



Sun Java System Application Server 9.1 高可用性管理指南



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

文件號碼：820-4611
2007 年 12 月

Copyright 2007 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. 版權所有。

本產品或文件受版權保護，且按照限制其使用、複製、發行和反編譯的授權進行發行。未經 Sun 及其授權人(如果有)事先的書面許可，不得使用任何方法、任何形式來複製本產品或文件的任何部分。協力廠商軟體，包含字型技術，其版權歸 Sun 供應商所有，經授權後使用。

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、docs.sun.com、AnswerBook、AnswerBook2、與 Solaris 是 Sun Microsystems, Inc. 在美國及其他國家/地區的商標或註冊商標。所有 SPARC 商標都是 SPARC International, Inc. 在美國及其他國家/地區的商標或註冊商標，經授權後使用。凡具有 SPARC 商標的產品都是採用 Sun Microsystems, Inc. 所開發的架構。

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

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	15
1 應用程式伺服器的高可用性	21
高可用性簡介	21
高可用性階段作業持續性	22
高可用性 Java 訊息服務	22
RMI-IIOP 負載平衡和容錯移轉	23
更多資訊	23
Application Server 如何提供高可用性	24
負載平衡外掛程式	24
階段作業狀態資料的儲存方式	25
高可用性叢集	26
故障後回復	27
使用 Sun Cluster	27
手動回復	27
使用 Netbackup	29
重新建立網域管理伺服器	30
▼ 遷移 DAS	30
2 安裝和設定高可用性資料庫	33
準備 HADB 設定	33
先決條件和限制	34
配置網路備援	34
配置共用記憶體和信號	37
▼ 在 Solaris 上配置共用記憶體和信號	37
▼ 在 Linux 上配置共用記憶體	38
同步化系統時鐘	39

安裝	40
HADB 安裝	40
節點管理員程序權限	40
▼ 授予節點管理員程序超級使用者權限	41
設定高可用性	42
▼ 準備系統使其具有高可用性	42
啟動 HADB 管理代理程式	42
配置叢集以使其具有高可用性	43
配置應用程式以使其具有高可用性	43
重新啟動叢集	43
重新啟動 Web 伺服器	44
▼ 清除做為負載平衡器執行的 Web 伺服器實例	44
升級 HADB	44
▼ 將 HADB 升級至較新的版本	45
註冊 HADB 套裝軟體	45
取消註冊 HADB 套裝軟體	46
替換管理代理程式啟動程序檔	47
▼ 驗證 HADB 升級	47
3 管理高可用性資料庫	49
使用 HADB 管理代理程式	49
啟動管理代理程式	50
管理代理程式指令語法	54
自訂管理代理程式配置	55
▼ 在 HADB 主機上自訂管理代理程式配置	55
使用 hadbm 管理指令	57
指令語法	57
安全性選項	58
一般選項	59
環境變數	60
配置 HADB	61
建立管理網域	61
建立資料庫	62
▼ 建立資料庫	63
檢視和修改配置屬性	67

配置 JDBC 連線池	72
管理 HADB	74
管理網域	75
管理節點	76
管理資料庫	78
回復階段作業資料損壞	81
▼ 使階段作業存放區返回一致狀態	82
延伸 HADB	82
增加現有節點的儲存空間	83
增加機器	83
▼ 新增機器至現有 HADB 實例	84
增加節點	84
重新分段資料庫	86
透過重新建立資料庫增加節點	87
▼ 透過重新建立資料庫增加節點	87
監視 HADB	88
取得 HADB 的狀態	88
取得裝置資訊	90
取得執行階段資源資訊	91
維護 HADB 機器	94
▼ 在一台機器上執行維護	94
▼ 在所有 HADB 機器上執行規劃的維護	95
▼ 在所有 HADB 機器上執行規劃的維護	95
▼ 發生故障時進行臨時維護	95
清除和歸檔歷史檔案	96
4 配置 Web 伺服器以用於負載平衡	97
配置 Sun Java System Web Server	98
▼ 配置 Sun Java System Web Server	98
配置 Sun Java System Web Server 以使用自動套用	99
▼ 在 SSL 模式中為 Sun Java System Web Server 6.1 設定負載平衡器	99
▼ 匯出及匯入 Sun Java System Web Server 6.1 的 DAS 憑證	101
在 SSL 模式中為 Web Server 7 設定負載平衡器	102
▼ 匯出及匯入 Sun Java System Web Server 7 的 DAS 憑證	103
使用 Apache Web Server	105

使用 Apache Web Server 的需求	106
在安裝負載平衡外掛程式之前先配置 Apache	107
▼ 安裝 SSL 可識別的 Apache	107
匯出及匯入 DAS 憑證	109
負載平衡外掛程式安裝程式所做的修改	110
安裝負載平衡外掛程式後配置 Apache	111
▼ 為 Apache 建立安全性憑證	111
在 Solaris 和 Linux 上啟動 Apache	112
確認設定	112
使用 Microsoft IIS	113
▼ 配置 Microsoft IIS 以使用負載平衡外掛程式	113
自動配置的 sun-passthrough 特性	114
5 配置 HTTP 負載平衡	117
負載平衡器外掛程式的新增功能	117
自動套用	117
加權循環	118
使用者定義的負載平衡	118
▼ 配置使用者定義的負載平衡	118
HTTP 負載平衡器的工作方式	119
HTTP 負載平衡演算法	119
設定 HTTP 負載平衡	120
設定負載平衡的先決條件	120
設定負載平衡的程序	121
▼ 使用管理主控台設定負載平衡	121
▼ 使用 asadmin 工具設定負載平衡	122
HTTP 負載平衡器部署	123
配置負載平衡器	124
在 DAS 上配置 HTTP 負載平衡器	125
建立 HTTP 負載平衡器參照	126
啓用用於負載平衡的伺服器實例	126
啓用用於負載平衡的應用程式	126
建立 HTTP 運作狀態檢查程式	127
匯出負載平衡器配置檔案	129
▼ 使用管理主控台匯出負載平衡器配置	129

▼ 使用 asadmin 工具匯出負載平衡器配置	129
變更負載平衡器配置	130
啓用動態重新配置	130
停用 (靜止) 伺服器實例或叢集	130
▼ 停用某個伺服器實例或叢集	131
停用 (靜止) 應用程式	131
▼ 停用應用程式	131
配置 HTTP 和 HTTPS 容錯移轉	132
透過負載平衡器使用重新導向	133
配置等冪 URL	136
配置多個 Web 伺服器實例	136
▼ 配置多個 Web 伺服器實例	136
在維持可用性的情況下升級應用程式	137
應用程式相容性	137
在單一叢集中升級	137
▼ 在單一叢集中升級應用程式	138
在多個叢集中升級	139
▼ 若要在兩個或更多叢集中升級相容的應用程式，請執行以下步驟：	139
升級不相容的應用程式	141
▼ 透過建立第二個叢集來升級不相容的應用程式	141
監視 HTTP 負載平衡器外掛程式	143
配置記錄訊息	143
記錄訊息類型	143
啓用負載平衡器記錄	144
▼ 開啓負載平衡器記錄	145
瞭解監視訊息	146
6 使用 Application Server 叢集	147
叢集簡介	147
群組管理服務	147
▼ 啓用或停用叢集的 GMS 功能	148
配置 GMS	148
使用叢集	149
▼ 建立叢集	149
▼ 爲叢集建立伺服器實例	150

▼ 配置叢集	151
▼ 啓動、停止和刪除叢集實例	152
▼ 配置叢集中的伺服器實例	152
▼ 爲叢集配置應用程式	153
▼ 爲叢集配置資源	153
▼ 刪除叢集	154
▼ 遷移 EJB 計時器	154
▼ 升級元件而不會使服務受到損失	155
7 管理配置	157
使用配置	157
配置	157
default-config 配置	158
建立實例或叢集時建立的配置	158
唯一連接埠號和配置	159
使用已命名的配置	160
▼ 建立已命名的配置	160
編輯已命名的配置的特性	160
▼ 編輯已命名的配置的特性	161
▼ 編輯參照配置之實例的連接埠號	161
▼ 檢視已命名的配置的目標	162
▼ 刪除已命名的配置	162
8 配置節點代理程式	165
何爲節點代理程式？	165
節點代理程式失敗後的伺服器實例運作方式	166
部署節點代理程式	167
▼ 線上部署節點代理程式	167
▼ 離線部署節點代理程式	168
同步化節點代理程式與網域管理伺服器	169
節點代理程式同步	169
伺服器實例同步	170
同步程式庫檔案	171
獨特的設定和配置管理	171
同步大型應用程式	172

檢視節點代理程式記錄	173
使用節點代理程式	173
如何執行節點代理程式作業	173
節點代理程式預留位置	174
▼ 建立節點代理程式預留位置	174
建立節點代理程式	175
▼ 建立節點代理程式	176
▼ 為 DNS 可存取的主機建立節點代理程式	176
啟動節點代理程式	177
停止節點代理程式	177
刪除節點代理程式	178
▼ 檢視一般節點代理程式資訊	178
▼ 刪除節點代理程式配置	179
▼ 編輯節點代理程式配置	179
▼ 編輯節點代理程式範圍	180
▼ 為 JMX 編輯節點代理程式的偵聽程式	180
9 配置高可用性階段作業持續性和容錯移轉	183
階段作業持續性和容錯移轉簡介	183
需求	183
限制	184
設定高可用性階段作業持續性	185
▼ 設定高可用性階段作業持續性	185
啟用階段作業可用性	186
HTTP 階段作業容錯移轉	188
配置 Web 容器的可用性	188
▼ 使用管理主控台啟用 Web 容器的可用性	189
配置個別 Web 應用程式的可用性	190
將階段作業容錯移轉與單次登入配合使用	191
有狀態階段作業 Bean 容錯移轉	192
配置 EJB 容器的可用性	193
▼ 啟用 EJB 容器的可用性	193
配置個別應用程式或 EJB 模組的可用性	194
配置個別 Bean 的可用性	195
指定進行檢查點操作方法	195

10	Java 訊息服務的負載平衡和容錯移轉	197
	Java 訊息服務簡介	197
	詳細資訊	197
	配置 Java 訊息服務	198
	Java 訊息服務整合	199
	JMS 主機清單	200
	連線池儲存和容錯移轉	200
	負載平衡的訊息流入	201
	JMS 服務高可用性	202
	將 MQ 叢集與應用程式伺服器配合使用	203
	高度可用的 MQ 叢集	203
	在本機模式中配置高度可用的代理程式叢集	203
	在遠端模式中配置高度可用的代理程式叢集	204
	非 HA 叢集的自動叢集功能	204
	▼ 以 Application Server 叢集啟用 MQ 叢集	205
11	RMI-IIOP 負載平衡和容錯移轉	209
	簡介	209
	需求	210
	演算法	210
	設定 RMI-IIOP 負載平衡和容錯移轉	211
	▼ 為應用程式用戶端容器設定 RMI-IIOP 負載平衡	211
	索引	215

表清單

表 2-1	hadbm registerpackage 選項	46
表 3-1	管理代理程式共用選項	55
表 3-2	管理代理程式服務選項 (僅適用於 Windows)	55
表 3-3	配置檔案設定	56
表 3-4	hadbm 安全性選項	59
表 3-5	hadbm 一般選項	59
表 3-6	HADB 選項與環境變數	60
表 3-7	hadbm create 選項	63
表 3-8	配置屬性	68
表 3-9	HADB 連線池設定	73
表 3-10	HADB 連線池特性	73
表 3-11	HADB JDBC 資源設定	74
表 3-12	hadbm clear 選項	81
表 3-13	hadbm addnodes 選項	85
表 3-14	HADB 狀態	89
表 3-15	hadbm resourceinfo 指令選項	92
表 5-1	負載平衡器配置參數	125
表 5-2	運作狀態檢查程式參數	127
表 5-3	運作狀態檢查程式手動特性	128
表 8-1	在遠端伺服器實例之間同步的檔案和目錄	170
表 8-2	如何執行節點代理程式作業	174

範例清單

範例 2-1	設定多重路徑	35
範例 2-2	取消註冊 HADB 的範例	47
範例 3-1	hadbm 指令範例	58
範例 3-2	建立 HADB 管理網域	62
範例 3-3	建立資料庫範例	65
範例 3-4	使用 hadbm get 之範例	67
範例 3-5	建立連線池	74
範例 3-6	啟動節點範例	77
範例 3-7	停止節點範例	77
範例 3-8	重新啟動節點範例	78
範例 3-9	啟動資料庫範例	78
範例 3-10	停止資料庫範例	79
範例 3-11	移除資料庫範例	81
範例 3-12	設定資料裝置大小範例	83
範例 3-13	增加節點範例	85
範例 3-14	重新分段資料庫範例	87
範例 3-15	取得 HADB 狀態範例	88
範例 3-16	取得裝置資訊範例	91
範例 3-17	資料緩衝區池資訊範例	93
範例 3-18	鎖定資訊範例	93
範例 3-19	記錄緩衝區資訊範例	93
範例 3-20	內部記錄緩衝區資訊範例	94
範例 8-1	建立節點代理程式	176
範例 9-1	啓用了可用性的 EJB 部署描述元範例	195
範例 9-2	指定方法檢查點之 EJB 部署描述元範例	196
範例 11-1	設定 RMI-IIOP 加權循環負載平衡的負載平衡權數	212

前言

本書說明 Application Server 中的高可用性功能，包括 HTTP 負載平衡、叢集、階段作業持續性和容錯移轉以及高可用性資料庫 (HADB)。

此前言說明整個 Sun Java™ 系統 Application Server 文件集的相關資訊與慣例。

Application Server 文件集

Application Server 文件集描述部署規劃與系統安裝。Application Server 文件的統一資源定址器 (URL) 為 <http://docs.sun.com/coll/1343.4> 和 <http://docs.sun.com/coll/1777.1>。如需有關 Application Server 的簡介，請依循下表排列順序參閱其中列出的書籍。

表 P-1 Application Server 文件集中的書籍

書籍標題	說明
文件中心	依作業及主旨編排的 Application Server 文件主題。
版本說明	軟體與文件的最新資訊。包含支援硬體、作業系統、Java 開發工具組 (JDK™)，以及資料庫驅動程式的完整表格形式摘要。
快速入門指南	如何開始使用 Application Server 產品。
安裝指南	安裝軟體及其元件。
部署規劃指南	評估系統需求和企業狀況，確保以最適合站點的方式部署 Application Server。此外還說明了部署伺服器時應該注意的常見問題。
應用程式部署指南	將應用程式及應用程式元件部署到 Application Server。包含部署描述元的相關資訊。
開發者指南	建立與實作要在 Application Server (遵循適用於 Java EE 元件及 API 的開放式 Java 標準模型) 上執行的 Java Platform, Enterprise Edition (Java EE 平台) 應用程式。其中包括有關開發者工具、安全性、除錯和建立生命週期模組的資訊。
Java EE 5 教學課程	使用 Java EE 5 平台技術及 API 來開發 Java EE 應用程式。
Java WSIT 教學課程	使用 Web 服務互通技術 (WSIT) 開發 Web 應用程式。說明如何、何時及為何要使用 WSIT 技術，以及每一項技術所支援的功能和選項。

表 P-1 Application Server 文件集中的書籍 (續)

書籍標題	說明
管理指南	Application Server 的系統管理，包括配置、監視、安全性、資源管理以及 Web 服務管理。
高可用性管理指南	高可用性資料庫安裝後的配置和管理說明。
管理參照	編輯 Application Server 配置檔案 domain.xml。
升級與遷移指南	升級舊版的 Application Server，或從其他廠商的應用程式伺服器遷移 Java EE 應用程式。本指南也說明有哪些相近的產品發行版本和配置選項差異，可能會導致產品規格不相容。
效能調校指南	調校 Application Server 以提昇效能。
疑難排解指南	解決 Application Server 問題。
錯誤訊息參照	解決 Application Server 錯誤訊息。
參考手冊	與 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 asadmin 指令行介面。

相關文件

Application Server 可單獨購買或隨附於 Sun Java Enterprise System (Java ES) 購買。後者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。若是以 Java ES 的元件方式購買 Application Server，則應熟讀位於 <http://docs.sun.com/coll/1286.3> 和 <http://docs.sun.com/coll/1412.3> 的系統文件。Java ES 及其元件的所有相關文件 URL 為 <http://docs.sun.com/prod/entsys.5>。

如需有關其他獨立 Sun Java System 伺服器產品的文件，請前往下列位置：

- Message Queue 文件 (<http://docs.sun.com/coll/1343.4>)
- (<http://docs.sun.com/coll/1224.1>) 和 Directory Server 文件 (<http://docs.sun.com/coll/1632.1>)
- (<http://docs.sun.com/coll/1308.3>) 和 Web Server 文件 (<http://docs.sun.com/coll/1425.2>)

Application Server 隨附套件的 Javadoc™ 工具參考手冊位於 <http://glassfish.dev.java.net/nonav/javaee5/api/index.html>。此外，以下資源可能會有用：

- Java EE 5 規格 (<http://java.sun.com/javaee/5/javatech.html>)
- Java EE Blueprints (<http://java.sun.com/reference/blueprints/index.html>)

如需在「NetBeans™ 整合開發環境 (IDE)」中建立企業應用程式的資訊，請參閱 <http://www.netbeans.org/kb/55/index.html>。

如需有關 Application Server 隨附的 Java DB 資料庫資訊，請參閱 <http://developers.sun.com/javadb/>。

GlassFish Samples 專案是應用程式範例的集合，展示了多項 Java EE 技術。Java EE 軟體開發套件 (SDK) 隨附了 GlassFish Samples，您也可以從 GlassFish Samples 專案網頁 (網址為 <https://glassfish-samples.dev.java.net/>) 取得。

預設路徑和檔案名稱

下表說明在本書中使用的預設路徑和檔案名稱。

表 P-2 預設路徑和檔案名稱

預留位置	說明	預設值
<i>as-install</i>	表示 Application Server 的基底安裝目錄。	Solaris™ 作業系統的 Java ES 安裝： /opt/SUNWappserver/appserver Linux 作業系統的 Java ES 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 的所有安裝： SystemDrive:\Sun\AppServer
<i>domain-root-dir</i>	表示包含所有網域的目錄。	Java ES Solaris 安裝： /var/opt/SUNWappserver/domains/ Java ES Linux 安裝： /var/opt/sun/appserver/domains/ 所有其他安裝： as-install/domains/
<i>domain-dir</i>	表示網域的目錄。 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>domain-root-dir/domain-dir</i>

表 P-2 預設路徑和檔案名稱 (續)

預留位置	說明	預設值
<i>instance-dir</i>	表示伺服器實例的目錄。	<i>domain-dir/instance-dir</i>

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-3 印刷排版慣例

字體	含義	範例
AaBbCc123	指令名稱、檔案名稱和目錄名稱以及螢幕畫面輸出	編輯 <code>.login</code> 檔案。 使用 <code>ls -a</code> 列出所有檔案。 <code>machine_name% you have mail.</code>
AaBbCc123	您所鍵入的資訊，與螢幕畫面輸出相對應。	<code>machine_name% su</code> Password:
<i>AaBbCc123</i>	將用實際的名稱或數值取代的預留位置	<code>rm filename</code> 指令可用來移除檔案。
<i>AaBbCc123</i>	書名、新的字彙或術語以及要強調的詞 (請注意，部分強調的項目在線上會以粗體顯示)	請參閱使用者指南中的第 6 章。 快取 是儲存在本機的副本。 請勿儲存檔案。

符號慣例

下表說明本書可能會使用的符號。

表 P-4 符號慣例

符號	說明	範例	含義
[]	包含選擇性引數和指令選項。	<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。
\${ }	表示變數參照。	<code>\${com.sun.javaRoot}</code>	參照 <code>com.sun.javaRoot</code> 變數的值。

表 P-4 符號慣例 (續)

符號	說明	範例	意義
-	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 鍵，然後放開，再依次按後面的按鍵。
→	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]。

文件、支援和訓練

Sun 網站提供以下其他資源的相關資訊：

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

搜尋 Sun 產品文件

除了從 docs.sun.comSM 網站搜尋 Sun 產品文件外，您也可以搜尋欄位中鍵入下列語法來使用搜尋引擎：

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

例如，若要搜尋「broker」，請鍵入：

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

若要在搜尋中加入其他 Sun 網站 (例如，java.sun.com、www.sun.com 和 developers.sun.com)，請在搜尋欄位中，使用 sun.com 替代 docs.sun.com。

協力廠商網站參考

本文件中提供了協力廠商 URL 以供參考，另亦提供其他相關的資訊。

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

Sun 歡迎您提出寶貴意見

Sun 致力於提高文件品質，因此誠心歡迎您提出意見與建議。若要分享您的意見，請至 <http://docs.sun.com>，然後按一下 [Send Comments (傳送意見)]。線上表單提供了完整的文件標題和文件號碼。文件號碼是一個七位或九位的數字，可以在書的標題頁面或文件的 URL 中找到。例如，本書的文件號碼為 820-4611。

應用程式伺服器的高可用性

本章說明叢集設定檔及企業設定檔提供的 Sun Java System Application Server 高可用性功能。

備註 – Sun Java System Application Server 的 Application Server 獨立發行版本 隨附了 HADB 軟體。如需有關 Sun Java System Application Server 的可用發行軟體資訊，請參閱「Sun Java System Application Server 9.1 Installation Guide」中的「Distribution Types and Their Components」。只有企業設定檔才提供 HADB 功能。如需有關設定檔的更多資訊，請參閱「Sun Java System Application Server 9.1 管理指南」中的「用法設定檔」。

本章包含以下主題。

- [第 21 頁的「高可用性簡介」](#)
- [第 24 頁的「Application Server 如何提供高可用性」](#)
- [第 27 頁的「故障後回復」](#)

高可用性簡介

高可用性應用程式和服務可持續提供其功能，而不考量硬體和軟體故障。這類應用程式有時亦稱為提供**五個九**的可靠性，因為開發這類產品的目的就是在 99.999% 的時間內都能供人使用。

Application Server 提供下列高可用性功能：

- 高可用性階段作業持續性
- 高可用性 Java 訊息服務
- RMI-IIOP 負載平衡和容錯移轉

高可用性階段作業持續性

Application Server 可為 HTTP 請求和階段作業資料 (HTTP 階段作業資料和有狀態階段作業 Bean 資料) 提供高可用性。

Java EE 應用程式通常具有大量階段作業狀態資料。Web 購物車即為階段作業狀態的經典範例。而且，應用程式可以快取頻繁需要的階段作業物件資料。實際上，幾乎所有需要進行大量使用者互動活動的應用程式均需要維護階段作業狀態。HTTP 階段作業和有狀態階段作業 Bean (SFSB) 均具有階段作業狀態資料。

對於一般使用者而言，在伺服器故障時保留階段作業狀態是很重要的。為了提供高可用性功能，Application Server 為階段作業狀態資料提供下列類型的儲存方式：

- 叢集內其他伺服器上的記憶體中複製
- 高可用性資料庫 (HADB)

如果主控使用者階段作業的 Application Server 實例故障，則可回復階段作業狀態，且階段作業可以繼續而無資訊遺失。

如需有關如何設定高可用性階段作業持續性的詳細說明，請參閱[第 9 章](#)。

高可用性 Java 訊息服務

Java 訊息服務 (JMS) API 是一種允許 Java EE 應用程式和元件建立、傳送、接收以及讀取郵件的郵件傳送標準。它可啟用可靠的非同步鬆耦合分散式通訊。實作 JMS 的 Sun Java System Message Queue (MQ) 與 Application Server 緊密整合在一起，可讓您建立諸如訊息驅動 Bean (MDB) 之類依賴 JMS 的元件。

JMS 透過連線池儲存和容錯移轉及 MQ 叢集來提供高可用性功能。如需更多資訊，請參閱[第 10 章](#)。

連線池儲存和容錯移轉

Application Server 支援 JMS 連線池儲存和容錯移轉。Application Server 可自動將 JMS 連線儲存在池中。依預設，Application Server 會從指定的主機清單中隨機選取其主要的 MQ 代理程式。發生容錯移轉時，MQ 不需設定即可將負載傳輸至其他代理程式並可保持 JMS 語義不變。

如需有關 JMS 連線池儲存和容錯移轉的更多資訊，請參閱[第 200 頁的「連線池儲存和容錯移轉」](#)。

MQ 叢集

MQ Enterprise Edition 支援多個互連的代理程式實例 (稱為代理程式叢集)。對於代理程式叢集，用戶端連線會分散至叢集中的所有代理程式。叢集可提供水平可延伸性並可提高可用性。

如需有關 MQ 叢集的更多資訊，請參閱第 203 頁的「將 MQ 叢集與應用程式伺服器配合使用」。

RMI-IIOP 負載平衡和容錯移轉

使用 RMI-IIOP 負載平衡時，IIOP 用戶端請求會分散至不同的伺服器實例或名稱伺服器，以便在叢集內平均分散負載，因而提高延伸性。結合了 EJB 叢集和可用性的 IIOP 負載平衡還可提供 EJB 容錯移轉。

當用戶端以 JNDI 查找物件時，命名服務會將請求連結到特定的伺服器實例。從這時開始，來自該用戶端的所有查找請求都會傳送至同一伺服器實例，因此所有 EJBHome 物件都會放在相同的目標伺服器上。之後取得的所有 Bean 參照也會建立在相同的目標主機上。這可以提供有效的負載平衡，因為所有用戶端在執行 JNDI 查找時，都會隨機製作目標伺服器清單。如果目標伺服器實例當機，則查詢或 EJB 方法呼叫將容錯移轉至其他伺服器實例。

IIOP 負載平衡與容錯移轉不需設定即可執行。在部署應用程式期間，無需特殊的步驟。如果部署應用程式用戶端的 Application Server 實例加入叢集，則此 Application Server 會自動在叢集中尋找目前所有使用中的 IIOP 端點。不過，用戶端至少應該為啟動程序指定兩個端點，以便在一個端點故障時使用另外一個端點。

如需有關 RMI-IIOP 負載平衡和容錯移轉的更多資訊，請參閱第 11 章。

更多資訊

如需有關規劃高可用性部署 (包括評估硬體需求、規劃網路配置和選取拓樸) 的資訊，請參閱「Sun Java System Application Server 9.1 部署規劃指南」。此手冊還提供以下概念的高階簡介：

- Application Server 元件，例如節點代理程式、網域和叢集
- 叢集中的 IIOP 負載平衡
- HADB 架構
- 訊息佇列容錯移轉

如需有關開發應用程式以運用高可用性功能優點的更多資訊，請參閱「Sun Java System Application Server 9.1 Developer's Guide」。

調校高可用性伺服器和應用程式

如需有關如何配置和調校應用程式及 Application Server 以取得高可用性最佳效能的資訊，請參閱「Sun Java System Application Server 9.1 Performance Tuning Guide」，該書說明以下主題：

- 調校持續性頻率和持續性範圍

- 對有狀態階段作業 Bean 進行檢查點操作
- 配置 JDBC 連線池
- 階段作業大小
- 調校 HADB 磁碟使用、記憶體分配、效能和作業系統配置
- 配置負載平衡器以取得最佳效能

Application Server 如何提供高可用性

Application Server 經由以下子元件和功能提供高可用性：

- [第 24 頁的「負載平衡外掛程式」](#)
- [第 25 頁的「階段作業狀態資料的儲存方式」](#)
- [第 26 頁的「高可用性叢集」](#)

負載平衡外掛程式

負載平衡外掛程式接受 HTTP / HTTPS 請求，然後將這些請求轉寄至叢集中的 Application Server 實例。如果實例出現故障，變得不可用 (因為網路故障) 或無回應，負載平衡器即會將請求重新導向至現有的可用機器。負載平衡器還可以識別出現故障的實例是否回復，並相應地重新分配負載。Application Server 為 Sun Java System Web Server、Apache Web Server 和 Microsoft Internet Information Server 提供了負載平衡器外掛程式。

透過在多台實體機器間分配工作負荷量，負載平衡器會提昇整體系統流量。負載平衡器還透過 HTTP 請求的容錯移轉提供更高的可用性。若要使 HTTP 階段作業資訊具有持續性，必須配置 HTTP 階段作業持續性。

對於簡單且無狀態的應用程式，負載平衡的叢集可能已足夠。但是，對於具有階段作業狀態的任務關鍵性應用程式，請將負載平衡叢集與 HADB 配合使用。

參與負載平衡的伺服器實例和叢集具有同質環境。這通常表示伺服器實例參照相同的伺服器配置，可以存取相同的實體資源，並部署相同的應用程式。同質環境確保故障前後，負載平衡器均始終在叢集使用中的實例間平均分配負載。

如需有關配置負載平衡和容錯移轉的資訊，請參閱[第 5 章](#)

階段作業狀態資料的儲存方式

儲存階段作業狀態資料後，即可在叢集的伺服器實例容錯移轉後回復階段作業狀態。回復階段作業狀態可繼續執行階段作業而不會遺失資訊。Application Server 為 HTTP 階段作業及有狀態的階段作業 Bean 資料提供了下列類型的高可用性儲存裝置：

- 叢集內其他伺服器上的記憶體中複製
- 高可用性資料庫

叢集內其他伺服器上的記憶體中複製

其他伺服器上的記憶體中複製可讓您輕鬆儲存階段作業狀態資料，而無需取得獨立的資料庫，例如 HADB。此類複製使用了其他伺服器上的記憶體，為 HTTP 階段作業及有狀態的階段作業 Bean 資料提供高可用性的儲存裝置。叢集伺服器實例可複製環狀拓樸的階段作業狀態。每個備份實例會在記憶體中儲存複製的資料。在其他伺服器的記憶體中複製階段作業狀態資料，即可分配階段作業。

如需使用記憶體中複製功能，需要啟用群組管理服務 (GMS)。如需有關 GMS 的更多資訊，請參閱第 147 頁的「群組管理服務」。

如果叢集中的各伺服器實例位於不同機器上，請確定符合下列必要條件：

- 為了確保 GMS 和記憶體中複製均能正確運作，各機器必須位於相同的子網路。
- 為了確保記憶體中複製正確運作，必須儘可能使叢集中所有機器的系統時鐘同步。

高可用性資料庫

備註 – Sun Java System Application Server 的 Application Server 獨立發行版本 隨附了 HADB 軟體。如需有關 Sun Java System Application Server 的可用發行軟體資訊，請參閱「Sun Java System Application Server 9.1 Installation Guide」中的「Distribution Types and Their Components」。只有企業設定檔才提供 HADB 功能。如需有關設定檔的更多資訊，請參閱「Sun Java System Application Server 9.1 管理指南」中的「用法設定檔」。

Application Server 提供高可用性資料庫 (HADB)，用於為 HTTP 階段作業和有狀態的階段作業 Bean 資料提供高可用性儲存機制。HADB 旨在使用負載平衡、容錯移轉和狀態回復功能，支援高達 99.999% 的服務和資料可用性。通常必須獨立於應用程式伺服器來配置和管理 HADB。

Application Server 不負責管理狀態的好處極多。Application Server 實例的所有時間都在充當可延伸的高效應用程式容器，將狀態複製工作委託給外部高可用性狀態服務。由於採用此鬆耦合架構，因此可以非常輕鬆地將 Application Server 實例增加至叢集或從叢集中刪除。HADB 狀態複製服務可以獨立延伸，以取得最佳可用性和效能。如果 Application Server 實例還執行複製，Java EE 應用程式的效能可能會降低，並且資源回收暫停時間也可能會變長。

如需有關規劃和設定 Application Server 安裝，以透過 HADB 實現高可用性 (包括決定硬體配置、調整大小和拓樸) 的資訊，請參閱「Sun Java System Application Server 9.1 部署規劃指南」中的「Planning for Availability」和「Sun Java System Application Server 9.1 部署規劃指南」中的第 3 章「選取拓樸」。

高可用性叢集

叢集是 Application Server 實例集合，以一個邏輯實體形式一起運作。叢集為一個或多個 Java EE 應用程式提供了執行階段環境。**高可用性叢集**將狀態複製服務與叢集和負載平衡器整合。

使用叢集可提供以下優勢：

- **高可用性** (透過允許為叢集中的伺服器實例提供容錯移轉保護來實現)。如果一個伺服器實例當機，其他伺服器實例將接管該伺服器實例正在服務的請求。
- **可延伸性** (透過允許向叢集中增加伺服器實例，從而增加系統的容量來實現)。負載平衡器外掛程式會將請求分散到叢集中的可用伺服器實例。當管理員向叢集中新增更多伺服器實例時，無需中斷服務。

叢集中的所有實例：

- 參照相同的配置。
- 具有相同的已部署應用程式集 (例如，Java EE 應用程式 EAR 檔案、Web 模組 WAR 檔案或 EJB JAR 檔案)。
- 具有相同的資源集，從而產生相同的 JNDI 名稱空間。

網域中的每一個叢集都具有唯一的名稱；此外，該名稱與所有節點代理程式名稱、伺服器實例名稱、叢集名稱和配置名稱都不重複。此名稱不能為 `domain`。您在叢集上執行的作業與在非叢集伺服器實例上執行之作業相同 (例如，部署應用程式和建立資源)。

叢集和配置

叢集的設定取自己命名的配置 (可能能夠與其他叢集共用)。不與其他伺服器實例或叢集共用配置的叢集稱之為具有**獨立配置**。依預設，此配置的名稱為 `cluster_name-config`，其中，`cluster_name` 為叢集的名稱。

與其他叢集或實例共用配置的叢集稱之為具有**共用配置**。

叢集、實例、階段作業和負載平衡

叢集、伺服器實例、負載平衡器和階段作業的相互關係如下：

- 伺服器實例不一定是叢集的一部分。但是，不屬於叢集的實例無法通過將階段作業狀態從一個實例傳送到其他實例來利用高可用性。

- 可以將叢集中的伺服器實例託管在一台或多台機器上。您可以將不同機器上的伺服器實例集合為一個叢集。
- 特定負載平衡器可以向多個叢集中的伺服器實例轉寄請求。您可以使用負載平衡器的此功能來執行線上升級，而不使服務受到損失。如需更多資訊，請參閱「Configuring Clusters」一章中的「Using Multiple Clusters for Online Upgrades Without Loss of Service」。
- 單一叢集可以從多個負載平衡器接收請求。如果叢集由多個負載平衡器提供服務，則必須以完全相同的方式在每個負載平衡器上配置叢集。
- 每個階段作業都與特定的叢集連結在一起。因此，雖然您可以在多個叢集上部署一個應用程式，但階段作業容錯移轉仍僅在一個叢集內發生。

因此，對於叢集中的伺服器實例，叢集可充當階段作業容錯移轉的安全邊界。在 Application Server 中，您可以使用負載平衡器並升級元件，而不會使服務受到任何損失。

故障後回復

- [第 27 頁的「使用 Sun Cluster」](#)
- [第 27 頁的「手動回復」](#)
- [第 29 頁的「使用 Netbackup」](#)
- [第 30 頁的「重新建立網域管理伺服器」](#)

使用 Sun Cluster

Sun Cluster 可為網域管理伺服器、節點代理程式、Application Server 實例、訊息佇列及 HADB 提供自動化的容錯移轉。如需更多資訊，請參閱「Sun Cluster Data Service for Sun Java System Application Server Guide for Solaris OS」。

使用標準乙太網路互連及部分的 Sun Cluster 產品。Java ES 包含這項功能。

手動回復

您可以使用各種技術手動回復個別子元件：

- [第 28 頁的「回復網域管理伺服器」](#)
- [第 28 頁的「回復節點代理程式和伺服器實例」](#)
- [第 28 頁的「回復負載平衡器及 Web 伺服器」](#)
- [第 29 頁的「回復訊息佇列」](#)
- [第 29 頁的「回復 HADB」](#)

回復網域管理伺服器

失去網域管理伺服器 (DAS) 只會影響到管理作業。即使無法連線至 DAS，Application Server 叢集和應用程式仍會如常地繼續運作。

可用下列任意方法回復 DAS：

- 定期執行 `asadmin` 備份指令就能取得定期快照。於發生硬體故障之後，在擁有相同網路識別的新機器上安裝應用程式伺服器並執行 `asadmin`，以從先前建立的備份進行復原。如需更多資訊，請參閱第 30 頁的「重新建立網域管理伺服器」。
- 將網域安裝和配置放在共用且牢固的檔案系統上 (例如 NFS)。若主要 DAS 機器故障，擁有相同 IP 位址的第二部機器就會啟動，並以手動操作或使用者提供的自動化程序接手運作。Sun Cluster 採用類似的方式讓 DAS 有容錯移轉功能。
- 壓縮 Application Server 安裝及網域根目錄。將壓縮檔復原到新機器上，並為其指定相同的網路識別。如果您是採用檔案型的安裝，這可能是最簡單的做法。
- 以 DAS 備份進行復原。請參閱 AS8.1 UR2 修補程式 4 指示。

回復節點代理程式和伺服器實例

節點代理程式和伺服器實例有兩種回復方式。

保存備份壓縮檔案。目前沒有針對節點代理程式和伺服器實例提供專屬的備份指令。只要建立一個壓縮檔，其中包含節點代理程式目錄的內容即可。發生故障之後，可將儲存的備份解壓縮至擁有相同主機名稱和 IP 位址的新機器上，並且使用相同的安裝目錄位置、作業系統等等。該機器上必須擁有檔案型安裝、套裝軟體型安裝或復原的備份影像。

手動回復。使用的新主機必須擁有相同的 IP 位址。

1. 在該機器上安裝 Application Server 節點代理程式位元。
2. 請參閱 AS8.1 UR2 修補程式 4 安裝指示。
3. 重新建立節點代理程式。您不需要建立任何伺服器實例。
4. 同步化處理會從 DAS 複製配置及資料，並做更新。

回復負載平衡器及 Web 伺服器

目前沒有單獨針對 Web 伺服器配置提供專屬備份指令。只要壓縮 Web 伺服器安裝目錄即可。發生故障之後，將儲存的備份解壓縮至擁有相同網路識別的新機器上。若新機器擁有不同的 IP 位址，則更新 DNS 伺服器或路由器。

備註 – 這裡假設您已經事先重新安裝 Web 伺服器或從影像復原 Web 伺服器。

負載平衡器外掛程式 (plugins 目錄) 及配置位於 Web 伺服器安裝目錄中，此目錄通常是 `/opt/SUNWwbsvr.web-install/web-instance/config` 目錄包含 `loadbalancer.xml` 檔案。

回復訊息佇列

訊息佇列 (MQ) 配置和資源儲存於 DAS，並且可和實例同步化。任何其他資料和配置資訊則位於 MQ 目錄中，這些目錄通常位於 `/var/imq` 下，因此請視需求備份及復原這些目錄。新機器上必須已經有 MQ 安裝。復原機器時，務必依常態啟動 MQ 代理程式。

回復 HADB

備註 – Sun Java System Application Server 的 Application Server 獨立發行版本 隨附了 HADB 軟體。如需有關 Sun Java System Application Server 的可用發行軟體資訊，請參閱「Sun Java System Application Server 9.1 Installation Guide」中的「Distribution Types and Their Components」。只有企業設定檔才提供 HADB 功能。如需有關設定檔的更多資訊，請參閱「Sun Java System Application Server 9.1 管理指南」中的「用法設定檔」。

如果您有兩個使用中 HADB 節點，則可以 (在別的機器上) 配置兩個備用節點，以便在發生故障時能夠接手。這種方式比較簡潔，因為備份及復原 HADB 可能會復原到舊的階段作業。

如需有關建立含有備用節點資料庫的資訊，請參閱第 62 頁的「[建立資料庫](#)」。如需有關對資料庫增加備援節點的資訊，請參閱第 84 頁的「[增加節點](#)」。如果回復及自我修復都失敗，備用節點即會自動接手。

使用 Netbackup

備註 – Sun QA 尚未測試過此程序。

使用 Veritas Netbackup 儲存每部機器的影像。若是 BPIP，則備份含有 Web 伺服器 and 應用程式伺服器的四部機器。

每部復原的機器都和原來機器使用相同配置，例如相同的主機名稱、IP 位址等等。

若是如應用程式伺服器之類的檔案型產品，只需備份及復原相關目錄。但是如 Web 伺服器影像之類的套裝軟體型安裝，則必須備份及復原整部機器。套裝軟體會安裝至 Solaris 套裝軟體資料庫。因此，若只備份目錄並於稍後復原至新系統，將會造成套裝軟體資料庫中沒有「已部署」Web 伺服器的資料。這會導致未來無法安裝修補程式或進行升級。

請勿手動複製及復原 Solaris 套裝軟體資料庫。別的可行做法是在安裝如 Web 伺服器之類的元件後，備份機器的影像。此影像稱為基準線 tar 檔案。當您變更 Web 伺服器時，只需備份如 `/opt/SUNWwbsvr` 下的這些目錄。復原時，先復原基準線 tar 檔案，再覆蓋做過修改的 Web 伺服器目錄。您也同樣可以對 MQ (BPIP 的套裝軟體型安裝) 使用這個程序。如果您在原始機器上進行升級或安裝修補程式，務必要建立新的基準線 tar 檔案。

若包含 DAS 的機器發生故障，在機器復原之前將無法使用 DAS。

DAS 是中央儲存庫。當您復原伺服器實例並重新啟動伺服器實例時，伺服器實例只會和 DAS 中的資訊同步化。因此，所有變更都必須透過 `asadmin` 或管理主控台執行。

HADB 的每日備份影像不一定可以用，因為這類影像可能包含舊的應用程式階段作業狀態。

重新建立網域管理伺服器

若主控網域管理伺服器 (DAS) 的機器發生故障，且先前已備份 DAS，則可重建 DAS。若要重新建立 DAS 的有效副本，您必須擁有：

- 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上安裝時相同。
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。
若要啟動指令行互動模式，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。
兩台機器的架構與安裝路徑完全相同時 (即兩台機器使用相同的 `as-install` 和 `domain-root-dir`)，才支援復原備份的網域。
- 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 將 ZIP 檔案復原到第三台機器。

```
asadmin restore-domain --filename domain-root-dir/sjsas_backup_v00001.zip
--clienthostname machine3 domain1
```

備註 – 只要指定 `--clienthostname` 選項，就不需要修改 `jmx-connector` 元素在 `domain.xml` 檔案中的 `client-hostname` 特性。

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為：`drwx-----` (or 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 在第三台機器的 `domain-root-dir/domain1/config/domain.xml` 檔案中，更新 `jms-service` 元素的 `host` 屬性值。

以下是此屬性的原始設定：

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

依下列方式修改此屬性的設定：

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

6 在 `machine3` 上啟動復原的網域：

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

DAS 會連絡所有執行中的節點代理程式，並為節點代理程式提供 DAS 的連絡資訊。節點代理程式會使用此資訊連絡 DAS。

7 對於重新啟動 DAS 時未執行的任何節點代理程式，請在 `machine2` 上變更 `as-install/nodeagents/nodeagent/agent/config/das.properties` 中的 `agent.das.host` 特性值。重新啟動 DAS 時，正在執行的節點代理程式不需要執行此步驟。

8 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。

安裝和設定高可用性資料庫

備註 – Sun Java System Application Server 的 Application Server 獨立發行版本 隨附了 HADB 軟體。如需有關 Sun Java System Application Server 的可用發行軟體資訊，請參閱「Sun Java System Application Server 9.1 Installation Guide」中的「Distribution Types and Their Components」。只有 企業 設定檔才提供 HADB 功能。如需有關設定檔的更多資訊，請參閱「Sun Java System Application Server 9.1 管理指南」中的「用法設定檔」。

本小節包含以下主題：

- [第 33 頁的「準備 HADB 設定」](#)
- [第 40 頁的「安裝」](#)
- [第 42 頁的「設定高可用性」](#)
- [第 44 頁的「升級 HADB」](#)

準備 HADB 設定

本小節論述以下主題：

- [第 34 頁的「先決條件和限制」](#)
- [第 34 頁的「配置網路備援」](#)
- [第 37 頁的「配置共用記憶體和信號」](#)
- [第 39 頁的「同步化系統時鐘」](#)

執行上述作業後，請參閱[第 3 章](#)。

如需有關 HADB 的最新資訊，請參閱「Sun Java System Application Server 9.1 版本說明」。

先決條件和限制

開始設定及配置 HADB 之前，請確定您的網路和硬體環境都符合「Sun Java System Application Server 9.1 版本說明」中指出的需求。此外特定的檔案系統還有限制，例如對 Veritas 的限制。如需更多資訊，請參閱「版本說明」。

HADB 在建立並連結至其共用記憶體區段時，會使用詳細共用記憶體 (SHM_SHARE_MMU 旗標)。此旗標主要用於將共用記憶體區段鎖定至實體記憶體，並防止對其進行頁面移出。因此 HADB 的共用記憶體會鎖定在實體記憶體內，很容易影響低階機器上的安裝。Application Server 與 HADB 並置時，請確保您擁有建議的記憶體容量。

配置網路備援

配置備援網路後，即使單一網路發生故障，HADB 也仍然可用。您可以使用以下兩種方式配置備援網路：

- 在 Solaris 9 上，設定網路多重路徑。
- 在 Windows Server 2003 以外的所有平台上配置雙網路。

設定網路多重路徑

設定網路多重路徑之前，請參閱「IP Network Multipathing Administration Guide」中的「管理網路多重路徑」一節。

▼ 配置已使用 IP 多重路徑的 HADB 主機電腦

1 設定網路介面故障偵測時間。

為使 HADB 能夠正確支援多重路徑容錯移轉，網路介面故障偵測時間不得超過一秒 (1000 毫秒)，如 /etc/default/mpathd 中的 FAILURE_DETECTION_TIME 參數所指定。如果此參數的原始值較高，請編輯檔案並將此參數值變更為 1000：

```
FAILURE_DETECTION_TIME=1000
```

若要使變更生效，請使用以下指令：

```
pkill -HUP in.mpathd
```

2 設定 IP 位址以與 HADB 配合使用。

如「IP Network Multipathing Administration Guide」中所述，多重路徑必須將實體網路介面分組為多重路徑介面群組。在此群組中的每個實體介面均具有兩個相關的 IP 位址：

- 用於傳輸資料的實體介面位址。
- 僅供 Solaris 內部使用的測試位址。

使用 `hadbm create --hosts` 時，請從多重路徑群組中僅指定一個實體介面位址。

範例 2-1 設定多重路徑

假設具有兩個名為 `host1` 和 `host2` 的主機電腦。如果它們分別具有兩個實體網路介面，則將這兩個介面設定為多重路徑群組。在每台主機上均執行 `ifconfig -a`。

`host1` 上的輸出為：

```
bge0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4>
mtu 1500 index 5 inet 129.159.115.10 netmask ffffffff00 broadcast 129.159.115.255
groupname mp0

bge0:1: flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER>
mtu 1500 index 5 inet 129.159.115.11 netmask ffffffff00 broadcast 129.159.115.255

bge1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4>
mtu 1500 index 6 inet 129.159.115.12 netmask ffffffff00 broadcast 129.159.115.255
groupname mp0

bge1:1: flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER>
mtu 1500 index 6 inet 129.159.115.13 netmask ff000000 broadcast 129.159.115.255
```

`host2` 上的輸出為：

```
bge0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4>
mtu 1500 index 3 inet 129.159.115.20 netmask ffffffff00 broadcast 129.159.115.255
groupname mp0

bge0:1: flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER>
mtu 1500 index 3 inet 129.159.115.21 netmask ff000000 broadcast 129.159.115.255

bge1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4>
mtu 1500 index 4 inet 129.159.115.22 netmask ffffffff00 broadcast 129.159.115.255
groupname mp0

bge1:1: flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER>
mtu 1500 index 4 inet 129.159.115.23 netmask ff000000 broadcast 129.159.115.255
```

在此範例中，兩台主機上的實體網路介面都列出在 `bge0` 和 `bge1` 之後。在 `bge0:1` 和 `bge1:1` 之後列出的介面是多重路徑測試介面 (在 `ifconfig` 輸出中標示為 `DEPRECATED`)，如「IP Network Multipathing Administration Guide」中所述。

若要在此環境中設定 HADB，請從每個主機中選取一個實體介面。在此範例中，HADB 在 `host1` 中使用 IP 位址 129.159.115.10，在 `host2` 中使用 129.159.115.20。若要針對每台主機都使用一個資料庫節點建立一個資料庫，請使用指令 `hadbm create --hosts`。例如：

```
hadbm create --hosts 129.159.115.10,129.159.115.20
```

若要在每個主機上使用兩個資料庫節點建立資料庫，請使用以下指令：

```
hadbm create --hosts 129.159.115.10,129.159.115.20,
129.159.115.10,129.159.115.20
```

在這兩種情況下，您必須在 `host1` 和 `host2` 上使用獨立參數配置代理程式，以指定代理程式應使用機器上的哪個介面。因此，在 `host1` 上使用：

```
ma.server.mainternal.interfaces=129.159.115.10
```

在 `host2` 上使用：

```
ma.server.mainternal.interfaces=129.159.115.20
```

如需有關 `ma.server.mainternal.interfaces` 變數的資訊，請參閱第 56 頁的「配置檔案」。

配置雙網路

若要啟用 HADB 以容許單一網路故障，請使用 IP 多重路徑 (如果受諸如 Solaris 之類的作業系統支援)。請勿在 Windows Server 2003 上配置具有雙網路的 HADB—作業系統在雙網路狀態下無法正常工作。

如果未將作業系統配置為使用 IP 多重路徑，且 HADB 主機配備兩個 NIC，則您可以將 HADB 配置為使用雙網路。對於每台主機，每個網路介面卡 (NIC) 的 IP 位址均必須位於獨立的 IP 子網路上。

在資料庫中，所有節點均必須連線至單一網路，或所有節點均必須連線至兩個網路。

備註 – 必須將子網路之間的路由器配置為可在子網路之間轉寄 UDP 多重播送訊息。

建立 HADB 資料庫時，請使用 `--hosts` 選項為每個節點指定兩個 IP 位址或主機名稱：每個 NIC IP 位址對應一個 IP 位址或主機名稱。對於每個節點，第一個 IP 位址位於 `net-0` 上，第二個 IP 位址位於 `net-1` 上。語法如下所示，其中同一節點的主機名稱以加號 (+) 分隔：

```
--hosts=node0net0name+node0net1name
,node1net0name+node1net1name
,node2net0name+node2net1name
, ...
```

