



Sun Java System Message Queue 3.7 UR1 管理指南



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

文件號碼：820-0844
2007年1月

Copyright 2006 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. 版權所有。

Sun Microsystems, Inc. 對本文件所述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)，這些智慧財產權可能包含一項或多項美國專利，或美國及其他國家/地區的申請中專利。

美國政府權利 - 商業軟體。政府使用者均應遵守 Sun Microsystems, Inc. 的標準授權合約和 FAR 及其增補文件中的適用條款。

本發行版本可能包括協力廠商開發的材料。

本產品中的某些部分可能源自加州大學授權的 Berkeley BSD 系統的開發成果。UNIX 是在美國及其他國家/地區的註冊商標，已獲得 X/Open Company, Ltd. 專屬授權。

Sun、Sun Microsystems、Sun 標誌、Solaris 標誌、Java 咖啡杯標誌、docs.sun.com、Java 與 Solaris 是 Sun Microsystems, Inc. 在美國及其他國家/地區的商標或註冊商標。所有 SPARC 商標都是 SPARC International, Inc. 在美國及其他國家/地區的商標或註冊商標，經授權後使用。凡具有 SPARC 商標的產品都是採用 Sun Microsystems, Inc. 所開發的架構。

OPEN LOOK 與 SunTM Graphical User Interface (Sun 圖形化使用者介面) 都是由 Sun Microsystems, Inc. 為其使用者與授權者所開發的技術。Sun 感謝 Xerox 公司在研究和開發視覺化或圖形化使用者介面之概念上，為電腦工業所做的開拓性貢獻。Sun 已向 Xerox 公司取得 Xerox 圖形化使用者介面之非獨占性授權，該授權亦適用於使用 OPEN LOOK GUI 並遵守 Sun 書面授權合約的 Sun 公司授權者。

本出版品所涵蓋的產品和所包含的資訊受到美國出口控制法規的控制，並可能受到其他國家/地區進出口法規的管轄。嚴禁將本產品直接或間接地用於核武器、飛彈、生化武器或海上核動力裝備，也不得將本產品直接或間接地提供給核武器、飛彈、生化武器或海上核動力裝備的一般使用者。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。

本文件以其「原狀」提供，對任何明示或暗示的條件、陳述或擔保，包括對適銷性、特殊用途的適用性或非侵權性的暗示保證，均不承擔任何責任，除非此免責聲明的適用範圍在法律上無效。

目錄

前言	21
第 1 部分 Message Queue 管理簡介	31
1 管理作業和工具	33
管理作業	33
開發環境中的管理	33
生產環境中的管理	34
管理工具	35
指令行公用程式	35
管理主控台	36
2 快速入門指導	37
啓動管理主控台	38
管理主控台線上說明	39
使用代理程式	40
啓動代理程式	40
將代理程式新增至管理主控台	41
▼ 若要將代理程式新增至管理主控台	41
連線至代理程式	42
▼ 連線至代理程式	43
檢視連線服務	43
▼ 檢視可用的連線服務	43
使用實體目標	45
建立實體目標	45
▼ 將實體目標新增至代理程式	45
檢視實體目標特性	47

▼ 檢視或修改實體目標的特性	47
清除實體目標的訊息	49
▼ 清除實體目標的訊息	49
刪除實體目標	50
▼ 刪除實體目標	50
使用物件存放區	50
新增物件存放區	50
▼ 將物件存放區新增至管理主控台	51
連線至物件存放區	52
▼ 連線至物件存放區	53
使用受管理物件	53
新增連線工廠	53
▼ 將連線工廠新增至物件存放區	53
新增目標	55
▼ 將目標新增至物件存放區	55
檢視受管理物件特性	57
▼ 檢視或修改受管理物件的特性	57
刪除受管理物件	57
▼ 刪除受管理物件	58
執行範例應用程式	58
▼ 執行範例應用程式	58
第 2 部分 管理作業	61
3 啟動代理程式和用戶端	63
準備系統資源	63
同步化系統時鐘	63
設定檔案描述元限制	64
啟動代理程式	64
以互動方式啟動代理程式	64
自動啟動代理程式	65
移除代理程式	68
移除 Solaris 或 Linux 上的代理程式	68
移除 Windows 上的代理程式服務	68
啟動用戶端	69

4	配置代理程式	71
	代理程式服務	71
	連線服務	71
	路由服務	74
	永久性服務	75
	安全性服務	77
	監視服務	80
	設定代理程式特性	82
	配置檔案	82
	配置永久性資料存放區	84
	配置檔案式存放區	85
	配置 JDBC 型存放區	85
	▼ 配置 JDBC 型資料存放區	85
	保護永久性資料	87
5	管理代理程式	89
	必要條件	89
	使用 imqcmd 公用程式	90
	顯示說明	90
	顯示產品版本	90
	指定使用者名稱和密碼	90
	指定代理程式名稱和連接埠	91
	範例	91
	顯示代理程式資訊	92
	更新代理程式特性	93
	暫停和重新繼續代理程式	94
	暫停代理程式	94
	重新繼續代理程式	94
	關閉和重新啓動代理程式	95
	顯示代理程式度量	95
	管理連線服務	96
	列出連線服務	97
	顯示連線服務資訊	97
	更新連線服務特性	98
	顯示連線服務度量	99

暫停和重新繼續連線服務	99
取得有關連線的資訊	100
管理長期訂閱	101
管理作業事件	103
6 管理實體目標	105
使用指令公用程式	106
子指令	106
建立實體目標	106
列出實體目標	108
顯示實體目標資訊	108
更新實體目標特性	110
暫停和重新繼續實體目標	110
清除實體目標	111
銷毀實體目標	112
壓縮實體目標	112
監視實體目標的磁碟使用	112
使用停用的訊息佇列	114
配置使用停用的訊息佇列	114
管理停用的訊息佇列	114
啟用停用的訊息之記錄功能	115
7 管理安全性	117
使用者認證	117
使用平面檔案使用者儲存庫	118
將 LDAP 伺服器用於使用者儲存庫	122
使用者授權：存取控制特性檔案	124
建立存取控制特性檔案	125
存取規則的語法	125
如何計算權限	126
連線服務的存取控制	127
實體目標的存取控制	128
自動建立實體目標的存取控制	129
訊息加密	130
使用自我簽署的憑證	130

▼ 使用自我簽署的憑證設定 SSL 型連線服務	131
使用簽署的憑證	135
▼ 使用簽署的憑證	135
密碼檔案	138
安全考量	138
密碼檔案內容	138
透過防火牆連線	139
▼ 透過防火牆啓用代理程式連線	139
稽核記錄	140
8 管理受管理物件	141
物件存放區	141
LDAP 伺服器物件存放區	141
檔案系統物件存放區	143
受管理物件屬性	143
連線工廠屬性	144
目標屬性	149
使用物件管理員公用程式	149
新增受管理物件	150
刪除受管理物件	152
列出受管理物件	152
檢視受管理物件資訊	153
修改受管理物件屬性	153
使用指令檔	154
9 使用代理程式叢集	157
叢集配置特性	157
設定個別代理程式的叢集特性	158
使用叢集配置檔案	158
管理叢集	159
連線代理程式	159
將代理程式新增至叢集	160
▼ 使用叢集配置檔案將新的代理程式新增至叢集	160
從叢集中移除代理程式	160
主代理程式	161

管理配置變更記錄	162
▼ 復原配置變更記錄	162
當主代理程式無法使用時	162
10 監視代理程式	163
監視工具簡介	163
配置及使用代理程式記錄功能	164
預設記錄配置	165
記錄訊息格式	166
變更記錄程式配置	166
▼ 變更代理程式的記錄程式配置	166
以互動方式顯示度量	169
imqcmd metrics	170
使用顯示度量資料的 metrics 子指令	171
▼ 使用 metrics 子指令	171
度量輸出：imqcmd 度量	171
imqcmd query	173
撰寫應用程式來監視代理程式	174
設定訊息型監視	174
▼ 若要設定訊息型監視	174
安全性與存取注意事項	175
度量輸出：度量訊息	176
11 分析與調校訊息服務	177
關於效能	177
效能調校程序	177
效能類型	178
效能評定	178
基準使用式樣	179
影響效能的因素	180
▼ 訊息傳送步驟	180
影響效能的應用程式設計因素	181
影響效能的訊息服務因素	185
調校配置以改善效能	188
系統調校	188

代理程式調校	191
用戶端執行階段訊息流量調校	192
12 排解疑難問題	195
用戶端無法建立連線	195
連線流量過慢	199
用戶端無法建立訊息產生器	200
訊息產生延遲或過慢	201
儲存訊息	203
代理程式的流量不穩定	206
訊息未送達用戶	207
停用的訊息佇列包含訊息	210
第 3 部分 參照	215
13 指令行參照	217
指令行語法	217
代理程式公用程式	218
指令公用程式	222
代理程式管理	223
連線服務管理	224
連線管理	225
實體目標管理	226
長期訂閱管理	227
作業事件管理	228
一般指令公用程式選項	228
物件管理員公用程式	229
資料庫管理者公用程式	231
使用者管理員公用程式	232
服務管理員公用程式	233
金鑰工具公用程式	234
14 代理程式特性參照	235
連線特性	235

路由特性	237
永久性特性	241
檔案式永久性	241
JDBC 型永久性	242
安全性特性	245
檢視特性	248
叢集配置特性	251
按字母順序排列的代理程式特性清單	253
15 實體目標特性參照	259
實體目標特性	259
16 受管理物件屬性參照	263
連線工廠屬性	263
連線處理	263
用戶端標識	266
可靠性與流量控制	266
佇列瀏覽器和伺服器階段作業	268
標準訊息特性	268
訊息標頭置換	269
目標屬性	269
SOAP 端點屬性	270
17 JMS 資源介面特性參照	271
ResourceAdapter JavaBean	271
ManagedConnectionFactory JavaBean	273
ActivationSpec JavaBean	274
18 度量參照	277
JVM 度量	277
代理程式通用度量	278
連線服務度量	279
目標度量	281

第 4 部分 附錄	285
A Message Queue 資料的特定平台位置	287
Solaris	287
Linux	288
Windows	289
B Message Queue 介面的穩定性	291
分類機制	291
介面穩定性	292
C HTTP/HTTPS 支援	295
HTTP/HTTPS 支援架構	295
啓用 HTTP 支援	297
▼ 啓用 HTTP 支援	297
步驟 1. 部署 HTTP 通道 Servlet	297
步驟 2. 配置 httpjms 連線服務	300
▼ 啓動 httpjms 連線服務	300
步驟 3. 配置 HTTP 連線	301
啓用 HTTPS 支援	303
▼ 啓用 HTTPS 支援	303
步驟 1. 爲 HTTPS 通道 Servlet 產生自我簽署的憑證	304
步驟 2. 修改 HTTP 通道 Servlet .war 檔案的描述元檔案	304
▼ 修改 HTTPS 通道 Servlet .war 檔案	305
步驟 3. 部署 HTTPS 通道 Servlet	305
步驟 4. 配置 httpsjms 連線服務	309
▼ 啓動 httpsjms 連線服務	309
步驟 5. 配置 HTTPS 連線	310
疑難排解	313
伺服器或代理程式發生故障	313
用戶端無法透過通道 Servlet 進行連線	313
▼ 如果用戶端無法連線	313

D 常用指令公用程式指令	315
語法	315
代理程式和叢集管理	315
代理程式配置特性 (-o 選項)	316
服務和連線管理	316
長期訂閱者管理	317
作業事件管理	317
目標管理	317
目標配置特性 (-o 選項)	317
度量	318
索引	319

圖清單

圖 1-1	本機與遠端管理公用程式	36
圖 2-1	管理主控台視窗	39
圖 2-2	管理主控台說明視窗	40
圖 2-3	[增加代理程式] 對話方塊	41
圖 2-4	管理主控台視窗中顯示的代理程式	42
圖 2-5	[連線至代理程式] 對話方塊	43
圖 2-6	檢視連線服務	44
圖 2-7	[服務特性] 對話方塊	44
圖 2-8	[增加代理程式目標] 對話方塊	46
圖 2-9	[代理程式目標屬性] 對話方塊	48
圖 2-10	[長期訂閱] 面板	49
圖 2-11	[增加物件存放區] 對話方塊	51
圖 2-12	管理主控台視窗中顯示的物件存放區	52
圖 2-13	[增加連線工廠物件] 對話方塊	54
圖 2-14	[增加目標物件] 對話方塊	56
圖 2-15	管理主控台視窗中顯示的目標物件	57
圖 4-1	永久性資料存放區	75
圖 4-2	安全性支援	78
圖 4-3	監視支援	80
圖 4-4	代理程式配置檔案	83
圖 11-1	透過 Message Queue 服務傳送訊息	180
圖 11-2	傳輸協定速度	186
圖 C-1	HTTP/HTTPS 支援架構	296

表清單

表 4-1	Message Queue 連線服務	72
表 4-2	度量主題目標	81
表 5-1	Message Queue 連線服務	96
表 5-2	imqcmd 更新的連線服務特性	98
表 6-1	指令公用程式的實體目標子指令	106
表 6-2	實體目標磁碟使用度量	113
表 6-3	停用的訊息佇列處理標準實體目標特性的方式	115
表 7-1	使用者儲存庫中的初始項目	118
表 7-2	imqusermgr 選項	119
表 7-3	存取規則的語法元素	125
表 7-4	實體目標存取控制規則的元素	129
表 7-5	自我簽署憑證所需之識別名稱資訊	131
表 7-6	使用密碼的指令	138
表 7-7	密碼檔案中的密碼	139
表 7-8	靜態連接埠位址的代理程式配置特性	139
表 8-1	LDAP 物件存放區屬性	142
表 8-2	檔案系統物件存放區屬性	143
表 10-1	度量監視工具的優點與限制	164
表 10-2	記錄層級	165
表 10-3	imqcmd metrics 子指令語法	170
表 10-4	imqcmd metrics 子指令選項	170
表 10-5	imqcmd query 子指令語法	173
表 10-6	度量主題目標	174
表 11-1	高可靠性和高效能方案的比較	182
表 13-1	代理程式公用程式選項	218
表 13-2	用於代理程式管理的指令公用程式子指令	224
表 13-3	連線服務管理的指令公用程式子指令	225
表 13-4	連線服務管理的指令公用程式子指令	225

表 13-5	用於實體目標管理的指令公用程式子指令	226
表 13-6	用於長期訂閱管理的指令公用程式子指令	227
表 13-7	用於作業事件管理的指令公用程式子指令	228
表 13-8	一般指令公用程式選項	228
表 13-9	物件管理員子指令	229
表 13-10	物件管理員選項	230
表 13-11	資料庫管理者子指令	231
表 13-12	資料庫管理者選項	231
表 13-13	使用者管理員子指令	232
表 13-14	一般使用者管理員選項	233
表 13-15	服務管理員子指令	233
表 13-16	服務管理員選項	234
表 14-1	代理程式連線特性	235
表 14-2	代理程式路由特性	237
表 14-3	自動建立的目標之代理程式特性	238
表 14-4	全域代理程式永久性特性	241
表 14-5	檔案式永久性的代理程式特性	241
表 14-6	JDBC 型永久性的代理程式特性	243
表 14-7	代理程式安全性特性	245
表 14-8	代理程式監視特性	248
表 14-9	叢集配置的代理程式特性	252
表 14-10	按字母順序排列的代理程式特性清單	253
表 15-1	實體目標特性	259
表 16-1	連線處理的連線工廠屬性	264
表 16-2	訊息代理程式定址機制	265
表 16-3	訊息代理程式位址範例	266
表 16-4	用戶端標識的連線工廠屬性	266
表 16-5	可靠性與流量控制的連線工廠屬性	267
表 16-6	佇列瀏覽器和伺服器階段作業的連線工廠屬性	268
表 16-7	標準訊息特性的連線工廠屬性	268
表 16-8	訊息標頭置換的連線工廠屬性	269
表 16-9	目標屬性	269
表 16-10	SOAP 端點屬性	270
表 17-1	資源介面特性	272
表 17-2	管理的連線工廠特性	273
表 17-3	啟動規格特性	274

表 18-1	JVM 度量	277
表 18-2	代理程式通用度量	278
表 18-3	連線服務度量	279
表 18-4	目標度量	281
表 A-1	Solaris 平台上的 Message Queue 資料位置	287
表 A-2	Linux 平台上的 Message Queue 資料位置	288
表 A-3	Windows 平台上的 Message Queue 資料位置	289
表 B-1	介面穩定性分類機制	291
表 B-2	Message Queue 介面的穩定性	292
表 D-1	代理程式配置特性 (-o 選項)	316
表 D-2	目標配置特性 (-o 選項)	317

範例清單

範例 3-1	顯示代理程式服務啟動選項	67
範例 8-1	新增連線工廠	151
範例 8-2	將目標新增至 LDAP 物件存放區	151
範例 8-3	將目標新增至檔案系統物件存放區	151
範例 8-4	刪除受管理物件	152
範例 8-5	列出所有受管理物件	152
範例 8-6	列出特定類型的受管理物件	152
範例 8-7	檢視受管理物件資訊	153
範例 8-8	修改受管理物件的屬性	153
範例 8-9	物件管理員指令檔語法	154
範例 8-10	指令檔範例	154
範例 8-11	部分指令檔	155
範例 8-12	使用部分指令檔	155

前言

本「Sun Java System™ Message Queue 管理指南」為系統管理員提供設定和管理 Sun Java System Message Queue™ 訊息傳送系統時所需的背景和資訊。

本書適用對象

本手冊適用於需要執行 Message Queue 管理作業的管理員和應用程式開發者。Message Queue 管理員負責設定和管理 Message Queue 訊息傳送系統，尤其是處於系統核心地位的訊息代理程式。

閱讀本書之前

閱讀本手冊之前，應先閱讀「Sun Java System Message Queue 3.7 UR1 技術摘要」，以熟悉 Java 訊息服務規格之 Message Queue 實作、Message Queue 服務的元件，以及開發、部署和管理 Message Queue 應用程式的基本程序。

本書架構

表 P-1 描述本手冊的內容。

表 P-1 本手冊的內容

部份/章節	說明
第 1 部分「Message Queue 管理簡介」	
第 1 章「管理作業和工具」	介紹 Message Queue 管理作業和工具。
第 2 章「快速入門指導」	提供實用指導，讓您熟悉 Message Queue 管理主控台。
第 2 部分「管理作業」	
第 3 章「啓動代理程式和用戶端」	描述如何啓動 Message Queue 代理程式和用戶端。

表 P-1 本手冊的內容 (續)

部份/章節	說明
第 4 章「配置代理程式」	描述如何設定及讀取配置特性，並介紹代理程式的可配置內容。此外，還描述如何設定檔案或資料庫以執行永久性功能。
第 5 章「管理代理程式」	描述代理程式管理作業。
第 6 章「管理實體目標」	描述與實體目標相關的管理作業。
第 7 章「管理安全性」	描述與安全性相關的作業，例如管理密碼檔案、認證、授權和加密。
第 8 章「管理受管理物件」	描述物件存放區，並說明如何執行與受管理物件(連線工廠和目標)相關的作業。
第 9 章「使用代理程式叢集」	描述如何設定及管理 Message Queue 代理程式叢集。
第 10 章「監視代理程式」	描述如何設定及使用 Message Queue 監視設備。
第 11 章「分析與調校訊息服務」	描述分析與最佳化訊息服務效能的技術。
第 12 章「排解疑難問題」	提供判斷一般 Message Queue 問題成因和解決方法的建議。
第 3 部分「參照」	
第 13 章「指令行參照」	提供 Message Queue 指令行公用程式的語法和說明。
第 14 章「代理程式特性參照」	描述 Message Queue 訊息代理程式的配置特性。
第 15 章「實體目標特性參照」	描述實體目標的配置特性。
第 16 章「受管理物件屬性參照」	描述受管理物件(連線工廠和目標)的配置特性。
第 17 章「JMS 資源介面特性參照」	描述可搭配應用程式伺服器使用之 Message Queue 資源介面的配置特性。
第 18 章「度量參照」	描述 Message Queue 訊息代理程式可提供用於監視、調校和診斷用途的度量資訊。
第 4 部分「附錄」	
附錄 A「Message Queue 資料的特定平台位置」	列出每個支援平台上的 Message Queue 檔案位置。
附錄 B「Message Queue 介面的穩定性」	描述不同 Message Queue 介面的穩定性。
附錄 C「HTTP/HTTPS 支援」	描述如何設定及使用超文字傳輸協定(HTTP)進行 Message Queue 通訊。
附錄 D「常用指令公用程式指令」	列出某些常用的 Message Queue 指令公用程式(imqcmd)指令。

相關文件

Message Queue 文件集所包含的文件依照一般使用順序列於下表。

表 P-2 Message Queue 文件集

文件	使用者	說明
「Sun Java System Message Queue 3.7 UR1 Installation Guide」	開發者與管理員	解釋如何在 Solaris、Linux 及 Windows 平台上安裝 Message Queue 軟體。
「Sun Java System Message Queue 3.7 UR1 版本說明」	開發者與管理員	包括對新功能、限定、已知錯誤以及技術性注意事項的描述。
「Sun Java System Message Queue 3.7 UR1 技術摘要」	開發者與管理員	介紹 Message Queue 概念、功能及元件。
「Sun Java System Message Queue 3.7 UR1 Developer's Guide for Java Clients」	開發者	為使用 Message Queue 實作 JMS 或 SOAP/JAXM API 的 Java 用戶端程式開發者，提供快速入門教學和程式設計資訊。
「Sun Java System Message Queue 3.7 UR1 管理指南」	管理員，也適用於開發者	提供使用 Message Queue 管理工具執行管理作業時所需的背景與資訊。
「Sun Java System Message Queue 3.7 UR1 Developer's Guide for C Clients」	開發者	針對使用 JMS API (C-API) 之 Message Queue C 實作的 C 用戶端程式開發者，提供程式設計和參照文件。

線上說明

Message Queue 3.7 UR1 包含執行 Message Queue 訊息服務管理作業的指令行公用程式。

Message Queue 3.7 UR1 還包括圖形化使用者介面 (GUI) 管理工具，即管理主控台 (imqadmin)。管理主控台中包含上下文相關的說明；請參閱第 39 頁的「[管理主控台線上說明](#)」。

JavaDoc

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

平台	位置
Solaris	/usr/share/javadoc/imq/index.html

平台	位置
Linux	/opt/sun/mq/javadoc/index.html
Windows	IMQ_HOME/javadoc/index.html

此文件可在任何 HTML 瀏覽器中檢視。它包括標準的 JMS API 文件，以及用於 Message Queue 受管理物件的 Message Queue 專用 API，這對訊息傳送應用程式的開發者很有幫助。

用戶端應用程式範例

Message Queue 提供多個用戶端應用程式範例以協助開發者使用。

Java 用戶端應用程式範例

Java 用戶端應用程式範例位於各個平台的下列目錄：請參閱這些目錄與其中各個子目錄中的 README 檔案。

平台	位置
Solaris	/usr/demo/imq/
Linux	/opt/sun/mq/examples
Windows	IMQ_HOME/demo/

C 用戶端程式範例

C 用戶端應用程式範例位於各個平台的下列目錄：請參閱位於這些目錄中的 README 檔案。

平台	位置
Solaris	/opt/SUNWimq/demo/C/
Linux	/opt/sun/mq/examples/C/
Windows	IMQ_HOME/demo/C/

Java Message Service (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。

目錄變數慣例

Message Queue 使用 3 個目錄變數，這些變數的設定方式因平台而異。表 P-3 說明這些變數，以及在 Solaris、Linux 和 Windows 平台上使用這些變數的方式。

備註 – 表 P-3 中的資訊僅適用於獨立安裝的 Message Queue。Message Queue 在安裝完成，並且附屬於應用程式伺服器安裝一起執行後，目錄變數的值會有不同的設定：IMQ_HOME 會設定為 *appServer_install_dir/imq* (其中 *appServer_install_dir* 是應用程式伺服器安裝目錄)，而 IMQ_VARHOME 會設定為 *appServer_domainName_dir/imq* (其中 *appServer_domainName_dir* 是啟動 Message Queue 代理程式之網域的網域目錄)。

表 P-3 目錄變數慣例

變數	說明
IMQ_HOME	<p>用於 Message Queue 文件，表示 Message Queue 基底目錄 (根安裝目錄)：</p> <ul style="list-style-type: none"> ■ 在 Solaris 和 Linux 上，沒有根 Message Queue 安裝目錄。因此，Message Queue 文件未使用 IMQ_HOME 表示 Solaris 和 Linux 的檔案位置。 ■ 在 Windows 上，根 Message Queue 安裝目錄是設定在 Message Queue 束解壓縮的目錄。
IMQ_VARHOME	<p>儲存 Message Queue 暫存檔或動態建立配置檔與資料檔的 /var 目錄。可以將其設定為指向任何目錄的環境變數。</p> <ul style="list-style-type: none"> ■ 在 Solaris 上，IMQ_VARHOME 預設為 /var/imq 目錄。 ■ 在 Solaris 上，針對 Sun Java System Application Server, Evaluation Edition，IMQ_VARHOME 預設為 IMQ_HOME/var 目錄。 ■ 在 Linux 上，IMQ_VARHOME 預設為 /var/opt/sun/mq 目錄。 ■ 在 Windows 上，IMQ_VARHOME 預設為 IMQ_HOME/var 目錄。

表 P-3 目錄變數慣例 (續)

變數	說明
IMQ_JAVAHOME	<p>指向 Message Queue 可執行檔所需 Java 執行階段環境 (JRE) 位置的環境變數：</p> <ul style="list-style-type: none"> ■ 在 Solaris 上，IMQ_JAVAHOME 依下列順序尋找 Java 執行階段，但是使用者可以選擇將該值設定為所需 JRE 駐留的任意位置。Solaris 8 或 9。 <ul style="list-style-type: none"> /usr/jdk/entsys-j2se /usr/jdk/latest /usr/jdk/jdk1.5.* /usr/jdk/j2sdk1.5.* /usr/j2se Solaris 10： <ul style="list-style-type: none"> /usr/jdk/entsys-j2se /usr/jdk/java /usr/jdk/latest /usr/j2se ■ 在 Linux 上，Message Queue 先依下列順序尋找 Java 執行階段，但是使用者可以選擇將 IMQ_JAVAHOME 值設定為所需 JRE 駐留的任意位置。 <ul style="list-style-type: none"> /usr/jdk/entsys-j2se /usr/java/jre1.5.* /usr/java/jdk1.5.* /usr/java/jre1.4.2* /usr/java/j2sdk1.4.2* ■ 在 Windows 上，IMQ_JAVAHOME 會設定為指向現有 Java 執行階段 (如果在系統上找到支援的版本)。如果找不到支援的版本，則會安裝一個支援的執行階段。

在本書中，IMQ_HOME、IMQ_VARHOME 和 IMQ_JAVAHOME 的顯示不附帶特定平台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 (/)。

印刷排版慣例

下表說明本書所使用的印刷排版慣例。

表 P-4 印刷排版慣例

字體	意義	範例
AaBbCc123	指令、檔案及目錄的名稱，以及電腦螢幕畫面輸出	編輯 .login 檔案。 使用 <code>ls -a</code> 列出所有檔案。 machine_name% you have mail.
AaBbCc123	鍵入的內容，與電腦螢幕畫面輸出內容相對應	machine_name% su Password:
<i>aabbcc123</i>	預留位置：使用實際名稱或值取代	用以移除檔案的指令為 <code>rm filename</code> 。
<i>AaBbCc123</i>	書籍標題、新術語與要強調的字詞	請參閱User's Guide 中的第 6 章。 快照 是指本機儲存的副本。 請勿儲存此檔案。 備註 ：某些強調的項目線上顯示為粗體。

指令範例中的 Shell 提示符號

下表列出針對 C shell、Bourne shell、Korn shell 以及 Windows 作業系統，預設的 UNIX® 系統提示符號及超級使用者提示符號。

表 P-5 Shell 提示符號

Shell	提示
C shell	machine_name%
C shell 超級使用者	machine_name#
Bourne shell 和 Korn shell	\$
Bourne shell 和 Korn shell 超級使用者	#
Windows	C:\

符號慣例

下表說明本書可能使用的符號。

表 P-6 符號慣例

符號	說明	範例	意義
[]	包含選用引數與指令選項。	ls [-l]	-l 選項不是必要選項。
{ }	包含必要指令選項的一組選擇。	-d {y n}	-d 選項要求使用 y 引數或 n 引數。
\${ }	表示變數參照。	\${com.sun.javaRoot}	參照 com.sun.javaRoot 變數的值。
-	同時按下多個按鍵。	Ctrl-A	同時按下 Ctrl 鍵和 A 鍵。
+	連續按下多個按鍵。	Ctrl+A+N	按下 Ctrl 鍵再放開，然後按下後續的按鍵。
→	表示圖形化使用者介面的功能表項目選取。	[檔案] → [新增] → [範本]	從 [檔案] 功能表中，選擇 [新增]。從 [新增] 子功能表選擇 [範本]。

文件、支援和培訓

Sun 網站提供下列其他資源的相關資訊：

- 文件 (<http://www.sun.com/documentation/>)
- 支援 (<http://www.sun.com/support/>)
- 培訓 (<http://www.sun.com/training/>)

搜尋 Sun 產品文件

除了從 docs.sun.com 網站搜尋 Sun 產品文件之外，您可以使用搜尋引擎，在搜尋欄位中輸入下列語法：

```
search-term site:docs.sun.com
```

例如，若要搜尋「broker」，請如下輸入：

```
broker site:docs.sun.com
```

若要在搜尋中包含其他 Sun 網站 (例如 java.sun.com、www.sun.com 和 developers.sun.com)，請在搜尋欄位中使用「sun.com」，而非「docs.sun.com」。

協力廠商網站參照

本文件中提供了協力廠商 URL 以供參考，另亦提供其他相關的資訊。

備註 – Sun 對本文件中提到的協力廠商網站的可用性不承擔任何責任。對於此類網站或資源中的 (或透過它們所取得的) 任何內容、廣告、產品或其他材料，Sun 並不表示認可，也不承擔任何責任。對於因使用或依靠此類網站或資源中的 (或透過它們所取得的) 任何內容、產品或服務而造成的、名義上造成的或連帶產生的任何實際或名義上之損壞或損失，Sun 概不負責，也不承擔任何責任。

Sun 歡迎您提出寶貴意見

Sun 致力於提高文件品質，因此誠心歡迎您提出意見與建議。若要分享您的意見，請至 <http://docs.sun.com>，並按一下 [Send Comments (傳送您的意見)]。在線上表單中，請提供完整的文件標題和文件號碼。文件號碼是一個七位或九位的數字，可以在書的標題頁面或文件的 URL 中找到。例如，本書的文件號碼為 820-0844。

在您提出意見時，可能需要在表單中輸入英文版書名和文件號碼，本書的英文版文件號碼和書名為：819-4467 和「Sun Java SystemMessageQueue 3.7 UR1 Administration Guide」。

第 1 部分

Message Queue 管理簡介

- 第 1 章
- 第 2 章

管理作業和工具

本章提供 Sun Java™ System Message Queue™ 管理作業及執行這些作業之工具的簡介，說明重點為指令行管理公用程式的一般功能。包含下列各節：

- 第 33 頁的「管理作業」
- 第 35 頁的「管理工具」

管理作業

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

開發環境中的管理

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

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

生產環境中的管理

在必須以可靠方式部署和執行應用程式的生產環境中，管理更為重要。執行的管理作業依訊息傳送系統及其必須支援之應用程式的複雜性而定。這種作業可分為兩大類：設定作業與維護作業。

設定作業

生產環境的管理設定作業，一般包含下列部分或全部的項目：

管理員安全性

- 設定預設管理使用者 (admin) 的密碼 (第 122 頁的「變更預設管理員密碼」)
- 控制個人或群組存取管理連線服務 (第 127 頁的「連線服務的存取控制」) 與停用的訊息佇列 (第 128 頁的「實體目標的存取控制」)
- 規範管理群組對檔案式或簡易目錄存取協定 (LDAP) 使用者儲存庫的存取 (第 120 頁的「群組」、第 124 頁的「設定管理員的存取控制」)

一般安全性

- 管理檔案式使用者儲存庫的內容 (第 121 頁的「寫入和管理使用者儲存庫」) 或配置讓代理程式使用現有的 LDAP 使用者儲存庫 (第 122 頁的「編輯實例配置檔案」)
- 控制對個別使用者或群組可獲得授權執行的作業 (第 124 頁的「使用者授權：存取控制特性檔案」)
- 使用安全通訊端層 (SSL) 設定加密服務 (第 130 頁的「訊息加密」)

受管理物件

- 設定及配置 LDAP 物件存放區 (第 141 頁的「LDAP 伺服器物件存放區」)
- 建立連線工廠和目標 (第 150 頁的「新增受管理物件」)

代理程式叢集

- 建立叢集配置檔案 (第 158 頁的「使用叢集配置檔案」)
- 指定主代理程式 (第 161 頁的「主代理程式」)

永久性

- 配置讓代理程式使用永久性存放區 (第 84 頁的「配置永久性資料存放區」)

記憶體管理

- 設定目標的配置特性，將記憶體使用率最佳化 (第 110 頁的「更新實體目標特性」、第 15 章)

維護作業

因為應用程式的效能、可靠性及安全性在生產環境中極為重要，所以必須透過持續不斷的管理維護作業，緊密監視並控制訊息服務資源，這些作業包括：

代理程式管理與調校

- 使用代理程式度量，以調校和重新配置代理程式 (第 11 章)
- 管理代理程式記憶體資源 (第 74 頁的「路由服務」)
- 建立與管理代理程式叢集，以平衡訊息負載 (第 9 章)
- 回復故障的代理程式 (第 64 頁的「啟動代理程式」)

受管理物件

- 調整連線工廠屬性，以確保用戶端應用程式以正確的方式運作 (第 144 頁的「連線工廠屬性」)
- 監視與管理實體目標 (第 6 章)
- 控制使用者對目標的存取 (第 128 頁的「實體目標的存取控制」)

用戶端管理

- 監視與管理長期訂閱 (請參閱第 101 頁的「管理長期訂閱」)。
- 監視與管理作業事件 (請參閱第 103 頁的「管理作業事件」)。

管理工具

Message Queue 管理工具分為兩類：

- 指令行公用程式
- 圖形化管理主控台

指令行公用程式

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

- **代理程式公用程式** (imqbrokerd) 會啟動代理程式並指定其配置特性，包括將多個代理程式一併連線至叢集。
- **指令行公用程式** (imqcmd) 會控制代理程式及其資源，並管理實體目標。
- **物件管理員公用程式** (imqobjmgr) 會管理物件存放區中不受限於提供者的**受管理物件**，這些受管理物件可透過 Java Naming and Directory Interface (JNDI) 進行存取。
- **資料庫管理者公用程式** (imqdbmgr) 會建立與管理符合 Java 資料庫連線 (JDBC) 標準的永久性存放區資料庫。
- **使用者管理員公用程式** (imqusermgr) 會寫入檔案式使用者儲存庫，以執行使用者認證與授權。
- **服務管理員公用程式** (imqsvcadmin) 會安裝與管理當成 Windows 服務的代理程式。
- **金鑰工具公用程式** (imqkeytool) 會產生自行簽署的憑證，以執行安全通訊端層 (SSL) 認證。

如需使用這些公用程式的詳細資訊，請參閱第 13 章。

管理主控台

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

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

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

遠端管理主機

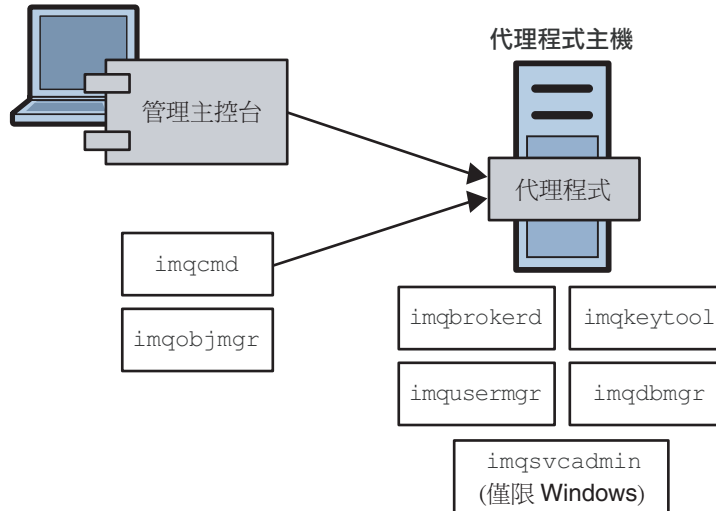


圖 1-1 本機與遠端管理公用程式

如需簡要實用的管理主控台介紹，請參閱第 2 章。更多詳細使用資訊，請參閱其各自的說明功能。

快速入門指導

本快速入門指導將引導您使用 Message Queue™ 管理主控台 (管理訊息代理程式及物件存放區的圖形化介面)，進行一些基本的管理作業，讓您對 Message Queue 管理有初步的認識。本章包含下列各節：

- 第 38 頁的「啟動管理主控台」
- 第 39 頁的「管理主控台線上說明」
- 第 40 頁的「使用代理程式」
- 第 45 頁的「使用實體目標」
- 第 50 頁的「使用物件存放區」
- 第 53 頁的「使用受管理物件」
- 第 58 頁的「執行範例應用程式」

本指導旨在設定當執行簡單 JMS 相容應用程式 HelloWorldMessageJNDI 時，所需的實體目標和受管理物件。此應用程式在範例應用程式目錄 (Solaris 和 Windows 平台的 demo 或 Linux 的 examples；請參閱附錄 A) 的 helloworld 子目錄中。您會在本指導的最後階段執行此應用程式。

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

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

- 啟動訊息代理程式
- 連線至代理程式以及使用管理主控台管理代理程式
- 在代理程式上建立實體目標
- 建立物件存放區，並且使用管理主控台連線至此物件存放區
- 將受管理物件增加至物件存放區，並檢視其特性

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

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

- 啓動代理程式
- 建立代理程式叢集
- 配置特定實體目標特性
- 管理用作永久性存放區的 JDBC 資料庫
- 管理使用者儲存庫
- 安裝當成 Windows 服務的代理程式
- 產生 SSL 憑證

本線上手冊會在後續各章討論上述所有工作。

啓動管理主控台

若要啓動管理主控台，請使用下列方法之一：

- 在 Solaris 上，輸入下列指令：

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

- 在 Linux 上，輸入下列指令：

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

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

顯示 [管理主控台] 視窗需等待數秒 (請參閱圖 2-1)。

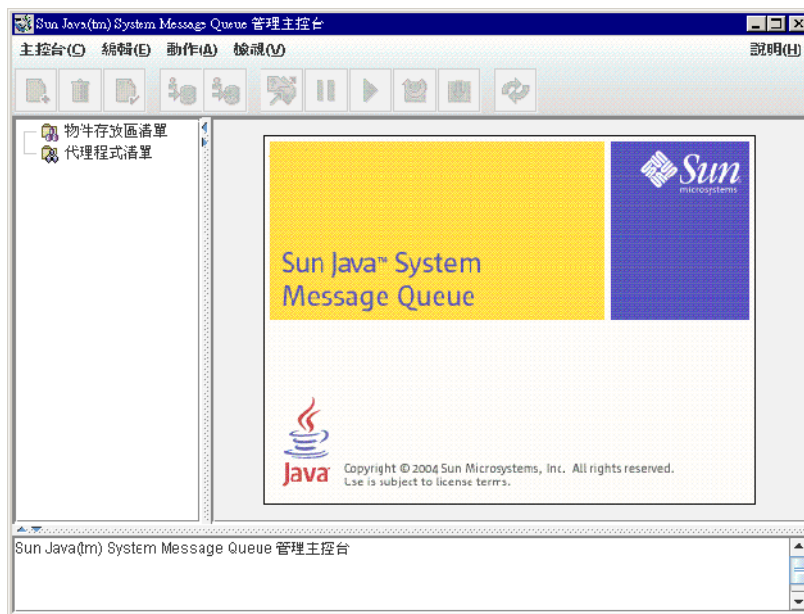


圖 2-1 管理主控台視窗

請花幾秒鐘的時間檢視 [管理主控台] 視窗。頂端有一個功能表列，功能表列下方為工具列，左側為瀏覽窗格，右側為結果窗格 (此時顯示的是表示 Sun Java System Message Queue 產品的圖形)，底部則為狀態窗格。

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

管理主控台線上說明

在 [管理主控台] 的說明功能中，包含有關如何使用主控台執行管理作業的完整資訊。若要使用說明功能，請從功能表列最右側的 [說明] 下拉式功能表中選擇 [概述]。即會顯示 [管理主控台] 的 [說明] 視窗 (圖 2-2)。

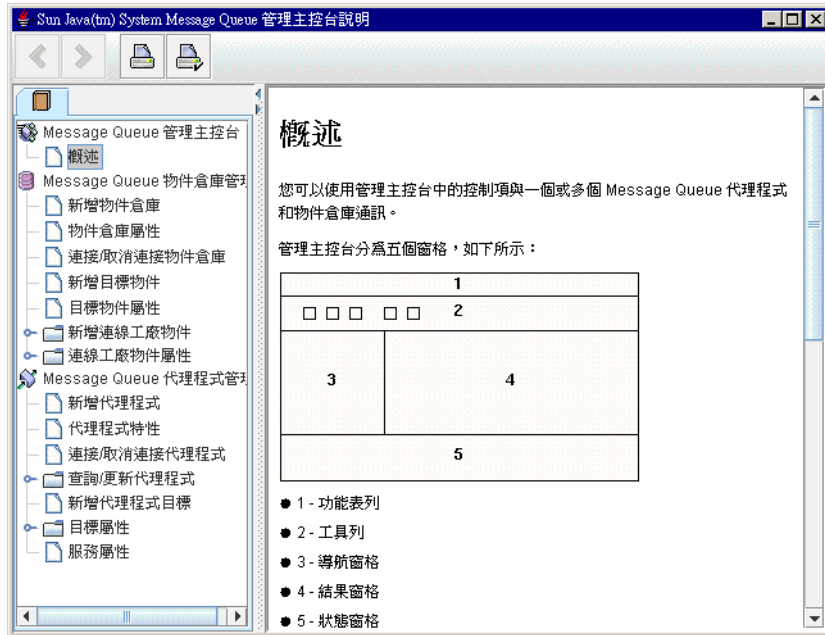


圖 2-2 管理主控台說明視窗

[說明] 視窗左側的瀏覽窗格，將所有主題區分為 3 個區域：Message Queue 管理主控台、Message Queue 物件存放區管理，以及 Message Queue 代理程式管理。各區域內皆有檔案及資料夾。資料夾內是多重標籤對話方塊的說明，檔案中則是簡單對話方塊或個別標籤的說明。當您選擇瀏覽窗格中的項目時，右側的結果窗格就會顯示該項目的內容。選擇 [概述] 項目時，會在結果窗格中顯示 [管理主控台] 視窗的框架檢視，並指明該視窗的各個窗格，如圖所示。

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

使用代理程式

本小節描述如何使用管理主控台，以連線至訊息代理程式並管理訊息代理程式。

啓動代理程式

您無法使用管理主控台啓動代理程式。請使用下列其中一種方法：

- 在 Solaris 上，輸入下列指令：


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

- 在 Linux 上，輸入下列指令：

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

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

如果使用 Windows 的 [開始] 功能表，會出現一個指令視窗，其中顯示如下的內容表示代理程式已就緒：

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

重新啟動 [管理主控台] 視窗。此時可將代理程式增加至主控台，並且與其連線。當您在 [管理主控台] 中為代理程式新增參照時，不必先啟動代理程式；但若要連線至代理程式，就必須先啟動它。

將代理程式新增至管理主控台

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

▼ 若要將代理程式新增至管理主控台

- 1 按一下 [管理主控台] 視窗瀏覽窗格中的 [代理程式清單] 項目，然後選擇 [動作] 功能表的 [增加代理程式]。

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

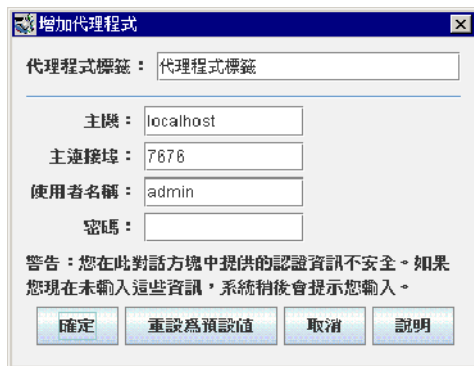


圖 2-3 [增加代理程式] 對話方塊

- 在 [代理程式標籤] 欄位中輸入代理程式名稱。

這可提供識別管理主控台中代理程式的標籤。

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

在此練習中，請將名稱 MyBroker 鍵入至 [代理程式標籤] 欄位。[密碼] 欄位請留白，在進行連線時指定密碼較為安全。

- 按一下 [確定] 以增加代理程式，並結束對話方塊。

新的代理程式會出現在瀏覽窗格的 [代理程式清單] 下，如圖 2-4 中所示。代理程式圖示上紅色的 X，表示代理程式目前尚未連線至管理主控台。

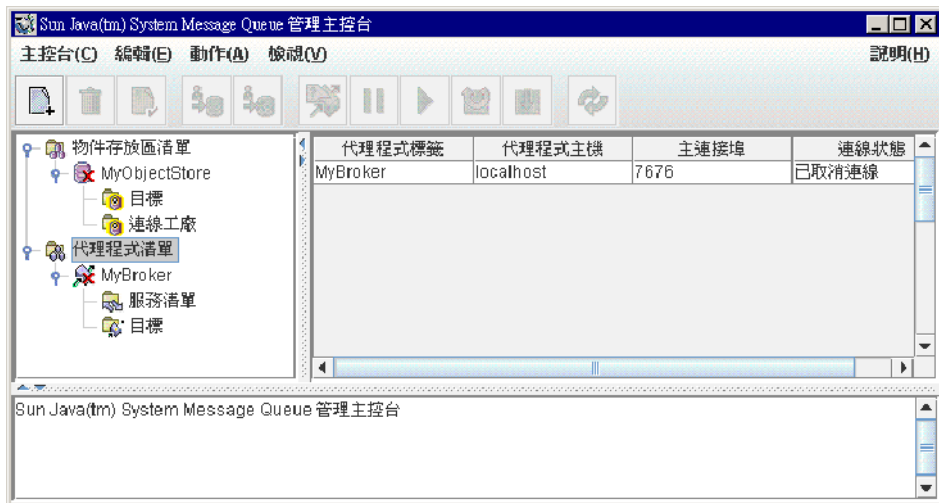


圖 2-4 管理主控台視窗中顯示的代理程式

只要增加了代理程式，就可使用 [動作] 功能表 (或快顯功能表) 的 [特性] 指令，顯示 [代理程式特性] 對話方塊 (與第 41 頁的「將代理程式新增至管理主控台」中所顯示的 [增加代理程式] 對話方塊類似)，以便您檢視或修改其中任一特性。

連線至代理程式

現在已將代理程式增加至管理主控台，接下來即可繼續連線作業。

▼ 連線至代理程式

- 1 按一下 [管理主控台] 視窗瀏覽窗格中的代理程式名稱，然後選擇 [動作] 功能表的 [連線至代理程式]。

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



圖 2-5 [連線至代理程式] 對話方塊

- 2 請輸入用以連線至代理程式的使用者名稱及密碼。

對話方塊一開始會顯示預設的使用者名稱 admin。在實際情況中，您應儘快建立安全的使用者名稱和密碼 (請參閱第 117 頁的「使用者認證」)；在此練習中，只需使用預設值即可。

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

- 3 按一下 [確定] 以連線至代理程式，並結束對話方塊。

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

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

檢視連線服務

透過代理程式提供的連線服務和它支援的實體目標可以識別此代理程式。

▼ 檢視可用的連線服務

- 1 在 [管理主控台] 視窗的瀏覽窗格中，選擇代理程式名稱下的 [服務清單]。

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



圖 2-6 檢視連線服務

- 在結果窗格中按一下服務名稱，以選擇該項服務。

在此練習中，請選擇名稱 jms。

- 從 [動作] 功能表中選擇 [特性]。

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

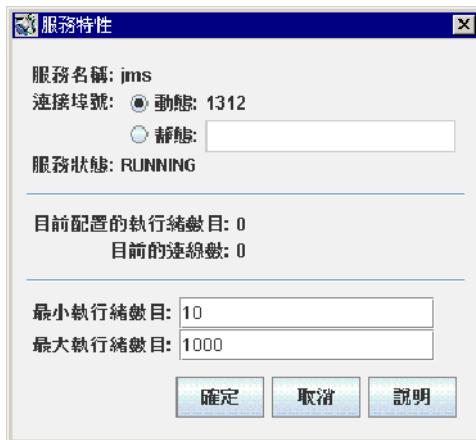


圖 2-7 [服務特性] 對話方塊

在此練習中，請勿變更任何連線服務的特性。

- 按一下 [確定] 以接受新的特性值，並結束對話方塊。

[動作] 功能表也包含暫停和重新繼續服務的指令。但是，當您選擇 admin 服務並且下拉 [動作] 功能表時，[暫停服務] 指令會是停用狀態。這是因為 admin 服務是 [管理主控台] 中與代理程式的連結：如果暫停此服務，將無法再存取代理程式。

使用實體目標

實體目標是訊息代理程式上的一個位置，其中保留接收自訊息產生器的訊息，以便稍後傳送至一個或多個訊息用戶。目標會依使用的**訊息傳送網域**分為兩種：**佇列**（點對點網域）與**主題**（發佈/訂閱網域）。如需有關訊息傳送網域及其相關聯目標的進一步討論，請參閱「Message Queue 技術摘要」。

建立實體目標

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

▼ 將實體目標新增至代理程式

- 1 在 [管理主控台] 視窗的瀏覽窗格中按一下代理程式名稱下的 [目標] 項目，然後選擇 [動作] 功能表的 [增加代理程式目標]。

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



圖 2-8 [增加代理程式目標] 對話方塊

- 2 在[目標名稱]欄位中輸入該實體目標的名稱。
請記下指定給此目標的名稱；稍後當您建立與此實體目標相對應的受管理物件時，將需要這個名稱。
在此練習中，請鍵入名稱 `MyQueueDest`。
- 3 選擇[佇列]或[主題]單選按鈕，以指定要建立的目標類型。
在此練習中，請選擇[佇列](如果尚未選取)。
- 4 按一下[確定]以增加實體目標，並結束對話方塊。
該實體目標會隨即出現在結果窗格中。

檢視實體目標特性

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

▼ 檢視或修改實體目標的特性

- 1 在 [管理主控台] 視窗的瀏覽窗格中，選擇代理程式名稱下的 [目標]。
結果窗格會出現可用實體目標的清單，其中顯示各目標的名稱、類型，以及目前的狀態。
- 2 在結果窗格中按一下實體目標名稱，以選擇該目標。
- 3 從 [動作] 功能表中選擇 [特性]。
即會出現 [代理程式目標屬性] 對話方塊 (圖 2-9)，顯示所選實體目標的目前狀態及配置資訊。您可使用此對話方塊變更各項配置特性，如該目標能容納的訊息、產生器及用戶上限。

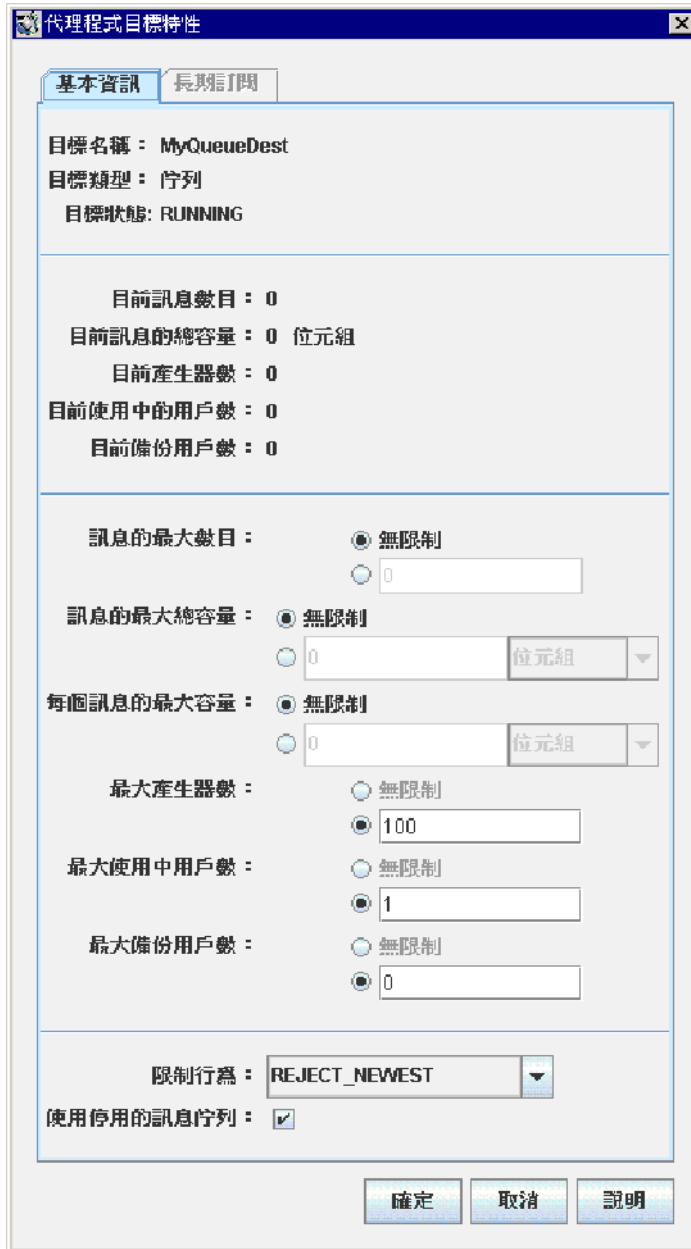


圖 2-9 [代理程式目標屬性] 對話方塊

在此練習中，請勿變更任何目標特性。

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

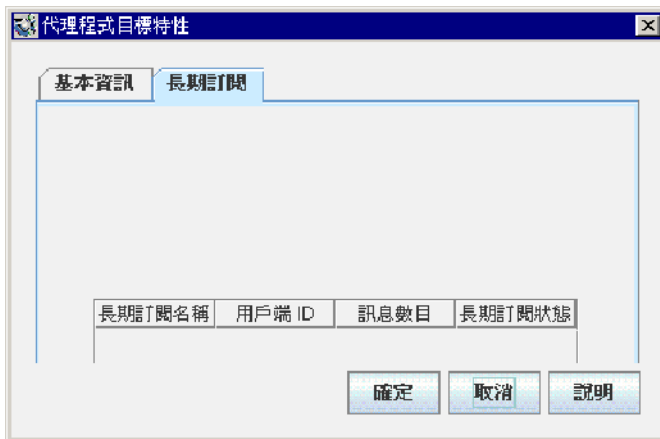


圖 2-10 [長期訂閱] 面板

您可使用 [長期訂閱] 面板中的 [清除] 和 [刪除] 按鈕，執行下列作業：

- 清除所有與長期訂閱相關聯的擱置訊息
- 移除主題的長期訂閱

佇列目標的 [長期訂閱] 標籤是停用的。

- 4 按一下 [確定] 以接受新的特性值，並結束對話方塊。

清除實體目標的訊息

清除實體目標的訊息，會移除所有與該目標相關聯的擱置訊息，從而清空該目標。

▼ 清除實體目標的訊息

- 1 在 [管理主控台] 視窗的瀏覽窗格中，選擇代理程式名稱下的 [目標]。
結果窗格會出現可用實體目標的清單，其中顯示各目標的名稱、類型，以及目前的狀態。
- 2 在結果窗格中按一下目標名稱，以選擇該目標。
- 3 從 [動作] 功能表中選擇 [清除訊息]。
即會出現確認對話方塊，要求您確認是否繼續該作業。

- 4 按一下 [是] 以確認作業，並結束確認對話方塊。

刪除實體目標

刪除目標會先清除目標的全部訊息，然後銷毀目標本身，將此目標從所屬的代理程式中永久移除。

▼ 刪除實體目標

- 1 在 [管理主控台] 視窗的瀏覽窗格中，選擇代理程式名稱下的 [目標]。
結果窗格會出現可用目標的清單，其中顯示各目標的名稱、類型，以及目前的狀態。
- 2 在結果窗格中按一下目標名稱，以選擇該目標。
- 3 從 [編輯] 功能表中選擇 [刪除]。
即會出現確認對話方塊，要求您確認是否繼續該作業。
- 4 按一下 [是] 以確認作業，並結束確認對話方塊。
在此練習中，請勿刪除先前建立的目標 MyQueueDest；請按一下 [否] 結束確認對話方塊，以免執行刪除作業。

使用物件存放區

物件存放區用以儲存 Message Queue 受管理物件，其中封裝特定 Message Queue 提供者的專用實作及配置資訊。物件存放區可以是簡易目錄存取協定 (LDAP) 目錄伺服器，也可以是本機檔案系統的目錄。

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

新增物件存放區

雖然管理主控台可讓您管理物件存放區，但是無法用來建立物件存放區；在建立物件存放區之前，用作物件存放區的 LDAP 伺服器或檔案系統目錄必須已經存在。然後，才能夠將現有的物件存放區增加至管理主控台，並為其建立參照，以便在主控台中操作此物件存放區。

備註 - 本章使用的範例應用程式，會假設物件存放區位於 C 磁碟機的 Temp 目錄中。如果 C 磁碟機上還沒有名為 Temp 的資料夾，請先建立一個，再進行後續的練習。(在非 Windows 平台上，您可以使用 /tmp 目錄，此目錄應已存在。)

▼ 將物件存放區新增至管理主控台

- 1 按一下 [管理主控台] 視窗瀏覽窗格中的 [物件存放區清單] 項目，然後選擇 [動作] 功能表的 [增加物件存放區]。

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



圖 2-11 [增加物件存放區] 對話方塊

- 2 在 [物件存放區標籤] 欄位中，輸入物件存放區名稱。
這可提供識別管理主控台中物件存放區的標籤。
在此練習中，請鍵入名稱 MyObjectStore。
- 3 輸入用來查找受管理物件的 JNDI 屬性值：
 - a. 請從 [名稱] 下拉式功能表中，選擇要指定的屬性名稱。
 - b. 在 [值] 欄位中鍵入屬性值。

c. 按一下 [增加] 按鈕，增加指定的屬性值。

特性摘要窗格中會出現特性及其值。

請為每個需要設定的屬性，重複步驟 a、b 和 c。

在此練習中，將 `java.naming.factory.initial` 屬性設定為

`com.sun.jndi.fscontext.RefFSContextFactory`

並且將 `java.naming.provider.url` 屬性設定為

`file:///C:/Temp`

(在 Solaris 或 Linux 平台上，則為 `file:///tmp`)。針對檔案系統物件存放區，您僅需設定這些屬性；如需 LDAP 存放區必要屬性值的相關資訊，請參閱第 141 頁的「LDAP 伺服器物件存放區」。

4 按一下 [確定] 以增加物件存放區，並結束對話方塊。

新的物件存放區會出現在瀏覽窗格的 [物件存放區清單] 下，如圖 2-12 中所示。物件存放區圖示上紅色的 X，表示物件存放區目前尚未連線至管理主控台。



圖 2-12 管理主控台視窗中顯示的物件存放區

當您按一下瀏覽窗格中的物件存放區時，其內容會在結果窗格中列出。因為您尚未在物件存放區中增加任何受管理物件，所以目標和連線工廠的 [計數] 欄都顯示 0。

只要增加了物件存放區，就可使用 [動作] 功能表 (或快顯功能表) 中的 [特性] 指令，顯示 [物件存放區特性] 對話方塊 (與圖 2-11 中所顯示的 [增加物件存放區] 對話方塊類似)，以便您檢視或修改其中任一特性。

連線至物件存放區

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

▼ 連線至物件存放區

- 按一下 [管理主控台] 視窗瀏覽窗格中的物件存放區名稱，然後選擇 [動作] 功能表的 [連線至物件存放區]。

您也可以在物件存放區名稱上按一下滑鼠右鍵，然後從快顯功能表中選擇 [連線至物件存放區]。不論何種方式，物件存放區圖示上紅色的 X 都會消失，這表示已經與管理主控台連線。

使用受管理物件

只要物件存放區連線至管理主控台，就可以繼續將受管理物件 (連線工廠及目標) 增加至此物件存放區。本節會描述作業方法。

備註 - 管理主控台僅會顯示 Message Queue 受管理物件。如果物件存放區含有非 Message Queue 物件，而該物件與所需增加的受管理物件具有相同的查找名稱，則嘗試執行增加作業時，會出現錯誤。

新增連線工廠

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

▼ 將連線工廠新增至物件存放區

- 1 確認物件存放區已連線至管理主控台 (請參閱第 52 頁的「連線至物件存放區」)。
- 2 在 [管理主控台] 視窗的瀏覽窗格中，按一下該物件存放區名稱下的 [連線工廠] 項目，然後選擇 [動作] 功能表的 [增加連線工廠物件]。

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

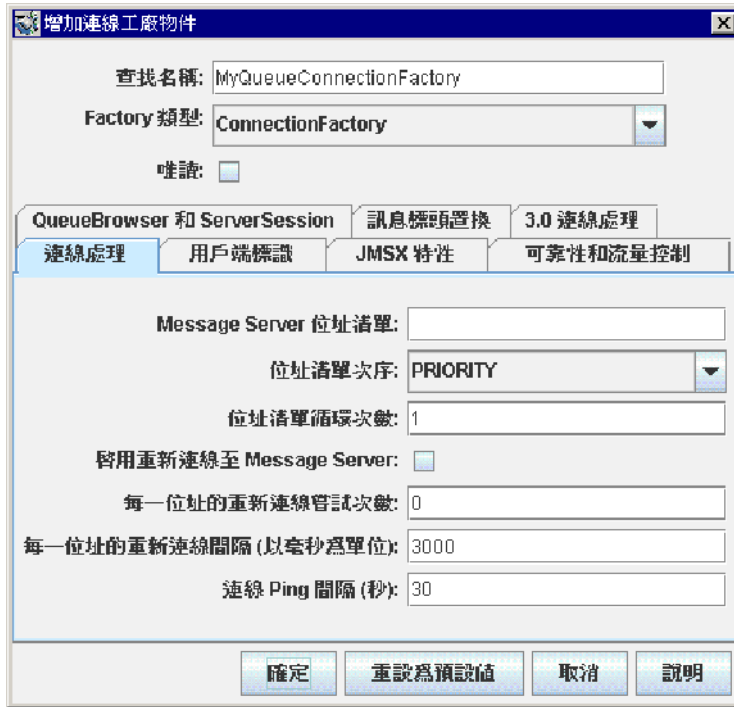


圖 2-13 [增加連線工廠物件] 對話方塊

- 3 在 [查找名稱] 欄位中輸入該連線工廠的名稱。
這是用戶端應用程式透過 JNDI 查找連線工廠時所使用的名稱。
在此練習中，請鍵入名稱 MyQueueConnectionFactory。
- 4 請從 [Factory 類型] 下拉式功能表中，選擇您希望建立的連線工廠類型。
在此練習中，請選擇 [QueueConnectionFactory]。
- 5 按一下 [連線處理] 標籤。
即會出現 [連線處理] 窗格，如圖 2-13 中所示。
- 6 在 [Message Server 位址清單] 欄位中，填入將要與此連線工廠建立連線的代理程式位址。
此位址清單可能包含一個代理程式；或在代理程式叢集的狀況下，包含多個代理程式。它會針對每個代理程式，指定代理程式連線服務、主機名稱及連接埠號等資訊。實際指定的資訊性質和語法會各有不同，需視所要使用的連線服務而定；如需詳細資訊，請參閱第 263 頁的「連線處理」。

