



Sun Java™ System

Message Queue 3

管理指南

2005Q4

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

☎ 件號碼 : 819-3562

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

Sun Microsystems, Inc. 對於本文件所述技術擁有智慧財產權。這些智慧財產權包含 <http://www.sun.com/patents> 上所列的一項或多項美國專利，以及在美國與其他國家 / 地區擁有的一項或多項其他專利或申請中專利，但並不以此為限。

美國政府權利 - 商業軟體。政府使用者均應遵守 Sun Microsystems, Inc. 的標準授權合約和 FAR 及其增補文件中的適用條款。應依照授權條款使用。本發行軟體包含由協力廠商所開發的材料。

Sun、Sun Microsystems、Sun 標誌、Java、Solaris、Sun[tm] ONE、JDK、Java Naming and Directory Interface、JavaMail、JavaHelp 以及 Javadoc 是 Sun Microsystems, Inc. 在美國及其他國家 / 地區的商標或註冊商標。

所有 SPARC 商標都是 SPARC International, Inc. 在美國及其他國家 / 地區的商標或註冊商標，經授權後使用。凡具有 SPARC 商標的產品都是採用 Sun Microsystems, Inc. 所開發的架構。

UNIX 是在美國及其他國家 / 地區的註冊商標，已獲得 X/Open Company, Ltd. 專屬授權。

本產品受美國出口控制法的管轄，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	11
表	13
程序	17
前言	19
本書適用對象	20
閱讀本書之前	20
本書架構	20
本書使用慣例	22
文字慣例	22
目錄變數慣例	22
相關文件	24
Message Queue 文件集	24
Java Message Service 規格	24
線上說明	25
JavaDoc	25
範例用戶端應用程式	25
相關協力廠商網站	26
Sun 歡迎您提出寶貴意見	26
第 I 部分 Message Queue 管理簡介	27
第 1 章 管理工作工具	29
管理工作	30
開發環境中的管理	30
生產環境中的管理	30
管理工具	32

指令行公用程式	32
管理主控台	33

第 2 章 快速入門指導	35
啟動管理主控台	37
管理主控台線上說明	38
使用代理程式	39
啟動代理程式	39
將代理程式新增至管理主控台	40
連線至代理程式	41
檢視連線服務	42
使用實體目標	44
建立實體目標	44
檢視實體目標特性	46
清除實體目標的訊息	48
刪除實體目標	48
使用物件存放區	49
新增物件存放區	49
連線至物件存放區	51
使用管理物件	52
新增連線工廠	52
新增目標	54
檢視管理物件特性	56
刪除管理物件	56
執行範例應用程式	57

第 II 部分 管理工作

61

第 3 章 啟動代理程式和用戶端	65
準備系統資源	66
同步化系統時鐘	66
設定檔案描述元限制	66
啟動代理程式	67
以互動方式啟動代理程式	67
自動啟動代理程式	68
移除代理程式	71
移除 Solaris 或 Linux 上的代理程式	71
移除 Windows 上的代理程式服務	71
啟動用戶端	72

第 4 章 管理代理程式	73
代理程式服務	74
連線服務	74
路由服務	76
永久性服務	78
安全性服務	81
監視服務	84
設定代理程式特性	86
配置檔案	86
配置永久資料存放區	89
配置檔案型存放區	89
配置 JDBC 型存放區	90
保護永久性資料	91
第 5 章 管理代理程式	93
必要條件	94
使用 imqcmd 公用程式	95
顯示說明	95
顯示產品版本	95
指定使用者名稱和密碼	95
指定代理程式名稱和連接埠	96
範例	96
顯示代理程式資訊	97
更新代理程式特性	98
暫停和繼續代理程式	99
暫停代理程式	99
繼續代理程式	99
關閉和重新啟動代理程式	100
顯示代理程式度量	100
管理連線服務	101
列出連線服務	102
顯示連線服務資訊	103
更新連線服務特性	103
顯示連線服務度量	104
暫停和繼續連線服務	105
取得有關連線的資訊	106
管理長期訂閱	107
管理作業事件	108
第 6 章 管理實體目標	111
使用指令公用程式	113
子指令	113
建立實體目標	114

列出實體目標	115
顯示實體目標資訊	116
更新實體目標特性	117
暫停和繼續實體目標	118
清除實體目標	119
銷毀實體目標	119
壓縮實體目標	120
配置使用停用訊息佇列	122
配置使用停用訊息佇列	122
配置和管理停用訊息佇列	122
啟用停用訊息記錄功能	123

第 7 章 管理安全性 125

認證使用者	126
使用平面檔案使用者儲存庫	126
將 LDAP 伺服器用於使用者儲存庫	132
授權使用者：存取控制特性檔案	134
建立存取控制特性檔案	135
存取規則的語法	135
如何計算權限	136
連線服務的存取控制	137
實體目標的存取控制	139
自動建立實體目標的存取控制	140
使用 SSL 型服務	140
TCP/IP 的安全連線服務	141
配置使用自身簽名的憑證	141
配置使用簽名的憑證	146
使用密碼檔案	149
安全考量	150
密碼檔案內容	150
建立稽核記錄	151

第 8 章 管理管理物件 153

物件存放區	154
LDAP 伺服器物件存放區	154
檔案系統物件存放區	155
管理物件屬性	156
連線工廠屬性	156
目標屬性	162
使用物件管理員公用程式	162
新增管理物件	163
刪除管理物件	165

列出管理物件	166
檢視管理物件資訊	166
修改管理物件屬性	167
使用指令檔	167
第 9 章 使用代理程式叢集	171
叢集配置特性	172
設定個別代理程式的叢集特性	172
使用叢集配置檔案	173
管理叢集	173
連線代理程式	173
將代理程式新增至叢集	174
從叢集中移除代理程式	175
主代理程式	176
管理配置變更記錄	176
當主代理程式無法使用時	177
第 10 章 監視訊息型機器	179
監視工具簡介	180
配置及使用代理程式記錄功能	181
預設記錄配置	181
記錄訊息格式	182
變更記錄程式配置	182
互動顯示度量	186
imqcmd metrics	186
使用顯示度量資料的 metrics 子指令	188
度量輸出：imqcmd metrics	188
imqcmd query	190
撰寫應用程式來監視代理程式	191
設定訊息型監視	191
安全性與存取注意事項	192
度量輸出：度量訊息	193
第 11 章 分析業務型訊息機器	195
關於效能	196
效能調校程序	196
效能類型	196
效能評定	197
基本使用式樣	198
影響效能的因素	199
影響效能的應用程式設計因素	200
影響效能的訊息服務因素	206

調校配置以改善效能	211
系統調校	211
代理程式調校	215
用戶端執行階段訊息流量調校	216

第 12 章 排解疑難問題	219
用戶端無法建立連線	220
連線流量過慢	224
用戶端無法建立訊息產生者	225
訊息產生延遲或過慢	226
儲存訊息	229
訊息伺服器的流量不穩定	233
訊息未送達用戶	234
停用的訊息佇列中包含訊息	238

第 III 部分 參照

第 13 章 指令行參照	247
指令行語法	248
代理程式公用程式	249
指令公用程式	252
代理程式管理	254
連線服務管理	255
連線管理	255
實體目標管理	256
長期訂閱管理	257
作業事件管理	258
一般指令公用程式選項	258
物件管理員公用程式	259
資料庫管理員公用程式	261
使用者管理員公用程式	262
服務管理員公用程式	263
金鑰工具公用程式	264

第 14 章 代理程式特性參照	265
連線特性	266
路由特性	267
永久性特性	271
檔案型永久性	272
JDBC 型永久性	273
安全性特性	276

檢視特性	280
叢集配置特性	283
按字母順序排列的代理程式特性清單	284
第 15 章 實體目標特性參照	289
第 16 章 管理物件屬性參照	293
連線工廠屬性	294
連線處理	294
用戶端標識	297
穩定性與流量控制	297
佇列瀏覽器和伺服器階段作業	299
設定標準訊息特性	299
訊息標頭覆寫	299
目標屬性	300
SOAP 端點屬性	300
第 17 章 JMS 資源介面特性參照	303
ResourceAdapter JavaBean	304
ManagedConnectionFactory JavaBean	305
ActivationSpec JavaBean	306
第 18 章 度量參照	309
JVM 度量	310
代理程式通用度量	310
連線服務度量	312
目標度量	314
第 IV 部分 附錄	317
附錄 A Message Queue 資料的平臺特定位置	319
Solaris	320
Linux	321
Windows	322
附錄 B Message Queue 介面的穩定性	323
附錄 C HTTP/HTTPS 支援	327
HTTP/HTTPS 支援架構	328
啟用 HTTP 支援	329

步驟 1：部署 HTTP 通道 Servlet	329
步驟 2：配置 httpjms 連線服務	333
步驟 3：配置 HTTP 連線	334
啓用 HTTPS 支援	336
步驟 1：爲 HTTPS 通道 Servlet 產生自身簽名的憑證	336
步驟 2：修改 HTTP 通道 Servlet.war 檔案的描述元檔案	337
步驟 3：部署 HTTPS 通道 Servlet	338
步驟 4：配置 httpsjms 連線服務	342
步驟 5：配置 HTTPS 連線	343
疑難排解	345
伺服器或代理程式發生故障	346
用戶端無法透過通道 Servlet 連線	346
附錄 D 管理指令與代理程式指令	347
語法	348
代理程式和叢集管理	348
代理程式配置特性 (-o 選項)	348
服務和連線管理	349
長期用戶管理	350
作業事件管理	350
目標管理	350
目標配置特性 (-o 選項)	350
標準資訊	351
目錄表	353
索引	355



圖 1-1	本機與遠端管理公用程式	33
圖 2-1	管理主控台視窗	37
圖 2-2	管理主控台說明視窗	38
圖 2-3	[新增代理程式] 對話方塊	40
圖 2-4	管理主控台視窗中顯示的代理程式	41
圖 2-5	[連線至代理程式] 對話方塊	42
圖 2-6	檢視連線服務	43
圖 2-7	[服務特性] 對話方塊	43
圖 2-8	[新增代理程式目標] 對話方塊	45
圖 2-9	[代理程式目標屬性] 對話方塊	47
圖 2-10	[長期訂閱] 畫面	48
圖 2-11	[新增物件存放區] 對話方塊	50
圖 2-12	管理主控台視窗中顯示的物件存放區	51
圖 2-13	[新增連線工廠物件] 對話方塊	53
圖 2-14	[新增目標物件] 對話方塊	55
圖 2-15	管理主控台視窗中顯示的目標物件	56
圖 4-1	永久性資料存放區	78
圖 4-2	安全性支援	82
圖 4-3	監視支援	84
圖 4-4	代理程式配置檔案	87
圖 11-1	透過 Message Queue 服務的訊息傳送	199
圖 11-2	傳送模式的效能影響	202
圖 11-3	訂閱類型的效能影響	204
圖 11-4	訊息大小的效能影響	205
圖 11-5	傳輸協定速度	208
圖 11-6	傳輸協定的效能影響	209
圖 11-7	在 1K (1,024 個位元組) 封包上變更 inbufsz 的效果	213

圖 11-8	在 1K (1,024 個位元組) 封包上變更 outbufsz 的效果	214
圖 12-1	QBrower 視窗	236
圖 12-2	QBrower 訊息的詳細資訊	237
圖 C-1	HTTP/HTTPS 支援架構	328

表 1	本手冊的內容	20
表 2	文字慣例	22
表 3	Message Queue 目錄變數	23
表 4	Message Queue 文件集	24
表 5	JavaDoc 位置	25
表 4-1	Message Queue 連線服務	74
表 4-2	度量主題目標	86
表 5-1	代理程式支援的連線服務	101
表 5-2	imqcmd 更新的連線服務特性	103
表 6-1	指令公用程式的實體目標子指令	113
表 6-2	實體目標磁碟使用度量	121
表 6-3	停用訊息佇列處理標準實體目標特性的方式	123
表 7-1	使用者儲存庫中的初始項目	127
表 7-2	imqusermgr 選項	129
表 7-3	存取規則的語法元素	136
表 7-4	實體目標存取控制規則的元素	139
表 7-5	自身簽名憑證所需之識別名稱資訊	142
表 7-6	使用密碼的指令	149
表 7-7	密碼檔案中的密碼	150
表 8-1	LDAP 物件存放區屬性	154
表 8-2	檔案系統物件存放區屬性	155
表 10-1	度量監視工具的優點與限制	180
表 10-2	記錄層級	182
表 10-3	imqbrokerd 記錄程式選項和對應的特性	183
表 10-4	imqcmd metrics 子指令語法	187
表 10-5	imqcmd metrics 子指令選項	187
表 10-6	imqcmd query 子指令語法	190

表 10-7	度量主題目標	191
表 11-1	高可靠性和高效能方案的比較	201
表 13-1	代理程式公用程式選項	249
表 13-2	代理程式管理的指令公用程式子指令	254
表 13-3	連線服務管理的指令公用程式子指令	255
表 13-4	連線服務管理的指令公用程式子指令	256
表 13-5	實體目標管理的指令公用程式子指令	256
表 13-6	長期訂閱管理的指令公用程式子指令	258
表 13-7	作業事件管理的指令公用程式子指令	258
表 13-8	一般指令公用程式選項	258
表 13-9	物件管理員子指令	259
表 13-10	物件管理員選項	260
表 13-11	資料庫管理員子指令	261
表 13-12	資料庫管理員選項	261
表 13-13	使用者管理員子指令	262
表 13-14	一般使用者管理員選項	263
表 13-15	服務管理員子指令	263
表 13-16	服務管理員選項	264
表 14-1	代理程式連線特性	266
表 14-2	代理程式路由特性	267
表 14-3	自動建立目標的代理程式特性	269
表 14-4	全域代理程式永久性特性	271
表 14-5	檔案型永久性的代理程式特性	272
表 14-6	JDBC 型永久性的代理程式特性	273
表 14-7	代理程式安全性特性	276
表 14-8	代理程式監視特性	280
表 14-9	叢集配置的代理程式特性	283
表 14-10	按字母順序排列的代理程式特性清單	284
表 15-1	實體目標特性	289
表 16-1	連線處理的連線工廠屬性	294
表 16-2	訊息伺服器定址機制	295
表 16-3	訊息伺服器位址範例	296
表 16-4	用戶端標識的連線工廠屬性	297
表 16-5	穩定性與流量控制的連線工廠屬性	297
表 16-6	佇列瀏覽器和伺服器階段作業的連線工廠屬性	299
表 16-7	標準訊息特性的連線工廠屬性	299
表 16-8	訊息標頭覆寫的連線工廠屬性	300

表 16-9	目標屬性	300
表 16-10	SOAP 端點屬性	301
表 17-1	資源介面特性	304
表 17-2	管理式連線工廠屬性	305
表 17-3	啟動規格特性	306
表 18-1	JVM 度量	310
表 18-2	代理程式通用度量	310
表 18-3	連線服務度量	312
表 18-4	目標度量	314
表 A-1	Solaris 平台上的 Message Queue 資料位置	320
表 A-2	Linux 平台上的 Message Queue 資料位置	321
表 A-3	Windows 平台上的 Message Queue 資料位置	322
表 B-1	介面穩定性分類機制	323
表 B-2	Message Queue 介面的穩定性	324
表 C-1	httpjms 連線服務特性	334
表 C-2	httpsjms 連線服務特性	342
表 D-1	代理程式配置特性 (-o 選項)	348
表 D-2	目標配置特性 (-o 選項)	351

將代理程式新增至管理主控台	40
連線至代理程式	41
檢視可用的連線服務	42
將實體目標新增至代理程式	44
檢視或修改實體目標的特性	46
清除實體目標的訊息	48
刪除實體目標	49
將物件存放區新增至管理主控台	50
連線至物件存放區	52
將連線工廠新增至物件存放區	52
將目標新增至物件存放區	54
檢視或修改管理物件的特性	56
刪除管理物件	57
執行範例應用程式	57
重新配置執行為 Windows 服務的代理程式	69
查閱記錄的服務錯誤事件	70
配置 JDBC 型資料存放區	90
建立實體目標	115
收回未使用的實體目標磁碟空間	121
編輯配置檔案以使用 LDAP 伺服器	132
設定管理使用者	134
設定 SSL 型連線服務	141
重新產生金鑰對	143
啓用代理程式中 SSL 型的服務	144
取得簽名的憑證	147
安裝簽名的憑證	147
配置 Java 用戶端執行階段	148

使用叢集配置檔案將新的代理程式新增至叢集	174
不使用叢集配置檔案將新的代理程式新增至叢集	175
使用指令行從叢集中移除代理程式	175
使用叢集配置檔案從叢集中移除代理程式	176
備份配置變更記錄	177
復原配置變更記錄	177
變更代理程式的記錄程式配置	182
使用記錄檔報告度量資訊	184
使用 <code>metrics</code> 子指令	188
設定訊息型監視	191
啓用 HTTP 支援	329
將 HTTP 通道 Servlet 部署為 .war 檔案	330
停用伺服器存取記錄	331
在應用程式伺服器環境中部署 HTTP 通道 Servlet	332
修改 Application Server 的伺服器策略檔案	333
啓動 <code>httpjms</code> 連線服務	333
啓用 HTTPS 支援	336
修改 HTTPS 通道 Servlet .war 檔案	338
將 HTTPS 通道 Servlet 部署為 .war 檔案	339
停用伺服器存取記錄	340
在應用程式伺服器環境中部署 HTTPS 通道 Servlet	340
修改 Application Server 的伺服器策略檔案	341
啓動 <code>httpsjms</code> 連線服務	342
配置 JSSE	343

「Sun Java™ System Message Queue 管理指南」描述 Sun Java System Message Queue 3 2005Q4 (Message Queue 3.6)，為您提供管理 Message Queue 訊息傳送系統所需要的資訊。

本前言包含下列各節：

- 第 20 頁的「本書適用對象」
- 第 20 頁的「閱讀本書之前」
- 第 20 頁的「本書架構」
- 第 22 頁的「本書使用慣例」
- 第 24 頁的「相關文件」
- 第 26 頁的「相關協力廠商網站」
- 第 26 頁的「Sun 歡迎您提出寶貴意見」

本書讀者對象

本手冊適用於需要執行 Message Queue 管理工作的管理員和應用程式開發者。Message Queue 管理員負責設定和管理 Message Queue 訊息傳送系統，尤其是處於系統核心地位的訊息伺服器。

閱讀本書之前

閱讀本手冊之前，應先閱讀「Message Queue 技術摘要」，以熟悉 Java 訊息規格之 Message Queue 實作、Message Queue 服務的元件，以及開發、部署和管理 Message Queue 應用程式的基本程序。

本書架構

表 1 簡述本手冊的內容。

表 1 本手冊的內容

部分 / 章	說明
第 I 部分「Message Queue 管理簡介」	
第 1 章「管理工作工具」	介紹 Message Queue 管理工作工具。
第 2 章「快速入門指導」	提供實作指導，讓您熟悉 Message Queue 管理主控台。
第 II 部分「管理工作」	
第 3 章「啟動代理程式和客戶端」	描述如何啟動 Message Queue 代理程式和客戶端。
第 4 章「配置代理程式」	描述如何設定及調整配置特性，並介紹代理程式的可配置功能。此外，還描述如何設定檔案或資料庫以執行永久性功能。
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」	描述與實體目標相關的管理工作。
第 7 章「管理安全性」	描述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」	描述物件存放區，並說明如何執行與管理物件（連線工廠和目標）相關的工作。
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。
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。

表 1 本手冊的內容 (續)

部分/章	說明
第 11 章 「分析與最佳化訊息服務效能」	描述分析與最佳化訊息服務效能的技術。
第 12 章 「排解困難問題」	提供判斷一般 Message Queue 問題成因和解決方法的建議。
第 III 部分 「參照」	
第 13 章 「指令行參照」	提供 Message Queue 指令行程式的語法和說明。
第 14 章 「代理程式特性參照」	列出並說明可以配置代理程式的特性。
第 15 章 「實體目標特性參照」	列出並說明可以配置實體目標的特性。
第 16 章 「管理物件屬性參照」	列出並說明可以配置管理物件 (連線工廠和目標) 的特性。
第 17 章 「JMS 資源介面特性參照」	列出並說明可以配置 Message Queue 資源介面的特性, 以便與應用程式伺服器時使用。
第 18 章 「度量參照」	列出並說明 Message Queue 代理程式產生的度量。
第 IV 部分 「附錄」	
附錄 A 「Message Queue 資料的平台特定位置」	列出每個支援平台上的 Message Queue 檔案位置。
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」	描述如何設定及使用超文本作業事件協定 (HTTP) 進行 Message Queue 通訊。
附錄 D 「常用指令行程式指令」	列出部分常用的 Message Queue 指令行程式 (mqcmd) 指令。

本書使用慣例

本節將描述本手冊所用的慣例。

文字慣例

表 2 摘要說明本線上手冊所用的文字慣例。

表 2 文字慣例

格式	說明
斜體	斜體文字表示應以適當陳述式或值取代的預留位置。亦用於所介紹的物件標題、強調內容，以及關鍵字等。
固定間距字體	固定間距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 (包括簽名中的所稱元素)、套裝軟體名稱、保留字以及 URL。
[]	指示指令行語法敘述中可選值的方括號。方括號也用於表示對話方塊中的所稱 UI 視窗、按鈕、選項、訊息和對話方塊標題等等。
全部大寫	全部大寫文字表示環境變數 (如 IMQ_HOME) 或首字母縮寫 (如 JMS、GIF 或 HTML)。
「書名」	書名。
鍵+ 鍵	同時按下的按鍵透過加號連接在一起：Ctrl+A 表示同時按下這兩個鍵。
鍵- 鍵	連續按下的按鍵透過連字號連接在一起：Esc-S 表示按下 Esc 鍵，然後釋放，接著按下 S 鍵。

目錄變數慣例

Message Queue 使用三個目錄變數，其設定方式因平台而異。表 3 描述這些變數，並概述在 Solaris™、Linux 和 Windows 平台上的使用方法。

備註	在本手冊中，不使用特定平台語法顯示目錄變數 (如 UNIX 為 \$IMQ_HOME)。路徑名稱通常使用 UNIX 目錄分隔符號表示法 (/)。
----	--

表 3 Message Queue 目錄變數

變數	說明
IMQ_HOME	<p>請參閱 Message Queue 基底目錄 (即根安裝目錄) :</p> <ul style="list-style-type: none"> • Solaris 和 Linux 上不使用此變數, 沒有 Message Queue 基底目錄。 • 在 Windows 上, 由 Message Queue 安裝程式設定 (預設位置為 C:\Program Files\Sun\MessageQueue3)。 • 如為 Solaris 及 Windows 上的 Sun Java System 應用程式伺服器, 則設為應用程式伺服器基底目錄下的 /imq。
IMQ_VARHOME	<p>儲存在 Message Queue 暫存或動態建立的配置和資料檔案的目錄, 可設為環境變數指代任何目錄。</p> <ul style="list-style-type: none"> • 在 Solaris 上, 預設位置為 /var/imq。 • 在 Linux 上, 預設位置為 /var/opt/sun/mq 目錄。 • 在 Windows 上, 預設位置為 IMQ_HOME\var。 • 在 Solaris 上, Sun Java System Application Server, Evaluation Edition 的預設位置為 IMQ_HOME/var。 • 在 Windows 上, Sun Java System Application Server 的預設位置為 IMQ_HOME\var。
IMQ_JAVAHOME	<p>Message Queue 可執行檔所需的 Java™ 執行階段 (JRE) 位置。依預設, 設定為按所示順序在下列位置中尋找, 但亦可設定為在所需 JRE 駐留的任意目錄中尋找。</p> <ul style="list-style-type: none"> • 在 Solaris 8 或 9 上 : <ul style="list-style-type: none"> /usr/jdk/entsys-j2se /usr/jdk/jdk1.5.* /usr/jdk/j2sdk1.5.* /usr/j2se • 在 Solaris 10 上 : <ul style="list-style-type: none"> /usr/jdk/entsys-j2se /usr/java /usr/j2se • 在 Linux 上 : <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 : <ul style="list-style-type: none"> IMQ_HOME\jre*

相關文件

本節列出的資訊資源，可提供本手冊以外有關 Message Queue 的進一步資訊。

Message Queue 文件集

Message Queue 文件集由表 4 列出的文件構成。

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	包括對新增功能、限制、已知錯誤以及技術性注意事項的描述。
「Message Queue 技術摘要」	開發者與管理員	介紹 Message Queue 概念、功能以及元件。
「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	提供使用 JMS 及 SOAP/JAXM 規格之 Message Queue 實作開發 Java 客戶端程式的相關資訊。
「Message Queue Developer's Guide for C Clients」	開發者	提供使用 Message Queue 訊息服務功能的 C 應用程式程式設計介面 (C-API)，開發 C 客戶端程式的相關資訊。

Java Message Service 規格

Message Queue 訊息服務符合「Java Message Service 規格」中說明的 Java Message Service (JMS) 應用程式程式設計介面。本文件位於 URL

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

線上說明

Message Queue 命令行公用程式提供線上說明，詳細資訊請參閱第 13 章「[指令行參照](#)」。Message Queue 圖形化使用者介面 (GUI) 管理工具 (即「[管理主控台](#)」) 也包括上下文相關的線上說明功能，請參閱第 38 頁的「[管理主控台線上說明](#)」。

JavaDoc

表 5 中所示的位置，會提供 JavaDoc 格式的 JMS 和 Message Queue API 文件。本文件可在任何 HTML 瀏覽器 (如 Netscape 或 Internet Explorer) 中檢視。其中包含標準的 JMS API 文件及 Message Queue 特定的 API。

表 5 JavaDoc 位置

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

範例用戶端應用程式

Message Queue 安裝包括一個內含數個範例用戶端應用程式的目錄。請參閱附錄 A「[Message Queue 資料的平台特定位置](#)」，以取得所用特定平台的確切位置。該目錄及其子目錄中的讀我檔案，皆提供了關於範例應用程式的說明資訊。

相關協力廠商網站

遇有相關主題時，本手冊會參照提供其他相關資訊的協力廠商 URL。

備註

Sun 對於本手冊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。若要提出您的意見，請至

<http://docs.sun.com>

並按一下 [傳送您的回饋意見] (Send Comments)。在線上表格中，請提供文件標題及文件號碼。文件號碼位於書本的標題頁或文件的頂部，通常是一組七位或九位數的數字。

提出意見時您還需要在表格中輸入此文件的英文標題和文件號碼。例如，本文件的英文文件號碼為 819-2571，完整標題為「Sun Java™ System Message Queue 3 Administration Guide」。

Message Queue 管理簡介

第 1 章 「管理工作和工具」

第 2 章 「快速入門指導」

管理工作和工具

本章提供執行 Sun Java™ System Message Queue 管理工作及工具之簡介，說明重點為指令行管理公用程式的一般功能。包含下列各節：

- [第 30 頁的「管理工作」](#)
- [第 32 頁的「管理工具」](#)

管理工作

一般執行的管理工作，會依執行 Message Queue 所在環境的本質而定。開發及測試 Message Queue 應用程式之軟體開發環境的需求，與部署此類應用程式以完成具體工作所需要之生產環境截然不同。後續章節將摘要說明這兩種不同環境類型的一般管理需求。

開發環境中的管理

開發環境重點強調的是彈性。Message Queue 訊息伺服器主要用於測試開發中的應用程式。因為程式設計師通常會自行管理其系統，所以一般不太需要管理。這種環境一般具有下列特徵：

- 啟動用於測試的代理程式非常容易
- 管理物件由用戶端程式碼而非管理端建立
- 自動建立的目標
- 檔案系統物件存放區
- 檔案型永久性
- 檔案型使用者儲存庫
- 多代理程式叢集中沒有主代理程式

生產環境中的管理

在必須可靠地部署和執行應用程式之生產環境中，管理變得更加重要。執行的管理工作依訊息傳送系統及其必須支援之應用程式的複雜性而定。這種工作可分為兩大類：設定作業與維護作業。

設定作業

生產環境的管理設定作業，一般包含下列部分或全部的項目：

管理員安全性

- 設定預設管理使用者的密碼 (admin) (第 131 頁的「變更預設管理員密碼」)
- 控制個人或群組對管理連線服務 (第 137 頁的「連線服務的存取控制」) 與停用訊息佇列 (第 139 頁的「實體目標的存取控制」) 的存取

- 規範管理群組對檔案型或簡易資料存取協定 (LDAP) 使用者儲存庫的存取 (第 129 頁的「群組」、第 133 頁的「設定管理員的存取控制」)

一般安全性

- 管理檔案型使用者儲存庫的內容 (第 130 頁的「寫入和管理使用者儲存庫」) 或配置代理程式使用現有的 LDAP 使用者儲存庫 (第 132 頁的「編輯實例配置檔案」)
- 控制個別使用者或群組授權執行的作業 (第 134 頁的「授權使用者：存取控制特性檔案」)
- 使用安全套接層 (SSL) 設定加密服務 (第 140 頁的「使用 SSL 型服務」)

管理物件

- 設定及配置 LDAP 物件存放區 (第 154 頁的「LDAP 伺服器物件存放區」)
- 建立連線工廠和目標 (第 163 頁的「新增管理物件」)

代理程式叢集

- 建立叢集配置檔案 (第 173 頁的「使用叢集配置檔案」)
- 指定主代理程式 (第 176 頁的「主代理程式」)

永久性

- 配置代理程式使用永久存放區 (第 89 頁的「配置永久資料存放區」)

記憶體管理

- 設定目標的配置特性，以最佳化記憶體使用情況 (第 117 頁的「更新實體目標特性」、第 15 章「實體目標特性參照」)

維護作業

因為應用程式的效能、穩定性及安全性在生產環境中極為重要，所以必須透過永久性的管理維護作業，緊密監視訊息伺服器資源，這些作業包括：

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 (第 11 章「分析與調校訊息服務」)
- 管理代理程式記憶體資源 (第 76 頁的「路由服務」)
- 建立與管理代理程式叢集，以平衡訊息負載 (第 9 章「使用代理程式叢集」)
- 回復失敗的代理程式 (第 67 頁的「啟動代理程式」)

管理物件

- 調整連線工廠屬性，以確保用戶端應用程式運作正確 (第 156 頁的「[連線工廠屬性](#)」)
- 監視與管理實體目標 (第 6 章「[管理實體目標](#)」)
- 控制使用者對目標的存取 (第 139 頁的「[實體目標的存取控制](#)」)

用戶端管理

- 監視與管理長期訂閱 (請參閱第 107 頁的「[管理長期訂閱](#)」)
- 監視與管理作業事件 (請參閱第 108 頁的「[管理作業事件](#)」)

管理工具

Message Queue 管理工具分為兩類：

- 指令行公用程式
- 圖形化管理主控台

指令行公用程式

所有 Message Queue 公用程式均可透過指令行介面存取。公用程式指令共用一般的格式、語法慣例與選項。這些部分包括：

- *代理程式公用程式 (mqbrokerd)* 會啟動代理程式，指定其配置特性，並將其一併連線至叢集。
- *指令行公用程式 (mqcmd)* 會控制代理程式及其資源，並管理實體目標。
- *物件管理員公用程式 (mqobjmgr)* 會管理物件存放區中，可透過 Java Naming and Directory Interface (JNDI) 來存取的獨立於提供者的**管理物件**。
- *資料庫管理員公用程式 (mqdbmgr)* 會建立與管理符合 Java 資料庫連線 (JDBC) 標準的永久存放區資料庫。
- *使用者管理員公用程式 (mqusermgr)* 會寫入檔案型使用者儲存庫，以執行使用者認證與授權。
- *服務管理員公用程式 (mqsvcadm)* 會安裝與管理作為 Windows 服務的代理程式。

- 金鑰工具公用程式 (imqkeytool) 會產生自身簽名的憑證，以執行安全套接層 (SSL) 認證。

如需使用這些公用程式的詳細資訊，請參閱第 13 章「指令行參照」。

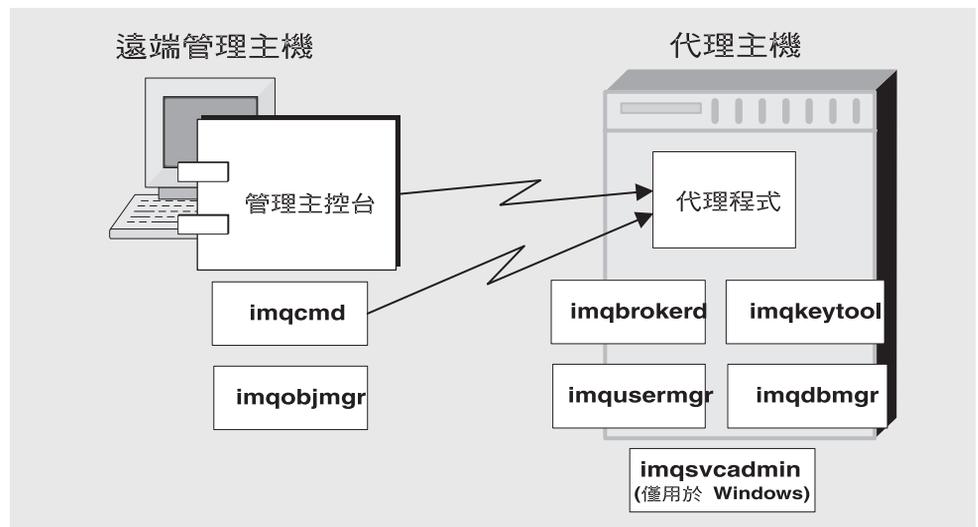
管理主控台

Message Queue 管理主控台結合了指令與物件管理員公用程式的部分功能。您可以使用它執行下列工作：

- 遠端連線並控制代理程式
- 建立與管理實體目標
- 在 JNDI 物件存放區中建立與管理物件

但是，您無法使用管理主控台執行諸如啓動代理程式、建立代理程式叢集、管理 JDBC 資料庫或使用者儲存庫、安裝作為 Windows 服務的代理程式，或是產生 SSL 憑證等作業。如要執行上述工作，您必須使用其他指令行公用程式 (代理程式、資料庫管理員、使用者管理員、服務管理員以及金鑰工具)，此類程式無法從遠端操作，同時必須與受其管理的代理程式在同一部主機上執行 (請參閱圖 1-1)。

圖 1-1 本機與遠端管理公用程式



如需管理主控台的簡易使用簡介，請參閱第 2 章「快速入門指導」。更多詳細使用資訊，請參閱其各自的說明功能。

快速入門指導

本快速入門指導將帶領您使用 Message Queue 管理主控台 (管理訊息代理程式及物件存放區的圖形化介面)，完成一些基本的管理工作，讓您對 Message Queue 管理有初步的認識。本章包含下列各節：

- 第 37 頁的「啓動管理主控台」
- 第 38 頁的「管理主控台線上說明」
- 第 39 頁的「使用代理程式」
- 第 44 頁的「使用實體目標」
- 第 49 頁的「使用物件存放區」
- 第 52 頁的「使用管理物件」
- 第 57 頁的「執行範例應用程式」

本指導旨在設定執行簡單 JMS 相容應用程式 HelloWorldMessageJNDI 時，所需的實體目標和管理物件。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 資料的平台特定位置」) 的 helloworld 子目錄中。您會在本指導的最後階段執行此應用程式。

備註 必須先安裝 Message Queue 產品，才能繼續本指導。如有需要，請參閱「Message Queue Installation Guide」的說明。

本指導僅為基本說明，無法取代文件的功用。遵循本指導中所述各步驟，將學習到如何執行下列作業：

- 啓動訊息代理程式
- 連線至代理程式以及使用管理主控台管理代理程式

- 在代理程式上建立實體目標
- 建立物件存放區以及使用管理主控台連線至此物件存放區
- 將管理物件新增至物件存放區，並檢視其特性

備註 本指導所提供的說明係針對 Windows 平台。必要時，會為其他平台使用者提供補充說明。

有些管理工作無法使用管理主控台完成。類似下列的工作，必須使用指令行公用程式來執行：

- 啟動代理程式
- 建立代理程式叢集
- 配置特定實體目標特性
- 管理永久存放區的 JDBC 資料庫
- 管理使用者儲存庫
- 安裝作為 Windows 服務的代理程式
- 產生 SSL 憑證

本線上手冊會在後續各章討論上述所有工作。

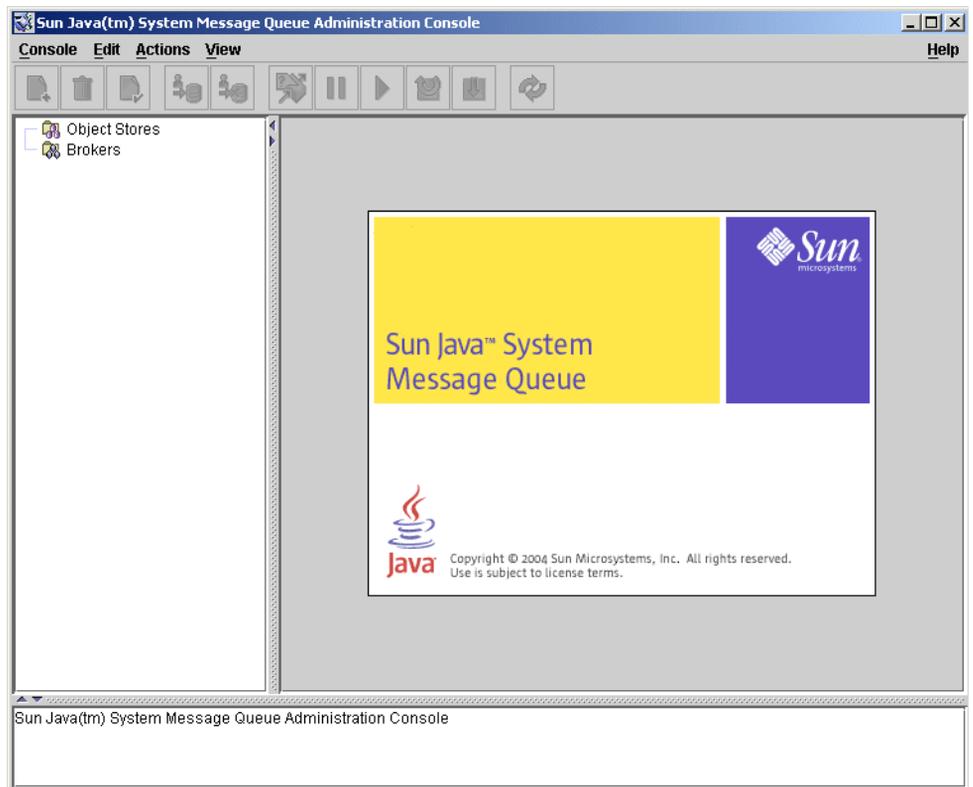
啟動管理主控台

若要啟動管理主控台，請執行下列方法之一：

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

顯示 [管理主控台] 視窗需等待數秒 (請參閱圖 2-1)。

圖 2-1 管理主控台視窗



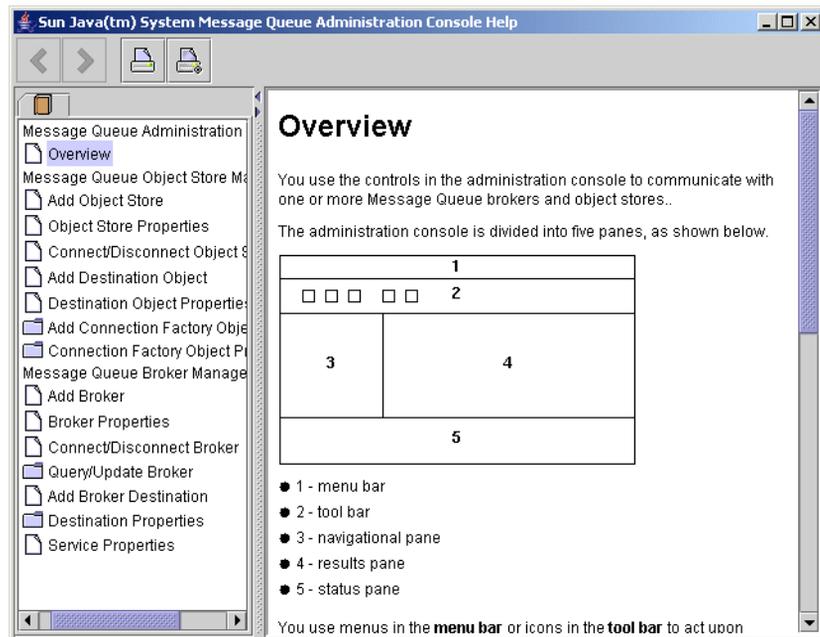
系統檢查 [管理主控台] 視窗會需要幾秒鐘的時間。頂端有一個功能表列，功能表列下方為工具列，左側為導覽窗格，右側為結果窗格 (此刻顯示的是 Sun Java System Message Queue 產品的圖形)，底部則為狀態窗格。

提示 使用 [管理主控台] 時，您可使用 [檢視] 功能表的 [重新整理] 指令，更新任何元素或元素群組的可視顯示畫面，如代理程式或物件存放區清單等。

管理主控台線上說明

[管理主控台] 的說明功能中，包含有關使用 [主控台] 執行管理工作的完整資訊。若要使用說明功能，請將功能表列最右邊的 [說明] 功能表向下拉開，然後選擇 [概述]。即會顯示 [管理主控台] 的 [說明] 視窗 (圖 2-2)。

圖 2-2 管理主控台說明視窗



[說明] 視窗左側的導覽窗格，將所有主題組織為三個區域：Message Queue 管理主控台、Message Queue 物件存放區管理及 Message Queue 代理程式管理。各區域內皆有檔案及資料夾。資料夾內是多重標籤對話方塊的說明，檔案中則是簡單對話方塊或個別標籤的說明。當您選取導覽窗格中的項目時，右側的結果窗格就會顯示該項目的內容。選取 [概述] 項目時，結果窗格會顯示 [管理主控台] 視窗的框架檢視，指出該視窗的各個窗格，如圖所示。

使用管理主控台執行的第一項工作，即是建立代理程式的參照。但在開始之前，請先查看 [說明] 視窗的資訊。按一下 [說明] 視窗導覽窗格中的 [新增代理程式] 項目，結果窗格的內容會變更為顯示文字，解釋新增代理程式的涵義並說明 [新增代理程式] 對話方塊中各欄位的用法。請閱讀說明文字，然後關閉 [說明] 視窗。

使用代理程式

本節描述如何使用管理主控台，連線至訊息代理程式與管理訊息代理程式。

啟動代理程式

您無法使用管理主控台啟動代理程式。請使用下列方法之一：

- 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`
- 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]。

若使用 Windows 的 [開始] 功能表，會出現表示代理程式已就緒的指令視窗，顯示如下所示的內容：

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

重新啟動 [管理主控台] 視窗。此時，即可將代理程式新增至 [主控台] 並與其連線。當您在 [管理主控台] 中為代理程式新增參照時，不必先啟動代理程式；但若需要連線至代理程式，就必須先啟動它。

將代理程式新增至管理主控台

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

► 將代理程式新增至管理主控台

1. 按一下 [管理主控台] 視窗之導覽窗格的 [代理程式] 項目，然後選擇 [動作] 功能表的 [新增代理程式]。

或者，您也可以按在 [代理程式] 上按一下滑鼠右鍵，然後從快顯功能表中選擇 [新增代理程式]。不論何種方式，都會出現 [新增代理程式] 對話方塊 (圖 2-3)。

圖 2-3 [新增代理程式] 對話方塊



2. 請在 [代理程式標籤] 欄位中輸入代理程式名稱。

這可提供識別管理主控台中代理程式的標籤。

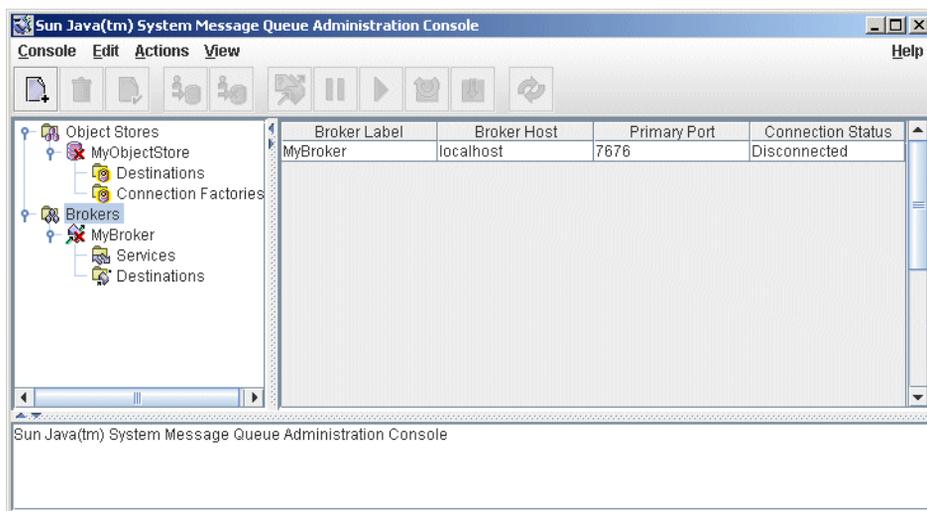
請注意對話方塊中指定的預設主機名稱 (localhost) 和主要連接埠 (7676)。稍後當您配置用戶端用以與此代理程式連線的連線工廠時，必須指定這些值。

在此練習中，請將名稱 MyBroker 輸入至 [代理程式標籤] 欄位。[密碼] 欄位請留白，於連線時再指定密碼較為安全。

3. 按一下 [確定] 以新增代理程式，並結束對話方塊。

新的代理程式會出現在導覽窗格的 [代理程式] 下，如圖 2-4 中所示。代理程式圖示上的紅色 X，表示其目前尚未連線至管理主控台。

圖 2-4 管理主控台視窗中顯示的代理程式



只要新增了代理程式，就可使用 [動作] 功能表 (或快顯功能表) 的 [特性] 指令，顯示 [代理程式特性] 對話方塊 (與圖 2-3 中所顯示的 [新增代理程式] 對話方塊類似)，檢視或修改其任一特性。

連線至代理程式

如此即已將代理程式新增至管理主控台，接著即可繼續連線作業。

► 連線至代理程式

1. 按一下 [管理主控台] 視窗之導覽窗格內的代理程式名稱，然後選擇 [動作] 功能表的 [連線至代理程式]。

或者，也可以在代理程式名稱上按一下滑鼠右鍵，然後從快顯功能表中選擇 [連線至代理程式]。不論何種方式，都會出現 [連線至代理程式] 對話方塊 (圖 2-5)。

圖 2-5 [連線至代理程式] 對話方塊



2. 請輸入用以連線至代理程式的使用者名稱及密碼。

對話方塊一開始會顯示預設的使用者名稱，即 `admin`。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 126 頁的「認證使用者」）；在此練習中，使用預設值即可。

與預設使用者名稱相關聯的密碼亦為 `admin`，請輸入至對話方塊的 [密碼] 欄位。如此將會連線至該代理程式，並擁有管理權限。

3. 按一下 [確定] 以連線至代理程式，並結束對話方塊。

只要連線至代理程式，就可使用 [動作] 功能表（或快顯功能表）的指令，對選取的代理程式執行下列作業：

- [暫停代理程式] 會暫時停止執行中的代理程式作業。
- [繼續代理程式] 會繼續暫停的代理程式作業。
- [重新啟動代理程式] 會重新初始化並重新啟動代理程式。
- [關閉代理程式] 會終止代理程式作業。
- [查詢 / 更新代理程式] 會顯示或修改代理程式的配置特性。
- [取消與代理程式的連線] 會終止代理程式和 [管理主控台] 之間的連線。

檢視連線服務

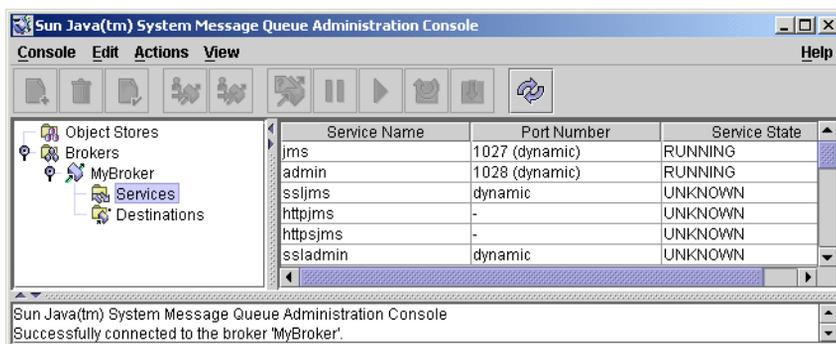
透過代理程式提供的連線服務和它支援的實體目標可以識別此代理程式。

► 檢視可用的連線服務

1. 在 [管理主控台] 視窗的導覽窗格中，選取代理程式名稱下的 [服務]。

結果窗格中會出現可用服務的清單（請參閱圖 2-6），顯示名稱、連接埠號碼以及各服務目前的狀態。

圖 2-6 檢視連線服務



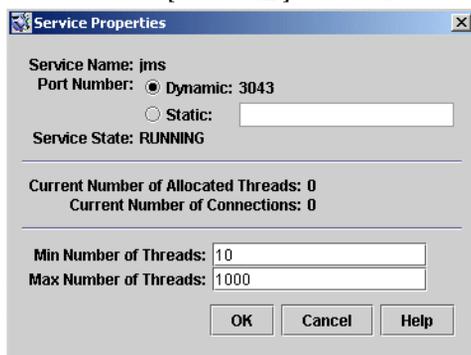
- 請在結果窗格中按一下服務名稱，以選取該項服務。

在此練習中，請選取名稱 `jms`。

- 從 [動作] 功能表中選擇 [特性]。

如此即會出現 [服務特性] 對話方塊 (圖 2-7)。您可使用此對話方塊為此服務指定靜態連接埠號，並變更為其配置之最小及最大執行緒數。

圖 2-7 [服務特性] 對話方塊



在本練習中，請勿變更任何連線服務的特性。

- 按一下 [確定] 以接受新的特性值，並結束對話方塊。

[動作] 功能表中也包含了暫停和繼續服務的指令。當您選取 `admin` 服務並下拉 [動作] 功能表時，[暫停服務] 指令會是停用狀態。這是因為 `admin` 服務是 [管理主控台] 對代理程式的連結：如果暫停此服務，將無法再存取代理程式。

