個方案 - 高可靠性和低效能方案、高效能和低可靠性方案 - 且顯示特定應用程式設計因素的選擇。在這兩個方案中，有許多影響可靠性和效能的選擇和平衡。

表 11-1 高可靠性和高效能方案的比較

應用程式設計因素	高可靠性 低效能方案	高效能 低可靠性方案
傳送模式	永久性訊息	非永久性訊息
作業事件使用	作業事件的階段作業	無作業事件
確認模式	AUTO_ACKNOWLEDGE CLIENT_ACKNOWLEDGE	或 DUPS_OK_ACKNOWLEDGE
長期 / 非長期訂閱	長期訂閱	非長期訂閱
選擇器的使用	訊息篩選	無訊息篩選
訊息容量	大量的小訊息	少量的大訊息
訊息內文類型	複雜的內文類型	簡單的內文類型

備註 以下各圖為使用基於檔案的持續性，在雙 CPU、1002 Mhz、Solaris 8 系統上產生的效能資料。效能測試首先會預熱 Message Queue 代理程式，以便 Just-In-Time 編譯器最佳化系統和預備的永久性資料庫。

預熱代理程式後，會建立一個產生者和用戶，並以 30 秒的時間產生訊息。會記錄用戶接收所有產生訊息的所需時間，且計算流量速率（每秒幾個訊息）。會在應用程式設計因素的不同組合，重複使用此方案，如表 11-1 所示。

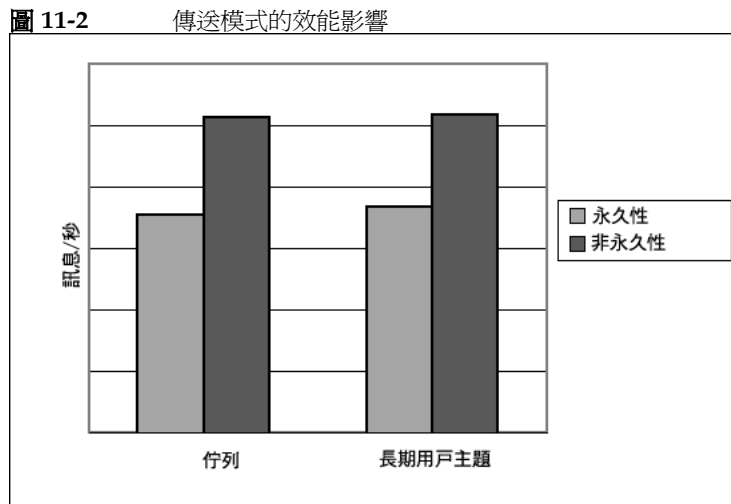
傳送模式（永久性 / 非永久性訊息）

永久性訊息可確保訊息伺服器故障情況下的訊息傳送。代理程式會在所有預定用戶確認他們已使用訊息前，將訊息儲存在永久性儲存體中。

永久性訊息的代理程式處理速度會比非永久性的訊息處理速度慢，原因如下：

- 代理程式必須可靠地儲存永久性訊息，如此一來，代理程式故障時才不會遺失永久性訊息。
- 代理程式必須確認每個永久性訊息的接收。一旦產生訊息的方法未傳回異常，則需確保至代理程式的傳送。
- 根據用戶端的確認模式，代理程式可能需要確認使用用戶端對永久性訊息的確認。

永久性和非永久性模式間效能的差異性非常明顯。圖 11-2 比較兩個可靠傳送情況下，永久性和非永久性訊息的流量：傳送 10K 大小的訊息到佇列和包含長期訂閱的主題。這兩種情況皆會使用 AUTO_ACKNOWLEDGE 確認模式。



作業事件使用

作業事件是所有作業事件階段作業中產生訊息和使用訊息的保證，兩種訊息的處理方法如下：處理作為一個單位、不處理（回轉）作為一個單位。

Message Queue 支援本機作業事件和分散式作業事件。

作業事件階段作業中產生或確認訊息的速度會比在非作業事件階段作業中的速度慢，原因如下：

- 附加資訊必須隨著每個產生訊息一起儲存。
- 在一些情況下，會儲存作業事件中的訊息，但通常是不會儲存這些訊息（例如，通常會刪除傳送到未訂閱之主題目標的永久性訊息，但是在作業事件開始時，會無法使用有關訂閱的資訊）。
- 作業事件中的訊息使用和確認資訊，必須在作業事件確定時進行儲存和處理。

確認模式

JMS 訊息傳送的確保可靠性機制，是用戶端用來確認 Message Queue 訊息伺服器所傳送到其自身的訊息使用。

如果在用戶端未確認訊息時關閉階段作業，或在處理確認前訊息伺服器發生故障，則代理程式會重新傳送訊息，並設定 `JMSRedelivered` 旗標。

對非作業事件階段作業而言，用戶端可以選擇三種確認模式中的其中一種模式，每一個模式皆擁有各自的效能特徵：

- `AUTO_ACKNOWLEDGE`。一旦用戶處理訊息後，系統會自動確認訊息。此模式可確保提供者發生故障後最多重新傳送一次訊息。
- `CLIENT_ACKNOWLEDGE`。此應用程式會控制確認訊息的端點。確認先前的確認後，所有訊息會在該階段作業中進行處理。如果在處理確認組時，訊息伺服器發生故障，則可能會重新傳送群組中一個或多個訊息。
- `DUPS_OK_ACKNOWLEDGE`。此模式引導系統在惰性狀態下確認訊息。提供者發生故障後，可以重新傳送多個訊息。

(使用 `CLIENT_ACKNOWLEDGE` 模式與使用作業事件相似，但在處理期間提供者發生故障時，前者不確保會一起處理所有確認。)

確認模式影響效能的原因如下：

- 在 `AUTO_ACKNOWLEDGE` 和 `CLIENT_ACKNOWLEDGE` 模式中，會要求代理程式與用戶端間的額外控制。額外控制訊息會新增額外處理耗用時間，且可干擾 JMS 有效負載訊息，因而造成處理延遲。
- 在 `AUTO_ACKNOWLEDGE` 和 `CLIENT_ACKNOWLEDGE` 模式中，用戶端必須在可使用其他訊息前，等待代理程式確認是否已處理用戶端的確認。(此用戶端確認可確保代理程式不會不慎重新傳送這些訊息。)
- 必須以用戶接收之所有永久性訊息的確認資訊來更新 `Message Queue` 永久性儲存，因此會降低效能。

長期與非長期訂閱

主題目標的用戶分為兩個種類，長期或非長期訂閱。

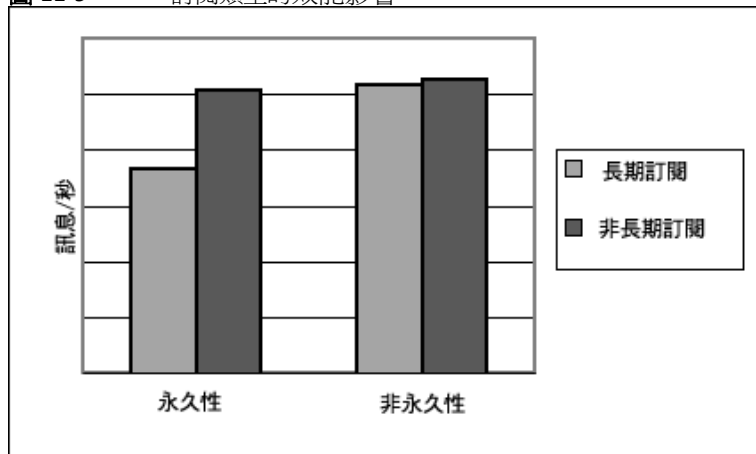
長期訂閱可提高可靠性，但會降低流量，原因如下：

- `Message Queue` 訊息伺服器必須持續儲存指定到每個長期訂閱的訊息清單，如此一來，一旦訊息伺服器發生故障，即可在回復後使用此清單。
- 會持續儲存長期訂閱的永久性訊息，如此一來，一旦訊息伺服器發生故障，即可在回復後重新傳送訊息，同時對應的用戶會變為使用中狀態。相反地，不會持續儲存非長期訂閱的永久性訊息 (一旦訊息伺服器發生故障，會中斷對應用戶的連線，且不會重新傳送訊息)。

圖 11-3 比較包含長期與非長期訂閱兩種情況的主題目標流量：永久性與非永久性的 10K 大小訊息。這兩種情況皆會使用 `AUTO_ACKNOWLEDGE` 確認模式。

您可以從圖 11-3 發現只有在永久性訊息的情況下，使用長期訂閱才會對效能產生明顯影響；如同上述說明，在此情況下發生影響是因為只會持續儲存用於長期訂閱的永久性訊息。

圖 11-3 訂閱類型的效能影響



選擇器的使用 (訊息篩選)

應用程式開發者通常會將訊息組指向特定用戶。他們會藉由將每個訊息組指向唯一實體目標，或使用單一實體目標並為用戶註冊一個或多個選擇器，以達成此目的。

選擇器為一字串，請求僅將包含符合此字串之特性值的訊息傳送到特定用戶。例如，選擇器 `NumberOfOrders >1` 僅會傳送包含 `NumberOfOrders` 特性值 2 或更高值的訊息。

以選擇器註冊用戶會降低效能 (與使用多個實體目標相比較的話)，因為需要額外處理每個訊息。使用選擇器時必須剖析選擇器，如此一來，選擇器可與之後的訊息相符合。此外，在路由每個訊息時，必須擷取每個訊息的訊息特性並與選擇器進行比較。但是，使用選擇器可在訊息傳送應用程式中提供更多彈性。

訊息容量

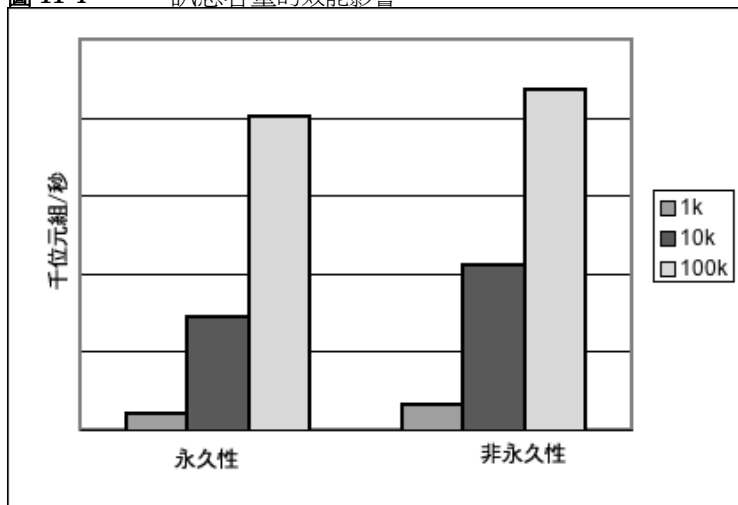
訊息容量會影響效能，因為必須將更多資料從生產型用戶端傳送到代理程式，再從代理程式傳送到使用用戶端，且必須儲存較大的永久性訊息。

但是，藉由將較小訊息批次到單一訊息，可以最小化個別訊息的路由和處理，以提供整體效能改善比率。在此情況下，會遺失個別訊息狀態的相關資訊。

圖 11-4 比較兩種情況下，1K、10K 和 100K 大小訊息每秒的流量 (以千位元組為單位)：永久性與非永久性的訊息。在所有情況下，皆會將訊息傳送到佇列目標，並使用 AUTO_ACKNOWLEDGE 確認模式。

圖 11-4 顯示在兩種情況下，傳送較大訊息的耗用時間會比傳送較小訊息的耗用時間短。您也可以發現在 1K 與 10K 大小訊息中，非永久性訊息的效能改善比率比永久性訊息的效能改善比率高出將近 50%，但在 100K 大小訊息中並沒有這種情形，這可能是因為在此情況下，網路頻寬成為訊息流量的瓶頸。

圖 11-4 訊息容量的效能影響



訊息的類型

JMS 支援五種訊息內文類型，以下大致上依其複雜性順序顯示：

- `BytesMessage`：包含應用程式判斷的位元組格式
- `TextMessage`：為簡單 Java 語言字串
- `StreamMessage`：包含 Java 原始值的串流
- `MapMessage`：包含名稱與值對的組合
- `ObjectMessage`：包含 Java 序列化物件

一般而言，依應用程式需求規定訊息類型時，更複雜的類型 (`MapMessage` 和 `ObjectMessage`) 會增加效能耗用。串列化和解串列化資料的效能。效能耗用取決於資料簡單或複雜的程度。

影響效能的訊息服務因素

訊息傳送應用程式的效能不只受到應用程式設計的影響，還受到執行訊息路由和傳送的訊息服務影響。

以下各節介紹可影響效能的這種訊息服務因素。瞭解這些因素的影響是調整訊息服務大小及診斷和解決已部署應用程式中可能會出現之效能瓶頸的關鍵。

影響 Message Queue 服務中效能的最重要因素如下所述：

- [硬體](#)
- [作業系統](#)
- [Java 虛擬機器 \(JVM\)](#)
- [連線](#)
- [代理程式限制和運作方式](#)
- [訊息伺服器架構](#)
- [資料儲存效能](#)
- [用戶端執行階段配置](#)

以下各節描述影響訊息傳送效能的各個因素。

硬體

對 Message Queue 訊息伺服器和用戶端應用程式而言，CPU 處理速度和可用記憶體，是決定訊息服務效能的主要因素。增加處理能力可以減少許多軟體限制，而增加記憶體可以同時增加處理速度和能力。然而，升級硬體以克服這些瓶頸通常需要一筆花費。

作業系統

因為不同作業系統的效率，效能會有所變化，即使是在同樣的硬體平台亦有差別。例如，作業系統部署的執行緒模型，對訊息伺服器可支援的運作連線數目會有極大影響。一般而言，在所有硬體皆同等的情況下，Solaris 系統通常會比 Linux 系統快，而 Linux 系統又比 Windows 系統快。

Java 虛擬機器 (JVM)

訊息伺服器為在主要 JVM 中執行且支援的 Java 程序。因此，JVM 處理是決定訊息伺服器路由和傳送訊息之速度和效率的重要因素。

要特別注意的是，JVM 的記憶體資源管理是非常重要的。必須在 JVM 配置足夠的記憶體，以適應增加的記憶體負載。除此之外，JVM 會定期回收未使用的記憶體，且記憶體回收會延遲訊息處理。JVM 記憶體堆疊的容量越多，記憶體回收過程中可能發生的延遲時間越長。

連線

用戶端和代理程式間的連線數目和速度，會影響訊息伺服器處理訊息的數目和傳送訊息的速度。

訊息伺服器連線限制

所有訊息伺服器的存取皆是透過連線。運作中連線數目的任何限制，皆會影響目前使用訊息伺服器之生產或使用用戶端的數目。

至訊息伺服器的連線數目通常會受到可用執行緒數目的限制。Message Queue 會使用執行緒池管理程式，以便配置支援專用的執行緒模型或共用的執行緒模型（請參閱第 75 頁「執行緒池管理程式」）。

專用的執行緒模型執行速度非常快，因為每個連線均有專用執行緒；但是，連線數目受到可用執行緒數目的限制（每個連線使用一個輸入執行緒和一個輸出執行緒）。共用執行緒模型在連線數目上沒有限制；但是，在連線數目超過共用執行緒時，會產生明顯的耗用時間和流量延遲，特別是在連線忙碌時。

傳輸協定

Message Queue 軟體可讓用戶端使用各種低層傳輸協定，與訊息伺服器進行通訊。Message Queue 支援第 73 頁「連線服務」中敘述的連線服務（和對應的協定）。

協定的選擇是根據應用程式需求而異（加密、透過防火牆的存取權），而選擇會影響整體效能。

圖 11-5 傳輸協定速度



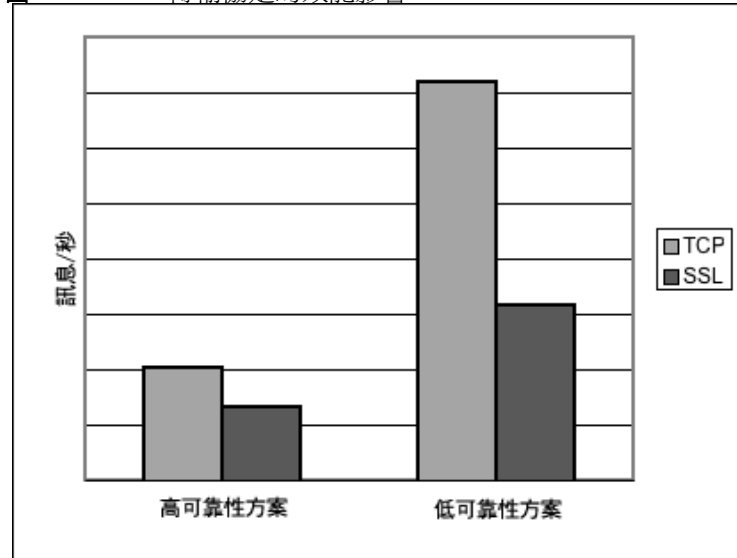
圖 11-5 反映各種協定技術的效能特徵：

- TCP 提供與代理程式進行通訊的最快方法。
- 傳送和接收性訊息時，SSL 的速率為 TCP 的 50% 到 70% (傳送和接收永久性訊息時為 TCP 速率的 50%，非永久訊息時則將近 70%)。此外，使用 SSL 建立初始連線速度較慢 (可能會花費數秒)，因為用戶端和代理程式 (或 HTTPS 情況下的 Web Server) 需要建立加密傳輸資料時會用到的私密密鑰。效能會因額外處理所需加密和解密每個低層 TCP 封包而降低。

圖 11-6 比較 TCP 和 SSL 兩種情況下的流量：高可靠性方案 (傳送到包含長期訂閱之主題目標的 1K 永久性訊息，並使用 AUTO_ACKNOWLEDGE 確認模式) 及高效能方案 (傳送到不包含長期訂閱之主題目標的非永久性訊息，並使用 DUPS_OK_ACKNOWLEDGE 確認模式)。

圖 11-6 顯示高可靠性情況下較不造成影響的協定。這可能是因為在高可靠性情況中需要的持續性耗用時間，是限制流量的重要因素，其影響比對協定速度的影響大。

圖 11-6 傳輸協定的效能影響



- HTTP 的速度比 TCP 與 SSL 速度慢。HTTP 將 Web Server 上的 Servlet 使用作為用戶端與代理程式間的代理。效能耗用時間受到 HTTP 請求中封裝封包，與訊息通過兩個躍點（用戶端至 Servlet、Servlet 至代理程式）以到達代理程式之需求的影響。
- HTTPS 速度較 HTTP 慢，因為 HTTPS 需要花費在用戶端與 Servlet 間和 Servlet 與代理程式間加密封裝的額外耗用時間。

訊息伺服器架構

可以將 Message Queue 訊息伺服器實作為單一代理程式或多重互連代理程式實例 - 代理程式叢集。

隨著連線至代理程式的用戶端數目的增加，以及要傳送的訊息數目的增加，代理程式將最終超過資源限制（如檔案描述元、執行緒和記憶體限制）。提供增加載入的一個方法為將多個代理程式實例新增至 Message Queue 訊息伺服器，從而在多重代理程式間分布用戶端連線與訊息路由和傳送。

一般而言，此調整會使運作達到最佳狀態，如果用戶端（尤其是訊息生產型用戶端）在叢集上平均分散的話。由於耗用時間受到叢集中代理程式間訊息傳送的影响，所以包含連線數目限制或訊息傳送速率限制的叢集，所表現效能可能會比單一代理程式低。

您還可以使用代理程式叢集來最佳化網路頻寬。例如，您可以在叢集中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實例。

如需更多叢集的相關資訊，請參閱第 9 章「使用代理程式叢集」。

代理程式限制和運作方式

訊息伺服器可能處理的訊息流量是訊息伺服器所支援之訊息傳送應用程式使用式樣的函數。然而，訊息伺服器在資源上受到限制：記憶體、CPU 週期等。因此，訊息伺服器可能會對點過度負載，造成沒有回應或不穩定的情況。

Message Queue 訊息伺服器有管理記憶體資源和預防代理程式耗盡記憶體的內建機制。這些機制的可配置限制包括代理程式可保留的訊息數目和訊息容量、其個別實體目標，以及達到實體目標限制時可採取的運作方式。

藉由謹慎監視與調校，這些可配置機制可用來平衡流入和流出訊息，如此一來，將不會發生系統過度負載情況。這些機制會耗用時間並限制訊息流量，但他們會維護作業完整性。

資料儲存效能

Message Queue 支援內建和外掛持續性。內建持續性是基於檔案的資料儲存。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。

內建持續性速度明顯比外掛持續性快；但是，JDBC 相容的資料庫系統可能會提供應用程式需要的容錯、安全與管理功能。

在使用內建持續性的情況下，您可以指定持續性作業同步化包含資料儲存的內部記憶體狀態，以最大化可靠性。這可以消除因系統當機而產生的資料遺失，但會影響效能。

用戶端執行階段配置

Message Queue 用戶端執行階段會提供用戶端應用程式到 Message Queue 訊息服務的介面。它支援用戶端將訊息傳送至實體目標和從此類目標接收訊息所需的所有作業。用戶端執行階段是可配置的（藉由設定連線工廠屬性值），它可讓您設定可改善效能與訊息流量的特性與運作方式。

例如，Message Queue 用戶端執行階段支援以下可配置運作方式：

- 連線流量計數 (imqConnectionFlowCount)，可幫助您避免 JMS 訊息和 Message Queue 控制訊息同時通過同一連線時，產生擁塞情形。
- 連線流量限制 (imqConnectionFlowLimit)，可幫助您藉由限制可透過連線傳送至用戶端執行階段的待使用訊息數目，來避免用戶端資源限制。

- 用戶流量限制 (`imqConsumerFlowLimit`)，可在多用戶佇列傳送情況中，幫助改善用戶之間的負載平衡（沒有用戶會收到過多的訊息數），並且有助於平衡所有連線用戶的流量。此特性限制每個用戶在連線中可傳送到用戶端執行階段的待使用訊息數目。也可將此特性配置為佇列目標特性 (`consumerFlowLimit`)。

如需更多用於配置之運作方式和屬性的相關資訊，請參閱第 225 頁「用戶端執行階段訊息流量調整」。

調整配置以改善效能

系統調整

以下各節描述您可對作業系統、JVM 和連線協定所做的調整。

Solaris 調校：CPU 使用率、分頁 / 交換 / 磁碟 I/O

請參閱您系統的文件來調校您的作業系統。

Java 虛擬機器調整

依預設，代理程式會使用 192 MB 的 JVM 堆疊大小。此容量對大量的訊息負載通常過小，所以應增加其大小。

當代理程式將要耗盡 Java 物件使用的 JVM 堆疊儲存區空間時，會使用各種技術（如流量控制和訊息交換）來釋放記憶體。在極端情況下，它甚至會關閉用戶端連線以釋放記憶體，並降低訊息的流入量。因此，最好將最大 JVM 堆疊儲存區空間設定得足夠大，以避免發生此類情況。

但是，如果將最大 Java 堆疊儲存區空間設定得過大（相對於系統實體記憶體），代理程式會持續增加 Java 堆疊儲存區空間，直至整個系統的記憶體用完。這會導致效能降低、不可預料的代理程式當機，和 / 或影響系統上執行的其他應用程式和服務的運作。一般而言，您需要允許足夠用於作業系統和機器上其他執行應用程式的實體記憶體。

一般說來，最好是先評估正常時和高峰時的系統記憶體足跡，然後再配置 Java 堆疊儲存區容量，使其足以提供良好的效能，但又不會太大，以至出現系統記憶體問題。

若要變更代理程式最小和最大的堆疊大小，請在啟動代理程式時使用 `-vmargs` 命令行選項。例如：

```
/usr/bin/imqbrokerd -vmargs "-Xms256m -Xmx1024m"
```

此指令會將啟動時的 Java 堆疊大小設為 256MB，且將最大的 Java 堆疊大小設為 1GB。

- 在 Solaris 或 Linux 上，如果經由 `/etc/rc*` (即 `/etc/init.d/imq`) 啟動代理程式，則請在 `/etc/imq/imqbrokerd.conf` (Solaris) 或 `/etc/opt/sun/mq/imqbrokerd.conf` (Linux) 檔案中指定代理程式指令行引數。請查看檔案中的註解，以取得更多資訊。
- 在 Windows 上，如果將代理程式作為 Windows 服務啟動，則請使用 `-vmargs` 選項將 JVM 引數指定到 `imqsvcadm` 安裝指令。請參閱第 13 章「指令參照」中的「`imqsvcadm`」。

在任何情況下，請藉由檢查代理程式的記錄檔或使用 `imqcmd metrics bkr -m cxn` 指令來驗證設定值。

調校傳輸協定

一旦選擇符合應用程式需求的協定，則額外調校 (以選定的協定為基礎) 可能會改善效能。

可以使用以下三個代理程式特性來修改協定效能：

- `imq.protocol.protocol_type.nodelay`
- `imq.protocol.protocol_type.inbufsz`
- `imq.protocol.protocol_type.outbufsz`

針對 TCP 和 SSL 協定，這些特性會影響用戶端和代理程式間訊息傳送的速度。針對 HTTP 和 HTTPS 協定，這些特性會影響 Message Queue 通道 Servlet (在 Web Server 上執行) 和代理程式間的訊息傳送速度。針對 HTTP/HTTPS 協定，有其他特性可以影響效能 (請參閱第 222 頁「HTTP/HTTPS 調校」)。

將會在以下各節描述協定調校特性。

nodelay

`nodelay` 特性會影響用於既定協定的 Nagle 演算法 (TCP/IP 上的 TCP_NODELAY 套接級別選項)。Nagle 演算法可使用慢速連線，例如廣域網路 (WAN)，來改善系統上的 TCP 效能。

使用該演算法時，TCP 會嘗試阻止一些小型資料區塊傳送到遠端系統 (藉由將資料隨附於較大封包中)。如果寫入套接的資料未填滿所需緩衝區大小，那麼協定會在緩衝區填滿或特定延遲時間結束前，延遲傳送封包。一旦緩衝區為填滿狀態或發生逾時，即會傳送封包。

針對大部分的訊息傳送應用程式，在傳送封包時沒有延遲即表示效能為最佳狀態（未啟用 Nagle 演算法）。這是因為大部分用戶端與代理程式間的互動為請求 / 回應互動：用戶端傳送資料封包給代理程式且等待回應。例如，典型的互動包括：

- 建立連線
- 建立產生者或用戶
- 傳送永久性訊息（代理程式會確認訊息接收）
- 在 `AUTO_ACKNOWLEDGE` 或 `CLIENT_ACKNOWLEDGE` 階段作業中傳送用戶端確認（代理程式會再確認確認處理）

針對這些互動，大部分的封包小於緩衝區大小。這表示如果使用 Nagle 演算法，代理程式在傳送回應到用戶時，會延遲數毫秒。

但是，Nagle 演算法會在連線速度緩慢和不需代理程式回應的情況下改善效能。這可能是用戶端傳送非永久性訊息，或用戶端確認未經代理程式確認的情況（`DUPS_OK_ACKNOWLEDGE` 階段作業）。

inbufsz/outbufsz

`inbufsz` 特性會設定輸入串流上的緩衝區大小，輸入串流可讀取來自套接的資料。同樣地，`outbufsz` 會設定輸出串流的緩衝區大小，輸出串流為代理程式用來將資料寫入套接。

一般而言，兩個參數值皆應稍微大於接收或傳送的平均封包大小。根據經驗，請將這些特性值設定為比平均封包大小多 1K（將近 1K）。

舉例來說，如果代理程式接收內文大小為 1K 的封包，那麼封包（訊息內文 + 標頭 + 特性）的總容量則約為 1,200 個位元組一個 2K（2,048 個位元組）的 `inbufsz` 即可產生適當效能。

將 `inbufsz` 或 `outbufsz` 增加到超過此大小可能會稍微改善效能；但是，這可能會增加每個連線的所需記憶體。

圖 11-7 顯示在 1K 的封包上變更 inbufsz 的結果。

圖 11-7 在 1K (1,024 個位元組) 封包上變更 inbufsz 的效果

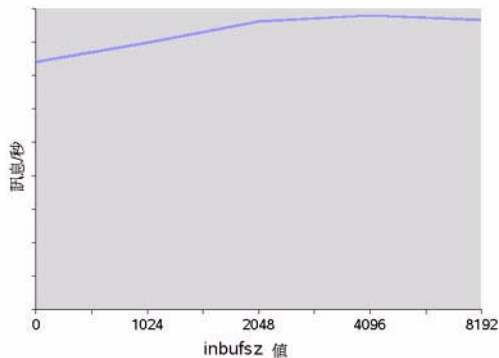
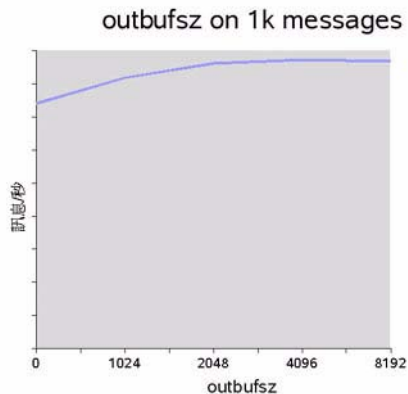


圖 11-8 顯示在 1K 的封包上變更 outbufsz 的結果。

圖 11-8 在 1K (1,024 個位元組) 封包上變更 outbufsz 的效果



HTTP/HTTPS 調校

除了前面兩節描述的一般特性，HTTP/HTTPS 效能受限於用戶端發出 HTTP 請求到管理 Message Queue 通道 Servlet 之 Web Server 的速度。

Web Server 可能需要最佳化，以便處理單一套接上的多個請求。使用 JDK 1.4 或更新版本，HTTP 至 Web Server 的連線會保持暢通（至 Web Server 的套接保持開啓狀態），以便在 Web Server 處理多個 HTTP 請求時，最小化 Web Server 使用的資源。如果在同一個用戶端應用程式上，使用 JDK 1.4 版的效能比使用較早 JDK 版次的效能低，那麼您可能需要調校 Web Server 保持連線暢通的配置參數，以改善效能。

除了對 Web Server 進行此類調校，您也可以調整用戶端輪詢 Web Server 的頻率。HTTP 為基於請求的協定。這表示使用基於 HTTP 協定的用戶端會定期檢查 Web Server，查看是否有等待訊息。img.httpjms.http.pullPeriod 代理程式特性（及其對應的 img.httpsjms.https.pullPeriod 特性）會指定 Message Queue 用戶端執行階段輪詢 Web Server 的頻率。

如果 pullPeriod 值為 -1（預設值），那麼用戶端執行階段會在前個請求傳回時馬上輪詢伺服器，以最大化個別用戶端的效能。因此，每個用戶端連線會獨占 Web Server 中的一個請求執行緒，這可能會過度使用 Web Server 資源。

如果 pullPeriod 值為負數，則用戶端執行階段會定期傳送請求到 Web Server，以查看是否有擱置的資料。在此情況下，用戶端不會獨占 Web Server 中的請求執行緒。因此，如果大量用戶端使用 Web Server，那麼您可能要將 pullPeriod 設為正值，以節省 Web Server 資源。

調校基於檔案的永久性儲存

如需調校基於檔案的永久性儲存，請參閱第 80 頁「持續性管理程式」。

代理程式調整

以下各節描述您可對代理程式特性進行的調整，並藉以改善效能。

記憶體管理：增加代理程式負載的穩定性

可以在目標對目標級別或系統範圍級別（用於所有目標，整體）上配置記憶體管理。

使用實體目標限制

如需實體目標限制的資訊，請參閱第 6 章「管理實體目標」。

使用系統範圍限制

如果訊息產生者嘗試過度執行訊息用戶，可能會在代理程式中累積訊息。代理程式包含一個在記憶體不足情況下，降低產生者速度和將訊息移出使用中記憶體的機制，所以對代理程式可保留的訊息總數目（和訊息位元組）做出硬式限制是明智之舉。

藉由設定 `imq.system.max_count` 和 `imq.system.max_size` 代理程式特性可控制這些限制。

例如：

```
imq.system.max_count=5000
```

以上定義的值表示代理程式最多只會保留 5,000 個未傳送 / 未確認的訊息。如果傳送額外訊息，則會被代理程式拒絕。如果為永久性訊息，那麼產生者在嘗試傳送訊息時會發生異常。如果為非永久性訊息，代理程式會直接丟棄訊息。

若要非永久性訊息如同永久性訊息一樣傳回異常，那麼請在用戶端使用的連線工廠物件上設定以下特性：

```
imqAckOnProduce = true
```

以上設定可能會降低傳送非永久性訊息至代理程式 (用戶端會在傳送下一個訊息前等待回應) 的效能，但是，這情況通常是可接受的，因為流入代理程式的訊息通常不是系統瓶頸。

當傳送訊息時傳回異常，用戶端應稍作暫停並試著再傳送一次。

多重用戶佇列效能

多重佇列用戶在佇列目標中處理訊息的效率，取決於下列可配置的佇列目標屬性：

- 即使用中用戶數目 (`maxNumActiveConsumers`)
- 單一批次中可傳送至用戶的最大訊息數目 (`consumerFlowLimit`)

若要達到最佳化訊息流量，必須有足以跟上用於佇列之訊息產生速率的使用中用戶數目，且必需路由佇列中的訊息，接著將訊息傳送到使用中用戶。此方法會最大化訊息使用率。Sun Java System Message Queue 技術概述中描述平衡多重用戶上訊息傳送的一般機制。

如果訊息累積在佇列中，那麼使用中用戶數目可能會不足以處理訊息負載。也有可能發生以下情形：以批次大小傳送到用戶的訊息會在用戶上進行備份。舉例來說，如果批次大小 (`consumerFlowLimit`) 過大，一個用戶可能會接收佇列中的所有訊息，而其他使用中用戶會收不到任何訊息。如果用戶處理速度非常快，這可能不會發生問題。

但是，如果用戶的速度相對較慢，那麼您必須將訊息平均分散給用戶，如此一來，批次大小也會變小。批次大小越小，傳送訊息至用戶所需的耗用時間就越長。然而，針對處理速度慢的用戶，網路效能改善比率通常會使用小型的批次大小。

用戶端執行階段訊息流量調整

本節描述影響效能的流量控制運作方式 (請參閱第 218 頁「用戶端執行階段配置」)。請將這些運作方式配置為連線工廠受管理物件的屬性。如需設定連線工廠屬性的資訊，請參閱第 8 章「管理受管理物件」。

訊息流量計數

用戶端傳送和接收的訊息 (JMS 訊息) 與 Message Queue 控制訊息，皆會忽略相同用戶端與代理程式間的連線。如果 JMS 訊息傳送阻擋控制訊息，則會造成控制訊息 (例如代理程式確認) 傳送的延遲。若要避免此擁塞情形，Message Queue 要計算連線上的 JMS 訊息流量。

JMS 訊息分為幾個批次 (以 `imqConnectionFlowCount` 特性指定)，因此只傳送一組數字。批次傳送後，JMS 訊息暫停傳送，只傳送擱置控制訊息。接著傳送其他 JMS 訊息，然後傳送擱置控制訊息，重複此循環。