在此練習中，不需要在 [Message Server 位址清單] 欄位中鍵入任何位址，這是因為範例應用程式 HelloWorldMessageJNDI 會讓連線工廠使用預設自動配置的標準位址清單屬性 (連線服務為 jms，主機名稱為 localhost，連接埠號為 7676)。

7 請視需要配置連線工廠的其他屬性。

[增加連線工廠物件] 對話方塊除了有 [連線處理] 窗格之外，另有其他多個面板，可用以配置連線工廠的各種屬性。

在此練習中，請勿變更其他任何屬性設定。但是，您還是可以按其他標籤，瞭解可指定的配置資訊種類，增進對此項作業的認識。請使用 [說明] 按鈕，瞭解有關這些其他配置面板的內容。

8 在適當情況下，按一下 [唯讀] 核取方塊。

這會鎖定連線工廠物件的配置屬性，以保持建立時所指定的值。不論是用戶端程式碼的程式設計方式或指令行的管理方式，都無法置換唯讀的受管理物件屬性。

在此練習中，請勿核取 [唯讀]。

9 按一下 [確定] 以建立連線工廠、將其增加至物件存放區，並結束對話方塊。

新的連線工廠會出現在結果窗格中。

新增目標

目標受管理物件代表代理程式上的實體目標，可讓用戶端將訊息傳送至該實體目標，而不受提供者專用配置及命名語法的影響。當用戶端傳送經由受管理物件定址的訊息時，代理程式會將該訊息傳送至對應的實體目標 (如果存在)。如果上述實體目標不存在，代理程式會自動建立一個實體目標 (如果已啟用自動建立功能，如第 45 頁的「[建立實體目標](#)」中所述)，並將訊息傳送至此；否則，就會產生一個錯誤，指出訊息無法傳送。

下列程序描述如何將目標受管理物件，增加至與現有實體目標相對應的物件存放區。

▼ 將目標新增至物件存放區

- 1 確認物件存放區已連線至管理主控台 (請參閱第 52 頁的「[連線至物件存放區](#)」)。
- 2 在 [管理主控台] 視窗的瀏覽窗格中，按一下物件存放區名稱下的 [目標] 項目，然後選擇 [動作] 功能表的 [增加目標物件]。

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



圖 2-14 [增加目標物件] 對話方塊

- 3 在 [查找名稱] 欄位中，輸入該目標受管理物件的名稱。
這是用戶端應用程式透過 JNDI 查找目標時所使用的名稱。
在此練習中，請鍵入名稱 MyQueue。
- 4 選擇 [佇列] 或 [主題] 單選按鈕，指定要建立的目標物件類型。
在此練習中，請選擇 [佇列] (如果尚未選取)。
- 5 在 [目標名稱] 欄位中，輸入對應實體目標的名稱。
這是您將實體目標增加至代理程式時，所指定的名稱 (請參閱第 45 頁的「使用實體目標」)。
在此練習中，請鍵入名稱 MyQueueDest。
- 6 在 [目標描述] 欄位中，輸入該目標的簡短說明；此為選擇性項目。
此欄位的內容僅供使用者參考，對用戶端作業沒有任何影響。
在此練習中，您可以刪除 [目標描述] 欄位的內容，或鍵入一些說明性文字，例如
Example destination for MQ Admin Guide tutorial
- 7 在適當情況下，按一下 [唯讀] 核取方塊。
這會鎖定目標物件的配置屬性，以保持建立時所指定的值。不論是用戶端程式碼的程式設計方式或指令行的管理方式，都無法置換唯讀的受管理物件屬性。
在此練習中，請勿核取 [唯讀]。
- 8 按一下 [確定] 以建立目標物件、將其增加至物件存放區，並結束對話方塊。
新的目標物件會隨即出現在結果窗格中，如圖 2-15 所示。

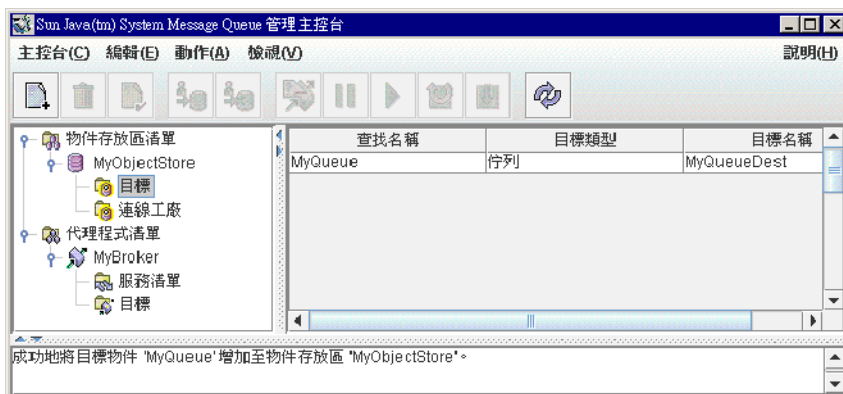


圖 2-15 管理主控台視窗中顯示的目標物件

檢視受管理物件特性

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

▼ 檢視或修改受管理物件的特性

- 1 在 [管理主控台] 視窗的瀏覽窗格中，選擇物件存放區名稱下的 [連線工廠] 或 [目標]。結果窗格會出現可用連線工廠或目標受管理物件的清單，其中顯示各項目的查找名稱和類型 (目標受管理物件清單另包括目標名稱)。
- 2 在結果窗格中按一下受管理物件的名稱，以選擇該受管理物件。
- 3 從 [動作] 功能表中選擇 [特性]。
此時會出現 [連線工廠物件特性] 或 [目標物件特性] 對話方塊，與 [增加連線工廠物件] (圖 2-13) 或 [增加目標物件] (圖 2-14) 對話方塊類似。您可使用此對話方塊，變更所選取物件的配置屬性。但請注意，您無法變更物件的查找名稱；唯一的方法是刪除該物件，然後再以所需的查找名稱增加新的受管理物件。
- 4 按一下 [確定] 以接受新的屬性值，並結束對話方塊。

刪除受管理物件

刪除受管理物件會將其從所屬物件存放區中永久移除。

▼ 刪除受管理物件

- 1 在 [管理主控台] 視窗的瀏覽窗格中，選擇物件存放區名稱下的 [連線工廠] 或 [目標]。結果窗格會出現可用連線工廠或目標受管理物件的清單，其中顯示各項目的查找名稱和類型 (目標受管理物件清單另包括目標名稱)。
- 2 請在結果窗格中按一下受管理物件的名稱，以選擇該受管理物件。
- 3 從 [編輯] 功能表中選擇 [刪除]。
此時會出現確認對話方塊，要求您確認是否繼續該作業。
- 4 按一下 [是] 以確認作業，並結束確認對話方塊。
在此練習中，請勿刪除先前建立的受管理物件 `MyQueue` 或 `MyQueueConnectionFactory`；請按一下 [否] 結束確認對話方塊，以免執行刪除作業。

執行範例應用程式

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

- 佇列實體目標的名稱是 `MyQueueDest`
- 佇列連線工廠受管理物件的 JNDI 查找名稱是 `MyQueueConnectionFactory`
- 佇列受管理物件的 JNDI 查找名稱是 `MyQueue`

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

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

▼ 執行範例應用程式

- 1 請使用下列其中一個指令 (依所用平台而定)，使包含 `HelloWorldmessageJNDI` 應用程式的目錄成為目前的目錄：

- 在 Solaris 上：

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

- 在 Linux 上：

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

- 在 Windows 上：

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

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

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

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

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

備註 - `jndi.jar` 檔案隨附於 JDK 1.4。除非使用舊版的 JDK，否則不需要將此檔案增加至 CLASSPATH。

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

- 在 Solaris 或 Linux 上：

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

- 在 Windows 上：

```
java HelloWorldMessageJNDI
```

如果應用程式成功執行，您應該會看到如下所示的輸出。

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

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

```
Creating connection to broker.
Connection to broker created.
```

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

第 2 部分

管理作業

- 第 3 章
- 第 4 章
- 第 5 章
- 第 6 章
- 第 7 章
- 第 8 章
- 第 9 章
- 第 10 章
- 第 11 章
- 第 12 章

啓動代理程式和用戶端

安裝 Sun Java System™ Message Queue™ 並執行一些基本準備步驟後，就可以開始啓動代理程式和用戶端。代理程式的配置由一組配置檔案所管理，可由傳送給代理程式公用程式 (imqbrokerd) 的指令行選項加以置換；如需更多資訊，請參閱第 4 章。

本章包含下列各節：

- 第 63 頁的「準備系統資源」
- 第 64 頁的「啓動代理程式」
- 第 68 頁的「移除代理程式」
- 第 69 頁的「啓動用戶端」

準備系統資源

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

同步化系統時鐘

啓動任何代理程式或用戶端之前，必須先將會與 Message Queue 系統互動的所有主機的時鐘同步化。如果您使用訊息過期設定 (即訊息有效存在的時間)，則此項同步化尤為重要。不同步的時鐘時間標記，可能會讓訊息無法按預期那樣過期或無法傳送訊息。同步化對代理程式叢集也十分重要。

您應該將系統配置為執行時間同步化協定，例如簡易網路時間協定 (SNTP)。Solaris 和 Linux 的 xntpd 常駐程式和 Windows 的 W32Time 服務通常可支援時間同步化功能。(如需配置此服務的詳細資訊，請參閱作業系統的文件。)代理程式執行後，請避免將系統時鐘往回設定。

設定檔案描述元限制

在 Solaris 和 Linux 平台上，執行用戶端或代理程式所在的 Shell，會對程序可以使用的檔案描述元數目指定軟式限制。在 Message Queue 中，用戶端建立的每個連線或代理程式接受的每個連線，均使用這些檔案描述元的其中一項。每個擁有永久性訊息的實體目標皆使用一個檔案描述元。

因此，檔案描述元限制會限制代理程式或用戶端可擁有的連線數目。依預設，Solaris 上的最大連線數目為 256，而 Linux 上為 1024。(實際上，因為會持續使用檔案描述元，所以連線限制會低於上述數目。)如果需要超過此限制的連線數目，您必須在執行用戶端或代理程式的每個 Shell 中，提高檔案描述元限制。如需如何執行此作業的資訊，請參閱 `ulimit` 線上手冊。

啓動代理程式

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

以互動方式啓動代理程式

您可以使用代理程式公用程式 (`imqbrokerd`)，從指令行以互動方式啓動代理程式。(也可以從 Windows 的 [開始] 功能表，啓動代理程式。)您無法使用管理主控台 (`imqadmin`) 或指令公用程式 (`imqcmd`) 來啓動代理程式，因為必須先執行代理程式，才能使用這些工具。

在 Solaris 和 Linux 平台上，代理程式實例一律必須由最初啓動該實例的同一位使用者啓動。每個代理程式實例都有自己的配置特性和檔案式訊息存放區。初次啓動代理程式實例時，Message Queue 會使用該使用者的檔案建立模式遮罩 (`umask`)，設定包含該代理程式實例配置資訊和永久性資料之目錄的權限。

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

```
imqbrokerd
```

該指令會在本地機器上使用預設連接埠 7676 上的連接埠對映器，啓動代理程式實例 `imqbroker` (請參閱第 72 頁的「[連接埠對映器](#)」)。

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

```
imqbrokerd -name myBroker
```

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


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

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

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

如需 `imqbrokerd` 指令的語法、子指令和選項的詳細資訊，請參閱第 218 頁的「代理程式公用程式」。如需這項資訊的快速摘要，請輸入下列指令：

```
imqbrokerd -help
```

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

```
imqbrokerd -license try
```

每次啓動代理程式時，都必須使用此選項，否則代理程式會預設為標準 Platform Edition 授權。

自動啓動代理程式

除了從指令行明確啓動代理程式之外，您也可以設定在系統啓動時自動啓動代理程式。執行此作業的方式，會因執行代理程式的平台 (Solaris、Linux 或 Windows) 而有所不同。

在 Solaris 和 Linux 上自動啓動

在 Solaris 和 Linux 系統上安裝 Message Queue 時，可啓用自動啓動作業的程序檔位於 `/etc/rc*` 樹狀目錄中。若要使用這些程序檔，您必須編輯配置檔案 `/etc/imq/imqbrokerd.conf` (Solaris) 或 `/etc/opt/sun/mq/imqbrokerd.conf` (Linux)，如下所示：

- 若要在系統啓動時自動啓動代理程式，請將 `AUTOSTART` 特性設定為 `YES`。
- 若要在異常結束後自動重新啓動代理程式，請將 `RESTART` 特性設定為 `YES`。
- 若要設定代理程式的啓動指令行引數，請指定一個或多個 `ARGS` 特性值。

在 Windows 上自動啓動

若要在 Windows 系統啓動時自動啓動代理程式，您必須將代理程式定義為 Windows 服務。然後就會在系統啓動時啓動代理程式，並且在背景中執行到系統關機為止。因此，除非您要啓動其他實例，否則不必使用 `imqbrokerd` 指令啓動代理程式。

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

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

在 Windows 系統上安裝 Message Queue 時，您可以將代理程式安裝為服務。安裝後，您可以使用服務管理員公用程式 (`imqsvcadmin`) 執行下列作業：

- 將代理程式新增為 Windows 服務
- 決定代理程式服務的啟動選項
- 移除當成 Windows 服務執行的代理程式

若要將啟動選項傳送給代理程式，可在 `imqsvcadmin` 指令中使用 `-args` 引數。它的作用等同於第 64 頁的「啟動代理程式」中描述的 `imqbrokerd` 指令之 `-D` 選項。使用指令公用程式 (`imqcmd`) 以控制代理程式照常運作。

如需 `imqsvcadmin` 指令的語法、子指令和選項的完整資訊，請參閱第 233 頁的「服務管理員公用程式」。

重新配置代理程式服務

如要重新配置安裝為 Windows 服務的代理程式，程序如下：

▼ 重新配置當成 Windows 服務執行的代理程式

1 停止服務。

- a. 從 Windows [開始] 功能表的 [設定] 子功能表中，選擇 [控制台]。
- b. 開啓 [系統管理工具] 控制台。
- c. 選擇 [服務] 圖示以執行服務工具，然後從 [檔案] 功能表或快顯式環境功能表選擇 [開啓]，或是直接連按兩下該圖示。
- d. 在 [服務 (本機)] 下方，選擇 [Message Queue Broker] 服務，然後從 [執行] 功能表中選擇 [內容]。
或者，您可以在 [訊息佇列代理程式] 上按一下滑鼠右鍵，然後從快顯功能表中選擇 [內容]，或直接連按兩下 [訊息佇列代理程式]。不論何種方式，都會出現 [訊息佇列代理程式特性] 對話方塊。
- e. 在 [內容] 對話方塊中的 [一般] 標籤下方，按一下 [停止] 即可停止代理程式服務。

2 移除服務。

在指令行上輸入指令

```
imqsvcadmin remove
```

- 3 重新安裝服務，使用 `-args` 選項指定不同的代理程式啓動選項，或使用 `-vmargs` 選項指定不同的 Java 版本引數。

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

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

使用替代 Java 執行階段

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

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

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

顯示代理程式服務啓動選項

若要決定代理程式服務的啓動選項，請在 `imqsvcadm` 指令中使用 `query` 選項，如範例 3-1 所示。

範例 3-1 顯示代理程式服務啓動選項

```
imqsvcadm query

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

服務啓動問題疑難排解

如果在啓動當成 Windows 服務的代理程式時出現錯誤，可以檢視記錄的錯誤事件：

▼ 查閱記錄的服務錯誤事件

- 1 開啓 Windows [系統管理工具] 控制台。
- 2 啓動 [事件檢視器] 工具。
- 3 選擇 [應用程式事件記錄檔]。

- 4 從 [執行] 功能表選擇 [重新整理]，以顯示任何錯誤事件。

移除代理程式

移除代理程式的程序會因平台不同而異，將會在以下各節說明。

移除 Solaris 或 Linux 上的代理程式

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

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

例如，如果代理程式的名稱是 `myBroker`，則指令如下：

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

此指令可刪除所指定代理程式的整個實例目錄。

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

如需 `imqbrokerd` 指令的語法、子指令和選項的詳細資訊，請參閱第 218 頁的「代理程式公用程式」。如需這項資訊的快速摘要，請輸入下列指令：

移除 Windows 上的代理程式服務

若要移除當成 Windows 服務執行的代理程式，請使用以下指令

```
imqcmd shutdown bkr
```

若要關閉代理程式，請在後面加上

```
imqsvcadmin remove
```

以移除服務。

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

請在移除代理程式服務之後，重新啟動您的電腦。

啓動用戶端

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

啓動 Java 用戶端應用程式時，請使用下列的指令行格式：

```
java clientAppName
```

若要啓動 C 用戶端應用程式，請使用應用程式開發者提供的格式。

應用程式的文件應會提供與應用程式設定的屬性值相關的資訊；您可能要從指令行置換部分屬性值。您也可能要爲任何使用 Java Naming and Directory Interface (JNDI) 查找功能尋找連線工廠的 Java 用戶端，在指令行中指定屬性。如果查找功能傳回的連線工廠比應用程式還舊，表示連線工廠不支援較新的屬性。在這種情況下，Message Queue 會將這些屬性設爲預設值；必要時，您可以使用指令行置換這些預設值。

若要從指令行指定 Java 應用程式的屬性值，請使用以下語法：

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

如第 16 章所述，*attribute* 的值必須是連線工廠受管理物件屬性。如果該值中含有空格，請將指令行的 *attribute= value* 部分括在引號內。

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

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

指令行中指定的主機名稱與連接埠，會置換應用程式本身的任何設定。

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

配置代理程式

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

本章包含下列各節：

- 第 71 頁的「代理程式服務」
- 第 82 頁的「設定代理程式特性」
- 第 84 頁的「配置永久性資料存放區」

如需代理程式配置特性的完整參考資訊，請參閱第 14 章。

代理程式服務

視服務或服務影響所及的代理程式元件之不同，代理程式配置特性可分為數個種類：

- **連線服務**：管理代理程式與其用戶端之間的實體連線，用來傳輸內送和外寄訊息。
- **路由服務**：路由和傳送 JMS 有效負載訊息，並控制由訊息服務所使用的訊息，以支援透過可靠的方式進行傳送。
- **永久性服務**：管理資料寫入永久性存放區，以及從永久性存放區擷取資料。
- **安全性服務**：對連線到代理程式的使用者進行認證，並授權使用者執行動作。
- **監視服務**：產生有關代理程式效能的度量與診斷資訊。

以下各節描述每項服務，以及針對特殊需求用以自訂服務的特性。

連線服務

訊息代理程式提供各種**連線服務**，以使用多種傳輸協定來支援應用程式與管理用戶端。與連線服務相關的代理程式配置特性，列於第 235 頁的「連線特性」中。

表 4-1 顯示可用的連線服務，依據兩種特徵區分：

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

表 4-1 Message Queue 連線服務

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

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

每個連線服務也支援特定的認證與授權功能；如需更多資訊，請參閱第 77 頁的「安全性服務」。

連接埠對映器

每種連線服務都是透過特定連接埠提供，此時需指定主機名稱 (或 IP 位址) 和連接埠號。您可以明確指定服務的靜態連接埠號，或由代理程式的**連接埠對映器**動態指定。連接埠對映器本身常駐於代理程式的**主連接埠**上，即通常位於標準連接埠號 7676。(若有必要，您可以使用代理程式配置特性 `imq.portmapper.port`，以其他連接埠號來置換此值。)依預設，每個連線服務皆會在啟動時對連接埠對映器進行自我註冊。當用戶端建立與代理程式的連線時，Message Queue 用戶端執行階段會先連絡連接埠對映器，以請求連接埠號供需要的連線服務使用。

此外，您也可以使用 `imq.serviceName.protocolType.port` 配置特性置換連接埠對映器，並明確指定連線服務的靜態連接埠號 (其中 `serviceName` 與 `protocolType` 表示特定的連線服務，如表 4-1 所示)。(只有 `jms`、`ssljms`、`admin` 與 `ssladmin` 連線服務能夠以這種方式進行配置；`httpjms` 與 `httpsjms` 服務使用不同的配置特性，如附錄 C 所述)。但是，靜態連接埠通常僅在特殊情況下使用，例如透過防火牆連線 (請參閱第 139 頁的「透過防火牆連線」)，一般不建議使用。

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

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

執行緒池管理

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

您選擇的執行緒模型會指定執行緒是專屬於單一連線，還是由多個連線共用：

- 在**專屬模型**中，代理程式的每個連線均需要兩個執行緒：一個用於內送訊息，另一個用於外寄訊息。這會限制可支援的連線數量，但可確保較高的效能。
- 在**共用模型**中，連線在傳送或接收訊息時會由共用執行緒進行處理。此模型中的各個連線不需要專屬執行緒，因此能夠提高可用連線的數量，但會降低效能，因為必須使用較多資源來管理執行緒。

代理程式的 `imq.serviceName.threadpool_model` 特性，可指定已知連線服務應使用這兩種模型中的哪一種。此特性會採用下列兩個字串值之一：`dedicated` 或 `shared`。若未明確設定特性，則會依預設採用 `dedicated`。

您也可以設定代理程式特性 `imq.serviceName.min_threads` 與 `imq.serviceName.max_threads`，來指定服務執行緒池的最小與最大執行緒數。當可用的執行緒超過此最小臨界值時，`Message Queue` 將會在執行緒成為閒置狀態時關閉這些執行緒，直到再次達到最小值為止，以節省記憶體資源。在負載量較大的情況下，執行緒的數目會增加，直至達到執行緒池的最大數目為止；此時會拒絕新的連線，直至有執行緒變為可用為止。

共用執行緒模型會使用**分散執行緒**，將執行緒指定給使用中的連線。代理程式特性 `imq.shared.connectionMonitor_limit` 可指定由單一分散執行緒監視的最大連線數。此特性的值越小，執行緒指定給連線的速度就越快。`imq.ping.interval` 特性會指定一個以秒為單位的時間間隔，讓代理程式依此間隔定期測試 (「ping」) 連線，以驗證該連線是否仍在使用中，如此就可以預先偵測到連線失敗，避免所嘗試執行的訊息傳輸失敗。

路由服務

用戶端連線至代理程式後，即可路由訊息及傳送訊息。在此階段中，代理程式必須建立及管理不同類型的實體目標，以確保訊息能順利傳送並有效利用資源。您可以使用第 237 頁的「路由特性」所述的代理程式配置特性，以根據應用程式需要的方式管理這些工作。

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

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

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

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

- **系統記憶體臨界值**定義了記憶體用量的層級，代理程式會依據這些層級採取越來越嚴格的動作，以便防止記憶體超載。這類的用量定義共有四種：
 - **綠色**：有大量記憶體可用。
 - **黃色**：代理程式記憶體開始減少。
 - **橙色**：代理程式記憶體不足。
 - **紅色**：代理程式無記憶體可用。

用以定義這些層級的記憶體用量百分比，分別由代理程式特性

`imq.green.threshold`、`imq.yellow.threshold`、`imq.orange.threshold` 與 `imq.red.threshold` 所指定，預設值為綠色 0%、黃色 80%、橙色 90%、紅色 98%。

記憶體用量到達下一層級時，代理程式即會按階段做出回應，首先將訊息由使用中記憶體移出至永久性存放區，然後減少產生器產生的非永久性訊息，直到最後停止訊息傳入代理程式中。(上述兩種方法皆會降低代理程式效能。)若要減少產生的訊息，必須將每個批次作業大小限制為 `imq.resourceState.count` 特性所指定的訊息數，其中 `resourceState` 分別為 `green`、`yellow`、`orange` 或 `red`。

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

永久性服務

如果要在代理程式故障時回復，則此代理程式需要重新建立其訊息傳送作業的狀態。為執行此作業，代理程式必須將狀態資訊儲存在**永久性資料存放區**。重新啟動代理程式時，它會使用儲存的資料來重新建立目標和長期訂閱、回復永久性訊息、回復開啓的作業事件，並為未傳送的訊息重新建立路由表。然後，它可以重新繼續訊息傳送。

Message Queue 支援檔案式與 JDBC 型永久性模組 (請參閱圖 4-1)。檔案式永久性會使用個別的檔案來儲存永久性資料；而 JDBC 型永久性則使用 Java 資料庫連結 (JDBC™) 介面，將代理程式連線至與 JDBC 相容的資料存放區。檔案式永久性通常快於 JDBC 型永久性；但是某些使用者偏好 JDBC 相容存放區所提供的備援和管理控制功能。代理程式配置特性 `imq.persist.store` (請參閱表 14-4) 可指定使用哪一種形式的永久性。

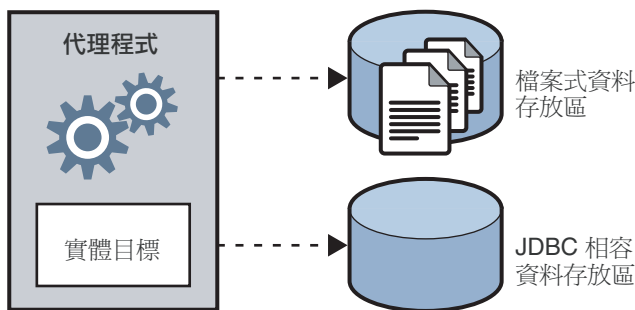


圖 4-1 永久性資料存放區

檔案式永久性

依預設，Message Queue 會使用檔案式永久性資料存放區，其中的個別檔案會儲存如訊息、目標、長期訂閱與作業事件等永久性資料。與檔案式永久性相關的代理程式配置特性，會在第 241 頁的「檔案式永久性」中列出。

檔案式存放區位於以資料存放區所屬之代理程式實例的名稱 (`instanceName`) 作為辨識依據的目錄中：

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

(請參閱附錄 A，以瞭解 instances 目錄的所在位置。)代理程式上的每個目標，都會將其保留訊息的子目錄傳送到該目標上。

備註 - 由於永久性資料存放區可能包含機密或專有訊息，因此您應保護 `.../instances/instanceName/fs350/` 目錄的安全性，以防止未經授權的存取；請參閱第 87 頁的「保護永久性資料」。

除了訊息以外的所有永久性資料，都儲存於個別的檔案中：其中一個檔案用於儲存目標、一個檔案用於儲存長期訂閱，一個檔案用於儲存作業事件狀態資訊。大部分訊息會儲存在由變數型記錄組成的單一檔案中。您可以壓縮此檔案，以便減少新增或移除訊息時的分段程序(請參閱第 112 頁的「壓縮實體目標」)。此外，超過特定臨界值大小的訊息會儲存於自身的個別檔案中，而非儲存於變數型記錄檔案。您可以使用代理程式特性 `imq.persist.file.message.max_record_size` 配置此臨界值大小。

代理程式會為這些個別訊息檔案維護一個檔案池：檔案不再使用時，不會被直接刪除，而會傳回其目標目錄中容納閒置檔案的檔案池中，以供日後其他訊息重複使用。代理程式特性 `imq.persist.file.destination.message.filepool.limit` 可指定池中的最大檔案數。當目標中的個別訊息檔案數超過此限制時，若檔案已不需使用，就會直接被刪除，而不會傳回池中。

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

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

JDBC 型永久性

您也可以不使用檔案式永久性，而設置代理程式存取任何可透過 JDBC 相容驅動程式存取的資料存放區。其中包含設定適當的 JDBC 相關代理程式配置特性，以及使用資料庫管理者公用程式 (`imqdbmgr`)，建立具有適當模式的資料庫。如需詳細資訊，請參閱第 85 頁的「配置 JDBC 型存放區」。

配置代理程式使用 JDBC 資料庫的特性，列於第 242 頁的「JDBC 型永久性」中。您可以透過以下方式指定這些特性：使用每個代理程式實例的實例配置檔 (config.properties)，或者使用代理程式公用程式 (imqbrokerd) 或資料庫管理者公用程式 (imqdbmgr) 的 -D 指令行選項。

imq.persist.jdbc.driver 特性可提供在連線資料庫時要使用的 JDBC 驅動程式之 Java 類別名稱。另外還有特性可指定連線至現有資料庫 (imq.persist.jdbc.opendburl)、建立新資料庫 (imq.persist.jdbc.createdburl) 以及關閉資料庫連線 (imq.persist.jdbc.closedburl) 時所需的 URL。

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

由多個代理程式實例共用的 JDBC 資料庫中，配置特性 imq.persist.jdbc.brokerid 可為每個實例指定唯一的實例識別碼，以附加到資料庫表格的名稱中。(由於內嵌式資料庫僅為一個代理程式實例儲存資料，因此通常不需要此屬性。)其餘與 JDBC 相關的配置特性，可用以自訂建立資料庫模式的 SQL 程式碼，每個資料庫表格各有一個適用的特性。例如，imq.persist.jdbc.table.IMQSV35 特性提供的 SQL 指令可用以建立版本表格；imq.persist.jdbc.table.IMQCCREC35 可用於建立配置變更記錄表格；imq.persist.jdbc.table.IMQDEST35 則用於建立目標表格... 等等。請參閱表 14-6 以取得完整清單。

備註 - 由於資料庫系統所需的 SQL 語法有所不同，因此請務必查看資料庫供應商提供的文件，以瞭解詳細資訊。

安全性服務

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

- **認證**可確保只有經過驗證的使用者才能與代理程式建立連線。
- **授權**可指定哪些使用者或群組有存取資源及執行特定作業的權限。
- **加密**可保護訊息，以防透過連線傳送時遭到篡改。

身為 Message Queue 管理員，您必須負責設定代理程式進行使用者認證及使用者授權作業時所需的資訊。與安全性服務相關的代理程式特性，列於第 245 頁的「安全性特性」中。布林值特性 imq.accesscontrol.enabled 的角色為主開關，可控制是否要將存取控制套用於整個代理程式；若要進行細部控制，您可置換特定連線服務的此設定，只要設定 imq.serviceName.accesscontrol.enabled 特性即可，其中 serviceName 是連線服務的名稱，如表 4-1 所示：例如 imq.httpjms.accesscontrol.enabled。

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

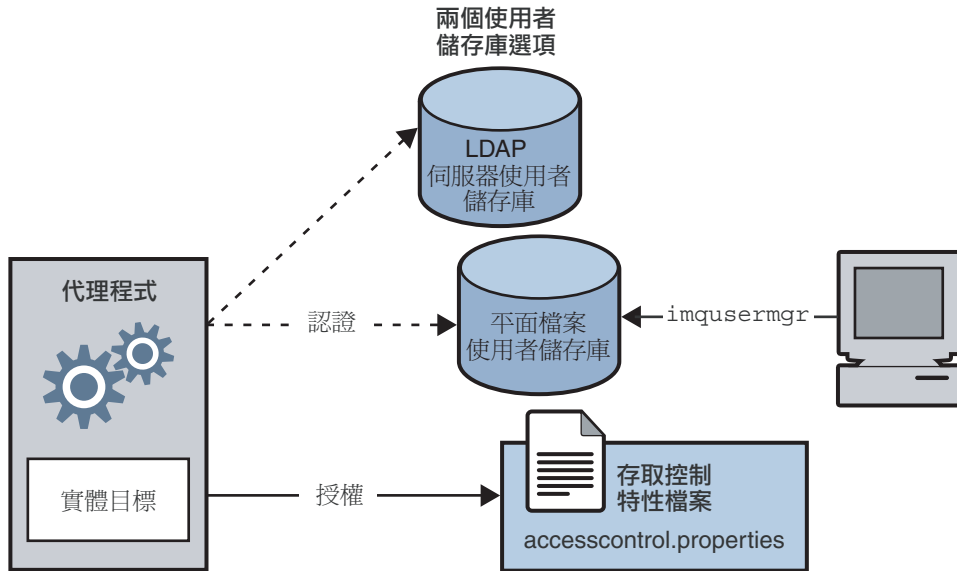


圖 4-2 安全性支援

如圖 4-2 所示，您可以將使用者資料儲存在隨附於 Message Queue 服務的平面檔案使用者儲存庫中，也可以外掛至預先存在的簡易目錄存取協定 (LDAP) 儲存庫：

- 若您選擇平面檔案儲存庫，就必須使用 Message Queue 使用者管理員公用程式 (`imqusermgr`) 管理儲存庫。此為內建選項，使用非常方便。
- 如果要使用現有的 LDAP 伺服器，可以使用 LDAP 供應商提供的工具來寫入和管理此使用者儲存庫。此外，您必須在代理程式的實例配置檔案中設定特性，讓代理程式能查詢 LDAP 伺服器，以取得使用者和群組的相關資訊。

代理程式的 `imq.authentication.basic.user_repository` 特性可指定要使用的儲存庫類型。一般而言，如果延展性很重要，或者如果需要不同代理程式共用儲存庫 (例如在使用代理程式叢集的情況下)，則使用 LDAP 儲存庫是不錯的選擇。如需設置平面檔案或 LDAP 使用者儲存庫的更多資訊，請參閱第 117 頁的「使用者認證」。

認證

請求連線至代理程式的用戶端必須提供使用者名稱與密碼，供代理程式將其與使用者儲存庫中儲存的資料進行比較。從用戶端傳輸到代理程式的密碼，會以 Base64 編碼 (適用於平面檔案儲存庫) 或訊息摘要 (MD5) 雜湊法 (適用於 LDAP 儲存庫) 進行編碼。選擇哪一種編碼方式，由代理程式的 `imq.authentication.type` 特性整體控制，或由特定連線服務的 `imq.serviceName.authentication.type` 控

制。`imq.authentication.client.response.timeout` 特性會設定認證請求的逾時時間間隔。

如第 138 頁的「密碼檔案」所述，您可以選擇將密碼存入**密碼檔案**中，而不需以互動方式提示您提供密碼。布林值代理程式特性 `imq.passfile.enabled` 可控制此選項。若此特性為 `true`，`imq.passfile.dirpath` 與 `imq.passfile.name` 特性就會提供密碼檔案的目錄路徑與檔案名稱。`imq.imqcmd.password` 特性 (可內嵌於密碼檔案中) 可指定管理使用者在使用指令公用程式 (`imqcmd`) 管理代理程式、連線服務、連線、實體目標、長期訂閱與作業事件所使用的認證密碼。

若您使用 LDAP 型使用者儲存庫，則能使用多種代理程式特性來配置 LDAP 查找的各種設定。LDAP 伺服器本身的位址 (主機名稱與連接埠號) 由

`imq.user_repository.ldap.server` 所指定。`imq.user_repository.ldap.principal` 特性會提供連結到 LDAP 儲存庫所需的辨別名稱，而 `imq.user_repository.ldap.password` 則會提供相關的密碼。其他特性會指定個別使用者與群組搜尋的目錄基底與選用 JNDI 篩選器，以及使用者和群組名稱的提供者專用屬性識別碼等等；請參閱第 245 頁的「**安全性特性**」以瞭解詳細資訊。

授權

經過認證後，使用者即有權執行各項 Message Queue 的相關活動。身為 Message Queue 管理員，您可以定義使用者群組，並指定其個別的使用者成員身份。預設的存取控制特性檔案僅明確參考一個群組 `admin` (請參閱第 120 頁的「群組」)。此群組中的使用者具有 `admin` 連線服務的連線權限，此權限可讓使用者執行管理功能，如建立目標、監視及控制代理程式。依預設，您定義的任何其他群組中的使用者，均無法取得 `admin` 服務連線。

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

- 連線至代理程式
- 存取目標：為任何給定目標或所有目標建立用戶、產生器或佇列瀏覽器
- 自動建立目標

加密

若要加密在用戶端與代理程式之間傳送的訊息，您需要使用基於安全通訊端層 (SSL) 標準的連線服務。透過在已啓用 SSL 的代理程式與用戶端之間建立加密的連線，SSL 可提供連線層級的安全性。

若要使用 SSL 型的 Message Queue 連線服務，您必須使用金鑰工具公用程式 (imqkeytool)，產生成對的私密金鑰/公開金鑰。此公用程式會將公開金鑰內嵌於自我簽署的憑證中，並放置在 Message Queue 金鑰庫中。金鑰庫本身受密碼保護；若要解除鎖定，您必須在啟動時提供由 `imq.keystore.password` 特性所指定的金鑰庫密碼。一旦金鑰庫解除鎖定，代理程式即可將憑證傳送至請求連線的任何用戶端。然後，用戶端即可使用此憑證設定與代理程式的加密連線。

`imq.audit.enabled` 代理程式特性可控制記錄到 Message Queue 代理程式記錄檔中的稽核記錄；如需更多資訊，請參閱第 140 頁的「稽核記錄」。

監視服務

代理程式包含可監視和診斷應用程式和代理程式效能的元件。這些元件包括：

- 產生資料的元件，即度量產生器與記錄事件的代理程式程式碼
- 記錄程式元件，可將輸出資訊寫入到多個輸出通道
- 度量訊息產生者，可將包含度量資訊的 JMS 訊息傳送到主題目標，以供 JMS 監視用戶端使用

圖 4-3 說明一般方案。用以配置監視服務的代理程式特性，列於第 248 頁的「檢視特性」中。

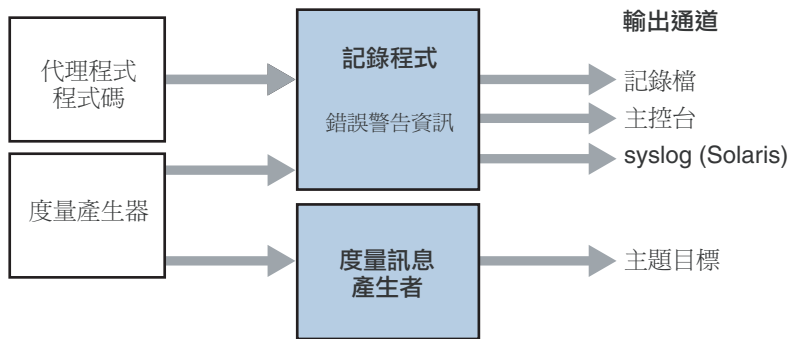


圖 4-3 監視支援

度量產生器

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

記錄程式

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

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

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

請參閱第 248 頁的「檢視特性」，以瞭解與記錄有關的其他代理程式特性，並參閱第 164 頁的「配置及使用代理程式記錄功能」，以進一步瞭解如何配置記錄程式，以及如何利用記錄程式取得效能資訊。

度量訊息產生者(企業版)

度量訊息產生者會在固定時間間隔接收度量產生器傳送的資訊，並將資訊寫入**度量訊息**，接著根據訊息中包含的度量資訊類型，將訊息傳送到其中一個度量主題目標(請參閱表 4-2)。訂閱這些度量主題目標的 Message Queue 用戶端，可以使用這些訊息並處理訊息所包含的度量資料。這允許開發者建立自訂監視工具以支援訊息傳送應用程式。如需每個度量訊息類型中所報告之度量數目的詳細資訊，請參閱「Message Queue Developer's Guide for Java Clients」。

表 4-2 度量主題目標

主題名稱	度量資訊的類型
<code>mq.metrics.broker</code>	代理程式度量
<code>mq.metrics.jvm</code>	Java 虛擬機器度量
<code>mq.metrics.destination_list</code>	目標及其類型的清單
<code>mq.metrics.destination.queue.queueName</code>	特定佇列的目標度量
<code>mq.metrics.destination.topic.topicName</code>	特定主題的目標度量

代理程式特性 `imq.metrics.topic.enabled` 與 `imq.metrics.topic.interval`，可分別控制訊息是否要傳送到度量主題目標及傳送的頻率。`imq.metrics.topic.timetolive` 與 `imq.metrics.topic.persist` 特性可指定此類訊息的使用期限及其是否具有永久性。

除了度量訊息內文包含的資訊外，每個訊息的標頭還包含提供下列額外資訊的特性：

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

這些特性對處理不同類型或不同代理程式的度量訊息之用戶端應用程式十分有用。

設定代理程式特性

您可以使用下列兩種方法之一來指定代理程式的配置特性：

- 編輯代理程式的配置檔案
- 直接從指令行提供特性值

以下兩節將說明這兩種配置代理程式的方法。

配置檔案

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

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

此外，各個代理程式實例也有其本身的**實例配置檔案**，如下所述。如果您連線叢集中的代理程式實例，可能還需要使用**叢集配置檔案**以指定叢集配置資訊；如需更多資訊，請參閱第 251 頁的「[叢集配置特性](#)」。

啟動時，代理程式會合併不同配置檔案中的特性值。如圖 4-4 所示，這些檔案會形成一個階層，在此階層中，實例配置檔案的指定值會置換安裝配置檔案中的指定值，而安裝配置檔案中的指定值則會置換預設配置檔案中的指定值。在階層頂端，您可以手動置換任何在配置檔案中指定的特性值，只要使用 `imqbrokerd` 指令的指令行選項即可。

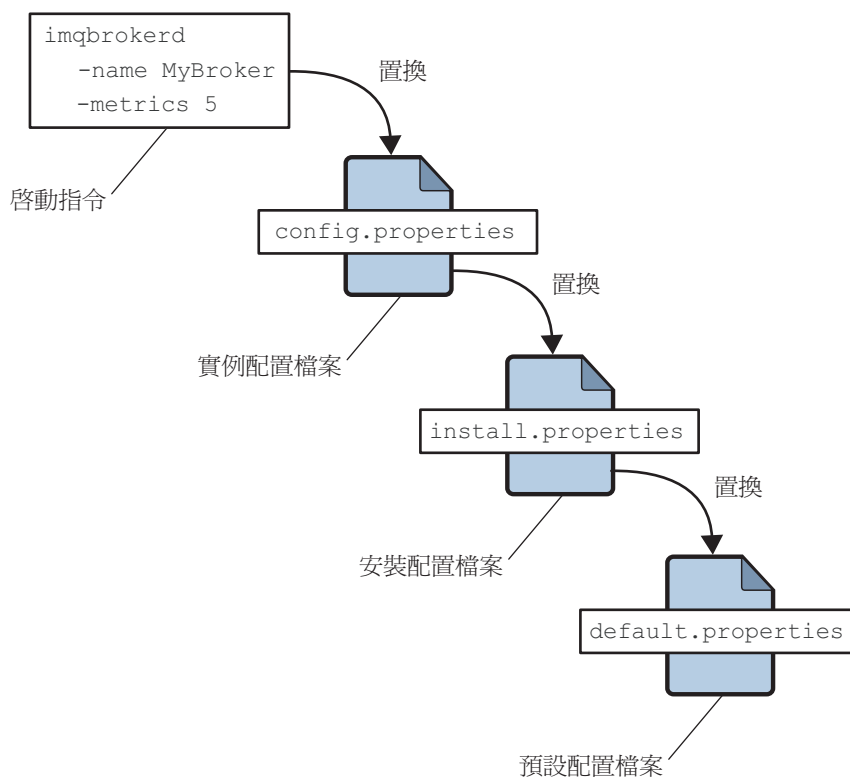


圖 4-4 代理程式配置檔案

編輯實例配置檔案

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

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

(請參閱附錄 A，以瞭解 `instances` 目錄的所在位置。)若該檔案不存在，您必須在啟動代理程式時使用 `-name` 選項 (請參閱第 218 頁的「代理程式公用程式」)，指定可供 Message Queue 建立檔案的實例名稱。

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

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

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

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

例如，以下項目指定代理程式會在記憶體和永久性存放區中最多保留 50,000 個訊息，之後則會拒絕其他訊息：

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

以下項目指定每天 (86,400 秒) 會建立一個新的記錄檔：

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

請參閱第 71 頁的「代理程式服務」與第 14 章，以瞭解可用代理程式配置特性及其預設值的相關資訊。

從指令行設定配置選項

您可以在啟動代理程式時或啟動之後，在指令行上輸入代理程式配置選項。

啟動時，可以使用代理程式公用程式 (`imqbrokerd`) 來啟動代理程式實例。使用指令的 `-D` 選項，可用以指定任何代理程式配置特性及其值；如需更多資訊，請參閱第 64 頁的「啟動代理程式」與第 218 頁的「代理程式公用程式」。如果您使用服務管理員公用程式 (`imqsvcadmin`) 將代理程式啟動為 Windows 服務，則 `-args` 選項可用來指定啟動配置特性；請參閱第 233 頁的「服務管理員公用程式」。

執行代理程式實例時，您也可以變更某些代理程式特性。若要修改執行中代理程式的配置，可以使用指令公用程式的 `imqcmd update bkr` 指令；請參閱第 93 頁的「更新代理程式特性」與第 223 頁的「代理程式管理」。

配置永久性資料存放區

代理程式的永久性資料存放區中含有實體目標、長期訂閱、訊息、作業事件與確認等項目的相關資訊。Message Queue 代理程式的預設配置會使用檔案式永久性存放區，但您可以重新配置，以外掛可透過 JDBC 相容驅動程式加以存取的任何資料存放區。代理程式配置特性 `imq.persist.store` (請參閱表 14-4) 可指定使用哪一種形式的永久性。

本節將說明如何設定代理程式使用永久性存放區。它包括以下主題：

- 第 85 頁的「配置檔案式存放區」

- 第 85 頁的「配置 JDBC 型存放區」
- 第 87 頁的「保護永久性資料」

配置檔案式存放區

建立代理程式實例時，會自動建立檔案式資料存放區。此存放區位於代理程式的實例目錄中；請參閱[附錄 A](#)，以瞭解確切位置。

依預設，Message Queue 會以非同步方式將作業寫入至磁碟。作業系統可以緩衝這些作業，以提供更好的效能。但是，如果寫入作業期間發生意外的系統失敗，則訊息可能會遺失。若要增強可靠性(必須犧牲效能)，您可以設定代理程式特性 `imq.persist.file.sync`，改為以同步方式寫入資料。如需此特性的詳細討論，請參閱第 75 頁的「檔案式永久性」與表 14-5。

啓動代理程式實例時，可以使用 `imqbrokerd` 指令的 `-reset` 選項來清除檔案系統存放區。如需此選項及其子選項的更多資訊，請參閱第 218 頁的「代理程式公用程式」。

配置 JDBC 型存放區

若要配置讓代理程式使用 JDBC 型永久性，請在代理程式的實例配置檔案中設定與 JDBC 相關的特性，並且建立適當的資料庫模式。Message Queue 資料庫管理者公用程式 (`imqdbmgr`) 使用 JDBC 驅動程式和代理程式配置特性來建立及管理資料庫。也可以使用資料庫管理者從資料庫中刪除毀壞的表格，或使用其他資料庫作為資料存放區。如需更多資訊，請參閱第 231 頁的「資料庫管理者公用程式」。

備註 - Oracle 和 PointBase 資料庫產品有範例配置可供使用。這些檔案的位置視平台而異，且已列在[附錄 A](#) 相關表格的「範例應用程式與配置」中。此外，PointBase 內嵌式版本、PointBase 伺服器版本與 Oracle 的範例，會在實例配置檔案 `config.properties` 中以註釋值的形式提供。

▼ 配置 JDBC 型資料存放區

- 1 在代理程式的配置檔案中設定與 JDBC 相關的特性。

相關特性在[第 76 頁的「JDBC 型永久性」](#)中討論並於表 14-6 中列出。特別是，您必須將代理程式 `imq.persist.store` 特性設為 `jdbc` (請參閱[第 241 頁的「永久性特性」](#))。

- 2 將 JDBC 驅動程式 `.jar` 檔案的副本或符號連結放入下列位置：

- 在 Solaris 上：

```
/usr/share/lib/imq/ext/
```

- 在 Linux 上：

```
/opt/sun/mq/share/lib/
```

- 在 Windows 上：

```
IMQ_VARHOME\\lib\\ext
```

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

```
% cp j2eeSDKInstallDirectory/pointbase/lib/pointbase.jar /usr/share/lib/imq/ext
```

下列指令則會建立符號連結：

```
% ln -s j2eeSDKID/lib/pointbase/pointbase.jar /usr/share/lib/imq/ext
```

3 建立 Message Queue 永久性所需的資料庫模式。

使用 `imqdbmgr create all` 指令 (針對內嵌式資料庫) 或 `imqdbmgr create tbl` 指令 (針對外部資料庫)；請參閱第 231 頁的「資料庫管理者公用程式」。

a. 變更 `imqdbmgr` 所在的目錄：

- 在 Solaris 上：

```
cd /usr/bin
```

- 在 Linux 上：

```
cd /opt/sun/mq/bin
```

- 在 Windows 上：

```
cd IMQ_HOME\\bin
```

b. 輸入 `imqdbmgr` 指令：

```
imqdbmgr create all
```

備註 – 若使用內嵌式資料庫，最好在下列目錄中建立：

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

如果內嵌式資料庫沒有使用者名稱和密碼保護，則可能由檔案系統權限所保護。若要確保代理程式可讀取和寫入資料庫，則執行此代理程式的使用者應與使用 `imqdbmgr` 指令建立內嵌式資料庫的使用者相同。

保護永久性資料

永久性存放區可以包含暫時儲存的訊息檔案和其他資訊。由於這些訊息可能包含專用資訊，請務必保護資料存放區，以防止未經授權的存取。本節將說明如何確保檔案式或 JDBC 型資料存放區的資料安全。

保護檔案式存放區

使用檔案式永久性的代理程式，會將永久性資料寫入平面檔案資料存放區中，而此存放區的位置會隨平台而有所不同 (請參閱[附錄 A](#))：

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

其中，*instanceName* 是識別代理程式實例的名稱。

instanceName/fs350/ 目錄會在代理程式實例首次啟動時建立。保護此目錄的程序取決於執行代理程式所在的作業系統平台：

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

保護 JDBC 型存放區

使用 JDBC 型永久性的代理程式，會將永久性資料寫入與 JDBC 相容的資料庫中。對於由資料庫伺服器所管理的資料庫 (例如 Oracle)，建議您建立使用者名稱和密碼，以存取 Message Queue 資料庫表格 (表格名稱以 IMQ 開頭)。如果資料庫不允許保護個別表格，請建立僅由 Message Queue 代理程式使用的專用資料庫。請參閱資料庫供應商提供的文件，以取得如何建立使用者名稱及密碼存取的相關資訊。

代理程式開啓資料庫連線所需的使用者名稱和密碼，可透過代理程式配置特性來提供。但若在啟動代理程式時將其作為指令行選項提供，則有更高的安全性，方法是使用 imqbrokerd 指令的 -dbuser 與 -dbpassword 選項 (請參閱第 218 頁的「代理程式公用程式」)。

對於代理程式透過資料庫的 JDBC 驅動程式直接存取的內嵌式資料庫，通常是透過設定檔案在儲存永久性資料的目錄上的權限來確保此資料庫的安全，第 87 頁的「保護檔案式存放區」中會有詳細說明。但是為確保代理程式和資料庫管理者公用程式都能讀取和寫入該資料庫，兩者應該由同一位使用者執行。

管理代理程式

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

- 第 89 頁的「必要條件」 第 89 頁的「必要條件」
- 第 90 頁的「使用 `imqcmd` 公用程式」 第 90 頁的「使用 `imqcmd` 公用程式」
- 第 92 頁的「顯示代理程式資訊」
- 第 93 頁的「更新代理程式特性」
- 第 94 頁的「暫停和重新繼續代理程式」
- 第 95 頁的「關閉和重新啓動代理程式」
- 第 95 頁的「顯示代理程式度量」
- 第 96 頁的「管理連線服務」
- 第 100 頁的「取得有關連線的資訊」
- 第 101 頁的「管理長期訂閱」
- 第 103 頁的「管理作業事件」

本章未包含有關管理代理程式的所有相關主題。下列幾章包含其他主題：

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

必要條件

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

- 使用 `imqbrokerd` 公用程式指令啓動代理程式。代理程式執行後，才能使用其他指令行公用程式。
- 決定是要設定 Message Queue™ 管理使用者，或是使用預設帳號。您必須指定使用者名稱和密碼，才能使用管理指令。

當您安裝 Message Queue 時，同時也會安裝預設平面檔案使用者儲存庫。此儲存庫出廠時隨附兩個預設項目：admin 使用者和 guest 使用者。測試 Message Queue 時，您可以使用預設使用者名稱和密碼 (admin/ admin) 來執行 imqcmd 公用程式。

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

- 如果您要與代理程式之間使用安全的連線，請在目標代理程式實例上設定並啟用 ssladmin 服務。如需更多資訊，請參閱第 130 頁的「訊息加密」。

使用 imqcmd 公用程式

imqcmd 公用程式能讓您管理代理程式及其服務。

如需 imqcmd 指令的語法、子指令和選項等參考資訊，請參閱第 13 章。管理實體目標的參考資訊另外位於第 15 章中。

顯示說明

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

例如，下列指令顯示 imqcmd 的說明：

```
imqcmd -H
```

如果您輸入的指令行除了子指令或其他選項之外，另包含 -h 或 -H 選項，則 imqcmd 公用程式只會處理 -h 或 -H 選項。忽略指令行的所有其他項目。

顯示產品版本

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

```
imqcmd -v
```

如果您輸入的指令行除了子指令或其他選項之外，另包含 -v 選項，則 imqcmd 公用程式只會處理 -v 選項。忽略指令行的所有其他項目。

指定使用者名稱和密碼

每個 imqcmd 子指令都要經由使用者儲存庫進行認證，因此需要使用者名稱和密碼。只有使用 -h 或 -H 選項來顯示說明的指令以及使用 -v 選項來顯示產品版本的指令除外。

指定使用者名稱

使用 `-u` 選項指定管理使用者名稱。如果省略使用者名稱，指令會提示您輸入。例如，下列指令會顯示預設代理程式的相關資訊：

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

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

指定密碼

您可以使用下列任一方法指定密碼：

- 建立密碼檔案 (passfile)，將密碼輸入該檔案中。在指令行中使用 `-passfile` 選項，提供密碼檔案名稱。
- 讓指令提示您輸入密碼。

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

指定代理程式名稱和連接埠

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

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

範例

本小節中的範例說明如何使用 `imqcmd`。

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

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

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

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

下列範例列出在使用連接埠 7676 的 `localhost` 上執行的代理程式特性。該指令的初始逾時設定爲 20 秒，重試 (逾時後) 次數設定爲 7。使用者密碼位於 `myPassfile` 密碼檔案中，該檔案位於呼叫指令時所在的目前目錄中。

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

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

顯示代理程式資訊

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

`query bkr` 子指令的語法如下：

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

該子指令會列出預設代理程式或指定主機和連接埠上的代理程式之特性目前的設定。它也會顯示連線到指定代理程式的執行中代理程式 (在多代理程式叢集中) 清單。

例如：

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

提示您輸入密碼後，指令會輸出類似下列的內容：

```
Version                               3.6  
Instance Name                         imqbroker  
Primary Port                          7676  
  
Current Number of Messages in System  0  
Current Total Message Bytes in System 0  
  
Current Number of Messages in Dead Message Queue 0  
Current Total Message Bytes in Dead Message Queue 0  
  
Log Dead Messages                     true  
Truncate Message Body in Dead Message Queue false  
  
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` 子指令更新下列代理程式特性：

- `imq.autocreate.queue`
- `imq.autocreate.topic`
- `imq.autocreate.queue.maxNumActiveConsumers`
- `imq.autocreate.queue.maxNumBackupConsumers`
- `imq.cluster.url`
- `imq.destination.DMQ.truncateBody`
- `imq.destination.logDeadMsgs`
- `imq.log.level`
- `imq.log.file.rolloversecs`
- `imq.log.file.rolloverbytes`
- `imq.system.max_count`
- `imq.system.max_size`
- `imq.message.max_size`
- `imq.portmapper.port`

`update bkr` 子指令的語法如下：

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

該子指令變更預設代理程式或指定主機和連接埠上的代理程式的指定屬性。例如，以下指令會關閉佇列目標的自動建立功能：

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

第 14 章中描述這些特性。

暫停和重新繼續代理程式

啓動代理程式之後，您可以使用 `imqcmd` 子指令控制代理程式的狀態。

暫停代理程式

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

但是，暫停代理程式不會暫停 `admin` 連線服務。此服務可讓您執行所需的管理作業，以管理代理程式的訊息流量。暫停代理程式也不會暫停 `cluster` 連線服務。但是，叢集中的訊息傳送會根據叢集中不同代理程式所執行的傳送功能而有差異。因此，暫停叢集中的代理程式可能會造成某些訊息流量變慢。

`pause bkr` 子指令的語法如下：

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

該指令會暫停預設代理程式或指定主機和連接埠上的代理程式。

以下指令會暫停連接埠 1588 上 `myhost` 執行的代理程式。

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

您也可以暫停個別連線服務和個別實體目標。如需更多資訊，請參閱第 99 頁的「暫停和重新繼續連線服務」和第 110 頁的「暫停和重新繼續實體目標」。

重新繼續代理程式

重新繼續代理程式會重新啓動代理程式的服務執行緒，而且此代理程式會重新繼續偵聽連接埠。

`resume bkr` 子指令的語法如下：

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

該子指令會重新繼續預設代理程式或指定主機和連接埠上的代理程式。

以下指令會重新繼續連接埠 7676 上 `localhost` 執行的代理程式。

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

關閉和重新啓動代理程式

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

shutdown bkr 子指令的語法如下：

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

該子指令會關閉預設代理程式或指定主機和連接埠上的代理程式。

以下指令會關閉連接埠 1572 上 ctrlsrv 執行的代理程式：

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

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

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

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

顯示代理程式度量

若要顯示有關代理程式的度量資訊，請使用 metrics bkr 子指令。

metrics bkr 子指令的語法如下：

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

該子指令顯示預設代理程式或指定主機和連接埠上的代理程式的代理程式度量。

您可以使用 -m 選項，指定下列一種要顯示的度量類型：

- **ttl** 顯示傳入和傳出代理程式的訊息和資料封包度量 (預設度量類型)。
- **rts** 顯示 (每秒) 傳入和傳出代理程式的訊息和資料封包流量速率度量。
- **cxn** 顯示連線、虛擬記憶體堆疊和執行緒。

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

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

例如，若要取得每 10 秒時間間隔，代理程式傳入與傳出的訊息速率，請使用：

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

此指令的輸出如下：

```
-----
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
-----
```

如需有關代理程式收集和報告之資料的詳細說明，請參閱第 278 頁的「代理程式通用度量」。

管理連線服務

imqcmd 公用程式包括允許您執行以下連線服務管理作業的子指令：

- 第 97 頁的「列出連線服務」
- 第 97 頁的「顯示連線服務資訊」
- 第 98 頁的「更新連線服務特性」
- 第 99 頁的「顯示連線服務度量」
- 第 99 頁的「暫停和重新繼續連線服務」

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

表 5-1 Message Queue 連線服務

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

表 5-1 Message Queue 連線服務 (續)

服務名稱	服務類型	協定類型
ssladmin (Enterprise Edition)	ADMIN	TLS (SSL 型安全性)

您可以使用 `imqcmd` 子指令來管理整個連線服務，或管理個別連線服務。如果子指令的目標是個別服務，請使用 `-n` 選項，以表 5-1 之 [服務名稱] 欄位中所列的其中一個名稱進行指定。

列出連線服務

若要列出代理程式上可用的連線服務，請使用 `list svc` 子指令。

`list svc` 子指令的語法如下：

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

該子指令列出預設代理程式或指定主機和連接埠上的代理程式上的所有連線服務。

以下指令列出在使用連接埠 7676 的 `localhost` 上所執行代理程式的所有服務：

```
imqcmd list svc -u 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
```

顯示連線服務資訊

若要查詢並顯示單一服務的相關資訊，請使用 `query` 子指令。

`query svc` 子指令的語法如下：

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

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

例如：

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

提示輸入密碼後，指令會輸出類似下列的內容：

```
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
```

更新連線服務特性

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

表 5-2 imqcmd 更新的連線服務特性

特性	說明
port	指定給需更新之服務的連接埠 (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量。
maxThreads	指定給服務的最大執行緒數目。

`update` 子指令的語法如下：

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

該子指令更新在預設代理程式或指定主機和連接埠上的代理程式上執行的指定服務的指定屬性。如需服務屬性的描述，請參閱第 235 頁的「連線特性」。

以下指令將指定給 jms 服務的執行緒最小數目變更為 20。

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

顯示連線服務度量

若要顯示有關單一服務的度量資訊，請使用 `metrics` 子指令。

`metrics` 子指令的語法如下：

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

該子指令顯示預設代理程式或指定主機和連接埠上的代理程式上的指定服務之度量。

使用 `-m` 選項指定要顯示的度量類型：

- **tll** 顯示經由指定連線服務，傳入和傳出代理程式的訊息和資料封包的度量。(預設度量類型)。
- **rts** 顯示經由指定連線服務，(每秒) 傳入和傳出代理程式的訊息和資料封包流量速率度量。
- **cxn** 顯示連線、虛擬記憶體堆疊和執行緒。

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

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

例如，若要取得由 `jms` 連線服務處理的訊息和資料封包等累計總數，請使用：

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

提示輸入密碼後，指令會輸出類似下列的內容：

```
-----
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
```

如需有關使用 `imqcmd` 以報告連線服務度量的更多詳細描述，請參閱第 279 頁的「[連線服務度量](#)」。

暫停和重新繼續連線服務

除 `admin` 服務無法暫停之外，若要暫停其他任何服務，請使用 `pause svc` 和 `resume svc` 子指令。

`pause svc` 子指令的語法如下：

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

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

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

暫停服務會造成以下影響：

- 代理程式會停止接受暫停服務上的新用戶端連線。如果 Message Queue 用戶端嘗試開啓新的連線，將會發生異常。
- 暫停服務上的所有現有連線會維持持續作用狀態，但在重新繼續服務前，代理程式會暫停此類連線上的所有訊息處理。(例如，如果用戶端嘗試傳送訊息，在重新繼續服務前，`send` 方法會阻斷此訊息。)
- 將會維護任何已被代理程式接收之訊息的訊息傳送狀態。(例如，作業事件不會中斷，並且會在重新繼續服務後，繼續訊息傳送。)

若要重新繼續服務，請使用 `resume svc` 子指令。

`resume svc` 子指令的語法如下：

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

該子指令會重新繼續在預設代理程式或指定主機和連接埠上代理程式執行的指定服務。

取得有關連線的資訊

`imqcmd` 公用程式包括可讓您列出和取得連線相關資訊的子指令。

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

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

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

例如，下列指令會列出預設代理程式上的所有連線：

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

提示輸入密碼後，指令會輸出類似下列的內容：

Listing all the connections on the broker specified by:

