



Sun Java System Application Server Enterprise Edition 8.1 2005Q2 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：819-4958
2007年4月

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

Sun Microsystems, Inc. 對本文件所述產品所採用的技術擁有相關智慧財產權。特別是 (但不僅限於)，這些智慧財產權可能包含一項或多項美國專利，或美國及其他國家/地區的待批專利。

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

本發行物可能包含由協力廠商開發的材料。

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

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

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

本出版物所涵蓋的產品和包含的資訊受到美國出口控制法規的控制，並可能受到其他國家/地區進出口法規的管轄。嚴禁核武、導彈、生化武器或海上核武等最終用途或一般使用者直接或間接使用本產品。嚴禁向被美國禁運的國家/地區或美國出口除外清單 (包括但不僅限於被拒人清單和特別指定的國家/地區清單) 上標識的實體出口或再出口本產品。

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

目錄

前言	17
1 應用程式伺服器高可用性功能	23
高可用性簡介	23
負載平衡外掛程式	23
高可用性資料庫	24
高可用性叢集	24
更多資訊	25
高可用性階段作業持續性	26
2 安裝和設定高可用性資料庫	27
高可用性資料庫簡介	27
HADB 和 Application Server	27
HADB 伺服器架構	28
HADB 節點	29
新功能和改進的功能	29
使用 HADB 的用戶支援	31
準備 HADB 設定	32
先決條件	32
配置網路備援	32
配置共用記憶體和信號	35
▼ 在 Solaris 上配置共用記憶體和信號	36
▼ 在 Linux 上配置共用記憶體	37
同步化系統時鐘	37
檔案系統支援	38
安裝	38
HADB 安裝	38

節點管理員程序權限	39
▼ 授予節點管理員程序超級使用者權限	40
設定高可用性	40
先決條件	40
▼ 準備系統以使其具有高可用性	41
啓動 HADB 管理代理程式	41
▼ 在 Solaris 或 Linux 上使用 Java Enterprise System 啓動管理代理程式	41
▼ 在 Windows 上使用 Java Enterprise System 啓動管理代理程式	42
▼ 在安裝了獨立應用程式伺服器的 Solaris 或 Linux 上啓動管理代理程式	42
▼ 在安裝了獨立應用程式伺服器的 Windows 上使用啓動管理代理程式	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
重新啓動叢集	43
重新啓動 Web 伺服器	43
▼ 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	44
註冊 HADB 套裝軟體	45
取消註冊 HADB 套裝軟體	46
替代管理代理程式啓動程序檔	46
3 管理高可用性資料庫	47
使用 HADB 管理代理程式	47
管理代理程式指令語法	47
自訂管理代理程式配置	49
▼ 在每台 HADB 主機上自訂管理代理程式配置	49
啓動管理代理程式	50
使用 hadbm 管理指令	54
指令語法	55
安全性選項	55
一般選項	56
環境變數	57
配置 HADB	59
建立管理網域	59
建立資料庫	60

▼ 建立資料庫	60
檢視和修改配置屬性	64
配置 JDBC 連線池	69
管理 HADB	72
管理網域	72
管理節點	73
管理資料庫	75
回復階段作業資料損壞	79
▼ 使階段作業存放區返回一致狀態	79
延伸 HADB	80
增加現有節點的儲存空間	80
增加機器	81
▼ 新增機器至現有 HADB 實例	81
增加節點	81
重新分段資料庫	83
透過重新建立資料庫增加節點	84
▼ 透過重新建立資料庫增加節點	84
監視 HADB	85
取得 HADB 的狀態	85
取得裝置資訊	87
取得執行階段資源資訊	89
維護 HADB 機器	91
▼ 在一台機器上執行維護	91
▼ 在所有 HADB 機器上執行規劃的維護	92
▼ 在所有 HADB 機器上執行規劃的維護	92
▼ 發生故障時執行未規劃的維護	92
清除和歸檔歷史檔案	93
4 配置負載平衡和容錯移轉	95
負載平衡器的工作方式	95
指定的請求與未指定的請求	96
HTTP 負載平衡演算法	96
應用程式範例	96
設定 HTTP 負載平衡	97
設定負載平衡的先決條件	97

HTTP 負載平衡器部署	97
設定負載平衡的程序	98
▼ 設定負載平衡	98
配置 Web 伺服器以用於負載平衡	99
對 Sun Java System Web Server 的修改	99
使用 Apache Web Server	100
▼ 配置 Apache 安全性檔案以與負載平衡器一起工作	104
對 Microsoft IIS 的修改	105
▼ 配置 Microsoft IIS 以使用負載平衡外掛程式	105
配置多個 Web 伺服器實例	107
▼ 配置多個 Web 伺服器實例	107
配置負載平衡器	108
建立 HTTP 負載平衡器配置	108
建立 HTTP 負載平衡器參照	109
啓用用於負載平衡的伺服器實例	109
啓用用於負載平衡的應用程式	109
建立 HTTP 運作狀態檢查程式	109
匯出負載平衡器配置檔案	111
▼ 匯出負載平衡器配置	111
變更負載平衡器配置	112
啓用動態重新配置	112
停用 (靜止) 伺服器實例或叢集	112
▼ 停用某個伺服器實例或叢集	113
停用 (靜止) 應用程式	113
▼ 停用應用程式	113
配置 HTTP 和 HTTPS 容錯移轉	114
配置等冪 URL	115
在不遺失可用性的情況下升級應用程式	115
應用程式相容性	115
在單一叢集中升級	116
▼ 在單一叢集中升級應用程式	116
在多個叢集中升級	118
▼ 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：	118
升級不相容的應用程式	119
▼ 透過建立第二個叢集來升級不相容的應用程式	119
監視 HTTP 負載平衡器外掛程式	121

配置記錄訊息	121
記錄訊息類型	121
啓用負載平衡器記錄	122
▼ 開啓負載平衡器記錄	123
瞭解監視訊息	123
5 使用應用程式伺服器叢集	125
叢集簡介	125
使用叢集	125
▼ 建立叢集	126
▼ 爲叢集建立伺服器實例	127
▼ 配置叢集	128
▼ 啓動、停止和刪除叢集實例	128
▼ 配置叢集中的伺服器實例	129
▼ 爲叢集配置應用程式	129
▼ 爲叢集配置資源	130
▼ 刪除叢集	131
▼ 遷移 EJB 計時器	131
▼ 升級元件而不會使服務受到損失	132
6 管理已命名的配置	135
關於已命名的配置	135
已命名的配置	135
default-config 配置	136
建立實例或叢集時建立的配置	136
唯一連接埠號和配置	137
使用已命名的配置	137
▼ 建立已命名的配置	138
編輯已命名的配置的特性	138
▼ 編輯已命名的配置的特性	139
▼ 編輯參照配置之實例的連接埠號	139
▼ 檢視已命名的配置的目標	140
▼ 刪除已命名的配置	140

7 配置節點代理程式	141
關於節點代理程式	141
何為節點代理程式？	141
節點代理程式預留位置	143
部署節點代理程式	143
▼ 線上部署節點代理程式	143
▼ 離線部署節點代理程式	144
節點代理程式和網域管理伺服器同步	146
檢視節點代理程式記錄	149
可以通過管理主控台和 <code>asadmin</code> 工具執行的作業	150
使用節點代理程式	151
▼ 檢視一般節點代理程式資訊	151
▼ 建立節點代理程式預留位置	152
▼ 刪除節點代理程式配置	153
▼ 編輯節點代理程式配置	153
▼ 編輯節點代理程式範圍	154
▼ 為 JMX 編輯節點代理程式的偵聽程式	154
透過 <code>asadmin</code> 使用節點代理程式	156
建立節點代理程式	156
啓動節點代理程式	157
停止節點代理程式	158
刪除節點代理程式	158
8 配置高可用性階段作業持續性和容錯移轉	159
階段作業持續性和容錯移轉簡介	159
需求	159
限制	160
應用程式範例	160
設定高可用性階段作業持續性	161
▼ 設定高可用性階段作業持續性	161
啓用階段作業可用性	162
HTTP 階段作業容錯移轉	163
配置 Web 容器的可用性	163
▼ 使用管理主控台啓用 Web 容器的可用性	163
配置個別 Web 應用程式的可用性	165

將階段作業容錯移轉與單次登入配合使用	165
有狀態階段作業 Bean 容錯移轉	167
配置 EJB 容器的可用性	167
▼ 啓用 EJB 容器的可用性	167
配置個別應用程式或 EJB 模組的可用性	169
配置個別 Bean 的可用性	169
指定進行檢查點操作方法	170
9 Java 訊息服務的負載平衡和容錯移轉	173
Java 訊息服務簡介	173
範例應用程式	173
詳細資訊	174
配置 Java 訊息服務	174
Java 訊息服務整合	175
JMS 主機清單	175
連線池儲存和容錯移轉	176
負載平衡的訊息流入	177
將 MQ 叢集與應用程式伺服器配合使用	178
▼ 啓用 MQ 叢集與應用程式伺服器叢集	178
10 RMI-IIOP 負載平衡和容錯移轉	183
簡介	183
需求	183
演算法	184
範例應用程式	184
設定 RMI-IIOP 負載平衡和容錯移轉	185
▼ 爲應用程式用戶端容器設定 RMI-IIOP 負載平衡	185
▼ 爲獨立用戶端設定 RMI-IIOP 負載平衡和容錯移轉	186
索引	189

圖清單

圖 2-1	HADB 架構	28
-------	---------------	----

表清單

表 2-1	hadbm registerpackage 選項	45
表 3-1	管理代理程式共用選項	48
表 3-2	管理代理程式服務選項 (僅適用於 Windows)	48
表 3-3	配置檔案設定	49
表 3-4	hadbm 安全性選項	56
表 3-5	hadbm 一般選項	57
表 3-6	HADB 選項與環境變數	57
表 3-7	hadbm create 選項	61
表 3-8	配置屬性	66
表 3-9	HADB 連線池設定	70
表 3-10	HADB 連線池特性	70
表 3-11	HADB JDBC 資源設定	71
表 3-12	hadbm clear 選項	78
表 3-13	hadbm addnodes 選項	82
表 3-14	HADB 狀態	86
表 3-15	hadbm resourceinfo 指令選項	89
表 4-1	負載平衡器配置參數	108
表 4-2	運作狀態檢查程式參數	110
表 4-3	運作狀態檢查程式手動特性	111
表 7-1	在遠端伺服器實例之間同步的檔案和目錄	147
表 7-2	通過管理主控台或是 asadmin 指令都可以執行的作業	150

範例清單

範例 2-1	設定多重路徑	33
範例 2-2	取消註冊 HADB 的範例	46
範例 3-1	hadbm 指令範例	55
範例 3-2	建立 HADB 管理網域	59
範例 3-3	建立資料庫範例	62
範例 3-4	使用 hadbm get 之範例	64
範例 3-5	建立連線池	71
範例 3-6	啓動節點範例	74
範例 3-7	停止節點範例	75
範例 3-8	重新啓動節點範例	75
範例 3-9	啓動資料庫範例	76
範例 3-10	停止資料庫範例	77
範例 3-11	移除資料庫範例	79
範例 3-12	設定資料裝置大小範例	81
範例 3-13	增加節點範例	82
範例 3-14	重新分段資料庫範例	84
範例 3-15	取得 HADB 狀態範例	86
範例 3-16	取得裝置資訊範例	88
範例 3-17	資料緩衝區池資訊範例	90
範例 3-18	鎖定資訊範例	90
範例 3-19	記錄緩衝區資訊範例	91
範例 3-20	內部記錄緩衝區資訊範例	91
範例 7-1	建立節點代理程式的範例	156
範例 8-1	啓用了可用性的 EJB 部署描述元範例	170
範例 8-2	指定方法檢查點之 EJB 部署描述元範例	170

前言

「高可用性管理指南」說明了 Sun Java™ System Application Server 的高可用性功能，其中包括如何：

- 安裝、配置以及管理高可用性資料庫 (HADB)。
- 安裝、配置以及使用 HTTP 負載平衡外掛程式。
- 使用已命名的配置以共用伺服器配置屬性。
- 設定和使用具有高可用性的叢集。
- 配置節點代理程式。
- 配置和使用高可用性階段作業持續性。
- 使用其他高可用性功能，例如 Java 訊息服務和 RMI-IIOP 容錯移轉。

本書適用對象

本指南適用於生產環境中的系統管理員。本書假設您熟悉：

- 基本系統管理
- 安裝軟體
- 使用 Web 瀏覽器
- 在終端機視窗中發佈指令

閱讀本書之前

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，您應熟悉位於 <http://docs.sun.com/coll/1286.1> 和 <http://docs.sun.com/coll/1412.1> 的系統文件。

本書架構

- 第 1 章提供有關 Application Server 的高可用性功能的簡介。
- 第 2 章說明如何安裝和設定高可用性資料庫。
- 第 3 章說明如何管理高可用性資料庫。
- 第 4 章說明如何安裝、配置和使用 HTTP 負載平衡外掛程式。
- 第 5 章說明 Application Server 叢集以及如何對其進行配置和管理。
- 第 6 章說明如何使用已命名的配置來共用 Application Server 配置屬性。
- 第 7 章說明節點代理程式以及如何對其進行管理。
- 第 8 章說明如何設定高可用性階段作業持續性。
- 第 9 章說明 Java 訊息服務負載平衡和容錯移轉。
- 第 10 章說明 RMI-IIOP 負載平衡和容錯移轉。

Application Server 文件集

Application Server 文件集描述部署規劃與系統安裝。獨立的 Application Server 文件位於 <http://docs.sun.com/app/docs/coll/1310.1> 和 <http://docs.sun.com/app/docs/coll/1416.1>。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
版本說明	軟體與文件的最新資訊。其中包括以表格形式對所支援的硬體、作業系統、JDK 和 JDBC/RDBMS 所做的全面概括。
Quick Start Guide	如何開始使用 Application Server 產品。
Installation Guide	安裝軟體及其元件。
Deployment Planning Guide	評估系統需求和企業狀況，確保以最適合站點的方式部署 Application Server。此外還說明了部署伺服器時應該注意的常見問題。
Developer's Guide	建立和實作要在 Application Server 上執行的 Java 2 Platform, Enterprise Edition (J2EE™ 平台) 應用程式，這些應用程式遵循針對 J2EE 元件和 API 的開放式 Java 標準模型。其中包括有關開發者工具、安全性、組合、部署、除錯和建立生命週期模組的一般資訊。

表 P-1 Application Server 文件集中的書籍 (續)

書籍標題	說明
J2EE 1.4 Tutorial	使用 J2EE 1.4 平台技術與 API 來開發 J2EE 應用程式。
管理指南	從管理主控台配置、管理和部署 Application Server 子系統和元件。
High Availability Administration Guide	高可用性資料庫安裝後的配置和管理說明。
Administration Reference	編輯 Application Server 配置檔案 domain.xml。
Upgrade and Migration Guide	將應用程式遷移到新的 Application Server 程式設計模型，特別是從 Application Server 6.x 和 7 進行遷移。該指南還說明相鄰產品發行版本的差異，以及可能導致與產品規格不相容的配置選項。
Performance Tuning Guide	調校 Application Server 以提昇效能。
Troubleshooting Guide	解決 Application Server 問題。
Error Message Reference	解決 Application Server 錯誤訊息。
Reference Manual	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。

相關書籍

如需其他 Sun Java System 伺服器文件，請參閱：

- Message Queue 文件
- Directory Server 文件
- Web Server 文件

有關 Java ES 及其元件的文件位於 <http://docs.sun.com/prod/entsys.05q4> 和 http://docs.sun.com/prod/entsys.05q4?l=zh_TW。

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>install-dir</i>	表示 Application Server 的基底安裝目錄。	<p>Solaris™ 平台上的 Sun Java Enterprise System 安裝：</p> <p><i>/opt/SUNWappserver/appserver</i></p> <p>Linux 平台上的 Sun Java Enterprise System 的安裝：</p> <p><i>/opt/sun/appserver/</i></p> <p>其他 Solaris 和 Linux 的安裝 (非超級使用者)：</p> <p><i>user's home directory/SUNWappserver</i></p> <p>其他 Solaris 和 Linux 的安裝 (超級使用者)：</p> <p><i>/opt/SUNWappserver</i></p> <p>Windows，所有安裝：</p> <p><i>SystemDrive:\Sun\AppServer</i></p>
<i>domain-root-dir</i>	表示包含所有網域的目錄。	<p>Solaris 平台上的 Sun Java Enterprise System 的安裝：</p> <p><i>/var/opt/SUNWappserver/domains/</i></p> <p>Linux 平台上的 Sun Java Enterprise System 的安裝：</p> <p><i>/var/opt/sun/appserver/domains/</i></p> <p>所有其他安裝：</p> <p><i>install-dir/domains/</i></p>
<i>domain-dir</i>	<p>表示網域的目錄。</p> <p>在配置檔案中，您可能會看到 <i>domain-dir</i> 顯示如下：</p> <p><code>#{com.sun.aas.instanceRoot}</code></p>	<i>domain-root-dir/domain-dir</i>
<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	意義	範例
AaBbCc123	指令、檔案及目錄的名稱；螢幕畫面輸出。	請編輯您的 <code>.login</code> 檔案。 請使用 <code>ls -a</code> 列出所有檔案。 <code>machine_name% you have mail.</code>
AaBbCc123	您所鍵入的內容 (與螢幕畫面輸出相區別)。	<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。	移除檔案的指令為 <code>rm 檔案名稱</code> 。
AaBbCc123	新術語，要強調的詞。	快取記憶體 是儲存在本機的副本。 請 不要 儲存此檔案。 備註 ：某些重點項目在線上以粗體顯示。 請參閱「使用者指南」中的第 6 章。
<code>「AaBbCc123」</code>	用於書名及章節名稱。	

