



Sun Java System Application Server Enterprise Edition 8.1 2005Q2 管理指南



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

文件號碼：819-3512

Copyright 2005 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/地區的待批專利。

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。

本發行物可能包含由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 Sun™ 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。

本出版物所涵蓋的產品和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。嚴禁核武、導彈、生化武器或海上核武等最終用途或一般使用者直接或間接使用本產品。嚴禁向被美國禁運的國家/地區或美國出口除外清單 (包括但不僅限於被拒人清單和特別指定的國家/地區清單) 上標識的實體出口或再出口本產品。

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
1 入門	31
關於 Sun Java System Application Server	31
什麼是 Application Server?	31
Application Server 架構	32
管理工具	33
Application Server 配置	35
配置 Application Server	35
配置網域	35
建立網域	36
刪除網域	36
列示網域	36
啓動網域	37
重新啓動伺服器或網域	37
停止網域	38
重新建立網域管理伺服器	38
▼ 遷移 DAS	38
Application Server 實例	40
關於 Application Server 實例	40
定義 Application Server 實例	41
關於獨立實例	42
檢視一般伺服器資訊	43
▼ 部署應用程式	43
▼ 建立新的資源類型	44
Administration Server 進階設定	44
▼ 配置自動部署設定	45
實例特定的配置特性	46

▼ 若要將值復原為預設值，請執行以下步驟：	46
▼ 增加實例特性	46
▼ 刪除特性	47
▼ 建立實例	47
▼ 啓動實例	48
恢復作業事件	49
▼ 停止實例	49
▼ 關閉 Administration Server	49
配置變更	49
變更 Application Server 配置	50
Application Server 中的連接埠	50
▼ 檢視連接埠號	51
▼ 變更 Administrative Server 連接埠	51
▼ 變更 HTTP 連接埠	51
▼ 變更 IIOP 連接埠	52
▼ 使用管理服務配置 JMX 連接器	52
▼ 編輯 JMX 連接器配置	53
變更 J2SE 軟體	53
使用線上說明	54
▼ 返回至上一個說明螢幕	54
詳細資訊	54
2 部署應用程式	55
關於部署	55
部署生命週期	55
J2EE 歸檔檔案的類型	56
命名慣例	57
有關部署應用程式的管理主控台作業	58
▼ 部署企業應用程式	58
▼ 編輯已部署的企業應用程式	60
▼ 部署 Web 應用程式	60
▼ 啓動已部署的 Web 應用程式	62
▼ 部署 EJB 模組	62
▼ 部署連接器模組	64
▼ 建立生命週期模組	65

▼ 部署應用程式用戶端模組	66
▼ 指定用於部署的歸檔檔案	68
有關列示、取消部署以及啓用應用程式的 管理主控台 作業	68
▼ 列示已部署的應用程式	68
▼ 列示子元件	69
▼ 檢視已部署的應用程式的模組描述元	69
▼ 取消部署應用程式或模組	70
▼ 啓用或停用應用程式或模組	70
▼ 管理應用程式目標	71
▼ 在其他虛擬伺服器上部署	71
重新部署到多個目標	71
▼ 配置動態重新載入	72
適用於開發者的部署方法	73
▼ 使用自動部署	73
▼ 部署目錄中未封裝的應用程式	74
使用 <code>deploytool</code> 公用程式	74
使用部署規劃	75
3 JDBC 資源	77
關於 JDBC 資源和連線池	77
JDBC 資源	77
JDBC 連線池	78
JDBC 資源和連線池如何協同工作	78
設定資料庫存取	78
▼ 設定資料庫存取	79
▼ 整合 JDBC 驅動程式	79
關於 JDBC 連線池	80
▼ 建立 JDBC 連線池	80
▼ 編輯 JDBC 連線池	81
▼ 驗證連線池設定	84
▼ 刪除 JDBC 連線池	84
關於 JDBC 資源	84
▼ 建立 JDBC 資源	84
▼ 編輯 JDBC 資源	85
▼ 刪除 JDBC 資源	86

▼ 啓用或停用 JDBC 資源	86
關於持續性管理程式資源	87
▼ 建立持續性管理員資源	87
▼ 編輯持續性管理員資源	88
▼ 管理資源目標	88
▼ 刪除持續性管理員資源	88
▼ 啓用或停用持續性管理員資源	89
4 配置 Java 訊息服務資源	91
關於 JMS 資源	91
Application Server 中的 JMS 提供者	91
JMS 資源	91
JMS 資源與連接器資源之間的關係	93
有關 JMS 連線工廠的管理主控台作業	93
▼ 建立 JMS 連線工廠資源	93
▼ 編輯 JMS 連線工廠資源	96
▼ 刪除 JMS 連線工廠資源	96
有關 JMS 目標資源的管理主控台作業	97
▼ 建立 JMS 目標資源	97
▼ 編輯 JMS 目標資源	98
▼ 刪除 JMS 目標資源	98
有關 JMS 實體目標的管理主控台作業	99
▼ 建立 JMS 實體目標	99
▼ 刪除 JMS 實體目標	100
有關 JMS 提供者的管理主控台作業	101
▼ 配置 JMS 提供者的一般特性	101
▼ 驗證 JMS 服務設定	104
▼ 建立 JMS 主機	104
▼ 編輯 JMS 主機	106
▼ 刪除 JMS 主機	106
5 配置 JavaMail 資源	109
關於 JavaMail	109
有關 JavaMail 的管理主控台作業	109
▼ 建立 JavaMail 階段作業	110

▼ 編輯 JavaMail 階段作業	111
▼ 刪除 JavaMail 階段作業	111
6 JNDI 資源	113
關於 Java 命名與目錄介面 (JNDI)	113
JNDI 名稱和資源	113
J2EE 命名服務	113
命名參考與連結資訊	114
關於自訂資源	115
使用自訂資源	115
▼ 建立自訂資源	115
▼ 編輯自訂資源	116
▼ 刪除自訂資源	116
列示自訂資源	117
關於外部 JNDI 儲存庫和資源	117
使用外部 JNDI 儲存庫和資源	117
▼ 建立外部資源	118
▼ 編輯外部資源	119
▼ 刪除外部資源	119
列示外部資源	120
7 連接器資源	121
關於連接器	121
有關連接器連線池的管理主控台作業	122
▼ 設定 EIS 存取	122
▼ 建立連接器連線池	122
▼ 編輯連接器連線池	123
▼ 刪除連接器連線池	124
有關連接器資源的管理主控台作業	125
▼ 建立連接器資源	125
▼ 編輯連接器資源	126
▼ 刪除連接器資源	126
▼ 配置連接器服務	126
有關管理物件資源的管理主控台作業	127
▼ 建立管理物件資源	127

▼ 編輯管理物件資源	128
▼ 刪除管理物件資源	128
8 J2EE 容器	131
關於 J2EE 容器	131
J2EE 容器的類型	131
Web 容器	131
EJB 容器	131
有關 J2EE 容器的管理主控台作業	132
配置一般 Web 容器設定	132
配置 Web 容器階段作業	132
▼ 設定階段作業逾時值	132
▼ 配置管理員特性	133
▼ 配置儲存特性	134
配置一般 EJB 設定	134
▼ 配置 EJB 池設定	135
▼ 配置 EJB 快取設定	136
配置訊息導引 Bean 設定	137
▼ 配置 MDB 池設定	137
配置 EJB 計時器服務設定	138
▼ 配置計時器服務	138
▼ 將外部資料庫與計時器服務搭配使用	138
9 配置安全性	141
關於 Application Server 的安全性	141
安全性簡介	141
關於認證與授權	146
瞭解使用者、群組、角色和範圍	148
證書和 SSL 簡介	151
關於防火牆	153
使用管理主控台管理安全性	154
有關安全性的管理主控台作業	156
▼ 配置安全性設定	156
▼ 授予對管理工具的存取權	158
有關範圍的管理主控台作業	159

▼ 建立範圍	159
▼ 編輯範圍	160
▼ 刪除範圍	161
▼ 設定預設範圍	162
特定範圍的其他資訊	162
有關 JACC 提供者的管理主控台作業	170
▼ 建立 JACC 提供者	170
▼ 編輯 JACC 提供者	171
▼ 刪除 JACC 提供者	172
▼ 設定使用中的 JACC 提供者	172
有關稽核模組的管理主控台作業	173
▼ 建立稽核模組	173
▼ 編輯稽核模組	174
▼ 刪除稽核模組	174
▼ 啓用或停用稽核記錄	175
▼ 設定使用中的稽核模組	175
▼ 使用 default 稽核模組	176
有關偵聽程式和 JMX 連接器的管理主控台作業	177
▼ 配置 HTTP 偵聽程式的安全性	177
▼ 配置 IIOP 偵聽程式的安全性	178
▼ 配置管理服務之 JMX 連接器的安全性	178
▼ 設定偵聽程式安全性特性	179
有關虛擬伺服器的管理主控台 安全性作業	179
▼ 配置單次登入 (SSO)	180
有關連接器連線池的管理主控台作業	181
關於連接器連線池	181
關於安全性對映	181
▼ 建立安全性對映	182
▼ 編輯安全性對映	183
▼ 刪除安全性對映	184
使用證書和 SSL	184
關於證書檔案	184
▼ 變更憑證檔案的位置	185
使用 Java 安全套接字延伸 (JSSE) 工具	185
▼ 使用 keytool 公用程式產生憑證	187
▼ 使用 keytool 公用程式簽署數位憑證	188

使用網路安全服務 (NSS) 工具	189
詳細資訊	192
10 配置訊息安全性	193
關於訊息安全性	193
訊息安全性概況	193
瞭解 Application Server 中的訊息安全性	194
確保 Web 服務的安全	197
確保範例應用程式的安全	198
配置 Application Server 以實現訊息安全性	198
▼ 配置其他安全性功能	199
▼ 配置 JCE 提供者	200
有關訊息安全性的管理主控台作業	201
▼ 啓用訊息安全性的提供者	201
▼ 配置訊息安全性提供者	203
▼ 建立訊息安全性提供者	205
▼ 刪除訊息安全性配置	207
▼ 刪除訊息安全性提供者	208
▼ 啓用應用程式用戶端的訊息安全性	208
設定應用程式用戶端配置的請求策略和回應策略	209
詳細資訊	210
11 作業事件	213
關於作業事件	213
何為作業事件？	213
J2EE 技術中的作業事件	214
有關作業事件的管理主控台作業	214
配置作業事件	214
▼ 配置 Application Server 如何從作業事件中回復	215
▼ 設定作業事件逾時值	215
▼ 設定作業事件記錄的位置	216
▼ 設定關鍵點間隔	217

12 配置 HTTP 服務	219
關於 HTTP 服務	219
什麼是 HTTP 服務?	219
虛擬伺服器	219
HTTP 偵聽程式	220
有關 HTTP 服務的管理主控台作業	222
▼ 配置 HTTP 服務	223
▼ 配置 HTTP 服務存取記錄	224
▼ 配置 HTTP 服務請求處理執行緒	226
▼ 配置 HTTP 服務持續作用子系統	226
▼ 配置 HTTP 服務連線池	227
▼ 為 HTTP 服務配置 HTTP 通訊協定	227
▼ 為 HTTP 服務配置 HTTP 檔案快取	228
有關虛擬伺服器的管理主控台作業	229
▼ 建立虛擬伺服器	229
▼ 編輯虛擬伺服器	231
▼ 刪除虛擬伺服器	232
有關 HTTP 偵聽程式的管理主控台作業	233
▼ 建立 HTTP 偵聽程式	233
▼ 編輯 HTTP 偵聽程式	235
▼ 選取要編輯的 HTTP 偵聽程式	235
13 配置物件請求代理程式	237
關於物件請求代理程式	237
CORBA	237
什麼是 ORB?	237
IIOP 偵聽程式	238
有關 ORB 的管理主控台作業	238
▼ 配置 ORB	238
有關 IIOP 偵聽程式的管理主控台作業	239
▼ 建立 IIOP 偵聽程式	239
▼ 編輯 IIOP 偵聽程式	240
▼ 刪除 IIOP 偵聽程式	241

14 執行緒池	243
關於執行緒池	243
有關執行緒池的管理主控台作業	244
▼ 建立執行緒池	244
▼ 編輯執行緒池	245
▼ 刪除執行緒池	246
15 配置記錄	247
關於記錄	247
記錄檔的記錄	247
記錄程式名稱空間階層結構	248
有關記錄的管理主控台作業	249
▼ 配置一般記錄設定	250
▼ 配置記錄層級	251
▼ 檢視伺服器記錄	252
16 監視元件和服務	255
關於監視	255
監視 Application Server	255
監視概述	255
關於可監視物件的樹狀結構	256
關於受監視的元件和服務的統計資訊	259
有關啓用和停用監視功能的管理主控台作業	279
▼ 使用管理主控台配置監視層級	279
▼ 使用 asadmin 工具配置監視功能	280
有關檢視監視資料的管理主控台作業	281
▼ 在管理主控台中檢視監視資料	281
使用 asadmin 工具檢視監視資料	283
▼ 使用 asadmin 工具檢視監視資料	283
▼ 使用 PetStore 範例	287
使用 JConsole	296
▼ 停用 JMX 連接器的安全性	296

17	Java 虛擬機器和進階設定	299
	有關 JVM 設定的管理主控台作業	299
	▼ 配置 JVM 一般設定	299
	▼ 配置 JVM 類別路徑設定	300
	▼ 配置 JVM 選項	301
	▼ 停用安全性管理員	302
	▼ 配置 JVM 效能評測器設定	302
	有關進階設定的管理主控台作業	303
	▼ 設定進階網域屬性	303
18	編譯和配置 Apache Web Server	305
	安裝 Apache	305
	適用於 Apache 1.3 的最低需求	305
	適用於 Apache 2 的最低需求	306
	▼ 安裝可識別 SSL 的 Apache	307
19	自動重新啟動網域或節點代理程式	311
	在 UNIX 平台上自動重新啟動	311
	在 Microsoft Windows 平台上自動重新啟動	312
	自動重新啟動的安全性	313
20	domain.xml 的含點名稱屬性	315
	頂層元素	315
	不能別名化的元素	317
	索引	319

圖清單

圖 1-1	Application Server 架構	32
圖 1-2	Application Server 實例	42
圖 9-1	角色對映	149

表清單

表 1-1	網域屬性值	46
表 1-2	使用連接埠的 Application Server 偵聽程式	50
表 6-1	JNDI 查找及其關聯的參考	114
表 9-1	Application Server 認證方法	147
表 9-2	ldap 範圍的必需特性	163
表 9-3	ldap 範圍的選擇性特性	163
表 9-4	ldap 範圍值範例	164
表 9-5	certificate 範圍的選擇性特性	166
表 9-6	file 範圍的必要特性	166
表 10-1	訊息保護策略與 WS-Security SOAP 訊息安全性作業對映	199
表 15-1	Application Server 記錄程式名稱空間	248
表 16-1	EJB 統計資訊	260
表 16-2	EJB 方法統計資訊	260
表 16-3	EJB 階段作業儲存統計	261
表 16-4	EJB 池統計資訊	262
表 16-5	EJB 快取統計	262
表 16-6	計時器統計資訊	263
表 16-7	Web 容器 (Servlet) 統計資訊	263
表 16-8	Web 容器 (Web 模組) 統計資訊	264
表 16-9	HTTP 服務統計資訊 (僅適用於 Platform Edition)	265
表 16-10	JDBC 連線池統計	266
表 16-11	連接器連線池統計資訊	267
表 16-12	連接器工作管理統計資訊	267
表 16-13	ORB 中連線管理程式的統計資訊	268
表 16-14	執行緒池統計資訊	268
表 16-15	作業事件服務統計資訊	268
表 16-16	JVM 統計資訊	269
表 16-17	J2SE 5.0 的 JVM 統計資訊 - 類別載入	269

表 16-18	J2SE 5.0 的 JVM 統計資訊 - 編譯	270
表 16-19	J2SE 5.0 的 JVM 統計資訊 - 資源回收	270
表 16-20	J2SE 5.0 的 JVM 統計資訊 - 記憶體	270
表 16-21	J2SE 5.0 的 JVM 統計資訊 - 作業系統	271
表 16-22	J2SE 5.0 的 JVM 統計資訊 - 執行階段	271
表 16-23	J2SE 5.0 的 JVM 統計 — 執行緒資訊	272
表 16-24	J2SE 5.0 的 JVM 統計資訊 - 執行緒	273
表 16-25	PWC 虛擬伺服器統計資訊 (僅限於 EE)	274
表 16-26	PWC 請求統計 (僅限於 EE)	274
表 16-27	PWC 檔案快取統計資訊 (僅限於 EE)	275
表 16-28	PWC 持續作用統計 (僅限於 EE)	276
表 16-29	PWC DNS 統計 (僅限於 EE)	276
表 16-30	PWC 執行緒池統計資訊 (僅限於 EE)	277
表 16-31	PWC 連線佇列統計資訊 (僅限於 EE)	277
表 16-32	PWC HTTP 服務統計資訊 (僅限於 EE)	278
表 16-33	頂層	291
表 16-34	應用程式層級	291
表 16-35	應用程式 - 企業應用程式和獨立模組	292
表 16-36	HTTP 服務層級	294
表 16-37	執行緒池層級	295
表 16-38	資源層級	295
表 16-39	作業事件服務層級	295
表 16-40	ORB 層級	296
表 16-41	JVM 層級	296

範例清單

範例 16-1	應用程式節點樹狀結構	256
範例 16-2	HTTP 服務示意圖 (Platform Edition 版)	257
範例 16-3	HTTP 服務示意圖 (Enterprise Edition 版)	258
範例 16-4	資源示意圖	258
範例 16-5	連接器服務示意圖	258
範例 16-6	JMS 服務示意圖	259
範例 16-7	ORB 示意圖	259
範例 16-8	執行緒池示意圖	259

前言

「管理指南」說明了如何從 Administration Console 配置、管理和部署 Application Server 子系統與元件。

本書適用對象

本「管理指南」適用於生產環境中的資訊技術管理員。本指南假設您已經熟悉以下主題：

- 基本系統管理作業
- 安裝軟體
- 使用 Web 瀏覽器
- 啓動資料庫伺服器
- 在終端機視窗中發佈指令

閱讀本書之前

Application Server 是 Sun Java™ Enterprise System 的一個元件，Sun Java Enterprise System 是一種軟體基礎架構，支援分散在網路或網際網路環境中的企業應用程式。您應熟悉 Sun Java Enterprise System 隨附的文件，該文件可從以下網站進行線上存取：<http://docs.sun.com/app/docs/prod/entsys.05q4#hic> 和 http://docs.sun.com/app/docs/prod/entsys.05q4#hic?l=zh_TW

本書架構

本指南的架構與 Administration Console 的配置相對應，Administration Console 是基於瀏覽器的工具，用於管理 Application Server。每章都以概念性資訊開頭，隨後的部分提供了如何使用 Administration Console 執行特定作業的程序性資訊。

本書使用的慣例

本小節中的表格描述了本書中使用的慣例。

印刷排版慣例

下表描述本書在印刷排版上所做的變更。

表 P-1 印刷排版慣例

字體	含義	範例
AaBbCc123 (固定間距)	API 和語言元素、HTML 標籤、網站 URL、指令名稱、檔案名稱、目錄路徑名稱、螢幕畫面輸出和範例碼。	編輯 <code>.login</code> 檔案。 使用 <code>ls -a</code> 列示所有檔案。 <code>% You have mail.</code>
AaBbCc123 (固定間距粗體)	您所鍵入的資訊，與螢幕畫面輸出相對應。	<code>% suPassword:</code>
<i>AaBbCc123</i> (斜體)	書籍標題、新術語和要強調的文字。 將用實際名稱或值替代的指令或路徑名稱中的萬用字元。	請參閱使用者指南中的第 6 章。 這些稱為 類別 選項。 請勿儲存檔案。 該檔案位於 <code>install-dir/bin</code> 目錄中。

符號

下表描述本書中使用的符號慣例。

表 P-2 符號慣例

符號	說明	範例	含義
[]	包含選擇性指令選項。	<code>ls [-l]</code>	無需 <code>-l</code> 選項。
{ }	包含為所需指令選項提供的一組選擇。	<code>-d {y n}</code>	<code>-d</code> 選項要求您使用 <code>y</code> 引數或 <code>n</code> 引數。
-	連接需同時按下的多個按鍵。	Control-A	同時按 Control 鍵和 A 鍵。
+	連接需連續按下的多個按鍵。	Ctrl+A+N	按 Control 按鍵，然後鬆隔 A 依次按後面的按鍵。

表 P-2 符號慣例 (續)

符號	說明	範例	意義
>	表示圖形化使用者介面中的功能表項目選取。	[檔案] > [新建] > [範本]	從 [檔案] 功能表中，選擇 [新建]。從 [新建] 子功能表中，選擇 [範本]。

預設路徑和檔案名稱

下表描述在本書中使用的預設路徑和檔案名稱。

表 P-3 預設路徑和檔案名稱

術語	說明
<i>install-dir</i>	<p>依預設，Application Server 的安裝目錄位於：</p> <ul style="list-style-type: none"> ■ Solaris™ 平台上的 Sun Java Enterprise System 安裝： /opt/SUNWappserver/appserver ■ Linux 平台上的 Sun Java Enterprise System 的安裝： /opt/sun/appserver/ ■ 其他 Solaris 和 Linux 的安裝 (非超級使用者)： user's home directory/SUNWappserver ■ 其他 Solaris 和 Linux 的安裝 (超級使用者)： /opt/SUNWappserver ■ Windows，所有安裝： SystemDrive:\Sun\AppServer
<i>domain-root-dir</i>	<p>依預設，包含所有網域的目錄位於：</p> <ul style="list-style-type: none"> ■ Solaris 平台上的 Sun Java Enterprise System 的安裝： /var/opt/SUNWappserver/domains/ ■ Linux 平台上的 Sun Java Enterprise System 的安裝： /var/opt/sun/appserver/domains/ ■ 所有其他安裝： <i>install-dir</i>/domains/

表 P-3 預設路徑和檔案名稱 (續)

術語	說明
<i>domain-dir</i>	依預設，每個網域目錄都位於以下位置： <i>domain-root-dir/domain-dir</i> 在配置檔案中，您可能會看到 <i>domain-dir</i> 顯示如下： <code>\${com.sun.aas.instanceRoot}</code>
<i>instance-dir</i>	依預設，每個實例目錄都位於以下位置： <i>domain-dir/instance-dir</i>

Shell 提示符號

下表描述本書中使用的 shell 提示符號。

表 P-4 Shell 提示符號

Shell	提示符號
UNIX 或 Linux 上的 C shell	<i>machine-name%</i>
UNIX 或 Linux 上的 C shell 超級使用者	<i>machine-name#</i>
UNIX 或 Linux 上的 Bourne shell 和 Korn shell	\$
UNIX 或 Linux 上的 Bourne shell 和 Korn shell 超級使用者	#
Windows 命令行	C:\

符號

下表描述本書中使用的符號慣例。

表 P-5 符號慣例

符號	說明	範例	含義
[]	包含選擇性指令選項。	<code>ls [-l]</code>	無需 <code>-l</code> 選項。
{ }	包含為所需指令選項提供的一組選擇。	<code>-d {y n}</code>	<code>-d</code> 選項要求您使用 <code>y</code> 引數或 <code>n</code> 引數。
-	連接需同時按下的多個按鍵。	Control-A	同時按 Control 鍵和 A 鍵。

表 P-5 符號慣例 (續)

符號	說明	範例	意義
+	連接需連續按下的多個按鍵。	Ctrl+A+N	按 Control 按鍵，然後鬆開A依次按後面的按鍵。
>	表示圖形化使用者介面中的功能表項目選取。	[檔案]>[新建]>[範本]	從 [檔案] 功能表中，選擇 [新建]。從 [新建] 子功能表中，選擇 [範本]。

預設路徑和檔案名稱

下表描述在本書中使用的預設路徑和檔案名稱。

表 P-6 預設路徑和檔案名稱

術語	說明
<i>install-dir</i>	<p>依預設，Application Server 的安裝目錄位於：</p> <ul style="list-style-type: none"> ■ Solaris 平台上的 Sun Java Enterprise System 安裝： /opt/SUNWappserver/appserver ■ Linux 平台上的 Sun Java Enterprise System 的安裝： /opt/sun/appserver/ ■ 其他 Solaris 和 Linux 的安裝 (非超級使用者)： <i>user's home directory</i>/SUNWappserver ■ 其他 Solaris 和 Linux 的安裝 (超級使用者)： /opt/SUNWappserver ■ Windows，所有安裝： <i>SystemDrive</i>:\Sun\AppServer
<i>domain-root-dir</i>	<p>依預設，包含所有網域的目錄位於：</p> <ul style="list-style-type: none"> ■ Solaris 平台上的 Sun Java Enterprise System 的安裝： /var/opt/SUNWappserver/domains/ ■ Linux 平台上的 Sun Java Enterprise System 的安裝： /var/opt/sun/appserver/domains/ ■ 所有其他安裝： <i>install-dir</i>/domains/

表 P-6 預設路徑和檔案名稱 (續)

術語	說明
<i>domain-dir</i>	依預設，每個網域目錄都位於以下位置： <i>domain-root-dir/domain-dir</i> 在配置檔案中，您可能會看到 <i>domain-dir</i> 顯示如下： \${com.sun.aas.instanceRoot}
<i>instance-dir</i>	依預設，每個實例目錄都位於以下位置： <i>domain-dir/instance-dir</i>

相關文件

<http://docs.sun.com>SM 網站可讓您存取 Sun 線上技術文件。您可以瀏覽歸檔檔案或搜尋特定書籍標題或主旨。

您可以在 *install-dir/docs/index.htm* 中找到有關官方技術規定的 URL 目錄。此外，以下資源可能會有用。

一般 J2EE 資訊：

「*The J2EE 1.4 Tutorial*」

： <http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html>

「*The J2EE Blueprints*」： <http://java.sun.com/reference/blueprints/index.html>

「*Core J2EE Patterns: Best Practices and Design Strategies*」，由 Deepak Alur、John Crupi 和 Dan Malks 編寫，Prentice Hall 出版

「*Java Security*」，由 Scott Oaks 編寫，O'Reilly 出版

使用 Servlet 和 JSP 檔案程式設計：

「*Java Servlet Programming*」，由 Jason Hunter 編寫，O'Reilly 出版

「*Java Threads, 2nd Edition*」，由 Scott Oaks 和 Henry Wong 編寫，O'Reilly 出版

使用 EJB 元件程式設計：

「*Enterprise JavaBeans*」，由 Richard Monson-Haefel 編寫，O'Reilly 出版

使用 JDBC 程式設計：

「*Database Programming with JDBC and Java*」，由 George Reese 編寫，O'Reilly 出版

「*JDBC Database Access With Java: A Tutorial and Annotated Reference (Java Series)*」，由 Graham Hamilton、Rick Cattell 和 Maydene Fisher 編寫

Java 文件：

Application Server 隨附之套裝軟體的 Java 文件位於 *install-dir/docs/api* 中。

此文件集中的書籍

Sun Java System Application Server 手冊可做為線上檔案以可攜式文件格式 (PDF) 和超文字標記語言 (HTML) 格式提供。

下表總結了包含在 Application Server 核心文件集中的書籍。

表 P-7 此文件集中的書籍

書籍標題	說明
版本說明	軟體與文件的最新資訊。其中包括以表格形式對所支援的硬體、作業系統、JDBC/RDBMS 所做的全面概括。
Quick Start Guide	如何開始使用 Application Server 產品。
Installation Guide	安裝 Application Server 軟體及其元件。
Deployment Planning Guide	評估系統需求和企業狀況，確保以最適合您的站點的方式部署 Sun Java System Application Server。此外還說明了部署應用伺服器時應該注意的常見問題。
Developer's Guide	建立和實作 Sun Java System Application Server 上執行的 Java 2 Platform, Enterprise Edition (J2EE™ 平台) 應用程式，這些應用程式遵循針對 J2EE 元件和 API 的標準模型。其中包括有關開發者工具、安全性、組合、部署、除錯和建立生組的一般資訊。
J2EE 1.4 Tutorial	使用 J2EE 1.4 平台技術和 API 開發 J2EE 應用程式，並在 Sun Java System Application Server 上部署這些應用程式。
管理指南	從管理主控台配置、管理和部署 Sun Java System Application Server 子系統和
High Availability Administration Guide	高可用性資料庫安裝後的配置和管理說明。
Administration Reference	編輯 Sun Java System Application Server 配置檔案 <code>domain.xml</code> 。
Upgrade and Migration Guide	將應用程式遷移到新的 Sun Java System Application Server 程式設計模型，特別是在 Application Server 6.x 和 7 進行遷移。該指南還說明可導致與產品規格不相容的發行版本和配置選項之間的差異。
Performance Tuning Guide	調校 Sun Java System Application Server 以提昇效能。
Troubleshooting Guide	解決 Sun Java System Application Server 問題。
Error Message Reference	解決 Sun Java System Application Server 錯誤訊息。

表 P-7 此文件集中的書籍 (續)

書籍標題	說明
Reference Manual	與 Sun Java System Application Server 一起提供的公用程式指令；以線上手冊樣式編寫。其中包含 <code>asadmin</code> 指令行介面。

其他伺服器文件

如需有關伺服器的其他文件，請至以下位置：

- Message Queue 文件 <http://docs.sun.com/db?p=prod/s1.s1msgqu>
(<http://docs.sun.com/db?p=prod/s1.s1msgqu>)
- Directory Server 文件 http://docs.sun.com/coll/DirectoryServer_04q2
(http://docs.sun.com/coll/DirectoryServer_04q2) 和
http://docs.sun.com/app/docs/coll/DirectoryServer_04q2_zh_TW
(http://docs.sun.com/app/docs/coll/DirectoryServer_04q2_zh_TW)
- Web Server 文件 http://docs.sun.com/coll/S1_websvr61_en
(http://docs.sun.com/coll/S1_websvr61_en) 和
http://docs.sun.com/app/docs/coll/S1_websvr61_zh_TW
(http://docs.sun.com/app/docs/coll/S1_websvr61_zh_TW)

存取 Sun 線上資源

如需有關產品下載、專業服務、修補程式及支援和其他開發者資訊，請至以下位置：

- 下載中心 <http://www.sun.com/software/download/>
(<http://www.sun.com/software/download/>)
- 專業服務 <http://www.sun.com/service/sunps/sunone/index.html>
(<http://www.sun.com/service/sunps/sunone/index.html>)
- Sun Enterprise Services、Solaris 修補程式和支援 <http://sunsolve.sun.com/>
(<http://sunsolve.sun.com/>)
- 開發者資訊 <http://developers.sun.com/prodtech/index.html>
(<http://developers.sun.com/prodtech/index.html>)

連絡 Sun 技術支援

如果您在本產品文件中找不到所需之本產品相關技術問題的解答，請至：<http://www.sun.com/service/contacting> (<http://www.sun.com/service/contacting>)。

協力廠商網站參照

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> (<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頂部找到。例如，本書的標題為「Sun Java System Application Server 2005Q2 管理指南」，文件號碼為 819-3512。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：819-2554 和「Sun Java System Application Server Enterprise Edition 8.1 2005Q2 Administration Guide」。

◆ ◆ ◆ 第 1 章

1

入門

本章說明 Sun Java™ System Application Server，並說明基本的管理作業。它包含以下小節：

- 第 31 頁的「關於 Sun Java System Application Server」
- 第 35 頁的「Application Server 配置」
- 第 40 頁的「Application Server 實例」
- 第 49 頁的「配置變更」

關於 Sun Java System Application Server

- 第 31 頁的「什麼是 Application Server？」
- 第 32 頁的「Application Server 架構」
- 第 33 頁的「管理工具」

什麼是 Application Server ？

Application Server 為開發、部署和管理企業應用程式提供了牢固的 J2EE 平台。主要功能包括作業事件管理、效能、可延伸性、安全性以及整合。Application Server 支援的服務包含從 Web 發佈到企業範圍內的作業事件處理，同時可讓開發人員基於 JavaServer Pages (JSP™)、Java servlet 以及 Enterprise JavaBeans™ (EJB™) 技術建置應用程式。

Application Server 提供了進階叢集和容錯移轉技術。這些功能讓您可以執行可延伸的且具有高可用性的 J2EE 應用程式。

- **叢集** - 叢集是一組 Application Server 實例，它們做為一個邏輯實體一起工作。叢集中的每個 Application Server 實例均部署有相同配置和相同的應用程式。
透過將 Application Server 實例增加至叢集來實現水平比例縮放，從而增加系統容量。可以在不中斷服務的情況下將 Application Server 實例增加至叢集。HTTP、RMI/IIOP 和 JMS 負載平衡系統會將請求分散到叢集中運作狀態良好的 Application Server 實例中。

- **高可用性** - 可用性允許對叢集中的 Application Server 實例進行容錯移轉保護。如果一個 Application Server 實例出現故障，則其他 Application Server 實例將接管指定給該故障伺服器的階段作業。階段作業資訊儲存在高可用性資料庫 (HADB) 中。HADB 支援 HTTP 階段作業和有狀態階段作業 Bean 的持續性。

Application Server 架構

本小節說明圖 1-1，該圖顯示了 Application Server 的高階架構。

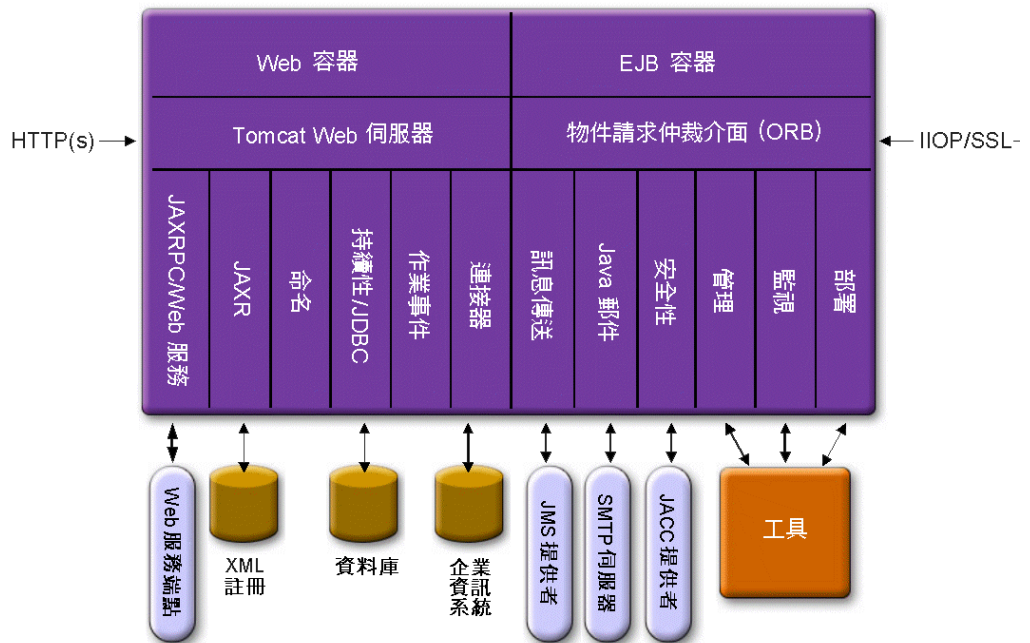


圖 1-1 Application Server 架構

- **容器** - 容器是一種執行階段環境，它為 J2EE 元件提供安全性和作業事件管理等服務。圖 1-1 顯示了兩種類型的 J2EE 容器：Web 和 EJB。Web 元件 (例如 JSP 頁面和 Servlet) 在 Web 容器內執行。企業 Bean (EJB 技術的元件) 在 EJB 容器內執行。
- **用戶端存取** - 在執行階段，瀏覽器用戶端透過 HTTP (在網際網路中使用的協定) 與 Web 伺服器進行通訊來存取 Web 應用程式。HTTPS 協定用於需要安全通訊的應用程式。企業 Bean 用戶端透過 IIOP 協定或 IIOP/SSL (安全) 協定與物件請求代理程式 (ORB) 通訊。Application Server 具有分別用於 HTTP 通訊協定、HTTPS 通訊協定、IIOP 協定和 IIOP/SSL 協定的偵聽程式。每個偵聽程式專用特定的連接埠號。

- **Web 服務** - 在 J2EE 平台上，可以部署一個 Web 應用程式，該應用程式可以提供由基於 XML 的 RPC 之 Java API (JAX-RPC) 實作的 Web 服務。J2EE 應用程式或元件還可以是其他 Web 服務的用戶端。應用程式通過用於 XML 登錄的 Java API (JAXR) 存取 XML 登錄。
- **用於應用程式的服務** - J2EE 平台旨在使容器為應用程式提供服務。圖 1-1 顯示了以下服務：
 - **命名** - 命名和目錄服務可將物件連結到名稱。J2EE 應用程式通過查找物件的 JNDI 名稱來找到物件。JNDI 代表 Java 命名和目錄介面 API。
 - **安全性** - Java 容器授權合約 (JACC) 是一組為 J2EE 容器定義的安全性合約。依照用戶端的身份，容器可限制對容器資源和服務的存取。
- **作業事件管理** - 作業事件是不可分割的工作單元。例如，在銀行帳戶之間轉帳是一個作業事件。作業事件管理服務用於確定完全完成作業事件或將作業事件轉返。

存取外部系統

J2EE 平台使應用程式能夠存取應用程式伺服器之外的系統。應用程式通過稱為資源的物件連線到這些系統。管理員的職責之一是資源配置。J2EE 平台可以通過以下 API 和元件存取外部系統：

- **JDBC** - 資料庫管理系統 (DBMS) 提供了用於儲存、組織和擷取資料的工具。大多數企業應用程式將資料儲存在關聯式資料庫中，這些應用程式透過 JDBC API 存取關聯式資料庫。由於資料庫中的資訊儲存在磁碟上並在應用程式結束之後仍然存在，因此通常將資料庫中的資訊稱為持續性資訊。Application Server 束包含 PointBase DBMS。
- **郵件傳送** - 郵件傳送是軟體元件或應用程式之間的一種通訊方法。訊息傳送用戶端可以向任何其他用戶端傳送訊息，也可以從任何其他用戶端接收訊息。應用程式通過 Java 訊息傳送服務 (JMS) API 存取訊息傳送提供者。Application Server 包含 JMS 提供者。
- **連接器** - J2EE 連接器架構允許 J2EE 應用程式和現有企業資訊系統 (EIS) 之間的整合。應用程式通過稱為連接器或資源配接卡的可攜式 J2EE 元件存取 EIS。
- **JavaMail** - 透過 JavaMail API，應用程式連線至 SMTP 伺服器以傳送和接收電子郵件。
- **伺服器管理** - 圖 1-1 的右下角顯示了由 Application Server 的管理員執行的一些作業。例如，管理員部署 (安裝) 應用程式並監視伺服器的效能。這些作業透過 Application Server 提供的管理工具來執行。

管理工具

- Application Server 包括三種管理工具：
 - 第 34 頁的「管理主控台」
 - 第 34 頁的「asadmin 公用程式」

- [第 35 頁的「Application Server Management Extension \(AMX\)」](#)

管理主控台

管理主控台是一種基於瀏覽器的工具，具有易於導覽的介面和線上說明。本手冊提供使用管理主控台的逐步說明。必須執行 Administration Server 才能使用管理主控台。

安裝 Application Server 時，您已為伺服器選擇連接埠號，或已使用預設連接埠 4849。還指定了使用者名稱和主密碼。

若要啟動管理主控台，請在 Web 瀏覽器中鍵入以下內容：

```
https://hostname:port
```

例如：

```
https://kindness.sun.com:4849
```

如果管理主控台在安裝了 Application Server 的機器上執行，請將 localhost 指定為主機名稱。

在 Windows 上，從 [開始] 功能表啟動 Application Server 管理主控台。

安裝程式將建立使用預設連接埠號 4849 的預設管理網域 (名為 domain1)，並會建立獨立於網域管理伺服器 (DAS) 的實例。安裝之後，還可以建立其他管理網域。每個網域都具有自己的網域管理伺服器，該伺服器具有唯一的連接埠號。為管理主控台指定 URL 時，請務必使用要管理的網域的連接埠號。

如果配置中包括遠端伺服器實例，請建立節點代理程式以便管理和簡化遠端伺服器實例。節點代理程式負責建立、啟動、停止和刪除伺服器實例。使用指令行介面 (CLI) 指令可以設定節點代理程式。

asadmin 公用程式

asadmin 公用程式是一個指令行工具。使用 asadmin 公用程式及其相關指令執行可以在管理主控台中可執行的相同作業集。例如，啟動和停止網域、配置伺服器以及部署應用程式。

可以在 Shell 的指令提示符號下使用這些指令，也可以從其他程序檔和程式呼叫這些指令。使用這些指令可以自動執行重複的管理作業。

若要啟動 asadmin 公用程式，請使用：

```
$ asadmin
```

若要列示 asadmin 中的可用指令，請使用：

```
asadmin> help
```

也可以在 Shell 的指令提示符號下發出 `asadmin` 指令：

```
$ asadmin help
```

若要檢視指令的語法和範例，請鍵入 `help` 並在其後鍵入指令名稱。例如：

```
asadmin> help create-jdbc-resource
```

所指定指令的 `asadmin help` 資訊可顯示此指令的 Unix 線上手冊。也可以使用 HTML 格式檢視這些線上說明手冊。

Application Server Management Extension (AMX)

Application Server Management eXtension 是一個 API，它可顯示所有 Application Server 配置，並可將 JMX 管理 Bean 做為實作 AMX 介面的、易於使用的用戶端動態代理伺服器來進行監視。

如需有關使用 Application Server Management Extension 的更多資訊，請參閱「Application Server Developer's Guide」中的「JMX」一章。

Application Server 配置

- 第 35 頁的「[配置 Application Server](#)」
- 第 35 頁的「[配置網域](#)」
- 第 37 頁的「[啟動網域](#)」
- 第 37 頁的「[重新啟動伺服器或網域](#)」
- 第 38 頁的「[停止網域](#)」
- 第 38 頁的「[重新建立網域管理伺服器](#)」

配置 Application Server

Application Server 網域是為協助管理員管理系統配置而建立的邏輯或實體單元。一個網域分為很多包括實例和節點代理程式的較小單元。伺服器實例是在單一實體機器上執行 Application Server 的單一 Java 虛擬機器 (JVM)。每個網域都有一個或多個實例。網域還必須至少有一個相關聯的節點代理程式才能使實例正常工作。可以將網域群組在一起以建立一個叢集。叢集使管理員可以管理多組硬體和軟體。

配置網域

管理網域提供了一種基本的安全結構，不同的管理員可以藉此管理 Application Server 實例的特定群組 (網域)。透過將伺服器實例群組至單獨的網域中，不同的組織和管理員可以共用單一 Application Server 安裝。每個網域都有自己的獨立於其他網域的配置、記錄檔和應用程式部署區域。如果變更某個網域的配置，其他網域的配置不會受到影響。

每個管理主控台階段作業均允許您配置和管理網域。如果建立了多個網域，則必須啓動其他 Administration Console 階段作業以管理每個網域。每個網域都具有自己的 Domain Administration Server (DAS)，該伺服器具有唯一的連接埠號。每個管理網域可以有 multiple 應用程式伺服器實例。但是，一個應用程式伺服器實例只能屬於一個網域。安裝 Application Server 時，將自動建立名為 `domain1` 的管理網域。

建立網域

網域是使用 `create-domain` 指令建立的。以下範例指令將建立名為 `mydomain` 的網域。Administration Server 在連接埠 1234 上進行偵聽，管理使用者名為 `hanan`。該指令提示您輸入管理密碼和主密碼。

```
$ asadmin create-domain --adminport 80 --adminuser hanan mydomain
```

若要為 `mydomain` 網域啓動管理主控台，請在瀏覽器中輸入以下 URL：

```
http://hostname:80
```

對於前面的 `create-domain` 範例，網域的記錄檔、配置檔案和部署的應用程式現在常駐於以下目錄中：

```
domain-root-dir/mydomain
```

若要在其他位置建立網域目錄，請指定 `--domaindir` 選項。如需完整的指令語法，請鍵入 `asadmin help create-domain`。

刪除網域

使用 `asadmin delete-domain` 指令可刪除網域。僅具有網域管理權限的作業系統使用者 (或 `root` 使用者) 才能成功地執行該指令。例如，若要刪除名為 `mydomain` 的網域，請鍵入以下指令：

```
$ asadmin delete-domain mydomain
```

列示網域

使用 `asadmin list-domains` 指令可找到在機器中建立的網域。若要列示預設 `domain-root-dir` 目錄中的網域，請鍵入以下指令：

```
$ asadmin list-domains
```

若要列示在其他目錄中建立的網域，請指定 `--domaindir` 選項。

啓動網域

啓動網域時，將啓動管理伺服器 and 應用程式伺服器實例。啓動應用程式伺服器實例之後，應用程式伺服器實例將持續執行、偵聽並接受請求。必須單獨啓動各個網域。

若要啓動網域，請鍵入 `asadmin start-domain` 指令並指定網域名稱。例如，若要啓動預設網域 (`domain1`)，請鍵入以下指令：

```
$ asadmin start-domain --user admin domain1
```

如果只有一個網域，則可以省略網域名稱。如需完整的指令語法，請鍵入 `asadmin help start-domain`。如果省略了密碼資料，系統將提示您提供此資料。

`asadmin start-domain domain1` 指令應由安裝伺服器的使用者來執行。如果未由該使用者執行，則應從安裝使用者的主目錄中將 `.asadmintruststore` 移動或複製到執行使用者的主目錄中。

請注意，如果將該檔案從安裝使用者的主目錄移動 (而非複製) 到執行使用者的主目錄，您可能會遇到應用程式升級問題，因為升級/安裝使用者的主目錄中將不再具有 `.asadmintruststore` 檔案。

復原已刪除的 `.asadmintruststore` 檔案

如果已刪除某網域的 `.asadmintruststore` 檔案，還是可以輕鬆建立新檔案

▼ 建立新的 `.asadmintruststore` 檔案

- 1 使用本機 `asadmin start-domain` 指令啓動您要管理的網域。
請注意，本機 `asadmin` 指令不需要 `.asadmintruststore` 檔案即可啓動網域。
- 2 執行任何遠端 `asadmin` 指令。
執行遠端 `asadmin` 指令時需要指定 `--user`、`--passwordfile` (`--password`)、`--host` 和 `--port` 選項，還要求目標網域處於執行狀態。
- 3 當系統提示時，按下「y」鍵即可接受新的網域憑證。

若要在 Windows 中啓動預設網域，請執行以下步驟：

在 Windows [開始] 功能表中，依次選取 [程式集] -> [Sun Microsystems] -> [Application Server] -> [啓動 Admin Server]。

重新啓動伺服器或網域

重新啓動伺服器與重新啓動網域相同。若要重新啓動網域或伺服器，請停止然後再啓動網域。

停止網域

停止網域將關閉該網域的管理伺服器 and 應用程式伺服器實例。停止網域時，伺服器實例將停止接受新的連線，然後等待所有未完成的連線完成。由於伺服器實例必須完成其關閉程序，因此該程序需要幾秒鐘時間。停止網域時，管理主控台 或大多數 `asadmin` 指令都無法使用。

若要停止網域，請鍵入 `asadmin stop-domain` 指令並指定網域名稱。例如，若要停止預設網域 (`domain1`)，請鍵入以下指令：

```
$ asadmin stop-domain domain1
```

如果只有一個網域，則網域名稱是選擇性的。如需完整語法，請鍵入 `asadmin help stop-domain`。

使用 管理主控台 停止網域：

- 在樹形元件中，選取 [獨立實例] 節點下的伺服器 (管理伺服器)。
- 在 [一般資訊] 頁面中，按一下 [停止伺服器]。

若要在 Windows 中停止預設網域，請執行以下步驟：

在 [開始] 功能表中，依次選取 [程式集] -> [Sun Microsystems] -> [Application Server] -> [停止 Admin Server]。

重新建立網域管理伺服器

若要進行鏡像並提供網域管理伺服器 (DAS) 的工作副本，您必須具有：

- 一台包含原始 DAS 的機器 (`machine1`)。
- 一台包含叢集的機器 (`machine2`)，該叢集具有執行應用程式並滿足用戶端需要的伺服器實例。該叢集是使用第一台機器上的 DAS 配置的。
- 一台備份機器 (`machine3`)，當第一台機器當機時，需要在該備份電腦上重新建立 DAS。

備註 - 必須保留一份第一台機器上的 DAS 的備份。使用 `asadmin backup-domain` 來備份目前網域。

▼ 遷移 DAS

以下步驟用於將 Domain Administration Server 從第一台機器 (`machine1`) 遷移到第三台機器 (`machine3`)：

- 1 將 **Application Server** 安裝在第三台機器上，方法與在第一台機器安裝時上相同。
爲了可以在第三台機器上正確地復原 DAS 並且不會發生路徑衝突，您必須執行此操作。
 - a. 使用指令行 (互動) 模式來安裝 **Application Server** 管理套裝軟體。若要啓動指令行互動模式，請使用 `console` 選項呼叫安裝程式：

```
./bundle-filename -console
```

若要使用指令行介面進行安裝，您必須具有 `root` 許可權。
 - b. 若要安裝預設網域，請取消選取該選項。
只有具有相同架構並具有完全相同的安裝路徑 (即，兩台機器使用相同的 `install-dir` 和 `domain-root-dir`) 的兩台機器才支援備份網域的復原。
- 2 將第一台機器上的備份 ZIP 檔案複製到第三台機器上的 `domain-root-dir` 目錄中。也可以透過 FTP 方式複製檔案。
- 3 執行 `asadmin restore-domain` 指令，以將 ZIP 檔案復原到第三台機器：

```
asadmin restore-domain --filename domain-root-dir/sjsas_backup_v00001.zip domain1
```

可以備份任何網域。但是，在重新建立網域時，網域名稱應與原始網域名稱相同。
- 4 變更第三台機器上的 `domain-root-dir/domain1/generated/tmp` 目錄的權限，以與第一台機器上相同目錄的權限相符。
該目錄的預設許可權爲：`?drwx-----?` (或 700)。
例如：

```
chmod 700 domain-root-dir/domain1/generated/tmp
```

以上範例假定您備份的是 `domain1`。如果備份的是其他名稱的網域，則應使用要備份網域的名稱取代上述的 `domain1`。
- 5 變更第三台機器器的 `domain.xml` 檔案中的主機特性值：
- 6 更新第三台機器上的 `domain-root-dir/domain1/config/domain.xml`。
例如，搜尋 `machine1` 並將其替代爲 `machine3`。這樣，您就可以將：

```
<jmx-connector><property name=client-hostname value=machine1/>...
```

變更爲：

```
<jmx-connector><property name=client-hostname value=machine3/>...
```
- 7 將：

```
<jms-service... host=machine1.../>
```

變更爲：

```
<jms-service... host=machine3.../>
```

8 在 machine3 上啟動復原的網域：

```
asadmin start-domain --user admin-user --password admin-password domain1
```

9 在 machine2 上變更節點代理程式下的 DAS 主機特性值。

10 在 machine2 上變更 `install-dir/nodeagents/nodeagent/agent/config/das.properties` 中的 `agent.das.host` 特性值。

11 在 machine2 上重新啟動節點代理程式。

備註 – 使用 `asadmin start-instance` 指令啟動叢集實例，以使這些實例與復原網域同步。

Application Server 實例

- 第 40 頁的「關於 Application Server 實例」
- 第 42 頁的「關於獨立實例」
- 第 44 頁的「Administration Server 進階設定」
- 第 46 頁的「實例特定的配置特性」
- 第 49 頁的「恢復作業事件」

關於 Application Server 實例

Application Server 在安裝時將建立一個稱爲 server 的 Application Server 實例。如果需要，可以刪除此伺服器實例，並建立一個其他名稱的新實例。

每個 Application Server 實例均有其專有的 J2EE 配置、J2EE 資源、應用程式部署區域以及伺服器配置設定。對一個應用程式伺服器實例所做的變更不會影響其他應用程式伺服器實例。在一個管理網域內，您可以擁有多個應用程式伺服器實例。

對於許多使用者而言，一個應用程式伺服器實例就符合他們的需要了。不過，依據您的環境，您可能想建立一個或多個附加的應用程式伺服器實例。例如，在開發環境下，您可以使用不同的 Application Server 實例來測試不同的 Application Server 配置，或比較和測試不同的應用程式部署。由於您可以輕易地增加或刪除 Application Server 實例，因而可以在開發時透過這些實例建立暫時的「沙箱」區域進行試驗。

此外，對於每個應用程式伺服器實例，您也可以建立虛擬伺服器。在單一安裝的應用程式伺服器實例內，您可以爲公司或個人提供網域名稱、IP 位址以及某些管理功能。

對於使用者而言，看起來好像使用者有自己的 Web 伺服器，但沒有硬體和基本的伺服器維護功能。這些虛擬伺服器不擴充應用程式伺服器實例。如需關於虛擬伺服器的更多資訊，請參閱第 299 頁的「配置 JVM 一般設定」。

在作業部署中，您可以使用虛擬伺服器代替多重應用程式伺服器實例，用於多種目的。但是，如果虛擬伺服器不能滿足需求，您也可以使用多個應用程式伺服器實例。

Application Server 實例不會自動啟動。啟動某個實例後，該實例將一直執行，直至您將其停止。若您停止一個應用程式伺服器實例，該實例將停止接受新連線，然後等待所有未完成的連線完成。如果您的機器當機或離線，伺服器將結束，其正在處理的任何請求均可能遺失。

定義 Application Server 實例

應用程式伺服器實例構成了應用程式部署的基礎。每個實例均屬於單一網域，並有自己的目錄結構、配置和已部署的應用程式。每個伺服器實例還包含了 J2EE 平台的 Web 和 EJB 容器。每個新的伺服器實例必須包含對節點代理程式名稱的參考，該名稱定義實例將要駐留的機器。

可以建立三種類型的伺服器實例。每個伺服器實例只能是其中的一種類型：

- 在**獨立伺服器實例**中，其他任何伺服器實例或叢集不能共用其配置。
- 在**共用伺服器實例**中，其他實例或叢集可以共用其配置。
- 在**叢集伺服器實例**中，叢集中的其他實例可以共用其配置。

叢集是一組共用相同的應用程式集、資源集和配置資訊集的伺服器實例。伺服器實例可以只屬於一個叢集。叢集用於透過在多台機器上分散負載來增強負載平衡，並透過實例層級的容錯移轉來提供高可用性。

圖 1-2 詳細顯示了 Application Server 實例。Application Server 實例是 Application Server Enterprise Edition 的叢集、負載平衡和階段作業持續性功能中的建置區段。

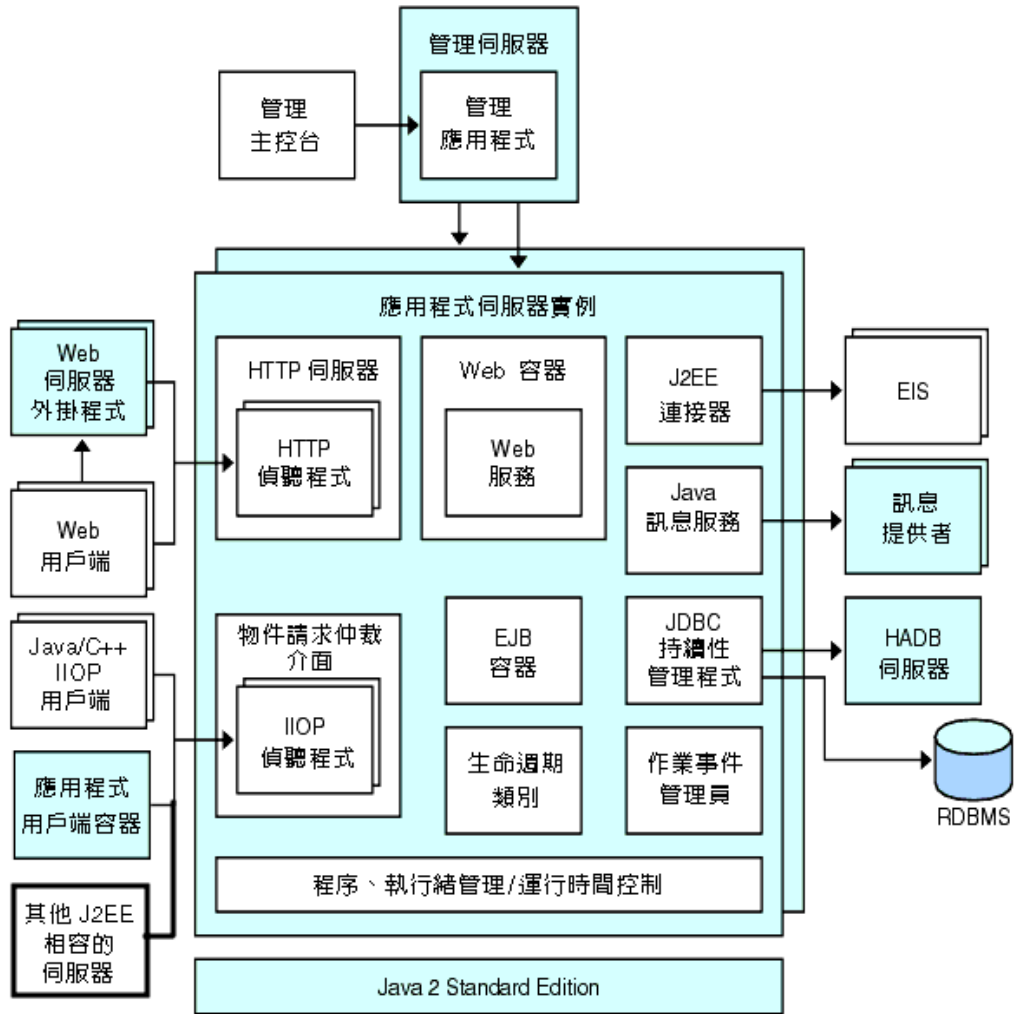


圖 1-2 Application Server 實例

關於獨立實例

Application Server 實例不會自動啟動。啟動某個實例後，該實例將一直執行，直至您將其停止。若您停止一個應用程式伺服器實例，該實例將停止接受新連線，然後等待所有未完成的連線完成。如果您的機器當機或離線，伺服器將結束，其正在處理的任何請求均可能遺失。

另請參閱：

- 第 47 頁的「建立實例」
- 第 48 頁的「啟動實例」
- 第 49 頁的「停止實例」
- 第 49 頁的「恢復作業事件」

檢視一般伺服器資訊

透過 [一般] 標籤可以執行以下作業：

- 按一下 [啟動實例] 以啟動實例。
- 按一下 [停止實例] 以停止實例。
- 按一下 [檢視記錄檔]，以開啓伺服器記錄檢視器。
- 按一下 [自動重建記錄檔]，以自動重建實例的記錄檔。

該動作將排程記錄檔以進行自動重建。實際的自動重建將在下一次向記錄檔寫入項目時發生。預設伺服器 (DAS) 的自動重建將立即發生，但其他獨立伺服器的自動重建將延遲。

- 按一下 [JNDI 瀏覽] 以瀏覽正在執行的實例的 JNDI 樹。
- 按一下 [恢復作業事件] 以恢復未完成的作業事件。

此外，您可以選取以下標籤以執行其他作業：

- [應用程式] 標籤：部署選取的應用程式。
- [資源] 標籤：管理選取的資源。
- [特性] 標籤：配置實例特定的特性。
- [監視] 標籤：檢視 JVM、伺服器、執行緒池、HTTP 服務和作業事件服務的監視資料。
- [進階] 標籤：設定用於部署應用程式的一般特性。

▼ 部署應用程式

通過 [應用程式] 標籤，您可以啓用、停用和部署與實例相關聯的選取應用程式。

- 1 選取所需應用程式的核取方塊。
- 2 從 [部署] 下拉式功能表中，選取要部署的應用程式模組的類型：
 - 企業應用程式：位於 EAR (企業應用程式歸檔) 檔案或目錄中的 J2EE 應用程式。
 - Web 應用程式：封裝在 WAR (Web 應用程式歸檔) 檔案或目錄中的 Web 資源集合 (例如 JavaServer Pages (JSP)、servlet 和 HTML 頁面等)。
 - EJB 模組：包含在 EJB JAR (Java 歸檔) 檔案或目錄中的一個或多個 Enterprise JavaBeans (EJB)。

- 連接器模組：連線至企業資訊系統 (EIS) 並封裝在 RAR (資源配接卡歸檔) 檔案或目錄中。
- 生命週期模組：在伺服器生命週期中有一個或多個事件觸發時執行作業。
- 應用程式用戶端模組：也稱為 J2EE 應用程式用戶端 JAR 檔案，它包含用戶端的伺服器端常式。

▼ 建立新的資源類型

透過 [資源] 標籤，您可以啟用、停用和建立與實例關聯的新資源類型。

- 1 選取所需資源的核取方塊。
- 2 從 [新建] 下拉式功能表中，選取要建立並與該實例相關聯的資源類型：
 - JDBC：為應用程式提供連線至資料庫的方法。
 - 持續性管理員：具有容器管理的持續性 Bean (用於向下相容) 的應用程式需要持續性管理員。
 - JMS 連線工廠：允許應用程式以程式化方式建立其他 JMS 物件的物件。
 - JMS 目標：表示 JavaMail API 中的郵件階段作業，JavaMail API 提供了一個獨立於平台和協定的框架來建置郵件和訊息傳送應用程式。
 - JavaMail：提供了一個獨立於平台和協定的框架來建置郵件和訊息傳送應用程式。
 - 自訂：表示具有已定義的 JNDI 子環境、資源類型和工廠類別的非標準資源。
 - 外部：使應用程式能夠查找位於簡易資料存取協定 (LDAP) 系統儲存庫中的外部資源物件。
 - 連接器：可為應用程式提供企業資訊系統 (EIS) 連線的程式物件。
 - 管理物件：配置 JSR-160 相容的遠端 JMX 連接器。

Administration Server 進階設定

Administration Server 進階設定可讓您設定用於部署應用程式的一般特性。這些特性可讓您確保並監視所部署應用程式的變更已被偵測出並且修改的類別已被重新載入。

設定應用程式配置

如果啟用了動態重新載入，伺服器將定期檢查已部署的應用程式檔案中的變更並自動重新載入包含變更的應用程式。動態重新載入在開發環境中非常有用，因為它能快速測試程式碼變更。但在生產環境中，動態重新載入可能會使效能降低。

動態重新載入旨在用於開發環境。它與階段作業持續性 (一種生產環境功能) 不相容。如果啟用了動態部署，請勿啟用階段作業持續性。

備註 – 動態重新載入僅適用於預設伺服器實例。

若要在 [應用程式配置] 頁面中配置動態重新載入，請配置以下項目：

- 重新載入：使用 [已啟用] 核取方塊來啟用或停用動態重新載入。
- 重新載入輪詢間隔：指定伺服器檢查已部署的應用程式中的變更的頻率。
- 管理階段作業逾時：指定管理階段作業逾時且必須重新登入之前的時間。

▼ 配置自動部署設定

自動部署功能使您能夠透過將預先封裝的應用程式或模組複製到 *domain-dir/autodeploy* 目錄，來部署該應用程式或模組。

例如，將名為 *hello.war* 的檔案複製到 *domain-dir/autodeploy* 目錄。若要取消部署應用程式，請從 *autodeploy* 目錄中移除 *hello.war* 檔案。

自動部署功能旨在用於開發環境。它與階段作業持續性（一種生產環境功能）不相容。如果已啟用自動部署，請勿啟用階段作業持續性。

備註 – 自動部署僅適用於預設伺服器實例。

- 1 請至 [應用程式配置] 頁面。
- 2 透過選取或取消選取 [已啟用] 核取方塊來啟用或停用自動部署。
- 3 在 [自動部署輪詢間隔] 欄位中，指定伺服器檢查自動部署目錄中的應用程式檔案或模組檔案的頻率。
變更輪詢間隔不會影響部署應用程式或模組所需的時間。
- 4 在 [自動部署目錄] 中，如果指定建立應用程式的目錄，則不必將檔案複製到預設自動部署目錄中。
依預設，可使用變數自動變更多個伺服器實例的目錄。
- 5 若要在部署之前執行檢驗器，請選取 [已啟用檢驗器] 核取方塊。
檢驗器將檢查檔案的結構和特性。大型應用程式的檢驗通常會很費時。
- 6 若要預編譯 JSP 頁面，請選取 [JSP] 核取方塊。
如果未選取此核取方塊，則首次存取 JSP 頁面時會在執行階段編譯這些頁面。由於編譯通常很費時，因此在生產環境中請選取此核取方塊。
- 7 按一下 [新增特性] 按鈕以指定其他設定。

設定網域屬性

包括以下網域屬性。

表 1-1 網域屬性值

特性	定義
com.sun.aas.installRoot	應用程式伺服器的安裝目錄。
com.sun.aas.instanceRoot	伺服器實例的頂層目錄。
com.sun.aas.hostName	主機 (機器) 的名稱。
com.sun.aas.javaRoot	..J2SE 安裝目錄。
com.sun.aas.imqLib	Sun Java System Message Queue 軟體的程式庫目錄。
com.sun.aas.configName	伺服器實例正在使用的配置的名稱。
com.sun.aas.instanceName	伺服器實例的名稱。該特性對於 default-config 不可用，但可用於自訂的配置。
com.sun.aas.clusterName	叢集的名稱。僅在叢集的伺服器實例上設定了該特性。該特性對於 default-config 不可用，但可用於自訂的配置。
com.sun.aas.domainName	網域的名稱。該特性對於 default-config 不可用，但可用於自訂的配置。

實例特定的配置特性

實例特定的配置特性將置換此實例的值。

備註 - 預設值定義在與實例關聯的配置中。

▼ 若要將值復原為預設值，請執行以下步驟：

- 1 移除置換值。
- 2 按一下 [儲存]。
如果未設定置換值，則使用預設值。

▼ 增加實例特性

- 按一下 [新增特性] 按鈕以指定其他設定。
可以使用以下屬性名稱/值對來配置資源：

特性	定義
HTTP_LISTENER_PORT	此連接埠用於偵聽 HTTP 請求。此特性指定 http-listener-1 的連接埠值為 1 到 65535。在 UNIX 中，建立在 1 到 1024 連接埠上進行偵聽請求具有超級使用者權限。
HTTP_SSL_LISTENER_PORT	此連接埠用於偵聽 HTTPS 請求。此特性指定 http-listener-2 的連接埠值為 1 到 65535。在 UNIX 中，建立在 1 到 1024 連接埠上進行偵聽請求要求具有超級使用者權限。
IIOF_LISTENER_PORT	此特性指定 orb-listener-1 偵聽 IIOF 連接的 ORB 偵聽程式連接埠。
IIOF_SSL_LISTENER_PORT	此連接埠用於保護 IIOF 連線。
JMX_SYSTEM_CONNECTOR_PORT	此特性指定 JMX 連接器進行偵聽的連接埠號。有效值為 1 到 65535。在 UNIX 中，建立在 1 到 1024 連接埠上進行偵聽的插槽要求具有超級使用者權限。
IIOF_SSL_MUTUALAUTH_PORT	此特性指定稱為 SSL_MUTUALAUTH 的 IIOF 偵聽程式偵聽 IIOF 連接埠。

▼ 刪除特性

- 1 按一下要刪除的特性。
- 2 按一下 [刪除特性] 按鈕。

▼ 建立實例

- 1 在樹形元件中，選取 [獨立實例] 節點。
- 2 在 [獨立伺服器實例] 頁面中，按一下 [新建]。
- 3 在 [名稱] 欄位中，識別新實例的專屬名稱。
- 4 選擇一個節點代理程式。
必須在節點代理程式主機上使用 `asadmin start-node-agent` 指令啟動節點代理程式，以使要建立的伺服器實例能夠與該節點代理程式相關聯。
- 5 選取所需的配置。
 - 參考現有配置。不新增新配置。
 - 建立現有配置的副本。新增伺服器實例或叢集時，將新增新配置。