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

-----
Connection ID      User    Service  Producers  Consumers  Host
-----
1964412264455443200  guest  jms      0          1          127.0.0.1
1964412264493829311  admin  admin    1          1          127.0.0.1
```

Successfully listed connections.

若要查詢並顯示單一服務的相關資訊，請使用 `query` 子指令。

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

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

例如：

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

提示輸入密碼後，指令會輸出類似下列的內容：

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

管理長期訂閱

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

- 列出長期訂閱
- 清除長期訂閱的所有訊息
- 銷毀長期訂閱

長期訂閱是由用戶端註冊為長期的主題訂閱；它具有唯一的身份，並要求代理程式保留此訂閱的訊息(即使它的用戶成為非使用中的用戶)。通常，代理程式僅在為長期訂閱者保留的訊息過期時，才刪除該訊息。

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

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

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

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

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

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。

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

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

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

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

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

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

管理作業事件

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

每個作業事件都有 Message Queue 作業事件 ID：代理程式上可用來唯一識別作業事件的 64 位元號碼。分散式作業事件另有由分散式作業事件管理程式指定的分散式作業事件 ID (XID)，此 ID 長達 128 個位元組。Message Queue 會維護 Message Queue 作業事件 ID 與 XID 的關聯。

如果發生故障，分散式作業事件很可能一直處於 PREPARED 狀態，而不會被確定。因此，作為管理員，您可能需要監視作業事件，然後回復或確定處於準備狀態的作業事件。

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

```
imqcmd list txn
```

例如，以下指令會列出代理程式中的所有作業事件。

```
imqcmd list txn
```

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

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

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

例如，如果代理程式的自動回復特性設定為 `false` (請參閱表 14-2)，則您必須在啟動代理程式時，手動確定或回復處於 PREPARED 狀態的作業事件。

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

`query` 子指令可讓您查閱相同的資訊，以及一些其他值：用戶端 ID、連線識別，以及分散式作業事件 ID (XID)。 `query txn` 子指令的語法如下：

```
imqcmd query txn -n transactionID
```

例如，下列範例的輸出內容如下：

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

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

使用 `commit` 和 `rollback` 子指令，確定或回復分散式作業事件。如前所述，僅可確定或回復處於 `PREPARED` 狀態的作業事件。

`commit` 子指令的語法如下：

```
imqcmd commit txn -n transactionID
```

例如：

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

`rollback` 子指令的語法如下。子指令：

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

如需更多資訊，請參閱表 14-2 中的 `imq.transaction.autorollback` 特性。

還可配置代理程式，以便在啓動代理程式時自動回復處於 `PREPARED` 狀態的作業事件。

管理實體目標

本章說明如何使用 `imqcmd` 公用程式，管理實體目標。Message Queue™ 訊息可以經由代理程式上的實體目標，路由至其使用者用戶端。代理程式會管理實體目標的相關記憶體和永久性存放區，並且設定其運作方式。

在代理程式叢集中，您只要在一個代理程式上建立實體目標，叢集會將此實體目標傳播至所有代理程式。應用程式用戶端可以訂閱或使用叢集中任何代理程式上的主題或佇列，因為代理程式會合作跨叢集路由訊息。但是唯有產生訊息的原始代理程式，可以管理訊息的永久性和確認。

本章涵蓋下列主題：

- 第 106 頁的「使用指令公用程式」
- 第 106 頁的「建立實體目標」
- 第 108 頁的「列出實體目標」
- 第 108 頁的「顯示實體目標資訊」
- 第 110 頁的「更新實體目標特性」
- 第 110 頁的「暫停和重新繼續實體目標」
- 第 111 頁的「清除實體目標」
- 第 112 頁的「銷毀實體目標」
- 第 112 頁的「壓縮實體目標」
- 第 114 頁的「使用停用的訊息佇列」

表 13-5 提供有關使用 `imqcmd` 子指令來管理實體目標並且完成這些工作的完整參考資訊。

如需實體目標的簡介，請參閱「Message Queue 技術摘要」。

備註 - 用戶端應用程式與實體目標互動時，會使用 `Destination` 物件。為了提供者獨立性和可移植性，用戶端通常使用管理員建立的目標物件，即目標管理物件。您可以配置供用戶端應用程式使用的受管理物件，如第 8 章所述。

使用指令公用程式

使用 Message Queue 指令公用程式 (imqcmd) 管理實體目標。imqcmd 指令的語法，與您將其用於管理其他代理程式服務時相同。

如需有關 imqcmd、其子指令和選項的參考資訊，請參閱第 13 章。

子指令

表 6-1 列出本章描述的 imqcmd 子指令用法。如需有關這些子指令的參考資訊，請參閱第 226 頁的「實體目標管理」。

表 6-1 指令公用程式的實體目標子指令

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

建立實體目標

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

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

例如，若要建立佇列目標，請輸入以下指令：

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

若要建立主題目標，請輸入以下指令：

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

建立實體目標時，您必須指定以下項目：

- 實體目標類型 t (topic) 或 q (queue)。
- 實體目標名稱。命名規則如下：
 - 名稱只能使用字母數字字元。不能包含空格。
 - 名稱可以用英文字母字元、底線字元 (_) 或錢幣符號 (\$) 開頭。不能以 mq 字元字串開頭。

實體目標特性的非預設值。

更新實體目標時，您也可以設定特性。

很多實體目標特性都會影響代理程式記憶體資源和訊息流量。例如，您可以指定能夠傳給實體目標的產生器數目、產生器可以傳送的訊息數量和容量，以及到達實體目標限制時代理程式的回應方式。這些限制類似於代理程式配置特性所控制的一切代理程式限制。

下列特性可用於佇列目標和主題目標：

- `maxNumMsgs`：指定實體目標中允許未使用訊息的最大數目。
- `maxTotalMsgBytes`：指定實體目標中允許給未使用訊息的最大總記憶體容量 (以位元組為單位)。
- `limitBehavior`：指定達到記憶體限制執行緒時，代理程式回應的方式。
- `maxBytesPerMsg`：指定實體目標中允許的單一訊息最大容量 (以位元組為單位)。
- `maxNumProducers`：指定實體目標中產生器的最大數目。
- `consumerFlowLimit`：指定單一批次中，傳送給用戶的最大訊息數目。
- `isLocalOnly`：僅適用於代理程式叢集。指定在其他代理程式上不複製實體目標，且限制實體目標僅傳送訊息給本機用戶 (即連線到已建立實體目標之代理程式的用戶)。
- `useDMQ`：指定捨棄實體目標之停用的訊息或放到停用的訊息佇列中。

下列特性僅用於佇列目標：

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

如需有關實體目標特性的完整參考資訊，請參閱第 15 章。

針對自動建立的目標，請在代理程式實例配置檔案中設定預設特性值。表 14-3 中有自動建立特性的參考資訊。

列出實體目標

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

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

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

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

如果省略目標類型，則會列出所有類型的實體目標。

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

例如，若要取得連接埠 4545 上 `myHost` 執行的代理程式的所有實體目標，請輸入以下指令：

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

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

顯示實體目標資訊

若要取得實體目標的目前特性，請使用 `query dst` 子指令。`query dst` 子指令的語法如下：

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

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

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

此指令的輸出如下：

```
-----  
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     1000
Is Local Destination    false
Local Delivery is Preferred false
Use Dead Message Queue  true

```

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

您可以使用 `update dst` 子指令來變更一個或多個特性值 (請參閱第 110 頁的「更新實體目標特性」)。

更新實體目標特性

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

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

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

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

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

請參閱第 15 章，以獲得可以更新的特性清單。

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

備註 – 停用的訊息佇列是特殊的實體目標，其特性與其他目標的特性不同。如需更多資訊，請參閱第 114 頁的「使用停用的訊息佇列」。

暫停和重新繼續實體目標

您可以暫停實體目標，以便控制從產生器傳送訊息到目標，或從目標傳送訊息到用戶，或同時控制上述兩個動作。要特別注意的是，您可以暫停進入目標的訊息流量，以避免目標在訊息產生速度高於使用速度時收到過多訊息。壓縮實體目標之前，必須先暫停該目標。

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

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

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

範例：

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

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

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

該子指令重新繼續向指定類型和名稱之暫停目標傳送訊息。如果沒有指定目標類型與名稱，則會重新繼續所有目標。

範例：

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

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

清除實體目標

您可以清除目前在實體目標上形成佇列的所有訊息。清除實體目標即表示會刪除儲存於該目標的所有訊息。

當累積的訊息佔用太多的系統資源時，您可能需要清除訊息。當佇列沒有註冊的用戶用戶端並要接收許多訊息時，可能需要清除訊息。如果非使用中的主題長期訂閱者未成為使用中訂閱者，也可能需要清除訊息。在這兩種情況下，沒有必要保留訊息。

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

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

此子指令會清除實體目標上指定類型與名稱的訊息。

範例：

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

如果已關閉代理程式，並且不想在其重新啟動時傳送舊的訊息，請使用 `-reset messages` 選項清除舊的訊息，例如：

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

這可避免重新啟動代理程式後清除目標的麻煩。

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

銷毀實體目標

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

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

此子指令會銷毀指定類型與名稱的實體目標。

範例：

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

銷毀實體目標會清除此目標上的所有訊息，並從代理程式中移除此目標；此作業不可復原。

您不能銷毀停用的訊息佇列。

壓縮實體目標

如果您使用檔案式資料存放區作為訊息的永久性存放區，即可監視磁碟使用，並於需要時壓縮磁碟。

檔案式訊息存放區的結構方式，是讓訊息可以儲存在與收納訊息之實體目標相對應的目錄中。在各個實體目標的目錄中，大部分訊息會儲存在一個檔案中，該檔案由大小不定的記錄所組成。(若要減少分段程序，大小超過可配置臨界值的訊息會儲存在其個別的檔案中。)

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

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

監視實體目標的磁碟使用

若要監視實體目標的磁碟使用，請使用下列指令：

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

此指令的輸出如下：

```
-----  
Reserved   Used      Utilization Ratio  
-----
```


806400	804096	99
1793024	1793024	100
2544640	2518272	98

子指令輸出中的欄有下列涵義：

表 6-2 實體目標磁碟使用度量

度量	說明
Reserved (已保留)	所有記錄使用的磁碟空間 (以位元組為單位)，包括保留使用中訊息的記錄以及等待被重新使用的可用記錄。
Used (已使用)	保留使用中訊息的記錄所使用的磁碟空間 (以位元組為單位)。
Utilization Ratio (使用率)	使用的磁碟空間除以保留的磁碟空間所得之商數。比率越高表示有越多磁碟空間會用來保留使用中訊息。

收回未使用的實體目標磁碟空間

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

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

一般而言，您會穩定保留的磁碟空間並保持高使用率。通常如果系統到達穩定狀態 (即保留的磁碟空間大致固定)，並且使用率高 (高於 75%)，則不需要收回未使用的磁碟空間。如果系統到達穩定狀態且使用率低 (低於 50%)，您可以壓縮磁碟，收回可用記錄佔用的磁碟空間。

`compact dst` 子指令可用來壓縮資料存放區。`compact dst` 子指令的語法如下：

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

此子指令會壓縮指定類型與名稱之實體目標的檔案式資料存放區。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實體目標。

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

▼ 收回未使用的實體目標磁碟空間

1 暫停目標。

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

2 壓縮磁碟。

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

3 重新繼續實體目標。

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

如果未指定目標類型和名稱，則會在**所有**實體目標上執行這些作業。

使用停用的訊息佇列

停用的訊息佇列 `mq.sys.dmq` 是系統建立的實體目標，保留代理程式和其他實體目標之停用的訊息。停用的訊息佇列是用來監視、調校系統效率以及疑難排解的工具。如需「停用的訊息」一詞的定義和停用的訊息佇列之詳細資訊，請參閱「[Message Queue 技術摘要](#)」。

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

配置使用停用的訊息佇列

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

下列範例建立的佇列 `myDist`，依預設使用停用的訊息佇列：

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

下列範例讓相同的佇列停用此停用的訊息佇列：

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

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

管理停用的訊息佇列

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

停用的訊息佇列特性

您可以使用配置其他佇列的方式來配置停用的訊息佇列，但是有些實體目標特性可能不適用或是預設值不同。表 6-3 列出的佇列特性，會由停用的訊息佇列以特殊方式處理。

表 6-3 停用的訊息佇列處理標準實體目標特性的方式

特性	停用的訊息佇列處理方式
limitBehavior	停用的訊息佇列之預設值是 REMOVE_OLDEST。(其他佇列的預設值是 REJECT_NEWEST。)停用的訊息佇列上不支援流量控制。
localDeliveryPreferred	不套用於停用的訊息佇列。
maxNumMsgs	停用的訊息佇列之預設值是 1000。其他佇列的預設值是 -1 (無限制)。
maxNumProducers	不套用於停用的訊息佇列。
maxTotalMsgBytes	停用的訊息佇列之預設值是 10 MB。其他佇列的預設值是 -1 (無限制)。
isLocalOnly	在代理程式叢集中，停用的訊息佇列總是全域實體目標，此特性永遠設定為 false。

訊息內容

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

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

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

如此將會捨棄訊息本文，而只保留標頭與特性資料。

啓用停用的訊息之記錄功能

依預設，不記錄停用的訊息。啓用停用的訊息之記錄功能，可讓代理程式記錄下列事件：

- 代理程式將訊息移至停用的訊息佇列
- 代理程式捨棄來自停用的訊息佇列之訊息，以及不使用停用的訊息佇列之任何實體目標的訊息
- 實體目標達其上限

下列指令會啓用停用的訊息之記錄功能：

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

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

管理安全性

您可透過下列方式管理安全性：配置使用者儲存庫以對使用者進行認證、定義存取控制、配置安全通訊端層 (SSL) 連線服務以對用戶端與代理程式間的通訊進行加密，以及設定用於啟動代理程式的密碼檔案。

本章包含下列各節：

- [第 117 頁的「使用者認證」](#)
- [第 124 頁的「使用者授權：存取控制特性檔案」](#)
- [第 130 頁的「訊息加密」](#)
- [第 138 頁的「密碼檔案」](#)
- [第 140 頁的「稽核記錄」](#)

使用者認證

當使用者嘗試連線至代理程式時，此代理程式將檢查使用者提供的名稱和密碼，以便進行認證。如果名稱和密碼與配置每個代理程式要參考的特定代理程式使用者儲存庫中的名稱和密碼相符，則代理程式承認此連線。

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

儲存庫可能是下列任一類型：

- [Message Queue™ 隨附的平面檔案儲存庫](#)
這種類型的使用者儲存庫使用非常容易。您可以使用使用者管理員公用程式 (`imqusermgr`) 寫入和管理此儲存庫。若要啓用認證，您應在使用者儲存庫中寫入每個使用者的名稱、密碼以及使用者群組名稱。
如需設定與管理使用者儲存庫的更多資訊，請參閱[第 118 頁的「使用平面檔案使用者儲存庫」](#)。
- [LDAP 伺服器](#)

可以是使用 LDAP v2 或 v3 協定之現有或新的 LDAP 目錄伺服器。它不如平面檔案儲存庫易於使用，但是它可延展，因此更適用於生產環境。

如果您要使用 LDAP 使用者儲存庫，可以利用 LDAP 供應商提供的工具來寫入和管理使用者儲存庫。如需更多資訊，請參閱第 122 頁的「將 LDAP 伺服器用於使用者儲存庫」。

使用平面檔案使用者儲存庫

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

建立使用者儲存庫

平面檔案使用者儲存庫是針對實例的。您啟動的每個代理程式實例，都會自動建立預設的使用者儲存庫 (稱為 passwd)。此使用者儲存庫的所在目錄名稱，與此儲存庫相關聯之代理程式實例名稱相同 (請參閱附錄 A)：

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

此儲存庫在建立時就有兩個項目。表 7-1 的每一列表示一個項目。

表 7-1 使用者儲存庫中的初始項目

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

這些初始項目可讓 Message Queue 代理程式在安裝之後立即使用，而無需管理員的介入：

- 初始 guest 使用者項目，可讓用戶端使用預設的 guest 使用者名稱和密碼，連線至代理程式實例。
- 初始 admin 使用者項目可讓您使用 imqcmd 指令和預設的 admin 使用者名稱和密碼來管理代理程式實例。您應該更新此初始項目以變更密碼 (請參閱第 122 頁的「變更預設管理員密碼」)。

以下各節介紹如何寫入和管理平面檔案使用者儲存庫。

使用者管理員公用程式

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

如需 `imqusermgr` 指令的完整參考資訊，請參閱第 13 章。

操作使用者管理員之前，請注意以下事項：

- 如果代理程式專用的使用者儲存庫不存在，您必須啟動對應的代理程式實例來建立儲存庫。
- 必須在安裝代理程式的主機上執行 `imqusermgr` 指令。
- 您必須擁有寫入儲存庫的適當權限：也就是說，在 Solaris 和 Linux 上，您必須是超級使用者或是第一個建立代理程式實例的使用者。

備註 – 以下各節中的範例均假設為預設的代理程式實例。

子指令

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

- `add` 子指令可以將使用者和相關密碼增加到指定的 (或預設的) 代理程式實例儲存庫，並可選擇性指定使用者的群組。子指令語法如下：


```
add [-i instanceName] -u userName -p passwd [-g group] [-s]
```
- `delete` 子指令可以從指定的 (或預設的) 代理程式實例儲存庫刪除指定的使用者。子指令語法如下：


```
delete [-i instanceName] -u userName [-s] [-f]
```
- `list` 子指令可以顯示有關指定的 (或預設的) 代理程式實例儲存庫中，指定的使用者或所有使用者的資訊。子指令語法如下：


```
list [-i instanceName] [-u userName]
```
- `update` 子指令可以更新指定的 (或預設的) 代理程式實例儲存庫中，指定使用者的密碼和 (或) 狀態。子指令語法如下：


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

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

指令選項

表 7-2 列出了 `imqusermgr` 指令的選項。

表 7-2 `imqusermgr` 選項

選項	說明
<code>-a activeState</code>	指定使用者的狀態是否應為使用中 (<code>true/false</code>)。值 <code>true</code> 表示狀態為使用中。這是預設值。
<code>-f</code>	未經使用者確認即執行動作。

表 7-2 imqusermgr 選項 (續)

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。
-i <i>instanceName</i>	指定要套用指令的代理程式實例名稱。如果未指定，則使用預設實例名稱 <code>imqbroker</code> 。
-p <i>passwd</i>	指定使用者密碼。
-g <i>group</i>	指定使用者群組。有效值為 <code>admin</code> 、 <code>user</code> 和 <code>anonymous</code> 。
-s	設定無訊息模式。
-u <i>userName</i>	指定使用者名稱。
-v	顯示版本資訊。不執行指令行上的其他任何指令。

群組

為代理程式實例的使用者儲存庫增加使用者項目時，您可以指定三個預先定義群組中的任一項：`admin`、`user` 或 `anonymous`。如果未指定群組，則預設為指定群組 `user`。請遵照以下說明指定群組：

- **admin 群組**：用於代理程式管理員。依預設，已指定至此群組的使用者可以配置和管理代理程式。您可以將多個使用者指定給 `admin` 群組。
- **user 群組**：用於一般 (非管理) Message Queue 用戶端使用者。大多數用戶端使用者屬於 `user` 群組。依預設，此群組中的使用者可以產生訊息到所有主題與佇列、使用來自所有主題與佇列的訊息，以及瀏覽任何佇列中的訊息。
- **anonymous 群組**：用於不想使用代理程式知道的使用者名稱的 Message Queue 用戶端，可能是因為這些用戶端應用程式不知道應使用的實際使用者名稱。此帳號與大多數 FTP 伺服器中存在的匿名帳號類似。您一次僅可將一個使用者指定給 `anonymous` 群組。與 `user` 群組不一樣的是，您應當限制此群組的存取權限，或者應在部署時移除此群組中的使用者。

若要變更使用者的群組，您必須刪除此使用者項目，然後為此使用者增加其他項目，指定新的群組。

您不能將系統建立的群組重新命名或刪除，也不能建立新群組。但是，您可以指定用來定義群組成員可執行作業的存取規則。如需更多資訊，請參閱第 124 頁的「使用者授權：存取控制特性檔案」。

使用者狀態

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

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


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

使用者名稱與密碼的格式

使用者名稱與密碼必須遵循以下準則：

- 使用者名稱中不能包含星號 (*)、逗號 (,)、冒號 (:)、新行或換行字元。
- 使用者名稱或密碼必須至少包含一個字元。
- 如果使用者名稱或密碼包含空格，則必須將整個名稱或密碼括在引號內。
- 密碼或使用者名稱的長度沒有限制，除非指令 shell 對指令行上可輸入的最大字元數有所限制。

寫入和管理使用者儲存庫

使用 `add` 子指令可增加使用者至儲存庫。例如，以下指令可將使用者 `Katharine` 和密碼 `sesame`，增加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` 的密碼變更爲 `aladdin`：

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

若要列出有關一個或所有使用者的資訊，請使用 `list` 指令。以下指令顯示名為 `isa` 的使用者之相關資訊：

```
imqusermgr list -u isa
```

```
% imqusermgr list -u isa
```

```
User repository for broker instance: imqbroker
```

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

以下指令列出所有使用者的相關資訊：

```
imqusermgr list
```

```
% imqusermgr list

User repository for broker instance: imqbroker
-----
User Name      Group          Active State
-----
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

爲了安全性起見，您應將 `admin` 的預設密碼變更爲只有您自己知道的密碼。以下指令將 `mybroker` 代理程式實例的預設管理員密碼 `admin` 變更爲 `grandpoobah`。

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

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

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

輸入新密碼 (`grandpoobah`) 即生效，輸入舊密碼即失敗。

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

將 LDAP 伺服器用於使用者儲存庫

若要將 LDAP 伺服器用於使用者儲存庫，請執行下列工作：

- 編輯實例配置檔案
- 設定管理員的存取控制

編輯實例配置檔案

若要代理程式使用目錄伺服器者，請在代理程式實例配置檔案 `config.properties` 中設定一些特性值。當使用者嘗試連線至代理程式實例或執行訊息傳送作業時，這些特性可讓代理程式實例查詢 LDAP 伺服器，以取得使用者和群組的相關資訊。

實例配置檔案位於代理程式實例目錄下的目錄中。路徑格式如下：

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

```
/props/config.properties
```

如需作業系統專用的實例目錄位置之詳細資訊，請參閱[附錄 A](#)。

▼ 編輯配置檔案以使用 LDAP 伺服器

- 1 透過設定以下特性，以指定您要使用 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 Base64 編碼 (basic) 還是以 MD5 編碼 (digest)，從用戶端傳送至代理程式。將 LDAP 目錄伺服器用於使用者儲存庫時，您必須將認證類型設定為 basic。例如：

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

- 3 您還必須設定可控制 LDAP 存取的代理程式特性。這些特性儲存在代理程式的實例配置檔案中。[第 77 頁的「安全性服務」](#)中討論這些特性，[第 245 頁的「安全性特性」](#)中也有摘要。

Message Queue 使用 JNDI API 與 LDAP 目錄伺服器通訊。如需這些特性中所參考之語法和專有名詞的更多資訊，請參閱 JNDI 文件。Message Queue 使用 Sun JNDI LDAP 提供者和簡單認證。

Message Queue 支援 LDAP 認證容錯移轉：您可以指定在進行認證時嘗試使用的 LDAP 目錄伺服器之清單 (請參閱 `imq.user.repos.ldap.server` 特性的參考資訊)。

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

- 4 如有必要，您需要編輯存取控制特性檔案中的使用者/群組和規則。如需使用存取控制特性檔案的更多資訊，請參閱[第 124 頁的「使用者授權：存取控制特性檔案」](#)。
- 5 如果在連線認證和群組搜尋期間，您要讓代理程式透過 SSL 與 LDAP 目錄伺服器通訊，則需要啟動 LDAP 伺服器中的 SSL，然後在代理程式配置檔案中設定以下特性：

- 指定 LDAP 伺服器用於 SSL 通訊的連接埠。例如：

```
imq.user_repository.ldap.server=myhost:7878
```

- 將代理程式特性 `imq.user_repository.ldap.ssl.enabled` 設為 true。

部署多台 LDAP 目錄伺服器時，請使用 `ldap://` 指定其他每一台目錄伺服器。例如：

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

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

設定管理員的存取控制

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

任何使用者或群組只要能夠建立 ADMIN 連線，即可下達管理指令。

▼ 設定管理使用者

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

- 2 開啓存取控制檔案 `accesscontrol.properties`。附錄 A 中會列出此檔案的位置。
此檔案包含一個項目，如下所示：

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

列出的項目僅為範例。請注意檔案式使用者儲存庫中有 `admin` 群組，但是依預設，LDAP 目錄中沒有此群組。您必須將 LDAP 目錄中已定義的群組名稱，替換為要授予 Message Queue 管理員權限的群組名稱。

- 3 若要將 Message Queue 管理員權限授予使用者，請如下所示，輸入使用者名稱：
`connection.ADMIN.allow.user= userName[[, userName2] ...]`
- 4 若要將 Message Queue 管理員權限授予群組，請如下所示，輸入群組名稱：
`connection.ADMIN.allow.group= groupName[[, groupName2] ...]`

使用者授權：存取控制特性檔案

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

無論使用者資訊放在平面檔案使用者儲存庫中還是放在 LDAP 使用者儲存庫中，均可使用 ACL 檔案。當用戶端應用程式執行以下作業時，代理程式會檢查其 ACL 檔案：

- 建立連線
- 建立產生器
- 建立用戶
- 瀏覽佇列

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

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

建立存取控制特性檔案

不同實例有不同的 ACL 檔案。每次啟動代理程式實例時，會在實例目錄中建立預設檔案 `accesscontrol.properties`。此檔案的路徑格式如下 (請參閱附錄 A)：

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

ACL 檔案的格式與 Java 特性檔案的格式類似。先定義檔案的版本，然後分三部分指定存取控制規則：

- 連線存取控制
- 實體目標存取控制
- 實體目標自動建立存取控制

`version` 特性定義 ACL 特性檔案的版本；您不能變更此項目。

```
version=JMQFileAccessControlModel/100
```

以下說明存取規則的基本語法並介紹如何計算權限，之後描述可指定存取控制規則的 ACL 檔案三部分。

存取規則的語法

在 ACL 特性檔案中，存取控制會定義特定使用者或群組對受保護的資源 (如實體目標和連線服務) 擁有何種存取權。存取控制由一條或一組規則表示，每條規則皆以 Java 特性表示：

這些規則的基本語法如下：

```
resourceType.resourceVariant
```

```
.operation.access.  
principalType=principals
```

表 7-3 描述了語法規則的元素。

表 7-3 存取規則的語法元素

元素	說明
<code>resourceType</code>	下列其中一項：connection、queue 或 topic。

表 7-3 存取規則的語法元素 (續)

元素	說明
<i>resourceVariant</i>	<i>resourceType</i> 指定的類型實例。例如， <code>myQueue</code> 。您可使用萬用字元符號 (*) 來表示所有連線服務類型或所有實體目標。
<i>operation</i>	值取決於要制定的存取規則種類。
<i>access</i>	下列其中一項： <code>allow</code> 或 <code>deny</code> 。
<i>principalType</i>	下列其中一項： <code>user</code> 或 <code>group</code> 。如需更多資訊，請參閱第 120 頁的「群組」。
<i>principals</i>	可以擁有在規則左側所指定存取權的使用者。如果 <i>principalType</i> 為 <code>user</code> ，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 <code>group</code> ，則此項可以為單一組或群組清單 (以逗號分隔的清單)。您可使用萬用字元符號 (*) 來表示所有使用者或所有群組。

以下為存取規則的一些範例：

- 以下規則表示任何使用者均可以將訊息傳送至名為 `q1` 的佇列。

```
queue.q1.produce.allow.user=*
```

- 以下規則表示任何使用者均可以將訊息傳送至任何佇列。

```
queue.*.produce.allow.user=*
```

備註 – 若要指定非 ASCII 的使用者、群組或目標名稱，請使用 Unicode 換碼 (\uXXXX) 表示法。如果您已經編輯並儲存具有非 ASCII 編碼名稱的 ACL 檔案，則可以使用 `Java native2ascii` 工具將其轉換為 ASCII。如需更多詳細資訊，請至

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

如何計算權限

檔案中有多個存取規則時，權限計算如下：

- 特定存取規則置換一般存取規則。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* 的存取權會置換賦予 * *principal* 的存取權。以下規則拒絕 Bob 產生訊息傳給 `tq1`，但允許其他所有使用者執行這項作業。

```
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。在下列範例中，即使 Bob 是 User 成員，也無法產生訊息傳給 tq1。User 的所有其他成員皆可執行這項作業。

```
queue.tq1.produce.allow.group=User
queue.tq1.produce.deny.user=Bob
```

- 任何未透過存取規則明確授予的存取權限，都會遭到隱式拒絕。例如，如果 ACL 檔案不包含存取規則，則所有使用者都無法執行所有作業。
- 相同使用者或群組的拒絕和允許權限會相互抵銷。例如，以下兩條規則導致 Bob 無法瀏覽 q1：

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

以下兩條規則導致 User 群組無法使用 q5 的訊息。

```
queue.q5.consume.allow.group=User
queue.q5.consume.deny.group=User
```

- 當存在多條相同的左側規則時，只有最後一項生效。

連線服務的存取控制

ACL 特性檔案中的連線存取控制區段，包含用於代理程式連線服務的存取控制規則。連線存取控制規則的語法如下：

```
connection.resourceVariant.
access.principalType=
principals
```

resourceVariant 定義有兩個值：NORMAL 和 ADMIN。您只能授權存取這些連線服務類型的預定值。

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

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

如果您使用檔案式的使用者儲存庫，使用者管理員公用程式會建立預設群組 admin。如果您使用 LDAP 使用者儲存庫，只要執行下列一項作業即可使用預設 ACL 特性檔案：

- 在 LDAP 目錄中定義 `admin` 群組。
- 將 ACL 特性檔案中的 `admin` 名稱，換成 LDAP 目錄中已定義的一個或多個群組名稱。

您可以限制連線存取權限。例如，以下規則拒絕 Bob 存取 `NORMAL`，但是允許其他所有使用者進行存取：

```
connection.NORMAL.deny.user=Bob
connection.NORMAL.allow.user=*
```

您可以使用星號 (*) 字元代表所有認證使用者或群組。

對於檔案式使用者儲存庫和 LDAP 使用者儲存庫，使用 ACL 特性檔案授權存取 `ADMIN` 連線的方式有下列不同：

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

實體目標的存取控制

存取控制特性檔案的目標存取控制區段，包含實體目標型的存取控制規則。這些規則決定何人(使用者/群組)可以在何處(實體目標)做什麼(作業)。這些規則所規定的存取權類型包括：將訊息傳送至佇列、將訊息發佈至主題、接收來自佇列的訊息、訂閱主題以及瀏覽佇列中的訊息。

依預設，任何使用者或群組對任何實體目標都擁有所有類型的存取權。您可以新增更多特定的目標存取規則，或編輯預設規則。本節的其餘部分介紹實體目標存取規則的語法，您必須理解此語法以編寫自己的規則。

目標規則的語法如下：

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

表 7-4 中有這些元素的描述：

表 7-4 實體目標存取控制規則的元素

元件	說明
<i>resourceType</i>	可以是 <code>queue</code> 或 <code>topic</code> 。
<i>resourceVariant</i>	某一個實體目標名稱；或所有的實體目標 (*)，表示所有佇列或所有主題。
<i>operation</i>	可以是 <code>produce</code> 、 <code>consume</code> 或 <code>browse</code> 。
<i>access</i>	可以是 <code>allow</code> 或 <code>deny</code> 。
<i>principalType</i>	可以是 <code>user</code> 或 <code>group</code> 。

可以將存取權賦予一個或多個使用者和 (或) 一個或多個群組。

以下範例說明了不同種類的實體目標存取控制規則：

- 允許所有使用者將訊息傳送至任何佇列目標。

```
queue.*.produce.allow.user=*
```

- 拒絕群組 `user` 的任何成員訂閱主題 `Admissions`。

```
topic.Admissions.consume.deny.group=user
```

自動建立實體目標的存取控制

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

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

依預設，任何使用者或群組均有權讓代理程式自動建立實體目標。此權限由以下規則指定：

```
queue.create.allow.user=*
```

```
topic.create.allow.user=*
```

您可以編輯 ACL 檔案以限制此類型的存取權。

實體目標自動建立存取規則的一般語法如下：

```
resourceType.create.access.principalType=principals
```

其中 *resourceType* 為 `queue` 或 `topic`。

例如，以下規則可讓代理程式為所有使用者 (除 Snoopy 之外) 自動建立主題目標。

```
topic.create.allow.user=*  
topic.create.deny.user=Snoopy
```

請注意，實體目標自動建立規則的效果必須與實體目標存取規則的效果一致。例如，如果您 1) 變更目標存取規則，以禁止任何使用者將訊息傳送至目標，但是 2) 啟用目標的自動建立功能，如果此目標不存在，則代理程式**將會**建立此實體目標，但**不會**將訊息傳送至此目標。

訊息加密

本節說明如何設定以安全通訊端層 (SSL) 標準為基礎的連線服務，在用戶端和代理程式之間傳送加密訊息。Message Queue 支援下列 SSL 型連線服務：

- `ssljms` 服務使用 TCP/IP 傳輸協定，在用戶端和代理程式之間傳送安全、加密的訊息。
- `ssladmin` 服務使用 TCP/IP 傳輸協定，在 Message Queue 指令公用程式 (`imqcmd`) 和代理程式之間建立安全、加密的連線。管理主控台 (`imqadmin`) 不支援加密的連線。
- `cluster` 服務使用 TCP/IP 傳輸協定，為叢集中的代理程式之間提供安全、加密的通訊。
- `httpsjms` 服務使用 HTTPS 通道 Servlet 和 HTTP 傳輸協定，在用戶端和代理程式之間傳送安全、加密的訊息。

本節後續部分說明如何使用 `ssljms`、`ssladmin` 和 `cluster` 連線服務，透過 TCP/IP 設定安全連線。如需有關使用 `httpsjms` 服務，透過 HTTP 設定安全連線的詳細資訊，請參閱[附錄 C](#)。

使用自我簽署的憑證

若要透過 TCP/IP 使用 SSL 型連線服務，您必須使用金鑰工具公用程式 (`imqkeytool`) 產生公開/私密金鑰對。此公用程式將公開金鑰內嵌於自我簽署的憑證中，此憑證將傳送至請求與代理程式連線的用戶端，此用戶端使用此憑證設定加密連線。本節描述如何使用此類自我簽署的憑證，設定 SSL 型服務。

若要提高認證層級，您可以使用憑證授權單位驗證的簽署憑證。相較於自我簽署的憑證，使用簽署憑證需要更多必要步驟。您必須先執行本節所述的步驟，接著執行[第 135 頁的「使用簽署的憑證」](#)中的額外步驟。

Message Queue 支援具有自我簽署憑證的 SSL，會認為用戶端正與已知並可信任的伺服器通訊，而保護所傳送的資料。下列程序說明使用自我簽署的憑證設定 SSL 型連線服務的必要步驟。後續小節會詳細說明每個步驟。

▼ 使用自我簽署的憑證設定 SSL 型連線服務

- 1 產生自我簽署的憑證。
- 2 啓用代理程式中的 `ssljms`、`ssladmin` 或 `cluster` 連線服務。
- 3 啓動代理程式。
- 4 配置並執行用戶端。

這個步驟僅適用於 `ssljms` 連線服務，不適用於 `ssladmin` 或 `cluster`。

產生自我簽署的憑證

執行金鑰工具公用程式 (`imkeytool`)，為代理程式產生自我簽署的憑證。(在 UNIX® 系統上，需要以超級使用者 (`root`) 的身份執行此公用程式，才擁有建立金鑰庫的權限。)相同的憑證可用於 `ssljms`、`ssladmin` 或 `cluster` 連線服務。

在指令提示符號下輸入以下內容：

```
imkeytool -broker
```

金鑰工具公用程式會提示您輸入金鑰庫密碼。

```
Generating keystore for the broker ...
Enter keystore password:
```

接著，公用程式會提示您輸入憑證所屬代理程式的識別資訊。您提供的資訊會組成 X.500 辨別名稱。表 7-5 顯示提示以及提供給每個提示的值。這些值不區分大小寫，而且可以包含空格。

表 7-5 自我簽署憑證所需之識別名稱資訊

提示	X.500 屬性	說明	範例
What is your first and last name?	commonName (CN)	執行代理程式的伺服器之完全合格的名稱	mqserver.sun.com
What is the name of your organizational unit?	organizationalUnit (OU)	部門名稱	purchasing
What is the name of your organization?	organizationName (ON)	較大的組織名稱，例如公司或政府實體。	My Company, Inc.
What is the name of your city or locality?	localityName (L)	城市或所在地名稱	San Francisco

表 7-5 自我簽署憑證所需之識別名稱資訊 (續)

提示	X.500 屬性	說明	範例
What is the name of your state or province?	stateName (ST)	州或省的全名 (不使用縮寫)	California
What is the two-letter country code for this unit?	country (C)	標準的雙字母國家代碼	US

輸入完畢後，金鑰工具公用程式會顯示您輸入的資訊，請您確認。例如：

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

若要接受目前的值並繼續執行，請輸入 **yes**；若要重新輸入這些值，請接受預設值或輸入 **no**。您確認後，此公用程式會產生金鑰對並暫停其他作業。

接著，公用程式會要求您提供用於鎖定金鑰對的密碼 (即金鑰密碼)。看到此提示請按下 **Return** 鍵，表示金鑰密碼與金鑰庫密碼相同。

備註 – 請記住您指定的密碼。啟動代理程式時必須提供此密碼，代理程式才能開啓金鑰庫。您可以在密碼檔案中儲存金鑰庫密碼 (請參閱第 138 頁的「密碼檔案」)。

金鑰工具公用程式會產生自我簽署的憑證，並將其放置在 Message Queue 的金鑰庫中。金鑰庫所在目錄取決於作業系統，請參閱附錄 A。

以下是 SSL 型連線服務之 Message Queue 金鑰庫的可配置特性：

- `imq.keystore.file.dirpath`：包含金鑰庫檔案之目錄的路徑。如需預設值，請參閱附錄 A。
- `imq.keystore.file.name`：金鑰庫檔案的名稱。
- `imq.keystore.password`：金鑰庫密碼。

有時您可能需要重新產生金鑰對，以解決某些問題：例如，您忘記了金鑰庫密碼，或者啓動代理程式時，SSL 型服務無法初始化，並且您遇到以下異常：

```
java.security.UnrecoverableKeyException:Cannot recover key
```

(如果在產生自我簽署的憑證時，所提供的金鑰密碼與金鑰庫密碼不同，則可能導致此異常。)

▼ 重新產生金鑰對

- 1 移除代理程式的金鑰庫，其位置如附錄 A 中所示。
- 2 重新執行 `imqkeytool` 以產生新的金鑰對，如上所述。

啓用 SSL 型連線服務

若要在代理程式中啓用 SSL 型連線服務，您需要將 `ssljms` (或 `ssladmin`) 增加到 `imq.service.activelist` 特性。

▼ 在代理程式中啓用 SSL 型服務

- 1 開啓代理程式的實例配置檔案。

實例配置檔案的所在目錄名稱，與此配置檔案相關聯之代理程式實例名稱 (`instanceName`) 相同 (請參閱附錄 A)：

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

- 2 增加表示 `imq.service.activelist` 特性的項目 (如果項目尚未存在)，並將所需的 SSL 型服務納入清單。

依預設，此特性包括 `jms` 和 `admin` 連線服務。增加 SSL 型服務或您要啓動的服務 (`ssljms`、`ssladmin`，或兩者)：

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

備註 – 請使用 `imq.cluster.transport` 特性啓用 SSL 型 `cluster` 連線服務，而非使用 `imq.service.activelist` 特性；請參閱第 159 頁的「連線代理程式」。

- 3 儲存並關閉實例配置檔案。

啓動代理程式

啓動代理程式，提供金鑰庫密碼。您可以透過以下任何一種方法提供密碼：

- 允許代理程式在啓動時提示您提供密碼：

```
imqbrokerd
Please enter Keystore password:
```

- 將密碼放在密碼檔案中，如第 138 頁的「密碼檔案」中所述。然後設定特性 `imq.passfile.enabled=true`，並執行下列其中一項作業：

- 將密碼檔案的位置傳送給 `imqbrokerd` 指令：

```
imqbrokerd -passfile /passfileDirectory/passfileName
```

- 啓動代理程式時，不使用 `--passfile` 選項，而是透過以下兩個代理程式配置特性來指定密碼檔案的位置：

```
imq.passfile.dirpath=/passfileDirectory
```

```
imq.passfile.name=passfileName
```

備註 – 啓動使用 SSL 的代理程式或用戶端時，CPU 使用率可能在幾秒鐘內劇增。這是因爲 Message Queue 使用 JSSE (Java Secure Socket Extension) 方法 `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。建立種子之後，CPU 使用率會降至正常。

配置並執行 SSL 型用戶端

將用戶端配置爲使用 SSL 型連線服務的程式，根據其爲應用程式用戶端 (使用 `ssljms` 連線服務) 還是 Message Queue 管理用戶端 (例如使用 `ssladmin` 連線服務的 `imqcmd`) 而有所不同。

應用程式用戶端

針對應用程式用戶端，您必須確保用戶端在其 `CLASSPATH` 變數中已指定下列 `.jar` 檔案：

```
imq.jar
jms.jar
```

如果您使用 1.4 版之前的 Java 2 Software Development Kit (J2SDK)，也必須包含下列 Java Secure Socket Extension (JSSE) 和 Java Naming and Directory Interface (JNDI) `.jar` 檔案：

```
jsse.jar
jnet.jar
jcert.jar
jndi.jar
```

(如果您使用 J2SDK 1.4 或更新版本，因爲已內建 JSSE 和 JNDI 支援，則不需要包含這些檔案。)

正確指定 `CLASSPATH` 檔案後，請啓動用戶端並將其連線至代理程式的 `ssljms` 連線服務，完成此作業的方法之一是輸入以下指令：

```
java -DimqConnectionType=TLS clientAppName
```

如此可告知連線使用 SSL 型連線服務。

管理用戶端

針對管理用戶端，您可以透過在呼叫 `imqcmd` 指令時加入 `-secure` 選項，來建立安全連線：例如：

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

其中 `adminName` 是 Message Queue 使用者儲存庫中的有效項目。此指令會提示您輸入密碼。(如果您使用平面檔案儲存庫，請參閱第 122 頁的「變更預設管理員密碼」)。

列出連線服務是驗證 `ssladmin` 服務正在執行的一種方法，也可以驗證是否能成功建立安全的管理連線，如下所示：

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.

使用簽署的憑證

簽署憑證提供的伺服器認證，比自我簽署憑證的伺服器認證更安全。您只能在用戶端和代理程式之間 (不能在叢集的多個代理程式之間) 實作簽署憑證。除了上述配置自我簽署憑證所需步驟之外，您還需要執行下列額外的步驟。後續小節會詳細說明這些步驟。

▼ 使用簽署的憑證

- 1 在金鑰庫中安裝憑證。
- 2 配置 Message Queue 用戶端在建立與代理程式的 SSL 型連線時，請求簽署憑證。

取得並安裝簽署的憑證

下列程序說明如何取得並安裝簽署的憑證。

▼ 取得簽署的憑證

- 1 使用 `J2SE keytool` 指令，為您在上節中產生的自我簽署憑證，產生憑證簽署請求 (CSR)。

下列網站提供有關 `keytool` 指令的資訊：

<http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html>

範例：

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

這會產生 CSR，將憑證封裝在指定的檔案中 (此範例中的 `certreq.csr`)。

2 使用 CSR 產生或請求簽署的憑證。

您可以使用下列方法之一：

- 請 Thawte 或 Verisign 等知名的憑證授權單位 (CA) 簽署憑證。如需如何執行此作業的詳細資訊，請參閱您的 CA 文件。
- 使用 SSL 簽署套裝軟體，自行簽署憑證。
簽出的憑證是一串 ASCII 字元。如果您收到 CA 簽署的憑證，它可能是一個電子郵件附件或是郵件中的文字。

3 將簽署的憑證儲存在檔案中。

以下說明中，使用 `broker.cer` 範例名稱來代表代理程式憑證。

▼ 安裝簽署的憑證

1 檢查 J2SE 依預設是否支援您的憑證授權單位。

以下指令會列出系統金鑰庫中的根 CA：

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

如果有列出您的 CA，請跳過下一個步驟。

2 如果 J2SE 未支援您的憑證授權單位，請將 CA 的根憑證匯入 Message Queue 金鑰庫中。

範例：

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

其中 `ca.cer` 檔案包含 CA 提供的根憑證。

如果您使用 CA 測試憑證，可能需要匯入測試 CA 根憑證。您的 CA 應該會指示如何取得該憑證的副本。

3 將簽署的憑證匯入金鑰庫中，替代原始的自我簽署憑證。

範例：

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

`broker.cer` 是您從 CA 獲得的簽署憑證所在的檔案。

現在 Message Queue 金鑰庫包含簽署憑證，可用於 SSL 連線。

配置 Message Queue 用戶端執行階段以請求簽署憑證

現在您必須配置 Message Queue 用戶端執行階段以請求簽署的憑證，並確保用戶端信任簽署此憑證的憑證授權單位。

▼ 配置用戶端執行階段以請求簽署的憑證

- 1 將連線工廠的 `imqSSLIsHostTrusted` 屬性設定為 `false`。

依預設，用戶端將用來建立代理程式連線的連線工廠物件之 `imqSSLIsHostTrusted` 屬性設定為 `true`，這表示用戶端執行階段會接受所出示的任何憑證。您必須將此值變更為 `false`，讓用戶端執行階段嘗試驗證所出示的所有憑證。如果憑證簽署者未出現在用戶端的信任清單存放區，驗證便會失敗。

- 2 驗證簽署的授權單位是否已在用戶端的信任清單存放區中註冊。

若要測試用戶端是否會接受您的憑證授權單位所簽署的憑證，請嘗試建立 SSL 連線，如上節第 134 頁的「配置並執行 SSL 型用戶端」中所述。如果 CA 列在用戶端的信任存放區中，連線便會成功，您可以跳過下一個步驟。如果連線失敗出現憑證驗證錯誤，請執行下一個步驟。

- 3 在用戶端的信任存放區中安裝簽署 CA 的根憑證。

依預設，用戶端會搜尋金鑰庫檔案 `cacerts` 和 `jssecacerts`，因此如果憑證安裝在以上其中一個檔案中，便不需要進一步配置。下列範例會將來自 Verisign 憑證授權單位，檔名為 `testrootca.cer` 的測試根憑證檔案，安裝到預設系統憑證檔案 `cacerts`。此範例假設 J2SE 安裝在 `$JAVA_HOME/usr/j2se` 目錄中：

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

建議使用的替代選項，是將根憑證安裝到替代系統憑證檔案 `jssecacerts`：

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

第三個可能方法是將根憑證安裝到其他金鑰庫檔案中，並且配置用戶端將其視為信任存放區。下列範例會安裝到 `/home/smith/.keystore` 檔案中：

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

由於用戶端依預設不會搜尋此金鑰庫，因此您必須明確向用戶端提供此位置作為信任存放區。若要執行這項操作，請在用戶端執行時，設定 Java 系統特性 `javax.net.ssl.trustStore`：

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

密碼檔案

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

表 7-6 使用密碼的指令

指令	用途	密碼的用途
imqbrokerd	啟動代理程式	存取 JDBC 型永久性資料存放區、SSL 憑證金鑰庫或 LDAP 使用者儲存庫
imqcmd	管理代理程式	認證已授權使用指令之管理使用者
imqdbmgr	管理 JDBC 型資料存放區	存取資料存放區

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

```
imqbrokerd -passfile myPassfile
```

備註 – 您可以在舊版的指令行中使用 `-p`、`-password`、`-dbpassword` 和 `-ldappassword` 選項來指定密碼。但是這些選項已經被停用，將從未來的版本中移除。在目前的版本中，指令行中這些選項的任一值，會取代密碼檔案中相關的值。

安全考量

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

密碼檔案沒有加密，您必須設定權限以防止無權限者存取此檔案。設定權限來限制哪些使用者檢視檔案，但是允許啟動代理程式的使用者讀取此檔案。

密碼檔案內容

密碼檔案是純文字檔，包含一組特性和值。每一個值是指令使用的密碼。

密碼檔案可以包含表 7-7 中所示密碼：

表 7-7 密碼檔案中的密碼

密碼	影響的指令	說明
imq.imqcmd.password	+imqcmd	指定 imqcmd 指令行的管理員密碼。每個指令都會認證此密碼。
imq.keystore.password	imqbrokerd	指定 SSL 型服務的金鑰庫密碼。
imq.persist.jdbc.password	imqbrokerd imdbmgr	必要時，請指定用於開啓資料庫連線的密碼。
imq.user_repository.ldap.password	imqbrokerd	指定與辨別名稱相關聯的密碼，此辨別名稱指定給代理程式以連結至已配置的 LDAP 使用者儲存庫。

Message Queue 產品提供密碼檔案範例。有關此範例檔案的位置，請參閱[附錄 A](#)。

透過防火牆連線

當用戶端應用程式和代理程式之間被防火牆分隔時，需要採取特殊措施來建立連線。一個方法是使用 `httpjms` 或 `httpsjms` 連線服務，在防火牆中「打出通道」；如需詳細資訊，請參閱[附錄 C](#)。不過 HTTP 連線比起其他連線服務較慢；較快速的替代方式是略過 Message Queue 連接埠對映器，並明確指定靜態連接埠位址給所需的連線服務，然後在防火牆開啓此特定的連接埠。這個方法可以使用 `jms` 或 `ssljms` 連線服務 (或在不尋常的狀況下，使用 `admin` 或 `ssladmin`)，透過防火牆連線。

表 7-8 靜態連接埠位址的代理程式配置特性

連線服務	配置特性
jms	imq.jms.tcp.port
ssljms	imq.ssljms.tls.port
admin	imq.admin.tcp.port
ssladmin	imq.ssladmin.tls.port

▼ 透過防火牆啓用代理程式連線

- 1 將靜態連接埠位址指派給要使用的連線服務。

若要略過連接埠對映器，並將靜態連接埠號直接指定給連線服務，請設定代理程式配置特性 `imq.serviceName.protocolType.port`，其中 `serviceName` 為連線服務名稱，`protocolType` 為協定類型 (請參閱[表 7-8](#))。如同所有代理程式配置特性，您可以在代理

程式的實例配置檔案中指定此特性，或在啓動代理程式時從命令行指定此特性。例如，若要將連接埠號 10234 指派給 jms 連線服務，請在配置檔案中併入下列命令行

```
imq.jms.tcp.port=10234
```

或使用下列指令啓動代理程式

```
imqbrokerd -name myBroker -Dimq.jms.tcp.port=10234
```

2 配置防火牆以允許連線至指派給連線服務的連接埠號。

您也必須允許透過防火牆連線至 Message Queue 的連接埠對映器連接埠 (通常是 7676，除非已重新指定連接埠對映器使用其他連接埠)。例如，在以上範例中，您必須針對連接埠 10234 和 7676 開啓防火牆。

稽核記錄

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

- 啓動、關機、重新啓動及移除代理程式實例
- 使用者認證和授權
- 重設永久性存放區
- 建立、清除及銷毀實體目標
- 管理長期訂閱者銷毀

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

管理受管理物件

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

本章說明如何使用物件管理員公用程式 (`imqobjmgr`) 來建立和管理受管理物件。它包含下面各節：

- 第 141 頁的「物件存放區」
- 第 143 頁的「受管理物件屬性」
- 第 149 頁的「使用物件管理員公用程式」
- 第 150 頁的「新增受管理物件」
- 第 152 頁的「列出受管理物件」
- 第 153 頁的「檢視受管理物件資訊」
- 第 153 頁的「修改受管理物件屬性」

物件存放區

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

LDAP 伺服器物件存放區

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

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

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

- **初始環境。** `java.naming.factory.initial` 屬性會指定伺服器上 JNDI 查找的初始環境。各 LDAP 物件存放區有其固定的此屬性值。
- **位置。** `java.naming.provider.url` 屬性會指定 LDAP 伺服器的 URL 和目錄路徑。您必須確認所指定的目錄路徑存在。
- **安全性。** `java.naming.security.principal`、`java.naming.security.credentials` 和 `java.naming.security.authentication` 屬性會在呼叫者嘗試存取物件存放區時，管理該呼叫者的認證程序。這些屬性的實際格式和數值視 LDAP 服務提供者而定；請參閱 LDAP 實作所隨附的文件以取得詳細資訊，並判斷所有作業是否均需要有安全性資訊，或者只有變更已儲存資料的作業需要安全性資訊。

表 8-1 LDAP 物件存放區屬性

屬性	說明
<code>java.naming.factory.initial</code>	JNDI 查找的初始環境 範例： <code>com.sun.jndi.ldap.LdapCtxFactory</code>
<code>java.naming.provider.url</code>	伺服器 URL 和目錄路徑 範例： <code>ldap://myD.com:389/ou=mq1,o=App</code> 其中受管理物件存放區儲存在 <code>/App/mq1</code> 目錄中。
<code>java.naming.security.principal</code>	認證呼叫者的主體驗證身份 此屬性的格式取決於認證方案：例如： <code>uid=homerSimpson,ou=People,o=mq</code> 如果沒有指定此屬性，則運作方式將由 LDAP 服務提供者決定。
<code>java.naming.security.credentials</code>	認證主體的憑證 此屬性的值取決於認證方案：例如，它可以是雜湊密碼、明文密碼、金鑰或憑證。 如果沒有指定此特性，則運作方式將由 LDAP 服務提供者決定。
<code>java.naming.security.authentication</code>	認證的安全層級 此屬性的值是下列關鍵字之一： <code>none</code> 、 <code>simple</code> 或 <code>strong</code> 。例如，如果您指定 <code>simple</code> ，則會提示您所有遺漏的主體或憑證值。這可讓您以更安全的方式提供識別資訊。 如果沒有指定此特性，則運作方式將由 LDAP 服務提供者決定。

檔案系統物件存放區

Message Queue 也支援在本機檔案系統中使用目錄，以作為受管理物件的物件存放區。我們不建議您對生產系統使用此方法，不過此方法具有在開發環境中非常容易使用的優點。但是請注意，如果跨多個電腦節點部署的用戶端將目錄用作集中物件存放區，則這些用戶端都必須擁有該目錄的存取權。除此之外，任何對此目錄擁有存取權的使用者，均可使用 Message Queue 管理工具以建立並管理受管理物件。

使用檔案系統目錄作為物件存放區時，您必須指定表 8-2 中所顯示的屬性。這些屬性大致和上述 LDAP 物件存放區相同，其中特別需要說明的是，`java.naming.provider.url` 屬性會指定保留物件存放區之目錄的路徑。該目錄必須存在，而且 Message Queue 管理工具的使用者與用戶端應用程式的使用者皆必須擁有適當的存取權限，以供存取存放區。

表 8-2 檔案系統物件存放區屬性

屬性	說明
<code>java.naming.factory.initial</code>	JNDI 查找的初始環境 範例： <code>com.sun.jndi.fscontext.RefFSContextFactory</code>
<code>java.naming.provider.url</code>	目錄路徑 範例： <code>file:///C:/myapp/mqobjjs</code>

受管理物件屬性

Message Queue 受管理物件有兩種基本類型：

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

備註 - 特殊 SOAP 端點受管理物件使用於 SOAP 訊息傳送；如需更多資訊，請參閱「Message Queue Developer's Guide for Java Clients」。

每種受管理物件類型均包含決定物件特性及運作方式的屬性。本節描述如何使用物件管理員指令行公用程式 (`imqobjmgr`) 來設定這些屬性，您也可以使用 GUI 管理主控台進行設定，如第 53 頁的「使用受管理物件」所述。

連線工廠屬性

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

Message Queue 定義兩種類別的連線工廠物件：

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

這兩種類別共用相同的配置屬性，您可以使用這些屬性最佳化資源、效能和訊息流量。第 16 章中列出這些屬性並有詳細說明，下列各節將討論這些屬性：

- 第 144 頁的「[連線處理](#)」
- 第 146 頁的「[用戶端標識](#)」
- 第 147 頁的「[可靠性與流量控制](#)」
- 第 148 頁的「[佇列瀏覽器和伺服器階段作業](#)」
- 第 148 頁的「[標準訊息特性](#)」
- 第 148 頁的「[訊息標頭置換](#)」

連線處理

連線處理屬性會指定要連線至哪個代理程式位址，並會視需要指定偵測連線失敗和嘗試重新連線的方式。在表 16-1 有這些屬性的摘要說明。

代理程式位址清單

最重要的連線處理屬性是 `imqAddressList`，該屬性會指定要與其建立連線的代理程式。此屬性值是一個字串，其中包含一個代理程式位址或 (如果是代理程式叢集) 由逗號分隔的多個位址。視所使用的連線服務 (請參閱第 71 頁的「[連線服務](#)」) 和建立連線的方法之不同，代理程式位址可使用多種定址方案：

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

這些定址方案在表 16-2 中有所摘要。

每個代理程式位址的一般格式為

```
scheme://address
```


其中 *scheme* 是上述所列的其中一個定址方案，而 *address* 則表示代理程式位址本身。指定位址的確切語法會因定址機制而異，如表 16-2 中的最後一欄所示。表 16-3 顯示不同位址格式範例。

在有多個代理程式的叢集環境中，位址清單可包含多個代理程式位址。如果第一個連線嘗試失敗，Message Queue 用戶端執行階段會嘗試連線至清單中的另一個位址，以此類推，一直到清單上所有位址都試完為止。有兩個額外的連線工廠屬性會控制完成此作業的方式：

- `imqAddressListBehavior` 會指定特定位址的嘗試順序。如果此屬性設定為 `PRIORITY` 字串，將會以在位址清單中出現的順序來嘗試連線位址。如果屬性值是 `RANDOM`，則會以隨機的順序來嘗試連線位址；例如，當有許多 Message Queue 用戶端共用相同的連線工廠物件時，這會有效避免用戶端全部嘗試連線至相同的代理程式位址。
- `imqAddressListIterations` 會指定在放棄並報告失敗之前，循環清單的次數。數值 `-1` 表示不限制循環次數：用戶端執行階段會繼續嘗試，直到成功建立連線或時間終了為止（視哪種情況先發生）。

自動重新連線

您可以將連線工廠的 `imqReconnectEnabled` 屬性設定為 `true`，讓用戶端在連線失敗時，能自動重新連線至代理程式。`imqReconnectAttempts` 屬性會控制重新連線至指定代理程式位址的嘗試次數；`imqReconnectInterval` 會指定兩次連線嘗試之間的等待時間間隔（以毫秒為單位）。

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

自動重新連線支援訊息使用的所有用戶端確認模式。連線一旦重新建立，代理程式就會重新傳送之前已傳送但尚未確認的所有訊息，並使用 `Redeiver` 旗標加以標記。應用程式碼可使用此旗標來判斷是否有已使用但尚未確認的訊息。（但是，如果不是長期訂閱者，一旦關閉連線，代理程式將不會保留訊息。因此當連線關閉時，為這類訂閱者所產生的任何訊息將無法在重新連線後傳送，而會遺失。）進行自動重新連線時會阻斷訊息的產生，因此訊息產生器在連線尚未重新建立之前，無法傳送訊息到代理程式。

自動重新連線可提供連線容錯移轉功能，但不提供資料容錯移轉功能：當用戶端重新連線至不同的代理程式實例時，失敗或連線中斷的代理程式所保留的永久性訊息和其他狀態資訊將會遺失。嘗試重新建立連線時，Message Queue 會維護用戶端執行階段所提供的物件（例如階段作業、訊息用戶和訊息產生器）。當連線失敗時，也會保留暫時目標一段時間，因為用戶端可能會重新連線並再次存取這些目標；在用戶端利用這段時間重新連線並使用這些目標之後，代理程式就會刪除這些目標。如果重新連線後，無法在代理程式上完全復原用戶端狀態（例如，當使用只有在連線期間才存在的已處理

之階段作業時)，將不會進行自動重新連線，而是會呼叫連線的異常處理程式。接著會由應用程式碼擷取異常、重新連線及復原狀態。

定期測試 (Ping) 連線

Message Queue 用戶端執行階段可以配置成定期測試或「Ping」連線，在嘗試的訊息傳輸失敗前，就事先偵測到連線失敗。如果用戶端應用程式只是使用訊息而並不產生這些訊息，則此項測試特別重要，因為這類應用程式無法以其他方式偵測到連線失敗。很少產生訊息的用戶端也可受惠於此功能。

連線工廠屬性 `imqPingInterval` 會指定使用 Ping 指令偵測連線的頻率 (以秒為單位)。依預設，時間間隔設定為 30 秒，數值 -1 表示停用 Ping 作業。

Ping 失敗的回應會根據不同的作業系統平台而有所差異。在某些系統上，會立即對用戶端應用程式的異常偵聽程式丟出一個異常。(如果用戶端沒有異常偵聽程式，則該用戶端下次嘗試使用連線時將會失敗。)其他的系統可能會繼續嘗試建立與代理程式的連線，緩衝連續的偵測，直到偵測成功或緩衝區溢位為止。

用戶端標識

表 16-4 中所列的連線工廠屬性，支援用戶端認證以及長期訂閱者的用戶端識別碼設定。

用戶端認證

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

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

用戶端識別碼

每當代理程式必須為用戶端維護持續性狀態時，Java 訊息服務規格會要求連線提供唯一的用戶端識別碼。Message Queue 使用這類用戶端識別碼，追蹤主題目標的長期訂閱者。如果長期訂閱者變為非使用中的狀態，則代理程式會保留該主題的所有內送訊息，等到此訂閱者回復為使用中狀態時，再傳送這些訊息。代理程式會藉由用戶端識別碼來識別用戶。

用戶端應用程式可能會使用連線物件的 `setClientID` 方法，透過程式設計來設定自己的用戶端識別碼，因此很難協調用戶端識別碼並確保每個識別碼都是唯一的。通常比較

好的方式是讓 Message Queue 在為用戶端建立連線時，自動指定唯一的識別碼。若要這麼做，請將連線工廠的 `imqConfiguredClientID` 屬性設定為下列格式的值

