



Sun Java System Web Server 7.0 管理員指南



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

文件號碼：820-0873
2006 年 10 月

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

本產品或文件受版權保護，在限制其使用、複製、發行及反編譯的授權下發行。未經 Sun 及其授權人(如果有) 事先的書面許可，不得使用任何方法、任何形式來複製本產品或文件的任何部分。協力廠商的軟體，包括字型技術，其版權歸 Sun 供應商所有，經授權後使用。

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

Sun、Sun Microsystems、Sun 標誌、docs.sun.com、AnswerBook、AnswerBook2、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。

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

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

目錄

前言	19
1 入門	25
簡介	25
新增功能	25
啓動 Administration Server	26
在 Unix/Linux 中啓動 Administration Server	26
▼ 在 Unix/Linux 中啓動 Administration Server	26
在 Windows 中啓動 Administration Server	26
不同的伺服器管理方法	26
使用管理主控台	27
管理主控台 GUI 畫面說明	29
使用 CLI	29
CLI 的模式	29
wadm CLI 的位置	30
以 CLI 進行認證	30
瞭解 Web Server 7.0	31
2 配置、實例和節點	35
簡介	35
管理配置	35
建立配置	36
複製伺服器配置	37
部署伺服器配置	38
刪除伺服器配置	38
管理伺服器實例	38
建立伺服器實例	38

啟動伺服器實例	39
停止伺服器實例	39
重新啟動伺服器實例	40
重新配置伺服器實例	40
刪除伺服器實例	41
自動配置實例	41
▼ 增加排定的事件	42
▼ 移除排定的事件	43
3 伺服器機群和叢集	45
Sun Java System Web Server 中的叢集支援	45
設定伺服器機群	45
▼ 設定伺服器機群	46
設定簡單叢集	47
▼ 配置叢集	49
4 部署案例	51
部署架構	51
部署簡介	53
部署前置作業需求	55
部署 Web Server	55
叢集環境	56
硬體和軟體要求	56
設定叢集	58
階段作業複製	60
階段作業複製與容錯移轉操作	60
啟用階段作業複製	61
為階段作業複製配置 Web 應用程式	62
監視叢集	63
Solaris Zones	63
5 使用虛擬伺服器	65
虛擬伺服器簡介	65
使用案例	65

預設配置	66
安全伺服器	66
企業網路主機作業	66
大量主機作業	67
管理虛擬伺服器	67
增加虛擬伺服器	67
▼ 增加虛擬伺服器	67
配置虛擬伺服器	68
▼ 配置虛擬伺服器	68
複製虛擬伺服器	69
▼ 複製虛擬伺服器	69
配置 HTTP 偵聽程式	69
建立 HTTP 偵聽程式	70
配置 HTTP 偵聽程式	70
6 憑證與金鑰	73
使用憑證進行認證	73
伺服器認證	74
用戶端認證	74
憑證金鑰類型	74
建立自我簽署的憑證	76
管理憑證	76
請求憑證	76
▼ 請求憑證	77
安裝憑證	78
▼ 安裝憑證	78
更新憑證	79
▼ 更新憑證	79
刪除憑證	79
▼ 刪除憑證	80
更新管理伺服器憑證	80
管理憑證撤銷清單 (CRL)	80
▼ 安裝 CRL	81
▼ 刪除 CRL	81
設定內部記號的密碼	82

▼ 設定記號密碼	82
為伺服器配置 SSL	83
啟用配置的 SSL 密碼	83
啟用 HTTP 偵聽程式的安全性	84
7 控制對伺服器的存取	87
何為存取控制	87
存取控制的運作方式	88
為使用者/群組設定存取控制	89
預設認證	89
基本認證	90
SSL 認證	90
摘要認證	91
為主機 IP 設定存取控制	92
配置 ACL 使用者快取記憶體	93
設定 ACL 快取特性	93
配置存取控制	94
增加存取控制清單 (ACL)	94
增加存取控制項目 (ACE)	96
使用 .htaccess 檔案	98
防止伺服器受到拒絕服務攻擊	99
限制對伺服器的請求	99
▼ 限制最大連線數	100
8 管理使用者和群組	101
存取關於使用者和群組的資訊	101
關於目錄服務	101
目錄服務類型	102
瞭解辨別名稱 (DN)	102
使用 LDIF	103
使用認證資料庫	103
建立認證資料庫	104
設定使用者和群組	105
▼ 增加使用者	105
▼ 增加群組	105

▼ 刪除使用者	106
▼ 刪除群組	107
靜態和動態群組	107
靜態群組	108
動態群組	108
9 管理伺服器內容	111
配置文件目錄	111
▼ 建立文件目錄	112
變更預設 MIME 類型	112
▼ 變更預設 MIME 類型	113
自訂使用者公用資訊目錄 (UNIX/Linux)	113
▼ 配置文件目錄	114
限定內容發佈	114
啟動時載入整個密碼檔案	115
設定 URL 重新導向	115
使用常規表示式進行 URL 重新導向	116
CGI 簡介	117
為您的伺服器配置 CGI 子系統	119
下載可執行檔	121
為 Windows 安裝 Shell CGI 程式	121
適用於 Windows 的 Shell CGI 程式簡介	121
自訂錯誤回應	122
變更字元集	123
▼ 變更字元集	124
設定文件註腳	124
▼ 設定文件頁尾	124
限制符號連結 (UNIX/Linux)	125
▼ 限制符號連結	125
設定伺服器剖析的 HTML	126
▼ 設定伺服器剖析的 HTML	126
設定快取控制指令	127
▼ 設定快取控制指令	127
為內容壓縮配置伺服器	128
配置伺服器以提供預先壓縮內容	128

▼ 變更預先壓縮的內容設定	128
將伺服器配置為依需求壓縮內容	129
▼ 依需求壓縮內容	129
配置反向代理伺服器	130
▼ 增加代理伺服器 URI	130
▼ 修改反向代理參數	131
設定 P3P	132
▼ 配置虛擬伺服器的 P3P 設定	132
10 透過 WebDAV 進行 Web 發佈	135
關於 WebDAV	136
常用 WebDAV 術語	136
在實例層級啟用 WebDAV	139
管理 WebDAV 集合	140
啟用 WebDAV 集合	140
停用 WebDAV 集合	140
增加 WebDAV 集合	140
列出 WebDAV 集合	140
移除 WebDAV 集合	140
配置 WebDAV 特性	141
設定 WebDAV 特性	141
檢視 WebDAV 特性	141
設定 WebDAV 集合特性	141
檢視 WebDAV 集合特性	141
修改 WebDAV 參數	142
在伺服器層級停用 WebDAV	143
管理 WebDAV 認證資料庫	143
在啟用了 WebDAV 的伺服器上使用來源 URI 和 Translate:f 標頭	144
鎖定和解除鎖定資源	144
專用鎖定	145
共用鎖定	145
最小鎖定逾時	145
11 使用 Java 和 Web 應用程式	147
配置 Java 以與 Sun Java System Web Server 搭配使用	147

▼ 為配置啓用 Java	147
設定 Java 類別路徑	148
▼ 設定 Java 類別路徑	148
配置 JVM	149
▼ 配置 JVM	149
增加 JVM 選項	149
增加 JVM 效能評測器	149
▼ 增加 JVM 效能評測器	150
為伺服器啓用 Java 除錯	150
▼ 啓用 JVM 除錯	150
部署 Java Web 應用程式	151
增加 Web 應用程式	151
▼ 部署 Web 應用程式	151
部署 Web 應用程式目錄	152
部署時預編譯 JSP	152
配置 Servlet 容器	153
▼ 設定 Servlet 容器	153
Servlet 容器全域參數	153
配置伺服器生命週期模組	154
伺服器生命週期簡介	154
▼ 增加生命週期模組	154
▼ 刪除生命週期模組	155
配置 Java 資源	156
配置 JDBC 資源	157
Sun Java System Web Server 支援的 JDBC 驅動程式	157
管理 JDBC 資源	160
▼ 增加新的 JDBC 資源	160
管理 JDBC 連線池	160
▼ 建立 JDBC 連線池	160
註冊自訂資源	162
▼ 增加自訂資源	162
使用外部 JNDI 資源	163
▼ 增加外部 JNDI 資源	163
配置郵件資源	164
▼ 增加郵件資源	164
配置 SOAP 認證提供者	165

▼ 增加 SOAP 認證提供者	165
SOAP 認證提供者參數	166
配置階段作業複製	167
設定階段作業複製	168
▼ 設定階段作業複製	168
管理認證範圍	169
▼ 增加認證範圍	171
12 使用搜尋集合	173
關於搜尋	173
配置搜尋特性	174
配置搜尋集合	175
支援的格式	175
增加搜尋集合	175
刪除搜尋集合	177
排程集合更新	177
執行搜尋	179
搜尋頁面	179
進行查詢	179
▼ 進行查詢	180
進階搜尋	180
▼ 進行進階搜尋查詢	180
文件欄位	181
搜尋查詢運算子	181
檢視搜尋結果	181
自訂搜尋頁面	181
搜尋介面元件	182
自訂搜尋查詢頁面	182
自訂搜尋結果頁面	183
在單獨的頁面中自訂表單和結果	185
標籤慣例	185
標籤規格	186
13 監視伺服器	187
Sun Java System Web Server 中的監視功能	187

透過管理主控台監視	188
▼ 檢視統計資料	188
修改監視參數	189
配置監視參數	190
配置 SNMP 子代理程式參數	190
配置 SNMP 子代理程式	191
使用 CLI 配置 SNMP	193
▼ 在 Solaris 上啟動 SNMP	193
▼ 在 Linux 上啟動 SNMP	194
▼ 在 Windows 上啟動 SNMP	194
▼ 配置基於對等端的主代理程式 (magt)	195
設定伺服器的記錄	195
記錄類型	196
檢視存取記錄及伺服器記錄	196
配置記錄參數	196
配置 Administration Server 的記錄設定	199
▼ 修改伺服器記錄位置	199
▼ 修改記錄詳細度級別	200
▼ 修改記錄的日期格式	200
14 國際化和本土化	201
輸入多位元組資料	201
檔案名稱或目錄名稱	201
LDAP 使用者與群組	201
支援多字元編碼	202
WebDAV	202
搜尋	202
配置伺服器以提供本土化內容	202
▼ 搜尋順序	202
A CLI 自先前版本後所經歷的變更	205
B FastCGI 外掛程式	209
簡介	209

