



Sun Java™ System

Message Queue 3

佇列程式

2005Q1

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

文件號碼 819-2224

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

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

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

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

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

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

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

目錄

圖目錄	7
表目錄	9
前言	11
本書適用對象	12
閱讀本書之前	12
本書架構	12
本書使用慣例	13
文字慣例	13
目錄變數慣例	14
相關文件	16
Message Queue 文件集	16
線上說明	16
JavaDoc	17
範例用戶端應用程式	17
Java Message Service (JMS) 規格	17
協力廠商文件	17
Sun 歡迎您提出寶貴意見	18
第 1 章 概念基礎	19
企業訊息傳送系統	20
企業訊息傳送系統的要求	20
集中式 (MOM) 訊息	20
基本訊息服務架構	21
Java Message Service (JMS) 基本資訊	22
JMS 訊息結構	23
JMS 程式設計模型	24
程式設計物件	24
程式設計網域：訊息傳送模型	25
可靠的訊息傳送	26

確認 / 作業事件	27
永久性儲存	28
JMS 受管理物件	28
第 2 章 Message Queue 簡介	31
訊息服務架構	32
訊息伺服器	33
用戶端執行階段	33
連線處理	35
用戶端標識	35
訊息分散至用戶	35
確保可靠的訊息傳送	36
訊息流量控制	37
覆寫訊息標頭值	37
其他功能	37
受管理物件	37
透過 JNDI 使用受管理物件	38
物件儲存	39
管理工具	39
產品功能	40
整合支援功能	40
多重傳輸支援	40
C 用戶端介面	41
SOAP (XML) 訊息傳送支援	41
J2EE 資源介面	42
安全性功能	42
延伸性功能	43
可延伸的連線功能	43
代理程式叢集	43
佇列傳送至多個用戶	43
可用性功能	44
訊息服務穩定性	44
自動重新連線到訊息伺服器	44
透過 Sun Cluster 提供高度可用性	44
可管理性功能	45
強固的管理工具	45
基於訊息監視的 API	45
可調校的效能	45
彈性的伺服器配置功能	45
可配置的永久性	46
LDAP 伺服器支援	46
產品版本	46
企業版	47

平台版	47
Sun 產品環境中的 Message Queue	48
第 3 章 可靠訊息傳送	49
訊息在系統中的行程	50
訊息傳送處理	51
訊息生產	52
訊息處理和訊息路由	52
佇列目標	52
主題目標	53
訊息使用	54
用戶端確認	54
作業事件	56
訊息使用終了	56
訊息的正常刪除	56
訊息的不正常刪除	57
效能問題	57
第 4 章 訊息伺服器	59
代理程式架構	60
代理程式元件	61
連線服務	61
連接埠對映器	62
執行緒儲存區管理程式	63
HTTP/HTTPS 支援	63
訊息路由器	64
實體目標	64
記憶體資源管理	66
持續性管理程式	67
安全性管理程式	68
認證	69
授權	69
加密	70
監視服務	70
度量產生器	70
記錄程式	71
度量訊息產生者 (企業版)	71
開發與生產環境	72
開發環境與工作	72
出廠即可使用的配置	72
開發慣例	72
生產環境與工作	73

設定作業	73
維護作業	74
第 5 章 代理程式叢集	75
叢集架構	76
訊息傳送	77
叢集配置	77
叢集同步化	77
部署環境	78
開發環境	78
生產環境	79
第 6 章 Message Queue 和 J2EE	81
JMS/J2EE 程式設計：訊息驅動 Bean	82
J2EE 應用程式伺服器支援	83
JMS 資源介面	84
附錄 A 選擇性 JMS 功能的 Message Queue 實作	87
字彙表	89
索引	93

圖目錄

圖 1-1	集中式訊息傳送和點對點訊息傳送	21
圖 1-2	訊息服務架構	22
圖 1-3	JMS 程式設計物件	24
圖 2-1	Message Queue 服務架構	32
圖 2-2	用戶端執行階段與訊息傳送作業	34
圖 2-3	訊息傳送至 Message Queue 用戶端執行階段	36
圖 3-1	訊息傳送步驟	50
圖 4-1	代理程式元件	60
圖 4-2	連線服務支援	62
圖 4-3	HTTP/HTTPS 支援架構	63
圖 4-4	持續性管理程式支援	68
圖 4-5	安全性管理程式支援	69
圖 4-6	監視服務支援	71
圖 5-1	叢集架構	76
圖 6-1	與 MDB 進行訊息傳送	83

表目錄

表 1	本書內容與架構	12
表 2	文件慣例	13
表 3	Message Queue 目錄變數	14
表 4	Message Queue 文件集	16
表 1-1	訊息內文類型	23
表 1-2	JMS 程式設計網域及物件	26
表 2-1	Message Queue 受管理物件類型	38
表 2-2	功能比較：企業版與平台版	46
表 4-1	主代理程式服務元件和功能	60
表 4-2	代理程式支援的連線服務	61
表 A-1	選擇性 JMS 功能	87

前言

本書（「Sun Java™ System Message Queue 3 2005Q1 技術概述」）提供 Message Queue 訊息傳送服務的技術、概念、架構與功能的說明。

因此，「Message Queue 技術概述」提供在 Message Queue 文件集中其他書籍的基礎。閱讀 Message Queue 文件集中的其他書籍之前，您應該先閱讀本書。

本前言包含以下各節：

- [第 12 頁](#) 「本書適用對象」
- [第 12 頁](#) 「閱讀本書之前」
- [第 12 頁](#) 「本書架構」
- [第 13 頁](#) 「本書使用慣例」
- [第 16 頁](#) 「相關文件」
- [第 17 頁](#) 「協力廠商文件」
- [第 18 頁](#) 「Sun 歡迎您提出寶貴意見」

本書讀者對象

本書適用於管理員、應用程式開發者與其他計劃使用 Message Queue 產品的使用者，或是希望瞭解產品技術、概念、架構與功能的使用者。

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

應用程式開發者需負責撰寫 Message Queue 用戶端應用程式 (使用 Message Queue 服務)，以與其他用戶端應用程式交換訊息。本書假設您不瞭解 Java Message Service (JMS) 規格 (由 Message Queue 服務實作)。

閱讀本書之前

閱讀本書沒有任何先決條件。閱讀 Message Queue 開發者與管理指南之前，您應該閱讀本書以瞭解 Message Queue 基本概念。

本書架構

本書的設計是從頭到尾完整閱讀；每個章節中所包含的資訊都是以前一個章節為基礎。下表簡要描述了每章的內容：

表 1 本書內容與架構

章	說明
第 1 章「概念基礎」	提供 Message Queue 的概念背景，說明企業訊息傳送系統，並介紹 Java Message Service 概念與專有名詞。
第 2 章「Message Queue 簡介」	討論 Message Queue 服務的架構與說明其企業發展 (enterprise-strength) 的功能。
第 3 章「可靠訊息傳送」	說明 Message Queue 服務如何為訊息傳送應用程式提供可靠的訊息傳送。
第 4 章「訊息伺服器」	討論代理程式的內部結構，說明各種代理程式元件與其功能。說明在開發與生產環境使用 Message Queue 的不同方式。
第 5 章「代理程式叢集」	討論 Message Queue 代理程式叢集的架構與內部功能。
第 6 章「Message Queue 和 J2EE」	探索在 J2EE 平台環境中實作 JMS 支援的結果。

表 1 本書內容與架構 (續)

章	說明
附錄 A 「選擇性 JMS 功能的 Message Queue 實作」	說明 Message Queue 產品如何處理 JMS 選項項目。
字彙表	提供使用 Message Queue 時，可能遇到的專有名詞和概念的資訊。

本書使用慣例

此節提供了有關本文件中所使用慣例的資訊。

文字慣例

表 2 文件慣例

格式	說明
ABC123 <i>斜體</i>	斜體文字表示定位字元。請在看到斜體文字時將其取代為適當的子句或值。斜體文字也用於指定需要強調的文件標題，或用於引入的單字或片語。
ABC123 固定間距字體	固定間距字體文字表示範例程式碼、在指令行上所輸入的指令、目錄名稱、檔案名稱或路徑名稱、錯誤訊息文字、類別名稱、方法名稱 (包括簽名中的所有元素)、套裝軟體名稱、保留字以及 URL。
[]	指示指令行語法敘述中選用值的方括號。
全部大寫	所有大寫文字表示檔案系統類型 (GIF、TXT、HTML 等等)、環境變數 (IMQ_HOME) 或縮寫字 (Message Queue、JSP)。
「書名」	書名
鍵 + 鍵	透過加號連接在一處並同時按下的按鍵：Ctrl+A 表示同時按下這兩個鍵。
鍵 - 鍵	透過連字號連接在一處的連續按下的按鍵：Esc-S 表示按下 Esc 鍵，然後釋放它，接著按下 S 鍵。

目錄變數慣例

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

表 3 Message Queue 目錄變數

變數	說明
IMQ_HOME	<p>它通常在 Message Queue 文件中使用，指的是 Message Queue 基底目錄 (根安裝目錄)：</p> <ul style="list-style-type: none"> 在 Solaris 上，沒有 Message Queue 根安裝目錄。因此，未在 Message Queue 文件中使用 IMQ_HOME 來指代 Solaris 上的檔案位置。 在 Solaris 上，Sun Java System Application Server 的 Message Queue 根安裝目錄位於 Application Server 基底目錄下的 /imq。 在 Windows 上，Message Queue 根安裝目錄由 Message Queue 安裝程式設定 (依預設，該目錄為 C:\Program Files\Sun\MessageQueue3)。 在 Windows 上，Sun Java System Application Server 的 Message Queue 根安裝目錄位於 Application Server 基底目錄下的 /imq。 在 Linux 上，沒有 Message Queue 根安裝目錄。因此，未在 Message Queue 文件中使用 IMQ_HOME 來指定 Linux 上的檔案位置。
IMQ_VARHOME	<p>這是儲存 Message Queue 暫存檔或動態建立的配置檔與資料檔的 /var 目錄。可以將它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 試用版，IMQ_VARHOME 預設為 IMQ_HOME/var 目錄。 在 Windows 上，IMQ_VARHOME 預設為 IMQ_HOME\var 目錄。 在 Windows 上，針對 Sun Java System Application Server，IMQ_VARHOME 預設為 IMQ_HOME\var 目錄。 在 Linux 上，IMQ_VARHOME 預設為 /var/opt/imq 目錄。

表 3 Message Queue 目錄變數 (續)

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

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

相關文件

除了本書以外，Message Queue 還提供了其他文件資源。

Message Queue 文件集

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

表 4 Message Queue 文件集

文件	使用者	說明
Message Queue Installation Guide	開發者與管理員	解釋如何在 Solaris、Linux 以及 Windows 平台上安裝 Message Queue 軟體。
Message Queue 版本說明	開發者與管理員	包括對新功能、限制、已知錯誤以及技術性注意事項的描述。
Message Queue 管理指南	管理員，也建議開發者使用	提供使用 Message Queue 管理工具執行管理工作時所需的背景與資訊。
Message Queue Developer's Guide for Java Clients	開發者	提供使用 Java 用戶端程式的開發者快速入門教學和程式設計資訊，以使用 JMS 和 SOAP/JAXM 規格的 Message Queue 實作。
Message Queue Developer's Guide for C Clients	開發者	提供 C 用戶端程式的開發者程式設計和參照文件，以使用 C 介面 (C-API) 到 Message Queue 訊息服務的功能。

線上說明

Message Queue 包括用於執行 Message Queue 訊息服務管理工作的指令行公用程式。若要存取這些公用程式的線上說明，請參閱「Message Queue 管理指南」。

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

JavaDoc

下列位置提供 JavaDoc 格式的 Message Queue Java 用戶端 API (包括 JMS API) 文件：

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

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

範例用戶端應用程式

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

請參閱該目錄及其各個子目錄中的 README 檔案。

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。

協力廠商文件

協力廠商 URL 為本文件的參照資料且提供了額外的相關資訊。

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

Sun 歡迎您提出寶貴意見

Sun 致力於提高文件品質，因此誠心歡迎您提出意見與建議。

若要提出意見，請到 <http://docs.sun.com> 並且按一下 [傳送您的回饋意見]。請在線上表單中提供文件標題和文件號碼。在書的標題頁或文件的上方可找到文件號碼，通常是 7 或 9 個數字。例如，本書的標題為「Sun Java System Message Queue 3 2005Q1 技術概述」，而文件號碼為 819-2224。

提供意見與建議時，您可能需要在表單中提供文件的英文標題及文件號碼。此文件的英文文件號碼及標題為：819-0069，「Sun Java System Message Queue 3 2005Q1 Technical Overview」。

概念基礎

Sun Java™ System Message Queue (Message Queue) 提供可靠的[非同步訊息傳送](#)，能整合企業中的分散式應用程式及元件。在不同平台和作業系統上執行的程序可連線至此服務來進行互動。

Message Queue 是基於標準的[訊息傳送](#)解決方案，其實作 Java™ Message Service (JMS) 開放式標準。此外，Message Queue 提供互通性、安全性、延伸性、可用性、可管理性和其他大規模企業部署所需的功能。

本章提供 Message Queue 的概念基礎。它涵蓋以下主題：

- [第 20 頁「企業訊息傳送系統」](#)
- [第 22 頁「Java Message Service \(JMS\) 基本資訊」](#)

如果您已熟悉 JMS 的概念和術語，您可跳至[第 2 章「Message Queue 簡介」](#)。

企業訊息傳送系統

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

為使大量元件能夠同時交換訊息，並支援高密度流量，訊息傳送不能取決於用戶是否可以接收訊息。如果訊息用戶忙碌或離線，系統必須能讓訊息在用戶準備就緒時被接收。這樣的訊息傳送和接收的去耦合便是所謂的非同步訊息傳送。

非同步訊息模型在整合複雜系統的工作上提供優越的性能，在執行工作時元件不用也不需保留其他元件。由於非同步訊息放棄某些同步訊息系統允許的控制，它大大增加了元件相互作用的靈活性。它也增加了強健性，只有一個元件的故障並不會轉變成全部故障。

企業訊息傳送系統的要求

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

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

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

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

可用性。 訊息傳送系統必須很少發生運作當機。這表示當故障發生時，系統含有足夠的容錯性能繼續提供訊息傳送服務。

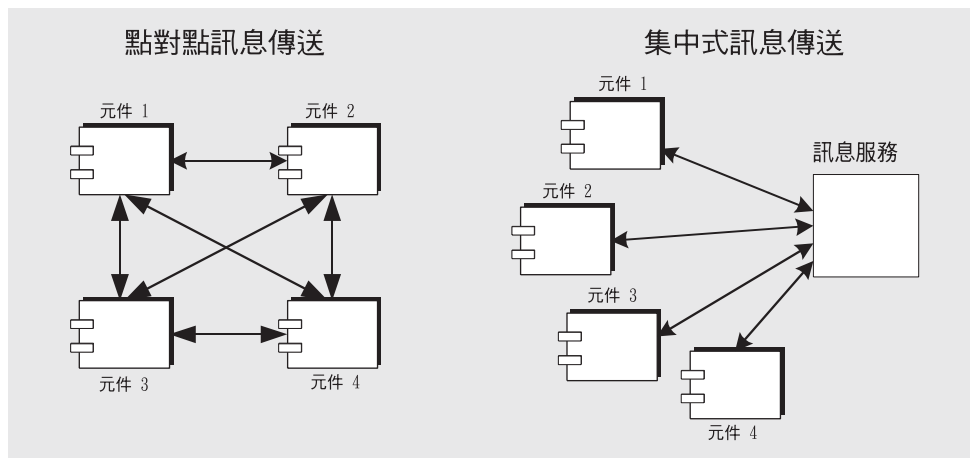
可管理性。 訊息傳送系統必須提供監視和管理訊息傳送的工具。管理員必須能最佳化系統資源和調校系統效能。

集中式 (MOM) 訊息

Message Queue 使用集中式訊息傳送系統，如圖 1-1 中所示。在此系統中，每個訊息傳送元件只需與一個中央訊息服務保持連線。元件透過定義完善的介面與訊息服務進行互動。

一個替代的點對點系統，其中每個訊息傳送元件與其他所有元件均保持連線，如左圖中的說明。點對點系統允許快速、安全且可靠的傳送；但是支援可靠性和安全性的程式碼必須駐留在每個元件中。訊息的傳送和接收緊密耦合，使得非同步傳送很難實現。隨著將元件不斷地加入系統，連線數目將成幾何級增加，因此系統的延伸性很差。在點對點系統中的集中式管理也會出現問題。

圖 1-1 集中式訊息傳送和點對點訊息傳送



在集中式系統中，企業訊息傳送的首選方法是，訊息服務提供元件之間的訊息傳送和路由，並負責可靠傳送及安全性。因為在此系統中的元件無密切關聯，非同步訊息傳送可輕易實現。

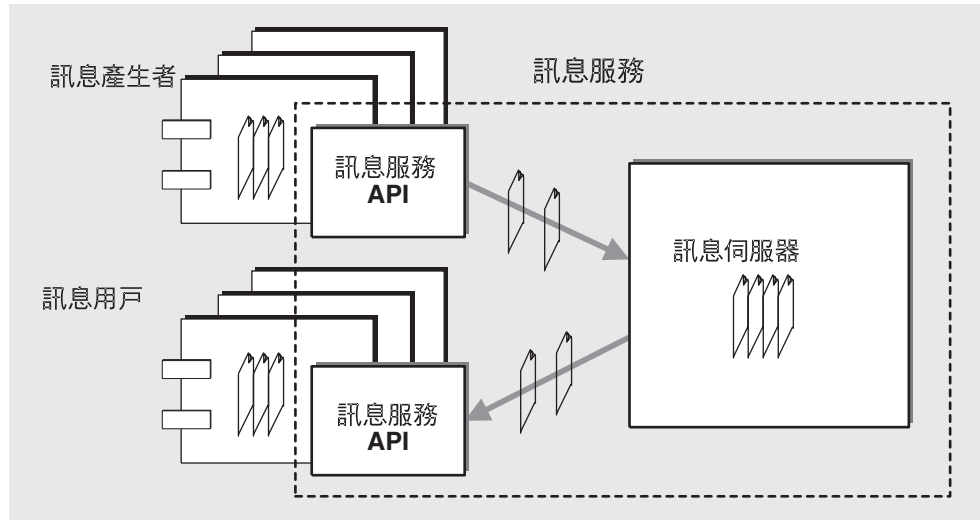
隨著將元件不斷地加入系統，連線的數目線性增長，這樣就可以透過調整訊息服務來更輕鬆地調整系統。除了連線訊息傳送用戶端之外，中央訊息服務也提供管理介面以配置運作方式、監督效能及調校服務以滿足每個訊息傳送用戶端要求。

基本訊息服務架構

集中式訊息傳送系統的基本架構如圖 1-2 所示。它包含訊息產生者和訊息用戶，兩者透過共用訊息服務來交換訊息。任何數目的訊息產生者和用戶均可駐留在同一個訊息傳送元件（或應用程式）中。

訊息產生者使用訊息服務程式設計 API 傳送訊息至訊息伺服器。訊息伺服器路由和傳送訊息至一個或多個已註冊接收此訊息的訊息用戶。用戶使用訊息服務程式設計 API 接收訊息。訊息服務負責保證將該訊息傳送至所有相應的用戶。

圖 1-2 訊息服務架構



也許對這個程序最好的比喻是交換郵件。雖然一封郵件最後終會寄到收件人手中，但它在收件人從郵箱中取出前會繞經郵局、並停留在數個中繼站中。

Java Message Service (JMS) 基本資訊

Message Queue 是實作 Java Message Service (JMS) 開放式標準的企業訊息傳送系統。它是一個 *JMS 提供者*。所以 JMS 概念對於理解 Message Queue 服務的工作原理很重要。

JMS 規格規定了一組管理可靠的、非同步的訊息傳送的規則和語義。該規格指定訊息結構、程式設計模型和 API。

本節說明了理解本書其餘各章所需的 JMS 概念和術語。它涵蓋以下主題：

- [第 23 頁「JMS 訊息結構」](#)
- [第 24 頁「JMS 程式設計模型」](#)
- [第 26 頁「可靠的訊息傳送」](#)
- [第 28 頁「JMS 受管理物件」](#)