```
${u}factoryID
```

`${u}` 字元必須為屬性值的前四個字元。(在括弧之間若有 `u` 以外的其他字元，會造成在建立連線時丟出一個異常；如果是在其他位置，則這些字元並不具有任何特殊意義，而會被當作是純文字來處理。) `factoryID` 的值是唯一與此連線工廠物件關聯的字元字串。

為特定用戶端建立連線時，Message Queue 會以 `u:userName` 取代 `${u}` 字元，以建構用戶端識別碼，其中 `userName` 是供連線認證的使用者名稱。這可確保指定連線工廠所建立的每個連線即便在其他方面都相同，也會有唯一的用戶端識別碼。例如，如果使用者名稱是 Calvin，而為連線工廠的 `imqConfiguredClientID` 屬性指定的字串是 `${u}Hobbes`，則指定的用戶端識別碼會是 `u:CalvinHobbes`。

備註 - 如果有兩個用戶端都使用預設的使用者名稱 `guest` 以嘗試取得連線，則無法使用此方案，因為對於這兩個用戶端的用戶端識別碼，其 `${u}` 部分會完全相同。在這種情況下，只有第一個請求連線的用戶端才能取得連線，第二個嘗試連線的用戶端則會失敗，因為 Message Queue 無法使用同一用戶端識別碼建立兩個連線。

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

可靠性與流量控制

由於用戶端傳送與接收的「有效負載」訊息以及 Message Queue 本身使用的控制訊息(例如代理程式確認)透過相同的用戶端與代理程式連線傳送，因此過量的有效負載流量可能會干擾控制訊息的傳送。若要協助減輕此問題，表 16-5 中所列的連線工廠屬性可讓您管理這兩類訊息的相對流量。這些屬性可分為四種：

- **確認逾時**：指定在丟出異常之前，等待代理程式確認的最長時間 (`imqAckTimeout`)。
- **連線流量計數**：會限制將有效負載訊息以指定的大小分批次進行傳輸 (`imqConnectionFlowCount`)，以確保可定期傳送任何累積的控制訊息。
- **連線流量控制**：會限制連線上可以保留之擱置的有效負載訊息數 (`imqConnectionFlowLimit`)。達到限制時，傳送到連線的有效負載訊息會暫停，直到等待使用的訊息數量降到限制以下為止。此功能的使用是由布林值旗標 (`imqConnectionFlowLimitEnabled`) 所控制。

- **用戶流量控制**：會限制任何單一用戶可以擱置等待使用的有效負載訊息數 (`imqConsumerFlowLimit`)。(此限制也可以指定為特定佇列目標的特性 `consumerFlowLimit`。)達到限制時，會暫停傳送有效負載訊息至用戶，直到等待使用的訊息數量 (`imqConsumerFlowLimit` 以百分比表示) 降到 `imqConsumerFlowThreshold` 屬性所指定的限制以下為止。這可以防止任何用戶佔用連線讓別人苦等，幫助改善多名用戶之間的負載平衡。

使用這些流量控制技術必須在可靠性與流量之間進行權衡；如需詳細說明，請參閱第 192 頁的「用戶端執行階段訊息流量調校」。

佇列瀏覽器和伺服器階段作業

表 16-6 會列出影響用戶端佇列瀏覽與伺服器階段作業的連線工廠屬性。`imqQueueBrowserMaxMessagesPerRetrieve` 屬性會指定瀏覽佇列目標內容時，一次擷取的最大訊息數；`imqQueueBrowserRetrieveTimeout` 會指定擷取訊息的最長等待時間。(請注意，`imqQueueBrowserMaxMessagesPerRetrieve` 不會影響所瀏覽的訊息總數，而只是影響將訊息分為資料塊以便傳送至用戶端執行階段的方式：分為數目少但內容多的資料塊，或是分為數目多但內容少的資料塊。用戶端應用程式一定會收到佇列中的所有訊息。變更屬性值可能會影響效能，但不會影響所擷取的資料總數。)布林值屬性 `imqLoadMaxToServerSession` 控制連線用戶在應用程式伺服器階段作業中的運作方式：若此屬性值為 `true`，用戶端就會將最大數量的訊息載入到伺服器階段作業中；若為 `false`，則用戶端一次僅載入一則訊息。

標準訊息特性

Java Message Service 規格會定義部分標準訊息特性，JMS 提供者 (例如 Message Queue) 可以選擇性支援這些特性。根據慣例，所有這類標準特性的名稱會以字母 `JMSX` 開頭。

表 16-7 中所列的連線工廠屬性，可控制用戶端執行階段是否設定這類標準特性中的部分特性。對於產生的訊息，包括下列特性：

- `JMSXUserID` 傳送訊息之使用者的身份識別
- `JMSXAppID` 傳送訊息之應用程式的身份識別
- `JMSXProducerTXID` 產生訊息之作業事件的作業事件識別碼

對於使用的訊息，則包括

- `JMSXConsumerTXID` 使用訊息之作業事件的作業事件識別碼
- `JMSXRcvTimestamp` 訊息傳送至用戶的時間

訊息標頭置換

您可以使用表 16-8 中所列的連線工廠屬性，置換用戶端為特定 JMS 訊息標頭欄位所設定的值。由該連線工廠取得的連線所產生之所有訊息會使用您指定的設定值。您可以以此方式置換的標頭欄位包括：

- `JMSDeliveryMode` 傳送模式 (永久性或非永久性)
- `JMSExpiration` 過期時間

- `JMSPriority` 優先權層級

如上所述的每個欄位均有兩個屬性：一個是控制能否置換該欄位的布林值，另一個屬性則指定該欄位的值。例如，設定優先權層級的屬性是 `imqOverrideJMSPriority` 和 `imqJMSPriority`。另外還有一個屬性 `imqOverrideJMSHeadersToTemporaryDestinations`，用來控制置換值是否會套用到暫時目標。

備註 - 由於置換訊息標頭可能會干擾特定應用程式的需求，因此請先諮詢應用程式設計者或使用者，然後再使用這些屬性。

目標屬性

識別實體佇列或主題目標的**目標**受管理物件只有兩個屬性，如表 16-7 中所列。其中的 `imqDestinationName` 屬性較為重要，該屬性提供此受管理物件所代表的實體目標名稱；這個名稱是在使用 `imqcmd create dst` 指令建立實體目標時透過 `-n` 選項所指定。(請注意，目標受管理物件和其代表的實體目標之間不一定存有一對一的關係：單一實體目標可以被多個受管理物件所參照，或不被任何受管理物件所參照。)另外還有一個選擇性描述字串 `imqDestinationDescription`，可用以協助您識別目標物件，讓其與您之前建立的物件有所區分。

使用物件管理員公用程式

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

<code>add</code>	將受管理物件增加至物件存放區
<code>delete</code>	從物件存放區刪除受管理物件
<code>list</code>	列出物件存放區中現有的受管理物件
<code>query</code>	顯示有關受管理物件的資訊
<code>update</code>	修改受管理物件的屬性

如需 `imqobjmgr` 指令的語法、子指令和選項之參考資訊，請參閱第 229 頁的「物件管理員公用程式」。

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

- 受管理物件的 **JNDI 查找名稱** (`-l`)
用戶端應用程式會依據這個邏輯名稱，使用 Java Naming and Directory Interface 在物件存放區中查找受管理物件。
- **JNDI 物件存放區的屬性** (`-j`)

如需有關可能的屬性及其值的資訊，請參閱第 141 頁的「物件存放區」。

- 受管理物件的**類型** (-t)

可能的類型包括：

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

- 受管理物件的**屬性** (-o)

如需有關可能的屬性及其值的資訊，請參閱第 143 頁的「受管理物件屬性」。

新增受管理物件

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

備註 - 物件管理員僅會列出與顯示 Message Queue 受管理物件。如果物件存放區中包含非 Message Queue 物件，而且其查找名稱與您想增加的受管理物件名稱相同，則當您嘗試執行增加作業時會出現錯誤。

新增連線工廠

若要讓用戶端應用程式能建立代理程式連線，請為要建立的連線類型增加連線工廠受管理物件：佇列連線工廠或主題連線工廠。範例 8-1 中顯示的指令，會將佇列連線工廠 (受管理物件類型 `qf`) 增加至 LDAP 物件存放區。物件具有查找名稱 `cn=myQCF`，並使用 `jms` 連線服務連線至連接埠號 7272 之主機 `myHost` 上所執行的代理程式。

範例 8-1 新增連線工廠

```

imqobjmgr add
-l "cn=myQCF"
-j "java.naming.factory.initial=com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=ldap://mydomain.com:389/o=imq"
-j "java.naming.security.principal=uid=homerSimpson,ou=People,o=imq"
-j "java.naming.security.credentials=doh"
-j "java.naming.security.authentication=simple"
-t qf
-o "imqAddressList=mq://myHost:7272/jms"

```

新增目標

建立代表目標的受管理物件時，最好先建立實體目標，再將受管理物件增加至物件存放區中。使用指令公用程式 (`imqcmd`) 建立實體目標，如第 106 頁的「[建立實體目標](#)」中所述。

範例 8-2 中顯示的指令，會將受管理物件增加至代表主題目標之 LDAP 物件存放區，且該目標的查找名稱爲 `myTopic`，實體目標名稱爲 `physTopic`。增加佇列目標的指令很類似，不同之處在於受管理物件類型 (`-t` 選項) 爲 `q` (表示「佇列目標」)，而非 `t` (表示「主題目標」)。

範例 8-2 將目標新增至 LDAP 物件存放區

```

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

```

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

範例 8-3 將目標新增至檔案系統物件存放區

```

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

```

範例 8-3 將目標新增至檔案系統物件存放區 (續)

```
-o "imqDestinationName=physTopic"
```

刪除受管理物件

若要從物件存放區刪除受管理物件，請使用 `imqobjmgr` 指令的 `delete` 子指令，指定要刪除的物件之查找名稱、類型和位置。範例 8-4 中顯示的指令會刪除上述第 151 頁的「新增目標」中增加的物件。

範例 8-4 刪除受管理物件

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

列出受管理物件

您可以使用物件管理員的 `list` 子指令，列出物件存放區中的所有受管理物件，或列出特定類型的受管理物件。範例 8-5 顯示如何列出 LDAP 伺服器上所有的受管理物件。

範例 8-5 列出所有受管理物件

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

範例 8-6 列出所有佇列目標 (類型 `q`)。

範例 8-6 列出特定類型的受管理物件

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


範例 8-6 列出特定類型的受管理物件 (續)

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

檢視受管理物件資訊

query 子指令會顯示特定受管理物件的相關資訊，識別該物件的方法，是利用其查找名稱和包含該物件的物件存放區屬性。範例 8-7 顯示查找名稱為 cn=myTopic 之物件的相關資訊。

範例 8-7 檢視受管理物件資訊

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

修改受管理物件屬性

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

第 153 頁的「修改受管理物件屬性」針對範例 8-8 中增加至物件存放區的佇列連線工廠，變更其 imqReconnectAttempts 屬性值。

範例 8-8 修改受管理物件的屬性

```
imqobjmgr update
-l "cn=myQCF"
-j "java.naming.factory.initial=com.sun.jndi.ldap.LdapCtxFactory"
-j "java.naming.provider.url=ldap://mydomain.com:389/o=imq"
-j "java.naming.security.principal=uid=homerSimpson,ou=People,o=imq"
-j "java.naming.security.credentials=doh"
-j "java.naming.security.authentication=simple"
-t qf
-o "imqReconnectAttempts=3"
```

使用指令檔

`imqobjmgr` 指令的 `-i` 選項，可讓您指定使用 Java 特性檔案語法的指令檔名稱，以表示所有或部分子指令陳述式。此功能在指定物件存放區屬性時極為有用，執行這類作業時一般需要大量的鍵入操作，並且在多次呼叫 `imqobjmgr` 時所鍵入的內容可能都相同。使用指令檔也可讓您避免超出指令行所允許的最大字元數。

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

範例 8-9 物件管理員指令檔語法

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

例如，之前在**範例 8-1**中所顯示的物件管理員指令，會將佇列連線工廠增加至 LDAP 物件存放區。此指令可以封裝在**範例 8-10**中所示的指令檔中。若是指令檔命名為 `MyCmdFile`，您可以使用指令行執行指令

```
imqobjmgr -i MyCmdFile
```

範例 8-10 指令檔範例

```
version=2.0
cmdtype=add
obj.lookupName=cn=myQCF
objstore.attrs.java.naming.factory.initial=com.sun.jndi.ldap.LdapCtxFactory
objstore.attrs.java.naming.provider.url=ldap://mydomain.com:389/o=imq
objstore.attrs.java.naming.security.principal=\
    uid=homerSimpson,ou=People,o=imq
objstore.attrs.java.naming.security.credentials=doh
objstore.attrs.java.naming.security.authentication=simple
obj.type=qf
obj.attrs.imqAddressList=mq://myHost:7272/jms
```

指令檔也可以用來僅指定 `imqobjmgr` 子指令陳述式的一部分，剩下部分則由指令行明確提供。例如，[範例 8-11](#) 中所顯示的指令檔僅指定了 LDAP 物件存放區的屬性值。

範例 8-11 部分指令檔

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

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

範例 8-12 使用部分指令檔

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

視您所使用的平台而定，可於以下位置找到指令檔的其他範例：

Solaris : `/usr/demo/imq/imqobjmgr` **Linux** : `/opt/sun/mq/examples/imqobjmgr`

Windows : `IMQ_HOME/demo/imqobjmgr`

使用代理程式叢集

Message Queue™ Enterprise Edition 支援使用**代理程式叢集**，也就是會共同協作，向用戶端提供訊息傳送服務的代理程式群組。叢集可使訊息服務透過分發用戶端的連線至數個代理程式，來延展其訊息流量的作業。如需叢集和其運作方式的一般說明，請參閱「Message Queue 技術摘要」。

本章描述如何管理代理程式叢集、將叢集與代理程式連線並進行配置。它包含下面各節：

- 第 157 頁的「叢集配置特性」
- 第 159 頁的「管理叢集」
- 第 161 頁的「主代理程式」

叢集配置特性

您可以為叢集的每個代理程式成員指定**叢集配置特性**，以定義叢集。您可以為叢集中的每個代理程式個別設定這些特性，但通常較方便的做法是將所有配置特性集中至所有代理程式均參照的**叢集配置檔案**中。這樣可以防止違反合約的設定，並確保叢集中的所有代理程式都能共用相同且一致的配置資訊。

叢集配置特性詳述於表 14-9 中。這些部分包括：

- `imq.cluster.brokerlist` 將主機名稱和連接埠號賦予屬於叢集的所有代理程式。
- `imq.cluster.masterbroker` 指定哪一個代理程式 (若有) 為記錄狀態變更的主代理程式。
- `imq.cluster.url` 指定叢集配置檔案 (若有) 的位置。
- `imq.cluster.hostname` 將主機名稱或 IP 位址賦予 `cluster` 連線服務，用於在叢集的各代理程式之間進行內部通訊。如果有多個主機，此設定才會有用：例如，如果一部電腦中有多個網路介面卡。
- `imq.cluster.port` 將連接埠號賦予 `cluster` 連線服務。

- `imq.cluster.transport` 指定 cluster 連線服務所使用的傳輸協定，例如 `tcp` 或 `ssl`。

您可以單獨設定個別代理程式的 `hostname` 和 `port` 特性，但是叢集中所有代理程式的 `brokerlist`、`masterbroker`、`url` 和 `transport` 值必須相同。

以下各節描述如何設定代理程式叢集配置特性，可以為叢集中的每個代理程式個別設定，或者使用叢集配置檔案集中設定。

設定個別代理程式的叢集特性

您可以在代理程式的實例配置檔案中 (或啟動代理程式時在指令行上) 設定叢集配置特性。例如，若要建立的叢集包含 `host1` 連接埠 `9876` 上、`host2` 連接埠 `5000` 上和 `ctrlhost` 預設連接埠 (`7676`) 上的代理程式，三個代理程式的實例配置檔案中都將包含以下的特性：

```
imq.cluster.brokerlist=host1:9876,host2:5000,ctrlhost
```

請注意，如果您需要變更叢集配置，則此方法要求更新叢集中所有代理程式的實例配置檔案。

使用叢集配置檔案

基於一致性以及便於維護的考量，建議您將所有的共用叢集配置特性集合至單一叢集配置檔案中，而不要為每個代理程式個別地進行設定。在此方法中，每個代理程式的實例配置檔案都必須設定 `imq.cluster.url` 特性指向叢集配置檔案的位置，例如：

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

接著，叢集配置檔案會為叢集中的所有代理程式定義共用配置特性，例如要連線的代理程式清單 (`imq.cluster.brokerlist`)、用於 cluster 連線服務的傳輸協定 (`imq.cluster.transport`)，以及可以選擇性設定的主代理程式的位址 (`imq.cluster.masterbroker`)。以下的程式碼定義與上一個範例中相同的叢集，其中在 `ctrlhost` 上執行的代理程式作為主代理程式：

```
imq.cluster.brokerlist=host1:9876,host2:5000,ctrlhost
imq.cluster.masterbroker=ctrlhost
```

管理叢集

本節說明如何連線一組代理程式以形成一個叢集、將新的代理程式增加至現有叢集，以及從叢集中移除代理程式。

連線代理程式

有兩種通用方法可將代理程式連線至叢集：從指令行使用 `-cluster` 選項進行連線，或在叢集配置檔案中設定 `imq.cluster.brokerlist` 特性。無論使用哪種方法，您啟動的每個代理程式每 5 秒鐘會嘗試連線至其他代理程式；在此叢集中的主代理程式 (如果已配置) 啟動後，連線即可成功。如果叢集中有代理程式比主代理程式先啟動，則此代理程式將保持暫停狀態，從而拒絕用戶端連線，直到主代理程式啟動時；然後暫停的代理程式的全部功能將自動可用。

若要從指令行配置代理程式叢集，請在啟動每個代理程式時，使用 `imqbrokerd` 指令的 `-cluster` 選項，指定叢集的完整代理程式清單。例如，以下指令啟動新的代理程式，並將其連線至在 `host1` 的預設連接埠 (7676) 上執行的代理程式、在 `host2` 的連接埠 5000 上執行的代理程式，以及在預設主機 (`localhost`) 的連接埠 9876 上執行的代理程式：

```
imqbrokerd -cluster host1,host2:5000,:9876
```

另一種更適合用於生產系統的替代方法，就是建立一個使用 `imq.cluster.brokerlist` 特性的叢集配置檔案，以指定要連線之代理程式的清單。叢集中的每個代理程式均必須將自己的 `imq.cluster.url` 特性設定為指向此叢集配置檔案。

無論您使用哪種方法，都必須確定叢集中沒有代理程式位址解譯為網路迴路 IP 位址 (127.0.0.1)。任何配置為這個位址的代理程式將無法連線至叢集中的其他代理程式。

備註 – 一些 Linux 安裝程式會自動將 `localhost` 項目設定為網路迴路位址。在這類系統上，您必須修改系統 IP 位址，這樣才能為叢集中的所有代理程式設定正確的位址。

在設定叢集時，請在參與叢集的所有 Linux 系統中檢查 `/etc/hosts` 檔案。如果系統使用的是靜態 IP 位址，請編輯 `/etc/hosts` 檔案，為 `localhost` 指定正確的位址。如果位址已註冊網域命名服務 (DNS)，請編輯 `/etc/nsswitch.conf` 檔案以變更項目的順序，讓系統可以在參考本機 `hosts` 檔案之前執行 DNS 查找。`/etc/nsswitch.conf` 中的行應如下所示：

```
hosts: dns files
```

如果您想要在叢集中的代理程式之間以安全加密的方式進行訊息傳送，請配置 `cluster` 連線服務以使用 SSL 型傳輸協定。針對叢集中的每個代理程式，設定 SSL 型連線服務，如第 130 頁的「訊息加密」中所述。然後再使用叢集配置檔案，或透過為每個代理程式個別設定的方式，將 `imq.cluster.transport` 特性設定為 `ssl`。

將代理程式新增至叢集

將新代理程式增加至叢集的程序取決於叢集是否使用叢集配置檔案。

▼ 使用叢集配置檔案將新的代理程式新增至叢集

- 1 將新代理程式增加至叢集配置檔案中的 `imq.cluster.brokerlist` 特性。

- 2 向叢集中的每個代理程式發出以下指令：

```
imqcmd reload cls
```

此指令強制所有代理程式重新載入叢集配置，確保此叢集中代理程式的所有永久性資訊均為最新的。請注意，您不需要為叢集中的每個代理程式發出此指令；只要對任何一個代理程式執行此指令，就會使得所有代理程式都重新載入叢集配置。

- 3 (選擇性) 在代理程式的 `config.properties` 檔案中設定 `imq.cluster.url` 特性的值，讓其指向叢集配置檔案。
- 4 啟動新的代理程式。

如果您並未執行第 160 頁的「將代理程式新增至叢集」，請使用 `imqbrokerd` 指令行上的 `-D` 選項以設定 `imq.cluster.url` 的值。

不使用叢集配置檔案將新的代理程式新增至叢集

您可以編輯 `config.properties` 檔案或使用 `imqbrokerd` 指令行上的 `-D` 選項來設定以下特性的值：

- `imq.cluster.brokerlist`
- `imq.cluster.masterbroker` (如有需要)
- `imq.cluster.transport` (如果您是使用安全 `cluster` 連線服務)

從叢集中移除代理程式

您用來從叢集中移除代理程式的方法，取決於原先是透過指令行建立叢集，還是使用集中叢集配置檔案所建立。

使用指令行移除代理程式

如果您是使用指令行的 `imqbrokerd` 指令將代理程式連線至叢集，則必須停止各代理程式，然後在指令行上指定一組新的叢集成員並重新啟動。程序如下：

▼ 使用指令行從叢集中移除代理程式

- 1 使用 `imqcmd` 指令停止叢集中的各代理程式。

- 2 重新啟動將保留在叢集中的代理程式 (使用 `imqbrokerd` 指令的 `-cluster` 選項，指定要保留的代理程式)。

例如，假設您原本使用以下的指令來啟動 A、B 和 C 代理程式來建立包含這三者的叢集：

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

若要從叢集中移除代理程式 A，請使用以下指令重新啟動代理程式 B 和 C：

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

使用叢集配置檔案移除代理程式

如果您原本是集中使用叢集配置檔案中的 `imq.cluster.brokerlist` 特性來指定叢集的成員代理程式以建立叢集，就不需要爲了移除其中一個代理程式而停止所有的代理程式。反之，您只要編輯配置檔案來排除要移除的代理程式，以強制剩餘的叢集成員重新載入叢集配置，並重新配置已排除的代理程式，這樣它就不再指向相同的叢集配置檔案。程序如下：

▼ 使用叢集配置檔案從叢集中移除代理程式

- 1 編輯叢集配置檔案，從以 `imq.cluster.brokerlist` 特性指定的清單中移除要排除的代理程式。
- 2 向叢集中每個保留的代理程式發出以下指令：

```
imqcmd reload cls
```

這會強制代理程式重新載入叢集配置。
- 3 停止要從叢集中移除的代理程式。
- 4 編輯此代理程式的 `config.properties` 檔案，移除 `imq.cluster.url` 特性或指定一個不同的值。

主代理程式

您可以選擇是否讓叢集擁有一個**主代理程式**，它可維護**配置變更記錄**，以記錄叢集持續性狀態中的任何變更。無論在叢集配置檔案或個別代理程式的實例配置檔案中，主代理程式都是由 `imq.cluster.masterbroker` 配置特性所識別。

配置變更記錄包含與叢集相關聯之永久性實體中的變更資訊，例如長期訂閱和管理員建立之實體目標。叢集中的所有代理程式會在啟動期間參考主代理程式，以更新這些永久性實體的資訊。主代理程式發生故障時，將無法進行這類同步化作業；如需更多資訊，請參閱第 162 頁的「當主代理程式無法使用時」。

管理配置變更記錄

由於配置變更記錄包含重要資訊，您必須定期進行備份，使其能夠在發生故障的情況下復原。雖然從備份中復原將會失去自當時備份後叢集持續性狀態中的任何變更，但經常定期備份可盡量減低此潛在的資訊遺失。備份與復原作業可以有效地壓縮和最佳化包含在配置變更記錄中的變更歷史，且配置變更記錄能夠隨著時間而迅速增長。

備份配置變更記錄

使用 `imqbrokerd` 指令的 `-backup` 選項來指定備份檔案的名稱。例如：

```
imqbrokerd -backup mybackuplog
```

▼ 復原配置變更記錄

- 1 關閉叢集中的所有代理程式。
- 2 使用下列指令從備份檔案中復原主代理程式的配置變更記錄

```
imqbrokerd -restore mybackuplog
```
- 3 如果您為主代理程式指定一個新的名稱或連接埠號，請相應地在叢集配置檔案中更新 `imq.cluster.brokerlist` 和 `imq.cluster.masterbroker` 特性。
- 4 重新啟動叢集中的所有代理程式。

當主代理程式無法使用時

由於叢集中的所有代理程式都需要主代理程式才能執行永久性作業，因此沒有可用的主代理程式時，叢集中所有代理程式的下列 `imqcmd` 子指令都會傳回錯誤：

- `create dst`
- `destroy dst`
- `update dst`
- `destroy dur`

自動建立的實體目標和暫時目標不受影響。

在未配置主代理程式的環境中，任何嘗試建立或取消長期訂閱的用戶端應用程式都會收到一個錯誤。但是用戶端可以指定並與現有的長期訂閱進行互動。

◆◆◆ 第 10 章

監視代理程式

本章描述可用來監視代理程式的工具，以及如何取得度量資料。本章包含下列各小節：

- 第 163 頁的「監視工具簡介」
- 第 164 頁的「配置及使用代理程式記錄功能」
- 第 169 頁的「以互動方式顯示度量」
- 第 174 頁的「撰寫應用程式來監視代理程式」

有關特定度量的參考資訊，請參閱第 18 章。

監視工具簡介

Message Queue™ 資訊有 3 個監視介面：記錄檔、互動式指令，以及可以取得度量的用戶端 API。各有下列優點和缺點：

- 記錄檔提供長期的度量資料記錄，但是不易剖析。
- 指令可讓您快速取得符合您需求的資訊範例，但無法查看歷程資訊或有計劃的處理資料。
- 用戶端 API 可讓您擷取資訊、處理資訊、處理資料、顯示圖表或傳送警示。然而若要使用它，您必須撰寫自訂應用程式來擷取和分析資料。

表 10-1 對不同的工具進行比較。

表 10-1 度量監視工具的優點與限制

度量監視工具	優點	限制
imqcmd metrics	遠端監視 便於抽查 在指令選項中設定報告間隔；可即時變更 易於選取偏好的特定資料 以簡單表格格式呈現的資料	沒有取得所有資料的單一指令 難以有計劃分析資料 未建立歷程記錄 難以查看歷程趨勢
記錄檔	固定取樣 建立歷程記錄	需要配置代理程式特性；必須關閉再重新啓動代理程式，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 無法即時變更報告時間間隔；所有度量資料的報告時間間隔都相同 不提供彈性選取資料 僅代理程式度量；不包括目標和連線服務度量 若時間間隔過短可能會影響效能
用戶端 API	遠端監視 易於選取偏好的特定資料 可以有計劃的分析資料並以任何格式呈現	需要配置代理程式特性；必須關閉再重新啓動代理程式，才可生效 您必須寫入您自己的度量監視用戶端 無法即時變更報告時間間隔；所有度量資料的報告時間間隔都相同

除了表格中顯示的差異之外，每個工具所收集之代理程式產生的度量資訊也會稍有不同。如需有關各個監視工具所收集度量資料的詳細資訊，請參閱第 18 章。

配置及使用代理程式記錄功能

Message Queue 記錄程式會取得代理程式程式碼、除錯程式和度量產生器所產生的資訊，並且將這些資訊寫入數個輸出通道：標準輸出 (主控台)、記錄檔，以及 Solaris™ 作業系統上的 `syslog` 常駐程式程序。

您可以指定記錄程式收集的資訊類型，以及寫入每個輸出通道的類型。要特別注意的是，您可以指定您要寫入記錄檔的度量資訊。

本節描述代理程式的預設記錄配置，並介紹如何將記錄資訊重新導向至替代輸出通道、如何變更記錄檔自動重建準則，以及如何將度量資料傳送到記錄檔。

預設記錄配置

代理程式會自動配置為將記錄輸出儲存到一組自動重建的記錄檔中。這些記錄檔的所在目錄名稱，與相關聯代理程式之實例名稱相同 (請參閱[附錄 A](#))：

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

備註 – 如果代理程式的生命週期由應用程式伺服器所控制，則記錄檔的位置是在啟動該代理程式所在網域的網域目錄子目錄中：

```
.../appServer_domainName_dir/imq/instances/imqbroker/log
```

記錄檔為簡單的文字檔案。其命名方式如下 (從最早建立到最新建立)：

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

依預設，每週自動重建一次記錄檔；而系統會維護 9 個備份檔案。

- 若要變更保留記錄檔的目錄，請將特性 `imq.log.file.dirpath` 設定為所需的路徑。
- 若要將記錄檔的根名稱從 `log` 變更為其他名稱，請設定 `imq.log.file.filename` 特性。

代理程式支援三種記錄層級：ERROR、WARNING、INFO。表 10-2 解釋各個層級的涵義。

表 10-2 記錄層級

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

設定某個記錄層級，則會收集此層級以及所有更高層級的訊息。預設記錄層級是 INFO，因此依預設，ERROR、WARNING 和 INFO 訊息會全部記錄。

記錄訊息格式

記錄的訊息包含時間標記、訊息程式碼和訊息本身。資訊容量因您所設定的記錄層級而有所不同。以下為 `INFO` 訊息的範例。

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

若要變更時間標記的時區，請參閱表 14-8 中有關描述 `imq.log.timezone` 特性的資訊。

變更記錄程式配置

表 14-8 描述記錄的相關特性。

▼ 變更代理程式的記錄程式配置

- 1 設定記錄層級。
- 2 為一個或多個記錄種類設定輸出通道 (檔案和/或主控台)。
- 3 如果您將輸出記錄至檔案，請為此檔案配置自動重建準則。
可以透過設定記錄程式特性來完成這些步驟。您可以用以下兩種方法之一執行此作業：
 - 啓動代理程式之前，請先變更或增加代理程式之 `config.properties` 檔案中的記錄程式特性。
 - 在啓動代理程式的 `imqbrokerd` 指令中，指定記錄程式指令行選項。您也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，會置換在代理程式實例配置檔案中所指定的特性。以下的 `imqbrokerd` 選項會影響記錄：

```
-metrics interval    代理程式度量的記錄時間間隔 (以秒為單位)
-loglevel level      記錄層級 (ERROR、WARNING、INFO 或 NONE)
-silent              無訊息模式 (不記錄至主控台)
-tty                 將所有訊息記錄至主控台
```

以下各節描述如何變更預設配置，以執行以下作業：

- 變更輸出通道 (記錄訊息的目標)
- 變更自動重建準則

變更輸出通道

依預設，錯誤和警告訊息可顯示在終端機上，也可記錄至記錄檔中。(在 Solaris 上，錯誤訊息也可寫入至系統的 `syslog` 常駐程式。)

您可以以下方法變更記錄訊息的輸出通道：

- 若要使**所有**記錄種類(對於指定的層級)輸出顯示在畫面上，請使用 `imqbrokerd` 指令的 `-tty` 選項。
- 若要避免記錄輸出顯示在畫面上，請使用 `imqbrokerd` 指令的 `-silent` 選項。
- 使用 `imq.log.file.output` 特性，指定應寫入至記錄檔的記錄資訊種類。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種類。例如：

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

- 在 Solaris 上，使用 `imq.log.syslog.output` 特性，指定應寫入至 Solaris `syslog` 的記錄資訊種類。例如：

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

備註 – 變更記錄程式輸出通道前，您必須確定所設定的記錄層級，支援要對映到輸出通道的資訊。例如，如果您將記錄層級設定為 `ERROR`，並且將 `imq.log.console.output` 特性設定為 `WARNING`，則不會記錄任何訊息，因為您未啓用 `WARNING` 訊息的記錄。

變更記錄檔自動重建準則

有兩個記錄檔自動重建準則：時間和容量。預設為使用時間準則，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
```

- 若要將自動重建準則變更為取決於檔案容量，您需要設定 `imq.log.file.rolloverbytes` 特性。例如，以下定義會讓代理程式在檔案達到 500,000 個位元組的限制時，就自動重建檔案。

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

如果同時設定與時間相關的自動重建特性以及與容量相關的自動重建特性，則先達到的限制將觸發自動重建作業。如上所述，代理程式最多可保留九個自動重建檔案。

代理程式執行時，您可以設定或變更記錄檔自動重建特性。若要設定這些特性，請使用 `imqcmd update bkr` 指令。

將度量資料傳送到記錄檔

本節描述使用代理程式記錄檔報告度量資訊的程序。如需有關配置記錄程式的一般資訊，請參閱第 164 頁的「配置及使用代理程式記錄功能」。

▼ 使用記錄檔報告度量資訊

1 配置代理程式的度量產生功能：

a. 確定 `imq.metrics.enabled=true`
依預設，會開啓用於記錄的度量產生功能。

b. 設定度量產生的合適秒數時間間隔。

```
imq.metrics.interval=interval
```

您可以在 `config.properties` 檔案中設定此值，或在啓動代理程式時，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。啓動代理程式時，您可以在 `config.properties` 檔案中設定此值，或使用 `-loglevel level` 指令行選項設定此值。

3 確定已設定記錄程式，將度量資訊寫入記錄檔：

```
imq.log.file.output=INFO
```

這是預設值。您可以在 `config.properties` 檔案中進行設定。

4 啓動代理程式。

以下所示為代理程式度量輸出到記錄檔的範例：

```
[21/Jul/2004:11:21:18 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)
```

如需有關度量資料的參考資訊，請參閱第 18 章。

記錄停用的訊息

您可以啟用代理程式停用的訊息之記錄功能來監視實體目標。無論您有沒有使用此停用的訊息佇列，都可以記錄這些停用的訊息。

啟用此停用的訊息之記錄功能後，代理程式會記錄下列類型的事件：

- 實體目標超出最大容量。
- 代理程式移除實體目標的訊息，原因可能如下：
 - 已達目標容量限制。
 - 訊息的時限已到。
 - 訊息太大。
 - 代理程式處理訊息時發生錯誤。

如果停用的訊息佇列在使用中，還會記錄下列類型的事件：

- 代理程式將訊息移動到停用的訊息佇列。
- 代理程式將停用的訊息佇列中的訊息移除並且捨棄。

以下為停用的訊息之記錄格式範例：

```
[29/Mar/2006:15:35:39 PST] [B1147]: Message 8-129.145.180.87(e7:6b:dd:5d:98:aa)-35251-1143675279400 from destination Q:q0 has been placed on the DMQ because [B0053]: Message on destination Q:q0 Expired: expiration time 1143675279402, arrival time 1143675279401, JMSTimestamp 1143675279400
```

依預設，不記錄停用的訊息。若要記錄，請設定代理程式屬性 `imq.destination.logDeadMsgs`。

以互動方式顯示度量

Message Queue 代理程式可以報告下列類型的度量：

- **Java 虛擬機器 (JVM) 度量**。有關 JVM 堆疊容量的資訊。
- **代理程式通用度量**。有關儲存於代理程式中的訊息、輸入和輸出代理程式的訊息流量，以及記憶體用途的資訊。追蹤訊息的數量和位元組數。
- **連線服務度量**。有關連線和連線執行緒資源的資訊，以及有關特定連線服務訊息流量的資訊。
- **目標度量**。有關傳入和傳出特定實體目標之訊息流量的資訊，以及有關實體目標用戶、記憶體和磁碟空間使用率的資訊。

`imqcmd` 指令可以取得整個代理程式、個別連線服務和個別實體目標的度量資訊。若要取得度量資料，通常可使用 `imqcmd` 的子指令 `metrics`。度量資料會依您指定的時間間隔或次數寫入主控台畫面。

您也可以使用 `query` 子指令，檢視也包含配置資訊的類似資料。如需更多資訊，請參閱第 173 頁的「`imqcmd query`」。

imqcmd metrics

`imqcmd metrics` 的語法和選項分別顯示於表 10-3 和表 10-4 中。

表 10-3 `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:portNumber</i>] [-m <i>metricType</i>] [-int <i>interval</i>] [-msp <i>numSamples</i>]</pre>	顯示預設代理程式或指定主機和連接埠上的代理程式的代理程式度量。
<pre>metrics svc -n <i>serviceName</i> [-b <i>hostName:portNumber</i>] [-m <i>metricType</i>] [-int <i>interval</i>] [-msp <i>numSamples</i>]</pre>	顯示預設代理程式或指定主機和連接埠上的代理程式上的指定服務之標準資訊。
<pre>metrics dst -t <i>destType</i> -n <i>destName</i> [-b <i>hostName:portNumber</i>] [-m <i>metricType</i>] [-int <i>interval</i>] [-msp <i>numSamples</i>]</pre>	顯示指定類型和名稱的實體目標之度量資訊。

表 10-4 `imqcmd metrics` 子指令選項

子指令選項	說明
<code>-b <i>hostName:portNumber</i></code>	指定報告度量資訊之代理程式的主機名稱和連接埠。預設值為 <code>localhost:7676</code> 。
<code>-int <i>interval</i></code>	指定顯示度量的時間間隔 (以秒為單位)。預設為 5 秒鐘。

表 10-4 imqcmd metrics 子指令選項 (續)

子指令選項	說明
-m <i>metricType</i>	指定要顯示的度量類型： ttl 顯示傳入和傳出代理程式、服務或目標的訊息與資料封包等度量 (預設度量類型)。 rts 顯示傳入和傳出代理程式、連線服務或目標的訊息和資料封包等流量速率度量 (每秒)。 cxn 顯示連線、虛擬記憶體堆疊和執行緒 (僅限代理程式和連線服務)。 con 顯示用戶相關的度量 (僅限目標)。 dsk 顯示磁碟使用率度量 (僅限目標)。
-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>	指定要針對其度量資料進行報告的實體目標 (如果存在) 名稱。沒有預設值。
-n <i>serviceName</i>	指定要針對其度量資料進行報告的連線服務 (如果存在)。沒有預設值。
-t <i>destType</i>	指定要針對其度量資料進行報告的實體目標 (如果存在) 類型 (佇列或主題)。沒有預設值。

使用顯示度量資料的 metrics 子指令

本小節描述使用 metrics 子指令報告度量資訊的程序。

▼ 使用 metrics 子指令

- 1 啟動需要度量資訊的代理程式。
請參閱第 64 頁的「啟動代理程式」。
- 2 發出適當的 imqcmd metrics 子指令和選項，如表 10-3 和表 10-4 中所示。

度量輸出：imqcmd 度量

本小節包含 imqcmd metrics 子指令的輸出範例。這些範例顯示代理程式範圍、連線服務和實體目標度量。

代理程式通用度量

若要取得訊息和封裝每隔 10 秒傳入和傳出代理程式的速率，請使用 `metrics bkr` 子指令：

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

此指令會產生與以下相類似的輸出 (請參閱表 18-2 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

連線服務度量

若要取得 `jms` 連線服務所處理的訊息和資料封包等累積總數，請使用 `metrics svc` 子指令：

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

此指令會產生與以下相類似的輸出 (請參閱表 18-3 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

實體目標度量

若要取得有關實體目標的度量資訊，請使用 `metrics dst` 子指令：

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

此指令會產生與以下相類似的輸出 (請參閱表 18-4 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
```

若要取得有關實體目標的用戶資訊，請使用以下 `metrics dst` 子指令：

```
imqcmd metrics dst -t q -n SimpleQueue -m con -u admin
```

此指令會產生與以下相類似的輸出 (請參閱表 18-4 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

表 10-5 中顯示 `imqcmd query` 的語法和選項，以及該指令所提供之度量資料的說明。

表 10-5 imqcmd query 子指令語法

子指令語法	提供的度量資料
<code>query bkr</code> [-b <i>hostName:portNumber</i>]	有關目前儲存於代理程式記憶體與永久性存放區中的訊息數目和訊息容量等資訊 (請參閱第 92 頁的「顯示代理程式資訊」)。
或	
<code>query svc -n <i>serviceName</i></code> [-b <i>hostName:portNumber</i>]	有關目前所指定連線服務的已配置執行緒數目和連線數目等資訊 (請參閱第 97 頁的「顯示連線服務資訊」)。
或	
<code>query dst -t <i>destType</i></code> -n <i>destName</i> [-b <i>hostName:portNumber</i>]	有關目前所指定目標的產生器、使用中用戶和備份用戶，以及儲存在記憶體和永久性存放區中的訊息和訊息容量等數目資訊 (請參閱第 108 頁的「顯示實體目標資訊」)。

備註 – 由於 `imqcmd query` 提供的度量資料有限，因此未在第 18 章的表格中列出此工具。

撰寫應用程式來監視代理程式

Message Queue 提供度量監視功能，此功能可讓代理程式將度量資料寫入 JMS 訊息，然後根據訊息中包含的度量資訊類型，將資料傳送到其中一個度量主題目標。

您可以藉由寫入訂閱度量主題目標、使用目標中的訊息並處理訊息中度量資訊的用戶端應用程式，來存取此度量資訊。

總共有 5 個度量主題目標，表 10-6 中顯示這些目標的名稱，以及應傳送到每個目標的度量訊息類型。

表 10-6 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.monitoredDestinationName	用於指定名稱之佇列的目標度量
mq.metrics.destination.topic.monitoredDestinationName	用於指定名稱之主題的目標度量

設定訊息型監視

本小節描述使用訊息型監視功能以收集度量資訊的程序。程序包括用戶端部署和管理作業。

▼ 若要設定訊息型監視

1 寫入度量監視用戶端。

請參閱「Message Queue Developer's Guide for Java Clients」，瞭解如何以程式設計方式讓用戶端訂閱度量主題目標、使用度量訊息，以及從這些訊息擷取度量資料。

2 設定 config.properties 檔案中的代理程式特性值，以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時間間隔 (以秒為單位)。

設定 `imq.metrics.topic.interval=interval`。

預設為 60 秒。

- c. 指定度量訊息是否為永久性訊息 (即訊息是否在代理程式故障後仍然存留)。

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。

請參閱以下第 175 頁的「安全性與存取注意事項」中的說明。

- 4 啟動度量監視用戶端。

用戶訂閱度量主題時，即會自動建立度量主題目標。建立度量主題之後，代理程式度量訊息產生者就會開始傳送度量訊息到度量主題。

安全性與存取注意事項

有兩個限制度量主題目標存取的原因：

- 度量資料可能包含有關代理程式及其資源的機密資訊。
- 度量主題目標的訂閱數過量，可能會增加代理程式的經常性耗用時間，因此會對效能產生不良影響。

由於這些考量，所以建議您限制度量主題目標的存取。

監視用戶端必須遵守與其他用戶端相同的認證和授權控制。只有 Message Queue 使用者儲存庫中的使用者可以連線到代理程式。

您可以使用存取控制特性檔案來限制對於特定度量主題目標的存取，以提供額外保護，如第 124 頁的「使用者授權：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任何使用者存取 `mq.metrics.broker` 度量主題，但 `user1` 和 `user2` 除外。

```
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者 `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
```

您也可以根據度量資料的機密程度，使用加密連線將度量監視用戶端連線到代理程式。如需有關使用加密連線的資訊，請參閱第 130 頁的「訊息加密」。

度量輸出：度量訊息

使用訊息型監視 API 取得度量資料輸出，是您編寫度量監視用戶端需具備的功能之一。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 18 章。

分析與調校訊息服務

本章涵蓋的數個主題，說明如何分析與調校 Message Queue™ 服務，以便將訊息傳送應用程式的效能最佳化。它包括以下主題：

- 第 177 頁的「關於效能」
- 第 180 頁的「影響效能的因素」
- 第 188 頁的「調校配置以改善效能」

關於效能

本節提供效能調校的一些背景資訊。

效能調校程序

訊息傳送應用程式的效能取決於應用程式和 Message Queue 服務之間的互動。因此，最佳化效能需要應用程式開發者和管理員共同的努力。

最佳化效能的程序從應用程式設計開始，然後是部署應用程式，接著再對訊息服務進行調校。效能調校程序包括以下階段：

- 定義應用程式的效能需求
- 設計應用程式時必須考量影響效能的因素 (特別是可靠性與效能之間的取捨)
- 建立基準效能測量
- 調校或重新配置訊息服務以最佳化效能

以上略述的程序通常是反覆的。在部署應用程式期間，Message Queue 管理員應評估訊息服務是否符合應用程式的一般效能需求。如果效能評定測試符合這些需求，管理員可以根據本章說明調校系統。但是，如果效能評定測試不符合效能需求，則可能需要重新設計應用程式或修改架構部署。

效能類型

一般而言，效能是指訊息服務從產生器傳送訊息到用戶的速度和效率之測量結果。但是，您可能會依您的需要關注效能的不同方面。

- 連線負載 系統可支援之訊息產生器、訊息用戶或同步運作連線的數目。
- 訊息流量 每秒導入訊息傳送系統的訊息數目或訊息容量。
- 延時 從訊息產生器傳送特定訊息到訊息用戶所需的時間。
- 穩定性 訊息服務的整體可用性，或者在負載量較大或失敗時訊息服務緩慢降低的程度。
- 效率 訊息傳送的效率，亦即與已使用之運算資源相關的訊息流量測量結果。

通常這些不同方面的效能都相互關聯。如果訊息流量高，則表示訊息不太可能積存於代理程式中；因此，延時時間會比較短(單一訊息可迅速傳送)。但是，延時可取決於很多因素：通訊連結的速度、代理程式處理速度，以及用戶端處理速度等。

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

效能評定是針對訊息傳送應用程式建立測試套件，並且為此測試套件測量訊息流量或其他方面效能的程序。

例如，您可以建立一個測試套件，讓數個訊息產生用戶端使用數個連線、階段作業和訊息產生器，以某個特定速率，將標準容量的永久性或非永久性訊息傳送到數個佇列或主題(這些都取決於您的訊息傳送應用程式設計)。同樣地，此測試套件包括數個訊息使用用戶端，這些用戶端會使用數個連線和階段作業，以及在測試套件實體目標中以特定確認模式使用訊息的特定類型訊息用戶。

使用標準的測試套件，您可以測量訊息產生和使用間所需的時間或平均訊息流量速率，並且可以監視系統以觀察連線執行緒用法、訊息儲存資料、訊息流量資料和其他相關度量。在效能受到不良影響之前，您就可以迅速增加訊息產生的速率、訊息產生器的數目或其他變數。您可以達到的最大流量為您訊息服務配置的效能評定。

使用此效能評定，您可以修改部分測試套件的特徵。請謹慎控制所有可能影響效能的因素(請參閱第 181 頁的「影響效能的應用程式設計因素」)，您可以對變更其中某些因素會如何影響效能評定進行記錄。例如，您可以將連線數目或訊息容量增加 5 倍或 10 倍，然後記下效能所受的影響。

反過來，您也可以保持以應用程式為基礎的因素不變，而以某種控制方法變更代理程式配置(例如，變更連線特性、執行緒池特性、JVM 記憶體限制、限制運作方式、檔案式與 JDBC 型永久性... 等等)，並記錄這些變更對於效能所產生的影響。

應用程式的效能評定會提供資訊，這些資訊在您要藉由調校訊息服務增加已部署應用程式的效能時會有所幫助。效能評定可以更加準確地預測一個變更或一組變更的效果。

一般來說，效能評定應在受控制的測試環境中執行，並且要時間足夠長以讓訊息服務穩定執行。(啓動時，及時(Just-In-Time)編譯會將 Java 程式碼轉為機器碼，這會對效能造成不良影響。)

基準使用式樣

部署並執行訊息傳送應用程式之後，請務必建立基準使用式樣。您必須知道尖峰需求出現的時間並可以將需求數目化。例如，需求通常會因一般使用者的數目、作業層級、每天的時間或所有這些因素而有所變化。

若要建立基準使用式樣，您必須長時間監視訊息服務，查看下列資料：

- 連線數目
- 儲存於代理程式中(或特定實體目標中)的訊息數目
- 傳入和傳出代理程式(或特定實體目標)的訊息流量
- 使用中用戶數目

您還可以使用度量資料中提供的平均值與尖峰值。

檢查這些出乎設計意料之外的基準度量是非常重要的。經由此動作，您會檢查用戶端程式碼是否正常運作：例如，連線未處於開啓狀態，或使用的訊息未處於未確認狀態。這些編碼錯誤會耗用代理程式資源，並且可能對效能有很明顯的影響。

基準使用式樣可幫助您判斷如何將系統調校到最佳化效能。例如：

- 如果某個實體目標的使用頻繁程度比其他實體目標高出許多，則您可能需要將此實體目標的訊息記憶體限制設定得較高，或者據此調整限制行爲。
- 如果所需的連線數目明顯大於最大執行緒池容量允許的大小，那麼您可能要增加執行緒池大小，或調校共用執行緒模型。
- 如果尖峰訊息流量遠大於平均流量，則可能會影響您在記憶體不足時使用的限制運作方式。

一般而言，您對使用式樣的瞭解越多，您調校系統用於未來需求的式樣與計劃的能力越好。

影響效能的因素

訊息延時和訊息流量是兩個主要的效能指標，通常取決於一般訊息完成訊息傳送程序的各個步驟所花費的時間。以下顯示的步驟適用於持續可靠傳送訊息的情況。下圖描述這些步驟。

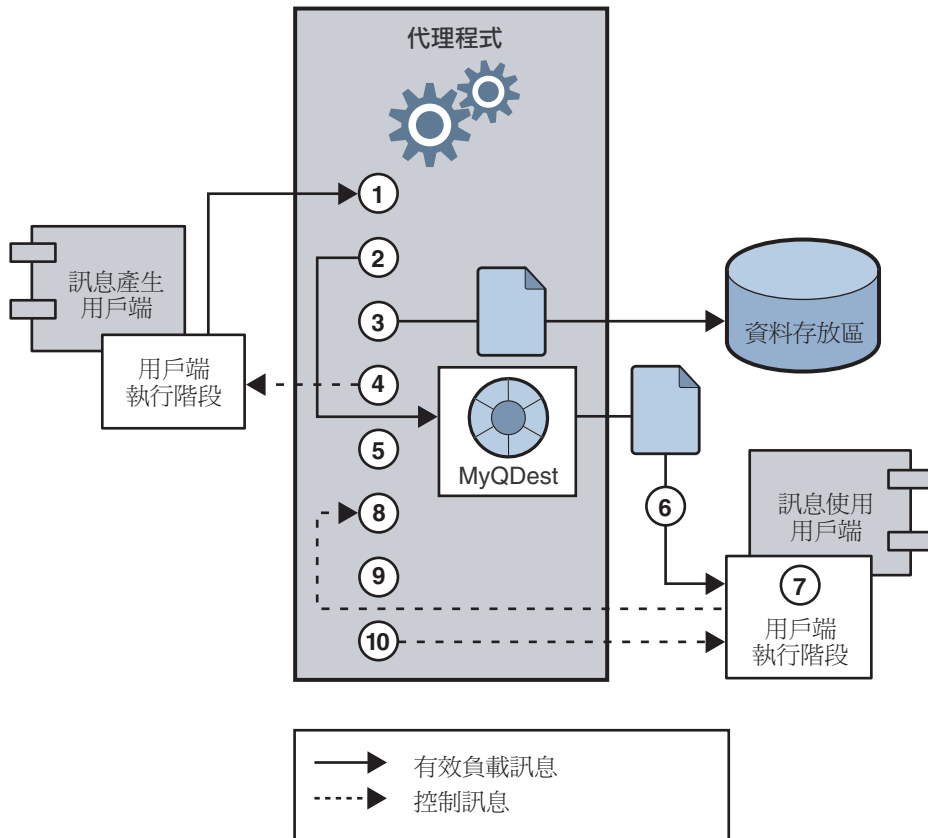


圖 11-1 透過 Message Queue 服務傳送訊息

▼ 訊息傳送步驟

- 1 訊息由訊息產生用戶端傳送到代理程式。
- 2 代理程式讀取訊息。
- 3 訊息會置於永久性存放區中，以確保其可靠性。
- 4 代理程式確認接收到訊息，以確保其可靠性。

- 5 代理程式判斷訊息的路由狀況。
- 6 代理程式寫出訊息。
- 7 訊息由代理程式傳送到訊息使用用戶端。
- 8 訊息使用用戶端確認接收到訊息，以確保其可靠性。
- 9 代理程式處理用戶端確認，以確保其可靠性。
- 10 代理程式確定已處理用戶端確認。

由於這些是連續的步驟，因此，在從訊息產生用戶端傳送訊息到使用用戶端時，其中任何步驟都可能成爲瓶頸。大部份的步驟都取決於訊息傳送系統的實體特徵：網路頻寬、電腦處理速度、訊息服務架構等。但是，還有些步驟是取決於訊息傳送應用程式的特徵和要求的可靠性層級。

以下各小節說明應用程式設計因素與訊息傳送系統因素對於效能的影響。當在訊息傳送中應用程式設計和訊息傳送系統因素相互操作時，將單獨考量每個種類。

影響效能的應用程式設計因素

應用程式設計決策會對整體訊息傳送效能產生顯著的效果。

影響訊息傳送可靠性的因素，對效能的影響最爲重要。這些因素如下：

- 第 182 頁的「傳送模式(永久性/非永久性訊息)」
- 第 182 頁的「作業事件的使用」
- 第 183 頁的「確認模式」
- 第 183 頁的「長期和非長期訂閱」

其他影響效能的應用程式設計因素如下：

- 第 184 頁的「選擇器的使用(訊息篩選)」
- 第 184 頁的「訊息容量」
- 第 184 頁的「訊息內文類型」

以下各小節說明影響訊息傳送效能的以上各個因素。一般來說，要在效能與可靠性之間進行取捨與平衡：提昇可靠性的因素往往會降低效能。

表 11-1 顯示各種應用程式設計因素通常如何影響訊息傳送效能。表格顯示兩種方案(一個可靠性高但效能低的方案，另一個效能高但可靠性低的方案)，以及每一個方案的應用程式設計因素選擇。在這兩種極端的方案之間，有多種選擇和取捨可影響可靠性及效能。

表 11-1 高可靠性和高效能方案的比較

應用程式設計因素	可靠性高但效能低的方案	效能高但可靠性低的方案
傳送模式	永久性訊息	非永久性訊息
使用作業事件	已處理的階段作業	無作業事件
確認模式	AUTO_ACKNOWLEDGE 或 CLIENT_ACKNOWLEDGE	DUPS_OK_ACKNOWLEDGE
長期/非長期訂閱	長期訂閱	非長期訂閱
選擇器的使用	訊息篩選	無訊息篩選
訊息容量	大量的小訊息	少量的大訊息
訊息內文類型	複雜的內文類型	簡單的內文類型

傳送模式 (永久性/非永久性訊息)

永久性訊息可在代理程式發生故障的情況下，確保訊息傳送。代理程式會在所有預定用戶確認他們已使用訊息前，將訊息儲存在永久性存放區中。

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：

- 代理程式必須以可靠的方式儲存永久性訊息，以便在代理程式發生故障時，才不會遺失永久性訊息。
- 代理程式必須確認每個永久性訊息的接收。一旦產生訊息的方法未傳回異常，就確保會傳送至代理程式。
- 代理程式可能需要確定訊息使用用戶端對於永久性訊息的確認，依用戶端的確認模式而定。

對於佇列以及長期訂閱者的主題而言，處理非永久性訊息的效能大約快 40%。我們是使用容量為 10k 的訊息和 AUTO_ACKNOWLEDGE 模式取得這些結果。

作業事件的使用

作業事件可確保在已處理的階段作業中產生的所有訊息以及使用的所有訊息，都會分別當成一個單位予以處理或不處理(回復)。

Message Queue 支援本機作業事件和分散式作業事件。

已處理的階段作業中產生或確認訊息的速度，會比在未處理階段作業中的速度慢，原因如下：

- 附加資訊必須隨每個產生訊息一起儲存。
- 某些狀況會儲存作業事件中通常不需要儲存的訊息(例如，傳送到未訂閱主題目標的永久性訊息通常會遭到刪除，但是在作業事件開始時，有關訂閱的資訊還無法取得)。

- 作業事件中的訊息使用和確認資訊，必須在確定作業事件時進行儲存和處理。

確認模式

一種確保 JMS 訊息可靠傳送的機制，用戶端會確認 Message Queue 代理程式所傳送的訊息已被使用。

如果在用戶端尚未確認訊息時關閉階段作業，或在處理確認前代理程式發生故障，則代理程式會重新傳送訊息，並設定 `JMSRedelivered` 旗標。

對於未處理的階段作業而言，用戶端可以選擇三種確認模式中的一種，而每一種模式都具有各自的效能特徵：

- `AUTO_ACKNOWLEDGE`。一旦用戶處理訊息後，系統會自動確認訊息。此模式可確保提供者發生故障後最多會重新傳送一次訊息。
- `CLIENT_ACKNOWLEDGE`。此應用程式會控制確認訊息的端點。確認先前的確認後，所有訊息會在該階段作業中進行處理。在處理一組確認時，如果代理程式發生故障，則會重新傳送群組中一個或多個訊息。
- `DUPS_OK_ACKNOWLEDGE`。此模式引導系統在惰性狀態下確認訊息。提供者發生故障後，可以重新傳送多個訊息。

(使用 `CLIENT_ACKNOWLEDGE` 模式與使用作業事件相似，但如果在處理期間提供者發生故障，則前者不確保會一併處理所有的確認。)

確認模式影響效能的原因如下：

- 在 `AUTO_ACKNOWLEDGE` 和 `CLIENT_ACKNOWLEDGE` 模式中，代理程式與用戶端之間需要額外的控制訊息。額外控制訊息會增加額外的處理經常性耗用時間，並且可能會干擾 JMS 有效負載訊息，因而造成處理延遲。
- 在 `AUTO_ACKNOWLEDGE` 和 `CLIENT_ACKNOWLEDGE` 模式中，用戶端必須等待代理程式確定已處理用戶端的確認，才能夠使用其他訊息。(代理程式的此項確認作業可確保代理程式不會意外重新傳送這些訊息。)
- 必須以用戶所接收全部永久性訊息的確認資訊來更新 Message Queue 永久性存放區，因此會降低效能。

長期和非長期訂閱

主題目標的訂閱者分為兩類：長期訂閱者和非長期訂閱者。

長期訂閱可提高可靠性，但會降低流量速度，原因如下：

- Message Queue 訊息服務必須永久儲存指定到每個長期訂閱的訊息清單，以便當代理程式發生故障時，可在回復後使用此清單。
- 長期訂閱的永久性訊息會永久儲存，以便於當代理程式發生故障時，仍然可以在代理程式回復後，於對應的用戶成為使用中狀態時重新傳送訊息。相反的，非長期訂閱的永久性訊息不會永久儲存(一旦代理程式故障，對應用戶的連線會中斷，而且不會重新傳送訊息)。

我們以兩個狀況比較長期訂閱者與非長期訂閱者的效能：容量為 10k 的永久性和非永久性訊息。這兩種狀況都使用 `AUTO_ACKNOWLEDGE` 確認模式。只有在永久性訊息的狀況下，長期訂閱的效能才會受到影響（總共降低了大約 30%）。

選擇器的使用 (訊息篩選)

應用程式開發者通常會將訊息組指向特定用戶。他們會藉由將每個訊息組指向唯一實體目標，或使用單一實體目標並為用戶註冊一個或多個選擇器，以達成此目的。

選擇器是一個字串，請求僅將包含符合此字串特性值的訊息傳送到特定用戶。例如，選擇器 `NumberOfOrders > 1` 僅會傳送包含 `NumberOfOrders` 特性值 2 或更高值的訊息。

使用選擇器建立用戶會降低效能（與使用多個實體目標比較），因為每個訊息都需要額外的處理。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

因為必須將更多資料從訊息產生用戶端傳送到代理程式，以及從代理程式傳送到訊息使用用戶端，而且對於永久性訊息，必須儲存較大的訊息，所以訊息容量會影響效能。

但是，藉由將較小訊息批次到單一訊息，可以最小化個別訊息的路由和處理，以改善整體效能。在此情況下，會遺失個別訊息狀態的相關資訊。

在我們的測試中，將傳送至佇列目標的容量為 1k、10k 和 100k 之訊息的流量（以 KB/秒為單位）相互比較（使用 `AUTO_ACKNOWLEDGE` 確認模式）；我們發現，非永久性訊息傳送在傳送 1k 訊息時大約快 50%，10k 訊息時大約快 20%，100k 訊息時大約快 5%。訊息容量會明顯影響永久性和非永久性訊息。100k 訊息大約是 10k 的 10 倍快，10k 大約是 1k 的 5 倍快。

訊息內文類型

JMS 支援 5 種訊息內文類型，依複雜性大致排列如下：

- `BytesMessage` 包含一組位元組，其格式由應用程式所決定。
- `TextMessage` 是一個簡單的 Java 字串。
- `StreamMessage` 包含一串 Java 原始值。
- `MapMessage` 包含一組「名稱-值」對。
- `ObjectMessage` 包含一個 Java 序列化物件。

雖然通常狀況下，使用何種訊息類型是依應用程式的需求而定，但是更複雜的類型（`MapMessage` 和 `ObjectMessage`）會增加效能耗用：因為對資料進行串列化和取消串列化作業會消耗效能。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，也受到執行訊息路由和傳送的訊息服務影響。

以下各節介紹可影響效能的這種訊息服務因素。瞭解這些因素的影響，是增減訊息服務功能、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 服務效能的最重要因素如下：

- 第 185 頁的「硬體」
- 第 185 頁的「作業系統」
- 第 185 頁的「Java 虛擬機器 (JVM)」
- 第 185 頁的「連線」
- 第 187 頁的「代理程式限制和運作方式」
- 第 187 頁的「訊息服務架構」
- 第 187 頁的「資料存放區效能」
- 第 188 頁的「用戶端執行階段配置」

以下各小節說明影響訊息傳送效能的以上各個因素。

硬體

對於 Message Queue 代理程式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也不同，因此即便是相同的硬體平台，效能也會各有差異。例如，作業系統使用的執行緒模型，對於代理程式可支援的同步運作之連線數目會有重大影響。一般而言，在全部硬體都相同的狀況下，Solaris 系統通常比 Linux 系統快，而 Linux 系統通常比 Windows 系統快。

Java 虛擬機器 (JVM)

代理程式是在主機 JVM 中執行且受其支援的 Java 處理程序。因此，JVM 處理是決定代理程式路由和傳送訊息的速度和效率的重要因素。

尤其是，JVM 的記憶體資源管理相當重要。必須為 JVM 配置足夠的記憶體，以因應增加的記憶體負載。另外，JVM 會定期收回未使用的記憶體，而記憶體收回作業會延遲訊息處理。JVM 記憶體堆疊越大，記憶體收回過程中可能的延遲時間就越長。

連線

用戶端和代理程式之間的連線數目和速度，會影響訊息服務可處理的訊息數目以及訊息傳送的速度。

代理程式連線限制

所有對代理程式的存取都透過連線進行。對於同步運作之連線數目的任何限制，都會影響目前可同時使用代理程式的訊息產生用戶端或訊息使用用戶端的數目。

與代理程式的連線數目，通常會受到可用執行緒數目的限制。您可以將 Message Queue 配置為支援專用的執行緒模型，或是支援共用的執行緒模型（請參閱第 73 頁的「執行緒池管理」）。

專用執行緒模型的執行速度非常快，因為每個連線都有專用執行緒；但是，連線數目受到可用執行緒數目的限制（每個連線都使用一個輸入執行緒和一個輸出執行緒）。共用執行緒模型在連線數目上沒有限制；但是，在連線數目超過共用執行緒時，會產生明顯的經常性耗用時間和流量延遲，特別是在連線忙碌時。

傳輸協定

Message Queue 軟體可讓用戶端使用各種低層級的傳輸協定，與代理程式進行通訊。Message Queue 支援第 71 頁的「連線服務」中描述的連線服務（以及對應的協定）。

協定是根據應用程式需求（加密、可透過防火牆存取）而選擇的，但是所選擇的結果會影響整體效能。

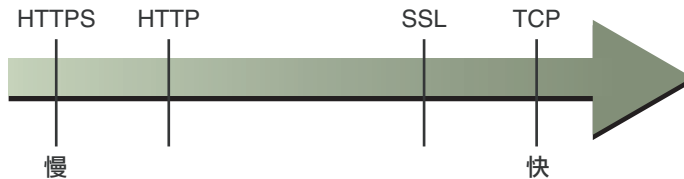


圖 11-2 傳輸協定速度

我們的測試比較了 TCP 和 SSL 在兩種狀況下的流量：高可靠性方案（將 1K 的永久性訊息傳送到包含長期訂閱的主題目標，並且使用 AUTO_ACKNOWLEDGE 確認模式），和高效能方案（將 1K 的非永久性訊息傳送到不包含長期訂閱的主題目標，並且使用 DUPS_OK_ACKNOWLEDGE 確認模式）。

一般而言，我們發現協定對於可靠性高的方案影響較小。這可能是因為，在可靠性高的方案中，所需要的永久性經常性耗用時間是限制流量的更重要因素，其影響比協定速度的影響更大。另外：

- TCP 提供與代理程式進行通訊的最快速方法。
- 傳送和接收訊息時，SSL 的速度比 TCP 慢了 50% 到 70%（如果是永久性訊息則慢了 50%；非永久訊息則慢了將近 70%）。此外，使用 SSL 建立初始連線的速度較慢（可能會花費數秒），因為用戶端和代理程式（若為 HTTPS，則是 Web 伺服器）需要建立私密金鑰以供加密所傳輸的資料時所使用。由於加密和解密每個低層級 TCP 資料封包都需要進行額外處理，因此效能會降低。

- HTTP 的速度比 TCP 或 SSL 慢。HTTP 將在 Web 伺服器上執行的 Servlet 當做用戶端與代理程式之間的代理。將資料封包封裝為 HTTP 請求，以及訊息必須通過兩個躍點 (用戶端至 Servlet、Servlet 至代理程式) 才能到達代理程式，都會影響效能的經常性耗用時間。
- HTTPS 的速度比 HTTP 慢，因為 HTTPS 需要額外的經常性耗用時間以加密用戶端與 Servlet 之間以及 Servlet 與代理程式之間的資料封包。

訊息服務架構

Message Queue 訊息服務可實作為單一代理程式，或實作為包含多個互連代理程式實例的叢集。

隨著連線至代理程式的用戶端數目增加，以及要傳送的訊息數目增加，代理程式最終會超過資源限制，例如，檔案描述元、執行緒和記憶體限制。因應負載增加的方法之一是，將多個代理程式實例增加至 Message Queue 訊息服務，從而在多個代理程式之間分發用戶端連線與訊息路由和傳送。

一般而言，如果用戶端 (尤其是訊息產生用戶端) 在叢集中均勻分布，如此延伸的效果會最好。由於經常性耗用時間受到叢集中代理程式間訊息傳送的影響，所以具有連線數目限制或訊息傳送速率限制的叢集，其表現效能可能會比單一代理程式低。

您還可以使用代理程式叢集來最佳化網路頻寬。例如，您可以在叢集中的一組遠端代理程式間使用速度較慢、距離較長的網路連結，而使用較高速度的連結將用戶端連線至其各自的代理程式實例。

如需有關叢集的更多資訊，請參閱第 9 章。

代理程式限制和運作方式

代理程式可能需要處理的訊息流量，是代理程式所支援之訊息傳送應用程式使用式樣的一個函數。然而，代理程式在資源上受到限制：記憶體、CPU 週期... 等等。因此，代理程式可能會在超過限制時當機，造成沒有回應或不穩定的狀況。

Message Queue 訊息代理程式有內建機制，可管理記憶體資源，並預防代理程式耗盡記憶體。這些機制的可配置限制包括代理程式可保留的訊息數目和訊息容量、其個別實體目標，以及達到實體目標限制時可採取的運作方式。

透過謹慎的監視與調校，使用這些可配置的機制以平衡傳入和傳出訊息的流量，這樣就不會發生系統超載的狀況。這些機制會消耗經常性耗用時間並限制訊息流量，但他們會維護作業完整性。

資料存放區效能

Message Queue 支援檔案式與 JDBC 型永久性模組。檔案式永久性使用個別的檔案來儲存永久性資料。JDBC 型永久性使用 Java 資料庫連結 (JDBC™) 介面，並且需要 JDBC 相容的資料存放區。檔案型永久性通常快於 JDBC 型永久性；但是，某些使用者偏好 JDBC 相容存放區所提供的備援和管理控制。

在使用檔案式永久性的情況下，您可以指定永久性作業同步化包含資料存放區的常駐記憶體狀態，以保持最佳可靠性。這可以消除因系統當機而產生的資料遺失，但會影響效能。

用戶端執行階段配置

Message Queue 用戶端執行階段為用戶端應用程式提供 Message Queue 訊息服務的介面。它支援用戶端將訊息傳送至實體目標和從此類目標接收訊息所需的所有作業。設定連線工廠屬性值就能夠配置用戶端執行階段，讓您控制它在改善效能與訊息流量等方面的運作方式，例如連線流量測量、用戶流量限制與連線流量限制。如需有關這些功能、以及用來配置這些功能的屬性之更多資訊，請參閱第 192 頁的「[用戶端執行階段訊息流量調校](#)」。

調校配置以改善效能

以下各小節說明配置調整如何影響效能。

系統調校

以下各小節描述您可對作業系統、JVM 和通訊協定所做的調整。

Solaris 調校：CPU 使用、分頁/交換/磁碟 I/O

請參閱您系統的文件來調校您的作業系統。

Java 虛擬機器調整

依預設，代理程式會使用 192MB 的 JVM 堆疊容量。此容量對大量的訊息負載通常過小，所以應增加其大小。

當代理程式將要耗盡 Java 物件使用的 JVM 堆疊儲存區空間時，會使用各種技術(如流量控制和訊息交換)來釋放記憶體。在極端情況下，它甚至會關閉用戶端連線以釋放記憶體，並降低訊息的傳入量。因此，最好將最大 JVM 堆疊儲存區空間設定得足夠大，以避免發生此類情況。

但是，如果將最大 Java 堆疊儲存區空間設定得過大(相對於系統實體記憶體)，代理程式會持續增加 Java 堆疊儲存區空間，直至整個系統的記憶體用完。這會導致效能降低、不可預料的代理程式當機，和/或影響系統上執行的其他應用程式和服務的運作。一般而言，您需要允許足夠用於作業系統和機器上其他執行應用程式的實體記憶體。

一般而言，最好先評估正常時段和尖峰時段的系統記憶體使用量，然後再配置 Java 堆疊容量，該容量大小應足以提供良好的效能，但又不至於大到導致系統記憶體問題。

若要變更代理程式的最小和最大堆疊容量，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 Java 堆疊容量的下限設定為 256MB，上限為 1GB。

- 在 Solaris 或 Linux 上，如果經由 `/etc/rc*` (也就是 `/etc/init.d/imq`) 啟動代理程式，則請在 `/etc/imq/imqbrokerd.conf` (Solaris) 或 `/etc/opt/sun/mq/imqbrokerd.conf` (Linux) 檔案中，指定代理程式指令行引數。請查看檔案中的註解，以取得更多資訊。
- 在 Windows 上，如果將代理程式當成 Windows 服務啟動，則請使用 `-vmargs` 選項，將 JVM 引數指定到 `imqsvcadm install` 指令。請參閱第 13 章中的第 233 頁的「服務管理員公用程式」。

在任何狀況下，都請檢查代理程式的記錄檔或使用 `imqcmd metrics bkr -m cxn` 指令，以驗證設定。

調校傳輸協定

選擇符合應用程式需求的協定後，可進行額外調校 (以選取的協定為基礎) 以改善效能。

以下 3 個代理程式特性可用來修改協定效能：

- `imq.protocol.protocolType .nodelay`
- `imq.protocol.protocolType .inbufsz`
- `imq.protocol.protocolType .outbufsz`

對於 TCP 和 SSL 協定，這些特性會影響用戶端和代理程式之間訊息傳送的速度。對於 HTTP 和 HTTPS 協定，這些特性會影響 Message Queue 通道 Servlet (在 Web 伺服器上執行) 和代理程式之間訊息傳送的速度。HTTP/HTTPS 協定還有其他會影響效能的特性 (請參閱第 190 頁的「HTTP/HTTPS 調校」)。

將會在以下各節描述協定調校特性。

nodelay

`nodelay` 特性會影響用於指定協定的 Nagle 演算法 (TCP/IP 上 `TCP_NODELAY` 傳輸層選項的值)。Nagle 演算法可用以改善使用慢速連線 (例如廣域網路 (WAN) 等) 之系統上的 TCP 效能。

使用該演算法時，TCP 會將資料包裝為較大的資料封包，從而避免傳送一些小的資料塊到遠端系統。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。一旦緩衝區填滿或發生逾時，即會傳送資料封包。

對於大部份的訊息傳送應用程式，如果傳送資料封包時沒有延遲 (未啟用 Nagle 演算法)，則效能為最佳。這是因為用戶端與代理程式之間大部份都是請求/回應式互動：用戶端傳送資料封包給代理程式且等待回應。例如，典型的互動包括：

- 建立連線
- 建立產生器或用戶
- 傳送永久性訊息 (代理程式會確定已接收訊息)
- 在 `AUTO_ACKNOWLEDGE` 或 `CLIENT_ACKNOWLEDGE` 階段作業中傳送用戶端確認 (代理程式會確定確認處理)

對於這些互動，大部份的資料封包都小於緩衝區大小。這表示，如果使用 Nagle 演算法，則代理程式在傳送回應到用戶之前，會延遲數毫秒。

但是，在連線速度緩慢和不需代理程式回應的狀況下，Nagle 演算法可改善效能。例如，在用戶端傳送非永久性訊息，或用戶端的確認未經代理程式確定 (`DUPS_OK_ACKNOWLEDGE` 階段作業) 等狀況下。

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通訊端的資料。同樣地，`outbufsz` 會設定輸出串流上的緩衝區大小，輸出串流由代理程式用來將資料寫入通訊端。

一般而言，兩個參數值皆應稍微大於接收或傳送的封包容量。根據經驗，請將這些特性值設定為比平均資料封包容量多 1 KB (四捨五入至最接近的以 KB 為單位之整數值)。例如，如果代理程式接收內文容量為 1 KB 的資料封包，則該封包 (訊息內文加上標頭和特性) 的總容量約為 1.2 KB；`inbufsz` 為 2 KB (2048 個位元組) 即可提供適當的效能。將 `inbufsz` 或 `outbufsz` 增加到超過此容量可能會稍微改善效能，但是將增加各個連線所需的記憶體。

HTTP/HTTPS 調校

除了前面兩小節描述的一般特性之外，HTTP/HTTPS 效能受限於用戶端發出 HTTP 請求到託管 Message Queue 通道 Servlet 之 Web 伺服器的速度。

Web 伺服器可能需要最佳化，以處理單一通訊端上的多個請求。從 JDK 1.4 版或更新版本開始，HTTP 與 Web 伺服器的連線會持續作用 (到 Web 伺服器的通訊端會保持開啓)，以便在 Web 伺服器處理多個 HTTP 請求時，將 Web 伺服器使用的資源降到最低。在相同用戶端應用程式上使用 JDK 1.4 版的效能，如果比使用較早 JDK 版本的效能低，則您可能需要調校 Web 伺服器持續作用的配置參數，以改善效能。

除了對 Web 伺服器進行此類調校，您也可以調整用戶端輪詢 Web 伺服器的頻率。HTTP 是請求型協定。這表示，使用 HTTP 型協定的用戶端會定期檢查 Web 伺服器，以查看是否有等待中的訊息。`imq.httpjms.http.pullPeriod` 代理程式特性 (以及類似的 `imq.httpsjms.https.pullPeriod` 特性) 會指定 Message Queue 用戶端執行階段輪詢 Web 伺服器的頻率。

如果 `pullPeriod` 值為 -1 (預設值)，則用戶端執行階段會在前一個請求傳回時立即輪詢伺服器，以便最有效發揮個別用戶端的效能。因此，各個用戶端連線均會獨占 Web 伺服器中的一個請求執行緒，因而可能會過度使用 Web 伺服器資源。

如果 `pullPeriod` 值為正數，則用戶端執行階段會定期傳送請求到 Web 伺服器，以查看是否有擱置的資料。在此狀況下，用戶端不會獨占 Web 伺服器中的請求執行緒。因此，如果大量用戶端使用 Web 伺服器，則您可能需要將 `pullPeriod` 設定為正值，以節省 Web 伺服器資源。

調校檔案式永久性存放區

如需有關調校檔案式永久性存放區的資訊，請參閱第 75 頁的「永久性服務」。

代理程式調校

以下各小節描述您可如何調整代理程式特性以改善效能。

記憶體管理：在代理程式負載過重時提高穩定性

您可以對每個目標個別配置記憶體管理，或是集中配置整個系統層級 (針對所有目標) 的記憶體管理。

使用實體目標限制

如需有關實體目標限制的資訊，請參閱第 6 章。

使用系統範圍限制

如果訊息產生器經常使訊息用戶超載，代理程式中可能會累積訊息。代理程式包含一個在記憶體不足情況下，降低產生器速度和將訊息移出使用中記憶體的機制，所以對代理程式可保留的訊息總數目 (和訊息容量) 做出硬式限制是明智之舉。

設定 `imq.system.max_count` 和 `imq.system.max_size` 代理程式特性，可控制這些限制。

例如：