例如，以下引數可建立兩個節點，每個節點均具有兩個網路介面。以下主機選項用於建立這些節點：

```
--hosts 10.10.116.61+10.10.124.61,10.10.116.62+10.10.124.62
```

因此，node0 的網路位址為

- 10.10.116.61 和 10.10.124.61
- node1 的網路位址為 10.10.116.62 和 10.10.124.62

請注意，10.10.116.61 和 10.10.116.62 位於同一子網路上，而 10.10.124.61 和 10.10.124.62 位於同一子網路上。

在此範例中，管理代理程式必須使用同一子網路。因此，舉例來說，必須將配置變數 `ma.server.maininternal.interfaces` 設定為 10.10.116.0/24。此設定可同時用於此範例中的兩個代理程式。

配置共用記憶體和信號

在安裝 HADB 之前，必須先配置共用記憶體和信號。該程序視您的作業系統而定。

如果您的主機除了執行 HADB 外還執行其他應用程式，請計算這些應用程式使用的共用記憶體和信號量，將結果增加至 HADB 需要的值。本小節建議的值足以在每台主機上執行多達 6 個 HADB 節點。如果您執行的 HADB 節點超過 6 個，或主機上執行的應用程式需要額外的共用記憶體和信號，只要調高這個值即可。

如果信號的數目過低，可能會導致 HADB 故障並顯示以下錯誤訊息：**裝置上無可用空間**。此情況可能會在啟動資料庫時或執行階段發生。

▼ 在 Solaris 上配置共用記憶體和信號

由於信號是一種全域性的作業系統資源，因此配置需要顧及主機上執行的所有程序，而不只是 HADB 而已。在 Solaris 上，可以編輯 `/etc/system` 檔案來配置信號設定。

- 1 以超級使用者身份登入。
- 2 配置共用記憶體。
 - 設定 `shminfo_shmmax` 以指定主機上單一共用記憶體區段的最大大小。將這個值設定為 HADB 主機電腦上安裝的 RAM 總數，值是以十六進位值表示，不得超過 2 GB。例如，以 2 GB 的 RAM 來說，請在 `/etc/system` 檔案中將值設定如下：

```
set shmsys:shminfo_shmmax=0x80000000
```

備註 – 若要確定主機電腦上的記憶體量，請使用以下指令：

```
prtconf | grep Memory
```

- 在 Solaris 8 或更低的版本中，請設定 `shminfo_shmseg`，這是每個程序可以附加的最大共用記憶體區段數。請將這個值設為每台主機節點數的 6 倍。若每台主機的節點數不超過 6 個，請在 `/etc/system` 檔案中增加以下指令：

```
set shmsys:shminfo_shmseg=36
```

在 Solaris 9 以及更高版本中，`shmsys:shminfo_shmseg` 已停用。

- 設定 `shminfo_shmmni`，這是全系統的最大共用記憶體區段數。由於每個 HADB 節點都會配置 6 個共用記憶體區段，因此 HADB 需要的記憶體值必須至少是每個主機節點數的 6 倍。在 Solaris 9 中，若每台主機的節點數不超過 6 個，就不需要更改預設值。

3 配置信號。

請在 `/etc/system` 檔案中檢查如下的信號配置項目：

```
set semsys:seminfo_semmni=10
set semsys:seminfo_semmns=60
set semsys:seminfo_semmnu=30
```

如果有上述項目存在，請依以下指示調高其值。

如果 `/etc/system` 檔案不包含這些項目，請將它們增加至檔案的結尾處：

- 設定 `seminfo_semmni`，這是信號識別碼的最大數目。每個 HADB 節點都需要有一個信號識別碼。在 Solaris 9 中，若每台主機的節點數不超過 6 個，就不需要更改預設值。例如：

```
set semsys:seminfo_semmni=10
```

- 設定 `seminfo_semmns`，這是全系統的最大信號數目。每個 HADB 節點都必須有 8 個信號。在 Solaris 9 中，若每台主機的節點數不超過 6 個，就不需要更改預設值。例如：

```
set semsys:seminfo_semmns=60
```

- 設定 `seminfo_semmnu`，這是系統內還原結構的最大數目。每個連線都需要一個還原結構 (配置變數 `NumberOfSessions`，預設值 100)。若每台主機的節點數不超過 6 個，將其設為 600：

```
set semsys:seminfo_semmnu=600
```

4 重新開機。

▼ 在 Linux 上配置共用記憶體

您必須在 Linux 上配置共用記憶體設定。預設的信號設定則不需要調整。

1 以超級使用者身份登入。

2 編輯 `/etc/sysctl.conf` 檔案。

若使用的是 Redhat Linux，您也可以修改 `sysctl.conf` 以設定核心參數。

3 如下所示，設定 `kernel.shmax` 和 `kernel.shmall` 的值：

```
echo MemSize > /proc/sys/shmmax
echo MemSize > /proc/sys/shmall
```

其中 `MemSize` 是位元組數。

`kernel.shmax` 參數定義共用記憶體區段的最大大小 (以位元組為單位)。`kernel.shmall` 參數可設定在系統上一次使用的共用記憶體總大小 (以頁為單位)。將這些參數的值設定為機器上實體記憶體的大小。將值指定為十進位位元組數。

例如，若要將兩個值都設為 2GB，請使用以下指令：

```
echo 2147483648 > /proc/sys/kernel/shmmax
echo 2147483648 > /proc/sys/kernel/shmall
```

4 使用以下指令重新開機：

```
sync; sync; reboot
```

用於 Windows 的程序

Windows 不需要任何特殊的系統設定。但是，如果您要使用現有的 J2SE 安裝，請將 `JAVA_HOME` 環境變數設定至安裝 J2SE 的位置。

同步化系統時鐘

您必須在 HADB 主機上同步化時鐘，因為 HADB 使用基於系統時鐘的時間戳記。HADB 使用系統時鐘來管理逾時和為記錄至歷史檔案中的事件標註時間戳記。對於疑難排解，由於 HADB 是分散式系統，所以必須同時分析所有歷史檔案。因此，同步化所有主機的時鐘非常重要。

請勿在正在執行的 HADB 系統上調整系統時鐘。否則會導致作業系統或其他軟體元件中出現問題，進而導致諸如 HADB 節點掛機或重新啟動之類的問題。向後調整時鐘會導致某些 HADB 伺服器程序掛機 (由於時鐘已被調整)。

若要同步化時鐘，請執行以下作業：

- 在 Solaris 上，使用 `xntpd` (網路時間協定常駐程式)。
- 在 Linux 上，使用 `ntpd`。
- 在 Windows 上，使用 `NTPTIME` (位於 Windows 上)

如果 HADB 偵測到時鐘調整大於一秒，則會將其記錄至節點歷史檔案，例如：

```
NSUP INF 2003-08-26 17:46:47.975 Clock adjusted.
Leap is +195.075046 seconds.
```

安裝

一般而言，您可以在與 Application Server 相同的系統上安裝 HADB (同位拓樸)，也可在獨立主機上安裝 HADB (獨立階層拓樸)。如需有關這兩個選項的更多資訊，請參閱「Sun Java System Application Server 9.1 部署規劃指南」中的第 3 章「選取拓樸」。

您必須安裝 HADB 管理用戶端，才能使用 `asadmin configure-ha-cluster` 指令設定高可用性。當使用 Java Enterprise System 安裝程式時，即使節點要安裝在獨立階層上，您也必須安裝整個 HADB 實例才能安裝管理用戶端。

HADB 安裝

在單一 CPU 或雙 CPU 系統上，如果系統擁有至少 2 GB 的記憶體，則您可以同時安裝 HADB 和 Application Server。否則，請在獨立系統上安裝 HADB，或使用其他硬體。若要使用 `asadmin configure-ha-cluster` 指令，必須同時安裝 HADB 和 Application Server。

每個 HADB 節點需要 512 MB 的記憶體，因此機器需要 1 GB 的記憶體來執行兩個 HADB 節點。如果機器的記憶體不足，請將每個節點設定在不同機器上。例如，您可以在以下系統上安裝兩個節點：

- 兩個單一 CPU 系統，每個具有 512 MB 至 1 GB 的記憶體
- 一個單一 CPU 或雙 CPU 系統，具有 1 GB 至 2 GB 的記憶體

您可以使用 Java Enterprise System 安裝程式或 Application Server 獨立安裝程式來安裝 HADB。在每個安裝程式中，從 [元件選取] 頁面中選擇要安裝 HADB (在 Java ES 中稱為高可用性階段作業儲存) 的選項。在您的主機上完成安裝。如果您使用 Application Server 獨立安裝程式，並選擇兩台獨立機器來執行 HADB，則必須在兩台機器上選擇相同的安裝目錄。

預設安裝目錄

在本手冊中，`HADB_install_dir` 表示安裝 HADB 的目錄。預設安裝目錄取決於您是否將 HADB 做為 Java Enterprise System 的一部分進行安裝。對於 Java Enterprise System，預設安裝目錄為 `/opt/SUNWhadb/4`。對於獨立的 Application Server 安裝程式，預設安裝目錄為 `/opt/SUNWappserver/hadb/4`。

節點管理員程序權限

節點管理員程序 (NSUP) 透過彼此交換「I'm alive」訊息來確保 HADB 的可用性。NSUP 可執行檔必須具有超級使用者權限才能盡快回應。`clu_nsup_srv` 程序不會消耗大量的 CPU 資源，而且其佔用空間較小，因此，使用即時優先權執行該程序不會影響效能。

備註 – Java Enterprise System 安裝程式可自動正確設定 NSUP 權限，因此您無需採取進一步動作。但是，如果您使用獨立 Application Server (非超級使用者) 安裝程式，則必須在建立資料庫之前手動設定該權限。

權限不足的徵兆

如果 NSUP 不具有適當的權限，您可能會注意到資源不足的徵兆，例如：

- HADB 歷史檔案中顯示警告「程序已被阻斷 *n* 秒」，然後出現錯誤的網路分割以及節點重新啟動。
- 作業事件中斷和其他異常。

限制

如果 NSUP 無法設定即時優先權，則 `errno` 在 Solaris 和 Linux 上被設定為 `EPERM`。在 Windows 上，會發出警告「無法設定即時優先權」。該錯誤會寫入 `ma.log` 檔案，並且程序會在沒有即時優先權的情況下繼續執行。

在以下情況中無法設定即時優先權：

- HADB 安裝在 Solaris 10 非全域區域中
- 在 Solaris 10 中撤銷 `PRIV_PROC_LOCK_MEMORY` (允許程序在實體記憶體中鎖定頁面) 和/或 `PRIV_PROC_PRIOCNTRL` 權限
- 使用者關閉 `setuid` 權限
- 使用者將軟體做為 `tar` 檔案 (應用程式伺服器非超級使用者安裝選項) 安裝

▼ 授予節點管理員程序超級使用者權限

- 1 以超級使用者身份登入。
- 2 將工作目錄變更為 `HADB_install_dir/lib/server`。
NSUP 可執行檔為 `clu_nsup_srv`。

- 3 使用以下指令設定檔案的 `suid` 位元：

```
chmod u+s clu_nsup_srv
```

- 4 使用以下指令將檔案的所有權設定為超級使用者：

```
chown root clu_nsup_srv
```

該指令會以超級使用者的身份啟動 `clu_nsup_srv` 程序，並可讓程序為其自身提供即時優先權。

若要避免對安全性產生任何影響，應在程序啟動後立即設定即時優先權，這樣，一旦優先權變更，程序會返回至有效的 UID。其他 HADB 程序使用一般優先權執行。

設定高可用性

本小節說明建立具有高可用性的叢集以及測試 HTTP 階段作業持續性的步驟。

本小節論述以下主題：

- 第 42 頁的「準備系統使其具有高可用性」
- 第 42 頁的「啓動 HADB 管理代理程式」
- 第 43 頁的「配置叢集以使其具有高可用性」
- 第 43 頁的「配置應用程式以使其具有高可用性」
- 第 43 頁的「重新啓動叢集」
- 第 44 頁的「重新啓動 Web 伺服器」
- 第 44 頁的「清除做為負載平衡器執行的 Web 伺服器實例」

▼ 準備系統使其具有高可用性

- 1 安裝 Application Server 實例和負載平衡外掛程式。

如需更多資訊，請參閱「*Java Enterprise System 安裝指南*」(如果您使用 Java ES) 或「Sun Java System Application Server 9.1 Installation Guide」(如果您使用獨立的 Application Server 安裝程式)。

- 2 建立 Application Server 網域和叢集。

如需有關如何建立網域的資訊，請參閱「Sun Java System Application Server 9.1 管理指南」中的「建立網域」。如需有關如何建立叢集的資訊，請參閱第 149 頁的「[建立叢集](#)」。

- 3 安裝並配置 Web 伺服器軟體。

- 4 設定並配置負載平衡。

如需更多資訊，請參閱第 120 頁的「[設定 HTTP 負載平衡](#)」。

啓動 HADB 管理代理程式

管理代理程式 `ma` 在 HADB 主機上執行管理指令，並透過在 HADB 節點管理員程序失敗時將其重新啓動，來確保其可用性。

可以使用以下兩種方法啓動管理代理程式：

- 做為服務啓動，用於生產。請參閱第 50 頁的「[將管理代理程式做為服務啓動](#)」。為確保管理代理程式的可用性，請確定其可在系統重新開機時自動重新啓動。請參閱第 51 頁的「[確保自動重新啓動管理代理程式](#)」。
- 做為常規程序(在主控台模式中)啓動，用於試用、測試或開發。請參閱第 53 頁的「[在主控台模式下啓動管理代理程式](#)」。

對於每種啟動方式，程序會視您使用的是 Java Enterprise System 還是獨立的應用程式伺服器而有所不同。

配置叢集以使其具有高可用性

開始本小節之前，您必須已建立一個或多個 Application Server 叢集。如需有關如何建立叢集的資訊，請參閱第 149 頁的「建立叢集」。

從執行網域管理伺服器的電腦上，使用以下指令將叢集配置為使用 HADB：

```
asadmin configure-ha-cluster --user admin --hosts hadb_hostname1,hadb_hostname2  
[,...] --devicesize 256 clusterName
```

以每一部執行 HADB 的電腦主機名稱取代 *hadb_hostname1*、*hadb_hostname2* (依此類推)，並以叢集名稱取代 *clusterName*。例如：

```
asadmin configure-ha-cluster --user admin --hosts host1,host2,host1,host2  
--devicesize 256 cluster1
```

此範例在每部電腦上各建立兩個節點，即使在 HADB 容錯移轉時，仍具有高度可用性。請注意，接在 `--hosts` 選項後面的主機名稱順序極為重要，因此上個範例就和 `--hosts host1,host1,host2,host2` 不同。

如果僅使用一台機器，則必須提供主機名稱兩次。我們建議在生產環境中使用多台機器。

配置應用程式以使其具有高可用性

在管理主控台中，在 [應用程式] > [企業應用程式] 下選取應用程式。設定 [已啓用的可用性]，然後按一下 [儲存]。

重新啟動叢集

若要在管理主控台中重新啟動叢集，請選擇 [叢集] > [*cluster-name*]。按一下 [停止實例]。實例停止後，按一下 [啟動實例]。

或者，使用以下 `asadmin` 指令：

```
asadmin stop-cluster --user admin cluster-name  
asadmin start-cluster --user admin cluster-name
```

如需有關這些指令的更多資訊，請參閱 `stop-cluster(1)` 和 `start-cluster(1)`。

重新啟動 Web 伺服器

若要重新啟動 Web 伺服器，請鍵入以下 Web 伺服器指令：

```
web_server_root/https-hostname/reconfig
```

使用 Web 伺服器根目錄替代 *web_server_root*，並使用主機電腦名稱替代 *hostname*。

▼ 清除做為負載平衡器執行的 Web 伺服器實例

- 1 刪除負載平衡器配置：

```
asadmin delete-http-lb-ref --user admin --config MyLbConfig FirstCluster
```

```
asadmin delete-http-lb-config --user admin MyLbConfig
```

- 2 如果已建立新的 Web 伺服器實例，則可以透過執行以下作業刪除該實例：

- a. 登入 Web 伺服器的管理主控台。

- b. 停止實例。

刪除實例。

升級 HADB

HADB 旨在提供不會因升級軟體而中斷的「Always On」服務。本小節說明如何在使資料庫處於離線狀態或不損失任何可用性的情況下，升級至新的 HADB 版本。此程序稱為「線上升級」。

以下小節說明如何升級 HADB 安裝：

- 第 45 頁的「將 HADB 升級至較新的版本」
- 第 45 頁的「註冊 HADB 套裝軟體」
- 第 46 頁的「取消註冊 HADB 套裝軟體」
- 第 47 頁的「替代管理代理程式啟動程序檔」
- 第 47 頁的「驗證 HADB 升級」

▼ 將 HADB 升級至較新的版本

- 1 安裝新的 HADB 版本。
- 2 如第 45 頁的「註冊 HADB 套裝軟體」中所述，註冊新的 HADB 版本。
在 HADB 管理網域中註冊 HADB 套裝軟體可輕鬆升級或變更 HADB 套裝軟體。管理代理程式可追蹤套裝軟體的位置，以及網域中主機的版本資訊。預設套裝軟體名稱爲以 V 開頭並包含 hadbm 程式之版本編號的字串。
- 3 變更資料庫使用的套裝軟體。
請輸入以下指令：

```
hadbm set PackageName=package
```


其中 *package* 爲新 HADB 套裝軟體的版本編號。
- 4 如第 46 頁的「取消註冊 HADB 套裝軟體」中所述，取消註冊現有的 HADB 安裝。
- 5 如有必要，請替代理程式啟動程序檔。
如需更多資訊，請參閱第 47 頁的「替代理程式啟動程序檔」。
- 6 如第 47 頁的「驗證 HADB 升級」中所述，驗證結果。
- 7 (可選擇) 移除舊版 HADB 的二進位檔案。
驗證 HADB 已正確升級後，就可以刪除舊版 HADB 套裝軟體。

註冊 HADB 套裝軟體

使用 `hadbm registerpackage` 指令註冊在管理網域中的主機上安裝的 HADB 套裝軟體。也可在使用 `hadbm create` 建立資料庫時，註冊 HADB 套裝軟體。

使用 `hadbm registerpackage` 指令之前，請確保：所有管理代理程式均已配置，並在主機清單中的所有主機上執行；該管理代理程式的儲存庫可用於更新；未使用相同的套裝軟體名稱註冊任何套裝軟體。

指令語法爲：

```
hadbm registerpackage --packagepath=path [--hosts=hostlist] [--adminpassword=password | --adminpasswordfile=file] [--agent=maurl] [[package-name]]
```

package-name 運算元爲套裝軟體的名稱。

下表說明了特殊的 `hadbm registerpackage` 指令選項。請參閱第 58 頁的「安全性選項」和第 59 頁的「一般選項」，以取得有關其他指令選項的說明。

表 2-1 hadbm registerpackage 選項

選項	說明
--hosts= <i>hostlist</i>	主機清單，以逗號分隔或置於雙引號中並以空格分隔。
-H	
--packagepath= <i>path</i>	HADB 軟體套裝軟體的路徑。
-L	

例如，以下指令在主機 `host1`、`host2` 和 `host3` 上註冊套裝軟體 `v4`：

```
hadbm registerpackage
--packagepath=hadb_install_dir/SUNWHadb/4.4
--hosts=host1,host2,host3 v4
```

回應為：

```
Package successfully registered.
```

如果您省略 `--hosts` 選項，則該指令會在網域中所有已啓用的主機上註冊套裝軟體。

取消註冊 HADB 套裝軟體

使用 `hadbm unregisterpackage` 指令移除在管理網域中註冊的 HADB 套裝軟體。

開始使用 `hadbm unregisterpackage` 指令之前，請確定：

- 已在 `hostlist` 中列出的所有主機上配置好所有管理代理程式，並可運作。
- 管理代理程式的儲存庫可進行更新。
- 新的 HADB 套裝軟體會註冊在管理網域內
- 現有的資料庫未配置為在即將取消註冊的套裝軟體中執行。

指令語法為：

```
hadbm unregisterpackage
--hosts=hostlist
[--adminpassword=password | --adminpasswordfile= file]
[--agent= maurl]
[package-name ]
```

package-name 運算元為套裝軟體的名稱。

請參閱第 45 頁的「註冊 HADB 套裝軟體」，以取得有關 `--hosts` 選項的說明。如果您省略 `--hosts` 選項，則主機清單會將在其中註冊套裝軟體的已啓用主機做為預設主機。請參閱第 58 頁的「安全性選項」和第 59 頁的「一般選項」，以取得有關其他指令選項的說明。

範例 2-2 取消註冊 HADB 的範例

若要從網域中特定主機上取消註冊套裝軟體 v4，請執行以下指令：

```
hadbm unregisterpackage --hosts=host1,host2,host3 v4
```

回應為：

```
Package successfully unregistered.
```

替代管理代理程式啟動程序檔

安裝新的 HADB 版本時，可能需要替代 `/etc/init.d/ma-initd` 中的管理代理程式啟動程序檔。檢查檔案 `HADB_install_dir/lib/ma-initd` 的內容。如果其內容與舊的 `ma-initd` 檔案不同，請使用新檔案替代舊檔案。

▼ 驗證 HADB 升級

請依照以下程序，驗證 HADB 是否已正確升級：

1 確認執行中的 HADB 程序之版本。

在所有 HADB 節點上輸入下列指令，以顯示 HADB 的版本：

```
new-path/bin/ma -v
```

```
new-path/bin/hadbm -v
```

其中 *new-path* 是指向新的 HADB 安裝之路徑。

結果應該顯示出新的 HADB 版本編號。

2 確認資料庫在執行中。

輸入以下指令：

```
new-path/bin/hadbm status -n
```

如果已成功升級，結果就會顯示所有 HADB 節點都處於 running 狀態。

3 確保使用 HADB 的產品都已將其配置設定變更為指向新的 HADB 路徑。

4 針對使用 HADB 的產品執行所有升級測試。

管理高可用性資料庫

備註 – Sun Java System Application Server 的 Application Server 獨立發行版本 隨附了 HADB 軟體。如需有關 Sun Java System Application Server 的可用發行軟體資訊，請參閱「Sun Java System Application Server 9.1 Installation Guide」中的「Distribution Types and Their Components」。只有 企業 設定檔才提供 HADB 功能。如需有關設定檔的更多資訊，請參閱「Sun Java System Application Server 9.1 管理指南」中的「用法設定檔」。

本章說明 Sun Java System Application Server 環境中的高可用性資料庫 (HADB)，並說明如何配置和管理 HADB。建立和管理 HADB 之前，必須首先確定系統拓樸，然後在不同的機器上安裝 HADB 軟體。

本章說明以下主題：

- 第 49 頁的「使用 HADB 管理代理程式」
- 第 57 頁的「使用 `hadbm` 管理指令」
- 第 61 頁的「配置 HADB」
- 第 74 頁的「管理 HADB」
- 第 82 頁的「延伸 HADB」
- 第 88 頁的「監視 HADB」
- 第 94 頁的「維護 HADB 機器」

使用 HADB 管理代理程式

管理代理程式 `ma` 在 HADB 主機上執行管理指令。管理代理程式還可在 HADB 節點管理程序失敗時將其重新啟動，從而確保 HADB 節點管理程序的可用性。

- 第 50 頁的「啟動管理代理程式」
- 第 54 頁的「管理代理程式指令語法」
- 第 55 頁的「自訂管理代理程式配置」

啟動管理代理程式

您可以以下列方式啟動管理代理程式：

- 做為服務啟動，用於生產。請參閱第 50 頁的「將管理代理程式做為服務啟動」。為確保管理代理程式的可用性，請確定其在系統重新開機時會自動重新啟動。請參閱第 51 頁的「確保自動重新啟動管理代理程式」。
- 做為常規程序 (在主控制台模式中) 啟動，用於試用、測試或開發。請參閱第 53 頁的「在主控制台模式下啟動管理代理程式」。
- 利用 Solaris 10 的服務管理功能 (SMF)。請參閱第 54 頁的「使用 Solaris 10 服務管理功能執行管理代理程式」。

對於每種啟動方式，程序會視您使用的是 Java Enterprise System 還是獨立的 Application Server 而有所不同。

將管理代理程式做為服務啟動

將管理代理程式做為服務啟動，可確保其繼續執行，除非系統關閉或您明確停止它。指令視您的安裝與平台而定：

- 第 50 頁的「Solaris 或 Linux 系統中的 Java Enterprise System」
- 第 50 頁的「Windows 系統中的 Java Enterprise System」
- 第 51 頁的「Solaris 或 Linux 系統中的獨立 Application Server」
- 第 51 頁的「Windows 系統中的獨立 Application Server」

Solaris 或 Linux 系統中的 Java Enterprise System

若要將管理代理程式做為服務啟動，請使用以下指令：

```
/etc/init.d/ma-initd start
```

若要停止服務，請使用以下指令：

```
/etc/init.d/ma-initd stop
```

Windows 系統中的 Java Enterprise System

若要將管理代理程式做為 Windows 服務啟動，請使用以下指令：

```
HADB_install_dir\bin\ma -i [config-file]
```

選擇性引數 *config-file* 用於指定管理代理程式配置檔案。僅在要變更預設管理代理程式配置時使用配置檔案。如需更多資訊，請參閱第 55 頁的「自訂管理代理程式配置」。

若要停止管理代理程式並做為服務移除 (取消註冊)，請使用以下指令：

```
HADB_install_dir\bin\ma -r [config-file]
```

若要執行管理，請選擇 [管理工具] | [服務] 來執行啟動和停止服務、停用自動啟動等作業。

Solaris 或 Linux 系統中的獨立 Application Server

若要將管理代理程式做為服務啟動，請使用以下指令：

```
HADB_install_dir/bin/ma-initd start
```

若要停止服務，請使用以下指令：

```
HADB_install_dir/bin/ma-initd stop
```

若要變更預設值，請編輯 shell 程序檔 `HADB_install_dir/bin/ma-initd`。將 `ma-initd` 複製至目錄 `/etc/init.d`。替代程序檔中 `HADB_ROOT` 和 `HADB_MA_CFG` 的預設值以反映安裝：

- `HADB_ROOT` 為 HADB 安裝目錄 `HADB_install_dir`。
- `HADB_MA_CFG` 為管理代理程式配置檔案的位置。如需更多資訊，請參閱第 55 頁的「自訂管理代理程式配置」。

Windows 系統中的獨立 Application Server

若要將管理代理程式做為 Windows 服務啟動，請使用以下指令：

```
HADB_install_dir\bin\ma -i [config-file]
```

選擇性引數 `config-file` 用於指定管理代理程式配置檔案。僅在要變更預設管理代理程式配置時使用配置檔案。

若要停止管理代理程式並做為服務移除 (取消註冊)，請使用以下指令：

```
HADB_install_dir\bin\ma -r [config-file]
```

若要執行管理，請選擇 [管理工具] | [服務] 來執行啟動和停止服務、停用自動啟動等作業。

確保自動重新啟動管理代理程式

在產品部署中，請將管理代理程式配置成自動重新啟動。如此一來可確保萬一 `ma` 程序失敗或作業系統重新開機，管理代理程式依然可用。

在 Windows 平台上，將管理代理程式做為服務啟動後，請即使用 Windows 管理工具將服務啟動類型設定為「自動」，並設定所需的回復選項。

在 Solaris 和 Linux 平台上，請使用本節說明的程序，將管理代理程式配置成自動重新啟動。以下程序可確保管理代理程式僅在系統進入以下執行層級時啟動：

- 執行層級 3 (Solaris) (預設)。
- 執行層級 5 (RedHat Linux) (圖形化模式中預設)。

進入其他執行層級會停止管理代理程式。

▼ 在安裝了 Java Enterprise System 的 Solaris 或 Linux 上配置自動重新啓動

開始之前 本小節假設您對作業系統初始化和執行層級已有基本的瞭解。如需有關這些主題的資訊，請參閱作業系統文件。

- 1 確定您的系統預設執行層級為 3 或 5。
若要檢查系統的預設執行層級，請檢視 `/etc/inittab` 檔案，並在頂部附近查找類似下行的內容：
`id:5:initdefault:`
本範例中為預設執行層級 5。
- 2 如第 52 頁的「建立符號連結」所述，請建立符號連結至 `/etc/init.d/ma-initd` 檔案。
- 3 重新開機。

接下來的步驟 若要關閉自動啓動和停止代理程式，請移除這些連結或將連結名稱中的字母 K 和 S 變更爲小寫。

▼ 在安裝了獨立 Application Server 的 Solaris 或 Linux 上配置自動重新啓動

- 1 在 shell 中，將目前目錄變更至 `HADB_install_dir/bin`。
- 2 編輯 shell 程序檔 `ma-initd`。
請確定程序檔中 `HADB_ROOT` 和 `HADB_MA_CFG` 的預設值表現您的安裝：
 - `HADB_ROOT` 爲 HADB 安裝目錄 `HADB_install_dir`。
 - `HADB_MA_CFG` 爲管理代理程式配置檔案的位置。如需更多資訊，請參閱第 55 頁的「自訂管理代理程式配置」。
- 3 將 `ma-initd` 複製至目錄 `/etc/init.d`。
- 4 如第 52 頁的「建立符號連結」所述，請建立符號連結至 `/etc/init.d/ma-initd` 檔案。

接下來的步驟 若要關閉自動啓動和停止代理程式，請移除這些連結或將連結名稱中的字母 K 和 S 變更爲小寫。

建立符號連結

在 Solaris 中建立下列符號連結：

```

/etc/rc0.d/K20ma-initd
/etc/rc1.d/K20ma-initd
/etc/rc2.d/K20ma-initd
/etc/rc3.d/S99ma-initd
/etc/rc5.d/K20ma-initd (僅適用於 Sun 4m 和 4u 架構)
/etc/rc6.d/K20ma-initd
/etc/rcS.d/K20ma-initd

```

在 Linux 中建立下列符號連結：

```

/etc/rc0.d/K20ma-initd
/etc/rc1.d/K20ma-initd
/etc/rc3.d/S99ma-initd
/etc/rc5.d/S99ma-initd
/etc/rc6.d/K20ma-initd

```

在主控台模式下啟動管理代理程式

您可能希望在主控台模式下啟動管理代理程式，以便進行試用或測試。請勿在生產環境中以此方式啟動管理代理程式，因為 `ma` 程序在系統或程序失敗後不會重新啟動，並在指令視窗關閉時終止。指令視您所使用的平台與安裝而定：

- [第 53 頁的「Solaris 或 Linux 系統中的 Java Enterprise System」](#)
- [第 53 頁的「Windows 系統中的 Java Enterprise System」](#)
- [第 54 頁的「Windows 系統中的獨立 Application Server」](#)
- [第 54 頁的「Solaris 或 Linux 系統中的獨立 Application Server」](#)

Solaris 或 Linux 系統中的 Java Enterprise System

若要在主控台模式下啟動 HADB 管理代理程式，請使用以下指令：

```
opt/SUNWhadb/bin/ma [config-file]
```

預設管理代理程式配置檔案為 `/etc/opt/SUNWhadb/mgt.cfg`

若要停止管理代理程式，請結束程序或關閉 shell 視窗。

Windows 系統中的 Java Enterprise System

若要在主控台模式下啟動管理代理程式，請使用以下指令：

```
HADB_install_dir\bin\ma [config-file]
```

選擇性引數 `config-file` 為管理代理程式配置檔案的名稱。如需有關配置檔案的更多資訊，請參閱[第 55 頁的「自訂管理代理程式配置」](#)。

若要停止代理程式，請結束此程序。

Windows 系統中的獨立 Application Server

若要在主控台模式下啟動管理代理程式，請使用以下指令：

```
HADB_install_dir\bin\ma [config-file]
```

選擇性引數 *config-file* 為管理代理程式配置檔案的名稱。如需更多資訊，請參閱第 55 頁的「自訂管理代理程式配置」。

若要停止管理代理程式，請結束此程序。

Solaris 或 Linux 系統中的獨立 Application Server

若要在主控台模式下啟動 HADB 管理代理程式，請使用以下指令：

```
HADB_install_dir/bin/ma [config-file]
```

預設管理代理程式配置檔案為 *HADB_install_dir/bin/ma.cfg*

若要停止管理代理程式，請結束程序或關閉 shell 視窗。

使用 Solaris 10 服務管理功能執行管理代理程式

服務管理功能 (SMF) 會提供重新啟動、檢視及管理 Solaris 10 服務的機制。您可以使用 SMF 來啟動、重新啟動並管理 HADB 管理代理程式。

管理代理程式的預設管理資源識別碼 (FMRI) 為 `svc:/application/hadb-ma`。

管理代理程式指令語法

管理代理程式 `ma` 指令的語法是：

```
ma [common-options]
  [ service-options]
  config-file
```

其中：

- *common-options* 為第 54 頁的「管理代理程式指令語法」中說明的一個或多個共用選項。
- *service-options* 為第 54 頁的「管理代理程式指令語法」中說明的 Windows 服務選項之一。
- *config-file* 為管理代理程式配置檔案的完整路徑。如需更多資訊，請參閱第 55 頁的「自訂管理代理程式配置」。

表 3-1 管理代理程式共用選項

選項	說明	預設
--define <i>name=value</i> -D	指定特性 名稱 的 值 ，其中，特性為第 56 頁的「配置檔案」中定義的特性之一。此選項可重複多次。	無
--help-?	顯示說明資訊。	False
--javahome <i>path</i> -j	使用位於 <i>path</i> 的 Java Runtime Environment (1.4 或更高版本)。	無
--systemroot <i>path</i> -y	作業系統根的路徑，通常在 %SystemRoot% 中設定。	無
--version-V	顯示版本資訊。	False

表 3-2 說明將管理代理程式當成 Windows 服務啟動的選項。-i、-r 和 -s 選項互斥，亦即一次只能使用其中一個。

在 Windows 中，在配置檔案或指令行指定特性值路徑時，使用雙引號 (") 來替換包含空格的檔案路徑。以雙引號和反斜線 "\: 和 \" 替換磁碟機和目錄分隔符號 : 和 \。

表 3-2 管理代理程式服務選項 (僅適用於 Windows)

選項	說明	預設
--install-i	將代理程式安裝做為 Windows 服務，並啟動該服務。僅使用 -i、-r 和 -s 選項其中之一。	False
--name <i>servicename</i> -n	在主機上執行多個代理程式時，使用指定的服務名稱。	HADBMgrt Agent
--remove-r	停止服務並從 Windows 服務管理員刪除代理程式。僅使用 -i、-r 和 -s 選項其中之一。	False
--service-s	將代理程式做為 Windows 服務執行。僅使用 -i、-r 和 -s 選項其中之一。	False

自訂管理代理程式配置

HADB 包括可用於自訂管理代理程式設定的配置檔案。啟動管理代理程式而未指定配置檔案時，代理程式將使用預設值。如果指定了配置檔案，管理代理程式將使用該檔案中的設定。可以在網域中的所有主機上重複使用該配置檔案。

▼ 在 HADB 主機上自訂管理代理程式配置

- 1 編輯管理代理程式配置檔案，並依需求設定值。
- 2 啟動管理代理程式，並將自訂的配置檔案指定為引數。

配置檔案

在 Java Enterprise System 上，配置檔案中的所有項目都有加入註釋。使用預設配置無需進行變更。若要自訂管理代理程式配置，請從檔案中移除註釋，按需變更值，然後啟動管理代理程式，並將配置檔案做為引數指定。

管理代理程式配置檔案安裝於：

- Solaris 和 Linux：`/etc/opt/SUNWhadb/mgt.cfg`。
- Windows：`install_dir\lib\mgt.cfg`。

如果使用獨立安裝程式，則管理代理程式配置檔案安裝於：

- Solaris 和 Linux：`HADB_install_dir/bin/ma.cfg`。
- Windows：`HADB_install_dir\bin\ma.cfg`。

下表說明配置檔案中的設定。

表 3-3 配置檔案設定

設定名稱	說明	預設
console.loglevel	主控台記錄層級。有效值為 SEVERE、ERROR、WARNING、INFO、FINE、FINER 及 FINEST	警告
logfile.loglevel	記錄檔記錄層級。有效值為 SEVERE、ERROR、WARNING、INFO、FINE、FINER 及 FINEST	資訊
logfile.name	記錄檔名稱和位置。必須為具有讀取/寫入存取權限的有效路徑。	Solaris 和 Linux： <code>/var/opt/SUNWhadb/ma/ma.log</code> Windows： <code>HADB_install_dir\ma.log</code>
ma.server.type	用戶端協定。僅支援 JMXMP。	jmxmp
ma.server.jmxmp.port	用於內部 (UDP) 和外部 (TCP) 通訊的連接埠號。必須為正整數。建議範圍為 1024-49151。	1862
ma.server.mainternal.interfaces	用於多介面機器進行內部通訊的介面。必須為有效的 IPv4 位址遮罩。網域中的所有管理代理程式必須使用同一子網路 例如，如果主機具有兩個介面 10.10.116.61 和 10.10.124.61，則使用 10.10.116.0/24 來使用第一個介面。斜線後的數字表示子網路遮罩中的位元數。	無
ma.server.dbdevicepath	儲存 HADB 裝置資訊的路徑。	Solaris 和 Linux： <code>/var/opt/SUNWhadb/4</code> Windows： <code>HADB_install_dir\device</code>

表 3-3 配置檔案設定 (續)

設定名稱	說明	預設
ma.server.dbhistorypath	儲存 HADB 歷史檔案的路徑。	Solaris 和 Linux：/var/opt/SUNWhadb Windows：REPLACEDIR (在執行階段由實際 URL 替代。)
ma.server.dbconfigpath	儲存節點配置資料的路徑。	Solaris 和 Linux： /var/opt/SUNWhadb/dbdef Windows：C:\Sun\SUNWhadb\dbdef
repository.dr.path	網域儲存庫檔案的路徑。	Solaris 和 Linux：/var/opt/SUNWhadb/repository Windows：C:\Sun\SUNWhadb\repository

使用 hadbm 管理指令

使用 `hadbm` 指令行公用程式可管理 HADB 網域、其資料庫實例和節點。`hadbm` 公用程式 (也稱為管理用戶端) 將管理請求傳送至做為管理伺服器的指定管理代理程式，其具有存取儲存庫中的資料庫配置的權限。

本小節說明 `hadbm` 指令行公用程式，包括以下主題：

- 第 57 頁的「指令語法」
- 第 58 頁的「安全性選項」
- 第 59 頁的「一般選項」
- 第 60 頁的「環境變數」

指令語法

`hadbm` 公用程式位於 `HADB_install_dir/bin` 目錄中。`hadbm` 指令的一般語法為：

```
hadbm subcommand
[-short-option [option-value]]
[--long-option [option-value]]
[operands]
```

子指令識別要執行的作業。子指令是區分大小寫的。大部分子指令都有一個運算元 (通常是 `dbname`)。

這些選項修改 `hadbm` 如何執行子指令。選項須大小寫相符。每個選項均具有長形式和短形式。短形式之前為一個破折號 (-)，長形式之前為兩個破折號 (--)。除必須存在以打開某項功能的布林選項之外，大多數選項均需要引數值。選項不是成功執行指令所必需的。

如果子指令需要資料庫名稱，但未指定，`hadbm`將使用預設資料庫 `hadb`。

範例 3-1 hadbm 指令範例

以下說明 `status` 子指令：

```
hadbm status --nodes
```

安全性選項

出於安全原因，所有 `hadbm` 指令均需要管理員密碼。使用 `--adminpassword` 選項可在建立資料庫或網域時設定密碼。然後，在該資料庫或網域上執行作業時必須指定該密碼。

出於進階安全性考量，可使用 `--adminpasswordfile` 選項指定包含密碼的檔案，而不是在指令行輸入密碼。使用以下行在密碼檔案中定義密碼：

```
HADBМ_ADMINPASSWORD=password
```

使用密碼替代 *password*。該檔案中的任何其他內容均被忽略。

如果同時指定了 `--adminpassword` 和 `--adminpasswordfile` 選項，則 `--adminpassword` 具有優先權。如果需要密碼，但未在指令中指定，`hadbm` 會提示您需要密碼。

備註–僅可以在建立資料庫或網域時設定管理員密碼，而且以後無法變更。

除管理員密碼外，HADB 還需要資料庫密碼以執行修改資料庫模式的作業。使用以下指令時必須同時使用這兩個密碼：`hadbm create`、`hadbm addnodes` 和 `hadbm refragment`。

可使用 `--dbpassword` 選項在指令行指定資料庫密碼。與管理員密碼類似，您還可以將密碼輸入檔案，並使用 `--dbpasswordfile` 選項指定檔案位置。使用以下行在密碼檔案中設定密碼：

```
HADBМ_DBPASSWORD=password
```

爲了測試或試用，可以使用 `--no-adminauthentication` 選項在建立資料庫或網域時關閉密碼認證。如需更多資訊，請參閱第 62 頁的「[建立資料庫](#)」和第 61 頁的「[建立管理網域](#)」。

下表概括了 `hadbm` 安全性指令行選項。

表 3-4 hadbm 安全性選項

選項 (短形式)	說明
--adminpassword= <i>password</i> -w	指定資料庫或網域的管理員密碼。如果在建立資料庫或網域時使用此選項，必須在每次使用 hadbm 在資料庫或網域上作業時提供此密碼。 使用此選項或 --adminpasswordfile，但不同時使用。
--adminpasswordfile= <i>filepath</i> -W	指定包含資料庫或網域管理員密碼的檔案。如果在建立資料庫或網域時使用此選項，必須在每次使用 hadbm 在資料庫或網域上作業時提供此密碼。 使用此選項或 --adminpassword，但不同時使用。
--no-adminauthentication -U	建立資料庫或網域時使用此選項指定無需管理員密碼。出於安全性原因，請勿在生產部署中使用此選項。
--dbpassword= <i>password</i> -P	指定資料庫密碼。如果在建立資料庫時使用此選項，則每次使用 hadbm 指令在資料庫上作業時均必須提供該密碼。建立 HADB 系統使用者密碼。至少 8 個字元。使用此選項或 --dbpasswordfile，但不同時使用。
--dbpasswordfile= <i>filepath</i> -P	指定包含 HADB 系統使用者密碼的檔案。使用此選項或 --dbpassword，但不同時使用。

一般選項

一般指令選項可與任何 hadbm 子指令配合使用。所有選項均為布林選項 (預設值為 false)。下表說明 hadbm 一般指令選項。

表 3-5 hadbm 一般選項

選項 (短形式)	說明
--quiet -q	不出現訊息執行子指令，無任何說明訊息。
--help -?	顯示此指令的簡要說明和所有受支援的子指令。無需子指令。
--version -V	顯示 hadbm 指令的版本詳細資訊。無需子指令。
--yes -y	在非互動模式下執行子指令。
--force -f	非互動執行指令，且在符合指令的發佈條件時不丟出錯誤。

表 3-5 hadbm 一般選項 (續)

選項 (短形式)	說明
--echo	顯示子指令、所有選項及其使用者定義的值或預設值，然後執行該子指令。
-e	
--agent= <i>URL</i>	管理代理程式的 URL。 <i>URL</i> 為： <i>hostlist:port</i> ，其中 <i>hostlist</i> 為逗號分隔的主機名稱或 IP 位址清單， <i>port</i> 為管理代理程式作業時所在的連接埠號。
-m	預設為 localhost:1862。
	備註：此選項對於 hadbm addnodes 無效。

環境變數

為方便起見，可以設定環境變數而非指定指令選項。下表說明對應 hadbm 指令選項的環境變數。

表 3-6 HADB 選項與環境變數

長形式	短形式	預設	環境變數
--adminpassword	-w	無	\$HADBM_ADMINPASSWORD
--agent	--m	localhost:1862	\$HADBM_AGENT
--datadevices	-a	1	\$HADBM_DATADEVICES
dbname	無	hadb	\$HADBM_DB
--dbpassword	-p	無	\$HADBM_DBPASSWORD
--dbpasswordfile	-P	無	\$HADBM_DBPASSWORDFILE
--devicepath	-d	Solaris 和 Linux：/var/opt/SUNWhadb Windows：C:\Sun\AppServer \SUNWhadb\vers，其中 vers 為 HADB 版本編號。	\$HADBM_DEVICEPATH
--devicesize	-z	無	\$HADBM_DEVICESIZE
--echo	-e	False	\$HADBM_ECHO
--fast	-F	False	\$HADBM_FAST
--force	-f	False	\$HADBM_FORCE
--help	-?	False	\$HADBM_HELP

表 3-6 HADB 選項與環境變數 (續)

長形式	短形式	預設	環境變數
--historypath	-t	Solaris 和 Linux：/var/opt/SUNWhadb Windows：REPLACEDIR (在執行 階段由實際 URL 替代)。	\$HADBM_HISTORYPATH
--hosts	-H	無	\$HADBM_HOSTS
--interactive	-i	True	\$HADBM_INTERACTIVE
--no-refragment	-r	False	\$HADBM_NOREFRAGMENT
--portbase	-b	15200	\$HADBM_PORTBASE
--quiet	-q	False	\$HADBM_QUIET
--repair	-R	True	\$HADBM_REPAIR
--rolling	-g	True	\$HADBM_ROLLING
--saveto	-o	無	\$HADBM_SAVETO
--set	-S	無	\$HADBM_SET
--spares	-s	0	\$HADBM_SPARES
--startlevel	-l	normal	\$HADBM_STARTLEVEL
--version	-V	False	\$HADBM_VERSION
--yes	-y	False	\$HADBM_YES

配置 HADB

本小節說明以下基本 HADB 配置作業：

- [第 61 頁的「建立管理網域」](#)
- [第 62 頁的「建立資料庫」](#)
- [第 67 頁的「檢視和修改配置屬性」](#)
- [第 72 頁的「配置 JDBC 連線池」](#)

建立管理網域

`hadbm createdomain` 指令可建立包含指定 HADB 主機的管理網域。此指令可初始化主機和持續性配置存放區之間的內部通訊通道。

該指令語法為：

```

hadbm createdomain
  [--adminpassword=password | --adminpasswordfile=
file | --no-adminauthentication] [--agent=maurl]
  hostlist

```

hostlist 運算元是以逗號分隔的 HADB 主機清單，其中每一個均為有效的 IPv4 網路位址。希望位於新網域中的所有主機都包括在 *hostlist* 中。

請參閱第 59 頁的「一般選項」，以取得指令選項說明。

使用此指令之前，請確定 HADB 管理代理程式可在 *hostlist* 中的每台主機上執行。此外，管理代理程式還必須：

- 不是現有網域的成員。
- 配置為使用同一連接埠。
- 能夠經由 UDP、TCP 和使用 IP 多重播送到達彼此。

hadbm 建立管理網域之後，會啟用網域中的所有主機。然後管理代理程式即可以用於管理資料庫了。建立 HADB 網域之後是建立 HADB 資料庫。如需有關建立 HADB 資料庫的更多資訊，請參閱第 62 頁的「建立資料庫」。

範例 3-2 建立 HADB 管理網域

以下範例在四個指定的主機上建立管理網域：

```
hadbm createdomain --adminpassword= password host1,host2,host3,host4
```

hadbm 成功執行指令後，會顯示以下訊息：

```
Domain host1,host2,host3, host4 created.
```

建立 HADB 網域後，將 HADB 套裝軟體的路徑和版本與管理代理程式一同註冊。

建立資料庫

使用 `hadbm create` 指令手動建立資料庫。

使用此指令建立資料庫之前，請首先建立管理網域並註冊 HADB 套裝軟體。如果在執行 `hadbm create` 時尚未執行這兩個步驟，則會隱式執行這兩個步驟。雖然這看似需要較少的工作，但任何指令中出現故障均會很難除錯。而且，`hadbm create` 是不可分割的，即如果其中一個隱式指令失敗，則已成功執行的指令將不會回復。因此，最好僅在建立網域並註冊 HADB 套裝軟體之後建立資料庫。

例如，如果 `hadbm createdomain` 和 `hadbm registerpackage` 執行成功，但 `hadbm create database` 失敗，則 `hadbm createdomain` 和 `hadbm registerpackage` 所做的變更仍會保留。

▼ 建立資料庫

- 1 建立管理網域。
如需更多資訊，請參閱第 61 頁的「建立管理網域」
- 2 註冊 HADB 套裝軟體。
如需更多資訊，請參閱第 45 頁的「註冊 HADB 套裝軟體」。
- 3 使用 `hadbm create` 指令建立資料庫。
如需有關指令語法的資訊，請參閱以下小節。

hadbm create 指令語法

```
hadbm create [--package=name] [--packagepath=path] [--historypath=path]
[--devicepath=path] [--datadevices=number ] [--portbase=number]
[--spares=number ] [--set=attr-val-list] [--agent=maurl] [--no-cleanup]
[ --no-clear ] [ --devicesize =size] [--dbpassword=password | --dbpasswordfile=file
] --hosts=host list [--adminpassword=password | --adminpasswordfile=file |
--no-adminauthentication ] [dbname ]
```

dbname 運算元指定資料庫名稱，該名稱必須是唯一的。若要確保資料庫名稱的唯一性，請使用 `hadbm list` 指令列出現有資料庫名稱。請使用預設資料庫名稱，除非需要建立多個資料庫。例如，若要在同一組 HADB 機器上建立具有獨立資料庫的多個叢集，請為每個叢集使用獨立的資料庫名稱。

`hadbm create` 指令將錯誤訊息寫入主控台，而非記錄檔。

表 3-7 說明特殊 `hadbm create` 指令選項。請參閱第 59 頁的「一般選項」，以取得其他指令選項說明。

表 3-7 hadbm create 選項

選項 (短形式)	說明	預設
--datadevices= <i>number</i>	每個節點上的資料裝置數 (1 到 8 [含])。資料裝置從 0 開始計數。	1
-a		

表 3-7 hadbm create 選項 (續)