外掛程式函數 (SAF)	210
auth-fastcgi	210
responder-fastcgi	210
filter-fastcgi	211
error-fastcgi	211
FastCGI SAF 參數	211
error-fastcgi SAF 錯誤原因字串	213
配置 Web Server 的 FastCGI 外掛程式	213
修改 magnus.conf	214
修改 MIME 類型 (可選擇)	214
修改 obj.conf	215
FastCGI 外掛程式疑難排解	216
開發 FastCGI 應用程式	217
▼ 執行 FastCGI 應用程式	217
FastCGI 應用程式範例	219
使用 PHP 撰寫的回應程式應用程式 (ListDir.php)	219
使用 Perl 撰寫的 Authorizer 應用程式 (SimpleAuth.pl)	219
使用 C 撰寫的 Filter 應用程式 (SimpleFilter.c)	220
 C Web 服務	225
在 Web Server 7.0 上執行 JWSDP 2.0 範例	225
▼ 執行 JWSDP 2.0 範例	225
 字彙表	229
 索引	237

圖清單

圖 4-1	說明在單一節點上部署 Web Server 的流程圖	54
圖 4-2	叢集設定	57
圖 4-3	說明叢集設定的流程圖	58

表清單

表 6-1	HTTP 偵聽程式安全性特性	85
表 7-1	摘要認證的不同情況	91
表 7-2	ACL 參數	94
表 7-3	ACE 參數	96
表 7-4	配置請求限制	99
表 8-1	動態群組：必要參數	110
表 9-1	URL 重新導向參數	116
表 9-2	CGI 參數	120
表 10-1	WebDAV 參數	142
表 10-2	WebDAV 認證資料庫特性	143
表 10-3	Sun Java System Web Server 處理鎖定請求的方式	146
表 11-1	Servlet 容器參數	153
表 11-2	一般與受支援的 JDBC 驅動程式清單	158
表 11-3	自訂資源特性	162
表 11-4	外部 JNDI 資源特性	164
表 11-5	郵件資源特性	165
表 11-6	SOAP 認證提供者參數	166
表 11-7	階段作業複製參數	169
表 11-8	範圍類型	170
表 12-1	欄位說明 > 新增搜尋事件排程	178
表 13-1	監視種類	188
表 13-2	欄位說明 > 一般監視設定	190
表 13-3	欄位說明 > SNMP 子代理程式設定	191
表 13-4	一般準則	192
表 13-5	欄位說明 > 編輯存取記錄喜好設定	197
表 13-6	欄位說明 > 編輯伺服器記錄喜好設定	197
表 13-7	欄位說明 > 設定記錄自動重建	198
表 A-1	CLI 自先前版本後所經歷的變更	205

範例清單

前言

本指南將說明如何配置及管理 Sun Java™ System Web Server 7.0。

本書適用對象

本書功能是為了讓 Sun Java System Web Server 管理員能管理生產環境中的伺服器。本指南假設您熟悉以下方面：

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

在閱讀本書之前

Sun Java System Web Server 7.0 可做為獨立產品或 Sun Java Enterprise System (Java ES) 的元件進行安裝，Java ES 是一種支援分散在網路或網際網路環境中的企業應用程式的軟體基礎架構。如果您要將 Sun Java System Web Server 7.0 做為 Java ES 的元件進行安裝，則應熟悉位於 <http://docs.sun.com/coll/1286.2> 和 <http://docs.sun.com/coll/1412.2> 上的系統文件。

Sun Java System Web Server 7.0 文件集

Sun Java System Web Server 7.0 文件集說明如何安裝及管理 Web Server。Sun Java System Web Server 7.0 文件的 URL 為 <http://docs.sun.com/coll/1308.3> 和 <http://docs.sun.com/coll/1425.2>。如需有關 Sun Java System Web Server 7.0 的簡介，請依循下表排列順序參閱其中列出的書籍。

表 P-1 Sun Java System Web Server 7.0 文件集中的書籍

文件標題	內容
「Sun Java System Web Server 7.0 Documentation Center」	依作業和主旨安排的 Web Server 文件主題
「Sun Java System Web Server 7.0 版本說明」	<ul style="list-style-type: none">■ 關於軟體與文件的最新資訊■ 安裝 Web Server 的支援平台及修補程式需求
「Sun Java System Web Server 7.0 Installation and Migration Guide」	執行安裝和遷移工作： <ul style="list-style-type: none">■ 安裝 Web Server 及其各種元件。■ 將資料從 Sun ONE Web Server 6.0 或 6.1 遷移至 Sun Java System Web Server 7.0
「Sun Java System Web Server 7.0 管理員指南」	執行下列管理作業： <ul style="list-style-type: none">■ 使用管理介面與指令行介面■ 配置伺服器喜好設定■ 使用伺服器實例■ 監視並記錄伺服器狀態■ 使用憑證與公用金鑰加密來保護伺服器的安全■ 配置存取控制來保護伺服器的安全■ 使用 Java 平台企業版 (Java EE) 安全性功能■ 部署應用程式■ 管理虛擬伺服器■ 定義伺服器工作負荷量與調整系統，以滿足效能需要■ 搜尋伺服器文件的內容與屬性，並建立文字搜尋介面■ 為內容壓縮配置伺服器■ 使用 WebDAV，為 Web 發佈與內容創作配置伺服器
「Sun Java System Web Server 7.0 Developer's Guide」	使用程式設計技術和 API 來執行下列作業： <ul style="list-style-type: none">■ 延伸與修改 Sun Java System Web Server■ 動態產生內容以回應用戶端請求並修改伺服器的內容
「Sun Java System Web Server 7.0 Update 1 NSAPI Developer's Guide」	建立自訂的 Netscape Server Application Programmer's Interface (NSAPI) 外掛程式
「Sun Java System Web Server 7.0 Developer's Guide to Java Web Applications」	在 Sun Java System Web Server 中實作 Java Servlet 和 JavaServer Pages™ (JSP™) 技術
「Sun Java System Web Server 7.0 Administrator's Configuration File Reference」	編輯配置檔案
「Sun Java System Web Server 7.0 Performance Tuning, Sizing, and Scaling Guide」	調整 Sun Java System Web Server 以最佳化效能
「Sun Java System Web Server 7.0 Troubleshooting Guide」	Web Server 疑難排解

相關書籍

Sun Java Enterprise System (Java ES) 及其元件的所有相關文件，其 URL 爲
<http://docs.sun.com/app/docs/prod/entsys.06q4> 和
http://docs.sun.com/app/docs/prod/entsys.06q4?l=zh_TW。

預設路徑及檔案名稱

下表說明本書中使用的預設路徑和檔案名稱。

表 P-2 預設路徑及檔案名稱

預留位置	說明	預設值
<i>install_dir</i>	表示 Sun Java System Web Server 7.0 的基底安裝目錄。	Solaris™ 平台上的 Sun Java Enterprise System (Java ES) 安裝： /opt/SUNWwbsvr7 Linux 和 HP-UX 平台上的 Java ES 安裝： /opt/sun/webserver/ Windows 平台上的 Java ES 安裝： System Drive :\Program Files\Sun\JavaES5\WebServer7 其他 Solaris、Linux 和 HP-UX 安裝 (非超級使用者)： user's home directory/sun/webserver7 其他 Solaris、Linux 和 HP-UX 安裝 (超級使用者)： /sun/webserver7 Windows 的所有安裝： System Drive :\Program Files\Sun\WebServer7
<i>instance_root</i>	包含實例專屬子目錄之目錄。	Solaris 上實例的預設位置爲： /var/opt/SUNWwbsvr7. Linux 和 HP-UX 上實例的預設位置爲： /var/opt/sun/webserver7 Windows 上實例的預設位置爲： System Drive :\Program Files\sun\WebServer7 對於 Java ES 安裝，Windows 上實例的預設位置爲： System Drive:\Program Files\Sun\JavaES5\WebServer7

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 .login 檔案。 請使用 ls -a 列出所有檔案。 machine_name% you have mail.
AaBbCc123	您所鍵入的內容 (與螢幕畫面輸出相區別)。	machine_name% su Password:
AaBbCc123	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 rm 檔案名稱。
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。
「AaBbCc123」 用於書名及章節名稱。		

符號慣例

下表說明在本書中可能用到的符號。

表 P-4 符號慣例

符號	說明	範例	涵義
[]	含有選擇性引數和指令選項。	ls [-l]	-l 非必要選項。
{ }	包含所需指令選項的一組選擇。	-d {y n}	-d 選項需要使用 y 引數或 n 引數。
\${ }	表明變數參照。	\${com.sun.javaRoot}	參照 com.sun.javaRoot 變數的值。
-	結合多個同步按鍵。	Ctrl-A	當您按住 A 鍵時按下 Ctrl 鍵。
+	結合多個連續按鍵。	Ctrl+A+N	按下 Ctrl 鍵，釋放該鍵，然後按下隨後的鍵。
→	指出圖形使用者介面中的功能表項目選項。	[檔案] → [新增] → [範本]	從 [檔案] 功能表中，選擇 [新增]。從 [新增] 子功能表中，選擇 [範本]。

存取 Sun 線上資源

<http://docs.sun.com> (docs.sun.comSM) 網站可讓您線上存取 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)

搜尋 Sun 產品文件

除了可以從 docs.sun.com 網站搜尋 Sun 產品文件之外，您還可以使用搜尋引擎，方法是在搜尋欄位中鍵入以下語法：

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

例如，若要搜尋「Web Server」，請鍵入以下內容：

```
Web Server site:docs.sun.com
```

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

協力廠商網站參照

本文件中提供了協力廠商 URL 以供參考，另亦提供其他相關的資訊。

備註 – Sun 對本文件中提到的協力廠商網站的可用性不承擔任何責任。對於此類網站或資源中的 (或透過它們取得的) 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 (或透過它們取得的) 任何內容、商品或服務而造成的、宣稱的或連帶產生的實際或名義損壞或損失，Sun 概不負責。

Sun 歡迎您提出寶貴意見

Sun 致力於提高文件品質，因此誠心歡迎您提出意見與建議。若要分享您的意見，請至 <http://docs.sun.com>，並按一下 [Send Comments (傳送您的意見)]。在線上表單中，請提供完整的文件標題和文件號碼。文件號碼是一個七位或九位的數字，可在書的標題頁面或文件的 URL 中找到。例如，本書的文件號碼為 820-0873。

◆◆◆ 第 1 章

入門

本章透過簡要說明本指南中使用的術語來介紹 Sun Java System Web Server 7.0 的基礎。

- 第 25 頁的「簡介」
- 第 25 頁的「新增功能」
- 第 26 頁的「啟動 Administration Server」
- 第 26 頁的「不同的伺服器管理方法」
- 第 27 頁的「使用管理主控台」
- 第 29 頁的「使用 CLI」
- 第 31 頁的「瞭解 Web Server 7.0」

簡介

Sun Java System Web Server 7.0 是建立於工業標準上的多程序、多重執行緒的安全 Web 伺服器。它可為中型到大型企業提供高效能、穩定性、可延伸性及管理能力。