JMS 訊息結構

在 Message Queue 中，資料的交換使用 JMS 訊息。依據 JMS 規格，由產生用戶端所建立的訊息由以下三部分組成：標頭、特性和內文。

標頭

每個 JMS 訊息都需要標頭。標頭欄位包含用來路由和辨別訊息的值。

標頭值可以數個方法來設定：

- 由 JMS 提供者設定，在產生或傳送訊息的過程中自動產生
- 由生產型用戶端設定，透過在建立訊息產生者時指定的設定
- 由生產型用戶端設定，根據訊息基礎在訊息上設定

關於 JMS 所定義的標頭欄位的詳細資訊，請參閱「Message Queue Developer's Guide for Java Clients」或「Message Queue Developer's Guide for C Clients」。標頭欄位能讓您定義訊息目標、到期時間、優先順序等等。

特性

訊息可包含選擇性的標頭欄位，這些稱為*特性*。它們被指定為特性名稱及特性值對。特性，可想成訊息標頭的延伸，可包含關於資料由什麼程序建立、建立的時間以及每個資料的結構等資訊。JMS 提供者也可加入影響訊息處理的特性，例如訊息該如何壓縮或在使用終了時如何銷毀。

JMS 提供者可使用訊息特性如*選擇器*將訊息排序和路由。生產型用戶端可在訊息中放入應用程式特定的特性，使用型用戶端可以選擇只接收某些特定特性值的訊息。例如，使用型用戶端可指明只對在 New Jersey 州的兼差員工的薪資訊息有興趣。不符合指定選取條件的訊息將不會傳送到用戶端。

選擇器簡化使用型用戶端的工作，並消除了不需要這些訊息的用戶端傳送訊息的耗用時間。但是，它們也使得處理選取條件的訊息服務增加了一些耗用時間。JMS 規格中對訊息選擇器的語法和語義進行了概述。

訊息內文類型

JMS 訊息的類型決定內文的內容，如表 1-1 中所指定。

表 1-1 訊息內文類型

類型	說明
StreamMessage	訊息內文包含 Java 原始值的串流。它會被依序寫入及讀取。
MapMessage	訊息內文包含名稱 - 值對的組合。項目的順序未加以定義。

表 1-1 訊息內文類型 (續)

類型	說明
TextMessage	訊息內文包含 Java 字串，如 XML 訊息。
ObjectMessage	訊息內文包含序列化 Java 物件。
BytesMessage	訊息內文包含未解譯位元組的串流。

JMS 程式設計模型

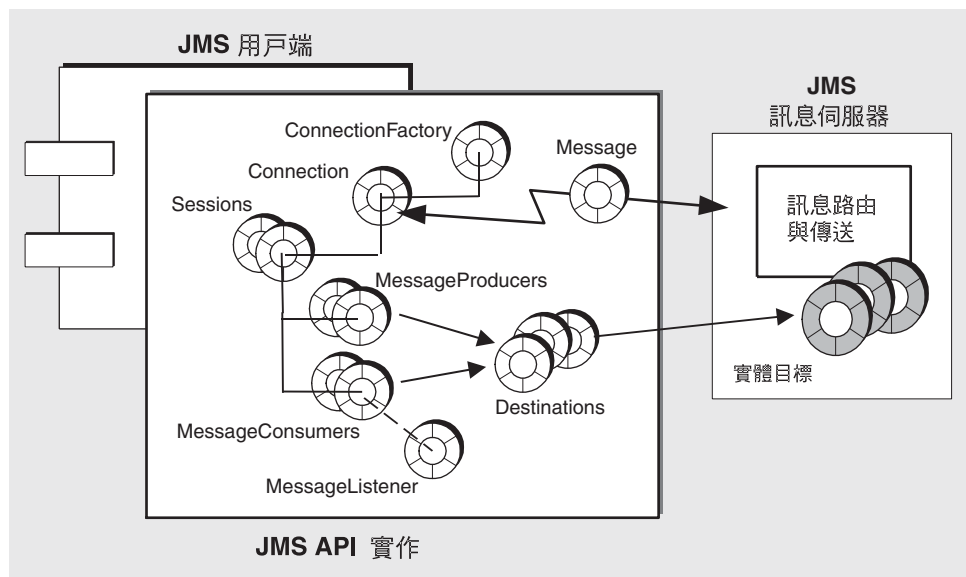
JMS 程式設計模型支援非同步訊息傳送服務的架構：JMS 用戶端經由 JMS 訊息服務交換訊息。JMS 提供者提供執行 JMS 訊息傳送所需要的物件；這些物件實作 JMS 應用程式設計介面 (API)。

本節說明 JMS 訊息傳送所需的程式設計物件並介紹用來傳送和接收訊息的傳送模型 (點對點與出版 / 訂閱)。

程式設計物件

設定 JMS 用戶端傳送訊息所需的物件，如圖 1-3 中所示。

圖 1-3 JMS 程式設計物件



在 JMS 程式設計模型中，JMS 用戶端使用 **連線工廠** 物件 (ConnectionFactory) 建立一個 **連線**，將訊息傳送至 JMS 訊息服務以及從 JMS 訊息服務接收訊息均透過此連線進行。連線物件 (Connection) 代表用戶端與訊息伺服器的使用中連線。

通訊資源的分配以及用戶端的認證均在建立連線時進行。相對而言，這是一個十分重要的物件，大多數用戶端均使用單一連線來進行所有的訊息傳送。

連線用於建立 **階段作業** 物件 (Session)。階段作業是用於生產和使用訊息的單執行緒環境。它用於建立訊息以及傳送和接收訊息的訊息產生者和用戶，並為所傳送的訊息定義傳送順序。階段作業透過大量的 **確認** 選項或作業事件，支援可靠的傳送。

用戶端使用訊息產生者物件 (MessageProducer) 將訊息傳送至指定實體 **目標**，在 API 中表示為目標物件。訊息產生者可指定預設訊息標頭值，例如傳送模式 (永久性與非永久性)、優先順序和存在時間值，以控制由產生者傳送至實體目標的所有訊息。

同樣，用戶端使用訊息用戶物件 (MessageConsumer) 從指定實體目標 (在 API 中表示為目標物件) 接收訊息。有兩種類型的目標，**佇列** 和 **主題**，取決於訊息傳送模型。

訊息用戶可以使用訊息選擇器，該選擇器可令訊息服務僅將符合選取條件的訊息傳送至訊息用戶。

訊息用戶可以支援同步或非同步的訊息使用。

- 同步使用表示用戶明確要求傳送訊息並隨後使用訊息。
- 非同步使用表示自動傳送訊息到已為用戶註冊的訊息偵聽物件 (MessageListener)。當階段作業執行緒呼叫訊息偵聽物件的 onMessage () 方法時，用戶端即可使用訊息。

程式設計緘域：訊息傳送模型

JMS 支援兩種完全不同的訊息 **傳送模型**：點對點與出版 / 訂閱。

點對點 (佇列目標) 將訊息從產生者傳送至單一用戶。在此傳送模型中，目標類型為 **佇列**。首先將訊息傳送至佇列目標，然後將訊息從該佇列傳送至已向佇列註冊的一個用戶，每次傳送一封訊息。可以向佇列目標傳送訊息的產生者沒有限制，但每封訊息只能傳送至一個用戶，並由一個用戶成功接收。如果沒有向佇列目標註冊的用戶，佇列將保留其接收到的訊息，並在用戶向該佇列註冊時將訊息傳送給他們。

出版 / 訂閱 (主題目標) 將訊息從產生者傳送至任意數目的用戶。在此傳送模型中，目標類型為 **主題**。訊息被首先傳送至主題目標，然後傳送至 **已訂閱** 該主題的 **所有** 使用中用戶。可以向主題目標傳送訊息的產生者沒有限制，並且每封訊息可以被傳送至任意數目的訂閱用戶。

主題目標也支援長期訂閱。長期訂閱表示用戶已向主題目標進行註冊，但在訊息到達目標時此用戶可以處於非作用狀態。當用戶再次處於使用中時，將接收此訊息。如果沒有用戶向主題目標進行註冊，則主題將只保留非使用中用戶長期訂閱的訊息。

這兩種訊息傳送模型使用表示不同程式設計網域的三組 API 物件（語義稍有差異）進行處理，如表 1-2 中所示。

表 1-2 JMS 程式設計網域及物件

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

* 取決於程式設計的方法，您必須指定特定的目標類型。

統一網域已在 JMS 版本 1.1 中說明過。如果您需要符合較早的 JMS 1.02b 規格，您可以使用網域特定的 API。使用網域特定的 API 也提供清理程式設計介面的優點，可以避免特定類型的程式設計錯誤：例如，為佇列目標建立長期訂閱。然而，網域特定 API 的缺點是您不能在同一作業事件或同一作業階段中結合點對點和出版 / 訂閱作業。如果您需要執行這些動作，您應該選擇統一網域的 API。

Message Queue 產品包含的範例應用程式以及 Message Queue 文件中提供的程式碼範例，均使用單獨程式設計網域。

可靠的訊息傳送

訊息的傳送模式可設成永久性或非永久性；此模式控制訊息傳送的可靠性。

- 永久性訊息保證僅被傳送一次並成功使用一次。永久性訊息在訊息服務故障的情況下不會遺失。對於這些訊息，可靠性是首要因素。
- 非永久性訊息保證最多被傳送一次。非永久性訊息在訊息服務故障的情況下會遺失。對於這些訊息，可靠性並非主要因素。

對於永久性訊息，確保可靠性有兩個方面。一個是要確保，透過使用確認和作業事件，成功的產生和使用訊息。另一個是要確保將訊息放進永久性儲存，讓訊息服務在將永久性訊息傳送至用戶之前不會遺失訊息。

以下各節描述確保可靠性的兩個方面。

確認 / 作業事件

可靠的訊息傳送取決於確保從訊息產生者至訊息伺服器上的實體目標和來自實體目標至訊息用戶的永久性訊息的成功傳送。可使用 JMS 階段作業支援的兩個一般機制中的任何一個來達到可靠性的目標，這兩個機制是：*確認*或*作業事件*。作業事件可以是本地作業事件或分散式作業事件（由分散式作業事件管理程式控制）。

確認

確認是用來確保傳送在用戶端和訊息服務間的訊息之可靠性。

在訊息產生的情況下，訊息服務確認收到傳送的訊息、放置訊息至它的目標以及永久儲存訊息。產生者的 `send()` 方法暫停運作直到傳回確認為止。

在訊息使用的情况下，訊息服務從目標刪除訊息之前，用戶端確認收到從目標傳送的訊息並加以使用。JMS 指定不同的確認模式來代表不同程度的可靠性。在這些模式中的某部份，用戶端暫停運作以等待訊息伺服器確認已刪除訊息，並因此而無法重新傳送訊息。

本地作業事件

也可以將階段作業配置為*作業事件*，這樣就可以將一個或多個訊息的生產和 / 或使用群組為不可分割的單元，即*作業事件*。JMS API 提供了啟動、確定或回復作業事件的方法。

在作業事件中生產或使用訊息時，訊息服務會追蹤各種傳送和接收過程，並在 JMS 用戶端發出確定作業事件的呼叫時完成這些作業。如果作業事件中特定的傳送或接收作業失敗，則會出現異常。用戶端程式碼可以透過忽略異常、重試作業或回復整個作業事件來處理異常。在作業事件確定後，將完成所有的作業。在作業事件回復後，將取消所有成功的作業。

本地作業事件的範圍始終是單一階段作業。即，可以將單一階段作業的環境中執行的一個或多個產生者或用戶作業群組為一個本地作業事件。

由於作業事件僅能是單一階段作業，因此不能有同時包含訊息生產和訊息使用的端對端作業事件。（換言之，至目標的訊息傳送和後續的至用戶端的傳送不能放在同一個作業事件中。）

分散式作業事件

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

支援分散式作業事件表示，訊息傳送用戶端可以透過 JTA 定義的 XAResource 介面參與分散式作業事件。此介面定義了實作兩階段確定的許多方法。當用戶端進行 API 呼叫時，JMS 訊息服務僅與分散式作業事件管理程式（由 Java Transaction Service (JTS) 提供）協作，來追蹤分散式作業事件中的各種傳送和接收作業以及作業事件狀態，並完成訊息傳送作業。

處理本地作業事件時，用戶端可以透過忽略異常、重試作業或回復整個分散式作業事件來處理異常。

永久性儲存

可靠性的另一個方面是，確保訊息服務在將永久性訊息傳送至用戶之前不會遺失訊息。這表示當永久性訊息到達它的實體目標時，訊息伺服器必須將它放在永久的資料儲存。如果訊息伺服器由於某種原因失敗，它可以回復該訊息並將其傳送至相應的用戶。

訊息伺服器還必須永久儲存長期訂閱。否則如果訊息伺服器發生故障，它將無法將訊息傳送至在訊息抵達主題目標後成為使用中狀態的長期訂閱者。

需要保證訊息傳送的訊息應用程式必須將訊息指定為永久性訊息，並傳送至主題目標的長期訂閱或佇列目標。

JMS 受管理物件

在 JMS 程式設計模型中使用的兩個物件，連線工廠和目標，會因提供者實作的 JMS 規格而有所不同。

- *連線工廠物件*用來建立連線，其運作方式取決於提供者用於傳送訊息的協定和機制。
- *目標物件*用於指定代理程式上實體目標的名稱，且取決於訊息伺服器上的特定命名慣例和實體目標功能。

爲了讓提供者在定義這些物件上有最大靈活性並讓用戶端可移植，JMS 規格定義了 [受管理物件](#) (針對連線工廠和目標) 可封裝提供者特定的資訊。這些物件由管理員建立和配置，儲存在 JNDI 名稱空間中 (物件儲存)，並由用戶端透過標準的 JNDI 查找程式碼存取。

受管理物件可讓 JMS 用戶端使用邏輯名稱查找和參照提供者特定的物件。這樣，用戶端程式碼無需瞭解提供者使用的特定命名語法、尋址語法或配置特性。從而使程式碼獨立於提供者之外。

[第 37 頁](#) 「[受管理物件](#)」一節提供關於用在 Message Queue 中的管理物件的額外資訊。

備註 JMS 規格並不要求您使用 JNDI 查找來存取管理物件。用戶端程式碼可創設連線工廠和目標物件，並設定它們的屬性值。然而，這表示用戶端程式碼無法移植至其他提供者。

Message Queue 簡介

Message Queue 是一個可靠的非同步訊息傳送服務，它符合 JMS 1.1 規格。此外，為滿足大型企業部署所需，Message Queue 還提供許多 JMS 規格要求以外的功能。

本章說明 Message Queue 服務架構，並介紹其企業功能。本章包含下列主題：

- [第 32 頁「訊息服務架構」](#)
- [第 40 頁「產品功能」](#)
- [第 46 頁「產品版本」](#)
- [第 48 頁「Sun 產品環境中的 Message Queue」](#)

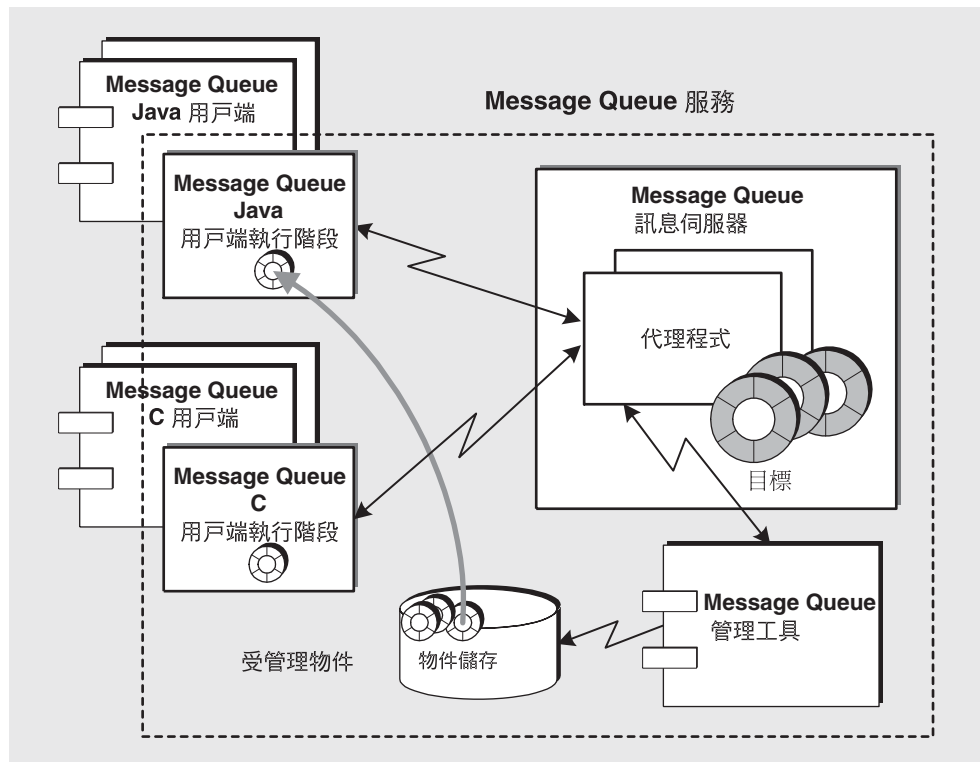
訊息服務架構

Message Queue 服務包含下列元素：

- 第 33 頁 「訊息伺服器」
- 第 33 頁 「用戶端執行階段」
- 第 37 頁 「受管理物件」
- 第 39 頁 「管理工具」

圖 2-1 顯示這些元件如何一起運作。

圖 2-1 Message Queue 服務架構



如此圖形所述，訊息佇列用戶端使用 Java 或 C API 來傳送或接收訊息。這些 API 是實作在 Java 或 C 用戶端執行階段程式庫中，它會執行建立代理程式連線與依據連線服務要求封裝位元的實際工作。若應用程式使用受管理物件，用戶端執行階段會在物件儲存中尋找這些物件，並使用它們來配置連線與尋找實體目標。代理程式會路由與傳送訊息。管理員會使用訊息佇列管理工具來管理代理程式，並新增受管理物件到物件儲存。

下列各節簡述每一個元素。

訊息伺服器

訊息伺服器是由一或多個代理程式所組成，它會執行訊息路由與傳送服務。這是 Message Queue 服務的核心。

訊息伺服器是由單一代理程式或一組協同運作的代理程式所組成（作為代理程式叢集），以執行訊息路由與傳送服務。代理程式是一個執行下列工作的程序：

- 認證使用者，以及針對使用者要執行的作業進行授權
- 設定與用戶端之間的通訊通道
- 從產生用戶端接收訊息，並將訊息放置到其各自的實體目標
- 路由與傳送訊息到一或多個使用用戶端
- 確保可靠的傳送
- 提供資料以便監視系統效能。

如需訊息伺服器、其內部元件與其功能的詳細資訊，請參閱第 59 頁第 4 章「訊息伺服器」。

Message Queue 企業版支援使用代理程式叢集，代理程式叢集是由多個互連的代理程式實例所組成，允許訊息伺服器根據訊息流量而調整。如需架構說明與叢集配置問題的詳細資訊，請參閱第 5 章「代理程式叢集」。

用戶端執行階段

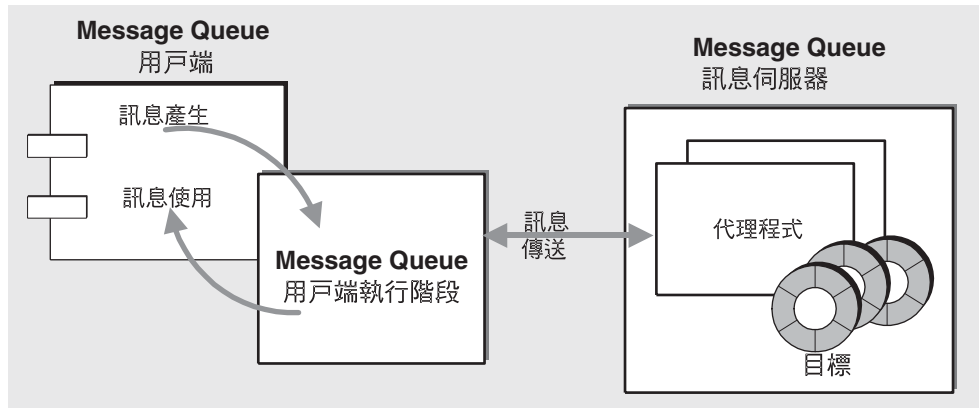
Message Queue 用戶端執行階段會提供用戶端應用程式到 Message Queue 服務的介面。用戶端執行階段支援 Message Queue 用戶端產生訊息（將它們傳送到目標）和使用訊息（從目標接收）所需的所有作業。