選項 (短形式)	說明	預設
--devicepath= <i>path</i> -d	裝置路徑。有四個裝置： <ul style="list-style-type: none">■ DataDevice■ NiLogDevice (節點內部記錄裝置)■ RelalgDevice (相關代數查詢裝置)■ NoManDevice (節點管理員裝置)。 此路徑必須存在且可寫入。若要為每個節點或每個裝置設定不同的此路徑，請參閱第 66 頁的「設定異質多工式裝置路徑」	Solaris 和 Linux：/var/opt/SUNWhadb Windows：C:\Sun\AppServer\SUNWhadb\vers，其中 vers 為 HADB 版本編號。 預設由管理代理程式配置檔案中的 ma.server.dbdevicepath 指定。如需更多詳細資訊，請參閱第 56 頁的「配置檔案」
--devicesize= <i>size</i> -z	每個節點的裝置大小。如需更多資訊，請參閱第 65 頁的「指定裝置大小」。 遵循第 83 頁的「增加現有節點的儲存空間」中的說明增加裝置大小。	1024MB 最大大小小於最大作業系統檔案大小或 256 GB。最小大小為： (4 x LogbufferSize + 16MB) / <i>n</i> 其中， <i>n</i> 為選項 --datadevices 指定的資料裝置數。
--historypath= <i>path</i> -t	歷史檔案的路徑。此路徑必須已經存在且可寫入。 如需有關歷史檔案的更多資訊，請參閱第 96 頁的「清除和歸檔歷史檔案」	預設由管理代理程式配置檔案中的 ma.server.dbhistorypath 指定。如需詳細資訊，請參閱第 56 頁的「配置檔案」 Solaris 和 Linux：/var/opt/SUNWhadb Windows：REPLACEDIR (在執行階段由實際 URL 替代。)
--hosts= <i>hostlist</i> -H	以逗號分隔的資料庫中節點之主機名稱或 IP 位址 (僅 IPv4) 清單。使用 IP 位址可避免對 DNS 查詢的依賴性。主機名稱必須為絕對的。localhost 或 127.0.0.1 不能用做主機名稱。 請參閱第 65 頁的「指定主機」，以取得更多資訊。	無
--package= <i>name</i> -k	HADB 套裝軟體名稱 (版本)。如果找不到套裝軟體，將註冊預設套裝軟體。 已停用此選項。使用 hadbm registerpackage 指令在網域中註冊套裝軟體。	無
--packagepath= <i>path</i> -L	HADB 軟體套裝軟體的路徑。僅當套裝軟體未在網域中註冊時使用。 已停用此選項。使用 hadbm registerpackage 指令在網域中註冊套裝軟體。	無

表 3-7 hadbm create 選項 (續)

選項 (短形式)	說明	預設
--portbase= <i>number</i> -b	用於節點 0 的連接埠基底號。在步驟 20 中自此數字開始為後繼節點指定連接埠基底號。每個節點使用其連接埠基底號及接下來連續的五個已編號連接埠。 若要在同一機器上執行數個資料庫，請明確配置連接埠號。	15200
--spares= <i>number</i> -s	備援節點數。此數字必須等於或小於在 --hosts 選項中指定的節點數。	0
--set=attr-val-list -S	以逗號分隔的資料庫配置屬性 (格式為 <i>name=value</i>) 清單。如需資料庫配置屬性說明，請參閱第 96 頁的「清除和歸檔歷史檔案」	無

範例 3-3 建立資料庫範例

以下指令為建立資料庫的範例：

```
hadbm create --spares 2 --devicesize 1024 --hosts n0,n1,n2,n3,n4,n5
```

指定主機

可使用 --hosts 選項指定以逗號分隔的資料庫節點之主機名稱或 IP 位址清單。hadbm create 指令可為清單中的每個主機名稱 (或 IP 位址) 建立一個節點。節點數必須相等。使用重複主機名稱在同一機器上建立具有不同連接埠號的多個節點。請確定相同機器上的節點並非鏡像節點，也不是來自不同的 DRU。

節點以此選項中列出的順序自零開始編號。第一個鏡像對為節點零 (0) 和一 (1)，第二個為二 (2) 和三 (3)，以此類推。奇數編號的節點在一個 DRU 中，偶數編號的節點在其他 DRU 中。使用 --spares 選項，備援節點為編號最大的節點。

如需有關配置雙網路介面的資訊，請參閱第 34 頁的「配置網路備援」

指定裝置大小

使用 --devicesize 選項可指定裝置大小。建議的裝置大小：

$$(4x / nd + 4l/d) / 0.99$$

其中，

- *x* 為使用者資料的總大小
- *n* 為節點數 (由 --hosts 選項指定)

- d 為每個節點上的裝置數 (由 `--datadevices` 選項指定)
- l 為記錄緩衝區大小 (由屬性 `LogBufferSize` 指定)

如果可能會發生重新分段 (例如，使用 `hadbm addnodes`)，則建議的裝置大小為：

$$(8x / nd + 4l/d) / 0.99$$

設定異質多工式裝置路徑

若要為每個節點或服務設定不同的裝置路徑，請使用 `hadbm create` 的 `--set` 選項。有四種類型的裝置：`DataDevice`、`NiLogDevice` (節點內部記錄裝置)、`RelalgDevice` (相關代數查詢裝置) 和 `NoManDevice` (節點管理員裝置)。每個 `name=value` 對的語法如下，其中，`-devno` 僅在 `device` 為 `DataDevice` 時需要：

```
node-nodeno.device-devno.Devicepath
```

例如：

```
--set Node-0.DataDevice-0.DevicePath=/disk0,
Node-1.DataDevice-0.DevicePath=/disk 1
```

還可以設定歷史檔案的異質多工式路徑，如下所示：

```
node-nodeno.historypath=path
```

如需有關歷史檔案的資訊，請參閱第 96 頁的「[清除和歸檔歷史檔案](#)」

未設定的特定節點或裝置裝置路徑均預設為 `--devicepath` 值。

備註 – 使用 `hadbm set` 和 `hadbm addnodes` 指令變更歷史檔案的裝置路徑和位置。

疑難排解

如果您在建立資料庫時遇到困難，請檢查以下內容：

- 確定在所有主機上啟動了管理代理程式並定義了 HADB 網域。如需詳細資訊，請參閱第 50 頁的「[啟動管理代理程式](#)」
- 必須設定檔案和目錄權限，以允許以下使用者對安裝、歷史、裝置和配置路徑擁有讀取、寫入和執行存取權限：
 - Sun Java System Application Server 管理使用者 (安裝期間設定)
 - HADB 系統使用者

如需有關設定使用者權限的詳細資訊，請參閱第 33 頁的「[準備 HADB 設定](#)」

Application Server 和 HADB 連接埠指定不可與同一機器上的其他連接埠指定衝突。建議的預設連接埠指定為：

- Sun Java System 訊息佇列：7676
- IIOP：3700
- HTTP 伺服器：80
- 管理伺服器：4848
- HADB 節點：每個節點使用六個連續的連接埠。例如，對於預設連接埠 15200，節點 0 使用 15200 至 15205，節點 1 使用 15220 至 15225，以此類推。

磁碟空間必須充足，請參閱「Sun Java System Application Server 9.1 版本說明」。

檢視和修改配置屬性

可以分別使用 `hadbm get` 和 `hadbm set` 指令檢視和修改資料庫配置。

取得配置屬性值

若要取得配置屬性值，請使用 `hadbm get` 指令。如需有效屬性清單，請參閱第 68 頁的「配置屬性」。指令語法為：

```
hadbm get attribute-list | --all
[dbname]
[--adminpassword=password | --adminpasswordfile=file]
[--agent=maurl]
```

dbname 運算元指定資料庫名稱。預設為 `hadb`。

attribute-list 運算元是以逗號分隔或引號引起的、並以空格分隔的屬性清單。`--all` 選項顯示所有屬性值。如需 `hadbm get` 的所有屬性清單，請參閱第 68 頁的「配置屬性」。

請參閱第 59 頁的「一般選項」，以取得指令選項說明。

範例 3-4 使用 `hadbm get` 之範例

```
hadbm get JdbcUrl,NumberOfSessions
```

設定配置屬性值

若要設定配置屬性值，請使用 `hadbm set` 指令。如需有效屬性清單，請參閱第 68 頁的「配置屬性」。

```
hadbm set [dbname] attribute
=value[,attribute=
value...]
[--adminpassword=password | --adminpasswordfile=file]
[--agent=maurl]
```

dbname 運算元指定資料庫名稱。預設為 `hadb`。

attribute=value 清單為以逗號分隔或引號引起的、並以空格分隔的屬性清單。

請參閱第 59 頁的「一般選項」，以取得指令選項說明。

如果此指令成功執行，則會重新啟動資料庫，並使其處於資料庫之前所處狀態或更佳狀態。如需有關資料庫狀態的資訊，請參閱第 88 頁的「取得 HADB 的狀態」。依照第 79 頁的「重新啟動資料庫」所述方式重新啟動 HADB。

無法使用 `hadbm set` 設定以下屬性，而應在建立資料庫時設定 (請參閱第 62 頁的「建立資料庫」)。

- `DatabaseName`
- `DevicePath`
- `HistoryPath`
- `NumberOfDataDevices`
- `Portbase`
- `JdbcUrl` (其值在資料庫建立期間根據 `--hosts` 和 `--portbase` 選項設定)。

備註 – 使用 `hadbm set` 設定配置屬性 (除 `ConnectionTrace` 或 `SQLTraceMode` 之外) 會導致 HADB 的輪替式重新啟動。在輪替式重新啟動中，會停止每個節點，並使用新的配置啟動，每次一個節點；HADB 服務不會中斷。

如果設定 `ConnectionTrace` 或 `SQLTraceMode`，不會發生輪替式重新啟動，但變更會僅對 Application Server 實例的 HADB 連線有效。

配置屬性

下表列出可使用 `hadbm set` 修改和使用 `hadbm get` 擷取的配置屬性。

表 3-8 配置屬性

屬性	說明	預設	範圍
<code>ConnectionTrace</code>	如果為 <code>true</code> ，則初始化或終止用戶端連線 (JDBC、ODBC) 時在 HADB 歷史檔案中記錄訊息。	<code>False</code>	<code>True</code> 或 <code>False</code>
<code>CoreFile</code>	請勿變更預設值。	<code>False</code>	<code>True</code> 或 <code>False</code>
<code>DatabaseName</code>	資料庫名稱。	<code>hadb</code>	
<code>DataBufferPoolSize</code>	共用記憶體中配置的資料緩衝區池的大小。	<code>200MB</code>	<code>16 - 2047 MB</code>

表 3-8 配置屬性 (續)

屬性	說明	預設	範圍
DataDeviceSize	<p>指定節點的裝置大小。如需有關建議的 DataDeviceSize 之資訊，請參閱第 65 頁的「指定裝置大小」</p> <p>最大值為 256GB 或最大作業系統檔案大小中的較小值。最小值為：</p> $(4 \times \text{LogbufferSize} + 16\text{MB}) / n$ <p>其中，n 為資料裝置數。</p>	1024MB	32 - 262144 MB
PackageName	資料庫使用之 HADB 軟體套裝軟體的名稱。	V4.x.x.x	無
DevicePath	<p>裝置位置：裝置為：</p> <ul style="list-style-type: none"> ■ 資料裝置 (DataDevice) ■ 節點內部記錄裝置 (NiLogDevice) ■ 相關代數查詢裝置 (RelalgDevice) 	<p>Solaris 和 Linux：/var/opt/SUNWhadb</p> <p>Windows：C:\Sun\AppServer\SUNWhadb\vers，其中 vers 為 HADB 版本編號。</p>	
EagerSessionThreshold	<p>確定使用的是一般還是急切閒置階段作業過期。</p> <p>在一般閒置階段作業過期中，閒置超過 SessionTimeout 秒的階段作業將過期。</p> <p>同步運作的階段作業數超過最大階段作業數的百分之 EagerSessionThreshold 時，閒置超過 EagerSessionTimeout 秒的階段作業將過期。</p>	NumberOfSessions 屬性的一半	0 - 100
EagerSessionTimeout	使用急切階段作業過期時，資料庫連線在過期之前可以閒置的時間 (以秒為單位表示)。	120 秒	0-2147483647 秒
EventBufferSize	<p>事件緩衝區的大小，在其中記錄資料庫事件。如果設定為 0，將不執行事件緩衝區記錄。</p> <p>出現故障期間，將傾印事件緩衝區。這可為故障原因提供有價值的資訊，並在試部署期間提供協助。</p> <p>將事件寫入記憶體會有效能懲罰。</p>	0 MB	0-2097152 MB
HistoryPath	<p>HADB 歷史檔案的位置，該檔案包含資訊、警告和錯誤訊息。</p> <p>此為唯讀屬性。</p>	<p>Solaris 和 Linux：/var/opt/SUNWhadb</p> <p>Windows：REPLACEDIR (在執行階段由實際 URL 替代。)</p>	

表 3-8 配置屬性 (續)

屬性	說明	預設	範圍
InternalLogbufferSize	節點內部記錄裝置大小，該裝置用於追蹤與儲存資料相關的作業。	12MB	4 - 128 MB
JdbcUrl	資料庫的 JDBC 連線 URL。 此為唯讀屬性。	無	
LogbufferSize	記錄緩衝區大小，該緩衝區可追蹤資料相關的作業。	48MB	4 - 2048 MB
MaxTables	HADB 資料庫中允許的最大表格數。	1100	100 - 1100
NumberOfDatadevices	HADB 節點使用的資料裝置數。 此為唯讀屬性。	1	1 - 8
NumberOfLocks	HADB 節點配置的鎖定數。	50000	20000 -1073741824
NumberOfSessions	可為 HADB 節點開啓的最大階段作業 (資料庫連線) 數。	100	1 - 10000
PortBase	用於為不同 HADB 程序建立不同的連接埠號之基底連接埠號。 此為唯讀屬性。	15200	10000 - 63000
RelalgDeviceSize	相關代數查詢中所使用裝置的大小。	128 MB	32 - 262144 MB
SessionTimeout	使用一般階段作業過期時資料庫連線在過期之前可以閒置的時間數。	1800 秒	0-2147483647 秒
SQLTraceMode	有關寫入歷史檔案的已執行 SQL 查詢之資訊量。 如果為 SHORT ，將記錄登入和登出 SQL 階段作業。如果為 FULL ，將記錄所有就緒和執行的 SQL 查詢，包括參數值。	NONE	NONE /SHORT /FULL
StartRepairDelay	備援節點允許出現故障的使用中節點執行節點回復的最長時間。如果出現故障的節點無法在此時間間隔內回復，備援節點即開始從出現故障的節點的鏡像複製資料，並變為使用中。建議不要變更預設值。	20 秒	0 - 100000 秒

表 3-8 配置屬性 (續)

屬性	說明	預設	範圍
StatInterval	<p>HADB 節點將流量和回應時間統計寫入其歷史檔案的間隔。若要停用，請設定為 0。</p> <p>此為統計行的範例：</p> <pre>Req-reply time: # 123, min= 69 avg= 1160 max= 9311 %=100.0</pre> <p>井號 (#) 後的數字為經由 StatInterval 服務的請求數。接下來的三個數字為經由 StatInterval 完成之作業事件的最小、平均和最大時間 (以微秒為單位)。百分比符號 (%) 後的數字為經由 StatInterval 在 15 毫秒內成功完成的作業事件數。</p>	600 秒	0 - 600 秒
SyslogFacility	<p>報告給 syslog 時使用的功能。應該配置 syslog 常駐程式 (請參閱 <code>man syslogd.conf</code>，以取得詳細資訊)。</p> <p>使用在同一機器上執行的其他應用程式未使用的功能。</p> <p>設定為 none 可停用 syslog 記錄。</p>	local0	local0、 local1、 local2、 local3、 local4、 local5、 local6、 local7、 kern、user、 mail、 daemon、 auth、 syslog、lpr、 news、 uucp、 cron、none
SysLogging	如果為 true，HADB 節點會將資訊寫入作業系統 syslog 檔案。	True	True 或 False
SysLogLevel	儲存至作業系統 syslog 檔案的 HADB 訊息之最低層級。將記錄該層級或更高層級的所有訊息。例如，「info」記錄所有訊息。	warning	nonealert errorwarning info
SyslogPrefix	插入所有 HADB 寫入 syslog 的訊息之前的文字字串。	hadb -dbname	
TakeoverTime	節點出現故障及其鏡像接管之間的時間。請勿變更預設值。	10000 (毫秒)	500 - 16000 毫秒

配置 JDBC 連線池

Application Server 使用 Java 資料庫連結 (JDBC) API 與 HADB 通訊。`asadmin configure-ha-cluster` 指令可自動建立 JDBC 連線池，以與 HADB 配合使用 (對於叢集 *cluster-name*)。連線池的名稱為「*cluster-name-hadb-pool*」。JDBC 資源的 JNDI URL 為「*jdbc/cluster-name-hastore*」。

連線池的初始配置通常是充足的。增加節點時，請變更持續的池大小，以便每個 HADB 使用中的節點均具有八條連線。請參閱第 84 頁的「增加節點」。

本小節包含以下主題：

- 第 72 頁的「取得 JDBC URL」
- 第 72 頁的「建立連線池」
- 第 74 頁的「建立 JDBC 資源」

如需有關連線池和 JDBC 資源的一般資訊，請參閱「Sun Java System Application Server 9.1 高可用性管理指南」。

取得 JDBC URL

設定 JDBC 連線池之前，需要使用 `hadbm get` 指令如下所示來確定 HADB 的 JDBC URL：

```
hadbm get JdbcUrl [dbname]
```

例如：

```
hadbm get JdbcUrl
```

此指令可顯示 JDBC URL，其形式如下：

```
jdbc:sun:hadb:host:port,  
host:port,...
```

移除 `jdbc:sun:hadb:` 前綴並使用 `host:port, host:port...` 部分做為 `serverList` 連線池特性值，如表 3-10 中所述。

建立連線池

下表概括 HADB 必需的連線池設定。增加節點時變更持續的池大小，但請勿變更其他設定。

表 3-9 HADB 連線池設定

設定	HADB 必需的值
名稱	HADB JDBC 資源的池名稱設定必須參照此名稱
資料庫供應商	HADB 4.4
全域作業事件支援	取消核取/false
資料來源類別名稱	com.sun.hadb.jdbc.ds.HadbDataSource
持續的池大小	為每個使用中的 HADB 節點使用 8 條連線。如需更多詳細資訊，請參閱系統部署指南。
必需的連線驗證	核取/true
驗證方法	meta-data
表名稱	不指定
所有連線故障	取消核取/false
作業事件隔離	repeatable-read
保證隔離層級	核取/true

下表概括 HADB 必需的連線池特性。增加節點時變更 `serverList`，但請勿變更其他特性。

表 3-10 HADB 連線池特性

特性	說明
username	用於 <code>asadmin create-session-store</code> 指令的 <code>storeuser</code> 名稱。
password	用於 <code>asadmin create-session-store</code> 指令的密碼 (<code>storepassword</code>)。
serverList	HADB 的 JDBC URL。若要確定此值，請參閱第 72 頁的「取得 JDBC URL」 如果要增加節點至資料庫，必須變更此值。請參閱第 84 頁的「增加節點」。
cacheDatabaseMetaData	為 <code>false</code> 時，必須確保對 <code>Connection.getMetaData()</code> 的呼叫可產生對資料庫的呼叫，才能確保連線有效。
eliminateRedundantEndTransaction	為 <code>true</code> 時，必須透過消除冗余確定和回復請求並在未開啓任何作業事件時忽略這些請求來提昇效能。
maxStatement	在磁碟機敘述池中快取的每條開啓連線的最大敘述數。此特性設定為 20。

範例 3-5 建立連線池

此為建立 HADB JDBC 連線池之 `asadmin create-jdbc-connection-pool` 指令的範例：

```
asadmin create-jdbc-connection-pool
--user adminname --password secret
--datasourceclassname com.sun.hadb.jdbc.ds.HadbDataSource
--steadypoolsize=32
--isolationlevel=repeatable-read
--isconnectvalidatereq=true
--validationmethod=meta-data
--property username=storename:password=secret456:serverList=
host\:port,host\:port,
host\:port,host\:port,
host\:port,host\:port
:cacheDatabaseMetaData=false:eliminateRedundantEndTransaction=true hadbpool
```

在 Solaris 上，使用雙反斜線 (\\) 替換特性值內的冒號字元 (:)。在 Windows 上，使用單反斜線 (\) 替換冒號字元 (:)。

建立 JDBC 資源

下表概括 HADB 必需的 JDBC 資源設定。

表 3-11 HADB JDBC 資源設定

設定	說明
JNDI 名稱	以下為階段作業持續性配置中的預設 JNDI 名稱：jdbc/hastore。可以使用預設名稱或其他名稱。 在啟動可用性服務時，還必須指定此 JNDI 名稱為 store-pool-jndi-name 持續性存放區特性的值。
池名稱	在此 JDBC 資源使用的 HADB 連線池名稱 (或 ID) 清單中選取。如需更多資訊，請參閱第 34 頁的「配置網路備援」
資料來源已啓用	核取/true

管理 HADB

通常需要在替代或升級網路、硬體、作業系統或 HADB 軟體時執行管理作業。以下小節說明各種管理作業：

- 第 75 頁的「管理網域」
- 第 76 頁的「管理節點」
- 第 78 頁的「管理資料庫」
- 第 81 頁的「回復階段作業資料損壞」

管理網域

可以在 HADB 網域上執行以下作業：

- 建立網域：如需更多資訊，請參閱第 61 頁的「建立管理網域」
- 第 75 頁的「延伸網域」
- 第 75 頁的「刪除網域」
- 第 75 頁的「列出網域中的主機」
- 第 75 頁的「從網域中移除主機」

請參閱第 58 頁的「安全性選項」和第 59 頁的「一般選項」，以取得指令選項說明。

延伸網域

使用 `extenddomain` 將主機增加至現有管理網域。指令語法為：

```
hadbm extenddomain
[--adminpassword=password | --adminpasswordfile=file]
[--agent=maurl]
hostlist
```

HADB 主機的 IP 位址必須為 IPv4 位址。

如需更多資訊，請參閱 `hadbm-extenddomain(1)`。

刪除網域

使用 `deletedomain` 移除管理網域。指令語法為：

```
hadbm deletedomain
[--adminpassword=password | --adminpasswordfile=file]
[--agent=maurl]
```

如需更多資訊，請參閱 `hadbm-deletedomain(1)`。

從網域中移除主機

使用 `reducedomain` 從管理網域中移除主機。指令語法為：

```
hadbm reducedomain
[--adminpassword=password | --adminpasswordfile=file]
[--agent=maurl]
host_list
```

如需更多資訊，請參閱 `hadbm-reducedomain(1)`。

列出網域中的主機

使用 `listdomain` 列出在管理網域中定義的所有主機。指令語法為：

```
hadbm listdomain  
[--adminpassword=password | --adminpasswordfile=file]  
[--agent=maurl]
```

如需更多資訊，請參閱 `hadbm-listdomain(1)`。

管理節點

可以在個別節點上執行以下作業：

- [第 76 頁的「啟動節點」](#)
- [第 77 頁的「停止節點」](#)
- [第 77 頁的「重新啟動節點」](#)

啟動節點

可能需要手動啟動因為使其主機離線以便進行硬體或軟體升級或替代而停止的 HADB 節點。而且，可能需要手動啟動由於某些原因 (而非雙重故障) 無法重新啟動的節點。如需有關如何從雙重故障回復的更多資訊，請參閱 [第 80 頁的「清除資料庫」](#)。

多數情況下，應該首先嘗試使用 `normal` 啟動層級啟動節點。如果 `normal` 啟動層級失敗或逾時，則必須使用 `repair` 啟動層級。

若要啟動資料庫中的節點，請使用 `hadbm startnode` 指令。語法為：

```
hadbm startnode  
  [--adminpassword=password | --adminpasswordfile=file]  
  [--agent=maurl]  
  [--startlevel=level]  
  nodeno  
  [dbname]
```

dbname 運算元指定資料庫名稱。預設為 `hadb`。

nodeno 運算元指定要啟動的節點數。使用 `hadbm status` 可顯示資料庫中所有節點數。

如需更多資訊，請參閱 `hadbm-startnode(1)`。

啟動層級選項

`hadbm startnode` 指令具有一個特殊選項，`--startlevel` (短形式 `-l`)，可指定要啟動節點的層級。

節點啟動層級有：

- **normal** (預設)：使用在節點上本機發現的資料 (記憶體或磁碟上資料裝置檔案中) 啟動節點，並使其與略過的最新更新之鏡像同步。

- **repair**：強制節點捨棄本機資料並從其鏡像複製。
- **clear**：重新初始化節點裝置並強制從其鏡像節點修復資料。需要初始化裝置檔案時使用，在裝置檔案損壞或包含裝置檔案的磁碟被替代時需要。

請參閱第 59 頁的「一般選項」，以取得其他指令選項說明。

範例 3-6 啟動節點範例

```
hadbm startnode 1
```

停止節點

可能需要停止節點以修復或升級主機電腦的硬體或軟體。若要停止節點，請使用 `hadbm stopnode` 指令。指令語法為：

```
hadbm stopnode
[--adminpassword=password | --adminpasswordfile=file]
[--agent=maurl]
[--no-repair]
nodeno
[dbname]
```

nodeno 運算元指定要停止的節點數。此節點個數的鏡像節點必須正在執行。使用 `hadbm status` 可顯示資料庫中所有節點數。

dbname 運算元指定資料庫名稱。預設為 `hadb`。

`hadbm stopnode` 指令具有一個特殊選項，`--no-repair` (短形式 `-R`)，表示使用無備援節點替代停止的節點。無此選項時，備援節點會啟動並接管停止的節點。

請參閱第 59 頁的「一般選項」，以取得其他指令選項說明。如需更多資訊，請參閱 `hadbm-stopnode(1)`。

範例 3-7 停止節點範例

```
hadbm stopnode 1
```

重新啟動節點

如果發現不正常的運作方式 (例如使用過多的 CPU)，則可能需要重新啟動節點。

若要重新啟動資料庫中的節點，請使用 `hadbm restartnode` 指令。指令語法為：

```
hadbm restartnode
[--adminpassword=password | --adminpasswordfile=file]
[--agent=maurl]
[--startlevel=level]
nodeno
[dbname]
```

dbname 運算元指定資料庫名稱。預設為 *hadb*。

nodeno 運算元指定要重新啟動的節點數。使用 *hadbm status* 可顯示資料庫中所有節點數。

hadbm restartnode 指令具有一個特殊選項，*--startlevel* (短形式 *-l*)，可指定要啟動節點的層級。請參閱第 76 頁的「啟動層級選項」，以取得更多資訊。

請參閱第 59 頁的「一般選項」，以取得其他指令選項說明。如需更多資訊，請參閱 *hadbm-restartnode(1)*。

範例 3-8 重新啟動節點範例

```
hadbm restartnode 1
```

管理資料庫

可以在 HADB 資料庫上執行以下作業：

- 第 78 頁的「啟動資料庫」
- 第 79 頁的「停止資料庫」
- 第 79 頁的「重新啟動資料庫」
- 第 80 頁的「列出資料庫」
- 第 80 頁的「清除資料庫」
- 第 81 頁的「移除資料庫」

啟動資料庫

若要啟動資料庫，請使用 *hadbm start* 指令。此指令可啟動所有在資料庫停止前執行的節點。個別停止 (離線) 的節點不會在資料庫停止後再啟動時啟動。

指令語法為：

```
hadbm start  
[--adminpassword=password | --adminpasswordfile=file]  
[--agent=maurl]  
[dbname]
```

dbname 運算元指定資料庫名稱。預設為 *hadb*。

請參閱第 59 頁的「一般選項」，以取得指令選項說明。如需更多資訊，請參閱 *hadbm-start(1)*。

範例 3-9 啟動資料庫範例

```
hadbm start
```

停止資料庫

在獨立的作業中停止和啟動資料庫時，資料在停止時不可用。若要使資料可用，可以如第 79 頁的「重新啟動資料庫」中所述重新啟動資料庫。

停止資料庫以：

- 移除資料庫。
- 執行影響所有 HADB 節點的系統維護。

停止資料庫前，停止使用資料庫的附屬 Application Server 實例，或將其配置為使用持續性類型而非 ha。

停止資料庫時，資料庫中所有正在執行的節點均會停止，並且資料庫的狀態變為停止。如需有關資料庫狀態的更多資訊，請參閱第 88 頁的「資料庫狀態」。

若要停止資料庫，請使用 `hadbm stop` 指令。指令語法為：

```
hadbm stop
[--adminpassword=password | --adminpasswordfile= file]
[--agent=maur]
[dbname]
```

dbname 運算元指定資料庫名稱。預設為 `hadb`。

請參閱第 59 頁的「一般選項」，以取得指令選項說明。如需更多資訊，請參閱 `hadbm-stop(1)`。

範例 3-10 停止資料庫範例

```
hadbm stop
```

重新啟動資料庫

如果發現奇怪的運作方式 (例如一致逾時問題)，可能要重新啟動資料庫。某些情況下，重新啟動可以解決這個問題。

重新啟動資料庫時，資料庫及其資料仍然可用。在獨立作業中停止和啟動 HADB 時，如果 HADB 停止，資料和資料庫服務不可用。這是因為依預設，`hadbm restart` 會執行節點輪流重新啟動：逐個停止並啟動節點。而 `hadbm stop` 則是同步停止所有節點。

若要重新啟動資料庫，請使用 `hadbm restart` 指令。指令語法為：

```
hadbm restart
[--adminpassword=password | --adminpasswordfile=file]
[--agent=maur]
[--no-rolling]
[dbname]
```

dbname 運算元指定資料庫名稱。預設為 `hadb`。

特殊選項 `--no-rolling` (簡捷形式 `-g`) 會指定立即重新啟動所有節點，導致服務中斷。無此選項時，此指令會將資料庫中的每個節點重新啟動至目前狀態或更佳的状态。

請參閱第 59 頁的「一般選項」，以取得其他指令選項說明。如需更多資訊，請參閱 `hadbm-restart(1)`。

例如：

```
hadbm restart
```

列出資料庫

若要列出 HADB 實例中的所有資料庫，請使用 `hadbm list` 指令。指令語法為：

```
hadbm list
[ --agent=maurl]
[ --adminpassword=password | --adminpasswordfile=file]
```

請參閱第 59 頁的「一般選項」，以取得指令選項說明。如需更多資訊，請參閱 `hadbm-list(1)`。

清除資料庫

在以下情況時清除資料庫：

- `hadbm status` 指令表明資料庫為不可作業的，或請參閱第 88 頁的「取得 HADB 的狀態」。
- 多個節點未回應，並長時間處於等待狀態。
- 自階段作業資料損毀中回復。請參閱第 81 頁的「回復階段作業資料損壞」。

`hadbm clear` 指令可停止資料庫節點、清除資料庫裝置，然後啟動節點。此指令可清除儲存在 HADB 中的 Application Server 模式資料，包括表格、使用者名稱和密碼。執行 `hadbm clear` 後，請使用 `asadmin configure-ha-cluster` 重新建立資料模式、重新配置 JDBC 連線池，並重新載入階段作業持續性存放區。

指令語法為：

```
hadbm clear [--fast] [--spares=number]
[ --dbpassword=password | --dbpasswordfile= file]
[ --adminpassword=password | --adminpasswordfile= file]
[ --agent=maurl]
[dbname]
```

dbname 運算元指定資料庫名稱。預設為 `hadb`。

下表說明特殊的 `hadbm clear` 指令選項。請參閱第 59 頁的「一般選項」，以取得其他選項說明。

如需更多資訊，請參閱 `hadbm-clear(1)`。

表 3-12 hadbm clear 選項

選項	說明	預設
--fast -F	初始化資料庫時略過裝置初始化。如果磁碟儲存裝置損壞，請勿使用。	不適用
--spares= <i>number</i> -s	經重新初始化的資料庫將具有的備援節點數。必須等於或小於資料庫中的節點數。	之前的備援數

例如：

```
hadbm clear --fast --spares=2
```

移除資料庫

若要移除現有資料庫，請使用 `hadbm delete` 指令。此指令可刪除資料庫的配置檔案、裝置檔案和歷史檔案，並可釋放共用記憶體資源。要移除的資料庫必須存在且停止。請參閱第 79 頁的「停止資料庫」。

指令語法為：

```
hadbm delete  
[ --adminpassword=password | --adminpasswordfile=file ]  
[ --agent=maurl ]  
[ dbname ]
```

dbname 運算元指定資料庫名稱。預設為 `hadb`。

請參閱第 59 頁的「一般選項」，以取得指令選項說明。如需更多資訊，請參閱 `hadbm-delete(1)`。

範例 3-11 移除資料庫範例

指令：

```
hadbm delete
```

刪除預設資料庫 `hadb`。

回復階段作業資料損壞

以下為階段作業資料可能已損壞的指示：

- 每次應用程式嘗試儲存階段作業狀態時，Application Server 系統記錄 (`server.log`) 中即會顯示錯誤訊息。

- 伺服器記錄中的錯誤訊息表示找不到階段作業，或者在階段作業啟動期間無法載入。
- 之前被鈍化後啟動的階段作業包含空的或不正確的階段作業資料。
- 實例出現故障時，容錯移轉階段作業包含空的或不正確的階段作業資料。
- 實例出現故障時，嘗試載入容錯移轉階段作業的實例會在伺服器記錄中導致錯誤，表明找不到階段作業或無法載入。

▼ 使階段作業存放區返回一致狀態

如果您確定階段作業存放區已損壞，請透過以下程序使其返回一致狀態：

1 清除階段作業存放區。

確定此動作是否更正了問題。如果已更正，則停止。如果未更正 (例如仍在伺服器記錄中看到錯誤)，請繼續執行。

2 重新初始化所有節點上的資料空間，並清除資料庫中的資料。

請參閱第 80 頁的「清除資料庫」。

確定此動作是否更正了問題。如果已更正，則停止。如果未更正 (例如仍在伺服器記錄中看到錯誤)，請繼續執行。

3 刪除並重新建立資料庫。

請參閱第 81 頁的「移除資料庫」和第 62 頁的「建立資料庫」

延伸 HADB

有兩種原因需要延伸原始 HADB 配置：

- 儲存的階段作業資料容量超出資料裝置中的現有儲存空間。由於資料裝置空間已滿，作業事件可能中斷。
- 使用者載入增加，耗盡系統資源。需要增加更多主機。

本小節說明如何不關閉 Application Server 叢集或資料庫延伸 HADB，尤其是：

- 第 83 頁的「增加現有節點的儲存空間」
- 第 83 頁的「增加機器」
- 第 84 頁的「增加節點」
- 第 86 頁的「重新分段資料庫」
- 第 87 頁的「透過重新建立資料庫增加節點」

另請參閱第 94 頁的「維護 HADB 機器」中的相關資訊。

增加現有節點的儲存空間

增加 HADB 儲存空間：

- 如果使用者作業事件重複中斷，並顯示以下錯誤訊息之一：
 - 4592: No free blocks on data devices
 - 4593: No unreserved blocks on data devices
- 如果 `hadbm deviceinfo` 指令始終報告自由空間不足，請參閱第 90 頁的「取得裝置資訊」。

如果節點上存在未使用的磁碟空間或增加磁碟容量時，還可能要增加現有節點的儲存空間。如需有關建議的資料裝置大小之資訊，請參閱第 65 頁的「指定裝置大小」。

若要增加節點的儲存空間，請使用 `hadbm set` 指令增加資料裝置大小。

指令語法為：

```
hadbm set DataDeviceSize=size
```

其中，*size* 為資料裝置大小 (MB)。

請參閱第 59 頁的「一般選項」，以取得指令選項說明。

變更 `FaultTolerant` 或更高狀態的資料庫資料裝置大小會升級系統，而沒有資料遺失或可用性損失。資料庫在重新配置期間仍在作業。變更非 `FaultTolerant` 或更好狀態之系統上的裝置大小導致資料遺失。如需有關資料庫狀態的更多資訊，請參閱第 88 頁的「資料庫狀態」。

範例 3-12 設定資料裝置大小範例

以下指令為設定資料裝置大小之範例：

```
hadbm set DataDeviceSize=1024
```

增加機器

如果 HADB 需要更多處理或儲存容量，您可能要增加機器。若要新增執行 HADB 的機器，請與 Application Server 一同安裝 HADB 套裝軟體或僅安裝套裝軟體，如第 2 章中所述。如需有關節點拓樸替代方案的說明，請參閱「Sun Java System Application Server 9.1 部署規劃指南」中的第 3 章「選取拓樸」。

▼ 新增機器至現有 HADB 實例

- 1 在所有新節點上啟動管理代理程式。
- 2 延伸新主機的管理網域。
如需詳細資訊，請參閱 `hadbm extenddomain` 指令。
- 3 啟動這些主機上的新節點。
如需詳細資訊，請參閱第 84 頁的「增加節點」

增加節點

若要增加 HADB 系統的處理和儲存容量，請建立新節點並將其增加至資料庫。

增加節點後，請更新 HADB JDBC 連線池的以下特性：

- `serverlist` 特性。
- 持續的池大小。通常可以為每個新節點增加 8 條連線。如需更多資訊，請參閱「Sun Java System Application Server 9.1 部署規劃指南」中的「System Sizing」。

若要增加節點，請使用 `hadbm addnodes` 指令。指令語法為：

```
hadbm addnodes [--no-refragment] [--spares=sparecount]
[--historypath=path]
[--devicepath=path]
[--set=attr-name-value-list]
[--dbpassword=password | --dbpasswordfile=file ]
[--adminpassword=password | --adminpasswordfile=file]
--hosts=hostlist [dbname]
```

`dbname` 運算元指定資料庫名稱。預設為 `hadb`。資料庫必須處於 `HAFaultTolerant` 或 `FaultTolerant` 狀態。如需有關資料庫狀態的更多資訊，請參閱第 88 頁的「資料庫狀態」。

如果未指定 `--devicepath` 和 `--historypath` 選項，新節點將具有與現有資料庫相同的裝置路徑，並使用相同的歷史檔案。

增加節點執行重新分段並重新分配現有資料作業，以將新節點包括在系統中。線上重新分段需要用於 HADB 節點的磁碟具有足夠空間以同時包含舊資料和新資料，直至重新分段完成，亦即使用者資料大小不得超過使用者資料可用空間的 50%。如需詳細資訊，請參閱第 90 頁的「取得裝置資訊」

備註 – 增加節點的最佳時間為系統負載較輕時。

範例 3-13 增加節點範例

例如：

```
hadbm addnodes -adminpassword=password --hosts n6,n7,n8,n9
```

下表說明特殊的 `hadbm addnodes` 指令選項。請參閱第 59 頁的「一般選項」，以取得其他選項說明。

表 3-13 hadbm addnodes 選項

選項	說明	預設
--no-refragment -r	建立節點期間請勿重新分段資料庫。在此情況下，可稍後使用 <code>hadbm refragment</code> 指令重新分段資料庫以使用新節點。如需有關重新分段的詳細資訊，請參閱第 86 頁的「重新分段資料庫」 如果沒有足夠的裝置空間進行重新分段，請重新建立包含更多節點的資料庫。請參閱第 87 頁的「透過重新建立資料庫增加節點」	不適用
--spares= number -s	除現有備援節點之外的新備援節點數。必須等於或小於增加的節點數。	0
--devicepath= path -d	裝置路徑。裝置為： <ul style="list-style-type: none">■ DataDevice■ NiLogDevice (節點內部記錄裝置)■ RelalgDevice (相關代數查詢裝置) 此路徑必須已經存在且可寫入。若要為每個節點或每個裝置設定不同的此路徑，請參閱第 66 頁的「設定異質多工式裝置路徑」	Solaris 和 Linux： <code>HADB_install_dir/device</code> Windows： <code>C:\Sun\AppServer\SUNWhadb\vers</code> ，其中 <code>vers</code> 為 HADB 版本編號。

表 3-13 hadbm addnodes 選項 (續)

選項	說明	預設
--hosts= <i>hostlist</i>	以逗號分隔的資料庫新節點的新主機名稱清單。為清單中每個以逗號分隔的項目建立一個節點。節點數必須相等。HADB 主機的 IP 位址必須為 IPv4 位址。	無
-H	使用重複主機名稱在同一機器上建立具有不同連接埠號的多個節點。確定同一機器上的節點並非鏡像節點。 奇數編號的節點在一個 DRU 中，偶數編號的節點在其他 DRU 中。如果使用了 --spares，則新的備用節點為編號最大的節點。 如果使用雙網路介面建立資料庫，必須以相同方式配置新節點。請參閱第 34 頁的「配置網路備援」。	

重新分段資料庫

重新分段資料庫以在新建立的節點中儲存資料。重新分段可在所有使用中節點上平均分配資料。

若要重新分段資料庫，請使用 `hadbm refragment` 指令。指令語法為：

```
hadbm refragment [--dbpassword=password | --dbpasswordfile=file]
[--adminpassword=password | --adminpasswordfile=file]
[--agent=maurl]
[dbname]
```

dbname 運算元指定資料庫名稱。預設為 `hadb`。資料庫必須處於 `HAFaultTolerant` 或 `FaultTolerant` 狀態。如需有關資料庫狀態的更多資訊，請參閱第 88 頁的「取得 HADB 的狀態」。

請參閱第 59 頁的「一般選項」，以取得指令選項說明。如需更多資訊，請參閱 `hadbm-fragment(1)`。

線上重新分段需要用於 HADB 節點的磁碟具有足夠空間以同時包含舊資料和新資料，直至重新分段完成，亦即使用者資料大小不得超過使用者資料可用空間的 50%。如需詳細資訊，請參閱第 90 頁的「取得裝置資訊」

備註 – 重新分段資料庫的最佳時間為系統負載較輕時。

如果此指令在多次嘗試之後仍然失敗，請參閱第 87 頁的「透過重新建立資料庫增加節點」

範例 3-14 重新分段資料庫範例

例如：

```
hadbm refragment
```

透過重新建立資料庫增加節點

如果線上重新分段在新增節點時始終失敗 (由於資料裝置空間不足或其他原因)，請重新建立包含新節點資料庫。這將導致現有使用者資料和模式資料遺失。

▼ 透過重新建立資料庫增加節點

此程序可讓您在整個過程中始終維持 HADB 可用性。

1 對於每個 Application Server 實例：

- a. 在負載平衡器中停用 Application Server 實例。
- b. 停用階段作業持續性。
- c. 重新啟動 Application Server 實例。
- d. 在負載平衡器中重新啟用 Application Server 實例。

如果無需維持可用性，可以在負載平衡器中立即停用並重新啟用所有伺服器實例。這會節省時間，並防止過期階段作業資料容錯移轉。

2 如第 79 頁的「停止資料庫」中所述停止資料庫。

3 如第 81 頁的「移除資料庫」中所述刪除資料庫。

4 如第 62 頁的「建立資料庫」中所述重新建立包含附加節點的資料庫。

5 如第 72 頁的「配置 JDBC 連線池」中所述重新配置 JDBC 連線池。

6 重新載入階段作業持續性存放區。

7 對於每個 Application Server 實例：

- a. 在負載平衡器中停用 Application Server 實例。
- b. 啟用階段作業持續性。
- c. 重新啟動 Application Server 實例。

d. 在負載平衡器中重新啓用 **Application Server** 實例。

如果無需維持可用性，可以在負載平衡器中立即停用並重新啓用所有伺服器實例。這會節省時間，並防止過期階段作業資料容錯移轉。

監視 HADB

可以透過以下方式監視 HADB 的活動：

- 第 88 頁的「取得 HADB 的狀態」
- 第 90 頁的「取得裝置資訊」
- 第 91 頁的「取得執行階段資源資訊」

這些小節簡要說明 `hadbm status`、`hadbm deviceinfo` 和 `hadbm resourceinfo` 指令。如需有關 HADB 資訊的說明，請參閱「Sun Java System Application Server 9.1 Performance Tuning Guide」中的「Performance」。

取得 HADB 的狀態

使用 `hadbm status` 指令可顯示資料庫或其節點的狀態。指令語法為：

```
hadbm status  
[ --nodes]  
[ --adminpassword=password | --adminpasswordfile=file]  
[ --agent=maurl]  
[ dbname]
```

`dbname` 運算元指定資料庫名稱。預設為 `hadb`。

`--nodes` 選項 (短形式 `-n`) 顯示資料庫中每個節點上的資訊。如需更多資訊，請參閱第 89 頁的「節點狀態」。請參閱第 59 頁的「一般選項」，以取得其他指令選項說明。

如需更多資訊，請參閱 `hadbm-status(1)`。

範例 3-15 取得 HADB 狀態範例

例如：

```
hadbm status --nodes
```

資料庫狀態

資料庫的**狀態**概括其目前情況。下表說明可能的資料庫狀態。

表 3-14 HADB 狀態

資料庫狀態	說明
高可用性錯誤容錯 (HAFaultTolerant)	資料庫是錯誤容錯的，並且在每個 DRU 上至少有一個備援節點。
錯誤容錯	所有鏡像的節點對均啟動並在執行。
可作業	每個鏡像的節點對中至少一個節點在執行。
不可作業	一個或多個鏡像的節點對均遺失兩個節點。 如果資料庫不可作業，請按照第 80 頁的「清除資料庫」中的說明清除資料庫。
已停止	資料庫中沒有節點在執行。
不明的	無法確定資料庫的狀態。

節點狀態

使用 `--nodes` 選項 `hadbm status` 指令可顯示資料庫中每個節點的以下資訊：

- 節點編號
- 執行節點的機器名稱
- 節點的連接埠號
- 節點的角色。如需角色及其意義的清單，請參閱第 89 頁的「節點角色」
- 節點狀態。如需狀態及其意義的清單，請參閱第 90 頁的「節點狀態」
- 相應鏡像節點數。

可以按照以下小節中的說明變更節點的角色和狀態：

- 第 89 頁的「節點角色」
- 第 90 頁的「節點狀態」

節點角色

在建立節點期間為其指定角色，角色可以是以下任意一個：

- **使用中**：儲存資料並允許用戶端存取。使用中的節點在鏡像對中。
- **備援**：允許用戶端存取，但不儲存資料。初始化資料裝置後，監視其他資料節點以在其他節點不可用時啟動修復程序。
- **離線**：不提供任何服務，除非角色變更。置回線上時，其角色會變更回先前的角色。
- **關閉**：使用中和離線之間的中間步驟，等待備援節點接管其功能。備援節點接管後，節點即離線。

節點狀態

節點可處於以下任意一種狀態：

- **正在啟動**：節點正在啟動。
- **等待中**：節點無法確定其啟動層級，處於離線狀態。如果某個節點處於此狀態超過兩分鐘，請停止該節點並在 `repair` 層級啟動節點。請參閱第 77 頁的「停止節點」、第 76 頁的「啟動節點」和第 80 頁的「清除資料庫」。
- **正在執行**：節點正在提供適合其角色的所有服務。
- **正在停止**：節點正在執行停止程序。
- **已停止**：節點為非使用中。禁止修復已停止的節點。
- **正在回復**：節點正在回復。節點出現故障時，鏡像節點會接管出現故障的節點的功能。出現故障的節點嘗試使用主記憶體中或磁碟上的資料和記錄回復。出現故障的節點使用鏡像節點中的記錄擷取在其當機時執行的作業事件。如果回復成功，節點即變為使用中。如果回復失敗，節點狀態變為正在修復。
- **正在修復**：正在修復節點。此作業重新初始化節點並從鏡像節點中複製資料和記錄。修復比回復更為耗時。

取得裝置資訊

監視 HADB 資料 (磁碟儲存) 裝置中的自由空間：

- 常規檢查磁碟空間的使用趨勢。
- 做為預防性維護的一部分：如果使用者載入增加，且您要重新調整資料庫配置大小或比例。
- 做為調整資料庫比例的一部分：執行 `hadbm addnodes` 新增節點至系統之前，請檢查是否有足夠的裝置空間。請記住，您在現有節點上需要大約 40-50% 的自由空間才能增加節點。
- 在歷史檔案和 `server.log` 檔案中看到諸如：
 - `No free blocks on data devices`
 - `No unreserved blocks on data devices` 之類的訊息時。

使用 `hadbm deviceinfo` 指令取得有關資料裝置中自由空間的資訊。此指令可顯示資料庫每個節點的以下資訊：

- 配置的總裝置大小，以 MB 表示 (Totalsize)。
- 自由空間，以 MB 表示 (Freesize)。
- 目前裝置使用百分比 (Usage)

指令語法為：

```
hadbm deviceinfo [--details]
[--adminpassword=password | --adminpasswordfile=file]
[--agent=maurl] [dbname]
```

dbname 運算元指定資料庫名稱。預設為 *hadb*。

--details 選項可顯示以下附加資訊：

- 裝置讀取作業的次數。
- 裝置寫入作業的次數。
- 裝置名稱。

請參閱第 59 頁的「一般選項」，以取得其他指令選項說明。

如需更多資訊，請參閱 `hadbm-deviceinfo(1)`。

若要確定可用於使用者資料的可用空間，請用總裝置大小減去為 HADB 保留的空間： $4 \times \text{LogBufferSize} + 1\% \times \text{裝置大小}$ 。如果不知道記錄緩衝區的大小，請使用指令 `hadbm get logbufferSize`。例如，如果總裝置大小為 128 MB，`LogBufferSize` 為 24 MB，則可用於使用者資料的空間為 $128 - (4 \times 24) = 32$ MB。在 32 MB 中，一半用於重複的資料，大約百分之一用於索引，僅 25% 可真正用於使用者資料。

使用者資料可用的空間為總大小和保留大小之間的差。如果將來重新分段資料，自由空間大小必須大約等於使用者資料可用空間的 50%。如果重新分段組無關緊要，則資料裝置可達至其最大值。系統裝置空間不足時，資源使用量警告會寫入歷史檔案。

如需有關調校 HADB 的更多資訊，請參閱 *Sun Java System Application Server Performance Tuning Guide*。

範例 3-16 取得裝置資訊範例

以下指令：

```
hadbm deviceinfo --details
```

顯示以下範例結果：

NodeNO	Totalsize	Freesize	Usage	NReads	NWrites	DeviceName
0	128	120	6%	10000	5000	C:\Sun\SUNWhadb\hadb.data.0
1	128	124	3%	10000	5000	C:\Sun\SUNWhadb\hadb.data.1
2	128	126	2%	9500	4500	C:\Sun\SUNWhadb\hadb.data.2
3	128	126	2%	9500	4500	C:\Sun\SUNWhadb\hadb.data.3