- 6 若要從其他配置進行複製，請在建立新實例時指定要複製的配置。
依預設，在建立新實例時，其配置是從 default-config 配置中複製的。
- 7 對於伺服器實例，新配置的名稱為 *instance-name-config*。
default-config 配置為預設配置，它用做建立獨立伺服器實例的範本。非叢集伺服器實例或叢集不允許參考 default-config 配置；只能對其進行複製以建立新配置。編輯預設配置，以確定從預設配置複製而來的新配置具有正確的初始設定。

更多資訊 **等效的 asadmin 指令**

create-instance

▼ 啟動實例

- 1 在樹形元件中，展開 [獨立實例] 節點。
- 2 選取要啟動的實例。
- 3 在 [一般] 標籤上，按一下 [啟動實例] 以啟動實例。
必須先使用 `asadmin start-node-agent` 指令啟動與實例相關的節點代理程式，才能成功啟動該實例。
啟動實例後，便可以從 [一般] 標籤上執行以下作業：
 - 按一下 [停止實例] 以停止實例。
 - 按一下 [JNDI 瀏覽] 以檢視該實例的 JNDI 項目。
 - 按一下 [檢視記錄檔] 以檢視 [記錄檢視器] 並指定記錄選項。
 - 按一下 [自動重建記錄檔]。
 - 按一下 [恢復作業事件] 以恢復未完成的作業事件。

更多資訊 **等效的 asadmin 指令**

start-instance

恢復作業事件

由於伺服器當機或資源管理員當機，作業事件可能未完成。完成這些中斷的作業事件並將其從故障中恢復至關重要。Application Server 可在伺服器啟動時從這些故障中回復並完成作業事件。

如果選取的伺服器正在執行，則將由該伺服器執行恢復。如果選取的伺服器未執行，則將由選取的目標伺服器執行恢復。

▼ 停止實例

- 1 在樹形元件中，展開 [獨立實例] 節點。
- 2 選擇要停止的實例。
- 3 在 [一般] 標籤上，按一下 [停止實例] 以停止實例。

更多資訊 **等效的 asadmin 指令**

```
stop-instance
```

▼ 關閉 Administration Server

- 1 在樹形元件中，選取 [獨立實例] 節點。
 - 2 選取 [Administration Server 實例]。
 - 3 按一下 [停止]。
- 螢幕將顯示確認對話方塊，以確認您是否要關閉 Administration Server。

配置變更

- 第 50 頁的「變更 Application Server 配置」
- 第 50 頁的「Application Server 中的連接埠」
- 第 51 頁的「檢視連接埠號」
- 第 51 頁的「變更 Administrative Server 連接埠」
- 第 51 頁的「變更 HTTP 連接埠」
- 第 52 頁的「變更 IIOP 連接埠」
- 第 52 頁的「使用管理服務配置 JMX 連接器」
- 第 53 頁的「編輯 JMX 連接器配置」

- 第 53 頁的「變更 J2SE 軟體」

變更 Application Server 配置

在進行以下任何配置變更時，請重新啓動伺服器以使變更生效：

- 變更 JVM 選項
- 變更連接埠號
- 管理 HTTP 服務、IIOP 服務和 JMS 服務
- 管理執行緒池

如需說明，請參閱第 37 頁的「重新啓動伺服器或網域」。

如果使用動態配置，大多數變更在伺服器執行時即可生效。若要進行以下配置變更，請勿重新啓動伺服器：

- 部署和取消部署應用程式
- 新增或移除 JDBC、JMS 與連接器資源和池
- 變更記錄層級
- 新增檔案範圍使用者
- 變更監視層級
- 啓用和停用資源和應用程式

請注意，`asadmin reconfig` 指令已停用，並且不再需要此指令。配置變更將動態套用到伺服器。

Application Server 中的連接埠

下表說明 Application Server 的連接埠偵聽程式。

表 1-2 使用連接埠的 Application Server 偵聽程式

偵聽程式	預設連接埠號	說明
管理伺服器	4849	可透過管理主控台和 <code>asadmin</code> 公用程式存取網域的管理伺服器。對於管理主控台，請在瀏覽器的 URL 中指定連接埠號。從遠端執行 <code>asadmin</code> 指令時，請使用 <code>--port</code> 選項指定連接埠號。
HTTP	8080	Web 伺服器偵聽連接埠上的 HTTP 請求。若要存取已部署的 Web 應用程式和服務，用戶端應連線到此連接埠。
HTTPS	8181	為安全通訊配置的 Web 應用程式在單獨的連接埠上進行偵聽。
IIOP		企業 Bean (EJB 元件) 的遠端用戶端通過 IIOP 偵聽程式存取 Bean。
IIOP、SSL		另一個連接埠由為安全通訊配置的 IIOP 偵聽程式使用。

表 1-2 使用連接埠的 Application Server 偵聽程式 (續)

偵聽程式	預設連接埠號	說明
IIOP、SSL 和相互認證		另一個連接埠由為相互 (用戶端和伺服器) 認證配置的 IIOP 偵聽程式使用。

▼ 檢視連接埠號

- 1 在樹形元件中，選取 [獨立實例] 節點下的一個實例。
- 2 選取 [特性] 標籤。
- 3 在 [實例特定的] 頁面中，識別了預設連接埠號。可以設定配置以置換這些值。

▼ 變更 Administrative Server 連接埠

- 1 在樹形元件中，展開 [配置] 節點。
- 2 展開 [server-config] (管理配置) 節點。
- 3 展開 [HTTP 服務] 節點。
- 4 展開 [HTTP 偵聽程式] 節點。
- 5 選取 [admin-listener] 節點。
- 6 在 [編輯 HTTP 偵聽程式] 頁面中，變更 [偵聽程式連接埠] 欄位的值。
- 7 重新啟動伺服器。

▼ 變更 HTTP 連接埠

- 1 在樹形元件中，展開 [HTTP 服務] 節點。
- 2 展開 [HTTP 偵聽程式] 節點。
- 3 選取要變更其連接埠號的 HTTP 偵聽程式。
- 4 在 [編輯 HTTP 偵聽程式] 頁面中，變更 [偵聽程式連接埠] 欄位的值。
- 5 按一下 [儲存]。

- 6 重新啟動伺服器。

▼ 變更 IIOP 連接埠

- 1 在樹形元件中，展開 [配置] 節點。
- 2 展開 [server-config] (管理配置) 節點。
- 3 展開 [ORB] 節點。
- 4 展開 [IIOP 偵聽程式] 節點。
- 5 選擇要變更其連接埠號的偵聽程式。
- 6 在 [編輯 IIOP 偵聽程式] 頁面中，變更 [偵聽程式連接埠] 欄位的值。
- 7 按一下 [儲存]。
- 8 重新啟動伺服器。

▼ 使用管理服務配置 JMX 連接器

使用管理服務可以配置 JSR-160 相容遠端 JMX 連接器，該連接器處理網域管理伺服器與遠端伺服器實例的節點代理程式 (管理主機電腦上的伺服器實例) 之間的通訊。

管理服務決定伺服器實例是一般實例、網域管理伺服器 (DAS)，還是兼具兩者。DAS 與 J2EE 伺服器實例類似，只是使用者應用程式和資源不會被部署到 DAS (儘管它能夠處理使用者應用程式請求)。DAS 和 J2EE 伺服器實例之間唯一比較顯著的區別在於：前者不能是叢集的一部分，而是伺服器實例的同質單元。

- 1 從樹中選擇 [配置]。
- 2 選取要配置的實例：
 - a. 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例、伺服器，請選取 [server-config] 節點。
 - b. 若要為將來的實例 (實例使用 default-config 的副本) 配置預設設定，請選取 [default-config] 節點。
- 3 從樹中選取 [管理服務]。

- 4 從 [類型] 下拉式功能表中，選取要管理服務配置的類型：DAS、DAS 和伺服器或者伺服器。選取 [DAS 和伺服器] 與選取 [DAS] 作用相同。[伺服器] 選項可選取非 DAS 伺服器實例。
- 5 在 [JMX 連接器名稱] 欄位中，輸入內部使用的 JMX 連接器的名稱。連接器的名稱為 system。

▼ 編輯 JMX 連接器配置

透過 [編輯 JMX 連接器] 螢幕，您可以編輯 JSR 160 相容的 JMX 連接器的配置。

- 1 從樹中選擇 [配置]。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例、伺服器，請選取 [server-config] 節點。
 - 若要為將來的實例 (實例使用 default-config 的副本) 配置預設設定，請選取 [default-config] 節點。
- 3 展開 [管理服務] 節點並按一下 [system]，它是內部使用的 JMX 連接器。
- 4 輸入 JMX 連接器伺服器的連接埠。
JMX 服務 URL 是協定、連接埠和位址的函式，如 JSR 160 1.0 規格所定義。
- 5 輸入此 JMX 連接器應該支援的協定。
Application Server 8.1 版僅支援 rmi_jrmp 協定。
- 6 在 [範圍名稱] 欄位中，輸入表示特殊管理範圍的名稱。
所有認證均依此範圍處理。
- 7 選取 [已啟用] 核取方塊以表示 JMX 連接器中應當使用傳輸層安全性。

變更 J2SE 軟體

Application Server 依賴於 Java 2 Standard Edition (J2SE™) 軟體。安裝 Application Server 時，已指定 J2SE 軟體的目錄。如需有關變更 J2SE 軟體的說明，請參閱第 299 頁的「[配置 JVM 一般設定](#)」。

使用線上說明

管理主控台的線上說明是上下文相關的：按一下右上角的 [說明] 連結後，說明瀏覽器視窗會顯示與目前管理主控台頁面相關的主題。如果目前頁面沒有說明資訊，則顯示 [使用線上說明] 主題。

線上說明包含了上下文有關的概念主題。若要檢視這些主題中的某一個主題，請從說明瀏覽器視窗的目錄中選取該主題。

▼ 返回至上一個說明螢幕

- 1 在說明瀏覽器視窗中，按一下滑鼠右鍵以顯示選取功能表。
- 2 選擇 [後退]。

詳細資訊

- Sun Microsystems 全球訓練 — Sun 及其授權中心每年透過基於 Web 的課程和遍佈 60 多個國家/地區的 250 多個訓練站點，訓練超過 250,000 名學員。如需更多資訊，請參閱：<http://training.sun.com/>。
- 「The J2EE 1.4 Tutorial」- 該指導文件專為開發者編寫，其中說明了配置 JMS、設定 JavaMail 資源和管理安全性的管理說明。若要存取此指導文件，請至：<http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html>。
- 「Application Server Developer's Guide」- 該指南包含特定於 Application Server 的開發資訊。
- asadmin 線上手冊 - 以 HTML 格式提供，這些線上手冊包含所有 Application Server 公用程式 (包括 asadmin 公用程式指令) 的語法和範例。
- *Application Server* 版本說明
- docs.sun.com：Sun 產品文件 - 從該站台您可以搜尋和存取我們產品的所有文件：<http://docs.sun.com/>。
- J2EE 1.4 Documentation 頁面 - 位於我們的公共網站上，該頁面具有指向 J2EE 1.4 平台的技術文件的連結：<http://java.sun.com/j2ee/1.4/docs/>。
- 「快速入門指南」- 此文件說明如何部署並執行簡單的 Web 應用程式。該指南位於 *install-dir/docs/QuickStart.html* 檔案中。

部署應用程式

本章說明如何在 Application Server 上部署 (安裝) J2EE 應用程式。它包含以下小節：

- 第 55 頁的「關於部署」
- 第 58 頁的「有關部署應用程式的管理主控台作業」
- 第 68 頁的「有關列示、取消部署以及啓用應用程式的管理主控台作業」
- 第 73 頁的「適用於開發者的部署方法」

關於部署

- 第 55 頁的「部署生命週期」
- 第 56 頁的「J2EE 歸檔檔案的類型」
- 第 57 頁的「命名慣例」

部署生命週期

安裝 Application Server 並啓動網域之後，即可部署 (安裝) J2EE 應用程式和模組。在部署期間和變更應用程式時，應用程式或模組可能會經過以下階段：

1. 初始部署

部署應用程式或模組之前，請啓動網域。

將應用程式或模組部署 (安裝) 到特定的獨立伺服器實例或叢集。由於應用程式和模組封裝在歸檔檔案中，因此在部署期間應指定歸檔檔案名稱。預設為部署到預設伺服器實例 server。

如果部署到伺服器實例或叢集，則應用程式或模組將存在於網域的中央儲存庫中，並由部署到的所有叢集或伺服器實例做為目標參照。

您還可以使用 `asadmin deploy` 指令 (而非管理主控台) 部署到網域。如果將應用程式或模組僅部署到網域，則應用程式或模組將存在於網域的中央儲存庫中，但要在增加參照之後才會有伺服器實例或叢集參照該應用程式或模組 (如第 55 頁的「部署生命週期」中所述)。

部署是動態的：部署應用程式或模組後，無需重新啟動伺服器實例即可使用應用程式。如果重新啟動了伺服器實例，所有已部署的應用程式和模組仍將處於部署狀態並且可用。

2. 啟用或停用

依預設，將啟用已部署的應用程式或模組，這表示如果應用程式或模組已部署到可存取的伺服器實例或叢集，則可以執行該應用程式或模組並且可由用戶端對其進行存取。若要防止存取，請停用應用程式或模組。在部署之後，已停用的應用程式或模組並未從網域中解除安裝，而且可以輕鬆地將其啟用。

3. 新增或刪除已部署應用程式或模組的目標

部署後，應用程式或模組將存在於中央儲存庫中，並可由多個伺服器實例和/或叢集參照。最初，做為目標部署到的伺服器實例或叢集將參考應用程式或模組。

在部署應用程式或模組之後，若要變更參照應用程式或模組的伺服器實例和叢集，請使用 管理主控台 變更應用程式或模組的目標，或使用 `asadmin` 工具變更應用程式參照。由於應用程式本身儲存在中央儲存庫中，因此新增或刪除目標將新增或刪除不同目標上同一版本的應用程式。但是，可以在一個目標上啟用而在另一個目標上停用部署到多個目標的應用程式，因此即使應用程式被一個目標參考，也只有在該目標上啟用它時使用者才能對其進行使用。

4. 重新部署

若要替代已部署的應用程式或模組，請將其重新部署。重新部署將自動取消部署之前已部署的應用程式或模組，並代之以新應用程式或模組。

當透過 管理主控台 重新部署時，重新部署的應用程式或模組將部署到網域中，並且所有參照該應用程式或模組的獨立或叢集伺服器實例將自動接收新的版本(如果已啟用動態重新配置)。如果使用 `asadmin deploy` 指令來重新部署，請將 `domain` 指定為目標。

對於生產環境，請使用捲動升級(升級應用程式而不中斷服務)。如需更多資訊，請參閱「關於滾動升級。」

5. 取消部署

若要解除安裝應用程式或模組，請取消部署應用程式或模組。

J2EE 歸檔檔案的類型

軟體供應商將應用程式或模組封裝在歸檔檔案中。若要部署應用程式或模組，請指定歸檔檔案名稱。歸檔檔案的內容和結構是按照 J2EE 平台的規格定義的。J2EE 歸檔檔案的類型有以下幾種：

- Web 應用程式歸檔 (WAR)：WAR 檔案由 Servlet 和 JSP 等 Web 元件以及靜態 HTML 頁面、JAR 檔案、標籤檔案庫和公用程式類別組成。WAR 檔案名稱具有 `.war` 副檔名。

- EJB JAR：EJB JAR 檔案包含一個或多個企業 Bean (用於 EJB 技術的元件)。EJB JAR 檔案還包括企業 Bean 所需的任何公用程式類別。EJB JAR 檔案的名稱具有 .jar 副檔名。
- J2EE 應用程式用戶端 JAR：該 JAR 檔案包含藉由 RMI/IIOP 存取伺服器端元件 (如企業 Bean) 的 J2EE 應用程式用戶端的程式碼。在管理主控台中，J2EE 應用程式用戶端被稱為「應用程式用戶端。」J2EE 應用程式用戶端 JAR 檔案的名稱具有 .jar 副檔名。
- 資源介面歸檔 (RAR)：RAR 檔案存有資源介面。資源介面是按照 J2EE 連接器架構規格定義的，它是允許企業 Bean 和 Web 元件和應用程式用戶端存取資源和外部企業系統的可攜式元件。資源介面經常稱為連接器。RAR 檔案名稱具有 .rar 副檔名。
- 企業應用程式歸檔 (EAR)：EAR 檔案存有一個或多個 WAR 檔案、EJB JAR 檔案、RAR 檔案或 J2EE 應用程式用戶端 JAR 檔案。EAR 檔案名稱具有 .ear 副檔名。

軟體供應商可以將應用程式組譯為單一 EAR 檔案或多個獨立的 WAR 檔案、EJB JAR 檔案和應用程式用戶端 JAR 檔案。在管理工具中，用於所有類型檔案的部署頁面和指令都是類似的。

命名慣例

在給定網域中，已部署的應用程式名稱和模組名稱必須是專屬名稱。

- 如果使用管理主控台進行部署，請在 [應用程式名稱] 欄位中指定名稱。
- 如果使用 `asadmin deploy` 指令進行部署，則應用程式或模組的預設名稱為要部署的 JAR 檔案的前綴。例如，如果部署 `hello.war` 檔案，則 Web 應用程式的名稱為 `hello`。若要置換預設名稱，請指定 `--name` 選項。

在一個應用程式中，不同類型的模組可以具有相同的名稱。部署應用程式時，將使用 `_jar`、`_war` 和 `_rar` 後綴來命名儲存個別模組的目錄。一個應用程式內，類型相同的模組必須具有專屬名稱。此外，在一個應用程式內，資料庫綱目檔的名稱必須是專屬名稱。

建議將類似於 Java 套裝軟體的命名機制用於在 `ejb-jar.xml` 檔案的 `<module-name>` 部分中找到的模組檔案名稱、EAR 檔案名稱、模組名稱，以及在 `ejb-jar.xml` 檔案的 `<ejb-name>` 部分找到的 EJB 名稱。使用這種類似於套裝軟體的命名機制可以確保不會發生名稱衝突。該命名慣例的優勢不僅適用於 Application Server，也適用於其他 J2EE Application Server。

EJB 元件的 JNDI 查詢名稱也必須是專屬名稱。建立連續的命名慣例可能會非常有用。例如，將應用程式名稱和模組名稱附加到 EJB 名稱中是保證名稱為專屬名稱的方式。在這種情況下，`mycompany.pkging.pkgingEJB.MyEJB` 即為模組 `pkgingEJB.jar` 內 EJB 的 JNDI 名稱，該模組封裝於應用程式 `pkging.ear` 中。

請確定該套裝軟體和檔案的名稱不含有空格或作業系統不支援的非法字元。

有關部署應用程式的管理主控台作業

- 第 58 頁的「部署企業應用程式」
- 第 60 頁的「部署 Web 應用程式」
- 第 62 頁的「啟動已部署的 Web 應用程式」
- 第 62 頁的「部署 EJB 模組」
- 第 66 頁的「部署應用程式用戶端模組」
- 第 65 頁的「建立生命週期模組」
- 第 65 頁的「建立生命週期模組」
- 第 66 頁的「部署應用程式用戶端模組」

▼ 部署企業應用程式

企業應用程式封裝於 EAR 檔案中，EAR 檔案是包含任意 J2EE 獨立模組 (例如 WAR 檔案和 EJB JAR 檔案) 的一種歸檔檔案。

- 1 在樹形元件中，展開 [應用程式] 節點。
- 2 選取 [企業應用程式] 節點。
- 3 在 [企業應用程式] 頁面中，按一下 [部署]。
- 4 在 [部署] 頁面中，指定要部署的 EAR 檔案的位置。

伺服器是執行應用程式伺服器和網域管理伺服器的主機。用戶端機器是可以用於透過瀏覽器來檢視管理主控台的主機。

- 當檔案常駐於用戶端機器上或可從用戶端機器存取時，按一下單選按鈕可以指定要上傳到 Application Server 的套裝軟體檔案。

按一下 [瀏覽] 以找到檔案，或鍵入檔案的完整路徑。

- 如果檔案位於伺服器上，或者要從展開的目錄部署未封裝的應用程式，則按一下單選按鈕以指定必須可以從伺服器上存取的套裝軟體檔案或目錄路徑。

鍵入檔案或目錄的完整路徑名稱。從展開的目錄部署適用於進階開發者，但不建議在生產環境下進行此操作。

- 5 按 [下一步]，以顯示 [部署企業應用程式] 頁面。
- 6 在 [部署企業應用程式] 頁面中，請指定應用程式的設定。

- a. 在 [應用程式名稱] 欄位中，既可以保留預設名稱 (即檔案名稱的字首)，也可以鍵入其他名稱。

如果您選擇上傳檔案，則會顯示預設名稱。應用程式名稱必須是一專屬名稱。

- b. 若要在部署之後停用應用程式以使其不可用，請選取 [已停用] 單選按鈕。
依預設，部署應用程式之後即可使用該應用程式。
 - c. 如果已部署應用程式，請選取 [重新部署] 核取方塊以對其進行重新部署；否則將顯示錯誤。
您也可以選擇其他應用程式名稱，並以新名稱對其進行部署。
 - d. 若要在部署之前檢驗檔案的結構和內容，請選取 [檢驗器] 核取方塊。
大型應用程式的檢驗會很費時。如果懷疑檔案已毀壞或為不可攜式，請檢驗檔案。
 - e. 若要預編譯 JSP 頁面，請選取 [JSP] 核取方塊。
如果未選取此核取方塊，則首次存取 JSP 頁面時會在執行階段編譯這些頁面。由於編譯通常很費時，因此在生產環境中請選取此核取方塊。
 - f. 選擇高可用性設定。
若要啓用應用程式的高可用性，請選取 [可用性] 核取方塊。如果啓用了應用程式的可用性，則必須也在所有更高層級 (指配置和 Web 容器或 EJB 容器) 啓用可用性。
 - g. 選擇要將應用程式部署到的目標。
從可用目標的清單中選擇目標，並按一下 [新增]。目標可以是叢集或獨立伺服器實例。如果不選取目標，則該應用程式將部署到預設伺服器實例 server。
如果您要重新部署，請勿選取目標。此時您所作的任何選取都將被忽略。參考已部署的應用程式的所有目標叢集或獨立伺服器實例都將自動參考新的、重新部署的應用程式 (如果已啓用叢集或獨立實例的動態重新配置)。如需有關如何在不中斷服務的情況下重新部署應用程式的更多資訊，請參閱「升級應用程式。」
 - h. 選擇是否產生 RMI 存根。
如果選擇產生 RMI 存根，則會產生靜態 RMI-IIOP 存根並將其儲存到用戶端 JAR 檔案。
- 7 按一下 [確定] 以部署應用程式。

更多資訊 等效的 asadmin 指令

deploy

▼ 編輯已部署的企業應用程式

- 1 在樹形元件中，展開 [應用程式] 節點。
- 2 展開 [企業應用程式] 節點。
- 3 選取已部署的應用程式的節點。
- 4 在 [企業應用程式] 頁面中，變更描述。
- 5 在企業版中，啓用或停用高可用性。

如果啓用了應用程式的可用性，則必須也在所有更高層級 (指配置和 Web 容器或 EJB 容器) 啓用可用性。

▼ 部署 Web 應用程式

Web 應用程式封裝於 WAR 檔案中，WAR 檔案是包含元件 (例如 servlet 和 JSP 檔案) 的一種歸檔檔案。

- 1 在樹形元件中，展開 [應用程式] 節點。
- 2 選取 [Web 應用程式] 節點。
- 3 在 [Web 應用程式] 頁面中，按一下 [部署]。
- 4 在 [部署] 頁面中，指定要部署的 WAR 檔案的位置。

伺服器是執行應用程式伺服器和網域管理伺服器的主機。用戶端機器是可以用於透過瀏覽器來檢視管理主控台的主機。

- 當檔案常駐於用戶端機器上或可從用戶端機器存取時，按一下單選按鈕可以指定要上傳到 Application Server 的套裝軟體檔案。
按一下 [瀏覽] 以找到檔案，或鍵入檔案的完整路徑。
 - 如果檔案位於伺服器上，或者要從展開的目錄部署未封裝的應用程式，則按一下單選按鈕以指定必須可以從伺服器上存取的套裝軟體檔案或目錄路徑。
鍵入檔案或目錄的完整路徑名稱。從展開的目錄部署適用於進階開發者，但不建議在生產環境下進行此操作。
- 5 按一下 [下一步] 以顯示 [部署 Web 應用程式] 頁面。

- 6 在 [部署 Web 應用程式] 頁面中，請指定應用程式的設定。
 - a. 在 [應用程式名稱] 欄位中，既可以保留預設名稱 (即檔案名稱的字首)，也可以鍵入其他名稱。

如果您選擇上傳檔案，則會顯示預設名稱。應用程式名稱必須是一專屬名稱。
 - b. 在 [環境根] 欄位中，輸入識別 Web 應用程式的字串。

在 Web 應用程式的 URL 中，環境根目錄要緊跟連接埠號 (http://host:port/context-root/...)。確定環境根以正斜線開頭，例如：/hello。
 - c. 若要在部署之後停用應用程式以使其不可用，請選取 [已停用] 單選按鈕。

依預設，部署應用程式之後即可使用該應用程式。
 - d. 如果已部署應用程式，請選取 [重新部署] 核取方塊以對其進行重新部署；否則將顯示錯誤。

您也可以選擇其他應用程式名稱，並以新名稱對其進行部署。
 - e. 若要在部署之前檢驗檔案的結構和內容，請選取 [檢驗器] 核取方塊。

大型應用程式的檢驗通常會很費時。如果懷疑檔案已毀壞或為不可攜式，請檢驗檔案。
 - f. 若要預編譯 JSP 頁面，請選取 [JSP] 核取方塊。

如果未選取此核取方塊，則首次存取 JSP 頁面時會在執行階段編譯這些頁面。由於編譯通常很費時，因此在生產環境中請選取此核取方塊。
 - g. 選擇高可用性設定。

若要啟用應用程式的高可用性，請選取 [可用性] 核取方塊。如果啟用了應用程式的可用性，則必須也在所有更高層級 (指配置和 Web 容器或 EJB 容器) 啟用可用性。
 - h. 選擇要將應用程式部署到的目標。

從可用目標的清單中選擇目標，並按一下 [新增]。目標可以是叢集或獨立伺服器實例。如果不選取目標，則該應用程式將部署到預設伺服器實例 server。

如果您要重新部署，請勿選取目標。此時您所作的任何選取都將被忽略。參考已部署的應用程式的所有目標叢集或獨立伺服器實例都將自動參考新的、重新部署的應用程式 (如果已啟用叢集或獨立實例的動態重新配置)。如需有關如何在不中斷服務的情況下重新部署應用程式的更多資訊，請參閱「關於滾動升級。」
 - i. 選擇是否產生 RMI 存根。

如果選擇產生 RMI 存根，則會產生靜態 RMI-IIOP 存根並將其儲存到用戶端 JAR 檔案。

- 7 按一下 [確定] 以部署應用程式。

更多資訊 等效的 asadmin 指令

deploy

▼ 啓動已部署的 Web 應用程式

部署應用程式之後，您可以從管理主控台啓動該應用程式。伺服器 and HTTP 偵聽程式必須正在執行，應用程式才能啓動。

- 1 在樹形元件中，展開 [應用程式] 節點。
- 2 按一下 [Web 應用程式]。
- 3 按一下 Web 應用程式的 [啓動] 連結。
- 4 在 [Web 應用程式連結] 頁面中按一下連結以啓動應用程式。

▼ 部署 EJB 模組

EJB 模組也稱爲 EJB JAR 檔案，它包含企業 Bean。

- 1 在樹形元件中，展開 [應用程式] 節點。
- 2 選取 [EJB 模組] 節點。
- 3 在 [EJB 模組] 頁面中，按一下 [部署]。
- 4 在 [部署] 頁面中，指定要部署的 JAR 檔案的位置。
伺服器是執行應用程式伺服器和網域管理伺服器的主機。用戶端機器是可以用於透過瀏覽器來檢視管理主控台的主機。
 - 當檔案常駐於用戶端機器上或可從用戶端機器存取時，按一下單選按鈕可以指定要上傳到 Application Server 的套裝軟體檔案。
按一下 [瀏覽] 以找到檔案，或鍵入檔案的完整路徑。
 - 如果檔案位於伺服器上，或者要從展開的目錄部署未封裝的應用程式，則按一下單選按鈕以指定必須可以從伺服器上存取的套裝軟體檔案或目錄路徑。
鍵入檔案或目錄的完整路徑名稱。從展開的目錄部署適用於進階開發者，但不建議在生產環境下進行此操作。
- 5 按 [下一步] 以顯示 [部署 EJB 模組] 頁面。

- 6 在 [部署 EJB 模組] 頁面中，請指定模組的設定。
 - a. 在 [應用程式名稱] 欄位中，既可以保留預設名稱 (即檔案名稱的字首)，也可以鍵入其他名稱。

如果您選擇上傳檔案，則會顯示預設名稱。應用程式名稱必須是一專屬名稱。
 - b. 若要在部署之後停用模組以使其不可用，請選取 [已停用] 單選按鈕。

依預設，模組部署之後即可使用。
 - c. 如果已部署模組，請選取 [重新部署] 核取方塊以對其進行重新部署；否則將顯示錯誤。

您也可以選擇其他應用程式名稱，並以新名稱對其進行部署。
 - d. 若要在部署之前檢驗檔案的結構和內容，請選取 [檢驗器] 核取方塊。

大型應用程式的檢驗會很費時。如果懷疑檔案已毀壞或為不可攜式，請檢驗檔案。
 - e. 選擇高可用性設定。

若要啓用模組的高可用性，請選取 [可用性] 核取方塊。如果啓用了模組的可用性，則必須也在所有更高層級 (指配置和 Web 容器或 EJB 容器) 啓用可用性。
 - f. 選擇要將模組部署到的目標。

從可用目標的清單中選擇目標，並按一下 [新增]。目標可以是叢集或獨立伺服器實例。如果不選取目標，則該模組將部署到預設伺服器實例 `server`。

如果您要重新部署，請勿選取目標。此時您所作的任何選取都將被忽略。參考已部署的模組的所有目標叢集或獨立伺服器實例都將自動參考新的、重新部署的模組 (如果已啓用叢集或獨立實例的動態重新配置)。如需有關如何在不中斷服務的情況下重新部署模組的更多資訊，請參閱「關於滾動升級。」
 - g. 選擇是否產生 RMI 存根。

如果選擇產生 RMI 存根，則會產生靜態 RMI-IIOP 存根並將其儲存到用戶端 JAR 檔案。
- 7 按一下 [確定] 以部署模組。

更多資訊 等效的 asadmin 指令

deploy

▼ 部署連接器模組

連接器也稱為資源介面，它封裝於一種稱為 RAR 檔案的歸檔檔案中。

1 在樹形元件中，展開 [應用程式] 節點。

2 選取 [連接器模組] 節點。

3 在 [連接器模組] 頁面中，按一下 [部署]。

4 在 [部署] 頁面中，指定要部署的 RAR 檔案的位置。

伺服器是執行應用程式伺服器和網域管理伺服器的主機。用戶端機器是可以用於透過瀏覽器來檢視管理主控台的主機。

- 當檔案常駐於用戶端機器上或可從用戶端機器存取時，按一下單選按鈕可以指定要上傳到 Application Server 的套裝軟體檔案。

按一下 [瀏覽] 以找到檔案，或鍵入檔案的完整路徑。

- 如果檔案位於伺服器上，或者要從展開的目錄部署未封裝的模組，則按一下單選按鈕以指定必須可以從伺服器上存取的套裝軟體檔案或目錄路徑。

鍵入檔案或目錄的完整路徑名稱。從展開的目錄部署適用於進階開發者，但不建議在生產環境下進行此操作。

5 按 [下一步] 以顯示 [部署連接器模組] 頁面。

6 在 [部署連接器模組] 頁面中，指定模組的設定。

- a. 在 [應用程式名稱] 欄位中，既可以保留預設名稱 (即檔案名稱的字首)，也可以鍵入其他名稱。

如果您選擇上傳檔案，則會顯示預設名稱。應用程式名稱必須是一專屬名稱。

- b. 在 [執行緒池 ID] 欄位中，為要部署的資源介面指定執行緒池。

依預設，Application Server 處理其預設執行緒池中所有資源介面的工作請求。使用該欄位可以關聯特定使用者建立的執行緒池以處理資源介面的工作請求。

- c. 若要在部署之後停用模組以使其不可用，請選取 [已停用] 單選按鈕。

依預設，模組部署之後即可使用。

啟用或停用連接器模組時，您也同時啟用或停用了指向該模組的連接器資源和連線池。

- d. 如果已部署模組，請選取 [重新部署] 核取方塊以對其進行重新部署；否則將顯示錯誤。

您也可以選擇其他應用程式名稱，並以新名稱對其進行部署。

- e. 若要在部署之前檢驗檔案的結構和內容，請選取 [檢驗器] 核取方塊。
大型應用程式的檢驗通常會很費時。如果懷疑檔案已毀壞或為不可攜式，請檢驗檔案。
- f. 如果資源介面被指定了附加特性，將顯示這些特性。
使用此表可以修改這些特性的預設值。
- g. 選擇要將模組部署到的目標。
從可用目標的清單中選擇目標，並按一下 [新增]。目標可以是叢集或獨立伺服器實例。如果不選取目標，則該模組將部署到預設伺服器實例 server。
如果您要重新部署，請勿選取目標。此時您所作的任何選取都將被忽略。參考已部署的模組的所有目標叢集或獨立伺服器實例都將自動參考新的、重新部署的模組 (如果已啟用叢集或獨立實例的動態重新配置)。如需有關如何在不中斷服務的情況下重新部署模組的更多資訊，請參閱「關於滾動升級。」

- 7 按一下 [確定] 以部署模組。

更多資訊 等效的 asadmin 指令

deploy

▼ 建立生命週期模組

當伺服器生命週期中的一個或多個事件觸發生命週期模組時，該模組會執行作業。這些伺服器事件包含：

- 初始化
- 啟動
- 服務請求準備就緒
- 關閉

生命週期模組不屬於 J2EE 規格，但該模組是 Application Server 的增強功能。

- 1 在樹形元件中，展開 [應用程式] 節點。
- 2 選取 [生命週期模組] 節點。
- 3 在 [生命週期模組] 頁面中，按一下 [新建]。
- 4 在 [建立生命週期模組] 頁面中，指定以下設定：
 - a. 在 [名稱] 欄位中，鍵入指定模組功能的名稱。

- b. 在 [類別名稱] 欄位中，鍵入生命週期模組之類別檔案的完全合格名稱。
 - c. 如果包含生命週期的 JAR 檔案位於伺服器的類別路徑中，請將 [類別名稱] 欄位保留為空白。否則，鍵入完全合格的路徑。
如果不指定類別路徑，則必須在 `domain-dir/applications/lifecycle-module/module-name` 中解壓縮類別。如果指定類別路徑，則無需其他操作。
 - d. 在 [載入順序] 欄位中，鍵入大於 100 且小於作業系統的 MAXINT 值的整數。
該整數將確定伺服器啟動時載入生命週期模組的次序。具有較小整數的模組可以更快地載入。
 - e. 啟動伺服器時，伺服器將載入已部署的生命週期模組。
依預設，如果載入失敗，伺服器仍將繼續執行啟動作業。若要在載入失敗時防止伺服器啟動，請選取 [載入失敗時] 核取方塊。
 - f. 若要在部署之後停用模組以使其不可用，請選取 [已停用] 單選按鈕。
由於在伺服器啟動時會呼叫生命週期模組，因此只有伺服器實例重新啟動後，被停用的生命週期模組才會停止作業。
 - g. 選擇要將模組部署到的目標。
從可用目標的清單中選擇目標，並按一下 [新增]。目標可以是叢集或獨立伺服器實例。如果不選取目標，則該模組將部署到預設伺服器實例 `server`。
- 5 按一下 [確定]。

更多資訊 等效的 asadmin 指令

```
create-lifecycle-module
```

▼ 部署應用程式用戶端模組

應用程式用戶端模組也稱為 J2EE 應用程式用戶端 JAR 檔案，它包含用戶端的伺服器端常式。

- 1 在樹形元件中，展開 [應用程式] 節點。
- 2 選取 [應用程式用戶端模組] 節點。
- 3 在 [應用程式用戶端模組] 頁面中，按一下 [部署]。

4 在 [部署] 頁面中，指定要部署的 JAR 檔案的位置。

伺服器是執行應用程式伺服器和網域管理伺服器的主機。用戶端機器是可以用於透過瀏覽器來檢視管理主控台的主機。

- 當檔案常駐於用戶端機器上或可從用戶端機器存取時，按一下單選按鈕可以指定要上傳到 Application Server 的套裝軟體檔案。
按一下 [瀏覽] 以找到檔案，或鍵入檔案的完整路徑。
- 如果檔案位於伺服器上，或者要從展開的目錄部署未封裝的模組，則按一下單選按鈕以指定必須可以從伺服器上存取的套裝軟體檔案或目錄路徑。
鍵入檔案或目錄的完整路徑名稱。從展開的目錄部署適用於進階開發者，但不建議在生產環境下進行此操作。

5 按 [下一步] 以顯示 [部署應用程式用戶端模組] 頁面。

6 在 [部署應用程式用戶端模組] 頁面中，請指定模組的設定。

- a. 在 [應用程式名稱] 欄位中，既可以保留預設名稱 (即檔案名稱的字首)，也可以鍵入其他名稱。
如果您選擇上傳檔案，則會顯示預設名稱。應用程式名稱必須是一專屬名稱。
- b. 如果已部署模組，請選取 [重新部署] 核取方塊以對其進行重新部署；否則將顯示錯誤。
您也可以選擇其他應用程式名稱，並以新名稱對其進行部署。
- c. 若要在部署之前檢驗檔案的結構和內容，請選取 [檢驗器] 核取方塊。
大型應用程式的檢驗會很費時。如果懷疑檔案已毀壞或為不可攜式，請檢驗檔案。
- d. 選擇要將模組部署到的目標。
從可用目標的清單中選擇目標，並按一下 [新增]。目標可以是叢集或獨立伺服器實例。如果不選取目標，則該模組將部署到預設伺服器實例 server。
如果您要重新部署，請勿選取目標。此時您所作的任何選取都將被忽略。參考已部署的模組的所有目標叢集或獨立伺服器實例都將自動參考新的、重新部署的模組 (如果已啟用叢集或獨立實例的動態重新配置)。如需有關如何在不中斷服務的情況下重新部署模組的更多資訊，請參閱「關於滾動升級。」
- e. 選擇是否產生 RMI 存根。
如果選擇產生 RMI 存根，則會產生靜態 RMI-IIOP 存根並將其儲存到用戶端 JAR 檔案。
對於用戶端常式：
 - 通常，應用程式供應商會發行包含用戶端常式的 JAR 檔案。

- 應用程式提供者透過指定 `asadmin deploy` 指令的 `--retrieve` 選項來取得用戶端存根。

7 按一下 [確定] 以部署模組。

更多資訊 等效的 `asadmin` 指令

`deploy`

▼ 指定用於部署的歸檔檔案

在應用程式或模組頁面上按一下 [部署]，以存取 [部署] 頁面。在 [部署] 頁面中，指定封裝應用程式或模組的歸檔檔案的位置。

伺服器是執行應用程式伺服器和網域管理伺服器的主機。用戶端機器是可以用於透過瀏覽器來檢視 管理主控台的主機。

1 當檔案常駐於用戶端機器上或可從用戶端機器存取時，按一下單選按鈕可以指定要上傳到 **Application Server** 的套裝軟體檔案。

按一下 [瀏覽] 以找到檔案，或鍵入檔案的完整路徑。

2 如果檔案位於伺服器上，或者要從展開的目錄部署未封裝的應用程式，則按一下單選按鈕以指定必須可以從伺服器上存取的套裝軟體檔案或目錄路徑。

鍵入檔案或目錄的完整路徑名稱。從展開的目錄部署適用於進階開發者，但不建議在生產環境下進行此操作。

有關列示、取消部署以及啓用應用程式的管理主控台作業

- 第 68 頁的「列示已部署的應用程式」
- 第 69 頁的「列示子元件」
- 第 69 頁的「檢視已部署的應用程式的模組描述元」
- 第 70 頁的「取消部署應用程式或模組」
- 第 70 頁的「啓用或停用應用程式或模組」
- 第 72 頁的「配置動態重新載入」

▼ 列示已部署的應用程式

1 在樹形元件中，展開 [應用程式] 節點。

2 展開應用程式或模組類型的節點。

若要檢視已部署的應用程式或模組的詳細資訊，請執行以下操作之一：

- 在樹形元件中，選取應用程式或模組的節點。
- 在頁面中選取 [應用程式名稱] 欄中的項目。

更多資訊 等效的 asadmin 指令

```
list-components
```

▼ 列示子元件

企業應用程式、Web 應用程式、EJB 模組和連接器模組都包含子元件。例如，Web 應用程式可能包含一個或多個 Servlet。

- 1 在樹形元件中，展開 [應用程式] 節點。
- 2 展開要檢視其描述元的應用程式或模組類型的節點。
- 3 選取已部署的應用程式或模組的節點。
- 4 在 [應用程式或模組] 頁面中，檢視 [子元件] 表的内容。

更多資訊 等效的 asadmin 指令

```
list-sub-components
```

▼ 檢視已部署的應用程式的模組描述元

對於企業應用程式、Web 應用程式、EJB 模組、連接器模組和應用程式用戶端模組，您可以檢視模組部署描述元。

- 1 在樹形元件中，展開 [應用程式] 節點。
- 2 選取要檢視其描述元的應用程式或模組類型的節點。
- 3 選取已部署的應用程式或模組的節點。
- 4 選取 [描述元] 標籤。
- 5 若要查看描述元檔案的文字，請按一下檔案名稱。頁面將顯示檔案內容。此資訊是唯讀的。

▼ 取消部署應用程式或模組

取消部署應用程式或模組將從網域中將其解除安裝並移除所有實例對它的參考。

- 1 在樹形元件中，展開 [應用程式] 節點。
- 2 選取要取消部署的應用程式或模組的類型的節點。
- 3 在列示已部署的應用程式的表中，選取要取消部署的應用程式或模組的核取方塊。
- 4 按一下 [取消部署]。

更多資訊 **等效的 asadmin 指令**

`undeploy`

▼ 啓用或停用應用程式或模組

啓用已部署的應用程式或模組後，用戶端便可對其進行存取。如果將其停用，它將仍然處於部署狀態，但不能透過用戶端對其進行存取。依預設，在部署應用程式或模組時，由於依預設選取了 [在所有目標上啓用] 單選按鈕，因此將啓用該應用程式或模組。

- 1 在樹形元件中，展開 [應用程式] 節點。
- 2 展開應用程式類型的節點。
- 3 若要啓用已部署的應用程式或模組，請選取已部署的應用程式或模組旁邊的核取方塊。
若要在單一目標上啓用應用程式，請執行以下步驟：
 - a. 選取應用程式的節點。
 - b. 按一下 [目標] 標籤。
 - c. 選取已部署的應用程式或模組旁邊的核取方塊。
- 4 按一下 [啓用] 或 [停用]。
這些按鈕用於在所有目標上啓用或停用應用程式。

更多資訊 **等效的 asadmin 指令**

`enable` 和 `disable`

▼ 管理應用程式目標

部署應用程式或模組後，可透過管理目標來管理參照該應用程式或模組的伺服器實例和叢集。

- 1 在樹形元件中，展開 [應用程式] 節點。
- 2 展開應用程式類型的節點。
- 3 選取已部署的應用程式的節點。
- 4 選取 [目標] 標籤。
- 5 若要在特定目標實例或叢集上啓用或停用應用程式，請按一下目標旁邊的核取方塊並按一下 [啓用] 或 [停用]。
- 6 若要新增或刪除應用程式的目標，則請選擇 [管理目標]。
- 7 新增或移除目標並按一下 [確定]。
此時，已修訂的目標清單中將包含該應用程式。

更多資訊 等效的 asadmin 指令

`create-application-ref` 和 `delete-application-ref`

▼ 在其他虛擬伺服器上部署

將應用程式或模組部署到目標伺服器實例或叢集之後，可以將其與其他虛擬伺服器關聯。

- 1 在已部署的應用程式或模組的 [目標] 頁面中，按一下目標旁邊的 [管理虛擬伺服器] 連結。
- 2 在可用虛擬伺服器的清單中，新增或移除虛擬伺服器目標。
- 3 按一下 [確定]。

重新部署到多個目標

如果已將應用程式部署到多個目標 (獨立伺服器實例或叢集)，則可以透過兩種方式來重新部署到多個目標。使用以下方法之一可以確定參考應用程式的所有伺服器實例接收到最新版本。

開發環境

在開發環境中，只重新部署應用程式。應用程式將被重新部署到網域，並且參考它的所有目標將自動接收到新的版本(如果已啓用目標伺服器實例的動態重新配置)。依預設，將啓用動態重新配置。如果未啓用伺服器實例的動態重新配置，將繼續使用舊版本，直至重新啓動伺服器實例。

生產環境

在生產環境中，請依循「關於滾動升級」中詳細說明的步驟執行作業。

▼ 配置動態重新載入

如果啓用了動態重新載入，伺服器將定期檢查已部署的應用程式中的變更並自動重新載入包含變更的應用程式。所做的變更將以您手動建立的 `.reload` 檔案的日期變更來表明。應用程式必須安裝在 `domain-dir/applications/j2ee-modules module-name` 或 `domain-dir/applications/j2ee-apps/ app-name` 中。

例如：

```
/opt/SUNWappserver/domain/domain1/applications/j2ee-modules/webapps-simple
```

動態重新載入在開發環境中非常有用，因為它能快速測試程式碼變更。但在生產環境中，動態重新載入可能會使效能降低。

備註 – 動態重新載入僅適用於預設伺服器實例。

動態重新載入旨在用於開發環境。它與階段作業持續性(一種生產環境功能)不相容。如果啓用了動態重新載入，請勿啓用階段作業持續性。

- 1 在樹形元件中，展開 [獨立實例] 節點。
- 2 按一下 [server] (管理伺服器)。
- 3 按一下 [進階]。
- 4 在 [應用程式配置] 頁面中，配置以下項目：
 - 重新載入：使用 [已啓用] 核取方塊來啓用或停用動態重新載入。
 - 重新載入輪詢間隔：指定伺服器檢查已部署的應用程式中的變更的頻率。
 - 管理階段作業逾時：指定管理階段作業逾時且必須重新登入之前的時間。

接下來的步驟

將系統配置為使用動態重新載入之後，請為每個要動態重新載入的應用程式建立一個名為 `.reload` 的檔案並將其置於應用程式的目錄中。此檔案沒有任何內容。變更應用程式時，會變更此檔案的日期(例如，使用 UNIX `touch` 指令)，並會自動重新載入變更。

適用於開發者的部署方法

另請參閱：

- 第 73 頁的「使用自動部署」
- 第 73 頁的「使用自動部署」
- 第 74 頁的「部署目錄中未封裝的應用程式」
- 第 74 頁的「使用 `deploytool` 公用程式」
- 第 75 頁的「使用部署規劃」

▼ 使用自動部署

自動部署功能使您能夠透過將預先封裝的應用程式或模組複製到 `domain-dir/autodeploy` 目錄來部署該應用程式或模組。

例如，將名為 `hello.war` 的檔案複製到 `domain-dir/autodeploy` 目錄。若要取消部署應用程式，請從 `autodeploy` 目錄中移除 `hello.war` 檔案。

您也可以使用 管理主控台 或 `asadmin` 工具來取消部署應用程式。在這種情況下，歸檔檔案將保留。

備註 - 自動部署僅適用於預設伺服器實例。

自動部署功能旨在用於開發環境。它與階段作業持續性 (一種生產環境功能) 不相容。如果已啟用自動部署，請勿啟用階段作業持續性。

- 1 在樹形元件中，展開 [獨立實例] 節點。
- 2 按一下 [server] (管理伺服器)。
- 3 按一下 [進階]。
- 4 在 [應用程式配置] 頁面中，配置以下項目：
 - a. 透過選取或取消選取 [已啟用] 核取方塊來啟用或停用自動部署。
 - b. 在 [自動部署輪詢間隔] 欄位中，指定伺服器檢查自動部署目錄中的應用程式檔案或模組檔案的頻率。
變更輪詢間隔不會影響部署應用程式或模組所需的時間。
 - c. 在 [自動部署目錄] 中，如果指定建立應用程式的目錄，則不必將檔案複製到預設自動部署目錄中。
預設目錄是伺服器實例的根目錄中名為 `autodeploy` 的目錄。

依預設，可使用變數自動變更多個伺服器實例的目錄。如需有關這些變數的更多資訊，請參閱第 303 頁的「設定進階網域屬性」。

- d. 若要在部署之前執行檢驗器，請選取 [檢驗器]。

檢驗器將檢查檔案的結構和特性。大型應用程式的檢驗通常會很費時。

- e. 若要預編譯 [JSP] 頁面，請選取 [預編譯]。

如果未選取此核取方塊，則首次存取 JSP 頁面時會在執行階段編譯這些頁面。由於編譯通常很費時，因此在生產環境中請選取此核取方塊。

▼ 部署目錄中未封裝的應用程式

此功能適用於進階開發者。

使用目錄部署僅部署到預設伺服器實例 (server)。您不能使用它來部署到叢集或獨立伺服器實例。

包含未封裝的應用程式或模組的目錄有時稱為展開的目錄。目錄的內容必須與對應的 J2EE 歸檔檔案內容相符。例如，如果部署某一目錄中的 Web 應用程式，則該目錄的內容必須與對應的 WAR 檔案的內容相同。如需有關必需的目錄內容的資訊，請參閱相應的規格。

您可以直接在展開的目錄中變更部署描述元檔案。

如果您的環境配置為使用動態重新載入，則還可以從目錄中動態重新載入已部署的應用程式。如需更多資訊，請參閱第 72 頁的「配置動態重新載入」。

- 1 在管理主控台中開始部署程序。請參閱第 60 頁的「部署 Web 應用程式」。
- 2 在 [部署] 頁面中，指定以下內容：
 - a. 按一下單選按鈕以指定必須可以從伺服器上存取的套裝軟體檔案或目錄路徑。
 - b. 在 [檔案或目錄] 欄位中，輸入展開的目錄的名稱。

更多資訊 等效的 asadmin 指令

```
deploydir
```

使用 deploytool 公用程式

適用於軟體開發者的 deploytool 公用程式可以封裝並部署 J2EE 應用程式和模組。如需有關如何使用 deploytool 的說明，請參閱「The J2EE 1.4 Tutorial」。

使用部署規劃

此功能適用於進階開發者。

部署規劃是指僅包含特定於 Application Server 的部署描述元的 JAR 檔案。有關這些部署描述元 (例如 `sun-application.xml`) 的說明，請參閱「Application Server Developer's Guide」。部署規劃是「JSR 88: J2EE Application Deployment」實作的一部分。使用部署規劃可以部署不包含特定於 Application Server 的部署描述元的應用程式或模組。

若要使用部署規劃進行部署，請指定 `asadmin deploy` 指令的 `--deploymentplan` 選項。例如，以下指令將根據 `mydeployplan.jar` 檔案中指定的規劃來部署 `myrosterapp.ear` 檔案中的企業應用程式。

```
$ asadmin deploy --user admin ---deploymentplan mydeployplan.jar myrosterapp.ear
```

在企業應用程式 (EAR) 的部署規劃檔案中，`sun-application.xml` 檔案位於根目錄下。根據以下語法來儲存每個模組的部署描述元：`module-name.sun-dd-name`，其中 `sun-dd-name` 取決於模組類型。如果模組包含 CMP 對映檔案，則該檔案命名為 `module-name.sun-cmp-mappings.xml`。`.dbschema` 檔案儲存在根層級目錄下，並用磅符號 (#) 替代每個正斜線字元 (/)。下面列示的內容顯示了企業應用程式 (EAR) 的部署規劃檔案的結構。

```
$ jar -tvf mydeployplan.jar
420 Thu Mar 13 15:37:48 PST 2003 sun-application.xml
370 Thu Mar 13 15:37:48 PST 2003 RosterClient.war.sun-web.xml
418 Thu Mar 13 15:37:48 PST 2003 roster-ac.jar.sun-application-client.xml
1281 Thu Mar 13 15:37:48 PST 2003 roster-ejb.jar.sun-ejb-jar.xml
2317 Thu Mar 13 15:37:48 PST 2003 team-ejb.jar.sun-ejb-jar.xml
3432 Thu Mar 13 15:37:48 PST 2003 team-ejb.jar.sun-cmp-mappings.xml
84805 Thu Mar 13 15:37:48 PST 2003 team-ejb.jar.RosterSchema.dbschema
```

在 Web 應用程式或模組檔案的部署規劃中，特定於 Application Server 的部署描述元位於根層級目錄下。如果獨立 EJB 模組包含 CMP Bean，則部署規劃包括位於根層級目錄的 `sun-cmp-mappings.xml` 和 `.dbschema` 檔案。在下面列示的內容中，部署規劃介紹了 CMP Bean。

```
$ jar r -tvf myotherplan.jar
3603 Thu Mar 13 15:24:20 PST 2003 sun-ejb-jar.xml
3432 Thu Mar 13 15:24:20 PST 2003 sun-cmp-mappings.xml
84805 Thu Mar 13 15:24:20 PST 2003 RosterSchema.dbschema
```


JDBC 資源

本章說明如何配置存取資料庫的應用程式所要求使用的 JDBC 資源。本章包括下列小節：

- 第 77 頁的「關於 JDBC 資源和連線池」
- 第 78 頁的「設定資料庫存取」
- 第 80 頁的「關於 JDBC 連線池」
- 第 84 頁的「關於 JDBC 資源」
- 第 87 頁的「關於持續性管理程式資源」

關於 JDBC 資源和連線池

- 第 77 頁的「JDBC 資源」
- 第 78 頁的「JDBC 連線池」
- 第 78 頁的「JDBC 資源和連線池如何協同工作」

JDBC 資源

爲了儲存、組織和擷取資料，大多數應用程式均使用關聯式資料庫。J2EE 應用程式透過 JDBC API 存取關聯式資料庫。

JDBC 資源 (資料源) 爲應用程式提供連線至資料庫的方法。通常，管理員要爲部署在網域中的應用程式存取的每個資料庫建立 JDBC 資源。(但是，可以爲一個資料庫建立多個 JDBC 資源。)

若要建立 JDBC 資源，請指定識別資源的專屬 JNDI 名稱。(請參閱「JNDI 名稱和資源」一節。)JDBC 資源的 JNDI 名稱應在 `java:comp/env/jdbc` 子環境中。例如，薪水帳單資料庫資源的 JNDI 名稱可爲 `java:comp/env/jdbc/payrolldb`。由於所有資源 JNDI 名稱均在 `java:comp/env` 子環境中，因此在管理主控台中指定 JDBC 資源的 JNDI 名稱時，僅需輸入 `jdbc/name`。例如，對於薪水帳單資料庫，可指定 `jdbc/payrolldb`。

JDBC 連線池

若要建立 JDBC 資源，請指定與其關聯的連線池。多個 JDBC 資源可指定單一連線池。

JDBC 連線池是針對特定資料庫的一組可重複使用的連線。由於每建立一個新的實體連線都會耗費時間，因此伺服器維護了可用連線池以提高效能。當應用程式請求連線時，它可以從池中取得一個連線。應用程式關閉連線時，連線將傳回池中。

連線池的特性可能會隨資料庫供應商的不同而有所不同。某些特性是通用的，例如資料庫名稱 (URL)、使用者名稱和密碼。

JDBC 資源和連線池如何協同工作

爲了儲存、組織和擷取資料，大多數應用程式均使用關聯式資料庫。J2EE 應用程式透過 JDBC API 存取關聯式資料庫。應用程式存取資料庫之前，必須先取得連線。

以下是在執行階段應用程式連線至資料庫時所發生的情況：

1. 應用程式透過 JNDI API 進行呼叫以獲取與資料庫關聯的 JDBC 資源 (資料源)。
如果給定了資源的 JNDI 名稱，命名和目錄服務將查找 JDBC 資源。每個 JDBC 資源指定一個連線池。
2. 通過 JDBC 資源，應用程式獲得一個資料庫連線。
應用程式伺服器秘密地從與該資料庫相對應的連線池中擷取實體連線。池定義資料庫名稱 (URL)、使用者名稱和密碼等連線屬性。
3. 由於已將應用程式連線至資料庫，因此該應用程式可以讀取和修改資料庫中的資料以及將資料增加到資料庫中。
應用程式透過對 JDBC API 進行呼叫來存取資料庫。JDBC 驅動程式可將應用程式的 JDBC 呼叫翻譯爲資料庫伺服器的協定。
4. 存取資料庫完成之後，應用程式將關閉該連線。
應用程式伺服器將連線傳回連線池。連線傳回連線池之後，下一個應用程式便可以使用該連線。

設定資料庫存取

- [第 79 頁的「設定資料庫存取」](#)
- [第 79 頁的「整合 JDBC 驅動程式」](#)

▼ 設定資料庫存取

- 1 安裝支援的資料庫產品。
如需有關 Application Server 支援的資料庫產品的清單，請參閱版本說明。
- 2 安裝適用於該資料庫產品的 JDBC 驅動程式。
- 3 使網域的伺服器實例可以存取此驅動程式的 JAR 檔案。請參閱第 79 頁的「[整合 JDBC 驅動程式](#)」。
- 4 建立資料庫。
通常，應用程式供應程式提供了用於建立和填入資料庫的程序檔。
- 5 為資料庫建立連線池。請參閱第 80 頁的「[建立 JDBC 連線池](#)」。
- 6 建立指向連線池的 JDBC 資源。請參閱第 84 頁的「[建立 JDBC 資源](#)」。

▼ 整合 JDBC 驅動程式

JDBC 驅動程式可將應用程式的 JDBC 呼叫翻譯為資料庫伺服器的協定。

- 若要將 JDBC 驅動程式整合到管理網域中，請執行以下操作之一：
 - 使通用類別載入器可以存取該驅動程式。
 - a. 將驅動程式的 JAR 檔案和 ZIP 檔案複製到 `domain-dir/lib` 目錄或將其類別檔案複製到 `domain-dir/lib/ext` 目錄中。
 - b. 重新啟動該網域。
 - 使系統類別載入器可以存取該驅動程式。
 - a. 在管理主控台的樹狀結構檢視(左窗格)中，選取[配置]。
 - b. 選取所需的配置(例如 `default-config`)。
 - c. 選取[JVM 設定]。
 - d. 在[JVM 設定]頁面中，按一下[路徑設定]標籤。
 - e. 在[類別路徑後綴]欄位中，輸入驅動程式之 JAR 檔案的完全合格的路徑名稱。

- f. 按一下 [儲存]。
- g. 重新啟動伺服器。

關於 JDBC 連線池

- 第 80 頁的「建立 JDBC 連線池」
- 第 81 頁的「編輯 JDBC 連線池」
- 第 84 頁的「驗證連線池設定」
- 第 84 頁的「刪除 JDBC 連線池」

▼ 建立 JDBC 連線池

JDBC 連線池是針對特定資料庫的一組可重複使用的連線。使用管理主控台建立連線池時，管理員實際上是在定義與特定資料庫的連線的各個方面。

開始之前 建立池之前，您必須首先安裝並整合 JDBC 驅動程式。請參閱第 79 頁的「整合 JDBC 驅動程式」。

建置 [建立連線池] 頁面時，必須輸入特定於 JDBC 驅動程式和資料庫供應商的特定資料。繼續操作之前，請先收集以下資訊：

- 資料庫供應商名稱
- 資源類型，例如 `javax.sql.DataSource` (僅限於本機作業事件)
`javax.sql.XADataSource` (全域作業事件)
- 資料源類別名稱
- 必需的特性，例如資料庫名稱 (URL)、使用者名稱和密碼

- 1 在管理主控台的樹狀結構檢視 (左窗格) 中，展開 [資源] 節點。
- 2 在 [資源] 節點下，展開 [JDBC] 節點。
- 3 在 [JDBC] 節點下，選取 [連線池] 節點。
- 4 在 [連線池] 頁面中，按一下 [新建]。
- 5 在 [建立連線池] 的第一個頁面中，指定以下一般設定：
 - a. 在 [名稱] 欄位中，輸入池的邏輯名稱。
建立 JDBC 資源時指定此名稱。
 - b. 從 [資源類型] 組合方塊中選取一個項目。

- c. 從 [資料庫供應商] 組合方塊中選取一個項目。
- 6 按 [下一步]。
- 7 在 [建立連線池] 的第二個頁面中，為 [DataSource 類別名稱] 欄位指定值。
如果 JDBC 驅動程式具有與在上一頁中指定的資源類型和資料庫供應商相應的 DataSource 類別，則系統會提供 [DataSource 類別名稱] 欄位的值。
- 8 按 [下一步]。
- 9 在 [建立連線池] 的第三個也就是最後一個頁面中，執行以下作業：
 - a. 在 [一般設定] 區段中，檢驗各個值是否正確。
 - b. 對於 [池設定]、[連線驗證] 和 [作業事件隔絕] 區段中的欄位，保留預設值。
您可以在以後非常方便地變更這些設定。請參閱第 81 頁的「編輯 JDBC 連線池」。
 - c. 在 [附加特性] 表中，新增必需的特性，例如資料庫名稱 (URL)、使用者名稱和密碼。
- 10 按一下 [完成]。

更多資訊 等效的 asadmin 指令

```
create-jdbc-connection-pool
```

▼ 編輯 JDBC 連線池

[編輯 JDBC 連線池] 頁面為您提供了變更現有池的所有設定 (池的名稱除外) 的方法。

- 1 在樹形元件中，展開 [資源] 節點。
- 2 在 [資源] 節點下，展開 [JDBC] 節點。
- 3 在 [JDBC] 節點下，展開 [連線池] 節點。
- 4 選擇要編輯的池的節點。
- 5 在 [編輯 JDBC 連線池] 頁面中，進行必要的變更。
 - a. 變更一般設定。
一般設定的值取決於安裝的特定 JDBC 驅動程式。這些設定是 Java 程式設計語言中的類別名稱或介面名稱。

參數	說明
DataSource 類別名稱	實作 DataSource 和/或 XADataSource API 的供應商特定的類別名稱。該類別位於 JDBC 驅動程式中。
資源類型	選項包括 javax.sql.DataSource (僅限於本機作業事件)、javax.sql.XADataSource (全域作業事件) 和 java.sql.ConnectionPoolDataSource (本機作業事件，可能會改善效能)。

b. 變更池設定。

一組實體資料庫連線保存在池中。應用程式請求連線時，將從池中移除該連線；而應用程式釋放該連線之後，連線將傳回到池中。

參數	說明
池的初始大小和最小大小	池中連線的最小數目。該值還確定了首次建立池或應用程式伺服器啟動時置於池中的連線的數目。
池的最大大小	池中連線的最大數目。
池設定大小數量	當池向最小池大小收縮時，將成批調整大小。此值確定批次中的連線數目。將該值設置過大會延遲連線循環；而將該值設置過小則會導致效率降低。
閒置逾時	連線可以在儲存區中間置的最大時間(以秒表示)。一旦超過此時間，即從池中移除該連線。
最長等待時間	請求連線的應用程式在達到連線逾時之前等待的時間。由於預設等待時間過長，應用程式可能會出現無限期當機的情況。

c. 變更連線驗證設定。

選擇性地，應用程式伺服器可以在將連線傳送給應用程式之前驗證連線。當由於網路出現故障或資料庫伺服器當機造成資料庫不可用時，此驗證可允許 Application Server 自動重新建立資料庫連線。連線驗證會耗用額外的時間，並會略微降低效能。

參數	說明
連線驗證	選取 [需要] 核取方塊以啟用連線驗證。

參數	說明
驗證方法	<p>應用程式伺服器可以使用三種方法來驗證資料庫連線：auto-commit、metadata 和 table。</p> <p>auto-commit 和 metadata - Application Server 透過呼叫 <code>con.setAutoCommit()</code> 方法和 <code>con.getMetaData()</code> 方法來驗證連線。但是，由於許多 JDBC 驅動程式快取了這些呼叫的結果，因此這兩種方法無法始終提供可靠的驗證。請與驅動程式供應商進行核實，以確定這些呼叫是否被快取。</p> <p>table - 應用程式將查詢指定的資料庫表格。此表必須存在並且可以存取，但不要求表的列數。請勿使用包含許多列的現有表或經常存取的表。</p>
表格名稱	如果從 [驗證方法] 組合方塊中選取了表，請在此處指定資料庫表格的名稱。
一旦失敗	選取標記為 [關閉所有連線] 的核取方塊之後，如果單一連線失敗，Application Server 將關閉池中的所有連線，然後重新建立這些連線。如果未選取此核取方塊，則僅在要使用個別連線時才會重新建立這些連線。

d. 變更作業事件隔離設定。

由於許多使用者通常可以同步運作地存取一個資料庫，因此可能出現一個作業事件在更新資料而另一個作業事件嘗試讀取同一資料的情況。作業事件的隔絕層級定義了正在更新的資料對於其他作業事件的可視程度。如需有關隔絕層級的詳細資訊，請參閱資料庫供應商的文件。

參數	說明
作業事件隔絕	使您可以為該池的連線選取作業事件隔絕層級。如果不指定此參數，連線將使用 JDBC 驅動程式提供的預設隔絕層級進行作業。
受保證隔絕層級	該項僅在指定了隔絕層級的情況下才適用。如果選取 [受保證] 核取方塊，則從池中獲取的所有連線都具有相同的隔絕層級。例如，如果上次使用連線時程式化 (使用 <code>con.setTransactionIsolation</code>) 變更了連線的隔離層級，這種機制會將狀態變更回指定的隔離層級。

e. 變更特性。

在 [附加特性] 表中，可以指定資料庫名稱 (URL)、使用者名稱和密碼等特性。由於隨資料庫供應商的不同，特性也會有所不同，因此請查閱供應商文件以取得詳細資訊。

6 按一下 [儲存]。

- 另請參閱
- 第 80 頁的「建立 JDBC 連線池」
 - 第 84 頁的「驗證連線池設定」

- [第 84 頁的「刪除 JDBC 連線池」](#)

▼ 驗證連線池設定

- 1 啟動資料庫伺服器。

- 2 按一下 [Ping]。

管理主控台 將嘗試連線至資料庫。如果顯示錯誤訊息，請檢查資料庫伺服器是否已重新啟動。

▼ 刪除 JDBC 連線池

- 1 在樹形元件中，展開 [資源] 節點。
- 2 在 [資源] 節點下，展開 [JDBC] 節點。
- 3 在 [JDBC] 節點下，選取 [連線池] 節點。
- 4 在 [連線池] 頁面中，選取要刪除的池的核取方塊。
- 5 按一下 [刪除]。

更多資訊 **等效的 asadmin 指令**

```
delete-jdbc-connection-pool
```

關於 JDBC 資源

- [第 84 頁的「建立 JDBC 資源」](#)
- [第 85 頁的「編輯 JDBC 資源」](#)
- [第 86 頁的「刪除 JDBC 資源」](#)
- [第 86 頁的「啟用或停用 JDBC 資源」](#)