Message Queue 用戶端執行階段有兩個語言實作，如第 32 頁圖 2-1 中所示：

- **Java 用戶端執行階段。**提供 Java 用戶端應用程式與元件，以及實作 JMS API 並與 Message Queue 訊息伺服器互動所需的所有物件。這些介面物件包括連線、階段作業、訊息、訊息產生者與訊息用戶。
- **C 用戶端執行階段。**提供 C 用戶端應用程式與元件，以及與 Message Queue 伺服器互動時所需的 C 程式設計介面。(C 用戶端執行階段支援 JMS API 訊息傳送模組的程序版本。)

圖 2-2 說明用戶端執行階段在 Message Queue 用戶端與訊息伺服器間扮演的中間角色。訊息產生與使用牽涉到用戶端與用戶端執行階段之間的互動，而訊息傳送是用戶端執行階段與訊息伺服器之間的互動。

圖 2-2 用戶端執行階段與訊息傳送作業



用戶端執行階段會執行下列功能：

- 管理傳送至訊息伺服器的訊息
- 設定連線
- 建立用戶端的識別
- 實作用戶端確認
- 控制訊息在連線間的流量
- 可以覆寫產生用戶端設定的訊息標頭值。

下列各節簡述用戶端執行階段功能。用戶端執行階段某些方面的運作方式可以透過配置連線工廠物件的特性來自訂。

連線處理

若要配置連線處理運作方式，您必須指定用戶端要連線至代理程式的主機名稱與連接埠，以及想要的連線服務類型。若要連線到叢集中的代理程式，您必須指定連線的地址清單。若其中一個代理程式未上線，用戶端執行階段可以連線到叢集中的另一個代理程式。

在企業版中，若連線失敗，用戶端執行階段會自動重新連線到代理程式。重新連線有可能連線到相同的代理程式，或者連線到與原始連線不同的代理程式（若用戶端是連線到叢集中的代理程式）。

若代理程式實例未使用共用的可用永久性儲存（可以透過整合 Message Queue 與 Sun Cluster 來完成），若重新連線到不同的代理程式實例，則由失敗（或中斷連線）的代理程式所保留的永久性訊息與其他狀態資訊可能會遺失。也就是說，重新連線提供連線防故障備用，但不提供資料可用性。

用戶端標識

若應用程式認為有幫助，可以在任何連線上設定用戶端 ID，必須設定用戶端 ID 才能識別長期訂閱者。

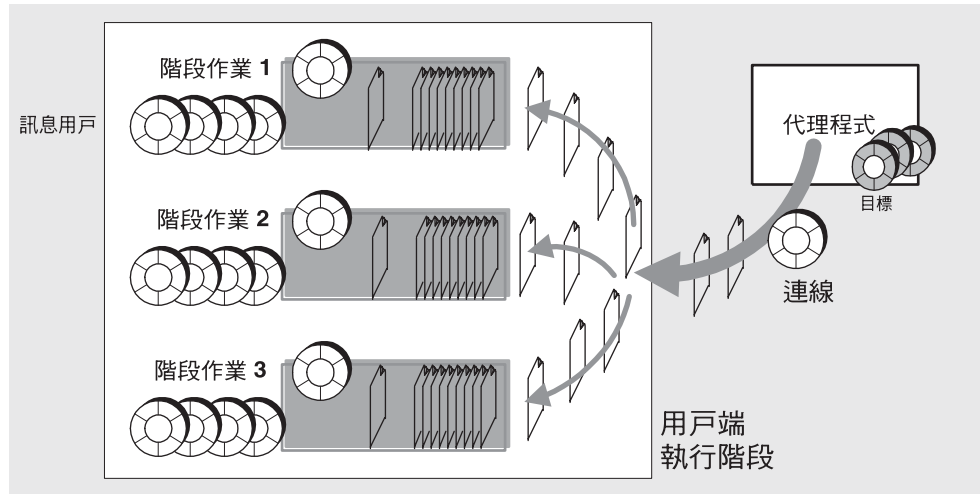
為追蹤長期訂閱，代理程式會使用唯一的用戶端標識。在訊息傳送至主題目標時，用戶端 ID 可用於識別長期訂閱者。代理程式會保留訊息的傳送目標資訊，以便在該訂閱者可用時傳送訊息。

因此，在已部署的應用程式中使用長期訂閱時，應該設定用戶端識別碼。Message Queue 功能能讓您在指定用戶端 ID 時使用特殊的變數名稱語法。這樣不論物件是由管理員建立或由程式碼建立，都能為每個從連線工廠物件取得的連線取得不同的用戶端 ID。如需更多資訊，請參閱「Message Queue 管理指南」。

訊息分散至用戶

代理程式透過連線傳送的訊息會由用戶端執行階段接收，並分散到適當的 Message Queue 階段作業，其中訊息會排入佇列以便由對應的訊息用戶使用，如第 36 頁圖 2-3 中所示。

圖 2-3 訊息傳送至 Message Queue 用戶端執行階段



訊息從每個階段作業佇列取回，每次一個，並同步使用 (透過呼叫 `receive()` 方法的用戶端執行緒) 或非同步使用 (透過呼叫訊息偵聽程式物件的 `onMessage()` 方法的階段作業執行緒)。(階段作業為單一執行緒的。)

傳送到用戶端執行階段的訊息流量是以每個用戶層級來計算。透過適當地調整連線工廠特性，您可以平衡訊息流量，使得傳送到一個階段作業的訊息不會影響同一時間傳送至其他階段作業的訊息。

確保可靠的訊息傳送

用戶端執行階段在確保可靠的訊息傳送時扮演重要的角色。它支援 JMS 規格的用戶端確認與作業事件模式，且控制用於保證可靠傳送的各種代理程式確認運作方式。

JMS 規格說明許多提供不同可靠性等級的用戶端確認模式。這些確認模式以及由 Message Queue 實作的附加模式，都可以在訊息使用環境中找到 (請參閱第 54 頁「用戶端確認」)。

在永久性訊息與可靠傳送的案例中，代理程式通常會在完成作業時通知用戶端執行階段，以便確保訊息的使用 (只確認一次)。您可以使用連線工廠特性來抑制此代理程式確認，因而節省網路頻寬與處理時間。當然，抑制代理程式確認，會降低對可靠傳送的保證。

訊息流量控制

用戶端執行階段是跨連線訊息流量的守門員。除了跨連線的正常 JMS 有效負載訊息之外，Message Queue 也會傳送各種控制訊息，這些控制訊息用於保證可靠的傳送、管理跨連線的訊息流量，並執行其他控制功能。

因為有效負載訊息與控制訊息會爭用相同的連線，它們可能會有所衝突，而導致暫停運作。用戶端執行階段會強制各種可配置的流量限制與計算方案，以降低有效負載訊息與控制訊息衝突的情況，因而最大化訊息流量。

覆寫訊息標頭值

用戶端執行階段可以覆寫指定訊息的永久性、使用期限和優先權的 JMS 訊息標頭欄位。

Message Queue 允許在連線層級覆寫訊息標頭：覆寫適用於所有在給定連線環境中產生的訊息。

用戶端執行階段覆寫訊息標頭值的功能讓 Message Queue 管理員可以更加控制訊息伺服器的資源。然而，覆寫這些欄位有干擾應用程式特定需求（例如，訊息永久性）的風險。因此，只有在諮詢適當的應用程式使用者或設計者後，才應使用此功能。

其他功能

用戶端執行階段會執行一些其他相關功能：

- **佇列瀏覽特徵。**您可以配置用戶端執行階段以設定一次會擷取的訊息數目，以及等候訊息的時間（當瀏覽佇列目標的內容時）。
- **訊息壓縮。**Java 用戶端執行階段可以在訊息產生時壓縮訊息，並在訊息使用時解壓縮。是否會進行壓縮與解壓縮，視用戶端建立訊息時是否在訊息標頭中設定 Message Queue 特定的訊息特性而定。

受管理物件

受管理物件會封裝提供者特定的實作與連線 / 目標相關的配置資訊。受管理物件可透過程式碼建立，也可以使用管理員工具來建立與配置，受管理物件是儲存在物件儲存中，用戶端應用程式可透過標準 JNDI 查找代碼來存取。

Message Queue 提供下表所示的受管理物件類型。

表 2-1 Message Queue 受管理物件類型

類型	說明
目標	代表代理程式中的實體目標。包含代理程式中提供者特定的實體目標名稱。訊息用戶和 / 或訊息產生者物件會使用目標受管理物件來存取對應的實體目標。
連線工廠	建立用戶端應用程式與 Message Queue 訊息伺服器之間的實體連線。同時也會配置 Message Queue 用戶端執行階段，它控制實體連線的運作方式。設定連線工廠受管理物件的屬性值時，您會指定套用至其所建立之物件的所有連線的特性。
XA 連線工廠	用於建立支援分散式作業事件的實體連線（請參閱第 28 頁「分散式作業事件」）。XA 連線工廠物件會與一般連線工廠物件共用一組屬性，但會啟用支援分散式作業事件所需的額外機制。
SOAP 端點	識別 SOAP 訊息的最終目標：這是可以接收 SOAP 訊息之 Servlet 的 URL。您可以配置 SOAP 端點受管理物件以指定多重 URL。也可以指定與物件關聯的查找名稱，以及物件儲存屬性。

透過 JNDI 使用受管理物件

雖然 JMS 規格並不需要 JMS 用戶端查找 JNDI 名稱空間中的受管理物件，這樣做有明顯的好處。它允許單一控制來源、允許配置與重新配置連線（用戶端執行階段運作方式）而不需要重新撰寫程式碼，它也允許您將用戶端移植到其他 JMS 提供者。

受管理物件讓您可以輕鬆地控制與管理 Message Queue 服務：

- 透過要求用戶端應用程式存取預先配置的連線工廠物件，管理員可以指定用戶端執行階段的運作方式。
- 透過要求用戶端應用程式存取與現有實體目標相符的預先配置目標，管理員可以控制實體目標的激增。

換言之，使用受管理物件可以允許 Message Queue 管理員控制訊息服務配置詳細資訊，同時允許將用戶端應用程式設定為獨立於提供者之外。

使用受管理物件表示用戶端程式設計師不需要瞭解提供者特定的語法、物件命名慣例或提供者特定的配置特性。實際上，透過將受管理物件指定為唯讀，管理員可以確保用戶端應用程式無法變更管理員在首次建立受管理物件時設定的屬性值。

而用戶端應用程式可以在其上創設連線工廠與目標受管理物件，此做法會逐漸損壞受管理附件的基本用途。**Message Queue** 管理員需要控制應用程式所需的代理程式資源，並調校訊息傳送效能。此外，直接創設受管理物件可將用戶端應用程式設定為獨立於應用程式提供者之外。

儘管有這些爭議，應用程式通常會在開發環境創設受管理物件，其中管理控制並不是問題。

物件儲存

Message Queue 受管理物件是放在物件儲存中 (請參閱第 32 頁圖 2-1)，它們可由用戶端應用程式透過 JNDI 查找來存取。**Message Queue** 支援兩種類型的物件儲存：標準的 LDAP 目錄伺服器與檔案系統物件儲存。

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

檔案系統物件儲存 **Message Queue** 支援檔案系統物件儲存，不建議您將它用於生產系統，但此類型的物件儲存在開發環境中非常易於使用。您所做的是在您的本機檔案系統上建立目錄，而不是設定 LDAP 伺服器。但是，檔案系統物件儲存不可作為部署在多重電腦節點之用戶端的集中物件儲存，除非這些用戶端可以存取物件儲存所在的目錄。

管理工具

Message Queue 管理工具是由一組命令行公用程式以及一個圖形化使用者介面 (GUI) 管理主控台所組成。

命令行公用程式 **Message Queue** 提供一組命令行公用程式，可執行所有 **Message Queue** 管理工作，例如啟動與管理代理程式、建立與管理實體目標、管理受管理物件，以及執行其他更加專用的管理工作。所有命令行公用程式都共用常用格式、語法慣例與選項。如需使用命令行公用程式的詳細資訊，請參閱 **Message Queue** 管理指南。

管理主控台 主控台提供 **Message Queue** 命令行公用程式功能的子集。您可以使用管理主控台來管理代理程式、建立與管理實體目標，以及管理受管理物件。然而，您無法執行某些只有使用命令行公用程式才能執行的專用管理工作。例如，您無法使用管理主控台來啟動代理程式、建立代理程式叢集，或管理使用者儲存庫。這些工作只能使用 **Message Queue** 命令行公用程式來執行。

「Message Queue 管理指南」提供簡短的實用指導，可讓您熟悉管理主控台並說明如何使用它來完成基本的工作。

管理主控台與某些指令行公用程式允許您遠端管理代理程式與實體目標。

產品功能

Message Queue 服務與之前各節所述的架構完全實作 JMS 1.1 規格，可提供可靠、非同步、彈性的訊息傳送。如需 JMS 有關相容議題的文件，請參閱附錄 A 「選擇性 JMS 功能的 Message Queue 實作」。

然而，Message Queue 的功能遠超過 JMS 規格的規範。這些功能可讓 Message Queue 與由許多分散式元件組成的系統整合，可在全天候的關鍵任務作業中交換數以萬計的訊息。

下面所述 Message Queue 的企業功能分成數種類別：

- [第 40 頁](#) 「整合支援功能」
- [第 42 頁](#) 「安全性功能」
- [第 43 頁](#) 「延伸性功能」
- [第 44 頁](#) 「可用性功能」
- [第 45 頁](#) 「可管理性功能」
- [第 45 頁](#) 「彈性的伺服器配置功能」

整合支援功能

由於 Message Queue 在 Message Queue 服務中包含許多傳輸協定與 C 用戶端介面，而且支援 SOAP (XML) 訊息且具有可插入的 J2EE 資源介面，因此可讓您整合企業中的異質應用程式與元件。

多重傳輸支援

Message Queue 支援用戶端使用多種不同的傳輸協定 (包括 TCP 和 HTTP 以及使用安全連線)，與 Message Queue 訊息伺服器互動的功能。

HTTP 連線 HTTP 傳輸允許透過防火牆傳送訊息。Message Queue 使用在 Web Server 執行的 HTTP 通道來實作 HTTP 支援。用戶端產生的訊息會使用 HTTP 透過防火牆傳送到通道 Servlet。通道 Servlet 會從 HTTP 請求擷取訊息，並透過 TCP/IP 將訊息傳送到代理程式。相同地，Message Queue 支援使用 HTTPS 通道 Servlet 的安全 HTTP 連線。如需建立 HTTP 連線架構的詳細資訊，請參閱第 63 頁「HTTP/HTTPS 支援」。如需設定與配置 HTTP/HTTPS 連線的詳細資訊，請參閱「Message Queue 管理指南」。

安全連線 Message Queue 提供使用安全套接層 (SSL) 標準 (透過 TCP/IP 與 HTTP 傳輸) 來安全地傳輸訊息的功能。這些基於 SSL 的連線服務允許加密在用戶端與代理程式之間傳送的訊息。

SSL 支援是以自身簽名的伺服器憑證為基礎。Message Queue 提供公用程式，可產生自身簽名憑證中的私密金鑰 / 公開密鑰對。此憑證會傳送給任何要求連線到代理程式的用戶端，且用戶端會使用此憑證來設定與加密連線。如需建立自身簽名憑證以啟用基於 SSL 之連線服務的詳細資訊，請參閱「Message Queue 管理指南」。

C 用戶端介面

除了支援 Java 語言訊息傳送用戶端，Message Queue 還提供 C 語言介面給 Message Queue 服務。C API 可讓傳統 C 應用程式與 C++ 應用程式參與基於 JMS 的訊息傳送。然而，使用 Message Queue C API 的用戶端無法移植至其他 JMS 提供者。

支援大部分標準 JMS 功能的 C 用戶端執行階段都支援 Message Queue C API，除了下列所述之外：使用受管理物件；對映、串流或物件訊息內文類型；分散式作業事件；與佇列瀏覽器。C 用戶端執行階段也不支援大部分 Message Queue 的企業功能。

如需 C API 功能與它如何使用 C 資料類型與函式實作 JMS 程式設計模型的詳細資訊，請參閱「Message Queue Developer's Guide for C Clients」。

SOAP (XML) 訊息傳送支援

Message Queue 支援建立與傳送符合簡單物件存取協定 (SOAP) 規格的訊息。SOAP 允許在集中式與分散式環境中的各點之間交換結構化 XML 資料或 SOAP 訊息。SOAP 訊息是一種 XML 文件，它也可包含附件 (附件格式不一定必須是 XML)。

SOAP 訊息是以 XML 編碼，因此 SOAP 訊息是獨立於訊息平台之外。它們可用於從舊式系統存取資料，並在企業之間共用資料。XML 提供的資料整合功能也讓此技術成為網路運算 (例如 Web 服務) 的本質。防火牆可以識別 SOAP 封包，並依據 SOAP 訊息標頭中揭露的資訊來篩選訊息。

Message Queue 以 Attachments API for Java (SAAJ) 規格實作 SOAP。SAAJ 是一個應用程式設計介面，您可以實作此介面以支援 SOAP 訊息傳送的程式設計模型並提供 Java 物件，以用於建構、傳送、接收與檢查 SOAP 訊息。SAAJ 定義兩個套裝軟體：

- `javax.xml.soap`：您可以使用此套裝軟體中的物件來定義 SOAP 訊息的部份，並組合與分解 SOAP 訊息。您也可以使用此套裝軟體來傳送 SOAP 訊息，無需提供者的支援。
- `javax.xml.messaging`：您可以使用此套裝軟體中的物件來傳送使用提供者的 SOAP 訊息，並接收 SOAP 訊息。

Message Queue 提供公用程式可將 SOAP 訊息轉換為 JMS 訊息，反之亦然。這些公用程式允許 Servlet 接收 SOAP 訊息、將訊息轉換為 JMS 訊息、由 Message Queue 服務將訊息傳送給 JMS 用戶、將訊息轉換回 SOAP 訊息，並將訊息傳送至 SOAP 端點。換言之，Message Queue 支援在 SOAP 端點間以可靠與非同步的方式交換 SOAP 訊息，更簡單的說，就是將 SOAP 訊息出版至 Message Queue 訂閱者。

如需詳細資訊，請參閱「[Message Queue Developer's Guide for Java Clients](#)」。

J2EE 資源介面

Java 2 Platform Enterprise Edition (J2EE 平台) 是一種 Java 程式設計環境中分散式元件模型的規格。J2EE 平台的其中一個要求是分散式元件必須能透過可靠的非同步訊息交換管道，與另一個分散式元件互動。簡言之，J2EE 平台需要 JMS 支援。

J2EE 程式設計模型中使用訊息驅動 Bean (MDB)、Enterprise Java Bean (EJB) 元件的特殊類型 (可由 JMS 訊息使用) 來提供此支援。J2EE 相容的應用程式伺服器必須提供支援 JMS 訊息傳送的 MDB 容器。這可藉由插入 JMS 資源介面至應用程式伺服器來達成。Message Queue 提供此類資源介面。

藉由將 Message Queue 資源介面插入至應用程式伺服器，在應用程式環境中部署並執行的 J2EE 元件 (包括 MDB) 可以在這些元件間與外部 JMS 元件間交換 JMS 訊息。這提供分散式元件的強大整合功能。

如需 Message Queue 資源介面的詳細資訊，請參閱第 6 章「[Message Queue 和 J2EE](#)」。

安全性功能

保護儲存與傳送中訊息資料對於大部分企業應用程式而言都非常重要。Message Queue 提供許多等級的安全性，包括使用者認證、資源存取的控制，以及訊息加密。

認證 Message Queue 支援基於密碼的使用者認證。會使用儲存在文字檔案或 LDAP 使用者儲存庫中的密碼，來驗證連線到訊息伺服器的使用者。所有連線嘗試資訊 (使用者與主機電腦) 都會被記錄下來，作為追蹤用途。

授權 存取控制清單 (ACL) 針對代理程式連線與實體目標，提供可詳細配置的存取控制。同時支援使用者與群組存取控制。授權則是針對個別代理程式，每個代理程式可以擁有不同的存取控制檔。

加密 SSL 支援允許使用強式 SSL 實作來加密訊息伺服器與其用戶端 (不論透過 TCP/IP 或 HTTP 連線) 之間的訊息流量。

如需寫入使用者儲存庫、管理存取控制清單與設定 SSL 支援的詳細資訊，請參閱「Message Queue 管理指南」。

延伸性功能

Message Queue 允許您在使用者、用戶端連線或訊息負載增加時，延伸您的應用程式。

可延伸的連線功能

Message Queue 代理程式可以處理數以千計的運作中連線。依預設，每個連線會由一個專屬的代理程式執行緒處理。因為這樣會佔用執行緒 (即使是連線閒置時)，您可以配置連線服務，讓多個連線能共用相同的執行緒。此共用執行緒儲存區模型可以顯著地擴充代理程式可支援的連線數目。如需更多資訊，請參閱第 63 頁「[執行緒儲存區管理程式](#)」。