取得執行階段資源資訊

`hadbm resourceinfo` 指令可顯示 HADB 執行階段資源資訊。可以使用此資訊協助識別資源內容，並減少效能瓶頸。如需詳細資訊，請參閱「*Sun Java System Application Server 9.1 Performance Tuning Guide*」中的「*Tuning HADB*」。

指令語法為：

```
hadbm resourceinfo [--databuf] [--locks] [--logbuf] [--nologbuf]
[--adminpassword=password | --adminpasswordfile=file]
[--agent=maurl]
[dbname]
```

dbname 運算元指定資料庫名稱。預設為 hadb。

下表說明 hadbm resourceinfo 特殊指令選項。請參閱第 59 頁的「一般選項」，以取得其他指令選項說明。

如需更多資訊，請參閱 hadbm-resourceinfo(1)。

表 3-15 hadbm resourceinfo 指令選項

選項	說明
--databuf	顯示資料緩衝區池資訊。
-d	請參閱以下第 92 頁的「資料緩衝區池資訊」，以取得更多資訊。
--locks	顯示鎖定資訊。
-l	請參閱以下第 93 頁的「鎖定資訊」，以取得更多資訊。
--logbuf	顯示記錄緩衝區資訊。
-b	請參閱以下第 93 頁的「記錄緩衝區資訊」，以取得更多資訊。
--nologbuf	顯示節點內部記錄緩衝區資訊。
-n	請參閱以下第 94 頁的「節點內部記錄緩衝區資訊」，以取得更多資訊。

資料緩衝區池資訊

資料緩衝區池資訊包含：

- NodeNo：節點編號。
- Avail：可用的池總空間 (MB)。
- Free：可用的自由空間 (MB)。
- Access：自啟動至目前從資料庫存取資料緩衝區的累積次數。
- Misses：自資料庫啟動至目前已發生頁面錯誤的累計次數。
- Copy-on-Write：出於檢查點操作目的而在資料緩衝區中內部複製頁面的累計次數。

使用者作業事件對記錄執行作業時，包含該記錄的頁面必須在資料緩衝區池中。如果不在，將會發生 miss 或頁面錯誤。然後，作業事件必須等待，直至從磁碟上的資料裝置檔案中擷取頁面。

如果遺失率高，請增大資料緩衝區池。因為遺失是累積的，請定期執行 hadbm resourceinfo，並使用兩次執行之間的差異查看遺失率趨勢。如果自由空間非常小，也不必擔心，因為檢查點機制將使新的區段變得可用。

範例 3-17 資料緩衝區池資訊範例

例如：

```
NodeNO Avail Free Access Misses Copy-on-Write
0 256 128 100000 50000 10001 256 128 110000 45000 950
```

鎖定資訊

鎖定資訊如下：

- NodeNo：節點編號。
- Avail：節點上可用鎖定的總數。
- Free：自由鎖定的數目。
- Waits：等待獲得鎖定之作業事件數。這是累積數。

一個單一作業事件無法使用節點上超過 25% 的可用鎖定。因此，執行大量作業的作業事件因該瞭解此限制。最好成批執行此類作業事件，其中每一批必須做為獨立作業事件處理，亦即每一批均進行確定。這是必須的，因為以 `repeatable read` 隔離層級執行的讀取作業與 `delete`、`insert` 和 `update` 作業，都是使用只有在作業事件終止後才會釋放的鎖定。

如要變更 `NumberOfLocks`，請參閱第 96 頁的「清除和歸檔歷史檔案」。

範例 3-18 鎖定資訊範例

例如：

```
NodeNO Avail Free Waits
0 50000 20000 101 50000 20000 0
```

記錄緩衝區資訊

記錄緩衝區資訊為：

- NodeNo：節點編號。
- Available：配置給記錄緩衝區的記憶體容量 (MB)
- Free：可用記憶體容量 (MB)

如果自由空間非常小，請勿擔心，因為 HADB 會壓縮記錄緩衝區。HADB 從環狀緩衝區開始壓縮，並在連續的記錄上執行壓縮。當 HADB 遇到節點尚未執行且鏡像節點尚未接收的記錄時，壓縮無法繼續。

範例 3-19 記錄緩衝區資訊範例

例如：

```
NodeNO Avail Free
0 16 21 16 3
```

節點內部記錄緩衝區資訊

節點內部記錄緩衝區資訊為：

- 節點編號。
- Available：配置給記錄裝置的記憶體容量 (MB)
- Free：可用記憶體容量 (MB)

範例 3-20 內部記錄緩衝區資訊範例

例如：

```
NodeNO Avail Free
```

```
0 16 21 16 3
```

維護 HADB 機器

HADB 透過複製鏡像節點上的資料實現錯誤容錯。在生產環境中，鏡像節點位於其鏡像之節點的獨立 DRU 上，如「Sun Java System Application Server 9.1 部署規劃指南」中所述。

故障指未預期事件，例如硬體故障、電源故障或作業系統重新開機。HADB 可容錯個體故障：即故障發生在一個節點、一台機器（無鏡像節點對）、屬於同一 DRU 的一台或多台機器或甚至一個完整 DRU 上。但是，HADB **不會**自動從雙重故障回復，即一個或多個鏡像節點對同時發生的故障。如果發生雙重故障，則必須清除 HADB，並重新建立其作業事件存放區（清除其所有資料）。

有不同的維護程序，取決於您需要在台機器還是多台機器上工作。

▼ 在一台機器上執行維護

此程序適用於規劃的和未規劃的維護，不會中斷 HADB 可用性。

1 執行維護程序，並使機器啟動和執行。

2 確定 ma 正在執行。

如果 ma 以 Windows 服務的形式或在 `init.d` 程序檔（建議用於部署）下執行，則應該已由作業系統啟動。如果沒有，請手動啟動。請參閱第 50 頁的「啟動管理代理程式」。

3 啟動機器上的所有節點。

如需更多資訊，請參閱第 76 頁的「啟動節點」。

4 檢查節點是否為使用中且正在執行。

如需更多資訊，請參閱第 88 頁的「取得 HADB 的狀態」。

▼ 在所有 HADB 機器上執行規劃的維護

規劃的維護包括硬體和軟體升級等作業。此程序不會中斷 HADB 可用性。

- 1 對於第一個 DRU 中的每台備用機器，請逐台重複單一機器程序，如第 94 頁的「在一台機器上執行維護」中所述。
- 2 對於第一個 DRU 中的每台使用中機器，逐台重複單一機器程序，如第 94 頁的「在一台機器上執行維護」中所述。
- 3 為第二個 DRU 重複步驟 1 和步驟 2。

▼ 在所有 HADB 機器上執行規劃的維護

此程序在 HADB 位於一台或多台機器上時適用。在維護程序期間會中斷 HADB 服務。

- 1 停止 HADB。請參閱第 79 頁的「停止資料庫」。
- 2 執行維護程序，並使所有機器啟動和執行。
- 3 確定 ma 正在執行。
- 4 啟動 HADB。
如需更多資訊，請參閱第 78 頁的「啟動資料庫」。
完成最後一步後，HADB 資料再次變得可用。

▼ 發生故障時進行臨時維護

- 檢查資料庫狀態。
請參閱第 88 頁的「取得 HADB 的狀態」
 - 如果資料庫狀態為可作業或更佳：

需要執行未規劃維護的機器不包含鏡像節點。為每台出現故障的機器執行單一機器程序，每次一個 DRU。HADB 伺服器不中斷。
 - 如果資料庫狀態為不可作業：

需要執行未規劃維護的機器包含鏡像節點。此類情況之一為整個 HADB 都在一台出現故障的機器上時。首先啟動並執行所有機器。然後清除 HADB，並重新建立階段作業存放區。請參閱第 80 頁的「清除資料庫」。這會中斷 HADB 服務。

清除和歸檔歷史檔案

HADB 歷史檔案會記錄所有資料庫作業和錯誤訊息。HADB 附加至現有歷史檔案的結尾，因此檔案隨著時間而增長。若要節省磁碟空間並防止檔案變得過大，請定期清除和歸檔歷史檔案。

若要清除資料庫的歷史檔案，請使用 `hadbm clearhistory` 指令。

指令語法為：

```
hadbm clearhistory
[--saveto=path]
[dbname]
[--adminpassword=password | --adminpasswordfile=file]
[--agent=maurl]
```

dbname 運算元指定資料庫名稱。預設為 `hadb`。

使用 `--saveto` 選項 (短形式 `-o`) 可指定在其中儲存舊的歷史檔案之目錄。此目錄必須具有相應的寫入權限。請參閱第 59 頁的「一般選項」，以取得其他指令選項說明。

如需更多資訊，請參閱 `hadbm-clearhistory(1)`。

`hadbm create` 指令的 `--historypath` 選項可確定歷史檔案的位置。歷史檔案的名稱格式為 `dbname.out.nodeno`。如需有關 `hadbm create` 的資訊，請參閱第 62 頁的「建立資料庫」。

歷史檔案格式

歷史檔案中的各訊息包含以下資訊：

- 產生訊息之 HADB 程序的縮寫名稱。
- 訊息類型為：
 - INF - 一般資訊
 - WRN - 警告
 - ERR - 錯誤
 - DBG - 除錯資訊
- 時間戳記。時間為主機電腦系統時鐘時間。
- 節點停止或啟動時，在系統中發生服務設定變更。

有關資源短缺的訊息包含字串「HIGH LOAD」。

無需詳細瞭解歷史檔案中的所有項目。如果出於任何原因您需要更加詳細地研究歷史檔案，請與 Sun 客戶支援連絡。

配置 Web 伺服器以用於負載平衡

本章說明如何針對 Application Server 9.1 與 GlassFish v2 提供的負載平衡器外掛程式，配置所支援的 Web 伺服器。Application Server 9.1 提供的負載平衡器外掛程式支援下列 Web 伺服器：

- Sun Java System Web Server 6.1 及 7.0
- Apache Web Server 2.0.x
- Microsoft IIS 5.0 及 6.0

備註 – GlassFish v2 僅支援 Sun Java System Web Server (6.1 版及 7.0 版)。若要搭配 GlassFish v2 使用負載平衡器外掛程式，需要手動安裝與配置負載平衡器外掛程式。如需一起安裝負載平衡器外掛程式與 GlassFish v2 的更多資訊，請參閱「Sun Java System Application Server 9.1 Installation Guide」中的第 1 章「Installing Application Server Software」。

負載平衡器外掛程式安裝程式附屬於 Application Server 9.1 安裝程式，會對 Web 伺服器的配置檔案做一些修改。這些變更取決於您所使用的 Web 伺服器。此外，某些 Web 伺服器得進行手動配置，負載平衡器才能正確運作。

備註 – 負載平衡器外掛程式可與 Sun Java System Application Server 9.1 一起 (或單獨) 安裝在執行受支援 Web 伺服器的電腦上。如需安裝程序的完整詳細資訊，請參閱「Sun Java System Application Server 9.1 Installation Guide」中的第 1 章「Installing Application Server Software」。

- [第 98 頁的「配置 Sun Java System Web Server」](#)
- [第 105 頁的「使用 Apache Web Server」](#)
- [第 113 頁的「使用 Microsoft IIS」](#)

配置 Sun Java System Web Server

若為 Sun Java System Web Server，則當您使用 Sun Java System Application Server 9.1 安裝精靈安裝負載平衡器外掛程式時，安裝精靈會自動完成所有必要的配置。不需要進行手動配置。Application Server 9.1 隨附的負載平衡器外掛程式支援下列 Sun Java System Web Server 版本：

- Sun Java System Web Server 6.1
- Sun Java System Web Server 7.0

但若是使用 GlassFish v2，則必須單獨從 http://download.java.net/javaee5/external/SunOS_X86/aslb/jars/aslb-9.1-MS4-b7.jar 下載 Application Server 負載平衡器外掛程式，並進行部份手動變更來設定。如需如何安裝和設定 GlassFish v2 外掛程式的詳細步驟，請參閱「Sun Java System Application Server 9.1 Installation Guide」中的「To Install the Load Balancing Plug-in (standalone)」一節。

▼ 配置 Sun Java System Web Server

開始之前

備註 – Application Server 9.1 的安裝程式會自動執行下列步驟。但若是使用 GlassFish v2，則需要手動執行這些步驟。

- 1 前往 Web 伺服器實例的 `magnus.conf` 檔案，新增下列幾行：

```
##BEGIN EE LB Plug-in Parameters
Init fn="load-modules"
shlib="web-server-install-dir/plugins/lbplugin/bin/libpassthrough.so"
funcs="init-passthrough,service-passthrough,name-trans-passthrough" Thread="no"
Init fn="init-passthrough"
##END EE LB Plug-in Parameters=
```

- 2 若不存在下行，請予以新增：

```
Init fn="load-modules" shlib="../../../libj2eeplugin.so" shlib_flags="(global|now)"
```

- 3 在 `web-server-install-dir/config/obj.conf` 檔案中，在首次出現字串 `nametrans` 前，於單行中插入下列資訊：

```
Nametrans fn="name-trans-passthrough" name="lbplugin"
config-file="web-server-install-dir/config/loadbalancer.xml"
```

`NameTrans` 項目在 `obj.conf` 中的顯示順序非常重要。安裝程式會將 `NameTrans` 項目擺在正確的位置，但是如果您為了其他目的而編輯 `obj.conf`，則必須確定維持正確的順序。尤其是負載平衡器資訊必須位在 `document-root` 函數之前。如需有關 `obj.conf` 檔案的更多資訊，請參閱「Sun Java System Web Server 7.0 Administrator's Configuration File Reference」。

- 將下列幾行新增到 `web-server-install-dir/config/obj.conf` 檔案：

```
<Object name = "lbplugin">
  ObjectType fn="force-type" type="magnus-internal/lbplugin"
  PathCheck fn="deny-existence" path="*/WEB-INF/*"
  Service type="magnus-internal/lbplugin" fn="service-passthrough"
  Error reason="Bad Gateway" fn="send-error" uri="$docroot/badgateway.html"
</Object>
```

- 編輯 `web-server-install-dir/start` 程序檔，更新 `LD_LIBRARY_PATH` 值以併入 `app-server-install-dir/lib/lbplugin/lib`。
`app-server-install-dir/lib/lbplugin/lib` 目錄包含負載平衡器外掛程式所需的二進位碼。
- (可選擇) 若為新的 DAS 型負載平衡器管理，請配置 Web 伺服器的 SSL 功能。
如需 Web Server 6.1 的詳細說明，請參閱第 99 頁的「在 SSL 模式中為 Sun Java System Web Server 6.1 設定負載平衡器」。
如需 Web Server 7 的詳細說明，請參閱第 102 頁的「在 SSL 模式中為 Web Server 7 設定負載平衡器」。
- 若 Web 伺服器未執行，請啟動 Web 伺服器。

配置 Sun Java System Web Server 以使用自動套用

「自動套用」是 Application Server 9.1 提供的功能，可透過網路將負載平衡器配置自動傳送到 Web 伺服器配置目錄。如需此功能的更多資訊，請參閱第 117 頁的「自動套用」。下列程序說明如何配置 Sun Java System Web Server (版本 6 及 7) 來使用此功能。

▼ 在 SSL 模式中為 Sun Java System Web Server 6.1 設定負載平衡器

備註—唯有當您想使用負載平衡器外掛程式的「自動套用」功能時，才需要執行本節中的步驟。此功能可協助您透過網路，將負載平衡器配置自動傳送到 Web 伺服器配置目錄。

- 使用瀏覽器存取 Web 伺服器的管理主控台並登入。
- 選取您的伺服器實例，然後按一下 [管理]。
- 按一下 [安全性] 標籤。

- 4 提供使用者名稱和密碼來初始化信任資料庫。此動作可使用 `certutil` 指令或 GUI 來完成。 `certutil` 指令的下列選項可用來初始化信任資料庫：

```
certutil -N -P "https-instance-name-hostname-" -d .
```

- 當 `certutil` 提示您時，請輸入密碼來加密您的金鑰。輸入密碼來加密您的金鑰。密碼必須至少有八個字元長，且至少包含一個非字母字元。
- 當系統提示您輸入新密碼時，請指定密碼。

- 5 使用下列指令建立本機憑證授權單位 (CA) 範例：

```
certutil -S -P "https-boqueron.virkki.com-boqueron-"
-d . -n SelfCA -s "CN=Self CA,OU=virkki.com,C=US"
-x -t "TC,TC,TC" -m 101 -v 99 -5
```

- a. 當系統提示您輸入 0-7 來表示憑證類型時，請鍵入 5 表示 SSL CA。重新出現提示時，請指定 9。
- b. 出現查詢「Is this a critical extension [y/n]？」時，請指定「y」。

- 6 使用上述 CA 範例來產生憑證

```
certutil -S -P "https-instance-name-hostname-"
-d . -n MyServerCert -s "CN=boqueron.virkki.com,C=US"
-c SelfCA -t "u,u,u" -m 102 -v 99 -5
```

- a. 當系統提示您輸入 0-7 來表示憑證類型時，鍵入 1 表示 SSL 伺服器。重新出現提示時，請指定 9。
- b. 出現查詢「Is this a critical extension [y/n]？」時，請指定「y」。

- 7 依照下列步驟中的說明，建立 HTTPS 偵聽程式：

- a. 登入 Web 伺服器的管理伺服器。
- b. 選取伺服器，再按一下 [管理]。
- c. 按一下 [新增偵聽通訊端]。在 [新增偵聽通訊端] 頁面中，執行下列步驟：
 - i. 指定連接埠號。
 - ii. 確定在 [伺服器名稱] 中指定伺服器的完全合格的網域名稱 (FQDN)。例如，如果主機名稱是 `machine1`，且網域名稱是 `server.example.com`，則 FQDN 是 `machine1.server.example.com`。
 - iii. 從 [安全性] 下拉式清單中，選取 [啓用]。

- iv. 按一下[確定]。
- d. 前往[編輯偵聽通訊端]頁面，然後選取剛建立的[偵聽通訊端]。
- e. 在[偵聽通訊端]頁面中，確認[伺服器憑證]名稱與您在步驟6中提供的憑證名稱是否相同。

▼ 匯出及匯入 Sun Java System Web Server 6.1 的 DAS 憑證

- 1 若是使用 Application Server 9.1，請執行下列指令來匯出 DAS 憑證：

```
<appserver_install_dir>/lib/upgrade/pk12util -d <domain root>/config -o sjsas.pl2-W
<file password> -K <master password> -n slas
```

- 若是使用 GlassFish v2，則必須使用下列指令匯出 DAS 憑證：

```
<JAVA_HOME>/bin/keytool -export -rfc -alias slas -keystore
<GLASSFISH_HOME>/domains/<DOMAIN_NAME>/config/keystore.jks-file slas.rfc
```

其中，<GLASSFISH_HOME> 表示 Application Server 安裝目錄，<DOMAIN_NAME> 則表示正在匯出其憑證的網域。

- 將憑證檔案複製到 Web 伺服器配置目錄。

- 2 若是使用 Application Server 9.1，請使用下列指令，將 DAS 憑證匯入到 Web 伺服器實例：

```
<webserver_install_dir>/bin/https/admin/bin/pk12util -i sjsas.pl2 -d
<webserver_install_dir>/alias -W<file password> -K <webserver security db password> -P
<instance-name>-<hostname>-
```

```
<webserver_install_dir>/bin/https/admin/bin/certutil -M -n slas -t "TCu"
-d <webserver_install_dir>/alias -P <instance-name>-<hostname>-
```

這些指令可將 Application Server CA 變成受信任的 CA，以便簽署用戶端憑證和伺服器憑證。

- 若是使用 GlassFish v2，請從使用 certutil (NSS 安全性工具) 所建立的 rfc 檔案匯入 DAS 憑證。

```
<webserver_install_dir>/bin/certutil -A -a -n slas -t "TCu" -i slas.rfc
-d <webserver_install_dir>/alias -P <instance-name>-<hostname>-
```

您可以使用下列指令來檢查此憑證是否存在，它會列出 `slas` 憑證與其他 CA 憑證，包括預設的伺服器憑證。請務必將此指令在單獨一行中鍵入。

```
<WS_INSTALL_ROOT>/bin/certutil -L
-d <webserver_install_dir>/alias -P <instance-name>-<hostname>-
```

- 3 如果 `obj.conf` 不含下列各行，請在檔案尾端加入這些行。若是使用 **Application Server 9.1**，則安裝程式會自動執行此步驟。

```
<Object ppath="*lbconfigupdate*">
PathCheck fn="get-client-cert" dorequest="1" require="1"
</Object>
<Object ppath="*lbgetmonitordata*">
PathCheck fn="get-client-cert" dorequest="1" require="1"
</Object>
```

- 4 您可以使用第 112 頁的「**確認設定**」一節中所提供的步驟，確認上述設定。您可以改用任何其他 CA 和伺服器憑證取代本機 CA。在這種情況下，您可以略過上一節中所示的步驟 5 和步驟 6，但需要匯入自其他 CA 所取得的伺服器憑證。

在 SSL 模式中為 Web Server 7 設定負載平衡器

1. 使用下列指令，啟動 Web 伺服器的管理伺服器。

```
webserver-install-dir/admin_server/bin/startserv
```

2. 依照下列步驟中的說明，建立 HTTPS 偵聽程式。若已存在 HTTP 偵聽程式，則可略過下列步驟跳到第 103 頁的「**匯出及匯入 Sun Java System Web Server 7 的 DAS 憑證**」一節。
 - a. 登入 Web 伺服器管理主控台。
 - b. 選取預設配置。一般而言，預設配置名稱與主機名稱相同。若要從 [常用工作] 頁面執行此動作，請從 [選取配置] 清單中選取配置，然後按一下 [編輯配置]。或者，可以開啓 [配置] 頁面，並按一下 [配置] 表格中的預設配置名稱。
 - c. 若位於 [常用工作] 頁面，請按一下 [請求伺服器憑證]。反之，若是位於 [配置] 頁面，請開啓 [憑證] 頁面，然後按一下 [伺服器憑證] 表格的 [請求] 按鈕。這是為此預設配置建立自我簽署伺服器憑證的必要動作。
 - d. 提供 [請求伺服器憑證] 視窗所要求的詳細資訊。

執行此動作時，請務必在「*伺服器名稱 (cn)」中，提供安裝了 Web 伺服器的電腦之完全合格網域名稱 (FQDN)。例如，如果主機名稱是 `machine1`，且網域名稱是 `server.example.com`，則 FQDN 是 `machine1.server.example.com`。選取提供的預設值。

您也可以使用下列指令建立自我簽署的憑證。請務必將此指令在單獨一行中鍵入。

```
webserver-install-dir/bin/wadm create-selfsigned-cert --user=
admin-user --server-name=host-name
--nickname=ServerCert --token=internal --config=config-name
```

- e. 返回選取的配置頁面。
- f. 開啓 [HTTP 偵聽程式] 頁面，然後按一下 [新增] 按鈕。此動作可用來建立 SSL 啓用的 HTTP 偵聽程式。
- g. 提供「新增 HTTP 偵聽程式」精靈所找到的詳細資訊。確定伺服器名稱是在較早步驟中所提供的 FQDN。選取 [SSL] 按鈕，並從 [憑證] 清單中選取先前建立的伺服器憑證。例如，cert-machine1.server.example.com。

您也可以使用下列指令來建立 HTTP 偵聽程式。請務必將每一個指令在單獨一行中鍵入。

```
webserver-install-dir/bin/wadm create-http-listener
--user=admin-user --server-name=host-name
--default-virtual-server-name=default-virtual-server-name
--listener-port=8090 --config=config-name http-listener-ssl
```

```
webserver-install-dir/bin/wadm set-ssl-prop
--user=admin-user --http-listener=http-listener-ssl
--config=config-name enabled=true server-cert-nickname=ServerCert
```

- h. 執行以上列出的步驟後，您會在管理主控台右上角看到「部署擱置中」警示。按一下此警示，並遵循說明完成部署。此步驟可確保將 Web 伺服器的管理伺服器中所儲存的配置變更，複製到 Web 伺服器實例。

▼ 匯出及匯入 Sun Java System Web Server 7 的 DAS 憑證

藉由匯出和匯入 DAS 憑證，可將 DAS 變成 Web 伺服器的受信任用戶端。使用 DAS 憑證的用戶端認證可確保唯有連線到 Web 伺服器的 DAS 可做為受信任的用戶端。

- 1 開啓終端機視窗，並使用下列指令設定 LD_LIBRARY_PATH：

```
export LD_LIBRARY_PATH=/opt/SUNWappserver/lib
```

- 2 若是使用 **Application Server 9.1**，請執行下列指令來匯出 DAS 憑證。DAS 憑證可同時做為伺服器憑證以及用戶端憑證。

```
<appserver_install_dir>/lib/upgrade/pk12util -d <domain root>/config -o slas.pk12 -W
<slas.pk12-file-password> -K <master password> -n slas
```

- 若是使用 **GlassFish v2**，請匯出 DAS 憑證，並使用 **Java SE 5.0** 安全性工具 (名為 **keytool**)，為該憑證提供別名「**s1as**」。執行此動作時，請選取 **-rfc** 選項來匯出可列印編碼格式 (由 **Internet RFC 1421** 標準定義) 的憑證。

在指令行中，可使用下列指令匯出 DAS 憑證：

```
<JAVA_HOME>/bin/keytool -export -rfc -alias slas -keystore
<GLASSFISH_HOME>/domains/<DOMAIN_NAME>/config/keystore.jks-file slas.rfc
```

其中，<GLASSFISH_HOME> 表示 **Application Server** 安裝目錄，<DOMAIN_NAME> 則表示正在匯出憑證的網域。

- 將憑證檔複製到 **Web 伺服器配置目錄**。

- 3 若是使用 **Application Server 9.1**，請使用下列指令，將 DAS 憑證匯入 **Web 伺服器實例**並設定憑證的信任屬性：

```
<webserver_install_dir>/bin/pk12util -i <path_to_slas.pk12-file>
-d <webserver_install_dir>/admin-server/config-store/<default-config-name>/config
-K <webserver security db password> -W <slas.pk12-file-passwd>
```

```
<webserver_install_dir>/bin/certutil -M -n slas -t "TCu"
-d <webserver_install_dir>/admin-server/config-store/<default-config-name>/config
```

這些指令可使 **Application Server CA** 成為信任的 CA，以便簽署用戶端和伺服器憑證二者。

- 若是使用 **GlassFish v2**，請從使用 **certutil** (**NSS** 安全性工具) 所建立的 **rfc** 檔匯入 DAS 憑證。

```
<webserver_install_dir>/bin/certutil -A -a -n slas -t "TCu" -i slas.rfc -d
<webserver_install_dir>/admin-server/config-store/<CONFIG_NAME>/config
```

其中，<webserver_install_dir> 表示 **Web 伺服器** 安裝目錄，<CONFIG_NAME> 則表示為預設 **Web 伺服器實例** 所建立的配置名稱。

您可以使用下列指令來檢查是否存在此憑證，它會列出 **s1as** 憑證與其他 CA 憑證，包括預設的伺服器憑證。請務必將此指令在單獨一行中完整鍵入。

```
<webserver_install_dir>/bin/certutil -L -d
<webserver_install_dir>/admin-server/config-store/
<DEFAULT_CONFIG_NAME>/config
```

您也可以使用 **Web 伺服器管理主控台** 來檢視此憑證。選取已匯入憑證的配置 (在此情況下為預設配置)，然後選取 [憑證] 標籤。若要查看所有可用的憑證，請選取 [憑證授權單位] 子標籤。

- 4 若是使用 GlassFish v2，請為 Web Server 7 執行下列配置變更。若是使用 Application Server 9.1，則可跳至下一個步驟。
 - a. 將下列幾行附加到
`<WS_INSTALL_ROOT>/admin-server/config-store/<DEFAULT_CONFIG_NAME>/config/` 的 `obj.conf` 檔案。確定鍵入這些指令時沒有尾隨空格。


```
<Object ppath="*lbconfigupdate*">
  PathCheck fn="get-client-cert" dorequest="1" require="1"
</Object>
<Object ppath="*lbgetmonitordata*">
  PathCheck fn="get-client-cert" dorequest="1" require="1"
</Object>
```
- 5 部署配置。執行在前幾個步驟中所列出的變更時，管理主控台會將此配置標記為已部署。
 - a. 在 [Web 伺服器管理主控台] 中選取 [部署擱置中] 圖示。您也可以使用 CLI 公用程式 `wadm` 來部署此配置，如下所示：


```
<webserver_install_dir>/bin/wadm deploy-config --user=<admin> <DEFAULT_CONFIG_NAME>
```
- 6 從 GlassFish DAS 測試此設定，查看其是否可透過 SSL 與已配置的 HTTP 負載平衡器進行通訊。如需更多資訊，請參閱第 112 頁的「確認設定」。

使用 Apache Web Server

Application Server 9.1 隨附的負載平衡器外掛程式支援 Apache Web Server 2.0.x。若要使用 Apache Web Server，必須在安裝負載平衡器外掛程式之前和之後執行特定的配置步驟。負載平衡外掛程式安裝也會對 Apache Web Server 做出其他修改。安裝外掛程式後，必須執行其他配置步驟。

備註 – 如果使用 `--with-mpm=worker` 選項進行編譯，則 Apache 2 將具有多重執行緒運作方式。

- 第 106 頁的「使用 Apache Web Server 的需求」
- 第 107 頁的「在安裝負載平衡外掛程式之前先配置 Apache」
- 第 110 頁的「負載平衡外掛程式安裝程式所做的修改」
- 第 111 頁的「安裝負載平衡外掛程式後配置 Apache」
- 第 112 頁的「在 Solaris 和 Linux 上啟動 Apache」

使用 Apache Web Server 的需求

若為 Apache Web Server，則您的安裝必須符合最低需求。

對於 Apache，負載平衡外掛程式需要：

- openssl-0.9.7e (原始碼)
- httpd-2.0.59 (原始碼)
- gcc-3.3-sol9-sparc-local 套裝軟體 (用於 Solaris 9 SPARC)。
- gcc-3.3-sol9-intel-local 套裝軟體 (用於 Solaris 9 x86)
- 預先安裝的 gcc (用於 Solaris 10)
- flex-2.5.4a-sol9-sparc-local 套裝軟體 (用於 Solaris 9 SPARC)
- flex-2.5.4a-sol9-intel-local 套裝軟體 (用於 Solaris 9 x86)
- 預先安裝的 flex (用於 Solaris 10)

軟體原始碼可從 <http://www.sunfreeware.com> 取得。

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 9 作業系統上，使用 pkgadd 以安裝 gcc 和 flex。請注意，pkgadd 需要超級使用者存取權限。
- 在 Solaris 9 作業系統上，請確保 gcc 3.3 版和 make 位於 PATH 中，並且已安裝了 flex。
- 在 Solaris 10 作業系統上，對 OpenSSL 執行 make 之前，請先執行位於 /usr/local/lib/gcc-lib/sparc-sun-solaris2.9/3.3/install-tools (在 Solaris SPARC 上) 或位於 /usr/local/lib/gcc-lib/i386-pc-solaris2.9/3.3/install-tools (在 Solaris x86 上) 中的 mkheaders。
- 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 – 如果除了使用 gcc 之外還要使用 C 編譯器，請在 PATH 環境變數中設定 C 編譯器和 make 公用程式的路徑。

套用 Apache Web Server 修補程式

安裝 Apache 的負載平衡器外掛程式之前，請為 Apache Web Server 問題 12355 套用修補程式。如需此問題的詳細資訊，請造訪

http://issues.apache.org/bugzilla/show_bug.cgi?id=12355。需要此修補程式才能使用「自動套用」功能。若要套用修補程式，請執行下列步驟。

1. 解壓縮 http-2.0.59.tar 並前往目錄 httpd-2.0.59。
2. 從 <http://issues.apache.org/bugzilla/attachment.cgi?id=16495> 下載修補程式，並將其另存為檔案，例如，12355.diff。

3. 從 `httpd-2.0.59/modules/ssl` 目錄中，執行下列指令：

```
patch < 12355.diff
```

在安裝負載平衡外掛程式之前先配置 Apache

必須編譯和建置 Apache 原始碼以使用 SSL 執行。本小節說明如欲順利編譯 Apache Web Server 以執行負載平衡器外掛程式，所需的最低需求和概略步驟。這些需求和步驟僅適用於此軟體的 Solaris 和 Linux 版本。如需有關 Apache 的 Windows 版本的資訊，請參閱 Apache 網站。

備註 – 在此提供的指示改編自 <http://httpd.apache.org/docs> 的指示內容。如需有關顧及 SSL 設定的 Apache 之詳細安裝指示，請參閱該網站。

▼ 安裝 SSL 可識別的 Apache

開始之前 您必須已下載並解壓縮了 Apache 軟體。

1 下載並解壓縮 OpenSSL 原始碼，原始碼可從 <http://openssl.org> 取得。

2 編譯和建置 OpenSSL。

如需完整的安裝指示，請參閱在您解壓縮 OpenSSL 的目錄內，名為 `INSTALL` 的檔案。該檔案說明如何在使用者指定的位置安裝 OpenSSL。

如需有關 OpenSSL 的更多資訊，請參閱 <http://www.openssl.org/>。

3 下載及解壓縮 Apache。

Apache 可從 <http://httpd.apache.org> 取得。

4 編譯及建立 Apache。配置來源樹狀結構：

a. `cd http-2.0_x.`

b. 執行以下指令：

```
./configure --with-ssl= OpenSSL-install-path --prefix= Apache-install-path
--enable-ssl --enable-so
```

在上述指令中，*x* 為 Apache 的版本編號，*open-ssl-install-path* 為安裝 OpenSSL 的目錄絕對路徑，*Apache-install-path* 為要在其中安裝 Apache 的目錄。

請注意，僅當您的 Apache 2 伺服器要接受 HTTPS 請求時，才需要使用 `--enable-ssl` `--enable-so` 選項。

5 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/application-server-install-dir/lib -L/opt/sun/private/lib
```

請注意，只有將 Application Server 安裝至 Java Enterprise System 底下時，才需要 `-L/opt/sun/private/lib`。

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/application-server-install-dir /lib -L/opt/sun/private/lib
```

c. 設定環境變數 `LD_LIBRARY_PATH`。

對於獨立式安裝，將此變數設為 Application Server：`as-install/lib`

對於 Java Enterprise System 安裝，將此變數設為 Application Server：`as-install/lib:opt/sun/private/lib`。

若您使用 Solaris 9，請將 `/usr/local/lib` 增加至 `LD_LIBRARY_PATH`。

6 按照適用於您所使用之版本的安裝說明編譯 Apache。

如需更多資訊，請參閱 <http://httpd.apache.org/>。

一般來說，這些步驟包括：

a. `make`

b. `make install`

7 請確定 Apache 的 `ssl.conf` 和 `httpd.conf` 檔案包含適用於作業環境的正確值。

- 在 `ssl.conf` 中，對於 `VirtualHost default:port`，請以安裝 Apache 的本機系統主機名稱和伺服器連接埠號，取代預設主機名稱和連接埠號。

若不做此變更，負載平衡器將無法運作。在 Solaris 上，Apache 可能無法啟動，在 Linux 上，HTTPS 請求可能無法運作。

- 在 `ssl.conf` 中，對於 `ServerName www.example.com:443`，請以安裝 Apache 的本機系統主機名稱取代 `www.example.com`。

若不做此變更，如果有安裝安全憑證，在啟動 Apache 時會出現下列警告：

```
[warn] RSA server certificate CommonName (CN)
hostname does NOT match server name!
```

如需有關安裝 Apache 憑證的更多資訊，請參閱第 111 頁的「為 Apache 建立安全性憑證」。

- 在 `httpd.conf` 中，對於 `ServerName www.example.com:80`，請以安裝 Apache 的本機系統主機名稱取代 `www.example.com`。

若沒有做此變更，則啟動 Apache 時您會看到的警告訊息，包括系統無法判定伺服器的完全合格網域名稱，以及有重疊的 `VirtualHost` 項目。

8 確定 Apache 使用者擁有 `apache-install-location/conf/` 目錄及此目錄中檔案的必要存取權限。

Apache 使用者是 Apache 伺服器用來回應請求的 UNIX 使用者身份。此使用者是在 `httpd.conf` 檔案中定義。

若您以超級使用者身份安裝 Apache，請參閱 `apache-install-location/conf/httpd.conf` 中，有關如何配置 Apache 使用者及群組的備註。

備註 – 請確定您的使用者及群組配置符合此目錄的安全性需求。例如，若要限制存取此目錄，請將 Apache 使用者新增到目錄所有者所屬的使用者群組。

- 若要確保「自動套用」功能正確運作，請將 `apache-install-location/conf/` 目錄的讀取、寫入及執行存取權授予 Apache 使用者。
 - 如果 Apache 使用者與此目錄的所有者位於相同的群組，請將模式變為 `775`。
 - 如果 Apache 使用者與此目錄的所有者位於不同的群組，請將模式變為 `777`。
- 若要在 Apache 啟動時確保負載平衡器外掛程式已初始化，請將下列檔案的讀取存取權和寫入存取權授予 Apache 使用者：
 - `apache-install-location /conf/loadbalancer.xml`
 - `apache-install-location /conf/sun-loadbalancer_1_2.dtd`

匯出及匯入 DAS 憑證

您必須使用下列指令來手動匯出 DAS 憑證：

```
appserver-install-dir/lib/upgrade/certutil -L -d appserver-instance-dir/config -n slas -a -o sjsas.crt
```

安裝負載平衡器外掛程式時需要使用此憑證。

Application Server 9.1 安裝程式會為您執行下列作業。

- 將 `sjsas.crt` 複製到 `apache-install-dir/conf/ssl.crt` 目錄時，匯入 DAS 憑證。
- 在 `httpd.conf` 中附加下列指令行。

```
<Location /lbconfigupdate>
SSLVerifyClient require
SSLVerifyDepth 1
SSLRequireSSL
SSLCACertificateFile apache-install-dir/conf/ssl.crt/sjsas.crt
SSLRequire ( %{SSL_CIPHER} !~ m/^(EXP|NULL)-/ \
and %{SSL_CLIENT_S_DN_O} eq "Sun Microsystems" \
and %{SSL_CLIENT_S_DN_OU} eq "Sun Java System Application Server" \
and %{SSL_CLIENT_M_SERIAL} eq "<*serial number*>" )
</Location>

<Location /getmonitordata>
SSLVerifyClient require
SSLVerifyDepth 1
SSLRequireSSL
SSLCACertificateFile apache-install-dir/conf/ssl.crt/sjsas.crt
SSLRequire ( %{SSL_CIPHER} !~ m/^(EXP|NULL)-/ \
and %{SSL_CLIENT_S_DN_O} eq "Sun Microsystems" \
and %{SSL_CLIENT_S_DN_OU} eq "Sun Java System Application Server" \
and %{SSL_CLIENT_M_SERIAL} eq "<*serial number*>" )
</Location>
```

負載平衡外掛程式安裝程式所做的修改

負載平衡外掛程式安裝程式會將必要的檔案，擷取至 Web 伺服器根目錄的 `modules` 目錄中：

安裝程式會將以下項目增加至 Web 伺服器實例的 `httpd.conf` 檔案：

```
##BEGIN EE LB Plugin Parameters
LoadModule apachelbplugin_module modules/mod_loadbalancer.so
#AddModule mod_apachelbplugin.cpp
<IfModule mod_apachelbplugin.cpp>
    config-file webserver-instance/httpd/conf/loadbalancer.xml
    locale en
</IfModule>
<VirtualHost machine-ip-address>
    DocumentRoot "webserver-instance/httpd/htdocs"
    ServerName server-name
</VirtualHost>
##END EE LB Plugin Parameters
```

安裝負載平衡外掛程式後配置 Apache

Apache Web Server 需要有正確的安全性檔案才能使用負載平衡外掛程式。負載平衡器依賴需要這些安全性資料庫檔案的 NSS (Network Security Service) 程式庫。這些安全資料庫檔案必須自 Application Server 取得，因此 Application Server 必須安裝於 Web 伺服器可存取的位置。

若要配置 Apache 安全性檔案來使用負載平衡器，請執行下列步驟：

將 `/usr/lib/mps` 附加至 `Apache-install-dir/bin/apachectl` 程序檔中的 `LD_LIBRARY_PATH`。

▼ 為 Apache 建立安全性憑證

在 Apache 上需要這些步驟來支援 HTTPS 請求。

如需有關在 Apache 上設定安全性憑證的詳細資訊，請參閱

http://httpd.apache.org/docs/2.2/ssl/ssl_faq.html 及

http://www.modssl.org/docs/2.8/ssl_faq.html 上的說明。下列程序改編自上述網站。

1 設定下列環境變數：

`OPENSSL_CONF=OpenSSL-installation-directory /apps/openssl.cnf.`

2 執行以下指令以建立伺服器憑證及金鑰：

`openssl req -new -x509 -keyout newreq.pem -out newreq.pem -days 365`

系統要求一般名稱時，請提供您打算在其中執行 Apache 的主機名稱。對於所有其他的提示，請輸入符合特定需求的值。

此指令會建立 `newreq.pem`。

3 自執行 `openssl` 指令的位置開啓剛建立的 `newreq.pem`。

4 複製從 `BEGIN CERTIFICATE` 開始到 `END CERTIFICATE` 為止的指令行，並貼到

`Apache-install-dir/conf/ssl.crt/server.crt` 中。例如：

```
-----BEGIN CERTIFICATE-----
....
...
-----END CERTIFICATE-----
```

5 複製從 `BEGIN RSA PRIVATE KEY` 開始到 `END RSA PRIVATE KEY` 為止的指令行，並貼到

`Apache-install-dir/conf/ssl.key/server.key` 中。例如：

```
-----BEGIN RSA PRIVATE KEY-----
...
...
...
-----END RSA PRIVATE KEY-----
```

- 6 請確定在 *Apache-install-dir/conf/ssl.conf* 中的變數 `SSLCertificateKeyFile` 和 `SSLCertificateFile` 都具有正確值。
- 7 請確定 `ServerName` 不是 `www.example.com`。`ServerName` 應是要執行 Apache 的實際主機名稱，並與您在建立伺服器憑證及金鑰時輸入的一般名稱相同。

修改 `httpd.conf` 參數以啓用居留式循環

若要使用居留式循環功能，請在 `httpd.conf` 檔案的 `prefork MPM` 區段下，確定將 `StartServers` 及 `maxclients` 參數值設為 1。否則，每個新的階段作業請求都會產生新的 Apache 程序，且將初始化負載平衡器外掛程式，導致請求被傳送到相同的實例。

在 Solaris 和 Linux 上啓動 Apache

通常啓動 Apache 時所用的使用者身份，應與安裝 Application Server 時所用的身份相同。只有在下列情況時，才必須以超級使用者身份啓動 Apache：

- 若您是 Java Enterprise System 使用者。
- 若您使用的連接埠號小於 1024。
- 若執行 Apache 的使用者不同於啓動 Apache 的使用者。

若要以 SSL 模式啓動 Apache，請使用下列其中一個指令：

`apachectl startssl` 或 `apachectl -k start -DSSL`

必要的話，請查看 Apache 網站上有關啓動 Apache 伺服器的最新資訊。

確認設定

1. 安裝負載平衡外掛程式。如需安裝外掛程式的詳細步驟，請參閱「Sun Java System Application Server 9.1 Installation Guide」。在安裝過程中，提供 DAS 憑證的路徑。
2. 登入「Application Server Admin Console」並建立新叢集。如需建立新叢集的步驟，請參閱管理主控台線上說明。
3. 建立新的 HTTP 負載平衡器。建立負載平衡器時，請將 Web 伺服器主機的 FQDN 指定為裝置主機名稱，將 Web 伺服器的「SSL 連接埠」指定為裝置連接埠，再將前一個步驟中所建立的叢集選為目標。如需建立新的 HTTP 負載平衡器的詳細步驟，請參閱管理主控台線上說明。
4. 若要確認 DAS 與 Web 伺服器之間的通訊是否正常運作，請在「管理主控台」中，瀏覽到 [HTTP 負載平衡器] 節點並按一下 [HTTP 負載平衡器]。在出現的 [負載平衡器裝置設定] 頁面中，按一下 [測試連線] 按鈕。

若未在建立負載平衡器時啓用 [自動套用變更] 選項，則必須前往 [匯出] 標籤並按一下 [立即套用變更]，來手動匯出負載平衡器配置。

5. 若測試連線失敗，請務必檢查 Application Server 網域記錄及 Web 伺服器記錄來疑難排解問題。另請檢查是否已正確執行所有配置步驟。

使用 Microsoft IIS

若要搭配使用 Microsoft Internet Information Services (IIS) 與負載平衡器外掛程式，請執行這些章節中所提供的步驟。

▼ 配置 Microsoft IIS 以使用負載平衡外掛程式

- 1 開啓網際網路服務管理程式。
- 2 選取您要為之啓用外掛程式的網站。
此網站通常命名為「預設網站」。
- 3 在網站上按一下滑鼠右鍵，選取 [特性] 以開啓 [特性] 筆記本。
- 4 依循以下步驟，增加新的 ISAPI 篩選器：
 - a. 開啓 [ISAPI 篩選器] 標籤。
 - b. 按一下 [新增]。
 - c. 在 [篩選器名稱] 欄位中，輸入 **Application Server**。
 - d. 在 [可執行] 欄位中，鍵入
`C:\Inetpub\wwwroot\sun-passthrough\sun-passthrough.dll`。
 - e. 按一下 [確定]，關閉 [特性] 筆記本。
- 5 建立並配置新的虛擬目錄：
 - a. 在預設網站上按一下滑鼠右鍵，選取 [新建]，然後選取 [虛擬目錄]。
[虛擬目錄建立精靈] 將開啓。
 - b. 在 [別名] 欄位中，鍵入 **sun-passthrough**。
 - c. 在 [目錄] 欄位中，鍵入 `C:\Inetpub\wwwroot\sun-passthrough`。
 - d. 核取 [執行許可權] 核取方塊。
使與許可權相關的所有其他核取方塊保持未核取狀態。

- e. 按一下[完成]。
- 6 將 `sun-passthrough.dll` 檔案的路徑、**Application Server** `as-install/bin` 和 **Application Server** `as-install/lib` 增加至系統的 PATH 環境變數中。
- 7 對於 IIS 6.0 使用者，請依下列步驟配置負載平衡器網頁服務延伸，以在 IIS 6 中執行：
 - a. 在 IIS 管理程式中展開本機電腦，並按一下[網頁服務延伸模組]。
 - b. 在[工作]窗格中，選取[新增網頁服務延伸]。
 - c. 輸入 **Sun-Passthrough** 作為延伸名稱，並按一下[新增]。
 - d. 將路徑 `C:\Inetpub\wwwroot\sun-passthrough` 鍵入到 `sun-passthrough.dll`。
 - e. 按一下[確定]。
 - f. 選取[設定延伸狀態成允許]。
- 8 對於 IIS 6.0 使用者，建立檔案 `C:\inetpub\wwwroot\sun-passthrough\lb.log`，並將 NTFS 寫入及修改權限給予檔案上的群組 `IIS_WPG`。
IIS 6.0 以工作者處理序隔離模式執行，因此它會以群組 `IIS_WPG` 的安全性權限執行 IIS 伺服器。
- 9 所有的 IIS 使用者請重新啟動機器。
- 10 確認 Web 伺服器、負載平衡器外掛程式和 **Application Server** 是否正常作業。
在 Web 瀏覽器中鍵入以下 URL，以存取 Web 應用程式環境根目錄：`http://web-server-name/web-application`，其中 `web-server-name` 為 Web 伺服器的主機名稱或 IP 位址，而 `web-application` 為在 `C:\Inetpub\wwwroot\sun-passthrough\sun-passthrough.properties` 檔案中列出的環境根目錄。

提示 – ISAPI 篩選器狀態應為綠色。若要檢查篩選器狀態，請存取網站的 [特性] 筆記本，並按一下 [ISAPI 篩選器] 標籤。若狀態不是綠色，請試著傳送任意 HTTP 請求到 IIS HTTP 連接埠。即使請求失敗也無妨。重新檢查 ISAPI 篩選器狀態。

自動配置的 **sun-passthrough** 特性

安裝程式會自動在 `sun-passthrough.properties` 中配置下列特性。您可以變更預設值。

特性	定義	預設值
lb-config-file	負載平衡器配置檔案的路徑	IIS-www-root\sun-passthrough \loadbalancer.xml
log-file	負載平衡器記錄檔的路徑	IIS-www-root\sun-passthrough\lb.log
log-level	Web 伺服器的記錄層級	資訊

備註 – Application Server 9.1 的「自動套用」功能目前不受 IIS 支援。

配置 HTTP 負載平衡

本章說明 HTTP 負載平衡器外掛程式。其中包含以下主題：

- 第 117 頁的「負載平衡器外掛程式的新增功能」
- 第 119 頁的「HTTP 負載平衡器的工作方式」
- 第 120 頁的「設定 HTTP 負載平衡」
- 第 124 頁的「配置負載平衡器」
- 第 136 頁的「配置多個 Web 伺服器實例」
- 第 137 頁的「在維持可用性的情況上升級應用程式」
- 第 143 頁的「監視 HTTP 負載平衡器外掛程式」

如需有關其他類型負載平衡的資訊，請參閱第 10 章和第 11 章。

本節討論 Application Server 內隨附之 HTTP 負載平衡器外掛程式的使用方式。另一個 HTTP 負載平衡選項會搭配 Application Server 使用 Sun Secure Application Switch，做為以硬體為基礎的負載平衡解決方案。如需有關配置此解決方案的指導文件，請參閱文章 [Clustering and Securing Web Applications: A Tutorial \(http://developers.sun.com/prodtech/appserver/reference/techart/load-balancing.html\)](http://developers.sun.com/prodtech/appserver/reference/techart/load-balancing.html)。

負載平衡器外掛程式的新增功能

Sun Java System Application Server 9.1 已經增強負載平衡器的功能，將透過下列功能提供更大的彈性且更容易使用。

自動套用

Application Server 可將您從「管理主控台」對負載平衡器配置所做的變更，自動透過網路傳送到 Web 伺服器配置目錄。使用舊版的應用程式伺服器時，必須先匯出負載平衡器配置，然後再複製到 Web 伺服器配置目錄。