使用實體目標

*實體目標*是訊息代理程式上的一個位置，其保留接收自訊息產生者的訊息，稍後再傳送至一個或多個訊息用戶。目標會依使用的*訊息傳送網域*分為兩種：*佇列*（點對點網域）與*主題*（出版 / 訂閱網域）。如需進一步有關訊息傳送網域及其相關聯目標的討論，請參閱「Message Queue 技術摘要」。

建立實體目標

依預設，訊息代理程式的配置，是當訊息產生者或用戶嘗試存取不存在的目標時，能夠自動建立新的實體目標。在軟體開發環境中測試用戶端程式碼時，這類*自動建立的目標*非常好用。但是在生產設定中，不但建議停用自動建立目標，最好所有目標都只能由管理員建立。下列程序說明如何將這種*管理員建立的目標*新增至代理程式。

► 將實體目標新增至代理程式

1. 請在 [管理主控台] 視窗的導覽窗格中，按一下代理程式名稱下的 [目標] 項目，然後選擇 [動作] 功能表的 [新增代理程式目標]。

或者，也可以在 [目標] 上按一下滑鼠右鍵，然後從快顯功能表中選擇 [新增代理程式目標]。不論何種方式，都會出現 [新增代理程式目標] 對話方塊 (圖 2-8)。

圖 2-8 [新增代理程式目標] 對話方塊

Destination Name:

Destination Type: Queue
 Topic

Max Number of Messages:
 Unlimited
 0

Max Total Message Bytes:
 Unlimited
 0

Max Bytes per Message:
 Unlimited
 0

Max Number of Producers:
 Unlimited
 100

Max Number of Active Consumers:
 Unlimited
 1

Max Number of Backup Consumers:
 Unlimited
 0

OK Reset To Defaults Cancel Help

2. 請在 [目標名稱] 欄位中輸入該實體目標的名稱。

請記下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管理物件時，將需要此名稱。

在此練習中，請輸入名稱 MyQueueDest。

3. 選取 [佇列] 或 [主題] 單選按鈕，指定要建立的目標類型。

在此練習中，若尚未選取，請選取 [佇列]。

4. 按一下 [確定] 以新增實體目標，並結束對話方塊。

該實體目標會隨即出現在結果窗格中。

檢視實體目標特性

您可以在 [管理主控台] 上，使用 [動作] 功能表的 [特性] 指令，檢視或修改實體目標的特性。

► 檢視或修改實體目標的特性

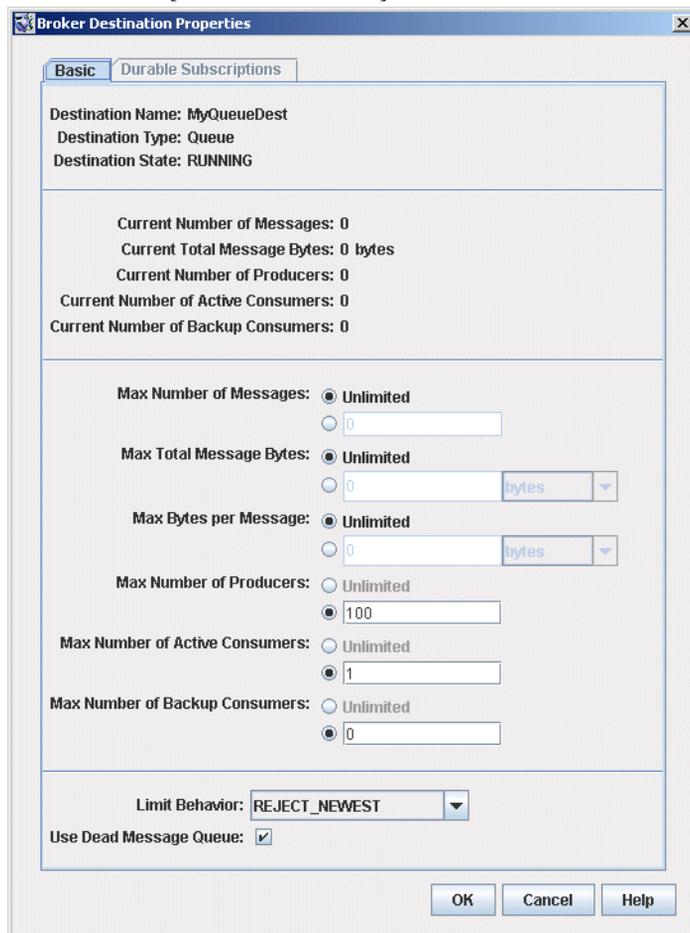
1. 在 [管理主控台] 視窗的導覽窗格中，選取代理程式名稱下的 [目標]。

結果窗格中會出現可用實體目標的清單，顯示名稱、類型以及各目標目前的狀態。

2. 請在結果窗格中按一下實體目標名稱，以選取該目標。
3. 從 [動作] 功能表中選擇 [特性]。

如此即會出現 [代理程式目標屬性] 對話方塊 ([圖 2-9](#))，顯示所選實體目標的目前狀態及配置資訊。您可使用此對話方塊變更各項配置特性，如該目標能容納的訊息、產生者及用戶上限。

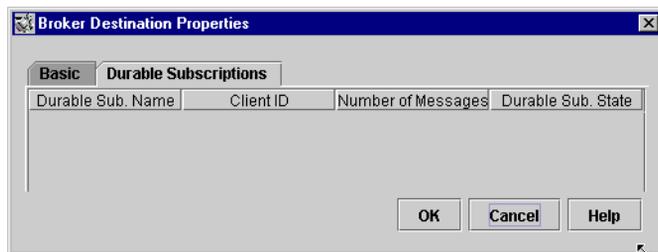
圖 2-9 [代理程式目標屬性] 對話方塊



在本練習中，請勿變更任何目標特性。

至於主題目標的 [代理程式目標屬性] 對話方塊中，則另有一個 [長期訂閱] 標籤。按一下此標籤即會顯示 [長期訂閱] 畫面 (圖 2-10)，列出目前與指定主題相關聯之所有長期訂閱的相關資訊。

圖 2-10 [長期訂閱] 畫面



您可使用 [長期訂閱] 畫面的 [清除] 和 [刪除] 按鈕，執行下列作業：

- 清除所有與長期訂閱相關聯的擱置訊息
- 移除主題的長期訂閱

[長期訂閱] 標籤對於佇列目標為停用狀態。

4. 按一下 [確定] 以接受新的特性值，並結束對話方塊。

清除實體目標的訊息

清除實體目標的訊息，會移除所有與該目標相關聯的擱置訊息，清空該目標。

► 清除實體目標的訊息

1. 在 [管理主控台] 視窗的導覽窗格中，選取代理程式名稱下的 [目標]。
結果窗格中會出現可用實體目標的清單，顯示名稱、類型以及各目標目前的狀態。
2. 請在結果窗格中按一下目標名稱，以選取該目標。
3. 從 [動作] 功能表中選擇 [清除訊息]。
如此即會出現確認對話方塊，要求您確認是否繼續該作業。
4. 按一下 [是] 以確認作業，並結束確認對話方塊。

刪除實體目標

刪除目標會先清除其全部訊息，然後再銷毀目標本身，將其自所屬之代理程式中永久移除。

► 刪除實體目標

1. 在 [管理主控台] 視窗的導覽窗格中，選取代理程式名稱下的 [目標]。
結果窗格中會出現可用目標的清單，顯示名稱、類型以及各目標目前的狀態。
2. 請在結果窗格中按一下目標名稱，以選取該目標。
3. 從 [編輯] 功能表中選擇 [刪除]。
如此即會出現確認對話方塊，要求您確認是否繼續該作業。
4. 按一下 [是] 以確認作業，並結束確認對話方塊。
在此練習中，請勿刪除之前建立的目標 MyQueueDest，而按一下 [否] 會結束確認對話方塊，不執行刪除作業。

使用物件存放區

物件存放區會用以儲存 Message Queue 管理物件，其中封裝了特定 Message Queue 提供者的專有實作及配置資訊。物件存放區可以是「簡易資料存取協定 (LDAP)」目錄伺服器，或本機檔案系統的目錄。

雖然可以直接從用戶端應用程式的程式碼內，建立與配置管理物件，但一般慣用的方式是讓管理員建立與配置這些物件，並將其儲存在物件存放區中，讓用戶端應用程式能夠使用 Java Naming and Directory Interface (JNDI) 存取這些物件。這樣可讓用戶端程式碼獨立於供應程式之外。

新增物件存放區

雖然管理主控台可讓您管理物件存放區，您卻無法用它建立物件存放區；用作物件存放區的 LDAP 伺服器或檔案系統目錄，必須在建立物件存放區之前即已存在。然後您才能將現有的物件存放區新增至管理主控台，並為其建立參照，用以在 [主控台] 內操作此物件存放區。

注意

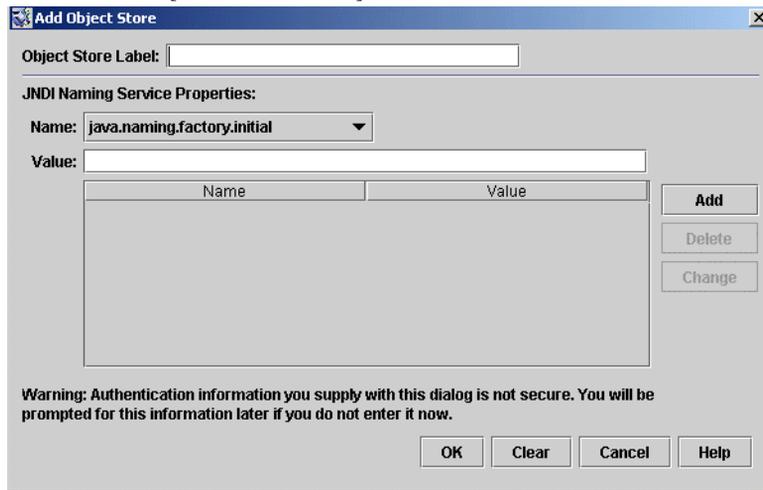
本章使用的範例應用程式，會假設物件儲存位於 C 磁碟機的 Temp 目錄中。如果 C 磁碟機上還沒有名為 Temp 的資料夾，請先建立一個再繼續後續練習。(在非 Windows 平台上，您可以使用 /tmp 目錄，此目錄應已存在。)

► 將物件存放區新增至管理主控台

1. 按一下 [管理主控台] 視窗之導覽窗格的 [物件存放區] 項目，然後選擇 [動作] 功能表的 [新增物件存放區]。

或者，您也可以按 [物件存放區] 上按一下滑鼠右鍵，然後從快顯功能表中選擇 [新增物件存放區]。不論何種方式，都會出現 [新增物件存放區] 對話方塊 (圖 2-11)。

圖 2-11 [新增物件存放區] 對話方塊



2. 請在 [物件存放區標籤] 欄位中，輸入物件存放區名稱。
這可提供識別管理主控台中物件存放區的標籤。
在此練習中，請輸入名稱 MyObjectStore。
3. 輸入要用以尋找管理物件的 JNDI 屬性值：
 - a. 請從 [名稱] 下拉式功能表中，選取要指定的屬性名稱。
 - b. 在 [數值] 欄位中輸入屬性值。
 - c. 按一下 [新增] 按鈕，增加指定的屬性值。
[特性摘要] 窗格中會出現特性及其值。
請為每個需要設定的屬性重複步驟 a 至 c。
在此練習中，將 java.naming.factory.initial 屬性設為

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

而將 `java.naming.provider.url` 屬性設為

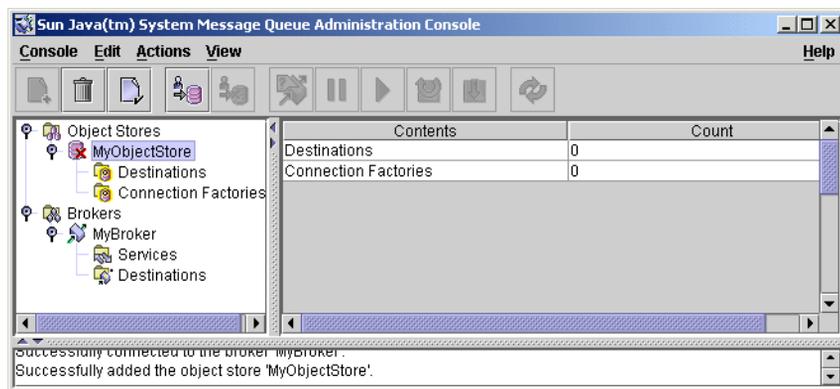
```
file:///C:/Temp
```

(在 Solaris 或 Linux 平台上則為 `file:///tmp`)。針對檔案系統物件存放區，您僅需設定這些屬性；如需 LDAP 存放區必要屬性值的相關資訊，請參閱第 154 頁的「LDAP 伺服器物件存放區」。

4. 按一下 [確定] 以新增物件存放區，並結束對話方塊。

新的物件存放區會出現在導覽窗格的 [物件存放區] 下，如圖 2-12 中所示。物件存放區圖示上的紅色 X 表示其目前尚未連線至管理主控台。

圖 2-12 管理主控台視窗中顯示的物件存放區



當您按一下導覽窗格中的物件存放區時，其內容會列在結果窗格中。因為您尚未在物件存放區中新增任何管理物件，所以目標和連線工廠的 [計數] 欄都會顯示 0。

只要新增了物件存放區，就可使用 [動作] 功能表 (或快顯功能表) 的 [特性] 指令，顯示 [物件存放區特性] 對話方塊 (與圖 2-11 的 [新增物件存放區] 對話方塊類似)，檢視或修改其任一特性。

連線至物件存放區

現在已在 [管理主控台] 中新增了物件存放區，就必須與其建立連線，才能將管理物件新增至此物件存放區中。

► **連線至物件存放區**

1. 按一下 [管理主控台] 視窗之導覽窗格內的物件存放區名稱，然後選擇 [動作] 功能表的 [連線至物件存放區]。

或者，您也可以物件存放區名稱上按一下滑鼠右鍵，然後從快顯功能表中選擇 [連線至物件存放區]。不論何種方式，物件存放區圖示的紅色 X 都會消失，表示其已與 [管理主控台] 連線。

使用管理物件

只要物件存放區連線至管理主控台，您就可繼續將管理物件 (連線工廠及目標) 新增至此物件存放區。本節會描述作業方法。

備註 管理主控台僅會顯示 Message Queue 管理物件。如果物件存放區中有非 Message Queue 物件，與想要新增的管理物件具有相同的查詢名稱，則當您嘗試新增作業時即會出現錯誤。

新增連線工廠

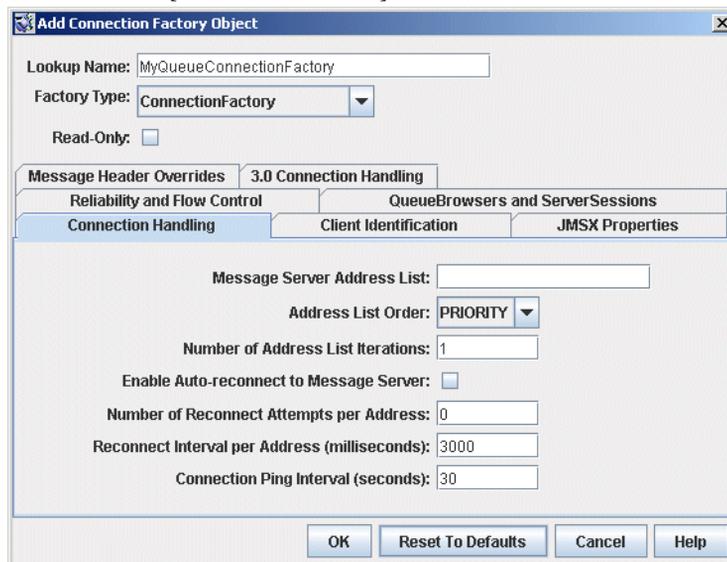
連線工廠 為用戶端應用程式所用，以建立與代理程式的連線。只要配置連線工廠，即可控制其所建連線的特性。

► **將連線工廠新增至物件存放區**

1. 請務必確認物件存放區已連線至管理主控台 (請參閱第 51 頁的「[連線至物件存放區](#)」)。
2. 請在 [管理主控台] 視窗的導覽窗格中，按一下該物件存放區名稱下的 [連線工廠] 項目，然後選擇 [動作] 功能表的 [新增連線工廠物件]。

或者，您也可以 [連線工廠] 上按一下滑鼠右鍵，然後從快顯功能表中選擇 [新增連線工廠物件]。不論何種方式，都會出現 [新增連線工廠物件] 對話方塊 ([圖 2-13](#))。

圖 2-13 [新增連線工廠物件] 對話方塊



3. 請在 [查詢名稱] 欄位中輸入該連線工廠的名稱。
這是用戶端應用程式以 JNDI 查詢連線工廠時所使用的名稱。
在此練習中，請輸入名稱 `MyQueueConnectionFactory`。
4. 請從 [工廠類型] 下拉式功能表中，選擇您希望建立的連線工廠類型。
在此練習中，請選擇 [`QueueConnectionFactory`]。
5. 按一下 [連線處理] 標籤。
即會出現 [連線處理] 窗格，如圖 2-13 中所示。
6. 請在 [訊息伺服器位址清單] 欄位中，填入將要與此連線工廠建立連線的代理程式位址。

此位址清單可能是由一個代理程式或 (若為代理程式叢集的情況) 多個代理程式所組成。它會針對每個代理程式指定代理程式連線服務、主機名稱及連接埠號碼等資訊。實際指定的資訊性質和語法各異，視所要使用的連線服務而定；請參閱第 294 頁的「連線處理」以獲得詳細資訊。

在此練習中，[訊息伺服器位址清單] 欄位中不必輸入任何位址，因為範例應用程式 `HelloWorldMessageJNDI` 預期連線工廠會使用預設自動配置的標準位址清單屬性 (連線服務為 `.jms`、主機名稱為 `localhost` 及連接埠號碼為 `7676`)。

7. 請視需要配置連線工廠的其他屬性。

[新增連線工廠物件] 對話方塊除了 [連線處理] 窗格外，另有數個其他畫面，可用以配置連線工廠的各種屬性。

在本練習中，請勿變更任何其他屬性設定。但是，您還是可以按其他標籤，瞭解可指定的配置資訊種類，增進對此項作業的認識。請使用 [說明] 按鈕瞭解有關這些其他配置畫面的內容。

8. 在適當情況下，按一下 [唯讀] 核取方塊。

如此會鎖定連線工廠物件的配置屬性，保持為建立時所指定的值。不論是用戶端程式碼的程式設計方式或指令行的管理方式，都無法覆寫唯讀的管理物件屬性。

在本練習中，請勿核取 [唯讀]。

9. 按一下 [確定] 以建立連線工廠，將其新增至物件儲存，然後結束對話方塊。

新的連線工廠會出現在結果窗格中。

新增目標

目標管理物件代表代理程式上的實體目標，可讓用戶端傳送訊息到該實體目標，而不受提供者特有配置及命名語法的影響。當用戶端傳送經由管理物件發出的訊息時，代理程式就會將該訊息傳送至對應的實體目標 (如存在)。如果上述實體目標不存在，代理程式就會自動建立一個實體目標 (如已啓用自動建立，如第 44 頁的「[建立實體目標](#)」中所述)，並將訊息傳送至此；否則，就會產生無法傳送訊息的錯誤訊息。

下列程序描述將目標管理物件，新增至與現有實體目標相對應之物件存放區的方法：

► 將目標新增至物件存放區

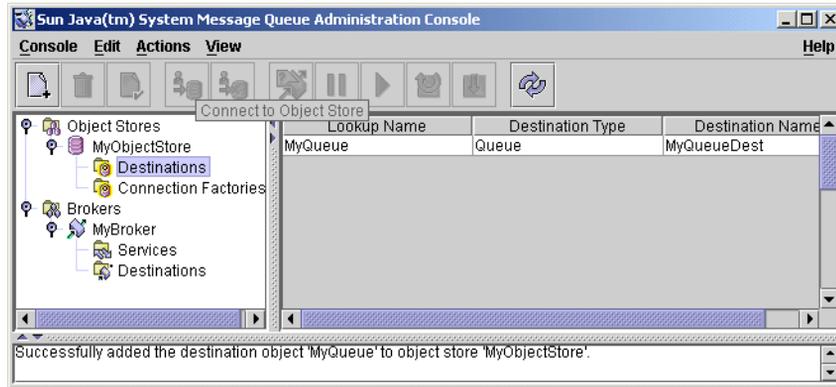
1. 請務必確認物件存放已連線至管理主控台 (請參閱第 51 頁的「[連線至物件存放區](#)」)。
2. 請在 [管理主控台] 視窗的導覽窗格中，按一下物件存放區名稱下的 [目標] 項目，然後選擇 [動作] 功能表的 [新增目標物件]。

或者，您也可以按 [目標] 上按一下滑鼠右鍵，然後從快顯功能表中選擇 [新增目標物件]。不論何種方式，都會出現 [新增目標物件] 對話方塊 (圖 2-14)。

圖 2-14 [新增目標物件] 對話方塊

3. 請在 [查詢名稱] 欄位中，輸入該目標管理物件的名稱。
這是用戶端應用程式以 JNDI 查詢目標時所使用的名稱。
在此練習中，請輸入名稱 MyQueue。
4. 選取 [佇列] 或 [主題] 單選按鈕，指定要建立的目標物件類型。
在此練習中，若尚未選取，請選取 [佇列]。
5. 請在 [目標名稱] 欄位中，輸入對應實體目標的名稱。
這是您在代理程式上新增實體目標時指定的名稱 (請參閱第 44 頁的「使用實體目標」)。
在此練習中，請輸入名稱 MyQueueDest。
6. 或在 [目標說明] 欄位中，輸入該目標的簡短說明。
此欄位的內容僅供使用者參考，對用戶端作業沒有任何影響。
在此練習中，您可以刪除 [目標說明] 欄位的內容，或輸入一些說明性的文字，例如
Example destination for MQ Admin Guide tutorial
7. 在適當情況下，按一下 [唯讀] 核取方塊。
如此會鎖定目標物件的配置屬性，保持為建立時所指定的值。不論是用戶端程式碼的程式設計方式或指令行的管理方式，都無法覆寫唯讀的管理物件屬性。
在本練習中，請勿核取 [唯讀]。
8. 按一下 [確定] 以建立目標物件，將其新增至物件存放區，然後結束對話方塊。
新的目標物件會隨即出現在結果窗格中，如圖 2-15 所示。

圖 2-15 管理主控台視窗中顯示的目標物件



檢視管理物件特性

您可以在 [管理主控台] 上，使用 [動作] 功能表的 [特性] 指令，檢視或修改管理物件的特性。

► 檢視或修改管理物件的特性

1. 在 [管理主控台] 視窗的導覽窗格中，選取物件存放區名稱下的 [連線工廠] 或 [目標]。

結果窗格中就會出現可用連線工廠或目標管理物件的清單，顯示各項目的查詢名稱和類型 (目標管理物件清單時還包括目標名稱)。

2. 請在結果窗格中按一下管理物件的名稱，以選取該管理物件。
3. 從 [動作] 功能表中選擇 [特性]。

即會出現 [連線工廠物件特性] 或 [目標物件特性] 對話方塊，與 [新增連線工廠物件] (第 53 頁的圖 2-13) 或 [新增目標物件] (第 55 頁的圖 2-14) 對話方塊類似。您可使用此對話方塊變更所選物件的配置屬性。但請注意，您無法變更物件的查詢名稱；唯一的方法是刪除該物件，然後再以想要的查詢名稱新增新的管理物件。

4. 按一下 [確定] 以接受新的屬性值，並結束對話方塊。

刪除管理物件

刪除管理物件會將其從所屬物件存放區中永久移除。

► 刪除管理物件

1. 在 [管理主控台] 視窗的導覽窗格中，選取物件存放區名稱下的 [連線工廠] 或 [目標]。

結果窗格中就會出現可用連線工廠或目標管理物件的清單，顯示各項目的查詢名稱和類型 (目標管理物件清單時還包括目標名稱)。

2. 請在結果窗格中按一下管理物件的名稱，以選取該管理物件。
3. 從 [編輯] 功能表中選擇 [刪除]。

即會出現確認對話方塊，要求您確認是否繼續該作業。

4. 按一下 [是] 以確認作業，並結束確認對話方塊。

在此練習中，請勿刪除之前所建立的管理物件 `MyQueue` 或 `MyQueueConnectionFactory`，按一下 [否] 結束確認對話方塊，不執行刪除作業。

執行佇列應冊 程式

本指導使用的範例應用程式為 `HelloWorldMessageJNDI`。此範例應用程式使用您建立的實體目標和管理物件：

- 佇列實體目標的名稱是 `MyQueueDest`
- 佇列連線工廠管理物件的 JNDI 查詢名稱是 `MyQueueConnectionFactory`
- 佇列管理物件的 JNDI 查詢名稱是 `MyQueue`

此程式碼會建立簡單的佇列傳送者和接收者，並傳送和接收 `Hello World` 訊息。

執行此應用程式前，請先開啓原始碼檔案 `HelloWorldMessageJNDI.java`，並閱讀程式碼。此程式很短，同時有詳細的說明，瞭解其運作方式應該不難。

► 執行範例應用程式

1. 請使用下列指令之一 (依所用平台而定)，將包含 `HelloWorldmessageJNDI` 應用程式的目錄成爲目前的目錄：

- 在 Solaris 上：

```
cd /usr/demo/imq/helloworld/helloworldmessagejndi
```

- 在 Linux 上：

```
cd /opt/sun/mq/examples/helloworld/helloworldmessagejndi
```

- 在 Windows 上：

```
cd IMQ_HOME\demo\helloworld\helloworldmessagejndi
```

您應會發現 HelloWorldMessageJNDI.class 檔案已存在。(應用程式若有變更，必須使用「Message Queue Developer's Guide for Java Clients」中提供的用戶端應用程式編譯程序，重新編譯該應用程式。)

2. 設定 CLASSPATH 變數，以納入包含檔案 HelloWorldMessageJNDI.class 的目前目錄以及包含在 Message Queue 產品中的 .jar 下列檔案：

```
jms.jar  
imq.jar  
jndi.jar  
fscontext.jar
```

請參閱「Message Queue Developer's Guide for Java Clients」，以取得設定 CLASSPATH 變數的資訊。

備註 jndi.jar 檔案隨附於 JDK 1.4。除非使用較舊版本的 JDK，否則不必將此檔案新增至 CLASSPATH。

3. 執行下列指令之一(依所用平台而定)，執行 HelloWorldMessageJNDI 應用程式：

- 在 Solaris 或 Linux 上：

```
% java HelloWorldMessageJNDI file:///tmp
```

- 在 Windows 上：

```
java HelloWorldMessageJNDI
```

如果此應用程式執行成功，您應該會看到如[程式碼範例 2-1](#)所示的輸出。

程式碼範例 2-1 範例應用程式的輸出

```
java HelloWorldMessageJNDI
Using file:///C:/Temp for Context.PROVIDER_URL

Looking up Queue Connection Factory object with lookup name: MyQueueConnectionFactory
Queue Connection Factory object found.
Looking up Queue object with lookup name: MyQueue
Queue object found.

Creating connection to broker.
Connection to broker created.

Publishing a message to Queue: MyQueueDest
Received the following message: 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，就可以開始啟動代理程式和用戶端。代理程式的配置由一組配置檔案所管理，可由傳送給代理程式公用程式 (imqbrokerd) 的指令行選項加以覆寫；請參閱第 4 章「配置代理程式」，以取得更多資訊。

本章包含下列各節：

- 第 66 頁的「準備系統資源」
- 第 67 頁的「啟動代理程式」
- 第 71 頁的「移除代理程式」
- 第 72 頁的「啟動用戶端」

準備系統資源

啓動代理程式之前，必須預先執行兩個系統層級的作業：同步化系統時鐘，以及 (在 Solaris 或 Linux 平台上) 設定檔案描述元限制。以下各節將描述這些作業。

同步化系統時鐘

啓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 (存在時間)，則同步化尤為重要。不同步的時鐘時間戳記，可能會讓訊息無法按預期那樣過期或無法傳送訊息。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，例如簡易網路管理協定 (SNTP)。Solaris 和 Linux 的 xntpd 常駐程式和 Windows 的 W32Time 服務，通常可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，請避免將系統時鐘往回設定。

設定檔案描述元限制

在 Solaris 和 Linux 平台上，執行用戶端或代理程式所在的 Shell，會對程序可以使用的檔案描述元的數目指定軟式限制。在 Message Queue 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一項。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，檔案描述元限制會限制代理程式或用戶端可擁有的連線數目。依預設，Solaris 上的最大連線數目為 256，而 Linux 上為 1024。(實際上，因為會持續使用檔案描述元，所以連線限制會低於上述數目)。如果需要超過此限制的連線數目，您必須在執行用戶端或代理程式的每個 Shell 中，增加檔案描述元限制。如需如何執行此作業的詳細資訊，請參閱 ulimit 線上手冊。

啓動代理程式

您可以使用 **Message Queue** 命令行公用程式或 Windows [開始] 功能表，以互動方式啓動代理程式，或是安排在系統啓動時自動啓動代理程式。以下各節描述如何：

以互動方式啓動代理程式

您可以使用代理程式公用程式 (`imqbrokerd`)，從命令行以互動方式啓動代理程式。(或者，您可以從 Windows 的 [開始] 功能表，啓動代理程式)。您不能使用 [管理主控台] (`imqadmin`) 或指令公用程式 (`imqcmd`) 來啓動代理程式，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，必須由最初啓動平台的同一位使用者啓動。每個代理程式實例都有自己的配置特性和檔案型訊息存放區。初次啓動代理程式實例時，**Message Queue** 會使用該使用者的檔案建立模式遮罩 (`umask`)，設定包含該代理程式實例之配置資訊和永久性資料目錄的權限。

代理程式實例的預設名稱是 `imqbroker`。您只要使用下列指令，即可從命令行以該名稱和預設配置來啓動代理程式：

```
imqbrokerd
```

該指令會在本地機器上使用預設連接埠 7676 上的連接埠對映器，啓動代理程式實例 `imqbroker` (請參閱第 75 頁的「[連接埠對映器](#)」)。

若要指定非預設的實例名稱，請在 `imqbrokerd` 指令上使用 `-name` 選項。下列指令啓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。此範例設定 `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、子指令和選項的完整資訊，請參閱第 249 頁的「[代理程式公用程式](#)」。如需這項資訊的快速摘要，請輸入下列指令：

```
imqbrokerd -help
```

注意 如果您有 Sun Java System Message Queue Platform Edition 授權，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

除了明確地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。執行此作業的方式會因執行代理程式的平台 (Solaris、Linux 或 Windows) 而異。

在 Solaris 和 Linux 上自動啟動

在 Solaris 和 Linux 系統上安裝 Message Queue 時，可以自動啟動的程序檔位於 `/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`。
- 若要設定代理程式的啟動指令行引數，請指定一個或多個 `ARGS` 特性值。

在 Windows 上自動啟動

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。然後就會在系統啟動時啟動代理程式，並且在背景中執行到系統關機為止。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。

一個系統只能有一個代理程式執行為 Windows 服務。[工作管理員] 會將這類代理程式當作兩個可執行的程序列出：

- 原生 Windows 服務包裝程式 `imqbrokersvc.exe`

- 執行代理程式的 Java 執行階段

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
- 移除執行為 Windows 服務的代理程式

若要將啟動選項傳給代理程式，可在 imqsvcadmin 指令中使用 -args 引數。這等同於第 67 頁的「啟動代理程式」中描述的 imqbrokerd 指令的 -D 選項。使用指令公用程式 (imqcmd) 以控制代理程式照常運作。

如需 imqsvcadmin 指令的語法、子指令和選項的完整資訊，請參閱第 263 頁的「服務管理員公用程式」。

重新配置代理程式服務

如下所示，重新配置安裝為 Windows 服務的代理程式：

► 重新配置執行為 Windows 服務的代理程式

1. 停止服務。
 - a. 從 Windows [開始] 功能表的 [設定] 子功能表，選擇 [控制台]。
 - b. 開啓 [系統管理工具] 控制台。
 - c. 選取 [服務] 圖示以執行服務工具，然後從 [檔案] 功能表或快顯式環境功能表選擇 [開啓]，或是直接連接兩下該圖示。
 - d. 在 [服務 (本機)] 下方，選取 [訊息佇列代理程式] 服務，然後從 [執行] 功能表選擇 [內容]。

或者，可以在 [訊息佇列代理程式] 上按一下滑鼠右鍵，然後從快顯式環境功能表選擇 [內容]，或是直接連接兩下 [訊息佇列代理程式]。不論何種方式，都會出現 [訊息佇列代理程式特性] 對話方塊。

- e. 在 [內容] 對話方塊中的 [一般] 標籤下方，按一下 [停止] 即可停止代理程式服務。
2. 移除服務。

在指令行上輸入指令：

```
imqsvcadmin remove
```

3. 重新安裝服務，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 版本引數。

例如，要將服務的主機名稱與連接埠號碼變更為 `broker1` 與 `7878`，可以使用以下指令：

```
imgsvcadmin install -args "-name broker1 -port 7878"
```

使用替代 Java 執行階段

您可以使用 `imgsvcadmin` 指令的 `-javahome` 或 `-jrehome` 選項，指定替代 Java 執行階段的位置。(您也可以在服務的 [內容] 對話方塊視窗中，於 [一般] 標籤下方的 [啟動參數] 欄位中，指定這些選項)。

注意 [啟動參數] 欄位會將反斜線字元 (\) 視為退出字元，因此當您以反斜線作為路徑區隔字元時，必須輸入兩次：例如，

```
-javahome c:\\j2sdk1.4.0
```

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 `imgsvcadmin` 指令中使用 `query` 選項，如程式碼範例 3-1 所示。

程式碼範例 3-1 顯示代理程式服務啟動選項

```
imgsvcadmin query

Service Message Queue Broker is installed.
Display Name:Message Queue Broker
Start Type:Automatic
Binary location:C:\Sun\MessageQueue\bin\mqbrokersvc.exe
JavaHome:c:\j2sdk1.4.0
Broker Args:-name broker1 -port 7878
```

服務啟動問題疑難排解

如果在啟動作為 Windows 服務的代理程式時出現錯誤，可以檢視已記錄的錯誤事件：

► **查閱記錄的服務錯誤事件**

1. 開啓 Windows [系統管理工具] 控制台。

2. 啟動 [事件檢視器] 工具。
3. 選取 [應用程式事件記錄檔]。
4. 從 [執行] 功能表選取 [重新整理]，以顯示任何錯誤事件。

移除代理程式

移除代理程式的程序會因平台不同而異，將會在以下各節說明。

移除 Solaris 或 Linux 上的代理程式

若要移除 Solaris 或 Linux 平台上的代理程式實例，請在 `imqbrokerd` 指令中使用 `-remove` 選項。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
```

此指令可刪除所指定代理程式的整個實例目錄。

如果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`。

如需 `imqbrokerd` 指令的語法、子指令和選項的詳細資訊，請參閱第 249 頁的「代理程式公用程式」。如需這項資訊的快速摘要，請輸入下列指令：

移除 Windows 上的代理程式服務

若要移除執行為 Windows 服務的代理程式，請使用以下指令：

```
imqcmd shutdown bkr
```

若要關閉代理程式，請在後面加上：

```
imqsvcadmin remove
```

以移除服務。

或者，您可以透過 [系統管理工具] 控制台，來使用 Windows [服務] 工具，以停止和移除代理程式服務。

請在移除代理程式服務之後，重新啓動您的電腦。

啓動用戶端

啓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。如果您啓動了 Java 用戶端應用程式，就必須正確設定 CLASSPATH 變數，並且確定已安裝正確的 .jar 檔案。「Message Queue Developer's Guide for Java Clients」中包含設定系統的一般步驟資訊，但是您的開發者可能會提供額外的資訊。

啓動 Java 用戶端應用程式時，請使用下列的指令行格式：

```
java clientAppName
```

啓動 C 用戶端應用程式時，請使用應用程式開發者提供的格式。

應用程式的文件應會提供與應用程式設定屬性值相關的資訊；您可能想要從指令行來覆寫部分屬性值。可能還要為任何使用 Java Naming and Directory Interface (JNDI) 查詢功能來尋找連線工廠的 Java 用戶端，在指令行中指定屬性。如果查詢功能傳回的連線工廠比應用程式還舊，表示連線工廠不支援較新的屬性。在這種情況下，Message Queue 會將這些屬性設為預設值；您可以在必要時，使用指令行來覆寫這些預設值。

若要從指令行指定 Java 應用程式的屬性值，請使用以下語法：

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

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

以下範例可啓動稱爲 MyMQClient 的用戶端應用程式，利用連接埠 7677 連線至主機 OtherHost 上的代理程式：

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

指令行中指定的主機名稱與連接埠，會覆寫應用程式本身的任何設定。

有時您無法使用指令行來指定屬性值。管理員可以將管理物件設為只能讀取，或應用程式開發者能以程式碼讓用戶端應用程式執行此作業。若要瞭解啓動用戶端程式的最佳方法，最好與應用程式開發者溝通交流。

配置代理程式

代理程式的配置由一組配置檔案和在啓動時傳給 `imqbrokerd` 指令的選項所管理。本章說明可用的配置特性，以及如何使用它們來配置代理程式。

本章包含下列各節：

- [第 74 頁的「代理程式服務」](#)
- [第 86 頁的「設定代理程式特性」](#)
- [第 89 頁的「配置永久資料存放區」](#)

如需代理程式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」](#)。

代理程式服務

視服務或服務影響所及的代理程式元件之不同，代理程式配置特性可分為數個類別：

- **連線服務**可管理代理程式與其用戶端之間的實體連線，執行內送和外寄訊息作業。
- **路由服務**可路由及傳送 JMS 有效負載訊息，以及控制訊息服務所使用的訊息，以支援可靠的傳送。
- **永久性服務**可管理永久存放區的寫入資料與擷取資料。
- **安全性服務**可認證連線至代理程式的使用者，並授權其動作。
- **監視服務**可產生有關代理程式效能的度量與診斷資訊。

以下各節將分別說明這些服務，以及您根據自身需要而自訂服務時所使用的特性。

連線服務

訊息代理程式透過多種傳輸協定提供各種支援應用程式與管理用戶端的**連線服務**。與連線服務相關的代理程式配置特性，列於第 266 頁的「**連線特性**」中。

表 4-1 說明可用的連線服務，依據兩種特徵區分：

- **服務類型**會指定此服務是否提供 JMS 訊息傳送 (NORMAL) 或是 Message Queue 管理服務 (ADMIN)。
- **協定類型**可指定基本的傳輸協定。

表 4-1 Message Queue 連線服務

服務名稱	服務類型	協定類型
jms	NORMAL	TCP
ssljms (Enterprise Edition)	NORMAL	TLS (SSL 型的安全性)
httpjms (Enterprise Edition)	NORMAL	HTTP
httpsjms (Enterprise Edition)	NORMAL	HTTPS (SSL 型的安全性)
admin	ADMIN	TCP
ssladmin	ADMIN	TLS (SSL 型的安全性)

透過設定代理程式的 `imq.service.activelist` 特性，即可將其配置為執行任一或所有這些連線服務。此特性的值是一組會在代理程式啟動時啟動的連線服務清單；若未明確指定此特性，將依預設啟動 `jms` 與 `admin` 服務。

每個連線服務也支援特定的認證與授權功能，請參閱第 81 頁的「安全性服務」，以取得更多資訊。

連接埠對映器

每種連線服務可用於特定的連接埠，此連接埠由主機名稱 (或 IP 位址) 和連接埠號碼指定。您可以明確地指定服務的靜態連接埠號碼，或由代理程式的 *連接埠對映器* 動態指定。連接埠對映器本身常駐於代理程式的 *主連接埠* 上，即通常位於標準連接埠號碼 7676。(若有必要，您可以使用代理程式配置特性 `imq.portmapper.port`，以其他連接埠號碼來覆寫此值。) 依預設，每個連線服務皆會在啟動時對連接埠對映器進行自我註冊。當用戶端建立與代理程式的連線時，Message Queue 用戶端執行階段會先連絡連接埠對映器，以請求連接埠號碼供需要的連線服務使用。

此外，您也可以使用 `imq.serviceName.protocolType.port` 配置特性，覆寫連接埠對映器，並明確指定連線服務的靜態連接埠號碼 (其中 `serviceName` 與 `protocolType` 表示特定的連線服務，如表 4-1 所示)。(只有 `jms`、`ssljms`、`admin` 與 `ssladmin` 連線服務能夠以這種方式進行配置，`httpjms` 與 `httpsjms` 服務使用不同的配置特性，如附錄 C「HTTP/HTTPS 支援」所述。) 但靜態連接埠通常只適用於特殊情況 (例如建立通過防火牆的連線)，而不建議用於一般情況。

備註 在具有兩部或更多主機的環境下 (例如在電腦上安裝了多張網路卡時)，您可以使用代理程式特性來指定連線服務所要連結的主機。`imq.hostname` 特性可為所有連線服務指定單一的預設主機；而此預設值可在必要時，以 `imq.serviceName.protocolType.hostname` (針對 `jms`、`ssljms`、`admin` 或 `ssladmin` 服務) 或 `imq.portmapper.hostname` (針對連接埠對映器本身) 進行覆寫。

同時收到多個連接埠管理員請求時，請求會儲存在作業系統儲存區中等候執行動作。`imq.portmapper.backlog` 特性會針對這些位於儲存區中的請求，指定數量上限。超過此上限時，就會拒絕其他請求，直到儲存區量降低為止。

執行緒池管理

每個連線服務均為多重執行緒，支援多重連線。代理程式會針對每個服務，在個別的执行緒池中維護這些連線所需的執行緒。在連線需要執行緒時，這些執行緒就會新增到執行緒池中，供支援該連線的服務使用。

您所選擇的執行緒模型可指定執行緒是由單一連線所專用，還是由多個連線所共用：

- 在**專屬模型**中，代理程式的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。這會限制可支援的連線數量，但可確保較高的效能。
- 在**共用模型**中，連線在傳送或接收訊息時會由共用執行緒進行處理。此模型中的各個連線不需要專屬執行緒，因此能夠提高可用連線的數量，但會降低效能，因為必須使用較多資源來管理執行緒。

代理程式的 `imq.serviceName.threadpool_model` 特性，可指定給定的連線服務應使用這兩種模型中的哪一種。此特性會採用下列兩個字串值之一：專屬或共用。若未明確設定特性，則會依預設採用專屬。

您也可以設定代理程式特性 `imq.serviceName.min_threads` 與 `imq.serviceName.max_threads`，來指定服務執行緒池的最小與最大執行緒數。當可用的執行緒數超過指定的最小臨界值時，**Message Queue** 將會在執行緒成為可用狀態時加以關閉，直到再次達到最小值為止，藉以節省記憶體資源。在負載較大的情況下，執行緒的數量會增加，直至達到執行緒池的最大數量為止；此時會拒絕新的連線，直到某執行緒成為可用狀態為止。

共用執行緒模型會使用**分散執行緒**，將執行緒指定給使用中的連線。代理程式特性 `imq.shared.connectionMonitor_limit` 可指定由單一分散執行緒監視的最大連線數。此特性的值越小，執行緒指定給連線的速度就越快。`imq.ping.interval` 特性會指定一個以秒為單位的時間間隔，讓代理程式依此間隔定期測試（「偵測」）連線，以驗證它是否仍在使用中，以便在嘗試傳輸訊息失敗前，即可預先偵測到連線失敗。

路由服務

用戶端連線至代理程式後，即可路由訊息及傳送訊息。在此階段中，代理程式必須建立及管理不同類型的實體目標，以確保訊息能順利傳送並有效利用資源。您可以使用第 267 頁的「**路由特性**」所述的代理程式配置特性，以根據應用程式需要的方式管理這些工作。

代理程式的效能和穩定性取決於如記憶體之類的可用系統資源，以及如何有效利用這類資源。您可以設定配置特性，以防止代理程式被大量內送訊息佔用，或防止記憶體不足。這些特性可在三個不同的層級上起作用，以確保訊息服務在資源不足的情況下仍可運作：

- **整個系統的訊息限制**完全適用於系統上所有的實體目標。其中包括代理程式保留訊息的數量上限 (`imq.system.max_count`)，以及此類訊息所佔用的位元組總數上限 (`imq.system.max_size`)。若達到其中一項限制，代理程式就會拒絕任何新的訊息，直到擱置的訊息低於該限制為止。此外對個別訊息的大小上限 (`imq.message.max_size`) 與收回過期訊息的時間間隔 (`imq.message.expiration.interval`) 方面，也有相關的限制。
- **個別目標限制**會針對傳送到特定實體目標的訊息制定流程。控制這些限制的配置特性在 [第 15 章「實體目標特性參照」](#) 中有所說明。這些限制包括：目標所能保留的訊息數量與大小、所能建立的訊息產生者與用戶的數量，以及可納入相同批次傳送到目標的訊息數量等等。

目標可配置為透過下列方式回應記憶體限制：減緩訊息產生者傳送訊息的速度、拒絕新的內送訊息，或丟棄最舊或優先順序最低的現有訊息。以這種方式從目標中刪除的訊息，可選擇性地移至停用訊息佇列中，而不會直接捨棄，代理程式特性 `imq.destination.DMQ.truncateBody` 可控制是整個訊息內文還是只有標頭與特性資料儲存在停用訊息佇列中。

為了在開發及測試應用程式時方便作業，您可以配置訊息代理程式，使其在訊息產生者或用戶嘗試存取不存在的目標時，自動建立新的實體目標。[第 269 頁的表 14-3](#) 中所摘要的代理程式特性與剛剛說明的特性相同，但適用於 *自動建立的目標*，而非管理員所建立的目標。

- **系統記憶體臨界值**可定義代理程式為防止記憶體超過負載而採取越來越重要的動作時，使用記憶體的層級。這類的用量定義共有四種：
 - **綠色**：有大量記憶體可用。
 - **黃色**：代理程式記憶體開始減少。
 - **橙色**：代理程式記憶體不足。
 - **紅色**：代理程式無記憶體可用。

用以定義這些層級的記憶體用量百分比，分別由代理程式特性 `imq.green.threshold`、`imq.yellow.threshold`、`imq.orange.threshold` 與 `imq.red.threshold` 所指定，預設值為綠色 0%、黃色 80%、橙色 90%、紅色 98%。

記憶體用量進入下一層級時，代理程式即會隨之作出回應，首先將訊息由使用中記憶體移出至永久存放區，然後減少非永久性訊息的產生者，最後停止訊息傳入代理程式中。(上述兩種方法皆會降低代理程式效能。) 若要減少訊息產生，必須限制每個傳送到訊息數的批次大小 (由 `imq.resourceState.count` 特性所指定，其中 `resourceState` 分別為綠色、黃色、橙色或紅色)。

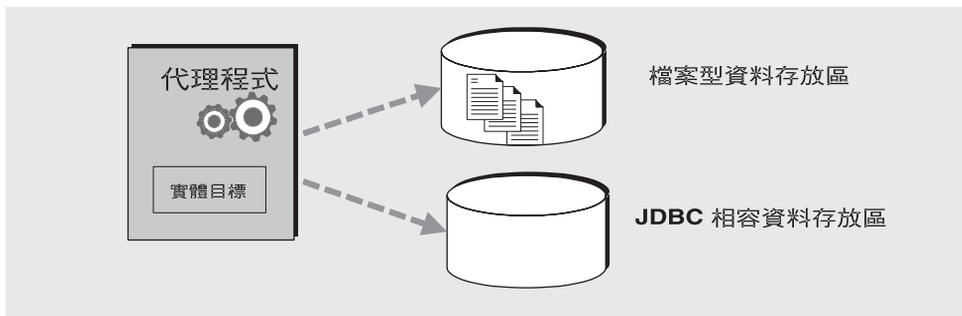
備註 這些系統記憶體臨界值被觸發時，可能表示整個系統與目標的訊息限制設定得太高。由於記憶體臨界值不一定總能及時偵測到可能的記憶體超載，因此您不應根據它們來控制記憶體用量，而應重新配置整個系統與目標限制，使記憶體資源最佳化。

永久性服務

如果代理程式要在失敗時回復，則此代理程式需要重新建立其訊息傳送作業的狀態。若要執行此作業，代理程式必須將狀態資訊儲存到**永久存放區**。代理程式重新啟動時，它會使用儲存的資料來重新建立目標和長期訂閱、回復永久性訊息、回復開啓的作業事件，以及為未傳送的訊息重新建立其路由表格。然後，可以繼續訊息遞送。

Message Queue 支援檔案型與 JDBC 型的永久性模組 (請參閱圖 4-1)。檔案型的永久性會使用個別的檔案來儲存永久性資料，而 JDBC 型的永久性則使用 Java 資料庫連線 (JDBC™) 介面，將代理程式連線至與 JDBC 相容的資料存放區。雖然檔案型的永久性速度通常比 JDBC 型的更快，但有些使用者還是偏好 JDBC 相容存放區提供的備援功能與管理控制功能。代理程式配置特性 `imq.persist.store` (請參閱第 271 頁的表 14-4) 可指定兩種形式的可用永久性。

圖 4-1 永久性資料存放區



檔案型永久性

依預設，Message Queue 會使用檔案型永久資料存放區，其中的個別檔案會儲存如訊息、目標、長期訂閱與永久性資料等。與檔案型永久性相關的代理程式配置特性，會在第 272 頁的「檔案型永久性」中列出。

檔案型存放區位於以資料存放區所屬之代理程式實例 (*instanceName*) 作為辨識依據的目錄中：

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

(請參閱附錄 A 「[Message Queue 資料的平台特定位置](#)」，以瞭解實例目錄的所在位置。) 代理程式上的每個目標，都會將其保留訊息的子目錄傳送到該目標上。