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。
-	連接需同時按下的多個按鍵。	Control-A	同時按 Control 鍵和 A 鍵。
+	連接需連續按下的多個按鍵。	Ctrl+A+N	按 Control 按鍵，然後鬆隔 A 依次按後面的按鍵。

表 P-4 符號慣例 (續)

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

存取 Sun 線上資源

您可以透過 docs.sun.comSM 網站線上存取 Sun 技術文件。您可以瀏覽 docs.sun.com 歸檔或搜尋特定文件標題或主題。這些文件以 PDF 和 HTML 格式的線上檔案方式提供。殘障人士可以使用輔助技術閱讀這兩種格式的檔案。

若要存取以下 Sun 資源，請至 <http://www.sun.com>：

- Sun 產品下載
- 服務和解決方案
- 支援 (包含修補程式和更新)
- 培訓
- 研究
- 社群 (例如，Sun Developer Network)

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 819-4958。

應用程式伺服器高可用性功能

本章說明 Sun Java System Application Server Enterprise Edition 中的高可用性功能，包含以下主題：

- 第 23 頁的「高可用性簡介」
- 第 26 頁的「高可用性階段作業持續性」

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。應用程式伺服器為 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。

應用程式伺服器經由以下子元件和功能提供高可用性：

- 第 23 頁的「負載平衡外掛程式」
- 第 24 頁的「高可用性資料庫」(HADB)
- 第 24 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 和 HTTPS 請求，然後將這些請求轉寄至叢集中的應用程式伺服器實例。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 Enterprise Edition 包含適用於 Sun Java System Web Server、Apache Web Server 和 Microsoft Internet Information Server 的負載平衡外掛程式。

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 4 章

高可用性資料庫

Application Server Enterprise Edition 提供高可用性資料庫 (HADB)，用於 HTTP 階段作業和有狀態階段作業 Bean 資料的高可用性儲存。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。

應用程式伺服器不承擔狀態管理職責，這具有顯著優勢。應用程式伺服器實例在其生命週期內做為可延伸且高效能的 Java™ 2 Platform, Enterprise Edition (J2EE™ 平台) 容器執行，並將狀態複製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應用程式伺服器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。如果應用程式伺服器實例還執行複製，J2EE 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應用程式伺服器安裝，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 Enterprise Edition 8.1 2005Q2 Deployment Planning Guide」中的「Planning for Availability」和「Sun Java System Application Server Enterprise Edition 8.1 2005Q2 Deployment Planning Guide」中的第 3 章「Selecting a Topology」。

高可用性叢集

叢集是應用程式伺服器實例集合，以一個邏輯實體形式一起運作。叢集為一個或多個 J2EE 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，J2EE 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` 是叢集的名稱。

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。

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 Enterprise Edition 8.1 2005Q2 Deployment Planning Guide」。此手冊還提供以下概念的高階簡介：

- 應用程式伺服器元件，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

如需有關開發和部署利用高可用性功能的應用程式之更多資訊，請參閱「Sun Java System Application Server Enterprise Edition 8.1 2005Q2 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應用程式伺服器以取得高可用性最佳效能的資訊，請參閱「Sun Java System Application Server Enterprise Edition 8.1 2005Q2 Performance Tuning Guide」，本書說明以下主題：

- 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

高可用性階段作業持續性

J2EE 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。為了取得高可用性，應用程式伺服器提供在 HADB 中持續階段作業狀態的功能。如果託管使用者階段作業的應用程式伺服器實例出現故障，階段作業狀態是可以回復的，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 8 章

安裝和設定高可用性資料庫

本小節包含以下主題：

- 第 27 頁的「高可用性資料庫簡介」
- 第 32 頁的「準備 HADB 設定」
- 第 38 頁的「安裝」
- 第 40 頁的「設定高可用性」
- 第 44 頁的「升級 HADB」

高可用性資料庫簡介

本小節說明高可用性資料庫 (HADB) 並說明如何設定和配置 HADB 以與應用程式伺服器配合使用。

本小節包含下列主題：

- 第 27 頁的「HADB 和 Application Server」
- 第 28 頁的「HADB 伺服器架構」
- 第 29 頁的「HADB 節點」

HADB 和 Application Server

HADB 是具有水平可延伸性的資料庫，可獨立於應用程式伺服器階層之外執行和管理。它旨在使用負載平衡、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。

Application Server 使用 HADB 儲存 HTTP 和有狀態階段作業 Bean (SFSB) 階段作業資料。如果不具有階段作業持續性機制，則在 Web 或 EJB 容器執行容錯移轉時，會遺失 HTTP 或 SFSB 階段作業狀態資料。

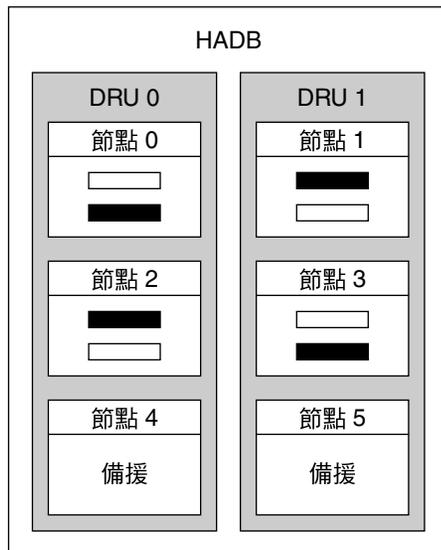
應用程式伺服器不負責狀態管理工作，這種作法的優點非常重要。應用程式伺服器實例在其生命週期內充當可延伸且高效能的 Java™ 2 Platform, Enterprise Edition (J2EE™ 平

台) 容器，並將狀態複製功能委託給外部高可用性狀態服務。由於採用此鬆耦合架構，因此可以輕鬆將應用程式伺服器實例增加至叢集或從叢集中移除應用程式伺服器實例。您可以獨立地縮放 HADB 狀態複製服務的比例，以獲得最佳化可用性和效能。

HADB 伺服器架構

高可用性是指無論發生由於升級而引起的預期故障，還是發生由於硬體或軟體故障所導致的未預期故障，系統仍然具有可用性。HADB 基於簡單資料模型和具有可延伸性的備援高效能技術。在高效能企業應用程式伺服器環境中，HADB 為傳送各種類型的階段作業狀態持續性提供了理想的平台。

下圖顯示了具有四個使用中節點和兩個備用節點的資料庫架構。節點 0 和 1 為鏡像節點對，節點 2 和 3 亦如此。



□ 主片段
 ■ 備用片段

圖 2-1 HADB 架構

HADB 透過對資料進行分段和複製來實現資料的高可用性。資料庫中的所有表格均進行了分割，以建立大大小小相同的子集 (稱為片段)。分段程序基於在資料庫節點中平均分散資料的雜湊功能。每個片段儲存兩次，一次在資料庫中，一次在鏡像節點中。這可確保資料具有容錯和快速回復功能。此外，如果節點發生故障或關閉，備用節點會接管執行，直至該節點可重新使用時為止。

HADB 節點被組織為兩個資料備援單元 (DRU)，這兩個單元互相鏡像。每個 DRU 均由半數使用中節點和半數備用節點組成，並包含該資料的一份完整副本。為確保具有容錯功能，支援某個 DRU 的電腦自身必須完全提供電源 (建議使用不間斷的電源供應)、處理單元以及儲存裝置。如果在一個 DRU 中發生斷電，則另一個 DRU 中的節點可以繼續處理請求，直至供電恢復正常。

如果不具有階段作業持續性機制，則在 Web 或 EJB 容器容錯移轉至另一個 Web 或 EJB 容器時，會遺失 HTTP 或 SFSB 階段作業狀態 (包括鈍化的階段作業狀態)。將 HADB 用於階段作業持續性可避免發生此情況。HADB 可在獨立但整合良好的持續性儲存階層中儲存和擷取狀態資訊。

刪除階段作業資料後，HADB 會收回空間。HADB 將階段作業資料記錄置於固定大小的區段中。刪除所有區段記錄後，會釋放該區段。可以隨機刪除區段記錄，這會在區段中建立孔。將新記錄插入至區段並需要連續空間時，會移除孔，並因此而壓縮該區段。

以上為架構的簡要概述。如需更多資訊，請參閱「Sun Java System Application Server Enterprise Edition 8.1 2005Q2 Deployment Planning Guide」。

HADB 節點

資料庫節點由一組程序、共用記憶體體的專屬區域以及一個或多個輔助儲存裝置組成。資料庫可儲存、更新並擷取階段作業資料。每個節點均有一個鏡像節點，因此節點成對出現。此外，為最大化可用性，還包含兩個或更多備用節點 (每個 DRU 中包含一個)，這樣，如果節點發生故障，備用節點便可在修復該節點時接管執行。

如需節點拓撲替代的說明，請參閱「Sun Java System Application Server Enterprise Edition 8.1 2005Q2 Deployment Planning Guide」中的第 3 章「Selecting a Topology」。

新功能和改進的功能

與 Sun Java System Application Server Enterprise Edition 8.1 一起提供的 HADB 版本擁有許多新功能和改進的功能。

透過變更管理系統的基礎元件，改進了 HADB 管理。舊的 hadbm 介面功能略做修改後被保留下來。這些變更還移除了對 SSH/RSH 的相依性。

管理代理程式伺服器程序 (ma) 構成網域，並將資料庫配置儲存在儲存庫中。該儲存庫資訊會在所有代理程式之間分散。

以下主題提供更多詳細資訊：

- 第 30 頁的「一般改進」
- 第 30 頁的「特定變更」

一般改進

對此 HADB 版本進行了以下一般改進：

- HADB 不再需要 SSH/RSH。
- HADB 管理的管理員密碼增強了安全性。
- 自動線上升級至新版本。
- 移除了對單一主機的相依性。
- 支援資料庫的異質性配置。可分別設定裝置路徑和歷程記錄路徑。
- 可統一管理多個平台。

特定變更

與舊版本相比，此 HADB 版本包含以下變更。

- 現在需要 UDP 多重播送以用於網路配置。
- 現在需要在所有 HADB 主機上執行管理代理程式 ma。
- 用於網域管理的新 hadbm 指令：`hadbm createdomain`、`hadbm deletedomain`、`hadbm extenddomain`、`hadbm reducedomain`、`hadbm listdomain`、`hadbm disablehost`。用於套裝軟體管理的新指令：`hadbm registerpackage`、`hadbm unregisterpackage`、`hadbm listpackage`。
- 所有 hadbm 指令均具有以下新選項：
 - `adminpassword`
 - `adminpasswordfile`
 - `no-adminauthentication`
 - `agent`
 - `javahome`

對 `hadbm create` 所做的變更：

- 新選項：
 - `no-clear`
 - `no-cleanup`
 - `package`
 - `packagepath`
 - `agent`

延伸選項

- `hosts` (在網域中註冊主機)。
- `set`

已移除的選項：

- `inetd`
- `inetdsetupdir`
- `configpath`

- `installpath`
- `set TotalDataDevideSizePerNode`
- `set managementProtocol`

已修改的：`devicesize` 現在為選擇性的，而非必需的。

`hadbm startnode` 和 `hadbm restartnode` 指令的 `startlevel` 選項有一個新值 `clear`。

對 `hadbm addnodes` 所做的變更：新選項：`set`、`historypath`、`devicepath`。移除了 `inetdsetupdir` 選項。

對 `hadbm get` 和 `hadbm set` 所做的變更：新屬性為 `historypath` (歷史檔案的異質性路徑) 和 `packagename`。刪除的屬性為：`managementProtocol`、`TotalDeviceSizePerNode`、`installpath` 以及 `syslogging`。

使用 HADB 的用戶支援

在就 HADB 問題致電用戶支援之前，請儘可能多地收集以下資訊：

- 系統使用設定檔：
 - 同步運作的主動使用者的數目
 - 被動使用者的數目
 - 每秒進入系統的使用者的數目
 - 平均階段作業大小
 - 階段作業狀態逾時時間 (`SessionTimeout` 值)
 - 每秒每個使用者的作業事件率

機器特性：

- RAM
- CPU 的數目
- CPU 速度
- 作業系統版本
- 實體磁碟數目
- 總磁碟大小
- 可用磁碟空間
- 資料傳輸容量

網路特性：

- 傳輸容量
- 每個節點的主機名稱 (網路介面) 數目

HADB 資料：

- 歷史檔案

- `cfg` 和 `meta` 檔案，位於 `dbconfigpath/databasename/nodeno` 目錄中。
`dbconfigpath` 在管理代理程式配置檔案中的變數 `ma.server.dbconfigpath` 中定義。
- 版本資訊 (`hadbm --version`)

準備 HADB 設定

本小節論述以下主題：

- 第 32 頁的「先決條件」
- 第 35 頁的「配置共用記憶體和信號」
- 第 32 頁的「配置網路備援」
- 第 37 頁的「同步化系統時鐘」
- 第 38 頁的「檔案系統支援」

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 Enterprise Edition 8.1 2005Q2 版本說明」。

先決條件

設定和配置 HADB 之前，請確保您的環境符合以下需求：

- 已啟用 IPv4。HADB 僅支援 IPv4。在用於 HADB 的介面上停用 IPv6。
- 必須將網路 (主機上的路由器、交換器和網路介面) 配置為可執行使用者資料封協定 (UDP) 多重播送。如果 HADB 主機跨越多個子網路，則在子網路之間配置路由器以在子網路之間轉寄 UDP 多重播送訊息。
- 配置位於 HADB 主機之間或位於 HADB 與應用程式伺服器主機之間的所有防火牆，以允許所有 UDP 流量 (一般和多重播送)。
- 請勿使用主機的動態 IP 位址 (由動態主機配置協定，即 DHCP 指定)，這些位址為 `hadbm createdomain`、`hadbm extenddomain`、`hadbm create` 或 `hadbm addnodes` 指令的一部分。

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，可以設定網路多重路徑。
- 配置在除 Windows Server 2003 之外的所有平台上均受支援的雙網路。

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「Administering Network Multipathing」一節。

▼ 若要配置已使用 IP 多重路徑的 HADB 主機電腦，請執行以下步驟：

1 設定網路介面故障偵測時間。

爲了讓 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 1 秒 (1000 毫秒)，此時間由 `/etc/default/mpathd` 中的 `FAILURE_DETECTION_TIME` 參數指定。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爲多重路徑介面群組。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

假設具有兩個名爲 `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爲多重路徑群組。在每台主機上均執行 `ifconfig -a`。

`host1` 上的輸出爲：

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

bge1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4>
mtu 1500 index 6 inet 129.159.115.12 netmask ffffffff 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 ffffffff0 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 ffffffff0 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列示在 `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。若要在每個主機上使用一個資料庫節點建立資料庫，請使用指令 `hadbm create --host`。例如：

```
hadbm create --host 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 --host 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 49 頁的「配置檔案」。

配置雙網路

若要啓用 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。

▼ 在 Solaris 上配置共用記憶體和信號

- 1 以超級使用者身份登入。
- 2 配置共用記憶體。

將 `shmmax` 的值設定為 HADB 主機電腦上實體記憶體的大小。共用記憶體區段的最大大小必須大於 HADB 資料庫緩衝區池的大小。例如，對於具有 2 GB (0x8000000 十六進制) 主記憶體的機器，將以下內容增加至 `/etc/system` 檔案：

```
set shmsys:shminfo_shmmax=0x80000000
set shmsys:shminfo_shmseg=20
```

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

將 `shminfo_shmmax` 設定為系統的記憶體總容量 (以十六進制表示法顯示的 `0x80000000` 值，表示 2 GB 的記憶體)。

備註 – 使用記憶體大小的十六進制值指定 `shmsys:shminfo_shmmax` 的值。若要確定主機的記憶體，請使用以下指令：

```
prtconf | grep Memory
```

- 3 配置信號。

檢查 `/etc/system` 檔案，以查找信號配置項目。此檔案可能已包含 `semnmi`、`semmns` 和 `semmnu` 項目。例如：

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

如果這些項目已存在，請將這些值分別增加 16、128 和 1000。這樣，上述範例中的項目將變更為：

```
set semsys:seminfo_semnmi=26
set semsys:seminfo_semmns=188
set semsys:seminfo_semmnu=1030
```

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：

```
set semsys:seminfo_semnmi=16
set semsys:seminfo_semmns=128
set semsys:seminfo_semmnu=1000
```

此作業足可使電腦上執行多達 16 個 HADB 節點。如需設定多於 16 個節點的資訊，請參閱「Sun Java System Application Server Enterprise Edition 8.1 2005Q1 Performance Tuning Guide」中的 HADB 章節。

- 4 重新開機。

▼ 在 Linux 上配置共用記憶體

- 1 以超級使用者身份登入。
- 2 編輯檔案 `/etc/sysctl.conf`
- 3 設定 `kernel.shmax` 和 `kernel.shmall` 參數。

`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，對於一台具有 512 MB 實體記憶體的機器：