加權循環

負載平衡器可改善 HTTP 請求的分配機制。管理員可使用「weight」屬性指定如何按比例將請求路由至實例。例如，假設叢集有兩個實例，且管理員將權數 100 指定給實例 x，並將權數 400 指定給實例 y。現在於每 100 個請求中，20 個請求將前往實例 x，80 個請求則前往實例 y。

使用者定義的負載平衡

Application Server 可讓管理員定義自訂策略來分配 HTTP 請求。自訂策略可以定義負載平衡器外掛程式必須使用的負載平衡演算法。換言之，管理員可定義哪個 Application Server 實例將處理 HTTP 請求。若要使用此功能，管理員需要開發共用程式庫 (例如，使用該程式庫來計算提供給它的內送請求標頭)，並遵守特定條件選取可處理此請求的實例。此共用程式庫將由負載平衡器載入。

共用程式庫必須實作 `loadbalancer.h` (可從 `appserver_install_dir/lib/install/templates` 下取得) 中定義的介面。

Application Server 也隨附 `roundrobin.c` 模組範例，此模組可實作基本的循環演算法。管理員可將此模組範例當成範本，以建立共用程式庫。此模組範例也可從 `appserver_install_dir/lib/install/templates` 下取得。

▼ 配置使用者定義的負載平衡

- 1 將 `roundrobin.c` 從 `appserver_install_dir/lib/install/templates` 複製到工作目錄 (例如：`/home/user/workspacelb`)。
- 2 使用 ANSI C/C++ 編譯器 (例如 Sun Studio 編譯器或 GCC) 編譯 `roundrobin.c`。請務必建立動態的共用程式庫，而非靜態的可執行檔。

- a. 如果使用「Sun Studio CC 編譯器」，請使用下列指令來編譯：

```
cc -G -I<appserver install dir>/lib/install/templates roundrobin.c -o roundrobin.so
```

- b. 如果使用 GCC，請使用此指令編譯共用程式庫：

```
gcc -shared -I<appserver install dir>/lib/install/templates  
roundrobin.c -o roundrobin.so
```

備註 – 如果發生重新配置錯誤，請使用「-fPIC」選項重新編譯。此指令格式為：

```
gcc -shared -fPIC -I <appserver install dir>/lib/install/templates
roundrobin.c -o roundrobin.so
```

在 Microsoft Windows 上，請從 <http://www.redhat.com/services/custom/cygwin> 下載 Cygwin 公用程式。此公用程式隨附 GCC。使用下列 GCC 指令可建立動態連結程式庫 (dll)：

```
gcc -shared -I<appserver_install_dir>/lib/install/templates
roundrobin.c -o roundrobin.dll
```

- 3 變更 loadbalancer.xml 以指向新建立的模組。這是編輯後的 loadbalancer.xml 外觀。

```
<cluster name="cluster1" policy="user-defined"
policy-module="home/user/workspace1b/roundrobin.so">
```
- 4 將 roundrobin.so 複製到 Web 伺服器實例目錄。
- 5 如果 Web 伺服器未執行，請予以啟動，或等到負載平衡器重新配置。

HTTP 負載平衡器的工作方式

負載平衡器嘗試在多個 Application Server 之獨立或叢集的實例間平均分散工作負荷量，從而增加了系統的整體流量。

HTTP 負載平衡器可使部署到 Java EE 應用程式伺服器的服務具有高可用性。此舉能在偵測到無法使用原始提供服務的實例，或者此實例運作不正常無法處理請求時，將階段作業請求容錯移轉至另一個伺服器實例。如果要使 HTTP 階段作業資訊持續運作，您必須使用叢集設定檔、安裝並設定 HADB，同時配置 HTTP 階段作業的持續性。如需更多資訊，請參閱第 9 章。

備註 – 負載平衡器不處理大於 8K 的 URI/URL。

HTTP 負載平衡演算法

Sun Java System Application Server 負載平衡器預設會使用**居留式循環演算法**對內送 HTTP 和 HTTPS 請求進行負載平衡。

新 HTTP 請求傳送到負載平衡器外掛程式後，系統將基於簡單的循環方案，將該請求轉寄至某個 Application Server 實例。如果是階段作業型應用程式的請求，則此類請求也將包括新階段作業的請求。相同用戶端對相同的階段作業型應用程式所提出的後續請

求，都會視為指定或居留式請求，且負載平衡器會將該請求路由至相同的實例。因此稱為居留式循環。非階段作業型應用程式請求以及階段作業型應用程式的第一個請求，稱為未指定請求。居留特性是透過 Cookie 或明確採用 URL 重寫功能達成。負載平衡器會自動決定居留方法。

負載平衡器外掛程式使用以下方法來決定階段作業居留：

- **Cookie 方法：**負載平衡器外掛程式使用獨立的 Cookie 來記錄路由資訊。HTTP 用戶端 (通常是 Web 瀏覽器) 必須支援 Cookie，才能使用基於 Cookie 的方法。如果 HTTP 用戶端無法接受 Cookie，則外掛程式會使用下列方法。
- **明確的 URL 重新寫入：**居留式資訊會附加至 URL。即使 HTTP 用戶端不支援 Cookie，也可以使用此方法。

負載平衡器外掛程式首先從居留式資訊確定此請求先前轉寄到哪個實例。如果發現該實例正常，負載平衡器外掛程式會將請求轉寄至該特定 Application Server 實例。因此，指定階段作業的所有請求都將會傳送到同一個 Application Server 實例。

設定 HTTP 負載平衡

本節說明如何設定負載平衡器外掛程式，其中包含以下小節：

- [第 120 頁的「設定負載平衡的先決條件」](#)
- [第 121 頁的「設定負載平衡的程序」](#)
- [第 123 頁的「HTTP 負載平衡器部署」](#)

設定負載平衡的先決條件

配置負載平衡器之前，您必須：

- 安裝與配置受支援的 Web 伺服器。如需有關配置受支援的 Web 伺服器的更多資訊，請參閱[第 4 章](#)。
- 安裝負載平衡器外掛程式。
如需有關安裝程序的更多資訊，請參閱「Sun Java System Application Server 9.1 Installation Guide」。
- 建立要參與負載平衡的 Application Server 叢集或伺服器實例。
- 將應用程式部署至這些叢集或實例。

備註 – 如果您的部署方案是將 Application Server 實例及負載平衡器安裝在不同網路網域，則必須使用 `--agentproperties` 選項來指定完全合格的網域名稱，以建立節點代理程式。例如，`asadmin create-node-agent --agentproperties remoteclientaddress=machine1.server.example.com test-na`。如需有關此指令的更多資訊，請參閱 `create-node-agent(1)`。

設定負載平衡的程序

使用管理主控台 GUI 或 `asadmin` 工具可在您的環境中配置負載平衡。下列各節提供了更多資訊。

▼ 使用管理主控台設定負載平衡

1 建立負載平衡器配置。

在管理主控台的左框架中，依序按一下 [HTTP 負載平衡器] 及 [新建]。在 [新建 HTTP 負載平衡器] 頁面中，提供裝置詳細資訊並選取目標叢集或實例。

2 針對要受負載平衡器管理的叢集或獨立伺服器實例新增參照。

若要使用管理主控台執行此動作，請在左框架中，按一下 [HTTP 負載平衡器] 節點，再於節點下按一下所需的負載平衡器。開啓 [目標] 標籤、按一下 [管理目標]，然後在 [管理目標] 頁面上選取所需的目標。

如果您建立了具有目標的負載平衡器配置，並且該目標是負載平衡器參照的唯一叢集或獨立伺服器實例，請跳過此步驟。

3 啓用負載平衡器所參照的叢集或獨立伺服器實例。

若要使用管理主控台啓用獨立伺服器實例，請在左框架中，按一下 [HTTP 負載平衡器] 節點，再於節點下按一下所需的負載平衡器。開啓 [目標] 標籤，然後在 [目標] 表格中，按一下要啓用的實例旁的核取方塊，再按一下 [啓用]。

若要啓用叢集中的伺服器實例，請如上所述選取負載平衡器，再於 [目標] 標籤中按一下所需的叢集。現在請開啓 [實例] 標籤、選取所需的實例，然後從 [負載平衡器動作] 下拉式清單中選取 [啓用負載平衡]。

`asadmin enable-http-lb-server` 是啓用叢集或獨立實例的等效指令。

4 啓用需負載平衡的應用程式。

若要使用管理主控台執行此動作，請如上所述開啓 [目標] 標籤，然後按一下所需的叢集。現在請開啓 [應用程式] 標籤，選取所需的應用程式，然後從 [更多動作] 下拉式清單中選取 [負載平衡器啓用]。

這些應用程式必須已在負載平衡器所參照的叢集或獨立實例上部署並可以使用。啓用應用程式以用於負載平衡，與啓用應用程式以加以使用，是兩個獨立的步驟。

5 建立運作狀態檢查程式

若要使用管理主控台執行此動作，請為上一個步驟中所述的負載平衡器開啓 [目標] 標籤，然後按一下 [目標] 表格中的 [編輯運作狀態檢查程式]。

運作狀態檢查程式監視工作異常的伺服器實例，以便在這些伺服器實例重新正常工作時，負載平衡器可以向它們傳送新的請求。

備註 – 如果使用 Sun Java System Web Server (6.1 或 7.0)，則您不必執行步驟 6 及步驟 7，則可用一個步驟就產生負載平衡器配置檔案，並透過網路將資料傳送到 Web 伺服器。

若要使用管理主控台執行此動作，請按一下所需的負載平衡器，然後開啓 [匯出] 標籤。在此標籤中，按一下 [立即套用變更]。如此會將資料傳送到 Web 伺服器配置目錄。

6 產生負載平衡器配置檔案。

若要使用管理主控台執行此動作，請按一下負載平衡器，然後開啓 [匯出] 標籤。在此標籤中，按一下 [立即匯出]。

此指令可產生要與 Sun Java System Application Server 隨附之負載平衡器外掛程式配合使用的配置檔案。

7 將負載平衡器配置檔案複製到 Web 伺服器的 config 目錄，該目錄中儲存了負載平衡器外掛程式配置檔案。

備註 – 若要在一個步驟中，自動產生負載平衡器配置檔案，並透過網路將資料傳送給 Web 伺服器，您需要配置 Web 伺服器的 SSL 設定，並匯入 DAS 憑證。如需有關配置 Sun Java System Web Server 的更多資訊，請參閱第 98 頁的「[配置 Sun Java System Web Server](#)」。

▼ 使用 asadmin 工具設定負載平衡

1 建立負載平衡器配置。

若要執行此動作，請使用 `asadmin create-http-lb-config` 指令。

備註 – 您只需使用一個 `asadmin` 指令 (`create-http-lb`) 及其選項就能執行下列所有步驟 (步驟 2 到步驟 7)。如需有關此指令的更多資訊，請參閱 `create-http-lb(1)`。

2 針對要受負載平衡器管理的叢集或獨立伺服器實例新增參照。

若要執行此動作，請使用 `asadmin create-http-lb-ref` 指令。如需有關此指令的更多資訊，請參閱 `create-http-lb-ref(1)`。

如果您建立了具有目標的負載平衡器配置，並且該目標是負載平衡器參照的唯一叢集或獨立伺服器實例，請跳過此步驟。

3 啟用負載平衡器所參照的叢集或獨立伺服器實例。

若要執行此動作，請使用 `asadmin enable-http-lb-server` 指令。如需有關此指令的更多資訊，請參閱 `enable-http-lb-server(1)`。

4 啓用用於負載平衡的應用程式。

若要執行此動作，請使用 `asadmin enable-http-lb-application` 指令。如需有關此指令的更多資訊，請參閱 `enable-http-lb-application(1)`。

這些應用程式必須已在負載平衡器所參照的叢集或獨立實例上部署並可以使用。啓用應用程式以用於負載平衡，與啓用應用程式以加以使用，是兩個獨立的步驟。

5 建立運作狀態檢查程式。

若要執行此動作，請使用 `asadmin create-http-health-checker` 指令。如需有關此指令的更多資訊，請參閱 `create-http-health-checker(1)`。

運作狀態檢查程式監視工作異常的伺服器實例，以便在這些伺服器實例重新正常工作時，負載平衡器可以向它們傳送新的請求。

備註 – 如果使用 Sun Java System Web Server (6.1 或 7.0)，則您不必執行步驟 6 及 7，則可用一個步驟就產生負載平衡器配置檔案，並透過網路將資料傳送到 Web 伺服器。

若要使用 `asadmin` 工具執行此動作，請將 `create-http-lb` 指令的 `--autoapplyenabled` 選項設為 `true`。如需有關此指令的更多資訊，請參閱 `create-http-lb(1)`。

6 產生負載平衡器配置檔案。

若要執行此動作，請使用 `asadmin export-http-lb-config` 指令。如需有關此指令的更多資訊，請參閱 `export-http-lb-config(1)`。此指令可產生要與 Sun Java System Application Server 隨附之負載平衡器外掛程式配合使用的配置檔案。

7 將負載平衡器配置檔案複製到 Web 伺服器的 `config` 目錄，該目錄中儲存了負載平衡器外掛程式配置檔案。

備註 – 若要用一個步驟自動產生負載平衡器配置檔案，並透過網路將資料傳送給 Web 伺服器，您需要配置 Web 伺服器的 SSL 設定，並匯入 DAS 憑證。如需有關配置 Sun Java System Web Server 的更多資訊，請參閱第 98 頁的「[配置 Sun Java System Web Server](#)」。

HTTP 負載平衡器部署

您可以使用不同的方式配置負載平衡器（視目標和環境而定），如以下小節中所述：

- 第 124 頁的「[使用叢集伺服器實例](#)」
- 第 124 頁的「[使用多個獨立實例](#)」

使用叢集伺服器實例

部署負載平衡器的最常用方法是使用伺服器實例的一個或多個叢集。依預設，叢集中的所有實例都具有相同的配置，而且這些實例中都部署了相同的應用程式。負載平衡器可在伺服器實例之間分散工作負荷量，並可發出從異常實例向正常實例執行容錯移轉的請求。如果您已配置了 HTTP 階段作業持續性，則在對請求執行容錯移轉時，階段作業資訊會持續。

如果有多個叢集，則可在各個叢集間平衡請求的負荷量，但容錯移轉只能在單一叢集的實例之間進行。在負載平衡器中使用多個叢集可簡化應用程式的輪替式升級。如需更多資訊，請參閱第 137 頁的「在維持可用性的情況下升級應用程式」。

備註 – 您無法平衡叢集和獨立實例之間的請求負載量。

使用多個獨立實例

也可將負載平衡器配置為使用多個獨立的實例，並在這些實例之間對請求進行負載平衡和容錯移轉。但是，在此配置中，您必須手動確定獨立實例具有同質環境，並且這些實例已部署了相同的應用程式。由於叢集會自動維護同質環境，因此大多數情況下，使用叢集會更加簡便有效。

配置負載平衡器

負載平衡器配置是在 `domain.xml` 檔中維護。負載平衡器的配置方式非常靈活：

- 儘管一個網域可以關聯多個負載平衡器，但一個負載平衡器只為一個網域服務。
- 儘管每個負載平衡器只有一個負載平衡器配置，但每個負載平衡器配置可以關聯多個負載平衡器。

這些小節更詳細地說明如何建立、修改和使用負載平衡器配置：

- 第 125 頁的「在 DAS 上配置 HTTP 負載平衡器」
- 第 126 頁的「建立 HTTP 負載平衡器參照」
- 第 126 頁的「啟用用於負載平衡的伺服器實例」
- 第 126 頁的「啟用用於負載平衡的應用程式」
- 第 127 頁的「建立 HTTP 運作狀態檢查程式」
- 第 129 頁的「匯出負載平衡器配置檔案」
- 第 130 頁的「變更負載平衡器配置」
- 第 130 頁的「啟用動態重新配置」
- 第 130 頁的「停用 (靜止) 伺服器實例或叢集」
- 第 131 頁的「停用 (靜止) 應用程式」
- 第 132 頁的「配置 HTTP 和 HTTPS 容錯移轉」
- 第 133 頁的「透過負載平衡器使用重新導向」
- 第 136 頁的「配置等羣 URL」

在 DAS 上配置 HTTP 負載平衡器

在 Application Server 9.1 上，您可以使用管理主控台或 `asadmin` 指令 `create-http-lb` 在 DAS 上建立負載平衡器配置。下列步驟說明如何執行這些動作。如需有關 `asadmin` 指令 `create-http-lb`、`delete-http-lb` 及 `list-http-lbs` 的更多資訊，請參閱「Sun Java System Application Server 9.1 Reference Manual」。

在管理主控台中，將左框架向下捲動，按一下 [HTTP 負載平衡器] 節點，然後在右側的 [HTTP 負載平衡器] 頁面上按一下 [新建]。在 [新建 HTTP 負載平衡器] 頁面中，提供下列主控負載平衡器之機器的詳細資訊。

欄位	說明
名稱	負載平衡器配置的名稱。
已啟用	按一下 [已啟用] 核取方塊，可將負載平衡器配置變更自動推送至 Web 伺服器配置目錄中的實體負載平衡器。
主機	安裝了 Web 伺服器實例的伺服器。
管理連接埠	Web 伺服器實例所使用的管理連接埠號。
代理主機	安裝了代理伺服器實例的伺服器。
代理連接埠	代理伺服器所使用的連接埠號。

您也可以使用 `asadmin` 指令 `create-http-lb-config` 來建立負載平衡器配置。[表 5-1](#) 會說明這些參數。請參閱「Sun Java System Application Server 9.1 Reference Manual」來取得有關指令 `create-http-lb-config`、`delete-http-lb-config` 及 `list-http-lb-configs` 的更多資訊。

表 5-1 負載平衡器配置參數

參數	說明
<code>response timeout</code>	伺服器實例需傳回回應的時限 (以秒為單位)。如果在該時間段內未收到任何回應，則認為伺服器處於異常狀態。預設值為 60。
<code>HTTPS routing</code>	對負載平衡器的 HTTPS 請求是否產生對伺服器實例的 HTTPS 或 HTTP 請求。如需更多資訊，請參閱 第 132 頁的「配置 HTTPS 路由」 。
<code>reload interval</code>	檢查負載平衡器配置檔案 <code>loadbalancer.xml</code> 是否有變更的時間間隔。當檢查偵測到變更時，系統將重新載入配置檔案。值為 0 時會停用重新載入。如需更多資訊，請參閱 第 130 頁的「啟用動態重新配置」 。
<code>monitor</code>	是否對負載平衡器啟用監視功能。

表 5-1 負載平衡器配置參數 (續)

參數	說明
routecookie	負載平衡器外掛程式用於記錄路由資訊的 Cookie 名稱。HTTP 用戶端必須支援 Cookie。如果您的瀏覽器設定為在儲存 Cookie 前詢問，則 Cookie 的名稱為 JROUTE。
target	負載平衡器配置的目標。如果指定目標，則相當於增加對該目標的參照。目標可為叢集或獨立實例。

建立 HTTP 負載平衡器參照

當您在負載平衡器中建立對獨立伺服器或叢集的參照時，會將該伺服器或叢集增加至負載平衡器所控制的目標伺服器和叢集的清單中。您仍需要先啓用所參照的伺服器或叢集，才會負載平衡對其所做的請求。如果建立了帶有目標的負載平衡器配置，則系統已增加了作為參照的該目標。

若要使用管理主控台建立參照，請在左框架中，按一下 [HTTP 負載平衡器] 節點，然後於節點下按一下所需的負載平衡器。開啓 [目標] 標籤，按一下 [管理目標]，然後在 [管理目標] 頁面上選取所要的目標。此外，也可以使用 `create-http-lb-ref` 建立參照。您必須提供負載平衡器配置名稱和目標伺服器實例或叢集。

若要刪除參照，請使用 `delete-http-lb-ref`。必須先使用 `disable-http-lb-server` 停用已參照的伺服器或叢集，才能刪除此參照。

如需有關這些指令的更多資訊，請參閱「Sun Java System Application Server 9.1 Reference Manual」。

啓用用於負載平衡的伺服器實例

建立對伺服器實例或叢集的參照後，請使用 `enable-http-lb-server` 啓用伺服器實例或叢集。如果在建立負載平衡器配置時使用某伺服器實例或叢集作為目標，則必須啓用該伺服器實例或叢集。若要使用管理主控台執行此動作，請在左框架中，按一下 [HTTP 負載平衡器] 節點，再於節點下按一下所要的負載平衡器。現在請開啓 [目標] 標籤，然後在 [目標] 表格中，按一下要啓用的實例旁的核取方塊，再按一下 [啓用]。

如需有關此指令的更多資訊，請參閱 `enable-http-lb-server(1)`。

啓用用於負載平衡的應用程式

負載平衡器管理的所有伺服器均必須具有同質配置，包括部署到這些伺服器的應用程式集必須相同。部署某個應用程式並啓用 (在部署期間或之後) 此應用程式以進行存取後，您必須啓用該應用程式以進行負載平衡。如果沒有為負載平衡啓用應用程式，則將無法對該應用程式的請求執行負載平衡和容錯移轉，即使已對該應用程式部署到的伺服器的請求執行了負載平衡和容錯移轉。

啓用應用程式時，請指定應用程式名稱和目標。如果負載平衡器管理了多個目標 (例如，兩個叢集)，請在所有目標上啓用該應用程式。

若要使用管理主控台啓用應用程式，請在左框架中，按一下 [HTTP 負載平衡器] 節點，然後於節點下按一下所需的負載平衡器。如上所述開啓 [目標] 標籤，然後按一下所需的叢集。現在請開啓 [應用程式] 標籤，選取所需的應用程式，再從 [其他動作] 下拉式清單中選取 [負載平衡器啓用]。如果要從指令行執行此動作，可以使用指令 `asadmin enable-http-lb-application`。如需更多資訊，請參閱 `enable-http-lb-application(1)`。

如果部署了新的應用程式，您也必須啓用該應用程式以進行負載平衡並再次匯出負載平衡器配置。

建立 HTTP 運作狀態檢查程式

負載平衡器的運作狀態檢查程式會定期檢查所有標記為異常的已配置 Application Server 實例。運作狀態檢查程式不是必需的，但如果沒有運作狀態檢查程式，或者停用了運作狀態檢查程式，異常實例的定期運行狀況檢查就不會執行。負載平衡器無法判斷運作不正常的實例在何時開始正常運作。

負載平衡器的運作狀態檢查機制使用 HTTP 與 Application Server 實例進行通訊。運作狀態檢查程式將 HTTP 請求傳送給指定的 URL 並等待回應。HTTP 回應標頭中的狀態碼在 100 到 500 之間時表示實例處於正常狀態。

備註 – 如果您的部署方案把負載平衡器置於叢集前端，而叢集中的實例使用安全連接埠並啓用用戶端憑證驗證功能，則運作狀態檢查程式將無法檢查這些實例的運作狀態。因此，那些實例一律會標記成異常，且不會將請求傳送給它們。

建立運作狀態檢查程式

若要指定運作狀態檢查程式特性，可使用管理主控台或 `asadmin create-http-health-checker` 指令。若要在管理主控台中執行此動作，請瀏覽到 [HTTP 負載平衡器] 節點，將其展開並選取負載平衡器。然後開啓 [目標] 標籤，並在 [目標] 表格中，為所需的目標按一下 [編輯運作狀態檢查程式] 連結。指定下列參數。

表 5-2 運作狀態檢查程式參數

參數	說明	預設
Load Balancer	按一下 [已啓用] 核取方塊，使選取的伺服器可進行負載平衡。	False/已停用
Disable Timeout	此伺服器停用後到達靜止狀態的分鐘數。	30 分鐘

表 5-2 運作狀態檢查程式參數 (續)

參數	說明	預設
url	指定負載平衡器檢查的偵聽程式之 URL 以確定其運作狀態。	“/”
interval	指定實例的運作狀態檢查發生的間隔 (以秒為單位)。指定為 0 將停用運作狀態檢查程式。	30 秒
timeout	指定逾時間隔 (以秒為單位)，必須在該時間間隔內獲得回應才能認為偵聽程式運作正常。	10 秒

如果 Application Server 實例標記為異常，運作狀態檢查程式將輪詢異常實例以確定實例的狀態是否已變為正常。運作狀態檢查程式使用指定的 URL 來檢查所有異常的 Application Server 實例，以確定這些異常的 Application Server 實例是否已返回到正常狀態。

如果運作狀態檢查程式發現某個異常實例已變為正常，該實例將被增加到正常實例的清單中。

如需有關此指令的更多資訊，請參閱 create-http-health-checker(1) 和 delete-http-health-checker(1)。

運作正常實例的其他運作狀態檢查特性

create-http-health-checker 建立的運作狀態檢查程式僅檢查異常實例。若要定期檢查運作正常的實例，請在匯出的 loadbalancer.xml 檔案中設定某些其他特性。

備註 – 您必須將 loadbalancer.xml 匯出之後，才能以手動編輯方式設定這些特性。沒有等效的 asadmin 指令可以使用。

若要檢查運作正常實例，請設定以下特性。

表 5-3 運作狀態檢查程式手動特性

特性	定義
active-healthcheck-enabled	True/False 標幟，用於表示是否要 Ping 正常伺服器實例以確定這些實例是否正常。若要 Ping 伺服器實例，請將標幟設定為 True。
number-healthcheck-retries	指定在將未回應的伺服器實例標記為異常之前，負載平衡器的運作狀態檢查程式對該伺服器實例執行 Ping 指令的次數。有效範圍在 1 到 1000 之間。設定的預設值為 3。

透過編輯 loadbalancer.xml 檔案來設定特性。例如：

```
<property name="active-healthcheck-enabled" value="true"/>
<property name="number-healthcheck-retries" value="3"/>
```


若您增加這些特性，然後進行編輯並再次匯出 `loadbalancer.xml` 檔案，新匯出的配置不會包含這些特性。您必須對新匯出的配置再次增加這些特性。

匯出負載平衡器配置檔案

Sun Java System Application Server 隨附的負載平衡器外掛程式使用名為 `loadbalancer.xml` 的配置檔案。配置負載平衡器後，可將配置詳細資訊從 `domain.xml` 匯出到 `loadbalancer.xml` 檔案。您可以使用管理主控台或 `asadmin` 公用程式來執行此動作。

▼ 使用管理主控台匯出負載平衡器配置

- 1 瀏覽到 [HTTP 負載平衡器] 節點並將其展開。
- 2 按一下所需的負載平衡器。
所有負載平衡器配置詳細資訊會顯示在 [一般]、[設定] 及 [目標] 標籤中。
- 3 開啓 [匯出] 標籤並按一下 [立即匯出]。
- 4 將已匯出的負載平衡器配置檔案複製到 Web 伺服器的配置目錄。

▼ 使用 `asadmin` 工具匯出負載平衡器配置

- 1 使用 `asadmin` 指令 `export-http-lb-config` 匯出 `loadbalancer.xml` 檔案。如需有關此指令的更多資訊，請參閱 `export-http-lb-config(1)`。

匯出特定負載平衡器配置的 `loadbalancer.xml` 檔案。您可以指定路徑和其他檔案名稱。如果未指定檔名，則該檔案會命名為 `loadbalancer.xml`。

`load-balancer-config-name`。若您未指定路徑，則會在 `domain-dir/generated` 目錄中建立檔案。

若要在 Windows 上指定路徑，請將路徑加上引號。例如，
`"C:\Sun\AppServer\loadbalancer.xml"`。

- 2 將已匯出的負載平衡器配置檔案複製到 Web 伺服器的配置目錄。

例如，對於 Sun Java System Web Server，該位置通常為 `web-server-root/config`。

Web 伺服器配置目錄中的負載平衡器配置檔案必須命名為 `loadbalancer.xml`。如果您的檔案使用其他名稱，例如 `loadbalancer.xml.load-balancer-config-name`，則必須重新命名該檔案。

變更負載平衡器配置

如果您透過建立或刪除對伺服器的參照、部署新的應用程式、啓用或停用伺服器或應用程式等方法來變更負載平衡器配置，則請重新匯出負載平衡器配置檔案並將其複製到 Web 伺服器的 `config` 目錄。如需更多資訊，請參閱第 129 頁的「匯出負載平衡器配置檔案」。

負載平衡器外掛程式將根據負載平衡器配置中指定的重新載入間隔，定期檢查有無已更新的配置。在指定的時間值後，如果負載平衡器發現新的配置檔案，則它將開始使用新配置。

啓用動態重新配置

透過動態重新配置，負載平衡器外掛程式會定期檢查有無已更新的配置。

若要啓用動態重新配置，請執行以下步驟：

- 建立負載平衡器配置時，將 `--reloadinterval` 選項與 `asadmin create-http-lb` 配合使用。如需有關此指令的更多資訊，請參閱 `create-http-lb(1)`。
此選項用於設定檢查負載平衡器配置檔案 `loadbalancer.xml` 有無變更的時間間隔。將值設定為 0 可停用動態重新配置。依預設，啓用動態重新配置，重新載入時間間隔為 60 秒。
- 如果您之前已將其停用，或要變更重新載入時間間隔，請使用 `asadmin set` 指令。
變更重新載入時間間隔之後，請重新匯出負載平衡器配置檔案，並將其複製到 Web 伺服器的 `config` 目錄，然後重新啓動 Web 伺服器。

備註 – 如果負載平衡器在嘗試重新配置時遇到硬碟讀取錯誤，它將使用記憶體中目前的配置。負載平衡器還確定了在覆寫現有配置之前，已修改的配置資料與 DTD 相容。

如果遇到磁碟讀取錯誤，相關的警告訊息會記錄至 Web 伺服器的錯誤記錄檔中。

Sun Java System Web Server 的錯誤記錄位於以下位置：
`web-server-install-dir/web-server-instance/logs/`。

停用 (靜止) 伺服器實例或叢集

在因任何原因而停止 Application Server 之前，實例應該完成正在處理的請求。正常停用伺服器實例或叢集的程序被稱為靜止。

負載平衡器使用以下策略來靜止 Application Server 實例：

- 如果已停用某個實例(獨立實例或叢集的一部分)，並且逾時尚未到期，居留式請求將繼續發送到該實例。但是，新請求將不會傳送到已停用的實例。
- 逾時到期後，該實例將停用。從負載平衡器到該實例的所有開啓的連線將關閉。負載平衡器不會將任何請求傳送到該實例，即使居留在該實例的所有階段作業都還有效。相反，負載平衡器會將居留式請求容錯移轉到另一個正常實例上。

▼ 停用某個伺服器實例或叢集

- 1 執行 `asadmin disable-http-lb-server`，設定逾時(以分鐘為單位)。如需有關此指令的更多資訊，請參閱 `disable-http-lb-server(1)`。
- 2 使用 `asadmin export-http-lb-config` 匯出負載平衡器配置檔案。如需有關此指令的更多資訊，請參閱 `export-http-lb-config(1)`。
- 3 將匯出的配置複製到 Web 伺服器的 `config` 目錄。
- 4 停止該伺服器實例或叢集。

停用(靜止)應用程式

在取消部署 Web 應用程式之前，應讓應用程式完成正在處理的請求。正常停用應用程式的程序稱為靜止。靜止應用程式時，您可指定逾時時間。依據逾時時間，負載平衡器會使用以下策略靜止應用程式：

- 如果逾時未過期，負載平衡器不會向應用程式轉寄新的請求，但會將其傳回 Web 伺服器。然而，負載平衡器會繼續轉寄居留式請求，直至逾時過期。
- 逾時過期後，負載平衡器不會接受應用程式的任何請求(包括居留式請求)。

當您從負載平衡器參照的每個伺服器實例或叢集中停用應用程式時，已停用的應用程式的使用者將遭受服務損失，直到再次啓用該應用程式。如果您從一個伺服器實例或叢集中停用應用程式而使該應用程式在其他伺服器實例或叢集中保持啓用狀態，則使用者仍可存取該應用程式。如需更多資訊，請參閱第 137 頁的「在維持可用性的情況下升級應用程式」。

▼ 停用應用程式

- 1 使用 `asadmin disable-http-lb-application`，並指定以下內容：
 - 逾時(以分鐘為單位)。
 - 要停用的應用程式之名稱。
 - 要在其上停用應用程式的目標叢集或實例。

如需有關此指令的更多資訊，請參閱 `disable-http-lb-application(1)`。

- 2 使用 `asadmin export-http-lb-config` 匯出負載平衡器配置檔案。如需有關此指令的更多資訊，請參閱 `export-http-lb-config(1)`。
- 3 將匯出的配置複製到 Web 伺服器的 `config` 目錄。

配置 HTTP 和 HTTPS 容錯移轉

如果 HTTP/HTTPS 階段作業所連線的原始 Application Server 實例變為不可用，負載平衡器外掛程式會將這些階段作業容錯移轉到其他 Application Server 實例上。本小節介紹如何配置負載平衡器外掛程式，以啟用 HTTP/HTTPS 路由和階段作業容錯移轉。

HTTPS 路由

負載平衡器外掛程式將所有內送的 HTTP 或 HTTPS 請求路由至 Application Server 實例。然而，如果啟用了 HTTPS 路由，則負載平衡器外掛程式會將 HTTPS 請求僅轉寄給使用 HTTPS 連接埠的 Application Server。HTTPS 路由是針對新請求和居留式請求而執行的。

如果收到了 HTTPS 請求且沒有正在進行的階段作業，負載平衡器外掛程式將選取使用已配置的 HTTPS 連接埠的可用 Application Server 實例，並將請求轉寄到該實例。

在正在進行的 HTTP 階段作業中，如果收到對同一個階段作業的新 HTTPS 請求，則使用在 HTTP 階段作業期間儲存的階段作業和居留式資訊來路由 HTTPS 請求。新的 HTTPS 請求將路由到此 HTTPS 連接埠上處理上一個 HTTP 請求的同一伺服器。

配置 HTTPS 路由

`create-http-lb-config` 指令的 `httpsrouting` 選項，用於控制是否為參與負載平衡的所有應用程式伺服器開啓 HTTPS 路由功能。如果此選項設定為 `false`，所有 HTTP 和 HTTPS 請求都將作為 HTTP 請求進行轉寄。如果設定為 `true`，HTTPS 將做為 HTTPS 請求進行轉寄。建立新的負載平衡器配置時，請設定 HTTPS 路由，或者以後使用 `asadmin set` 指令加以變更。

備註 –

- 若要使 HTTPS 路由工作，必須配置一個或多個 HTTPS 偵聽程式。
 - 如果 `https-routing` 設定為 `true`，而新請求或居留式請求傳入的叢集中沒有正常的 HTTPS 偵聽程式，則該請求將產生一個錯誤。
-

已知問題

負載平衡器對 HTTP/HTTPS 請求處理具有以下限制。

- 如果某個階段作業使用 HTTP 和 HTTPS 請求的組合，則第一個請求必須是 HTTP 請求。如果第一個請求為 HTTPS 請求，則其後不能接 HTTP 請求。這是因為與 HTTPS 階段作業關聯的 Cookie 不是由瀏覽器傳回的。瀏覽器將兩個不同的協定解釋為兩個不同的伺服器，並啟動新的階段作業。僅當將 `httpsrouting` 設定為 `true` 時，此限制才有效。
- 如果某個階段作業具有 HTTP 和 HTTPS 請求的組合，則必須將 Application Server 實例配置為同時具有 HTTP 和 HTTPS 偵聽程式。僅當 `httpsrouting` 設定為 `true` 時，此限制才有效。
- 如果某個階段作業具有 HTTP 和 HTTPS 請求的組合，則必須將 Application Server 實例配置為具有使用標準連接埠號 (即，對於 HTTP 為 80，對於 HTTPS 為 443) 的 HTTP 和 HTTPS 偵聽程式。`httpsrouting` 無論設定為哪個值，皆需遵守此限制。

透過負載平衡器使用重新導向

使用重新導向將請求從一個 URL 重新導向至另一個 URL。例如，使用重新導向將使用者傳送至不同的網站 (例如從舊版應用程式重新導向至較新版本)，或是從 HTTP 重新導向至 HTTPS，或是從 HTTPS 重新導向至 HTTP。您可以在應用程式中以多種方式啟用重新導向 (例如，以 Servlet 為基礎的重新導向，`web.xml` 重新導向)。然而，如果要從負載平衡器傳送重新導向 URL，可能需要對 Application Server 或負載平衡器進行其他配置。請注意，重新導向和使用 HTTPS 路由轉寄的請求不同。使用重新導向時，請將 `httpsrouting` 設定為 `false`。如果將 HTTPS 請求配置成需轉寄至 HTTP，請使用第 132 頁的「HTTPS 路由」。

下列特性會影響重新導向：HTTP 服務或 HTTP 偵聽程式的 `authPassthroughEnabled` 和 `proxyHandler` 特性，以及 `loadbalancer.xml` 檔案中的 `rewrite-location` 特性。

authPassthroughEnabled 特性

當 Application Server `authPassthroughEnabled` 特性設定為 `true` 時，則會以自訂請求標頭將有關原始用戶端請求的資訊 (例如用戶端 IP 位址、SSL 金鑰大小，以及認證的用戶端憑證鏈) 傳送至 HTTP 偵聽程式。如果您安裝了硬體加速器，`authPassthroughEnabled` 特性會讓您利用硬體加速器加快 SSL 認證的進行。在負載平衡器上配置硬體加速器，比在每個叢集的 Application Server 實例上進行配置更為容易。



注意 – 只有在 Application Server 受防火牆保護後，才能將 `authPassthroughEnabled` 設定為 `true`。

使用 `asadmin set` 指令來設定 HTTP 服務或個別 HTTP 偵聽程式上的 `authPassthroughEnabled` 特性。個別 HTTP 偵聽程式的設定優先於 HTTP 服務的設定。

如果要在所有的 HTTP/HTTPS 偵聽程式上設定 `authPassthroughEnabled` 特性，請使用下列指令：

```
asadmin set  
cluster-name-config.http-service.property.authPassthroughEnabled=true
```

如果要在個別的偵聽程式上進行設定，請使用下列指令：

```
asadmin set cluster-name-config.http-service.http-listener.  
listener-name.property.authPassthroughEnabled=true
```

proxyHandler 特性

Application Server 的代理伺服器處理程式負責擷取遭代理伺服器 (在此例中指的是負載平衡器) 攔截、並轉寄至應用程式伺服器之原始用戶端請求的相關資訊，同時負責將此資訊提供給做為用戶端請求之目標的 Web 應用程式 (部署於應用程式伺服器上)。如果代理伺服器的截取遭 SSL 終止，代理伺服器處理程式會擷取並提供有關原始請求的其他資訊，例如原始請求是否為 HTTPS 請求，以及 SSL 用戶端認證是否啟用。只有當 `authPassThroughEnabled` 設定為 `true` 時，才使用 `proxyHandler` 屬性。

代理伺服器處理程式會檢查內送請求中是否有自訂請求標頭，而代理伺服器就是藉由該自訂請求標頭傳送和原始用戶端請求相關的資訊，並使用標準的 `ServletRequest` API 向 Application Server 上的 Web 應用程式提供這項資訊。

代理伺服器處理程式實作可透過 `proxyHandler` 特性，在 HTTP 服務層級上進行全域配置，或是針對個別的 HTTP 偵聽程式進行配置，而該特性的值會指定 `com.sun.appserv.ProxyHandler` 抽象類別之實作的完全合格類別名稱。可進行配置的代理伺服器處理程式實作允許 Application Server 搭配任何代理伺服器，只要代理伺服器處理程式實作知道 HTTP 請求的標頭名稱，並了解其值的格式，而代理伺服器就是以此方式傳送與原始用戶端請求相關的資訊。

Application Server 的代理伺服器處理程式會從請求標頭讀取並剖析 SSL 憑證鏈。此舉讓後端 Application Server 實例得以擷取原始用戶端請求 (遭到終止 SSL 的代理伺服器所攔截；該代理伺服器在此指的是負載平衡器) 的相關資訊。您可以使用預設的代理伺服器處理程式設定，或是使用 HTTP 服務或 HTTP/HTTPS 偵聽程式的 `proxyHandler` 特性自行加以配置。`proxyHandler` 特性會針對該偵聽程式或所有偵聽程式所使用的 `com.sun.appserv.ProxyHandler` 抽象類別之自訂實作，指定完全合格類別名稱。

此抽象類別的實作會檢視自訂請求標頭的指定請求，而代理伺服器就是透過該自訂請求標頭，將原始用戶端請求的相關資訊傳送至 Application Server 實例，並將該資訊傳回至其呼叫者。預設實作會從名為 `Proxy-ip` 的 HTTP 請求標頭讀取用戶端 IP 位址、從名為 `Proxy-keysize` 的 HTTP 請求標頭讀取 SSL 金鑰大小，並從名為 `Proxy-auth-cert` 的 HTTP 請求標頭讀取 SSL 用戶端憑證鏈。`Proxy-auth-cert` 值必須包含 BASE-64 編碼的用戶端憑證鏈且無 BEGIN CERTIFICATE 和 END CERTIFICATE 邊界，並以 `%d%a` 取代 `\n`。

只有當 `authPassThroughEnabled` 設定為 `true` 時，才能使用這項特性。如果您在個別的 HTTP 或 HTTPS 偵聽程式上設定 `proxyHandler` 特性，該特性會覆寫所有偵聽程式的預設設定。

使用 `asadmin set` 指令來設定 HTTP 服務或個別 HTTP 偵聽程式上的 `proxyHandler` 特性。

如果要在所有的 HTTP/HTTPS 偵聽程式上設定 `proxyHandler` 特性，請使用下列指令：

```
asadmin set cluster-name-config.http-service.property.proxyHandler= classname
```

如果要在個別的偵聽程式上進行設定，請使用下列指令：

```
asadmin set cluster-name-config.http-service.http-listener.  
listener-name.property.proxyHandler= classname
```

rewrite-location 特性

如果設定為 `true`，`rewrite-location` 特性就會重寫原始的請求資訊，並納入通訊協定 (HTTP 或 HTTPS)、主機和連接埠資訊。依預設，`rewrite-location` 特性會設定為 `true`，以維持與先前 Application Server 發行版本的向下相容性。

您無法經由 `asadmin create-http-lb-config` 或 `asadmin set` 指令取得 `rewrite-location` 特性。如果要使用這項特性，請在匯出負載平衡器配置後，將該特性手動增加至 `loadbalancer.xml` 檔案。例如，將下列特性增加至所匯出的 `loadbalancer.xml` 檔案：

```
<property name="rewrite-location" value="false"/>
```

設定 `rewrite-location` 特性時，請記住下列要點：

- 如果 `httpsrouting` 為 `false`，而 Application Server 並未啟用 `authPassthroughEnabled`，則請將 `rewrite-location` 特性設定為 `true`。在 `authPassthroughEnabled` 未啟用的情況下，Application Server 將無法得知原始請求的通訊協定 (HTTP 或 HTTPS)。如果將 `rewrite-location` 設定為 `true`，負載平衡器會適當修改重寫位置的通訊協定部分。也就是說，如果用戶端正在傳送 HTTPS 請求，則負載平衡器會將用戶端重新導向至負載平衡器上已啟用 HTTPS 的偵聽程式連接埠。這項程序和 HTTP 請求相同。
- 如果 `httpsrouting` 為 `false`，並且啟用了 Application Server 上的 `authPassthroughEnabled`，則可將 `rewrite-location` 設定為 `true` 或 `false`，因為 Application Server 能夠得知用戶端請求是 HTTP 還是 HTTPS。啟用 `authPassthroughEnabled` 時，Application Server 會適當修改重寫位置的通訊協定部分。如果 `rewrite-location` 設定為 `false`，則負載平衡器不會重寫重新導向之 URL 的位置。這項特性如果設定為 `true`，則會重寫重新導向之 URL 的位置。但是重寫是沒有必要的，因為 Application Server 已經得知來自用戶端的 HTTPS 連線。另外，如果應用程式需要將 HTTP 重新導向至 HTTPS，或是將 HTTPS 重新導向至 HTTP，則必須將 `rewrite-location` 參數設定為 `false`。

配置等冪 URL

等冪請求是一種在重試時不會在應用程式中造成任何變更或不一致的請求。在 HTTP 中，某些方法 (例如 GET) 是等冪的，而其他方法 (例如 POST) 則不是。重試等冪 URL 不會導致伺服器上或資料庫中的值發生變更。唯一的差異在於使用者收到的回應會有所不同。

等冪請求的範例包括搜尋引擎查詢和資料庫查詢。基礎原則是重試不會導致資料的更新或修改。

若要增強已部署應用程式的可用性，請在負載平衡器所服務的所有 **Application Server** 實例上，將環境配置成等冪 HTTP 請求失敗時必須重試。此選項用於唯讀請求 (例如，重試搜尋請求)。

請在 `sun-web.xml` 檔案中配置等冪 URL。當您匯出負載平衡器配置時，等冪 URL 資訊將自動增加到 `loadbalancer.xml` 檔案中。

如需有關配置等冪 URL 的更多資訊，請參閱「Sun Java System Application Server 9.1 Developer's Guide」中的「Configuring Idempotent URL Requests」。

配置多個 Web 伺服器實例

Sun Java System Application Server 安裝程式不允許在單一機器上安裝多個負載平衡器外掛程式。若要在單一叢集或多個叢集中的單一機器上安裝多個帶有負載平衡器外掛程式的 Web 伺服器，需要手動執行一些步驟來配置負載平衡器外掛程式。

▼ 配置多個 Web 伺服器實例

- 1 配置新的 Web 伺服器實例以使用負載平衡器外掛程式。
如需詳細指示，請參閱第 98 頁的「配置 Sun Java System Web Server」。
- 2 從現有 Web 伺服器實例的 `config` 目錄中，將 DTD 檔案 `sun-loadbalancer_1_1.dtd` 複製到新實例的 `config` 目錄中。
- 3 設定負載平衡器配置檔案。或者：
 - 複製現有負載平衡器配置。
使用現有負載平衡器配置，從現有 Web 伺服器實例的 `config` 目錄中，將 `loadbalancer.xml` 檔案複製到新實例的 `config` 目錄中。
 - 建立新的負載平衡器配置：
 - a. 使用 `asadmin create-http-lb-config` 建立新的負載平衡器配置。

- b. 使用 `asadmin export http-lb-config` 將新配置匯出至 `loadbalancer.xml` 檔案中。
- c. 將該 `loadbalancer.xml` 檔案複製至新 Web 伺服器的 `config` 目錄中。
如需有關建立負載平衡器配置並將其匯出至 `loadbalancer.xml` 檔案的資訊，請參閱第 125 頁的「在 DAS 上配置 HTTP 負載平衡器」。

在維持可用性的情況下升級應用程式

將應用程式升級至新版本但使用者仍可使用應用程式的作法，即稱為**輪替式升級**。在升級過程中，小心管理這兩個版本的應用程式，不但可以確保應用程式的目前使用者在無中斷的情況下完成作業，同時可讓新使用者不需設定即可取得新的應用程式版本。使用輪替式升級時，使用者不會察覺正在進行升級。

應用程式相容性

依據兩個應用程式版本之間的變更大小，輪替式升級會造成不同程度的困難。

如果變更很小 (例如對靜態文字和影像的變更)，則此應用程式的兩個版本可以**相容**，並可在相同叢集中同時執行。

相容的應用程式必須：

- 使用相同的階段作業資訊
- 使用相容的資料庫模式
- 具有一般相容應用程式層級的業務邏輯
- 使用相同的實體資料來源

您可以在單一叢集或多個叢集中執行相容應用程式的輪替式升級。如需更多資訊，請參閱第 137 頁的「在單一叢集中升級」。

如果應用程式的兩個版本不滿足上述所有條件，則應用程式會被視為**不相容**。在某個叢集中執行不相容的應用程式版本會毀壞應用程式資料，並導致階段作業容錯移轉無法正常執行。該問題取決於不相容性的類型和程度。好的做法是透過建立要向其部署新版本應用程式的「隱式叢集」，來升級不相容的應用程式，然後緩慢靜止舊叢集和應用程式。如需更多資訊，請參閱第 141 頁的「升級不相容的應用程式」。

應用程式開發者和管理員是確定應用程式版本是否相容的最佳人選。如有疑問，請假設版本不相容，因為這是最安全的方法。

在單一叢集中升級

如果未與任何其他叢集共用叢集配置，則可以對部署至單一叢集的應用程式執行輪替式升級。

▼ 在單一叢集中升級應用程式

1 儲存舊版本的應用程式或備份網域。

若要備份網域，請使用 `asadmin backup-domain` 指令。如需有關此指令的更多資訊，請參閱 `backup-domain(1)`。

2 關閉叢集的動態重新配置 (如果已啟用)。

從管理主控台執行此動作：

a. 展開配置節點。

b. 按一下叢集配置的名稱。

c. 在 [配置系統特性] 頁面中，取消核取 [已啟用動態重新配置] 方塊。

d. 按一下 [儲存]

或者，使用以下指令：

```
asadmin set --user user --passwordfile password-file  
cluster-name-config.dynamic-reconfiguration-enabled=false
```

3 將已升級的應用程式重新部署到目標 domain 中。

如果使用管理主控台進行重新部署，網域將自動成為目標。如果使用 `asadmin`，請指定目標 domain。由於已停用動態重新配置，因此舊應用程式將繼續在叢集上執行。

4 使用 `asadmin enable-http-lb-application` 為實例啟用已重新部署的應用程式。如需有關此指令的更多資訊，請參閱 `enable-http-lb-application(1)`。

5 從負載平衡器靜止叢集中的某個伺服器實例。

請依照下列步驟執行：

a. 使用 `asadmin disable-http-lb-server` 停用伺服器實例。如需有關此指令的更多資訊，請參閱 `disable-http-lb-server(1)`。

b. 使用 `asadmin export-http-lb-config` 匯出負載平衡器配置檔案。如需有關此指令的更多資訊，請參閱 `export-http-lb-config(1)`。

c. 將已匯出的配置檔案複製到 Web 伺服器實例的配置目錄。

例如，對於 Sun Java System Web Server，位置為

`web-server-install-dir/https-host-name /config/loadbalancer.xml`。若要確保負載平衡器載入新的配置檔案，請務必確定已透過在負載平衡器配置中設定 `reloadinterval` 啟用了動態重新配置。