備註

由於永久存放區可能包含敏感或專有訊息，因此您應保護

`.../instances/instanceName/fs350/` 目錄，以防止未經授權的存取，請參閱第 91 頁的「[保護永久性資料](#)」。

所有非訊息的永久性資料，都儲存於個別檔案中：其中一個檔案用於儲存目標、一個檔案用於儲存長期訂閱，一個檔案用於儲存作業事件狀態資訊。大部分訊息會儲存在由變數型記錄組成的單一檔案中。您可以壓縮此檔案，以便減少新增或移除訊息時的分段程序 (請參閱第 120 頁的「[壓縮實體目標](#)」)。此外，超過特定臨界值大小的訊息會儲存於自身的個別檔案中，而非儲存於變數型記錄檔案。您可以使用代理程式特性 `imq.persist.file.message.max_record_size` 配置此臨界值大小。

代理程式會為這些個別訊息檔案保留檔案池：檔案不再使用時，不會被直接刪除，而會傳回其目標目錄的可用檔案池中，以供日後其他訊息重複使用。代理程式特性 `imq.persist.file.destination.message.filepool.limit` 可指定池中的最大檔案數。當目標中的個別訊息檔案數超過此限制時，若檔案已不需使用，就會直接被刪除，而不會傳回池中。

將檔案傳回檔案池時，代理程式可直接將檔案標示為可供重複使用，而不需刪除其先前的內容，這樣做可節省時間，但會耗用儲存空間。您可以使用 `imq.persist.file.message.filepool.cleanratio` 代理程式特性，指定每個目標的檔案池中有多少百分比的檔案應保持為清除 (空白) 狀態，而非直接標示為可供重複使用的狀態。此值設得越高，檔案池所需的空間就越少，但在檔案傳回池後，則需要較多的系統資源來清除檔案的內容。若代理程式的 `imq.persist.file.message.cleanup` 特性為 `true`，則池中所有的檔案就會在代理程式關閉時被清空，而成為清除狀態；如此將可節省儲存空間，但會減緩關閉程序的速度。

作業系統將資料寫入永久存放區時，可選擇應以同步還是延遲 (非同步) 的方式寫入資料。在系統當機的情況下，若代理程式認為資料已寫入永久存放區，但事實上並非如此時，延遲儲存就可能導致資料遺失。為確保完全的可靠性 (必須犧牲效能)，您可以將代理程式特性 `imq.persist.file.sync.enabled` 設為 `true`，使所有資料皆以同步方式寫入。如此一來，當系統從當機狀態回復時將必定會有可用的資料，讓代理程式能可靠地繼續執行作業。但請注意，雖然資料並未遺失，但叢集內的任何其他代理程式無法使用這些資料，因為叢集代理程式此時並未共用資料。

JDBC 型永久性

您也可以不使用檔案型永久性，而設定代理程式透過與 JDBC 相容的驅動程式存取任何可存取的資料存放區。其中包含設定適當的 JDBC 相關代理程式配置特性，以及使用資料庫管理員公用程式 (`imqdbmgr`)，建立具有適當模式的資料庫。如需特定資訊，請參閱第 90 頁的「配置 JDBC 型存放區」。

配置代理程式使用 JDBC 資料庫的特性，列於第 273 頁的「JDBC 型永久性」中。您可以透過以下方式指定這些特性：使用每個代理程式實例的實例配置檔 (`config.properties`)，或者使用代理程式公用程式 (`imqbrokerd`) 或資料庫管理員公用程式 (`imqdbmgr`) 的 `-D` 指令行選項。

`imq.persist.jdbc.driver` 特性可提供在連線資料庫時要使用的 JDBC 驅動程式之 Java 類別名稱。另外，其他特性亦可指定下列作業所需的 URL：連線至現有的資料庫 (`imq.persist.jdbc.opendburl`)、建立新資料庫 (`imq.persist.jdbc.createdburl`) 以及關閉資料庫連線 (`imq.persist.jdbc.closedburl`)。

`imq.persist.jdbc.user` 與 `imq.persist.jdbc.password` 特性，可提供存取資料庫所需的使用者名稱與密碼；`imq.persist.jdbc.needpassword` 是指定是否需要密碼的布林值旗標。基於安全理由，密碼僅能在透過 `-passfile` 指令行選項所指定的密碼檔案中；若未指定此密碼檔案，`imqbrokerd` 與 `imqdbmgr` 指令就會以互動方式提示使用者提供密碼。同樣地，使用者名稱也可以從指令行上透過 `imqbrokerd` 指令的 `-dbuser` 選項，或 `imqdbmgr` 的 `-u` 選項提供。

由多個代理程式實例共用的 JDBC 資料庫中，配置特性 `imq.persist.jdbc.brokerid` 可為每個實例指定唯一的實例識別碼，以附加到資料庫表格的名稱中。(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。) 其餘與 JDBC 相關的配置特性，可用以自訂建立資料庫模式的 SQL 程式碼，每個資料庫表格各有一個適用的特性。例如，

`imq.persist.jdbc.table.IMQSV35` 特性提供的 SQL 指令可用以建立版本表格；
`imq.persist.jdbc.table.IMQCCREC35` 可用於建立配置變更記錄表格；
`imq.persist.jdbc.table.IMQDEST35` 則用於建立目標表格，依此類推。請參閱
 第 273 頁的表 14-6 以取得完整清單。

備註 由於資料庫系統在實際所需的 SQL 語法中有所不同，因此請確定查看資料庫供應商提供的文件，以瞭解詳細資訊。

安全性服務

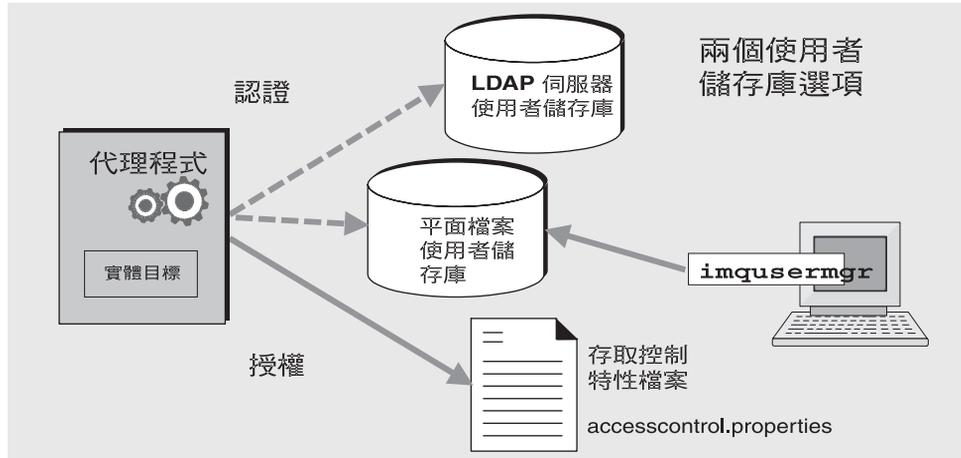
Message Queue 可提供使用者存取控制 (認證與授權) 與加密所需的安全性服務：

- *認證*可確保只有經過驗證的使用者可建立與代理程式的連線。
- *認證*可指定哪些使用者或群組有權可存取資源以及執行特定作業。
- *加密*可保護訊息，以防止透過連線執行傳送時遭到篡改。

身為 Message Queue 管理員，您必須負責設定代理程式進行使用者認證及使用者授權作業時所需的資訊。與安全性服務相關的代理程式特性，列於第 276 頁的「[安全性特性](#)」中。布林值特性 `imq.accesscontrol.enabled` 的角色為主要參數，可控制存取控制是否套用於整個代理程式中；若要進一步控制，您可以為特定的連線服務覆寫此設定，只要設定 `imq.serviceName.accesscontrol.enabled` 特性即可，其中 `serviceName` 是連線服務的名稱，如第 74 頁的表 4-1 所示：例如 `imq.httpjms.accesscontrol.enabled`。

圖 4-2 顯示代理程式提供認證與授權服務時所需的元件。這些服務取決於包含訊息傳送系統使用者相關資訊的 *使用者儲存庫*：其名稱、密碼與群組成員身份。此外，若要為使用者或群組的特定作業授權，代理程式必須查詢 *存取控制特性檔案*，因為此檔案指定了使用者或群組可執行的作業。您可以使用配置特性 `imq.accesscontrol.file.filename`，為整個代理程式指定單一存取控制特性檔案，也可以使用 `imq.serviceName.accesscontrol.file.filename`，為單一連線服務進行指定。

圖 4-2 安全性支援



如圖 4-2 所示，您可以將使用者資料儲存在隨附於 Message Queue 服務的平面檔案使用者儲存庫中，也可以插入預先存在的簡易資料存取協定 (LDAP) 儲存庫：

- 若您選擇平面檔案儲存庫，就必須使用 Message Queue 使用者管理員公用程式 (imqusermgr) 管理儲存庫。此為內建選項，使用非常方便。
- 如果您要使用現有的 LDAP 伺服器，可以使用 LDAP 供應商所提供的工具來寫入和管理此使用者儲存庫。您還必須設定代理程式實例配置檔中的特性，使代理程式能夠查詢 LDAP 伺服器，以取得使用者與群組的相關資訊。

代理程式的 `imq.authentication.basic.user_repository` 特性，可指定要使用的儲存庫類型。一般而言，若延伸性是主要考量，或者儲存庫需要由不同的代理程式共用 (例如，您使用的是代理程式叢集)，則使用 LDAP 儲存庫是不錯的選擇。如需設定平面檔案或 LDAP 使用者儲存庫的相關資訊，請參閱第 126 頁的「[認證使用者](#)」。

認證

請求連線至代理程式的用戶端必須提供使用者名稱與密碼，供代理程式將其與使用者儲存庫中儲存的資料進行比較。從用戶端傳輸到代理程式的密碼，會以基本 64 編碼 (適用於平面檔案儲存庫) 或訊息摘要 (MD5) 雜湊 (適用於 LDAP 儲存庫) 進行編碼。選擇哪一種編碼方式由代理程式的 `imq.authentication.type` 特性整體控制，或由 `imq.serviceName.authentication.type` 針對特定連線服務加以控制。`imq.authentication.client.response.timeout` 特性會設定認證請求的逾時間隔。

如第 149 頁的「使用密碼檔案」所述，您可以選擇將密碼存入密碼檔案中，而不需以互動方式提示您提供密碼。布林值代理程式特性 `imq.passfile.enabled` 可控制此選項。若此特性為 `true`，則 `imq.passfile.dirpath` 與 `imq.passfile.name` 特性就會提供密碼檔案的目錄路徑與檔案名稱。`imq.imqcmd.password` 特性 (可內嵌於密碼檔案中) 可指定管理使用者在使用指令公用程式 (`imqcmd`) 管理代理程式、連線服務、連線、實體目標、長期訂閱與作業事件前，所使用的認證密碼。

若您使用 LDAP 型使用者儲存庫，則會有全部的代理程式特性可供您配置各方面的 LDAP 查詢。LDAP 伺服器本身的位址 (主機名稱與連接埠號碼) 由 `imq.user_repository.ldap.server` 所指定。`imq.user_repository.ldap.principal` 特性會提供連結到 LDAP 儲存庫所需的識別名稱，而 `imq.user_repository.ldap.password` 則會提供相關的密碼。其他特性會指定個別使用者與群組搜尋的目錄基底與選用 JNDI 篩選器，以及使用者和群組名稱的提供者特定屬性識別碼等等；請參閱第 276 頁的「安全性特性」以瞭解詳細資訊。

授權

經過認證後，使用者即有權執行各項 Message Queue 的相關活動。身為 Message Queue 管理員，您可以定義使用者群組，並指定其個別的使用者成員身份。預設存取控制特性檔案會明確參考一個群組 `admin` (請參閱第 129 頁的「群組」)。此群組中的使用者具有 `admin` 連線服務的連線權限，此權限可讓使用者執行管理功能，如建立目標、監視及控制代理程式。依預設，您定義的任何其他群組中的使用者，均無法取得 `admin` 服務連線。

當使用者嘗試執行作業時，代理程式會檢查使用者名稱和群組成員身份 (從使用者儲存庫)，是否與為存取此作業所指定的那些名稱和成員身份 (在存取控制特性檔案中) 相符。存取控制特性檔案可指定使用者或群組執行下列作業時的權限：

- 連線至代理程式
- 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

加密

若要對在用戶端與代理程式之間傳送的訊息加密，需要使用安全套接層 (SSL) 型標準的連線服務。透過在已啟用 SSL 的代理程式與用戶端之間建立加密的連線，SSL 可提供連線層級的安全性。

若要使用 SSL 型的 Message Queue 連線服務，您必須使用金鑰工具公用程式 (imqkeytool)，產生成對的私密金鑰 / 公開金鑰。此公用程式會將公開金鑰內嵌於自身簽名的憑證中，並放置在 Message Queue 金鑰存放區中。金鑰存放區本身受密碼保護，若要解除鎖定，您必須提供在啟動時提供的金碼存放區密碼，此密碼由 imq.keystore.password 特性所指定。一旦金鑰存放區解除鎖定，代理程式即可將憑證傳送至請求連線的任何用戶端。然後，用戶端即可使用此憑證設定與代理程式的加密連線。

imq.audit.enabled 代理程式特性可控制記錄到 Message Queue 代理程式記錄檔中的稽核記錄；如需相關資訊，請參閱第 151 頁的「[建立稽核記錄](#)」。

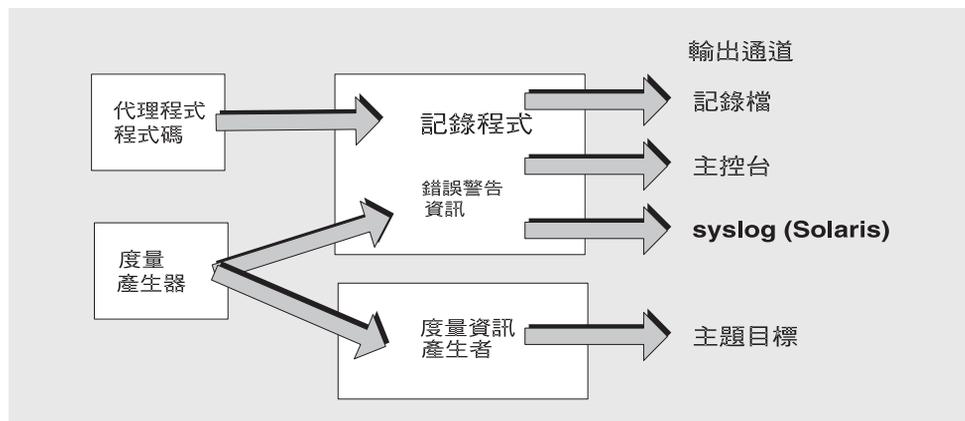
監視服務

代理程式包含可監視及診斷應用程式與代理程式效能的元件。這些元件包括：

- 產生資料的元件 (記錄事件的度量產生器與代理程式程式碼)
- 將輸出資訊寫入到多個輸出通道的記錄程式元件
- 度量訊息產生器，即將包含度量資訊的 JMS 訊息傳送到 JMS 監視用戶端使用的主題目標

圖 4-3 中會說明一般方案。用以配置及監視服務的代理程式特性，列於第 280 頁的「[檢視特性](#)」中。

圖 4-3 監視支援



度量產生器

度量產生器可提供代理程式活動的相關資訊，例如傳入和傳出代理程式的訊息流量、代理程式記憶體中的訊息數量和使用的記憶體、開啓連線的數量，以及使用的執行緒數量。布林值代理程式特性 `imq.metrics.enabled` 可控制是否記錄此類資訊；`imq.metrics.interval` 則指定其頻率。

記錄程式

記錄程式會擷取代理程式程式碼與度量產生器所產生的資訊，並將此資訊寫入標準輸出 (主控台)、記錄檔；若在 Solaris 平台上，在發生錯誤時則會寫入 `syslog` 常駐程式程序。要使用的記錄檔由 `imq.log.file.dirpath` 與 `imq.log.file.filename` 代理程式特性識別；`imq.log.console.stream` 會指定主控台輸出應導向 `stdout` 還是導向 `stderr`。

`imq.log.level` 特性可控制記錄程式所收集的度量資訊種類：ERROR、WARNING 或 INFO。每個層級都會包含更高的層級，因此，若您指定 WARNING 作為記錄層級，則也會記錄錯誤訊息。`imq.log.console.output` 與 `imq.log.file.output` 特性分別會控制哪些指定的種類會寫入主控台與記錄檔。但在此情況下，種類不`包含`比自身更高的層級；例如，如果要將錯誤與警告寫入記錄檔，且將參考訊息寫入主控台時，必須明確將 `imq.log.file.output` 設為 `ERROR|WARNING`，將 `imq.log.console.output` 設為 `INFO`。在 Solaris 平台上有另一個特性 `imq.log.syslog.output`，可指定要寫入 `syslog` 常駐程式的度量資訊種類。另外還有 `imq.destination.logDeadMsgs` 特性，可指定在停用訊息被捨棄或移至停用訊息佇列時，是否要加以記錄。

在記錄檔中，您可以指定何時關閉檔案並將輸出自動重建至新檔案。一旦記錄檔達到指定的大小 (`imq.log.file.rolloverbytes`) 或存在時間 (`imq.log.file.rolloversecs`)，就會將其儲存並建立新的記錄檔。

請參閱第 280 頁的「檢視特性」，以瞭解與記錄有關的其他代理程式特性，並參閱第 181 頁的「配置及使用代理程式記錄功能」，以進一步瞭解如何配置記錄程式，以及如何利用記錄程式取得效能資訊。

度量資訊產生者 (Enterprise Edition)

度量訊息產生者會在固定時間間隔，從度量產生器接收資訊，並將該資訊寫入度量訊息，然後根據訊息中包含的度量資訊類型，將訊息傳送到眾多度量主題目標之一 (請參閱表 4-2)。訂閱這些度量主題目標的 Message Queue 用戶端，可以使用這些訊息並處理訊息所包含的度量資料。這允許開發者建立自訂監視工具以支援訊息傳送應用程式。如需每個度量訊息類型中所報告之度量數量詳細資訊，請參閱「Message Queue Developer's Guide for Java Clients」。

表 4-2 度量主題目標

主題名稱	度量資訊的類型
<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.queueName</code>	特定佇列的目標度量
<code>mq.metrics.destination.topic.topicName</code>	特定主題的目標度量

代理程式特性 `mq.metrics.topic.enabled` 與 `mq.metrics.topic.interval`，可分別控制訊息是否要傳送到度量主題目標及其頻率。

`mq.metrics.topic.timetolive` 與 `mq.metrics.topic.persist` 特性可指定此類訊息的有效期及其是否具有永久性。

除了度量訊息內文包含的資訊外，每個訊息標頭還包括提供下列資訊的特性：

- 訊息類型
- 傳送訊息之代理程式的位址 (主機名稱與連接埠號碼)
- 度量範例需要的時間

這些特性對處理不同類型或不同代理程式的度量訊息之用戶端應用程式十分有用。

設定代理程式特性

您可以使用下列兩種方法之一來指定代理程式的配置特性：

- 編輯代理程式的配置檔案
- 直接從指令行提供特性值

以下兩節將說明這兩種配置代理程式的方法。

配置檔案

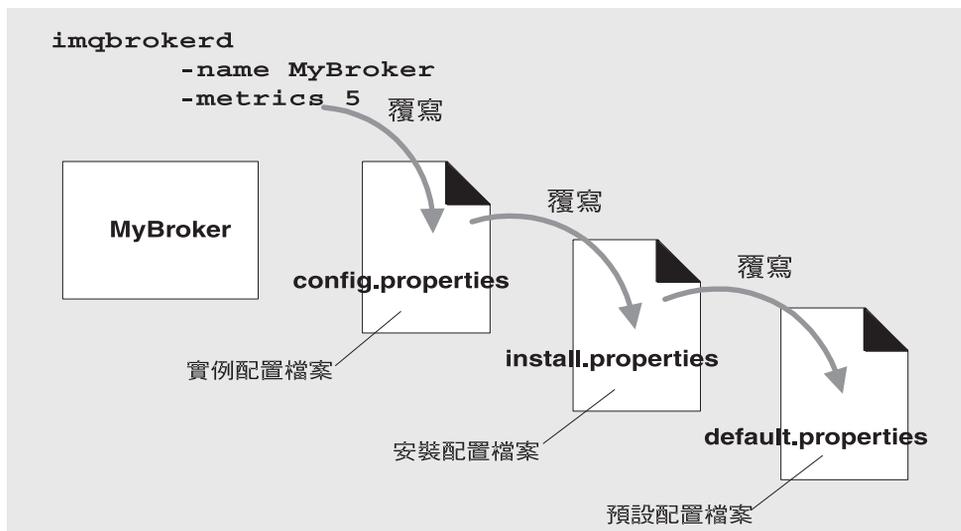
代理程式配置檔案含有配置代理程式所需的特性設定。這些設定會根據您所使用的作業系統平台存放在適當的目錄中；請參閱附錄 A 「[Message Queue 資料的平台特定位置](#)」，以瞭解詳細資訊。此目錄儲存以下檔案：

- **預設配置檔案** `default.properties`，在啟動時載入。此檔案不可編輯，但您可以閱讀此檔案以確定預設值，並找到要變更之特性的確切名稱。
- **安裝配置檔案** `install.properties`，包含安裝 Message Queue 時指定的所有屬性。此檔案在安裝後無法編輯。

此外，各個代理程式實例也有其本身的**實例配置檔案**，如下所述。如果您連線叢集中的代理程式實例，可能還需要使用**叢集配置檔案**以指定叢集配置資訊；如需相關資訊，請參閱第 283 頁的「叢集配置特性」。

啟動時，代理程式會合併不同配置檔案中的特性值。如圖 4-4 所示，這些檔案會形成一個階層，實例配置檔案的指定值會在其中覆寫安裝配置檔案中的指定值，然後再覆寫預設配置檔案中的指定值。在階層頂端，您可以手動覆寫任何在配置檔案中指定的特性值，只要使用 `imqbrokerd` 指令的指令行選項即可。

圖 4-4 代理程式配置檔案



編輯實例配置檔案

首次執行代理程式時，會建立一個含有該代理程式實例之配置特性的實例配置檔案。此實例配置檔案名為 `config.properties`，儲存在以其所屬之代理程式實例名稱所識別的目錄中：

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

(請參閱附錄 A 「[Message Queue 資料的平台特定位置](#)」，以瞭解 instances 目錄的位置。) 若該檔案不存在，您必須在啟動代理程式時使用 `-name` 選項 (請參閱第 249 頁的「[代理程式公用程式](#)」)，指定可供 Message Queue 建立檔案的實例名稱。

注意 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啟動。

實例配置檔案由代理程式實例所維護，並會在您使用 Message Queue 管理公用程式變更配置時進行修改。您也可以手動編輯實例配置檔案，以自訂代理程式的運作方式與資源使用。若要執行此動作，您必須是 instances/*instanceName* 目錄的擁有者，或以 root 的身份登入，以變更目錄的存取權限。

代理程式只能在啟動時讀取其實例配置檔案。若要對代理程式的配置進行永久性的變更，必須先關閉代理程式並編輯檔案，然後再重新啟動代理程式。該檔案 (或任何配置檔案) 中的特性定義會使用下列語法：

```
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存放區中最多保留 50,000 個訊息。

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

以下項目指定每天 (86,400 秒) 會建立一個新的記錄檔：

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

請參閱第 74 頁的「[代理程式服務](#)」與第 14 章「[代理程式特性參照](#)」，以瞭解可用代理程式配置特性及其預設值的相關資訊。

從指令行設定配置選項

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，可以使用代理程式公用程式 `imqbrokerd` 來啟動代理程式實例。指令的 `-D` 選項，可用以指定任何代理程式配置特性及其值；如需相關資訊，請參閱第 67 頁的「[啟動代理程式](#)」與第 249 頁的「[代理程式公用程式](#)」。如果您使用服務管理員公用程式 (`imqsvcadmin`) 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；請參閱第 263 頁的「[服務管理員公用程式](#)」。

執行代理程式實例時，您也可以變更某些代理程式特性。若要修改執行中代理程式的配置，可以使用指令公用程式的 `imqcmd update bkr` 指令；請參閱第 98 頁的「更新代理程式特性」與第 254 頁的「代理程式管理」。

配置永久資料存放區

代理程式的永久資料存放區中含有實體目標、長期訂閱、訊息、作業事件與確認等項目的相關資訊。Message Queue 代理程式預設會使用檔案型永久存放區，但您可以重新配置，以插入任何可透過 JDBC 相容驅動程式加以存取的資料存放區。代理程式配置特性 `imq.persist.store` (請參閱第 271 頁的表 14-4) 可指定兩種形式的可用永久性。

本節將說明如何設定代理程式使用永久存放區。它包括以下主題：

- 第 89 頁的「配置檔案型存放區」
- 第 90 頁的「配置 JDBC 型存放區」
- 第 91 頁的「保護永久性資料」

配置檔案型存放區

建立代理程式實例時，會自動建立檔案型資料存放區。此存放區位於代理程式的實例目錄中；請參閱附錄 A「Message Queue 資料的平台特定位置」，以瞭解確切位置。

依預設，Message Queue 會非同步寫入作業至磁碟。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。若要增強可靠性 (必須犧牲效能)，您可以設定代理程式特性 `imq.persist.file.sync`，改為以同步方式以寫入資料。如需此特性的詳細討論，請參閱第 78 頁的「檔案型永久性」與第 272 頁的表 14-5。

啟動代理程式實例時，可以使用 `imqbrokerd` 指令的 `-reset` 選項來清除檔案系統存放區。如需此選項及其子選項的更多資訊，請參閱第 249 頁的「代理程式公用程式」。

配置 JDBC 型存放區

若要配置代理程式使用 JDBC 型永久性，請在代理程式實例配置檔案中設定與 JDBC 相關的特性，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。也可以使用資料庫管理員從資料庫中刪除這些毀壞的表格，或使用其他資料庫作為資料存放區。請參閱第 261 頁的「資料庫管理員公用程式」，以獲得更多資訊。

備註 Oracle 和 PointBase 資料庫產品有配置範例可供使用。這些檔案的位置視平台而異，在附錄 A 「Message Queue 資料的平台特定位置」相關表格的「範例應用程式與配置」中會有列出。此外，PointBase 內嵌版本、PointBase 伺服器版本與 Oracle 的範例，會在實例配置檔案 config.properties 中以註解值的形式提供。

► 配置 JDBC 型資料存放區

1. 在代理程式的配置檔案中設定與 JDBC 相關的特性。

相關特性在第 80 頁的「JDBC 型永久性」中討論，並在第 273 頁的表 14-6 中列出。特別是，您必須將代理程式的 imq.persist.store 特性設為 jdbc (請參閱第 271 頁的表 14-4)。

2. 將 JDBC 驅動程式 .jar 檔案的副本或符號連結放入下列位置：

```
/usr/share/lib/imq/ext/ (Solaris)
/opt/sun/mq/share/lib/ (Linux)
IMQ_VARHOME\lib\ext (Windows)
```

例如，若您要在 Solaris 系統上使用 PointBase，下列指令就會將驅動程式的 .jar 檔案複製到適當的位置上：

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

下列指令則會建立符號連結：

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

3. 建立 Message Queue 永久性所需的資料庫模式。

使用 imqdbmgr create all 指令 (針對內嵌式資料庫) 或 imqdbmgr create tbl 指令 (針對外部資料庫)；請參閱第 261 頁的「資料庫管理員公用程式」。

- a. 變更為 imqdbmgr 所在的目錄：

```
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

若使用內嵌式資料庫，最好在下列目錄中建立：

```
.../instances/instanceName/dbstore/databaseName
```

如果內嵌式資料庫未由使用者名稱和密碼保護，則可能由檔案系統權限所保護。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。

保護永久性資料

永久存放區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，請務必保護資料存放區，以防止未經授權的存取。本節將說明如何確保檔案型或 JDBC 型之資料存放區的資料安全。

保護檔案型存放區

使用檔案型永久性的代理程式，會將永久性資料寫入平面檔案資料存放區中，而此存放區的位置會隨平台而有所不同（請參閱附錄 A 「[Message Queue 資料的平台特定位置](#)」）：

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

`instanceName` 是辨別代理程式實例的名稱。

`instanceName/fs350/` 目錄會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執行代理程式所在的作業系統平台：

- 在 Solaris 與 Linux 上，目錄的權限取決於啟動代理程式實例之使用者的檔案模式建立遮罩 (umask)。因此，透過適當地設定遮罩，即可限制啟動代理程式實例和讀取其永久性檔案的權限。此外，管理員（超級使用者）可以透過將 `instances` 目錄的權限設定為 700，保護永久性資料。
- 在 Windows 上，目錄的權限可透過 Windows 作業系統所提供的機制來設定。這通常需要開啓目錄的 [內容] 對話方塊。

保護 JDBC 型存放區

使用 JDBC 型永久性的代理程式，會將永久性資料寫入與 JDBC 相容的資料庫中。對於由資料庫伺服器所管理的資料庫 (例如 Oracle)，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。請參閱資料庫供應商提供的文件，以取得如何建立使用者名稱 / 密碼存取的相關資訊。

代理程式開啓資料庫連線所需的使用者名稱和密碼，可以提供為代理程式配置特性。但若在啓動代理程式時將其提供作為指令行選項，則有更高的安全性，方法是使用 imqbrokerd 指令的 -dbuser 與 -dbpassword 選項 (請參閱第 249 頁的「代理程式公用程式」)。

對於代理程式透過資料庫的 JDBC 驅動程式直接存取的內嵌式資料庫，通常是透過設定檔案在儲存永久性資料的目錄上的權限來確保此資料庫的安全，「[保護檔案型存放區](#)」中會有詳細說明。但是為確保代理程式和資料庫管理員公用程式都能讀取和寫入該資料庫，兩者應該由同一位使用者執行。

管理代理程式

本章說明如何使用 `imqcmd` 公用程式來管理代理程式及其服務。本章包含下列各節：

- 第 94 頁的「必要條件」
- 第 95 頁的「使用 `imqcmd` 公用程式」
- 第 97 頁的「顯示代理程式資訊」
- 第 98 頁的「更新代理程式特性」
- 第 99 頁的「暫停和繼續代理程式」
- 第 100 頁的「關閉和重新啓動代理程式」
- 第 100 頁的「顯示代理程式度量」
- 第 101 頁的「管理連線服務」
- 第 106 頁的「取得有關連線的資訊」
- 第 107 頁的「管理長期訂閱」
- 第 108 頁的「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` 指令行公用程式來管理代理程式。管理代理程式前，請先執行以下作業：

- 使用 `imqbrokerd` 公用程式指令啓動代理程式。代理程式執行後，才能使用其他指令行公用程式。
- 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，同時也安裝了預設平面檔案使用者儲存庫。此儲存庫出廠時隨附兩個預設項目：管理使用者和 `guest` 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。如需設定檔案型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啓用 `ssladmin` 服務。如需更多資訊，請參閱第 140 頁的「[使用 SSL 型服務](#)」。

使用 imqcmd 公用程式

imqcmd 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 247 頁的第 13 章「指令行參照」。如需管理實體目標的詳細資訊，請參閱第 289 頁的第 15 章「實體目標特性參照」。

顯示說明

若要顯示 imqcmd 公用程式的說明，請使用 `-h` 或 `-H` 選項，且不可使用子指令。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或其他選項之外，還包含 `-h` 或 `-H` 選項，則 imqcmd 公用程式只會處理 `-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或其他選項之外，還包含 `-v` 選項，則 imqcmd 公用程式只會處理 `-v` 選項。忽略指令行的所有其他項目。

指定使册者名稱和密碼

每個 imqcmd 子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。只有使用 `-h` 或 `-H` 選項來顯示說明的指令以及使用 `-v` 選項來顯示產品版本的指令除外。

指定使册者名稱

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。例如，下列指令會顯示預設代理程式的相關資訊：

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

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。

舊版 **Message Queue** 可讓您在 `imqcmd` 指令行上使用 `-p` 選項來指定密碼。此選項已經被停用，將從未來的版本中移除。

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。

對遠端主機上執行的代理程式或非預設連接埠上偵聽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
```

下列範例會列出在使用連接埠 1564 的主機 `myserver` 上執行之代理程式的特性。使用者名稱是 `aladdin`。(若要讓此指令可以運作，需要更新使用者儲存庫，並將使用者名稱 `aladdin` 加入 `admin` 群組中。)

```
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。使用者密碼位於 `myPassfile` 密碼檔案中，該檔案位於呼叫指令時所在的目前目錄中。

```
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:portNumber
```

該子指令會列出預設代理程式或指定主機和連接埠上的代理程式之特性目前的設定。它還顯示連線至指定代理程式的執行中代理程式 (在多重代理程式叢集中) 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：

版本	3.6
實例名稱	imqbroker
主要連接埠	7676
系統中目前的訊息數目	0
系統中的目前訊息容量 (位元組)	0
停用訊息佇列中目前的訊息數目	0
停用訊息佇列中目前訊息的總位元組	0
記錄停用的訊息	true
截斷停用訊息佇列的訊息內文	false
系統中的最大訊息數目	無限制 (-1)
系統中的訊息最大容量	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

記錄層級

記錄自動重建的間隔 (以秒為單位)

記錄自動重建的大小 (位元組)

INFO

604800

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:portNumber] -o attribute=value [[-o attribute=value1] ...]
```

該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。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
```

[第 14 章「代理程式特性參照」](#)中描述了這些特性。

暫停和繼續代理程式

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、接收訊息或派送訊息。

但是，暫停代理程式不會暫停 `admin` 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。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所執行的傳送功能而有差異。因此，暫停叢集中的代理程式可能會造成某些訊息流量變慢。

`pause bkr` 子指令的語法如下：

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

該指令會暫停預設代理程式或指定主機和連接埠上的代理程式。

以下指令會暫停連接埠 1588 上 `myhost` 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 105 頁的「暫停和繼續連線服務」和第 118 頁的「暫停和繼續實體目標」。

繼續代理程式

繼續代理程式會重新啓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:portNumber]
```

該子指令會繼續預設代理程式或指定主機和連接埠上的代理程式。

以下指令會繼續連接埠 7676 上 `localhost` 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:portNumber]
```

該子指令會關閉預設代理程式或指定主機和連接埠上的代理程式。

以下指令會關閉連接埠 1572 上 ctrlsrv 執行的代理程式：

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

使用 restart bkr 子指令關閉和重新啟動代理程式。restart bkr 子指令的語法如下：

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

該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。

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:portNumber]  
                  [-m metricType] [-int interval] [-msp numSamples]
```

該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：

- **t11** 顯示傳入和傳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) 傳入和傳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，代理程式傳入與傳出的訊息速率，請使用：

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

此指令的輸出如下：

訊息 / 秒鐘		訊息容量 / 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

如需有關代理程式收集和報告之資料的詳細說明，請參閱第 310 頁的「代理程式通用度量」。

管理連線服務

imqcmd 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
- 暫停和繼續連線服務

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。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 (SSL 型安全性)

表 5-1 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 (SSL 型的安全性)
admin	ADMIN	tcp
ssladmin (Enterprise Edition)	ADMIN	tls (SSL 型的安全性)

您可以使用 `mqcmd` 子指令來管理整個連線服務，或管理個別連線服務。如果子指令的目標是個別服務，請使用 `-n` 選項，以表 5-1 之 [服務名稱] 欄位中所列的名稱之一進行指定。

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：

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

該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。

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：

```
mqcmd 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 `query` 子指令。

`query svc` 子指令的語法如下：

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

`query svc` 子指令會顯示在預設代理程式或指定主機和連接埠的代理程式上所執行的指定服務之相關資訊。

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
目前的連線數目	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 中所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
port	指定給要更新的服務之連接埠 (不適用於 httpjms 或 httpsjms)。0 值表示連接埠對映器會動態匹配連接埠。
minThreads	指定給服務的執行緒之最小數量。
maxThreads	指定給服務的最大執行緒數目。

`update` 子指令的語法如下：

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

該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。如需服務屬性的描述，請參閱第 266 頁的「連線特性」。

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:portNumber] [-m metricType]
    [-int interval] [-msp numSamples]
```

該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，傳入和傳出代理程式的訊息和封包的度量。(預設度量類型)。
- **rts** 顯示代理程式經由指定連線服務，(每秒) 傳入和傳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)。

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		訊息容量		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 312 頁的「[連線服務度量](#)」。

暫停和繼續連線服務

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:portNumber]
```

該子指令暫停在預設代理程式或指定主機和連接埠上代理程式執行的指定服務。例如，下列指令會暫停預設代理程式上所執行的 `httpjms` 服務。

```
imqcmd pause svc -n httpjms -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` 方法會阻斷此訊息。)
- 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:portNumber]
```

該子指令會繼續在預設代理程式或指定主機和連接埠上代理程式執行的指定服務。

取得有關連線的資訊

imqcmd 公用程式包括可讓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:portNumber]
```

該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，下列指令會列出預設代理程式上的所有連線：

```
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:portNumber]
```

該子指令會顯示在預設代理程式或指定主機和連接埠之代理程式上，指定連線的相關資訊。

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

您可以執行下列一項或多項作業，使用 `imqcmd` 子指令來管理代理程式的長期訂閱：

- 列出長期訂閱
- 清除長期訂閱的所有訊息
- 銷毀長期訂閱

*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，代理程式僅在為持久訂閱者保留的訊息過期時，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。

`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 clientID
```

`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 clientID
```

例如，下列指令會銷毀長期訂閱 `myDurable` 和 `clientID myClientID`。

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

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) 管理員所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，您可能需要監視作業事件，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 txn` 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：

事務處理 ID	狀態	使用者名稱	訊息數目 / 回應數目	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 267 頁的表 14-2)，則您必須在啟動代理程式時手動確定或回轉處於 `PREPARED` 狀態的作業事件。

`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`) 中回應的訊息數目。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 transactionID
```

例如，下列範例的輸出內容應該如下：

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

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

使用 `commit` 和 `rollback` 子指令，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 transactionID
```

例如：

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

`rollback` 子指令的語法如下：

```
imqcmd rollback txn -n transactionID
```

請參閱第 267 頁的表 14-2 中的 `imq.transaction.autorollback` 特性，以取得更多資訊。

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

管理實體目標

本章說明如何使用 `imqcmd` 公用程式，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，可以管理訊息的永久性和確認。

本章涵蓋下列主題：

- [第 113 頁的「使用指令公用程式」](#)
- [第 114 頁的「建立實體目標」](#)
- [第 115 頁的「列出實體目標」](#)
- [第 116 頁的「顯示實體目標資訊」](#)
- [第 117 頁的「更新實體目標特性」](#)
- [第 118 頁的「暫停和繼續實體目標」](#)
- [第 119 頁的「清除實體目標」](#)
- [第 119 頁的「銷毀實體目標」](#)
- [第 120 頁的「壓縮實體目標」](#)
- [第 122 頁的「配置使用停用訊息佇列」](#)

[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。

如需實體目標的簡介，請參閱「[Message Queue 技術摘要](#)」。

備註

用戶端應用程式與實體目標互動時，會使用 Destination 物件。爲了提供者獨立性和可移植性，用戶端通常使用管理員建立的目標物件，即目標管理物件。您可以如第 8 章「[管理受管理物件](#)」中所描述，配置供用戶端應用程式使用的管理物件。

使用指令公用程式

使用 Message Queue 指令公用程式 (imqcmd) 管理實體目標。imqcmd 指令的語法，與您將其用於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 247 頁的第 13 章「指令行參照」。

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 256 頁的「實體目標管理」。

表 6-1 指令公用程式的實體目標子指令

子指令與代數	描述
compact dst	壓縮於一個或多個實體目標的檔案型資料存放區。
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並銷毀實體目標。
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。

很多實體目標特性都會影響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。這些限制類似於代理程式配置特性所控制的一切代理程式限制。

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執行緒時，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`。指定單一批次中，傳送給用戶的最大訊息數目。
- `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 (連線至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訊息佇列中。

下列特性僅用於佇列目標：

- `maxNumActiveConsumers`。指定從佇列目標之負載平衡傳送中可成爲使用中用戶的最大數目。
- `maxNumBackupConsumers`。指定最大備份用戶數目，如果無法從佇列目標進行負載平衡發送，則這些用戶可以取代使用中用戶。
- `localDeliveryPreferred`。僅適用於代理程式叢集中的負載平衡佇列發送。如果本機代理程式沒有用戶，則指定訊息僅發送給遠端用戶。

如需實體目標特性的完整參照資訊，請參閱第 289 頁的第 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 269 頁的表 14-3 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：

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

- 若要建立主題目標，請輸入以下指令：

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

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。

若要尋找欲取得其相關資訊的實體目標，請使用 `list dst` 子指令，列出代理程式上的所有實體目標。`list dst` 子指令的語法如下：

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

該指令可以列出所指定類型的實體目標。目標類型值 (-t) 的選項可爲 `q` (佇列) 或 `t` (主題)。

如果省略目標類型，則會列出所有類型的實體目標。

`list dst` 子指令可選擇性地指定要列出的目標類型，或納入暫存目標 (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，若要取得連接埠 4545 上 myHost 執行的代理程式的所有實體目標，請輸入以下指令：

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

除非您指定目標類型 *t* 只包含主題，否則除了其他實體目標，也一定會顯示停用訊息佇列 `mq.sys.dmq` 的資訊。

顯示實體目標資訊

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

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

該指令列出指定類型和名稱之目標的相關資訊。例如，下列指令會顯示佇列目標 XQueue 的相關資訊：

```
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
目前訊息容量 (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)
訊息最大容量 (位元組)	無限制 (-1)
每條訊息的最大容量 (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
限制行爲	REJECT_NEWEST
用戶流量限制	1000
是本機目標	false
偏好本機發送	false
使用停用訊息佇列 Queue	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` 子指令來變更一個或多個特性值 (請參閱第 117 頁的「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]...]
```

該指令會更新指定目標上的指定特性的值。特性名稱可以是表 15-1 中所列出的任何特性。

您可以多次使用 `-o` 選項，更新多個特性。例如，以下指令將 `maxBytesPerMsg` 特性變更爲 1000，並將 `MaxNumMsgs` 特性變更爲 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 122 頁的「配置使用停用訊息佇列」。

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，您可以暫停進入目標的訊息流量，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。壓縮實體目標之前，必須先予以暫停。

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，該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]
```

該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，則會繼續所有目標。

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。必須予以個別暫停。

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。清除實體目標即表示會刪除儲存於該目標的所有訊息。

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。當佇列沒有註冊的用戶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用戶，也可能需要清除訊息。在這兩種情況下，沒有比必要保留訊息。

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
```

此子指令會清除實體目標上指定類型與名稱的訊息。

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
```

此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。

您不能銷毀停用訊息佇列。

壓縮實體目標

如果您使用檔案型資料存放區作為訊息的永久存放區，即可於需要時監視磁碟使用情況與壓縮磁碟。

檔案型訊息存放區的結構方式，是讓訊息可以儲存在與收納訊息之實體目標相對應的目錄中。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收回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：

已保留	已使用	使用率
806400	804096	99
1793024	1793024	100
2544640	2518272	98

子指令輸出中的欄有下列涵義：

表 6-2 實體目標磁碟使用度量

度量	說明
已保留	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。
已使用	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會用來保留使用中訊息。

收回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。根據傳入和傳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，則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，則使用率會比較低。

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)，則就沒有需要收回未使用的磁碟空間。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，收回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]
```

此子指令會壓縮指定類型與名稱之實體目標的檔案型資料存放區。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 289 頁的表 15-1)。

► 收回未使用的實體目標磁碟空間

1. 暫停目標。

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

2. 壓縮磁碟。

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

3. 繼續實體目標。

```
imqcmd 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。

配置使用停用訊息佇列

停用訊息佇列 `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「停用訊息」一詞的定義和停用訊息佇列的詳細資訊，請參閱「[Message Queue 技術摘要](#)」。

代理程式啟動時會自動建立停用訊息佇列。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訊息佇列來保留已捨棄的訊息。使用停用訊息佇列提供的資訊，有助於系統的疑難排解。

配置使用停用訊息佇列

依預設，系統會配置實體目標以使用停用訊息佇列。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訊息佇列。

下列範例建立的佇列 `myDist`，依預設使用停用訊息佇列：

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

下列範例讓相同的佇列停用此停用訊息佇列：

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

設定 `imq.autocreate.destination.useDMQ` 代理程式特性，可以讓代理程式上所有自動建立的實體目標使用停用訊息佇列，或不讓其使用停用訊息佇列。

配置和管理停用訊息佇列

您可使用 `Message Queue` 指令公用程式 (`imqcmd`) 管理停用訊息佇列，方式與管理其他佇列僅有些微差異。例如，因為停用訊息佇列是系統所建立的，因此您不能建立、暫停或銷毀它。此外，如表 6-3 中所示，停用訊息佇列之預設值，有時會和正常佇列的預設值不相同。

停用訊息佇列特性

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 列出停用訊息佇列另以特殊方式處理的佇列特性。

表 6-3 停用訊息佇列處理標準實體目標特性的方式

特性	停用訊息佇列的處理方式
limitBehavior	停用訊息佇列之預設值是 REMOVE_OLDEST。(其他佇列的預設值是 REJECT_NEWEST。)停用訊息佇列上不支援流量控制。
localDeliveryPreferred	不套用於停用訊息佇列。
maxNumMsgs	停用訊息佇列之預設值是 1000。其他佇列的預設值是 -1 (無限制)。
maxNumProducers	不套用於停用訊息佇列。
maxTotalMsgBytes	停用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訊息佇列總是本機實體目標，其特性永遠設定為 true。但是，本機代理程式的停用訊息佇列可以包含叢集中其他代理程式的客戶端產生的訊息，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訊息佇列中，或是捨棄訊息內文的內容，只保留標頭和特性資料。依預設，停用訊息佇列儲存整個訊息。

若想要降低停用訊息佇列大小，且不打算復原停用的訊息，請考慮將 `imq.destination.DMQ.truncateBody` 代理程式特性設為 true：

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

如此將會捨棄訊息本文，並只保留標頭與特性資料。

啟用停用訊息記錄功能

依預設，不記錄停用的訊息。啟用停用訊息的記錄，可讓代理程式記錄下列事件：

- 代理程式會將訊息移至停用訊息佇列
- 此代理程式會捨棄來自停用訊息佇列的訊息，以及不使用停用訊息佇列之實體目標的訊息

- 實體目標達其上限

下列指令會啟用停用訊息的記錄：

```
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

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

本章包含下列各節：

- [第 126 頁的「認證使用者」](#)
- [第 134 頁的「授權使用者：存取控制特性檔案」](#)
- [第 140 頁的「使用 SSL 型服務」](#)
- [第 149 頁的「使用密碼檔案」](#)
- [第 151 頁的「建立稽核記錄」](#)

認證使用者

當使用者嘗試連線至代理程式時，此代理程式將透過檢查提供的名稱和密碼，認證此使用者。如果名稱和密碼與配置每個代理程式要參考的特定代理程式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。

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。

儲存庫可能是下列任一類型：

- **Message Queue 隨附的平面檔案儲存庫**

這種類型的使用者儲存庫使用非常容易。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。若要啓用認證，您應在使用者儲存庫中寫入每個使用者名稱、密碼以及使用者群組名稱。

如需有關設定與管理使用者儲存庫的更多資訊，請參閱「[使用平面檔案使用者儲存庫](#)」。

- **LDAP 伺服器**

可以是使用 LDAP v2 或 v3 協定的現有或新的 LDAP 目錄伺服器。它不如平面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，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 132 頁的「[將 LDAP 伺服器用於使用者儲存庫](#)」。

使用平面檔案使用者儲存庫

Message Queue 提供了平面檔案使用者儲存庫和指令行工具，即使用者管理員公用程式 (`imqusermgr`)，您可以使用此工具寫入和管理平面檔案使用者儲存庫。以下各節描述平面檔案使用者儲存庫，以及您如何使用使用者管理員公用程式寫入和管理儲存庫。

建立使用者儲存庫

平面檔案使用者儲存庫是針對實例的。您啟動的每個代理程式實例，都會自動建立預設的使用者儲存庫 (稱為 passwd)。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 資料的平台特定位置」)：

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

此儲存庫建立有兩個項目：表 7-1 的每一行表示一個項目。

表 7-1 使用者儲存庫中的初始項目

使用者名稱	密碼	權限	狀態
admin	admin	admin	使用中
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，而無需管理員的介入：

- 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，連線至代理程式實例。
- 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以變更密碼 (請參閱第 131 頁的「變更預設管理員密碼」)。

以下各節介紹如何寫入和管理平面檔案使用者儲存庫。

使用者管理員公用程式

Message Queue 使用者管理員公用程式 (imqusermgr) 可讓您編輯或寫入平面檔案使用者儲存庫。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「指令行參照」。

使用使用者管理員前，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。
- 必須在安裝代理程式的主機上執行 imqusermgr 指令。

- 您必須擁有寫入儲存庫的適當權限：也就是說，在 Solaris 和 Linux 上，您必須是超級使用者或是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 子指令可以更新指定的 (或預設的) 代理程式實例儲存庫中，指定使用者的密碼和 / 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>activeState</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	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。
-i <i>instanceName</i>	指定要執行指令的代理程式實例名稱。如果未指定，則假設為預設實例名稱 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。
-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。應如下所示指定群組：

- **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)。此帳號與大多數 FTP 伺服器中存在的匿名帳號類似。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)，或者應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 134 頁的「[授權使用者：存取控制特性檔案](#)」。

使用者狀態

依預設，新增使用者至儲存庫時，使用者的狀態預設為使用中。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。您必須刪除非使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 (:)、新行或換行字元。
- 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。

登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

為了安全性起見，您應將 `admin` 的預設密碼變更爲只有您自己知道的密碼。以下指令將 `mybroker` 代理程式實例的預設管理員密碼 `admin` 變更爲 `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 伺服器用於使用者儲存庫

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。當使用者嘗試連線至代理程式實例或執行訊息作業時，這些特性可讓代理程式實例查詢 LDAP 伺服器，以取得使用者和群組的相關資訊。

實例配置檔案位於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 資料的平台特定位置」。

► 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 64 基本編碼 (basic) 還是以 MD5 編碼 (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。第 81 頁的「安全性服務」中討論了這些特性，並摘要於第 276 頁的「安全性特性」。

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

Message Queue 支援 LDAP 認證容錯移轉：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 134 頁的「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`。

使用多台 LDAP 目錄伺服器時，請使用 `ldap://` 指定其他目錄伺服器。例如：