Web Server 7.0 可提供完備的指令行介面支援、合併的配置、增強的安全性 (支援橢圓曲線加密) 以及叢集支援。此外，它還提供強大的內建遷移工具，用於協助將應用程式與配置從 Web Server 6.0 和 6.1 遷移至 Sun Java System Web Server 7.0。

新增功能

請參閱「Sun Java System Web Server 7.0 版本說明」中的第 1 章「Sun Java System Web Server 版本說明」，以取得有關 Sun Java System Web Server 7.0 新增功能和增強功能的更多資訊。

啓動 Administration Server

若要使用管理介面，您需要啓動 Administration Server。

在 Unix/Linux 中啓動 Administration Server

如需啓動 Administration Server，請執行以下作業：

▼ 在 Unix/Linux 中啓動 Administration Server

- 1 移至 `install_root/admin-server/bin` 目錄 (例如 `/usr/sjsws7.0/admin-server/bin`)
- 2 鍵入 `./startserv`。
該指令將使用您在安裝時指定的連接埠號啓動 Administration Server。

在 Windows 中啓動 Administration Server

Sun Java System Web Server 安裝程式可為 Windows 平台建立包含多個圖示的程式群組。程式群組包含以下圖示：

- 版本說明
- 啓動 Administration Server
- 解除安裝 Web Server

請注意，Administration Server 會以服務 applet 的形式執行，因此，您也可以使用控制面板直接啓動該服務。

不同的伺服器管理方法

您可以使用下列使用者介面管理 Sun Java System Web Server：

- Administration Console (GUI)。
- 指令行介面 (`wadm shell`)。

您可以使用 `wadm shell` 介面 (稍後將在本章中進行討論) 或網路型管理主控台來管理實例。請注意，管理節點只能有一個特定配置實例處於執行中狀態。

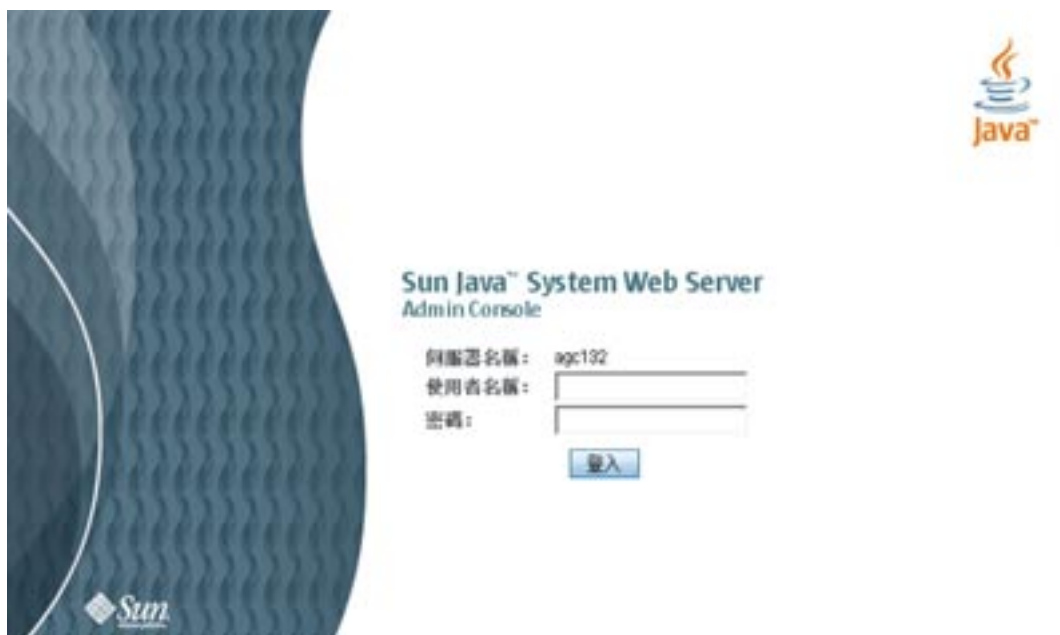
使用管理主控台

安裝 Sun Java System Web Server 後，即可使用瀏覽器存取管理主控台。

用於瀏覽至 Administration Server 頁面的 URL，取決於您在安裝 Sun Java System Web Server 時為 Administration Server 選擇的電腦主機名稱和連接埠號。例如，如果您在 SSL 連接埠 1234 上安裝了 Administration Server，則 URL 如下所示：

`https://myserver.sun.com:1234/`

您需要登入到管理主控台才能執行伺服器管理。您可於在電腦上安裝 Sun Java System Web Server 時設定管理員的使用者名稱和密碼。下圖顯示的是認證畫面：



存取 Administration Server 時，首先看到的是 [常用工作] 頁面。您可以使用此頁面上的按鈕來管理、增加、移除以及遷移 Sun Java System Web Server。下圖顯示常用工作頁面：



備註 - 按一下其中任何標籤都會使其子標籤顯示在該頁面上。子標籤提供的動作特定於父標籤的功能。

下圖顯示所選標籤的子標籤：



按一下標籤將在同一視窗中開啓頁面。某些作業涉及逐步收集使用者的資料。Administration Console 具有此類作業的精靈介面。這些精靈永遠在新視窗中開啓。

管理主控台 GUI 畫面說明

所有表單元素和 GUI 元件均具有詳細的內嵌說明，其中提供有關驗證和可選參數的資訊。對於精靈介面，在執行精靈中的任何步驟時，按一下 [說明] 標籤即可取得特定於目前作業的說明。

使用 CLI

本節說明 Sun Java System Web Server 7.0 的指令行介面，並定義所有支援的伺服器配置和管理指令。

Sun Java System Web Server 7.0 引入了一種稱為 `wadm` 的新 CLI。

舊版的伺服器支援少數不連續的指令行，這些指令行組合在一起只能處理 GUI 所提供的全部管理功能中的一小部分。Sun Java System Web Server 6.1 支援的指令行介面包括 `HttpServerAdmin`、`wdeploy` 和 `flexanlg`。新 CLI (`wadm`) 具有下列功能：

- 內嵌式 JACL shell，用於產生程序檔。
- 可延伸的 CLI—藉由協力廠商外掛程式可將更多指令增加到 CLI。

備註 – Sun Java System Web Server 7.0 不支援 `HttpServerAdmin`。

備註 – Sun Java System Web Server 7.0 支援 `wdeploy` 只是為了與 6.x 版本向下相容，且只能在 Administration Server 節點上運作。

CLI 的模式

`wadm` 支援三種不同模式的呼叫。它們是：

- **獨立模式**—在此模式下，將從指令 `shell` 呼叫 `wadm`，指定所需的指令、選項和運算元。指令執行完成後，CLI 將結束並返回至 `shell`。此模式可支援以互動式及非互動式方法執行指令。互動式執行為預設設定，如果密碼檔案 (透過 `--password-file` 選項傳送) 中尚未指定密碼，則將提示輸入密碼。如果未指定 `--password-file` 選項，非互動式執行將產生錯誤。例如，

```
wadm> create-config --user=admin --password-file=./admin.pwd
--http-port=2222 --server-name=syrinx myconfig
```

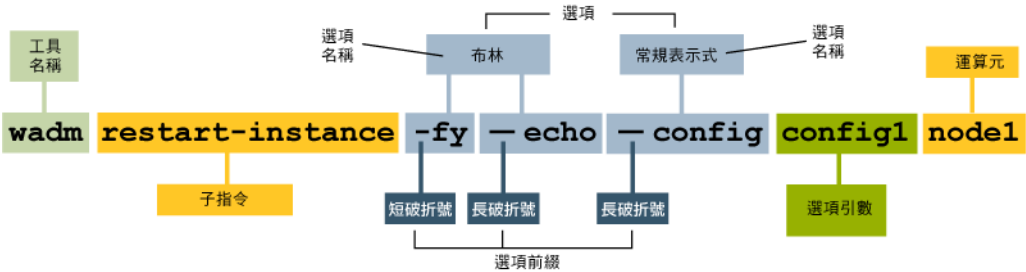
- **Shell 模式**—在此模式下，將從無任何指令的指令 `shell` 呼叫 `wadm`。`wadm` 會提示使用者輸入指令。執行指令後，它將返回至 `shell`。鍵入結束或退出指令可結束此 `shell`。互動式及非互動式執行均可採用此模式。例如，

```
wadm -user=admin -host=serverhost --password-file=admin.pwd --port=8989
```

- **檔案模式**—在此模式下，您可在檔案中增加指令清單，並將該檔案做為引數傳送至 wadm。例如，

```
wadm -user=admin -host=serverhost --password-file=admin.pwd
--port=8989 -commands-file=/space/scripts/admscr
```

下圖描述呼叫 wadm 指令的語法。



備註 – 藉由 wadm CLI 可執行使用管理主控台能夠完成的所有作業。

wadm CLI 的位置

問題: 在何處可以找到用於 Sun Java System Web Server 7.0 管理的 CLI？

答案: 管理 CLI 位於 install-root/bin/wadm。若要使用 CLI，您需要瞭解下列資訊：

- 管理伺服器的主機名稱 (預設為 localhost)。
- 管理伺服器的 SSL 連接埠 (預設為 8989)。
- 管理伺服器的使用者名稱 (預設為 admin)。
- 管理伺服器的密碼。

備註 – 管理伺服器需要處於執行狀態才能使用 CLI。執行 install-root/admin-server/bin/startserv 即可啟動伺服器。

以 CLI 進行認證

wadm 將使用管理員的使用者名稱和密碼，來通過 Administration Server 的認證。您必須將有效的使用者名稱和密碼檔案當成引數，傳送至在單一模式下執行的每個指令。Shell 模式在呼叫 wadm 可執行檔時，可接受使用者名稱和密碼檔案。在 shell 模式下呼叫的指令不需要連線選項 (例如 user、password-file、host、port 和 ssl)。如果已指定選項，則會將其忽略。

CLI 支援的一些指令需要輸入密碼。例如 `bindpw`、`user-password` 和 `token-pin`。使用者可在包含管理使用者密碼的同一檔案中指定這些密碼。如果指令中未指定 `password-file`，則將會提示使用者輸入密碼。

如果 Administration Server 已啓用 SSL，則 `wadm` 會透過 SSL 與該 Administration Server 進行通訊。驗證由 Administration Server 傳送的憑證時，會以 `truststore (~/.wadmtruststore)` 為依據。如果憑證存在並且有效，指令將繼續正常執行。否則，`wadm` 將顯示憑證，並讓使用者選擇是否接受此憑證。如果使用者接受憑證，則會將此憑證增加到 `truststore`，並且指令將繼續正常執行。