```
imq.system.max_count=5000
```

以上定義的值，表示代理程式最多只會保留 5000 個未傳送/未確認的訊息。如果傳送額外訊息，則會被代理程式拒絕。如果為永久性訊息，那麼產生器在嘗試傳送訊息時會發生異常。如果為非永久性訊息，代理程式會直接丟棄訊息。

當傳送訊息時傳回異常，用戶端應稍作暫停並試著再傳送一次。(請注意，永遠不會因為代理程式接收訊息失敗而發生異常；所發生的異常都是由傳送端的用戶端所偵測到的異常。)

多重用戶佇列效能

多個佇列用戶在佇列目標中處理訊息的效率，取決於下列可配置的佇列目標屬性：

- 使用中的用戶數目 (`maxNumActiveConsumers`)
- 單一批次中可傳送到用戶的最大訊息數目 (`consumerFlowLimit`)

若要達到最佳的訊息流量，必須有充足的使用中用戶數目以滿足佇列產生訊息的速率，並且將佇列中的訊息路由並傳送到使用中用戶時，必須能讓使用的速率達到最快。「Sun Java System™ Message Queue 技術摘要」中描述平衡多個用戶上訊息傳送的一般機制。

如果訊息累積在佇列中，那麼使用中用戶數目可能會不足以處理訊息負載。也有可能發生以下情形：以批次大小傳送到用戶的訊息會在用戶上進行備份。例如，如果批次大小 (`consumerFlowLimit`) 過大，一個用戶可能會接收佇列中的所有訊息，而其他使用中用戶則收不到任何訊息。如果用戶處理速度非常快，這可能不會發生問題。

但是，如果用戶的速度相對較慢，那麼您必須將訊息平均分發給用戶，如此一來，批次大小也會變小。批次大小越小，傳送訊息至用戶所需的經常性耗用時間就越長。然而，針對處理速度慢的用戶，網路效能改善比率通常會使用小型的批次大小。

用戶端執行階段訊息流量調校

本小節將討論影響效能的流量控制運作方式 (請參閱第 188 頁的「用戶端執行階段配置」)。請將這些運作方式配置為連線工廠受管理物件的屬性。如需有關設定連線工廠屬性的資訊，請參閱第 8 章。

訊息流量計數

用戶端傳送和接收的訊息 (有效負載訊息) 與 Message Queue 控制訊息，都會透過用戶端與代理程式之間的相同連線進行傳送。如果有效負載訊息的傳送阻擋了控制訊息，則會造成控制訊息 (例如代理程式確認) 延遲傳送。若要避免此擁塞狀況，Message Queue 會計算連線上的有效負載訊息流量。

有效負載訊息分為多個批次 (如連線工廠屬性 `imqConnectionFlowCount` 所指定)，因此只傳送已設定數目的有效負載訊息。批次傳送後，會暫停傳送有效負載訊息，而只傳送擱置的控制訊息。接著傳送其他有效負載訊息，然後傳送擱置控制訊息，重複此循環。

如果用戶端正在進行需要代理程式大量回應的作業，則 `imqConnectionFlowCount` 的值應保持較低：例如，如果用戶端正在使用 `CLIENT_ACKNOWLEDGE` 或 `AUTO_ACKNOWLEDGE` 模式、永久性訊息、作業事件或佇列瀏覽器，或者正在新增或移除用戶。而另一方面，如果用戶端在連線上有簡易用戶，並且使用 `DUPS_OK_ACKNOWLEDGE` 模式，則您可以增加 `imqConnectionFlowCount` 的值，而不會影響效能。

訊息流量限制

在達到本機資源限制 (例如記憶體限制) 之前，Message Queue 用戶端執行階段可處理的有效負載訊息數目也會有所限制。到達此限制時，會對效能造成影響。因此，Message Queue 會限制每個用戶 (或每個連線) 在連線中傳送的訊息數目，以及在用戶端執行階段中進行緩衝以等待使用的訊息數目。

用戶流量限制

如果傳送到用戶端執行階段的有效負載訊息數超過了為任何用戶所設定的 `imqConsumerFlowLimit` 值，則會停止傳送訊息給該用戶。只有在傳送給該用戶的未使用訊息數低於此 `imqConsumerFlowThreshold` 設定值時，才會重新繼續傳送訊息。

以下範例說明這些限制的使用：考量主題用戶的設定值：

```
imqConsumerFlowLimit=1000
imqConsumerFlowThreshold=50
```

建立用戶時，代理程式會不間斷傳送第一批 1000 個訊息 (假設存在這些訊息) 給該用戶。傳送這 1000 個訊息之後，代理程式會停止傳送，直到用戶端執行階段要求傳送更多訊息。應用程式處理這些訊息前，用戶端執行階段會保留這些訊息。在要求代理程式傳送下一個批次之前，用戶端執行階段會允許應用程式使用至少 50% (`imqConsumerFlowThreshold`) 的訊息緩衝容量 (即 500 個訊息)。

在相同情況下，如果臨界值為 10%，則在要求傳送下一個批次之前，用戶端執行階段會等待應用程式至少使用 900 個訊息。

以下計算下一個批次的大小：

$$\text{imqConsumerFlowLimit} - (\text{current number of pending msgs in buffer})$$

因此，如果 `imqConsumerFlowThreshold` 為 50%，則下一個批次大小可介於 500 到 1000 之間，其實際數量取決於應用程式處理訊息的速度。

如果 `imqConsumerFlowThreshold` 設定過高 (接近 100%)，則代理程式會嘗試傳送較小的批次，以降低訊息流量。如果該值過低 (接近 0%)，則在代理程式傳送下一組批次之前，用戶端也許能完成處理其餘的緩衝訊息，這也會造成訊息流量下降。一般而言，除非您有特定的效能或可靠性考量，否則不需要變更 `imqConsumerFlowThreshold` 屬性的預設值。

用戶型流量控制 (尤其是 `imqConsumerFlowLimit`) 是管理用戶端執行階段中記憶體的最佳方法。通常，根據用戶端應用程式，您會知道您在任何連線需要支援的用戶數目、訊息容量和可用於用戶端執行階段的總記憶體容量。

連線流量限制

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

連線層級流量控制可限制連線上**所有**用戶的緩衝訊息總數。如果此數目超過 `imqConnectionFlowLimit` 的值，則會停止透過連線傳送訊息，直到總數目低於連線限制為止。(只有將 `imqConnectionFlowLimitEnabled` 設定為 `true`，才會啓用 `imqConnectionFlowLimit` 屬性。)

階段作業中形成佇列的訊息數目為使用階段作業之用戶數目及用於每個用戶之訊息負載的函數。如果用戶端在產生和使用訊息中發生延遲，通常您可以藉由重新設定應用程式，以分發較大階段作業數目上的訊息產生器和用戶，或者分發較大連線數目上的階段作業，並因此改善效能。

排解疑難問題

本章說明如何瞭解及解決下列問題：

- 第 195 頁的「用戶端無法建立連線」
- 第 199 頁的「連線流量過慢」
- 第 200 頁的「用戶端無法建立訊息產生器」
- 第 201 頁的「訊息產生延遲或過慢」
- 第 203 頁的「儲存訊息」
- 第 206 頁的「代理程式的流量不穩定」
- 第 207 頁的「訊息未送達用戶」
- 第 210 頁的「停用的訊息佇列包含訊息」

發生問題時，檢查所安裝 Message Queue™ 軟體的版本編號，通常很有幫助。版本編號可讓您確定所用文件的編號是否與軟體版本相符。向 Sun 報告問題時，也需要版本編號。若要檢查版本編號，請發出以下指令：

```
imqcmd -v
```

用戶端無法建立連線

徵兆：

- 用戶端無法建立新的連線。
- 用戶端無法自動重新連線失敗的連線。

可能原因：

- 用戶端應用程式未關閉連線，導致連線數目超過資源限制
- 未執行代理程式或網路連結發生問題
- 連線服務不在使用中或已暫停
- 可用於所需連線數目的可用執行緒太少
- Solaris 或 Linux 平台上要求用於連線數目的檔案描述元太少
- TCP 積存區會限制可同時新建連線請求的數目

- 作業系統限制同步運作中連線的數目
- 使用者的認證或授權失敗

可能原因：用戶端應用程式未關閉連線，導致連線數目超過資源限制。

若要確認此問題的原因：列出代理程式的所有連線：

```
imqcmd list cxn
```

輸出會列出所有連線和建立每個連線的主機，從而瞭解哪些用戶端開啓了過多的連線。

若要解決此問題：重新寫入違例用戶端，以關閉不使用的連線。

可能原因：未執行代理程式或網路連結發生問題。

若要確認此問題的原因：

- 使用 Telnet 連線至代理程式主要連接埠 (例如，預設的 7676 連接埠)，並透過連接埠對映器的輸出來驗證代理程式是否作出回應。
- 驗證代理程式程序是否在主機上執行。

若要解決此問題：

- 啓動代理程式。
- 修復網路連結問題。

可能原因：連線服務不在使用中或已暫停。

若要確認此問題的原因：檢查所有連線服務的狀態：

```
imqcmd list svc
```

如果連線服務狀態顯示為 `unknown` 或 `paused`，則用戶端無法使用該服務建立連線。

若要解決此問題：

- 如果連線服務狀態顯示為 `unknown`，此服務不會顯示在使用中服務的清單 (`imq.service.active`) 上。在 SSL 型服務中，服務也可能會配置錯誤，造成代理程式在代理程式記錄中產生以下項目：

```
ERROR [B3009]: Unable to start service ssljms: [B4001]: Unable to open  
protocol tls for ssljms service...
```

此項目後面是發生異常的基本原因說明。

若要正確配置 SSL 服務，請參閱第 130 頁的「訊息加密」。

- 如果連線服務狀態顯示為 `paused`，請重新繼續該服務 (請參閱第 99 頁的「暫停和重新繼續連線服務」)。

可能原因：可用於所需連線數目的可用執行緒太少。

若要確認此問題的原因：檢查代理程式記錄中的以下項目：

```
WARNING [B3004]: No threads are available to process a new connection on service  
... Closing the new connection.
```

並且使用下列其中一種格式，檢查連線服務上的連線數目，以及目前使用中的執行緒數目：

```
imqcmd query svc -n serviceName imqcmd metrics svc -n serviceName -m cxn
```

每個連線均需要 2 個執行緒：一個用於內送訊息，另一個用於外寄訊息 (請參閱第 73 頁的「執行緒池管理」)。

若要解決此問題：

- 如果您使用專用執行緒池模型 (`imq.serviceName.threadpool_model=dedicated`)，那麼最大的連線數目是執行緒池中最大執行緒數目的一半。因此，若要增加連線數目，請增加執行緒池的容量 (`imq.serviceName.max_threads`)，或改為使用共用執行緒池模型。
- 如果您使用共用執行緒池模型 (`imq.serviceName.threadpool_model=shared`)，那麼最大的連線數目是連線監視限制 (`imq.serviceName.connectionMonitor_limit`) 與最大執行緒數目 (`imq.serviceName.max_threads`) 積的一半。因此，若要增加連線數目，請增加執行緒池的大小或增加連線監視限制。
- 可支援的連線數目 (或連線上的流量) 最終會達到輸入/輸出限制。在此情況下，請使用多代理程式叢集，以分發叢集中代理程式實例上的連線。

可能原因： Solaris 或 Linux 平台上要求用於連線數目的檔案描述元太少。

如需有關此問題的更多資訊，請參閱第 64 頁的「設定檔案描述元限制」。

若要確認此問題的原因： 檢查代理程式記錄中是否有以下相似項目：

Too many open files

若要解決此問題： 增加檔案描述元限制，如 `ulimit` 線上手冊所述。

可能原因： TCP 積存區會限制可同時新建連線請求的數目。

TCP 積存區會限制同時儲存在系統積存區內的連線請求數目 (`imq.portmapper.backlog`)，連接埠對映器會在超過此限制之後拒絕其他請求。(Windows 平台上有程序內定的積存區限制：Windows 個人版作業系統的數目限制為 5 個，Windows 伺服器版作業系統的數目為 200 個。)

因為不會經常出現同時有大量的連線請求，所以因儲存區限制的請求拒絕通常為短暫現象。

若要確認此問題的原因： 檢查代理程式記錄。首先，檢查代理程式在拒絕其他連線的同時是否有接受某些連線。接著，檢查拒絕連線的說明訊息。如果找到這類訊息，可能不是 TCP 積存區的問題，因為代理程式不會記錄 TCP 積存區造成的拒絕連線。如果記錄中有成功的連線，但是未記錄任何拒絕連線，則可能是 TCP 積存區的問題。

若要解決此問題：

- 將用戶端設定成每隔一小段時間就重試連線 (此方法通常可行，因為這是暫時性的問題)。
- 增加 `imq.portmapper.backlog` 的值。
- 檢查用戶端關閉和開啓連線的次數是否過於頻繁。

可能原因：作業系統限制同步運作中連線的數目。

Windows 作業系統授權會限制所支援的同步運作遠端連線的數目。

若要確認此問題的原因：檢查可用於連線的執行緒數目是否夠用 (使用 `imqcmd query svc`)，並檢查 Windows 授權合約的條款。如果您是從本地用戶端建立連線，而不是從遠端用戶端，作業系統限制可能是造成此問題的原因。

若要解決此問題：

- 升級 Windows 授權，以允許更多的連線。
- 設定多代理程式叢集，以分發一些代理程式實例的連線。

可能原因：使用者的認證或授權失敗。

認證可能因為下列任一原因而失敗：

- 密碼不正確
- 使用者儲存庫中沒有使用者的項目
- 使用者沒有存取連線服務的權限

若要確認此問題的原因：檢查代理程式記錄中是否有 `Forbidden` 錯誤訊息的項目。這表示有驗證錯誤，但不表示為問題發生的原因。

- 如果您使用檔案式使用者儲存庫，請輸入以下指令：

```
imqusermgr list -i instanceName -u userName
```

如果輸出顯示有使用者，則可能是提交了錯誤的密碼。如果輸出顯示下列錯誤，則使用者儲存庫中沒有該使用者的任何項目：

```
Error [B3048]: User does not exist in the password file
```

- 如果您使用 LDAP 伺服器使用者儲存庫，請使用適當的工具檢查是否有使用者的任何項目。
- 檢查存取控制特性檔案，查看是否有存取連線服務的限制。

若要解決此問題：

- 如果使用的密碼錯誤，請提供正確的密碼。
- 如果使用者儲存庫中沒有使用者的任何項目，請增加一個項目 (請參閱第 121 頁的「寫入和管理使用者儲存庫」)。
- 如果使用者沒有存取連線服務的權限，請編輯存取控制特性檔案以授予這類權限 (請參閱第 127 頁的「連線服務的存取控制」)。

連線流量過慢

徵兆：

- 訊息流量未符合預期速度。
- 支援連線至代理程式的數目並未如第 195 頁的「用戶端無法建立連線」中所述受到限制，而是因為訊息輸入/輸出速率受到限制。

可能原因：

- 網路連線或 WAN 太慢
- 連線服務協定的速度原本就比 TCP 的速度慢
- 連線服務協定未調校為最佳效能
- 訊息過大導致使用過多的頻寬
- 實際上，連線流量過慢的問題，是訊息傳送程序的某些步驟中會遇到的瓶頸

可能原因：網路連線或 WAN 太慢。

若要確認此問題的原因：

- Ping 網路以查看 Ping 指令傳回所需的時間，並洽詢網路管理員。
- 使用本地用戶端傳送和接受訊息，並且與遠端用戶端 (使用網路連結) 比較傳送時間。

若要解決此問題：升級網路連結。

可能原因：連線服務協定的速度原本就比 TCP 的速度慢。

例如，SSL 型或 HTTP 型協定的速度會比 TCP 的速度慢 (請參閱第 186 頁的「傳輸協定」)。

若要確認此問題的原因：如果您使用 SSL 型或 HTTP 型協定，請嘗試使用 TCP 並比較傳送時間。

若要解決此問題：通常，應用程式需求決定要使用的協定，所以除了如第 189 頁的「調校傳輸協定」中所述嘗試調校協定效能外，沒有更多的辦法。

可能原因：連線服務協定未調校為最佳效能。

若要確認此問題的原因：嘗試調校協定，並查看是否有任何差異。

若要解決此問題：嘗試調校協定，如第 189 頁的「調校傳輸協定」中所述。

可能原因：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：

- 請應用程式開發者修改應用程式以使用訊息壓縮功能，如「Message Queue Developer's Guide for Java Clients」中所述。
- 將訊息用作要傳送的資料通知，但使用其他協定移動資料。

可能原因：實際上，連線流量過慢的問題，是訊息傳送程序的某些步驟中會遇到的瓶頸。

若要確認此問題的原因：如果上述原因都不是造成連線流量速度過慢的原因，則請參閱第 180 頁的「影響效能的因素」，找出其他可能的瓶頸，並檢查是否有任何與以下問題相關的徵兆：

- 第 201 頁的「訊息產生延遲或過慢」
- 第 203 頁的「儲存訊息」
- 第 206 頁的「代理程式的流量不穩定」

若要解決此問題：依照上述疑難排解各節提供的問題解決方案進行排解。

用戶端無法建立訊息產生器

徵兆：

- 無法建立用於實體目標的訊息產生器；用戶端接收到異常。

可能原因：

- 已將實體目標配置為僅允許有限數目的產生器
- 存取控制特性檔案中的設定未授權使用者建立訊息產生器

可能原因：已將實體目標配置為僅允許有限數目的產生器。

為了防止在實體目標上累積訊息，方法之一就是限制實體目標可支援的產生器數目 (maxNumProducers)。

若要確認此問題的原因：檢查實體目標：

```
imqcmd query dst
```

(請參閱第 108 頁的「顯示實體目標資訊」)。輸出會顯示目前的產生器數目和 maxNumProducers 的值。如果兩個值相同，表示產生器數目到達配置的限制。代理程式拒絕新的產生器時，該代理程式會傳回異常

```
ResourceAllocationException [C4088]: A JMS destination limit was reached
```

並且在代理程式記錄中產生以下項目：

```
[B4183]: Producer can not be added to destination
```

若要解決此問題：增加 maxNumProducers 屬性的值 (請參閱第 110 頁的「更新實體目標特性」)。

可能原因：存取控制特性檔案中的設定未授權使用者建立訊息產生器。

若要確認此問題的原因：代理程式拒絕新的產生器時，代理程式會傳回異常

```
JMSSecurityException [C4076]: Client does not have permission to create producer on destination
```

並且在代理程式記錄中產生以下項目：

```
[B2041]: Producer on destination denied[B4051]: Forbidden 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 128 頁的「實體目標的存取控制」)。

訊息產生延遲或過慢

徵兆：

- 傳送永久性訊息時，send方法不會傳回，而且用戶端會阻斷。
- 傳送永久性訊息時，用戶端接收到異常。
- 訊息產生用戶端的速度變慢。

可能原因：

- 訊息積存於代理程式上，因此回應訊息產生器的速度會變慢
- 代理程式無法將永久性訊息儲存到資料存放區
- 代理程式回應逾時過短
- 訊息產生用戶端受到 JVM 限制

可能原因：訊息積存於代理程式上，因此回應訊息產生器的速度會變慢。

積存的代理程式會將訊息累積在代理程式記憶體中。實體目標記憶體中的訊息數目或訊息容量到達所配置的限制時，代理程式會依指定的限制運作方式，嘗試節省記憶體資源。以下限制運作方式會降低訊息產生器的速度：

- FLOW_CONTROL：代理程式不會立即確認已收到永久性訊息 (因此會阻斷訊息產生用戶端)。
- REJECT_NEWEST：代理程式拒絕新的永久性訊息。

同樣地，整個代理程式記憶體 (用於所有實體目標) 中的訊息數目或訊息容量到達所配置的限制時，代理程式會拒絕最新的訊息，以嘗試節省記憶體資源。再者，到達系統記憶體限制時，因為實體目標或整個代理程式的限制未正確設定，所以代理程式會採取愈來愈重要的動作，以防記憶體超載。這些動作包括減低訊息產生器的速度。

若要確認此問題的原因：當代理程式按照所配置的訊息限制來拒絕訊息時，代理程式會傳回異常

JMSEException [C4036]: A server error occurred

並且在代理程式記錄中產生以下項目：

[B2011]: Storing of JMS message from IMQconn failed

此訊息後面還有一則訊息指出已經達到限制：

[B4120]: Cannot store message on destination *destName* because capacity of *maxNumMsgs* would be exceeded.

這是超出實體目標訊息限制時的狀況；或者

[B4024]: The maximum number of messages currently in the system has been exceeded, rejecting message.

這是超出整個代理程式訊息限制時的狀況。

一般而言，您可以在遭到拒絕前，依以下方式檢查訊息限制條件：

- 查詢實體目標和代理程式，以檢查兩者所配置的訊息限制設定。
- 使用適當的 `imqcmd` 指令，監視目前實體目標上或整個代理程式範圍內的訊息數目或訊息容量。如需可監視的度量和取得度量之指令的詳細資訊，請參閱第 18 章。

若要解決此問題：

- 請小心修改實體目標 (或整個代理程式) 的訊息限制，不要超過記憶體資源。
一般來說，您應該在個別目標層級上管理記憶體，如此便可確保不會達到整個代理程式的訊息限制。如需更多資訊，請參閱第 191 頁的「代理程式調校」。
- 變更目標上的限制運作方式，使其在到達訊息限制時不會降低訊息產生速度，而是丟棄記憶體中的訊息。
例如，您可以指定 `REMOVE_OLDEST` 和 `REMOVE_LOW_PRIORITY` 限制運作方式，以刪除記憶體中累積的訊息 (請參閱表 15-1)。

可能原因：代理程式無法將永久性訊息儲存到資料存放區。

如果代理程式無法存取資料存放區或將永久性訊息寫入資料存放區，則表示訊息產生用戶端被阻斷。這種情況亦會在目標或整個代理程式訊息到達限制時發生，如上所述。

若要確認此問題的原因：如果代理程式無法寫入資料存放區，則會在代理程式記錄中產生以下其中一個項目：

```
[B2011]: Storing of JMS message from connectionID failed [B4004]: Failed to persist message messageID
```

若要解決此問題：

- 若使用檔案式永久性，請嘗試增加檔案式資料存放區的磁碟空間。
- 若使用 JDBC 相容資料存放區，請檢查是否已正確配置 JDBC 型永久性 (請參閱第 84 頁的「配置永久性資料存放區」)。如果已正確配置，請洽詢您的資料庫管理者以排解其他資料庫問題。

可能原因：代理程式回應逾時過短。

由於連線速度慢或代理程式緩慢 (由於高 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確認永久性訊息的接收，所需要的時間會超過連線工廠 `imqAckTimeout` 屬性所允許的值。

若要確認此問題的原因：如果超過 `imqAckTimeout` 值，代理程式會傳回異常

```
JMSEException [C4000]: Packet acknowledge failed
```

若要解決此問題：變更 `imqAckTimeout` 連線工廠屬性的值 (請參閱第 147 頁的「可靠性與流量控制」)。

可能原因：訊息產生用戶端受到 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的錯誤。
- 使用執行階段方法檢查 JVM 堆疊中的可用記憶體，例如：`freeMemory`、`maxMemory`和`totalMemory`。

若要解決此問題：調整 JVM (請參閱第 188 頁的「Java 虛擬機器調整」)。

儲存訊息

徵兆：

- 訊息產生延遲或代理程式拒絕產生的訊息。
- 訊息需要超長的時間才能到達用戶。
- 代理程式 (或特定目標) 中的訊息數目或訊息容量隨著時間穩定增加。

若要查看是否累積訊息，請檢查代理程式中的訊息數目或訊息容量如何不斷變化，並與配置的限制進行比較。請先檢查配置的限制：

```
imqcmd query bkr
```

備註 – `imqcmd metrics bkr` 子指令不會顯示此資訊。

接著檢查每個目標中的訊息累積：

```
imqcmd list dst
```

若要查看訊息是否超過所配置的目標或整個代理程式的限制，請檢查代理程式記錄中是否有以下項目：

```
[B2011]: Storing of JMS message from ... failed.
```

此項目後面還有其他項目，說明已超過限制。

可能原因：

- 主題目標上有非使用中的長期訂閱
- 佇列中可使用訊息的用戶太少
- 訊息用戶處理速度過慢，無法跟上訊息產生器的速度
- 用戶端確認處理會降低訊息使用速度
- 代理程式無法跟上產生訊息的速度
- 用戶端程式碼缺點；用戶不確認訊息

可能原因：主題目標上有非使用中的長期訂閱。

如果長期訂閱為非使用中，則訊息會儲存在目標中，直到對應的用戶變為使用中狀態並能使用這些訊息為止。

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：

- 清除違例長期訂閱的所有訊息 (請參閱第 101 頁的「管理長期訂閱」)。
- 指定用於主題的訊息限制和限制運作方式屬性 (請參閱表 15-1)。例如，您可以指定 REMOVE_OLDEST 和 REMOVE_LOW_PRIORITY 限制運作方式，以刪除記憶體中累積的訊息。
- 從對應的目標清除所有訊息 (請參閱第 111 頁的「清除實體目標」)。
- 重新寫入訊息產生用戶端以設定每個訊息的存在時間值，來限制訊息可保留在記憶體中的時間。您可以置換共用一條連線之所有產生器的任何此類設定值，方法為設定 `imqOverrideJMSEExpiration` 和 `imqJMSEExpiration` 連線工廠屬性 (請參閱第 269 頁的「訊息標頭置換」)。

可能原因：佇列中可使用訊息的用戶太少。

如果可接收所傳送訊息的使用中用戶太少，則訊息累積時可能會積存在佇列目標中。此情況會因以下任何一種原因而發生：

- 目標中存在的使用中用戶太少。
- 訊息使用用戶端建立連線失敗。
- 沒有使用中用戶使用符合佇列中訊息的選擇器。

若要確認此問題的原因：請檢查目標上使用中用戶的數目，以便幫助您判斷用戶為何無法使用：

```
imqcmd metrics dst -n destName -t q -m con
```

若要解決此問題：根據無法使用用戶的原因：

- 啟動其他訊息使用用戶端，為佇列建立更多使用中用戶。
- 調整 `imq.consumerFlowLimit` 代理程式特性，使得佇列傳送訊息至多個用戶的效能最佳化 (請參閱第 191 頁的「多重用戶佇列效能」)。
- 指定用於佇列的訊息限制和限制運作方式屬性 (請參閱表 15-1)。例如，您可以指定 REMOVE_OLDEST 和 REMOVE_LOW_PRIORITY 限制運作方式，以刪除記憶體中累積的訊息。
- 從對應的目標清除所有訊息 (請參閱第 111 頁的「清除實體目標」)。
- 重新寫入訊息產生用戶端以設定各個訊息的存在時間值，來限制訊息可保留在記憶體中的時間。您可以置換用於共用連線之所有產生器的任何此類設定值，方法為設定 `imqOverrideJMSEExpiration` 和 `imqJMSEExpiration` 連線工廠屬性 (請參閱第 269 頁的「訊息標頭置換」)。

可能原因：訊息用戶處理速度過慢，無法跟上訊息產生器的速度。

在此狀況下，主題訂閱者或佇列接收者使用訊息的速度，會比產生器傳送訊息的速度慢。因為此不平衡現象，所以一個或多個目標中會出現訊息積存。

若要確認此問題的原因：檢查傳入和傳出代理程式之訊息流量的速率：

```
imqcmd metrics bkr -m rts
```

然後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：

- 最佳化訊息使用用戶端程式碼。
- 增加用於佇列目標的使用中用戶數目 (請參閱第 191 頁的「多重用戶佇列效能」)。

可能原因：用戶端確認處理會降低訊息使用速度。

有兩個因素會影響用戶端確認處理：

- 處理用戶端確認時會大量使用代理程式資源。因此，使用這些確認模式可能會降低訊息使用的速度，因為訊息使用用戶端會阻斷，直到代理程式確定用戶端確認為止。
- JMS 有效負載訊息和 Message Queue 控制訊息 (例如用戶端確認) 共用同一個連線。因此，JMS 有效負載訊息可能會阻擋控制訊息，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，則用戶端確認可能有問題。
- 檢查用戶端是否接收到下列異常：

```
JMSEException [C4000]: Packet acknowledge failed
```

若要解決此問題：

- 修改用戶端所使用的確認模式：例如，改為使用 DUPS_OK_ACKNOWLEDGE 或 CLIENT_ACKNOWLEDGE。
- 如果使用 CLIENT_ACKNOWLEDGE 或已處理的階段作業，請將更大數目的訊息集成單一確認的群組。
- 調整用戶和連線流量控制參數 (請參閱第 192 頁的「用戶端執行階段訊息流量調校」)。

可能原因：代理程式無法跟上產生訊息的速度。

在此狀況下，訊息傳入代理程式的速度，會比代理程式路由和派送訊息到用戶的速度快。代理程式運作遲緩可以是因為以下任何或所有限制造成：

- CPU
- 網路通訊端讀取/寫入作業
- 磁碟讀取/寫入作業
- 記憶體分頁
- 永久性存放區
- JVM 記憶體限制

若要確認此問題的原因：檢查是否有其他可能造成此問題的原因。

若要解決此問題：

- 升級您電腦或資料存放區的速度。

- 使用代理程式叢集，分發一些代理程式實例上的負載。

可能原因：用戶端程式碼缺點：用戶不確認訊息。

在所有要接收所傳送之訊息的用戶確認訊息之前，這些訊息會被保留在目標中。如果用戶端不確認使用的訊息，那麼訊息會累積在目標中且不被刪除。

舉例來說，用戶端程式碼可能會有以下缺點：

- 使用 CLIENT_ACKNOWLEDGE 確認模式或已處理的階段作業的用戶，可能不會定期呼叫 `Session.acknowledge` 或 `Session.commit`。
- 使用 AUTO_ACKNOWLEDGE 確認模式的用戶可能因某些原因而當機。

若要確認此問題的原因：首先，檢查本節中列出的所有其他可能原因。然後，使用下列指令，列出目標：

```
imqcmd list dst
```

請注意 `UnAcked` 標頭下列出的訊息數目，是否與目標中的訊息數目相同。此標頭下的訊息已傳送到用戶，但是尚未經過確認。如果這個數目與訊息總數相同，表示代理程式已經送出所有訊息，正在等待確認。

若要解決此問題：請應用程式開發者協助除錯此問題。

代理程式的流量不穩定

徵兆：

- 訊息流量偶爾會降低，接著又恢復正常效能。

可能原因：

- 代理程式的記憶體資源非常少
- 正在執行 JVM 記憶體收回 (廢棄項目收集)
- JVM 使用及時 (Just-In-Time) 編譯器以提高效能

可能原因：代理程式的記憶體資源非常少。

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嚴格的動作，以防記憶體超載，這會導致在清除積存的訊息前，代理程式運作緩慢。

若要確認此問題的原因：檢查記憶體不足狀況下的代理程式記錄

```
[B1089]: In low memory condition, broker is attempting to free up resources
```

後面還有一個項目，描述新的記憶體狀態和所使用的記憶體總數。也請檢查 JVM 堆疊中可用的記憶體：

```
imqcmd metrics bkr -m cxn
```

JVM 記憶體總數接近 JVM 記憶體最大值時，可用記憶體會不足。

若要解決此問題：

- 調整 JVM (請參閱第 188 頁的「Java 虛擬機器調整」)。

- 增加系統交換空間。

可能原因：正在執行 JVM 記憶體收回 (廢棄項目收集)。

記憶體收回程序會定期檢查整個系統，以釋放記憶體。發生此情況時，會阻斷所有執行緒。釋放出來的記憶體數量越多，JVM 堆疊的容量越大，因記憶體收回而導致的延遲時間就越長。

若要確認此問題的原因：監視電腦的 CPU 使用率。執行記憶體收回時，會大幅降低 CPU 使用率。

另外，請使用以下指令行選項以啟動代理程式：

```
- vmargs -verbose:gc
```

標準輸出會指出執行記憶體收回的時間。

若要解決此問題：在具有多個 CPU 的電腦中，設定同時執行記憶體收回：

```
-XX:+UseParallelGC=true
```

可能原因：JVM 使用及時 (Just-In-Time) 編譯器以提高效能。

若要確認此問題的原因：檢查是否有其他可能造成此問題的原因。

若要解決此問題：讓系統執行一段時間，效能應會改善。

訊息未送達用戶

徵兆：

- 用戶未收到產生器傳送的訊息。

可能原因：

- 限制運作方式導致訊息在代理程式上被刪除
- 訊息的逾時值過期
- 時鐘不同步
- 訊息使用用戶端無法啟動連線上的訊息傳送

可能原因：限制運作方式導致訊息在代理程式上被刪除。

目標記憶體中的訊息數目或訊息容量達到所配置的限制時，代理程式會嘗試節省記憶體資源。達到這些限制時，代理程式會採用 3 個可配置的運作方式，並因此導致訊息遺失：

- REMOVE_OLDEST：刪除最舊的訊息。
- REMOVE_LOW_PRIORITY：根據訊息存在時間，刪除優先權最低的訊息。
- REJECT_NEWEST：拒絕新的永久性訊息。

若要確認此問題的原因：如第 210 頁的「停用的訊息佇列包含訊息」中所描述，檢查停用的訊息佇列。尤其必須遵照「訊息數目或訊息容量超出目標限制」中的指示。尋找 REMOVE_OLDEST 或 REMOVE_LOW_PRIORITY 原因。

若要解決此問題：增加目標限制。例如：

```
imqcmd update dst -n MyDest -o maxNumMsgs=1000
```

可能原因：訊息的逾時值過期。

代理程式會刪除逾時值過期的訊息。如果目標積存了太多的訊息，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：使用 QBrower 示範應用程式，查看停用的訊息佇列內容及訊息是否逾時。如需 QBrower 示範程式在特定平台上的位置，請參閱附錄 A，查看表格中的「應用程式範例與位置」。

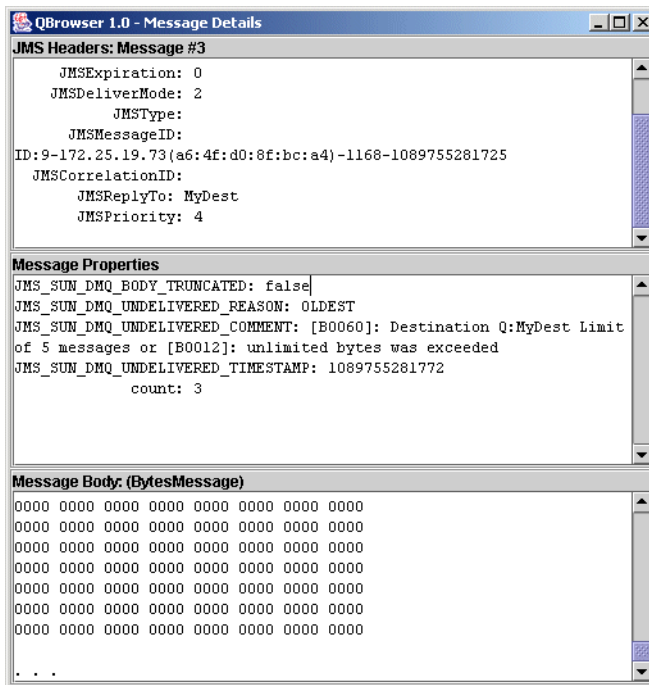
以下是 Windows 平台上的呼叫範例：

```
cd \MessageQueue3\demo\applications\qbrower java QBrower
```

QBrower 主視窗出現時，請選擇佇列名稱 mq.sys.dmq，然後按一下 [瀏覽]。會出現類似下面的清單：

#	Timestamp	Type	Mode	Priority
0	13Jul/2004:14:48:01 PDT	BytesMessage	P	4
1	13Jul/2004:14:48:01 PDT	BytesMessage	P	4
2	13Jul/2004:14:48:01 PDT	BytesMessage	P	4
3	13Jul/2004:14:48:01 PDT	BytesMessage	P	4
4	13Jul/2004:14:48:01 PDT	BytesMessage	P	4
5	13Jul/2004:14:48:01 PDT	BytesMessage	P	4
6	13Jul/2004:14:48:01 PDT	BytesMessage	P	4
7	13Jul/2004:14:48:01 PDT	BytesMessage	P	4
8	13Jul/2004:14:48:01 PDT	BytesMessage	P	4
9	13Jul/2004:14:48:01 PDT	BytesMessage	P	4
10	13Jul/2004:14:48:01 PDT	BytesMessage	P	4
11	13Jul/2004:14:48:01 PDT	BytesMessage	P	4
12	13Jul/2004:14:48:01 PDT	BytesMessage	P	4
13	13Jul/2004:14:48:01 PDT	BytesMessage	P	4
14	13Jul/2004:14:48:01 PDT	BytesMessage	P	4
15	13Jul/2004:14:48:01 PDT	BytesMessage	P	4
16	13Jul/2004:14:48:01 PDT	BytesMessage	P	4
17	13Jul/2004:14:48:01 PDT	BytesMessage	P	4
18	13Jul/2004:14:48:01 PDT	BytesMessage	P	4
19	13Jul/2004:14:48:01 PDT	BytesMessage	P	4
20	13Jul/2004:14:53:50 PDT	BytesMessage	P	4
21	13Jul/2004:14:53:50 PDT	BytesMessage	P	4
22	13Jul/2004:14:53:50 PDT	BytesMessage	P	4
23	13Jul/2004:14:53:50 PDT	BytesMessage	P	4
24	13Jul/2004:14:53:50 PDT	BytesMessage	P	4
25	13Jul/2004:14:53:50 PDT	BytesMessage	P	4
26	13Jul/2004:14:53:50 PDT	BytesMessage	P	4
27	13Jul/2004:14:53:50 PDT	BytesMessage	P	4
28	13Jul/2004:14:53:50 PDT	BytesMessage	P	4

在任何訊息上連按兩下，即可顯示此訊息的詳細資訊。



請注意訊息 `JMS_SUN_DMQ_UNDELIVERED_REASON` 特性的值是否為 `EXPIRED`。

若要解決此問題：連絡應用程式開發者並增加使用期限值。

可能原因：時鐘不同步。

如果時鐘不同步，代理程式計算訊息的使用期限會發生錯誤，如此會導致訊息逾時並被刪除。

若要確認此問題的原因：查看代理程式記錄檔中有無下列訊息：`B2102`、`B2103`、`B2104`。這些訊息都報告偵測到時鐘可能不準。

若要解決此問題：如第 63 頁的「準備系統資源」中所述，檢查有無執行時間同步化程式。

可能原因：訊息使用用戶端無法啟動連線上的訊息傳送。

在用戶端程式碼建立連線並啟動該連線上的訊息傳送之前，將無法傳送訊息。

若要確認此問題的原因：檢查用戶端程式碼是否建立連線並啟動訊息傳送。

若要解決此問題：重新寫入用戶端程式碼，以建立連線並啟動訊息傳送。

停用的訊息佇列包含訊息

徵兆：

- 您列出目標時，發現停用的訊息佇列中包含訊息。例如，發出下列指令：

```
imqcmd list dst
```

提供使用者名稱和密碼之後，出現類似下面的輸出：

Listing all the destinations on the broker specified by:

```
-----
Host          Primary Port
-----
localhost     7676
-----
Name          Type      State   Producers  Consumers  Msgs
              Total    Count  UnAck    Avg Size
-----
MyDest        Queue    RUNNING  0          0          5        0        1177.0
mq.sys.dmqs   Queue    RUNNING  0          0          35       0        1422.0
Successfully listed destinations.
```

在此範例中，停用的訊息佇列 `mq.sys.dmqs` 包含 35 個訊息。

可能原因：

- 訊息數目或訊息容量超出目標限制
- 代理程式時鐘和產生器時鐘不同步
- 用戶在訊息逾時前一直未收到訊息
- 產生器數目對於用戶數目而言過多
- 產生器的速度比用戶快
- 用戶的速度太慢
- 用戶端未確定訊息
- 用戶無法確認訊息
- 長期用戶處於非使用中狀態
- 發生非預期的代理程式錯誤

可能原因：訊息數目或訊息容量超出目標限制。

若要確認此問題的原因：使用 QBrower 示範應用程式，查看停用的訊息佇列內容。如需 QBrower 示範程式在特定作業系統上的位置，請參閱附錄 A，查看表格中的「應用程式範例與位置」。

以下是 Windows 平台的呼叫範例：

```
cd \MessageQueue3\demo\applications\qbrower java QBrower
```

QBrower 主視窗出現時，請選擇佇列名稱 `mq.sys.dmqs`，然後按一下 [瀏覽]。出現的清單與先前在「訊息的逾時值過期」下顯示的清單類似。連按兩下任何訊息，以顯示此訊息的詳細資訊，如「訊息的逾時值過期」下所示。

請注意下列訊息特性值：

- JMS_SUN_DMQ_UNDELIVERED_REASON
- JMS_SUN_DMQ_UNDELIVERED_COMMENT
- JMS_SUN_DMQ_UNDELIVERED_TIMESTAMP

請注意 JMS Headers 下的 JMSDestination 值，以便判斷將要停用其訊息的目標。

若要解決此問題：增加目標限制。例如：

```
imqcmd update dst -n MyDest -o maxNumMsgs=1000
```

可能原因：代理程式時鐘和產生器時鐘不同步。

若要確認此問題的原因：使用 QBrowser 應用程式，檢視停用的訊息佇列中訊息的詳細資訊。檢查 JMS_SUN_DMQ_UNDELIVERED_REASON 的值，尋找原因為 EXPIRED 的訊息。

查看代理程式記錄檔中有無下列訊息：B2102、B2103、B2104。這些訊息都報告偵測到時鐘可能不準。

若要解決此問題：如第 63 頁的「準備系統資源」中所述，檢查有無執行時間同步化程式。

可能原因：用戶在訊息逾時前一直未收到訊息。

若要驗證此問題的原因：使用 QBrowser 應用程式，檢視停用的訊息佇列中訊息的詳細資訊。檢查 JMS_SUN_DMQ_UNDELIVERED_REASON 的值，尋找原因為 EXPIRED 的訊息。

檢查目標上有無任何用戶。例如：

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

檢查列出的 Current Number of Active Consumers 值。如果有使用中的用戶，下列其中一項為 true：

- 用戶連線暫停。
- 訊息逾時過短，無法配合用戶的執行速度。

若要解決此問題：請應用程式開發者增加訊息存在時間值。

可能原因：產生器數目對於用戶數目而言過多。

若要確認此問題的原因：使用 QBrowser 應用程式，檢視停用的訊息佇列中訊息的詳細資訊。檢查 JMS_SUN_DMQ_UNDELIVERED_REASON 的值。如果原因是 REMOVE_OLDEST 或 REMOVE_LOW_PRIORITY，請使用 imqcmd query dst 指令，檢查目標上的產生器數目和用戶數目。如果產生器數目超過用戶數目，產生速率可能高於使用速率。

若要解決此問題：使用如下所示的指令，增加更多的訊息使用用戶端，或將目標的限制運作方式設定為 FLOW_CONTROL (以使用速率控制產生速率)：

```
imqcmd update dst -n myDst -t q -o consumerFlowLimit=FLOW_CONTROL
```

可能原因：產生器的速度比用戶快。

若要確認此問題的原因：若要判斷是否因為用戶速度慢而導致產生器的速度變慢，請使用如下所示的指令，將目標的限制運作方式設定為 FLOW_CONTROL (以使用速率控制產生速率)：

```
imqcmd update dst -n myDst -t q -o consumerFlowLimit=FLOW_CONTROL
```

利用如下所示的指令，使用度量來檢查目標的輸入與輸出：

```
imqcmd metrics dst -n myDst -t q -m rts
```

在度量輸出中，檢查下列的值：

- **Msgs/sec Out**：顯示代理程式每秒移除多少訊息。所有用戶確認收到訊息後，代理程式便移除訊息，讓度量反映使用速率。
- **Msgs/sec In**：顯示代理程式每秒從產生器接收多少訊息。度量會反映產生速率。

流量控制會針對使用調整產生，注意產生速度是否變慢或停止。如果產生速度變慢或停止，則表示產生器和用戶的處理速度有差異。您也可以使用 `imqcmd list dst` 指令，檢查已傳送的未確認 (UnAcked) 訊息數目。如果未確認的訊息數目小於目標容量，表示目標有其他容量受到用戶端的流量控制所控制。

若要解決此問題：如果產生速率一直快於使用速率，請考慮定期使用流量控制讓系統步調一致。後續幾節介紹如何解決下面的可能因素：

- 用戶的速度太慢
- 用戶端未確定訊息
- 用戶無法確認訊息
- 長期用戶處於非使用中狀態
- 發生非預期的代理程式錯誤

可能原因：用戶的速度太慢。

若要確認此問題的原因：如「[產生器的速度比用戶快](#)」中所描述，使用度量來判斷產生速率和使用速率。

若要解決此問題：

- 使用以下所示的指令，將目標的限制運作方式設定為 `FLOW_CONTROL`：

```
imqcmd update dst -n myDst -t q -o consumerFlowLimit=FLOW_CONTROL
```

使用流量控制功能以根據使用速率減慢產生速率，防止在代理程式上累積訊息。產生器應用程式會保留訊息，直到目標可以處理這些訊息為止，以降低過期風險。

- 向應用程式開發者詢問產生器是以固定的速率傳送訊息，或是定期以資料組的方式傳送訊息。如果應用程式是以資料組的方式傳送訊息，請按照下一項目的指示，增加目標限制。
- 根據訊息的數目和(或)位元組，增加目標限制。若要變更目標的訊息數目，請輸入下列格式的指令：

```
imqcmd update dst -n destName -t {q|t} -o maxNumMsgs=number
```

若要變更目標的容量，請輸入下列格式的指令：

```
imqcmd update dst -n destName -t {q|t} -o maxTotalMsgBytes=number
```

請注意，提高限制會增加代理程式使用的記憶體容量。如果限制太高，代理程式可能會用盡記憶體而無法處理訊息。

- 考慮在產生大量負載期間，能否承受訊息流失。

可能原因：用戶端未確定訊息。

若要確認此問題的原因：向應用程式開發者查詢應用程式是否使用作業事件。如果應用程式使用作業事件，請列出使用中作業事件，如下所示：

```
imqcmd list txn
```

下列是指令輸出的範例：

```
-----
Transaction ID      State      User name  # Msgs/# Acks  Creation time
-----
6800151593984248832  STARTED  guest     3/2           7/19/04 11:03:08 AM
```

請注意訊息和確認的數目。如果訊息數目大於確認數目，表示產生器可能有傳送個別訊息，但卻無法確定作業事件。代理程式收到確定後，才能路由並傳送該項作業事件的訊息。如果確認數目大於訊息數目，表示用戶可能有傳送個別訊息，但卻無法確定作業事件。代理程式收到確定後，才能移除該項作業事件的確認。

若要解決此問題：請連絡應用程式開發者，修正程式錯誤。

可能原因：用戶無法確認訊息。

若要確認此問題的原因：連絡應用程式開發者，判斷應用程式使用系統型確認或用戶端型確認。如果應用程式使用系統型確認，請略過本節。如果應用程式使用用戶端型確認 (CLIENT_ACKNOWLEDGE)，請使用如下所示的指令，先減少用戶端上儲存的訊息數目：

```
imqcmd update dst -n myDst -t q -o consumerFlowLimit=1
```

然後判斷是因為用戶的速度太慢而使代理程式緩衝訊息，或是用戶處理訊息的速度很快，只是未確認訊息。使用下列指令，列出目標：

```
imqcmd list dst
```

提供使用者名稱和密碼之後，出現類似下列的輸出：

Listing all the destinations on the broker specified by:

```
-----
Host          Primary Port
-----
localhost     7676
-----
Name          Type      State   Producers  Consumers  Msgs
                                     Total    Count  UnAck  Avg Size
-----
MyDest        Queue    RUNNING  0           0          5      200    1177.0
mq.sys.dmqs   Queue    RUNNING  0           0          35      0      1422.0
```

Successfully listed destinations.

