



Sun Java™ System

Message Queue 3.5

管理指南

Service Pack 1

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

文件號碼：817-7213

Copyright © 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. 版權所有。

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

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

Sun、Sun Microsystem、Sun 標誌、Java、Solaris、Sun[tm] ONE、JDK、Java 命名和目錄介面、Javadoc、JavaMail、JavaHelp、Java Coffee Cup 標誌以及 Sun[tm] ONE 標誌為 Sun Microsystems, Inc. 在美國和其他國家 / 地區的商標或註冊商標。

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

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

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

目錄

圖	15
表	17
程序	21
前言	23
本指南的使用者	23
本指南的組織架構	23
慣例	25
字慣例	25
目錄變數慣例	25
其他說明文件資源	28
Message Queue 說明文件集	28
線上說明	29
JavaDoc	29
範例非戶端應用程式	29
Java 訊息服務 (JMS) 規格	29
相關的第三方網站參考	29
第 1 章 簡介	31
何謂 Sun Java System Message Queue ?	31
產品版本	33
平台版	33
企業版	33
企業訊息傳送系統	34
企業訊息傳送系統的要求	34
集中式訊息傳送和點對點訊息傳送	34
訊息傳送系統概念	35
訊息	35
訊息服務架構	36

訊息發送模型	36
JMS 規格	37
JMS 訊息結構	37
JMS 程式設計模型	37
JMS 受管理物件	39
JMS/J2EE 程式設計：訊息驅動 Bean	39
訊息驅動 Bean	40
J2EE 應付程式伺服器支援	41
JMS 訊息傳送問題	41
JMS 提供者獨立性	41
程式設計領域	42
用戶端識別碼	43
可靠的訊息傳送	43
確認 / 異動	44
永久性儲存	45
效能平衡	45
訊息選取	46
訊息順序和優先順序	46
第 2 章 Message Queue 訊息傳送系統	47
Message Queue 訊息伺服器	48
代理程式	48
連接服務	50
訊息路由器	53
持續性管理程式	57
多全性管理程式	60
監視服務	63
實體目標	68
佇列目標	69
主題目標	70
自動建立的 (對應管理員建立的) 目標	70
暫存目標	72
多重代理程式叢集 (企業版)	72
多重代理程式架構	73
使非開發環境中的叢集	76
叢集配置特性	76
Message Queue 用戶端運行時間	77
訊息生產	77
訊息使用	78
Message Queue 受管理物件	79
連線工廠受管理物件	80
目標受管理物件	80
啟動用戶端時置換屬性值	81

第 3 章 Message Queue 管理工具	83
Message Queue 管理工作	83
開發環境	83
生產環境	84
設定作業	84
若要設定生產環境	84
維護作業	85
若要設定生產環境	85
Message Queue 管理工具	87
管理主控台	87
指令行公用程式摘要	88
指令行語法	89
公用程式指令選項	90
第 4 章 管理主控台指導	91
準備工作	92
啟動管理主控台	92
若要啟動管理主控台	92
取得說明	93
若要顯示管理主控台說明資訊	93
使用代理程式	95
啟動代理程式	95
若要啟動代理程式	95
新增代理程式	96
若要將代理程式新增至管理主控台	96
變更管理員密碼	98
若要變更管理員密碼	98
連接到代理程式	98
若要連接至代理程式	98
檢視連接服務	99
若要檢視可用的連接服務	99
將實體目標新增至代理程式	100
若要將佇列目標新增至代理程式	101
使用實體目標	101
若要檢視實體目標的特性	102
若要從目標中清除訊息	103
若要刪除目標	103
取得有關目標的資訊	103
使用物件倉庫	104
新增物件倉庫	104
若要新增檔案系統物件倉庫	104
檢查物件倉庫特性	106
若要顯示物件倉庫的特性	106

連接至物件倉庫	107
若要連接至物件倉庫	107
新增連線工廠受管理物件	107
若要將連線工廠新增至物件倉庫	107
新增目標受管理物件	109
若要將目標新增至物件倉庫	109
受管理物件特性	110
若要檢視或更新目標物件的特性	110
更新主控台資訊	111
執行範例應酬程式	111
若要執行 HelloWorldMessageJNDI 應用程式	111
第 5 章 啟動與配置代理程式	115
配置檔錄	115
實例配置檔錄	116
合併特性值	116
特性命名語法	117
編輯實例配置檔錄	117
啟動代理程式	122
imqbrokerd 指令的語法	122
啟動範例	123
若要啟動預設代理程式名稱和配置的代理程式實例	123
若要啟動使用試用企業版授權的代理程式實例	124
若要啟動包含外掛永久性的已命名代理程式實例	124
imqbrokerd 選項的摘要	124
使用叢集 (企業版)	127
叢集配置特性	127
連接代理程式	129
連接方法	129
將代理程式連接到叢集	129
完全代理程式互連連接	130
若要在叢集內配置安全連接	130
管理叢集中的代理程式	130
將代理程式新增至叢集	130
若要新增新的代理程式至現有叢集	130
重新啟動叢集中的代理程式	131
若要重新啟動已為現有叢集一員的代理程式	131
從叢集中移除代理程式	131
若要從現有叢集移除代理程式	131
管理主代理程式的配置變更記錄	132
備份配置變更記錄	132
若要備份配置變更記錄	132
復原配置變更記錄	132

在發生故障的情況下復原主代理程式	132
記錄	133
預設記錄配置	133
日誌訊息格式	133
變更記錄程式配置	134
變更代理程式的記錄程式配置	134
變更輸出通道	135
變更記錄檔自動重建條件	135
第 6 章 代理程式和目標程式管理	137
指令公用程式	137
imqcmd 指令的語法	138
imqcmd 子指令	138
imqcmd 選項摘要	139
使用 imqcmd 指令	141
範例 imqcmd 的用法	142
管理代理程式	142
顯示代理程式資訊	144
更新代理程式特性	144
控制代理程式的狀態	145
暫停與繼續代理程式	145
關閉和重新啟動代理程式	146
顯示代理程式度量	146
管理連接服務	147
列出連接服務	148
顯示連接服務資訊	149
更新連接服務特性	149
顯示連接服務度量	150
暫停與繼續連接服務	150
取得連接資訊	151
管理目標	152
建立目標	153
列出目標	156
顯示目標資訊	156
更新目標屬性	157
顯示目標度量	158
暫停和繼續目標	158
清除目標	159
銷毀目標	159
壓縮目標	159
監視目標的磁碟使用	160
回收未使用的目標磁碟空間	160
若要回收未使用的目標磁碟空間	161

管理長期訂閱	161
管理異動	162

第 7 章 管理受管理物件 165

關於物件倉庫	165
LDAP 伺服器物件倉庫	166
檔案系統物件倉庫	167
受管理物件	168
連線工廠受管理物件屬性	168
目標受管理物件屬性	170
物件管理員公用程式 (imqobjmgr)	170
imqobjmgr 指令的語法	171
imqobjmgr 子指令	171
imqobjmgr 指令選項摘要	171
需要的資訊	173
使用指令檔	173
新增和刪除受管理物件	176
新增連線工廠	176
新增主題佇列	177
刪除受管理物件	178
取得資訊	179
列出受管理物件	179
有關單一物件的資訊	180
更新受管理物件	180

第 8 章 管理安全性 183

認證使用者	184
使用子字檔或使用者儲存庫	184
建立使用者儲存庫	184
使用者管理員公用程式 (imqusermgr)	185
群組	187
狀態	187
使用者名稱與密碼的格式	188
登入和管理使用者儲存庫	188
變更預設管理員密碼	189
將 LDAP 伺服器註冊於使用者儲存庫	190
若要編輯配置檔案以使用 LDAP 伺服器	190
授權使用者：	
存取控制特性檔案	192
建立存取控制特性檔案	193
存取規則語法	193
許可權運算	194

連接存取控制	196
目標存取控制	196
目標自動建立存取控制	197
10.6 使用基於 SSL 的服務 (企業版)	198
透過 TCP/IP 設定基於 SSL 的服務	198
若要設定基於 SSL 的連接服務	199
步驟 1. 產生自身簽名的憑證	199
若要重新產生鍵對	200
步驟 2. 啟用代理程式中基於 SSL 的服務	200
若要啟用代理程式中基於 SSL 的服務	201
步驟 3. 啟動代理程式	201
步驟 4. 配置並執行基於 SSL 的用戶端	202
透過 HTTP 設定基於 SSL 的服務	203
使用密碼檔	204
第 9 章 分析與監控訊息佇列	205
關於效能	205
效能調整程序	205
效能類型	206
效能設定	206
基本使用式樣	207
影響效能的因素	208
影響效能的應用程式設計因素	209
發送模式 (永久性 / 非永久性訊息)	210
異動使用	211
確認模式	212
長期與非長期訂閱	212
選擇器的使用 (訊息過濾)	213
訊息容量	214
訊息佇列類型	214
影響效能的訊息服務因素	215
硬體	215
作業系統	216
Java 虛擬機器 (JVM)	216
連接	216
訊息伺服器架構	218
代理程式限制和行爲	219
資料庫效能	219
用戶端運行時間配置	219
監視訊息伺服器	220
監視工具	220
Message Queue 指令行公用程式 (mqcmd)	220
若要使用 metrics 子指令	222

Message Queue 代理程式日誌檔	225
若要用日誌檔報告度量資訊	225
基於訊息監視的 API	226
若要設定基於訊息的監視	227
選擇正確的監視工具	228
度量資料的說明	229
JVM 度量	230
代理程式策略度量	230
連接服務度量	232
目標度量	234
排解效能問題	236
問題：客戶端無法建立連接	237
徵兆：	237
可能原因：	237
問題：連接流量過慢	240
徵兆：	240
可能原因：	240
問題：客戶端無法建立訊息產生者	242
徵兆：	242
可能原因：	242
問題：訊息產生延遲或過慢	242
徵兆：	242
可能原因：	243
問題：訊息儲存在訊息伺服器中	245
徵兆：	245
可能原因：	245
問題：訊息伺服器的流量不穩定	249
徵兆：	249
可能原因：	249
問題：訊息未送達使用者	250
徵兆：	250
可能原因：	250
調整您的配置以改善效能	251
系統調整	251
Solaris 調整：CPU 使用率、分頁 / 交換 / 磁碟 I/O	251
Java 虛擬機器調整	252
調整傳輸協定	252
調整基於檔案的永久性倉庫	255
代理程式調整	255
記憶體管理：增加代理程式負載的穩定性	255
多重使用者佇列效能	256
客戶端運行時間訊息流量調整	257
訊息流量計數	257

訊息流量限制	257
附錄 A Message Queue 資料的位置	259
Solaris	259
Linux	260
Windows	261
附錄 B 設定外掛持續性	263
介紹	263
插入 JDBC 可存取資料倉庫	264
插入 JDBC 可存取的資料倉庫	264
與 JDBC 相關的代理程式配置特性	265
資料庫管理員公用程式 (imqdbmgr)	269
imqdbmgr 指令的語法	269
imqdbmgr 子指令	269
imqdbmgr 指令選項摘要	271
附錄 C HTTP/HTTPS 支援 (企業版)	273
HTTP/HTTPS 支援架構	273
啟用 HTTP 支援	274
啟用 HTTP 支援	275
步驟 1. 在 Web Server 上部署 HTTP 通道 Servlet	275
部署為 JAR 檔案	275
部署為 Web 歸檔檔案	276
步驟 2. 配置 httpjms 連接服務	276
啟動 httpjms 連接服務	276
步驟 3. 配置 HTTP 連接	277
配置連線工廠	277
使用單一 Servlet 存取多重代理程式	278
使用 HTTP 代理	278
範例 1：在 Sun Java System Web Server 上部署 HTTP 通道 Servlet	279
部署為 JAR 檔案	279
新增通道 Servlet	279
若要配置通道 Servlet 的虛擬路徑 (Servlet URL)	280
在啟動 Web Server 時載入通道 Servlet	280
停用伺服器存取記錄	281
部署為 WAR 檔案	281
將 HTTP 通道 Servlet 部署為 WAR 檔案	281
範例 2：在 Sun Java System Application Server 7.0 上部署 HTTP 通道 Servlet	282
使用部署工具	282
在 Application Server 7.0 環境中部署 HTTP 通道 Servlet	282
修改 server.policy 檔案	283

若要修改 Application Server 的伺服器策略檔案	283
啓用 HTTPS 支援	283
啓用 HTTPS 支援	284
步驟 1. 爲 HTTPS 通道 Servlet 產生自身簽名的憑證	284
步驟 2. 在 Web Server 上部署 HTTPS 通道 Servlet	285
部署爲 JAR 檔案	285
部署爲 Web 歸檔檔案	286
步驟 3. 配置 httpsjms 連接服務	286
啓動 httpsjms 連接服務	286
步驟 4. 配置 HTTPS 連接	287
配置 JSSE	288
配置 JSSE	288
匯入超級使用者憑證	288
配置連線工廠	288
使單一 Servlet 存取多重代理程式	289
使 HTTP 代理	290
範例 3 : 在 Sun Java System Web Server 上部署 HTTPS 通道 Servlet	290
部署爲 JAR 檔案	290
新增通道 Servlet	290
若要配置通道 Servlet 的虛擬路徑 (Servlet URL)	291
在啓動 Web Server 時載入通道 Servlet	292
停用伺服器存取記錄	292
部署爲 WAR 檔案	292
修改 HTTPS 通道 Servlet WAR 檔案	292
將 HTTPS 通道 Servlet 部署爲 WAR 檔案	293
範例 4 : 在 Sun Java System Application Server 7.0 上部署 HTTPS 通道 Servlet	294
使部署工具	294
在 Application Server 7.0 環境中部署 HTTPS 通道 Servlet	294
修改 server.policy 檔案	295
若要修改 Application Server 的伺服器策略檔案	295
附錄 D 將代理程式用作 Windows 服務	297
將代理程式作爲 Windows 服務執行	297
服務管理員公用程式 (imqsvcadmin)	297
imqsvcadmin 指令的語法	298
imqsvcadmin 子指令	298
imqsvcadmin 選項摘要	298
移除代理程式服務	299
重新配置代理程式服務	299
使替代 Java 運行時間	300
查詢代理程式服務	300
疑難排解	300
查閱記錄的服務錯誤事件	300

附錄 E 技術說明	301
系統時鐘設定	301
建議巨步化	301
避免將系統時鐘配置為早於當前時間	302
定義作業系統檔案描述元限制	302
保護永久性資料	302
創建永久性倉庫	303
外掛永久性倉庫	304
附錄 F Message Queue 資源介面	305
附錄 G 選擇性 JMS 功能的 Message Queue 實施	307
附錄 H Message Queue 介面的穩定性	309
書表	313
索引	317



圖 1-1	集中式訊息傳送和點對點訊息傳送	35
圖 1-2	訊息服務架構	36
圖 1-3	JMS 程式設計物件	38
圖 1-4	與 MDB 進行訊息傳送	40
圖 2-1	Message Queue 系統架構	47
圖 2-2	代理程式服務元件	49
圖 2-3	連接服務支援	51
圖 2-4	持續性管理程式支援	58
圖 2-5	安全性管理程式支援	61
圖 2-6	監視服務支援	64
圖 2-7	多重代理程式 (叢集) 架構	74
圖 2-8	訊息傳送作業	77
圖 2-9	訊息發送至 Message Queue 用戶端運行時間	78
圖 3-1	本機與遠端管理公用程式	88
圖 5-1	代理程式配置檔案	117
圖 9-1	透過 Message Queue 服務的訊息發送	208
圖 9-2	發送模式的效能影響	211
圖 9-3	訂閱類型的效能影響	213
圖 9-4	訊息容量的效能影響	214
圖 9-5	傳輸協定速度	217
圖 9-6	傳輸協定的效能影響	218
圖 9-7	在 1K (1,024 個位元組) 封包上變更 <code>inbufsz</code> 的效果	254
圖 9-8	在 1K (1,024 個位元組) 封包上變更 <code>outbufsz</code> 的效果	254
圖 C-1	HTTP/HTTPS 支援架構	274

表 1	本書內容	24
表 2	文件慣例	25
表 3	Message Queue 目錄變數	26
表 4	Message Queue 說明文件集	28
表 1-1	JMS 程式設計物件	42
表 2-1	主代理程式服務元件和功能	49
表 2-2	代理程式支援的連接服務	50
表 2-3	連接服務特性	52
表 2-4	訊息路由器特性	56
表 2-5	持續性管理程式特性	59
表 2-6	安全性管理程式特性	62
表 2-7	記錄種類	64
表 2-8	度量主題目標	65
表 2-9	監視服務特性	66
表 2-10	自動建立配置特性	71
表 2-11	目標屬性	81
表 3-1	共用 Message Queue 指令行選項	90
表 5-1	代理程式實例配置特性	118
表 5-2	imqbrokerd 選項	124
表 5-3	叢集配置特性	127
表 5-4	imqbrokerd 記錄程式選項和對應的特性	134
表 6-1	imqcmd 子指令	138
表 6-2	imqcmd 選項	140
表 6-3	用來管理代理程式的 imqcmd 子指令	143
表 6-4	imqcmd 更新的代理程式特性	144
表 6-5	用於管理連接服務的 imqcmd 子指令	147
表 6-6	代理程式支援的連接服務	148

表 6-7	imqcmd 更新的連接服務特性	149
表 6-8	用於管理連接服務的 imqcmd 子指令	151
表 6-9	用於管理目標的 imqcmd 子指令	152
表 6-10	目標屬性	154
表 6-11	目標磁碟使用度量	160
表 6-12	用於管理長期訂閱的 imqcmd 子指令	161
表 6-13	用於管理異動的 imqcmd 子指令	162
表 7-1	LDAP 物件倉庫屬性	166
表 7-2	檔案系統物件倉庫屬性	167
表 7-3	連線工廠受管理物件屬性	168
表 7-4	目標受管理物件屬性	170
表 7-5	imqobjmgr 子指令	171
表 7-6	imqobjmgr 選項	171
表 7-7	命名慣例範例	177
表 8-1	使用者儲存庫中的初始項目	185
表 8-2	imqusermgr 子指令	186
表 8-3	imqusermgr 選項	186
表 8-4	無效的使用者名稱與密碼字元	188
表 8-5	與 LDAP 相關的特性	191
表 8-6	存取規則的語法元素	194
表 8-7	目標存取控制規則的元素	197
表 8-8	密鑰儲存特性	200
表 8-9	密碼檔案中的密碼	204
表 9-1	高可靠性和高效能方案的比較	210
表 9-2	imqcmd metrics 子指令語法	221
表 9-3	imqcmd metrics 子指令選項	221
表 9-4	imqcmd query 子指令語法	224
表 9-5	度量主題目標	226
表 9-6	度量監視工具的優缺點	229
表 9-7	JVM 度量	230
表 9-8	代理程式範圍度量	231
表 9-9	連接服務度量	232
表 9-10	目標度量	234
表 A-1	Solaris 上 Message Queue 資料的位置	259
表 A-2	Linux 上 Message Queue 資料的位置	260
表 A-3	Windows 上 Message Queue 資料的位置	261
表 B-1	與 JDBC 相關的特性	266

表 B-2	imqdbmgr 子指令	269
表 B-3	imqdbmgr 選項	271
表 C-1	httpjms 連接服務特性	276
表 C-2	用於部署 HTTP 通道 Servlet JAR 檔案的 Servlet 引數	280
表 C-3	httpsjms 連接服務特性	287
表 C-4	用於部署 HTTPS 通道 Servlet JAR 檔案的 Servlet 引數	291
表 D-1	imqsvcadmin 子指令	298
表 D-2	imqsvcadmin 選項	298
表 G-1	選擇性 JMS 功能	307
表 H-1	Message Queue 介面的穩定性	309
表 H-2	介面穩定性分類機制	310

若要配置生產環境	84
若要配置生產環境	85
若要啓動管理主控台	92
若要顯示管理主控台說明資訊	93
若要啓動代理程式	95
若要將代理程式新增至管理主控台	96
若要變更管理員密碼	98
若要連接至代理程式	98
若要檢視可用的連接服務	99
若要將佇列目標新增至代理程式	101
若要檢視實體目標的特性	102
若要從目標中清除訊息	103
若要刪除目標	103
若要新增檔案系統物件倉庫	104
若要顯示物件倉庫的特性	106
若要連接至物件倉庫	107
若要將連線工廠新增至物件倉庫	107
若要將目標新增至物件倉庫	109
若要檢視或更新目標物件的特性	110
若要執行 HelloWorldMessageJNDI 應用程式	111
若要啓動預設代理程式名稱和配置的代理程式實例	123
若要啓動使用試用企業版授權的代理程式實例	124
若要啓動包含外掛永久性的已命名代理程式實例	124
將代理程式連接到叢集	129
若要在叢集內配置安全連接	130
若要新增新的代理程式至現有叢集	130
若要重新啓動已爲現有叢集一員的代理程式	131

若要從現有叢集移除代理程式	131
若要備份配置變更記錄	132
在發生故障的情況下復原主代理程式	132
變更代理程式的記錄程式配置	134
若要回收未使用的目標磁碟空間	161
若要編輯配置檔案以使用 LDAP 伺服器	190
若要配置基於 SSL 的連接服務	199
若要重新產生鍵對	200
若要啓用代理程式中基於 SSL 的服務	201
若要使用 metrics 子指令	222
若要使用日誌檔報告度量資訊	225
若要配置基於訊息的監視	227
插入 JDBC 可存取的資料倉庫	264
啓用 HTTP 支援	275
啓動 httpjms 連接服務	276
新增通道 Servlet	279
若要配置通道 Servlet 的虛擬路徑 (Servlet URL)	280
在啓動 Web Server 時載入通道 Servlet	280
停用伺服器存取記錄	281
將 HTTP 通道 Servlet 部署為 WAR 檔案	281
在 Application Server 7.0 環境中部署 HTTP 通道 Servlet	282
若要修改 Application Server 的伺服器策略檔案	283
啓用 HTTPS 支援	284
啓動 httpsjms 連接服務	286
配置 JSSE	288
新增通道 Servlet	290
若要配置通道 Servlet 的虛擬路徑 (Servlet URL)	291
在啓動 Web Server 時載入通道 Servlet	292
停用伺服器存取記錄	292
修改 HTTPS 通道 Servlet WAR 檔案	292
將 HTTPS 通道 Servlet 部署為 WAR 檔案	293
在 Application Server 7.0 環境中部署 HTTPS 通道 Servlet	294
若要修改 Application Server 的伺服器策略檔案	295
查閱記錄的服務錯誤事件	300

「Sun Java™ System Message Queue 3.5 SP1 管理指南」這本書提供執行 Message Queue 訊息傳送系統管理工作所需的背景和資訊。

本前言包含以下各節：

- 第 23 頁的「本指南的使用者」
- 第 23 頁的「本指南的組織架構」
- 第 25 頁的「慣例」
- 第 28 頁的「其他說明文件資源」

本指南的使用者

本指南可供需要執行 Message Queue 管理工作的管理員和應用程式開發人員使用。

Message Queue 管理員負責配置和管理 Message Queue 訊息傳送系統，尤其是位於本系統中心的 Message Queue 訊息伺服器。本書假設您不瞭解訊息傳送系統知識。

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

本指南的組織架構

本指南需要您完整閱讀。下表簡要描述了每章的內容：

表 1 本書內容

章	說明
第 1 章「簡介」	描述 Message Queue 訊息傳送系統以及術語的高級概念簡介。
第 2 章「Message Queue 訊息傳送系統」	描述 Message Queue 訊息傳送系統，重點介紹可共同提供訊息傳送服務的 Message Queue 代理程式和 Message Queue 用戶端運行時間。
第 3 章「Message Queue 管理工作和工具」	描述 Message Queue 管理工作和工具，並介紹用於管理的命令行公用程式及其共用功能。
第 4 章「管理主控台指導」	提供實用指導，使您熟悉管理主控台 (即 Message Queue 訊息伺服器的圖形介面)。
第 5 章「啓動與配置代理程式」	介紹如何啓動和配置 Message Queue 代理程式及代理程式叢集。
第 6 章「代理程式和應用程式管理」	介紹如何執行與管理 Message Queue 代理程式相關的 (獨立於應用程式) 工作，以及用於管理訊息傳送應用程式的工作。
第 7 章「管理受管理物件」	介紹如何執行與建立和管理 Message Queue 受管理物件相關的工作。
第 8 章「管理安全性」	介紹如何執行與應用程式相關的安全性工作，如管理認證、授權以及加密。
第 9 章「分析與調整訊息服務」	描述監視和分析訊息伺服器效能的技術，並解釋如何將訊息伺服器調整到最佳化效能。
附錄 A「Message Queue 資料的位置」	描述不同種類 Message Queue 資料的位置。
附錄 B「設定外掛持續性」	介紹如何配置 Message Queue 以使用 JDBC 相容資料庫來執行持續性功能。
附錄 C「HTTP/HTTPS 支援 (企業版)」	介紹如何在訊息傳送用戶端與 Message Queue 訊息伺服器之間配置 HTTP 連接服務。
附錄 D「將代理程式用作 Windows 服務」	介紹如何使用 Message Queue 服務管理公用程式 (mqsvcadm) 來安裝、查詢及移除代理程式 (作為 Windows 服務執行)。
附錄 E「技術說明」	在本書的相關主題中提供大量的專用技術說明，但未提供 Message Queue 特定管理的部份。
附錄 F「Message Queue 資源介面」	描述何謂 Message Queue 資源介面，如何部署資源介面，以及如何配置與使用資源介面。
附錄 G「選擇性 JMS 功能的 Message Queue 實施」	描述 Message Queue 產品如何處理 JMS 規格中列出的每個項目，將其作為 JMS 提供者實施的選擇性功能。
附錄 H「Message Queue 介面的穩定性」	描述不同 Message Queue 介面的穩定性。
「詞彙表」	定義 Message Queue 說明文件中使用的專有名詞。

慣例

此節提供了有關本文件中所使用慣例的資訊。

文字慣例

表 2 文件慣例

格式	說明
<i>斜體</i>	斜體文字表示定位字元。請在看到斜體文字時將其取代為適當的子句或值。斜體文字也用於指定需要強調的文件標題，或用於引入的單字或片語。
固定間距	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 (包括簽名中的所有元素)、套裝軟體名稱、保留字以及 URL。
[]	指示指令行語法敘述中選用值的方括號。
全部大寫	所有大寫文字表示檔案系統類型 (GIF、TXT、HTML 等等)、環境變數 (IMQ_HOME) 或縮寫字 (Message Queue、JSP)。
鍵 + 鍵	透過加號連接在一處並同時按下的按鍵：Ctrl+A 表示同時按下這兩個鍵。
鍵 - 鍵	透過連字號連接在一處的連續按下的按鍵：Esc-S 表示按下 Esc 鍵，然後釋放它，接著按下 S 鍵。

目錄變數慣例

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

表 3 Message Queue 目錄變數

變數	說明
IMQ_HOME	<p>它通常在 Message Queue 說明文件中使用，指的是 Message Queue 基底目錄 (根安裝目錄)：</p> <ul style="list-style-type: none"> 在 Solaris 上，沒有 Message Queue 根安裝目錄。因此，未在 Message Queue 說明文件中使用 IMQ_HOME 來指代 Solaris 上的檔案位置。 在 Solaris 上，Sun Java System Application Server 的 Message Queue 根安裝目錄位於 Application Server 基底目錄下的 /imq。 在 Windows 上，Message Queue 根安裝目錄由 Message Queue 安裝程式配置 (依預設，該目錄為 C:\Program Files\Sun\MessageQueue3)。 在 Windows 上，Sun Java System Application Server 的 Message Queue 根安裝目錄位於 Application Server 基底目錄下的 /imq。 在 Linux 上，沒有 Message Queue 根安裝目錄。因此，未在 Message Queue 說明文件中使用 IMQ_HOME 來指定 Linux 上的檔案位置。
IMQ_VARHOME	<p>這是儲存 Message Queue 暫存檔或動態建立的配置檔與資料檔的 /var 目錄。可以將它配置為指向任何目錄的環境變數。</p> <ul style="list-style-type: none"> 在 Solaris 上，IMQ_VARHOME 預設為 /var/imq 目錄。 在 Solaris 上，針對 Sun Java System Application Server, Evaluation Edition，IMQ_VARHOME 預設為 IMQ_HOME/var 目錄。 在 Windows 上，IMQ_VARHOME 預設為 IMQ_HOME/var 目錄。 在 Windows 上，針對 Sun Java System Application Server，IMQ_VARHOME 預設為 IMQ_HOME/var 目錄。 在 Linux 上，IMQ_VARHOME 預設為 /var/opt/imq 目錄。
IMQ_JAVAHOME	<p>這是指向 Message Queue 可執行檔所需的 Java™ 執行環境 (JRE) 位置的環境變數：</p> <ul style="list-style-type: none"> 在 Solaris 上，IMQ_JAVAHOME 預設為 /usr/j2se/jre 目錄，但是，使用者可以選擇性地將值配置為所需 JRE 駐留的任意位置。 在 Windows 上，IMQ_JAVAHOME 預設為 IMQ_HOME/jre，但是，使用者可以選擇性地將值配置為所需 JRE 駐留的任意位置。 在 Linux 上，Message Queue 首先在 /usr/java/j2sdkVersion 目錄中查找 Java 執行階段，然後在 /usr/java/j2reVersion 目錄中查找，但是，使用者可以選擇性地將 IMQ_JAVAHOME 的值配置為所需 JRE 駐留的任意位置。

在本指南中，`IMQ_HOME`、`IMQ_VARHOME` 以及 `IMQ_JAVAHOME` 的顯示不附帶平台特定的環境變數表示法或語法 (例如 UNIX® 上的 `$IMQ_HOME`)。路徑名稱通常使用 UNIX 目錄分隔符號表示法 (/)。

其他說明文件資源

除了本指南以外，Message Queue 還提供了其他說明文件資源。

Message Queue 說明文件集

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

表 4 Message Queue 說明文件集

文件	使用者	說明
Message Queue 安裝指南	開發人員與管理員	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	開發人員與管理員	包括對新功能、限定、已知錯誤以及技術性注意事項的描述。
Message Queue 管理指南	管理員，也建議開發人員使用	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 Java 用戶端開發人員指南	開發人員	提供使用 Java 用戶端程式的開發人員快速入門教學和程式設計資訊，以使用 JMS 和 SOAP/JAXM 規格的 Message Queue 實施。
Message Queue C 用戶端開發人員指南	開發人員	提供 C 用戶端程式的開發人員程式設計和參考說明文件，以使用 C 介面 (C-API) 到 Message Queue 訊息服務的功能。

線上說明

Message Queue 包括用於執行 Message Queue 訊息服務管理工作的命令行公用程式。若要存取這些公用程式的線上說明，請參閱第 90 頁的「共用指令行選項」。

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

JavaDoc

Message Queue Java 用戶端 API (包括 JMS API) 說明文件的格式為 JavaDoc，它的所在目錄因不同作業系統而異 (請參閱附錄 A 「Message Queue 資料的位置」)。

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

範例用戶端應用程式

大量的範例應用程式提供了範例用戶端應用程式碼，它們的所在目錄位置因作業系統而異 (請參閱附錄 A 「Message Queue 資料的位置」)。

請參閱該目錄及其各個子目錄中的 README 檔案。

Java 訊息服務 (JMS) 規格

JMS 規格可在以下位置找到：

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

規格包括範例用戶端程式碼。

相關的第三方網站參考

第三方 URL 為本文件的參考資料且提供了額外的相關資訊。

備註

Sun 對本文件中所提到之第三方網站資料可用性不負任何責任。Sun 對這類網站或資源中提及的內容、廣告、產品或其他資料的可用性，不做任何保證或負任何責任。Sun 對任何實際或可能的損壞，或者因使用或信任此類網站或資源上可用之內容、商品或服務而造成的損失，不負任何責任或法律責任。

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

何謂 Sun Java System Message Queue ?

Message Queue 產品為分散式應用程式的可靠、非同步訊息傳送提供了基於標準的解決方案。Message Queue 是實施 Java™ Message Service (JMS) 開放式標準的企業訊息傳送系統：事實上即是作為 JMS 參考實施。然而，Message Queue 也是具備完整功能的 JMS 提供者，且擁有輔佐企業發展的功能。

JMS 規格描述了一組訊息傳送語義和行爲、以及應用程式設計介面 (API)，它們為 Java 語言應用程式提供了在分散式環境中建立、傳送、接收和讀取訊息的常用方法 (請參閱第 37 頁的「[JMS 程式設計模型](#)」)。除了支援 Java 訊息傳送應用程式，Message Queue 還提供 C 語言介面給 Message Queue 服務 (Message Queue C-API)。

使用 Sun Java System Message Queue 軟體，在不同平台和作業系統上執行的程序可連接至一個共用 Message Queue 訊息服務 (請參閱第 36 頁的「[訊息服務架構](#)」)，以傳送和接收資訊。應用程式開發人員便可以將精力集中在應用程式的業務邏輯上，而不是集中在應用程式應如何在網路中進行可靠通訊這樣的低層細節上。

Message Queue 的功能已超出 JMS 規格的最低要求。這些功能如下：

集中式管理。 提供指令行和 GUI 工具，它們可以管理 Message Queue 服務，且管理應用程式相依的項目，例如目標、異動、長期訂閱和安全性。Message Queue 也支援 Message Queue 服務的遠端監視。

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

用戶端連接防故障備用。 自動復原用戶端連接至 Message Queue 訊息伺服器失敗的連接。

可調整的效能。 在發送可靠性要求較低的情況下，則可讓您提高 Message Queue 服務的效能。

多重傳輸。 支援 Message Queue 用戶端使用多種不同的傳輸方式 (包括 TCP 和 HTTP 以及使用安全 (SSL) 連接)，與 Message Queue 訊息伺服器進行通訊的功能。

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

SOAP 訊息傳送支援。 支援經由 JMS 訊息傳送來建立和發送 SOAP 訊息 (符合簡單物件存取協定 (SOAP) 規格的訊息)。SOAP 允許在分散式環境中的各點之間交換結構化 XML 資料。請參閱「*Message Queue Java Client Developer' Guide*」，以取得更多資訊。

請參閱附錄 G 「*選擇性 JMS 功能的 Message Queue 實施*」，以取得 JMS 有關相容議題的說明文件。

產品版本

Sun Java System Message Queue 有兩個可用版本：平台版與企業版 - 每個版本對應於不同的功能組與授權容量，如下所述。(將 Message Queue 從一個版本升級到另外一個版本的說明位於「[Message Queue 安裝指南](#)」)。

平台版

該版本可以從 Sun 網站免費下載，且隨附於 Sun Java System Application Server 平台。平台版未對 Message Queue 訊息伺服器所支援的用戶端連接做出數目限制。該版本附帶有兩種授權，如下所述：

- **基本授權。**該授權提供基本的 JMS 支援 (它是一個完整的 JMS 供應程式)，但不包括下列企業功能，如：負載平衡 (多重代理程式訊息服務)、HTTP/HTTPS 連接、安全連接服務、可延伸式連接功能、用戶端連接防故障備用、至多個使用者的佇列發送、遠端且基於訊息的監視以及 C-API 支援。該授權的有效期沒有限定，因此可以用於無較高要求的生產環境。
- **具有 90 天試用期的企業授權。**該授權包括基本授權中未提供的所有企業功能 (如支援多重代理程式訊息服務、HTTP/HTTPS 連接、安全連接服務、可延伸式連接功能、用戶端連接防故障備用、至多個使用者的佇列發送、遠端且基於訊息的監視以及 C-API 支援)。然而，軟體強制規定該授權只有 90 天的有效期，所以，它適合於評估企業版產品提供的企業功能 (請參閱「[企業版](#)」)。

備註	您可以啓動 Message Queue 訊息伺服器 - Message Queue 代理程式實例 - 來啓用 90 天試用授權，如第 124 頁的「 若要啓動使用試用企業版授權的代理程式實例 」中所述。
-----------	---

企業版

該版本用於在生產環境中部署與執行訊息傳送應用程式。它支援多重代理程式訊息服務、HTTP/HTTPS 連接、安全連接服務、可延伸式連接功能、用戶端連接防故障備用、至多個使用者的佇列發送、遠端且基於訊息的監視以及 C-API 支援。您也可以使用企業版來進行開發、載入測試訊息傳送應用程式與組件並對其進行除錯。企業版具有無限制的授權有效期，並且該授權對多重代理程式訊息服務中的代理程式數目也沒有限制，但是必須基於使用的 CPU 數目而定。

企業訊息傳送系統

企業訊息傳送系統可讓獨立的分散式應用程式或應用程式元件透過訊息互動。這些元件（無論是在同一主機、同一網路上執行，還是透過網際網路鬆散地連接在一起）均使用訊息傳送來傳輸資料以及協調各自的功能。

企業訊息傳送系統的要求

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

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

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

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

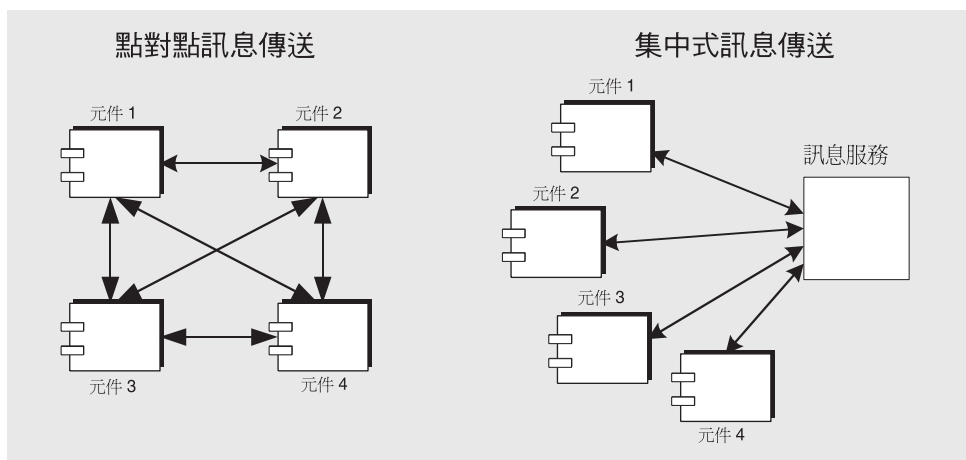
延伸性。 訊息傳送系統必須能夠在不降低效能或訊息流量的情況下，適應不斷增加的負載，即不斷增多的使用者和訊息數目。隨著商業和應用程式的延伸，這將成爲一項很重要的要求。

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

集中式訊息傳送和點對點訊息傳送

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

圖 1-1 集中式訊息傳送和點對點訊息傳送



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

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

訊息傳送系統概念

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

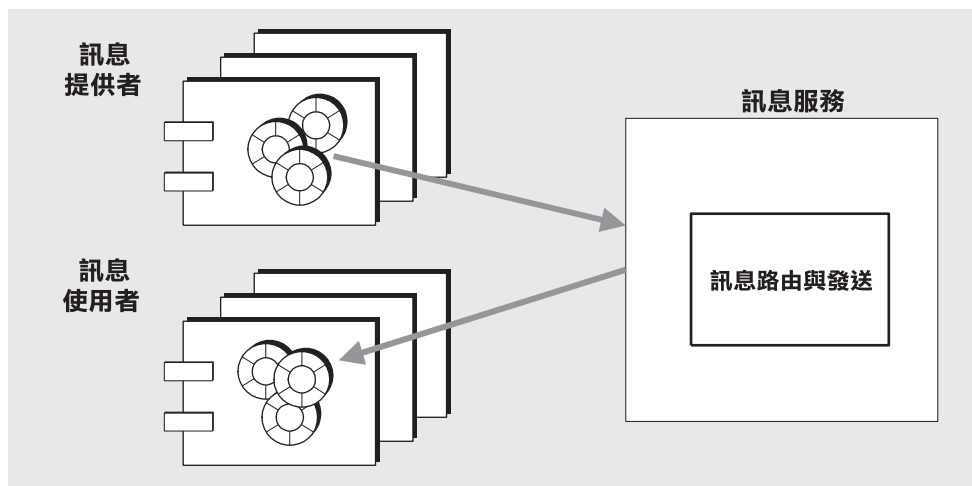
訊息

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

訊息服務架構

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

圖 1-2 訊息服務架構



訊息發送模型

產生者與使用者之間存在多種關係：一對一、一對多和多對多。例如，您發送訊息的方式可能是：

- 從一個產生者至一個使用者
- 從一個產生者至多個使用者
- 從多個產生者至一個使用者
- 從多個產生者至多個使用者

這些關係通常被簡化為兩個訊息發送模型：點對點與出版/訂閱訊息傳送。點對點發送模型主要處理由特定產生者創建並由特定使用者接收的訊息。出版/訂閱發送模型主要處理由任意數目的產生者創建並由任意數目的使用者接收的訊息。這兩個訊息發送模型可以同時使用。

過去訊息傳送系統支援這兩個訊息發送模型各種組合。Java Message Service (JMS) 規格使用 Java 程式設計的 API，建立了標準的訊息傳送語義。它同時支援點對點訊息發送模型和出版 / 訂閱訊息發送模型 (請參閱第 42 頁的「程式設計網域」)。

JMS 規格

JMS 規格規定了一組管理訊息傳送的規則和語義，包括程式設計模型、訊息架構和 API。因為 Message Queue 提供了 JMS 的實施方法，所以 JMS 概念對於理解 Message Queue 訊息傳送系統的工作原理很重要。本簡介說明了理解本書其餘各章所需的概念和術語。

JMS 訊息結構

JMS 訊息由三個部分組成：標頭、特性和內文。

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

特性 特性 (可稱為標頭的延伸) 是選擇性的 - 應用程式可以使用特性提供的值，根據各種選取條件來過濾訊息。特性為選擇性的。

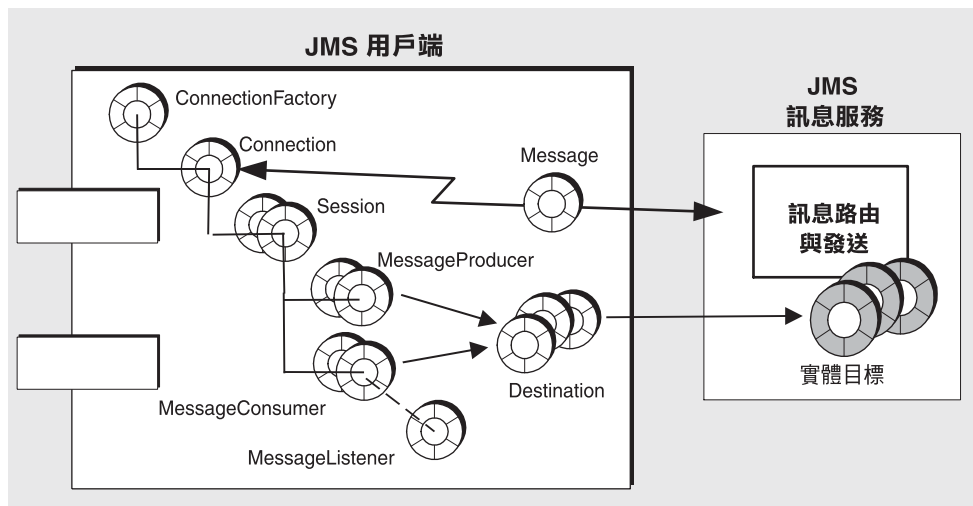
訊息內文 訊息內文包含要交換的實際資料。JMS 支援六種內文類型。

JMS 程式設計模型

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

本節介紹實施 JMS API 的物件，以及用於為訊息發送配置 JMS 用戶端的物件 (如需更多資訊，請參閱「*Message Queue Java Client Developer' Guide*」)。圖 1-3 顯示用於設計訊息發送程式的 JMS 物件。

圖 1-3 JMS 程式設計物件



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

連接用於建立階段作業。`Session` 是用於生產和使用訊息的單執行緒環境。它用於建立傳送和接收訊息的訊息產生者和使用者，並為所發送的訊息定義發送順序。階段作業透過大量的確認選項或異動，支援可靠的發送。

用戶端使用 `MessageProducer` 將訊息傳送至指定實體目標（在 API 中表示為目標識別物件）。訊息產生者可以指定預設發送模式（永久性與非永久性訊息）、優先順序和存在時間值，以控制由產生者傳送至實體目標的所有訊息。

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

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

JMS 受管理物件

JMS 規格藉由指定 *受管理物件* (可封裝提供者特定的配置資訊) 促成提供者相依的用戶端。

第 37 頁的「[JMS 程式設計模型](#)」中描述的兩個物件取決於 JMS 提供者實施 JMS 訊息服務的方法。連線工廠物件取決於提供者用於發送訊息的基本協定和機制，目標物件取決於提供者使用的特定命名慣例和實體目標功能。

通常，這些提供者特定的特徵使得 JMS 用戶端程式碼取決於特定的 JMS 實施。然而，JMS 規格要求在連線工廠和目標物件中封裝提供者特定的實施和配置資訊，接著，即可使用標準的非提供者特定的方式來進行存取。

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

受管理物件的兩種類型，即連線工廠和目標，會封裝提供者特定的資訊，但它們在用戶端中的用途有很大的差異。連線工廠用於建立至訊息伺服器的連接，而目標物件用於識別實體目標。

JMS/J2EE 程式設計：訊息驅動的 Bean

除第 37 頁的「[JMS 程式設計模型](#)」中介紹的一般 JMS 用戶端程式設計模型之外，還有專用於 Java 2 平台企業版 (J2EE 平台) 應用程式環境的 JMS。此專用的 JMS 用戶端稱為 *訊息驅動 Bean*，是 EJB 2.0 規格 (<http://java.sun.com/products/ejb/docs.html>) 中指定的企業 JavaBeans (EJB) 元件系列之一。

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

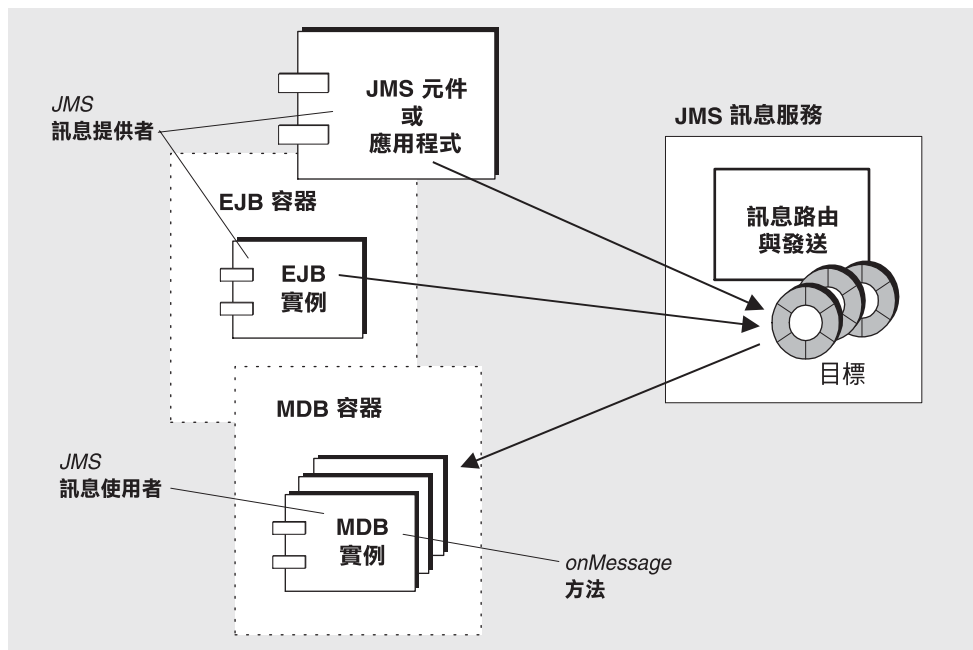
但是，非同步訊息傳送是許多企業應用程式中必不可少的要求。這些應用程式大都需要伺服器端元件在不佔用伺服器資源的情況下，能夠互相通訊和確認。因此就出現了對 EJB 元件的需求，該元件不需要緊密耦合到訊息產生者即可接收和使用訊息。對於伺服器端元件必須確認應用程式事件的任何應用程式，均必須具備此功能。在企業應用程式中，此功能還必須能夠依據不斷增加的負載來進行延伸。

訊息驅動 Bean

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

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

圖 1-4 與 MDB 進行訊息傳送



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

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

J2EE 應用程式伺服器支援

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

在 Sun Java System Application Server 中，JMS 訊息傳送資源由 Sun Java System Message Queue 提供：

- 對 Sun Java System Application Server 7.0 而言，Message Queue 訊息傳送系統已與應用程式伺服器整合，作為其原生 JMS 提供者。
- 對 Sun J2EE 1.4 Application Server 而言，Message Queue 外掛到應用程式伺服器，作為內嵌的 JMS 資源介面 (請參閱附錄 F 「Message Queue 資源介面」)。
- 如需新版的 Application Server，Message Queue 會使用標準的資源介面部署和配置方法，外掛到應用程式伺服器。

JMS 訊息傳送問題

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

JMS 提供者獨立性

JMS 指定受管理物件的用法 (請參閱第 39 頁的「JMS 受管理物件」)，以支援可以移植到其他 JMS 提供者的用戶端應用程式的開發。受管理物件可讓 JMS 用戶端使用邏輯名稱查找和參考提供者特定的物件。這樣，用戶端程式碼無需瞭解提供者使用的特定命名語法、尋址語法或配置特性，從而使程式碼獨立於提供者。

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

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

如需有關受管理物件的更多資訊，請參閱第 79 頁的「Message Queue 受管理物件」。

程式設計網域

JMS 支援兩種完全不同的訊息發送模型：點對點與出版 / 訂閱。

點對點 (佇列目標) 將訊息從產生者發送至一個使用者。在此發送模型中，目標為佇列。首先將訊息發送至佇列目標，然後根據佇列的發送策略 (請參閱第 69 頁的「佇列目標」) 將訊息從該佇列發送至已向佇列註冊的一個使用者，每次發送一封訊息。可以向佇列目標傳送訊息的產生者沒有限制，但每封訊息只能發送至一個使用者，並由一個使用者成功接收。如果沒有向佇列目標註冊的使用者，佇列將保留其接收到的訊息，並在使用者向該佇列註冊時將訊息發送給他們。

出版 / 訂閱 (主題目標) 將訊息從產生者發送至任意數目的使用者。在此發送模型中，目標為主題。訊息被首先發送至主題目標，然後發送至已訂閱該主題的所有作用中使用者。可以向主題目標傳送訊息的產生者沒有限制，並且每封訊息可以被發送至任意數目的訂閱使用者。主題目標還支援長期訂閱。長期訂閱表示使用者已向主題目標進行註冊，但在發送訊息時此使用者可以處於非作用狀態。然後，當使用者再處於作用中時，將接收此訊息。如果沒有使用者向主題目標進行註冊，則主題將不保留其收到的訊息，除非有非作用中使用者使用長期訂閱。

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

表 1-1 JMS 程式設計物件

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

1. 取決於程式設計的方法，您可以指定特定的目標類型。

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

用戶端識別碼

JMS 提供者必須支援用戶端識別碼概念，用戶端識別碼將 JMS 用戶端至訊息服務的連接與代表用戶端的訊息服務所維護的狀態資訊相關聯。根據定義，用戶端識別碼是唯一的，並且每次僅套用至一個使用者。用戶端識別碼與長期訂閱名稱（請參閱第 42 頁的「出版 / 訂閱（主題目標）」）結合使用，可以確保每個長期訂閱僅對應一個使用者。

JMS 規格允許用戶端透過 API 方法呼叫來配置用戶端識別碼，但是建議您使用連線工廠受管理物件（請參閱第 39 頁的「JMS 受管理物件」）從管理級別來配置它。但是如果硬連接至連線工廠，則每個使用者均需要個別的連線工廠以擁有唯一身份。

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

如需有關如何使用此 Message Queue 功能的詳細資訊，請參閱「*Message Queue Java Client Developer' Guide*」中有關連線工廠屬性的說明。

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

可靠的訊息傳送

JMS 定義了兩種發送模式：

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

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

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

確認 / 異動

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

確認

可以將階段作業配置為使用確認來保證可靠發送。

對於產生者，這意味著在產生者的 `send()` 方法返回之前，訊息服務將向其目標確認永久性訊息的發送。對於使用者，這意味著在訊息服務從該目標刪除永久性訊息之前，用戶端將確認來自該目標的永久性訊息的發送與使用。

本地異動

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

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

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

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

分散式異動

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

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

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

Message Queue 透過 XA 連線工廠實施對分散式異動的支援，這可讓您建立 XA 連接，而 XA 連接又可讓您建立 XA 階段作業 (請參閱第 37 頁的「[JMS 程式設計模型](#)」)。此外，對分散式異動的支援還要求協力廠商 JTS 或與 J2EE 相容的 Application Server (提供 JTS)。

永久性儲存

可靠性的另一個重要方面是，確保將永久性訊息發送至其目標後，訊息服務在將其發送至使用者之前不會遺失這些訊息。這意味著將永久性訊息發送至其目標時，訊息服務必須將其放在永久性資料存儲區中 (請參閱第 57 頁的「[持續性管理程式](#)」)。如果訊息服務由於某種原因失敗，它可以回復該訊息並將其發送至相應的使用者。雖然這增加了訊息發送的耗用時間，但卻提昇了可靠性。

訊息服務還必須儲存長期訂閱。這是因為要確保主題目標下的發送，因此僅回復永久性訊息是不夠的。訊息服務還必須回復有關主題長期訂閱的資訊，否則它將無法將訊息發送至在訊息到達時處於非作用中狀態、隨後又處於作用中狀態的用戶。

需要可靠訊息發送的訊息傳送應用程式必須將訊息指定為永久性訊息，並使用佇列目標或主題目標的長期訂閱。

效能平衡

訊息發送的可靠性越高，需要的耗用時間和頻寬就越多。可靠性與效能之間的平衡是設計時要考慮的重點。您可以透過選擇生產和使用非永久性訊息來最大化效能。另一方面，您也可以透過生產和使用永久性訊息並使用異動階段作業來最大化可靠性。在這兩種極端之間有許多選擇，這取決於應用程式的需要，包括使用 Message Queue 特定的連接和確認特性 (請參閱「[Message Queue Java Client Developer's Guide](#)」)。這些平衡在第 209 頁的「[影響效能的應用程式設計因素](#)」中有更完整說明。

訊息選取

JMS 提供了一種機制，透過它，訊息服務可以根據訊息選擇器中的條件執行訊息過濾和路由。生產型用戶端可在訊息中放入應用程式特定的特性，使用型用戶端可以使用基於這些特性的選取條件來表明它對訊息的興趣。這樣簡化了用戶端的工作並排除了給不需要訊息的用戶端額外傳送。但是，它也使得處理選取條件的訊息服務增加了一些額外耗用時間。JMS 規格中對訊息選擇器的語法和語義進行了概述。

訊息順序和優先順序

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

此外，用戶端應用程式使用訊息的順序與訊息的生產順序僅具有一個粗略的關係。這是因為向目標發送訊息以及從這些目標發送訊息，可以取決於多個影響時間的問題，如訊息的傳送順序、傳送訊息的階段作業（連接）、是否為永久性訊息、訊息的存在時間、訊息的優先順序、佇列目標的訊息發送策略（請參閱第 69 頁的「佇列目標」）以及訊息服務的可用性。

對於使用多個互連代理程式（請參閱第 72 頁的「多重代理程式叢集（企業版）」）的 Message Queue 訊息伺服器，用戶端使用訊息的排序更為複雜，因為從不同代理程式的目標發送訊息的順序是不確定的。因此，由一個代理程式發送的訊息可能會在另一個代理程式所發送的訊息之前，即使後者可能會先收到訊息。

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

Message Queue 訊息傳送系統

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

圖 2-1 Message Queue 系統架構

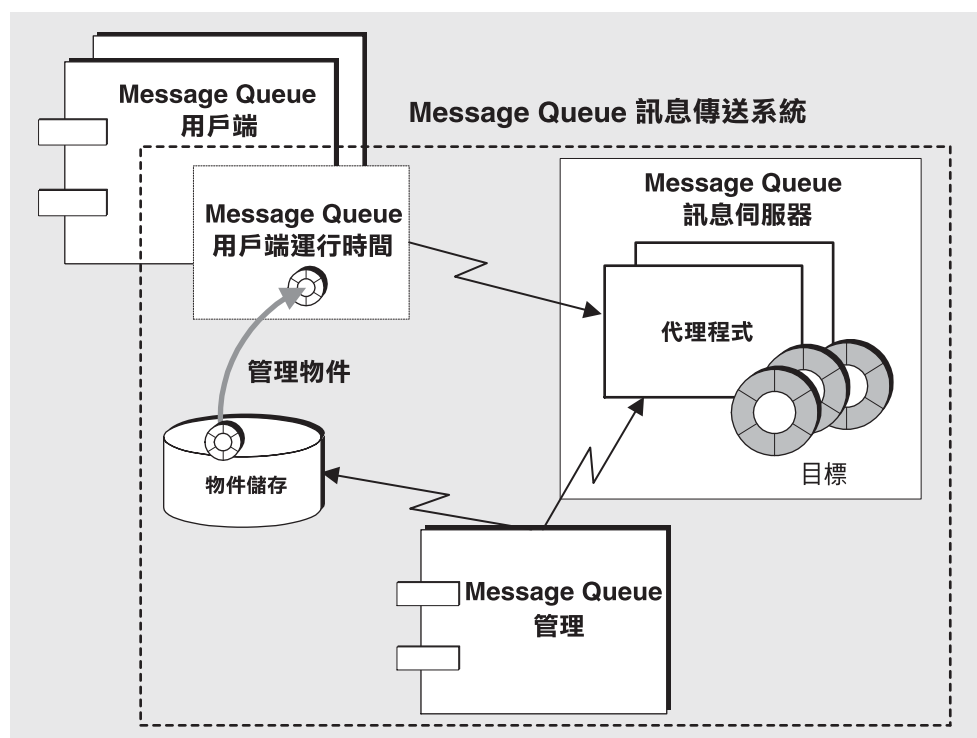


圖 2-1 中所示的 Message Queue 訊息傳送系統的主要部分如下：

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

前三個部分將在以下各節中詳細說明。最後一個部分將在第 3 章「Message Queue 管理工作和工具」中介紹。

Message Queue 訊息伺服器

本節描述第 47 頁的圖 2-1 中所示的 Message Queue 訊息伺服器的不同部分。這些部分包括：

代理程式 Message Queue 代理程式為 Message Queue 訊息傳送系統提供發送服務。訊息發送依賴一些支援元件，這些元件可處理連接服務、訊息路由與發送、持續性、安全性以及記錄（請參閱「代理程式」以獲得更多資訊）。訊息伺服器可使用一個或多個代理程式實例（請參閱第 72 頁的「多重代理程式叢集（企業版）」）。

實體目標 訊息發送過程分為兩個階段 - 從生產型用戶端發送至由代理程式維護的實體目標，然後從目標發送至一個或多個使用用戶端。實體目標表示在代理程式的實體記憶體和 / 或永久性儲存體中的位置（請參閱第 68 頁的「實體目標」以獲得更多資訊）。

代理程式

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

若要執行這套複雜的功能，代理程式將使用一些不同的內部元件，並且每個元件在發送過程中都有特定的作用。圖 2-2 中說明主代理程式元件，表 2-1 中則簡單描述這些元件。訊息路由器元件執行主要的訊息路由和發送服務，其他的訊息路由和發送服務則提供重要的支援服務給訊息路由器。

圖 2-2 代理程式服務元件

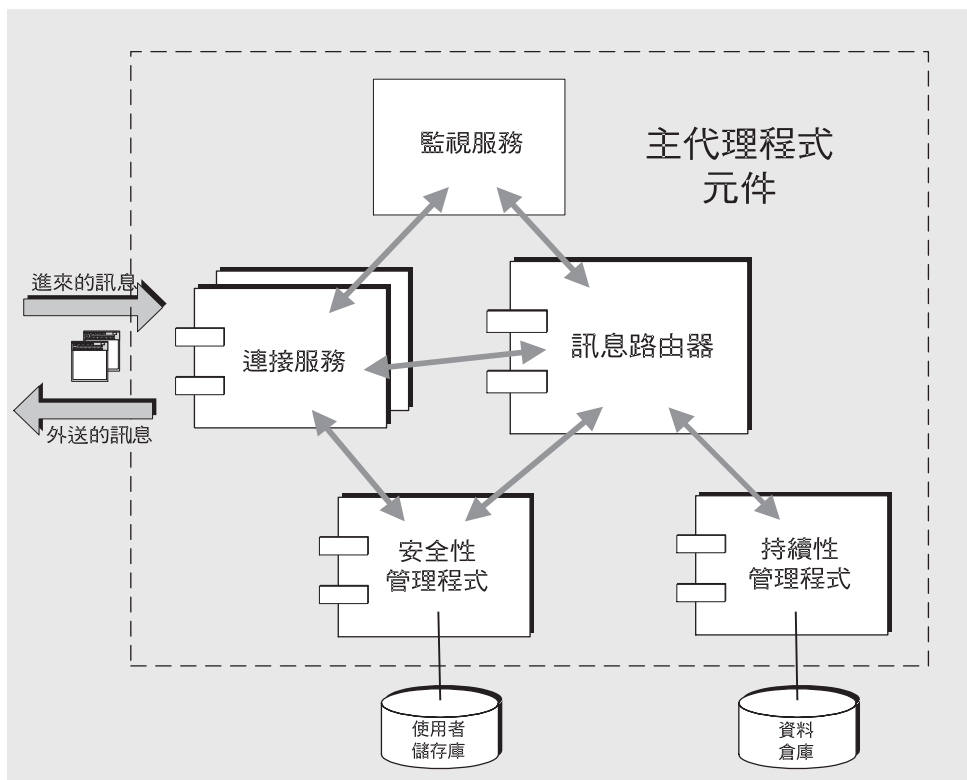


表 2-1 主代理程式服務元件和功能

元件	說明 / 功能
訊息路由器	管理訊息的路由與發送：這些訊息包括 JMS 訊息以及 Message Queue 訊息傳送系統為支援 JMS 訊息發送而使用的控制訊息。
連接服務	管理代理程式與用戶端之間的實體連接，為進來和外送的訊息提供傳輸。
持續性管理程式	管理向永久性儲存體的資料寫入，以便系統故障不會導致發送 JMS 訊息的故障。
安全性管理程式	為請求連接至代理程式的使用者提供認證服務，為已認證的使用者提供授權服務 (存取控制)。