▼ 建立 JDBC 資源

JDBC 資源 (資料源) 為應用程式提供連線至資料庫的方法。

開始之前 建立 JDBC 資源之前，先建立 JDBC 連線池。請參閱[第 80 頁的「建立 JDBC 連線池」](#)。

- 1 在樹形元件中，展開 [資源] 節點。

- 2 在 [資源] 節點下，展開 [JDBC] 節點。
- 3 在 [JDBC] 節點下，選取 [JDBC 資源] 節點。
- 4 在 [JDBC 資源] 頁面中，按一下 [新建]。
- 5 在 [建立 JDBC 資源] 頁面中，指定資源的設定：
 - a. 在 [JNDI 名稱] 欄位中，鍵入專屬名稱。
依照慣例，該名稱應以 jdbc/ 字串開頭。例如：jdbc/payrolldb。請勿遺漏正斜線。
 - b. 從 [池名稱] 組合方塊中選擇要與新的 JDBC 資源關聯的連線池。
 - c. 如果要使資源不可用，請取消選取 [啟用] 核取方塊。
依預設，建立資源之後立即可以使用資源 (已啟用)。
 - d. 在 [描述] 欄位中，鍵入資源的簡短描述。
 - e. 在 [目標] 區段，指定具有可用資源的目標 (叢集和獨立伺服器實例)。
在左側選取所需的目標，然後按一下 [增加] 以將其增加到選取目標的清單中。
- 6 按一下 [確定]。

更多資訊 **等效的 asadmin 指令**

```
create-jdbc-resource
```

▼ 編輯 JDBC 資源

- 1 在樹形元件中，展開 [資源] 節點。
- 2 在 [資源] 節點下，展開 [JDBC] 節點。
- 3 在 [JDBC] 節點下，展開 [JDBC 資源] 節點。
- 4 選取要編輯的 JDBC 資源的節點。
- 5 在 [編輯 JDBC 資源] 頁面中，可以執行以下作業：
 - a. 從 [池名稱] 組合方塊中選取其他連線池。
 - b. 在 [描述] 欄位中，變更資源的簡短描述。

- c. 選取或取消選取核取方塊以啓用或停用資源。
 - d. 選取 [目標] 標籤，以變更具有可用資源的目標 (叢集和獨立伺服器實例)。
選取清單中的現有目標的核取方塊，然後按一下 [啓用] 以啓用該目標的資源，或者按一下 [停用] 以停用該目標的資源。
按一下 [管理目標] 可以在清單中增加或移除目標。在 [管理目標] 頁面中，從左側的 [可用] 清單中選取所需的目標，然後按一下 [增加] 以將其增加到選取目標的清單中。按一下 [移除] 可以從 [已選取] 清單中移除目標。
按一下 [確定] 以儲存對可用目標的變更。
- 6 按一下 [儲存] 以套用編輯。

▼ 刪除 JDBC 資源

- 1 在樹形元件中，展開 [資源] 節點。
- 2 在 [資源] 節點下，展開 [JDBC] 節點。
- 3 在 [JDBC] 節點下，選取 [連線池] 節點。
- 4 在 [連線池] 頁面中，選取要刪除的池的核取方塊。
- 5 按一下 [刪除]。

▼ 啓用或停用 JDBC 資源

- 1 在樹形元件中，展開 [JDBC 資源] 節點或展開 [獨立實例] 以選取 [伺服器實例節點資源] 標籤。
- 2 在 [資源] 頁面中，選取要啓用或停用的資源的核取方塊。
- 3 按一下 [啓用] 或 [停用]。

關於持續性管理程式資源

- 第 87 頁的「建立持續性管理員資源」
- 第 88 頁的「編輯持續性管理員資源」
- 第 88 頁的「管理資源目標」
- 第 88 頁的「刪除持續性管理員資源」
- 第 89 頁的「啟用或停用持續性管理員資源」

▼ 建立持續性管理員資源

向下相容性需要此功能。若要在 Application Server 7 版本上執行，使用容器管理的持續性 Bean (一種 EJB 元件) 的應用程式需要持續性管理員資源。建議改為使用 JDBC 資源。

- 1 在樹形元件中，展開 [資源] 節點。
- 2 在 [資源] 節點下，選取 [持續性管理程式] 節點。
- 3 在 [持續性管理程式] 頁面中，按一下 [新建]。
- 4 在 [建立持續性管理程式] 頁面中，指定以下設定：
 - a. 在 [JNDI 名稱] 欄位中，鍵入專屬名稱。
例如：jdo/mypm。請勿遺漏正斜線。
 - b. 在 [工廠類別] 欄位中，保留此版本提供的預設類別，或鍵入其他實作類別。
 - c. 從 [連線池] 組合方塊中選擇新的持續性管理程式資源所屬的連線池。
 - d. 若要停用新的持續性管理員資源，請取消選取 [啟用] 核取方塊。
依預設，將啟用新的持續性管理程式資源。
 - e. 在 [目標] 區段，指定具有可用資源的目標 (叢集和獨立伺服器實例)。
在左側選取所需的目標，然後按一下 [增加] 以將其增加到選取目標的清單中。
- 5 按一下 [確定]。

更多資訊 **等效的 asadmin 指令**

```
create-persistence-resource
```

▼ 編輯持續性管理員資源

- 1 在 [編輯持續性管理程式特性] 標籤中，選取 [增加特性] 按鈕。
[附加特性] 表中將增加一個新列。
- 2 增加所需的特性和值。

▼ 管理資源目標

- 1 選取 [目標] 標籤，以變更資源所在的目標 (叢集和獨立伺服器實例)。
- 2 選取清單中的現有目標的核取方塊，然後按一下 [啟用] 以啟用該目標的資源，或者按一下 [停用] 以停用該目標的資源。
- 3 按一下 [管理目標] 可以在清單中增加或移除目標。
在 [管理目標] 頁面中，從左側的 [可用] 清單中選取所需的目標，然後按一下 [增加] 以將其增加到選取目標的清單中。按一下 [移除] 可以從 [已選取] 清單中移除目標。
- 4 按一下 [確定] 以儲存對可用目標的變更。
- 5 按一下 [儲存]。

▼ 刪除持續性管理員資源

- 1 在樹形元件中，展開 [持續性管理程式] 節點。
- 2 選取 [持續性管理程式] 節點。
- 3 在 [持續性管理程式] 頁面中，選取要刪除的持續性管理程式的核取方塊。
- 4 按一下 [刪除]。

更多資訊 **等效的 asadmin 指令**

```
delete-persistence-resource
```


▼ 啓用或停用持續性管理員資源

- 1 在樹形元件中，展開 [持續性管理程式] 節點。
- 2 選取要啓用或停用的資源的核取方塊。
- 3 按一下 [啓用] 或 [停用]。

配置 Java 訊息服務資源

本章描述如何為使用 Java 訊息服務 (JMS) API 的應用程式配置資源。它包含以下小節：

- 第 91 頁的「關於 JMS 資源」
- 第 93 頁的「有關 JMS 連線工廠的管理主控台作業」
- 第 97 頁的「有關 JMS 目標資源的管理主控台作業」
- 第 99 頁的「有關 JMS 實體目標的管理主控台作業」
- 第 101 頁的「有關 JMS 提供者的管理主控台作業」

關於 JMS 資源

- 第 91 頁的「Application Server 中的 JMS 提供者」
- 第 91 頁的「JMS 資源」
- 第 93 頁的「JMS 資源與連接器資源之間的關係」

Application Server 中的 JMS 提供者

Application Server 透過將 Sun Java System Message Queue (原來稱為 Sun ONE Message Queue) 軟體整合到 Application Server 中，實作了 Java 訊息服務 (JMS) API。對於基本的 JMS API 管理作業，請使用 Application Server 管理主控台。對於進階作業 (包括管理 Message Queue 叢集)，請使用 `MQ-install-dir/imq/bin` 目錄中提供的工具。

如需有關管理 Message Queue 的詳細資訊，請參閱「Message Queue Administration Guide」。

JMS 資源

Java 訊息服務 (JMS) API 使用兩種管理物件：

- 允許應用程式以程式化方式建立其他 JMS 物件的連線工廠物件。
- 用作訊息儲存庫的目標。

這些物件是以管理方式建立的，而建立物件的方式則特定於每個 JMS 實作。在 Application Server 中執行以下作業：

- 透過建立連線工廠資源來建立連線工廠
- 透過建立兩個物件來建立目標：
 - 實體目標
 - 參考實體目標的目標資源

JMS 應用程式使用 JNDI API 來存取連線工廠和目標資源。通常，JMS 應用程式至少使用一個連線工廠和一個目標。若要瞭解應建立的資源，請研究應用程式或向應用程式開發者洽詢。

連線工廠分為三種類型：

- QueueConnectionFactory 物件，用於點對點通訊
- TopicConnectionFactory 物件，用於出版訂閱通訊
- ConnectionFactory 物件，可用於點對點通訊和出版訂閱通訊；建議將這些物件用於新的應用程式

有兩種類型的目標：

- Queue 物件，用於點對點通訊
- Topic 物件，用於出版訂閱通訊

「J2EE 1.4 Tutorial」中有關 JMS 的章節提供了有關此兩種通訊和 JMS 其他方面的詳細資訊(請參閱 <http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html>)。

建立資源的次序並不重要。

對於 J2EE 應用程式，請在 Application Server 部署描述元中指定連線工廠和目標資源，如下所示：

- 在 resource-ref 或 mdb-connection-factory 元素中指定連線工廠 JNDI 名稱。
- 在訊息導引 Bean 的 ejb 元素和 message-destination 元素中指定目標資源 JNDI 名稱。
- 在 message-destination-link 元素中指定實體目標名稱，該元素在企業 Bean 部署描述元的 message-driven 元素或 message-destination-ref 元素內。此外，還應在 message-destination 元素中指定該實體目標名稱。(message-destination-ref 元素替代了在新應用程式中已停用的 resource-env-ref 元素。)在 Application Server 部署描述元的 message-destination 元素中，將實體目標名稱與目標資源名稱連結起來。

JMS 資源與連接器資源之間的關係

Application Server 透過使用名為 `jmsra` 的系統資源配接卡實作 JMS。使用者建立 JMS 資源時，Application Server 會自動建立連接器資源，這些連接器資源將顯示在管理主控台樹狀結構檢視的 [連接器] 節點下。

對於使用者建立的每個 JMS 連線工廠，Application Server 均會建立連接器連線池和連接器資源。對於使用者建立的每個 JMS 目標，Application Server 均會建立管理物件資源。使用者刪除 JMS 資源時，Application Server 會自動刪除連接器資源。

您可以使用管理主控台的 [連接器] 節點 (而非 [JMS 資源] 節點) 為 JMS 系統資源配接卡建立連接器資源。請參閱第 7 章，以取得詳細資訊。

有關 JMS 連線工廠的管理主控台作業

- 第 93 頁的「建立 JMS 連線工廠資源」
- 第 96 頁的「編輯 JMS 連線工廠資源」
- 第 96 頁的「刪除 JMS 連線工廠資源」

▼ 建立 JMS 連線工廠資源

- 1 在樹形元件中，展開 [資源] 節點，然後展開 [JMS 資源] 節點。
- 2 選取 [連線工廠] 節點。
- 3 在 [JMS 連線工廠] 頁面中，按一下 [新建]。
將顯示 [建立 JMS 連線工廠] 頁面。
- 4 在 [JNDI 名稱] 欄位中，鍵入連線工廠的名稱。
建議的作業是使用 JMS 資源的命名子環境前綴 `jms/`。例如：`jms/ConnectionFactory1`。
- 5 從 [類型] 下拉式清單中，選擇 `javax.jms.ConnectionFactory`、`javax.jms.QueueConnectionFactory` 或 `javax.jms.TopicConnectionFactory`。
- 6 選取 [啟用] 核取方塊以在執行階段啟用該資源。
- 7 在 [進階] 區域中，變更連線工廠屬性所需的值。
如需有關這些屬性的詳細資訊，請參閱第 96 頁的「編輯 JMS 連線工廠資源」。Application Server 會將這些屬性套用至為連線工廠建立的連接器連線池。

8 請為 JMS 連線工廠資源指定 [作業事件支援] 值，如下所示：

- 為可用於作業事件 (此作業事件包含在作業事件範圍內使用的多個資源) 的資源指定 XATransaction (預設值)。
例如，此資源加上 JDBC 資源、連接器資源或其他 JMS 連線工廠資源。此值提供了最大的靈活性。配置為 XATransaction 的資源將參與分為兩個階段的確定作業。
- 為可用於作業事件 (此作業事件僅包含作業事件範圍內的一個資源) 的資源，或用做分散式作業事件 (此分散式作業事件包含多個 XA 資源) 中最後一個代理程式的資源指定 LocalTransaction。
使用此值可以獲得更好的效能。配置為 LocalTransaction 的資源不會被用於分為兩個階段的確定作業。
- 為永遠無法參與作業事件的資源指定 NoTransaction。
此設定在 JMS 應用程式中的使用受限。

9 在 [附加特性] 區域中，為應用程式所需的特性提供值。下表列示了可用的特性。

特性名稱	說明
ClientId	為長期訂戶將要使用的連線工廠指定用戶端 ID。
AddressList	指定訊息代理程式實例或應用程式要與其通訊的實例的名稱和連接埠號 (可選擇)。清單中的每個位址都指定了要連線的主機名稱 (和連接埠號與連線服務 [選擇性地])。例如，該值可能為 earth 或 earth:7677。當訊息代理程式在預設連接埠 (7676) 以外的連接埠上執行時，請指定連接埠號。如果特性設定在叢集環境中指定了多個主機和連接埠，則將使用該清單中的第一個可用主機 (除非將 AddressListBehavior 特性設定為 RANDOM)。 如需詳細資訊，請參閱「 <i>Message Queue Developer's Guide for Java Clients</i> 」。 預設：本地主機和預設連接埠號 (7676)。用戶端將嘗試連線至位於本地主機連接埠 7676 上的代理程式。
MessageServiceAddressList	與 AddressList 相同。此特性名稱已被拒絕。請改為使用 AddressList。
UserName	進入連線工廠的使用者名稱。 預設：guest
Password	進入連線工廠的密碼。 預設：guest
ReconnectEnabled	如果啟用此特性 (值 = true)，請指定用戶端執行階段在遺失連線時嘗試重新連線至訊息伺服器 (或 AddressList 中的位址清單)。 預設：TRUE

特性名稱	說明
ReconnectAttempts	<p>指定用戶端執行階段嘗試連線 (或重新連線) AddressList 清單中每個位址的次數。到達這個值後，用戶端執行階段將嘗試連線清單中的下一個位址。值 -1 表示重新連線嘗試次數沒有限制 (用戶端執行階段將嘗試連線至第一個位址直到連線成功)。</p> <p>預設：3</p>
ReconnectInterval	<p>指定兩次連線嘗試之間的間隔 (以毫秒為單位)。此屬性適用於對 AddressList 中每個位址的嘗試，及對該清單中連續位址的嘗試。如果該間隔太短，則代理程式將沒有時間恢復。如果該間隔太長，則重新連線會變得遲緩，以至於讓人無法接受。</p> <p>預設：30000</p>
AddressListBehavior	<p>指定是按 AddressList 屬性中的位址順序 (PRIORITY) 還是按隨機順序 (RANDOM) 嘗試連線。</p> <p>RANDOM 表示重新連線將從 AddressList 中隨機選擇位址。如果許多用戶端嘗試使用同一個連線工廠來進行連線，則使用該值可以阻止它們全部連線至同一個位址。</p> <p>PRIORITY 表示重新連線時始終嘗試連線 AddressList 中的第一個伺服器位址，而僅在第一個代理程式不可用時才使用其他位址。</p> <p>預設：RANDOM</p>
AddressListIterations	<p>指定用戶端執行階段建立 (或重新建立) 連線時，在 AddressList 中反覆運算的次數。值 -1 表示嘗試次數沒有限制。</p> <p>預設：3</p>

- 10 在 [目標] 區域中，執行以下操作：
 - a. 從 [可用] 欄中選取一個或多個目標，在這些目標上將部署使用了資源的應用程式。可用目標包括可用叢集和可用伺服器實例，以及預設的伺服器實例 server。
 - b. 按一下 [新增] 以將目標移至 [選取] 欄中。
- 11 按一下 [確定] 以儲存連線工廠。

更多資訊 等效的 asadmin 指令

```
create-jms-resource
```

▼ 編輯 JMS 連線工廠資源

- 1 在樹形元件中，展開 [資源] 節點，然後展開 [JMS 資源] 節點。
- 2 展開 [連線工廠] 節點。
- 3 選取要編輯的連線工廠。
- 4 在 [編輯 JMS 連線工廠] 頁面中，可以執行以下作業：
 - 修改 [描述] 欄位中的文字。
 - 選取或取消選取 [啓用] 核取方塊以啓用或停用資源。
 - 變更 [進階] 區域中的屬性值。
 - 新增、移除或修改特性。
- 5 此外，按一下 [目標] 標籤以移至 [JMS 連線工廠資源目標] 頁面。在此頁面中，執行以下操作：
 - a. 按一下 [管理目標] 以開啓 [管理資源目標] 頁面。
在此頁面中，在 [可用] 欄與 [選取] 欄之間移動目標。確定將目標放入 [選取] 欄中。使用了資源的應用程序將部署在這樣的目標上。可用目標包括可用的叢集和伺服器實例，以及預設的伺服器實例 server。按一下 [確定] 以儲存變更。
 - b. 選取目標的核取方塊，然後按一下 [啓用] 或 [停用]，以啓用或停用目標的資源。
- 6 按一下 [儲存] 以儲存變更。

▼ 刪除 JMS 連線工廠資源

- 1 在樹形元件中，展開 [資源] 節點，然後展開 [JMS 資源] 節點。
- 2 選取 [連線工廠] 節點。
- 3 在 [JMS 連線工廠] 頁面中，選取要刪除的連線工廠的名稱旁邊的核取方塊。
- 4 按一下 [刪除]。

更多資訊 **等效的 asadmin 指令**

```
delete-jms-resource
```


有關 JMS 目標資源的管理主控台作業

- 第 97 頁的「建立 JMS 目標資源」
- 第 98 頁的「編輯 JMS 目標資源」
- 第 98 頁的「刪除 JMS 目標資源」

▼ 建立 JMS 目標資源

- 1 在樹形元件中，展開 [資源] 節點，然後展開 [JMS 資源] 節點。
- 2 選取 [目標資源] 節點。
- 3 在 [JMS 目標資源] 頁面中，按一下 [新建]。將顯示 [建立 JMS 目標資源] 頁面。
- 4 在 [JNDI 名稱] 欄位中，鍵入資源的名稱。
建議的作業是使用 JMS 資源的命名子環境前綴 jms/。例如：jms/Queue。
- 5 從 [類型] 下拉式清單中，選擇 javax.jms.Topic 或 javax.jms.Queue。
- 6 選取 [啟用] 核取方塊以在執行階段啟用該資源。
- 7 在 [附加特性] 區域中，為特性提供值。
下表列示了可用的特性。

特性名稱	說明
Name	(需要) 資源參考的實體目標的名稱。
Description	實體目標的描述。

- 8 在 [目標] 區域中，執行以下操作：
 - a. 從 [可用] 欄中選取一個或多個目標，在這些目標上將部署使用了資源的應用程式。
可用目標包括可用的叢集和伺服器實例，以及預設的伺服器實例 server。
 - b. 按一下 [新增] 以將目標移至 [選取] 欄中。
- 9 按一下 [確定]。

更多資訊 **等效的 asadmin 指令**

```
create-jms-resource
```

▼ 編輯 JMS 目標資源

- 1 在樹形元件中，展開 [資源] 節點，然後展開 [JMS 資源] 節點。
- 2 展開 [目標資源] 節點。
- 3 選取要編輯的目標資源。
- 4 在 [編輯 JMS 目標資源] 頁面中，可以執行以下作業：
 - 變更資源的類型。
 - 修改 [描述] 欄位中的文字。
 - 選取或取消選取 [啓用] 核取方塊以啓用或停用資源。
 - 新增、移除或修改 Name 或 Description 特性。
- 5 按一下 [儲存] 以儲存變更。
- 6 此外，按一下 [目標] 標籤以移至 [JMS 目標資源目標] 頁面。在此頁面中，執行以下操作：
 - a. 按一下 [管理目標] 以開啓 [管理資源目標] 頁面。

在此頁面中，在 [可用] 欄與 [選取] 欄之間移動目標。確定將目標放入 [選取] 欄中。使用了資源的應用程序將部署在這樣的目標上。可用目標包括可用的叢集和伺服器實例，以及預設的伺服器實例 server。按一下 [確定] 以儲存變更。
 - b. 選取目標的核取方塊，然後按一下 [啓用] 或 [停用]，以啓用或停用目標的資源。

▼ 刪除 JMS 目標資源

- 1 在樹形元件中，展開 [資源] 節點，然後展開 [JMS 資源] 節點。
- 2 選取 [目標資源] 節點。
- 3 在 [JMS 目標資源] 頁面中，選取要刪除的目標資源的名稱旁邊的核取方塊。
- 4 按一下 [刪除]。

更多資訊 等效的 asadmin 指令

```
delete-jms-resource
```

有關 JMS 實體目標的管理主控台作業

- 第 99 頁的「建立 JMS 實體目標」
- 第 100 頁的「刪除 JMS 實體目標」

▼ 建立 JMS 實體目標

若要進行生產，務必建立實體目標。但是，在開發和測試期間，不需要執行此步驟。應用程式首次存取目標資源時，Message Queue 會自動建立目標資源的 Name 特性指定的實體目標。該實體目標是暫時的，並且將在 Message Queue 配置特性指定的時間段後過期。

- 1 在樹形元件中，展開 [配置] 節點，然後展開 [Java 訊息服務] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 server，請選取 [server-config] 節點。
 - 若要為將來的實例 (實例使用 default-config 的副本) 配置預設設定，請選取 [default-config] 節點。
- 3 選取 [實體目標] 節點。
- 4 在 [實體目標] 頁面中，按一下 [新建]。
將顯示 [建立實體目標] 頁面。
- 5 在 [實體目標名稱] 欄位中，鍵入目標的名稱 (例如，PhysicalQueue)。
- 6 從 [類型] 下拉式清單中，選擇 [topic] 或 [queue]。
- 7 在 [附加特性] 區域中，按一下 [增加特性] 以增加特性。
下表列示了目前可用的一個特性。

特性名稱	說明
maxNumActiveConsumers	在佇列目標的負載平衡傳送中處於使用中狀態的使用者的最大數目。如果值為 -1，則表示沒有數目限制。預設為 1 (如果為獨立伺服器實例建立目標) 和 -1 (如果為叢集建立目標)。

若要修改此特性的值或要指定其他實體目標特性，請使用 `MQ-install-dir/imq/bin/imqcmd` 指令。請參閱「Message Queue Administration Guide」，以取得更多資訊。

- 按一下 [確定]。

更多資訊 等效的 asadmin 指令

```
create-jmsdest
```

系統目標

[實體目標] 頁面可顯示系統目標 (即名為 `mq.sys.dm` 的佇列)，過期和無法傳送的訊息將重新導向至該佇列。可以為此目標建立目標資源、使用者和瀏覽器，但不能刪除它或向其傳送訊息。

▼ 刪除 JMS 實體目標

- 在樹形元件中，展開 [配置] 節點，然後展開 [Java 訊息服務] 節點。
- 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 `server`，請選取 [server-config] 節點。
 - 若要為將來的實例 (實例使用 `default-config` 的副本) 配置預設設定，請選取 [default-config] 節點。
- 選取 [實體目標] 節點。
- 在 [實體目標] 頁面中，選取要刪除的目標的名稱旁邊的核取方塊。
- 按一下 [刪除]。
如果嘗試刪除系統目標 `mq.sys.dm`，則會顯示錯誤訊息。

更多資訊 等效的 asadmin 指令

```
delete-jmsdest
```

有關 JMS 提供者的管理主控台作業

- 第 101 頁的「配置 JMS 提供者的一般特性」
- 第 104 頁的「建立 JMS 主機」
- 第 106 頁的「編輯 JMS 主機」
- 第 106 頁的「刪除 JMS 主機」

▼ 配置 JMS 提供者的一般特性

使用 [JMS 服務] 頁面配置所有 JMS 連線都使用的特性。

如需有關配置 JMS 服務的更多資訊，請參閱「Application Server Developer's Guide」。

- 1 在樹形元件中，選取 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 `server`，請選取 [server-config] 節點。
 - 若要為將來的實例 (實例使用 `default-config` 的副本) 配置預設設定，請選取 [default-config] 節點。
- 3 選取 [Java 訊息服務] 節點以開啓 [JMS 服務] 頁面。
- 4 請編輯 [啓動逾時] 欄位中的值，以變更 Application Server 在中斷啓動之前等待 JMS 服務啓動的時間。

在運行緩慢或超載的系統中，在預設逾時值 (60) 的基礎上增大該值。
- 5 從 [類型] 下拉式清單中：
 - 選擇 [LOCAL] (`server-config` 配置的預設設定) 以存取本地主機上的 JMS 服務。該 JMS 服務由 Application Server 啓動和管理。
 - 選擇 [REMOTE] (`default-config` 配置的預設設定) 以存取其他系統或叢集上的 JMS 服務。如果選擇了 [REMOTE]，則在下次啓動伺服器時 JMS 服務不會由 Application Server 啓動。而是通過 Message Queue 來啓動和管理 JMS 服務，因而必須單獨啓動 Message Queue 代理程式。如需有關啓動代理程式的資訊，請參閱「*Message Queue Administration Guide*」。如果選擇此值並使用遠端主機，請依循第 106 頁的「編輯 JMS 主機」中的說明來指定遠端主機的名稱。
- 6 在 [啓動引數] 欄位中，鍵入引數以自訂 JMS 服務啓動。

可使用透過 `MQ-install-dir/imq/bin/imqbrokerd` 指令提供的任何引數。

- 7 連線遺失時，使用 [重新連線] 核取方塊指定 JMS 服務是否嘗試重新連線至訊息伺服器 (或 AddressList 中的位址清單)。
依預設，啟用重新連線。
- 8 在 [重新連線間隔] 欄位中，鍵入重新連線嘗試之間的秒數。
此屬性適用於對 AddressList 中每個位址的嘗試，及對該清單中連續位址的嘗試。如果該間隔太短，則代理程式將沒有時間恢復。如果該間隔太長，則重新連線會變得遲緩，以至於讓人無法接受。
其預設值為 60 秒。
- 9 在 [重新連線嘗試] 欄位中，鍵入用戶端執行階段嘗試連線 (或重新連線) AddressList 清單中每個位址的次數。到達這個值後，用戶端執行階段將嘗試連線清單中的下一個位址。
值 -1 表示重新連線嘗試次數沒有限制 (用戶端執行階段將嘗試連線至第一個位址直到連線成功)。預設值為 3。
- 10 從 [預設 JMS 主機] 下拉式清單中選擇一個主機。預設為 default_JMS_host。
- 11 在 [位址清單運作方式] 下拉式清單中，選擇是按 AddressList 中的位址順序 (priority) 還是按隨機順序 (random) 嘗試連線。
priority 表示重新連線時始終嘗試連線 AddressList 中的第一個伺服器位址，而僅在第一個代理程式不可用時才使用其他位址。
如果許多用戶端嘗試使用同一個連線工廠來進行連線，則應指定 random 以防止這些用戶端全部連線至同一個位址。
預設為 random。
- 12 在 [位址清單反覆運算] 欄位中，鍵入 JMS 服務建立 (或重新建立) 連線時，在 AddressList 中反覆運算的次數。
值 -1 表示嘗試次數沒有限制。
預設值為 3。
- 13 若要使用非預設方案或服務，請在 [MQ 方案] 和 [MQ 服務] 欄位中，鍵入 Message Queue 位址方案名稱和 Message Queue 連線服務名稱。
訊息服務位址的完整語法為
`scheme://address-syntax`
其中，`scheme` 和 `address-syntax` 將在下表說明。
MQ 方案和 MQ 服務是下表的前兩欄中顯示的值。

方案名稱	連線服務	說明	位址語法
mq	jms 和 ssljms	Message Queue 用戶端執行階段將連線位於指定主機和連接埠的 MQ 連接埠對映器。該連接埠對映器傳回動態建立的連線服務連接埠的清單，然後 Message Queue 用戶端執行階段將連線託管指定連線服務的連接埠。	<code>[hostName][:port][/serviceName]</code> 預設： <code>hostName = localhost, port = 7676, serviceName = jms</code> 預設僅適用於 jms 連線服務。對於 ssljms 連線服務，需要指定所有的變數。 範例： <code>mq:MyHost:7677/ssljms</code>
mqtcp	jms	Message Queue 用戶端執行階段將透過與指定的主機和連接埠建立 TCP 連線 (略過 MQ 連接埠對映器) 來建立連線。	<code>hostName:port / jms</code> 範例： <code>mqtcp:localhost:7676/jms</code>
mqssl	ssljms	Message Queue 用戶端執行階段將透過與指定的主機和連接埠建立安全 SSL 連線 (略過 MQ 連接埠對映器) 來建立連線。	<code>hostName:port / ssljms</code> 範例： <code>mqssl:localhost:7676/ssljms</code>
http	httpjms	Message Queue 用戶端執行階段將與指定 URL 處的 Message Queue 隧道 Servlet 建立 HTTP 連線。(必須配置代理程式，以存取 Message Queue 管理員指南中所說明的 HTTP 隧道 Servlet。)	<code>hostName:port / contextRoot/tunnel</code> 如果多個代理程式實例使用同一個隧道 Servlet，則連線特定代理程式實例 (而不是隨機選取的實例) 的語法為： <code>http://hostName:port/contextRoot / tunnel?serverName =hostName:instanceName</code>
https	httpsjms	Message Queue 用戶端執行階段將與指定的 Message Queue 隧道 Servlet URL 建立安全 HTTPS 連線。(必須配置代理程式，以存取 Message Queue 管理員指南中所說明的 HTTPS 隧道 Servlet。)	<code>hostName:port / contextRoot/tunnel</code> 如果多個代理程式實例使用同一個隧道 Servlet，則連線特定代理程式實例 (而不是隨機選取的實例) 的語法為： <code>http://hostName:port/contextRoot / tunnel?serverName =hostName:instanceName</code>

14 在 [附加特性] 區域中，按一下 [增加特性] 以增加特性。

下表列出了可用的 Sun Java System Message Queue 代理程式配置特性。

特性名稱	說明
<code>instance-name</code>	指定完整的 Message Queue 代理程式實例名稱。預設為 <code>imqbroker</code> 。
<code>instance-name-suffix</code>	指定要增加到完整的 Message Queue 代理程式實例名稱中的後綴。該後綴與實例名稱之間以底線字元 () 分隔。例如，如果實例名稱為 <code>imqbroker</code> ，則在附加後綴 <code>xyz</code> 之後，實例名稱將變更為 <code>imqbroker_xyz</code> 。

特性名稱	說明
append-version	如果為 <code>true</code> ，則應在完整的 Message Queue 代理程式實例名稱後附加主要和次要版本號碼，這兩個版本號碼之前都有一個底線字元 (<code>_</code>)。例如，如果實例名稱為 <code>imqbroker</code> ，則在附加版本號碼之後，實例名稱將變更為 <code>imqbroker_8_0</code> 。預設為 <code>false</code> 。

- 15 按一下 [儲存] 以儲存變更，或按一下 [載入預設值] 以復原服務的預設值。

更多資訊 存取遠端伺服器

將提供者和主機變更為遠端系統會使所有 JMS 應用程式都在遠端伺服器上執行。若要在使用本機伺服器的同時使用一個或多個遠端伺服器，請透過使用 `AddressList` 特性建立連線工廠資源來建立可存取遠端伺服器的連線。請參閱第 93 頁的「[建立 JMS 連線工廠資源](#)」。

▼ 驗證 JMS 服務設定

- 1 在樹形元件中，選取 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 `server`，請選取 `[server-config]` 節點。
 - 若要為將來的實例 (實例使用 `default-config` 的副本) 配置預設設定，請選取 `[default-config]` 節點。
- 3 選取 [Java 訊息服務] 節點以開啓 [JMS 服務] 頁面。
- 4 按一下 [Ping]。

如果 JMS 服務已啓動並正在執行，則會顯示「Ping 成功：JMS 服務正在執行」的訊息。

更多資訊 等效的 `asadmin` 指令

```
jms-ping
```

▼ 建立 JMS 主機

Application Server Platform Edition 僅需要一個 JMS 主機，其預設名稱為 `default_JMS_host`。您可以建立其他主機，但是 Application Server 將無法識別它們。

- 1 在樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 `server`，請選取 [server-config] 節點。
 - 若要為將來的實例 (實例使用 `default-config` 的副本) 配置預設設定，請選取 [default-config] 節點。
- 3 展開 [Java 訊息服務] 節點。
- 4 選取 [JMS 主機] 節點。
- 5 在 [JMS 主機] 頁面中，按一下 [新建]。將顯示 [建立 JMS 主機] 頁面。
- 6 在 [名稱] 欄位中，鍵入主機的名稱。例如：
`NewJmsHost`
- 7 在 [主機] 欄位中，鍵入執行 JMS 主機的系統的名稱或網際網路協定 (IP) 位址 (`localhost` 或者本機或遠端系統的名稱)。
- 8 在 [連接埠] 欄位中，鍵入 JMS 服務的連接埠號。
僅當要使用的 JMS 服務在非預設連接埠上執行時，才需要變更此欄位。(預設連接埠號為 7676。)
- 9 在 [管理使用者名稱] 和 [管理密碼] 欄位中，鍵入 Message Queue 代理程式使用者名稱和密碼。
該使用者名稱和密碼與 Application Server 的使用者名稱和密碼不同。僅當已使用 `MQ-install-dir/imq/bin/imqusermgr` 指令變更了 Message Queue 代理程式值時，才需要編輯這些欄位。預設值為 `admin` 和 `admin`。
- 10 按一下 [確定]。

更多資訊 等效的 `asadmin` 指令

```
create-jms-host
```

▼ 編輯 JMS 主機

- 1 在樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 server，請選取 [server-config] 節點。
 - 若要為將來的實例 (實例使用 default-config 的副本) 配置預設設定，請選取 [default-config] 節點。
- 3 展開 [Java 訊息服務] 節點。
- 4 選取 [JMS 主機] 節點。
- 5 在 [JMS 主機] 頁面中，選取要編輯的主機。
- 6 在 [編輯 JMS 主機] 頁面中，可以執行以下作業：
 - 在 [主機] 欄位中，變更主機名稱或網際網路通訊協定 (IP) 地址。
 - 在 [連接埠] 欄位中，變更 JMS 服務的連接埠號。
 - 變更 [管理使用者名稱] 和 [管理密碼] 欄位中的值。
- 7 按一下 [儲存] 以儲存變更，或按一下 [載入預設值] 以復原主機的預設值。

▼ 刪除 JMS 主機

- 1 在樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 server，請選取 [server-config] 節點。
 - 若要為將來的實例 (實例使用 default-config 的副本) 配置預設設定，請選取 [default-config] 節點。
- 3 展開 [Java 訊息服務] 節點。
- 4 選取 [JMS 主機] 節點。

- 5 在 [JMS 主機] 頁面中，選取要刪除的主機名稱旁邊的核取方塊。
- 6 按一下 [刪除]。



注意 - 請勿刪除所有的 JMS 主機。這樣做會使 Application Server 無法重新啓動。必須至少留下一個 JMS 主機。

更多資訊 **等效的 asadmin 指令**

`delete-jms-host`

配置 JavaMail 資源

本章介紹如何為使用 JavaMail API 的應用程式配置資源。它包含以下小節：

- 第 109 頁的「關於 JavaMail」
- 第 109 頁的「有關 JavaMail 的管理主控台作業」

關於 JavaMail

JavaMail API 是一組用於建立郵件系統模型的抽象 API。API 提供了一個獨立於平台和協定的架構來建置郵件和訊息傳送應用程式。JavaMail API 提供了多個工具用於讀取和傳送電子郵件。服務提供者可實作特定協定。

JavaMail API 可做為 Java 平台選擇性套裝軟體實作，還可做為 J2EE 平台的一部分使用。

Application Server 包含 JavaMail API 以及 JavaMail 服務提供者，使應用程式元件可以透過網際網路傳送電子郵件通知，以及從 IMAP 和 POP3 郵件伺服器讀取電子郵件。

如需有關 JavaMail API 的更多資訊，請至 JavaMail 網站 (<http://java.sun.com/products/javamail/>)。

有關 JavaMail 的管理主控台作業

- 第 110 頁的「建立 JavaMail 階段作業」
- 第 111 頁的「編輯 JavaMail 階段作業」
- 第 111 頁的「刪除 JavaMail 階段作業」

▼ 建立 JavaMail 階段作業

- 1 在樹形元件中，展開 [資源] 節點，然後選取 [JavaMail 階段作業] 節點。
- 2 在 [JavaMail 階段作業] 頁面中，按一下 [新建]。
將顯示 [建立 JavaMail 階段作業] 頁面。
- 3 在 [JNDI 名稱] 欄位中，鍵入階段作業的名稱。
建議的作業是使用 JavaMail 資源的命名子環境前綴 mail/。例如：mail/MySession。
- 4 在 [郵件主機] 欄位中，鍵入預設郵件伺服器的主機名稱。
如果未提供協定特定的主機特性，Store 和 Transport 物件的連線方法將使用該值。名稱必須可以解析為實際的主機名稱。
- 5 在 [預設使用者] 欄位中，鍵入連線至郵件伺服器時要提供的使用者名稱。
如果未提供協定特定的使用者名稱特性，Store 和 Transport 物件的連線方法使用該值。
- 6 在 [預設傳回位址] 欄位中，鍵入預設使用者的郵件位址，格式為 `username@host.domain`。
- 7 如果您不希望此時啟用郵件階段作業，請取消選取 [啟用] 核取方塊。
- 8 在 [進階] 區域中，僅當已將 Application Server 的郵件提供者重新配置為使用非預設儲存或傳輸協定時，才需要變更這些欄位值。
依預設，儲存協定為 `imap`；儲存協定類別為 `com.sun.mail.imap.IMAPStore`；傳輸協定為 `smtp`；傳輸協定類別為 `com.sun.mail.smtp.SMTPTransport`。
- 9 選取 [除錯] 核取方塊以啟用郵件階段作業的附加除錯輸出 (包括通訊協定追蹤)。
如果將 JavaMail 記錄層級設為 `FINE` 或更精細，則系統會產生除錯輸出，並且該輸出會包含在系統記錄檔中。請參閱第 251 頁的「配置記錄層級」，以取得有關設定記錄層級的資訊。
- 10 在 [附加特性] 區域中，按一下 [增加特性] 以新增應用程式所需的特性 (如協定特定的主機或使用者名稱特性)。
JavaMail API 文件列出了可用的特性 (<http://java.sun.com/products/javamail/javadocs/index.html>)。
- 11 在 [目標] 區域中，執行以下操作：
 - a. 從 [可用] 欄中選取一個或多個目標，在這些目標上將部署使用了資源的應用程式。
可用目標包括可用叢集和可用伺服器實例，以及預設的伺服器實例 `server`。

b. 按一下 [新增] 以將目標移至 [選取] 欄中。

12 按一下 [確定] 以儲存階段作業。

更多資訊 **等效的 asadmin 指令**

```
create-javamail-resource
```

▼ 編輯 JavaMail 階段作業

- 1 在樹形元件中，展開 [資源] 節點，然後選取 [JavaMail 階段作業] 節點。
- 2 在 [JavaMail 階段作業] 頁面中，選取要編輯的階段作業。
- 3 在 [編輯 JavaMail 階段作業] 頁面中，您可以執行以下作業：
 - 修改 [郵件主機]、[預設使用者]、[預設傳回位吞和 [描述] 欄位中的值。
 - 選取或取消選取 [啟用] 核取方塊以啟用或停用資源。
 - 修改 [進階] 欄位中的值。
 - 新增、移除或修改特性。
- 4 按一下 [目標] 標籤以移至 [JavaMail 階段作業目標] 頁面。在此頁面中，執行以下操作：
 - a. 按一下 [管理目標] 以開啓 [管理資源目標] 頁面。
在此頁面中，在 [可用] 欄與 [選取] 欄之間移動目標。確定將目標放入 [選取] 欄中。使用了資源的應用程序將部署在這樣的目標上。可用目標包括可用叢集和可用伺服器實例，以及預設的伺服器實例 server。按一下 [確定] 以儲存變更。
 - b. 選取目標的核取方塊，然後按一下 [啟用] 或 [停用]，以啟用或停用目標的資源。
- 5 按一下 [儲存] 以儲存變更，或按一下 [載入預設值] 以復原郵件階段作業的預設值。

▼ 刪除 JavaMail 階段作業

- 1 在樹形元件中，展開 [資源] 節點，然後選取 [JavaMail 階段作業] 節點。
- 2 在 [JavaMail 階段作業] 頁面中，選取要刪除的階段作業的名稱旁邊的核取方塊。
- 3 按一下 [刪除]。

[更多資訊](#) **等效的 asadmin 指令**

```
delete-javamail-resource
```


JNDI 資源

- 第 113 頁的「關於 Java 命名與目錄介面 (JNDI)」
- 第 115 頁的「關於自訂資源」
- 第 117 頁的「關於外部 JNDI 儲存庫和資源」

關於 Java 命名與目錄介面 (JNDI)

本小節將討論 Java 命名和目錄介面 (JNDI)。JNDI 是用於存取不同類型的命名和目錄服務的應用程式設計介面 (API)。J2EE 元件透過呼叫 JNDI 查找方法來尋找物件。

本小節包含以下主題：

- 第 113 頁的「JNDI 名稱和資源」
- 第 113 頁的「J2EE 命名服務」
- 第 114 頁的「命名參考與連結資訊」

JNDI 名稱和資源

JNDI 是 Java 命名和目錄介面 API 的首字母縮略。透過對此 API 進行呼叫，應用程式可以尋找資源和其他程式物件。資源是提供與系統 (如資料庫伺服器和郵件傳送系統) 的連線的程式物件。(JDBC 資源有時被稱為資料源。)每個資源物件都是由專屬的易懂名稱識別，稱為 JNDI 名稱。Application Server 包含的命名和目錄服務將資源物件及其 JNDI 名稱連結在一起。若要建立新資源，需要將新的名稱-物件連結輸入到 JNDI 中。

J2EE 命名服務

JNDI 名稱是易懂的的物件名稱。這些名稱透過 J2EE 伺服器提供的命名和目錄服務連結到其物件。由於 J2EE 元件透過 JNDI API 存取此服務，因此物件通常使用其 JNDI 名稱。例如，PointBase 資料庫的 JNDI 名稱為 jdbc/Pointbase。當 Application Server 啟動時，將從配置檔案中讀取資訊，並自動將 JNDI 資料庫名稱增加到名稱空間。

需要 J2EE 應用程式用戶端、企業 Bean 與 Web 元件來存取 JNDI 命名環境。

應用程式元件的命名環境是一種機制，使用它可以在部署或組譯期間自訂應用程式元件的企業邏輯。使用應用程式元件的環境即可對應用程式元件進行自訂，而無需存取或變更應用程式元件的源代碼。

J2EE 容器實作應用程式元件的環境，並將該環境做為 JNDI 命名環境提供給應用程式元件實例。應用程式元件的環境的使用方式如下：

- 應用程式元件的商業方法使用 JNDI 介面存取該環境。應用程式元件提供者在部署描述元中宣告應用程式元件需要其執行階段環境提供的所有環境項目。
- 容器提供儲存應用程式元件環境的 JNDI 命名環境的實作。容器還提供了部署程式可以用於建立和管理每個應用程式元件的環境的工具。
- 部署程式使用容器提供的工具，可以初始化應用程式元件的部署描述元中宣告的環境項目。部署程式可以設定和修改環境項目的值。
- 容器使環境命名環境在執行階段可用於應用程式元件實例。應用程式元件的實例使用 JNDI 介面獲取環境項目的值。

每個應用程式元件定義了其本身的環境項目集。一個應用程式元件在同一容器內的所有實例共用相同的環境項目。不允許應用程式元件實例在執行階段修改環境。

命名參考與連結資訊

資源參照是部署描述元中的元素，可以為資源識別元件的編碼名稱。更具體地說，編碼名稱參考資源的連線工廠。在下一小節給出的範例中，資源參照名稱為 `jdbc/SavingsAccountDB`。

資源的 JNDI 名稱與資源參照的名稱不同。使用此命名方法，您需要在進行部署之前先對映這兩個名稱，但此方法也用於分離元件與資源。由於具有此分離功能，因此如果元件在以後需要存取其他資源，則無需變更名稱。這一靈活性使您可以更輕鬆地從預先存在的元件編譯 J2EE 應用程式。

下表列出了 Application Server 所使用的 J2EE 資源的 JNDI 查詢及其關聯的參照。

表 6-1 JNDI 查找及其關聯的參考

JNDI 查找名稱	關聯的參考
<code>java:comp/env</code>	應用程式環境項目
<code>java:comp/env/jdbc</code>	JDBC DataSource 資源管理程式連線 Factory
<code>java:comp/env/ejb</code>	EJB 參照
<code>java:comp/UserTransaction</code>	UserTransaction 參考

表 6-1 JNDI 查找及其關聯的參考 (續)

JNDI 查找名稱	關聯的參考
java:comp/env/mail	JavaMail 階段作業連線 Factory
java:comp/env/url	URL 連線 Factory
java:comp/env/jms	JMS 連線 Factory 與目標
java:comp/ORB	應用程式元件共用的 ORB 實例

關於自訂資源

- 第 115 頁的「使用自訂資源」
- 第 115 頁的「建立自訂資源」
- 第 116 頁的「編輯自訂資源」
- 第 116 頁的「刪除自訂資源」
- 第 117 頁的「列示自訂資源」

使用自訂資源

自訂資源存取本機 JNDI 儲存庫，外部資源存取外部 JNDI 儲存庫。這兩種類型的資源都需要使用者指定的工廠類別元素、JNDI 名稱屬性等。在本小節中，我們將討論如何為 J2EE 資源配置 JNDI 連線工廠資源，以及如何存取這些資源。

在 Application Server 中，您可以建立、刪除和列示資源以及 `list-jndi-entities`。

▼ 建立自訂資源

- 1 在管理主控台的左窗格中，為要修改的 JNDI 配置開啓 Application Server 實例。
- 2 開啓 [JNDI] 標籤，並按一下 [自訂資源]。
任何已經建立的自訂資源會在右側窗格中列示。若要建立新的自訂資源，按一下 [新建]。開啓 [JNDI] 標籤，並按一下 [新建]。將顯示一個頁面，用於增加新的自訂資源。
- 3 在 [JNDI 名稱] 欄位中，輸入用於存取資源的名稱。
此名稱將註冊到 JNDI 命名服務中。
- 4 在 [資源類型] 欄位中，輸入完全合格的類型定義，如上面範例中所示。
[資源類型] 定義遵循的格式為 `xxx.xxx`。

- 5 在 [工廠類別] 欄位中，為要建立的自訂資源輸入工廠類別名稱。
該欄位中的值是使用者指定的工廠類別的名稱。此類別可實作 `javax.naming.spi.ObjectFactory` 介面。
- 6 在 [描述] 欄位中，為要建立的資源輸入描述。
此描述是字串值，最多可以包含 250 個字元。
- 7 在 [附加特性] 區段，增加特性名稱和值。
- 8 標示 [啓用自訂資源] 核取方塊，以啓用自訂資源。
- 9 按一下 [確定]，以儲存自訂資源。
如果已在叢集或獨立實例上部署自訂資源，則可以使用 [目標] 標籤管理目標。此標籤將在建立自訂資源之後顯示。透過輸入目標名稱並按一下 [確定] 來設定目標。

更多資訊 等效的 asadmin 指令

```
create-custom-resource
```

▼ 編輯自訂資源

- 1 在管理主控台的左窗格中，為要修改的 JNDI 配置開啓 Application Server 實例。
- 2 開啓 [JNDI]，並選取 [自訂資源]。
任何已經建立的自訂資源會在右側窗格中列示。
- 3 請在右窗格中按一下檔案名稱。
- 4 編輯 [資源類型] 欄位、[工廠類別] 欄位或 [描述] 欄位。
- 5 標示 [啓用自訂資源] 核取方塊，以啓用自訂資源。
- 6 按一下 [儲存] 以儲存對自訂資源的變更。

▼ 刪除自訂資源

- 1 在管理主控台的左窗格中，開啓 [JNDI] 標籤。
- 2 按一下 [自訂資源]。
任何已經建立的自訂資源會在右側窗格中列示。

- 3 請按一下要刪除資源的名稱旁邊的方塊。
- 4 按一下 [刪除]。此自訂資源即被刪除。

更多資訊 等效的 asadmin 指令

```
delete-custom-resource
```

列示自訂資源

若要列示自訂資源，請鍵入 `asadmin list-custom-resources` 指令。例如，若要列示主機 `plum` 上的自訂資源，請鍵入以下指令：

```
$asadmin list-custom-resources --host plum target6
```

如需完整的環境，請鍵入 `asadmin help list-custom-resources`。

關於外部 JNDI 儲存庫和資源

- 第 117 頁的「使用外部 JNDI 儲存庫和資源」
- 第 118 頁的「建立外部資源」
- 第 119 頁的「編輯外部資源」
- 第 119 頁的「刪除外部資源」
- 第 120 頁的「列示外部資源」

使用外部 JNDI 儲存庫和資源

通常，Application Server 上執行的應用程式需要存取儲存在外部 JNDI 儲存庫中的資源。例如，一般的 Java 物件可能會以 Java 模式儲存在 LDAP 伺服器中。外部 JNDI 資源元素允許使用者配置此類外部資源儲存庫。外部 JNDI 工廠必須實作 `javax.naming.spi.InitialContextFactory` 介面。

使用外部 JNDI 資源的範例為：

```
<resources>
<!-- external-jndi-resource element specifies how to access J2EE resources
-- stored in an external JNDI repository. The following example
-- illustrates how to access a java object stored in LDAP.
-- factory-class element specifies the JNDI InitialContext factory that
-- needs to be used to access the resource factory. property element
-- corresponds to the environment applicable to the external JNDI context
-- and jndi-lookup-name refers to the JNDI name to lookup to fetch the
```

```

-- designated (in this case the java) object.
-->
<external-jndi-resource jndi-name="test/myBean"
  jndi-lookup-name="cn=myBean"
  res-type="test.myBean"
  factory-class="com.sun.jndi.ldap.LdapCtxFactory">
  <property name="PROVIDER-URL" value="ldap://ldapserver:389/o=myObjects" />
  <property name="SECURITY_AUTHENTICATION" value="simple" />
  <property name="SECURITY_PRINCIPAL", value="cn=joeSmith, o=Engineering" />
  <property name="SECURITY_CREDENTIALS" value="changeit" />
</external-jndi-resource>
</resources>

```

▼ 建立外部資源

- 1 在管理主控台的左窗格中，為要修改的 JNDI 配置開啓 Application Server 實例。
- 2 開啓 [JNDI] 並選取 [外部資源]。
任何已經建立的外部資源會在右側窗格中列示。
- 3 若要建立新的外部資源，按一下 [新建]。
- 4 在 [JNDI 名稱] 欄位中，輸入用於存取資源的名稱。
此名稱將註冊到 JNDI 命名服務中。
- 5 在 [資源類型] 欄位中，輸入完全合格的類型定義，如上面範例中所示。
[資源類型] 定義遵循的格式為 xxx.xxx。
- 6 在 [JNDI 查找] 欄位中，輸入要在外部儲存庫中查找的 JNDI 值。
例如，在建立一個與外部儲存庫連線的外部資源時，為了測試某個 Bean 類別，[JNDI 查詢] 可能會如下所示： `cn=testmybean`。
- 7 在 [工廠類別] 欄位中，輸入 JNDI 工廠類別外部儲存庫 (例如，`com.sun.jndi.ldap`)。
此類別可實作 `javax.naming.spi.ObjectFactory` 介面。
- 8 在 [描述] 欄位中，為要建立的資源輸入描述。
此描述是字串值，最多可以包含 250 個字元。
- 9 在 [附加特性] 區段，增加特性名稱和值。
- 10 標示 [啓用外部資源] 核取方塊，以啓用外部資源。

11 按一下 [確定] 以儲存外部資源。

如果已在叢集或獨立實例上部署外部資源，則可以使用 [目標] 標籤管理目標。此標籤將在建立外部資源之後顯示。透過輸入目標名稱並按一下 [確定] 來設定目標。

更多資訊 等效的 asadmin 指令

```
create-jndi-resource
```

▼ 編輯外部資源

- 1 在管理主控台的左窗格中，為要修改的 JNDI 配置開啓 Application Server 實例。
- 2 開啓 [JNDI] 並選取 [外部資源]。
任何已經建立的外部資源會在右側窗格中列示。
- 3 若要編輯外部資源，請在右側窗格中按一下檔案名稱。
- 4 編輯 [資源類型] 欄位、[JNDI 查找] 欄位、[工廠類別] 欄位或 [描述] 欄位。
- 5 標示 [啓用外部資源] 核取方塊，以啓用外部資源。
- 6 按一下 [儲存] 以儲存對外部資源的變更。

▼ 刪除外部資源

- 1 在管理主控台的左窗格中，開啓 [JNDI] 標籤。
- 2 按一下 [外部資源]。
任何已經建立的外部資源會在右側窗格中列示。
- 3 請按一下要刪除資源的名稱旁邊的方塊。
- 4 按一下 [刪除]。此外部資源即被刪除。

更多資訊 等效的 asadmin 指令

```
delete-jndi-resource
```

列示外部資源

若要列示外部資源，請鍵入 `asadmin list-jndi-resources` 指令並指定 JNDI 名稱。例如，鍵入以下指令可以列示外部資源：

```
$asadmin list-jndi-resources --user adminuser --host plum jndi_name_test
```

如需完整的環境，請鍵入 `asadmin help list-jndi-resources`。

連接器資源

本章描述如何配置用於存取企業資訊系統 (EIS) 的連接器。本章包括下列小節：

- 第 121 頁的「關於連接器」
- 第 122 頁的「有關連接器連線池的管理主控台作業」
- 第 125 頁的「有關連接器資源的管理主控台作業」
- 第 127 頁的「有關管理物件資源的管理主控台作業」

關於連接器

連接器模組也稱為資源介面，是允許應用程式與企業資訊系統 (EIS) 進行互動式操作的 J2EE 元件。EIS 軟體包含各種類型的系統：包括企業資源計劃 (ERP)、主機作業事件處理和非關聯式資料庫。類似其他 J2EE 模組，安裝連接器模組即是部署該連接器模組。

連接器連線池是針對特定 EIS 的一組可重複使用的連線。若要建立連接器連線池，請指定與池關聯的連接器模組 (資源介面)。

連接器資源是為應用程式提供到 EIS 的連線的程式物件。若要建立連接器資源，請指定其 JNDI 名稱及其關聯的連線池。多個連接器資源可以指定單一連線池。應用程式可透過查找資源的 JNDI 名稱定位資源。(如需有關 JNDI 的更多資訊，請參閱 [JNDI 名稱和資源] 部分)。EIS 的連接器資源的 JNDI 名稱通常位於 `java:comp/env/eis-specific` 子環境中。

Application Server 使用連接器模組 (資源介面) 實作 JMS。請參閱「JMS 資源和連接器資源的關係」一節。

有關連接器連線池的管理主控台作業

- 第 122 頁的「設定 EIS 存取」
- 第 122 頁的「建立連接器連線池」
- 第 123 頁的「編輯連接器連線池」
- 第 124 頁的「刪除連接器連線池」

▼ 設定 EIS 存取

- 1 部署 (安裝) 連接器。請參閱第 64 頁的「部署連接器模組」。
- 2 為連接器建立連線池。請參閱第 122 頁的「建立連接器連線池」。
- 3 建立與連線池關聯的連接器資源。請參閱第 125 頁的「建立連接器資源」。

▼ 建立連接器連線池

開始之前 建立池之前，先部署與該池關聯的連接器模組 (資源介面)。為新池指定的值取決於部署的連接器模組。

- 1 在樹形元件中，展開 [資源] 節點，然後展開 [連接器] 節點。
- 2 選取 [連接器連線池] 節點。
- 3 在 [連接器連線池] 頁面中，按一下 [新建]。
- 4 在 [建立連接器連線池] 的第一個頁面中，指定以下設定：
 - a. 在 [名稱] 欄位中，輸入池的邏輯名稱。
建立連接器資源時將指定此名稱。
 - b. 從 [資源介面] 組合方塊中選取一個項目。
該組合方塊顯示已部署的資源介面 (連接器模組) 的清單。
- 5 按 [下一步]。
- 6 在 [建立連接器連線池] 的第二個頁面中，從 [連線定義] 組合方塊中選取一個值。
組合方塊中的選擇取決於資源介面。通常，會指定一種 ConnectionFactory (工廠實例) 以建立與 EIS 的連線。
- 7 按 [下一步]。

- 8 在 [建立連接器連線池] 的第三個也即最後一個頁面中，執行以下作業：
 - a. 在 [一般設定] 區段中檢驗各個值是否正確。
 - b. 對於 [池設定] 區段中的欄位，可以保留預設值。
可以在以後變更這些設定。請參閱「編輯連接器連線池」。
 - c. 在 [附加特性] 表中，增加任何所需的特性。
在 [建立連接器連線池] 的上一個頁面中，從 [連線定義] 組合方塊中選取了一個類別。如果此類別位於伺服器的類別路徑中，則 [附加特性] 表將顯示預設特性。
- 9 按一下 [完成]。

更多資訊 等效的 asadmin 指令

```
create-connector-connection-pool
```

▼ 編輯連接器連線池

- 1 在樹形元件中，展開 [資源] 節點，然後展開 [連接器] 節點。
- 2 展開 [連接器連線池] 節點。
- 3 選擇要編輯的池的節點。
- 4 在 [編輯連接器連線池] 頁面中，您可以變更控制池中連線的數目的設定。請參閱下表。

參數	說明
池的初始大小和最小大小	池中連線的最小數目。該值還確定了首次建立池或應用程式伺服器啟動時置於池中的連線的數目。
池的最大大小	池中連線的最大數目。
池設定大小數量	當池向最小池大小收縮時，將成批調整大小。此值確定批次中的連線數目。將該值設置過大會延遲連線循環；而將該值設置過小則會導致效率太低。
閒置逾時	連線可以在儲存區中閒置的最大時間 (以秒表示)。一旦超過此時間，即從池中移除該連線。
最長等待時間	已請求連線的應用程式在達到連線逾時之前等待的時間。由於預設等待時間過長，應用程式可能會出現無限期當機的情況。

參數	說明
一旦失敗	<p>選擇標記為 [關閉所有連線] 的核取方塊之後，如果單一連線失敗，Application Server 將關閉池中的所有連線，然後重新建立這些連線。如果未選取此核取方塊，則僅在使用個別連線時才會重新建立這些連線。</p>
作業事件支援	<p>使用 [作業事件支援] 清單可以為連線池選取作業事件支援類型。選擇的作業事件支援以向下相容方式置換與此連線池關聯的資源介面中的作業事件支援屬性。也就是說，它可以支援比資源介面中指定的作業事件層級低或與其相同的作業事件層級，但它不能指定更高的層級。</p> <p>作業事件支援選項包括以下內容：</p> <p>[作業事件支援] 功能表中的 [無] 選項表示資源介面不支援本機資源管理員或 JTA 作業事件，並且不實作 XAResource 或 LocalTransaction 介面。</p> <p>[本機] 作業事件支援表示資源介面將透過實作 LocalTransaction 介面來支援本機作業事件。本機作業事件的管理在資源管理員內部進行，不涉及任何外部作業事件管理員。</p> <p>XA 作業事件支援表示資源介面將透過實作 LocalTransaction 和 XAResource 介面來支援本機資源管理員和 JTA 作業事件。XA 作業事件由作業事件管理員在資源管理員外部進行控制和協調。本機作業事件的管理在資源管理員內部進行，不涉及任何外部作業事件管理員。</p>

- 5 在 [附加特性] 表中，指定名稱-值對。
指定的特性取決於此池使用的資源介面。部署程式使用此表指定的名稱-值對可用於置換由資源介面供應商定義的特性的預設值。
- 6 在 [安全性對映] 標籤窗格中，建立或修改用於連線池的安全性對映。
請參閱第 181 頁的「關於安全性對映」，以取得有關如何建立安全對映的資訊。
- 7 按一下 [儲存]。

▼ 刪除連接器連線池

- 1 在樹形元件中，展開 [資源] 節點，然後展開 [連接器] 節點。
- 2 選取 [連接器連線池] 節點。
- 3 在 [連接器連線池] 頁面中，選取要刪除池的核取方塊。
- 4 按一下 [刪除]。

更多資訊 等效的 asadmin 指令

```
delete-connector-connection-pool
```

有關連接器資源的管理主控台作業

- 第 125 頁的「建立連接器資源」
- 第 126 頁的「編輯連接器資源」
- 第 126 頁的「刪除連接器資源」
- 第 126 頁的「配置連接器服務」

▼ 建立連接器資源

連接器資源 (資料源) 為應用程式提供了與 EIS 的連線。

開始之前 建立連接器資源之前，請先建立連接器連線池。

- 1 在樹形元件中，展開 [資源] 節點，然後展開 [連接器] 節點。
- 2 展開 [連接器資源] 節點。
- 3 在 [連接器資源] 頁面中，按一下 [新建]。
- 4 在 [建立連接器資源] 頁面中，指定資源的設定：
 - a. 在 [JNDI 名稱] 欄位中，鍵入專屬名稱，例如：eis/myERP。
請勿遺漏正斜線。
 - b. 從 [池名稱] 組合方塊中選擇新連接器資源所屬的連線池。
 - c. 若要將資源變更為不可用，請選取 [在所有目標上停用] 單選按鈕。
依預設，建立資源之後立即可以使用資源 (已啟用)。
 - d. 在此頁面的 [目標] 區段中，從 [可用] 欄位中選取連接器資源要常駐的網域、叢集或伺服器實例，然後按一下 [增加]。
如果您不想將連接器資源部署到 [已選取] 欄位中所列的某個網域、叢集或伺服器實例，請從欄位中選取該實例，然後按一下 [移除]。
- 5 按一下 [確定]。

更多資訊 等效的 asadmin 指令

```
create-connector-resource
```

▼ 編輯連接器資源

- 1 在樹形元件中，展開 [資源] 節點，然後展開 [連接器] 節點。
- 2 展開 [連接器資源] 節點。
- 3 選取要編輯的連接器資源的節點。
- 4 在 [編輯連接器資源] 頁面中，您可以從 [池名稱] 功能表中選取不同的連線池。
- 5 在 [目標] 標籤窗格中，可以透過按一下 [管理目標] 來編輯連接器資源部署到的目標。請參閱第 125 頁的「建立連接器資源」，以取得有關目標的更多資訊。
- 6 按一下 [儲存] 以套用編輯。

▼ 刪除連接器資源

- 1 在樹形元件中，展開 [資源] 節點，然後展開 [連接器] 節點。
- 2 選取 [連接器資源] 節點。
- 3 在 [連接器資源] 頁面中，選取要刪除的資源的核取方塊。
- 4 按一下 [刪除]。

更多資訊 等效的 asadmin 指令

```
delete-connector-resource
```

▼ 配置連接器服務

使用 [連接器服務] 螢幕可以為部署到此叢集或伺服器實例的所有資源介面配置連接器容器。

- 1 從樹中選擇 [配置]。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 `server`，請選取 `server-config` 節點。

- 若要為將來的實例 (實例使用 default-config 的副本) 配置預設設定，請選取 default-config 節點。
- 3 選取 [連接器服務] 節點。
 - 4 在 [關閉逾時] 欄位中指定關閉逾時 (以秒為單位)。
請輸入一個整數，此整數表示 Application Server 可以等待連接器模組實例的 ResourceAdapter.stop 方法完成的秒數。應用程式伺服器將忽略所需時間超過指定關閉逾時的的資源介面，並且關閉程序將繼續。預設的關閉逾時為 30 秒。按一下 [載入預設值]，可以為部署到此叢集或伺服器實例的資源介面選取預設關閉逾時。

有關管理物件資源的管理主控台作業

- [第 127 頁的「建立管理物件資源」](#)
- [第 128 頁的「編輯管理物件資源」](#)
- [第 128 頁的「刪除管理物件資源」](#)

▼ 建立管理物件資源

封裝在資源介面 (連接器模組) 中的受管理物件為應用程式提供專用功能。例如，受管理物件可以提供對特定於資源介面及其關聯的 EIS 的剖析器的存取。可以管理該物件，亦即，管理員可配置該物件。若要配置物件，請在 [建立或編輯管理物件資源] 頁面上增加名稱-值特性對。建立受管理物件資源時，請將受管理物件與 JNDI 名稱相關聯。

Application Server 使用資源介面實作 JMS。對於已建立的每個 JMS 目標，Application Server 均會自動建立一個管理物件資源。

- 1 在樹形元件中，展開 [資源] 節點，然後展開 [連接器] 節點。
- 2 展開 [管理物件資源] 節點。
- 3 在 [受管理物件資源] 頁面中，按一下 [新建]。
- 4 在 [受管理物件資源] 頁面中，指定以下設定：
 - a. 在 [JNDI 名稱] 欄位中，鍵入識別資源的專屬名稱。
 - b. 在 [資源類型] 欄位中，輸入資源的 Java 類型。
 - c. 從 [資源介面] 組合方塊中，選取包含受管理物件的資源介面。
 - d. 選取或取消選取核取方塊以啟用或停用資源。

- e. 按[下一步]。
- 5 在[建立受管理物件資源]的第二個頁面中，可以執行以下作業。
 - a. 若要使用名稱-值特性對配置受管理物件，請按一下[新增特性]。
 - b. 在此頁面的[目標]區段中，從[可用]欄位中選取管理物件要常駐的網域、叢集或伺服器實例，然後按一下[增加]。
若要從[已選取]欄位中所列的某個網域、叢集或伺服器實例中取消部署管理物件，請從該欄位中選取該項，然後按一下[移除]。
- 6 按一下[完成]。

更多資訊 **等效的 asadmin 指令**

```
create-admin-object
```

▼ 編輯管理物件資源

- 1 在樹形元件中，展開[資源]節點，然後展開[連接器]節點。
- 2 展開[受管理物件資源]節點。
- 3 選取要編輯的受管理物件資源的節點。
- 4 在[編輯受管理物件資源]頁面中，修改在[建立受管理物件資源]中指定的值。
- 5 在[目標]標籤窗格中，可以透過按一下[管理目標]來編輯受管理物件部署到的目標。請參閱第 127 頁的「[建立管理物件資源](#)」，以取得有關目標的更多資訊。
- 6 按一下[儲存]以套用編輯。

▼ 刪除管理物件資源

- 1 在樹形元件中，展開[資源]節點，然後展開[連接器]節點。
- 2 選取[受管理物件資源]節點。
- 3 在[受管理物件資源]頁面中，選取要刪除的資源的核取方塊。
- 4 按一下[刪除]。

更多資訊 **等效的 asadmin 指令**

`delete-admin-object`

J2EE 容器

本章說明如何配置伺服器中包含的 J2EE 容器。它包含以下小節：

- 第 131 頁的「關於 J2EE 容器」
- 第 132 頁的「有關 J2EE 容器的管理主控台作業」

關於 J2EE 容器

本小節說明 Application Server 中包含的 J2EE 容器。

- 第 131 頁的「J2EE 容器的類型」
- 第 131 頁的「Web 容器」
- 第 131 頁的「EJB 容器」

J2EE 容器的類型

J2EE 容器為 J2EE 應用程式元件提供執行階段支援。J2EE 應用程式元件使用容器的協定和方法存取伺服器提供的其他應用程式元件和服務。Application Server 提供應用程式用戶端容器、applet 容器、Web 容器和 EJB 容器。如需有關顯示容器的圖解，請參閱第 32 頁的「Application Server 架構」。