UnAck 數目代表代理程式傳送出去後等待確認的訊息數目。如果此數目變大或增加，表示代理程式正在傳送訊息，不等待速度較慢的用戶。您也可以看出用戶未確認訊息。

若要解決此問題：請連絡應用程式開發者，修正編碼錯誤。

可能原因：長期用戶處於非使用中狀態。

若要確認此問題的原因：使用下列的指令格式，查看主題的長期訂閱者：

```
imqcmd list dur -d topicName
```

若要解決此問題：

- 使用 `imqcmd purge dur` 指令，清除長期用戶。
- 重新啟動用戶應用程式。

可能原因：發生非預期的代理程式錯誤。

若要確認此問題的原因：如「[產生器的速度比用戶快](#)」中所描述，使用 QBrowser 檢查訊息。如果 `JMS_SUN_DMQ_UNDELIVERED_REASON` 的值是 `ERROR`，表示代理程式發生錯誤。

若要解決此問題：

- 檢查代理程式記錄檔，尋找相關錯誤。
- 連絡 Sun 技術支援，報告代理程式的問題。

第 3 部分

参照

- 第 13 章
- 第 14 章
- 第 15 章
- 第 16 章
- 第 17 章
- 第 18 章

◆◆◆ 13

第 13 章

指令行參照

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

- 第 217 頁的「指令行語法」
- 第 218 頁的「代理程式公用程式」
- 第 222 頁的「指令公用程式」
- 第 229 頁的「物件管理員公用程式」
- 第 231 頁的「資料庫管理者公用程式」
- 第 232 頁的「使用者管理員公用程式」
- 第 233 頁的「服務管理員公用程式」
- 第 234 頁的「金鑰工具公用程式」

指令行語法

Message Queue 指令行公用程式是 Shell 指令。公用程式名稱是一個指令，其子指令或選項是傳給指令的引數。個別的指令不需要啟動或退出公用程式。

所有指令行公用程式共用以下指令語法：

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

其中，*utilityName* 是以下其中之一：

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

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

以下指令可啟動預設代理程式，是沒有子指令子句的指令行範例：

```
imqbrokerd
```

此為更完整的範例：

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

此指令會銷毀名稱為 `myQueue` 的佇列目標 (目標類型 `q`)。對於使用者名稱 `admin` 會執行認證；該指令會提示您輸入密碼。該指令以無訊息模式執行，將不會提示您確認 (`-f` 選項)，亦不會顯示任何輸出 (`-s` 選項)。

代理程式公用程式

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

表 13-1 顯示 `imqbrokerd` 指令的選項，以及每個選項所置換的配置特性 (若有)。

表 13-1 代理程式公用程式選項

選項	被置換的特性	說明
<code>-name instanceName</code>	<code>imq.instancename</code>	代理程式的實例名稱 在同一台主機上執行的多個代理程式實例，必須擁有不同的實例名稱。 預設值：imqbroker
<code>-port portNumber</code>	<code>imq.portmapper.port</code>	代理程式連接埠對映器的連接埠號 Message Queue 用戶端可使用此連接埠號連線至代理程式。在同一台主機上執行的多個代理程式實例，必須擁有不同的連接埠對映器連接埠號。 預設值：7676

表 13-1 代理程式公用程式選項 (續)

選項	被置換的特性	說明
<code>-cluster broker1 [[, broker2]...]</code>	<code>imq.cluster.brokerlist</code>	將代理程式連線至叢集 ¹ 指定的代理程式會與 <code>imq.cluster.brokerlist</code> 特性中的清單合併。每個代理程式引數的格式如下 <i>hostName: portNumber</i> <i>hostName</i> <i>:portNumber</i> 如果省略 <i>hostName</i> ，則預設值為 <code>localhost</code> ；如果省略 <i>portNumber</i> ，則預設值為 <code>7676</code> 。
<code>-Dproperty= value</code>	實例配置檔案中的對應特性	設定配置特性 請參閱第 14 章，以取得代理程式配置特性的相關資訊。 注意： 請仔細檢查使用此選項設定的特性的拼寫與格式。不正確的值會被忽略，不會另行通知或警告。
<code>-reset props</code>	無	重設配置特性 以空白檔案替代代理程式現有的實例配置檔案 (<code>config.properties</code>)；所有特性皆為預設值。
<code>-reset store</code>	無	重設永久性資料存放區 清除資料存放區的所有永久性資料 (包括永久性訊息、長期訂閱與作業事件資訊)，讓您在啟動代理程式實例時「重新來過」。若要防止在後續的重新啟動時重設永久性存放區，請不使用 <code>-reset</code> 選項重新啟動代理程式實例。 若只要清除永久性訊息或長期訂閱，請改用 <code>-reset messages</code> 或 <code>-reset durables</code> 。
<code>-reset messages</code>	無	清除資料存放區的永久性訊息
<code>-reset durables</code>	無	清除資料存放區的長期訂閱
<code>-backup fileName</code>	無	將配置變更記錄備份至檔案 ¹ 如需更多資訊，請參閱第 162 頁的「管理配置變更記錄」。

¹ 僅適用於代理程式叢集。

表 13-1 代理程式公用程式選項 (續)

選項	被置換的特性	說明
-restore <i>fileName</i>	無	從備份檔案復原配置變更記錄 ¹ 此備份檔案必須已使用 -backup 選項事先建立。 如需更多資訊，請參閱第 162 頁的「管理配置變更記錄」。
-remove instance	無	移除代理程式實例 ² 刪除此實例配置檔案、記錄檔、永久性存放區以及與此實例相關的其他檔案和目錄。
-password <i>keyPassword</i>	imq.keystore.password	SSL 憑證金庫庫的密碼 ³
-dbuser <i>userName</i>	imq.persist.jdbc.user	JDBC 型永久性資料存放區的使用者名稱
-dbpassword <i>dbPassword</i>	imq.persist.jdbc.password	JDBC 型永久性資料存放區的密碼 ³
-ldappassword <i>ldapPassword</i>	imq.user_repository.ldap.password	LDAP 使用者儲存庫的密碼 ³
-passfile <i>filePath</i>	imq.passfile.enabled imq.passfile.dirpath imq.passfile.name	密碼檔案的位置 將代理程式的 imq.passfile.enabled 特性設為 true，imq.passfile.dirpath 設為包含密碼檔案的路徑，而 imq.passfile.name 設為檔案名稱。 如需更多資訊，請參閱第 138 頁的「密碼檔案」。
-shared	imq.jms.threadpool_model	使用共用的執行緒池模型來實作 jms 連線服務。 將會在連線之間共用執行執行緒，以增加所支援的連線數目。 將代理程式的 imq.jms.threadpool_model 特性設為 shared。
-javahome <i>path</i>	無	替代 Java 執行階段的位置 預設運作方式： 使用安裝在系統上或是隨附於 Message Queue 的執行階段。

¹ 僅適用於代理程式叢集。² 除非也指定了 -force，否則此選項會要求使用者確認。³ 此選項已經停用而且會永遠移除。請省略密碼 (將會以互動方式提示使用者輸入密碼)，或使用 -passfile 選項指定包含密碼的檔案。

表 13-1 代理程式公用程式選項 (續)

選項	被置換的特性	說明
<code>-vmargs arg1 [[arg2] ...]</code>	無	將引數傳送給 Java 虛擬機器。 以空格分隔引數。若要傳送多個引數，或包含空格的引數，請將引數清單括在引號內。 VM 引數只能從指令行傳送；實例配置檔案中並沒有相關的配置特性。
<code>-license [licenseName]</code>	無	要載入的授權 (如果與已安裝的 Message Queue 產品版本的預設值不同)： pe：具備基本功能的 Platform Edition try：具備企業功能的 Platform Edition (90 天試用期) unl：Enterprise Edition 如果未指定授權名稱，此選項會列出系統上已安裝的所有授權。
<code>-upgrade-store-nobackup</code>	無	從不相容的版本升級至 Message Queue 3.5 或 3.5 SPx 時，自動移除舊的資料存放區 ² 如需更多資訊，請參閱「Message Queue 安裝指南」。
<code>-force</code>	無	執行動作而無需使用者確認 此選項僅適用於通常會要求確認的 <code>-remove instance</code> 與 <code>-upgrade-store-nobackup</code> 選項。
<code>-loglevel level</code>	<code>imq.broker.log.level</code>	記錄層級： NONE ERROR WARNING INFO 預設值：INFO
<code>-metrics interval</code>	<code>imq.metrics.interval</code>	代理程式度量的記錄時間間隔 (以秒為單位)
<code>-tty</code>	<code>imq.log.console.output</code>	將所有訊息記錄至主控台 將代理程式的 <code>imq.log.console.output</code> 特性設為 ALL。 如果未指定，將只記錄錯誤與警告訊息。

² 除非也指定了 `-force`，否則此選項會要求使用者確認。

表 13-1 代理程式公用程式選項 (續)

選項	被置換的特性	說明
-s -silent	imq.log.console.output	無訊息模式 (不記錄至主控台) 將代理程式的 imq.log.console.output 特性設為 NONE。
-version	無	顯示版本資訊 ⁴
-h -help	無	顯示用法說明 ⁴

⁴ 會忽略指令行上指定的任何其他選項。

指令公用程式

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

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

第 223 頁的「代理程式管理」

shutdown bkr	關閉代理程式
restart bkr	重新啟動代理程式
pause bkr	暫停代理程式
resume bkr	重新繼續代理程式
update bkr	設定代理程式特性
reload cls	重新載入叢集配置
query bkr	列出代理程式特性值
metrics bkr	顯示代理程式度量

第 224 頁的「連線服務管理」

pause svc	暫停連線服務
resume svc	重新繼續連線服務
update svc	設定連線服務特性
list svc	列出代理程式上可用的連線服務

query svc	列出連線服務特性值
metrics svc	顯示連線服務度量
第 225 頁的「連線管理」	
list cxn	列出代理程式上的連線
query cxn	顯示連線資訊
第 226 頁的「實體目標管理」	
create dst	建立實體目標
destroy dst	銷毀實體目標
pause dst	暫停將訊息傳送給實體目標
resume dst	重新繼續將訊息傳送給實體目標
update dst	設定實體目標特性
purge dst	清除來自實體目標的所有訊息
compact dst	壓縮實體目標
list dst	列出實體目標
query dst	列出實體目標特性值
metrics dst	顯示實體目標度量
第 227 頁的「長期訂閱管理」	
destroy dur	銷毀長期訂閱
purge dur	清除長期訂閱的所有訊息
list dur	列出主題的長期訂閱
第 228 頁的「作業事件管理」	
commit txn	確定作業事件
rollback txn	回復作業事件
list txn	列出代理程式追蹤的作業事件
query txn	顯示作業事件資訊

代理程式管理

要使用代理程式公用程式 (imqbrokerd) 才能啟動代理程式；無法使用指令公用程式啟動代理程式。啟動代理程式之後，就可以使用表 13-2 中列出的 imqcmd 子指令對該代理程式進行管理與控制。

表 13-2 用於代理程式管理的指令公用程式子指令

語法	說明
<code>shutdown bkr [-b <i>hostName</i> :<i>portNumber</i>]</code>	關閉代理程式
<code>restart bkr [-b <i>hostName</i> :<i>portNumber</i>]</code>	重新啟動代理程式 關閉代理程式，然後使用原來啟動時所指定的相同選項重新啟動該代理程式。
<code>pause bkr [-b <i>hostName</i> :<i>portNumber</i>]</code>	暫停代理程式 如需更多資訊，請參閱第 94 頁的「暫停代理程式」。
<code>resume bkr [-b <i>hostName</i> :<i>portNumber</i>]</code>	重新繼續代理程式
<code>update bkr [-b <i>hostName</i> :<i>portNumber</i>] -o <i>property1</i>= <i>value1</i> [[-o <i>property2</i> =<i>value2</i>] ...]</code>	設定代理程式特性 請參閱第 14 章，以取得代理程式特性的相關資訊。
<code>reload cls</code>	重新載入叢集配置 ¹ 強制更新所有永久性資訊以保持最新的狀態。
<code>query bkr -b <i>hostName</i> :<i>portNumber</i></code>	列出代理程式特性值 還會列出所有連線至叢集中指定代理程式的執行中代理程式。
<code>metrics bkr [-b <i>hostName</i> :<i>portNumber</i>] [-m <i>metricType</i>] [-int <i>interval</i>] [-msp <i>numSamples</i>]</code>	顯示代理程式度量 -m 選項可指定要顯示的度量類型： ttl：傳入和傳出代理程式的訊息與封包 rts：代理程式的訊息與封包每秒傳入和傳出的速率 cxn：連線、虛擬記憶體堆疊與執行緒 預設值： ttl。 -int 選項可指定顯示度量的時間間隔 (以秒為單位)。 預設值： 5。 -msp 選項可指定要顯示的範例數目。 預設值： 不限制 (無限)。

¹ 僅適用於代理程式叢集。

連線服務管理

表 13-3 列出可管理連線服務的 `imqcmd` 子指令。

表 13-3 連線服務管理的指令公用程式子指令

語法	說明
<pre>pause svc -n serviceName [-b hostName: portNumber]</pre>	<p>暫停連線服務</p> <p>無法暫停 admin 連線服務。</p>
<pre>resume svc -n serviceName [-b hostName: portNumber]</pre>	重新繼續連線服務
<pre>update svc -n serviceName [-b hostName: portNumber] -o property1= value1 [[-o property2=value2] ...]</pre>	<p>設定連線服務特性</p> <p>請參閱第 235 頁的「連線特性」，以取得連線服務特性的相關資訊。</p>
<pre>list svc [-b hostName :portNumber]</pre>	列出代理程式上可用的連線服務
<pre>query svc -n serviceName [-b hostName: portNumber]</pre>	列出連線服務特性值
<pre>metrics svc -n serviceName [-b hostName: portNumber] [-m metricType] [-int interval] [-msp numSamples]</pre>	<p>顯示連線服務度量</p> <p>-m 選項可指定要顯示的度量類型：</p> <ul style="list-style-type: none"> t1l：利用指定的連線服務，傳入和傳出代理程式的訊息與封包 r1s：利用指定的連線服務，代理程式的訊息與封包每秒傳入和傳出的速率 cxn：連線、虛擬記憶體堆疊與執行緒 <p>預設值： t1l。</p> <p>-int 選項可指定顯示度量的時間間隔 (以秒為單位)。 預設值： 5。</p> <p>-msp 選項可指定要顯示的範例數目。預設值： 不限制 (無限)。</p>

連線管理

表 13-4 列出可管理連線的 imqcmd 子指令。

表 13-4 連線服務管理的指令公用程式子指令

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

表 13-4 連線服務管理的指令公用程式子指令 (續)

語法	說明
query cxn -n <i>connectionID</i> [-b <i>hostName: portNumber</i>]	顯示連線資訊

實體目標管理

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

- q：佇列目標
- t：主題目標

表 13-5 用於實體目標管理的指令公用程式子指令

語法	說明
create dst -t <i>destType</i> -n <i>destName</i> [-o <i>property1=value1</i>] [[-o <i>property2=value2</i>] ...]	建立實體目標 ¹ 目標名稱 <i>destName</i> 只能包含字母數字字元 (無空格)，而且必須以字母字元或底線 () 或錢幣符號 (\$) 開頭。而且開頭不能是 mq 字元。
destroy dst -t <i>destType</i> -n <i>destName</i>	銷毀實體目標 ¹ 此作業不適用於由系統建立的目標，例如停用的訊息佇列。
pause dst [-t <i>destType</i> -n <i>destName</i>] [-pst <i>pauseType</i>]	暫停將訊息傳送給實體目標 暫停將訊息傳送給利用 <code>-t</code> 與 <code>-n</code> 選項指定的實體目標。如果未指定這些選項，就會暫停所有目標。 -pst 選項可指定要暫停的訊息傳送類型： CONSUMERS：暫停傳送給訊息用戶 PRODUCERS：暫停傳送給訊息產生器 ALL：暫停所有訊息傳送 預設值：ALL
resume dst [-t <i>destType</i> -n <i>destName</i>]	重新繼續將訊息傳送給實體目標 重新繼續將訊息傳送給利用 <code>-t</code> 與 <code>-n</code> 選項指定的實體目標。如果未指定這些選項，就會重新繼續所有目標。
update dst -t <i>destType</i> -n <i>destName</i> -o <i>property1=value1</i> [[-o <i>property2=value2</i>] ...]	設定實體目標特性 請參閱第 15 章，以取得實體目標特性的相關資訊。

¹ 不能在主代理程式暫時無法使用的代理程式叢集上執行此作業。

表 13-5 用於實體目標管理的指令公用程式子指令 (續)

語法	說明
<code>purge dst -t destType -n destName</code>	清除實體目標中的所有訊息
<code>compact dst [-t destType -n destName]</code>	壓縮實體目標 壓縮由 <code>-t</code> 與 <code>-n</code> 選項所指定之實體目標的檔案式永久性資料存放區。如果未指定這些選項，就會壓縮所有目標。 必須在壓縮目標前暫停目標。
<code>list dst [-t destType] [-tmp]</code>	列出實體目標 列出由 <code>-t</code> 選項指定的所有實體目標類型。如果未指定目標類型，就會列出佇列與主題目標。如果已指定 <code>-tmp</code> 選項，也會列出暫時目標。
<code>query dst -t destType -n destName</code>	列出實體目標特性值
<code>metrics dst -t destType -n destName</code> <code>[-m metricType]</code> <code>[-int interval]</code> <code>[-msp numSamples]</code>	顯示實體目標度量 <code>-m</code> 選項可指定要顯示的度量類型： <code>ttl</code> ：傳入和傳出目標以及常駐於記憶體中的訊息與封包。 <code>rts</code> ：代理程式的訊息與封包每秒傳入和傳出的速率，以及其他速率資訊 <code>con</code> ：與訊息用戶相關的度量 <code>dsk</code> ：磁碟使用率 預設值 ： <code>ttl</code> 。 <code>-int</code> 選項可指定顯示度量的時間間隔 (以秒為單位)。 預設值 ：5。 <code>-msp</code> 選項可指定要顯示的範例數目。 預設值 ：不限制 (無限)。

長期訂閱管理

表 13-6 列出可管理長期訂閱的 `imqcmd` 子指令。

表 13-6 用於長期訂閱管理的指令公用程式子指令

語法	說明
<code>destroy dur -c clientID</code> <code>-n subscriberName</code>	銷毀長期訂閱 ¹

¹ 不能在主代理程式暫時無法使用的代理程式叢集上執行此作業。

表 13-6 用於長期訂閱管理的指令公用程式子指令 (續)

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

作業事件管理

表 13-7 列出可管理作業事件的 `imqcmd` 子指令。

表 13-7 用於作業事件管理的指令公用程式子指令

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

一般指令公用程式選項

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

表 13-8 一般指令公用程式選項

選項	說明
<code>-secure</code>	透過 <code>ssladmin</code> 連線服務安全連線至代理程式
<code>-u userName</code>	認證的使用者名稱 如果省略此選項，指令公用程式就會以互動方式提示您輸入。
<code>-p password</code>	認證的密碼 ¹
<code>-passfile path</code>	密碼檔案位置 如需更多資訊，請參閱第 138 頁的「密碼檔案」。

¹ 此選項已經停用而且會永遠移除。請省略密碼 (將會以互動方式提示使用者輸入密碼)，或使用 `-passfile` 選項指定包含密碼的檔案。

表 13-8 一般指令公用程式選項 (續)

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

² 會忽略指令行上指定的任何其他選項。

³ 不需要使用者名稱與密碼

物件管理員公用程式

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

表 13-9 物件管理員子指令

子指令	說明
add	將受管理物件新增至物件存放區
delete	刪除物件存放區內的受管理物件
list	列出物件存放區中的受管理物件
query	顯示受管理物件資訊
update	修改受管理物件

表 13-10 列出 imqobjmgr 指令的選項。

表 13-10 物件管理員選項

選項	說明
-l <i>lookupName</i>	受管理物件的 JNDI 查找名稱
-j <i>attribute= value</i>	JNDI 物件存放區 (請參閱第 141 頁的「物件存放區」) 的屬性
-t <i>objectType</i>	受管理物件的類型： q：佇列目標 t：主題目標 cf：連線工廠 qf：佇列連線工廠 tf：主題連線工廠 xcf：分散式作業事件的連線工廠 xqf：分散式作業事件的佇列連線工廠 xtf：分散式作業事件的主題連線工廠 e：SOAP 端點 (請參閱「Message Queue Developer's Guide for Java Clients」)
-o <i>attribute= value</i>	受管理物件的屬性 (請參閱第 143 頁的「受管理物件屬性」與第 16 章)
-r <i>readOnlyState</i>	受管理物件是否為唯讀？ 如果為 true，用戶端就無法修改物件的屬性。預設值：false。
-i <i>fileName</i>	包含所有或部分子指令子句的指令檔名稱
-pre	無需執行指令即能預覽結果 此選項對檢查預設屬性的值很有用。
-javahome <i>path</i>	替代 Java 執行階段的位置 預設運作方式 ：使用安裝在系統上或是隨附於 Message Queue 的執行階段。
-f	執行動作而無需使用者確認
-s	無訊息模式 (不顯示輸出)
-v	顯示版本資訊 ¹
-h	顯示用法說明 ¹
-H	顯示詳細的用法說明，包括屬性清單與範例 ¹

¹ 會忽略指令行上指定的任何其他選項。

資料庫管理者公用程式

資料庫管理者公用程式 (imqdbmgr) 可設定 JDBC 型永久性資料存放區的資料庫模式。您也可以用該指令刪除已毀壞的 Message Queue 資料庫表格，或變更資料存放區。

表 13-11 列出可用的子指令。

表 13-11 資料庫管理者子指令

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

表 13-12 列出 imqdbmgr 指令的選項。

表 13-12 資料庫管理者選項

選項	說明
-b <i>instanceName</i>	代理程式的實例名稱
-D <i>property= value</i>	設定代理程式配置特性 請參閱第 241 頁的「永久性特性」，以取得與永久性相關的代理程式配置特性之資訊。 注意： 請仔細檢查使用此選項設定的特性的拼寫與格式。不正確的值會被忽略，不會另行通知或警告。
-u <i>name</i>	認證的使用者名稱

表 13-12 資料庫管理者選項 (續)

選項	說明
-p <i>password</i>	認證的密碼 ¹
-passfile <i>path</i>	密碼檔案位置 如需更多資訊，請參閱第 138 頁的「密碼檔案」。
-v	顯示版本資訊 ²
-h	顯示用法說明 ²

¹ 此選項已經停用而且會永遠移除。請省略密碼 (將會以互動方式提示使用者輸入密碼)，或使用 `-passfile` 選項指定包含密碼的檔案。

² 會忽略指令行上指定的任何其他選項。

使用者管理員公用程式

使用者管理員公用程式 (`imqusermgr`) 可用於寫入或編輯平面檔案使用者儲存庫。此公用程式必須在安裝代理程式的同一台主機上執行；如果代理程式專用的使用者儲存庫尚不存在，您必須先啟動相對的代理程式實例才能建立。您還需要適當的權限才能寫入儲存庫：也就是說，在 Solaris 或 Linux 平台上，您必須是超級使用者或代理程式實例的建立者。

表 13-13 列出 `imqusermgr` 指令提供的子指令。在所有情況下，`-i` 選項都可指定代理程式實例名稱，指令會套用到該代理程式的使用者儲存庫；如果未指定，就會使用預設名稱 `imqbroker`。

表 13-13 使用者管理員子指令

語法	說明
<pre>add [-i <i>instanceName</i>] -u <i>userName</i> -p <i>password</i> [-g <i>group</i>]</pre>	<p>將使用者與密碼新增至儲存庫</p> <p><code>-g</code> 選項是選擇性的，可指定要將這位使用者指定給哪個群組：</p> <ul style="list-style-type: none"> admin user anonymous
<pre>delete [-i <i>instanceName</i>] -u <i>userName</i></pre>	刪除儲存庫的使用者

表 13-13 使用者管理員子指令 (續)

語法	說明
<pre>update [-i instanceName] -u userName -p password [-a activeState] update [-i instanceName] -u userName -a activeState [-p password]</pre>	設定使用者的密碼和 (或) 使用中狀態 -a 選項是一個布林值，可指定使用者為使用中 (true) 或非使用中 (false)。預設值：true。
<pre>list [-i instanceName] [-u userName]</pre>	顯示使用者資訊 如果未指定使用者名稱，就會列出儲存庫中的所有使用者。

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

表 13-14 一般使用者管理員選項

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

¹ 會忽略指令行上指定的任何其他選項。

服務管理員公用程式

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

表 13-15 服務管理員子指令

子指令	說明
<code>install</code>	安裝服務
<code>remove</code>	移除服務
<code>query</code>	顯示啟動選項 啟動選項可包括服務是手動啟動還是自動啟動、服務的位置、Java 執行階段的位置，以及啟動時傳送至代理程式的引數值 (請參閱表 13-16)。

表 13-16 列出 `imqsvcadmin` 指令的選項。

表 13-16 服務管理員選項

選項	說明
-javahome <i>path</i>	替代 Java 執行階段的位置 預設運作方式： 使用安裝在系統上或是隨附於 Message Queue 的執行階段。
-jrehome <i>path</i>	替代 Java 執行階段環境 (JRE) 的位置
-vmargs <i>arg1</i> [[<i>arg2</i>] ...]	要傳送給執行代理程式服務的 Java 虛擬機器 (JVM) 的其他引數 ¹ 範例： imqsvcadm install -vmargs "-Xms16m -Xmx128m"
-args <i>arg1</i> [[<i>arg2</i>] ...]	要傳送給代理程式服務的其他指令行引數 ¹ 範例： imqsvcadm install -args "-passfile d:\imqpassfile" 請參閱第 218 頁的「代理程式公用程式」，以取得代理程式指令行引數的相關資訊。
-h	顯示用法說明 ²

¹ 也可以使用 Windows [系統管理工具] 控制面板中的 [服務] 工具，開啓服務的 [內容] 視窗，並於 [一般] 標籤下方的 [啓動參數] 欄位中，指定這些引數。

² 會忽略指令行上指定的任何其他選項。

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

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\imq_Broker\Parameters

金鑰工具公用程式

金鑰工具公用程式 (imkeytool) 會為代理程式產生自我簽署的憑證，此憑證可用於 ssljms、ssladmin 或 cluster 連線服務。語法為

```
imkeytool -broker
```

若為 UNIX 系統，需要從超級使用者 (root) 帳號執行公用程式。

代理程式特性參照

本章提供有關訊息代理程式配置特性的參考資訊。包含下列各節：

- 第 235 頁的「連線特性」
- 第 237 頁的「路由特性」
- 第 241 頁的「永久性特性」
- 第 245 頁的「安全性特性」
- 第 248 頁的「檢視特性」
- 第 251 頁的「叢集配置特性」
- 第 253 頁的「按字母順序排列的代理程式特性清單」

連線特性

表 14-1 列出與連線服務相關的代理程式特性。

表 14-1 代理程式連線特性

特性	類型	預設	說明
<code>imq.service.activelist</code>	字串	<code>jms,admin</code>	要在代理程式啟動時啟用之連線服務的清單 (由逗號分隔)
<code>imq.hostname</code>	字串	所有可用的 IP 位址	所有連線服務的預設主機名稱或 IP 位址
<code>imq.portmapper.hostname</code>	字串	無	連接埠對映器的主機名稱或 IP 位址 如果指定，會置換 <code>imq.hostname</code>

表 14-1 代理程式連線特性 (續)

特性	類型	預設	說明
<code>imq.portmapper.port¹</code>	整數	7676	連接埠對映器的連接埠號 備註 - 如果在同一台主機上執行多個代理程式實例，那麼每一個實例都必須指定唯一的連接埠對映器連接埠。
<code>imq.serviceName.protocolType.hostname²</code>	字串	無	連線服務的主機名稱或 IP 位址 如果指定，會置換所指定連線服務的 <code>imq.hostname</code>
<code>imq.serviceName.protocolType.port²</code>	整數	0	連線服務的連接埠號 0 值指定連接埠對映器應動態配置連接埠號。
<code>imq.portmapper.backlog</code>	整數	50	作業系統儲存區中的最大擱置連接埠對映器請求數
<code>imq.serviceName.threadpool_model³</code>	字串	dedicated	用於執行緒池管理的執行緒模型： <ul style="list-style-type: none"> dedicated：每個連線都有兩個專屬執行緒，一個用於內送訊息，另一個用於外寄訊息 shared：連線在傳送或接收訊息時會由共用執行緒進行處理 專屬模型會限制所能支援的連線數目，但可以提供較高的效能；共用模型可增加連線數目，但因為需要額外的經常性耗用時間進行執行緒管理，所以效能較低。
<code>imq.serviceName.min_threads</code>	整數	jms : 10 ssljms : 10 httpjms : 10 httpsjms : 10 admin : 4 ssladmin : 4	連線服務的執行緒池中所維護的最小執行緒數 當可用的執行緒數超過此臨界值時，將會在執行緒閒置時將其關閉，直到達到臨界值下限為止。 如其所示，預設值因連線服務而異。
<code>imq.serviceName.max_threads</code>	整數	jms : 1000 ssljms : 500 httpjms : 500 httpsjms : 500 admin : 10 ssladmin : 10	執行緒數目，一旦超過此數目，系統便不會將新的執行緒增加至執行緒池，供已命名的連線服務使用 此數目必須大於 0，並且必須大於 <code>imq.serviceName.min_threads</code> 的值。 如其所示，預設值因連線服務而異。

¹ 可與 `imqcmd update bkr` 指令一起使用² 僅限 jms、ssljms、admin 與 ssladmin 服務；請參閱附錄 C 以取得更多有關配置 httpjms 與 httpsjms 服務的資訊³ 僅限 jms 與 admin 服務

表 14-1 代理程式連線特性 (續)

特性	類型	預設	說明
<code>imq.shared.connectionMonitor_limit⁴</code>	整數	Solaris : 512 Linux : 512 Windows : 64	由分散執行緒監視的最大連線數目 系統配置了足夠的分散執行緒以監視所有連線。此特性的值愈小，將執行緒指定給使用中連線的速度愈快。-1 值表示不限制每個執行緒的連線數目。 如其所示，預設值因作業系統平台而異。
<code>imq.ping.interval</code>	整數	120	測試用戶端與代理程式之間連線的時間間隔 (以秒為單位) 0 或 -1 值可停用定期的連線測試。

⁴ 僅限共用執行緒模型

路由特性

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

表 14-2 代理程式路由特性

特性	類型	預設	說明
<code>imq.system.max_count¹</code>	整數	-1	代理程式保留訊息的最大數目 -1 值表示不限制訊息計數。
<code>imq.system.max_size¹</code>	字串	-1	代理程式保留訊息的最大總容量 此值可能會使用下列字尾，以位元組、千位元組或百萬位元組表示： b : 位元組 k : 千位元組 (1024 位元組) m : 百萬位元組 (1024 × 1024 = 1,048,576 位元組) 無字尾的值表示以位元組為單位；-1 值表示不限制訊息容量。 範例： 1600 : 1600 位元組 1600b : 1600 位元組 16k : 16 千位元組 (= 16,384 位元組) 16m : 16 百萬位元組 (= 16,777,216 位元組) -1 : 不限制

¹ 可與 `imqcmd update bkr` 指令一起使用

表 14-2 代理程式路由特性 (續)

特性	類型	預設	說明
<code>imq.message.max_size</code> ¹	字串	70m	單一訊息內文的最大容量 語法與 <code>imq.system.max_size</code> 相同 (請參閱前述內容)。
<code>imq.message.expiration.interval</code>	整數	60	收回已過期訊息的時間間隔 (以秒為單位)
<code>imq.resourceState.threshold</code>	整數	green : 0 yellow : 80 orange : 90 red : 98	可觸發記憶體資源狀態的記憶體用量百分比 (其中 <code>resourceState</code> 為 green、yellow、orange 或 red)
<code>imq.resourceState.count</code>	整數	green : 5000 yellow : 500 orange : 50 red : 0	檢查記憶體是否已達到資源狀態臨界值之前，每個批次中可允許的最大內送訊息數目 (其中 <code>resourceState</code> 是 green、yellow、orange 或 red) 系統記憶體變得逐漸不足時，此限制會減少訊息產生器產生訊息的數量。
<code>imq.destination.DMQ.truncateBody</code> ¹	布林值	false	儲存至停用的訊息佇列之前，是否先移除訊息內文？ 如果為 true，將只會儲存訊息標頭與特性資料。
<code>imq.transaction.autorollback</code>	布林值	false	是否要自動回復在代理程式啟動時為 PREPARED 狀態的分散式作業事件？ 如果為 false，就必須使用指令公用程式 (<code>imqcmd</code>) 以手動確定或回復作業事件。

¹ 可與 `imqcmd update bkr` 指令一起使用

表 14-3 自動建立的目標之代理程式特性

特性	類型	預設	說明
<code>imq.autocreate.queue</code> ^{1,2}	布林值	true	允許自動建立佇列目標？
<code>imq.autocreate.topic</code> ³	布林值	true	允許自動建立主題目標？
<code>imq.autocreate.destination.maxNumMsgs</code>	整數	100000	未使用訊息的最大數目 -1 值表示不限制訊息數目。

¹ 可與 `imqcmd update bkr` 指令一起使用

² 僅限佇列目標

³ 僅限主題目標

表 14-3 自動建立的目標之代理程式特性 (續)

特性	類型	預設	說明
<code>imq.autocreate.destination.maxBytesPerMsg</code>	字串	10k	<p>任何單一訊息的最大容量 (以位元組為單位)</p> <p>此值可能會使用下列字尾，以位元組、千位元組或百萬位元組表示：</p> <ul style="list-style-type: none"> b：位元組 k：千位元組 (1024 位元組) m：百萬位元組 ($1024 \times 1024 = 1,048,576$ 位元組) <p>無字尾的值是以位元組表示；-1 值表示不限制訊息容量。</p> <p>範例：</p> <ul style="list-style-type: none"> 1600：1600 位元組 1600b：1600 位元組 16k：16 千位元組 (= 16,384 位元組) 16m：16 百萬位元組 (= 16,777,216 位元組) -1：不限制
<code>imq.autocreate.destination.maxTotalMsgBytes</code>	字串	10m	<p>未使用訊息的最大總記憶體容量 (以位元組為單位)</p> <p>語法與 <code>imq.autocreate.destination.maxBytesPerMsg</code> 相同 (請參閱前述內容)。</p>
<code>imq.autocreate.destination.limitBehavior</code>	字串	REJECT_NEWEST	<p>達到記憶體限制臨界值時的代理程式運作方式：</p> <ul style="list-style-type: none"> FLOW_CONTROL：減緩產生器速度 REMOVE_OLDEST：捨棄最舊的訊息 REMOVE_LOW_PRIORITY：根據訊息存在時間，捨棄優先權最低的訊息；不會對訊息產生用戶端傳送通知 REJECT_NEWEST：拒絕最新的訊息；只有當訊息為永久性時，才通知訊息產生用戶端發生異常 <p>如果值為 REMOVE_OLDEST 或 REMOVE_LOW_PRIORITY，而且 <code>imq.autocreate.destination.useDMQ</code> 特性為 true，則超過限制的訊息就會移至停用的訊息佇列。</p>

表 14-3 自動建立的目標之代理程式特性 (續)

特性	類型	預設	說明
<code>imq.autocreate.destination.maxNumProducers</code>	整數	100	目標的最大訊息產生器數目 達到限制時，將無法建立新的產生器。-1 值表示不限制產生器數目。
<code>imq.autocreate.queue.maxNumActiveConsumers²</code>	整數	1	在佇列目標的負載平衡傳送中，使用中訊息用戶的最大數目 -1 值表示不限制用戶數目。
<code>imq.autocreate.queue.maxNumBackupConsumers²</code>	整數	0	在佇列目標的負載平衡傳送中，備份訊息用戶的最大數目 -1 值表示不限制用戶數目。
<code>imq.autocreate.queue.consumerFlowLimit²</code>	整數	1000	單一批次中傳送給佇列用戶的最大訊息數目 在負載平衡佇列傳送中，此數目是開始進行負載平衡前，路由到使用中用戶的佇列訊息初始數目。目標用戶可以在連線上指定較低的值來置換這項限制。 -1 值表示不限制訊息數目。
<code>imq.autocreate.topic.consumerFlowLimit³</code>	整數	1000	單一批次中傳送給主題用戶的最大訊息數目 -1 值表示不限制用戶數目。
<code>imq.autocreate.destination.isLocalOnly</code>	布林值	false	僅限本機傳送？ 此特性僅適用於代理程式叢集，而且目標一經建立之後便無法變更。如果為 true 則其他代理程式上不會重複目標，且限制目標僅會傳送訊息給本機用戶 (會連線至建立目標的代理程式)。
<code>imq.autocreate.queue.localDeliveryPreferred²</code>	布林值	false	偏好本機傳送？ 此特性僅適用於代理程式叢集中的負載平衡佇列傳送。如果為 true，只有當本機代理程式上沒有用戶時，才會將訊息傳送給遠端用戶；目標不可設定為僅限於本機傳送 (即 <code>imq.autocreate.destination.isLocalOnly</code> 必須為 false)。
<code>imq.autocreate.destination.useDMQ</code>	布林值	true	要將停用的訊息傳送給停用的訊息佇列？ 如果為 false，停用的訊息將會被捨棄。

² 僅限佇列目標

³ 僅限主題目標

永久性特性

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

表 14-4 全域代理程式永久性特性

特性	類型	預設	說明
<code>imq.persist.store</code>	字串	file	永久性資料存放區的模型： file：檔案式永久性 jdbc：JDBC 型永久性

檔案式永久性

表 14-5 列出與檔案式永久性相關的代理程式特性。

表 14-5 檔案式永久性的代理程式特性

特性	類型	預設	說明
<code>imq.persist.file.message.max_record_size</code>	字串	1m	<p>新增至訊息存放區檔案的最大訊息容量</p> <p>超過此容量的任何訊息，都會以個別的檔案各自儲存。</p> <p>此值可能會使用下列字尾，以位元組、千位元組或百萬位元組表示：</p> <ul style="list-style-type: none"> b：位元組 k：千位元組 (1024 位元組) m：百萬位元組 (1024 × 1024 = 1,048,576 位元組) <p>無字尾的值表示以位元組為單位。</p> <p>範例：</p> <ul style="list-style-type: none"> 1600：1600 位元組 1600b：1600 位元組 16k：16 千位元組 (= 16,384 位元組) 16m：16 百萬位元組 (= 16,777,216 位元組)

表 14-5 檔案式永久性的代理程式特性 (續)

特性	類型	預設	說明
<code>imq.persist.file.destination.message.filepool.limit</code>	整數	100	目標檔案池中可供重新使用的最大可用檔案數目。 超過此限制的可用檔案將會被刪除。代理程式將依需要建立與刪除超過此限制的其他檔案。 此限制愈高，代理程式處理永久性資料的速度就愈快。
<code>imq.persist.file.message.filepool.cleanratio</code>	整數	0	可用檔案池中要維持在清除 (空白) 狀態的檔案百分比 此值越高，檔案池所需的磁碟空間就越小，但在作業期間清除檔案所需的經常性耗用時間會越長。
<code>imq.persist.file.message.cleanup</code>	布林值	false	要在關機時清除可用檔案池中的檔案？ 將此特性設為 true 可節省磁碟空間以供檔案存放區使用，但會減緩代理程式關閉的速度。
<code>imq.persist.file.sync.enabled</code>	布林值	false	要與實體儲存裝置同步化常駐記憶體狀態？ 將此特性設為 true 可避免因系統當機而造成資料遺失，但效能會降低。 備註 - 如果 Message Queue 執行 Sun Cluster 和 Sun Cluster Data Service，請在所有叢集節點上將代理程式的此特性設定為 true。

JDBC 型永久性

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

表 14-6 JDBC 型永久性的代理程式特性

特性	範例	說明
<code>imq.persist.jdbc.brokerid</code>	PointBase 內嵌式版本不需要	(選擇性) 代理程式實例識別碼 識別碼必須為字母數字字串，其長度不能超過 $n-12$ 個字元，其中 n 為資料庫所允許的最長表格名稱長度。 若有多個代理程式實例使用相同的資料庫作為永久性資料存放區，則此識別碼會附加至資料庫表格名稱，使這些資料庫表格名稱均是唯一的。由於內嵌式資料庫僅會儲存一個代理程式實例的資料，因此通常不需要此屬性。
<code>imq.persist.jdbc.driver</code>	<code>com.pointbase.jdbc.jdbcUniversalDriver</code>	可供連線至資料庫的 JDBC 驅動程式 Java 類別名稱
<code>imq.persist.jdbc.openburl</code>	<code>jdbc:pointbase:embedded:dbName ; database.home= .../instances/instanceName/dbstore</code>	可供開啓現有資料庫連線的 URL
<code>imq.persist.jdbc.createdburl</code>	<code>jdbc:pointbase:embedded:dbName ; new,database.home= .../instances/instanceName/dbstore</code>	(選擇性) 可供建立新資料庫的 URL 僅當使用 Message Queue 資料庫管理者公用程式 (<code>imqdbmgr</code>) 建立資料庫時，才需要此特性。
<code>imq.persist.jdbc.closedburl</code>	PointBase 不需要	(選擇性) 可供關閉資料庫連線的 URL
<code>imq.persist.jdbc.user</code>		(選擇性) 必要時，可供開啓資料庫連線的使用者名稱 基於安全性原因，可改用指令行選項 <code>imqbrokerd -dbuser</code> 與 <code>imqdbmgr -u</code> 來指定該值。
<code>imq.persist.jdbc.needpassword</code>		(選擇性) 資料庫是否需要密碼以供代理程式存取使用？ 如果為 <code>true</code> ， <code>imqbrokerd</code> 與 <code>imqdbmgr</code> 指令將會提示您輸入密碼，除非您使用 <code>-passfile</code> 選項指定包含密碼的密碼檔案。
<code>imq.persist.jdbc.password¹</code>		(選擇性) 可供開啓資料庫連線的密碼

¹ 應只在密碼檔案中使用

表 14-6 JDBC 型永久性的代理程式特性 (續)

特性	範例	說明
imq.persist.jdbc.table.IMQSV35	CREATE TABLE <i>\${name}</i> (STOREVERSION INTEGER NOT NULL, BROKERID VARCHAR(100))	可供建立版本表格的 SQL 指令
imq.persist.jdbc.table.IMQCCREC35	CREATE TABLE <i>\${name}</i> (RECORDTIME BIGINT NOT NULL, RECORD BLOB(10k))	可供建立配置變更記錄表格的 SQL 指令
imq.persist.jdbc.table.IMQDEST35	CREATE TABLE <i>\${name}</i> (DID VARCHAR(100) NOT NULL, DEST BLOB(10k), <i>primaryKey</i> (DID))	可供建立目標表格的 SQL 指令
imq.persist.jdbc.table.IMQINT35	CREATE TABLE <i>\${name}</i> (CUID BIGINT NOT NULL, INTEREST BLOB(10k), <i>primaryKey</i> (CUID))	可供建立偏好表格的 SQL 指令
imq.persist.jdbc.table.IMQMSG35	CREATE TABLE <i>\${name}</i> (MID VARCHAR(100) NOT NULL, DID VARCHAR(100), MSGSIZE BIGINT, MSG BLOB(1m), <i>primaryKey</i> (MID))	可供建立訊息表格的 SQL 指令 MSG 的預設最大長度為 1 MB (1m)。如果您預期訊息長度會大於此長度，請相應設定長度。如果已經建立表格，那麼您必須重新建立這些表格，才能變更最大的訊息長度。
imq.persist.jdbc.table.IMQPROPS35	CREATE TABLE <i>\${name}</i> (PROPNAME VARCHAR(100) NOT NULL, PROPVALUE BLOB(10k), <i>primaryKey</i> (PROPNAME))	可供建立特性表格的 SQL 指令
imq.persist.jdbc.table.IMQILIST35	CREATE TABLE <i>\${name}</i> (MID VARCHAR(100) NOT NULL, CUID BIGINT, DID VARCHAR(100), STATE INTEGER, <i>primaryKey</i> (MID, CUID))	可供建立偏好狀態表格的 SQL 指令
imq.persist.jdbc.table.IMQTXN35	CREATE TABLE <i>\${name}</i> (TUID BIGINT NOT NULL, STATE INTEGER, TSTATEOBJ BLOB(10K), <i>primaryKey</i> (TUID))	可供建立作業事件表格的 SQL 指令

表 14-6 JDBC 型永久性的代理程式特性 (續)

特性	範例	說明
<code>imq.persist.jdbc.table.IMQTACK35</code>	<pre>CREATE TABLE \${name} (TUID BIGINT NOT NULL, TXNACK BLOB(10k))</pre>	可供建立作業事件確認表格的 SQL 指令

安全性特性

表 14-7 列出與安全性服務相關的代理程式特性。

表 14-7 代理程式安全性特性

特性	類型	預設	說明
<code>imq.accesscontrol.enabled</code>	布林值	true	<p>使用存取控制？</p> <p>如果為 true，系統將會檢查存取控制特性檔案，以驗證認證的使用者已取得授權使用連線服務或執行與特定目標相關的特定作業。</p>
<code>imq.serviceName.accesscontrol.enabled</code>	布林值	無	<p>要使用連線服務的存取控制？</p> <p>如果指定，會置換指定之連線服務的 <code>imq.accesscontrol.enabled</code>。</p> <p>如果為 true，系統將會檢查存取控制特性檔案，以驗證認證的使用者已取得授權使用連線服務或執行與特定目標相關的特定作業。</p>
<code>imq.accesscontrol.file.filename</code>	字串	<code>accesscontrol.properties</code>	<p>存取控制特性檔的名稱</p> <p>檔案名稱可指定與存取控制目錄相關的路徑 (請參閱附錄 A)。</p>
<code>imq.serviceName.accesscontrol.file.filename</code>	字串	無	<p>連線服務的存取控制特性檔名稱</p> <p>如果指定，會置換指定之連線服務的 <code>imq.accesscontrol.file.filename</code>。</p> <p>檔案名稱可指定與存取控制目錄相關的路徑 (請參閱附錄 A)。</p>

表 14-7 代理程式安全性特性 (續)

特性	類型	預設	說明
<code>imq.authentication.type</code>	字串	<code>digest</code>	密碼編碼方法： <code>basic</code> ：Base64 <code>digest</code> ：MD5
<code>imq.serviceName.authentication.type</code>	字串	無	連線服務的密碼編碼方法： <code>basic</code> ：Base64 <code>digest</code> ：MD5 如果指定，會置換指定之連線服務的 <code>imq.authentication.type</code> 。
<code>imq.authentication.basic.user_repository</code>	字串	<code>file</code>	Base64 認證的使用者儲存庫類型： <code>file</code> ：檔案式 <code>ldap</code> ：LDAP
<code>imq.authentication.client.response.timeout</code>	整數	<code>180</code>	等待用戶端回應認證請求的時間間隔 (以秒為單位)
<code>imq.passfile.enabled</code>	布林值	<code>false</code>	要從密碼檔案取得密碼？
<code>imq.passfile.dirpath</code>	字串	請參閱附錄 A	前往包含密碼檔的路徑
<code>imq.passfile.name</code>	字串	<code>passfile</code>	密碼檔案名稱
<code>imq.imqcmd.password</code>	字串	無	管理使用者的密碼 指令公用程式 (<code>imqcmd</code>) 可使用此密碼，在執行指令之前認證該使用者。
<code>imq.user_repository.ldap.server</code>	字串	無	LDAP 伺服器的主機名稱與連接埠號 值的格式為 <code>hostName:port</code> 其中 <code>hostName</code> 是執行 LDAP 伺服器的主機之完全合格 DNS 名稱，而 <code>port</code> 是該伺服器使用的連接埠號。 若要指定容錯移轉伺服器的清單，請使用下列語法： <code>host1:port1</code> <code>ldap://host2:port2</code> <code>ldap://host3:port3</code> ...

表 14-7 代理程式安全性特性 (續)

特性	類型	預設	說明
			清單中的項目以空格分開。請注意，每個容錯移轉伺服器位址都會以 <code>ldap://</code> 開頭。即使您使用 SSL 並且將特性 <code>imq.user_repository.ldap.ssl.enabled</code> 設定為 <code>true</code> ，也請使用此格式。您不需要在位址中指定 <code>ldaps</code> 。
<code>imq.user_repository.ldap.principal</code>	字串	無	可供連結到 LDAP 使用者儲存庫的辨別名稱 如果 LDAP 伺服器允許匿名搜尋，則不需要。
<code>imq.user_repository.ldap.password¹</code>	字串	無	可供連結到 LDAP 使用者儲存庫的密碼 如果 LDAP 伺服器允許匿名搜尋，則不需要。
<code>imq.user_repository.ldap.propertyName</code>	敬請期待	敬請期待	敬請期待
<code>imq.user_repository.ldap.base</code>	字串	無	LDAP 使用者項目的目錄基底
<code>imq.user_repository.ldap.uidattr</code>	字串	無	LDAP 使用者名稱的提供者專用屬性識別碼
<code>imq.user_repository.ldap.usrfilter</code>	字串	無	(選擇性) LDAP 使用者搜尋的 JNDI 篩選器
<code>imq.user_repository.ldap.grpsearch</code>	布林值	<code>false</code>	啟用 LDAP 群組搜尋？ 備註 - Message Queue 不支援巢式群組。
<code>imq.user_repository.ldap.grpbase</code>	字串	無	LDAP 群組項目的目錄基底
<code>imq.user_repository.ldap.gidattr</code>	字串	無	LDAP 群組名稱的提供者專用屬性識別碼
<code>imq.user_repository.ldap.memattr</code>	字串	無	LDAP 群組中使用者名稱的提供者專用屬性識別碼
<code>imq.user_repository.ldap.grpfilter</code>	字串	無	(選擇性) LDAP 群組搜尋的 JNDI 篩選器
<code>imq.user_repository.ldap.timeout</code>	整數	280	LDAP 搜尋的時間限制 (以秒為單位)

¹ 應只在密碼檔案中使用

表 14-7 代理程式安全性特性 (續)

特性	類型	預設	說明
<code>imq.user_repository.ldap.ssl.enabled</code>	布林值	<code>false</code>	要使用 SSL 與 LDAP 伺服器進行通訊？
<code>imq.keystore.file.dirpath</code>	字串	請參閱附錄 A	前往包含金鑰庫檔案的路徑
<code>imq.keystore.file.name</code>	字串	<code>keystore</code>	金鑰庫檔的名稱
<code>imq.keystore.password¹</code>	字串	無	金鑰庫檔的密碼
<code>imq.audit.enabled²</code>	布林值	<code>false</code>	要啟動代理程式記錄檔的稽核記錄功能？

¹ 應只在密碼檔案中使用

² 只適用於 Message Queue Enterprise Edition

檢視特性

表 14-8 列出與監視服務相關的代理程式特性。

表 14-8 代理程式監視特性

特性	類型	預設	說明
<code>imq.log.level¹</code>	字串	<code>INFO</code>	記錄層級 指定可寫入至輸出通道的記錄資訊種類。由高至低的值為： ERROR WARNING INFO 每個層級都包含高於它的層級 (例如 <code>WARNING</code> 會包含 <code>ERROR</code>)。
<code>imq.destination.logDeadMsgs¹</code>	布林值	<code>false</code>	要記錄有關停用的訊息之資訊？ 如果為 <code>true</code> ，將會記錄以下事件： <ul style="list-style-type: none"> ■ 目標已滿，已達最大容量或訊息計數。 ■ 代理程式不是因為管理指令或傳送確認而捨棄訊息。 ■ 代理程式將訊息移動到停用的訊息佇列。

¹ 可與 `imqcmd update bkr` 指令一起使用

表 14-8 代理程式監視特性 (續)

特性	類型	預設	說明
imq.log.console.stream	字串	ERR	主控台輸出的目標： OUT : stdout ERR : stderr
imq.log.console.output	字串	ERROR WARNING	要寫入至主控台的記錄資訊種類： NONE ERROR WARNING INFO ALL ERROR、WARNING 與 INFO 種類並不包含它們之上的層級，因此必要時必須明確指定。可用豎條符號 () 分隔，指定任意種類組合。
imq.log.file.dirpath	字串	請參閱附錄 A	前往包含記錄檔的路徑
imq.log.file.filename	字串	log.txt	記錄檔名稱
imq.log.file.output	字串	ALL	要寫入至記錄檔的記錄資訊種類： NONE ERROR WARNING INFO ALL ERROR、WARNING 與 INFO 種類並不包含它們之上的層級，因此必要時必須明確指定。可用豎條符號 () 分隔，指定任意種類組合。
imq.log.file.rolloverbytes ¹	整數	-1	記錄檔長度 (以位元組為單位)，一旦超出即需將輸出自動重建至新記錄檔 -1 值表示不限制位元組數目 (不會根據檔案長度進行自動重建)。
imq.log.file.rolloversecs ¹	整數	604800 (一週)	檔案存在時間 (以秒為單位)，一旦超出即需將輸出自動重建至新記錄檔 -1 值表示不限制秒數 (不會根據檔案存在時間進行自動重建)。

¹ 可與 imqcmd update bkr 指令一起使用

表 14-8 代理程式監視特性 (續)

特性	類型	預設	說明
<code>imq.log.syslog.output²</code>	字串	ERROR	要寫入至 <code>syslogd(1M)</code> 的記錄資訊種類： NONE ERROR WARNING INFO ALL ERROR、WARNING 與 INFO 種類並不包含它們之上的層級，因此必要時必須明確指定。可用豎條符號 () 分隔，指定任意種類組合。
<code>imq.log.syslog.facility²</code>	字串	LOG_DAEMON	記錄訊息的 <code>syslog</code> 設備 可能的值對應 <code>syslog(3C)</code> 線上手冊中所列出的值。用於 Message Queue 的適當值包含： LOG_USER LOG_DAEMON LOG_LOCAL0 LOG_LOCAL1 LOG_LOCAL2 LOG_LOCAL3 LOG_LOCAL4 LOG_LOCAL5 LOG_LOCAL6 LOG_LOCAL7
<code>imq.log.syslog.identity²</code>	字串	<code>imqbrokerd_{imq.instanceName}</code>	要在記錄至 <code>syslog</code> 的所有訊息前面放置的識別字串
<code>imq.log.syslog.logpid²</code>	布林值	true	記錄訊息時也要記錄代理程式程序 ID ？
<code>imq.log.syslog.logconsole²</code>	布林值	false	如果無法將訊息傳送至 <code>syslog</code> ，便將它們寫入至系統主控台？

² 僅限 Solaris 平台

表 14-8 代理程式監視特性 (續)

特性	類型	預設	說明
<code>imq.log.timezone</code>	字串	本地時區	記錄時間標記的時區 這些值與 <code>java.util.TimeZone.getTimeZone</code> 方法 使用的值相同。 範例： GMT GMT-8:00 America/LosAngeles Europe/Rome Asia/Tokyo
<code>imq.metrics.enabled</code>	布林值	true	啓用將度量資訊寫入記錄程式的功能？ 不會影響度量訊息的產生 (由 <code>imq.metrics.topic.enabled</code> 控制)。
<code>imq.metrics.interval</code>	整數	-1	將度量資訊寫入記錄程式的時間間隔 (以秒為單位) 不會影響產生度量訊息的時間間隔 (由 <code>imq.metrics.topic.interval</code> 控制)。 -1 值表示無限期 (永不將度量資訊寫入 至記錄程式)。
<code>imq.metrics.topic.enabled</code>	布林值	true	啓用將所產生度量訊息寫入度量主題目 標的功能？ 如果為 false，就會在嘗試訂閱度量主 題目標時，丟出一個用戶端異常。
<code>imq.metrics.topic.interval</code>	整數	60	產生度量訊息至度量主題目標的時間間 隔 (以秒為單位)
<code>imq.metrics.topic.persist</code>	布林值	false	傳送至度量主題目標的度量訊息為永久 性訊息？
<code>imq.metrics.topic.timetolive</code>	整數	300	傳送到度量主題目標之度量訊息的使用 期限 (以秒為單位)

叢集配置特性

表 14-9 列出與代理程式叢集相關的配置特性。

表 14-9 叢集配置的代理程式特性

特性	類型	預設	說明
<code>imq.cluster.brokerlist</code> ¹	字串	無	代理程式位址的清單 此清單由以逗號分隔的一個或多個位址組成。每個位址都可指出叢集中某個代理程式的主機名稱與連接埠對映器之連接埠號，使用的格式為 <i>hostName: portNumber</i> 。 範例： <code>host1:3000,host2:8000,ctrlhost</code>
<code>imq.cluster.hostname</code> ²	字串	無	<code>cluster</code> 連線服務的主機名稱或 IP 位址 如果指定，會置換 <code>cluster</code> 連線服務的 <code>imq.hostname</code> (請參閱表 14-1)。
<code>imq.cluster.port</code> ²	整數	0	<code>cluster</code> 連線服務的連接埠號 0 值指定連接埠對映器應動態配置連接埠號。
<code>imq.cluster.transport</code> ¹	字串	tcp	<code>cluster</code> 連線服務的網路傳輸協定 為了在代理程式之間以安全方式傳送加密的訊息，請將此特性設定為 <code>ssl</code> 。
<code>imq.cluster.url</code> ^{1,3}	字串	無	<code>cluster</code> 配置檔案的 URL (如果存在) 範例： <code>http://webserver/imq/cluster.properties</code> (適用於 Web 伺服器上的檔案) <code>file:/net/mfsserver/imq/cluster.properties</code> (適用於共用磁碟機上的檔案)
<code>imq.cluster.masterbroker</code> ¹	字串	無	叢集主代理程式的主機名稱和連接埠號 (如果存在) 值的格式為 <i>hostName: portNumber</i> ，其中 <i>hostName</i> 是主代理程式的主機名稱， <i>portNumber</i> 是其連接埠對映器的連接埠號。 範例： <code>ctrlhost:7676</code>

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

按字母順序排列的代理程式特性清單

第 253 頁的「按字母順序排列的代理程式特性清單」是按字母順序排列的代理程式配置特性清單，可與本章中的相關表格交叉參照。

表 14-10 按字母順序排列的代理程式特性清單

特性	表格
imq.accesscontrol.enabled	表 14-7
imq.accesscontrol.file.filename	表 14-7
imq.audit.enabled	表 14-7
imq.authentication.basic.user_repository	表 14-7
imq.authentication.client.response.timeout	表 14-7
imq.authentication.type	表 14-7
imq.autocreate.destination.isLocalOnly	表 14-3
imq.autocreate.destination.limitBehavior	表 14-3
imq.autocreate.destination.maxBytesPerMsg	表 14-3
imq.autocreate.destination.maxNumMsgs	表 14-3
imq.autocreate.destination.maxNumProducers	表 14-3
imq.autocreate.destination.maxTotalMsgBytes	表 14-3
imq.autocreate.destination.useDMQ	表 14-3
imq.autocreate.queue	表 14-3
imq.autocreate.queue.consumerFlowLimit	表 14-3
imq.autocreate.queue.localDeliveryPreferred	表 14-3
imq.autocreate.queue.maxNumActiveConsumers	表 14-3
imq.autocreate.queue.maxNumBackupConsumers	表 14-3
imq.autocreate.topic	表 14-3
imq.autocreate.topic.consumerFlowLimit	表 14-3
imq.cluster.brokerlist	表 14-9
imq.cluster.hostname	表 14-9
imq.cluster.masterbroker	表 14-9
imq.cluster.port	表 14-9

表 14-10 按字母順序排列的代理程式特性清單 (續)

特性	表格
imq.cluster.transport	表 14-9
imq.cluster.url	表 14-9
imq.destination.DMQ.truncateBody	表 14-2
imq.destination.logDeadMsgs	表 14-8
imq.hostname	表 14-1
imq.imqcmd.password	表 14-7
imq.keystore.file.dirpath	表 14-7
imq.keystore.file.name	表 14-7
imq.keystore.password	表 14-7
imq.keystore. <i>propertyName</i>	表 14-7
imq.log.console.output	表 14-8
imq.log.console.stream	表 14-8
imq.log.file.dirpath	表 14-8
imq.log.file.filename	表 14-8
imq.log.file.output	表 14-8
imq.log.file.rolloverbytes	表 14-8
imq.log.file.rolloversecs	表 14-8
imq.log.level	表 14-8
imq.log.syslog.facility	表 14-8
imq.log.syslog.identity	表 14-8
imq.log.syslog.logconsole	表 14-8
imq.log.syslog.logpid	表 14-8
imq.log.syslog.output	表 14-8
imq.log.timezone	表 14-8
imq.message.expiration.interval	表 14-2
imq.message.max_size	表 14-2
imq.metrics.enabled	表 14-8
imq.metrics.interval	表 14-8

表 14-10 按字母順序排列的代理程式特性清單 (續)

特性	表格
imq.metrics.topic.enabled	表 14-8
imq.metrics.topic.interval	表 14-8
imq.metrics.topic.persist	表 14-8
imq.metrics.topic.timetolive	表 14-8
imq.passfile.dirpath	表 14-7
imq.passfile.enabled	表 14-7
imq.passfile.name	表 14-7
imq.persist.file.destination.message.filepool.limit	表 14-5
imq.persist.file.message.cleanup	表 14-5
imq.persist.file.message.filepool.cleanratio	表 14-5
imq.persist.file.message.max_record_size	表 14-5
imq.persist.file.sync.enabled	表 14-5
imq.persist.jdbc.brokerid	表 14-6
imq.persist.jdbc.closedburl	表 14-6
imq.persist.jdbc.createdburl	表 14-6
imq.persist.jdbc.driver	表 14-6
imq.persist.jdbc.needpassword	表 14-6
imq.persist.jdbc.opendburl	表 14-6
imq.persist.jdbc.password	表 14-6
imq.persist.jdbc.table.IMQCCREC35	表 14-6
imq.persist.jdbc.table.IMQDEST35	表 14-6
imq.persist.jdbc.table.IMQILIST35	表 14-6
imq.persist.jdbc.table.IMQINT35	表 14-6
imq.persist.jdbc.table.IMQMSG35	表 14-6
imq.persist.jdbc.table.IMQPROPS35	表 14-6
imq.persist.jdbc.table.IMQSV35	表 14-6
imq.persist.jdbc.table.IMQTACK35	表 14-6
imq.persist.jdbc.table.IMQTXN35	表 14-6

表 14-10 按字母順序排列的代理程式特性清單 (續)

特性	表格
imq.persist.jdbc.user	表 14-6
imq.persist.store	表 14-4
imq.ping.interval	表 14-1
imq.portmapper.backlog	表 14-1
imq.portmapper.hostname	表 14-1
imq.portmapper.port	表 14-1
imq.resourceState .count	表 14-2
imq.resourceState .threshold	表 14-2
imq.service.activelist	表 14-1
imq.serviceName .accesscontrol.enabled	表 14-7
imq.serviceName .accesscontrol.file.filename	表 14-7
imq.serviceName .authentication.type	表 14-7
imq.serviceName .max_threads	表 14-1
imq.serviceName .min_threads	表 14-1
imq.serviceName .protocolType.hostname	表 14-1
imq.serviceName .protocolType.port	表 14-1
imq.serviceName .threadpool_model	表 14-1
imq.shared.connectionMonitor_limit	表 14-1
imq.system.max_count	表 14-2
imq.system.max_size	表 14-2
imq.transaction.autorollback	表 14-2
imq.user_repository.ldap.base	表 14-7
imq.user_repository.ldap.gidattr	表 14-7
imq.user_repository.ldap.grpbase	表 14-7
imq.user_repository.ldap.grpfilter	表 14-7
imq.user_repository.ldap.grpsearch	表 14-7
imq.user_repository.ldap.memattr	表 14-7
imq.user_repository.ldap.password	表 14-7

表 14-10 按字母順序排列的代理程式特性清單 (續)

特性	表格
<code>imq.user_repository.ldap.principal</code>	表 14-7
<code>imq.user_repository.ldap.propertyName</code>	表 14-7
<code>imq.user_repository.ldap.server</code>	表 14-7
<code>imq.user_repository.ldap.ssl.enabled</code>	表 14-7
<code>imq.user_repository.ldap.timeout</code>	表 14-7
<code>imq.user_repository.ldap.uidattr</code>	表 14-7
<code>imq.user_repository.ldap.usrfilter</code>	表 14-7

實體目標特性參照

本章提供有關實體目標配置特性的參考資訊。您可在建立或更新實體目標時，設定這些特性。針對自動建立的目標，已在代理程式實例配置檔案中設定為預設值 (請參閱表 14-3)。