如果用戶端正從代理程式要求許多回應 (例如用戶端使用 `CLIENT_ACKNOWLEDGE` 或 `AUTO_ACKNOWLEDGE` 模式、永久性訊息、作業事件、佇列瀏覽器) 或用戶端正在新增或移除用戶時，`imqConnectionFlowCount` 的值應保持較低。相反地，如果用戶端只有使用 `DUPS_OK_ACKNOWLEDGE` 模式來連線簡易用戶時，那麼您可以增加 `imqConnectionFlowCount` 的值，而不需顧及效能。

訊息流量限制

JMS 訊息的數目有一個限制，即在遇到本地資源限制 (例如記憶體) 前，Message Queue 用戶端執行階段可處理的 JMS 訊息數目限制。到達此限制時，會對效能造成影響。因此，Message Queue 可限制每個用戶可在連線中傳送，且在用戶端執行階段內進行緩衝的待使用訊息數目 (或每個連線的訊息數目)。

基於用戶的限制

如果任何用戶傳送至用戶端執行階段的 JMS 訊息數超過 `imqConsumerFlowLimit` 的值時，則會停止傳送訊息給該用戶。只有在傳送給該用戶的未使用訊息數低於此 `imqConsumerFlowThreshold` 之設定值時，才恢復傳送訊息。

以下範例說明這些限制的使用：考慮主題用戶的設定值

```
imqConsumerFlowLimit=1000
```

```
imqConsumerFlowThreshold=50
```

建立用戶時，代理程式會不間斷地傳送包含 1,000 個訊息的初始批次 (假設訊息已存在) 給該用戶。傳送 1,000 個訊息後，代理程式會停止傳送，直到用戶端執行階段要求傳送其他訊息。應用程式處理這些訊息前，用戶端執行階段會保留這些訊息。接著，在要求代理程式傳送下一個批次前，用戶端執行階段會允許應用程式使用至少 50% (`imqConsumerFlowThreshold`) 的訊息緩衝容量 (例如 500 個訊息)。

在相同情況下，如果臨界值為 10% 時，在要求傳送下一個批次前，用戶端執行階段會等待應用程式至少使用 900 個訊息。

以下計算下一個批次的大小：

$$\text{imqConsumerFlowLimit} - (\text{目前緩衝區中擱置的訊息數目})$$

如此一來，如果 `imqConsumerFlowThreshold` 為 50%，那麼下一個批次大小介於 500 到 1,000 之間，其實際數量取決於應用程式處理訊息的速度。

如果 `imqConsumerFlowThreshold` 設得太高 (接近 100%)，那麼代理程式會嘗試傳送較小的批次，以降低訊息流量。如果值過低 (接近 0%)，那麼用戶端可能可以在代理程式傳送下一組批次前，完成處理其餘的緩衝訊息，這會造成訊息流量下降。一搬來說，除非您有特殊效能或可靠性考量，否則您不需變更 `imqConsumerFlowThreshold` 屬性的預設值。

基於用戶的流量控制 (特別是 `imqConsumerFlowLimit`) 為管理用戶端執行階段中的最佳方法。通常，根據用戶端應用程式，您會曉得您在任何連線上需要支援的用戶數目、訊息大小和可用於用戶端執行階段的總記憶體容量。

基於連線的限制

然而，在一些用戶端應用程式中，用戶數目可能是不正確的，它取決於一般使用者的選擇。在此情況下，您仍可以使用連線級別流量限制來管理記憶體。

連線級別流量控制可限制用於連線上所有用戶的總緩衝訊息數目。如果此數目超過 `imqConnectionFlowLimit`，則會停止透過連線傳送訊息，直到總數目低於連線限制。(只有在您將 `imqConnectionFlowLimitEnabled` 特性設為 `true` 時，才可啟用 `imqConnectionFlowLimit`。)

階段作業中形成佇列的訊息數目為使用階段作業之用戶數目及用於每個用戶之訊息負載的函數。如果用戶端在產生和使用訊息中發生延遲，通常您可以藉由重新設定應用程式，以分散較大階段作業數目上的訊息產生者和用戶，或者分散較大連線數目上的階段作業，並因此改善效能。

排解疑難問題

本章說明如何瞭解及解決下列問題：

- 第 228 頁 「用戶端無法建立連線」
- 第 232 頁 「連線流量過慢」
- 第 234 頁 「用戶端無法建立訊息產生者」
- 第 235 頁 「訊息產生延遲或過慢」
- 第 237 頁 「儲存訊息」
- 第 241 頁 「訊息伺服器的流量不穩定」
- 第 242 頁 「訊息未送達用戶」
- 第 246 頁 「停用的訊息佇列中包含訊息」

發生問題時，檢查所安裝的 **Message Queue** 軟體的版本編號，通常很有幫助。版本編號能讓您確定您使用的文件的編號，是否與軟體版本相符。此外，向 Sun 報告問題時，也需要版本編號。若要檢查版本編號，請發出以下指令：

```
imqcmd -v
```

用戶端無法建立連線

這個問題有下列徵兆：

- 用戶端無法建立新的連線。
- 用戶端無法自動重新連線失敗的連線。

本節提供下列幾個可能原因：

- 用戶端應用程式未關閉連線，導致連線數目超過資源限制
- 未執行代理程式或網路連結發生問題
- 連線服務不在使用中或已暫停
- 可用於所需連線數目的可用執行緒太少
- Solaris 或 Linux 作業系統上要求用於連線數目的檔案描述元太少
- TCP 儲存區會限制同時新建之連線請求的數目
- 作業系統會限制運作中連線的數目
- 使用者的驗證或授權失敗

用戶端應用程式未關閉連線，導致連線數目超過資源限制

確認此問題的原因

列出所有代理程式的連線：

```
imqcmd list cxn
```

輸出會列出所有連線和建立每個連線的主機，顯示特定用戶端已開啓之不經常使用的連線數目。

若要解決此問題

重新寫入違例用戶端，以關閉不使用的連線。

未執行代理程式或網路連結發生問題

確認這個問題的原因

- 使用 Telnet 連線至用戶端主要連接埠 (例如, 預設值為 7676), 並以連接埠對映器輸出驗證代理程式回應。
- 驗證代理程式程序是否在主機上執行。

若要解決此問題

- 啟動代理程式。
- 修復網路連結問題。

連線服務不在使用中或已暫停

確認這個問題的原因

檢查所有連線服務的狀態：

```
imqcmd list svc
```

如果連線服務狀態顯示如下 unknown (未知) 或 paused (暫停中), 用戶端無法使用服務建立連線。

若要解決此問題

- 如果連線服務狀態顯示為 unknown (未知), 此服務會從使用中服務清單移出 (imq.service.active)。在基於 SSL 服務的情況中, 可能會錯誤配置此服務, 造成代理程式在代理程式記錄中產生以下項目: 錯誤 [B3009]:無法啟動服務 ssljms:[B4001]:無法開啓 ssljms 服務的協定 tls..., 並隨附發生異常的基本原因說明。
若要正確配置 SSL 服務, 請參閱第 146 頁「使用基於 SSL 的服務」。
- 如果連線服務狀態顯示為 paused (暫停中), 會繼續服務 (請參閱第 112 頁「暫停與繼續連線服務」)。

可用於所需連線數目的可用執行緒太少

確認這個問題的原因

檢查代理程式記錄中是否有以下項目：

```
WARNING [B3004]:沒有可用於服務上處理新連線的執行緒...關閉新的連線。
```

再使用下列其中一種格式, 檢查連線服務上的連線數目和目前使用中的執行緒數目：

```
imqcmd query svc -n serviceName
```

```
imqcmd metrics svc -n serviceName -m cxn
```

每個連線均需要兩個執行緒：一個用於內送訊息，另一個用於外寄訊息（請參閱第 75 頁「執行緒池管理程式」）。

若要解決此問題

- 如果您使用專用執行緒池模型 (`imq.service_name.threadpool_model=dedicated`)，那麼最大的連線數目為執行緒池中最大執行緒數目的一半。因此，若要增加連線數目，請增加執行緒池的容量 (`imq.service_name.max_threads`) 或切換為共用執行緒池模型。
- 如果您使用共用執行緒池模型 (`imq.service_name.threadpool_model=shared`)，那麼最大的連線數目為以下兩種特性產品的一半：連線監視限制 (`imq.service_name.connectionMonitor_limit`) 和最大執行緒數目 (`imq.service_name.max_threads`)。因此，若要增加連線數目，請增加執行緒池的容量或增加連線監視限制。
- 可支援的連線數目（或連線上的流量）最終會到達輸入 / 輸出限制。在此情況下，請使用多重代理程式叢集，以分散叢集中代理程式實例上的連線。

Solaris 或 *Linux* 作業系統上要求用於連線數目的檔案描述元太少

如需有關此問題的更多資訊，請參閱第 64 頁「設定檔案描述元限制 (*Solaris* 或 *Linux*)」。

確認這個問題的原因

檢查代理程式記錄中是否有以下相似項目：太多開啓檔案。

若要解決此問題

增加檔案描述元限制，如 `ulimit` 線上手冊所述。

TCP 儲存區會限制同時新建之連線請求的數目

TCP 儲存區會在連接埠對映器拒絕其他請求前，限制同時儲存在系統儲存區內的連線請求數目 (`imq.portmapper.backlog`)。（在 *Windows* 作業系統上有程序內定的儲存區限制：*Windows Desktop* 的數目為 5，*Windows Server* 的數目為 200。）

因為不經常有高數目的同時連線請求，所以因儲存區限制的請求拒絕通常為短暫現象。

確認此問題的原因

檢查代理程式記錄。首先，檢查代理程式在拒絕其他連線的同時是否有接受某些連線。接著，檢查拒絕連線的說明訊息。如果找到這類訊息，可能不是 *TCP* 儲存區的問題，因為代理程式不記錄 *TCP* 儲存區造成的拒絕連線。

如果有記錄連線成功而且沒有記錄拒絕連線，則可能是 *TCP* 儲存區的問題。

若要解決此問題

可以用以下方法來解決 TCP 儲存區限制：

- 經過一小段間隔時間後，讓用戶端重試連線（此方法通常可行，因為此問題為暫時性的問題）。
- 增加 `imq.portmapper.backlog` 的值。
- 檢查用戶端關閉和開啓連線的次數是否過於頻繁。

作業系統會限制運作中連線的數目

Windows 作業系統授權會限制支援的運作遠端連線的數目。

確認此問題的原因

檢查可用於連線的執行緒數目（使用 `imqcmd query svc`），並檢查 Windows 授權合約的條款。如果您是從本地用戶端建立連線，而不是從遠端用戶端，作業系統限制可以是造成此問題的原因。

若要解決此問題

- 升級 Windows 授權以允許更多連線。
- 設定多重代理程式叢集，以分散一些代理程式實例的連線。

使用者的驗證或授權失敗

驗證可能會因為錯誤密碼而失敗，因為沒有使用者進入使用者儲存庫的項目，或因為使用者沒有存取連線服務的許可權。

確認此問題的原因

檢查代理程式記錄中是否有禁用錯誤訊息的項目。這表示有驗證錯誤，但不表示為問題發生的原因。

- 如果您使用基於檔案的使用者儲存庫，請輸入以下指令：

```
imqusermgr list -i instanceName -u userName
```
- 如果輸出顯示使用者，可能是提交了錯誤密碼。如果輸出下列錯誤，使用者儲存庫中不會有項目：

```
Error [B3048]:使用者不在密碼檔案中，
```
- 如果您使用 LDAP 伺服器使用者儲存庫，則請使用適當的工具檢查是否有用於使用者的項目。
- 檢查存取控制特性檔案，查看是否有存取連線服務的限制。

若要解決此問題

- 如果使用者儲存庫中沒有用於使用者的項目，請將使用者新增到使用者儲存庫 (請參閱第 136 頁「寫入和管理使用者儲存庫」)。
- 如果使用的密碼錯誤，請提供正確的密碼。
- 如果存取控制特性未正確設定，請編輯存取控制特性檔案，以取得連線服務許可權 (請參閱第 144 頁「連線服務的存取控制」)。

連線流量過慢

這個問題有下列徵兆：

- 訊息流量未符合預期速度。
- 支援連線到代理程式的數目不受到限制，如第 228 頁「用戶端無法建立連線」所述，但訊息輸入 / 輸出速率受到限制。

本節提供下列幾個可能原因：

- 網路連線或 WAN 太慢
- 連線服務協定的速度原本就比 TCP 的速度慢
- 連線服務協定未調校到最佳化
- 訊息過大導致使用過多的頻寬
- 實際上，連線流量過慢的問題，是訊息傳送程序的某些步驟中會遇到的瓶頸

網路連線或 WAN 太慢

確認此問題的原因

偵測網路，查看偵測到網路回應所需的時間，並洽詢網路管理員。您也可以使用本地用戶端傳送和接受訊息，並與遠端用戶端 (使用網路連結) 比較傳送時間。

若要解決此問題

如果連線過慢，請升級網路連結。

連線服務協定的速度原本就比 TCP 的速度慢

例如，基於 SSL 或基於 HTTP 協定的速度會較 TCP 的速度慢 (請參閱第 216 頁圖 11-5)。

確認此問題的原因

如果您使用基於 SSL 或基於 HTTP 的協定，請嘗試使用 TCP 並比較傳送時間。

若要解決此問題

應用程式需求通常會表示要使用的協定，這是您可以參考的一點。除此之外，您還可以嘗試調校協定，如第 220 頁「調校傳輸協定」所述。

連線服務協定未調校到最佳化

確認此問題的原因

嘗試調校協定，並查看是否有何差異。

若要解決此問題

嘗試調校協定，如第 220 頁「調校傳輸協定」所述。

訊息過大導致使用過多的頻寬

確認此問題的原因

嘗試以較小訊息執行效能評定。

若要解決此問題

- 請應用程式開發者修改應用程式，使用 Message Queue Developer's Guide for Java Clients 中描述的訊息壓縮功能。
- 將訊息使用作為要傳送的資料通知，但使用其他協定移動資料。

實際上，連線流量過慢的問題，是訊息傳送程序的某些步驟中會遇到的瓶頸

確認此問題的原因

如果上述原因都不是造成連線流量速度過慢的原因，則請查閱第 207 頁圖 11-1，找出其他可能的瓶頸，並檢查是否有與以下問題相關的徵兆：

- 第 235 頁「訊息產生延遲或過慢」
- 第 237 頁「儲存訊息」
- 第 241 頁「訊息伺服器的流量不穩定」

若要解決此問題

依照以上問題排解各節提供的問題解決方針。

用戶端無法建立訊息產生者

這個問題有下列徵兆：

- 無法建立用於實體目標的訊息產生者；用戶端發生異常。

本節提供下列幾個可能原因：

- 已將實體目標配置為僅允許一定數目的產生者
- 由於存取控制特性檔案中，未設定授權使用者建立訊息產生者

已將實體目標配置為僅允許一定數目的產生者

若要防止實體目標上訊息累積，其中一個方法是限制實體目標可支援的產生者數目 (maxNumProducers)。

確認此問題的原因

檢查實體目標 (請參閱第 123 頁「顯示實體目標資訊」)：

```
imqcmd query dst
```

輸出會顯示目前的產生者數目和 maxNumProducers 的值。如果兩個值相同，表示產生者數目到達配置的限制。代理程式拒絕新的產生者時，代理程式會傳回 ResourceAllocationException [C4088]：到達 JMS 目標限制，且在代理程式記錄中產生以下項目：[B4183]：無法將產生者新增至目標。

若要解決此問題

增加 maxNumProducers 屬性的值 (請參閱第 124 頁「更新實體目標特性」)。

由於存取控制特性檔案中，未設定授權使用者建立訊息產生者

確認此問題的原因

代理程式拒絕新的產生者時，代理程式會傳回下列訊息：

```
JMSSecurityException [C4076]：用戶端沒有權限在目標上建立產生者
```

代理程式也會在代理程式記錄中記錄下列項目：

```
[B2041]：在目標上的產生者被拒絕 和 [B4051]：禁用 guest。
```

若要解決此問題

變更存取控制特性，以允許使用者產生訊息 (請參閱第 145 頁「實體目標的存取控制」)。

訊息產生延遲或過慢

這個問題有下列徵兆：

- 傳送永久性訊息時，不會傳回 `send()` 方法且阻塞用戶端。
- 傳送永久性訊息時，用戶端發生異常。
- 產生用戶端的速度變慢。

本節提供下列幾個可能原因：

- [訊息伺服器進行儲存動作並且做出降低訊息產生者速度的回應](#)
- [代理程式無法將永久性訊息儲存到資料儲存](#)
- [代理程式確認逾時過短](#)
- [生產型用戶端受到 JVM 限制](#)

訊息伺服器進行儲存動作並且做出降低訊息產生者速度的回應
進行儲存動作的伺服器將訊息累積在代理程式記憶體中。

實體目標中的訊息數目或訊息容量到達配置限制時，代理程式會依指定的限制運作方式，嘗試節省記憶體資源。以下限制運作方式會降低訊息產生者的速度：

- `FLOW_CONTROL`：代理程式不會馬上確認永久性訊息的接收（因此會阻塞生產型用戶端）。
- `REJECT_NEWEST`：代理程式拒絕新的永久性訊息。

同樣地，代理程式範圍記憶體（用於所有實體目標）中的訊息數目或訊息容量到達配置限制時，代理程式會藉由拒絕最新訊息，嘗試節省記憶體資源。

再者，到達系統記憶體限制時，因為實體目標或代理程式範圍的限制未正確設定，代理程式會採取越來越重要的動作，以防記憶體超過負載。這些動作包括減低訊息產生者的速度。

確認此問題的原因

代理程式因配置的訊息限制拒絕訊息時，代理程式會傳回下列訊息：

```
JMSEException [C4036]：伺服器出錯
```

代理程式也會在代理程式記錄中記錄此項目：

```
警告 [B2011]：儲存來自 IMQconn 的 JMS 訊息失敗
```

此訊息後面還有一則訊息指出已經到達限制。如果訊息限制位於實體目標上，那麼代理程式會產生類似以下的項目：

[B4120]：無法在目標 *destName* 上儲存訊息，因為會超出 *maxNumMsgs* 的容量。

如果訊息限制為代理程式範圍，那麼代理程式會產生類似以下的項目：

[B4024]：已經超出目前系統中訊息的最大數目，正在拒絕訊息。

一般而言，您可以在拒絕訊息前檢查訊息限制條件，方法為：

- 查詢實體目標和代理程式，檢查配置的訊息限制設定。
- 使用適當的 *imqcmd* 指令，監視目前實體目標或整體代理程式中的訊息數目或訊息容量。如需可監視的度量和度量取得指令的詳細資訊，請參閱第 18 章「度量參照」。

若要解決此問題

有數種方法可以解決訊息儲存時造成產生者速度變慢的問題。

- 修改實體目標（或代理程式範圍）上的訊息限制，保持不要超過記憶體資源。
一般來說，您應該在目標對目標層級上管理記憶體，如此一來，才永遠不會到達代理程式範圍的訊息限制。如需更多資訊，請參閱第 223 頁「代理程式調整」。
- 變更目標上的限制運作方式，使其在到達訊息限制時不會降低訊息產生速度，而非拋棄記憶體中的訊息。

例如，您可以指定可刪除記憶體中累積訊息的 *REMOVE_OLDEST* 和 *REMOVE_LOW_PRIORITY* 限制運作方式（請參閱第 303 頁表 15-1）。

代理程式無法將永久性訊息儲存到資料儲存

如果代理程式無法存取資料儲存或將永久性訊息寫入資料儲存，則表示生產型用戶端被封鎖。這種情況亦會在目標或代理程式範圍訊息到達限制時發生，如上所述。

確認此問題的原因

如果代理程式無法寫入資料儲存，則會在代理程式記錄中產生以下其中一個項目：

[B2011]：儲存來自 *connectionID* 的 JMS 訊息失敗 ... 或 [B4004]：無法存留訊息 *messageID*...

若要解決此問題

- 在內建持續性的情況下，請嘗試增加基於檔案的資料儲存的磁碟空間。
- 在使用 JDBC 相容資料儲存的情況下，請檢查是否正確配置外掛持續性（請參閱第 4 章「配置代理程式」）。如果已正確配置，請洽詢您的資料庫管理員以排解其他資料庫問題。

代理程式確認逾時過短

由於連線速度慢或訊息伺服器緩慢 (因高 CPU 使用率或記憶體資源不足所引起)，代理程式可能需要更多時間來確認持久性訊息的接收，且時間超過連線工廠的 `imqAckTimeout` 屬性所允許的值。

確認此問題的原因

如果超過 `imqAckTimeout` 的值，代理程式會傳回下列訊息：

```
JMSEException [C4000]: 資料封包確認失敗
```

若要解決此問題

變更 `imqAckTimeout` 連線工廠屬性值 (請參閱第 162 頁「連線工廠屬性」)。

生產型用戶端受到 JVM 限制

確認此問題的原因

- 檢查用戶端應用程式是否接收到「記憶體不足」錯誤。
- 使用執行階段方法，例如 `freeMemory()`、`MaxMemory()` 和 `totalMemory()`，來檢查 JVM 堆疊中的可用記憶體。

若要解決此問題

調整 JVM (請查看第 219 頁「Java 虛擬機器調整」)。

儲存訊息

這個問題有下列徵兆：

- 代理程式 (或特定目標) 中的訊息數目或訊息容量會不斷穩定增加。

若要查看是否累積訊息，請檢查代理程式中的訊息數目或訊息容量如何不斷變化，並與配置的限制進行比較。請先檢查配置的限制：

```
imqcmd query bkr
```

(注意：`imqcmd metrics bkr` 子指令不會顯示此資訊。)

接著檢查每個目標中的訊息累積：

```
imqcmd list dst
```

若要查看訊息是否超過配置的目標或代理程式範圍限制，請檢查代理程式中是否有以下項目：警告 [B2011]：儲存 JMS 訊息，來自 ... 失敗。此項目會隨附其他項目，說明已超過限制。

- 訊息產生延遲或代理程式拒絕產生的訊息。
- 訊息花費一段不尋常的長時間到達用戶。

本節提供下列幾個可能原因：

- 主題目標上有非使用中的長期訂閱
- 佇列中可使用訊息的用戶太少
- 訊息用戶處理速度過慢，無法跟上訊息產生者的速度
- 用戶端確認處理會降低訊息使用速度
- 代理程式無法跟上產生訊息的速度
- 用戶端程式碼缺點：用戶不確認訊息

主題目標上有非使用中的長期訂閱

如果長期訂閱為非使用中，訊息會儲存在目標中，直到對應的用戶變為使用中狀態，且可使用這些訊息為止。

確認此問題的原因

檢查每個主題目標上的長期訂閱狀態：

```
imqcmd list dur -d destName
```

若要解決此問題

您可以採取以下任何動作：

- 清除所有用於違例長期訂閱的訊息（請參閱第 115 頁「管理長期訂閱」）。
- 指定用於主題的訊息限制和訊息運作方式屬性（請參閱第 303 頁表 15-1）。例如，您可以指定可刪除記憶體中累積訊息的 REMOVE_OLDEST 和 REMOVE_LOW_PRIORITY 限制運作方式。
- 從對應的目標清除所有訊息（請參閱第 125 頁「清除實體目標」）。
- 時間訊息的限制可保留在記憶體中。您可以重新寫入生產型用戶端，以設定每個訊息上的存在時間值。您可以覆寫用於所有產生者共用連線的任何設定值，方法為設定 imqOverrideJMSEExpiration 和 imqJMSEExpiration 的連線工廠屬性（請參閱第 312 頁「訊息標頭覆寫」）。

佇列中可使用訊息的用戶太少

如果可傳送訊息的使用中用戶太少，那麼可以在訊息累積時儲存佇列目標。此情況會因以下任何一種原因而發生：

- 目標中存在的使用中用戶太少。

- 使用用戶端建立連線失敗。
- 沒有使用中用戶使用符合佇列中訊息的選擇器。

確認此問題的原因

若要判斷無法使用用戶的原因，請檢查目標上使用中用戶的數目：

```
imqcmd metrics dst -n destName -t q -m con
```

若要解決此問題

您可以根據無法使用用戶的原因，採取以下任何一個動作：

- 啟動其他使用用戶端，以建立更多用於佇列的使用中用戶。
- 調整 `imq.consumerFlowLimit` 代理程式特性，以最佳化至多個用戶的佇列傳送（請參閱第 224 頁「多重用戶佇列效能」）。
- 指定用於佇列的訊息限制和訊息運作方式屬性（請參閱第 303 頁表 15-1）。例如，您可以指定可刪除記憶體中累積訊息的 `REMOVE_OLDEST` 和 `REMOVE_LOW_PRIORITY` 限制運作方式。
- 從對應的目標清除所有訊息（請參閱第 125 頁「清除實體目標」）。
- 時間訊息的限制可保留在記憶體中。您可以重新寫入生產型用戶端，以設定每個訊息的存在時間值。另外，還可覆寫用於所有產生者共用連線的任何設定值，方法為設定 `imqOverrideJMSEExpiration` 和 `imqJMSEExpiration` 連線工廠屬性（請參閱第 312 頁「訊息標頭覆寫」）。

訊息用戶處理速度過慢，無法跟上訊息產生者的速度

在此情況下，主題用戶或佇列接收者使用訊息的速度，會比產生者傳送訊息的速度慢。因為此不平衡現象，所以會儲存一個或多個包含訊息的目標。

確認此問題的原因

檢查流入和流出代理程式的訊息速率：

```
imqcmd metrics bkr -m rts
```

接著檢查每個個別目標的流量速率：

```
imqcmd metrics bkr -t destType -n destName -m rts
```

若要解決此問題

- 最佳化使用用戶端程式碼。
- 增加用於佇列目標的使用中用戶數目（請參閱第 224 頁「多重用戶佇列效能」）。

用戶端確認處理會降低訊息使用速度

兩個影響用戶端確認處理的因素：

- 處理用戶端確認時會大量使用代理程式資源。因此，可能會降低阻塞使用用戶端之確認模式中的訊息使用速度，直到代理程式確認用戶端確認為止。
- JMS 有效負載訊息和 Message Queue 控制訊息 (例如用戶端確認) 共用同一個連線。因此，JMS 有效負載訊息可阻擋控制訊息，因而降低訊息使用的速度。

確認此問題的原因

- 檢查與封包流量相關的訊息流量。如果每秒的封包數目與訊息數目不成比例，用戶端確認可能有問題。
- 檢查用戶端是否接收到下列訊息：

```
JMSException [C4000]：封包確認失敗
```

若要解決此問題

- 修改用戶端使用的確認模式，例如切換為 `DUPS_OK_ACKNOWLEDGE` 或 `CLIENT_ACKNOWLEDGE`。
- 如果使用 `CLIENT_ACKNOWLEDGE` 或作業事件階段作業，請將更大的訊息數目設為單一確認的群組。
- 調整用戶和連線流量控制參數 (請參閱第 225 頁「[用戶端執行階段訊息流量調整](#)」)。

代理程式無法跟上產生訊息的速度

在此情況下，訊息流入代理程式的速度，會比代理程式路由和派送訊息到用戶的速度快。代理程式運作遲緩可以是因為以下任何或所有限制造成：CPU、網路套接讀 / 寫作業、磁碟讀 / 寫作業、記憶體分頁、永久性儲存或 JVM 記憶體限制。

確認此問題的原因

檢查是否有其他造成此問題的原因。

若要解決此問題

- 升級您電腦或資料儲存的速度。
- 使用代理程式叢集，分散一些代理程式實例上的負載。

用戶端程式碼缺點：用戶不確認訊息

所有用戶確認傳送的訊息前，訊息會被保留在目標中。如果用戶端不確認使用的訊息，那麼訊息會累積在目標中且不被刪除。

舉例來說，用戶端程式碼可能會有以下缺點：

- 使用 `CLIENT_ACKNOWLEDG` 確認或作業事件階段作業的用戶可能不會定期呼叫 `Session.acknowledge()` 或 `Session.commit()`。
- 使用 `AUTO_ACKNOWLEDGE` 階段作業的用戶可能會因某些原因擱置。

確認此問題的原因

首先，檢查本節中列出的所有其他可能原因。然後，使用下列指令，列出目標：

```
imqcmd list dst
```

請注意 `UnAcked` 標頭下列出的訊息數目，是否與目標中的訊息數目相同。`UnAcked` 標頭下的訊息是傳給用戶但尚未確認的訊息。如果這個數目與訊息總數相同，表示代理程式已經送出所有訊息，正在等待確認。

若要解決此問題

請應用程式開發者協助除錯這個問題。

訊息伺服器流量不穩定

這個問題有下列徵兆：

- 訊息流量偶爾會降低，接著又恢復正常效能。

本節提供下列幾個可能原因：

- [代理程式在記憶體資源上的速度非常緩慢](#)
- [執行 JVM 記憶體回收 \(廢棄項目收集\)](#)
- [JVM 使用 Just-In-Time 編譯器來提高效率](#)

代理程式在記憶體資源上的速度非常緩慢

因為未正確設定目標與代理程式限制，代理程式會採取越來越重要的動作，以防記憶體超過負載。另外，在清除訊息儲存前，會導致代理程式運作緩慢。

確認此問題的原因

檢查記憶體不足情況下的代理程式記錄 ([B1089]：在記憶體不足的條件下，代理程式正嘗試釋放資源)，並隨附一個項目，描述新的記憶體狀態和使用的記憶體總數。

還要檢查 JVM 堆疊中可用的記憶體：

```
imqcmd metrics bkr -m cxn
```

JVM 記憶體總數接近最大 JVM 記憶體值時，可用記憶體會不足。

若要解決此問題

- 調整 JVM (請查看第 219 頁「Java 虛擬機器調整」)。
- 增加系統交換空間。

執行 JVM 記憶體回收 (廢棄項目收集)

會定期透過系統清除記憶體回收，釋放記憶體。發生此情況時，會阻塞所有執行緒。被釋放的記憶體容量越多，JVM 堆疊的大小就越大，且因記憶體回收的延遲時間也越長。

確認此問題的原因

檢視您電腦的 CPU 使用。執行記憶體回收時，會大幅降低 CPU 使用。

再者，請使用以下指令行選項啟動代理程式：

```
-vmargs -verbose:gc
```

標準輸出說明執行記憶體回收的時間。

若要解決此問題

在多重 CPU 的電腦中，設定同時執行的記憶體回收：

```
-XX:+UseParallelGC=true
```

JVM 使用 Just-In-Time 編譯器來提高效能

確認此問題的原因

檢查是否有其他造成此問題的原因。

若要解決此問題

讓系統執行一段時間，效能應會改善。

訊息未送達用戶

這個問題有下列徵兆：

- 用戶未收到產生者傳送的訊息。

本節提供下列幾個可能原因：

- [限制運作方式導致訊息在代理程式上被刪除](#)

- 訊息的逾時值過期
- 時鐘不同步
- 使用用戶端無法啟動連線上的訊息傳送

限制運作方式導致訊息在代理程式上被刪除

目標記憶體中的訊息數目或訊息容量到達配置的限制時，代理程式嘗試節省記憶體資源。代理程式會配置三個可配置的運作方式，而到達這些限制時會使訊息遺失：

- REMOVE_OLDEST：刪除最舊的訊息
- REMOVE_LOW_PRIORITY：根據訊息存在時間，刪除最後順位的訊息
- REJECT_NEWEST：拒絕新的永久性訊息

代理程式記憶體中的訊息數目或訊息容量到達配置限制時，代理程式藉由拒絕最新訊息，嘗試節省記憶體資源。

確認此問題的原因

如第 246 頁「停用的訊息佇列中包含訊息」中所描述，檢查停用的訊息佇列。尤其要遵照第 247 頁「訊息數目或訊息容量超出目標限制」中的指示。尋找 REMOVE_OLDEST 或 REMOVE_LOW_PRIORITY 原因。

若要解決此問題

增加目標限制。例如：

```
imqcmd update dst -n MyDest -o maxNumMsgs=1000
```

訊息的逾時值過期

代理程式會刪除逾時值過期的訊息。如果目標儲存足夠的訊息，則可能會刪除存在時間值過短的訊息。

確認此問題的原因

檢查停用的訊息佇列有無訊息過期。

您可以使用 QBrowser 示範應用程式，查看 DMQ 內容。QBrowser 示範程式在作業系統的特定位置中，請參閱附錄 A「Message Queue 資料的作業系統特定位置」，查看「範例應用程式與位置」的表格。

這是 Windows 上的呼叫範例：

```
cd \MessageQueue3\demo\applications\qbrowser java QBrowser
```

QBrowser 主視窗出現時，請選取佇列名稱 `mq.sys.dmq`，然後按一下 [瀏覽]。會出現類似下面的清單。

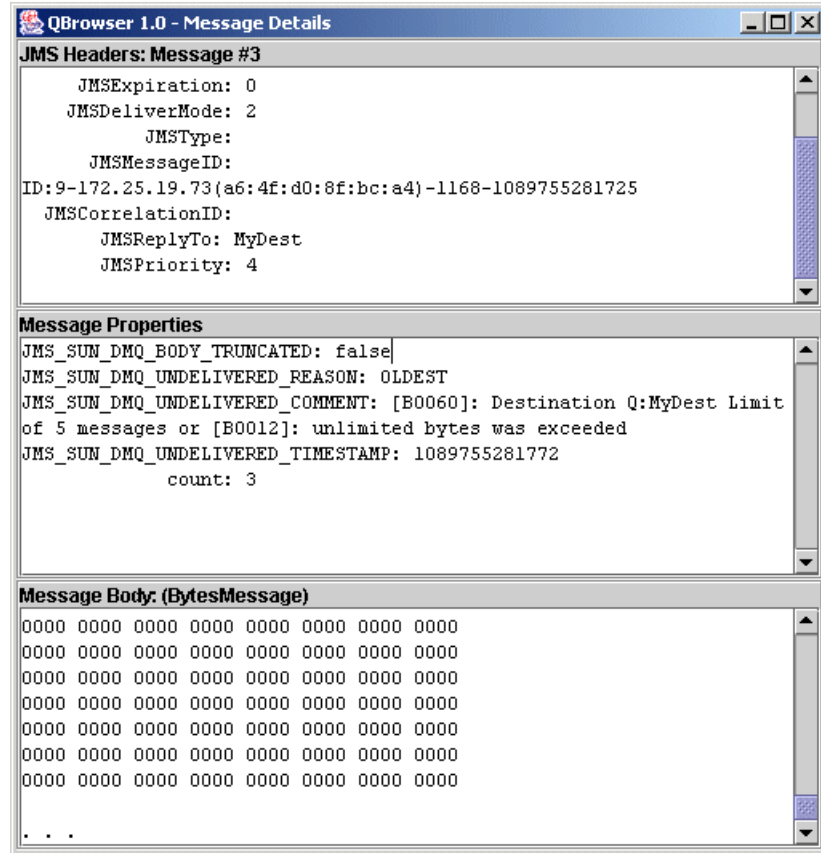
圖 12-1 QBrowser 視窗

File					
Queue Name: mq.sys.dmq					Browse
#	Timestamp	Type	Mode	Priority	
0	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4	
1	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4	
2	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4	
3	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4	
4	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4	
5	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4	
6	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4	
7	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4	
8	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4	
9	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4	
10	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4	
11	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4	
12	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4	
13	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4	
14	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4	
15	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4	
16	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4	
17	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4	
18	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4	
19	13/Jul/2004:14:48:01 PDT	BytesMessage	P	4	
20	13/Jul/2004:14:53:50 PDT	BytesMessage	P	4	
21	13/Jul/2004:14:53:50 PDT	BytesMessage	P	4	
22	13/Jul/2004:14:53:50 PDT	BytesMessage	P	4	
23	13/Jul/2004:14:53:50 PDT	BytesMessage	P	4	
24	13/Jul/2004:14:53:50 PDT	BytesMessage	P	4	
25	13/Jul/2004:14:53:50 PDT	BytesMessage	P	4	
26	13/Jul/2004:14:53:50 PDT	BytesMessage	P	4	
27	13/Jul/2004:14:53:50 PDT	BytesMessage	P	4	
28	13/Jul/2004:14:53:50 PDT	BytesMessage	P	4	

mq.sys.dmq: 35 Details...