- d. 請等待，直至逾時值到期。
監視負載平衡器的記錄檔以確定實例已離線。如果使用者看到重試 URL，請略過靜止期間，並立即重新啟動伺服器。
- 6 在叢集中的其他實例仍處於執行狀態的情況下，重新啟動已停用的伺服器實例。
重新啟動操作將使伺服器與網域同步，並更新應用程式。
- 7 測試重新啟動的伺服器上的應用程式，以確定應用程式執行正常。
- 8 在負載平衡器中重新啟用伺服器實例。
請依照下列步驟執行：
 - a. 使用 `asadmin enable-http-lb-server` 啟用伺服器實例。如需有關此指令的更多資訊，請參閱 `enable-http-lb-server(1)`。
 - b. 使用 `asadmin export-http-lb-config` 匯出負載平衡器配置檔案。如需有關此指令的更多資訊，請參閱 `export-http-lb-config(1)`。
 - c. 將配置檔案複製到 Web 伺服器的配置目錄。
- 9 對叢集中的每個實例重複步驟 5 至 8。
- 10 當所有伺服器實例具有新的應用程式並且這些伺服器實例正在執行時，您可以再次為叢集啟用動態重新配置。

在多個叢集中升級

▼ 若要在兩個或更多叢集中升級相容的應用程式，請執行以下步驟：

- 1 儲存舊版本的應用程式或備份網域。
若要備份網域，請使用 `asadmin backup-domain` 指令。如需有關此指令的更多資訊，請參閱 `backup-domain(1)`。
- 2 關閉所有叢集的動態重新配置 (如果已啟用)。
從管理主控台執行此動作：
 - a. 展開配置節點。
 - b. 按一下一個叢集配置的名稱。
 - c. 在 [配置系統特性] 頁面中，取消核取 [已啟用動態重新配置] 方塊。

d. 按一下[儲存]

e. 對其他叢集重複此程序

或者，使用以下指令：

```
asadmin set --user user --passwordfile password-file cluster-name
-config.dynamic-reconfiguration-enabled=false
```

3 將已升級的應用程式重新部署到目標 domain 中。

如果使用管理主控台進行重新部署，網域將自動成為目標。如果使用 `asadmin`，請指定目標 domain。由於已停用動態重新配置，因此舊應用程式將繼續在叢集上執行。

4 使用 `asadmin enable-http-lb-application` 為叢集啟用已重新部署的應用程式。如需有關此指令的更多資訊，請參閱 `enable-http-lb-application(1)`。

5 靜止負載平衡器中的某個叢集

a. 使用 `asadmin disable-http-lb-server` 停用叢集。如需有關此指令的更多資訊，請參閱 `disable-http-lb-server(1)`。

b. 使用 `asadmin export-http-lb-config` 匯出負載平衡器配置檔案。如需有關此指令的更多資訊，請參閱 `export-http-lb-config(1)`。

c. 將已匯出的配置檔案複製到 Web 伺服器實例的配置目錄。

例如，對於 Sun Java System Web Server，位置為 `web-server-install-dir/https-host-name/config/loadbalancer.xml`。必須為負載平衡器啟用動態重新配置 (透過在負載平衡器配置中設定 `reloadinterval`)，以便能夠自動載入新的負載平衡器配置檔案。

d. 請等待，直至逾時值到期。

監視負載平衡器的記錄檔以確定實例已離線。如果使用者看到重試 URL，請略過靜止期間，並立即重新啟動伺服器。

6 在其他叢集仍處於執行狀態的情況下，重新啟動已停用的叢集。

重新啟動操作將使叢集與網域同步，並更新應用程式。

7 測試重新啟動的叢集上的應用程式，以確定應用程式執行正常。

8 在負載平衡器中啟用叢集：

a. 使用 `asadmin enable-http-lb-server` 啟用叢集。如需有關此指令的更多資訊，請參閱 `enable-http-lb-server(1)`。

- b. 使用 `asadmin export-http-lb-config` 匯出負載平衡器配置檔案。如需有關此指令的更多資訊，請參閱 `export-http-lb-config(1)`。
 - c. 將配置檔案複製到 Web 伺服器的配置目錄。
- 9 對其他叢集重複步驟 5 至 8。
 - 10 當所有伺服器實例都具有新的應用程式並且這些伺服器實例正在執行時，您可以為所有叢集再次啓用動態重新配置。

升級不相容的應用程式

如果新版應用程式與舊版不相容，請使用下列程序。有關應用程式相容要素的相關資訊，請參閱第 137 頁的「[應用程式相容性](#)」。此外，您必須在兩個或更多叢集中升級不相容的應用程式。如果您僅有一個叢集，請為升級建立「隱式叢集」，如下所述。

升級不相容的應用程式時，請：

- 為應用程式的新版本指定與其舊版本不同的名稱。以下步驟假設已重新命名應用程式。
- 如果資料模式不相容，請在規劃資料遷移之後使用不同的實體資料來源。
- 在與其中部署了舊版本的叢集不同的叢集中部署新版本。
- 讓執行舊應用程式的叢集離線之前，為其設定適當長的逾時，因為應用程式的請求不會容錯移轉至新叢集。這些使用者階段作業將失敗。

▼ 透過建立第二個叢集來升級不相容的應用程式

- 1 儲存舊版本的應用程式或備份網域。
若要備份網域，請使用 `asadmin backup-domain` 指令。如需有關此指令的更多資訊，請參閱 `backup-domain(1)`。
- 2 在相同或不同的一組機器上建立「隱式叢集」，將其做為現有叢集。如果您已經有第二個叢集，請跳過這個步驟。
 - a. 使用管理主控台建立新叢集並參照現有叢集的已命名配置。
在每個機器上為新實例自訂連接埠，以避免與現有使用中連接埠衝突。
 - b. 對於與叢集相關聯的所有資源，使用 `asadmin create-resource-ref` 將資源參照增加至新建的叢集中。如需有關此指令的更多資訊，請參閱 `create-resource-ref(1)`。
 - c. 使用 `asadmin create-application-ref`，從新建的叢集中，建立對部署至該叢集的所有其他應用程式（目前重新部署的應用程式除外）的參照。如需有關此指令的更多資訊，請參閱 `create-application-ref(1)`。

- d. 使用 `asadmin configure-ha-cluster` 將叢集配置為具有高可用性。如需有關此指令的更多資訊，請參閱 `configure-ha-cluster(1)`。
- e. 使用 `asadmin create-http-lb-ref` 在負載平衡器配置檔案中建立對新建叢集的參照。如需有關此指令的更多資訊，請參閱 `create-http-lb-ref(1)`。
- 3 為應用程式的新版本指定與舊版本不同的名稱。
- 4 透過將新叢集做為目標來部署新應用程式。使用不同的一個或多個環境根目錄。
- 5 使用 `asadmin enable-http-lb-application` 為叢集啟用已部署的新應用程式。如需有關此指令的更多資訊，請參閱 `enable-http-lb-application(1)`。
- 6 在其他叢集仍處於執行狀態的情況下，啟動新的叢集。
啟動操作將導致叢集與網域同步，並使用新應用程式進行更新。
- 7 測試新叢集上的應用程式，以確定其是否正常執行。
- 8 使用 `asadmin disable-http-lb-server` 從負載平衡器停用舊叢集。如需有關此指令的更多資訊，請參閱 `disable-http-lb-server(1)`。
- 9 為延遲階段作業存在的時間設定逾時。
- 10 使用 `asadmin enable-http-lb-server` 從負載平衡器啟用新叢集。如需有關此指令的更多資訊，請參閱 `enable-http-lb-server(1)`。
- 11 使用 `asadmin export-http-lb-config` 匯出負載平衡器配置檔案。如需有關此指令的更多資訊，請參閱 `export-http-lb-config(1)`。
- 12 將已匯出的配置檔案複製到 Web 伺服器實例的配置目錄。
例如，對於 Sun Java System Web Server，位置為 `web-server-install-dir/https-host-name/config/loadbalancer.xml`。必須為負載平衡器啟用動態重新配置 (透過在負載平衡器配置中設定 `reloadinterval`)，以便能夠自動載入新的負載平衡器配置檔案。
- 13 在逾時時間過期後，或舊應用程式的所有使用者結束後，停止舊叢集並刪除舊應用程式。

監視 HTTP 負載平衡器外掛程式

- 第 143 頁的「配置記錄訊息」
- 第 143 頁的「記錄訊息類型」
- 第 144 頁的「啓用負載平衡器記錄」
- 第 146 頁的「瞭解監視訊息」

配置記錄訊息

負載平衡器外掛程式使用 Web 伺服器的記錄機制寫入記錄訊息。Application Server 的預設記錄層級已設定為 Sun Java System Web Server (INFO)、Apache Web Server (WARN) 和 Microsoft IIS (INFO) 的預設記錄層級。Application Server 記錄層級 (FINE、FINER 和 FINEST) 對映到 Web 伺服器上的 DEBUG 層級。

這些記錄訊息將寫入 Web 伺服器記錄檔，並且以可使用程序檔進行剖析或匯入試算表以計算所需公制的原始資料形式進行。

記錄訊息類型

負載平衡器外掛程式會產生以下類型的記錄訊息：

- 第 143 頁的「負載平衡器配置器記錄訊息」
- 第 144 頁的「請求派送和執行階段記錄訊息」
- 第 144 頁的「配置器錯誤訊息」

負載平衡器配置器記錄訊息

使用等冪 URL 和錯誤頁面設定時，將記錄這些訊息。

等冪 URL 式樣配置的輸出包含以下資訊：

- 當記錄層級設定為 FINE 時：
CONFxxxx: IdempotentUrlPattern configured <url-pattern> <no-of-retries> for web-module : <web-module>
- 當記錄層級設定為 SEVERE 時：
CONFxxxx: Duplicate entry of Idempotent URL element <url-pattern> for webModule <web-module> in loadbalancer.xml."
- 當記錄層級設定為 WARN 時：
CONFxxxx: Invalid IdempotentUrlPatternData <url-pattern> for web-module <web-module>
錯誤頁面 URL 配置的輸出包含以下資訊 (記錄層級設定為 WARN)：
CONFxxxx: Invalid error-url for web-module <web-module>

請求派送和執行階段記錄訊息

這些記錄訊息在負載平衡和派送請求時產生。

- 每個方法隆I的標準記錄的輸出均包含以下資訊 (記錄層級設定為 FINE) :
ROUTxxxx: Executing Router method <method_name>
- 每個方法隆I的路由器記錄的輸出均包含以下資訊 (記錄層級設定為 INFO) :
ROUTxxxx: Successfully Selected another ServerInstance for idempotent request
<Request-URL>
- 執行階段記錄的輸出包含以下資訊 (記錄層級設定為 INFO) :
RNTMxxxx: Retrying Idempotent <GET/POST/HEAD> Request <Request-URL>

配置器錯誤訊息

如果存在配置問題 (例如，缺少參照的自訂錯誤頁面)，將顯示這些錯誤。

- 記錄層級設定為 INFO :
ROUTxxxx: Non Idempotent Request <Request-URL> cannot be retried
例如：ROUTxxxx: Non Idempotent Request http://sun.com/addToDB?x=11&abc=2
cannot be retried
- 記錄層級設定為 FINE :
RNTMxxxx: Invalid / Missing Custom error-url / page: <error-url> for
web-module: <web-module>
例如：RNTMxxxx: Invalid / Missing Custom error-url / page: myerror1xyz for
web-module: test

啓用負載平衡器記錄

負載平衡器外掛程式記錄以下資訊：

- 每個請求的請求隆I/停止資訊。
- 請求從異常實例容錯移轉到正常實例時的容錯移轉請求資訊。
- 每個運作狀態檢查週期結束時的異常實例清單。

備註 – 啓用負載平衡器記錄後，如果將 Web 伺服器記錄層級設定為 `DEBUG` 或顯示詳細訊息，負載平衡器會將 HTTP 階段作業 ID 寫入 Web 伺服器記錄檔中。因此，如果主控負載平衡器外掛程式的 Web 伺服器位於 DMZ 中，請勿在生產環境中使用 `DEBUG` 或類似的記錄層級。

如果必須使用 `DEBUG` 記錄層級，請在 `loadbalancer.xml` 中將 `require-monitor-data` 特性設定為 `false`，以關閉負載平衡器記錄。

▼ 開啓負載平衡器記錄

1 設定 Web 伺服器中的記錄選項。該程序取決於 Web 伺服器：

- 對於 **Sun Java System Web Server**
在伺服器的 Admin console 中，請至 [Magnus Editor] 標籤，然後將 [Log Verbose] 選項設定為 [On]。
- 對於 **Apache Web Server**，請將記錄層級設定為 `DEBUG`。
- 對於 **Microsoft IIS**，請在 `sun-passthrough.properties` 檔案中，將記錄層級設定為 `FINE`。

2 將負載平衡器配置的 `monitor` 選項設定為 `true`。

使用 `asadmin create-http-lb-config` 指令可以在最初建立負載平衡器配置時將監視設定為 `true`，也可以稍後使用 `asadmin set` 指令將其設定為 `true`。依預設，停用監視。

瞭解監視訊息

負載平衡器外掛程式記錄訊息的格式如下所示：

- HTTP 請求的黃Y處包含以下資訊：

RequestStart Sticky(New) <req-id> <time-stamp> <URL>

時間戳記值是從 1970 年 1 月 1 日開始的毫秒數。例如：

RequestStart New 123456 602983

http://austen.sun.com/Webapps-simple/servlet/Example1

- HTTP 請求的結尾處包含 RequestExit 訊息，如下所示：

RequestExit Sticky(New) <req-id> <time-stamp> <URL> <listener-id>

<response-time> Failure-<reason for error>(incase of a failure)

例如：

RequestExit New 123456 603001

http://austen.sun.com/Webapps-simple/servlet/Example1 http://austen:2222 18

備註 – 在 RequestExit 訊息中，*response-time* 表示從負載平衡器外掛程式的角度來看，請求往返所耗的總時間(以毫秒為單位)。

- 異常實例清單，如下所示：

UnhealthyInstances <cluster-id> <time-stamp> <listener-id>, <listener-id>...

例如：

UnhealthyInstances cluster1 701923 http://austen:2210, http://austen:3010

- 容錯移轉請求清單，如下所示：

FailedoverRequest <req-id> <time-stamp> <URL> <session-id>

<failed-over-listener-id> <unhealthy-listener-id>

例如：

FailedoverRequest 239496 705623

http://austen.sun.com/Apps/servlet/SessionTest 16dfdac3c7e80a40

http://austen:4044 http://austen:4045

使用 Application Server 叢集

本章說明如何使用 Application Server 叢集。它包含以下小節：

- 第 147 頁的「叢集簡介」
- 第 147 頁的「群組管理服務」
- 第 149 頁的「使用叢集」

叢集簡介

叢集是共用相同應用程式、資源和配置資訊的已命名伺服器實例集合。可以將不同機器上的伺服器實例分組為一個邏輯叢集，然後將其做為一個單位來管理。可以使用 DAS 輕鬆控制多機器叢集的使用週期。

叢集可啓用水平可延伸性、負載平衡和容錯移轉保護。根據定義，叢集中的所有實例均具有相同的資源和應用程式配置。叢集中的伺服器實例或機器故障時，負載平衡器會偵測到該故障，並將通訊從出現故障的實例重新導向至叢集中的其他實例，然後回復使用者階段作業狀態。因為相同應用程式和資源均位於叢集中的所有實例上，因此實例可以容錯移轉至叢集中的任何其他實例。

群組管理服務

群組管理服務 (GMS) 是針對叢集中實例所啓用的基礎架構元件。啓用 GMS 時，如果叢集實例失敗，則叢集和 Domain Administration Server 會知道叢集實例失敗的狀況並採取行動。Application Server 的許多功能皆取決於 GMS。例如，IIOP 容錯移轉、記憶體中複製、作業事件服務及計時器服務功能會使用 GMS。

如果叢集中的各伺服器實例位於不同機器上，請務必使這些機器位於相同的子網路中。

備註 – 開發者設定檔無法使用 GMS 功能。叢集設定檔和企業設定檔依預設會啟用 GMS。

GMS 是 Shoal 架構的核心服務。如需有關 Shoal 的資訊，請造訪 [Project Shoal 首頁](https://shoal.dev.java.net/) (<https://shoal.dev.java.net/>)。

▼ 啓用或停用叢集的 GMS 功能

- 1 在樹形元件中，選取 [叢集]。
- 2 按一下叢集的名稱。
- 3 在 [一般資訊] 下，請務必視需要核取或取消核取 [啓用活動訊號] 核取方塊。
 - 若要啓用 GMS，請務必核取 [啓用活動訊號] 核取方塊。
 - 若要停用 GMS，請務必取消核取 [啓用活動訊號] 核取方塊。
- 4 如果啓用 GMS，且需要為這些預設值設定不同值，請變更 GMS 的預設連接埠和 IP 位址。
- 5 按一下 [儲存]。

配置 GMS

如需為作業環境配置 GMS，請變更設定以決定 GMS 檢查失敗的頻率。例如，您可以變更失敗偵測嘗試之間的逾時、可能失敗成員的重試次數，或檢查叢集成員時的逾時。

若要在管理主控台中配置監視，請移至 Application Server 節點 -> [配置] -> [群組管理服務]。

等效的 asadmin 指令有 get 及 set。

使用叢集

- 第 149 頁的「建立叢集」
- 第 150 頁的「為叢集建立伺服器實例」
- 第 151 頁的「配置叢集」
- 第 152 頁的「啟動、停止和刪除叢集實例」
- 第 152 頁的「配置叢集中的伺服器實例」
- 第 153 頁的「為叢集配置應用程式」
- 第 153 頁的「為叢集配置資源」
- 第 154 頁的「刪除叢集」
- 第 154 頁的「遷移 EJB 計時器」
- 第 155 頁的「升級元件而不會使服務受到損失」

▼ 建立叢集

- 1 在樹形元件中，選取[叢集]節點。
- 2 在[叢集]頁面中，按一下[新建]。
將顯示 [建立叢集] 頁面。
- 3 在[名稱]欄位中，鍵入叢集的名稱。
名稱必須符合下列要求：
 - 僅由大小寫字母、數字、底線、連字符和點號(.)組成
 - 與所有節點代理程式名稱、伺服器實例名稱、叢集名稱和配置名稱都不重複
 - 不能是 domain
- 4 在[配置]欄位中，從下拉式清單中選擇配置。
 - 若要建立不使用共同配置的叢集，請選擇 default-config。
保留選取標有「複製已選取的配置」的單選按鈕。預設配置副本的名稱將為 *cluster_name-config*。
 - 若要建立使用共用配置的叢集，請從下拉式清單中選擇配置。
選取標有「參照已選取的配置」的單選按鈕，以建立使用指定的現有共用配置之叢集。
- 5 可選擇地增加伺服器實例。
還可以在建立叢集後增加伺服器實例。
為叢集新增伺服器實例之前，請先建立一個或多個節點代理程式或節點代理程式預留位置字元。請參閱第 174 頁的「建立節點代理程式預留位置」

若要建立伺服器實例，請執行以下作業：

- a. 在 [要建立的伺服器實例] 區域，按一下 [新增]。
 - b. 在 [實例名稱] 欄位中為實例輸入名稱。
 - c. 從 [節點代理程式] 下拉式清單中選擇節點代理程式。
- 6 按一下 [確定]。
 - 7 在顯示的 [已成功建立叢集] 頁面中按一下 [確定]。

更多資訊 等效的 asadmin 指令

```
create-cluster
```

另請參閱

- [第 151 頁的「配置叢集」](#)
- [第 150 頁的「為叢集建立伺服器實例」](#)
- [第 153 頁的「為叢集配置應用程式」](#)
- [第 153 頁的「為叢集配置資源」](#)
- [第 154 頁的「刪除叢集」](#)
- [第 155 頁的「升級元件而不會使服務受到損失」](#)

如需有關如何管理叢集、伺服器實例和節點代理程式的更多詳細資訊，請參閱[第 167 頁的「部署節點代理程式」](#)。

▼ 為叢集建立伺服器實例

開始之前 為叢集建立伺服器實例之前，必須先建立節點代理程式或節點代理程式預留位置字元。請參閱[第 174 頁的「建立節點代理程式預留位置」](#)

- 1 在樹形元件中，展開 [叢集] 節點。
- 2 選擇叢集的節點。
- 3 按一下 [實例] 標籤以顯示 [叢集伺服器實例] 頁面。
- 4 按一下 [新建] 以顯示 [建立叢集伺服器實例] 頁面。
- 5 在 [名稱] 欄位中，鍵入伺服器實例的名稱。
- 6 從 [節點代理程式] 下拉式清單中選擇節點代理程式。
- 7 按一下 [確定]。

更多資訊 等效的 asadmin 指令

`create-instance`

- 另請參閱
- 第 165 頁的「何為節點代理程式？」
 - 第 149 頁的「建立叢集」
 - 第 151 頁的「配置叢集」
 - 第 153 頁的「為叢集配置應用程式」
 - 第 153 頁的「為叢集配置資源」
 - 第 154 頁的「刪除叢集」
 - 第 155 頁的「升級元件而不會使服務受到損失」
 - 第 152 頁的「配置叢集中的伺服器實例」

▼ 配置叢集

- 1 在樹形元件中，展開[叢集]節點。

- 2 選擇叢集的節點。

在[一般資訊]頁面中，您可以執行以下作業：

- 按一下[啓動實例]以啓動叢集伺服器實例。
- 按一下[停止實例]以停止叢集伺服器實例。
- 按一下[遷移 EJB 計時器]以將 EJB 計時器從已停止的伺服器實例遷移到叢集中的其他伺服器實例。

更多資訊 等效的 asadmin 指令

`start-cluster`、`stop-cluster`、`migrate-timers`

- 另請參閱
- 第 149 頁的「建立叢集」
 - 第 150 頁的「為叢集建立伺服器實例」
 - 第 153 頁的「為叢集配置應用程式」
 - 第 153 頁的「為叢集配置資源」
 - 第 154 頁的「刪除叢集」
 - 第 155 頁的「升級元件而不會使服務受到損失」
 - 第 154 頁的「遷移 EJB 計時器」

▼ 啟動、停止和刪除叢集實例

- 1 在樹形元件中，展開 [叢集] 節點。
- 2 展開包含伺服器實例的叢集的節點。
- 3 按一下 [實例] 標籤以顯示 [叢集伺服器實例] 頁面。
在此頁面上，您可以：
 - 選取實例對應的核取方塊，然後按一下 [刪除]、[啟動] 或 [停止]，以在所有指定的伺服器實例上執行已選取的動作。
 - 按一下實例的名稱，以顯示 [一般資訊] 頁面。

▼ 配置叢集中的伺服器實例

- 1 在樹形元件中，展開 [叢集] 節點。
- 2 展開包含伺服器實例的叢集的節點。
- 3 選取伺服器實例節點。
- 4 在 [一般資訊] 頁面上，您可以：
 - 按一下 [啟動實例] 以啟動實例。
 - 按一下 [停止實例] 以停止正在執行的實例。
 - 按一下 [JNDI 瀏覽] 以瀏覽正在執行的實例的 JNDI 樹。
 - 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。
 - 按一下 [恢復作業事件] 以恢復未完成的作業事件。
 - 按一下 [特性] 標籤，以修改實例的連接埠號。
 - 按一下 [監視] 標籤，以變更監視特性。

- 另請參閱
- [第 149 頁的「建立叢集」](#)
 - [第 151 頁的「配置叢集」](#)
 - [第 150 頁的「為叢集建立伺服器實例」](#)
 - [第 153 頁的「為叢集配置應用程式」](#)
 - [第 153 頁的「為叢集配置資源」](#)

- [第 154 頁的「刪除叢集」](#)
- [第 155 頁的「升級元件而不會使服務受到損失」](#)
- 「Sun Java System Application Server 9.1 管理指南」中的「配置 Application Server 如何從作業事件中回復」

▼ 為叢集配置應用程式

- 1 在樹形元件中，展開 [叢集] 節點。
- 2 選擇叢集的節點。
- 3 按一下 [應用程式] 標籤，以顯示 [應用程式] 頁面。
在此頁面上，您可以：
 - 從 [部署] 下拉式清單中，選取要部署的應用程式的類型。在顯示的 [部署] 頁面中，指定應用程式。
 - 從 [篩選器] 下拉式清單中，選取要在清單中顯示的應用程式的類型。
 - 若要編輯應用程式，請按一下應用程式名稱。
 - 選取應用程式旁邊的核取方塊，然後選擇 [啟用] 或 [停用] 以啟用或停用用於叢集的應用程式。

另請參閱

- [第 149 頁的「建立叢集」](#)
- [第 151 頁的「配置叢集」](#)
- [第 150 頁的「為叢集建立伺服器實例」](#)
- [第 153 頁的「為叢集配置資源」](#)
- [第 154 頁的「刪除叢集」](#)
- [第 155 頁的「升級元件而不會使服務受到損失」](#)

▼ 為叢集配置資源

- 1 在樹形元件中，展開 [叢集] 節點。
- 2 選擇叢集的節點。
- 3 按一下 [資源] 標籤，以顯示 [資源] 頁面。
在此頁面上，您可以：
 - 為叢集建立新的資源：從 [新建] 下拉式清單中，選取要建立的資源類型。建立資源時，請確定將叢集指定為目標。

- 全域啟用或停用資源：選取資源旁的核取方塊，並按一下 [啟用] 或 [停用]。該動作不會移除資源。
- 僅顯示特定類型的資源：從 [篩選器] 下拉式清單中，選取要在清單中顯示的資源類型。
- 編輯資源：按一下資源名稱。

- 另請參閱
- [第 149 頁的「建立叢集」](#)
 - [第 151 頁的「配置叢集」](#)
 - [第 150 頁的「為叢集建立伺服器實例」](#)
 - [第 153 頁的「為叢集配置應用程式」](#)
 - [第 154 頁的「刪除叢集」](#)

▼ 刪除叢集

- 1 在樹形元件中，選取[叢集]節點。
- 2 在[叢集]頁面中，選取叢集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-cluster
```

- 另請參閱
- [第 149 頁的「建立叢集」](#)
 - [第 151 頁的「配置叢集」](#)
 - [第 150 頁的「為叢集建立伺服器實例」](#)
 - [第 153 頁的「為叢集配置應用程式」](#)
 - [第 153 頁的「為叢集配置資源」](#)
 - [第 155 頁的「升級元件而不會使服務受到損失」](#)

▼ 遷移 EJB 計時器

如果伺服器實例非正常或未預期地停止執行，則可能需要將該伺服器實例上安裝的 EJB 計時器移至叢集中正在執行的伺服器實例。若要完成此操作，請執行以下步驟：

- 1 在樹形元件中，展開[叢集]節點。
- 2 選擇叢集的節點。
- 3 在[一般資訊]頁面中，按一下[遷移 EJB 計時器]。

- 4 在[遷移 EJB 計時器]頁面上：
 - a. 從[源]下拉式清單中，選擇要遷移的計時器所在的已停止的伺服器實例。
 - b. (可選擇) 從[目標]下拉式清單中，選擇要將計時器遷移至的正在執行的伺服器實例。
如果保留此欄位為空，將隨機選擇正在執行的伺服器實例。
 - c. 按一下[確定]。
- 5 停止並重新啟動目標伺服器實例。
如果來源伺服器實例正在執行或目標伺服器實例未在執行，則管理主控台將顯示錯誤訊息。

更多資訊 等效的 asadmin 指令

migrate-timers

- 另請參閱
- [第 151 頁的「配置叢集」](#)
 - [用來配置 EJB 計時器服務設定的管理主控台線上說明](#)

▼ 升級元件而不會使服務受到損失

在 Application Server 中，您可以使用負載平衡器和多個叢集來升級元件，而不會使服務受到任何損失。例如，元件可以是 JVM、Application Server 或 Web 應用程式。

此方法在以下情況不可用：

- 變更高可用性資料庫 (HADB) 的模式時。如需更多資訊，請參閱[第 3 章](#)

備註 – Sun Java System Application Server 的 Application Server 獨立發行版本隨附了 HADB 軟體。如需有關 Sun Java System Application Server 的可用發行軟體資訊，請參閱「Sun Java System Application Server 9.1 Installation Guide」中的「Distribution Types and Their Components」。只有企業設定檔才提供 HADB 功能。如需有關設定檔的更多資訊，請參閱「Sun Java System Application Server 9.1 管理指南」中的「用法設定檔」。

- 執行應用程式升級會使應用程式資料庫模式有所變更時。



注意 – 請同時升級叢集中的所有伺服器實例。否則，可能會出現由從執行不同版本的元件的一個實例到另一個實例的階段作業容錯移轉引起的版本不匹配。

- 1 使用叢集的 [一般資訊] 頁面上的 [停止叢集] 按鈕來停止其中一個叢集。
- 2 升級該叢集中的元件。
- 3 使用叢集的 [一般資訊] 頁面上的 [啓動叢集] 按鈕來啓動叢集。
- 4 對其他叢集逐個重複此程序。

由於一個叢集中的階段作業決不會容錯移轉至其他叢集中的階段作業，因此階段作業在從執行某版本元件的伺服器實例容錯移轉至執行其他版本元件的伺服器實例 (位於其他叢集中) 時不會出現版本不相符的情況。因此，對於叢集中的伺服器實例，叢集可充當階段作業容錯移轉的安全邊界。

另請參閱

- [第 149 頁的「建立叢集」](#)
- [第 151 頁的「配置叢集」](#)
- [第 150 頁的「為叢集建立伺服器實例」](#)
- [第 153 頁的「為叢集配置應用程式」](#)
- [第 153 頁的「為叢集配置資源」](#)
- [第 154 頁的「刪除叢集」](#)

管理配置

本章描述在 Application Server 中新增、變更以及使用已命名的伺服器配置。它包含以下小節：

- [第 157 頁的「使用配置」](#)
- [第 160 頁的「使用已命名的配置」](#)

使用配置

- [第 157 頁的「配置」](#)
- [第 158 頁的「default-config 配置」](#)
- [第 158 頁的「建立實例或叢集時建立的配置」](#)
- [第 159 頁的「唯一連接埠號和配置」](#)

配置

配置為一組伺服器配置資訊，包含諸如 HTTP 偵聽程式、ORB/IIOP 偵聽程式、JMS 代理程式、EJB 容器、安全性、記錄和監視等的設定。已命名的配置中未定義應用程式和資源。

配置存在於管理網域中。該網域中的多個伺服器實例或叢集可以參照相同的配置，也可以有各自獨立的配置。

對於叢集，叢集中的所有伺服器實例都繼承了叢集的配置，從而確保叢集實例具有同質環境。

由於配置包含許多的必需設定，因此請透過複製現有已命名的配置來建立新配置。變更新建配置的配置設定之前，該配置與被複製的配置完全相同。

叢集或實例使用配置的方式有三種：

- **獨立**：獨立伺服器實例或叢集不與其他伺服器實例或叢集共用其配置，亦即其他伺服器實例或叢集不參照已命名的配置。可透過複製和重新命名現有配置來建立獨立伺服器或叢集。
- **共用**：共用伺服器實例或叢集與其他伺服器實例或叢集共用配置；也就是說，多個實例或叢集參照相同的已命名的配置。可以透過參照 (而非複製) 現有配置來建立共用伺服器實例或叢集。
- **叢集**：叢集伺服器實例繼承叢集的配置。
另請參閱：
 - [第 158 頁的「default-config 配置」](#)
 - [第 158 頁的「建立實例或叢集時建立的配置」](#)
 - [第 159 頁的「唯一連接埠號和配置」](#)
 - [第 160 頁的「建立已命名的配置」](#)
 - [第 160 頁的「編輯已命名的配置的特性」](#)

default-config 配置

default-config 配置是做為範本的特殊配置，用於建立獨立伺服器實例或獨立叢集配置。叢集和個別伺服器實例無法參照 default-config；僅能將其複製用以建立新配置。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。

如需更多資訊，請參閱：

- [第 158 頁的「建立實例或叢集時建立的配置」](#)
- [第 157 頁的「配置」](#)
- [第 160 頁的「建立已命名的配置」](#)
- [第 160 頁的「編輯已命名的配置的特性」](#)
- [第 161 頁的「編輯參照配置之實例的連接埠號」](#)

建立實例或叢集時建立的配置

建立新伺服器實例或新叢集時，可以執行以下操作之一：

- 參照現有配置。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。

依預設，您必須複製 default-config 配置中的配置，以建立新叢集或實例。若要從其他配置進行複製，請在建立新實例或叢集時指定要複製的配置。

對於伺服器實例，新配置命名為 *instance_name* -config。對於叢集，新配置命名為 *cluster-name* -config。

如需更多資訊，請參閱：

- [第 158 頁的「default-config 配置」](#)
- [第 157 頁的「配置」](#)
- [第 160 頁的「建立已命名的配置」](#)
- [第 160 頁的「編輯已命名的配置的特性」](#)

叢集配置同步化

建立叢集配置時，Application Server 會在

`domain-root/domain-dir/config/cluster-config` 內的網域管理伺服器上建立叢集配置目錄。本目錄用於同步化叢集中所有實例的配置。

唯一連接埠號和配置

如果同一主機電腦上的多個實例參照相同的配置，則每個實例均必須偵聽唯一連接埠號。例如，如果兩個伺服器實例都參照某個已命名的配置，並且該配置包含一個位於連接埠 80 上的 HTTP 偵聽程式，則連接埠衝突將阻止其中一個伺服器實例啟動。變更用於定義連接埠號 (各個伺服器實例在這些連接埠號上進行偵聽) 的特性，從而確定各個實例使用唯一的連接埠。

以下原則適用於連接埠號設定：

- 各個伺服器實例的連接埠號最初是從配置繼承而來的。
- 建立伺服器實例時，如果該連接埠號已經被使用，則請在實例層級上置換繼承的預設值，以防止發生連接埠衝突。
- 假設實例正在共用配置。配置連接埠號為 n 。如果使用相同的配置在機器上建立新實例，則為新實例指定的連接埠號為 $n+1$ (如果此連接埠號可用)。如果此連接埠號不可用，將選擇 $n+1$ 後下一個可用的連接埠。
- 如果您變更了配置的連接埠號，則繼承該連接埠號的伺服器實例將自動繼承變更後的連接埠號。
- 如果變更了實例的連接埠號，然後又變更了配置的連接埠號，則實例的連接埠號將保持不變。

如需更多資訊，請參閱：

- [第 161 頁的「編輯參照配置之實例的連接埠號」](#)
- [第 160 頁的「編輯已命名的配置的特性」](#)
- [第 157 頁的「配置」](#)

使用已命名的配置

- [第 160 頁的「建立已命名的配置」](#)
- [第 160 頁的「編輯已命名的配置的特性」](#)
- [第 161 頁的「編輯參照配置之實例的連接埠號」](#)
- [第 162 頁的「檢視已命名的配置的目標」](#)
- [第 162 頁的「刪除已命名的配置」](#)

▼ 建立已命名的配置

- 1 在樹形元件中，選取 [配置] 節點。
- 2 在 [配置] 頁面中，按一下 [新建]。
- 3 在 [建立配置] 頁面中，為配置輸入唯一名稱。
- 4 選擇要複製的配置。

default-config 配置是建立獨立伺服器實例或獨立叢集時所使用的預設配置。

更多資訊 等效的 asadmin 指令

copy-config

另請參閱

- [第 157 頁的「配置」](#)
- [第 158 頁的「default-config 配置」](#)
- [第 160 頁的「編輯已命名的配置的特性」](#)
- [第 161 頁的「編輯參照配置之實例的連接埠號」](#)
- [第 162 頁的「檢視已命名的配置的目標」](#)
- [第 162 頁的「刪除已命名的配置」](#)

編輯已命名的配置的特性

下表說明為配置預先定義的特性。

預先定義的特性為連接埠號。有效的連接埠值為 1–65535。在 UNIX 上，建立偵聽連接埠 1–1024 的通訊端需要超級使用者權限。如果系統中存在多個伺服器實例，則它們各自的連接埠號必須是唯一的。

特性名稱	說明
HTTP_LISTENER_PORT	http-listener-1 的連接埠號。
HTTP_SSL_LISTENER_PORT	http-listener-2 的連接埠號。
IIOP_SSL_LISTENER_PORT	用於 IIOP 連線的 ORB 偵聽程式連接埠 (IIOP 偵聽程式 SSL_MUTUALAUTH 在其上進行偵聽)。
IIOP_LISTENER_PORT	orb-listener-1 偵聽的用於 IIOP 連線的 ORB 偵聽程式連接埠。
JMX_SYSTEM_CONNECTOR_PORT	JMX 連接器偵聽的連接埠號。
IIOP_SSL_MUTUALAUTH_PORT	IIOP 偵聽程式 SSL_MUTUALAUTH 在其上進行偵聽的用於 IIOP 連線的 ORB 偵聽程式連接埠。

▼ 編輯已命名的配置的特性

- 1 在樹形元件中，展開 [配置] 節點。
- 2 選取已命名的配置的節點。
- 3 在 [配置系統特性] 頁面中，選擇是否啓用動態重新配置。
如果已啓用，則對配置所作的變更將套用至伺服器實例，而無需重新啓動伺服器。
- 4 依需要增加、刪除或修改特性。
- 5 若要編輯與配置關聯的所有實例的特性的目前值，請按一下 [實例值]。

更多資訊 等效的 asadmin 指令

set

- 另請參閱
- [第 157 頁的「配置」](#)
 - [第 160 頁的「建立已命名的配置」](#)
 - [第 162 頁的「檢視已命名的配置的目標」](#)
 - [第 162 頁的「刪除已命名的配置」](#)

▼ 編輯參照配置之實例的連接埠號

每個參照已命名的配置的實例最初都從該配置繼承連接埠號。由於系統中的連接埠號必須是唯一的，因而可能需要置換繼承的連接埠號。

- 1 在樹形元件中，展開 [配置] 節點。

2 選取已命名的配置的節點。

管理主控台會顯示 [配置系統特性] 頁面。

3 按一下要編輯之實例變數旁的 [實例值]。

例如，如果按一下 HTTP-LISTENER-PORT 實例變數旁的 [實例值]，您會看到參照該配置之每個伺服器實例的 HTTP-LISTENER-PORT 值。

4 依需要變更值，並按一下 [儲存]。

更多資訊 等效的 asadmin 指令

set

- 另請參閱
- [第 159 頁的「唯一連接埠號和配置」](#)
 - [第 157 頁的「配置」](#)
 - [第 160 頁的「編輯已命名的配置的特性」](#)

▼ 檢視已命名的配置的目標

[配置系統特性] 頁面將顯示使用該配置的所有目標的清單。對於叢集配置，目標是叢集。對於實例配置，目標是實例。

1 在樹形元件中，展開 [配置] 節點。

2 選取已命名的配置的節點。

- 另請參閱
- [第 159 頁的「唯一連接埠號和配置」](#)
 - [第 157 頁的「配置」](#)
 - [第 160 頁的「建立已命名的配置」](#)
 - [第 160 頁的「編輯已命名的配置的特性」](#)
 - [第 162 頁的「刪除已命名的配置」](#)

▼ 刪除已命名的配置

1 在樹形元件中，選取 [配置] 節點。

2 在 [配置] 頁面中，選取要刪除的已命名的配置的核取方塊。
無法刪除 default-config 配置。

3 按一下 [刪除]。

更多資訊 **等效的 asadmin 指令**

`delete-config`

另請參閱

- [第 157 頁的「配置」](#)
- [第 160 頁的「建立已命名的配置」](#)
- [第 160 頁的「編輯已命名的配置的特性」](#)
- [第 162 頁的「檢視已命名的配置的目標」](#)

配置節點代理程式

本章說明 Application Server 中的節點代理程式。它包含以下小節：

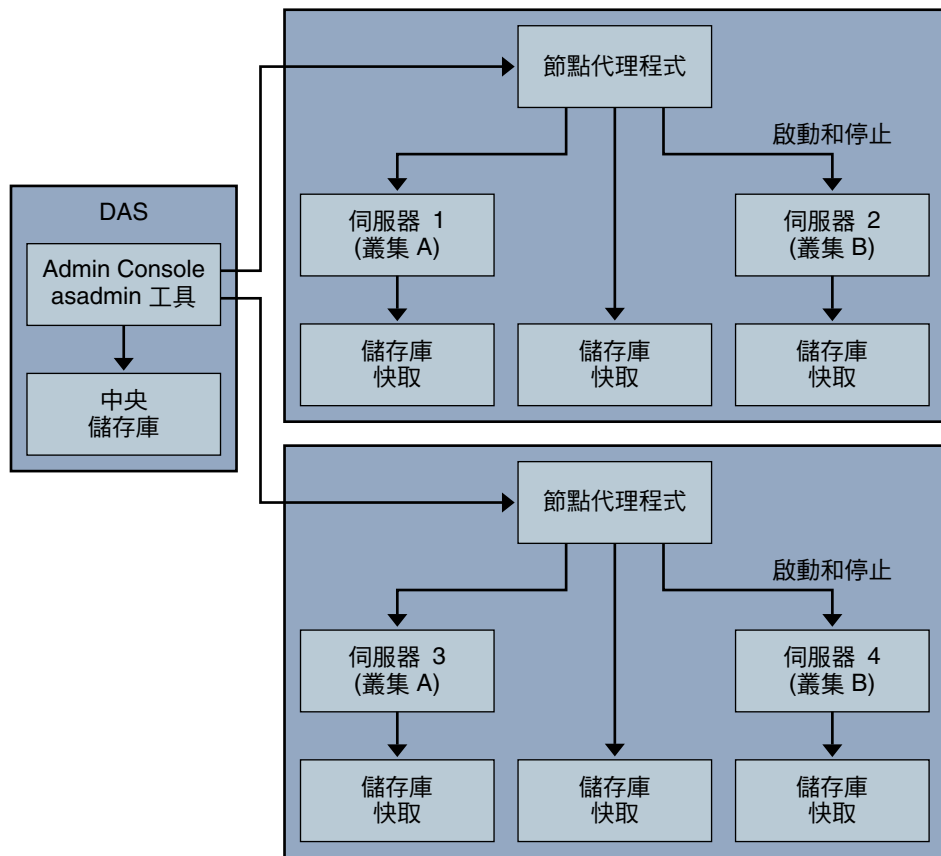
- [第 165 頁的「何為節點代理程式？」](#)
- [第 166 頁的「節點代理程式失敗後的伺服器實例運作方式」](#)
- [第 167 頁的「部署節點代理程式」](#)
- [第 169 頁的「同步化節點代理程式與網域管理伺服器」](#)
- [第 173 頁的「檢視節點代理程式記錄」](#)
- [第 173 頁的「使用節點代理程式」](#)

何為節點代理程式？

節點代理程式是託管伺服器實例的每台機器 (包括託管網域管理伺服器 (DAS) 的機器) 上，都需要的簡易程序。節點代理程式可以：

- 按照網域管理伺服器的指示啟動、停止、建立和刪除伺服器實例。
- 重新啟動有故障的伺服器實例。
- 提供有故障的伺服器的記錄檔檢視。
- 使每個伺服器實例的本機配置儲存庫與網域管理伺服器的中央儲存庫同步。每個本機儲存庫只包含與該伺服器實例或節點代理程式相關的資訊。

下圖說明了節點代理程式的整體架構：



安裝應用程式伺服器時，依預設將使用該機器的主機名稱建立節點代理程式。必須先在本地機器上手動啟動節點代理程式，該節點代理程式才能執行。

即使未執行節點代理程式，您也可以建立和刪除伺服器實例。但是，節點代理程式必須處於執行狀態，您才能用它來啟動和停止伺服器實例。

一個節點代理程式服務單一網域。如果某台機器託管在多個網域中執行的實例，則該機器必須執行多個節點代理程式。

節點代理程式失敗後的伺服器實例運作方式

例如，節點代理程式可能因軟體失敗或其他錯誤而意外地停止。在此情況下，節點代理程式將無法再管理其所管理的任何伺服器實例。不過，此類伺服器實例會繼續執行，DAS 可繼續存取該實例。仍可從 Application Server 管理介面取得伺服器實例的相關資訊，且仍可存取在伺服器實例上所部署的應用程式。

如果節點代理程式重新啟動，則未受管理的伺服器實例仍將未受管理。節點代理程式不會繼續管理這些伺服器實例。例如，如果未受管理的伺服器實例因軟體失敗或其他錯誤而意外地停止，節點代理程式就無法重新啟動伺服器實例。

如果必須繼續執行未受管理的伺服器實例，節點代理程式將無法繼續管理伺服器實例。繼續管理未受管理伺服器實例的唯一方法，是在重新啟動節點代理程式後停止伺服器實例，然後重新啟動。

部署節點代理程式

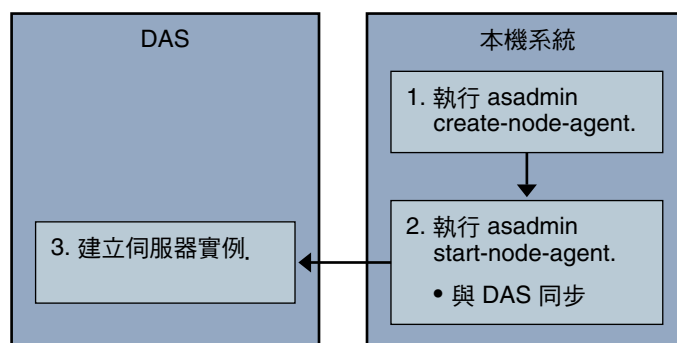
您可以使用以下兩種方法配置和部署節點代理程式：

- **線上部署**，如果您知道網域的拓樸並擁有適用於該網域的硬體，請使用線上部署。
- **離線部署**，如果是在設定整個環境之前配置網域和伺服器實例，請使用離線部署。

▼ 線上部署節點代理程式

如果您已經知道網域拓樸並擁有適用於網域的硬體，請使用線上部署。

下圖是離線部署的示意圖：



開始之前 安裝並啟動網域管理伺服器。啟動並執行之後，便可以進行線上或離線部署。

1 在將要託管伺服器實例的每台機器上安裝節點代理程式。

使用安裝程式或 `asadmin create-node-agent` 指令。如果某台機器需要多個節點代理程式，請使用 `asadmin create-node-agent` 指令來建立這些節點代理程式。

請參閱第 175 頁的「[建立節點代理程式](#)」，以取得更多資訊。

2 使用 `asadmin start-node-agent` 指令啟動節點代理程式。

啟動之後，節點代理程式將與網域管理伺服器 (DAS) 進行通訊。當節點代理程式到達 DAS 時，將在 DAS 上建立該節點代理程式的配置。具備了配置之後，即可在管理主控台中檢視該節點代理程式。

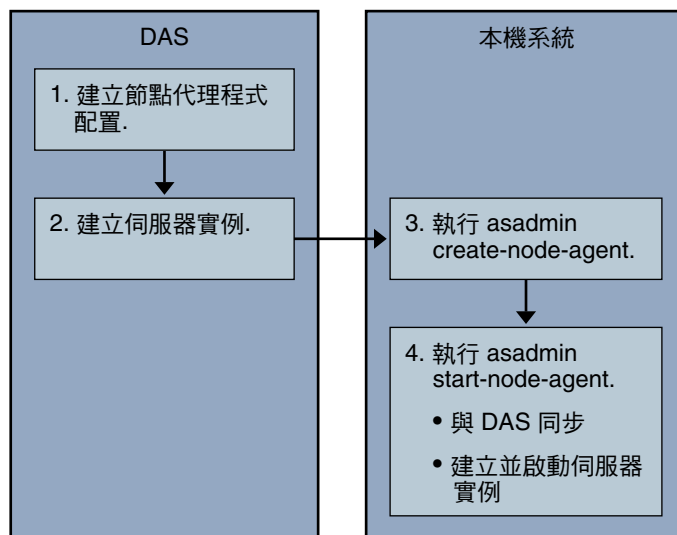
請參閱第 177 頁的「啟動節點代理程式」，以取得更多資訊。

- 3 配置網域：建立伺服器實例、叢集，並部署應用程式。

▼ 離線部署節點代理程式

在配置個別本地機器之前，使用離線部署在網域中部署節點代理程式。

下圖概括了離線部署。



開始之前 安裝並啟動網域管理伺服器。啟動並執行之後，便可以進行線上或離線部署。

- 1 在網域管理伺服器中建立預留位置節點代理程式。

請參閱第 174 頁的「建立節點代理程式預留位置」，以取得更多資訊。

- 2 建立伺服器實例和叢集並部署應用程式。

建立伺服器實例時，請確保指定的是未處於使用中的連接埠號。由於是離線完成配置，因此網域無法在建立時檢查連接埠衝突。

- 3 在將要託管伺服器實例的每台機器上安裝節點代理程式。

使用安裝程式或 `asadmin create-node-agent` 指令。節點代理程式的名稱必須與先前建立的預留位置節點代理程式相同。

請參閱第 175 頁的「建立節點代理程式」，以取得更多資訊。

4 使用 `asadmin start-node-agent` 指令啟動節點代理程式。

節點代理程式啟動之後，將連結到網域管理伺服器並建立先前已與該節點代理程式相關聯的所有伺服器實例。

請參閱第 177 頁的「啟動節點代理程式」，以取得更多資訊。

同步化節點代理程式與網域管理伺服器

由於配置資料儲存在網域管理伺服器的儲存庫 (中央儲存庫) 中，並在節點代理程式的本地機器上快取，因此，兩者必須同步化。使用者必須透過管理工具明確執行動作才能使快取同步。

本小節包含下列主題：

- 第 169 頁的「節點代理程式同步」
- 第 170 頁的「伺服器實例同步」
- 第 171 頁的「同步程式庫檔案」
- 第 171 頁的「獨特的設定和配置管理」
- 第 172 頁的「同步大型應用程式」

節點代理程式同步

第一次啟動節點代理程式時，該節點代理程式將向網域管理伺服器 (DAS) 傳送一個請求，以獲得中央儲存庫中的最新資訊。當節點代理程式成功地與 DAS 取得聯絡並獲得配置資訊時，該節點代理程式即被連結到該 DAS。

備註 – 依預設，`asadmin start-node-agent` 指令將自動啟動遠端伺服器實例，而不會與 DAS 同步。如果您要啟動的遠端伺服器實例已經與 DAS 管理的中央儲存庫同步，請指定 `asadmin start-node-agent` 指令的 `--startinstances=false` 選項。然後使用 `asadmin start-instance` 指令啟動遠端伺服器實例。

如果您在 DAS 上建立了預留位置節點代理程式，則第一次啟動節點代理程式時，該節點代理程式將從 DAS 的中央儲存庫中獲取其配置。初始啟動節點代理程式期間，如果由於沒有執行 DAS 而使該節點代理程式無法連結 DAS，該節點代理程式將停止並保持未連結狀態。

如果在網域中變更了節點代理程式的配置，這些變更將在節點代理程式執行時自動傳送給本地機器中的節點代理程式。

如果在 DAS 中刪除了節點代理程式的一個配置，則下次該節點代理程式進行同步時將自行停止，並將自身標記為待刪除狀態。使用本機 `asadmin delete-node-agent` 指令手動將其刪除。

伺服器實例同步

如果使用管理主控台或 `asadmin` 工具明確地啟動伺服器實例，則伺服器實例將與中央儲存庫同步。如果此同步失敗，則伺服器實例不會啟動。

如果節點代理程式啟動伺服器實例的方法並非向管理主控台或 `asadmin` 工具傳送明確的請求，則該伺服器實例的儲存庫快取不會同步。該伺服器實例將以儲存在其快取中的配置執行。您不能在遠端伺服器實例的快取中增加或移除檔案。

遠端伺服器實例的配置被視為快取 (`nodeagents/nal/server1` 下的所有檔案) 並由應用程式伺服器擁有。在極少數情況下，如果使用者移除了遠端伺服器實例的所有檔案並重新啟動節點代理程式，則將重新建立遠端伺服器實例 (例如，`server1`) 並同步所有必需的檔案。

應用程式伺服器將使以下檔案和目錄繼續保持同步。

表 8-1 在遠端伺服器實例之間同步的檔案和目錄

檔案或目錄	說明
<code>applications</code>	所有已部署的應用程式。此目錄 (及子目錄) 已同步的部分取決於伺服器實例參照的應用程式。節點代理程式不會同步應用程式，因為它沒有參照任何應用程式。
<code>config</code>	包含整個網域的配置檔案。此目錄中的所有檔案均會同步，但執行階段暫存檔案 (例如 <code>admch</code> 、 <code>admsn</code> 、 <code>secure.seed</code> 、 <code>.timestamp</code> 和 <code>__timer_service_shutdown__.dat</code>) 除外。
<code>config/config_name</code>	用於儲存要由所有實例 (使用名為 <code>config_name</code> 的配置) 共用的檔案的目錄。在 <code>domain.xml</code> 中定義的每個配置都會有這樣的目錄。此目錄中的所有檔案均與使用 <code>config_name</code> 的伺服器實例同步。
<code>config/config_name/lib/ext</code>	放置 Java 延伸類別 (如 <code>zip</code> 或 <code>jar</code> 歸檔檔案) 的資料夾。此資料夾供部署到伺服器實例 (使用名為 <code>config_name</code> 的配置) 的應用程式使用。您必須使用 Java 延伸機制載入這些 <code>jar</code> 檔案。
<code>docroot</code>	HTTP 文件根目錄。在即開即用 (<code>out of the box</code>) 配置中，網域中的所有伺服器實例均使用相同的 <code>docroot</code> 。需要將虛擬伺服器的 <code>docroot</code> 特性配置為使伺服器實例使用不同的 <code>docroot</code> 。
<code>generated</code>	已產生的 Java EE 應用程式檔案和模組檔案，例如 <code>EJB stub</code> 、已編譯的 <code>JSP</code> 類別和安全策略檔案。此目錄與 <code>applications</code> 目錄同時同步。因此，僅同步應用程式 (由伺服器實例所參照) 對應的目錄。
<code>lib</code> 、 <code>lib/classes</code>	可放置共用 Java 類別檔案或者 <code>jar</code> 和 <code>zip</code> 歸檔檔案 (供部署到整個網域的應用程式使用) 的資料夾。您必須使用應用程式伺服器的類別載入器載入這些類別。類別載入器的載入順序為： <code>lib/classes</code> 、 <code>lib/*.jar</code> 、 <code>lib/*.zip</code> 。

表 8-1 在遠端伺服器實例之間同步的檔案和目錄 (續)

檔案或目錄	說明
lib/ext	可放置 Java 延伸類別 (例如 zip 或 jar 歸檔檔案，以供部署到整個網域的應用程式使用) 的資料夾。您必須使用 Java 延伸機制載入這些 jar 檔案。
lib/applibs	將附屬 jar 置於 domains/<domain_name>lib/applibs 之下，並透過 libraries 選項，指定 jar 檔案之相對路徑。 例如，asadmin deploy --libraries commons-coll.jar,X1.jar foo.ear
java-web-start	此目錄 (與子目錄) 同步的部分，是根據從伺服器實例所參照的應用程式而決定。

同步程式庫檔案

您可以使用應用程式的 --libraries 部署時間屬性來指定應用程式的執行階段相依性。指定相對路徑後 (jar 名稱即可)，Application Server 會嘗試在 domain-dir /lib/applibs 中找出指定的程式庫。

為使程式庫可用於整個網域，可以將 JAR 檔案放在 domain-dir/lib 或 domain-dir/lib/classes 中。(如需更多資訊，請參閱「Sun Java System Application Server 9.1 Developer's Guide」中的「Using the Common Class Loader」。)JDBC 驅動程式以及網域中所有應用程式共用的其他公用程式程式庫，通常需要如此。

為了用在整個叢集或整個獨立伺服器中，請將 jar 複製到 domain-dir/domain1/config/xyz-config/lib 目錄中。接下來，將 jar 增加至 xyz-config 的 classpath-suffix 或 classpath-prefix 元素中。這將針對所有使用 xyz-config 的伺服器實例同步 jar。

總而言之：

- domains/domain1/lib - 用於整個網域範圍、使用共用類別載入器、自動增加 jar。
- domains/domain1/config/cluster1、config/lib - 用於配置範圍、更新 classpath-prefix 或 classpath-suffix。
- domains/domain1/lib/applibs - 範圍為應用程式，自動增加至應用程式類別載入器。
- domains/domain1/config/cluster1,config/lib/ext - 自動增加至 <http://java.sun.com/j2se/1.5.0/docs/guide/extensions/extensions.html>。

獨特的設定和配置管理

整個網域中的配置檔案 (位於 domains/domain1/config 下) 都會同步。如果要為獨立伺服器實例 (server1) 使用的 server1-config 自訂 server.policy 檔案，請將修改後的 server.policy 檔案放在 domains/domain1/config/server1-config 目錄下。

此修改後的 `server.policy` 檔案僅針對獨立伺服器實例 `server1` 進行同步。請記得更新 `jvm-option`。例如：`<jvm-config> ...`
`<jvm-options>-Djava.security.policy=${com.sun.aas.instanceRoot}/config`
`/server1-config/server.policy</jvm-options></java-config>`

同步大型應用程式

當環境中有大型應用程式要進行同步或者可用記憶體受到限制時，您可以調整 JVM 選項以限制記憶體的使用。這種調整將減少收到記憶體不足錯誤的可能性。實例同步 JVM 使用的是預設設定，但您可以配置 JVM 選項來變更這些設定。

使用 `INSTANCE-SYNC-JVM-OPTIONS` 特性設定 JVM 選項。用於設定特性的指令為：