```
kernel.shmax=536870912
kernel.shmall=536870912
```

- 4 重新開機。使用以下指令：

```
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.
```

檔案系統支援

本小節說明 HADB 對某些檔案系統的限制。

Red Hat Enterprise Linux

HADB 支援 Red Hat Enterprise Linux 3.0 上的 ext2 和 ext3 檔案系統。對於 Red Hat Enterprise Linux 2.1，HADB 支援 ext2 檔案系統。

Veritas 檔案系統

在 Solaris 上使用 Veritas 檔案系統時，HADB 會將訊息 `WRN: Direct disk I/O mapping failed` 寫入至歷史檔案。此訊息表示 HADB 無法開啓資料和記錄裝置的直接輸入/輸出 (I/O)。直接 I/O 可降低寫入磁碟頁面的 CPU 消耗。它也會減少管理作業系統中「已使用過的」資料頁面的經常性耗用時間。

若要同時使用直接 I/O 與 Veritas 檔案系統，請執行以下操作之一：

- 在透過選項 `mincache=direct` 掛載的檔案系統上建立資料和記錄裝置。此選項會套用至檔案系統上建立的所有檔案。請查閱 `mount_vxfs(1M)` 指令，以取得詳細資訊。
- 使用 Veritas Quick I/O 工具對檔案系統檔案執行原始 I/O。如需詳細資訊，請參閱「VERITAS File System 4.0 Administrator's Guide for Solaris」。

備註 – 這些配置尚未在 Sun Java System Application Server 中進行測試。

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 Enterprise Edition 8.1 2005Q2 Deployment Planning Guide」中的第 3 章「Selecting a Topology」。但是，您必須安裝 HADB 管理用戶端，才能使用 `asadmin ha-config-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 ha-configure-cluster command`，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`。對於獨立的應用程式伺服器安裝程式，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儘快回應。`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 可執行檔不具有適當的權限，您可能會注意到資源不足的徵兆，例如：

- 效能問題或 HADB 歷程記錄中的 HIGH LOAD 訊息。
- HADB 歷史檔案中顯示警告「程序已被阻斷 *x* 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_PRIOCNTL` 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：

```
chown root 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：

```
chmod u+s 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 32 頁的「先決條件」
- 第 41 頁的「啟動 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「重新啟動叢集」

先決條件

配置 HADB 之前，請執行以下作業：

▼ 準備系統以使其具有高可用性

- 1 安裝應用程式伺服器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 Enterprise Edition 8.1 2005Q2 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。

如需更多資訊，請參閱「Sun Java System Application Server Enterprise Edition 8.1 2005Q2 管理指南」。

- 3 安裝並配置 Web 伺服器軟體。

如需更多資訊，請參閱第 99 頁的「配置 Web 伺服器以用於負載平衡」。

- 4 設定並配置負載平衡。

如需更多資訊，請參閱第 97 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。

對於生產部署，將管理代理程式做為服務啓動以確保其可用性。本小節提供了有關使用預設配置將管理代理程式做為服務啓動的簡短說明。

如需更多詳細資訊 (包括有關在主控制台模式中啓動管理代理程式以進行測試或評估的說明，以及有關自訂其配置的資訊)，請參閱第 47 頁的「使用 HADB 管理代理程式」。

本小節說明在使用 Java Enterprise System 時，如何使用預設配置將管理代理程式做為服務啓動。

▼ 在 Solaris 或 Linux 上使用 Java Enterprise System 啓動管理代理程式

- 1 建立與檔案 `/etc/init.d/ma-initd` 的以下符號連結：

```
/etc/rc0.d/K20ma-initd
/etc/rc1.d/K20ma-initd
/etc/rc2.d/K20ma-initd
/etc/rc3.d/S99ma-initd
/etc/rc5.d/S99ma-initd
/etc/rcS.d/K20ma-initd
```

2 重新開機。

若要關閉自動啟動和停止代理程式，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 Windows 上使用 Java Enterprise System 啟動管理代理程式

1 開啟指令視窗。

2 輸入指令：`HADB_install_dir\bin\ma -i`。

此指令可使用預設配置安裝並啟動管理代理程式。

接下來的步驟 若要停止管理代理程式並做爲服務移除 (取消註冊)，請使用以下指令：`HADB_install_dir\bin\ma -r`

▼ 在安裝了獨立應用程式伺服器的 Solaris 或 Linux 上啟動管理代理程式

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`。

替代程序檔中 `HADB_ROOT` 和 `HADB_MA_CFG` 的預設值以反映安裝：

- `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 49 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 建立與檔案 `/etc/init.d/ma-initd` 的以下符號連結：

```
/etc/rc0.d/K20ma-initd
/etc/rc1.d/K20ma-initd
/etc/rc2.d/K20ma-initd
/etc/rc3.d/S99ma-initd
/etc/rc5.d/S99ma-initd
/etc/rcS.d/K20ma-initd
```

▼ 在安裝了獨立應用程式伺服器的 Windows 上使用啟動管理代理程式

1 開啟指令視窗。

2 輸入指令：`HADB_install_dir\bin\ma -i ma.cfg`

現在，如果程序失敗或機器重新啟動，則管理代理程式將自動重新啟動。

接下來的步驟 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 ma.cfg`

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 126 頁的「建立叢集」。

從執行網域管理伺服器的機器上，使用以下指令將叢集配置為使用 HADB：

```
asadmin configure-ha-cluster --user admin --hosts hadb_hostname ,hadb_hostname
--devicesize 256 clusterName
```

使用執行 HADB 的機器之主機名稱替代 `hadb_hostname`，並使用叢集名稱替代 `clusterName`。如果僅使用一台機器，則必須提供兩次主機名稱。

該簡化範例在同一機器上執行兩個 HADB 節點。在生產設定中，建議使用多台機器。

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]。

重新啓動叢集

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

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)`。

重新啓動 Web 伺服器

若要重新啓動 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。

以下小節說明如何升級 HADB 安裝：

- 第 44 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 46 頁的「替代理程式代理程式啟動程序檔」

▼ 將 HADB 升級至較新的版本

- 1 安裝新的 HADB 版本。
- 2 取消註冊現有 HADB 安裝，如第 46 頁的「取消註冊 HADB 套裝軟體」中所述。
- 3 註冊新的 HADB 版本，如第 45 頁的「註冊 HADB 套裝軟體」中所述。

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。預設套裝軟體名稱為以 V 開頭並包含 hadbm 程式之版本編號的字串。

- 4 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* 為新 HADB 套裝軟體的版本編號。

- 5 如有必要，替代理程式啟動程序檔。
如需更多資訊，請參閱第 46 頁的「替代理程式啟動程序檔」。

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 `hadm 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。

指令語法為：

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

package-name 運算元為套裝軟體的名稱。

下表說明了特殊的 `hadbm registerpackage` 指令選項。請參閱第 55 頁的「安全性選項」和第 56 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
<code>--hosts=<i>hostlist</i></code>	主機清單，以逗號分隔或置於雙引號中並以空格分隔。
<code>-H</code>	
<code>--packagepath=<i>path</i></code>	HADB 軟體套裝軟體的路徑。
<code>-L</code>	

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啟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。

使用 `hadbm unregisterpackage` 指令之前，請確保：所有管理代理程式均已配置且在主機清單中的所有主機上執行；管理代理程式的儲存庫可用於更新；已在管理網域中註冊套裝軟體；未將任何現有資料庫配置為在要取消註冊的套裝軟體上執行。

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。

請參閱上述第 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啟用主機做為預設主機。請參閱第 55 頁的「安全性選項」和第 56 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。

管理高可用性資料庫

本章說明 Sun Java System Application Server Enterprise Edition 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模，然後在不同的機器上安裝 HADB 軟體。

本章說明以下主題：

- 第 47 頁的「使用 HADB 管理代理程式」
- 第 54 頁的「使用 `hadbm` 管理指令」
- 第 59 頁的「配置 HADB」
- 第 72 頁的「管理 HADB」
- 第 80 頁的「延伸 HADB」
- 第 85 頁的「監視 HADB」
- 第 91 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啓動，從而確保 HADB 節點管理程序的可用性。

- 第 47 頁的「管理代理程式指令語法」
- 第 49 頁的「自訂管理代理程式配置」
- 第 50 頁的「啓動管理代理程式」

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 47 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 47 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 49 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
<code>--define name=value-D</code>	指定特性名稱的 值 ，其中，特性為第 49 頁的「配置檔案」中定義的特性之一。此選項可重複多次。	無
<code>--help-?</code>	顯示說明資訊。	False
<code>--javahome path-j</code>	使用位於 <i>path</i> 的 Java Runtime Environment (1.4 或更高版本)。	無
<code>--systemroot path-y</code>	作業系統根的路徑，通常在 %SystemRoot% 中設定。	無
<code>--version-V</code>	顯示版本資訊。	False

第 47 頁的「管理代理程式指令語法」說明將管理代理程式做為 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 (") 替換包含空格的檔案路徑。使用雙引號和反斜線替換冒號 (:) 磁碟分隔符號和反斜線 (\) 目錄分隔符號，如下所示："*x*: 和 "*x*\"。

表 3-2 管理代理程式服務選項 (僅適用於 Windows)

選項	說明	預設
<code>--install-i</code>	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
<code>--name servicename-n</code>	在主機上執行多個代理程式時，使用指定的服務名稱。	HADBMgrt Agent
<code>--remove-r</code>	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
<code>--service-s</code>	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。啓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。

▼ 在每台 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 啓動管理代理程式，並將自訂的配置檔案指定為引數。

配置檔案

使用 Java Enterprise System，檔案中的所有項目均已加入註釋。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，然後啓動管理代理程式，並將配置檔案做爲引數指定。

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。有效值爲 SEVERE、ERROR、WARNING、INFO、FINE、FINER 及 FINEST	警告
logfile.loglevel	記錄檔記錄層級。有效值爲 SEVERE、ERROR、WARNING、INFO、FINE、FINER 及 FINEST	資訊
logfile.name	記錄檔名稱和位置。必須爲具有讀取/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

表 3-3 配置檔案設定 (續)

設定名稱	說明	預設
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 : /var/opt/SUNWhadb/4 Windows : HADB_install_dir\device
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

啓動管理代理程式

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，請確定其在系統重新開機時會自動重新啓動。請參閱第 52 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」。
- 對於每種啓動方式，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。

將管理代理程式做為服務啓動

將管理代理程式做為服務啓動，可確保其繼續執行，除非系統關閉或您明確停止它。

在安裝了 Java Enterprise System 的 Solaris 或 Linux 上將管理代理程式做為服務啟動

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
```

在安裝了 Java Enterprise System 的 Windows 上將管理代理程式做為服務啟動

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：
命令：*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。

在安裝了獨立的應用程式伺服器的 Solaris 或 Linux 上將管理代理程式做為服務啟動

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
```

在安裝了獨立的應用程式伺服器的 Windows 上將管理代理程式做為服務啟動

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：
命令：*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 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，使用本小節中的程序來確保管理代理程式在 ma 程序失敗或作業系統重新開機時的可用性。對於生產部署，同樣適宜執行此作業。

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 上配置自動重新啟動

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，並在頂部查找類似於以下行的行：

```
id:5:initdefault:
```

本範例中為預設執行層級 5。

2 建立至檔案 `/etc/init.d/ma-initd` 的以下符號連結：

```
/etc/rc0.d/K20ma-initd  
/etc/rc1.d/K20ma-initd  
/etc/rc2.d/K20ma-initd  
/etc/rc3.d/S99ma-initd  
/etc/rc5.d/S99ma-initd  
/etc/rc5.d/K20ma-initd
```

3 重新開機。

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

▼ 在安裝了獨立應用程式伺服器的 Solaris 或 Linux 上配置自動重新啟動

- 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` 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 49 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 建立至 `/etc/init.d/ma-initd` 檔案的以下符號連結：

```
/etc/rc0.d/K20ma-initd
/etc/rc1.d/K20ma-initd
/etc/rc2.d/K20ma-initd
/etc/rc3.d/S99ma-initd
/etc/rc5.d/S99ma-initd
/etc/rcS.d/K20ma-initd
```

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

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手動啟動管理代理程式，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。

在安裝了 Java Enterprise System 的 Solaris 或 Linux 上的主控台模式下啟動管理代理程式

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。

在安裝了 Java Enterprise System 的 Windows 上的主控台模式下啓動管理代理程式

若要在主控台模式下啓動管理代理程式，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 49 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。

在安裝了獨立應用程式伺服器的 Windows 上的主控台模式下啓動管理代理程式

若要在主控台模式下啓動管理代理程式，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 49 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。

在安裝了獨立應用程式伺服器的 Solaris 或 Linux 上的主控台模式下啓動管理代理程式

若要在主控台模式下啓動 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 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 55 頁的「指令語法」
- 第 55 頁的「安全性選項」

- 第 56 頁的「一般選項」
- 第 57 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。大多數指令具有一個運算元 (通常為 `dbname`)，但有些沒有，有些具有兩個。

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：

```
HADBM_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，`--adminpassword` 具有優先順序。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：

```
HADB_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 60 頁的「建立資料庫」和第 59 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
<code>--adminpassword=<i>password</i></code> <code>-w</code>	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 <code>hadbm</code> 在資料庫或網域上作業時提供此密碼。 使用此選項或 <code>--adminpasswordfile</code> ，但不同時使用。
<code>--adminpasswordfile=<i>filepath</i></code> <code>-W</code>	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 <code>hadbm</code> 在資料庫或網域上作業時提供此密碼。 使用此選項或 <code>--adminpassword</code> ，但不同時使用。
<code>--no-adminauthentication</code> <code>-U</code>	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。
<code>--dbpassword=<i>password</i></code> <code>-P</code>	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 <code>hadbm</code> 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 <code>--dbpasswordfile</code> ，但不同時使用。
<code>--dbpasswordfile=<i>filepath</i></code> <code>-P</code>	指定包含 HADB 系統使用者密碼的檔案。使用此選項或 <code>--dbpassword</code> ，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。所有選項均爲布林選項 (預設值爲 `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。
--echo -e	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。
--agent= <i>URL</i> -m	管理代理程式的 URL。URL 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> 為管理代理程式在其上作業的連接埠號。 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

表 3-6 HADB 選項與環境變數 (續)

長形式	短形式	預設	環境變數
--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
--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 59 頁的「建立管理網域」
- 第 60 頁的「建立資料庫」
- 第 64 頁的「檢視和修改配置屬性」
- 第 69 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 56 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，會啓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 60 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` 成功執行，但 `hadbm create database` 失敗，`hadbm createdomain` 和 `hadbm registerpackage` 進行的變更將繼續。

▼ 建立資料庫

1 建立管理網域。

如需更多資訊，請參閱第 59 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 說明特殊的 `hadbm create` 指令選項。請參閱第 56 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> -a	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
--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 63 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 <i>vers</i> 為 HADB 版本編號。 預設由管理代理程式配置檔案中的 <code>ma.server.dbdevicepath</code> 指定。如需更多詳細資訊，請參閱第 49 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 63 頁的「指定裝置大小」。 遵循第 80 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： $(4 \times \text{LogbufferSize} + 16\text{MB}) / n$ 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 93 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 <code>ma.server.dbhistorypath</code> 指定。如需詳細資訊，請參閱第 49 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 62 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 <code>hadbm registerpackage</code> 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
--packagepath= <i>path-L</i>	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 <code>hadbm registerpackage</code> 指令在網域中註冊套裝軟體。	無
--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 <code>--hosts</code> 選項中指定的節點數。	0
--set= <i>attr-val-list</i> -S	以逗號分隔的資料庫配置屬性 (格式為 <code>name=value</code>) 清單。如需資料庫配置屬性說明，請參閱第 93 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 --dbpassword secret123
--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。確定同一機器上的節點並非鏡像節點。

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 32 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 93 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 32 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 Release Notes」。

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 65 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 65 頁的「配置屬性」。

請參閱第 56 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 65 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 56 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 85 頁的「取得 HADB 的狀態」。重新啟動 HADB (如第 77 頁的「重新啟動資料庫」中所述)。

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 60 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，但變更會僅對應用程式伺服器實例的 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
ConnectionTrace	如果為 true，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。	False	True 或 False
CoreFile	請勿變更預設值。	False	True 或 False
DatabaseName	資料庫名稱。	hadb	
DataBufferPoolSize	共用記憶體中配置的資料緩衝區池的大小。	200MB	16 - 2047 MB
DataDeviceSize	指定節點的裝置大小。如需有關建議的 DataDeviceSize 之資訊，請參閱第 63 頁的「指定裝置大小」 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： $(4 \times \text{LogbufferSize} + 16\text{MB}) / n$ 其中， n 為資料裝置數。	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	裝置位置：裝置為： <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) 	Solaris 和 Linux：/var/opt/SUNWhadb Windows：C:\Sun\AppServer\SUNWhadb\vers，其中 vers 為 HADB 版本編號。	
EagerSessionThreshold	確定使用的是一般還是急切閒置階段作業過期。 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。 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。	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

表 3-8 配置屬性 (續)

屬性	說明	預設	範圍
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>	
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	<p>資料庫的 JDBC 連線 URL。</p> <p>此為唯讀屬性。</p>	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	<p>HADB 節點使用的資料裝置數。</p> <p>此為唯讀屬性。</p>	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	<p>用於為不同 HADB 程序建立不同的連接埠號之基底連接埠號。</p> <p>此為唯讀屬性。</p>	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

表 3-8 配置屬性 (續)

屬性	說明	預設	範圍
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
StatInterval	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。 此為統計行的範例： Req-reply time: # 123, min= 69 avg= 1160 max= 9311 %=100.0 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)。百分號 (%) 後的數字為經由 StatInterval 在 15 毫秒內成功完成的作業事件數。	600 秒	0 - 600 秒
SyslogFacility	報告給 syslog 時使用的功能。應該配置 syslog 常駐程式 (請參閱 man syslogd.conf，以取得詳細資訊)。 使用在同一機器上執行的其他應用程式未使用的功能。 設定為 none 可停用 syslog 記錄。	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

表 3-8 配置屬性 (續)

屬性	說明	預設	範圍
SysLogLevel	儲存至作業系統 <code>syslog</code> 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 <code>syslog</code> 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 81 頁的「增加節點」。

本小節包含以下主題：

- 第 69 頁的「取得 JDBC URL」
- 第 70 頁的「建立連線池」
- 範例 3-5
- 第 71 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...` 部分做為表 3-10 中所述之 `serverList` 連線池特性的值。

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	<code>com.sun.hadb.jdbc.ds.HadbDataSource</code>
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	<code>meta-data</code>
表名稱	不指定
所有連線故障	取消核取/false
作業事件隔離	<code>repeatable-read</code>
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
<code>username</code>	用於 <code>asadmin create-session-store</code> 指令的 <code>storeuser</code> 名稱。
<code>password</code>	用於 <code>asadmin create-session-store</code> 指令的密碼 (<code>storepassword</code>)。
<code>serverList</code>	HADB 的 JDBC URL。若要確定此值，請參閱第 69 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 81 頁的「增加節點」。

表 3-10 HADB 連線池特性 (續)

特性	說明
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，這可確保連線是有效的。
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	<p>以下為階段作業持續性配置中的預設 JNDI 名稱：<code>jdbc/hastore</code>。可以使用預設名稱或其他名稱。</p> <p>在啓動可用性服務時，還必須指定此 JNDI 名稱為 <code>store-pool-jndi-name</code> Persistence Store 特性的值。</p>
池名稱	<p>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 32 頁的「配置網路備援」。</p>

表 3-11 HADB JDBC 資源設定 (續)

設定	說明
資料來源已啟用	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 72 頁的「管理網域」
- 第 73 頁的「管理節點」
- 第 75 頁的「管理資料庫」
- 第 79 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 59 頁的「建立管理網域」
- 第 72 頁的「延伸網域」
- 第 72 頁的「刪除網域」
- 第 73 頁的「列出網域中的主機」
- 第 73 頁的「從網域中移除主機」

請參閱第 55 頁的「安全性選項」和第 56 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 73 頁的「啟動節點」
- 第 74 頁的「停止節點」
- 第 75 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 77 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 56 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 56 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
```