代理程式叢集

當連線數目與透過代理程式傳送之訊息的數目增加時，可透過新增額外的代理程式實例至 Message Queue 伺服器，以管理增加的負載。代理程式叢集可以平衡用戶端連線與透過多個代理程式實例傳送的訊息，讓訊息伺服器具有高度的可延伸性。代理程式實例可位於相同的主機或分散在網路中的不同主機。當企業需求成長時，叢集是提升訊息流量與擴充訊息傳送頻寬的理想方式。第 75 頁第 5 章「[代理程式叢集](#)」中簡述代理程式叢集，而「Message Queue 管理指南」中則有更完整的說明。

佇列傳送至多個用戶

根據 JMS 規格，佇列目標中的訊息只能傳送給單一用戶。Message Queue 則允許多重用用戶註冊佇列。接著，代理程式可以分散訊息到不同的註冊用戶，以平衡負載並允許系統延伸。

佇列傳送到多重用戶實作是使用可配置的負載平衡方式進行。使用此方法時，您可以指定使用中用戶的最大數目，以及發生失敗時用以取代使用中用戶的備份用戶最大數目。此外，負載平衡機制也會考慮用戶的目前處理能力與訊息處理率。

如需負載平衡佇列傳送的詳細資訊，請參閱第 53 頁「佇列傳送至多個用戶」。

可用性功能

Message Queue 提供一些可以降低服務當機時間的功能。這些功能包括失敗預防機制，以及允許與 Sun Cluster 整合以提供高度可用性的機制。

訊息服務穩定性

確保訊息服務可用性其中一個最有效率的方式是：提供高效能服務並降低失敗的發生率。Message Queue 提供相關機制，可避免記憶體過度負載或效能暫停運作。這些機制可用於訊息伺服器與用戶端執行階段。

訊息伺服器資源管理 因為訊息伺服器受限於記憶體與 CPU 資源，當訊息伺服器無法回應或不穩定時，可能會造成過度負載。當訊息產生速度遠超過使用速度時，通常會發生此情形。為避免此情形，您可以在個別實體目標層級或系統範圍層級配置代理程式，以預防記憶體過度執行的情況。如需更多資訊，請參閱第 66 頁「記憶體資源管理」。

用戶端執行階段訊息流量控制 此外，Message Queue 還提供機制可控制傳送至用戶端執行階段的訊息。您可以使用流量控制機制來最佳化傳送到用戶端執行階段的訊息，同時預防用戶端耗盡記憶體。如需更多資訊，請參閱第 37 頁「訊息流量控制」。

自動重新連線到訊息伺服器

Message Queue 提供自動重新連線功能：若訊息伺服器與用戶端之間的連線失敗，Message Queue 會維護用戶端狀態，並同時嘗試重新建立該連線。在大多數情況下，當重新建立連線後，通常會繼續產生與使用訊息。如需詳細資訊，請參閱「Message Queue 管理指南」。

透過 Sun Cluster 提供高度可用性

雖然 Message Queue 的代理程式叢集可提供延伸度極高的訊息伺服器，它目前並不支援叢集中一個代理程式實例到另一個代理程式實例的故障。然而，Message Queue 可與 Sun Cluster 軟體整合以提供具有高度可用性的訊息伺服器。使用針對 Message Queue 開發的 Sun Cluster 代理程式，Sun Cluster 可以確保代理程式失敗時不會遺失任何狀態資料，允許訊息伺服器立即復原以降低當機時間。

可管理性功能

Message Queue 提供一些功能，您可以利用這些功能來監視並管理訊息服務，以及調校訊息服務的效能。

強悍的管理工具

Message Queue 同時提供指令行與 GUI 工具，可用來管理 Message Queue 訊息伺服器，並管理目標、作業事件、長期訂閱與安全性（請參閱第 39 頁「管理工具」）。

Message Queue 也支援訊息伺服器的遠端監視與管理，並提供工具以管理 JMS 受管理物件、使用者儲存庫、插入式 JDBC 相容資料儲存與自身簽名的伺服器憑證。如需使用這些管理工具的詳細資訊，請參閱「Message Queue 管理指南」。

基於訊息監視的 API

Message Queue 提供簡單的基於 JMS 監視 API，您可以使用此 API 來建立自訂的監視應用程式。這些監視應用程式是一種會從特殊主題目標擷取度量訊息的用戶。度量訊息包括由 Message Queue 代理程式提供的監視資料（請參閱第 71 頁「度量訊息產生者（企業版）」）。

如需每個度量訊息類型中報告的度量數目詳細資訊，請參閱「Message Queue Developer's Guide for Java Clients」，以瞭解如何開發使用度量訊息的 Message Queue 用戶端。如需配置度量訊息產生的詳細資訊，請參閱「Message Queue 管理指南」。

可調校的效能

Message Queue 提供許多調校訊息伺服器與用戶端執行階段，以達到最佳化效能。您可以監視關鍵資源並調整記憶體使用情況、執行緒資源、訊息流量、連線服務、可靠性參數，以及其他會影響訊息流量與系統效能的元素。如需調校訊息服務效能的詳細資訊，請參閱「Message Queue 管理指南」。

彈性的伺服器配置功能

Message Queue 允許您選擇如何儲存永久性物件、使用者資訊與受管理物件。

可配置的永久性

為確保訊息傳送，Message Queue 會儲存訊息與其他永久性物件，直到訊息被使用。除了提供高效能的基於檔案的永久性儲存，Message Queue 也支援可配置的永久性。這能讓您將永久性訊息儲存在內嵌或外部 JDBC 相容的資料庫，例如 Oracle 8i。如需更多資訊，請參閱第 67 頁「持續性管理程式」。

LDAP 伺服器支援

Message Queue 為認證與授權所需的受管理物件與使用者資訊提供基於檔案的儲存體。不過，Message Queue 也支援使用 LDAP 伺服器來儲存受管理物件與使用者儲存庫。LDAP 伺服器提供儲存與擷取此類資訊的安全標準方式，建議用於生產系統。如需使用 LDAP 伺服器來儲存受管理物件與使用者儲存庫的資訊，請參閱「Message Queue 管理指南」。

產品功能

Message Queue 有兩個可用版本：企業版與平台版。兩個版本都支援完整的 JMS 規格實作，但每個版本都對應到不同的功能集與授權容量。下表比較兩個版本支援的功能集。如需這些功能的說明，請參閱第 40 頁「產品功能」。

表 2-2 功能比較：企業版與平台版

企業版	平台版
<i>進階的整合支援功能</i>	
HTTP 支援、TCP 支援	TCP 支援
基於安全套接層 (SSL) 標準的安全連線	基於安全套接層 (SSL) 標準的安全連線
C 語言 API、Java API	Java API (C API 只能搭配試用版授權使用。)
SOAP (XML) 訊息傳送支援	SOAP (XML) 訊息傳送支援
J2EE 資源介面	J2EE 資源介面
<i>安全性功能</i>	
從文字檔案或 LDAP 使用者儲存庫提供認證、使用存取控制檔案進行授權、提供 SSL 加密。	與企業版相同
<i>延伸性功能</i>	
可延伸的連線功能	固定的連線功能

表 2-2 功能比較：企業版與平台版 (續)

企業版	平台版
訊息伺服器可實作為代理程式叢集	單一代理程式訊息伺服器
佇列傳送到不限數量的訊息用戶 (每個佇列)	最多只能佇列傳送到三個訊息用戶 (每個佇列)
<i>可用性功能</i>	
透過記憶體資源管理與訊息流量控制提供訊息服務穩定性	與企業版相同
在叢集中用戶端連線至不同的代理程式故障，或自動重新連線到相同的代理程式。	不提供用戶端連線故障備用功能。允許自動重新連線到相同的代理程式。
透過 Sun Cluster 提供高度可用性	透過 Sun Cluster 提供高度可用性
<i>可管理性功能</i>	
強固的管理工具	強固的管理工具
除了管理工具與記錄功能外，還提供基於訊息的監視 API	只提供管理工具與記錄功能，不提供基於訊息的監視 API
可調校的效能	可調校的效能
<i>彈性的伺服器配置功能</i>	
可插入的永久性	可插入的永久性
LDAP 伺服器支援	LDAP 伺服器支援

平台版與企業版提供的授權容量如下表所示。

企業版

Message Queue 企業版允許您在企業生產環境中部署與執行訊息傳送應用程式。您也可以使用企業版來進行開發、載入測試訊息傳送應用程式與組件並對其進行除錯。企業版具有不限數目的長期授權 (基於使用的 CPU 數量)。授權未對多重代理程式訊息服務中的代理程式數目做出限制。

平台版

Message Queue 平台版未對訊息伺服器所支援的用戶端連線做出數目限制。它提供基本的授權或 90 天的試用授權：

- **基本授權**沒有時間限制。平台版提供基本授權，可在需求不高的生產環境中作為 JMS 提供者。此授權不包含企業版功能。
- **90 天的試用企業授權**包含基本授權中未包含的所有企業版功能。然而，軟體強制規定該授權只有 90 天的有效期，它適合於評估企業版中可用的功能。如需使用 90 天試用企業授權的詳細資訊，請參閱「Message Queue 管理指南」中所述的啓動選項。

平台版可以從 Sun 網站免費下載，且隨附於 Sun Java System Application Server 平台。您可以在「Message Queue Installation Guide」中找到將 Message Queue 從平台版升級為企業版的說明。

備註 對於所有版本的 Message Queue，產品的某部份 (Message Queue 用戶端執行階段) 可免費再分發於商業用途。產品中的其他部份則不能任意再分發。可免費再分發的部份允許授權者開發 Message Queue 用戶端應用程式 (可連線到 Message Queue 服務)，被授權者可出售給協力廠商而不需支付任何授權費用。而向您購買之協力廠商則必須購買 Message Queue 以存取 Message Queue 訊息伺服器，或連線到已安裝並執行 Message Queue 訊息伺服器的第三方。

Sun 產品環境中的 Message Queue

除了做為直接供應用程式使用的中介軟體之外，Message Queue 也可由其他中介軟體以及由 Sun 提供之伺服器與應用程式使用。為達成此目的，Solaris、Java Enterprise System 與 Sun Java System Application Server 中皆提供 Message Queue。

在 Application Server 中，Message Queue 可滿足 J2EE 平台必須具備 JMS 提供者的 JMS 需求。它會由應用程式 (由 Application Server 控管) 直接使用。如需更多資訊，請參閱第 81 頁第 6 章「Message Queue 和 J2EE」。

可靠訊息傳送

本章說明 Message Queue 服務如何提供可靠訊息傳送。它透過系統追蹤訊息的路徑，描述用來路由和傳送訊息至相應用戶的不同機制，並保證傳送訊息。

本章包含下列主題：

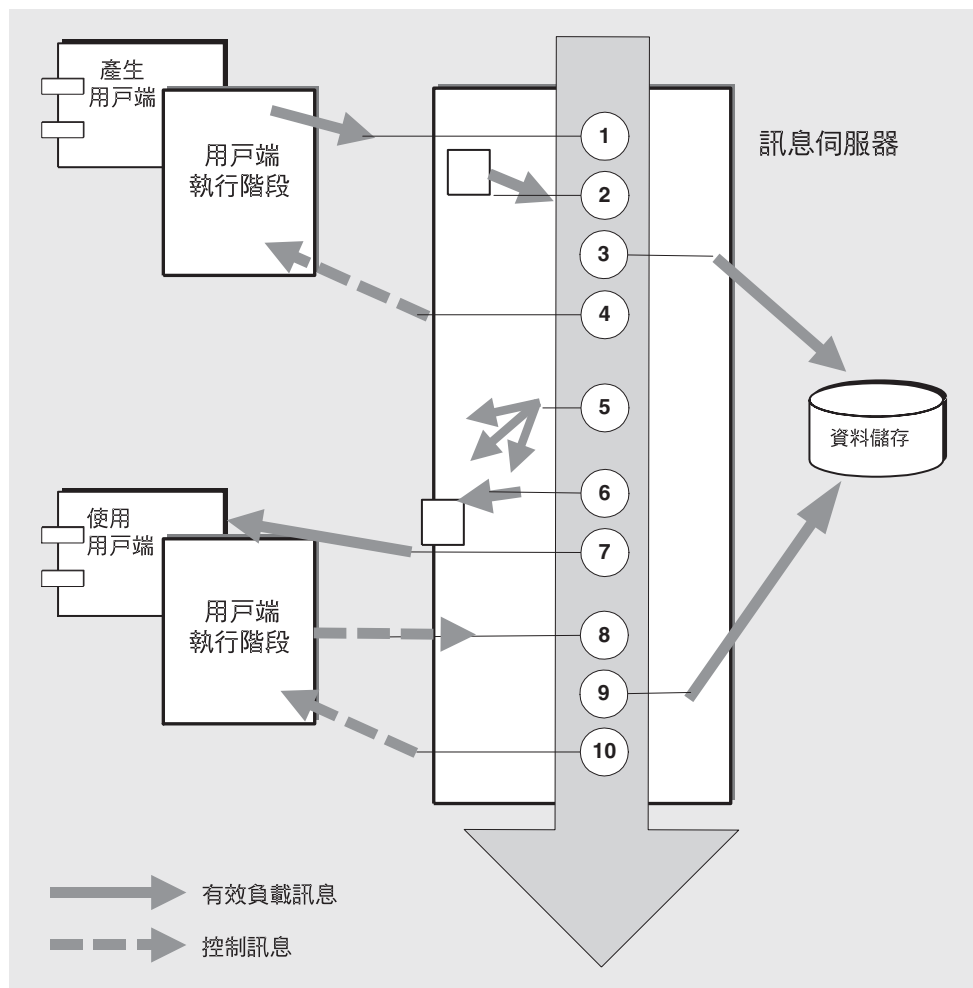
- [第 50 頁](#) 「訊息在系統中的行程」
- [第 51 頁](#) 「訊息傳送處理」
- [第 57 頁](#) 「效能問題」

本章內容同時適合開發者和管理員閱讀，並對第 2 章「[Message Queue 簡介](#)」的資訊加以補充。

訊息在系統中的行程

利用 Message Queue 訊息服務傳送訊息，從訊息生產者到訊息用戶的過程說明於圖 3-1。以下小節提供更多傳送過程中每個階段的詳細描述。

圖 3-1 訊息傳送步驟



一個永久性、可靠且已傳送之訊息的訊息傳送步驟如下：

訊息生產

1. 用戶端執行階段透過連線從訊息生產者傳送訊息至訊息伺服器。

訊息處理和訊息路由

2. 訊息伺服器從連線讀入訊息並將之放入相應的目標。
3. 訊息伺服器將 (永久性) 訊息放入資料儲存。
4. 訊息伺服器向訊息生產者的用戶端執行階段確認訊息接收。
5. 訊息伺服器會判斷訊息的路由。
6. 訊息伺服器由其目標將訊息寫至相應的連線。

訊息使用

7. 訊息用戶的用戶端執行階段從連線傳送訊息至訊息用戶。
8. 訊息用戶的用戶端執行階段確認訊息至訊息伺服器的使用。

訊息使用終了

9. 訊息伺服器處理用戶端確認，同時從目標和資料儲存刪除 (永久性) 訊息。
10. 訊息伺服器向用戶的用戶端執行階段確定用戶端確認已處理，且訊息無法再被傳送。

在系統處理訊息的這些傳送步驟中的作法分為兩類：

- **有效負載訊息**。JMS 訊息由生產型用戶端傳送至使用型用戶端。
- **控制訊息**。確認和其他非有效負載訊息通過訊息伺服器和用戶端執行階段之間，以確保有效負載訊息已被成功傳送並控制在連線上的訊息流量。

訊息傳送處理

利用 Message Queue 服務處理訊息，在訊息從生產者至用戶的傳送方法中，有幾個處理階段，如圖 3-1 中步驟的說明。

階段如下：

- [第 52 頁「訊息生產」](#)
- [第 52 頁「訊息處理和訊息路由」](#)
- [第 54 頁「訊息使用」](#)

- [第 56 頁](#) 「訊息使用終了」

這些階段在下面的小節中會有說明。

訊息生產

在訊息生產中，訊息由用戶端建立，且用戶端執行階段透過連線傳送至代理程式上的目標。

如果訊息的傳送模式被設定成永久性（保證傳送，一次並僅使用一次，即使代理程式故障），代理程式，依預設，會將一個控制訊息（一個代理程式確認）傳送回用戶端執行階段。代理程式確認指出代理程式已傳送訊息至目標，且將訊息存放在代理程式的資料儲存。用戶端執行緒會進行封鎖直到它收到代理程式確認。

如果訊息的傳送模式被設定成非永久性，則代理程式，依預設，不會將代理程式確認傳送回用戶端執行階段，且用戶端執行緒也不會封鎖。然而，如果要知道代理程式是否收到非永久訊息很重要，您可以啓用代理程式確認。事實上，在達到目標的記憶體限制時，必須啓用代理程式確認來減緩訊息生產（請參閱[第 66 頁](#)「目標訊息限制」）。

訊息處理和訊息路由

當代理程式收到一個進來的 JMS 有效負載訊息時，它會將訊息放在指定目標並隨後將訊息路由至對應的一個或多個用戶。

一般來說，所有訊息會保留在實體目標（記憶體）中，直至它們已被傳送或過期。如果代理程式出現故障，則這些訊息會遺失。如果是永久性訊息，代理程式會將它存放在資料庫或檔案系統中並在故障之後回復訊息。

訊息的處理取決於它的目標類型 - 佇列或主題 - 如以下小節中的說明。它也取決於當管理員建立實體目標時為目標設定的目標特性。

佇列目標

佇列目標用於點對點訊息傳送，其中訊息僅用來傳送給一位用戶並供其使用。

將佇列中的任何一個訊息僅傳送給單一用戶時，Message Queue 允許多個用戶註冊佇列。代理程式可隨後將訊息分散到不同的註冊用戶，以平衡負載。

基本的路由機制

當訊息從產生者到達時，它們會形成佇列。當訊息到達佇列前面時會被路由至註冊佇列的單一用戶。訊息到達佇列前面的順序，取決於它到達的順序及優先順序。

如果選擇器特性值被設到訊息中，代理程式會將它和註冊用戶指定的任何選擇器值作比較，並在路由訊息至用戶前確保符合選擇器值。

佇列傳送至多個用戶

實作到多個用戶的佇列傳送會根據數個佇列目標特性來使用一個可配置的負載平衡方法：

- 您可以設定在負載平衡的佇列傳送使用中的最大用戶數。
- 您可設定在使用中的用戶故障時，可取代他們的最大備份用戶數。

如果用戶數目超過這兩個特性的數目總和，將會拒絕新的用戶。(Message Queue 平台版在每個佇列最多支援 3 個用戶 - 兩個使用中和一個備份而 Message Queue 企業版則支援無限個用戶。)

負載平衡機制會考慮不同用戶的訊息使用率。在可配置大小的批次中，佇列目標中的訊息會路由到可用的新使用中用戶，以使用戶註冊佇列 (佇列目標的用戶流量限制特性)。一旦傳送這些訊息，其他到達佇列的訊息會在用戶為可用狀態時在批次中路由到用戶。當用戶使用先前傳送給他的訊息的可配置百分比後，用戶成為啟用狀態。換言之，每個用戶的派送比例根據用戶目前的容量和訊息處理比例而異。

如果一個使用中的用戶失敗，第一位備份用戶就會變成使用中，並取代失敗用戶的工作。因為這些機制，如果佇列目標有一個以上的使用中用戶時，則無法保證訊息使用的順序。

訊息產生率低時，代理程式可能會不規則地派送訊息給使用中用戶。如果使用用戶多於所需數目時，部分的用戶可能會收不到任何訊息。

在代理程式叢集環境中，您可以將傳送到多個用戶的功能，設定為先傳送給本地用戶。您可使用佇列目標特性來指定只有在產生者的本機代理程式 (即產生者向其傳送訊息的代理程式) 沒有用戶時，才會將訊息傳送給遠端用戶。在路由到遠端用戶 (透過用戶的本機代理程式) 可能會降低流量速率的情況下，此設定可讓您增加路由到遠端用戶的效能。