表 2-1 主代理程式服務元件和功能 (續)

元件	說明 / 功能
監視服務	產生可以寫入大量輸出通道的度量和診斷資訊，另外，管理員可以使用輸出通道來監視和管理代理程式。

根據載入條件、應用程式的複雜性等，您可以配置這些內部元件，以最佳化代理程式的效能。以下各節更全面地探究不同的元件所執行的功能，以及可配置為影響元件運作方式的特性。

連接服務

Message Queue 代理程式支援與 Message Queue 應用程式用戶端和 Message Queue 管理用戶端的通訊 (請參閱第 87 頁的「[Message Queue 管理工具](#)」)。每種服務均由其服務類型和協定類型指定。

服務類型 指定此服務提供 JMS 訊息發送 (NORMAL) 或提供 Message Queue 管理 (ADMIN) 服務

協定類型 指定支援此服務的基本傳輸協定層。

目前，Message Queue 代理程式可提供的連接服務如表 2-2 中所示：

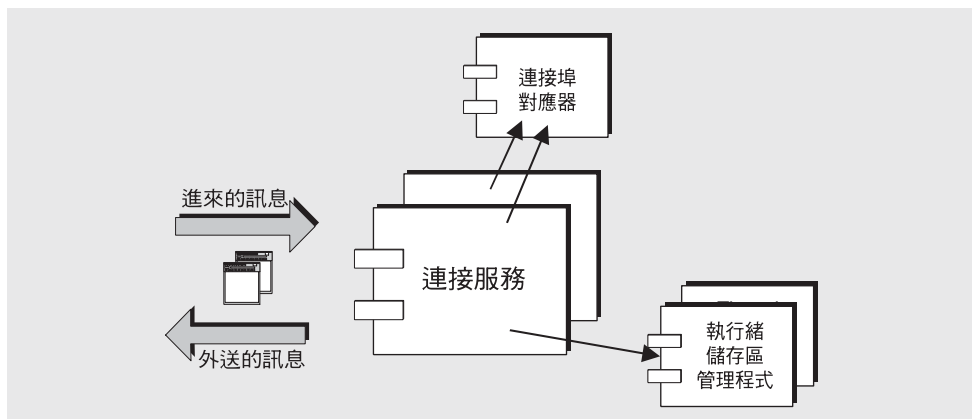
表 2-2 代理程式支援的連接服務

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

您可以將代理程式配置為執行這些連接服務的任何一種或全部。每種連接服務可用於特定的連接埠，此連接埠由代理程式的主機名稱和連接埠號指定。可以動態配置連接埠，或您可以指定連接服務為可用時的連接埠。

每種服務皆使用共用連接埠對映器登錄，但它們均有自己的執行緒儲存區管理程式，如圖 2-3 所示。

圖 2-3 連接服務支援



連接埠對應器

Message Queue 會提供一個**連接埠對應器**，以將連接埠對映到不同的連接服務。連接埠對應器本身常駐在標準的連接埠號 7676 上。當用戶端配置與代理程式的連接時，它會先聯絡連接埠對應器，以請求其所需的連接服務的連接埠號。

您也可以在配置這些連接服務時，為 `jms`、`ssljms`、`admin` 和 `ssladmin` 連接服務指定一個**靜態**連接埠號，但只有在特殊情況下才會這麼做（例如透過防火牆建立連接時），且通常不建議這麼做。使用附錄 C「[HTTP/HTTPS 支援 \(企業版\)](#)」的**第 276 頁的表 C-1** 和 **第 287 頁的表 C-3** 中描述的特性，可分別配置 `httpjms` 和 `httpsjms` 服務。

執行緒儲存區管理程式

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

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

專屬模型 每個到代理程式的連接均要求兩個專屬執行緒：一個專門負責處理連接進來的訊息，另一個則專門處理連接外送的訊息。這可將連接數目限制為執行緒儲存區中最大執行緒數目的一半，但它可提供高效能。

共用模型 (企業版) 傳送或接收訊息時，共用執行緒會負責處理連接。因為每個連接皆沒有要求專屬執行緒，所以此模型會增加一個連接服務 (和代理程式) 可支援的連接數目。但也會增加共用執行緒中效能耗用的時間。執行緒儲存區管理程式使用一套分散執行緒，可監視連接狀態並依需要將連接指定給執行緒。透過限制每個此類分散執行緒監視的連接數目，在此活動中效能耗用的時間可以縮到最短。

安全性

每種連接服務可支援特定認證和授權 (存取控制) 功能 (請參閱第 60 頁的「安全性管理程式」)。

連接服務特性

表 2-3 中顯示與連接服務相關的可配置特性。(如需有關配置這些特性的說明，請參閱第 5 章「啟動與配置代理程式」)

表 2-3 連接服務特性

特性名稱	說明
<code>imq.service.activelist</code>	要在啟動代理程式時變為作用中的連接服務的清單，這些連接服務依名稱列出並以逗號分隔。支援的服務為： <code>jms</code> 、 <code>ssljms</code> 、 <code>httpjms</code> 、 <code>httpsjms</code> 、 <code>admin</code> 和 <code>ssladmin</code> 。 預設： <code>jms</code> 和 <code>admin</code>
<code>imq.ping.interval</code>	代理程式透過連接，連續嘗試偵測 Message Queue 用戶端執行時間的期間。 預設：120 秒
<code>imq.hostname</code>	如果有多台主機可用 (例如，如果一台電腦中有多張網路介面卡)，則指定所有連接服務所連結的主機 (主機或 IP 位址)。 預設：所有可用 IP 位址
<code>imq.portmapper.port</code>	指定代理程式的主要連接埠 - 連接埠對映器常駐的連接埠。如果您要在主機上執行多個代理程式實例，則必須為每個實例指定唯一的連接埠對映器連接埠。 預設：7676
<code>imq.portmapper.hostname</code>	如果有多台主機可用 (例如，如果一台電腦中有多張網路介面卡)，則指定連接埠對映器所連結的主機 (主機或 IP 位址)。 預設：繼承 <code>imq.hostname</code> 的值
<code>imq.portmapper.backlog</code>	指定拒絕請求前，連接埠對映器可以處理運作請求的最大數目。此特性可以配置請求數目，另外，這些請求可儲存在作業系統儲存區，等待連接埠對映器進行處理。 預設：50

表 2-3 連接服務特性 (續)

特性名稱	說明
<code>imq.service_name.protocol_type¹.port</code>	僅用於 <code>jms</code> 、 <code>ssljms</code> 、 <code>admin</code> 和 <code>ssladmin</code> 服務，為已命名的連接服務指定連接埠號。 預設：0 (連接埠由連接埠對映器動態配置) 若要配置 <code>httpjms</code> 和 <code>httpsjms</code> 連接服務，請參閱附錄 C 「HTTP/HTTPS 支援 (企業版)」。
<code>imq.service_name.protocol_type¹.hostname</code>	僅用於 <code>jms</code> 、 <code>ssljms</code> 、 <code>admin</code> 和 <code>ssladmin</code> 服務，如果有多台主機可用 (例如，如果一台電腦中有多張網路介面卡)，指定已命名的連接服務所連結的主機 (主機名稱或 IP 位址)。 預設：繼承 <code>imq.hostname</code> 的值
<code>imq.service_name.min_threads</code>	指定執行緒數目，一旦達到指定數目，執行緒便會在執行緒儲存區中維護，供已命名的連接服務使用。 預設：取決於連接服務 (請參閱第 118 頁的表 5-1)。
<code>imq.service_name.max_threads</code>	指定執行緒數目，一旦超過指定數目，系統便不會將新的執行緒新增至執行緒儲存區，供已命名的連接服務使用。此數目必須大於零，並且其值必須大於 <code>min_threads</code> 的值。 預設：取決於連接服務 (請參閱第 118 頁的表 5-1)。
<code>imq.service_name.threadpool_model</code>	指定執行緒是專屬於連接 (<code>dedicated</code>) 或依需要由連接共用 (<code>shared</code>)，以用於已命名的連接服務。共用模型 (執行緒儲存區管理) 會增加代理程式所支援的連接數目，但僅實施用於 <code>jms</code> 和 <code>admin</code> 連接服務。 預設：取決於連接服務 (請參閱第 118 頁的表 5-1)。
<code>imq.shared.connectionMonitor_limit</code>	僅用於共用執行緒儲存區模型，指定可由分散執行緒監視的最大連接數目。(系統配置了足夠的分散執行緒以監視所有連接。) 此值越小，系統將作用中的連接指定給執行緒的速度就越快。值 -1 表示沒有限制。 預設：取決於作業系統 (請參閱第 118 頁的表 5-1)。

1. `protocol_type` (協定類型) 在表 2-2 中指定。

訊息路由器

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

基本的發送機制

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

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

- 如果目標為主題，則 JMS 訊息將立即路由至此主題的所有作用中用戶。如果用戶為非作用中的長期用戶，則訊息路由器將保留此訊息，直至此用戶成為作用中用戶，然後將此訊息發送至此用戶。
- 如果目標為佇列，則 JMS 訊息將放置在相應的佇列中，然後在此訊息到達佇列前面時被發送至適當的使用者。訊息到達佇列前面的順序，取決於它們到達的順序及優先順序。

一旦訊息路由器將訊息發送至所有預定的使用者，它便會將此訊息從記憶體中清除，如果此訊息為永久性的（請參閱第 43 頁的「可靠的訊息傳送」），它會將其從代理程式的永久性資料儲存區中移除。

可靠發送：確認與異動

當為可靠發送（請參閱第 43 頁的「可靠的訊息傳送」）新增需求時，剛才描述的發送機制會變得更加複雜。可靠發送包含兩個方面：確保向代理程式傳送及從其中發出的訊息發送成功，並確保實際發送訊息之前，代理程式不會遺失訊息或發送資訊。

為確保訊息成功遞送至代理程式及從其中成功發出，Message Queue 使用了一些名為確認的控制訊息。

例如，當產生者將 JMS 訊息（收費載入訊息而不是控制訊息）傳送至目標時，代理程式會傳送回控制訊息（即代理程式確認），表明它已接收到此 JMS 訊息。（依預設，僅當產生者指定 JMS 訊息為永久性時，Message Queue 才會執行此作業。）生產型用戶端使用代理程式確認，以確保發送至目標（請參閱第 77 頁的「訊息生產」）。

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

如果用戶為主題的長期訂閱，則訊息路由器會將每個 JMS 訊息保留在目標中，並當每個長期用戶成為作用中使用者時發送此訊息。訊息路由器接收用戶端確認時會記錄這些確認，並僅在已接收所有確認後才刪除 JMS 訊息（除非 JMS 訊息在此之前已過期）。

而且，透過將代理程式確認傳送回用戶端，訊息路由器確認用戶端確認的接收。使用用戶端使用代理程式確認，以確保代理程式不會多次發送同一 JMS 訊息（請參閱第 78 頁的「訊息使用」）。如果由於某種原因而使代理程式無法接收到用戶端確認，則可能發生上述情況。

如果代理程式沒有接收到用戶端確認並再次發送 JMS 訊息，則此訊息會標記有重新發送旗標。如果用戶端連接在代理程式接收到用戶端確認之前關閉，並隨後開啓新的連接，則此代理程式通常會重新發送 JMS 訊息。例如，如果佇列的訊息使用者在確認訊息之前離線，而且另一個使用者隨後註冊此佇列，則代理程式將未確認的訊息重新發送至此新的使用者。

以上描述的用戶端與代理程式確認過程，還適用於群組至異動中的 JMS 訊息發送。在這些情況下，用戶端與代理程式確認可對異動作業，亦可對傳送或接收的個別 JMS 訊息作業。確定異動時，代理程式確認會被自動傳送。

代理程式會追蹤異動，並可在發生故障時對異動進行確定或回轉。本異動管理還支援本機異動，此為較大型分散式異動（請參閱第 44 頁的「分散式異動」）的一部分。代理程式會追蹤這些異動的狀態，直至它們被確定。代理程式啓動時，它會檢查所有未確定的異動，並依預設，會回轉所有異動（那些處於 PREPARED 狀態的異動除外）。

可靠發送：持續性

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

產生者用戶端可指定訊息為永久性的，在此情況下，訊息路由器會將此訊息傳送至將此訊息儲存在資料庫或檔案系統中的**持續性管理程式**（請參閱第 57 頁的「持續性管理程式」），以便在代理程式出現故障時回復此訊息。

管理記憶體資源和訊息流量

代理程式的效能和穩定性取決於可用的系統資源，以及如何有效利用記憶體之類的資源。要特別注意的是，訊息產生速度快於使用速度時，會使用到所有的記憶體資源，因而訊息路由器可能過度負載。若要預防發生此情形，訊息路由器會使用三個記憶體保護級別，以在資源不足時維持系統作業：

個別目標上的訊息限制 您可以配置實體目標上的屬性，以指定訊息數目和訊息使用的總記憶體（請參閱第 154 頁的表 6-10）。您也可以指定到達任何此類限制時，訊息路由器將產生的四個確認類型。四個限制行爲：

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

系統範圍訊息限制 系統範圍訊息限制為第二道保護。您可以指定適用於系統上之所有目標的系統範圍限制：訊息總數目和所有訊息使用的記憶體（請參閱 [第 56 頁的表 2-4](#)）。如果到達任何系統範圍訊息限制，訊息路由器會拒絕新的訊息。

系統記憶體執行緒 系統記憶體執行緒為第三道保護。您可以指定代理程式採取越來越重要的動作時可用的系統記憶體執行緒，以防止記憶體超過負載。此動作取決於記憶體資源的狀態：`green`（大量記憶體可用）、`yellow`（代理程式記憶體正在減少）、`orange`（代理程式記憶體不足）以及 `red`（代理程式無記憶體）。在代理程式的記憶體狀態從 `green` 變為 `yellow` 和 `orange`，然後變為 `red` 的過程中，代理程式會採取越來越重要的動作，動作類型如下所示：

- 將訊息移出作用中記憶體並放入永久性儲存體（請參閱 [第 57 頁的「持續性管理程式」](#)）；通常不會儲存的非永久性訊息，可能會被移出，如此系統即可回收記憶體。
- 減低非永久性訊息的產生者，最終使訊息停止流入代理程式（永久性訊息流量會自動受到代理程式確認訊息之要求的限制）。

上述兩種方法皆會降低效能。

如果到達系統記憶體執行緒，那表示您尚未適當地按目標配置訊息限制和系統範圍訊息限制。在某些情況下，您無法及時為執行緒擷取潛在的記憶體負載。因此，您不應依賴此功能來控制記憶體資源，而是應個別且整體地配置目標，以最佳化記憶體資源。

訊息路由特性

[表 2-4](#) 中詳細說明了管理記憶體資源的系統範圍限制和系統記憶體執行緒。（如需有關配置這些特性的說明，請參閱 [第 5 章「啟動與配置代理程式」](#)）

表 2-4 訊息路由器特性

特性名稱	說明
<code>imq.message.expiration.interval</code>	指定回收已過期訊息發生的頻率（以秒為單位）。預設：60
<code>imq.system.max_count</code>	指定代理程式保留訊息的最大數目。其他訊息將被拒絕。值 <code>-1</code> 表示沒有限制。預設： <code>-1</code>
<code>imq.system.max_size</code>	指定代理程式保留訊息的最大總容量（以位元組、千位元組或百萬位元組為單位）。其他訊息將被拒絕。值 <code>-1</code> 表示沒有限制。預設： <code>-1</code>

表 2-4 訊息路由器特性 (續)

特性名稱	說明
<code>imq.message.max_size</code>	指定訊息內文的最大允許容量 (以位元組、千位元組或百萬位元組為單位)。任何大於該配置的訊息將被拒絕。值 -1 表示沒有限制。 預設：70m (百萬位元組)
<code>imq.resource_state.threshold</code>	指定啟動每種記憶體狀態時的記憶體利用百分比。資源狀態的值可以為 <code>green</code> 、 <code>yellow</code> 、 <code>orange</code> 和 <code>red</code> 。 預設：分別為 0、80、90 和 98
<code>imq.resource_state.count</code>	指定在啟動每個記憶體資源狀態時，批次中允許進來的訊息的最大數目。系統記憶體變得逐漸不足時，此限制會減低訊息產生者。 預設：分別為 5000、500、50 和 0
<code>imq.transaction.autorollback</code>	指定處於 <code>PREPARED</code> 狀態的分散式異動是否可在啟動代理程式時自動回轉 (<code>true/false</code>)。如果為 <code>false</code> ，則您必須使用 <code>imqcmd</code> 手動確定或回轉異動 (請參閱第 162 頁的「管理異動」)。 預設： <code>false</code>

持續性管理程式

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

若要回復發生故障的代理程式，不僅只要復原未發送的訊息。代理程式還必須可執行以下作業：

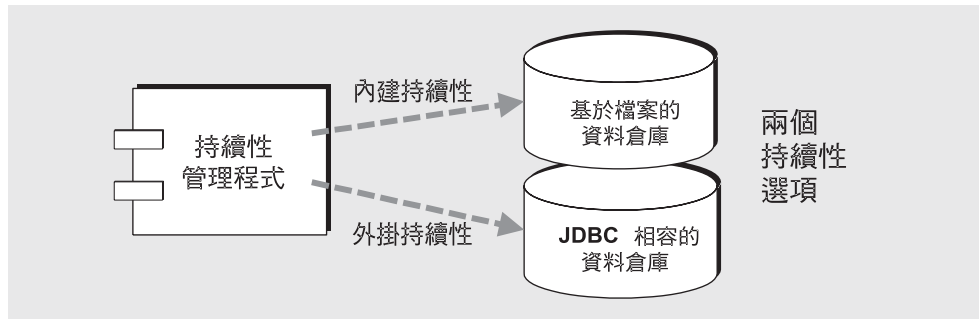
- 重新建立目標
- 復原每個主題的長期訂閱清單
- 復原每個訊息的確認清單
- 重新生產所有已確定異動的狀態

持續性管理程式管理所有此狀態資訊的儲存和擷取。

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

Message Queue 支援內建和外掛持續性模組 (請參閱圖 2-4)。內建持續性是基於檔案的資料倉庫。外掛持續性使用 Java 資料庫連接 (JDBC™) 介面，並需要 JDBC 相容的資料倉庫。內建持續性通常快於外掛持續性；但是某些使用者更喜歡使用了 JDBC 相容資料庫系統的容錯和管理功能。

圖 2-4 持續性管理程式支援



內建持續性

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

基於檔案的資料倉庫儲存於以代理程式實例 (*instanceName*) 作為辨別依據的目錄中，且代理程式實例與資料倉庫相互關聯 (請參閱附錄 A 「Message Queue 資料的位置」)：

```

_/instances/instanceName/fs350/

```

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

若要減少因新增和移除訊息而產生的分段程序，您可以壓縮智慧型記錄檔案 (請參閱第 159 頁的「壓縮目標」)。除此之外，內建的持續性管理程式會將大小超過可配置執行緒的訊息 (`imq.persist.file.message.max_record_size`)，儲存在其個別檔案中，而非儲存於智慧型記錄檔案。針對這些個別檔案，要維護檔案儲存區，以便檔案被重新使用。當不再需要一個訊息檔案時，不會將它刪除，而是將它新增到其目標目錄中可用檔案的儲存區，以使用來儲存新的訊息。

您可以在檔案儲存區中配置檔案的最大數目

(`imq.persist.file.destination.message.filepool.limit`)，並在檔案儲存區中，指定欲清除之可用檔案的百分比 (the `imq.persist.file.message.filepool.cleanratio`)，即截斷至零，而不是僅做標記以供重新使用 (未截斷)。已清除檔案的百分比越高，維護此檔案儲存區所需的磁碟空間就越少，但耗用時間就越多。您還可以指定是否要在關機時清除已標記檔案 (`imq.persist.file.message.cleanup`)。如果這些檔案被清除，則它們將佔用很少的磁碟空間，但關閉代理程式則需較長時間。

所有其他永久性資料 (目標、長期訂閱和異動) 皆儲存在其個別檔案中：一個檔案中的所有目標、其他檔案中的所有長期訂閱等等。

若要最大化可靠性，您可以指定 (`imq.persist.file.sync.enabled`) 持續性作業同步化包含實體儲存裝置的內部記憶體狀態同步。這可以消除因系統當機而產生的資料遺失，但會影響效能。

因為資料倉庫可以包含敏感或專用性質的訊息，所以建議您保護 `...instances/instanceName/fs350/` 目錄，以防未經授權的存取。如需說明，請參閱第 302 頁的「保護永久性資料」。

外掛持續性

您可以配置代理程式，以透過 JDBC 驅動程式存取任何可存取的資料倉庫。這包含配置一些與 JDBC 相關的代理程式配置特性，以及使用資料庫管理員公用程式 (`imqdbmgr`) 來建立具有適當綱目的資料倉庫。附錄 B「設定外掛持續性」中詳細介紹了這些程序和相關的配置特性

持續性管理程式特性

第 59 頁的表 2-5 中詳細介紹了與持續性相關的配置特性。(如需有關配置這些特性的說明，請參閱第 5 章「啟動與配置代理程式」)

除了這些特性中的第一個屬性，所有表 2-5 中的屬性皆僅附屬內建持續性。第 266 頁的表 B-1 中說明了附屬插入式持續性的特性。

表 2-5 持續性管理程式特性

特性名稱	說明
<code>imq.persist.store</code>	指定代理程式使用內建、基於檔案 (<code>file</code>) 的持續性，或使用外掛 JDBC 相容 (<code>jdbc</code>) 的持續性。 預設： <code>file</code>
<code>imq.persist.file.sync.enabled</code>	指定持續性作業是否與實體儲存裝置以記憶體狀態同步化。如果為 <code>true</code> ，則會結束由系統當機引起的資料遺失，但會降低持續性作業的效能。 預設： <code>false</code>
<code>imq.persist.file.message.max_record_size</code>	針對內建且基於檔案的持續性，請指定訊息新增到訊息儲存檔案的最大容量，而非儲存到個別檔案。 預設： <code>1m</code> (百萬位元組)
<code>imq.persist.file.destination.message.filepool.limit</code>	針對內建且基於檔案的持續性，指定目標檔案儲存區中，可供重新使用之可用檔案的最大數目。此數目越大，代理程式處理永久性資料的速度就越快。超過此數目的可用檔案將被刪除。代理程式將依需要建立與刪除超過此限制的其他檔案。 預設： <code>100</code>

表 2-5 持續性管理程式特性 (續)

特性名稱	說明
imq.persist.file.message. filepool.cleanratio	針對內建且基於檔案的持續性，指定目標檔案儲存區中被維護處於清除狀態之可用檔案的百分比 (截斷至零)。此值越高，則在作業期間清除檔案所需的耗用時間就越多，但檔案儲存區所需的磁碟空間就越少。 預設：0
imq.persist.file.message. cleanup	針對內建且基於檔案的持續性，請指定代理程式關閉時，是否會清除目標檔案儲存區中的可用檔案。值 <code>false</code> 會加速代理程式的關閉速度，但檔案倉庫需要更多的磁碟空間。 預設： <code>false</code>

安全性管理程式

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

認證與授權功能因使用者儲存庫而異 (請參閱第 61 頁的圖 2-5)：包含有關訊息傳送系統使用者的資訊 (如使用者名稱、密碼和群組成員身份) 的檔案、目錄或資料庫。當請求與代理程式連接時，這些名稱和密碼用於認證使用者。使用者名稱和群組成員身份，連同存取控制檔案一起用於授權作業 (如針對目標生產或使用訊息)。

Message Queue 管理員可寫入 Message Queue 所提供的使用者儲存庫 (請參閱第 184 頁的「使用文字檔案使用者儲存庫」)，或將預先存在的 LDAP 使用者儲存庫插入安全性管理程式元件中 (請參閱第 190 頁的「將 LDAP 伺服器用於使用者儲存庫」)。文字檔案使用者儲存庫易於使用，但其安全性易受侵害，因此應僅將其用於評估與開發，而 LDAP 使用者儲存庫很安全，因此最適合用於生產。

認證

Message Queue 安全性支援基於密碼的認證。當用戶端請求連接至代理程式時，它必須提交使用者名稱和密碼。安全性管理程式將此用戶端提交的名稱和密碼與儲存在使用者儲存庫中的名稱和密碼進行比較。將此密碼從用戶端傳送至代理程式時，系統會使用基本 64 編碼或訊息摘要 (MD5) 對密碼進行編碼。如需更加安全的傳送，請參閱第 62 頁的「加密 (企業版)」。您可以分別配置每種連接服務所使用的編碼類型，或在代理程式範圍基礎上配置編碼。

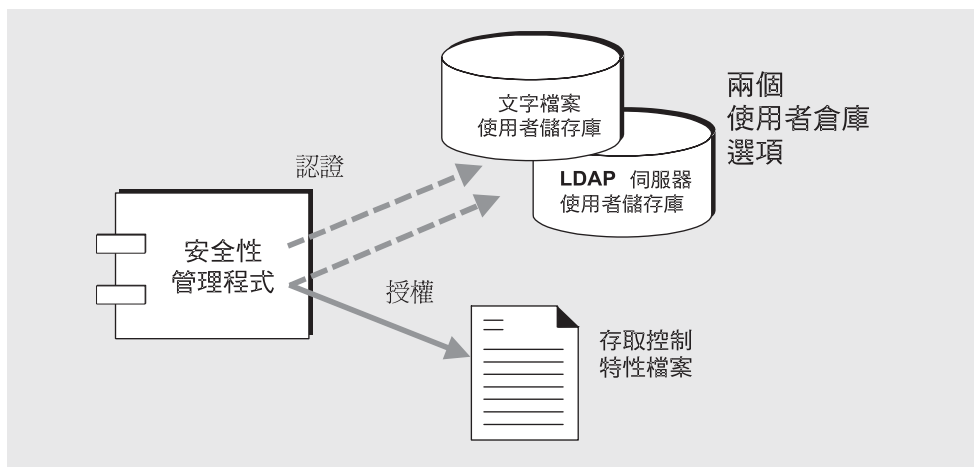
授權

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

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

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

圖 2-5 安全性管理程式支援



預設存取控制特性檔案僅明確參考一個群組：*admin* (請參閱第 187 頁的「群組」)。*admin* 群組中的使用者擁有管理服務連接許可權。管理服務可讓使用者執行管理功能，如建立目標以及監視與控制代理程式。依預設，您定義的任何其他群組中的使用者均無法取得管理服務連接。

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

此外，如果代理程式已配置為允許自動建立目標（請參閱第 70 頁的「自動建立的（對應管理員建立的）目標」），則您可以透過編輯存取控制特性檔案來控制代理程式可為何人自動建立目標。

加密（企業版）

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

若要使用 Message Queue 基於 SSL 的連接服務，您將使用鍵值工具公用程式 (imqkeytool) 產生私密密鑰 / 公開密鑰對。此公用程式將公開密鑰內嵌在自身簽名的憑證中，並放置在 Message Queue 密鑰儲存區中。Message Queue 密鑰儲存區自身具有密碼保護；若要對其解除鎖定，您必須在啟動時提供密鑰儲存區密碼。請參閱第 198 頁的「加密：使用基於 SSL 的服務（企業版）」。

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

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

表 2-6 安全性管理程式特性

特性名稱	說明
imq.authentication.type	指定密碼應以基本 64 程式碼 (basic) 傳送，或作為 MD5 摘要 (digest) 傳送。為代理程式支援的所有連接服務配置編碼。 預設：digest
imq.service_name.authentication.type	指定密碼應以基本 64 程式碼 (basic) 傳送，或作為 MD5 摘要 (digest) 傳送。為已命名的連接服務配置編碼，並置換任何代理程式範圍的配置。 預設：繼承 imq.authentication.type 的值
imq.authentication.basic.user_repository	(用於基本 64 程式碼) 指定用於認證的使用者儲存庫的類型，即基於檔案 (file) 或 LDAP (ldap)。如需其他 LDAP 特性，請參閱第 191 頁的表 8-5。 預設：file

表 2-6 安全性管理程式特性 (續)

特性名稱	說明
imq.authentication. client.response.timeout	指定系統等待用戶端確認來自代理程式的認證請求的時間 (以秒為單位)。 預設：180 (秒)
imq.accesscontrol. enabled	為代理程式支援的所有連接服務配置存取控制 (true/false)。表示系統是否將檢查已認證的使用者擁有使用連接服務的許可權，或擁有執行與特定目標相關的特定 Message Queue 作業的許可權 (如存取控制特性檔案中所指定)。 預設：true
imq.service_name. accesscontrol.enabled	為已命名的連接服務配置存取控制 (true/false)，並置換代理程式範圍的配置。表示系統是否將檢查已認證的使用者擁有使用已命名連接服務的許可權，或擁有執行與特定目標相關的特定 Message Queue 作業的許可權 (如存取控制特性檔案中所指定)。 預設：繼承 imq.accesscontrol.enabled 的值
imq.accesscontrol.file. filename	為代理程式實例支援的所有連接服務，指定存取控制特性檔案的名稱。檔案名稱可指出到存取控制目錄的相關檔案路徑 (請參閱附錄 A 「 Message Queue 資料的位置 」)。 預設：accesscontrol.properties
imq.service_name. accesscontrol.file. filename	為代理程式實例的已命名連接服務，指定存取控制特性檔案的名稱。檔案名稱可指出到存取控制目錄的相關檔案路徑 (請參閱附錄 A 「 Message Queue 資料的位置 」)。 預設：繼承 imq.accesscontrol.file.filename 的值。
imq.passfile.enabled	指定是否已在密碼檔案中指定用於安全通訊的使用者密碼 (用於 SSL、LDAP、JDBC™) (true/false)。 預設：false
imq.passfile.dirpath	指定包含密碼檔案之目錄的路徑 (因作業系統而異)。 預設：請參閱附錄 A 「 Message Queue 資料的位置 」
imq.passfile.name	指定密碼檔案的名稱。 預設：passfile
imq.keystore.property_name	用於基於 SSL 的服務：指定與 SSL 密鑰儲存相關的安全性特性。請參閱第 200 頁的表 8-8。

監視服務

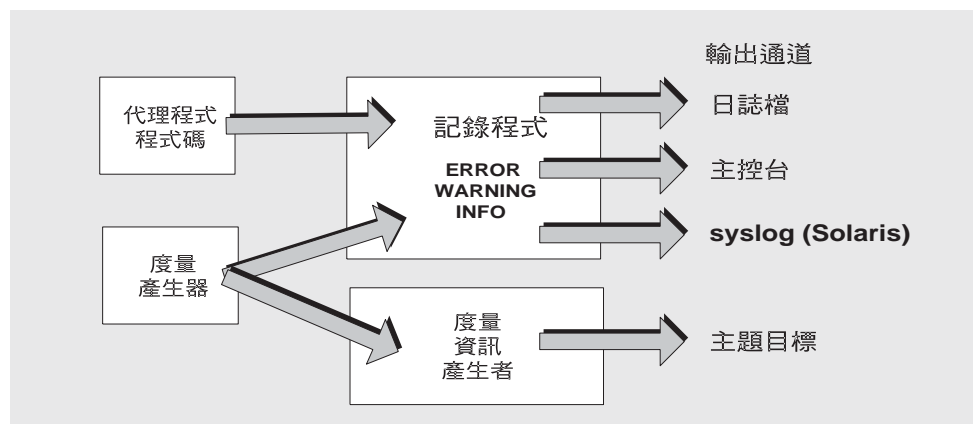
代理程式包括一些用於監視和診斷其作業的元件。這些元件如下：

- 產生資料的元件 (記錄事件與度量產生器的代理程式程式碼)
- 記錄程式元件 (請參閱「[記錄程式](#)」)，可透過一些輸出通道寫出資訊

- 訊息產生器，可將包含度量資訊的 JMS 訊息傳送到 JMS 監視用戶端使用的主題目標。

圖 2-6 中說明了一般機制。

圖 2-6 監視服務支援



度量產生器

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

您可以開啓或關閉度量資料的產生，並指定產生度量報告的頻率。

記錄程式

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

您可以指定記錄程式收集的資訊類型，以及寫入每個輸出通道的類型。

例如您可以指定記錄程式級別，即記錄程式收集的資訊類型，範圍從最嚴重且最重要的資訊 (錯誤) 到不太重要的資訊 (度量資料)。表 2-7 中以重要性的降序方式顯示了資訊的種類：

表 2-7 記錄種類

種類	說明
ERROR (錯誤)	表示可導致系統故障問題的訊息。

表 2-7 記錄種類 (續)

種類	說明
WARNING (警告)	應加以注意但不會導致系統故障的警示。
INFO (資訊)	度量和其他資訊性訊息的報告。

若要配置記錄程式級別，您可指定其中一個種類。記錄程式將寫出指定種類和所有更高級種類的資料。例如，如果您指定 WARNING 級別的記錄，則記錄程式將寫出警告資訊和錯誤資訊。

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

在日誌檔中，您還可以指定何時關閉日誌檔並將輸出自動重建至新檔案。一旦日誌檔達到指定容量或存在時間，系統將儲存此日誌檔並建立新的日誌檔。日誌檔會被寫入以代理程式實例名稱 (*instanceName*) 作為辨別依據的目錄中，且代理程式實例與日誌檔相互關聯 (請參閱附錄 A 「Message Queue 資料的位置」)：

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

當建立新的自動重建日誌檔時，將會保留 9 個最新日誌檔的歸檔檔案。

如需配置記錄程式的相關資訊，請參閱第 66 頁的表 2-9 和第 134 頁的「變更記錄程式配置」。

度量資訊產生者 (企業版)

訊息產生者元件會在固定時間間隔，從度量產生器元件收到資訊，並將資訊寫入訊息，接著根據訊息中包含的度量資訊類型，將訊息傳送到一些度量主題目標，

總共有 5 個度量主題目標。表 2-8 中顯示了這些目標的名稱，和發送到每個目標的度量訊息類型。

表 2-8 度量主題目標

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

表 2-8 度量主題目標

主題目標名稱	度量訊息的類型
<code>mq.metrics.destination.topic.monitoredDestinationName</code>	用於指定名稱之主題的目標度量

指定至這些度量主題目標的 Message Queue 用戶端，會使用目標中的訊息並處理訊息中包含的度量資訊。例如，用戶端可以指定到 `mq.metrics.broker` 目標，以接收和處理像是代理程式中訊息總數目的這類資訊。

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

度量訊息產生者產生的訊息類型為 MapMessage；它們是根據訊息包含的度量類型，由一些名稱 / 值對而組成。每個名稱 / 值對會對應到一個度量數目和度量數目的值。例如，代理程式度量訊息包含 12 個左右的度量數目的值，這些數目包括流入和流出代理程式的訊息數目和大小，以及目前記憶體中的訊息數目和大小等。如需每個度量訊息類型中報告的度量數目詳細資訊，請參閱 Message Queue Java 用戶端開發人員指南，以瞭解如何寫入使用度量訊息的 Message Queue 用戶端。

除了包含於度量訊息主體的度量資訊外，每個訊息標頭還有兩個特性：一個是指定度量訊息的類型，另一個是記錄時間標記。Message Queue 用戶端應用程式可使用這些標頭特性，以便從度量訊息擷取資料和記錄對應的時間標記。

監視服務特性

表 2-9 中顯示代理程式的可配置特性，它們用於配置資訊的產生、記錄和度量訊息產生。(如需有關配置這些特性的說明，請參閱第 5 章「啟動與配置代理程式」)

表 2-9 監視服務特性

特性名稱	說明
<code>imq.metrics.enabled</code>	指定 (true/false) 是否要將度量資訊寫入記錄程式。不會影響度量訊息的產生 (請參閱 <code>imq.metrics.topic.enabled</code>)。預設：true
<code>imq.metrics.interval</code>	如果已啓用度量記錄程式 (<code>imq.metrics.enabled=true</code>)，則請指定將度量資訊寫入記錄程式的時間間隔 (以秒為單位)。值 -1 表示永不報告。不會影響產生度量訊息的時間間隔 (請參閱 <code>imq.metrics.topic.interval</code>)。預設：-1

表 2-9 監視服務特性 (續)

特性名稱	說明
<code>imq.log.level</code>	指定記錄程式級別：可寫入至輸出通道的輸出種類。包括指定種類以及所有更高級的種類。從高至低的值為：ERROR、WARNING 和 INFO。 預設：INFO
<code>imq.log.file.output</code>	指定寫入至日誌檔的記錄資訊的種類。允許的值為：由垂直列 () 分隔的任何一組記錄種類，或者 ALL 或 NONE。 預設：ALL
<code>imq.log.file.dirpath</code>	指定包含日誌檔之目錄的路徑 (因作業系統而異)。 預設：請參閱附錄 A 「Message Queue 資料的位置」
<code>imq.log.file.filename</code>	指定日誌檔的名稱。 預設：log.txt
<code>imq.log.file.rolloverbytes</code>	指定日誌檔輸出自動重建至新日誌檔時，此日誌檔的容量 (以位元組為單位)。值 -1 表示沒有基於檔案容量的自動重建。 預設：-1
<code>imq.log.file.rolloversecs</code>	指定日誌檔輸出自動重建至新日誌檔時，此日誌檔的存在時間 (以秒為單位)。值 -1 表示沒有基於檔案存在時間的自動重建。 預設：604800 (一週)
<code>imq.log.console.output</code>	指定寫入至主控台的記錄資訊的種類。允許的值為由垂直列 () 分隔的任何一組記錄種類，或者 ALL 或 NONE。 預設：ERROR WARNING
<code>imq.log.console.stream</code>	指定主控台輸出寫入至 <code>stdout</code> (OUT)，或寫入至 <code>stderr</code> (ERR)。 預設：ERR
<code>imq.log.syslog.facility</code>	(僅適用於 Solaris) 指定 Message Queue 代理程式應記錄為的 syslog (系統日誌) 設備。值鏡射 <code>syslog(3C)</code> 線上援助頁中列出的那些值。用於 Message Queue 的適當值為：LOG_USER、LOG_DAEMON 以及 LOG_LOCAL0 至 LOG_LOCAL7。 預設：LOG_DAEMON
<code>imq.log.syslog.logpid</code>	(僅適用於 Solaris) 指定是否同時記錄訊息與代理程式 ID (true/false)。 預設：true
<code>imq.log.syslog.logconsole</code>	(僅適用於 Solaris) 指定如果訊息無法傳送至 syslog (系統日誌)，是否將它們寫入至系統主控台 (true/false)。 預設：false

表 2-9 監視服務特性 (續)

特性名稱	說明
<code>imq.log.syslog.identity</code>	(僅適用於 Solaris) 指定應前置於每個訊息的身份字串是否記錄至 <code>syslog</code> (系統日誌)。 預設： <code>imqbrokerd_</code> 以及代理程式實例名稱。
<code>imq.log.syslog.output</code>	(僅適用於 Solaris) 指定寫入至 <code>syslogd(1M)</code> 的記錄資訊的種類。允許的值為由垂直列 () 分隔的任何記錄種類，或者 <code>ALL</code> 或 <code>NONE</code> 。 預設：錯誤
<code>imq.log.timezone</code>	指定日誌時間標記的時區。這些識別碼與 <code>java.util.TimeZone.getTimeZone()</code> 使用的識別碼相同。 例如： <code>GMT</code> 、 <code>America/LosAngeles</code> 、 <code>Europe/Rome</code> 、 <code>Asia/Tokyo</code> 。 預設：本地時區
<code>imq.metrics.topic.enabled</code>	指定 (<code>true/false</code>) 是否啓用量訊息產生功能。若為 <code>false</code> ，那麼會拋出一個用戶端異常，表示嘗試指向度量主題目標。 預設： <code>true</code>
<code>imq.metrics.topic.interval</code>	指定產生傳送到度量主題目標之度量訊息的時間間隔 (以秒為單位)。 預設：60
<code>imq.metrics.topic.persist</code>	指定 (<code>true/false</code>) 度量訊息是否為永久性訊息。 預設： <code>false</code>
<code>imq.metrics.topic.timetolive</code>	指定傳送到度量主題目標之度量訊息的有效期 (以秒為單位) 預設值：300

實體目標

Message Queue 訊息傳送以兩個階段的訊息發送為前提：首先，從產生者用戶端至代理程式上的目標的訊息發送，然後，從代理程式上的目標至一個或多個使用者用戶端的訊息發送。有兩種類型的目標 (請參閱第 42 頁的「程式設計網域」)：佇列 (點對點發送模型) 和主題 (出版 / 訂閱發送模型)。這些目標表示在代理程式實體記憶體中的位置，進來的訊息在路由至使用者用戶端之前，在這些位置進行整理。

您可以使用 Message Queue 管理工具建立實體目標 (請參閱第 151 頁的「取得連接資訊」)。目標還可自動建立，如第 70 頁的「自動建立的 (對應管理員建立的) 目標」中所述。

本節描述兩種類型實體目標的特性與運作方式：佇列與主題。

佇列目標

佇列目標用於點對點訊息傳送，其中訊息表示最終僅發送至已在目標中註冊其興趣的一些使用者之一。當訊息從訊息產生者到達時，它們會形成佇列並發送到單一訊息使用者。

佇列發送到多個使用者

將佇列目標中的任何一個訊息僅發送給單一使用者時，Message Queue 允許多個使用者註冊佇列。代理程式會將進來的訊息路由到不同的使用者，以平衡負載。

實施到多個使用者的佇列發送，會根據以下的佇列目標屬性來使用一個可配置的負載平衡方法：

- `maxNumActiveConsumers`：指定在負載平衡的佇列發送中作用中使用者的數目 - 一個或多個。
- `maxNumBackupConsumers`：指定作用中使用者發生問題時，可以取代作用中使用者之備份使用者的數目 - 一個或多個。

如果使用者數目超過這兩個屬性的數目總和，將會拒絕新的使用者。(Message Queue 平台版在每個佇列最多支援 3 個使用者 - 兩個作用中使用者和一個備份使用者；Message Queue 企業版則支援無限個使用者。

負載平衡機制會考慮不同使用者的訊息使用率。它的運作方式如下：

在可配置大小的批次中，佇列目標中的訊息會路由到可用的新作用中使用者，以便使用者註冊佇列 (佇列目標的 `consumerFlowLimit` 屬性)。一旦發送這些訊息，其他到達佇列的訊息，會在使用者為可用狀態時 (即是使用者已使用一個訊息的可配置百分比，這些訊息先前已發送給使用者)，在批次中路由到使用者。每個使用者的派送比例根據使用者目前的容量和訊息處理比例而異。

訊息產生率低時，代理程式可能會不規則地派送訊息給作用中使用者。如果作用中使用者多於所需數目時，部分的使用者可能會收不到任何訊息。

如果作用中使用者發生問題，那麼第一個備份使用者會變成作用中狀態，且接手發生問題之使用者的工作。當佇列目標有一個以上的作用中使用者時，則無法保證訊息使用的順序。

在代理程式叢集環境中，可以將發送到多個使用者的功能，配置為先發送給本地使用者。佇列目標屬性 `localDeliveryPreferred` 可讓您指定只有在產生者的主代理程式沒有使用者時 - 意即產生者傳送訊息到代理程式 (本地代理程式)，才會將訊息發送給遠端使用者。在路由到遠端使用者 (透過使用者的主代理程式) 可能會降低流量速率的情況下，此配置可讓您增加路由到遠端使用者的效能。(此屬性要求目標範圍不僅限於本地發送，請參閱第 154 頁的表 6-10。)

記憶體注意事項

由於訊息可在佇列中保留較長一段時間，因此會出現記憶體資源問題。您不應為佇列配置太多的記憶體（記憶體未完全利用），也不應配置太少記憶體（訊息可能被拒絕）。考慮到基於每個佇列載入需求的靈活性，您可在建立佇列時配置實體特性：佇列中訊息的最大數目、為佇列中訊息配置的最大記憶體，以及佇列中任何訊息的最大容量（請參閱第 154 頁的表 6-10）。

主題目標

主題目標用於出版 / 訂閱訊息傳送，其中訊息表示最終發送至已在目標中註冊其興趣的所有使用者。當訊息從產生者到達時，它們會被路由至訂閱此主題的所有使用者。如果使用者已註冊此主題的長期訂閱，則他們不會在訊息發送至此主題時成為作用中使用者 - 代理程式將儲存訊息直至使用者再次成為作用中使用者，然後發送訊息。

訊息通常不會在主題目標中保留較長一段時間，因此通常不會出現大的記憶體資源問題。但是，您可以配置目標（請參閱第 154 頁的表 6-10）接收的任何訊息所允許的最大容量。

自動建立的（對應管理員建立的）目標

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

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

Message Queue 支援這種自動建立功能。啟用自動建立後，每當 MessageConsumer 或 MessageProducer 嘗試存取不存在的目標時，代理程式便會自動建立目標。（用戶端應用程式的使用者必須擁有自動建立的權限，請參閱第 197 頁的「目標自動建立存取控制」）。

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

您可以使用表 2-10 中所示的特性配置 Message Queue 訊息伺服器，以啟用或停用自動建立功能。（如需有關配置這些特性的說明，請參閱第 5 章「啟動與配置代理程式」）

表 2-10 自動建立配置特性

特性名稱	說明
<code>imq.autocreate.topic</code>	指定是否允許代理程式自動建立主題目標 (true/false)。 預設：true
<code>imq.autocreate.queue</code>	指定是否允許代理程式自動建立佇列目標 (true/false)。 預設：true
<code>imq.autocreate.destination.maxNumMsgs</code>	指定自動建立的目標中，允許給未使用訊息的最大數目。 預設：100,000
<code>imq.autocreate.destination.maxTotalMsgBytes</code>	指定目標中允許給未使用訊息的最大總記憶體容量 (以字元組為單位)。 預設：10m (百萬位元組)
<code>imq.autocreate.destination.limitBehavior</code>	指定達到記憶體限制執行緒時，代理程式確認的方式。值為： FLOW_CONTROL - 減緩產生者速度 REMOVE_OLDEST - 拋出最舊的訊息 REMOVE_LOW_PRIORITY - 根據訊息存在時間拋出最低優先權的訊息 REJECT_NEWEST - 拒絕最新的訊息 預設：REJECT_NEWEST
<code>imq.autocreate.destination.maxBytesPerMsg</code>	指定自動建立的目標中允許的任何單一訊息的最大容量 (以位元組為單位)。 預設：10k (10,240)
<code>imq.autocreate.destination.maxNumProducers</code>	指定允許給目標中產生者的最大數目。達到限制時，將無法建立新的產生者。 預設：100
<code>imq.autocreate.destination.isLocalOnly</code>	僅適用於代理程式叢集。指定在其他代理程式上不重複目標，且限制目標僅發送訊息給本地使用者 (連接到目標上所建立之代理程式的使用者)。一旦建立目標，即無法更新屬性。 預設：false
<code>imq.autocreate.queue.maxNumActiveConsumers</code>	指定最大使用者數，此數值可作用於來自自動建立之佇列目標的負載平衡發送。值 -1 表示沒有限制數目。 預設：1
<code>imq.autocreate.queue.maxNumBackupConsumers</code>	指定最大備份使用者數，如果無法從自動建立的佇列目標進行負載平衡發送，那麼這些使用者可以取代作用中的使用者。值 -1 表示沒有限制數目。 預設：0

表 2-10 自動建立配置特性 (續)

特性名稱	說明
imq.autocreate.queue. consumerFlowLimit	指定最大訊息數目，這些訊息將會發送給單一批次中的某個使用者。在負載平衡佇列發送中，負載平衡開始前，此數目是路由到作用中使用者的佇列訊息初始數目。(請參閱第 69 頁的「佇列發送至多個使用者」)。目標的使用者在其個別連接上，可配置一個較低的值來置換此限制(請參閱「Message Queue Java 用戶端開發人員指南」中連線工廠屬性的相關資訊)。值 -1 表示沒有限制數目。 預設：1000
imq.autocreate.topic. consumerFlowLimit	指定最大訊息數目，這些訊息將會發送給單一批次中的某個使用者。值 -1 表示沒有限制數目。 預設：1,000
imq.autocreate.queue. localDeliveryPreferred	僅適用於代理程式叢集中的負載平衡佇列發送。如果本地代理程式沒有使用者，則指定訊息僅發送給遠端使用者。要求自動建立的目標不僅限於本地發送(isLocalOnly = false)。 預設：false

暫存目標

暫存目標由用戶端明確建立和銷毀(使用 JMS API)，這些應用程式需要目標以接收傳送至其他用戶端的訊息回覆。這些目標由代理程式維護，僅用於為建立目標所需連接的持續時間。只要暫存目標處於使用中，它無法由管理員銷毀，也無法由用戶端應用程式銷毀：即，如果它有作用中的訊息使用者。暫存目標不同於管理員建立的或自動建立的目標(包含永久性訊息)，它們無法永久儲存，且當重新啟動代理程式時從不重新建立。但是，它們對於 Message Queue 管理工具而言為可視(請參閱第 152 頁的表 6-9)。

多重代理程式叢集 (企業版)

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

隨著連接至代理程式的用戶端數目的增加，以及要發送的訊息數目的增加，代理程式將最終超過資源限制(如檔案描述元和記憶體限制)。提供增加載入的一個方法為將多個代理程式(即多個代理程式實例)新增至 Message Queue 訊息伺服器，從而在多重代理程式間分布用戶端連接和訊息發送。

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

然而，使用代理程式叢集還有其他原因（例如，提供具有不同使用者儲存庫的工作群組，或處理防火牆限制），但故障轉移不是這些原因之一。當 Message Queue 允許使用叢集中不同的代理程式來重新建立失敗的連接時，將會遺失狀態資訊。因此，叢集中的代理程式無法用作出現故障的其他代理程式的自動備份。

換句話說，Message Queue 目前不支援高度可用性的訊息伺服器。但是，您可以使用 Sun Cluster 軟體和提供代理程式故障轉移的高度可用資料庫。您也可以設計一個訊息傳送應用程式，以使用多個代理程式來實施自訂的故障轉移解決方案。

第 127 頁的「[使用叢集 \(企業版\)](#)」中提供有關配置和管理代理程式叢集的資訊。

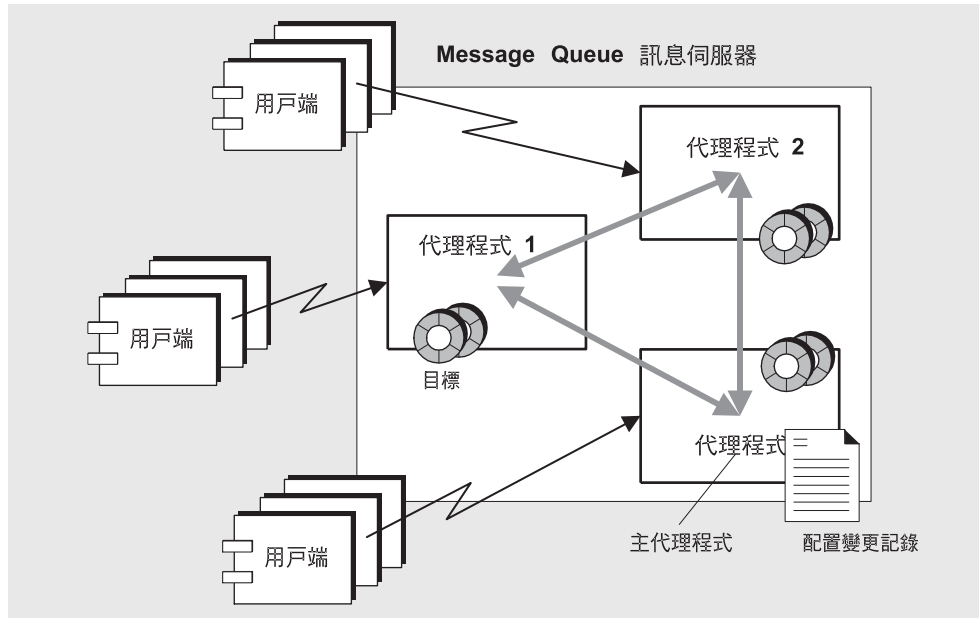
以下各節介紹 Message Queue 代理程式叢集的架構和內部功能。

多重代理程式架構

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

理論上，叢集中的代理程式可用任一隨機拓撲連接。但是，Message Queue 僅支援完全連接的叢集（即拓撲），在拓撲中每個代理程式均直接連接至叢集中所有其他代理程式，如圖 2-7 中所示。

圖 2-7 多重代理程式 (叢集) 架構



訊息發送

在多重代理程式配置中，叢集中所有代理程式上均會重複每個目標。(儘管會發生一些異常，叢集環境中的目標屬性通常會整個套用到目標的所有實例，意即套用到整個叢集，而非套用到個別的目标實例。再者，isLocalOnly 屬性設為 true 的目標在叢集裡不會被重複。如需更多資訊，請參閱目標屬性說明：[第 154 頁的表 6-10](#)。)

每個代理程式均瞭解已註冊所有其他代理程式之目標的訊息使用者。因此，每個代理程式均可將訊息從其直接連接的訊息產生者路由至遠端訊息使用者，並將訊息從遠端產生者發送至其直接連接的使用者。

在叢集配置中，與每個訊息產生者直接連接的代理程式為此產生者傳送至它的訊息執行路由。因此，永久性訊息由其主代理程式儲存和路由。

若要最小化叢集裡代理程式間的流量，訊息發送 (從目標到用戶端運行時間) 受到使用者連接上的流量控制機制所牽制。在此方法中，只有在訊息將發送給連接到目標代理程式的使用者時，才會將訊息從一個代理程式傳送到另一個代理程式，從而避免將不需要的訊息從一個代理程式傳送到另一個代理程式。同時，在某些情況下，例如到多個使用者的佇列發送情況下，您可以最小化代理程式間的流量，方法為指定優先發送到本地使用者，之後再發送到遠端使用者 (請參閱 localDeliveryPreferred 佇列目標屬性，[第 154 頁的表 6-10](#))。

某些情況要求用戶端和訊息伺服器間實現安全的加密訊息發送，此時可以對叢集配置，使叢集中代理程式間的訊息發送也實現安全性（請參閱第 130 頁的「安全代理程式互通連接」）。

叢集同步化

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

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

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

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

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

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

在任務重要的應用程式環境中，應定期備份配置變更記錄，以防止意外損毀記錄並避免主代理程式發生故障。您可以使用 `mqbrokerd` 指令的 `-backup` 選項（請參閱第 124 頁的表 5-2）執行此作業，此選項可提供建立包含配置變更記錄的備份檔案的方法。隨後，您可以使用 `-restore` 選項復原此配置變更記錄。

如有必要，您可以透過備份配置變更記錄及修改適當的叢集配置特性 (請參閱第 127 頁的表 5-3) 來變更作為主代理程式服務的代理程式，以指定新的主代理程式，並使用 `-restore` 選項來重新啟動此新的主代理程式。

使用開發環境中的叢集

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

在測試情況下，目標通常是自動建立的 (請參閱第 70 頁的「自動建立的 (對應管理員建立的) 目標」)，這些目標的長期訂閱由要進行測試的應用程式建立和銷毀。對目標和長期訂閱所作的這些變更將在叢集中傳遞。但是，如果您重新配置環境以使用主代理程式，則 Message Queue 將重新強加需求，即主代理程式處於執行狀態，以便對目標與長期訂閱進行變更，並使這些變更在叢集中傳遞。

叢集配置特性

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

叢集中的所有代理程式皆應使用一組共用的叢集配置屬性。透過將叢集配置特性放在啟動時每個代理程式均參考的一個中央叢集配置檔案中，您便可以達到此共通性。

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

如需有關叢集配置特性的更多資訊，請參閱第 127 頁的「使用叢集 (企業版)」。

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

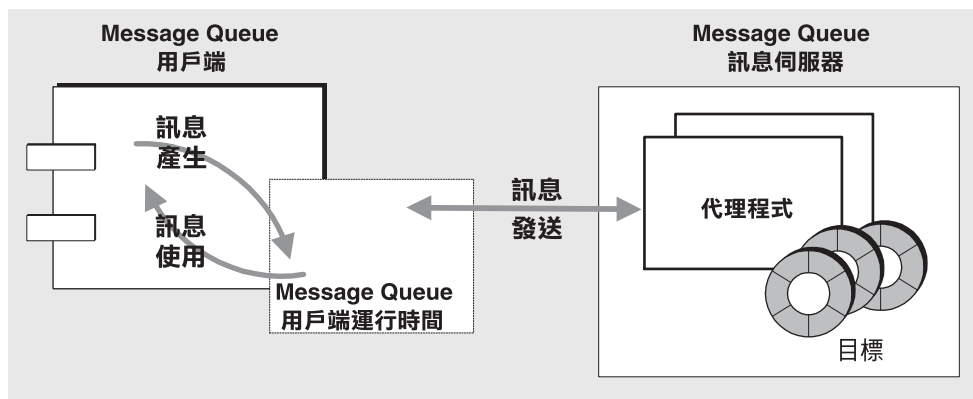
Message Queue 用戶端運行時間

Message Queue 用戶端運行時間提供到 Message Queue 訊息服務的介面給用戶端應用程式 - 它提供第 37 頁的「JMS 程式設計模型」中介紹的所有 JMS 程式設計物件給 Java 用戶端應用程式，並提供對應的 C 介面給 C 用戶端應用程式。Message Queue 用戶端運行時間支援用戶端將訊息傳送至目標，和從此類目標接收訊息所需的所有作業。

本節提供 Message Queue 用戶端運行時間如何工作的詳細描述。Message Queue Java 用戶端開發人員指南和 Message Queue C 用戶端開發人員指南中個別說明影響 Java 用戶端運行時間和 C 用戶端運行時間的用戶端應用程式設計和效能的因素。

圖 2-8 說明訊息生產與使用如何在用戶端應用程式與 Message Queue 用戶端運行時間之間進行互動，而訊息發送如何在 Message Queue 用戶端運行時間與 Message Queue 訊息伺服器之間進行互動。

圖 2-8 訊息傳送作業



訊息生產

在訊息生產中，訊息由用戶端建立，並透過連接傳送至代理程式上的目標。如果 MessageProducer 物件的訊息遞送模式已配置為永久性（可確保的遞送，一次且僅有一次），則用戶端執行緒會封鎖，直至此代理程式確認訊息已遞送至其目標並儲存在代理程式的永久性資料倉庫中。如果訊息不是永久性的，則代理程式不會傳回代理程式確認訊息（在特性名稱中稱為「Ack」），並且用戶端執行緒不會封鎖。

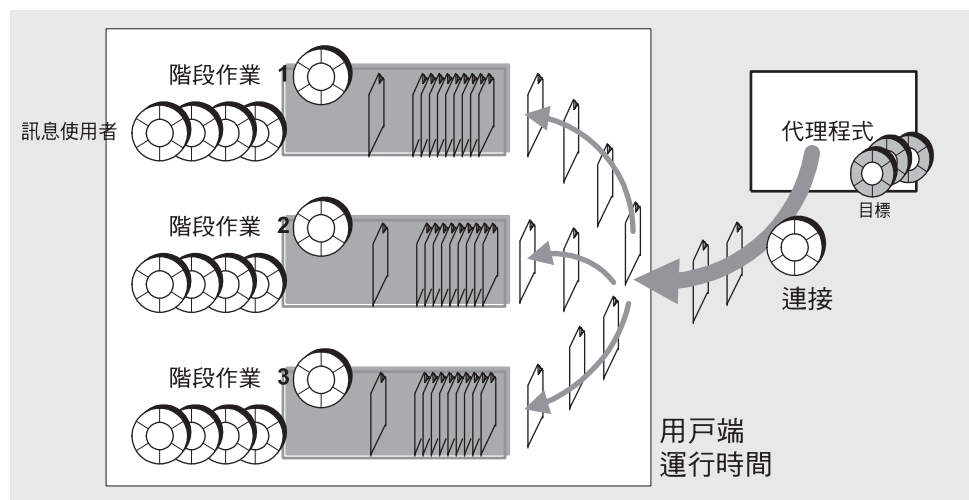
訊息使用

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

- 用戶端已為給定的目標配置使用者
- 使用者的選取條件 (如果有) 符合到達給定目標的訊息的條件
- 系統已告知連接啟動訊息發送。

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

圖 2-9 訊息發送至 Message Queue 用戶端運行時間



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

Message Queue 受管理物件

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

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

受管理物件使控制和管理 Message Queue 訊息伺服器變得非常容易：

- 透過要求用戶端應用程式存取預先配置的 ConnectionFactory 物件 (請參閱第 168 頁的「連線工廠受管理物件屬性」)，您可以控制連接的運作方式。
- 透過要求用戶端應用程式存取與現有實體目標相符的預先配置的 Destination (目標) 物件，您可以控制實體目標的激增。(您還必須停用代理程式的自動建立功能，請參閱第 70 頁的「自動建立的 (對應管理員建立的) 目標」)。
- 透過置換用戶端應用程式配置的訊息標頭值 (請參閱第 168 頁的「連線工廠受管理物件屬性」)，您可以控制 Message Queue 訊息伺服器資源。