在訊息上連接兩下，即可顯示此訊息的詳細資訊。

圖 12-2 QBrower 訊息的詳細資訊



請注意訊息的 JMS_SUN_DMQ_UNDELIVERED_REASON 特性值是否為 EXPIRED。

若要解決此問題

連絡應用程式開發者並增加存在時間值。

時鐘不同步

如果時鐘不同步，代理程式計算訊息的存在時間會發生錯誤，如此會導致訊息逾時並被刪除。

確認此問題的原因

查看代理程式記錄檔中有無下列訊息：B2102、B2103、B2104。這些訊息都報告偵測到時鐘可能不準。

若要解決此問題

如第 64 頁「準備系統資源」中所述，檢查有無執行時間同步化程式。

使用用戶端無法啟動連線上的訊息傳送

用戶端程式碼建立連線並啟動連線上的訊息傳送前，將無法傳送訊息。

確認此問題的原因

檢查用戶端程式碼是否建立連線並啟動訊息傳送。

若要解決此問題

重新寫入用戶端程式碼，以建立連線並啟動訊息傳送。

停用的訊息佇列中包含訊息

這個問題有下列徵兆：

- 您列出目標時，會看到停用的訊息佇列中包含訊息。例如，請發出下列指令。

```
imqcmd list dst
```

您提供使用者名稱和密碼後，會出現下列輸出：

正在列示由以下所指定的代理程式的所有目標：							
Host	Primary Port						
localhost	7676						
Name	Type	State	Producers	Consumers	Msgs Total Count	UnAck	Avg Size
MyDest	Queue	RUNNING	0	0	5	0	1177.0
mq.sys.dmq	Queue	RUNNING	0	0	35	0	1422.0

成功地列示目標。

這個範例中的停用的訊息佇列 mq.sys.dmq，包含 35 個訊息。

本節提供下列幾個可能原因：

- 訊息數目或訊息容量超出目標限制
- 代理程式時鐘和產生者時鐘不同步
- 用戶在訊息逾時前一直未收到訊息
- 產生者對用戶數目而言過多
- 產生者的速度比用戶快
- 用戶的速度太慢
- 用戶端未確定訊息
- 長期用戶非使用中
- 發生非預期的代理程式錯誤

訊息數目或訊息容量超出目標限制

確認此問題的原因

您可以使用 QBrower 示範應用程式，查看停用的訊息佇列的內容。QBrower 示範程式在作業系統的特定位置中，請參閱附錄 A 「Message Queue 資料的作業系統特定位置」，查看「範例應用程式與位置」的表格。

這是 Windows 上的呼叫範例：

```
cd \MessageQueue3\demo\applications\qbrowser java QBrower
```

QBrower 主視窗出現時，請選取佇列名稱 mq.sys.dmq，然後按一下 [瀏覽]。出現第 244 頁圖 12-1 中顯示的清單。

在任何訊息上連按兩下，即可顯示此訊息的詳細資訊。出現第 245 頁圖 12-2 中顯示的視窗。

請注意下列訊息特性值：

- JMS_SUN_DMQ_UNDELIVERED_REASON
- JMS_SUN_DMQ_UNDELIVERED_COMMENT
- JMS_SUN_DMQ_UNDELIVERED_TIMESTAMP

請注意 JMS 標頭下的 JMSDestination 值，判斷訊息將要停用的目標。

解決問題

增加目標限制。例如：

```
imqcmd update dst -n MyDest -o maxNumMsgs=1000
```

代理程式時鐘和產生者時鐘不同步

若要確認此問題的原因：

您可以使用 **QBrowser** 應用程式，在停用的訊息佇列中檢視訊息的詳細資訊。檢查 **JMS_SUN_DMQ_UNDELIVERED_REASON** 值，尋找原因為 **EXPIRED** 的訊息。

查看代理程式記錄檔中有無下列訊息：**B2102**、**B2103**、**B2104**。這些訊息都報告偵測到時鐘可能不準。

解決問題

如第 64 頁「[準備系統資源](#)」中所述，檢查有無執行時間同步化程式。

用戶在訊息逾時前一直未收到訊息

驗證此問題的原因

您可以使用 **QBrowser** 應用程式，在停用的訊息佇列中檢視訊息的詳細資訊。檢查 **JMS_SUN_DMQ_UNDELIVERED_REASON** 值，尋找原因為 **EXPIRED** 的訊息。

檢查目標上有無任何用戶。例如：

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

檢查列出的 [目前使用中用戶數目] 值。如果有使用中用戶，下列其中一項為 **true**：

- 用戶暫停連線。
- 訊息逾時過短，無法配合用戶的執行速度。

若要解決此問題

請應用程式開發者增加訊息存在時間值。

產生者對用戶數目而言過多

確認此問題的原因

您可以使用 **QBrowser** 應用程式，在停用的訊息佇列中檢視訊息的詳細資訊。檢查 **JMS_SUN_DMQ_UNDELIVERED_REASON** 值。

如果原因是 **REMOVE_OLDEST** 或 **REMOVE_LOW_PRIORITY**，請使用 `imqcmd query dst` 指令，檢查目標上的產生者數目和用戶數目。如果產生者數目超過用戶數目，產生率可能高於使用率。

若要解決此問題

新增更多的用戶用戶端或設定目標使用 **FLOW_CONTROL** 限制運作方式。**FLOW_CONTROL** 限制運作方式是以使用率來控制產生率。

如下列範例所示，使用指令來啟動流量控制運作方式：

```
imqcmd update dst -n myDst -t q -o consumerFlowLimit=FLOW_CONTROL
```

產生者的速度比用戶快

確認此問題的原因

若要判斷用戶速度慢是否導致產生者的速度變慢，請將目標限制運作方式設定為 FLOW_CONTROL。FLOW_CONTROL 限制運作方式是以使用率來控制產生率。

如下列範例所示，使用指令來啟動流量控制運作方式：

```
imqcmd update dst -n myDst -t q -o consumerFlowLimit=FLOW_CONTROL
```

發出下列範例中的指令，使用度量來檢查目標的輸入與輸出：

```
imqcmd metrics dst -n myDst -t q -m rts
```

檢查度量輸出中的下列值：

- Msgs/sec Out

此值顯示代理程式每秒移除多少訊息。所有用戶確認收到訊息後，代理程式便移除訊息，讓度量反映使用率。

- Msgs/sec In

此值顯示代理程式每秒接收到多少產生者的訊息。度量會反映產生率。

流量控制會對齊產生與使用，注意產生速度是否變慢或停止。如果產生率變慢或停止，表示產生者和用戶的處理速度有差異。

您也可以使用 `imqcmd list dst` 指令，檢查已傳送的未確認 (UnAcked) 訊息數目。如果未確認的訊息數目小於目標容量，表示目標還有其他容量被用戶端的流量控制所控制。

若要解決此問題

如果要產生率一直快於使用率，請考慮定期使用流量控制讓系統步調一致。

後續幾節介紹如何解決下面的可能因素：

- [用戶的速度太慢](#)
- [用戶端未確定訊息](#)
- [用戶無法確認訊息](#)
- [長期用戶非使用中](#)
- [發生非預期的代理程式錯誤](#)

用戶的速度太慢

確認此問題的原因

如第 249 頁「產生者的速度比用戶快」中所描述，使用度量來判斷產生率和使用率。

若要解決此問題

嘗試下列一或多種操作：

- 設定目標使用 FLOW_CONTROL 限制運作方式。您可以使用如下的指令：

```
imqcmd update dst -n myDst -t q -o consumerFlowLimit=FLOW_CONTROL
```

使用流量控制讓產生率不超過使用率，防止代理程式上的訊息累積。產生者應用程式會保留訊息，直到目標可以及時處理為止，降低過期風險。

- 向應用程式開發者詢問產生者是以固定的速率傳送訊息或是定期資料組傳送。如果應用程式是資料組傳送訊息，請按照下一個項目的指示，增加目標限制。
- 根據訊息的數目和 / 或位元組，增加目標限制。

若要變更目標的訊息數目，請輸入下列格式的指令：

```
imqcmd update dst -n destName -t {q/t} -o maxNumMsgs=number
```

若要變更目標的訊息容量，請輸入下列格式的指令：

```
imqcmd update dst -n destName -t {q/t} -o maxTotalMsgBytes=number
```

請注意增加限制，表示增加代理程式使用的記憶體容量。如果限制太高，代理程式可能會用光記憶體而無法處理訊息。

- 考慮在高產生負荷級別期間，能否承受訊息流失。

用戶端未確定訊息

確認此問題的原因

向應用程式開發者查詢應用程式是否使用作業事件。如果應用程式使用作業事件，請下達下列指令，列出使用中作業事件：

```
imqcmd list txn
```

下面是指令的輸出範例：

Transaction ID	State	User name	# Msgs/# Acks	Creation time
6800151593984248832	STARTED	guest	3/2	7/19/04 11:03:08 AM

請注意訊息和確認的數目。

如果訊息數目大於確認數目，表示產生者可能有傳送個別訊息，但卻無法確定作業事件。代理程式收到確定後，才能路由並傳送該項作業事件的訊息。

如果確認數目大於訊息數目，表示用戶可能有傳送個別訊息，但卻無法確定作業事件。代理程式收到確定後，才能移除該項作業事件的確認。

解決問題

連絡應用程式開發者，修正程式碼錯誤。

用戶無法確認訊息

確認此問題的原因

連絡應用程式開發者，判斷應用程式使用基於系統或基於用戶端的確認。如果應用程式使用基於系統的確認，請略過本節。

如果應用程式使用基於用戶端的確認 (CLIENT_ACKNOWLEDGE 類型)，請先減少用戶端上儲存的訊息數目。您可以使用類似下面的指令：

```
imqcmd update dst -n myDst -t q -o consumerFlowLimit=1
```

接著判斷代理程式緩衝訊息是因為用戶的速度太慢，還是用戶處理訊息的速度很快，但確認的速度不快。

使用下列指令，列出目標：

```
imqcmd list dst
```

您提供使用者名稱和密碼後，會出現類似下面的輸出：

正在列示由以下所指定的代理程式的所有目標：

Host	Primary Port						
localhost	7676						
Name	Type	State	Producers	Consumers	Msgs Total Count	UnAck	Avg Size
MyDest	Queue	RUNNING	0	0	5	200	1177.0
mq.sys.dmq	Queue	RUNNING	0	0	35	0	1422.0

成功地列示目標。

UnAck 數目代表代理程式傳送出去後等待確認的訊息數目。如果 UnAck 數目變大或增加，表示代理程式正在傳送訊息，不等速度較慢的用戶。您也可以看出用戶未確認訊息。

若要解決此問題

請連絡應用程式開發者，修正程式碼錯誤。

長期用戶非使用中

確認此問題的原因

使用下面的指令格式，查看主題的長期用戶：

```
imqcmd list dur -d topicName
```

若要解決此問題

- 使用 `imqcmd purge dur` 指令，清除長期用戶。
- 重新啟動用戶應用程式。

發生非預期的代理程式錯誤

確認此問題的原因

如第 249 頁「產生者的速度比用戶快」中所描述，使用 QBrowser 檢查訊息。

如果 `JMS_SUN_DMQ_UNDELIVERED_REASON` 值是 `ERROR`，表示代理程式發生錯誤。

若要解決此問題

- 檢查代理程式記錄檔，尋找相關錯誤。
- 連絡 Sun 技術支援，報告代理程式的問題。

第 13 章	指令參照
第 14 章	代理程式特性參照
第 15 章	實體目標特性參照
第 16 章	受管理物件屬性參照
第 17 章	JMS 資源介面屬性參照
第 18 章	度量參照

指令參照

本章有一節描述一般指令行語法，並提供每個 Message Queue 指令的參照資訊。本章包含以下各節：

- 第 256 頁 「指令行語法」
- 第 257 頁 「imqbrokerd」
- 第 261 頁 「imqcmd」
- 第 271 頁 「imqobjmgr」
- 第 273 頁 「imqdbmgr」
- 第 275 頁 「imqusermgr」
- 第 277 頁 「imqsvcadmin」
- 第 279 頁 「imqkeytool」

指令行語法

Message Queue 指令行公用程式是 Shell 指令。公用程式名稱是一個指令，其子指令或選項是傳給指令的引數。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 Message Queue 公用程式的名稱，例如，`imqcmd`、`imqobjmgr` 和 `imqusermgr` 等。

指令的輸入規則

輸入指令時，有幾個一般規則：

- 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。
- 如果選項值包含空格，請將整個值括在引號內。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` (Version) 或 `-h/-H` (Help) 選項，則系統不會執行此指令行上的其他任何選項。
- 使用空格將子指令、引數、選項和選項引數分隔。

指令行範例

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
```

下列指令更複雜。這個指令銷毀名稱為 `myQueue` 的 `queue` 類型目標。根據使用者 `admin` 執行認證；指令會提示您輸入使用者的密碼。`-f` 選項指定無確認，`-s` 選項指定以無訊息模式執行指令。

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

共用的指令選項

表 13-1 描述了所有 Message Queue 管理公用程式共用的選項。在指令行上指定這些選項時，必須放在子指令的後面。選項的輸入順序不拘。

表 13-1 共用 Message Queue 指令行選項

選項	說明
-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 imqcmd 和 imqobjmgr）。
-s	開啓無訊息模式：不顯示輸出。為 imqbrokerd 指定 -silent。
-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 imqobjmgr）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 相容執行階段（預設為使用系統上的執行階段，或 Message Queue 隨附的執行階段）。

imqbrokerd

imqbrokerd 指令可以啓動代理程式。指令行選項可覆寫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。

語法

```
imqbrokerd [[ -Dproperty=value] ...]
  [ -backup fileName]
  [ -cluster "[broker1] [,broker2]..." ]
  [ -dbuser userName]
  [ -force]
  [ -h|-help]
  [ -javahome path]
  [ -license licenseName]
  [ -loglevel level]
  [ -metrics interval]
  [ -name instanceName]
  [ -passfile fileName]
```

```

[ -port number]
[ -remove instance]
[ -reset data]
[ -restore fileName]
[ -shared]
[ -silent|-s] [ -tty]
[ -upgrade-store-nobackup]
[ -version]
[ -vmargs arg1 [arg2]...]

```

指令選項

表 13-2 描述了 `imqbrokerd` 指令的選項，還描述了每個選項影響的配置特性（如果有）。

表 13-2 `imqbrokerd` 選項

選項	影響的特性	說明
<code>-backup <i>fileName</i></code>	無影響的特性。	僅套用於代理程式叢集。將主代理程式的配置變更記錄備份至指定檔案。請參閱第 184 頁「管理配置變更記錄」。
<code>-cluster" [<i>broker1</i> [<i>broker2</i>]...] "</code>	將 <code>imq.cluster.brokerlist</code> 覆寫成與其連線的代理程式清單。	僅套用於代理程式叢集。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> 特性中的清單合併。如果您沒有為主機指定值，則使用 <code>localhost</code> 。如果您沒有為 <code>port</code> 指定值，則使用值 7676。請參閱第 179 頁「使用代理程式叢集」，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。
其中 <i>broker</i> 為以下之一		
• <code>host</code>		
• <code>:port</code>		
• <code>host:port</code>		
<code>-dbpassword <i>password</i></code>	將 <code>imq.persist.jdbc.password</code> 覆寫成指定的密碼	為外掛 JDBC 相容資料庫指定密碼。這個選項已經被停用，將從未來的版本中移除。請使用下列其中一種替代方案： <ul style="list-style-type: none"> 在指令行上省略密碼，讓指令提示您輸入密碼。 使用 <code>-passfile</code> 選項，指定包含資料庫密碼的檔案。
<code>-dbuser <i>userName</i></code>	將 <code>imq.persist.jdbc.user</code> 覆寫成指定的使用者名稱	為外掛 JDBC 相容資料庫指定使用者名稱。請參閱第 93 頁「設定永久性儲存」。

表 13-2 imqbrokerd 選項 (續)

選項	影響的特性	說明
<code>-Dproperty=value</code>	設定系統特性。覆寫實例配置檔案中的相應特性值。	將指定的特性設定為指定的值。請參閱第 14 章「代理程式特性參照」，以獲得代理程式配置特性的相關資訊。 警告： 請仔細檢查使用 <code>-D</code> 選項設定的特性的拼寫與格式。如果您傳送了錯誤的值，系統將不警告您，並且 <code>Message Queue</code> 亦無法設定這些值。
<code>-force</code>	無影響的特性。	未經使用者確認即執行動作。此選項僅套用通常要求確認的 <code>-remove instance</code> 和 <code>-upgrade-store-nobackup</code> 選項。
<code>-h -help</code>	無影響的特性。	顯示說明。不執行指令行上的其他任何指令。
<code>-javahome path</code>	無影響的特性。	指定替代 Java 2 相容 JDK 的路徑。預設為使用隨附的執行階段。
<code>-ldappassword password</code>	將 <code>imq.user_repository.ldap.password</code> 覆寫成指定的密碼	為存取 LDAP 使用者儲存庫指定密碼。這個選項已經被停用，將從未來的版本中移除。請使用下列其中一種替代方案： <ul style="list-style-type: none"> 在指令行上省略密碼，讓指令提示您輸入密碼。 使用 <code>-passfile</code> 選項，指定包含 LDAP 密碼的檔案。
<code>-license [licenseName]</code>	無影響的特性。	如果要載入的授權與 <code>Message Queue</code> 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。根據已安裝的 <code>Message Queue</code> 版本， <code>licenseName</code> 值為 <code>pe</code> (Platform Edition - 基本特性)、 <code>try</code> (Platform Edition - 90 天試用企業特性) 和 <code>unl</code> (Enterprise Edition)。
<code>-loglevel level</code>	將 <code>imq.broker.log.level</code> 覆寫成指定的級別。	將記錄級別指定為以下之一： <code>NONE</code> 、 <code>ERROR</code> 、 <code>WARNING</code> 或 <code>INFO</code> 。預設值為 <code>INFO</code> 。
<code>-metrics interval</code>	將 <code>imq.metrics.interval</code> 覆寫成指定的秒數。	指定按指定間隔時間 (以秒為單位) 將代理程式度量寫入記錄程式中。
<code>-name instanceName</code>	將 <code>imq.instancename</code> 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，則將實例名稱設定為 <code>imqbroker</code> 。 備註： 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。

表 13-2 imqbrokerd 選項 (續)

選項	影響的特性	說明
<code>-passfile fileName</code>	覆寫 <code>imq.passfile.enabled</code> ，並且設定為 <code>true</code> 。將 <code>imq.passfile.dirpath</code> 覆寫成檔案路徑。將 <code>imq.passfile.name</code> 覆寫成檔案名稱。	指定檔案的名稱，您可從此檔案讀取 <code>imqcmd</code> 指令公用程式、SSL 密鑰儲存、LDAP 使用者儲存庫、JDBC 相容資料庫或上述任何組合的密碼。如需更多資訊，請參閱第 155 頁「使用密碼檔案」。
<code>-password keypassword</code>	將 <code>imq.keystore.password</code> 覆寫成指定的密碼。	指定 SSL 憑證密鑰儲存的密碼。這個選項已經被停用，將從未來的版本中移除。請使用下列其中一種替代方案： <ul style="list-style-type: none"> 在指令行上省略密碼，讓指令提示您輸入密碼。 使用 <code>-passfile</code> 選項，指定包含 SSL 憑證密鑰儲存密碼的檔案。
<code>-port number</code>	將 <code>imq.portmapper.port</code> 覆寫成指定的號碼。	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有不同的連接埠號。Message Queue 用戶端會使用此連接埠號連線至代理程式實例。
<code>-remove instance</code>	無影響的特性。	導致移除代理程式實例：刪除此實例配置檔案、記錄檔、永久性儲存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 <code>-force</code> 選項。
<code>-reset store messages durables props</code>	無影響的特性。	依據給定的引數，重設資料儲存 (或資料儲存的子集) 或代理程式實例的配置特性。 重設資料儲存，清除所有永久性資料，包括永久性訊息、長期訂閱和作業事件資訊。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長期訂閱。(如果您不想在後續的重新啟動中重設永久性儲存，請重新啟動未使用 <code>-reset</code> 選項的代理程式實例。) 重設代理程式的特性，用空白檔案取代現有實例配置檔案 (<code>config.properties</code>)：所有特性均假設為預設值。
<code>-restore fileName</code>	無影響的特性。	僅套用於代理程式叢集。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 <code>-backup</code> 選項事先建立。請參閱第 184 頁「管理配置變更記錄」。

表 13-2 imqbrokerd 選項 (續)

選項	影響的特性	說明
-shared	覆寫 imq.jms.threadpool_model，並且設定為 shared。	指定要使用共用執行緒池模型實作的 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目。
-silent -s	覆寫 imq.log.console.output，並且設定為 NONE。	關閉對主控台的記錄。
-tty	覆寫 imq.log.console.output，並且設定為 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。
-upgrade-store-nobackup	無影響的特性	指定從不相容的版本升級到 Message Queue 3.5 或 Message Queue 3.5 SPx 時會自動移除舊的資料儲存。如需其他詳細資訊，請參閱 Message Queue Installation Guide。
-version	無影響的特性。	顯示安裝產品的版本編號。
-vmargs arg1 [[arg2]...]	無影響的特性	<p>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：</p> <pre>imqbrokerd -tty -vmargs "-Xmx128m -Xincgc"</pre> <p>您只能在指令行上傳送這些引數。 config.props 檔案中沒有相關的配置特性。</p>

另請參閱

如需有關使用 imqbrokerd 的詳細資訊和指令範例，請參閱第 64 頁「[以互動方式啓動代理程式](#)」。

imqcmd

imqcmd 指令公用程式能讓您管理代理程式及其服務。

語法

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

子指令

您使用 `imqcmd` 時一定使用子指令，除非是顯示說明或產品版本。表 13-3 列出 `imqcmd` 子指令，並指出子指令參照資訊的位置。

表 13-3 `imqcmd` 子指令

子指令與引數	說明	參照
<code>commit txn</code>	確定作業事件。	第 268 頁「作業事件管理子指令」
<code>destroy dur</code>	銷毀長期訂閱。	第 268 頁「長期訂閱子指令」
<code>list cxn</code>	列出代理程式的連線。	第 268 頁「連線子指令」
<code>list dur</code>	列出主題的長期訂閱。	第 268 頁「長期訂閱子指令」
<code>list svc</code>	列出代理程式上的服務。	第 267 頁「連線服務管理子指令」
<code>list txn</code>	列出代理程式上的作業事件。	第 268 頁「作業事件管理子指令」
<code>metrics bkr</code>	顯示代理程式度量。	第 264 頁「代理程式管理子指令」
<code>metrics svc</code>	顯示服務度量。	第 267 頁「連線服務管理子指令」
<code>pause bkr</code>	暫停代理程式上的所有服務。	第 264 頁「代理程式管理子指令」
<code>pause svc</code>	暫停代理程式上的單一服務。	第 267 頁「連線服務管理子指令」
<code>purge dur</code>	清除長期訂閱上的所有訊息而不銷毀長期訂閱。	第 268 頁「長期訂閱子指令」
<code>query bkr</code>	查詢並顯示代理程式的相關資訊。	第 264 頁「代理程式管理子指令」
<code>query cxn</code>	查詢並顯示連線的相關資訊。	第 268 頁「連線子指令」
<code>query svc</code>	查詢並顯示服務的相關資訊。	第 267 頁「連線服務管理子指令」
<code>query txn</code>	查詢並顯示作業事件的相關資訊。	第 268 頁「作業事件管理子指令」
<code>reload cls</code>	重新載入代理程式叢集配置。	第 264 頁「代理程式管理子指令」

表 13-3 imqcmd 子指令 (續)

子指令與引數	說明	參照
restart bkr	重新啓動目前執行的代理程式實例。	第 264 頁「代理程式管理子指令」
resume bkr	繼續代理程式上的所有服務。	第 264 頁「代理程式管理子指令」
resume svc	繼續服務。	第 267 頁「連線服務管理子指令」
rollback txn	回轉作業事件。	第 268 頁「作業事件管理子指令」
shutdown bkr	關閉代理程式實例。	第 264 頁「代理程式管理子指令」
update bkr	更新代理程式的屬性。	第 264 頁「代理程式管理子指令」
update svc	更新服務的屬性。	第 267 頁「連線服務管理子指令」

imqcmd 指令公用程式也有子指令要用於代理程式的實體目標。第 6 章「管理實體目標」中有描述目標子指令。

下面幾節依功能列出 imqcmd 的子指令。

代理程式管理子指令

表 13-4 列出用來管理代理程式的 imqcmd 子指令。如果未指定主機名稱或連接埠，則使用預設值 (localhost:7676)。

表 13-4 imqcmd 子指令用於管理代理程式

子指令語法	說明
<pre>metrics bkr [-b hostName:port] [-m metricType] [-int interval] [-msp numSamples]</pre>	<p>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度量。</p> <p>使用 -m 選項指定要顯示的度量類型：</p> <p>ttl 顯示代理程式流入和流出訊息和封包的度量。(預設度量類型)</p> <p>rts 顯示代理程式(每秒)流入和流出訊息和封包之流量速率的度量。</p> <p>cxn 顯示連線、虛擬記憶體堆疊和執行緒。</p> <p>使用 -int 選項，指定顯示度量的間隔時間(以秒為單位)。預設為 5 秒鐘。</p> <p>使用 -msp 選項，指定輸出中顯示的範例數目。預設的數目為沒有限制(無限)。</p>
<pre>pause bkr [-b hostName:port]</pre>	<p>暫停預設代理程式或指定主機和連接埠上的代理程式。請參閱第 106 頁「暫停與繼續代理程式」。</p>
<pre>query bkr -b hostName:port</pre>	<p>列出預設代理程式或指定主機和連接埠上的代理程式特性的目前設定。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。</p>
<pre>reload cls</pre>	<p>僅套用於代理程式叢集。強制叢集中的所有代理程式重新載入 imq.cluster.brokerlist 特性並更新叢集資訊。請參閱第 182 頁「將代理程式新增至叢集」，以獲得更多資訊。</p>
<pre>restart bkr [-b hostName:port]</pre>	<p>使用啟動代理程式時指定的選項，關閉預設代理程式或指定主機和連接埠上的代理程式後再重新啟動。</p>
<pre>resume bkr [-b hostName:port]</pre>	<p>繼續預設代理程式或指定主機和連接埠上的代理程式。</p>
<pre>shutdown bkr [-b hostName:port]</pre>	<p>關閉預設代理程式或指定主機和連接埠上的代理程式。</p>

表 13-4 imqcmd 子指令用於管理代理程式 (續)

子指令語法	說明
update bkr [-b <i>hostName:port</i>] -o <i>attribute=value</i> [-o <i>attribute=value1</i>]...	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。

實體目標管理指令

表 13-5 列出用來管理實體目標的 imqcmd 子指令。如果未指定主機名稱或連接埠，則使用預設值 (localhost:7676)。

表 13-5 用於管理目標的 imqcmd 子指令

子指令語法	說明
compact dst [-t <i>destType</i>] -n <i>destName</i>	壓縮用於指定類型與名稱之目標的內建且基於檔案的資料儲存。如果沒有指定目標類型與名稱，所有目標都會壓縮。必須在壓縮目標前暫停目標。
create dst -t <i>destType</i> -n <i>destName</i> [-o <i>attribute=value</i>] [-o <i>attribute=value1</i>]...	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「_」和「\$」開頭。不能以「mq」字元字串開頭。 不能在主代理程式暫時無法使用的叢集上執行這個作業。
destroy dst -t <i>destType</i> -n <i>destName</i>	銷毀指定類型和名稱的目標。不能銷毀系統建立的目標，例如，停用的訊息佇列。 不能在主代理程式暫時無法使用的叢集上執行這個作業。
list dst [-t <i>destType</i>] [-tmp]	列出指定類型的所有目標，以及列出暫存目標的選項。 類型引述可以有兩個值： <i>destType</i> = q (queue) <i>destType</i> = t (topic) 如果未指定類型，則會列出所有類型的所有目標。

表 13-5 用於管理目標的 imqcmd 子指令 (續)

子指令語法	說明
<pre>metrics dst -t <i>destType</i> -n <i>destName</i> [-m <i>metricType</i>] [-int <i>interval</i>] [-msp <i>numSamples</i>]</pre>	<p>顯示有關指定類型和名稱之目標的度量資訊。</p> <p>使用 -m 選項指定要顯示的度量類型：</p> <p>ttl 顯示目標流入和流出訊息和封包以及常駐記憶體度量。(預設度量類型)</p> <p>rts 顯示目標(每秒)流入和流出訊息和封包之流量速率的度量，及其他速率資訊。</p> <p>con 顯示用戶相關的度量。</p> <p>dsk 顯示磁碟使用度量。</p> <p>使用 -int 選項，指定顯示度量的間隔時間(以秒為單位)。預設為 5 秒鐘。</p> <p>使用 -msp 選項，指定輸出中顯示的範例數目。預設的數目為沒有限制(無限)。</p>
<pre>pause dst [-t <i>destType</i> -n <i>destName</i>] [-pst <i>pauseType</i>]</pre>	<p>針對指定類型和名稱的目標，暫停至用戶的訊息傳送(-pst CONSUMERS)，或暫停來自產生者的訊息傳送(-pst PRODUCERS)，或者兩者皆暫停(-pst ALL)。如果沒有指定目標類型與名稱，則會暫停所有目標。預設為 ALL。</p>
<pre>purge dst -t <i>destType</i> -n <i>destName</i></pre>	<p>清除在指定類型與名稱的目標的訊息。</p>
<pre>query dst -t <i>destType</i> -n <i>destName</i></pre>	<p>列出指定類型和名稱之目標的相關資訊。</p>
<pre>resume dst [-t <i>destType</i> -n <i>destName</i>]</pre>	<p>繼續用於指定類型和名稱之暫停目標的訊息傳送。如果沒有指定目標類型與名稱，則會繼續所有目標。</p>
<pre>update dst -t <i>destType</i> -n <i>destName</i> -o <i>attribute=value</i> [-o <i>attribute=value1</i>]...</pre>	<p>更新指定目標上的指定屬性的值。</p> <p>屬性名稱可能是表 15-1 中描述的任何屬性，除非目標是停用的訊息佇列 mq.sys.dmq。</p>

連線服務管理 子指令

表 13-6 列出用來管理連線服務的 imqcmd 子指令。如果未指定主機名稱或連接埠，則使用預設值 (localhost:7676)。

表 13-6 用於管理連線服務的 imqcmd 子指令

子指令語法	說明
<code>list svc [-b <i>hostName:port</i>]</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 <i>serviceName</i> [-b <i>hostName:port</i>] [-m <i>metricType</i>] [-int <i>interval</i>] [-msp <i>numSamples</i>]</code>	顯示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 -m 選項指定要顯示的度量類型： ttl 顯示代理程式經由指定服務，流入和流出訊息和封包的度量。(預設度量類型) rts 顯示代理程式經由指定連線服務，(每秒)流入和流出訊息和封包之流量速率的度量。 cxn 顯示連線、虛擬記憶體堆疊和執行緒。 使用 -int 選項，指定顯示度量的間隔時間(以秒為單位)。預設為 5 秒鐘。 使用 -msp 選項，指定輸出中顯示的範例數目。預設的數目為沒有限制(無限)。
<code>pause svc -n <i>serviceName</i> [-b <i>hostName:port</i>]</code>	暫停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。您無法暫停管理服務。
<code>query svc -n <i>serviceName</i> [-b <i>hostName:port</i>]</code>	顯示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相關資訊。
<code>resume svc -n <i>serviceName</i> [-b <i>hostName:port</i>]</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 <i>serviceName</i> [-b <i>hostName:port</i>] -o <i>attribute=value</i> [-o <i>attribute=value1</i>] ...</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。如需服務屬性的說明，請參閱第 285 頁「連線服務特性」。

連線子指令

表 13-7 列出套用於連線的 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 13-7 用於管理連線服務的 imqcmd 子指令

子指令語法	說明
list cxn [-svn <i>serviceName</i>] [-b <i>hostName:port</i>]	列出預設代理程式上指定服務名稱的所有連線，或指定主機和連接埠上代理程式上的所有連線。如果未指定服務名稱，則會列出所有服務。
query cxn -n <i>connectionID</i> [-b <i>hostName:port</i>]	顯示在預設代理程式或指定主機和連接埠上的代理程式上，指定服務的相關資訊。

長期訂閱子指令

表 13-8 提供了 imqcmd 長期訂閱子指令的摘要。如果未指定主機名稱或連接埠，則使用預設值 (localhost 和 7676)。

表 13-8 用於管理長期訂閱的 imqcmd 子指令

子指令	說明
list dur -d <i>destName</i>	列出指定目標的所有長期訂閱。
destroy dur -n <i>subscrName</i> -c <i>client_id</i>	使用指定的用戶端識別碼銷毀指定的長期訂閱。 不能在主代理程式暫時無法使用的叢集上執行這個作業。
purge dur -n <i>subscrName</i> -c <i>client_id</i>	使用指定的用戶端識別碼清除指定之長期訂閱的所有訊息。

作業事件管理子指令

表 13-9 提供了 imqcmd 作業事件子指令的摘要。如果未指定主機名稱或連接埠，則使用預設值 (localhost 和 7676)。

表 13-9 用於管理作業事件的 imqcmd 子指令

子指令	說明
list txn	列出要由代理程式追蹤的所有作業事件。

表 13-9 用於管理作業事件的 imqcmd 子指令 (續)

子指令	說明
query txn -n <i>transaction_id</i>	列出指定作業事件的相關資訊。
commit txn -n <i>transaction_id</i>	確定指定的作業事件。
rollback txn -n <i>transaction_id</i>	回轉指定的作業事件。

指令選項

表 13-10 列出 imqcmd 指令的選項。

表 13-10 imqcmd 選項

選項	說明
-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>	將長期用戶的 ID 指定至主題。請參閱第 115 頁「管理長期訂閱」。
-d <i>destinationName</i>	指定主題名稱。與 list dur 和 destroy dur 子指令配合使用。請參閱第 115 頁「管理長期訂閱」。
-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。 這個選項不需要使用者名稱和密碼。
-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。 這個選項不需要使用者名稱和密碼。
-int <i>interval</i>	指定 metrics bkr、metrics dst 與 metrics svc 子指令顯示度量輸出的時間間隔 (以秒為單位)。
-javahome <i>path</i>	指定要使用的替代 Java 2 相容執行階段 (預設為使用系統上的執行階段, 或 Message Queue 隨附的執行階段)。
-m <i>metricType</i>	指定要顯示的度量資訊類型。將此選項與 metrics dst、metrics svc 或 metrics bkr 子指令配合使用。metricType 的值會根據是否產生用於目標、服務或代理程式的度量而變化。
-msp <i>numSamples</i>	指定度量輸出中, 顯示作為 metrics bkr、metrics dst 與 metrics svc 子指令範例的度量數目。

表 13-10 imqcmd 選項 (續)

選項	說明
-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、長期訂閱名稱、連線 ID 或作業事件 ID，這取決於子指令。
-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 102 頁「使用 imqcmd 指令公用程式 」) 的屬性、服務 (請參閱第 109 頁「 管理連線服務 」) 的屬性或目標 (請參閱第 115 頁「 管理長期訂閱 」) 的屬性，這取決於子指令引數。
-p <i>password</i>	指定您的 (管理員的) 密碼。這個選項已經被停用，未來的版本不支援。請使用下列其中一種替代方案： <ul style="list-style-type: none"> 在指令行上省略密碼，讓指令提示您輸入密碼。 使用 <code>-passfile</code> 選項，指定包含管理員密碼的檔案。
-passfile <i>path</i>	指定包含使用者下達指令時所需密碼的檔案路徑。如需更多資訊，請參閱第 155 頁「 使用密碼檔案 」。
-pst <i>pauseType</i>	暫停目標時，指定是否要暫停產生者、用戶或兩者皆暫停。請參閱第 115 頁「 管理長期訂閱 」。
-rtm <i>timeout</i>	指定 imqcmd 子指令的初始 (重試) 逾時期間 (以秒為單位)。送出請求到代理程式後，imqcmd 子指令將等待的逾時時間長度。每個子指令的後續重試會使用一個逾時值，該逾時值為初始逾時期間的倍數。預設值：10
-rtr <i>numRetries</i>	指定 imqcmd 子指令第一次逾時後，嘗試重試的次數。預設值：5
-s	無訊息模式。不顯示輸出。
-secure	使用 <code>ssladmin</code> 連線服務指定與代理程式的安全管理連線 (請參閱第 151 頁「 步驟 4. 配置並執行基於 SSL 的用戶端 」)。如果省略這個選項，連線可能會不安全。
-svn <i>serviceName</i>	指定所列出連線的服務。請參閱第 113 頁「 取得有關連線的資訊 」。
-t <i>destType</i>	指定目標的類型：t (主題) 或 q (佇列)。請參閱第 115 頁「 管理長期訂閱 」。
-tmp	顯示暫存目標。請參閱表 13-5 頁碼是 265。
-u <i>userName</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。 這個選項不需要使用者名稱和密碼。

另請參閱

如需有關使用 `imqcmd` 的詳細資訊和指令範例，請參閱第 5 章「管理代理程式」和第 6 章「管理實體目標」。

imqobjmgr

物件管理員公用程式 (`imqobjmgr`) 可以建立並管理 Message Queue 受管理物件。

語法

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

子指令

物件管理員公用程式 (`imqobjmgr`) 包括表 13-3 中列出的子指令：

表 13-11 imqobjmgr 子指令

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

指令選項

表 13-12 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 13-12 imqobjmgr 選項

選項	說明
-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>	指定指令檔的名稱，此檔案包含所有或部分子指令子句，可指定物件類型、查找名稱、物件屬性、物件儲存屬性或其他選項。通常用於重複資訊，如物件儲存屬性。
-j <i>attribute=value</i>	指定識別和存取 JNDI 物件儲存所需的屬性。請參閱第 160 頁「關於物件儲存」。
-j <i>javahome path</i>	指定要使用的替代 Java 2 相容執行階段（預設為使用系統上的執行階段，或 Message Queue 隨附的執行階段）。
-l <i>lookupName</i>	指定受管理物件的 JNDI 查找名稱。此名稱在物件儲存的環境中必須唯一。
-o <i>attribute=value</i>	指定受管理物件的屬性。請參閱第 16 章第 307 頁「受管理物件屬性參照」。
-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 true 表示受管理物件為唯讀物件。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 false。
-s	無訊息模式。不顯示輸出。

表 13-12 imqobjmgr 選項 (續)

選項	說明
-t <i>objectType</i>	指定 Message Queue 受管理物件的類型： q = 佇列 t = 主題 cf = 連線工廠 qf = 佇列連線工廠 tf = 主題連線工廠 xcf = XA 連線工廠 (分散式作業事件) xqf = XA 佇列連線工廠 (分散式作業事件) xtf = XA 主題連線工廠 (分散式作業事件) e = SOAP 端點 (這個受管理物件類型如 Message Queue Developer's Guide for Java Clients 中所描述，是用來支援 SOAP 訊息。)
-v	顯示版本資訊。不執行指令行上的其他任何指令。

另請參閱

如需有關 imqobjmgr 的詳細資訊和指令範例，請參閱第 8 章「管理受管理物件」

imqdbmgr

資料庫管理員公用程式 (imqdbmgr) 可以設定持續性所需的模式。您也可以使用 imqdbmgr 指令來刪除毀損的 Message Queue 資料庫表格，或變更資料儲存。

語法