主題目標

主題目標用於出版 / 訂閱訊息傳送，其中訊息表示用來傳送至已在目標中註冊其興趣的所有用戶。

基本的路由機制

當一個訊息從產生者到達時，它會被路由至訂閱此主題的所有用戶。如果用戶已註冊此主題的長期訂閱，則他們不需在訊息傳送至此主題時成為使用中用戶：代理程式會儲存訊息直至用戶再次成為使用中用戶，然後傳送訊息。

如果選擇器特性值被設到訊息中，代理程式會將它和註冊用戶指定的任何選擇器值作比較，並在路由訊息至用戶前確保符合選擇器值。

長期訂閱和用戶端識別碼

一個用戶只能有一個主題的長期訂閱。當使用者開啓和關閉至訊息伺服器的連線時，使用者的識別必須維持不變。[用戶端識別碼](#)就是用來確保每個長期訂閱都只對應到一個使用者。

用戶端識別碼將用戶端至訊息伺服器的連線與代表用戶端的訊息伺服器所維護的狀態資訊相關聯。依定義，用戶端識別碼是唯一的。

若要建立長期訂閱，用戶端識別碼必須由用戶端使用 **JMS API** 方式呼叫來透過程式設定，或在用戶端使用的連線工廠物件中進行管理級別的配置。

訊息使用

一旦訊息被路由，它們會被分別傳送至它們的用戶。當用戶收到一個有效負載訊息後，此使用用戶端執行階段會傳送回它已接收到並已處理此訊息的確認給代理程式。代理程式在從它的目標刪除訊息之前，會等待這項用戶端確認。用戶端確認可以套用到個別訊息、群組訊息或作業事件。

用戶端確認

根據 **JMS** 規格，用戶端在建立階段作業時可指定三個基本確認模式中的一個。選擇何種模式取決於您想要的訊息傳送可靠性：

Message Queue 利用額外的 **NO_ACKNOWLEDGE** 模式，擴充用戶端確認模式的設定。基本和擴充模式如下面小節中的說明。

AUTO_ACKNOWLEDGE 模式

在 **AUTO_ACKNOWLEDGE** 模式中，階段作業自動確認用戶端使用的每個訊息。此外，階段作業執行緒會封鎖，以等待代理程式確定已處理每個使用訊息用戶端的確認。這個確定就稱為代理程式確認。

- 針對訊息偵聽程式使用的非同步訊息，訊息會在 `onmessage()` 方法返回後確認。
- 針對同步訊息使用，訊息會在 `receive()` 方法返回之前確認。在此種情況下，如果系統在訊息被使用之前發生故障，已部分處理的訊息仍有可能遺失。爲了增加可靠性，使用 **CLIENT_ACKNOWLEDGE** 模式或作業事件作業階段以保證如果系統發生故障，訊息不會遺失。

CLIENT_ACKNOWLEDGE 模式

CLIENT_ACKNOWLEDGE 模式給予用戶端最大的控制權。在這個模式中，用戶端會在使用一個或多個訊息之後明確確認。用戶端呼叫訊息物件的 `acknowledge()` 方法，致使階段作業確認自上次呼叫該方法以來它所使用的所有訊息。（這可能包含在階段作業中由不同訊息偵聽程式非同步使用的訊息，並和它們的使用順序無關。）

此外，階段作業執行緒會封鎖，等待返回的代理程式確認訊息使用的批次，以確認代理程式已處理用戶端確認。

因為用戶端確認和代理程式確認通常會分為幾個批次（而不是一個個傳送），CLIENT_ACKNOWLEDGE 模式和 AUTO_ACKNOWLEDGE 模式相比，前者通常會保存連線頻寬並降低代理程式確認的耗用時間。當然，如果在這個模式，因為用戶端會確認每個訊息，所以將不會發生分批傳送，而確認就會一個個傳送。

備註 Message Queue 也提供一個指定模式讓您可以在 CLIENT_ACKNOWLEDGE 模式中使用，透過它您可以只確認呼叫方式上的個別訊息，而不使用標準運作方式。這可以利用在「[Message Queue Developer's Guide for Java Clients](#)」中說明的程式設計技術來達成。

DUPS_OK_ACKNOWLEDGE 模式

在 DUPS_OK_ACKNOWLEDGE 模式中，階段作業會在使用十個訊息後確認。這個值目前不能配置。和 AUTO_ACKNOWLEDGE 或 CLIENT_ACKNOWLEDGE 模式不同，階段作業執行緒不會封鎖等待代理程式確認，因為在 DUPS_OK_ACKNOWLEDGE 模式中不需要代理程式確認。

這表示不能保證訊息已被傳送且僅使用一次。一般來說，訊息將不會常常被重新傳送；它們僅會在故障發生時重新傳送，因為代理程式沒有收到用戶端確認訊息已送達。如果不在意重複傳送的話，用戶端應使用 DUPS_OK_ACKNOWLEDGE 模式。

因為用戶端確認是分批傳送，且用戶端執行緒不產生封鎖，訊息流量通常會較其他模式高許多。

NO_ACKNOWLEDGE 模式

在 NO_ACKNOWLEDGE 模式中，代理程式執行代表用戶端的用戶端確認，因此不保證訊息已被使用用戶端成功處理。

當訊息流量很重要而不重視可靠傳送時可使用此模式。這是可能會發生的狀況，例如在一段短時間間隔內訊息被定期傳送時，訊息負載會較高，而訊息的遺失沒有太大關係。

這個模式擴充了 JMS 規格且僅能由不需要與其他 JMS 提供者工作的用戶端使用。

作業事件

以上描述的用戶端與代理程式確認過程，還適用於群組至作業事件中的 JMS 訊息傳送。在這些情況下，用戶端與代理程式確認可對作業事件級別作業，包括在作業事件中所有涉及的訊息。確定作業事件時，代理程式確認會被自動傳送。

代理程式追蹤作業事件，並可在發生故障時對作業事件進行確定或回復。本作業事件管理還支援本機作業事件，此為較大型分散式作業事件（請參閱第 28 頁「[分散式作業事件](#)」）的一部分。代理程式會追蹤這些作業事件的狀態，直至它們被確定。代理程式啟動時，它會檢查所有未確定的作業事件，並依預設，會回復所有作業事件（那些處於 PREPARED 狀態必須手動解決的作業事件除外）。

Message Queue 透過 XA 連線工廠實作對分散式作業事件的支援，這可讓您建立 XA 連線，而 XA 連線又可讓您建立 XA 階段作業。此外，對分散式作業事件的支援還要求協力廠商 Java Transaction Service (JTS) 或與 J2EE 相容的 Application Server（提供 JTS）。

訊息使用終了

代理程式在成功傳送訊息後從目標記憶體刪除訊息。然而，有時訊息還是可能在還未成功傳送前被捨棄。下面的小節中說明在哪些情況下訊息會被捨棄。

訊息的正常刪除

在正常情況下，代理程式在訊息成功傳送並收到用戶端確認後才從目標記憶體刪除訊息。

當代理程式傳送一個訊息給用戶時，它會標示該訊息為已傳送，但不確知訊息是否已被接受並且被使用。因此，代理程式會在將訊息從實體目標和永久性儲存刪除前等待用戶端確認。

如果訊息是送至一個主題，代理程式不會刪除該訊息直到從每個接收訊息的訊息用戶處收到用戶端確認。如果用戶為主題的長期訂閱，則代理程式會將每個訊息保留在目標中，並當每個長期用戶成為使用中用戶時傳送此訊息。代理程式接收用戶端確認時會記錄這些確認，並僅在已接收所有確認後才刪除訊息（除非訊息在此之前已過期）。根據用戶端的確認模式，代理程式可能透過傳送一個代理程式確認回傳給用戶端來確認用戶端確認的接收。

如果代理程式或連線故障，代理程式可能收不到用戶端確認而重新傳送所有之前已傳送但未確認的訊息，並為它們標示一個 `Redelivered` 旗標。例如，如果佇列用戶在確認訊息的接收之前離線，且其他用戶（或甚至同一用戶）在之後註冊該佇列，代理程式將重新傳送該未確認訊息給新用戶，並為該訊息標示一個 `Redelivered` 旗標。訊息重新傳送的用戶端應用程式在此種情況下應該檢查訊息的旗標。

備註 透過 JMS API (回復階段作業)，用戶端能明確要求重新傳送已接收但用戶端未確認的訊息。重新傳送這些訊息時，代理程式會為它們標示 `Redelivered` 旗標。

訊息的不正常刪除

當訊息無法被傳送時，取決於造成訊息無法傳送的情況，會被捨棄或放置在停用訊息佇列上。

一個訊息在下列情況下可能在尚未被成功傳送和使用前遭到代理程式捨棄：

- 您使用管理工具來清除一個或多個訊息的目標
- 您移除或重新定義一個長期訂閱，在主題目標中留下不可能傳送的訊息。

然而，在這些情況下，訊息將被視為停用和捨棄或放置在停用訊息佇列上，取決於您配置的運作方式：

- 訊息到期，超過設定在訊息標頭中 `JMSExpiration` 的值。
- 一個目標的「移除」限制運作方式被呼叫是因為該目標超出記憶體限制閾值。
- 訊息傳送失敗是因為訊息不能被使用（由用戶端拋出一個異常）。

您可選擇保留這些訊息並將它們放置在停用訊息佇列中。在放置訊息至停用訊息佇列時，代理程式寫入指定的 `Message Queue` 特性值至訊息，指出放入訊息的時間和原因。

您可在之後從停用訊息佇列中擷取訊息供診斷目的使用。請參閱第 65 頁「停用的訊息佇列」，以獲得更多資訊。

效能問題

訊息傳送的可靠性越高，需要的耗用時間和頻寬就越多。因此，可靠性與效能之間的平衡是設計時要考慮的重點。您可以透過選擇生產和使用非永久性訊息來最大化效能。另一方面，您也可以透過生產和使用永久性訊息並使用作業事件階段作業來最大化可靠性。在這兩種極端之間有許多選擇，這取決於每個應用程式的需要。

例如，訊息處理速率就受到許多因素的影響，包括訊息應用程式的設計、訊息伺服器的配置和用戶端執行階段的配置。雖然這些因素相當不同，它們的互動卻能讓最大效能的工作複雜化。

本節簡要複習一些在可靠性與效能之間取得平衡時要計算的因素。

傳送模式 傳送模式指定訊息為最多只傳送一次（非永久性）或一次且僅有一次（永久性）。永久性訊息的管理需要使用代理程式確認連線上的訊息流量和使用用戶端確認模式來封鎖，以等待收到代理程式的確認。若要增加流量，您可設定用戶端執行階段以抑制代理程式確認，但這麼做會取消保證永久性訊息已被傳送一次且僅有一次。

用戶端確認模式 四個用戶端確認模式中的每一個都需要不同的處理級別和頻寬耗用時間。AUTO_ACKNOWLEDGE 模式使用最大耗用時間但根據訊息基礎保證可靠度，CLIENT_ACKNOWLEDGE 模式分批傳送確認，也因此只需要較少的頻寬耗用時間，而DUPS_OK_ACKNOWLEDGE 模式則使用最少耗用時間，但允許訊息的重複傳送。NO_ACKNOWLEDGE 模式有最佳的效能但可能要付出遺失訊息的代價。

用戶端應用程式設計 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。如果用戶端在產生或使用訊息中速度緩慢，通常您可以藉由重新設定應用程式，以分散較大階段作業數目上的訊息產生者和用戶，或者分散較大連線數目上的階段作業，並因此改善效能。影響效能的設計問題在「Message Queue Developer's Guide for Java Clients」和「Message Queue Developer's Guide for C Clients」中說明。

訊息流量計數 連線頻寬的控制訊息和有效負載訊息間的競爭，可以透過用戶端執行階段來管理。適當配置用戶端執行階段能有助於加速代理程式確認的傳送、釋放封鎖階段作業執行緒並加速訊息的使用。請參閱「Message Queue 管理指南」，以取得更多資訊。

訊息流量限制 訊息使用速度可能在達到用戶端執行階段資源限制時變慢。藉由限制保留在用戶端執行階段中等待被一個或多個用戶使用的訊息數目，可以避免這些資源限制。請參閱「Message Queue 管理指南」，以取得更多資訊。

訊息伺服器

第 33 頁「訊息伺服器」中所介紹的 Message Queue 訊息伺服器，是由單一代理程式或一組共同運作的代理程式（代理程式叢集）所組成，負責執行訊息路由與傳送作業。

本章討論代理程式的內部架構、描述其各種元件，並說明在開發與生產環境配置與管理代理程式所需的步驟。它由下列部份所組成：

- 第 60 頁「代理程式架構」
- 第 61 頁「代理程式元件」
- 第 72 頁「開發與生產環境」

瞭解代理程式的功能元件可協助您配置想要的代理程式運作方式、調整作業，以及最佳化效能。本章內容較適用於管理員，而非應用程式開發者。

代理程式架構

若要執行訊息傳送，代理程式必須設定與用戶端通訊的通道、執行認證與授權、適當路由訊息、確保可靠傳送，並為監視系統效能提供資料。

若要執行這套複雜的功能，代理程式將使用一些不同的內部元件，並且每個元件在傳送過程中都有特定的作用。圖 4-1 中說明代理程式元件，表 4-1 中則簡單描述這些元件。訊息路由器元件執行主要的訊息路由和傳送服務，其他元件提供訊息路由器依存的重要之支援服務。

圖 4-1 代理程式元件

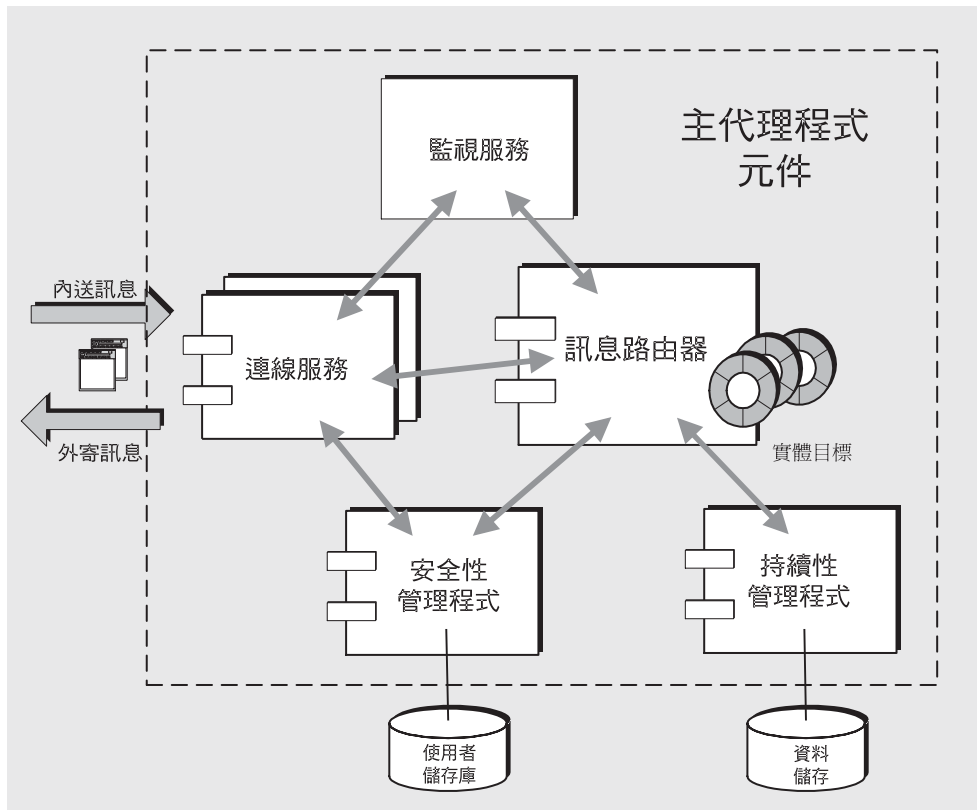


表 4-1 主代理程式服務元件和功能

元件	說明 / 功能
連線服務	管理代理程式與用戶端之間的實體連線，為進來和外送的訊息提供傳輸。

表 4-1 主代理程式服務元件和功能 (續)

元件	說明 / 功能
訊息路由器	管理訊息的路由與傳送：這些訊息包括 JMS 訊息以及 Message Queue 訊息傳送系統為支援 JMS 訊息傳送而使用的控制訊息。
持續性管理程式	管理向永久性儲存體的資料寫入，以便系統故障不會導致傳送 JMS 訊息的故障。
安全性管理程式	為請求連線至代理程式的使用者提供認證服務，為已認證的使用者提供授權服務 (存取控制)。
監視服務	產生可以寫入大量輸出通道的度量和診斷資訊，另外，您可以使用輸出通道來監視和管理代理程式。

根據載入條件、應用程式的複雜性等，您可以配置代理程式，以最佳化代理程式的效能。

代理程式元件

下列各節描述圖 4-1 中所列的各個代理程式元件、代理程式的功能，以及其運作方式。如需代理程式的個別特性與配置程序的詳細資訊，請參閱 Message Queue 管理指南。

連線服務

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

服務類型 指定此服務提供 JMS 訊息傳送 (NORMAL) 或提供支援管理工具的 Message Queue 管理服務 (ADMIN)。

協定類型 指定支援此服務的基本傳輸協定層。

目前，Message Queue 代理程式可提供的連線服務如表 4-2 中所示。

表 4-2 代理程式支援的連線服務

服務名稱	服務類型	協定類型
jms	NORMAL	TCP

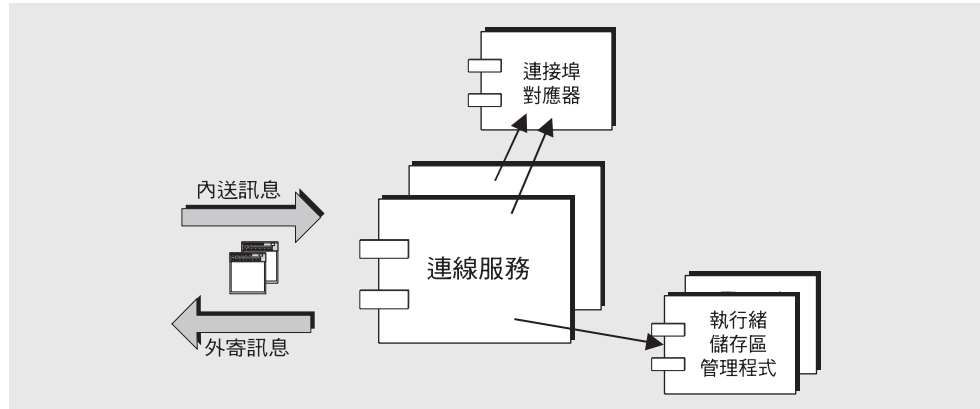
表 4-2 代理程式支援的連線服務 (續)

服務名稱	服務類型	協定類型
ssljms	NORMAL	TLS (基於 SSL 的安全性)
httpjms (企業版)	NORMAL	HTTP
httpsjms (企業版)	NORMAL	HTTPS (基於 SSL 的安全性)
admin	ADMIN	TCP
ssladmin	ADMIN	TLS (基於 SSL 的安全性)

您可以將代理程式配置為執行這些連線服務的任何一種或全部。每種連線服務可支援特定認證和授權 (存取控制) 功能 (請參閱第 68 頁「安全性管理程式」)。每種連線服務均為多重執行緒，支援多重連線。

每種連線服務可用於特定的連接埠，此連接埠由代理程式的主機名稱和連接埠號指定。可以動態配置連接埠，或您可以指定連線服務為可用時的連接埠。圖 4-2 中說明了一般方案。

圖 4-2 連線服務支援



連接埠對映器

共用 *連接埠對映器* 會為連線服務指定連接埠。連接埠對映器本身常駐在標準的連接埠號 7676 上。當 Message Queue 用戶端執行階段設定與代理程式連線時，它會先聯絡連接埠對映器，以請求其所需的連線服務的連接埠號。

配置連線服務時，您可以為 `jms`、`ssljms`、`admin` 與 `ssladmin` 連線服務指定靜態連接埠號，以覆寫連接埠對映器的設定。但是，靜態連接埠通常只用於特殊情況，例如透過防火牆連線（請參閱第 63 頁「[HTTP/HTTPS 支援](#)」），一般不建議使用。