重新啓動節點

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

若要重新啓動資料庫中的節點，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 運算元指定要重新啓動的節點數。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`)，可指定要啓動節點的層級。請參閱第 74 頁的「啓動層級選項」，以取得更多資訊。

請參閱第 56 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 重新啓動節點範例

```
hadbm restartnode 1
```

管理資料庫

可以在 HADB 資料庫上執行以下作業：

- 第 76 頁的「啓動資料庫」
- 第 76 頁的「停止資料庫」
- 第 77 頁的「重新啓動資料庫」
- 第 77 頁的「列出資料庫」
- 第 77 頁的「清除資料庫」
- 第 78 頁的「移除資料庫」

啓動資料庫

若要啓動資料庫，請使用 `hadbm start` 指令。此指令可啓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啓動時啓動。

指令語法爲：

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

dbname 運算元指定資料庫名稱。預設爲 `hadb`。

請參閱第 56 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 啓動資料庫範例

```
hadbm start
```

停止資料庫

在獨立的作業中停止和啓動資料庫時，資料在停止時不可用。若要使資料可用，可以如第 77 頁的「重新啓動資料庫」中所述重新啓動資料庫。

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，或將其配置爲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爲停止。如需有關資料庫狀態的更多資訊，請參閱第 85 頁的「取得 HADB 的狀態」。

若要停止資料庫，請使用 `hadbm stop` 指令。指令語法爲：

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

dbname 運算元指定資料庫名稱。預設爲 `hadb`。

請參閱第 56 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` 執行節點輪替式重新啟動：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=maurl]  
[--no-rolling]  
[dbname]
```

dbname 運算元指定資料庫名稱。預設為 `hadb`。

此指令具有一個特殊選項，`--no-rolling` (短形式 `-g`)，可指定立即重新啟動導致服務受到損失的所有節點。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狀態。

請參閱第 56 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 56 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 85 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 79 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 中的應用程式伺服器模式資料，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 56 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
<code>--fast</code>	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
<code>-F</code>		
<code>--spares= number</code>	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
<code>-s</code>		

例如：

```
hadbm clear --fast --spares=2 --dbpassword secret123
```

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 76 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 56 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)，則繼續。

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 77 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，例如您在伺服器記錄中仍能看到錯誤，則繼續執行作業。

3 刪除並重新建立資料庫。

請參閱第 78 頁的「移除資料庫」和第 60 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 80 頁的「增加現有節點的儲存空間」
- 第 81 頁的「增加機器」
- 第 81 頁的「增加節點」
- 第 83 頁的「重新分段資料庫」
- 第 84 頁的「透過重新建立資料庫增加節點」

另請參閱第 91 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 87 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 63 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 56 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 86 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。如需節點拓樸替代方案的說明，請參閱「Sun Java System Application Server Enterprise Edition 8.1 2005Q2 Deployment Planning Guide」中的第 3 章「[Selecting a Topology](#)」。

▼ 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 81 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 Enterprise Edition 8.1 2005Q2 Deployment Planning Guide」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 85 頁的「取得 HADB 的狀態」。

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 87 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 --dbpassword secret123 -adminpassword=
password --hosts n6,n7,n8,n9

```

下表說明特殊的 `hadbm addnodes` 指令選項。請參閱第 56 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
<code>--no-refragment</code>	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 83 頁的「重新分段資料庫」	不適用
<code>-r</code>	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 84 頁的「透過重新建立資料庫增加節點」	

表 3-13 hadbm addnodes 選項 (續)

選項	說明	預設
--spares= <i>number</i> -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= <i>path</i> -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 63 頁的「設定異質多工式裝置路徑」	Solaris 和 Linux : <i>HADB_install_dir</i> /device Windows : C:\Sun\AppServer\SUNWhadb\vers，其中 vers 為 HADB 版本編號。
--hosts= <i>hostlist</i> -H	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。 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，則新的備援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 32 頁的「配置網路備援」。	無

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 85 頁的「取得 HADB 的狀態」。

請參閱第 56 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 87 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 84 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 --dbpassword secret123
```

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 76 頁的「停止資料庫」中所述停止資料庫。
- 3 如第 78 頁的「移除資料庫」中所述刪除資料庫。
- 4 如第 60 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 69 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。

監視 HADB

可以透過以下方式監視 HADB 的活動：

- 第 85 頁的「取得 HADB 的狀態」
- 第 87 頁的「取得裝置資訊」
- 第 89 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。如需有關解譯 HADB 資訊的資訊，請參閱「Sun Java System Application Server Enterprise Edition 8.1 2005Q2 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 86 頁的「節點狀態」。請參閱第 56 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啓動並在執行。
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 77 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 86 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 87 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 86 頁的「節點角色」
- 第 87 頁的「節點狀態」

節點角色

在建立節點期間爲其指定角色，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啓動修復程序。

- **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 74 頁的「停止節點」、第 73 頁的「啟動節點」和第 77 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 56 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 Enterprise Edition 8.1 2005Q2 Performance Tuning Guide」中的「Tuning HADB」。

指令語法為：

```
hadbm resourceinfo [--databuf] [--locks] [--logbuf] [--nilogbuf]
[--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 56 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 89 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 90 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 90 頁的「記錄緩衝區資訊」，以取得更多資訊。
--nilogbuf	顯示節點內部記錄緩衝區資訊。
-n	請參閱以下第 91 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：自資料庫啟動至今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，因為讀取作業以 `repeatable read` 隔離層級執行，`delete`、`insert` 和 `update` 作業使用僅在作業事件終止後釋放的鎖定。

如要變更 `NumberOfLocks`，請參閱第 93 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，鏡像節點與其所鏡像之節點位於不同的 DRU 上，如「Sun Java System Application Server Enterprise Edition 8.1 2005Q2 Deployment Planning Guide」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，取決於您需要在在一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 做為 Windows 服務或在 `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 啓動機器上的所有節點。
如需更多資訊，請參閱第 73 頁的「啓動節點」。
- 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 85 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 中的每台備援機器，請逐台重複單一機器程序，如第 91 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 91 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 76 頁的「停止資料庫」。
- 2 執行維護程序，並使所有機器啓動和執行。
- 3 確定 ma 正在執行。
- 4 啓動 HADB。
如需更多資訊，請參閱第 76 頁的「啓動資料庫」。
完成最後一步後，HADB 資料再次變得可用。

▼ 發生故障時執行未規劃的維護

- 檢查資料庫狀態。
請參閱第 85 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。首先啓動並執行所有機器。然後清除 HADB，並重新建立階段作業存放區。請參閱第 77 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 56 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 60 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。
- 節點停止或啓動時，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。

配置負載平衡和容錯移轉

本小節說明 HTTP 負載平衡外掛程式。其中包含以下主題：

- 第 95 頁的「負載平衡器的工作方式」
- 第 97 頁的「設定 HTTP 負載平衡」
- 第 99 頁的「配置 Web 伺服器以用於負載平衡」
- 第 108 頁的「配置負載平衡器」
- 第 114 頁的「配置 HTTP 和 HTTPS 容錯移轉」
- 第 115 頁的「在不遺失可用性的情況上升級應用程式」

負載平衡器的工作方式

負載平衡器嘗試在多個應用程式伺服器實例（獨立或叢集）之間平均分散工作負荷量，從而增加了系統的總流量。

使用負載平衡器還可以滿足從一個伺服器實例容錯移轉到另一個伺服器實例的請求。對於要持續的 HTTP 階段作業資訊，請配置 HTTP 階段作業持續性。如需更多資訊，請參閱第 8 章。

如需有關配置負載平衡的完整說明，請參閱「Sun Java System Application Server 高可用性管理指南」。

使用 `asadmin` 工具而非管理主控台配置 HTTP 負載平衡。

- 第 96 頁的「指定的請求與未指定的請求」
- 第 96 頁的「HTTP 負載平衡演算法」
- 第 96 頁的「應用程式範例」

另請參閱：

- 第 97 頁的「設定負載平衡的先決條件」
- 第 96 頁的「指定的請求與未指定的請求」
- 第 96 頁的「HTTP 負載平衡演算法」

- 第 98 頁的「設定負載平衡的程序」

指定的請求與未指定的請求

首次從 HTTP 用戶端傳入負載平衡器的請求為新階段作業的請求。新階段作業的請求稱為**未指定的**請求。負載平衡器會根據循環演算法將此請求路由到叢集中的應用程式伺服器實例。

在某個應用程式伺服器實例中建立階段作業後，負載平衡器會將此階段作業的所有後續請求都路由到該特定實例而且僅路由到該實例。現有階段作業的請求稱為**指定的**請求或**居留式**請求。

HTTP 負載平衡演算法

Sun Java System Application Server 負載平衡器使用**居留式循環演算法**對內送 HTTP 和 HTTPS 請求進行負載平衡。給定階段作業的所有請求都將會傳送到同一個應用程式伺服器實例。使用居留式負載平衡器，階段作業資料將快取在單一應用程式伺服器上，而不會分布到叢集中的所有實例。

因此，居留式循環方案能夠帶來顯著的效能優勢，這種優勢通常超過了使用純循環方案所獲得的使負載更加平均分佈的優勢。

新 HTTP 請求傳送到負載平衡外掛程式後，系統將基於簡單的循環方案將該請求轉寄至某個應用程式伺服器實例。隨後，透過使用 Cookie 或明確的 URL 重新寫入將該請求「居留」在此特定的應用程式伺服器實例上。

從居留式資訊中，負載平衡器外掛程式將首先確定請求先前轉寄到的實例。如果發現該實例正常工作，負載平衡器外掛程式會將請求轉寄至該特定應用程式伺服器實例。因此，給定階段作業的所有請求都將會傳送到同一個應用程式伺服器實例。

負載平衡器外掛程式使用以下方法來確定階段作業居留性：

- **Cookie 方法**：負載平衡器外掛程式使用一個單獨的 Cookie 來記錄路由資訊。若要使用基於 Cookie 的方法，HTTP 用戶端必須支援 Cookie。
- **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。

應用程式範例

以下目錄包含用於演示負載平衡和容錯移轉的範例應用程式：

```
install_dir/samples/ee-samples/highavailability  
install_dir/samples/ee-samples/failover
```

ee-samples 目錄還包含設定環境以執行範例的資訊。

設定 HTTP 負載平衡

本小節說明如何設定負載平衡外掛程式，其中包含以下小節：

- 第 97 頁的「設定負載平衡的先決條件」
- 第 97 頁的「HTTP 負載平衡器部署」
- 第 98 頁的「設定負載平衡的程序」

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 Web 伺服器。
- 安裝負載平衡外掛程式。
如需有關安裝程序的資訊，請參閱 Sun Java System Application Server 安裝指南 (如果使用獨立應用程式伺服器) 或 *Sun Java Enterprise System 安裝指南* (如果使用 Java Enterprise System)。
- 配置 Web 伺服器。如需更多資訊，請參閱第 99 頁的「配置 Web 伺服器以用於負載平衡」。
- 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。

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 97 頁的「使用叢集伺服器實例」
- 第 98 頁的「將單一獨立的實例與用做反向代理外掛程式的負載平衡器配合使用」
- 第 98 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。

如果您有多個叢集，則請求僅會在單一叢集的實例之間進行負載平衡和容錯移轉。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 115 頁的「在不遺失可用性的情況下升級應用程式」。

將單一獨立的實例與用做反向代理外掛程式的負載平衡器配合使用

您也可以將負載平衡器配置為使用獨立伺服器實例，而非叢集。此配置可讓負載平衡外掛程式做為反向代理外掛程式(有時稱為通道外掛程式)工作。當 Web 伺服器接收到在負載平衡器中啓用的應用程式之請求時，會將請求直接轉寄至應用程式伺服器。

為通道外掛程式配置負載平衡器的程序與為伺服器實例的叢集配置負載平衡器的程序相同。

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。

設定負載平衡的程序

使用 `asadmin` 工具可在您的環境中配置負載平衡。如需有關在這些步驟中所使用的 `asadmin` 指令的更多資訊，請參閱第 108 頁的「配置負載平衡器」。

▼ 設定負載平衡

- 1 使用 `asadmin` 指令 `create-http-lb-config` 建立負載平衡器配置。
- 2 使用 `asadmin create-http-lb-ref` 為要管理的負載平衡器增加對叢集或獨立伺服器實例的參照。
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 使用 `asadmin enable-http-lb-server` 啓用負載平衡器參照的叢集或獨立伺服器實例。
- 4 使用 `asadmin enable-http-lb-application` 啓用要用於負載平衡的應用程式。
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以使用這些應用程式是兩個獨立的步驟。
- 5 使用 `asadmin create-health-checker` 建立運作狀態檢查程式。
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。
- 6 使用 `asadmin export-http-lb-config` 產生負載平衡器配置檔案。
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。

- 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。

配置 Web 伺服器以用於負載平衡

負載平衡外掛程式安裝程式將對 Web 伺服器的配置檔案進行一些修改。所作的變更取決於該 Web 伺服器。

備註 - 負載平衡器外掛程式可隨 Sun Java System Application Server Enterprise Edition 一起安裝，也可在執行支援的 Web 伺服器的機器上單獨安裝。如需有關安裝程序的完整詳細資訊，請參閱「Sun Java System Application Server Enterprise Edition 8.1 2005Q2 Installation Guide」中的第 1 章「Installing Application Server Software」(如果使用獨立的 Application Server) 或「Sun Java Enterprise System 2005Q5 Installation Guide」(如果使用 Java Enterprise System)。

- 第 99 頁的「對 Sun Java System Web Server 的修改」
- 第 100 頁的「使用 Apache Web Server」
- 第 38 頁的「安裝」
- 第 107 頁的「配置多個 Web 伺服器實例」

對 Sun Java System Web Server 的修改

安裝程式會將以下項目增加至 Sun Java System Web Server 的配置檔案：

對於 Web 伺服器實例的 magnus.conf 檔案，會增加：

```
##EE lb-pluginInit
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 addition for EE lb-plugin
```

對於 Web 伺服器實例的 obj.conf 檔案，會增加：

```
<Object name=default>
NameTrans fn="name-trans-passthrough" name="lbplugin"
config-file="web_server_install_dir/web_server_instance/config/loadbalancer.xml"
<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>
```