```
imqdbmgr subcommand argument [options]
imqdbmgr -h|-help
imqdbmgr -v|-version
```

子指令

資料庫管理員公用程式 (imqdbmgr) 包括表 13-13 中列出的子指令：

表 13-13 imqdbmgr 子指令

子指令和引數	說明
create all	建立新資料庫和 Message Queue 永久性儲存模式。此指令可在內嵌式資料庫系統上使用，當使用此指令時，需要指定特性 <code>imq.persist.jdbc.createdburl</code> 。
create tbl	在現有資料庫系統中建立 Message Queue 永久性儲存模式。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 Message Queue 資料庫表格。
delete oldtbl	刪除所有較早版本之永久性儲存資料庫中的 Message Queue 資料庫表格。在自動將永久性儲存轉移到 Message Queue 目前版本後使用。
recreate tbl	在目前永久性儲存資料庫中刪除現有 Message Queue 資料庫表格，然後重新建立 Message Queue 永久性儲存模式。
reset lck	重設鎖定狀態，以便其他程序可以使用永久性儲存資料庫。

指令選項

表 13-14 列出了 imqdbmgr 指令的選項。

表 13-14 imqdbmgr 選項

選項	說明
<code>-Dproperty=value</code>	將指定的特性設定為指定的值。
<code>-b instanceName</code>	指定代理程式實例名稱並使用相應的實例配置檔案。
<code>-h</code>	顯示用法說明。不執行指令行上的其他任何指令。
<code>-p password</code>	指定資料庫密碼。這個選項已經被停用，未來的版次不支援。請使用下列其中一種替代方案： <ul style="list-style-type: none"> 在指令行上省略密碼，讓指令提示您輸入密碼。 使用 <code>-passfile</code> 選項，指定包含資料庫密碼的檔案。
<code>-passfile path</code>	指定包含資料庫密碼的檔案路徑。如需更多資訊，請參閱第 155 頁「使用密碼檔案」。
<code>-u name</code>	指定資料庫使用者名稱。

表 13-14 imqdbmgr 選項 (續)

選項	說明
-v	顯示版本資訊。不執行指令行上的其他任何指令。

另請參閱

如需有關設定永久性儲存的詳細資訊，請參閱第 93 頁「設定永久性儲存」。

imqusermgr

使用者管理員公用程式 (imqusermgr) 可讓您編輯或寫入文字檔案使用者儲存庫。使用 imqusermgr 前，請注意以下事項：

- 如果代理程式特定的使用者儲存庫不存在，您必須啟動對應的代理程式實例來建立儲存庫。
- 必須在代理程式安裝的主機上執行 imqusermgr 指令。
- 您需要寫入儲存庫的適當許可權：也就是說，在 Solaris 和 Linux 上，您必須為超級使用者或為第一個建立代理程式實例的使用者。

語法

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

子指令

表 13-15 列出本章中描述的 imqusermgr 子指令。

表 13-15 imqusermgr 子指令

子指令	說明
add [-i <i>instanceName</i>] -u <i>userName</i> -p <i>passwd</i> [-g <i>group</i>] [-s]	將使用者和相關密碼新增到指定的 (或預設的) 代理程式實例儲存庫，並選擇性地指定使用者的群組。
delete [-i <i>instanceName</i>] -u <i>userName</i> [-s] [-f]	從指定的 (或預設的) 代理程式實例儲存庫刪除指定的使用者。
list [-i <i>instanceName</i>] [-u <i>userName</i>]	顯示有關指定的 (或預設的) 代理程式實例儲存庫中，指定的使用者或所有使用者的資訊。
update [-i <i>instanceName</i>] -u <i>userName</i> -p <i>passwd</i> [-a <i>state</i>] [-s] [-f]	更新指定的 (或預設的) 代理程式實例儲存庫中，指定之使用者的密碼和 / 或狀態。
update [-i <i>instanceName</i>] -u <i>userName</i> -a <i>state</i> [-p <i>passwd</i>] [-s] [-f]	

指令選項

表 13-16 列出了 imqusermgr 指令的選項。

表 13-16 imqusermgr 選項

選項	說明
-a <i>active_state</i>	指定使用者的狀態是否應為使用中 (true/false)。值 true 表示狀態為使用中。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。
-i <i>instanceName</i>	指定代理程式實例使用者儲存庫套用的指令。如果未指定，則假設為預設實例名稱值 imqbroker。
-p <i>passwd</i>	指定使用者密碼。
-g <i>group</i>	指定使用者群組。有效值為 admin、user 和 anonymous。
-s	設定無訊息模式。

表 13-16 imqusermgr 選項 (續)

選項	說明
-u <i>userName</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。

另請參閱

如需有關設定及管理使用文字檔案使用者儲存庫的詳細資訊和 `imqusermgr` 指令範例，請參閱第 132 頁「[使用文字檔案使用者儲存庫](#)」。

imqsvcadmin

服務管理 (`imqsvcadmin`) 公用程式可以將代理程式安裝為 Windows 服務。

語法

```
imqsvcadmin subcommand [options]
```

```
imqsvcadmin -h
```

子指令

Message Queue 服務管理員公用程式 (`imqsvcadmin`) 包括表 13-17 中列出的子指令：

表 13-17 imqsvcadmin 子指令

子指令	說明
<code>install</code>	安裝服務並指定啟動選項。
<code>query</code>	顯示 <code>imqsvcadmin</code> 指令的啟動選項。這包括服務是手動啟動還是自動啟動、服務的位置、Java 執行階段的位置以及啟動時傳送至代理程式的引數的值。
<code>remove</code>	移除服務。

指令選項

表 13-18 列出 imqsvcadmin 指令的選項。

表 13-18 imqsvcadmin 選項

選項	說明
-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>	指定要使用的替代 Java 2 相容執行階段的路徑 (預設為使用系統上的執行階段，或 Message Queue 隨附的執行階段)。 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 選項的說明，請參閱第 257 頁「 imqbrokerd 」。 (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 選項時指定的資訊，儲存在 Window 登入檔中的鍵值 JREHome、JVMArgs 和 ServiceArgs 下，路徑為：

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet
\Services\IMQ_Broker\Parameters
```

另請參閱

如需有關將 Message Queue 當做 Windows 服務來執行的詳細資訊，請參閱第 66 頁「[在 Windows 自動啟動](#)」。

imqkeytool

`imqusermgr` 指令可以為代理程式產生自身簽名的憑證。同樣的憑證可以用於 `ssljms`、`ssladmin` 或 `cluster` 連線服務。若是 UNIX 系統，您可能需要從 `superuser` (超級使用者) 帳號，執行 `imqkeytool`。

語法

```
imqkeytool -broker
```

另請參閱

如需有關設定安全連線的詳細資訊，請參閱第 146 頁「[使用基於 SSL 的服務](#)」。

imqkeytool

代理程式特性參照

本章列出並描述代理程式配置特性。第一節按字母順序列出所有的代理程式特性，並且參照有完整說明的章節。其他章節則按功能將代理程式特性分組，並且提供完整的特性說明。

本章包含以下各節：

- [第 282 頁](#) 「按字母順序排列的特性清單」
- [第 285 頁](#) 「連線服務特性」
- [第 287 頁](#) 「訊息路由器特性」
- [第 289 頁](#) 「持續性管理程式特性」
- [第 293 頁](#) 「安全性管理程式特性」
- [第 297 頁](#) 「監視和記錄特性」
- [第 300 頁](#) 「叢集配置特性」

說明表格中會標出可以用指令設定的 `imqcmd update bkr` 特性。

按字母順序排列的特性清單

表 14-1 是按字母順序排列的代理程式實例特性清單。您可以用此清單判斷出任何特性的種類，然後根據種類說明，在本章別處找到完整的特性說明。

表格左欄按字母順序列出每個特性。右欄顯示特性的所屬種類，並提供適當章節的交叉參照。

表 14-1 代理程式實例配置特性

特性名稱	參照
<code>imq.accesscontrol.enabled</code>	第 293 頁 「安全性管理程式特性」
<code>imq.accesscontrol.file.filename</code>	第 293 頁 「安全性管理程式特性」
<code>imq.audit.enabled</code>	第 293 頁 「安全性管理程式特性」
<code>imq.authentication.basic.user_repository</code>	第 293 頁 「安全性管理程式特性」
<code>imq.authentication.client.response.timeout</code>	第 293 頁 「安全性管理程式特性」
<code>imq.authentication.type</code>	第 293 頁 「安全性管理程式特性」
<code>imq.autocreate.destination.isLocalOnly</code>	第 288 頁 「自動建立配置特性」
<code>imq.autocreate.destination.limitBehavior</code>	第 288 頁 「自動建立配置特性」
<code>imq.autocreate.destination.maxBytesPerMsg</code>	第 288 頁 「自動建立配置特性」
<code>imq.autocreate.destination.maxNumMsgs</code>	第 288 頁 「自動建立配置特性」
<code>imq.autocreate.destination.maxNumProducers</code>	第 288 頁 「自動建立配置特性」
<code>imq.autocreate.destination.maxTotalMsgBytes</code>	第 288 頁 「自動建立配置特性」
<code>imq.autocreate.destination.useDMQ</code>	第 288 頁 「自動建立配置特性」
<code>imq.autocreate.queue</code>	第 288 頁 「自動建立配置特性」
<code>imq.autocreate.queue.consumerFlowLimit</code>	第 288 頁 「自動建立配置特性」
<code>imq.autocreate.queue.localDeliveryPreferred</code>	第 288 頁 「自動建立配置特性」
<code>imq.autocreate.queue.maxNumActiveConsumers</code>	第 288 頁 「自動建立配置特性」
<code>imq.autocreate.queue.maxNumBackupConsumers</code>	第 288 頁 「自動建立配置特性」
<code>imq.autocreate.topic</code>	第 288 頁 「自動建立配置特性」
<code>imq.autocreate.topic.consumerFlowLimit</code>	第 288 頁 「自動建立配置特性」
<code>imq.cluster.<i>property_name</i></code>	第 300 頁 「叢集配置特性」
<code>imq.destination.DMQ.truncateBody</code>	第 287 頁 「訊息路由器特性」
<code>imq.destination.logDeadMsgs</code>	第 297 頁 「監視和記錄特性」

表 14-1 代理程式實例配置特性 (續)

特性名稱	參照
imq.hostname	第 285 頁「連線服務特性」
imq.httpjms.http. <i>property_name</i>	第 355 頁表 C-3
imq.httpsjms.https. <i>property_name</i>	第 355 頁表 C-3
imq.imqcmd.password	第 293 頁「安全性管理程式特性」
imq.keystore. <i>property_name</i>	第 293 頁「安全性管理程式特性」
imq.log.console.output	第 297 頁「監視服務特性」
imq.log.console.stream	第 297 頁「監視服務特性」
imq.log.file.dirpath	第 297 頁「監視服務特性」
imq.log.file.filename	第 297 頁「監視服務特性」
imq.log.file.output	第 297 頁「監視服務特性」
imq.log.file.rolloverbytes	第 297 頁「監視服務特性」
imq.log.file.rolloversecs	第 297 頁「監視服務特性」
imq.log.level	第 297 頁「監視服務特性」
imq.log.syslog.facility	第 297 頁「監視服務特性」
imq.log.syslog.identity	第 297 頁「監視服務特性」
imq.log.syslog.logconsole	第 297 頁「監視服務特性」
imq.log.syslog.logpid	第 297 頁「監視服務特性」
imq.log.syslog.output	第 297 頁「監視服務特性」
imq.log.timezone	第 297 頁「監視服務特性」
imq.message.expiration.interval	第 287 頁「訊息路由器特性」
imq.message.max_size	第 287 頁「訊息路由器特性」
imq.metrics.enabled	第 287 頁「訊息路由器特性」
imq.metrics.interval	第 287 頁「訊息路由器特性」
imq.metrics.topic.enabled	第 287 頁「訊息路由器特性」
imq.metrics.topic.interval	第 287 頁「訊息路由器特性」
imq.metrics.topic.persist	第 287 頁「訊息路由器特性」
imq.metrics.topic.timetolive	第 297 頁「監視服務特性」
imq.passfile.dirpath	第 293 頁「安全性管理程式特性」
imq.passfile.enabled	第 293 頁「安全性管理程式特性」

表 14-1 代理程式實例配置特性 (續)

特性名稱	參照
<code>imq.passfile.name</code>	第 293 頁 「安全性管理程式特性」
<code>imq.persist.file.destination.message.filepool.limit</code>	第 290 頁 「基於檔案的持續性的特性」
<code>imq.persist.file.message.cleanup</code>	第 287 頁 「訊息路由器特性」
<code>imq.persist.file.message.filepool.cleanratio</code>	第 287 頁 「訊息路由器特性」
<code>imq.persist.file.message.max_record_size</code>	第 287 頁 「訊息路由器特性」
<code>imq.persist.file.sync.enabled</code>	第 290 頁 「基於檔案的持續性的特性」
<code>imq.persist.jdbc.property_name</code>	第 289 頁 「持續性管理程式特性」
<code>imq.persist.store</code>	第 287 頁 「訊息路由器特性」
<code>imq.ping.interval</code>	第 285 頁 「連線服務特性」
<code>imq.portmapper.backlog</code>	第 285 頁 「連線服務特性」
<code>imq.portmapper.hostname</code>	第 285 頁 「連線服務特性」
<code>imq.portmapper.port</code>	第 285 頁 「連線服務特性」
<code>imq.resource_state.count</code>	第 287 頁 「訊息路由器特性」
<code>imq.resource_state.threshold</code>	第 287 頁 「訊息路由器特性」
<code>imq.service.activelist</code>	第 285 頁 「連線服務特性」
<code>imq.service_name.accesscontrol.enabled</code>	第 293 頁 「安全性管理程式特性」
<code>imq.service_name.accesscontrol.file.filename</code>	第 293 頁 「安全性管理程式特性」
<code>imq.service_name.authentication.type</code>	第 293 頁 「安全性管理程式特性」
<code>imq.service_name.max_threads</code>	第 285 頁 「連線服務特性」
<code>imq.service_name.min_threads</code>	第 285 頁 「連線服務特性」
<code>imq.service_name.protocol_type.hostname</code>	第 285 頁 「連線服務特性」
<code>imq.service_name.protocol_type.port</code>	第 285 頁 「連線服務特性」
<code>imq.service_name.threadpool_model</code>	第 285 頁 「連線服務特性」
<code>imq.shared.connectionMonitor_limit</code>	第 285 頁 「連線服務特性」
<code>imq.system.max_count</code>	第 287 頁 「訊息路由器特性」
<code>imq.system.max_size</code>	第 287 頁 「訊息路由器特性」
<code>imq.transaction.autorollback</code>	第 287 頁 「訊息路由器特性」
<code>imq.user_repository.ldap.property_name</code>	第 293 頁 「安全性管理程式特性」

連線服務特性

表 14-2 列出連線服務特性。第一欄列出特性名稱。每個特性名稱的第二欄描述此特性，第三欄指定特性的類型，第四欄提供特性的預設值。

表 14-2 連線服務特性

特性名稱	說明	類型	預設
<code>imq.service.activelist</code>	要在啟動代理程式時變為使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> 。	清單	<code>jms</code> 、 <code>admin</code>
<code>imq.ping.interval</code>	代理程式透過連線，連續嘗試偵測 Message Queue 用戶端執行階段的期間（以秒為單位）。	整數	120
<code>imq.hostname</code>	如果有多台主機可用（例如，如果一台電腦中有多張網路介面卡），則為所有連線服務所連結的主機（主機名稱或 IP 位址）。	字串	所有可用的 IP 位址
<code>imq.portmapper.port¹</code>	代理程式的主要連接埠 <i>n</i> 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。	整數	7676
<code>imq.portmapper.hostname</code>	如果有多台主機可用（例如，如果一台電腦中有多張網路介面卡），則為連接埠對映器所連結的主機（主機名稱或 IP 位址）。	字串	繼承 <code>imq.hostname</code>
<code>imq.portmapper.backlog</code>	拒絕請求前，連接埠對映器可以處理運作請求的最大數目。此特性可以設定請求數目，另外，這些請求可儲存在作業系統儲存區，等待連接埠對映器進行處理。	整數	50
<code>imq.service_name.protocol_type².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的連接埠號。 若要配置 <code>httpjms</code> 和 <code>httpsjms</code> 連線服務，請參閱附錄 C 「HTTP/HTTPS 支援」	整數	0（零） 連接埠對映器動態地配置連接埠。
<code>imq.service_name.protocol_type².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張網路介面卡），則為已命名的連線服務所連結的主機（主機名稱或 IP 位址）。	字串	繼承 <code>imq.hostname</code>

表 14-2 連線服務特性 (續)

特性名稱	說明	類型	預設
imq.service_name. min_threads	執行緒數目，一旦達到指定數目，執行緒便會在執行緒池中維護，供已命名的連線服務使用。 預設值因連線服務而異。	整數	10 (jms)
			10 (ssljms)
			10 (httpjms)
			10 (httpsjms)
			4 (admin)
			4 (ssladmin)
imq.service_name. max_threads	執行緒數目，一旦超過指定數目，系統便不會將新的執行緒新增至執行緒池，供已命名的連線服務使用。此數目必須大於零，並且其值必須大於 min_threads 的值。 預設值因連線服務而異。	整數	1000 (jms)
			500 (ssljms)
			500 (httpjms)
			500 (httpsjms)
			10 (admin)
			10 (ssladmin)
imq.service_name. threadpool_model	一個字串，指定執行緒是專屬於連線 (dedicated) 或依需要由連線共用 (shared)，以用於已命名的連線服務。共用模型 (執行緒池管理) 會增加代理程式所支援的連線數目，但僅實作用於 jms 和 admin 連線服務。 預設值因連線服務而異。	字串	dedicated (jms)
			dedicated (ssljms)
			dedicated (httpjms)
			dedicated (httpsjms)
			dedicated (admin)
			dedicated (ssladmin)
imq.shared. connectionMonitor_limit	僅用於共用執行緒池模型，可由分散執行緒監視的最大連線數目。(系統配置了足夠的分散執行緒以監視所有連線。) 此值越小，系統將使用中的連線指定給執行緒的速度就越快。值 -1 表示沒有限制。 預設值因作業系統而異。	整數	512 (Solaris 和 Linux)
			64 (Windows)

1. 這個特性可用於 imqcmd update bkr 指令。
2. protocol_type (協定類型) 在表 4-2 中指定。

訊息路由 器特性

表 14-3 列出訊息路由器特性。第一欄列出特性名稱。每個特性名稱的第二欄描述此特性，第三欄指定特性的類型，第四欄提供特性的預設值。

第 288 頁表 14-4 中列出能配置訊息伺服器自動建立目標的自動建立特性。

表 14-3 訊息路由器特性

特性名稱	說明	類型	預設
imq.destination. DMQ.truncateBody ¹	這個布林值指定代理程式能否先移除訊息內文，然後才儲存至停用的訊息佇列中。值 <code>true</code> 表示代理程式只儲存訊息標頭和特性資料。值 <code>false</code> 表示代理程式儲存標頭和內文。	布林	false
imq.message. expiration.interval	過期訊息的回收間隔 (以秒為單位)。	整數	60
imq.system.max_count ¹	代理程式保留訊息的最大數目。其他訊息將被拒絕。值 -1 表示沒有限制。	整數	-1
imq.system.max_size ¹	代理程式保留訊息的最大總容量 (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 -1 表示沒有限制。	位元組字串 ²	-1
imq.message.max_size ¹	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)。任何大於該設定的訊息將被拒絕。值 -1 表示沒有限制。	位元組字串 ²	70m
imq.resource_state. threshold	啟動每種記憶體資源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> 。	整數 (百分比)	0 (green) 80 (yellow) 90 (orange) 98 (red)
imq.resource_state.count	批次中允許內送的訊息的最大數目，之後則會檢查系統記憶體是否已達新的記憶體臨界值。系統記憶體變得逐漸不足時，此限制會減低訊息產生者。	整數 (百分比)	5000 (green) 500 (yellow) 50 (orange) 0 (red)
imq.transaction. autorollback	這個布林值指定處於 <code>PREPARED</code> 狀態的分散式作業事件是否可在啟動代理程式時自動回轉。如果為 <code>false</code> ，則您必須使用 <code>imqcmd</code> 手動確定或回轉作業事件 (請參閱第 116 頁「管理作業事件」)。	布林	false

1. 這個特性可用於 `imqcmd update bkr` 指令。

2. 以位元組字串方式輸入的值，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 14-4 列出代理程式自動建立目標時使用的特性。

表 14-4 自動建立配置特性

特性名稱	說明	類型	預設
<code>imq.autocreate.destination.isLocalOnly</code>	(僅套用於代理程式叢集。) 這個布林值指定在其他代理程式上不重複目標，且限制目標僅傳送訊息給本地用戶(連線到目標上所建立之代理程式的用戶)。一旦建立目標，即無法更新屬性。	布林	false
<code>imq.autocreate.destination.limitBehavior</code>	<p>這個字串指定達到記憶體限制臨界值時，代理程式回應的方式。值為：</p> <ul style="list-style-type: none"> • <code>FLOW_CONTROL</code> - 減緩產生者速度。 • <code>REMOVE_OLDEST</code> - 拋出最舊的訊息。 • <code>REMOVE_LOW_PRIORITY</code> - 根據訊息存在時間拋出最後順位的訊息。 • <code>REJECT_NEWEST</code> - 拒絕最新的訊息。僅在拒絕永久性訊息時，生產型用戶端會發生異常。若要將這個限制運作方式用在非永久性訊息上，請設定 <code>imqAckOnProduce</code> 連線工廠屬性。 <p>如果您將這個特性設定為 <code>REMOVE_OLDEST</code> 或 <code>REMOVE_LOW_PRIORITY</code>，並將 <code>imq.autocreate.destination.useDMQ</code> 設定為 <code>true</code>，代理程式會將超過此限制的訊息移動至停用的訊息佇列。</p>	字串	REJECT NEWEST
<code>imq.autocreate.destination.maxBytesPerMsg</code>	自動建立的目標中允許的任何單一訊息的最大容量(以位元組為單位)。值 -1 表示不限制訊息大小。	位元組 字串 ²	10k
<code>imq.autocreate.destination.maxNumMsgs</code>	自動建立的目標中，允許給未使用訊息的最大數目。值 -1 表示不限制數目。	整數	100,000
<code>imq.autocreate.destination.maxNumProducers</code>	允許給目標中產生者的最大數目。達到限制時，將無法建立新的產生者。值 -1 表示不限制產生者的數目。	整數	100
<code>imq.autocreate.destination.maxTotalMsgBytes</code>	目標中允許給未使用訊息的最大總記憶體容量(以位元組為單位)。值 -1 表示不限制記憶體。	位元組 字串 ²	10m
<code>imq.autocreate.destination.useDMQ</code>	這個布林值指定代理程式是否將自動建立目標的停用訊息，移動至停用的訊息佇列。	布林	true
<code>imq.autocreate.queue¹</code>	這個布林值指定是否允許代理程式自動建立佇列目標。	布林	true

表 14-4 自動建立配置特性 (續)

特性名稱	說明	類型	預設
imq.autocreate.queue.consumerFlowLimit	最大訊息數目，這些訊息將會傳送給單一批次中的某個用戶。在負載平衡佇列傳送中，負載平衡開始前，此數目是路由到使用中用戶的佇列訊息初始數目。目標的用戶在其個別連線上，可設定一個較低的值來覆寫此限制。值 -1 表示沒有限制數目。	整數	1000
imq.autocreate.queue.localDeliveryPreferred	(僅套用於代理程式叢集中的負載平衡佇列傳送。) 這個布林值指定如果本地代理程式沒有用戶，則訊息傳送給遠端用戶。要求自動建立的目標不僅限於本地傳送 (isLocalOnly = false)。	布林	false
imq.autocreate.queue.maxNumActiveConsumers	最大用戶數，此數值可作用於來自自動建立之佇列目標的負載平衡傳送。值 -1 表示沒有限制數目。	整數	1
imq.autocreate.queue.maxNumBackupConsumers	最大備份用戶數，如果無法從自動建立的佇列目標進行負載平衡傳送，那麼這些用戶可以取代使用中的用戶。值 -1 表示沒有限制數目。	整數	0 (零)
imq.autocreate.topic	這個布林值指定是否允許代理程式自動建立主題目標。	布林	true
imq.autocreate.topic.consumerFlowLimit	最大訊息數目，這些訊息將會傳送給單一批次中的某個用戶。值 -1 表示沒有限制數目。	整數	1000

1. 這個特性可用於 `imqcmd update bkr`。
2. 以位元組字串方式輸入的值，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)

持續性管理程式特性

若要配置代理程式的持續性特性，您必須指定 `imq.persist.store` 值或接受預設值。

表 14-5 必要的持續性管理程式特性

特性名稱	說明	類型	預設
imq.persist.store	這個字串指定代理程式使用基於檔案的內建持續性，或使用相容的外掛 JDBC 持續性。 此值必須是 <code>file</code> 或 <code>jdbc</code> 。	字串	file

下面幾節會描述支援基於檔案的持續性和基於 JDBC 的持續性的特性。

基於檔案的持續性

表 14-6 中列出支援基於檔案的持續性的特性。第一欄列出特性名稱。每個特性名稱的第二欄描述此特性，第三欄指定特性的類型，第四欄提供特性的預設值。

表 14-6 基於檔案的持續性的特性

特性名稱	說明	類型	預設
<code>imq.persist.file.sync.enabled</code>	<p>這個布林值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。如果這個特性設定為 <code>true</code>，則會結束由系統當機引起的資料遺失，但會降低持續性作業的效能。</p> <p>如果您為 Message Queue 執行 Sun Cluster 和 Sun Cluster Data Service，請在所有節點上將代理程式的此特性設定為 <code>true</code>。</p>	布林	<code>false</code>
<code>imq.persist.file.message.max_record_size</code>	針對內建且基於檔案的持續性，將要新增到訊息儲存檔案（而非儲存到個別檔案）的訊息最大容量。	位元組字串 ¹	1m
<code>imq.persist.file.destination.message.filepool.limit</code>	針對內建且基於檔案的持續性，目標檔案池中，可供重新使用之可用檔案的最大數目。此數目越大，代理程式處理永久性資料的速度就越快。超過此數目的可用檔案將被刪除。代理程式將依需要建立與刪除超過此限制的其他檔案。	整數	100
<code>imq.persist.file.message.filepool.cleanratio</code>	針對內建且基於檔案的持續性，目標檔案池中被維護處於清除狀態之可用檔案的百分比（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，但檔案池所需的磁碟空間就越少。	整數	0（零）
<code>imq.persist.file.message.cleanup</code>	針對內建且基於檔案的持續性，這個布林值指定代理程式關閉時，是否會清除目標檔案池中的可用檔案。值 <code>false</code> 會加速代理程式的關閉速度，但檔案儲存需要更多的磁碟空間。	布林	<code>false</code>

1. 以位元組字串方式輸入的值，可以用位元組、千位元組和百萬位元組表示。範例：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）

基於 JDBC 的持續性

表 14-7 包含支援基於 JDBC 的持續性的特性。此表列出特性、描述特性，並且提供範例說明如何配置使用 PointBase 產品。

表 14-7 基於 JDBC 的持續性的特性

特性名稱	說明	範例
<code>imq.persist.store</code>	這個字串指定基於檔案或基於 JDBC 的資料儲存。	<code>jdbc</code>
<code>imq.persist.jdbc.brokerid</code>	<p>(選用) 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的情況下，這些資料庫表格名稱均是唯一的。</p> <p>由於內嵌式資料庫僅為一個代理程式實例儲存資料，因此通常不需要此屬性。</p> <p>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。</p>	(PointBase 內嵌式版本不需要)
<code>imq.persist.jdbc.driver</code>	JDBC 驅動程式的 Java 類別名稱，用來連線至資料庫。	<code>com.pointbase.jdbc.jdbcUniversalDriver</code>
<code>imq.persist.jdbc.opendburl</code>	資料庫 URL，用來與現有資料庫開啓連線。	<code>jdbc:pointbase:embedded:dbName;database.home=.../instances/instanceName/dbstore</code>
<code>imq.persist.jdbc.createdburl</code>	<p>(選用) 資料庫 URL，用來開啓連線建立資料庫。</p> <p>使用 <code>imqdbmgr</code> 建立資料庫時，才指定這個屬性。</p>	<code>jdbc:pointbase:embedded:dbName;new, database.home=.../instances/instanceName/dbstore</code>
<code>imq.persist.jdbc.closedburl</code>	(選用) 資料庫 URL，用來在關閉代理程式時關閉目前資料庫連線。	PointBase 不需要
<code>imq.persist.jdbc.user</code>	<p>(選用) 必要時，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：</p> <p><code>imqbrokerd -dbuser</code> 和 <code>imqdbmgr -u</code></p>	

表 14-7 基於 JDBC 的持續性的特性 (續)