執行緒儲存區管理程式

每種連線服務均為多重執行緒，支援多重連線。這些連線所需的執行緒，在執行緒儲存區管理程式所管理的執行緒儲存區中維護。您可以配置此執行緒儲存區管理程式，以設定此執行緒儲存區中所維護的執行緒的最小數目和最大數目。由於連線需要執行緒，因此可將執行緒新增至執行緒儲存區。當超過最小數目時，系統將關閉執行緒（因為這些執行緒將成為自由執行緒，直至達到最小的臨界值），從而節省記憶體資源。執行緒儲存區中的執行緒可專屬於單一連線，或依需要指定給多重連線。

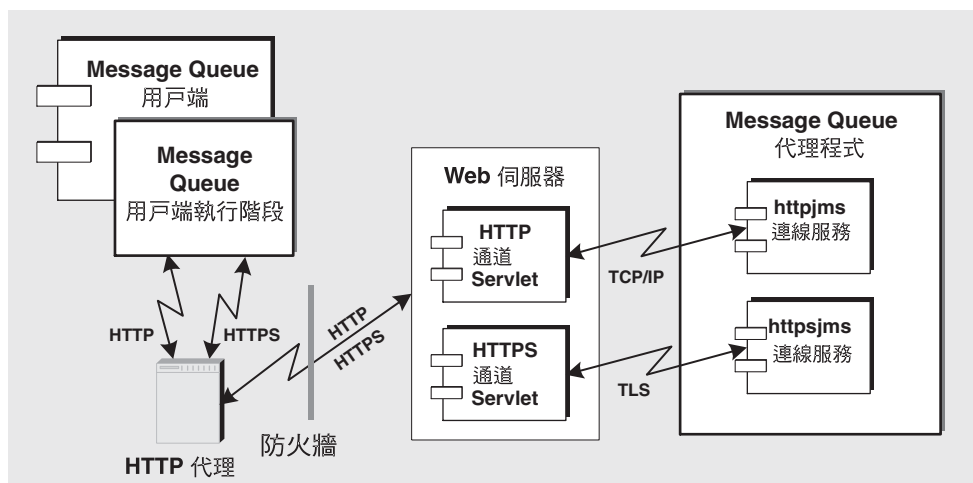
HTTP/HTTPS 支援

HTTP/HTTPS 支援可讓 Message Queue 用戶端使用 HTTP 協定與代理程式互動，而不需透過直接 TCP 連線。若用戶端必須與代理程式透過防火牆區隔，您可以使用 HTTP/HTTPS 服務，因為它允許透過防火牆進行通訊。

備註 HTTP/HTTPS 支援可用於 Java 用戶端，但不適用於 C 用戶端。

圖 4-3 顯示提供 HTTP/HTTPS 支援所需的主要元件。

圖 4-3 HTTP/HTTPS 支援架構



- 在 Message Queue 用戶端，HTTP 或 HTTPS 傳輸驅動程式將 JMS 訊息封裝在 HTTP 請求中，並確保這些請求以正確的序列傳送至 Web Server。
- 如有必要，用戶端可選擇使用 HTTP 代理伺服器與 Web Server 互動。
- HTTP 或 HTTPS 通道 Servlet (均隨附於 Message Queue) 被載入 Web Server 中，並用於在將 JMS 訊息轉寄至代理程式之前，將它們從用戶端 HTTP 請求中移除。HTTP/HTTPS 通道 Servlet 也會將代理程式回應傳回用戶端。單一 HTTP/HTTPS 通道 Servlet 可用於存取多重代理程式。
- 在代理程式端，當代理程式啟動時，會建立代理程式到通道 Servlet 的連線。當訊息從 HTTP 或 HTTPS 通道 Servlet 傳送時，會分別由 httpjms 或 httpsjms 連線服務展開訊息，並將它提交到代理程式的訊息路由器元件。

圖 4-3 中顯示 HTTP 與 HTTPS 架構非常類似。主要區別在於，對於 HTTPS (httpsjms 連線服務)，通道 Servlet 具有與用戶端和代理程式的安全連線。

Message Queue 的 HTTPS 通道 Servlet 會將自身簽名的憑證傳送至請求連線的任何代理程式。代理程式使用此憑證來設定與 HTTPS 通道 Servlet 的加密連線。一旦建立此連線，就可協商 Message Queue 用戶端與通道 Servlet 之間的安全連線。

使用「Message Queue 管理指南」中描述的特性，可配置 httpjms 和 httpsjms 服務。

訊息路由器

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

Message Queue 訊息傳送以兩個階段的訊息傳送為前提：首先，從生產型用戶端到代理程式上的實體目標。接著，從代理程式上的目標到一或多個使用用戶端。訊息路由器會負責將到達的訊息存放在適當的目標，然後路由並傳送該訊息到適當的用戶。

本節說明不同的目標類型，個別且整體的管理這些目標的記憶體資源。您可以在第 3 章「可靠訊息傳送」中找到路由與傳送訊息機制的說明。

實體目標

實體目標表示代理程式在實體記憶體中的位置，進來的訊息在路由至使用用戶端之前，會先被儲存。

根據建立方式與用途而定，目標分為數個種類：管理員建立的目標、自動建立的目標、暫存目標與停用的訊息佇列。

管理員建立的目標

管理員建立的目標是管理員使用 Message Queue 管理工具所建立的目標。這些目標對應到利用程式建立的邏輯目標，或對應到由目標管理的物件（供用戶端查找）。

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

自動建立的目標

自動建立的目標是由代理程式在需要時自動建立，而無需管理員的介入。要特別注意的是，當訊息用戶或訊息產生者嘗試存取不存在的目標時，會自動建立此類型的目標。這些目標用於需要動態建立目標的情況：一般是在開發與測試循環期間。您可以配置代理程式，以啟用或停用自動建立功能。

當自動建立目標時，可能導致不同的用戶端應用程式（使用相同的目標名稱）之間的衝突，或降低系統效能（歸因於支援目標所需的資源）。由於此原因，當不再使用自動建立的目標時，此目標將被代理程式自動銷毀；即，當此目標不再有訊息用戶用戶端且不再包含任何訊息時。若代理程式重新啟動，則只有在它們包含永久性訊息時，才會重新建立自動建立的目標。

暫存目標

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

停用的訊息佇列

停用的訊息佇列是代理程式啟動時自動建立的專用目標，用於儲存作為診斷目的之停用訊息。**停用的訊息**是由於某種原因從系統移除，而非因為正常處理或明確管理動作而移除。可能成為停用訊息的原因有：到期、因為超過記憶體限制而從目標移除或嘗試傳送失敗。

有兩種方式可將訊息放置在停用的訊息佇列：

- 您可以配置目標，將訊息放在停用的訊息佇列，而非捨棄訊息。
- 用戶端開發者也可以在建立訊息時設定特性值，以決定當訊息即將停用時，是否應該將它放置到停用的訊息佇列。

當訊息被放置到停用的訊息佇列時，會在該訊息中寫入額外的特性資訊，提供您訊息停用的相關資訊。

記憶體資源管理

訊息伺服器的資訊受到限制：記憶體、CPU 週期等。因此，視代理程式所支援之訊息傳送應用程式的使用式樣而定，當訊息伺服器無法回應或不穩定時，可能會造成過度負載。要特別注意的是，若針對目標產生訊息的速度快過於使用速度，就會發生問題。

訊息路由器有管理記憶體資源和預防代理程式耗盡記憶體的機制。它會使用三個記憶體保護級別，以在資源不足時維持系統作業：目標訊息限制、系統範圍限制和系統記憶體執行緒。

目標訊息限制

由於訊息可在目標保留較長一段時間，因此會出現記憶體資源問題。您不應為目標配置太多記憶體（因為系統記憶體有限），也不應配置太少記憶體（訊息可能被拒絕）。

考慮到基於每個目標載入需求的靈活性，您可設定特性以管理每個目標的記憶體資源與訊息流量。例如，您可以指定允許用於目標的產生者最大數目、目標中允許的最大訊息數量（或大小），或每個單一訊息的最大大小。

您也可以指定當達到此類限制時，訊息路由器應採取下列四種回應中的哪一種回應。四個限制運作方式為：

- 減緩訊息產生者速度
- 拋出最舊的訊息
- 根據訊息存在時間，拋出最後順位的訊息
- 拒絕最新的訊息

系統範圍訊息限制

系統範圍訊息限制為第二道保護。您可以指定適用於代理程式上之所有目標的系統範圍限制：訊息總數目和所有訊息使用的記憶體。如果到達任何系統範圍訊息限制，訊息路由器會拒絕新的訊息。

系統記憶體執行緒

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

- 拋出資料儲存中永久性訊息的內部記憶體副本。

- 減低非永久性訊息的產生者，最終使訊息停止流入代理程式。永久性訊息流量會自動受到代理程式確認每個訊息之要求的限制。

上述兩種方法皆會降低效能。

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

藉由謹慎監視與調校，基於目標的限制與運作方式可用來平衡流入和流出訊息，如此一來，將不會發生系統過度負載情況。這些機制會耗用時間並限制訊息流量，但他們會維護作業完整性。

持續性管理程式

如果要讓代理程式在發生失敗時回復，則此代理程式需要重新建立其訊息傳送作業的狀態。此操作要求它將所有永久性訊息以及重要的路由與傳送資訊儲存至資料儲存。若要回復，代理程式還必須可執行以下作業：

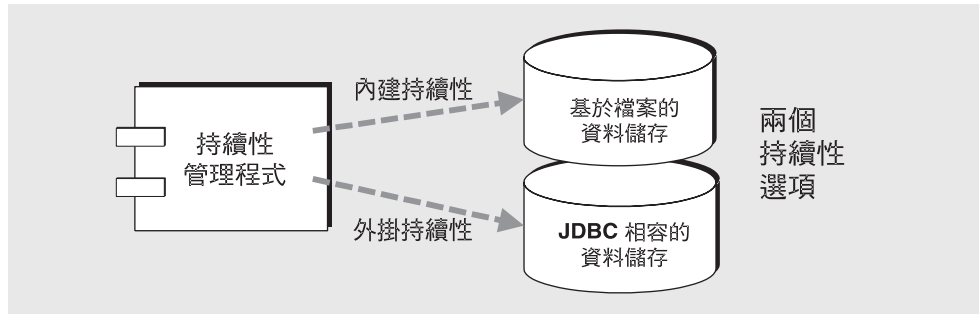
- 重新建立目標
- 復原每個主題的長期訂閱清單
- 復原每個訊息的確認清單
- 重新生產所有已確定作業事件的狀態

持續性管理程式管理所有此狀態資訊的儲存和擷取。

當重新啓動代理程式時，它會使用持續性管理程式所管理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，並為未傳送訊息重新建立其路由表格。然後，它可以繼續訊息傳送。

Message Queue 支援內建和外掛持續性模組（請參閱圖 4-4）。內建持續性是基於檔案的資料儲存。外掛持續性使用 Java 資料庫連線 (JDBC™) 介面，並需要 JDBC 相容的資料儲存。內建持續性通常快於外掛持續性；但是某些使用者更喜歡使用 JDBC 相容資料庫系統所提供的容錯和管理控制功能。

圖 4-4 持續性管理程式支援



內建持續性 預設 Message Queue 永久性儲存解決方案為基於檔案的資料儲存，基於檔案的資料儲存會使用個別檔案來儲存永久性資料。若要減少因新增和移除訊息而產生的分段程序，您可以壓縮基於檔案的資料儲存。若要最大化可靠性，您可以指定持續性作業同步化包含實體儲存裝置的內部記憶體狀態同步。這可以消除因系統當機而產生的資料遺失，但會降低效能。因為資料儲存可以包含敏感或專用性質的訊息，您應該保護資料儲存檔案的安全性，以防未經授權的存取。

外掛持續性 您可以設定代理程式，以透過 JDBC 驅動程式存取任何可存取的資料儲存。這包含設定一些與 JDBC 相關的代理程式配置特性，以及使用 Message Queue 的資料庫管理員公用程式來建立具有適當模式的資料儲存。「Message Queue 管理指南」中詳細介紹了這些程序和相關的配置特性。

安全性管理程式

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

- **認證**可確保只有經過授權的使用者可以建立與訊息伺服器的連線。
- **授權**則指定哪些使用者有權可以存取例如連線服務或目標等資源，以執行訊息服務所支援的特定作業。
- **加密**可保護訊息，以防透過連線傳送時遭到篡改。

認證與授權功能因使用者儲存庫而異 (請參閱第 69 頁圖 4-5)：包含有關訊息傳送系統使用者的資訊 (如使用者名稱、密碼和群組成員身份) 的檔案、目錄或資料庫。當請求與代理程式連線時，這些名稱和密碼用於認證使用者。使用者名稱和群組成員身份，連同存取控制檔案一起用於授權作業 (如針對目標生產或使用訊息)。

您可以寫入 Message Queue 提供的使用者儲存庫，或將預先存在的 LDAP 使用者儲存庫插入代理程式中。文字檔案使用者儲存庫非常易於使用；但是其安全性易受侵害，因此應僅將其用於評估與開發。LDAP 使用者儲存庫很安全，因此最適合用於生產。

以下小節說明認證、授權與加密。如需更多詳細資訊，請參閱 Message Queue 管理指南。

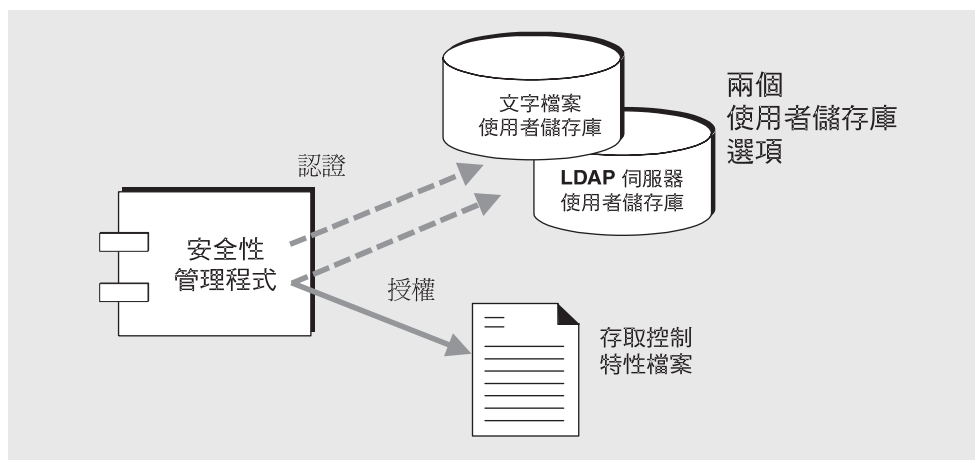
認證

Message Queue 安全性支援基於密碼的認證。當用戶端請求連線至代理程式時，它必須提供使用者名稱和密碼。安全性管理程式將此用戶端提交的名稱和密碼與儲存在使用者儲存庫中的名稱和密碼進行比較。將此密碼從用戶端傳送至代理程式時，系統會使用基本 64 編碼或訊息摘要 (MD5) 對密碼進行編碼。如需更加安全的傳送，請參閱第 70 頁「加密」。您可以分別配置每種連線服務所使用的編碼類型，或在代理程式範圍基礎上設定編碼。

授權

一旦認證用戶端應用程式的使用者，此使用者便可被授權執行各種與 Message Queue 相關的作業。安全性管理程式支援基於使用者和基於群組的存取控制：根據使用者儲存庫中的使用者名稱或指定給此使用者的群組，此使用者可擁有執行某些 Message Queue 作業的許可權。您可以在存取控制特性檔案 (請參閱圖 4-5) 中指定存取控制。

圖 4-5 安全性管理程式支援



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

- 連線至代理程式
- 存取目標：為給定目標或所有目標建立用戶、產生者或佇列瀏覽器
- 自動建立目標

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

此外，如果代理程式已配置為允許自動建立目標（請參閱第 65 頁「自動建立的目標」），則您可以透過編輯存取控制特性檔案來控制代理程式可為何人自動建立目標。

加密

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

監視服務

代理程式包括一些用於監視和診斷其作業的元件。這些元件如下：

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

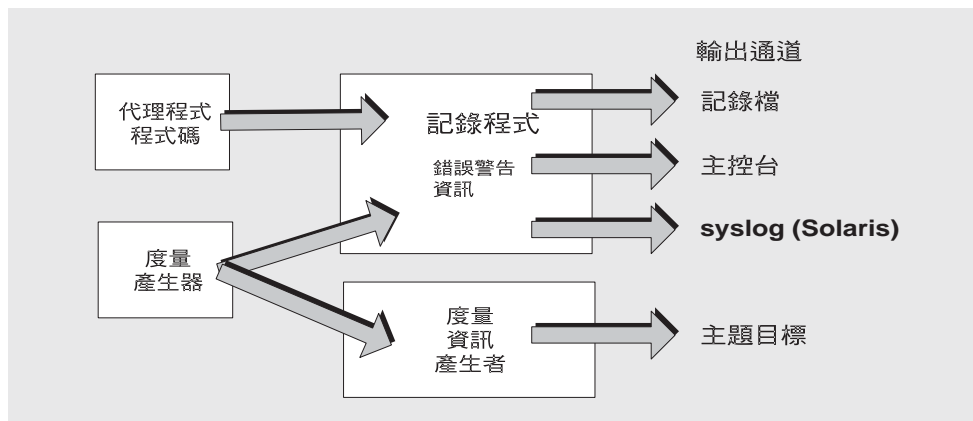
第 71 頁圖 4-6 中說明了一般方案。

度量產生器

圖 4-6 顯示的度量產生器提供代理程式活動的相關資訊，例如流入和流出代理程式的訊息流量、代理程式記憶體中的訊息數目和使用的記憶體、開啓連線的數目，以及使用的執行緒數目。

您可以開啓或關閉度量資料的產生，並指定產生度量報告的頻率。

圖 4-6 監視服務支援



記錄程式

圖 4-6 中顯示的 Message Queue 記錄程式會取得代理程式程式碼產生的資訊和度量產生器，並將資訊寫入一些輸出通道：標準輸出（主控台）、記錄檔及 Solaris™ 平台上的 syslog 常駐程式。

您可以指定記錄程式收集的資訊類型，以及寫入每個輸出通道的類型。在記錄檔中，您還可以指定何時關閉記錄檔並將輸出自動重建至新檔案。一旦記錄檔達到指定容量或存在時間，系統將儲存此記錄檔並建立新的記錄檔。

如需配置記錄程式以及如何使用它來取得效能資訊的詳細資訊，請參閱「Message Queue 管理指南」。

度量訊息產生者（企業版）

圖 4-6 中顯示的度量訊息產生者元件會在固定時間間隔，從度量產生器元件收到資訊，並將資訊寫入訊息，接著根據訊息中包含的度量資訊類型，將訊息傳送到一些度量主題目標。

指定至這些度量主題目標的 Message Queue 用戶端，可使用目標中的訊息並處理訊息中包含的度量資訊。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報告的度量數目詳細資訊，請參閱「Message Queue Developer's Guide for Java Clients」，以瞭解如何開發使用度量訊息的 Message Queue 用戶端。如需配置度量訊息產生的詳細資訊，請參閱「Message Queue 管理指南」。

開發與生產環境

開發與測試訊息傳送應用程式時，以及在生產環境部署與管理那些應用程式時，都需要 Message Queue 服務所提供的訊息傳送基礎結構。

本節介紹在開發與生產環境使用 Message Queue 的不同方法。它涵蓋下列主題：

- [第 72 頁「開發環境與工作」](#)
- [第 73 頁「生產環境與工作」](#)

雖然管理員需負責設定與管理生產環境和工作，但開發者也必須瞭解此環境，才能提供管理員設定與配置此環境以及判斷用戶端是否如預期般運作的相關資訊。

開發環境與工作

在開發環境中，工作重點為 Message Queue 用戶端應用程式的程式設計。Message Queue 服務主要用於測試。

出廠即可使用的配置

Message Queue 產品的設計是出廠即可使用。您可以使用預設值來啟動代理程式，它會提供您預設的資料儲存、使用者儲存庫，以及存取控制特性檔案。

預設的使用者儲存庫是使用預設項目所建立，它允許在安裝後立即使用 Message Queue 代理程式，無需管理員介入。換言之，要使用的代理程式無需初始使用者 / 密碼設定。預設使用者名稱 (guest) 與密碼 (guest) 可用於認證用戶端使用者。