在上述程式碼中，`lbplugin` 是 Object 的唯一識別名稱；`web_server_install_dir/web_server_instance/config/loadbalancer.xml` 是虛擬伺服器（負載平衡器即配置要在此執行）的 XML 配置檔案位置。

安裝之後，請依循第 97 頁的「設定 HTTP 負載平衡」中的說明配置負載平衡器。

使用 Apache Web Server

若要使用 Apache Web Server，您必須在安裝負載平衡外掛程式之前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 - 在 Apache 1.3 上，當多個 Apache 子程序執行時，每個程序都有自己的負載平衡循環序列。例如，如果有兩個 Apache 子程序正在執行，並且負載平衡器外掛程式在兩個應用程式伺服器實例上進行負載平衡，則第一個請求將傳送至實例 #1，第二個請求也將傳送至實例 #1。而第三個請求將傳送至實例 #2，第四個請求也將傳送至實例 #2。此模式會不斷重複（實例 1、實例 1、實例 2、實例 2 等）。此運作方式可能與您預期的運作方式（即，實例 1、實例 2、實例 1、實例 2 等）不同。在 Sun Java System Application Server 中，用於 Apache 的負載平衡器外掛程式將為每個 Apache 程序創設一個負載平衡器實例，從而建立獨立的負載平衡順序。

如果使用 `--with-mpm=worker` 選項進行編譯，則 Apache 2.0 將具有多重執行緒運作方式。

-
- 第 100 頁的「使用 Apache Web Server 的需求」
 - 第 102 頁的「安裝負載平衡外掛程式之前的配置」
 - 第 104 頁的「應用程式伺服器安裝程式所做的修改」
 - 第 104 頁的「配置 Apache 安全性檔案以與負載平衡器一起工作」

使用 Apache Web Server 的需求

對於 Apache Web Server，您的安裝必須滿足最小需求（視 Apache 版本而定）。

適用於 Apache 1.3 的需求

對於 Apache 1.3，負載平衡外掛程式需要：

- `openssl-0.9.7e` (原始碼)
- `mod_ssl-2.8.16-1.3.x` (原始碼)，其中 *x* 表示 Apache 的版本。mod_ssl 的版本必須與 Apache 的版本相符。

- gcc-3.3-sol9-sparc-local 套裝軟體 (適用於 Solaris SPARC)
 - gcc-3.3-sol9-intel-local 套裝軟體 (適用於 Solaris x86)
 - flex-2.5.4a-sol9-sparc-local 套裝軟體 (適用於 Solaris SPARC)
 - flex-2.5.4a-sol9-intel-local 套裝軟體 (適用於 Solaris x86)
- 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 作業系統上，確保 gcc 3.3 版和 make 均位於 PATH 中，並且已安裝 flex。
 - 在 Solaris 10 作業系統上，對 OpenSSL 執行 make 之前，請先執行位於 /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 - 若要使用 gcc 以外的 C 編譯器，請設定 C 編譯器的路徑並使公用程式位於 PATH 環境變數中。例如，對於 sh shell：export LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:appserver_installdir/lib

適用於 Apache 2 的最低需求

對於 Apache 2.0，負載平衡外掛程式需要：

- openssl-0.9.7e (原始碼)
- httpd-2.0.49 (原始碼)
- gcc-3.3-sol9-sparc-local 套裝軟體 (適用於 Solaris SPARC)。
- gcc-3.3-sol9-intel-local 套裝軟體 (適用於 Solaris x86)
- flex-2.5.4a-sol9-sparc-local 套裝軟體 (適用於 Solaris SPARC)
- flex-2.5.4a-sol9-intel-local 套裝軟體 (適用於 Solaris x86)

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 作業系統上，確保 gcc 3.3 版和 make 均位於 PATH 中，並且已安裝 flex。
- 在 Solaris 10 作業系統上，對 OpenSSL 執行 make 之前，請先執行位於 /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 - 若要使用 gcc 以外的 C 編譯器，請設定 C 編譯器的路徑並使公用程式位於 PATH 環境變數中。例如，對於 sh shell：export

`LD_LIBRARY_PATH=app_server_install_dir/lib:$LD_LIBRARY_PATH`。

安裝負載平衡外掛程式之前的配置

為 Apache 安裝負載平衡外掛程式之前，請先安裝 Apache Web Server。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。

▼ 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 下載和解壓縮 OpenSSL 原始碼。

2 編譯和建置 OpenSSL。

如果已安裝了 OpenSSL 0.9.7.e，則在 Linux 平台上無需執行此步驟。

輸入以下指令：

```
cd openssl-0.9.7e
make
make install
```

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

3 依據 Apache 的版本，請執行以下程序之一：

■ 對於 Apache 1.3，請依循以下步驟使用 mod_ssl 配置 Apache：

a. 解壓縮 mod_ssl 原始碼。

b. `cd mod_ssl-2.8.14-1.3.x`

c. `./configure --with-apache=../apache_1.3.x --with-ssl=../openssl-0.9.7e
--prefix=install_path --enable-module=ssl --enable-shared=ssl
--enable-rule=SHARED_CORE --enable-module=so`

在上述指令中，*x* 為 Apache 的版本編號，*install_path* 為要在其中安裝 Apache 的目錄。

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

- 對於 Apache 2.0，配置原始碼樹狀結構：
 - a. `cd http-2.0_x`。
 - b. 執行 `./configure --with-ssl= open_ssl_install_path --prefix= 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，*install_path* 為要在其中安裝 Apache 的目錄。

- 4 對於在 Linux 2.1 上執行的 Apache，編譯之前請：
 - a. 開啓 `src/Makefile` 並尋找自動產生的區段的結束位置。
 - b. 在自動產生的區段之後的前四行的後面增加以下行：


```
LIBS+= -licuuc -licui18n -lnspr4 -lpthread -lxerces-c
-lsupport -lnsprwrap -lns-httpd40
LDFLAGS+= -L/appserver_installdir/lib -L/opt/sun/private/lib
```

 請注意，只有將應用程式伺服器安裝至 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/appserver_installdir /lib -L/opt/sun/private/lib
```
 - c. 設定環境變數 `LD_LIBRARY_PATH`。
 對於所有安裝，請將其設定為：`appserver_install_dir/lib`
 針對 Java Enterprise System 安裝，請將其設定為
`appserver_install_dir/lib:opt/sun/private/lib`。

- 5 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 certificate (僅適用於 Apache 1.3)

c. make install

指令 `make certificate` 需要安全密碼。請記住此密碼，因為啟動安全 Apache 需要它。

6 為您的環境配置 Apache。

應用程式伺服器安裝程式所做的修改

負載平衡外掛程式安裝程式將必要檔案擷取至 Web 伺服器根目錄中的目錄：

- 對於 Apache 1.3，目錄為 `libexec`。
- 對於 Apache 2.0，目錄為 `modules`。

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

```
<VirtualHost machine_name:443>
##Addition for EE lb-plugin
LoadFile /usr/lib/libCstd.so.1
LoadModule apachelbplugin_module libexec/mod_loadbalancer.so
##AddModule mod_apachelbplugin.cpp
<IfModule mod_apachelbplugin.cpp>
    config-file webserver_instance/conf/loadbalancer.xml
    locale en
</IfModule>
<VirtualHost machine_ip_address>
    DocumentRoot "webserver_instance/htdocs"
    ServerName server_name
</VirtualHost>
##END EE LB Plugin ParametersVersion 7
```

▼ 配置 Apache 安全性檔案以與負載平衡器一起工作

Apache Web Server 必須具有正確的安全性檔案才能與負載平衡外掛程式一起正常工作。

- 1 在 `apache_install_dir` 下建立一個名為 `sec_db_files` 的目錄。
- 2 將 `application_server_domain_dir/config/*.db` 複製到 `apache_install_dir/sec_db_files`。
- 3 視平台而定，執行其他配置。
 - 在 Solaris 平台上：

將路徑 `/usr/lib/mps/secv1` 增加至 `apache_install_dir/bin/apachectl` 程序檔中的 `LD_LIBRARY_PATH` 中。此路徑必須增加在 `/usr/lib/mps` 的前面。

- 在 Linux 系統中：
 - 將路徑 `/opt/sun/private/lib` 增加至 `apache_install_dir/bin/apachectl` 程序檔中的 `LD_LIBRARY_PATH` 中。此路徑必須增加在 `/usr/lib` 的前面。
- 在 Microsoft Windows 上：
 - a. 將新路徑增加至 Path 環境變數。
 - 按一下 [開啓] -> [設定] -> [控制台] -> [系統] -> [進階] -> [環境變數] -> [系統變數]。
 - 將 `application_server_install_dir/bin` 增加至 Path 環境變數。
 - b. 將環境變數 `NSPR_NATIVE_THREADS_ONLY` 設定為 1。
 - 在 [環境變數] 視窗中的 [系統變數] 下，按一下 [新增]。輸入變數名稱 `NSPR_NATIVE_THREADS_ONLY` 以及變數值 1。
 - c. 重新啓動機器。

對 Microsoft IIS 的修改

若要配置 Microsoft 網際網路資訊服務 (IIS)，以使用負載平衡外掛程式，請在 Windows 網際網路服務管理員中修改某些特性。網際網路服務管理程式位於 [控制台] 資料夾內的 [管理工具] 資料夾中。

安裝 Sun Java System Application Server 之後請進行以下修改。

▼ 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 檔案的路徑和 *application_server_install_dir/bin* 增加至系統的 PATH 環境變數。
- 7 重新啓動機器。
- 8 停止然後啓動 Web 伺服器以使新設定生效。
若要停止 Web 伺服器，請在此網站上按一下滑鼠右鍵，然後選取 [Stop]。若要啓動 Web 伺服器，請在此網站上按一下滑鼠右鍵，然後選取 [Start]。
- 9 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://webserver_name/web_application，其中，*webserver_name* 為 Web 伺服器的主機名稱或 IP 位址，*web_application* 為在 C:\Inetpub\wwwroot\sun-passthrough\sun-passthrough.properties 檔案中列示的環境根目錄。

自動配置的 Sun-passthrough 特性

安裝程式將自動配置 sun-passthrough.properties 中的下列特性。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

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外掛程式。

請執行第 99 頁的「對 Sun Java System Web Server 的修改」、第 100 頁的「使用 Apache Web Server」或第 38 頁的「安裝」中的步驟。

2 複製 DTD 檔案。

從現有 Web 伺服器實例的 config 目錄中，將 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 108 頁的「建立 HTTP 負載平衡器配置」。

配置負載平衡器

負載平衡器配置為 `domain.xml` 檔案中已命名的配置。負載平衡器配置非常靈活：

- 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。
- 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。

本小節說明如何建立、修改和使用負載平衡器配置，其中包含以下主題：

- 第 108 頁的「建立 HTTP 負載平衡器配置」
- 第 109 頁的「建立 HTTP 負載平衡器參照」
- 第 109 頁的「啓用用於負載平衡的伺服器實例」
- 第 109 頁的「啓用用於負載平衡的應用程式」
- 第 109 頁的「建立 HTTP 運作狀態檢查程式」
- 第 111 頁的「匯出負載平衡器配置檔案」
- 第 112 頁的「變更負載平衡器配置」
- 第 112 頁的「啓用動態重新配置」
- 第 112 頁的「停用 (靜止) 伺服器實例或叢集」
- 第 113 頁的「停用 (靜止) 應用程式」

建立 HTTP 負載平衡器配置

使用 `asadmin` 指令 `create-http-lb-config` 建立負載平衡器配置。第 108 頁的「建立 HTTP 負載平衡器配置」說明負載平衡器參數。如需更多資訊，請參閱 `create-http-lb-config`、`delete-http-lb-config` 和 `list-http-lb-configs` 的文件。

表 4-1 負載平衡器配置參數

參數	說明
<code>response timeout</code>	伺服器實例必須傳回回應的時間段 (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 114 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> 有無變更的檢查時間間隔。當檢查偵測到變更時，系統將重新載入配置檔案。將值設定為 0 可停用重新載入。如需更多資訊，請參閱第 112 頁的「啓用動態重新配置」。
<code>monitor</code>	是否對負載平衡器啓用監視功能。
<code>routecookie</code>	負載平衡外掛程式用於記錄路由資訊的 Cookie 的名稱。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 <code>JROUTE</code> 。
<code>target</code>	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。仍需要先啓用所參照的伺服器或叢集 (使用 `enable-http-lb-server`)，然後才能對向該伺服器或叢集發出的請求執行負載平衡。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。

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`。若要刪除某個參照，必須先使用 `disable-http-lb-server` 停用所參照的伺服器或叢集。

如需更多資訊，請參閱 `create-http-lb-ref` 和 `delete-http-lb-ref` 的文件。

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。

如需更多資訊，請參閱 `enable-http-lb-server` 的文件。

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相同應用程式集。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。

如需更多資訊，請參閱 `enable-http-lb-application` 的線上說明。

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。

負載平衡器的運作狀態檢查機制使用 HTTP 與應用程式伺服器實例進行通訊。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。

建立運作狀態檢查程式

若要建立運作狀態檢查程式，請使用 `asadmin create-http-health-checker` 指令。指定下列參數：

表 4-2 運作狀態檢查程式參數

參數	說明	預設
<code>url</code>	指定負載平衡器檢查的偵聽程式之 URL 以確定其運作狀態。	<code>/</code>
<code>interval</code>	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
<code>timeout</code>	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

如果應用程式伺服器實例標示為異常，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 來檢查所有異常的應用程式伺服器實例，以確定這些異常的應用程式伺服器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。

如需更多資訊，請參閱 `create-http-health-checker` 和 `delete-http-health-checker` 的文件。

正常實例的額外運作狀態檢查特性

`create-http-health-checker` 建立的運作狀態檢查程式僅檢查異常實例。若要定期檢查正常實例，請在匯出的 `loadbalancer.xml` 檔案中設定幾個額外特性。

備註 – 僅在您已將 `loadbalancer.xml` 匯出之後，才可透過手動對其進行編輯，來設定這些特性。沒有等效的 `asadmin` 指令可以使用。

若要檢查正常的實例，請設定以下特性：

表 4-3 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"/>
```

如果您增加這些特性，並隨後再次編輯並匯出 `loadbalancer.xml` 檔案，您必須重新將這些特性增加至該檔案，因為新匯出的配置不包含這些特性。

匯出負載平衡器配置檔案

Sun Java System Application Server 隨附的負載平衡外掛程式使用名為 `loadbalancer.xml` 的配置檔案。使用 `asadmin` 工具可以在 `domain.xml` 檔案中建立負載平衡器配置。配置負載平衡環境後，請將其匯出成檔案。

▼ 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。

匯出用於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。如果您未指定檔名，則該檔案會命名為 `loadbalancer.xml.load_balancer_config_name`。如果您未指定路徑，則該檔案會建立在 `application_server_install_dir/domains/domain_name/generated` 目錄中。

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，該位置可能為 `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 111 頁的「匯出負載平衡器配置檔案」。

負載平衡器外掛程式將基於在負載平衡器配置中指定的重新載入間隔定期檢查已更新的配置。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，負載平衡外掛程式會定期檢查已更新的配置。

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-config` 配合使用。
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–如果嘗試進行自身重新配置時，負載平衡器遇到了硬碟讀取錯誤，它將使用記憶體中的目前配置。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/webserver_instance/logs/`。

停用 (靜止) 伺服器實例或叢集

在因任何原因而停止應用程式伺服器之前，您希望實例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應用程式伺服器實例：

- 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)。
- 2 使用 `asadmin export-http-lb-config` 匯出負載平衡器配置檔案。
- 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，您希望應用程式完成處理請求。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。

▼ 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。
- 2 使用 `asadmin export-http-lb-config` 匯出負載平衡器配置檔案。
- 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應用程式伺服器實例變為不可用，負載平衡器外掛程式會將這些階段作業容錯移轉到其他應用程式伺服器實例上。本小節介紹如何配置負載平衡器外掛程式以啓用 HTTP/HTTPS 路由和階段作業容錯移轉。

本小節論述以下主題：

- 第 114 頁的「HTTPS 路由」
- 第 115 頁的「配置等幕 URL」

HTTPS 路由

HTTP 安全 (HTTPS) 通訊協定使用安全套接字層 (SSL) 來提供對 HTTP 請求的解密和加密，以實現安全通訊。若要使 HTTPS 路由工作，必須配置一個或多個 HTTPS 偵聽程式。

負載平衡外掛程式將所有內送的 HTTP 或 HTTPS 請求路由至應用程式伺服器實例。然而，如果啓用了 HTTPS 路由，則負載平衡外掛程式會將 HTTPS 請求僅轉寄給使用 HTTPS 連接埠的應用程式伺服器。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 連接埠的可用應用程式伺服器實例，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 請求將路由到在此 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。建立新的負載平衡器配置時，請將此選項設定為 `true`，或者以後使用 `asadmin set` 指令變更此選項的設定。

備註 - 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，並啓動新的階段作業。

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，則必須將應用程式伺服器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，則必須將應用程式伺服器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。

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應用程式伺服器實例上，將環境配置為重試失敗的等冪 HTTP 請求。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 Enterprise Edition 8.1 2005Q2 Developer's Guide」中的「Configuring Idempotent URL Requests」。