Web 容器

Web 容器是容納 Web 應用程式的 J2EE 容器。Web 容器透過為開發者提供執行 servlet 和 JavaServer Page (JSP 檔案) 的環境，延伸了 Web 伺服器的功能。

EJB 容器

企業 Bean (EJB 元件) 是包含商務邏輯的 Java 程式設計語言伺服器元件。EJB 容器提供對企業 Bean 的本機和遠端存取。

企業 Bean 分為三種類型：階段作業 Bean、實體 Bean 和訊息導引 Bean。階段作業 Bean 表示暫態物件和程序，並且通常由單一用戶端使用。實體 Bean 表示持續性資料，通常維護在資料庫中。訊息導引 Bean 用於將訊息非同步傳送到應用程式模組和服務中。

容器負責建立企業 Bean、將企業 Bean 連結至命名服務以使其他應用程式元件可以存取企業 Bean、確定僅授權的用戶端才能存取企業 Bean 的方法、將 Bean 的狀態儲存到永久性儲存裝置中、快取 Bean 的狀態以及在必要時啟動或鈍化 Bean。

有關 J2EE 容器的管理主控台作業

- 第 132 頁的「配置一般 Web 容器設定」
- 第 134 頁的「配置一般 EJB 設定」
- 第 137 頁的「配置訊息導引 Bean 設定」
- 第 138 頁的「配置 EJB 計時器服務設定」

配置一般 Web 容器設定

在此發行版本中，管理主控台中沒有用於 Web 容器的容器範圍的設定。

配置 Web 容器階段作業

本小節說明 Web 容器中的 HTTP 階段作業設定。HTTP 階段作業是唯一將狀態資料寫入持續性儲存的 Web 階段作業。

- 第 132 頁的「設定階段作業逾時值」
- 第 133 頁的「配置管理員特性」
- 第 134 頁的「配置儲存特性」

▼ 設定階段作業逾時值

- 1 在樹形元件中，選取 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 server，請選取 [server-config] 節點。
 - 若要配置所有實例的預設設定，請選取 [default-config] 節點。
- 3 選取 [Web 容器] 節點。
- 4 按一下 [階段作業特性] 標籤。

- 5 在 [階段作業逾時] 欄位中，輸入階段作業有效的秒數。
- 6 按一下 [儲存]。

▼ 配置管理員特性

階段作業管理員提供配置建立和銷毀階段作業的方式、儲存階段作業狀態的位置以及最大階段作業數的方法。

若要變更階段作業管理員設定，請執行以下步驟：

- 1 在樹形元件中，選取 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 server，請選取 [server-config] 節點。
 - 若要配置所有實例的預設設定，請選取 [default-config] 節點。
- 3 選取 [Web 容器] 節點。
- 4 按一下 [管理員特性] 標籤。
- 5 設定 [清除間隔] 的值。
[清除間隔] 欄位是從儲存中刪除非使用中階段作業資料之前的秒數。
- 6 設定 [最大階段作業數] 的值。
[最大階段作業數] 欄位是允許的最大階段作業數目。
- 7 設定 [階段作業檔案名稱] 的值。
[階段作業檔案名稱] 欄位是包含階段作業資料的檔案。
- 8 設定 [階段作業 ID 產生器類別名稱] 的值。

[階段作業 ID 產生器類別名稱] 欄位使您可以指定用於產生唯一的階段作業 ID 的自訂類別。每個伺服器實例只允許有一個階段作業 ID 產生器類別，並且叢集中的所有實例必須使用同一階段作業 ID 產生器，以防止階段作業金鑰衝突。

自訂階段作業 ID 產生器類別必須實作 `com.sun.enterprise.util.uuid.UuidGenerator` 介面：

```
package com.sun.enterprise.util.uuid;  
  
public interface UuidGenerator {
```

```
public String generateUuid();  
public String generateUuid(Object obj); //obj is the session object  
}
```

類別必須位於 Application Server 類別路徑中。

9 按一下 [儲存]。

▼ 配置儲存特性

1 在樹形元件中，選取 [配置] 節點。

2 選取要配置的實例：

- 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 server，請選取 [server-config] 節點。
- 若要配置所有實例的預設設定，請選取 [default-config] 節點。

3 選取 [Web 容器] 節點。

4 按一下 [儲存特性] 標籤。

5 設定 [清除間隔]。

[清除間隔] 欄位是從儲存中刪除非使用中階段作業資料之前的秒數。

6 按一下 [儲存]。

配置一般 EJB 設定

本小節描述以下適用於伺服器上所有企業 Bean 容器的設定：

- 第 134 頁的「階段作業儲存位置」
- 第 135 頁的「配置 EJB 池設定」
- 第 136 頁的「配置 EJB 快取設定」

若要置換每個容器的預設值，請調整企業 Bean 的 sun-ejb-jar.xml 檔案中的值。如需詳細資訊，請參閱「Application Server Developer's Guide」。

階段作業儲存位置

[階段作業儲存位置] 欄位指定在檔案系統上儲存鈍化 Bean 和持續的 HTTP 階段作業所在的目錄。

鈍化 Bean 是已將其狀態寫入到檔案系統上的檔案中的企業 Bean。通常，鈍化的 Bean 已經閒置了一段時間並且目前未被用戶端存取。

與鈍化 Bean 類似，持續的 HTTP 階段作業是已將其狀態寫入到檔案系統上的檔案中的各個 Web 階段作業。

[確定選項] 欄位用於指定容器如何快取作業事件之間的鈍化實體 Bean 實例。

[選項 B] 用於快取作業事件之間的實體 Bean 實例，並且是預設選項。[選項 C] 用於停用快取。

▼ 配置 EJB 池設定

容器維護了一個企業 Bean 池，以便在不建立 Bean 來實現效能的情況下回應用戶端請求。這些設定僅適用於無狀態階段作業 Bean 和實體 Bean。

當您在使用已部署之企業 Bean 的應用程式中遇到效能問題時，建立池或增加現有池所維護的 Bean 的數目會有助於提高應用程式的效能。

依預設，容器維護企業 Bean 池。

- 1 在樹形元件中，選取 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 server，請選取 [server-config] 節點。
 - 若要配置所有實例的預設設定，請選取 [default-config] 節點。
- 3 選取 [EJB 容器] 節點。
- 4 在 [池設定] 下的 [池的初始大小和最小大小] 欄位中，輸入容器在池中建立的 Bean 的最小數目。
- 5 在 [池的最大大小] 欄位中，輸入任何時候容器在池中維護的 Bean 的最大數目。
- 6 在 [池設定大小數量] 欄位中輸入 Bean 的數目，此數目是當 Bean 處於閒置狀態的時間超過 [池閒置逾時] 中指定的時間時，要從池中移除的 Bean 的數目。
- 7 在 [池閒置逾時] 欄位中輸入在將池中的 Bean 從池中移除之前 Bean 可以保持閒置狀態的時間 (以秒為單位)。
- 8 按一下 [儲存]。
- 9 重新啟動 Application Server。

▼ 配置 EJB 快取設定

容器維護最常用企業 Bean 的企業 Bean 資料快取。這樣，容器可以更快回應來自其他應用程式模組的企業 Bean 資料請求。本小節只適用於有狀態階段作業 Bean 和實體 Bean。

被快取的企業 Bean 處於以下三種狀態之一：使用中、閒置或鈍化。使用中企業 Bean 是目前正被用戶端存取的企業 Bean。閒置企業 Bean 的資料目前儲存在快取中，但無用戶端存取 Bean。鈍化 Bean 的資料是暫時儲存的，如果用戶端請求此 Bean，則會將其資料讀回快取中。

- 1 在樹形元件中，選取 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 server，請選取 [server-config] 節點。
 - 若要配置所有實例的預設定，請選取 [default-config] 節點。
- 3 選取 [EJB 容器] 節點。
- 4 在 [最大快取大小] 欄位中調整最大快取大小。

增加要快取的 Bean 的最大數目，以結束建立和損毀 Bean 的系統耗用。但是，如果增加快取，伺服器將消耗更多記憶體和資源。請確定作業環境足夠用於快取設定。
- 5 在 [快取設定大小數量] 欄位中調整快取設定大小數量。

達到快取的 Bean 的最大數目之後，容器將從備份儲存中移除一些鈍化 Bean，預設定為 32。
- 6 在 [快取閒置逾時] 欄位中，調整為實體 Bean 排程的快取清除速率 (以秒為單位)。

如果快取的實體 Bean 在特定時間內一直處於閒置狀態，它將被鈍化。亦即，將 Bean 的狀態寫入備份儲存。
- 7 在 [移除逾時] 欄位中，調整將有狀態階段作業 Bean 從快取或鈍化儲存中移除之前的時間 (以秒為單位)。
- 8 在 [移除選取策略] 欄位中，配置容器用來移除有狀態階段作業 Bean 的策略。

容器將根據在 [移除選取策略] 欄位中設定的策略決定移除哪個有狀態階段作業 Bean。容器可使用三種可能的策略從快取中移除 Bean：

 - 最近未使用 (NRU)
 - 先入先出 (FIFO)
 - 最近最少使用 (LRU)

NRU 策略可移除最近未使用的 Bean。FIFO 策略移除快取中最早的 Bean。LRU 策略移除最近最少存取的 Bean。依預設，容器使用 NRU 策略。

通常使用 FIFO 策略移除實體 Bean。

- 9 按一下 [儲存]。
- 10 重新啟動 Application Server。

配置訊息導引 Bean 設定

訊息導引 Bean 的池與第 135 頁的「配置 EJB 池設定」中說明的階段作業 Bean 的池類似。依預設，容器維護訊息導引 Bean 的池。

若要調整該池的配置，請執行以下步驟：

▼ 配置 MDB 池設定

- 1 在樹形元件中，選取 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 server，請選取 [server-config] 節點。
 - 若要配置所有實例的預設設定，請選取 [default-config] 節點。
- 3 選取 [EJB 容器] 節點。
- 4 按一下 [MDB 設定] 標籤。
- 5 在 [池設定] 下的 [初始和最小池大小] 欄位中，輸入容器在池中建立的訊息 Bean 的最小數目。
- 6 在 [最大池大小] 欄位中，輸入任何時候容器在池中維護的 Bean 的最大數目。
- 7 請在 [池設定大小數量] 欄位中輸入 Bean 的數目，此數目是當 Bean 處於閒置狀態的時間超過 [池閒置逾時] 中指定的時間時，要從池中移除的 Bean 的數目。
- 8 在 [池閒置逾時] 欄位中輸入在將池中的 Bean 從池中移除之前 Bean 可以保持閒置狀態的時間 (以秒為單位)。
- 9 按一下 [儲存]。

10 重新啟動 Application Server。

配置 EJB 計時器服務設定

計時器服務是由企業 Bean 容器提供的用於排程企業 Bean 使用的通知或事件的持續性和作業事件通知服務。所有企業 Bean (有狀態階段作業 Bean 除外) 均可從計時器服務接收通知。關閉或重新啟動伺服器時，不會銷毀服務設定的計時器。

▼ 配置計時器服務

- 1 在樹形元件中，選取 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 server，請選取 [server-config] 節點。
 - 若要配置所有實例的預設設定，請選取 [default-config] 節點。
- 3 選取 [EJB 容器] 節點。
- 4 按一下 [EJB 計時器服務] 標籤。
- 5 在 [最小傳送間隔] 欄位中設定最小傳送間隔時間 (以毫秒為單位)。
最小傳送間隔是特定計時器的下一個計時器到期之前允許的最小毫秒數。如果此間隔時間設定得過小，可能會導致伺服器超載。
- 6 在 [最大重新傳送次數] 欄位中，設定計時器服務嘗試傳送通知的最大次數。
- 7 在 [重新傳送間隔] 欄位中，設定兩次重新傳送嘗試之間的時間 (以毫秒為單位)。
- 8 按一下 [儲存]。
- 9 重新啟動 Application Server。

▼ 將外部資料庫與計時器服務搭配使用

依預設，計時器服務使用內嵌式資料庫儲存計時器。

- 1 請依循第 84 頁的「[建立 JDBC 資源](#)」中的說明，為資料庫設定 JDBC 資源。
- 2 在 [計時器資料源] 欄位中輸入資源的 JNDI 名稱。

3 按一下 [儲存]。

4 重新啟動 **Application Server**。

在 `install-dir/lib/install/databases/` 中提供了 PointBase 和 Oracle 的計時器資料庫建立檔案範例。

配置安全性

本章說明某些核心 Application Server 的安全性概念，以及如何為 Application Server 配置安全性。本章包含以下主題：

- 第 141 頁的「關於 Application Server 的安全性」
- 第 156 頁的「有關安全性的管理主控台作業」
- 第 159 頁的「有關範圍的管理主控台作業」
- 第 170 頁的「有關 JACC 提供者的管理主控台作業」
- 第 173 頁的「有關稽核模組的管理主控台作業」
- 第 156 頁的「有關安全性的管理主控台作業」
- 第 177 頁的「有關偵聽程式和 JMX 連接器的管理主控台作業」
- 第 181 頁的「有關連接器連線池的管理主控台作業」
- 第 184 頁的「使用證書和 SSL」
- 第 192 頁的「詳細資訊」

關於 Application Server 的安全性

- 第 141 頁的「安全性簡介」
- 第 146 頁的「關於認證與授權」
- 第 148 頁的「瞭解使用者、群組、角色和範圍」
- 第 151 頁的「證書和 SSL 簡介」
- 第 153 頁的「關於防火牆」
- 第 154 頁的「使用管理主控台管理安全性」

安全性簡介

安全性實質上就是資料保護：如何在儲存或傳輸資料時防止對資料進行未授權的存取或竄改。Application Server 具有基於 J2EE 標準的動態可延伸安全性架構。它內置的安全性功能包括密碼學、認證與授權以及公開金鑰基礎架構。Application Server 是基於 Java

安全性模型建置的，該安全性模型使用了一個沙箱，應用程式可以在沙箱中安全地執行，而不會給系統或使用者帶來潛在的危險。本節論述以下主題：

- 第 142 頁的「瞭解應用程式和系統安全性」
- 第 142 頁的「管理安全性的工具」
- 第 143 頁的「管理密碼安全性」
- 第 145 頁的「指定安全性職責」

瞭解應用程式和系統安全性

從廣泛意義上講，應用程式安全性有兩種：

- 在**程式安全性**中，開發者編寫的應用程式碼負責處理安全性事務。做為管理員，您對此機制沒有任何控制權。由於程式化安全性將安全性配置硬編碼到應用程式中而不是透過 J2EE 容器對其進行管理，因此一般不提倡使用這種程式化安全性。
- 在**宣告性安全性**中，容器 (Application Server) 透過應用程式的部署描述元處理安全性。您可以透過直接編輯部署描述元或使用 `deploytool` 等工具來控制宣告性安全性。由於可以在完成應用程式開發之後變更部署描述元，因此宣告性安全性具有更大靈活性。

除了應用程式安全性以外，還有影響 Application Server 系統中所有應用程式的**系統安全性**。

程式安全性受應用程式開發者的控制，因此本文件不對其進行論述；宣告性安全性受應用程式開發者的控制要少一些，本文件中只偶爾涉及到宣告性安全性。本文件主要針對系統管理員，因此主要論述系統安全性。

管理安全性的工具

Application Server 提供了以下用於管理安全性的工具：

- 管理主控台，一種基於瀏覽器的工具，用於配置整個伺服器的安全性，管理使用者、群組和範圍以及執行系統範圍內的其他安全性作業。如需有關管理主控台的一般介紹，請參閱第 33 頁的「管理工具」。如需有關使用管理主控台可以執行之安全性作業的簡介，請參閱第 154 頁的「使用管理主控台管理安全性」。
- `asadmin`，命令行工具，可以執行管理主控台能夠執行的許多作業。您還可以使用 `asadmin` 執行使用管理主控台無法執行的某些作業。您可以從指令提示符號或程序檔中執行 `asadmin` 指令，以自動執行重複的作業。如需有關 `asadmin` 的一般簡介，請參閱第 33 頁的「管理工具」。
- `deploytool`，圖形化封裝和部署工具，用於編輯應用程式部署描述元，從而控制個別應用程式的安全性。由於 `deploytool` 主要針對應用程式開發者，因此本文件未對該工具的使用做詳細說明。如需有關使用 `deploytool` 的說明，請參閱此工具的線上說明以及位於以下位置的「The J2EE 1.4 Tutorial」：
：<http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html>。

Java 2 Platform, Standard Edition (J2SE) 提供了兩個用於管理安全性的工具：

- `keytool`，命令行公用程式，用於管理數位憑證和鍵對。使用 `keytool` 可以管理 `certificate` 範圍內的使用者。
- `policytool`，圖形化公用程式，用於管理系統範圍內的 Java 安全策略。做為管理員，您很少會用到 `policytool`。

如需有關使用 `keytool`、`policytool` 和其他 Java 安全性工具的更多資訊，請參閱位於以下位置的「Java 2 SDK Tools and Utilities」

：<http://java.sun.com/j2se/1.4.2/docs/tooldocs/tools.html#security>。

在 Enterprise Edition 中，還可以使用另外兩個實作網路安全服務 (NSS) 的工具來管理安全性。如需有關 NSS 的更多資訊，請至

<http://www.mozilla.org/projects/security/pki/nss/>。用於管理安全性的工具包括：

- `certutil`，命令行公用程式，用於管理憑證和金鑰資料庫。
- `pk12util`，命令行公用程式，用於以 PKCS12 格式在憑證/金鑰資料庫和檔案之間匯入和匯出金鑰及憑證。

如需有關使用 `certutil`、`pk12util` 和其他 NSS 安全性工具的更多資訊，請參閱位於以下位置的「NSS Security Tools」

：<http://www.mozilla.org/projects/security/pki/nss/tools>。

管理密碼安全性

在 Application Server 的此發行版本中，包含特定網域規格的 `domain.xml` 檔案最初包含純文字形式的 Sun Java System Message Queue 代理程式密碼。`domain.xml` 檔案中包含此密碼的元素為 `jms-host` 元素的 `admin-password` 屬性。由於在安裝期間不能變更此密碼，因此它不會對安全性產生很大的影響。

但是，您可以使用管理主控台增加使用者和資源，並為這些使用者和資源指定密碼。部分密碼將以純文字形式寫入 `domain.xml` 檔案，例如用於存取資料庫的密碼。將這些純文字形式的密碼保存在 `domain.xml` 檔案中可能會破壞安全性。透過執行以下程序，您可以加密 `domain.xml` 中的任何密碼，包括 `admin-password` 屬性或資料庫密碼。

▼ 加密 `domain.xml` 中的密碼

- 1 在 `domain.xml` 檔案常駐的目錄 (依預設，此目錄為 `domain-dir/config`) 中，執行以下 `asadmin` 指令：

```
asadmin create-password-alias --user admin alias-name
```

例如，

```
asadmin create-password-alias --user admin jms-password
```

螢幕將顯示輸入密碼提示 (在本範例中為 `admin`)。請參閱 `create-password-alias`、`list-password-aliases` 和 `delete-password-alias` 指令的線上手冊，以取得更多資訊。

- 2 移除並替代 `domain.xml` 中的密碼。使用 `asadmin set` 指令可以完成此作業。使用 `set` 指令實現此目的的範例如下：

```
asadmin set --user admin server.jms-service.jms-host.  
default_JMS_host.admin-password=${ALIAS=jms-password}
```

- 3 重新啟動相關網域的 Application Server。

保護具有編碼密碼的檔案

某些檔案包含需要使用檔案系統許可權進行保護的編碼密碼。這些檔案包括：

- `domain-dir/master-password`
此檔案包含編碼主密碼，並且應使用檔案系統許可權 600 對其進行保護。
- 對於使用 `--passwordfile` 引數並做為引數傳送給 `asadmin` 的任何已建立之密碼檔案，均應使用檔案系統許可權 600 對其進行保護。

▼ 變更主密碼

主密碼 (MP) 是全局性的共用密碼。它從不用於認證，也從不會在網路上傳輸。此密碼是全局安全性的要點；使用者可以選擇在需要時手動輸入此密碼，也可以將其隱藏在檔案中。它是系統中最敏感的資料。使用者可以透過移除此檔案強制系統提示輸入 MP。變更主密碼後，系統會將其重新儲存到主密碼金鑰存放區中。

- 1 停止網域的 Application Server。使用 `asadmin change-master-password` 指令會提示輸入舊密碼和新密碼，然後重新加密所有附屬項目。例如，

```
asadmin change-master-password>  
Please enter the master password>  
Please enter the new master password>  
Please enter the the new master password again>
```

- 2 重新啟動 Application Server。



注意 - 此時，不能啟動正在執行的伺服器實例並且不能重新啟動正在執行的伺服器實例，除非已變更這些實例所對映的節點代理程式上的 SMP。如果在變更伺服器實例的 SMP 之前重新啟動了該伺服器實例，它將無法啟動。

- 3 一次停止一個節點代理程式及其相關伺服器。再次執行 `asadmin change-master-password` 指令，然後重新啟動節點代理程式及其相關伺服器。
- 4 繼續對下一個節點代理程式執行這些步驟，直到對所有節點代理程式均已執行這些步驟。這樣可以完成輪替變更。

▼ 變更管理員密碼

第 143 頁的「管理密碼安全性」中論述了如何加密管理員密碼。強烈建議您加密管理員密碼。若要在加密管理員密碼之前變更管理員密碼，請使用 `asadmin set` 指令。使用 `set` 指令實現此目的的範例如下：

```
asadmin set --user admin
server.jms-service.jms-host.default_JMS_host.admin-password=new_pwd
```

還可以使用 管理主控台 變更管理員密碼，如以下程序所示。

- 1 在管理主控台 樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定實例，請展開該實例的配置節點。例如，對於預設實例 `server`，請展開 [server-config] 節點。
 - 若要為所有實例配置預設設定，請展開 [default-config] 節點。
- 3 展開 [安全性] 節點。
- 4 展開 [範圍] 節點。
- 5 選取 [admin-realm] 節點。
- 6 在 [編輯範圍] 頁面中，按一下 [管理使用者] 按鈕。
- 7 選取名為 `admin` 的使用者。
- 8 輸入新密碼並確認密碼。
- 9 按一下 [儲存] 以儲存新密碼，或者按一下 [關閉] 以關閉頁面而不儲存新密碼。

指定安全性職責

將為以下角色指定安全性職責：

- 第 146 頁的「應用程式開發人員」
- 第 146 頁的「應用程式部署人員」
- 第 146 頁的「系統管理員」

應用程式開發人員

應用程式開發者負責：

- 為應用程式元件指定角色和基於角色的存取限制。
- 定義應用程式的認證方法並指定受保護的應用程式部分。

應用程式部署者可以使用 `deploytool` 等工具編輯應用程式部署描述元。「The J2EE 1.4 Tutorial」中的「Security」一章詳細論述了這些安全性作業，您可以從以下 URL 中檢視該指導文件：<http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html>。

應用程式部署人員

應用程式部署人員負責：

- 將使用者或群組 (或兩者) 對映至安全性角色。
- 存取元件方法所需的權限以滿足特定部署方案的要求。

應用程式部署者可以使用 `deploytool` 等工具編輯應用程式部署描述元。「The J2EE 1.4 Tutorial」中的「Security」一章詳細論述了這些安全性作業，您可以從以下 URL 中檢視該指導文件：<http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html>。

系統管理員

系統管理員負責：

- 配置安全性範圍。
- 管理使用者帳戶和群組。
- 管理稽核記錄。
- 管理伺服器憑證，並配置伺服器對安全套接字層 (SSL) 的使用。
- 處理系統範圍內的其他各類安全性功能，例如**連接器連線池的安全性對映**、其他 JACC 提供者等等。

系統管理員使用**管理主控台**管理伺服器安全性設定，並使用 `certutil` 管理憑證。本文件主要適用於系統管理員。

關於認證與授權

認證與授權是應用程式伺服器安全性的核心概念。以下主題論述了與認證和授權相關的內容：

- [第 147 頁的「認證實體」](#)
- [第 148 頁的「對使用者進行授權」](#)
- [第 148 頁的「指定 JACC 提供者」](#)
- [第 148 頁的「稽核認證與授權決策」](#)

- 第 148 頁的「配置訊息安全性」

認證實體

認證是實體 (使用者、應用程式或元件) 用於確定其他實體是否是其宣告的實體的方法。實體使用**安全性憑證**對其自身進行認證。憑證可以是使用者名稱和密碼、數位證書或其他憑證。

通常，認證表示使用者使用使用者名稱和密碼登入某個應用程式；也可以指 EJB 從伺服器請求資源時，提供安全性憑證。通常，伺服器或應用程式要求使用者端進行認證；另外，使用者端也可以要求伺服器對其自身進行認證。如果認證是雙向的，則稱為相互認證。

當實體嘗試存取受保護的資源時，Application Server 將使用為該資源配置的認證機制來確定是否授予存取權。例如，使用者可以在 Web 瀏覽器中輸入使用者名稱和密碼，如果應用程式順利完成了對這些憑證的驗證，則表示該使用者已通過認證。在此階段作業的剩餘時間內，該使用者將始終與這個經過認證的安全性身份相關聯。

Application Server 支援四種類型的認證，如第 147 頁的「認證實體」所示。應用程式在其部署描述元中指定所使用的認證類型。如需有關使用 `deploytool` 配置應用程式認證方法的更多資訊，請參閱位於以下位置的「The J2EE 1.4 Tutorial」

：<http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html>。

表 9-1 Application Server 認證方法

認證方法	通訊協定	說明	使用者憑證加密
基本	HTTP (SSL 選取性)	使用伺服器的內建快顯式登入對話方塊。	無，除非使用 SSL。
基於表單	HTTP (SSL 選取性)	應用程式提供它自己的自訂登入頁面和錯誤頁面。	無，除非使用 SSL。
使用者端證書	HTTPS (基於 SSL 的 HTTP)	伺服器使用公開金鑰證書認證使用者端。	SSL

驗證單次登入

單次登入可以讓一個虛擬伺服器實例中的多個應用程式共用使用者認證狀態。使用單次登入，登入到一個應用程式的使用者也會隱式登入到需要相同認證資訊的其他應用程式。

單次登入基於群組。其部署描述元可定義相同的**群組**並可使用相同認證方法 (基本、表單、摘要、憑證) 的所有 Web 應用程式均共用單次登入。

對於為 Application Server 定義的虛擬伺服器，依預設已啟用單次登入。如需有關停用單次登入的資訊，請參閱第 180 頁的「配置單次登入 (SSO)」。

對使用者進行授權

使用者通過認證後，**授權層級**將決定該使用者可以執行哪些作業。使用者的授權基於其**角色**。例如，人力資源應用程式可以授權管理者檢視所有員工的個人資訊，但每個員工僅能檢視自己的個人資訊。如需有關角色的更多資訊，請參閱第 148 頁的「[瞭解使用者、群組、角色和範圍](#)」。

指定 JACC 提供者

JACC (Java 容器授權合約) 屬於 J2EE 1.4 規格，它定義可插接式授權提供者介面。這可以讓管理員設置協力廠商外掛程式模組以執行授權。

依預設，Application Server 提供符合 JACC 規格並基於檔案的簡單授權引擎。還可以指定其他協力廠商 JACC 提供者。

JACC 提供者使用 Java 認證與授權服務 (JAAS) API。JAAS 可以讓服務對使用者進行認證並強制對使用者進行存取控制。JAAS 實作了 Java 技術版本的標準可插接式認證模組 (PAM) 框架。

稽核認證與授權決策

Application Server 可以透過**稽核模組**提供對所有認證與授權決策的稽核追蹤。

Application Server 提供預設的稽核模組，還提供了自訂稽核模組的功能。如需有關開發自訂稽核模組的資訊，請參閱「[Application Server Developer's Guide](#)」。

配置訊息安全性

訊息安全性可以讓伺服器在訊息層執行 Web 服務呼叫和回應的點對點認證。

Application Server 使用 SOAP 層上的訊息安全性提供者實作訊息安全性。訊息安全性提供者提供以下資訊，例如請求訊息和回應訊息所需的認證類型。受支援的認證類型包括：

- 寄件者認證，包括使用者名稱密碼認證。
- 特性認證，包括 XML 數位簽名。

此發行版本包括兩個訊息安全性提供者。可以為 SOAP 層的認證配置訊息安全性提供者。可以配置的提供者包括 `ClientProvider` 和 `ServerProvider`。

訊息層安全性支援以(可插接式)認證模組的形式整合在 Application Server 及其用戶端容器中。依預設，Application Server 中的訊息層安全性處於停用狀態。

您可以為整個 Application Server 或者為特定的應用程式或方法配置訊息層安全性。第 10 章中論述了如何配置 Application Server 層級的訊息安全性。Developer's Guide 中的「[Securing Applications](#)」一章論述了如何配置應用程式層級的訊息安全性。

瞭解使用者、群組、角色和範圍

Application Server 對以下實體強制執行其認證與授權策略：

- 第 149 頁的「使用者」：*Application Server* 中定義的個人身份。通常，使用者是指個人、一個軟體元件 (例如企業 Bean)，甚至是一種服務。經過認證的使用者有時被稱為主體。使用者有時被稱為個人。
- 第 150 頁的「群組」：*Application Server* 中定義的一組使用者，依循一般特性進行分類。
- 第 150 頁的「角色」：由應用程式定義的命名授權層級。可以將角色比作開鎖的鑰匙。許多人都可以有此鑰匙的複製鑰匙。鎖不關心誰要造訪，而只關心使用的鑰匙是否正確。
- 第 150 頁的「範圍」：包含使用者和群組資訊及其相關安全性憑證的儲存庫。範圍也稱為安全策略網域。

備註 - 儘管使用者和群組是為整個 *Application Server* 指定的，但是每個應用程式都需要定義自己的角色。封裝和部署應用程式時，應用程式會指定使用者/群組和角色之間的對映，如下圖所示。

使用者

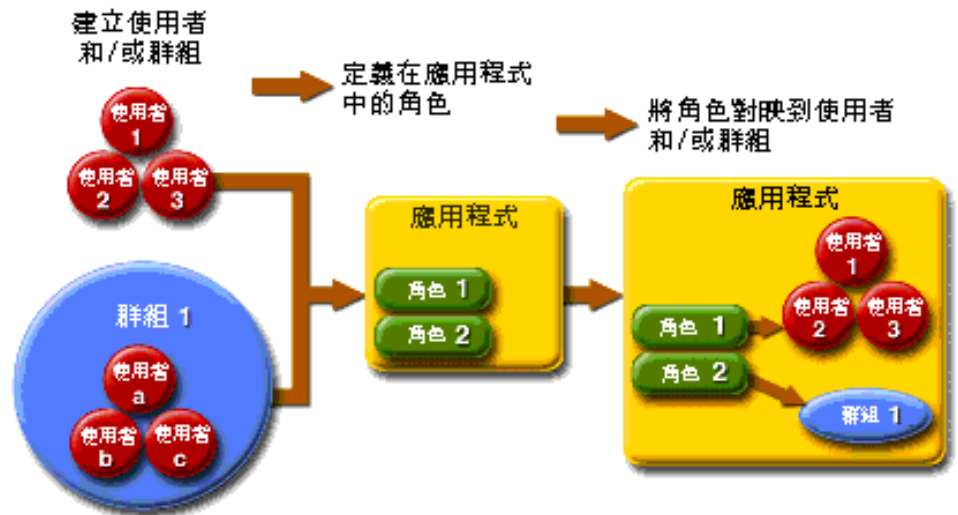


圖 9-1 角色對映

使用者是已在 *Application Server* 中定義的個人 (或應用程式) 身份。使用者可以與群組關聯。*Application Server* 認證服務可以管理多個範圍中的使用者。

群組

J2EE 群組 (或簡稱群組) 是依循一般特性 (例如職務或用戶設定檔) 進行分類的使用者類別。例如, 假定電子商務應用程式的使用者屬於 `customer` 群組, 但是大用戶可以屬於 `preferred` 群組。將使用者進行群組分類可以簡化有大量使用者時的存取控制。

角色

角色定義使用者可以存取哪些應用程式以及每個應用程式的哪些部分, 以及使用者可以執行的作業。亦即, 角色決定了使用者的授權層級。

例如, 假定在人事應用程式中, 所有員工均可以存取電話號碼和電子郵件位址但只有管理者才能存取薪水資訊。該應用程式至少可以定義兩個角色: `employee` 和 `manager`; 僅允許 `manager` 角色中的使用者檢視薪水資訊。

角色與使用者群組的不同之處在於, 角色在應用程式中定義功能, 而使用者群組是以某一方式相關的一組使用者。例如, 假定在人事應用程式中有 `full-time`、`part-time` 和 `on-leave` 幾個群組, 但所有這些群組中的使用者仍是 `employee` 角色。

角色是在應用程式部署描述元中定義的。相反, 群組是針對整個伺服器和範圍而定義的。應用程式開發者或部署者在每個應用程式的部署描述元中將角色對映至一個或多個群組。

範圍

範圍 (也稱為安全策略網域或安全網域), 是伺服器定義和強制執行一般安全策略的範圍。在實際應用中, 範圍是伺服器儲存使用者和群組資訊的儲存庫。

Application Server 預先配置了三個範圍: `file` (初始預設範圍)、`certificate` 和 `admin-realm`。您還可以設定 `ldap`、`solaris` 範圍或自訂範圍。應用程式可以在其部署描述元中指定要使用的範圍。如果應用程式不指定範圍, Application Server 將使用其預設範圍。

在 `file` 範圍中, 伺服器將使用者憑證儲存在本地名為 `keyfile` 的檔案中。您可以使用管理主控台來管理 `file` 範圍中的使用者。如需更多資訊, 請參閱第 167 頁的「管理 `file` 範圍使用者」。

在 `certificate` 範圍中, 伺服器將使用者憑證儲存在憑證資料庫中。使用 `certificate` 範圍時, 伺服器結合使用憑證和 HTTPS 通訊協定認證 Web 用戶端。如需有關憑證的更多資訊, 請參閱第 151 頁的「證書和 SSL 簡介」。

`admin-realm` 也是一個 `FileRealm`, 它將管理員使用者憑證儲存在本地名為 `admin-keyfile` 的檔案中。您可以使用管理主控台管理此範圍中的使用者, 其方法與您管理 `file` 範圍中的使用者的方法相同。如需更多資訊, 請參閱第 167 頁的「管理 `file` 範圍使用者」。

在 ldap 範圍中，伺服器將從簡易資料存取協定 (LDAP) 伺服器 (例如 Sun Java System Directory Server) 中取得使用者憑證。LDAP 是一種可以讓任何人在網路 (無論是公共網際網路還是企業內部網路) 中尋找組織、個人和其他資源 (例如檔案和裝置) 的協定。如需有關管理 ldap 範圍中的使用者和群組的資訊，請參閱您的 LDAP 伺服器文件。

在 solaris 範圍中，伺服器將從 Solaris 作業系統中取得使用者憑證。Solaris 9 OS 和更高版本支援此範圍。如需有關管理 solaris 範圍中的使用者和群組的資訊，請參閱您的 Solaris 文件。

自訂範圍是使用者憑證的任何其他儲存庫，例如關聯式資料庫或協力廠商元件。如需更多資訊，請參閱第 164 頁的「[建立自訂範圍](#)」。

證書和 SSL 簡介

本節論述以下主題：

- [第 151 頁的「關於數位證書」](#)
- [第 152 頁的「關於安全套接字層」](#)

關於數位證書

數位憑證 (或簡稱憑證) 是在網際網路上唯一識別人員和資源的電子檔案。證書還可以使兩個實體之間能夠進行安全、機密的通訊。

證書有很多種類型，例如個人證書 (由個人使用) 和伺服器證書 (用於透過安全套接字層 [SSL] 技術在伺服器和使用者端之間建立安全階段作業)。如需有關 SSL 的更多資訊，請參閱第 152 頁的「[關於安全套接字層](#)」。

憑證以**公開金鑰密碼學**為基礎，公開金鑰密碼學使用數位**金鑰** (很長的數位) 對資訊進行**加密**或編碼，從而使資訊只能被指定的接收者讀取。然後，接收者對資訊進行**解密** (解碼)，即可讀取該資訊。

一個金鑰對包含一個公開金鑰和一個私密金鑰。所有者對公開金鑰進行發放並使任何人都可以使用該公開金鑰。但是所有者永遠不會發放私密金鑰；私密金鑰始終是保密的。由於金鑰與數學相關，因此使用了金鑰對中的一個金鑰進行加密的資料只能透過金鑰對中的另一個金鑰進行解密。

證書就好像一本護照：它可以識別持有者並提供其他重要資訊。憑證由稱為**憑證授權單位 (CA)** 的可信任的協力廠商發行。CA 類似於護照申領辦公室：它將驗證憑證持有者的身份並對憑證進行簽名，以使他人無法偽造或竄改憑證。CA 對證書進行簽名之後，持有者可以提供該證書做為身份證明並建立加密的機密通訊。

最重要的是，憑證會將所有者的公開金鑰連結至所有者身份。與護照將照片連結至其持有者的個人資訊類似，證書將公開金鑰連結至有關其所有者的資訊。

除了公開金鑰以外，證書通常還包括以下資訊：

- 持有者的姓名和其他標識，例如使用憑證的 Web 伺服器的 URL 或個人的電子郵件位址。
- 發行證書的 CA 的名稱。
- 過期日期。

數位憑證受 x.509 格式的技術規格約束。若要驗證 certificate 範圍中某個使用者的身份，認證服務將使用 X.509 憑證的一般名稱欄位做為主體名稱對 X.509 憑證進行驗證。

關於證書鏈

Web 瀏覽器已預先配置了一組瀏覽器自動信任的**根 CA 憑證**。來自其他憑證授權單位的所有憑證都必須附帶**憑證鏈**，以驗證這些憑證是否有效。憑證鏈是由連續 CA 憑證發行的憑證序列，最終以根 CA 憑證結束。

憑證最初產生時是**自簽名憑證**。自簽名證書是其發行者(簽名者)與主旨(其公開金鑰由該證書進行認證的實體)相同的證書。如果所有者向 CA 傳送證書簽名請求(CSR)，然後輸入回應，自簽名證書將被證書鏈取代。鏈的底部是由 CA 發行的、用於認證主旨的公開金鑰證書(回覆)。鏈中的下一個證書是認證 CA 公開金鑰的證書。通常，這是一個自簽名證書(即，來自 CA、用於認證其自身公開金鑰的證書)，並且是鏈中的最後一個證書。

在其他情況下，CA 可能會傳回一個證書鏈。在此情況下，鏈的底部證書是相同的(由 CA 簽名的證書，用於認證金鑰項目的公開金鑰)，但是鏈中的第二個證書是由其他 CA 簽名的證書，用於認證您向其傳送了 CSR 的 CA 的公開金鑰。然後，鏈中的下一個憑證是用於認證第二個 CA 金鑰的憑證，依此類推，直至到達自簽名的**根憑證**。因此，鏈中的每個證書(第一個證書之後的證書)都需要認證鏈中前一個證書的簽名者的公開金鑰。

關於安全套接字層

安全套接字層 (SSL) 是用於確定網際網路通訊和作業事件保護的最常見標準。Web 應用程式使用 HTTPS(基於 SSL 的 HTTP)，HTTPS 使用數位證書來確保在伺服器和使用者端之間進行安全、機密的通訊。在 SSL 連線中，使用者端和伺服器在傳送資料之前都要對資料進行加密，然後由接受者對其進行解密。

Web 瀏覽器(用戶端)需要與某個安全站點建立連線時，則會發生 **SSL 交換**：

- 瀏覽器將透過網路傳送請求安全階段作業的訊息(通常請求為以 https 開頭的 URL，而非 http 開頭的 URL)。
- 伺服器透過傳送其證書(包括公開金鑰)進行回應。
- 瀏覽器將驗證伺服器憑證是否有效，並驗證該憑證是否由其憑證位於瀏覽器的資料庫中的 CA(並且是可信任的 CA)而簽名。它還驗證 CA 證書是否已過期。
- 如果憑證有效，瀏覽器將產生一個一次性的、唯一的**階段作業金鑰**，並使用伺服器的公開金鑰對該階段作業進行加密。然後，瀏覽器將把加密的階段作業金鑰傳送給伺服器，這樣伺服器和瀏覽器都擁有該階段作業金鑰副本。

- 伺服器可以使用其私密金鑰對訊息進行解密，然後恢復該階段作業金鑰。

交握之後，即表示使用者端已驗證了網站的身份，並且只有該使用者端和 Web 伺服器擁有該階段作業金鑰副本。從現在開始，使用者端和伺服器便可以使用該階段作業金鑰對彼此間的所有通訊進行加密。這樣就確保了使用者端和伺服器之間的通訊的安全。

SSL 標準的最新版本稱為 TLS (傳輸層安全性)。Application Server 支援安全套接字層 (SSL) 3.0 和傳輸層安全性 (TLS) 1.0 加密協定。

若要使用 SSL，Application Server 必須擁有接受安全連線的每個外部介面或 IP 位址的憑證。只有安裝了數位證書之後，大多數 Web 伺服器的 HTTPS 服務才能夠執行。請使用第 187 頁的「使用 keytool 公用程式產生憑證」中說明的程序來設定 Web 伺服器可以用於 SSL 的數位憑證。

關於加密算法

加密算法是用於加密或解密的加密演算法。SSL 和 TLS 協定支援用於伺服器和使用者端彼此進行認證、傳輸證書和建立階段作業金鑰的各種加密算法。

某些加密算法比其他加密算法更強大且更安全。使用者端和伺服器可以支援不同的密碼組。從 SSL3 和 TLS 協定中選取加密算法。在安全連線期間，使用者端和伺服器同意在通訊中使用它們均已啓用的最強大的加密算法，因此通常需要啓用所有加密算法。

使用基於名稱的虛擬主機

對安全應用程式使用基於名稱的虛擬主機可能會帶來問題。這是 SSL 協定本身的設計限制。必須先進行 SSL 交握 (使用者端瀏覽器在此時接受伺服器證書)，然後才能存取 HTTP 請求。這樣，在認證之前就無法確定包含虛擬主機名稱的請求資訊，因此也不能將多個證書指定給單一 IP 位址。

如果單一 IP 位址上的所有虛擬主機均需要對照同一證書進行認證，則增加多個虛擬主機將不會影響伺服器上正常的 SSL 作業。但是請注意，大多數瀏覽器會將伺服器的網域名稱與憑證中列示的網域名稱 (如果有，且主要適用於官方 CA 簽名憑證) 進行對照。如果網域名稱不匹配，這些瀏覽器將顯示警告。通常在生產環境中，只將基於位址的虛擬主機與 SSL 配合使用。

關於防火牆

防火牆控制兩個或多個網路之間的資料流，並管理網路之間的連結。防火牆可能包含硬體和軟體元素。本節描述了一些一般防火牆架構及其配置。此處的資訊主要針對 Application Server。如需有關特定防火牆技術的詳細資訊，請參閱防火牆供應商提供的文件。

通常，需要對防火牆進行配置，以便使用者端存取所需的 TCP/IP 連接埠。例如，如果 HTTP 偵聽程式正在連接埠 8080 上作業，則將防火牆配置為僅允許處理連接埠 8080 上的 HTTP 請求。同樣地，如果為連接埠 8181 設定了 HTTPS 請求，則必須將防火牆配置為允許處理連接埠 8181 上的 HTTPS 請求。

如果需要從網際網路對 EJB 模組進行直接的 RMI-IIOP (全稱為 Remote Method Invocations over Internet Inter-ORB Protocol [通過網際網路 ORB 交換協定的遠端方法呼叫]) 存取，則還需要開啓 RMI-IIOP 偵聽程式連接埠，但強烈建議您不要這樣做，因為這樣可能會破壞安全性。

在雙防火牆架構中，您必須將外部防火牆配置為允許處理 HTTP 和 HTTPS 作業事件。您必須將內部防火牆配置為允許 HTTP 伺服器外掛程式與防火牆後面的 Application Server 進行通訊。

使用管理主控台管理安全性

管理主控台 提供了對安全性的以下方面進行管理的方法：

- [第 154 頁的「伺服器安全性設定」](#)
- [第 154 頁的「範圍和 file 範圍使用者」](#)
- [第 155 頁的「JACC 提供者」](#)
- [第 155 頁的「稽核模組」](#)
- [第 155 頁的「訊息安全性」](#)
- [第 155 頁的「HTTP 和 IIOP 偵聽程式安全性」](#)
- [第 156 頁的「管理服務安全性」](#)
- [第 156 頁的「安全性對映」](#)

伺服器安全性設定

在 [安全性設定] 頁面中，設定整個伺服器的特性，包括指定預設範圍、匿名角色和預設的主體使用者名稱和密碼。如需更多資訊，請參閱 [第 156 頁的「配置安全性設定」](#)。

範圍和 file 範圍使用者

[第 148 頁的「瞭解使用者、群組、角色和範圍」](#) 中介紹了範圍的概念。

- 建立新範圍
- 刪除現有範圍
- 修改現有範圍的配置
- 增加、修改和刪除 file 範圍中的使用者
- 設定預設範圍

請參閱 [第 159 頁的「有關範圍的管理主控台作業」](#)，以取得有關這些作業的詳細資訊。

JACC 提供者

第 148 頁的「指定 JACC 提供者」中介紹了 JACC 提供者。使用 管理主控台 執行以下作業：

- 增加新的 JACC 提供者
- 刪除或修改現有 JACC 提供者

請參閱第 170 頁的「有關 JACC 提供者的 管理主控台 作業」，以取得有關這些作業的詳細資訊。

稽核模組

第 148 頁的「稽核認證與授權決策」中介紹了稽核模組。稽核是記錄重要事件 (例如錯誤或安全性漏洞) 以進行後續檢查的方法。所有認證事件都被記錄到 Application Server 記錄中。完整的存取記錄提供了 Application Server 存取事件的循序追蹤。

使用 管理主控台 執行以下作業：

- 增加新的稽核模組
- 刪除或修改現有稽核模組

請參閱第 173 頁的「有關稽核模組的 管理主控台 作業」，以取得有關這些作業的詳細資訊。

訊息安全性

第 148 頁的「配置訊息安全性」中介紹了訊息安全性的概念。使用 管理主控台 執行以下作業：

- 啟用訊息安全性
- 配置訊息安全性提供者
- 刪除或配置現有訊息安全性配置或提供者

請參閱第 10 章，以取得有關這些作業的詳細資訊。

HTTP 和 IIOP 偵聽程式安全性

HTTP 服務中的每個虛擬伺服器都透過一個或多個 *HTTP 偵聽程式* 提供網路連線。如需有關 HTTP 服務和 HTTP 偵聽程式的一般資訊，請參閱第 219 頁的「什麼是 HTTP 服務？」。

Application Server 支援 CORBA (共用物件請求代理程式架構) 物件，這類物件使用網際網路 Orb 交換協定 (IIOP) 在網路上進行通訊。*IIOP 偵聽程式* 接受來自 EJB 的遠端用戶端和其他基於 CORBA 之用戶端的內送連線。如需有關 IIOP 偵聽程式的一般資訊，請參閱第 238 頁的「IIOP 偵聽程式」。

使用 管理主控台 執行以下作業：

- 建立新的 HTTP 或 IIOP 偵聽程式，並指定該偵聽程式所使用的安全性。
- 修改現有 HTTP 或 IIOP 偵聽程式的安全性設定。

請參閱第 177 頁的「有關偵聽程式和 JMX 連接器的管理主控台作業」，以取得有關這些作業的詳細資訊。

管理服務安全性

管理服務決定伺服器實例是一般實例、網域管理伺服器 (DAS)，還是兼具兩者。使用管理服務可以配置 JSR-160 相容遠端 JMX 連接器，該連接器處理網域管理伺服器與遠端伺服器實例的節點代理程式 (管理主機電腦上的伺服器實例) 之間的通訊。

使用管理主控台執行以下作業：

- 管理管理服務
- 編輯 JMX 連接器
- 修改 JMX 連接器的安全性設定

請參閱第 178 頁的「配置管理服務之 JMX 連接器的安全性」，以取得有關這些作業的詳細資訊。

安全性對映

第 181 頁的「關於安全性對映」中介紹了用於連接器連線池的安全性對映的概念。使用管理主控台執行以下作業：

- 將安全性對映增加至現有連接器連線池中
- 刪除或配置現有安全性對映

請參閱第 181 頁的「有關連接器連線池的管理主控台作業」，以取得有關這些作業的詳細資訊。

有關安全性的管理主控台作業

- 第 156 頁的「配置安全性設定」
- 第 158 頁的「授予對管理工具的存取權」
- 第 169 頁的「配置相互認證」
- 第 180 頁的「配置單次登入 (SSO)」

▼ 配置安全性設定

管理主控台 中的 [安全性] 頁面可以讓您設定各種系統範圍內的安全性設定。

- 1 在管理主控台 樹形元件中，展開 [配置] 節點。

2 選取要配置的實例：

- 若要配置特定實例，請展開該實例的配置節點。例如，對於預設實例 server，請展開 [server-config] 節點。
- 若要為所有實例配置預設設定，請展開 [default-config] 節點。

3 選取 [安全性] 節點。

將顯示 [安全性] 頁面。

4 依需要修改值。

下表論述了一般安全性選項。

設定	說明
稽核記錄	選取以啟用稽核記錄。如果啟用稽核記錄，伺服器將載入並執行在 [稽核模組] 設定中指定的所有稽核模組。如果停用稽核記錄，伺服器將不存取稽核模組。依預設，稽核記錄處於停用狀態。
預設範圍	伺服器用於進行認證的使用中 (預設) 範圍。除非應用程式在其部署描述元中指定了其他範圍，否則應用程式將使用此範圍。清單中將顯示所有已配置的範圍。初始預設範圍為 file 範圍。
匿名角色	預設或匿名角色的名稱。匿名角色將被指定給所有使用者。應用程式可以在其部署描述元中使用此角色以向任何人授予許可權。
預設主體	指定預設使用者名稱。如果未提供任何主體，伺服器將使用此預設主體。如果在此欄位中輸入了值，則請在 [預設主體密碼] 欄位中輸入相應的值。 一般伺服器作業不需要此屬性。
預設主體密碼	在 [預設主體] 欄位中指定的預設主體的密碼。 一般伺服器作業不需要此屬性。
JACC	已配置的 JACC 提供者的類別名稱。請參閱第 170 頁的「 建立 JACC 提供者 」
稽核模組	稽核模組提供者類別清單 (以逗號分隔)。此處所列示的模組必須是已配置模組。如果啟用 [稽核記錄]，則此設定必須列示稽核模組。依預設，伺服器使用名為 default 的稽核模組。如需有關建立新稽核模組的資訊，請參閱第 173 頁的「 建立稽核模組 」。

5 在 [附加特性] 區段輸入要傳送至 Java 虛擬機器 (JVM) 的附加特性。

有效的特性取決於 [預設範圍] 欄位中所選取範圍的類型。在以下小節中會論述有效特性：

- 第 166 頁的「[編輯 file 和 admin-realm 範圍](#)」
- 第 165 頁的「[編輯 certificate 範圍](#)」
- 第 164 頁的「[建立 solaris 範圍](#)」

- 第 162 頁的「建立 ldap 範圍」
 - 第 164 頁的「建立自訂範圍」
- 6 選取 [儲存] 以儲存變更，或者選取 [載入預設值] 以復原預設值。

▼ 授予對管理工具的存取權

僅 asadmin 群組中的使用者才能存取 管理主控台 和 asadmin 命令行公用程式。

若要授予使用者對這些管理工具的存取權，請將這些工具增加至 admin-realm 的 asadmin 群組中。

- 1 在管理主控台 樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定實例，請展開該實例的配置節點。例如，對於預設實例 server，請展開 [server-config] 節點。
 - 若要為所有實例配置預設設定，請展開 [default-config] 節點。
- 3 展開 [安全性] 節點。
- 4 展開 [範圍] 節點。
- 5 選取 [admin-realm] 節點。
- 6 在 [編輯範圍] 頁面中，按一下 [管理使用者] 按鈕。

初次完成安裝之後，安裝期間輸入的管理員使用者名稱和密碼將列示在名為 admin-keyfile 的檔案中。依預設，此使用者屬於 asadmin 群組，該群組可授予修改 Application Server 的權限。請僅在您要為使用者授予 Application Server 的管理員權限時，才將使用者指定至該群組。

如果您將使用者增加至 admin-realm 範圍，但將使用者指定給 asadmin 以外的群組，則雖然使用者資訊仍會被寫入名為 admin-keyfile 的檔案，但使用者不具有對管理工具或 file 範圍中的應用程式的存取權。
- 7 按一下 [新建] 以將新使用者增加至 admin-realm 範圍。
- 8 在 [使用者 ID]、[密碼]、和 [群組清單] 欄位中輸入正確資訊。

若要授予使用者修改 Application Server 的權限，請將 asadmin 群組包含在 [群組清單] 中。

- 按一下 [確定] 以將此使用者增加至 `admin-realm` 範圍中，或按一下 [取消] 以退出而不儲存變更。

有關範圍的管理主控台作業

- 第 159 頁的「建立範圍」
- 第 160 頁的「編輯範圍」
- 第 161 頁的「刪除範圍」
- 第 162 頁的「設定預設範圍」
- 第 162 頁的「特定範圍的其他資訊」

▼ 建立範圍

Application Server 預先配置了三個範圍：`file`、`certificate` 和 `admin-realm`。您還可以建立 `ldap`、`solaris` 範圍和自訂範圍。通常，一個伺服器上會有每種類型的一個範圍，但在 Application Server 上有兩個 `file` 範圍：`file` 和 `admin-realm`。這兩個範圍類型相同但用於兩種不同的目的。您的系統也可以為每個虛擬伺服器配備不同的證書資料庫。

- 在管理主控台樹形元件中，展開 [配置] 節點。
- 選取要配置的實例：
 - 若要配置特定實例，請展開該實例的配置節點。例如，對於預設實例 `server`，請展開 [server-config] 節點。
 - 若要為所有實例配置預設設定，請展開 [default-config] 節點。
- 展開 [安全性] 節點。
- 選取 [範圍] 節點。
- 在 [範圍] 頁面中，按一下 [新建]。
將顯示 [建立範圍] 頁面。
- 在 [名稱] 欄位中為範圍輸入名稱。
- 指定要建立的範圍的類別名稱。
下表列出了有效選項：

範圍名稱	類別名稱
file	com.sun.enterprise.security.auth.realm.file.FileRealm
certificate	com.sun.enterprise.security.auth.realm.certificate.CertificateRealm
ldap	com.sun.enterprise.security.auth.realm.ldap.LDAPRealm
solaris	com.sun.enterprise.security.auth.realm.solaris.SolarisRealm
自訂	登入範圍類別的名稱

- 8 增加範圍必需特性和任何所需的選擇性特性。
 - a. 按一下 [增加特性]。
 - b. 在 [名稱] 欄位中，輸入特性的名稱。
 - 如需有關 file 範圍特性的說明，請參閱第 166 頁的「編輯 file 和 admin-realm 範圍」。
 - 如需有關 certificate 範圍特性的說明，請參閱第 165 頁的「編輯 certificate 範圍」。
 - 如需有關 ldap 範圍特性的說明，請參閱第 162 頁的「建立 ldap 範圍」。
 - 如需有關 solaris 範圍特性的說明，請參閱第 164 頁的「建立 solaris 範圍」。
 - 如需有關自訂範圍特性的說明，請參閱第 164 頁的「建立自訂範圍」。
 - c. 在 [值] 欄位中輸入特性的值。
- 9 按一下 [確定]。

更多資訊 等效的 asadmin 指令

```
create-auth-realm
```

▼ 編輯範圍

- 1 在管理主控台 樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定實例，請展開該實例的配置節點。例如，對於預設實例 server，請展開 [server-config] 節點。
 - 若要為所有實例配置預設設定，請展開 [default-config] 節點。
- 3 展開 [安全性] 節點。

- 4 展開 [範圍] 節點。
- 5 選取現有範圍的名稱。
將顯示 [編輯範圍] 頁面。
- 6 依需要編輯現有特性及其值。
- 7 若要增加附加特性，請按一下 [增加特性] 按鈕。
該頁面將顯示一個新行。輸入有效的特性名稱和特性值。
 - 如需有關 file 範圍特性的說明，請參閱第 166 頁的「[編輯 file 和 admin-realm 範圍](#)」。
 - 如需有關 certificate 範圍特性的說明，請參閱第 165 頁的「[編輯 certificate 範圍](#)」。
 - 如需有關 ldap 範圍特性的說明，請參閱第 162 頁的「[建立 ldap 範圍](#)」。
 - 如需有關 solaris 範圍特性的說明，請參閱第 164 頁的「[建立 solaris 範圍](#)」。
 - 如需有關自訂範圍特性的說明，請參閱第 164 頁的「[建立自訂範圍](#)」。
- 8 按一下 [儲存] 以儲存變更。

▼ 刪除範圍

- 1 在管理主控台樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定實例，請展開該實例的配置節點。例如，對於預設實例 server，請展開 [server-config] 節點。
 - 若要為所有實例配置預設設定，請展開 [default-config] 節點。
- 3 展開 [安全性] 節點。
- 4 選取 [範圍] 節點。
- 5 按一下要刪除的範圍旁邊的方塊。
- 6 按一下 [刪除]。

更多資訊 等效的 asadmin 指令

```
delete-auth-realm
```

▼ 設定預設範圍

預設範圍是當應用程式部署描述元未指定範圍時，Application Server 用於認證與授權的範圍。

- 1 在管理主控台 樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定實例，請展開該實例的配置節點。例如，對於預設實例 server，請展開 [server-config] 節點。
 - 若要為所有實例配置預設設定，請展開 [default-config] 節點。
- 3 選取 [安全性] 節點。
將顯示 [安全性] 頁面。
- 4 在 [預設範圍] 欄位中，從下拉式清單中選取所需的範圍。
- 5 按一下 [儲存] 以儲存變更，或按一下 [載入預設值] 以刪除變更並復原 Application Server 的預設值。
- 6 如果主控台中顯示「需要重新啟動」，請重新啟動伺服器。

特定範圍的其他資訊

本小節包含以下主題：

- 第 162 頁的「建立 ldap 範圍」
- 第 164 頁的「建立 solaris 範圍」
- 第 164 頁的「建立自訂範圍」
- 第 165 頁的「編輯 certificate 範圍」
- 第 166 頁的「編輯 file 和 admin-realm 範圍」
- 第 166 頁的「使用網路安全服務 (NSS) 管理使用者」
- 第 167 頁的「管理 file 範圍使用者」
- 第 169 頁的「配置相互認證」

建立 ldap 範圍

ldap 範圍使用 LDAP 伺服器資訊執行認證。使用者資訊包括使用者名稱、密碼和使用
者所屬的群組。若要使用 LDAP 範圍，必須已在 LDAP 目錄中定義了使用者和群組。

若要建立 LDAP 範圍，請依循第 159 頁的「建立範圍」中的步驟增加新範圍，然後增加下表中所示的特性。

表 9-2 ldap 範圍的必需特性

特性名稱	說明	值
directory	目錄伺服器的 LDAP URL。	LDAP URL 的格式為： <code>ldap://hostname:port</code> ，例如： <code>ldap://myldap.foo.com:389</code> 。
base-dn	使用者資料位置的基底辨別名稱 (DN)，由於將執行樹範圍的搜尋，因此該使用者資料的位置可以是高於使用者資料的任何層級。搜尋樹越小，效能越好。	搜尋的網域，例如： <code>dc=siliconvalley, dc=BayArea, dc=sun, dc=com</code> 。
jaas-context	要用於此範圍的登入模組類型。	必須為 <code>ldapRealm</code> 。

下表列出了 ldap 範圍的選擇性特性。

表 9-3 ldap 範圍的選擇性特性

特性名稱	說明	預設
search-filter	用於尋找使用者的搜尋篩選器。	<code>uid=%s</code> (%s 擴展為主旨名稱)。
group-base-dn	群組資料位置的基底 DN。	與 <code>base-dn</code> 相同，但必要時可以進行調校。
group-search-filter	用於尋找使用者的群組成員關係的搜尋篩選器。	<code>uniquemember=%d</code> (%d 擴展為使用者元素 DN)。
group-target	包含群組名稱項目的 LDAP 屬性名稱。	CN
search-bind-dn	用於向目錄認證以執行 <code>search-filter</code> 查找的選擇性 DN。只有不允許進行匿名搜尋的目錄才需要。	
search-bind-password	<code>search-bind-dn</code> 中所指定之 DN 的 LDAP 密碼。	

範例

例如，假定在 LDAP 目錄中定義了一個 LDAP 使用者 Joe Java，如下所示：

```
uid=jjava,ou=People,dc=acme,dc=com
uid=jjava
givenName=joe
objectClass=top
objectClass=person
objectClass=organizationalPerson
objectClass=inetorgperson
sn=java
cn=Joe Java
```

使用範例程式碼，在建立或編輯 ldap 範圍時，您可以輸入下表中所示的值。

表 9-4 ldap 範圍值範例

特性名稱	特性值
directory	伺服器的 LDAP URL，例如： <code>ldap://ldap.acme.com:389</code>
base-dn	<code>ou=People,dc=acme,dc=com</code> 。 可以向更高層次進行搜尋 (例如 <code>dc=acme</code> 、 <code>dc=com</code>)，但搜尋將遍歷樹狀結構的更大部分，從而導致效能降低。
jaas-context	<code>LdapRealm</code>

建立 solaris 範圍

solaris 範圍從基礎 Solaris 使用者資料庫 (由系統配置確定) 取得使用者和群組資訊。solaris 範圍呼叫基礎 PAM 基礎架構以用於認證。如果已配置的 PAM 模組需要超級使用者權限，則網域必須以超級使用者身份執行才能使用該範圍。請參閱 Solaris 文件中有關安全服務的詳細資訊。

solaris 範圍有一個必要特性 `jaas-context`，該特性指定要使用的登入模組類型。特性值必須為 `solarisRealm`。

備註 - 僅 Solaris 9 或更高版本才支援 solaris 範圍。

建立自訂範圍

除了四個內建範圍之外，您還可以建立以其他方式儲存使用者資料 (例如儲存在關聯式資料庫中) 的自訂範圍。自訂範圍的開發不在本文件的論述範圍之內。如需更多資訊，請參閱「Application Server Developer's Guide」中的「Securing Applications」一章。

做為管理員，您需要瞭解的主要事項是：自訂範圍是由源自 Java 認證與授權服務 (JAAS) 套裝軟體的類別 (稱為 `LoginModule`) 實作的。

▼ 建立自訂範圍

- 1 依循第 159 頁的「建立範圍」中說明的程序，輸入自訂範圍的名稱和 `LoginModule` 類別的名稱。

自訂範圍可以使用任何專屬名稱，例如 `myCustomRealm`。

- 2 增加下表中所示的自訂範圍特性。

特性名稱	特性值
jaas-context	LoginModule 類別名稱，例如 simpleCustomRealm
auth-type	對範圍的說明，例如「自訂範圍的簡單範例」。

- 按一下 [確定]。
- 編輯網域的登入配置檔案 `domain-dir/config/login.conf`，並在檔案末尾增加 JAAS LoginModule 之完全合格的類別名稱，如下所示：

```
realmName {
    fully-qualified-LoginModule-classname required;
};
```

例如，

```
myCustomRealm {
    com.foo.bar.security.customrealm.simpleCustomLoginModule required;
};
```

- 將 LoginModule 類別和所有附屬類別複製到 `domain-dir/lib/classes` 目錄中。
- 如果主控台中顯示「需要重新啟動」，請重新啟動伺服器。
- 確定已正確載入範圍。

檢查 `domain-dir/logs/server.log` 以確定伺服器已載入該範圍。伺服器應呼叫該範圍的 `init()` 方法。

編輯 certificate 範圍

certificate 範圍支援 SSL 認證。該範圍在 Application Server 安全性環境中設定使用者身份，並使用從信任庫檔案和金鑰庫檔案 (請參閱第 184 頁的「關於證書檔案」) 中以加密方式驗證的用戶端憑證獲得的使用者資料寫入該身份。使用 `certutil` 將使用者增加至這些檔案中。

透過 certificate 範圍，J2EE 容器可以基於每個使用者憑證中的辨別名稱 (DN) 來執行授權處理。DN 是證書對其公開金鑰進行識別的實體的名稱。此名稱使用 X.500 標準，因此它在網際網路中應該是一專屬名稱。如需有關金鑰庫和信任庫的更多資訊，請參閱第 189 頁的「使用網路安全服務 (NSS) 工具」中的 `certutil` 文件。

下表列出了 certificate 範圍的選擇性特性。

表 9-5 certificate 範圍的選擇性特性

特性	說明
assign-groups	以逗號分隔的群組名稱清單。提供有效證書的所有用戶端均被指定給這些群組。例如， <code>employee,manager</code> ，它們是使用者群組的名稱。
jaas-context	要用於此範圍的登入模組類型。對於 <code>certificate</code> 範圍，該值必須為 <code>certificateRealm</code> 。

編輯 file 和 admin-realm 範圍

伺服器在 `file` 範圍的名為 `keyfile` 的檔案和 `admin-realm` 範圍的名為 `admin-keyfile` 的檔案中維護所有使用者、群組和密碼資訊。對於這兩個範圍，`file` 特性均指定 `keyfile` 的位置。下表列出了 `file` 範圍的必要特性。

表 9-6 file 範圍的必要特性

特性名稱	說明	預設值
<code>file</code>	<code>keyfile</code> 的完整路徑和名稱。	<code>domain-dir/config/keyfile</code>
<code>jaas-context</code>	要用於此範圍的登入模組類型。	<code>fileRealm</code> 是唯一的有効值

`keyfile` 最初為空，因此在使用 `file` 範圍之前，必須先增加使用者。如需說明，請參閱第 167 頁的「管理 `file` 範圍使用者」。

`admin-keyfile` 最初包含管理使用者名稱、加密格式的管理密碼和該使用者所屬的群組（預設為 `asadmin`）。如需有關將使用者增加至 `admin-realm` 的更多資訊，請參閱第 158 頁的「授予對管理工具的存取權」。

備註 – `admin-realm` 的 `asadmin` 群組中的使用者已被授權，可以使用管理主控台和 `asadmin` 工具。只能將具有伺服器管理權限的使用者增加至該群組中。

使用網路安全服務 (NSS) 管理使用者

僅在 **Enterprise Edition** 中，您才可以使用管理主控台來管理使用者（如第 167 頁的「管理 `file` 範圍使用者」中所述），或者可以使用 NSS 工具來管理使用者。網路安全服務 (NSS) 是一組程式庫，用於支援啓用安全性的用戶端與伺服器應用程式的跨平台開發。使用 NSS 建立的應用程式可以支援 SSL v2 和 v3、TLS、PKCS #5、PKCS #7、PKCS #11、PKCS #12、S/MIME、X.509 v3 憑證以及其他安全性標準。如需詳細資訊，請連結到以下 URL：

- 位於 <http://www.mozilla.org/projects/security/pki/nss/> 的 Network Security Services (NSS)
- 位於 <http://www.mozilla.org/projects/security/pki/nss/tools/> 的 NSS Security Tools

- 位於 <http://www.mozilla.org/projects/security/pki/nss/overview.html> 的 Overview of NSS