此外，也提供一些範例應用程式，可引導您開發新應用程式。

開發慣例

在開發環境中強調靈活性，且通常您會採取以下作業：

- 使用最小管理 - 主要包含為開發者啟動代理程式以用於測試。
- 使用資料儲存的預設實作 (內建基於檔案的持續性)、使用者儲存庫 (基於檔案的使用者儲存庫) 與存取控制特性檔案。這些預設實作通常足以用於開發測試。
- 您可以使用簡單的檔案系統物件儲存 (透過建立特定用途的目錄) 以儲存受管理物件，或者您可以在用戶端程式碼中創設受管理物件而完全不使用物件儲存。
- 若要執行多重代理程式測試，則可能不使用主代理程式 (請參閱[第 77 頁「叢集同步化」](#))。
- 您通常會使用自動建立的目標，而不明確建立目標。

生產環境與工作

在生產環境中，必須部署、管理與調校應用程式，以最大化效能。在此狀況下，管理與調校訊息傳送基礎結構非常重要，雖不能稱為必要，但卻是基本需求。

此外，部署必須支援企業需求，不論是在擴充性或可用性方面。這需要針對訊息服務進行自訂配置與設定、調校效能與系統以符合日益增加的負載，以及針對訊息服務與應用程式特定資源執行每日監視與管理作業。因此，管理工作取決於訊息傳送系統的複雜性及其必須支援的應用程式的複雜性。下列各節將管理工作分成設定作業與維護作業來說明。如需執行這些工作所需的程序，請參閱「Message Queue 管理指南」。

設定作業

生產環境需要設定安全存取、配置連線工廠與目標物件、設定叢集、配置永久性儲存，以及管理記憶體。

► 若要設定生產環境

通常，您必須至少執行以下設定作業中的一些作業（如果不是全部）：

- **安全管理存取**（受保護的管理工具使用）：
 - 確定已啟動 admin 連線服務。
 - 授權：允許存取用於特定個人或 admin 群組的 admin 連線服務。
 - 如果授權給群組，則請確定管理員屬於管理群組。
 - 基於檔案的使用者儲存庫：有一個預設的 admin 群組。確定管理員在 admin 群組中，或在使用預設 admin 使用者時，變更 admin 密碼。
 - LDAP 使用者儲存庫：確定管理員在 admin 群組中
- **安全用戶端存取**：
 - 認證：將項目放入基於檔案的使用者儲存庫，或將代理程式配置為使用現有 LDAP 使用者儲存庫。
(至少，您需要密碼保護管理功能。)
 - 授權：修改存取控制特性檔案中的存取設定。
 - 加密：設定基於 SSL 的連線服務。
- **建立實體目標**
- **建立受管理物件**：
 - 配置或設定 LDAP 物件儲存。
 - 建立連線工廠和目標受管理物件。

- **建立代理程式叢集（如果有需要）：**
 - 建立中央配置檔案。
 - 指定主代理程式。
- **配置代理程式以使用外掛式持續性。**
- **配置記憶體管理**

設定目標屬性，以便訊息數目和訊息配置的記憶體容量，符合可用的代理程式記憶體資源。

維護作業

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

► 維護生產環境

- **管理應用程式運作方式**
 - 停用代理程式的自動建立功能。
 - 建立作為應用程式的實體目標。
 - 設定使用者對目標的存取權。
 - 監視與管理目標。
 - 監視與管理長期訂閱。
 - 監視與管理作業事件。
- **監視與調校代理程式**
 - 使用代理程式標準資訊調校和重新配置代理程式。
 - 管理代理程式記憶體資源。
 - 將代理程式新增至叢集以平衡負載。
 - 回復失敗的代理程式。
- **管理受管理物件：**
 - 依需要建立其他連線工廠和目標受管理物件。
 - 調整連線工廠屬性值以提昇效能與流量。

代理程式叢集

Message Queue 企業版支援使用代理程式叢集：代理程式的群組一起工作，為用戶端提供訊息傳送服務。叢集可使訊息伺服器透過分散用戶端的連線至數個代理程式，來調整其訊息流量的作業。

本章說明代理程式叢集的架構和內部功能。它涵蓋以下主題：

- [第 76 頁「叢集架構」](#)
- [第 78 頁「部署環境」](#)

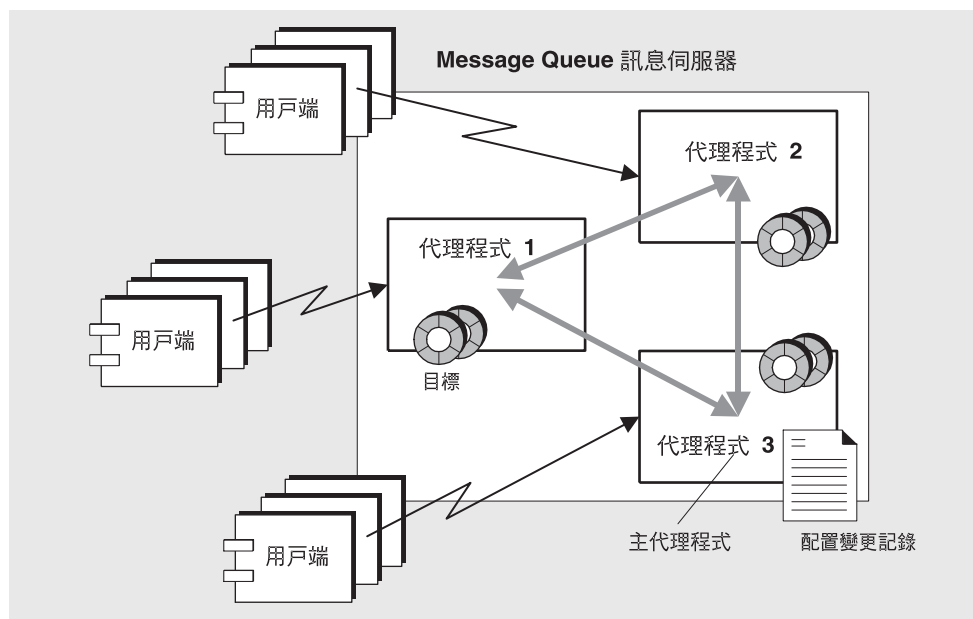
如果您是負責配置和管理代理程式叢集的管理員或是需要測試使用叢集的訊息應用程式之開發者，則您必須閱讀本章。

叢集架構

圖 5-1 顯示代理程式叢集的 Message Queue 架構。叢集中的每個代理程式均與所有其他代理程式直接相連。每個用戶端 (訊息產生者或用戶) 擁有單獨的本機代理程式並利用它直接溝通、傳送和接收訊息，猶如此本機代理程式為叢集中唯一的代理程式。不過真相是，本機代理程式與叢集中其他代理程式一起合作，分擔為所有連線的用戶端提供傳送服務的負載。

在叢集中的一個代理程式可被指定為主代理程式。主代理程式會維護配置變更記錄其中記錄了對叢集的永久性實體 (目標與長期訂閱) 所做的變更。此記錄是用來傳遞此類變更資訊給變更發生時離線的代理程式;請參閱以下「叢集同步化」中的詳細說明。

圖 5-1 叢集架構



下列各節說明如何在叢集中執行訊息傳送以及代理程式如何配置和同步，即使一或多個代理程式處於離線的狀態。

訊息傳送

在叢集配置中，每個目標都會在叢集中的所有代理程式上複製。每個代理程式均瞭解已註冊所有其他代理程式之目標的訊息用戶。因此，每個代理程式均可將訊息從其直接連線的訊息產生者路由至遠端訊息用戶，並將訊息從遠端產生者傳送至其直接連線的用戶。

訊息產生者的本機代理程式處理所有儲存和路由，並處理所有用戶端對該產生者所創建訊息的確認。若要最小化叢集中的訊息流量，只有在訊息將傳送給連線到目標代理程式的用戶時，才會將訊息從一個代理程式傳送到另一個代理程式。在某些情況下（如至多個用戶的佇列傳送），流量可再進一步降低，方法為指定優先傳送到本地用戶，之後再傳送到遠端用戶。某些情況要求用戶端和訊息伺服器間實現安全的加密訊息傳送，此時可以對叢集配置，使叢集中代理程式間的訊息傳送也實現安全性。

備註 在某些異常狀況中，叢集環境中的目標特性通常會整個套用到整個叢集，而非套用到個別的目標實例。請參閱「Message Queue 管理指南」以獲得特定目標特性的資訊。

叢集配置

若要在啟動時在叢集中的代理程式之間建立連線，每個代理程式必須傳送所有其他代理程式的主機名稱與連接埠號碼（包括主代理程式，如果有的話）。此資訊由一組**叢集配置特性**指定，而叢集中所有代理程式的這些設定都應該相同。雖然您可個別指定每個代理程式的配置特性，但此法易產生錯誤並可能導致叢集配置的不一致性。反之，我們建議您將叢集配置特性放在啟動時每個代理程式均參照的一個中央**叢集配置檔案**。此舉確保所有代理程式共用相同的配置資訊。

請參閱「Message Queue 管理指南」以獲得叢集配置特性的詳細資訊。

備註 雖然叢集配置檔案原意是用作配置目的，但它也是用來儲存叢集中所有代理程式共用的其他特性的好地方。

叢集同步化

每當叢集的配置變更時，關於變更的資訊即自動傳遞至叢集中的所有代理程式。此類配置變更包括：

- 叢集的其中一個代理程式上的目標被建立或銷毀。

- 訊息用戶以其本機代理程式註冊。
- 訊息用戶與其本機代理程式中斷連線（不論是明確中斷連線，或者因為用戶端、代理程式或網路故障）。
- 訊息用戶建立長期的主題訂閱。

此類配置變更資訊將立即傳遞至變更發生時叢集中所有連線中的代理程式。但是，離線的代理程式（例如，已損毀的代理程式）將不會在變更發生時收到通知。爲了顧及到離線的代理程式，**Message Queue** 爲叢集維護了一個**配置變更記錄**，記錄所有被建立或銷毀的永久性實體（目標與長期訂閱）。當離線的代理程式重新連線時（或當新的代理程式加入叢集中時），它會參考此記錄取得關於目標與長期訂閱的資訊，然後與其他代理程式交換有關目前使用中的訊息用戶的資訊。

叢集中的一個代理程式被指定爲**主代理程式**，負責維護配置變更記錄。因爲其他代理程式無法在主代理程式不在線上時完成它們的初始化，所以主代理程式應永遠是叢集中最先啓動的代理程式。如果主代理程式離線，則配置資訊將無法在叢集中傳遞，因爲其他代理程式無法存取配置變更記錄。在這些情況下，如果您嘗試建立或銷毀目標或長期訂閱，或嘗試執行如重新啓動長期訂閱的相關作業，則會出現異常。（但非管理訊息的傳送則會繼續正常工作。）

部署環境

代理程式叢集的使用取決於它們是部署在開發環境還是生產環境。

開發環境

在開發環境中，叢集用於測試，而延展性與代理程式回復並非重要注意事項，因此不需要主代理程式。在測試情況下，目標通常是自動建立的（請參閱第 65 頁「[自動建立的目標](#)」）且這些目標的長期訂閱由要進行測試的應用程式建立和銷毀。在未配置主代理程式的環境中，**Message Queue** 不需要主代理程式處於執行狀態以啓動其他代理程式，並且允許對目標與長期訂閱進行變更並將它們傳遞至所有執行中的代理程式。（但是，如果代理程式離線並隨後復原，則它將不會與其離線時所作的變更同步。）如果您重新配置環境以使用主代理程式，**Message Queue** 將重新強加正常需求。

生產環境

在生產環境中，延展性與代理程式回復是重要注意事項，因此必須使用主代理程式並維護配置變更記錄。此舉確保如果代理程式離線並隨後復原，則它將會與其離線時所作的變更同步。

事實上，最好定期備份配置變更記錄，以防止意外損毀記錄並避免主代理程式發生故障。**Message Queue** 為備份和復原配置變更記錄提供了指令行選項。如有必要，您也可變更作為主代理程式的代理程式。請參閱「**Message Queue 管理指南**」，以取得更多資訊。

部署環境

Message Queue 和 J2EE

Java 2 Platform Enterprise Edition (J2EE 平台) 是標準伺服器平台裝載多層式和精簡型用戶端企業應用程式的規格。J2EE 平台的其中一個要求是分散式元件能透過可靠的、非同步的訊息交換而彼此互動。這樣的互動能透過 JMS 提供者而達成。事實上，Message Queue 即是 J2EE 平台的參照 JMS 實作。

本章說明在 J2EE 平台環境中實作 JMS 支援的結果情形。本章包含下列主題：

- [第 82 頁「JMS/J2EE 程式設計：訊息驅動 Bean」](#)
- [第 83 頁「J2EE 應用程式伺服器支援」](#)

因為本章同時包含程式設計和部署 J2EE 元件，所以同時適合應用程式開發者和管理員閱讀。

JMS/J2EE 程式設計：訊息驅動 Bean

除第 24 頁「JMS 程式設計模型」中介紹的一般 JMS 用戶端程式設計模型之外，還有專用於 J2EE 平台應用程式環境的 JMS。此專用的 JMS 用戶端稱為**訊息驅動 Bean**，是 EJB 2.0 規格 (<http://java.sun.com/products/ejb/docs.html>) 中描述的 Enterprise JavaBean (EJB) 元件系列之一。

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

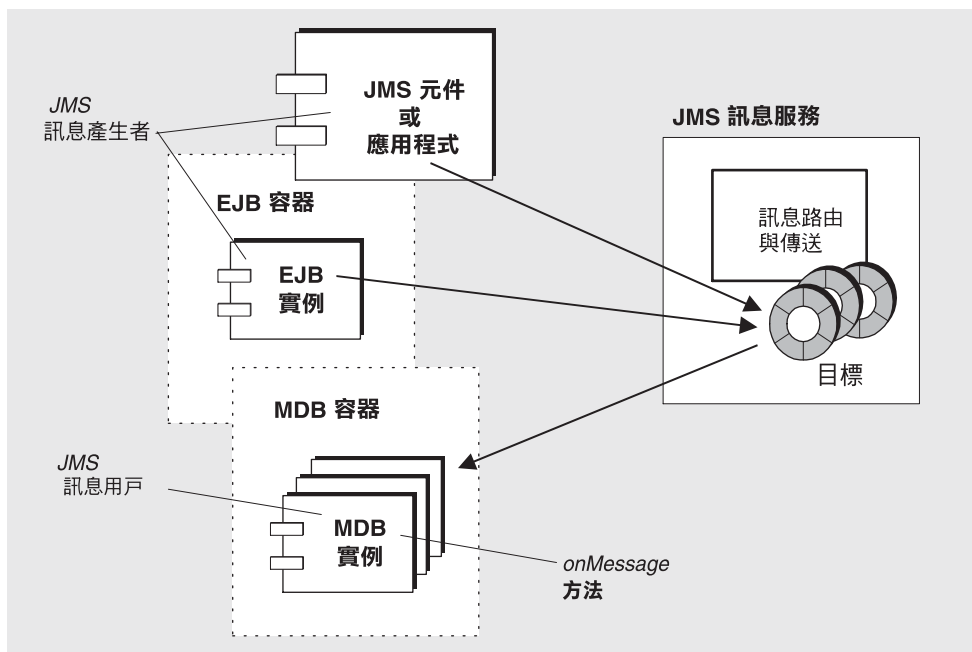
但是，非同步訊息傳送是許多企業應用程式中必不可少的要求。這些應用程式大都需要伺服器端元件在不佔用伺服器資源的情況下，能夠互相通訊和回應。因此，就出現了對 EJB 元件的需求，該元件不需要緊密耦合到訊息產生者即可接收和使用訊息。對於伺服器端元件必須回應應用程式事件的任何應用程式，均必須具備此功能。在企業應用程式中，此功能還必須能夠依據不斷增加的負載來進行延伸。

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

訊息驅動 Bean MDB 是實作 JMS MessageListener 介面的 JMS 訊息用戶。在 MDB 容器接收訊息時，會呼叫 onMessage 方法 (由 MDB 開發者編寫)。

onMessage() 方法使用該訊息的方式與標準 MessageListener 物件的 onMessage() 方法使用訊息的方式一樣。您不能以呼叫其他 EJB 元件的方法來遠端呼叫 MDB：因此不存在與其關聯的本機介面或遠端介面。MDB 可以使用來自單一目標的訊息。獨立式 JMS 應用程式、JMS 元件、EJB 元件或 Web 元件均可以生產訊息，如圖 6-1 中所示。

圖 6-1 與 MDB 進行訊息傳送



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

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

J2EE 應用程式伺服器支援

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

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

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

JMS 資源介面

資源介面是一個能將附加功能外掛到符合 J2EE 1.4 規格的應用程式伺服器中的標準方式。(該標準是由 J2EE Connector Architecture (J2EECA) 1.5 規格所定義。)此架構允許任何符合 J2EE 1.4 規格的應用程式伺服器，以標準方法與外部系統進行互動。外部系統可包括各種企業資訊系統 (EIS)，以及各種訊息傳送系統：例如，JMS 提供者。Message Queue 包括一個 JMS 資源介面能讓應用程式伺服器將 Message Queue 當作 JMS 提供者使用。

J2EECA 1.5 促成的標準互動包括連線集區、執行緒集區、作業事件、安全環境傳遞，以及各式訊息驅動 Bean 的支援。此規格亦包括建立連線工廠與其他受管理物件的標準方法。

插入 JMS 資源介面至應用程式伺服器，能讓在應用程式伺服器中已部署及執行的 J2EE 元件交換 JMS 訊息。使用 J2EE 應用程式伺服器管理工具，可以建立並配置這些元件所需的 JMS 連線工廠與目標受管理物件。

但是，像是管理訊息伺服器與實體目標的其他管理作業不包括在 J2EECA 規格中且這些作業只能使用提供者的特定工具來執行。

Message Queue 資源介面整合在 Sun J2EE 1.4 應用程式伺服器中。但是，還沒有任何其他 J2EE 1.4 應用程式伺服器認證過資源介面。

Message Queue 資源介面為單一檔案 (mqjmsra.rar) 位於作業系統的某個目錄中，目錄位置因作業系統而異 (請參閱「Message Queue 管理指南」)。mqjmsra.rar 檔案包含資源介面部署描述元 (ra.xml) 以及為讓應用程式伺服器使用該介面所需的 JAR 檔案。

遵循應用程式伺服器隨附的資源介面部署與配置說明，您即可使用任何 J2EE-1.4 相容應用程式伺服器中的 Message Queue 資源介面。當您可取得商業用 J2EE 1.4 應用程式伺服器，且 Message Queue 資源介面已經應用程式伺服器認證，Message Queue 文件將提供相關部署與配置程序的特定資訊。

選擇性 JMS 功能的 Message Queue 實作

JMS 規格會指出某些選擇性項目：每個 JMS 提供者（供應商）會選擇是否實作它們。本附錄描述 Message Queue 產品如何處理 JMS 選擇性項目。

本資料和應用程式開發者最有關係。

對每個 JMS 選擇性項目的處理指示於表 A-1：

表 A-1 選擇性 JMS 功能

JMS 規格中的小節	說明和 Message Queue 處理
3.4.3 JMSMessageID	<p>“Since message IDs take some effort to create and increase a message's size, some JMS providers may be able to optimize message overhead if they are given a hint that message ID is not used by an application. JMS Message Producer provides a hint to disable message ID.”</p> <p>Message Queue 實作： 產品不停用訊息 ID 的產生 (MessageProducer 中的所有 setDisableMessageID() 呼叫均被忽略)。所有訊息均將包含有效的 MessageID 值。</p>
3.4.12 Overriding Message Header Fields	<p>“JMS does not define specifically how an administrator overrides these header field values. A JMS provider is not required to support this administrative option.”</p> <p>Message Queue 實作： Message Queue 產品支援透過配置用戶端執行階段來管理覆寫訊息標頭欄位中的值（請參閱第 37 頁「覆寫訊息標頭值」）。</p>

表 A-1 選擇性 JMS 功能 (續)

JMS 規格中的小節	說明和 Message Queue 處理
3.5.9 JMS Defined Properties	<p>“JMS Reserves the 'JMSX' Property name prefix for JMS defined properties.”</p> <p>“Unless noted otherwise, support for these properties is optional.”</p> <p>Message Queue 實作： Message Queue 產品支援 JMS 1.1 規格定義的 JMSX 特性 (請參閱「Message Queue 管理指南」)。</p>
3.5.10 Provider-specific Properties	<p>“JMS reserves the 'JMS_<vendor_name>' property name prefix for provider-specific properties.”</p> <p>Message Queue 實作： 提供者特定的特性之用途是提供所需的特殊功能，以支援 JMS 與提供者原生用戶端一同使用。它們不應用於 JMS 以進行 JMS 訊息傳送。Message Queue 3 不會使用提供者特定的特性。</p>
4.4.8 Distributed Transactions	<p>“JMS does not require that a provider support distributed transactions.”</p> <p>Message Queue 實作： 此版本的 Message Queue 產品支援分散式作業事件 (請參閱第 56 頁「作業事件」)。</p>
4.4.9 Multiple Sessions	<p>“For PTP <point-to-point distribution model>, JMS does not specify the semantics of concurrent <code>QueueReceivers</code> for the same queue; however, JMS does not prohibit a provider from supporting this.” See section 5.8 of the JMS specification for more information.</p> <p>Message Queue 實作： Message Queue 實作支援到多個使用者的佇列傳送。如需更多資訊，請參閱第 53 頁「佇列傳送至多個用戶」。</p>