實體目標特性

表 15-1 實體目標特性

特性	類型	預設	說明
maxNumMsgs ¹	整數	-1	未使用訊息的最大數目 -1 值表示不限制訊息數目。 對於停用的訊息佇列，預設值是 1000。
maxBytesPerMsg	字串	-1	任何單一訊息的最大容量 (以位元組為單位) 若永久性訊息被拒絕，就會對訊息產生用戶端報告發生異常；若為非永久性訊息，則不會傳送通知。 此值可能會使用下列字尾，以位元組、千位元組或百萬位元組表示： b：位元組 k：千位元組 (1024 位元組) m：百萬位元組 (1024 × 1024 = 1,048,576 位元組) 無字尾的值是以位元組表示；-1 值表示不限制訊息容量。

¹ 在叢集環境中，會套用至目標的各個實例，而非整體套用至叢集中的所有實例

表 15-1 實體目標特性 (續)

特性	類型	預設	說明
			範例： 1600：1600 位元組 1600b：1600 位元組 16k：16 千位元組 (= 16,384 位元組) 16m：16 百萬位元組 (= 16,777,216 位元組) -1：不限制
maxTotalMsgBytes ¹	字串	-1	未使用訊息的最大總記憶體容量 (以位元組為單位) 語法與 maxBytesPerMsg 相同 (請參閱前述內容)。 對於停用的訊息佇列，預設值是 10m。
limitBehavior	字串	REJECT_NEWEST	達到記憶體限制臨界值時的代理程式運作方式： FLOW_CONTROL：減緩產生器速度 REMOVE_OLDEST：捨棄最舊的訊息 REMOVE_LOW_PRIORITY：根據訊息存在時間，捨棄優先權最低的訊息；不會對訊息產生用戶端傳送通知 REJECT_NEWEST：拒絕最新的訊息；只有當訊息為永久性訊息時，才通知訊息產生用戶端發生異常 如果值為 REMOVE_OLDEST 或 REMOVE_LOW_PRIORITY，而且 useDMQ 特性為 true，超過限制的訊息就會移至停用的訊息佇列。對於停用的訊息佇列本身，預設的限制運作方式是 REMOVE_OLDEST，而且無法設為 FLOW_CONTROL。
maxNumProducers ²	整數	-1	目標的最大訊息產生器數目 達到限制時，將無法建立新的產生器。-1 值表示不限制產生器數目。
maxNumActiveConsumers ³	整數	1	從佇列目標進行負載平衡傳送時，使用中訊息用戶的最大數目 -1 值表示不限制用戶數目。在 Sun Java System Message Queue™ Platform Edition 中，此值會限制為 2。
maxNumBackupConsumers ³	整數	0	從佇列目標進行負載平衡傳送時，備份訊息用戶的最大數目 -1 值表示不限制用戶數目。在 Sun Java System Message Queue Platform Edition 中，此值會限制為 1。

¹ 在叢集環境中，會套用至目標的各個實例，而非整體套用於叢集中的所有實例

² 無法針對停用的訊息佇列進行設定

³ 僅限佇列目標

表 15-1 實體目標特性 (續)

特性	類型	預設	說明
consumerFlowLimit	整數	1000	單一批次中傳送給用戶的最大訊息數目 在負載平衡佇列傳送中，此數目是開始進行負載平衡前，路由到使用中用戶的佇列訊息初始數目。目標用戶可以在連線上指定較低的值來置換這項限制。 -1 值表示不限制訊息數目。
isLocalOnly ²	布林值	false	僅限本機傳送？ 此特性僅適用於代理程式叢集的目標，而且目標一經建立之後便無法變更。如果為 true，則目標不會被複製到其他代理程式上，並且只會傳送訊息給本機用戶 (即連線至建立該目標之代理程式的用戶)。
localDeliveryPreferred ^{2,3}	布林值	false	偏好本機傳送？ 此特性僅適用於代理程式叢集中的負載平衡佇列傳送。如果為 true，則只有當本機代理程式上沒有用戶時，才會將訊息傳送給遠端用戶；目標不可設定為僅限於本機傳送 (即 isLocalOnly 必須為 false)。
useDMQ ²	布林值	true	要將停用的訊息傳送給停用的訊息佇列？ 如果為 false，停用的訊息將會被捨棄。

² 無法針對停用的訊息佇列進行設定³ 僅限佇列目標

◆◆◆ 第 16 章

受管理物件屬性參照

本章提供受管理物件屬性的參考資訊。包含下列各節：

- 第 263 頁的「連線工廠屬性」
- 第 269 頁的「目標屬性」
- 第 270 頁的「SOAP 端點屬性」

連線工廠屬性

連線工廠物件的屬性分為幾個種類，在下列各節中有所介紹：

- 第 263 頁的「連線處理」
- 第 266 頁的「用戶端標識」
- 第 266 頁的「可靠性與流量控制」
- 第 268 頁的「佇列瀏覽器和伺服器階段作業」
- 第 268 頁的「標準訊息特性」
- 第 269 頁的「訊息標頭置換」

連線處理

表 16-1 列出連線處理的連線工廠屬性。

表 16-1 連線處理的連線工廠屬性

屬性	類型	預設	說明
imqAddressList	字串	若存在，為現有的 Message Queue™ 3.0 位址；若不存在，則為表 16-2 中的第一個項目	代理程式位址的清單 此清單由以逗號分隔的一個或多個代理程式位址組成。每個位址都會指定 (或表示) 主機名稱、連接埠號，以及用戶端可以連線的代理程式實例的連線服務。位址語法會因連線服務與連接埠指定方法而異；請參閱下面內容以取得詳細資訊。
imqAddressListBehavior	字串	PRIORITY	嘗試連線至代理程式位址的順序： PRIORITY：位址清單中指定的順序 RANDOM：隨機順序 備註 - 如果許多用戶端共用相同的連線工廠，請指定隨機連線順序，以免全都嘗試連線至相同的位址。
imqAddressListIterations	整數	5	在位址清單上嘗試建立或重新建立連線的重複次數 -1 值表示不限制重複次數。
imqPingInterval	整數	30	測試用戶端與代理程式間之連線的時間間隔 (以秒為單位) 0 或 -1 值可停用有規律的連線測試。
imqReconnectEnabled	布林值	false	嘗試重新建立中斷的連線？
imqReconnectAttempts	整數	0	移至位址清單中的下一個位址之前，嘗試連線 (或重新連線) 至每個位址的次數 -1 值表示不限制嘗試連線的次數：會重複嘗試連線至第一個位址，直到成功為止。
imqReconnectInterval	長整數	3000	嘗試重新連線的時間間隔 (以毫秒為單位) 此值適用於連續嘗試一個指定的位址，以及連續依序嘗試清單中連續的位址。 備註 - 過小的值可能會讓代理程式來不及回復；而過大的值則可能會導致令人無法忍受的延遲。
imqSSLIsHostTrusted	布林值	true	信任代理程式所提出的任何憑證？ 若設為 false，Message Queue 用戶端執行階段將會驗證收到的所有憑證。若憑證的簽署者不在用戶端的信任清單存放區中，則驗證將會失敗。 若設為 true，將會略過憑證驗證。例如，使用自我簽署的憑證對軟體進行測試時，這樣設定很有用。 備註：若要使用憑證授權單位簽署的憑證，請將此屬性設定為 false。

`imqAddressList` 屬性的值是以逗號分隔的字串，指定要連線的一個或多個代理程式位址。每個位址的一般語法如下：

scheme://*address*

其中 *scheme* 是表 16-2 第一欄中顯示的其中一個定址機制，而 *address* 為代理程式位址本身。指定位址的確切語法會因定址機制而異，如表格的最後一欄所示。

表 16-2 訊息代理程式定址機制

方案	服務	語法	說明
mq	jms 或 ssljms	<code>[hostName][:portNumber]/[serviceName]</code>	動態指定 jms 或 ssljms 連線服務的連接埠位址清單項目可指定 Message Queue 連接埠對映器的主機名稱與連接埠號。連接埠對映器本身會動態指定連線使用的連接埠。 預設值： <code>hostName = localhost</code> <code>portNumber = 7676</code> <code>serviceName = jms</code> 若為 ssljms 連線服務，則必須明確指定所有變數。
mqtcp	jms	<code>hostName:portNumber /jms</code>	使用 jms 連線服務連線至指定的連接埠略過連接埠對映器，並直接與指定的主機名稱與連接埠號建立 TCP 連線。
mqssl	ssljms	<code>hostName:portNumber /ssljms</code>	使用 ssljms 連線服務連線至指定的連接埠略過連接埠對映器，並直接與指定的主機名稱與連接埠號建立安全的 SSL 連線。
http	httpjms	<code>http://hostName: portNumber/contextRoot /tunnel</code> 如果多個代理程式實例使用相同的通道 Servlet，下面語法可連線至特定 (而非隨機選定) 代理程式實例： <code>http://hostName: portNumber/contextRoot /tunnel? ServerName=hostName :instanceName</code>	使用 httpjms 連線服務連線至指定的連接埠在指定的 URL 與 Message Queue 通道 Servlet 建立 HTTP 連線。代理程式必須配置為存取 HTTP 通道 Servlet。
https	httpsjms	<code>https://hostName: portNumber/contextRoot /tunnel</code> 如果多個代理程式實例使用相同的通道 Servlet，下面語法可連線至特定 (而非隨機選定) 代理程式實例： <code>https://hostName: portNumber/contextRoot /tunnel? ServerName=hostName :instanceName</code>	使用 httpsjms 連線服務連線至指定的連接埠在指定的 URL 建立與 Message Queue 通道 Servlet 的安全 HTTPS 連線。代理程式必須配置為存取 HTTPS 通道 Servlet。

表 16-3 訊息代理程式位址範例

服務	代理程式主機	連接埠	範例位址
未指定	未指定	未指定	無位址 (mq://localhost:7676/jms)
未指定	指定主機	未指定	myBkrHost (mq://myBkrHost:7676/jms)
未指定	未指定	指定連接埠對映器連接埠	1012 (mq://localhost:1012/jms)
ssljms	本地主機	標準連接埠對映器連接埠	mq://localhost:7676/ssljms
ssljms	指定主機	標準連接埠對映器連接埠	mq://myBkrHost:7676/ssljms
ssljms	指定主機	指定連接埠對映器連接埠	mq://myBkrHost:1012/ssljms
jms	本地主機	指定服務連接埠	mqtcp://localhost:1032/jms
ssljms	指定主機	指定服務連接埠	mssql://myBkrHost:1034/ssljms
httpjms	不適用	不適用	http://webrvr1:8085/imq/tunnel
httpsjms	不適用	不適用	https://webrvr2:8090/imq/tunnel

用戶端標識

表 16-4 列出用戶端標識的連線工廠屬性。

表 16-4 用戶端標識的連線工廠屬性

屬性	類型	預設	說明
imqDefaultUsername	字串	guest	以代理程式進行認證時使用的預設使用者名稱
imqDefaultPassword	字串	guest	以代理程式進行認證時使用的預設密碼
imqConfiguredClientID	字串	null	管理配置的用戶端識別碼
imqDisableSetClientID	布林值	false	是否要使用setClientID方法，以防止用戶端變用戶端識別碼？

可靠性與流量控制

表 16-5 列出可靠性與流量控制的連線工廠屬性。

表 16-5 可靠性與流量控制的連線工廠屬性

屬性	類型	預設	說明
<code>imqAckTimeout</code>	字串	0	<p>指定在丟出異常之前，等待代理程式確認的最長時間(以毫秒為單位)</p> <p>0 值表示無逾時(永遠等待)。</p> <p>備註- 有時過小的值會導致發生提早逾時的情況：例如，首次使用安全 (SSL) 連線向 LDAP 使用者儲存庫認證使用者時，可能需要超過 30 秒的時間。</p>
<code>imqConnectionFlowCount</code>	整數	100	<p>度量批次中的有效負載訊息數</p> <p>超過此訊息數時，就會暫時停止傳送有效負載訊息至用戶端，而會傳送所有累積的控制訊息。依照用戶端執行階段的通知，重新繼續傳送有效負載訊息，直至計數再次到達指定數目為止。</p> <p>0 值可停用對訊息傳送進行度量，這可能會導致大量有效負載訊息流量阻斷 Message Queue 控制訊息。</p>
<code>imqConnectionFlowLimitEnabled</code>	布林值	false	要限制連線層級的訊息流量？
<code>imqConnectionFlowLimit</code>	整數	1000	<p>每個連線可傳送和緩衝供使用的最大訊息數</p> <p>擱置的未使用有效負載訊息數(受到由 <code>imqConnectionFlowCount</code> 控制之流量度量的管制) 超過此限制時，連線就會停止傳送訊息。只有當擱置的訊息低於限制時，才會重新繼續傳送。如此可避免用戶端因擱置訊息導致耗盡記憶體，而造成過度負載。</p> <p>如果 <code>imqConnectionFlowLimitEnabled</code> 為 false，則可忽略此屬性。</p>
<code>imqConsumerFlowLimit</code>	整數	100	<p>每個用戶可傳送和緩衝供使用的最大訊息數</p> <p>該用戶擱置未使用的有效負載訊息數超過此限制時，就會停止將訊息傳送給指定的用戶。只有當該用戶的擱置訊息低於 <code>imqConsumerFlowThreshold</code> 指定的百分比時，才會重新繼續傳送。如此可改善多個用戶間的負載平衡，並防止任何用戶獨佔連線，以免讓使用相同連線的其他用戶苦等。</p> <p>您可以為佇列自身的 <code>consumerFlowLimit</code> 屬性設定一個較小的值，以置換此限制(請參閱第 15 章)。請注意，將訊息傳送給一個連線上所有用戶時，必須遵守 <code>imqConnectionFlowLimit</code> 所指定的整體限制。</p>
<code>imqConsumerFlowThreshold</code>	整數	50	<p>在用戶端執行階段緩衝的每位用戶之訊息數，以 <code>imqConsumerFlowLimit</code> 百分比表示，當低於該值時，就可以重新繼續傳送訊息</p>

佇列瀏覽器和伺服器階段作業

表 16-6 列出佇列瀏覽和伺服器階段作業的連線工廠屬性。

表 16-6 佇列瀏覽器和伺服器階段作業的連線工廠屬性

屬性	類型	預設	說明
imqQueueBrowserMaxMessagesPerRetrieve	整數	1000	瀏覽佇列目標的內容時，一次可擷取的最大訊息數 備註 - 此屬性不會影響所瀏覽的訊息總數，而只是影響將訊息分為資料塊以便傳送到用戶端執行階段的方式 (分為數目少但內容多的資料塊，或是分為數目多但內容少的資料塊)。用戶端應用程式一定會收到佇列中的所有訊息。變更該屬性的值可能會影響效能，但不會影響所擷取的資料總數。
imqQueueBrowserRetrieveTimeout	長整數	60000	瀏覽佇列目標的內容時，在丟出異常之前，等待擷取訊息的最長時間 (以毫秒為單位)
imqLoadMaxToServerSession	布林值	true	要將最大訊息數載入伺服器階段作業？ 如果指定為 false，用戶端一次只載入一個訊息。 此屬性僅適用於 JMS 應用程式伺服器設備。

標準訊息特性

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

表 16-7 標準訊息特性的連線工廠屬性

特性	類型	預設	說明
imqSetJMSXUserID	布林值	false	要為產生的訊息設定 JMSXUserID 特性 (傳送訊息的使用者身份識別)？
imqSetJMSXAppID	布林值	false	要為產生的訊息設定 JMSXAppID 特性 (傳送訊息的應用程式身份識別)？
imqSetJMSXProducerTXID	布林值	false	要為產生的訊息設定 JMSXProducerTXID 特性 (產生訊息的作業事件之作業事件識別碼)？
imqSetJMSXConsumerTXID	布林值	false	要為使用的訊息設定 JMSXConsumerTXID 特性 (使用訊息的作業事件之作業事件識別碼)？
imqSetJMSXRcvTimestamp	布林值	false	要為使用的訊息設定 JMSXRcvTimestamp 特性 (訊息傳送給用戶的時間)？

訊息標頭置換

表 16-8 列出用以置換 JMS 訊息標頭欄位的連線工廠屬性。

表 16-8 訊息標頭置換的連線工廠屬性

屬性	類型	預設	說明
imqOverrideJMSDeliveryMode	布林值	false	允許置換用戶端設定的傳送模式？
imqJMSDeliveryMode	整數	2	置換傳送模式的值： 1 非永久性模式 2 永久性模式
imqOverrideJMSExpiration	布林值	false	允許置換用戶端設定的過期時間？
imqJMSExpiration	長整數	0	置換過期時間的值 (以毫秒為單位) 0 值表示不限制過期時間 (訊息永不過期)。
imqOverrideJMSPriority	布林值	false	允許置換用戶端設定的優先層級？
imqJMSPriority	整數	4 (一般)	置換優先權層級的值 (0 至 9)
imqOverrideJMSHeadersToTemporaryDestinations	布林值	false	要將置換套用至暫時目標？

目標屬性

表 16-9 列出可為目標受管理物件設定的屬性。

表 16-9 目標屬性

屬性	類型	預設	說明
imqDestinationName	字串	Untitled_Destination_Object	實體目標名稱 目標名稱只能包含字母數字字元 (無空格)，而且必須以字母字元或底線 () 或錢幣符號 (\$) 開頭。而且開頭不能是 mq 字元。
imqDestinationDescription	字串	無	目標的描述性字串

SOAP 端點屬性

表 16-10 列出用以配置端點 URL 的屬性，以供使用簡單物件存取協定 (SOAP) 的應用程式使用；如需更多資訊，請參閱「Message Queue Developer's Guide for Java Clients」。

表 16-10 SOAP 端點屬性

屬性	類型	預設	說明
<code>imqSOAPEndpointList</code>	字串	無	<p>一個或多個表示 SOAP 端點之 URL 的清單 (以空格相互分隔)，訊息將傳送至這些端點</p> <p>每個 URL 應該有一個可以接收並處理 SOAP 訊息的相關聯 Servlet。</p> <p>範例：</p> <pre>http://www.serv1/ http://www.serv2/</pre> <p>如果該清單指定多個 URL，則會向所有的 URL 廣播訊息。</p>
<code>imqEndpointName</code>	字串	<code>Untitled_Endpoint_Object</code>	SOAP 端點名稱
<code>imqEndpointDescription</code>	字串	無	<p>SOAP 端點的描述性字串</p> <p>範例：</p> <pre>My endpoints for broadcast</pre>

JMS 資源介面特性參照

本章說明 Message Queue™ JMS 資源介面 (JMS RA) 的配置特性，可讓您透過標準 J2EE 連接器架構 (JCA)，以整合 Sun Java System™ Message Queue 與任何 J2EE 1.4 應用程式伺服器。當資源介面外掛至應用程式伺服器之後，可讓部署在該應用程式伺服器中的應用程式使用 Message Queue 來傳送及接收 JMS 訊息。

Message Queue JMS 資源介面透過三個 JavaBean 元件，展現其配置特性：

- ResourceAdapter JavaBean (第 271 頁的「ResourceAdapter JavaBean」) 會影響資源介面整體的運作方式。
- ManagedConnectionFactory JavaBean (第 273 頁的「ManagedConnectionFactory JavaBean」) 會影響資源介面所建立供訊息驅動 Bean (MDB) 使用的連線。
- ActivationSpec JavaBean (第 274 頁的「ActivationSpec JavaBean」) 會影響與訊息傳送系統互動時，代表 MDB 的訊息端點。

若要設定這些實體的特性值，您可以使用應用程式伺服器所提供的工具，以用於配置及部署資源介面及部署 MDB。

本章列出並描述 Message Queue JMS 資源介面的配置特性。它包含下面各節：

ResourceAdapter JavaBean

ResourceAdapter 配置可以配置預設的 JMS 資源介面運作方式。表 17-1 列出並描述用以配置此 JavaBean 的特性。

表 17-1 資源介面特性

特性	類型	預設	說明
addressList	字串	mq://localhost:7676/jms	(必需) 可供連線至 Message Queue 服務的訊息服務位址 等同於 connectionURL (如下所述)；您必須設定其中一個。
connectionURL	字串	mq://localhost:7676/jms	可供連線至 Message Queue 服務的訊息服務位址 等同於 addressList (如上所述)；您必須設定其中一個。
userName	字串	guest	(必需) 可供連線至 Message Queue 服務的預設使用者名稱
password	字串	guest	(必需) 可供連線至 Message Queue 服務的預設密碼
addressListBehavior	字串	PRIORITY	嘗試連線至 Message Queue 服務的順序： PRIORITY：位址清單中指定的順序 RANDOM：隨機順序 備註 - 不論針對此特性設定的值為何，連線失敗後的重新連線嘗試都會從連線失敗的代理程式開始，並依次連線該位址清單中的後續位址。
addressListIterations	整數	1	在位址清單上嘗試建立或重新建立連線的重複次數
reconnectEnabled	布林值	false	嘗試重新建立斷開的連線？
reconnectAttempts	整數	6	移至位址清單中的下一個位址之前，嘗試重新連線至每個位址的次數
reconnectInterval	長整數	30000	嘗試重新連線的時間間隔 (以毫秒為單位)

ManagedConnectionFactory JavaBean

管理的連線工廠會定義資源介面提供給訊息驅動 Bean 的連線。表 17-2 顯示 ManagedConnectionFactory JavaBean 的特性；若設定這些特性，它們會置換 ResourceAdapter JavaBean 的對應特性。

表 17-2 管理的連線工廠特性

特性	類型	預設	說明
addressList	字串	繼承自 ResourceAdapter JavaBean (請參閱表 17-1)	可供連線至 Message Queue 服務的訊息服務位址清單
userName	字串	guest	(選擇性) 可供連線至 Message Queue 服務的使用者名稱
password	字串	guest	(選擇性) 可供連線至 Message Queue 服務的密碼
clientID	字串	無	可供連線至 Message Queue 服務的用戶端識別碼
addressListBehavior	字串	PRIORITY	嘗試連線至 Message Queue 服務的順序： PRIORITY：位址清單中指定的順序 RANDOM：隨機順序 備註 - 不論針對此特性設定的值為何，連線失敗後的重新連線嘗試都會從連線失敗的代理程式開始，並依次連線該位址清單中的後續位址。
addressListIterations	整數	1	在位址清單上嘗試建立或重新建立連線的重複次數
reconnectEnabled	布林值	false	嘗試重新建立斷開的連線？
reconnectAttempts	整數	6	移至位址清單中的下一個位址之前，嘗試重新連線至每個位址的次數
reconnectInterval	長整數	30000	嘗試重新連線的時間間隔 (以毫秒為單位)

ActivationSpec JavaBean

表 17-3 顯示 ActivationSpec JavaBean 的可配置特性。應用程式伺服器會使用這些特性指示資源介面啟動訊息端點，並將它與訊息驅動 Bean 關聯。

表 17-3 啟動規格特性

特性	類型	預設	說明
addressList ¹	字串	繼承自 ResourceAdapter JavaBean	(選擇性) 可供連線至 Message Queue 服務的訊息服務位址
destination ²	字串	無	(必需) 使用訊息時所在的目標名稱 此值必須是 Message Queue 目標受管理物件的 destinationName 特性。
destinationType ²	字串	無	(必需) destination 特性指定的目標類型： javax.jms.Queue：佇列目標 javax.jms.Topic：主題目標
messageSelector ²	字串	無	(選擇性) 可供篩選傳送到用戶之訊息的訊息選擇器
subscriptionName ²	字串	無	長期訂閱的名稱 若 subscriptionDurability 是設為 Durable，則必須設定此特性。
subscriptionDurability ²	字串	NonDurable	主題目標用戶的長期性： Durable：長期用戶 NonDurable：非長期用戶 只有當 destinationType 是設為 javax.jms.Topic 時，此特性才有效；此特性對於非長期訂閱是選擇性的，對於長期訂閱則是必需的。若設為 Durable，也必須設定 clientID 和 subscriptionName 特性。
clientId ²	字串	無	可供連線至 Message Queue 服務的用戶端 ID 若 subscriptionDurability 是設為 Durable，則必須設定此特性。

¹ Message Queue JMS 資源介面專用的特性

² 標準 Enterprise JavaBean (EJB) 和 J2EE 連接器架構 (CA) 特性

表 17-3 啓動規格特性 (續)

特性	類型	預設	說明
acknowledgeMode ²	字串	Auto-acknowledge	(選擇性) 確認模式： Auto-acknowledge： Auto-acknowledge 模式 Dups-ok-acknowledge： Dups-OK-acknowledge 模式
customAcknowledgeMode	字串	無	MDB 訊息使用的確認模式 有效值為 No_acknowledge 或空值。 您只能針對未處理作業事件的非長期主題訂閱使用 no-acknowledge 模式；如果將此設定用於已處理作業事件訂閱或長期訂閱，將無法啓動訂閱。
endpointExceptionRedeliveryAttempts	整數	6	當 MDB 在傳送訊息期間丟出異常時，要重新傳送訊息的次數
sendUndeliverableMsgsToDMQ	布林值	true	當 MDB 丟出執行階段異常而且嘗試重新傳送的次數超出 endpointExceptionRedeliveryAttempts 值時，是否將訊息放入停用的訊息佇列中？ 如果是 false，Message Queue 代理程式會嘗試向任何有效的用戶 (包含此 MDB) 重新傳送訊息。

² 標準 Enterprise JavaBean (EJB) 和 J2EE 連接器架構 (CA) 特性

度量參照

本章說明 Message Queue™ 訊息代理程式可提供用於監視、調校和診斷用途的度量資訊。此類資訊透過以下各種方式提供：

- 在記錄檔中 (請參閱第 168 頁的「將度量資料傳送到記錄檔」)
- 透過指令公用程式的 `metrics bkr` 指令 (請參閱第 223 頁的「代理程式管理」)
- 在傳送到度量主題目標的度量訊息中 (請參閱第 174 頁的「撰寫應用程式來監視代理程式」)

本章中的表格列出可用度量資訊的種類，以及提供度量資訊的形式。對於透過指令公用程式之 `metrics bkr` 指令提供的度量，表格會列出可請求的度量類型；對於在度量訊息中提供的度量，表格會列出要將度量傳送到其中的度量主題目標。本章包含下列各節：

- 第 277 頁的「JVM 度量」
- 第 278 頁的「代理程式通用度量」
- 第 279 頁的「連線服務度量」
- 第 281 頁的「目標度量」

JVM 度量

表 18-1 顯示代理程式為代理程式程序 JVM (Java 虛擬機器) 堆疊報告的度量資訊。

表 18-1 JVM 度量

度量數目	說明	記錄檔 ?	metrics bkr 度量類型	度量主題
JVM 堆疊：記憶體總容量	目前的記憶體總容量 (以位元組為單位)	是	cxn	mq.metrics.jvm
JVM 堆疊：可用記憶體	目前的可用記憶體容量 (以位元組為單位)	是	cxn	mq.metrics.jvm
JVM 堆疊：最大記憶體	可允許的最大堆疊容量 (以位元組為單位)	是	無	mq.metrics.jvm

代理程式通用度量

表 18-2 顯示代理程式報告的代理程式通用度量資訊。

表 18-2 代理程式通用度量

度量數目	說明	記錄檔 ?	metrics bkr 度量類型	度量主題
連線				
連線數目	所有連線服務的目前連線數目總數	是	cxn	mq.metrics.broker
執行緒數目	所有連線服務的目前執行緒數目總數	是	cxn	無
最小執行緒數目	所有連線服務的最小執行緒數目總數	是	cxn	無
最大執行緒數目	所有連線服務的最大執行緒數目總數	是	cxn	無
儲存的訊息				
訊息數目	目前儲存在記憶體和永久性存放區中的有效負載訊息數目	不需要	無 ¹	mq.metrics.broker
訊息容量總數	目前儲存在記憶體和永久性存放區中的有效負載訊息容量總數 (以位元組為單位)	不需要	無 ¹	mq.metrics.broker
訊息流量				
傳入的訊息數目	自代理程式啟動後，累積的已接收有效負載訊息數目	是	ttl	mq.metrics.broker
傳出的訊息數目	自代理程式啟動後，累積的已傳送有效負載訊息數目	是	ttl	mq.metrics.broker
訊息傳入的速率	目前傳入代理程式的有效負載訊息流量速率	是	rts	無
訊息傳出的速率	目前傳出代理程式的有效負載訊息流量速率	是	rts	無
傳入的訊息容量	自代理程式啟動後，累積的已接收有效負載訊息容量 (以位元組為單位)	是	ttl	mq.metrics.broker
傳出的訊息容量	自代理程式啟動後，累積的已傳送有效負載訊息容量 (以位元組為單位)	是	ttl	mq.metrics.broker
訊息容量傳入的速率	目前傳入代理程式之有效負載訊息容量的流量速率	是	rts	無
訊息容量傳出的速率	目前傳出代理程式之有效負載訊息容量的流量速率	是	rts	無
傳入的資料封包數目	自代理程式啟動後，累積的已接收有效負載和控制資料封包數目	是	ttl	mq.metrics.broker

¹ 改用 query bkr 指令

表 18-2 代理程式通用度量 (續)

度量數目	說明	記錄檔?	metrics bkr 度量類型	度量主題
傳出的資料封包數目	自代理程式啟動後，累積的已傳送有效負載和控制資料封包數目	是	ttl	mq.metrics.broker
資料封包傳入的速率	目前傳入代理程式之有效負載和控制資料封包流量的速率	是	rts	無
資料封包傳出的速率	目前傳出代理程式之有效負載和控制資料封包流量的速率	是	rts	無
傳入的資料封包容量	自代理程式啟動後，累積的已接收有效負載和控制資料封包容量 (以位元組為單位)	是	ttl	mq.metrics.broker
傳出的資料封包容量	自代理程式啟動後，累積的已傳送有效負載和控制資料封包容量 (以位元組為單位)	是	ttl	mq.metrics.broker
資料封包容量傳入的速率	目前傳入代理程式之有效負載和控制資料封包容量的流量速率	是	rts	無
資料封包容量傳出的速率	目前傳出代理程式之有效負載和控制資料封包容量的流量速率	是	rts	無
目標				
目標數目	目前的實體目標數目	不需要	無	mq.metrics.broker

連線服務度量

表 18-3 顯示代理程式報告的個別連線服務之度量資訊。

表 18-3 連線服務度量

度量數目	說明	記錄檔?	metrics svc 度量類型	度量主題
連線				
連線數目	目前的連線數目	不需要	cxn ¹	無
執行緒數目	目前的執行緒數目	不需要	cxn ¹	無
最小執行緒數目	指定給服務的最小執行緒數目	不需要	cxn	無
最大執行緒數目	指定給服務的最大執行緒數目	不需要	cxn	無
訊息流量				

¹ 也可以使用 query svc 指令來取得

表 18-3 連線服務度量 (續)

度量數目	說明	記錄檔?	metrics svc 度量類型	度量主題
傳入的訊息數目	自代理程式啟動後，透過連線服務接收的累積有效負載訊息數目	不需要	ttl	無
傳出的訊息數目	自代理程式啟動後，透過連線服務傳送的累積有效負載訊息數目	不需要	ttl	無
訊息傳入的速率	目前透過連線服務傳入代理程式的有效負載訊息流量速率	不需要	rts	無
訊息傳出的速率	目前透過連線服務傳出代理程式的有效負載訊息流量速率	不需要	rts	無
傳入的訊息容量	自代理程式啟動後，透過連線服務接收的累積有效負載訊息容量 (以位元組為單位)	不需要	ttl	無
傳出的訊息容量	自代理程式啟動後，透過連線服務傳送的累積有效負載訊息容量 (以位元組為單位)	不需要	ttl	無
訊息容量傳入的速率	目前透過連線服務傳入代理程式之有效負載訊息容量的流量速率	不需要	rts	無
訊息容量傳出的速率	目前透過連線服務傳出代理程式之有效負載訊息容量的流量速率	不需要	rts	無
傳入的資料封包數目	自代理程式啟動後，透過連線服務接收的累積有效負載和控制資料封包數目	不需要	ttl	無
傳出的資料封包數目	自代理程式啟動後，透過連線服務傳送的累積有效負載和控制資料封包數目	不需要	ttl	無
資料封包傳入的速率	目前透過連線服務傳入代理程式之有效負載和控制資料封包的流量速率	不需要	rts	無
資料封包傳出的速率	目前透過連線服務傳出代理程式之有效負載和控制資料封包的流量速率	不需要	rts	無
傳入的資料封包容量	自代理程式啟動後，透過連線服務接收之有效負載和控制資料封包的累積容量 (以位元組為單位)	不需要	ttl	無
傳出的資料封包容量	自代理程式啟動後，透過連線服務傳送之有效負載和控制資料封包的累積容量 (以位元組為單位)	不需要	ttl	無
資料封包容量傳入的速率	目前透過連線服務傳入代理程式之有效負載和控制資料封包容量的流量速率	不需要	rts	無
資料封包容量傳出的速率	目前透過連線服務傳出代理程式之有效負載和控制資料封包容量的流量速率	不需要	rts	無

目標度量

表 18-4 顯示代理程式報告的個別目標之度量資訊。

表 18-4 目標度量

度量數目	說明	記錄檔？	metrics dst 度量類型	度量主題
訊息用戶				
用戶數目	目前所關聯的訊息用戶數目 對於佇列目標，此屬性包含使用中用戶和備份用戶。對於主題目標，它包含非長期訂閱者和(使用中及非使用中的)長期訂閱者，等於「使用中用戶的數目」。	不需要	con	mq.metrics.destination.queue. queueNamemq.metrics.destination.topic. topicName
用戶的尖峰數目	自代理程式啟動後，所關聯訊息用戶的尖峰數目 對於佇列目標，此屬性包含使用中用戶和備份用戶。對於主題目標，它包含非長期訂閱者和(使用中及非使用中的)長期訂閱者，等於「使用中用戶的尖峰數目」。	不需要	con	mq.metrics.destination.queue. queueNamemq.metrics.destination.topic. topicName
用戶的平均數目	自代理程式啟動後，所關聯訊息用戶的平均數目 對於佇列目標，此屬性包含使用中用戶和備份用戶。對於主題目標，它包含非長期訂閱者和(使用中及非使用中的)長期訂閱者，等於「使用中用戶的平均數目」。	不需要	con	mq.metrics.destination.queue. queueNamemq.metrics.destination.topic. topicName

表 18-4 目標度量 (續)

度量數目	說明	記錄檔?	metrics dst 度量類型	度量主題
使用中用戶的數目	目前所關聯的使用中訊息用戶數目 對於主題目標，此屬性包含非長期訂閱者和(使用中及非使用中的)長期訂閱者，等於「用戶數目」。	不需要	con	mq.metrics.destination.queue.queueName mq.metrics.destination.topic.topicName
使用中用戶的尖峰數目	自代理程式啟動後，所關聯使用中訊息用戶的尖峰數目 對於主題目標，此屬性包含非長期訂閱者和(使用中及非使用中的)長期訂閱者，等於「用戶的尖峰數目」。	不需要	con	mq.metrics.destination.queue.queueName mq.metrics.destination.topic.topicName
使用中用戶的平均數目	自代理程式啟動後，所關聯使用中訊息用戶的平均數目 對於主題目標，此屬性包含非長期訂閱者和(使用中及非使用中的)長期訂閱者，等於「用戶的平均數目」。	不需要	con	mq.metrics.destination.queue.queueName mq.metrics.destination.topic.topicName
備份用戶的數目 ¹	目前所關聯的備份訊息用戶數目	不需要	con	mq.metrics.destination.queue.queueName mq.metrics.destination.topic.topicName
備份用戶的尖峰數目 ¹	自代理程式啟動後，所關聯備份訊息用戶的尖峰數目	不需要	con	mq.metrics.destination.queue.queueName mq.metrics.destination.topic.topicName
備份用戶的平均數目 ¹	自代理程式啟動後，所關聯備份訊息用戶的平均數目	不需要	con	mq.metrics.destination.queue.queueName mq.metrics.destination.topic.topicName
儲存的訊息				
訊息數目	目前儲存在記憶體和永久性存放區中的訊息數目	不需要	conttlrts ²	mq.metrics.destination.queue.queueName mq.metrics.destination.topic.topicName

¹ 僅限佇列目標² 也可以使用 query dst 指令來取得

表 18-4 目標度量 (續)

度量數目	說明	記錄檔?	metrics dst 度量類型	度量主題
訊息尖峰數目	自代理程式啟動後，儲存在記憶體和永久性存放區中之訊息的尖峰數目	不需要	contlrts	mq.metrics.destination.queue.queueName mq.metrics.destination.topic.topicName
訊息平均數目	自代理程式啟動後，儲存在記憶體和永久性存放區中之訊息的平均數目	不需要	contlrts	mq.metrics.destination.queue.queueName mq.metrics.destination.topic.topicName
訊息容量總數	目前儲存在記憶體和永久性存放區中之訊息的容量總數 (以位元組為單位)	不需要	tllrts ²	mq.metrics.destination.queue.queueName mq.metrics.destination.topic.topicName
訊息的尖峰容量總數	自代理程式啟動後，儲存在記憶體和永久性存放區中之訊息的尖峰容量總數 (以位元組為單位)	不需要	tllrts	mq.metrics.destination.queue.queueName mq.metrics.destination.topic.topicName
訊息的平均容量	自代理程式啟動後，儲存在記憶體和永久性存放區中之訊息的平均容量總數 (以位元組為單位)	不需要	tllrts	mq.metrics.destination.queue.queueName mq.metrics.destination.topic.topicName
訊息流量				
傳入的訊息數目	自代理程式啟動後，累積的已接收訊息數目	不需要	tll	mq.metrics.destination.queue.queueName mq.metrics.destination.topic.topicName
傳出的訊息數目	自代理程式啟動後，累積的已傳送訊息數目	不需要	tll	mq.metrics.destination.queue.queueName mq.metrics.destination.topic.topicName
傳入的訊息容量	自代理程式啟動後，累積的已接收訊息容量 (以位元組為單位)	不需要	tll	mq.metrics.destination.queue.queueName mq.metrics.destination.topic.topicName
傳出的訊息容量	自代理程式啟動後，累積的已傳送訊息容量 (以位元組為單位)	不需要	tll	mq.metrics.destination.queue.queueName mq.metrics.destination.topic.topicName
訊息的尖峰容量	自代理程式啟動後，所接收的最大單一訊息的容量 (以位元組為單位)	不需要	tllrts	mq.metrics.destination.queue.queueName mq.metrics.destination.topic.topicName

² 也可以使用 query dst 指令來取得

目標度量

表 18-4 目標度量 (續)

度量數目	說明	記錄檔 ?	metrics dst 度量類型	度量主題
訊息傳入的速率	目前已接收訊息流量的速率	不需要	rts	無
訊息傳出的速率	目前已傳送訊息流量的速率	不需要	rts	無
訊息容量傳入的速率	目前已接收訊息容量的流量速率	不需要	rts	無
訊息容量傳出的速率	目前已傳送訊息容量的流量速率	不需要	rts	無
磁碟使用				
保留的磁碟 ³	為目標保留的磁碟空間 (以位元組為單位)	不需要	dsk	mq.metrics.destination.queue.queueName mq.metrics.destination.topic.topicName
已使用的磁碟 ³	目前目標所使用的磁碟空間 (以位元組為單位)	不需要	dsk	mq.metrics.destination.queue.queueName mq.metrics.destination.topic.topicName
磁碟使用率 ³	為目標保留之磁碟空間的磁碟使用率	不需要	dsk	mq.metrics.destination.queue.queueName mq.metrics.destination.topic.topicName

³ 僅限檔案式永久性

第 4 部分

附錄

- 附錄 A
- 附錄 B
- 附錄 C
- 附錄 D

Message Queue™ 資料的特定平台位置

Sun Java System™ Message Queue 資料在不同作業系統平台上會儲存在不同位置。下表顯示各式不同類型的 Message Queue 資料，在下列平台的位置：

- 第 287 頁的「Solaris」
- 第 288 頁的「Linux」
- 第 289 頁的「Windows」

下列表格中，*instanceName* 表示與資料相關的代理程式實例名稱。

Solaris

表 A-1 顯示 Solaris 作業系統上 Message Queue 資料的位置。如果您透過獨立版本的 Sun Java System Application Server 在 Solaris 上使用 Message Queue，其目錄結構將與第 289 頁的「Windows」中所述的結構相似。

表 A-1 Solaris 平台上的 Message Queue 資料位置

資料種類	位置
代理程式實例配置特性	<code>/var/imq/instances/<i>instanceName</i> /props/config.properties</code>
代理程式配置檔案範本	<code>/usr/share/lib/imq/props/broker/</code>
永久性存放區 (訊息、目標、長期訂閱、作業事件)	<code>/var/imq/instances/<i>instanceName</i> /fs350</code> 或 JDBC 可存取的資料存放區
代理程式實例記錄檔目錄 (預設位置)	<code>/var/imq/instances/<i>instanceName</i> /log/</code>
受管理物件 (物件存放區)	您選擇的本機目錄或 LDAP 伺服器
安全性：使用者儲存庫	<code>/var/imq/instances/<i>instanceName</i> /etc/passwd</code> 或 LDAP 伺服器

表 A-1 Solaris 平台上的 Message Queue 資料位置 (續)

資料種類	位置
安全性：存取控制檔案 (預設位置)	<i>/var/imq/instances/instanceName</i> <i>/etc/accesscontrol.properties</i>
安全性：密碼檔案目錄 (預設位置)	<i>/var/imq/instances/instanceName /etc/</i>
安全性：範例密碼檔案	<i>/etc/imq/passfile.sample</i>
安全性：代理程式金鑰庫檔案位置	<i>/etc/imq/</i>
JavaDoc API 文件	<i>/usr/share/javadoc/imq/index.html</i>
範例應用程式與配置	<i>/usr/demo/imq/</i>
Java 歸檔 (.jar)、Web 歸檔 (.war) 與資源 介面歸檔 (.rar) 檔案	<i>/usr/share/lib/</i>

Linux

表 A-2 顯示 Linux 作業系統上 Message Queue 資料的位置。

表 A-2 Linux 平台上的 Message Queue 資料位置

資料種類	位置
代理程式實例配置特性	<i>/var/opt/sun/mq/instances/instanceName</i> <i>/props/config.properties</i>
代理程式配置檔案範本	<i>/opt/sun/mq/private/share/lib/props/</i>
永久性存放區 (訊息、目標、長期 訂閱、作業事件)	<i>/var/opt/sun/mq/instances/instanceName /fs350/</i> 或 JDBC 可存 取的資料存放區
代理程式實例記錄檔目錄 (預設位 置)	<i>/var/opt/sun/mq/instances/instanceName /log/</i>
受管理物件 (物件存放區)	您選擇的本機目錄或 LDAP 伺服器
安全性：使用者儲存庫	<i>/var/opt/sun/mq/instances/instanceName /etc/passwd</i> 或 LDAP 伺服器
安全性：存取控制檔案 (預設位 置)	<i>/var/opt/sun/mq/instances/instanceName</i> <i>/etc/accesscontrol.properties</i>
安全性：密碼檔案目錄 (預設位 置)	<i>/var/opt/sun/mq/instances/instanceName /etc/</i>
安全性：範例密碼檔案	<i>/etc/opt/sun/mq/passfile.sample</i>

表 A-2 Linux 平台上的 Message Queue 資料位置 (續)

資料種類	位置
安全性：代理程式金鑰庫檔案位置	/etc/opt/sun/mq/
JavaDoc API 文件	/opt/sun/mq/javadoc/index.html
範例應用程式與配置	/opt/sun/mq/examples/
Java 歸檔 (.jar)、Web 歸檔 (.war) 與資源介面歸檔 (.rar) 檔案	/opt/sun/mq/share/lib/
共用程式庫 (.so) 檔案	/opt/sun/mq/lib/

Windows

表 A-3 顯示 Windows 作業系統上 Message Queue 資料的位置。當 Message Queue 隨附於獨立版本的 Sun Java System Application Server 時，此表格也適用於 Solaris 平台。該版本的應用程式伺服器並不是隨附於 Solaris 或 Sun Java Enterprise System。使用表 A-3 中的路徑名稱，但將斜線字元的方向從 Windows 反斜線 (\) 變更為 Solaris 正斜線 (/)。請參閱第 25 頁的「目錄變數慣例」以取得 IMQ_HOME 和 IMQ_VARHOME 目錄變數的定義。

表 A-3 Windows 平台上的 Message Queue 資料位置

資料種類	位置
代理程式實例配置特性	IMQ_VARHOME\instances\ <i>instanceName</i> \props\config.properties
代理程式配置檔案範本	IMQ_HOME\lib\props\broker\
永久性存放區 (訊息、目標、長期訂閱、作業事件)	IMQ_VARHOME\instances\ <i>instanceName</i> \fs350\ 或 JDBC 可存取的資料存放區
代理程式實例記錄檔目錄 (預設位置)	IMQ_VARHOME\instances\ <i>instanceName</i> \log\
受管理物件 (物件存放區)	您選擇的本機目錄或 LDAP 伺服器
安全性：使用者儲存庫	IMQ_VARHOME\instances\ <i>instanceName</i> \etc\passwd 或 LDAP 伺服器
安全性：存取控制檔案 (預設位置)	IMQ_VARHOME\instances\ <i>instanceName</i> \etc\accesscontrol.properties
安全性：密碼檔案目錄 (預設位置)	IMQ_HOME\etc\
安全性：範例密碼檔案	IMQ_HOME\etc\passfile.sample
安全性：代理程式金鑰庫檔案位置	IMQ_HOME\etc\

表 A-3 Windows 平台上的 Message Queue 資料位置 (續)

資料種類	位置
JavaDoc API 文件	IMQ_HOME\javadoc\index.html
範例應用程式與配置	IMQ_HOME\demo\
Java 歸檔 (.jar)、Web 歸檔 (.war) 與資源介面歸檔 (.rar) 檔案	IMQ_HOME\lib\

Message Queue™ 介面的穩定性

Sun Java System™ Message Queue 使用許多介面以協助管理員自動進行工作。本附錄可根據穩定性對介面進行分類。介面愈穩定，在後續版本中變更的可能性愈小。

附錄中沒有列出來的介面都是私有的介面，且不提供給客戶使用。

分類機制

附錄 B 中描述穩定性分類機制。

表 B-1 介面穩定性分類機制

分類	說明
私有	不由客戶直接使用。可在任何版次中變更或移除。
改進中	由客戶使用。主要發行版本 (如 3.0、4.0) 或次要發行版本 (如 3.1、3.2) 可能會有不相容的變更。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。僅主要發行版本 (如 3.0 或 4.0) 可能會有不相容的變更。
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。主要發行版本 (如 3.0、4.0) 或次要發行版本 (如 3.1、3.2) 可能會有不相容的變更。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。

介面穩定性

附錄 B 列出各種介面和介面的分類。

表 B-2 Message Queue 介面的穩定性

介面	分類
指令行介面	
imqbrokerd 指令行介面	改進中
imqadmin 指令行介面	不穩定
imqcmd 指令行介面	改進中
imqdbmgr 指令行介面	不穩定
imqkeytool 指令行介面	改進中
imqobjmgr 指令行介面	改進中
imqusermgr 指令行介面	不穩定
從 imqbrokerd、imqadmin、imqcmd、imqdbmgr、imqkeytool、imqobjmgr 和 imqusermgr 的輸出	不穩定
指令	
imqobjmgr 指令檔案	改進中
imqbrokerd 指令	穩定
imqadmin 指令	不穩定
imqcmd 指令	穩定
imqdbmgr 指令	不穩定
imqkeytool 指令	穩定
imqobjmgr 指令	穩定
imqusermgr 指令	不穩定
API	
JMS API (javax.jms)	標準
JAXM API (javax.xml)	標準
C-API	改進中
C-API 環境變數	不穩定
訊息型監視 API	改進中

表 B-2 Message Queue 介面的穩定性 (續)

介面	分類
受管理物件 API (com.sun.messaging)	改進中
.jar 與 .war 檔案	
imq.jar 位置和名稱	穩定
jms.jar 位置和名稱	改進中
imqbroker.jar 位置和名稱	私有
imqutil.jar 位置和名稱	私有
imqadmin.jar 位置和名稱	私有
imqservlet.jar 位置和名稱	改進中
imqhttp.war 位置和名稱	改進中
imqhttps.war 位置和名稱	改進中
imqjmsra.rar 位置和名稱	改進中
imqxm.jar 位置和名稱	改進中
jaxm-api.jar 位置和名稱	改進中
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
檔案	
代理程式記錄檔的位置和內容格式	不穩定
密碼檔案	不穩定
accesscontrol.properties 檔案	不穩定
系統目標	
mq.sys.dmq 目標	穩定
mq.metrics.* 目標	改進中
配置特性	
Message Queue JMS 資源介面配置特性	改進中

表 B-2 Message Queue 介面的穩定性 (續)

介面	分類
Message Queue JMS 資源介面 JavaBean 和 ActivationSpec 配置特性	改進中
訊息特性和格式	
停用的訊息佇列訊息特性，JMSXDeliveryCount	標準
停用的訊息佇列訊息特性，JMS_SUN_*	改進中
Message Queue 用戶端訊息特性：JMS_SUN_*	改進中
度量或監視訊息的 JMS 訊息格式	改進中
其他	
Message Queue JMS 資源介面套裝軟體，com.sun.messaging.jms.ra	改進中
儲存永久性訊息的 JDBC 模式	改進中

HTTP/HTTPS 支援

Message Queue™, Enterprise Edition 包含對 Java 用戶端的支援，以便藉由 HTTP 或安全 HTTP (HTTPS) 傳輸 (而不是直接透過 TCP 連線) 與代理程式進行通訊。C 用戶端不支援 HTTP/HTTPS。

本附錄描述用於啓用此支援的架構，並介紹讓用戶端使用 HTTP 型連線進行 Message Queue 訊息傳送所需的設定工作。它包含下面各節：

- 第 295 頁的「HTTP/HTTPS 支援架構」
- 第 297 頁的「啓用 HTTP 支援」
- 第 303 頁的「啓用 HTTPS 支援」
- 第 313 頁的「疑難排解」

HTTP/HTTPS 支援架構

Message Queue 訊息傳送可在 HTTP/HTTPS 連線的上層執行。由於防火牆通常會允許 HTTP/HTTPS 連線通過防火牆，因此允許存在用戶端應用程式與代理程式被防火牆分隔的狀況。

圖 C-1 顯示提供 HTTP/HTTPS 支援所需的主要元件。

- 在用戶端，HTTP 或 HTTPS 傳輸驅動程式會將 Message Queue 訊息封裝在 HTTP 請求中，並確保這些請求會以正確的順序傳送至 Web 伺服器/應用程式伺服器。
- 如有必要，用戶端可使用 HTTP 代理伺服器與代理程式進行通訊。代理伺服器的位址在啓動用戶端時使用指令行選項指定。如需更多資訊，請參閱第 303 頁的「使用 HTTP 代理伺服器」。
- HTTP 或 HTTPS 通道 Servlet (均隨附於 Message Queue) 會載入至 Web 伺服器/應用程式伺服器中，用來將有效負載訊息從用戶端 HTTP 請求中取出，然後將它們轉寄至代理程式。HTTP/HTTPS 通道 Servlet 也會將代理程式訊息傳送回用戶端，以回應用戶端發出的 HTTP 請求。單一 HTTP/HTTPS 通道 Servlet 可用於存取多個代理程式。

- 在代理程式端，httpjms 或 httpsjms 連線服務會展開從相應通道 Servlet 傳送的內送訊息，並對這些訊息進行多工解碼處理。
- 如果 Web 伺服器/應用程式伺服器發生故障並重新啟動，則所有連線均會復原且不會影響用戶端。如果代理程式發生故障並重新啟動，則系統會丟出異常，且用戶端必須重新建立其連線。如果 Web 伺服器/應用程式伺服器與代理程式均發生故障(這種情況不太可能發生)，且未重新啟動代理程式，則 Web 伺服器/應用程式伺服器會復原用戶端連線，並繼續等待代理程式連線，而不會通知用戶端。若要避免這種情況，請務必重新啟動代理程式。

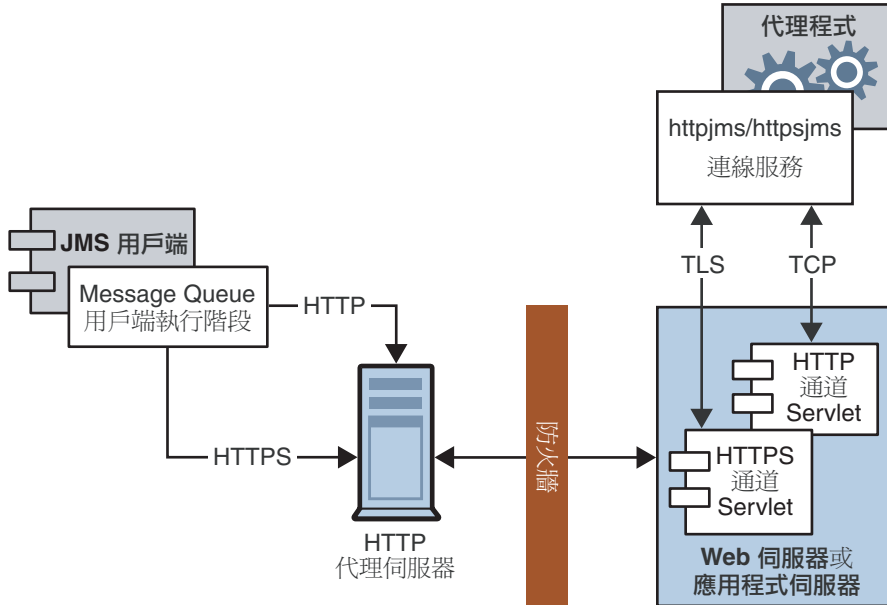


圖 C-1 HTTP/HTTPS 支援架構

如圖 C-1 中所示，HTTP 和 HTTPS 支援的架構非常相似。主要區別在於，對於 HTTPS (httpsjms 連線服務)，通道 Servlet 與用戶端應用程式以及與代理程式兩者之間，都具有安全的連線。

與代理程式的安全連線是透過已啟用 SSL 的通道 Servlet (Message Queue 的 HTTPS 通道 Servlet) 所提供。此通道 Servlet 會將自我簽署的憑證傳送至請求連線的任何代理程式。代理程式使用此憑證來建立與 HTTPS 通道 Servlet 的加密連線。一旦建立此連線，用戶端應用程式和 Web 伺服器/應用程式伺服器即透過用戶端應用程式與通道 Servlet 之間的安全連線進行通訊。

啓用 HTTP 支援

以下各節描述啓用 HTTP 支援所需的步驟。

▼ 啓用 HTTP 支援

- 1 部署 HTTP 通道 Servlet。您可以在下列伺服器上部署 HTTP 通道 Servlet：
 - Sun Java System™ Web Server
 - Sun Java System Application Server
- 2 配置代理程式的 `httpjms` 連線服務，並啓動代理程式。
- 3 配置 HTTP 連線。

步驟 1. 部署 HTTP 通道 Servlet

您可以將 HTTP 通道 Servlet 部署為 Sun Java System Web Server 或 Sun Java System Application Server 上的 Web 歸檔 (.war) 檔案。

將 HTTP 通道 Servlet 部署為 .war 檔案，需要使用 Web 伺服器/應用程式伺服器所提供的部署機制。HTTP 通道 Servlet .war 檔案 (`imqhttp.war`) 位於包含 .jar、.war 和 .rar 檔案的目錄中，此目錄的位置會因作業系統而異 (請參閱[附錄 A](#))。

.war 檔案包括部署描述元，而該描述元會包含 Web 伺服器/應用程式伺服器載入和執行 Servlet 所需之基本配置資訊。因 Web 伺服器/應用程式伺服器的不同，您也可能必須指定 Servlet URL 的環境根目錄部分。

部署為 Web 歸檔檔案

如需在 Sun Java System Web Server 上進行部署的相關資訊，請參閱第 297 頁的「[在 Sun Java System Web Server 上部署 HTTP 通道 Servlet](#)」。

如需在 Sun Java System Application Server 上進行部署的相關資訊，請參閱第 299 頁的「[在 Sun Java System Application Server 上部署 HTTP 通道 Servlet](#)」。

在 Sun Java System Web Server 上部署 HTTP 通道 Servlet

以下說明是在 Sun Java System Web Server 上的部署。您可以透過使用 Web 瀏覽器存取 Servlet URL，來驗證 HTTP 通道 Servlet 是否已部署成功。它應顯示狀態資訊。

▼ 將 HTTP 通道 Servlet 部署為 .war 檔案

- 1 在瀏覽器型的管理 GUI 中，選擇 [虛擬伺服器類別] 標籤，然後選擇 [管理類別]。
- 2 選擇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 選擇 [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 選擇 [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] 欄位選擇適當的值，讓其指向 imqhttp.war 檔案，您可以在某個目錄 (目錄位置因作業系統而異) 中找到此檔案 (請參閱附錄 A)。
- 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* 部分：

```
http://hostName:portNumber/contextRoot/tunnel
```

舉例來說，如果您將 *contextRoot* 設定至 imq，則 [應用程式 URI] 欄位應為：

```
/imq
```

- 9 輸入應部署 Servlet 的安裝目錄路徑 (通常位於 Sun Java System Web Server 安裝根目錄下)。
 - 10 按一下 [確定]。
 - 11 重新啓動 Web 伺服器實例。
- 此時，即可從以下位址存取 Servlet：

```
http://hostName:portNumber/  
contextRoot/tunnel
```

現在，用戶端就可以使用此 URL 連線至使用 HTTP 連線的訊息服務。

停用伺服器存取記錄

您不必停用伺服器存取記錄，但是如果您停用記錄，將會獲得更好的效能。

▼ 停用伺服器存取記錄

- 1 選擇 [狀態] 標籤。
- 2 選擇 [記錄喜好設定頁面]。
使用記錄用戶端存取控制來停用記錄。

在 Sun Java System Application Server 上部署 HTTP 通道 Servlet

本節描述如何在 Sun Java System Application Server 上將 HTTP 通道 Servlet 部署為 .war 檔案，接著再配置通道 Servlet，以便接受來自 Message Queue 代理程式的連線。

您需要執行下列兩個步驟：

- 使用應用程式伺服器部署工具來部署 HTTP 通道 Servlet。
- 修改應用程式伺服器實例的 server.policy 檔案。

使用部署工具

▼ 在應用程式伺服器環境中部署 HTTP 通道 Servlet

- 1 在網路型的管理 GUI 中，選擇
[應用程式伺服器] > [實例] > [server1] > [應用程式] > [Web 應用程式]
- 2 按一下 [部署] 按鈕。
- 3 在 [檔案路徑:] 文字欄位輸入 HTTP 通道 Servlet .war 檔案 (imqhttp.war) 的位置，再按一下 [確定]。
imqhttp.war 檔案的位置會因您的作業系統而異 (請參閱附錄 A)。
- 4 設定 [環境根目錄] 文字欄位的值，再按一下 [確定]。

[環境根目錄] 欄位值為通道 Servlet URL 的 /contextRoot 部分：

```
http://hostName:portNumber/contextRoot/tunnel
```

例如，您可以將 [環境根目錄] 欄位設定為 /imq。

此時會出現一個確認畫面，確認通道 Servlet 已成功部署並且依預設已啓用，以及該通道 Servlet 的位置。在本例中其位置如下：

```
/var/opt/SUNWappserver8/domains/domain1/server1/applications/  
j2ee-modules/imqhttp_1
```

此時，即可從以下 URL 存取 Servlet：

```
http://hostName:portNumber/  
contextRoot/tunnel
```

現在，用戶端就可以使用此 URL 連線至使用 HTTP 連線的訊息服務。

修改 `server.policy` 檔案

應用程式伺服器會實施一組預設的安全策略，除非將其修改，否則這組安全策略會阻止 HTTP 通道 Servlet 接受來自 Message Queue 代理程式的連線。

每個應用程式伺服器實例，皆有一個包含其安全策略或規則的檔案。例如在 Solaris 上，`server1` 實例的此檔案位置是：

```
/var/opt/SUNWappserver8/domains/domain1/server1/config/
server.policy
```

若要配置通道 Servlet 接受來自 Message Queue 代理程式的連線，則必須在此檔案中新增一個項目。

▼ 修改應用程式伺服器的 `server.policy` 檔案

- 1 開啓 `server.policy` 檔案。

- 2 增加以下項目：

```
grant codeBase
"file:/var/opt/SUNWappserver8/domains/domain1/server1/
  applications/j2ee-modules/imqhttp_1/-"
{
    permission java.net.SocketPermission "*",
        "connect,accept,resolve";
};
```

步驟 2. 配置 `httpjms` 連線服務

預設情況下，不會針對代理程式啓動 HTTP 支援，所以您必須重新配置代理程式，以啓動 `httpjms` 連線服務。重新配置後，便可如第 64 頁的「啓動代理程式」中所述啓動代理程式。

▼ 啓動 `httpjms` 連線服務

- 1 開啓代理程式的實例配置檔案。

實例配置檔案的儲存目錄名稱，與此配置檔案相關聯之代理程式實例名稱 (`instanceName`) 相同 (請參閱附錄 A)：

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

- 2 將 `httpjms` 值增加至 `imq.service.activelist` 特性：

```
imq.service.activelist=jms,admin,httpjms
```

啓動時，代理程式會尋找在其主機電腦上執行的 Web 伺服器/應用程式伺服器和 HTTP 通道 Servlet。但是，若要存取遠端通道 Servlet，您可以重新配置 `servletHost` 和 `servletPort` 連線服務特性。

您還可以重新配置 `pullPeriod` 特性，以提昇效能。[第 300 頁的「步驟 2. 配置 httpjms 連線服務」](#) 中詳細介紹了 `httpjms` 連線服務配置特性。

特性	說明
<code>imq.httpjms.http.servletHost</code>	如有必要，請變更此值以指定執行 HTTP 通道 Servlet 的主機之名稱 (主機名稱或 IP 位址)。(可以是遠端主機，或本地主機上的特定主機名稱。)預設值： <code>localhost</code> 。
<code>imq.httpjms.http.servletPort</code>	變更此值，以指定代理程式用於存取 HTTP 通道 Servlet 的連接埠號。(如果 Web 伺服器上的預設連接埠已變更，則您必須相應地變更此特性。)預設值： <code>7675</code> 。
<code>imq.httpjms.http.pullPeriod</code>	指定用戶端執行階段發出從代理程式取出訊息之 HTTP 請求的時間間隔 (以秒為單位)。(請注意，此特性於代理程式上進行設定，並傳播至用戶端執行階段。)如果值為零或負數，則用戶端將始終擱置一個 HTTP 請求，以便隨時做好準備從而儘快取出訊息。若用戶端數目過大，可能會導致耗用大量的 Web 伺服器/應用程式伺服器資源，造成伺服器無法回應。在此類情況下，您應將 <code>pullPeriod</code> 特性設定為正的秒數。此特性設定用戶端的 HTTP 傳輸驅動程式在發出後續取出請求之前，所需等待的時間。將此值設定為正數可節省 Web 伺服器/應用程式伺服器資源，但會影響用戶端所監視的回應次數。預設值： <code>-1</code> 。
<code>imq.httpjms.http.connectionTimeout</code>	指定丟出異常之前，用戶端執行階段等待來自 HTTP 通道 Servlet 之回應的時間 (以秒為單位)。(請注意，此特性於代理程式上進行設定，並傳播至用戶端執行階段。)此特性也會指定代理程式與 HTTP 通道 Servlet 進行通訊後，需等待多長時間後才釋放連線。因為在此情況下，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，因此必須設定逾時時間。預設值： <code>60</code> 。