管理 file 範圍使用者

使用管理主控台管理 file 範圍使用者。file 範圍中的使用者和群組列在 keyfile 中，keyfile 的位置由 file 特性指定。

備註 – 還可以使用以下步驟將使用者增加至任何 file 範圍 (包括 admin-realm) 中。您只需在目標範圍名稱中取代本小節中參照的 file 範圍即可完成此作業。

file 範圍中的使用者可以屬於 *J2EE 群組* (依一般特性分類的使用者種類)。例如，電子商務應用程式的用戶可以屬於 CUSTOMER 群組，但是大用戶應屬於 PREFERRED 群組。將使用者進行群組分類可以簡化有大量使用者時的存取控制。

初次安裝 Application Server 之後，唯一的使用者是管理員在安裝期間輸入的使用者。依預設，此使用者屬於 admin-realm 範圍中的 asadmin 群組，該群組可授予修改 Application Server 的權利。指定給此群組的任何使用者都將具有管理員權限，亦即，這些使用者將具有對 asadmin 工具和管理主控台的存取權。

若要管理 file 範圍使用者，請執行以下作業：

- 第 167 頁的「存取 [file 使用者] 頁面」
- 第 168 頁的「增加使用者」
- 第 168 頁的「編輯使用者資訊」
- 第 169 頁的「刪除使用者」

▼ 存取 [file 使用者] 頁面

- 1 在管理主控台樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定實例，請展開該實例的配置節點。例如，對於預設實例 server，請展開 [server-config] 節點。
 - 若要為所有實例配置預設設定，請展開 [default-config] 節點。
- 3 展開 [安全性] 節點。
- 4 展開 [範圍] 節點。
- 5 選取 [file] 節點。

- 6 在 [編輯範圍] 頁面中，按一下 [管理使用者] 按鈕。
螢幕上將顯示 [file 使用者] 頁面。在此頁面上，執行以下作業：
 - 第 168 頁的「增加使用者」
 - 第 168 頁的「編輯使用者資訊」
 - 第 169 頁的「刪除使用者」

▼ 增加使用者

- 1 按一下 [新建] 以將新使用者增加至 file 範圍。
- 2 在 [file 使用者] 頁面中輸入以下資訊：
 - 使用者 ID (必填) - 使用者的名稱。
 - 密碼 (必填) - 使用者密碼。
 - 確認密碼 (必填) - 再次輸入使用者密碼以進行驗證。
 - 群組清單 (可選擇) - 使用者所屬群組的清單 (以逗號分隔)。無需在其他位置定義這些群組。
- 3 按一下 [確定] 以將該使用者增加至 file 範圍中的使用者清單。按一下 [取消] 以退出而不儲存變更。

更多資訊 等效的 asadmin 指令

```
create-file-user
```

▼ 編輯使用者資訊

- 1 在 [使用者 ID] 欄中，按一下要修改的使用者名稱。
螢幕上將顯示 [編輯 file 範圍使用者] 頁面。
- 2 在 [密碼] 和 [確認密碼] 欄位中輸入新密碼，以變更使用者密碼。
- 3 在 [群組清單] 欄位中增加或刪除群組，以變更使用者所屬的群組。
用逗號將群組名稱分隔開。不需要先定義群組。
- 4 按一下 [儲存] 以將該使用者儲存至 file 範圍中的使用者清單中。
按一下 [關閉] 以退出而不儲存變更。

▼ 刪除使用者

- 1 選取要刪除的使用者名稱左側的核取方塊。
- 2 按一下 [刪除]。
- 3 按一下 [關閉] 以返回至 [編輯範圍] 頁面。

更多資訊 等效的 asadmin 指令

```
delete-file-user
```

配置相互認證

- 第 169 頁的「在應用程式中啟用相互 SSL 認證」
- 第 169 頁的「針對所有應用程式啟用相互認證」

在相互認證中，伺服器端和用戶端認證都會啟用。若要測試相互認證，必須存在一個包含有效證書的用戶端。如需有關相互認證的資訊，請參閱位於以下位置的「*The J2EE 1.4 Tutorial*」中的「Security」一

章：<http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html>。

在應用程式中啟用相互 SSL 認證

若要針對特定應用程式啟用相互認證，請使用 `deploytool` 將認證方法設定為 `Client-Certificate`。如需有關使用 `deploytool` 的更多資訊，請參閱位於以下位置的「*The J2EE 1.4 Tutorial*」中的「Security」一

章：<http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html>。

▼ 針對所有應用程式啟用相互認證

Application Server 使用 `certificate` 範圍進行 HTTPS 認證。

- 1 在管理主控台樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定實例，請展開該實例的配置節點。例如，對於預設實例 `server`，請展開 [server-config] 節點。
 - 若要為所有實例配置預設設定，請展開 [default-config] 節點。
- 3 展開 [安全性] 節點。
- 4 展開 [範圍] 節點。

- 5 選取 [certificate] 範圍。
- 6 按一下 [增加特性] 按鈕。
 - a. 在 [名稱] 欄位中，輸入 `clientAuth`。
 - b. 在 [值] 欄位中，輸入 `true`。
- 7 按一下 [儲存]。
- 8 如果主控台中顯示 [需要重新啟動]，請重新啟動 **Application Server**。
重新啟動伺服器之後，需要對使用 `certificate` 範圍的所有應用程式進行用戶端認證。

有關 JACC 提供者的管理主控台作業

- [第 170 頁的「建立 JACC 提供者」](#)
- [第 171 頁的「編輯 JACC 提供者」](#)
- [第 172 頁的「刪除 JACC 提供者」](#)
- [第 172 頁的「設定使用中的 JACC 提供者」](#)

▼ 建立 JACC 提供者

JACC (Java 容器授權合約) 屬於 J2EE 1.4 規格，它定義可插接式授權提供者介面。這可以讓管理員設定協力廠商外掛程式模組以執行授權。依預設，Application Server 提供簡單的、JACC 相容的、基於檔案的授權引擎。

- 1 在管理主控台 樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定實例，請展開該實例的配置節點。例如，對於預設實例 `server`，請展開 [server-config] 節點。
 - 若要為所有實例配置預設設定，請展開 [default-config] 節點。
- 3 展開 [安全性] 節點。
- 4 選取 [JACC 提供者] 節點。
- 5 在 [JACC 提供者] 頁面上，按一下 [新建]。
- 6 在 [建立 JACC 提供者] 頁面中，輸入以下資訊：

- **名稱** – 用於識別此提供者的名稱。
 - **策略配置** – 用於實作策略配置工廠的類別名稱。預設提供者使用 `com.sun.enterprise.security.provider.PolicyConfigurationFactoryImpl`。
 - **策略提供者** – 用於實作策略工廠的類別名稱。預設提供者使用 `com.sun.enterprise.security.provider.PolicyWrapper`。
- 7 按一下 [增加特性] 按鈕向提供者增加特性。有效特性包括：
 - `repository` – 包含策略檔案的目錄。對於預設提供者，此值為 `${com.sun.aas.instanceRoot}/generated/policy`。
 - 8 按一下 [確定] 儲存此配置，或按一下 [取消] 結束而不進行儲存。

▼ 編輯 JACC 提供者

- 1 在管理主控台樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定實例，請展開該實例的配置節點。例如，對於預設實例 `server`，請展開 [server-config] 節點。
 - 若要為所有實例配置預設設定，請展開 [default-config] 節點。
- 3 展開 [安全性] 節點。
- 4 展開 [JACC 提供者] 節點。
- 5 選取要編輯的 JACC 提供者的節點。
- 6 在 [編輯 JACC 提供者] 頁面上，依需要修改提供者資訊：
 - **策略配置** – 用於實作策略配置工廠的類別名稱。
 - **策略提供者** – 用於實作策略工廠的類別名稱。
- 7 若要增加特性，請按一下 [增加] 按鈕。輸入特性的名稱和值。有效項目包括：
 - `repository` – 包含策略檔案的目錄。對於預設提供者，此值為 `${com.sun.aas.instanceRoot}/generated/policy`。
- 8 若要刪除現有特性，請按一下該特性左側的核取方塊，然後按一下 [刪除特性]。
- 9 按一下 [儲存] 以儲存變更，或按一下瀏覽器上的 [向後] 按鈕以取消而不儲存變更。

▼ 刪除 JACC 提供者

- 1 在管理主控台 樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定實例，請展開該實例的配置節點。例如，對於預設實例 server，請展開 [server-config] 節點。
 - 若要為所有實例配置預設設定，請展開 [default-config] 節點。
- 3 展開 [安全性] 節點。
- 4 選取 [JACC 提供者] 節點。
- 5 按一下要刪除的 JACC 提供者左側的核取方塊。
- 6 按一下 [刪除]。

▼ 設定使用中的 JACC 提供者

- 1 在管理主控台 樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定實例，請展開該實例的配置節點。例如，對於預設實例 server，請展開 [server-config] 節點。
 - 若要為所有實例配置預設設定，請展開 [default-config] 節點。
- 3 選取 [安全性] 節點。
將顯示 [安全性] 頁面。
- 4 在 [JACC] 欄位中，輸入伺服器要使用的 JACC 提供者的名稱。
如果不知道哪些 JACC 提供者可用，請展開樹狀結構中的 [JACC 提供者] 元件，以檢視所有已配置的 JACC 提供者。
- 5 選取 [儲存] 以儲存變更，或選取 [載入預設值] 以復原為預設值。
- 6 如果主控台中顯示 [需要重新啟動]，請重新啟動 Application Server。

有關稽核模組的管理主控台作業

- 第 173 頁的「建立稽核模組」
- 第 174 頁的「編輯稽核模組」
- 第 174 頁的「刪除稽核模組」
- 第 175 頁的「設定使用中的稽核模組」
- 第 175 頁的「啟用或停用稽核記錄」

▼ 建立稽核模組

Application Server 提供了簡單的預設稽核模組；如需更多資訊，請參閱第 176 頁的「使用 default 稽核模組」。

- 1 在管理主控台樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定實例，請展開該實例的配置節點。例如，對於預設實例 server，請展開 [server-config] 節點。
 - 若要為所有實例配置預設設定，請展開 [default-config] 節點。
- 3 展開 [安全性] 節點。
- 4 選取 [稽核模組] 節點。
- 5 在 [稽核模組] 頁面上，按一下 [新建]。
- 6 在 [建立稽核模組] 頁面上，輸入以下資訊：
 - 名稱 - 用於識別此稽核模組的名稱。
 - 類別名稱 - 用於實作此模組的類別之完全合格的名稱。預設稽核模組的類別名稱為 `com.sun.enterprise.security.Audit`。
- 7 若要將 JVM 特性增加至此模組，請按一下 [增加特性]。指定每個特性的名稱和值。有效特性包括：
 - `auditOn` - 指定是否啟用此實作類別。有效值為 `true` 和 `false`。
- 8 按一下 [確定] 以儲存項目，或按一下 [取消] 以退出而不儲存變更。

▼ 編輯稽核模組

依預設，未開啓稽核模組。如需有關如何啓動稽核模組的更多資訊，請參閱第 175 頁的「[啓用或停用稽核記錄](#)」。

- 1 在管理主控台 樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定實例，請展開該實例的配置節點。例如，對於預設實例 server，請展開 [server-config] 節點。
 - 若要為所有實例配置預設設定，請展開 [default-config] 節點。
- 3 展開 [安全性] 節點。
- 4 展開 [稽核模組] 節點。
- 5 選取要編輯的稽核模組的節點。
- 6 在 [編輯稽核模組] 頁面上，依需要修改類別名稱。
- 7 透過選取 [增加] 按鈕並輸入特性的名稱和值，來輸入模組的任何附加特性。有效特性包括：
 - auditOn - 指定是否使用此稽核模組。有效值為 true 和 false。
- 8 透過選取要修改的名稱或值並在文字欄位中直接輸入變更來修改任何現有特性。
- 9 透過選取特性左側的核取方塊並按一下 [刪除特性] 來刪除特性。
- 10 按一下 [儲存] 以儲存變更，或按一下瀏覽器上的 [向後] 按鈕以取消而不儲存變更。

▼ 刪除稽核模組

- 1 在管理主控台 樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定實例，請展開該實例的配置節點。例如，對於預設實例 server，請展開 [server-config] 節點。
 - 若要為所有實例配置預設設定，請展開 [default-config] 節點。

- 3 展開 [安全性] 節點。
- 4 選取 [稽核模組] 節點。
- 5 按一下要刪除的稽核模組左側的核取方塊。
- 6 按一下 [刪除]。

▼ 啓用或停用稽核記錄

- 1 在管理主控台 樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定實例，請展開該實例的配置節點。例如，對於預設實例 server，請展開 [server-config] 節點。
 - 若要為所有實例配置預設設定，請展開 [default-config] 節點。
- 3 選取 [安全性] 節點。
將顯示 [安全性] 頁面。
- 4 若要啓用記錄，請選取 [稽核記錄] 核取方塊。若要停用記錄，請取消選取該核取方塊。
選取此選項可以載入稽核模組，並確保這些模組在稽核時被 Application Server 的稽核程式庫呼叫。
- 5 若要啓用稽核記錄，請指定預設稽核模組 (如第 175 頁的「設定使用中的稽核模組」中所述)。
- 6 選取 [儲存] 以儲存變更。
- 7 如果主控台中顯示 [需要重新啓動]，請重新啓動 Application Server。

▼ 設定使用中的稽核模組

開始之前 若要指定伺服器使用的稽核模組，請先啓用稽核記錄 (如第 175 頁的「啓用或停用稽核記錄」中所述)。

- 1 在 [稽核模組] 欄位中，輸入伺服器要使用的稽核模組名稱。
預先配置的稽核模組稱為 default。確定已依照第 176 頁的「使用 default 稽核模組」中所述將該稽核模組的 auditOn 設定為 true。
- 2 選取 [儲存] 以儲存變更，或選取 [載入預設值] 以取消變更。
- 3 如果主控台中顯示 [需要重新啟動]，請重新啟動 Application Server。

▼ 使用 default 稽核模組

default 稽核模組將認證與授權請求記錄至伺服器記錄檔中。如需有關變更記錄檔位置的資訊，請參閱第 250 頁的「配置一般記錄設定」。

認證記錄項目包括以下資訊：

- 已嘗試進行認證的使用者的名稱。
- 處理過存取請求的範圍。
- 請求的 Web 模組 URI 或 EJB 元件。
- 請求成功或失敗。

無論是否啓用稽核記錄，Application Server 均會記錄所有被拒絕的認證事件。

授權記錄項目包括以下資訊：

- 經過認證的使用者 (如果有) 的名稱。
- 請求的 Web URI 或 EJB 元件。
- 請求成功或失敗。

- 1 在管理主控台 樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定實例，請展開該實例的配置節點。例如，對於預設實例 server，請展開 [server-config] 節點。
 - 若要為所有實例配置預設設定，請展開 [default-config] 節點。
- 3 展開 [安全性] 節點。
- 4 展開 [稽核模組] 節點。
- 5 按一下 [default] 節點。
- 6 將 auditOn 特性的值設定為 true。

- 7 選取 [儲存] 以儲存變更。
- 8 如果主控台中顯示 [需要重新啟動]，請重新啟動 Application Server。

有關偵聽程式和 JMX 連接器的管理主控台作業

- 第 177 頁的「配置 HTTP 偵聽程式的安全性」
- 第 178 頁的「配置 IIOP 偵聽程式的安全性」
- 第 178 頁的「配置管理服務之 JMX 連接器的安全性」
- 第 179 頁的「設定偵聽程式安全性特性」

▼ 配置 HTTP 偵聽程式的安全性

HTTP 服務中的每個虛擬伺服器都透過一個或多個 *HTTP 偵聽程式* 提供網路連線。使用管理主控台可以建立新的 HTTP 偵聽程式和編輯現有 HTTP 偵聽程式的安全性設定。

- 1 在管理主控台樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定實例，請展開該實例的配置節點。例如，對於預設實例 server，請展開 [server-config] 節點。
 - 若要為所有實例配置預設設定，請展開 [default-config] 節點。
- 3 展開 [HTTP 服務] 節點。
- 4 選取 [HTTP 偵聽程式] 節點。
- 5 選取 HTTP 偵聽程式以編輯現有的偵聽程式，或按一下 [新建] 並依照第 233 頁的「[建立 HTTP 偵聽程式](#)」中的程序來建立新的偵聽程式。
- 6 依照第 179 頁的「[設定偵聽程式安全性特性](#)」中的程序來設定安全性特性。
- 7 按一下 [儲存] 以儲存變更，或按一下瀏覽器上的 [向後] 按鈕以取消而不儲存變更。

更多資訊 [等效的 asadmin 指令](#)

```
create-http-listener
```

▼ 配置 IIOP 偵聽程式的安全性

Application Server 支援 CORBA (共用物件請求代理程式架構) 物件，這類物件使用網際網路 Orb 交換協定 (IIOP) 在網路上進行通訊。IIOP 偵聽程式接受來自 EJB 的遠端用戶端和其他基於 CORBA 之用戶端的內送連線。使用管理主控台可以建立新的 IIOP 偵聽程式和編輯現有 IIOP 偵聽程式的設定。

- 1 在管理主控台 樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定實例，請展開該實例的配置節點。例如，對於預設實例 server，請展開 [server-config] 節點。
 - 若要為所有實例配置預設設定，請展開 [default-config] 節點。
- 3 展開 [ORB] 節點。
- 4 選取 [IIOP 偵聽程式] 節點。
- 5 選取 IIOP 偵聽程式以對其進行編輯，或按一下 [新建] 並依照第 239 頁的「[建立 IIOP 偵聽程式](#)」中的程序來建立新的偵聽程式。
- 6 依照第 179 頁的「[設定偵聽程式安全性特性](#)」中的程序來設定安全性特性。
- 7 按一下 [儲存] 以儲存變更，或按一下 [載入預設值] 以復原為特性的預設值。
如果已建立新的偵聽程式，則該新的偵聽程式會立即列在 [IIOP 偵聽程式] 頁面上的 [目前的偵聽程式] 表中。

更多資訊 [等效的 asadmin 指令](#)

```
create-iiop-listener
```

▼ 配置管理服務之 JMX 連接器的安全性

- 1 在管理主控台 樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定實例，請展開該實例的配置節點。例如，對於預設實例 server，請展開 [server-config] 節點。
 - 若要為所有實例配置預設設定，請展開 [default-config] 節點。

- 3 展開 [管理服務] 節點。
- 4 選取要修改的管理服務。
- 5 依照第 179 頁的「設定偵聽程式安全性特性」中的程序來設定安全性特性。
- 6 按一下 [儲存] 以儲存變更，或按一下 [載入預設值] 以復原為特性的預設值。

▼ 設定偵聽程式安全性特性

此程序適用於 HTTP 偵聽程式、IIOP 偵聽程式和 JMX 連接器安全性特性。

- 1 在 [編輯 HTTP 偵聽程式]、[編輯 IIOP 偵聽程式] 或 [編輯 JMX 連接器] 頁面中，移至標有 [SSL] 的區段。
- 2 核取 [安全性] 欄位中的 [已啟用] 方塊，以啟用此偵聽程式的安全性。選取此選項後，您必須選取 [SSL3] 或 [TLS] 才能指定啟用何種類型的安全性，並且必須輸入憑證暱稱。
- 3 如果在使用此偵聽程式時，用戶端要向 Application Server 認證其自身，請核取 [用戶端認證] 欄位中的 [已啟用] 方塊。
- 4 如果已核取 [已啟用] 方塊，請在 [憑證暱稱] 欄位中輸入金鑰庫別名。金鑰庫別名是用於識別現有伺服器金鑰對和憑證的單一值。預設金鑰庫的憑證暱稱為 `slas`。
若要尋找憑證暱稱，請使用 `certutil` 公用程式 (如第 189 頁的「使用網路安全服務 (NSS) 工具」中所述)。
- 5 如果已選取 [已啟用] 方塊，請選取 [SSL3] 和/或 [TLS]。依預設，SSL3 和 TLS 均處於啟用狀態。
- 6 依需要啟用個別密碼組。依預設，已啟用所有受支援的密碼組。第 153 頁的「關於加密算法」中論述了加密算法。
- 7 選取 [儲存] 以儲存變更，或選取 [載入預設值] 以取消變更。

有關虛擬伺服器的管理主控台 安全性作業

- 第 180 頁的「配置單次登入 (SSO)」

▼ 配置單次登入 (SSO)

單次登入可以讓多個應用程式共用使用者登入資訊，而不需要每個應用程式都單獨進行使用者登入。使用單次登入的應用程式對使用者進行一次認證，然後此認證資訊將被傳遞到其他所有涉及的應用程式。

單次登入適用於為同一範圍和虛擬伺服器配置的 Web 應用程式。

備註 - 單次登入使用 HTTP cookie 來傳輸記號，而該記號將每個請求與已儲存的使用者身份關聯起來，因此僅當瀏覽器用戶端支援 cookie 時才能使用單次登入。

單次登入根據以下規則作業：

- 使用者存取 Web 應用程式中受保護的資源時，伺服器會要求使用者使用為該 Web 應用程式定義的方法對其自身進行認證。
- 認證之後，Application Server 將使用與該使用者關聯的角色在虛擬伺服器中的所有 Web 應用程式之間進行授權決策，而不要求使用者分別向每個應用程式進行認證。
- 使用者登出一個 Web 應用程式 (明確登出，或由於階段作業過期而登出) 後，該使用者在所有 Web 應用程式中的階段作業都將變為無效。此後，使用者需要先登入才能存取任一應用程式中的受保護資源。

1 在管理主控台 樹形元件中，展開 [配置] 節點。

2 選取要配置的實例：

- 若要配置特定實例，請展開該實例的配置節點。例如，對於預設實例 server，請展開 [server-config] 節點。
- 若要為所有實例配置預設設定，請展開 [default-config] 節點。

3 展開 [HTTP 服務] 節點。

4 展開 [虛擬伺服器] 節點，然後選取要配置為支援單次登入的虛擬伺服器。

5 按一下 [增加特性]。

一個空白的特性項目將被增加至清單末尾。

6 在 [名稱] 欄位中輸入 sso-enable。

7 在 [值] 欄位中輸入 false 以停用 SSO，輸入 true 以啟用 SSO。

依預設，SSO 處於啟用狀態。

- 8 按一下 [增加特性] 並配置所有適用的 SSO 特性，以增加或變更任何其他單次登入特性。下表說明了虛擬伺服器的有效 SSO 特性。

特性名稱	說明	值
ssso-max-inactive-seconds	如果未接收到任何用戶端作業，在可以清除使用者的單次登入記錄之前等待的時間 (以秒為單位)。對虛擬伺服器中的任何應用程式的存取都可以使單次登入記錄保持使用中狀態。	預設為 300 秒 (5 分鐘)。值越高，為使用者提供的持續性越長，但伺服器上的記憶體消耗也會越多。
ssso-reap-interval-seconds	清除過期的單次登入記錄的間隔 (以秒為單位)。	預設為 60。

- 9 按一下 [儲存]。
- 10 如果主控台中顯示 [需要重新啟動]，請重新啟動 Application Server。

有關連接器連線池的管理主控台作業

- [第 181 頁的「關於連接器連線池」](#)
- [第 181 頁的「關於安全性對映」](#)

關於連接器連線池

連接器模組 (也稱為資源配接卡) 可讓 J2EE 應用程式與企業資訊系統 (EIS) 進行互動。連接器資源可讓應用程式與 EIS 連線。連接器連線池是針對特定 EIS 的一組可重複使用的連線。

安全性對映可以建立 J2EE 使用者和群組與 EIS 使用者和群組之間的對映。使用管理主控台可以建立、更新、列出和刪除連接器連線池的安全性對映。

備註 - 在此環境中，使用者被稱為主體。企業資訊系統 (EIS) 是保存此資訊的任何系統。它可以是主機、訊息傳送系統、資料庫系統或應用程式。

關於安全性對映

使用安全性對映可以將應用程式 (主體或使用者群組) 的呼叫者身份對映至適當的 EIS 主體，此主體包含在容器管理的基於作業事件的分析藍本中。應用程式主體向 EIS 發出請求後，Application Server 將首先使用為連接器連線池定義的安全性對映檢查確切的主

體，以確定已對映的後端 EIS 主體。如果沒有完全相符的主體，Application Server 將使用萬用字元規格 (如果有) 來確定已對映的後端 EIS 主體。應用程式使用者需要執行 EIS 作業 (需要以 EIS 中的特定身份來執行) 時，將使用安全性對映。

若要管理安全性對映，請使用 管理主控台 中的以下程序：

- 第 182 頁的「建立安全性對映」
- 第 183 頁的「編輯安全性對映」
- 第 184 頁的「刪除安全性對映」

▼ 建立安全性對映

連接器連線池的安全性對映可將應用程式使用者和群組 (主體) 對映至 EIS 主體。應用程式使用者需要執行 EIS 作業 (需要 EIS 中的特定身份) 時，請使用安全性對映。

- 1 展開 [資源] 節點
- 2 展開 [連接器] 節點。
- 3 選取 [連接器連線池] 節點。
- 4 透過從目前池清單中選取連接器連線池的名稱來選取連接器連線池，或透過從目前池清單中選取 [新建] 並依照第 80 頁的「建立 JDBC 連線池」中的說明建立新的連接器連線池。
- 5 選取 [安全性對映] 頁面。
- 6 按一下 [新建] 以建立新的安全性對映。
- 7 在 [建立安全性對映] 頁面上，輸入以下特性。
 - 名稱 – 輸入用於參照此特定安全性對映的名稱。
 - 使用者群組 – 要對映至適當 EIS 主體的應用程式之呼叫者身份。請輸入應用程式特定的使用者群組清單 (以逗號分隔)，或輸入萬用字元星號 (*) 來表示所有使用者或所有使用者群組。指定 [主體] 或 [使用者群組] 選項，但不能同時指定這兩個選項。
 - 主體 – 要對映至適當 EIS 主體的應用程式之呼叫者身份。請輸入應用程式特定的主體清單 (以逗號分隔)，或輸入萬用字元星號 (*) 來表示所有主體。指定 [主體] 或 [使用者群組] 選項，但不能同時指定這兩個選項。
- 8 在 [後端主體] 區段中，輸入以下特性。
 - 使用者名稱 – 輸入 EIS 使用者名稱。企業資訊系統 (EIS) 是保存此資訊的任何系統。它可以是主機、訊息傳送系統、資料庫系統或應用程式。
 - 密碼 – 輸入 EIS 使用者的密碼。

- 9 按一下 [確定] 以建立安全性對映，或按一下 [取消] 以退出而不儲存變更。

更多資訊 等效的 asadmin 指令

```
create-connector-security-map
```

▼ 編輯安全性對映

- 1 展開 [資源] 節點。
- 2 展開 [連接器] 節點。
- 3 選取 [連接器連線池] 節點。
- 4 透過從目前池清單中選取連接器連線池的名稱來選取一個連接器連線池。
- 5 選取 [安全性對映] 頁面。
- 6 在 [安全性對映] 頁面上，從目前安全性對映清單中選取安全性對映。
- 7 在 [編輯安全性對映] 頁面上，依需要修改以下特性。
 - 使用者群組 – 要對映至適當 EIS 主體的應用程式之呼叫者身份。請輸入應用程式特定的使用者群組清單 (以逗號分隔)，或輸入萬用字元星號 (*) 來表示所有使用者或所有使用者群組。指定 [主體] 或 [使用者群組] 選項，但不能同時指定這兩個選項。
 - 主體 – 要對映至適當 EIS 主體的應用程式之呼叫者身份。請輸入應用程式特定的主體清單 (以逗號分隔)，或輸入萬用字元星號 (*) 來表示所有主體。指定 [主體] 或 [使用者群組] 選項，但不能同時指定這兩個選項。
- 8 在 [後端主體] 區段中，輸入以下特性。
 - 使用者名稱 – 輸入 EIS 使用者名稱。企業資訊系統 (EIS) 是保存此資訊的任何系統。它可以是主機、訊息傳送系統、資料庫系統或應用程式。
 - 密碼 – 輸入 EIS 使用者的密碼。
- 9 按一下 [儲存] 以儲存對安全性對映所做的變更。

更多資訊 有用的 asadmin 指令

```
list-connector-security-maps 和 update-connector-security-maps
```

▼ 刪除安全性對映

- 1 展開 [資源] 節點。
- 2 展開 [連接器] 節點。
- 3 選取 [連接器連線池] 節點。
- 4 透過從目前池清單中選取連接器連線池的名稱來選取一個連接器連線池。
- 5 選取 [安全性對映] 頁面。
- 6 在 [安全性對映] 頁面上，按一下要刪除的安全性對映名稱左側的核取方塊。
- 7 按一下 [刪除]。

更多資訊 等效的 asadmin 指令

```
delete-connector-security-map
```

使用證書和 SSL

- 第 184 頁的「關於證書檔案」
- 第 185 頁的「變更憑證檔案的位置」
- 第 185 頁的「使用 Java 安全套接字延伸 (JSSE) 工具」
- 第 189 頁的「使用網路安全服務 (NSS) 工具」

關於證書檔案

安裝 Application Server 會產生適合內部測試的 NSS (網路安全服務) 格式的數位憑證。依預設，Application Server 將其憑證資訊儲存在 *domain-dir/config* 目錄下的憑證資料庫中：

- **金鑰庫檔案** *key3.db* 包含 Application Server 的憑證 (包括其私密金鑰)。金鑰庫檔案受密碼保護。可使用 `asadmin change-master-password` 指令變更該密碼。如需有關 `certutil` 的更多資訊，請參閱第 189 頁的「使用 `certutil` 公用程式」。

每個金鑰存放區項目都有專屬別名。安裝後，Application Server 金鑰庫會有一個別名為 *s1as* 的項目。

- **信任庫檔案** *cert8.db* 包含 Application Server 的可信任憑證 (包括其他實體的公開金鑰)。對於可信任的憑證，伺服器已確認憑證中的公開金鑰屬於憑證的所有者。受信任的證書通常包括那些憑證授權單位 (CA) 的證書。

在 Platform Edition 的伺服器端，Application Server 使用 JSSE 格式，該格式使用 `keytool` 來管理憑證和金鑰庫。在 Enterprise Edition 的伺服器端，Application Server 使用 NSS 格式，該格式使用 `certutil` 來管理儲存私密金鑰和憑證的 NSS 資料庫。在兩種版本中，用戶端 (應用程式用戶端或獨立用戶端) 均使用 JSSE 格式。

依預設，已將 Application Server 配置為具有金鑰庫和信任庫，它們將與範例應用程式配合使用並可用於開發目的。若要用於生產目的，您可能需要變更憑證別名、將其他憑證增加至信任庫或變更金鑰庫檔案和信任庫檔案的名稱和/或位置。

▼ 變更憑證檔案的位置

供開發使用的金鑰庫檔案和信任庫檔案儲存在 `domain-dir/config` 目錄中。

- 1 在管理主控台樹狀結構中，展開 [配置]。
- 2 展開 [server-config] (管理配置) 節點。
- 3 選取 [JVM 設定] 節點。
- 4 按一下 [JVM 選項] 標籤。
- 5 在 [JVM 選項] 頁面上，在 [值] 欄位中增加或修改以下值，以反映憑證檔案的新位置：
`-Dcom.sun.appserv.nss.db=${com.sun.aas.instanceRoot}/NSS-database-directory`
 其中，`NSS-database-directory` 是 NSS 資料庫的位置。
- 6 按一下 [儲存]。
- 7 如果主控台中顯示 [需要重新啟動]，請重新啟動 Application Server。

使用 Java 安全套接字延伸 (JSSE) 工具

使用 `keytool` 可以設定和使用 JSSE (Java 安全套接字延伸) 數位憑證。在 Platform Edition 的伺服器端，Application Server 使用 JSSE 格式來管理憑證和金鑰庫。在 Platform Edition 和 Enterprise Edition 中，用戶端 (應用程式用戶端或獨立用戶端) 均使用 JSSE 格式。

J2SE SDK 附帶有 `keytool`，可以讓管理員管理公開/私密金鑰對和關聯憑證。還可以讓使用者快取正與其通訊的另一方的公開金鑰 (以證書形式)。

若要執行 `keytool`，必須配置 shell 環境，以使 J2SE /bin 目錄位於路徑中，或者指令行中必須存在該工具的完整路徑。如需有關 `keytool` 的更多資訊，請參閱位於以下位置的 `keytool` 文

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

使用 keytool 公用程式

以下範例說明了與使用 JSSE 工具處理憑證相關的法：

- 使用 RSA 金鑰演算法在類型為 JKS 的金鑰庫中建立自我簽署憑證。RSA 是 RSA Data Security, Inc. 開發的公開金鑰加密技術。RSA 的首字母縮略分別代表該技術的三位發明者：Rivest、Shamir 和 Adelman。

```
keytool -genkey -noprompt -trustcacerts -keyalg RSA -alias ${cert.alias}
-dname ${dn.name} -keypass ${key.pass} -keystore ${keystore.file}
-storepass ${keystore.pass}
```

[第 187 頁的「使用 keytool 公用程式產生憑證」](#)中提供了建立憑證的另一個範例。

- 使用預設金鑰演算法在類型為 JKS 的金鑰庫中建立自我簽署憑證。

```
keytool -genkey -noprompt -trustcacerts -alias ${cert.alias} -dname
${dn.name} -keypass ${key.pass} -keystore ${keystore.file} -storepass
${keystore.pass}
```

[第 188 頁的「使用 keytool 公用程式簽署數位憑證」](#)中提供了簽署憑證的範例。

- 顯示類型為 JKS 的金鑰庫中的可用憑證。

```
keytool -list -v -keystore ${keystore.file} -storepass ${keystore.pass}
```

- 顯示類型為 JKS 的金鑰庫中的憑證資訊。

```
keytool -list -v -alias ${cert.alias} -keystore ${keystore.file}
-storepass ${keystore.pass}
```

- 將 RFC/文字格式的憑證匯入 JKS 庫中。憑證通常會以網際網路 RFC (註釋請求) 1421 標準定義的可列印編碼格式 (而非憑證的二進位編碼) 加以儲存。此憑證格式 (也稱為 *Base 64 編碼*) 便於透過電子郵件或某些其他機制將憑證匯出至其他應用程式。

```
keytool -import -noprompt -trustcacerts -alias ${cert.alias} -file
${cert.file} -keystore ${keystore.file} -storepass ${keystore.pass}
```

- 從類型為 JKS 的金鑰庫中以 PKCS7 格式匯出憑證。公開金鑰加密標準 #7 (加密訊息語法標準) 定義的回覆格式除了包含已核發的憑證外，還包含支援憑證鏈。

```
keytool -export -noprompt -alias ${cert.alias} -file ${cert.file}
-keystore ${keystore.file} -storepass ${keystore.pass}
```

- 從類型為 JKS 的金鑰庫中以 RFC/文字格式匯出憑證。

```
keytool -export -noprompt -rfc -alias ${cert.alias} -file
${cert.file} -keystore ${keystore.file} -storepass ${keystore.pass}
```

- 從類型為 JKS 的金鑰庫中刪除憑證。

```
keytool -delete -noprompt -alias ${cert.alias} -keystore ${keystore.file}
-storepass ${keystore.pass}
```

第 188 頁的「使用 keytool 公用程式刪除憑證」中提供了從金鑰庫中刪除憑證的另一個範例。

▼ 使用 keytool 公用程式產生憑證

使用 keytool 產生、匯入和匯出憑證。依預設，keytool 將在其執行目錄中建立金鑰庫檔案。

1 變更至要執行憑證的目錄。

始終在包含金鑰庫檔案和信任庫檔案的目錄中產生憑證，依預設，該目錄為 `domain-dir/config`。如需有關變更這些檔案位置的資訊，請參閱第 185 頁的「變更憑證檔案的位置」。

2 輸入以下 keytool 指令，以在金鑰庫檔案 `keystore.jks` 中產生憑證：

```
keytool -genkey -alias keyAlias-keyalg RSA
-keypass changeit
-storepass changeit
-keystore keystore.jks
```

使用任何專屬名稱做為 *keyAlias*。如果您已變更金鑰庫或私密金鑰密碼的預設值，請使用新密碼取代以上指令中的 `changeit`。

螢幕上將顯示提示，要求您提供姓名、組織和其他資訊，keytool 將使用這些資訊產生憑證。

3 輸入以下 keytool 指令，以將產生的憑證匯出至檔案 `server.cer` (或 `client.cer` [如果您願意])：

```
keytool -export -alias keyAlias-storepass changeit
-file server.cer
-keystore keystore.jks
```

4 如需由憑證授權機構簽署的憑證，請參閱第 188 頁的「使用 keytool 公用程式簽署數位憑證」。

5 若要建立信任庫檔案 `cacerts.jks`，並將憑證增加至該信任庫，請輸入以下 keytool 指令：

```
keytool -import -v -trustcacerts
-alias keyAlias
-file server.cer
-keystore cacerts.jks
-keypass changeit
```

如果您已變更金鑰庫或私密金鑰密碼的預設值，請使用新密碼取代以上指令中的 `changeit`。

工具將顯示有關證書的資訊並提示您是否要信任該證書。

- 6 鍵入 `yes`，然後按下 **Enter** 鍵。

`keytool` 便會顯示如下資訊：

```
Certificate was added to keystore  
[Saving cacerts.jks]
```

- 7 重新啟動 **Application Server**。

▼ 使用 `keytool` 公用程式簽署數位憑證

建立數位憑證之後，所有者必須簽署該憑證以防偽造。電子商務站點或那些身份認證對其很重要的站點可以從知名的憑證授權機構 (CA) 購買憑證。如果無需考量認證 (例如當私密安全通訊可以滿足全部需求時)，則可節省獲取 CA 憑證所花費的時間和費用並使用自我簽署憑證。

- 1 請依照 CA 網站上的說明產生憑證金鑰對。

- 2 下載產生的證書金鑰對。

將憑證儲存在包含金鑰庫檔案和信任庫檔案的目錄中，依預設，該目錄為 `domain-dir/config`。請參閱第 185 頁的「變更憑證檔案的位置」。

- 3 在 `shell` 中，變更至包含憑證的目錄。

- 4 使用 `keytool` 將憑證匯入本機金鑰庫和本機信任庫 (如有必要)。

```
keytool -import -v -trustcacerts  
-alias keyAlias  
-file server.cer  
-keystore cacerts.jks  
-keypass changeit  
-storepass changeit
```

如果金鑰庫密碼或私密金鑰密碼不是預設密碼，請使用新密碼取代以上指令中的 `changeit`。

- 5 重新啟動 **Application Server**。

使用 `keytool` 公用程式刪除憑證

若要刪除現有憑證，請使用 `keytool -delete` 指令，例如：

```
keytool -delete  
-alias keyAlias  
-keystore keystore-name  
-storepass password
```

如需有關 `-delete` 指令之可能選項的完整清單，請參閱位於以下位置的 `keytool` 文件：<http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html>。

使用網路安全服務 (NSS) 工具

在 Enterprise Edition 的伺服器端，使用網路安全服務 (NSS) 數位憑證管理儲存私密金鑰和憑證的資料庫。對於用戶端 (應用程式用戶端或獨立用戶端)，使用第 185 頁的「使用 Java 安全套接字延伸 (JSSE) 工具」中所述的 JSSE 格式。

使用網路安全服務 (NSS) 管理安全性的工具包括：

- `certutil`，命令行公用程式，用於管理憑證和金鑰資料庫。第 189 頁的「使用 `certutil` 公用程式」中提供了使用 `certutil` 公用程式的一些範例。
- `pk12util`，命令行公用程式，用於以 PKCS12 格式在憑證/金鑰資料庫和檔案之間匯入和匯出金鑰及憑證。第 191 頁的「使用 `pk12util` 公用程式匯入和匯出憑證」中提供了使用 `pk12util` 公用程式的一些範例。
- `modutil`，命令行公用程式，用於管理 `secmod.db` 檔案或硬體記號中的 PKCS #11 模組資訊。第 192 頁的「使用 `modutil` 增加和刪除 PKCS11 模組」中提供了使用 `modutil` 公用程式的一些範例。

這些工具位於 `install-dir/lib/` 目錄中。以下環境變數用於指向 NSS 安全性工具的位置：

- `LD_LIBRARY_PATH=${install-dir}/lib`
- `${os.nss.path}`

在這些範例中，憑證一般名稱 (CN) 是指用戶端或伺服器的名稱。CN 還用於在 SSL 交換過程中比較憑證名稱和產生憑證的主機名稱。如果憑證名稱和主機名稱不符，則在 SSL 交換過程中會產生警告或異常。在某些範例中，為方便起見，使用憑證一般名稱 `CN=localhost`，以便所有使用者均可以使用該憑證，而不必使用其真實主機名稱建立新憑證。

以下小節中的範例說明了與使用 NSS 工具處理憑證相關的用法：

- 第 189 頁的「使用 `certutil` 公用程式」
- 第 191 頁的「使用 `pk12util` 公用程式匯入和匯出憑證」
- 第 192 頁的「使用 `modutil` 增加和刪除 PKCS11 模組」

使用 `certutil` 公用程式

憑證資料庫工具 `certutil` 是 NSS 命令行公用程式，可建立和修改 Netscape Communicator `cert8.db` 和 `key3.db` 資料庫檔案。該公用程式還可以列出、產生、修改或刪除 `cert8.db` 檔案中的憑證，以及建立或變更密碼、產生新的公開和私密金鑰對、顯示金鑰資料庫的內容或刪除 `key3.db` 檔案中的金鑰對。

金鑰和憑證管理程序通常以在金鑰資料庫中建立金鑰開始，然後在憑證資料庫中產生和管理憑證。以下文件論述了如何使用 NSS (包括 `certutil` 公用程式的語法) 管理憑證

和金鑰資料

庫：<http://www.mozilla.org/projects/security/pki/nss/tools/certutil.html>。

以下清單中的每個項目均提供了使用 NSS 和 JSSE 安全性工具建立和/或管理憑證的範例。

- 產生自我簽署的伺服器 and 用戶端憑證。在此範例中，CN 必須為 `hostname.domain.[com|org|net|...]` 格式。

在此範例中，目錄為 `domain-dir/config`。 `serverseed.txt` 和 `clientseed.txt` 檔案可以包含任何隨機文字。此隨機文字將用於產生金鑰對。

```
certutil -S -n $SERVER_CERT_NAME -x -t "u,u,u"
-s "CN=$HOSTNAME.$HOSTDOMAIN, OU=Java Software, O=Sun Microsystems Inc.,
  L=Santa Clara, ST=CA, C=US"
-m 25001 -o $CERT_DB_DIR/Server.crt
-d $CERT_DB_DIR -f passfile <$CERT_UTIL_DIR/serverseed.txt
```

產生用戶端憑證。此憑證也是自我簽署憑證。

```
certutil -S -n $CLIENT_CERT_NAME -x -t "u,u,u"
-s "CN=MyClient, OU=Java Software, O=Sun Microsystems Inc.,
  L=Santa Clara, ST=CA, C=US"
-m 25002 -o $CERT_DB_DIR/Client.crt
-d $CERT_DB_DIR -f passfile <$CERT_UTIL_DIR/clientseed.txt
```

- 驗證在之前的分項符號中產生的憑證。

```
certutil -V -u V -n $SERVER_CERT_NAME -d $CERT_DB_DIR
certutil -V -u C -n $CLIENT_CERT_NAME -d $CERT_DB_DIR
```

- 顯示可用憑證。

```
certutil -L -d $CERT_DB_DIR
```

- 將 RFC/文字格式的憑證匯入 NSS 憑證資料庫中。

```
certutil -A -a -n ${cert.nickname} -t ${cert.trust.options}
-f ${pass.file} -i ${cert.rfc.file}
-d ${admin.domain.dir}/${admin.domain}/config
```

- 以 RFC 格式從 NSS 憑證資料庫中匯出憑證。

```
certutil -L -a -n ${cert.nickname} -f ${pass.file}
-d ${admin.domain.dir}/${admin.domain}/config > cert.rfc
```

- 從 NSS 憑證資料庫中刪除憑證。

```
certutil -D -n ${cert.nickname} -f ${pass.file}
-d ${admin.domain.dir}/${admin.domain}/config
```

- 將 NSS 資料庫中的憑證轉換為 JKS 格式

```
certutil -L -a -n ${cert.nickname}
-d ${admin.domain.dir}/${admin.domain}/config > cert.rfc
keytool -import -noprompt -trustcacerts -keystore ${keystore.file}
-storepass ${keystore.pass} -alias ${cert.alias} -file cert.rfc
```

使用 pk12util 公用程式匯入和匯出憑證

pk12util 指令行公用程式用於以 PKCS12 格式在憑證/金鑰資料庫和檔案之間匯入與匯出金鑰和憑證。PKCS12 為公開金鑰加密標準 (PKCS) #12，個人資訊交換語法標準。如需有關 pk12util 公用程式的說明，請至

<http://www.mozilla.org/projects/security/pki/nss/tools/pk12util.html>。

- 將 PKCS12 格式的憑證匯入 NSS 憑證資料庫中。

```
pk12util -i ${cert.pkcs12.file} -k ${certdb.pass.file}
-w ${cert.pass.file} -d ${admin.domain.dir}/${admin.domain}/config
```

- 將 PKCS12 格式的憑證匯入 NSS 憑證資料庫記號模組中。

```
pk12util -i ${cert.pkcs12.file} -h ${token.name} -k ${certdb.pass.file}
-w ${cert.pass.file} -d ${admin.domain.dir}/${admin.domain}/config
```

- 以 PKCS12 格式從 NSS 憑證資料庫中匯出憑證。

```
pk12util -o -n ${cert.nickname} -k ${pass.file} -w ${cert.pass.file}
-d ${admin.domain.dir}/${admin.domain}/config
```

- 以 PKCS12 格式從 NSS 憑證資料庫記號模組中匯出憑證 (對硬體加速功能配置有用)。

```
pk12util -o -n ${cert.nickname} -h ${token.name} -k ${pass.file}
-w ${cert.pass.file} -d ${admin.domain.dir}/${admin.domain}/config
```

- 將 PKCS12 憑證轉換為 JKS 格式 (需要 Java 來源)：

```
<target name="convert-pkcs12-to-jks" depends="init-common">
  <delete file="${jks.file}" failonerror="false"/>
  <java classname="com.sun.enterprise.security.KeyTool">
    <arg line="-pkcs12"/>
    <arg line="-pkcsFile ${pkcs12.file}"/>
    <arg line="-pkcsKeyStorePass ${pkcs12.pass}"/>
    <arg line="-pkcsKeyPass ${pkcs12.pass}"/>
    <arg line="-jksFile ${jks.file}"/>
    <arg line="-jksKeyStorePass ${jks.pass}"/>
    <classpath>
      <pathelement path="${slas.classpath}"/>
      <pathelement path="${env.JAVA_HOME}/jre/lib/jsse.jar"/>
    </classpath>
  </java>
</target>
```

使用 modutil 增加和刪除 PKCS11 模組

安全性模組資料庫工具 modutil 是指令行公用程式，用於管理 secmod.db 檔案或硬體記號中的 PKCS #11 (加密記號介面標準) 模組資訊。您可以使用此工具來增加和刪除 PKCS #11 模組、變更密碼、設定預設值、列出模組內容、啟用或停用槽、啟用或停用 FIPS-140-1 規範遵循，以及為加密作業指定預設提供者。此工具還可以建立 key3.db、cert7.db 和 secmod.db 安全性資料庫檔案。如需有關此工具的更多資訊，請至 <http://www.mozilla.org/projects/security/pki/nss/tools/modutil.html>。

- 增加新的 PKCS11 模組或記號。

```
modutil -add ${token.module.name} -nocertdb -force -mechanisms RSA:DSA:RC4:DES
-libfile ${SCA.lib.path} -dbdir ${admin.domain.dir}/${admin.domain}/config
```

- 從 NSS 庫中刪除 PKCS11 模組。

```
modutil -delete ${token.module.name} -nocertdb -force -mechanisms RSA:DSA:RC4:DES
-libfile ${SCA.lib.path} -dbdir ${admin.domain.dir}/${admin.domain}/config
```

- 列出 NSS 庫中的可用記號模組。

```
modutil -list -dbdir ${admin.domain.dir}/${admin.domain}/config
```

詳細資訊

- 您可透過以下 URL 檢視 Java 2 Standard Edition 的安全性論述：<http://java.sun.com/j2se/1.5.0/docs/guide/security/index.html>。
- 您可透過以下 URL 檢視「J2EE 1.4 Tutorial」中的「Security」一章：<http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html>。
- 「Administration Guide」中的第 10 章。
- Developer's Guide 中的「Securing Applications」一章。

配置訊息安全性

本章說明如何在 Application Server 中為 Web 服務配置訊息層安全性。本章包含以下主題：

- 第 141 頁的「關於 Application Server 的安全性」
- 第 156 頁的「有關安全性的管理主控台作業」

本章中的某些內容假設您對安全性和 Web 服務概念已有基本的瞭解。若要瞭解有關這些概念的更多資訊，請在開始閱讀本章之前先仔細閱讀第 192 頁的「詳細資訊」中列出的資源。

關於訊息安全性

- 第 141 頁的「安全性簡介」
- 第 194 頁的「瞭解 Application Server 中的訊息安全性」
- 第 197 頁的「確保 Web 服務的安全」
- 第 198 頁的「確保範例應用程式的安全」
- 第 198 頁的「配置 Application Server 以實現訊息安全性」

訊息安全性概況

在**訊息安全性**中，安全性資訊會插入到訊息中，以使其透過網路層傳輸並附帶訊息到達訊息目標。訊息安全性與傳輸層安全性(在「J2EE 1.4 Tutorial」中的「Security」一章中論述)不同，因為訊息安全性可用於將訊息保護從訊息傳輸中分離開，從而使訊息在傳輸後仍保持受保護狀態。

Web 服務安全性：SOAP 訊息安全性 (WS-Security) 為互通 Web 服務安全性的國際標準，是由 Web 服務技術的所有主要提供者 (包括 Sun Microsystems) 在 OASIS 中協作開發的。WS-Security 是一種訊息安全性機制，它使用 XML 加密和 XML 數位簽名來確保

經由 SOAP 傳送的 Web 服務訊息的安全。WS-Security 規格定義了多種安全性記號 (包括 X.509 證書、SAML 指定和使用者名稱/密碼記號) 的使用，以認證和加密 SOAP Web 服務訊息。

您可透過 <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf> 檢視 WS-Security 規格。

瞭解 Application Server 中的訊息安全性

Application Server 在其 Web 服務用戶端和伺服器端容器中提供對 WS-Security 標準的整合式支援。整合此功能後，Web 服務安全性可由代表應用程式之 Application Server 的容器來強制執行，並且可以用於保護任何 Web 服務應用程式，無需變更應用程式的實作。Application Server 透過提供可將 SOAP 層訊息安全性提供者和訊息保護策略連結到容器和容器中部署的應用程式之功能，來達到此效果。

指定訊息安全性職責

在 Application Server 中，第 146 頁的「系統管理員」和第 146 頁的「應用程式部署人員」角色應主要負責配置訊息安全性。儘管在通常情況下，其他兩個角色之一在不涉及到開發者的條件下就可以確保現有應用程式的安全，且無需變更其實作，但在某些情況下，第 146 頁的「應用程式開發人員」可能也會發揮作用。以下小節定義了各種角色的職責：

- 第 146 頁的「系統管理員」
- 第 146 頁的「應用程式部署人員」
- 第 146 頁的「應用程式開發人員」

系統管理員

系統管理員負責：

- 在 Application Server 中配置訊息安全性提供者。
- 管理使用者資料庫。
- 管理金鑰儲存區和信任儲存區檔案。
- 在使用加密並執行 1.5.0 版之前的 Java SDK 時，配置 Java 加密擴展 (JCE) 提供者。
- 安裝範例伺服器。僅在 xms 範例應用程式用於說明如何使用訊息層 Web 服務安全性時，才執行此作業。

系統管理員使用 管理主控台 來管理伺服器安全性設定，並使用指令行工具來管理憑證資料庫。在 Platform Edition 中，憑證和私密金鑰均儲存在金鑰庫中，並透過 `keytool` 進行管理。Standard Edition 和 Enterprise Edition 將憑證和私密金鑰儲存在 NSS 資料庫中，並使用 `certutil` 對其進行管理。本文件主要適用於系統管理員。如需有關訊息安全性作業的簡介，請參閱第 198 頁的「[配置 Application Server 以實現訊息安全性](#)」。

應用程式部署人員

應用程式部署人員負責：

- 如果上游角色 (開發者或組譯者) 尚未指定任何所需的特定於應用程式的訊息保護策略，則 (在應用程式組譯時) 指定這些策略。
- 修改特定於 Sun 的部署描述元，以向 Web 服務端點和服務參照指定特定於應用程式的訊息保護策略資訊 (即 message-security-binding 元素)。

「Developers' Guide」中的「Securing Applications」一章將論述這些安全性作業。如需有關指向該章的連結，請參閱第 192 頁的「詳細資訊」。

應用程式開發人員

應用程式開發者可以啟用訊息安全性，但並不是必須要這樣做。訊息安全性可以由系統管理員設定，以便確保所有 Web 服務的安全；如果連結到應用程式的提供者或保護策略不同於連結到容器的提供者或保護策略，訊息安全性則由應用程式部署人員設定。

應用程式開發者或組譯者負責：

- 確定應用程式是否需要特定於應用程式的訊息保護策略。如果需要，則透過與應用程式部署人員交流，確保在應用程式組譯時指定所需策略。

關於安全性記號和安全性機制

WS-Security 規格為使用安全性記號來認證和加密 SOAP Web 服務訊息提供了可延伸機制。與 Application Server 一起安裝的 SOAP 層訊息安全性提供者可用於採用使用者名稱/密碼和 X509 憑證安全性記號，以認證和加密 SOAP Web 服務訊息。採用其他安全性記號 (包括 SAML 指定) 的其他提供者將與 Application Server 的後續發行版本一起安裝。

關於使用者名稱記號

Application Server 使用 SOAP 訊息中的**使用者名稱記號**來建立訊息**傳送者**的認證身份。包含使用者名稱記號 (在內嵌式密碼中) 的訊息接收者透過確認傳送者是否知道使用者的秘密 (即密碼)，來驗證訊息傳送者是否經過授權成為使用者 (在記號中識別)。

使用使用者名稱記號時，必須在 Application Server 中配置有效的使用者資料庫。如需有關該主題的更多資訊，請參閱第 160 頁的「編輯範圍」。

關於數位簽名

Application Server 使用 XML 數位簽名將認證身份連結到訊息**內容**。用戶端使用數位簽名來建立呼叫者身份，其方法與使用傳輸層安全性時用基本認證或 SSL 用戶端證書認證建立呼叫者身份的方法相似。訊息接收者將驗證數位簽名以認證訊息內容的來源 (可能與訊息傳送者不同)

使用數位簽名時，必須在 Application Server 中配置有效的金鑰庫和信任庫檔案。如需有關該主題的更多資訊，請參閱第 184 頁的「關於證書檔案」。

關於加密

加密的目的是將資料修改為只有目標讀者才能理解的形式。加密程序透過用加密元素替換原始內容來完成。與公開金鑰加密指出的那樣，可以使用加密來建立能夠閱讀訊息的一方或多方身份。

使用加密時，必須先安裝支援加密的 JCE 提供者。如需有關該主題的更多資訊，請參閱第 200 頁的「配置 JCE 提供者」。

關於訊息保護策略

訊息保護策略是針對請求訊息處理和回應訊息處理而定義的，並根據對源和/或收信人認證的需求來進行表示。來源認證策略代表一個需求，即必須在訊息中建立已傳送訊息或已定義訊息內容的實體之身份，以使其可以由訊息收件者進行認證。接收者認證策略代表一個需求，即必須傳送訊息，以使可接收訊息的實體之身份可由訊息傳送者建立。提供者套用特定的訊息安全性機制，以使訊息保護策略在 SOAP Web 服務訊息的上下文中實現。

請求和回應訊息保護策略是在將提供者配置到容器時定義的。特定於應用程式的訊息保護策略 (細緻程度為 Web 服務連接埠或作業) 也可以在應用程式或應用程式用戶端的特定於 Sun 的部署描述元中進行配置。在任何情況下，如果定義了訊息保護策略，則用戶端的請求和回應訊息保護策略必須與伺服器的請求和回應訊息保護策略相符 (即二者等效)。如需有關針對各應用程式定義訊息保護策略的更多資訊，請參閱「Developers' Guide」中的「Securing Applications」一章。

訊息安全性術語字彙表

以下內容描述本文件中所使用的術語。第 198 頁的「配置 Application Server 以實現訊息安全性」中也論述了這些概念。

- 認證層

認證層是必須執行認證處理的訊息層。Application Server 在 SOAP 層強制執行 Web 服務訊息安全性。

- 認證提供者

在 Application Server 的此發行版本中，Application Server 呼叫**認證提供者**來處理 SOAP 訊息層安全性。

- **用戶端提供者**用於建立 (透過簽名或使用者名稱/密碼) 請求訊息的來源身份和/或保護 (透過加密) 請求訊息，從而使這些訊息僅可由其目標接收者檢視。用戶端提供者還可以將其容器建立為接收的回應之授權收信人 (透過成功地解密)，並驗證回應中的密碼或簽名以認證與回應相關聯的源身份。在 Application Server 中配置的用戶端提供者可用來保護由做為其他服務的用戶端的伺服器端元件 (即 Servlet 和 EJB 元件) 所傳送的請求訊息和所接收的回應訊息。

- **伺服器端提供者**用於將其容器建立為所接收請求的授權接收者 (透過成功地解密)，並驗證請求中的密碼或簽名以認證與該請求相關的來源身份。伺服器端提供者還將建立 (透過簽名或使用者名稱/密碼) 回應訊息的源身份和/或保護 (透過加密) 回應訊息，以使這些訊息只能由其目標收信人檢視。**伺服器端提供者**僅可由伺服器端容器呼叫。
- **預設伺服器提供者**
預設伺服器提供者用於為尚未連結特定伺服器提供者的任何應用程式識別要呼叫的伺服器提供者。**預設伺服器提供者**有時被稱為**預設提供者**。
- **預設用戶端提供者**
預設用戶端提供者用於為尚未連結特定用戶端提供者的任何應用程式識別要呼叫的用戶端提供者。
- **請求策略**
請求策略定義與認證提供者執行的請求處理相關的認證策略需求。按照訊息寄件者的順序表示這些策略，從而使內容之後出現的加密請求意味著訊息收件者將在驗證簽名之前先要對訊息進行解密。
- **回應策略**
回應策略定義與認證提供者執行的回應處理相關的認證策略需求。按照訊息寄件者的順序表示這些策略，從而使內容之後出現的加密請求意味著訊息收件者將在驗證簽名之前先要對訊息進行解密。

確保 Web 服務的安全

透過將 SOAP 層訊息安全性提供者和訊息保護策略連結到部署有應用程式的容器或連結到應用程式提供的 Web 服務端點，來確保在 Application Server 中部署的 Web 服務的安全。透過將 SOAP 層訊息安全性提供者和訊息保護策略連結到用戶端容器或連結到由用戶端應用程式宣告的可攜式服務參照，從而在 Application Server 的用戶端容器中配置 SOAP 層訊息安全性功能。

安裝 Application Server 後，將在 Application Server 的用戶端和伺服器端容器中配置 SOAP 層訊息安全性提供者，其中這些提供者可由容器或者由容器中部署的個別應用程式或用戶端進行連結使用。在安裝程序中，這些提供者配置簡單的訊息保護策略，即如果連結到容器或者連結到容器中的應用程式或用戶端，則將導致所有請求和回應訊息中內容的源均由 XML 數位簽名進行認證。

可以採用 Application Server 的管理介面，從而連結現有提供者以供 Application Server 的伺服器端容器使用，修改由提供者強制執行的訊息保護策略，或使用替代的訊息保護策略來建立新的提供者配置。第 156 頁的「有關安全性的管理主控台作業」中定義了這些作業。您可以在應用程式用戶端容器的 SOAP 訊息層安全性配置上執行類似的管理作業 (如第 208 頁的「啟用應用程式用戶端的訊息安全性」中所定義)。

依預設，Application Server 中的訊息層安全性處於停用狀態。若要配置 Application Server 的訊息層安全性，請執行第 198 頁的「配置 Application Server 以實現訊息安全

性」中列出的步驟。如果您要將 Web 服務安全性用於保護在 Application Server 上部署的所有 Web 服務應用程式，請執行第 201 頁的「啟用訊息安全性的提供者」中的步驟。

完成以上步驟(可能需要重新啟動 Application Server)後，Web 服務安全性將套用於在 Application Server 上部署的所有 Web 服務應用程式。

配置特定於應用程式的 Web 服務安全性

透過在應用程式的特定於 Sun 的部署描述元中定義 message-security-binding 元素，來配置特定於應用程式的 Web 服務安全性功能(在應用程式組譯時)。這些 message-security-binding 元素用於將特定提供者或訊息保護策略與 Web 服務端點或服務參照相關聯，並符合要求以使其套用到相應端點或參照的服務的特定連接埠或方法。

如需有關針對各應用程式定義訊息保護策略的更多資訊，請參閱「*Developers' Guide*」中的「Securing Applications」一章。第 192 頁的「詳細資訊」包含指向該章的連結。

確保範例應用程式的安全

Application Server 隨附名為 xms 的範例應用程式。xms 應用程式具有簡單的 Web 服務功能，可由 J2EE EJB 端點和 Java Servlet 端點共同實作。這兩個端點共用同一個服務端點介面。服務端點介面定義了單一作業(sayHello)，此作業可取得字串引數，並傳回由呼叫引數加前置的 Hello 所組成的 String。

xms 範例應用程式用於說明如何使用 Application Server 的 WS-Security 功能來確保現有 Web 服務應用程式的安全。範例隨附的指令說明了如何啟用 Application Server 的 WS-Security 功能，以將其用於確保 xms 應用程式的安全。範例還說明了 WS-Security 功能與應用程式的直接連結(如第 198 頁的「配置特定於應用程式的 Web 服務安全性」中所述)。

xms 範例應用程式安裝在以下目錄

中：`install-dir/samples/webservices/security/ejb/apps/xms/`。

如需有關編譯、封裝和執行 xms 範例應用程式的資訊，請參閱「*Developers' Guide*」中的「Securing Applications」一章。

配置 Application Server 以實現訊息安全性

- 第 198 頁的「請求策略配置和回應策略配置的動作」
- 第 199 頁的「配置其他安全性功能」
- 第 200 頁的「配置 JCE 提供者」

請求策略配置和回應策略配置的動作

下表列出了訊息保護策略配置，以及由該配置的 WS-Security SOAP 訊息安全性提供者所執行的最終訊息安全性作業。

表 10-1 訊息保護策略與 WS-Security SOAP 訊息安全性作業對映

訊息保護策略	最終的 WS-Security SOAP 訊息保護作業
auth-source="sender"	此訊息包含 wsse:Security 標頭，此標頭包含 wsse:UsernameToken (帶有密碼)。
auth-source="content"	對 SOAP 訊息內文的內容進行簽名。此訊息包含 wsse:Security 標頭，此標頭包含以 ds:Signature。
auth-source="sender" auth-recipient="before-content" OR auth-recipient="after-content"	SOAP 訊息內文的內容已加密，並由最終的 xend:EncryptedData 替代。此訊息包含 wsse:Security 標頭，此標頭包含 wsse:UsernameToken (帶有密碼) 和 xenc:EncryptedKey。 xenc:EncryptedKey 包含用於加密 SOAP 訊息內文的金鑰。此金鑰在收信人的公開金鑰中加密。
auth-source="content" auth-recipient="before-content"	SOAP 訊息內文的內容已加密，並由最終的 xend:EncryptedData 替代。xenc:EncryptedData 已簽名。此訊息包含 wsse:Security 標頭，此標頭包含 xenc:EncryptedKey 和 ds:Signature。xenc:EncryptedKey 包含用於加密 SOAP 訊息內文的金鑰。此金鑰在收信人的公開金鑰中加密。
auth-source="content" auth-recipient="after-content"	SOAP 訊息內文的內容已簽名和加密，並由最終的 xend:EncryptedData 替代。此訊息包含 wsse:Security 標頭，此標頭包含 xenc:EncryptedKey 和 ds:Signature。xenc:EncryptedKey 包含用於加密 SOAP 訊息內文的金鑰。此金鑰在收信人的公開金鑰中加密。
auth-recipient="before-content" OR auth-recipient="after-content"	SOAP 訊息內文的內容已加密，並由最終的 xend:EncryptedData 替代。此訊息包含 wsse:Security 標頭，此標頭包含 xenc:EncryptedKey。xenc:EncryptedKey 包含用於加密 SOAP 訊息內文的金鑰。此金鑰在收信人的公開金鑰中加密。
未指定策略。	模組未執行任何安全性作業。

▼ 配置其他安全性功能

Application Server 使用 SOAP 處理層中整合的訊息安全性提供者來實作訊息安全性。訊息安全性提供者取決於 Application Server 的其他安全性功能。

- 1 如果使用的 Java SDK 版本早於 1.5.0，而且使用了加密技術，請配置 JCE 提供者。
第 200 頁的「配置 JCE 提供者」中論述了如何配置 JCE 提供者。
- 2 如果使用了使用者名稱記號，請在必要時配置使用者資料庫。使用使用者名稱/密碼記號時，必須配置適當的範圍，並為該範圍配置適當的使用者資料庫。
第 160 頁的「編輯範圍」中論述了如何配置使用者資料庫。
- 3 管理證書和私密金鑰 (如有必要)。
第 184 頁的「關於證書檔案」中論述了如何管理憑證和私密金鑰。

接下來的步驟 如果已將 Application Server 的功能配置為供訊息安全性提供者使用，則可能會啟用隨 Application Server 一起安裝的提供者 (如第 201 頁的「啟用訊息安全性的提供者」中所述)。

▼ 配置 JCE 提供者

J2SE 1.4.x 中包含的 Java 加密擴展 (JCE) 提供者不支援 RSA 加密。由於由 WS-Security 定義的 XML 加密通常是基於 RSA 加密，因此，若要使用 WS-Security 來加密 SOAP 訊息，您必須下載並安裝支援 RSA 加密的 JCE 提供者。

備註 – RSA 是 RSA Data Security, Inc. 開發的公開金鑰加密技術。RSA 的首字母縮略分別代表該技術的三位發明者：Rivest、Shamir 和 Adelman。

如果您是在 Java SDK 1.5 版中執行 Application Server，則 JCE 提供者已進行正確配置。如果您是在 Java SDK 1.4.x 版中執行 Application Server，請執行以下步驟，以靜態方式將 JCE 提供者增加為 JDK 環境的一部分：

1 下載並安裝 JCE 提供者 JAR (Java 歸檔) 檔案。

透過以下 URL 可以獲得支援 RSA 加密的 JCE 提供者之清單：http://java.sun.com/products/jce/jce14_providers.html。

2 將 JCE 提供者 JAR 檔案複製到 `java-home/jre/lib/ext/`。

3 停止 Application Server。

如果未停止 Application Server 而後來又在該程序中將其重新啟動，則 Application Server 將無法識別 JCE 提供者。

4 在任何一種文字編輯器中編輯 `java-home/jre/lib/security/java.security` 特性檔案。將剛下載的 JCE 提供者增加至此檔案。

`java.security` 檔案包含用於增加該提供者的詳細說明。通常，您需要在具有類似特性的某個位置處增加一行，其格式如下：

```
security.provider.n=provider-class-name
```

在此範例中，*n* 是 Application Server 計算安全性提供者時所使用的喜好設定順序。將剛才增加的 JCE 提供者的 *n* 設定為 2。

例如，如果下載的是「The Legion of the Bouncy Castle JCE」提供者，則應增加以下行。

```
security.provider.2=org.bouncycastle.jce.provider.  
BouncyCastleProvider
```

確定將 Sun 安全性提供者保持在最高的喜好設定順序，其值為 1。

```
security.provider.1=sun.security.provider.Sun
```