因此，此排序可讓您 (作為 Message Queue 管理員) 控制訊息伺服器配置的詳細資訊，同時還允許用戶端應用程式獨立於供應程式：它們不必瞭解供應程式特定的語法和物件命名慣例 (請參閱第 41 頁的「JMS 提供者獨立性」)，或供應程式特定的配置特性。

您可以使用 Message Queue 管理工具來建立受管理物件，如第 7 章「管理受管理物件」中所述。建立受管理物件時，您可以將物件指定為唯讀 - 即防止用戶端應用程式變更建立物件時已配置的 Message Queue 特定的配置值。換言之，用戶端程式碼無法配置有關唯讀受管理物件的屬性值，您也無法使用用戶端應用程式啟動選項來置換這些值，如第 81 頁的「啟動用戶端時置換屬性值」中所述。

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

連線工廠受管理物件

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

如果您希望支援分散式異動 (請參閱第 44 頁的「本地異動」)，則需要使用可支援分散式異動的特殊 XAConnectionFactory 物件。

若要建立 ConnectionFactory 受管理物件，請參閱第 176 頁的「新增連線工廠」。

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

- 連接規格
- 自動重新連接運作方式
- 用戶端識別
- 訊息標頭置換
- 可靠性與流量控制
- 佇列瀏覽器運作方式
- 應用程式伺服器支援
- JMS 定義的特性支援

Message Queue Java 用戶端開發人員指南中詳細說明了這些種類中的各項目及其相應的屬性。而系統可能要求您 (作為 Message Queue 管理員) 調整這些屬性的值，通常由應用程式開發人員決定需要調整哪些屬性以調整用戶端應用程式的效能。第 168 頁的表 7-3 描述了按字母順序列出的屬性摘要。

目標受管理物件

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

若要建立 Destination (目標) 受管理物件，請參閱第 177 頁的「新增主題佇列」。

表 2-11 目標屬性

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

啟動用戶端時置換屬性值

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

若要在啟動用戶端應用程式時置換受管理物件的配置，您可以使用以下指令行語法：

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

其中，*attribute* 與第 168 頁的「[連線工廠受管理物件屬性](#)」中提供的任何 `ConnectionFactory` 受管理物件屬性相符。

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

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

Message Queue 管理工作和工具

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

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

Message Queue 管理工作

您需要執行的特定工作取決於您是處於開發環境還是生產環境。

開發環境

在開發環境中，工作重點為 Message Queue 用戶端應用程式的程式設計。Message Queue 訊息伺服器主要用於測試。在開發環境中強調靈活性，且通常您會採取以下作業：

- 使用最小管理 - 主要包含為開發人員啟動代理程式以用於測試。
- 使用資料倉庫的預設實施（內建基於檔案的持續性）、使用者儲存庫（基於檔案的使用者儲存庫）、存取控制特性檔案與物件倉庫（檔案系統倉庫）。這些預設實施通常足以用於開發測試。
- 如果您要執行多重代理程式測試，則可能不使用主代理程式。
- 您通常會使用自動建立的目標，而不明確建立目標
- 您可能需要以用戶端程式碼創設受管理物件，而非使用中央管理的受管理物件。

生產環境

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

設定作業

► 若要設定生產環境

通常，您必須至少執行以下設定作業中的一些作業（如果不是全部）：

- **管理員安全性**（受保護的管理工具使用）：
 - 確定已啟動管理連接服務（請參閱第 52 頁的表 2-3）
 - 授權：允許存取用於特定個人或管理群組的管理連接服務（請參閱第 196 頁的「連接存取控制」）
 - 如果授權給群組，則請確定管理員屬於管理群組。
 - 基於檔案的使用者儲存庫：有一個預設的 admin（管理）群組。確定管理員在 admin（管理）群組中，或在使用預設管理使用者時，變更管理密碼（請參閱第 189 頁的「變更預設管理員密碼」）。
 - LDAP 使用者儲存庫：確定管理員在 admin（管理）群組中

- **一般安全性** (請參閱第 8 章「管理安全性」):
 - 認證：將項目放入基於檔案的使用者儲存庫，或將代理程式設定為使用現有 LDAP 使用者儲存庫
(至少，您需要密碼保護管理功能。)
 - 授權：修改存取控制特性檔案中的存取配置
 - 加密：配置基於 SSL 的連接服務
- **受管理物件** (請參閱第 7 章「管理受管理物件」):
 - 配置或配置 LDAP 物件倉庫
 - 建立 ConnectionFactory 和目標受管理物件
- **代理程式叢集** (請參閱第 127 頁的「使用叢集 (企業版)」):
 - 建立中央配置檔案
 - 使用主代理程式
- **持續性**：將代理程式配置為使用外掛持續性，而不是使用內建持續性 (請參閱附錄 B「設定外掛持續性」)
- **記憶體管理**：配置目標屬性，以便訊息數目和訊息配置的記憶體容量，符合可用的代理程式記憶體資源 (請參閱第 154 頁的表 6-10)

維護作業

► 若要設定生產環境

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

- **應用程式管理**：
 - 停用代理程式的自動建立功能 (請參閱第 71 頁的表 2-10)
 - 建立作為應用程式的實體目標 (請參閱第 153 頁的「建立目標」)
 - 配置使用者對目標的存取權 (請參閱第 192 頁的「授權使用者：存取控制特性檔案」)
 - 監視與管理目標 (請參閱第 152 頁的「管理目標」)
 - 監視與管理長期訂閱 (請參閱第 161 頁的「管理長期訂閱」)
 - 監視與管理異動 (請參閱第 162 頁的「管理異動」)

- **代理程式管理與調整：**
 - 使用代理程式度量調整和重新配置代理程式 (請參閱第 9 章第 205 頁的「[分析與調整訊息服務](#)」)
 - 管理代理程式記憶體資源 (請參閱第 9 章第 205 頁的「[分析與調整訊息服務](#)」)
 - 將代理程式新增至叢集以平衡負載 (請參閱第 127 頁的「[使用叢集 \(企業版\)](#)」)
 - 回復失敗的代理程式 (請參閱第 122 頁的「[啟動代理程式](#)」)
- **管理受管理物件：**
 - 依需要建立其他 `ConnectionFactory` 和目標受管理物件 (請參閱第 176 頁的「[新增和刪除受管理物件](#)」)
 - 調整 `ConnectionFactory` 屬性值以提昇效能與流量 (請參閱第 168 頁的「[連線工廠受管理物件屬性](#)」和 第 180 頁的「[更新受管理物件](#)」)

Message Queue 管理工具

Message Queue 管理工具分為兩類：命令行公用程式與圖形使用者介面 (GUI) 管理主控台 (imqadmin)。主控台合併了兩種命令行公用程式的功能：指令公用程式 (imqcmd) 和物件管理員公用程式 (imqobjmgr)。您可以使用主控台 (和這兩種命令行公用程式) 來遠端管理代理程式以及管理 Message Queue 受管理物件。其他命令行公用程式 (imqbrokerd、imqusermgr、imqdbmgr 和 imqkeytool) 必須與它們相關的代理程式在相同主機上執行，如第 88 頁的圖 3-1 中所示。

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

管理主控台

您可以使用管理主控台執行以下作業：

- 連接至代理程式並管理它。
- 在代理程式上建立並管理實體目標
- 連接至物件倉庫
- 將受管理物件新增至物件倉庫並管理這些物件。

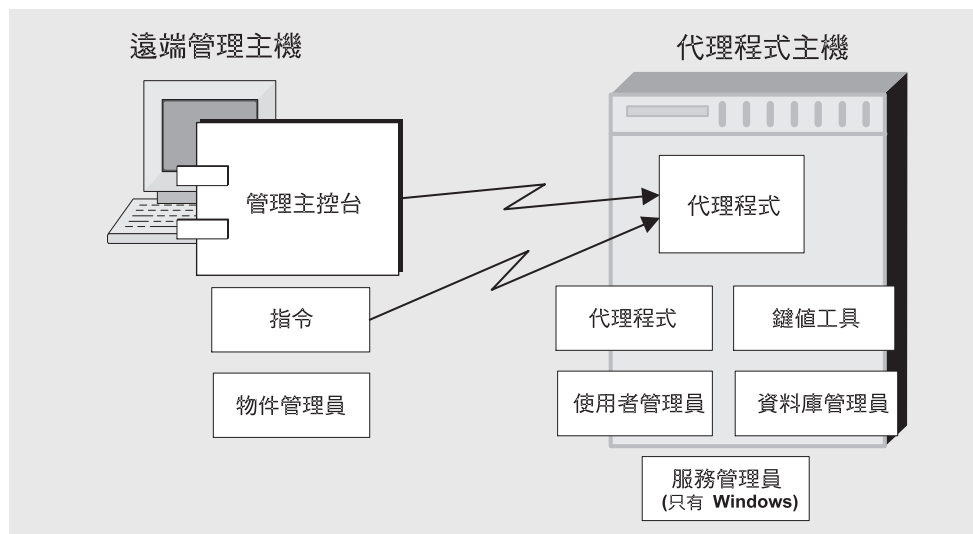
您無法使用管理主控台執行某些工作；其中主要包括啟動代理程式、建立代理程式叢集、配置代理程式和實體目標的更加專用的特性以及管理使用者資料庫。

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

指令行公用程式摘要

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

圖 3-1 本機與遠端管理公用程式



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

代理程式 (imqbrokerd) 您可以使用代理程式公用程式啟動代理程式。您可以使用 `imqbrokerd` 指令的選項指定是否應在叢集中連接代理程式，並指定其他配置資訊。`imqbrokerd` 指令在 [第 5 章「啟動與配置代理程式」](#) 進行說明。

指令 (imqcmd) 啟動代理程式後，您可以使用指令行公用程式建立、更新和刪除實體目標，控制代理程式及其連接服務，以及管理代理程式的資源。`imqcmd` 指令在 [第 6 章「代理程式和應用程式管理」](#) 進行說明。

物件管理員 (imqobjmgr) 您可以使用物件管理員公用程式在透過 JNDI 可存取的物件倉庫中新增、列示、更新和刪除受管理物件。透過將 JMS 用戶端與 JMS 供應程式特定的命名和配置格式分離，受管理物件可讓 JMS 用戶端獨立於供應程式。`imqobjmgr` 指令在 [第 7 章「管理受管理物件」](#) 進行說明。

使用者管理員 (imqusermgr) 您可以使用使用者管理員公用程式，寫入用於認證和授權使用者的基於檔案的使用者儲存庫。imqusermgr 指令在[第 8 章「管理安全性」](#)進行說明。

鍵值工具 (imqkeytool) 您可以使用鍵值工具公用程式，產生用於 SSL 認證的自身簽名憑證。imqkeytool 指令在[第 8 章「管理安全性」](#)和[附錄 C「HTTP/HTTPS 支援 \(企業版\)」](#)進行說明。

資料庫管理員 (imqdbmgr) 您可以使用資料庫管理員公用程式建立和管理用於永久性儲存的 JDBC 相容資料庫。imqdbmgr 指令在[附錄 B「設定外掛持續性」](#)進行說明

服務管理員 (imqsvcadmin) 您可以使用服務管理員公用程式安裝、查詢和移除作為 Windows 服務的代理程式。imqsvcadmin 指令在[附錄 D「將代理程式用作 Windows 服務」](#)進行說明

指令行語法

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

所有指令行公用程式共用以下指令語法：

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

Utility_Name 指定 Message Queue 公用程式的名稱，例如，imqcmd、imqobjmgr、imqusermgr 等。

必須記住以下四項重要內容：

- 在子指令 (和引數，如果公用程式接受兩種類型的運算元) 之後指定選項。
- 如果引數含有空格，請將整個引數括在引號內。通常將屬性值對括在引號內是最安全的。
- 如果您在指令行上指定 -v (版本) 或 -h/-H (說明) 選項，則系統不會執行此指令行上的其他任何選項。請參閱[第 90 頁的表 3-1](#)，以獲得共用選項的說明。
- 使用空格將子指令、引數、選項和選項引數分隔。

以下為沒有子指令子句的指令行範例。此指令啟動預設代理程式。

```
imqbrokerd
```

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

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

共用指令行選項

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

表 3-1 共用 Message Queue 指令行選項

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

管理主控台指導

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

- 啟動代理程式以及使用主控台連接至代理程式並管理它
- 在代理程式上建立實體目標
- 建立物件倉庫並使用主控台連接至此物件倉庫
- 將受管理物件新增至物件倉庫

配置所需目標與受管理物件以執行 JMS 相容的應用程式 (HelloWorldMessageJNDI，可於範例應用程式的 helloworld 子目錄 /demo 目錄中取得) 時，可參考此指導 (請參閱附錄 A 「Message Queue 資料的位置」)。在本指導的最後部分，您將執行此應用程式。

提供的本指導主要是爲了指導您使用管理主控台執行基本的管理工作。本指導不能替代整本「Message Queue Java 用戶端開發人員指南」或「*管理指南*」的其他章。

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

- 配置某些實體目標特性
某些實體目標特性無法使用管理主控台來配置。可以以第 152 頁的「[管理目標](#)」中的章節描述來配置這些代理程式特性。
- 建立代理程式叢集
請參閱第 127 頁的「[使用叢集 \(企業版\)](#)」，以獲得更多資訊。
- 管理使用者資料庫
請參閱第 184 頁的「[認證使用者](#)」，以獲得更多資訊。

準備工作

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

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

啓動管理主控台

管理主控台是您用於執行以下作業的圖形工具：

- 建立代理程式的參考並連接至代理程式
- 管理代理程式
- 在代理程式上建立實體目標，代理程式可使用這些實體目標發送訊息
- 連接至您可將 Message Queue 受管理物件放在其中的物件倉庫

受管理物件可讓您管理 JMS 相容應用程式的訊息傳送需要。如需更多資訊，請參閱第 79 頁的「Message Queue 受管理物件」。

► 若要啓動管理主控台

1. 選擇 [開始] > [程式集] > Sun Java System Message Queue 3.5 SP1 > [管理]。

可能需要等待幾秒鐘之後才會顯示主控台視窗。

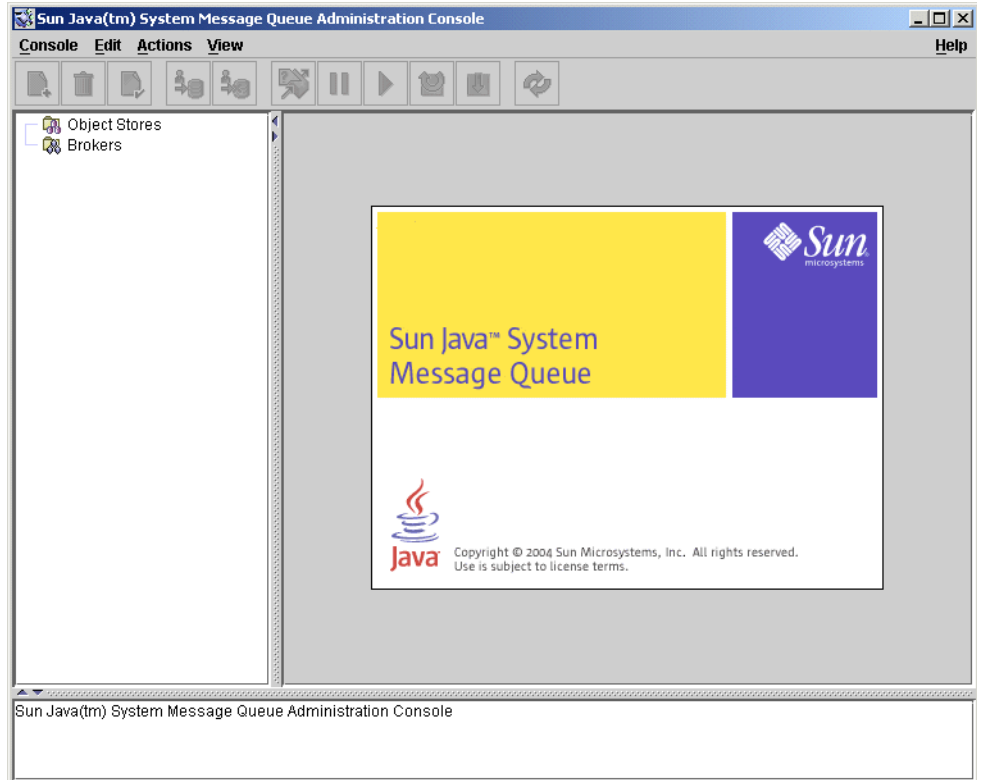
非 Windows 使用者：在命令提示字元輸入以下指令：

在 Solaris 上為 `/usr/bin/imqadmin`

在 Linux 上為 `/opt/imq/bin/imqadmin`

2. 花費幾秒鐘查看主控台視窗。

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



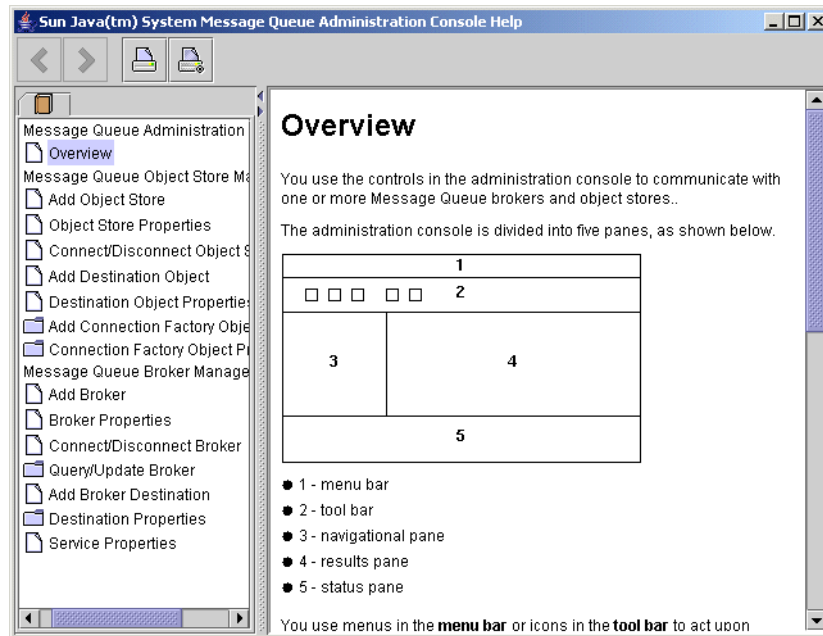
沒有指導可以提供完全的資訊，因此讓我們首先找到取得管理主控台說明資訊的方法。

取得說明

在功能表列的最右端可以找到 [Help] 功能表。

► 若要顯示管理主控台說明資訊

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



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

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

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

您的第一項主控台管理工作 (第 96 頁的「新增代理程式」) 將是建立可透過主控台管理的代理程式的參考。但是開始之前，請先檢查線上說明以獲得資訊。

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

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

4. 閱讀此說明文字。
5. 關閉 [Help] 視窗。

使用代理程式

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

使用代理程式包括以下工作：

- 啟動並配置代理程式

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

備註 您無法使用管理主控台啟動代理程式實例。

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

代理程式支援與應用程式用戶端以及管理用戶端的通訊。它透過不同的連接服務進行通訊，因此您可以將代理程式配置為執行這些服務的任何一項或者全部。如需有關連接服務的更多資訊，請參閱第 50 頁的「[連接服務](#)」。

啟動代理程式

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

▶ 若要啟動代理程式

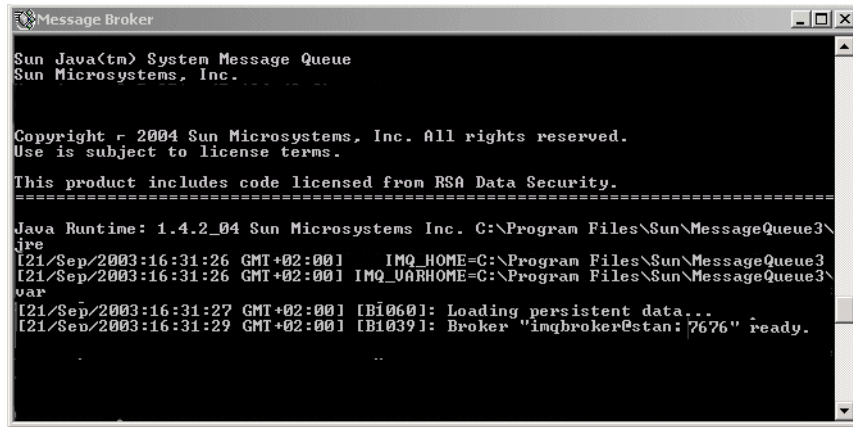
1. 選擇 [開始] > [程式集] > Sun Java System Message Queue 3.5 SP1 > [Message Broker]。

非 Windows：輸入以下指令啟動代理程式。

在 Solaris 上為 `/usr/bin/imqbrokerd`

在 Linux 上為 `/opt/imq/bin/imqbrokerd`

接著會顯示命令提示字元視窗，表示代理程式已準備就緒。



```
Message Broker
Sun Java(tm) System Message Queue
Sun Microsystems, Inc.

Copyright © 2004 Sun Microsystems, Inc. All rights reserved.
Use is subject to license terms.

This product includes code licensed from RSA Data Security.
=====
Java Runtime: 1.4.2_04 Sun Microsystems Inc. C:\Program Files\Sun\MessageQueue3\
jre
[21/Sep/2003:16:31:26 GMT+02:00] IMQ_HOME=C:\Program Files\Sun\MessageQueue3
[21/Sep/2003:16:31:26 GMT+02:00] IMQ_UARHOME=C:\Program Files\Sun\MessageQueue3\
var
[21/Sep/2003:16:31:27 GMT+02:00] [B1060]: Loading persistent data...
[21/Sep/2003:16:31:29 GMT+02:00] [B1039]: Broker "imqbroker@stan:7676" ready.
```

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

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

新增代理程式

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

► 若要將代理程式新增至管理主控台

1. 在導覽窗格中的 [Broker] 上按一下滑鼠右鍵，然後選擇 [Add Broker]。
2. 在 [Broker Label] 欄位中輸入 MyBroker。
這可提供識別管理主控台中代理程式的標籤。

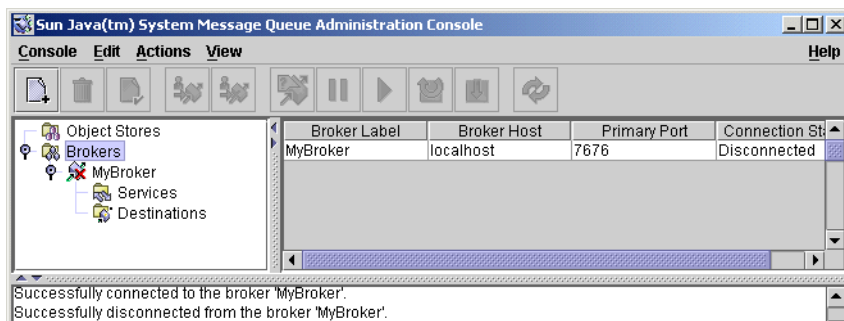


請注意對話方塊中指定的預設主機名稱 (localhost) 和主要連接埠 (7676)。稍後當您配置連線工廠 (用戶端將用來配置與此代理程式的連接) 時，您將需要指定這些值。

保留 [Password] 欄位為空白。如果您在連接時指定密碼將更安全。

3. 按一下 [OK] 以新增此代理程式。

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



4. 在 [MyBroker] 上按一下滑鼠右鍵，然後從快顯功能表中選擇 [Properties]。

畫面上將會顯示代理程式特性對話方塊。您可以使用此對話方塊更新您在新增此代理程式時指定的任何特性。

5. 按一下 [Cancel] 以關閉此對話方塊。

變更管理員密碼

當您連接至代理程式時，如果您未在新增代理程式時指定密碼，則系統會提示您提供密碼。依預設，[Administration Console] 可以用使用者 `admin`，密碼 `admin` 連接至代理程式。為了更高的安全性，最好在連接前變更預設管理員密碼 (`admin`)。

► 若要變更管理員密碼

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

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

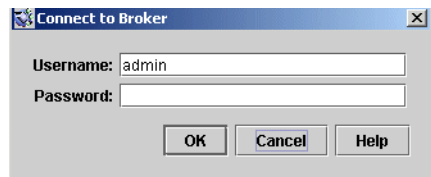
變更會立即生效。每當您使用 `Message Queue` 指令行公用程式或管理主控台之一時，您必須指定新的密碼。

雖然用戶端與管理員使用不同的連接服務，但是系統還為用戶端指定了預設使用者名稱和密碼，以便您可以測試 `Message Queue` 而不必進行大量的管理配置。依預設，用戶端可以用使用者 `guest`，密碼 `guest` 連接至代理程式。但是，您應儘快為用戶端建立安全的使用者名稱和密碼。請參閱第 184 頁的「認證使用者」，以獲得更多資訊。

連接至代理程式

► 若要連接至代理程式

1. 在 `MyBroker` 上按一下滑鼠右鍵，然後選擇 [Connect to Broker]。
畫面上將會顯示對話方塊，可讓您指定您的名稱和密碼。



2. 在 [Password] 欄位中輸入 `admin`，或輸入您在第 98 頁的「變更管理員密碼」中為密碼指定的任何值。

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

3. 按一下 [OK] 以連接至此代理程式。

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

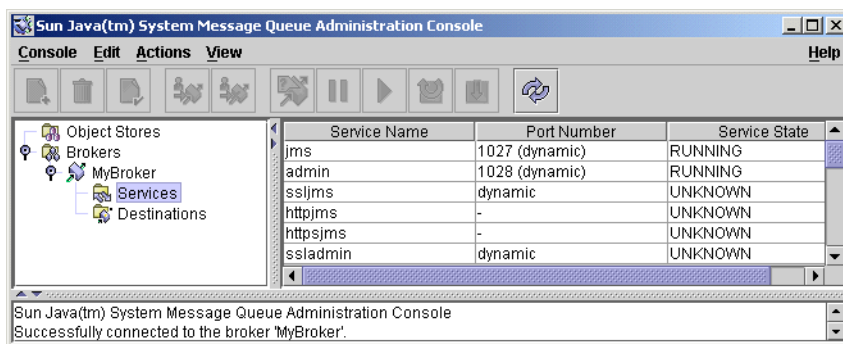
檢視連接服務

透過代理程式提供的連接服務和它支援的實體目標可以識別此代理程式。

► 若要檢視可用的連接服務

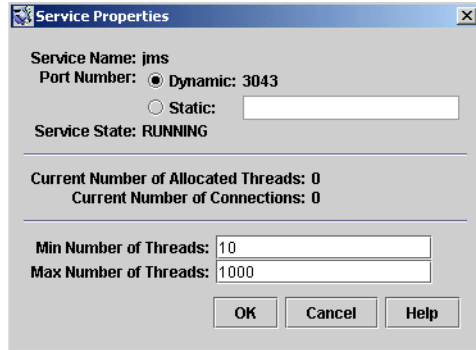
1. 在導覽窗格中選取 [Services]。

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



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

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



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

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

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

將實體目標新增至代理程式

依預設，已啟用代理程式自動建立目標的功能，以便讓代理程式動態建立實體目標。因此，在開發環境中，您不需明確建立目標來測試用戶端程式碼。

但是，在生產配置中，會建議您明確建立實體目標。這樣可讓您（管理員）充分地瞭解在代理程式上使用的目標。

透過配置 `img.autocreate.topic` 或 `img.autocreate.queue` 特性，您可以控制代理程式是否可以新增自動建立的目標。如需更多資訊，請參閱第 70 頁的「[自動建立的（對應管理員建立的）目標](#)」。

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

► 若要將佇列目標新增至代理程式

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

畫面上將會顯示以下對話方塊：

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

此時此目標將會在結果窗格中出現。

使用實體目標

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

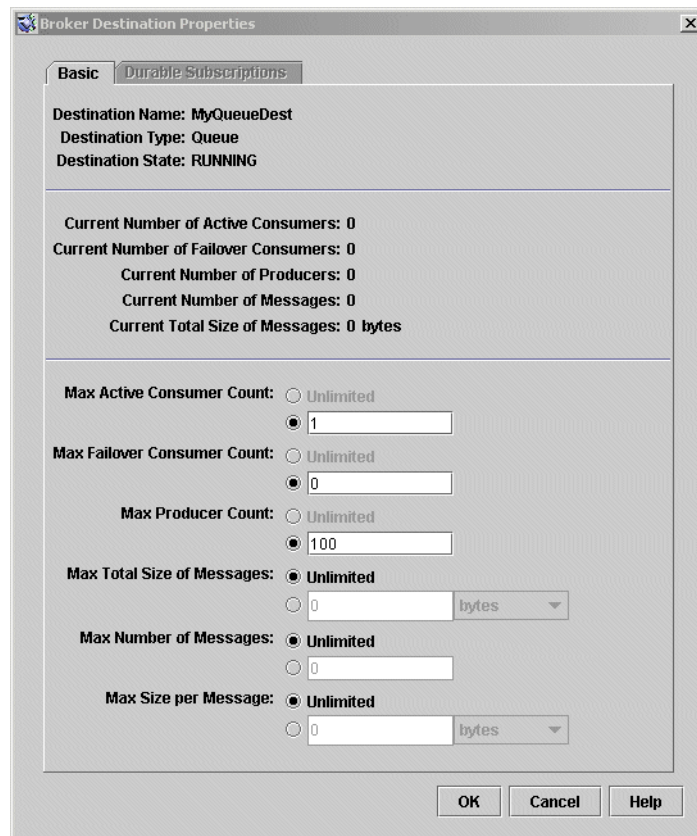
- 檢視與更新實體目標的特性

- 清除目標上的訊息
- 刪除目標

► 若要檢視實體目標的特性

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

畫面上將會顯示以下對話方塊：



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

4. 按一下 [Cancel] 以關閉此對話方塊。

► 若要從目標中清除訊息

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

畫面上將會顯示確認對話方塊。

清除訊息可移除訊息並保留空白目標。

► 若要刪除目標

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

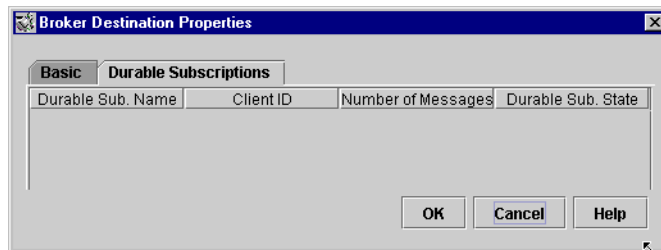
畫面上將會顯示確認對話方塊。

備註 請勿刪除 MyQueueDest 佇列目標。

刪除目標可清除此目標上的訊息並移除此目標。

取得有關主題目標的資訊

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



您可以使用此對話方塊：

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

使用物件倉庫

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

雖然可以在用戶端程式碼中創設與配置受管理物件，但是您 (管理員) 最好在物件倉庫中建立、配置並儲存這些物件，用戶端應用程式可使用 JNDI 查找程式碼存取此物件倉庫。這樣可讓用戶端程式碼獨立於供應程式。

如需有關受管理物件的更多資訊，請參閱第 79 頁的「[Message Queue 受管理物件](#)」。

您無法使用管理主控台建立物件倉庫。您必須提前進行此作業，如以下一節中所述。

新增物件倉庫

新增物件倉庫可在管理主控台中建立現有物件倉庫的參考。即使您退出並重新啟動主控台，此參考仍可保留。

► 若要新增檔案系統物件倉庫

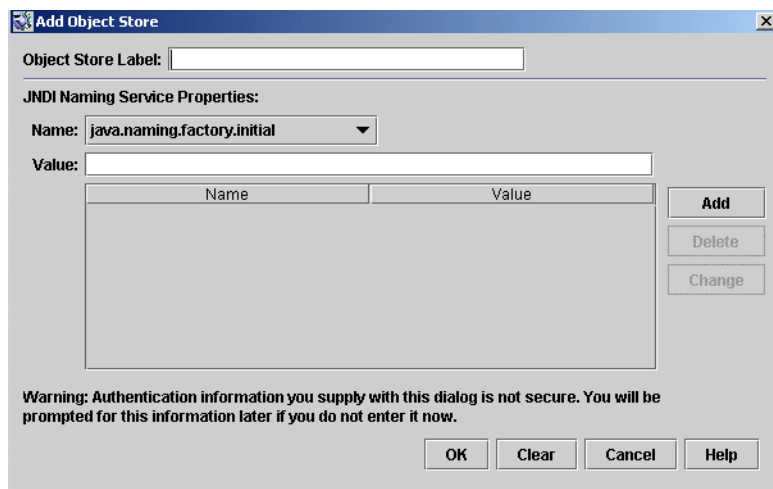
1. 如果您的 C 磁碟機上還沒有名為 Temp 的資料夾，請現在建立一個。

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

非 Windows：您可以使用 /tmp 目錄，此目錄應已存在。

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

畫面上將會顯示以下對話方塊：



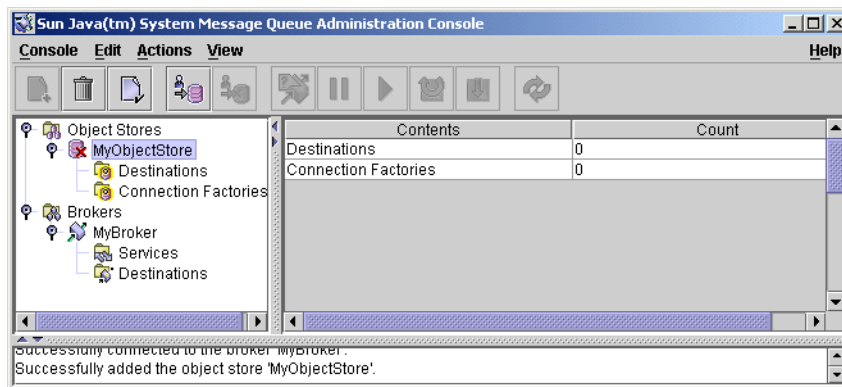
3. 在名為 [ObjectStoreLabel] 的欄位中輸入 MyObjectStore。
此項僅可為管理主控台內的物件倉庫顯示提供標籤。
在以下步驟中，您將需要輸入 JNDI 名稱 / 值對。JMS 相容應用程式將使用這些名稱 / 值對來查找受管理物件。
4. 從 [Name] 下拉式清單中選取 `java.naming.factory.initial`。
此特性可讓您指定要使用何種 JNDI 服務供應程式。例如，檔案系統服務供應程式或 LDAP 服務供應程式。
5. 在 [Value] 欄位中輸入以下內容
`com.sun.jndi.fscontext.RefFSContextFactory`
這表示您將使用檔案系統倉庫。(對於 LDAP 倉庫，您將指定 `com.sun.jndi.ldap.LdapCtxFactory`。)
在生產環境中，您可能要使用 LDAP 目錄伺服器作為物件倉庫。如需有關配置伺服器以及進行 JNDI 查找的資訊，請參閱第 166 頁的「LDAP 伺服器物件倉庫」。
6. 按一下 [Add] 按鈕。
請注意此時特性摘要窗格中列出的特性及其值。
7. 從 [Name] 下拉式清單中選取 `java.naming.provider.url`。
此特性可讓您指定物件倉庫的確切位置。對於檔案系統類型的物件倉庫，此項將為現有目錄的名稱。

8. 在 [Value] 欄位中輸入以下內容
file:///C:/Temp

(在 Solaris 和 Linux 上為 file:///tmp)
9. 按一下 [Add] 按鈕。

請注意此時屬性摘要窗格中列出的特性及其值。如果您要使用 LDAP 伺服器，您可能還必須指定認證資訊；而檔案系統倉庫則無需此項。
10. 按一下 [OK] 以新增此物件倉庫。
11. 如果尚未在導覽窗格中選取節點 [MyObjectStore]，請現在選取它。

管理主控台此時顯示如下：



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

導覽窗格中的物件倉庫圖示上畫有紅色的 X。這表示此物件倉庫已中斷連接。在使用物件倉庫之前，您將需要與其連接。

檢查物件倉庫特性

當中斷管理主控台與物件倉庫的連接時，您可以檢查與變更此物件倉庫的某些特性。

► 若要顯示物件倉庫的特性

1. 在導覽窗格中的 [MyObjectStore] 上按一下滑鼠右鍵。

2. 從快顯功能表中選擇 [Properties]。

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

3. 按一下 [OK] 或 [Cancel] 以關閉此對話方塊。

連接至物件倉庫

您必須連接至物件倉庫之後才可以將物件新增至此倉庫。

▶ 若要連接至物件倉庫

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

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

新增連線工廠受管理物件

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

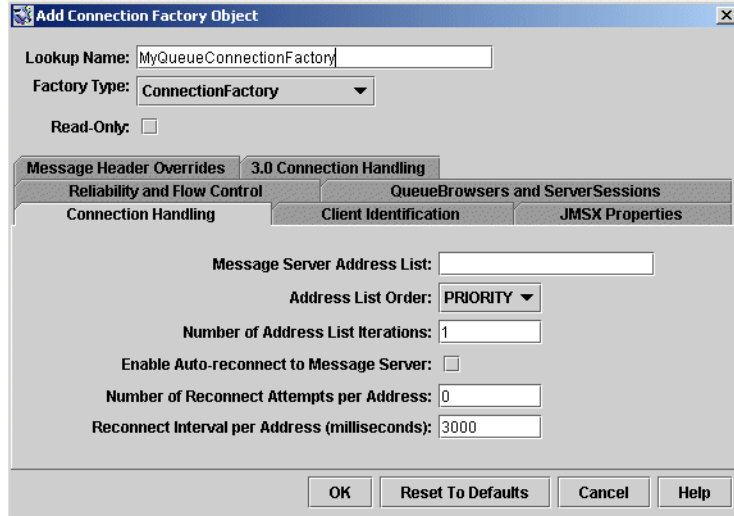
如需有關配置連線工廠的資訊，請參閱線上說明與「Message Queue Java 用戶端開發人員指南」。

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

▶ 若要將連線工廠新增至物件倉庫

1. 如果尚未連接，請連接至 MyObjectStore (請參閱第 107 頁的「[連接至物件倉庫](#)」)。
2. 在 [Connection Factories] 節點上按一下滑鼠右鍵，然後選擇 [Add Connection Factory Object]。

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



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

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

```
qcf= (javax.jms.QueueConnectionFactory)
      ctx.lookup ("MyQueueConnectionFactory")
```
4. 從下拉式清單中選取 [QueueConnectionFactory]，以指定連線工廠的類型。
5. 按一下 [Connection Handling] 標籤。
6. 您通常可以在 [Message Server Address List] 欄位輸入用戶端連接之代理程式的位址。此欄位的範例如下所示：


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

依預設，您不需輸入任何值，因爲將連接 Facotry 配置爲在連接埠 7676 上的本地主機執行以連接至代理程式，此配置爲本指導所預期的範例。
7. 按一下此對話方塊的所有標籤，以查閱您可以爲連線工廠配置的資訊種類。使用 [Add Connection Factory Object] 對話方塊右下角的 [Help] 按鈕，以獲得有關個別標籤的資訊。此時請勿變更任何預設值。
8. 按一下 [OK] 以建立此佇列連線工廠。
9. 查看結果窗格：已列出新建立的連線工廠的查找名稱與類型。

新增目標受管理物件

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

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

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

在本指導的下一部分，您將要新增與您先前新增的實體目標相對應的受管理物件。

► 若要將目標新增至物件倉庫

1. 在導覽窗格的中的 [Destination] 節點（於 [MyObjectStore] 節點下）上按一下滑鼠右鍵。
2. 選擇 [Add Destination Object]。

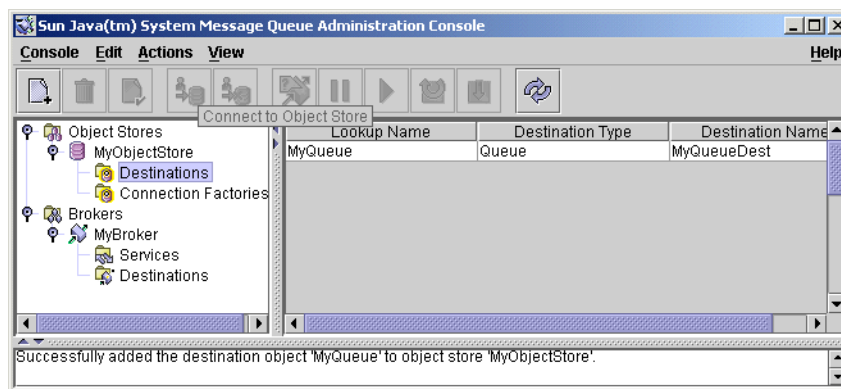
管理主控台會顯示 [Add Destination Object] 對話方塊，您可用來指定有關此物件的資訊。

3. 在 [Lookup Name] 欄位中輸入「MyQueue」。

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

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

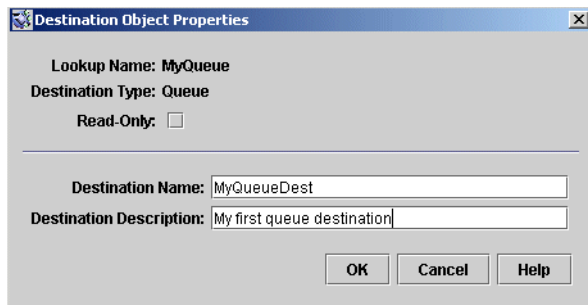
- 為 [Destination Type] 選擇 [Queue] 單選按鈕。
- 在 [Destination Name] 欄位中輸入 MyQueueDest。
此名稱為您在代理程式上新增實體目標時指定的名稱（請參閱第 100 頁的「將實體目標新增至代理程式」）。
- 按一下 [OK]。
- 在導覽窗格中選取 [Destination]，並注意有關剛新增的佇列目標受管理物件的資訊在結果窗格中的顯示方式。



受管理物件特性

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

- ▶ 若要檢視或更新目標物件的特性
 - 選取導覽窗格中 [MyObjectStore] 的 [Destination] 節點。
 - 在結果窗格中選取 [MyQueue]。
 - 選擇 [Actions] > [Properties] 以檢視 [Destination Object Properties] 對話方塊。
請注意您僅可以變更目標名稱與描述的值。若要變更查找名稱，您必須刪除此物件，然後新增具有所需查找名稱之新的佇列受管理物件。
 - 按一下 [Cancel] 以關閉此對話方塊。



更新主控台資訊

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

執行範例應用程式

提供的範例應用程式 `HelloWorldMessageJNDI`，與本指導配合使用（如需取得位置，請參閱以下的步驟 1）。它使用您在上述指導中建立的實體目標與受管理物件：命名為 `MyQueueDest` 的佇列實體目標，以及分別包含 JNDI 查找名稱 `MyQueueConnectionFactory` 和 `MyQueue` 的佇列連線工廠受管理物件，

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

► 若要執行 `HelloWorldMessageJNDI` 應用程式

1. 使包含 `HelloWorldmessageJNDI` 應用程式的目錄成為目前目錄；例如：

在 Windows 上為 `cd IMQ_HOME\demo\helloworld\helloworldmessagejndi`

在 Solaris 上為 `cd /usr/demo/imq/helloworld/helloworldmessagejndi`

在 Linux 上為 `cd /opt/imq/demo/helloworld/helloworldmessagejndi`

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

2. 配置 CLASSPATH 變數，以納入目前包含檔案 HelloWorldMessageJNDI.class 的目錄與 Message Queue 產品的 JAR 檔案，JAR 檔案如下所示：jms.jar、imq.jar 和 fscontext.jar。請參閱 Message Queue Java 用戶端開發人員指南，以獲得配置 CLASSPATH 的說明。

JNDI jar 檔案 (jndi.jar) 隨附於 JDK 1.4。如果您在使用 JDK，您則不需新增 jndi.jar 至您的 CLASSPATH 配置。如果您是使用較早版本的 JDK，您則必須將 jndi.jar 納入您的 CLASSPATH。請參閱「Message Queue Java 用戶端開發人員指南」，以取得其他資訊。

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

在 Windows 上為 `java HelloWorldMessageJNDI`

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

如果此應用程式執行成功，您應看到以下輸出：

```
java HelloWorldMessageJNDI
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

啟動與配置代理程式

安裝 Sun Java™ System Message Queue 之後，您可以使用 `imqbrokerd` 指定啟動代理程式。代理程式實例的配置由一組配置檔案和使用 `imqbrokerd` 指令傳送的選項管理，這些選項置換配置檔案中的相應特性。

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

- 編輯代理程式實例配置檔案
- 使用代理程式叢集
- 控制代理程式的記錄

如需如何啟動與使用作為 Windows 服務的代理程式的說明，請參閱第 297 頁的「將代理程式用作 Windows 服務」。

配置檔案

用來配置代理程式的已安裝代理程式配置檔案範本，位於作業系統的某個目錄中，目錄位置因作業系統而異，如附錄 A 「Message Queue 資料的位置」中所述。

此目錄儲存以下檔案：

- 啟動時載入的**預設配置檔案**。此檔案名稱為 `default.properties`，並且不可編輯。您可能需要閱讀此檔案，以確定預設值並找到您要變更之特性的確切名稱。
- 包含安裝 Message Queue 時指定的所有特性的**安裝配置檔案**。此檔案名稱為 `install.properties`；安裝之後無法編輯。

實例配置檔案

當您首次執行代理程式時，會建立實例配置檔案，您可以用來指定代理程式實例的配置特性。實例配置檔案儲存於以代理程式實例名稱 (*instanceName*) 作為辨別依據的目錄中，且代理程式實例與配置檔案相互關聯 (請參閱附錄 A 「[Message Queue 資料的位置](#)」)：

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

備註 `ö/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。之後必須以同一個使用者來啟動所有代理程式實例。

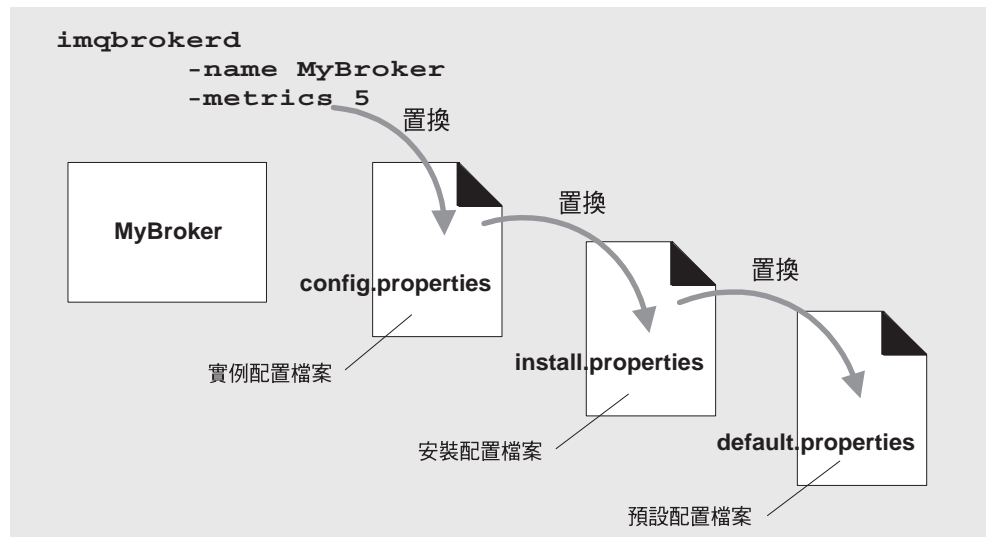
代理程式實例會維護實例配置檔案。當您使用管理工具變更配置時，即會修改實例配置檔案。您也可以以手動方式編輯實例配置檔案來變更配置 (請參閱第 117 頁的「[編輯實例配置檔案](#)」)。若要執行此動作，您必須為 `.../instances/instanceName` 目錄的擁有者，或以超級使用者身分登入，才可變更目錄上的權限。

如果您連接叢集中的代理程式實例 (請參閱第 72 頁的「[多重代理程式叢集 \(企業版\)](#)」)，您可能還需要使用 [叢集配置檔案](#) 以指定叢集配置資訊。如需更多資訊，請參閱第 127 頁的「[叢集配置特性](#)」。

合併特性值

啟動時，系統會合併不同配置檔案中的特性值。它使用在安裝和實例配置檔案中配置的值置換在預設配置檔案中指定的值。您可以透過使用 `imqbrokerd` 指令選項置換結果值。此機制在圖 5-1 中有說明。

圖 5-1 代理程式配置檔案



特性命名語法

配置檔案中的任何 Message Queue 特性定義均使用以下命名語法：

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

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

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

下列項目指定每天 (86400 秒) 會建立一個新的日誌檔：

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

第 118 頁的表 5-1 以字母順序列出代理程式配置特性 (及其預設值)。

編輯實例配置檔案

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

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

- 使用管理工具。如需有關可使用 `imqcmd` 配置之特性的資訊，請參閱第 144 頁的表 6-4。
- 當代理程式實例關閉時編輯 `config.properties` 檔案；然後重新啓動此實例。（在 Solaris 和 Linux 平台上，僅首次啓動此代理程式實例的使用者有權編輯 `config.properties` 檔案。）

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

表 5-1 代理程式實例配置特性

特性名稱	類型	預設值	參考
<code>imq.accesscontrol.enabled</code>	布林	<code>true</code>	第 62 頁的表 2-6
<code>imq.accesscontrol.file.filename</code>	字串	<code>accesscontrol.properties</code>	第 62 頁的表 2-6
<code>imq.authentication.basic.user_repository</code>	字串	<code>file</code>	第 62 頁的表 2-6
<code>imq.authentication.client.response.timeout</code>	整數 (秒)	180	第 62 頁的表 2-6
<code>imq.authentication.type</code>	字串	<code>digest</code>	第 62 頁的表 2-6
<code>imq.autocreate.destination.isLocalOnly</code>	布林	<code>false</code>	第 71 頁的表 2-10
<code>imq.autocreate.destination.limitBehavior</code>	字串	<code>REJECT_NEWEST</code>	第 71 頁的表 2-10
<code>imq.autocreate.destination.maxBytesPerMsg</code>	位元組字串 ¹	10k	第 71 頁的表 2-10
<code>imq.autocreate.destination.maxNumMsgs</code>	整數	100,000	第 71 頁的表 2-10
<code>imq.autocreate.destination.maxNumProducers</code>	整數	100	第 71 頁的表 2-10
<code>imq.autocreate.destination.maxTotalMsgBytes</code>	位元組字串 ¹	10m	第 71 頁的表 2-10
<code>imq.autocreate.queue</code>	布林	<code>true</code>	第 71 頁的表 2-10
<code>imq.autocreate.queue.consumerFlowLimit</code>	整數	1000	第 71 頁的表 2-10

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

表 5-1 代理程式實例配置特性 (續)

特性名稱	類型	預設值	參考
imq.autocreate.queue.localDeliveryPreferred	布林	false	第 71 頁的表 2-10
imq.autocreate.queue.maxNumActiveConsumers	整數	1	第 71 頁的表 2-10
imq.autocreate.queue.maxNumBackupConsumers	整數	0	第 71 頁的表 2-10
imq.autocreate.topic	布林	true	第 71 頁的表 2-10
imq.autocreate.topic.consumerFlowLimit	整數	1,000	第 71 頁的表 2-10
imq.cluster. <i>property_name</i>			第 127 頁的表 5-3
imq.hostname	字串	所有可用 IP 位址	第 52 頁的表 2-3
imq.httpjms.http. <i>property_name</i>			第 276 頁的表 C-1
imq.httpsjms.https. <i>property_name</i>			第 287 頁的表 C-3
imq.keystore. <i>property_name</i>			第 200 頁的表 8-8
imq.log.console.output	字串	ERROR WARNING	第 66 頁的表 2-9
imq.log.console.stream	字串	ERR	第 66 頁的表 2-9
imq.log.file.dirpath	字串	請參閱附錄 A 「Message Queue 資料的位置」	第 66 頁的表 2-9
imq.log.file.filename	字串	log.txt	第 66 頁的表 2-9
imq.log.file.output	字串	ALL	第 66 頁的表 2-9
imq.log.file.rolloverbytes	整數 (位元組)	-1 (無自動重建)	第 66 頁的表 2-9
imq.log.file.rolloversecs	整數 (秒)	604800	第 66 頁的表 2-9
imq.log.level	字串	INFO	第 66 頁的表 2-9
imq.log.syslog.facility	字串	LOG_DAEMON	第 66 頁的表 2-9
imq.log.syslog.identity	字串	imqbrokerd_\${imq.instanceName}	第 66 頁的表 2-9
imq.log.syslog.logconsole	布林	false	第 66 頁的表 2-9
imq.log.syslog.logpid	布林	true	第 66 頁的表 2-9

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

表 5-1 代理程式實例配置特性 (續)

特性名稱	類型	預設值	參考
imq.log.syslog.output	字串	ERROR	第 66 頁的表 2-9
imq.log.timezone	字串	本地時區	第 66 頁的表 2-9
imq.message.expiration.interval	整數 (秒)	60	第 56 頁的表 2-4
imq.message.max_size	位元組字串 ¹	70m	第 56 頁的表 2-4
imq.metrics.enabled	布林	true	第 66 頁的表 2-9
imq.metrics.interval	整數 (秒)	-1 (永不)	第 66 頁的表 2-9
imq.metrics.topic.enabled	布林	true	第 66 頁的表 2-9
imq.metrics.topic.interval	整數 (秒)	60	第 66 頁的表 2-9
imq.metrics.topic.persist	布林	false	第 66 頁的表 2-9
imq.metrics.topic.timetolive	整數 (秒)	300	第 66 頁的表 2-9
imq.passfile.dirpath	字串	請參閱附錄 A 「Message Queue 資料的位置」	第 62 頁的表 2-6
imq.passfile.enabled	布林	false	第 62 頁的表 2-6
imq.passfile.name	字串	密碼檔案	第 62 頁的表 2-6
imq.persist.file.destination.message.filepool.limit	整數	100	第 59 頁的表 2-5
imq.persist.file.message.cleanup	布林	false	第 59 頁的表 2-5
imq.persist.file.message.filepool.cleanratio	整數	0	第 59 頁的表 2-5
imq.persist.file.message.max_record_size	位元組字串 ¹	1m	第 59 頁的表 2-5
imq.persist.file.sync.enabled	布林	false	第 59 頁的表 2-5
imq.persist.jdbc.property_name			第 266 頁的表 B-1
imq.persist.store	字串	file	第 59 頁的表 2-5

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

表 5-1 代理程式實例配置特性 (續)

特性名稱	類型	預設值	參考
imq.ping.interval	整數	120	第 52 頁的表 2-3
imq.portmapper.backlog	整數	50	第 52 頁的表 2-3
imq.portmapper.hostname	字串	繼承自 imq.hostname	第 52 頁的表 2-3
imq.portmapper.port	整數	7676	第 52 頁的表 2-3
imq.resource_state.count	整數 (百分比)	5000 (green) 500 (yellow) 50 (orange) 0 (red)	第 56 頁的表 2-4
imq.resource_state.threshold	整數 (百分比)	0 (green) 80 (yellow) 90 (orange) 98 (red)	第 56 頁的表 2-4
imq.service.activelist	清單	jms, admin	第 52 頁的表 2-3
imq.service_name.accesscontrol.enabled	布林	繼承系統範圍的特性的值	第 62 頁的表 2-6
imq.service_name.accesscontrol.file.filename	字串	繼承系統範圍的特性的值	第 62 頁的表 2-6
imq.service_name.authentication.type	字串	繼承系統範圍的特性的值	第 62 頁的表 2-6
imq.service_name.max_threads	整數	1000 (jms) 500 (ssljms) 500 (httpjms) 500 (httpsjms) 10 (admin) 10 (ssladmin)	第 52 頁的表 2-3
imq.service_name.min_threads	整數	10 (jms) 10 (ssljms) 10 (httpjms) 10 (httpsjms) 4 (admin) 4 (ssladmin)	第 52 頁的表 2-3
imq.service_name.protocol_type.hostname	字串	繼承自 imq.hostname	第 52 頁的表 2-3
imq.service_name.protocol_type.port	整數	0 (動態配置)	第 52 頁的表 2-3

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

表 5-1 代理程式實例配置特性 (續)

特性名稱	類型	預設值	參考
imq.service_name. threadpool_model	字串	dedicated (jms) dedicated (ssljms) dedicated (httpjms) dedicated (httpsjms) dedicated (admin) dedicated (ssladmin)	第 52 頁的表 2-3
imq.shared. connectionMonitor_limit	整數	512 (Solaris & Linux) 64 (Windows)	第 52 頁的表 2-3
imq.system.max_count	整數， 0 (無限制)	-1	第 56 頁的表 2-4
imq.system.max_size	位元組字串 1， 0 (無限制)	-1	第 56 頁的表 2-4
imq.transaction.autorollback	布林	false	第 56 頁的表 2-4
imq.user_repository.ldap. property_name			第 191 頁的表 8-5

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

啟動代理程式

啟動代理程式實例可使用 imqbrokerd 指令。

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

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

imqbrokerd 指令的語法

imqbrokerd 指令的語法如下 (選項和引數用空格分隔)：

```

imqbrokerd [[ -Dproperty=value]...]
  [ -backup fileName]
  [ -cluster "[broker1] [[,broker2]...]"]
  [ -dbuser userName] [ -dbpassword password]
  [ -force]
  [ -h|-help]
  [ -javahome path]
  [ -ldappassword password]
  [ -license licenseName]
  [ -loglevel level]
  [ -metrics interval]
  [ -name instanceName]
  [ -password keypassword] [ -passfile fileName]
  [ -port number]
  [ -remove instance]
  [ -reset data]
  [ -restore fileName]
  [ -shared]
  [ -silent|-s] [ -tty]
  [ -upgrade-store-nobackup]
  [ -version]
  [ -vmargs arg1 [[arg2]Ö]

```

備註 在 Solaris 上，您可以將代理程式配置為異常結束後自動重新啓動，方法為透過將 /etc/imq/imqborkerd.conf 配置檔案中的 RESTART 特性配置為 YES。

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

啓動範例

以下範例說明 imqbrokerd 指令的使用：如需更多 imqbrokerd 指令行選項的資訊，請參閱第 124 頁的表 5-2。

- 若要啓動預設代理程式名稱和配置的代理程式實例
請使用以下指令：

```
imqbrokerd
```

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

► 若要啟動使用試用企業版授權的代理程式實例

如果您有平台版的授權，但想要試用為期 90 天的企業版功能，您可以使用 `-license` 指令行選項，並將「`try`」作為要使用的授權傳送，以啟動試用企業版授權：

```
imqbrokerd -license try
```

每次啟動代理程式實例時都必須使用此選項，否則它會預設回基本的平台版授權。

► 若要啟動包含外掛永久性的已命名代理程式實例

若要啟動使用外掛資料倉庫的 `myBroker` 代理程式 (請參閱附錄 B 第 263 頁的「設定外掛持續性」)，且要求輸入使用者和密碼，請使用以下指令：