在不遺失可用性的情況下升級應用程式

在不遺失對使用者的可用性的情況下，將應用程式升級至新版本稱為**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 116 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 119 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。

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操作：

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，網域將自動成爲目標。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` 為實例啓用已重新部署的應用程式。
- 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。
 - b. 使用 `asadmin export-http-lb-config` 匯出負載平衡器配置檔案。
 - 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，其位置爲 `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` 啓用了動態重新配置。
 - 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，並立即重新啓動伺服器。
- 6 在叢集中的其他實例仍處於執行狀態的情況下，重新啓動已停用的伺服器實例。
重新啓動操作將使伺服器與網域同步，並更新應用程式。
- 7 測試重新啓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啓用伺服器實例。
請依照下列步驟執行：
 - a. 使用 `asadmin enable-http-lb-server` 啓用伺服器實例。
 - b. 使用 `asadmin export-http-lb-config` 匯出負載平衡器配置檔案。
 - c. 將配置檔案複製到 Web 伺服器的配置目錄，如第 116 頁的「在單一叢集中升級」的第 116 頁的「在單一叢集中升級」中所述。
- 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，您可以為叢集重新啓用動態重新配置。

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。

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操作：

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，網域將自動成爲目標。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。

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。

b. 使用 `asadmin export-http-lb-config` 匯出負載平衡器配置檔案。

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，其位置爲

`web_server_install_dir/https-host-name/config/loadbalancer.xml`。必須爲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 在負載平衡器中重新啟用叢集：
 - a. 使用 `asadmin enable-http-lb-server` 啟用叢集。
 - b. 使用 `asadmin export-http-lb-config` 匯出負載平衡器配置檔案。
 - 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，您可以為所有叢集重新啟用動態重新配置。

升級不相容的應用程式

如需有關應用程式相容條件的資訊，請參閱第 115 頁的「應用程式相容性」，應用程式的新版本與其舊版本不相容。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轉移至新叢集。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。

- 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。
 - 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。
 - 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。
 - d. 使用 `asadmin configure-ha-cluster` 將叢集配置為具有高可用性。
 - 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。
- 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。
- 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` 從負載平衡器上停用舊叢集。
- 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` 從負載平衡器中啟用新叢集。
- 11 使用 `asadmin export-http-lb-config` 匯出負載平衡器配置檔案。
- 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，其位置為 `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 121 頁的「配置記錄訊息」
- 第 121 頁的「記錄訊息類型」
- 第 122 頁的「啓用負載平衡器記錄」
- 第 123 頁的「瞭解監視訊息」

配置記錄訊息

負載平衡外掛程式使用 Web 伺服器的記錄機制寫入記錄訊息。應用程式伺服器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。應用程式伺服器記錄層級 (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

負載平衡外掛程式會產生以下類型的記錄訊息：

- 第 121 頁的「負載平衡器配置器記錄訊息」
- 第 122 頁的「請求派送和執行階段記錄訊息」
- 第 122 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。

- 每個方法上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>
- 每個方法上的路由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/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 – 啟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，如果託管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**
在伺服器的管理主控台中，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 00:00:00 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>(in case 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，處理請求的來回總時間 (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

使用應用程式伺服器叢集

本章說明如何使用應用程式伺服器叢集。它包含以下小節：

- 第 125 頁的「叢集簡介」
- 第 125 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。

使用叢集

- 第 126 頁的「建立叢集」
- 第 127 頁的「為叢集建立伺服器實例」
- 第 128 頁的「配置叢集」
- 第 131 頁的「刪除叢集」
- 第 129 頁的「配置叢集中的伺服器實例」
- 第 129 頁的「為叢集配置應用程式」
- 第 130 頁的「為叢集配置資源」
- 第 131 頁的「遷移 EJB 計時器」
- 第 132 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。
 - 若要建立不使用共用配置的叢集，請選擇 default-config。
保留選取標有「複製已選取的配置」的單選按鈕。預設配置的副本名稱將為 *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 152 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 128 頁的「配置叢集」
 - 第 127 頁的「為叢集建立伺服器實例」
 - 第 129 頁的「為叢集配置應用程式」
 - 第 130 頁的「為叢集配置資源」
 - 第 131 頁的「刪除叢集」
 - 第 132 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 143 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 152 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 141 頁的「何為節點代理程式？」
 - 第 126 頁的「建立叢集」
 - 第 128 頁的「配置叢集」
 - 第 129 頁的「為叢集配置應用程式」
 - 第 130 頁的「為叢集配置資源」
 - 第 131 頁的「刪除叢集」

- 第 132 頁的「升級元件而不會使服務受到損失」
- 第 129 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 126 頁的「建立叢集」
 - 第 127 頁的「為叢集建立伺服器實例」
 - 第 129 頁的「為叢集配置應用程式」
 - 第 130 頁的「為叢集配置資源」
 - 第 131 頁的「刪除叢集」
 - 第 132 頁的「升級元件而不會使服務受到損失」
 - 第 131 頁的「遷移 EJB 計時器」

▼ 啓動、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]、[啓動]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 [啓動實例] 以啓動實例。
 - 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 126 頁的「建立叢集」](#)
- [第 128 頁的「配置叢集」](#)
- [第 127 頁的「為叢集建立伺服器實例」](#)
- [第 129 頁的「為叢集配置應用程式」](#)
- [第 130 頁的「為叢集配置資源」](#)
- [第 131 頁的「刪除叢集」](#)
- [第 132 頁的「升級元件而不會使服務受到損失」](#)
- 「Sun Java System Application Server Enterprise Edition 8.1 2005Q2 管理指南」中的「恢復作業事件」

▼ 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 126 頁的「建立叢集」
- 第 128 頁的「配置叢集」
- 第 127 頁的「為叢集建立伺服器實例」
- 第 130 頁的「為叢集配置資源」
- 第 131 頁的「刪除叢集」
- 第 132 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 126 頁的「建立叢集」
- 第 128 頁的「配置叢集」
- 第 127 頁的「為叢集建立伺服器實例」
- 第 129 頁的「為叢集配置應用程式」
- 第 131 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 126 頁的「建立叢集」
 - 第 128 頁的「配置叢集」
 - 第 127 頁的「為叢集建立伺服器實例」
 - 第 129 頁的「為叢集配置應用程式」
 - 第 130 頁的「為叢集配置資源」
 - 第 132 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 128 頁的「配置叢集」
- 「Sun Java System Application Server Enterprise Edition 8.1 2005Q2 管理指南」中的「配置 EJB 計時器服務設定」

▼ 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
- 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 [啟動叢集] 按鈕來啟動叢集。
- 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 126 頁的「建立叢集」
- 第 128 頁的「配置叢集」
- 第 127 頁的「為叢集建立伺服器實例」
- 第 129 頁的「為叢集配置應用程式」

- 第 130 頁的「為叢集配置資源」
- 第 131 頁的「刪除叢集」

管理已命名的配置

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 135 頁的「關於已命名的配置」
- 第 137 頁的「使用已命名的配置」

關於已命名的配置

- 第 135 頁的「已命名的配置」
- 第 136 頁的「default-config 配置」
- 第 136 頁的「建立實例或叢集時建立的配置」
- 第 137 頁的「唯一連接埠號和配置」

已命名的配置

已命名的配置是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。

配置是在管理網域中建立的。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。

由於已命名的配置包含如此多的必需配置設定，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 136 頁的「[default-config 配置](#)」
 - 第 136 頁的「[建立實例或叢集時建立的配置](#)」
 - 第 137 頁的「[唯一連接埠號和配置](#)」
 - 第 138 頁的「[建立已命名的配置](#)」
 - 第 138 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。非叢集伺服器實例或叢集不允許參照 `default-config` 配置；只能對其進行複製以建立新配置。編輯預設配置，以確定從預設配置複製而來的新配置具有正確的初始設定。

如需更多資訊，請參閱：

- 第 136 頁的「[建立實例或叢集時建立的配置](#)」
- 第 135 頁的「[已命名的配置](#)」
- 第 138 頁的「[建立已命名的配置](#)」
- 第 138 頁的「[編輯已命名的配置的特性](#)」
- 第 139 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，新配置的名稱為 `instance_name-config`。對於叢集，新配置的名稱為 `cluster_name-config`。

如需更多資訊，請參閱：

- 第 136 頁的「default-config 配置」
- 第 135 頁的「已命名的配置」
- 第 138 頁的「建立已命名的配置」
- 第 138 頁的「編輯已命名的配置的特性」

唯一連接埠號和配置

如果同一主機上的多個實例參照相同的配置，則每個實例偵聽的連接埠號均不得重複。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 139 頁的「編輯參照配置之實例的連接埠號」
- 第 138 頁的「編輯已命名的配置的特性」
- 第 135 頁的「已命名的配置」

使用已命名的配置

- 第 138 頁的「建立已命名的配置」
- 第 138 頁的「編輯已命名的配置的特性」
- 第 139 頁的「編輯參照配置之實例的連接埠號」
- 第 140 頁的「檢視已命名的配置的目標」
- 第 140 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 135 頁的「已命名的配置」
- 第 136 頁的「default-config 配置」
- 第 138 頁的「編輯已命名的配置的特性」
- 第 139 頁的「編輯參照配置之實例的連接埠號」
- 第 140 頁的「檢視已命名的配置的目標」
- 第 140 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。

特性名稱	說明
IOP_SSL_MUTUALAUTH_PORT	IIOIP 偵聽程式 SSL_MUTUALAUTH 在其上進行偵聽的用於 IIOIP 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 135 頁的「已命名的配置」
 - 第 138 頁的「建立已命名的配置」
 - 第 140 頁的「檢視已命名的配置的目標」
 - 第 140 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 137 頁的「唯一連接埠號和配置」
 - 第 135 頁的「已命名的配置」
 - 第 138 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 137 頁的「唯一連接埠號和配置」
 - 第 135 頁的「已命名的配置」
 - 第 138 頁的「建立已命名的配置」
 - 第 138 頁的「編輯已命名的配置的特性」
 - 第 140 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 135 頁的「已命名的配置」
 - 第 138 頁的「建立已命名的配置」
 - 第 138 頁的「編輯已命名的配置的特性」
 - 第 140 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 141 頁的「關於節點代理程式」
- 第 151 頁的「使用節點代理程式」
- 第 156 頁的「透過 `asadmin` 使用節點代理程式」

關於節點代理程式

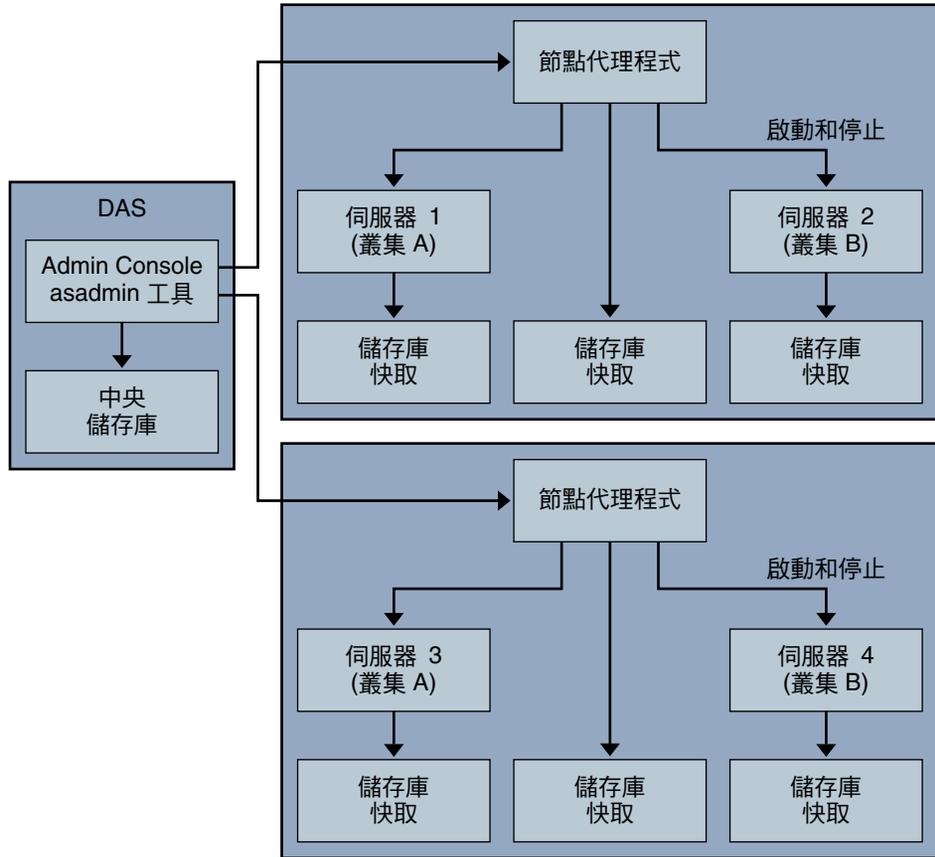
- 第 141 頁的「何為節點代理程式？」
- 第 143 頁的「節點代理程式預留位置」
- 第 143 頁的「部署節點代理程式」
- 第 146 頁的「節點代理程式和網域管理伺服器同步」
- 第 149 頁的「檢視節點代理程式記錄」
- 第 150 頁的「可以通過管理主控台和 `asadmin` 工具執行的作業」

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。

如果停止節點代理程式，則該節點代理程式所管理的伺服器實例也將被停止。

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。

另請參閱

- 第 143 頁的「部署節點代理程式」
- 第 143 頁的「節點代理程式預留位置」
- 第 146 頁的「節點代理程式和網域管理伺服器同步」
- 第 152 頁的「建立節點代理程式預留位置」

- 第 156 頁的「建立節點代理程式」
- 第 157 頁的「啟動節點代理程式」
- 第 158 頁的「停止節點代理程式」
- 第 158 頁的「刪除節點代理程式」

節點代理程式預留位置

您可以在缺少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。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，在網域管理伺服器 (DAS) 上建立的節點代理程式配置。

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 156 頁的「建立節點代理程式」和第 157 頁的「啟動節點代理程式」。

另請參閱：

- 第 152 頁的「建立節點代理程式預留位置」
- 第 141 頁的「何為節點代理程式？」
- 第 143 頁的「部署節點代理程式」
- 第 146 頁的「節點代理程式和網域管理伺服器同步」

部署節點代理程式

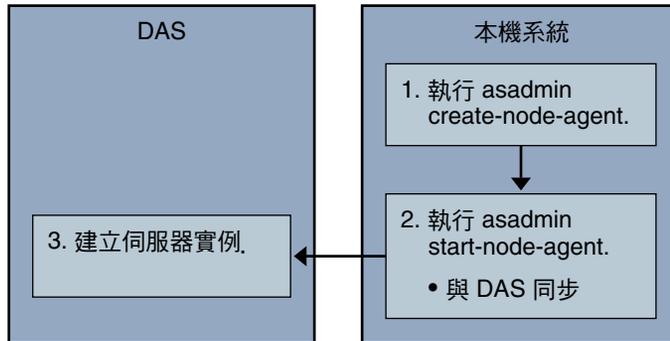
您可以使用以下兩種方法配置和部署節點代理程式：

- **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 156 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。

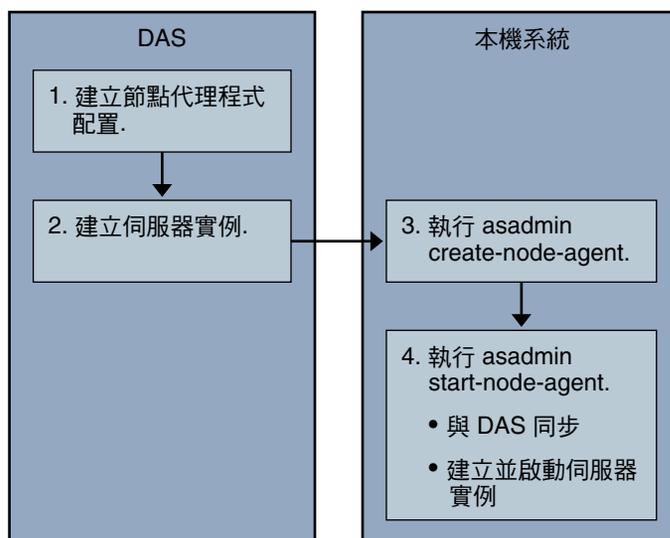
請參閱第 157 頁的「[啟動節點代理程式](#)」，以取得更多資訊。

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 152 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 156 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 157 頁的「啟動節點代理程式」，以取得更多資訊。

- 另請參閱**
- 第 141 頁的「何為節點代理程式？」
 - 第 143 頁的「節點代理程式預留位置」
 - 第 146 頁的「節點代理程式和網域管理伺服器同步」
 - 第 150 頁的「可以通過管理主控台和 `asadmin` 工具執行的作業」
 - 第 156 頁的「建立節點代理程式」

- 第 152 頁的「建立節點代理程式預留位置」
- 第 157 頁的「啟動節點代理程式」

節點代理程式和網域管理伺服器同步

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，並且快取在節點代理程式的本地機器上，因此這兩者必須同步。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 146 頁的「節點代理程式同步」
- 第 146 頁的「伺服器實例同步」
- 第 147 頁的「同步程式庫檔案」
- 第 148 頁的「獨特的設定和配置管理」
- 第 148 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。