備註 – `truststore` 不需要密碼保護，因為其中並不包含機密資料。

瞭解 Web Server 7.0

Web Server 包括新的管理架構，可在伺服器機群中為多個伺服器提供增強的分散式管理功能。藉由強力的管理功能，可使用圖形化介面與指令行介面來遠端管理和部署 Web Server。您可以在伺服器機群內集中管理多台伺服器，並可將伺服器分散到一個或多個節點以建立伺服器實例。此外還能對這些伺服器實例進行監視與生命週期管理。

Web Server 經過配置可讓您開啓或關閉各種功能，確定如何回應個別用戶端請求，以及撰寫可在伺服器上執行並與伺服器作業互動的程式。用於識別這些選項的指示 (稱為指令) 儲存在配置檔案中。Sun Java System Web Server 在啓動時和用戶端請求期間讀取配置檔案，以便將您的選擇與所需的伺服器活動對映。

如需有關這些檔案的更多資訊，請參閱 Sun Java System Web Server 7.0 的「*Administrator's Configuration File Reference Guide*」。

在 Web Server 7.0 中，伺服器實例的所有可配置元素 (如 Web 應用程式、配置檔案和搜尋集合索引) 均以邏輯方式組成群組，這稱為**配置**。使用 CLI 或網路型管理介面可建立、修改或刪除配置。您可以一次管理多個配置。「配置」這個專有名詞也指一組中介資料，可配置伺服器的執行階段服務。例如，執行階段服務從已配置的文件根目錄為網頁提供服務。伺服器執行階段使用配置中介資料載入內建服務和協力廠商外掛程式，以及設定其他伺服器延伸 (例如，為網頁和動態 Web 應用程式提供服務的資料庫驅動程式)。

備註 – 所有與配置相關的檔案都儲存在檔案系統的儲存庫中，該儲存庫稱為**配置存放區**。除非本指南中明確指定，否則絕對不能手動編輯此儲存庫中的任何檔案。

在 Web Web Server 中，使用 CLI 或透過基於 Web 的管理介面對配置所做的任何變更都將首先變更配置存放區，然後會部署該配置。因此，變更會複製到實例目錄。Web 應用程式會部署到下列位置：

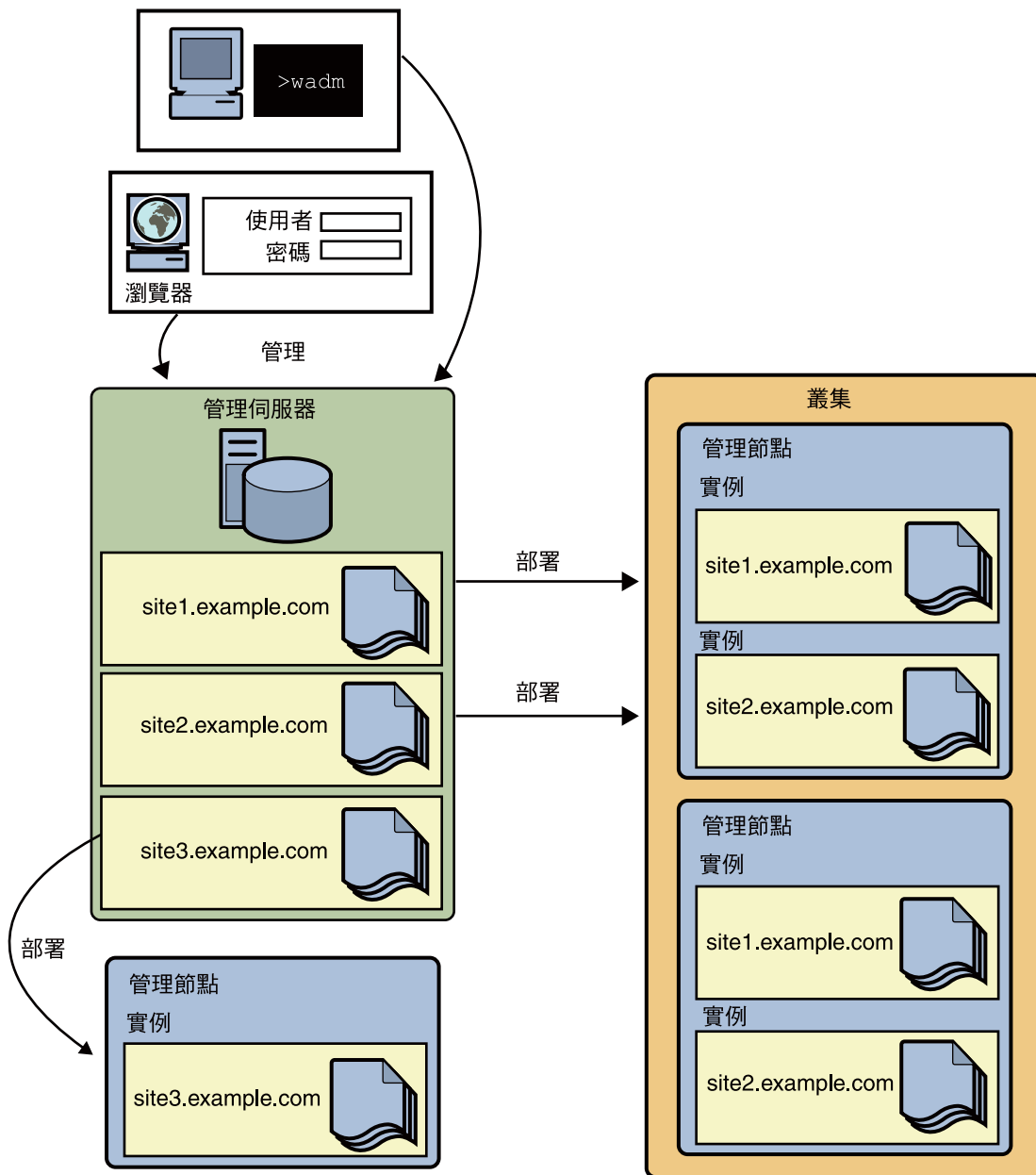
```
<install_dir>/admin-server/config-store/<config_name>/web-app/<virtual_servername>/
```

部署配置時，config-store 下的整個 Web 應用程式目錄和配置目錄都會壓縮並複製到伺服器實例目錄。此檔案為 current.zip 檔案，位於下列位置：

```
<install_dir>/admin-server/config-store/<config_name>
```

因此，對所選配置進行部署可能需要一些時間才能完成，視 Web 應用程式的大小而定。

下圖顯示了如何將配置部署到管理節點的概要圖表：



當您將配置部署到節點 (網路資源，如伺服器或主機) 時，將會建立該配置的實例。實例包含記錄檔及其他執行階段檔案，如實例所需的鎖定資料庫、快取和暫存檔案。您可以透過 CLI 或基於 Web 的管理介面來管理這些實例。

實例也可跨越一個或多個節點形成**叢集**。在叢集情況下，形成叢集的所有節點必須具有相同配置。叢集中的所有節點必須同質。它們必須具有相同的作業系統，以相同的方式配置並提供相同的服務。

伺服器機群中的某一個節點擁有一個已部署管理應用程式的執行中伺服器。這台專門配置的伺服器稱為 **Administration Server**，所部署的管理應用程式是網路型**管理主控台**。您可以使用管理主控制台來控制伺服器實例的生命週期。

Administration Server 可控制該節點 (稱為**管理節點**) 中其他伺服器的動作。管理節點不提供 GUI 介面。伺服器機群中的一個節點已安裝 Administration Server。伺服器機群中的其他所有節點均已安裝管理節點。管理節點在安裝時會向 Administration Server 進行註冊。此動作將使 Administration Server 可以識別該管理節點。

Administration Server 和管理節點自動透過 SSL 進行通訊。Administration Server 與管理節點彼此互相認證，即 Administration Server 信任管理節點的伺服器憑證，管理節點信任 Administration Server 提供的用戶端憑證。在管理節點註冊期間，Administration Server 將為該管理節點產生伺服器憑證，接著將此憑證下載並安裝到管理節點上。伺服器憑證的核發者也會安裝到管理節點上。

配置、實例和節點

上一章向您介紹了 Web Server 7.0 中的一些新概念。管理員的主要工作是配置和管理 Server 的執行階段服務。本章說明管理配置的各種方法，以及如何部署配置以在節點上啟動實例。

- [第 35 頁的「簡介」](#)
- [第 35 頁的「管理配置」](#)
- [第 38 頁的「管理伺服器實例」](#)
- [第 41 頁的「自動配置實例」](#)

簡介

實例是指指定節點上 Web 伺服器常駐程式的環境，包括其配置、記錄檔以及其他執行階段工件 (例如鎖定資料庫、快取記憶體和暫存檔)。

節點是網路資源，例如伺服器或主機。在一般資料中心中，節點網路稱為**伺服器機群**。本節討論如何使用管理主控台 GUI 配置節點。

您可以將一個或多個實例部署到節點。此外，您也可以將同一個實例部署到多個節點，並可形成不同叢集的一部分。

爲了管理，可以啟動、停止、啟動或動態重新配置實例。

管理配置

- [第 36 頁的「建立配置」](#)
- [第 37 頁的「複製伺服器配置」](#)
- [第 38 頁的「部署伺服器配置」](#)
- [第 38 頁的「刪除伺服器配置」](#)

建立配置

為使用 Web 伺服器進行啟動，需要建立配置。

若要建立新配置，請執行以下作業：

1. 按一下 **[配置]** 標籤。
2. 按一下 **[新增]** 按鈕。

精靈將指導您完成可用於建立配置的設定。以下小節說明精靈頁面中可用的欄位：

步驟 1 – 設定配置資訊

此精靈頁面可讓您為新配置設定一般資訊

請在精靈頁面中設定以下參數：

- **配置名稱**—為配置增加新的唯一名稱。
- **伺服器名稱**—為新配置增加伺服器名稱。此名稱可與配置名稱相同。
- **文件根目錄**—輸入有效的文件根目錄，所有部署的 Web 應用程式均在其中維護它們的目錄。預設值為 `./docs`。您可以輸入伺服器上任何有效目錄的路徑。
- **64 位元**—啟用/停用對 Web 伺服器的 64 位元支援。預設為**停用**。
- **伺服器使用者**—如果伺服器是在基於 UNIX 的系統上執行，請為伺服器程序提供有效的使用者名稱。例如：`root`。