將其他安全性提供者的層級調低，從而使每個層級上只有一個安全性提供者。

以下是提供必要 JCE 提供者並將現有提供者保持在正確位置的 `java.security` 檔案範例。

```
security.provider.1=sun.security.provider.Sun
security.provider.2=org.bouncycastle.jce.provider.
    BouncyCastleProvider
security.provider.3=com.sun.net.ssl.internal.ssl.Provider
security.provider.4=com.sun.rsa.jca.Provider
security.provider.5=com.sun.crypto.provider.SunJCE
security.provider.6=sun.security.jgss.SunProvider
```

- 5 儲存並關閉檔案。
- 6 重新啟動 Application Server。

有關訊息安全性的管理主控台作業

為使用訊息安全性而對 Application Server 設定的大多數步驟都可以透過使用管理主控台、`asadmin` 指令行工具或透過手動編輯系統檔案來完成。通常，不建議編輯系統檔案，因為它可能會做出無意識的變更而使 Application Server 無法正常工作，因此，如有可能，建議優先選擇使用管理主控台來配置 Application Server，然後選擇使用 `asadmin` 工具指令。僅在無管理主控台或等效的 `asadmin` 時，才手動編輯系統檔案。

訊息層安全性支援以(可插接式)認證模組的形式整合在 Application Server 及其用戶端容器中。依預設，Application Server 中的訊息層安全性處於停用狀態。以下小節提供用於啟用、建立、編輯和刪除訊息安全性配置和提供者的詳細資訊。

- 第 201 頁的「啟用訊息安全性的提供者」
- 第 203 頁的「配置訊息安全性提供者」
- 第 205 頁的「建立訊息安全性提供者」
- 第 207 頁的「刪除訊息安全性配置」
- 第 208 頁的「刪除訊息安全性提供者」
- 第 208 頁的「啟用應用程式用戶端的訊息安全性」

大多數情況下，在執行以上管理作業後應重新啟動 Application Server。此步驟尤其適用於作業完成後，您要將管理變更的效果套用至 Application Server 上已部署應用程式中的情況。

▼ 啟用訊息安全性的提供者

若要為在 Application Server 中部署的 Web 服務端點啟用訊息安全性，則必須指定要在伺服器端依預設使用的提供者。若要啟用訊息安全性的預設提供者，您還需要啟用由

Application Server 中部署的 Web 服務之用戶端所使用的提供者。第 208 頁的「[啓用應用程式用戶端的訊息安全性](#)」中論述了有關啓用用戶端使用的提供者之資訊。

若要啓用源自已部署端點的 Web 服務呼叫之訊息安全性，您必須指定預設用戶端提供者。如果已為 Application Server 啓用了預設用戶端提供者，則您必須確保在 Application Server 中部署的端點所呼叫的所有服務均配置為與訊息層安全性相容。

- 1 在管理主控台 樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 server，請選取 [server-config] 節點。
 - 若要配置所有實例的預設設定，請選取 [default-config] 節點。
- 3 展開 [安全性] 節點。
- 4 展開 [訊息安全性] 節點。
- 5 選取 [SOAP] 節點。
- 6 選取 [訊息安全性] 標籤。
- 7 在 [編輯訊息安全性配置] 頁面中，為尚未連結特定提供者的所有應用程式，指定一個要用於伺服器端的提供者和一個要用於用戶端的提供者。

透過修改以下選擇性特性來完成此操作：

- **預設提供者** – 用於為尚未連結特定伺服器提供者的任何應用程式識別要呼叫的伺服器提供者。

依預設，沒有為 Application Server 選取任何提供者配置。若要識別伺服器端提供者，請選取 [ServerProvider]。選取空選項意味著不會在伺服器端 (依預設) 呼叫訊息安全性提供者。

在該欄位中，通常應選取 [ServerProvider]。

- **預設用戶端提供者** – 用於為尚未連結特定伺服器提供者的任何應用程式識別要呼叫的用戶端提供者。

依預設，沒有為 Application Server 選取任何提供者配置。若要識別用戶端提供者，請選取 [ClientProvider]。選取空選項意味著不會在用戶端 (依預設) 呼叫訊息安全性提供者。

在該欄位中，通常應選取空值。如果您要啓用預設提供者和訊息保護策略，以套用於源自部署於 Application Server 上的 Web 服務端點之 Web 服務呼叫，則應選取 [ClientProvider]。

- 8 按一下 [儲存]。
- 9 如果您已啓用用戶端或伺服器提供者，並且要修改已啓用的提供者之訊息保護策略，請參閱第 203 頁的「配置訊息安全性提供者」，以取得有關修改在此步驟中所啓用的訊息安全性提供者的資訊。

更多資訊 等效的 asadmin 指令

- 若要指定預設伺服器提供者，請使用：

```
asadmin set --user admin-user --port admin-port
server-config.security-service.message-security-config.SOAP.
default_provider=ServerProvider
```

- 若要指定預設用戶端提供者，請使用：

```
asadmin set --user admin-user --port admin-port
server-config.security-service.message-security-config.SOAP.
default_client_provider=ClientProvider
```

▼ 配置訊息安全性提供者

通常應重新配置提供者以修改其訊息保護策略 (儘管提供者類型、實作類別和特定於提供者的配置特性可能也需要修改)。

- 1 在管理主控台 樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 `server`，請選取 [server-config] 節點。
 - 若要配置所有實例的預設設定，請選取 [default-config] 節點。
- 3 展開 [安全性] 節點。
- 4 展開 [訊息安全性] 節點。
- 5 選取 [SOAP] 節點。
- 6 選取 [提供者] 標籤。
- 7 選取要編輯的訊息安全性提供者。
Application Server 隨附 ClientProvider 和 ServerProvider。

8 在 [編輯提供者配置] 頁面的 [提供者配置] 區段，可以修改以下特性：

- **提供者類型** – 選取 [client]、[server] 或 [client-server]，以確定是將提供者用做用戶端認證提供者、伺服器認證提供者還是兼用做此兩種提供者 (主從式提供者)。
- **類別名稱** – 輸入提供者的 Java 實作類別。用戶端認證提供者必須實作 `com.sun.enterprise.security.jauth.ClientAuthModule` 介面。伺服器端提供者必須實作 `com.sun.enterprise.security.jauth.ServerAuthModule` 介面。一個提供者可以同時實作這兩種介面，但必須實作對應於其提供者類型的介面。

9 在 [建立提供者配置] 頁面的 [請求策略] 區段，根據需要輸入以下選擇性值。

這些特性是選擇性的，但如果未進行指定，則不會對請求訊息套用任何認證。

請求策略 定義與認證提供者執行的請求處理相關的認證策略需求。按照訊息寄件者的順序表示這些策略，從而使內容之後出現的加密請求意味著訊息收件者將在驗證簽名之前先要對訊息進行解密。

- **認證來源** – 選取 [sender]、[content] 或空值 (空白選項)，以定義對訊息層傳送者認證的需求 (例如使用者名稱、密碼)、內容認證 (例如數位簽名) 或不對請求訊息進行任何認證。如果指定為空值，則不需要進行請求的來源認證。
- **認證接收者** – 選取 [beforeContent] 或 [afterContent]，以定義傳送者對請求訊息的接收者之訊息層認證的需求 (例如，透過 XML 加密)。如果不指定該值，則其為預設值 `afterContent`。

如需有關以下訊息保護策略所導致的由 SOAP 訊息安全性提供者執行的動作之說明，請參閱第 198 頁的「請求策略配置和回應策略配置的動作」。

10 在 [建立提供者配置] 頁面的 [回應策略] 區段，根據需要輸入以下選擇性特性。

這些特性是選擇性的，但如果未進行指定，則不會對回應訊息套用任何認證。

回應策略 定義與認證提供者執行的回應處理相關的認證策略需求。按照訊息寄件者的順序表示這些策略，從而使內容之後出現的加密請求意味著訊息收件者將在驗證簽名之前先要對訊息進行解密。

- **認證來源** – 選取 [sender]、[content] 或空值 (空白選項)，以定義要套用到回應訊息的訊息層傳送者認證 (例如使用者名稱、密碼) 或內容認證 (例如數位簽名) 之需求。如果指定為空值，則不需要進行回應的來源認證。
- **認證接收者** – 選取 [beforeContent] 或 [afterContent]，以定義傳送者對回應訊息的接收者之訊息層認證的需求 (例如，透過 XML 加密)。如果不指定該值，則其為預設值 `afterContent`。

如需有關以下訊息保護策略所導致的由 SOAP 訊息安全性提供者執行的動作之說明，請參閱第 198 頁的「請求策略配置和回應策略配置的動作」。

11 透過按一下 [增加特性] 按鈕增加其他特性。

ProductName; 隨附的提供者可支援以下特性。如果使用的是其他提供者，請參閱其他提供者的文件，以獲得有關特性和有效值的更多資訊。

- `server.config` – 包含伺服器配置資訊的 XML 檔案之目錄和檔案名稱。例如，`domain-dir/config/wss-server-config.xml`。

12 按一下[儲存]。

更多資訊 等效的 `asadmin` 指令

若要設定回應策略，請用 `response` 替代以下指令中的 `request` 字詞。

- 若要將請求策略增加到用戶端並設定認證來源，請使用：

```
asadmin set --user admin-user --port admin-port
server-config.security-service.message-security-config.SOAP.
provider-config.ClientProvider.request-policy.auth_source=
sender | content
```

- 若要將請求策略增加到伺服器並設定認證來源，請使用：

```
asadmin set --user admin-user --port admin-port
server-config.security-service.message-security-config.SOAP.
provider-config.ServerProvider.request-policy.auth_source=
sender | content
```

- 若要將請求策略增加到用戶端並設定認證收信人，請使用：

```
asadmin set --user admin-user --port admin-port
server-config.security-service.message-security-config.SOAP.
provider-config.ClientProvider.request-policy.auth_recipient=
before-content | after-content
```

- 若要將請求策略增加到伺服器並設定認證收信人，請使用：

```
asadmin set --user admin-user --port admin-port
server-config.security-service.message-security-config.SOAP.
provider-config.ServerProvider.request-policy.auth_recipient=
before-content | after-content
```

▼ 建立訊息安全性提供者

若要配置現有提供者，請執行第 203 頁的「配置訊息安全性提供者」中的步驟。

- 1 在管理主控台 樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 `server`，請選取 [server-config] 節點。

- 若要配置所有實例的預設設定，請選取 [default-config] 節點。
- 3 展開 [安全性] 節點。
- 4 展開 [訊息安全性] 節點。
- 5 選取 [SOAP] 節點。
- 6 選取 [提供者] 標籤。
- 7 在 [提供者配置] 頁面中，按一下 [新建]。
- 8 在 [建立提供者配置] 頁面的 [提供者配置] 區段，輸入以下資訊：
 - 預設提供者 - 核取此欄位旁邊的方塊，以使尚未連結特定提供者的所有應用程式可以呼叫新的訊息安全性提供者。提供者是成為預設用戶端提供者、預設伺服器提供者還是兩者兼而有之，取決於為 [提供者類型] 選取的值。
 - 提供者類型 - 選取 [client]、[server] 或 [client-server]，以確定是將提供者用做用戶端認證提供者、伺服器認證提供者還是兼用做此兩種提供者 (主從式提供者)。
 - 提供者 ID - 輸入此提供者配置的識別碼。此名稱將顯示在 [目前的提供者配置] 清單中。
 - 類別名稱 - 輸入提供者的 Java 實作類別。用戶端認證提供者必須實作 `com.sun.enterprise.security.jauth.ClientAuthModule` 介面。伺服器端提供者必須實作 `com.sun.enterprise.security.jauth.ServerAuthModule` 介面。一個提供者可以同時實作這兩種介面，但必須實作對應於其提供者類型的介面。
- 9 在 [建立提供者配置] 頁面的 [請求策略] 區段，根據需要輸入以下選擇性值。這些特性是選擇性的，但如果未進行指定，則不會對請求訊息套用任何認證。
 - 認證來源 - 選取 [sender]、[content] 或空值 (空白選項)，以定義對訊息層傳送者認證的需求 (例如使用者名稱、密碼)、內容認證 (例如數位簽名) 或不對請求訊息進行任何認證。如果指定為空值，則不需要進行請求的來源認證。
 - 認證接收者 - 選取 [beforeContent] 或 [afterContent]，以定義傳送者對請求訊息的接收者之訊息層認證的需求 (例如，透過 XML 加密)。如果不指定該值，則其為預設值 `afterContent`。

如需有關以下訊息保護策略所導致的由 SOAP 訊息安全性提供者執行的動作之說明，請參閱第 198 頁的「請求策略配置和回應策略配置的動作」。
- 10 在 [建立提供者配置] 頁面的 [回應策略] 區段，根據需要輸入以下選擇性特性。這些特性是選擇性的，但如果未進行指定，則不會對回應訊息套用任何認證。
 - 認證來源 - 選取 [sender]、[content] 或空值 (空白選項)，以定義要套用到回應訊息的訊息層傳送者認證 (例如使用者名稱、密碼) 或內容認證 (例如數位簽名) 之需求。如果指定為空值，則不需要進行回應的來源認證。

- **認證接收者** – 選取 [beforeContent] 或 [afterContent]，以定義傳送者對回應訊息的接收者之訊息層認證的需求 (例如，透過 XML 加密)。如果不指定該值，則其為預設值 afterContent。

如需有關以下訊息保護策略所導致的由 SOAP 訊息安全性提供者執行的動作之說明，請參閱第 198 頁的「請求策略配置和回應策略配置的動作」。

11 透過按一下 [增加特性] 按鈕增加其他特性。

ProductName; 隨附的提供者可支援以下特性。如果使用的是其他提供者，請參閱其他提供者的文件，以獲得有關特性和有效值的更多資訊。

- `server.config` – 包含伺服器配置資訊的 XML 檔案之目錄和檔案名稱。例如，`domain-dir/config/wss-server-config.xml`。

12 按一下 [確定] 儲存此配置，或按一下 [取消] 結束而不進行儲存。

更多資訊 等效的 asadmin 指令

```
create-message-security-provider
```

▼ 刪除訊息安全性配置

- 1 在管理主控台 樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 server，請選取 [server-config] 節點。
 - 若要配置所有實例的預設設定，請選取 [default-config] 節點。
- 3 展開 [安全性] 節點。
- 4 選取 [訊息安全性] 節點。
- 5 按一下要刪除的 [訊息安全性配置] 左側的核取方塊。
- 6 按一下 [刪除]。

▼ 刪除訊息安全性提供者

- 1 在管理主控台樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 `server`，請選取 `[server-config]` 節點。
 - 若要配置所有實例的預設設定，請選取 `[default-config]` 節點。
- 3 展開 [安全性] 節點。
- 4 展開 [訊息安全性] 節點。
- 5 選取 [SOAP] 節點。
- 6 選取 [提供者] 頁面。
- 7 按一下要刪除的 [提供者配置] 左側的核取方塊。
- 8 按一下 [刪除]。

更多資訊 等效的 `asadmin` 指令

```
delete-message-security-provider
```

▼ 啓用應用程式用戶端的訊息安全性

必須配置用戶端提供者的訊息保護策略，以使其等效於將與其進行互動式操作的伺服器端提供者的訊息保護策略。在安裝 Application Server 時已配置 (但未啓用) 的提供者符合此情況。

若要啓用用戶端應用程式的訊息安全性，請修改特定於 Application Server 的應用程式用戶端容器配置。

- 1 停止所有取決於用戶端容器描述元的用戶端應用程式。
- 2 在文字編輯器中，開啓 Sun 應用程式用戶端容器描述元 (位於 `domain-dir/config/sun-acc.xml`) 中。

- 3 將 `default-client-provider` 元素增加到檔案，以啟用應用程式用戶端的預設用戶端提供者。

還提供了其他代碼，以顯示啟用用戶端應用程式的訊息安全性之代碼所在的位置。其他代碼在安裝中可能略有不同。請勿變更這些代碼。

```
<client-container>
  <target-server name="your-host" address="your-host"
    port="your-port" />
  <log-service file="" level="WARNING" />
  <message-security-config auth-layer="SOAP"
    default-client-provider="ClientProvider">
    <provider-config
      class-name="com.sun.enterprise.security.jauth.ClientAuthModule"
      provider-id="ClientProvider" provider-type="client">
      <request-policy auth-source="sender" />
      <response-policy/>
      <property name="security.config"
        value="C:/Sun/AppServer/lib/appclient/wss-client-config.xml" />
    </provider-config>
  </message-security-config>
</client-container>
```

用戶端容器中配置的訊息安全性提供者還需要具有對私密金鑰和受信任證書的存取權。這將透過為以下系統特性 (位於應用程式用戶端啟動程序檔中) 定義適當的值來完成。

```
-Djavax.net.ssl.keyStore
```

```
-Djavax.net.ssl.trustStore
```

設定應用程式用戶端配置的請求策略和回應策略

請求策略和回應策略定義與認證提供者執行的請求處理和回應處理相關的認證策略請求。按照訊息寄件者的順序表示這些策略，從而使內容之後出現的加密請求意味著訊息收件者將在驗證簽名之前先要對訊息進行解密。

若要獲得訊息安全性，必須同時在伺服器 and 用戶端中啟用請求策略和回應策略。在用戶端和伺服器中配置策略時，請確定應用程式層級訊息連結之請求/回應保護的用戶端策略與伺服器策略相匹配。

若要設定應用程式用戶端配置的請求策略，請依循第 208 頁的「[啟用應用程式用戶端的訊息安全性](#)」中的說明修改特定於 Application Server 的應用程式用戶端容器配置。在應用程式用戶端配置檔案中，增加以下所示的 `request-policy` 和 `response-policy` 元素，以設定請求策略。

提供的其他代碼可用做參照。其他代碼在安裝中可能略有不同。請勿變更這些代碼。

```
<client-container>
  <target-server name="your-host" address="your-host"
    port="your-port" />
  <log-service file="" level="WARNING" />
  <message-security-config auth-layer="SOAP"
    default-client-provider="ClientProvider">
    <provider-config
      class-name="com.sun.enterprise.security.jauth.ClientAuthModule"
      provider-id="ClientProvider" provider-type="client">
      <request-policy auth-source="sender | content"
        auth-recipient="after-content | before-content" />
      <response-policy auth-source="sender | content"
        auth-recipient="after-content | before-content" />
      <property name="security.config"
        value="install-dir/lib/appclient/wss-client-config.xml" />
    </provider-config>
  </message-security-config>
</client-container>
```

auth-source 的有效值包括 sender 和 content。auth-recipient 的有效值包括 before-content 和 after-content。第 198 頁的「請求策略配置和回應策略配置的動作」中提供了用於說明這些值的各種組合結果的表。

如果不想指定請求策略或回應策略，請將該元素保留為空白，例如：

```
<response-policy/>
```

詳細資訊

- 您可透過以下 URL 檢視 Java 2 Standard Edition 的安全性論述：<http://java.sun.com/j2se/1.4.2/docs/guide/security/index.html>。
- 您可透過以下 URL 檢視「J2EE 1.4 Tutorial」中的「Security」一章：<http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html>。
- 「Administration Guide」中的第 9 章。
- 「Developer's Guide」中的「Securing Applications」一章。
- 您可透過以下 URL 檢視「Oasis Web Services Security: SOAP Message Security (WS-Security)」規格：<http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf>。
- 您可透過以下 URL 檢視 OASIS Web Services Security Username Token Profile 1.0：<http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0.pdf>。

- 您可透過以下 URL 檢視 OASIS Web Services Security X.509 Certificate Token Profile 1.0：<http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0.pdf>。
- 您可透過以下 URL 檢視「XML-Signature Syntax and Processing」文件：<http://www.w3.org/TR/xmlsig-core/>。
- 您可透過以下 URL 檢視「XML Encryption Syntax and Processing」文件：<http://www.w3.org/TR/xmlenc-core/>。

作業事件

透過將一個或多個步驟納入不可分的工作單元，作業事件可確保資料的完整性和一致性。本章包括下列小節：

- 第 213 頁的「關於作業事件」
- 第 214 頁的「有關作業事件的管理主控台作業」

關於作業事件

- 第 213 頁的「何為作業事件？」
- 第 214 頁的「J2EE 技術中的作業事件」

何為作業事件？

作業事件是應用程式中一系列嚴密的動作，所有動作必須成功完成，否則每個動作中的所有變更會被撤消。例如，將資金從支票帳戶轉入儲蓄帳戶是一項作業事件，步驟如下：

1. 檢查支票帳戶是否有足夠的資金來支付此轉帳操作。
2. 如果支票帳號中有足夠的資金，則將該筆資金記入此帳號的借方。
3. 將這些資金記入儲蓄帳戶的貸方。
4. 將此次轉帳記錄到支票帳戶記錄中。
5. 將此次轉帳記錄到儲蓄帳戶記錄中。

如果這些步驟的任何一個步驟失敗，則必須撤消在前面的步驟中所做的所有變更，而且支票帳戶和儲蓄帳戶的狀態必須與它們在作業事件開始之前的狀態相同。該事件稱為**回復**。如果所有步驟均成功完成，則該作業事件處於**已確定**狀態。作業事件以確定或轉返狀態結束。

J2EE 技術中的作業事件

J2EE 技術中的作業事件處理包括以下五個參與者：

- 作業事件管理員
- Application Server
- 資源管理員
- 資源介面
- 使用者應用程式。

透過實作不同的 API 和功能，每個實體均有助於提高作業事件處理的可靠性，如下所述：

- 作業事件管理員提供支援作業事件分隔、作業事件資源管理、同步化及作業事件上下文傳遞所需的服務和管理功能。
- Application Server 提供支援應用程式執行階段環境 (包括作業事件狀態管理) 所需的基礎架構。
- 資源管理員 (透過資源介面) 提供應用程式對資源的存取權。資源管理員參與分散式作業事件，其方法為：實作由作業事件管理員使用的作業事件資源介面，以與作業事件關聯、作業事件完成以及回復工作通訊。關聯式資料庫伺服器便是這樣一個資源管理員。
- 資源介面是一個系統層級的軟體程式庫，應用程式伺服器或用戶端可使用該程式庫連線到資源管理員。資源介面通常專用於資源管理員。它可以做為程式庫，在使用它的用戶端位址空間中使用。JDBC 驅動程式便是此類資源介面的一個範例。
- 開發用於應用程式伺服器環境的作業事件使用者應用程式使用 JNDI 來查找作業事件資料源及作業事件管理員 (可選)。應用程式可以使用企業 Bean 的宣告性作業事件屬性設定或明確的程式化作業事件分隔。

有關作業事件的管理主控台作業

Application Server 根據管理主控台 中的設定來處理作業事件。

配置作業事件

本小節說明如何配置以下作業事件設定：

- 第 215 頁的「配置 Application Server 如何從作業事件中回復」
- 第 215 頁的「設定作業事件逾時值」
- 第 216 頁的「設定作業事件記錄的位置」
- 第 217 頁的「設定關鍵點間隔」

如需有關作業事件的其他資訊，請參閱以下小節：

- 第 213 頁的「何為作業事件？」

- 第 214 頁的「J2EE 技術中的作業事件」

▼ 配置 Application Server 如何從作業事件中回復

由於伺服器當機或資源管理員當機，作業事件可能未完成。完成這些中斷的作業事件並將其從故障中恢復至關重要。Application Server 可在伺服器啟動時從這些故障中回復並完成作業事件。

執行恢復作業時，如果無法訪問某些資源，則伺服器重新啟動作業可能被延遲，因為伺服器正在嘗試恢復作業事件。

如果作業事件跨伺服器進行，啟動此作業事件的伺服器會連絡其他伺服器以獲得作業事件的結果。如果無法訪問其他伺服器，則該作業事件將使用 [啓發式決策] 欄位來確定結果。

- 1 在樹形元件中，選取 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 server，請選取 [server-config] 節點。
 - 若要配置所有實例的預設設定，請選取 [default-config] 節點。
- 3 選取 [作業事件服務] 節點。
- 4 若要恢復未完成的作業事件，請在 [重新啟動時] 欄位中選取 [恢復]。
- 5 在 [重試逾時] 欄位中，設定 Application Server 嘗試連線無法存取的伺服器的時間 (以秒為單位)。預設值為 10 分鐘 (600 秒)。
- 6 在 [啓發式決策] 欄位中，為作業事件中無法訪問的伺服器設定策略。
除非有充分的理由將 [啓發式決策] 欄位設定為 [確定]，否則將其保留設定為 [轉返]。確定不確定的作業事件會破壞應用程式的資料完整性。
- 7 按一下 [儲存]。
- 8 重新啟動 Application Server。

▼ 設定作業事件逾時值

依預設，伺服器不會使作業事件逾時。也就是說，伺服器無限期地等待作業事件完成。在為作業事件設定了逾時值後，如果作業事件在配置的時間內未完成，則 Application Server 將回復該作業事件。

- 1 在樹形元件中，選取 [配置] 節點。

- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 server，請選取 [server-config] 節點。
 - 若要配置所有實例的預設設定，請選取 [default-config] 節點。
- 3 選取 [作業事件服務] 節點。
- 4 在 [作業事件逾時] 欄位中，輸入作業事件逾時之前等待的秒數。
作業事件逾時的預設值為 0 秒。這將停用作業事件逾時。
- 5 按一下 [儲存]。
- 6 重新啟動 Application Server。

▼ 設定作業事件記錄的位置

爲了保持被呼叫資源的資料完整性，同時爲了能夠從故障中恢復，作業事件記錄將記錄有關每個作業事件的資訊。作業事件記錄儲存在 [作業事件記錄位置] 欄位所指定目錄的 tx 子目錄中。這些記錄無法進行人爲讀取。

- 1 在樹形元件中，選取 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 server，請選取 [server-config] 節點。
 - 若要配置所有實例的預設設定，請選取 [default-config] 節點。
- 3 選取 [作業事件服務] 節點。
- 4 在 [作業事件記錄位置] 欄位中輸入作業事件記錄的位置。
將建立 tx 子目錄，而且作業事件記錄儲存在該目錄下。
預設值為 `${com.sun.aas.instanceRoot}/logs`。`${com.sun.aas.instanceRoot}` 變數是實例的名稱，該變數在啟動 Application Server 實例時設定。若要查看 `${com.sun.aas.instanceRoot}` 的值，請按一下 [實際值]。
- 5 按一下 [儲存]。
- 6 重新啟動 Application Server。

▼ 設定關鍵點間隔

關鍵點作業可壓縮作業事件記錄檔。關鍵點間隔是對記錄執行關鍵點作業之間的作業事件數目。關鍵點作業可以減小作業事件記錄檔的大小。關鍵點間隔數越大(例如，2048)，作業事件記錄檔也越大，但關鍵點作業較少，效能可能更佳。關鍵點間隔越小(例如，256)，記錄檔也越小，而同時由於關鍵點作業較為頻繁，效能會略微降低。

- 1 在樹形元件中，選取 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 server，請選取 [server-config] 節點。
 - 若要配置所有實例的預設設定，請選取 [default-config] 節點。
- 3 選取 [作業事件服務] 節點。
- 4 在 [關鍵點間隔] 欄位中，輸入關鍵點作業之間的作業事件數目。
預設值為 2048。
- 5 按一下 [儲存]。
- 6 重新啟動 Application Server。

配置 HTTP 服務

本章說明如何為 Application Server 的 HTTP 服務元件配置虛擬伺服器和 HTTP 偵聽程式。

- 第 219 頁的「關於 HTTP 服務」
- 第 222 頁的「有關 HTTP 服務的管理主控台作業」
- 第 229 頁的「有關虛擬伺服器的管理主控台作業」
- 第 233 頁的「有關 HTTP 偵聽程式的管理主控台作業」

關於 HTTP 服務

- 第 219 頁的「什麼是 HTTP 服務？」
- 第 219 頁的「虛擬伺服器」
- 第 220 頁的「HTTP 偵聽程式」

什麼是 HTTP 服務？

HTTP 服務是 Application Server 的元件，提供用於部署 Web 應用程式和使 HTTP 用戶端能夠存取所部署 Web 應用程式的功能。(請參閱第 60 頁的「部署 Web 應用程式」。) 這些功能透過兩種相關物件 (即虛擬伺服器和 HTTP 偵聽程式) 提供。

虛擬伺服器

虛擬伺服器 (有時稱為虛擬主機) 是一種物件，允許同一實體伺服器託管多個網際網路網域名稱。同一個實體伺服器上託管的所有虛擬伺服器共用該實體伺服器的網際網路協定 (IP) 位址。虛擬伺服器將某個伺服器的網域名稱 (例如 `www.aaa.com`) 與執行 Application Server 的特定伺服器相關聯。

備註 – 請勿將網際網路網域與 Application Server 的管理網域混淆。

例如，假設您要在實體伺服器上託管以下網域：

```
www.aaa.com  
www.bbb.com  
www.ccc.com
```

同時假設 `www.aaa.com`、`www.bbb.com` 和 `www.ccc.com` 都分別具有與之關聯的 Web 模組 `web1`、`web2` 和 `web3`。

這意味著以下 URL 將全部由您的實體伺服器處理：

```
http://www.aaa.com:8080/web1  
http://www.bbb.com:8080/web2  
http://www.ccc.com:8080/web3
```

第一個 URL 將對映到虛擬主機 `www.aaa.com`，第二個 URL 將對映到虛擬主機 `www.bbb.com`，第三個 URL 將對映到虛擬主機 `www.ccc.com`。

另一方面，由於未在 `www.bbb.com` 註冊 `web3`，以下 URL 將導致 404 回覆碼：

```
http://www.bbb.com:8080/web3
```

若要使此對映有效，請確保 `www.aaa.com`、`www.bbb.com` 和 `www.ccc.com` 均可解析為實體伺服器的 IP 位址。這些網域名稱需要在您的網路的 DNS 伺服器中註冊。此外，在 UNIX 系統上，應將這些網域增加到 `/etc/hosts` 檔案中 (如果 `/etc/nsswitch.conf` 檔案中的 `hosts` 設定包含 `files`)。

啓動 Application Server 後，將自動啓動以下虛擬伺服器：

- 名為 `server` 的虛擬伺服器，用於託管所有使用者定義的 Web 模組
- 名為 `__asadmin` 的虛擬伺服器，用於託管所有與管理相關的 Web 模組 (特別是管理主控台)。該伺服器是一個受限制的伺服器，您不能將 Web 模組部署到該虛擬伺服器上。

如果是在非生產環境中開發、測試和部署 Web 服務，通常只需要使用 `server` 虛擬伺服器。在生產環境中，其他虛擬伺服器可為使用者和用戶提供託管功能，這樣，儘管只有一個實體伺服器，但每個使用者和用戶好像都有自己的 Web 伺服器。

HTTP 偵聽程式

每個虛擬伺服器都透過一個或多個 HTTP 偵聽程式來提供伺服器與用戶端之間的連線。每個 HTTP 偵聽程式都是具有 IP 位址、連接埠號、伺服器名稱以及預設虛擬伺服器的偵聽通訊端。

HTTP 偵聽程式必須具有唯一的連接埠號和 IP 位址組合。例如，透過將 IP 位址指定為 0.0.0.0，HTTP 偵聽程式可以在機器的給定連接埠上偵聽所有已配置的 IP 位址。或者，HTTP 偵聽程式可以為每個偵聽程式指定唯一的 IP 位址，但使用相同的連接埠。

由於 HTTP 偵聽程式是 IP 位址和連接埠號的組合，因此您可以擁有多個 IP 位址相同但連接埠號不同 (例如 1.1.1.1:8081 和 1.1.1.1:8082) 的 HTTP 偵聽程式，或 IP 位址不同但連接埠號相同 (例如 1.1.1.1:8081 和 1.2.3.4:8081) 的 HTTP 偵聽程式 (如果已將機器配置為可以回應這些位址)。

不過，如果 HTTP 偵聽程式使用 0.0.0.0 IP 位址 (偵聽某個連接埠上的所有 IP 位址)，您便無法建立其他 IP 位址的 HTTP 偵聽程式 (偵聽特定 IP 位址的同一連接埠)。例如，如果 HTTP 偵聽程式使用 0.0.0.0:8080 (連接埠 8080 上的所有 IP 位址)，則其他 HTTP 偵聽程式不能使用 1.2.3.4:8080。

由於執行 Application Server 的系統通常只能存取一個 IP 位址，因此 HTTP 偵聽程式通常使用 0.0.0.0 IP 位址和不同的連接埠號，其中每個連接埠號用於不同目的。如果系統可以存取多個 IP 位址，則每個位址均可用於不同目的。

依預設，Application Server 啟動時，它具有以下 HTTP 偵聽程式：

- 名為 `http-listener-1` 和 `http-listener-2` 的兩個 HTTP 偵聽程式，這兩個偵聽程式與名為 `server` 的虛擬伺服器相關聯。名為 `http-listener-1` 的偵聽程式未啟用安全性；而 `http-listener-2` 已啟用了安全性。
- 名為 `admin-listener` 的 HTTP 偵聽程式，該偵聽程式與名為 `__asadmin` 的虛擬伺服器相關聯。此偵聽程式已啟用安全性。

所有這些偵聽程式均使用 IP 位址 0.0.0.0 和在安裝 Application Server 期間指定為 HTTP 伺服器連接埠號的連接埠號。如果 Application Server 使用預設連接埠號值，則 `http-listener-1` 使用連接埠 8080，`http-listener-2` 使用連接埠 8181，`admin-listener` 使用連接埠 4849。

每個 HTTP 偵聽程式均有一個預設虛擬伺服器。當請求 URL 的主機元件與 HTTP 偵聽程式關聯的任何虛擬伺服器 (在虛擬伺服器的 `http-listeners` 屬性中列示 HTTP 偵聽程式，即可將虛擬伺服器與該 HTTP 偵聽程式關聯起來) 均不相符時，HTTP 偵聽程式會將所有請求 URL 路由至預設虛擬伺服器。

此外，還應在 HTTP 偵聽程式中指定接收器執行緒的數目。接收器執行緒即為等待連線的執行緒。執行緒接受連線並將其放入佇列 (稱為連線佇列) 中，在佇列中將由工作者執行緒接受這些連線。配置足夠數量的接收器執行緒，以便在傳入新請求時始終有一個接收器執行緒可用，但此數目應儘可能小，以免給系統造成太重負擔。連線佇列既包括接收器執行緒剛剛接受的新連線，又包括持續作用連線管理子系統管理的永久性連線。

一組請求處理執行緒將從連線佇列中擷取內送的 HTTP 請求並對其進行處理。這些執行緒將剖析 HTTP 標頭，選取適當的虛擬伺服器並透過請求處理引擎處理請求。當沒有更多要處理的請求，但可以保持持續連線 (透過使用 HTTP/1.1 或傳送 `Connection: keep-alive` 標頭) 時，請求處理執行緒會假定連線處於閒置狀態，並將連線傳送給持續作用連線管理子系統。

持續作用子系統會定期輪詢此類閒置連線，並在連線佇列中對那些使用中的連線進行排隊，以便將來進行處理。請求處理執行緒將再次從連線佇列中擷取連線並處理其請求。持續作用子系統包含多個執行緒，可以管理大約數萬個連線。透過將大量連線分成較小的子集，使用有效的輪詢技術來確定哪些連線已就緒並具有請求，以及哪些連線由於處於閒置狀態的時間較長而被視為已關閉 (超過允許的持續作用逾時的最大值)。

HTTP 偵聽程式的伺服器名稱即為重新導向期間由伺服器傳送給用戶端的 URL 中顯示的主機名稱。此屬性會影響伺服器自動產生的 URL，但不會影響儲存在伺服器中的目錄和檔案的 URL。如果伺服器使用一個別名，則該名稱應為此別名。如果用戶端傳送 Host: 標頭，則在重新導向中該主機名稱將取代 HTTP 偵聽程式的伺服器名稱。

要使用不同於原始請求中指定連接埠號的連接埠號，請指定重新導向連接埠。如果出現以下某一種情況，則會發生**重新導向**：

- 如果用戶端嘗試存取已不存在於指定 URL 處的資源 (即該資源已移至其他位置)，伺服器將傳回一個指定的回應碼，並在回應的位置標頭中包含新的位置，從而將用戶端重新導向至新位置 (而不是傳回 404)。
- 如果用戶端嘗試透過常規 HTTP 連接埠存取受保護的資源 (例如 SSL)，則伺服器會將此請求重新導向至啓用了 SSL 的連接埠上。在此情況下，伺服器將在位置回應標頭中傳回一個新的 URL，其中的原始非安全連接埠已被替代為啓用了 SSL 的連接埠。用戶端隨後會連線到這個新的 URL。

此外，還應指定是否為 HTTP 偵聽程式啓用安全性以及使用哪種類型的安全性 (例如使用哪個 SSL 協定以及哪些密碼)。

若要存取部署在 Application Server 上的 Web 應用程式，請使用 URL `http://localhost:8080/` (或者，如果是安全應用程式，則使用 `https://localhost:8181/`) 和為此 Web 應用程式指定的環境根目錄。若要存取管理主控台，請使用 URL `https://localhost:4849/` 或 `https://localhost:4849/asadmin/` (其預設環境根目錄)。

由於虛擬伺服器必須指定一個現有的 HTTP 偵聽程式，並且不能指定其他虛擬伺服器已使用的 HTTP 偵聽程式，因此在建立新虛擬伺服器之前，應至少建立一個 HTTP 偵聽程式。

有關 HTTP 服務的管理主控台作業

- 第 223 頁的「配置 HTTP 服務」
- 第 224 頁的「配置 HTTP 服務存取記錄」

▼ 配置 HTTP 服務

- 1 在樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 server，請選取 [server-config] 節點。
 - 若要為將來的實例 (實例使用 default-config 的副本) 配置預設設定，請選取 [default-config] 節點。
- 3 選取 [HTTP 服務] 節點。
- 4 在 [HTTP 服務] 頁面中，您可以設定套用至該服務的所有 HTTP 偵聽程式的特性。下表列出了這些特性。

特性名稱	說明	預設值
traceEnabled	如果設定為 true，則啟用 TRACE 作業。將此特性設定為 false 可減少跨站台程序檔攻擊對 Application Server 的影響。	false
monitoringCacheEnabled	如果設定為 true，Application Server 將快取 HTTP 服務統計的本地值，以回應統計查詢。此值可用於提高效能。 如果設定為 false，Application Server 將查詢 HTTP 服務的每個統計值。	true
monitoringCacheRefreshInMillis	指定更新監視快取之前的間隔時間 (以毫秒為單位)。	5000
sslCacheEntries	指定可快取的 SSL 階段作業的數目。其值沒有上限。	10000
sslSessionTimeout	指定 SSL2 階段作業逾時之前的秒數。	100
ssl3SessionTimeout	指定 SSL3 階段作業逾時之前的秒數。	86400
sslClientAuthDataLimit	指定用戶端憑證交換階段緩衝的應用程式資料的最大數量 (以位元組為單位)。	1048576
sslClientAuthTimeout	指定用戶端憑證交換階段逾時之前的秒數。	60

特性名稱	說明	預設值
keepAliveQueryMeanTime	指定所需的持續作用延時 (以毫秒為單位)。	100
keepAliveQueryMaxSleepTime	指定輪詢持續作用連線的進一步請求之後休眠時間的上限 (以毫秒為單位)。	100
stackSize	指定本機執行緒的最大堆疊大小。	依作業系統/機器而定
statsProfilingEnabled	如果設定為 <code>false</code> ，則會停用 HTTP 服務的監視統計記錄 (可以提高效能)。如果將此特性設定為 <code>false</code> ，則啓用 HTTP 服務的監視功能不會有任何作用。	<code>true</code>
chunkedRequestBufferSize	指定取消堆積請求資料的預設緩衝區大小 (以位元組為單位)。	8192
chunkedRequestTimeoutSeconds	指定取消堆積請求資料的預設逾時 (以秒為單位)。	60
dnsCacheEnabled	如果設定為 <code>true</code> ，則允許使用者監視與 DNS 快取相關的統計。此特性僅在選取了 [HTTP 通訊協定] 標籤中的 [DNS 查詢] 方塊之後才會生效。否則，將忽略該特性設定。	<code>false</code>

- 按一下 [存取記錄] 標籤以配置存取記錄自動重建。
按一下其他標籤以配置請求處理、持續作用子系統、連線池、HTTP 通訊協定和 HTTP 檔案快取。
- 按一下 [儲存]。

▼ 配置 HTTP 服務存取記錄

使用此頁面可以為虛擬伺服器啓用和配置存取記錄自動重建。這些記錄位於 `domain-dir/logs/access` 目錄中，其命名如下：`virtual-server-name_access_log%YYYY;%MM;%DD-%hh;h:mm;m:ss;s`。

按一下 [預設值] 可以載入預設值。

- 核取 [檔案自動重建] 方塊以啓用檔案自動重建。
依預設，將啓用檔案自動重建。
- 從 [自動重建策略] 下拉式清單中選擇策略。
唯一可用的策略為 `time`。

- 3 在 [自動重建間隔] 欄位中，鍵入數值以指定存取記錄自動重建之間的分鐘數。
僅當自動重建策略為 time 時，此欄位才有效。預設為 1440 分鐘。
- 4 在 [自動重建後綴] 欄位中，鍵入字串值以指定自動重建後要增加到記錄檔名稱中的後綴。
預設為 %YYYY;%MM;%DD;-%hh;h:mm;m:ss;s。
- 5 在 [格式] 欄位中，輸入字串值以指定存取記錄的格式。
請使用下表中顯示的格式。預設格式為 %client.name% %auth-user-name% %datetime% %request% %status% %response.length%。

資料	記號
用戶端主機名稱	%client.name%
用戶端 DNS	%client.dns%
系統日期	%datetime%
完整的 HTTP 請求行	%request%
狀態	%status%
回應內容長度	%response.length%
參照標頭	%header.referer%
使用者代理程式	%header.user-agent%
HTTP 方法	%http-method%
HTTP URI	%http-uri%
HTTP 查詢字串	%query-str%
HTTP 通訊協定版本	%http-version%
接受標頭	%header.accept%
日期標頭	%header.date%
If-Modified-Since 標頭	%header.if-mod-since%
授權標頭	%header.auth%
RFC 2616 中定義的任何有效 HTTP 標頭值 (any 也是有效的標頭值；此處將其指定為變數)	%header.any%
授權使用者的名稱	%auth-user-name%
Cookie 的值	%cookie.value%

資料	記號
虛擬伺服器 ID	%vs.id%

- 6 按一下 [儲存] 以儲存變更，或者按一下 [載入預設值] 以復原為預設設定。

▼ 配置 HTTP 服務請求處理執行緒

- 1 按一下 [載入預設值] 以載入預設值。
- 2 在 [執行緒計數] 欄位中，鍵入數值以指定請求處理執行緒的最大數目。
預設為 128。
- 3 在 [初始執行緒計數] 欄位中，鍵入伺服器啟動時可用的請求處理執行緒的數目。
預設為 48。
- 4 在 [執行緒遞增量] 欄位中，鍵入請求數目超過初始執行緒計數時要增加的請求處理執行緒的數目。
預設為 10。
- 5 在 [請求逾時] 欄位中，鍵入請求逾時之前的秒數。
預設為 30 秒。
- 6 在 [緩衝區長度] 欄位中，鍵入請求處理執行緒用於讀取請求資料的緩衝區的大小 (以位元組為單位)。
預設為 4096 位元組。
- 7 按一下 [儲存] 以儲存變更，或者按一下 [載入預設值] 以復原為預設設定。

▼ 配置 HTTP 服務持續作用子系統

- 1 按一下 [載入預設值] 以載入預設值。
- 2 在 [執行緒計數] 欄位中，鍵入要使用的持續作用執行緒的數目。
預設為 1。
- 3 在 [最大連線數] 欄位中，鍵入要維護的持續連線的最大數目。
預設為 256。

- 4 在 [逾時] 欄位中，鍵入持續作用連線應保持開啓狀態的最大秒數。
預設為 30 秒。
- 5 按一下 [儲存] 以儲存變更，或者按一下 [載入預設值] 以復原為預設設定。

▼ 配置 HTTP 服務連線池

- 1 按一下 [載入預設值] 以載入預設值。
- 2 在 [最大擱置計數] 欄位中，鍵入 HTTP 偵聽程式允許的最大擱置連線數目。
預設為 4096。
- 3 在 [佇列大小] 欄位中，鍵入連線佇列大小的最大值 (以位元組為單位)。
此值還指定伺服器可以擁有的最大未執行連線數目。預設為 4096。
- 4 在 [接收緩衝區大小] 欄位中，鍵入 HTTP 偵聽程式的接收緩衝區大小。
預設為 4096。
- 5 在 [傳送緩衝區大小] 欄位中，鍵入 HTTP 偵聽程式的傳送緩衝區大小。
預設為 8192。
- 6 按一下 [儲存] 以儲存變更，或者按一下 [載入預設值] 以復原為預設設定。

▼ 為 HTTP 服務配置 HTTP 通訊協定

- 1 按一下 [載入預設值] 以載入預設值。
- 2 在 [版本] 欄位中，鍵入要使用的 HTTP 通訊協定的版本 (HTTP/1.0 或 HTTP/1.1)。
預設為 HTTP/1.1。
- 3 選取 [DNS 查詢] 方塊以啓用用戶端 DNS 項目的查詢。
預設為 false。
- 4 取消選取 [SSL] 方塊可以全域停用伺服器中的安全性。
將此值設定為 true 可以將 SSL 用於任何啓用了安全性的偵聽程式。預設為 true。
- 5 在 [已強制回應類型] 欄位中，鍵入在沒有與副檔名相符的可用 MIME 對映時要使用的回應類型。
預設為 text/html; charset=iso-8859-1。

- 6 在 [預設回應類型] 欄位中，鍵入預設回應類型。
預設為 `text/html; charset=iso-8859-1`。此值是以分號分隔的字串，由內容類型、編碼、語言和字元集組成。
- 7 按一下 [儲存] 以儲存變更，或者按一下 [載入預設值] 以復原為預設設定。

▼ 為 HTTP 服務配置 HTTP 檔案快取

檔案快取可儲存靜態內容，以便伺服器能夠快速處理對此類內容的請求。

- 1 按一下 [載入預設值] 以載入預設值。
- 2 核取 [全域] 方塊以啟用檔案快取。
預設為 `true`。
- 3 核取 [檔案傳輸] 方塊以在 Windows 中啟用 `TransmitFileSystem` 方法。
預設為 `false`。
- 4 在 [最長存在時間] 欄位中，鍵入有效快取項目的最長存在時間 (以秒為單位)。
預設為 30 秒。
- 5 在 [最長存在時間] 欄位中，鍵入有效快取項目的最長存在時間 (以秒為單位)。
預設為 1024。
- 6 在 [Hash 初始大小] 欄位中，鍵入 Hash 儲存區的初始數目。
預設為零。
- 7 在 [中型檔案大小限制] 欄位中，鍵入可快取為記憶體對映檔案的檔案大小的最大值 (以位元組為單位)。
預設為 537,600 位元組。
- 8 在 [中型檔案大小] 欄位中，鍵入快取為記憶體對映檔案的所有檔案大小的總值 (以位元組為單位)。
預設為 10,485,760 位元組。
- 9 在 [小型檔案大小限制] 欄位中，鍵入可讀入記憶體的檔案大小的最大值 (以位元組為單位)。
預設為 2048 位元組。
- 10 在 [小型檔案大小] 欄位中，鍵入讀入記憶體的所有檔案大小的總值 (以位元組為單位)。
預設為 10,48,576 位元組。

- 11 從 [啟用檔案快取] 下拉式清單中選擇 ON 或 OFF，以在檔案大小小於中型檔案大小限制時啟用或停用檔案內容的快取。
預設為 ON。
- 12 按一下 [儲存] 以儲存變更，或者按一下 [載入預設值] 以復原為預設設定。

有關虛擬伺服器的管理主控台作業

- 第 229 頁的「建立虛擬伺服器」
- 第 231 頁的「編輯虛擬伺服器」
- 第 232 頁的「刪除虛擬伺服器」

▼ 建立虛擬伺服器

- 1 在樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 server，請選取 [server-config] 節點。
 - 若要為將來的實例 (實例使用 default-config 的副本) 配置預設設定，請選取 [default-config] 節點。
- 3 展開 [HTTP 服務] 節點。
- 4 選取 [虛擬伺服器] 節點。
- 5 在 [虛擬伺服器] 頁面中，按一下 [新建]。
螢幕將顯示 [建立虛擬伺服器] 頁面。
- 6 在 [ID] 欄位中，鍵入虛擬伺服器的專屬名稱。
此值用於內部識別虛擬伺服器，而不會提供給 HTTP 用戶端。提供給 HTTP 用戶端的主機名稱必須在 [主機] 欄位中指定。
- 7 在 [主機] 欄位中，鍵入主機名稱或執行伺服器的機器名稱。
請使用已在網路的 DNS 伺服器註冊 (對於 UNIX 系統，則是已在 /etc/hosts 檔案中註冊) 的實體主機名稱或虛擬主機名稱。
- 8 在相對的 [狀態] 區域中，選取 [開啓]、[關閉] 或 [停用]。
預設為 [開啓]。

9 將 [HTTP 偵聽程式] 欄位保留為空白。

在建立 HTTP 偵聽程式並將其與此伺服器關聯時，系統將自動填寫此欄位。

使用此欄位時，需要指定現有的 HTTP 偵聽程式。但是，您不能指定由其他虛擬伺服器使用的偵聽程式；否則，伺服器記錄檔中將顯示錯誤。由於偵聽程式在建立時必須與一個現有的虛擬伺服器相關聯，因此所有的現有偵聽程式均已被其他虛擬伺服器使用。

10 在 [預設 Web 模組] 下拉式清單中，選擇已部署的 Web 模組 (如果有)，此模組將回應所有無法對映到已部署至虛擬伺服器的其他 Web 模組的請求。

如果不指定預設 Web 模組，系統將使用具有空環境根目錄的 Web 模組。如果沒有環境根目錄為空的 Web 模組，則會建立和使用系統預設的 Web 模組。

11 在 [記錄檔] 欄位中，鍵入將顯示此虛擬伺服器記錄訊息之檔案的路徑名稱。

若要將記錄訊息傳送到預設伺服器記錄檔 *domain-dir/logs/server.log*，請將此欄位保留為空。

12 在 [其他特性] 區域中，按一下 [增加特性] 以便為虛擬伺服器增加特性。

無論是否指定特性，新伺服器均會將預設特性 *docroot* 和 *accesslog* 設定為預設值。

下表列出了可用的虛擬伺服器特性。

特性名稱	說明
<i>docroot</i>	伺服器文件根目錄的絕對路徑。 預設為 <i>domain-dir/docroot</i> 。
<i>accesslog</i>	伺服器存取記錄的絕對路徑。 預設為 <i>domain-dir/logs/access</i> 。
<i>sso-enabled</i>	如果為 <i>false</i> ，則此虛擬伺服器將停用單次登入，使用者必須在使用虛擬伺服器上的每個應用程式時進行分別認證。 Servlet 和 JSP 頁面支援在 Application Server 上跨應用程式的單次登入。此功能允許需要同一使用者登入資訊的多個應用程式共用登入資訊，使用者不必在使用每個應用程式時進行分別登入。 預設為 <i>true</i> 。

特性名稱	說明
sso-max-inactive-seconds	<p>指定當未接收到任何用戶端活動時，在清除使用者的單次登入記錄前等待的時間 (以秒為單位)。由於單次登入套用到同一個虛擬伺服器上的多個應用程式，因此對其中任何一個應用程式的存取均會使單次登入記錄保持使用中狀態。</p> <p>預設為 300 秒 (5 分鐘)。較高的值為使用者提供了較長的單次登入持續性，但會佔用較多的伺服器記憶體。</p>
sso-reap-interval-seconds	<p>指定清除過期單次登入記錄的時間間隔 (以秒為單位)。</p> <p>預設為 60。</p>
allowLinking	<p>如果為 true，則會為部署到該虛擬伺服器上的所有 Web 應用程式提供符號連結資源。透過使用 sun-web.xml 檔案中的 sun-web-app 特性 allowLinking，個別 Web 應用程式可置換該設定。</p> <pre><sun-web-app> <property name="allowLinking" value="{true false}"/> </sun-web-app></pre> <p>預設為 true。</p>

- 13 按一下 [確定] 以儲存該虛擬伺服器。

更多資訊 等效的 asadmin 指令

```
create-virtual-server
```

▼ 編輯虛擬伺服器

- 1 在樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 server，請選取 [server-config] 節點。
 - 若要為將來的實例 (實例使用 default-config 的副本) 配置預設設定，請選取 [default-config] 節點。
- 3 展開 [HTTP 服務] 節點。
- 4 選取 [虛擬伺服器] 節點。

- 5 選取要編輯的虛擬伺服器。
- 6 在 [編輯虛擬伺服器] 頁面中，您可以執行以下作業：
 - 在 [主機] 欄位中變更主機名稱。
 - 變更 [狀態] 設定的值。
 - 增加或移除 HTTP 偵聽程式。
 - 變更 [預設 Web 模組] 的選取。
 - 變更 [記錄檔] 的值。
 - 增加、移除或修改特性。
- 7 按一下 [儲存] 以儲存變更。

▼ 刪除虛擬伺服器

- 1 在樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 server，請選取 [server-config] 節點。
 - 若要為將來的實例 (實例使用 default-config 的副本) 配置預設設定，請選取 [default-config] 節點。
- 3 展開 [HTTP 服務] 節點。
- 4 選取 [虛擬伺服器] 節點。
- 5 在 [虛擬伺服器] 頁面中，核取要刪除虛擬伺服器的名稱旁邊的方塊。
- 6 按一下 [刪除]。

可以刪除 __asadmin 虛擬伺服器，但是建議不要執行此作業。若要執行此作業，請先將 Application Server 的 domain.xml 檔案中的 virtual-server 元素複製到安全的位置，以便可以在需要時復原這些設定。

更多資訊 **等效的 asadmin 指令**

`delete-virtual-server`

有關 HTTP 偵聽程式的管理主控台作業

- 第 233 頁的「[建立 HTTP 偵聽程式](#)」
- 第 235 頁的「[編輯 HTTP 偵聽程式](#)」
- 第 235 頁的「[選取要編輯的 HTTP 偵聽程式](#)」

▼ 建立 HTTP 偵聽程式

- 1 在樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 `server`，請選取 [server-config] 節點。
 - 若要為將來的實例 (實例使用 `default-config` 的副本) 配置預設設定，請選取 [default-config] 節點。
- 3 展開 [HTTP 服務] 節點。
- 4 選取 [HTTP 偵聽程式] 節點。
- 5 在 [HTTP 偵聽程式] 頁面中，按一下 [新建]。
螢幕將顯示 [建立 HTTP 偵聽程式] 頁面。
- 6 在 [名稱] 欄位中，鍵入偵聽程式的名稱。
- 7 如果您不希望在伺服器重新啟動時啟用偵聽程式，請在 [偵聽程式] 欄位中取消核取 [啟用] 方塊。
- 8 在 [網路位址] 欄位中，如果希望偵聽程式偵聽此伺服器的所有 IP 位址，請鍵入 `0.0.0.0` 並使用唯一的連接埠值。否則，請鍵入此伺服器的有效 IP 位址。
- 9 在 [偵聽程式連接埠] 欄位中，如果 [網路位址] 欄位為 `0.0.0.0`，請鍵入唯一的連接埠值，或者，如果要使用其他 IP 位址，請鍵入所需的連接埠值。
- 10 從 [預設虛擬伺服器] 下拉式清單中選擇一個虛擬伺服器。

- 11 在 [伺服器名稱] 欄位中，鍵入要在伺服器傳送給用戶端的 URL 中使用的主機名稱。如果伺服器使用別名，則該名稱應為此別名。
如果伺服器未使用別名，請將此欄位保留為空。
- 12 在 [進階] 區域中，可以執行以下任何作業：
 - 若要將請求重新導向至其他連接埠，請在 [重新導向連接埠] 欄位中鍵入一個值。如果滿足以下兩個條件，Application Server 將自動重新導向請求：
 - 此偵聽程式支援非 SSL 請求。
 - 接收了符合安全性限制需要 SSL 傳輸的請求。
依預設，Application Server 使用原始請求中指定的連接埠號。
 - 變更接收器執行緒的數目。
 - 取消核取 [Powered By] 方塊以在 Servlet 產生的 HTTP 回應標頭中停用 X-Powered-By: Servlet/2.4 標頭。
Java Servlet 2.4 規格中定義了此標頭，其容器可以增加到 Servlet 產生的回應。類似地，JavaServer Pages™ (JSP™) 2.0 規格中定義了一個 X-Powered-By: JSP/2.0 標頭，此標頭將增加到使用 JSP 技術的回應 (可選擇)。依預設，將為 Web 應用程式啟用 X-Powered-By: JSP/2.0 標頭。這些標頭的目標是幫助網站管理員收集有關使用 Servlet 和 JSP 技術的統計資料。
如需有關啟用和停用 JSP 頁面的 X-Powered-By 標頭的資訊，請參閱「*Application Server Developer's Guide*」中的「Deployment Descriptor Files」一章。請參閱第 54 頁的「[詳細資訊](#)」，以取得指向此文件的連結。
根據生產環境的不同，可以省略 X-Powered-By 標頭的產生，以隱藏基礎技術。
- 13 若要建立不安全的偵聽程式，請按一下 [確定]。
- 14 在此頁面的 [SSL] 區段，您可以將偵聽程式配置為使用 SSL 安全性、TLS 安全性或同時使用 SSL 和 TLS 安全性。
若要設定安全偵聽程式，請執行以下步驟：
- 15 在 [安全性] 欄位中核取 [啟用] 方塊。
- 16 若要強制用戶端在使用此偵聽程式時自行向伺服器進行認證，請在 [用戶端認證] 欄位中核取 [啟用] 方塊。
- 17 在 [憑證暱稱] 欄位中輸入現有伺服器金鑰組和憑證的名稱。請參閱「安全性」一章以取得更多資訊。
- 18 在 [SSL3/TLS] 區段中：
 - a. 核取要在偵聽程式上啟用的安全性協定。核取 [SSL3]、[TLS] 或同時核取此兩者。

- b. 核取協定所使用的密碼組。若要啟用所有密碼組，請核取 [所有受支援的密碼組]。您也可以啟用個別密碼組。
- 19 此時，偵聽程式將在已指定為預設虛擬伺服器的虛擬伺服器的 [HTTP 偵聽程式] 欄位中列示。

更多資訊 等效的 asadmin 指令

`create-http-listener` 和 `create-ssl`

▼ 編輯 HTTP 偵聽程式

- 1 在樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 `server`，請選取 [server-config] 節點。
 - 若要為將來的實例 (實例使用 `default-config` 的副本) 配置預設設定，請選取 [default-config] 節點。
- 3 展開 [HTTP 服務] 節點。
- 4 選取 [HTTP 偵聽程式] 節點。
- 5 選取要編輯的 HTTP 偵聽程式。
- 6 在 [編輯 HTTP 偵聽程式] 頁面中，修改其中的任何設定。
- 7 按一下 [儲存] 以儲存變更。

▼ 選取要編輯的 HTTP 偵聽程式

- 1 在樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 `server`，請選取 [server-config] 節點。

- 若要為將來的實例 (實例使用 default-config 的副本) 配置預設設定，請選取 [default-config] 節點。
- 3 展開 [HTTP 服務] 節點。
 - 4 選取 [HTTP 偵聽程式] 節點。
 - 5 在 [HTTP 偵聽程式] 頁面中，核取要刪除 HTTP 偵聽程式的名稱旁邊的方塊。
 - 6 按一下 [刪除]。

可以刪除 http-listener-1、http-listener-2 和 admin-listener HTTP 偵聽程式，但是建議不要執行此作業。若要執行此作業，請先將 Application Server 的 domain.xml 檔案中的 http-listener 元素複製到安全的位置，以便可以在需要時復原這些設定。

更多資訊 等效的 asadmin 指令

delete-http-listener

配置物件請求代理程式

本章描述如何配置物件請求代理程式 (ORB) 和 IIOP 偵聽程式。它包含以下小節：

- 第 237 頁的「關於物件請求代理程式」
- 第 238 頁的「有關 ORB 的管理主控台作業」
- 第 239 頁的「有關 IIOP 偵聽程式的管理主控台作業」

關於物件請求代理程式

- 第 237 頁的「CORBA」
- 第 237 頁的「什麼是 ORB？」
- 第 238 頁的「IIOP 偵聽程式」

CORBA

Application Server 支援標準的協定集和格式集，可確保互通的功能。這些協定之間的協定是由 CORBA 定義的。

CORBA (共用物件請求代理程式架構) 模型以請求分散式物件服務或伺服器服務的用戶端為基礎，透過明確定義的介面，以遠端方法請求形式發送物件請求。遠端方法請求傳送有關需要執行的作業的資訊，其中包括被呼叫方法的服務供應商的物件名稱 (稱為物件參考) 和參數 (如果有)。CORBA 自動處理物件註冊、物件位置、物件啟動、請求非多工、錯誤處理、排列與作業派送等網路程式設計作業。

什麼是 ORB ?