```
imq.user_repository.ldap.server=myHost:7878 ldap://otherHost:7878
...
```

以空格分隔每一台目錄伺服器。清單中的所有目錄伺服器必須使用相同的值，以供其他與 LDAP 相關的特性使用。

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 資料的平台特定位置](#)」中會列出此檔案的位置。

此檔案包含一個項目，如下所示：

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

列出的項目僅為範例。請注意檔案型使用者儲存庫中有 `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，替換為要授與 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。

無論使用者資訊放在平面檔案使用者儲存庫中還是放在 LDAP 使用者儲存庫中，均可使用 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

- 建立產生者
- 建立用戶
- 瀏覽佇列

代理程式會檢查 ACL 檔案，判斷提出請求的使用者或該使用者所屬的群組是否已有授權執行此作業。

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

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 資料的平台特定位置](#)」)：

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

ACL 檔案的格式與 Java 特性檔案的格式類似。先是定義檔案的版本，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權限，之後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
<code>resourceType</code>	為以下之一：connection、queue 或 topic。
<code>resourceVariant</code>	<code>resourceType</code> 指定的類型實例。例如，myQueue。星號 (*) 可用於表示所有連線服務類型或所有實體目標。
<code>operation</code>	值取決於要制定的存取規則種類。
<code>access</code>	為以下之一：allow 或 deny。
<code>principalType</code>	為以下之一：user 或 group。如需更多資訊，請參閱第 129 頁的「群組」。
<code>principals</code>	可以擁存在規則左側指定的存取權的使用者。如果 <code>principalType</code> 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 <code>principalType</code> 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）。星號 (*) 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。如需更多詳細資訊，請至

<http://java.sun.com/j2se/1.4/docs/guide/intl/faq.html>

如何計算權限

檔案中有多個存取規則時，權限計算如下：

- 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* 的存取權會覆寫賦予 ** 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向 `tq1` 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** 的所有其他成員皆可執行這項作業。


```
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權限，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權限會相互抵銷。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
```

如果您使用檔案型的使用者儲存庫，使用者管理員公用程式會建立預設群組 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 目錄中定義的一個或多個群組名稱。

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，與檔案型的使用者儲存庫和 LDAP 使用者儲存庫有下列不同：

- **檔案型使用者儲存庫。**
 - 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，包含實體目標型的存取控制規則。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、將訊息發佈至主題、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。您可以新增更多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

元素	說明
<i>resourceType</i>	可能是 queue 或 topic。
<i>resourceVariant</i>	實體目標名稱或所在實體目標 (*)，表示所在佇列或所在主題。
<i>operation</i>	可能是 produce、consume 或 browse。
<i>access</i>	可能是 allow 或 deny。
<i>principalType</i>	可能是 user 或 group。

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。

當使用者在並不存在的實體目標上建立產生者或用戶時，如果已啟用代理程式的自動建立特性，則代理程式將會建立此目標。

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 SSL 型服務

安全套接層 (SSL) 型標準的連線服務，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 SSL 型連線服務。

Message Queue 支援安全套接層 (SSL) 標準型的下列連線服務：

- `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 支援自身簽名伺服器憑證或簽名憑證的 SSL 型加密。

若要使用 SSL 型連線服務，您將使用金鑰工具公用程式 (imqkeytool) 產生私密金鑰 / 公開金鑰對。此公用程式將公開金鑰內嵌於自身簽名的憑證中，此憑證將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 SSL 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 之 SSL 型連線服務 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，請參閱第 327 頁的附錄 C 「HTTP/HTTPS 支援」。

TCP/IP 的安全連線服務

下列 SSL 型連線服務可提供透過 TCP/IP 的直接安全的連線：

- ssljms 服務透過安全加密的連線在用戶端與代理程式之間傳送訊息。
- ssladmin 服務會在 Message Queue 指令公用程式 (imqcmd) 和代理程式之間建立安全加密的連線。管理主控台 (imqadmin) 不支援安全連線。
- cluster 服務透過安全的加密連線在叢集中的代理程式之間傳送訊息，並且提供代理程式之間的通訊 (請參閱第 174 頁的「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 SSL 型服務。

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，然後再至第 146 頁的「配置使用簽名的憑證」執行其他步驟。

► 設定 SSL 型連線服務

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 (僅適用於 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

具有自身簽名憑證的 Message Queue SSL 支援會認為用戶端正與已知並可信任的伺服器通訊，而保護線上資料。

執行金鑰工具公用程式，以便為代理程式產生自身簽名的憑證。在 UNIX® 系統上，若要擁有建立金鑰存放區的權限，您可能需要以超級使用者 (root) 的身份執行金鑰工具。

相同的憑證可用於 ssljms、ssladmin 或 cluster 連線服務。

在指令提示符號下輸入以下內容：

```
imqkeytool -broker
```

公用程式會提示您輸入金鑰存放區密碼。

```
代理程式正在產生金鑰存放區 ...
輸入金鑰存放區密碼：
```

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 自身簽名憑證所需之識別名稱資訊

提示	證書	範例
您的姓名是什麼？	X.500 commonName (CN)。請輸入執行代理程式的伺服器完全合格的名稱。	myhost.sun.com
您的組織單位名稱是什麼？	X.500 organizationalUnit (OU)。請輸入部門名稱。	purchasing
您的組織名稱是什麼？	X.500 organizationName (ON)。較大的組織名稱，例如公司或政府實體。	My Company, Inc.
您的城市或所在地名稱是什麼？	X.500 localityName (L)。	San Francisco
您的州或省名稱是什麼？	X.500 stateName (ST)。請輸入州或省的名稱，不使用縮寫。	California
表示此單位的雙字母代碼是什麼？	X.500 country (C)。	US

輸入完畢後，金鑰工具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，金鑰工具會在產生成對的金鑰時暫停。

接著，金鑰工具會要求您提供密碼，以鎖定特定的成對金鑰（金鑰密碼）。請對此提示按 Return 鍵，表示金鑰密碼和金鑰存放區密碼相同。

備註 請記住您提供的密碼。您啟動代理程式時必須提供此密碼，代理程式才能開啓金鑰存放區。您可以在密碼檔案中儲存金鑰存放區（請參閱第 149 頁的「使用密碼檔案」）。

執行 `imqkeytool` 指令，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 金鑰存放區中，且如附錄 A「Message Queue 資料的平台特定位置」中所示，其目錄位置會因作業系統而異。

此金鑰存放區的格式與 JDK1.2 `keytool` 公用程式支援的格式相同。

這些是 Message Queue 金鑰存放區的可配置特性：

- `imq.keystore.file.dirpath`。用於 SSL 型服務：指定進入包含金鑰存放區檔案之目錄的路徑。如需預設值，請參閱附錄 A「Message Queue 資料的平台特定位置」。
- `imq.keystore.file.name`。用於 SSL 型服務：指定金鑰存放區檔案的名稱。
- `imq.keystore.password`。用於 SSL 型服務：指定金鑰存放區密碼。

您可能需要重新產生成對金鑰，以解決某些問題，例如：

- 您忘記了金鑰存放區密碼。
- 啟動代理程式時，SSL 型服務無法初始化，並且您遇到以下異常：
`java.security.UnrecoverableKeyException` 無法回復金鑰。

可能導致此異常的原因是，所提供的金鑰密碼與您在第 142 頁的「步驟 1：產生自身簽名的憑證」中所產生之自身簽名憑證時的金鑰存放區密碼不同。

➤ 重新產生金鑰對

1. 移除代理程式的金鑰存放區，如附錄 A「Message Queue 資料的平台特定位置」中所示。

2. 重新執行 `imqkeytool` 以產生金鑰對，如第 142 頁的「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 SSL 型服務

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

注意 請使用 `imq.cluster.transport` 特性啓用 SSL 型 cluster (叢集) 連線服務，而非使用 `imq.service.activelist` 特性。請參閱第 174 頁的「代理程式之間的安全連線」。

► 啓用代理程式中 SSL 型的服務

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 資料的平台特定位置」)：

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

2. 新增表示 `imq.service.activelist` 特性的項目 (如果項目不存在)，並將 SSL 型服務納入清單。

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，或同是增加兩者 (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，提供金鑰存放區密碼。您可以用以下任何一種方法提供密碼：

- 允許代理程式啓動時提示您提供密碼：

```
imqbrokerd
請輸入金鑰存放區密碼：myPassword
```

- 將密碼放在密碼檔案中，如第 149 頁的「使用密碼檔案」中所述。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作業：

- 將密碼檔案的位置傳送給 `imqbrokerd` 指令：

```
imqbrokerd -passfile /tmp/myPassfile
```

- 啟動不具有 `-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) 以實施 SSL。JSSE 使用 `java.security.SecureRandom` 產生隨機數字。此方法建立初始隨機數種子要佔用大量時間，這就是您所看到 CPU 使用率增加的原因。建立種子之後，CPU 使用率會降至正常。

步驟 4：配置並執行 SSL 型用戶端

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 時加入 `-secure` 選項，建立安全管理連線。例如：

```
imqcmd list svc -b hostName:portNumber -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。(如果您是使用平面檔案儲存庫，請參閱第 131 頁的「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，如下所示：

列出指定代理程式上的所有服務：

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，請在金鑰存放區中安裝簽名的憑證，然後配置 Message Queue 用戶端，這樣在建立與 imqbrokerd 的 SSL 連線時會需要簽名的憑證。

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。

以下指示是假設您已完成在第 141 頁的「配置使用自身簽名的憑證」中說明的步驟。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

► 取得簽名的憑證

1. 使用 J2SE keytool，為您剛產生的自身簽名憑證，產生憑證簽名請求 (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，請將其儲存在檔案中。

這些指示使用 broker.cer 範例名稱來代表代理程式憑證。

► 安裝簽名的憑證

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

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

此指令會列出系統金鑰存放區中的根 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 金鑰存放區中。

範例：

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

ca.cer 值是從 CA 獲得的 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。

- 將簽名的憑證匯入金鑰存放區中，取代原始的自身簽名憑證。

例如：

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

broker.cer 值是您從 CA 獲得的簽名憑證所在的檔案。

現在 imqbrokerd 金鑰存放區具有簽名憑證，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。

- 若要配置用戶端向 imqbrokerd 請求有效的簽名憑證，請將用戶端 ConnectionFactory 物件的 imqSSLIsHostTrusted 屬性設定為 false。
- 請如第 145 頁的「步驟 4：配置並執行 SSL 型用戶端」所述，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。

- 請如下面幾節所述，在用戶端的信任存放區中安裝簽名 CA 根憑證。

配置有信任存放區的用戶端時，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。建議您使用此選項。
- 將根 CA 安裝到任何金鑰存放區檔案中，並且配置用戶端將此視為信任存放區。

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

此範例將根 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，用戶端會搜尋此金鑰存放區，因此不需要進一步配置用戶端。

裝到 `jssecacerts` 中

此範例將根 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，用戶端會搜尋此金鑰存放區，因此不需要進一步配置用戶端。

裝到其他檔案中

此範例將根 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，用戶端不會搜尋此金鑰存放區，因此您必須為用戶端提供信任存放區的位置。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>	啟動代理程式	存取 JDBC 型永久資料存放區、SSL 憑證金鑰存放區或 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 JDBC 型資料存放區	存取資料存放區

您可以在密碼檔案中指定這些密碼，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
```

注意 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。在目前的版本中，指令行中這些選項的任一值，會取代密碼檔案中相關的值。

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，您必須設定權限以防止無權限者存取此檔案。設定權限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 中所示密碼：

表 7-7 密碼檔案中的密碼

密碼	影響的指令	說明
imq.imqcmd.password	imqcmd	指定 imqcmd 指令行的管理員密碼。每個指令都會讀取此密碼。
imq.keystore.password	imqbrokerd	指定 SSL 型服務的金鑰存放區密碼。
imq.persist.jdbc.password	imqbrokerd imdbmgr	如非必要，則指定用於開啓資料庫連線的密碼。
imq.user_repository.ldap.password	imqbrokerd	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到已配置的 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 資料的平台特定位置」。

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
- 重設永久存放區
- 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`。

建立稽核記錄

管理受管理物件

管理物件會封裝提供者特定的配置和命名資訊，這樣能開發可以從 JMS 提供者移植到其他提供者的用戶端應用程式。Message Queue 管理員通常會為用戶端應用程式建立管理物件，使其能夠與代理程式建立連線，進行傳送與接收訊息。

本章說明如何使用物件管理員公用程式 (mqobjmgr) 來建立與管理受管理物件。包含下列各節：

- 第 154 頁的「物件存放區」
- 第 156 頁的「管理物件屬性」
- 第 162 頁的「使用物件管理員公用程式」
- 第 163 頁的「新增管理物件」
- 第 166 頁的「列出管理物件」
- 第 166 頁的「檢視管理物件資訊」
- 第 167 頁的「修改管理物件屬性」

物件存放區

管理物件會置於即時可用的物件存放區中，用戶端應用程式可透過 Java Naming and Directory Interface (JNDI) 存取此處的管理物件。您可以使用兩種類型的物件存放區：標準的簡易資料存取協定 (LDAP) 目錄伺服器或本機檔案系統的目錄。

LDAP 伺服器物件存放區

LDAP 伺服器是用於生產訊息傳送系統的建議物件存放區。LDAP 伺服器是專為在分散式系統中使用而設計的，並提供在生產環境中非常有用的安全性功能。

有多家供應商提供 LDAP 實作。若要使用 Message Queue 管理工具來管理 LDAP 伺服器上的物件存放區，首先需要將伺服器配置成可以儲存 Java 物件並執行 JNDI 查詢。請參閱 LDAP 實作所隨附的文件，以取得詳細資訊。

使用 LDAP 伺服器作為物件存放區時，您必須指定表 8-1 中所顯示的屬性。這些屬性分為以下幾個種類：

- **初始環境。** `java.naming.factory.initial` 屬性會指定伺服器上 JNDI 查詢的初始環境。此屬性的值會針對所指定的 LDAP 物件存放區進行修正。
- **位置。** `java.naming.provider.url` 屬性會指定 LDAP 伺服器的 URL 和目錄路徑。您必須確認所指定的目錄路徑存在。
- **安全性。** `java.naming.security.principal`、`java.naming.security.credentials` 和 `java.naming.security.authentication` 屬性，管理對嘗試存取物件存放區的呼叫者進行認證。這些屬性的實際格式和數值會視 LDAP 服務提供者而定；請參閱 LDAP 實作所隨附的文件，以取得詳細資訊，並判斷所有作業是否均需要安全性資訊，或者只有變更儲存資料的作業需要安全性資訊。

表 8-1 LDAP 物件存放區屬性

屬性	說明
<code>java.naming.factory.initial</code>	JNDI 查詢的初始環境 範例： <code>com.sun.jndi.ldap.LdapCtxFactory</code>
<code>java.naming.provider.url</code>	伺服器 URL 和目錄路徑 範例： <code>ldap://mydomain.com:389/ou=mqobjs,o=myapp</code> 其中管理物件存放區儲存在 <code>/myapp/mqobjs</code> 目錄中。

表 8-1 LDAP 物件存放區屬性 (續)

屬性	說明
<code>java.naming.security.principal</code>	<p>認證呼叫者的主體驗證身份</p> <p>此屬性的格式取決於認證機制：例如，<code>uid=homerSimpson,ou=People,o=mq</code></p> <p>如果沒有指定此屬性，則運作方式將由 LDAP 服務提供者決定。</p>
<code>java.naming.security.credentials</code>	<p>認證主體的憑證</p> <p>此屬性的值取決於認證方案：例如，它可以是雜湊密碼、純文字密碼、金鑰或憑證。</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>
<code>java.naming.security.authentication</code>	<p>認證的安全層級</p> <p>此屬性的值是下列關鍵字之一：none、simple 或 strong。例如，如果您指定 simple，則會提示您所選擇的主體或憑證值。這可讓您可以更完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>

檔案系統物件存放區

Message Queue 也支援在本機檔案系統中使用目錄，以作為管理物件的物件存放區。建議您對生產系統不要使用此方法，此方法在開發環境中具有使用簡單方便的優點。但是請注意，如果跨多個電腦節點部署的用戶端將目錄用作集中物件存放區，則這些用戶端都必須擁有該目錄的存取權。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。

使用檔案系統目錄作為物件存放區時，您必須指定表 8-2 中所顯示的屬性。這些屬性大致和上述 LDAP 物件存放區相同，特別是 `java.naming.provider.url` 屬性會指定保留物件存放區之目錄的路徑。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權限，以供存取存放區。

表 8-2 檔案系統物件存放區屬性

屬性	說明
<code>java.naming.factory.initial</code>	<p>JNDI 查詢的初始環境</p> <p>範例：</p> <p><code>com.sun.jndi.fscontext.RefFSContextFactory</code></p>

表 8-2 檔案系統物件存放區屬性 (續)

屬性	說明
java.naming.provider.url	目錄路徑 範例： file:///C:/myapp/mqobjs

管理物件屬性

Message Queue 管理物件有兩種基本類型：

- **連線工廠**：用戶端應用程式可用以建立與代理程式的連線。
- **目標**：表示代理程式上用戶端應用程式與其交換 (傳送和接收) 訊息的位置。

這兩種管理物件類型均包含決定物件特性及運作方式的屬性。本節描述如何使用物件管理員指令行公用程式 (imqobjmgr) 來設定這些屬性，您也可以使用 GUI 管理主控台進行設定，如第 2 章所述 (請參閱第 52 頁的「使用管理物件」)。

連線工廠屬性

用戶端應用程式使用**連線工廠**管理物件，建立與代理程式交換訊息的連線。連線工廠的屬性會定義由其建立的所有連線之特性。連線一旦建立後，就無法變更其特性，因此配置連線特性的唯一方法，就是設定用以建立連線之連線工廠的屬性。

Message Queue 定義兩種類別的連線工廠物件：

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

這兩種類別共用相同的配置屬性，您可以使用這些屬性最佳化資源、效能和訊息流量。第 16 章「管理物件屬性參照」中列出這些屬性並有詳細說明，下列各節將進行討論：

- 第 157 頁的「連線處理」
- 第 159 頁的「用戶端標識」
- 第 160 頁的「可靠性和流量控制」
- 第 161 頁的「佇列瀏覽器和伺服器階段作業」
- 第 161 頁的「標準訊息特性」

- [第 162 頁的「訊息標頭覆寫」](#)

連線處理

連線處理屬性會指定要連線的訊息伺服器位址，並會視需要指定偵測連線失敗和嘗試重新連線的方式。在[第 294 頁的表 16-1](#) 中會對其進行摘要。

訊息伺服器位址清單

最重要的連線處理屬性是 `imqAddressList`，該屬性會指定要與其建立連線的代理程式。此屬性值是包含訊息伺服器位址或 (如果是代理程式叢集) 由逗號分隔的多項位址之字串。視所使用的連線服務 (請參閱[第 74 頁的「連線服務」](#)) 和建立連線的方法之不同，伺服器位址可使用多種定址方案：

- `mq` 使用代理程式的连接埠對映器，為 `.jms` 或 `ssljms` 連線服務動態地指定连接埠。
- `mqtcp` 會略過连接埠對映器，並使用 `jms` 連線服務，直接連線至指定的连接埠。
- `mqssl` 使用 `ssljms` 連線服務，為指定的连接埠建立安全套接層 (SSL) 連線。
- `http` 使用 `httpjms` 連線服務，將超文字傳輸協定 (HTTP) 連線至指定 URL 的 Message Queue 通道 Servlet。
- `https` 使用 `httpsjms` 連線服務，將安全超文字傳輸協定 (HTTPS) 連線至指定 URL 的 Message Queue 通道 Servlet。

這些定址方案在[第 295 頁的表 16-2](#) 中有所摘要。

每個訊息伺服器位址的一般格式為：

```
scheme://address
```

其中 *scheme* 是上述所列的其中一個定址方案，而 *address* 則表示伺服器位址本身。指定位址的實際語法會根據定址方案而有所差異，如[表 16-2](#) 最後一欄所示。[第 296 頁的表 16-3](#) 會顯示各種位址格式的範例。

在多個代理程式叢集環境中，位址清單可包含多個伺服器位址。如果第一個連線嘗試失敗，Message Queue 用戶端執行階段會嘗試連線至清單中的其他位址，一直到使用完清單上所有位址為止。有兩個其他的連線工廠屬性會控制完成此作業的方式：

- `imqAddressListBehavior` 會指定特定位址的嘗試順序。如果此屬性設定為 `PRIORITY` 字串，將會以在位址清單中出現的順序來嘗試位址。如果屬性值是 `RANDOM`，則會以隨機的順序來嘗試位址；例如，當有許多 Message Queue 用戶端共用相同的連線工廠物件時，這會有效避免用戶端全部嘗試連線至相同的伺服器位址。

- `imqAddressListIterations` 會指定在放棄並報告失敗之前，循環清單的次數。數值 `-1` 表示不限制循環次數；用戶端執行階段會繼續嘗試，直到成功建立連線或時間終了為止，要視哪種情況先發生。

自動重新連線

您可以將連線工廠的 `imqReconnectEnabled` 屬性設定成 `true`，可讓用戶端在連線失敗時，能自動重新連線至代理程式。`imqReconnectAttempts` 屬性會控制重新連線至指定伺服器位址的嘗試次數；`imqReconnectInterval` 會指定每次嘗試之間等待的時間間隔（以毫秒為單位）。

訊息伺服器位址清單 (`imqAddressList`) 在代理程式叢集中會指定多個位址，失敗的連線不只在原始代理程式上復原，也會在叢集中其他代理程式上復原。如果重新連線至原始代理程式失敗，則用戶端執行階段將會嘗試清單中的其他位址。如上節所述，`imqAddressListBehavior` 和 `imqAddressListIterations` 屬性會控制嘗試位址的順序，以及清單循環的次數。每個位址會以 `imqReconnectInterval` 毫秒的間隔重複嘗試，嘗試次數上限由 `imqReconnectAttempts` 指定。

自動重新連線支援訊息使用的所有用戶端確認模式。連線一旦重新建立，代理程式就會重新傳送之前已傳送但尚未確認的所有訊息，並使用「重新傳送」旗標加以標示。應用程式碼可使用此旗標來判斷是否有已使用但尚未確認的訊息。（如果不是長期用戶，一旦關閉連線，訊息伺服器並不會保留訊息。因此當連線關閉時，這類用戶所產生的任何訊息將無法在重新連線後傳送，且會遺失。）進行自動重新連線時會阻斷訊息的產生，因此訊息產生者在連線尚未重新建立之前，無法傳送訊息到伺服器。

自動重新連線可提供連線容錯移轉，但不提供資料容錯移轉：當用戶端自動重新連線至不同的代理程式實例時，失敗或斷線的代理程式所保留的永久性訊息和其他狀態資訊將會遺失。嘗試重新建立連線時，`Message Queue` 會維護用戶端執行階段所提供的物件（例如階段作業、訊息用戶和訊息產生者）。當連線失敗時，也會短暫維護暫時目標，因為用戶端可能會重新連線並再次存取這些目標；在提供用戶端時間重新連線並使用這些目標之後，代理程式即會刪除這些目標。如果重新連線後，用戶端狀態無法完全在代理程式上復原（例如，當使用已處理的階段作業時，只有在連線期間才存在這些階段作業），將不會發生自動重新連線，且會呼叫連線的異常處理程式。接著會由應用程式碼擷取異常、重新連線及復原狀態。

定期測誤 (偵測) 連線

`Message Queue` 用戶端執行階段可以配置成定期測試或「偵測」連線，在嘗試的訊息傳輸失敗前，就事先偵測到連線失敗。當用戶端應用程式只會使用訊息而不會產生訊息時，此測試特別重要，因為這類應用程式無法偵測到連線失敗。很少產生訊息的用戶端也可受惠於此功能。

連線工廠屬性 `imqPingInterval` 會指定偵測連線的頻率 (以秒為單位)。依預設，間隔時間設定為 30 秒，數值 -1 表示停用偵測作業。

偵測失敗的回應會根據不同的作業系統平台而有所差異。在某些系統上，會立即對用戶端應用程式的異常偵聽程式產生一個異常。(如果用戶端沒有異常偵聽程式，下次嘗試使用連線將失敗。) 其他的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為止。

用戶端標識

第 297 頁的表 16-4 中所列的連線工廠屬性，支援用戶端認證以及長期用戶的用戶端識別碼設定。

用戶端認證

連線至代理程式的所有嘗試，必須透過將使用者名稱和密碼與訊息伺服器維護的使用者儲存庫進行對照的方式認證。如果用戶端在建立連線時，未明確提供使用者名稱和密碼，則會由連線工廠屬性 `imqDefaultUsername` 和 `imqDefaultPassword` 指定要使用的預設使用者名稱和密碼。

Message Queue 提供 `guest` 使用者帳戶 (其中使用者名稱和密碼均為 `guest`)，以省去開發者在應用程式開發和測試期間需要寫入使用者儲存庫的麻煩。這也是 `imqDefaultUsername` 和 `imqDefaultPassword` 屬性的預設值，因此如果這些屬性沒有明確指定，用戶端一律可以使用 `guest` 帳戶取得連線。在生產環境中，對代理程式連線的存取應僅限於在使用者儲存庫中明確註冊的使用者。

用戶端識別碼

每當訊息伺服器必須代表用戶端維護永久性狀態時，「Java Message Service 規格」會要求連線提供唯一的用戶端識別碼。Message Queue 使用這類用戶端識別碼，追蹤主題目標的長期用戶。如果長期用戶變為非使用中，代理程式會保留該主題的所有內送訊息，等到用戶恢復為使用中時再傳送這些訊息。代理程式會藉由用戶端識別碼來識別用戶。

用戶端應用程式可能會使用連線物件的 `setClientID` 方法，透過程式設計來設定自己的用戶端識別碼，因此很難協調用戶端識別碼並確保每個識別碼都是唯一的。通常比較好的方式是讓 Message Queue 在代表用戶端建立連線時，自動指定唯一的識別碼。若要這麼做，請將連線工廠的 `imqConfiguredClientID` 屬性設定為下列格式的值

```
#{u}factoryID
```

`{u}` 字元必須為屬性值的前四個字元。(在括弧之間若有 `u` 以外的其他字元，會造成在建立連線時產生一個異常；如果是在其他位置，則這些字元並不具有任何特殊意義，而會被當作是純文字來處理。) `factoryID` 的值是與此連線工廠物件唯一關聯的字元字串。

為特定用戶端建立連線時，`Message Queue` 會以 `u:userName` 取代 `{u}` 字元，以建構用戶端識別碼，其中 `userName` 是供連線認證的使用者名稱。這可確保指定連線工廠所建立的連線雖然其他各方面皆完全相同，但仍會有各自唯一的用戶端識別碼。例如，如果使用者名稱是 Calvin，而為連線工廠的 `imqConfiguredClientID` 屬性指定的字串是 `{u}Hobbes`，則指定的用戶端識別碼會是 `u:CalvinHobbes`。

注意 如果有兩個用戶端使用預設的使用者名稱 `guest` 嘗試取得連線，則無法使用此方案，因為這兩個用戶端的用戶端識別碼會有相同的 `{u}` 元件。在這種情況下，只有第一個請求連線的用戶端才能取得連線，第二個嘗試連線的用戶端則會失敗，因為 `Message Queue` 無法使用同一用戶端識別碼建立兩個連線。

即使您使用 `imqConfiguredClientID` 指定用戶端識別碼，用戶端應用程式還是會使用連線方法 `setClientID` 覆寫此設定值。您可以將連線工廠的 `imqDisableSetClientID` 屬性設定為 `true`，以避免這種情況。請注意，對於使用長期用戶的應用程式，其用戶端識別碼必須以下列兩種方法之一進行設定：使用 `imqConfiguredClientID` 以管理員方式設定，或者使用 `setClientID` 以程式設計方式設定。

可靠性和流量控制

由於用戶端傳送與接收的「有效負載」訊息以及 `Message Queue` 本身使用的控制訊息 (例如代理程式確認) 透過相同的用戶端與代理程式連線傳送，因此過量的有效負載流量可能會干擾控制訊息的傳送。若要協助減輕此問題，[第 297 頁的表 16-5](#) 中所列的連線工廠屬性可讓您管理這兩種訊息類型的相關流量。這些屬性可分為四種：

- **確認逾時** 指定產生一個異常之前，等待代理程式確認的最長時間 (`imqAckTimeout`)。
- **連線流量度量** 會限制有效負載訊息傳輸為指定大小的批次 (`imqConnectionFlowCount`)，確定可定期傳送任何累積的控制訊息。

- **連線流量控制**會限制連線上可以保留擱置等待的有效負載訊息數 (imqConnectionFlowLimit)。達到限制時，傳送到連線的有效負載訊息會暫停，直到等待使用的訊息數量降到限制以下為止。此功能的使用是由布林旗標 (imqConnectionFlowLimitEnabled) 所控制。
- **用戶流量控制**會限制任何單一用戶可以擱置等待使用的有效負載訊息數 (imqConsumerFlowLimit)。(此限制也可以指定為特定佇列目標的特性 consumerFlowLimit。)達到限制時，傳送到用戶有效負載訊息會暫停，直到等待使用的訊息數量 (以 imqConsumerFlowLimit 百分比表現) 降到 imqConsumerFlowThreshold 屬性所指定的限制以下為止。這可以防止任何用戶佔用連線讓別人苦等，幫助改善多名用戶之間的負載平衡。

使用這些流量控制技術必須在可靠性與流量之間進行取捨；如需詳細說明，請參閱第 216 頁的「用戶端執行階段訊息流量調校」。

佇列瀏覽器和伺服器階段作業

第 299 頁的表 16-6 會列出影響用戶端佇列瀏覽與伺服器階段作業的連線工廠屬性。imqQueueBrowserMaxMessagesPerRetrieve 屬性會指定瀏覽佇列目標內容時，一次擷取的最大訊息數量；imqQueueBrowserRetrieveTimeout 會提供擷取訊息的最長等待時間。布林屬性 imqLoadMaxToServerSession 管理連線用戶在應用程式伺服器階段作業中的運作方式：若此屬性值為 true，用戶端將上載最大數量的訊息到伺服器階段作業中；若為 false，用戶端則僅會一次載入一個訊息。

標準訊息特性

「Java Message Service 規格」會定義部分標準訊息特性，JMS 提供者 (例如，Message Queue) 會選擇性支援這些特性。根據慣例，所有這類標準特性的名稱會以字母 JMSX 開頭。第 299 頁的表 16-7 中所列的連線工廠屬性可控制 Message Queue 用戶端執行階段是否設定特定的標準特性。若是產生的訊息，則包括下列特性：

JMSXUserID	傳送訊息的使用者驗證身份
JMSXAppID	傳送訊息的應用程式驗證身份
JMSXProducerTXID	作業事件的作業事件識別碼， 訊息產生於此作業事件內

若是使用的訊息，則包括

JMSXConsumerTXID	作業事件的作業事件識別碼， 訊息於此作業事件內使用
JMSXRcvTimestamp	訊息傳送至用戶的時間

訊息標頭覆寫

您可以使用第 300 頁的表 16-8 中所列的連線工廠屬性，覆寫用戶端為部分 JMS 訊息標頭欄位設定的值。由該連線工廠取得的連線所產生之所有訊息會使用您指定的設定值。您可以此方式覆寫的標頭欄位包括：

- JMSDeliveryMode 傳送模式 (永久性或非永久性)
- JMSExpiration 過期時間
- JMSPriority 優先順序層級

每個欄位均有兩個屬性：一個是控制能否覆寫欄位的布林值，另一個是指定該布林值。例如，設定優先順序層級的屬性有 `imqOverrideJMSPriority` 和 `imqJMSPriority`。另外還有一個屬性 `imqOverrideJMSHeadersToTemporaryDestinations`，它會控制覆寫值是否會套用到暫時目標。

注意 由於覆寫這些訊息標頭可能會干擾特定應用程式的需求，請先諮詢應用程式設計者或使用者後，再使用這些屬性。

目標屬性

識別實體佇列或主題目標的目標管理物件只有兩個屬性，如第 300 頁的表 16-9 所列。重要的屬性是 `imqDestinationName`，其提供此管理物件所代表的實體目標名稱；該名稱已使用建立實體目標的 `imqcmd create dst` 指令之 `-n` 選項來指定。另外還有一個選擇性描述字串 `imqDestinationDescription`，可用以協助您識別目標物件，並與您之前建立的物件進行區分。

使用物件管理員公用程式

Message Queue 物件管理員公用程式 (`imqobjmgr`) 可讓您建立並管理受管理物件。`imqobjmgr` 指令提供下列子指令，以在管理物件上執行各種作業：

- `add` 將管理物件新增至物件存放區
- `delete` 從物件存放區刪除管理物件
- `list` 列出物件存放區中現有的管理物件
- `query` 顯示有關管理物件的資訊

update 修改管理物件的屬性

如需 imqobjmgr 指令的語法、子指令和選項之參照資訊，請參閱第 259 頁的「物件管理員公用程式」。

大多數物件管理員作業需要您指定下列資訊以作為 imqobjmgr 指令的選項：

- **管理物件的 JNDI 查詢名稱** (-l 選項)

這是用戶端應用程式使用 Java Naming and Directory Interface，在物件存放區中查詢管理物件所根據的邏輯名稱。

- **JNDI 物件存放區的屬性** (-j 選項)

如需有關可能的屬性及其值的資訊，請參閱第 154 頁的「物件存放區」。

- **管理物件的類型** (-t 選項)

可能的類型包括：

- q 佇列目標
- t 主題目標
- cf 連線工廠
- qf 佇列連線工廠
- tf 主題連線工廠
- xcf 分散式作業事件的連線工廠
- xqf 分散式作業事件的佇列連線工廠
- xtf 分散式作業事件的主題連線工廠
- e SOAP 終點

- **管理物件的屬性** (-o 選項)

如需有關可能的屬性及其值的資訊，請參閱第 156 頁的「管理物件屬性」。

新增管理物件

imqobjmgr 指令的 add 子指令，會將連線工廠和主題或佇列目標的管理物件新增至物件存放區。儲存在 LDAP 物件存放區中的管理物件必須擁有以 cn= 開頭的查詢名稱字首；檔案系統物件存放區中的查詢名稱則不需要有特定的字首，但是絕對不能包含斜線字元 (/)。

注意 物件管理員僅會列出與顯示 Message Queue 管理物件。如果物件存放區應包含的非 Message Queue 物件，與您想新增的管理物件具有相同的查詢名稱，則當您嘗試新增作業時會出現錯誤。

新增連線工廠

若要讓用戶端應用程式能建立代理程式連線，請為要建立的連線類型新增連線工廠管理物件：佇列連線工廠或主題連線工廠。[程式碼範例 8-1](#) 顯示將佇列連線工廠 (管理物件類型 `qf`) 新增至 LDAP 物件存放區所使用的指令。物件具有查詢名稱 `cn=myQCF`，並使用 `jms` 連線服務連線至連接埠號碼 7272 之主機 `myHost` 上所執行的代理程式。

程式碼範例 8-1 新增連線工廠

```
imqobjmgr add
-l "cn=myQCF"
-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=homerSimpson,ou=People,o=imq"
-j "java.naming.security.credentials=doh"
-j "java.naming.security.authentication=simple"
-t qf
-o "imqAddressList=mq://myHost:7272/jms"
```

新增目標

建立代表目標的管理物件時，最好先建立實體目標，再將管理物件新增至物件存放區中。使用指令公用程式 (`imqcmd`) 建立實體目標，如[第 114 頁的「建立實體目標」](#)所述。

[程式碼範例 8-2](#) 中顯示的指令會將管理物件新增至代表主題目標之 LDAP 物件存放區，且該目標的查詢名稱為 `myTopic`，實體目標名稱為 `physTopic`。新增佇列目標的指令很類似，不同之處在於管理物件類型 (`-t` 選項) 為 `q` (表示佇列目標)，而非 `t` (表示主題目標)。

程式碼範例 8-2 將目標新增至 LDAP 物件存放區

```
imqobjmgr add
-l "cn=myTopic"
-j "java.naming.factory.initial=com.sun.jndi.ldap.LdapCtxFactory"
```

程式碼範例 8-2 將目標新增至 LDAP 物件存放區

```

-j "java.naming.provider.url=ldap://mydomain.com:389/o=img"
-j "java.naming.security.principal=uid=homerSimpson,ou=People,o=img"
-j "java.naming.security.credentials=doh"
-j "java.naming.security.authentication=simple"
-t t
-o "imgDestinationName=physTopic"

```

程式碼範例 8-3 顯示相同的指令，但是管理物件存放區在 Solaris 檔案系統中，而不是在 LDAP 伺服器中。

程式碼範例 8-3 將目標新增至檔案系統物件存放區

```

imgobjmgr add
-l "cn=myTopic"
-j "java.naming.factory.initial=
    com.sun.jndi.fscontext.RefFSContextFactory"
-j "java.naming.provider.url=file:///home/foo/img_admin_objects"
-t t
-o "imgDestinationName=physTopic"

```

刪除管理物件

若要從物件存放區刪除管理物件，請使用 `imgobjmgr` 指令的 `delete` 子指令，指定要刪除的物件之查詢名稱、類型和位置。**程式碼範例 8-4** 中顯示的指令會刪除上述 **程式碼範例 8-2** 中新增的物件。

程式碼範例 8-4 刪除管理物件

```

imgobjmgr delete
-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=homerSimpson,ou=People,o=img"
-j "java.naming.security.credentials=doh"
-j "java.naming.security.authentication=simple"
-t t

```

列出管理物件

您可以使用物件管理員的 `list` 子指令，會列出物件存放區中或特定類型的所有管理物件。[程式碼範例 8-5](#) 顯示如何列出 LDAP 伺服器上所有的管理物件。

程式碼範例 8-5 列出所有管理物件

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

[程式碼範例 8-6](#) 列出所有佇列目標 (類型 `q`)。

程式碼範例 8-6 列出特定類型的管理物件

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

檢視管理物件資訊

`query` 子指令會顯示特定管理物件的相關資訊，該物件經由物件的查詢名稱和包含物件的物件存放區屬性加以識別。[程式碼範例 8-7](#) 會顯示查詢名稱為 `cn=myTopic` 之物件的相關資訊。

程式碼範例 8-7 檢視管理物件資訊

```
imqobjmgr query
-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=homerSimpson,ou=People,o=imq"
```

程式碼範例 8-7 檢視管理物件資訊

```
-j "java.naming.security.credentials=doh"
-j "java.naming.security.authentication=simple"
```

修改管理物件屬性

若要修改管理物件的屬性，請使用 `imqobjmgr update` 子指令。請提供物件的查詢名稱和位置，再使用 `-o` 選項指定新的屬性值。

程式碼範例 8-8 是針對第 164 頁的程式碼範例 8-1 中新增至物件存放區的佇列連線工廠變更了 `imqReconnectAttempts` 屬性值。

程式碼範例 8-8 修改管理物件的屬性

```
imqobjmgr update
-l "cn=myQCF"
-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=homerSimpson,ou=People,o=imq"
-j "java.naming.security.credentials=doh"
-j "java.naming.security.authentication=simple"
-t qf
-o "imqReconnectAttempts=3"
```

使用指令檔

`imqobjmgr` 指令的 `-i` 選項，可讓您指定使用 Java 特性檔案語法的指令檔名稱，以表示所有或部分子指令子句。此功能對指定物件存放區屬性極為有用，尤其是在物件存放區屬性一般需要大量的鍵入操作，並有可能在多次呼叫 `imqobjmgr` 中執行時作用顯著。使用指令檔也可讓您避免超出指令行所允許的最大字元數。

程式碼範例 8-9 顯示物件管理員指令檔的一般語法。請注意，`version` 屬性不是指令行選項：這表示指令檔本身的版本（不是 Message Queue 產品的版本），且必須將值設定為 2.0。

程式碼範例 8-9 物件管理員指令檔語法

```
version=2.0
cmdtype=[ add | delete | list | query | update ]
```

程式碼範例 8-9 物件管理員指令檔語法

```
obj.lookupName=lookup name
objstore.attrs.objStoreAttrName1=value1
objstore.attrs.objStoreAttrName2=value2
.
.
.
objstore.attrs.objStoreAttrNameN=valueN
obj.type=[ q | t | cf | qf | tf | xcf | xqf | xtf | e ]
obj.attrs.objAttrName1=value1
obj.attrs.objAttrName2=value2
.
.
.
obj.attrs.objAttrNameN=valueN
```

例如，之前在第 164 頁的程式碼範例 8-1 中所顯示的物件管理員指令，會將佇列連線工廠新增至 LDAP 物件存放區。此指令可以封裝在程式碼範例 8-10 中所顯示的指令檔中。若是指令檔命名為 MyCmdFile，您可以使用指令行執行指令

```
imqobjmgr -i MyCmdFile
```

程式碼範例 8-10 範例指令檔

```
version=2.0
cmdtype=add
obj.lookupName=cn=myQCF
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=homerSimpson,ou=People,o=imq
objstore.attrs.java.naming.security.credentials=doh
objstore.attrs.java.naming.security.authentication=simple
obj.type=qf
obj.attrs.imqAddressList=mq://myHost:7272/jms
```

指令檔也可以用來僅指定 imqobjmgr 子指令子句的一部分，剩下部分則由指令行明確提供。例如，程式碼範例 8-12 中所顯示的指令檔僅指定了 LDAP 物件存放區的屬性值。

程式碼範例 8-11 部分指令檔

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

程式碼範例 8-11 部分指令檔

```
objstore.attrs.java.naming.security.credentials=doh  
objstore.attrs.java.naming.security.authentication=simple
```

您可以接著使用此指令檔，以 `imgobjmgr` 指令來指定物件存放區，同時明確提供其餘選項，如[程式碼範例 8-12](#) 所示。

程式碼範例 8-12 使用部分指令檔

```
imgobjmgr add  
-l "cn=myQCF"  
-i MyCmdFile  
-t qf  
-o "imgAddressList=mq://myHost:7272/jms"
```

根據您使用的平台，可於以下位置找到指令檔的其他範例：

Solaris : /usr/demo/imq/imgobjmgr
Linux : /opt/sun/mq/examples/imgobjmgr
Windows : IMQ_HOME/demo/imgobjmgr

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，為用戶端提供訊息傳送服務。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 技術摘要」。

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。它包含下列各節：

- [第 172 頁的「叢集配置特性」](#)
- [第 173 頁的「管理叢集」](#)
- [第 176 頁的「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 283 頁的表 14-9 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 (可選擇) 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至叢集：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。

若要從指令行配置代理程式叢集，請在啟動每個代理程式叢集時，使用 `imqbrokerd` 指令的 `-cluster` 選項，指定叢集的完整代理程式清單。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。如果位址是以網域命名服務 (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 SSL 型傳輸協定。針對叢集中的每個代理程式，設定 SSL 型連線服務，如第 140 頁的「使用 SSL 型服務」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. (可選擇) 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 177 頁的「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，但經常定期備份可盡量減低此潛在的資訊遺失。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。

► 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。但是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。本章包含下列各節：

- 第 180 頁的「監視工具簡介」
- 第 181 頁的「配置及使用代理程式記錄功能」
- 第 186 頁的「互動顯示度量」
- 第 191 頁的「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 資料格式難以擷取或剖析；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 長間隔時間遲延可能會影響效能
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 您必須輸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 309 頁的 [第 18 章「度量參照」](#)。

配置及使能代理程式記錄功能

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為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 資料的平台特定位置」](#)）：

```
.../instances/instanceName/log/
```

記錄檔為簡單的文字檔案。其命名如下（從最舊到最新）：

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

依預設，記錄檔會每週自動重建一次；而系統會保留九個備份檔案。

- 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層級：ERROR、WARNING、INFO。表 10-2 解釋每個層級的涵義。

表 10-2 記錄層級

層級	說明
ERROR	表示可能導致系統故障問題的訊息。
WARNING	應予以注意但不會導致系統故障的警示。
INFO	度量和其他資訊性訊息的報告。

設定記錄層級可收集此層級以上 (含此層級) 的訊息。預設層級是 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層級而有所不同。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
```

若要變更時間戳記的時區，請參閱第 280 頁的表 14-8 中有關描述 `mq.log.timezone` 特性的資訊。

變更記錄程式配置

第 280 頁的表 14-8 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. 設定記錄層級。
2. 為一個或多個記錄種類設定輸出通道 (檔案、主控台或兩者)。
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。您可以用以下兩種方法之一執行此作業：

- 在啟動代理程式之前，請先變更或新增其 `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置換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>	將記錄層級設定為以下之一：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 層級的訊息。

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 (對於給定的層級) 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 的記錄資訊種類。例如

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

注意 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層級。例如，如果您要將記錄層級設定為 `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 181 頁的「配置及使用代理程式記錄功能」。

► 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` 命令行選項設定此值。

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` 命令行選項設定此值。

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 個位組) 0 個封包 (0 個位元組)
  輸出：0 個訊息 (0 個位組) 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

您可以啓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。

啓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。
- **代理程式通用度量**。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**。有關傳入和傳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 190 頁的「[imqcmd query](#)」，以獲得更多資訊。

imqcmd metrics

`imqcmd metrics` 的語法和選項分別顯示於表 10-4 和表 10-5 中。

表 10-4 imqcmd metrics 子指令語法

指令語法	提供的度量資料
<pre>metrics bkr [-b hostName:portNumber] [-m metricType] [-int interval] [-msp numSamples]</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 serviceName [-b hostName:portNumber] [-m metricType] [-int interval] [-msp numSamples]</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 destType -n destName [-b hostName:portNumber] [-m metricType] [-int interval] [-msp numSamples]</pre>	顯示指定類型和名稱的實體目標度量資訊。

表 10-5 imqcmd metrics 子指令選項

指令選項	說明
-b <i>hostName:portNumber</i>	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 localhost:7676。
-int <i>interval</i>	指定顯示度量的間隔時間 (以秒為單位)。預設為 5 秒鐘。
-m <i>metricType</i>	指定要顯示的度量類型：
ttl	顯示傳入和傳出代理程式、服務或目標的訊息與封包之度量 (預設度量類型)。
rts	顯示傳入和傳出代理程式、連線服務或目標的訊息和封包之流量速率的度量 (每秒)。
cxn	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)。
con	顯示用戶相關的度量 (僅目標)。
dsk	顯示磁碟使用度量 (僅目標)。

表 10-5 imqcmd metrics 子指令選項 (續)

子指令選項	說明
-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)。沒有預設值。
-n <i>serviceName</i>	指定報告度量的連線服務 (如果有)。沒有預設值。
-t <i>destType</i>	指定報告度量資料之實體目標 (如果有) 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。

► 使用 metrics 子指令

1. 啟動需要度量資訊的代理程式。
請參閱第 67 頁的「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。

代理程式連接度量

若要取得訊息和封裝每隔 10 秒傳入和傳出代理程式的速率，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類似的輸出 (請參閱第 310 頁的表 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類似的輸出 (請參閱第 312 頁的表 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類似的輸出 (請參閱第 314 頁的表 18-4 中的資料說明)：

訊息		訊息位元組		訊息計數			訊息總位元組 (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類似的輸出（請參閱第 314 頁的表 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

指令語法	提供的度量資料
<code>query bkr</code> [-b <i>hostName:portNumber</i>]	目前儲存在代理程式記憶體中永久存放區中的訊息數目和訊息位元組的資訊（請參閱第 97 頁的「顯示代理程式資訊」）。
或	
<code>query svc -n <i>serviceName</i></code> [-b <i>hostName:portNumber</i>]	指定連線服務目前已配置的執行緒數目和連線數目相關資訊（請參閱第 103 頁的「顯示連線服務資訊」）。
或	
<code>query dst -t <i>destType</i></code> -n <i>destName</i> [-b <i>hostName:portNumber</i>]	指定目標目前產生者、使用中 and 備份用戶，以及儲存在記憶體和永久存放區中的訊息和訊息位元組等數目的資訊（請參閱第 116 頁的「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 309 頁的第 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 虛擬機容度量
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

設定訊息型監視

本節描述使用訊息型監視功能，收集度量資訊的程序。程序包括用戶端部署和管理工作。

► 設定訊息型監視

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：

- 度量資料可能包含有關代理程式及其資源的機密資訊。
- 度量主題目標的訂閱數過量, 可能會增加代理程式的經常性耗用時間, 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至代理程式。

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 134 頁的「[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至代理程式。如需使用加密連線的資訊，請參閱第 140 頁的「使用 SSL 型服務」。

度量輸出：度量訊息

您使用訊息型監視的 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 309 頁的「度量參照」。

撰寫應酬程式來監視代理程式

分析與調校訊息服務

本章涵蓋一些如何分析與調校 Message Queue 服務的主題，以最佳化訊息傳送應用程式的效能。它包括下列主題：

- 第 196 頁的「關於效能」
- 第 199 頁的「影響效能的因素」
- 第 211 頁的「調校配置以改善效能」

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。

最佳化效能的程序首先是應用程式設計，然後是部署應用程式，之後是調校訊息服務。效能調校程序包括以下階段：

- 定義應用程式的效能需求
- 設計應用程式時要考量影響效能的因素 (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。部署應用程式期間，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，效能是測量訊息服務從產生者傳送訊息到用戶的速度和效率。但是，您可能會依您的需要關注效能的不同方面。

連線負載 系統可支援之訊息產生者、訊息用戶或同步運作連線的數目。

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需的時間。

穩定性 訊息服務的整體可用性，或者在負載量較大或失敗時訊息服務緩慢降低的程度。

效率 訊息傳送的效率；是與計算已使用資源相關的訊息流量的測量。

通常這些不同方面的效能是相互關聯的。如果訊息流量高，則表示訊息不太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