特性名稱	說明	範例
imq.persist.jdbc.needpassword	(選用) 這個布林值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。 如果設定這個選項，imqbrokerd 和 imqdbmgr 指令會提示您輸入密碼，除非您使用 -passfile 選項來指定包含密碼的檔案。	
imq.persist.jdbc.password	(選用) 必要時，用於開啓資料庫連線的密碼。 只在密碼檔案中指定這個特性。	
imq.persist.jdbc.table.IMQSV35	用來建立版本表格的 SQL 指令。	CREATE TABLE \${name} (STOREVERSION INTEGER NOT NULL, BROKERID VARCHAR(100))
imq.persist.jdbc.table. IMQCCREC35	用來建立配置變更記錄表格的 SQL 指令。	CREATE TABLE \${name} (RECORDTIME BIGINT NOT NULL, RECORD BLOB(10k))
imq.persist.jdbc.table. IMQDEST35	用來建立目標表格的 SQL 指令。	CREATE TABLE \${name} (DID VARCHAR(100) NOT NULL, DEST BLOB(10k), primary key(DID))
imq.persist.jdbc.table.IMQINT35	用來建立偏好表格的 SQL 指令。	CREATE TABLE \${name} (CUID BIGINT NOT NULL, INTEREST BLOB(10k), primary key(CUID))
imq.persist.jdbc.table.IMQMSG35	用來建立訊息表格的 SQL 指令。 MSG 的預設最大欄寬為 1 Megabyte (1m)。如果您預期訊息 長度會大於此長度，請相應地設定 長度。如果已建立表格，您則必須 重新建立表格來進行變更。	CREATE TABLE \${name} (MID VARCHAR(100) NOT NULL, DID VARCHAR(100), MSGSIZE BIGINT, MSG BLOB(1m), primary key(MID))
imq.persist.jdbc.table. IMQPROPS35	用來建立特性表格的 SQL 指令。	CREATE TABLE \${name} (PROPNAME VARCHAR(100) NOT NULL, PROPVALUE BLOB(10k), primary key(PROPNAME))

表 14-7 基於 JDBC 的持續性的特性 (續)

特性名稱	說明	範例
imq.persist.jdbc.table.IMQILIST35	用來建立偏好狀態表格的 SQL 指令。	CREATE TABLE \${name} (MID VARCHAR(100) NOT NULL, CUID BIGINT, DID VARCHAR(100), STATE INTEGER, primary key(MID, CUID))
imq.persist.jdbc.table.IMQTXN35	用來建立作業事件表格的 SQL 指令。	CREATE TABLE \${name} (TUID BIGINT NOT NULL, STATE INTEGER, TSTATEOBJ BLOB(10K), primary key(TUID))
imq.persist.jdbc.table.IMQTACK35	用來建立作業事件確認表格的 SQL 指令。	CREATE TABLE \${name} (TUID BIGINT NOT NULL, TXNACK BLOB(10k))

安全性管理程式特性

表 14-8 列出安全管理程式特性。第一欄列出特性名稱。每個特性名稱的第二欄描述此特性，第三欄指定特性的類型，第四欄提供特性的預設值。

如果您使用 SSL，請參照後面表 14-9 中列出的密鑰儲存配置特性。

表 14-8 安全管理程式特性

特性名稱	說明	類型	預設
imq.accesscontrol.enabled	這個布林值指定是否為代理程式支援的所有連線服務，設定存取控制。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 Message Queue 作業的許可權 (如存取控制特性檔案中所指定)。	布林	true
imq.accesscontrol.file.filename	代理程式實例支援的所有連線服務的存取控制特性檔案的名稱。檔案名稱可指出到存取控制目錄的相關檔案路徑 (請參閱附錄 A 「Message Queue 資料的作業系統特定位置」)。	字串	accesscontrol.properties
imq.audit.enabled	這個布林值指定是否啟動代理程式記錄檔的稽核記錄功能 (僅 Enterprise Edition)。	布林	false

表 14-8 安全性管理程式特性 (續)

特性名稱	說明	類型	預設
imq.authentication.basic.user_repository	(用於基本 64 程式碼) 這個字串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)。	字串	file
imq.authentication.client.response.timeout	系統等待用戶端回應來自代理程式的認證請求的間隔時間 (以秒為單位)。	整數	180
imq.authentication.type	這個字串指定密碼應以基本 64 程式碼 (basic) 傳送，或作為 MD5 摘要 (digest) 傳送。為代理程式支援的所有連線服務設定編碼。	字串	digest
imq.imqcmd.password	管理使用者的密碼。imqcmd 指令公用程式先使用這個密碼認證指令使用者，然後才執行作業。	字串	無
imq.keystore.property_name	針對基於 SSL 的服務，這個字串指定 SSL 密鑰儲存的相關安全性特性。請參閱第 297 頁表 14-9	字串	無
imq.passfile.dirpath	包含密碼檔案之目錄的路徑 (因作業系統而異)。	字串	請參閱附錄 A
imq.passfile.enabled	這個布林值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™)。	布林	false
imq.passfile.name	密碼檔案的名稱。	字串	密碼檔案
imq.service_name.accesscontrol.enabled	這個布林值指定是否設定存取控制已命名的連線服務，覆寫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 Message Queue 作業的許可權 (如存取控制特性檔案中所指定)。	布林	繼承系統範圍的特性 imq.accesscontrol.enabled.
imq.service_name.accesscontrol.file.filename	代理程式實例的已命名連線服務的存取控制特性檔案的名稱。檔案名稱可指出到存取控制目錄的相關檔案路徑 (請參閱附錄 A 「Message Queue 資料的作業系統特定位置」)。 預設值繼承系統範圍的特性 imq.accesscontrol.file.filename	字串	請參閱說明
imq.service_name.authentication.type	這個字串指定密碼應以基本 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，並覆寫任何代理程式範圍的設定。 預設值繼承系統範圍的特性 imq.authentication.type。	字串	請參閱說明
imq.user_repository.ldap.base	用於使用者項目的目錄庫。	字串	無

表 14-8 安全性管理程式特性 (續)

特性名稱	說明	類型	預設
imq.user_repository. ldap.gidattr	供應程式特定的屬性識別碼，其值為群組名稱。	字串	無
imq.user_repository. ldap.grpbase	用於群組項目的目錄庫。	字串	無
imq.user_repository. ldap.grpfiltler	JNDI 搜尋篩選器 (表示為邏輯表示式的搜尋查詢)。透過為群組指定搜尋篩選器，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 此特性可不必設定。	字串	無
imq.user_repository. ldap.grpsearch	這個布林值指定您是否要啟用群組搜尋。請參閱您的 LDAP 供應商提供的文件，以確定您是否可以將使用者與群組相關聯。 請注意，Message Queue 不支援巢式群組。	布林	false
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。	字串	無
imq.user_repository. ldap.password	與代理程式所用識別名稱相關聯的密碼。 只在密碼檔案中指定這個特性。 如果目錄伺服器允許匿名搜尋，則無需密碼。	字串	無
imq.user_repository. ldap.principal	識別名稱，代理程式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，則無需為此特性指定值。	字串	無
imq.user_repository. ldap.property_name	等待支援	等待支援	等待支援

表 14-8 安全性管理程式特性 (續)

特性名稱	說明	類型	預設
imq.user_repository. ldap.server	<p>用於 LDAP 伺服器的 <i>host:port</i>。host 可指定執行目錄伺服器之主機的完全合格的 DNS 名稱，而 <i>port</i> 指定用於通訊之目錄伺服器的連接埠號。</p> <p>若要指定防故障備用伺服器的清單，請使用下列語法：</p> <pre>host1:port1 ldap://host2:port2 ldap://host3:port3...</pre> <p>清單中的項目以空格分開。請注意，每個防故障備用伺服器位址是以 ldap:// 開頭。</p> <p>即使您使用 SSL 並且將特性 imq.user_repository.ldap.ssl.enabled 設定為 true，也請使用這個格式。您不必在位址中指定「ldaps」。</p>	字串	無
imq.user_repository. ldap.ssl.enabled	這個布林值指定代理程式與 LDAP 通訊時是否應使用 SSL 協定。	布林	false
imq.user_repository. ldap.timeout	搜尋的時間限制 (以秒為單位)。	整數	280
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。	字串	無
imq.user_repository. ldap.usrfilter	<p>JNDI 搜尋篩選器 (表示為邏輯表示式的搜尋查詢)。透過為使用者指定搜尋篩選器，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。</p> <p>此特性可不必設定。</p>	字串	無

Message Queue 密鑰儲存的可配置特性如表 14-9 中所示。使用這些特性可以用於 SSL。

表 14-9 密鑰儲存特性

特性名稱	說明	類型	預設
imq.keystore.file.dirpath	基於 SSL 的服務的密鑰儲存檔案所在目錄的路徑。預設值：請參閱附錄 A「Message Queue 資料的作業系統特定位置」。	字串	無
imq.keystore.file.name	用於基於 SSL 的服務：密鑰儲存檔案的名稱。	字串	密鑰儲存
imq.keystore.password	用於基於 SSL 的服務：密鑰儲存密碼。 只在密碼檔案中指定這個特性。	字串	無

監視和記錄特性

表 14-10 列出監視和記錄的相關特性。第一欄列出特性名稱。每個特性名稱的第二欄描述此特性，第三欄指定特性的類型，第四欄提供特性的預設值。

表 14-10 監視服務特性

特性名稱	說明	類型	預設
imq.destination.logDeadMsgs ¹	這個布林值指定代理程式是否記錄下列類型的事件： <ul style="list-style-type: none"> 目標已滿，已達最大容量或最大訊息計數。 代理程式不是因為管理指令或傳送確認而丟棄訊息。 代理程式將訊息移動到停用的訊息佇列。 	布林	false
imq.log.console.output	這個字串指定寫入至主控台的記錄資訊的種類。可能是下列其中一值： <ul style="list-style-type: none"> ALL NONE 下列一或多個值，中間以垂直列 () 區隔：ERROR、WARNING 和 INFO。分別指定每個記錄訊息的種類。一個訊息種類不會包含其他訊息種類。 	字串	ERROR WARNING
imq.log.console.stream	這個字串指定主控台輸出寫入至 stdout (OUT)，或寫入至 stderr (ERR)。	字串	ERR

表 14-10 監視服務特性 (續)

特性名稱	說明	類型	預設
imq.log.file.dirpath	記錄檔所在目錄的路徑 (因作業系統而異)。	字串	請參閱附錄 A
imq.log.file.filename	記錄檔的名稱。	字串	log.txt
imq.log.file.output	要寫入到主控台的記錄資訊種類。可能是下列其中一值： <ul style="list-style-type: none"> • ALL • NONE • 下列一或多個值，中間以垂直列 () 區隔：ERROR、WARNING 和 INFO。分別指定每個記錄訊息的種類。一個訊息種類不會包含其他訊息種類。 	字串	ALL
imq.log.file.rolloverbytes ¹	記錄檔輸出自動重建至新記錄檔時，此記錄檔的容量 (以位元組為單位)。值 -1 表示停用基於檔案容量的自動重建。	整數	-1
imq.log.file.rolloversecs ¹	記錄檔輸出自動重建至新記錄檔時，此記錄檔的存在時間 (以秒為單位)。值 -1 表示停用基於檔案存在時間的自動重建。	整數	604800 (一週)
imq.log.level ¹	這個字串指定記錄程式級別：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。	字串	INFO
imq.log.syslog.facility	(僅適用於 Solaris) 這個字串指定 Message Queue 代理程式應記錄為的 syslog (系統記錄) 設備。值鏡射 syslog(3C) 線上手冊中列出的那些值。用於 Message Queue 的適當值為：LOG_USER、LOG_DAEMON 以及 LOG_LOCAL0 至 LOG_LOCAL7。	字串	LOG_DAEMON
imq.log.syslog.identity	(僅適用於 Solaris) 應前置於記錄至 syslog (系統記錄) 的每個訊息的身份字串。 預設值是 imqbrokerd_\${imq.instanceName}	字串	請參閱說明
imq.log.syslog.logconsole	(僅用於 Solaris) 這個布林值指定如果訊息無法傳送至 syslog (系統記錄)，是否將它們寫入至系統主控台。	布林	false
imq.log.syslog.logpid	(僅適用於 Solaris) 這個布林值指定是否同時記錄訊息與代理程式程序 ID (true/false)。	布林	true

表 14-10 監視服務特性 (續)

特性名稱	說明	類型	預設
imq.log.syslog.output	(僅適用於 Solaris) 這個字串指定寫入至 syslogd(1M) 的記錄資訊的種類。可能是下列其中一值： <ul style="list-style-type: none"> ALL NONE 下列一或多個值，中間以垂直列 () 區隔：ERROR、WARNING 和 INFO。分別指定每個記錄訊息的種類。一個訊息種類不會包含其他訊息種類。 	字串	ERROR
imq.log.timezone	這個字串代表記錄時間戳記的時區。這些識別碼與 java.util.TimeZone.getTimeZone() 使用的識別碼相同。例如：GMT、America/LosAngeles、Europe/Rome、Asia/Tokyo。	字串	本地時區
imq.metrics.enabled	這個布林值指定是否要將度量資訊寫入記錄程式。不會影響度量訊息的產生 (請參閱 imq.metrics.topic.enabled)。	布林	true
imq.metrics.interval	如果已啓用度量記錄程式 (imq.metrics.enabled=true)，則請指定將度量資訊寫入記錄程式的時間間隔 (以秒為單位)。不會影響產生度量訊息的時間間隔 (請參閱 imq.metrics.topic.interval)。 值 -1 表示永不報告。	整數	-1
imq.metrics.topic.enabled	這個布林值指定是否啓用度量訊息產生功能。若為 false，那麼會拋出一個用戶端異常，表示嘗試指向度量主題目標。	布林	true
imq.metrics.topic.interval	產生傳送到度量主題目標之度量訊息的時間間隔 (以秒為單位)。	整數	60
imq.metrics.topic.persist	這個布林值指定度量訊息是否為永久性的。	布林	false
imq.metrics.topic.timetolive	傳送到度量主題目標之度量訊息的有效期 (以秒為單位)。	整數	300

1. 這個特性可用於 imqcmd update bkr。

叢集配置特性

表 14-11 摘要出代理程式叢集的相關配置特性。

表 14-11 叢集配置特性

特性名稱	說明	類型	預設
imq.cluster.brokerlist	<p>用來識別叢集中所有代理程式的 <i>host:port</i> 項目清單 (中間以逗號分隔), 其中 <i>host</i> 是代理程式的主機名稱, <i>port</i> 是它的連接埠對映器連接埠號。</p> <p>範例:</p> <pre>host1:3000,host2:8000,ctrlhost</pre> <p>叢集中所有代理程式的此值必須相同。</p>	字串	無
imq.cluster.masterbroker	<p>叢集主代理程式的主機名稱和連接埠號 (若有)。</p> <p>值的格式是 <i>host:port</i>, 其中 <i>host</i> 是主代理程式的主機名稱, <i>port</i> 是它的連接埠對映器連接埠號。</p> <p>範例:</p> <pre>ctrlhost:7676</pre> <p>叢集中所有代理程式的此值必須相同。</p>	字串	無
imq.cluster.url ¹	<p>叢集配置檔案 URL (若有)。</p> <p>範例:</p> <pre>http://webserver/imq/cluster.properties</pre> <p>(web 伺服器上的檔案)</p> <pre>file:/net/mfsserver/imq/cluster.properties</pre> <p>(共用磁碟機上的檔案)</p> <p>叢集中所有代理程式的此值必須相同。</p>	字串	無
imq.cluster.hostname	<p>電腦有多個可用網路介面卡時, 叢集中的代理程式之間進行內部通訊時使用的 <i>cluster</i> 連線服務所連結的主機名稱或 IP 位址: 例如, 如果一部電腦中有多个網路介面卡。</p> <p>可以為叢集中的每個代理程式個別指定。</p>	字串	繼承 imq.host name 的值 (請參閱第 285 頁 表 14-2)
imq.cluster.port	<p>叢集連線服務的連接埠號。</p> <p>可以為叢集中的每個代理程式個別指定。</p>	整數	0 (動態配置。)

表 14-11 叢集配置特性 (續)

特性名稱	說明	類型	預設
imq.cluster.transport	叢集連線服務使用的網路傳輸通訊協定。爲了在代理程式之間安全的傳送加密訊息，請將這個特性設定爲ssl。 叢集中所有代理程式的此值必須相同。	字串	tcp

1. 這個特性可用於 imqcmd update bkr。

實體目標特性參照

本章描述您可以為每個實體目標類型設定的特性。您建立或更新實體目標時，可以設定特性值。

針對自動建立的目標，請在代理程式實例配置檔案中設定預設值（請參閱第 288 頁表 14-4）。

表 15-1 實體目標特性

特性	目標類型	預設值	說明
maxNumMsgs ¹	佇列主題	-1 (無限制)	目標中允許給未使用訊息的最大數目。 停用的訊息佇列的預設值是 1000。
maxTotalMsgBytes ¹	佇列 主題	-1 (無限制)	目標中允許給未使用訊息的最大總記憶體容量 (以位元組為單位)。 停用的訊息佇列的預設值是 10 MB。

表 15-1 實體目標特性 (續)

特性	目標類型	預設值	說明
limitBehavior	佇列 主題	REJECT_ NEWEST	<p>這個字串指定達到記憶體限制臨界值時，代理程式回應的方式。值為：</p> <p>FLOW_CONTROL - 減緩產生者速度。</p> <p>REMOVE_OLDEST - 拋出最舊的訊息。</p> <p>REMOVE_LOW_PRIORITY - 根據訊息存在時間，拋出最後順位的訊息 (生產型用戶端不會收到訊息刪除的通知)。</p> <p>REJECT_NEWEST - 拒絕最新的訊息。拒絕永久性訊息時，生產型用戶端會發生異常。若要將這個限制運作方式用在非永久性訊息上，請設定 <code>imqAckOnProduce</code> 連線工廠屬性。</p> <p>如果您將這個特性設定為 REMOVE_OLDEST 或 REMOVE_LOW_PRIORITY，並將目標特性 <code>useDMQ</code> 設定為 <code>true</code>，代理程式會將超過此限制的訊息移動至停用的訊息佇列。</p> <p>停用的訊息佇列本身有預設限制運作方式 REMOVE_OLDEST，不能設定 FLOW_CONTROL 運作方式，這點與其他目標不同。</p>
maxBytesPerMsg	佇列 主題	-1 (無限制)	<p>目標中允許的任何單一訊息的最大容量 (以位元組為單位)。拒絕永久性訊息時，生產型用戶端會發生異常，但拒絕非永久性訊息時則不會收到通知，除非設定 <code>ackOnProduce</code> 特性。</p>
maxNumProducers	佇列 主題	-1 (無限制)	<p>允許給目標中產生者的最大數目。達到限制時，將無法建立新的產生者。</p> <p>您不能為停用的訊息佇列設定這個特性。</p>
maxNumActiveConsumers	僅限佇列	1	<p>最大用戶數目，此數值可作用於來自佇列目標的負載平衡傳送。值 -1 表示沒有限制數目。</p> <p>Platform Edition 會將此值限制為 2。</p>
maxNumBackupConsumers	僅限佇列	0	<p>最大備份用戶數目，如果無法從佇列目標進行負載平衡傳送，那麼這些用戶可以取代使用中的用戶。值 -1 表示沒有限制數目。</p> <p>Platform Edition 會將此值限制為 1 (一)。</p>

表 15-1 實體目標特性 (續)

特性	目標類型	預設值	說明
consumerFlowLimit	佇列 主題	主題：1000 佇列：1000	<p>最大訊息數目，這些訊息將會傳送給單一批次中的某個用戶。在負載平衡佇列傳送中，負載平衡開始前，此數目是路由到使用中用戶的佇列訊息初始數目。</p> <p>目標用戶可以在連線上指定較低值，覆寫這項限制。值 -1 表示沒有限制數目。</p>
localDeliveryPreferred	僅限佇列	false	<p>針對代理程式叢集中的負載平衡佇列傳送，這個布林值指定當本地代理程式沒有用戶時，是否將訊息傳送給遠端用戶。要求目標不僅限於本地傳送 (isLocalOnly = false)。</p> <p>這個特性不套用於停用的訊息佇列。</p>
isLocalOnly	佇列 主題	false	<p>針對代理程式叢集中的目標，這個布林值指定目標是否限於本地傳送。如果為 true，則其他代理程式上不重複實體目標，且限制目標僅傳送訊息給本地用戶 (連線到有建立目標的代理程式的用戶)。一旦建立目標，即無法變更此特性。</p> <p>這個特性不套用於停用的訊息佇列。</p>
useDMQ	佇列 主題	true	<p>這個布林值指定是否將停用的訊息傳送至停用的訊息佇列而不捨棄。</p> <p>這個特性不套用於停用的訊息佇列。</p>

1. 在叢集環境中，此特性套用於叢集中的每個目標實例，但整體上，不套用於叢集中的所有實例。

受管理物件屬性參照

這一章提供受管理物件屬性的參照資訊。它包含下面幾節：

- [第 308 頁「目標特性」](#)
- [第 308 頁「連線工廠屬性」](#)
- [第 315 頁「SOAP 端點屬性」](#)

目標特性

表 16-1 列出用來配置目標受管理物件的屬性。

表 16-1 目標受管理物件屬性

屬性名稱	說明	類型	預設
imqDestinationDescription	目標物件的說明。	字串	無
imqDestinationName	實體目標名稱。	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。

連線工廠屬性

這一節包含用來配置連線工廠受管理物件的屬性的參照資訊。屬性分類後分配到下面幾節：

- [第 309 頁「連線處理」](#)
- [第 312 頁「用戶端標識」](#)
- [第 312 頁「訊息標頭覆寫」](#)
- [第 313 頁「穩定性與流量控制」](#)
- [第 314 頁「佇列瀏覽器運作方式和伺服器階段作業」](#)
- [第 315 頁「JMS 定義的特性支援」](#)

表 16-2 是連線工廠受管理物件屬性的索引。第一欄按字母順序列出每個屬性，第二欄列出屬性類別，第三欄為交叉參照屬性說明表。

連線處理

表 16-2 列出連線處理的連線工廠屬性。

表 16-2 連線工廠屬性：連線處理

屬性名稱	說明	類型	預設
imqAddressList	<p>一或多個用逗號分隔的訊息伺服器位址清單。有幾個特定定址方案可用來指定您想使用的連線服務和連接埠之方法。</p> <p>如需如何指定位址清單和清單項目說明範例的詳細資訊，請參閱第 310 頁「imqAddressList 屬性值的語法」。</p>	字串	現有的 Message Queue 3.0 位址 (若有)，若無，則為第 310 頁表 16-3 的第一個項目。
imqAddressListBehavior	這個字串指定連線時，按照 imqAddressList 屬性中的位址次序 (PRIORITY) 或按照隨機次序 (RANDOM)。如果您有許多用戶端使用相同的連線工廠以嘗試連線，則可以使用隨機次序以防止這些用戶端全部連線至相同的位址。	字串	PRIORITY
imqAddressListIterations	用戶端執行階段透過 imqAddressList 建立或重新建立連線的最大次數。值 -1 表示嘗試次數不受限制。	整數	5
imqPingInterval	<p>用戶端執行階段測試應用程式和代理程式之間連線的頻率 (以秒為單位)。</p> <p>值 -1 或 0 (零) 表示用戶端執行階段不定期測試連線。</p>	整數	30
imqReconnectEnabled	這個布林值指定連線中斷時，用戶端執行階段是否重新連線到訊息伺服器 (或 imqAddressList 中的位址清單)。	布林	false
imqReconnectAttempts	用戶端執行階段嘗試連線 (或重新連線) imqAddressList 中每個位址的最大次數，到達這個次數後就接著處理清單中的下一個位址。值 -1 表示不限制嘗試重新連線的次數 (用戶端執行階段一直嘗試，到連線成功為止)。	整數	0 (零)
imqReconnectInterval	嘗試重新連線的間隔 (以毫秒為單位)。此值套用於嘗試連線 imqAddressList 中的每個位址和清單中連續的位址。如果此值太小，代理程式回復時間會不足。如果此值太大，重新連線的延遲可能令人無法忍受。	長	3000 (毫秒)
imqSSLIsHostTrusted	這個布林值指定用戶端能否接受代理程式自身簽名的憑證。若要使用認證機構簽名的憑證，請將此值設定為 false。	布林	true

imqAddressList 屬性值的語法

imqbrokerlist 值中的每個位址，對應用戶端執行階段可以連線的代理程式實例。

請為每個連線服務指定不同的代理程式位址。一般語法如下：

```
scheme://address_syntax
```

若要為清單新增位址，請新增一個逗點，再新增一個位址。清單可以有許多項目，格式如下：

```
scheme://address_syntax,scheme://address_syntax...
```

scheme 變數指定您使用的下列一種定址類型，表 16-3 中有詳細說明：mq、mqtcp、mqssl、http 或 https。*address_syntax* 變數表示方案特定的代理程式位址。表 16-3 列出定址方案。第一欄包含定址方案名稱，第二欄顯示此名稱的相關連線服務，第三欄是說明，第四欄提供適用語法。

表 16-3 imqAddressList 屬性的定址方案

方案	連線服務	說明	語法
mq	ms ssljms	指定 jms 或 ssljms 服務使用的動態連接埠。 您可以指定連接埠對映器主機和連接埠。連接埠對映器會動態指定連線用的連接埠。	<code>[hostName][:port][/serviceName]</code> jms 連線服務可以套用下列預設值： <code>hostName = localhost</code> <code>port = 7676</code> <code>serviceName = jms</code> ssljms 連線服務沒有預設值。您必須指定所有變數。
mqtcp	jms	指定連接埠號碼並且使用 jms 連線服務。 Message Queue 用戶端執行階段與指定的主機和連接埠進行 tcp 連線，以建立連線。	<code>hostName:port/jms</code>
mqssl	ssljms	指定連接埠號碼並且使用 ssljms 連線服務。 Message Queue 用戶端執行階段與指定的主機和連接埠進行安全的 ssl 連線，以建立連線。	<code>port/ssljms</code>

表 16-3 imqAddressList 屬性的定址方案

方案	連線服務	說明	語法
http	httpjms	使用 httpjms 連線服務。 用戶端執行階段在指定的 URL，與 Message Queue 通道 Servlet 進行 HTTP 連線。您必須配置代理程式才能存取 HTTP 通道 Servlet。	http://hostName:port/contextRoot/tunnel 如果多個代理程式實例使用相同的通道 Servlet，下面是連線到特定（而非隨機選定）代理程式實例的語法： http://hostName:port/contextRoot/tunnel?ServerName=hostName:instanceName
https	httpsjms	使用 httpsjms 連線服務。 Message Queue 用戶端執行階段與指定的 Message Queue 通道 Servlet URL 進行安全的 HTTPS 連線。您必須配置代理程式才能存取 HTTPS 通道 Servlet。	https://hostName:port/contextRoot/tunnel 如果多個代理程式實例使用相同的通道 Servlet，下面是連線到特定（而非隨機選定）代理程式實例的語法： https://hostName:port/contextRoot/tunnel?ServerName=hostName:instanceName

表 16-4 包含定址格式範例。第一欄是連線服務名稱。第二欄指定範例中的主機是本地主機、未指定主機、指定主機或不適用。第三欄指定範例中的連接埠是指定、未指定或不適用，第四欄是範例。

表 16-4 訊息伺服器位址範例

連線服務	代理程式主機	連接埠	範例位址
未指定	本地主機	未指定	預設 (mq://localhost:7676/jms)
未指定	指定主機	未指定	myBkrHost (mq://myBkrHost:7676/jms)
未指定	未指定	指定連接埠對映器的連接埠	1012 (mq://localhost:1012/jms)
ssljms	本地主機	未指定連接埠對映器的連接埠	mq://localhost:7676/ssljms
ssljms	指定主機	連接埠對映器連接埠	mq://myBkrHost:7676/ssljms
ssljms	指定主機	指定連接埠對映器連接埠	mq://myBkrHost:1012/ssljms
jms	本地主機	指定服務連接埠	mqtcp://localhost:1032/jms
ssljms	指定主機	指定服務連接埠	mqssl://myBkrHost:1034/ssljms

表 16-4 訊息伺服器位址範例 (續)

連線服務	代理程式主機	連接埠	範例位址
httpjms	不適用	不適用	http://websrvr1:8085/imq/tunnel
httpsjms	不適用	不適用	https://websrvr2:8090/imq/tunnel

用戶端標識

表 16-5 列出用戶端標識的連線工廠屬性。

表 16-5 連線工廠屬性：用戶端標識

屬性名稱	說明	類型	預設
imqDefaultUsername	由代理程式認證的預設使用者名稱。	字串	guest
imqDefaultPassword	由代理程式認證的預設密碼。	字串	guest
imqConfiguredClientID	以管理級別配置的用戶端 ID。	字串	null
imqDisableSetClientID	這個布林值指定是否在 JMS API 中使用 setClientID() 方法，防止用戶端變用戶端 ID。	布林	false

訊息標頭覆寫

表 16-6 列出用來覆寫 JMS 訊息標頭欄位的連線工廠屬性。

表 16-6 連線工廠屬性：訊息標頭覆寫

屬性名稱	說明	類型	預設
imqOverrideJMSDeliveryMode	這個布林值指定能否覆寫用戶端設定的 JMSDeliveryMode 欄位。	布林	false
imqJMSDeliveryMode	JMSDeliveryMode 的覆寫值。值為 1 (非永久性) 和 2 (永久性)。	整數	2
imqOverrideJMSExpiration	這個布林值指定能否覆寫用戶端設定的 JMSExpiration 欄位。	布林	false
imqJMSExpiration	JMSExpiration 的覆寫值 (以毫秒為單位)。	長	0 (不過期)
imqOverrideJMSPriority	這個布林值指定能否覆寫用戶端設定的 JMSPriority 欄位。	布林	false

表 16-6 連線工廠屬性：訊息標頭覆寫 (續)

屬性名稱	說明	類型	預設
imqJMSPriority	JMSPriority 的覆寫值 (整數 0 到 9)。	整數	4 (一般)
imqOverrideJMSHeadersToTemporaryDestinations	這個布林值指定臨時目標是否套用覆寫。	布林	false

穩定性與流量控制

表 16-7 列出用來配置穩定性與流量控制的連線工廠屬性。

表 16-7 連線工廠屬性：穩定性與流量控制

屬性名稱	說明	類型	預設
imqAckTimeout	<p>指定在拋出異常之前，用戶端執行階段等待代理程式回應的最長時間 (以毫秒為單位)。值 0 表示無逾時，用戶端執行階段會一直等待。</p> <p>有時此值太小會導致用戶端執行階段逾時。例如，代理程式初次使用 LDAP 使用者儲存庫和安全 (SSL) 連線來認證時，可能需要費時 30 秒鐘以上。</p>	字串	0
imqAckOnProduce	<p>這個字串指定代理程式如何回應生產型用戶端的訊息。</p> <p>如果這個屬性設定為 true，代理程式會回應所有從生產型用戶端接收到的 JMS 訊息 (永久性和非永久性的)。等待回應時，生產型用戶端執行緒會進行鎖定。</p> <p>如果未指定這個屬性，則代理程式只回應永久性訊息。等待回應時，生產型用戶端執行緒會進行鎖定。</p>	字串	未指定
imqConnectionFlowCount	<p>計數批次中的最大 JMS 訊息數。當此數量的 JMS 訊息傳送至用戶端執行階段時，傳送將臨時暫停，並允許傳送任何被阻擋的控制訊息。依照用戶端執行階段的通知，恢復傳送有效負載訊息，直至計數再次到達指定數目。</p> <p>如果將計數設定為 0，則在計數批次中對 JMS 訊息數量不做限制。非零設定允許用戶端執行階段計算訊息流量，以免因為 JMS 訊息傳送過量而鎖定 Message Queue 控制訊息。</p>	整數	100
imqConnectionFlowLimitEnabled	這個布林值指定是否使用 imqConnectionFlowLimit 值來限制連線層級的訊息流量。	布林	false

表 16-7 連線工廠屬性：穩定性與流量控制 (續)

屬性名稱	說明	類型	預設
imqConnectionFactoryLimit	<p>可透過連線傳送的，以及可以在用戶端執行階段緩衝待用的最大訊息數量。但是請注意，除非啓用 <code>imqConnectionFactoryIsLimited</code>，否則不檢查這項限制。</p> <p>當傳送至用戶端執行階段的 JMS 訊息數量（依照由 <code>imqConnectionFactoryCount</code> 所控制的流量測量）超過了該限制時，訊息傳送就會停止。僅當未使用的訊息數目低於此屬性設定的值時，才恢復傳送訊息。</p> <p>此限制對於耗費大量時間處理訊息的使用中用戶端，可使其免於受擱置訊息所苦，因為這些訊息可能會導致系統記憶體不足。</p>	整數	1000
imqConsumerFlowLimit	<p>每位用戶可透過連線傳送的，以及可以在用戶端執行階段緩衝待用的最大訊息數量。這項限制可以在多用戶佇列傳送情況中，改善用戶之間的負載平衡（沒有用戶會收到過多的訊息數）。代理程式上設定佇列的 <code>consumerFlowLimit</code> 屬性較小值，可以覆寫這項限制（請參閱 <i>Message Queue 管理指南</i> 中的目標屬性資訊）。</p> <p>這項限制還可以防止用戶霸佔連線讓別人苦等。</p> <p>如果傳送至用戶端執行階段的 JMS 訊息數超過此限制時，則會停止傳送訊息給該用戶。只有在傳送給該用戶的未使用訊息數低於此 <code>imqConsumerFlowThreshold</code> 之設定值時，才恢復傳送訊息。</p> <p>請注意若連線上所有用戶的緩衝訊息總數超過 <code>imqConnectionFactoryLimit</code>，連線上的訊息傳送會停止，直到訊息總數低於連線限制為止）。</p>	整數	100
imqConsumerFlowThreshold	<p>用戶端執行階段緩衝的每位用戶之訊息數（<code>imqConsumerFlowLimit</code> 的百分比）低於該值時，便會繼續傳送訊息給用戶。</p>	整數	50

佇列瀏覽器運作方式和伺服器階段作業

表 16-8 說明會影響用戶端佇列瀏覽的屬性。

表 16-8 連線工廠屬性：佇列瀏覽器運作方式

屬性名稱	說明	類型	預設
imqQueueBrowserMaxMessagesPerRetrieve	<p>瀏覽佇列目標內容時，用戶端執行階段一次將擷取的 最大訊息數量。</p>	整數	1000

表 16-8 連線工廠屬性：佇列瀏覽器運作方式 (續)

屬性名稱	說明	類型	預設
imqQueueBrowserRetrieveTimeout	瀏覽佇列目標內容時，用戶端執行階段擷取訊息所需等待的最長時間 (以毫秒為單位)，一旦逾時，立即拋出異常。	長	60000
imqLoadMaxToServerSession	JMS 應用程式伺服器設備的這個布林值指定 Message Queue 連線用戶是否應該將訊息最大數目 maxMessages 上載到 ServerSession 階段作業。如果指定 false，用戶端一次只載入一個訊息。		true

JMS 定義的特性支援

JMS 定義的特性是 JMS 保留的名稱，JMS 提供者可以選擇支援加強用戶端程式設計功能。表 16-9 說明 Message Queue 支援的 JMS 定義特性。