物件請求代理程式 (ORB) 是 CORBA 的中央元件。ORB 提供所需的基礎架構來識別並尋找物件、處理連線管理、傳送資料並請求通訊。

CORBA 物件之間從不直接進行通訊，該物件是透過遠端存根向在本機機器中執行的 ORB 發出請求。然後，本機 ORB 將請求發送至使用網際網路 Orb 交換協定 (縮寫為 IIOP) 的另一台機器中的 ORB。然後，遠端 ORB 找到適當的物件、處理請求並傳回結果。

使用 RMI-IIOP，應用程式或物件可將 IIOP 用作遠端方法呼叫 (RMI) 協定。企業 Bean (EJB 模組) 的遠端用戶端透過 RMI-IIOP 與 Application Server 進行通訊。

IIOP 偵聽程式

IIOP 偵聽程式是一個偵聽插槽，它接受來自企業 Bean 的遠端用戶端和其他基於 CORBA 的用戶端的進來的連線。可以為 Application Server 配置多個 IIOP 偵聽程式。為每個偵聽程式指定一個連接埠號、一個網路位址和 (選擇性地) 多個安全性屬性。如需更多資訊，請參閱第 239 頁的「建立 IIOP 偵聽程式」。

有關 ORB 的管理主控台作業

- 第 238 頁的「配置 ORB」

▼ 配置 ORB

- 1 在樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 server，請選取 [server-config] 節點。
 - 若要為將來的實例 (實例使用 default-config 的副本) 配置預設設定，請選取 [default-config] 節點。
- 3 選取 [ORB] 節點。
- 4 從 [執行緒池 ID] 下拉式清單中選擇 ORB 使用的執行緒池。

ORB 使用執行緒池回應來自通過 RMI-IIOP 進行通訊的企業 Bean 的遠端用戶端和其他用戶端的請求。如需更多資訊，請參閱第 243 頁的「關於執行緒池」和第 244 頁的「建立執行緒池」。
- 5 在 [最大訊息片段大小] 欄位中，設定 IIOP 訊息的最大片段大小。

大於此大小的訊息將被分段。

- 6 在 [連線總數] 欄位中，設定所有 IIOP 偵聽程式的最大進來的連線數。
- 7 如果需要進行 IIOP 用戶端認證，請選取 [需要] 核取方塊。
- 8 按一下 [儲存] 以儲存變更，或者按一下 [載入預設值] 以載入預設值。
- 9 重新啟動伺服器。

有關 IIOP 偵聽程式的管理主控台作業

- 第 239 頁的「建立 IIOP 偵聽程式」
- 第 240 頁的「編輯 IIOP 偵聽程式」
- 第 241 頁的「刪除 IIOP 偵聽程式」

▼ 建立 IIOP 偵聽程式

- 1 在樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 server，請選取 [server-config] 節點。
 - 若要為將來的實例 (實例使用 default-config 的副本) 配置預設設定，請選取 [default-config] 節點。
- 3 展開 [ORB] 節點。
- 4 選取 [IIOP 偵聽程式]。
- 5 按一下 [新增]。
- 6 在 [名稱] 欄位中輸入用於識別偵聽程式的名稱。
- 7 在 [網路位址] 欄位中輸入偵聽程式的網路位址
此位址可以是 IP 位址，也可以是 DNS 可解析的主機名稱。
- 8 在 [偵聽程式連接埠] 欄位中，輸入偵聽程式要進行偵聽的連接埠號。
- 9 在 [偵聽程式] 欄位中，核取 [啟用] 方塊以啟用偵聽程式。
- 10 在 [附加特性] 區域中，為應用程式所需的特性提供值。

- 11 若要建立偵聽程式，請執行以下步驟：
 - 若要建立不安全的偵聽程式，請按一下 [確定]。
 - 若要設定安全偵聽程式，請執行以下操作：
 - a. 在 [安全性] 欄位中核取 [啓用] 方塊。
 - b. 若要強制用戶端在使用此偵聽程式時自行向伺服器進行驗證，請在 [用戶端驗證] 欄位中核取 [啓用] 方塊。
 - c. 在 [憑證暱稱] 欄位中輸入現有伺服器金鑰組和憑證的名稱。
 - d. 在 [SSL3/TLS] 區段：
 - i. 核取要在偵聽程式上啓用的安全性協定。核取 [SSL3] 或 [TLS]，或同時啓用這兩種協定。
 - ii. 選取協定所使用的密碼組。
若要啓用所有密碼組，請核取 [所有支援的密碼組]。您還可以啓用單一密碼組。
 - e. 按一下 [確定]。
現在，該偵聽程式將列示在 [IIOP 偵聽程式] 頁面的 [目前偵聽程式] 表中。

更多資訊 等效的 asadmin 指令

create-iiop-listener 和 create-ssl

▼ 編輯 IIOP 偵聽程式

- 1 在樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 server，請選取 [server-config] 節點。
 - 若要為將來的實例 (實例使用 default-config 的副本) 配置預設設定，請選取 [default-config] 節點。
- 3 展開 [ORB] 節點。

- 4 選取 [IIOP 偵聽程式] 節點。
- 5 在 [目前偵聽程式] 表中選取要修改的偵聽程式。
- 6 修改該偵聽程式的設定。
請參閱第 239 頁的「建立 IIOP 偵聽程式」，以取得有關可修改欄位的說明。
- 7 如果變更了偵聽程式的連接埠號，請重新啓動伺服器。

▼ 刪除 IIOP 偵聽程式

- 1 在樹形元件中，展開 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 server，請選取 [server-config] 節點。
 - 若要為將來的實例 (實例使用 default-config 的副本) 配置預設設定，請選取 [default-config] 節點。
- 3 展開 [ORB] 節點。
- 4 選取 [IIOP 偵聽程式] 節點。
- 5 在 [目前偵聽程式] 表中核取要刪除的偵聽程式。
- 6 按一下 [刪除]。

更多資訊 **等效的 asadmin 指令**

```
delete-iiop-listener
```


執行緒池

本章說明如何建立、編輯和刪除執行緒池。它包含以下小節：

- 第 243 頁的「關於執行緒池」
- 第 244 頁的「有關執行緒池的管理主控台作業」

關於執行緒池

本小節說明執行緒池及其在 Application Server 中的工作方式。

Java 虛擬機器 (JVM) 可以支援一次執行多個執行緒。爲了提昇效能，Application Server 可維護一個或多個執行緒池。可以將特定的執行緒池指定給連接器模組和 ORB。

一個執行緒池可以爲多個連接器模組和企業 Bean 提供服務。請求執行緒處理使用者對應用程式元件的請求。伺服器收到請求時，它會將該請求指定給執行緒池中的空閒執行緒。執行緒會執行用戶端請求，並傳回結果。例如，如果請求需要使用目前被佔用的系統資源，則此執行緒將等待，直至此資源可用時，才允許請求使用此資源。

指定爲來自應用程式的請求保留的最大執行緒數和最小執行緒數。可以在這兩個值之間，動態調整執行緒池。指定的最小執行緒池大小將通知伺服器爲應用程式請求至少分配該大小的保留執行緒數。該數目可以增加至所指定的最大執行緒池大小。

增加程序可用的執行緒數目，可讓程序同時回應更多的應用程式請求。

透過將 Application Server 執行緒分到不同的執行緒池中，避免在一個資源介面或應用程式佔用 Application Server 中的所有執行緒時，出現執行緒不足的情況。

有關執行緒池的管理主控台作業

- 第 244 頁的「建立執行緒池」
- 第 245 頁的「編輯執行緒池」
- 第 246 頁的「刪除執行緒池」

▼ 建立執行緒池

- 1 在樹形元件中，選取 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 server，請選取 [server-config] 節點。
 - 若要配置所有實例的預設設定，請選取 [default-config] 節點。
- 3 選取 [執行緒池] 節點。
- 4 在 [目前的池] 下按一下 [新建]。
- 5 在 [執行緒池 ID] 欄位中輸入執行緒池的名稱。
- 6 在 [最小執行緒池大小] 欄位中，輸入為此佇列中的請求提供服務的執行緒池中執行緒的最小數目。
創設此執行緒池時將預先建立這些執行緒。
- 7 在 [最大執行緒池大小] 欄位中，輸入為此佇列中的請求提供服務的執行緒池中執行緒的最大數目。
這是存在於此執行緒池中的執行緒數上限。
- 8 在 [閒置逾時] 欄位中輸入數值 (以秒為單位)，超過此時間段之後將從池中移除閒置執行緒。
- 9 在 [工作佇列數] 欄位中，輸入由此執行緒池提供服務的工作佇列的總數。
- 10 按一下 [確定]。
- 11 重新啓動 Application Server。

更多資訊 等效的 asadmin 指令

```
create-threadpool
```

▼ 編輯執行緒池

- 1 在樹形元件中，選取 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 server，請選取 [server-config] 節點。
 - 若要配置所有實例的預設設定，請選取 [default-config] 節點。
- 3 選取 [執行緒池] 節點。
- 4 在 [目前的池] 下選取要變更的執行緒池的名稱。
- 5 在 [最小執行緒池大小] 欄位中，輸入為此佇列中的請求提供服務的執行緒池中執行緒的最小數目。
創設此執行緒池時將預先建立這些執行緒。
- 6 在 [最大執行緒池大小] 欄位中，輸入為此佇列中的請求提供服務的執行緒池中執行緒的最大數目。
這是存在於此執行緒池中的執行緒數上限。
- 7 在 [閒置逾時] 欄位中輸入數值 (以秒為單位)，超過此時間段之後將從池中移除閒置執行緒。
- 8 在 [工作佇列數] 欄位中，輸入由此執行緒池提供服務的工作佇列的總數。
- 9 按一下 [儲存]。
- 10 重新啟動 Application Server。

▼ 刪除執行緒池

- 1 在樹形元件中，選取 [配置] 節點。
- 2 選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 `server`，請選取 [server-config] 節點。
 - 若要配置所有實例的預設設定，請選取 [default-config] 節點。
- 3 選取 [執行緒池] 節點。
- 4 在 [目前的池] 表中選取要刪除的執行緒池名稱。
- 5 按一下 [刪除]。
- 6 重新啓動 **Application Server**。

更多資訊 **等效的 asadmin 指令**

```
delete-threadpool
```

配置記錄

本章簡要說明如何使用 管理主控台 來配置記錄和檢視伺服器記錄。它包含以下小節：

- 第 247 頁的「關於記錄」
- 第 249 頁的「有關記錄的管理主控台作業」

關於記錄

- 第 247 頁的「記錄檔的記錄」
- 第 248 頁的「記錄程式名稱空間階層結構」

記錄檔的記錄

Application Server 使用 JSR 047 中指定的 Java 2 平台記錄 API。Application Server 記錄訊息記錄在伺服器記錄中，通常位於 *domain-dir/logs/server.log* 中。

domain-dir/logs 目錄中除了包含伺服器記錄之外，還包含兩種其他類型的記錄。*access* 子目錄中包含 HTTP 服務存取記錄，*tx* 子目錄中包含作業事件服務記錄。如需有關這些記錄的資訊，請參閱第 224 頁的「配置 HTTP 服務存取記錄」和第 214 頁的「配置作業事件」。

Application Server 的元件可產生記錄輸出。應用程式元件也可以產生記錄輸出。

應用程式元件可以使用 Apache Commons Logging Library 來記錄訊息。但是，建議採用平台標準 JSR 047 API 以獲得更好的記錄配置。

記錄檔的記錄遵循以下統一格式：

```
[#|yyyy-mm-ddThh:mm:ss.SSS-Z|Log Level|ProductName-Version|LoggerName|Key Value Pairs|Message|#]
```

例如：

```
[#|2004-10-21T13:25:53.852-0400|INFO|sun-appserver-e8.1|javax.enterprise.
system.core|_ThreadID=13;|CORE5004: Resource Deployed:
[cr:jms/DurableConnectionFactory].|#]
```

在本範例中，

- [# 和 #] 標示該記錄的開頭和結尾。
- 垂直線 (|) 用於分隔記錄欄位。
- 2004-10-21T13:25:53.852-0400 指定了日期和時間。
- Log Level 為 INFO。此層級可以是以下任何值：SEVERE、WARNING、INFO、CONFIG、FINE、FINER 和 FINEST。
- ProductName-Version 為 sun-appserver-ee8.1。
- LoggerName 是用於識別記錄模組來源的階層式記錄程式名稱空間，在此範例中為 javax.enterprise.system.core。
- Key Value Pairs 為鍵名和鍵值，通常為執行緒 ID，例如 _ThreadID=14;。
- Message 是記錄訊息的文字。對於所有的 Application Server SEVERE 和 WARNING 訊息以及多種 INFO，它均以包含模組代碼和數值的訊息 ID 開頭 (在此範例中為 CORE5004)。

以後的版本中，可能會變更或增強記錄檔的記錄格式。

記錄程式名稱空間階層結構

Application Server 為它的每個模組都提供了記錄程式。下表依照模組名稱和名稱空間在管理主控台的 [記錄層級] 頁面中的顯示方式 (請參閱第 251 頁的「配置記錄層級」)，依字母順序列出每個記錄程式的模組名稱和名稱空間。[記錄層級] 頁面中未顯示表中最後三個模組。

表 15-1 Application Server 記錄程式名稱空間

模組名稱	名稱空間
管理	javax.enterprise.system.tools.admin
Classloader	javax.enterprise.system.core.classloading
CMP	javax.enterprise.system.container.cmp
配置	javax.enterprise.system.core.config
連接器	javax.enterprise.resource.resourceadapter
CORBA	javax.enterprise.resource.corba
部署	javax.enterprise.system.tools.deployment

表 15-1 Application Server 記錄程式名稱空間 (續)

模組名稱	名稱空間
EJB 容器	javax.enterprise.system.container.ejb
JavaMail	javax.enterprise.resource.javamail
JAXR	javax.enterprise.resource.webservices.registry
JAX-RPC	javax.enterprise.resource.webservices.rpc
JDO	javax.enterprise.resource.jdo
JMS	javax.enterprise.resource.jms
JTA	javax.enterprise.resource.jta
JTS	javax.enterprise.system.core.transaction
MDB 容器	javax.enterprise.system.container.ejb.mdb
命名	javax.enterprise.system.core.naming
節點代理程式 (僅限於 Enterprise Edition)	javax.ee.enterprise.system.nodeagent
根	javax.enterprise
SAAJ	javax.enterprise.resource.webservices.saaj
安全性	javax.enterprise.system.core.security
伺服器	javax.enterprise.system
同步 (僅限於 Enterprise Edition)	javax.ee.enterprise.system.tools.synchronization
Util	javax.enterprise.system.util
檢驗器	javax.enterprise.system.tools.verifier
Web 容器	javax.enterprise.system.container.web
核心	javax.enterprise.system.core
系統輸出 (System.out.println)	javax.enterprise.system.stream.out
系統錯誤 (System.err.println)	javax.enterprise.system.stream.err

有關記錄的管理主控台作業

- 第 250 頁的「配置一般記錄設定」
- 第 251 頁的「配置記錄層級」
- 第 252 頁的「檢視伺服器記錄」

▼ 配置一般記錄設定

- 1 在樹形元件中，展開 [節點代理程式] 節點或 [配置] 節點。
- 2 選取節點代理程式，或者選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 `server`，請選取 [server-config] 節點。
 - 若要為將來的實例 (實例使用 `default-config` 的副本) 配置預設設定，請選取 [default-config] 節點。
- 3 對於節點代理程式，請選取 [記錄程式設定] 標籤。對於配置，請選取 [記錄程式設定] 節點。
- 4 在 [記錄設定] 頁面中，可以使用以下欄位來自訂記錄：
 - **記錄檔** — 若要為伺服器記錄檔指定替代名稱或位置，請在文字欄位中鍵入新的路徑名稱。依預設，該位置為 `domain-dir/logs/server.log`。
 - **警示** — 若要經由 JMX 架構路由 SEVERE 和 WARNING 訊息，請選取 [啟用] 核取方塊。
 - **寫入系統記錄** — 僅在 Solaris 和 Linux 系統中，當除了要將記錄輸出傳送給伺服器記錄外，還要將其傳送給 `syslog` 工具時，請選取 [啟用] 核取方塊。
 - **記錄處理程式** — 若要將記錄傳送到 `server.log` 或 `syslog` 以外的目標，您可以插入自訂記錄處理程式。自訂處理程式必須延伸 `java.util.logging.Handler` 類別 (JSR 047 相容 API)。在 [記錄處理程式] 欄位中鍵入處理程式的絕對類別名稱。還應將處理程式類別置於 `Application Server` 類別路徑中，以便在伺服器啟動期間安裝該處理程式。自訂處理程式之記錄檔記錄將具有第 247 頁的「記錄檔的記錄」中說明的格式。
 - **記錄篩選器** — 若要篩選傳送給目標 (例如 `server.log`、`syslog` 或由自訂記錄處理程式指定的目標) 的記錄檔記錄，可以插入自訂記錄篩選器。該自訂篩選器必須實作 `java.util.logging.Filter` 介面。在 [記錄篩選器] 欄位中鍵入篩選器的絕對類別名稱。還應將篩選器類別置於 `Application Server` 類別路徑中，以便在伺服器啟動期間安裝該篩選器。
 - **檔案自動重建限制**：如果伺服器記錄的位元組數達到了指定大小，請建立一個名為 `server.log` 的新的空檔案，並將舊檔案重新命名為 `server.log_date`，其中 `date` 是自動重建檔案的日期和時間。預設值為 2 百萬位元組。該限制的最小值為 500 千位元組；如果指定較低的值，則要在達到 500 KB 時該檔案才自動重建。若要關閉記錄檔自動重建，請將該值設定為 0。
 - **檔案自動重建時間限制** — 達到指定的分鐘數之後才自動重建伺服器記錄。預設值為零，這表示檔案達到 [檔案自動重建限制] 欄位中指定的大小即會自動重建。如果指定了非零的時間限制，則該時間限制優先於上述大小限制。

- 按一下 [儲存]，以儲存變更。按一下 [檢視記錄檔] 以檢視伺服器記錄。

▼ 配置記錄層級

- 在樹形元件中，展開 [節點代理程式] 節點或 [配置] 節點。
- 選取節點代理程式，或者選取要配置的實例：
 - 若要配置特定的實例，請選取該實例的配置節點。例如，對於預設實例 `server`，請選取 `[server-config]` 節點。
 - 若要為將來的實例 (實例使用 `default-config` 的副本) 配置預設設定，請選取 `[default-config]` 節點。
- 對於節點代理程式，請選取 [記錄層級] 標籤。對於配置，請選取 [記錄程式設定] 節點，然後選取 [記錄層級] 標籤。

- 在 [模組記錄層級] 頁面中，請從要變更記錄層級的一個或多個模組相對的下拉式清單中選擇一個新值。

預設層級為 `INFO`，表示處於該層級或更高層級 (`WARNING`、`SEVERE`) 的訊息將顯示在記錄中。可以選擇以下任一值 (以最高層級至最低層級的順序列示)：

- SEVERE
- WARNING
- INFO
- CONFIG
- FINE
- FINER
- FINEST
- OFF

- 使用 [附加特性] 區域可以為任何應用程式記錄程式配置記錄層級。

特性名稱是記錄程式名稱空間，其值為上述八個可能的層級之一。例如，特性名稱可為 `samples.logging.simple.servlet`，值可為 `FINE`。

還可以使用此區域來變更子模組的記錄層級，例如 `CORBA` 模組的傳輸子模組：

```
javax.enterprise.resource.corba.ORBId.transport
```

- 按一下 [儲存] 以儲存變更，或按一下 [預設] 以復原預設值。

對 `System.out.println` 的呼叫均在 `INFO` 層級使用記錄程式名稱

`javax.enterprise.system.stream.out` 予以記錄。對 `System.err.println` 的呼叫均在 `WARNING` 層級使用記錄程式名稱 `javax.enterprise.system.stream.err` 予以記錄。若要從這些來源關閉記錄，請在 [附加特性] 區域中將記錄程式名稱的值指定為 `OFF`。

對記錄層級設定所做的變更將立即生效。同時將在 domain.xml 檔案中儲存這些變更，以供伺服器重新啟動時使用。

▼ 檢視伺服器記錄

- 1 在樹形元件中，展開要檢視其記錄的伺服器實例的節點。
- 2 在[一般資訊]頁面中，按一下[檢視記錄檔]。
- 3 使用[搜尋條件]區域自訂和篩選記錄檢視器。使用如下所示的基本欄位：
 - **實例名稱** — 從下拉式清單中選擇一個實例名稱，以檢視該伺服器實例的記錄。預設值為目前伺服器實例。
 - **記錄檔** — 從下拉式清單中選擇一個記錄檔名稱，以檢視該記錄的內容。預設為 server.log。
 - **時間戳記** — 若要檢視最新的訊息，請選取[最新](預設)。如果只檢視特定時間段內的訊息，請選取[特定範圍]並在顯示的[從]欄位和[到]欄位中鍵入日期和時間值。對於時間值，其語法必須採用以下格式(SSS表示毫秒)：

hh:mm:ss.SSS

例如：

17:10:00.000

如果[從]欄位中的時間值遲於[到]欄位中的時間值，將顯示錯誤訊息。

- **記錄層級** — 若要依照記錄層級篩選訊息，請從下拉式清單中選擇一種記錄層級。依預設，將顯示伺服器記錄中選取記錄層級和更嚴重記錄層級的所有訊息。選取標有[不包含更高層級的訊息]的核取方塊，可僅顯示所選層級的訊息。

若要確定您要檢視的訊息都顯示在伺服器記錄中，請先在[記錄層級]頁面中設定適當的記錄層級。請參閱第 251 頁的「配置記錄層級」。

如果您選擇基於記錄層級篩選記錄訊息，則將只顯示符合指定篩選條件的訊息。不過，這種篩選不影響那些記錄到伺服器記錄中的訊息。

將顯示伺服器記錄中最新的 40 個項目以及在[記錄設定]和[記錄層級]頁面中指定的設定。

按一下[時間戳記]標頭旁邊的三角形對這些訊息進行排序，以使最新的訊息顯示在最後。

若要檢視訊息的格式化版本，請按一下標示的連結

(details)

將顯示標有[記錄項目詳細資訊]的視窗，該視窗包含了訊息的格式化版本。

在項目清單的末尾，按一下按鈕以檢視記錄檔中較早或較晚的項目。

按一下 [搜尋條件] 區域中的 [進階搜尋] 以進一步細化記錄檢視器的搜尋條件。使用如下所示的 [進階選項] 欄位：

- **記錄程式** — 若要依模組進行篩選，請從下拉式清單中選擇一個或多個名稱空間。按住 Shift 鍵並按一下或按住 Ctrl 鍵並按一下來選擇多個名稱空間。

選取更高層級的名稱空間也就選取了該名稱空間下的所有名稱空間。例如，選取 `javax.enterprise.system` 的同時也就選取了該名稱空間下所有模組的記錄程式：`javax.enterprise.system.core`、`javax.enterprise.system.tools.admin` 等等。

- **自訂記錄程式** — 若要檢視特定於某個特定應用程式的記錄程式中的訊息，請在文字欄位中鍵入記錄程式名稱 (每行一個名稱)。如果應用程式具有多個模組，您可以檢視任何模組，也可以檢視所有模組。例如，假定應用程式具有使用以下名稱的記錄程式：

```
com.mycompany.myapp.module1
com.mycompany.myapp.module2
com.mycompany.myapp.module3
```

若要檢視應用程式中的所有模組的訊息，請鍵入 `com.mycompany.myapp`。若要僅檢視 `module2` 的訊息，請鍵入 `com.mycompany.myapp.module2`。

如果指定了一個或多個自訂記錄程式，則僅當您在 [記錄程式] 區域中明確指定 Application Server 模組的訊息後，才會顯示這些訊息。

- **名稱值對** — 若要檢視特定執行緒的輸出，請在文字欄位中鍵入該執行緒的鍵名和鍵值。鍵名為 `_ThreadID`。例如：

```
_ThreadID=13
```

假定 `com.mycompany.myapp.module2` 在多個執行緒中執行。若要限制記錄檢視器使其只顯示單一執行緒的輸出，請在 [自訂記錄程式] 欄位中指定該模組的記錄程式，然後在此欄位中指定執行緒 ID。

- **顯示** — 若要一次檢視 40 條以上的訊息 (預設)，請從下拉式清單中選擇其他可用的值 (100、250 或 1000)。

若要檢視堆疊追蹤，請取消選取 [限制過長訊息] 核取方塊。依預設，檢視器中不會顯示堆疊追蹤；若要檢視堆疊追蹤，請按一下訊息的 [(詳細資訊)] 連結。

按一下 [基本搜尋] 可以隱藏 [進階選項] 區域。

◆◆◆ 第 16 章

監視元件和服務

本章包含有關使用 Application Server 管理主控台 來監視元件的資訊。本章包括下列小節：

- 第 255 頁的「關於監視」
- 第 279 頁的「有關啓用和停用監視功能的管理主控台作業」
- 第 281 頁的「有關檢視監視資料的管理主控台作業」
- 第 296 頁的「使用 JConsole」

關於監視

- 第 255 頁的「監視 Application Server」
- 第 255 頁的「監視概述」
- 第 256 頁的「關於可監視物件的樹狀結構」
- 第 259 頁的「關於受監視的元件和服務的統計資訊」

監視 Application Server

使用監視功能可以觀察在 Application Server 的伺服器實例中所部署的各種元件和服務的執行階段狀態。利用有關執行階段元件和程式狀態的資訊，可以確定效能瓶頸以便進行調校、和容量規劃、預測故障、在發生故障時分析根本原因，以及確保一切執行正常。

啓用監視功能會因增加系統耗用而使效能降低。

監視概述

若要監視 Application Server，請執行以下步驟：

1. 使用 管理主控台 或 `asadmin` 工具啓用對特定服務和元件的監視功能。

如需有關此步驟的更多資訊，請參閱第 279 頁的「有關啓用和停用監視功能的管理主控台作業」。

2. 使用管理主控台或 `asadmin` 工具檢視指定服務或元件的監視資料。

如需有關此步驟的更多資訊，請參閱第 281 頁的「有關檢視監視資料的管理主控台作業」。

關於可監視物件的樹狀結構

Application Server 使用樹狀結構追蹤可監視物件。由於監視物件的樹是動態的，因此在實例中增加、更新或移除元件時該樹會相應地發生變更。樹狀結構中的根物件是伺服器實例名稱 (例如 `server`)。(在 Platform Edition 中，僅允許使用一個伺服器實例。)

以下指令顯示了樹的頂層：

```
asadmin> list --user adminuser --monitor server
server.applications
server.http-service
server.connector-service
server.jms-service
server.jvm
server.orb
server.resources
server.thread-pools
```

以下小節描述了這些子樹：

- 第 256 頁的「應用程式樹」
- 第 257 頁的「HTTP 服務樹」
- 第 258 頁的「資源樹」
- 第 258 頁的「連接器服務樹」
- 第 258 頁的「JMS 服務樹」
- 第 259 頁的「ORB 樹」
- 第 259 頁的「執行緒池樹」

應用程式樹

以下示意圖顯示了企業應用程式的各種元件的頂層節點和子節點。具有可用的監視統計資訊的節點標有星號 (*)。如需更多資訊，請參閱第 260 頁的「EJB 容器統計資訊」。

範例 16-1 應用程式節點樹狀結構

```
applications
  |-- application1
  |   |-- ejb-module-1
  |       |-- ejb1 *
```


範例 16-3 HTTP 服務示意圖 (Enterprise Edition 版)

```

http-service *
  |--connection-queue *
  |--dns *
  |--file-cache *
  |--keep-alive *
  |--pwc-thread-pool *
  |--virtual-server-1*
  |       |-- request *
  |--virtual-server-2*
  |       |-- request *

```

資源樹

資源節點包含 JDBC 連線池和連接器連線池等池的可監視屬性。以下示意圖顯示了各種資源元件的頂層節點和子節點。具有可用的監視統計資訊的節點標有星號 (*)。請參閱第 266 頁的「JDBC 連線池統計資訊」。

範例 16-4 資源示意圖

```

resources
  |--connection-pool1(either connector-connection-pool or jdbc)*
  |--connection-pool2(either connector-connection-pool or jdbc)*

```

連接器服務樹

連接器服務節點包含連接器連線池等池的可監視屬性。以下示意圖顯示了各種連接器服務元件的頂層節點和子節點。具有可用的監視統計資訊的節點標有星號 (*)。請參閱第 266 頁的「JMS/連接器服務統計資訊」。

範例 16-5 連接器服務示意圖

```

connector-service
  |-- resource-adapter-1
  |       |-- connection-pools
  |       |       |-- pool-1 (All pool stats for this pool)
  |       |-- work-management (All work mgmt stats for this RA)

```

JMS 服務樹

JMS 服務節點包含連接器連線池等池的可監視屬性。以下示意圖顯示了各種 JMS 服務元件的頂層節點和子節點。具有可用的監視統計資訊的節點標有星號 (*)。

範例 16-6 JMS 服務示意圖

```
jms-service
|-- connection-factories [AKA conn. pools in the RA world]
|   |-- connection-factory-1 (All CF stats for this CF)
|-- work-management (All work mgmt stats for the MQ-RA)
```

ORB 樹

ORB 節點包含連線管理員的可監視屬性。以下示意圖顯示了 ORB 元件的頂層節點和子節點。具有可用的監視統計資訊的節點標有星號 (*)。請參閱第 268 頁的「ORB 中連線管理程式的統計資訊」。

範例 16-7 ORB 示意圖

```
orb
|--- connection-managers
|   |--- connection-manager-1 *
|   |--- connection-manager-1 *
```

執行緒池樹

執行緒池節點包含連線管理程式的可監視屬性。以下示意圖顯示了 ORB 元件的頂層節點和子節點。具有可用的監視統計資訊的節點標有星號 (*)。請參閱第 268 頁的「執行緒池統計資訊」。

範例 16-8 執行緒池示意圖

```
thread-pools
|   |--- thread-pool-1 *
|   |--- thread-pool-2 *
```

關於受監視的元件和服務的統計資訊

本小節描述可用的監視統計資訊：

- 第 260 頁的「EJB 容器統計資訊」
- 第 263 頁的「Web 容器統計資訊」
- 第 265 頁的「HTTP 服務統計資訊」
- 第 266 頁的「JDBC 連線池統計資訊」
- 第 266 頁的「JMS/連接器服務統計資訊」
- 第 268 頁的「ORB 中連線管理程式的統計資訊」
- 第 268 頁的「執行緒池統計資訊」
- 第 268 頁的「作業事件服務統計資訊」
- 第 269 頁的「Java 虛擬機器 (JVM) 統計資訊」
- 第 269 頁的「J2SE 5.0 中的 JVM 統計資訊」

- 第 273 頁的「生產 Web 容器 (PWC) 統計資訊」

EJB 容器統計資訊

下表說明 EJB 統計。

表 16-1 EJB 統計資訊

屬性名稱	資料類型	說明
createcount	計數統計資訊	呼叫 EJB 的 <code>create</code> 方法的次數。
removecount	計數統計資訊	呼叫 EJB 的 <code>remove</code> 方法的次數。
pooledcount	範圍統計	處於匯集狀態的實體 Bean 的數目。
readycount	範圍統計	處於就緒狀態的實體 Bean 的數目。
messagecount	計數統計資訊	訊息導引 Bean 收到的訊息數。
methodreadycount	範圍統計	處於 <code>MethodReady</code> 狀態的有狀態或無狀態階段作業 Bean 的數目。
passivecount	範圍統計	處於 <code>Passive</code> 狀態的有狀態階段作業 Bean 的數目。

下表中列出了可用於 EJB 方法呼叫的統計。

表 16-2 EJB 方法統計資訊

屬性名稱	資料類型	說明
methodstatistic	時間統計	作業被呼叫的次數；呼叫期間所花費的總時間等資訊。
totalnumerrors	計數統計資訊	執行方法導致異常的次數。如果為 EJB 容器啓用了監視功能，則此統計資訊是為無狀態和有狀態階段作業 Bean 和實體 Bean 而收集的。
totalnumsuccess	計數統計資訊	方法成功執行的次數。如果為 EJB 容器啓用了監視功能，則此統計資訊是為無狀態和有狀態階段作業 Bean 和實體 Bean 收集的。

表 16-2 EJB 方法統計資訊 (續)

屬性名稱	資料類型	說明
executiontime	計數統計資訊	上次成功/不成功嘗試執行方法作業所花費的時間 (ms)。如果在 EJB 容器中啓用了監視功能，則此統計資訊是為無狀態和有狀態階段作業 Bean 和實體 Bean 而收集的。

下表中列出了 EJB 階段作業儲存的統計。

表 16-3 EJB 階段作業儲存統計

屬性名稱	資料類型	說明
currentSize	範圍統計	目前位於儲存中的鈍化階段作業數或檢查點階段作業數。
activationCount	計數統計資訊	從儲存中啓動的階段作業數。
activationSuccessCount	計數統計資訊	從儲存中成功啓動的階段作業數
activationErrorCount	計數統計資訊	上次成功/不成功嘗試執行方法作業所花費的時間 (ms)。如果在 EJB 容器中啓用了監視功能，則此統計資訊是為無狀態和有狀態階段作業 Bean 和實體 Bean 而收集的。
passivationCount	計數統計資訊	使用此儲存鈍化 (取消啓動) 的階段作業數。
passivationSuccessCount	計數統計資訊	使用此儲存成功鈍化的階段作業數。
passivationErrorCount	計數統計資訊	無法使用此儲存鈍化的階段作業數。
expiredSessionCount	計數統計資訊	此儲存移除的過期階段作業數。
passivatedBeanSize	計數統計資訊	由此儲存鈍化的總位元組數，包括總數、最小值和最大值。
passivationTime	計數統計資訊	將 Bean 鈍化到儲存所花費的時間，包括總時間值、最小值和最大值。
checkpointCount (僅限於 EE)	計數統計資訊	使用此儲存進行階段作業檢查點操作的階段作業數。

表 16-3 EJB 階段作業儲存統計 (續)

屬性名稱	資料類型	說明
checkpointSuccessCount (僅限於 EE)	計數統計資訊	成功進行檢查點操作的階段作業數。
checkpointErrorCount (僅限於 EE)	計數統計資訊	無法進行檢查點操作的階段作業數。
checkpointedBeanSize (僅限於 EE)	值統計	由該儲存進行檢查點操作的 Bean 的總數。
checkpointTime (僅限於 EE)	時間統計	通過檢查點操作將 Bean 放入儲存中所花費的時間。

下表中列出了可用於 EJB 池的統計。

表 16-4 EJB 池統計資訊

屬性名稱	資料類型	說明
numbeansinpool	限制範圍統計	相關聯池中的 EJB 數，可提供有關池的變更方式的資訊。
numthreadswaiting	限制範圍統計	等待自由 Bean 的執行緒數，指出請求可能擁塞。
totalbeanscreated	計數統計資訊	從開始收集資料以來相關聯池中所建立的 Bean 的數目。
totalbeansdestroyed	計數統計資訊	自開始收集資料以來從相關聯池中銷毀的 Bean 的數目。
jmsmaxmessagesload	計數統計資訊	針對要服務的訊息導引 Bean 而一次載入 JMS 階段作業的最大訊息數。預設值為 1。僅套用於訊息導引 Bean 的池。

下表中列出了可用於 EJB 快取的統計。

表 16-5 EJB 快取統計

屬性名稱	資料類型	說明
cachemisses	限制範圍統計	使用者請求未在快取中找到 Bean 的次數。
cachehits	限制範圍統計	使用者請求找到快取記憶體中某個項目的次數。

表 16-5 EJB 快取統計 (續)

屬性名稱	資料類型	說明
numbeansincache	限制範圍統計	快取記憶體中 Bean 的數目。這便是快取記憶體目前的大小。
numpassivations	計數統計資訊	鈍化 Bean 的數目。僅套用於有狀態階段作業 Bean。
numpassivationerrors	計數統計資訊	鈍化期間發生的錯誤數。僅套用於有狀態階段作業 Bean。
numexpiredsessionsremoved	計數統計資訊	清除執行緒所移除的過期階段作業數目。僅套用於有狀態階段作業 Bean。
numpassivationsuccess	計數統計資訊	鈍化成功完成的次數。僅套用於有狀態階段作業 Bean。

下表中列出了可用於計時器的統計。

表 16-6 計時器統計資訊

統計資訊	資料類型	說明
numtimerscreated	計數統計資訊	系統中建立的計時器的數目。
numtimersdelivered	計數統計資訊	系統所傳送的計時器的數目。
numtimersremoved	計數統計資訊	從系統中移除的計時器的數目。

Web 容器統計資訊

物件樹狀結構中包含了 Web 容器，如第 256 頁的「應用程式樹」所示。系統為每個單獨的 Web 應用程式都顯示了 Web 容器統計資訊。第 263 頁的「Web 容器統計資訊」中列出了可用於 Servlet 的 Web 容器的統計，第 263 頁的「Web 容器統計資訊」中列出了可用於 Web 模組的統計。

表 16-7 Web 容器 (Servlet) 統計資訊

統計資訊	單位	資料類型	注釋
errorcount	數目	計數統計資訊	回應碼大於或等於 400 的情況的累積次數。
maxtime	毫秒	計數統計資訊	Web 容器等待請求的最長時間。
processingtime	毫秒	計數統計資訊	處理每個請求所需時間的累積值。處理時間是總請求處理時間除以請求計數所得的平均值。

表 16-7 Web 容器 (Servlet) 統計資訊 (續)

統計資訊	單位	資料類型	注釋
requestcount	數目	計數統計資訊	到目前為止所處理的請求總數。

第 263 頁的「Web 容器統計資訊」中列出了可用於 Web 模組的統計。

表 16-8 Web 容器 (Web 模組) 統計資訊

統計資訊	資料類型	注釋
jspcount	計數統計資訊	已載入 Web 模組的 JSP 頁面的數目。
jspreloadcount	計數統計資訊	已重新載入 Web 模組的 JSP 頁面的數目。
sessionstotal	計數統計資訊	已為 Web 模組建立的階段作業總數。
activesessionscurrent	計數統計資訊	Web 模組的目前處於使用中狀態的階段作業數。
activesessionshigh	計數統計資訊	Web 模組的同時處於使用中狀態的階段作業最大數。
rejectedsessionstotal	計數統計資訊	Web 模組的被拒絕的階段作業總數。是指由於允許處於使用中狀態的階段作業數已達到最大值而未被建立的階段作業數。
expiredsessionstotal	計數統計資訊	Web 模組的已過期階段作業的總數。
sessionsize (僅限於 EE)	平均範圍統計	Web 模組的階段作業大小。值可以為大、小或平均值，或以位元組為單位 (用於序列化階段作業)。
containerlatency (僅限於 EE)	平均範圍統計	整個延時請求中 Web 容器部分的延時。值可以是長、短或平均值。
sessionpersisttime (僅限於 EE)	平均範圍統計	將 HTTP 階段作業狀態保持到 Web 模組的後端儲存中所花費的時間 (以 ms 為單位，短、長或平均值)。
cachedsessionscurrent (僅限於 EE)	計數統計資訊	快取在記憶體中用於 Web 模組的目前階段作業數。

表 16-8 Web 容器 (Web 模組) 統計資訊 (續)

統計資訊	資料類型	注釋
passivatedsessionscurrent (僅限於 EE)	計數統計資訊	Web 模組的目前已鈍化的階段作業數。

HTTP 服務統計資訊

第 265 頁的「HTTP 服務統計資訊」中列出了可用於 HTTP 服務的統計。這些統計資訊僅適用於 Platform Edition。如需有關 Enterprise Edition 上的 HTTP 服務的統計，請參閱第 273 頁的「生產 Web 容器 (PWC) 統計資訊」。

表 16-9 HTTP 服務統計資訊 (僅適用於 Platform Edition)

統計資訊	單位	資料類型	注釋
bytesreceived	位元組	計數統計資訊	每個請求處理器接收到的位元組累積值。
bytessent	位元組	計數統計資訊	每個請求處理器所傳送的位元組累積值。
currentthreadcount	數目	計數統計資訊	目前位於偵聽程式執行緒池中的處理執行緒數。
currentthreadsbusy	數目	計數統計資訊	處理請求的偵聽程式執行緒池中目前正在使用的請求處理執行緒的數目。
errorcount	數目	計數統計資訊	錯誤計數的累積值，錯誤計數是指回應碼大於或等於 400 這類情況發生的次數。
maxsparethreads	數目	計數統計資訊	可以存在的未使用回應處理執行緒的最大數目。
minsparethreads	數目	計數統計資訊	可以存在的未使用回應處理執行緒的最小數目。
maxthreads	數目	計數統計資訊	偵聽程式所建立的請求處理執行緒的最大數目。
maxtime	毫秒	計數統計資訊	處理執行緒的最長時間。
processing-time	毫秒	計數統計資訊	處理每個請求所花費時間的累積值。處理時間是總請求處理時間除以請求計數所得的平均值。
request-count	數目	計數統計資訊	到目前為止所處理的請求總數。

JDBC 連線池統計資訊

用於在執行階段監視 JDBC 資源，以測量效能並擷取資源使用情況。由於建立 JDBC 連線的成本很高並且常常會導致應用程式出現效能瓶頸問題，因此對 JDBC 連線池如何釋放和建立新連線以及正在等待從特定池中擷取連線的執行緒數的監視是至關重要的。

下表中列出了可用於 JDBC 連線池的統計。

表 16-10 JDBC 連線池統計

統計資訊	單位	資料類型	說明
numconnfailedvalidation	數目	計數統計資訊	從開始時間到上次取樣時間為止在連線池中驗證失敗的連線總數。
numconnused	數目	範圍統計	提供連線使用統計資訊。目前正使用的連線總數，以及有關使用過的連線的最大數目的資訊 (高水印)。
numconnfree	數目	範圍統計	上次取樣時池中的自由連線總數。
numconntimedout	數目	限制範圍統計	開始時間與上次取樣時間之間池中的逾時連線總數。
averageconnwaittime	數目	計數統計資訊	指示嘗試與連接器連線池建立成功連線請求的平均等待時間。
waitqueuelength	數目	計數統計資訊	佇列中正在等待處理的連線請求數。
connectionrequestwaittime		範圍統計	連線請求的最長和最短等待時間。目前值表示上次由池處理的請求的等待時間。
numconncreated	毫秒	計數統計資訊	自上次重設以來建立的實體連線數。
numconndestroyed	數目	計數統計資訊	自上次重設以來已銷毀的實體連線數。
numconnacquired	數目	計數統計資訊	從池中獲取的邏輯連線數。
numconnreleased	數目	計數統計資訊	釋放到池中的邏輯連線數。

JMS/連接器服務統計資訊

第 266 頁的「JMS/連接器服務統計資訊」中列出了可用於連接器連線池的統計。第 266 頁的「JMS/連接器服務統計資訊」中列出了連接器工作管理的統計。

表 16-11 連接器連線池統計資訊

統計資訊	單位	資料類型	說明
numconnfailedvalidation	數目	計數統計資訊	從開始時間到上次取樣時間為止在連線池中驗證失敗的連線總數。
numconnused	數目	範圍統計	提供連線使用統計資訊。目前正使用的連線總數，以及有關使用過的連線的最大數目的資訊(高水印)。
numconnfree	數目	範圍統計	上次取樣時池中的自由連線總數。
numconntimeout	數目	計數統計資訊	開始時間與上次取樣時間之間池中的逾時連線總數。
averageconnwaittime	數目	計數統計資訊	連線由連線池處理之前的平均等待時間。
waitqueueleight	數目	計數統計資訊	佇列中正在等待處理的連線請求數。
connectionrequestwaittime		範圍統計	連線請求的最長和最短等待時間。目前值表示上次由池處理的請求的等待時間。
numconncreated	毫秒	計數統計資訊	自上次重設以來建立的實體連線數。
numconndestroyed	數目	計數統計資訊	自上次重設以來已銷毀的實體連線數。
numconnacquired	數目	計數統計資訊	從池中獲取的邏輯連線數。
numconnreleased	數目	計數統計資訊	釋放到池中的邏輯連線數。

第 266 頁的「JMS/連接器服務統計資訊」中列出了可用於連接器工作管理的統計。

表 16-12 連接器工作管理統計資訊

統計資訊	資料類型	說明
activeworkcount	範圍統計	由連接器執行的工作物件數。
waitqueuelength	範圍統計	執行前在佇列中等待的工作物件數。
workrequestwaittime	範圍統計	工作物件在被執行前所等待的最長和最短時間。
submittedworkcount	計數統計資訊	由連接器模組提交的工作物件數。
rejectedworkcount	計數統計資訊	Application Server 拒絕的工作物件數。
completedworkcount	計數統計資訊	完成的工作物件數。

ORB 中連線管理程式的統計資訊

第 268 頁的「ORB 中連線管理程式的統計資訊」中列出了可用於 ORB 中連線管理員的統計。

表 16-13 ORB 中連線管理程式的統計資訊

統計資訊	單位	資料類型	說明
connectionsidle	數目	計數統計資訊	提供與 ORB 的閒置連線的總數。
connectionsinuse	數目	計數統計資訊	提供 ORB 的使用中連線總數。
totalconnections	數目	限制範圍統計	與 ORB 的連線總數。

執行緒池統計資訊

下表中列出了可用於執行緒池的統計。

表 16-14 執行緒池統計資訊

統計資訊	單位	資料類型	說明
averagetimeinqueue	毫秒	範圍統計	在得到處理之前請求在佇列中等待的平均時間(以毫秒為單位)。
averageworkcompletion-time	毫秒	範圍統計	完成指定所花費的平均時間(以毫秒為單位)。
currentnumberofthreads	數目	限制範圍統計	目前的請求處理執行緒的數目。
numberofavailablethreads	數目	計數統計資訊	可用的執行緒數。
numberofbusythreads	數目	計數統計資訊	處於忙碌狀態的執行緒數。
totalworkitemsadded	數目	計數統計資訊	到目前為止增加到工作佇列中的工作項目總數。

作業事件服務統計資訊

作業事件服務允許用戶端凍結作業事件子系統，以回復作業事件，並確定在凍結期間進行處理的作業事件。下表中列出了可用於作業事件服務的統計。

表 16-15 作業事件服務統計資訊

統計資訊	資料類型	說明
activecount	計數統計資訊	目前處於使用中狀態的作業事件數。

表 16-15 作業事件服務統計資訊 (續)

統計資訊	資料類型	說明
activeids	字串統計資訊	目前處於使用中狀態的作業事件的 ID。每個此類作業事件均可在凍結作業事件服務後轉返。
committedcount	計數統計資訊	已確定的作業事件數。
rolledbackcount	計數統計資訊	已轉返的作業事件數。
state	字串統計資訊	指示作業事件是否已凍結。

Java 虛擬機器 (JVM) 統計資訊

JVM 具有始終處於啟用狀態的可監視屬性。下表中列出了可用於 JVM 的統計。

表 16-16 JVM 統計資訊

統計資訊	資料類型	說明
heapsize	限制範圍統計	包含 JVM 的記憶體堆疊大小上限和下限的常駐記憶體佔用空間。
uptime	計數統計資訊	JVM 已執行的時間。

J2SE 5.0 中的 JVM 統計資訊

如果將 Application Server 配置為在 J2SE 5.0 版或更高版本上執行，則可以從 JVM 中取得附加監視資訊。將監視層級設定為 [低] 以啟用這些附加資訊的顯示。將監視層級設定為 [高] 還可以檢視與系統中每個活性執行緒相關的資訊。如需有關 J2SE 5.0 中可用的附加監視功能的更多資訊，請參閱以下位置中標題為「*Monitoring and Management for the Java Platform*」的文件：<http://java.sun.com/j2se/1.5.0/docs/guide/management/>。

以下位置中論述了 J2SE 5.0 監視工

具：<http://java.sun.com/j2se/1.5.0/docs/tooldocs/#manage>。

第 269 頁的「J2SE 5.0 中的 JVM 統計資訊」中列出了可用於在 J2SE 5.0 的 JVM 中進行類別載入的統計。

表 16-17 J2SE 5.0 的 JVM 統計資訊 - 類別載入

統計資訊	資料類型	說明
loadedclasscount	計數統計資訊	目前載入 JVM 的類別的數目。
totalloadedclasscount	計數統計資訊	自 JVM 開始執行以來已載入的類別的總數。

表 16-17 J2SE 5.0 的 JVM 統計資訊 - 類別載入 (續)

統計資訊	資料類型	說明
unloadedclasscount	計數統計資訊	自 JVM 開始執行以來已從 JVM 中卸載的類別的數目。

第 269 頁的「[J2SE 5.0 中的 JVM 統計資訊](#)」中列出了可用於在 J2SE 5.0 的 JVM 中進行編譯的統計。

表 16-18 J2SE 5.0 的 JVM 統計資訊 - 編譯

統計資訊	資料類型	說明
totalcompilationtime	計數統計資訊	編譯所花費的累積時間 (以毫秒為單位)。

第 269 頁的「[J2SE 5.0 中的 JVM 統計資訊](#)」中列出了可用於在 J2SE 5.0 的 JVM 中進行資源回收的統計。

表 16-19 J2SE 5.0 的 JVM 統計資訊 - 資源回收

統計資訊	資料類型	說明
collectioncount	計數統計資訊	已發生的回收的總數。
collectiontime	計數統計資訊	累積的收集時間 (以毫秒為單位)。

第 269 頁的「[J2SE 5.0 中的 JVM 統計資訊](#)」中列出了可用於 J2SE 5.0 的 JVM 中的記憶體統計。

表 16-20 J2SE 5.0 的 JVM 統計資訊 - 記憶體

統計資訊	資料類型	說明
objectpendingfinalizationcount	計數統計資訊	擱置結束操作的物件的大約數目。
initheapsize	計數統計資訊	最初由 JVM 請求的堆疊的大小。
usedheapsize	計數統計資訊	目前正在使用的堆疊的大小。
maxheapsize	計數統計資訊	可用於記憶體管理的最大記憶體容量 (以位元組為單位)。
committedheapsize	計數統計資訊	確定可由 JVM 使用的記憶體容量 (以位元組為單位)。
initnonheapsize	計數統計資訊	最初由 JVM 請求的非堆疊區域的大小。

表 16-20 J2SE 5.0 的 JVM 統計資訊 - 記憶體 (續)

統計資訊	資料類型	說明
usednonheapsize	計數統計資訊	目前正在使用的非堆疊區域的大小。
maxnonheapsize	計數統計資訊	可用於記憶體管理的最大記憶體容量 (以位元組為單位)。
committednonheapsize	計數統計資訊	確定可由 JVM 使用的記憶體容量 (以位元組為單位)。

第 269 頁的「J2SE 5.0 中的 JVM 統計資訊」中列出了可用於 J2SE 5.0 的 JVM 中的作業系統的統計。

表 16-21 J2SE 5.0 的 JVM 統計資訊 - 作業系統

統計資訊	資料類型	說明
arch	字串統計資訊	作業系統架構。
availableprocessors	計數統計資訊	可用於 JVM 的處理器的數目。
name	字串統計資訊	作業系統名稱。
version	字串統計資訊	作業系統版本。

第 269 頁的「J2SE 5.0 中的 JVM 統計資訊」中顯示了可用於 J2SE 5.0 的 JVM 中的執行階段的統計。

表 16-22 J2SE 5.0 的 JVM 統計資訊 - 執行階段

統計資訊	資料類型	說明
name	字串統計資訊	表示正在執行的 JVM 的名稱
vmname	字串統計資訊	JVM 實作名稱。
vmvendor	字串統計資訊	JVM 實作供應商。
vmversion	字串統計資訊	JVM 實作版本。
specname	字串統計資訊	JVM 規格名稱。
specvendor	字串統計資訊	JVM 規格供應商。
specversion	字串統計資訊	JVM 規格版本。
managementspecversion	字串統計資訊	由 JVM 實作的管理規格版本。
classpath	字串統計資訊	系統類別載入器搜尋類別檔案時所使用的類別路徑。

表 16-22 J2SE 5.0 的 JVM 統計資訊 - 執行階段 (續)

統計資訊	資料類型	說明
librarypath	字串統計資訊	Java 程式庫路徑。
bootclasspath	字串統計資訊	啟動程式類別載入器搜尋類別檔案時所使用的類別路徑。
inputarguments	字串統計資訊	傳送給 JVM 的輸入引數。不包括 main 方法的引數。
uptime	計數統計資訊	JVM 的正常執行時間 (以毫秒為單位)。

第 269 頁的「J2SE 5.0 中的 JVM 統計資訊」中列出了可用於 J2SE 5.0 的 JVM 中的 ThreadInfo 的統計。

表 16-23 J2SE 5.0 的 JVM 統計 — 執行緒資訊

統計資訊	資料類型	說明
threadid	計數統計資訊	執行緒 ID。
threadname	字串統計資訊	執行緒名稱。
threadstate	字串統計資訊	執行緒狀態。
blockedtime	計數統計資訊	執行緒進入 BLOCKED 狀態以來所經歷的時間 (以毫秒為單位)。如果已停用執行緒競爭狀態監視功能，則傳回 -1。
blockedcount	計數統計資訊	執行緒進入 BLOCKED 狀態的總次數。
waitedtime	計數統計資訊	執行緒處於 WAITING 狀態以來所經歷的時間 (以毫秒為單位)。如果已停用執行緒競爭狀態監視功能，則傳回 -1。
waitedcount	計數統計資訊	執行緒處於 WAITING 或 TIMED_WAITING 狀態的總次數。
lockname	字串統計資訊	監視鎖定的字串表示，表示執行緒被暫停而無法進入或執行緒正在透過 Object.wait 方法等待接收通知。
lockownerid	計數統計資訊	包含某個物件的監視鎖定的執行緒 ID，該執行緒在該物件上發生區段化。
lockownername	字串統計資訊	包含某個物件的監視所定的執行緒名稱，該執行緒在該物件上發生區段化。

表 16-23 J2SE 5.0 的 JVM 統計 — 執行緒資訊 (續)

統計資訊	資料類型	說明
stacktrace	字串統計資訊	與該執行緒相關聯的堆疊追蹤。

第 269 頁的「J2SE 5.0 中的 JVM 統計資訊」中列出了可用於 J2SE 5.0 的 JVM 中的執行緒的統計。

表 16-24 J2SE 5.0 的 JVM 統計資訊 - 執行緒

統計資訊	資料類型	說明
threadcount	計數統計資訊	目前的活性常駐程式執行緒和非常駐程式執行緒數。
peakthreadcount	計數統計資訊	JVM 啟動或峰重設以來的峰活性執行緒計數。
totalstartedthreadcount	計數統計資訊	JVM 啟動以來建立和/或啟動的執行緒總數。
daemonthreadcount	計數統計資訊	目前的活性常駐程式執行緒數。
allthreadids	字串統計資訊	所有活性執行緒 ID 清單。
currentthreadcputime	計數統計資訊	如果已啓用 CPU 時間測量，則表示目前執行緒的 CPU 時間 (以納秒為單位)。如果已停用 CPU 時間測量，則傳回 -1。
monitordeadlockedthreads	字串統計資訊	處於監視死結狀態的執行緒 ID 清單。

生產 Web 容器 (PWC) 統計資訊

可用於 Application Server Enterprise Edition (EE) 上的以下 PWC 元件和服務的統計：

- 第 273 頁的「生產 Web 容器 (PWC) 統計資訊」，PWC 虛擬伺服器
- 第 273 頁的「生產 Web 容器 (PWC) 統計資訊」，PWC 請求
- 第 273 頁的「生產 Web 容器 (PWC) 統計資訊」，PWC 檔案快取
- 第 273 頁的「生產 Web 容器 (PWC) 統計資訊」，PWC 持續作用
- 第 273 頁的「生產 Web 容器 (PWC) 統計資訊」，PWC DNS
- 第 177 頁的「有關偵聽程式和 JMX 連接器的管理主控台作業」，PWC 執行緒池
- 第 273 頁的「生產 Web 容器 (PWC) 統計資訊」，PWC 連線佇列
- 第 273 頁的「生產 Web 容器 (PWC) 統計資訊」，PWC HTTP 服務

第 273 頁的「生產 Web 容器 (PWC) 統計資訊」中列出了 PWC 虛擬伺服器的統計。

表 16-25 PWC 虛擬伺服器統計資訊 (僅限於 EE)

屬性名稱	資料類型	說明
id	字串統計資訊	虛擬伺服器的 ID。
mode	字串統計資訊	虛擬伺服器所處的模式。選項包括 <code>unknown</code> 或 <code>active</code> 。
hosts	字串統計資訊	由該虛擬伺服器提供服務的主機的名稱。
interfaces	字串統計資訊	配置了虛擬伺服器的介面 (偵聽程式) 的類型。

下表中列出了可用於 PWC 請求的統計。

表 16-26 PWC 請求統計 (僅限於 EE)

屬性名稱	資料類型	說明
method	字串統計資訊	用於請求的方法。
uri	字串統計資訊	上一個被處理的 URI。
countrequests	計數統計資訊	已處理的請求數。
countbytestransmitted	計數統計資訊	傳輸的位元組數，如果此資訊不可用，則值為 0。
countbytesreceived	計數統計資訊	接收的位元組數，如果此資訊不可用，則值為 0。
ratebytesreceived	計數統計資訊	在某段伺服器定義的時間間隔內的資料接收速率，如果此資訊不可用，則值為 0。
maxbytestransmissionrate	計數統計資訊	資料在某段伺服器定義的時間間隔內的最大傳輸速率，如果此資訊不可用，則值為 0。
countopenconnections	計數統計資訊	目前開啓的連線數，如果此資訊不可用，則值為 0。
maxopenconnections	計數統計資訊	同時開啓的連線的最大數目，如果此資訊不可用，則值為 0。
count2xx	計數統計資訊	代碼為 2XX 的回應的總數。
count3xx	計數統計資訊	代碼為 3XX 的回應的總數。
count4xx	計數統計資訊	代碼為 4XX 的回應的總數。
count5xx	計數統計資訊	代碼為 5XX 的回應的總數。
countother	計數統計資訊	具有其他回應代碼的回應總數。

表 16-26 PWC 請求統計 (僅限於 EE) (續)

屬性名稱	資料類型	說明
count200	計數統計資訊	代碼為 200 的回應的總數。
count302	計數統計資訊	代碼為 302 的回應的總數。
count304	計數統計資訊	代碼為 304 的回應的總數。
count400	計數統計資訊	代碼為 400 的回應的總數。
count401	計數統計資訊	代碼為 401 的回應的總數。
count403	計數統計資訊	代碼為 403 的回應的總數。
count404	計數統計資訊	代碼為 404 的回應的總數。
count503	計數統計資訊	代碼為 503 的回應的總數。

快取資訊小節提供了有關目前檔案快取的使用方式的資訊。下表列出了 PWC 檔案快取的統計。

表 16-27 PWC 檔案快取統計資訊 (僅限於 EE)

屬性名稱	資料類型	說明
flagenabled	計數統計資訊	指示是否啓用了檔案快取。禁用時有效值為 0，啓用時有效值為 1。
secondsmaxage	計數統計資訊	有效快取項目的最長存在時間 (以秒為單位)。
countentries	計數統計資訊	目前的快取項目數。單一快取項目表示單一 URI。
maxentries	計數統計資訊	同步式快取項目的最大數目。
countopenentries	計數統計資訊	與開啓的檔案關聯的項目數。
maxopenentries	計數統計資訊	與開啓的檔案關聯的同步式快取項目的最大數目。
sizeheapcache	計數統計資訊	用於快取內容的堆疊儲存區空間。
maxheapcachesize	計數統計資訊	用於快取檔案內容的最大堆疊儲存區空間。
sizemapcache	計數統計資訊	記憶體對映的檔案內容所使用的位址空間大小。
maxmmapcachesize	計數統計資訊	檔案快取用於記憶體對映檔案內容的最大位址空間大小。
counthits	計數統計資訊	快取查找成功的次數。
countmisses	計數統計資訊	快取查找失敗的次數。

表 16-27 PWC 檔案快取統計資訊 (僅限於 EE) (續)

屬性名稱	資料類型	說明
countinfohits	計數統計資訊	檔案資訊查找成功的次數。
countinfomisses	計數統計資訊	快取的檔案資訊遺失的數目。
countcontenthits	計數統計資訊	快取的檔案內容的命中次數。
countcontentmisses	計數統計資訊	檔案資訊查找失敗的次數。

本小節提供有關伺服器的 HTTP 層級持續作用的系統的資訊。下表中列出了可用於 PWC 持續作用的統計。

表 16-28 PWC 持續作用統計 (僅限於 EE)

屬性名稱	資料類型	說明
countconnections	計數統計資訊	處於持續作用模式的連線數。
maxconnections	計數統計資訊	允許同步處於持續作用模式的最大連線數。
counthits	計數統計資訊	處於持續作用模式的連線隨後進行了有效請求的總次數。
countflushes	計數統計資訊	伺服器關閉持續作用的連線的次數。
countrefusals	計數統計資訊	可能由於有太多的持續性連線而使伺服器無法將連線傳送到持續作用執行緒的次數。
counttimeouts	計數統計資訊	伺服器因用戶端連線逾時且沒有任何活動而終止持續作用連線的次數。
secondstimeout	計數統計資訊	關閉閒置持續作用連線之前經歷的時間 (以秒為單位)。

DNS 快取快取 IP 位址或 DNS 名稱。依預設，伺服器的 DNS 快取處於停用狀態。單一快取項目表示單一 IP 位址或 DNS 名稱查詢。下表中列出了可用於 PWC DNS 的統計。

表 16-29 PWC DNS 統計 (僅限於 EE)

屬性名稱	資料類型	說明
flagcacheenabled	計數統計資訊	指出 DNS 快取是否被啟用 (開啓)。禁用時為 0，啟用時為 1。
countcacheentries	計數統計資訊	目前位於快取中的 DNS 項目數。
maxcacheentries	計數統計資訊	快取中可容納的 DNS 項目的最大數目。

表 16-29 PWC DNS 統計 (僅限於 EE) (續)

屬性名稱	資料類型	說明
countcachehits	計數統計資訊	DNS 快取查找成功的次數。
countcachemisses	計數統計資訊	DNS 快取查找失敗的次數。
flagasyncenabled	計數統計資訊	指示是否啓用了非同步 DNS 查找。禁用時為 0，啓用時為 1。
countasyncnamelookups	計數統計資訊	非同步 DNS 名稱查找的總數。
countasynccaddrlookups	計數統計資訊	非同步 DNS 位址查找的總數。
countasynclookupsinprogress	計數統計資訊	正在進行的非同步查找的數目。

下表列出了 PWC 執行緒池的統計。

表 16-30 PWC 執行緒池統計資訊 (僅限於 EE)

屬性名稱	資料類型	說明
id	字串統計資訊	執行緒池 ID。
countthreadsidle	計數統計資訊	目前處於閒置狀態的請求處理執行緒數。
countthreads	計數統計資訊	目前的請求處理執行緒的數目。
maxthreads	計數統計資訊	可同時存在的請求處理執行緒的最大數目。
countqueued	計數統計資訊	佇列等候由此執行緒池進行處理的請求數。
peakqueued	計數統計資訊	佇列中同步容納的最大請求數。
maxqueued	計數統計資訊	佇列一次可容納的最大請求數。

連線佇列是請求被處理前保存這些請求的佇列。連線佇列的統計資訊顯示佇列中的階段作業數以及連線被接受前的平均延遲。下表列出了 PWC 連線佇列的統計。

表 16-31 PWC 連線佇列統計資訊 (僅限於 EE)

屬性名稱	資料類型	說明
id	字串統計資訊	連線佇列的 ID。
counttotalconnections	計數統計資訊	已接受的連線總數。
countqueued	計數統計資訊	目前位於佇列中的連線數。
peakqueued	計數統計資訊	佇列中同步容納的最大連線數。
maxqueued	計數統計資訊	連線佇列的最大大小。

表 16-31 PWC 連線佇列統計資訊 (僅限於 EE) (續)

屬性名稱	資料類型	說明
countoverflows	計數統計資訊	佇列太滿而無法容納連線的次數。
counttotalqueued	計數統計資訊	已佇列的連線總數。某個給定連線可能會被多次加入佇列，因此 counttotalqueued 可能大於或等於 counttotalconnections。
tickstotalqueued	計數統計資訊	連線在佇列中所花費的週期總數。週期是由系統決定的時間單位。
countqueued1minuteaverage	計數統計資訊	前 1 分鐘內處於佇列狀態的平均連線數。
countqueued5minuteaverage	計數統計資訊	前 5 分鐘內處於佇列狀態的平均連線數。
countqueued15minuteaverage	計數統計資訊	前 15 分鐘內處於佇列狀態的平均連線數。

下表列出了 PWC HTTP 服務的統計。

表 16-32 PWC HTTP 服務統計資訊 (僅限於 EE)

屬性名稱	資料類型	說明
id	字串統計資訊	HTTP 服務的實例名稱。
versionserver	字串統計資訊	HTTP 服務的版本編號。
timestarted	字串統計資訊	啟動 HTTP 服務的時間 (GMT)。
secondsrunning	計數統計資訊	HTTP 服務啟動以來所經歷的時間 (以秒為單位)。
maxthreads	計數統計資訊	每個實例中的最大工作者執行緒數。
maxvirtualservers	計數統計資訊	每個實例中可以配置的最大虛擬伺服器數目。
flagprofilingenabled	計數統計資訊	是否啓用了 HTTP 服務效能設定檔。有效值為 0 或 1。
flagvirtualserveroverflow	計數統計資訊	表示是否配置了多於 maxvirtualservers 的虛擬伺服器。如果設定為 1，則不會為所有的虛擬伺服器追蹤統計資料。
load1minuteaverage	計數統計資訊	前 1 分鐘內請求的平均負載。
load5minuteaverage	計數統計資訊	前 5 分鐘內請求的平均負載。
load15minuteaverage	計數統計資訊	前 15 分鐘內請求的平均負載。

表 16-32 PWC HTTP 服務統計資訊 (僅限於 EE) (續)

屬性名稱	資料類型	說明
ratebytestransmitted	計數統計資訊	在某段伺服器定義的時間間隔內資料的傳輸速率。如果此資訊不可用，則結果為 0。
ratebytesreceived	計數統計資訊	在某段伺服器定義的時間間隔內資料的接收速率。如果此資訊不可用，則結果為 0。

有關啓用和停用監視功能的管理主控台作業

- 第 279 頁的「使用管理主控台配置監視層級」
- 第 280 頁的「使用 asadmin 工具配置監視功能」

▼ 使用管理主控台配置監視層級

1 存取 [監視服務] 頁面。若要完成此操作，請執行以下步驟：

- 在樹形元件中，展開 [配置] 節點。
- 在樹狀結構中，展開要配置為用於監視功能之伺服器實例的節點，例如 server-config。
- 在樹中，選取 [監視]。

2 在 [監視服務] 頁面中，從要變更其監視層級的元件或服務相對的組合方塊中選擇適當的值。

依預設，所有元件和服務的監視功能均處於關閉狀態。若要啓用監視功能，請在組合方塊中選取 [低] 或 [高]。若要關閉監視功能，請在組合方塊中選取 [關閉]。可以開啓或關閉以下元件和服務的監視功能：

- **JVM** - 將此選項的監視層級設定為 [低]，可監視 Java 虛擬機器。
- **HTTP 服務** - 將此選項的監視層級設定為 [低]，可監視所有 HTTP 偵聽程式和虛擬伺服器。
- **作業事件服務** - 將此選項的監視層級設定為 [低]，可監視任意作業事件子系統。
- **JMS/連接器服務** - 將此選項的監視層級設定為 [低]，可監視任意 Java 訊息服務 (JMS)。
- **ORB** - 將此選項的監視層級設定為 [低]，可監視由 Application Server 核心及其連線管理員使用的系統 ORB。
- **Web 容器** - 將此選項的監視層級設定為 [低]，可監視所有已部署的 Servlet。
- **EJB 容器** - 將此選項的監視層級設定為 [低]，可監視所有已部署的 EJB 元件、EJB 池和 EJB 快取。將此方法設定為 [高] 還可以監視 EJB 商業方法。

- **JDBC 連線池** - 將此選項的監視層級設定為 [低]，可監視所有 JDBC 連線池。
- **執行緒池** - 將此選項的監視層級設定為 [低]，可監視所有執行緒池。

3 按一下 [儲存]。

此版本中沒有 [附加監視服務特性]，因此請忽略 [附加特性] 表。

更多資訊 等效的 asadmin 指令

set

例如，若要開啓對 HTTP 服務的監視功能，請使用以下 asadmin 指令：

```
asadmin> set --user admin-user
server.monitoring-service.module-monitoring-levels.http-service=LOW
```

▼ 使用 asadmin 工具配置監視功能

若要關閉監視功能，或設定監視元件或服務的層級，您可以依循第 279 頁的「[使用管理主控台配置監視層級](#)」中的說明使用管理主控台。

1 使用 get 指令可以查找目前已對哪些服務和元件啓用了監視功能：

```
asadmin> get --user admin-user server.monitoring-service.module-monitoring-levels.*
```

傳回：

```
server.monitoring-service.module-monitoring-levels.
connector-connection-pool = OFF
server.monitoring-service.module-monitoring-levels.
connector-service = OFF
server.monitoring-service.module-monitoring-levels.ejb-container = OFF
server.monitoring-service.module-monitoring-levels.http-service = OFF
server.monitoring-service.module-monitoring-levels.jdbc-connection-pool = OFF
server.monitoring-service.module-monitoring-levels.jms-service = OFF
server.monitoring-service.module-monitoring-levels.jvm = OFF
server.monitoring-service.module-monitoring-levels.orb = OFF
server.monitoring-service.module-monitoring-levels.thread-pool = OFF
server.monitoring-service.module-monitoring-levels.transaction-service = OFF
server.monitoring-service.module-monitoring-levels.web-container = OFF
```

2 使用 set 指令可以啓用監視功能。

例如，若要啓用對 HTTP 服務的監視功能，請輸入以下指令：

```
asadmin> set --user admin-user
server.monitoring-service.module-monitoring-levels.http-service=LOW
```

若要停用監視功能，請使用 set 指令並將監視層級指定為 OFF。

有關檢視監視資料的管理主控台作業

- 第 281 頁的「在管理主控台中檢視監視資料」
- 第 283 頁的「使用 asadmin 工具檢視監視資料」

▼ 在管理主控台中檢視監視資料

如需有關每個元件或服務之屬性的更多資訊，請參閱第 259 頁的「關於受監視的元件和服務的統計資訊」。

- 1 存取 [監視] 頁面。若要執行此作業，請：
 - a. 在樹形元件中，展開 [獨立實例] 節點。
 - b. 從清單中選取獨立伺服器實例，例如同伺服器 (Admin Server)。
 - c. 選取 [監視] 頁面。
 - d. 在 [監視] 頁面上選取 [監視] 標籤。
- 2 在 [檢視] 清單中，選取已部署到伺服器實例並已為其啓用了監視功能的元件或服務。所選元件或服務的監視資料顯示在 [檢視] 欄位下。如需有關可監視特性的說明，請參閱第 259 頁的「關於受監視的元件和服務的統計資訊」。

如果已針對 JVM、伺服器、執行緒池、HTTP 服務和作業事件服務啓用監視功能，則可以在此頁面中檢視這些元件和服務的監視資料。第 256 頁的「關於可監視物件的樹狀結構」以圖解說明這些元件和服務的組織架構。
- 3 如果沒有在此清單中看到您要監視的元件或服務，請選取 [配置監視] 連結，以針對所選元件和服務啓用和停用監視功能。

選取 [關閉] 以停用對元件或服務的監視功能。選取 [低] 或 [高]，以針對元件或服務啓用監視功能。

如需有關啓用和停用監視功能的更多資訊，請參閱第 279 頁的「使用管理主控台配置監視層級」或第 280 頁的「使用 asadmin 工具配置監視功能」。
- 4 選取 [監視] 頁面的 [應用程式] 標籤，以檢視部署至伺服器實例並已啓用監視功能的應用程式元件的監視資料。從 [應用程式] 清單中選取應用程式。從 [元件] 清單中選取特定元件。

如果未顯示應用程式或元件的監視資料，請選取 [配置監視] 連結，以針對元件或服務啓用或停用監視功能。若要監視應用程式，請開啓執行這些應用程式的容器。例如，針對 Web 應用程式的 Web 容器和/或 EJB 應用程式的 EJB 容器選取 [低] 或 [高]。

如果未顯示應用程式的監視資料，則該應用程式很可能不存在或未執行。僅當應用程式存在、其監視功能已啟用且正在運行時，應用程式監視資料才可用。執行了應用程式後，將在監視註冊中註冊該應用程式並顯示其監視資料。

使用管理主控台監視遠端應用程式和實例。遠端實例必須正在執行並已設定配置才能執行此操作。

所選取元件的監視資料顯示在所選取元件下面。如需有關可監視特性的說明，請參閱第 259 頁的「關於受監視的元件和服務的統計資訊」。您可以在第 256 頁的「關於可監視物件的樹狀結構」中檢視說明這些元件和服務如何針對應用程式進行組織的圖解。

- 5 選取 */資源/* 頁面可檢視部署至伺服器實例並啟用監視功能的資源的監視資料。從 [檢視] 清單中選取資源。

如果未顯示您要檢視其監視資料的資源，請選取 [配置監視] 連結，以針對資源啟用或停用監視功能。

如果未顯示資源的監視資料，則該資源很可能不存在或未執行。僅當資源存在、已針對此資源啟用 [高] 層級的監視功能，並且正在執行此資源時，才能取得此資源的監視資料。例如，如果您已建立 JDBC 連接器服務，但使用該連接器服務的應用程式尚未從服務請求連接器，則表示該服務尚未建立。因而，不存在任何服務，也沒有任何可用的監視資料。JDBC 應用程式執行並從服務請求連接器之後，將在監視登錄中註冊該服務並顯示其監視資料。

所選元件或服務的監視資料顯示在 [檢視] 欄位下。如需有關可監視特性的說明，請參閱第 259 頁的「關於受監視的元件和服務的統計資訊」。您可以在第 256 頁的「關於可監視物件的樹狀結構」中檢視說明這些元件和服務如何針對資源進行組織的圖解。