```
imqbrokerd -name myBroker -dbuser myName -dbpassword myPassword
```

imqbrokerd 選項的摘要

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

表 5-2 imqbrokerd 選項

選項	影響的特性	說明
<code>-backup fileName</code>	無影響的特性。	僅適用於代理程式叢集。將主代理程式的配置變更記錄備份至指定檔案。請參閱第 132 頁的「備份配置變更記錄」。
<code>-cluster" [broker1] [[,broker2] ..."</code> 其中 <i>broker</i> 為以下之一	將 <code>imq.cluster.brokerlist</code> 配置為與其連接的代理程式清單。	僅適用於代理程式叢集。連接至指定主機和連接埠上的所有代理程式。此清單已與 <code>imq.cluster.brokerlist</code> 特性中的清單合併。如果您沒有為 <i>主機</i> 指定值，則使用 <code>localhost</code> 。如果您沒有為 <i>連接埠</i> 指定值，則使用值 <code>7676</code> 。請參閱第 127 頁的「使用叢集 (企業版)」，以獲得有關如何使用此選項連接多重代理程式的更多資訊。
• <code>host[:port]</code> • <code>[host]:port</code>		
<code>-dbpassword password</code>	將 <code>imq.persist.jdbc.password</code> 配置為指定的密碼	為外掛 JDBCS 相容資料倉庫指定密碼。請參閱附錄 B「設定外掛持續性」
<code>-dbuser userName</code>	將 <code>imq.persist.jdbc.user</code> 配置為指定的使用者名稱	為外掛 JDBCS 相容資料庫指定使用者名稱。請參閱附錄 B「設定外掛持續性」

表 5-2 imqbrokerd 選項 (續)

選項	影響的特性	說明
-Dproperty=value	配置系統特性。置換實例配置檔案中的相應特性值。	將指定的特性配置為指定的值。請參閱第 118 頁的表 5-1，以獲得代理程式配置特性。 警告： 請仔細檢查使用 D 選項配置的特性的拼寫與格式。如果您傳送了錯誤的值，系統將不警告您，並且 Message Queue 亦無法配置這些值。
-force	無影響的特性。	未經使用者確認即執行動作。此選項僅套用通常要求確認的 -remove instance 和 -upgrade-store-nobackup 選項。
-h -help	無影響的特性。	顯示說明。不執行指令行上的其他任何指令。
-javahome path	無影響的特性。	指定替代 Java 2 相容 JDK 的路徑。預設為使用隨附的運行時間。
-ldappassword password	將 imq.user_repository.ldap.password 配置為指定的密碼	為存取 LDAP 使用者儲存庫指定密碼。請參閱第 190 頁的「將 LDAP 伺服器用於使用者儲存庫」。
-license [licenseName]	無影響的特性。	如果要載入的授權與 Message Queue 產品版本預設的授權不同，則指定要載入的授權。如果您尚未指定授權名稱，則此項會列出系統上安裝的所有授權。根據已安裝的 Message Queue 版本，licenseName 值為 pe (平台版 - 基本功能)、try (平台版 - 90 天試用企業功能) 和 unl (企業版)。請參閱第 33 頁的「產品版本」。
-loglevel level	將 imq.broker.log.level 配置為指定的級別。	將記錄級別指定為以下之一：NONE、ERROR、WARNING 或 INFO。預設值為 INFO。如需更多資訊，請參閱第 64 頁的「記錄程式」。
-metrics interval	Sets imq.metrics.interval 配置為指定的秒數。	指定寫入記錄程式的代理程式度量的指定間隔時間 (以秒為單位)。
-name instanceName	將 imq.instanceName 配置為指定的名稱。	指定此代理程式的實例名稱，並使用相應的實例配置檔案。如果您沒有指定代理程式名稱，則將實例名稱配置為 imqbroker。 注意： 如果您在同一主機上執行代理程式的多個實例，則每個實例必須具有唯一的名稱。
-passfile fileName	將 imq.passfile.enabled 配置為 true。將 imq.passfile.dirpath 配置為包含此檔案的路徑。 將 imq.passfile.name 配置為此檔案的名稱。	指定檔案的名稱，您可從此檔案讀取 SSL 密鑰儲存、LDAP 使用者儲存庫或 JDBC 相容資料庫的密碼。如需更多資訊，請參閱第 204 頁的「使用密碼檔案」。
-password keypassword	將 imq.keystore.password 配置為指定的密碼。	為 SSL 憑證密鑰儲存指定密碼。如需更多資訊，請參閱第 60 頁的「安全性管理程式」。

表 5-2 imqbrokerd 選項 (續)

選項	影響的特性	說明
-port <i>number</i>	將 <code>imq.portmapper.port</code> 配置為指定的號碼。	指定代理程式的連接埠對映器連接埠號。依預設，此項配置為 7676 。若要在同一伺服器上執行代理程式的兩個實例，則每個代理程式的連接埠對映器必須有埠同的連接埠號。 Message Queue 用戶端會使用此連接埠號連接至代理程式實例。
-remove instance	無影響的特性。	導致移除代理程式實例：刪除此實例配置檔案、日誌檔、永久性倉庫以及與此實例相關聯的其他檔案和目錄。要求使用者確認，除非也指定了 <code>-force</code> 選項。
-reset store messages durables props	無影響的特性。	<p>依據給定的引數，重設資料倉庫 (或資料倉庫的子集) 或代理程式實例的配置特性。</p> <p>重設資料倉庫，清除所有永久性資料，包括永久性訊息、長期訂閱和異動資訊。此項可讓您啟動沒有記錄的代理程式實例。您還可以僅清除所有永久性訊息或僅清除所有長期訂閱。(如果您不想在後續的重新啟動中重設永久性倉庫，請重新啟動未使用 <code>-reset</code> 選項的代理程式實例。) 如需更多資訊，請參閱第 57 頁的「持續性管理程式」。</p> <p>重設代理程式的特性，用空白檔案取代現有實例配置檔案 (<code>config.properties</code>)：所有特性均假設為預設值。</p>
-restore <i>fileName</i>	無影響的特性。	僅適用於代理程式叢集。用指定的備份檔案取代主代理程式的配置變更記錄。此檔案必須已使用 <code>-backup</code> 選項事先建立。請參閱第 132 頁的「復原配置變更記錄」。
-shared	將 <code>imq.jms.threadpool_model</code> 配置為 <code>shared</code> 。	指定要使用共用執行緒儲存區模型實施的 jms 連接服務，在此模型中，執行緒在各連接之間共用以增加代理程式實例支援的連接數目。如需更多資訊，請參閱第 50 頁的「連接服務」。
-silent -s	將 <code>imq.log.console.output</code> 配置為 <code>NONE</code> 。	關閉對主控台的記錄。
-tty	將 <code>imq.log.console.output</code> 配置為 <code>ALL</code> 。	指定所有訊息均在主控台顯示。依預設，僅顯示 <code>WARNING</code> 和 <code>ERROR</code> 級別的訊息。
-upgrade-store- nobackup	無影響的特性	指定從不相容的版本升級到 Message Queue 3.5 或 Message Queue 3.5 SPx 時會自動移除舊的資料倉庫。如需其他詳細資訊，請參閱「 Message Queue 安裝指南 」。
-version	無影響的特性。	顯示安裝產品的版本編號。

表 5-2 imqbrokerd 選項 (續)

選項	影響的特性	說明
-vmargs <i>arg1</i> [<i>arg2</i>]...	無影響的特性	指定要傳送至 Java VM 的引數。用空格分隔各引數。如果您要傳送多個引數，或者如果引數包含空格，請使用括起的引號。例如： imqbrokerd -tty -vmargs "-Xmx128m -Xincgc"

使用叢集 (企業版)

本節描述用於配置多重代理程式叢集的特性以及連接代理程式的兩種方法，並介紹如何管理叢集。如需叢集的介紹，請參閱第 72 頁的「[多重代理程式叢集 \(企業版\)](#)」。

使用叢集時，請確保叢集中所有代理程式的主機時鐘同步 (請參閱第 301 頁的「[系統時鐘設定](#)」)。

叢集配置特性

當您將代理程式連接至叢集時，所有連接的代理程式必須指定為叢集配置特性的配置。這些特性描述了叢集中代理程式的參與情況。[表 5-3](#) 總結了與叢集相關的配置特性。以星號 (*) 標記的特性必須跟叢集中的所有代理程式擁有一樣的值。

表 5-3 叢集配置特性

特性名稱	說明
imq.cluster.brokerlist*	指定叢集中的所有代理程式。由以逗號分隔的主機: 連接埠項目清單組成，其中主機為每個代理程式的主機名稱，連接埠為它的連接埠對映器連接埠號。例如： host1:3000、host2:8000、ctrhost
imq.cluster.masterbroker*	指定叢集中的哪個代理程式 (如果有) 為記錄狀態變更的主代理程式。特性由 <i>host:port</i> 組成，其中 <i>host</i> 為主代理程式的主機名稱， <i>port</i> 為它的連接埠對映器連接埠號。配置生產環境的內容。例如，ctrhost:7676
imq.cluster.url*	指定叢集配置檔案的位置。用於代理程式參考中央叢集配置檔案，而不是個別配置。由 URL 字串組成：如果保留在 Web 伺服器上，則可使用一般的 <code>http:URL</code> 存取它。如果保留在共用磁碟機上，則可使用 <code>file:URL</code> 存取它。 例如： <code>http://webserver/imq/cluster.properties</code> <code>file:/net/mfsserver/imq/cluster.properties</code>

表 5-3 叢集配置特性 (續)

特性名稱	說明
<code>imq.cluster.port</code>	對於叢集中的每個代理程式，均可用於為 <code>cluster</code> (叢集) 連接服務指定連接埠號。 <code>cluster</code> (叢集) 連接服務用於在叢集的各代理程式之間進行內部通訊。 預設值：0 (已動態配置連接埠)
<code>imq.cluster.hostname</code>	對於叢集中的每個代理程式，如果有多台主機可用 (例如，如果一台電腦中有多個網路介面卡)，則可用於指定 <code>cluster</code> (叢集) 連接服務所連結的主機 (主機名稱或 IP 位址)。 <code>cluster</code> (叢集) 連接服務用於在叢集的各代理程式之間進行內部通訊。 預設值：繼承 <code>imq.hostname</code> 的值 (請參閱第 52 頁的表 2-3)
<code>imq.cluster.transport*</code>	指定 <code>cluster</code> (叢集) 連接服務用於叢集中，代理程式間內部通訊的網路傳輸。為顧及代理程式間加密訊息發送的安全，請針對叢集中的所有代理程式，將此特性配置為 <code>ssl</code> 。預設值： <code>tcp</code>

您可以使用以下兩種方法之一配置叢集特性：

- 您可以在每個代理程式的實例配置檔案中 (或在啟動每個代理程式的指令行中)，配置與叢集相關的配置特性。例如，若要連接代理程式 A (在 `host1` 上，連接埠為 7676)、代理程式 B (在 `host2` 上，連接埠為 5000) 和代理程式 C (在 `ctrlhost` 上，連接埠為 7676)，則用於代理程式 A、B 和 C 的實例配置檔案需要配置以下特性。

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

如果您決定變更叢集配置，則此方法要求更新所有代理程式中與叢集相關的特性。

- 您可以在一個中央叢集配置檔案中配置叢集配置特性。這些特性可能包括連接的代理程式清單 (`imq.cluster.brokerlist`)、用於叢集連接服務的網路傳輸 (`imq.cluster.transport`)，以及 (選擇性) 主代理程式的位址 (`imq.cluster.masterbroker`)。

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

以下程式碼範例顯示叢集配置檔案的內容。`host1` 與 `ctrlhost` 均正在預設連接埠上執行。這些特性指定在叢集中連接到 `host1`、`host2` 和 `ctrlhost`，且 `ctrlhost` 為主代理程式。


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

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

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

連接代理程式

本節描述如何將代理程式連接到叢集，以及如何配置叢集的安全性，即配置叢集中代理程式間的加密訊息發送。

連接方法

有兩種通用方法可將代理程式連接到叢集：使用或不使用叢集配置檔案進行連接。

無論您使用哪種方法，您啟動的每個代理程式每 5 秒鐘會嘗試連接至其他代理程式；一旦啟動此叢集中的主代理程式，嘗試即可成功。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，從而拒絕用戶端連接。當主代理程式啟動時，暫停的代理程式的全部功能將自動可用。

方法 1：不使用叢集配置檔案進行連接

► 將代理程式連接到叢集

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

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

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

方法 2：使用叢集配置檔案進行連接

還可以建立叢集配置檔案，此檔案指定要連接的代理程式清單 (和選擇性地指定主代理程式的位址)。這種定義叢集的方法更適合用於生產系統。如果您要使用此方法，叢集中的每個代理程式均必須將 `imq.cluster.url` 特性的值配置為指向此叢集配置檔案。

安全代理程式互連連接

要求叢集中代理程式間的加密訊息發送的安全性時，您必須配置 `cluster` (叢集) 連接服務，以使用基於 SSL 的傳輸協定，方法如下所述：

► 若要在叢集內配置安全連接

1. 針對叢集中的每個代理程式，配置基於 SSL 的連接服務。

請參閱第 198 頁的「透過 TCP/IP 設定基於 SSL 的服務」中的說明。

2. 將 `imq.cluster.transport` 叢集配置特性配置為 `ssl`。

如果您沒有使用叢集配置檔案，那麼您必須在叢集中的每個代理程式配置此特性。

管理叢集中的代理程式

一旦您已配置代理程式叢集，您可能需要新增新的代理程式或重新啓動此叢集中已存在的代理程式，或者自叢集移除代理程式。

將代理程式新增至叢集

► 若要新增新的代理程式至現有叢集

- 如果您正在使用叢集配置檔案，那麼
 - a. 將此新代理程式新增至叢集配置檔案中的 `imq.cluster.brokerlist` 特性。
 - b. 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 `imq.cluster.brokerlist` 特性，並確保此叢集中代理程式的所有永久性資訊均為最新的。

- c. 啟動新的代理程式，並使用 `-D` 選項在指令行上指定 `imq.cluster.url` 特性。

這會將代理程式指向叢集配置檔案。

- 如果您沒有使用叢集配置檔案，那麼在啟動新的代理程式時，請使用 `-D` 選項在指令行上指定 `imq.cluster.brokerlist`、`imq.cluster.transport` (使用安全叢集連接服務時) 和 (如有必要) `imq.cluster.masterbroker` 特性。

重新啟動叢集中的代理程式

如果叢集中的代理程式已毀損或因某種原因而關閉，那麼您必須將它重新啟動為叢集的一員。

► 若要重新啟動已為現有叢集一員的代理程式

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

從叢集中移除代理程式

► 若要從現有叢集移除代理程式

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

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

然而，您需要使用以下指令行重新啟動所有其他代理程式：

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

然後，您需要啟動代理程式 A 而不指定 `-cluster` 選項。

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

管理主代理程式的配置變更記錄

每個叢集可以具有一個主代理程式，以記錄此叢集之永久性狀態的所有變更。狀態包括長期訂閱和管理員建立之實體目標的資訊。所有代理程式在啟動期間會參考主代理程式 (或參考其配置變更記錄)，以使有關這些永久性物件的資訊同步。因此，主代理程式發生故障可能會導致無法同步化。所以，如果主代理程式發生故障，您則無法建立或刪除實體目標或長期訂閱。

因為它包含了重要資訊，所以定期備份主代理程式的配置變更記錄，以及在發生故障的情況下復原主代理程式是非常重要的。

以下各節說明如何備份和復原配置變更記錄。

備份配置變更記錄

► 若要備份配置變更記錄

使用 `imqbrokerd` 指令的 `-backup` 選項。例如，

```
imqbrokerd -backup mybackuplog
```

適時地進行備份很重要。復原很舊的備份可能會導致資訊遺失：任何從上次備份後實體目標或長期訂閱中的變更皆會遺失。

復原配置變更記錄

► 在發生故障的情況下復原主代理程式

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

```
imqbrokerd -restore mybackuplog
```

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

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

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

記錄

本節描述代理程式的預設記錄配置，並介紹如何變更此配置，以便將日誌資訊重新導向至替代輸出通道並變更日誌檔自動重建條件。如需記錄的介紹，請參閱第 64 頁的「記錄程式」。如需使用記錄以記錄代理程式度量的資訊，請參閱第 220 頁的「監視工具」。

預設記錄配置

啓動代理程式時，會自動配置為將日誌輸出儲存到一組自動重建的日誌檔中，這些日誌檔位於以代理程式實例名稱 (*instanceName*) 作為辨別的目錄中，且代理程式實例與日誌檔相互關聯 (請參閱附錄 A 「Message Queue 資料的位置」)：

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

日誌檔為簡單的文字檔案。它們命名如下，從最舊的到最新的：

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

依預設，日誌檔每週自動重建一次；系統保留九個備份檔案。

- 若要變更保留日誌檔的目錄，請將特性 `imq.log.file.dirpath` 配置為所需路徑。
- 若要將日誌檔的根名稱從 `log` 變更為其他名稱，請配置 `imq.log.file.filename` 特性。

代理程式支援三種日誌種類：ERROR、WARNING 和 INFO (請參閱第 64 頁的表 2-7)。配置記錄級別可收集此級別以上 (含此級別) 的訊息。預設日誌級別為 INFO。這表示依預設記錄了所有 ERROR、WARNING 和 INFO 的訊息。

日誌訊息格式

記錄的訊息包含時間標記 (若要變更時區時間標記，請參閱第 66 頁的表 2-9)、訊息程式碼和訊息本身。資訊容量因您所配置的日誌級別而有所不同。以下為 INFO 訊息的範例。

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

變更記錄程式配置

第 66 頁的表 2-9 中描述了所有記錄程式特性。

► 變更代理程式的記錄程式配置

1. 配置日誌級別。
2. 為一個或多個記錄種類配置輸出通道 (檔案、主控台或二者)。
3. 如果您將輸出記錄至檔案，請為此檔案配置自動重建條件。

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

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

表 5-4 `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>	將日誌級別配置為以下之一：ERROR、WARNING 和 INFO。
<code>-silent</code>	關閉對主控台的記錄。
<code>-tty</code>	將所有訊息傳送至主控台。依預設，僅顯示 WARNING 和 ERROR 級別的訊息。

以下各節描述如何變更預設配置以便執行以下作業：

- 變更輸出通道 (日誌訊息的目標)
- 變更自動重建條件

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，也可記錄至日誌檔。(在 Solaris 上，錯誤訊息還可寫入至系統的 syslog (系統日誌) 常駐程式。)

您可以以下方法變更日誌訊息的輸出通道：

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

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

```
imq.log.console.output=INFO
```
- 在 Solaris 上，使用 `imq.log.syslog.output` 特性指定應寫入至 Solaris syslog 的記錄資訊的種類。例如，

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

備註 變更記錄程式輸出通道前，您必須確定已將記錄配置為支援您要對映到輸出通道的資訊的級別。例如，如果您要將日誌級別配置為 `ERROR`，並將 `imq.log.console.output` 特性配置為 `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日誌檔：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，您需要配置 `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
```

如果您同時配置了與時間相關的和與容量相關的特性，則先達到的限制將啓動自動重建。如上所述，代理程式最多可保留九個自動重建檔案。

代理程式和應用程式管理

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

- 控制代理程式的狀態：您可以暫停、繼續、關閉以及重新啓動代理程式。
- 查詢與更新代理程式特性
- 管理連接服務

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

- **Message Queue** 訊息經由代理程式目標路由至它們的接收者或用戶。您負責在代理程式上建立這些目標。
- **Message Queue** 為長期用戶配置並維護資源，即使擁有長期訂閱的用戶端成為非作用中的用戶端。使用 **Message Queue** 指令工具可取得長期訂閱的相關資訊，並可銷毀長期訂閱或清除它們的訊息以便儲存 **Message Queue** 資源。
- **Message Queue** 異動與分散式異動由代理程式追蹤。如果發生故障，您可能需要手動確定或回轉異動。

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

指令公用程式

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

imqcmd 指令的語法

imqcmd 指令的一般語法如下：

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

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

```
imqcmd restart bkr -v
```

imqcmd 子指令

指令公用程式 (imqcmd) 包括表 6-1 中列出的子指令：本章的相關工作各節內容會更加詳細說明子指令。

表 6-1 imqcmd 子指令

子指令與引數	說明
commit txn	確定異動。
compact dst	壓縮用於一個或多個目標的內建且基於檔案的資料倉庫。
create dst	建立目標。
destroy dst	銷毀目標。
destroy dur	銷毀長期訂閱。
list cxn	列出代理程式的連接。
list dst	列出代理程式上的目標。
list dur	列出主題的長期訂閱。
list svc	列出代理程式上的服務。
list txn	列出代理程式上的異動。
metrics bkr	顯示代理程式度量。
metrics dst	顯示目標度量。
metrics svc	顯示服務度量。
pause bkr	暫停代理程式上的所有服務。
pause dst	暫停代理程式上的一個或多個目標。

表 6-1 imqcmd 子指令 (續)

子指令與引數	說明
pause svc	暫停代理程式上的單一服務。
purge dst	清除目標上的所有訊息而不銷毀目標。
purge dur	清除長期訂閱上的所有訊息而不銷毀長期訂閱。
query bkr	查詢並顯示代理程式的相關資訊。
query cxn	查詢並顯示連接的相關資訊。
query dst	查詢並顯示目標的相關資訊。
query svc	查詢並顯示服務的相關資訊。
query txn	查詢並顯示異動的相關資訊。
reload cls	重新載入代理程式叢集配置。
restart bkr	重新啟動目前執行的代理程式實例。不能用於啟動新的代理程式實例。
resume bkr	繼續代理程式上的所有服務。
resume dst	繼續代理程式上一個或多個暫停的目標。
resume svc	繼續服務。
rollback txn	回轉異動。
shutdown bkr	關閉代理程式實例。接著可以使用 imqbrokerd 指令啟動，而不是使用 imqcmd 的 restart bkr 子指令。
update bkr	更新代理程式的屬性。
update dst	更新目標的屬性。
update svc	更新服務的屬性。

imqcmd 選項摘要

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

表 6-2 imqcmd 選項

選項	說明
-b <i>hostName:port</i>	指定代理程式主機名稱及其連接埠號。預設值為 <code>localhost:7676</code> 。 僅指定連接埠：-b :7878 僅指定名稱：-b somehost
-c <i>clientID</i>	將長期用戶的 ID 指定至主題。請參閱第 161 頁的「 管理長期訂閱 」。
-d <i>destinationName</i>	指定主題名稱。與 <code>list dur</code> 和 <code>destroy dur</code> 子指令配合使用。請參閱第 161 頁的「 管理長期訂閱 」。
-f	未經使用者確認即執行動作。
-h	顯示用法說明。不執行指令行上的其他任何指令。
-H	顯示用法說明、屬性清單和範例。不執行指令行上的其他任何指令。
-int <i>interval</i>	指定 <code>metrics bkr</code> 、 <code>metrics dst</code> 與 <code>metrics svc</code> 子指令顯示度量輸出的時間間隔 (以秒為單位)。
-javahome <i>path</i>	指定要使用的替代 Java 2 相容運行時間 (預設為使用系統上的運行時間，或 Message Queue 隨附的運行時間)。
-m <i>metricType</i>	指定要顯示的度量資訊類型。將此選項與 <code>metrics dst</code> 、 <code>metrics svc</code> 或 <code>metrics bkr</code> 子指令配合使用。 <i>metricType</i> 的值會根據是否產生用於目標、服務或代理程式的度量而變化。
-msp <i>numSamples</i>	指定度量輸出中，顯示作為 <code>metrics bkr</code> 、 <code>metrics dst</code> 與 <code>metrics svc</code> 子指令範例的度量數目。
-n <i>argumentName</i>	指定子指令引數的名稱。此選項可能為服務名稱、實體目標名稱、長期訂閱名稱、連接 ID 或異動 ID，這取決於子指令。
-o <i>attribute=value</i>	指定屬性的值。此選項可以為代理程式 (請參閱第 142 頁的「 管理代理程式 」) 的屬性、服務 (請參閱第 147 頁的「 管理連接服務 」) 的屬性或目標 (請參閱第 152 頁的「 管理目標 」) 的屬性，這取決於子指令引數。
-p <i>password</i>	指定您的 (管理員的) 密碼。如果您遺漏此值，系統將提示您提供此名稱。
-pst <i>pauseType</i>	暫停目標時，指定是否要暫停產生者、使用者或兩者皆暫停。請參閱第 152 頁的「 管理目標 」。
-rtm <i>timeout</i>	指定 <code>imqcmd</code> 子指令的初始 (重試) 逾時期間 (以秒為單位)。送出請求到代理程式後， <code>imqcmd</code> 子指令將等待的逾時時間長度。每個子指令的後續重試會使用一個逾時值，該逾時值為初始逾時期間的倍數。預設值：10
-rtr <i>numRetries</i>	指定 <code>imqcmd</code> 子指令第一次逾時後，嘗試重試的次數。預設值：5

表 6-2 imqcmd 選項 (續)

選項	說明
-s	無訊息模式。不顯示輸出。
-secure	使用 <code>ssladmin</code> 連接服務指定與代理程式的安全管理連接 (請參閱第 202 頁的「步驟 4. 配置並執行基於 SSL 的用戶端」)。
-svn <i>serviceName</i>	指定所列出連接的服務。請參閱第 151 頁的「取得連接資訊」。
-t <i>destType</i>	指定目標的類型：t (主題) 或 q (佇列)。請參閱第 152 頁的「管理目標」。
-tmp	顯示暫存目標。請參閱第 152 頁的表 6-9。
-u <i>userName</i>	指定您的 (管理員的) 名稱。如果您遺漏此值，系統將提示您提供此名稱。
-v	顯示版本資訊。不執行指令行上的其他任何指令。

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

備註 若想要使用 -secure 選項，您必須先配置並啟用目標代理程式實例中的 `ssladmin` 服務，如第 198 頁的「透過 TCP/IP 設定基於 SSL 的服務」所述。

使用 imqcmd 指令

為了使用 `imqcmd` 指令以管理代理程式，您必須執行以下作業：

- 使用 `imqbrokerd` 指令啟動代理程式。
請參閱第 122 頁的「啟動代理程式」。您僅可以使用指令公用程式管理執行中的代理程式；您不能使用它啟動代理程式。
- 使用 -b 選項指定目標代理程式，除非代理程式正在使用連接埠 7676 的本地主機上執行。
- 指定適當的管理員使用者名稱和密碼。如果您未指定，系統將提示您提供名稱和密碼。而且，請注意您使用 `imqcmd` 執行的每項作業都將由使用者儲存庫認證。如需更多資訊，請參閱第 184 頁的「認證使用者」。

當您安裝 Message Queue 時，同時也安裝了預設文字檔案使用者儲存庫。此儲存庫出廠時隨附兩個項目：一個用於擁有管理員權限的使用者，一個用於客人使用者。這些項目可讓您連接至代理程式實例而不執行任何其他工作。例如，如果您僅要測試 Message Queue，則可以使用預設的使用者名稱和密碼 (admin/admin) 來執行 imqcmd 公用程式。

如果您要配置生產系統，則將需要執行一些其他工作以認證和授權使用者 (請參閱第 8 章，「管理安全性」)。要特別注意的是，您必須在 Message Queue 使用者儲存庫中建立項目 (請參閱第 184 頁的「使用文字檔案使用者儲存庫」)。您還可以選擇使用您的使用者儲存庫的現有 LDAP 目錄伺服器 (請參閱第 190 頁的「將 LDAP 伺服器用於使用者儲存庫」)。

範例 imqcmd 的用法

以下範例說明 imqcmd 指令的使用：

- 若要列出在連接埠 7676 的 localhost 執行之代理程式的特性：

```
imqcmd query bkr -u admin -p admin
```
- 若要列出在連接埠 1564 的 myserver 執行之代理程式的特性，則使用者名稱爲 alladin，使用者密碼爲 abracadabra。

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

假設使用者名稱 alladin 已指定給 admin 群組，您將作為管理用戶端連接至指定的代理程式。
- 若要列出在連接埠 7676 的 localhost 執行之代理程式的特性，則該指令的初始逾時配置爲 20 秒，且重試 (逾時後) 次數配置爲 7。

```
imqcmd query bkr -u admin -p admin -rtm 20 -rtr 7
```

管理代理程式

指令公用程式包括您可用來執行以下代理程式管理工作的子指令：

- [顯示代理程式資訊](#)
- [更新代理程式特性](#)
- [顯示代理程式度量](#)
- [控制代理程式的狀態](#)

若要管理代理程式的連接服務，請參閱第 147 頁的「管理連接服務」。若要管理代理程式目標，請參閱第 152 頁的「管理目標」。

表 6-3 列出用來管理代理程式的 `imgcmd` 子指令。如果未指定主機名稱或連接埠，則使用預設值 (`localhost` 和 `7676`)。

表 6-3 用來管理代理程式的 `imgcmd` 子指令

子指令語法	說明
<code>metrics bkr [-b hostName:port]</code> <code>[-m metricType]</code> <code>[-int interval]</code> <code>[-msp numSamples]</code>	<p>顯示預設代理程式或指定主機和連接埠上的代理程式的代理程式度量。</p> <p>使用 <code>-m</code> 選項指定要顯示的度量類型：</p> <p>ttl 顯示代理程式流入和流出訊息和封包的度量。(預設度量類型)</p> <p>rts 顯示代理程式(每秒)流入和流出訊息和封包之流量速率的度量。</p> <p>cxn 顯示連接、虛擬記憶體堆疊和執行緒。</p> <p>使用 <code>-int</code> 選項，指定顯示度量的間隔時間(以秒為單位)。預設為 5 秒鐘。</p> <p>使用 <code>-msp</code> 選項，指定輸出中顯示的範例數目。預設的數目為沒有限制(無限)。</p>
<code>pause bkr [-b hostName:port]</code>	暫停預設代理程式或指定主機和連接埠上的代理程式。請參閱第 145 頁的「暫停與繼續代理程式」。
<code>query bkr -b hostName:port</code>	列出預設代理程式或指定主機和連接埠上的代理程式特性的目前配置。還顯示連接至指定代理程式的執行中的代理程式(在多重代理程式叢集中)之清單。
<code>reload cls</code>	僅適用於代理程式叢集。強制叢集中的所有代理程式重新載入 <code>img.cluster.brokerlist</code> 特性並更新叢集資訊。請參閱第 130 頁的「將代理程式新增至叢集」，以獲得更多資訊。
<code>restart bkr [-b hostName:port]</code>	關閉並重新啟動預設代理程式或指定主機和連接埠上的代理程式。
	請注意，此指令使用首次啟動代理程式時指定的選項重新啟動此代理程式。如果您要使用不同的選項，則必須關閉代理程式然後再次啟動它，從而指定您需要的選項。
<code>resume bkr [-b hostName:port]</code>	繼續預設代理程式或指定主機和連接埠上的代理程式。
<code>shutdown bkr [-b hostName:port]</code>	關閉預設代理程式或指定主機和連接埠上的代理程式。
<code>update bkr [-b hostName:port]</code> <code>-o attribute=value</code> <code>[-o attribute=value1]...</code>	變更預設代理程式或指定主機和連接埠上的代理程式的指定屬性。

請記住，當使用表 6-3 中列出的任何子指令時，您必須指定代理程式主機名稱和連接埠號，除非您的目標為在使用連接埠 7676 的 localhost 上執行的代理程式。

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。例如：

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

此指令的輸出如下：

Version	3.5 SP1
Instance Name	imqbroker
Primary Port	7676
Current Number of Messages in System	0
Current Total Message Bytes in System	0
Max Number of Messages in System	unlimited (-1)
Max Total Message Bytes in System	unlimited (-1)
Max Message Size	70m
Auto Create Queues	true
Auto Create Topics	true
Auto Created Queue Max Number of Active Consumers	1
Auto Created Queue Max Number of Backup Consumers	0
Cluster Broker List (active)	
Cluster Broker List (configured)	
Cluster Master Broker	
Cluster URL	
Log Level	INFO
Log Rollover Interval (seconds)	604800
Log Rollover Size (bytes)	unlimited (-1)

更新代理程式特性

您可以使用 `update bkr` 子指令更新表 6-4 中列出的任何代理程式特性。請注意，對代理程式的更新會自動寫入代理程式的實例配置檔案。

表 6-4 imqcmd 更新的代理程式特性

特性	參考
imq.autocreate.queue	第 71 頁的表 2-10

表 6-4 imqcmd 更新的代理程式特性 (續)

特性	參考
imq.autocreate.topic	第 71 頁的表 2-10
imq.autocreate.queue.maxNumActiveConsumers	第 71 頁的表 2-10
imq.autocreate.queue.maxNumBackupConsumers	第 71 頁的表 2-10
imq.cluster.url	第 127 頁的表 5-3
imq.log.level	第 66 頁的表 2-9
imq.log.file.rolloversecs	第 66 頁的表 2-9
imq.log.file.rolloverbytes	第 66 頁的表 2-9
imq.system.max_count	第 56 頁的表 2-4
imq.system.max_size	第 56 頁的表 2-4
imq.message.max_size	第 56 頁的表 2-4
imq.portmapper.port	第 52 頁的表 2-3

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 -p admin
```

控制代理程式的狀態

啟動代理程式之後，您可以使用以下 imqcmd 子指令控制代理程式的狀態。

暫停與繼續代理程式

- **暫停代理程式。** 暫停代理程式會暫停代理程式的連接服務執行緒，這將導致代理程式停止偵聽連接埠。因此，代理程式將無法繼續接受新的連接、接收訊息和派送訊息。

但是，暫停代理程式不會暫停管理連接服務。此服務可讓您執行所需的管理工作，以管理代理程式的訊息流量。例如，如果某特定目標受到訊息的攻擊，則您可以暫停代理程式，並執行以下可能幫助您解決問題的任何動作：追蹤訊息來源、限制目標大小或銷毀目標。

暫停代理程式也不會暫停叢集連接服務。但是，叢集中的訊息發送會根據叢集中不同代理程式執行的發送功能而有差異。

以下指令暫停在使用連接埠 1588 的 myhost 上執行的代理程式。

```
imqcmd pause bkr -b myhost:1588 -u admin -p admin
```

(您也可以暫停個別連接服務 (請參閱第 150 頁的「暫停與繼續連接服務」) 和個別目標 (請參閱第 158 頁的「暫停和繼續目標」))

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

```
imqcmd resume bkr -u admin -p admin
```

關閉和重新啟動代理程式

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

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

- **重新啟動代理程式。**關閉並重新啟動代理程式。以下指令重新啟動在使用連接埠 7676 的 localhost 上執行的代理程式：

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

顯示代理程式度量

若要顯示有關代理程式的度量資訊，請使用 `metrics bkr` 子指令。例如，若要取得每 10 秒時間間隔，代理程式流入與流出的訊息速率，請使用：

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

此指令的輸出如下：

Msgs/sec		Msg Bytes/sec		Pkts/sec		Pkt Bytes/sec	
In	Out	In	Out	In	Out	In	Out
0	0	27	56	0	0	38	66
10	0	7365	56	10	10	7457	1132
0	0	27	56	0	0	38	73
0	10	27	7402	10	20	1400	8459
0	0	27	56	0	0	38	73

如需使用 `imqcmd` 報告代理程式度量的更多詳細說明，請參閱第 220 頁的「監視工具」。

管理連接服務

指令公用程式包括允許您執行以下連接服務管理工作的子指令：

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

如需 Message Queue 連接服務的簡介，請參閱第 50 頁的「連接服務」。

表 6-5 列出用來管理連接服務的 `imqcmd` 子指令。如果未指定主機名稱或連接埠，則使用預設值 (`localhost` 和 `7676`)。

表 6-5 用於管理連接服務的 `imqcmd` 子指令

子指令語法	說明
<code>list svc [-b hostName:port]</code>	列出預設代理程式或指定主機和連接埠上的代理程式上的所有連接服務。
<code>metrics svc -n serviceName</code> <code>[-b hostName:port]</code> <code>[-m metricType]</code> <code>[-int interval]</code> <code>[-msp numSamples]</code>	顯示預設代理程式或指定主機和連接埠上的代理程式上的指定服務之標準資訊。 使用 <code>-m</code> 選項指定要顯示的度量類型： ttl 顯示代理程式經由指定服務，流入和流出訊息和封包的度量。(預設度量類型) rts 顯示代理程式經由指定服務，(每秒) 流入和流出訊息和封包之流量速率的度量。 cxn 顯示連接、虛擬記憶體堆疊和執行緒。 使用 <code>-int</code> 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 <code>-msp</code> 選項，指定輸出中顯示的範例數目。預設的數目為沒有限制 (無限)。
<code>pause svc -n serviceName</code> <code>[-b hostName:port]</code>	暫停在預設代理程式或指定主機和連接埠上的代理程式上執行的指定服務。您無法暫停管理服務。

表 6-5 用於管理連接服務的 imqcmd 子指令 (續)

子指令語法	說明
query svc -n <i>serviceName</i> [-b <i>hostName:port</i>]	顯示在預設代理程式或指定主機和連接埠上的代理程式上執行的指定服務的相關資訊。
resume svc -n <i>serviceName</i> [-b <i>hostName:port</i>]	繼續在預設代理程式或指定主機和連接埠上的代理程式上執行的指定服務。
update svc -n <i>serviceName</i> [-b <i>hostName:port</i>] -o <i>attribute=value</i> [-o <i>attribute=value1</i>] ...	更新在預設代理程式或指定主機和連接埠上的代理程式上執行的指定服務的指定屬性。如需服務屬性的說明，請參閱第 149 頁的表 6-7。

代理程式支援來自應用程式用戶端和管理用戶端的連接。目前，Message Queue 代理程式可提供的連接服務如表 6-6 中所示。[服務名稱] 欄中的值是指用於為 -n 選項指定服務名稱的值。如表格中所示，每種服務均由它使用的服務類型和基本傳輸層指定，所使用的服務類型為 NORMAL (應用程式用戶端) 或 ADMIN (管理用戶端)。

表 6-6 代理程式支援的連接服務

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

列出連接服務

若要列出代理程式上可用的連接服務，請使用以下指令：

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

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

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

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

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

此指令將輸出以下資訊：

Service Name	Port Number	Service State
admin	41844 (dynamic)	RUNNING
httpjms	-	UNKNOWN
httpsjms	-	UNKNOWN
jms	41843 (dynamic)	RUNNING
ssladmin	dynamic	UNKNOWN
ssljms	dynamic	UNKNOWN

顯示連接服務資訊

若要查詢並顯示單一服務的相關資訊，請使用查詢子指令。例如，

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

此指令的輸出如下：

Service Name	jms
Service State	RUNNING
Port Number	60920 (dynamic)
Current Number of Allocated Threads	0
Current Number of Connections	0
Min Number of Threads	10
Max Number of Threads	1000

更新連接服務特性

您可以使用 update 子指令變更表 6-7 中列出的一個或多個服務特性的值。

表 6-7 imqcmd 更新的連接服務特性

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配置連接埠。
minThreads	指定給服務的執行緒之最小數目。

表 6-7 imqcmd 更新的連接服務特性

特性	說明
maxThreads	指定給服務的最大執行緒數目。

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"
```

顯示連接服務度量

若要顯示有關單一服務的度量資訊，請使用 `metrics` 子指令。例如，若要取得由 `jms` 連接服務處理之訊息和封包的累計總數，請使用：

```
imqcmd metrics svc -n jms -m ttl -u admin -p admin
```

此指令的輸出如下：

```
-----
Msgs      Msg Bytes    Pkts      Pkt Bytes
In  Out    In   Out    In  Out    In   Out
-----
164 100 120704 73600 282 383 135967 102127
657 100 483552 73600 775 876 498815 149948
-----
```

如需使用 `imqcmd` 報告連接服務度量的更多詳細說明，請參閱第 220 頁的「[監視工具](#)」。

暫停與繼續連接服務

若要暫停管理服務（無法暫停）之外的任何服務，請使用以下指令：

```
imqcmd pause svc -n serviceName -u admin -p 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接。如果 Message Queue 用戶端嘗試開啓新的連接，將會發生異常。

- 暫停服務上的所有現有連接會維持持續狀態，但在繼續服務前，代理程式會暫停此類連接上的所有訊息處理。(例如，如果用戶端嘗試傳送訊息，在繼續服務前，`send()` 方法會封鎖此訊息。)
- 將會維護任何已被代理程式接收之訊息的訊息發送狀態。(例如，繼續服務時，不會中斷異動且將繼續訊息發送。)

若要繼續服務，請使用以下指令：

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

取得連接資訊

指令公用程式包括允許您列出和取得連接相關資訊的子指令。

表 6-8 列出適用於連接的 `imqcmd` 子指令。如果未指定主機名稱或連接埠，則假設它們為 `localhost` 和 `7676`。

表 6-8 用於管理連接服務的 `imqcmd` 子指令

子指令語法	說明
<code>list cxn [-svn serviceName] [-b hostName:port]</code>	列出預設代理程式上指定服務名稱的所有連接，或指定主機和連接埠上代理程式上的所有連接。如果未指定服務名稱，則會列出所有服務。
<code>query cxn -n connectionID [-b hostName:port]</code>	顯示在預設代理程式或指定主機和連接埠上的代理程式上，指定服務的相關資訊。

若要查詢並顯示單一服務的相關資訊，請使用查詢子指令。例如，

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

此指令的輸出如下：

```
Connection ID    421085509902214374
User             guest
Service         jms
Producers       0
Consumers       1
Host             111.22.333.444
Port            60953
Client ID
Client Platform
```

管理目標

所有 Message Queue 訊息均可經由在特定代理程式上建立的佇列和主題目標，路由至它們的使用者用戶端。

指令公用程式包括允許您執行以下目標管理工作的子指令：

- 建立目標
- 列出目標
- 顯示目標資訊
- 更新目標屬性
- 顯示目標度量
- 暫停和繼續目標
- 清除目標
- 銷毀目標
- 壓縮目標

如需目標的介紹，請參閱第 68 頁的「實體目標」。

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

表 6-9 用於管理目標的 imqcmd 子指令

子指令語法	說明
<code>compact dst [-t <i>destType</i> -n <i>destName</i>]</code>	壓縮用於指定類型與名稱之目標的內建且基於檔案的資料倉庫。如果沒有指定目標類型與名稱，則會壓縮所有目標。必須在壓縮目標前暫停目標。
<code>create dst -t <i>destType</i> -n <i>destName</i> [-o <i>attribute=value</i>] [-o <i>attribute=value1</i>]...</code>	建立指定類型、指定名稱以及指定屬性的目標。目標名稱必須僅包含字母數字字元 (無空格)，並且以字母字元或字元「_」和「\$」開頭。它們不可以以「mq」字元字串開頭。
<code>destroy dst -t <i>destType</i> -n <i>destName</i></code>	銷毀指定類型和名稱的目標。

表 6-9 用於管理目標的 imqcmd 子指令 (續)

子指令語法	說明
list dst [-t <i>destType</i>] [-tmp]	<p>列出所有指定類型的目標，以及列出暫存目標的選項 (請參閱第 72 頁的「暫存目標」)。</p> <p>類型引述可以有兩個值：</p> <p><i>destType</i> = q (queue) <i>destType</i> = t (topic)</p> <p>如果未指定類型，則會列出所有類型的所有目標。</p>
metrics dst -t <i>destType</i> -n <i>destName</i> [-m <i>metricType</i>] [-int <i>interval</i>] [-msp <i>numSamples</i>]	<p>顯示有關指定類型和名稱之目標的度量資訊。</p> <p>使用 -m 選項指定要顯示的度量類型：</p> <p>ttl 顯示目標流入和流出訊息和封包以及常駐記憶體 的度量。(預設度量類型)</p> <p>rts 顯示目標(每秒)流入和流出訊息和封包之流量 速率的度量，及其他速率資訊。</p> <p>con 顯示使用者相關的度量。</p> <p>dsk 顯示磁碟使用度量。</p> <p>使用 -int 選項，指定顯示度量的間隔時間 (以秒為單位))。預設為 5 秒鐘。</p> <p>使用 -msp 選項，指定輸出中顯示的範例數目。預設的 數目為沒有限制 (無限)。</p>
pause dst [-t <i>destType</i>] -n <i>destName</i> [-pst <i>pauseType</i>]	<p>針對指定類型和名稱的目標，暫停至使用者的訊息發送 (-pst CONSUMERS)，或暫停來自產生者的訊息發送 (-pst PRODUCERS)，或者兩者皆暫停 (-pst ALL)。如果 沒有指定目標類型與名稱，則會暫停所有目標。預設為 ALL。</p>
purge dst -t <i>destType</i> -n <i>destName</i>	<p>清除在指定類型與名稱的目標的訊息。</p>
query dst -t <i>destType</i> -n <i>destName</i>	<p>列出指定類型和名稱之目標的相關資訊。</p>
resume dst [-t <i>destType</i>] -n <i>destName</i>	<p>繼續用於指定類型和名稱之暫停目標的訊息發送。如果 沒有指定目標類型與名稱，則會繼續所有目標。</p>
update dst -t <i>destType</i> -n <i>destName</i> -o <i>attribute=value</i> [-o <i>attribute=value1</i>]...	<p>更新指定目標上的指定屬性的值。</p> <p>屬性名稱可以為表 6-10 中所述的任何屬性。</p>

建立目標

建立目標時，您必須指定以下項目：

- 目標類型：主題或佇列
- 目標名稱：必須僅包含字母數字字元（無空格），並且以字母字元或字元「_」和「\$」開頭。名稱不可以以「mq」字元字串開頭。
- 任何用於目標屬性的非預設值

很多目標屬性是用來管理代理程式記憶體資源和訊息流量。例如，您可以指定允許用於目標的產生者最大數目，或目標中允許的最大訊息數目（或大小）。這些限制與使用代理程式配置特性，在代理程式範圍基礎上進行配置相似（請參閱第 55 頁的「管理記憶體資源和訊息流量」）。您也可以指定達到這些限制時，代理程式確認的方式。

還有僅適用於佇列目標的目標屬性。這些屬性可用來指定使用中或備份使用者的數目，這些使用者在至多個使用者的訊息負載平衡發送時使用（請參閱第 69 頁的「佇列目標」）。

表 6-10 描述適用於每個目標類型的屬性。建立或更新目標時，可以配置屬性值。針對自動建立的目標，請在代理程式實例配置檔案中配置預設特性值。（請參閱第 115 頁的「配置檔案」）。

表 6-10 目標屬性

目標類型	屬性	預設值	說明
佇列和主題	maxNumMsgs ¹	-1 (無限制)	指定目標中允許給未使用訊息的最大數目。
佇列和主題	maxTotalMsgBytes ¹	-1 (無限制)	指定目標中允許給未使用訊息的最大總記憶體容量（以字元組為單位）。
佇列和主題	limitBehavior	REJECT_ NEWEST	指定達到記憶體限制執行緒時，代理程式確認的方式。值為： FLOW_CONTROL - 減緩產生者速度 REMOVE_OLDEST - 拋出最舊的訊息 REMOVE_LOW_PRIORITY - 根據訊息存在時間，拋出最低優先權的訊息（生產型用戶端不會收到訊息刪除的通知） REJECT_NEWEST - 拒絕最新資訊（拒絕永久性訊息時，生產型用戶端會發生異常，但拒絕非永久性訊息時則不會收到通知）
佇列和主題	maxBytesPerMsg	-1 (無限制)	指定允許給目標中任何單一訊息的最大容量（以字元組為單位）。拒絕永久性訊息時，生產型用戶端會發生異常，但拒絕非永久性訊息時則不會收到通知。

1. 在叢集環境中，此特性適用於叢集中的每個目標實例，但整體上，不適用於叢集中的所有實例。

表 6-10 目標屬性 (續)

目標類型	屬性	預設值	說明
佇列和主題	maxNumProducers ¹	-1 (無限制)	指定允許給目標中產生者的最大數目。達到限制時，將無法建立新的產生者。
僅佇列	maxNumActiveConsumers	1	指定最大使用者數，此數值可作用於來自佇列目標的負載平衡發送。值 -1 表示沒有限制數目。(平台版本會將此值限制為 2。)
僅佇列	maxNumBackupConsumers	0	指定最大備份使用者數，如果無法從佇列目標進行負載平衡發送，那麼這些使用者可以取代作用中的使用者。值 -1 表示沒有限制數目。
佇列和主題	consumerFlowLimit	主題：1000 佇列：1000	指定最大訊息數目，這些訊息將會發送給單一批次中的某個使用者。在負載平衡佇列發送中，負載平衡開始前，此數目是路由到作用中使用者的佇列訊息初始數目。(請參閱第 69 頁的「佇列發送至多個使用者」)。目標的使用者在其個別連接上，可指定一個較低的值來置換此限制(請參閱「Message Queue Java 用戶端開發人員指南」中連線工廠屬性的相關資訊)。值 -1 表示沒有限制數目。
僅佇列	localDeliveryPreferred	false	僅適用於代理程式叢集中的負載平衡佇列發送。如果本地代理程式沒有使用者，則指定訊息僅發送給遠端使用者。要求目標不僅限於本地發送(isLocalOnly = false)。
佇列和主題	isLocalOnly	false	僅適用於代理程式叢集。指定在其他代理程式上不重複目標，且限制目標僅發送訊息給本地使用者(連接到目標上所建立之代理程式的使用者)。一旦建立目標，即無法更新屬性。

1. 在叢集環境中，此特性適用於叢集中的每個目標實例，但整體上，不適用於叢集中的所有實例。

- 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"
```

請注意：目標名稱必須僅包含字母數字字元(無空格)，且以字母字元或字元「_」和「\$」開頭。它不可以以「mq」(保留供度量主題目標使用)字元字串開頭(請參閱第 65 頁的表 2-8)。

- 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"
```

列出目標

您可以取得目標目前屬性值的相關資訊，這些屬性值是關於與目標相關之產生者或使用者的數目，以及訊息傳送度量，例如目標中訊息的數目和大小。

若要尋找您要取得相關資訊的目標，您可以先使用 `list dst` 子指令列出特定代理程式上的所有目標。例如，若要取得在連接埠 4545 的 `myHost` 上執行之代理程式的所有目標，請輸入以下指令：

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

`list dst` 子指令可選擇性指定要列出的目標類型，或選擇性納入暫存目標 (使用 `-tmp` 選項)。用戶端建立的暫存目標，通常用來接收傳送到其他用戶端的訊息確認 (請參閱第 72 頁的「暫存目標」)。

顯示目標資訊

若要取得目標目前屬性值的相關資訊，請使用 `query dst` 子指令，例如下列指令：

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

此指令的輸出如下：

```
-----
Destination Name      Destination Type
-----
XQueue                Queue

On the broker specified by:

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

Destination Name      XQueue
Destination Type      Queue
Destination State     RUNNING
Created Administratively true

Current Number of Messages      0
Current Total Message Bytes     0
Current Number of Producers     0
Current Number of Active Consumers 0
Current Number of Backup Consumers 0

Max Number of Messages      unlimited (-1)
Max Total Message Bytes     unlimited (-1)
```

Max Bytes per Message	unlimited (-1)
Max Number of Producers	100
Max Number of Active Consumers	1
Max Number of Backup Consumers	0
Limit Behavior	REJECT_NEWEST
Consumer Flow Limit	100
Is Local Destination	false
Local Delivery is Preferred	false

輸出也會顯示與目標相關之產生者與使用者的數目。對佇列目標而言，此輸出同時包括作用中和備份使用者。

您可以使用 `update dst` 子指令變更一個或多個屬性的值（請參閱第 157 頁的「更新目標屬性」）。

更新目標屬性

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

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

請參閱第 154 頁的表 6-10，以獲得您可以更新的屬性清單。

您無法使用 `update dst` 子指令更新目標的 *類型* 或更新 `isLocalOnly` 屬性。

顯示目標度量

若要取得與目標相關的訊息度量資訊，請使用 `metrics dst` 子指令，例如下列指令：

```
imqcmd metrics dst -t q -n XQueue -m ttl -u admin -p admin
```

此指令的輸出如下：

Msgs		Msg Bytes		Msg Count			Total Msg Bytes (k)			Largest
In	Out	In	Out	Current	Peak	Avg	Current	Peak	Avg	Msg (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

如需使用 `imqcmd` 報告目標度量的更多詳細說明，請參閱第 220 頁的「[監視工具](#)」。

暫停和繼續目標

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

若要暫停發送到目標的訊息或暫停從目標發送的訊息，請使用 `pause dst` 子指令，如以下指令所示：

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

```
imqcmd pause dst -n myTopic -t t -pst CONSUMERS -u admin -p admin
```

當您暫停目標後想要繼續發送，請輸入以下指令：

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

在多重代理程式叢集中，目標實例會常駐在叢集中的每個代理程式。您必須個別暫停這些目標。

清除目標

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

若要清除在目標的訊息，請使用 `purge dst` 子指令，如以下指令所示：

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

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

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

這可避免重新啓動代理程式後清除目標的麻煩。

在多重代理程式叢集中，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。

銷毀目標

若要銷毀目標，請使用 `destroy dst` 子指令，如以下指令所示：

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

銷毀目標會清除此目標上的所有資訊，並從代理程式中移除此目標；此作業不可復原。

壓縮目標

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

建立基於檔案的訊息倉庫，以便根據訊息於目標的位置，將訊息儲存在目錄中。在各個目標的目錄中，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，大小超過可配置執行緒的訊息將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目標監視磁碟使用及回收可用磁碟空間的子指令。

監視目標的磁碟使用

若要監視目標的磁碟使用，請使用以下 `imqcmd` 子指令：

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

此指令的輸出如下：

```
-----
Reserved   Used      Utilization Ratio
-----
806400     804096    99
1793024    1793024   100
2544640    2518272   98
```

子指令輸出中的欄有以下意思：

表 6-11 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（以字元組為單位），包括保留作用中訊息和等待被重新使用之可用記錄的記錄
已使用	保留作用中訊息的記錄所使用的磁碟空間（以字元組為單位）
使用率	使用的磁碟空間除以保留的磁碟空間所得之商數比率越高表示有越多磁碟空間被用來保留作用中訊息。

回收未使用的目標磁碟空間

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

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

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

如果保留的磁碟空間不斷增加，您應藉由配置目標記憶體限制特性和限制行爲，重新配置目標的記憶體管理 (請參閱第 154 頁的表 6-10)。

► 若要回收未使用的目標磁碟空間

1. 暫停目標。

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

2. 壓縮磁碟。

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

3. 繼續目標。

```
imqcmd resume dst -t q -n myQueue -u admin -p admin
```

如果未指定目標類型和名稱，則會在所有目標上執行這些作業。

管理長期訂閱

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

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

表 6-12 用於管理長期訂閱的 `imqcmd` 子指令

子指令	說明
<code>list dur -d destName</code>	列出指定目標的所有長期訂閱。
<code>destroy dur -n subscrName -c client_id</code>	使用指定的用戶端識別碼銷毀指定的長期訂閱 (請參閱第 43 頁的「用戶端識別碼」)。
<code>purge dur -n subscrName -c client_id</code>	使用指定的用戶端識別碼清除指定之長期訂閱的所有訊息 (請參閱第 43 頁的「用戶端識別碼」)。

例如，以下指令列出主題 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（作用中 / 非作用中）。例如：

Name	Client ID	Number of Messages	Durable Sub State
-----	-----	-----	-----
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。使用訂閱名稱和用戶端 ID 識別此訂閱。例如：

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

管理異動

由用戶端應用程式啟動的所有異動可由代理程式追蹤。這些異動可以是簡單的 Message Queue 異動或由 XA 資源管理員管理的分散式異動（請參閱第 44 頁的「本地異動」）。每個異動都有 Message Queue 異動 ID，即在代理程式上唯一識別異動的 64 位元的號碼。分散式異動還有由分散式異動管理員指定的分散式異動 ID (XID)，此 ID 長達 128 個位元組。Message Queue 可維護 Message Queue 異動 ID 與 XID 的關聯。

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

表 6-13 提供了 imqcmd 異動子指令的摘要。如果代理程式不是預設 (localhost:7676) 代理程式，請記得指定它的主機名稱和連接埠。

表 6-13 用於管理異動的 imqcmd 子指令

子指令	說明
list txn	列出要由代理程式追蹤的所有異動。
query txn -n <i>transaction_id</i>	列出指定異動的相關資訊。

表 6-13 用於管理異動的 imqcmd 子指令 (續)

子指令	說明
<code>commit txn -n transaction_id</code>	確定指定的異動。
<code>rollback txn -n transaction_id</code>	回轉指定的異動。

例如，以下指令列出代理程式中的所有異動。

```
imqcmd list txn
```

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

Transaction ID	State	User name	# Msgs/ # Acks	Creation time
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異動和分散式異動。您僅能確定或回轉處於 PREPARED 狀態的異動。如果您知道由於故障使異動處於此狀態，並且此異動未處於由分散式異動管理員確定的過程中，您應確定或回轉此異動。

例如，如果代理程式的自動回轉特性配置為 `false` (請參閱第 56 頁的表 2-4)，那麼您必須在啟動代理程式時手動確定或回轉處於 PREPARED 狀態的異動。

`list` 子指令還顯示在異動中產生的訊息數目，以及在異動 (#Msgs/#Acks) 中確認的訊息數目。這些訊息不會被發送，並且確認也不會被處理，直至異動被確定。

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

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

產生的輸出如下：

Client ID	
Connection	guest@192.18.116.219:62209->jms:62195
Creation time	1/30/02 10:08:31 AM

```
Number of acknowledgements 0
Number of messages         4
State                      PREPARED
Transaction ID             64248349708800
User name                  guest
XID
6469706F6C7369646577696E6465723130313234313431313030373230
```

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

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

還可配置代理程式，以在啓動代理程式時自動回轉處於 `PREPARED` 狀態的異動。請參閱第 56 頁的表 2-4 中的 `imq.transaction.autorollback` 特性，以獲得更多資訊。

管理受管理物件

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

如需受管理物件的簡介，請參閱第 79 頁的「Message Queue 受管理物件」。

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

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

本章介紹如何使用物件管理員公用程式 (mqobjmgr) 以執行這些工作。因為這些工作涉及您使用的物件倉庫與您建立的受管理物件之屬性理解，在描述如何使用 mqobjmgr 管理受管理物件前，本章會先提供這兩個主題的背景。

如需使用管理主控台的資訊，請參閱第 4 章「管理主控台指導」。

關於物件倉庫

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

LDAP 伺服器物件倉庫

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

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

使用 LDAP 伺服器作為物件倉庫時，您必須指定表 7-1 中顯示的屬性。這些屬性分為以下幾個種類：

- **初始環境**：此屬性的值針對 LDAP 伺服器物件倉庫進行修正。
- **位置**：指定 LDAP 伺服器中，設定用來儲存您受管理物件的 URL 與目錄路徑。要特別注意的是，您必須檢查指定的路徑是否存在。
- **安全性資訊**：根據 LDAP 提供者而異。您應查閱 LDAP 實施隨附的說明文件，以確定是否在所有作業上還是僅在變更已儲存資料的作業上需要安全性資訊。

表 7-1 LDAP 物件倉庫屬性

屬性	說明
java.naming.factory.initial	LDAP 伺服器上 JNDI 查找的初始環境 com.sun.jndi.ldap.LdapCtxFactory
java.naming.provider.url	LDAP 伺服器 URL 與目錄路徑資訊。例如： ldap://mydomain.com:389/ou=mqobjs, o=myapp 受管理物件儲存於 /myapp/mqobjs 目錄中
java.naming.security.principal	認證 LDAP 伺服器之呼叫者的負責人的身份。此項目的格式取決於認證機制。例如： uid=fooUser, ou=People, o=mq 如果沒有指定此特性，則運作方式將由 LDAP 服務提供者決定。
java.naming.security.credentials	認證 LDAP 伺服器之呼叫者的負責人的憑證。此特性的值取決於認證機制：它可以是隨機密碼、清除文字密碼、密鑰以及憑證等。例如： fooPasswd 如果沒有指定此特性，則運作方式將由 LDAP 服務提供者決定。

表 7-1 LDAP 物件倉庫屬性 (續)

屬性	說明
java.naming.security.authentication	<p>要使用的安全性級別。它的值為以下其中一各關鍵字： none、simple 和 strong。</p> <p>舉例來說，如果您指定 simple，則會提示您任何遺漏的負責人或憑證值。這可讓您以更安全的方式提供識別資訊。</p> <p>如果沒有指定此特性，則運作方式將由 LDAP 服務提供者決定。</p>

檔案系統物件倉庫

Message Queue 還支援檔案系統物件倉庫實施。當檔案系統物件倉庫未經過完全測試並因此不建議用於生產系統時，它的優點是易於在開發環境中使用。您所要做的是在您的本機檔案系統上建立目錄，而不是設定 LDAP 伺服器。

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

使用檔案系統物件倉庫時，您必須指定表 7-2 中顯示的屬性。這些屬性分為以下幾個種類：

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

表 7-2 檔案系統物件倉庫屬性

屬性	說明
java.naming.factory.initial	<p>檔案系統物件倉庫上 JNDI 查找的初始環境： com.sun.jndi.fscontext. RefFSContextFactory</p>
java.naming.provider.url	<p>目錄路徑資訊例如： file:///C:/myapp/mqobjjs</p>

受管理物件

如需受管理物件的簡介，請參閱第 79 頁的「[Message Queue 受管理物件](#)」。

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

依據訊息發送模型 (點對點或出版 / 訂閱) 來使用特定類型的連線工廠和目標。例如，在點對點程式設計中，可使用佇列連線工廠和佇列目標。同樣，在出版和訂閱程式設計中，可使用主題連線工廠和主題目標。還有非特定的連線工廠和目標受管理物件類型可用，如支援分散式異動的連線工廠類型 (請參閱第 42 頁的表 1-1，以獲得所有支援的類型)。

使用屬性值對指定的受管理物件的屬性。以下各節將描述這些屬性。

連線工廠受管理物件屬性

連線工廠 (和 XA 連線工廠) 受管理物件具有表 7-3 中列出的屬性。與您有主要關係的屬性為 `imqAddressList`，它可用於指定用戶端將與其建立連接的代理程式。第 176 頁的「[新增連線工廠](#)」一節介紹當您將連線工廠受管理物件新增至您的物件倉庫時，如何指定屬性。

如需連線工廠屬性的更多說明與有關如何使用它們的資訊，請參閱「[Message Queue Java Client Developer's Guide](#)」和用於 Message Queue 類別的 JavaDoc API 說明文件：`com.sun.messaging.ConnectionConfiguration`。

表 7-3 連線工廠受管理物件屬性

屬性 / 特性名稱	類型	預設值
<code>imqAckOnAcknowledge</code>	字串	無
<code>imqAckOnProduce</code>	字串	無
<code>imqAckTimeout</code>	字串	0 毫秒
<code>imqAddressList</code>	字串	無
<code>imqAddressListIterations</code>	整數	1
<code>imqAddressListBehavior</code>	字串	PRIORITY
<code>imqBrokerHostName (Message Queue 3.0)</code>	字串	localhost

表 7-3 連線工廠受管理物件屬性 (續)

屬性 / 特性名稱	類型	預設值
imqBrokerHostPort (Message Queue 3.0)	整數	7676
imqBrokerServicePort (Message Queue 3.0)	整數	0
imqConfiguredClientID	字串	無
imqConnectionFlowCount	整數	100
imqConnectionFlowLimit	整數	1000
imqConnectionFlowLimitEnabled	布林	false
imqConnectionType (Message Queue 3.0)	字串	TCP
imqConnectionURL (Message Queue 3.0)	字串	http://localhost/imq/ tunnel
imqConsumerFlowLimit	整數	1000
imqConsumerFlowThreshold	整數	50
imqDefaultPassword	字串	guest
imqDefaultUsername	字串	guest
imqDisableSetClientID	布林	false
imqJMSDeliveryMode	整數	2 (永久性的)
imqJMSExpiration	時間長度	0 (不會過期)
imqJMSPriority	整數	4 (一般)
imqLoadMaxToServerSession	布林	true
imqOverrideJMSDeliveryMode	布林	false
imqOverrideJMSExpiration	布林	false
imqOverrideJMSHeadersToTemporaryDestinations	布林	false
imqOverrideJMSPriority	布林	false
imqQueueBrowserMaxMessagesPerRetrieve	整數	1000
imqQueueBrowserRetrieveTimeout	時間長度	60,000 (毫秒)
imqReconnectAttempts	整數	0
imqReconnectEnabled	布林	false
imqReconnectInterval	時間長度	3000 (毫秒)

表 7-3 連線工廠受管理物件屬性 (續)

屬性 / 特性名稱	類型	預設值
imqSetJMSXAppID	布林	false
imqSetJMSXConsumerTXID	布林	false
imqSetJMSXProducerTXID	布林	false
imqSetJMSXRcvTimestamp	布林	false
imqSetJMSXUserID	布林	false
imqSSLIsHostTrusted (Message Queue 3.0)	布林	true

目標受管理物件屬性

目標受管理物件識別實體主題或佇列目標，其具有表 7-4 中列出的屬性。第 177 頁的「新增主題佇列」一節介紹當您將目標受管理物件新增至您的物件倉庫時，如何指定這些屬性。

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

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

表 7-4 目標受管理物件屬性

屬性 / 特性名稱	類型	預設
<code>imqDestinationDescription</code>	字串	目標物件的說明
<code>imqDestinationName</code>	字串 ¹	<code>Untitled_Destination_Object</code>

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

物件管理員公用程式 (imqobjmgr)

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

imgobjmgr 指令的語法

imgobjmgr 指令的一般語法如下：

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

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