表 16-9 連線工廠屬性：JMS 定義的特性支援

特性名稱	說明	類型	預設
imqSetJMSXUserID	這個布林值指定是否在產生的訊息上設定 JMS 定義的特性 JMSXUserID (傳送訊息的使用者的識別)。	布林	false
imqSetJMSXAppID	這個布林值指定是否在產生的訊息上設定 JMS 定義的特性 JMSXAppID (傳送訊息的應用程式的識別)。	布林	false
imqSetJMSXProducerTXID	這個布林值指定是否在產生的訊息上設定 JMS 定義的特性 JMSXProducerTXID (產生了此訊息的作業事件識別碼)。	布林	false
imqSetJMSXConsumerTXID	這個布林值指定是否在使用的訊息上設定 JMS 定義的特性 JMSXConsumerTXID (使用了此訊息的作業事件識別碼)。	布林	false
imqSetJMSXRcvTimestamp	這個布林值指定是否在使用的訊息上設定 JMS 定義的特性 JMSXRcvTimestamp (訊息傳送至用戶的時間)。	布林	false

SOAP 端點屬性

表 16-10 列出用來配置應用程式 (使用 SOAP) 端點 URL 的屬性。如需這些使用 SOAP 應用程式的詳細資訊，請參閱 Message Queue Developer's Guide for Java Clients。

表 16-10 SOAP 端點屬性

屬性名稱	說明	類型	預設
imqSOAPEndpointList	<p>一或多個用空格分隔的 URL 清單，URL 代表訊息要傳往的 SOAP 端點。</p> <p>如果指定多個 URL，則會向清單中的所有 URL 廣播訊息。每個 URL 應該有一個可以接收並處理 SOAP 訊息的相關 Servlet。</p> <p>範例：</p> <p>http://www.myServlet/ http://www.myServlet2/</p>	字串	
imqEndpointName	<p>SOAP 端點名稱。</p> <p>範例：MyTopicEndpoint</p>	字串	Untitled_Endpoint_Object
imqEndpointDescription	<p>SOAP 端點說明。</p> <p>範例：</p> <p>"imqEndpointDescription=my endpoints for broadcast"</p>	字串	端點物件說明

JMS 資源介面屬性參照

Message Queue JMS 資源介面 (JMS RA) 可讓您使用標準 J2EE Connector Architecture (JCA)，為 Sun Java System Message Queue 整合任何 J2EE 1.4 應用程式伺服器。Message Queue JMS 資源介面外掛應用程式伺服器後，此應用程式伺服器部署的應用程式可以使用 Message Queue 來傳送及接收 JMS 訊息。

Message Queue JMS 資源介面透過三個 JavaBean 元件來展現它的配置屬性：

- `ResourceAdapter` 配置全面影響資源介面的運作方式。
- `ManagedConnectionFactory` 配置影響資源介面建立供訊息驅動 Bean (MDB) 使用的連線。
- `ActivationSpec` 配置影響與訊息傳送系統互動時代表訊息驅動 Bean MDB 的訊息端點。

若要設定這些實體屬性，您可以使用應用程式伺服器在配置及部署資源介面及部署 MDB 時提供的工具。

本章列出並描述 Message Queue JMS 資源介面的配置屬性。它包含下面幾節：

- [第 318 頁](#) 「`ResourceAdapter JavaBean`」
- [第 319 頁](#) 「`ManagedConnectionFactory JavaBean`」
- [第 320 頁](#) 「`ActivationSpec JavaBean`」

ResourceAdapter JavaBean

ResourceAdapter 配置可以配置預設的 JMS 資源介面運作方式。表 17-1 列出並描述您可以用來配置這個 JavaBean 的屬性。註腳會標示出每個必要的特性。

表 17-1 資源介面屬性

名稱	說明	預設
addressList ¹	<p>資源介面與 Message Queue 服務的連線，使用訊息服務位址格式來指定。</p> <p>資源介面提供預設值。</p> <p>addressList 是 Sun Java System Message Queue 特有的屬性名稱，但是意義與標準屬性 connectionURL 相同。Sun Java System Message Queue 提供這兩種屬性名稱。您必須設定 connectionURL 或 addressList，二者是相等的。</p>	mq://localhost:7676/jms
addressListBehavior	<p>這個字串指定資源介面如何連線至 Message Queue 服務。此值是 PRIORITY 或 RANDOM。</p> <p>PRIORITY 連線選取位址清單 (addressList) 中指定的第一個 Message Queue 代理程式。</p> <p>RANDOM 連線隨機選取位址清單中的 Message Queue 代理程式。</p> <p>連線失敗後的重新連線方式，同 PRIORITY 和 RANDOM。代理程式連線失敗時，會嘗試重新連線。如果嘗試後仍然失敗，資源介面會繼續對使用中位址清單執行連線。</p>	PRIORITY
addressListIterations	<p>在位址清單中重複連線的次數。此值套用初始連線和繼續嘗試重新連線。</p>	1
connectionURL	<p>資源介面與 Message Queue 服務的連線，使用訊息服務位址格式來指定。</p> <p>等同於 addressList 屬性；如需詳細資訊，請參閱前述說明。</p>	mq://localhost:7676/jms
userName ¹	<p>資源介面連線至 Message Queue 服務時使用的預設使用者名稱。</p> <p>資源介面提供預設值。</p>	guest
password ¹	<p>資源介面連線至 Message Queue 服務時使用的預設密碼。</p> <p>資源介面提供預設值。</p>	guest

表 17-1 資源介面屬性 (續)

名稱	說明	預設
reconnectAttempts	在位址清單中嘗試重新連線單一項目的次數。 reconnectEnabled 設定為 true 時，即可使用此屬性。	6
reconnectEnabled	這個布林值指定連線失敗後是否嘗試重新連線。 嘗試重新連線的運作方式，是由 reconnectInterval 和 reconnectAttempts 值來 控制。	false
reconnectInterval	嘗試重新連線的間隔 (以毫秒為單位)。 reconnectEnabled 設定為 true 時，即可使用此屬性。	30000

1. 這是必要的特性。

ManagedConnectionFactory JavaBean

受管理連線工廠提供並定義資源介面提供給訊息驅動 Bean 的連線。如果您設定的屬性 ResourceAdapter JavaBean 也有類似的屬性，此設定會抑制 ResourceAdapter Bean 的類似指定值。

表 17-2 列出並描述 Message Queue 資源介面提供的受管理連線工廠的可配置屬性。

表 17-2 受管理連線工廠屬性

名稱	說明	預設
addressList	衍生自此受管理連線工廠的連線清單。 這個特性的格式如第 318 頁表 17-1 中所描述，符合訊息 服務 addressList。如果未設定此值，則連線如該表中 所描述，使用 ResourceAdapter JavaBean 的 addressList 指定值。	無

表 17-2 受管理連線工廠屬性 (續)

名稱	說明	預設
addressListBehavior	<p>這個字串指定資源介面如何連線至 Message Queue 服務。此值是 PRIORITY 或 RANDOM。</p> <p>PRIORITY 連線選取位址清單 (addressList) 中指定的第一個 Message Queue 代理程式。</p> <p>RANDOM 連線隨機選取位址清單中的 Message Queue 代理程式。</p> <p>連線失敗後的重新連線方式，同 PRIORITY 和 RANDOM。代理程式連線失敗時，會嘗試重新連線。如果嘗試後仍然失敗，資源介面會繼續對使用中位址清單執行連線。</p>	PRIORITY
addressListIterations	在位址清單中重複連線的次數。此值套用初始連線和繼續嘗試重新連線。	1
clientID	衍生自此受管理連線工廠的連線所使用的用戶端識別碼。	無
password	<p>(選用) 連線密碼。</p> <p>如果未設定此值，則連線如第 318 頁表 17-1 中所描述，使用 ResourceAdapter JavaBean 的指定密碼。</p>	guest
reconnectAttempts	在位址清單中嘗試重新連線單一項目的次數。	6
reconnectEnabled	<p>這個布林值指定連線失敗後是嘗試重新連線或嘗試新的連線。</p> <p>嘗試重新連線，是由 reconnectInterval 和 reconnectAttempts 特性來控制。</p>	false
reconnectInterval	嘗試重新連線至 Message Queue 服務的最小間隔毫秒數。	30000
userName	<p>(選用) 連線的使用者名稱。</p> <p>如果未設定此值，則連線如第 318 頁表 17-1 中所描述，使用 ResourceAdapter JavaBean 的指定使用者名稱。</p>	guest

ActivationSpec JavaBean

應用程式伺服器指示資源介面啟動訊息端點並且讓訊息端點與訊息驅動 Bean 產生關聯時，會使用 ActivationSpec JavaBean 特性。

表 17-3 列出並描述訊息端點啟動規格的可配置屬性。此表指出 Message Queue 資源介面的特有特性，以及 Enterprise JavaBean 2.1 標準或 J2EE Connector Architecture (J2EE CA) 1.5 標準的特有特性。

表 17-3 啓動規格屬性

名稱	說明	預設
acknowledgeMode	(選用) 用戶使用的 JMS 階段作業確認模式。 這是標準 EJB 2.1 和 J2EE CA 1.5 特性。 此值可能是 Auto-acknowledge 或 Dups-ok-acknowledge。	Auto-acknowledge
addressList	(選用) 資源介面為訊息端點製作的連線規格。 這是 Message Queue JMS 資源介面的特有屬性。 有效值必須符合訊息服務連線位址語法。	繼承 ResourceAdapter JavaBean 配置中的 addressList
clientId	為此用戶建立的 JMS 連線所使用的 JMS 用戶端 ID。 如果您將 subscriptionDurability 屬性設定為 Durable，就必須設定此屬性。 這是標準 EJB 2.1 和 J2EE CA 1.5 特性。	無
customAcknowledgeMode	這個字串指定 MDB 訊息使用模式。 這個屬性的有效值是 No_acknowledge 或空值。 您可以將 No_acknowledge 模式只用於非作業事件的非長期主題訂閱。如果此設定用於作業事件訂閱或長期訂閱，會無法啓動訂閱。	無
destination	MDB 從中使用訊息的目標名稱。 這是必要的屬性。這是標準 EJB 2.1 和 J2EE CA1.5 特性。 此值必須設定為 Message Queue 目標受管理物件的 destinationName 特性值。	無
destinationType	destination 屬性指定的目標類型。有效值是 javax.jms.Queue 或 javax.jms.Topic。 這是必要的屬性。這是標準 EJB 2.1 和 J2EE CA1.5 特性。	無
endpointExceptionRedeliveryAttempts	MDB 在傳送訊息期間拋出異常時，向 MDB 重新傳送訊息的次數。	6
messageSelector	(選用) 在篩選傳送給用戶的訊息時所使用的 JMS 訊息選擇器。此值是字串類型。 這是標準 EJB 2.1 和 J2EE CA 1.5 特性。	無

表 17-3 啓動規格屬性 (續)

名稱	說明	預設
sendUndeliverableMsgsToDMQ	<p>這個布林值指定當 MDB 拋出執行階段異常而且嘗試重新傳送的次數超出 <code>endpointExceptionRedeliveryAttempts</code> 值時，是否將訊息放入停用的訊息佇列中。</p> <p>如果是 <code>false</code>，Message Queue 代理程式會嘗試向任何有效的用戶 (包含此 MDB) 重新傳送訊息。</p>	true
subscriptionDurability	<p>這個字串指定主題目標的用戶是長期或非長期。此值可能是 <code>NonDurable</code> 或 <code>Durable</code>。</p> <p>這個屬性是非長期訂閱的選用屬性，是長期訂閱的必要屬性。如果您將此值設定為 <code>Durable</code>，還必須設定屬性 <code>clientId</code> 和 <code>subscriptionName</code>。</p> <p>這是標準 EJB 2.1 和 J2EE CA1.5 特性，而且只在 <code>destinationType</code> 屬性設定為 <code>avax.jms.Topic</code> 時有效。</p>	NonDurable
subscriptionName	<p>這個字串可以為長期訂閱命名。</p> <p>如果您將 <code>subscriptionDurability</code> 屬性設定為 <code>Durable</code>，就必須設定此屬性。</p> <p>這是標準 EJB 2.1 和 J2EE CA 1.5 特性。</p>	無

度量參照

本章列出並描述 Message Queue 產品產生的度量。本章包含以下各節：

- [第 324 頁「JVM 度量」](#)
- [第 324 頁「代理程式範圍度量」](#)
- [第 326 頁「連線服務度量」](#)
- [第 328 頁「目標度量」](#)

JVM 度量

表 18-1 列出並描述代理程式為代理程式程序 JVM 堆疊產生的度量資料。表格顯示出提供每個度量的度量監視工具。

表 18-1 JVM 度量

度量數目	說明	imqcmd metrics bkr (metricType)	記錄檔	度量訊息 (度量主題) ²
JVM 堆疊：可用記憶體	可在 JVM 堆疊中使用的可用記憶體容量。	需要 (cxn)	需要	需要 (...jvm)
JVM 堆疊：總記憶體	目前的 JVM 堆疊容量	需要 (cxn)	需要	需要 (...jvm)
JVM 堆疊：最大記憶體	可增加 JVM 堆疊容量的最大值。	不需要	需要 ¹	需要 (...jvm)

1. 僅在代理程式啟動時顯示。

2. 如需度量主題目標的名稱，請參閱第 199 頁表 10-7。

代理程式範圍度量

表 18-2 列出並描述代理程式報告關於代理程式範圍度量資訊的資料。它亦顯示使用不同度量監視工具可取得的資料類型。

表 18-2 代理程式範圍度量

度量數目	說明	imqcmd metrics bkr (metricType)	記錄檔	度量訊息 (度量主題) ¹
連線資料				
連線數目	目前開啓至代理程式連線的數目	需要 (cxn)	需要	需要 (...broker)
執行緒數目	目前所有連線服務使用的執行緒總數目	需要 (cxn)	需要	不需要
最小執行緒數目	一旦到達執行緒數目，便會在執行緒池中維護執行緒數目，以供連線服務使用	需要 (cxn)	需要	不需要
最大執行緒數目	一旦超過執行緒數目，便不會再新增執行緒到執行緒池，以供連線服務使用	需要 (cxn)	需要	不需要

表 18-2 代理程式範圍度量 (續)

度量數目	說明	imqcmd metrics bkr (metricType)	記錄檔	度量訊息 (度量主題) ¹
儲存的訊息資料				
訊息數目	目前儲存於代理程式記憶體和永久性儲存的 JMS 訊息數目	不需要 使用 query bkr	不需要	需要 (...broker)
訊息總容量	目前儲存於代理程式記憶體和永久性儲存的 JMS 訊息容量	不需要 使用 query bkr	不需要	需要 (...broker)
訊息流量資料				
流入的訊息數目	從上次代理程式啟動後，流入代理程式的 JMS 訊息數目	需要 (ttl)	需要	需要 (...broker)
流入的訊息容量	從上次代理程式啟動後，流入代理程式的 JMS 訊息容量	需要 (ttl)	需要	需要 (...broker)
流入的封包數目	從上次代理程式啟動後，流入代理程式的封包數目；包括 JMS 訊息和控制訊息	需要 (ttl)	需要	需要 (...broker)
流入的封包容量	從上次代理程式啟動後，流入代理程式的封包容量；包括 JMS 訊息和控制訊息	需要 (ttl)	需要	需要 (...broker)
流出的訊息數目	從上次代理程式啟動後，流出代理程式的 JMS 訊息數目	需要 (ttl)	需要	需要 (...broker)
流出的訊息容量	從上次代理程式啟動後，流出代理程式的 JMS 訊息容量	需要 (ttl)	需要	需要 (...broker)
流出的封包數目	從上次代理程式啟動後，流出代理程式的封包數目；包括 JMS 訊息和控制訊息。	需要 (ttl)	需要	需要 (...broker)
流出的封包容量	從上次代理程式啟動後，流出代理程式的封包容量；包括 JMS 訊息和控制訊息	需要 (ttl)	需要	需要 (...broker)
流入訊息的速率	目前流入代理程式的 JMS 訊息速率	需要 (rts)	需要	不需要
流入訊息容量的速率	目前流入代理程式之 JMS 訊息容量的速率	需要 (rts)	需要	不需要
流入的封包速率	目前流入代理程式的封包速率；包括 JMS 訊息和控制訊息	需要 (rts)	需要	不需要
流入封包容量的速率	目前流入代理程式之封包容量的速率；包括 JMS 訊息和控制訊息	需要 (rts)	需要	不需要

表 18-2 代理程式範圍度量 (續)

度量數目	說明	imqcmd metrics bkr (metricType)	記錄檔	度量訊息 (度量主題) ¹
流出訊息的速率	目前流出代理程式的 JMS 訊息速率	需要 (rts)	需要	不需要
流出訊息容量的速率	目前流出代理程式之 JMS 訊息容量的速率	需要 (rts)	需要	不需要
流出封包的速率	目前流出代理程式的封包速率；包括 JMS 訊息和控制訊息	需要 (rts)	需要	不需要
流出封包容量的速率	目前流出代理程式之封包容量的速率；包括 JMS 訊息和控制訊息	需要 (rts)	需要	不需要
目標資料				
目標數目	代理程式中實體目標的數目	不需要	不需要	需要 (...broker)

1. 如需度量主題目標的名稱，請參閱第 199 頁表 10-7。

連線服務度量

表 18-3 列出並描述代理程式報告用於個別連線服務的度量資料。它亦顯示使用不同度量監視工具可取得的資料類型。

表 18-3 連線服務度量

度量數目	說明	imqcmd metrics svc (metricType)	記錄檔	度量訊息 (度量主題)
連線資料				
連線數目	目前開啓連線的數目	需要 (cxn) 亦可使用 query svc	不需要	不需要
執行緒數目	目前使用中的執行緒數目	需要 (cxn) 亦可使用 query svc	不需要	不需要

表 18-3 連線服務度量 (續)

度量數目	說明	imqcmd metrics svc (metricType)	記錄檔	度量訊息 (度量主題)
最小執行緒數目	一旦到達執行緒數目，便會在執行緒池中維護執行緒數目，以供連線服務使用；此數目為所有連線服務的加總	需要 (cxn)	不需要	不需要
最大執行緒數目	一旦超過執行緒數目，便不會再新增執行緒到執行緒池，以供連線服務使用；此數目為所有連線服務的加總	需要 (cxn)	不需要	不需要
訊息流量資料				
流入的訊息數目	從上次代理程式啟動後，流入連線服務的 JMS 訊息數目	需要 (ttl)	不需要	不需要
流入的訊息容量	從上次代理程式啟動後，流入連線服務的 JMS 訊息容量	需要 (ttl)	不需要	不需要
流入的封包數目	從上次代理程式啟動後，流入連線服務的封包數目；包括 JMS 訊息和控制訊息。	需要 (ttl)	不需要	不需要
流入的封包容量	從上次代理程式啟動後，流入連線服務的封包容量；包括 JMS 訊息和控制訊息。	需要 (ttl)	不需要	不需要
流出的訊息數目	從上次代理程式啟動後，流出連線服務的 JMS 訊息數目	需要 (ttl)	不需要	不需要
流出的訊息容量	從上次代理程式啟動後，流出連線服務的 JMS 訊息容量	需要 (ttl)	不需要	不需要
流出的封包數目	從上次代理程式啟動後，流出連線服務的封包數目；包括 JMS 訊息和控制訊息	需要 (ttl)	不需要	不需要
流出的封包容量	從上次代理程式啟動後，流出連線服務的封包容量；包括 JMS 訊息和控制訊息	需要 (ttl)	不需要	不需要
流入訊息的速率	目前通過連線服務流入代理程式的 JMS 訊息速率。	需要 (rts)	不需要	不需要
流入訊息容量的速率	目前流入連線服務之 JMS 訊息容量的速率	需要 (rts)	不需要	不需要
流入的封包速率	目前流入連線服務的封包速率；包括 JMS 訊息和控制訊息	需要 (rts)	不需要	不需要
流入封包容量的速率	目前流入連線服務之封包容量的速率；包括 JMS 訊息和控制訊息	需要 (rts)	不需要	不需要
流出訊息的速率	目前流出連線服務的 JMS 訊息速率	需要 (rts)	不需要	不需要

表 18-3 連線服務度量 (續)

度量數目	說明	imqcmd metrics svc (metricType)	記錄檔	度量訊息 (度量主題)
流出訊息容量的速率	目前流出連線服務之 JMS 訊息容量的速率	需要 (rts)	不需要	不需要
流出封包的速率	目前流出連線服務的封包速率；包括 JMS 訊息和控制訊息	需要 (rts)	不需要	不需要
流出封包容量的速率	目前流出連線服務之封包容量的速率；包括 JMS 訊息和控制訊息	需要 (rts)	不需要	不需要

目標度量

表 18-4 列出並描述代理程式報告用於個別目標的度量資料。它亦顯示使用不同度量監視工具可取得的資料類型。

表 18-4 目標度量

度量數目	說明	imqcmd metrics dst (metricType)	記錄檔	度量訊息 (度量主題)'
用戶資料				
用戶數目	目前的用戶數目。主題的此值包含非長期訂閱、使用中長期訂閱和非使用中長期訂閱。佇列的此值包含使用中用戶和備份用戶。	需要 (con)	不需要	需要 (...destName)
用戶的平均數目	從上次代理程式啟動後，用戶的平均數目	需要 (con)	不需要	需要 (...destName)
用戶的尖峰數目	從上次代理程式啟動後，用戶的尖峰數目	需要 (con)	不需要	需要 (...destName)
使用中用戶的數目	目前使用中用戶的數目	需要 (con)	不需要	需要 (...destName)
使用中用戶的平均數目	從上次代理程式啟動後，使用中用戶的平均數目	需要 (con)	不需要	需要 (...destName)
使用中用戶的尖峰數目	從上次代理程式啟動後，使用中用戶的尖峰數目	需要 (con)	不需要	需要 (...destName)
備份用戶的數目	目前備份用戶的資料 (僅套用於佇列)	需要 (con)	不需要	需要 (...destName)

表 18-4 目標度量 (續)

度量數目	說明	imqcmd metrics dst (metricType)	記錄檔	度量訊息 (度量主題) ¹
備份用戶的平均數目	從上次代理程式啟動後，備份用戶的平均數目 (僅套用於佇列)	需要 (con)	不需要	需要 (...destName)
備份用戶的尖峰數目	從上次代理程式啟動後，備份用戶的尖峰數目 (僅套用於佇列)	需要 (con)	不需要	需要 (...destName)
儲存的訊息資料				
訊息數目	目前儲存於目標記憶體和永久性儲存的 JMS 訊息數目	需要 (con) (ttl) (rts) 亦可使用 query dst	不需要	需要 (...destName)
訊息平均數目	從上次代理程式啟動後，儲存於目標記憶體和永久性儲存中的 JMS 訊息平均數目	需要 (con) (ttl) (rts)	不需要	需要 (...destName)
訊息尖峰數目	從上次代理程式啟動後，儲存於目標記憶體和永久性儲存中的 JMS 訊息尖峰數目	需要 (con) (ttl) (rts)	不需要	需要 (...destName)
訊息總容量	目前儲存於目標記憶體和永久性儲存的 JMS 訊息容量	需要 (ttl) (rts) 亦可使用 query dst	不需要	需要 (...destName)
訊息的總平均容量	從上次代理程式啟動後，儲存於目標記憶體和永久性儲存中的 JMS 訊息平均容量	需要 (ttl) (rts)	不需要	需要 (...destName)
訊息的總尖峰容量	從上次代理程式啟動後，儲存於目標記憶體和永久性儲存中的 JMS 訊息尖峰容量	需要 (ttl) (rts)	不需要	需要 (...destName)
訊息的尖峰容量	從上次代理程式啟動後，目標接收的單一訊息中 JMS 訊息的尖峰容量	需要 (ttl) (rts)	不需要	需要 (...destName)
訊息流量資料				
流入的訊息數目	從上次代理程式啟動後，流入目標的 JMS 訊息數目	需要 (ttl)	不需要	需要 (...destName)

表 18-4 目標度量 (續)

度量數目	說明	imqcmd metrics dst (metricType)	記錄檔	度量訊息 (度量主題) ¹
流入的訊息容量	從上次代理程式啟動後，流入目標的 JMS 訊息容量	需要 (ttl)	不需要	需要 (...destName)
流出的訊息數目	從上次代理程式啟動後，流出目標的 JMS 訊息數目	需要 (ttl)	不需要	需要 (...destName)
流出的訊息容量	從上次代理程式啟動後，流出目標的 JMS 訊息容量	需要 (ttl)	不需要	需要 (...destName)
流入訊息的速率	目前流入目標的 JMS 訊息速率	需要 (rts)	不需要	不需要
流出訊息的速率	目前流出目標式的 JMS 訊息速率	需要 (rts)	不需要	不需要
流入訊息容量	目前流入目標之 JMS 訊息容量的速率	需要 (rts)	不需要	不需要
流出訊息容量的	目前流出目標之 JMS 訊息容量的速率	需要 (rts)	不需要	不需要
磁碟使用資料				
保留的磁碟空間	基於檔案的目標儲存中，所有訊息記錄 (使用中或可用) 使用的磁碟空間 (以位元組為單位)	需要 (dsk)	不需要	需要 (...destName)
使用的磁碟空間	基於檔案的目標儲存中，使用中訊息記錄使用的磁碟空間 (以位元組為單位)	需要 (dsk)	不需要	需要 (...destName)
磁碟使用率	使用的磁碟空間與保留的磁碟空間的比率。比率越高表示有越多磁碟空間被用來保留使用中訊息	需要 (dsk)	不需要	需要 (...destName)

1. 如需度量主題目標的名稱，請參閱第 199 頁表 10-7。

- 附錄 A Message Queue 資料的作業系統特定位置
- 附錄 B Message Queue 介面的穩定性
- 附錄 C HTTP/HTTPS 支援

Message Queue 資料的作業系統特定位置

Sun Java System Message Queue 資料儲存在不同作業系統的不同位置中，如以下各節所示。

本附錄提供下列作業系統上各種 Message Queue 資料的位置：

- [第 334 頁「Solaris」](#)
- [第 335 頁「Linux」](#)
- [第 336 頁「Windows」](#)

下列表格中，*instanceName* 可辨別與資料相關的代理程式實例名稱。

Solaris

表 A-1 顯示 Solaris 作業系統上 Message Queue 資料的位置。

如果您透過獨立版本的 Sun Java System Application Server 在 Solaris 上使用 Message Queue，其目錄結構將與 第 336 頁「Windows」中所描述的結構相似。

表 A-1 Solaris 上 Message Queue 資料的位置

資料種類	Solaris 上的位置
代理程式實例配置內容	<code>/var/imq/instances/instanceName/props/config.properties</code>
代理程式配置檔案範本	<code>/usr/share/lib/imq/props/broker/</code>
永久性儲存 (訊息、目標、長期訂閱和作業事件)	<code>/var/imq/instances/instanceName/fs350/</code> 或 JDBC 可存取的資料儲存
代理程式實例記錄檔目錄 (預設位置)	<code>/var/imq/instances/instanceName/log/</code>
受管理物件 (物件儲存)	您選擇的本機目錄 或 LDAP 伺服器
安全性：使用者儲存庫	<code>/var/imq/instances/instanceName/etc/passwd</code> 或 LDAP 伺服器
安全性：存取控制檔 (預設位置)	<code>/var/imq/instances/instanceName/etc/accesscontrol.properties</code>
安全性：密碼檔案目錄 (預設位置)	<code>/var/imq/instances/instanceName/etc/</code>
安全性：範例密碼檔案	<code>/etc/imq/passfile.sample</code>
安全性：代理程式的密鑰儲存檔案位置	<code>/etc/imq/</code>
JavaDoc API 文件	<code>/usr/share/javadoc/imq/index.html</code>
範例應用程式與配置	<code>/usr/demo/imq/</code>
Java 歸檔 (.jar)、Web 歸檔 (.war) 與資源介面歸檔 (.rar) 檔案	<code>/usr/share/lib/</code>

Linux

表 A-2 顯示 Linux 作業系統上 Message Queue 資料的位置。

表 A-2 Linux 上 Message Queue 資料的位置

資料種類	Windows 上的位置
代理程式實例配置內容	/var/opt/sun/mq/instances/ <i>instanceName</i> /props/ config.properties
代理程式配置檔案範本	/opt/sun/mq/private/share/lib/props/
永久性儲存 (訊息、目標、長期訂閱和作業事件)	/var/opt/sun/mq/instances/ <i>instanceName</i> /fs350/ 或 JDBC 可存取的資料儲存
代理程式實例記錄檔目錄 (預設位置)	/var/opt/sun/mq/instances/ <i>instanceName</i> /log/
受管理物件 (物件儲存)	您選擇的本機目錄或 LDAP 伺服器
安全性：使用者儲存庫	/var/opt/sun/mq/instances/ <i>instanceName</i> /etc/passwd 或 LDAP 伺服器
安全性：存取控制檔 (預設位置)	/var/opt/sun/mq/instances/ <i>instanceName</i> /etc/ accesscontrol.properties
安全性：密碼檔案目錄 (預設位置)	/var/opt/sun/mq/instances/ <i>instanceName</i> /etc/
安全性：範例密碼檔案	/etc/opt/sun/mq/passfile.sample
安全性：代理程式的密鑰儲存 檔案位置	/etc/opt/sun/mq/
JavaDoc API 文件	/opt/sun/mq/javadoc/index.html
範例應用程式與配置	/opt/sun/mq/examples/
Java 歸檔 (.jar)、Web 歸檔 (.war) 與資源介面歸檔 (.rar) 檔案	/opt/sun/mq/share/lib/
共用程式庫 (.so) 檔案	/opt/sun/mq/lib/

Windows

表 A-3 顯示 Windows 作業系統上 Message Queue 資料的位置。

當 Message Queue 隨附在獨立版本的 Sun Java System Application Server 時，此表格也會顯示 Solaris 上 Message Queue 資料的位置。該版本的 Application Server 並不是隨附於 Solaris 或 Sun Java Enterprise System。使用表 A-3 中的路徑名稱，但要將斜線字元的方向從 Windows 反斜線 (\) 變更為 Solaris 正斜線 (/)。如需更多資訊，請參閱第 25 頁表 3 中 IMQ_HOME 和 IMQ_VARHOME 的定義。

表 A-3 Windows 上 Message Queue 資料的位置

資料種類	Windows 上的位置
代理程式實例配置內容	IMQ_VARHOME\instances\ <i>instanceName</i> \props\ config.properties
代理程式配置檔案範本	IMQ_HOME\lib\props\broker\
永久性儲存 (訊息、目標、長期 訂閱和作業事件)	IMQ_VARHOME\instances\ <i>instanceName</i> \fs350\ 或 JDBC 可存取的資料儲存
代理程式實例記錄檔目錄 (預設 位置)	IMQ_VARHOME\instances\ <i>instanceName</i> \log\
受管理物件 (物件儲存)	您選擇的本機目錄 或 LDAP 伺服器
安全性：使用者儲存庫	IMQ_VARHOME\instances\ <i>instanceName</i> \etc\ passwd 或 LDAP 伺服器
安全性：存取控制檔 (預設值)	IMQ_VARHOME\instances\ <i>instanceName</i> \ etc\accesscontrol.properties
安全性：密碼檔案目錄 (預設位置)	IMQ_HOME\etc\ etc\password.properties
安全性：範例密碼檔案	IMQ_HOME\etc\passfile.sample
安全性：代理程式的密鑰儲存檔 案位置	IMQ_HOME\etc\ etc\secret.properties
JavaDoc API 文件	IMQ_HOME\javadoc\index.html
範例應用程式與配置	IMQ_HOME\demo\ demo\
Java 歸檔 (.jar)、Web 歸檔 (.war) 與資源介面歸檔 (.rar) 檔 案	IMQ_HOME\lib\ lib\

Message Queue 介面的穩定性

Sun Java System Message Queue 使用的許多介面可能對管理員自動管理工作很有用。本附錄可根據穩定性對介面進行分類。介面如果越穩定，則越不可能在下一版本的產品中進行變更。

附錄中沒有列出來的介面都是私有的介面，且不提供給客戶使用。

表 B-1 描述穩定性分類方案。

表 B-1 介面穩定性分類方案

分類	說明
私有	不由客戶直接使用。可在任何版次中變更或移除。
改進中	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。

表 B-2 列出各種介面和介面的分類。

表 B-2 Message Queue 介面的穩定性

介面	分類
命令行介面	
imqbrokerd 命令行介面	改進中
imqadmin 命令行介面	不穩定
imqcmd 命令行介面	改進中
imqdbmgr 命令行介面	不穩定
imqkeytool 命令行介面	改進中
imqobjmgr 命令行介面	改進中
imqusermgr 命令行介面	不穩定
從 imqbrokerd、imqadmin、imqcmd、imqdbmgr、imqkeytool、imqobjmgr 和 imqusermgr 的輸出	不穩定
指令	
imqobjmgr 指令檔案	改進中
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
API	
JMS API (javax.jms)	標準
JAXM API (javax.xml)	標準
C-API	改進中
C-API 環境變數	不穩定
基於訊息監視的 API	改進中
受管理物件 API (com.sun.messaging)	改進中

表 B-2 Message Queue 介面的穩定性 (續)

介面	分類
JAR 檔案和 WAR 檔案	
imq.jar 位置和名稱	穩定
jms.jar 位置和名稱	改進中
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	改進中
imqhttp.war 位置和名稱	改進中
imqhttps.war 位置和名稱	改進中
imqjmsra.rar 位置和名稱	改進中
imqxm.jar 位置和名稱	改進中
jaxm-api.jar 位置和名稱	改進中
saaj-api.jar 位置和名稱	改進中
saaj-impl.jar 位置和名稱	改進中
activation.jar 位置和名稱	改進中
mail.jar 位置和名稱	改進中
dom4j.jar 位置和名稱	私有
fscontext.jar 位置和名稱	不穩定
檔案	
代理程式記錄檔的位置和內容格式	不穩定
密碼檔案	不穩定
accesscontrol.properties 檔案	不穩定
系統目標	
mq.sys.dmq 目標	穩定
mq.metrics.* 目標	改進中
配置特性	
Message Queue JMS 資源介面配置特性	改進中
Message Queue JMS 資源介面 JavaBean 和 ActivationSpec 配置特性	改進中

表 B-2 Message Queue 介面的穩定性 (續)

介面	分類
訊息特性和格式	
停用的訊息佇列訊息特性，JMSXDeliveryCount	標準
停用的訊息佇列訊息特性，JMS_SUN_*	改進中
Message Queue 用戶端訊息特性：JMS_SUN_*	改進中
度量或監視訊息的 JMS 訊息格式	改進中
其他	
Message Queue JMS 資源介面套裝軟體，com.sun.messaging.jms.ra	改進中
儲存永久性訊息的 JDBC 模式	改進中

HTTP/HTTPS 支援

Message Queue Enterprise Edition 包含了對 Java 用戶端的支援，以藉由 HTTP 或安全 HTTP (HTTPS) 傳輸，而不是直接的 TCP 連線，與代理程式進行通訊。HTTP/HTTPS 支援無法提供給 C 用戶端。

本附錄描述用於啓用此支援的架構，並介紹允許用戶端使用基於 HTTP 的連線進行 Message Queue 訊息傳送所需的設定工作。其包含以下各節：

- [第 342 頁「HTTP/HTTPS 支援架構」](#)
- [第 343 頁「啓用 HTTP 支援」](#)
- [第 352 頁「啓用 HTTPS 支援」](#)
- [第 364 頁「疑難排解」](#)