```
asadmin set
domain.node-agent.node_agent_name.property.INSTANCE-SYNC-JVM-OPTIONS="JVM_options"
```

例如：

```
asadmin set
domain.node-agent.node0.property.INSTANCE-SYNC-JVM-OPTIONS="-Xmx32m -Xss2m"
```

在此範例中，節點代理程式為 `node0`，JVM 選項為 `-Xmx32m -Xss2m`。

如需更多資訊，請參閱 <http://java.sun.com/docs/hotspot/VMOptions.html>。

備註 – 變更 `INSTANCE-SYNC-JVM-OPTIONS` 特性後，請重新啟動節點代理程式，因為在節點代理程式的配置中新增或變更特性時節點代理程式並不會自動同步。

使用 `doNotRemoveList` 標幟

若您的應用程式需要儲存和讀取透過應用程式伺服器所同步之目錄 (`applications`、`generated`、`docroot`、`config`、`lib` 以及 `java-web-star`) 中的檔案，請使用 `doNotRemoveList` 旗標。此屬性接受以逗號分隔的檔案或目錄清單。即使應用程式相關檔案不存在於 DAS 所管理的中央儲存庫中，在伺服器啟動期間也不會移除這些檔案。如果中央儲存庫中存在相同的檔案，則在同步期間將覆寫這些檔案。

使用 `INSTANCE-SYNC-JVM-OPTIONS` 特性傳入 `doNotRemoveList` 屬性。

例如：

```
<node-agent name="na1" ...>
...
<property name="INSTANCE-SYNC-JVM-OPTIONS"
value="-Dcom.sun.appserv.doNotRemoveList=applications/j2ee-modules
/<webapp_context>/logs/generated/mylogdir"/>
```

```
</node-agent>
```

檢視節點代理程式記錄

每個節點代理程式均有其自己的記錄檔。如果使用節點代理程式時遇到問題，請參閱位於以下位置的記錄檔：

`node_agent_dir/node_agent_name/agent/logs/server.log`。

有時，節點代理程式記錄會指示您檢視伺服器的記錄，以取得有關所出現問題的詳細訊息。

伺服器記錄位於：

`node_agent_dir/node_agent_name/server_name/logs/server.log`

`node_agent_dir`的預設位置為 `install_dir/nodeagents`

使用節點代理程式

- [第 173 頁的「如何執行節點代理程式作業」](#)
- [第 174 頁的「節點代理程式預留位置」](#)
- [第 174 頁的「建立節點代理程式預留位置」](#)
- [第 175 頁的「建立節點代理程式」](#)
- [第 177 頁的「啟動節點代理程式」](#)
- [第 177 頁的「停止節點代理程式」](#)
- [第 178 頁的「刪除節點代理程式」](#)
- [第 178 頁的「檢視一般節點代理程式資訊」](#)
- [第 179 頁的「刪除節點代理程式配置」](#)
- [第 179 頁的「編輯節點代理程式配置」](#)
- [第 180 頁的「編輯節點代理程式範圍」](#)
- [第 180 頁的「為 JMX 編輯節點代理程式的偵聽程式」](#)

如何執行節點代理程式作業

部分節點代理程式作業要求您在節點代理程式的執行系統本機上使用 `asadmin` 工具。您可以於遠端使用管理主控台或 `asadmin` 執行其他作業。

下表概括了這些作業以及執行這些作業的位置：

表 8-2 如何執行節點代理程式作業

作業	管理主控台	asadmin 指令
在網域管理伺服器中建立節點代理程式預留位置	[建立節點代理程式預留位置] 頁面	create-node-agent-config
建立節點代理程式	不可用	create-node-agent
啟動節點代理程式	不可用	start-node-agent
停止節點代理程式	不可用	stop-node agent
從網域管理伺服器中刪除節點代理程式配置	[節點代理程式] 頁面	delete-node-agent-config
從本地機器中刪除節點代理程式	不可用	delete-node-agent
編輯節點代理程式配置	[節點代理程式] 頁面	set
列出節點代理程式	[節點代理程式] 頁面	list-node-agents

節點代理程式預留位置

您可以在沒有現有節點代理程式的情況下，使用**節點代理程式預留位置**建立和刪除伺服器實例。節點代理程式預留位置是在節點代理程式的本機系統中建立節點代理程式本身之前，建立於網域管理伺服器 (DAS) 上的。

如需有關建立節點代理程式預留位置的資訊，請參閱第 174 頁的「[建立節點代理程式預留位置](#)」。

備註 – 當您建立預留位置節點代理程式之後，即可使用其在網域中建立實例。但是在啟動實例之前，您必須先使用 `asadmin` 指令在實例將要常駐的機器上，直接從本機建立並啟動實際的節點代理程式。請參閱第 175 頁的「[建立節點代理程式](#)」和第 177 頁的「[啟動節點代理程式](#)」。

▼ 建立節點代理程式預留位置

節點代理程式是一個本機監視程式，可監視遠端機器上所執行的伺服器實例。因此必須在主控伺服器實例的電腦上建立節點代理程式。爲了滿足此要求，您可以使用管理主控台，僅爲節點代理程式建立預留位置。此預留位置是尚不存在的節點代理程式的節點代理程式配置。

建立預留位置之後，請在主控節點代理程式的機器中，使用 `asadmin` 指令 `create-node-agent` 完成建立程序。如需更多資訊，請參閱第 175 頁的「[建立節點代理程式](#)」。

如需用於建立和使用節點代理程式之步驟的清單，請參閱第 167 頁的「部署節點代理程式」。

- 1 在樹形元件中，選取 [節點代理程式] 節點。
- 2 在 [節點代理程式] 頁面中，按一下 [新建]。
- 3 在 [目前節點代理程式預留位置] 頁面中，為新節點代理程式輸入一個名稱。
此名稱必須與網域中的所有節點代理程式名稱、伺服器實例名稱、叢集名稱和配置名稱都不重複。
- 4 按一下 [確定]。
新節點代理程式的預留位置將列在 [節點代理程式] 頁面中。

更多資訊 等效的 asadmin 指令

```
create-node-agent-config
```

建立節點代理程式

若要建立節點代理程式，請在執行節點代理程式的機器上，直接從本機執行 `asadmin` 指令 `create-node-agent`。

節點代理程式的預設名稱爲該節點代理程式建立時所在的主機的名稱。

如果已建立節點代理程式預留位置，請使用與節點代理程式預留位置相同的名稱來建立相關聯的節點代理程式。如果尚未建立節點代理程式預留位置，而 DAS 已啓動並且可存取，則 `create-node-agent` 指令還將在 DAS 上建立節點代理程式配置 (預留位置)。

如需有關指令語法的完整描述，請參閱該指令的線上說明。

可配置 DAS 與節點代理程式來進行安全通訊。在此情況下啓動節點代理程式時，它必須驗證 DAS 傳送給節點代理程式的憑證。爲了驗證憑證，節點代理程式會在受主密碼保護的信任清單存放區中查找憑證。若要啓動節點代理程式而不提示使用者輸入密碼，請在建立節點代理程式時，將節點代理程式的主密碼儲存到檔案。若未將節點代理程式的主密碼儲存到檔案，則每當使用者啓動節點代理程式時，系統會提示使用者輸入主密碼。

備註 – 在某些情況下，必須指定可透過 DNS 使用的主機名稱。如需更多資訊，請參閱第 176 頁的「爲 DNS 可存取的主機建立節點代理程式」。

▼ 建立節點代理程式

- 請鍵入下列指令：

```
asadmin create-node-agent --host das-host --port port-no --user das-user
[--savemasterpassword=true] nodeagent
```

若要啟動節點代理程式而不提示使用者輸入密碼，請將節點代理程式的主密碼儲存到檔案。若要將節點代理程式的主密碼儲存到檔案，請在指令中，將 `--savemasterpassword` 選項設為 `true` 來建立節點代理程式。

如果將 `--savemasterpassword` 設為 `true`，則系統會提示您輸入主密碼。否則**不會**提示您輸入密碼。

`--host das-host` 指定正在執行 Domain Administration Server (DAS) 的主機名稱。

`-port port-no` 指定用來管理網域的 HTTP 或 HTTPS 連接埠號。

`--user das-user` 指定 DAS 使用者。

`nodeagent` 指定所建立的節點代理程式名稱。此名稱在網域中必須唯一。

範例 8-1 建立節點代理程式

```
asadmin create-node-agent --host myhost --port 4848 --user admin nodeagent1
```

此指令會建立名為 `nodeagent1` 的節點代理程式。與節點代理程式進行通訊的 DAS 正在電腦 `myhost` 上執行。用來管理代理程式網域的 HTTP 連接埠是 4848。DAS 使用者的名稱是 `admin`。

▼ 為 DNS 可存取的主機建立節點代理程式

在下列情況下，正在執行 DAS 的主機必須可以透過 DNS 存取：

- 網域橫跨子網路邊界，換言之，節點代理程式與 DAS 位於不同的網域，例如 `sun.com` 和 `java.com`。
- 正在使用 DHCP 電腦，該電腦的主機名稱並未在 DNS 中註冊。

1 若要在 `create-domain` 指令中建立網域，請指定

`--domainproperties domain.hostName=das-host-name` 選項。

`das-host-name` 是執行 DAS 的電腦名稱。

2 若要在 `create-node-agent` 指令中建立節點代理程式，請指定下列選項：

- `--host das-host-name`，其中 `das-host-name` 是您在步驟 1 中指定的 DAS 主機名稱。此選項與 `as-install/nodeagents/nodeagentname/agent/config/das.properties` 檔案中的 `agent.das.host` 特性相對應。

- `--agentproperties remoteclientaddress=node-agent-host-name`，其中 `node-agent-host-name` 是 DAS 用來連線到節點代理程式的主機名稱。此選項與 `as-install/nodeagents/nodeagentname/agent/config/nodeagent.properties` 檔案中的 `agent.client.host` 特性相對應。

更多資訊 更新 hosts 檔案以指定主機

另外一種解決方案是更新該平台的 `hosts` 主機名稱/IP 解析檔案，從而將主機名稱解析為正確的 IP 位址。但是，使用 DHCP 重新連線時，可能會指定給您一個不同的 IP 位址。在這種情況下，您必須更新每個伺服器中的主機解析檔案。

啓動節點代理程式

節點代理程式必須先執行，才能管理伺服器實例。在節點代理程式常駐的系統中直接從本機執行 `asadmin` 指令 `start-node-agent`，以啓動節點代理程式。

如需有關指令語法的完整描述，請參閱該指令的線上說明。

例如：

```
asadmin start-node-agent --user admin --startinstances=false nodeagent1
```

其中，`admin` 為管理使用者，`nodeagent1` 為要啓動的節點代理程式。

依預設，在重新啓動節點代理程式時，節點代理程式實例的快取儲存庫不會與中央儲存庫同步化。若要強制實例的快取儲存庫與中央儲存庫同步化，請在 `asadmin start-node-agent` 指令中將 `--syncinstances` 選項設為 `true`。

備註 – 若將 `--syncinstances` 選項設為 `true`，則在重新啓動節點代理程式時，會同步化所有實例的儲存庫。

重新啓動節點代理程式後，請使用 `asadmin start-instance` 指令啓動伺服器實例。

停止節點代理程式

在節點代理程式常駐的系統上執行 `asadmin` 指令 `stop-node-agent`，以停止正在執行的節點代理程式。`stop-node-agent` 指令將停止該節點代理程式所管理的所有伺服器實例。

如需有關指令語法的完整描述，請參閱該指令的線上說明。

例如：

```
asadmin stop-node-agent nodeagent1
```

其中，`nodeagent1` 為節點代理程式的名稱。

刪除節點代理程式

刪除節點代理程式之前，必須先停止節點代理程式。您還可以刪除從未啟動過或者從未成功地與網域管理伺服器聯絡 (即尚未連結) 的節點代理程式。

若要刪除節點代理程式檔案，請在該節點代理程式常駐的系統中執行 `asadmin delete-node-agent`。

如需有關指令語法的完整描述，請參閱該指令的線上說明。

例如：

```
asadmin delete-node-agent nodeagent1
```

其中，`nodeagent1` 是節點代理程式。

刪除某個節點代理程式時，您還必須使用 Admin Console 或 `asadmin delete-node-agent-config` 指令從網域管理伺服器中刪除該節點代理程式的配置。

▼ 檢視一般節點代理程式資訊

- 1 在樹形元件中，選取 [節點代理程式] 節點。

- 2 按一下節點代理程式的名稱。

如果節點代理程式已存在但未在此處顯示，請使用 `asadmin start-node-agent` 在節點代理程式的主機電腦上啟動該節點代理程式。請參閱第 177 頁的「[啟動節點代理程式](#)」

- 3 檢查節點代理程式的主機名稱。

如果主機名稱為 [未知主機]，則該節點代理程式尚未同網域管理伺服器 (DAS) 進行初始聯絡。

- 4 檢查節點代理程式的狀態。

狀態可為：

- **正在執行：**節點代理程式已正確建立，並且目前處於執行狀態。
- **未執行：**已在本機機器上建立節點代理程式，但從未啟動，或節點代理程式啟動過但已停止。

- **等待會合：**節點代理程式是從未在本地機器中建立的預留位置。

請參閱第 175 頁的「[建立節點代理程式](#)」和第 177 頁的「[啟動節點代理程式](#)」。

5 選擇啟動節點代理程式時是否啟動實例。

選擇 [是]，將在啟動節點代理程式時自動啟動與該節點代理程式關聯的伺服器實例。

選擇 [否] 將手動啟動這些實例。

6 確定節點代理程式是否已同網域管理伺服器進行了聯絡。

如果節點代理程式從未與網域管理伺服器進行過聯絡，則表明該節點代理程式從未成功啟動過。

7 管理與節點代理程式關聯的伺服器實例。

如果節點代理程式正在執行，則可以透過按一下實例名稱旁邊的核取方塊並按一下 [啟動] 或 [停止] 來啟動或停止實例。

▼ 刪除節點代理程式配置

透過管理主控台，只能刪除網域中的節點代理程式配置，而不能刪除實際的節點代理程式。若要刪除節點代理程式自身，請在節點代理程式的本地機器上執行 `asadmin` 指令 `delete-node-agent`。如需更多資訊，請參閱第 178 頁的「[刪除節點代理程式](#)」。

刪除節點代理程式配置之前，該節點代理程式必須停止並且不能有任何關聯的實例。若要停止節點代理程式，請使用 `asadmin` 指令 `stop-node-agent`。請參閱第 177 頁的「[停止節點代理程式](#)」，以取得更多資訊。

- 1 在樹形元件中，選取 [節點代理程式] 節點。
- 2 在 [節點代理程式] 頁面中，選取要刪除的節點代理程式旁邊的核取方塊。
- 3 按一下 [刪除]。

更多資訊 等效的 `asadmin` 指令

```
delete-node-agent-config
```

▼ 編輯節點代理程式配置

- 1 在樹形元件中，展開 [節點代理程式] 節點。
- 2 選取要編輯的節點代理程式配置。

- 3 啟動代理程式後，在 [啟動] 上核取 [啟動實例] 以啟動該代理程式的伺服器實例。

您還可以從此頁面手動啟動和停止實例。

如果此配置是預留位置節點代理程式的配置，則使用 `asadmin create-node-agent` 建立實際的節點代理程式時，實際的節點代理程式將使用此配置。如需有關建立節點代理程式的資訊，請參閱第 175 頁的「[建立節點代理程式](#)」。

如果此配置是現有節點代理程式的配置，則將自動同步此節點代理程式配置資訊。

▼ 編輯節點代理程式範圍

您必須為連線至節點代理程式的使用者設定認證範圍。只有管理使用者才能存取節點代理程式。

- 1 在樹形元件中，展開 [節點代理程式] 節點。

- 2 選取要編輯的節點代理程式配置。

- 3 按一下 [認證範圍] 標籤。

- 4 在 [節點代理程式編輯範圍] 頁面中，輸入一個範圍。

預設範圍為 `admin-realm`，它是您在建立節點代理程式時建立的。若要使用其他範圍，請替代由網域控制的所有元件中的範圍，否則這些元件將無法正常通訊。

- 5 在 [類別名稱] 欄位中，指定實作該範圍的 Java 類別。

- 6 新增需要的所有特性。

認證範圍需要特定於供應商的特性，這些特性會因特定實作的需求不同而有所不同。

▼ 為 JMX 編輯節點代理程式的偵聽程式

節點代理程式使用 JMX 與網域管理伺服器通訊。因此，該節點代理程式必須具有偵聽 JMX 請求的連接埠以及其他偵聽程式資訊。

- 1 在樹形元件中，展開 [節點代理程式] 節點。

- 2 選取要編輯的節點代理程式配置。

- 3 按一下 [JMX] 標籤。

4 在 [位址] 欄位中，輸入 IP 位址或主機名稱。

如果偵聽程式使用唯一連接埠偵聽伺服器的所有 IP 位址，請輸入 `0.0.0.0`。否則，請輸入此伺服器的有效 IP 位址。

5 在 [連結埠] 欄位中，鍵入節點代理程式的 JMX 連接器要偵聽的連接埠。

如果 IP 位址為 `0.0.0.0`，則連接埠號必須是唯一的。

6 在 [JMX 協定] 欄位中，鍵入 JMX 連接器支援的協定。

預設值為 `rmi_jrmp`。

7 按一下 [接受所有位址] 旁邊的核取方塊以允許連線到所有 IP 位址。

節點代理程式將偵聽與網路卡相關聯的特定 IP 位址或偵聽所有 IP 位址。如果接受所有位址，則會將值 `0.0.0.0` 放入 [偵聽主機位址] 特性中。

8 在 [範圍名稱] 欄位中，鍵入為偵聽程式處理認證的範圍的名稱。

在此頁面的 [安全性] 區段中，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，請執行以下操作：

9 在 [安全性] 欄位中核取 [已啟用] 方塊。

依預設已啟用 [安全性]。

10 設定用戶端認證。

若需要用戶端在使用此偵聽程式時向伺服器認證自己的身份，請在 [用戶端認證] 欄位中核取 [已啟用] 方塊。

11 輸入憑證暱稱。

在 [憑證暱稱] 欄位中輸入現有伺服器金鑰組和憑證的名稱。

如需有關使用憑證及 SSL 的更多資訊，請參閱管理主控台線上說明。

12 在 [SSL3/TLS] 區段：

a. 核取要在偵聽程式上啟用的安全性協定。

必須核取 SSL3 和/或 TLS。

b. 選取協定所使用的密碼組。

若要啟用所有密碼組，請核取 [所有支援的密碼組]。

13 按一下 [儲存]。

配置高可用性階段作業持續性和容錯移轉

本章說明如何啓用和配置高可用性階段作業持續性。

- 第 183 頁的「階段作業持續性和容錯移轉簡介」
- 第 185 頁的「設定高可用性階段作業持續性」
- 第 188 頁的「HTTP 階段作業容錯移轉」
- 第 192 頁的「有狀態階段作業 Bean 容錯移轉」

階段作業持續性和容錯移轉簡介

Application Server 經由 HTTP 階段作業資料和有狀態階段作業 Bean (SFSB) 階段作業資料的**容錯移轉**功能，提供高可用性階段作業持續性。容錯移轉表示在伺服器實例或硬體出現故障時，其他伺服器實例接管分散式階段作業。

需求

在滿足以下條件時，分散式階段作業可在多個 Sun Java System Application Server 實例中執行：

- 每一個伺服器實例都能存取相同的階段作業狀態資料。Application Server 為 HTTP 階段作業及有狀態的階段作業 Bean 資料提供了下列類型的高可用性儲存裝置：
 - 叢集內其他伺服器上的記憶體中複製。預設會使用叢集設定檔啓用記憶體中複製。

如需使用記憶體中複製功能，需要啓用群組管理服務 (GMS)。如需有關 GMS 的更多資訊，請參閱第 147 頁的「群組管理服務」。

如果叢集中的各伺服器實例位於不同機器上，請確定符合下列必要條件：

- 爲了確保 GMS 和記憶體中複寫均能正確運作，各機器必須位於相同的子網路。
- 爲了確保記憶體中複製正確運作，必須儘可能使叢集中所有機器的系統時鐘同步。
- 高可用性資料庫 (HADB)。如需有關如何啓用此資料庫的資訊，請參閱 `configure-ha-cluster(1)`。

備註 – Sun Java System Application Server 的 Application Server 獨立發行版本 隨附了 HADB 軟體。如需有關 Sun Java System Application Server 的可用發行軟體資訊，請參閱「Sun Java System Application Server 9.1 Installation Guide」中的「Distribution Types and Their Components」。只有企業設定檔才提供 HADB 功能。如需有關設定檔的更多資訊，請參閱「Sun Java System Application Server 9.1 管理指南」中的「用法設定檔」。

- 每個伺服器實例均部署相同的可分散 Web 應用程式。`web.xml` 部署描述元檔案的 `web-app` 元素必須包含 `distributable` 元素。
- Web 應用程式使用高可用性階段作業持續性。如果將不可分散的 Web 應用程式配置爲使用高可用性階段作業持續性，伺服器會向記錄檔寫入錯誤。
- 必須使用 `deploy` 或 `deploydir` 指令 (`--availabilityenabled` 選項設定爲 `true`) 部署 Web 應用程式。如需有關這些指令的更多資訊，請參閱 `deploy(1)` 和 `deploydir(1)`。

限制

階段作業容錯移轉時，對開啓的檔案或網路連線的任何參照都會遺失。編碼應用程式時必須牢記此限制。

僅可以將某些物件連結至支援容錯移轉的分散式階段作業。與 Servlet 2.4 規格相反，如果不支援容錯移轉的物件類型連結至分散式階段作業，Sun Java System Application Server 不會丟出 `IllegalArgumentException`。

可以將以下物件連結至支援容錯移轉的分散式階段作業：

- 所有 EJB 元件的本地主機參照和物件參照。
- 並置的無狀態作業階段、有狀態作業階段或實體 Bean 參照。
- 分散式無狀態作業階段、有狀態作業階段或實體 Bean 參照。
- `InitialContext` 和 `java:comp/env` 的 JNDI 環境。
- `UserTransaction` 物件。但是，如果出現故障的實例從未重新啓動，則任何就緒的全域作業事件都將遺失，且可能無法正確回復或確定。
- 可序列化 Java 類型。

無法將以下物件類型連結至支援容錯移轉的階段作業：

- JDBC DataSource
- Java Message Service (JMS) ConnectionFactory 和 Destination 物件
- JavaMail™ 階段作業
- 連線工廠
- 受管理的物件
- Web 服務參照

通常，對於這些物件，容錯移轉不可用。但是，容錯移轉可能在某些情況下可用，例如，如果物件是可序列化的。

設定高可用性階段作業持續性

本小節說明如何設定高可用性階段作業持續性，包含以下主題：

- [第 185 頁的「設定高可用性階段作業持續性」](#)
- [第 186 頁的「啟用階段作業可用性」](#)

▼ 設定高可用性階段作業持續性

開始之前 高可用性階段作業持續性與動態部署、動態重新載入和自動部署不相容。這些功能適用於開發而非生產環境，因此必須在啟用 HA 階段作業持續性之前停用它們。如需如何停用這些功能的更多資訊，請參閱「Sun Java System Application Server 9.1 Application Deployment Guide」。

- 1 **建立 Application Server 叢集。**
如需更多資訊，請參閱[第 149 頁的「建立叢集」](#)。
- 2 **若是使用 HADB 來儲存階段作業狀態資料，請為叢集建立 HADB 資料庫。**

備註 – Sun Java System Application Server 的 Application Server 獨立發行版本 隨附了 HADB 軟體。如需有關 Sun Java System Application Server 的可用發行軟體資訊，請參閱「Sun Java System Application Server 9.1 Installation Guide」中的「Distribution Types and Their Components」。只有企業設定檔才提供 HADB 功能。如需有關設定檔的更多資訊，請參閱「Sun Java System Application Server 9.1 管理指南」中的「用法設定檔」。

若是針對階段作業狀態資料，在叢集內其他伺服器上使用記憶體中複製，請略過此步驟。

如需有關建立 HADB 資料庫的更多資訊，請參閱 `configure-ha-cluster(1)`。

- 3 **為叢集設定 HTTP 負載平衡。**
如需更多資訊，請參閱[第 120 頁的「設定 HTTP 負載平衡」](#)

4 啓用所需 Application Server 實例和 Web 或 EJB 容器的可用性。

然後配置階段作業持續性設定。選擇以下方法之一：

- 使用管理主控台。請參閱第 187 頁的「啓用伺服器實例的可用性」。
- 使用 `asadmin` 指令行公用程式。請參閱 `set(1)` 和 `configure-ha-persistence(1)`。

5 重新啓動叢集中的每個伺服器實例。

如果實例目前正在處理請求，請在重新啓動該實例前將其靜止，這樣，它就有足夠的時間來處理正在處理的請求。如需更多資訊，請參閱第 130 頁的「停用 (靜止) 伺服器實例或叢集」。

6 啓用所需的特定 SFSB 的可用性。

選取需要針對階段作業狀態進行檢查點操作的方法。請參閱第 195 頁的「配置個別 Bean 的可用性」。

7 如果要使每個 Web 模組均具有高可用性，請使每個 Web 模組均可分散。

8 在部署期間啓用個別應用程式、Web 模組或 EJB 模組的可用性。

請參閱第 194 頁的「配置個別應用程式或 EJB 模組的可用性」。

在管理主控台中，核取 [啓用可用性] 方塊，或使用 `--availabilityenabled` 選項設定為 `true` 的 `asadmin deploy` 指令。

啓用階段作業可用性

您可以啓用五個不同範圍 (最高至最低) 的階段作業可用性：

1. 伺服器實例，依預設啓用。啓用伺服器實例階段作業可用性，表示所有在伺服器實例上執行的應用程式皆可具有高可用性階段作業持續性。如需說明，請參閱下一小節第 187 頁的「啓用伺服器實例的可用性」。
2. 容器 (Web 或 EJB)，依預設啓用。如需有關在容器層級啓用可用性的資訊，請參閱：
 - 第 188 頁的「配置 Web 容器的可用性」
 - 第 193 頁的「配置 EJB 容器的可用性」
3. 應用程式，依預設停用。
4. 獨立 Web 或 EJB 模組，依預設停用。
5. 個別 SFSB，依預設停用。

若要在指定範圍啓用可用性，還必須在所有更高層級也啓用。例如，若要啓用應用程式層級的可用性，必須在伺服器實例層級和容器層級啓用可用性。

指定層級的預設值是其上一層級的設定值。例如，如果已啓用容器層級的可用性，則依預設啓用應用程式層級的可用性。

如果已停用伺服器實例層級的可用性，則啓用其他任何層級的可用性都不生效。如果已啓用伺服器實例層級的可用性，除非已明確停用，否則將啓用所有層級的可用性。

啓用伺服器實例的可用性

若要啓用伺服器實例的可用性，請使用 `asadmin set` 指令將配置的 `availability-service.availability-enabled` 特性設為 `true`。

例如，如果配置名稱為 `config1`：

```
asadmin set --user admin --passwordfile password.txt
--host localhost
--port 4849
config1.availability-service.availability-enabled="true"
```

▼ 使用管理主控台啓用伺服器實例的可用性

- 1 在樹形元件中，展開 [配置] 節點。
- 2 展開要編輯的配置之節點。
- 3 選取 [可用性服務] 節點。
- 4 在 [可用性服務] 頁面，核取 [可用性服務] 核取方塊來啓用實例層級的可用性。
若要停用可用性，請取消核取該方塊。

此外，如果為獲得階段作業持續性而變更了用於連線 HADB 的 JDBC 資源，可以變更儲存池名稱。如需詳細資訊，請參閱 `configure-ha-cluster(1)`。

備註 – Sun Java System Application Server 的 Application Server 獨立發行版本 隨附了 HADB 軟體。如需有關 Sun Java System Application Server 的可用發行軟體資訊，請參閱「Sun Java System Application Server 9.1 Installation Guide」中的「Distribution Types and Their Components」。只有企業設定檔才提供 HADB 功能。如需有關設定檔的更多資訊，請參閱「Sun Java System Application Server 9.1 管理指南」中的「用法設定檔」。

- 5 按一下 [儲存] 按鈕。
- 6 停止和重新啓動伺服器實例。

HTTP 階段作業容錯移轉

Java EE 應用程式通常具有大量階段作業狀態資料。Web 購物車即為階段作業狀態的經典範例。而且，應用程式可以快取頻繁需要的階段作業物件資料。實際上，幾乎所有需要進行大量使用者互動活動的應用程式均需要維護階段作業狀態。

配置 Web 容器的可用性

備註 – Sun Java System Application Server 的 Application Server 獨立發行版本 隨附了 HADB 軟體。如需有關 Sun Java System Application Server 的可用發行軟體資訊，請參閱「Sun Java System Application Server 9.1 Installation Guide」中的「Distribution Types and Their Components」。只有企業設定檔才提供 HADB 功能。如需有關設定檔的更多資訊，請參閱「Sun Java System Application Server 9.1 管理指南」中的「用法設定檔」。

若是使用 HADB，請使用 `asadmin configure-ha-persistence` 來啓用和配置 Web 容器可用性。如需有關此指令的更多資訊，請參閱 `configure-ha-persistence(1)`。

此外，還可使用 `asadmin set` 指令將配置的 `availability-service.web-container-availability.availability-enabled` 特性設為 `true`，然後使用 `configure-ha-persistence` 根據需要設定特性。

備註 – 若是使用記憶體中複製來儲存階段作業狀態資料，則**必須**使用 `asadmin set` 指令來啓用 Web 容器可用性並設定特性。`configure-ha-persistence` 指令**只能**用於 HADB。

例如，按如下所示使用 `set` 指令，其中 `config1` 為配置名稱：

```
asadmin set --user admin --passwordfile password.txt
--host localhost --port 4849
config1.availability-service.web-container-availability.availability-enabled="true"
asadmin configure-ha-persistence --user admin --passwordfile secret.txt
--type ha
--frequency web-method
--scope modified-session
--store jdbc/hastore
--property maxSessions=1000:reapIntervalSeconds=60 cluster1
```

▼ 使用管理主控台啓用 Web 容器的可用性

- 1 在樹形元件中，選取所需配置。
- 2 按一下 [可用性服務]。
- 3 選取 [Web 容器可用性] 標籤。
核取 [可用性服務] 核取方塊以啓用可用性。若要停用可用性，請取消核取該方塊。
- 4 變更其他設定，如以下第 189 頁的「可用性設定」小節中所述。
- 5 重新啟動伺服器實例。

可用性設定

備註 – Sun Java System Application Server 的 Application Server 獨立發行版本 隨附了 HADB 軟體。如需有關 Sun Java System Application Server 的可用發行軟體資訊，請參閱「Sun Java System Application Server 9.1 Installation Guide」中的「Distribution Types and Their Components」。只有 企業 設定檔才提供 HADB 功能。如需有關設定檔的更多資訊，請參閱「Sun Java System Application Server 9.1 管理指南」中的「用法設定檔」。

[可用性服務] 的 [Web 容器可用性] 標籤讓您可以變更以下可用性設定：

持續性類型：為啓用了可用性的 Web 應用程式指定階段作業持續性機制。允許的值有 memory (無持續性)、file (檔案系統)、replicated (其他伺服器的記憶體) 以及 ha (HADB)。

必須配置並啓用 HADB 後，才能使用 ha 階段作業持續性。如需配置詳細資訊，請參閱 configure-ha-cluster(1)。

若已啓用 Web 容器可用性，則預設的持續性類型將取決於設定檔，如下表所示。

設定檔	持續性類型
開發者	memory
叢集	replicated
企業	ha

對於需要階段作業持續性的生產環境，請使用 ha 或 replicated。memory 持續性類型和 file 持續性類型並未提供高可用性階段作業持續性。

如果停用 Web 容器可用性，則預設的持續性類型為 memory。

持續性頻率：指定儲存階段作業狀態的頻率。唯有當「持續性類型」為 `ha` 或 `replicated` 時才適用。允許的值包括：

- `web-method` - 在每個 Web 請求結束時、將回應傳送回用戶端之前，儲存階段作業狀態。此模式為發生故障時完全更新階段作業狀態提供了最好的保證。此為預設值。
- `time-based` - 根據 `reapIntervalSeconds` 儲存特性設定的頻率，在背景儲存階段作業狀態。此模式不能保證完全更新階段作業狀態。但是，它可以提供很大的效能改善，因為在每個請求之後都不儲存狀態。

持續性範圍：指定階段作業物件數和儲存階段作業狀態的頻率。唯有當「持續性類型」為 `ha` 或 `replicated` 時才適用。允許的值包括：

- `session` - 每次都儲存整個階段作業狀態。此模式為正確儲存任何可分散 Web 應用程式的階段作業資料提供了最好的保證。此為預設值。
- `modified-session` - 如果階段作業狀態已修改，則儲存整個階段作業狀態。如果呼叫了 `HttpSession.setAttribute()` 或 `HttpSession.removeAttribute()`，則系統將認為階段作業已修改。必須保證每次變更屬性時都呼叫 `setAttribute()`。這不是 Java EE 規格的需求，但是此模式需要這樣做才能正常運作。
- `modified-attribute` - 僅儲存修改的階段作業屬性。若要使此模式正常運作，您必須遵循一些指導原則：
 - 每次修改階段作業狀態時都必須呼叫 `setAttribute()`。
 - 確定各屬性之間沒有交叉參照。系統將對每個不同屬性關鍵字下的物件圖形單獨進行序列化並單獨儲存。如果每個單獨的關鍵字下的物件之間存在物件交叉參照，則它們將不會進行正確序列化和反序列化。
 - 在多個屬性之間分布階段作業狀態，或者至少在唯讀屬性和可修改屬性之間分布階段作業狀態。

單次登入狀態：核取此方塊以啟用單次登入狀態的持續性。若要停用可用性，請取消核取該方塊。如需更多資訊，請參閱第 191 頁的「[將階段作業容錯移轉與單次登入配合使用](#)」。

HTTP 階段作業儲存：如果為獲得階段作業持續性而變更了用於連線 HADB 的 JDBC 資源，可以變更 HTTP 階段作業儲存。如需詳細資訊，請參閱 `configure-ha-cluster(1)`。

配置個別 Web 應用程式的可用性

若要啟用和配置個別 Web 應用程式的可用性，請編輯應用程式部署描述元檔案 `sun-web.xml`。應用程式部署描述元中的設定會置換 Web 容器的可用性設定。

`session-manager` 元素的 `persistence-type` 屬性決定應用程式使用之階段作業持續性的類型。必須將該屬性設定為 `ha` 或 `replicated` 才能啟用高可用性階段作業持續性。

如需有關 `sun-web.xml` 檔案的更多資訊，請參閱「Sun Java System Application Server 9.1 Application Deployment Guide」中的「The `sun-web.xml` File」。

範例

```
<sun-web-app> ...
  <session-config>
    <session-manager persistence-type=ha>
      <manager-properties>
        <property name=persistenceFrequency value=web-method />
      </manager-properties>
      <store-properties>
        <property name=persistenceScope value=session />
      </store-properties>
    </session-manager> ...
  </session-config> ...
```

將階段作業容錯移轉與單次登入配合使用

在單一 Application Server 實例中，使用者經過應用程式認證後，即無需分別經在相同實例上執行的其他應用程式重新認證。這稱為**單次登入**。如需更多資訊，請參閱「Sun Java System Application Server 9.1 Developer's Guide」中的「User Authentication for Single Sign-on」。

若要使該功能在 HTTP 階段作業容錯移轉到叢集中另一個實例後仍然可用，必須將單次登入資訊保留在 HADB 中。若要保持單次登入資訊，首先請啟用伺服器實例和 Web 容器的可用性，然後啟用單次登入狀態容錯移轉。

備註 – Sun Java System Application Server 的 Application Server 獨立發行版本 隨附了 HADB 軟體。如需有關 Sun Java System Application Server 的可用發行軟體資訊，請參閱「Sun Java System Application Server 9.1 Installation Guide」中的「Distribution Types and Their Components」。只有企業設定檔才提供 HADB 功能。如需有關設定檔的更多資訊，請參閱「Sun Java System Application Server 9.1 管理指南」中的「用法設定檔」。

您可以用管理主控台在 [可用性服務] 的 [Web 容器可用性] 標籤中，啟用單次登入狀態容錯移轉，如第 188 頁的「[配置 Web 容器的可用性](#)」所述。您也可以使用 `asadmin set` 指令，將配置的

`availability-service.web-container-availability.sso-failover-enabled` 特性設為 `true`。

例如，按如下所示使用 `set` 指令，其中 `config1` 為配置名稱：

```
asadmin set --user admin --passwordfile password.txt
--host localhost --port 4849
config1.availability-service.web-container-availability.
sso-failover-enabled="true"
```


單次登入群組

可使用單一名稱和密碼組合存取的多個應用程式，即構成**單次登入群組**。對於與應用程式(是單次登入群組的一部分)相對應的 HTTP 階段作業，如果其中一個階段作業逾時，其他階段作業並不會失效，並且仍然可用。這是因為一個階段作業的逾時不應影響其他階段作業的可用性。

因此，如果一個階段作業逾時並且您嘗試從執行該階段作業的同一瀏覽器視窗存取相應的應用程式，則無需再次進行認證。但是將建立一個新的階段作業。

以屬於含有其他兩個應用程式的單次登入群組的購物車應用程式為例。假設其他兩個應用程式的階段作業逾時值大於購物車應用程式的階段作業逾時值。如果購物車應用程式的階段作業逾時，並且您嘗試從執行該階段作業的同一瀏覽器視窗執行購物車應用程式，則無需再次進行認證。但是，前一輛購物車將遺失，並且必須建立一輛新的購物車。即使執行購物車應用程式的階段作業已逾時，其他兩個應用程式也會繼續照常執行。

類似地，可假定與其他兩個應用程式中的任何一個應用程式對應的階段作業逾時。當從執行該階段作業的同一瀏覽器視窗連線應用程式時，您無需再次進行認證。

備註 – 此運作方式僅適用於階段作業逾時情況。如果啓用了單次登入，並且使用 `HttpSession.invalidate()` 令其中一個階段作業失效，則屬於單次登入群組的所有應用程式之階段作業都將失效。如果您嘗試存取屬於單次登入群組的任一應用程式，則需要再次進行認證，系統將為存取該應用程式的用戶端建立一個新的階段作業。

有狀態階段作業 Bean 容錯移轉

有狀態階段作業 Bean (SFSB) 包含用戶端特定的狀態。用戶端與有狀態階段作業 Bean 之間存在一對一關係。建立時，EJB 容器會為每個 SFSB 提供一個唯一的階段作業 ID，此 ID 將 SFSB 連結至用戶端。

在伺服器實例出現故障時，SFSB 的狀態可以儲存在永久存放區中。SFSB 的狀態在其使用週期中預先定義的時間儲存於持續性存放區中。此動作稱為**檢查點操作**。如果啓用，通常會在 Bean 完成了任何作業事件之後(即使該作業事件回復)執行檢查點操作。

但是，如果 SFSB 參與了 Bean 管理的作業事件，則可能會在 Bean 方法執行到一半時確定該作業事件。因為 Bean 的狀態可能由於方法呼叫而處於過渡過程中，因此不是對 Bean 狀態進行檢查點操作的恰當時間。這種情況下，如果 Bean 在方法結束時不在其他作業事件的範圍內，EJB 容器會在相應方法結束時對 Bean 的狀態進行檢查點操作。如果 Bean 管理的作業事件使用多個方法，檢查點操作將延遲，直至後續方法結束時沒有使用中的作業事件。

SFSB 的狀態無需是作業事件，且可能做為非作業事件商務方法而進行了重大修改。如果某個 SFSB 出現這種情況，則您可以指定一系列檢查點方法，如第 195 頁的「[指定進行檢查點操作方法](#)」中所述。

如果可分散 Web 應用程式參照 SFSB，且 Web 應用程式的階段作業容錯移轉，則 EJB 參照也將容錯移轉。

如果在 Application Server 實例停止時取消部署使用階段作業持續性的 SFSB，則可能不會清除持續性存放區中的階段作業資料。若要防止發生此情況，請在 Application Server 實例執行時取消部署 SFSB。

配置 EJB 容器的可用性

▼ 啓用 EJB 容器的可用性

- 1 選取 [EJB 容器可用性] 標籤。
- 2 核取 [可用性服務] 核取方塊。
若要停用可用性，請取消核取此核取方塊。
- 3 請按照第 194 頁的「可用性設定」中的說明變更其他設定
- 4 按一下 [儲存] 按鈕。
- 5 重新啓動伺服器實例。

更多資訊 等效的 asadmin 指令

若要啓用 EJB 容器的可用性，請使用 `asadmin set` 指令為配置設定以下三個特性：

- `availability-service.ejb-container-availability.availability-enabled`
- `availability-service.ejb-container-availability.sfsb-persistence-type`
- `availability-service.ejb-container-availability.sfsb-ha-persistence-type`

例如，如果 `config1` 為配置名稱，則使用以下指令：

```
asadmin set --user admin --passwordfile password.txt --host localhost --port 4849 config1.availability-service.ejb-container-availability
.availability-enabled="true"
```

```
asadmin set --user admin --passwordfile password.txt --host localhost --port 4849 config1.availability-service.ejb-container-availability
.sfsb-persistence-type="file"
```

```
asadmin set --user admin --passwordfile password.txt --host localhost --port 4849 config1.availability-service.ejb-container-availability
.sfsb-ha-persistence-type="ha"
```

可用性設定

備註 – Sun Java System Application Server 的 Application Server 獨立發行版本 隨附了 HADB 軟體。如需有關 Sun Java System Application Server 的可用發行軟體資訊，請參閱「Sun Java System Application Server 9.1 Installation Guide」中的「Distribution Types and Their Components」。只有企業設定檔才提供 HADB 功能。如需有關設定檔的更多資訊，請參閱「Sun Java System Application Server 9.1 管理指南」中的「用法設定檔」。

[可用性服務] 的 [EJB 容器可用性] 標籤可讓您變更以下可用性設定：

HA 持續性類型：為啓用了可用性的 SFSB 指定階段作業持續性和鈍化機制。允許值有 `file` (檔案系統)、`replicated` (其他伺服器上的記憶體) 及 `ha` (HADB)。預設值為 `ha`。對於需要階段作業持續性的生產環境，請使用預設值 `ha` 或**重複**。

SFSB 持續性類型：為尚未啓用可用性的 SFSB 指定鈍化機制。允許值有 `file` (預設值)、`replicated` 及 `ha`。

如果將任一持續性類型設定為 `file`，EJB 容器都將指定檔案系統位置，以儲存鈍化的階段作業 Bean 狀態。檔案系統的檢查點操作對於測試很有用，但不適用於生產環境。如需配置儲存特性的相關資訊，請參閱管理主控台線上說明。

如果任何其他伺服器實例出現故障，HA 持續性會使伺服器實例的叢集可以回復 SFSB 狀態。HADB 也用做鈍化和啓動儲存。在需要 SFSB 狀態持續性的生產環境中使用此選項。如需更多資訊，請參閱 `configure-ha-cluster(1)`。

SFSB 儲存池名稱：如果為獲得階段作業持續性而變更了用於連線 HADB 的 JDBC 資源，可以變更 SFSB 儲存池名稱。如需詳細資訊，請參閱 `configure-ha-cluster(1)`。

停用可用性時配置 SFSB 階段作業儲存

如果停用了可用性，本機檔案系統將用於 SFSB 狀態鈍化，但不用於持續性。若要變更 SFSB 狀態的儲存位置，請變更 EJB 容器中的 [階段作業儲存位置] 設定。如需配置儲存特性的更多資訊，請參閱管理主控台線上說明。

配置個別應用程式或 EJB 模組的可用性

部署期間，可以啓用個別應用程式或 EJB 模組的 SFSB 可用性：

- 如果使用管理主控台部署，請核取 [啓用可用性] 核取方塊。
- 如果使用 `asadmin deploy` 或 `asadmin deploydir` 指令部署，請將 `--availabilityenabled` 選項設定為 `true`。如需更多資訊，請參閱 `deploy(1)` 和 `deploydir(1)`。

配置個別 Bean 的可用性

若要啟用個別 SFSB 的可用性並選取進行檢查點操作方法，請使用 `sun-ejb-jar.xml` 部署。

若要啟用高可用性階段作業持續性，請在 `ejb` 元素中設定 `availability-enabled="true"`。若要控制 SFSB 快取的大小和運作方式，請使用以下元素：

- `max-cache-size`：指定快取中保留的階段作業 Bean 的最大數。如果快取溢出 (Bean 的個數超出 `max-cache-size`)，則容器會鈍化一些 Bean 或將 Bean 的序列化狀態寫入檔案。用於建立檔案的目錄來自使用配置 API 的 EJB 容器。
- `resize-quantity`
- `cache-idle-timeout-in-seconds`
- `removal-timeout-in-seconds`
- `victim-selection-policy`

如需有關 `sun-ejb-jar.xml` 的更多資訊，請參閱「Sun Java System Application Server 9.1 Application Deployment Guide」中的「The sun-ejb-jar.xml File」。

範例 9-1 啟用了可用性的 EJB 部署描述元範例