效能評定為針對訊息傳送應用程式建立測試套件以及測量訊息流量的程序，或者適用於此測試套件的其他方面效能。

例如，您可能建立一套測試套件，數個產生者用戶端會依據此測試套件，使用數個連線、階段作業和訊息產生者以某個特定速率，將大小標準的永久性或非永久性訊息傳送到數個佇列或主題（皆取決於您訊息傳送應用程式的設計）。同樣地，測試套件包括數個使用數個連線和階段作業的用戶用戶端，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，您可以修改部分測試套件的特徵。小心控制所有可能影響效能的因素（請參閱第 200 頁的「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是如何影響效能評定的。例如，您可以將連線數目或訊息大小增加五倍或十倍，即會注意到效能所受的影響。

相反地，您可以保持應用程式的因素不變，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、限制連戶方式、檔案型與 JDBC 型永久性等），並注意這些變更對效能所產生的影響。

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，效能評定應在受控制的測試環境中執行，並且要時間足夠長以讓訊息服務穩定執行。（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

一旦部署並執行了訊息傳送應用程式，建立基本使用式樣是很重要的。您必須知道尖峰需求出現的時間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、活動層級、每天的時間或所有這些因素而有所變化。

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，查看下列資料：

- 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
- 傳入或和傳出代理程式 (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值與尖峰值。

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：

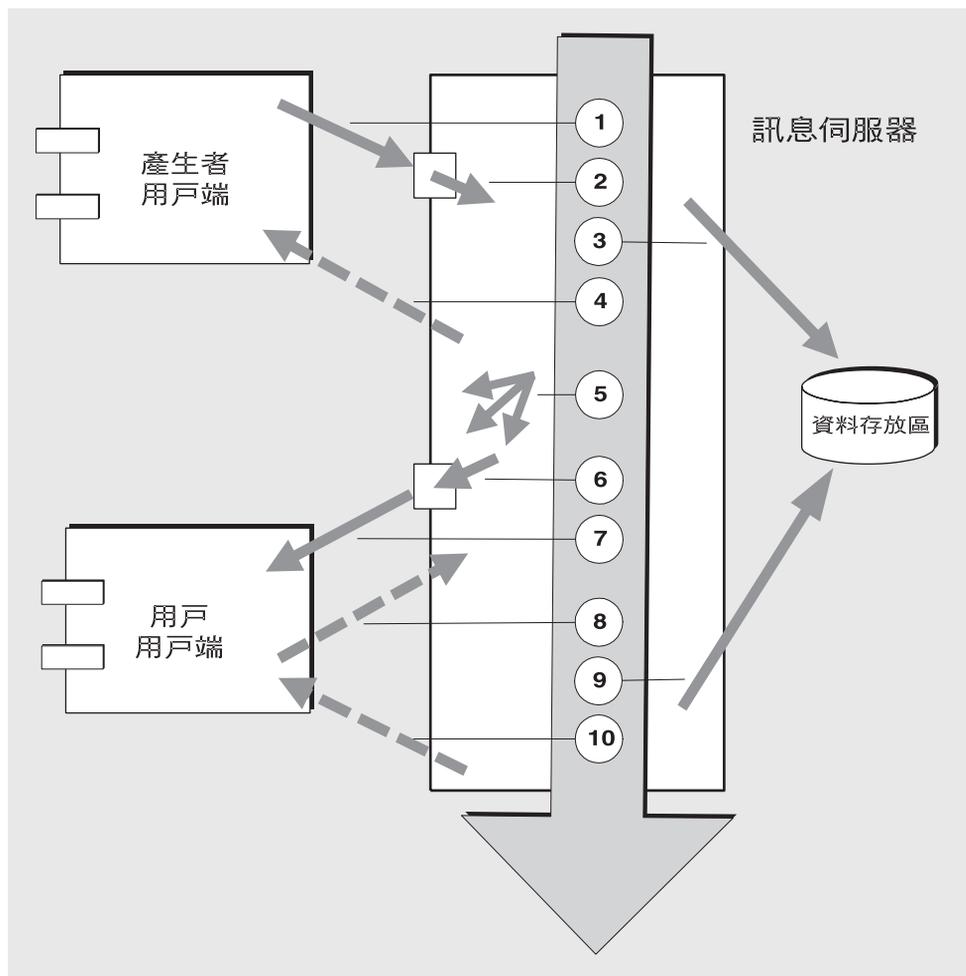
- 如果某個實體目標的使用度較其他實體目標高，則您可能要將此實體目標的訊息記憶體限制設定的比其他實體目標的限制高，或者要相應地調校限制運作方式。
- 如果所需的連線數目明顯大於最大執行緒池容量允許的大小，則您可能要增加執行緒池大小，或調校共用執行緒模型。
- 如果尖峰訊息流量遠遠大於平均流量，則在記憶體不足時，可能會影響您使用的限制運作方式。

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標，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。以下顯示的步驟適用於持續可靠傳送訊息的情況。下圖描述這些步驟。

圖 11-1 透過 Message Queue 服務的訊息傳送



1. 這些訊息由產生者用戶端傳送到訊息伺服器。
2. 訊息伺服器讀取訊息。

3. 訊息會置於永久存放區中，以確保其可靠性。
4. 訊息伺服器會確認接收到訊息，以確保其可靠性。
5. 訊息伺服器會判斷訊息的路由。
6. 訊息伺服器會寫出訊息。
7. 這些訊息會由訊息伺服器傳送至用戶用戶端。
8. 用戶用戶端會確認接收到訊息，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，所以在從產生者用戶端傳送訊息至用戶用戶端時，其中任何步驟都可能是瓶頸。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層級。

下列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，將單獨考量每個種類。

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)
- 作業事件的使用
- 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)
- 訊息大小
- 訊息內文類型

下列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：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 / 低效能方案和高效能 / 低可靠性方案) 和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	高性能 低可靠性方案
傳送模式	永久性訊息	非永久性訊息
使用作業事件	異動的階段作業	無作業事件
確認模式	AUTO_ACKNOWLEDGE 或 CLIENT_ACKNOWLEDGE	DUPS_OK_ACKNOWLEDGE
長期 / 非長期間隔	長期間隔	非長期間隔
選擇性的使用	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
訊息的類型	複雜的類型	簡單的類型

備註

以下各圖為使用檔案型永久性，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。

代理程式暖機後，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)。此方案會針對應用程式設計因素的不同組合而重複，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發生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。

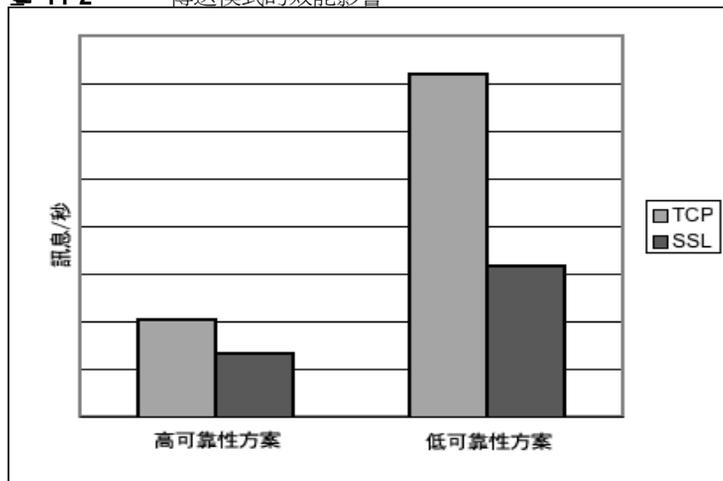
永久性訊息的代理程式處理速度會比非永久性的訊息處理速度慢，原因如下：

- 代理程式必須可靠地儲存永久性訊息，如此一來，代理程式發生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，就確保會傳送至代理程式。
- 根據用戶端的確認模式，代理程式可能需要確認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可能會非常明顯。圖 11-2 比較了在兩種可靠傳送的情況下，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。

圖 11-2 傳送模式的效能影響



作業事件的使用

作業事件可確保所有異動階段作業中產生的訊息和使用的訊息，皆會作為一個單位來處理或不處理（回轉）。

Message Queue 支援本機作業事件和分散式作業事件。

異動階段作業中產生或確認訊息的速度會比在非異動階段作業中的速度慢，原因如下：

- 附加資訊必須隨每個產生訊息一起儲存。
- 在一些情況下，會儲存作業事件屬中的訊息，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，但是在作業事件屬開始時，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

確保 JMS 訊息可靠傳送的機制，是用戶端確認已使用 Message Queue 訊息伺服器傳送來的訊息。

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。

對於非異動階段作業而言，用戶端可以從三種確認模式中選擇一種，每一種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會重新傳送一次訊息。
- `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。如果在處理一組確認時，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。(此代理程式確認可確保代理程式不會意外重新傳送這些訊息。)
- Message Queue 永久存放區必須以用戶接收的所有永久性訊息的確認資訊來更新，因此會降低效能。

長期與非長期訂閱

主題目標的訂閱者分為兩個種類，長期或非長期訂閱。

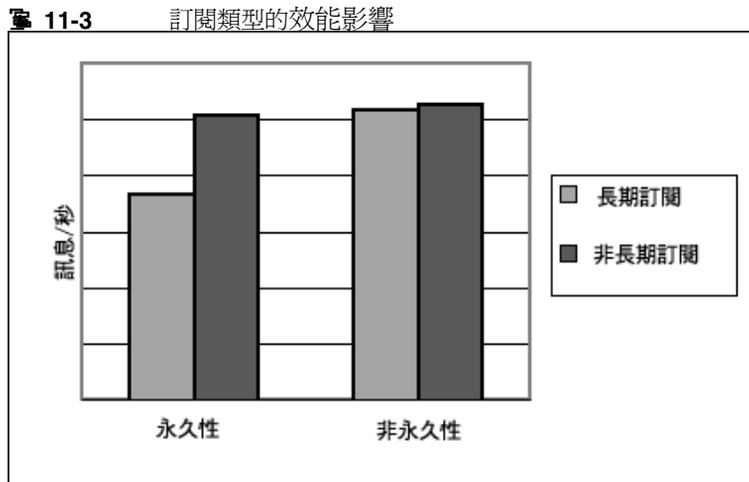
長期訂閱可提高可靠性，但會降低流量，原因如下：

- Message Queue 訊息伺服器必須永久儲存指定到每個長期訂閱的訊息清單，這樣，一旦訊息伺服器發生故障，即可在回復後使用此清單。

- 長期訂閱的永久性訊息會被永久儲存，這樣，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，非長期訂閱的永久性訊息不會被永久儲存 (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。



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)，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。

訊息大小

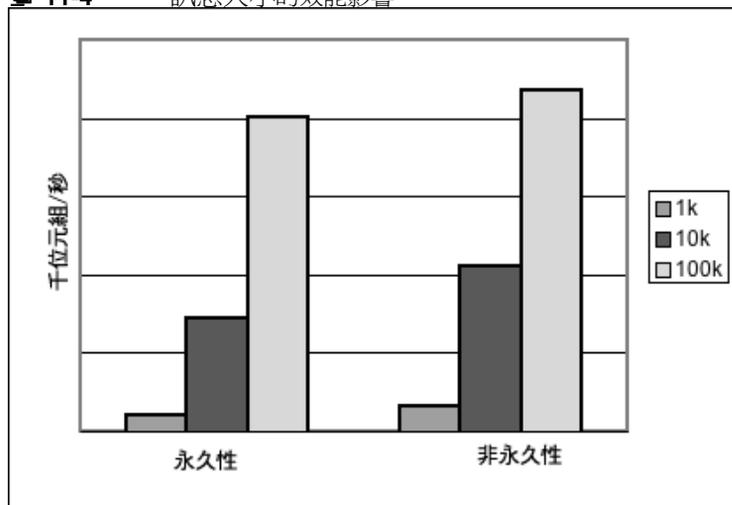
訊息大小會影響效能，因為必須將更多資料從產生者用戶端傳送到代理程式，再從代理程式傳送到用戶用戶端，且必須儲存較大的永久性訊息。

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，以改善整體效能。在此情況下，會遺失個別訊息狀態的相關資訊。

圖 11-4 會比較兩種情況下，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 大小的訊息中，非永久性訊息的效能比永久性訊息的效能高出將近 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 訊息大小的效能影響



訊息內容類型

JMS 支援五種訊息內容類型，依複雜性大致排列如下：

- `BytesMessage`：包含應用程式所判斷的位元組格式。
- `TextMessage`：為簡單 Java 語言字串。
- `StreamMessage`：包含 Java 原始值的串流。

- `MapMessage`：包含名稱與值成對的組合。
- `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，即序列化和取消序列化資料的效能。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校訊息服務大小、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
- 資料存放區效能
- 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)

訊息伺服器為在主機 JVM 中執行且受其支援的 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 會定期收回未使用的記憶體，且記憶體收回會延遲訊息處理。JVM 記憶體堆疊的容量越多，記憶體收回過程中可能的延遲時間越長。

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進行的。對同時運作的連線數目有任何限制，皆會影響目前可同時使用訊息伺服器之產生者用戶端或用戶用戶端的數目。

與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。可配置 Message Queue 以支援專用的執行緒模型或共用的執行緒模型（請參閱第 75 頁的「執行緒池管理」）。

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 74 頁的「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



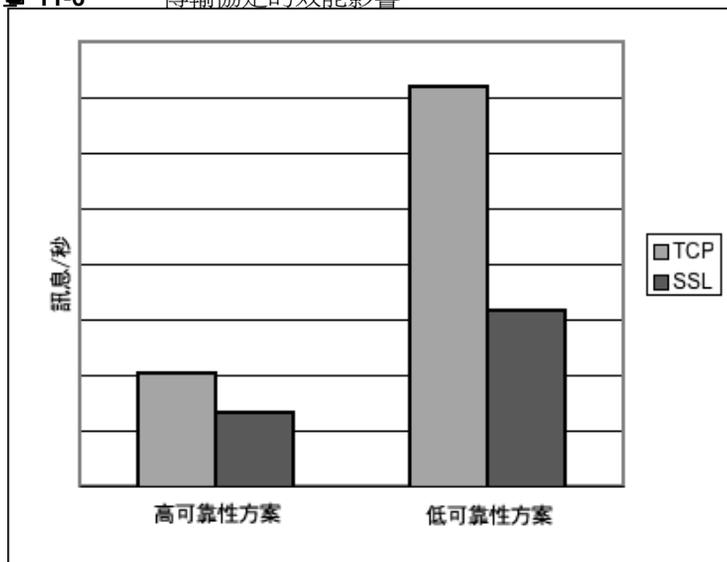
圖 11-5 會反映各種協定技術的效能特徵：

- 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金鑰。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 會比較 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影響較少的協定。這可能是因為在高可靠性情況中需要的永久性耗用時間，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 將在 Web Server 上執行的 Servlet 用作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 需要額外的時間加密用戶端與 Servlet 間和 Servlet 與代理程式間的封裝。

訊息伺服器架構

可以將 Message Queue 訊息伺服器實施作為單一代理程式或多重互連代理程式實例的代理程式叢集。

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)。提供增加負載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發行用戶端連線與訊息路由和傳送。

一般而言，如果用戶端 (尤其是訊息生產型用戶端) 在叢集上分散均勻，此調校會使運作達到最佳狀態。由於耗用時間受到叢集中代理程式間訊息傳送的影響，所以具有連線數目限制或訊息傳送速率限制的叢集，其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，您可以在叢集中的一組遠端代理程式間使用速度較慢、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傳入和傳出訊息，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。

資料存放區效能

Message Queue 可支援檔案型與 JDBC 永久性模組。檔案型永久性使用個別的檔案來儲存永久性資料。JDBC 型永久性使用 Java 資料庫連線 (JDBC™) 介面，並需要 JDBC 相容的資料存放區。檔案型永久性的速度通常比 JDBC 型更快；但是，部分使用者喜愛使用 JDBC 相容存放區提供的備援與管理控制功能。

在使用檔案型永久性的情況下，您可以指定永久性作業同步化包含資料存放區的內部記憶體狀態，以保持最佳可靠性。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。設定連線工廠屬性值就能夠配置用戶端執行階段，讓您控制它在改善效能與訊息流量等方面的運作方式，例如連線流量測量、用戶流量限制與連線流量限制。如需用於配置運作方式的這些功能與屬性的詳細資訊，請參閱第 216 頁的「[用戶端執行階段訊息流量調校](#)」。

調校配置以改善效能

系統調校

以下各節說明您可對作業系統、JVM 和連線協定所做的調校。

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 虛擬機器調校

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，並降低訊息的傳入量。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。

一般而言，最好是先評估正常時和尖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，而導致系統記憶體問題。

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` 指令行選項。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「指令行參照」中的「服務管理員公用程式」。

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.protocolType.nodelay`
- `imq.protocol.protocolType.inbufsz`
- `imq.protocol.protocolType.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。HTTP/HTTPS 協定有其他可以影響效能的特性 (請參閱第 214 頁的「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` 套接層選項)。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 會嘗試阻止一些小資料區塊傳送到遠端系統 (藉由將資料隨附於較大的封包中)。如果寫入套接的資料未填滿所需緩衝區大小，則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 / 確認互動：用戶端傳送資料封包給代理程式且等待確認。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，代理程式在傳送確認到用戶時，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確認的情況下改善效能。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

該 *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，則封包 (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

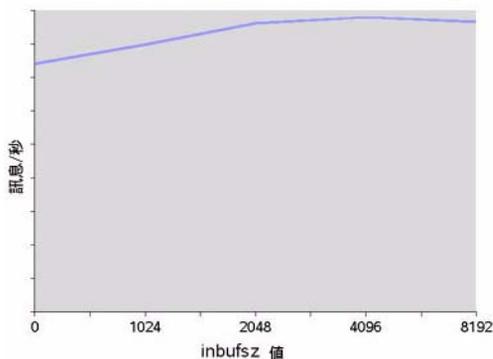
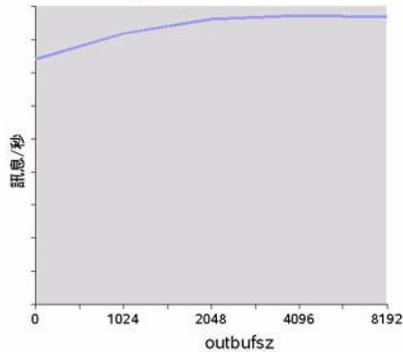


圖 11-8 顯示在 1K 的封包上變更 outbufsz 的結果。

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，則您可能需要調校 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，您也可以調校用戶端輪詢 Web Server 的頻率。HTTP 為請求型協定。這表示使用 HTTP 型協定的用戶端會定期檢查 Web Server，查看是否有等待訊息。imq.httpjms.http.pullPeriod 代理程式特性 (及其對應的 imq.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)，則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，則您可能要將 pullPeriod 設為正值，以節省 Web Server 資源。

調校檔案型永久存放區

如需調校檔案型永久存放區，請參閱第 78 頁的「永久性服務」。

代理程式調校

以下各節描述您可對代理程式特性進行的調校，並藉以改善效能。

記憶體管理：增加代理程式負載的穩定性

可以在目標對目標層級或系統範圍層級（用於整體的所有目標）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」。

使用系統範疇限制

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，則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。

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。（請注意，因為代理程式無法接收訊息，所以異常將永遠不會到期；唯一會發生的異常就是由傳送端的用戶端偵測到的異常）。

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：

- 使用中的用戶數目 (`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，必須有足夠的使用中用戶數目，以滿足佇列產生訊息的速率，並且必須路由佇列中的訊息，然後將其傳送到使用中用戶。此方法會最大化訊息使用率。「Sun Java System Message Queue 技術摘要」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，則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，則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

本節討論影響效能的流量控制運作方式 (請參閱第 210 頁的「用戶端執行階段配置」)。請將這些運作方式配置為連線工廠管理物件的屬性。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 (有效負載訊息) 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。如果有效負載訊息傳送阻擋了控制訊息，則會造成延遲傳送控制訊息 (例如代理程式確認)。若要避免此擁塞情形，Message Queue 會計算連線上的有效負載訊息流量。

有效負載訊息分為幾個批次 (如連線工廠屬性 `imqConnectionFlowCount` 所指定)，因此只傳送一組數字。批次傳送後，會暫停傳送有效負載訊息，而只傳送擱置的控制訊息。接著傳送其他有效負載訊息，然後傳送擱置控制訊息，重複此循環。

如果用戶端會進行需要代理程式大量確認的作業，則 `imqConnectionFlowCount` 的值應保持較低：例如，若用戶端正在使用 `CLIENT_ACKNOWLEDGE` 或 `AUTO_ACKNOWLEDGE` 模式、永久性訊息、作業事件，或佇列瀏覽器，或者正在新增或移除用戶時。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，則您可以增加 `imqConnectionFlowCount` 的值，而不需顧及效能。

訊息流量限制

有效負載訊息的數目有一個限制，即在遇到本機資源限制（例如記憶體）前，`Message Queue` 用戶端執行階段可處理的有效負載訊息數目會受到限制。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）。

用戶流量限制

如果任何用戶傳送至用戶端執行階段的有效負載訊息數超過 `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` 設定值時，才恢復傳送訊息。

以下範例說明這些限制的使用：考量主題用戶的設定值：

```
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%，則下一個批次大小可介於 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%)，則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%)，則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，這會再次造成訊息流量下降。一般來說，除非您有特定的效能或可靠性考量，否則您不需變更 `imqConsumerFlowThreshold` 屬性的預設值。

用戶型的流量控制 (特別是 `imqConsumerFlowLimit`) 為管理用戶端執行階段中記憶體的最佳方法。通常，根據用戶端應用程式，您會知道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。

連線流量限制

然而，在一些用戶端應用程式中，用戶數目可能是不確定的，它取決於一般用戶的選擇。在此情況下，您仍可以使用連線層級流量限制來管理記憶體。

連線層級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` 的值，則會停止透過連線傳送訊息，直到總數目低於連線限制為止。(只有在您將 `imqConnectionFlowLimitEnabled` 設為 `true` 時，才會啟用 `imqConnectionFlowLimit` 屬性。)

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 220 頁的「用戶端無法建立連線」
- 第 224 頁的「連線流量過慢」
- 第 225 頁的「用戶端無法建立訊息產生者」
- 第 226 頁的「訊息產生延遲或過慢」
- 第 229 頁的「儲存訊息」
- 第 233 頁的「訊息伺服器的流量不穩定」
- 第 234 頁的「訊息未送達用戶」
- 第 238 頁的「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。版本編號可讓您確定所用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

此問題有下列徵兆：

- 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。

本節提供下列幾種可能原因：

- 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 儲存區會限制可同時新建連線請求的數目
- 作業系統會限制運作中連線的數目
- 使用者的認證或授權失敗

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，顯示特定用戶端異常多的開啓連線。

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
確認此問題的原因

- 使用 Telnet 連線至代理程式主要連接埠 (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

- 啓動代理程式。

- 修復網路連結問題。

連線服務不在使用中或已暫停

確認此問題的原因

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)，此服務不會顯示在使用中服務清單 (`imq.service.active`) 上。在 SSL 型服務中，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 140 頁的「使用 SSL 型服務」。

- 如果連線服務狀態顯示為 `paused` (暫停中)，會繼續服務 (請參閱第 105 頁的「暫停和繼續連線服務」)。

可於所需連線數目的可執行緒太少

確認此問題的原因

檢查代理程式記錄中是否有以下項目：

```
警告 [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 頁的「執行緒池管理」)。

若要解決此問題

- 如果您使用專用執行緒池模型 (`imq.serviceName.threadpool_model=dedicated`)，則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.serviceName.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.serviceName.threadpool_model=shared`)，則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.serviceName.connectionMonitor_limit`) 和最大執行緒數目 (`imq.serviceName.max_threads`)。因此，若要增加連線數目，請增加執行緒池的大小或增加連線監視限制。
- 可支援的連線數目 (或連線上的流量) 最終會達到輸入 / 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 66 頁的「設定檔案描述元限制」。

確認此問題的原因

檢查代理程式記錄中是否有以下相似項目：開啓的檔案太多。

若要解決此問題

增加檔案描述元限制，如 `ulimit` 線上援助頁所述。

TCP 儲存區會限制可同時新建連線請求的數目

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。)

因為不會經常出現同時有大量的連線請求，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但沒有記錄拒絕連線，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同時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，作業系統限制可能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。

使用者的認證或授權失敗

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權限。

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。

- 如果您使用檔案型使用者儲存庫，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：
錯誤 [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 130 頁的「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，以取得連線服務權限 (請參閱第 137 頁的「連線服務的存取控制」)。

連線流量過慢

此問題有下列徵兆：

- 訊息流量未符合預期速度。
- 支援連線至代理程式的數目不受到限制，如第 220 頁的「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，SSL 型或 HTTP 型協定的速度會較 TCP 的速度慢 (請參閱第 208 頁的圖 11-5)。

確認此問題的原因

如果您使用 SSL 型或 HTTP 型的協定，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，所以除了如第 212 頁的「調校傳輸協定」所述嘗試調校協定外，還可以執行進行一些其他作業。

連線服務協定未調校到最佳化

確認此問題的原因

嘗試調整協定，並查看是否有何差異。

若要解決此問題

嘗試調整協定，如第 212 頁的「調校傳輸協定」所述。

訊息過大導致使用過慢的頻寬 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。
- 將訊息用作要傳送的資料通知，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 199 頁的圖 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 226 頁的「訊息產生延遲或過慢」
- 第 229 頁的「儲存訊息」
- 第 233 頁的「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案。

用戶端無法建立訊息產生者

此問題有下列徵兆：

- 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 116 頁的「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 117 頁的「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 139 頁的「實體目標的存取控制」)。

訊息產生延遲或遲慢

此問題有下列徵兆：

- 傳送永久性訊息時，send 方法不會傳回，且用戶端會阻斷。
- 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存放區
- 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（因此會暫停運作生產型用戶端）。
- REJECT_NEWEST：代理程式拒絕新的永久性訊息。

同樣地，整個代理程式記憶體（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，因為實體目標或整個代理程式的限制未正確設定，所以代理程式會採取愈來愈重要的動作，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，則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，則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。

- 請小心修改實體目標 (或整個代理程式) 上的訊息限制，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，才永遠不會到達整個代理程式的訊息限制。如需更多資訊，請參閱第 215 頁的「代理程式調校」。
- 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 289 頁的表 15-1)。

代理程式無法將永久性訊息儲存到資料存放區

如果代理程式無法存取資料存放區或將永久性訊息寫入資料存放區，則表示生產型用戶端被封鎖。這種情況亦會在目標或整個代理程式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存放區，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

- 在檔案型永久性的情況下，請嘗試增加檔案型資料存放區的磁碟空間。
- 在使用 JDBC 相容資料存放區的情況下，請檢查是否已正確配置 JDBC 型永久性 (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。

代理程式確認逾時延遲

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永久性訊息的接收，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 160 頁的「可靠性和流量控制」)。

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`，檢查 JVM 堆疊中的可用記憶體。

若要解決此問題

調整 JVM (請參閱第 211 頁的「Java 虛擬機器調校」)。

儲存訊息

此問題有下列徵兆：

- 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
```

(備註：`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整個代理程式的限制時，請檢查代理程式記錄中是否有以下項目：警告 [B2011]：儲存來自 JMS 訊息 ... 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。
- 訊息需要超長的時間才能到達用戶。

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 107 頁的「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 289 頁的表 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 119 頁的「清除實體目標」)。
- 限制訊息可保留在記憶體中的時間。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。您可以置換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 299 頁的「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，則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。
- 用戶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：

- 啟動其他用戶用戶端，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 216 頁的「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 289 頁的表 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 119 頁的「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 299 頁的「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，主題使用者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

檢查傳入和傳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

- 最佳化用戶用戶端程式碼。
- 增加用於佇列目標的使用中用戶數目 (請參閱第 216 頁的「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，可能會降低暫停運作用戶用戶端之確認模式中的訊息使用速度，直到代理程式確定用戶端確認為止。

- 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:

JMSEException [C4000]: 資料封包確認失敗

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 216 頁的「用戶端執行階段訊息流量調校」)。

代理程式無法跟上產生訊息的速度

在此情況下, 訊息傳入代理程式的速度, 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、永久存放區或 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

- 升級您電腦或資料存放區的速度。
- 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, 則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 確認或處理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 [未確認] 標頭下列出的訊息數目，是否與目標中的訊息數目相同。[未確認] 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此問題。

訊息伺服器流量不穩定

此問題有下列徵兆：

- 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 記憶體收回 \(廢棄項目收集\)](#)
- [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 (請參閱第 211 頁的「Java 虛擬機器調校」)。

- 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，且因記憶體收回的延遲時間也越長。

確認此問題的原因

監視您電腦的 CPU 使用情況。執行記憶體收回時，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收回的時間。

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收回：

```
-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

此問題有下列徵兆：

- 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](#)

- 用戶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，刪除優先權最低的訊息。
- 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 238 頁的「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 239 頁的「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 資料的平台特定位置」，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

QBrowser 1.0 - localhost:7676

File

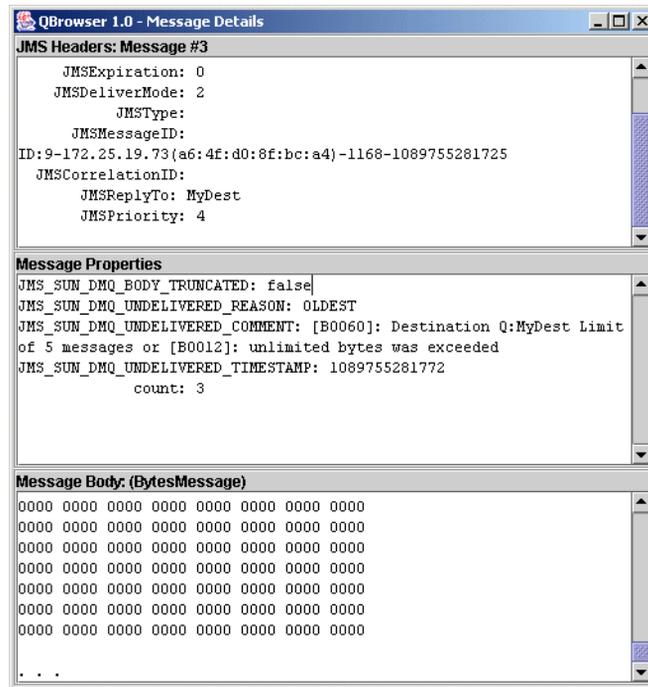
Queue Name: **mq.sys.dmq** Browse

#	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...

在訊息上連按兩下，即可顯示此訊息的詳細資訊。

圖 12-2 QBrowser 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。

解決問題

如第 66 頁的「準備系統資源」中所述，檢查是否在執行時間同步化程式。

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

此問題有下列徵兆：

- 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，會出現類似下面的輸出：

列出指定代理程式上的所有目標：							
主機	主要連接埠						
localhost	7676						
名稱	類型	狀態	產生者	用戶訊息 總數	未確認	平均	大小
MyDest	Queue	執行中	0	0	5	0	1177.0
mq.sys.dmq	Queue	執行中	0	0	35	0	1422.0

成功地列示目標。

此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
- 長期用戶處於非使用中
- 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 資料的平台特定位置」，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\qbrower 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 236 頁的圖 12-1 中顯示的清單。

在任何訊息上連接兩下，即可顯示此訊息的詳細資訊。出現第 237 頁的圖 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 QBrower 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 66 頁的「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 [目前使用中的使用者數] 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。

流量控制會針對使用調整產生，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](#)
- [長期用戶處於非使用中](#)
- [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 241 頁的「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

嘗試下列一種或多種操作：

- 設定目標使用 FLOW_CONTROL 限制運作方式。您可以使用類似下面的指令：

```
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以某資料量傳送。

如果應用程式是依資料量傳送訊息，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，代理程式可能會用盡記憶體而無法處理訊息。

- 考慮在高產生負荷層級期間，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，請如下所示列出使用中作業事件：

```
imqcmd list txn
```

下面是指令的輸出範例：

作業事件 ID	狀態	使用者名稱	訊息數目 / 確認數目	建立時間
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

請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，判斷應用程式使用系統型或用戶端型的確認。如果應用程式使用系統型的確認，請略過本節。

如果應用程式用戶端型的確認 (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：

列出指定代理程式上的所有目標：							
主機	主要連接埠						
localhost	7676						
名稱	類型	狀態	產生者	用戶訊息 總數	未確認	平均	大小
MyDest	Queue	執行中	0	0	5	200	1177.0
mq.sys.dmq	Queue	執行中	0	0	35	0	1422.0

成功地列示目標。

未確認數目代表代理程式傳送出去後等待確認的訊息數目。如果未確認數目變大或增加，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。

長期用戶處於非使用中

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。

發生非預期中的代理程式錯誤

確認此問題的原因

如第 241 頁的「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 「指令行參照」
- 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 「管理物件屬性參照」
- 第 17 章 「JMS 資源介面特性參照」
- 第 18 章 「度量參照」

指令行參照

本章提供使用 Message Queue 指令行管理公用程式的參照資訊。它包含下列各節：

- 第 248 頁的「指令行語法」
- 第 249 頁的「代理程式公用程式」
- 第 252 頁的「指令公用程式」
- 第 259 頁的「物件管理員公用程式」
- 第 261 頁的「資料庫管理員公用程式」
- 第 262 頁的「使用者管理員公用程式」
- 第 263 頁的「服務管理員公用程式」
- 第 264 頁的「金鑰工具公用程式」

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。個別的指令不需要啟動或退出公用程式。

所有指令行公用程式共用以下指令語法：

```
utilityName [subcommand] [commandArgument] [[-optionName [-optionArgument]]...]
```

其中，*utilityName* 是以下其中之一：

- `imqbrokerd` (代理程式公用程式)
- `imqcmd` (指令公用程式)
- `imqobjmgr` (物件管理員公用程式)
- `imqdbmgr` (資料庫管理員公用程式)
- `imqusermgr` (使用者管理員公用程式)
- `imqsvcadm` (服務管理員公用程式)
- `imqkeytool` (金鑰工具公用程式)

子指令與指令層級引數 (若有) 必須處於所有選項及其引數之前；但選項的顯示順序可隨意。以空格分隔所有子指令、指令引數、選項和選項引數。如果選項引數的值包含空格，就必須將整個值括在引號內。(通常將任意成對的屬性值括在引號內最為安全)。

以下指令可啟動預設代理程式，是沒有子指令子句的指令行範例：

```
imqbrokerd
```

此處為更完整的範例：

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

此指令可毀損稱為 `myQueue` 的佇列目標 (目標類型 `q`)。認證在使用者名稱 `admin` 上執行；該指令會提示您輸入密碼。該指令以無訊息模式執行，將不會提示您確認 (`-f` 選項)，亦不會顯示任何輸出 (`-s` 選項)。

代理程式公用程式

代理程式公用程式 (imqbrokerd) 可啟動代理程式。指令行選項可覆寫代理程式配置檔案中的值，但僅限於目前的代理程式階段作業。

表 13-1 顯示 imqbrokerd 指令的選項，以及每個選項覆寫的配置特性 (若有)。

表 13-1 代理程式公用程式選項

選項	被覆寫的特性	說明
-name <i>instanceName</i>	imq.instanceName	代理程式的實例名稱 在單一主機上執行的多個代理程式實例，必須擁有不同的實例名稱。 預設值：imqbroker
-port <i>portNumber</i>	imq.portmapper.port	代理程式的連接埠對映器連接埠號碼 Message Queue 客戶端可使用此連接埠號碼連線至代理程式。在單一主機上執行的多個代理程式實例，必須擁有不同的連接埠對映器連接埠號碼。 預設值：7676
-cluster <i>broker1</i> [[, <i>broker2</i>]...]	imq.cluster.brokerlist	將代理程式連線至叢集 指定的代理程式會與 imq.cluster.brokerlist 特性中的清單合併。每個代理程式引數的格式如下 <i>hostName</i> : <i>portNumber</i> <i>hostName</i> : <i>portNumber</i> 如果省略 <i>hostName</i> ，則預設值為 localhost；如果省略 <i>portNumber</i> ，則預設值為 7676。
-D <i>property=value</i>	實例配置檔案中的對應特性	設定配置特性 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。 注意：請仔細檢查使用此選項設定特性的拼寫與格式。不正確的值會被忽略，不會另行通知或警告。
-reset props	無	重設配置特性 以空白檔案替代代理程式目前的實例配置檔案 (config.properties)；假設所有特性皆為預設值。

表 13-1 代理程式公用程式選項 (續)

選項	移除的特性	說明
-reset store	無	重設永久性資料存放區。 清除資料存放區的所有永久性資料 (包括永久性訊息、長期訂閱與作業事件資訊)，可讓您以乾淨狀態啟動代理程式實例。不使用 -reset 選項重新啟動代理程式實例，可防止在後續的重新啟動時重設永久存放區。 它只清除永久性訊息或長期訂閱，請改用 -reset messages 或 -reset durables。
-reset messages	無	清除資料存放區的永久性訊息
-reset durables	無	清除資料存放區的長期訂閱
-backup fileName	無	將配置變更記錄備份至檔案 ¹ 請參閱第 176 頁的「管理配置變更記錄」，以獲得更多資訊。
-restore fileName	無	從備份檔案復原配置變更記錄 ¹ 此備份檔案必須已使用 -backup 選項事先建立。 請參閱第 176 頁的「管理配置變更記錄」，以獲得更多資訊。
-remove instance	無	移除代理程式實例 ² 刪除此實例配置檔案、記錄檔、永久存放區以及與此實例相關的其他檔案和目錄。
-password keyPassword	imq.keystore.password	SSL 憑證金鑰存放區的密碼 ³
-dbuser userName	imq.persist.jdbc.user	JDBC 型永久資料存放區的使用者名稱
-dbpassword dbPassword	imq.persist.jdbc.password	JDBC 型永久資料存放區的密碼 ³
-ldappassword ldapPassword	imq.user_repository.ldap.password	LDAP 使用者儲存庫的密碼 ³
-passfile filePath	imq.passfile.enabled imq.passfile.dirpath imq.passfile.name	密碼檔案位置 將代理程式的 imq.passfile.enabled 特性設為 true，imq.passfile.dirpath 設為包含密碼檔案的路徑，而 imq.passfile.name 設為檔案名稱。 請參閱第 149 頁的「使用密碼檔案」，以獲得更多資訊。

表 13-1 代理程式公用程式選項 (續)

選項	配置的特性	說明
-shared	imq.jms.threadpool_model	<p>使用共用的執行緒池模型，實作 jms 連線服務。</p> <p>將會在連線之間共用執行緒，以增進所支援的連線數目。</p> <p>將代理程式的 imq.jms.threadpool_model 特性設為 shared。</p>
-javahome path	無	<p>替代 Java 執行階段的位置。</p> <p>預設值：使用安裝在系統上或是隨附於 Message Queue 的執行階段。</p>
-vmargs arg1 [[arg2]...]	無	<p>將引數傳送給 Java 虛擬機器。</p> <p>以空格分隔引數。若要傳送多個引數，或包含空格的引數，請將引數括在引號內。</p> <p>VM 引數只能從指令行傳送；實例配置檔中並沒有相關的配置特性。</p>
-license [licenseName]	無	<p>如果與安裝的 Message Queue 產品版本的預設值不同，要授權載入：</p> <ul style="list-style-type: none"> pe 具備基本功能的 Platform Edition try 具備企業功能的 Platform Edition (90 天試用期) unl Enterprise Edition <p>如果未指定授權名稱，此選項會列出系統上已安裝的所有授權。</p>
-upgrade-store-nobackup	無	<p>從不相容的版本升級至 Message Queue 3.5 或 3.5 SPx 時，自動移除舊的資料存放區²。</p> <p>請參閱「Message Queue Installation Guide」，以取得更多資訊。</p>
-force	無	<p>執行動作時無需使用者確認。</p> <p>此選項僅適用於通常會要求確認的 -remove instance 與 -upgrade-store-nobackup 選項。</p>
-loglevel level	imq.broker.log.level	<p>記錄層級：</p> <ul style="list-style-type: none"> NONE ERROR WARNING INFO <p>預設值：INFO</p>

表 13-1 代理程式公用程式選項 (續)

選項	秘密字的特性	說明
-metrics <i>interval</i>	imq.metrics.interval	代理程式度量的記錄間隔 (以秒為單位)
-tty	imq.log.console.output	將所有訊息記錄至主控台 將代理程式的 imq.log.console.output 特性設為 ALL。 如果未指定，將指為記錄錯誤與警告訊息。
-s -silent	imq.log.console.output	無訊息模式 (不記錄至主控台) 將代理程式的 imq.log.console.output 特性設為 NONE。
-version	無	顯示版本資訊 ⁴
-h -help	無	顯示用法說明 ⁴

1. 此選項僅適用於代理程式叢集。
2. 除非也指定了 -force，否則此選項會要求使用者確認。
3. 此選項已經停用而且會永遠移除。改為完全省略密碼 (如此指令就會以互動方式提示您輸入密碼)，或使用 -passfile 選項，指定包含密碼的密碼檔案。
4. 忽略指令行上指定的任何其他選項。

指令公用程式

指令公用程式 (imqcmd) 可用於管理代理程式、連線服務、連線、實體目標、長期訂閱與作業事件。

所有 imqcmd 指令都必須包含子指令 (除了使用 -v 或 -h 選項以顯示產品版本資訊或用法說明以外)。此處會列出可能的子指令，還會在以下各節詳細地說明。在所有情況下，如果子指令接受代理程式位址 (-b 選項)，而且未指定主機名稱或連接埠號，則預設會假設為 localhost 與 7676。

代理程式管理

shutdown bkr	關閉代理程式
restart bkr	重新啟動代理程式
pause bkr	暫停代理程式
resume bkr	繼續代理程式
update bkr	設定代理程式特性
reload cls	重新載入叢集配置

query bkr	列出代理程式特性值
metrics bkr	顯示代理程式度量

連線明碼管理

pause svc	暫停連線服務
resume svc	繼續連線服務
update svc	設定連線服務特性
list svc	列出代理程式上可用的連線服務
query svc	列出連線服務特性值
metrics svc	顯示連線服務度量

連線管理

list cxn	列出代理程式上的連線
query cxn	顯示連線資訊

實體目標管理

create dst	建立實體目標
destroy dst	銷毀實體目標
pause dst	暫停將訊息傳送給實體目標
resume dst	繼續將訊息傳送給實體目標
update dst	設定實體目標特性
purge dst	清除來自實體目標的所有訊息
compact dst	壓縮實體目標
list dst	列出實體目標
query dst	列出實體目標特性值
metrics dst	顯示實體目標度量

長期訂閱管理

destroy dur	銷毀長期訂閱
purge dur	清除長期訂閱的所有訊息
list dur	列出主題的長期訂閱

作業事件管理

commit txn	確定作業事件
rollback txn	回復作業事件
list txn	列出代理程式追蹤的作業事件
query txn	顯示作業事件資訊

代理程式管理

要使用代理程式公用程式 (imqbrokerd) 才能啟動代理程式，而不是使用指令公用程式。啟動代理程式之後，就可以使用表 13-2 中列出的 imqcmd 子指令，來進行管理與控制。

表 13-2 代理程式管理的指令公用程式子指令

語法	說明
shutdown bkr [-b <i>hostName:portNumber</i>]	關閉代理程式
restart bkr [-b <i>hostName:portNumber</i>]	重新啟動代理程式 關閉代理程式，然後使用原來啟動時指定的相戶選項加以重新啟動。
pause bkr [-b <i>hostName:portNumber</i>]	暫停代理程式 請參閱第 99 頁的「暫停代理程式」，以取得更多資訊。
resume bkr [-b <i>hostName:portNumber</i>]	繼續代理程式
update bkr [-b <i>hostName:portNumber</i>] -o <i>property1=value1</i> [[-o <i>property2=value2</i>] ...]	設定代理程式特性 請參閱第 14 章「代理程式特性參照」，以取得代理程式特性的相關資訊。
reload cls	重新載入叢集配置 ¹ 強制更新所有永久性資訊以保持最新的狀態。
query bkr -b <i>hostName:portNumber</i>	列出代理程式特性值 還會列出所有連線到叢集中指定代理程式的執行中代理程式。
metrics bkr [-b <i>hostName:portNumber</i>] [-m <i>metricType</i>] [-int <i>interval</i>] [-msp <i>numSamples</i>]	顯示代理程式度量 -m 選項可指定要顯示的度量類型： ttl 傳入和傳出代理程式的訊息與封包。 rts 代理程式的訊息與封包每秒傳入和傳出的速率 cxn 連線、虛擬記憶體數量與執行緒 預設值：ttl。 -int 選項可指定顯示度量的間隔(以秒為單位)。預設值：5。 -msp 選項可指定要顯示的範例數目。預設值：不限制(無限)。

1. 此選項僅適用於代理程式叢集。

連線服務管理

表 13-3 列出可管理連線服務的 `imqcmd` 子指令。

表 13-3 連線服務管理的指令公用程式子指令

語法	說明
<code>pause svc -n serviceName</code> <code>[-b hostName:portNumber]</code>	暫停連線服務 無法暫停 <code>admin</code> 連線服務。
<code>resume svc -n serviceName</code> <code>[-b hostName:portNumber]</code>	繼續連線服務
<code>update svc -n serviceName</code> <code>[-b hostName:portNumber]</code> <code>-o property1=value1</code> <code>[[-o property2=value2] ...]</code>	設定連線服務特性 請參閱第 266 頁的「連線特性」，以取得連線服務特性的相關資訊。
<code>list svc [-b hostName:portNumber]</code>	列出代理程式上可用的連線服務
<code>query svc -n serviceName</code> <code>[-b hostName:portNumber]</code>	列出連線服務特性值
<code>metrics svc -n serviceName</code> <code>[-b hostName:portNumber]</code> <code>[-m metricType]</code> <code>[-int interval]</code> <code>[-msp numSamples]</code>	顯示連線服務度量 -m 選項可指定要顯示的度量類型： <code>ttl</code> 利用指定的連線服務傳入和傳出代理程式的訊息與封包 <code>rts</code> 代理程式的訊息與封包利用指定的連線服務，每秒傳入和傳出的速率 <code>cxn</code> 連線、虛擬記憶體堆疊與執行緒 預設值： <code>ttl</code> 。 -int 選項可指定顯示度量的間隔（以秒為單位）。預設值： <code>5</code> 。 -msp 選項可指定要顯示的範例數目。預設值：不限制（無限）。

連線管理

表 13-4 列出可管理連線的 `imqcmd` 子指令。

表 13-4 連線服務管理的指令公用程式子指令

語法	說明
<code>list cxn [-svn serviceName] [-b hostName:portNumber]</code>	列出代理程式上的連線 列出指定連線服務在代理程式上的所有連線。如果未指定連線服務，則會列出所有連線。
<code>query cxn -n connectionID [-b hostName:portNumber]</code>	顯示連線資訊

實體目標管理

表 13-5 列出可管理實體目標的 `mqcmd` 子指令。在所有情況下，`-t` (目標類型) 選項可使用下列兩項值之一：

- q 佇列目標
- t 主題目標

表 13-5 實體目標管理的指令公用程式子指令

語法	說明
<code>create dst -t destType -n destName [-o property1=value1] [[-o property2=value2] ...]</code>	建立實體目標 ¹ 目標名稱 <code>destName</code> 可能僅包含字母數字字元 (無空格)，而且必須以字母字元或底線 (<code>_</code>) 或錢幣符號 (<code>\$</code>) 開頭。而可能不會以 <code>mq</code> 字元開頭。
<code>destroy dst -t destType -n destName</code>	銷毀實體目標 ¹ 此作業不適用於由系統建立的目標，例如，佇列的訊息佇列。
<code>pause dst [-t destType -n destName] [-pst pauseType]</code>	暫停將訊息傳送給實體目標 暫停將訊息傳送給利用 <code>-t</code> 與 <code>-n</code> 選項指定的實體目標。如果未指定這些選項，就會暫停所有目標。 <code>pst</code> 選項可指定要暫停的訊息傳送類型： <ul style="list-style-type: none"> CONSUMERS 暫停傳送給訊息用戶 PRODUCERS 暫停傳送給訊息產生者 ALL 暫停所有訊息傳送 預設值：ALL
<code>resume dst [-t destType -n destName]</code>	繼續將訊息傳送給實體目標 繼續將訊息傳送給利用 <code>-t</code> 與 <code>-n</code> 選項指定的實體目標。如果未指定這些選項，就會繼續所有目標。

表 13-5 實體目標管理的指令公用程式子指令 (續)

語法	說明
update dst -t <i>destType</i> -n <i>destName</i> -o <i>property1=value1</i> [[-o <i>property2=value2</i>] ...]	設定實體目標特性 請參閱第 15 章「實體目標特性參照」，以取得實體目標特性的相關資訊。
purge dst -t <i>destType</i> -n <i>destName</i>	清除來自實體目標的所有訊息
compact dst [-t <i>destType</i> -n <i>destName</i>]	壓縮實體目標 壓縮由 -t 與 -n 選項所指定之實體目標的檔案型永久資料存放區。如果未指定這些選項，就會壓縮所有目標。 必須在壓縮目標前暫停目標。
list dst [-t <i>destType</i>] [-tmp]	列出實體目標 列出由 -t 選項指定的所有實體目標類型。如果未指定目標類型，就會列出佇列與主題目標。如果已指定 -tmp 選項，也會列出暫存目標。
query dst -t <i>destType</i> -n <i>destName</i>	列出實體目標特性值
metrics dst -t <i>destType</i> -n <i>destName</i> [-m <i>metricType</i>] [-int <i>interval</i>] [-msp <i>numSamples</i>]	顯示實體目標度量 -m 選項可指定要顯示的度量類型： ttl 傳入和傳出目標以及常駐於記憶體中的訊息與封包。 rts 代理程式的訊息與封包每秒傳入和傳出的速率，以及其他速率資訊 con 與訊息用戶相關的度量 dsk 磁碟使用率 預設值：ttl。 -int 選項可指定顯示度量的間隔(以秒為單位)。預設值：5。 -msp 選項可指定要顯示的範例數目。預設值：不限制(無限)。

1. 不能在主代理程式暫時無法使用的代理程式叢集上，執行此作業。

長期訂閱管理

表 13-6 列出可管理長期訂閱的 imqcmd 子指令。

表 13-6 長期訂閱管理的指令公用程式子指令

語法	說明
<code>destroy dur -c clientID -n subscriberName</code>	銷毀長期訂閱 ¹
<code>purge dur -c clientID -n subscriberName</code>	清除長期訂閱的所有訊息
<code>list dur -d topicName</code>	列出主題的長期訂閱

1. 不能在主代理程式暫時無法使用的代理程式叢集上，執行此作業。

作業事件管理

表 13-7 列出可管理作業事件的 `imqcmd` 子指令。

表 13-7 作業事件管理的指令公用程式子指令

語法	說明
<code>commit txn -n transactionID</code>	確定作業事件
<code>rollback txn -n transactionID</code>	回復作業事件
<code>list txn</code>	列出代理程式追蹤的作業事件
<code>query txn -n transactionID</code>	顯示作業事件資訊

一般指令公用程式選項

表 13-8 中列出的其他選項，可套用至 `imqcmd` 指令的任何子指令。

表 13-8 一般指令公用程式選項

選項	說明
<code>-secure</code>	以 <code>ssladmin</code> 連線服務，完全連線至代理程式
<code>-u userName</code>	認證的使用者名稱 如果省略此選項，指令公用程式就會以互動方式提示您輸入。
<code>-p password</code>	認證的密碼 ¹
<code>-passfile path</code>	密碼檔案位置 請參閱第 149 頁的「使用密碼檔案」，以取得更多資訊。

表 13-8 一般指令公用程式選項 (續)

選項	說明
-rtm <i>timeoutInterval</i>	初始逾時間隔 (以秒為單位) 重試請求之前，指令公用程式等待代理程式回覆的初始時間長度。每次後續重試都會使逾時間隔，而此間隔為此初始間隔的倍數。 預設值：10。
-rtr <i>numRetries</i>	代理程式請求逾時之後的重試次數 預設值：5。
-javahome <i>path</i>	替代 Java 執行階段的位置 預設值：使用安裝在系統上或是隨附於 Message Queue 的執行階段。
-f	執行動作而無需使用者確認
-s	無訊息模式 (不顯示輸出)
-v	顯示版本資訊 ^{2、3}
-h	顯示用法說明 ^{2、3}
-H	顯示延伸用法說明，包括屬性清單與範例 ^{2、3}

1. 此選項已經停用而且會永遠移除。改為完全省略密碼 (如此指令就會互動提示要輸入密碼) 或使用 `-passfile` 選項以指定包含密碼的密碼檔案。
2. 忽略指令行上指定的任何其他選項。
3. 此選項不需要使用者名稱與密碼。

物件管理員公用程式

物件管理員公用程式 (`imqobjmgr`) 可建立和管理 Message Queue 管理物件。表 13-9 列出了可用的子指令。

表 13-9 物件管理員子指令

子指令	說明
add	將管理物件新增至物件存放區
delete	刪除物件存放區中的管理物件
list	列出物件存放區中的管理物件
query	顯示管理物件資訊
update	修改管理物件

表 13-10 列出 `imgobjmgr` 指令的選項。

表 13-10 物件管理員選項

選項	說明
<code>-l lookupName</code>	管理物件的 JNDI 查詢名稱
<code>-j attribute=value</code>	JNDI 物件存放區的屬性 (請參閱第 154 頁的「物件存放區」)
<code>-t objectType</code>	管理物件的類型： <ul style="list-style-type: none"> <code>q</code> 佇列目標 <code>t</code> 主題目標 <code>cf</code> 連線工廠 <code>qf</code> 佇列連線工廠 <code>tf</code> 主題連線工廠 <code>xcf</code> 分散式作業事件的連線工廠 <code>xqf</code> 分散式作業事件的佇列連線工廠 <code>xtf</code> 分散式作業事件的主題連線工廠 <code>e</code> SOAP 端點 (請參閱「Message Queue Developer's Guide for Java Clients」)
<code>-o attribute=value</code>	管理物件的屬性 (請參閱第 156 頁的「管理物件屬性」與第 16 章「管理物件屬性參照」)
<code>-r readOnlyState</code>	管理物件是否為唯讀？ 如果為 <code>true</code> ，則 J 端就無法修改物件的屬性。預設值： <code>false</code> 。
<code>-i fileName</code>	包含所有或部分子指令子集的指令檔名稱
<code>-pre</code>	無需執行指令即能預覽結果 此選項對檢查預設屬性的值很有用。
<code>-javahome path</code>	替代 Java 執行階段的位置 預設值：使用安裝在系統上或是隨附於 Message Queue 的執行階段。
<code>-f</code>	執行動作，無需使用者確認
<code>-s</code>	無訊息模式 (不顯示輸出)
<code>-v</code>	顯示版本資訊 ¹
<code>-h</code>	顯示用法說明 ¹
<code>-H</code>	顯示延伸用法說明，包括屬性清單與範例 ¹

1. 忽略指令行上指定的任何其他選項。

資料庫管理員公用程式

資料庫管理員公用程式 (imqdbmgr) 可設定 JDBC 型永久資料存放區的資料庫模式。您也可以用它刪除已損毀的 Message Queue 資料庫表格，或變更資料存放區。

表 13-11 列出了可用子指令。

表 13-11 資料庫管理員子指令

子指令	說明
create all	建立新資料庫與永久存放區模式。 用於內嵌式資料庫系統。— 定要指定代理程式特性 imq.persist.jdbc.createdburl。
create tbl	建立現存資料庫的永久存放區模式。 用於外部資料庫系統。
delete tbl	刪除目前永久存放區的 Message Queue 資料庫表格。
delete oldtbl	刪除舊版永久存放區的 Message Queue 資料庫表格。 在自動將永久存放區遷移到 Message Queue 目前版本後使用。
recreate tbl	重新建立永久存放區模式。 刪除目前永久存放區所存現存的 Message Queue 資料庫表格，然後重新建立模式。
reset lck	重設永久性存放區鎖定狀態。 重設鎖定狀態，以便其他程序可以使用永久存放區資料庫。

表 13-12 列出了 imqdbmgr 指令的選項。

表 13-12 資料庫管理員選項

選項	說明
-b <i>instanceName</i>	代理程式的實例名稱
-D <i>property=value</i>	設定代理程式配置特性 請參閱第 271 頁的「永久性特性」，以取得與永久性相關的代理程式配置特性之資訊。 注意： 請仔細檢查使用此選項設定的特性的拼字與格式。不正確的值會被忽略，不會另行通知或警告。
-u <i>name</i>	認證的使用者名稱
-p <i>password</i>	認證的密碼

表 13-12 資料庫管理員選項 (續)

選項	說明
-passfile <i>path</i>	密碼檔案位置 請參閱第 149 頁的「使用密碼檔案」，以獲得更多資訊。
-v	顯示版本資訊 ²
-h	顯示用法說明 ²

1. 此選項已經停用而且會永遠移除。改為完全省略密碼 (如此指令就會互動地提示要輸入密碼) 或使用 -passfile 選項以指定包含密碼的密碼檔案。
2. 忽略指令行上指定的任何其他選項。

使用者管理員公用程式

使用者管理員公用程式 (imqusermgr) 可用以寫入或編輯平面檔案使用者儲存庫。此公用程式必須在安裝代理程式的同一台主機上執行；如果代理程式特有的使用者儲存庫尚不存在，您必須先啟動相對的代理程式實例才能建立。您也需要適當的權限以寫入儲存庫：在 Solaris 或 Linux 平台上，這表示您必須作為超級使用者或最初建立該代理程式實例的使用者。

表 13-13 列出可與 imqusermgr 指令一起使用的子指令。在所有情況下，-i 選項可指定指令要對其套用使用者儲存庫的代理程式實例名稱；如果未指定，就會假設為預設名稱 imqbroker。

表 13-13 使用者管理員子指令

語法	說明
add [-i <i>instanceName</i>] -u <i>userName</i> -p <i>password</i> [-g <i>group</i>]	將使用者與密碼新增至儲存庫 可選擇的 -g 選項，可指定要將這位使用者指定給哪個群組： admin user anonymous
delete [-i <i>instanceName</i>] -u <i>userName</i>	刪除儲存庫的使用者
update [-i <i>instanceName</i>] -u <i>userName</i> -p <i>password</i> [-a <i>activeState</i>]	設定使用者的密碼或使用者狀態 (或兩者) -a 選項可以是指定使用者為使用中 (true) 或非使用中 (false) 的布林值。預設值：true。
update [-i <i>instanceName</i>] -u <i>userName</i> -a <i>activeState</i> [-p <i>password</i>]	

表 13-13 使用者管理員子指令 (續)

語法	說明
list [-i instanceName] [-u userName]	顯示使用者資訊 如果未指定使用者名稱，就會列出儲存庫中的所有使用者。

此外，表 13-14 中所列出的選項，可套用至 imqusermgr 指令的任何子指令。

表 13-14 一般使用者管理員選項

選項	說明
-f	執行動作時，無需使用者確認使用者執行動作。
-s	無訊息模式 (不顯示輸出)
-v	顯示版本資訊 ¹
-h	顯示用法說明 ¹

1. 忽略指令行上指定的任何其他選項。

服務管理員公用程式

服務管理員公用程式 (imqsvcadmin) 可將代理程式安裝為 Windows 服務。表 13-15 列出了可用的子指令。

表 13-15 服務管理員子指令

子指令	說明
install	安裝服務
remove	移除服務
query	顯示啟動選項 啟動選項可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值。

表 13-16 列出了 imqsvcadmin 指令的選項。

表 13-16 服務管理員選項

選項	說明
-jvahome <i>path</i>	替代 Java 執行階段的位置 預設值：使程序裝在系統上或是隨附於 Message Queue 的執行階段。
-jrehome <i>path</i>	替代 Java 執行階段環境 (JRE) 的位置
-vmargs <i>arg1</i> [<i>arg2</i>] ...]	要傳送給執行代理程式服務的 Java 虛擬機器的其他引數 ¹ 範例： imqsvcadm install -vmargs "-Xms16m -Xmx128m"
-args <i>arg1</i> [<i>arg2</i>] ...]	要傳送給代理程式服務的其他指令行引數 ¹ 範例： imqsvcadm install -args "-passfile d:\imqpassfile" 請參閱第 249 頁的「代理程式公用程式」，以取得有關代理程式指令行引數的資訊。
-h	顯示用法說明 ²

1. 也可以在服務的 [內容] 視窗中，於 [一般] 標籤下方的 [啟動參數] 欄位中，指定這些引數 (可透過 Windows [系統管理工具] 控制面板中的 [服務] 工具取得)。
2. 會忽略指令行上指定的任何其他選項。

您使用 -javahome、-vmargs 和 -args 選項指定的任何資訊，都會儲存在 Windows 登錄檔中的金鑰 JREHome、JVMArgs 和 ServiceArgs 下，路徑為

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\iMQ_Broker\Parameters
```