步驟 2—為配置建立偵聽程式

此精靈頁面可讓您為新配置設定 HTTP 偵聽程式特性。

請在精靈頁面中設定以下參數：

- **連接埠**—配置要與其連結並偵聽請求的連接埠號。
- **IP 位址**—主機電腦的 IP 位址。鍵入 * 以設定所有可用的 IP 位址。

步驟 3—配置 Java、CGI 和 SHTML

此精靈頁面可讓您配置與 Java/CGI 和 SHTML 相關的特性。

請在精靈頁面中設定以下參數：

- **Java**—已啟用。依預設，Java 已啟用。**警告：**如果需要使用此配置部署基於 Java 的 Web 應用程式，請勿停用 Java。設定 Java SE 目錄的主目錄。預設值為指向隨附的 Java SE 目錄的目錄。您可以選取預設的 Java SE 目錄或指定新路徑。
- **CGI**—無 (停用 CGI 支援)、啟用為檔案類型 (啟用 CGI 支援) 和目錄 (指定儲存 CGI 文件的 URI 和路徑)。
- **SHTML**—依預設會停用 SHTML。

步驟 4— 建立實例

此精靈頁面可讓您為新配置建立實例。

請在精靈頁面中設定以下參數：

- 配置—新配置的名稱。
- 選取節點—選取用於建立新配置實例的節點。請從可用清單中選取節點，然後按一下 [增加] 或 [全部增加] 按鈕來增加節點。

備註—使用 CLI

若要透過 CLI 建立配置，請執行以下指令：

```
wadm> create-config --user=admin --password-file=admin.pwd  
--host=serverhost --port=8989 --http-port=8800 --server-user=user  
--server-name=servername config1
```

config1 為新配置的名稱。

請參閱 CLI 參照 create-config(1)。

複製伺服器配置

您可以複製伺服器配置並建立新配置。剛複製的配置與現有配置相同。但是，新配置並不包含任何實例，即使其複製的來源配置中包含實例亦如此。

若要複製配置，請執行以下作業：

1. 按一下 [配置] 標籤。
2. 從清單中選取配置。
3. 按一下 [複製] 按鈕。
4. 在快顯式視窗中，輸入新配置的名稱，然後按一下 [確定]。

備註—使用 CLI

若要透過 CLI 執行動作，請執行以下指令：

```
wadm> copy-config --user=admin --password-file=admin.pwd --host=serverhost  
--port=8989 --config=config1 copyconfig1
```

copyconfig1 為新配置的名稱。

請參閱 CLI 參照 copy-config(1)。

部署伺服器配置

您需要首先建立伺服器才能在節點上進行部署。

若要部署現有配置，請執行以下作業：

1. 按一下 **[配置]** 標籤。
2. 透過選取配置核取方塊識別配置。
3. 從配置動作清單中選取 **[部署配置]**。

刪除伺服器配置

備註 – 如果配置的實例已部署到節點，則無法刪除此配置。即使實例已部署但未執行，也不允許您刪除伺服器配置。停止正在執行的實例並進行部署，以便刪除配置。

若要刪除配置，請執行以下作業：

1. 按一下 **[配置]** 標籤。
2. 透過選取配置核取方塊識別配置。
3. 從配置動作清單中選取 **[刪除配置]**。

管理伺服器實例

- [第 38 頁的「建立伺服器實例」](#)
- [第 39 頁的「啟動伺服器實例」](#)
- [第 39 頁的「停止伺服器實例」](#)
- [第 40 頁的「重新啟動伺服器實例」](#)
- [第 40 頁的「重新配置伺服器實例」](#)
- [第 41 頁的「刪除伺服器實例」](#)

建立伺服器實例

建立新伺服器實例之前，請執行以下檢查：

1. 檢查您是否已建立配置。建立新伺服器實例需要指定現有實例配置。
2. 檢查伺服器機群中的所有可用節點是否均已具有所需配置的實例。您無法建立重複的實例。

透過執行以下作業建立新伺服器實例：

1. 按一下 **[配置]** 標籤，並從動作清單中選取 **[新增]**。

2. 在 [新增實例精靈] 頁面中，選取需要為其建立實例的配置，然後按 [下一步] 按鈕。
3. 選取在其之上應存在所選配置 [步驟 2] 之實例的節點。按 [下一步] 按鈕。
4. 檢視您的選取摘要。按 [下一步] 按鈕以檢視作業的結果。

備註 – 使用 CLI

若要建立伺服器實例，請執行以下指令：

```
wadm> create-instance --user=admin --password-file=admin.pwd  
--host=serverhost --port=8989 --config=config1 serverhost
```

請參閱 CLI 參照 `create-instance(1)`。

啓動伺服器實例

1. 按一下 [節點] 標籤，以檢視在伺服器中配置的節點清單。
2. 透過選取節點名稱核取方塊來選取節點。
3. 按一下 [啓動實例] 按鈕以開啓頁面視窗，此視窗列出該節點控制的所有實例。
4. 選取實例，然後按一下 [啓動實例] 按鈕以啓動實例。
5. 檢查實例狀態是否為執行中，然後關閉視窗。

備註 – 使用 CLI

若要透過 CLI 啓動伺服器實例，請執行以下指令：

```
wadm> start-instance --user=admin --password-file=admin.pwd --host=serverhost  
--port=8989 --config=config1 nodehost1
```

請參閱 CLI 參照 `start-instance(1)`。

停止伺服器實例

1. 按一下 [節點] 標籤，以檢視在伺服器中配置的節點清單。
2. 透過選取節點名稱核取方塊來選取節點。
3. 按一下 [停止實例] 按鈕以開啓頁面視窗，此視窗列出該節點控制的所有實例。
4. 選取實例，然後按一下 [停止實例] 按鈕以停止實例。
5. 檢查實例狀態是否為未執行，然後關閉視窗。

備註 – 使用 CLI

若要透過 CLI 停止伺服器實例，請執行以下指令：

```
wadm> stop-instance --user=admin --password-file=admin.pwd --host=serverhost  
--port=8989 --config=config1 nodehost1
```

請參閱 CLI 參照 `stop-instance(1)`。

重新啟動伺服器實例

1. 按一下 **[節點]** 標籤，以檢視在伺服器中配置的節點清單。
2. 透過選取節點名稱核取方塊來選取節點。
3. 按一下 **[重新啟動實例]** 按鈕以開啓頁面視窗，此視窗列出該節點控制的所有實例。
4. 選取實例，然後按一下 **[重新啟動實例]** 按鈕以重新啟動實例。
5. 檢查實例狀態是否為執行中，然後關閉視窗。

備註 – 使用 CLI

```
wadm> restart-instance --user=admin --password-file=admin.pwd  
--host=serverhost --port=8989 --config=config1 nodehost1
```

請參閱 CLI 參照 `restart-instance(1)`。

重新配置伺服器實例

對配置進行變更時，在某些情況下不需要重新啟動實例。Administration Server 支援重新配置伺服器實例，以引入對配置存放區所做的變更。在此配置中，無須重新啟動伺服器便已將變更反映在實例上。僅配置中可動態重新配置的變更會受到影響。

備註 – 重新配置後，使用者、temp-path、記錄、thread-pool、pkcs11、統計資料、CGI、DNS、DNS-cache、file-cache、ACL-cache、SSL-session-cache、access-log-buffer 和 JVM (log-level 除外) 設定的變更將不會生效。執行重新配置時，將記錄需要重新啟動的所有此類變更。重新配置檔案快取需要重新啟動伺服器。

1. 按一下 **[節點]** 標籤，以檢視在伺服器中配置的節點清單。
2. 透過選取節點名稱核取方塊來選取節點。
3. 按一下 **[重新配置實例]** 按鈕以開啓頁面視窗，此視窗列出部署在該節點上的所有實例。
4. 選取實例，然後按一下 **[重新配置實例]** 按鈕以重新配置實例。

5. 檢查實例狀態是否為執行中，然後關閉視窗。

備註 – 使用 CLI

若要透過 CLI 重新配置伺服器實例，請執行以下指令：

```
wadm> reconfig-instance --user=admin --password-file=admin.pwd  
--host=serverhost --port=8989 --config=config1 serverhost
```

請參閱 CLI 參照 `reconfig-instance(1)`。

刪除伺服器實例

備註 – 只有伺服器實例不處於執行中狀態時才能夠刪除。

1. 按一下 **[配置]** 標籤，以檢視可用配置清單。
2. 從配置清單中選取配置。
3. 按一下 **[實例]** 子標籤。
4. 從 **[節點]** 區段下的已部署實例清單中選取實例。
5. 從動作下拉式清單中選取 **[刪除實例]**，以刪除選取的實例。

備註 – 使用 CLI

若要透過 CLI 刪除伺服器實例，請執行以下指令：

```
wadm> delete-instance --user=admin --password-file=admin.pwd --host=serverhost  
--port=8989 --config=config1 serverhost
```

請參閱 CLI 參照 `delete-instance(1)`。

自動配置實例

可以根據排定的事件重新配置或重新啟動實例。您可以設定特定時間和間隔，以排定自動重新配置實例。

如需排程事件，請執行以下作業：

1. 按一下 **[配置]** 標籤，然後選取配置。
2. 按一下 **[一般]** 子標籤 > **[排定的事件]** 子標籤。

▼ 增加排定的事件

1 選取配置。

從按一下 [配置] 標籤後所顯示的清單中選取配置。

2 按一下 [一般] > [排定的事件] 子標籤。

3 按一下 [新增] 按鈕。

4 配置以下特性：

■ 事件

- **重新啟動實例**—此排定的事件將重新啟動配置的所有已部署和執行中的實例。
- **重新配置實例**—此排定的事件將重新配置配置的所有已部署和執行中的實例。
- **自訂指令行**—提供將要執行的檔案之絕對路徑。

■ 排程

配置的事件啟動時間。從下拉式方塊中選取小時和分鐘值。

- **每一天**—每天在指定時間啟動指定的事件。
- **指定日期**—在指定日期啟動指定的事件。
 - 1. **星期幾**—指定從星期日到星期六之間的任何一天。
 - 2. **日期**—以逗號分隔的項目指定每月 1 日到 31 日之間的任何一天。例如 4,23,9
- **特定月份**—在指定時間和月份啟動指定的事件。指定從 1 月至 12 月之間的月份。
- **間隔**

在此時間段之後啟動指定的事件。

 - 1. **間隔小時數**—從下拉式方塊中選取小時數。
 - 2. **間隔秒數**—在文字欄位中輸入秒數。

備註 – 使用 CLI

若要透過 CLI 排定事件，請執行以下指令：

```
wadm> create-event --user=admin --password-file=admin.pwd --host=serverhost  
--port=8989 --config=config1 --time=10:10 --command=restart
```

請參閱 CLI 參照 create-event(1)。

▼ 移除排定的事件

- 1 選取配置。
從按一下 [配置] 標籤後所顯示的清單中選取配置。
- 2 按一下 [一般] > [排定的事件] 子標籤。
- 3 選取排定的事件，然後按一下 [刪除] 按鈕。