```
imgobjmgr list -v
```

imgobjmgr 子指令

物件管理員公用程式 (imgobjmgr) 包括表 7-5 中列出的子指令：

表 7-5 imgobjmgr 子指令

子指令	說明
add	將受管理物件新增至物件倉庫。
delete	從物件倉庫刪除受管理物件。
list	列出物件倉庫中的受管理物件。
query	顯示有關指定的受管理物件的資訊。
update	修改物件倉庫中的現有受管理物件。

imgobjmgr 指令選項摘要

表 7-6 列出 imgobjmgr 指令的選項。如需它們用法的說明，請參閱以下基於工作的各節。

表 7-6 imgobjmgr 選項

選項	說明
-f	未經使用者確認即執行動作。
-h	顯示用法說明。不執行指令行上的其他任何指令。
-H	顯示用法說明、屬性清單和範例。不執行指令行上的其他任何指令。

表 7-6 imqobjmgr 選項 (續)

選項	說明
-i <i>fileName</i>	指定指令檔的名稱，此檔案包含所有或部分子指令子句，可指定物件類型、查找名稱、物件屬性、物件倉庫屬性或其他選項。通常用於重複資訊，如物件倉庫屬性。
-j <i>attribute=value</i>	指定識別和存取 JNDI 物件倉庫所需的屬性。請參閱第 166 頁的「LDAP 伺服器物件倉庫」和第 167 頁的「檔案系統物件倉庫」。
-j <i>javahome path</i>	指定要使用的替代 Java 2 相容運行時間 (預設為使用系統上的運行時間，或 Message Queue 隨附的運行時間)。
-l <i>lookupName</i>	指定受管理物件的 JNDI 查找名稱。此名稱在物件倉庫的環境中必須唯一。
-o <i>attribute=value</i>	指定受管理物件的屬性。請參閱第 168 頁的「連線工廠受管理物件屬性」和第 170 頁的「目標受管理物件屬性」。
-pre	預覽模式。表示在不執行指令的情況下將進行的動作。
-r <i>read-only_state</i>	指定受管理物件是否為唯讀物件。值 true 表示受管理物件為唯讀物件。用戶端無法修改唯讀受管理物件的屬性。依預設，唯讀狀態設定為 false。
-s	無訊息模式。不顯示輸出。
-t <i>objectType</i>	指定 Message Queue 受管理物件的類型： q = 佇列 t = 主題 cf = 連線工廠 qf = 佇列連線工廠 tf = 主題連線工廠 xcf = XA 連線工廠 (分散式異動) xqf = XA 佇列連線工廠 (分散式異動) xtf = XA 主題連線工廠 (分散式異動) e = SOAP 終點 ¹
-v	顯示版本資訊。不執行指令行上的其他任何指令。

1. 此受管理物件類型用於支援 SOAP 訊息 (請參閱「Message Queue Java Client Developer's Guide」)。

下節描述使用任何 imqobjmgr 子指令時需要提供的資訊。

需要的資訊

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

- **受管理物件類型：**

[表 7-6](#) 中顯示了允許的類型。

- **受管理物件的 JNDI 查找名稱：**

這是將在用戶端程式碼中使用的邏輯名稱，以參考物件倉庫中的受管理物件 (使用 JNDI)。

- **受管理物件屬性 (add 與 update 子指令特別需要)：**

- 對於目標：代理程式上實體目標的名稱。這是使用 `imqcmd create dst` 子指令的 `-n` 選項指定的名稱。如果您沒有指定此名稱，則將使用 `Untitled_Destination_Object` 的預設名稱。
- 對於連線工廠：大部分的通用屬性為位址清單 (`imqAddressList`)，它可指定用戶端嘗試連接的訊息伺服器位址 (一個或多個)。如果您沒有指定此資訊，則會使用本地主機與預設連接埠號 (7676)，這表示用戶端將會連接到本地主機連接埠 7676 上的代理程式。[第 176 頁的「新增連線工廠」](#)一節介紹如何指定物件屬性。

如需其他屬性，請參閱[第 168 頁的「連線工廠受管理物件屬性」](#)。

- **物件倉庫屬性：**

此資訊取決於您要使用檔案系統倉庫還是 LDAP 伺服器，但必須包括以下屬性：

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

如需有關物件倉庫屬性的更多資訊，請參閱[第 166 頁的「LDAP 伺服器物件倉庫」](#)和[第 167 頁的「檔案系統物件倉庫」](#)。

使用指令檔

`imqobjmgr` 指令可讓您指定使用 Java 特性檔案語法的指令檔的名稱，以表示所有或部分 `imqobjmgr` 子指令子句。

將指令檔與物件管理員公用程式 (imgobjmgr) 配合使用，對指定物件倉庫屬性極為有用，這可能與跨 imgobjmgr 的多重呼叫一樣，並且一般需要大量的鍵入。使用指令檔還可讓您避免鍵入可能超出指令行所允許的最大字元數的情況。

imgobjmgr 指令檔的一般語法如下 (版本內容會反映指令檔版本，但不會反映 Message Queue 產品的版本 - 它不是指令行選項且其值必須設定為 2.0)：

```
version=2.0
cmdtype=[ add | delete | list | query | update ]
obj.type=[ q | t | qf | tf | cf | xqf | xtf | xcf | e ]
obj.lookupName=lookup name
obj.attrs.objAttrName1=value1
obj.attrs.objAttrName2=value2
obj.attrs.objAttrNameN=valueN
0
objstore.attrs.objStoreAttrName1=value1
objstore.attrs.objStoreAttrName2=value2
objstore.attrs.objStoreAttrNameN=valueN
0
```

作為如何使用指令檔的範例，請考量以下 imgobjmgr 指令：

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

此指令可封裝在檔案 (如 MyCmdFile) 中，此檔案具有以下內容：

```
version=2.0
cmdtype=add
obj.type=qf
obj.lookupName=cn=myQCF
obj.attrs.imgAddressList=mq://foo:777/jms
```

```

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

```

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

```
imgobjmgr -i MyCmdFile
```

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

```

imgobjmgr add
-t qf
-l "cn=myQCF"
-o "imgAddressList=mq://foo:777/jms"
-i MyCmdFile

```

在此例中，檔案 `MyCmdFile` 將包含以下定義：

```

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

```

指令檔的其他範例可在以下位置找到：

```
IMQ_HOME/demo/imgobjmgr
```

新增和刪除受管理物件

本節介紹如何將連線工廠和主題或佇列目標的受管理物件新增至物件倉庫。

備註 物件管理員公用程式 (mqobjmgr) 僅列出並顯示 Message Queue 受管理物件。如果物件倉庫應包含的非 Message Queue 物件與您想新增的受管理物件具有相同的查找名稱，那麼當您嘗試新增作業時會出現錯誤。

新增連線工廠

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

若要新增佇列連線工廠，請使用以下指令：

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

上述指令建立查找名稱為 cn=myQCF 的受管理物件，此物件將連接至在 myHost 上執行的代理程式，並偵聽連接埠 7272。此受管理物件儲存在 LDAP 伺服器中。您可以透過將指令檔指定為 mqobjmgr 指令的引數來完成相同的作業。如需更多資訊，請參閱第 173 頁的「使用指令檔」。

備註 **命名慣例：**如果您要使用 LDAP 伺服器儲存受管理物件，則指定字首為「cn=」的查找名稱很重要（如上述範例所示：cn=myQCF）。您可以使用 -l 選項指定查找名稱。如果您正在使用檔案系統物件倉庫，那麼您不需使用 cn 字首，但請勿使用包含 "/" 的查找名稱。請參閱表 7-7。

表 7-7 命名慣例範例

物件倉庫類型	正確名稱	錯誤名稱
LDAP 伺服器	cn=myQCF	myQCF
檔案系統	myTopic	myObjects/myTopic

新增主題佇列

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

在將相應的受管理物件新增至物件倉庫前，最好先建立實體目標。使用指令公用程式 (imgcmd) 以在代理程式上建立實體目標，這些實體目標可由物件倉庫中的目標受管理物件識別。如需有關建立實體目標的資訊，請參閱第 151 頁的「取得連接資訊」。

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

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

這是相同的指令，只是受管理物件儲存在 Solaris 檔案系統中：

```
imgobjmgr add
-t t
-l "cn=myTopic"
-o "imgDestinationName=TestTopic"
-j "java.naming.factory.initial=
```

```

        com.sun.jndi.fscontext.RefFSContextFactory"
-j "java.naming.provider.url=
    file:///home/foo/imq_admin_objects"

```

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

```

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

```

使用 `-i` 選項將檔案傳送至 `imqobjmgr` 指令：

```
imqobjmgr -i MyCmdFile
```

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

除了為 `-t` 選項指定 `q`，新增佇列物件的方法完全相同。

刪除受管理物件

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

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

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

取得資訊

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

列出受管理物件

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

以下指令列出所有物件。

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

以下指令列出類型 `queue` 的所有物件。

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

有關單一物件的資訊

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

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

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

更新受管理物件

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

此指令變更表示主題連線工廠受管理物件的屬性：

```
imgobjmgr update
-t tf
-l "cn=MyTCF"
```

```
-o imgReconnectAttempts=3
-j "java.naming.factory.initial=
    com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=
    ldap://mydomain.com:389/o=img"
-j "java.naming.security.principal=
    uid=fooUser, ou=People, o=img"
-j "java.naming.security.credentials=fooPasswd"
-j "java.naming.security.authentication=simple"
```

更新受管理物件

管理安全性

此章介紹如何執行與安全性相關的工作。這些工作包括認證、授權和加密。

認證使用者 您負責維護使用者儲存庫中的使用者、使用者群組以及密碼清單。您可以在每個代理程式實例使用不同的使用者儲存庫。本章的第一部分介紹如何建立、寫入和管理使用者儲存庫。如需 Message Queue 安全性的介紹，請參閱第 60 頁的「安全性管理程式」。

授權使用者：存取控制特性檔案 您負責編輯存取控制特性檔案，該檔案將每個使用者對代理程式作業的存取權，對映到使用者的名稱或群組成員身份。您可以在每個代理程式實例，使用不同的存取控制特性檔案。本章的第二部分介紹如何自訂此檔案。

加密：使用基於 SSL 的服務 (企業版) 使用基於安全套接層 (SSL) 標準的連接服務，可讓您加密在用戶端與代理程式之間傳送的訊息。如需 Message Queue 如何處理加密的介紹，請參閱第 62 頁的「加密 (企業版)」。本章的最後一部分介紹如何設定基於 SSL 的連接服務，並提供有關使用 SSL 的其他資訊。

對於代理程式安全存取 SSL 密鑰儲存、LDAP 使用者儲存庫或 JDBC 相容永久性倉庫所需的密碼，有三種方法可提供此類密碼：

- 透過啓動代理程式時讓系統提示您輸入密碼
- 透過啓動代理程式時將密碼作為指令行選項輸入 (請參閱第 122 頁的「啓動代理程式」與第 124 頁的表 5-2)
- 透過啓動代理程式時將密碼儲存在系統可存取的密碼檔案中 (請參閱第 204 頁的「使用密碼檔案」)

認證使用者

當使用者嘗試連接到代理程式時，此代理程式將透過檢查提供的名稱和密碼認證此使用者，如果名稱和密碼，與系統配置各代理程式要參考之代理程式特定使用者儲存庫中的名稱和密碼相符，則代理程式承認此連接。此儲存庫有兩種類型：

- Message Queue 隨附的文字檔案儲存庫

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

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

- LDAP 伺服器

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

如果您要使用 LDAP 使用者儲存庫，則將需要使用 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 190 頁的「[將 LDAP 伺服器用於使用者儲存庫](#)」。

使用文字檔案使用者儲存庫

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

建立使用者儲存庫

文字檔案使用者儲存庫適用於特定實例。建立的預設使用者儲存庫 (命名為 `passwd`)，用於您啟動的每個代理程式實例。此使用者儲存庫位於以代理程式實例名稱 (`instanceName`) 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 資料的位置](#)」)：

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

此儲存庫隨附兩個項目 (列)，如以下表 8-1 中所述。

表 8-1 使用者儲存庫中的初始項目

使用者名稱	密碼	群組	狀態
admin	admin	admin	作用中
guest	guest	anonymous	作用中

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

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

初始 admin 使用者項目可讓您使用 imqcmd 指令和預設的 admin 使用者名稱和密碼來管理代理程式實例。我們建議您更新此初始項目來變更密碼 (請參閱第 189 頁的「變更預設管理員密碼」)。

以下各節介紹如何寫入和管理文字檔案使用者儲存庫。

使用者管理員公用程式 (imqusermgr)

使用者管理員公用程式 (imqusermgr) 可讓您編輯或寫入文字檔案使用者儲存庫。

本節描述基本的 imqusermgr 指令語法、提供子指令的清單，並總結 imqusermgr 指令選項。後續各節介紹如何使用 imqusermgr 子指令完成特定的工作。

使用 imqusermgr 前，請注意以下事項：

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

imqusermgr 指令的語法

imqusermgr 指定的一般語法如下：

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

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

```
imqusermgr list -v
```

imqusermgr 子指令

表 8-2 列出了 `imqusermgr` 子指令。

表 8-2 `imqusermgr` 子指令

子指令	說明
<code>add [-i instanceName] -u userName -p passwd [-g group] [-s]</code>	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。
<code>delete [-i instanceName] -u userName [-s] [-f]</code>	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。
<code>list [-i instanceName] [-u userName]</code>	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。
<code>update [-i instanceName] -u userName -p passwd [-a state] [-s] [-f]</code>	更新指定的 (或預設的) 代理程式實例儲存庫中，指定之使用者的密碼和 / 或狀態。

備註 以下各節中的範例均假設為預設的代理程式實例。

imqusermgr 指令選項摘要

表 8-3 列出了 `imqusermgr` 指令的選項。

表 8-3 `imqusermgr` 選項

選項	說明
<code>-a active_state</code>	指定使用者的狀態是否應為作用中 (<code>true/false</code>)。值 <code>true</code> 表示狀態為作用中。這是預設值。
<code>-f</code>	執行動作而無需使用者確認
<code>-h</code>	顯示用法說明。不執行指令行上的其他任何指令。
<code>-i instanceName</code>	指定代理程式實例使用者儲存庫套用的指令。如果未指定，則預設值假設為 <code>instanceName</code> 和 <code>imqbroker</code> 。
<code>-p passwd</code>	指定使用者密碼。
<code>-g group</code>	指定使用者群組。有效值為 <code>admin</code> 、 <code>user</code> 和 <code>anonymous</code> 。

表 8-3 imqusermgr 選項 (續)

選項	說明
-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

將使用者項目新增到用於代理程式實例的使用者儲存庫時，您可以為此使用者指定預先定義的三個群組之一：admin、user 或 anonymous。如果未指定群組，則指定預設群組 user。

- **admin 群組** 用於代理程式管理員。依預設，已指定此群組的使用者可以配置和管理代理程式。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會存取已在 user 群組中認證的代理程式。依預設，此群組中的應用程式用戶端使用者可以產生訊息到所有主題與佇列，以及使用來自所有主題與佇列的訊息，或者瀏覽任何佇列中的訊息。
- **anonymous 群組** 用於 Message Queue 用戶端，這些用戶端不想使用代理程式知道的使用者名稱 (可能因為用戶端應用程式不知道要使用的實際使用者名稱)。這對於大多數 FTP 伺服器中存在的匿名帳號均類似。您一次僅可將一個使用者指定給 anonymous 群組。預計您將透過存取控制限制此群組的存取權限 (與 user 群組相比較)，或者您將在部署時從此群組中移除使用者。

為了變更使用者的群組，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。

您可以指定定義群組成員可執行之作業的存取規則。如需更多資訊，請參閱第 192 頁的「授權使用者：存取控制特性檔案」。

狀態

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

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

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

使用者名稱與密碼的格式

使用者名稱與密碼必須遵循以下原則：

- 使用者名稱不能包含表 8-4 中所列的字元。

表 8-4 無效的使用者名稱與密碼字元

字元	說明
*	星號
,	逗號
:	冒號

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

加入和管理使用者儲存庫

使用 `add` 子指令可新增使用者至儲存庫。例如，以下指定可將使用者 `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 `alladin`：

```
imqusermgr update -u Katharine -p alladin
```

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  作用中狀態
-----
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  作用中狀態
-----
admin      admin  true
guest      anonymous true
isa        admin  true
testuser1  user   true
testuser2  user   true
testuser3  user   true
testuser4  user   false
testuser5  user   false
```

變更預設管理員密碼

為了安全性，您應將 `admin` 的預設密碼變更為只有您自己知道的密碼。您需要使用 `imqusermgr` 工具來完成此作業。

以下指令可將 `mybroker` 代理程式實例的密碼變更為 `grandpoobah`。

```
imqusermgr update -i 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，以下指令應生效：

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

使用舊的密碼應會失敗。

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

如果您要將 LDAP 伺服器用於使用者儲存庫，則必須在實例配置檔案中設定某些代理程式特性。當使用者嘗試連接到代理程式實例或執行某些訊息作業時，這些特性可讓代理程式實例查詢 LDAP 伺服器，以獲得使用者和群組的相關資訊。實例配置檔案 (config.properties) 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 資料的位置」)：

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

► 若要編輯配置檔案以使用 LDAP 伺服器

1. 透過設定以下特性指定您要使用 LDAP 使用者儲存庫：

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

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

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

3. 您還必須設定可控制 LDAP 存取的代理程式特性。這些儲存在代理程式的實例配置文件中的特性，在表 8-5 中有描述。Message Queue 會使用 JNDI API 與 LDAP 目錄伺服器進行通訊。請參閱 JNDI 說明文件，以獲得這些特性中所參考之語法和術語的更多相關資訊。Message Queue 會 Sun JNDI LDAP 供應商和簡單認證。

Message Queue 支援 LDAP 認證防故障備：您可以指定用於嘗試認證的 LDAP 目錄伺服器清單 (請參閱表 8-5 中的 `imq.user.repos.ldap.server` 特性)。

表 8-5 與 LDAP 相關的特性

特性	說明
imq.user_repository. ldap.server	用於 LDAP 伺服器的 <i>host:port</i> 。 <i>host</i> 可指定執行目錄伺服器之主機的完全合格的 DNS 名稱，而 <i>port</i> 指定用於通訊之目錄伺服器的連接埠號。若要指定防故障備用伺服器的清單，請使用下列語法： <i>host1:port1 ldap://host2:port2 ldap://host3:port3...</i> 清單中的項目以空格分開。請注意，第一個防故障備用伺服器後的每個防故障備用伺服器位址皆以 <i>ldap</i> 開頭。
imq.user_repository. ldap.principal	識別名稱，代理程式將使用此名稱連結至目錄伺服器以進行搜尋。如果目錄伺服器允許匿名搜尋，則無需為此特性指定值。
imq.user_repository. ldap.password	與代理程式所用識別名稱相關聯的密碼。此特性僅能在密碼檔案中指定（請參閱第 204 頁的「使用密碼檔案」）。 有數種方法可以提供密碼。最安全的方法是讓代理程式提示您提供密碼。較不安全的方法是使用密碼檔案，並讀取保護密碼檔案。最不安全的方法是使用以下指令行選項指定密碼： <i>imqbrokerd -ldappassword</i> 。 如果目錄伺服器允許匿名搜尋，則無需密碼。
imq.user_repository. ldap.base	用於使用者項目的目錄庫。
imq.user_repository. ldap.uidattr	供應程式特定的屬性識別碼，其值可唯一識別使用者。例如： <i>uid</i> 和 <i>cn</i> 等。
imq.user_repository. ldap.usrfilter	JNDI 搜尋過濾器（表示為邏輯表示式的搜尋查詢）。透過為使用者指定搜尋過濾器，代理程式可以縮小搜尋範圍以提高效率。如需更多資訊，請參閱位於以下位置的 JNDI 指導 http://java.sun.com/products/jndi/tutorial 。 此特性可不必設定。
imq.user_repository. ldap.grpsearch	指定您是否要啟用群組搜尋的布林。請參閱您的 LDAP 供應商提供的說明文件，以確定您是否可以將使用者與群組相關聯。 請注意， Message Queue 不支援巢式群組。 預設值： <i>false</i>
imq.user_repository. ldap.grpbased	用於群組項目的目錄庫。
imq.user_repository. ldap.gidattr	供應程式特定的屬性識別碼，其值為群組名稱。
imq.user_repository. ldap.memattr	群組項目中的屬性識別碼，其值為群組成員的識別名稱。

表 8-5 與 LDAP 相關的特性 (續)

特性	說明
imq.user_repository. ldap.grpfilter	JNDI 搜尋過濾器 (表示為邏輯表示式的搜尋查詢)。透過為群組指定搜尋過濾器，代理程式可以縮小搜尋範圍以便提高效率。如需更多資訊，請參閱位於以下位置的 JNDI 教學。 http://java.sun.com/products/jndi/tutorial 此特性可不必設定。
imq.user_repository. ldap.timeout	指定搜尋時間限制的整數 (以秒為單位)。依預設可設定為 180 秒。
imq.user_repository. ldap.ssl.enabled	指定代理程式與 LDAP 通訊時是否應使用 SSL 協定的布林。依預設可設定為 false。

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

- 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 192 頁的「[授權使用者：存取控制特性檔案](#)」。
- 如果在連接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`。

授權使用者： 存取控制特性檔案

連接至代理程式實例之後，使用者可能會產生訊息、使用目標上的訊息，或瀏覽佇列目標上的訊息。當使用者嘗試上述作業時，代理程式會檢查代理程式特定的 *存取控制特性檔案* (ACL 檔案)，查看該使用者是否已被授權執行此作業。ACL 檔案包含的規則可指定特定使用者 (或使用者群組) 已被授權執行哪些作業。依預設，允許所有認證使用者在任何目標上產生和使用訊息。您可以編輯 ACL 檔案，以限制某些使用者和群組執行這些作業。

無論使用者資訊放置在文字檔案使用者儲存庫 (請參閱第 184 頁的「使用文字檔案使用者儲存庫」) 或 LDAP 使用者儲存庫 (請參閱第 190 頁的「將 LDAP 伺服器用於使用者儲存庫」) 中，均可單獨使用 ACL 檔案。

建立存取控制特性檔案

ACL 檔案為特定實例。建立的預設檔案 (命名為 `accesscontrol.properties`)，用於您啟動的每個代理程式實例。ACL 特性檔案位於以代理程式名稱 (`instanceName`) 作為辨別依據的目錄中，且代理程式與 ACL 檔案相互關聯 (請參閱附錄 A「Message Queue 資料的位置」)：

```
.../instances/instanceName/etc/accesscontrol.properties
```

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

- 連接存取控制
- 目標存取控制
- 目標自動建立存取控制

`version` 特性定義 ACL 特性檔案的版本；您不能變更此項目。

```
version=JMQFileAccessControlModel/100
```

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

存取規則語法

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 (如目標和連接服務) 擁有何種存取權。存取控制由一條規則或一組規則表示，每條規則用一個 Java 特性表示：

這些規則的基本語法如下：

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

表 8-6 描述了語法規則的元素。

表 8-6 存取規則的語法元素

元素	說明
<i>resourceType</i>	為以下之一：connection、queue 或 topic。
<i>resourceVariant</i>	<i>resourceType</i> 指定的類型實例。例如，myQueue。萬用字元符號 (*) 可用於表示所有連接服務類型或所有目標。
<i>operation</i>	值取決於要制定的存取規則種類。
<i>access</i>	為以下之一：allow 或 deny。
<i>principalType</i>	為以下之一：user 或 group。如需更多資訊，請參閱第 187 頁的「群組」。
<i>principals</i>	可以擁有在規則左側指定的存取權的使用者。如果 <i>principalType</i> 為 user，則此項可以為個別使用者或使用者清單（以逗號分隔）；如果 <i>principalType</i> 為 group，則此項可以為單一群組或群組清單（以逗號分隔的清單）。萬用字元符號 (*) 可用於表示所有使用者或所有群組。

以下為存取規則的一些範例：

- 以下規則表示所有使用者均可以將訊息傳送至名為 q1 的佇列。
queue.q1.produce.allow.user=*
- 以下規則表示任何使用者均可以將訊息傳送至任何佇列。
queue.*.produce.allow.user=*

備註 若要指定非 ASCII 的使用者、群組或目標名稱，您必須使用統一碼退出符號 (\uXXXX) 表示法。如果您已經編輯並儲存具有非 ASCII 編碼名稱的 ACL 檔案，則可以使用 Java native2ascii 工具將其轉換為 ASCII。如需更加詳細的資訊，請參閱 <http://java.sun.com/j2se/1.4/docs/guide/intl/faq.html>

許可權運算

當運算一系列規則所包含的許可權時，可套用以下原則：

- 特定存取規則置換一般存取規則。套用以下兩條規則之後，所有使用者均可以將訊息傳送至所有佇列，但是 Bob 卻無法將訊息傳送至 tq1。
queue.*.produce.allow.user=*

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

- 賦予明確 *principal* 的存取權置換賦予 ** principal* 的存取權。以下規則拒絕 Bob 向 tq1 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* 規則置換用於群組的相應 ** principal* 規則。例如，以下兩條規則允許所有認證使用者將訊息傳送至 tq1。

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

```
queue.tq1.produce.deny.group=*
```

- 授予使用者的存取權會置換授予使用者群組的存取權。在以下範例中，如果 Bob 為 User 成員，則他將被拒絕向 tq1 產生訊息，但是 User 的所有其他成員均可以向 tq1 產生訊息。

```
queue.tq1.produce.allow.group=User
```

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

- 任何未透過存取規則明確授予的存取許可權，都會被默認拒絕。例如，如果 ACL 檔案不包含存取規則，則所有使用者均被拒絕所有作業。
- 相同使用者或群組的拒絕和允許許可權會相互抵銷。例如，以下兩條規則導致 Bob 無法瀏覽 q1：

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

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

以下兩條規則導致群組 User 無法使用 q5 上的訊息。

```
queue.q5.consume.allow.group=User
```

```
queue.q5.consume.deny.group=User
```

- 當存在多條相同的左側規則時，只有最後一項生效。

連接存取控制

ACL 特性檔案中的連接存取控制部分，包含用於代理程式連接服務的存取控制規則。連接存取控制規則的語法如下：

```
connection.resourceVariant.access.principalType = principals
```

resourceVariant 定義有兩個值：NORMAL 與 ADMIN。依預設，所有使用者對 NORMAL 類型連接服務擁有存取權，但是只有那些群組為 admin 的使用者才對 ADMIN 類型連接服務擁有存取權。

您可以編輯連接存取控制規則，以限制使用者的連接存取權限。例如，以下規則拒絕 Bob 存取 NORMAL，但是允許其他所有使用者進行存取：

```
connection.NORMAL.deny.user=Bob
```

```
connection.NORMAL.allow.user=*
```

您可以使用星號 (*) 字元來指定所有認證使用者或群組。

您不能建立自己的服務類型；您必須將自己限制於由常數 NORMAL 與 ADMIN 所指定的預先定義類型之內。

目標存取控制

存取控制特性檔案的目標存取控制部分，包含基於目標的存取控制規則。這些規則確定何人（使用者 / 群組）可以在何處（目標）做什麼（作業）。這些規則所規定的存取權類型包括：將訊息傳送至佇列、將訊息出版至主題、接收來自佇列的訊息、訂閱主題以及瀏覽佇列中的訊息。

依預設，任何使用者或群組對任何目標擁有所有類型的存取權。您可以新增多條特定的目標存取規則，或編輯預設規則。本節的其餘部分介紹目標存取規則的語法，您必須理解此語法以編寫自己的規則。

目標規則的語法如下：

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

表 8-7 描述了這些元素：

表 8-7 目標存取控制規則的元素

元件	說明
<i>resourceType</i>	必須為 <code>queue</code> 或 <code>topic</code> 之一。
<i>resourceVariant</i>	目標名稱或所有目標 (*), 表示所有佇列或所有主題。
<i>operation</i>	必須為 <code>produce</code> 、 <code>consume</code> 或 <code>browse</code> 之一。
<i>access</i>	必須為 <code>allow</code> 或 <code>deny</code> 之一。
<i>principalType</i>	必須為 <code>user</code> 或 <code>group</code> 之一。

存取權可以賦予一個或多個使用者和 / 或一個或多個群組。

以下範例說明了不同種類的目標存取控制規則：

- 允許所有使用者將訊息傳送至任何佇列目標。
`queue.*.produce.allow.user=*`
- 拒絕群組 `user` 的任何成員訂閱主題 `Admissions`。
`topic.Admissions.consume.deny.group=user`

目標自動建立存取控制

ACL 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目標的存取規則。

當使用者在並不存在的目標上建立產生者或使用者時，如果代理程式的自動建立特性已啟用且實體目標不存在，則代理程式將建立此目標。

依預設，任何使用者或群組均有權讓代理程式自動建立目標。此權限由以下規則指定：

```
queue.create.allow.user=*
topic.create.allow.user=*
```

您可以編輯 ACL 檔案以限制此類型的存取權。

目標自動建立存取規則的一般語法如下：

```
resourceType.create.access.principalType = principals
```

其中 *resourceType* 為 *queue* 或 *topic*。

例如，以下規則可讓代理程式為除 *Snoopy* 之外的所有使用者自動建立主題目標。

```
topic.create.allow.user=*
topic.create.deny.user=Snoopy
```

請注意，目標自動建立規則的效果必須與目標存取規則的效果一致。例如，如果您 1) 變更目標存取規則，以禁止任何使用者將訊息傳送至目標，但是 2) 啟用目標的自動建立功能，如果此目標不存在，則代理程式將會建立此目標，但不會將訊息發送至此目標。

加密：使用基於 SSL 的服務 (企業版)

Message Queue 企業版支援基於安全套接層 (SSL) 標準的連接服務：透過 TCP/IP (*ssljms* 與 *ssladmin*) 和透過 HTTP (*httpsjms*)。這些基於 SSL 的連接服務允許加密在用戶端與代理程式之間傳送的訊息。目前的 Message Queue 版次支援基於自身簽名伺服器憑證的 SSL 加密。

若要使用基於 SSL 的連接服務，您需要使用鍵值工具公用程式 (*imqkeytool*) 產生私密密鑰 / 公開密鑰對。此公用程式將公開密鑰內嵌於自身簽名的憑證中，此憑證被傳送至請求與代理程式連接的用戶端，此用戶端使用此憑證設定加密連接。

雖然 Message Queue 的基於 SSL 的連接服務在概念上相似，但是在設定方法上卻有些差異。因此，以下各節分別說明透過 TCP/IP 和 HTTP 的安全連接。

透過 TCP/IP 設定基於 SSL 的服務

三種基於 SSL 的連接服務可提供透過 TCP/IP 的直接且安全的連接。

ssljms 此連接服務用於在用戶端與代理程式之間，透過安全且加密的連接發送訊息。

ssladmin 此連接服務用於在 Message Queue 指令公用程式 (*imqcmd*) 這種命令行管理工具與代理程式之間建立安全且加密的連接。管理主控台 (*imqadmin*) 不支援安全連接。

叢集 此連接服務用於叢集中代理程式之間，透過安全且加密的連接發送訊息 (請參閱第 130 頁的「安全代理程式互通連接」)。

► 若要設定基於 SSL 的連接服務

1. 產生自身簽名的憑證。
2. 啟用 `ssljms` 連接服務、`ssladmin` 連接服務，或代理程式中的叢集連接服務。
3. 啟動代理程式。
4. 設定並執行用戶端 (僅適用於 `ssljms` 連接服務)。

除步驟 4 (配置並執行用戶端) 之外，設定 `ssljms` 連接服務的程序與設定 `ssladmin` 連接服務的程序相同。

以下各節較詳細地說明每個步驟。

步驟 1. 產生自身簽名的憑證

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，僅使用自身簽名的憑證即可實施 SSL。

執行 `imqkeytool` 指令，以便為代理程式產生自身簽名的憑證。同樣的憑證可以用於 `ssljms` 連接服務、`ssladmin` 連接服務或叢集連接服務。在命令提示字元下輸入以下內容：

```
imqkeytool -broker
```

此公用程式將提示您提供它所需的資訊。(在 Unix® 系統上，為了擁有建立密鑰儲存的許可權，您可能需要以超級使用者 (root) 的身份執行 `imqkeytool`。)

首先，`imqkeytool` 會提示您提供密鑰儲存密碼以及某些組織資訊，然後提示您確認。接收到確認之後，它會在產生鍵對時暫停。然後它會要求您提供密碼，以鎖定特定的鍵對 (鍵密碼)；您應按 [Return] 以確認此提示：這會使鍵密碼與密鑰儲存密碼相同。

備註 請記住您所提供的密碼，稍後您需要將此密碼提供給代理程式 (當您啟動它時)，以便代理程式可開啓密鑰儲存。您還可以將密鑰儲存密碼儲存在密碼檔案中 (請參閱第 204 頁的「使用密碼檔案」)。

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

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

Message Queue 密鑰儲存的可配置特性如表 8-8 中所示。(如需有關配置這些特性的說明，請參閱第 5 章「啟動與配置代理程式」。)

表 8-8 密鑰儲存特性

特性名稱	說明
imq.keystore.file.dirpath	用於基於 SSL 的服務：指定包含密鑰儲存檔案之目錄的路徑。預設值：請參閱附錄 A「Message Queue 資料的位置」。
imq.keystore.file.name	用於基於 SSL 的服務：指定密鑰儲存檔案的名稱。預設值：keystore
imq.keystore.password	用於基於 SSL 的服務：指定密鑰儲存密碼。此特性僅能在密碼檔案中指定 (請參閱第 204 頁的「使用密碼檔案」)。有數種方法可以提供密碼。最安全的方法是讓代理程式提示您提供密碼。較不安全的方法是使用密碼檔案，並讀取保護密碼檔案。最不安全的方法是使用以下指令行選項指定密碼：imqbrokerd -ldappassword。

您可能需要重新產生鍵對以解決某些問題；例如：

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

可能導致此異常的原因為，您提供的鍵密碼與您在第 199 頁的「步驟 1. 產生自身簽名的憑證」中產生自身簽名憑證時的密鑰儲存密碼不同。

► 若要重新產生鍵對

1. 移除代理程式的密鑰儲存，如附錄 A「Message Queue 資料的位置」中所示
2. 重新執行 imqkeytool 以產生鍵對，如第 199 頁的「步驟 1. 產生自身簽名的憑證」中所述。

步驟 2. 啓用代理程式中基於 SSL 的服務

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

備註 請使用 imq.cluster.transport 特性啓用基於 SSL 的 cluster (叢集) 連接服務，而非使用 imq.service.activelist 特性。請參閱第 130 頁的「安全代理程式互通連接」。

► 若要啓用代理程式中基於 SSL 的服務

1. 開啓代理程式的實例配置檔案。

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

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

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

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

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

步驟 3. 啓動代理程式

啓動代理程式，提供密鑰儲存密碼。您可以以下任何一種方法提供密碼：

- 當代理程式啓動時讓它提示您提供密碼

```
imqbrokerd
Please enter Keystore password:mypassword
```

- 使用 `imqbrokerd` 指令的 `-password` 選項：

```
imqbrokerd -password mypassword
```

- 將密碼放在密碼檔案中 (請參閱第 204 頁的「[使用密碼檔案](#)」)，代理程式啓動時將存取此檔案。您必須先設定以下代理程式配置特性 (請參閱第 117 頁的「[編輯實例配置檔案](#)」)：

```
imq.passfile.enabled=true
```

一旦設定此特性，您便可以用以下兩種方法之一存取密碼檔案：

- 將密碼檔案的位置傳送給 `imqbrokerd` 指令：

```
imqbrokerd -passfile /tmp/mypassfile
```

- 啓動不帶 `-passfile` 選項的代理程式，但使用以下兩個代理程式配置特性指定密碼檔案的位置：

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

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

如需與密碼檔案相關的代理程式特性的清單，請參閱第 62 頁的表 2-6。

當您啟動具有 SSL 的代理程式或用戶端時，您可能會注意到它消耗了很多 CPU 週期，長達幾秒鐘。這是因為 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. 配置並執行基於 SSL 的用戶端

最後，您需要將用戶端配置為使用安全連接服務。依據您使用的連接服務，有兩種類型的用戶端：使用 `ssljms` 的應用程式用戶端，和使用 `ssladmin` 的 Message Queue 管理用戶端 (例如 `imqcmd`)。這兩種用戶端將在以下各節中分別描述。

應用程式用戶端

您必須確保用戶端在其類別路徑中擁有必要的安全套接延伸 (JSSE) JAR 檔案，而且您需要告知用戶端使用 `ssljms` 連接服務。

1. 如果您的用戶端未使用 J2SDK1.4 (具有內建的 JSSE 與 JNDI 支援)，請確保用戶端在其類別路徑中擁有以下 JAR 檔案：

`jsse.jar`、`jnet.jar`、`jcrt.jar` 和 `jndi.jar`

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

`imq.jar` 和 `jms.jar`

3. 啟動用戶端並將其連接至代理程式的 `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` 可告知連接使用 SSL。

如需有關在用戶端應用程式中使用 `ssljms` 連接服務的更多資訊，請參閱「Message Queue Java 用戶端開發人員指南」中有關使用受管理物件的章節。

管理用戶端 (`imqcmd`)

您可以透過使用 `imqcmd` 時包括的 `-secure` 選項，建立安全管理連接 (請參閱第 140 頁的表 6-2)，例如：

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

其中 `adminName` 和 `adminPassword` 在 Message Queue 使用者儲存庫中為有效項目。(如果您是使用文字檔案儲存庫，請參閱第 189 頁的「變更預設管理員密碼」)。

列出連接服務是顯示 `ssladmin` 服務正在執行的一種方法，也是您可以成功建立安全 `admin` 連接的一種方法，如以下輸出中所示：

Listing all the services on the broker specified by:

Host	Primary Port
localhost	7676

Service Name	Port Number	Service State
admin	33984 (dynamic)	RUNNING
httpjms	-	UNKNOWN
httpsjms	-	UNKNOWN
jms	33983 (dynamic)	RUNNING
ssladmin	35988 (dynamic)	RUNNING
ssljms	dynamic	UNKNOWN

Successfully listed services.

透過 HTTP 設定基於 SSL 的服務

在此基於 SSL 的連接服務 (httpsjms) 中，用戶端和代理程式藉由 HTTPS 通道 Servlet 建立安全連接。附錄 C 第 273 頁的「HTTP/HTTPS 支援 (企業版)」中描述了 HTTPS 支援的架構和實施。

使用密碼檔案

當您想讓代理程式啟動而不提示您提供所需密碼，或不要求您將這些密碼作為 `imqbrokerd` 指令的選項提供時，您可以將所需密碼放在 *密碼檔案* 中。接著，可以在啟動代理程式時，使用 `-passfile` 選項指定密碼檔案：

```
imqbrokerd -passfile myPassfile
```

密碼檔案為包含密碼的簡單文字檔。此檔案未加密，因此不如手動提供密碼安全。然而，您可以設定此檔案的許可權，限制何人有權檢視它。密碼檔案上設定的許可權，需要提供啟動代理程式的使用者讀取權限。

密碼檔案可以包含 [表 8-9](#) 中所示密碼：

表 8-9 密碼檔案中的密碼

密碼	說明
<code>imq.keystore.password</code>	為基於 SSL 的服務指定密鑰儲存密碼 (請參閱 第 200 頁的表 8-8)。
<code>imq.user_repository.ldap.password</code>	指定與識別名稱相關聯的密碼，此識別名稱指定給代理程式以連結至已配置的 LDAP 使用者儲存庫 (請參閱 第 191 頁的表 8-5)。
<code>imq.persist.jdbc.password</code>	如有必要，則指定用於開啓資料庫連接的密碼 (請參閱 第 266 頁的表 B-1)。

可以在某個位置找到密碼檔案範例，其所在位置因作業系統而異，如 [附錄 A](#) 「[Message Queue 資料的位置](#)」中所示。

分析與調整訊息服務

本章涵蓋一些如何分析與調整 **Message Queue** 服務的主題，以最佳化您訊息傳送應用程式的效能。它包括以下主題：

- [關於效能](#)
- [影響效能的因素](#)
- [監視訊息伺服器](#)
- [排解效能問題](#)
- [調整您的配置以改善效能](#)

關於效能

效能調整程序

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人員和管理員共同的努力。

最佳化效能的程序以應用程式設計為起點，接著在部署應用程式後，調整訊息服務繼續。效能調整程序包括以下階段：

- 定義應用程式的效能需求
- 設計考慮影響效能因素的應用程式（特別是可靠性與效能間的平衡）
- 建立基本效能測量
- 調整或重新配置訊息服務以最佳化效能。

以上略述的程序通常是反覆的。應用程式部署期間，**Message Queue** 管理員會評估應用程式一般效能需求的訊息伺服器合適性。如果效能評定測試符合這些需求，管理員可以根據本章說明調整系統。但是，如果效能評定測試不符合效能需求，則可能需要重新設計應用程式或修改架構部署。

效能類型

一般而言，效能為訊息服務從產生者發送訊息到使用者之速度和效率的測量。但是，依您的需要，您可能需要注意幾個不同方面的效能。

連接負載 可支援訊息產生者、訊息使用者或運作之連接系統的數目。

訊息流量 每秒導入訊息傳送系統的訊息數目或訊息位元組。

延時 從訊息產生者發送特定訊息到訊息使用者所花的時間。

穩定性 訊息服務的整體可用性，或如何逐漸降低可用性以防大量負載或失敗。

效率 訊息發送的效率；與計算已使用訊息相關的訊息流量測量。

通常這些不同方面的效能是相互關聯的。如果訊息流量高，則表示訊息較不可能儲存於訊息伺服器中，因此，延時時間短（單一訊息可迅速發送）。但是，延時可取決於很多因素：通訊連結的速度、訊息伺服器處理速度和用戶端處理速度等。

無論如何，會有數個不同方面的效能。對您最重要的方面通常取決於特定應用程式的需求。

效能評定

效能評定為建立用於您訊息傳送應用程式與測量訊息流量之測試套件的程序，或用於此測試套件的其他方面效能。

例如，您可能會以數個生產型用戶端，使用數個連接、階段作業和訊息產生者來建立測試套件，並以某特定速率，將標準大小的永久性或非永久性訊息傳送到數個佇列或主題（所有皆取決於您訊息傳送應用程式設計）。同樣地，測試套件包括數個使用數個連接和階段作業的使用用戶端以及在測試套件目標中，以特定確認模式使用訊息的訊息使用者（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接執行緒用法、訊息儲存資料、訊息流量資料和其他相關度量。接著，您可以在對效能產生不良影響前，急速增加訊息產生的速率、訊息產生者的數目或其他變數。您可以達到的最大流量為您訊息服務配置的效能評定。

使用此效能評定，您可以修改部分您測試套件的特徵。藉由小心控制所有可能影響效能的因素（請參閱第 209 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會如何影響效能評定。例如，您可以將連接數目或訊息大小增加五倍或十倍，即可發現效能受到影響。

相反地，您可以將基於應用程式的因素維持一致，並以某種控制方法變更您的代理程式配置（例如，變更連接特性、執行緒儲存區特性、JVM 記憶體限制、限制行爲、內建與外掛持續性等），接著您可以注意到這些變更如何影響效能。

您應用程式的效能評定會提供資訊，這些資訊在您要藉由調整訊息服務，以增加已部署應用程式的效能時會有所幫助。效能評定可以更加準確地預測一個變更或一組變更的效果。

一般來說，應在受控制的測試環境中，於一段足以穩定您訊息服務的長時間內執行效能評定。（啟動時，Just-In-Time 編譯會將 Java 程式碼轉為機器程式碼，這會對效能造成不良影響。）

基本使用式樣

一旦部署並執行訊息傳送應用程式，建立基本使用式樣是很重要的。您必須知道尖峰需求何時出現並可以將需求數目化。例如，需求通常會因一般使用者的數目、活動級別、天數或前述所有因素而有變化。

若要建立基本使用式樣，您必須長時間監視您的訊息伺服器，查看像是連接數目、儲存於代理程式中（或特定目標中）的訊息數目、流入或和流出代理程式（或特定目標）的訊息流量、作用中使用者數目等等的資料。您還可以使用度量資料中提供的平均與尖峰值。

檢查這些出乎設計意料之外的基本度量是非常重要的。經由此動作，您會檢查用戶端程式碼是否正常運作：例如，連接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。

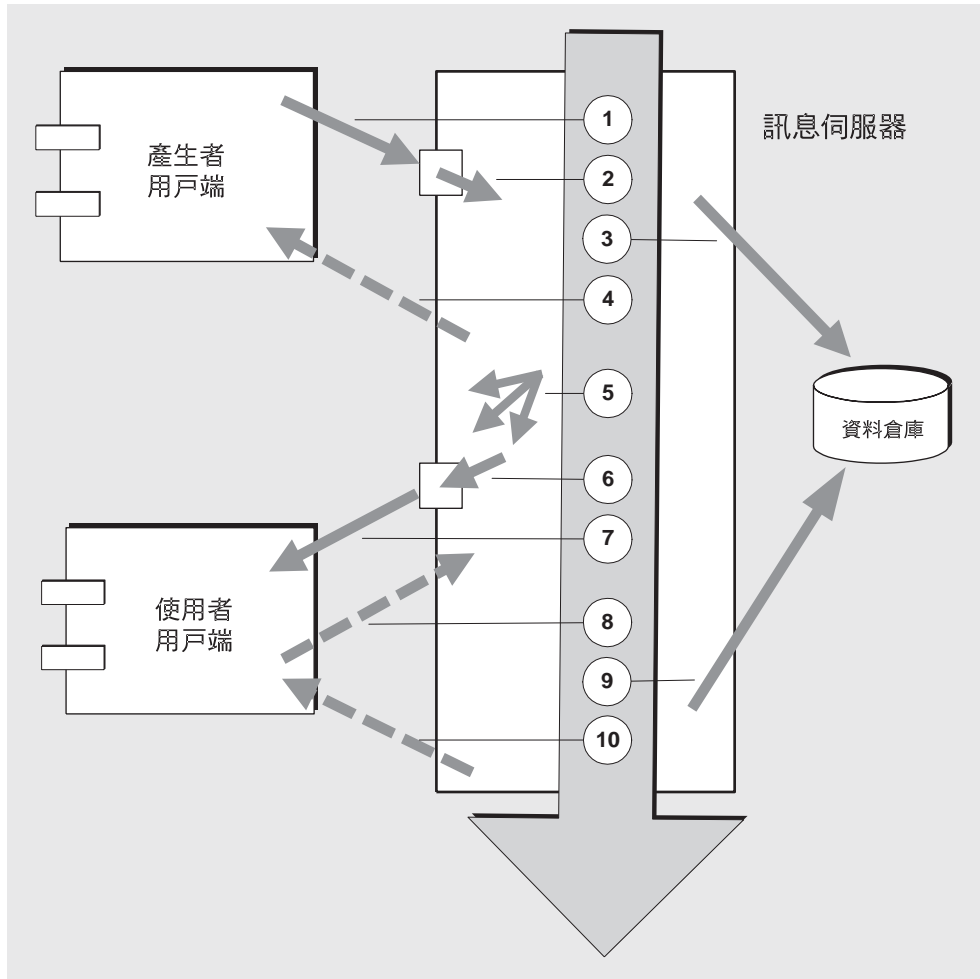
基本使用式樣可幫助您判斷如何將系統調整到最佳化效能。舉例來說，如果某個目標的使用度較其他目標高，那麼您可能要將此目標的訊息記憶體限制設定的比其他目標的限制高，或者要相應地調整限制行爲。如果所需的連接數目明顯大於最大執行緒儲存區容量允許的大小，那麼您可能要增加執行緒儲存區大小，或調整共用執行緒模型。如果尖峰訊息流量遠遠大於平均流量，那麼在記憶體不足時，可能會影響您使用的限制行爲。

一般而言，您對使用式樣的瞭解越多，您調整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發送程序中各個步驟所花費的時間。以下顯示的步驟用於永久性、可靠且已發送之訊息的情況下。下圖描述這些步驟。

圖 9-1 透過 Message Queue 服務的訊息發送



1. 這些訊息是由生產型用戶端發送到訊息伺服器。

2. 訊息伺服器在訊息中進行讀取
3. 訊息置於永久性儲存體中，以確保其可靠性
4. 訊息伺服器會確認訊息的接收，以確保其可靠性
5. 訊息伺服器會判斷訊息的路由
6. 訊息伺服器會在訊息外等候
7. 這些訊息是由訊息伺服器發送至使用用戶端
8. 使用用戶端會確認訊息的接收，以確保其可靠性
9. 訊息伺服器會處理用戶端確認，以確保其可靠性
10. 訊息伺服器會確認是否已處理用戶端確認

由於這些步驟是連續性的，所以在從生產型用戶端發送訊息至使用用戶端時，這其中任何一個步驟都是可能遇到的瓶頸。大部分的步驟取決於訊息傳送系統的實體特徵：網路頻寬、電腦處理速度、訊息伺服器架構等。但是，還有些步驟是取決於訊息傳送應用程式的特徵和要求的可靠性級別。

以下小節說明應用程式設計因素與訊息傳送系統因素對效能的影響。當在訊息傳送中應用程式設計和訊息傳送系統因素相互操作時，將單獨的考慮每個種類。

影響效能的應用程式設計因素

應用程式設計決策會對整體訊息傳送效能產生顯著的效果。

影響效能最重要的因素為影響訊息發送可靠性的因素。這些因素如下：

- 發送模式 (永久性 / 非永久性訊息)
- 異動使用
- 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 (訊息過濾)
- 訊息容量
- 訊息內文類型

以下各節描述影響訊息傳送效能的各個因素。一般來說，要在效能與可靠性間取得平衡：即提昇可靠性的因素往往會降低效能。

以下表格顯示各種應用程式設計因素通常如何影響訊息傳送效能。表格顯示兩個方案 - 高可靠性和低效能方案、高效能和低可靠性方案 - 且顯示特定應用程式設計因素的選擇。在這兩個方案中，有許多影響可靠性和效能的選擇和平衡。

表 9-1 高可靠性和高效能方案的比較

應用程式設計因素	高可靠性 低效能方案	高效能 低可靠性方案
發送模式	永久性訊息	非永久性訊息
異動使用	異動的階段作業	無異動
確認模式	AUTO_ACKNOWLEDGE 或 CLIENT_ACKNOWLEDGE	DUPS_OK_ACKNOWLEDGE
長期 / 非長期訂閱	長期訂閱	非長期訂閱
選擇器的使用	訊息過濾	無訊息過濾
訊息容量	小訊息	大訊息
訊息內文類型	複雜的內文類型	簡單的內文類型

備註 以下各圖為使用基於檔案的持續性，在雙 CPU、1002 Mhz、Solaris 8 系統上產生的效能資料。效能測試首先會預熱 Message Queue 代理程式，以便 Just-In-Time 編譯器最佳化系統和預備的永久性資料庫。

預熱代理程式後，會建立一個產生者和使用者，並以 30 秒的時間產生訊息。會記錄使用者接收所有產生訊息的所需時間，且計算流量速率（每秒幾個訊息）。會在應用程式設計因素的不同組合，重複使用此方案，如表 9-1 所示。

發送模式（永久性 / 非永久性訊息）

如第 43 頁的「可靠的訊息傳送」中所述，永久性訊息可確保訊息伺服器故障情況下的訊息發送。代理程式會在所有預定使用者確認他們已使用訊息前，將訊息儲存在永久性儲存體中。

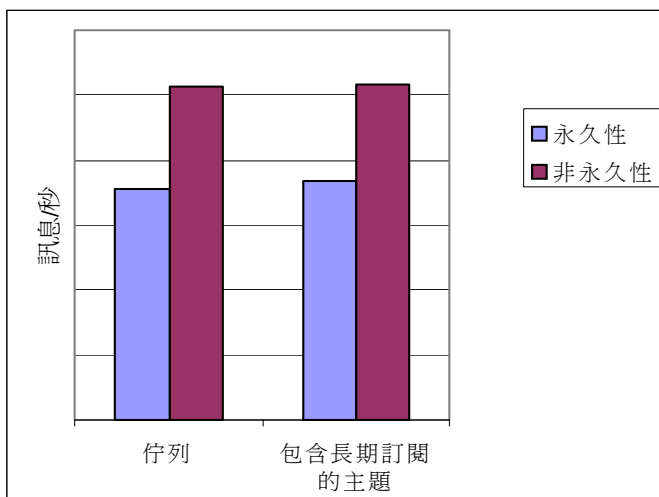
永久性訊息的代理程式處理速度會比非永久性的訊息處理速度慢，原因如下：

- 代理程式必須可靠地儲存永久性訊息，如此一來，代理程式故障時才不會遺失永久性訊息。

- 代理程式必須確認每個永久性訊息的接收。一旦產生訊息的方法未傳回異常，則需確保至代理程式的發送。
- 根據用戶端的確認模式，代理程式可能需要確認使用用戶端對永久性訊息的確認。

永久性和非永久性模式間效能的差異性非常明顯。圖 9-2 比較兩個可靠發送情況下，永久性和非永久性訊息的流量：發送 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 9-2 發送模式的效能影響



異動使用

異動是所有異動階段作業中產生訊息和使用訊息的保證，兩種訊息的處理方法如下：處理作為一個單位、不處理（回轉）作為一個單位。

Message Queue 支援本地和分散式異動（請個別參閱「本地異動」和第 44 頁的「分散式異動」，以獲得更多資訊）。

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

- 附加資訊必須隨著每個產生訊息一起儲存。
- 在一些情況下，會儲存異動中的訊息，但通常是不會儲存這些訊息（例如，通常會刪除發送到未訂閱之主題目標的永久性訊息，但是在異動開始時，會無法使用有關訂閱的資訊）。

- 異動中的訊息使用和確認資訊，必須在異動確定時進行儲存和處理。

確認模式

JMS 訊息發送的可確保可靠性機制，是用戶端用來確認 Message Queue 訊息伺服器所發送到其自身的訊息使用（請參閱第 54 頁的「可靠發送：確認與異動」）。

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，則代理程式會重新發送訊息，並設定 JMSRedelivered 旗標。

對非異動階段作業而言，用戶端可以選擇三種確認模式中的其中一種模式，每一個模式皆擁有各自的效能特徵：

- `AUTO_ACKNOWLEDGE`。一旦使用者處理訊息後，系統會自動確認訊息。此模式可確保提供者發生故障後最多重新發送一次訊息。
- `CLIENT_ACKNOWLEDGE`。此應用程式會控制確認訊息的端點。確認先前的確認後，所有訊息會在該階段作業中進行處理。如果在處理確認組時，訊息伺服器發生故障，則可能會重新發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，可以重新發送多個訊息。

（使用 `CLIENT_ACKNOWLEDGE` 模式與使用異動相似，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。）

效能受到確認模式影響的原因如下：

- 在 `AUTO_ACKNOWLEDGE` 和 `CLIENT_ACKNOWLEDGE` 模式中，會要求代理程式與用戶端間的額外控制。額外控制訊息會新增額外處理耗用時間，且可干擾 JMS 有效負載訊息，因而造成處理延遲。
- 在 `AUTO_ACKNOWLEDGE` 和 `CLIENT_ACKNOWLEDGE` 模式中，用戶端必須在可使用其他訊息前，等待代理程式確認是否已處理用戶端的確認。（此用戶端確認可確保代理程式不會不慎重新發送這些訊息。）
- 必須以使用者接收之所有永久性訊息的確認資訊來更新 Message Queue 永久性儲存體，因此會降低效能。

長期與非長期訂閱

如第 42 頁的「出版 / 訂閱（主題目標）」中所述，至主題目標的用戶分為兩個種類，使用長期或非長期訂閱：

長期訂閱可提高可靠性，但會降低流量，原因如下：

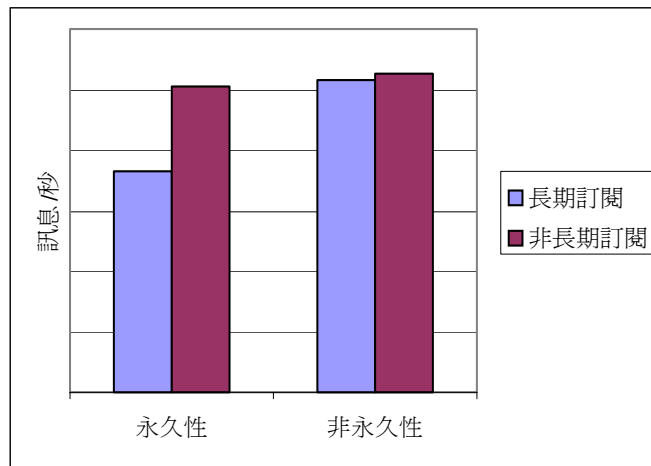
- Message Queue 訊息伺服器必須持續儲存指定到每個長期訂閱的訊息清單，如此一來，一旦訊息伺服器發生故障，即可在回復後使用此清單。

- 會持續儲存長期訂閱的永久性訊息，如此一來，一旦訊息伺服器發生故障，即可在回復後重新發送訊息，同時對應的使用者會變為作用中狀態。相反地，不會持續儲存非長期訂閱的永久性訊息（一旦訊息伺服器發生故障，會中斷對應使用者的連接，且不會重新發送訊息）。

圖 9-3 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 9-3 發現只有在永久性訊息的情況下，使用長期訂閱才會對效能產生明顯影響；如同上述說明，在此情況下發生影響是因為只會持續儲存用於長期訂閱的永久性訊息。

圖 9-3 訂閱類型的效能影響



選擇器的使用 (訊息過濾)

應用程式開發人員通常會將訊息組指向特定使用者。他們會藉由將每個訊息組指向唯一目標，或使用單一目標並為使用者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（請參閱第 37 頁的「[JMS 訊息結構](#)」）的訊息發送到特定使用者。例如，選擇器 `NumberOfOrders > 1` 僅會發送包含 `NumberOfOrders` 特性值 2 或更高值的訊息。

以選擇器註冊使用者會降低效能（與使用多個目標相比較的話），因為需要額外處理每個訊息。使用選擇器時必須剖析選擇器，如此一來，選擇器可與之後的訊息相符合。此外，在路由每個訊息時，必須擷取每個訊息的訊息特性並與選擇器進行比較。但是，使用選擇器可在訊息傳送應用程式中提供更多彈性。

訊息容量

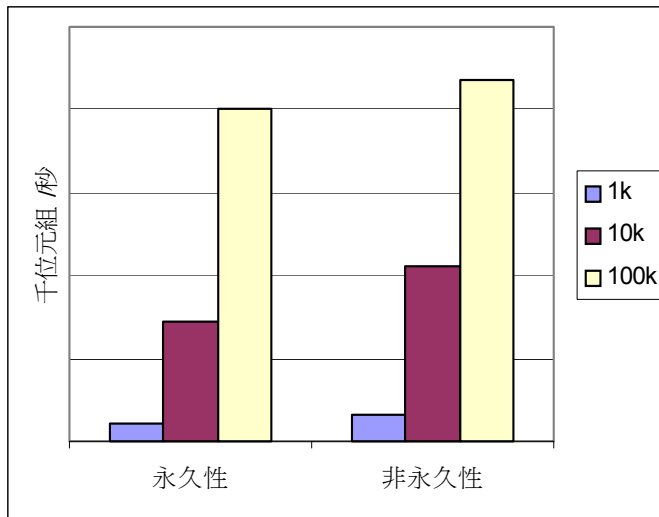
訊息容量會影響效能，因為必須將更多資料從生產型用戶端傳送到代理程式，再從代理程式傳送到使用用戶端，且必須儲存較大的永久性訊息。

但是，藉由將較小訊息批次到單一訊息，可以最小化個別訊息的路由和處理，以提供整體效能改善比率。在此情況下，會遺失個別訊息狀態的相關資訊。

圖 9-4 比較兩種情況下，1K、10K 和 100K 大小訊息每秒的流量（以千位元組為單位）：永久性與非永久性的訊息。在所有情況下，皆會將訊息傳送到佇列目標，並使用 AUTO_ACKNOWLEDGE 確認模式。

圖 9-4 顯示在兩種情況下，發送較大訊息的耗用時間會比發送較小訊息的耗用時間短。您也可以發現在 1K 與 10K 大小訊息中，非永久性訊息的效能改善比率比永久性訊息的效能改善比率高出將近 50%，但在 100K 大小訊息中並沒有這種情形，這可能是因為在此情況下，網路頻寬成爲訊息流量的瓶頸。

圖 9-4 訊息容量的效能影響



訊息內容類型

JMS 支援五種訊息內文類型，以下大致上依其複雜性順序顯示：

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

- `MapMessage`：包含名稱與值對的組合
- `ObjectMessage`：包含 Java 序列化物件

一般而言，依應用程式需求規定訊息類型時，更複雜的類型 (`MapMessage` 和 `ObjectMessage`) 會增加效能耗用 - 序列化和解序列化資料的效能。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發送的訊息服務影響。

以下各節介紹可影響效能的這種訊息服務因素。瞭解這些因素的影響是調整訊息服務大小、及診斷和解決已部署應用程式中可能會出現之效能瓶頸的關鍵。

影響 `Message Queue` 服務中效能的最重要因素如下所述：

- 硬體
- 作業系統
- Java 虛擬機器 (JVM)
- 連接
- 代理程式限制和行爲
- 訊息伺服器架構
- 資料倉庫效能
- 用戶端運行時間配置

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接數目會有極大影響。一般而言，在所有硬體皆同等的情況下，**Solaris** 系統通常會比 **Linux** 系統快，而 **Linux** 系統又比 **Windows** 系統快。

Java 虛擬機器 (JVM)

訊息伺服器為在主要 JVM 中執行且支援的 Java 程序。因此，JVM 處理是決定訊息伺服器路由和發送訊息之速度和效率的重要因素。

要特別注意的是，JVM 的記憶體資源管理是非常重要的。必須在 JVM 配置足夠的記憶體，以適應增加的記憶體負載。除此之外，JVM 會定期回收未使用的記憶體，且記憶體回收會延遲訊息處理。JVM 記憶體堆疊的容量越多，記憶體回收過程中可能發生的延遲時間越長。

連接

用戶端和代理程式間的連接數目和速度，會影響訊息伺服器處理訊息的數目和發送訊息的速度。

訊息伺服器連接限制

所有訊息伺服器的存取皆是透過連接。運作中連接數目的任何限制，皆會影響目前使用訊息伺服器之生產或使用用戶端的數目。

至訊息伺服器的連接數目通常會受到可用執行緒數目的限制。**Message Queue** 會使用執行緒儲存區管理程式，以便配置支援專用的執行緒模型或共用的執行緒模型（請參閱第 51 頁的「執行緒儲存區管理程式」）。專用的執行緒模型執行速度非常快，因為每個連接均有專用執行緒；但是，連接數目受到可用執行緒數目的限制（每個連接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接數目上沒有限制；但是，在連接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，特別是在連接忙碌時。

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 51 頁的「連接服務支援」中敘述的連接服務（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 9-5 傳輸協定速度



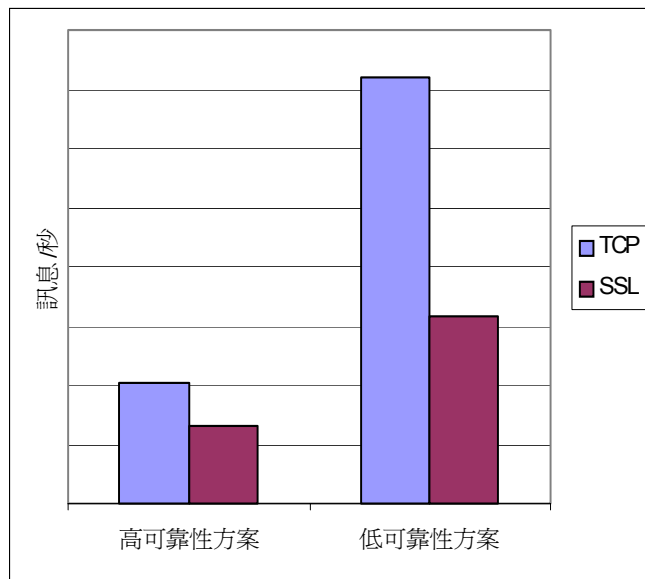
圖 9-5 反映各種協定技術的效能特徵：

- TCP 提供與代理程式進行通訊的最快方法。
- 傳送和接收性訊息時，SSL 的速率為 TCP 的 50% 到 70% (傳送和接收永久性訊息時為 TCP 速率的 50%，非永久訊息時則將近 70%)。此外，使用 SSL 建立初始連接速度較慢 (可能會花費數秒)，因為用戶端和代理程式 (或 HTTPS 情況下的 Web Server) 需要建立加密傳輸資料時會用到的私密密鑰。效能會因額外處理所需加密和解密每個低層 TCP 封包而降低。

圖 9-6 比較 TCP 和 SSL 兩種情況下的流量：高可靠性方案 (傳送到包含長期訂閱之主題目標的 1K 永久性訊息，並使用 AUTO_ACKNOWLEDGE 確認模式) 及高性能方案 (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 9-6 顯示高可靠性情況下較不造成影響的協定。這可能是因為在高可靠性情況中需要的持續性耗用時間，是限制流量的重要因素，其影響比對協定速度的影響大。

圖 9-6 傳輸協定的效能影響



- HTTP 的速度比 TCP 與 SSL 速度慢。HTTP 將 Web Server 上的 Servlet 使用作為用戶端與代理程式間的代理。效能耗用時間受到 HTTP 請求中封裝封包，與訊息通過兩個躍點（用戶端至 Servlet、Servlet 至代理程式）以到達代理程式之需求的影響。
- HTTPS 速度較 HTTP 慢，因為 HTTPS 需要花費在用戶端與 Servlet 間和 Servlet 與代理程式間加密封裝的額外耗用時間。

訊息伺服器架構

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

隨著連接至代理程式的用戶端數目的增加，以及要發送的訊息數目的增加，代理程式將最終超過資源限制（如檔案描述元、執行緒和記憶體限制）。提供增加載入的一個方法為將多個代理程式實例新增至 Message Queue 訊息伺服器，從而在多重代理程式間分布用戶端連接與訊息路由和發送。

一般而言，此調整會使運作達到最佳狀態，如果用戶端（尤其是訊息生產型用戶端）在叢集上平均分散的話。由於耗用時間受到叢集中代理程式間訊息發送的影響，所以包含連接數目限制或訊息發送速率限制的叢集，所表現效能可能會比單一代理程式低。

您還可以使用代理程式叢集來最佳化網路頻寬。例如，您可以在叢集中一組遠端代理程式間使用較慢、距離較長的網路連結，而使用較高速度的連結將用戶端連接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 72 頁的「[多重代理程式叢集 \(企業版\)](#)」和第 127 頁的「[使用叢集 \(企業版\)](#)」。

代理程式限制和行爲

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，造成沒有確認或不穩定的情況。

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、其個別目標，以及達到目標限制時可採取的行爲 (請參閱第 55 頁的「[管理記憶體資源和訊息流量](#)」)。

藉由謹慎監視與調整，這些可配置機制可用來平衡流入和流出訊息，如此一來，將不會發生系統過度負載情況。這些機制會耗用時間並限制訊息流量，但他們會維護作業完整性。

資料倉庫效能

Message Queue 支援內建和外掛持續性 (請參閱第 57 頁的「[持續性管理程式](#)」)。內建持續性是基於檔案的資料倉庫。外掛持續性使用 **Java** 資料庫連接 (**JDBC™**) 介面，並需要 **JDBC** 相容的資料倉庫。

內建持續性速度明顯比外掛持續性快；但是，**JDBC** 相容的資料庫系統可能會提供應用程式需要的容錯、安全與管理功能。

在使用內建持續性的情況下，您可以指定持續性作業同步化包含資料倉庫的內部記憶體狀態，以最大化可靠性。這可以消除因系統當機而產生的資料遺失，但會影響效能。

用戶端淨行時間配置

Message Queue 用戶端運行時間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目標和從此類目標接收訊息所需的所有作業。用戶端執行時間是可配置的 (藉由設定連線工廠屬性值)，它可讓您設定可改善效能與訊息流量的特性與行爲。

例如，**Message Queue** 用戶端執行時間支援以下可配置行爲：

- 連接流量計數 (`imqConnectionFlowCount`)，可幫助您避免 **JMS** 訊息和 **Message Queue** 控制訊息同時通過同一連接時，產生擁塞情形。

- 連接流量限制 (`imqConnectionFlowLimit`)，可幫助您藉由限制可透過連接發送至用戶端運行時間的待使用訊息數目，來避免用戶端資源限制。
- 使用者流量限制 (`imqConsumerFlowLimit`)，可在多使用者佇列發送情況中，幫助改善使用者之間的負載平衡（沒有使用者會收到過多的訊息數），並且有助於平衡所有連接使用者的流量。此特性限制每個使用者在連接中可發送到用戶端運行時間的待使用訊息數目。也可將此特性配置為佇列目標特性 (`consumerFlowLimit`)。

如需更多用於配置之行為和屬性的相關資訊，請參閱第 257 頁的「用戶端運行時間訊息流量調整」。

監視訊息伺服器

可配置 Message Queue 伺服器，以提供用來監視其效能的度量資訊。本節描述您可用來監視訊息伺服器的各種工具，以及使用這些工具來取得的度量資料。

如需如何使用度量資料以排解效能問題或分析與調整伺服器效能的資訊，請參閱第 236 頁的「排解效能問題」。

監視工具

您可以使用以下工具取得度量資訊：

- [Message Queue 命令行公用程式 \(`imqcmd`\)](#)
- [Message Queue 代理程式日誌檔](#)
- [基於訊息監視的 API](#)

以下各節描述如何使用這些工具來取得度量資訊。如需不同工具的比較，請參閱第 228 頁的「選擇正確的監視工具」。

Message Queue 命令行公用程式 (`imqcmd`)

命令行公用程式 (`imqcmd`) 為 Message Queue 的基本命令行管理工具。它可讓您管理代理程式及其連接服務，亦可管理應用程式特定的資源，例如實體目標、長期訂閱和異動。`imqcmd` 指令在第 6 章，「代理程式和應用程式管理」進行說明

`imqcmd` 指令的其中一個功能是可取得用於作為整體之代理程式、個別連接服務和個別目標的度量資訊。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 (請參閱第 224 頁的「`imqcmd query`」) 取得更多限制的度量資料子集。

imqcmd metrics

`imqcmd metrics` 的語法和選項個別顯示於表 9-2 和表 9-3 中。

表 9-2 `imqcmd metrics` 子指令語法

子指令語法	提供的度量資料
<pre>metrics bkr [-b <i>hostName:port</i>] [-m <i>metricType</i>] [-int <i>interval</i>] [-msp <i>numSamples</i>] [-u <i>userName</i>] [-p <i>password</i>]</pre>	顯示預設代理程式或指定主機和連接埠上的代理程式的代理程式度量。
或	
<pre>metrics svc -n <i>serviceName</i> [-b <i>hostName:port</i>] [-m <i>metricType</i>] [-int <i>interval</i>] [-msp <i>numSamples</i>] [-u <i>userName</i>] [-p <i>password</i>]</pre>	顯示預設代理程式或指定主機和連接埠上的代理程式上的指定服務之標準資訊。
或	
<pre>metrics dst -t <i>destType</i> -n <i>destName</i> [-b <i>hostName:port</i>] [-m <i>metricType</i>] [-int <i>interval</i>] [-msp <i>numSamples</i>] [-u <i>userName</i>] [-p <i>password</i>]</pre>	顯示有關指定類型和名稱之目標的度量資訊。

表 9-3 `imqcmd metrics` 子指令選項

子指令選項	說明
<code>-b <i>hostName:port</i></code>	指定報告度量資訊之代理程式的主機名稱和連接埠。預設值為 <code>localhost:7676</code>

表 9-3 imqcmd metrics 子指令選項

子指令選項	說明
-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 顯示代理程式流入和流出訊息和封包的度量 (預設度量類型) rts 顯示代理程式 (每秒) 流入和流出訊息和封包之流量速率的度量 cxn 顯示連接、虛擬記憶體堆疊和執行緒 (僅代理程式和連接服務) con 顯示使用者相關的度量 (僅目標) disk 顯示磁碟使用度量 (僅目標)。
-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目標的目標名稱 (如果有的話)。沒有預設值。
-n <i>serviceName</i>	指定報告度量的連接服務 (如果有的話)。沒有預設值。
-t <i>destTyp</i>	指定報告度量資料之目標 (如果有的話) 的類型 (佇列或主題)。沒有預設值。
-u <i>userName</i>	指定您的 (管理員的) 名稱。如果您遺漏此值，系統將提示您提供此名稱。
-p <i>password</i>	指定您的 (管理員的) 密碼。如果您遺漏此值，系統將提示您提供此名稱。

程序：使用顯示度量資料的 metrics 子指令

本節描述使用 metrics 子指令報告度量資訊的程序。

► 若要使用 metrics 子指令

1. 啟動需要度量資訊的代理程式。
請參閱第 122 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 9-2 和表 9-3。

度量輸出：imqcmd metrics

本節顯示用於代理程式範圍、連接服務和目標度量的度量子指令輸出範例。

代理程式範圍度量。 若要取得訊息和封裝每隔 10 秒流入和流出代理程式的速率，請使用 `metrics bkr` 子指令：

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

此指令會產生與以下相似的輸出 (請參閱第 231 頁的表 9-8 中的資料說明)：

Msgs/sec		Msg Bytes/sec		Pkts/sec		Pkt Bytes/sec	
In	Out	In	Out	In	Out	In	Out

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

連接服務度量。 若要取得由 `jms` 連接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 -p admin
```