金鑰工具公用程式

金鑰工具公用程式 (imqkeytool) 會產生自身簽名憑證，此憑證可用於 ssljms、ssladmin 或叢集連線服務。語法為

```
imqkeytool -broker
```

若為 UNIX 系統，會需要從 superuser (超級使用者) 帳號執行公用程式。

代理程式特性參照

本章提供有關訊息代理程式配置特性的參照資訊。它包含下列各節：

- [第 266 頁的「連線特性」](#)
- [第 267 頁的「路由特性」](#)
- [第 271 頁的「永久性特性」](#)
- [第 276 頁的「安全性特性」](#)
- [第 280 頁的「檢視特性」](#)
- [第 283 頁的「叢集配置特性」](#)
- [第 284 頁的「按字母順序排列的代理程式特性清單」](#)

連線特性

表 14-1 列出與連線服務相關的代理程式特性。

表 14-1 代理程式連線特性

特性	類別	預設	說明
<code>imq.service.activelist</code>	字串	<code>jms, admin</code>	以逗號分隔要在啟動代理程式時啟動的連線服務清單
<code>imq.hostname</code>	字串	所有可用 IP 位址	所有連線服務的預設主機名稱或 IP 位址
<code>imq.portmapper.hostname</code>	字串	無	連接埠對映器的主機名稱或 IP 位址 如果指定，會覆蓋 <code>imq.hostname</code>
<code>imq.portmapper.port¹</code>	整數	7676	連接埠對映器的連接埠號碼 注意： 如果在同一台主機上執行多個代理程式實例，則每一個實例都必須指定唯一的連接埠對映器連接埠。
<code>imq.serviceName.protocolType.hostname</code>	字串	無	連線服務的主機名稱或 IP 位址 ² 如果指定，會覆蓋所指定連線服務的 <code>imq.hostname</code>
<code>imq.serviceName.protocolType.port</code>	整數	0	連線服務的連接埠號碼 ² 0 值會指定連接埠對映器動態配置的連接埠號碼。
<code>imq.portmapper.backlog</code>	整數	50	作業系統儲存區中的最大擺置連接埠對映器請求數
<code>imq.serviceName.threadpool_model</code>	字串	<code>dedicated</code>	執行緒池管理的執行緒模型： <code>dedicated</code> 每個連線有兩個專屬執行緒，一個供傳送訊息使用，另一個供接收訊息使用 <code>shared</code> ³ 傳送或接收訊息時，由共用執行緒處理的連線 專屬模型會限制所能支援的連線數目，但可以提供較高的效能；共用模型可增加連線數目，但因為需要額外的耗用時間進行執行緒管理，所以效能較低。
<code>imq.serviceName.min_threads</code>	整數	<code>jms: 10</code> <code>ssljms: 10</code> <code>httpjms: 10</code> <code>httpsjms: 10</code> <code>admin: 4</code> <code>ssladmin: 4</code>	連線服務的執行緒池中所需的最小執行緒數 當可用的執行緒數超過此臨界值時，將會在執行緒成為自由執行緒時將其關閉，直到達到最小數目的臨界值為止。 如其所示，預設值因連線服務而異。

表 14-1 代理程式連線特性 (續)

特性	類別	預設	說明
<code>imq.serviceName.max_threads</code>	整數	jms: 1000 ssljms: 500 httpjms: 500 httpsjms: 500 admin: 10 ssladmin: 10	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 <code>min_threads</code> 的值。 如其所示，預設值因連線服務而異。
<code>imq.shared.connectionMonitor_limit</code>	整數	Solaris: 512 Linux: 512 Windows: 64	由分散執行緒監視的最大連線數目 ⁴ 系統配置了足夠的分散執行緒以監視所有連線。此特性的值愈小，將執行緒指定給使用中連線的速度愈快。-1 值表示不限制每個執行緒的連線數目。 如其所示，預設值因作業系統平台而異。
<code>imq.ping.interval</code>	整數	120	測試客戶端與代理程式間連線的間隔 (以秒為單位) 0 或 -1 值可停用有規律的連線測試。

1. 可與 `imqcmd update bkr` 指令一起使用

2. 僅 `jms`、`ssljms`、`admin` 與 `ssladmin` 服務；請參閱附錄 C「HTTP/HTTPS 支援」以取得更多有關配置 `httpjms` 與 `httpsjms` 服務的資訊

3. 僅限 `jms` 與 `admin` 服務

4. 僅限共用執行緒模型

路由特性

表 14-2 列出與路由服務相關的代理程式特性。可配置自動建立目標的特性列於表 14-3。

表 14-2 代理程式路由特性

特性	類別	預設	說明
<code>imq.system.max_count¹</code>	整數	-1	代理程式保留訊息的最大數目 -1 值表示不限制訊息數目。

表 14-2 代理程式路由特性 (續)

特性	類別	預設	說明
<code>imq.system.max_size¹</code>	字串	-1	代理程式保留訊息的最大總容量 此值可能會使下列字尾，以位元組、千位元組或百萬位元組表示： b 位元組 k 千位元組(1024 位元組) m 百萬位元組(1024 x 1024= 1,048,576 位元組) 無字尾的值表示以位元組為單位；-1 值表示不限制訊息容量。 範例： 1600 1600 位元組 1600b 1600 位元組 16k 16 千位元組(= 16,384 位元組) 16m 16 百萬位元組(= 16,777,216 位元組) -1 不限制
<code>imq.message.max_size¹</code>	字串	70m	單一訊息的最大容量 語法與 <code>imq.system.max_size</code> 相同 (請參閱前述)。
<code>imq.message.expiration.interval</code>	整數	60	收中逾期訊息的間隔 (以秒為單位)
<code>imq.resourceState.threshold</code>	整數	green: 0 yellow: 80 orange: 90 red: 98	啟動每種記憶體資源狀態時的記憶體容量百分比 (其中 <code>resourceState</code> 為 green、yellow、orange 或 red)
<code>imq.resourceState.count</code>	整數	green: 5000 yellow: 500 orange: 50 red: 0	檢查記憶體是否已達到資源狀態臨界值之前，批次中可存的最大傳送訊息數目 (其中 <code>resourceState</code> 是 green、yellow、orange 或 red) 系統記憶體變得逐漸不足時，此限制會減低訊息產生者。
<code>imq.destination.DMQ.truncateBody¹</code>	布林	false	儲存在停排訊息佇列之前，先移除訊息內容？ 如果為 true，將只會儲存訊息標頭與特性資料。
<code>imq.transaction.autorollback</code>	布林	false	要在代理程式啟動時，自動回轉準備狀態中遺留下的分數式作業事件？ 如果為 false，您必須使用指令公用程式 (<code>imqcmd</code>)，手動確定或回轉作業事件。

1. 可與 `imqcmd update bkr` 指令一起使用

表 14-3 自動建立目標的代理程式特性

特性	類型	預設	說明
imq.autocreate.queue ^{1,2}	布林	true	允許自動建立佇列目標？
imq.autocreate.topic ³	布林	true	允許自動建立主題目標？
imq.autocreate.destination.maxNumMsgs	整數	100000	未使用訊息的最大數目 -1 值表示不限制訊息數目。
imq.autocreate.destination.maxBytesPerMsg	字串	10k	任何單一訊息的最大容量 (以位元組為單位) 此值可能會使用下列字尾，以位元組、千位元組或百萬位元組表示： b 位元組 k 千位元組 (1024 位元組) m 百萬位元組 (1024 x 1024 = 1,048,576 位元組) 無字尾的值會以位元組表示；-1 值表示不限制訊息容量。 範例： 1600 1600 位元組 1600b 1600 位元組 16k 16 千位元組 (= 16,384 位元組) 16m 16 百萬位元組 (= 16,777,216 位元組) -1 不限制
imq.autocreate.destination.maxTotalMsgBytes	字串	10m	未使用訊息的最大總記憶體容量 (以位元組為單位) 語法與 imq.autocreate.destination.maxBytesPerMsg 相同 (請參閱前述)。

表 14-3 自動建立目標的代理程式特性 (續)

特性	類型	預設	說明
imq.autocreate.destination.limitBehavior	字串	REJECT_NEWEST	<p>達到記憶體限制臨界值時的代理程式運作方式：</p> <p>FLOW_CONTROL 減緩產生者速度</p> <p>REMOVE_OLDEST 捨棄最舊的訊息</p> <p>REMOVE_LOW_PRIORITY 根據訊息存在時間， 捨棄優先權最低的 訊息；不會對生產型客戶端 傳送 通知</p> <p>REJECT_NEWEST 拒絕最新的訊息； 只有當訊息為永久性時， 才通知生產型客戶端 發生異常</p> <p>如果值為 REMOVE_OLDEST 或 REMOVE_LOW_PRIORITY，而且 imq.autocreate.destination. useDMQ 特性為 true，則超過限制的 訊息就會移至停用的訊息佇列。</p>
imq.autocreate.destination.maxNumProducers	整數	100	<p>目標的最大訊息產生者數目</p> <p>達到限制時，將無法建立新的產生者。 -1 值表示不限制產生者數目。</p>
imq.autocreate.queue.maxNumActiveConsumers ²	整數	1	<p>來自佇列目標的負載平衡佇列傳送 隊，最大的使用中訊息客戶數目</p> <p>-1 值表示不限制客戶數目。</p>
imq.autocreate.queue.maxNumBackupConsumers ²	整數	0	<p>來自佇列目標的負載平衡佇列傳送 隊，最大的備份訊息客戶數目</p> <p>-1 值表示不限制客戶數目。</p>
imq.autocreate.queue.consumerFlowLimit ²	整數	1000	<p>單一批次中傳送給佇列客戶的最大 訊息數目</p> <p>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 客戶的佇列訊息初始數目。目標客戶 可以在連線上指定較低值，覆蓋這 項限制。</p> <p>-1 值表示不限制客戶數目。</p>

表 14-3 自動建立目標的代理程式特性 (續)

特性	類型	預設	說明
imq.autocreate.topic.consumerFlowLimit ³	整數	1000	單一 批次中傳送給主 題冊戶的最大 訊息數目 -1 值表示不限制冊戶數目。
imq.autocreate.destination.isLocalOnly	布林	false	僅限本機傳送？ 此特性僅適用於代理程式叢集，而 且目標一經建立之後便無法變更。 如果為 true，則其他代理程式上 不會重複實體目標，且限制目標僅 會傳送訊息給本機冊戶（會連線至 建立目標的代理程式）。
imq.autocreate.queue.localDeliveryPreferred ²	布林	false	偏好本機傳送？ 此特性僅適用於代理程式叢集中的 負載平衡佇列傳送。如果為 true， 只在當本機代理程式上沒有冊戶 時，才會將訊息傳送給遠端冊戶； 目標絕不可僅限於本機傳送 (imq.autocreate.destination.isLocalOnly 必須為 false)。
imq.autocreate.destination.useDMQ	布林	true	要將停冊 訊息傳送給停冊 訊息佇 列？ 如果為 false，停冊 訊息將會被捨 棄。

1. 可與 imqcmd update bkr 指令一起使用
2. 僅限佇列目標
3. 僅限主題目標

永久性特性

Message Queue 支援永久性資料存放區的檔案型與 JDBC 型模型。代理程式特性 imq.persist.store (表 14-4) 可指定要使用哪種模型。以下各節描述兩種模型的代理程式配置特性。

表 14-4 全域代理程式永久性特性

特性	類型	預設	說明
imq.persist.store	字串	file	永久性資料存放區的模型： file 檔案型永久性 jdbc JDBC 型永久性

檔案型永久性

表 14-5 列出與檔案型永久性相關的代理程式特性。

表 14-5 檔案型永久性的代理程式特性

特性	類別	預設	說明
<code>imq.persist.file.message.max_record_size</code>	字串	1m	<p>新增至訊息存放區檔案的最大訊息容量</p> <p>超過此容量的任何訊息，都會以個別的檔案名自儲存。</p> <p>此值可能會使下列字尾，以位元組、千位元組或百萬位元組表示：</p> <ul style="list-style-type: none"> b 位元組 k 千位元組 (1024 位元組) m 百萬位元組 (1024 x 1024 = 1,048,576 位元組) <p>無字尾的值表示以位元組為單位。</p> <p>範例：</p> <ul style="list-style-type: none"> 1600 1600 位元組 1600b 1600 位元組 16k 16 千位元組 (= 16,384 位元組) 16m 16 百萬位元組 (= 16,777,216 位元組)
<code>imq.persist.file.destination.message.filepool.limit</code>	整數	100	<p>目標檔案池中可供重新使用的最大可掛檔案數目。</p> <p>超過此限制的已掛檔案將會被刪除。代理程式將依需要建立與刪除超過此限制的其他檔案。</p> <p>此限制愈高，代理程式處理永久性資料的速度就愈快。</p>
<code>imq.persist.file.message.filepool.cleanratio</code>	整數	0	<p>已掛檔案池中要維持在清除 (空白) 狀態的檔案百分比</p> <p>此值越高，檔案池所需的磁碟空間就愈少，但在作業期間清除檔案所需的耗掛時間會愈長。</p>

表 14-5 檔案型永久性的代理程式特性 (續)

特性	類別	預設	說明
imq.persist.file.message.cleanup	布林	false	要在關機時清除目錄檔池中的檔案？ 將此特性設為 true 可節省磁碟空間以供檔案存放區使用，但會減緩代理程式關閉的速度。
imq.persist.file.sync.enabled	布林	false	要同步實體存放裝置的內部記憶體狀態？ 將此特性設為 true 可避免因系統當機而造成資料遺失，但效能會降低。 備註： 如果 Message Queue 執行 Sun Cluster 和 Sun Cluster Data Service，請在所有叢集節點上將代理程式的此特性設定為 true。

JDBC 型永久性

表 14-6 列出與 JDBC 型永久性相關的代理程式特性。所示範例為 DataMirror Mobile Solutions, Inc. 資料庫產品的 PointBase® 系列。

表 14-6 JDBC 型永久性的代理程式特性

特性	說明	類別
imq.persist.jdbc.brokerid	<p>(可選擇) 代理程式實例識別碼</p> <p>識別碼必須為字母數字串，其長度不能超過 $n - 12$，其中 n 為資料庫所允許的最長表名名稱長度。</p> <p>在多個代理程式實例使用相同的資料庫作為永久性資料存放區的情況下，此識別碼會附加到資料庫表名，使這些資料庫表名均是唯一的。由於嵌入式資料庫僅會儲存一個代理程式實例的資料，因此通常不需要此屬性。</p>	PointBase 的嵌入式版本不需要

表 14-6 JDBC 型永久性的代理程式特性 (續)

特性	說明	類別
imq.persist.jdbc.driver	可供連線至資料庫的 JDBC 驅動程式 Java 類別名稱	com.pointbase.jdbc.jdbcUniversalDriver
imq.persist.jdbc.opendburl	可供開啓現存資料庫連線的 URL	jdbc:pointbase:embedded: <i>dbName</i> ; database.home= .../instances/ <i>instanceName</i> /dbstore
imq.persist.jdbc.createdburl	(可選擇) 供建立新資料庫的 URL 僅當使用 Message Queue 資料庫管理員公用程式 (imqdbmgr) 建立的資料庫時，才需要此特性。	jdbc:pointbase:embedded: <i>dbName</i> ; new,database.home= .../instances/ <i>instanceName</i> /dbstore
imq.persist.jdbc.closedburl	(可選擇) 可供關閉資料庫連線的 URL	PointBase 不需要
imq.persist.jdbc.user	(可選擇) 必要時，可供開啓資料庫連線的使用者名稱。 基於安全性原因，可改用指令行選項 imqbrokerd -dbuser 與 imqdbmgr -u 來指定值。	
imq.persist.jdbc.needpassword	(可選擇) 資料庫是否需要密碼以供代理程式存取使用？ 如果為 true，imqbrokerd 與 imqdbmgr 指令將會提示您輸入密碼，除非您使用 -passfile 選項指定包含密碼的密碼檔。	
imq.persist.jdbc.password	(可選擇) 可供開啓資料庫連線的密碼 此特性僅能在密碼檔中指定。	
imq.persist.jdbc.table.IMQSV35	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	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	建立目標表格的 SQL 指令	CREATE TABLE $\{name\}$ (DID VARCHAR(100) NOT NULL, DEST BLOB(10k), <i>primaryKey</i> (DID))

表 14-6 JDBC 型永久性的代理程式特性 (續)

特性	說明	範例
imq.persist.jdbc.table.IMQINT35	建立偏好表格的 SQL 指令	<pre>CREATE TABLE \${name} (CUID BIGINT NOT NULL, INTEREST BLOB(10k), primaryKey(CUID))</pre>
imq.persist.jdbc.table.IMQMSG35	建立訊息表格的 SQL 指令 MSG 的預設最大欄寬為 1 百萬位元組 (1m)。如果您預期訊息長度會大於此長度，請相應設定長度。如果已經建立表格，則您可以重新建立，以變更最大的訊息長度。	<pre>CREATE TABLE \${name} (MID VARCHAR(100) NOT NULL, DID VARCHAR(100), MSGSIZE BIGINT, MSG BLOB(1m), primaryKey(MID))</pre>
imq.persist.jdbc.table.IMQPROPS35	可建立特性表格的 SQL 指令	<pre>CREATE TABLE \${name} (PROPNAME VARCHAR(100) NOT NULL, PROPVALUE BLOB(10k), primaryKey(PROPNAME))</pre>
imq.persist.jdbc.table.IMQILIST35	可建立偏好狀態表格的 SQL 指令	<pre>CREATE TABLE \${name} (MID VARCHAR(100) NOT NULL, CUID BIGINT, DID VARCHAR(100), STATE INTEGER, primaryKey(MID, CUID))</pre>
imq.persist.jdbc.table.IMQTXN35	建立作業事件表格的 SQL 指令	<pre>CREATE TABLE \${name} (TUID BIGINT NOT NULL, STATE INTEGER, TSTATEOBJ BLOB(10K), primaryKey(TUID))</pre>
imq.persist.jdbc.table.IMQTACK35	建立作業事件確認表格的 SQL 指令	<pre>CREATE TABLE \${name} (TUID BIGINT NOT NULL, TXNACK BLOB(10k))</pre>

安全性特性

表 14-7 列出與安全性服務相關的代理程式特性。

表 14-7 代理程式安全性特性

特性	類型	預設	說明
<code>imq.accesscontrol.enabled</code>	布林	true	<p>使用存取控制？</p> <p>如果為 true，系統將會檢查存取控制特性檔，以驗證授權是否使使用者使用連線服務，或執行與特定目標相關的特定作業。</p>
<code>imq.serviceName.accesscontrol.enabled</code>	布林	無	<p>對使用連線服務的存取控制？</p> <p>如果指定，會覆蓋指定連線服務的 <code>imq.accesscontrol.enabled</code>。</p> <p>如果為 true，系統將會檢查存取控制特性檔，以驗證授權是否使使用者使用所指定的連線服務，或執行與特定目標相關的特定作業。</p>
<code>imq.accesscontrol.file.filename</code>	字串	<code>accesscontrol.properties</code>	<p>存取控制特性檔的名稱</p> <p>檔名稱會指定與存取控制目錄相關的路徑 (請參閱附錄 A)。</p>
<code>imq.serviceName.accesscontrol.file.filename</code>	字串	無	<p>連線服務的存取控制特性檔名稱</p> <p>如果指定，會覆蓋指定連線服務的 <code>imq.accesscontrol.file.filename</code>。</p> <p>檔名稱可指定與存取控制目錄相關的路徑 (請參閱附錄 A)。</p>
<code>imq.authentication.type</code>	字串	digest	<p>密碼編碼方法：</p> <ul style="list-style-type: none"> basic Base-64 digest MD5

表 14-7 代理程式安全性特性 (續)

特性	類型	預設	說明
<code>imq.serviceName.authentication.type</code>	字串	無	連線服務的密碼編碼方法： basic Base-64 digest MD5 如果指定，會覆蓋指定連線服務的 <code>imq.authentication.type</code> 。
<code>imq.authentication.basic.user_repository</code>	字串	file	Base-64 認證的使用者儲存庫類型： file 檔案型 ldap LDAP
<code>imq.authentication.client.response.timeout</code>	整數	180	等待用戶端回應認證請求的間隔 (以秒為單位)
<code>imq.passfile.enabled</code>	布林	false	要從密碼檔取得密碼嗎？
<code>imq.passfile.dirpath</code>	字串	請參閱附錄 A	前往包含密碼檔的路徑
<code>imq.passfile.name</code>	字串	密碼檔	密碼檔名稱
<code>imq.imqcmd.password</code>	字串	無	管理使用者的密碼 指令公用程式 (<code>imqcmd</code>) 可使用此密碼，在執行指令之前認證使用者。

表 14-7 代理程式安全性特性 (續)

特性	類型	預設	說明
<code>imq.user_repository.ldap.server</code>	字串	無	<p>LDAP 伺服器的主機名稱與連接埠號碼</p> <p>值的格式為</p> <p><i>hostName:port</i></p> <p>其中 <i>hostName</i> 是執行 LDAP 伺服器的主機所擁有的完全合格 DNS 名稱，而 <i>port</i> 是該伺服器使用的連接埠號碼。</p> <p>若要指定多個轉移伺服器的清單，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，每個空轉移伺服器位址都會以 <code>ldap://</code> 開頭。即使您使用 SSL 並且將特性 <code>imq.user_repository.ldap.ssl.enabled</code> 設定為 <code>true</code>，也請使用此格式。您不需要在位址中指定 <code>ldaps</code>。</p>
<code>imq.user_repository.ldap.principal</code>	字串	無	<p>連結至 LDAP 使用者儲存庫的識別名稱</p> <p>如果 LDAP 伺服器允許匿名搜尋，則不需要。</p>
<code>imq.user_repository.ldap.password</code>	字串	無	<p>連結至 LDAP 使用者儲存庫的密碼</p> <p>如果 LDAP 伺服器允許匿名搜尋，則不需要。</p> <p>此特性應只在密碼檔中指定。</p>
<code>imq.user_repository.ldap.propertyName</code>	敬請期待	敬請期待	敬請期待
<code>imq.user_repository.ldap.base</code>	字串	無	LDAP 使用者項目的目錄基底

表 14-7 代理程式安全性特性 (續)

特性	類型	預設	說明
imq.user_repository.ldap.uidattr	字串	無	LDAP 使用者名稱的提供者特定屬性識別碼
imq.user_repository.ldap.usrfilter	字串	無	(可選擇) LDAP 使用者搜尋的 JNDI 篩選器
imq.user_repository.ldap.grpsearch	布林	false	要啟用 LDAP 群組搜尋? 備註: Message Queue 不支援巢式群組。
imq.user_repository.ldap.grpbase	字串	無	LDAP 群組項目的目錄基底
imq.user_repository.ldap.gidattr	字串	無	LDAP 群組名稱的提供者特定屬性識別碼
imq.user_repository.ldap.memattr	字串	無	LDAP 群組中使用者名稱的提供者特定屬性識別碼
imq.user_repository.ldap.grpfilter	字串	無	(可選擇) LDAP 群組搜尋的 JNDI 篩選器
imq.user_repository.ldap.timeout	整數	280	LDAP 搜尋的時間限制 (以秒為單位)
imq.user_repository.ldap.ssl.enabled	布林	false	要使用 SSL 與 LDAP 伺服器進行通訊?
imq.keystore.file.dirpath	字串	請參閱附錄 A	前往包含金鑰存放區檔案的路徑
imq.keystore.file.name	字串	金鑰存放區	金鑰存放區檔的名稱
imq.keystore.password	字串	無	金鑰存放區檔的密碼 此特性只能在密碼檔中指定。
imq.audit.enabled	布林	false	要啟動代理程式記錄檔的稽核記錄功能? 此選項只適用於 Message Queue Enterprise Edition。

檢視特性

表 14-8 列出與監視服務相關的代理程式特性。

表 14-8 代理程式監視特性

特性	類型	預設	說明
imq.log.level ¹	字串	INFO	<p>記錄層級</p> <p>指定可登入至輸出管道的記錄資訊種類。由高至低的值為：</p> <p>ERROR WARNING INFO</p> <p>每個層級都會包含高於它的層級 (例如 WARNING 會包含 ERROR)。</p>
imq.destination.logDeadMsgs ¹	布林	false	<p>要記錄有關停卹訊息的資訊？</p> <p>如果為 true，將會記錄以下事件：</p> <ul style="list-style-type: none"> 目標已滿，已達最大容量或訊息計數。 代理程式不是因為管理指令或傳送確認而捨棄訊息。 代理程式將訊息移動到停卹訊息佇列。
imq.log.console.stream	字串	ERR	<p>主控台輸出的目標：</p> <p>OUT stdout ERR stderr</p>
imq.log.console.output	字串	ERROR WARNING	<p>要登入至主控台的記錄資訊種類：</p> <p>NONE ERROR WARNING INFO ALL</p> <p>ERROR、WARNING 與 INFO 種類並不會包含它們之上的層級，因此必要時必須明確地加以指定。可將零條 () 分隔，指定任意種類組合。</p>
imq.log.file.dirpath	字串	請參閱附錄 A	前往包含記錄檔的路徑
imq.log.file.filename	字串	log.txt	記錄檔名稱

表 14-8 代理程式監視特性 (續)

特性	類型	預設	說明
img.log.file.output	字串	ALL	要輸入至記錄檔的記錄資訊種類： NONE ERROR WARNING INFO ALL ERROR、WARNING 與 INFO 種類並不會包含它們之上的層級，因此必須小心地指定。可將單條 () 分隔，指定任意種類組合。
img.log.file.rolloverbytes ¹	整數	-1	將輸出自動重建至新記錄檔的檔案長度 (以位元組為單位) -1 值表示不限制位元組數目 (不會根據檔案長度進行自動重建)。
img.log.file.rolloversecs ¹	整數	604800 (-1 秒)	將輸出自動重建至新記錄檔時，此檔案的存在時間 (以秒為單位) -1 值表示不限制秒數 (不會根據檔案存在時間進行自動重建)。
img.log.syslog.output ²	字串	ERROR	要輸入至 syslogd (1M) 的記錄資訊種類： NONE ERROR WARNING INFO ALL ERROR、WARNING 與 INFO 種類並不會包含它們之上的層級，因此必須小心地指定。可將單條 () 分隔，指定任意種類組合。

表 14-8 代理程式監視特性 (續)

特性	類型	預設	說明
<code>imq.log.syslog.facility²</code>	字串	LOG_DAEMON	記錄訊息的 syslog 設備 可能的值對應 syslog(3C) 線上援頁 頁上所列出的值。用於 Message Queue 的適當值包括： LOG_USER LOG_DAEMON LOG_LOCAL0 LOG_LOCAL1 LOG_LOCAL2 LOG_LOCAL3 LOG_LOCAL4 LOG_LOCAL5 LOG_LOCAL6 LOG_LOCAL7
<code>imq.log.syslog.identity²</code>	字串	<code>imqbrokerd_\${imq.instanceName}</code>	要在記錄至 syslog 的所有訊息前 放置的識別字串
<code>imq.log.syslog.logpid²</code>	布林	true	何時記錄代理程式處理 ID 與訊息？
<code>imq.log.syslog.logconsole²</code>	布林	false	如果無法將訊息傳送至 syslog，便 將它們寫入至系統主控台？
<code>imq.log.timezone</code>	字串	本地時區	記錄時間戳記的時區。 這些值會與 <code>java.util.TimeZone.getTimeZone</code> 方法使用的值相同。範例： GMT GMT-8:00 美國 / 洛杉磯 歐洲 / 羅馬 亞洲 / 東京
<code>imq.metrics.enabled</code>	布林	true	啟用將度量資訊寫入至記錄程式的 功能？ 不會影響度量訊息的產生 (由 <code>imq.metrics.topic.enabled</code> 控制)。
<code>imq.metrics.interval</code>	整數	-1	將度量資訊寫入至記錄程式的時間 間隔 (以秒為單位) 不會影響產生度量訊息的時間間隔 (由 <code>imq.metrics.topic.enabled</code> 控制)。 -1 值表示無限期 (永不將度量資訊 寫入至記錄程式)。

表 14-8 代理程式監視特性 (續)

特性	類型	預設	說明
imq.metrics.topic.enabled	布林	true	要啓用將產生的度量訊息存入至度量主題目標的功能嗎？ 如果為 false，則會產生一個用戶端異常，表示嘗試指戶度量主題目標。
imq.metrics.topic.interval	整數	60	要將產生的度量訊息存入至度量主題目標的時間間隔 (以秒為單位)。
imq.metrics.topic.persist	布林	false	要將度量訊息傳送至永久性的度量主題目標嗎？
imq.metrics.topic.timetolive	整數	300	要將度量訊息傳送到度量主題目標的使冊期限 (以秒為單位)。

1. 可與 imqcmd update bkr 指令一起使用
2. 僅限 Solaris 平台

叢集配置特性

表 14-9 列出與代理程式叢集相關的配置特性。

表 14-9 叢集配置的代理程式特性

特性	類型	預設	說明
imq.cluster.brokerlist ¹	字串	無	代理程式位址的清單 此清單由以逗號分隔的一個或多個位址組成。每個位址都可指出叢集中代理程式的主機名稱與連接埠對映器連接埠號碼，使用的格式為 <code>hostName:portNumber</code> 。範例： <code>host1:3000,host2:8000,ctrlhost</code>
imq.cluster.hostname ²	字串	無	叢集連線服務的主機名稱或 IP 位址 如果指定，會覆蓋叢集連線服務的 imq.hostname (請參閱第 266 頁的表 14-1)
imq.cluster.port ²	整數	0	叢集連線服務的連接埠號碼 0 值指定連接埠對映器動態配置連接埠號碼。
imq.cluster.transport ¹	字串	tcp	叢集連線服務的網路傳輸協定 為了在代理程式之間安全傳送加密訊息，請將此特性設定為 <code>ssl</code> 。

表 14-9 叢集配置的代理程式特性 (續)

特性	類型	預設	說明
<code>imq.cluster.url^{1,3}</code>	字串	無	叢集配置檔案的 URL (若有) 範例： <code>http://webserver/imq/cluster.properties</code> (Web 伺服器上的檔案) <code>file:/net/mfsserver/imq/cluster.properties</code> (共用磁碟機上的檔案)
<code>imq.cluster.masterbroker¹</code>	字串	無	叢集主代理程式的主機名稱和連接埠號碼 (若有) 值的格式是 <code>hostName:portNumber</code> ，其中 <code>hostName</code> 是主代理程式的主機名稱， <code>portNumber</code> 是它的連接埠對映器連接埠號碼。範例： <code>ctrlhost:7676</code>

1. 叢集中所有代理程式的值必須相同
2. 可以為叢集中的每個代理程式個別指定
3. 可與 `imqcmd update bkr` 指令一起使用

按字母順序排列的代理程式特性清單

表 14-10 是按字母順序排列的代理程式配置特性清單，可與本章中的相關表格相互參照。

表 14-10 按字母順序排列的代理程式特性清單

特性	表格
<code>imq.accesscontrol.enabled</code>	第 276 頁的表 14-7
<code>imq.accesscontrol.file.filename</code>	第 276 頁的表 14-7
<code>imq.audit.enabled</code>	第 276 頁的表 14-7
<code>imq.authentication.basic.user_repository</code>	第 276 頁的表 14-7
<code>imq.authentication.client.response.timeout</code>	第 276 頁的表 14-7
<code>imq.authentication.type</code>	第 276 頁的表 14-7
<code>imq.autocreate.destination.isLocalOnly</code>	第 269 頁的表 14-3
<code>imq.autocreate.destination.limitBehavior</code>	第 269 頁的表 14-3
<code>imq.autocreate.destination.maxBytesPerMsg</code>	第 269 頁的表 14-3
<code>imq.autocreate.destination.maxNumMsgs</code>	第 269 頁的表 14-3
<code>imq.autocreate.destination.maxNumProducers</code>	第 269 頁的表 14-3

表 14-10 按字母順序排列的代理程式特性清單 (續)

特性	表格
<code>imq.autocreate.destination.maxTotalMsgBytes</code>	第 269 頁的表 14-3
<code>imq.autocreate.destination.useDMQ</code>	第 269 頁的表 14-3
<code>imq.autocreate.queue</code>	第 269 頁的表 14-3
<code>imq.autocreate.queue.consumerFlowLimit</code>	第 269 頁的表 14-3
<code>imq.autocreate.queue.localDeliveryPreferred</code>	第 269 頁的表 14-3
<code>imq.autocreate.queue.maxNumActiveConsumers</code>	第 269 頁的表 14-3
<code>imq.autocreate.queue.maxNumBackupConsumers</code>	第 269 頁的表 14-3
<code>imq.autocreate.topic</code>	第 269 頁的表 14-3
<code>imq.autocreate.topic.consumerFlowLimit</code>	第 269 頁的表 14-3
<code>imq.cluster.brokerlist</code>	第 283 頁的表 14-9
<code>imq.cluster.hostname</code>	第 283 頁的表 14-9
<code>imq.cluster.masterbroker</code>	第 283 頁的表 14-9
<code>imq.cluster.port</code>	第 283 頁的表 14-9
<code>imq.cluster.transport</code>	第 283 頁的表 14-9
<code>imq.cluster.url</code>	第 283 頁的表 14-9
<code>imq.destination.DMQ.truncateBody</code>	第 267 頁的表 14-2
<code>imq.destination.logDeadMsgs</code>	第 280 頁的表 14-8
<code>imq.hostname</code>	第 266 頁的表 14-1
<code>imq.imqcmd.password</code>	第 276 頁的表 14-7
<code>imq.keystore.file.dirpath</code>	第 276 頁的表 14-7
<code>imq.keystore.file.name</code>	第 276 頁的表 14-7
<code>imq.keystore.password</code>	第 276 頁的表 14-7
<code>imq.keystore.propertyName</code>	第 276 頁的表 14-7
<code>imq.log.console.output</code>	第 280 頁的表 14-8
<code>imq.log.console.stream</code>	第 280 頁的表 14-8
<code>imq.log.file.dirpath</code>	第 280 頁的表 14-8
<code>imq.log.file.filename</code>	第 280 頁的表 14-8
<code>imq.log.file.output</code>	第 280 頁的表 14-8
<code>imq.log.file.rolloverbytes</code>	第 280 頁的表 14-8
<code>imq.log.file.rolloversecs</code>	第 280 頁的表 14-8

表 14-10 按字母順序排列的代理程式特性清單 (續)

特性	表格
imq.log.level	第 280 頁的表 14-8
imq.log.syslog.facility	第 280 頁的表 14-8
imq.log.syslog.identity	第 280 頁的表 14-8
imq.log.syslog.logconsole	第 280 頁的表 14-8
imq.log.syslog.logpid	第 280 頁的表 14-8
imq.log.syslog.output	第 280 頁的表 14-8
imq.log.timezone	第 280 頁的表 14-8
imq.message.expiration.interval	第 267 頁的表 14-2
imq.message.max_size	第 267 頁的表 14-2
imq.metrics.enabled	第 280 頁的表 14-8
imq.metrics.interval	第 280 頁的表 14-8
imq.metrics.topic.enabled	第 280 頁的表 14-8
imq.metrics.topic.interval	第 280 頁的表 14-8
imq.metrics.topic.persist	第 280 頁的表 14-8
imq.metrics.topic.timetolive	第 280 頁的表 14-8
imq.passfile.dirpath	第 276 頁的表 14-7
imq.passfile.enabled	第 276 頁的表 14-7
imq.passfile.name	第 276 頁的表 14-7
imq.persist.file.destination.message.filepool.limit	第 272 頁的表 14-5
imq.persist.file.message.cleanup	第 272 頁的表 14-5
imq.persist.file.message.filepool.cleanratio	第 272 頁的表 14-5
imq.persist.file.message.max_record_size	第 272 頁的表 14-5
imq.persist.file.sync.enabled	第 272 頁的表 14-5
imq.persist.jdbc.brokerid	第 273 頁的表 14-6
imq.persist.jdbc.closedburl	第 273 頁的表 14-6
imq.persist.jdbc.createdburl	第 273 頁的表 14-6
imq.persist.jdbc.driver	第 273 頁的表 14-6
imq.persist.jdbc.needpassword	第 273 頁的表 14-6
imq.persist.jdbc.opendburl	第 273 頁的表 14-6
imq.persist.jdbc.password	第 273 頁的表 14-6

表 14-10 按字母順序排列的代理程式特性清單 (續)

特性	表格
<code>imq.persist.jdbc.table.IMQCCREC35</code>	第 273 頁的表 14-6
<code>imq.persist.jdbc.table.IMQDEST35</code>	第 273 頁的表 14-6
<code>imq.persist.jdbc.table.IMQILIST35</code>	第 273 頁的表 14-6
<code>imq.persist.jdbc.table.IMQINT35</code>	第 273 頁的表 14-6
<code>imq.persist.jdbc.table.IMQMSG35</code>	第 273 頁的表 14-6
<code>imq.persist.jdbc.table.IMQPROPS35</code>	第 273 頁的表 14-6
<code>imq.persist.jdbc.table.IMQSV35</code>	第 273 頁的表 14-6
<code>imq.persist.jdbc.table.IMQTACK35</code>	第 273 頁的表 14-6
<code>imq.persist.jdbc.table.IMQTXN35</code>	第 273 頁的表 14-6
<code>imq.persist.jdbc.user</code>	第 273 頁的表 14-6
<code>imq.persist.store</code>	第 271 頁的表 14-4
<code>imq.ping.interval</code>	第 266 頁的表 14-1
<code>imq.portmapper.backlog</code>	第 266 頁的表 14-1
<code>imq.portmapper.hostname</code>	第 266 頁的表 14-1
<code>imq.portmapper.port</code>	第 266 頁的表 14-1
<code>imq.resourceState.count</code>	第 267 頁的表 14-2
<code>imq.resourceState.threshold</code>	第 267 頁的表 14-2
<code>imq.service.activelist</code>	第 266 頁的表 14-1
<code>imq.serviceName.accesscontrol.enabled</code>	第 276 頁的表 14-7
<code>imq.serviceName.accesscontrol.file.filename</code>	第 276 頁的表 14-7
<code>imq.serviceName.authentication.type</code>	第 276 頁的表 14-7
<code>imq.serviceName.max_threads</code>	第 266 頁的表 14-1
<code>imq.serviceName.min_threads</code>	第 266 頁的表 14-1
<code>imq.serviceName.protocolType.hostname</code>	第 266 頁的表 14-1
<code>imq.serviceName.protocolType.port</code>	第 266 頁的表 14-1
<code>imq.serviceName.threadpool_model</code>	第 266 頁的表 14-1
<code>imq.shared.connectionMonitor_limit</code>	第 266 頁的表 14-1
<code>imq.system.max_count</code>	第 267 頁的表 14-2
<code>imq.system.max_size</code>	第 267 頁的表 14-2
<code>imq.transaction.autorollback</code>	第 267 頁的表 14-2

表 14-10 按字母順序排列的代理程式特性清單 (續)