伺服器機群和叢集

前幾章介紹了配置，以及如何將配置部署到節點。在本章中，我們將設定簡單的伺服器機群和叢集。

- 第 45 頁的「[Sun Java System Web Server 中的叢集支援](#)」
- 第 45 頁的「[設定伺服器機群](#)」
- 第 47 頁的「[設定簡單叢集](#)」

Sun Java System Web Server 中的叢集支援

叢集是一組實例，跨越一個或多個節點，全部執行相同的配置並提供相同的執行階段服務集。每個叢集必須包含一台指定作為管理伺服器的伺服器。如果您有一個以上的叢集，則可以從單一主管理伺服器管理所有叢集。主管理伺服器可擷取關於所有叢集的資訊，並提供用於管理 Sun Java System Web Server (安裝在其各自的叢集中) 的介面。

備註 – 叢集中的所有實例必須同質。例如，這些實例在相同的作業系統版本 (和修補程式) 及服務軟體上執行，執行相同的 Web 伺服器配置並提供相同服務。

設定伺服器機群

若要設定叢集，首先需要安裝一台管理伺服器，以及一個或多個管理節點。您必須將管理節點分別註冊到管理伺服器中，才能對其進行管理。此動作可在節點安裝期間或安裝之後透過 wadm CLI 來執行。

▼ 設定伺服器機群

1 安裝管理伺服器和管理節點

安裝管理伺服器。您可以透過 Sun Java System Web Server 安裝程式 GUI 或透過 wadm CLI 來安裝管理伺服器。

您可以選擇 **[快速安裝]** 選項，將管理伺服器安裝在連接埠 8989 上。或者，選擇 **[自訂安裝]** 選項設定喜好設定。若要安裝管理伺服器，請在安裝程式設定畫面上選擇選項 **[將伺服器安裝為管理伺服器]**。您需要指定 SSL 連接埠，對於非 SSL 連接埠可以指定也可以不指定。

備註 – 如果指定非 SSL 連接埠，在管理伺服器節點中便會建立管理節點，並且這並不需要向管理伺服器明確註冊。

若要安裝管理節點，請選擇 **[自訂安裝]**，然後再選擇 **[將伺服器安裝為管理節點]**。指定用於安裝的連接埠。您無法選取非 SSL 連接埠，因為管理伺服器和管理節點之間的通訊一律透過安全通道。在安裝期間，將詢問您是否需要向管理伺服器註冊節點。如果在安裝期間選擇不註冊節點，則可使用 wadm CLI 來執行此動作。

備註 – 您無法透過快速安裝來安裝管理節點。

2 向管理伺服器註冊管理節點

您必須將管理節點註冊到管理伺服器，才能使其加入叢集或伺服器機群。除非將管理節點註冊到管理伺服器，否則管理節點將不會啟動。若要註冊管理節點，請透過 wadm CLI 執行以下指令。

```
wadm> register-node --user <admin-user> --port <SSL Port> --host <node name>
```

此連接埠是在管理伺服器安裝期間指定的連接埠。Host 是安裝管理伺服器所在節點的主機名稱。

此動作將向管理伺服器註冊節點。

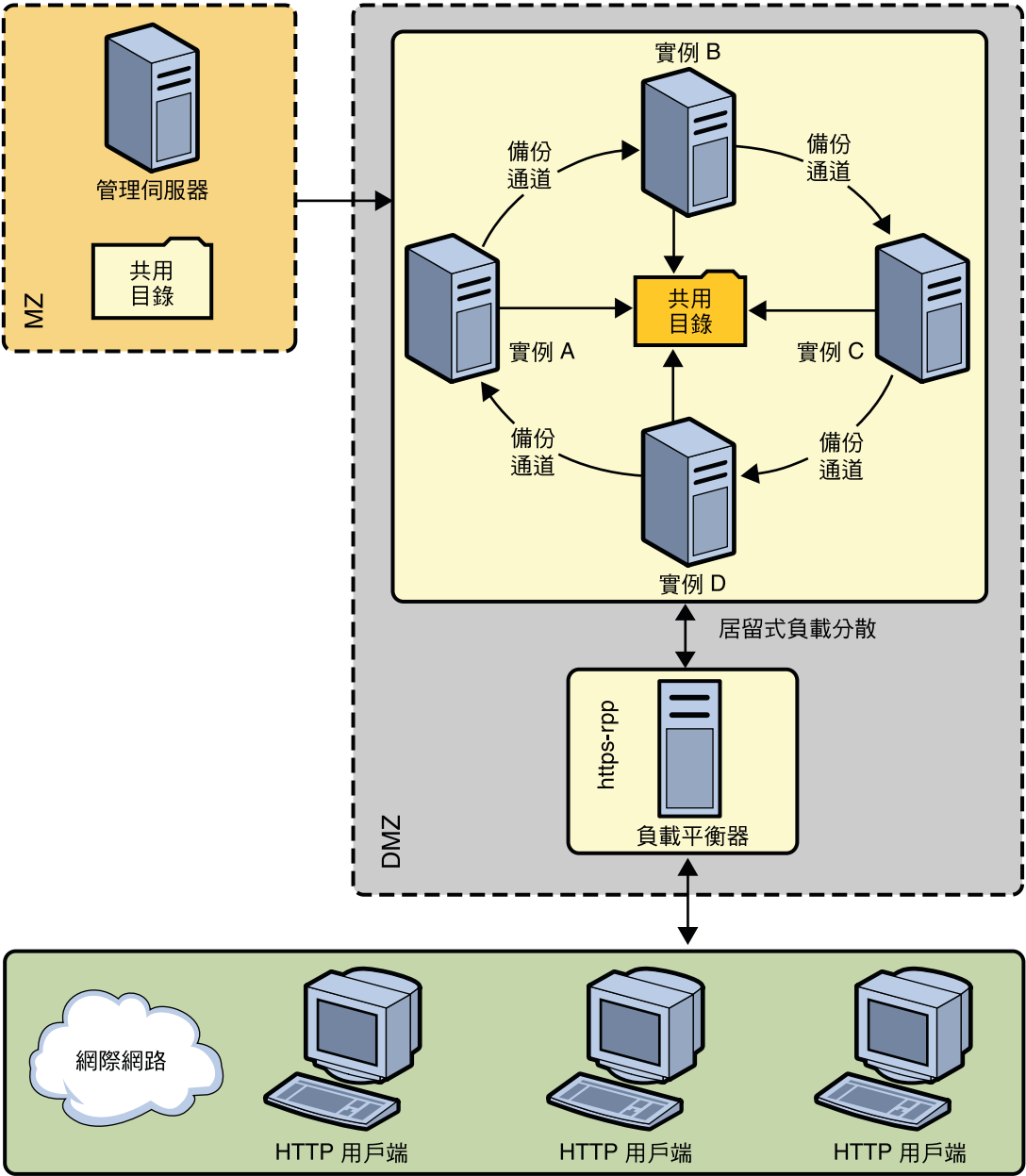
備註 – 只能從同一節點來註冊節點。您無法移至管理伺服器的 CLI 註冊任意節點。此外，您只能在 SSL 模式下向管理伺服器註冊節點。

若要為建立的叢集設定階段作業複製，請參閱第 167 頁的「配置階段作業複製」。

設定簡單叢集

爲了以此範例進行說明，我們設定一個叢集，由一個負載平衡器，一台管理伺服器及已啓用階段作業複製的四個 Web 伺服器實例所組成。階段作業複製可爲 Java Web 應用程式階段作業提供高可用性。將一個 Web 伺服器實例之記憶體中常駐的階段作業複製到另一個 Web 伺服器實例，即可實現此目的。因此，在一般作業條件下，每個階段作業至少有 2 個副本，每一個都位於單獨的 JVM 中，最理想的情況是位於單獨的機器上。

下圖描述了簡單叢集：



▼ 配置叢集

開始之前 識別以下機器：

- 機器 A —同時具有負載平衡器和管理伺服器。
- 機器 B、機器 C、機器 D 和機器 E —具有管理節點，且 Web 伺服器實例處於執行中狀態。

1 在機器 A 上安裝管理伺服器。

請參閱第 46 頁的「[設定伺服器機群](#)」，以取得有關安裝管理伺服器的資訊。一般安裝程序還將安裝 Web 伺服器實例。在此案例中，我們並不使用該實例。

2 在機器 B、機器 C、機器 D 和機器 E 上安裝管理節點。

在所有 4 台機器上安裝管理節點。向管理伺服器註冊管理節點。

3 配置 Web 應用程式。

為 Web 應用程式啟用階段作業複製。按照以下方式修改 WEB-INF/sun-web.xml 檔案：

```
<session-manager persistence-type="replicated"/>
```

4 配置實例。

- 啟動 wadm。

```
wadm --host MachineA --port 8089
```

- 為負載平衡器建立新配置。

```
wadm> create-config --http-port=8080 --server-name=SampleCluster lb
```

- 設定反向代理伺服器(負載平衡器)。

```
wadm> create-reverse-proxy --config=lb --vs=lb
-uri-prefix=/ --server="http://MachineB:8080,http://MachineC:8080,
ttp://MachineD:8080,http://MachineE:8080"
```

- 建立實例。

```
wadm> create-instance --config=lb MachineA
```

- 部署配置。

```
wadm> deploy-config lb
wadm> start-instance --config=lb
```

5 建立並啟動叢集。

建立具有四個實例的新配置。

- 為叢集建立新配置。

```
wadm> create-config --http-port=8080 --server-name=SampleCluster clusterOf4
```

- 啟用階段作業複製。

```
wadm> set-session-replication-prop --config=clusterOf4 enabled=true
```

- 增加 Web 應用程式。

```
wadm> add-webapp --config=clusterOf4 --uri=/simple webapps-simple.war
```

- 建立實例。

```
wadm> create-instance --config=clusterOf4 MachineB MachineC MachineD MachineE
```

- 啟動叢集。

```
wadm> start-instance --config=clusterOf4
```

備註 – 如果未指定 H80 指令的主機名稱，此動作將啟動已部署該配置之所有節點上的實例。

部署案例

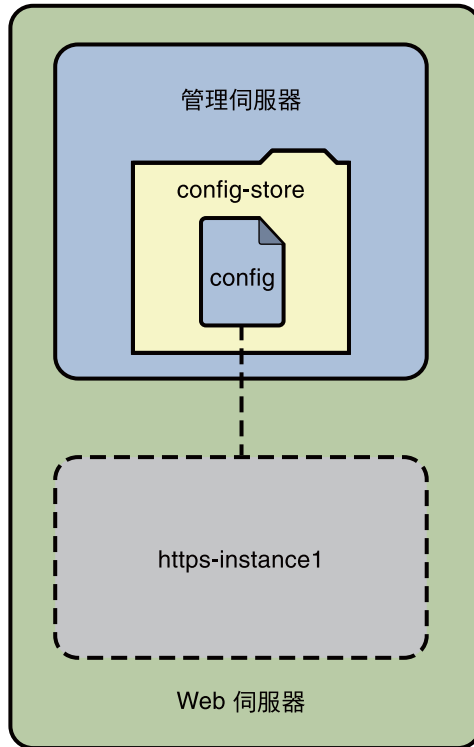
本章說明如何在單一節點和叢集環境中部署 Sun Java System Web Server 7.0。本章將討論以下主題：