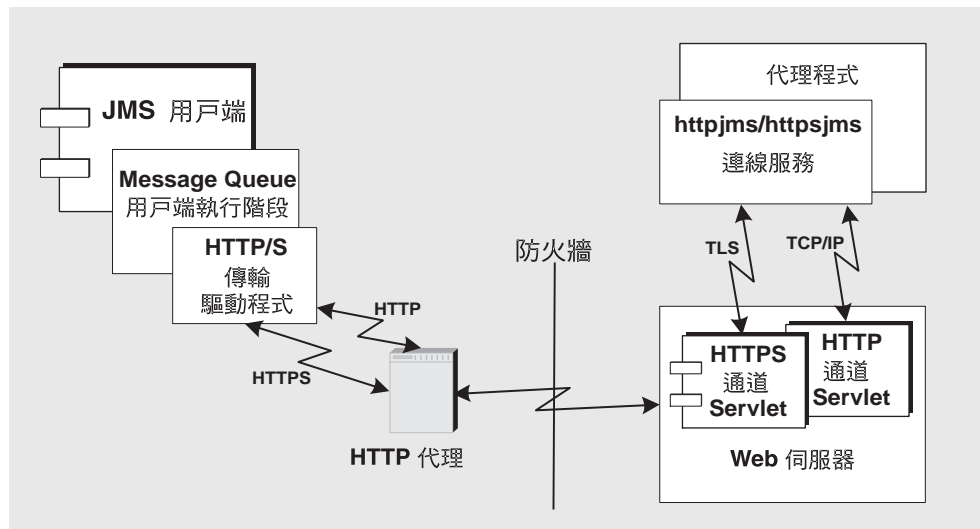
HTTP/HTTPS 支援架構

Message Queue 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 342 頁圖 C-1 顯示提供 HTTP/HTTPS 支援所需的主要元件。

- 在用戶端，HTTP 或 HTTPS 傳輸驅動程式將 Message Queue 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，用戶端可使用 HTTP 代理伺服器與代理程式進行通訊。代理的位址在啟動用戶端時使用命令行選項指定。請參閱第 347 頁「使用 HTTP 代理」，以獲得更多資訊。
- HTTP 或 HTTPS 通道 Servlet (均隨附於 Message Queue) 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 也會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 C-1 HTTP/HTTPS 支援架構



- 在代理程式端，會展開 httpjms 或 httpsjms 連線服務並多路傳輸從相應通道 Servlet 內送的訊息。

- 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，且不通知用戶端。若要避免這種情況，請務必重新啓動代理程式。

如圖 C-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。

可透過已啟用 SSL 的通道 Servlet (Message Queue 的 HTTPS 通道 Servlet) 提供與代理程式的安全連線。此通道 Servlet 將自身簽名的憑證傳送至請求連線的任何代理程式。代理程式使用此憑證來設定與 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。

啟用 HTTP 支援

以下各節描述啟用 HTTP 支援所需採取的步驟。

► 啟用 HTTP 支援

1. 在 Web Server 上部署 HTTP 通道 Servlet。
2. 配置代理程式的 httpjms 連線服務並啓動代理程式。
3. 配置 HTTP 連線。

步驟 1. 在 Web Server 上部署 HTTP 通道 Servlet

在 Web Server 上部署 HTTP 通道 Servlet 有兩種常用方法：

- 將其部署為 JAR 檔案 - 用於支援 Servlet 2.1 或較低版本的 Web Server
- 將其部署為 Web 歸檔 (WAR) 檔案 - 用於支援 Servlet 2.2 或較高版本的 Web Server

部署為 JAR 檔案

部署 Message Queue 通道 Servlet 由三個部分組成：使主機 Web Server 存取適當的 JAR 檔案，配置此 Web Server 以在啓動時載入 Servlet，以及指定 Servlet URL 的上下文根部分。

通道 Servlet JAR 檔案 (imqservlet.jar) 包含 HTTP 通道 Servlet 需要的所有類別，您可以在某個目錄 (目錄位置因作業系統而異) 中找到這個檔案 (請參閱附錄 A 「Message Queue 資料的作業系統特定位置」)。

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.HttpTunnelServlet
```

Web Server 必須可查看 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。

您也必須配置 Web Server，以便在啟動時載入此 Servlet。另外，您可能必須指定 Servlet URL 的上下文根部分 (請參閱第 347 頁 「範例 1：在 Sun Java System Web Server 上部署 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。

部署為 Web 歸檔檔案

將 HTTP 通道 Servlet 部署為 WAR 檔案，需要使用 Web Server 提供的部署機制。HTTP 通道 Servlet WAR 檔案 (imqhttp.war) 位於包含 .JAR、.WAR 與 .RAR 檔案的目錄中，目錄位置因作業系統而異 (請參閱附錄 A 「Message Queue 資料的作業系統特定位置」)。

WAR 檔案包括部署描述元，此描述元包含 Web Server 載入和執行 Servlet 所需的基本配置資訊。因 Web Server 的不同，您可能也必須指定 Servlet URL 的上下文根部分 (請參閱第 350 頁 「範例 2：在 Sun Java System Application Server 7.0 上部署 HTTP 通道 Servlet」)。

步驟 2. 配置 httpjms 連線服務

預設情況下，代理程式不會啟動 HTTP 支援，所以您必須重新配置代理程式，以啟動 httpjms 連線服務。一旦重新配置，此代理程式便可啟動，如第 64 頁 「以互動方式啟動代理程式」 中所述。

► 啓動 **httpjms** 連線服務

1. 開啓代理程式的實例配置檔案。

實例配置檔案儲存於以代理程式實例名稱 (*instanceName*) 作為辨別依據的目錄中，且代理程式實例與配置檔案相互關聯 (請參閱附錄 A 「[Message Queue 資料的作業系統特定位置](#)」)：

```
.../instances/instanceName/props/config.properties
```

2. 將 `httpjms` 值新增至 `imq.service.activelist` 特性：

```
imq.service.activelist=jms,admin,httpjms
```

啓動時，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` 連線服務特性。

您還可以重新配置 `pullPeriod` 特性，以提昇效能。第 345 頁表 C-1 中詳細介紹了 `httpjms` 連線服務配置特性。

表 C-1 `httpjms` 連線服務特性

特性名稱	說明
<code>imq.httpjms.http.servletHost</code>	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。) 預設值： <code>localhost</code>
<code>imq.httpjms.http.servletPort</code>	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特性。) 預設值： <code>7675</code>
<code>imq.httpjms.http.pullPeriod</code>	指定用戶端發出從代理程式移除訊息的 HTTP 請求的間隔時間 (以秒為單位)。(請注意，此特性於代理程式上進行設定，且傳遞至用戶端執行階段。) 如果值為零或負數，則用戶端將始終擱置一個 HTTP 請求，以準備儘快移除訊息。若用戶端數目過大，可能會導致 Web Server 資源大量外流，且伺服器可能沒有回應。在此類情況下，您應將 <code>pullPeriod</code> 特性設定為正的秒數。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。預設值： <code>-1</code>
<code>imq.httpjms.http.connectionTimeout</code>	指定用戶端執行階段在系統拋出異常前，等待從 HTTP 通道 Servlet 發出回應的時間 (以秒為單位)。(請注意，此特性於代理程式上進行設定，且傳遞至用戶端執行階段。) 此特性也會指定中斷連線前，代理程式與 HTTP 通道 Servlet 進行通訊後，代理程式等待的時間。因為代理程式與通道 Servlet 無法得知存取 HTTP Servlet 的用戶端是否已異常中斷，在此情況下，必須設定逾時時間。預設值： <code>60</code>

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。

配置連線工廠

若要啓用 HTTP 支援，您必須將連線工廠的 `imqAddressList` 屬性設定至 HTTP 通道 Servlet URL。HTTP 通道 Servlet URL 的一般語法如下：

```
http://hostName:port/contextRoot/tunnel
```

`hostName:port` 為代管 HTTP 通道 Servlet 之 Web Server 的名稱和連接埠；`contextRoot` 為在 Web Server 上部署通道 Servlet 的路徑設定。

如需連線工廠屬性的一般資訊與 `imqAddressList` 屬性的特定資訊，請參閱 *Message Queue Developer's Guide for Java Clients*。

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 173 頁「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。
- 使用可啓動用戶端的指令的 `-D` 選項（請參閱 *Message Queue Developer's Guide for Java Clients*）。
- 當您有計劃藉由用戶端程式碼建立連線工廠後，請使用 API 呼叫設定它的屬性（請參閱 *Message Queue Developer's Guide for Java Clients*）。

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。如果多重代理程式實例正共用一個通道 Servlet，您必須配置 `imqAddressList` 連線工廠屬性，配置內容如下：

```
http://hostName:port/contextRoot/tunnel?ServerName=bkrHostName:instanceName
```

`bkrHostName` 為代理程式實例主機名稱；`instanceName` 為您用戶端存取的特定代理程式實例名稱。

若要檢查您是否已為 `bkrHostName` 和 `instanceName` 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 通道 Servlet 準備就緒。
Servlet 啓動時間：Thu May 30 01:08:18 PDT 2002
接受連接埠上來自代理程式的 TCP 連線：7675
代理程式可用總數 = 2
代理程式清單：
    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` 系統特性設定為代理伺服器主機名稱。
- 將 `http.proxyPort` 系統特性設定為代理伺服器連接埠號。

您可以使用可啓動用戶端應用程式之指令的 `-D` 選項來設定這些特性。

範例 1：在 Sun Java System Web Server 上部署 HTTP 通道 Servlet

本節描述如何在 Sun Java System Web Server 上將 HTTP 通道 Servlet 部署為 JAR 檔案和 WAR 檔案。您使用的方法取決於 Sun Java System 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 檔案部署。

部署為 JAR 檔案

以下說明參照了使用基於瀏覽器的管理 GUI 之 Sun Java System Web Server 6.1 上的部署。此程序包含以下常用步驟：

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 部署。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，指定通道 Servlet 的名稱。
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.HttpTunnelServlet
5. 在 [Servlet 類別路徑] 欄位中輸入 imqservlet.jar 的完整路徑。例如：
/usr/share/lib/imq/imqservlet.jar (Solaris)
/opt/sun/mq/share/lib/imqservlet.jar (Linux)
IMQ_HOME/lib/imqservlet.jar (Windows)
6. 在 [Servlet 引數] 欄位中，輸入任何選擇性引數，如表 C-2 中所示：

表 C-2 用於部署 HTTP 通道 Servlet JAR 檔案的 Servlet 引數

引數	預設值	參照
servletHost	所有主機	請參閱第 345 頁表 C-1
servletPort	7675	請參閱第 345 頁表 C-1

如果要使用這兩個引數，請使用逗號將它們分隔：

```
servletPort=portNumber, servletHost=...
```

servletHost 和 servletPort 引數僅套用於 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特性（請參閱第 345 頁表 C-1），例如：

```
imq.httppjms.http.servletPort
```

配置 Servlet 虛擬路徑 (Servlet URL)

► 若要配置通道 Servlet 的虛擬路徑 (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。

「虛擬路徑」為通道 Servlet URL 的 `/contextRoot/tunnel` 的一部份：

```
http://hostName:port/contextRoot/tunnel
```

舉例來說，如果您將 `contextRoot` 設定至 `imq`，那麼 [虛擬路徑] 欄位則為：

```
/imq/tunnel
```

4. 將 [Servlet 名稱] 欄位設定為與第 348 頁「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 Servlet] 欄位中，輸入與第 348 頁「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

部署為 WAR 檔案

以下說明指的是 Sun Java System Web Server 6.0 Service Pack 2 上的部署。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 部署。它應顯示狀態資訊。

► 將 HTTP 通道 Servlet 部署為 WAR 檔案

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 (例如 `defaultClass`)，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 [部署 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 `imghttp.war` 檔案的位置。您可以在某個目錄（目錄位置因作業系統而異）中找到這個檔案（請參閱附錄 A 「[Message Queue 資料的作業系統特定位置](#)」）。
8. 在 [應用程式 URI] 欄位中輸入路徑。

[應用程式 URI] 欄位值為通道 Servlet URL 的一部份：

```
http://hostName:port/contextRoot/tunnel
```

舉例來說，如果您將 `contextRoot` 設定至 `img`，[應用程式 URI] 欄位應為：

```
/img
```

9. 輸入應部署 Servlet 的安裝目錄路徑（通常位於 Sun Java System Web Server 安裝根目錄下）。
10. 按一下 [確定]。
11. 重新啟動 Web Server 實例。

此時，Servlet 可從以下位址獲得：

```
http://hostName:port/contextRoot/tunnel
```

此時，用戶端可使用此 URL，以使用 HTTP 連線來連線至訊息服務。

範例 2：在 Sun Java System Application Server 7.0 上部署 HTTP 通道 Servlet

本節描述如何在 Sun Java System Application Server 7.0 上將 HTTP 通道 Servlet 部署為 WAR 檔案。

您需要執行兩個步驟：

- 使用 Application Server 7.0 部署工具來部署 HTTP 通道 Servlet
- 修改應用程式伺服器實例的 `server.policy` 檔案

使用部署工具

► 在 Application Server 7.0 環境中部署 HTTP 通道 Servlet

1. 在基於瀏覽器的管理 GUI 中選取
應用程式伺服器 > 實例 > `server1` > 應用程式 > Web 應用程式。
2. 按一下 [部署] 按鈕。
3. 在 [檔案路徑:] 文字欄位輸入 HTTP 通道 Servlet WAR 檔案 (`imqhttp.war`) 的位置。

`imqhttp.war` 檔案的位置會因您的作業系統而有差異 (請參閱附錄 A 「[Message Queue 資料的作業系統特定位置](#)」)。

4. 按一下 [確定]。
5. 在下一個畫面，於 [上下文根] 文字欄位中設定數值。

[上下文根] 欄位值為通道 Servlet URL 的 `/contextRoot` 的一部份：

```
http://hostName:port/contextRoot/tunnel
```

例如，您可以將 [上下文根] 欄位設定為 `/imq`。

6. 按一下 [確定]。

依預設，接著會出現下一個畫面，表示通道 Servlet 已成功部署在：

```
/var/opt/SUNWappserver7/domains/domain1/server1/applications/  
j2ee-modules/imqhttp_1
```

此時，Servlet 可從以下位址獲得：

```
http://hostName:port/contextRoot/tunnel
```

此時，用戶端可使用此 URL，以使用 HTTP 連線來連線至訊息服務。

修改 `server.policy` 檔案

Application Server 7.0 會強制一組預設安全策略，除非修改的檔案可以阻止 HTTP 通道 Servlet 接受來自 Message Queue 代理程式的連線。

每個應用程式伺服器實例，皆有一個包含安全策略或規則的檔案。例如，Solaris 上用於 `server1` 實例的檔案位置爲：

```
/var/opt/SUNWappserver7/domains/domain1/server1/config/  
server.policy
```

若要讓通道 Servlet 接受來自 Message Queue 代理程式的連線，則必須在此檔案新增其他項目。

► **若要修改 Application Server 的伺服器策略檔案**

1. 開啓伺服器策略檔案。
2. 新增以下項目：

```
grant codeBase  
"file:/var/opt/SUNWappserver7/domains/domain1/server1/  
  applications/j2ee-modules/imqhttp_1/-"  
{  
  permission java.net.SocketPermission "*",  
    "connect,accept,resolve";  
};
```

啓用 HTTPS 支援

以下各節描述啓用 HTTPS 支援所需採取的步驟。它們與第 343 頁「啓用 HTTP 支援」中的那些步驟，以及產生和存取 SSL 憑證所需的步驟相似。

► **啓用 HTTPS 支援**

1. 爲 HTTPS 通道 Servlet 產生自身簽名的憑證。
2. 在 Web Server 上部署 HTTPS 通道 Servlet。
3. 配置代理程式的 `httpsjms` 連線服務並啓動代理程式。
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 產生自身簽名的憑證

在假設用戶端正與已知且可信任的伺服器通訊的情況下，Message Queue 的 SSL 支援可保護線上資料。因此，僅使用自身簽名的伺服器憑證即可實作 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 `imqkeytool` 公用程式，以便為通道 Servlet 產生自身簽名的憑證。在命令提示字元下輸入以下內容：

```
imqkeytool -servlet keystore_location
```

此公用程式將提示您提供它所需的資訊。(在 Unix 系統上，為了擁有建立密鑰儲存的許可權，您可能需要以超級使用者 (root) 的身份執行 `imqkeytool`。)

首先，`imqkeytool` 會提示您提供密鑰儲存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)；您應按 [返回] 以回應此提示：這會使鍵密碼與密鑰儲存密碼相同。

備註 請記住您所提供的密碼 - 稍後您必須將此密碼提供給通道 Servlet，以便它可開啓密鑰儲存。

執行 `imqkeytool` 可執行 JDK `keytool` 公用程式，以產生自身簽名的憑證，並將其放在位於 `keystore_location` 引數所指定的 Message Queue 密鑰儲存檔案中。(此密鑰儲存的格式與 JDK1.2 `keytool` 支援的格式相同。)

備註 HTTPS 通道 Servlet 必須可查看此密鑰儲存。請確定您將位於 `keystore_location` 中已產生的密鑰儲存移動 / 複製到 HTTPS 通道 Servlet 可存取的位置 (請參閱第 353 頁「[步驟 2. 在 Web Server 上部署 HTTPS 通道 Servlet](#)」)。

步驟 2. 在 Web Server 上部署 HTTPS 通道 Servlet

在 Web Server 上部署 HTTPS 通道 Servlet 有兩種常用方法：

- 將其部署為 JAR 檔案 - 用於支援 Servlet 2.1 或較低版本的 Web Server

- 將其部署為 Web 歸檔 (WAR) 檔案 - 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。

部署為 JAR 檔案

部署 Message Queue 通道 Servlet 由三個部分組成：使主機 Web Server 存取適當的 JAR 檔案，配置此 Web Server 以在啟動時載入 Servlet，以及指定 Servlet URL 的上下文根部分。

通道 Servlet JAR 檔案 (imqservlet.jar) 包含 HTTPS 通道 Servlet 需要的所有類別，您可以在某個目錄 (目錄位置因作業系統而異) 中找到這個檔案 (請參閱[附錄 A](#)「[Message Queue 資料的作業系統特定位置](#)」)。

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 必須可查看 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。

您也必須配置 Web Server，以便在啟動時載入此 Servlet。另外，您可能必須指定 Servlet URL 的上下文根部分 (請參閱[第 359 頁](#)「[範例 3：在 Sun Java System Web Server 上部署 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憑證的位置和密碼，以便 HTTPS 通道 Servlet 使用此憑證來建立與代理程式的安全連線。您必須將[第 353 頁](#)「[步驟 1. 為 HTTPS 通道 Servlet 產生自身簽名的憑證](#)」中建立的密鑰儲存在 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。

部署為 Web 歸檔檔案

將 HTTPS 通道 Servlet 部署為 WAR 檔案，需要使用 Web Server 提供的部署機制。HTTPS 通道 Servlet WAR 檔案 (imqhttps.war) 位於您作業系統的某個目錄中，目錄位置因作業系統而異 (請參閱[附錄 A](#)「[Message Queue 資料的作業系統特定位置](#)」)。

WAR 檔案包括部署描述元，此描述元包含 Web Server 載入和執行 Servlet 所需的基本配置資訊。因 Web Server 的不同，您可能必須指定 Servlet URL 的上下文根部分 (請參閱第 363 頁「範例 4：在 Sun Java System Application Server 7.0 上部署 HTTPS 通道 Servlet」)。

但是，`imqhttps.war` 檔案的部署描述元無法知道您已將通道 Servlet 所需的密鑰儲存檔案放在何處 (請參閱第 353 頁「步驟 1. 為 HTTPS 通道 Servlet 產生自身簽名的憑證」)。這要求您編輯通道 Servlet 的部署描述元 (XML 檔案)，以在部署 `imqhttps.war` 檔案之前指定密鑰儲存的位置。

步驟 3. 配置 httpsjms 連線服務

預設情況下，代理程式不會啟動 HTTPS 支援，所以您必須重新配置代理程式，以啟動 `httpsjms` 連線服務。一旦重新配置，此代理程式便可啟動，如第 64 頁「以互動方式啟動代理程式」中所述。

► 啟動 httpsjms 連線服務

1. 開啓代理程式的實例配置檔案。

實例配置檔案儲存於以代理程式實例名稱 (*instanceName*) 作為辨別依據的目錄中，且代理程式實例與配置檔案相互關聯 (請參閱附錄 A「Message Queue 資料的作業系統特定位置」)：

```
.../instances/instanceName/props/config.properties
```

2. 將 `httpsjms` 值新增至 `imq.service.activelist` 特性：

```
imq.service.activelist=jms,admin,httpsjms
```

啟動時，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` 連線服務特性。

您還可以重新配置 `pullPeriod` 特性，以提昇效能。表 C-3 中詳細介紹了 `httpsjms` 連線服務配置特性。

表 C-3 httpsjms 連線服務特性

特性名稱	說明
<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。)預設值： <code>localhost</code>

表 C-3 httpsjms 連線服務特性 (續)

特性名稱	說明
imq.httpsjms.https.servletPort	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特性。) 預設值：7674
imq.httpsjms.https.pullPeriod	指定每個用戶端發出從代理程式中移除訊息的 HTTP 請求的間隔時間 (以秒為單位)。(請注意，此特性於代理程式上進行設定，且傳遞至用戶端執行階段。) 如果值為零或負數，則用戶端將始終擱置一個 HTTP 請求，以準備儘快移除訊息。若用戶端數目過大，可能會導致 Web Server 資源大量外流，且伺服器可能沒有回應。在此類情況下，您應將 pullPeriod 特性設定為正的秒數。此作業將設定用戶端的 HTTP 傳輸驅動程式在發出後繼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。預設值：-1
imq.httpsjms.https.connectionTimeout	指定用戶端執行階段在系統拋出異常前，等待從 HTTPS 通道 Servlet 發出回應的時間 (以秒為單位)。(請注意，此特性於代理程式上進行設定，且傳遞至用戶端執行階段。) 此特性也會指定中斷連線前，代理程式與 HTTPS 通道 Servlet 進行通訊後，代理程式等待的時間。因為代理程式與通道 Servlet 無法得知存取 HTTPS Servlet 的用戶端是否已異常中斷，在此情況下，必須設定逾時時間。預設值：60

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，並且還必須具有根憑證。SSL 程式庫隨附於 JDK 1.4。如果您有較低版本的 JDK，請參閱「[配置 JSSE](#)」，否則請繼續執行「[匯入根憑證](#)」。

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 特性：

```
java.protocol.handler.pkgs=com.sun.net.ssl.internal.www.protocol
```

匯入根憑證

如果 CA (為您的 Web Server 憑證簽名) 的根憑證依預設不在信任資料庫中，或者您使用的是專用 Web Server 憑證，則您必須將此憑證新增至信任資料庫。如果出現此情況，請遵循以下說明，否則請參照「[配置連線工廠](#)」中的說明。

假設此憑證儲存在 `cert_file` 中，且 `trust_store_file` 為您的密鑰儲存，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：Trust this certificate? (是否信任此憑證?)

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 特性：

```
javax.net.ssl.trustStore=trust_store_file
javax.net.ssl.trustStorePassword=trust_store_passwd
```

配置連線工廠

若要啟用 HTTPS 支援，您必須將連線工廠的 `imqAddressList` 屬性設定至 HTTPS 通道 Servlet URL。HTTPS 通道 Servlet URL 的一般語法如下：

```
https://hostName:port/contextRoot/tunnel
```

`hostName:port` 為代管 HTTPS 通道 Servlet 之 Web Server 的名稱和連接埠；

`contextRoot` 為在 Web Server 上部署通道 Servlet 的路徑設定。

如需連線工廠屬性的一般資訊與 `imqAddressList` 屬性的特定資訊，請參閱 [Message Queue Developer's Guide for Java Clients](#)。

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 173 頁「[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 *Message Queue Developer's Guide for Java Clients*)。
- 當您有計劃藉由用戶端應用程式碼建立連線工廠後，請使用 API 呼叫設定它的屬性 (請參閱 *Message Queue Developer's Guide for Java Clients*)。

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。如果多重代理程式實例正共用一個通道 Servlet，您必須配置 `imqAddressList` 連線工廠屬性，配置內容如下：

```
https://hostName:port/contextRoot/tunnel?ServerName=bkrHostName:instanceName
```

`bkrHostName` 為代理程式實例主機名稱；`instanceName` 為您用戶端存取的特定代理程式實例名稱。

若要檢查您是否已為 `bkrhostName` 和 `instanceName` 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 通道 Servlet 準備就緒。
Servlet 啟動時間：Thu May 30 01:08:18 PDT 2002
接受連接埠上來自代理程式的安全連線： 7674
代理程式可用總數 = 2
代理程式清單：
  jpgserv:broker2
  cochin:broker1
```

使用 HTTP 代理

如果您要使用 HTTP 代理存取 HTTPS 通道 Servlet：

- 將 `http.proxyHost` 系統特性設定為代理伺服器主機名稱。
- 將 `http.proxyPort` 系統特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` 選項來設定這些特性。

範例 3：在 Sun Java System Web Server 上部署 HTTPS 通道 Servlet

本節描述如何在 Sun Java System Web Server 上將 HTTPS 通道 Servlet 部署為 JAR 檔案和 WAR 檔案。您使用的方法取決於 Sun Java System 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 檔案部署。

部署為 JAR 檔案

以下說明參照了使用基於瀏覽器的管理 GUI 之 Sun Java System Web Server 6.1 上的部署。此程序包含以下常用步驟：

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 HTTPS 通道 Servlet 部署。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：


```
/usr/share/lib/imq/imqervlet.jar (Solaris)  
/opt/sun/mq/share/lib/imqervlet.jar (Linux)  
IMQ_HOME/lib/imqervlet.jar (Windows)
```
6. 在 [Servlet 引數] 欄位中，輸入所需引數和選擇性引數，如表 C-4 中所示。

表 C-4 用於部署 HTTPS 通道 Servlet JAR 檔案的 Servlet 引數

引數	預設值	需要嗎？
keystoreLocation	無	需要
keystorePassword	無	需要
servletHost	所有主機	不需要
servletPort	7674	不需要

使用逗號將引數分隔。例如：

```
keystoreLocation=keystore_location, keystorePassword=keystore_password,
servletPort=portnumber
```

servletHost 和 servletPort 引數僅套用於 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特性（請參閱第 355 頁表 C-3）。例如：

```
imq.httpsjms.https.servletPort
```

配置 Servlet 虛擬路徑 (Servlet URL)

► 若要配置通道 Servlet 的虛擬路徑 (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。

「虛擬路徑」為通道 Servlet URL 的 `/contextRoot/tunnel` 的一部份：

```
https://hostName:port/contextRoot/tunnel
```

舉例來說，如果您將 `contextRoot` 設定至 `imq`，那麼 [虛擬路徑] 欄位則為：

```
/imq/tunnel
```

4. 將 [Servlet 名稱] 欄位設定為與第 359 頁「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 Servlet] 欄位中，輸入與第 359 頁「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

部署為 WAR 檔案

以下說明指的是 Sun Java System Web Server 6.0 Service Pack 2 上的部署。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 部署。它應顯示狀態資訊。

在部署 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 `jcrt.jar` 複製到 `IWS60_TOPDIR/bin/https/jre/lib/ext`。

此外，在部署 HTTPS 通道 Servlet 之前，您必須修改其部署描述元，以指向您放置密鑰儲存檔案的位置，並指定密鑰儲存密碼。

► 修改 HTTPS 通道 Servlet WAR 檔案

1. 將 WAR 檔案複製到暫存目錄。

```
cp /usr/share/lib/imq/imqhttps.war /tmp (Solaris)
```

```
cp /opt/sun/mq/share/lib/imqhttps.war /tmp (Linux)
```

```
cp IMQ_HOME/lib/imqhttps.war /tmp (Windows)
```

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 imqhttps.war
```

4. 列出 WAR 檔案的部署描述元。

```
$ ls -l WEB-INF/web.xml
```

5. 編輯 web.xml 檔案,以便爲 keystoreLocation 和 keystorePassword 引數 (如有必要,還包括 servletPort 和 servletHost 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 檔案來部署 HTTPS 通道 Servlet。(如果您擔心密鑰儲存密碼洩漏,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 部署爲 WAR 檔案

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 (例如 defaultClass),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 [部署 Web 應用程式]。
7. 爲 [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 361 頁「修改 HTTPS 通道 Servlet WAR 檔案」)。
8. 在 [應用程式 URI] 欄位中輸入路徑。

[應用程式 URI] 欄位值爲通道 Servlet URL 的 /contextRoot 的一部份:

```
https://hostName:port/contextRoot/tunnel
```

舉例來說,如果您將 contextRoot 設定至 imq, [應用程式 URI] 欄位應爲:

```
/imq
```

9. 輸入應部署 Servlet 的安裝目錄路徑 (通常位於 Sun Java System Web Server 安裝根目錄下)。
10. 按一下 [確定]。
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。

範例 4：在 Sun Java System Application Server 7.0 上部署 HTTPS 通道 Servlet

本節描述如何在 Sun Java System Application Server 7.0 上將 HTTPS 通道 Servlet 部署為 WAR 檔案。

您需要執行兩個步驟：

- 使用 Application Server 7.0 部署工具來部署 HTTPS 通道 Servlet
- 修改應用程式伺服器實例的 `server.policy` 檔案

使用部署工具

► 在 Application Server 7.0 環境中部署 HTTPS 通道 Servlet

1. 在基於瀏覽器的管理 GUI 中選取
應用程式伺服器 > 實例 > `server1` > 應用程式 > Web 應用程式。
2. 按一下 [部署] 按鈕。
3. 在 [檔案路徑:] 文字欄位輸入 HTTPS 通道 Servlet WAR 檔案 (`imqhttps.war`) 的位置。

`imqhttps.war` 檔案的位置會因您的作業系統而有差異 (請參閱附錄 A「[Message Queue 資料的作業系統特定位置](#)」)。

4. 按一下 [確定]。
5. 在下一個畫面，於 [上下文根] 文字欄位中設定數值。

[上下文根] 欄位值為通道 Servlet URL 的 `/contextRoot` 的一部份：

```
https://hostName:port/contextRoot/tunnel
```

例如，您可以將 [上下文根] 欄位設定為：

```
/imq
```

6. 按一下 [確定]。

依預設，接著會出現下一個畫面，表示通道 Servlet 已成功部署在：

```
/var/opt/SUNWappserver7/domains/domain1/server1/applications/  
j2ee-modules/imqhttps_1
```

此時，Servlet 可從以下位址獲得：

```
https://hostName:port/contextRoot/tunnel
```

此時，用戶端可使用此 URL，以使用 HTTPS 連線而連線至訊息服務。

修改 server.policy 檔案

Application Server 7.0 會強制一組預設安全策略，除非修改的檔案可以阻止 HTTPS 通道 Servlet 接受來自 Message Queue 代理程式的連線。

每個應用程式伺服器實例，皆有一個包含安全策略或規則的檔案。例如，Solaris 上用於 server1 實例的檔案位置為：

```
/var/opt/SUNWappserver7/domains/domain1/server1/config/
server.policy
```

若要讓通道 Servlet 接受來自 Message Queue 代理程式的連線，則必須在此檔案新增其他項目。

► 若要修改 Application Server 的伺服器策略檔案

1. 開啓伺服器策略檔案。
2. 新增以下項目：

```
grant codeBase
"file:/var/opt/SUNWappserver7/domains/domain1/server1/
    applications/j2ee-modules/imqhttps_1/-"
{
    permission java.net.SocketPermission "*",
        "connect,accept,resolve";
};
```

疑難排解

本節描述 HTTP 或 HTTPS 連線可能發生的問題，並提供處理這些問題的方法。

伺服器或代理程式發生故障

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。不過，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，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，請確定代理程式已重新啓動。

用戶端無法透過通道 Servlet 連線

如果 HTTPS 用戶端無法透過通道 Servlet 連線代理程式，請執行以下作業：

1. 啓動 Servlet 和代理程式。
2. 使用瀏覽器透過 HTTPS 通道 Servlet URL 以手動存取 Servlet。
3. 使用以下管理指令暫停並繼續連線：