特性	表名
imq.user_repository.ldap.base	第 276 頁的表 14-7
imq.user_repository.ldap.gidattr	第 276 頁的表 14-7
imq.user_repository.ldap.grpbase	第 276 頁的表 14-7
imq.user_repository.ldap.grpfilter	第 276 頁的表 14-7
imq.user_repository.ldap.grpsearch	第 276 頁的表 14-7
imq.user_repository.ldap.memattr	第 276 頁的表 14-7
imq.user_repository.ldap.password	第 276 頁的表 14-7
imq.user_repository.ldap.principal	第 276 頁的表 14-7
imq.user_repository.ldap. <i>propertyName</i>	第 276 頁的表 14-7
imq.user_repository.ldap.server	第 276 頁的表 14-7
imq.user_repository.ldap.ssl.enabled	第 276 頁的表 14-7
imq.user_repository.ldap.timeout	第 276 頁的表 14-7
imq.user_repository.ldap.uidattr	第 276 頁的表 14-7
imq.user_repository.ldap.usrfilter	第 276 頁的表 14-7

實體目標特性參照

本章提供有關實體目標配置特性的參照資訊。您可在建立或更新實體目標時，設定這些特性。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 269 頁的表 14-3）。

表 15-1 實體目標特性

特性	類型	預設	說明
maxNumMsgs ¹	整數	-1	未使用訊息的最大數目 -1 值表示不限制訊息數目。 對於停用訊息佇列，預設值是 1000。
maxBytesPerMsg	字串	-1	任何單一訊息的最大大小（以位元組為單位） 拒絕永久性訊息就會對生產型用戶端報告發生異常；若為非永久性訊息，則不會傳送通知。 此值可能會使下列字尾，以位元組、千位元組或百萬位元組表示： b 位元組 k 千位元組 (1024 位元組) m 百萬位元組 (1024 x 1024 = 1,048,576 位元組) 無字尾的值是以位元組表示；-1 值表示不限制訊息容量。 範例： 1600 1600 位元組 1600b 1600 位元組 16k 16 千位元組 (= 16,384 位元組) 16m 16 百萬位元組 (= 16,777,216 位元組) -1 不限制
maxTotalMsgBytes ¹	字串	-1	未使用訊息的最大總記憶體容量（以位元組為單位） 語法與 maxBytesPerMsg 相同（請參閱前述內容）。 對於停用訊息佇列，預設值是 10m。

表 15-1 實體目標特性 (續)

特性	類型	預設	說明
limitBehavior	字串	REJECT_NEWEST	<p>達到記憶體限制臨界值時的代理程式運作方式：</p> <ul style="list-style-type: none"> FLOW_CONTROL 減緩產生者速度 REMOVE_OLDEST 捨棄最舊的訊息 REMOVE_LOW_PRIORITY 根據訊息存在時間，捨棄優先權最低的訊息；不會對生產型客戶端傳送通知 REJECT_NEWEST 拒絕最新的訊息；只有在當訊息為永久性時，才通知生產型客戶端發生異常 <p>如果值為 REMOVE_OLDEST 或 REMOVE_LOW_PRIORITY，而且 useDMQ 特性為 true，超過限制的訊息就會移至停駐的訊息佇列。對於停駐訊息佇列本身，預設的限制運作方式是 REMOVE_OLDEST，而且無法設為 FLOW_CONTROL。</p>
maxNumProducers ²	整數	-1	<p>目標的最大訊息產生者數目</p> <p>達到限制時，將無法建立新的產生者。-1 值表示不限制產生者數目。</p>
maxNumActiveConsumers ³	整數	1	<p>在佇列目標的負載平衡傳送中，使客戶訊息客戶的最大數目</p> <p>-1 值表示不限制客戶數目。在 Sun Java System Message Queue Platform Edition 中，此值會限制為 2。</p>
maxNumBackupConsumers ³	整數	0	<p>在佇列目標的負載平衡傳送中，備份訊息客戶的最大數目</p> <p>-1 值表示不限制客戶數目。在 Sun Java System Message Queue Platform Edition 中，此值會限制為 1。</p>
consumerFlowLimit	整數	1000	<p>單一批次中傳送給客戶的最大訊息數目</p> <p>在負載平衡佇列傳送中，開始負載平衡前，此數目是藉由到使客戶客戶的佇列訊息初始數目。目標客戶可以在連線上指定較低的值來覆蓋這項限制。</p> <p>-1 值表示不限制客戶數目。</p>
isLocalOnly ²	布林	false	<p>僅限本機傳送？</p> <p>此特性僅適用於代理程式叢集，而且目標一經建立之後便無法變更。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機客戶（連線至本機目標的代理程式的客戶）。</p>
localDeliveryPreferred ^{2a} ³	布林	false	<p>偏好本機傳送？</p> <p>此特性僅適用於代理程式叢集中的負載平衡佇列傳送。如果為 true，只有在當本機代理程式上沒有客戶時，才會將訊息傳送給遠端客戶；目標絕不可僅限於本機傳送 (isLocalOnly 必須為 false)。</p>

表 15-1 實體目標特性 (續)

特性	類型	預設	說明
useDMQ ²	布林	true	是否將停冊訊息傳送給停冊訊息佇列？ 如果為 false，停冊訊息將會被捨棄。

1. 在叢集環境中，會套用至目標的各個實例，而非整體套用至叢集中的所有實例
2. 不套用於停用的訊息佇列
3. 僅限佇列目標

管理物件屬性參照

本章提供管理物件屬性的參照資訊。包含下列各節：

- 第 294 頁的「連線工廠屬性」
- 第 300 頁的「目標屬性」
- 第 300 頁的「SOAP 端點屬性」

連線工廠屬性

連線工廠物件的屬性分爲幾個種類，在下列各節中有所介紹：

- 第 294 頁的「連線處理」
- 第 297 頁的「用戶端標識」
- 第 297 頁的「穩定性與流量控制」
- 第 299 頁的「佇列瀏覽器和伺服器階段作業」
- 第 299 頁的「設定標準訊息特性」
- 第 299 頁的「訊息標頭覆寫」

連線處理

表 16-1 列出了連線處理的連線工廠屬性。

表 16-1 連線處理的連線工廠屬性

屬性	類別	預設	說明
imqAddressList	字串	若存在，是現有的 Message Queue 3.0 位址；若不存在，則爲第 295 頁的表 16-2 的第一個項目。	代理程式位址的清單 此清單由以逗號分隔的一個或多個訊息伺服器位址組成。每個位址都會指定 (或表示) 主機名稱、連接埠號碼，以及用戶端可以連線的代理程式實例的連線服務。位址語法會因連線服務與連接埠指定方法而異；請參閱下列各節以取得詳細資訊。
imqAddressListBehavior	字串	PRIORITY	嘗試連線至伺服器位址的順序： PRIORITY 位址清單中指定的順序 RANDOM 隨機順序 備註： 如果訊息用戶端共用相同的連線工廠，請指定隨機連線順序，以免全都嘗試連線至相同的位址。
imqAddressListIterations	整數	5	在位址清單上嘗試建立或重新建立連線的重複次數 -1 值表示不限制重複次數。
imqPingInterval	整數	30	測試用戶端與代理程式間連線的間隔 (以秒爲單位) 0 或 -1 值會停用定期的連線測試。
imqReconnectEnabled	布林	false	嘗試重新建立斷開的連線？

表 16-1 連線處理的連線工廠屬性 (續)

屬性	類型	預設	說明
imqReconnectAttempts	整數	0	移至位址清單中的下一個位址之前，嘗試連線 (或重新連線) 至每個位址的次數。 -1 值表示不限制嘗試連線的次數；可重複嘗試連線至下一個位址，直到成功為止。
imqReconnectInterval	長整數	3000	嘗試重新連線間的間隔 (以毫秒為單位)。 此值會套用於指定位址上的連續嘗試，以及清單中連續的位址。 備註： 過小的值可能會讓代理程式來不及回復；而過大的值則可能會導致令人無法忍受的延遲。
imqSSLIsHostTrusted	布林	true	要接受代理程式的自身簽名證書憑證？ 備註： 若要使用非證書機構簽名的憑證，請將此值設定為 false。

imqAddressList 屬性的值是以逗號分隔的字串，指定要連線的一個或多個訊息伺服器位址。每個位址的一般語法如下：

scheme://*address*

其中 *scheme* 可識別表 16-2 第一欄中顯示的其中一個定址機制，而 *address* 表示伺服器位址本身。指定位址的確切語法會因定址機制而異，如表格的最後一欄所示。

表 16-2 訊息伺服器定址機制

方案	變數	語法	說明
mq	jms 或 ssljms	[<i>hostName</i>][: <i>portNumber</i>]/[<i>serviceName</i>]	動態指定 jms 或 ssljms 連線服務的連接埠。 位址清單項目可指定 Message Queue 連接埠對映器的主機名稱與連接埠號碼。連接埠對映器本身會動態指定連線使用的連接埠。 預設值： <i>hostName</i> = localhost <i>portNumber</i> = 7676 <i>serviceName</i> = jms 若為 ssljms 連線服務，則必須明確指定所有變數。

表 16-2 訊息伺服器定址機制 (續)

方案	明符	語法	說明
mqtcp	jms	<i>hostName:portNumber/jms</i>	<p>使用 jms 連線服務連線至指定的連接埠。</p> <p>透過連接埠對映器，並直接對指定的主機名稱與連接埠號碼建立 TCP 連線。</p>
mqssl	ssljms	<i>hostName:portNumber/ssljms</i>	<p>使用 ssljms 連線服務，連線至指定的連接埠。</p> <p>透過連接埠對映器，並直接對指定的主機名稱與連接埠號碼建立完全的 SSL 連線。</p>
http	httpjms	<p><i>http://hostName:portNumber/contextRoot/tunnel</i></p> <p>如果多個代理程式實例使用相同的通道 Servlet，下列語法可連線至特定 (而非隨機選定) 的代理程式實例：</p> <p><i>http://hostName:portNumber/contextRoot/tunnel?ServerName=hostName:instanceName</i></p>	<p>使用 httpjms 連線服務，連線至指定的連接埠。</p> <p>在指定的 URL 建立與 Message Queue 通道 Servlet 的 HTTP 連線。您必須配置代理程式才能存取 HTTP 通道 Servlet。</p>
https	httpsjms	<p><i>https://hostName:portNumber/contextRoot/tunnel</i></p> <p>如果多個代理程式實例使用相同的通道 Servlet，下列語法可連線至特定 (而非隨機選定) 代理程式實例：</p> <p><i>https://hostName:portNumber/contextRoot/tunnel?ServerName=hostName:instanceName</i></p>	<p>使用 httpsjms 連線服務，連線至指定的連接埠。</p> <p>在指定的 URL 建立與 Message Queue 通道 Servlet 的完全 HTTPS 連線。您必須配置代理程式才能存取 HTTPS 通道 Servlet。</p>

表 16-3 顯示不同位址格式範例。

表 16-3 訊息伺服器位址範例

明符	代理程式主機	連接埠	範例位址
未指定	未指定	未指定	無位址 (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	指定連接埠對映器連接埠	1012 (mq://localhost:1012/jms)
ssljms	主機名稱	標準連接埠對映器連接埠	mq://localhost:7676/ssljms

表 16-3 訊息伺服器位址範例 (續)

明碼	代理程式主機	連接埠	範例位址
ssljms	指定主機	標準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	本機主機	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
httpjms	不註冊	不註冊	http://websrvr1:8085/imq/tunnel
httpsjms	不註冊	不註冊	https://websrvr2:8090/imq/tunnel

用戶端標識

表 16-4 列出用戶端標識的連線工廠屬性。

表 16-4 用戶端標識的連線工廠屬性

屬性	類型	預設	說明
imqDefaultUsername	字串	guest	以代理程式進行認證時使用的預設使用者名稱
imqDefaultPassword	字串	guest	以代理程式進行認證時使用的預設密碼
imqConfiguredClientID	字串	null	管理區內的用戶端識別碼
imqDisableSetClientID	布林	false	要使用 setClientID 方法，防止用戶端變更用戶端識別碼？

穩定性與流量控制

表 16-5 列出了供穩定性與流量控制使用的連線工廠屬性。

表 16-5 穩定性與流量控制的連線工廠屬性

屬性	類型	預設	說明
imqAckTimeout	字串	0	<p>等待代理程式回應的最長時間 (以毫秒為單位)，一旦逾時立即會產生異常。</p> <p>0 值表示無逾時 (永遠等待)。</p> <p>備註：在時延小的值會導致發生提早逾時的情況：例如，使用安全 (SSL) 連線對 LDAP 使用者儲存庫進行初始使用者認證時，可能需要 30 秒鐘以上。</p>

表 16-5 穩定性與流量控制的連線工廠屬性 (續)

屬性	類型	預設	說明
imqConnectionFactoryCount	整數	100	<p>度量批次中的非效負載訊息數</p> <p>超過此訊息數，就會暫時暫停傳送非效負載訊息至客戶端，以傳送所有累積的控制訊息。依照客戶端執行階段的通知，恢復傳送非效負載訊息，直至計數再次到達指定數目為止。</p> <p>0 值可停用訊息傳送度量，但可能會導致大量非效負載訊息流量阻斷的 Message Queue 控制訊息。</p>
imqConnectionFactoryLimitEnabled	布林	false	要限制連線層級的訊息流量？
imqConnectionFactoryLimit	整數	1000	<p>每個連線可傳送和緩衝供使用的最大訊息數</p> <p>擱置的未使用非效負載訊息數 (受到 imqConnectionFactoryCount 管理之流量度量的管制) 超過此限制時，連線就會停止傳送訊息。只有當擱置的訊息低於限制時，才會繼續傳送。如此可避免客戶端因擱置訊息導致耗盡記憶體，而造成過度負載。</p> <p>如果 imqConnectionFactoryLimitEnabled 為 false，則可忽略此屬性。</p>
imqConsumerFlowLimit	整數	100	<p>每個客戶端可傳送和緩衝供使用的最大訊息數</p> <p>該客戶端擱置未使用的非效負載訊息數超過此限制時，就會停止將訊息傳送給指定的客戶端。只有當該客戶端的擱置訊息低於 imqConsumerFlowThreshold 指定的百分比時，才會繼續傳送。如此可改善多個使用者間的負載平衡，並防止單一客戶端霸佔連線，以免讓使用者間連線的其他客戶端等。</p> <p>可以較小的值觀察此限制，以供佇列自身的 consumerFlowLimit 屬性使用 (請參閱第 15 章「實體目標特性參照」)。請注意，在連線上將訊息傳送給所有客戶端時，必須遵守 imqConnectionFactoryLimit 所指定的整體限制。</p>
imqConsumerFlowThreshold	整數	50	<p>在客戶端執行階段緩衝的每位客戶端之訊息數，以 imqConsumerFlowLimit 的百分比表示，當低於該值時，便會繼續傳送訊息</p>

佇列瀏覽器和伺服器階段作業

表 16-6 列出佇列瀏覽和伺服器階段作業的連線工廠屬性。

表 16-6 佇列瀏覽器和伺服器階段作業的連線工廠屬性

屬性	類別	預設	說明
imqQueueBrowserMaxMessagesPerRetrieve	整數	1000	瀏覽佇列目標時，一次可擷取的最大訊息數。
imqQueueBrowserRetrieveTimeout	長整數	60000	瀏覽佇列目標時，等待擷取訊息的最長時間 (以毫秒為單位)，一旦逾時立即產生異常。
imqLoadMaxToServerSession	布林	true	要將最大訊息數載入伺服器階段作業？ 如果指定 false，則一端一次只載入一個訊息。 此屬性僅適用於 JMS 應付程式伺服器設備。

設定標準訊息特性

表 16-7 列出的連線工廠屬性，可控制 Message Queue 用戶端執行階段是否設定「Java Message Service 規格」中所定義的某些標準訊息特性。

表 16-7 標準訊息特性的連線工廠屬性

特性	類別	預設	說明
imqSetJMSXUserID	布林	false	要為產生的訊息設定 JMSXUserID 特性 (傳送訊息的用戶端身份)？
imqSetJMSXAppID	布林	false	要為產生的訊息設定 JMSXAppID 特性 (傳送訊息的應付程式身份)？
imqSetJMSXProducerTXID	布林	false	要為產生的訊息設定 JMSXProducerTXID 特性 (產生訊息的作業事件之作業事件識別碼)？
imqSetJMSXConsumerTXID	布林	false	要為產生的訊息設定 JMSXConsumerTXID 特性 (產生訊息的作業事件之作業事件識別碼)？
imqSetJMSXRcvTimestamp	布林	false	要為接收的訊息設定 JMSXRcvTimestamp 特性 (傳送給用戶端的時間戳)？

訊息標頭覆寫

表 16-8 列出用以覆寫 JMS 訊息標頭欄位的連線工廠屬性。

表 16-8 訊息標頭覆寫的連線工廠屬性

屬性	類型	預設	說明
imqOverrideJMSDeliveryMode	布林	false	允許覆寫客戶端設定的傳送模式？
imqJMSDeliveryMode	整數	2	覆寫傳送模式的值： 1 非永久性的 2 永久性的
imqOverrideJMSExpiration	布林	false	允許覆寫客戶端設定的逾期時間？
imqJMSExpiration	長整數	0	覆寫逾期時間的值（以毫秒為單位） 0 值表示不限制逾期時間（訊息永不逾期）。
imqOverrideJMSPriority	布林	false	允許覆寫客戶端設定的優先層級？
imqJMSPriority	整數	4（一般）	覆寫優先層級的值 （0 至 9）
imqOverrideJMSHeadersToTemporaryDestinations	布林	false	要將覆寫客戶端至暫時目標？

目標屬性

表 16-9 列出可為目標管理物件設定的屬性。

表 16-9 目標屬性

屬性	類型	預設	說明
imqDestinationName	字串	Untitled_Destination_Object	實體目標名稱 目標名稱可能僅包含字母數字字元（無空格），而且必須以字母字元或底線（_）或錢幣符號（\$）開頭。它不會以 mq 字元開頭。
imqDestinationDescription	字串	無	目標的描述性字串

SOAP 端點屬性

表 16-10 列出用以配置端點 URL，以供使用簡單物件存取協定 (SOAP) 的應用程式使用；請參閱「Message Queue Developer's Guide for Java Clients」，以獲得更多資訊。

表 16-10 SOAP 端點屬性

屬性	類型	預設	說明
imgSOAPEndpointList	字串	無	<p>一個或多個表示接收訊息之 SOAP 端點的 URL 清單，用空格相互分隔。</p> <p>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> <pre>http://www.serv1/ http://www.serv2/</pre> <p>如果該清單指定多個 URL，則會向所有的 URL 廣播訊息。</p>
imgEndpointName	字串	Untitled_Endpoint_Object	SOAP 端點名稱
imgEndpointDescription	字串	無	<p>SOAP 端點的描述性字串</p> <p>範例：</p> <pre>My endpoints for broadcast</pre>

JMS 資源介面的特性參照

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，即可使用 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，展現其配置特性：

- ResourceAdapter 配置全面影響資源介面的運作方式。
- ManagedConnectionFactory 配置會影響資源介面所建立供訊息驅動 Bean (MDB) 使用的連線。
- ActivationSpec 配置會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。

若要設定這些實體的特性值，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 資源介面的配置特性。它包含下面各節：

- [第 304 頁的「ResourceAdapter JavaBean」](#)
- [第 305 頁的「ManagedConnectionFactory JavaBean」](#)
- [第 306 頁的「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以配置此 JavaBean 的特性。註腳會標示出每個必要的特性。

表 17-1 資源介面特性

特性	預設	註腳
addressList ¹	mq://localhost:7676 /jms	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。 資源介面提供預設值。 addressList 是 Sun Java System Message Queue 特性的特性名稱，但是意義與標準特性 connectionURL 相同。Sun Java System Message Queue 提供這兩種特性名稱。您必須設定 connectionURL 或 addressList，兩者是相等的。
addressListBehavior	PRIORITY	此字串指定資源介面如何連線至 Message Queue 服務。此值是 PRIORITY 或 RANDOM。 PRIORITY 連線選取位址清單 (addressList) 中指定的第一個 Message Queue 代理程式。 RANDOM 連線隨機選取位址清單中的 Message Queue 代理程式。 連線失敗後的重新連線方式對於 PRIORITY 和 RANDOM 是相同的。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。
addressListIterations	1	在位址清單中重複連線的次數。此值會防止站連線和繼續嘗試重新連線。
connectionURL	mq://localhost:7676 /jms	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。 等同於 addressList 特性；如需詳細資訊，請參閱前述註腳。
userName ¹	guest	資源介面連線至 Message Queue 服務時，所使用的預設使用者名稱。 資源介面提供預設值。
password ¹	guest	資源介面連線至 Message Queue 服務時，所使用的預設密碼。 資源介面提供預設值。
reconnectAttempts	6	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，即可使用此特性。
reconnectEnabled	false	此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。

表 17-1 資源介面特性 (續)

特性	類型	說明
reconnectInterval	30000	嘗試重新連線的間隔 (以毫秒為單位)。reconnectEnabled 設定為 true 時，即可使用此特性。

1. 這是必要的特性。

ManagedConnectionFactory JavaBean

管理式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 也有類似的特性，此設定會抑制 ResourceAdapter Bean 的類似指定值。

表 17-2 會列出並描述 Message Queue 資源介面提供的管理式連線工廠之可配置屬性。

表 17-2 管理式連線工廠屬性

屬性	類型	說明
addressList	無	衍生自此管理式連線工廠的連線清單。 此屬性的格式如第 304 頁的表 17-1 中所描述，符合訊息服務 addressList。如果未設定此值，則連線工廠表中所描述，使用 ResourceAdapter JavaBean 的 addressList 指定值。
addressListBehavior	PRIORITY	此字串指定資源介面如何連線至 Message Queue 服務。此值是 PRIORITY 或 RANDOM。 PRIORITY 連線隨機取位址清單 (addressList) 中指定的第一個 Message Queue 代理程式。 RANDOM 連線隨機取位址清單中的 Message Queue 代理程式。 連線失敗後的重重新連線方式對於 PRIORITY 和 RANDOM 是相同的。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位址清單執行連線。
addressListIterations	1	在位址清單中重複連線的次數。此值會用於始連線和繼續嘗試重新連線。
clientID	無	衍生自此管理式連線工廠的連線所使用的使用端識別碼。
password	guest	(可選擇) 連線的密碼。 如果未設定此值，則連線工廠如第 304 頁的表 17-1 中所描述，使用 ResourceAdapter JavaBean 的指定密碼。
reconnectAttempts	6	在位址清單中嘗試重新連線單一項目的次數。

表 17-2 管理式連線工廠屬性 (續)

屬性	預設	說明
reconnectEnabled	false	此布林值指定連線失敗後是嘗試重新連線或嘗試新的連線。嘗試重新連線，由 reconnectInterval 和 reconnectAttempts 屬性所控制。
reconnectInterval	30000	嘗試重新連線到 Message Queue 服務的最小間隔毫秒數。
userName	guest	(可選擇) 連線的使用者名稱。 如果未設定此值，則連線如同 304 頁的表 17-1 中所描述，使 ResourceAdapter JavaBean 的指定使用者名稱。

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 會列出並描述訊息端點啟動規格的可配置特性。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 啟動規格特性

特性	預設	說明
acknowledgeMode	Auto-acknowledge	(可選擇) 使用者建立的 JMS 階段作業回應模式。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。
addressList	繼承 ResourceAdapter JavaBean 配置中的 addressList	(可選擇) 資源介面為訊息端點製作的連線規格。這是 Message Queue JMS 資源介面的特有特性。 有效值必須符合訊息服務連線位址語法。
clientId	無	為此使用者建立的 JMS 連線所使用的使用者 JMS 使用者端 ID。 如果您將 subscriptionDurability 特性設定為 Durable，就必須設定此特性。 這是標準 EJB 2.1 和 J2EE CA 1.5 特性。

表 17-3 啓動規格特性 (續)

特性	類型	說明
customAcknowledgeMode	無	此字串指定 MDB 訊息使用模式。 此特性的有效值是 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，會無法啟動訂閱。
destination	無	MDB 從中接收訊息的目標名稱。 這是必需的特性。這是標準 EJB 2.1 和 J2EE CA1.5 特性。 此值必須設定為 Message Queue 目標管理物件的 destinationName 特性值。
destinationType	無	destination 特性指定的目標類型。有效值是 javax.jms.Queue 或 javax.jms.Topic。 這是必需的特性。這是標準 EJB 2.1 和 J2EE CA1.5 特性。
endpointExceptionRedeliveryAttempts	6	MDB 在傳送訊息期間拋出異常時，由 MDB 重新傳送訊息的次數。
messageSelector	無	(可選擇) 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。
sendUndeliverableMsgsToDMQ	true	此布林值指定當 MDB 拋出執行階段異常並且嘗試重新傳送的次數超出 endpointExceptionRedeliveryAttempts 值時，是否將訊息放入停用的訊息佇列中。 如果是 false，Message Queue 代理程式會嘗試向任何有效的用戶 (包括此 MDB) 重新傳送訊息。
subscriptionDurability	NonDurable	此字串指定主題目標的用戶是長期或非長期。此值可能是 NonDurable 或 Durable。 此特性是非長期訂閱可選擇的特性，但對長期訂閱為必需特性。如果您將此值設定為 Durable，還必須設定特性 clientID 和 subscriptionName。 這是標準 EJB 2.1 和 J2EE CA1.5 特性，而且只在 destinationType 特性設定為 avax.jms.Topic 時有效。
subscriptionName	無	此字串可以為長期訂閱命名。 如果您將 subscriptionDurability 特性設定為 Durable，就必須設定此特性。 這是標準 EJB 2.1 和 J2EE CA 1.5 特性。

ActivationSpec JavaBean

度量參照

本章列出並說明 Message Queue 產品產生的度量。本章包含以下各節：

- 第 310 頁的「JVM 度量」
- 第 310 頁的「代理程式通用度量」
- 第 312 頁的「連線服務度量」
- 第 314 頁的「目標度量」

JVM 度量

表 18-1 列出並說明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 (metrics topic) 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 191 頁的表 10-7。

代理程式通用度量

表 18-2 列出並說明代理程式報告關於代理程式整體度量資訊的資料。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 代理程式通用度量

度量數目	說明	imqcmd metrics bkr (metricType)	記錄檔	度量訊息 (metrics topic) 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
儲存的消息資料				

表 18-2 代理程式通用度量 (續)

度量數目	說明	imqcmd metrics bkr (metricType)	記錄性	度量訊息 (metrics topic) ¹
訊息數目	目前儲存在代理程式記憶體和永久存放區中的非效負載訊息數目	不需要 使用 query bkr	不需要	需要 (...broker)
訊息總容量	目前儲存在代理程式記憶體和永久存放區中的非效負載訊息容量	不需要 使用 query bkr	不需要	需要 (...broker)
訊息流量資料				
傳入的訊息數目	從上次代理程式啟動後，傳入代理程式的非效負載訊息數目	需要 (ttl)	需要	需要 (...broker)
傳入的訊息容量	從上次代理程式啟動後，傳入代理程式的非效負載訊息容量	需要 (ttl)	需要	需要 (...broker)
傳入的封包數目	從上次代理程式啟動後，傳入代理程式的封包數目；包括非效負載訊息和控制訊息	需要 (ttl)	需要	需要 (...broker)
傳入的封包容量	從上次代理程式啟動後，傳入代理程式的封包容量；包括非效負載訊息和控制訊息	需要 (ttl)	需要	需要 (...broker)
傳出的訊息數目	從上次代理程式啟動後，傳出代理程式的非效負載訊息數目	需要 (ttl)	需要	需要 (...broker)
傳出的訊息容量	從上次代理程式啟動後，傳出代理程式的非效負載訊息容量	需要 (ttl)	需要	需要 (...broker)
傳出的封包數目	從上次代理程式啟動後，傳出代理程式的封包數目；包括非效負載訊息和控制訊息。	需要 (ttl)	需要	需要 (...broker)
傳出的封包容量	從上次代理程式啟動後，傳出代理程式的封包容量；包括非效負載訊息和控制訊息。	需要 (ttl)	需要	需要 (...broker)
訊息傳入的速率	目前傳入代理程式的非效負載訊息速率	需要 (rts)	需要	不需要
訊息容量傳入的速率	目前傳入代理程式之非效負載訊息容量的速率	需要 (rts)	需要	不需要
封包傳入的速率	目前傳入代理程式的封包速率；包括非效負載訊息和控制訊息	需要 (rts)	需要	不需要
封包容量傳入的速率	目前傳入代理程式之封包容量的速率；包括非效負載訊息和控制訊息	需要 (rts)	需要	不需要
訊息傳出的速率	目前傳出代理程式的非效負載訊息速率	需要 (rts)	需要	不需要
訊息容量傳出的速率	目前傳出代理程式之非效負載訊息容量的速率	需要 (rts)	需要	不需要

表 18-2 代理程式通用度量 (續)

度量數目	說明	imqcmd metrics bkr (metricType)	記錄檔	度量訊息 (metrics topic) ¹
封包傳出的速率	目前傳出代理程式的封包速率；包括非效負載訊息和控制訊息	需要 (rts)	需要	不需要
封包容量傳出的速率	目前傳出代理程式之封包容量的速率；包括非效負載訊息和控制訊息	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 191 頁的表 10-7。

連線服務度量

表 18-3 列出並說明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 (metrics topic)
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
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總數	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總數	需要 (cxn)	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 (metrics topic)
訊息流量資料				
傳入的訊息數目	從上次代理程式啟動後，傳入連線服務的 有效負載訊息數目。	需要 (ttl)	不需要	不需要
傳入的訊息容量	從上次代理程式啟動後，傳入連線服務的 有效負載訊息容量。	需要 (ttl)	不需要	不需要
傳入的封包數目	從上次代理程式啟動後，傳入連線服務的 封包數目；包括有效負載訊息和控制訊 息。	需要 (ttl)	不需要	不需要
傳入的封包容量	從上次代理程式啟動後，傳入連線服務的 封包容量；包括有效負載訊息和控制訊 息。	需要 (ttl)	不需要	不需要
傳出的訊息數目	從上次代理程式啟動後，傳出連線服務的 有效負載訊息數目	需要 (ttl)	不需要	不需要
傳出的訊息容量	從上次代理程式啟動後，傳出連線服務的 有效負載訊息容量	需要 (ttl)	不需要	不需要
傳出的封包數目	從上次代理程式啟動後，傳出連線服務的 封包數目；包括有效負載訊息和控制訊 息。	需要 (ttl)	不需要	不需要
傳出的封包容量	從上次代理程式啟動後，傳出連線服務的 封包容量；包括有效負載訊息和控制訊 息。	需要 (ttl)	不需要	不需要
訊息傳入的速率	目前透過連線服務傳入代理程式的有效負 載訊息速率	需要 (rts)	不需要	不需要
訊息容量傳入的速率	目前傳入連線服務之有效負載訊息容量的 速率	需要 (rts)	不需要	不需要
封包傳入的速率	目前傳入連線服務的封包速率；包括有效 負載訊息和控制訊息	需要 (rts)	不需要	不需要
封包容量傳入的速率	目前傳入連線服務之封包容量的速率；包 括有效負載訊息和控制訊息	需要 (rts)	不需要	不需要
訊息傳出的速率	目前傳出連線服務的有效負載訊息速率	需要 (rts)	不需要	不需要
訊息容量傳出的速率	目前傳出連線服務之有效負載訊息容量的 速率	需要 (rts)	不需要	不需要
封包傳出的速率	目前傳出連線服務的封包速率；包括有效 負載訊息和控制訊息	需要 (rts)	不需要	不需要

表 18-3 連線服務度量 (續)

度量數目	描述	imqcmd metrics svc (metricType)	記錄檔	度量訊息 (metrics topic)
封包容量傳出的速率	目前傳出連線服務的封包容量速率；包括有效負載訊息和控制訊息	需要 (rts)	不需要	不需要

目標度量

表 18-4 列出並說明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	描述	imqcmd metrics dst (metricType)	記錄檔	度量訊息 (metrics topic) ¹
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間隔、使用中長期間隔和非使用中長期間隔。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 (僅適用於佇列)	需要 (con)	不需要	需要 (...destName)
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 (僅適用於佇列)	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 (僅適用於佇列)	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)	記錄性	度量訊息 (metrics topic) ¹
儲存的訊息資料				
訊息數目	目前儲存於目標記憶體和永久存放區的 非負載訊息數目	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 憶體和永久存放區中的非負載訊息平 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 憶體和永久存放區中的非負載訊息尖 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存放區的 非負載訊息容量	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 憶體和永久存放區中的非負載訊息平 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 憶體和永久存放區中的非負載訊息尖 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 一訊息中非負載訊息的尖峰容量	需要 (ttl) (rts)	不需要	需要 (...destName)
訊息流量資料				
傳入的訊息數目	從上次代理程式啟動後，傳入此目標的 非負載訊息數目	需要 (ttl)	不需要	需要 (...destName)
傳入的訊息容量	從上次代理程式啟動後，傳入此目標的 非負載訊息容量	需要 (ttl)	不需要	需要 (...destName)
傳出的訊息數目	從上次代理程式啟動後，傳出此目標的 非負載訊息數目	需要 (ttl)	不需要	需要 (...destName)
傳出的訊息容量	從上次代理程式啟動後，傳出此目標的 非負載訊息容量	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)	記錄性	度量訊息 (metrics topic) ¹
訊息傳入的速率	目前傳入目標的每秒負載訊息速率	需要 (rts)	不需要	不需要
訊息傳出的速率	目前傳出目標的每秒負載訊息速率	需要 (rts)	不需要	不需要
訊息容量傳入的速率	目前傳入此目標之每秒負載訊息容量的速率	需要 (rts)	不需要	不需要
訊息容量傳出的速率	目前傳出此目標之每秒負載訊息容量的速率	需要 (rts)	不需要	不需要
磁碟使用資料				
保留的磁碟	檔案型目標儲存中，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	檔案型目標儲存中，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會用於保留使用中訊息	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 191 頁的表 10-7。

附錄

- 附錄 A 「Message Queue 資料的平台特定位置」
- 附錄 B 「Message Queue 介面的穩定性」
- 附錄 C 「HTTP/HTTPS 支援」

Message Queue 資料的平台特定位置

Sun Java System Message Queue 資料儲存在不同作業系統平台的不同位置。下表顯示各式不同類型的 Message Queue 資料，在下列平台的位置：

- 第 320 頁的「[Solaris](#)」
- 第 321 頁的「[Linux](#)」
- 第 322 頁的「[Windows](#)」

下列表格中，*instanceName* 表示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 322 頁的「Windows」中所述的結構相似。

表 A-1 Solaris 平台上的 Message Queue 資料位置

資料種類	位置
代理程式實例配置檔案	<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>
永久存放區(訊息、目標、長期訂閱、作業事件)	<code>/var/imq/instances/instanceName/fs350</code> 或 JDBC 可存取資料存放區
代理程式實例記錄檔目錄 (預設位置)	<code>/var/imq/instances/instanceName/log/</code>
管理物件 (物件存放區)	您選擇的本機目錄或 LDAP 伺服器
安全性：使用者信託庫	<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：代理程式金鑰存放區檔案位置	<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 平台上的 Message Queue 資料位置

資料種類	位置
代理程式實例配置檔案	<code>/var/opt/sun/mq/instances/instanceName/props/config.properties</code>
代理程式配置檔案範本	<code>/opt/sun/mq/private/share/lib/props/</code>
永久存放區(訊息、目標、長期訂購、作業事件)	<code>/var/opt/sun/mq/instances/instanceName/fs350/</code> 或 JDBC 可存取之資料存放區
代理程式實例記錄檔目錄(預設位置)	<code>/var/opt/sun/mq/instances/instanceName/log/</code>
管理物件(物件存放區)	您選擇之本機目錄或 LDAP 伺服器
安全性：使用者信託庫	<code>/var/opt/sun/mq/instances/instanceName/etc/passwd</code> 或 LDAP 伺服器
安全性：存取控制檔(預設位置)	<code>/var/opt/sun/mq/instances/instanceName/etc/accesscontrol.properties</code>
安全性：密碼檔案目錄(預設位置)	<code>/var/opt/sun/mq/instances/instanceName/etc/</code>
安全性：範例密碼檔案	<code>/etc/opt/sun/mq/passfile.sample</code>
安全性：代理程式金鑰存放區檔案位置	<code>/etc/opt/sun/mq/</code>
JavaDoc API 物件	<code>/opt/sun/mq/javadoc/index.html</code>
範例應用程式與配置	<code>/opt/sun/mq/examples/</code>
Java 歸檔(.jar)、Web 歸檔(.war)與 資源介庫歸檔(.rar)檔案	<code>/opt/sun/mq/share/lib/</code>
共用程式庫(.so)檔案	<code>/opt/sun/mq/lib/</code>

Windows

表 A-3 顯示 Windows 作業系統上 Message Queue 資料的位置。當 Message Queue 隨附於獨立版本的 Sun Java System Application Server 時，此表格也適用於 Solaris 平台。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，但將斜線字元的方向從 Windows 反斜線 (\) 變更為 Solaris 正斜線 (/)。請參閱第 22 頁的「目錄變數慣例」以取得 IMQ_HOME 和 IMQ_VARHOME 目錄變數的定義。

表 A-3 Windows 平台上的 Message Queue 資料位置

資料種類	位置
代理程式實例配置檔案	IMQ_VARHOME\instances\ <i>instanceName</i> \props\config.properties
代理程式配置檔案範本	IMQ_HOME\lib\props\broker\
永久存放區 (訊息、目標、長期訂閱、作業事件)	IMQ_VARHOME\instances\ <i>instanceName</i> \fs350\ 或 JDBC 可存取之資料存放區
代理程式實例記錄檔目錄 (預設位置)	IMQ_VARHOME\instances\ <i>instanceName</i> \log\
管理物件 (物件存放區)	您選擇之本機目錄或 LDAP 伺服器
安全性：使用者信託庫	IMQ_VARHOME\instances\ <i>instanceName</i> \etc\passwd 或 LDAP 伺服器
安全性：存取控制檔 (預設位置)	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\
安全性：範例密碼檔案	IMQ_HOME\etc\passfile.sample
安全性：代理程式金鑰存放區檔案位置	IMQ_HOME\etc\
JavaDoc API 文件	IMQ_HOME\javadoc\index.html
範例應用程式與配置	IMQ_HOME\demo\
Java 歸檔 (.jar)、Web 歸檔 (.war) 與資源 介庫歸檔 (.rar) 檔案	IMQ_HOME\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 介面穩定性分類機制

分類	說明
私有	不由客戶直接使用。可在任何版本中變更或移除。
改進中	由客戶使用。用於主要版本（如 3.0、4.0）或次要版本（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。僅用於主要版本（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。用於主要版本（如 3.0、4.0）或次要版本（如 3.1、3.2）的不相容變更。通知客戶在未來版本中，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
指令行介面	
imqbrokerd 指令行介面	改進中
imqadmin 指令行介面	不穩定
imqcmd 指令行介面	改進中
imqdbmgr 指令行介面	不穩定
imqkeytool 指令行介面	改進中
imqobjmgr 指令行介面	改進中
imqusermgr 指令行介面	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
訊息型監視 API	改進中
管理物件 API (com.sun.messaging)	改進中
.jar 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

表 B-2 Message Queue 介面的穩定性 (續)

介面	分類
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
saa-j-api.jar 位置和名稱	改進中
saa-j-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
權限	
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.dmqs 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
訊息特性和格式	
停用訊息佇列訊息特性, JMSXDeliveryCount	標準
停用訊息佇列訊息特性, JMS_SUN_*	改進中
Message Queue 客戶端訊息特性: JMS_SUN_*	改進中
度量或監視訊息的 JMS 訊息格式	改進中
其他	

表 B-2 Message Queue 介面的穩定性 (續)

介面	分類
Message Queue JMS 資源介面安裝載體, 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 HTTP 型連線進行 Message Queue 訊息傳送所需的設定工作。它包含下列各節：

- [第 328 頁的「HTTP/HTTPS 支援架構」](#)
- [第 329 頁的「啓用 HTTP 支援」](#)
- [第 336 頁的「啓用 HTTPS 支援」](#)
- [第 345 頁的「疑難排解」](#)

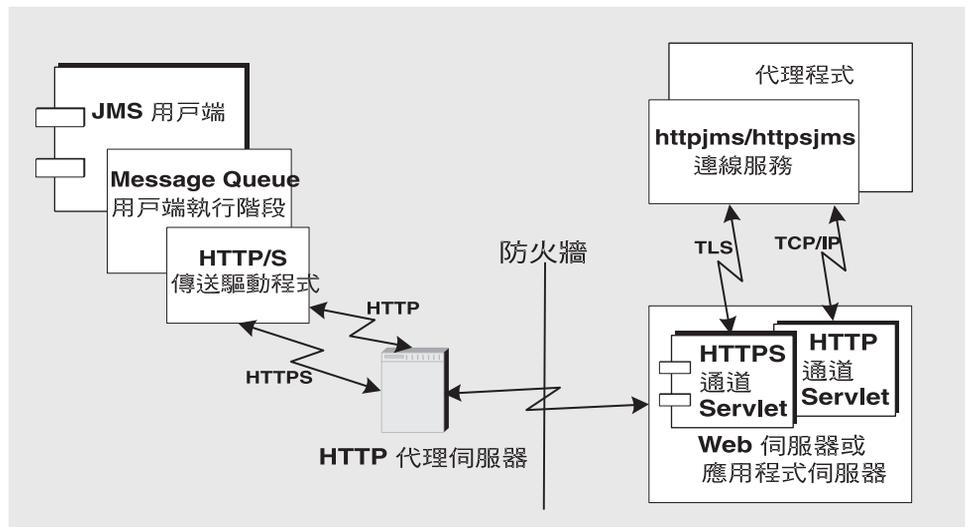
HTTP/HTTPS 支援架構

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 328 頁的圖 C-1 顯示提供 HTTP/HTTPS 支援所需的主要元件。

- 在用戶端，HTTP 或 HTTPS 傳輸驅動程式會將 Message Queue 訊息封裝在 HTTP 請求中，並確保這些請求會以正確的順序傳送至 Web 伺服器 / 應用程式伺服器。
- 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。代理的地址在啟動用戶端時使用命令行選項指定。請參閱第 335 頁的「使用 HTTP 代理」，以獲得更多資訊。
- HTTP 或 HTTPS 通道 Servlet (均隨附於 Message Queue) 會載入 Web 伺服器 / 應用程式伺服器中，並在將有效負載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，並對從相應通道 Servlet 傳送的內送訊息取消多工處理。

- 如果 Web 伺服器 / 應用程式伺服器發生故障並重新啟動，則所有連線均會復原且不會影響用戶端。如果代理程式發生故障並重新啟動，則系統會發生異常，且用戶端必須重新建立其連線。如果 Web 伺服器 / 應用程式伺服器與代理程式均發生故障，且未重新啟動代理程式 (這種情況不太可能發生)，則 Web 伺服器 / 應用程式伺服器會復原用戶端連線，並繼續等待代理程式連線，且不會通知用戶端。若要避免這種情況，請務必重新啟動代理程式。

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 會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。一旦建立了此連線，用戶端應用程式與通道 Servlet 之間的安全連線即可由用戶端應用程式和 Web 伺服器 / 應用程式伺服器協商。

啟用 HTTP 支援

以下各節描述啟用 HTTP 支援所需的步驟。

► 啟用 HTTP 支援

1. 部署 HTTP 通道 Servlet。您可以在下列伺服器上部署 HTTP 通道 Servlet：
 - Sun Java System Web Server
 - Sun Java System Application Server
2. 配置代理程式的 httpjms 連線服務，並啟動代理程式。
3. 配置 HTTP 連線。

步驟 1：部署 HTTP 通道 Servlet

您可以將 HTTP 通道 Servlet 部署為 Sun Java System Web Server 或 Sun Java System Application Server 上的 Web 歸檔 (.war) 檔案。

將 HTTP 通道 Servlet 部署為 .war 檔案，需要使用 Web 伺服器 / 應用程式伺服器所提供的部署機制。HTTP 通道 Servlet .war 檔案 (imghttp.war) 位於包含 .jar、.war 和 .rar 檔案的目錄中，同時目錄位置會因作業系統而異 (請參閱附錄 A 「Message Queue 資料的平台特定位置」)。

.war 檔案包括部署描述元，而該描述元會包含 Web 伺服器 / 應用程式伺服器載入和執行 Servlet 所需之基本配置資訊。因 Web 伺服器 / 應用程式伺服器的不同，您也可能必須指定 Servlet URL 的環境根目錄部分。

部署為 Web 歸檔檔案

如需在 Sun Java System Web Server 上部署的相關資訊，請參閱第 330 頁的「在 Sun Java System Web Server 上部署 HTTP 通道 Servlet」。

如需在 Sun Java System Application Server 上部署的相關資訊，請參閱第 331 頁的「在 Sun Java System Application Server 上部署 HTTP 通道 Servlet」。

在 Sun Java System Web Server 上部署 HTTP 通道 Servlet

以下說明代表 Sun Java System Web Server 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. 在瀏覽器型的管理 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) 中找到此檔案 (請參閱附錄 A 「Message Queue 資料的平台特定位置」)。

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 之 `/contextRoot` 的一部分：

```
http://hostName:portNumber/contextRoot/tunnel
```

例如，如果您將 `contextRoot` 設定至 `img`，則 [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 伺服器實例。

此時，Servlet 可從以下位址獲得：

```
http://hostName:portNumber/contextRoot/tunnel
```

此時，用戶端可使用此 URL 連線至使用 HTTP 連線的訊息服務。

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 [記錄喜好設定頁面]。

使用記錄用戶端存取控制來停用記錄。

在 Sun Java System Application Server 上部署 HTTP 通道 Servlet

本節描述如何在 Sun Java System Application Server 上將 HTTP 通道 Servlet 部署為 `.war` 檔案，接著再配置通道 Servlet，接受來自 Message Queue 代理程式的連線。

您需要執行下列兩個步驟：

- 使用應用程式伺服器部署工具來部署 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

► 在應用程式伺服器環境中部署 HTTP 通道 Servlet

1. 在網路型管理 GUI 中，選擇
[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，再按一下 [確定]。

`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 資料的平台特定位置](#)」)。

4. 於 [環境根目錄] 文字欄位中設定數值，再按一下 [確定]。

[環境根目錄] 欄位值為通道 Servlet URL 之 `/contextRoot` 的一部分：

```
http://hostName:portNumber/contextRoot/tunnel
```

例如，您可以將 [環境根目錄] 欄位設定為 `/imq`。

預設會啟用在通道 Servlet 已成功部署時出現確認畫面，在本例中位於：

```
/var/opt/SUNWappserver8/domains/domain1/server1/applications/  
j2ee-modules/imqhttp_1
```

此時，Servlet 可從以下 URL 取得：

```
http://hostName:portNumber/contextRoot/tunnel
```

此時，用戶端可使用此 URL 連線至使用 HTTP 連線的訊息服務。

修改 `server.policy` 檔案

應用程式伺服器會實施一組預設的安全策略，除非將其修改，否則這組安全策略可以阻止 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` 實例的檔案位置為：

```
/var/opt/SUNWappserver8/domains/domain1/server1/config/  
server.policy
```

若要配置通道 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。

► **修改 Application Server 的伺服器策略檔案**

1. 開啓 server.policy 檔案。
2. 新增以下項目：

```
grant codeBase
"file:/var/opt/SUNWappserver8/domains/domain1/server1/
  applications/j2ee-modules/imqhttp_1/-"
{
  permission java.net.SocketPermission "*",
    "connect,accept,resolve";
};
```

步驟 2：配置 httpjms 連線服務

預設情況下，代理程式不會啓動 HTTP 支援，所以您必須重新配置代理程式，以啓動 httpjms 連線服務。一旦重新配置，此代理程式便可啓動，如第 67 頁的「啓動代理程式」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 資料的平台特定位置」)：

.../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，代理程式會查詢在其主機上執行的 Web 伺服器 / 應用程式伺服器和 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。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	描述
imq.httpjms.http.servletHost	如非必要，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) 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：localhost。
imq.httpjms.http.servletPort	變更此值，以指定代理程式用於存取 HTTP 通道 Servlet 的連接埠號碼。(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：7675。
imq.httpjms.http.pullPeriod	指定客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，且傳遞至客戶端執行階段。) 如果值為零或負數，則客戶端將始終擱置一個 HTTP 請求，以準備儘快移除訊息。若客戶端數目過大，可能會導致 Web 伺服器/應用程式伺服器資源大量外流，且伺服器可能沒有反應。在此類情況下，您應將 pullPeriod 特性設定為正的秒數。此作業將設定客戶端的 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節省 Web 伺服器/應用程式伺服器資源，但會影響客戶端所監視的回應次數。預設值：-1。
imq.httpjms.http.connectionTimeout	指定客戶端執行階段在系統發生異常前，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，且傳遞至客戶端執行階段。) 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 的客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 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

若要啟用 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:portNumber/contextRoot/tunnel
```

其中 `hostName:portNumber` 為代管 HTTP 通道 Servlet 之 Web 伺服器 / 應用程式伺服器的名稱和連接埠；`contextRoot` 為在 Web 伺服器 / 應用程式伺服器上部署通道 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管理物件 (請參閱第 164 頁的「[新增連線工廠](#)」) 之 `imqobjmgr` 指令的 `-o` 選項，或在使用管理主控台 (`imqadmin`) 建立連線工廠管理物件時設定屬性。
- 使用可啓動用戶端之指令的 `-D` 選項 (請參閱「[Message Queue Developer's Guide for Java Clients](#)」)。
- 當您透過程式設計由用戶端程式碼建立連線工廠後，請使用 API 呼叫設定屬性 (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 伺服器 / 應用程式伺服器和 Servlet 實例。您可以在同時執行的代理程式中共用單一 Web 伺服器 / 應用程式伺服和 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:portNumber/contextRoot/tunnel?ServerName=bkrHostName:instanceName
```

其中 `bkrHostName` 為代理程式實例主機名稱；`instanceName` 為用戶端存取의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 2005
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
  cochín: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。