- 6 選取 */作業事件/* 頁面可固定作業事件子系統，進而回復作業事件並確定固定時正在進行的作業事件。
- 7 若要針對作業事件服務啟用監視功能，請選取 [配置監視] 連結，並確定將 [作業事件服務] 設定為 [低]。

若要固定作業事件服務以便回復作業事件，請選取 [固定]。若要回復作業事件，請選取作業事件旁邊的核取方塊，並按一下 [回復]。

更多資訊 等效的 asadmin 指令

```
get --monitor
```

例如，若要檢視 JVM 的監視資料，請使用以下 asadmin 指令：

```
asadmin> get --user adminuser --monitor server.jvm.*
```

使用 `asadmin` 工具檢視監視資料

- 第 283 頁的「使用 `asadmin` 工具檢視監視資料」
- 第 284 頁的「瞭解和指定含點名稱」
- 第 285 頁的「`list` 和 `get` 指令的範例」
- 第 285 頁的「`list --user admin-user --monitor` 指令的範例」
- 第 286 頁的「`get --user admin-user --monitor` 指令的範例」
- 第 287 頁的「使用 `PetStore` 範例」
- 第 291 頁的「`list` 和 `get` 指令在所有層級上的預期輸出」

▼ 使用 `asadmin` 工具檢視監視資料

若要使用 `asadmin` 工具檢視監視資料，請使用 `asadmin list` 和 `asadmin get` 指令，後面跟著可監視物件的帶點名稱，方法如下：

- 1 若要檢視可監視物件的名稱，請使用 `asadmin list` 指令。

例如，若要檢視伺服器實例上已啟用監視功能的應用程式元件和子系統的清單，請在終端機視窗中鍵入以下指令：

```
asadmin> list --user adminuser --monitor server
```

上述指令將傳回已啟用監視功能的應用程式元件與子系統清單，例如：

```
server.resources
server.connector-service
server.orb
server.jms-service
server.jvm
server.applications
server.http-service
server.thread-pools
```

如需有關使用 `list` 指令的更多範例，請參閱第 285 頁的「`list` 和 `get` 指令的範例」。如需有關可與 `list` 指令配合使用之帶點名稱的更多資訊，請參閱第 284 頁的「瞭解和指定含點名稱」。

- 2 若要顯示已啟用監視功能的應用程式元件或子系統的監視統計，請使用 `asadmin get` 指令。

若要取得統計，請在終端機視窗中鍵入 `asadmin get` 指令，並指定在先前步驟中由 `list` 指令顯示的名稱。以下範例嘗試獲取某個特定物件的子系統的所有屬性：

```
asadmin> get --user adminuser --monitor server.jvm.*
```

該指令將傳回以下屬性和資料：

```
server.jvm.dotted-name = server.jvm
server.jvm.heapsize-current = 21241856
server.jvm.heapsize-description = Provides statistical information about
```

```
the JVM's memory heap size.
server.jvm.heapsize-highwatermark = 21241856
server.jvm.heapsize-lastsampletime = 1080232913938
server.jvm.heapsize-lowerbound = 0
server.jvm.heapsize-lowwatermark = 0
server.jvm.heapsize-name = JvmHeapSize
server.jvm.heapsize-starttime = 1080234457308
server.jvm.heapsize-unit = bytes
server.jvm.heapsize-upperbound = 518979584
server.jvm.uptime-count = 1080234457308
server.jvm.uptime-description = Provides the amount of time the JVM has
    been running.
server.jvm.uptime-lastsampletime = 1080234457308
server.jvm.uptime-name = JvmUpTime
server.jvm.uptime-starttime = 1080232913928
server.jvm.uptime-unit = milliseconds
```

如需有關使用 `get` 指令的更多範例，請參閱第 285 頁的「`list` 和 `get` 指令的範例」。如需有關可與 `get` 指令配合使用之帶點名稱的更多資訊，請參閱第 284 頁的「瞭解和指定含點名稱」。

瞭解和指定含點名稱

在 `asadmin list` 和 `get` 指令中，指定可監視物件的帶點名稱。所有子物件均使用點 (.) 字元做為分隔符來命名，因此這些子物件稱為**帶點名稱**。如果子節點為單一型，則僅需監視物件類型即可命名物件；否則，需要形式為 `type.name` 的名稱來命名物件。

例如，`http-service` 就是一種有效的可監視物件類型，並且為單一型。若要命名表示實例 `server` 之 `http-service` 的單一型子節點，則帶點名稱為：

```
server.http-service
```

另一個範例，`application` 為一種有效的可監視物件類型，但並非單一型。例如，若要命名表示應用程式 `Petstore` 的非單一型子節點，則帶點名稱為：

```
server.applications.petstore
```

含點名稱也可以命名可監視物件中的特定屬性。例如，`http-service` 具有名為 `bytesreceived-lastsampletime` 的可監視屬性。以下名稱可以命名 `bytesreceived` 屬性：

```
server.http-service.server.http-listener-1.
    bytesreceived-lastsampletime
```

管理員無需知道 `asadmin list` 和 `get` 指令的有效帶點名稱。使用 `list` 指令可以顯示可用的可監視物件，而使用帶有萬用字元參數的 `get` 指令可以檢查任意可監視物件的所有可用屬性。

使用具有帶點名稱的 `list` 和 `get` 指令的基本假設為：

- 使用任何具有帶點名稱且後面不帶有萬用字元(*)的 `list` 指令，得到的結果為目前節點的直接子節點。例如，`list --user adminuser --monitor server` 將列出屬於 `server` 節點的所有直接子節點。
- 使用任何具有帶點名稱且後面帶有 `.*` 形式的萬用字元的 `list` 指令，得到的結果為目前節點的子節點階層式樹狀結構。例如，`list --user adminuser --monitor server.applications.*` 將列出 `applications` 的所有子節點及其後續子節點等。
- 使用任何具有帶點名稱且前面或後面帶有 `*dottedname`、`dotted *name` 或 `dotted name *` 形式的萬用字元的 `list` 指令，得到的結果為符合常規表示式 (由提供的符合式樣建立) 的所有節點及其子節點。
- 使用後面帶有 `.*` 或 `*` 的 `get` 指令，得到的結果為屬於要符合的目前節點的一組屬性及其值。

如需更多資訊，請參閱第 291 頁的「`list` 和 `get` 指令在所有層級上的預期輸出」。

list 和 get 指令的範例

本小節包含下列主題：

- [第 285 頁的「`list --user admin-user --monitor` 指令的範例」](#)
- [第 286 頁的「`get --user admin-user --monitor` 指令的範例」](#)

list --user admin-user --monitor 指令的範例

`list` 指令可針對指定伺服器實例名稱，提供有關目前正在監視之應用程式元件與子系統的資訊。使用此指令，您可以查看伺服器實例的可監視元件與子元件。如需 `list` 範例的更完整清單，請參閱第 291 頁的「`list` 和 `get` 指令在所有層級上的預期輸出」。

範例 1

```
asadmin> list --user admin-user --monitor server
```

上述指令將傳回已啟用監視功能的應用程式元件與子系統清單，例如：

```
server.resources
server.orb
server.jvm
server.jms-service
server.connector-service
server.applications
server.http-service
server.thread-pools
```

還可以列示指定的伺服器實例中目前所監視的應用程式。這對於使用 `get` 指令從某應用程式中搜尋特定監視統計很有幫助。

範例 2

```
asadmin> list --user admin-user --monitor server.applications
```

傳回：

```
server.applications.adminapp
  server.applications.admingui
server.applications.myApp
```

如需更完整的範例，請參閱第 287 頁的「使用 [PetStore 範例](#)」。

get --user admin-user --monitor 指令的範例

此指令可擷取以下受監視的資訊：

- 一個元件或子系統內的全部受監視屬性
- 一個元件或子系統內特定的受監視屬性

當特定元件或子系統所請求的屬性不存在時，將會傳回錯誤。同樣，當元件或子系統所請求的特定屬性不在作用中時，也會傳回錯誤。

請參閱第 291 頁的「[list 和 get 指令在所有層級上的預期輸出](#)」，以取得有關使用 `get` 指令的更多資訊。

範例 1

嘗試從某子系統中取得特定物件的所有屬性：

```
asadmin> get --user admin-user --monitor server.jvm.*
```

傳回：

```
server.jvm.dotted-name= server.jvm
server.jvm.heapsize-current = 21241856
server.jvm.heapsize-description = Provides statistical information about
  the JVM's memory heap size.
server.jvm.heapsize-highwatermark = 21241856
server.jvm.heapsize-lastsampletime = 1080232913938
server.jvm.heapsize-lowerbound = 0
server.jvm.heapsize-lowwatermark = 0
server.jvm.heapsize-name = JvmHeapSize
server.jvm.heapsize-starttime = 1080234457308
server.jvm.heapsize-unit = bytes
server.jvm.heapsize-upperbound = 518979584
server.jvm.uptime-count = 1080234457308
server.jvm.uptime-description = Provides the amount of time the JVM has
  been running.
server.jvm.uptime-lastsampletime = 1080234457308
```

```
server.jvm.uptime-name = JvmUpTime
server.jvm.uptime-starttime = 1080232913928
server.jvm.uptime-unit = milliseconds
```

範例 2

嘗試從 J2EE 應用程式中取得所有屬性：

```
asadmin> get --user admin-user --monitor server.applications.myJ2eeApp.*
```

傳回：

```
No matches resulted from the wildcard expression.
CLI137 Command get failed.
```

該 J2EE 應用程式層級上沒有提供可監視的屬性，因此顯示此回覆。

範例 3

嘗試從某子系統中取得特定屬性：

```
asadmin> get --user admin-user --monitor server.jvm.uptime-lastsampletime
```

傳回：

```
server.jvm.uptime-lastsampletime = 1093215374813
```

範例 4

嘗試從某子系統屬性中取得不明的屬性：

```
asadmin> get --user admin-user --monitor server.jvm.badname
```

傳回：

```
No such attribute found from reflecting the corresponding Stats
interface: [badname]
CLI137 Command get failed.
```

▼ 使用 PetStore 範例

以下範例說明如何將 `asadmin` 工具用於監視目的。

使用者要檢查將範例 PetStore 應用程式部署至 Application Server 之後，在該應用程式中呼叫某個方法的次數。已部署該應用程式的實例名稱爲 `server`。結合 `list` 與 `get` 指令，即可存取所需的方法統計。

1 啓動 Application Server 和 `asadmin` 工具。

- 2 設定一些有用的環境變數，以避免為每個指令均輸入這些變數：

```
asadmin> export AS_ADMIN_USER=admin AS_ADMIN_PASSWORD=admin123
asadmin>export AS_ADMIN_HOST=localhost AS_ADMIN_PORT=4849
```

- 3 列出實例 server 的可監視元件：

```
asadmin> list --user adminuser --monitor server*
```

傳回 (輸出將與以下內容類似)：

```
server
server.applications
server.applications.CometEJB
server.applications.ConverterApp
server.applications.petstore
server.http-service
server.resources
server.thread-pools
```

可監視元件清單包括 thread-pools、http-service、resources 以及所有已部署 (與已啟用) 的 applications。

- 4 列出 PetStore 應用程式中的可監視子元件 (可以使用 -m 替代 --monitor)：

```
asadmin> list -m server.applications.petstore
```

傳回：

```
server.applications.petstore.signon-ejb_jar
server.applications.petstore.catalog-ejb_jar
server.applications.petstore.uidgen-ejb_jar
server.applications.petstore.customer-ejb_jar
server.applications.petstore.petstore-ejb_jar
server.applications.petstore.petstore\war
server.applications.petstore.AsyncSenderJAR_jar
server.applications.petstore.cart-ejb_jar
```

- 5 列出 PetStore 應用程式之 EJB 模組 signon-ejb_jar 中的可監視子元件：

```
asadmin> list -m server.applications.petstore.signon-ejb_jar
```

傳回：

```
server.applications.petstore.signon-ejb_jar.SignOnEJB
server.applications.petstore.signon-ejb_jar.UserEJB
```

- 6 列出 PetStore 應用程式之 EJB 模組 signon-ejb_jar 的實體 Bean UserEJB 中的可監視子元件：

```
asadmin> list -m server.applications.petstore.signon-ejb_jar.UserEJB
```


傳回 (為節省空間而移除帶點名稱) :

```
server.applications.petstore.signon-ejb_jar.UserEJB.bean-cache
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods
server.applications.petstore.signon-ejb_jar.UserEJB.bean-pool
```

- 7 列出實體 Bean UserEJB (位於 PetStore 應用程式的 EJB 模組 signon-ejb_jar 中) 之 getUsername 方法中的可監視子元件 :

```
asadmin> list -m
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.getUsername
```

傳回 :

```
Nothing to list at server.applications.petstore.signon-ejb_jar.
UserEJB.bean-methods.getUsername. To get the valid names beginning with a
string, use the wildcard "*" character. For example, to list all names
that begin with "server", use "list server*".
```

- 8 該方法沒有可監視的子元件。取得 getUsername 方法的所有可監視統計。

```
asadmin> get -m
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.getUsername.*
```

傳回 :

```
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.
getUsername.executiontime-count = 0
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.
getUsername.executiontime-description = Provides the time in milliseconds
    spent during the last successful/unsuccessful attempt to execute the
    operation.
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.
getUsername.executiontime-lastsampletime = 1079981809259
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.
getUsername.executiontime-name = ExecutionTime
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.
getUsername.executiontime-starttime = 1079980593137
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.
getUsername.executiontime-unit = count
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.
getUsername.methodstatistic-count = 0
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.
getUsername.methodstatistic-description = Provides the number of times an
    operation was called, the total time that was spent during the
    invocation and so on.
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.
getUsername.methodstatistic-lastsampletime = 1079980593137
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.
getUsername.methodstatistic-maxtime = 0
```

```
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.  
getUserName.methodstatistic-mintime = 0  
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.  
getUserName.methodstatistic-name = ExecutionTime  
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.  
getUserName.methodstatistic-starttime = 1079980593137  
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.  
getUserName.methodstatistic-totaltime = 0  
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.  
getUserName.methodstatistic-unit =  
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.  
getUserName.totalnumerrors-count = 0  
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.  
getUserName.totalnumerrors-description = Provides the total number of errors  
that occurred during invocation or execution of an operation.  
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.  
getUserName.totalnumerrors-lastsampletime = 1079981809273  
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.  
getUserName.totalnumerrors-name = TotalNumErrors  
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.  
getUserName.totalnumerrors-starttime = 1079980593137  
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.  
getUserName.totalnumerrors-unit = count  
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.  
getUserName.totalnumsuccess-count = 0  
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.  
getUserName.totalnumsuccess-description = Provides the total number of  
successful invocations of the method.  
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.  
getUserName.totalnumsuccess-lastsampletime = 1079981809255  
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.  
getUserName.totalnumsuccess-name = TotalNumSuccess  
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.  
getUserName.totalnumsuccess-starttime = 1079980593137  
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.  
getUserName.totalnumsuccess-unit = count
```

9 如果還需要取得執行時間等特定統計，請使用如下指令：

```
asadmin> get -m server.applications.petstore.signon-ejb_jar.  
UserEJB.bean-methods.getUserName.executiontime-count
```

傳回：

```
server.applications.petstore.signon-ejb_jar.UserEJB.bean-methods.  
getUserName.executiontime-count = 1
```

list 和 get 指令在所有層級上的預期輸出

下表顯示了樹的各個層級的指令、含點名稱及相應的輸出。

表 16-33 頂層

指令	含點名稱	輸出
list -m	server	server.applicationserver.thread-poolserver. resourceserver.http-serviceserver.transaction- serviceserver.orb.connection-managerserver.orb. connection-managers.orb\.Connections\.Inbound\ AcceptedConnectionsserver.jvm
list -m	server.*	此節點下的子節點的階層結構。
get -m	server.*	無輸出，但顯示一條訊息，說明此節點上沒有屬性。

下表列出了應用程式層級的指令、帶點名稱及對應的輸出。

表 16-34 應用程式層級

指令	含點名稱	輸出
list -m	server.applications 或 *applications	appllapp2web-module1_warejb-module2_jar...
list -m	server.applications.* 或 *applications.*	此節點下的子節點的階層結構。
get -m	server.applications.* 或 *applications.*	無輸出，但顯示一條訊息，說明此節點上沒有屬性。

下表列出了應用程式層級上的獨立模組和企業應用程式的指令、帶點名稱及對應的輸出。

表 16-35 應用程式 - 企業應用程式和獨立模組

指令	查點名稱	輸出
list -m	server.applications.app1 或 *app1 備註：僅在已部署企業應用程式之後，此層級才可用。如果部署了獨立模組，則此層級不可用。	ejb-module1_jarweb-module2_warejb-module3_jarweb-module3_war...
list -m	server.applications.app1.* 或 *app1.*	此節點下的子節點的階層結構。
get -m	server.applications.app1.* 或 *app1.*	無輸出，但顯示一條訊息，說明此節點上沒有屬性。
list -m	server.applications.app1.ejb-module1_jar 或 *ejb-module1_jar 或 server.applications.ejb-module1_jar	bean1bean2bean3...
list -m	server.applications.app1.ejb-module1_jar 或 *ejb-module1_jar 或 server.applications.ejb-module1_jar	此節點下的子節點的階層結構。
get -m	server.applications.app1.ejb-module1_jar.* 或 *ejb-module1_jar.* 或 server.applications.ejb-module1_jar.*	無輸出，但顯示一條訊息，說明此節點上沒有屬性。
list -m	server.applications.app1.ejb-module1_jar.bean 備註：在獨立模組中，不會顯示包含應用程式名稱 (此範例中為 app1) 的節點。	子節點清單： bean-poolbean-cachebean-method

表 16-35 應用程式 - 企業應用程式和獨立模組 (續)

指令	端點名稱	輸出
list -m	server.applications.app1.ejb-module1_jar.bean 備註：在獨立模組中，不會顯示包含應用程式名稱 (此範例中為 app1) 的節點。	子節點的階層結構及該節點和所有後續子節點的所有屬性的清單。
get -m	server.applications.app1.ejb-module1_jar.bean 備註：在獨立模組中，不會顯示包含應用程式名稱 (此範例中為 app1) 的節點。	以下屬性及其關聯值： CreateCount_CountCreateCount_ DescriptionCreateCount_ LastSampleTimeCreateCount_ NameCreateCount_ StartTimeCreateCount_ UnitMethodReadyCount_ CurrentMethodReadyCount_ DescriptionMethodReadyCount_ HighWaterMarkMethodReadyCount_ LastSampleTimeMethodReadyCount_ LowWaterMarkMethodReadyCount_ NameMethodReadyCount_ StartTimeMethodReadyCount_ UnitRemoveCount_CountRemoveCount_ DescriptionRemoveCount_ LastSampleTimeRemoveCount_ NameRemoveCount_StartTimeAttribute RemoveCount_Unit
list -m	server.applications.app1.ejb-module1_jar.bean 備註：在獨立模組中，不會顯示包含應用程式名稱 (此範例中為 app1) 的節點。	無屬性，但顯示一條訊息，說明：server.applications.app1.ejb-module1_jar.bean1-cache 上沒有要列出的內容。若要取得以字串開始的有效名稱，請使用萬用字元 (*)。例如，若要列出以 server 開始的所有名稱，請使用 list server*。
get -m	server.applications.app1.ejb-module1_jar.bean 備註：在獨立模組中，不會顯示包含應用程式名稱 (此範例中為 app1) 的節點。	與 J2EE 池屬性對應的屬性和值清單，如表 1-4 中所示。
list -m	server.applications.app1.ejb-module1_jar.bean 備註：在獨立模組中，不會顯示包含應用程式名稱 (此範例中為 app1) 的節點。	無屬性，但顯示一條訊息，說明「使用帶有 --monitor 選項的 get 指令可以檢視此節點的屬性和值。」
get -m	server.applications.app1.ejb-module1_jar.bean 備註：在獨立模組中，不會顯示包含應用程式名稱 (此範例中為 app1) 的節點。	與 J2EE 快取記憶體屬性對應的屬性和值清單，如表 1-5 中所示。
list -m	server.applications.app1.ejb-module1_jar.bean 備註：在獨立模組中，不會顯示包含應用程式名稱 (此範例中為 app1) 的節點。	無屬性，但顯示一條訊息，說明「使用帶有 --monitor 選項的 get 指令可以檢視此節點的屬性和值。」

表 16-35 應用程式 - 企業應用程式和獨立模組 (續)

指令	倉點名稱	輸出
get -m	server.applications.app1.ejb-module1_jar.bean 備註：在獨立模組中，不會顯示包含應用程式名稱 (此範例中為 app1) 的節點。	與 EJB 方法屬性對應的屬性和值清單，如表 1-2 中所示。
list -m	server.applications.app1.web-module1_war	顯示指定給模組的虛擬伺服器。
get -m	server.applications.app1.web-module1_war.*	無輸出，但顯示一條訊息，說明此節點上沒有屬性。
list -m	server.applications.app1.web-module1_war.virtual-server	顯示已註冊的 Servlet 清單。
get -m	server.applications.app1.web-module1_war.virtual-server	無輸出，但顯示一條訊息，說明此節點上沒有屬性。
list -m	server.applications.app1.web-module1_war.virtual-server Servlet	無屬性，但顯示一條訊息，說明「使用帶有 --monitor 選項的 get 指令可以檢視此節點的屬性和值。」
get -m	server.applications.app1.web-module1_war.virtual-server Servlet	與 Web 容器 (Servlet) 屬性對應的屬性和值清單，如表 1-7 中所示。

下表列出了 HTTP 服務層級的指令、帶點名稱及對應的輸出。

表 16-36 HTTP 服務層級

指令	倉點名稱	輸出
list -m	server.http-service	虛擬伺服器清單。
get -m	server.http-service.*	無輸出，但顯示一條訊息，說明此節點上沒有屬性。
list -m	server.http-service.server	HTTP 偵聽程式清單。
get -m	server.http-service.server.*	無輸出，但顯示一條訊息，說明此節點上沒有屬性。
list -m	server.http-service.server.http-listener	無屬性，但顯示一條訊息，說明「使用帶有 --monitor 選項的 get 指令可以檢視此節點的屬性和值。」
get -m	server.http-service.server.*	與 HTTP 服務屬性對應的屬性和值清單，如表 1-9 中所示。

下表列出了執行緒池層級的指令、帶點名稱及對應的輸出。

表 16-37 執行緒池層級

指令	倉點名稱	輸出
list -m	server.thread-pools	執行緒池名稱清單。
get -m	server.thread-pools.*	無輸出，但顯示一條訊息，說明此節點上沒有屬性。
list -m	server.thread-pools.orb\threadpool\thread-pool	無屬性，但顯示一條訊息，說明「使用帶有 --monitor 選項的 get 指令可以檢視此節點的屬性和值。」
get -m	server.thread-pools..orb\threadpool\thread-pool	與執行緒池屬性對應的屬性和值清單，如表 1-14 中所示。

下表列出了資源層級的指令、帶點名稱及對應的輸出。

表 16-38 資源層級

指令	倉點名稱	輸出
list -m	server.resources	池名稱清單。
get -m	server.resources.*	無輸出，但顯示一條訊息，說明此節點上沒有屬性。
list -m	server.resources.jdbc-connection-pool-pool.connection-pool	無屬性，但顯示一條訊息，說明「使用帶有 --monitor 選項的 get 指令可以檢視此節點的屬性和值。」
get -m	server.resources.jdbc-connection-pool-pool.connection-pool	與連線池屬性對應的屬性和值清單，如表 1-10 中所示。

下表列出了作業事件服務層級的指令、帶點名稱及對應的輸出。

表 16-39 作業事件服務層級

指令	倉點名稱	輸出
list -m	server.transaction-service	無屬性，但顯示一條訊息，說明「使用帶有 --monitor 選項的 get 指令可以檢視此節點的屬性和值。」
get -m	server.transaction-service.*	與作業事件服務屬性對應的屬性和值清單，如表 1-15 中所示。

下表列出了 ORB 層級的指令、帶點名稱及對應的輸出。

表 16-40 ORB 層級

指令	倉點名稱	輸出
list -m	server.orb	server-orb.connection-managers
get -m	server.orb.*	無輸出，但顯示一條訊息，說明此節點上沒有屬性。
list -m	server.orb.connection-managers	ORB 連線管理員的名稱。
get -m	server.orb.connection-managers.*	無輸出，但顯示一條訊息，說明此節點上沒有屬性。
list -m	server.orb.connection-managers.orb\ConnectionManager	無屬性，但顯示一條訊息，說明「使用帶有 --monitor 選項的 get 指令可以檢視此節點的屬性和值。」
get -m	server.orb.connection-managers.orb\ConnectionManager	與 ORB 連線管理員屬性對應的屬性和值清單，如表 1-13 中所示。

下表列出了 JVM 層級的指令、帶點名稱及對應的輸出。

表 16-41 JVM 層級

指令	倉點名稱	輸出
list -m	server.jvm	無屬性，但顯示一條訊息，說明「使用帶有 --monitor 選項的 get 指令可以檢視此節點的屬性和值。」
get -m	server.jvm.*	與 JVM 屬性對應的屬性和值清單，如表 1-16 中所示。

使用 JConsole

若要使 JConsole 可以與 Application Server 配合使用，必須停用 JMX 連接器的安全性。依預設，Application Server 的目前版本 (SE/EE 版本) 已啟用安全性。

▼ 停用 JMX 連接器的安全性

- 1 使用 管理主控台 停用 JMX 連接器的安全性。若要從 管理主控台 執行此作業，請：
 - a. 展開 [節點代理程式]。
 - b. 選取 [節點代理程式]。

- c. 取消選取 [JMX] 標籤中的 [SSL3] 和 [TLS]。
 - d. 選取 [儲存]。
- 2 使用 `asadmin` 停用 JMX 連接器的安全性。若要從終端機視窗或指令提示符號執行此操作，請執行以下步驟：
 - a. 輸入以下指令：

```
asadmin set server.admin-service.jmx-connector.system.security-enabled=false
```
 - b. 重新啟動網域應用程式伺服器 (DAS)。

針對 Platform Edition 版本，則 JMX 連接器依預設會處於停用狀態，因此無需變更 Platform Edition 的任何配置。
- 3 啟動 JConsole，然後在 [進階] 標籤中輸入 JMX URL、使用者名稱和密碼以登入。JMX URL 的格式為：

```
service:jmx:rmi:///jndi/rmi://<您的機器名稱>:<連接埠>/management/rmi-jmx-connector
```

備註：如果您搜尋 message ADM1501，便可從管理 `server.log` 檔案中取得確切的 JMX URL。

Java 虛擬機器和進階設定

本章說明如何配置 Java 虛擬機器 (JVM™) 和其他進階設定。它包含以下小節：

- 第 299 頁的「有關 JVM 設定的管理主控台作業」
- 第 303 頁的「有關進階設定的管理主控台作業」

有關 JVM 設定的管理主控台作業

- 第 299 頁的「配置 JVM 一般設定」
- 第 300 頁的「配置 JVM 類別路徑設定」
- 第 301 頁的「配置 JVM 選項」
- 第 302 頁的「停用安全性管理員」
- 第 302 頁的「配置 JVM 效能評測器設定」

▼ 配置 JVM 一般設定

Java 虛擬機器 (JVM) 包含在 Java 2 Standard Edition (J2SE™) 軟體中，是 Application Server 所需的軟體。由於不正確的 JVM 設定將導致伺服器停止執行，因此您應當謹慎地變更這些設定。

- 1 在樹形元件中，選取 [配置]。
- 2 在目標伺服器實例中按一下 [JVM 設定] 節點。
- 3 依預設，標籤下的 [一般] 連結已處於選取狀態。

- 4 在 [JVM 一般設定] 頁面中，您可以指定以下內容：
 - a. 在 [Java 首頁] 欄位中，輸入 **Java 2 Standard Edition (J2SE)** 軟體的安裝目錄名稱。
Application Server 依賴於 J2SE 軟體執行。若要驗證此版本是否支援您所指定的 J2SE 版本，請參閱「版本說明」。(請參閱第 54 頁的「詳細資訊」小節中的連結。)

備註 – 如果輸入不存在的目錄名稱或輸入不受支援的 J2SE 軟體版本的安裝目錄名稱，則 Application Server 將無法啟動。

 - b. 在 [Javac] 欄位中，鍵入 **Java 程式設計語言編譯器** 的指令行選項。
部署 EJB 元件後，Application Server 將執行編譯器。
 - c. 若要通過 **JPDA (Java 平台除錯程式架構)** 來設定除錯，請選取 [啟用除錯] 核取方塊並在 [除錯選項] 欄位中指定選項。
JPDA 由應用程式開發者使用。如需更多資訊，請參閱「Application Server Developer's Guide」中的「Debugging J2EE Applications」一章。(如需指向此指南的連結，請參閱第 54 頁的「詳細資訊」。)
 - d. 在 [RMI 編譯選項] 欄位中，鍵入 **rmic 編譯器** 的指令行選項。
部署 EJB 元件後，Application Server 將執行 **rmic** 編譯器。
 - e. 在 [位元碼預處理程式] 欄位中，鍵入以逗號分隔的類別名稱清單。
每個類別都必須實作 `com.sun.appserv.BytecodePreprocessor` 介面。將按指定次序呼叫這些類別。
效能評測器等工具也許需要 [位元碼預處理程式] 欄位中的項目。效能評測器產生用於分析伺服器效能的資訊。如需有關效能評測的更多資訊，請參閱「Application Server Developer's Guide」中的「Debugging J2EE Applications」一章。
- 5 按一下 [儲存]。
- 6 重新啟動伺服器。

▼ 配置 JVM 類別路徑設定

類別路徑就是 JAR 檔案的清單，Java 執行階段環境將在此清單中搜尋類別和其他資源檔案。

- 1 在樹形元件中，選取 [Application Server] 節點。
- 2 按一下 [JVM 設定] 標籤。

- 3 選取標籤下面的 [路徑設定] 連結。
- 4 在 [JVM 類別路徑設定] 頁面中，您可以指定以下內容：
 - a. 在 [環境類別路徑] 核取方塊中，保留預設選取以忽略 CLASSPATH 環境變數。
對於程式設計中的指導而言，CLASSPATH 環境變數使用起來很方便，但是不建議將此變數用於企業環境。
 - b. 若要檢視 Application Server 的類別路徑，請檢查 [伺服器類別路徑] 欄位中的唯讀內容。
 - c. 若要將 JAR 檔案插入伺服器類別路徑的開頭，請在 [類別路徑前綴] 欄位中輸入此檔案的完整路徑名稱。
 - d. 若要將 JAR 檔案增加至伺服器類別路徑的結尾，請在 [類別路徑後綴] 欄位中輸入此檔案的完整路徑名稱。
例如，假設您要指定資料庫驅動程式的 JAR 檔案。請參閱第 79 頁的「[整合 JDBC 驅動程式](#)」。
 - e. 在 [本機庫路徑字首] 和 [本機庫路徑字尾] 欄位中，您可以將項目附加到本機庫路徑的開頭或結尾。
本機程式庫路徑是伺服器的相對路徑 (用於其本機共用程式庫)、標準 JRE 本機程式庫路徑、Shell 環境設定 (UNIX 上的 LD_LIBRARY_PATH) 以及 [JVM 效能評測器設定] 頁面上指定的任何路徑的鏈結。
- 5 按一下 [儲存]。
- 6 重新啟動伺服器。

▼ 配置 JVM 選項

在 [JVM 選項] 頁面中，您可以為執行 Application Server 的 Java 應用程式啟動程式 (java 工具) 指定選項。-D 選項用於指定 Application Server 的專用特性。

- 1 在樹形元件中，選取 [Application Server] 節點。
- 2 按一下 [JVM 設定] 標籤。
- 3 選取標籤下面的 [JVM 選項] 連結。
- 4 在 [JVM 選項] 頁面中，通過編輯 [值] 欄位來修改選項。

- 5 若要增加選項，請執行以下步驟：
 - a. 按一下 [增加 JVM 選項]。
 - b. 在顯示的空白列中，在 [值] 欄位中鍵入資訊。
- 6 若要移除選項，請執行以下步驟：
 - a. 選取該選項旁邊的核取方塊。
 - b. 按一下 [刪除]。
- 7 按一下 [儲存]。
- 8 重新啟動伺服器。

如需有關 JVM 選項的更多資訊，請參閱
<http://java.sun.com/j2se/1.4.2/docs/tooldocs/tools.html> 和
<http://java.sun.com/docs/hotspot/VMOptions.html>。

▼ 停用安全性管理員

停用 Application Server 的安全性管理員可以改善某些類型的應用程式的效能。即使安全性管理程式已停用，J2EE 授權和驗證功能仍可以起作用。您可以在開發環境中停用安全性管理程式，但不應在生產環境中停用安全性管理程式。

- 1 請至管理主控台的 [JVM 選項] 頁面。

如需說明，請參閱第 301 頁的「配置 JVM 選項」。
- 2 在 [JVM 選項] 頁面中，移除以下選項：
-Djava.security.policy
- 3 按一下 [儲存]。
- 4 重新啟動伺服器。

▼ 配置 JVM 效能評測器設定

效能評測器工具可產生用於分析效能和識別潛在瓶頸的資料。

- 1 在樹形元件中，選取 [Application Server] 節點。
- 2 按一下 [JVM 設定] 標籤。

- 3 選取標籤下面的 [效能評測器] 連結。
- 4 您在 [JVM 效能評測器設定] 頁面中指定的資訊取決於您所使用的效能評測器產品。如需範例和說明，請參閱「Application Server Developer's Guide」中的「Debugging J2EE Applications」一章。(如需指向此指南的連結，請參閱第 54 頁的「詳細資訊」)。
- 5 按一下 [儲存]。
- 6 重新啟動伺服器。

有關進階設定的管理主控台作業

- 第 303 頁的「設定進階網域屬性」

▼ 設定進階網域屬性

- 1 在樹形元件中，展開 [獨立實例] 並選取伺服器實例節點。
- 2 選取 [進階] 標籤。
- 3 在 [網域屬性] 頁面中，您可以執行以下操作：
 - a. 在 [應用程式根] 欄位中，識別將在其中部署應用程式的完整目錄路徑。
 - b. 在 [記錄根] 欄位中，指定保存伺服器實例記錄檔的目錄。
 - c. 通常，您需要將 [語言環境] 欄位保留為空，以使用主機的預設語言環境。
語言環境是一個識別碼，用於指定特定的語言和區域組合。例如，美國英語的語言環境為 en_US，日語的語言環境為 ja_JP。若要使用非英語語言環境，Application Server 必須經過本土化，即必須將英語翻譯成其他語言。
- 4 按一下 [儲存]。
- 5 重新啟動伺服器。

編譯和配置 Apache Web Server

本附錄說明如何編譯 Apache 原始碼和配置 Apache Web Server 的安裝，以使用 Sun Java System Application Server 負載平衡器外掛程式。

請下載相應的 Apache 原始碼。如需有關 Application Server 支援的 Apache Web Server 之版本和平台的資訊，請參閱「Application Server 版本說明」。

本附錄包含下列主題：

- [第 305 頁的「安裝 Apache」](#)

安裝 Apache

本小節包含下列主題：

- [第 305 頁的「適用於 Apache 1.3 的最低需求」](#)
- [第 306 頁的「適用於 Apache 2 的最低需求」](#)
- [第 307 頁的「安裝可識別 SSL 的 Apache」](#)

適用於 Apache 1.3 的最低需求

本小節說明成功編譯 Apache 1.3 Web Server 以執行負載平衡器外掛程式的最低需求。必須編譯和建置 Apache 原始碼以使用 SSL 執行。

UNIX 和 Linux 平台的需求：

- openssl-0.9.7d (原始碼)
- mod_ssl-2.8.16-1.3.29 (原始碼)
- apache_1.3.29 (原始碼)
- gcc-3.3-sol9-sparc-local 套裝軟體 (適用於 Solaris 9 SPARC/x86)。
- flex-2.5.4a-sol9-sparc-local 套裝軟體 (適用於 Solaris 9 SPARC)
- flex-2.5.4a-sol9-intel-local 套裝軟體 (適用於 Solaris 9 SPARC)

此外，在編譯 Apache 之前，請執行以下步驟：

- 在 Linux 上，在同一台機器上安裝 Sun Java System Application Server。
- 在 Solaris 8 上，確保 gcc 和 make 均位於 PATH 中。
- 在 Solaris 9 上，確保 gcc 3.3 版和 make 均位於 PATH 中，並且已安裝 flex。
- 如果您要在 Red Hat Enterprise Linux Advanced Server 2.1 上使用 gcc，則 gcc 的版本必須在 3.0 以上。

備註-

- 要使用其他 C 編譯器，請設定 C 編譯器的路徑並使公用程式位於 PATH 環境變數中。例如：

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:install-dir/lib
```
 - 這些軟體原始碼可以從 <http://www.sunfreeware.com> 取得。
-

適用於 Apache 2 的最低需求

本小節說明成功編譯 Apache 2 Web Server 以執行負載平衡器外掛程式的最低需求。必須編譯和建置 Apache 原始碼以使用 SSL 執行。

UNIX 和 Linux 平台的需求：

- openssl-0.9.7e (原始碼)
- httpd-2.0.49 (原始碼)
- gcc-3.3-sol9-sparc-local 套裝軟體 (適用於 Solaris SPARC)。
- gcc-3.3-sol9-intel-local 套裝軟體 (適用於 Solaris x86)
- flex-2.5.4a-sol9-sparc-local 套裝軟體 (適用於 Solaris SPARC)
- flex-2.5.4a-sol9-intel-local 套裝軟體 (適用於 Solaris x86)

此外，在編譯 Apache 之前，請執行以下步驟：

- 在 Linux 平台上，在同一台機器上安裝 Sun Java System Application Server。
- 在 Solaris 作業系統上，確保 gcc 3.3 版和 make 均位於 PATH 中，並且已安裝 flex。
- 在 Solaris 10 作業系統上，對 OpenSSL 執行 make 之前，請先執行

```
/usr/local/lib/gcc-lib/sparc-sun-solaris2.9/3.3/install-tools/mkheaders
```


(Solaris SPARC) 或

```
/usr/local/lib/gcc-lib/i386-pc-solaris2.9/3.3/install-tools/mkheaders
```


(Solaris x86)。
- 如果您要在 Red Hat Enterprise Linux Advanced Server 2.1 上使用 gcc，則 gcc 的版本必須在 3.0 以上。

備註 -

- 要使用其他 C 編譯器，請設定 C 編譯器的路徑並使公用程式位於 PATH 環境變數中。例如：`export LD_LIBRARY_PATH=install-dir/lib:$LD_LIBRARY_PATH`。此範例是針對 sh 的。
 - 這些軟體原始碼可以從 <http://www.sunfreeware.com> 取得。
-

▼ 安裝可識別 SSL 的 Apache

開始之前 您必須已下載並解壓縮了 Apache 軟體。

- 1 編譯和建置 OpenSSL。如需有關 OpenSSL 的更多資訊，請參閱

<http://www.openssl.org/>。

如果隨 Linux 安裝的 OpenSSL 版本是 0.9.7.e，則在 Linux 平台上不需要執行此步驟。

下載和解壓縮 OpenSSL 原始碼。

a. `cd openssl-0.9.7e`

b. `make`

c. `make install`

- 2 對於 Apache 1.3，使用 `mod_ssl` 配置 Apache。對於 Apache 2，您無需完成此步驟。如需有關 `mod_ssl` 的更多資訊，請參閱 <http://www.modssl.org/>。

解壓縮 `mod_ssl` 原始碼並執行以下步驟。

a. `cd mod_ssl-2.8.14-1.3.x`

b. 執行 `./configure --with-apache=../apache_1.3.x --with-ssl=../openssl-0.9.7e --prefix=install-dir --enable-module=ssl --enable-shared=ssl --enable-rule=SHARED_CORE --enable-module=so`

上述指令範例中指定的目錄是一個變數。`prefix` 引數表示要安裝 Apache 的位置。版本編號中的 `x` 表示您的實際版本。

- 3 對於 Apache 2.0，配置原始碼樹狀結構：

a. 透過執行 `cd` 指令轉至 `http-2.0_x` 目錄。

- b. 執行 `./configure --with-ssl=open-ssl-install-path --prefix=install-dir --enable-ssl --enable-so`

上述指令範例中指定的目錄是一個變數。 *prefix* 引數表示要安裝 Apache 的位置。版本編號中的 *x* 表示您的實際版本。

4 對於在 Linux 2.1 上執行的 Apache，編譯之前請：

- a. 開啓 `src/Makefile` 並尋找自動產生的區段的結束位置。

- b. 在自動產生的區段之後的前四行的後面增加以下行：

```
LIBS+= -licuuc -licui18n -lnspr4 -lpthread -lxerces-c -lsupport -lnsprwrap
-lns-httpd40
```

```
LDFLAGS+= -L/install-dir/lib -L/opt/sun/private/lib
```

備註 – 僅當您將 Application Server 做為 Java Enterprise System 安裝的一部分進行安裝時，才需要 `-L/opt/sun/private/lib` 部分。

例如：

```
## (End of automatically generated section)##
```

```
CFLAGS=$(OPTIM) $(CFLAGS1) $(EXTRA_CFLAGS) LIBS=$(EXTRA_LIBS)
$(LIBS1) INCLUDES=$(INCLUDES1) $(INCLUDES0)
$(EXTRA_INCLUDES) LDFLAGS=$(LDFLAGS1) $(EXTRA_LDFLAGS)"
```

```
LIBS+= -licuuc -licui18n -lnspr4 -lpthread -lxerces-c -lsupport -lnsprwrap
-lns-httpd40 LDFLAGS+= -L/install-dir/lib -L/opt/sun/private/lib
```

- c. 建立環境變數 `LD_LIBRARY_PATH`，它等於：`install-dir/lib` (適用於所有安裝) 和 `install-dir/lib:opt/sun/private/lib` (適用於做為 Java Enterprise System 安裝的一部分進行安裝的 Application Server)。

5 按照適用於您所使用之版本的安裝說明編譯 Apache。完整文件可從 <http://httpd.apache.org/> 取得。

一般來說，這些步驟包括：

- a. `make`

- b. `make certificate` (僅適用於 Apache 1.3)

- c. `make install`

指令 `make certificate` 需要安全密碼。請記住此密碼，因為啓動安全 Apache 需要它。

6 為您的環境配置 Apache。

完成 Apache 安裝後，請參閱「對 Apache Web Server 的修改」以取得在安裝外掛程式後配置 Apache 的資訊。

自動重新啓動網域或節點代理程式

如果網域或節點代理程式意外停止 (例如，您需要重新啓動機器)，您可以將系統配置爲自動重新啓動網域或節點代理程式。

本附錄包含下列主題：

- 第 311 頁的「在 UNIX 平台上自動重新啓動」
- 第 312 頁的「在 Microsoft Windows 平台上自動重新啓動」
- 第 313 頁的「自動重新啓動的安全性」

在 UNIX 平台上自動重新啓動

若要在 UNIX 平台上重新啓動網域，請在 `/etc/inittab` 檔案中增加一行文字。

例如，若要重新啓動安裝在 `opt/SUNWappserver` 目錄中的 Application Server 之 `domain1`，請使用名爲 `password.txt` 的密碼檔案：

```
das:3:respawn:/opt/SUNWappserver/bin/asadmin start-domain --user admin
--passwordfile /opt/SUNWappserver/password.txt domain1
```

將這些文字放在一行上。前三個字母是程序的唯一指示符，可以進行更改。

重新啓動節點代理程式的語法與此相似。例如，若要重新啓動安裝在 `opt/SUNWappserver` 目錄中的 Application Server 之 `agent1`，請使用名爲 `password.txt` 的密碼檔案：

```
das:3:respawn:/opt/SUNWappserver/bin/asadmin start-node-agent --user admin
--passwordfile /opt/SUNWappserver/password.txt agent1
```

在 Microsoft Windows 平台上自動重新啟動

若要在 Microsoft Windows 上自動重新啟動，請建立一個 Windows 服務。將 Sun Java System Application Server 隨附的 `appservService.exe` 和 `appserverAgentService.exe` 可執行檔與 Microsoft 提供的服務控制指令 (`sc.exe`) 配合使用。

Windows XP 隨附了 `sc.exe` 指令，該指令位於 `C:\windows\system32` 目錄或 `C:\winnt\system32` 目錄。編寫本文件時，已可以從 <http://ftp://ftp.microsoft.com/reskit/win2000/sc.zip> 下載 Windows 2000 `sc.exe`。如需有關使用 `sc.exe` 的更多資訊，請參閱 http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndllpro/html/msdn_scmslite.asp。

使用 `appservService.exe` 和 `appservAgentService.exe`，如下所示：

```
C:\winnt\system32\sc.exe create service-name binPath= \"fully-qualified-path-to-appservService.exe\"
\"fully-qualified-path-to-asadmin.bat start-command\"
\"fully-qualified-path-to-asadmin.bat stop-command\"
start= auto DisplayName= \"display-name\"
```

例如，若要建立可以啟動和停止網域 `domain1` 的名為 `SunJavaSystemAppServer DOMAIN1` 的服務，請使用密碼檔案 `C:\Sun\AppServer\password.txt`：

```
C:\windows\system32\sc.exe create domain1 binPath=
\"C:\Sun\AppServer\lib\appservService.exe \"C:\Sun\AppServer\bin\asadmin.bat
start-domain --user admin --passwordfile C:\Sun\AppServer\password.txt domain1\"
\"C:\Sun\AppServer\bin\asadmin.bat stop-domain domain1\" start= auto
DisplayName= \"SunJavaSystemAppServer DOMAIN1\"
```

若要建立可以啟動和停止節點代理程式 `agent1` 的服務，請使用：

```
C:\windows\system32\sc.exe create agent1 binPath=
\"C:\Sun\AppServer\lib\appservAgentService.exe \"C:\Sun\AppServer\bin\asadmin.bat
start-node-agent --user admin --passwordfile C:\Sun\AppServer\password.txt agent1\"
\"C:\Sun\AppServer\bin\asadmin.bat stop-node-agent agent1\" start= auto
DisplayName= \"SJESAS_SE8.1 AGENT1\"
```

備註 – 做為 `binPath=` 參數的一部分而輸入的啟動和停止指令必須具有正確的語法。請在指令提示符號下執行這些指令以進行測試。如果指令無法正確啟動或停止網域或節點代理程式，則說明該服務無法正常工作。

備註 – 請勿將服務與 `asadmin start` 和 `stop` 指令混合使用來進行啟動和停止。否則，將導致伺服器狀態不同步。例如，即使元件沒有執行，該服務可能也會顯示已啟動元件。為避免發生這種情況，請在使用服務時始終使用 `sc.exe` 指令啟動和停止元件。

自動重新啟動的安全性

當按以下方式之一啟動時，需要輸入所需的密碼和主密碼：

- 在 Microsoft Windows 上，將服務配置為要求使用者輸入密碼。
 1. 在 [服務] 控制台中，按兩下您所建立的服務。
 2. 在 [特性] 視窗中，按一下 [登入] 標籤。
 3. 核取 [允許服務與桌面互動]，以在啟動元件時提示輸入所需的密碼。
必須登入才能看到提示，鍵入的項目不會在螢幕上顯示出來。如果選擇使用服務，這種方法最為安全，但需要進行使用者互動才能啟動服務。
如果未設定 [與桌面互動] 選項，服務將保持「啟動擱置」狀態，並且將顯示為掛機。中止該服務程序即可從此狀態中恢復。
- 在 Windows 或 UNIX 上，使用 `--savemasterpassword=true` 選項建立網域，並建立儲存管理員密碼的密碼檔案。啟動元件時，使用 `--passwordfile` 選項指向包含密碼的檔案。

例如：

1. 建立具有已儲存主密碼的網域。在下面的語法中，系統將提示您輸入管理員密碼和主密碼：

```
asadmin create-domain --adminport 4848 --adminuser admin
--savemasterpassword=true --instanceport 8080 domain1
```

2. 在 Windows 上，建立使用密碼檔案的服務，以填入管理員密碼：

```
C:\windows\system32\sc.exe create domain1 binPath=
"C:\Sun\AppServer\lib\appservService.exe \"C:\Sun\AppServer\bin
\asadmin.bat start-domain --user admin --passwordfile C:\Sun
\AppServer\password.txt domain1\" \"C:\Sun\AppServer\bin
\asadmin.bat stop-domain domain1\" start= auto
DisplayName= "SJESAS_PE8.1 DOMAIN1"
```

密碼檔案 `password.txt` 的路徑為 `C:\Sun\AppServer\password.txt`。該檔案包含以下格式的密碼

```
AS_ADMIN_password=password
```

例如，對於密碼 `adminadmin`：

```
AS_ADMIN_password=adminadmin
```

3. 在 UNIX 上，請使用為 `inittab` 檔案增加的行中的 `--passwordfile` 選項：

```
das:3:respawn:/opt/SUNWappserver/bin/asadmin start-domain --user admin
--passwordfile /opt/SUNWappserver/password.txt domain1
```

密碼檔案 `password.txt` 的路徑為 `/opt/SUNWappserver/password.txt`。該檔案包含以下格式的密碼

```
AS_ADMIN_password=password
```

例如，對於密碼 `adminadmin`：

```
AS_ADMIN_password=adminadmin
```

domain.xml 的含點名稱屬性

此附錄說明可用於描述 MBean 及其屬性的帶點名稱屬性。domain.xml 檔案中的每個元素均有對應的 MBean。由於使用這些名稱的語法是利用小數點號來分隔名稱，所以這些名稱稱為「帶點名稱」。

本附錄包含下列主題：

- [第 315 頁的「頂層元素」](#)
- [第 317 頁的「不能別名化的元素」](#)

頂層元素

domain.xml 檔案中的所有頂層元素均必須符合以下條件：

- 每個伺服器、配置、叢集或節點代理程式的名稱都必須是專屬名稱。
- 不能將伺服器、配置、叢集或節點代理程式命名為 domain。
- 不能將伺服器實例命名為 agent。

下表列出了頂層元素及其對應的含點名稱字首。

元素名稱	含點名稱字首
應用程式	domain.applications
resources	domain.resources
配置	domain.configs
伺服器	domain.servers 此元素中包含的每個伺服器均可做為 <i>server-name</i> 進行存取。其中， <i>server-name</i> 是伺服器子元素的名稱屬性值。

元素名稱	含點名稱字首
叢集	domain.clusters 此元素中包含的每個叢集均可做為 <i>cluster-name</i> 進行存取。其中， <i>cluster-name</i> 是叢集子元素的名稱屬性值。
node-agents	domain.node-agents
lb-configs	domain.lb-configs
system-property	domain.system-property

別名包括兩個層級：

1. 透過第一層級的別名可以存取伺服器實例或叢集的屬性而無需 `domain.servers` 或 `domain.clusters` 前綴。因此，舉例來說，形式為 `server1` 格式的帶點名稱與帶點名稱 `domain.servers.server1` (其中 `server1` 為伺服器實例) 對映。
2. 第二層級的別名用於表示叢集或獨立伺服器實例(目標)的配置、應用程式和資源。

下表列出了以伺服器名稱或叢集名稱黃Y的含點名稱，這些含點名稱被別名化為網域下的頂層名稱：

含點名稱	別名化為	注釋
<i>target.applications.*</i>	<code>domain.applications.*</code>	該別名解析為僅由目標參照的應用程式。
<i>target.resources.*</i>	<code>domain.resources.*</code>	該別名解析為由目標參照的所有 <code>jdbc-connection-pool</code> 、 <code>connector-connection-pool</code> 、 <code>resource-adapter-config</code> 和所有其他資源。

下表列出了以伺服器名稱或叢集名稱黃Y的含點名稱，這些含點名稱在伺服器或叢集所參考的配置中被別名化為頂層名稱。

含點名稱	別名化為
<i>target.http-service</i>	<code>config-name.http-service</code>
<i>target.iiop-service</i>	<code>config-name.iiop-service</code>
<i>target.admin-service</i>	<code>config-name.admin-service</code>
<i>target.web-container</i>	<code>config-name.web-container</code>
<i>target.ejb-container</i>	<code>config-name.ejb-container</code>
<i>target.mdb-container</i>	<code>config-name.mdb-container</code>
<i>target.jms-service</i>	<code>config-name.jms-service</code>

含點名稱	別名化為
<i>target.log-service</i>	<i>config-name.log-service</i>
<i>target.security-service</i>	<i>config-name.security-service</i>
<i>target.transaction-service</i>	<i>config-name.transaction-service</i>
<i>target.monitoring-service</i>	<i>config-name.monitoring-service</i>
<i>target.java-config</i>	<i>config-name.java-config</i>
<i>target.availability-service</i>	<i>config-name.availability-service</i>
<i>target.thread-pools</i>	<i>config-name.thread-pools</i>

不能別名化的元素

不應對叢集實例進行別名化。若要取得叢集實例的系統特性，應依循以下方法使用帶點名稱屬性：`domain.servers.clustered-instance-name.system-property`，而非 `clustered-instance-name.system-property`。

索引

A

ACC

請參閱容器

應用程式用戶端, 131

accesslog 特性, 虛擬伺服器, 230

AddressList 特性, 94

AddressListBehavior 特性, 95

AddressListIterations 特性, 95

Admin Console, 34

allowLinking 特性, 虛擬伺服器, 231

append-version 特性, 104

applet, 131

Application Server, 關閉, 138

Application Server 網域, 35

asadmin 公用程式, 34

asadmin 指令, 245, 246

create-threadpool, 245

delete-threadpool, 246

asadmintruststore 檔案, 37

B

bean-cache, 監視屬性名稱, 262

C

cache-hits, 262

cache-misses, 262

chunkedRequestBufferSize 特性, 224

chunkedRequestTimeoutSeconds 特性, 224

ClientID 特性, 94

CORBA, 237

執行緒, 243

create-domain 指令, 36

create-jndi-resource 指令, 119

D

delete-domain 指令, 36

Description 特性, JMS 目標資源, 97

dnsCacheEnabled 特性, 224

docroot 特性, 虛擬伺服器, 230

E

EAR 檔案, 57

EJB JAR 檔案, 57

EJB 模組, 部署, 62

Enterprise Java Beans, 執行緒, 243

Enterprise JavaBeans

永久性, 132

有狀態階段作業, 136-137, 138

池, 135, 137-138

快取, 132, 135, 136-137

建立, 132

使用中, 136

計時器服務, 138-139

訊息導引, 132, 137-138

階段作業, 132

授權, 132

從快取中移除, 136

Enterprise JavaBeans (續)

- 啟動, 132
- 移除閒置, 137
- 閒置, 135, 136
- 無狀態階段作業, 135
- 鈍化, 132, 134-135, 136
- 實體, 132, 135, 136-137
- execution-time-millis, 261

G

- get 指令, 監視資料, 286

H

- HTTP protocol, HTTP service, 227-228
- HTTP service, HTTP protocol, 227-228
- HTTP 服務
 - chunkedRequestBufferSize 特性, 224
 - chunkedRequestTimeoutSeconds 特性, 224
 - dnsCacheEnabled 特性, 224
 - HTTP 偵聽程式, 220-222
 - HTTP 檔案快取, 228-229
 - keepAliveQueryMaxSleepTime 特性, 224
 - keepAliveQueryMeanTime 特性, 224
 - monitoringCacheEnabled 特性, 223
 - monitoringCacheRefreshInMillis 特性, 223
 - ssl3SessionTimeout 特性, 223
 - sslCacheEntries 特性, 223
 - sslClientAuthDataLimit 特性, 223
 - sslClientAuthTimeout 特性, 223
 - sslSessionTimeout 特性, 223
 - stackSize 特性, 224
 - statsProfilingEnabled 特性, 224
 - traceEnabled 特性, 223
 - 存取記錄, 224-226
 - 持續作用子系統, 222, 226-227
 - 連線池, 227
 - 配置, 223-224
 - 虛擬伺服器, 219-220
 - 請求處理執行緒, 221, 226
 - 簡介, 219
- HTTP 連接埠, 變更, 51-52

- HTTP 階段作業, 132
- HTTP 偵聽程式
 - 刪除, 235-236
 - 建立, 233-235
 - 接收器執行緒, 221
 - 預設虛擬伺服器, 221
 - 編輯, 235
 - 簡介, 220-222
- HTTP 檔案快取, HTTP 服務, 228-229

I

- IIOp 連接埠, 變更, 52
- IIOp 偵聽程式, 238
 - 刪除, 241
 - 建立, 239-240
 - 編輯, 240-241
- instance-name-suffix 特性, 103
- instance-name 特性, 103

J

- J2EE 群組, 167
- J2SE 軟體, 53
- Java 文件, 27
- Java 命名和目錄服務, 請參閱 JNDI, 132
- Java 訊息服務 (JMS), 請參閱 JMS 資源, 91
- JavaMail, 33
- JavaMail API, 簡介, 109
- JavaMail 階段作業
 - 刪除, 111-112
 - 建立, 110-111
 - 編輯, 111
- JavaServer Page, 131
- JCE 提供者
 - 配置, 200
- JDBC, 33
 - 資源, 138
 - 驅動程式, 214
- jms-max-messages-load, 262
- JMS 主機
 - 刪除, 106-107
 - 建立, 104-105

JMS 主機 (續)

編輯, 106

JMS 提供者, 91

append-version 特性, 104
instance-name-suffix 特性, 103
instance-name 特性, 103
JMS 主機, 104-105, 106
配置, 101-104

JMS 資源

目標資源, 91-92, 97, 98
主題, 91-92
佇列, 91-92
連線工廠資源, 91-92, 93-95, 96
實體目標, 91-92, 99-100, 100
簡介, 91-92

jmsra 系統資源配接卡, 93

JNDI, 132

EJB 元件的查詢名稱, 57
外部資源, 刪除, 119
外部資源, 建立, 118-119
外部資源, 編輯, 119
外部儲存庫, 117
名稱, 113, 138
自訂資源, 刪除, 116-117, 117
自訂資源, 建立, 115-116
自訂資源, 使用, 115
查詢及其關聯的參照, 114

JSP, 請參閱 JavaServer Page, 131

K

keepAliveQueryMaxSleepTime 特性, 224
keepAliveQueryMeanTime 特性, 224
keystore.jks 檔案, 185

L

list-custom-resources 指令, 117
list-domains 指令, 36
list-jndi-resource 指令, 120
list 指令, 監視, 285

M

maxNumActiveConsumers 特性, JMS 實體目標, 100
Message Queue 軟體, 91
MessageServiceAddressList 特性, 94
monitoringCacheEnabled 特性, 223
monitoringCacheRefreshInMillis 特性, 223

N

Name 特性, JMS 目標資源, 97
numbeansinpool, 262
numexpiredsessionsremoved, 263
numpassivationerrors, 263
numpassivations, 263
numpassivationsuccess, 263
numthreadswaiting, 262

O

Oasis Web Services Security, 參閱 WSS
Oracle, 139
ORB, 237
IIOP 偵聽程式, 238
服務, 監視, 268
配置, 238-239
請參閱物件請求代理程式, 243
簡介, 237-238

P

Password 特性, 94
PointBase, 139

R

RAR 檔案, 57
ReconnectAttempts 特性, 95
ReconnectEnabled 特性, 94
ReconnectInterval 特性, 95
RSA 加密, 200

S

servlet, 131
Solaris
 支援, 28
 修補程式, 28
ssl3SessionTimeout 特性, 223
sslCacheEntries 特性, 223
sslClientAuthDataLimit 特性, 223
sslClientAuthTimeout 特性, 223
sslSessionTimeout 特性, 223
sso-enabled 特性, 虛擬伺服器, 230
sso-max-inactive-seconds 特性, 虛擬伺服器, 231
sso-reap-interval-seconds 特性, 虛擬伺服器, 231
stackSize 特性, 224
start-domain 指令, 37, 117, 119
statsProfilingEnabled 特性, 224
stop-domain 指令, 38
Sun Java System Message Queue 軟體, 91

T

total-beans-created, 262
total-beans-destroyed, 262
total-num-errors, 260
total-num-success, 260
traceEnabled 特性, 223
truststore.jks 檔案, 185

U

UserName 特性, 94

W

WAR 檔案, 56
Web 服務, 33
Web 階段作業, 請參閱 HTTP 階段作業, 132
Web 應用程式, 56
 啟動, 62
 部署, 60
工作佇列, 請參閱執行緒池, 244
文件, 簡介, 26-28

中央儲存庫, 應用程式部署到, 55
生命週期模組
 建立, 65
 載入順序, 66
 類別路徑, 66
用戶端存取, 32
用於應用程式的服務, 33
目標
 已部署的應用程式, 56
 管理應用程式, 71
目標, JMS
 Description 特性, 97
 maxNumActiveConsumers 特性, 100
 Name 特性, 97
 刪除目標資源, 98-99
 刪除實體目標, 100
 建立目標資源, 97
 建立實體目標, 99-100
 編輯目標資源, 98
 簡介, 91-92
目錄部署, 74
外部資源
 刪除, 119
 建立, 118-119
 編輯, 119
外部儲存庫, 存取, 117
主題, JMS, 91-92
企業應用程式, 57
 部署, 58
回復
 請參閱作業事件
 回復, 213
安全性, 33
存取記錄, HTTP 服務, 224-226
有狀態階段作業 Bean, 請參閱 Enterprise
 JavaBeans, 136-137
自訂資源
 列示, 117
 刪除, 116-117
 建立, 115-116
 使用, 115
自動部署應用程式, 73
池
 Enterprise JavaBeans, 135, 137-138

- 作業事件, 213
 - Enterprise JavaBeans, 135
 - JMS 連線工廠, 94
 - 分隔, 214
 - 分散式, 214
 - 回復, 213, 214, 215
 - 完成, 214
 - 記錄, 216
 - 逾時, 215-216
 - 管理員, 214
 - 確定, 213
 - 關聯, 214
 - 屬性, 214
- 作業事件服務, 監視, 268-269
- 作業事件管理, 33
- 作業事件管理員
 - 請參閱作業事件管理員, 214
- 快取
 - Enterprise JavaBeans, 135
 - 停用, 135
 - 清除, 136
 - 逾時, 136
- 伺服器記錄, 檢視, 252-253
- 伺服器管理, 33
- 佇列
 - 工作
 - 請參閱執行緒池, 244
- 佇列, JMS, 91-92
- 物件請求代理程式, 執行緒, 243
- 物件請求代理程式 (ORB), 237
 - 配置, 238-239
 - 簡介, 237-238
- 命名, JNDI 和資源參照, 114
- 命名和目錄服務, 33
- 命名服務, 33
- 命名慣例, 針對應用程式, 57
- 定義叢集, 41
- 取消部署應用程式, 70
- 服務, 計時器, 138-139
- 重新啟動伺服器, 37
- 重新部署應用程式, 56, 71
- 持續作用子系統
 - HTTP 服務, 222, 226-227
- 計時器
 - 請參閱 Enterprise JavaBeans 計時器服務, 138-139
- 計時器服務
 - 請參閱 Enterprise JavaBeans 計時器服務, 138-139
- 記錄
 - 作業事件, 216
 - 記錄程式名稱空間, 248-249
 - 配置一般設定, 250-251
 - 配置層級, 251-252
 - 檢視伺服器記錄, 252-253
 - 簡介, 247-248
- 記錄層級, 配置, 251-252
- 記錄檔的記錄, 247-248
- 連接埠偵聽程式, 50
- 連接埠號, 檢視, 51
- 連接埠號, 變更, 51
- 連接器, 33
 - 模組, 243
- 連接器連線池, JMS 資源和, 93
- 連接器資源, JMS 資源和, 93
- 連接器模組, 部署, 64
- 連線工廠, JMS
 - AddressList 特性, 94
 - AddressListBehavior 特性, 95
 - AddressListIterations 特性, 95
 - ClientID 特性, 94
 - MessageServiceAddressList 特性, 94
 - Password 特性, 94
 - ReconnectAttempts 特性, 95
 - ReconnectEnabled 特性, 94
 - ReconnectInterval 特性, 95
 - UserName 特性, 94
 - 刪除, 96
 - 作業事件支援, 94
 - 建立, 93-95
 - 編輯, 96
 - 簡介, 91-92
- 連線池, HTTP 服務, 227
- 效能
 - 問題, 135
 - 執行緒池, 243
 - 提高, 135

- 高可用性, 32
- 容器, 32
 - applet, 131
 - Enterprise JavaBeans, 131, 134-137
 - 配置, 134-137
 - J2EE, 131
 - servlet
 - Web, 131
 - 請參閱容器, 131
 - Web, 131
 - web, 131
 - 應用程式用戶端, 131
- 階段作業
 - HTTP, 132, 134-135
 - ID, 133
 - 自訂 ID, 133
 - 刪除, 134
 - 刪除資料, 133
 - 非使用中, 133, 134
 - 配置, 132-134
 - 逾時, 133
 - 管理, 133
 - 檔案名稱, 133
 - 儲存, 134-135
 - 儲存資料, 133
- 階段作業管理員, 133
- 停用應用程式, 70
- 執行緒
 - 移除, 244, 245
 - 請參閱執行緒池, 243
- 執行緒池, 243
 - 工作佇列, 244, 245
 - 刪除, 246
 - 命名, 244-245
 - 建立, 244-245
 - 效能, 243
 - 執行緒不足, 243
 - 逾時, 244, 245
 - 閒置, 244, 245
 - 編輯, 245
- 接收器執行緒, 在 HTTP 偵聽程式中, 221
- 啟用應用程式, 70
- 清除間隔, 133, 134
- 單次登入, 虛擬伺服器特性, 230
- 逾時, 135, 136, 137
 - 執行緒池, 244, 245
- 無狀態階段作業 Bean, 請參閱 Enterprise JavaBeans, 135
- 虛擬伺服器
 - accesslog 特性, 230
 - allowLinking 特性, 231
 - docroot 特性, 230
 - sso-enabled 特性, 230
 - sso-max-inactive-seconds 特性, 231
 - sso-reap-interval-seconds 特性, 231
 - 刪除, 232-233
 - 建立, 229-231
 - 將應用程式部署到其他, 71
 - 編輯, 231-232
 - 簡介, 219-220
- 資料庫
 - JNDI 名稱, 113
 - Oracle, 139
 - PointBase, 139
 - 資源參照, 114
- 資源 RAR 檔案, 57
- 資源介面, 214
 - 部署, 64
- 資源配接卡, jmsra, 93
- 資源參照, 114
- 資源管理員, 214
- 載入順序, 在生命週期模組中°, 66
- 網域, 35
 - 建立, 36
 - 部署應用程式到, 55
- 實例, 41
- 實體 Bean
 - 請參閱 Enterprise JavaBeans 實體, 135
- 監視
 - bean-cache 屬性, 262
 - ORB 服務, 268
 - 作業事件服務, 268-269
 - 使用 get 指令, 286
 - 使用 list 指令, 285
 - 容器子系統, 256-257
- 模組描述元, 檢視, 69
- 範圍, 憑證, 152

- 郵件傳送, 33
- 部署規劃, 75
- 線上手冊, 34
- 線上說明, 54
- 請求處理執行緒
 - HTTP 服務, 221, 226
- 應Application Serve, 重新啟動, 138
- 應用程式
 - 目錄部署, 74
 - 列示已部署的, 68-69
 - 列示子元件, 69
 - 在虛擬伺服器上部署, 71
 - 自動部署, 73
 - 命名慣例, 57
 - 取消部署, 70
 - 重新部署, 56, 71
 - 效能, 135
 - 停用, 70
 - 啟用, 70
 - 模組描述元, 69
 - 部署規劃, 75
- 應用程式用戶端 JAR 檔案, 57
- 應用程式用戶端模組, 部署, 66
- 應用程式的子元件, 列示, 69
- 叢集, 31
- 叢集, 定義, 41
- 關鍵點作業, 217
- 關鍵點間隔, 217
- 類別路徑, 在生命週期模組中, 66