此指令會產生與以下相似的輸出 (請參閱第 232 頁的表 9-9 中的資料說明)：

Msgs		Msg Bytes		Pkts		Pkt Bytes	
In	Out	In	Out	In	Out	In	Out

164	100	120704	73600	282	383	135967	102127
657	100	483552	73600	775	876	498815	149948

目標度量。 若要取得有關目標的度量資訊，請使用 `metrics dst` 子指令：

```
imqcmd metrics dst -t q -n XQueue -m ttl -u admin -p admin
```

此指令會產生與以下相似的輸出 (請參閱第 234 頁的表 9-10 中的資料說明)：

Msgs		Msg Bytes		Msg Count			Total Msg Bytes (k)			Largest
In	Out	In	Out	Current	Peak	Avg	Current	Peak	Avg	Msg (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

若要取得目標的使用者相關資訊，請使用以下 `metrics dst` 子指令：

```
imqcmd metrics dst -t q -n SimpleQueue -m con -u admin -p admin
```

此指令會產生與以下相似的輸出 (請參閱第 234 頁的表 9-10 中的資料說明)：

Active Consumers			Backup Consumers			Msg Count		
Current	Peak	Avg	Current	Peak	Avg	Current	Peak	Avg
1	1	0	0	0	0	944	1000	525

imqcmd query

`imqcmd query` 的語法和選項顯示於表 9-4 中，並隨附該指令提供的度量資料說明。

表 9-4 `imqcmd query` 子指令語法

子指令語法	提供的度量資料
<pre>query bkr [-b hostName:port] [-int interval] [-msp numSamples]</pre>	目前儲存於代理程式記憶體與永久性倉庫之訊息數目和訊息容量的資訊 (請參閱第 144 頁的「顯示代理程式資訊」)
或	
<pre>metrics svc -n serviceName [-b hostName:port] [-int interval] [-msp numSamples]</pre>	目前已配置執行緒的數目和指定連接服務之連接數目的資訊 (請參閱第 149 頁的「顯示連接服務資訊」)
或	
<pre>metrics dst -t destType -n destName [-b hostName:port] [-int interval] [-msp numSamples]</pre>	目前產生者、作用中和備份使用者，與用於指定目標中記憶體和永久性倉庫中儲存的訊息和訊息容量的資訊 (請參閱第 156 頁的「顯示目標資訊」)

備註 因為 `imqcmd query` 提供有限的度量資料，所以未在第 229 頁上的「度量資料的說明」小節中的表格列出此工具。

Message Queue 代理程式日誌檔

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）、日誌檔及 Solaris™ 平台上的 `syslog`（系統日誌）常駐程式。第 64 頁的「記錄程式」中說明了此記錄程式。

您可以指定記錄程式收集的資訊類型，以及寫入每個輸出通道的類型。要特別注意的是，您可以指定您要寫入日誌檔的度量資訊。

程序：使用代理程式日誌檔報告度量資訊

本節描述使用代理程式日誌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 133 頁的「記錄」。

► 若要使用日誌檔報告度量資訊

1. 配置代理程式的度量產生功能：

a. 確定 `imq.metrics.enabled=true`

依預設，會開啓用於記錄的度量產生功能。

b. 設定度量產生的合適秒數間隔時間。

`imq.metrics.interval=interval`

您可以在啓動代理程式時，在 `config.properties` 檔案中或使用 `-metrics interval` 指令行選項設定此值。

2. 確定記錄程式已收集度量資訊：

`imq.log.level=INFO`

這是預設值。您可以在啓動代理程式時，在 `config.properties` 檔案中或使用 `-loglevel level` 指令行選項設定此值。

3. 確定已設定記錄程式，將度量資訊寫入日誌檔：

`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。

度量輸出：日誌檔

以下顯示輸出到日誌檔的代理程式度量輸出範例 (請參閱表 9-7 和第 231 頁的表 9-8 中度量資料的說明)：

```
[21/Jul/2003:11:21:18 PDT]
Connections: 0    JVM Heap: 8323072 bytes (7226576 free) Threads: 0 (14-1010)
  In: 0 msgs (0bytes) 0 pkts (0 bytes)
  Out: 0 msgs (0bytes) 0 pkts (0 bytes)
Rate In: 0 msgs/sec (0 bytes/sec) 0 pkts/sec (0 bytes/sec)
Rate Out: 0 msgs/sec (0 bytes/sec) 0 pkts/sec (0 bytes/sec)
```

基於訊息監視的 API

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。第 65 頁的「度量資訊產生者 (企業版)」中描述了一般機制。

總共有 5 個度量主題目標。表 9-5 中顯示了這些目標的名稱，和發送到每個目標的度量訊息類型。

表 9-5 度量主題目標

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

程序：設定基於訊息的監視

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

► 若要設定基於訊息的監視

1. 寫入度量監視用戶端。

請參閱 *Message Queue Java 用戶端開發人員指南*，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。

使用者訂閱度量主題時，即會自動建立度量主題目標。一旦建立度量主題，代理程式度量訊息產生者會開始傳送度量訊息到度量主題。

安全性與存取注意事項

有兩個限制度量主題目標存取的原因：

- 度量資料可能包括有關代理程式或其資源的關聯資訊
- 過量的度量主題目標訂閱可能會增加代理程式耗用時間，並對效能產生不良影響

由於這些考量，所以建議您限制度量主題目標的存取。

監視用戶端必須遵守與其他用戶端相同的認證和授權控制。只有 **Message Queue** 使用者儲存庫中的使用者可以連接到代理程式。

您可以藉由存取控制特性檔限制存取特定度量主題目標，藉以提供額外保護，如 [第 192 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接，將度量監視用戶端連接到代理程式。如需使用加密連接的資訊，請參閱 [第 198 頁的「加密：使用基於 SSL 的服務 \(企業版\)」](#)。

度量輸出：度量訊息

您使用基於訊息監視的 **API** 取得的度量資料輸出，為您寫入度量監視用戶端的函數。您僅受限於代理程式中度量產生器提供的資料。如需此資料的完整清單，請參閱 [第 229 頁的「度量資料的說明」](#)。

選擇正確的監視工具

前面章節中提及的每個監視工具皆有其優點和缺點。

例如，使用 `imqcmd metrics` 指令可讓您快速取得符合您需求的資訊，但在查看歷程資訊或有計劃處理資料時會稍有難度。

相反地，日誌檔會提供長期的度量資料記錄，但日誌檔中的資訊會難以剖析成有意義的資訊。

基於訊息監視的 API 可讓您輕易擷取您所需的資訊，進而處理資訊、有計劃處理或格式化資料、呈現圖表或傳送警示；然而，您必須寫入自訂的應用程式以擷取和分析資料。

此外，每個工具皆會收集由代理程式產生且稍微不同的度量資訊子集。如需度量資料為何種監視工具所收集的資訊，請參閱第 229 頁的「度量資料的說明」。

表 9-6 比較不同工具，顯示每個工具的優缺點。

表 9-6 度量監視工具的優缺點

標準資訊監視工具	優點	缺點
imqcmd metrics	遠端監視 便於抽查 在指令選項中設定報告間隔時間；可即時變更 易於選取偏好的特定資料 以簡單表格格式呈現的資料	沒有取得所有資料的單一指令 難以有計劃分析資料 未建立歷程記錄 難以查看歷程趨勢
日誌檔	固定取樣 建立歷程記錄	需要配置代理程式特性；必須關閉並重新啟動代理程式才可生效 僅本地監視 資料格式難以讀取或剖析；沒有剖析工具 無法即時變更報告間隔時間；所有度量資料的報告間隔時間皆相同 不提供彈性選取資料 僅代理程式度量；不包括目標和連接服務度量 若間隔時間過短可能會影響效能
基於訊息監視的 API	遠端監視 易於選取偏好的特定資料 可透過電腦分析資料並以任何格式呈現	需要配置代理程式特性；必須關閉並重新啟動代理程式才可生效 您必須寫入您自己的度量監視用戶端 無法即時變更報告間隔時間；所有度量資料的報告間隔時間皆相同

度量資料的說明

可以將代理程式報告的度量資訊分為以下幾個種類：

- **Java 虛擬機器 (JVM) 度量。** 有關 JVM 堆疊容量的資訊。
- **代理程式範圍度量。** 有關儲存於代理程式中的訊息、輸入和輸出代理程式的訊息流量以及訊息數目、容量 (及速率) 的資訊。此種類亦包括記憶體使用的相關資訊。
- **連接服務度量。** 有關連接和連接執行緒資源的資訊，以及有關特定連接服務之訊息流量的資訊。
- **目標度量。** 有關流入和流出特定目標的訊息流量的資訊，以及有關目標使用者、記憶體和磁碟空間使用的資訊。

以下各節描述每個種類中可用的度量資料。如需以下表格中提及之監視工具的資訊，請參閱第 220 頁的「監視工具」。

JVM 度量

表 9-7 列出並描述代理程式產生用於代理程式處理 JVM 堆疊的度量資料，並顯示可使用不同度量監視工具取得的資料類型。

表 9-7 JVM 度量

度量數目	說明	imqcmd metrics bkr (metricType)	日誌檔	度量訊息 (度量主題) ²
JVM 堆疊： 可用記憶體	可用記憶體的容量可在 JVM 堆疊中使用。	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 226 頁的表 9-5。

代理程式範疇度量

表 9-8 列出並描述代理程式報告關於代理程式範圍度量資訊的資料。它亦顯示使用不同度量監視工具可取得的資料類型。

表 9-8 代理程式範圍度量

度量數目	說明	imqcmd metrics bkr (metricType)	日誌檔	度量訊息 (度量主題) ¹
連接資料				
連接數目	目前開啓至代理程式連接的數目	需要 (cxn)	需要	需要 (... 代理程式)
執行緒數目	目前使用中的執行緒數目	需要 (cxn)	需要	不需要
最小執行緒數目	一旦到達執行緒數目，便會在執行緒儲存區中維護執行緒數目，以供連接服務使用	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儲存區，以供連接服務使用	需要 (cxn)	需要	不需要
儲存的訊息資料				
訊息數目	目前儲存於代理程式記憶體和永久性倉庫的 JMS 訊息數目	不需要 使用 query bkr	不需要	需要 (... 代理程式)
訊息總容量	目前儲存於代理程式記憶體和永久性倉庫的 JMS 訊息容量	不需要 使用 query bkr	不需要	需要 (... 代理程式)
訊息流量資料				
流入的訊息數目	從上次代理程式啓動後，流入代理程式的 JMS 訊息數目	需要 (ttl)	需要	需要 (... 代理程式)
流入的訊息容量	從上次代理程式啓動後，流入代理程式的 JMS 訊息容量	需要 (ttl)	需要	需要 (... 代理程式)
流入的封包數目	從上次代理程式啓動後，流入代理程式的封包數目；包括 JMS 訊息和控制訊息	需要 (ttl)	需要	需要 (... 代理程式)
流入的封包容量	從上次代理程式啓動後，流入代理程式的封包容量；包括 JMS 訊息和控制訊息	需要 (ttl)	需要	需要 (... 代理程式)
流出的訊息數目	從上次代理程式啓動後，流出代理程式的 JMS 訊息數目	需要 (ttl)	需要	需要 (... 代理程式)
流出的訊息容量	從上次代理程式啓動後，流出代理程式的 JMS 訊息容量	需要 (ttl)	需要	需要 (... 代理程式)
流出的封包數目	從上次代理程式啓動後，流出代理程式的封包數目；包括 JMS 訊息和控制訊息。	需要 (ttl)	需要	需要 (... 代理程式)
流出的封包容量	從上次代理程式啓動後，流出代理程式的封包容量；包括 JMS 訊息和控制訊息。	需要 (ttl)	需要	需要 (... 代理程式)

表 9-8 代理程式範圍度量 (續)

度量數目	說明	imqcmd metrics bkr (metricType)	日誌檔	度量訊息 (度量主題) ¹
流入訊息的速率	目前流入代理程式的 JMS 訊息速率	需要 (rts)	需要	不需要
流入訊息容量的速率	目前流入代理程式之 JMS 訊息容量的速率	需要 (rts)	需要	不需要
流入的封包速率	目前流入代理程式的封包速率；包括 JMS 訊息和控制訊息	需要 (rts)	需要	不需要
流入封包容量的速率	目前流入代理程式之封包容量的速率；包括 JMS 訊息和控制訊息	需要 (rts)	需要	不需要
流出訊息的速率	目前流出代理程式的 JMS 訊息速率	需要 (rts)	需要	不需要
流出訊息容量的速率	目前流出代理程式之 JMS 訊息容量的速率	需要 (rts)	需要	不需要
流出封包的速率	目前流出代理程式的封包速率；包括 JMS 訊息和控制訊息	需要 (rts)	需要	不需要
流出封包容量的速率	目前流出代理程式之封包容量的速率；包括 JMS 訊息和控制訊息	需要 (rts)	需要	不需要
目標資料				
目標數目	代理程式中實體目標的數目	不需要	不需要	需要 (... 代理程式)

1. 如需度量主題目標的名稱，請參閱第 226 頁的表 9-5。

連接服務度量

表 9-9 列出並描述代理程式報告用於個別連接服務的度量資料。它亦顯示使用不同度量監視工具可取得的資料類型。

表 9-9 連接服務度量

度量數目	說明	imqcmd metrics svc (metricType)	日誌檔	度量訊息 (度量主題)
連接資料				
連接數目	目前開啟連接的數目	需要 (cxn) 亦可使用 query svc	不需要	不需要

表 9-9 連接服務度量 (續)

度量數目	說明	imqcmd metrics svc (metricType)	日誌 檔	度量訊息 (度量主題)
執行緒數目	目前使用中的執行緒數目，為所有連接服務的加總	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，便會在執行緒儲存區中維護執行緒數目，以供連接服務使用；此數目為所有連接服務的加總	需要 (cxn)	不 需 要	不 需 要
最大執行緒數目	一旦超過執行緒數目，便不會再新增執行緒到執行緒儲存區，以供連接服務使用；此數目為所有連接服務的加總	需要 (cxn)	不 需 要	不 需 要
訊息流量資料				
流入的訊息數目	從上次代理程式啟動後，流入連接服務的 JMS 訊息數目	需要 (ttl)	不 需 要	不 需 要
流入的訊息容量	從上次代理程式啟動後，流入連接服務的 JMS 訊息容量	需要 (ttl)	不 需 要	不 需 要
流入的封包數目	從上次代理程式啟動後，流入連接服務的封包數目；包括 JMS 訊息和控制訊息。	需要 (ttl)	不 需 要	不 需 要
流入的封包容量	從上次代理程式啟動後，流入連接服務的封包容量；包括 JMS 訊息和控制訊息。	需要 (ttl)	不 需 要	不 需 要
流出的訊息數目	從上次代理程式啟動後，流出連接服務的 JMS 訊息數目	需要 (ttl)	不 需 要	不 需 要
流出的訊息容量	從上次代理程式啟動後，流出連接服務的 JMS 訊息容量	需要 (ttl)	不 需 要	不 需 要
流出的封包數目	從上次代理程式啟動後，流出連接服務的封包數目；包括 JMS 訊息和控制訊息。	需要 (ttl)	不 需 要	不 需 要
流出的封包容量	從上次代理程式啟動後，流出連接服務的封包容量；包括 JMS 訊息和控制訊息。	需要 (ttl)	不 需 要	不 需 要
流入訊息的速率	目前通過連接服務流入代理程式的 JMS 訊息速率	需要 (rts)	不 需 要	不 需 要
流入訊息容量的速率	目前流入連接服務之 JMS 訊息容量的速率	需要 (rts)	不 需 要	不 需 要
流入的封包速率	目前流入連接服務的封包速率；包括 JMS 訊息和控制訊息	需要 (rts)	不 需 要	不 需 要
流入封包容量的速率	目前流入連接服務之封包容量的速率；包括 JMS 訊息和控制訊息	需要 (rts)	不 需 要	不 需 要

表 9-9 連接服務度量 (續)

度量數目	說明	imqcmd metrics svc (metricType)	日誌 檔	度量訊息 (度量主題)
流出訊息的速率	目前流出連接服務的 JMS 訊息速率	需要 (rts)	不 需 要	不 需 要
流出訊息容量的速率	目前流出連接服務之 JMS 訊息容量的速率	需要 (rts)	不 需 要	不 需 要
流出封包的速率	目前流出連接服務的封包速率；包括 JMS 訊息和控制訊息	需要 (rts)	不 需 要	不 需 要
流出封包容量的速率	目前流出連接服務之封包容量的速率；包括 JMS 訊息和控制訊息	需要 (rts)	不 需 要	不 需 要

目標度量

表 9-9 列出並描述代理程式報告用於個別目標的度量資料。它亦顯示使用不同度量監視工具可取得的資料類型。

表 9-10 目標度量

度量數目	說明	imqcmd metrics dst (metricType)	日誌 檔	度量訊息 (度量主題) ¹
使用者資料				
作用中使用者的數目	目前作用中使用者的數目	需要 (con)	不 需 要	需要 (...destName)
作用中使用者的平均數目	從上次代理程式啟動後，作用中使用者的平均數目	需要 (con)	不 需 要	需要 (...destName)
作用中使用者的尖峰數目	從上次代理程式啟動後，作用中使用者的尖峰數目	需要 (con)	不 需 要	需要 (...destName)
備份使用者的數目	目前備份使用者的資料 (僅適用於佇列)	需要 (con)	不 需 要	需要 (...destName)
備份使用者的平均數目	從上次代理程式啟動後，備份使用者的平均數目 (僅適用於佇列)	需要 (con)	不 需 要	需要 (...destName)
備份使用者的尖峰數目	從上次代理程式啟動後，備份使用者的尖峰數目 (僅適用於佇列)	需要 (con)	不 需 要	需要 (...destName)

表 9-10 目標度量 (續)

度量數目	說明	imqcmd metrics dst (metricType)	日誌 檔	度量訊息 (度量主題) ¹
儲存的訊息資料				
訊息數目	目前儲存於目標記憶體和永久性倉庫的 JMS 訊息數目	需要 (con) (ttl) (rts) 亦可使用 query dst	不需 要	需要 (...destName)
訊息平均數目	從上次代理程式啟動後，儲存於目標記憶體 和永久性倉庫中的 JMS 訊息平均數目	需要 (con) (ttl) (rts)	不需 要	需要 (...destName)
訊息尖峰數目	從上次代理程式啟動後，儲存於目標記憶體 和永久性倉庫中的 JMS 訊息尖峰數目	需要 (con) (ttl) (rts)	不需 要	需要 (...destName)
訊息總容量	目前儲存於目標記憶體和永久性倉庫的 JMS 訊息容量	需要 (ttl) (rts) 亦可使用 query dst	不需 要	需要 (...destName)
訊息的總平均容量	從上次代理程式啟動後，儲存於目標記憶體 和永久性倉庫中的 JMS 訊息平均容量	需要 (ttl) (rts)	不需 要	需要 (...destName)
訊息的總尖峰容量	從上次代理程式啟動後，儲存於目標記憶體 和永久性倉庫中的 JMS 訊息尖峰容量	需要 (ttl) (rts)	不需 要	需要 (...destName)
訊息的尖峰容量	從上次代理程式啟動後，目標接收的單一訊 息中 JMS 訊息的尖峰容量	需要 (ttl) (rts)	不需 要	需要 (...destName)
訊息流量資料				
流入的訊息數目	從上次代理程式啟動後，流入目標的 JMS 訊息數目	需要 (ttl)	不需 要	需要 (...destName)
流入的訊息容量	從上次代理程式啟動後，流入目標的 JMS 訊息容量	需要 (ttl)	不需 要	需要 (...destName)
流出的訊息數目	從上次代理程式啟動後，流出目標的 JMS 訊息數目	需要 (ttl)	不需 要	需要 (...destName)
流出的訊息容量	從上次代理程式啟動後，流出目標的 JMS 訊息容量	需要 (ttl)	不需 要	需要 (...destName)

表 9-10 目標度量 (續)

度量數目	說明	imqcmd metrics dst (metricType)	日誌 檔	度量訊息 (度量主題) ¹
流入訊息的速率	目前流入目標的 JMS 訊息速率	需要 (rts)	不需要	不需要
流出訊息的速率	目前流出目標式的 JMS 訊息速率	需要 (rts)	不需要	不需要
流入訊息容量的 速率	目前流入目標之 JMS 訊息容量的速率	需要 (rts)	不需要	不需要
流出訊息容量的 速率	目前流出目標之 JMS 訊息容量的速率	需要 (rts)	不需要	不需要
磁碟使用資料				
保留的磁碟空間	基於檔案的目標倉庫中，所有訊息記錄 (作用中或可用) 使用的磁碟空間 (以位元組為單位)	需要 (dsk)	不需要	需要 (...destName)
使用的磁碟空間	基於檔案的目標倉庫中，作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	使用的磁碟空間除以保留的磁碟空間所得之商數比率越高表示有越多磁碟空間被用來保留作用中訊息	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 226 頁的表 9-5。

排解效能問題

使用 Message Queue 服務支援應用程式時會發生一些效能問題。這些問題包括：

- 問題：用戶端無法建立連接
- 問題：連接流量過慢
- 問題：用戶端無法建立訊息產生者
- 問題：訊息產生延遲或過慢
- 問題：訊息儲存在訊息伺服器中
- 問題：訊息伺服器的流量不穩定
- 問題：訊息未送達使用者

以下將描述每個問題的可能原因和解決方案。

問題：用戶端無法建立連接

徵兆：

- 用戶端無法建立新的連接。
- 用戶端無法自動重新連接失敗的連接。

可能原因：

- 用戶端應用程式為關閉連接，導致連接數目超過資源限制。

若要確認此問題的原因：

列出所有代理程式的連接：

```
imqcmd list cxn
```

輸出會列出所有連接和建立每個連接的主機，顯示特定用戶端已開啓之不經常使用的連接數目。

若要解決此問題：

重新寫入違例用戶端，以關閉不使用的連接。

- 未執行代理程式或網路連接發生問題。

若要確認此問題的原因：

- 使用 Telnet 連接至用戶端主要連接埠（例如，預設值為 7676），並以連接埠對映器輸出驗證代理程式確認。
- 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。
- 修復網路連接問題。
- 連接服務不在作用中或已暫停。

若要確認此問題的原因：

檢查所有連接服務的狀態：

```
imqcmd list svc
```

如果連接服務狀態顯示如下 unknown（未知）或 paused（暫停中），那麼用戶端無法使用服務建立連接。

若要解決此問題：

- 如果連接服務狀態顯示為 `unknown` (未知)，那麼此服務會從作用中服務清單移出 (`imq.service.active`)。在基於 SSL 服務的情況中，可能會錯誤配置此服務，造成代理程式在代理程式日誌中產生以下項目：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 198 頁的「透過 TCP/IP 設定基於 SSL 的服務」。

- 如果連接服務狀態顯示為 `paused` (暫停中)，那麼會繼續服務 (請參閱第 150 頁的「暫停與繼續連接服務」)。
- 可用於所需連接數目的可用執行緒太少。

若要確認此問題的原因：

檢查代理程式日誌中是否有以下項目：警告 [B3004]：沒有可用於服務上處理新連接的執行緒... 關閉新的連接。

並檢查連接服務上的連接數目和目前使用中的執行緒數目：

```
imqcmd query svc -n serviceName
或
imqcmd metrics svc -n serviceName -m cxn
```

每個連接均需要兩個執行緒：一個用於進來的訊息，另一個用於外送的訊息 (請參閱第 51 頁的「執行緒儲存區管理程式」)。

若要解決此問題：

- 如果您使用專用執行緒儲存區模型 (`imq.service_name.threadpool_model=dedicated`)，那麼最大的連接數目為執行緒儲存區中最大執行緒數目的一半。因此，若要增加連接數目，請增加執行緒儲存區的容量 (`imq.service_name.max_threads`) 或切換為共用執行緒儲存區模型。
- 如果您使用共用執行緒儲存區模型 (`imq.service_name.threadpool_model=shared`)，那麼最大的連接數目為以下兩種特性產品的一半：連接監視限制 (`imq.service_name.connectionMonitor_limit`) 和最大執行緒數目 (`imq.service_name.max_threads`)。因此，若要增加連接數目，請增加執行緒儲存區的容量或增加連接監視限制。
- 可支援的連接數目 (或連接上的流量) 最終會到達輸入 / 輸出限制。在此情況下，請使用多重代理程式叢集 (請參閱第 127 頁的「使用叢集 (企業版)」)，以分散叢集中代理程式實例上的連接。

- Solaris 或 Linux 平台上要求用於連接數目的檔案描述元太少 (請參閱第 302 頁的「定義作業系統檔案描述元限制」)。

若要確認此問題的原因：

檢查代理程式日誌中是否有以下相似項目：太多開啓檔案。

若要解決此問題：

增加檔案描述元限制，如 ulimit 線上援助頁所述。

- TCP 儲存區會限制同時新建之連接請求的數目。

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接請求數目 (imq.portmapper.backlog)。(在 Windows 平台上有程式內定的儲存區限制：Windows Desktop 的數目為 5，Windows Server 的數目為 200。)

因為不經常有高數目的同時連接請求，所以因儲存區限制的請求拒絕通常為短暫現象。

若要確認此問題的原因：

檢查代理程式日誌，查看在拒絕其他連接請求時，是否在幾乎相同的時間接受其他連接請求。拒絕的連接請求會傳回 java.net.ConnectException：拒絕連接。

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，讓用戶端重試連接 (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接的次數是否過於頻繁。
- 作業系統會限制運作中連接的數目。

Windows 作業系統授權會限制支援的運作遠端連接的數目。

若要確認此問題的原因：

檢查可用於連接的執行緒數目 (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接，而不是從遠端用戶端，那麼作業系統限制可以是造成此問題的原因。

若要解決此問題：

- 升級 Windows 授權以允許更多連接。

- 設定多重代理程式叢集，以分散一些代理程式實例的连接。
- 使用者的驗證或授權失敗。

驗證可能會因為錯誤密碼而失敗，因為沒有使用者進入使用者儲存庫的項目，或因為使用者沒有存取連接服務的許可權。

若要確認此問題的原因：

檢查代理程式日誌中是否有禁用錯誤訊息的項目。這表示有驗證錯誤，但不表示為問題發生的原因。

- 如果您使用基於檔案的使用者儲存庫，請輸入以下指令：

```
imqusermgr list -i instanceName -u userName
```

如果輸出顯示使用者，那麼可能是提交了錯誤密碼。如果輸出顯示為錯誤 [B3048]：使用者不存在於密碼檔案中，則沒有進入使用者儲存庫的項目。

- 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，查看是否有存取連接服務的限制。

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，那麼請將使用者新增到使用者儲存庫（請查看第 188 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，以取得連接服務許可權（請參閱第 196 頁的「連接存取控制」）。

問題：連接流量過慢

徵兆：

- 訊息流量未符合預期速度。
- 支援連接到代理程式的數目不受到限制，如第 237 頁的「問題：用戶端無法建立連接」所述，但訊息輸入 / 輸出速率受到限制。

可能原因：

- 網路連接或 WAN 太慢。

若要確認此問題的原因：

偵測網路，查看偵測到網路確認所需的時間，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）比較發送時間。

若要解決此問題：

如果連接過慢，請升級網路連結。

- 連接服務協定的速度原本就比 TCP 的速度慢。例如，基於 SSL 或基於 HTTP 協定的速度會較 TCP 的速度慢（請參閱第 217 頁的圖 9-5）。

若要確認此問題的原因：

如果您使用基於 SSL 或基於 HTTP 的協定，請嘗試使用 TCP 並比較發送時間。

若要解決此問題：

應用程式需求通常會表示要使用的協定，這是您可以參考的一點。除此之外，您還可以嘗試調整協定，如第 252 頁的「調整傳輸協定」所述。

- 連接服務協定未調整到最佳化。

若要確認此問題的原因：

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 252 頁的「調整傳輸協定」所述。

- 訊息過大導致使用過多的頻寬。

若要確認此問題的原因：

嘗試以較小訊息執行效能評定。

若要解決此問題：

- 使用 `java.util.zip` 壓縮訊息內文。
- 將訊息使用作為要傳送的資料通知，但使用其他協定移動資料。
- 實際上，連接流量過慢的問題，是訊息發送程序的某些步驟中會遇到的瓶頸。

若要確認此問題的原因：

如果上述原因都不是造成連接流量速度過慢的原因，則請查閱第 208 頁的圖 9-1，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 242 頁的「問題：訊息產生延遲或過慢」
- 第 245 頁的「問題：訊息儲存在訊息伺服器中」

- [第 249 頁](#) 的「問題：訊息伺服器的流量不穩定」

若要解決此問題：

依照以上問題排解各節提供的問題解決方針。

問題：用戶端無法建立訊息產生者

徵兆：

- 無法建立用於目標的訊息產生者；用戶端發生異常。

可能原因：

- 已將目標配置為僅允許一定數目的產生者。

若要防止目標上訊息累積，其中一個方法是限制目標可支援的產生者數目 (maxNumProducers)。

若要確認此問題的原因：

檢查目標 (請參閱[第 156 頁](#)的「顯示目標資訊」)：

```
imgcmd 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，則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，且在代理程式日誌中產生以下項目：[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 157 頁](#)的「更新目標屬性」)。

- 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。

若要確認此問題的原因：

代理程式拒絕新的產生者時，代理程式會傳回 JMSSecurityException [C4076]：用戶端沒有在目標上建立產生者的許可權，在代理程式日誌中產生以下項目：[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 196 頁](#)的「目標存取控制」)。

問題：訊息產生延遲或過慢

徵兆：

- 傳送永久性訊息時，不會傳回 `send()` 方法且阻塞用戶端。
- 傳送永久性訊息時，用戶端發生異常。
- 產生用戶端的速度變慢。

可能原因：

- 訊息伺服器進行儲存動作（訊息會累積在代理程式記憶體中），且做出降低訊息產生者速度的確認。

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行爲，嘗試節省記憶體資源。以下限制行爲會降低訊息產生者的速度：

- `FLOW_CONTROL`：代理程式不會馬上確認永久性訊息的接收（因此會阻塞生產型用戶端）
- `REJECT_NEWEST`：代理程式拒絕新的訊息（並針對每個拒絕的永久性用戶端，拋出異常）。

同樣地，代理程式範圍記憶體（用於所有目標）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（因為目標或代理程式範圍的限制未正確設定），代理程式會採取越來越重要的動作，以防記憶體超過負載，包括減低訊息產生者的速度。

如需這些機制的說明，請參閱第 55 頁的「管理記憶體資源和訊息流量」。

若要確認此問題的原因：

代理程式因配置的訊息限制拒絕訊息時，代理程式會傳回 `JMSEException` [C4036]：伺服器發生錯誤，並在代理程式日誌中產生項目：警告 [B2011]：儲存來自 `IMQconn` 的 JMS 訊息失敗，並隨附一個訊息，說明已到達限制：

- 如果訊息限制位於目標上，那麼代理程式會產生類似以下的項目：
[B4120]：無法在目標 `destName` 上儲存訊息，因為會超出 `maxNumMsgs` 的容量。
- 如果訊息限制為代理程式範圍，那麼代理程式會產生類似以下的項目：
[B4024]：已經超出目前系統中訊息的最大數目，正在拒絕訊息。

一般而言，您可以在拒絕訊息前檢查訊息限制條件，方法為查詢目標和代理程式、檢查配置的訊息限制設定，並使用 `imqcmd` 指令監視目前目標（或作為整體代理程式）中的訊息數目和訊息容量（請個別查看第 234 頁的表 9-10 和 第 231 頁的表 9-8）。

若要解決此問題：

有數種方法可以解決訊息儲存時造成產生者速度變慢的問題。

- 修改目標（或代理程式範圍）上的訊息限制，保持不要超過記憶體資源。一般來說，您要在目標對目標層級上管理記憶體，如此一來，才永遠不會到達代理程式範圍的訊息限制。如需更多資訊，請參閱第 255 頁的「代理程式調整」。
- 變更目標上的限制行為，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` 限制行為（請參閱第 154 頁的表 6-10）。
- 代理程式將永久性訊息儲存到資料倉庫。

如果代理程式無法存取資料倉庫或將永久性訊息寫入資料倉庫，則表示生產型用戶端被封鎖。這種情況亦會在目標或代理程式範圍訊息到達限制時發生，如上所述。

若要確認此問題的原因：

如果代理程式無法寫入資料倉庫，則會在代理程式日誌中產生以下其中一個項目：[B2011]：儲存來自 `connectionID` 的 JMS 訊息失敗... 或 [B4004]：無法存留訊息 `messageID...`

若要解決此問題：

- 在內建持續性的情況下，請嘗試增加基於檔案的資料倉庫的磁碟空間。
- 在使用 JDBC 相容資料倉庫的情況下，請檢查是否正確配置外掛持續性（請參閱附錄 B「設定外掛持續性」）。如果已正確配置，請洽詢您的資料庫管理員以排解其他資料庫問題。
- 代理程式確認逾時過短。

由於連接速度慢或訊息伺服器緩慢（因高 CPU 使用率或記憶體資源不足所引起），代理程式可能需要更多時間來確認持久性訊息的接收，且時間超過連線工廠的 `imqAckTimeout` 屬性所允許的值。

若要確認此問題的原因：

如果超過 `imqAckTimeout` 的值，代理程式會傳回 `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 168 頁的「連線工廠受管理物件屬性」)。

- 生產型用戶端受到 JVM 限制。

若要確認此問題的原因：

- 檢查用戶端應用程式是否接收到「記憶體不足」錯誤。
- 使用運行時間方法，例如 `freeMemory()`、`MaxMemory()` 和 `totalMemory()`，來檢查 JVM 堆疊中的可用記憶體。

若要解決此問題：

調整 JVM (請查看第 252 頁的「Java 虛擬機器調整」)。

問題：訊息儲存在訊息伺服器中

徵兆：

- 代理程式 (或特定目標) 中的訊息數目或訊息容量會不斷穩定增加。

若要查看是否累積訊息，請檢查代理程式中的訊息數目或訊息容量如何不斷變化，並與配置的限制進行比較。請先檢查配置的限制：

```
imqcmd query bkr
```

(注意：`imqcmd metrics bkr` 子指令不會顯示此資訊。)

接著檢查每個目標中的訊息累積：

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

或

```
imqcmd metrics dst -t destType -n destName -m ttl
```

若要查看訊息是否超過配置的目標或代理程式範圍限制，請檢查代理程式中是否有以下項目：警告 [B2011]：儲存來自 ... 的 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。
- 訊息花費一段不尋常的長時間到達使用者。

可能原因：

- 用戶端程式碼缺點：使用者不確認訊息。

所有使用者確認傳送的訊息前，訊息會被保留在目標中。因此，如果用戶端不確認使用的訊息，那麼訊息會累積在目標中且不被刪除。

舉例來說，用戶端程式碼可能會有以下缺點：

- 使用 `CLIENT_ACKNOWLEDGEMENT` 或異動階段作業的使用者可能不會定期呼叫 `Session.acknowledge()` 或 `Session.commit()`。
- 使用 `AUTO_ACKNOWLEDGE` 階段作業的使用者可能會因某些原因擱置。

若要確認此問題的原因：

如果訊息伺服器不為忙碌狀態，即流入和流出目標的訊息速率為低時，則可能會因為沒有確認訊息而累積這些訊息。

檢查流入和流出代理程式的訊息速率：

```
imqcmd metrics bkr -m rts
```

接著檢查每個個別目標的流量速率：

```
imqcmd metrics bkr -t destType -n destName -m rts
```

還要檢查用戶端程式碼，查看是否正確確認訊息。

- 主題目標上有非作用中的長期訂閱。

如果長期訂閱為非作用中，那麼訊息會儲存在目標中，直到對應的使用者變為作用中狀態，且可使用這些訊息為止。

若要確認此問題的原因：

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 161 頁的「管理長期訂閱」)。
- 指定用於主題的訊息限制和訊息行為屬性 (請參閱第 154 頁的表 6-10)。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` 限制行為。
- 從對應的目標清除所有訊息 (請參閱第 159 頁的「清除目標」)。
- 時間訊息的限制可保留在記憶體中：您可以重新寫入生產型用戶端，以設定每個訊息上的存在時間值。您可以置換用於所有產生者共用連接的任何設定值，方法為設定 `imqOverrideJMSEExpiration` 和 `imqJMSEExpiration` 的連線工廠屬性 (請參閱第 168 頁的表 7-3)。

- 佇列中可使用訊息的使用者太少。

如果可發送訊息的作用中使用者太少，那麼可以在訊息累積時儲存佇列目標。此情況會因以下任何一種原因而發生：

- 目標中存在的作用中使用者太少。
- 使用用戶端建立連接失敗。
- 沒有作用中使用者使用符合佇列中訊息的選擇器。

若要確認此問題的原因：

若要判斷無法使用使用者的原因，請檢查目標上作用中使用者的數目：

```
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使用者的原因，採取以下任何一個動作：

- 啟動其他使用用戶端，以建立更多用於佇列的作用中使用者。
 - 調整 `imq.consumerFlowLimit` 代理程式特性，以最佳化至多個使用者的佇列發送（請參閱第 256 頁的「多重使用者佇列效能」）。
 - 指定用於佇列的訊息限制和訊息行為屬性（請參閱第 154 頁的表 6-10）。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` 限制行為。
 - 從對應的目標清除所有訊息（請參閱第 159 頁的「清除目標」）。
 - 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，還可置換用於所有產生者共用連接的任何設定值，方法為設定 `imqOverrideJMSEExpiration` 和 `imqJMSEExpiration` 連線工廠屬性（請參閱第 168 頁的表 7-3）。
- 訊息使用者處理速度過慢，無法跟上訊息產生者的速度。

在此情況下，主題用戶或佇列接收者使用訊息的速度，會比產生者傳送訊息的速度慢。因為此不平衡現象，所以會儲存一個或多個包含訊息的目標。

若要確認此問題的原因：

檢查流入和流出代理程式的訊息速率：

```
imqcmd metrics bkr -m rts
```

接著檢查每個個別目標的流量速率：

```
imqcmd metrics bkr -t destType -n destName -m rts
```

若要解決此問題：

- 最佳化使用用戶端程式碼。
- 增加用於佇列目標的作用中使用者數目 (請參閱第 256 頁的「多重使用者佇列效能」)。
- 用戶端確認處理會降低訊息使用速度。

兩個影響用戶端確認處理的因素：

- 處理用戶端確認時會大量使用代理程式資源。因此，可能會降低阻塞使用用戶端之確認模式中的訊息使用速度，直到代理程式確認用戶端確認為止。
- JMS 有效負載訊息和 Message Queue 控制訊息 (例如用戶端確認) 共用同一個連接。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。

若要確認此問題的原因：

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，那麼用戶端確認可能有問題。

還要檢查用戶端是否接收到以下訊息 `JMSEException [C4000]`：封包確認失敗。

若要解決此問題：

- 修改用戶端使用的確認模式，例如切換為 `DUPS_OK_ACKNOWLEDGEMENT` 或 `CLIENT_ACKNOWLEDGEMENT`。
- 如果使用 `CLIENT_ACKNOWLEDGEMENT` 或異動階段作業，請將更大的訊息數目設為單一確認的群組。
- 調整使用者和連接流量控制參數 (請參閱第 257 頁的「用戶端運行時間訊息流量調整」)。
- 代理程式無法跟上產生訊息的速度。

在此情況下，訊息流入代理程式的速度，會比代理程式路由和派送訊息到使用者的速度快。代理程式運作遲緩可以是因為以下任何或所有限制造成：CPU、網路套接讀 / 寫作業、磁碟讀 / 寫作業、記憶體分頁、永久性倉庫或 JVM 記憶體限制。

若要確認此問題的原因：

檢查是否有其他造成此問題的原因。

若要解決此問題：

- 升級您電腦或資料倉庫的速度。
- 使用代理程式叢集，分散一些代理程式實例上的負載。

問題：訊息伺服器的流量不穩定

徵兆：

- 訊息流量偶爾會降低，接著又恢復正常效能。

可能原因：

- 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。

若要確認此問題的原因：

檢查記憶體不足情況下的代理程式日誌 ([B1089]：在記憶體不足的條件下，代理程式正嘗試釋放資源)，並隨附一個項目，描述新的記憶體狀態和使用的記憶體總數。

還要檢查 JVM 堆疊中可用的記憶體：

```
imqcmd metrics bkr -m cxn
```

JVM 記憶體總數接近最大 JVM 記憶體值時，可用記憶體會不足。

若要解決此問題：

- 調整 JVM (請查看第 252 頁的「Java 虛擬機器調整」)。
- 增加系統交換空間。
- 執行 JVM 記憶體回收 (廢棄項目收集)。

會定期透過系統清除記憶體回收，釋放記憶體。發生此情況時，會阻塞所有執行緒。被釋放的記憶體容量越多，JVM 堆疊的大小就越大，且因記憶體回收的延遲時間也越長。

若要確認此問題的原因：

檢視您電腦的 CPU 使用。執行記憶體回收時，會大幅降低 CPU 使用。

再者，請使用以下指令行選項啟動代理程式：

```
-vmargs -verbose:gc
```

標準輸出說明執行記憶體回收的時間。

若要解決此問題：

在多重 CPU 的電腦中，設定同時執行的記憶體回收：

`-XX:+UseParallelGC=true`

- JVM 使用 Just-In-Time 編譯器來提高效能。

若要確認此問題的原因：

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。

問題：訊息未送達使用者

徵兆：

- 使用者未收到產生者傳送的訊息。

可能原因：

- 限制行為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，代理程式會嘗試節省記憶體資源。代理程式會配置三個可配置的行為，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，刪除具最低優先權的訊息
- REJECT_NEWEST：拒絕新的訊息（針對拒絕的永久性訊息拋出異常）。

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，代理程式會藉由拒絕最新訊息，嘗試節省記憶體資源。

若要確認此問題的原因：

檢查代理程式是否有以下項目：警告 [B2011]：儲存來自 ... 的 JMS 訊息失敗。此項目會隨附其他項目，說明已超過限制。但是，沒有顯示訊息刪除的項目。

若要解決此問題：

變更限制或變更行為。

- 訊息逾時值過期

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。

若要確認此問題的原因：

檢查代理程式是否有以下項目；過期訊息：已過期 n 個訊息。

若要解決此問題：

使用置換

- 兩台不同電腦的時間未同步化。

如果時鐘不同步，那麼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。

若要確認此問題的原因：

檢查所有電腦的時鐘。

若要解決此問題：

同步化時鐘 (請參閱第 301 頁的「系統時鐘設定」)。

- 使用用戶端無法啟動連接上的訊息發送。

用戶端程式碼建立連接並啟動連接上的訊息發送前，將無法發送訊息。

若要確認此問題的原因：

檢查用戶端程式碼是否建立連接並啟動訊息發送。

若要解決此問題：

重新寫入用戶端程式碼，以建立連接並啟動訊息發送。

調整您的配置以改善效能

系統調整

以下各節描述您可對作業系統、JVM 和連接協定所做的調整。

Solaris 調整：CPU 使用率、分頁 / 交換 / 磁碟 I/O

請參閱您系統的說明文件來調整您的作業系統。

調整 Java 虛擬機配置

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接以釋放記憶體，並降低訊息的流入量。因此，最好將最大 JVM 堆疊儲存區空間設定得足夠大，以避免發生此類情況。

但是，如果將最大 Java 堆疊儲存區空間設定得過大（相對於系統實體記憶體），代理程式會持續增加 Java 堆疊儲存區空間，直至整個系統的記憶體用完。這會導致效能降低、不可預料的代理程式當機，和 / 或影響系統上執行的其他應用程式和服務的運作。一般而言，您需要允許足夠用於作業系統和機器上其他執行應用程式的實體記憶體。

一般說來，最好是先評估正常時和高峰時的系統記憶體足跡，然後再配置 Java 堆疊儲存區容量，使其足以提供良好的效能，但又不會太大，以至出現系統記憶體問題。

若要變更代理程式最小和最大的堆疊大小，請在啟動代理程式時使用 `-vmargs` 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 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` 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 297 頁的「[服務管理員公用程式 \(imqsvcadm\)](#)」。

在任何情況下，請藉由檢查代理程式的日誌檔或使用 `imqcmd metrics bkr -m cxn` 指令來驗證設定值。

調整傳輸協定

一旦選擇符合應用程式需求的協定，則額外調整（以選定的協定為基礎）可能會改善效能。

可以使用以下三個代理程式特性來修改協定效能：

- `imq.protocol protocol_type nodelay`

- `imq.protocol protocol_type inbufsz`
- `imq.protocol protocol_type outbufsz`

針對 TCP 和 SSL 協定，這些特性會影響用戶端和代理程式間訊息發送的速度。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發送速度。針對 HTTP/HTTPS 協定，有其他特性可以影響效能 (請參閱第 255 頁的「HTTP/HTTPS 調整」)。

將會在以下各節描述協定調整特性。

nodelay

`nodelay` 特性會影響用於既定協定的 Nagle 演算法 (TCP/IP 上的 TCP_NODELAY 套接級別選項)。Nagle 演算法可使用慢速連接，例如廣域網路 (WAN)，來改善系統上的 TCP 效能。

使用該演算法時，TCP 會嘗試阻止一些小型資料區塊傳送到遠端系統 (藉由將資料隨附於較大封包中)。如果寫入套接的資料未填滿所需緩衝區大小，那麼協定會在緩衝區填滿或特定延遲時間結束前，延遲傳送封包。一旦緩衝區為填滿狀態或發生逾時，即會傳送封包。

針對大部分的訊息傳送應用程式，在傳送封包時沒有延遲即表示效能為最佳狀態 (未啟用 Nagle 演算法)。這是因為大部分用戶端與代理程式間的互動為請求 / 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：

- 建立連接
- 建立產生者或使用者
- 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的程序)

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使用者時，會延遲數毫秒。

但是，Nagle 演算法會在連接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

`inbufsz` 特性會設定輸入串流上的緩衝區大小，輸入串流可讀取來自套接的資料。同樣地，`outbufsz` 會設定輸出串流的緩衝區大小，輸出串流為代理程式用來將資料寫入套接。

一般而言，兩個參數值皆應稍微大於接收或傳送的平均封包大小。根據經驗，請將這些特性值設定為比平均封包大小多 1K (將近 1K)。

舉例來說，如果代理程式接收內文大小為 1K 的封包，那麼封包 (訊息內文 + 標頭 + 特性) 的總容量則約為 1,200 個位元組一個 2K (2,048 個位元組) 的 `inbufsz` 即可產生適當效能。