```
<sun-ejb-jar>
...
  <enterprise-beans>
    ...
    <ejb availability-enabled="true">
      <ejb-name>MySFSB</ejb-name>
    </ejb>
    ...
  </enterprise-beans>
</sun-ejb-jar>
```

指定進行檢查點操作方法

如果啟用，通常會在 Bean 完成了任何作業事件之後 (即使該作業事件回復) 執行檢查點操作。若要在導致 Bean 狀態重大修改之非作業事件商務方法結束時，指定額外的選擇性 SFSB 檢查點操作功能，請使用 `sun-ejb-jar.xml` 部署描述元檔案的 `ejb` 元素中的 `checkpoint-at-end-of-method` 元素。

如果要在建立之後立即對 SFSB 的初始狀態進行檢查點操作，`checkpoint-at-end-of-method` 元素中的非作業事件方法可以是：

- 在 SFSB 的主介面中定義的 `create()` 方法

- 對於僅使用容器管理的作業事件之 SFSB，Bean 遠端介面中的方法使用作業事件屬性 TX_NOT_SUPPORTED 或 TX_NEVER 標記
- 對於僅使用 Bean 管理的作業事件之 SFSB，即不啟動也不確定 Bean 管理作業事件的方法

忽略此清單中提及的任何其他方法。在這些方法呼叫的每一個結尾，EJB 容器均將 SFSB 的狀態儲存至持續性存放區。

備註 – 如果 SFSB 未參與任何作業事件，且如果 `checkpoint-at-end-of-method` 元素中未明確指定其任何方法，則根本不會對 Bean 的狀態進行檢查點操作 (即使此 Bean 的 `availability-enabled="true"`)

爲了取得更好的效能，請指定方法的小子集。此方法應該完成對 Bean 的狀態做出重要修改的大量工作或結果。

範例 9-2 指定方法檢查點之 EJB 部署描述元範例

```
<sun-ejb-jar>
...
  <enterprise-beans>
    ...
    <ejb availability-enabled="true">
      <ejb-name>ShoppingCartEJB</ejb-name>
      <checkpoint-at-end-of-method>
        <method>
          <method-name>addToCart</method-name>
        </method>
      </checkpoint-at-end-of-method>
    </ejb>
    ...
  </enterprise-beans>
</sun-ejb-jar>
```

Java 訊息服務的負載平衡和容錯移轉

本章說明如何配置 Java 訊息服務 (JMS) 的負載平衡和容錯移轉，以與 Application Server 配合使用。其中包含以下主題：

- 第 197 頁的「Java 訊息服務簡介」
- 第 198 頁的「配置 Java 訊息服務」
- 第 200 頁的「連線池儲存和容錯移轉」
- 第 203 頁的「將 MQ 叢集與應用程式伺服器配合使用」

Java 訊息服務簡介

Java 訊息服務 (JMS) API 是一種允許 Java EE 應用程式和元件建立、傳送、接收以及讀取郵件的郵件傳送標準。它可啟用可靠的非同步鬆耦合分散式通訊。實作 JMS 的 Sun Java System Message Queue (MQ)，與 Application Server 緊密整合在一起，可讓您建立諸如訊息驅動 Bean (MDB) 之類的元件。

MQ 透過**連接器模組** (也稱為資源配接卡，由 Java EE 連接器架構規格 1.5 版定義) 與 Application Server 整合在一起。部署至 Application Server 的 Java EE 元件則需使用透過連接器模組整合的 JMS 提供者交換 JMS 訊息。在 Application Server 中建立 JMS 資源會在背景中建立連接器資源。因此，每個 JMS 作業均會在背景中呼叫連接器執行階段並使用 MQ 資源配接卡。

您可以透過管理主控台或 `asadmin` 指令行公用程式來管理 Java 訊息服務。

詳細資訊

如需有關配置 JMS 資源的更多資訊，請參閱「Sun Java System Application Server 9.1 管理指南」中的第 4 章「配置 Java 訊息服務資源」。如需有關 JMS 的更多資訊，請參閱「Sun Java System Application Server 9.1 Developer's Guide」中的第 18 章「Using the

Java Message Service」。如需有關連接器 (資源配接卡) 的更多資訊，請參閱「Sun Java System Application Server 9.1 Developer's Guide」中的第 12 章「Developing Connectors」。

如需有關 Sun Java System Message Queue 的更多資訊，請參閱 [Message Queue 文件 \(http://docs.sun.com/coll/1343.4\)](http://docs.sun.com/coll/1343.4)。如需有關 JMS API 的一般資訊，請參閱 [JMS 網頁 \(http://java.sun.com/products/jms/index.html\)](http://java.sun.com/products/jms/index.html)

配置 Java 訊息服務

Java 訊息服務配置可用於與 Sun Java System Application Server 叢集或實例的所有內收和外送連線。您可以使用以下方法配置 Java 訊息服務：

- 管理主控台。在相關配置下開啓 Java 訊息服務元件。如需更多資訊，請參閱「Sun Java System Application Server 9.1 管理指南」中的第 4 章「配置 Java 訊息服務資源」。
- `asadmin set` 指令。您可以設定以下屬性：

```
server.jms-service.init-timeout-in-seconds = 60
server.jms-service.type = LOCAL
server.jms-service.start-args =
server.jms-service.default-jms-host = default_JMS_host
server.jms-service.reconnect-interval-in-seconds = 60
server.jms-service.reconnect-attempts = 3
server.jms-service.reconnect-enabled = true
server.jms-service.addresslist-behavior = random
server.jms-service.addresslist-iterations = 3
server.jms-service.mq-scheme = mq
server.jms-service.mq-service = jms
```

您也可設定以下特性：

```
server.jms-service.property.instance-name = imqbroker
server.jms-service.property.instance-name-suffix =
server.jms-service.property.append-version = false
```

使用 `asadmin get` 指令可列出所有的 Java 訊息服務屬性和特性。如需有關 `asadmin get` 的更多資訊，請參閱 `get(1)`。如需有關 `asadmin set` 的更多資訊，請參閱 `set(1)`。

您可以使用 JMS 連線工廠設定置換 Java 訊息服務配置。如需詳細資訊，請參閱「Sun Java System Application Server 9.1 管理指南」中的「JMS 連線工廠」。

備註 – 變更 Java 訊息服務的配置後，必須重新啟動 Application Server 實例。

如需有關 JMS 管理的更多資訊，請參閱「Sun Java System Application Server 9.1 管理指南」中的第 4 章「配置 Java 訊息服務資源」

Java 訊息服務整合

MQ 可透過以下三種方式與 Application Server 進行整合：LOCAL、REMOTE 及 EMBEDDED。這些模式在管理主控台中由 Java 訊息服務 Type 屬性表示。

LOCAL Java 訊息服務

當 Type 屬性為 LOCAL (叢集實例的預設值) 時，Application Server 將啟動和停止被指定為預設 JMS 主機的 MQ 代理程式。MQ 程序是跨處理序啟動，與 Application Server 程序位於不同的 VM。Application Server 為代理程式提供了額外的連接埠。代理程式會使用此連接埠來啟動 RMI 登錄。此連接埠號將等於該實例已配置的 JMS 連接埠號加 100。例如，如果 JMS 連接埠號為 37676，則此額外的連接埠號為 37776。

若要在 Application Server 實例和 Message Queue 代理程式之間建立一對一的關係，請將類型設定為 LOCAL 並為每個 Application Server 實例指定不同的預設 JMS 主機。無論叢集是否在 Application Server 或 MQ 中定義，您均可執行此作業。

對於 LOCAL 類型，請使用 Start Arguments 屬性指定 MQ 代理程式的啟動參數。

REMOTE Java 訊息服務

當 Type 屬性為 REMOTE 時，必須單獨啟動 MQ 代理程式。如需有關啟動代理程式的資訊，請參閱「Sun Java System Message Queue Administration Guide」。

在此情況下，Application Server 將使用外部配置的代理程式或代理程式叢集。此外，您必須從 Application Server 分別啟動和停止 MQ 代理程式，並使用 MQ 工具配置和調校代理程式或代理程式叢集。REMOTE 類型最適合於 Application Server 叢集。

對於 REMOTE 類型，您必須使用 MQ 工具指定 MQ 代理程式啟動參數。可忽略 Start Arguments 屬性。

EMBEDDED Java 訊息服務

如果 JMS Type 屬性為 EMBEDDED，則表示 Application Server 和 JMS 代理程式並置於相同的 VM 中，且將在同處理序啟動 JMS 服務，並由 Application Server 管理。在此模式中的 JMS 作業會規避網路堆疊，以使效能最佳化。

JMS 主機清單

JMS 主機代表 MQ 代理程式。Java 訊息服務包含 **JMS 主機清單** (也稱為 `AddressList`)，該清單包含 Application Server 使用的所有 JMS 主機。

JMS 主機清單中包含指定 MQ 代理程式的主機和連接埠，並且每當 JMS 主機配置變更時，此清單均會更新。在建立 JMS 資源或部署 MDB 時，它們會繼承 JMS 主機清單。

備註 – 在 Sun Java System Message Queue 軟體中，`AddressList` 特性稱為 `imqAddressList`。

預設 JMS 主機

JMS 主機清單中有一台主機被指定為預設 JMS 主機，名為 `Default_JMS_host`。將 Java 訊息服務類型配置為 `LOCAL` 後，Application Server 實例會啟動預設的 JMS 主機。

如果您已在 Sun Java System Message Queue 軟體中建立了具有多個代理程式的叢集，請刪除預設 JMS 主機，然後增加 Message Queue 叢集的代理程式，並將其做為 JMS 主機。在此情況下，預設的 JMS 主機便成為 JMS 主機清單中的第一台主機。

當 Application Server 使用 Message Queue 叢集時，它會在預設 JMS 主機上執行 Message Queue 特定的指令。例如，為具有三個代理程式的 Message Queue 叢集建立實體目標後，雖然用於建立實體目標的指令在預設的 JMS 主機上執行，但該實體目標可由叢集中的所有三個代理程式使用。

建立 JMS 主機

您可以使用以下方法建立其他 JMS 主機：

- 使用管理主控台。開啓相關配置下的 [Java 訊息服務] 元件，選取 [JMS 主機] 元件，然後按一下 [新建]。如需更多資訊，請參閱管理主控台線上說明。
- 使用 `asadmin create-jms-host` 指令。如需詳細資訊，請參閱 `create-jms-host(1)`。每當 JMS 主機配置發生變更時均會更新 JMS 主機清單。

連線池儲存和容錯移轉

Application Server 支援 JMS 連線池儲存和容錯移轉。Sun Java System Application Server 可自動地將 JMS 連線儲存到連線池。當 `Address List Behavior` 屬性為 `random` (預設) 時，Application Server 會從 JMS 主機清單中隨機選取其主代理程式。發生容錯移轉時，MQ 不需設定即可將負載傳輸至其他代理程式並可保持 JMS 語義不變。如果 JMS 的類型為 `LOCAL`，則 `Address List Behavior` 屬性的預設值為 `priority`。

若要指定在連線中斷後，Application Server 是否嘗試重新連線至主代理程式，請選取 [重新連線] 核取方塊。啓用後，如果主代理程式中斷，Application Server 會嘗試重新連線至 JMS 主機清單中的其他代理程式。

啓用 [重新連線] 後，還需指定以下屬性：

- **位址清單運作方式**：連線嘗試是依循 JMS 主機清單中的位址順序 (priority) 還是依循隨機順序 (random)。如果設定為 Priority，則 Java 訊息服務會嘗試連線至 JMS 主機清單中指定的第一個 MQ 代理程式，並且，當第一個代理程式不可用時，會使用其他代理程式。如果設定為 Random，則 Java 訊息服務會從 JMS 主機清單中隨機選取 MQ 代理程式。如果許多用戶端嘗試使用同一個連線工廠來進行連線，請使用此設定以防止它們全部嘗試連線至同一個位址。
- **Address List Iterations**：Java 訊息服務透過 JMS 主機清單反覆運算，以建立 (或重新建立) 連線的次數。值 -1 表示嘗試次數沒有限制。
- **重新連線嘗試**：用戶端執行階段嘗試連線 (或重新連線) JMS 主機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到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)。
- **重新連線間隔**：兩次重新連線嘗試之間的時間隔秒數。此屬性適用於對 JMS 主機清單中每個位址的連線嘗試，及對該清單中連續位址的連線嘗試。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。

您可以使用 JMS 連線工廠設定置換這些設定。如需詳細資訊，請參閱「Sun Java System Application Server 9.1 管理指南」中的「JMS 連線工廠」。

負載平衡的訊息流入

您可以使用 `activation-config-property` 元素，在 `sun-ejb-jar.xml` 檔案中為訊息驅動 Bean 配置 `jsra` 資源配接卡的 `ActivationSpec` 特性。每當部署訊息驅動 Bean (EndPointFactory) 時，連接器執行階段引擎會尋找這些特性，接著配置在資源配接卡中。請參閱「Sun Java System Application Server 9.1 Application Deployment Guide」中的「`activation-config-property`」。

Application Server 不需設定即可將訊息隨機傳送到具有相同 ClientID 的訊息驅動 Bean。長期用戶需要 ClientID。

對於其中未配置 ClientID 的非長期訂閱者，訂閱相同主題的特定訊息驅動 Bean 之所有實例均視為相等。將訊息驅動 Bean 部署到 Application Server 的多個實例時，只有其中一個訊息驅動 Bean 會收到訊息。如果有多個不同的訊息驅動 Bean 訂閱相同的主題，則每個訊息驅動 Bean 的一個實例會接收到訊息的副本。

若要支援使用相同佇列的多個用戶，請將實體目標的 `maxNumActiveConsumers` 特性設定為較大的值。如果已設定此特性，Sun Java System Message Queue 軟體最多會允許該數量的訊息驅動 Bean 使用相同佇列的訊息。會將訊息隨機傳遞到訊息驅動 Bean。如果將 `maxNumActiveConsumers` 設定為 -1，則對用戶數目沒有限制。

若要確保優先使用本機傳遞，請將 `addresslist-behavior` 設定為 `Priority`。此設定指定先選取 `AddressList` 中的第一個代理程式。第一個代理程式是本機並置的 `Message Queue` 實例。如果無法使用此代理程式，則會依 `AddressList` 中列出的代理程式順序嘗試連線。此設定是屬於叢集的 `Application Server` 實例的預設值。

備註 – 開發者設定檔中無法使用叢集功能。如需有關設定檔的資訊，請參閱「Sun Java System Application Server 9.1 管理指南」中的「用法設定檔」。

JMS 服務高可用性

JMS 元件有兩層可用性：

- **服務可用性** – 此層級的重點在於 JMS 服務的可用性，即使有一段時間無法使用訊息也無關緊要。只要將連線容錯移轉到提供服務的可用新實例，JMS 元件就會將服務視為可用且會正常運作。「Sun Java System Application Server 9.1 Developer's Guide」中的「Connection Failover」說明了此可用性層級。
- **資料可用性** – 此層級的服務和持續性訊息可用性都不可或缺。此層級也處理一次 (且只有一次) 傳送及訊息排序作業的 JMS 語義。

您可以在包含 Java 訊息服務 (JMS) 的 Sun Java System Message Queue 叢集中啟用資料可用性。訊息會保留在共用持續性存放區中，並可從叢集中的其他所有代理程式實例或高可用性資料庫 (HADB) (但必須已安裝，並已選取企業設定檔) 取得。如需有關設定檔的資訊，請參閱「Sun Java System Application Server 9.1 管理指南」中的「用法設定檔」。您必須先啟用 `Application Server` 實例的可用性，才能為相應的代理程式啟用資料可用性。

備註 – 個別的應用程式和模組無法控制或覆寫 JMS 可用性。

若要啟用資料可用性，請在管理主控台中選取相關配置下的「可用性服務」元件。核取 [可用性服務] 核取方塊。若要為 JMS 服務啟用可用性，請選取 [JMS 可用性] 標籤，然後核取 [可用性服務] 方塊。`Application Server` 叢集中的所有實例應該具有相同的實例可用性和 JMS 可用性設定，才能確保運作方式一致。如需詳細資訊，請參閱「Sun Java System Application Server 9.1 高可用性管理指南」。

備註 – 開發者設定檔中無法使用叢集功能。如需有關設定檔的資訊，請參閱「Sun Java System Application Server 9.1 管理指南」中的「用法設定檔」。

將 MQ 叢集與應用程式伺服器配合使用

MQ Enterprise Edition 支援多個互連的代理程式實例 (稱為**代理程式叢集**)。對於代理程式叢集，用戶端連線會分散至叢集中的所有代理程式。叢集可提供水平可延伸性並可提高可用性。

本小節說明如何配置 Application Server 以使用具有高可用性的 Sun Java System Message Queue 叢集。其中說明了如何啟動與配置 Message Queue 叢集。

如需有關 Application Server 拓樸和 MQ 部署的更多資訊，請參閱「Sun Java System Application Server 9.1 部署規劃指南」中的「Planning Message Queue Broker Deployment」。

高度可用的 MQ 叢集

Sun Java System Message Queue 4.1 透過新的「高度可用」叢集類型，提供高度可用的訊息服務。在此類型的 MQ 叢集中，所有代理程式實例都會共享對等式關係，且其所有代理程式實例都使用相同的持續性資料存放區。各實例會自動偵測是否有故障的實例，並透過動態接管選擇的方式，接管故障代理程式的持續性訊息。因此 Application Server 中部署的應用程式元件可利用這些可用性功能。

此叢集類型可讓佇列或長期主題訂閱不會遺失任何處理的持續性訊息。當用戶端執行階段所連線的代理程式無法使用時，就可能會遺失傳送至非長期訂閱者的非持續性訊息或持續性訊息。

在本機模式中配置高度可用的代理程式叢集

1. 啟動 HADB。
2. 建立並啟動 Application Server 網域。若要建立並啟動網域，請分別使用 `asadmin` 指令 `create-domain` 和 `start-domain`。如需這些指令的更多資訊，請參閱 `create-domain(1)` 和 `start-domain(1)`。
3. 建立並啟動節點代理程式。若要建立並啟動節點代理程式，請分別使用 `asadmin` 指令 `create-node-agent` 和 `start-node-agent`。如需這些指令的更多資訊，請參閱 `create-node-agent(1)` 和 `start-node-agent(1)`。
4. 建立叢集。您可以使用 `asadmin` 指令 `create-cluster` 或管理主控台來建立叢集。如需有關 `create-cluster` 指令的更多資訊，請參閱 `create-cluster(1)`。如需如何使用管理主控台建立叢集的更多資訊，請參閱管理主控台線上說明。
5. 在叢集中建立實例。建立實例時，請指定遠端代理程式所使用的 JMS 提供者連接埠號。如果未指定，則使用預設的 JMS 提供者連接埠號。您可以使用管理主控台或 `asadmin` 指令 `create-instance` 來建立實例。如需如何使用管理主控台建立實例的更多資訊，請參閱管理主控台線上說明。如需有關 `create-instance` 指令的資訊，請參閱 `create-instance(1)`。

6. 啟動叢集。您可以從管理主控台或使用 `asadmin` 指令 `start-cluster` 執行此動作。如需有關如何使用管理主控台啟動叢集的更多資訊，請參閱管理主控台線上說明。如需有關 `start-cluster` 指令的資訊，請參閱 `start-cluster(1)`。
7. 使用 `asadmin` 指令 `configure-ha-cluster` 配置 HA 叢集。如需有關此指令的資訊，請參閱 `configure-ha-cluster(1)`。

在遠端模式中配置高度可用的代理程式叢集

1. 啟動 HADB。
2. 建立資料庫表格。
3. 若是建立高度可用的代理程式叢集，請複製 HA 驅動程式。

```
cp $AS_HOME/hadb/4.4.3-6/lib/hadbjdbc4.jar $SIAS_HOME/imq/lib/ext
```

4. 建立並啟動網域。若要執行此動作，請使用指令 `asadmin` 指令 `create-domain` 和 `start-domain`。如需這些指令的更多資訊，請參閱 `create-domain(1)` 和 `start-domain(1)`。
5. 建立並啟動節點代理程式。若要執行此動作，請使用 `asadmin` 指令 `create-domain` 和 `start-node-agent`。如需這些指令的更多資訊，請參閱 `create-node-agent(1)` 和 `start-node-agent(1)`。
6. 建立叢集。您可以使用 `asadmin` 指令 (`create-cluster`) 或管理主控台來建立叢集。如需更多資訊，請參閱 `create-cluster(1)`。如需如何使用管理主控台建立叢集的更多資訊，請參閱管理主控台線上說明。
7. 在叢集中建立實例。建立實例時，請指定遠端代理程式所使用的 JMS 提供者連接埠號。如果未指定，則使用預設的 JMS 提供者連接埠號。
8. 刪除預設的 JMS 主機，並建立實例可連線的 JMS 主機。請確定將每一個代理程式新增為獨立的 JMS 主機。如需有關 JMS 主機的更多資訊，請參閱 [第 200 頁的「JMS 主機清單」](#)。
9. 將 JMS 類型設定為 Remote。您可以使用 `asadmin` 指令 `set` 或從管理主控台的 [JMS 服務] 頁面執行此動作。
10. 若是配置高度可用的代理程式，請將 [JMS 可用性] 設定為 `true`。您可以使用 `asadmin` 指令 `set` 或從管理主控台的 [JMS 可用性] 頁面執行此動作。
11. 啟動代理程式實例。
12. 啟動叢集。如需更多資訊，請參閱 `start-cluster(1)`。

非 HA 叢集的自動叢集功能

以前管理員必須依照本節的下列程序分別設定「非高度可用的」MQ 叢集 (具有主代理程式的 MQ 叢集)。在此發行版本中，除了 MQ 叢集的手動設定程序 (屬於類型 REMOTE) 外，Application Server 也提供了「自動叢集功能」，這意味著當使用者建立

Application Server 叢集時，將自動建立並置的非 HA 叢集 (屬於類型 LOCAL)。這將是建立 MQ 叢集的預設模式。例如，當管理員建立一個內含三個 Application Server 實例的 Application Server 叢集時，會將每個 Application Server 實例配置成使用一個並置的代理程式，因此不需設定即可使三個 MQ 代理程式實例形成一個 MQ 叢集。第一個 Application Server 實例的 MQ 代理程式會被設定為主代理程式。不過自動叢集功能也有一個缺點。如果管理員將實例新增到叢集，自動建立的 MQ 代理程式實例將無法加入到叢集中。此運作方式也適用於從叢集移除實例的情況。

▼ 以 Application Server 叢集啓用 MQ 叢集

開始之前 如果叢集的類型是 REMOTE，請執行下列步驟。如果叢集的類型是 LOCAL，則步驟 1 到步驟 4 不適用。

- 1 如果其中之一尚不存在，則建立 Application Server 叢集。
如需有關建立叢集的資訊，請參閱第 149 頁的「建立叢集」。

- 2 建立 MQ 代理程式叢集。

首先，刪除參照由網域管理伺服器啟動的代理程式的預設 JMS 主機，然後，建立三個要包含在 MQ 代理程式叢集中的外部代理程式 (JMS 主機)。

使用管理主控台或 `asadmin` 指令行公用程式建立 JMS 主機。

若要使用 `asadmin`，請參照以下的指令範例：

```
asadmin delete-jms-host --target cluster1 default_JMS_host
asadmin create-jms-host --target cluster1
    --mqhost myhost1 --mqport 6769
    --mquser admin --mqpassword admin broker1
asadmin create-jms-host --target cluster1
    --mqhost myhost2 --mqport 6770
    --mquser admin --mqpassword admin broker2
asadmin create-jms-host --target cluster1
    --mqhost myhost3 --mqport 6771
    --mquser admin --mqpassword admin broker3
```

若要使用管理主控台建立主機，請執行以下步驟：

- a. 導覽至 [JMS 主機] 節點 ([配置] > [config-name] > [Java 訊息服務] > [JMS 主機])。
- b. 刪除預設代理程式 (default_JMS_host)。
選取該代理程式旁邊的核取方塊，然後按一下 [刪除]。
- c. 按一下 [新建] 以建立每個 JMS 主機並輸入其特性值。
填入主機名稱、DNS 名稱或 IP 位址、連接埠號碼、管理使用者名稱和密碼的值。

3 啟動主 MQ 代理程式和其他 MQ 代理程式。

除了在 JMS 主機電腦上啟動三個外部代理程式以外，在任一機器上啟動一個主代理程式。此主代理程式不必為代理程式叢集的一部分。例如：

```
/usr/bin/imqbrokerd -tty -name brokerM -port 6772  
-cluster myhost1:6769,myhost2:6770,myhost2:6772,myhost3:6771  
-D"imq.cluster.masterbroker=myhost2:6772"
```

4 啟動該叢集中的 Application Server 實例。

5 在叢集上建立 JMS 資源：

a. 建立 JMS 實體目標。

例如，使用 asadmin：

```
asadmin create-jmsdest --desttype queue --target cluster1 MyQueue  
asadmin create-jmsdest --desttype queue --target cluster1 MyQueue1
```

若要使用管理主控台，請執行以下步驟：

i. 導覽至 [JMS 主機] 頁面 ([配置] > [config-name] > [Java 訊息服務] > [實體目標])。

ii. 按一下 [新建] 以建立每個 JMS 實體目標。

iii. 對於每個目標，輸入其名稱和類型 (佇列)。

b. 建立 JMS 連線工廠。

例如，使用 asadmin：

```
asadmin create-jms-resource --target cluster1  
--restype javax.jms.QueueConnectionFactory jms/MyQcf  
asadmin create-jms-resource --target cluster1  
--restype javax.jms.QueueConnectionFactory jms/MyQcf1
```

若要使用管理主控台，請執行以下步驟：

i. 導覽至 [JMS 連線工廠] 頁面 ([資源] > [JMS 資源] > [連線工廠])。

ii. 若要建立每個連線工廠，按一下 [新建]。

將開啓 [建立 JMS 連線工廠] 頁面。

iii. 對於每個連線工廠，輸入 JNDI 名稱 (例如 jms/MyQcf) 和類型
javax.jms.QueueConnectionFactory

iv. 從該頁面底部可用目標的清單中選取叢集，然後按一下 [增加]。

v. 按一下 [確定] 以建立連線工廠。

c. 建立 JMS 目標資源。

例如，使用 asadmin：

```
asadmin create-jms-resource --target cluster1
--restype javax.jms.Queue
--property imqDestinationName=MyQueue jms/MyQueue
asadmin create-jms-resource --target cluster1
--restype javax.jms.Queue
--property imqDestinationName=MyQueue1 jms/MyQueue1
```

若要使用管理主控台，請執行以下步驟：

- i. 導覽至 [JMS 目標資源] 頁面 ([資源] > [JMS 資源] > [連線工廠])。
- ii. 若要建立每個目標資源，按一下 [新建]。
將開啓 [建立 JMS 目標資源] 頁面。
- iii. 對於每個目標資源，輸入 JNDI 名稱 (例如 jms/MyQueue) 和類型 javax.jms.Queue。
- iv. 從該頁面底部可用目標的清單中選取叢集，然後按一下 [增加]。
- v. 按一下 [確定] 以建立目標資源。

6 使用 `--retrieve` 選項為應用程式用戶端部署應用程式。例如：

```
asadmin deploy --target cluster1
--retrieve /opt/work/MQapp/mdb-simple3.ear
```

7 存取應用程式並對其進行測試，以確保其正常運作。

8 若要將 Application Server 復原為其預設 JMS 配置，請刪除您建立的 JMS 主機，然後重新建立預設。例如：

```
asadmin delete-jms-host --target cluster1 broker1
asadmin delete-jms-host --target cluster1 broker2
asadmin delete-jms-host --target cluster1 broker3
asadmin create-jms-host --target cluster1
--mqhost myhost1 --mqport 7676
--mquser admin --mqpassword admin
default_JMS_host
```

您也可使用管理主控台執行等效作業。

疑難排解 如果遇到問題，請考量以下作業：

- 檢視 `as-install-dir/nodeagents/node-agent-name/instance-name/logs/server.log` 上的 Application Server 記錄檔。如果您在記錄檔中看到 MQ 代理程式未回應訊息，請停止該代理程式，然後將其重新啓動。

- 檢視 `as-install-dir/nodeagents/node-agent-name/instance-name/imq/imq-instance-name/log/log.txt` 中提供的代理程式記錄。
- 對於 Remote JMS 類型，請務必先啟動 MQ 代理程式，再啟動 Application Server 實例。
- 當所有 MQ 代理程式均中斷後，使用 Java 訊息服務中的預設值中斷或啟動 Application Server 需花費 30 分鐘。調校 Java 訊息服務值可取得此逾時可接受的值。例如：

```
asadmin set --user admin --password administrator  
cluster1.jms-service.reconnect-interval-in-seconds=5
```


RMI-IIOP 負載平衡和容錯移轉

本章說明如何將 Sun Java System Application Server 的高可用性功能透過 RMI-IIOP 用於遠端 EJB 參照和 JNDI 物件。

- [第 209 頁的「簡介」](#)
- [第 211 頁的「設定 RMI-IIOP 負載平衡和容錯移轉」](#)

簡介

透過 RMI-IIOP 負載平衡，可將 IIOP 用戶端請求分散至不同的伺服器實例或名稱伺服器中。其目的為在整個叢集上平均分散負載，從而提供可延伸性。結合了 EJB 叢集和可用性的 IIOP 負載平衡還可提供 EJB 容錯移轉。

當用戶端執行 JNDI 物件查詢時，命名服務會建立與特定伺服器實例相關聯的 `InitialContext` (IC) 物件。從那時起，使用該 IC 物件進行的所有查詢請求均會傳送至相同的伺服器實例。使用該 `InitialContext` 查詢的所有 `EJBHome` 物件均會在相同的目標伺服器上託管。之後取得的所有 `Bean` 參照也會建立在相同的目標主機上。由於建立 `InitialContext` 物件時，所有用戶端會隨機選取即時目標伺服器的清單，因此，此作業可有效地提供負載平衡。如果目標伺服器實例當機，則查詢或 EJB 方法呼叫將容錯移轉至其他伺服器實例。

IIOP 負載平衡與容錯移轉不需設定即可執行。在部署應用程式期間，無需特殊的步驟。Application Server 的 IIOP 負載平衡與容錯移轉支援動態重新配置的叢集。如果部署應用程式用戶端的 Application Server 實例加入叢集，則此 Application Server 會自動在叢集中尋找目前所有使用中的 IIOP 端點。因此，若將新實例新增至叢集，或從叢集刪除實例，則不需要手動更新端點清單。不過，用戶端至少應該為啟動程序指定兩個端點，以便在一個端點發生故障時使用另外一個端點。

需求

如果符合以下所有條件，則 Sun Java System Application Server 可透過 RMI-IIOP 提供遠端 EJB 參照和 NameService 物件的高可用性：

- 您的部署包含至少具有兩個 Application Server 實例的叢集。
- Java EE 應用程式已部署到所有加入負載平衡的 Application Server 實例和叢集。
- 已啟用 RMI-IIOP 用戶端應用程式以用於負載平衡。

Application Server 可針對在應用程式用戶端容器 (ACC) 中執行的 Java 應用程式進行負載平衡。請參閱第 211 頁的「[設定 RMI-IIOP 負載平衡和容錯移轉](#)」。

備註 – Application Server 不支援透過安全套接字層 (SSL) 執行 RMI-IIOP 負載平衡和容錯移轉。

演算法

Application Server 使用隨機演算法和循環演算法，進行 RMI-IIOP 負載平衡和容錯移轉。

在 RMI-IIOP 用戶端首次建立新的 InitialContext 物件時，會針對該用戶端產生隨機的可用 Application Server IIOP 終點清單。對於該 InitialContext 物件，負載平衡器會將查找請求和其他 InitialContext 作業導向至隨機清單中的第一個終點。如果第一個終點不可用，則使用清單中的第二個終點，依此類推。

隨後，每當用戶端建立新的 InitialContext 物件時，均會自動重建端點清單，從而將不同的 IIOP 端點用於 InitialContext 作業。

當您從透過 InitialContext 物件取得的參照中取得或建立 Bean 時，會在服務於指定給 InitialContext 物件的 IIOP 端點的 Application Server 實例上建立這些 Bean。對這些 Bean 的參照包括叢集中所有 Application Server 實例的 IIOP 終點位址。

主端點是與 InitialContext 端點 (用於查找或建立 Bean) 相對應的 Bean 端點。叢集中的其他 IIOP 終點將指定為**替代終點**。如果 Bean 的主端點變得不可用，則該 Bean 上的其他請求將容錯移轉至其中的一個替代端點。

您可以配置 RMI-IIOP 負載平衡和容錯移轉來使用 ACC 中執行的應用程式。

設定 RMI-IIOP 負載平衡和容錯移轉

您可以為應用程式用戶端容器 (ACC) 中執行的應用程式設定 RMI-IIOP 負載平衡和容錯移轉。也支援加權的循環負載平衡。

▼ 為應用程式用戶端容器設定 RMI-IIOP 負載平衡

此程序提供了有關將 RMI-IIOP 負載平衡和容錯移轉與應用程式用戶端容器 (ACC) 配合使用所需步驟的簡介。如需有關 ACC 的其他資訊，請參閱「Sun Java System Application Server 9.1 Developer's Guide」中的「Developing Clients Using the ACC」。

- 1 請至 `install_dir/bin` 目錄。
- 2 執行 `package-appclient`。

此公用程式可產生 `appclient.jar` 檔案。如需有關 `package-appclient` 的更多資訊，請參閱 `package-appclient(1M)`。
- 3 將 `appclient.jar` 檔案複製到用戶端所在位置的機器，並將其解壓縮。
- 4 編輯 `asenv.conf` 或 `asenv.bat` 路徑變數，以參照該機器上的正確目錄值。

該檔案位於 `appclient-install-dir/config/`。

如需要更新的路徑變數清單，請參閱 `package-appclient(1M)`。
- 5 如果需要，使 `appclient` 程序檔成為可執行檔。

例如，在 UNIX 上使用 `chmod 700`。
- 6 尋找叢集中至少兩個實例的 IIOP 偵聽程式連接埠號。

您可以在[步驟 7](#)中，將 IIOP 偵聽程式指定為端點。

為每個實例取得 IIOP 偵聽程式連接埠的方法如下：

 - a. 在管理主控台的樹形元件中，展開 [叢集] 節點。
 - b. 展開該叢集。
 - c. 選取叢集中的實例。
 - d. 在右窗格中，按一下 [特性] 標籤。
 - e. 記下實例的 IIOP 偵聽程式連接埠。
- 7 在 `sun-acc.xml` 檔案中至少新增兩個 `target-server` 元素。

備註 – 開發者設定檔中無法使用叢集功能。如需有關設定檔的資訊，請參閱「Sun Java System Application Server 9.1 管理指南」中的「用法設定檔」。

使用在[步驟 6](#)中所取得的端點。

如果部署應用程式用戶端的 Application Server 實例加入叢集，則此應用程式伺服器會自動在叢集中尋找目前所有使用中的 IIOP 端點。不過，用戶端至少應該為啟動程序指定兩個端點，以便在一個端點故障時使用另外一個端點。

`target-server` 元素可指定負載平衡所使用的一或多個 IIOP 端點。`address` 屬性是 IPv4 位址或主機名稱，`port` 屬性則指定連接埠號。請參閱「Sun Java System Application Server 9.1 Application Deployment Guide」中的「client-container」。

您可以改用 `endpoints` 特性來替代 `target-server` 元素，如下所示：

```
jvmarg value = "-Dcom.sun.appserv.iiop.endpoints=host1:port1,host2:port2,..."
```

8 如果需要加權的循環負載平衡，請執行下列步驟：

a. 設定每一個伺服器實例的負載平衡權數。

```
asadmin set instance-name.lb-weight=weight
```

b. 在 `sun-acc.xml` 中，將 ACC 的 `com.sun.appserv.iiop.loadbalancingpolicy` 特性設為 `ic-based-weighted`。

```
...
<client-container send-password="true">
  <property name="com.sun.appserv.iiop.loadbalancingpolicy" value="ic-based-weighted"/>
...
```

9 使用 `--retrieve` 選項部署用戶端應用程式，以取得用戶端 `jar` 檔案。

將用戶端 `jar` 檔案儲存在用戶端機器上。

例如：

```
asadmin deploy --user admin --passwordfile pw.txt --retrieve /my_dir myapp
```

10 執行應用程式用戶端，如下所示：

```
apclient -client clientjar -name appname
```

範例 11-1 設定 RMI-IIOP 加權循環負載平衡的負載平衡權數

此範例會依照下表設定一個叢集 (內含三個實例) 中的負載平衡權數。

實例名稱	負載平衡權數
i1	100
i2	200
i3	300

以下是設定這些負載平衡權數的指令序列：

```
asadmin set i1.lb-weight=100
asadmin set i2.lb-weight=200
asadmin set i3.lb-weight=300
```

接下來的步驟 若要測試容錯移轉，請停止叢集中的某個實例，然後查看該應用程式是否正常工作。您也可在用戶端應用程式中包含中斷點(或暫停)。

若要測試負載平衡，請使用多個用戶端，然後查看負載在所有端點中的分散方式。

索引

A

- active-healthcheck-enabled, 128
- AddressList, 和預設 JMS 主機, 200
- Apache Web Server
 - Application Server 安裝程式所做的修改, 110
 - 安全性檔案, 111
 - 使用負載平衡器, 105
 - 負載平衡器外掛程式所做的修改, 107-109
- asadmin create-jms-host 指令, 200
- asadmin get 指令, 198
- asadmin set 指令, 198
- authPassthroughEnabled, 133

C

- cacheDatabaseMetaData 特性, 73
- checkpoint-at-end-of-method 元素, 195
- ConnectionTrace 屬性, 68
- CoreFile 屬性, 68
- create-http-lb-config 指令, 124
- create-http-lb-ref 指令, 126
- create-node-agent 指令, 175

D

- DatabaseName 屬性, 68
- databuf 選項, 92
- DataBufferPoolSize 屬性, 68
- datadevices 選項, 63
- DataDeviceSize 屬性, 69, 83

- dbpassword 選項, 59
- dbpasswordfile 選項, 59
- default-config 配置, 158
- delete-http-lb-ref 指令, 126
- delete-node-agent 指令, 178
- devicepath 選項, 64
- DevicePath 屬性, 69, 85
- devicesize 選項, 64
- disable-http-lb-application 指令, 131
- disable-http-lb-server 指令, 130

E

- EagerSessionThreshold 屬性, 69
- EagerSessionTimeout 屬性, 69
- EJB 容器, 可用性, 193
- eliminateRedundantEndTransaction 特性, 73
- enable-http-lb-application 指令, 126
- enable-http-lb-server 指令, 126
- EventBufferSize 屬性, 69
- export-http-lb-config 指令, 129

F

- fast 選項, 81

H

- HADB
 - nodes, 89

HADB (續)

- 列出資料庫, 80
- 延伸節點, 83
- 取得 JDBC URL, 72
- 取得狀態, 88-90
- 取得資源資訊, 91-94
- 取得裝置資訊, 90
- 使用於 JMS 資料, 202
- 重新分段, 86
- 重新啟動資料庫, 79
- 重新啟動節點, 77
- 連接埠指定, 66
- 連線池特性, 73-74
- 連線池設定, 73
- 配置, 61-74
- 設定屬性, 65, 67
- 停止資料庫, 79
- 停止節點, 77
- 啟動資料庫, 78
- 啟動節點, 76
- 移除資料庫, 81
- 清除資料庫, 80
- 異質多工式裝置路徑, 66
- 資料庫名稱, 63
- 資料損壞, 81-82
- 監視, 88-94
- 增加節點, 84
- 增加機器, 83-84
- 歷史檔案, 96
- 機器維護, 94
- 環境變數, 60
- 雙網路, 36-37

HADB 配置

- 時間同步化, 39
- 節點管理員程序, 40-41
- 網路配置, 34-37

HADB 設定, 33

HADB 管理代理程式, 啟動, 42-43, 49-57

- hadbm addnodes 指令, 84
- hadbm clear 指令, 80
- hadbm clearhistory 指令, 96
- hadbm create 指令, 62
- hadbm delete 指令, 81
- hadbm deviceinfo 指令, 90

- hadbm get 指令, 67
- hadbm list 指令, 80
- hadbm refragment 指令, 86
- hadbm resourceinfo 指令, 91-94
- hadbm restart 指令, 79
- hadbm restartnode 指令, 77
- hadbm start 指令, 78
- hadbm startnode 指令, 76
- hadbm status 指令, 88-90
- hadbm stop 指令, 79
- hadbm stopnode 指令, 77
- hadbm 指令, 57-61
- historypath 選項, 64
- HistoryPath 屬性, 69
- hosts 選項, 64, 86

HTTP

- HTTPS 路由, 132
- 階段作業容錯移轉, 132-133

HTTP_LISTENER_PORT 特性, 161

HTTP_SSL_LISTENER_PORT 特性, 161

HTTP 階段作業, 22

- 分散式, 183-184

HTTPS

- 階段作業容錯移轉, 132-133
- 路由, 132

HTTPS 路由, 132-133

I

- IIOPLISTENER_PORT 特性, 161
- IIOSSL_MUTUALAUTH_PORT 特性, 161
- InternalLogbufferSize 屬性, 70
- IOP_SSL_LISTENER_PORT 特性, 161

J

- JdbcUrl 屬性, 70

JMS

- 建立主機, 200
- 連線池儲存, 22, 200
- 連線容錯移轉, 200
- 配置, 198
- 高可用性, 202

JMS 主機清單, 及連線, 200
JMX_SYSTEM_CONNECTOR_PORT 特性, 161
JMX 偵聽程式, 節點代理程式, 180
JNDI 名稱設定, 74

L

loadbalancer.xml 檔案, 129
locks 選項, 92
logbuf 選項, 92
LogbufferSize 屬性, 70

M

maxStatement 特性, 73
MaxTables 屬性, 70
Microsoft Internet Information Services (IIS), 負載平衡的修改, 113

N

nilogbuf 選項, 92
no-refragment 選項, 85
no-repair 選項, 77
nodes 選項, 89
number-healthcheck-retries, 128
NumberOfDatadevices 屬性, 70
NumberOfLocks 屬性, 70
NumberOfSessions 屬性, 70

P

password 特性, 73
portbase 選項, 65
Portbase 屬性, 70

R

RelalgdeviceSize 屬性, 70
rewrite-location 特性, 135

S

saveto 選項, 96
serverList 特性, 73
SessionTimeout 屬性, 70
set 選項, 65, 66
spares 選項, 65, 81, 85
SQLTraceMode 屬性, 70
start-node-agent 指令, 177
startlevel 選項, 76, 78
StartRepairDelay 屬性, 70
StatInterval 屬性, 71
stop-node-agent 指令, 177
sun-ejb-jar.xml 檔案, 195
Sun Java System Message Queue, 連接器, 197
Sun Java System Web Server, 由負載平衡器修改, 98
sun-passthrough.properties 檔案, 和記錄層級, 145
sun-passthrough 特性, 114
SyslogFacility 屬性, 71
SysLogging 屬性, 71
SysLogLevel 屬性, 71
SyslogPrefix 屬性, 71

T

TakeoverTime 屬性, 71

U

username 特性, 73

W

Web 伺服器, 多個實例和負載平衡, 136
Web 容器, 可用性, 188
Web 應用程式, 可分散, 186
已命名的配置
 default-config, 158
 共用, 158
 連接埠號與, 159
 預設名稱, 158
 關於, 157
分散式 HTTP 階段作業, 183-184

- 中央儲存庫,節點代理程式同步化, 169
- 可分散的 Web 應用程式, 186
- 可用性
 - EJB 容器層級, 195
 - 啟用和停用, 186
 - 適用於 Web 模組, 183-184
 - 適用於有狀態階段作業 Bean, 192
 - 層級, 186
- 目標,負載平衡器配置, 126
- 主端點,RMI-IIOP 容錯移轉, 210
- 永久存放區,適用於有狀態階段作業 Bean 狀態, 192
- 必需的連線驗證設定, 73
- 全域作業事件支援設定, 73
- 回應逾時, 125
- 名稱設定, 73
- 有狀態階段作業 Bean, 192
 - 階段作業持續性, 192,194
- 有狀態階段作業 Bean 狀態檢查點操作, 186
- 池名稱設定, 74
- 作業事件
 - 及階段作業持續性, 192,195
- 作業事件隔離設定, 73
- 作業階段持續性,及單次登入, 191-192
- 伺服器,叢集, 147
- 伺服器實例
 - 為負載平衡啟用, 126
 - 靜止, 130
- 居留式循環負載平衡, 119
- 所有連線故障設定, 73
- 重新載入間隔, 125
- 保證隔離層級設定, 73
- 持續的池大小設定, 73
- 持續性,階段作業, 22
- 信號, 37
- 負載平衡
 - Apache Web Server, 105
 - HTTP,關於, 119
 - HTTP 演算法, 119
 - Microsoft IIS, 113
 - RMI-IIOP 需求, 210
 - Sun Java System Web Server, 98
 - 多個 Web 伺服器實例, 136
 - 居留式循環, 119
 - 建立負載平衡器配置, 124
 - 負載平衡 (續)
 - 建立參照, 126
 - 記錄訊息, 143
 - 動態重新配置, 130
 - 設定, 121
 - 階段作業容錯移轉, 132-133
 - 啟用伺服器實例, 126
 - 啟用應用程式, 126
 - 等冪 URL, 136
 - 運作狀態檢查程式, 127
 - 匯出配置檔案, 129
 - 靜止伺服器實例或叢集, 130
 - 靜止應用程式, 131
 - 變更配置, 130
 - 表名稱設定, 73
 - 記錄
 - 負載平衡器, 143
 - 檢視節點代理程式記錄, 173
 - 時間同步化, 39
 - 連接埠號,與配置, 159
 - 連線池
 - HADB 特性, 73-74
 - HADB 設定, 73
 - 配置,參閱已命名的配置
 - 容錯移轉
 - JMS 連線, 200
 - RMI-IIOP 需求, 210
 - 有狀態階段作業 Bean 狀態, 192
 - 適用於 Web 模組階段作業, 183-184
 - 關於 HTTP, 119
 - 動態重新配置,負載平衡器的, 130
 - 階段作業
 - HTTP, 22
 - 持續性, 22
 - 階段作業持續性
 - 適用於 Web 模組, 183-184
 - 適用於有狀態階段作業 Bean, 192,194
 - 階段作業容錯移轉,HTTP 與 HTTPS, 132-133
 - 階段作業儲存位置
 - 適用於 HTTP 階段作業, 189
 - 適用於有狀態階段作業 Bean, 194
 - 基於 Cookie 的階段作業居留, 120
 - 等冪 URL, 136
 - 單次登入,及作業階段持續性, 191-192

- 循環負載平衡,居留式, 119
- 異常伺服器實例, 127
- 運作狀態檢查程式, 127
- 替代端點, RMI-IIOP 容錯移轉, 210
- 資料來源已啟用設定, 74
- 資料來源類別名稱設定, 73
- 資料庫供應商設定, 73
- 路由 Cookie, 126
- 節點代理程式
 - JMX 偵聽程式, 180
 - 安裝, 168
 - 刪除, 178, 179
 - 建立, 175
 - 記錄, 173
 - 停止, 177
 - 啟動, 177
 - 預留位置, 174
 - 與網域管理伺服器同步化, 169
 - 認證範圍, 180
 - 部署, 167
 - 關於, 165
- 節點管理員程序和高可用性, 40-41
- 演算法
 - HTTP 負載平衡, 119
 - RMI-IIOP 容錯移轉, 210
- 管理主控台
 - 使用以建立 JMS 主機, 200
 - 使用以配置 JMS 服務, 198
- 端點, RMI-IIOP 容錯移轉, 210
- 網域管理伺服器
 - 伺服器實例同步, 170
 - 節點代理程式同步化, 169
- 網路配置需求, 34-37
- 認證範圍, 節點代理程式, 180
- 範圍, 節點代理程式認證, 180
- 部署, 設定可用性, 186
- 輪替式升級, 137
- 靜止
 - 伺服器實例或叢集, 130
 - 應用程式, 131
- 應用程式
 - 在維持可用性的情況下升級, 137
 - 為負載平衡啟用, 126
 - 靜止, 131
- 檢查點操作, 192
 - 選取方法, 192, 195
- 叢集, 147
 - 共用, 26
 - 獨立, 26
 - 靜止, 130
- 叢集伺服器實例, 配置, 158
- 驗證方法設定, 73