- [第 51 頁的「部署架構」](#)
- [第 53 頁的「部署簡介」](#)
- [第 56 頁的「叢集環境」](#)
- [第 60 頁的「階段作業複製」](#)
- [第 63 頁的「監視叢集」](#)
- [第 63 頁的「Solaris Zones」](#)

部署架構

本節說明了單一節點部署架構。

下圖說明單一節點部署設定中的 Web Server。



在此圖中，Web Server 部署設定包含下列元件：

- **管理伺服器**- 管理伺服器是專門配置的 Web 伺服器實例。您可以在管理伺服器上部署 Web 應用程式。
- **管理節點**- 管理節點部署於伺服器機群內的節點或伺服器/主機上，並且可以與遠端管理伺服器進行通訊。您可以將管理伺服器內的可用伺服器配置部署到此節點上。伺服器機群內的所有管理節點都必須是同質的。也即是說，所有節點都必須使用同一作業系統並具有相同的硬體架構。
- **配置**- 配置是指 Web Server 實例的所有可配置元素集，如 Web 應用程式、配置檔案和搜尋集合索引。可以建立、修改或刪除配置。Web Server 可以管理多個配置。您可針對一個配置建立多個實例。部署已修改的配置將更新該配置的實例。
- **config-store**，這是一個基於檔案系統的儲存庫，所有配置都儲存在其中。



注意 – 請勿編輯 `config-store` 目錄下的任何檔案。此目錄下的檔案由 Sun Java System Web Server 建立以供內部使用。

如果您必須手動編輯 `config-store` 目錄下的配置檔案，請使用 `wadm deploy-config` 指令來部署配置。

如需有關使用此指令的更多資訊，請參閱「Sun Java System Web Server 7.0 CLI Reference Manual」。

- **實例** - 實例是指指定節點上 Web 伺服器的環境，包括其配置、記錄檔以及其他執行階段工件 (例如鎖定資料庫、快取和暫存檔)。出於管理目的，可以對實例執行啟動、停止、重新啟動或動態重新配置操作。

部署簡介

您可以考慮在單一節點上部署 Web Server 以達到以下目的：

- 託管簡單的 Web 或 CGI 應用程式。
- 開發和測試 Web 應用程式。

以下流程圖提供在節點上部署 Web Server 的概要說明：

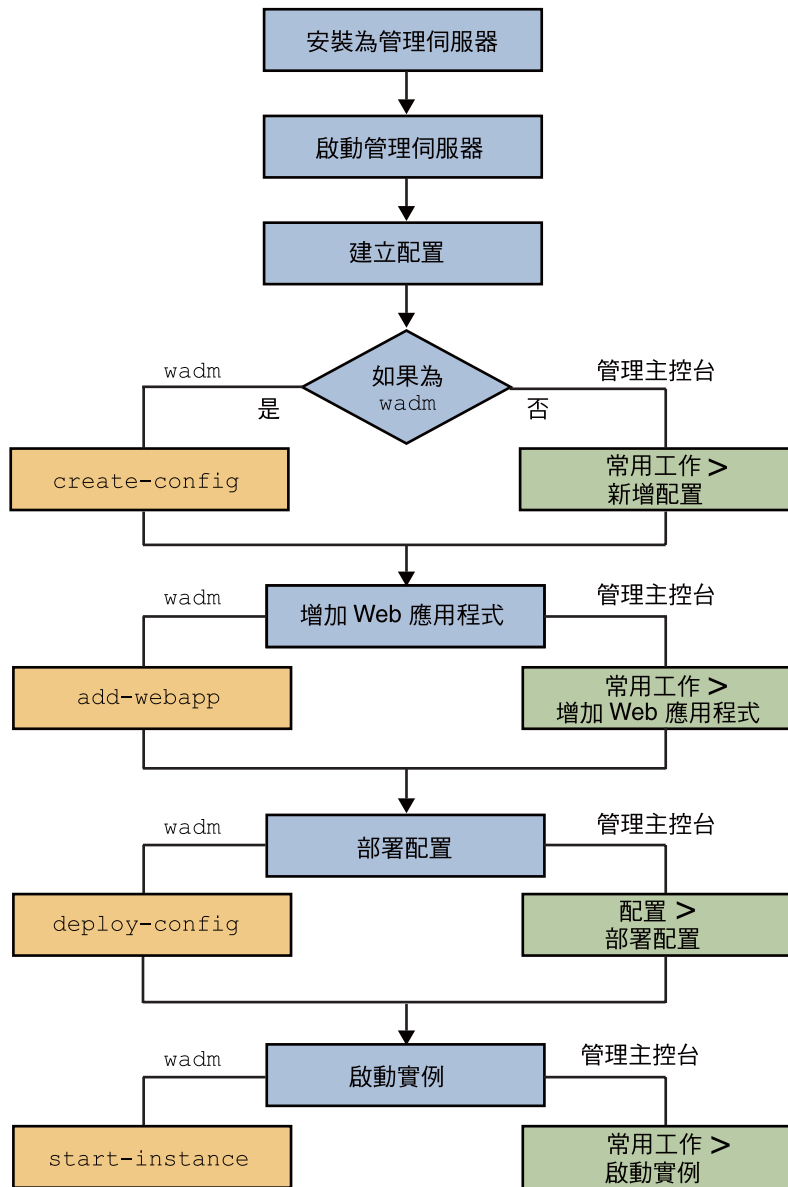


圖 4-1 說明在單一節點上部署 Web Server 的流程圖

部署程序將在以下各節中說明：

- 第 55 頁的「部署前置作業需求」
- 第 55 頁的「部署 Web Server」

部署前置作業需求

若要在單一節點上部署 Web Server，請執行以下作業讓系統做好準備：

1. 在節點上安裝 Web Server。

如果在安裝 Web Server 時選擇 [快速安裝] 選項，將會建立以下預設實體：

- Administration Server。
- 建立包含一個 HTTP 偵聽程式和虛擬伺服器的預設配置。配置和虛擬伺服器的名稱與主機名稱相同。
- 預設配置的實例。

如需有關安裝 Web Server 的資訊，請參閱「Sun Java System Web Server 7.0 Installation and Migration Guide」中的第 2 章「Installing the Web Server」。

如需有關支援的平台與系統需求的資訊，請參閱「Sun Java System Web Server 7.0 版本說明」中的「支援的平台」。

2. 啟動 Administration Server。

Administration Server 在指定的 SSL 連接埠上開始執行。

部署 Web Server

使用以下程序在節點上部署 Web Server：

1. 您可以使用預設配置或建立新配置。

如果要建立新配置，請為配置指定唯一名稱。新配置會建立虛擬伺服器和預設的 HTTP 偵聽程式。

備註 – 如果要使用管理主控台建立配置，精靈將提示您建立新實例。如果要使用 CLI，則必須使用 `create-instance` 指令明確建立配置實例。

所有配置都儲存在 `<install_dir>/admin-server/` 目錄下的 `config-store` 目錄中。



注意 – 請勿編輯 `config-store` 目錄下的任何檔案。此目錄下的檔案由 Sun Java System Web Server 建立以供內部使用。

2. 部署已修改的配置。

叢集環境

叢集是多個伺服器實例的群組，跨越一個以上的節點，全部執行相同的配置。叢集中的所有實例協同作業，以提供高可用性、穩定性和可延伸性。

藉由負載平衡，叢集可提供容錯移轉和階段作業複製功能，讓服務得以不中斷、階段作業資料擁有持續性。

硬體和軟體要求

依據本節說明的使用案例，Web Server 叢集由以下實體組成：

- 1) 四個實例 (在四個相同節點上執行)
- 2) 一台管理伺服器
- 3) 一台用於對 HTTP 請求進行負載平衡的反向代理伺服器

若要設定叢集，需要兩個以上安裝相同作業系統版本和修補程式的相同節點。例如，如果選取安裝有 Solaris® 9 SPARC® 作業系統的機器，則叢集中的其他機器也必須安裝有 Solaris 9 SPARC。