將 `inbufsz` 或 `outbufsz` 增加到超過此大小可能會稍微改善效能；但是，這可能會增加每個連接的所需記憶體。

圖 9-6 顯示在 1K 的封包上變更 `inbufsz` 的結果。

圖 9-7 在 1K (1,024 個位元組) 封包上變更 `inbufsz` 的效果

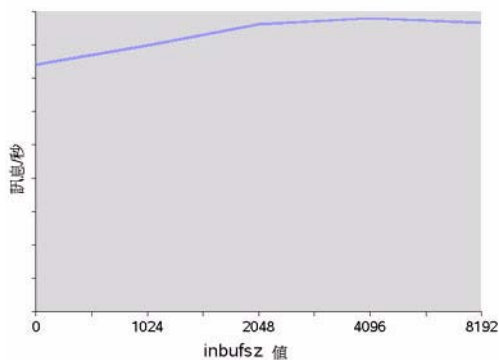
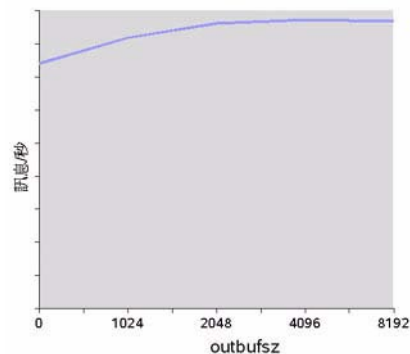


圖 9-8 顯示在 1K 的封包上變更 `outbufsz` 的結果。

圖 9-8 在 1K (1,024 個位元組) 封包上變更 `outbufsz` 的效果



HTTP/HTTPS 調整

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 的連接會保持暢通 (至 Web Server 的套接保持開啓狀態)，以便在 Web Server 處理多個 HTTP 請求時，最小化 Web Server 使用的資源。如果在同一個用戶端應用程式上，使用 JDK 1.4 版的效能比使用較早 JDK 版次的效能低，那麼您可能需要調整 Web Server 保持連接暢通的配置參數，以改善效能。

除了對 Web Server 進行此類調整，您也可以調整用戶端輪詢 Web Server 的頻率。HTTP 為基於請求的協定。這表示使用基於 HTTP 協定的用戶端會定期檢查 Web Server，查看是否有等待訊息。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 用戶端運行時間輪詢 Web Server 的頻率。

如果 pullPeriod 值為 -1 (預設值)，那麼用戶端運行時間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，每個用戶端連接會獨占 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，則用戶端運行時間會定期傳送請求到 Web Server，以查看是否有擱置的資料。在此情況下，用戶端不會獨占 Web Server 中的請求執行緒。因此，如果大量用戶端使用 Web Server，那麼您可能要將 pullPeriod 設為正值，以節省 Web Server 資源。

調整基於檔案的永久性倉庫

如需調整基於檔案的永久性倉庫，請參閱第 58 頁的「內建持續性」。

代理程式調整

以下各節描述您可對代理程式特性進行的調整，並藉以改善效能。

記憶體管理：增加代理程式負載的穩定性

可以在目標對目標級別或系統範圍級別 (用於所有目標，整體) 上配置記憶體管理。

使用目標限制

如需目標限制的資訊，請參閱第 152 頁的「管理目標」。

使用系統資源限制

如果訊息產生者嘗試過度執行訊息使用者，那麼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作用中記憶體之機制（請參閱第 55 頁的「管理記憶體資源和訊息流量」），所以對代理程式可保留的訊息總數目（和容量）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。請參閱第 117 頁的「編輯實例配置檔案」或第 124 頁的「`imqbrokerd` 選項的摘要」，以獲得設定代理程式特性的相關資訊。

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 個未發送 / 未確認的訊息。如果傳送額外訊息，則會被代理程式拒絕。如果為永久性訊息，那麼產生者在嘗試傳送訊息時會發生異常。如果為非永久性訊息，那麼代理程式會直接丟棄訊息。

若要非永久性訊息如同永久性訊息一樣傳回異常，那麼請在用戶端使用的連線工廠物件上設定以下特性：

```
imqAckOnProduce = true
```

以上設定可能會降低傳送非永久性訊息至代理程式（用戶端會在傳送下一個訊息前等待確認）的效能，但是，這情況通常是可接受的，因為流入代理程式的訊息通常不是系統瓶頸。

當傳送訊息時傳回異常，用戶端應稍作暫停並試著再傳送一次。

多重使用者佇列效能

多重佇列使用者在佇列目標中處理訊息的效率，取決於可配置的佇列目標屬性，即作用中使用者數目 (`maxNumActiveConsumers`)，以及在單一批次中可發送至使用者的最大訊息數目 (`consumerFlowLimit`)。第 154 頁的表 6-10 中描述了這些屬性。

若要達到最佳化訊息流量，必須有足以跟上用於佇列之訊息產生速率的作用中使用者數目，且必需路由佇列中的訊息，接著將訊息發送到作用中使用者。此方法會最大化訊息使用率。第 69 頁的「佇列發送至多個使用者」中描述平衡多重使用者上訊息發送的一般機制。

如果訊息累積在佇列中，那麼作用中使用者數目可能會不足以處理訊息負載。也有可能發生以下情形：以批次大小發送到使用者的訊息會在使用者上進行備份。舉例來說，如果批次大小 (`consumerFlowLimit`) 過大，一個使用者可能會接收佇列中的所有訊息，而其他作用中使用者會收不到任何訊息。如果使用者處理速度非常快，這可能不會發生問題。

但是，如果使用者的速度相對較慢，那麼您必須將訊息平均分散給使用者，如此一來，批次大小也會變小。批次大小越小，發送訊息至使用者所需的耗用時間就越長。然而，針對處理速度慢的使用者，網路效能改善比率通常會使用小型的批次大小。

用戶端運行時間訊息流量調整

本節描述影響效能的流量控制行爲（請參閱第 219 頁的「用戶端運行時間配置」）。請將這些行爲配置為連線工廠受管理物件的屬性。如需設定連線工廠屬性的資訊，請參閱第 7 章，「管理受管理物件」。

訊息流量計數

用戶端傳送和接收的訊息（JMS 訊息）與 Message Queue 控制訊息，皆會忽略相同用戶端與代理程式間的連接。如果 JMS 訊息發送阻擋控制訊息，則會造成控制訊息（例如代理程式確認）發送的延遲。若要避免此擁塞情形，Message Queue 要計算連接上的 JMS 訊息流量。

將 JMS 訊息分為幾個批次（以 `imqConnectionFlowCount` 特性指定），如此一來，會在發送批次後暫停 JMS 訊息發送並在發送擱置控制訊息後發送一組數字。控制訊息形成佇列後，發送 JMS 訊息的其他批次時會重複此循環。

如果用戶端正從代理程式要求許多確認（例如用戶端使用 `CLIENT_ACKNOWLEDGE` 或 `AUTO_ACKNOWLEDGE` 模式、永久性訊息、異動、佇列瀏覽器）或用戶端正在新增或移除使用者時，`imqConnectionFlowCount` 的值應保持較低。相反地，如果用戶端只有使用 `DUPS_OK_ACKNOWLEDGE` 模式來連接簡易使用者時，那麼您可以增加 `imqConnectionFlowCount` 的值，而不需顧及效能。

訊息流量限制

JMS 訊息的數目有一個限制，即在遇到本地資源限制（例如記憶體）前，Message Queue 用戶端運行時間可處理的 JMS 訊息數目限制。到達此限制時，會對效能造成影響。因此，Message Queue 可限制每個使用者可在連接中發送，且在用戶端運行時間內進行緩衝的待使用訊息數目（或每個連接的訊息數目）。

基於使用者的限制

如果任何使用者發送至用戶端運行時間的 JMS 訊息數超過 `imqConsumerFlowLimit` 的值時，則會停止發送訊息給該使用者。只有在發送給該使用者的未使用訊息數低於此 `imqConsumerFlowThreshold` 之設定值時，才恢復發送訊息。

以下範例說明這些限制的使用：考慮主題使用者的設定值

```
imqConsumerFlowLimit=1000
```

```
imqConsumerFlowThreshold=50
```

建立使用者時，代理程式會不間斷地發送包含 1,000 個訊息的初始批次 (假設訊息已存在) 給該使用者。傳送 1,000 個訊息後，代理程式會停止發送，直到用戶端運行時間要求發送其他訊息。應用程式處理這些訊息前，用戶端運行時間會保留這些訊息。接著，在要求代理程式傳送下一個批次前，用戶端運行時間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，用戶端運行時間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：

```
imqConsumerFlowLimit - ( 目前緩衝區中擱置的訊息數目 )
```

如此一來，如果 `imqConsumerFlowThreshold` 為 50%，那麼下一個批次大小介於 500 到 1,000 之間，其實際數量取決於應用程式處理訊息的速度。

如果 `imqConsumerFlowThreshold` 設得太高 (接近 100%)，那麼代理程式會嘗試傳送較小的批次，以降低訊息流量。如果值過低 (接近 0%)，那麼用戶端可能可以在代理程式發送下一組批次前，完成處理其餘的緩衝訊息，這會造成訊息流量下降。一般來說，除非您有特殊效能或可靠性考量，否則您不需變更 `imqConsumerFlowThreshold` 屬性的預設值。

基於使用者的流量控制 (特別是 `imqConsumerFlowLimit`) 為管理用戶端運行時間中的最佳方法。通常，根據用戶端應用程式，您會曉得您在任何連接上需要支援的使用者數目、訊息大小和可用於用戶端運行時間的總記憶體容量。

基於連接的限制

然而，在一些用戶端應用程式中，使用者數目可能是不正確的，它取決於一般使用者的選擇。在此情況下，您仍可以使用連接級別流量限制來管理記憶體。

連接級別流量控制可限制用於連接上所有使用者的總緩衝訊息數目。如果此數目超過 `imqConnectionFlowLimit`，那麼在總數目降低到低於連接限制前，會停止透過連接的訊息發送。(只有在您將 `imqConnectionFlowLimitEnabled` 特性設為 `true` 時，才可啓用 `imqConnectionFlowLimit`。)

階段作業中形成佇列的訊息數目為使用階段作業之使用者數目及用於每個使用者之訊息負載的函數。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使用者，或者分散較大連接數目上的階段作業，並因此改善效能。

Message Queue 資料的位置

Sun Java System Message Queue 會使用多個資料種類，根據作業系統的不同，每一個資料種類會儲存在不同的位置，以下各節將分別介紹。下列表格中，*instanceName* 可辨別與資料相關的代理程式實例名稱。

Solaris

表 A-1 顯示 Solaris 平台上 Message Queue 資料的位置。

備註 Solaris 上，隨附於 Sun Java System Application Server 之 Message Queue 的資料位置顯示在第 261 頁的表 A-3 中。

表 A-1 Solaris 上 Message Queue 資料的位置

資料種類	Solaris 上的位置
代理程式實例配置內容	<code>/var/imq/instances/instanceName/props/config.properties</code>
代理程式配置檔案範本	<code>/usr/share/lib/imq/props/broker/</code>
永久性倉庫 (訊息、目標、長期訂閱和異動)	<code>/var/imq/instances/instanceName/Es350/</code> 或 JDBC 可存取的資料倉庫
代理程式實例日誌檔目錄 (預設位置)	<code>/var/imq/instances/instanceName/log/</code>
受管理物件 (物件倉庫)	您選擇的本機目錄 或 LDAP 伺服器

表 A-1 Solaris 上 Message Queue 資料的位置 (續)

資料種類	Solaris 上的位置
安全性：使用者儲存庫	/var/imq/instances/instanceName/etc/passwd 或 LDAP 伺服器
安全性：存取控制檔 (預設位置)	/var/imq/instances/instanceName/etc/accesscontrol.properties
安全性：密碼檔案目錄 (預設位置)	/var/imq/instances/instanceName/etc/
安全性：範例密碼檔案	/etc/imq/passfile.sample
安全性：代理程式的密鑰儲存檔案位置	/etc/imq/
JavaDoc API 說明文件	/usr/share/javadoc/imq/index.html
範例應用程式與配置	/usr/demo/imq/
Java 歸檔 (.jar)、Web 歸檔檔案 (.war) 與資源介面歸檔檔案 (.rar) 檔案	/usr/share/lib/

Linux

表 A-2 顯示 Linux 平台上 Message Queue 資料的位置。

表 A-2 Linux 上 Message Queue 資料的位置

資料種類	Windows 上的位置
代理程式實例配置內容	/var/opt/imq/instances/instanceName/props/config.properties
代理程式配置檔案範本	/opt/imq/lib/props/broker/
永久性倉庫 (訊息、目標、長期訂閱和異動)	/var/opt/imq/instances/instanceName/fs350/ 或 JDBC 可存取的資料倉庫
代理程式實例日誌檔目錄 (預設位置)	/var/opt/imq/instances/instanceName/log/
受管理物件 (物件倉庫)	您選擇的本機目錄 或 LDAP 伺服器
安全性：使用者儲存庫	/var/opt/imq/instances/instanceName/etc/passwd 或 LDAP 伺服器

表 A-2 Linux 上 Message Queue 資料的位置 (續)

資料種類	Windows 上的位置
安全性：存取控制檔 (預設位置)	/var/opt/imq/instances/ <i>instanceName</i> /etc/accesscontrol.properties
安全性：密碼檔案目錄 (預設位置)	/var/opt/imq/instances/ <i>instanceName</i> /etc/
安全性：範例密碼檔案	/etc/opt/imq/passfile.sample
安全性：代理程式的密鑰儲存檔案位置	/etc/opt/imq/
JavaDoc API 說明文件	/opt/imq/javadoc/index.html
範例應用程式與配置	/opt/imq/demo/
Java 歸檔 (.jar)、Web 歸檔檔案 (.war) 與資源介面歸檔檔案 (.rar) 檔案	/opt/imq/lib/

Windows

表 A-3 顯示 Windows 平台及某些隨附於 Sun Java System Application Server 的 Message Queue 安裝上，Message Queue 資料的位置。如需更多資訊，請參閱第 26 頁的表 3，以及 IMQ_HOME 和 IMQ_VARHOME 的定義。

表 A-3 Windows 上 Message Queue 資料的位置

資料種類	Windows 上的位置
代理程式實例配置內容	IMQ_VARHOME\instances\ <i>instanceName</i> \props\config.properties
代理程式配置檔案範本	IMQ_HOME\lib\props\broker\
永久性倉庫 (訊息、目標、長期訂閱和異動)	IMQ_VARHOME\instances\ <i>instanceName</i> \fs350\ 或 JDBC 可存取的資料倉庫
代理程式實例日誌檔目錄 (預設位置)	IMQ_VARHOME\instances\ <i>instanceName</i> \log\
受管理物件 (物件倉庫)	您選擇的本機目錄 或 LDAP 伺服器
安全性：使用者儲存庫	IMQ_VARHOME\instances\ <i>instanceName</i> \etc\passwd 或 LDAP 伺服器

表 A-3 Windows 上 Message Queue 資料的位置 (續)

資料種類	Windows 上的位置
安全性：存取控制檔 (預設值)	IMQ_VARHOME\instances\instanceName\ 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：代理程式的密鑰儲存檔 案位置	IMQ_HOME\etc\
JavaDoc API 說明文件	IMQ_HOME\javadoc\index.html
範例應用程式與配置	IMQ_HOME\demo\
Java 歸檔 (.jar)、Web 歸檔檔案 (.war) 與資源介面歸檔檔案 (.rar) 檔案	IMQ_HOME\lib\

設定外掛持續性

本附錄介紹如何設定代理程式，以使用外掛持續性來存取 JDBC 可存取的資料倉庫。

介紹

Message Queue 代理程式包括「持續性管理程式」元件，可管理永久性資訊的寫入與擷取（請參閱第 57 頁的「持續性管理程式」）。依預設，此持續性管理程式已配置，可存取內建、基於檔案的資料倉庫，但是您可以重新配置它，以便透過 JDBC 相容的驅動程式插入任何可存取的資料倉庫。

若要配置代理程式以使用外掛持續性，您需要在代理程式實例配置檔案中設定一些與 JDBC 相關的特性。您還需要建立適當的資料庫綱目，以執行 Message Queue 持續性作業。Message Queue 提供會公用程式，即資料庫管理員 (mqdbmgr)，此程式使用您的 JDBC 驅動程式和代理程式配置特性以建立並管理外掛資料庫。

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

您可以在附錄 A，「[Message Queue 資料的位置。](#)」中顯示的範例位置找到 Oracle 與 PointBase 的範例配置。除此之外，實體配置檔案中會提供 PointBase 內嵌版本的範例、PointBase 伺服器版本、Oracle 與 Cloudscape 的範例作為註釋值。

插入 JDBC 可存取的资料倉庫

插入 JDBC 可存取的资料倉庫僅需要幾個步驟。

► 插入 JDBC 可存取的资料倉庫

1. 在代理程式的配置檔案中設定與 JDBC 相關的特性。
請參閱第 266 頁上的第表 B-1 段 中的特性說明文件。
2. 將您的 JDBC 驅動程式 JAR 檔案的副本或符號式連結放在以下路徑中：

在 Solaris 上為 /usr/share/lib/imq/ext/

在 Linux 上為 /opt/imq/lib/ext/

在 Windows 上為 IMQ_VARHOME\lib\ext

副本範例(Solaris)：

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

符號式連結範例(Solaris)：

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

3. 建立 Message Queue 持續性所需的資料庫綱目。

使用 `imqdbmgr create all` 指令 (對於內嵌式資料庫) 或 `imqdbmgr create tbl` 指令 (對於外部資料庫)。請參閱第 269 頁的「[資料庫管理員公用程式 \(imqdbmgr\)](#)」。

範例：

- a. 變更 `imqdbmgr` 常駐的目錄。

在 Solaris 上為 `cd /usr/bin`

在 Linux 上為 `cd /opt/imq/bin`

在 Windows 上為 `cd IMQ_HOME/bin`

- b. 輸入 `imqdbmgr` 指令。

```
imqdbmgr create all
```


備註 如果已使用內嵌式資料庫，則建議在以下目錄下建立此資料庫：

```
.../instances/instanceName/dbstore/databatbaseName。
```

如果內嵌式資料庫未由使用者名稱和密碼保護，則可能由檔案系統許可權保護。若要確保代理程式可讀取和寫入資料庫，則執行此代理程式的使用者應與使用 `imqdbmgr` 指令建立內嵌式資料庫的使用者相同（請參閱第 269 頁的「[資料庫管理員公用程式 \(imqdbmgr\)](#)」）。

與 JDBC 相關的代理程式配置特性

代理程式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 資料的位置](#)」）：

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

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

表 B-1 描述了當您插入 JDBC 可存取的資料倉庫時，需要設定的配置特性。您可以使用外掛持續性在每個代理程式實例的實例配置檔案 (`config.properties`) 中設定這些特性。

這些實例配置特性可讓您自訂建立 Message Queue 資料庫綱要的 SQL 程式碼：有一個指定 SQL 程式碼的可配置特性，此 SQL 程式碼可建立資料庫表格。需要這些特性來正確指定外掛資料庫使用的資料類型。

因為與 SQL 語法相關的資料庫供應商之間會有不相容的情形，所以請確切檢查您資料庫供應商提供的相對說明文件，並隨之調整**表 B-1** 中的特性。例如，使用 PointBase 資料庫時，您可能需要將 IMQMSG35 表格中的 MSG 欄寬調整到最大（請參閱 `imq.persist.jdbc.table.IMQMSG35` 特性）。

表 B-1 包括您要指定用於 PointBase DBMS 範例的值。

表 B-1 與 JDBC 相關的特性

特性名稱	說明
<code>imq.persist.store</code>	指定基於檔案或基於 JDBC 的資料倉庫。 <i>範例：</i> <code>jdbc</code>
<code>imq.persist.jdbc.brokerid</code> (可選)	指定附加至資料庫表格名稱的代理程式實例識別碼，以便在多個代理程式實例使用相同的資料庫作為永久性資料倉庫的情況下，這些資料庫表格名稱均是唯一的。(由於內嵌式資料庫僅為一個代理程式實例儲存資料，因此通常不需要指定識別碼。)識別碼必須為字母數字字串，其長度不超過資料庫所允許的最長表格名稱長度減去 12。 <i>範例：</i> <code>not needed for PointBase embedded version.</code>
<code>imq.persist.jdbc.driver</code>	指定 JDBC 驅動程式的 Java 類別名稱，以連接至資料庫。 <i>範例：</i> <code>com.pointbase.jdbc.jdbcUniversalDriver</code>
<code>imq.persist.jdbc.opendburl</code>	指定資料庫 URL，以開啓與現有資料庫的連接。 <i>範例：</i> <code>jdbc:pointbase:embedded:dbName;</code> <code>database.home= Ö/instances/instanceName/dbstore</code>
<code>imq.persist.jdbc.createdburl</code> (可選)	指定資料庫 URL，以開啓連接建立資料庫。(僅在使用 <code>imqdbmgr</code> 建立資料庫時指定。) <i>範例：</i> <code>jdbc:pointbase:embedded:dbName;new,</code> <code>database.home= Ö/instances/instanceName/dbstore</code>
<code>imq.persist.jdbc.closedburl</code> (可選)	指定資料庫 URL，以在關閉代理程式時關閉目前資料庫連接。 <i>範例：</i> <code>not required for PointBase</code>
<code>imq.persist.jdbc.user</code> (可選)	如有必要，則指定用於開啓資料庫連接的使用者名稱。考慮到安全性因素，可以指定此值，而不是使用指令行選項： <code>imqbrokerd -dbuser</code> 和 <code>imqdbmgr -u</code>
<code>imq.persist.jdbc.needpassword</code> (可選)	指定資料庫是否需要密碼以存取代理程式。值 <code>true</code> 表示需要密碼。可以使用以下指令行選項指定密碼： <code>imqbrokerd -dbpassword</code> <code>imqdbmgr -p</code> 如果未使用指令行選項或密碼檔案提供密碼 (請參閱第 204 頁的「使用密碼檔案」)，則代理程式將提示您提供密碼。

表 B-1 與 JDBC 相關的特性 (續)

特性名稱	說明
imq.persist.jdbc.password (可選)	如有必要，則指定用於開啓資料庫連接的密碼。此特性僅能在密碼檔案中指定 (請參閱第 204 頁的「使用密碼檔案」)。 有數種方法可以提供密碼。最安全的方法是讓代理程式提示您提供密碼。較不安全的方法是使用密碼檔案，並讀取保護密碼檔案。最不安全的方法是使用以下指令行選項指定密碼： imqbrokerd -dbpassword imqdbmgr -p
imq.persist.jdbc.table.IMQSV35	用來建立版本表格的 SQL 指令。 <i>範例：</i> CREATE TABLE \${name} (STOREVERSION INTEGER NOT NULL, BROKERID VARCHAR(100))
imq.persist.jdbc.table.IMQCCREC35	用來建立配置變更記錄表格的 SQL 指令。 <i>範例：</i> CREATE TABLE \${name} (RECORDTIME BIGINT NOT NULL, RECORD BLOB(10k))
imq.persist.jdbc.table.IMQDEST35	用來建立目標表格的 SQL 指令。 <i>範例：</i> CREATE TABLE \${name} (DID VARCHAR(100) NOT NULL, DEST BLOB(10k), primary key(DID))
imq.persist.jdbc.table.IMQINT35	用來建立偏好表格的 SQL 指令。 <i>範例：</i> CREATE TABLE \${name} (CUID BIGINT NOT NULL, INTEREST BLOB(10k), primary key(CUID))
imq.persist.jdbc.table.IMQMSG35	用來建立訊息表格的 SQL 指令。 <i>範例：</i> CREATE TABLE \${name} (MID VARCHAR(100) NOT NULL, DID VARCHAR(100), MSGSIZE BIGINT, MSG BLOB(1m), primary key(MID)) MSG 的預設最大欄寬為 1 Megabyte (1m)。如果您預期訊息長度會大於此長度，請相應地設定長度。如果已建立表格，您則必須重新建立表格來進行變更。
imq.persist.jdbc.table.IMQPROPS35	用來建立特性表格的 SQL 指令。 <i>範例：</i> CREATE TABLE \${name} (PROPNAME VARCHAR(100) NOT NULL, PROPVALUE BLOB(10k), primary key(PROPNAME))

表 B-1 與 JDBC 相關的特性 (續)

特性名稱	說明
imq.persist.jdbc.table.IMQILIST35	用來建立狀態表格的 SQL 指令。 範例： CREATE TABLE \${name} (MID VARCHAR(100) NOT NULL, CUID BIGINT, DID VARCHAR(100), STATE INTEGER, primary key(MID, CUID))
imq.persist.jdbc.table.IMQTXN35	用來建立事務處理表格的 SQL 指令。 範例： CREATE TABLE \${name} (TUID BIGINT NOT NULL, STATE INTEGER, TSTATEOBJ BLOB(10K), primary key(TUID))
imq.persist.jdbc.table.IMQTACK35	用來建立事務處理確認表格的 SQL 指令。 範例： CREATE TABLE \${name} (TUID BIGINT NOT NULL, TXNACK BLOB(10k))

如同所有的代理程式配置特性，可以使用 `-D` 指令行選項設定值。如果資料庫需要設定某些資料庫特定的特性，則當啟動代理程式 (`imqbrokerd`) 或資料庫管理員公用程式 (`imqdbmgr`) 時，還可以使用 `-D` 指令行選項設定這些值。

範例：

對於 PointBase 內嵌式資料庫範例，並非在資料庫連接 URL (如表 B-1 範例中所示的那些 URL) 中指定絕對路徑，`-D` 指令行選項可用於定義 PointBase 系統目錄：

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

在這種情況下，要建立和開啓資料庫的 URL 僅可分別指定為：

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

和

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

資料庫管理員公用程式 (imqdbmgr)

Message Queue 提供有資料庫管理員公用程式 (imqdbmgr)，以設定持續性所需的綱目。如果 Message Queue 資料庫表格損毀或者您想使用其他資料庫作為資料倉庫，還可此公用程式來刪除這些表格。

本節描述基本的 imqdbmgr 指令語法、提供子指令的清單，並總結 imqdbmgr 指令選項。

mqdbmgr 指令的語法

imqdbmgr 指令的一般語法如下：

```
imqdbmgr subcommand argument [options]
imqdbmgr -h|-help
imqdbmgr -v|-version
```

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

```
imqdbmgr create all -v
```

imqdbmgr 子指令

資料庫管理員公用程式 (imqdbmgr) 包括表 B-2 中列出的子指令：

表 B-2 imqdbmgr 子指令

子指令和引數	說明
create all	建立新資料庫和 Message Queue 永久性倉庫綱目。此指令可在內嵌式資料庫系統上使用，當使用此指令時，需要指定特性 <code>imq.persist.jdbc.createdburl</code> 。
create tbl	在現有資料庫系統中建立 Message Queue 永久性倉庫綱目。此指令可在外部資料庫系統上使用。
delete tbl	在目前永久性倉庫資料庫中刪除現有 Message Queue 資料庫表格。
delete oldtbl	刪除所有較早版本之永久性倉庫資料庫中的 Message Queue 資料庫表格。在自動將永久性倉庫轉移到 Message Queue 目前版本後使用。
recreate tbl	在目前永久性倉庫資料庫中刪除現有 Message Queue 資料庫表格，然後重新建立 Message Queue 永久性倉庫綱目。

表 B-2 imqdbmgr 子指令 (續)

子指令 和引數	說明
reset lck	重設鎖定狀態，以便其他程序可以使用永久性倉庫資料庫。

imqdbmgr 指令選項摘要

表 B-3 列出了 imqdbmgr 指令的選項。

表 B-3 imqdbmgr 選項

選項	說明
-D <i>property=value</i>	將指定的特性設定為指定的值。
-b <i>instanceName</i>	指定代理程式實例名稱並使用相應的實例配置檔案。
-h	顯示用法說明。不執行指令行上的其他任何指令。
-p <i>password</i>	指定資料庫密碼。
-u <i>name</i>	指定資料庫使用者名稱。
-v	顯示版本資訊。不執行指令行上的其他任何指令。

資料庫管理員公用程式 (imqdbmgr)

HTTP/HTTPS 支援 (企業版)

Message Queue 企業版 (請參閱第 33 頁的「產品版本」) 包括 HTTP 與 HTTPS 連接的支援。(HTTPS 為安全套接層 - SSL - 上 HTTP 的傳輸連接。) 此支援可讓用戶端應用程式使用 HTTP 協定 (並非直接的 TCP 連接) 與代理程式通訊。本附錄描述用於啟用此支援的架構，並介紹允許用戶端使用基於 HTTP 的連接進行 Message Queue 訊息傳送所需的設定工作。

備註 HTTP/HTTPS 支援可用於 Java 用戶端，但不適用於 C 用戶端。

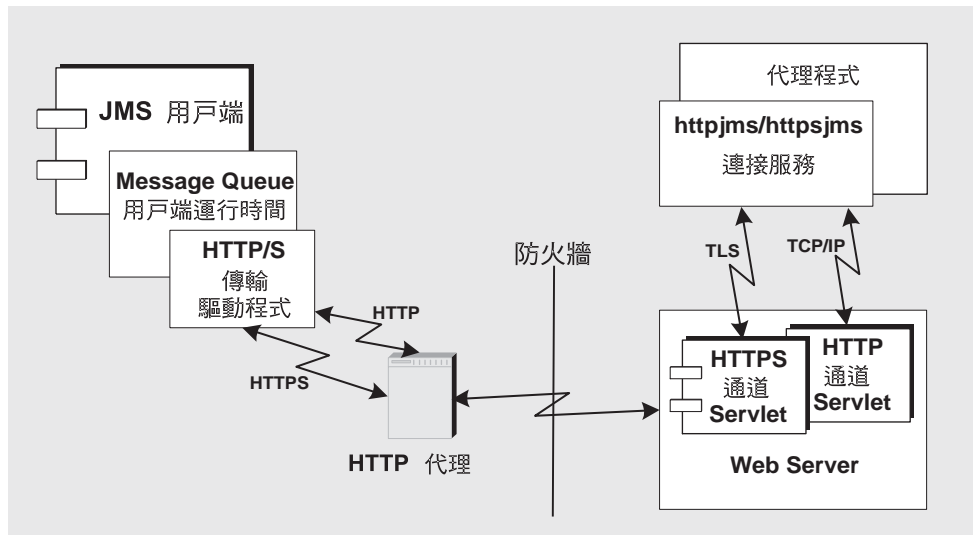
HTTP/HTTPS 支援架構

Message Queue 訊息傳送可在 HTTP/HTTPS 連接的頂層執行。由於通常可允許 HTTP/HTTPS 連接通過防火牆，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 274 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 傳輸驅動程式將 Message Queue 訊息封裝在 HTTP 請求中，並確保這些請求以正確的序列傳送至 Web Server。
- 如有必要，用戶端可使用 HTTP 代理伺服器與代理程式進行通訊。代理的位址在啟動用戶端時使用命令行選項指定。請參閱第 278 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) 被載入 Web Server 中，並用於在將 JMS 訊息轉寄至代理程式之前，將它們從用戶端 HTTP 請求中移除。HTTP/HTTPS 通道 Servlet 也會將代理程式訊息傳送回用戶端，以確認用戶端發出的 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 連接服務並多路傳輸從相應通道 Servlet 進來的訊息。
- 如果 Web Server 發生故障並重新啓動，則所有連接均會復原且不影響用戶端。如果代理程式發生故障並重新啓動，則系統會拋出異常，且用戶端必須重新建立它們的連接。如果 Web Server 與代理程式均發生故障，且未重新啓動代理程式 (這種情況不太可能發生)，則 Web Server 會復原用戶端連接並繼續等待代理程式連接，且不通知用戶端。若要避免這種情況，請務必重新啓動代理程式。

如圖 C-1 中所示，HTTP 和 HTTPS 支援的架構非常相似。主要區別在於，對於 HTTPS (httpsjms 連接服務)，通道 Servlet 具有與用戶端應用程式和代理程式的安全連接。

可透過已啓用 SSL 的通道 Servlet (Message Queue 的 HTTPS 通道 Servlet) 提供與代理程式的安全連接。此通道 Servlet 將自身簽名的憑證傳送至請求連接的任何代理程式。代理程式使用此憑證來設定與 HTTPS 通道 Servlet 的加密連接。一旦建立此連接，用戶端應用程式與通道 Servlet 之間的安全連接可由用戶端應用程式和 Web Server 協商。

啓用 HTTP 支援

以下各節描述啓用 HTTP 支援所需採取的步驟。

► 啟用 HTTP 支援

1. 在 Web Server 上部署 HTTP 通道 Servlet。
2. 配置代理程式的 `httpjms` 連接服務並啟動代理程式。
3. 配置 HTTP 連接。

步驟 1. 在 Web Server 上部署 HTTP 通道 Servlet

在 Web Server 上部署 HTTP 通道 Servlet 有兩種常用方法：

- 將其部署為 JAR 檔案 - 用於支援 Servlet 2.1 或較低版本的 Web Server
- 將其部署為 Web 歸檔 (WAR) 檔案 - 用於支援 Servlet 2.2 或較高版本的 Web Server

部署為 JAR 檔案

部署 Message Queue 通道 Servlet 由三個部分組成：使主機 Web Server 存取適當的 JAR 檔案，配置此 Web Server 以在啟動時載入 Servlet，以及指定 Servlet URL 的環境根式部分。

通道 Servlet JAR 檔案 (`imqServlet.jar`) 包含 HTTP 通道 Servlet 需要的所有類別，您可以在某個目錄（目錄位置因作業系統而異）中找到這個檔案（請參閱[附錄 A 「Message Queue 資料的位置」](#)）。

支援 Servlet 2.x 的任何 Web Server 均可用於載入此 Servlet。Servlet 類別名稱為：

```
com.sun.messaging.jmq.transport.  
httpunnel.servlet.HttpTunnelServlet
```

Web Server 必須可查看 `imqServlet.jar` 檔案。如果您想在不同的主機上執行 Web Server 和代理程式，則應將 `imqServlet.jar` 檔案的副本放在 Web Server 可存取的位置。

您也必須配置 Web Server，以便在啟動時載入此 Servlet。另外，您可能必須指定 Servlet URL 的環境根式部分（請參閱第 279 頁的「[範例 1：在 Sun Java System Web Server 上部署 HTTP 通道 Servlet](#)」）。

為了提昇效能，我們還建議您停用 Web Server 的存取記錄功能。

部署為 Web 歸檔檔案

將 HTTP 通道 Servlet 部署為 WAR 檔案，需要使用 Web Server 提供的部署機制。HTTP 通道 Servlet WAR 檔案 (imqhttp.war) 位於包含 .JAR、.WAR 與 .RAR 檔案的目錄中，目錄位置因作業系統而異 (請參閱附錄 A 「Message Queue 資料的位置」)。

WAR 檔案包括部署描述元，此描述元包含 Web Server 載入和執行 Servlet 所需的基本配置資訊。因 Web Server 的不同，您也可能必須指定 Servlet URL 的環境根式部分 (請參閱第 282 頁的「範例 2：在 Sun Java System Application Server 7.0 上部署 HTTP 通道 Servlet」)。

步驟 2. 配置 httpjms 連接服務

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 連接服務。一旦重新配置，此代理程式便可啓動，如第 122 頁的「啓動代理程式」中所述。

► 啓動 httpjms 連接服務

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 資料的位置」)：

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

2. 將 httpjms 值新增至 imq.service.activelist 特性：

```
imq.service.activelist=jms,admin,httpjms
```

啓動時，代理程式會查找在其主機上執行的 Web Server 和 HTTP 通道 Servlet。但是，若要存取遠端通道 Servlet，您可以重新配置 `servletHost` 和 `servletPort` 連接服務特性。

您還可以重新配置 `pullPeriod` 特性，以提昇效能。第 276 頁的表 C-1 中詳細介紹了 httpjms 連接服務配置特性。

表 C-1 httpjms 連接服務特性

特性名稱	說明
imq.httpjms.http.servletHost	如有必要，請變更此值，以指定 HTTP 通道 Servlet 正在其上執行的主機 (主機名稱或 IP 位址) 的名稱。(這可以為遠端主機，或本地主機上的特定主機名稱。) 預設值：localhost

表 C-1 httpjms 連接服務特性 (續)

特性名稱	說明
imq.httpjms.http.servletPort	變更此值，以指定代理程式用於存取 HTTP 通道 Servlet 的連接埠號。(如果 Web Server 上的預設連接埠已變更，則您必須相應地變更此特性。)預設值：7675
imq.httpjms.http.pullPeriod	指定用戶端發出從代理程式移除訊息的 HTTP 請求的間隔時間 (以秒為單位)。(請注意，此特性於代理程式上進行設定，且傳遞至用戶端執行階段。)如果值為零或負數，則用戶端將始終擱置一個 HTTP 請求，以準備儘快移除訊息。若用戶端數目過大，可能會導致 Web Server 資源大量外流，且伺服器可能沒有確認。在此類情況下，您應將 pullPeriod 特性設定為正的秒數。此作業將設定用戶端的 HTTP 傳輸驅動程式在發出後續移除請求之前等待的時間。將此值設定為正數可保存 Web Server 資源，但會影響用戶端所監視的確認次數。預設值：-1
imq.httpjms.http.connectionTimeout	指定用戶端執行階段在系統拋出異常前，等待從 HTTP 通道 Servlet 發出確認的時間 (以秒為單位)。(請注意，此特性於代理程式上進行設定，且傳遞至用戶端執行階段。)此特性也會指定中斷連接前，代理程式與 HTTP 通道 Servlet 進行通訊後，代理程式等待的時間。因為代理程式與通道 Servlet 無法得知存取 HTTP Servlet 的用戶端是否已異常中斷，在此情況下，必須設定逾時時間。預設值：60

步驟 3. 配置 HTTP 連接

用戶端應用程式必須使用已適當配置的連線工廠受管理物件，以建立與代理程式的 HTTP 連接。本節說明 HTTP 連接配置問題。

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:port/contextRoot/tunnel
```

hostName:port 為代管 HTTP 通道 Servlet 之 Web Server 的名稱和連接埠；contextRoot 為在 Web Server 上部署通道 Servlet 的路徑設定。

如需連線工廠屬性的一般資訊與 imqAddressList 屬性的特定資訊，請參閱「Message Queue Java 用戶端開發人員指南」。

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 176 頁的「新增連線工廠」) 之 imqobjmgr 指令的 -o 選項，或在使用管理主控台 (imqadmin) 建立連線工廠受管理物件時設定屬性。

- 使用可啟動用戶端的指令的 `-D` 選項 (請參閱「Message Queue Java 用戶端開發人員指南」)。
- 當您有計劃藉由用戶端程式碼建立連線工廠後，請使用 API 呼叫設定它的屬性 (請參閱「Message Queue Java 用戶端開發人員指南」)。

使用單一 Servlet 存取多重代理程式

如果您要執行多重代理程式，則無需配置多重 Web Server 和 Servlet 實例。您可以在同時執行的代理程式中共用單一 Web Server 和 HTTP 通道 Servlet 實例。如果多重代理程式實例正共用一個通道 Servlet，您必須配置 `imqAddressList` 連線工廠屬性，配置內容如下：

```
http://hostName:port/contextRoot/tunnel?ServerName=bkrHostName:instanceName
```

`bkrHostName` 為代理程式實例主機名稱；`instanceName` 為您用戶端存取的特定代理程式實例名稱。

若要檢查您是否已為 `bkrHostName` 和 `instanceName` 輸入正確的字串，則應透過從瀏覽器存取 Servlet URL 產生 HTTP 通道 Servlet 的狀態報告。此報告將列出 Servlet 要存取的所有代理程式：

```
HTTP tunnel servlet ready.
Servlet Start Time : Thu May 30 01:08:18 PDT 2002
Accepting TCP connections from brokers on port : 7675
Total available brokers = 2
Broker List :
  jpgserv:broker2
  cochin:broker1
```

使用 HTTP 代理

如果您要使用 HTTP 代理存取 HTTP 通道 Servlet：

- 將 `http.proxyHost` 系統特性設定為代理伺服器主機名稱。
- 將 `http.proxyPort` 系統特性設定為代理伺服器連接埠號。

您可以使用可啟動用戶端應用程式之指令的 `-D` 選項來設定這些特性。

範例 1：在 Sun Java System Web Server 上部署 HTTP 通道 Servlet

本節描述如何在 Sun Java System Web Server 上將 HTTP 通道 Servlet 部署為 JAR 檔案和 WAR 檔案。您使用的方法取決於 Sun Java System Web Server 的版本：如果它不支援 Servlet 2.2 或較高版本，它將無法處理 WAR 檔案部署。

部署為 JAR 檔案

以下說明參考了使用基於瀏覽器的管理 GUI 之 Sun Java System Web Server 6.1 上的部署。此程序包含以下常用步驟：

1. 新增 Servlet
2. 配置 Servlet 虛擬路徑
3. 載入 Servlet
4. 停用 Servlet 存取記錄

這些步驟將在以下各小節中描述。您可以透過使用 Web 瀏覽器存取 Servlet URL 來驗證成功的 HTTP 通道 Servlet 部署。它應顯示狀態資訊。

新增 Servlet

► 新增通道 Servlet

1. 選取 [Servlets] 標籤。
2. 選擇 [配置 Servlet 屬性]。
3. 在 [Servlet 名稱] 欄位中，指定通道 Servlet 的名稱。
4. 將 [Servlet 程式碼 (類別名稱)] 欄位設定為以下值：


```
com.sun.messaging.jmq.transport.  
httptunnel.servlet.HttpTunnelServlet
```
5. 在 [Servlet 類別路徑] 欄位中輸入 imqservlet.jar 的完整路徑。例如：
 - 在 Solaris 上為 /usr/share/lib/imq/imqservlet.jar
 - 在 Linux 上為 /opt/imq/lib/imqservlet.jar
 - 在 Windows 上為 IMQ_HOME/lib/imqservlet.jar
6. 在 [Servlet 引數] 欄位中，輸入任何選擇性引數，如表 C-2 中所示：

表 C-2 用於部署 HTTP 通道 Servlet JAR 檔案的 Servlet 引數

引數	預設值	參考
servletHost	所有主機	請參閱第 276 頁的表 C-1
servletPort	7675	請參閱第 276 頁的表 C-1

如果要使用這兩個引數，請使用逗號將它們分隔：

```
servletPort= 連接埠號, servletHost=...
```

servletHost 和 servletPort 引數僅適用於 Web Server 和代理程式之間的通訊，並且僅在預設值有問題時才設定。但是，在此情況下，您還必須相應地設定代理程式配置特性（請參閱第 276 頁的表 C-1），例如：

```
imq.httpjms.http.servletPort
```

配置 Servlet 虛擬路徑 (Servlet URL)

► 若要配置通道 Servlet 的虛擬路徑 (Servlet URL)

1. 選取 [Servlets] 標籤。
2. 選擇 [配置 Servlet 虛擬路徑轉換]。
3. 設定 [虛擬路徑] 欄位。

「虛擬路徑」為通道 Servlet URL 的 `/contextRoot/tunnel` 的一部份：

```
http://hostName:port/contextRoot/tunnel
```

舉例來說，如果您將 `contextRoot` 設定至 `imq`，那麼 [虛擬路徑] 欄位則為：

```
/imq/tunnel
```

4. 將 [Servlet 名稱] 欄位設定為與第 279 頁的「新增 Servlet」的步驟 3 中相同的值。

載入 Servlet

► 在啟動 Web Server 時載入通道 Servlet

1. 選取 [Servlets] 標籤。
2. 選擇 [配置全域屬性]。
3. 在 [啟動 Servlet] 欄位中，輸入與第 279 頁的「新增 Servlet」的步驟 3 中相同的 Servlet 名稱值。

停用伺服器存取記錄

您不必停用伺服器存取記錄，但是如果您停用記錄，將會獲得更好的效能。

► 停用伺服器存取記錄

1. 選取 [狀態] 標籤。
2. 選擇 [記錄個人喜好頁面]。
3. 使用記錄用戶端存取控制來停用記錄

部署為 WAR 檔案

以下說明參考了 Sun Java System Web Server 6.0 Service Pack 2 上的部署。您可以透過使用 Web 瀏覽器存取 Servlet URL 來驗證成功的 HTTP 通道 Servlet 部署。它應顯示狀態資訊。

► 將 HTTP 通道 Servlet 部署為 WAR 檔案

1. 在基於瀏覽器的管理 GUI 中，選取 [虛擬伺服器類別] 標籤並選取 [管理類別]。
2. 選取適當的虛擬伺服器類別名稱 (例如 defaultClass)，然後按一下 [管理] 按鈕。
3. 選取 [管理虛擬伺服器]。
4. 選取適當的虛擬伺服器名稱，然後按一下 [管理] 按鈕。
5. 選取 [Web 應用程式] 標籤。
6. 按一下 [部署 Web 應用程式]。
7. 為 [WAR 檔案開啓] 和 [WAR 檔案路徑] 欄位選取適當的值，以便指向 `imqhttp.war` 檔案的位置。您可以在某個目錄 (目錄位置因作業系統而異) 中找到這個檔案 (請參閱 附錄 A 「Message Queue 資料的位置」)。
8. 在 [應用程式 URI] 欄位中輸入路徑。

[應用程式 URI] 欄位值為通道 Servlet URL 的 `/contextRoot` 的一部份：

```
http://hostName:port/contextRoot/tunnel
```

舉例來說，如果您將 `contextRoot` 設定至 `imq`，那麼 [應用程式 URI] 欄位則為：

```
/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 可從以下位址獲得：

```
http://hostName:port/contextRoot/tunnel
```

此時，用戶端可使用此 URL，以使用 HTTP 連接來連接至訊息服務。

範例 2：在 Sun Java System Application Server 7.0 上部署 HTTP 通道 Servlet

本節描述如何在 Sun Java System Application Server 7.0 上將 HTTP 通道 Servlet 部署為 WAR 檔案。

您需要執行兩個步驟：

- 使用 Application Server 7.0 部署工具來部署 HTTP 通道 Servlet
- 修改應用程式伺服器實例的 `server.policy` 檔案

使用部署工具

► 在 Application Server 7.0 環境中部署 HTTP 通道 Servlet

1. 在基於瀏覽器的管理 GUI 中選取
應用程式伺服器 > 實例 > server1 > 應用程式 > Web 應用程式。
2. 按一下 [部署] 按鈕。
3. 在 [檔案路徑:] 文字欄位輸入 HTTP 通道 Servlet WAR 檔案 (imqhttp.war) 的位置。

imqhttp.war 檔案的位置會因您的作業系統而有差異 (請參閱 [附錄 A](#) 「[Message Queue 資料的位置](#)」)。

4. 按一下 [確定]。
5. 在下一個畫面，於 [環境根式] 文字欄位中設定數值。

[環境根式] 欄位值為通道 Servlet URL 的 `/contextRoot` 的一部份：

```
http://hostName:port/contextRoot/tunnel
```

例如，您可以將 [環境根式] 欄位設定為：

```
/imq
```

6. 按一下 [確定]。

依預設，接著會出現下一個畫面，表示通道 Servlet 已成功部署在：

```
/var/opt/SUNWappserver7/domains/domain1/server1/applications/  
j2ee-modules/imqhttp_1
```

此時，Servlet 可從以下位址獲得：

```
http://hostName:port/contextRoot/tunnel
```

此時，用戶端可使用此 URL，以使用 HTTP 連接來連接至訊息服務。

修改 server.policy 檔案

Application Server 7.0 會強制一組預設安全策略，除非修改的檔案可以阻止 HTTP 通道 Servlet 接受來自 Message Queue 代理程式的連接。

每個應用程式伺服器實例，皆有一個包含安全策略或規則的檔案。例如，Solaris 上用於 server1 實例的檔案位置為：

```
/var/opt/SUNWappserver7/domains/domain1/server1/config/  
server.policy
```

若要讓通道 Servlet 接受來自 Message Queue 代理程式的連接，則必須在此檔案新增其他項目。

► 若要修改 Application Server 的伺服器策略檔案

1. 開啓伺服器策略檔案。
2. 新增以下項目：

```
grant codeBase  
"file:/var/opt/SUNWappserver7/domains/domain1/server1/  
applications/j2ee-modules/imqhttp_1/-"  
{  
    permission java.net.SocketPermission "*",  
        "connect,accept,resolve";  
};
```

啓用 HTTPS 支援

以下各節描述啓用 HTTPS 支援所需採取的步驟。它們與第 274 頁的「啓用 HTTP 支援」中的那些步驟，以及產生和存取 SSL 憑證所需的步驟相似。

► 啟用 HTTPS 支援

1. 為 HTTPS 通道 Servlet 產生自身簽名的憑證。
2. 在 Web Server 上部署 HTTPS 通道 Servlet。
3. 配置代理程式的 `httpsjms` 連接服務並啟動代理程式。
4. 配置 HTTPS 連接。

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` 連接服務架構中，HTTPS 通道 Servlet 是代理程式與應用程式用戶端的伺服器。

執行 `imqkeytool` 公用程式，以便為通道 Servlet 產生自身簽名的憑證。在命令提示字元下輸入以下內容：

```
imqkeytool -servlet keystore_location
```

此公用程式將提示您提供它所需的資訊。(在 Unix 系統上，為了擁有建立密鑰儲存的許可權，您可能需要以超級使用者 (`root`) 的身份執行 `imqkeytool`。)

首先，`imqkeytool` 會提示您提供密鑰儲存密碼以及某些組織資訊，然後提示您確認。接收到確認之後，它會在產生鍵對時暫停。然後它會要求您提供密碼，以鎖定特定的鍵對 (鍵密碼)；您應按 [Return] 以確認此提示：這會使鍵密碼與密鑰儲存密碼相同。

備註 請記住您所提供的密碼 -- 稍後您需要將此密碼提供給通道 Servlet，以便它可開啓密鑰儲存。

執行 `imqkeytool` 可執行 JDK `keytool` 公用程式，以產生自身簽名的憑證，並將其放在位於 `keystore_location` 引數所指定的 Message Queue 密鑰儲存檔案中。(此密鑰儲存的格式與 JDK1.2 `keytool` 支援的格式相同。)

備註 HTTPS 通道 Servlet 必須可查看此密鑰儲存。請確定您將位於 `keystore_location` 中已產生的密鑰儲存移動 / 複製到 HTTPS 通道 Servlet 可存取的位置 (請參閱第 285 頁的「[步驟 2. 在 Web Server 上部署 HTTPS 通道 Servlet](#)」)。

步驟 2. 在 Web Server 上部署 HTTPS 通道 Servlet

在 Web Server 上部署 HTTPS 通道 Servlet 有兩種常用方法：

- 將其部署為 JAR 檔案 - 用於支援 Servlet 2.1 或較低版本的 Web Server
- 將其部署為 Web 歸檔 (WAR) 檔案 - 用於支援 Servlet 2.2 或較高版本的 Web Server

在這兩種情況下，您都應確保為 Web Server 啟動加密，從而讓用戶端與代理程式之間進行端對端安全通訊。

部署為 JAR 檔案

部署 Message Queue 通道 Servlet 由三個部分組成：使主機 Web Server 存取適當的 JAR 檔案，配置此 Web Server 以在啟動時載入 Servlet，以及指定 Servlet URL 的環境根式部分。

通道 Servlet JAR 檔案 (`imqservlet.jar`) 包含 HTTPS 通道 Servlet 需要的所有類別，您可以在某個目錄 (目錄位置因作業系統而異) 中找到這個檔案 (請參閱[附錄 A 「Message Queue 資料的位置」](#))。

支援 Servlet 2.x 的任何 Web Server 均可用於載入此 Servlet。Servlet 類別名稱為：

```
com.sun.messaging.jmq.transport.  
httpunnel.servlet.HttpsTunnelServlet
```

Web Server 必須可查看 `imqservlet.jar` 檔案。如果您想在不同的主機上執行 Web Server 和代理程式，則應將 `imqservlet.jar` 檔案的副本放在 Web Server 可存取的位置。

您也必須配置 Web Server，以便在啟動時載入此 Servlet。另外，您可能必須指定 Servlet URL 的環境根式部分 (請參閱第 290 頁的「[範例 3：在 Sun Java System Web Server 上部署 HTTPS 通道 Servlet](#)」)。

確保 JSSE JAR 檔案在類別路徑中，以便在 Web Server 中執行 Servlet。檢查 Web Server 的說明文件，以便瞭解如何執行此作業。

配置 Web Server 的一個重要方面是指定自身簽名憑證的位置和密碼，以便 HTTPS 通道 Servlet 使用此憑證來建立與代理程式的安全連接。您必須將第 284 頁的「[步驟 1. 為 HTTPS 通道 Servlet 產生自身簽名的憑證](#)」中建立的密鑰儲存在 HTTPS 通道 Servlet 可存取的位置。

爲了提昇效能，我們還建議您停用 Web Server 的存取記錄功能。

部署爲 Web 歸檔檔案

將 HTTPS 通道 Servlet 部署爲 WAR 檔案，需要使用 Web Server 提供的部署機制。HTTPS 通道 Servlet WAR 檔案 (imqhttps.war) 位於您作業系統的某個目錄中，目錄位置因作業系統而異 (請參閱 [附錄 A 「Message Queue 資料的位置」](#))。

WAR 檔案包括部署描述元，此描述元包含 Web Server 載入和執行 Servlet 所需的基本配置資訊。因 Web Server 的不同，您也可能必須指定 Servlet URL 的環境根式部分 (請參閱第 294 頁的「[範例 4：在 Sun Java System Application Server 7.0 上部署 HTTPS 通道 Servlet](#)」)。

但是，imqhttps.war 檔案的部署描述元無法知道您已將通道 Servlet 所需的密鑰儲存檔案放在何處 (請參閱第 284 頁的「[步驟 1. 為 HTTPS 通道 Servlet 產生自身簽名的憑證](#)」)。這要求您編輯通道 Servlet 的部署描述元 (XML 檔案)，以在部署 imqhttps.war 檔案之前指定密鑰儲存的位置。

步驟 3. 配置 httpsjms 連接服務

預設情況下，代理程式不會啓動 HTTPS 支援，所以您必須重新配置代理程式，以啓動 httpsjms 連接服務。一旦重新配置，此代理程式便可啓動，如第 122 頁的「[啓動代理程式](#)」中所述。

► 啓動 httpsjms 連接服務

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*) 作爲辨別依據的目錄中，且代理程式實例與配置檔案相互關聯 (請參閱 [附錄 A 「Message Queue 資料的位置」](#))：

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

2. 將 httpsjms 值新增至 imq.service.activelist 特性：

```
imq.service.activelist=jms,admin,httpsjms
```

啟動時，代理程式會查找在其主機上執行的 Web Server 和 HTTPS 通道 Servlet。但是，若要存取遠端通道 Servlet，您可以重新配置 `servletHost` 和 `servletPort` 連接服務特性。

您還可以重新配置 `pullPeriod` 特性，以提昇效能。表 C-3 中詳細介紹了 `httpsjms` 連接服務配置特性。

表 C-3 `httpsjms` 連接服務特性

特性名稱	說明
<code>imq.httpsjms.https.servletHost</code>	如有必要，請變更此值，以指定 HTTPS 通道 Servlet 正在其上執行的主機 (主機名稱或 IP 位址) 的名稱。(這可以為遠端主機，或本地主機上的特定主機名稱。)預設值： <code>localhost</code>
<code>imq.httpsjms.https.servletPort</code>	變更此值，以指定代理程式用於存取 HTTPS 通道 Servlet 的連接埠號。(如果 Web Server 上的預設連接埠已變更，則您必須相應地變更此特性。)預設值： <code>7674</code>
<code>imq.httpsjms.https.pullPeriod</code>	指定每個用戶端發出從代理程式中移除訊息的 HTTP 請求的間隔時間 (以秒為單位)。(請注意，此特性於代理程式上進行設定，且傳遞至用戶端執行階段。)如果值為零或負數，則用戶端將始終擱置一個 HTTP 請求，以準備儘快移除訊息。若用戶端數目過大，可能會導致 Web Server 資源大量外流，且伺服器可能沒有確認。在此類情況下，您應將 <code>pullPeriod</code> 特性設定為正的秒數。此作業將設定用戶端的 HTTP 傳輸驅動程式在發出後續移除請求之前等待的時間。將此值設定為正數可保存 Web Server 資源，但會影響用戶端所監視的確認次數。預設值： <code>-1</code>
<code>imq.httpsjms.https.connectionTimeout</code>	指定用戶端執行階段在系統拋出異常前，等待從 HTTPS 通道 Servlet 發出確認的時間 (以秒為單位)。(請注意，此特性於代理程式上進行設定，且傳遞至用戶端執行階段。)此特性也會指定中斷連接前，代理程式與 HTTPS 通道 Servlet 進行通訊後，代理程式等待的時間。因為代理程式與通道 Servlet 無法得知存取 HTTPS Servlet 的用戶端是否已異常中斷，在此情況下，必須設定逾時時間。預設值： <code>60</code>

步驟 4. 配置 HTTPS 連接

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 連接。

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」。

一旦解決這些問題，您可以繼續配置 HTTPS 連接。

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、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，則您需要使用可啓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 (為您的 Web Server 憑證簽名) 的超級使用者憑證依預設不在信任資料庫中，或者您使用的是專用 Web Server 憑證，則您必須將此憑證新增至信任資料庫。如果出現此情況，請遵循以下說明，否則請參考「[配置連線工廠](#)」中的說明。

假設此憑證儲存在 `cert_file` 中，且 `trust_store_file` 為您的密鑰儲存，請執行以下指令：

```
JRE_HOME/bin/keytool -import -trustcacerts
-alias alias_for_certificate -file cert_file
-keystore trust_store_file
```

對以下問題回答 YES：Trust this certificate? (是否信任此憑證?)