您可以使用可啟動用戶端應用程式之指令的 `-D` 選項來設定這些特性。

啟用 HTTPS 支援

下列各節描述啟用 HTTP 支援的步驟。它們與第 329 頁的「啟用 HTTP 支援」中的步驟，以及產生和存取 SSL 憑證所需的步驟相似。

► 啟用 HTTPS 支援

1. 為 HTTPS 通道 Servlet 產生自身簽名的憑證。
2. 修改 HTTP 通道 Servlet `.war` 檔案的部署描述元至：
 - 指向您放置憑證金鑰存放區的位置
 - 指定憑證金鑰存放區密碼
3. 部署 HTTP 通道 Servlet。您可以在下列伺服器上部署 HTTP 通道 Servlet：
 - Sun Java System Web Server
 - Sun Java System Application Server
4. 配置代理程式的 `httpsjms` 連線服務並啟動代理程式。
5. 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施 SSL。在 `httpsjms` 連線服務架構中，HTTPS 通道 Servlet 是代理程式與應用程式用戶端的伺服器。

執行 `keytool` 公用程式，以為通道 Servlet 產生自身簽名的憑證。在指令提示符號下輸入以下內容：

```
JRE_HOME/bin/keytool -servlet keyStoreLocation
```

此公用程式將提示您提供所需的資訊。(在 Unix 系統上，需要以超級使用者 (root) 的身份執行 keytool，才擁有建立金鑰存放區的權限。)

首先，keytool 會提示您提供金鑰存放區密碼，以及某些組織資訊，然後提示您確認。接收到確認之後，它會在產生金鑰對時暫停。然後它會要求您提供密碼，以鎖定特定的成對金鑰 (金鑰密碼)；您應按 Return 鍵以回應此提示：這會使金鑰密碼與金鑰存放區密碼相同。

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，以便它可開啓金鑰存放區。

JDK keytool 公用程式會產生自身簽名憑證，並將其放置在 Message Queue 的金鑰存放區檔案，該檔案位於 *keyStoreLocation* 引數所指定的位置。

備註 HTTPS 通道 Servlet 必須可查看此金鑰存放區。請確定您將位於 *keyStoreLocation* 中所產生的金鑰存放區移動 / 複製到 HTTPS 通道 Servlet 可存取的位置 (請參閱第 338 頁的「[步驟 3：部署 HTTPS 通道 Servlet](#)」)。

步驟 2：修改 HTTP 通道 Servlet .war 檔案的描述元檔案

HTTP 通道 Servlet 的 .war 檔案包括部署描述元，其包含 Web 伺服器 / 應用程式伺服器負載和執行 Servlet 所需的基本配置資訊。

imqhttps.war 檔案的部署描述元，無法知道您已將通道 Servlet 所需的金鑰存放區檔案置於何處。所以您需要編輯通道 Servlet 的部署描述元 (XML 檔案)，以在部署 imqhttps.war 檔案之前指定金鑰存放區的位置和密碼。

► 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，以便為 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。(如果您擔心金鑰存放區密碼洩漏，則可以使用檔案系統權限來限制其他人對 imqhttps.war 檔案的存取。)

步驟 3：部署 HTTPS 通道 Servlet

您可以將 HTTP 通道 Servlet 部署為 Sun Java System Web Server 或 Sun Java System Application Server 上的 Web 歸檔 (WAR) 檔案。

將 HTTPS 通道 Servlet 部署為 .war 檔案，需要使用 Web 伺服器 / 應用程式伺服器提供的部署機制。HTTPS 通道 Servlet .war 檔案 (imqhttps.war) 位於您作業系統的某個目錄中，但目錄位置依作業系統而異 (請參閱附錄 A 「[Message Queue 資料的平台特定位置](#)」)。

您應確保已為 Web 伺服器啟動加密，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 Web 歸檔檔案

如需在 Sun Java System Web Server 上部署的相關資訊，請參閱第 339 頁的「在 Sun Java System Web Server 上部署 HTTPS 通道 Servlet」。

如需在 Sun Java System Application Server 上部署的相關資訊，請參閱第 340 頁的「在 Sun Java System Application Server 上部署 HTTPS 通道 Servlet」。

在 Sun Java System Web Server 上部署 HTTPS 通道 Servlet

本節描述如何將 HTTPS 通道 Servlet 部署為 Sun Java System Web Server 上的 .war 檔案。您可以透過使用 Web 瀏覽器存取 Servlet URL，來驗證已成功地部署 HTTPS 通道 Servlet。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 伺服器的類別路徑中。執行此作業的最簡單方法是將 jsse.jar、jnet.jar 和 jcert.jar 複製到 WebServer_TOPDIR/bin/https/jre/lib/ext。

► 將 HTTPS 通道 Servlet 部署為 .war 檔案

1. 在瀏覽器型的管理 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. 為 [WAR 檔案開啓] 和 [WAR 檔案路徑] 欄位選取適當的值，以便指向已修改的 imqhttps.war 檔案 (請參閱第 338 頁的「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] 欄位值為通道 Servlet URL 之 /contextRoot 的一部分：

```
https://hostName:portNumber/contextRoot/tunnel
```

舉例來說，如果您將 contextRoot 設定至 imq，[應用程式 URI] 欄位應為：

```
/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. 重新啟動 Web Server 實例。

此時，Servlet 可從以下 URL 獲得：

```
https://hostName:portNumber/img/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。

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 [記錄喜好設定頁面]。

使用記錄用戶端存取控制來停用記錄。

在 Sun Java System Application Server 上部署 HTTPS 通道 Servlet
本節描述如何將 HTTPS 通道 Servlet 部署為 Sun Java System Application Server 上的 .war 檔案。

您需要執行下列兩個步驟：

- 使用應用程式伺服器部署工具，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

► 在應用程式伺服器環境中部署 HTTPS 通道 Servlet

1. 在瀏覽器型的管理 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 (imghttps.war) 的位置，再按一下 [確定]。

imghttps.war 檔案的位置會因您的作業系統而異 (請參閱附錄 A 「[Message Queue 資料的平台特定位置](#)」)。

4. 於 [環境根目錄] 文字欄位中設定數值，再按一下 [確定]。

[環境根目錄] 欄位值為通道 Servlet URL 的 `/contextRoot` 的一部分：

```
https://hostName:portNumber/contextRoot/tunnel
```

例如，您可以將 [環境根目錄] 欄位設定為：

```
/img
```

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/SUNWappserver8/domains/domain1/server1/applications/  
j2ee-modules/imghttps_1
```

此時，Servlet 可從以下 URL 獲得：

```
https://hostName:portNumber/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

應用程式伺服器會實施一組預設的安全策略，除非將其修改，否則這組安全策略可以阻止 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/SUNWappserver8/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。

► 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/SUNWappserver8/domains/domain1/server1/  
    applications/j2ee-modules/imghttps_1/-"  
{  
    permission java.net.SocketPermission "*",  
        "connect,accept,resolve";  
};
```

步驟 4：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 67 頁的「[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 資料的平台特定位置](#)」)：

```
.../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，代理程式會查詢在其主機上執行的 Web 伺服器 and 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-2 中詳細介紹了 httpsjms 連線服務配置特性。

表 C-2 httpsjms 連線服務特性

特性	描述
imq.httpsjms.https.servletHost	如非必要，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：localhost。
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	指定每個客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，且傳遞至客戶端執行階段。) 如果值為零或負數，則客戶端將始終攔截一個 HTTP 請求，以準備儘快移除訊息。若客戶端數目過大，可能會導致 Web 伺服器資源大量外流，且伺服器可能沒有反應。在此類情況下，您應將 pullPeriod 特性設定為正的秒數。此作業將設定客戶端的 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節省 Web 伺服器資源，但會影響客戶端所發出的回應次數。預設值：-1。

表 C-2 httpsjms 連線服務特性 (續)

特性	描述
imq.httpsjms.https.connectionTimeout	指定客戶端執行階段在系統產生異常前，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，且傳遞至客戶端執行階段。)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 的客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 5：配置 HTTPS 連線

用戶端應用程式必須使用已適當配置的連線工廠受管理物件，以建立與代理程式的 HTTPS 連線。

但是，用戶端還必須對 Java 安全套接延伸 (JSSE) 提供的 SSL 程式庫擁有存取權，並且還必須具有超級使用者憑證。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「[匯入超級使用者憑證](#)」。

解決這些問題後，即可以繼續配置 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
```

匯入超級使用者憑證

如果 CA (簽名 Web 伺服器憑證的單位) 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 伺服器 / 應用程式伺服器憑證，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，否則請參考「[配置連線工廠](#)」中的說明。

假設此憑證儲存在 *certFile* 中，且 *trustStoreFile* 為您的金鑰存放區，請執行以下指令：

```
JRE_HOME/bin/keytool -import -trustcacerts
  -alias aliasForCertificate -file certFile
  -keystore trustStoreFile
```

對以下問題回答 YES：是否信任此憑證？

您還需要使用可啓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=trustStoreFile
javax.net.ssl.trustStorePassword=trustStorePasswd
```

配置連線工廠

若要啓用 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:portNumber/contextRoot/tunnel
```

其中 *hostName:portNumber* 為代管 HTTPS 通道 Servlet 之 Web 伺服器名稱和連接埠；*contextRoot* 為在 Web 伺服器上部署通道 Servlet 時的路徑設定。

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 164 頁的「[新增連線工廠](#)」) 之 *imqobjmgr* 指令的 -o 選項，或在使用管理主控台 (*imqadmin*) 建立連線工廠受管理物件時設定屬性。
- 使用可啓動用戶端應用程式之指令的 -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 伺服器 and Servlet 實例。您可以在同時執行的代理程式中共用單一 Web 伺服器和 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:portNumber/contextRoot/tunnel?ServerName=bkrHostName:instanceName
```

其中 `bkrHostName` 為代理程式實例主機名稱；`instanceName` 為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。

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 伺服器發生故障並重新啓動，則所有連線均會復原且不會影響用戶端。但是，如果代理程式發生故障並重新啓動，則系統會產生異常，且用戶端必須重新建立它們的連線。

如果 Web 伺服器與代理程式均發生故障，且未重新啓動代理程式，則 Web 伺服器會復原用戶端連線，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 以手動方式存取 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。

常見指令 公用程式指令

本附錄列出常用之 Message Queue 指令公用程式 (imqcmd) 指令。如需指令行可用之指令選項和屬性的詳細清單，請參閱第 13 章「指令行參照」中的第 252 頁的「指令公用程式」。

語法

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

-H 或 -h 會提供詳細的說明。-v 子命令提供版本資訊。

當您使用 `imqcmd` 時，指令公用程式會提示您輸入密碼。若要避免出現提示 (同時增加安全性)，您可以使用 `-passfile pathToPassfile` 選項，將公用程式指向包含管理員使用者名稱和密碼的密碼檔案。

範例：`imqcmd query bkr -u adminUserName -passfile pathToPassfile -b myServer:7676`

代理程式和叢集管理

```
imqcmd query bkr
imqcmd pause bkr
imqcmd restart bkr
imqcmd resume bkr
imqcmd shutdown bkr -b myBroker:7676
imqcmd update bkr -o "imq.system.max_count=1000"
imqcmd reload cls
```

代理程式配置特性 (-o 選項)

表 D-1 會列出常用代理程式配置特性。如需代理程式配置特性及其說明的完整清單，請參閱第 14 章「代理程式特性參照」。

表 D-1 代理程式配置特性 (-o 選項)

特性	備註
<code>imq.autocreate.queue</code>	
<code>imq.autocreate.queue.maxNumActiveConsumers</code>	指定 -1 表示無限制
<code>imq.autocreate.queue.maxNumBackupConsumers</code>	指定 -1 表示無限制

表 D-1 代理程式配置特性 (-o 選項) (續)

特性	備註
imq.autocreate.topic	
imq.cluster.url	
imq.destination.DMQ.truncateBody	
imq.destination.logDeadMessages	
imq.log.file.rolloverbytes	指定 -1 表示無限制
imq.log.file.rolloversecs	指定 -1 表示無限制
imq.log.level	NONE ERROR WARNING INFO
imq.message.max_size	指定 -1 表示無限制
imq.portmapper.port	
imq.system.max_count	指定 -1 表示無限制
imq.system.max_size	指定 -1 表示無限制

服務和連線管理

```

imqcmd list svc
imqcmd query svc
imqcmd update svc -n jms -o "minThreads=200" -o "maxThreads=400" -o
"port=8995"
imqcmd pause svc -n jms
imqcmd resume svc -n jms
imqcmd list cxn -svn jms
imqcmd query cxn -n 1234567890

```

長期用戶管理

```
imqcmd list dur -d MyTopic
imqcmd destroy dur -n myDurSub -c "clientID-111.222.333.444"
imqcmd purge dur -n myDurSub -c "clientID-111.222.333.444"
```

作業事件管理

```
imqcmd list txn
imqcmd commit txn -n 1234567890
imqcmd query txn -n 1234567890
imqcmd rollback txn -n 1234567890
```

目標管理

```
imqcmd create dst -n MyQueue -t q -o "maxNumMsgs=1000" -o
"maxNumProducers=5"
imqcmd update dst -n MyTopic -t t -o "limitBehavior=FLOW_CONTROL|
REMOVE_OLDEST|REJECT_NEWEST|REMOVE_LOW_PRIORITY"
imqcmd compact dst -n MyQueue -t q
imqcmd purge dst -n MyQueue -t q
imqcmd pause dst -n MyQueue -t q -pst PRODUCERS|CONSUMERS|ALL
imqcmd resume dst -n MyQueue -t q
imqcmd destroy dst -n MyQueue -t q
imqcmd query dst -n MyQueue -t q
imqcmd list dst -tmp
```

目標配置特性 (-o 選項)

表 D-2 會列出常用目標配置特性。如需目標配置特性及其說明的完整清單，請參閱第 15 章「實體目標特性參照」。

表 D-2 目標配置特性 (-o 選項)

特性	備註
consumerFlowLimit	指定 -1 表示無限制
isLocalOnly (僅限建立)	
limitBehavior	FLOW_CONTROL REMOVE_OLDEST REJECT_NEWEST REMOVE_LOW_PRIORITY
localDeliveryPreferred (僅限佇列)	
maxNumActiveConsumers (僅限佇列)	指定 -1 表示無限制
maxNumBackupConsumers (僅限佇列)	指定 -1 表示無限制
maxBytesPerMsg	指定 -1 表示無限制
maxNumMsgs	指定 -1 表示無限制
maxNumProducers	指定 -1 表示無限制
maxTotalMsgBytes	指定 -1 表示無限制
useDMQ	

標準資訊

```
imqcmd metrics bkr -m cxn|rts|ttl -int 5 -msp 20
imqcmd metrics svc -m cxn|rts|ttl
imqcmd metrics dst -m con|dsk|rts|ttl
```


字彙表

如需 Message Queue 專有名詞的詳細資訊，請參閱「Message Queue 技術摘要」中的字彙表。如需 Sun Java System 產品套件中使用的專有名詞完整清單，請參閱「Java Enterprise System 字彙表」(<http://docs.sun.com/doc/819-4632>)。

A

acknowledgeMode 啓動規格屬性 306
ActivationSpec JavaBean 306
addressList 啓動規格屬性 306
addressList 資源介面屬性 304, 305
addressList 管理式連線工廠屬性 305
addressListBehavior 資源介面屬性 304
addressListBehavior 管理式連線工廠屬性 305
addressListIterations 資源介面屬性 304
addressListIterations 管理式連線工廠屬性 305
admin 使用者 127, 131, 133
ADMIN 服務類型 74
admin 連線服務 74, 102
admin 群組 129
anonymous 群組 129
API 文件 320, 321, 322
AUTOSTART 特性 68

C

clientId 啓動規格屬性 306, 307
clientID 管理式連線工廠屬性 305
config.properties 檔案 87, 174, 175, 176
connectionURL 資源介面屬性 304

customAcknowledgeMode 啓動規格屬性 307

D

default.properties 檔案 87
destination 啓動規格屬性 307
destinationType 啓動規格屬性 307

E

endpointExceptionRedeliveryAttempts 啓動規格屬性 307
/etc/hosts 檔案 (Linux) 174

G

guest 使用者 127

H

hosts 檔案 (Linux) 174
HTTP

- 支援架構 328
 - 代理 328
 - 連線服務, 請參閱 `httpjms` 連線服務
 - 傳輸驅動程式 328
 - HTTP 通道 Servlet
 - 部署 329
 - 關於 328
 - HTTP 連線
 - 支援 328
 - 多重代理程式, 用於 335
 - 通道 Servlet, 請參閱 HTTP 通道 Servlet
 - 請求間隔時間 334
 - `httpjms` 連線服務
 - 配置 333
 - 設定 329
 - 關於 74, 102
 - HTTPS
 - 支援架構 328
 - 連線服務, 請參閱 `httpsjms` 連線服務
 - HTTPS 通道 Servlet
 - 部署 338
 - 關於 328
 - HTTPS 連線
 - 支援 328
 - 多重代理程式, 用於 345
 - 通道 Servlet, 請參閱 HTTPS 通道 Servlet
 - 請求間隔時間 342
 - `httpsjms` 連線服務
 - 配置 342
 - 設定 336
 - 關於 74, 102
-
- `imq.accesscontrol.enabled` 特性 81, 276, 284
 - `imq.accesscontrol.file.filename` 特性 81, 276, 284
 - `imq.audit.enabled` 特性 84, 151, 279, 284
 - `imq.authentication.basic.user_repository` 特性 82, 277, 284
 - `imq.authentication.client.response.timeout` 特性 82, 277, 284
 - `imq.authentication.type` 特性 82, 276, 284
 - `imq.autocreate.destination.isLocalOnly` 特性 271, 284
 - `imq.autocreate.destination.limitBehavior` 特性 270, 284
 - `imq.autocreate.destination.maxBytesPerMsg` 特性 269, 284
 - `imq.autocreate.destination.maxCount` 特性 269, 284
 - `imq.autocreate.destination.maxNumMsgs` 特性 269, 284
 - `imq.autocreate.destination.maxNumProducers` 特性 270, 284
 - `imq.autocreate.destination.maxTotalMsgBytes` 特性 269, 271, 285
 - `imq.autocreate.destination.useDMQ` 特性 122
 - `imq.autocreate.queue` 特性 98, 269, 285
 - `imq.autocreate.queue.consumerFlowLimit` 特性 270, 271, 285
 - `imq.autocreate.queue.localDeliveryPreferred` 特性 271, 285
 - `imq.autocreate.queue.maxNumActiveConsumers` 特性 98, 270, 285
 - `imq.autocreate.queue.maxNumBackupConsumers` 特性 98, 270, 285
 - `imq.autocreate.topic` 特性 98, 269, 285
 - `imq.cluster.brokerlist` 特性 172, 173, 174, 175, 176, 283, 285
 - `imq.cluster.masterbroker` 特性 172, 175, 176, 284, 285
 - `imq.cluster.port` 特性 172, 283, 285
 - `imq.cluster.transport` 特性 172, 174, 175, 283, 285
 - `imq.cluster.url` 特性 98, 172, 173, 174, 175, 176, 284, 285
 - `imq.destination.DMQ.truncateBody` 特性 77, 98, 268, 285
 - `imq.destination.logDeadMsgs` 特性 85, 98, 280, 285
 - `imq.hostname` 特性 75, 266, 285
 - `imq.httpjms.http.connectionTimeout` 特性 334

- imq.httpjms.http.pullPeriod 特性 334
- imq.httpjms.http.servletHost 特性 334
- imq.httpjms.http.servletPort 特性 334
- imq.httpsjms.https.connectionTimeout 特性 343
- imq.httpsjms.https.pullPeriod 特性 342
- imq.httpsjms.https.servletHost 特性 342
- imq.httpsjms.https.servletPort 特性 342
- imq.imqcmd.password 特性 83, 277, 285
- imq.keystore.file.dirpath 特性 143, 279, 285
- imq.keystore.file.name 特性 143, 285
- imq.keystore.password 特性 84, 143, 150, 285
- imq.keystore.*property_name* 特性 285
- imq.log.console.output 特性 85, 280, 285
- imq.log.console.stream 特性 85, 280, 285
- imq.log.file.dirpath 特性 85, 280, 285
- imq.log.file.filename 特性 85, 280, 285
- imq.log.file.output 特性 85, 281, 285
- imq.log.file.rolloverbytes 特性 85, 98, 281, 285
- imq.log.file.rolloversecs 特性 85, 98, 281, 285
- imq.log.level 特性 85, 98, 280, 286
- imq.log.syslog.facility 特性 282, 286
- imq.log.syslog.identity 特性 282, 286
- imq.log.syslog.logconsole 特性 282, 286
- imq.log.syslog.logpid 特性 282, 286
- imq.log.syslog.output 特性 85, 281, 286
- imq.log.timezone 特性 282, 286
- imq.message.expiration.interval 特性 77, 268, 286
- imq.message.max_size 特性 77, 98, 268, 286
- imq.metrics.enabled 特性 85, 282, 286
- imq.metrics.interval property 86
- imq.metrics.interval 特性 85, 282, 286
- imq.metrics.topic.enabled 特色 283, 286
- imq.metrics.topic.interval property 86
- imq.metrics.topic.interval 特色 283, 286
- imq.metrics.topic.persist property 86
- imq.metrics.topic.persist 特性 283, 286
- imq.metrics.topic.timetolive property 86
- imq.metrics.topic.timetolive 特性 283, 286
- imq.passfile.dirpath 特性 83, 277, 286
- imq.passfile.enabled 特性 83, 277, 286
- imq.passfile.name 特性 83, 277, 286
- imq.persist.file.destination.message.filepool.limit 特性 79, 272, 286
- imq.persist.file.message.cleanup 特性 79, 273, 286
- imq.persist.file.message.filepool.cleanratio 特性 79, 272, 286
- imq.persist.file.message.max_record_size 特性 272, 286
- imq.persist.file.message.vrfile.max_record_size 特性 79
- imq.persist.file.sync 特性 89
- imq.persist.file.sync.enabled 特性 80, 273, 286
Sun Cluster 需求 273, 286
- imq.persist.jdbc.brokerid 特性 80, 273, 286
- imq.persist.jdbc.closedburl 特性 80, 274, 286
- imq.persist.jdbc.createdburl 特性 80, 274, 286
- imq.persist.jdbc.driver 特性 80, 274, 286
- imq.persist.jdbc.needpassword 特性 80, 274, 286
- imq.persist.jdbc.opendburl 特性 80, 274, 286
- imq.persist.jdbc.password 特性 80, 150, 274, 286
- imq.persist.jdbc.table.IMQCCREC35 特性 81, 274, 287
- imq.persist.jdbc.table.IMQDEST35 特性 81, 274, 287
- imq.persist.jdbc.table.IMQINT35 特性 275, 287
- imq.persist.jdbc.table.IMQLIST35 特性 275, 287
- imq.persist.jdbc.table.IMQMSG35 特性 275, 287
- imq.persist.jdbc.table.IMQPROPS35 特性 275, 287
- imq.persist.jdbc.table.IMQSV35 特性 81, 274, 287
- imq.persist.jdbc.table.IMQTACK35 特性 275, 287
- imq.persist.jdbc.table.IMQTXN35 特性 275, 287
- imq.persist.jdbc.user 特性 80, 274, 287
- imq.persist.store 特性 78, 90, 271, 287
- imq.ping.interval 特性 76, 267, 287

- imq.portmapper.backlog 特性 75, 266, 287
- imq.portmapper.hostname 特性 75, 266, 287
- imq.portmapper.port 特性 75, 98, 266, 287
- imq.protocol *protocolType* inbufsz 212
- imq.protocol *protocolType* nodelay 212
- imq.protocol *protocolType* outbufsz 212
- imq.resource_state.count 特性 268, 287
- imq.resource_state.threshold 特性 268, 287
- imq.resourceState.count 特性 77
- imq.service.activelist 特性 75, 266, 287
- imq.service_name.accesscontrol.enabled 特性 276, 287
- imq.service_name.accesscontrol.file.filename 特性 276, 287
- imq.service_name.authentication.type 特性 277, 287
- imq.service_name.max_threads 特性 267, 287
- imq.service_name.min_threads 特性 266, 287
- imq.service_name.protocol_type.hostname 特性 172, 266, 283, 285, 287
- imq.service_name.protocol_type.port 特性 266, 287
- imq.service_name.threadpool_model 特性 266, 287
- imq.serviceName.accesscontrol.enabled 特性 81
- imq.serviceName.accesscontrol.file.filename 特性 81
- imq.serviceName.authentication.type 特性 82
- imq.serviceName.max_threads 特性 76
- imq.serviceName.min_threads 特性 76
- imq.serviceName.protocolType.hostname 特性 75
- imq.serviceName.protocolType.port 特性 75
- imq.serviceName.threadpool_model 特性 76
- imq.shared.connectionMonitor_limit 特性 76, 267, 287
- imq.system.max_count 特性 77, 98, 267, 287
- imq.system.max_size 特性 77, 98, 268, 287
- imq.transaction.autorollback 特性 110, 268, 287
- imq.user_repository.ldap.base 特性 278, 288
- imq.user_repository.ldap.gidattr 特性 279, 288
- imq.user_repository.ldap.grpbase 特性 279, 288
- imq.user_repository.ldap.grpfilter 特性 279, 288
- imq.user_repository.ldap.grpsearch 特性 279, 288
- imq.user_repository.ldap.memattr 特性 279, 288
- imq.user_repository.ldap.password 特性 83, 150, 278, 288
- imq.user_repository.ldap.principal 特性 83, 278, 288
- imq.user_repository.ldap.property_name 特性 278, 288
- imq.user_repository.ldap.server 特性 83, 278, 288
- imq.user_repository.ldap.ssl.enabled 特性 279, 288
- imq.user_repository.ldap.timeout 特性 279, 288
- imq.user_repository.ldap.uidattr 特性 279, 288
- imq.user_repository.ldap.usrfilter 特性 279, 288
- imqAckTimeout 屬性 297
- imqAddressList 屬性 294
- imqAddressListBehavior 屬性 294
- imqAddressListIterations 屬性 294
- imqbrokerd 指令 67
 - 在密碼檔案中 149
 - 配置檔案 (Solaris、Linux) 68, 71
 - 參照 249
 - 將代理程式新增至叢集 175
 - 從叢集中移除代理程式 175
 - 清除資料存放區 89
 - 清除資料儲存 119
 - 移除代理程式 71
 - 設定記錄特性 183
 - 連線代理程式 173
 - 備份配置變更記錄 177
 - 復原配置變更記錄 177
 - 傳送引數至 88
 - 選項 249
 - 關於 32
- imqbrokerd.conf 檔案 68, 71
- imqcmd 指令
 - 一般選項 258, 263
 - 在密碼檔案中 149
 - 作業事件管理 108
 - 取決於主代理程式 177

- 長期訂閱子指令 107
 - 度量監視 186
 - 參照 252
 - 實體目標子指令 (表) 113
 - 實體目標管理 111
 - 與代理程式的安全連線 146, 258
 - 關於 32
 - imqConfiguredClientID 屬性 297
 - imqConnectionFlowCount 屬性 298
 - imqConnectionFlowLimit 屬性 298
 - imqConnectionFlowLimitEnabled 屬性 298
 - imqConsumerFlowLimit 屬性 298
 - imqConsumerFlowThreshold 屬性 298
 - imqdbmgr 指令
 - 在密碼檔案中 149
 - 參照 261
 - 選項 261
 - 關於 32
 - imqDefaultPassword 屬性 297
 - imqDefaultUsername 屬性 297
 - imqDestinationDescription 屬性 300
 - imqDestinationName 屬性 300
 - imqDisableSetClientID 屬性 297
 - imqFlowControlLimit attribute 298
 - imqJMSDeliveryMode 屬性 300
 - imqJMSExpiration 屬性 300
 - imqJMSPriority 屬性 162, 300
 - imqkeytool 指令
 - 使用 142
 - 指令語法 142, 336
 - 參照 264
 - 關於 33
 - imqLoadMaxToServerSession 屬性 161, 299
 - imqobjmgr 指令
 - 子指令 259
 - 參照 259
 - 選項 260
 - 關於 32
 - imqOverrideJMSExpiration 屬性 300
 - imqOverrideJMSHeadersToTemporaryDestinations 屬性 162, 300
 - imqOverrideJMSPriority 屬性 162, 300
 - imqQueueBrowserMax MessagesPerRetrieve 屬性 161, 299
 - imqQueueBrowserRetrieveTimeout 屬性 161, 299
 - imqReconnectAttempts 屬性 295
 - imqReconnectEnabled 屬性 294
 - imqReconnectInterval 屬性 295
 - imqSetJMSXAppID 屬性 299
 - imqSetJMSXConsumerTXID 屬性 299
 - imqSetJMSXProducerTXID 屬性 299
 - imqSetJMSXRcvTimestamp 屬性 299
 - imqSetJMSXUserID 屬性 299
 - imqSSLIsHostTrusted 屬性 295
 - imqsvcadm 指令
 - 子指令 263
 - 參照 263
 - 選項 263
 - 關於 32
 - imqusermgr 指令
 - 子指令 262
 - 用於 127
 - 使用者名稱 130
 - 參照 262
 - 密碼 130
 - 選項 262
 - 關於 32
 - install.properties 檔案 87
- ## J
- J2EE Connector Architecture (JCA) 303, 306
 - Java Message Service 規格 24
 - Java 執行階段
 - 指定路徑 251, 259, 260, 264

L

適用於 Windows 服務 70

Java 虛擬機器, *請參閱*JVM

java.naming.factory.initial 屬性 154, 155

java.naming.provider.url 屬性 154, 156

java.naming.security.authentication 屬性 155

java.naming.security.credentials 屬性 155

java.naming.security.principal 屬性 155

javahome 選項 70

JCA (J2EE Connector Architecture) 303, 306

JDBC 支援

配置 89

關於 80

驅動程式 274

JDBC 型永久性

效能調校 215

設定 90

關於 80

jms 連線服務 74, 101

JMSDeliveryMode 訊息標頭欄位 162

JMSExpiration 訊息標頭欄位 162

JMSPriority 訊息標頭欄位 162

JNDI

位置 (提供者 URL) 154

初始環境 154

物件存放區 32, 154

物件存放區屬性 154, 163

查找 49

查詢名稱 163

jrehome 選項 70

JVM

度量, *請參閱*JVM 度量

效能影響 207

效能調校 211

JVM 度量

使用 imqcmd metrics 187

使用代理程式記錄檔 185

使用訊息型監視 191

度量數目 310

L

LDAP 伺服器

使用者儲存庫存取 133

物件存放區屬性 154

做為使用者儲存庫 132

認證容錯移轉 133

M

ManagedConnectionFactory JavaBean 305

MDB, *請參閱*訊息驅動 Bean

messageSelector 啟動規格屬性 307

N

NORMAL 服務類型 74

nsswitch.conf 檔案 (Linux) 174

O

Oracle 90, 92

P

password 資源介面屬性 304

password 管理式連線工廠屬性 305

PointBase 90

properties

實體目標, *請參閱*實體目標, 特性

R

reconnectAttempts 資源介面屬性 304
 reconnectAttempts 管理式連線工廠屬性 305, 306
 reconnectEnabled 資源介面屬性 304, 305
 reconnectEnabled 管理式連線工廠屬性 306
 reconnectInterval 資源介面屬性 304, 305
 reconnectInterval 管理式連線工廠屬性 306
 reset messages 選項 119
 ResourceAdapter JavaBean 304
 RESTART 特性 68

S

sendUndeliverableMsgsToDMQ 啓動規格屬性 307
 SNTP 66
 SSL
 加密, 與 140
 啓用 144
 連線服務, 請參閱SSL 型連線服務
 透過 TCP/IP 141
 關於 83
 SSL 型連線服務
 啓動 144
 設定 140, 141
 ssladmin 連線服務
 設定 141
 關於 74, 102
 ssljms 連線服務
 設定 141
 關於 74, 101
 subscriptionDurability 啓動規格屬性 306, 307
 subscriptionName 啓動規格屬性 307
 Sun Cluster
 配置 273
 syslog (系統記錄) 85, 184

T

TCP 74, 101
 TLS 74, 101

U

ulimit 指令 66
 userName 資源介面屬性 304
 userName 管理式連線工廠屬性 306

W

W32Time 服務 66
 Windows 服務, 請參閱服務 (Windows)

X

xntpd 常駐程式 66

三書

工具, 管理, 請參閱管理工具

ㄗ書

分散式作業事件
 XA 資源管理員 108

目錄

主代理程式

- 指定 172, 173
- 配置變更記錄 176
- 無法使用 177

主題

- 自動建立的 269, 285
- 新增管理物件 164

代理程式

- HTTP 支援 329
- httpjms 連線服務特性 334
- HTTPS 支援 336
- httpsjms 連線服務特性 342
- 互連, 請參閱代理程式叢集
- 列出連線服務 102
- 存取控制, 請參閱授權
- 自動建立實體目標特性 269
- 自動重新啟動 68
- 更新特性 98
- 使用 SSL 啟動 144
- 度量, 請參閱代理程式度量
- 查詢 97
- 重新啟動 78, 100, 254
- 限制運作方式 77, 210
- 時鐘同步化 66
- 特性 (參照) 265, 289
- 記憶體管理 76, 114, 210
- 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 77, 98, 267, 287
- 配置檔案, 請參閱配置檔案
- 停用訊息佇列 122
- 從故障回復 78
- 啟動時需要的權限 67
- 移除 71
- 連線 173
- 當作 Windows 服務來執行 68
- 實例名稱 249
- 實例配置特性 87
- 管理 93
- 暫停 99, 254

叢集, 請參閱代理程式叢集

- 關閉 100
- 繼續 99, 254
- 顯示特性 97

代理程式 Windows 服務的啟動參數 70

代理程式元件

- 永久性服務 74, 78
- 安全性服務 74, 81
- 連線服務 74
- 路由服務 74, 76
- 監視服務 74, 84

代理程式回應

- 用戶端的等待期間 297

代理程式度量

- 使用 imqcmd 100, 188, 190
- 使用代理程式記錄檔 185
- 使用訊息型監視 191
- 度量訊息 85
- 度量數目 (表) 310
- 記錄程式特性 85, 184, 282
- 報告間隔, 記錄程式 252

代理程式故障和安全的連線 346

代理程式監視服務

- 特性 280

代理程式叢集

- 安全代理程式互通連線 174
- 使用的原因 209
- 架構 209
- 效能影響 209
- 配置特性 172, 283
- 配置檔案 172, 173, 174, 284
- 配置變更記錄 176
- 連線代理程式 173
- 新增代理程式 174
- 實體目標的複製 114
- 暫停實體目標 118

加密

- SSL 型服務, 與 140
- 金鑰工具與 84
- 關於 81, 83

可靠的傳送

- 效能平衡 200

- 可靠傳送 160
- 永久性
 - JDBC 型, *請參閱* JDBC 型永久性
 - JDBC, *請參閱* JDBC 永久性
 - 安全性 91
 - 特性 272
 - 選項 (圖) 78
 - 檔案型 78
 - 關於 78
- 永久性服務, 代理程式 74, 78
- 生產環境
 - 設定 30
 - 管理工作 30
 - 維護 31
- 用戶端
 - 時鐘同步化 66
 - 啟動 72
- 用戶端執行階段
 - 訊息流量調校 216
 - 配置 210
- 用戶端應用程式
 - 影響效能的因素 200
 - 範例 25, 320, 321, 322
- 用戶端識別碼 (ClientID) 159
 - 銷毀長期訂閱 108
- 用法說明 259
- 目標度量
 - 使用 `imqcmd metrics` 187, 189
 - 使用 `imqcmd query` 190
 - 使用訊息型監視 191
 - 度量數目 314
- 目標管理物件
 - 屬性 162
- 目錄變數
 - IMQ_HOME 23
 - IMQ_JAVAHOME 23
 - IMQ_VARHOME 23
- 目錄變數 IMQ_HOME 23
- 目錄變數 IMQ_JAVAHOME 23
- 目錄變數 IMQ_VARHOME 23

六

- 同步化
 - 時鐘 66
 - 磁碟記憶體 89
- 存在時間, *請參閱* 訊息過期
- 存取控制檔
 - 位置 320, 321, 322
- 存取控制檔案
 - 用於 134
 - 存取規則 136
 - 版本 135
 - 格式 135
- 存取規則 136
- 安全性
 - 加密, *請參閱* 加密
 - 物件存放區, 用於 154
 - 授權, *請參閱* 授權
 - 管理員, *請參閱* 安全性管理員
 - 認證, *請參閱* 認證
- 安全性服務, 代理程式 74, 81
- 安全性管理員
 - 關於 81
- 安全性管理程式
 - 特性 276
- 安全套接層標準, *請參閱* SSL
- 成對金鑰
 - 重新產生 143
 - 產生 143
- 自身簽名的憑證 142, 336
- 自動建立實體目標
 - 存取控制 83, 140
 - 特性 (表) 269
- 自動重新連線
 - 限制 158
 - 屬性 158

六

佇列

- 自動建立的 269, 285
- 新增管理物件 165
- 佇列負載平衡發送
 - 特性 115
- 佇列負載平衡傳送
 - 特性 270, 290
- 伺服器故障和安全的連線 346
- 作業系統
 - 效能影響 207
 - 調校 Solaris 效能 211
- 作業事件
 - 回復 258
 - 回轉 108
 - 效能影響 202
 - 資訊 258
 - 管理 108
 - 確定 110, 258
- 刪除
 - 代理程式實例 71
- 刪除目標 119
- 更新
 - 代理程式 98
 - 連線服務 103, 104, 106, 255
- 更新 dst 子指令
 - 限制 117
- 系統時鐘同步化 66
- 防火牆 328

八畫

- 使用者名稱 297
 - 格式 130
 - 預設 127
- 使用者群組 129
 - 刪除指定 129
 - 預先定義的 129
 - 預設 83
- 使用者儲存庫
 - LDAP 132
 - LDAP 伺服器 133

- 平台相依性 128, 262
- 平面檔案 126
- 位置 320, 321, 322
- 使用者狀態 130
- 使用者群組 130
- 初始項目 127
 - 管理 130
 - 寫入 130
 - 關於 81
- 使用率 121
- 協定, *請參閱* 傳輸協定
- 協定類型
 - HTTP 74, 102
 - TCP 74, 101
 - TLS 74, 101
- 永久
- 所有指令的語法 248
- 服務 (Windows)
 - Java 執行階段適用於 70
 - 代理程式執行為 68
 - 重新配置 69
 - 啟動參數適用於 70
 - 移除代理程式 71
 - 疑難排解啟動問題 70
- 服務類型
 - ADMIN 74
 - NORMAL 74
- 版本 259
- 物件存放區 154
 - LDAP 伺服器 154
 - LDAP 伺服器屬性 154
 - 位置 320, 321, 322
 - 檔案系統 155
 - 檔案系統存放區屬性 155
- 物件存放區位置 154
- 金鑰存放區
 - file 143
 - 檔案 337
- 長期訂閱: 列出 107, 258
- 長期訂閱: 效能影響 203
- 長期訂閱: 清除訊息 258

長期訂閱：管理 107
 長期訂閱：銷毀 108, 258

1 畫

度量

主題目標 86, 191
 訊息，請參閱度量訊息
 資料，請參閱度量資料
 關於 85

度量訊息

關於 191
 類型 86, 191

度量資料

代理程式，請參閱代理程式度量
 使用 imqcmd metrics 188
 使用代理程式記錄檔 184
 使用訊息型監視的 API 191
 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 180
 訊息佇列公用程式 (imqcmd) 186
 訊息佇列記錄檔 184
 訊息型監視的 API 191

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 248
 說明 259
 關於 32
 顯示版本 259

指令行語法 248

指令選項

配置覆寫 72

指令檔 167

指導 35

查詢

代理程式 97

連線服務 103, 106

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 216

重新啟動代理程式 100, 254

重新連線，自動，請參閱自動重新連線

限制行爲

實體目標 114

限制運作方式

代理程式 77

實體目標 290

十畫

效能

可靠性平衡 200

指標 196

效能評定 197

基本式樣 198

瓶頸 200

最佳化，請參閱效能調校

測量 196

疑難排解 219

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 196

效能因素

JVM 207

代理程式限制運作方式 210

作業系統 207

作業事件 202

長期訂閱 203

訊息大小 205

訊息內文類型 205

- 訊息伺服器架構 209
- 訊息流量控制 216
- 連線 207
- 硬體 206
- 傳送模式 201
- 傳輸協定 207
- 資料存放區 210
- 確認模式 203
- 選擇器 204
- 檔案同步化 273, 286

效能評定, 效能 197

效能監視

- 工具, 請參閱度量監視工具 179
- 度量資料, 請參閱度量資料

效能調校

- 代理程式調校 215
- 用戶端執行階段 216
- 系統調校 211
- 程序簡介 196

時間同步化服務 66

時鐘同步化 66

特性

- httpjms 連線服務 334
- httpsjms 連線服務 342
- 代理程式實例配置 87
- 代理程式監視服務 280
- 永久性 272
- 安全性 276
- 自動建立 269
- 記憶體管理 114
- 記錄程式 280
- 連線服務 266
- 路由服務 267
- 與 JDBC 相關 87, 273
- 語法 88
- 叢集配置 283

記憶體管理

- 用於代理程式 76
- 使用實體目標特性 114
- 效能調校 215

記錄, 請參閱記錄程式

記錄程式

- 自動重建條件 184
- 度量資訊 282
- 重新導向記錄訊息 184
- 訊息格式 182
- 設定特性 182
- 種類 181
- 寫入至主控台 85, 252, 280, 285
- 層級 85, 181, 251, 280, 286
- 輸出通道 85, 181, 183
- 關於 85
- 變更配置 183

記錄檔

- 名稱 181
- 自動重建條件 85, 184, 281, 285
- 自動重建頻率 181
- 記錄停用的訊息 185
- 設定特性 182
- 報告度量 184
- 預設位置 320, 321, 322
- 變更預設名稱 181
- 變更預設位置 181

訊息

- 大小, 與效能 205
- 內文類型與效能 205
- 分段程序 79
- 代理程式限制 77, 98, 267, 287
- 可靠傳送 160
- 永久性 78
- 目標限制 269, 289
- 收回已過期的 77, 268, 286
- 延時 196
- 度量訊息, 請參閱度量訊息
- 流量效能 196
- 流量控制, 請參閱訊息流量控制
- 從實體目標清除 119, 257
- 實體目標限制 114
- 暫停流量 118

訊息分段程序 79

訊息伺服器架構 209

訊息服務效能 206

訊息流量控制

- 代理程式 76, 114
- 計數 216
- 限制 217
- 效能影響 216
- 效能調校 216
- 屬性 160
- 訊息過期
 - 時鐘同步化與 66
- 訊息標頭覆寫 162
- 訊息驅動 Bean
 - 資源介面配置 303, 306
- 迴路位址 174
- 配置檔案 86
 - 代理程式 (圖) 87
 - 安裝 87
 - 位置 320, 321, 322
 - 預設 87
 - 實例 87, 172, 320, 321, 322
 - 範本 320, 321, 322
 - 範本位置 320, 321, 322
 - 編輯 87
 - 叢集 172, 173, 174, 284
- 配置變更記錄 176
 - 備份 177
 - 復原 177

十一 畫

- 停用的訊息
 - 另請參閱*停用的訊息佇列記錄 85
- 停用的訊息佇列
 - 記錄 85
- 停用訊息佇列
 - maxNumMsgs 值 123
 - maxTotalMsgBytes 值 123
 - 限制運作方式 123
 - 記錄 123
 - 配置 122
- 執行緒池管理

- 共用執行緒 76
- 專屬執行緒 76
- 關於 75
- 密碼
 - JDBC 150
 - LDAP 150
 - SSL 金鑰存放區 143, 150, 250
 - 命名慣例 130
 - 密碼檔案, *請參閱*密碼檔案
 - 預設 297
 - 管理員 131
 - 編碼 276, 277
- 密碼檔案
 - 代理程式配置特性 83, 277
 - 位置 150, 320, 321, 322
 - 使用 149
 - 指令行選項 250
- 授權
 - 另請參閱*存取控制
 - 使用者群組 83
 - 啟動選項 251
 - 管理 134
 - 關於 83
- 啓動
 - SSL 型連線服務 144
 - 用戶端 72
- 清除, 實體目標的訊息 119
- 瓶頸, 效能 200
- 產生者
 - 目標限制 270, 290
 - 實體目標限制 114
- 移除
 - 代理程式 71
 - 實體目標 119
- 通道 Servlet 連線 346
- 連接服務度量
 - 使用 imqcmd metrics 189
 - 使用 imqcmd query 190
- 連接埠對映器
 - 連接埠指定 249
 - 關於 75
- 連線

- 由檔案描述元限制 66
- 列出 106, 256
- 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 346
- 查詢 106, 256
- 容錯移轉, 請參閱自動重新連線
- 效能影響 207
- 連線工廠的管理物件
 - 佇列瀏覽器運作方式屬性 299
 - 標準訊息特性 299
- 連線工廠管理物件
 - JMS 特性支援屬性 161
 - 可靠性與流量控制屬性 160
 - 用戶端標識屬性 159
 - 佇列瀏覽器運作方式屬性 161
 - 連線處理屬性 157
 - 應用程式伺服器支援屬性 299
 - 覆寫訊息標頭欄位 162
 - 屬性 156
- 連線代理程式 173
- 連線服務
 - admin 74, 102
 - HTTP, 請參閱HTTP 連線
 - httpjms 74, 102
 - HTTPS, 請參閱HTTPS 連線
 - httpsjms 74, 102
 - jms 74, 101
 - SSL 型 143
 - ssladmin, 請參閱ssladmin 連線服務
 - ssljms, 請參閱ssljms 連線服務
 - 在啟動時啟動 266
 - 存取控制 81, 276
 - 更新 103, 104, 106, 255
 - 協定類型 74
 - 服務類型 74
 - 度量資料, 請參閱連線服務度量
 - 指令影響 255
 - 查詢 103, 106
 - 特性 103, 266
 - 執行緒分配 103
 - 執行緒池管理 75
 - 連接埠對映器, 請參閱連接埠對映器

- 暫停 105, 255
- 叢集 141, 174
- 繼續 105, 255
- 顯示特性 103
- 連線服務, 代理程式 74
- 連線服務度量
 - 使用 imqcmd metrics 104
 - 度量數目 312

十二畫

- 硬體, 效能影響 206
- 開發環境管理工作 30

十三畫

- 傳送模式
 - 效能影響 201
- 傳輸協定
 - 協定類型, 請參閱協定類型
 - 相關速度 208
 - 效能影響 207
 - 效能調校 212
- 資料存放區
 - 內容 89
 - 平面檔案 78
 - 位置 320, 321, 322
 - 重設 250
 - 效能影響 210
 - 配置 89
 - 磁碟同步化 89
 - 與 JDBC 相容 80
 - 壓縮 79
 - 關於 78
- 資源介面 303
 - 重新連線 304, 305, 306
- 路由服務
 - 特性 267

路由服務,代理程式 74,76

十 二 畫

實例目錄

和檔案型資料存放區 89

移除 71

與實例配置檔案 132

實例配置檔案,請參閱配置檔案

實體目標

用於發送的批次訊息 114

用於傳送的批次訊息 270,271,290

列出 115

收回磁碟空間 121

自動建立的 140

更新特性 117

更新屬性 257

使用停用訊息佇列 122

取得資訊 116,257

度量,請參閱實體目標度量

建立 114

限制行爲 114

限制運作方式 290

特性 289

特性值 116

停用訊息佇列 122

清除訊息 119,257

資訊 116

磁碟使用 120

管理 111

暫存 115

暫停 118,256

銷毀 119

壓縮 120

壓縮基於檔案的資料倉庫 121

壓縮檔案型資料存放區 257

叢集中限制的範圍 114,271,290

類型 115,256

繼續 118

顯示特性值 116

實體目標屬性 289

疑難排解 219

Windows 服務啓動 70

監視,請參閱效能監視

監視服務,代理程式 74,84

磁碟空間

收回 121

實體目標的使用 120

管理工作

生產環境 30

開發環境 30

管理工具 32

命令行公用程式 32

管理主控台 33

管理主控台

指導 35

啓動 37

管理物件

XA 連線工廠,請參閱連線工廠管理物件

主題,請參閱主題

列出 166

佇列,請參閱佇列

刪除 165

更新 167

物件存放區,請參閱物件存放區

查詢 166

管理 153

需要的資訊 163

屬性(參照) 293

管理員密碼 131

認證

另請參閱存取控制

管理 126

關於 82

說明(指令行) 259

十 三 畫

寫入作業(針對檔案型存放區) 89

十六畫

暫存實體目標 115
暫停
 代理程式 99, 254
 連線服務 105, 255
 實體目標 118, 256
稽核記錄 151
範例應用程式 25, 320, 321, 322
銷毀實體目標 119

十六畫

憑證 142, 336
選擇器
 效能影響 204
 關於 204

十六畫

優先順序 (配置特性的) 87
壓縮
 實體目標 120
 檔案型資料存放區 79
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 273, 286
 與 Sun Cluster 273, 286
檔案型永久性 78
檔案描述元限制 66
 連線限制和 66
環境變數, 請參閱目錄變數
金鑰工具 84

十八畫

叢集, 請參閱代理程式叢集

叢集的目錄查找 (Linux) 174
叢集配置特性 172, 283
叢集配置檔案 172, 173, 174, 284
叢集連線服務 141, 174
 主機名稱或 IP 位址 172, 283
 連接埠號用於 172, 283
 網路傳輸用於 172, 173, 283
簡易網路管理協定 66
覆寫
 在指令行中 72
 為訊息標頭 162

十七畫

關閉代理程式 100, 254
 當作 Windows 服務 71

二十畫

繼續
 代理程式 99, 254
 連線服務 105, 255
 實體目標 118

二十二畫

權限
 內嵌式資料庫 91
 存取控制特性檔案 83, 135
 使用者儲存庫 128, 262
 金鑰存放區 337
 計算 136
 密碼檔案 150
 資料存放區 79
 管理服務 83

二十三卷

顯示產品版本 259