如果節點代理程式啓動伺服器實例的方法並非向管理主控台或 `asadmin` 工具傳透明確的請求，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啓動節點代理程式，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 7-1 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> 。
<code>lib/ext</code>	可放置 Java 延伸類別 (例如 <code>zip</code> 或 <code>jar</code> 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 <code>jar</code> 檔案。

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。

爲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 Enterprise Edition 8.1 2005Q2 Developer's Guide」中的「Using the Common Classloader」。)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。

爲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/config/cluster1`、`config/lib/ext` - 自動增加至 `java.ext.dirs` (<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。如果要爲獨立伺服器實例 (`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：`<java-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爲：

```
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，節點代理程式爲 `node0`，JVM 選項爲 `-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，請重新啓動節點代理程式，因爲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

如果應用程式需要在由應用程式伺服器同步的目錄 (applications、generated、docroot、config、lib) 中儲存和讀取檔案，請使用 doNotRemoveList 標幟。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，在伺服器啓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 的預設位置爲 *install_dir/nodeagents*

可以通過管理主控台和 **asadmin** 工具執行的作業

對於節點代理程式，有些作業必須從執行該節點代理程式的系統上本機執行，而其他作業則可以在網域管理伺服器中執行。需要直接從本機執行的作業，只能在節點代理程式常駐的機器上透過 **asadmin** 工具來執行。而在網域管理伺服器中執行的作業，可透過管理主控台和 **asadmin** 工具執行。

下表概括了這些作業以及執行這些作業的位置：

表 7-2 通過管理主控台或是 **asadmin** 指令都可以執行的作業

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/配置	[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

如需更多資訊，請參閱：

- 第 141 頁的「何為節點代理程式？」
- 第 143 頁的「部署節點代理程式」
- 第 152 頁的「建立節點代理程式預留位置」
- 第 153 頁的「刪除節點代理程式配置」
- 第 153 頁的「編輯節點代理程式配置」
- 第 156 頁的「建立節點代理程式」
- 第 157 頁的「啟動節點代理程式」
- 第 158 頁的「停止節點代理程式」
- 第 158 頁的「刪除節點代理程式」

使用節點代理程式

- 第 151 頁的「檢視一般節點代理程式資訊」
- 第 152 頁的「建立節點代理程式預留位置」
- 第 153 頁的「刪除節點代理程式配置」
- 第 153 頁的「編輯節點代理程式配置」
- 第 154 頁的「編輯節點代理程式範圍」
- 第 154 頁的「為 JMX 編輯節點代理程式的偵聽程式」

▼ 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 157 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機機器中建立的預留位置。

請參閱第 156 頁的「建立節點代理程式」和第 157 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。

- 另請參閱
- 第 156 頁的「建立節點代理程式」
 - 第 157 頁的「啓動節點代理程式」
 - 第 150 頁的「可以通過管理主控台和 `asadmin` 工具執行的作業」
 - 第 141 頁的「何為節點代理程式？」
 - 第 143 頁的「節點代理程式預留位置」
 - 第 146 頁的「節點代理程式和網域管理伺服器同步」
 - 第 153 頁的「編輯節點代理程式配置」
 - 第 153 頁的「刪除節點代理程式配置」

▼ 建立節點代理程式預留位置

由於您必須從託管節點代理程式的機器上，直接從本機建立節點代理程式，因此您只能透過管理主控台為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，請在託管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 156 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 143 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
```

- 另請參閱
- 第 150 頁的「可以通過管理主控台和 `asadmin` 工具執行的作業」
 - 第 141 頁的「何為節點代理程式？」
 - 第 143 頁的「節點代理程式預留位置」
 - 第 156 頁的「建立節點代理程式」
 - 第 157 頁的「啓動節點代理程式」
 - 第 153 頁的「編輯節點代理程式配置」

- [第 153 頁的「刪除節點代理程式配置」](#)

▼ 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 158 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 158 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
```

- 另請參閱
- [第 150 頁的「可以通過管理主控台和 `asadmin` 工具執行的作業」](#)
 - [第 141 頁的「何為節點代理程式？」](#)
 - [第 143 頁的「節點代理程式預留位置」](#)
 - [第 158 頁的「停止節點代理程式」](#)
 - [第 152 頁的「建立節點代理程式預留位置」](#)
 - [第 153 頁的「編輯節點代理程式配置」](#)
 - [第 158 頁的「刪除節點代理程式」](#)

▼ 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 啓動代理程式後，在 [啓動] 上核取 [啓動實例] 以啓動該代理程式的伺服器實例。您還可以從此頁面手動啓動和停止實例。

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 156 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。

- 另請參閱
- 第 150 頁的「可以通過管理主控台和 `asadmin` 工具執行的作業」
 - 第 141 頁的「何為節點代理程式？」
 - 第 143 頁的「節點代理程式預留位置」
 - 第 146 頁的「節點代理程式和網域管理伺服器同步」
 - 第 152 頁的「建立節點代理程式預留位置」
 - 第 156 頁的「建立節點代理程式」
 - 第 157 頁的「啟動節點代理程式」
 - 第 153 頁的「刪除節點代理程式配置」

▼ 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。

- 另請參閱
- 第 141 頁的「何為節點代理程式？」
 - 第 143 頁的「節點代理程式預留位置」
 - 第 153 頁的「編輯節點代理程式配置」

▼ 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 [接受所有位吞 旁邊的核取方塊以允許連線到所有 IP 位均C
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 [已啓用] 方塊。
依預設已啓用 [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 [已啓用] 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。如需更多資訊，請參閱「Sun Java System Application Server Enterprise Edition 8.1 2005Q2 管理指南」中的「使用證書和 SSL」。
- 12 在 [SSL3/TLS] 區段：
 - a. 核取要在偵聽程式上啓用的安全性協定。
必須核取 SSL3 和/或 TLS。

b. 選取協定所使用的密碼組。

若要啓用所有密碼組，請核取 [所有支援的密碼組]。

13 按一下 [儲存]。

- 另請參閱
- 第 141 頁的「何為節點代理程式？」
 - 第 143 頁的「節點代理程式預留位置」
 - 第 153 頁的「編輯節點代理程式配置」

透過 asadmin 使用節點代理程式

您可以使用 `asadmin` 執行以下節點代理程式作業：

- 第 156 頁的「建立節點代理程式」
- 第 157 頁的「啓動節點代理程式」
- 第 158 頁的「停止節點代理程式」
- 第 158 頁的「刪除節點代理程式」

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。

範例 7-1 建立節點代理程式的範例

以下指令會建立節點代理程式：

```
asadmin create-node-agent --host myhost --port 4849 ---user admin nodeagent1
```

其中，`myhost` 是網域管理伺服器 (DAS) 主機名稱、4849 是 DAS 連接埠號、`admin` 是 DAS 使用者、`nodeagent1` 是要建立的節點代理程式的名稱。

備註 - 在以下情況下，必須指定一個可連線 DNS 的主機名稱：

- 網域跨越了子網路邊界 (即，節點代理程式和網域管理伺服器 [DAS] 處於不同的網域中，例如 sun.com 和 java.com)
- 如果使用主機名稱未在 DNS 中註冊的 DHCP 機器，

建立網域和節點代理程式時，可透過明確指定網域和節點代理程式的主機名稱來指定一個可連線 DNS 的主機名稱：

```
create-domain --domainproperties domain.hostName=DAS-host-name
create-node-agent --hostDAS-host-name
--agentproperties remoteclientaddress=node-agent-host-name
```

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。

如需更多資訊，請參閱：

- 第 141 頁的「何為節點代理程式？」
- 第 143 頁的「節點代理程式預留位置」
- 第 150 頁的「可以通過管理主控台和 `asadmin` 工具執行的作業」
- 第 143 頁的「部署節點代理程式」
- 第 152 頁的「建立節點代理程式預留位置」

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 nodeagent1
```

其中，`admin` 為管理使用者，`nodeagent1` 為要啓動的節點代理程式。

如需更多資訊，請參閱：

- 第 141 頁的「何為節點代理程式？」
- 第 143 頁的「節點代理程式預留位置」
- 第 150 頁的「可以通過管理主控台和 `asadmin` 工具執行的作業」
- 第 143 頁的「部署節點代理程式」
- 第 153 頁的「編輯節點代理程式配置」

停止節點代理程式

若要停止正在執行的節點代理程式，請在該節點代理程式常駐的系統中執行 `asadmin` 指令 `stop-node-agent`。 `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。

如需更多資訊，請參閱：

- 第 141 頁的「何為節點代理程式？」
- 第 143 頁的「部署節點代理程式」
- 第 150 頁的「可以通過管理主控台和 `asadmin` 工具執行的作業」
- 第 157 頁的「啟動節點代理程式」

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` 指令 `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。

刪除節點代理程式時，您還必須使用管理主控台或 `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。

如需更多資訊，請參閱：

- 第 141 頁的「何為節點代理程式？」
- 第 143 頁的「部署節點代理程式」
- 第 150 頁的「可以通過管理主控台和 `asadmin` 工具執行的作業」
- 第 158 頁的「停止節點代理程式」

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 159 頁的「階段作業持續性和容錯移轉簡介」
- 第 161 頁的「設定高可用性階段作業持續性」
- 第 163 頁的「HTTP 階段作業容錯移轉」
- 第 167 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：

- 每個伺服器實例均可以存取同一高可用性資料庫 (HADB)。如需有關如何啓用此資料庫的資訊，請參閱 `configure-ha-cluster(1)`。
- 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 應用程式配置為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` 選項設定為 `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。
- 共同存在的實體 Bean、有狀態階段作業 Bean 和分散式實體 Bean 遠端主機參照、遠端參照
- 分散式階段作業 Bean 遠端主機和遠端參照
- `InitialContext` 和 `java:comp/env` 的 JNDI 上下文。
- `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
- 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。

應用程式範例

以下目錄包含說明階段作業持續性的範例應用程式：

```
install_dir/samples/ee-samples/highavailability  
install_dir/samples/ee-samples/failover
```

以下範例應用程式說明 SFSB 階段作業持續性：

```
install_dir/samples/ee-samples/failover/apps/sfsbfailover
```

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 161 頁的「設定高可用性階段作業持續性」
- 第 162 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。如需有關如何停用這些功能的資訊，請參閱「Sun Java System Application Server Enterprise Edition 8.1 2005Q2 管理指南」中的第 2 章「部署應用程式」。

1 建立 Application Server 叢集。

如需更多資訊，請參閱第 126 頁的「建立叢集」。

2 為叢集建立 HADB 資料庫。

如需更多資訊，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 97 頁的「設定 HTTP 負載平衡」

4 啟用所需應用程式伺服器實例和 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 162 頁的「啟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 重新啟動叢集中的每個伺服器實例。

如果實例目前正在處理請求，請在重新啟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 112 頁的「停用 (靜止) 伺服器實例或叢集」

6 啟用所需的特定 SFSB 的可用性。

選取需要對階段作業狀態設定檢查點的方法。請參閱第 169 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 在部署期間啟用個別應用程式、Web 模組或 EJB 模組的可用性。

請參閱第 169 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。如需說明，請參閱下一小節第 162 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 163 頁的「配置 Web 容器的可用性」
 - 第 167 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

若要在給定範圍啓用可用性，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。

給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` 特性設定為 `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)`。
- 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

J2EE 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

若要使用 `asadmin` 啓用和配置 Web 容器的可用性，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` 特性設定為 `true`，然後使用 `configure-ha-persistence` 根據需要設定特性。

例如，如下使用 `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 164 頁的「可用性設定」小節中所述。
- 5 重新啓動伺服器實例。

可用性設定

[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。允許的值包括 `memory` (無持續性)、`file` (檔案系統) 和 `ha` (HADB)。

必須配置並啓用 HADB 後，才能使用 `ha` 階段作業持續性。如需詳細配置資訊，請參閱 `configure-ha-cluster(1)`。

如果啓用了 Web 容器可用性，預設為 `ha`。否則，預設為 `memory`。對於需要階段作業持續性的生產環境，請使用 `ha`。前兩個類型 (`memory` 和 `file`) 的持續性不提供高可用性階段作業持續性。

持續性頻率：指定儲存階段作業狀態的頻率。僅當持續性類型為 `ha` 時適用。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。僅當持續性類型為 `ha` 時適用。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狀態已修改。您必須保證每次變更屬性時都呼叫 `setAttribute()`。這不是 J2EE 規格的需求，但是此模式需要這樣做才能正常工作。
- `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：核取此方塊以啓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 165 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

若要啓用和配置個別 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` 才能啓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 Enterprise Edition 8.1 2005Q2 Developer's 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

在單一應用程式伺服器實例中，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 Enterprise Edition 8.1 2005Q2 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，首先請啓用伺服器實例和 Web 容器的可用性，然後啓用單次登入狀態容錯移轉。

透過管理主控台中的 [可用性服務] 的 [Web 容器可用性] 標籤，您可以啓用單次登入狀態容錯移轉，如第 163 頁的「配置 Web 容器的可用性」中所述。您也可以使用 `asadmin set` 指令將配置的

`availability-service.web-container-availability.sso-failover-enabled` 特性設定為 `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您使用 `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 的狀態可以儲存在持續性存放區中。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。這稱為**檢查點操作**。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。

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 170 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

▼ 啟用 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 168 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 重新啟動伺服器實例。

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  
4849config1.availability-service.ejb-container-availability  
.availability-enabled="true"
```

```
asadmin set --user admin --passwordfile password.txt --host localhost --port  
4849config1.availability-service.ejb-container-availability  
.sfsb-persistence-type="file"
```

```
asadmin set --user admin --passwordfile password.txt --host localhost --port  
4849config1.availability-service.ejb-container-availability  
.sfsb-ha-persistence-type="ha"
```

可用性設定

[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。允許的值包括 `file` (檔案系統) 和 `ha` (HADB)。對於需要階段作業持續性的生產環境，請使用預設值 `ha`。

SFSB 持續性類型：為尚未啓用可用性的 SFSB 指定鈍化機制。允許的值包括 `file` (預設值) 和 `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。如需更多資訊，請參閱「Sun Java System Application Server Enterprise Edition 8.1 2005Q2 管理指南」中的「配置儲存特性」。

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。如需更多資訊，請參閱「Sun Java System Application Server Enterprise Edition 8.1 2005Q2 管理指南」中的「配置儲存特性」。

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

若要啓用個別 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。

若要啓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 Enterprise Edition 8.1 2005Q2 Developer's Guide」中的「The `sun-ejb-jar.xml` File」。

範例 8-1 啓用了可用性的 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

如果啓用，通常會在 Bean 完成了任意作業事件之後（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，指定額外選擇性的 SFSB 檢查點設定功能，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，即不啓動也不確定 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，並且沒有在 `checkpoint-at-end-of-method` 元素中明確指定 SFSB 的方法，則不會對 Bean 的狀態設定檢查點（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 8-2 指定方法檢查點之 EJB 部署描述元範例

```
<sun-ejb-jar>
...
  <enterprise-beans>
```

範例 8-2 指定方法檢查點之 EJB 部署描述元範例 (續)

```
...
<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 173 頁的「Java 訊息服務簡介」
- 第 174 頁的「配置 Java 訊息服務」
- 第 176 頁的「連線池儲存和容錯移轉」
- 第 178 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 J2EE 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。它可啓用可靠的非同步鬆耦合分散式通訊。實作 JMS 的 Sun Java System Message Queue 3 2005Q1 (MQ) 與 Application Server 緊密整合在一起，可讓您建立諸如訊息導引 Bean (MDB) 之類的元件。

MQ 透過**連接器模組** (也稱為資源介面，由 J2EE 連接器架構規格 1.5 定義) 與 Application Server 整合在一起。部署至 Application Server 的 J2EE 元件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 資源介面。

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。

範例應用程式

`mqfailover` 範例應用程式說明了在訊息導引 Bean 接收來自 JMS 主題的內送訊息時的 MQ 容錯移轉。此範例包含 MDB 和應用程式用戶端。應用程式伺服器使 MDB 具有高可用性。如果某個代理程式中斷，則交談式狀態 (由 MDB 接收的訊息) 不需設定即可遷移至叢集中其他可用的代理程式實例。

此範例將安裝至以下目錄：

```
install_dir/samples/ee-samples/failover/apps/mqfailover
```

詳細資訊

如需有關 JMS 的更多資訊，請參閱「Sun Java System Application Server Enterprise Edition 8.1 2005Q2 Developer's Guide」中的第 14 章「Using the Java Message Service」。如需有關連接器 (資源介面) 的更多資訊，請參閱「Sun Java System Application Server Enterprise Edition 8.1 2005Q2 Developer's Guide」中的第 9 章「Developing Connectors」。

如需有關 Sun Java System Message Queue 的更多資訊，請參閱 Sun Java System Message Queue 文件。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。如需詳細資訊，請參閱「Sun Java System Application Server Enterprise Edition 8.1 2005Q2 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` 指令可列示所有的 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 Enterprise Edition 8.1 2005Q2 管理指南」中的「有關 JMS 連線工廠的管理主控台作業」。