您還需要使用可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=trust_store_file
javax.net.ssl.trustStorePassword=trust_store_passwd
```

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:port/contextRoot/tunnel
```


hostName:port 為代管 HTTPS 通道 Servlet 之 Web Server 的名稱和連接埠；
contextRoot 為在 Web Server 上部署通道 Servlet 的路徑設定。

如需連線工廠屬性的一般資訊與 *imqAddressList* 屬性的特定資訊，請參閱「Message Queue Java 用戶端開發人員指南」。

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 176 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的指令的 *-D* 選項 (請參閱「Message Queue Java 用戶端開發人員指南」)。
- 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 Java 用戶端開發人員指南」)。

使用單一 Servlet 存取多重代理程式

如果您要執行多重代理程式，則無需配置多重 Web Server 和 Servlet 實例。您可以在同時執行的代理程式中共用單一 Web Server 和 HTTPS 通道 Servlet 實例。如果多重代理程式實例正共用一個通道 Servlet，您必須配置 *imqAddressList* 連線工廠屬性，配置內容如下：

```
https://hostName:port/contextRoot/tunnel?ServerName=bkrHostName:instanceName
```

bkrHostName 為代理程式實例主機名稱；*instanceName* 為您用戶端存取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 HTTPS 通道 Servlet 的狀態報告。此報告將列出 Servlet 要存取的所有代理程式：

```
HTTPS tunnel servlet ready.
Servlet Start Time : Thu May 30 01:08:18 PDT 2002
Accepting secured connections from brokers on port : 7674
Total available brokers = 2
Broker List :
    jpgserv:broker2
    cochin:broker1
```

使用 HTTP 代理

如果您要使用 HTTP 代理存取 HTTPS 通道 Servlet：

- 將 `http.proxyHost` 系統特性設定為代理伺服器主機名稱。
- 將 `http.proxyPort` 系統特性設定為代理伺服器連接埠號。

您可以使用可啟動用戶端應用程式之指令的 `-D` 選項來設定這些特性。

範例 3：在 Sun Java System Web Server 上部署 HTTPS 通道 Servlet

本節描述如何在 Sun Java System Web Server 上將 HTTPS 通道 Servlet 部署為 JAR 檔案和 WAR 檔案。您使用的方法取決於 Sun Java System Web Server 的版本：如果它不支援 Servlet 2.2 或較高版本，它將無法處理 WAR 檔案部署。

部署為 JAR 檔案

以下說明參考了使用基於瀏覽器的管理 GUI 之 Sun Java System Web Server 6.1 上的部署。此程序包含以下常用步驟：

1. 新增 Servlet
2. 配置 Servlet 虛擬路徑
3. 載入 Servlet
4. 停用 Servlet 存取記錄

這些步驟將在以下各小節中描述。您可以透過使用 Web 瀏覽器存取 Servlet URL 來驗證成功的 HTTPS 通道 Servlet 部署。它應顯示狀態資訊。

新增 Servlet

► 新增通道 Servlet

1. 選取 [Servlets] 標籤。
2. 選擇 [配置 Servlet 屬性]。
3. 在 [Servlet 名稱] 欄位中，為通道 Servlet 指定名稱。
4. 將 [Servlet 程式碼 (類別名稱)] 欄位設定為以下值：

```
com.sun.messaging.jmq.transport.  
httptunnel.servlet.HttpsTunnelServlet
```

5. 在 [Servlet 類別路徑] 欄位中輸入 `imqervlet.jar` 的完整路徑。例如：
 在 Solaris 上為 `/usr/share/lib/imq/imqervlet.jar`
 在 Linux 上為 `/opt/imq/lib/imqervlet.jar`
 在 Windows 上為 `IMQ_HOME/lib/imqervlet.jar`
6. 在 [Servlet 引數] 欄位中，輸入所需引數和選擇性引數，如表 C-4 中所示。

表 C-4 用於部署 HTTPS 通道 Servlet JAR 檔案的 Servlet 引數

引數	預設值	需要嗎？	另請參閱
<code>keystoreLocation</code>	無	需要	第 200 頁的表 8-8
<code>keystorePassword</code>	無	需要	第 200 頁的表 8-8
<code>servletHost</code>	所有主機	不需要	第 287 頁的表 C-3
<code>servletPort</code>	7674	不需要	第 287 頁的表 C-3

使用逗號將引數分隔，例如：

```
keystoreLocation=keystore_location, keystorePassword=keystore_password,
servletPort=portnumber
```

`servletHost` 和 `servletPort` 引數僅適用於 Web Server 和代理程式之間的通訊，並且僅在預設值有問題時才設定。但是，在此情況下，您還必須相應地設定代理程式配置特性（請參閱第 287 頁的表 C-3），例如：

```
imq.httpsjms.https.servletPort
```

配置 Servlet 虛擬路徑 (Servlet URL)

► 若要配置通道 Servlet 的虛擬路徑 (Servlet URL)

1. 選取 [Servlets] 標籤。
2. 選擇 [配置 Servlet 虛擬路徑轉換]。
3. 設定 [虛擬路徑] 欄位。

「虛擬路徑」為通道 Servlet URL 的 `/contextRoot/tunnel` 的一部份：

```
https://hostName:port/contextRoot/tunnel
```

舉例來說，如果您將 `contextRoot` 設定至 `imq`，那麼 [虛擬路徑] 欄位則為：

```
/imq/tunnel
```

4. 將 [Servlet 名稱] 欄位設定為與第 290 頁的「新增 Servlet」的步驟 3 中相同的值。

載入 Servlet

► 在啟動 Web Server 時載入通道 Servlet

1. 選取 [Servlets] 標籤。
2. 選擇 [配置全域屬性]。
3. 在 [啟動 Servlet] 欄位中，輸入與第 290 頁的「新增 Servlet」的步驟 3 中相同的 Servlet 名稱值。

停用伺服器存取記錄

您不必停用伺服器存取記錄，但是如果您停用記錄，將會獲得更好的效能。

► 停用伺服器存取記錄

1. 選取 [狀態] 標籤。
2. 選擇 [記錄個人喜好頁面]。
3. 使用記錄用戶端存取控制來停用記錄

部署為 WAR 檔案

以下說明參考了 Sun Java System Web Server 6.0 Service Pack 2 上的部署。您可以透過使用 Web 瀏覽器存取 Servlet URL 來驗證成功的 HTTPS 通道 Servlet 部署。它應顯示狀態資訊。

在部署 HTTPS 通道 Servlet 之前，請確保 JSSE JAR 檔案包含在 Web Server 的類別路徑中。執行此作業的最簡單方法為將 jsse.jar、jnet.jar 和 jcert.jar 複製到 IWS60_TOPDIR/bin/https/jre/lib/ext。

此外，在部署 HTTPS 通道 Servlet 之前，您必須修改其部署描述元，以指向您放置密鑰儲存檔案的位置，並指定密鑰儲存密碼。

► 修改 HTTPS 通道 Servlet WAR 檔案

1. 將 WAR 檔案複製到暫存目錄。

在 Solaris 上為 `cp /usr/share/lib/imq/imqhttps.war /tmp`

在 Linux 上為 `cp /opt/imq/lib/imqhttps.war /tmp`

在 Windows 上為 `cp IMQ_HOME/lib/imqhttps.war /tmp`

- 將此暫存目錄變為您的目前目錄。

```
$ cd /tmp
```

- 擷取 WAR 檔案的內容。

```
$ jar xvf imqhttps.war
```

- 列出 WAR 檔案的部署描述元。

```
$ ls -l WEB-INF/web.xml
```

- 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。

- 重新組合 WAR 檔案的內容。

```
$ jar uvf imqhttps.war WEB-INF/web.xml
```

此時，您可以使用已修改的 imqhttps.war 檔案來部署 HTTPS 通道 Servlet。(如果您擔心密鑰儲存密碼洩漏，則可以使用檔案系統許可權來限制他人對 imqhttps.war 檔案的存取。)

► 將 HTTPS 通道 Servlet 部署為 WAR 檔案

- 在基於瀏覽器的管理 GUI 中，選取 [虛擬伺服器類別] 標籤。按一下 [管理類別]。
- 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。
- 選取 [管理虛擬伺服器]。
- 選取適當的虛擬伺服器名稱，然後按一下 [管理] 按鈕。
- 選取 [Web 應用程式] 標籤。
- 按一下 [部署 Web 應用程式]。
- 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 292 頁的「[修改 HTTPS 通道 Servlet WAR 檔案](#)」)。
- 在 [應用程式 URI] 欄位中輸入路徑。

[應用程式 URI] 欄位值為通道 Servlet URL 的 /contextRoot 的一部份：

```
https://hostName:port/contextRoot/tunnel
```

舉例來說，如果您將 contextRoot 設定至 imq，那麼 [應用程式 URI] 欄位則為：

```
/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 可從以下位址獲得：

```
https://hostName:port/imq/tunnel
```

此時，用戶端可使用此 URL，以使用安全 HTTPS 連接而連接至訊息服務。

範例 4：在 Sun Java System Application Server 7.0 上部署 HTTPS 通道 Servlet

本節描述如何在 Sun Java System Application Server 7.0 上將 HTTPS 通道 Servlet 部署為 WAR 檔案。

您需要執行兩個步驟：

- 使用 Application Server 7.0 部署工具來部署 HTTPS 通道 Servlet
- 修改應用程式伺服器實例的 `server.policy` 檔案

使用部署工具

► 在 Application Server 7.0 環境中部署 HTTPS 通道 Servlet

1. 在基於瀏覽器的管理 GUI 中選取
應用程式伺服器 > 實例 > server1 > 應用程式 > Web 應用程式。
2. 按一下 [部署] 按鈕。
3. 在 [檔案路徑:] 文字欄位輸入 HTTPS 通道 Servlet WAR 檔案 (`imqhttps.war`) 的位置。

`imqhttps.war` 檔案的位置會因您的作業系統而有差異 (請參閱 [附錄 A](#) 「[Message Queue 資料的位置](#)」)。

4. 按一下 [確定]。

5. 在下一個畫面，於 [環境根式] 文字欄位中設定數值。

[環境根式] 欄位值為通道 Servlet URL 的 */contextRoot* 的一部份：

```
https://hostName:port/contextRoot/tunnel
```

例如，您可以將 [環境根式] 欄位設定為：

```
/img
```

6. 按一下 [確定]。

依預設，接著會出現下一個畫面，表示通道 Servlet 已成功部署在：

```
/var/opt/SUNWappserver7/domains/domain1/server1/applications/  
j2ee-modules/imghttps_1
```

此時，Servlet 可從以下位址獲得：

```
https://hostName:port/contextRoot/tunnel
```

此時，用戶端可使用此 URL，以使用 HTTPS 連接而連接至訊息服務。

修改 server.policy 檔案

Application Server 7.0 會強制一組預設安全策略，除非修改的檔案可以阻止 HTTPS 通道 Servlet 接受來自 Message Queue 代理程式的連接。

每個應用程式伺服器實例，皆有一個包含安全策略或規則的檔案。例如，Solaris 上用於 server1 實例的檔案位置為：

```
/var/opt/SUNWappserver7/domains/domain1/server1/config/  
server.policy
```

若要讓通道 Servlet 接受來自 Message Queue 代理程式的連接，則必須在此檔案新增其他項目。

► 若要修改 Application Server 的伺服器策略檔案

1. 開啓伺服器策略檔案。
2. 新增以下項目：

```
grant codeBase  
"file:/var/opt/SUNWappserver7/domains/domain1/server1/  
applications/j2ee-modules/imghttps_1/-"
```

```
{  
    permission java.net.SocketPermission "*",  
        "connect,accept,resolve";  
};
```


將代理程式作為 Windows 服務

本附錄介紹如何使用服務管理員 (imqsvcadmin) 公用程式來安裝、查詢和移除作為 Windows 服務執行的代理程式。

將代理程式作為 Windows 服務執行

安裝 Message Queue 時，您可以選擇將代理程式作為 Windows 服務安裝。您也可以先在安裝 Message Queue 後，使用 imqsvcadmin 將代理程式作為 Windows 服務安裝。

將代理程式作為 Windows 服務安裝，意味著此代理程式將在系統啟動時啟動，並在後台執行直至關機。因此，您不必使用 imqbrokerd 指令啟動代理程式，除非您要啟動其他實例。若要將所有啟動選項傳送至代理程式，您可以使用 imqsvcadmin 指令 (請參閱第 298 頁的表 D-2) 的 -args 引數，並確切地指定您已為 imqbrokerd 指令 (請參閱第 122 頁的「啟動代理程式」) 使用的相同選項。使用 imqcmd 指令以控制代理程式照常作業。

當代理程式作為 Windows 服務執行時，[工作管理員] 會將代理程式作為兩個可執行程序列出。第一個程序是 imqbrokersvc.exe，這是原生 Windows 服務包裝。第二個程序是實際執行代理程式的 Java 運行時間。

每次只能將一個代理程式作為 Windows 服務安裝並執行。

服務管理員公用程式 (imqsvcadmin)

服務管理員公用程式 (imqsvcadmin) 可讓您安裝、查詢和移除代理程式 (作為 Windows 服務執行)。本節描述 imqsvcadmin 指令的基本語法、提供子指令的清單、總結 imqsvcadmin 指令選項，並介紹如何使用這些指令來完成特定的工作。

imqsvcadmin 指令的語法

imqsvcadmin 指令的一般語法如下：

```
imqsvcadmin subcommand [options]
imqsvcadmin -h
```

請注意，如果您指定 `-v`、`-h`、或 `-H` 選項，則系統不會執行指令行上指定的其他子指令。例如，如果您輸入以下指令，則系統會顯示說明資訊但不會執行 `query` 子指令。

```
imqsvcadmin query -h
```

imqsvcadmin 子指令

Message Queue 服務管理員公用程式 (imqsvcadmin) 包括表 D-1 中列出的子指令：

表 D-1 imqsvcadmin 子指令

子指令	說明
install	安裝服務並指定啟動選項。
query	顯示 imqsvcadmin 指令的啟動選項。這包括服務是手動啟動還是自動啟動、服務的位置、Java 運行時間的位置以及啟動時傳送至代理程式的引數的值。
remove	移除服務。

imqsvcadmin 選項摘要

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

表 D-2 imqsvcadmin 選項

選項	說明
-h	顯示用法說明。不執行指令行上的其他任何指令。
-javahome <i>path</i>	指定要使用的替代 Java 2 相容運行時間的路徑 (預設為使用系統上的運行時間，或 Message Queue 隨附的運行時間)。 範例：imqsvcadmin -install -javahome d:\jdk1.4

表 D-2 imqsvcadmin 選項 (續)

選項	說明
-jrehome <i>path</i>	指定 Java 2 相容的 JRE 的路徑。 範例：imqsvcadmin -install -jrehome d:\jre\1.4
-vmargs <i>arg</i> [[<i>arg</i>]...]	指定要傳送至正在執行代理程式服務的 Java VM 的其他引數。(您還可以在 [Windows 服務控制台] 的 [啟動參數] 欄位中指定這些引數。) 範例：-vmargs "-Xms16m -Xmx128m"
-args <i>arg</i> [[<i>arg</i>]...]	指定要傳送至代理程式服務的其他命令行引數。如需 imqbrokerd 選項的說明，請參閱第 122 頁的「啟動代理程式」。 (您還可以在 [Windows 服務控制台] 的 [啟動參數] 欄位中指定這些引數。) 例如， imqsvcadmin -install -args "-passfile d:\imqpassfile"

您使用 -javahome、-vmargs 和 -args 選項指定的資訊，儲存在 Window 登入檔中的鍵值 JavaHome、JVMargs 和 ServiceArgs 下，路徑為

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

移除代理程式服務

在移除代理程式服務之前，您應使用 imqcmd shutdown bkr 指令關閉代理程式。然後使用 imqsvcadmin remove 指令移除服務，並重新啟動電腦。

重新配置代理程式服務

若要重新配置服務，請先移除此服務，然後將其重新安裝，並指定帶有 -args 引數的不同啟動選項。

使用替代 Java 運行時間

您可以使用 `-javahome` 或 `-jrehome` 選項來指定替代 Java 運行時間的位置。您還可以在 [Windows 服務控制台] 的 [啟動參數] 欄位中指定這些選項。請注意，[啟動參數] 欄位將反斜線 (\) 看作一個退出字元，因此當將反斜線用作路徑分割元時，您必須將其鍵入兩次；例如，`-javahome d:\\jdk1.3`。

查詢代理程式服務

若要確定代理程式服務的啟動選項，請使用 `imqsvcadmin` 指令的 `-q` 選項。

```
imqsvcadmin -query

Service iMQ_Broker is installed.
Display Name: iMQ_Broker
Start Type: Manual
Binary location: c:\Program Files\Sun Microsystems\
                  Message Queue 3.5\bin\imqbrokersvc
JavaHome: c:\j2sdk1.4.0
Broker Args: -passfile d:\imqpassfile
```

疑難排解

如果嘗試啟動服務時出現錯誤，您可以查閱透過執行以下作業記錄的錯誤事件。

► 查閱記錄的服務錯誤事件

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

本附錄包含以下主題的簡短說明：

- 系統時鐘設定
- 定義作業系統檔案描述元限制
- 保護永久性資料

系統時鐘設定

使用 Message Queue 系統時，您應仔細地同步化系統時鐘，並避免將它們設定為早於當前的時間。

建議同步化

建議您同步化與 Message Queue 系統互動的所有主機上的時鐘。如果您使用訊息逾時 (TimeToLive)，則此操作尤為重要。同步化主機時鐘失敗可能會導致 TimeToLive 無法按預期工作 (訊息可能不會發送)。您應在啟動任何代理程式之前同步化時鐘。

Solaris 您可以在本地主機上發出 `rdate` 指令，以與遠端主機同步化。(您必須是超級使用者 - 即 `root` - 才能執行該指令。) 例如，以下指令可將本地主機 (稱其為 `Host 2`) 與遠端主機 `Host1` 同步化：

```
# rdate Host1
```

Linux 指令與 Solaris 類似，但是您必須提供 `-s` 選項：

```
# rdate -s Host1
```

Windows 您可以發出帶有時間子指令的網路指令，以將您的本地主機與遠端主機同步化。例如，以下指令可將本地主機（稱其為 Host 2）與遠端主機 Host1 同步化：

```
net time \\Host1 /set
```

避免將系統時鐘配置為早於當前時間

您應避免將執行 Message Queue 代理程式的系統上之時鐘設定為早於當前時間。Message Queue 使用時間標記來幫助識別內部物件，如異動和長期訂閱。如果將系統時鐘設定為早於當前時間，從理論上講可能會產生重複的內部識別碼。代理程式透過引入識別碼的某些隨機性，以及在執行時偵測時鐘調整來嘗試進行補償，但是如果代理程式未執行時，將系統時鐘調整為遠遠早於當前時間，則仍有識別碼重複的可能。

如果您需要將執行代理程式的系統上之系統時鐘設定為比當前時間早幾秒鐘，建議您在沒有異動或長期訂閱時進行，或者在代理程式未執行時進行，然後等待一段時間（您調整的時鐘時間），再重新啟動代理程式。

但是，理想的方法是在啟動任何代理程式之前同步化時鐘，然後使用相應的技術以確保時鐘不會在部署後有較大調整。

定義作業系統檔案描述元限制

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

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

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

保護永久性資料

代理程式使用永久性倉庫，可包含與其他資訊一起暫時儲存的訊息檔案。由於這些訊息可能包含專用資訊，建議您保護資料倉庫，以防止未經授權的存取。

代理程式可以使用內建永久性倉庫或外掛永久性倉庫。

內建永久性倉庫

使用內建永久性倉庫的代理程式，會將永久性資料寫入目錄（目錄位置因平台而異）中的文字檔案資料倉庫（請參閱附錄 A 「[Message Queue 資料的位置](#)」）：

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

instanceName 是辨別代理程式實例的名稱。

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

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

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

外掛永久性倉庫

使用外掛永久性倉庫的代理程式將永久性資料寫入 JDBC 相容的資料庫中。

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

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

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 (例如 Cloudscape 資料庫)，通常透過在要儲存永久性資料的目錄上設定檔案許可權 (如上面的「內建永久性倉庫，」中所述)，來提供安全性。但是，若要確保代理程式和 imqdbmgr 公用程式均可讀取和寫入資料庫，則兩者應由同一個使用者執行。

Message Queue 資源介面

Message Queue 包括一個 JMS 資源介面。

資源介面為插入附加功能至 J2EE 1.4 相容應用程式伺服器 (與 J2EE Connector Architecture (J2EECA) 1.5 規格相容) 的標準方法。此架構允許任何 J2EE 1.4 相容的應用程式伺服器，以標準方法與外部系統進行互動。這些外部系統包括各種企業資訊系統 (EIS)，以及各種訊息傳送系統，例如 JMS 提供者。

J2EECA 1.5 促成的標準互動包括連接集區、執行緒集區、異動、安全環境傳遞，以及各式訊息驅動 Bean 的支援。此規格亦包括建立連線工廠與其他受管理物件的標準方法。

藉由插入 JMS 資源介面至應用程式伺服器，在應用程式伺服器環境中已部署及執行的 J2EE 元件可交換 JMS 訊息。使用 J2EE 應用程式伺服器管理工具，可以建立並配置這些元件所需的 JMS 連線工廠與目標受管理物件。

但是，像是管理訊息伺服器與實體目標的其他管理作業不包括在 J2EECA 規格中，這些作業只能使用提供者的特定工具來執行。

Message Queue 資源介面內嵌於 Sun J2EE 1.4 應用程式伺服器中。但是，還沒有任何其他 J2EE 1.4 應用程式伺服器認證過 Message Queue 資源介面。

Message Queue 資源介面為單一檔案 (imqjmsra.rar)，位於作業系統的某個目錄中，目錄位置因作業系統而異，請參考附錄 A 「Message Queue 資料的位置」的內容。imqjmsra.rar 檔案包含資源介面部署描述元 (ra.xml)，以及必須由應用程式伺服器使用以便使用介面的 JAR 檔案。

遵循應用程式伺服器隨附的資源介面部署與配置說明，您即可使用任何 J2EE 1.4 相容應用程式伺服器中的 Message Queue 資源介面。當您可取得商業用 J2EE 1.4 應用程式伺服器，且 Message Queue 資源介面已經應用程式伺服器認證，此附錄將提供相關部署與配置程序的特定資訊。

選擇性 JMS 功能的 Message Queue 實施

JMS 規格會指出某些選擇性項目 -- 每個 JMS 提供者 (供應商) 會選擇是否實施它們。Message Queue 產品對每個選擇性項目的處理如下所示：

表 G-1 選擇性 JMS 功能

JMS 規格中的小節	說明和 Message Queue 處理
3.4.3 JMSTimestamp	<p>「由於訊息 ID 建立和增加訊息的容量較為複雜，如果某些 JMS 提供者收到應用程式未使用訊息 ID 的提示，則可能會最佳化訊息耗用時間。JMS 訊息生產者會提供提示，以停用訊息 ID。」</p> <p>Message Queue 實施： 產品不停用訊息 ID 的產生 (MessageProducer 中的所有 setDisableMessageID() 呼叫均被忽略)。所有訊息均將包含有效的訊息 ID 值。</p>
3.4.12 置換訊息標頭欄位	<p>「JMS 未明確定義管理員如何置換這些標頭欄位值。JMS 提供者不需要支援此管理選項。」</p> <p>Message Queue 實施： Message Queue 產品支援透過配置連線工廠受管理物件來管理置換訊息標頭欄位中的值 (請參閱第 168 頁的表 7-3)。</p>
3.5.9 JMS 定義的特性	<p>「JMS 可為 JMS 定義的屬性保留 『JMSX』 特性名稱字首。」</p> <p>「對這些特性的支援是選擇性的，除非另有說明。」</p> <p>Message Queue 實施： Message Queue 產品支援 JMS 1.1 規格定義的 JMSX 特性 (請參閱第 168 頁的表 7-3)。</p>
3.5.10 提供者特定的特性	<p>「JMS 可為提供者特定的特性保留 『JMS_<vendor_name>』 特性名稱字首。」</p> <p>Message Queue 實施： 提供者特定的特性之用途是提供所需的特殊功能，以支援 JMS 與提供者原生用戶端一同使用。它們不應用於 JMS 以進行 JMS 訊息傳送。Message Queue 3.5 SP1 不會使用提供者特定的特性。</p>

表 G-1 選擇性 JMS 功能 (續)

JMS 規格中的小節	說明和 Message Queue 處理
4.4.8 分散式異動	<p>「JMS 不需要提供者支援分散式異動。」</p> <p>Message Queue 實施： 此版本的 Message Queue 產品支援分散式異動 (請參閱第 44 頁的「分散式異動」)。</p>
4.4.9 多重階段作業	<p>「對於 PTP < 點對點分佈模式 >，JMS 沒有為相同的佇列指定並行運作的 QueueReceiver 之語義；但是，JMS 並不禁止提供者支援此功能。」請參閱 JMS 規格的第 5.8 節，已獲得更多資訊。</p> <p>Message Queue 實施： Message Queue 實施支援到多個使用者的佇列發送。如需更多資訊，請參閱第 69 頁的「佇列發送至多個使用者」。</p>

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幫助。表 H-1 根據這些介面的穩定程度 (即它們在以後的產品版本中不進行變更的可能性) 對其進行分類。第 310 頁的表 H-2 中描述了分類機制。

表 H-1 Message Queue 介面的穩定性

介面	分類
imqbrokerd 命令行介面	改進中
imqadmin 命令行介面	不穩定
imqcmd 命令行介面	改進中
imqdbmgr 命令行介面	不穩定
imqkeytool 命令行介面	改進中
imqobjmgr 命令行介面	改進中
imqusermgr 命令行介面	不穩定
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
JMS API (javax.jms)	標準
JAXM API (javax.xml)	標準
C-API	改進中

表 H-1 Message Queue 介面的穩定性 (續)

介面	分類
基於訊息監視的 API	改進中
受管理物件 API (com.sun.messaging)	改進中
imq.jar 位置和名稱	穩定
jms.jar 位置和名稱	改進中
imqbroker.jar 位置和名稱	私有
imqutil.jar 位置和名稱	私有
imqadmin.jar 位置和名稱	私有
imqservlet.jar 位置和名稱	改進中
imqhttp.war 位置和名稱	改進中
imqhttps.war 位置和名稱	改進中
imqjmsra.rar 位置和名稱	改進中
imqxm.jar 位置和名稱	改進中
jaxm-api.jar 位置和名稱	改進中
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
從 imqbrokerd、imqadmin、imqcmd、imqdbmgr、imqkeytool、 imqobjmgr 和 imqusermgr 的輸出	不穩定
代理程式日誌檔的位置和內容格式	不穩定
密碼檔案	不穩定
accesscontrol.properties	不穩定

表 H-2 介面穩定性分類機制

分類	說明
私有	不由客戶直接使用。可在任何版次中變更或移除。

表 H-2 介面穩定性分類機制 (續)

分類	說明
改進中	由客戶使用。用於主要版次 (如 3.0、4.0) 或次要版次 (如 3.1、3.2) 的不相容變更。應謹慎而緩慢地進行變更。應進行適當的操作以確保所有的變更均相容，但是這點無法保證。
穩定	由客戶使用。僅用於主要版次 (如 3.0、4.0) 的不相容變更。
標準	由客戶使用。這些介面由正式標準定義，並由標準組織控制。幾乎沒有與這些介面不相容的變更。
不穩定	由客戶使用。用於主要版次 (如 3.0、4.0) 或次要版次 (如 3.1、3.2) 的不相容變更。通知客戶在未來版次中，這些介面可能會以不相容的方式進行大幅度移除或變更。建議客戶不要在不穩定的介面上建立明確的相依性。

本詞彙表提供有關使用 Sun Java System Message Queue 時，可能遇到的專有名詞和概念的資訊。

受管理物件 預先配置的 Message Queue 物件 (連線工廠或目標)，由管理員建立以供一個或多個 JMS 用戶端使用。

使用受管理物件可讓 JMS 用戶端獨立於供應程式；即，使用戶端脫離供應程式的專用方面。這些物件由管理員放置在 JNDI 名稱空間中，並由使用 JNDI 查找的 JMS 用戶端存取。

非同步通訊 一種通訊模式，在此模式中，訊息傳送者在繼續其他工作之前無需等待傳送方法的傳回。

授權 訊息服務決定使用者是否可存取訊息服務資源 (如連接服務或目標) 的過程。

代理程式 Message Queue 實體，可管理訊息路由、發送、持續性、安全性以及記錄，並提供可讓管理員監視和調整效能與資源使用的介面。

用戶端 與使用訊息服務交換訊息的其他用戶端進行互動式操作的應用程式 (或軟體元件)。

用戶端識別碼 將連接及其物件與作為用戶端的 Message Queue 訊息伺服器所維護的狀態相關聯的識別碼。

用戶端運行時間 請參閱 Message Queue 用戶端運行時間。

叢集 以串聯方式工作可提供訊息傳送服務的兩個或多個互連代理程式。

配置檔案 包含用於配置代理程式的 Message Queue 設定的一個或多個文字檔案。特性為實例特定的或與叢集相關的。

連接 1) 與 Message Queue 訊息伺服器的作用中連接。可以為佇列連接或主題連接。2) 階段作業的 Factory，使用構成 Message Queue 訊息伺服器基礎的連接以生產和使用訊息。

連線工廠 用戶端用於建立與 Message Queue 訊息伺服器連接的受管理物件。可以為 QueueConnectionFactory 物件或 TopicConnectionFactory 物件。

使用 訊息使用者從目標接收訊息。

使用者 階段作業建立的用於從目標接收訊息的物件 (MessageConsumer)。在點對點發送模型中，使用者為接收者或瀏覽者 (QueueReceiver or QueueBrowser)；在出版 / 訂閱發送模型中，使用者為用戶 (TopicSubscriber)。

資料倉庫 永久儲存代理程式所需資訊 (長期訂閱、有關目標的資料、永久性訊息以及稽核資料) 的資料庫。

發送模式 訊息傳送可靠性的指標：保證訊息發送並成功使用一次且僅為一次 (永久性發送模式)，或保證訊息至多發送一次 (非永久性發送模式)。

發送模型 訊息發送的模型：點對點或出版 / 訂閱。在 JMS 中，每種發送模型都有單獨的程式設計網域，並且使用特定用戶端運行時間物件和特定目標類型 (佇列或主題)，以及統一的程式設計網域。

發送策略 當多個訊息使用者註冊時，佇列如何路由訊息的規格。這些策略為：單一、故障轉移和循環。

目標 Message Queue 訊息伺服器中的實體目標，已生產的訊息發送至此目標，以便路由並隨後發送至使用者。此實體目標由受管理物件識別和封裝，用戶端使用此受管理物件指定它要生產訊息和 / 或使用訊息的目標。

網域 JMS 用戶端用於程式化 JMS 訊息傳送作業的一組物件。有兩種程式設計網域：一種用於點對點發送模型，另一種用於出版 / 訂閱發送模型。

JMS (Java 訊息服務) 定義 Java 用戶端如何存取訊息服務設備的標準的一組介面和語義。這些介面為 Java 程式提供標準的方法，以建立、傳送、接收和讀取訊息。

JMS 供應程式 為訊息傳送系統實施 JMS 介面，並新增完整產品所需的管理和控制功能的產品。

Message Queue 用戶端運行時間 為 JMS 用戶端提供 Message Queue 訊息伺服器介面的軟體。用戶端運行時間支援用戶端將訊息傳送至目標和從此類目標接收訊息所需的所有作業。

Message Queue 訊息伺服器 為 Message Queue 訊息傳送系統提供遞送服務的軟體，其中包括與 JMS 用戶端的連線、訊息路由與遞送、持續性、安全性以及記錄。訊息伺服器維護實體目標，JMS 用戶端將訊息傳送至此目標，並且訊息從此目標發送至使用用戶端。

訊息選擇器 使用者選擇基於 JMS 訊息標頭中特性值 (選擇器) 的訊息之方法。訊息服務執行基於訊息選擇器中所存在條件的訊息過濾與路由。

訊息服務 請參閱 Message Queue 訊息伺服器。

訊息 JMS 用戶端使用的非同步請求、報告或事件。訊息包含標頭 (其他欄位可新增至此) 和內文。訊息標頭指定標準欄位和選擇性特性。訊息內文包含要傳送的資料。

訊息傳送 企業應用程式使用的非同步請求、報告或事件的系統，可讓無密切關聯的應用程式可靠且安全地傳送資訊。

點對點發送模型 產生者將訊息傳送至特定佇列；使用者從為保留其訊息而建立的佇列擷取訊息。一個訊息僅發送至一個訊息使用者。

生產 將訊息傳送至用戶端運行時間，以便發送至目標。

產生者 由階段作業建立的物件 (MessageProducer)，用於將訊息傳送至目標。在點對點發送模型中，產生者為傳送者 (QueueSender)；在出版 / 訂閱發送模型中，產生者為出版者 (TopicPublisher)。

出版 / 訂閱發送模型 出版者和用戶通常為匿名，並可動態出版或訂閱主題。系統將來自主題的多個出版者的訊息分布至其多個用戶。

佇列 由管理員建立的可實施點對點發送模型的物件。佇列始終用於保留訊息，即使使用其訊息的用戶端處於非作用中。佇列被用作產生者與使用者之間的媒介保留位置。

階段作業 用於傳送和接收訊息的單一執行緒環境。可以為佇列階段作業或主題階段作業。

主題 由管理員建立的可實施出版 / 訂閱發送模型的物件。主題可作為內容階層 (負責收集和分布傳送給它的訊息) 中的節點來檢視。透過將主題用作媒介，訊息出版者可與訊息用戶保持獨立。

異動 必須完成或完全回轉的原子單位的工作。

使用者群組 Message Queue 用戶端使用者所屬的群組，旨在授權對 Message Queue 訊息伺服器資源（如連接和目標）的存取權。

A

admin 連接服務 50, 148
API 說明文件 29, 260, 261, 262

C

Cloudscape 263

H

HTTP

支援架構 273
代理 273
連接服務, 請參閱 [httpjms 連接服務](#)
傳輸驅動程式 273

HTTP 通道 Servlet

部署 275
關於 273

HTTP 連接

支援 273
多重代理程式, 用於 278
通道 Servlet, 請參閱 [HTTPS 通道 Servlet](#)
請求間隔時間 277

httpjms 連接服務

配置 276
設定 274

關於 50, 148

HTTPS

支援架構 273

HTTPS 通道 Servlet

部署 285
關於 273

HTTPS 連接

支援 273
多重代理程式, 用於 289
通道 Servlet, 請參閱 [HTTPS 通道 Servlet](#)
請求間隔時間 287

httpsjms 連接服務

配置 286
設定 283
關於 50, 148

I

imq.accesscontrol.enabled 特性 63, 118

imq.accesscontrol.file.filename 特性 63, 118

imq.authentication.basic.user_repository 特性 62, 118

imq.authentication.client.response.timeout 特性 63, 118

imq.authentication.type 特性 62, 118

imq.autocreate.destination.isLocalOnly 特性 71, 118

imq.autocreate.destination.limitBehavior 特性 71, 118

- imq.autocreate.destination.maxBytesPerMsg 特性 71, 118
- imq.autocreate.destination.maxCount 特性 71, 118
- imq.autocreate.destination.maxNumMsgs 特性 71
- imq.autocreate.destination.maxNumProducers 特性 71, 118
- imq.autocreate.destination.maxTotalMsgBytes 特性 71, 118
- imq.autocreate.queue 特性 71, 118, 144
- imq.autocreate.queue.consumerFlowLimit 特性 72, 118, 119
- imq.autocreate.queue.localDeliveryPreferred 特性 72, 119
- imq.autocreate.queue.maxNumActiveConsumers 特性 71, 119, 145
- imq.autocreate.queue.maxNumBackupConsumers 特性 71, 119, 145
- imq.autocreate.topic 特性 71, 119, 145
- imq.cluster.brokerlist 特性 127
- imq.cluster.masterbroker 特性 127
- imq.cluster.port 特性 128
- imq.cluster.transport 特性 128
- imq.cluster.url 特性 127, 145
- imq.hostname 特性 52, 119
- imq.httpjms.http.connectionTimeout 特性 277
- imq.httpjms.http.pullPeriod 特性 277
- imq.httpjms.http.servletHost 特性 276
- imq.httpjms.http.servletPort 特性 277
- imq.httpsjms.https.connectionTimeout 特性 287
- imq.httpsjms.https.pullPeriod 特性 287
- imq.httpsjms.https.servletHost 特性 287
- imq.httpsjms.https.servletPort 特性 287
- imq.keystore.file.dirpath 特性 200
- imq.keystore.file.name 特性 200
- imq.keystore.password 特性 200, 204
- imq.log.console.output 特性 67, 119
- imq.log.console.stream 特性 67, 119
- imq.log.file.dirpath 特性 67, 119
- imq.log.file.filename 特性 67
- imq.log.file.name 特性 119
- imq.log.file.output 特性 67, 119
- imq.log.file.rolloverbytes 特性 67, 119, 145
- imq.log.file.rolloversecs 特性 67, 119, 145
- imq.log.level 特性 67, 119, 145
- imq.log.syslog.facility 特性 67, 119
- imq.log.syslog.identity 特性 68, 119
- imq.log.syslog.logconsole 特性 67, 119
- imq.log.syslog.logpid 特性 67, 119
- imq.log.syslog.output 特性 68, 120
- imq.log.timezone 特性 68, 120
- imq.message.expiration.interval 特性 56, 120
- imq.message.max_size 特性 57, 120, 145
- imq.metrics.enabled 特性 66, 120
- imq.metrics.interval 特性 66, 120
- imq.metrics.topic.enabled 特性 68, 120
- imq.metrics.topic.interval 特性 68, 120
- imq.metrics.topic.persist 特性 68, 120
- imq.metrics.topic.timetolive 特性 68, 120
- imq.passfile.dirpath 特性 63, 120
- imq.passfile.enabled 特性 63, 120
- imq.passfile.name 特性 63, 120
- imq.persist.file.destination.message.filepool.limit 特性 59, 120
- imq.persist.file.message.cleanup 特性 60, 120
- imq.persist.file.message.filepool.cleanratio 特性 60, 120
- imq.persist.file.message.max_record_size 特性 59, 120
- imq.persist.file.message.vrfile.max_record_size 特性 58
- imq.persist.file.sync.enabled 特性 59, 120
- imq.persist.jdbc.brokerid 特性 266
- imq.persist.jdbc.closedburl 特性 266
- imq.persist.jdbc.createdburl 特性 266
- imq.persist.jdbc.driver 特性 266
- imq.persist.jdbc.needpassword 特性 266
- imq.persist.jdbc.opendburl 特性 266

- imq.persist.jdbc.password 特性 204, 267
- imq.persist.jdbc.table.IMQCCREC35 特性 267
- imq.persist.jdbc.table.IMQDEST35 特性 267
- imq.persist.jdbc.table.IMQINT35 特性 267
- imq.persist.jdbc.table.IMQLIST35 特性 268
- imq.persist.jdbc.table.IMQMSG35 特性 267
- imq.persist.jdbc.table.IMQPROPS35 特性 267
- imq.persist.jdbc.table.IMQSV35 特性 267
- imq.persist.jdbc.table.IMQTACK35 特性 268
- imq.persist.jdbc.table.IMQTXN35 特性 268
- imq.persist.jdbc.user 特性 266
- imq.persist.store 特性 59, 120, 266
- imq.ping.interval 特性 52, 121
- imq.portmapper.backlog 特性 52, 121
- imq.portmapper.hostname 特性 52, 121
- imq.portmapper.port 特性 52, 121, 145
- imq.protocol protocol_type inbufsz 253
- imq.protocol protocol_type nodelay 252
- imq.protocol protocol_type outbufsz 253
- imq.resource_state.count 特性 57, 121
- imq.resource_state.threshold 特性 57, 121
- imq.service.activelist 特性 52, 121
- imq.service_name.accesscontrol.enabled 特性 63, 121
- imq.service_name.accesscontrol.file.filename 特性 63, 121
- imq.service_name.authentication.type 特性 62, 121
- imq.service_name.max_threads 特性 53, 121
- imq.service_name.min_threads 特性 53, 121
- imq.service_name.protocol_type.hostname 特性 53, 121, 128
- imq.service_name.protocol_type.port 特性 53, 121
- imq.service_name.threadpool_model 特性 53, 122
- imq.shared.connectionMonitor_limit 特性 53, 122
- imq.system.max_count 特性 56, 122, 145
- imq.system.max_size 特性 56, 122, 145
- imq.transaction.autorollback 特性 57, 122, 164
- imq.user_repository.ldap.base 特性 191
- imq.user_repository.ldap.gidattr 特性 191
- imq.user_repository.ldap.grpbase 特性 191
- imq.user_repository.ldap.grpfilter 特性 192
- imq.user_repository.ldap.grpsearch 特性 191
- imq.user_repository.ldap.memattr 特性 191
- imq.user_repository.ldap.password 特性 191, 204
- imq.user_repository.ldap.principal 特性 191
- imq.user_repository.ldap.server 特性 191
- imq.user_repository.ldap.ssl.enabled 特性 192
- imq.user_repository.ldap.timeout 特性 192
- imq.user_repository.ldap.uidattr 特性 191
- imq.user_repository.ldap.usrfilter 特性 191
- imqAckOnAcknowledge 屬性 168
- imqAckOnProduce 屬性 168
- imqAckTimeout 屬性 168
- imqAddressList 屬性 168
- imqAddressListBehavior 屬性 168
- imqAddressListIterations 屬性 168
- imqbrokerd 指令
 - 使用 122
 - 指令語法 122
 - 選項 124
 - 關於 88
- imqBrokerHostName 屬性 (Message Queue 3.0) 168
- imqBrokerHostPort 屬性 (Message Queue 3.0) 169
- imqBrokerServicePort 屬性 (Message Queue 3.0) 169
- imqcmd 指令
 - 子指令 138
 - 用於 137
 - 目標管理 152
 - 度量監視 220
 - 指令語法 138
 - 異動管理 162
 - 連接至代理程式 141
 - 與代理程式的安全連接 141, 202
 - 選項 139
 - 關於 88
- imqConfiguredClientID 屬性 169
- imqConnectionFlowCount 屬性 169
- imqConnectionFlowLimit 屬性 169
- imqConnectionFlowLimitEnabled 屬性 169

imqConnectionType 屬性 (Message Queue 3.0) 169

imqConnectionURL 屬性 (Message Queue 3.0) 169

imqConsumerFlowLimit 屬性 169

imqConsumerFlowThreshold 屬性 169

imqdbmgr 指令

子指令 269

指令語法 269

選項 271

關於 89

imqDefaultPassword 屬性 169

imqDefaultUsername 屬性 169

imqDestinationDescription 屬性 81, 170

imqDestinationName 屬性 81, 170

imqDisableSetClientID 屬性 169

imqFlowControlLimit 屬性 169

imqJMSDeliveryMode 屬性 169

imqJMSExpiration 屬性 169

imqJMSPriority 屬性 169

imqkeytool 指令

使用 199, 284

指令語法 199, 284

關於 89

imqLoadMaxToServerSession 屬性 169

imqobjmgr 指令

子指令 171

用於 170

指令語法 171

選項 171

關於 88

imqOverrideJMSDeliveryMode 屬性 169

imqOverrideJMSExpiration 屬性 169

imqOverrideJMSHeadersToTemporaryDestinations
屬性 169

imqOverrideJMSPriority 屬性 169

imqQueueBrowserMax MessagesPerRetrieve 屬
性 169

imqQueueBrowserRetrieveTimeout 屬性 169

imqReconnectAttempts 屬性 169

imqReconnectEnabled 屬性 169

imqReconnectInterval 屬性 169

imqSetJMSXAppID 屬性 170

imqSetJMSXConsumerTXID 屬性 170

imqSetJMSXProducerTXID 屬性 170

imqSetJMSXRcvTimestamp 屬性 170

imqSetJMSXUserID 屬性 170

imqSSLIsHostTrusted 屬性 (Message Queue 3.0) 170

imqsvcadm 指令

子指令 298

用於 297

指令語法 298

選項 298

關於 89

imqusermgr 指令

子指令 186

用於 185

使用者名稱 188

指令語法 185

密碼 188

選項 186

關於 89

J

J2EE 應用程式

EJB 規格 39

JMS, 和 39

訊息驅動 Bean, [請參閱訊息驅動 Bean](#)

Java 虛擬機器, [請參閱JVM](#)

JDBC 支援

設定 263

關於 59

驅動程式 263, 266

JDK

指定路徑 125

要指定路徑的選項 140, 172, 298

JMS

訊息結構 37

規格 29, 31, 37

程式設計模型 37

jms 連接服務 50, 148

JNDI

Message Queue 支援 32

位置 (提供者 URL) 166, 167

受管理物件, 和 39, 41

物件倉庫 88, 165

物件倉庫屬性 166, 173

初始環境 166, 167

查找 79, 81, 104, 173

查找名稱 173, 176

訊息驅動 Bean, 和 41

JVM

度量, 請參閱JVM 度量

效能影響 216

效能調整 252

JVM 度量

使用 imqcmd metrics 222

使用代理程式日誌檔 226

使用基於訊息的監視 226

度量數目 230

L

LDAP 伺服器

使用者儲存庫存取 190

物件倉庫屬性 166

認證錯誤修復 190

M

MDB, 請參閱訊息驅動 Bean

O

Oracle 263

P

PointBase 263

S

Servlet, 通道, 請參閱HTTP/HTTP 通道 Servlet

SOAP 32

SSL

加密, 與 198

連接服務, 請參閱基於 SSL 的連接服務

透過 HTTP 203

透過 TCP/IP 198

關於 62

ssladmin 連接服務

設定 198

關於 50, 148

ssljms 連接服務

設定 198

關於 50, 148

syslog (系統日誌) 65, 135

T

TCP 50, 148

TLS 50, 148

W

Windows 服務, 代理程式執行爲 297

X

XA 連接工廠 45

另請參閱連接工廠受管理物件

XA 資源管理程式, [請參閱分散式異動](#)

三 畫

工具, 管理, [請參閱管理工具](#)

ㄇ 畫

元件

EJB [39](#)

MDB [40](#)

內建持續性 [58](#)

分散式異動

XA 資源管理員 [44, 162](#)

關於 [44](#)

日誌檔

捲動條件 [67](#)

預設位置 [65, 259, 260, 261](#)

ㄩ 畫

主代理程式 [75, 76](#)

主題

如實體目標 [70](#)

自動建立的 [71, 118](#)

新增受管理物件 [177](#)

關於 [42](#)

屬性 [154](#)

主題目標, [請參閱主題](#)

代理程式

HTTP 支援 [275](#)

httpjms 連接服務特性 [276](#)

HTTPS, 支援 [284](#)

httpsjms 連接服務特性 [287](#)

JDBC 支援, [請參閱JDBC 支援](#)

Windows 服務, 執行為 [297](#)

互連, [請參閱代理程式叢集](#)

元件和功能 [48](#)

主代理程式 [75](#)

列出連接服務 [148](#)

多重代理程式叢集, [請參閱代理程式叢集](#)

存取控制, [請參閱授權](#)

安全性管理程式, [請參閱安全性管理程式](#)

自動建立目標特性 [71](#)

更新特性 [144](#)

度量, [請參閱代理程式度量](#)

持續性管理程式, [請參閱持續性管理程式](#)

查詢 [144](#)

重新啓動 [57, 143, 146](#)

限制行爲 [56, 219](#)

特性 [144](#)

記憶體管理 [55, 154, 219](#)

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 [56, 122, 145](#)

訊息路由, [請參閱訊息路由器](#)

配置檔案, [請參閱配置檔案](#)

從故障回復 [57](#)

控制狀態 [145](#)

啓動 [122](#)

啓動基於 SSL 的服務 [201](#)

連接在一起 [129](#)

連接至 [141](#)

連接服務, [請參閱連接服務](#)

實例名稱 [125](#)

實例配置特性 [118](#)

監視, [請參閱代理程式監視服務](#)

管理 [142](#)

暫停 [143, 145](#)

確認 (Ack) [54, 168](#)

叢集, [請參閱代理程式叢集](#)

關於 [48](#)

關閉 [146](#)

繼續 [143, 145](#)

顯示特性 [144](#)

代理程式度量

使用 imqcmd [146, 223, 224](#)

使用代理程式日誌檔 [226](#)

使用基於訊息的監視 [226](#)

- 度量訊息 66
- 度量數目 230
- 記錄程式特性 66, 225
- 報告間隔時間, 記錄程式 125
- 代理程式實例, 請參閱代理程式
- 代理程式監視服務
 - 特性 66
 - 關於 63
- 代理程式叢集
 - 主代理程式 75, 76
 - 同步化時鐘 127
 - 在僅開發環境中 76
 - 安全代理程式互通連接 130
 - 使用的原因 72, 218
 - 架構 73, 218
 - 要指定的選項 124
 - 重新啟動代理程式 131
 - 效能影響 218
 - 配置特性 76, 127
 - 配置變更記錄 75
 - 設定特性 128
 - 連接代理程式 129
 - 新增代理程式 130
 - 資訊傳遞 75
 - 叢集配置檔案 76, 127
- 出版 / 訂閱發送 42
- 加密
 - 基於 SSL 的服務, 與 198
 - 鍵值工具, 與 62
 - 關於 62
- 可靠的發送 43
 - 效能平衡 45, 209
- 可攜型, 請參閱供應程式獨立性
- 外掛持續性
 - 效能調整 255
 - 設定 263
 - 關於 59
- 平台版 33
- 永久性訊息 43
- 用戶端
 - 程式設計模型 37
 - 運行時間, 請參閱用戶端運行時間
 - 應用程式, 請參閱用戶端應用程式
 - 識別碼 (ClientID) 43
 - 用戶端運行時間
 - 訊息流量調整 257
 - 配置 219
 - 關於 77
 - 用戶端應用程式
 - 系統特性, 與 81
 - 供應程式獨立性 41
 - 影響效能的因素 209
 - 範例 29, 260, 261, 262
 - 目標
 - 介紹的 48
 - 主題, 請參閱主題
 - 用於發送的批次訊息 72, 155
 - 列出 153, 156
 - 存取控制 196
 - 自動建立的 70, 197
 - 更新屬性 153
 - 取得資訊 153
 - 度量, 請參閱目標度量
 - 建立 153
 - 限制行爲 55, 154
 - 清除訊息 153, 159
 - 資訊 156
 - 實體 68
 - 管理 152
 - 暫存 72, 156
 - 暫停 153, 158
 - 銷毀 152, 153, 159
 - 壓縮基於檔案的資料倉庫 152
 - 叢集中限制的範圍 71, 155
 - 類型 68, 153
 - 繼續 153, 158
 - 屬性 154
 - 屬性值 156
 - 顯示屬性值 156
 - 目標受管理物件
 - 介紹的 38
 - 關於 80
 - 屬性 170
 - 目標度量

使用 imqcmd metrics 153, 221, 223

使用 imqcmd query 224

使用基於訊息的監視 226

度量數目 234

目錄變數

IMQ_HOME 26

IMQ_JAVAHOME 26

IMQ_VARHOME 26

目錄變數 IMQ_HOME 26

目錄變數 IMQ_JAVAHOME 26

目錄變數 IMQ_VARHOME 26

大書

企業版 33

存取控制檔

用於 193

存取規則 194

位置 260, 261, 262

版本 193

格式 193

存取規則 194

安全性

加密, 請參閱加密

物件倉庫, 用於 166

授權, 請參閱授權

管理程式, 請參閱安全性管理程式

認證, 請參閱認證

安全性管理程式

作為代理程式元件 49

特性 62

關於 60

安全套接層標準, 請參閱SSL

自身簽名的憑證 199, 284

自動建立目標

特性 71

關於 70

六書

佇列 69

自動建立的 71, 118

負載平衡的發送, 請參閱負載平衡的佇列發送

新增受管理物件 177

屬性 154

佇列負載平衡發送

屬性 155

作業系統

效能影響 216

調整 Solaris 效能 251

更新

代理程式 144

連接服務 148, 149, 151

系統特性, 設定 81

防火牆 51, 273

八書

使用者 38

使用者名稱

格式 188

預設 185

屬性 169

使用者群組

刪除指定 187

預先定義的 187

預設 61

關於 61

使用者儲存庫

LDAP 伺服器 190

文字檔案 184

平台相依性 185

位置 260, 261

使用者狀態 187

使用者群組 187

特性 62

移入 188

管理 188

關於 60

- 供應程式獨立性
 - 受管理物件 79
 - 關於 41
 - 協定, *請參閱*傳輸協定
 - 協定類型
 - HTTP 50, 148
 - TCP 50, 148
 - TLS 50, 148
 - 受管理物件
 - XA 連接工廠, *請參閱*連接工廠受管理物件
 - 主題, *請參閱*主題
 - 目標, *請參閱*目標受管理物件
 - 列出 179
 - 刪除 178
 - 更新 180
 - 供應程式獨立性 79
 - 物件倉庫, *請參閱*物件倉庫
 - 查找名稱 172
 - 查詢 180
 - 連接工廠, *請參閱*連接工廠受管理物件
 - 需要的資訊 173
 - 關於 39, 79
 - 類型 39, 79, 168
 - 屬性 168
 - 服務類型
 - ADMIN 50
 - NORMAL 50
 - 版本, 產品
 - 平台 33
 - 企業 33
 - 關於 33
 - 物件倉庫
 - LDAP 伺服器 166
 - LDAP 伺服器屬性 166
 - 位置 259, 260, 261
 - 檔案系統倉庫 167
 - 檔案系統倉庫屬性 167
 - 關於 165
 - 長期訂閱
 - ClientID, 和 43
 - ID 140
 - 列出 161
 - 效能影響 212
 - 清除訊息 161
 - 管理 161
 - 銷毀 161, 162
 - 關於 42
 - 長期訂閱, *請參閱*長期訂閱
- ## 1 畫
- 度量
 - 主題目標 65, 226
 - 訊息, *請參閱*度量訊息
 - 資料, *請參閱*度量資料
 - 監視工具, *請參閱*度量監視工具
 - 關於 64
 - 度量訊息
 - 內容 66
 - 關於 65, 226
 - 類型 65, 226
 - 度量資料
 - 代理程式, *請參閱*代理程式度量
 - 目標, *請參閱*目標度量
 - 使用 imqcmd metrics 222
 - 使用代理程式日誌檔 225
 - 使用基於訊息監視的 API 226
 - 連接服務, *請參閱*連接服務度量
 - 說明清單 229
 - 度量監視工具
 - 比較的 228
 - 訊息佇列公用程式 (imqcmd) 220
 - 訊息佇列日誌檔 225
 - 基於訊息監視的 API 226
 - 持續性
 - 內建 58
 - 外掛, *請參閱*外掛持續性
 - 持續性管理程式, *請參閱*持續性管理程式
 - 發送模式, *請參閱*發送模式
 - 資料倉庫 *請參閱*資料倉庫
 - 持續性管理程式
 - JDBC 資料倉庫 265
 - 外掛持續性, 與 263
 - 作為代理程式元件 49

- 特性 59
 - 資料倉庫, [請參閱資料倉庫](#)
 - 關於 57
- 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 89
 - 選項共用於 90
 - 關於 88
- 指令行語法 89
- 指令選項 90
- 指令檔 174
- 故障轉移 73
- 查詢
 - 代理程式 144
 - 連接服務 148, 149, 151
- 流量控制, [請參閱訊息流量控制](#)
- 訂閱
 - 長期, [請參閱長期訂閱](#)
 - 關於 42
- 負載平衡的佇列發送
 - 效能調整 256
 - 關於 69
 - 屬性 71
- 重新啓動代理程式 143, 146
- 重新發送旗標 55
- 限制行爲
 - 代理程式 56
 - 目標 55, 154

十 畫

- 容器
 - EJB 40
 - MDB 40

- 效能
 - 可靠性平衡 45, 209
 - 指標 206
 - 效能評定 206
 - 基本式樣 207
 - 瓶頸 209
 - 最佳化, [請參閱效能調整](#)
 - 測量 206
 - 疑難排解 236
 - 監視, [請參閱效能監視](#)
 - 影響因素, [請參閱效能影響因素](#)
 - 調整, [請參閱效能調整](#)
 - 關於 205
- 效能評定, 效能 206
- 效能監視
 - 工具, [請參閱度量監視工具](#) 220
 - 度量資料, [請參閱度量資料](#)
- 效能影響因素
 - JVM 216
 - 代理程式限制行爲 219
 - 作業系統 216
 - 長期訂閱 212
 - 訊息內文類型 214
 - 訊息伺服器架構 218
 - 訊息流量控制 219
 - 訊息容量 214
 - 異動 211
 - 連接 216
 - 發送模式 210
 - 硬體 215
 - 傳輸協定 216
 - 資料倉庫 219
 - 確認模式 212
 - 選擇器 213
- 效能調整
 - 代理程式調整 255
 - 用戶端運行時間 257
 - 系統調整 251
 - 程序概述 205
- 特性
 - httpjms 連接服務 276
 - httpsjms 連接服務 287

- 代理程式,更新 144
- 代理程式實例配置 118
- 代理程式監視服務 66
- 目標,請參閱目標,屬性
- 安全性 62
- 自動建立 71
- 受管理物件,請參閱受管理物件,屬性
- 持續性 59
- 記憶體管理 56, 154
- 記錄程式 66
- 訊息路由器 56
- 密鑰儲存 200
- 連接服務 52
- 與 JDBC 相關 266
- 叢集配置 127
- 記憶體管理
 - 用於代理程式 55
 - 使用目標屬性 154
 - 效能調整 255
- 記錄,請參閱記錄程式
- 記錄程式
 - 作為代理程式元件 50
 - 度量資訊 66
 - 重新導向日誌訊息 135
 - 級別 64, 67, 125
 - 訊息格式 133
 - 捲動條件 135
 - 種類 64
 - 寫入至主控台 67, 126
 - 輸出通道 64, 135, 225
 - 關於 64
 - 變更配置 134
- 訊息
 - SOAP 32
 - 介紹的 37
 - 內文類型,與效能 214
 - 代理程式限制 56, 122, 145
 - 出版 / 訂閱發送 42
 - 可靠發送 44
 - 永久性 43
 - 生產 77
 - 目標限制 154
 - 回收已過期的 56
 - 使用 78
 - 延時 206
 - 度量 65
 - 度量訊息,請參閱度量訊息
 - 持續性 55, 57
 - 流量效能 206
 - 流量控制,請參閱訊息流量控制
 - 負載平衡的佇列發送 69
 - 重新發送 55
 - 容量,與效能 214
 - 偵聽程式 38, 78
 - 從目標清除 153
 - 控制 54
 - 排序 46
 - 發送模式,請參閱發送模式
 - 發送模型 36, 42
 - 結構 37
 - 路由與發送 53
 - 過濾,請參閱選擇器
 - 確認 54
 - 優先順序排序 46
 - 點對點發送 42
- 訊息伺服器
 - 多重代理程式,請參閱代理程式叢集 72
 - 架構 218
 - 關於 48
- 訊息服務
 - 影響效能的因素 215
 - 關於 36
- 訊息流量控制
 - 代理程式 55, 154
 - 計數 257
 - 限制 257
 - 效能影響 219
 - 效能調整 257
- 訊息產生者,請參閱產生者
- 訊息發送模型 36, 42
- 訊息傳送系統
 - Message Queue 架構 48
 - 架構 36
 - 訊息服務 36

訊息路由器

作為代理程式元件 49

特性 56

關於 53

訊息驅動 Bean

MDB 容器 40

部署描述元 41

應用程式伺服器支援 41

關於 40

配置檔案

安裝 115

位置 259, 260, 261

預設 115

實例 116, 128, 144, 259, 260, 261

範本 259, 260, 261

範本位置 259, 260, 261

編輯 117

配置變更記錄 75

十一 畫

偵聽程式 38, 40

基於 SSL 的連接服務

啟動 201

設定 183, 199

執行緒儲存區管理程式

共用執行緒 52

專屬執行緒 52

關於 51

密碼

JDBC 204

LDAP 204

SSL 密鑰儲存 125, 200, 204

命名慣例 188

密碼檔案, 請參閱密碼檔案

編碼 62

密碼, 預設值 169

密碼檔案

代理程式配置特性 63

位置 204, 260, 261, 262

使用 204

指令行選項 125

密碼檔案, 請參閱密碼檔案

密鑰儲存

file 199

特性 200

檔案 200, 284

控制訊息 54

授權

另請參閱存取控制檔

用於 Message Queue 版本 33

使用者群組 61

使用試用企業版授權啟動 124

啟動選項 125

管理 192

關於 61

啟動

代理程式 122

基於 SSL 的連接服務 201

叢集中的代理程式 131

清除, 從目標的訊息 159

瓶頸, 效能 209

產生者

目標限制 71, 155

關於 38

異動

分散式, 請參閱分散式異動

回轉 162

效能影響 211

資訊 162

管理 162

確定 163

確認, 與 55

關於 44

許可權

Message Queue 作業 61

內嵌式資料庫 265

存取控制特性檔案 61, 193

使用者儲存庫 185

密碼檔案 204

密鑰儲存 284

運算 194

- 管理服務 61
- 通道 Servlet, *請參閱* HTTP/HTTP 通道 Servlet
- 連接
 - 介紹的 38
 - 列出 151
 - 指令影響 151
 - 查詢 151
 - 效能影響 216
- 連接工廠受管理物件
 - ClientID, 和 43
 - JNDI 查找 39
 - 介紹的 38
 - 新增 176
 - 置換 81
 - 關於 80
 - 屬性 80, 168
- 連接, 至代理程式 141
- 連接服務
 - admin 50, 148
 - HTTP, *請參閱* HTTP 連接
 - httpjms 50, 148
 - HTTPS, *請參閱* HTTP 連接
 - httpsjms 50, 148
 - jms 50, 148
 - ssladmin, *請參閱* ssladmin 連接服務
 - ssljms, *請參閱* ssljms 連接服務
 - 在啟動時啟動 52
 - 存取控制 63
 - 更新 148, 149, 151
 - 服務類型 50
 - 度量資料, *請參閱* 連接服務度量
 - 指令影響 147
 - 查詢 148, 151
 - 特性 52, 149
 - 基於 SSL 的 200
 - 執行緒分配 149
 - 執行緒儲存區管理程式 51
 - 連接埠對映器, *請參閱* 連接埠對映器
 - 連接類型 50
 - 暫停 147, 150
 - 靜態連接埠 52
 - 叢集 130, 198
 - 關於 49

- 繼續 148, 150
- 顯示特性 149
- 連接服務度量
 - 使用 imqcmd metrics 150, 223
 - 使用 imqcmd query 224
 - 度量數目 232
- 連接埠, 動態配置 51
- 連接埠對映器
 - 連接埠指定 52, 126
 - 關於 51

十二畫

- 發送, 可靠的 43
- 發送模式
 - 永久性 43
 - 非永久性 43
 - 效能影響 210
- 硬體, 效能影響 215
- 程式設計網域 42
- 階段作業
 - 介紹的 38
 - 異動的 44
 - 確認選項 44

十三畫

- 傳輸協定
 - 協定類型, *請參閱* 協定類型
 - 相關速度 217
 - 效能影響 216
 - 效能調整 252
- 資料, Message Queue, 位置 259
- 資料倉庫
 - JDBC 可存取的 59
 - 文字檔案 58
 - 位置 259, 260, 261
 - 重設 126

十二節

效能影響 219

關於 57

資源介面 41

路由, *請參閱*訊息路由器

十三節

實例配置檔案, *請參閱*配置檔案

疑難排解 236

監視, *請參閱*效能監視

管理

代理程式 142

目標 152

管理工作

生產環境 84

開發環境 83

管理工具

指令行公用程式 88

管理主控台 87

關於 87

網域 42

認證

管理 184

關於 60

十四節

暫存目標 72, 156

暫停

代理程式 143, 145

目標 153, 158

連接服務 147, 150

確認

代理程式 54

用戶端 54, 78

異動, 與 55

發送, 屬於 54

等待期間 168

關於 44, 54

範例應用程式 29, 260, 261, 262

十五節

憑證 199, 284

選擇器

作為訊息特性 37

效能影響 213

關於 46

十六節

應用程式, *請參閱*用戶端應用程式

應用程式伺服器, 和 Message Queue 41

檔案描述元限制 302

環境變數, *請參閱*目錄變數

鍵值工具 62

鍵對

重新產生 200

產生 199

點對點發送 42

十七節

叢集, *請參閱*代理程式叢集

叢集配置檔案 76

叢集連接服務

設定, 安全 130, 198

連接埠號用於 128

十八節

關閉代理程式 143, 146

二十

繼續

代理程式 143, 145

目標 158

連接服務 148, 150

二十一

屬性

目標 154

受管理物件 168