```
imqcmd pause svc -n httpsjms -u admin  
imqcmd resume svc -n httpsjms -u admin
```

當服務繼續時，HTTPS 用戶端應該就能透過通道 Servlet 連線代理程式。

字彙表

如需 Message Queue 專有名詞的詳細資訊，請參閱 Message Queue 技術概述中的字彙表。如需 Sun Java System 產品套件中使用的專有名詞的完整清單，請參閱 Java Enterprise System 字彙表 (<http://docs.sun.com/doc/819-1936>)。

A

acknowledgeMode 啓動規格屬性 321
ActivationSpec JavaBean 320
addressList 受管理連線工廠屬性 319
addressList 啓動規格屬性 321
addressList 資源介面屬性 318, 319
addressListBehavior 受管理連線工廠屬性 320
addressListBehavior 資源介面屬性 318
addressListIterations 受管理連線工廠屬性 320
addressListIterations 資源介面屬性 318
admin 使用者 133, 137, 139
ADMIN 服務類型 73
admin 連線服務 74, 109
admin 群組 135
Administration Console
啓動 40
anonymous 群組 135
API 文件 334, 335, 336
AUTOSTART 特性 66

C

clientID 受管理連線工廠屬性 320
clientId 啓動規格屬性 321, 322
config.properties 檔案 92, 182, 183, 184
connectionURL 資源介面屬性 318

customAcknowledgeMode 啓動規格屬性 321

D

default.properties 檔案 90
destination 啓動規格屬性 321
destinationType 啓動規格屬性 321, 322

E

endpointExceptionRedeliveryAttempts 啓動規格屬性 321, 322
/etc/hosts 檔案 (Linux) 182

G

guest 使用者 133

H

hosts 檔案 (Linux) 182
HTTP
支援架構 342

I 部分

- 代理 342
 - 連線服務, 請參閱 `httpjms` 連線服務
 - 傳輸驅動程式 342
 - HTTP 通道 Servlet
 - 部署 343
 - 關於 342
 - HTTP 連線
 - 支援 342
 - 多重代理程式, 用於 346
 - 通道 Servlet, 請參閱 HTTP 通道 Servlet
 - 請求間隔時間 345
 - `httpjms` 連線服務
 - 配置 344
 - 設定 343
 - 關於 74, 109
 - HTTPS
 - 支援架構 342
 - 連線服務, 請參閱 `httpsjms` 連線服務
 - HTTPS 通道 Servlet
 - 部署 353
 - 關於 342
 - HTTPS 連線
 - 支援 342
 - 多重代理程式, 用於 358
 - 通道 Servlet, 請參閱 HTTPS 通道 Servlet
 - 請求間隔時間 356
 - `httpsjms` 連線服務
 - 配置 355
 - 設定 352
 - 關於 74, 109
-
- `imq.accesscontrol.enabled` 特性 86, 282, 293
 - `imq.accesscontrol.file.filename` 特性 86, 282, 293
 - `imq.audit.enabled` 特性 282, 293
 - `imq.authentication.basic.user_repository` 特性 86, 282, 294
 - `imq.authentication.client.response.timeout` 特性 86, 282, 294
 - `imq.authentication.type` 特性 86, 282, 294
 - `imq.autocreate.destination.isLocalOnly` 特性 282, 288
 - `imq.autocreate.destination.limitBehavior` 特性 282, 288
 - `imq.autocreate.destination.maxBytesPerMsg` 特性 282, 288
 - `imq.autocreate.destination.maxCount` 特性 282, 288
 - `imq.autocreate.destination.maxNumMsgs` 特性 288
 - `imq.autocreate.destination.maxNumProducers` 特性 282, 288
 - `imq.autocreate.destination.maxTotalMsgBytes` 特性 282, 288
 - `imq.autocreate.destination.useDMQ` 特性 129, 282
 - `imq.autocreate.queue` 特性 105, 282, 288
 - `imq.autocreate.queue.consumerFlowLimit` 特性 282, 289
 - `imq.autocreate.queue.localDeliveryPreferred` 特性 282, 289
 - `imq.autocreate.queue.maxNumActiveConsumers` 特性 105, 282, 289
 - `imq.autocreate.queue.maxNumBackupConsumers` 特性 105, 282, 289
 - `imq.autocreate.topic` 特性 105, 282, 289
 - `imq.cluster.brokerlist` 特性 180, 181, 182, 183, 184, 300
 - `imq.cluster.masterbroker` 特性 180, 183, 184, 300
 - `imq.cluster.port` 特性 180, 300
 - `imq.cluster.property_name` 特性 282
 - `imq.cluster.transport` 特性 180, 182, 183, 301
 - `imq.cluster.url` 特性 105, 180, 181, 182, 183, 184, 300
 - `imq.destination.DMQ.truncateBody` 特性 79, 105, 282, 287
 - `imq.destination.logDeadMsgs` 特性 90, 105, 282, 297
 - `imq.hostname` 特性 75, 283, 285
 - `imq.httpjms.http.connectionTimeout` 特性 345
 - `imq.httpjms.http.property_name` 特性 283
 - `imq.httpjms.http.pullPeriod` 特性 345
 - `imq.httpjms.http.servletHost` 特性 345
 - `imq.httpjms.http.servletPort` 特性 345

- imq.httpsjms.https.connectionTimeout 特性 356
- imq.httpsjms.https.*property_name* 特性 283
- imq.httpsjms.https.pullPeriod 特性 356
- imq.httpsjms.https.servletHost 特性 355
- imq.httpsjms.https.servletPort 特性 356
- imq.imqcmd.password 特性 283, 294
- imq.keystore.file.dirpath 特性 149, 297
- imq.keystore.file.name 特性 149, 297
- imq.keystore.password 特性 149, 156, 297
- imq.keystore.*property_name* 特性 87
- imq.keystore.*property_name* 特性 283, 294
- imq.log.console.output 特性 90, 283, 297
- imq.log.console.stream 特性 90, 283, 297
- imq.log.file.dirpath 特性 89, 283, 298
- imq.log.file.filename 特性 89, 298
- imq.log.file.name 特性 283
- imq.log.file.output 特性 89, 283, 298
- imq.log.file.rolloverbytes 特性 89, 106, 283, 298
- imq.log.file.rolloversecs 特性 90, 106, 283, 298
- imq.log.level 特性 89, 105, 283, 298
- imq.log.syslog.facility 特性 90, 283, 298
- imq.log.syslog.identity 特性 90, 283, 298
- imq.log.syslog.logconsole 特性 90, 283, 298
- imq.log.syslog.logpid 特性 90, 283, 298
- imq.log.syslog.output 特性 90, 283, 299
- imq.log.timezone 特性 90, 283, 299
- imq.message.expiration.interval 特性 79, 283, 287
- imq.message.max_size 特性 79, 106, 283, 287
- imq.metrics.enabled 特性 89, 283, 299
- imq.metrics.interval property 90, 283, 299
- imq.metrics.interval 特性 89, 283, 299
- imq.metrics.topic.interval property 90, 283, 299
- imq.metrics.topic.persist property 90, 283, 299
- imq.metrics.topic.timetolive property 90, 283, 299
- imq.passfile.dirpath 特性 86, 283, 294
- imq.passfile.enabled 特性 86, 283, 294
- imq.passfile.name 特性 86, 284, 294
- imq.persist.file.destination.message.filepool.limit 特性 81, 82, 284, 290
- imq.persist.file.message.cleanup 特性 81, 82, 284, 290
- imq.persist.file.message.filepool.cleanratio 特性 82, 284, 290
- imq.persist.file.message.max_record_size 特性 82, 284, 290
- imq.persist.file.message.vrfile.max_record_size 特性 81
- imq.persist.file.sync 特性 94
- imq.persist.file.sync.enabled 特性 82, 284, 290
Sun Cluster 需求 290
- imq.persist.jdbc.brokerid 特性 83, 97, 291
- imq.persist.jdbc.closedburl 特性 83, 97, 291
- imq.persist.jdbc.createdburl 特性 83, 97, 291
- imq.persist.jdbc.driver 特性 83, 97, 291
- imq.persist.jdbc.needpassword 特性 83, 97, 292
- imq.persist.jdbc.opendburl 特性 83, 97, 291
- imq.persist.jdbc.password 特性 83, 97, 156, 292
- imq.persist.jdbc.*property_name* 特性 284
- imq.persist.jdbc.table.IMQCCREC35 特性 83, 97, 292
- imq.persist.jdbc.table.IMQDEST35 特性 83, 97, 292
- imq.persist.jdbc.table.IMQINT35 特性 83, 97, 292
- imq.persist.jdbc.table.IMQLIST35 特性 83, 97, 293
- imq.persist.jdbc.table.IMQMSG35 特性 83, 97, 292
- imq.persist.jdbc.table.IMQPROPS35 特性 83, 97, 292
- imq.persist.jdbc.table.IMQSV35 特性 83, 97, 292
- imq.persist.jdbc.table.IMQTACK35 特性 83, 97, 293
- imq.persist.jdbc.table.IMQTXN35 特性 83, 97, 293
- imq.persist.jdbc.user 特性 83, 97, 291
- imq.persist.store 特性 82, 97, 284, 289, 291
- imq.ping.interval 特性 284, 285
- imq.portmapper.backlog 特性 76, 284, 285
- imq.portmapper.hostname 特性 76, 284, 285
- imq.portmapper.port 特性 76, 106, 284, 285
- imq.protocol protocol_type inbufsz 220
- imq.protocol protocol_type nodelay 220
- imq.protocol protocol_type outbufsz 220

`imq.resource_state.count` 特性 80, 284, 287
`imq.resource_state.threshold` 特性 79, 284, 287
`imq.service.activelist` 特性 75, 284, 285
`imq.service_name.accesscontrol.enabled` 特性 86, 284, 294
`imq.service_name.accesscontrol.file.filename` 特性 86, 284, 294
`imq.service_name.authentication.type` 特性 86, 284, 294
`imq.service_name.max_threads` 特性 76, 284, 286
`imq.service_name.min_threads` 特性 76, 284, 286
`imq.service_name.protocol_type.hostname` 特性 76, 180, 284, 285, 300
`imq.service_name.protocol_type.port` 特性 76, 284, 285
`imq.service_name.threadpool_model` 特性 76, 284, 286
`imq.shared.connectionMonitor_limit` 特性 76, 284, 286
`imq.system.max_count` 特性 79, 106, 284, 287
`imq.system.max_size` 特性 79, 106, 284, 287
`imq.transaction.autorollback` 特性 78, 80, 117, 284, 287
`imq.user_repository.ldap.base` 特性 139, 294
`imq.user_repository.ldap.gidattr` 特性 139, 295
`imq.user_repository.ldap.grpbase` 特性 139, 295
`imq.user_repository.ldap.grpfilter` 特性 139, 295
`imq.user_repository.ldap.grpsearch` 特性 139, 295
`imq.user_repository.ldap.memattr` 特性 139, 295
`imq.user_repository.ldap.password` 特性 139, 156, 295
`imq.user_repository.ldap.principal` 特性 139, 295
`imq.user_repository.ldap.property_name` 特性 284, 295
`imq.user_repository.ldap.server` 特性 139, 296
`imq.user_repository.ldap.ssl.enabled` 特性 139, 296
`imq.user_repository.ldap.timeout` 特性 139, 296
`imq.user_repository.ldap.uidattr` 特性 139, 296
`imq.user_repository.ldap.usrfilter` 特性 139, 296
`imqAckOnProduce` 屬性 313
`imqAckTimeout` 屬性 166, 313

`imqAddressList` 屬性 309
`imqAddressListBehavior` 屬性 309
`imqAddressListIterations` 屬性 309
`imqbrokerd` 指令 64
 配置檔案 (Solaris、Linux) 66, 69
 參照 257
 密碼檔案中 155
 將代理程式新增至叢集 183
 從叢集中移除代理程式 183
 清除資料儲存 94, 126
 移除代理程式 69
 設定記錄特性 191
 連線代理程式 181
 備份配置變更記錄 184
 復原配置變更記錄 185
 傳送引數至 93
 語法 257
 選項 258
 關於 37
`imqbrokerd.conf` 檔案 66, 69
`imqcmd` 指令
 作業事件管理 116
 取決於主代理程式 185
 長期訂閱子指令 115
 度量監視 194
 參照 261
 密碼檔案中 155
 實體目標子指令 (表格) 120
 實體目標管理 119
 與代理程式的安全連線 152, 270
 語法 261
 選項 269
 關於 37
`imqConfiguredClientID` 屬性 166, 312
`imqConnectionFlowCount` 屬性 166, 313
`imqConnectionFlowLimit` 屬性 167, 313, 314
`imqConnectionFlowLimitEnabled` 屬性 167, 313
`imqConsumerFlowLimit` 屬性 167, 314
`imqConsumerFlowThreshold` 屬性 167, 314
`imqdbmgr` 指令
 子指令 274

參照 273
 密碼檔案中 155
 語法 273
 選項 274
 關於 37

imqDefaultPassword 屬性 166, 312
 imqDefaultUsername 屬性 166, 312
 imqDestinationDescription 屬性 308
 imqDestinationName 屬性 308
 imqDisableSetClientID 屬性 166, 312
 imqFlowControlLimit 屬性 167, 314
 imqJMSDeliveryMode 屬性 168, 312
 imqJMSExpiration 屬性 168, 312
 imqJMSPriority 屬性 168, 313

imqkeytool 指令
 使用 148, 353
 指令語法 148, 353
 參照 279
 關於 37

imqLoadMaxToServerSession 屬性 167, 315

imqobjmgr 指令
 子指令 271
 參照 271
 語法 271
 選項 272
 關於 37

imqOverrideJMSDeliveryMode 屬性 168, 312
 imqOverrideJMSExpiration 屬性 168, 312
 imqOverrideJMSHeadersToTemporaryDestinations
 屬性 168, 313

imqOverrideJMSPriority 屬性 168, 312

imqQueueBrowserMax MessagesPerRetrieve 屬
 性 167, 314

imqQueueBrowserRetrieveTimeout 屬性 167, 315
 imqReconnectAttempts 屬性 309
 imqReconnectEnabled 屬性 309
 imqReconnectInterval 屬性 309
 imqSetJMSXAAppID 屬性 167, 315
 imqSetJMSXConsumerTXID 屬性 167, 315
 imqSetJMSXProducerTXID 屬性 167, 315

imqSetJMSXRcvTimestamp 屬性 168, 315
 imqSetJMSXUserID 屬性 167, 315
 imqSSLIsHostTrusted 屬性 309

imqsvcadm 指令
 子指令 277
 參照 277
 語法 277
 選項 278
 關於 38

imqusermgr 指令
 子指令 276
 用於 133
 使用者名稱 136
 參照 275
 密碼 136
 語法 275
 選項 276
 關於 37

install.properties 檔案 91

J

J2EE Connector Architecture (JCA) 317, 320

Java 執行階段 257
 適用於 Windows 服務 67

Java 虛擬機器, 請參閱 JVM

java.naming.factory.initial 屬性 160, 161
 java.naming.provider.url 屬性 160, 162
 java.naming.security.authentication 屬性 161
 java.naming.security.credentials 屬性 161
 java.naming.security.principal 屬性 160
 javahome 選項 67, 257

JCA (J2EE Connector Architecture) 317, 320

JDBC 支援
 配置 93
 設定 94
 關於 82
 驅動程式 83, 93, 97, 291

JDK

L 部分

指定路徑 259, 269, 272, 278

JMS 規格 28

jms 連線服務 73, 109

JMSDeliveryMode 訊息標頭欄位 168

JMSExpiration 訊息標頭欄位 168

JMSPriority 訊息標頭欄位 168

JNDI

位置 (提供者 URL) 160, 161

物件儲存 37, 160

物件儲存屬性 160, 170

初始環境 160, 161

查找 52, 169

查找名稱 169, 173

jrehome 選項 67

JVM

度量, 請參閱 JVM 度量

效能影響 214

效能調校 219

JVM 度量

使用 imqcmd metrics 196

使用代理程式記錄檔 193

使用基於訊息的監視 199

度量數目 324

L

LDAP 伺服器

使用者儲存庫存取 138

物件儲存屬性 160

做為使用者儲存庫 137

認證防故障備用 138

M

ManagedConnectionFactory JavaBean 319

MDB, 請參閱訊息驅動 Bean

messageSelector 啟動規格屬性 321

N

NORMAL 服務類型 73

nsswitch.conf 檔案 (Linux) 182

O

Oracle 94, 99

P

password 受管理連線工廠屬性 320

password 資源介面屬性 318

PointBase 94

properties

實體目標, 請參閱實體目標, 特性

R

reconnectAttempts 受管理連線工廠屬性 320

reconnectAttempts 資源介面屬性 319

reconnectEnabled 受管理連線工廠屬性 320

reconnectEnabled 資源介面屬性 319

reconnectInterval 受管理連線工廠屬性 320

reconnectInterval 資源介面屬性 319

reset messages 選項 126

ResourceAdapter JavaBean 318

RESTART 特性 66

S

sendUndeliverableMsgsToDMQ 啟動規格屬性 322

Simple Network Time Protocol 64

SNTP 64

SSL

加密, 與 146
 啟用 150
 連線服務, 請參閱基於 SSL 的連線服務
 透過 TCP/IP 147
 關於 85

ssladmin 連線服務
 設定 147
 關於 74, 109

ssljms 連線服務
 設定 147
 關於 73, 109

subscriptionDurability 啟動規格屬性 321, 322

subscriptionName 啟動規格屬性 322

Sun Cluster
 同步化屬性與 82
 配置 290

syslog 88, 192

T

TCP 73, 109

TimeToLive 功能
 時鐘同步化與 64

TLS 73, 109

U

ulimit 指令 64

userName 受管理連線工廠屬性 320

userName 資源介面屬性 318

W

W32Time 服務 64

Windows 服務, 請參閱服務 (Windows)

X

xntpd 常駐程式 64

三畫

工具, 管理, 請參閱管理工具

四畫

內建持續性 81

分散式作業事件
 XA 資源管理員 116

五畫

主代理程式

指定 180, 181

配置變更記錄 184

無法使用 185

主題

自動建立的 282, 288

新增受管理物件 174

代理程式

HTTP 支援 343

httpjms 連線服務特性 345

HTTPS 支援 352

httpsjms 連線服務特性 355

互連, 請參閱代理程式叢集

元件和功能 (表格) 73

列出連線服務 110

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

自動建立實體目標特性 288

自動重新啟動 66

更新特性 105

使用 SSL 啟動 150

服務 (圖) 72

- 度量,請參閱代理程式度量
- 持續性管理程式,請參閱持續性管理程式查詢 104
- 重新啓動 80, 107, 108, 264
- 限制運作方式 79, 218
- 時鐘同步化 64
- 特性 (參照) 281
- 記憶體管理 78, 121, 218
- 記錄,請參閱記錄程式
- 訊息流量控制,請參閱訊息流量控制
- 訊息容量 79, 106, 284, 287
- 訊息路由,請參閱訊息路由器
- 配置檔案,請參閱配置檔案
- 停用的訊息佇列 129
- 從故障回復 80
- 啓動時需要的許可權 65
- 移除 69
- 連線 181
- 連線服務,請參閱連線服務
- 當作 Windows 服務來執行 66
- 實例名稱 259
- 實例配置特性 92
- 監視,請參閱代理程式監視服務
- 管理 101
- 暫停 106, 107, 264
- 叢集,請參閱代理程式叢集
- 關閉 107
- 繼續 106, 107, 264
- 顯示特性 104
- 代理程式 Windows 服務的啓動參數 67
- 代理程式回應
 - 生產 313
 - 用戶端的等待期間 166, 313
- 代理程式度量
 - 使用 imqcmd 108, 197, 198
 - 使用代理程式記錄檔 193
 - 使用基於訊息的監視 199
 - 度量訊息 89
 - 度量數目 (表格) 324
 - 記錄程式特性 89, 193, 299
 - 報告間隔時間,記錄程式 259
- 代理程式故障和安全的連線 364
- 代理程式監視服務
 - 特性 297
 - 關於 87
- 代理程式叢集
 - 安全代理程式互通連線 182
 - 使用的原因 217
 - 架構 217
 - 要指定的選項 258
 - 效能影響 217
 - 配置特性 180, 300
 - 配置檔案 180, 181, 182, 300
 - 配置變更記錄 184
 - 連線代理程式 181
 - 新增代理程式 182
 - 實體目標的複製 121
 - 暫停實體目標 125
- 加密
 - 基於 SSL 的服務,與 146
 - 鍵值工具,與 86
 - 關於 85
- 可靠的傳送 166
 - 效能平衡 208
- 外掛持續性
 - 效能調校 223
 - 設定 94
 - 關於 82
- 正在壓縮
 - 基於檔案的資料儲存 81
- 生產環境
 - 設定 34
 - 管理工作 34
 - 維護 35
- 用戶端
 - 時鐘同步化 64
 - 啓動 68
- 用戶端執行階段
 - 訊息流量調校 225
 - 配置 218
- 用戶端應用程式
 - 影響效能的因素 208
 - 範例 28, 334, 335, 336
- 用戶端識別碼 (ClientID) 165

- 銷毀長期訂閱 115
- 用法說明 257
- 目標受管理物件
 - 屬性 168
- 目標度量
 - 使用 imqcmd metrics 195, 197, 266
 - 使用 imqcmd query 199
 - 使用基於訊息的監視 199
 - 度量數目 328
- 目錄變數
 - IMQ_HOME 25
 - IMQ_JAVAHOME 26
 - IMQ_VARHOME 25
- 目錄變數 IMQ_HOME 25
- 目錄變數 IMQ_JAVAHOME 26
- 目錄變數 IMQ_VARHOME 25

六畫

- 同步化
 - 時鐘 64
 - 磁碟記憶體 82, 94
- 存取控制檔
 - 用於 141
 - 存取規則 143
 - 位置 334, 335, 336
 - 版本 141
 - 格式 142
- 存取規則 143
- 安全性
 - 加密, 請參閱加密
 - 物件儲存, 用於 160
 - 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 73
 - 特性 293
 - 關於 83
- 安全套接層標準, 請參閱 SSL

- 自身簽名的憑證 148, 353
- 自動建立實體目標
 - 存取控制 146
 - 特性 (表格) 288
 - 配置 85
 - 停用 35
- 自動重新連線功能
 - 屬性: 164

七畫

- 佇列
 - 自動建立的 282, 288
 - 新增受管理物件 174
- 佇列負載平衡傳送
 - 屬性 122, 304
- 伺服器故障和安全的連線 364
- 作業系統
 - 效能影響 214
 - 調校 Solaris 效能 219
- 作業事件
 - 回轉 116, 268
 - 效能影響 210
 - 資訊 269
 - 管理 116
 - 確定 117, 269
 - 確認與 78
- 刪除
 - 代理程式實例 69
- 刪除目標 126
- 更新
 - 代理程式 105
 - 連線服務 111, 114, 267
- 更新 dst 子指令
 - 限制 124
- 系統時鐘同步化 64
- 防火牆 342

八畫

- 使用者名稱 166, 312
 - 格式 136
 - 預設 133
- 使用者群組 135
 - 刪除指定 135
 - 預先定義的 135
 - 預設 85
 - 關於 84
- 使用者儲存庫
 - LDAP 137
 - LDAP 伺服器 138
 - 文字檔案 132
 - 平台相依性 133, 275
 - 位置 334, 335, 336
 - 使用者狀態 135
 - 使用者群組 135
 - 初始項目 133
 - 特性 86
 - 管理 136
 - 寫入 136
 - 關於 84
- 使用率 127
- 協定, 請參閱傳輸協定
- 協定類型
 - HTTP 74, 109
 - TCP 73, 109
 - TLS 73, 109
- 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
 - 列出 176
 - 佇列, 請參閱佇列
 - 刪除 175
 - 更新 177
 - 物件儲存, 請參閱物件儲存
 - 查找名稱 272
 - 查詢 177
 - 需要的資訊 169
 - 屬性 (參照) 307
- 所有指令的語法 256
- 服務 (Windows)

- Java 執行階段適用於 67
- 重新配置 67
- 執行代理程式 66
- 啟動參數適用於 67
- 移除代理程式 68
- 疑難排解啟動問題 68

- 服務類型
 - ADMIN 73
 - NORMAL 73
- 版本 257
- 物件儲存
 - LDAP 伺服器 160
 - LDAP 伺服器屬性 160
 - 位置 334, 335, 336
 - 檔案系統儲存 161
 - 檔案系統儲存屬性 161
 - 關於 160
- 物件儲存的位置 160, 161
- 長期訂閱
 - ID 269
 - 列出 115, 268
 - 效能影響 211
 - 清除訊息 268
 - 管理 115
 - 銷毀 115, 268

九畫

- 度量
 - 主題目標 88, 199
 - 訊息, 請參閱度量訊息
 - 資料, 請參閱度量資料
 - 關於 87
- 度量訊息
 - 內容 89
 - 關於 88, 199
 - 類型 88, 199
- 度量資料
 - 代理程式, 請參閱代理程式度量
 - 使用 `imqcmd metrics` 196
 - 使用代理程式記錄檔 193

- 使用基於訊息監視的 API 199
 - 連線服務, 請參閱連線服務度量
 - 實體目標, 請參閱實體目標度量
 - 度量監視工具
 - 比較的 188
 - 訊息佇列公用程式 (imqcmd) 194
 - 訊息佇列記錄檔 193
 - 基於訊息監視的 API 199
 - 持續性
 - JDBC, 請參閱 JDBC 持續性
 - 內建 81
 - 外掛, 請參閱外掛持續性
 - 安全性 98
 - 持續性管理程式, 請參閱持續性管理程式
 - 資料儲存請參閱資料儲存
 - 選項 (圖) 80
 - 持續性管理程式
 - 外掛持續性, 與 93
 - 作為代理程式元件 73
 - 特性 290
 - 資料儲存, 請參閱資料儲存
 - 關於 80
 - 指令行公用程式
 - imqbrokerd, 請參閱, imqbrokerd 指令
 - imqcmd, 請參閱, imqcmd 指令
 - imqdbmgr 請參閱, imqdbmgr 指令
 - imqkeytool, 請參閱, imqkeytool 指令
 - imqobjmgr, 請參閱, imqobjmgr 指令
 - imqsvcadmin, 請參閱, imqsvcadmin 指令
 - imqusermgr, 請參閱, imqusermgr 指令
 - 基本語法 256
 - 說明 257
 - 選項共用於 257
 - 關於 36
 - 顯示版本 257
 - 指令行語法 256
 - 指令選項 257
 - 配置覆寫 68
 - 指令檔 170
 - 指導 39
 - 查詢
 - 代理程式 104
 - 連線服務 110, 114, 267
 - 流量控制, 請參閱訊息流量控制
 - 負載平衡的佇列傳送
 - 效能調校 224
 - 屬性 289
 - 重新啟動代理程式 107, 108, 264
 - 重新連線, 自動請參閱自動重新連線
 - 重新傳送旗標 78
 - 限制運作方式
 - 代理程式 79
 - 實體目標 78, 121, 304
- ## 十畫
- 效能
 - 可靠性平衡 208
 - 指示器 204
 - 效能評定 205
 - 基本式樣 205
 - 瓶頸 208
 - 最佳化, 請參閱效能調校
 - 測量 204
 - 疑難排解 227
 - 監視, 請參閱效能監視
 - 影響因素, 請參閱效能因素
 - 調校, 請參閱效能調校
 - 關於 204
 - 效能因素
 - JVM 214
 - 代理程式限制運作方式 218
 - 作業系統 214
 - 作業事件 210
 - 長期訂閱 211
 - 訊息內文類型 213
 - 訊息伺服器架構 217
 - 訊息流量控制 218
 - 訊息容量 212
 - 連線 215
 - 硬體 214
 - 傳送模式 209

十畫 部分

- 傳輸協定 215
- 資料儲存 218
- 確認模式 211
- 選擇器 212
- 檔案同步化 290
- 效能評定, 效能 205
- 效能監視
 - 工具, 請參閱度量監視工具 187
 - 度量資料, 請參閱度量資料
- 效能調校
 - 代理程式調整 223
 - 用戶端執行階段 225
 - 系統調整 219
 - 程序概述 204
- 時間同步化服務 64
- 時鐘同步化 64
- 特性
 - httpjms 連線服務 345
 - httpsjms 連線服務 355
 - 代理程式實例配置 92
 - 代理程式監視服務 297
 - 安全性 293
 - 自動建立 288
 - 持續性 290
 - 記憶體管理 121, 287
 - 記錄程式 297
 - 訊息路由器 287
 - 密鑰儲存 297
 - 連線服務 285
 - 與 JDBC 相關 96, 291
 - 語法 92
 - 叢集配置 300
- 記憶體管理
 - 用於代理程式 78
 - 使用實體目標特性 121
 - 效能調校 223
 - 臨界值 79
- 記錄, 請參閱記錄程式
- 記錄程式
 - 自動重建條件 192
 - 作為代理程式元件 73
 - 度量資訊 89, 299
 - 重新導向記錄訊息 192
 - 級別 89, 190, 259, 298
 - 訊息格式 190
 - 設定特性 191
 - 種類 190
 - 寫入至主控台 90, 261, 297
 - 輸出通道 88, 189, 191
 - 關於 88
 - 變更配置 191
- 記錄檔
 - 自動重建條件 89, 298
 - 預設位置 334, 335, 336
- 訊息
 - 內文類型與效能 213
 - 分段程序 81
 - 代理程式限制 79, 106, 284, 287
 - 可靠傳送 166
 - 目標限制 303
 - 回收已過期的 79, 287
 - 延時 204
 - 度量 88
 - 度量訊息, 請參閱度量訊息
 - 持續性 78, 80
 - 流量效能 204
 - 流量控制, 請參閱訊息流量控制
 - 重新傳送 78
 - 容量, 與效能 212
 - 從實體目標清除 126, 266
 - 路由與傳送 76
 - 實體目標限制 121
 - 暫停流量 125
 - 確認 77
- 訊息分段程序 81
- 訊息伺服器架構 217
- 訊息服務效能 214
- 訊息流量控制
 - 代理程式 78, 121
 - 計數 225
 - 限制 225
 - 效能影響 218
 - 效能調校 225
 - 屬性 166

- 訊息路由器
 - 作為代理程式元件 73
 - 特性 287
 - 關於 76
- 訊息標頭覆寫 168
- 訊息驅動 Bean
 - 資源介面配置 317, 320
- 迴路位址 182
- 配置檔案 90
 - 代理程式 (圖) 92
 - 安裝 91
 - 位置 334, 335, 336
 - 預設 91
 - 實例 91, 180, 334, 335, 336
 - 範本 334, 335, 336
 - 範本位置 334, 335, 336
 - 編輯 93
 - 叢集 180, 181, 182, 300
- 配置變更記錄 184
 - 備份 184
 - 復原 184

十一畫

- 停用的訊息
 - 另請參閱停用的訊息佇列記錄 90
- 停用的訊息佇列
 - maxNumMsgs 值 129
 - maxTotalMsgBytes 值 130
 - 限制運作方式 129
 - 記錄 90
 - 訊息截斷 79
 - 配置 128
- 停用訊息佇列
 - 記錄 130
- 基於 SSL 的連線服務
 - 啓動 150
 - 設定 146, 147
- 基於檔案的持續性 81

- 另請參閱基於持續性管理程式檔案的持續性 81
- 執行緒池管理程式
 - 共用執行緒 75
 - 專屬執行緒 75
 - 關於 75
- 密碼
 - JDBC 156
 - LDAP 156
 - SSL 密鑰儲存 149, 156, 260
 - 命名慣例 136
 - 密碼檔案, 請參閱密碼檔案預設 166, 312
 - 管理員 137
 - 編碼 294
- 密碼檔案
 - 代理程式配置特性 86, 294
 - 位置 156, 334, 335, 336
 - 使用 155
 - 指令行選項 260
- 密碼檔案, 請參閱密碼檔案
- 密鑰儲存
 - file 297
 - 特性 297
 - 檔案 149, 353
- 控制訊息 76
- 授權
 - 另請參閱存取控制檔案使用者群組 85
 - 啓動選項 259
 - 管理 140
 - 關於 84
- 啓動
 - 用戶端 68
 - 基於 SSL 的連線服務 150
- 清除, 實體目標的訊息 125
- 瓶頸, 效能 208
- 產生者
 - 目標限制 288, 304
 - 實體目標限制 121
- 移除
 - 代理程式 69
 - 實體目標 126

許可權

- Message Queue 作業 84
- 內嵌式資料庫 96
- 存取控制特性檔案 84, 141
- 使用者儲存庫 133, 275
- 密碼檔案 156
- 密鑰儲存 353
- 資料儲存 82
- 運算 143
- 管理服務 85

通道 Servlet 連線 365

連接埠對映器

- 連接埠指定 260
- 關於 74

連線

- 由檔案描述元限制 64
- 列出 113, 268
- 自動重新連線, 請參閱自動重新連線伺服器或代理程式發生故障 364
- 防故障備用, 請參閱自動重新連線查詢 114, 268
- 效能影響 215

連線工廠受管理物件

- JMS 特性支援屬性 167, 315
- 可靠性與流量控制屬性 166
- 用戶端識別屬性 165
- 佇列瀏覽器運作方式屬性 167, 314
- 連線處理的屬性 163
- 新增 173
- 應用程式伺服器支援屬性 167, 315
- 覆寫訊息標頭欄位 168
- 屬性 162

連線代理程式 181

連線服務

- admin 74, 109
- HTTP, 請參閱 HTTP 連線
- httpjms 74, 109
- HTTPS, 請參閱 HTTPS 連線
- httpsjms 74, 109
- jms 73, 109
- ssladmin, 請參閱 ssladmin 連線服務
- ssljms, 請參閱 ssljms 連線服務
- 在啟動時啟動 285

存取控制 86, 293

- 更新 111, 114, 267
- 服務類型 73
- 度量資料, 請參閱連線服務度量指令影響 267
- 查詢 110, 114, 267
- 特性 111, 285
- 基於 SSL 的 149
- 執行緒分配 111
- 執行緒池管理程式 75
- 連接埠對映器, 請參閱連接埠對映器連線類型 73
- 暫停 113, 267
- 叢集 147, 182
- 關於 73
- 繼續 113, 267
- 顯示特性 110

連線服務度量

- 使用 imqcmd metrics 111, 197
- 使用 imqcmd query 198
- 度量數目 326

十二畫

- 硬體, 效能影響 214
- 開發環境管理工作 34

十三畫

傳送模式

- 效能影響 209

傳輸協定

- 協定類型, 請參閱協定類型
- 相關速度 216
- 效能影響 215
- 效能調校 220

資料儲存

- JDBC 可存取的 82
- 內容 93

- 文字檔案 81
- 正在壓縮 81
- 位置 334, 335, 336
- 重設 260
- 效能影響 218
- 配置 93
- 磁碟同步化 94
- 關於 80
- 資源介面 317
 - 重新連線 318, 319, 320
- 路由, 請參閱訊息路由器

十四畫

實例目錄

- 和基於檔案的資料儲存 94
- 移除 69
- 與實例配置檔案 138

實例配置檔案, 請參閱配置檔案

實體目標

- 用於傳送的批次訊息 121, 289, 305
- 列出 122, 265
- 回收磁碟空間 127
- 自動建立的 146
- 更新特性 124
- 更新屬性 266
- 使用停用的訊息佇列 129
- 取得資訊 123, 266
- 度量, 請參閱實體目標度量
- 建立 121
- 限制運作方式 78, 121, 304
- 特性 303
- 特性值 123
- 停用的訊息佇列 128
- 清除訊息 125, 126, 266
- 資訊 123
- 磁碟使用 127
- 管理 119
- 暫存 122
- 暫停 125, 266

- 銷毀 126, 265
- 壓縮 127
- 壓縮基於檔案的資料儲存 128, 265
- 叢集中限制的範圍 121, 288, 305
- 類型 122, 265
- 繼續 125, 266
- 顯示特性值 123
- 實體目標屬性 303
- 截斷停用的訊息佇列 79
- 疑難排解 227
 - Windows 服務啓動 68
- 監視, 請參閱效能監視
- 磁碟空間
 - 回收 127
 - 實體目標的使用 127
- 管理工作
 - 生產環境 34
 - 開發環境 34
- 管理工具 36
 - 命令行公用程式 36
 - 管理主控台 38
- 管理主控台
 - 快速入門 39
- 管理員密碼 137
- 認證
 - 管理 132
 - 關於 84
- 說明 (指令行) 257

十五畫

- 寫入作業 (基於檔案的儲存) 94
- 暫存實體目標 122
- 暫停
 - 代理程式 106, 107, 264
 - 連線服務 113, 267
 - 實體目標 125, 266
- 確認
 - 用戶端 77
 - 作業事件與 78

十六畫 部分

傳送, 屬於 77
稽核記錄 157
範例應用程式 28, 334, 335, 336
銷毀實體目標 126

十六畫

憑證 148, 353

選擇器

效能影響 212

關於 212

十七畫

優先順序 (配置特性的) 91

壓縮

實體目標 127

應用程式, 請參閱用戶端應用程式

檔案同步化

imq.persist.file.sync.enabled 選項 290

與 Sun Cluster 290

檔案描述元限制 64

連線限制和 64

環境變數, 請參閱目錄變數

臨界值

記憶體 79

鍵值工具 86

鍵對

重新產生 149

產生 149

十八畫

叢集, 請參閱代理程式叢集

叢集的目錄查找 (Linux) 182

叢集配置特性 180, 300

叢集配置檔案 180, 181, 182, 300

叢集連線服務 147, 182

主機名稱或 IP 位址 300

主機名稱或 IP 位址用於 180

連接埠號 300

連接埠號用於 180

網路傳輸用於 180, 181, 301

覆寫

在指令行中 68

為訊息標頭 168

十九畫

關閉代理程式 107, 264

當作 Windows 服務 68

二十畫

繼續

代理程式 106, 107, 264

連線服務 113, 267

實體目標 125

二十三畫

顯示產品版本 257