字彙表

本字彙表提供有關使用 Message Queue 時，可能遇到的專有名詞和概念的資訊。

確認 控制用戶端和訊息伺服器間的訊息交換，以確保可靠性。有兩種通用類型的確認：用戶端確認和代理程式確認。

受管理物件 預先配置的物件（連線工廠或目標）、封裝提供者特定的實作細節，由管理員建立以供一個或多個 JMS 用戶端使用。使用受管理物件可讓 JMS 用戶端獨立於提供者之外。受管理物件被放置在 JNDI 名稱空間中，並由使用 JNDI 查找的 JMS 用戶端存取。

非同步訊息傳送 一種訊息的交換，其訊息傳送不取決於用戶是否可以接收訊息。換言之，訊息傳送者在繼續其他工作之前無需等待傳送方法的傳回。如果訊息用戶忙碌或離線，訊息會先被傳送，然後在用戶準備就緒時接收訊息。

認證 用來確認只有經驗證的使用者才允許設定連線至訊息伺服器的過程。

授權 訊息服務決定使用者是否可存取訊息服務資源（如連線服務或目標）的過程，以執行訊息服務支援的特定作業。

代理程式 Message Queue 實體，可管理訊息路由、傳送、持續性、安全性以及記錄，並提供監視和調校效能與資源使用的介面。

用戶端 與使用訊息服務交換訊息的其他用戶端進行互動式操作的應用程式（或軟體元件）。用戶端可以是生產型用戶端、使用型用戶端，或兩者皆是。

用戶端識別碼 將連線及其物件與作為用戶端的 Message Queue 訊息伺服器所維護的狀態相關聯的識別碼。

用戶端執行階段 為訊息用戶端提供 Message Queue 訊息伺服器介面的 Message Queue 軟體。用戶端執行階段支援用戶端將訊息傳送至目標和從目標接收訊息所需的所有作業。藉由設定 ConnectionFactory 特性配置用戶端執行階段。

叢集 兩個或多個互連代理程式一起合作以提供可延伸的訊息傳送服務。

連線 在用戶端和訊息伺服器間的通訊通道，傳送有效負載訊息和控制訊息。

連線工廠 用戶端用於建立與訊息伺服器連線的受管理物件。這可能是一個 `ConnectionFactory` 物件、`QueueConnectionFactory` 物件或 `TopicConnectionFactory` 物件。

用戶 階段作業建立的用於接收從目標傳送訊息的物件 (`MessageConsumer`)。在點對點傳送模型中，用戶為接收者或瀏覽者 (`QueueReceiver` 或 `QueueBrowser`)；在出版 / 訂閱傳送模型中，用戶為訂戶 (`TopicSubscriber`)。

資料儲存 永久儲存代理程式所需資訊 (長期訂閱、有關目標的資料、永久性訊息以及稽核資料) 的資料庫。

停用的訊息 因正常處理或明確的管理員動作之外的原因而從系統上被移除的訊息。可能成為停用訊息的原因有：到期、因為超過記憶體限制而從目標移除或嘗試傳送失敗。您可選擇將停用的訊息儲存在停用訊息佇列上。

停用的訊息佇列 在代理程式啟動後，會自動建立一個專用目標，用於儲存作為診斷目的之停用訊息。

傳送模式 訊息傳送可靠性的指標：保證訊息傳送並成功使用一次且僅為一次 (永久性傳送模式)，或保證訊息至多傳送一次 (非永久性傳送模式)。

傳送模型 訊息傳送的模型：點對點或出版 / 訂閱。在 JMS 中，每種傳送模型都有單獨的程式設計網域，並且使用特定用戶端執行階段物件和特定目標類型 (佇列或主題)，以及統一的程式設計網域。

目標 `Message Queue` 訊息伺服器中的實體目標，已生產的訊息傳送至此目標，以便路由並隨後傳送至用戶。此實體目標由受管理物件識別和封裝，用戶端使用此受管理物件指定它要生產訊息和 / 或使用訊息的目標。

網域 JMS 用戶端用於程式化 JMS 訊息傳送作業的一組物件。有兩種程式設計網域：一種用於點對點傳送模型，另一種用於出版 / 訂閱傳送模型。

加密 一個保護訊息免於在透過連線傳送時被篡改的機制。

群組 `Message Queue` 用戶端用戶所屬的群組，旨在授權對連線、目標和特定作業的存取權。

JMS 提供者 為訊息傳送系統實作 JMS 介面，並新增配置和管理系統所需的管理和控制功能的產品。

訊息伺服器 一個或多個為 Message Queue 服務提供集中式傳送服務的代理程式，包括連線至用戶端、訊息處理和路由、持續性、安全性和監視。訊息伺服器維護實體目標，生產型用戶端將訊息傳送至此目標，並且訊息從此目標傳送至使用用戶端。

訊息服務 一個中介軟體服務提供非同步、在分散式元件或應用程式間的可靠訊息交換。它包含訊息伺服器、用戶端執行階段和數個執行訊息伺服器功能所需的資料儲存。

訊息 訊息用戶端使用的非同步請求、報告或事件。訊息包含標頭（其他欄位可新增至此）和內文。訊息標頭指定標準欄位和選擇性特性。訊息內文包含要傳送的資料。

訊息傳送 企業應用程式使用的非同步請求、報告或事件的系統，可讓無密切關聯的應用程式可靠且安全地傳送資訊。

產生者 由階段作業建立的物件 (MessageProducer)，用於將訊息傳送至目標。在點對點傳送模型中，產生者為傳送者 (QueueSender)；在出版 / 訂閱傳送模型中，產生者為出版者 (TopicPublisher)。

佇列 由管理員建立的可實作點對點傳送模型的物件。佇列始終用於保留訊息，即使使用其訊息的用戶端處於非使用中。佇列被用作產生者與用戶之間的媒介保留位置。

選擇器 訊息標頭特性用於訊息排序和路由。訊息服務執行基於訊息選擇器中所存在條件的訊息篩選與路由。

階段作業 用於傳送和接收訊息的單一執行緒環境。可以為佇列階段作業或主題階段作業。

主題 由管理員建立的可實作出版 / 訂閱傳送模型的物件。主題可作為內容階層（負責收集和分布傳送給它的訊息）中的節點來檢視。透過將主題用作媒介，訊息出版者可與訊息用戶保持獨立。

作業事件 必須完成或完全回復的不可分割的單元之工作。

A

admin 連線服務 62
API 文件 17
AUTO_ACKNOWLEDGE 模式 54

C

CLIENT_ACKNOWLEDGE 模式 55

D

DUPS_OK_ACKNOWLEDGE 模式 55

H

HTTP

支援架構 63
代理 64
功能說明 41
通道 Servlet 64
連線服務, 請參閱httpjms 連線服務
傳輸驅動程式 64

HTTP 連線

支援 64

通道 Servlet, 請參閱HTTP 通道 Servlet

httpjms 連線服務 62

HTTPS

支援架構 63
通道 Servlet 64
連線服務, 請參閱httpsjms 連線服務

HTTPS 連線

支援 64
通道 Servlet, 請參閱HTTPS 通道 Servlet

httpsjms 連線服務 62

J

J2EE 應用程式

EJB 規格 82
JMS, 和 82
支援, 功能說明 42
訊息驅動 Bean, 請參閱訊息驅動 Bean

JDBC 支援

功能說明 46
關於 68

JMS

訊息結構 23
規格 17
程式設計網域 25
程式設計模型 24

jms 連線服務 61

JMS 程式設計網域 25

L 部分

JNDI

- 受管理物件, 和 [29](#)
- 物件儲存 [39](#)
- 查找 [37](#)
- 訊息驅動 Bean, 和 [83](#)

L

LDAP 伺服器功能說明 [46](#)

M

MDB, *請參閱* 訊息驅動 Bean

N

NO_ACKNOWLEDGE 模式 [55](#)

S

SAAJ API

- javax.xml.messaging 套裝軟體 [42](#)
- javax.xml.soap 套裝軟體 [42](#)

SOAP

- 功能說明 [41](#)
- 端點受管理物件 [38](#)

SSL

- 功能說明 [41](#)
- 連線服務, *請參閱* 基於 SSL 的連線服務
關於 [70](#)

ssladmin 連線服務 [62](#)

ssljms 連線服務 [62](#)

T

TCP [61](#)

TLS [62](#)

W

Web 服務 [41](#)

X

XA 連線工廠

另請參閱 連線工廠受管理物件
訊息使用, 和 [56](#)

XA 資源管理程式, *請參閱* 分散式作業事件

XML 訊息傳送支援, 功能說明 [41](#)

三書

工具, 管理, *請參閱* 管理工具

二書

元件

- EJB [82](#)
- MDB [82](#)

內建持續性 [68](#)

分散式作業事件

- XA 資源管理員 [28](#)
- 另請參閱* XA 連線工廠
關於 [28](#)

二 書

主代理程式 77, 78

主題

 訊息路由, 和 53

 關於 26

代理程式

 互連, 請參閱代理程式叢集

 元件和功能 60

 主代理程式 77, 78

 多重代理程式叢集, 請參閱代理程式叢集

 安全性管理程式, 請參閱安全性管理程式

 度量, 請參閱代理程式度量

 持續性管理程式, 請參閱持續性管理程式

 重新啟動 67

 限制運作方式 66

 記憶體管理 66

 記錄, 請參閱記錄程式

 訊息流量控制, 請參閱訊息流量控制

 訊息路由, 請參閱訊息路由器

 從故障回復 67

 連線服務, 請參閱連線服務

 確認 (Ack) 52

 關於 33

代理程式確認

 用戶端執行階段的實作 36

 訊息使用, 和 54

代理程式叢集

 主代理程式 77, 78

 功能說明 43

 在生產環境中 79

 在開發環境中 78

 架構 76

 負載平衡的佇列傳送, 和 53

 配置變更記錄 78

 資訊傳遞 77

 叢集配置特性 77

 叢集配置檔案 77

出版 / 訂閱傳送 26

加密

 功能說明 41, 43

 關於 70

功能, 訊息佇列 40

可用性

 功能 44

 企業要求 20

 透過 Sun Cluster 44

可移植型, 請參閱提供者獨立性

可管理性

 功能 45

 企業要求 20

可靠的傳送

 JMS 規格 26

 用戶端執行階段功能 36

 企業要求 20

 效能平衡 57

外掛持續性 68

平台版 47

永久性訊息

 使用, 和 54

 定義 26

 訊息生產, 和 52

用戶

 作為 JMS 程式設計物件 25

 關於 21

用戶端

 C 語言支援, 功能說明 41

 JMS 程式設計模型 24

 效能, 請參閱效能

 執行階段, 請參閱用戶端執行階段

用戶端執行階段

 C 實作 34

 Java 實作 34

 分散訊息至用戶 35

 可靠的傳送功能 36

 用戶端標識, 與 35

 用戶端確認模式 54

 佇列瀏覽特徵 37

 流量控制, 功能說明 44

 訊息佇列, 說明 33

 訊息流量控制功能 37

 訊息壓縮 37

 連線處理功能 35

 覆寫訊息標頭值 37

六書 部分

用戶端設計,和效能 58

用戶端確認

訊息刪除,和 56

訊息使用,和 54

模式,請參閱用戶端確認模式

關於 27

用戶端確認模式

AUTO_ACKNOWLEDGE 54

CLIENT_ACKNOWLEDGE 55

DUPS_OK_ACKNOWLEDGE 55

NO_ACKNOWLEDGE 55

自訂訊息確認 55

效能,和 58

訊息使用,和 54

用戶端應用程式,範例 17

用戶端識別碼 (ClientID) 35

目標

介紹的 64

主題,請參閱主題

自動建立的 65

佇列,請參閱佇列

限制運作方式 66

訊息路由,和 52

停用的訊息佇列 65

管理員建立的 65

暫存 65

目標受管理物件

作為 JMS 程式設計物件 25

說明 38

目錄變數

IMQ_HOME 14

IMQ_JAVAHOME 15

IMQ_VARHOME 14

目錄變數 IMQ_HOME 14

目錄變數 IMQ_JAVAHOME 15

目錄變數 IMQ_VARHOME 14

六書

企業版 47

安全性

功能 42

企業要求 20

管理程式,請參閱安全管理程式

安全管理程式

作為代理程式元件 61

關於 68

安全套接層標準,請參閱SSL

有效負載訊息 51

自訂用戶端確認 55

自動建立的目標 65

自動重新連線

功能說明 44

六書

佇列

負載平衡的傳送,請參閱負載平衡的佇列傳送

訊息路由,和 52

瀏覽特徵 37

關於 25

佇列目標,請參閱佇列

作業事件

JMS 可靠性,和 27

分散式,請參閱分散式作業事件

訊息使用,和 56

確認,與 56

防火牆 63

八書

使用者群組 69

使用者儲存庫 68

協定,請參閱傳輸協定

協定類型

HTTP 62

TCP 61

TLS 62

受管理物件

- JMS 規格, 和 28
- SOAP 端點 38
- XA 連線工廠, 請參閱連線工廠受管理物件已說明 37
- 目標 38
- 物件儲存, 請參閱物件儲存
- 連線工廠, 請參閱連線工廠受管理物件
- 提供者獨立性, 與 39
- 管理控制, 與 38
- 類型 29, 37
- 延伸性
 - 功能 43
 - 企業要求 20
- 服務類型
 - ADMIN 61
 - NORMAL 61
- 版本, 產品
 - 比較的 46
 - 平台 47
 - 企業 47
- 物件儲存
 - JNDI, 和 39
 - LDAP 伺服器 39
 - 訊息佇列, 說明 39
 - 檔案系統儲存 39
- 長期訂閱
 - ClientID, 和 54
 - 訊息路由 54
 - 關於 26
- 長期訂閱, 請參閱長期訂閱
- 非同步訊息傳送
 - JMS 程式設計模型 24
 - 企業要求 20

I 畫

- 度量
 - 訊息 71
 - 訊息產生者 71
 - 報告 70
 - 資料, 請參閱代理程式度量

- 持續性
 - 內建 68
 - 可配置的, 功能說明 46
 - 外掛, 請參閱外掛持續性
 - 持續性管理程式, 請參閱持續性管理程式
 - 傳送模式, 請參閱傳送模式
 - 資料儲存, 請參閱資料儲存
- 持續性管理程式
 - 作為代理程式元件 61
 - 資料儲存, 請參閱資料儲存
 - 關於 67
- 指令行公用程式 39
- 訂閱
 - 長期, 請參閱長期訂閱
 - 關於 25
- 負載平衡的佇列傳送
 - 功能說明 43
 - 機制 53
- 重新傳送旗標 57
- 限制運作方式
 - 代理程式 66
 - 目標 66

十畫

- 容器
 - EJB 83
 - MDB 83
- 效能
 - 代理程式限制運作方式, 與 66
 - 可靠性平衡 57
 - 用戶端設計, 和 58
 - 用戶端確認模式, 和 58
 - 因素影響 57
 - 訊息流量控制, 和 58
 - 訊息流量控制, 與 37
 - 傳送模式, 和 58
 - 調校, 功能說明 45
- 記憶體管理
 - 用於代理程式 66
- 記錄, 請參閱記錄程式

記錄程式

- 作為代理程式元件 61
- 輸出通道 71
- 關於 71

訊息

- JMS 23
- JMS 內文類型 23
- JMS 特性 23
- 代理程式確認 36
- 出版 / 訂閱傳送 26
- 可靠的傳送 49
- 可靠傳送 27
- 永久性, 請參閱永久性訊息
- 永久性儲存體 52
- 有效負載 51
- 使用 54
- 使用終了 56
- 持續性 67
- 負載平衡的佇列傳送 53
- 重新傳送 57
- 偵聽程式 25, 36
- 控制 51
- 產生 52
- 結構 23
- 傳送模式, 請參閱傳送模式
- 傳送模型 25
- 路由 52
- 標頭, 請參閱訊息標頭欄位
- 選取及篩選 23
- 壓縮 37
- 點對點傳送 25

訊息用戶, 請參閱用戶

訊息伺服器

- 訊息佇列, 說明 33
- 資源管理, 功能說明 44
- 關於 21

訊息服務

- JMS 22
- 架構 21
- 訊息佇列服務架構 32

訊息流量控制

- 代理程式 66
- 效能, 和 58

效能, 與 37

訊息偵聽程式, 請參閱偵聽程式

訊息產生者, 請參閱產生者

訊息傳送

- 可靠性 49
- 步驟和階段 50
- 使用終了 56
- 非同步, 請參閱非同步傳送
- 訊息生產 52
- 訊息使用 54
- 處理和路由 52

訊息傳送, 非同步, 請參閱非同步訊息傳送

訊息傳送系統

- 企業 20
- 架構 21
- 訊息服務 21

訊息傳送模型 25

訊息路由器

- 作為代理程式元件 61
- 關於 64

訊息監視 API, 功能說明 45

訊息標頭欄位

- JMS 訊息 23
- 覆寫 37

訊息驅動 Bean

- MDB 容器 83
- 部署描述元 83
- 應用程式伺服器支援 83
- 關於 82

十一 卷

停用的訊息佇列 65

偵聽程式

- MDB, 和 82
- 作為 JMS 程式設計物件 25

執行緒儲存區管理程式

- 關於 63

控制訊息 37, 51

授權

- 另請參閱存取控制檔案
- 功能說明 43
- 關於 69
- 授權, 訊息佇列版本 47, 48
- 產生者
 - 作為 JMS 程式設計物件 25
 - 關於 21
- 許可權
 - Message Queue 作業 69
 - 存取控制特性檔案 70
 - 資料儲存 68
- 連接埠, 動態配置 63
- 連接埠對映器 62
- 連線
 - 可延伸的, 功能說明 43
 - 作為 JMS 程式設計物件 25
 - 防故障備用, 請參閱自動重新連線
- 連線工廠受管理物件
 - ClientID, 和 54
 - JNDI 查找 29
 - 用戶端標識屬性 35
 - 作為 JMS 程式設計物件 25
 - 說明 38
- 連線服務
 - admin 62
 - httpjms 62
 - httpsjms 62
 - jms 61
 - 執行緒儲存區管理程式 63
 - 連接埠對映器, 請參閱連接埠對映器
 - 關於 60

十二 畫

- 提供者獨立性 39
- 階段作業
 - JMS 用戶端確認 27
 - 作為 JMS 程式設計物件 25
 - 作業事件的 27

十三 畫

- 傳送, 可靠的 20
- 傳送, 可靠的, 請參閱可靠的傳送
- 傳送模式
 - 永久性 26
 - 非永久性 26
 - 效能, 和 58
 - 訊息生產, 和 52
- 傳輸協定
 - 功能說明 40
 - 協定類型, 請參閱協定類型
- 資料儲存
 - JDBC 可存取的 68
 - 文字檔案 68
 - 關於 67
- 資源介面
 - Message Queue 實作 84
 - 功能說明 42
- 路由, 請參閱訊息路由器

十四 畫

- 管理工作
 - 生產環境 73
 - 開發環境 72
- 管理工具
 - 功能說明 45
 - 指令行公用程式 39
 - 管理主控台 39
 - 關於 39
- 管理主控台 39
- 管理員建立的目標 65
- 網域 25
- 認證
 - 功能說明 43
 - 關於 69

十三 章

暫存目標 [65](#)

確認

JMS 可靠性, 和 [27](#)

代理程式 [56](#)

代理程式, 和訊息生產 [52](#)

用戶端, *請參閱*用戶端確認

作業事件, 與 [56](#)

範例應用程式 [17](#)

十四 章

應用程式, *請參閱*用戶端應用程式

應用程式伺服器, 和 Message Queue [83](#)

環境變數, *請參閱*目錄變數

點對點傳送 [25](#)

十五 章

叢集配置特性 [77](#)

叢集配置檔案 [77](#)

十六 章

穩定性, 功能說明 [44](#)