步驟 3. 配置 HTTP 連線

用戶端應用程式必須使用已適當配置的連線工廠受管理物件，以建立與代理程式的 HTTP 連線。本節說明 HTTP 連線配置問題。

配置連線工廠

若要啓用 HTTP 支援，您必須將連線工廠的 `imqAddressList` 屬性設定爲 HTTP 通道 Servlet URL。HTTP 通道 Servlet URL 的一般語法如下：

```
http://hostName:portNumber

/contextRoot/tunnel
```

其中 `hostName:portNumber` 爲託管 HTTP 通道 Servlet 之 Web 伺服器/應用程式伺服器的名稱和連接埠；`contextRoot` 爲在 Web 伺服器/應用程式伺服器上部署通道的路徑設定。

如需連線工廠屬性的一般資訊與 `imqAddressList` 屬性的特定資訊，請參閱「Message Queue Developer's Guide for Java Clients」。

您可以透過以下方法之一設定連線工廠屬性：

- 使用可建立連線工廠受管理物件 (請參閱第 150 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。
- 使用可啓動用戶端之指令的 `-D` 選項 (請參閱「Message Queue Developer's Guide for Java Clients」)。
- 當您透過程式設計由用戶端程式碼建立連線工廠後，請使用 API 呼叫以設定屬性 (請參閱「Message Queue Developer's Guide for Java Clients」)。

使用單一 Servlet 存取多重代理程式

若要執行多個代理程式，您無需配置多個 Web 伺服器/應用程式伺服器和 Servlet 實例。您可以讓同時執行的代理程式，共用單一 Web 伺服器/應用程式伺服器和 HTTP 通道 Servlet 實例。如果多個代理程式實例共用一個通道 Servlet，您必須配置 `imqAddressList` 連線工廠屬性，配置內容如下：

```
http://hostName:portNumber

/contextRoot/tunnel?ServerName=
bkrHostName:instanceName
```

其中 `bkrHostName` 爲代理程式實例主機名稱；`instanceName` 爲用戶端所要存取的特定代理程式實例名稱。

若要檢查您爲 `bkrHostName` 和 `instanceName` 輸入的字串是否正確，請從瀏覽器存取 Servlet URL，並產生 HTTP 通道 Servlet 的狀態報告。此報告將列出 Servlet 所存取的所有代理程式：

```
HTTP tunnel servlet ready.
Servlet Start Time : Thu May 30 01:08:18 PDT 2005
Accepting TCP connections from brokers on port : 7675
```

```
Total available brokers = 2
Broker List :
  jpgserv:broker2
  cochin:broker1
```

使用 HTTP 代理伺服器

如果您要使用 HTTP 代理伺服器存取 HTTP 通道 Servlet：

- 將 `http.proxyHost` 系統特性設定為代理伺服器主機名稱。
- 將 `http.proxyPort` 系統特性設定為代理伺服器連接埠號。

您可以使用可啓動用戶端應用程式之指令的 `-D` 選項來設定這些特性。

啓用 HTTPS 支援

以下各節描述啓用 HTTPS 支援所需的步驟。這些步驟與第 297 頁的「啓用 HTTP 支援」中的步驟相似，但是增加了產生和存取 SSL 憑證所需的步驟。

▼ 啓用 HTTPS 支援

- 1 為 HTTPS 通道 Servlet 產生自我簽署的憑證
- 2 修改 HTTP 通道 Servlet `.war` 檔案的部署描述元，使其：
 - 指向您放置憑證金鑰庫的位置
 - 指定憑證金鑰庫密碼
- 3 部署 HTTP 通道 Servlet。您可以在下列伺服器上部署 HTTP 通道 Servlet：
 - Sun Java System Web Server
 - Sun Java System Application Server
- 4 配置代理程式的 `httpsjms` 連線服務，並啓動代理程式。
- 5 配置 HTTPS 連線。

以下各節中將更加詳細地說明這些步驟中的每一步。

步驟 1. 為 HTTPS 通道 Servlet 產生自我簽署的憑證

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，僅使用自我簽署的伺服器憑證即可實施 SSL。在 `httpsjms` 連線服務架構中，HTTPS 通道 Servlet 所扮演的角色，就是代理程式與應用程式用戶端的伺服器。

執行 `keytool` 公用程式，以便為通道 Servlet 產生自我簽署的憑證。在指令提示符號下輸入以下內容：

```
JRE_HOME/bin/keytool -servlet keyStoreLocation
```

此公用程式將提示您提供所需的資訊。(在 Unix 系統上，需要以超級使用者 (root) 的身份執行 `keytool`，才擁有建立金鑰庫的權限。)

首先，`keytool` 會提示您提供金鑰庫密碼，以及某些組織資訊，然後提示您確認。接收到確認之後，它會在產生金鑰對時暫停。然後它會要求您提供密碼，以鎖定特定的金鑰對 (金鑰密碼)；請按 `Return` 鍵以回應此提示：這會使金鑰密碼與金鑰庫密碼相同。

備註 - 請記住您提供的密碼：稍後您必須將此密碼提供給通道 Servlet，它才可開啓金鑰庫。

JDK `keytool` 公用程式會產生自我簽署的憑證，並將它放置在 Message Queue 的金鑰庫檔案中，該檔案位於 `keyStoreLocation` 引數所指定的位置。

備註 - HTTPS 通道 Servlet 必須能夠查看此金鑰庫。請確定將位於 `keyStoreLocation` 中所產生的金鑰庫移動/複製到 HTTPS 通道 Servlet 可存取的位置 (請參閱第 305 頁的「[步驟 3. 部署 HTTPS 通道 Servlet](#)」)。

步驟 2. 修改 HTTP 通道 Servlet .war 檔案的描述元檔案

HTTP 通道 Servlet 的 `.war` 檔案包括部署描述元，其中包含 Web 伺服器/應用程式伺服器載入和執行 Servlet 所需的基本配置資訊。

`imqhttps.war` 檔案的部署描述元無法知道您已將通道 Servlet 所需的金鑰庫檔案置於何處。所以您需要編輯通道 Servlet 的部署描述元 (XML 檔案)，以便在部署 `imqhttps.war` 檔案之前指定金鑰庫的位置和密碼。

▼ 修改 HTTPS 通道 Servlet .war 檔案

- 1 將 .war 檔案複製到暫存目錄。

```
cp /usr/share/lib/imq/imqhttps.war /tmp (Solaris)
cp /opt/sun/mq/share/lib/imqhttps.war /tmp (Linux)
cp IMQ_HOME/lib/imqhttps.war /tmp (Windows)
```

- 2 將此暫存目錄變為您的目前目錄。

```
$ cd /tmp
```

- 3 擷取 .war 檔案的內容。

```
$ jar xvf imqhttps.war
```

- 4 列出 .war 檔案的部署描述元。

```
$ ls -l WEB-INF/web.xml
```

- 5 編輯 web.xml 檔案，為 keystoreLocation 和 keystorePassword 引數 (如有必要，還包括 servletPort 和 servletHost 引數) 提供正確的值。

- 6 重新組合 .war 檔案的內容。

```
$ jar uvf imqhttps.war WEB-INF/web.xml
```

此時，您就可以使用已修改的 imqhttps.war 檔案來部署 HTTPS 通道 Servlet。(如果您擔心金鑰庫密碼洩漏，則可以使用檔案系統權限來限制其他人對 imqhttps.war 檔案的存取。)

步驟 3. 部署 HTTPS 通道 Servlet

您可以將 HTTP 通道 Servlet 部署為 Sun Java System Web Server 或 Sun Java System Application Server 上的 Web 歸檔 (WAR) 檔案。

將 HTTPS 通道 Servlet 部署為 .war 檔案，需要使用 Web 伺服器/應用程式伺服器所提供的部署機制。HTTPS 通道 Servlet .war 檔案 (imqhttps.war) 的所在目錄位置依作業系統而異 (請參閱附錄 A)。

您應確保已為 Web 伺服器啓動加密，從而讓用戶端與代理程式之間進行端對端安全通訊。

部署為 Web 歸檔檔案

如需在 Sun Java System Web Server 上進行部署的相關資訊，請參閱第 306 頁的「在 Sun Java System Web Server 上部署 HTTPS 通道 Servlet」。

如需在 Sun Java System Application Server 上進行部署的相關資訊，請參閱第 307 頁的「在 Sun Java System Application Server 上部署 HTTPS 通道 Servlet」。

在 Sun Java System Web Server 上部署 HTTPS 通道 Servlet

本節描述如何將 HTTPS 通道 Servlet 部署為 Sun Java System Web Server 上的 .war 檔案。您可以透過使用 Web 瀏覽器存取 Servlet URL，來驗證 HTTPS 通道 Servlet 是否已部署成功。它應顯示狀態資訊。

在部署 HTTPS 通道 Servlet 之前，請確保 JSSE .jar 檔案包含在 Web 伺服器的類別路徑中。執行此作業的最簡單方法是將 jsse.jar、jnet.jar 和 jcert.jar 檔案複製到 WebServer_TOPDIR/bin/https/jre/lib/ext。

▼ 將 HTTPS 通道 Servlet 部署為 .war 檔案

- 1 在瀏覽器型的管理 GUI 中，選擇 [虛擬伺服器類別] 標籤。按一下 [管理類別]。
- 2 選擇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 選擇 [管理虛擬伺服器]。
- 4 選擇適當的虛擬伺服器名稱，然後按一下 [管理] 按鈕。
- 5 選擇 [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] 欄位選擇適當的值，讓其指向已修改的 imqhttps.war 檔案 (請參閱第 304 頁的「步驟 2. 修改 HTTP 通道 Servlet.war 檔案的描述元檔案」)。
- 8 在 [應用程式 URI] 欄位中輸入路徑。
[應用程式 URI] 欄位值為通道 Servlet URL 的 /contextRoot 部分：
`https://hostName:portNumber/contextRoot/tunnel`
舉例來說，如果您將 contextRoot 設定為 imq，則 [應用程式 URI] 欄位應為：
`/imq`
- 9 輸入應部署 Servlet 的安裝目錄路徑 (通常位於 Sun Java System Web Server 安裝根目錄下)。
- 10 按一下 [確定]。

11 重新啓動 Web 伺服器實例。

此時，即可從以下 URL 存取 Servlet：

`https://hostName:portNumber/imq/tunnel`

現在，用戶端就可以使用此 URL 連線至使用安全 HTTPS 連線的訊息服務。

停用伺服器存取記錄

您不必停用伺服器存取記錄，但是如果您停用記錄，將會獲得更好的效能。

▼ 停用伺服器存取記錄

- 1 選擇 [狀態] 標籤。
- 2 選擇 [記錄喜好設定頁面]。
使用記錄用戶端存取控制來停用記錄。

在 Sun Java System Application Server 上部署 HTTPS 通道 Servlet

本節描述如何將 HTTPS 通道 Servlet 部署為 Sun Java System Application Server 上的 .war 檔案。

您需要執行下列兩個步驟：

- 使用應用程式伺服器部署工具來部署 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

以下程序顯示如何在應用程式伺服器環境中部署 HTTPS 通道 Servlet。

▼ 在應用程式伺服器環境中部署 HTTPS 通道 Servlet

- 1 在網路型的管理 GUI 中，選擇
[應用程式伺服器] > [實例] > [server1] > [應用程式] > [Web 應用程式]
- 2 按一下 [部署] 按鈕。
- 3 在 [檔案路徑:] 文字欄位輸入 HTTPS 通道 Servlet .war 檔案 (imqhttps.war) 的位置，再按一下 [確定]。
imqhttps.war 檔案的位置會因您的作業系統而異 (請參閱附錄 A)。
- 4 設定 [環境根目錄] 文字欄位的值，再按一下 [確定]。
[環境根目錄] 欄位值為通道 Servlet URL 的 `/contextRoot` 部分：

```
https://hostName:portNumber/contextRoot/tunnel
```

例如，您可以將 [環境根目錄] 欄位設定爲：

```
/imq
```

接下來的畫面會顯示，通道 Servlet 已成功部署並且依預設已啓用，以及該通道 Servlet 的位置。在本例中其位置如下：

```
/var/opt/SUNWappserver8/domains/domain1/server1/applications/  
j2ee-modules/imqhttps_1
```

此時，即可從以下 URL 存取 Servlet：

```
https://hostName:portNumber/  
contextRoot/tunnel
```

現在，用戶端就可以使用此 URL 連線至使用 HTTPS 連線的訊息服務。

修改 server.policy 檔案

應用程式伺服器會實施一組預設的安全策略，除非將其修改，否則這組安全策略會阻止 HTTPS 通道 Servlet 接受來自 Message Queue 代理程式的連線。

每個應用程式伺服器實例，皆有一個包含安全策略或規則的檔案。例如在 Solaris 上，server1 實例的此檔案位置是：

```
/var/opt/SUNWappserver8/domains/domain1/server1/config/  
server.policy
```

若要讓通道 Servlet 接受來自 Message Queue 代理程式的連線，則必須在此檔案中新增一個項目。

▼ 修改應用程式伺服器的 server.policy 檔案

1 開啓伺服器策略檔案。

2 增加以下項目：

```
grant codeBase  
"file:/var/opt/SUNWappserver8/domains/domain1/server1/  
applications/j2ee-modules/imqhttps_1/-"  
{  
    permission java.net.SocketPermission "*",  
        "connect,accept,resolve";  
};
```

步驟 4. 配置 httpsjms 連線服務

預設情況下，不會針對代理程式啓動 HTTPS 支援，所以您必須重新配置代理程式，以啓動 httpsjms 連線服務。重新配置後，便可如第 64 頁的「啓動代理程式」中所述啓動代理程式。

▼ 啓動 httpsjms 連線服務

1 開啓代理程式的實例配置檔案。

實例配置檔案的儲存目錄名稱，與此配置檔案相關聯之代理程式實例名稱 (*instanceName*) 相同 (請參閱附錄 A)：

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

2 將 httpsjms 值增加至 imq.service.activelist 特性：

```
imq.service.activelist=jms,admin,httpsjms
```

啓動時，代理程式會尋找在其主機電腦上執行的 Web 伺服器 and HTTPS 通道 Servlet。但是，若要存取遠端通道 Servlet，您可以重新配置 `servletHost` 和 `servletPort` 連線服務特性。

您還可以重新配置 `pullPeriod` 特性，以提昇效能。第 309 頁的「步驟 4. 配置 httpsjms 連線服務」中詳細介紹了 httpsjms 連線服務配置特性。

特性	說明
<code>imq.httpsjms.https.servletHost</code>	如有必要，請變更此值以指定執行 HTTPS 通道 Servlet 的主機之名稱 (主機名稱或 IP 位址)。(可以是遠端主機，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 伺服器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> 。

特性	說明
<code>imq.httpsjms.https.pullPeriod</code>	指定每個用戶端發出從代理程式取出訊息之 HTTP 請求的時間間隔 (以秒為單位)。(請注意，此特性於代理程式上進行設定，並傳播至用戶端執行階段。)如果值為零或負數，則用戶端將始終擱置一個 HTTP 請求，以便隨時做好準備從而儘快取出訊息。若用戶端數目過大，可能會導致耗用大量的 Web 伺服器/應用程式伺服器資源，造成伺服器無法回應。在此類情況下，您應將 <code>pullPeriod</code> 特性設定為正的秒數。此特性設定用戶端的 HTTP 傳輸驅動程式在發出後續取出請求之前，所需等待的時間。將此值設定為正數可節省 Web 伺服器資源，但會影響用戶端所監視的回應次數。預設值：-1。
<code>imq.httpsjms.https.connectionTimeout</code>	指定丟出異常之前，用戶端執行階段等待來自 HTTPS 通道 Servlet 之回應的時間 (以秒為單位)。(請注意，此特性於代理程式上進行設定，並傳播至用戶端執行階段。)此特性也會指定代理程式與 HTTPS 通道 Servlet 進行通訊後，需等待多長時間後才釋放連線。因為在此情況下，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，因此必須設定逾時時間。預設值：60。

步驟 5. 配置 HTTPS 連線

用戶端應用程式必須使用已適當配置的連線工廠受管理物件，以建立與代理程式的 HTTPS 連線。

然而，用戶端必須能夠存取 Java Secure Socket Extension (JSSE) 所提供的 SSL 程式庫，而且必須具有根憑證。SSL 程式庫隨附於 JDK 1.4。如果您使用較舊的 JDK 版本，請參閱第 310 頁的「配置 JSSE」，否則請繼續執行第 311 頁的「匯入超級使用者憑證」。

解決這些問題後，就可以繼續配置 HTTPS 連線。

配置 JSSE

▼ 配置 JSSE

- 1 將 JSSE .jar 檔案複製到 `JRE_HOME/lib/ext` 目錄。
`jse.jar, jnet.jar, jcert.jar`
- 2 以靜態方式增加 JSSE 安全性提供者：將
`security.provider.n=com.sun.net.ssl.internal.ssl.Provider`

增加至 `JRE_HOME/lib/security/java.security` 檔案 (其中 n 為安全性提供者套裝軟體的下一個可用優先權號碼)。

- 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 (簽署 Web 伺服器憑證的單位) 的根憑證依預設不在信任資料庫中，或者您使用的是專用 Web 伺服器/應用程式伺服器憑證，則必須將此憑證增加至信任資料庫。如果是此情況，請遵循以下說明，否則請移至第 311 頁的「配置連線工廠」。

假設此憑證儲存在 `certFile` 中，且 `trustStoreFile` 為您的金鑰庫，請執行以下指令：

```
JRE_HOME/bin/keytool -import -trustcacerts
  -alias aliasForCertificate -file certFile

-keystore trustStoreFile
```

對以下問題回答 YES：Trust this certificate?

您還需要使用可啓動用戶端應用程式之指令的 `-D` 選項，來指定以下 JSSE 特性：

```
javax.net.ssl.trustStore=trustStoreFile
javax.net.ssl.trustStorePassword=trustStorePasswd
```

配置連線工廠

若要啓用 HTTPS 支援，您必須將連線工廠的 `imqAddressList` 屬性設定為 HTTPS 通道 Servlet URL。HTTPS 通道 Servlet URL 的一般語法如下：

```
https://hostName:portNumber
/contextRoot/tunnel
```

其中 `hostName:portNumber` 為託管 HTTPS 通道 Servlet 之 Web 伺服器名稱和連接埠；`contextRoot` 為在 Web 伺服器上部署通道 Servlet 時的路徑設定。

如需連線工廠屬性的一般資訊與 `imqAddressList` 屬性的特定資訊，請參閱「Message Queue Developer's Guide for Java Clients」。

您可以透過以下方法之一設定連線工廠屬性：

- 使用可建立連線工廠受管理物件 (請參閱第 150 頁的「新增連線工廠」) 之 `imqobjmgr` 指令的 `-o` 選項，或在使用管理主控台 (`imqadmin`) 建立連線工廠受管理物件時設定屬性。

- 使用可啓動用戶端應用程式之指令的 `-D` 選項 (請參閱「Message Queue Developer's Guide for Java Clients」)。
- 當您透過程式設計由用戶端應用程式碼建立連線工廠後，請使用 API 呼叫以設定屬性 (請參閱「Message Queue Developer's Guide for Java Clients」)。

使用單一 Servlet 存取多重代理程式

若要執行多個代理程式，您無需配置多個 Web 伺服器和 Servlet 實例。您可以讓同時執行的代理程式，共用單一 Web 伺服器和 HTTPS 通道 Servlet 實例。如果多個代理程式實例共用一個通道 Servlet，您必須配置 `imqAddressList` 連線工廠屬性，配置內容如下：

```
https://hostName:portNumber
```

```
/contextRoot/tunnel?ServerName=  
bkrHostName:instanceName
```

其中 `bkrHostName` 為代理程式實例主機名稱；`instanceName` 為用戶端所要存取的特定代理程式實例名稱。

若要檢查您為 `bkrHostName` 和 `instanceName` 輸入的字串是否正確，請從瀏覽器存取 Servlet URL，並產生 HTTPS 通道 Servlet 的狀態報告。此報告將列出 Servlet 所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 代理伺服器存取 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。

您可以使用可啓動用戶端應用程式之指令的 `-D` 選項來設定這些特性。

疑難排解

本節描述 HTTP 或 HTTPS 連線可能發生的問題，並提供處理這些問題的方法。

伺服器或代理程式發生故障

如果 Web 伺服器發生故障並重新啓動，則所有連線均會復原且不會影響用戶端。但是，如果代理程式發生故障並重新啓動，則系統會丟出異常，且用戶端必須重新建立它們的連線。

如果 Web 伺服器與代理程式均發生故障，且未重新啓動代理程式，則 Web 伺服器會復原用戶端連線並繼續等待代理程式連線，而不通知用戶端。若要避免這種情況，請確定代理程式已重新啓動。

用戶端無法透過通道 Servlet 進行連線

如果 HTTPS 用戶端無法透過通道 Servlet 連線至代理程式，請執行以下作業：

▼ 如果用戶端無法連線

- 1 啓動 Servlet 和代理程式。
- 2 使用瀏覽器，透過 HTTPS 通道 Servlet URL 以手動方式存取 Servlet。
- 3 使用以下管理指令暫停並重新繼續連線：

```
imqcmd pause svc -n httpsjms -u admin  
imqcmd resume svc -n httpsjms -u admin
```

當服務重新繼續時，HTTPS 用戶端應該就能透過通道 Servlet 連線至代理程式。

常用指令公用程式指令

本附錄列出常用之 Message Queue™ 指令公用程式 (imqcmd) 指令。如需指令行可用之指令選項和屬性的詳細清單，請參閱第 222 頁的「指令公用程式」中的第 222 頁的「指令公用程式」。

語法

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

-H 或 -h 會提供詳細的說明。-v 子命令提供版本資訊。

當您使用 imqcmd 時，指令公用程式會提示您輸入密碼。若要避免出現提示 (同時增加安全性)，您可以使用 -passfile *pathToPassfile* 選項，將公用程式指向包含管理員使用者名稱和密碼的密碼檔案。

範例：imqcmd query bkr -u *adminUserName* -passfile *pathToPassfile* -b *myServer:7676*

代理程式和叢集管理

```
imqcmd query bkr  
imqcmd pause bkr  
imqcmd restart bkr  
imqcmd resume bkr  
imqcmd shutdown bkr -b myBroker:7676  
imqcmd update bkr -o "imq.system.max_count=1000"  
imqcmd reload cls
```

代理程式配置特性 (-o 選項)

第 316 頁的「代理程式配置特性 (-o 選項)」中列出常用的代理程式配置特性。如需代理程式配置特性及其說明的完整清單，請參閱第 14 章。

表 D-1 代理程式配置特性 (-o 選項)

特性	備註
imq.autocreate.queue	
imq.autocreate.queue.maxNumActiveConsumers	指定 -1 表示無限制
imq.autocreate.queue.maxNumBackupConsumers	指定 -1 表示無限制
imq.autocreate.topic	
imq.cluster.url	
imq.destination.DMQ.truncateBody	
imq.destination.logDeadMessages	
imq.log.file.rolloverbytes	指定 -1 表示無限制
imq.log.file.rolloversecs	指定 -1 表示無限制
imq.log.level	NONEERRORWARNINGINFO
imq.message.max_size	指定 -1 表示無限制
imq.portmapper.port	
imq.system.max_count	指定 -1 表示無限制
imq.system.max_size	指定 -1 表示無限制

服務和連線管理

```
imqcmd list svc
imqcmd query svc
imqcmd update svc -n jms -o "minThreads=200" -o "maxThreads=400" -o "port=8995"
imqcmd pause svc -n jms
imqcmd resume svc -n jms
imqcmd list cxn -svn jms
imqcmd query cxn -n 1234567890
```

長期訂閱者管理

```
imqcmd list dur -d MyTopic
imqcmd destroy dur -n myDurSub -c "clientID-111.222.333.444"
imqcmd purge dur -n myDurSub -c "clientID-111.222.333.444"
```

作業事件管理

```
imqcmd list txn
imqcmd commit txn -n 1234567890
imqcmd query txn -n 1234567890
imqcmd rollback txn -n 1234567890
```

目標管理

```
imqcmd create dst -n MyQueue -t q -o "maxNumMsgs=1000" -o "maxNumProducers=5"
imqcmd update dst -n MyTopic -t t -o "limitBehavior=FLOW_CONTROL|REMOVE_OLDEST|REJECT_NEWEST|REMOVE_LOW_PRIORITY"
imqcmd compact dst -n MyQueue -t q
imqcmd purge dst -n MyQueue -t q
imqcmd pause dst -n MyQueue -t q -pst PRODUCERS|CONSUMERS|ALL
imqcmd resume dst -n MyQueue -t q
imqcmd destroy dst -n MyQueue -t q
imqcmd query dst -n MyQueue -t q
imqcmd list dst -tmp
```

目標配置特性 (-o 選項)

第 317 頁的「目標配置特性 (-o 選項)」中列出常用的目標配置特性。如需目標配置特性及其說明的完整清單，請參閱第 15 章。

表 D-2 目標配置特性 (-o 選項)

特性	備註
consumerFlowLimit	指定 -1 表示無限制
isLocalOnly (僅限建立時使用)	
limitBehavior	FLOW_CONTROLREMOVE_OLDESTREJECT_NEWESTREMOVE_LOW_PRIORITY
localDeliveryPreferred (僅限佇列)	
maxNumActiveConsumers (僅限佇列)	指定 -1 表示無限制

表 D-2 目標配置特性 (-o 選項) (續)

特性	備註
maxNumBackupConsumers (僅限佇列)	指定 -1 表示無限制
maxBytesPerMsg	指定 -1 表示無限制
maxNumMsgs	指定 -1 表示無限制
maxNumProducers	指定 -1 表示無限制
maxTotalMsgBytes	指定 -1 表示無限制
useDMQ	

度量

```
imqcmd metrics bkr -m cxn|rts|ttl -int 5 -msp 20
imqcmd metrics svc -m cxn|rts|ttl
imqcmd metrics dst -m con|dsk|rts|ttl
```

索引

A

acknowledgeMode 啓動規格屬性, 275
ActivationSpec JavaBean, 274
addressList 啓動規格屬性, 274
addressList 資源介面屬性, 272
addressList 管理的連線工廠屬性, 273
addressListBehavior 資源介面屬性, 272
addressListBehavior 管理的連線工廠屬性, 273
addressListIterations 資源介面屬性, 272
addressListIterations 管理的連線工廠屬性, 273
ADMIN 服務類型, 72
admin 使用者, 118, 122, 124
admin 連線服務, 72, 96
admin 群組, 120
anonymous 群組, 120
API 文件, 288, 289, 290
AUTOSTART 特性, 65

C

clientID 啓動規格屬性, 274
clientID 管理的連線工廠屬性, 273
cluster 連線服務, 130
config.properties 檔案, 83, 160, 161
connectionURL 資源介面屬性, 272
customAcknowledgeMode 啓動規格屬性, 275

D

default.properties 檔案, 82

destination 啓動規格屬性, 274
destinationType 啓動規格屬性, 274

E

endpointExceptionRedeliveryAttempts 啓動規格屬性, 275
/etc/hosts 檔案 (Linux), 159

G

guest 使用者, 118

H

hosts 檔案 (Linux), 159
HTTP
支援架構, 295-296
代理伺服器, 295
連線服務
 參閱httpjms 連線服務
傳輸驅動程式, 295
HTTP 通道 Servlet
部署, 297-300
關於, 295
HTTP 連線
支援, 295
多個代理程式, 302-303
通道 Servlet
 參閱HTTP 通道 Servlet

HTTP 連線 (續)

請求時間間隔, 301

httpjms 連線服務

配置, 300-301

設置, 297-303

關於, 72, 96

HTTPS

支援架構, 295-296

連線服務

參閱httpsjms 連線服務

HTTPS 通道 Servlet

部署, 305-308

關於, 295

HTTPS 連線

支援, 295

多個代理程式, 312

通道 Servlet

參閱HTTPS 通道 Servlet

請求時間間隔, 310

httpsjms 連線服務, 130

配置, 309-310

設置, 303-312

關於, 72, 96

I

imq.accesscontrol.enabled 特性, 77, 245, 253

imq.accesscontrol.file.filename 特性, 78, 245, 253

imq.audit.enabled 特性, 80, 140, 248, 253

imq.authentication.basic.user_repository 特性, 78, 246, 253

imq.authentication.client.response.timeout 特性, 79, 246, 253

imq.authentication.type 特性, 79, 246, 253

imq.autocreate.destination.isLocalOnly 特性, 240, 253

imq.autocreate.destination.limitBehavior 特性, 239, 253

imq.autocreate.destination.maxBytesPerMsg 特性, 239, 253

imq.autocreate.destination.maxCount 特性, 238, 253

imq.autocreate.destination.maxNumMsgs 特性, 238, 253

imq.autocreate.destination.maxNumProducers 特性, 240, 253

imq.autocreate.destination.maxTotalMsgBytes 特性, 239, 240, 253

imq.autocreate.destination.useDMQ 特性, 114

imq.autocreate.queue.consumerFlowLimit 特性, 240, 253

imq.autocreate.queue.localDeliveryPreferred 特性, 240, 253

imq.autocreate.queue.maxNumActiveConsumers 特性, 93, 240, 253

imq.autocreate.queue.maxNumBackupConsumers 特性, 93, 240, 253

imq.autocreate.queue 特性, 93, 238, 253

imq.autocreate.topic 特性, 93, 238, 253

imq.cluster.brokerlist 特性, 157, 159, 160, 161, 252, 253

imq.cluster.masterbroker 特性, 157, 160, 161, 252, 253

imq.cluster.port 特性, 157, 252, 253

imq.cluster.transport 特性, 158, 159, 160, 252, 254

imq.cluster.url 特性, 93, 157, 158, 159, 160, 161, 252, 254

imq.destination.DMQ.truncateBody 特性, 74, 93, 238, 254

imq.destination.logDeadMsgs 特性, 81, 93, 248, 254

imq.hostname 特性, 73, 235, 254

imq.httpjms.http.servletHost 特性, 301

imq.imqcmd.password 特性, 79, 246, 254

imq.keystore.file.dirpath 特性, 132, 248, 254

imq.keystore.file.name 特性, 132, 254

imq.keystore.password 特性, 80, 132, 139, 254

imq.keystore.property_name 特性, 254

imq.log.console.output 特性, 81, 249, 254

imq.log.console.stream 特性, 81, 254

imq.log.file.dirpath 特性, 81, 249, 254

imq.log.file.filename 特性, 81, 249, 254

imq.log.file.output 特性, 81, 249, 254

imq.log.file.rolloverbytes 特性, 81, 93, 249, 254

imq.log.file.rolloversecs 特性, 81, 93, 249, 254

imq.log.level 特性, 81, 93, 248, 254

imq.log.syslog.facility 特性, 250, 254

imq.log.syslog.identity 特性, 250, 254

imq.log.syslog.logconsole 特性, 250, 254

imq.log.syslog.logpid 特性, 250, 254

imq.log.syslog.output 特性, 81, 250, 254

imq.log.timezone 特性, 251, 254

- imq.message.expiration.interval 特性, 74, 238, 254
 imq.message.max_size 特性, 74, 93, 238, 254
 imq.metrics.enabled 特性, 80, 251, 254
 imq.metrics.interval 特性, 80, 251, 254
 imq.metrics.topic.enabled 特性, 82, 251, 255
 imq.metrics.topic.interval 特性, 82, 251, 255
 imq.metrics.topic.persist 特性, 82, 251, 255
 imq.metrics.topic.timetolive 特性, 82, 251, 255
 imq.passfile.dirpath 特性, 79, 246, 255
 imq.passfile.enabled 特性, 79, 246, 255
 imq.passfile.name 特性, 79, 246, 255
 imq.persist.file.destination.message.filepool.limit 特性, 76, 242, 255
 imq.persist.file.message.cleanup 特性, 76, 242, 255
 imq.persist.file.message.filepool.cleanratio 特性, 76, 242, 255
 imq.persist.file.message.max_record_size 特性, 241, 255
 imq.persist.file.message.vrfile.max_record_size 特性, 76
 imq.persist.file.sync.enabled 特性, 76, 242, 255
 Sun Cluster 需求, 242, 255
 imq.persist.file.sync 特性, 85
 imq.persist.jdbc.brokerid 特性, 77, 243, 255
 imq.persist.jdbc.closedburl 特性, 77, 243, 255
 imq.persist.jdbc.createdburl 特性, 77, 243, 255
 imq.persist.jdbc.driver 特性, 77, 243, 255
 imq.persist.jdbc.needpassword 特性, 77, 243, 255
 imq.persist.jdbc.opendburl 特性, 77, 243, 255
 imq.persist.jdbc.password 特性, 77, 139, 243, 255
 imq.persist.jdbc.table.IMQCCREC35 特性, 77, 244, 255
 imq.persist.jdbc.table.IMQDEST35 特性, 77, 244, 255
 imq.persist.jdbc.table.IMQINT35 特性, 244, 255
 imq.persist.jdbc.table.IMQLIST35 特性, 244, 255
 imq.persist.jdbc.table.IMQMSG35 特性, 244, 255
 imq.persist.jdbc.table.IMQPROPS35 特性, 244, 255
 imq.persist.jdbc.table.IMQSV35 特性, 77, 244, 255
 imq.persist.jdbc.table.IMQTACK35 特性, 245, 255
 imq.persist.jdbc.table.IMQTXN35 特性, 244, 255
 imq.persist.jdbc.user 特性, 77, 243, 256
 imq.persist.store 特性, 75, 85, 241
 imq.ping.interval 特性, 73, 237, 256
 imq.portmapper.backlog 特性, 73, 236, 256
 imq.portmapper.hostname 特性, 73, 235, 256
 imq.portmapper.port 特性, 72, 93, 236, 256
 imq.protocol.protocolType.inbufsz, 189
 imq.protocol.protocolType.nodelay, 189
 imq.protocol.protocolType.outbufsz, 189
 imq.resource_state.count 特性, 238, 256
 imq.resource_state.threshold 特性, 238, 256
 imq.resourceState.count 特性, 75
 imq.service.activelist 特性, 72, 235, 256
 imq.service_name.accesscontrol.enabled 特性, 256
 imq.service_name.accesscontrol.file.filename 特性, 245, 256
 imq.service_name.authentication.type 特性, 246, 256
 imq.service_name.max_threads 特性, 236, 256
 imq.service_name.min_threads 特性, 236, 256
 imq.service_name.protocol_type.hostname 特性, 157, 236, 252, 253, 256
 imq.service_name.protocol_type.port 特性, 236, 256
 imq.service_name.threadpool_model 特性, 236, 256
 imq.serviceName.accesscontrol.enabled 特性, 77
 imq.serviceName.accesscontrol.file.filename 特性, 78
 imq.serviceName.authentication.type 特性, 79
 imq.serviceName.max_threads 特性, 73
 imq.serviceName.min_threads 特性, 73
 imq.serviceName.protocolType.hostname 特性, 73
 imq.serviceName.protocolType.port 特性, 72
 imq.serviceName.threadpool_model 特性, 73
 imq.shared.connectionMonitor_limit 特性, 73, 237, 256
 imq.system.max_count 特性, 74, 93, 237, 256
 imq.system.max_size 特性, 74, 93, 237, 256
 imq.transaction.autorollback 特性, 104, 238, 256
 imq.user_repository.ldap.base 特性, 247, 256
 imq.user_repository.ldap.gidattr 特性, 247, 256
 imq.user_repository.ldap.grpbase 特性, 247, 256
 imq.user_repository.ldap.grpfilter 特性, 247, 256
 imq.user_repository.ldap.grpsearch 特性, 247, 256
 imq.user_repository.ldap.memattr 特性, 247, 256
 imq.user_repository.ldap.password 特性, 79, 139, 247, 256
 imq.user_repository.ldap.principal 特性, 79, 247, 257
 imq.user_repository.ldap.property_name 特性, 247, 257
 imq.user_repository.ldap.server 特性, 79, 246, 257

- imq.user_repository.ldap.ssl.enabled 特性, 248, 257
- imq.user_repository.ldap.timeout 特性, 247, 257
- imq.user_repository.ldap.uidattr 特性, 247, 257
- imq.user_repository.ldap.usrfilter 特性, 247, 257
- imqAckTimeout 屬性, 267
- imqAddressList 屬性, 264
- imqAddressListBehavior 屬性, 264
- imqAddressListIterations 屬性, 264
- imqbrokerd.conf 檔案, 65, 68
- imqbrokerd 指令, 64
 - 在密碼檔案中, 138
 - 連線代理程式, 159
 - 配置檔案 (Solaris、Linux), 65, 68
 - 參照, 218
 - 設定記錄特性, 166
 - 從叢集中移除代理程式, 161
 - 移除代理程式, 68
 - 將代理程式增加至叢集, 160
 - 清除資料存放區, 85, 111
 - 備份配置變更記錄, 162
 - 復原配置變更記錄, 162
 - 傳送引數至, 84
 - 選項, 218-222
 - 關於, 35
- imqcmd 指令
 - 一般選項, 228, 233
 - 在密碼檔案中, 138
 - 作業事件管理, 103
 - 長期訂閱子指令, 101
 - 相依於主代理程式, 162
 - 度量監視, 169-173
 - 參照, 222
 - 與代理程式的安全連線, 134-135, 228
 - 實體目標子指令 (表格), 106
 - 實體目標管理, 105
 - 關於, 35
- imqConfiguredClientID 屬性, 266
- imqConnectionFlowCount 屬性, 267
- imqConnectionFlowLimit 屬性, 267
- imqConnectionFlowLimitEnabled 屬性, 267
- imqConsumerFlowLimit 屬性, 267
- imqConsumerFlowThreshold 屬性, 267
- imqdbmgr 指令
 - 在密碼檔案中, 138
- imqdbmgr 指令 (續)
 - 參照, 231
 - 選項, 231-232
 - 關於, 35
- imqDefaultPassword 屬性, 266
- imqDefaultUsername 屬性, 266
- imqDestinationDescription 屬性, 269
- imqDestinationName 屬性, 269
- imqDisableSetClientID 屬性, 266
- imqFlowControlLimit 屬性, 267
- imqJMSDeliveryMode 屬性, 269
- imqJMSExpiration 屬性, 269
- imqJMSPriority 屬性, 149, 269
- imqkeytool 指令
 - 使用, 131
 - 指令語法, 131, 304
 - 參照, 234
 - 關於, 35
- imqLoadMaxToServerSession 屬性, 148, 268
- imqobjmgr 指令
 - 子指令, 229
 - 參照, 229
 - 選項, 229
 - 關於, 35
- imqOverrideJMSDeliveryMode 屬性, 269
- imqOverrideJMSExpiration 屬性, 269
- imqOverrideMSHeadersToTemporaryDestinations 屬性, 149, 269
- imqOverrideJMSPriority 屬性, 149, 269
- imqQueueBrowserMax MessagesPerRetrieve 屬性, 148, 268
- imqQueueBrowserRetrieveTimeout 屬性, 148, 268
- imqReconnectAttempts 屬性, 264
- imqReconnectEnabled 屬性, 264
- imqReconnectInterval 屬性, 264
- imqSetJMSXAppID 屬性, 268
- imqSetJMSXConsumerTXID 屬性, 268
- imqSetJMSXProducerTXID 屬性, 268
- imqSetJMSXRcvTimestamp 屬性, 268
- imqSetJMSXUserID 屬性, 268
- imqSSLIsHostTrusted 屬性, 264
- imqsvcadm 指令
 - 子指令, 233
 - 參照, 233

imqsvcadm 指令 (續)

選項, 233

關於, 35

imqusermgr 指令

子指令, 232

用於, 119

使用者名稱, 121

參照, 232

密碼, 121

選項, 232

關於, 35

install.properties 檔案, 82

J

J2EE 連接器架構 (JCA), 271

java.naming.factory.initial 屬性, 142, 143

java.naming.provider.url 屬性, 142, 143

java.naming.security.authentication 屬性, 142

java.naming.security.credentials 屬性, 142

java.naming.security.principal 屬性, 142

Java 執行階段

指定路徑, 220, 229, 230, 234

適用於 Windows 服務, 67

Java 虛擬機器, 參閱JVM

javahome 選項, 67

JCA (J2EE 連接器架構), 271

JDBC 支援

配置, 84

關於, 76-77

驅動程式, 243

JDBC 型永久性

效能調校, 191

設定, 85-87

關於, 76-77

jms 連線服務, 72, 96

JMSDeliveryMode 訊息標頭欄位, 148

JMSExpiration 訊息標頭欄位, 148

JMSPriority 訊息標頭欄位, 149

JNDI

物件存放區, 35, 141

物件存放區屬性, 142-143, 149

查找, 50

查找名稱, 149, 150

jrehome 選項, 67

JVM

度量

參閱JVM 度量

效能影響, 185

效能調校, 188-189

JVM 度量

使用 imqcmd 度量, 171

使用代理程式記錄檔, 168

使用訊息型監視, 174

度量數目, 277-278

L

LDAP 伺服器

作為使用者儲存庫, 122

物件存放區屬性, 142

使用者儲存庫存取, 123

認證容錯移轉, 123

M

ManagedConnectionFactory JavaBean, 273

MDB, 參閱訊息驅動 Bean

messageSelector 啟動規格屬性, 274

N

NORMAL 服務類型, 72

nsswitch.conf 檔案 (Linux), 159

O

Oracle, 85, 87

P

password 資源介面屬性, 272

password 管理的連線工廠屬性, 273

PointBase, 85

R

reconnectAttempts 資源介面屬性, 272
 reconnectAttempts 管理的連線工廠屬性, 273
 reconnectEnabled 資源介面屬性, 272
 reconnectEnabled 管理的連線工廠屬性, 273
 reconnectInterval 資源介面屬性, 272
 reconnectInterval 管理的連線工廠屬性, 273
 reset messages 選項, 111
 ResourceAdapter JavaBean, 271
 RESTART 特性, 65

S

sendUndeliverableMsgsToDMQ 啟動規格屬性, 275
 SNTP, 63
 SSL
 加密, 和, 130-137
 連線服務
 參閱SSL 型連線服務
 啟用, 133
 關於, 79
 SSL 型連線服務
 設定, 130, 131
 啟動, 133-134
 ssladmin 連線服務, 130
 關於, 72, 97
 ssljms 連線服務, 130
 關於, 72, 96
 subscriptionDurability 啟動規格屬性, 274
 subscriptionName 啟動規格屬性, 274
 Sun Cluster, 配置, 242
 syslog, 81, 167

T

TCP, 72, 96
 TLS, 72, 96

U

ulimit 指令, 64
 update dst 子指令, 限制, 110

user 群組, 120
 刪除指定, 120
 預先定義的, 120
 userName 資源介面屬性, 272
 userName 管理的連線工廠屬性, 273

W

W32Time 服務, 63
 Windows 服務, 參閱服務 (Windows)

X

xntpd 常駐程式, 63
 工具, 管理, 參閱管理工具
 分散式作業事件, XA 資源管理員, 103
 可靠的傳送, 147-148
 可靠傳送, 效能取捨, 181
 生產環境
 設定, 34
 管理作業, 34-35
 維護, 34
 加密
 SSL 型服務, 130-137
 金鑰工具和, 80
 關於, 77, 79-80
 用戶端
 時鐘同步化, 63
 啟動, 69
 用戶端執行階段
 訊息流量調校, 192
 配置, 188
 用戶端應用程式
 影響效能的因素, 181-184
 範例, 288, 289, 290
 用戶端識別碼 (ClientID), 146-147
 銷毀長期訂閱, 102
 用法說明, 229
 目標受管理物件, 屬性, 149
 目標度量
 使用 imqcmd 查詢, 173
 使用 imqcmd 度量, 170, 172-173
 使用訊息型監視, 174

目標度量 (續)

- 度量數目, 281-284

主代理程式

- 指定, 157, 158
- 配置變更記錄, 161
- 無法使用, 162

主題

- 自動建立的, 238, 253
- 增加受管理物件, 151

代理程式

- HTTP 支援, 297
- HTTPS 支援, 303
- 互連
 - 參閱代理程式叢集
- 列出連線服務, 97
- 存取控制
 - 參閱授權
- 自動建立實體目標特性, 238-241
- 自動重新啟動, 65
- 更新特性, 93
- 作為 Windows 服務來執行, 65
- 限制運作方式, 74, 187
- 使用 SSL 啟動, 133-134
- 查詢, 92
- 重新啟動, 75, 95, 224
- 重新繼續, 94, 224
- 故障後回復, 75
- 度量
 - 參閱代理程式度量
- 記錄
 - 參閱記錄程式
- 記憶體管理, 74, 107, 187
- 訊息流量控制
 - 參閱訊息流量控制
- 訊息容量, 74, 93, 237, 256
- 時鐘同步化, 63
- 連線, 159
- 配置檔案
 - 參閱配置檔案
- 特性 (參照), 235, 259
- 停用的訊息佇列, 114
- 啟動時需要的權限, 64
- 移除, 68
- 管理, 89

代理程式 (續)

- 實例名稱, 218
- 實例配置特性, 83
- 暫停, 94, 224
- 叢集
 - 參閱代理程式叢集
- 關閉, 95
- 顯示特性, 92-93
- 代理程式 Windows 服務的啟動參數, 67
- 代理程式元件
 - 永久性服務, 71, 75-77
 - 安全性服務, 71, 77-80
 - 連線服務, 71
 - 路由服務, 71, 74
 - 監視服務, 71, 80-82
- 代理程式回應, 用戶端的等待期間, 267
- 代理程式故障和安全的連線, 313
- 代理程式度量
 - 使用 imqcmd, 95-96, 172, 173
 - 使用代理程式記錄檔, 168
 - 使用訊息型監視, 174
 - 度量訊息, 81
 - 度量數目 (表格), 278-279
 - 記錄程式特性, 81, 168, 251
 - 報告時間間隔, 記錄程式, 221
- 代理程式監視服務, 特性, 248-251
- 代理程式叢集
 - 安全代理程式互通連線, 159
 - 使用的原因, 187
 - 架構, 187
 - 連線代理程式, 159
 - 配置特性, 157, 252
 - 配置檔案, 157, 158, 159, 252
 - 配置變更記錄, 161
 - 效能影響, 187
 - 將代理程式增加至, 160
 - 實體目標的複製, 107
 - 暫停實體目標, 111
- 永久性
 - JDBC
 - 參閱JDBC 永久性
 - JDBC 型
 - 參閱JDBC 型永久性
 - 安全性, 87

- 永久性 (續)
 - 特性, 241-242
 - 資料存放區
 - 參閱資料存放區
 - 選項 (圖), 75
 - 檔案式, 75-76
 - 關於, 75
- 永久性服務, 代理程式, 71, 75-77
- 安全性
 - 加密
 - 參閱加密
 - 物件存放區, 適用於, 142
 - 授權
 - 參閱授權
 - 管理員
 - 參閱安全性管理員
 - 認證
 - 參閱認證
- 安全性服務, 代理程式, 71, 77-80
- 安全性管理員
 - 特性, 245-248
 - 關於, 77
- 安全通訊端層標準, 參閱SSL
- 存在時間, 參閱訊息過期
- 存取規則, 126
- 存取控制檔案
 - 用於, 124
 - 存取規則, 126
 - 位置, 288, 289
 - 版本, 125
 - 格式, 125
- 同步化
 - 記憶體至磁碟, 85
 - 時鐘, 63
- 自我簽署的憑證, 130-135, 304
- 自動建立實體目標
 - 存取控制, 79, 129-130
 - 特性 (表格), 238-241
- 自動重新連線
 - 限制, 146
 - 屬性, 145
- 防火牆, 295
- 系統時鐘同步化, 63
- 更新
 - 代理程式, 93
 - 連線服務, 98, 101, 225
- 刪除, 代理程式實例, 68
- 刪除目標, 112
- 作業系統
 - 效能影響, 185
 - 調校 Solaris 效能, 188
- 作業事件
 - 回復, 103, 228
 - 效能影響, 182-183
 - 資訊, 228
 - 管理, 103-104
 - 確定, 104, 228
- 伺服器故障和安全的連線, 313
- 佇列
 - 自動建立的, 238, 253
 - 增加受管理物件, 151
- 佇列負載平衡傳送
 - 特性, 107, 240, 260
- 版本, 229
- 協定
 - 參閱傳輸協定
- 協定類型
 - HTTP, 72, 96
 - TCP, 72, 96
 - TLS, 72, 96
- 物件存放區, 141-143
 - LDAP 伺服器, 141-143
 - LDAP 伺服器屬性, 142
 - 位置, 287, 288, 289
 - 檔案系統, 143
 - 檔案系統存放區屬性, 143
- 物件存放區的位置, 142
- 受管理物件
 - XA 連線工廠
 - 參閱連線工廠受管理物件
- 主題
 - 參閱主題
- 更新, 153
- 刪除, 152
- 佇列
 - 參閱佇列

受管理物件 (續)

- 物件存放區
 - 參閱物件存放區
- 查詢, 153
- 清單, 152
- 需要的資訊, 149
- 管理, 141
- 屬性 (參照), 263

限制運作方式

- 代理程式, 74
- 實體目標, 107, 260

服務 (Windows)

- Java 執行階段適用於, 67
- 代理程式執行爲, 65
- 重新配置, 66
- 啓動參數適用於, 67
- 移除代理程式, 68
- 疑難排解啓動問題, 67

服務類型

- ADMIN, 72
- NORMAL, 72

使用者名稱, 266

- 格式, 121
- 預設, 118

使用者群組, 預設, 79

使用者儲存庫

- LDAP, 122
- LDAP 伺服器, 123
- user 群組, 120
- 平台相依性, 119, 232
- 平面檔案, 118
- 初始項目, 118
- 位置, 287, 288, 289
- 使用者狀態, 120
- 管理, 121
- 寫入, 121
- 關於, 78

使用率, 113

長期訂閱

- 列出, 102, 228
- 效能影響, 183-184
- 清除訊息, 228
- 管理, 102
- 銷毀, 102, 227

金鑰工具, 80

金鑰庫

- 檔案, 132, 304

金鑰對

- 重新產生, 132
- 產生, 132

所有指令的語法, 217-218

查詢

- 代理程式, 92-93
- 連線服務, 97-98, 101

流量控制, 參閱訊息流量控制

重新連線, 自動, 參閱自動重新連線

重新啓動代理程式, 95, 224

重新繼續

- 代理程式, 94, 224
- 連線服務, 100, 225
- 實體目標, 111

指令行公用程式

- imqbrokerd, 請參閱 imqbrokerd 指令, 35
- imqcmd, 請參閱 imqcmd 指令, 35
- imqdbmgr, 請參閱 imqdbmgr 指令, 35
- imqkeytool, 請參閱 imqkeytool 指令, 35
- imqobjmgr, 請參閱 imqobjmgr 指令, 35
- imqsvcadm, 請參閱 imqsvcadm 指令, 35
- imqusermgr, 請參閱 imqusermgr 指令, 35
- 基本語法, 217
- 說明, 229
- 關於, 35
- 顯示版本, 229

指令行語法, 217

指令選項, 配置置換, 69

指令檔, 154

指導, 37

迴路位址, 159

度量

- 主題目標, 81-82, 174

訊息

- 參閱度量訊息

資料

- 參閱度量資料

關於, 80

度量訊息

- 關於, 174
- 類型, 81, 174

度量資料

代理程式

- 參閱代理程式度量

- 使用 imqcmd 度量, 171

- 使用代理程式記錄檔, 168

- 使用訊息型監視 API, 174-175

連線服務

- 參閱連線服務度量

實體目標

- 參閱實體目標度量

度量監視工具

- Message Queue 指令公用程式 (imqcmd), 169-173

- Message Queue 記錄檔, 168

- 比較的, 163-164

- 訊息型監視 API, 174-176

- 負載平衡的佇列傳送, 效能調校, 191-192

- 記錄, 參閱記錄程式

記錄程式

- 自動重建準則, 167

- 重新導向記錄訊息, 167

- 度量資訊, 251

- 訊息格式, 166

- 設定特性, 166

- 停用的訊息格式, 169

- 種類, 165

- 層級, 81, 165, 221, 248, 254

- 寫入至主控台, 81, 222, 249, 254

- 輸出通道, 81, 164, 167

- 關於, 81

- 變更配置, 166

記錄檔

- 名稱, 165

- 自動重建條件, 81

- 自動重建準則, 167, 249, 254

- 自動重建頻率, 165

- 設定特性, 166

- 停用的訊息之記錄功能, 169

- 報告度量, 168

- 預設位置, 287, 288, 289

- 變更預設位置, 165

記憶體管理

- 使用實體目標特性, 107

- 針對代理程式, 74

- 效能調校, 191

訊息

- 分段程序, 76

- 內文類型與效能, 184

- 可靠的傳送, 147-148

- 目標限制, 239, 260

- 代理程式限制, 74, 93, 237, 256

- 永久性, 75

- 收回已過期的, 74, 238, 254

流量控制

- 參閱訊息流量控制

度量訊息

- 參閱度量訊息

- 容量, 和效能, 184

- 從實體目標清除, 111, 227

- 實體目標限制, 107

- 暫停流量, 110

- 訊息分段程序, 76

- 訊息服務架構, 187

- 訊息服務效能, 185-188

訊息流量控制

- 代理程式, 74, 107

- 限制, 192-194

- 計數, 192

- 效能影響, 192

- 效能調校, 192

- 屬性, 147-148

- 訊息過期, 時鐘同步化與, 63

- 訊息標頭置換, 148-149

訊息驅動 Bean

- 資源介面配置, 271, 274

- 通道 Servlet 連線, 313

- 時間同步化服務, 63

- 時鐘同步化, 63

連接埠對映器

- 連接埠指定, 218

- 關於, 72

連線

- 由檔案描述元限制所限制, 64

- 列出, 100, 225

- 自動重新連線

- 參閱自動重新連線

- 伺服器或代理程式發生故障, 313

- 查詢, 101, 226

- 效能影響, 185-187

連線 (續)

容錯移轉

參閱自動重新連線

連線工廠受管理物件

JMS 特性支援屬性, 148

可靠性與流量控制屬性, 147-148

用戶端標識屬性, 146-147

佇列瀏覽器運作方式屬性, 148, 268

連線處理屬性, 144-146

置換訊息標頭欄位, 148

標準訊息特性, 268-269

應用程式伺服器支援屬性, 268

屬性, 144-149

連線代理程式, 159

連線服務

admin, 72, 96

cluster

參閱cluster 連線服務

HTTP

參閱HTTP 連線

httpjms, 72, 96

HTTPS

參閱HTTPS 連線

httpsjms, 72, 96

參閱httpsjms 連線服務

jms, 72, 96

SSL 型, 132

ssladmin

參閱ssladmin 連線服務

ssljms

參閱ssljms 連線服務

存取控制, 77, 245

在啟動時啟動, 235

更新, 98, 101, 225

協定類型, 72

服務類型, 72

查詢, 97, 101

重新繼續, 100, 225

指令影響, 225

度量資料

參閱連線服務度量

連接埠對映器

參閱連接埠對映器

特性, 98, 235-237

連線服務 (續)

執行緒分配, 98

執行緒池管理, 73

暫停, 100, 225

叢集, 159

顯示特性, 97-98

連線服務, 代理程式, 71

連線服務度量

使用 imqcmd 度量, 99, 172, 173

度量數目, 279-280

配置檔案, 82

代理程式 (圖), 82

安裝, 82

位置, 287, 288, 289

預設, 82

實例, 83, 158, 287, 288, 289

範本, 287, 288, 289

範本位置, 287, 288, 289

編輯, 83-84

叢集, 157, 158, 159, 252

配置變更記錄, 161

備份, 162

復原, 162

效能

可靠性取捨, 181

指標, 178

效能評定, 178-179

瓶頸, 181

基準式樣, 179

測量, 178

最佳化

參閱效能調校

疑難排解, 195

監視

參閱效能監視

影響因素

參閱效能因素

調校

參閱效能調校

關於, 177-179

效能因素

JVM, 185

代理程式限制運作方式, 187

作業系統, 185

效能因素 (續)

- 作業事件, 182-183
 - 長期訂閱, 183-184
 - 訊息內文類型, 184
 - 訊息服務架構, 187
 - 訊息流量控制, 192
 - 訊息容量, 184
 - 連線, 185-187
 - 硬體, 185
 - 傳送模式, 182
 - 傳輸協定, 186-187
 - 資料存放區, 187-188
 - 確認模式, 183
 - 選擇器, 184
 - 檔案同步化, 242, 255
- 效能評定, 效能, 178-179
- 效能監視
- 工具
 - 參閱度量監視工具
 - 度量資料
 - 參閱度量資料

效能調校

- 用戶端執行階段調整, 192-194
- 代理程式調整, 191-192
- 系統調整, 188-191
- 程序簡介, 177

特性

- 代理程式實例配置, 83
- 代理程式監視服務, 248-251
- 永久性, 241-242
- 安全性, 245-248
- 自動建立, 238-241
- 記錄程式, 248-251
- 記憶體管理, 107
- 連線服務, 235-237
- 路由服務, 237-238
- 與 JDBC 相關, 83, 243-245
- 實體目標
 - 參閱實體目標, 特性
- 語法, 84
- 叢集配置, 252

瓶頸, 效能, 181

停用的訊息

- 另請參閱停用的訊息佇列

停用的訊息 (續)

- 記錄, 81
- 停用的訊息佇列
 - maxNumMsgs 值, 115
 - maxTotalMsgBytes 值, 115
 - 限制運作方式, 115
 - 記錄, 81, 115-116
 - 配置, 114-116

執行緒池管理

- 共用執行緒, 73
- 專屬執行緒, 73
- 關於, 73

產生器

- 目標限制, 240, 260
- 實體目標限制, 107

授權

- 另請參閱存取控制
- 使用者群組, 79
- 啟動選項, 221
- 管理, 124-130
- 關於, 79

啓動

- SSL 型連線服務, 133-134
- 用戶端, 69

移除

- 代理程式, 68
- 實體目標, 112

密碼

- JDBC, 139
- LDAP, 139
- SSL 金鑰存放區, 139
- SSL 金鑰庫, 132, 220
- 命名慣例, 121
- 密碼檔案
 - 參閱密碼檔案
- 預設, 266
- 管理員, 122
- 編碼, 246

密碼檔案

- 代理程式配置特性, 79, 246
- 位置, 139, 288, 289
- 使用, 138-139
- 指令行選項, 220

- 清除, 實體目標的訊息, 111

- 硬體, 效能影響, 185
- 開發環境管理作業, 33
- 傳送模式, 效能影響, 182
- 傳輸協定
 - 協定類型
 - 參閱協定類型
 - 相對速度, 186
 - 效能影響, 186-187
 - 效能調校, 189-191
- 資料存放區
 - JDBC 相容, 76-77
 - 內容, 84
 - 平面檔案, 75-76
 - 位置, 287, 288, 289
 - 重設, 219
 - 配置, 84
 - 效能影響, 187-188
 - 磁碟同步化, 85
 - 壓縮, 76
 - 關於, 75
- 資源介面, 271
 - 重新連線, 272, 273
- 路由服務, 特性, 237-238
- 路由服務, 代理程式, 71, 74
- 置換
 - 在指令行中, 69
 - 訊息標頭, 148-149
- 管理工具, 35-36
 - 指令行公用程式, 35
 - 管理主控台, 36
- 管理主控台
 - 指導, 37
 - 啟動, 38
- 管理作業
 - 生產環境, 34-35
 - 開發環境, 33
- 管理員密碼, 122
- 疑難排解, 195
 - Windows 服務啟動, 67
- 實例目錄
 - 和實例配置檔案, 122
 - 和檔案式資料存放區, 85
 - 移除, 68
- 實例配置檔案, 參閱配置檔案
- 實體目標
 - 用於傳送的批次訊息, 107, 240, 261
 - 收回磁碟空間, 113
 - 自動建立, 129
 - 更新特性, 110
 - 更新屬性, 226
 - 取得相關資訊, 108
 - 取得資訊, 227
 - 限制運作方式, 107, 260
 - 建立, 106-108
 - 使用停用的訊息佇列, 114
 - 重新繼續, 111
 - 相關資訊, 108-109
 - 度量
 - 參閱實體目標度量
 - 特性, 259-261
 - 特性值, 108
 - 停用的訊息佇列, 114-116
 - 清除訊息, 111, 227
 - 清單, 108
 - 管理, 105
 - 磁碟使用, 112-114
 - 暫時, 108
 - 暫停, 110-111, 226
 - 銷毀, 112
 - 壓縮, 112-114
 - 壓縮檔案式資料存放區, 113, 227
 - 叢集中限制的範圍, 107, 240, 261
 - 類型, 108, 226
 - 顯示特性值, 108-109
- 實體目標屬性, 259-261
- 監視, 參閱效能監視
- 監視服務, 代理程式, 71, 80-82
- 說明 (指令行), 229
- 認證
 - 另請參閱存取控制
 - 管理, 117-124
 - 關於, 79
- 磁碟空間
 - 收回, 113
 - 實體目標使用, 112
- 暫時實體目標, 108
- 暫停
 - 代理程式, 94, 224

暫停 (續)

- 連線服務, 100, 225
- 實體目標, 110-111, 226
- 寫入作業 (針對檔案式存放區), 85
- 稽核記錄, 140
- 範例應用程式, 288, 289, 290
- 銷毀實體目標, 112
- 選擇器
 - 效能影響, 184
 - 關於, 184
- 憑證, 130-135, 304
- 檔案式永久性, 75-76
- 檔案同步化
 - imq.persist.file.sync.enabled 選項, 242, 255
 - 與 Sun Cluster, 242, 255
- 檔案描述元, 64
- 檔案描述元限制, 連線限制和, 64
- 壓縮
 - 實體目標, 112-114
 - 檔案式資料存放區, 76
- 優先權 (配置特性的), 82
- 叢集, 參閱代理程式叢集
- 叢集的目錄查找 (Linux), 159
- 叢集連線服務, 159
 - 主機名稱或 IP 位址, 157, 252
 - 連接埠號, 157, 252
 - 網路傳輸, 158, 252
- 叢集配置特性, 157, 251, 252
- 叢集配置檔案, 157, 158, 159, 252
- 簡易網路時間協定, 63
- 關閉代理程式, 95, 224
 - 作為 Windows 服務, 68
- 權限
 - 內嵌式資料庫, 86
 - 存取控制特性檔案, 79, 125
 - 使用者儲存庫, 119, 232
 - 金鑰庫, 304
 - 計算, 126-127
 - 密碼檔案, 138
 - 資料存放區, 76
 - 管理服務, 79
- 顯示產品版本, 229