備註 – 變更 Java 訊息服務的配置後，必須重新啓動 Application Server 實例。

如需有關 JMS 管理的更多資訊，請參閱「Sun Java System Application Server Enterprise Edition 8.1 2005Q2 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 可透過以下兩種方式與 Application Server 進行整合：LOCAL 與 REMOTE，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 (獨立 Application Server 實例的預設) 時，Application Server 將啓動和停止被指定為預設 JMS 主機的 MQ 代理程式。LOCAL 類型最適合於獨立的 Application Server 實例。

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 代理程式的啓動參數。

REMOTE Java 訊息服務

當 Type 屬性為 REMOTE 時，必須單獨啓動 MQ 代理程式。如果在 Application Server 中定義了叢集，則此為預設。如需有關啓動代理程式的資訊，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 分別啓動和停止 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 代理程式啓動參數。可忽略 Start Arguments 屬性。

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，並且每當變更 JMS 主機配置時，均會進行更新。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「Sun Java System Application Server Enterprise Edition 8.1 2005Q2 管理指南」中的「建立 JMS 主機」。
- 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 可自動池儲存 JMS 連線。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。

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 Enterprise Edition 8.1 2005Q2 管理指南」中的「有關 JMS 連線工廠的管理主控台作業」。

負載平衡的訊息流入

Application Server 可將訊息隨機傳送至具有相同 ClientID 的 MDB。長期用戶需要 ClientID。

對於其中未配置 ClientID 的非長期用戶，訂閱相同主題的特定 MDB 之所有實例均視為相等。如果將 MDB 部署至 Application Server 的多個實例，則僅一個 MDB 會接收訊息。如果多個不同的 MDB 訂閱相同的主題，則每個 MDB 會有一個實例接收訊息的副本。

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，則 MQ 可允許多達該數目的 MDB 使用同一佇列中的訊息。訊息會隨機傳送至 MDB。如果將 `maxNumActiveConsumers` 設定為 -1，則對用戶數目沒有限制。

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 Enterprise Edition 8.1 2005Q2 Deployment Planning Guide」中的「Planning Message Queue Broker Deployment」。

▼ 啓用 MQ 叢集與應用程式伺服器叢集

- 1 如果其中之一尚不存在，則建立應用程式伺服器叢集。

如需有關建立叢集的資訊，請參閱第 126 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` 命令行公用程式建立 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 啟動該叢集中的應用程式伺服器實例。

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：

- 檢視應用程式伺服器記錄檔。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。
- 請務必先啓動 MQ 代理程式，然後再啓動應用程式伺服器實例。

- 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啓動應用程式伺服器需花費 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 183 頁的「簡介」
- 第 185 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。但是，向叢集增加新實例或從叢集中刪除實例將不會更新該叢集的現有用戶端檢視。若要如此，必須在用戶端手動更新端點清單。

需求

如果以下所有條件均適用，則 Sun Java System Application Server Enterprise Edition 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應用程式伺服器實例的叢集。

- J2EE 應用程式已部署到所有參與負載平衡的應用程式伺服器實例和叢集。
- 已啟用 RMI-IIOP 用戶端應用程式以用於負載平衡。

對於存取 Application Server 上部署的 EJB 元件的以下 RMI-IIOP 用戶端，Application Server 支援其負載平衡。

- 在應用程式用戶端容器 (ACC) 中執行的 Java 應用程式。請參閱第 185 頁的「為應用程式用戶端容器設定 RMI-IIOP 負載平衡」。
- 不在 ACC 中執行的 Java 應用程式。請參閱第 186 頁的「為獨立用戶端設定 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。

隨後每次用戶端建立新的 `InitialContext` 物件時，將自動重建終點清單，使用不同的 IIOP 終點進行 `InitialContext` 作業。

在您從透過 `InitialContext` 物件獲得的參照中獲取或建立 Bean 時，將在提供 IIOP 終點 (指定給 `InitialContext` 物件) 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。

主端點是與用於查詢或建立 Bean 的 `InitialContext` 端點相對應的 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。

您可以將 RMI-IIOP 負載平衡和容錯移轉配置為與在 ACC 中執行的應用程式以及獨立的 Java 用戶端配合使用。

範例應用程式

以下目錄包含用於演示透過和不透過 ACC 使用 RMI-IIOP 防故障備用的範例應用程式。

`install_dir/samples/ee-samples/sfsbfailover`

請參閱此範例隨附的 `index.html` 檔案，以取得有關在使用和不使用 ACC 的情況下執行應用程式的說明。 `ee-samples` 目錄還包含有關設定環境以執行範例的資訊。

設定 RMI-IIOP 負載平衡和容錯移轉

您可以為在應用程式用戶端容器 (ACC) 中執行的應用程式和獨立用戶端應用程式設定 RMI-IIOP 負載平衡和容錯移轉。

▼ 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 的額外資訊，請參閱「Sun Java System Application Server Enterprise Edition 8.1 2005Q2 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/`。
如需欲更新的路徑變數之清單，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 尋找叢集中實例的 IIOP 偵聽程式連接埠號。
將 IIOP 偵聽程式指定為用於確定哪個 IIOP 偵聽程式接收請求的端點。若要在管理主控台中顯示 IIOP 偵聽程式，請執行以下作業：
 - 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。
記下特定實例的 IIOP 偵聽程式連接埠。
- e. 對每個實例重複該程序。

7 編輯端點值的 sun-acc.xml。

使用之前步驟中的 IIOP 偵聽程式，依循以下格式建立端點值：

```
machine1:instance1-iiop-port,machine2:instance2-iiop-port
```

例如：

```
<property name="com.sun.appserv.iiop.endpoints"  
value="host1.sun.com:3335,host2.sun.com:3333,host3.sun.com:3334">
```

8 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
```

9 執行應用程式用戶端，如下所示：

```
appliance -client clientjar -name appname
```

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。

▼ 為獨立用戶端設定 RMI-IIOP 負載平衡和容錯移轉

1 使用 --retrieve 選項部署應用程式，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
```

2 執行用戶端 jar 以及所需的 jar 檔案，然後指定端點並將 InitialContext 指定為 -D 值。

例如：

```
java -Dcom.sun.appserv.iiop.endpoints=  
host1.sun.com:33700,host2.sun.com:33700,host3.sun.com:33700  
samples.rmiiopclient.client.Standalone_Client
```

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，然後確認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, 111
AddressList, 和預設 JMS 主機, 176
Apache, 負載平衡外掛程式所做的修改, 102
asadmin create-jms-host 指令, 176
asadmin get 指令, 174
asadmin set 指令, 174

C

cacheDatabaseMetaData 特性, 71
checkpoint-at-end-of-method 元素, 170
ConnectionTrace 屬性, 66
CoreFile 屬性, 66
create-http-lb-config 指令, 108
create-http-lb-ref 指令, 109
create-node-agent 指令, 156

D

DatabaseName 屬性, 66
databuf 選項, 89
DataBufferPoolSize 屬性, 66
datadevices 選項, 61
DataDeviceSize 屬性, 66, 80
dbpassword 選項, 56
dbpasswordfile 選項, 56
default-config 配置, 136
delete-http-lb-ref 指令, 109
delete-node-agent 指令, 158

devicepath 選項, 61
DevicePath 屬性, 66, 83
devicesize 選項, 61
disable-http-lb-application 指令, 113
disable-http-lb-server 指令, 112

E

EagerSessionThreshold 屬性, 66
EagerSessionTimeout 屬性, 66
EJB 容器, 可用性, 167-168
eliminateRedundantEndTransaction 特性, 71
enable-http-lb-application 指令, 109
enable-http-lb-server 指令, 109
EventBufferSize 屬性, 67
export-http-lb-config 指令, 111

F

fast 選項, 78

H

HADB
nodes, 86
用戶支援, 31
列出資料庫, 77
延伸節點, 80-81
取得 JDBC URL, 69-70
取得狀態, 85-87

HADB (續)

- 取得資源資訊, 89-91
- 取得裝置資訊, 87
- 架構, 28-29
- 重新分段, 83
- 重新啟動資料庫, 77
- 重新啟動節點, 75
- 連接埠指定, 64
- 連線池特性, 70-71
- 連線池設定, 70
- 配置, 59-72
- 設定屬性, 62, 64
- 停止資料庫, 76
- 停止節點, 74
- 啟動資料庫, 76
- 啟動節點, 73
- 移除資料庫, 78
- 清除資料庫, 78
- 異質多工式裝置路徑, 63
- 資料庫名稱, 60
- 資料損壞, 79-80
- 節點, 29
- 監視, 85-91
- 增加節點, 81
- 增加機器, 81
- 歷史檔案, 93
- 機器維護, 91
- 環境變數, 57
- 雙網路, 34-35

HADB 配置

- 時間同步化, 37
- 節點管理員程序, 39-40
- 網路配置, 32-35
- 檔案系統支援, 38

HADB 設定, 32

HADB 管理代理程式, 啟動, 41-43, 47-54

- hadbm addnodes 指令, 81
- hadbm clear 指令, 78
- hadbm clearhistory 指令, 93
- hadbm create 指令, 60
- hadbm delete 指令, 78
- hadbm deviceinfo 指令, 87
- hadbm get 指令, 64
- hadbm list 指令, 77

- hadbm refragment 指令, 83
- hadbm resourceinfo 指令, 89-91
- hadbm restart 指令, 77
- hadbm restartnode 指令, 75
- hadbm start 指令, 76
- hadbm startnode 指令, 73
- hadbm status 指令, 85-87
- hadbm stop 指令, 76
- hadbm stopnode 指令, 74
- hadbm 指令, 54-58
- historypath 選項, 61
- HistoryPath 屬性, 67
- hosts 選項, 61, 83

HTTP

- HTTPS 路由, 114
- 階段作業容錯移轉, 114-115

HTTP_LISTENER_PORT 特性, 138

HTTP_SSL_LISTENER_PORT 特性, 138

HTTP 階段作業, 26

- 分散式, 159

HTTPS

- routing, 108
- 階段作業容錯移轉, 114-115
- 路由, 114

I

- IIOB_LISTENER_PORT 特性, 138
- IIOB_SSL_MUTUALAUTH_PORT 特性, 139
- InternalLogbufferSize 屬性, 67
- IOP_SSL_LISTENER_PORT 特性, 138

J

- JdbcUrl 屬性, 67

JMS

- 建立主機, 176
- 連線池儲存, 176
- 連線容錯移轉, 176
- 配置, 174

JMS 主機清單, 連線, 176

- JMX_SYSTEM_CONNECTOR_PORT 特性, 138
- JMX 偵聽程式, 節點代理程式, 154

JNDI 名稱設定, 71

L

loadbalancer.xml 檔案, 111

locks 選項, 89

logbuf 選項, 89

LogbufferSize 屬性, 67

M

magnus.conf 檔案, Web 伺服器, 99

maxStatement 特性, 71

MaxTables 屬性, 67

Microsoft 網際網路資訊服務 (IIS), 對負載平衡的修改, 105

N

nilogbuf 選項, 89

no-refragment 選項, 82

no-repair 選項, 74

nodes 選項, 86

number-healthcheck-retries, 111

NumberOfDatadevices 屬性, 67

NumberOfLocks 屬性, 67

NumberOfSessions 屬性, 67

O

obj.conf 檔案, Web 伺服器, 99

P

password 特性, 70

portbase 選項, 62

Portbase 屬性, 67

R

RelalgdeviceSize 屬性, 67

routecookie, 108

S

saveto 選項, 93

serverList 特性, 70

SessionTimeout 屬性, 67

set 選項, 62, 63

spares 選項, 62, 78, 83

SQLTraceMode 屬性, 68

start-node-agent 指令, 157

startlevel 選項, 74, 75

StartRepairDelay 屬性, 68

StatInterval 屬性, 68

stop-node-agent 指令, 158

sun-ejb-jar.xml 檔案, 170

Sun Java System Message Queue, 連接器用於, 174

sun-passthrough.properties 檔案, 和記錄層級, 123

Sun Web Server, 負載平衡器所做的修改, 99

SyslogFacility 屬性, 68

SysLogging 屬性, 68

SysLogLevel 屬性, 69

SyslogPrefix 屬性, 69

T

TakeoverTime 屬性, 69

U

username 特性, 70

W

Web 伺服器

用於負載平衡的修改, 99-107

多個實例和負載平衡, 107

Web 容器, 可用性, 163

Web 應用程式, 可分散, 161

- 已命名的配置
 - default-config, 136
 - 共用, 136
 - 連接埠號, 137
 - 預設名稱, 136
 - 關於, 135
- 分散式 HTTP 階段作業, 159
- 反向代理外掛程式, 98
- 中央儲存庫, 節點代理程式同步, 146
- 可分散的 Web 應用程式, 161
- 可用性
 - EJB 容器層級, 169-170
 - 啟用和停用, 162
 - 適用於 Web 模組, 159
 - 適用於有狀態階段作業 Bean, 167
 - 層級, 162
- 目標, 負載平衡器配置, 108
- 主端點, RMI-IIOP 容錯移轉, 184
- 未指定的請求, 96
- 必需的連線驗證設定, 70
- 全域作業事件支援設定, 70
- 名稱設定, 70
- 有狀態階段作業 Bean, 167
 - 階段作業持續性, 167, 169
- 池名稱設定, 71
- 作業事件
 - 階段作業持續性, 167, 170
- 作業事件隔離設定, 70
- 伺服器, 叢集, 125
- 伺服器實例
 - 為負載平衡啟用, 109
 - 靜止, 112
- 居留式循環負載平衡, 96
- 所有連線故障設定, 70
- 保證隔離層級設定, 70
- 持續的池大小設定, 70
- 持續性, 階段作業, 26
- 持續性存放區, 適用於有狀態階段作業 Bean 狀態, 167
- 指定的請求, 96
- 負載平衡
 - HTTP, 關於, 95
 - HTTP 演算法, 96
 - RMI-IIOP 需求, 183
- 負載平衡 (續)
 - 多個 Web 伺服器實例, 107
 - 居留式循環, 96
 - 建立負載平衡器配置, 108
 - 建立參照, 109
 - 指定的請求, 96
 - 記錄訊息, 121
 - 動態重新配置, 112
 - 設定, 98
 - 階段作業容錯移轉, 114-115
 - 做為反向代理外掛程式使用, 98
 - 啟用伺服器實例, 109
 - 啟用應用程式, 109
 - 等冪 URL, 115
 - 運作狀態檢查程式, 109
 - 匯出配置檔案, 111
 - 靜止伺服器實例或叢集, 112
 - 靜止應用程式, 113
 - 變更配置, 112
- 表名稱設定, 70
- 記錄
 - 負載平衡器, 121
 - 檢視節點代理程式記錄, 149
- 通道外掛程式, 98
- 時間同步化, 37
- 連接埠號, 和配置, 137
- 連線池
 - HADB 特性, 70-71
 - HADB 設定, 70
- 配置。參閱已命名的配置
- 容錯移轉
 - JMS 連線, 176
 - RMI-IIOP 需求, 183
 - 有狀態階段作業 Bean 狀態, 167
 - 適用於 Web 模組階段作業, 159
 - 關於 HTTP, 95
- 動態重新配置, 負載平衡器的, 112
- 階段作業
 - HTTP, 26
 - 持續性, 26
- 階段作業持續性
 - 單次登入, 165-166
 - 適用於 Web 模組, 159
 - 適用於有狀態階段作業 Bean, 167, 169

- 階段作業容錯移轉, HTTP 和 HTTPS, 114-115
- 階段作業儲存
 - 適用於 HTTP 階段作業, 164
 - 適用於有狀態階段作業 Bean, 168, 169
- 基於 Cookie 的階段作業居留性, 96
- 等冪 URL, 115
- 單次登入, 階段作業持續性, 165-166
- 循環負載平衡, 居留式, 96
- 異常伺服器實例, 109
- 運作狀態檢查程式, 109
- 替代端點, RMI-IIOP 容錯移轉, 184
- 資料來源已啟用設定, 72
- 資料來源類別名稱設定, 70
- 資料庫供應商設定, 70
- 節點代理程式
 - JMX 偵聽程式, 154
 - 安裝, 145
 - 附加, 142-143
 - 刪除, 153, 158
 - 建立, 156
 - 記錄, 149
 - 停止, 158
 - 啟動, 157
 - 預留位置, 143, 152
 - 與網域管理伺服器同步, 146
 - 認證範圍, 154
 - 部署, 143
 - 關於, 141
- 節點管理員程序和高可用性, 39-40
- 演算法
 - HTTP 負載平衡, 96
 - RMI-IIOP 容錯移轉, 184
- 管理主控台
 - 使用以建立 JMS 主機, 176
 - 使用以配置 JMS 服務, 174
- 端點, RMI-IIOP 容錯移轉, 184
- 網域管理伺服器
 - 伺服器實例同步, 146
 - 節點代理程式同步, 146
- 網路配置需求, 32-35
- 對有狀態階段作業 Bean 狀態設定檢查點, 161
- 認證範圍, 節點代理程式, 154
- 範圍, 節點代理程式認證, 154
- 部署, 設定可用性, 161
- 靜止
 - 伺服器實例或叢集, 112
 - 應用程式, 113
- 應用程式
 - 為負載平衡啟用, 109
 - 靜止, 113
- 檔案系統支援, 38
- 檢查點操作, 167
 - 選取方法, 167, 170
- 叢集, 125
 - 共用, 25
 - 獨立, 25
 - 靜止, 112
- 叢集伺服器實例, 配置, 136
- 驗證方法設定, 70