如需有關支援的平台和修補程序需求的資訊，請參閱「Sun Java System Web Server 7.0 版本說明」。

下圖說明了叢集環境。

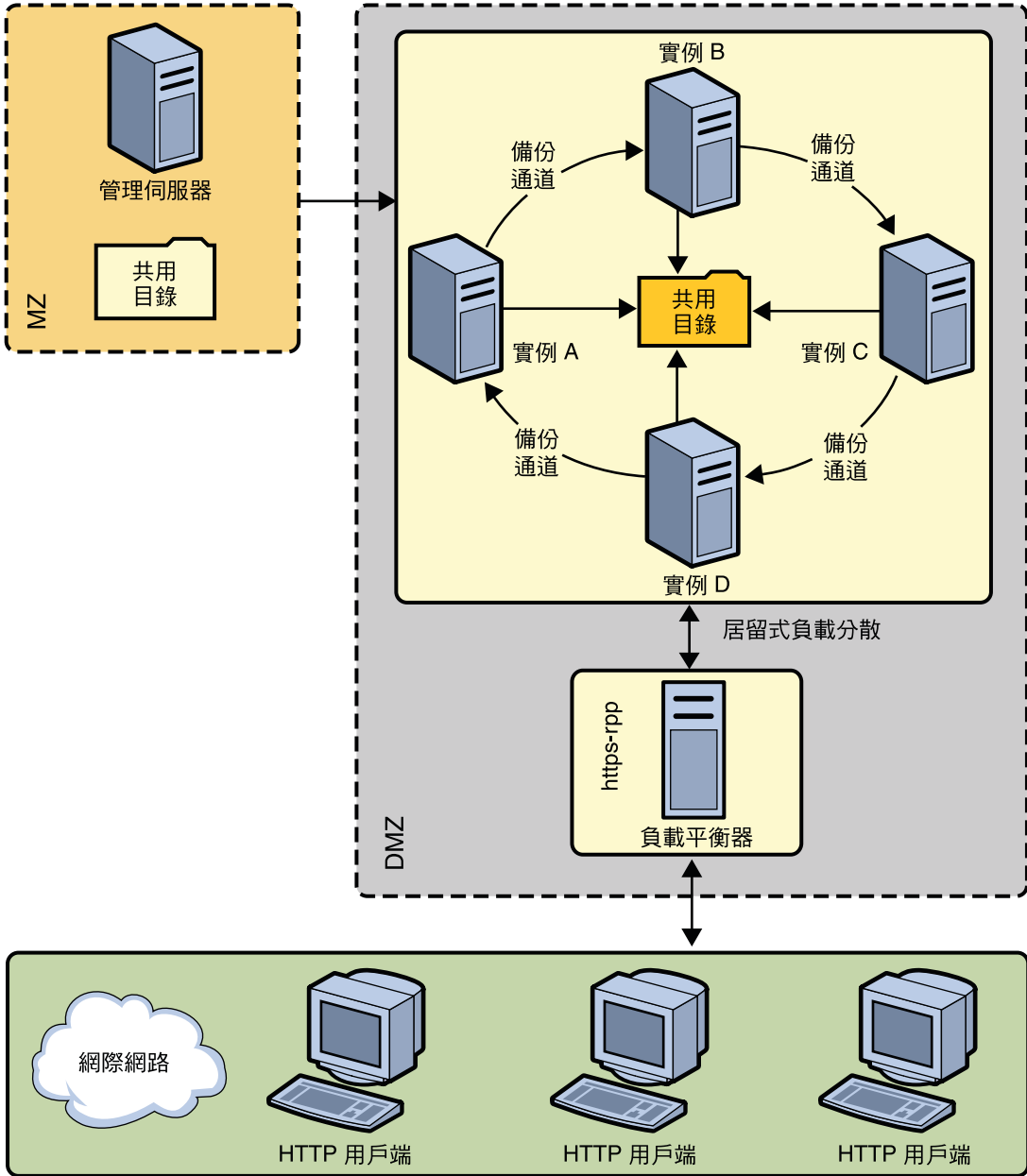


圖 4-2 叢集設定

在此圖中，節點是在非警戒區域 (DMZ) 中配置的。管理伺服器是在防火牆 (警戒區域) 後配置的，用於限制和避免所有人存取管理伺服器。另一個節點配置為反向代理伺服器。反向代理伺服器位於 DMZ 之內，以提高安全性。

備註 - 僅 Solaris 10 作業系統支援 Solaris Zone 功能。

設定叢集

本節說明設定叢集和啓用反向代理伺服器，來支援對 HTTP 請求進行負載平衡的程序。

以下流程圖說明設定叢集的程序。

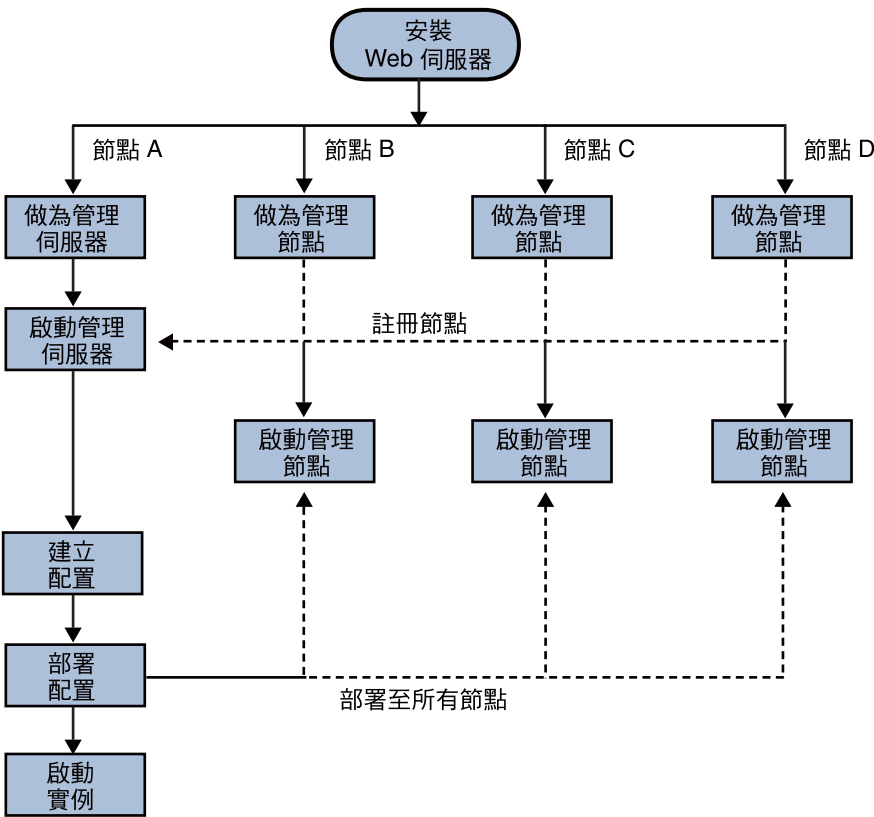


圖 4-3 說明叢集設定的流程圖

1. 在其中一個節點上，將 Web Server 安裝為叢集中的 Administration Server。

2. 在其他三個節點上安裝 Web Server。選取將 Web Server 安裝為管理節點的選項。在安裝期間，請選擇向伺服器註冊節點的選項。
3. 請確定管理伺服器是使用 SSL 連接埠進行通訊，因為只能在安全模式下向伺服器註冊管理節點。
4. 請確定安裝 Administration Server 和管理節點的所有節點，都有相同的系統日期與時間。建立與伺服器關聯的憑證時，需依據 Administration Server 安裝所在節點的系統日期和時間。如果管理節點的系統日期早於 Administration Server，註冊將會失敗，因為 Administration Server 的憑證尚未有效。同理，如果憑證已過期，可能仍被視為有效。
5. 從 `install_dir/admin-server/bin/` 目錄啟動 Administration Server。


```
install_dir/admin-server/bin>./startserv
```
6. 從管理節點啟動 wadm 指令行工具。wadm 指令行工具位於 `install_dir/bin` 目錄中。


```
install_dir/bin>./wadm
```
7. 向 Administration Server 註冊每個管理節點。使用 `register-node` 指令向伺服器註冊每個節點。
例如：

```
./wadm register-node -user=admin --host=abc.sfbay.sun.com --port=8989
```

其中，

`abc.sfbay.sun.com` 是向其註冊管理節點的 Administration Server 之主機名稱。

`port` 是 Administration Server 的 SSL 連接埠號。

8. 系統將提示您輸入管理密碼。請輸入 Administration Server 的管理密碼。

Administration Server 與管理節點彼此互相認證，即 Administration Server 信任管理節點的伺服器憑證，管理節點信任 Administration Server 提供的用戶端憑證。在管理節點註冊期間，管理伺服器會產生該管理節點的伺服器憑證，此憑證之後會被下載並安裝到管理節點上。伺服器憑證的核發者也會安裝在管理節點上。

備註 – 註冊只能透過 SSL 來完成。

如需有關註冊節點的資訊，請參閱「Sun Java System Web Server 7.0 Installation and Migration Guide」中的「Registering the Administration Node From the Command-Line」。

9. 從 `install_dir/admin-server/bin/` 目錄使用 `startserv` 指令來啟動所有管理節點。
10. 使用管理主控台或 CLI 在 Administration Server 中建立新配置。
為新配置提供配置資訊，如配置名稱、HTTP 偵聽程式連接埠及伺服器名稱。
11. 在所有節點上建立配置實例。

12. 啟動所有節點上的實例。

備註 – Web Server 為延伸或減少叢集提供靈活性。您可以隨時在叢集中增加或移除實例。

配置用於負載平衡的反向代理伺服器

Web Server 7.0 提供先進的內建負載平衡器，即反向代理伺服器。反向代理伺服器是伺服器機群中 Web Server 的閘道。透過配置反向代理伺服器，請求會轉寄到多台配置類似的 Web Server。

請在 Web Server 7.0 中使用以下程序啟用反向代理伺服器：

1. 在要用於配置反向代理伺服器的節點上安裝 Web Server。
2. 建立配置。例如 `rp`。
3. 使用管理主控台，選取 [配置] > [虛擬伺服器] > [內容處理] > [反向代理伺服器] 標籤。按一下 [新增] 按鈕。
4. 輸入叢集中所有機器的反向代理伺服器 URI 和伺服器 URL (以逗號分隔)。
伺服器 URL 的輸入格式為 *hostname:portnumber*。
5. 儲存變更。
6. 部署已修改的配置，將變更套用至此配置。
7. 啟動此已修改配置的所有實例。

此舉可完成反向代理伺服器的配置，以便對 HTTP 請求進行負載平衡。

階段作業複製

階段作業複製機制用於在不同實例之間，複製儲存在階段作業中的資料。但是所複製的實例必須屬於同一個叢集。在叢集環境中啟用階段作業複製時，整個階段作業資料會複製到複製的實例上。但是階段作業複製操作並不會複製階段作業中的不可串列屬性，以及任何實例專屬的資料。

階段作業複製和負載平衡功能可為 Web 應用程式提供良好的容錯移轉功能。

階段作業複製與容錯移轉操作

本節將詳細說明階段作業複製操作。

Web 請求結束後，Web Server 將決定是否需要透過儲存在伺服器配置檔案 `server.xml` 中的階段作業複製配置，來複製階段作業資料。

現在研究由四個實例形成一個叢集、並在 Administration Server 上啓用階段作業複製的使用案例。

在由執行於四個節點上的四個實例 (A、B、C 和 D) 所組成的 Web Server 叢集中，階段作業複製的程序如下：

- 實例 A 爲 D 的備份，B 爲 A 的備份，C 爲 B 的備份，D 爲 C 的備份。這樣便會形成完整的備份環。
- 叢集中的每個實例都會記錄一份靜態清單，其中列出了叢集中的所有實例以及使用中的備份實例。
- 階段作業資料會在每個請求結束後同步傳送至備份實例，依配置而定。

Web Server 叢集環境中的容錯移轉程序運作方式如下：

- 如果實例 A 失敗，負載平衡器會將以實例 A 爲目的地的內送 Web 請求，重新導向至叢集中其餘的實例，並且備份環會以下列方式重新配置：
 - D 偵測到其備份 A 當機時，便會在順序清單上選取 A 的下一個實例做爲新備份實例。
 - 在此情況下將會選取 B，並且 D 會與 B 建立新的備份連線。此時 B 持有兩個備份：A 的唯讀備份和 D 的使用中備份。
 - 此時完整的備份環形成：B 備份到 C，C 備份到 D，D 備份到 B。
 - 當故障的實例 A 再次可用時，會向指定的備份實例 B 傳送重新連結訊息以重新連結備份環，並與 B 建立備份連線。
 - D 收到 A 傳回的 ping 成功結果或收到來自 A 的訊息，進而偵測到 A 在線上時，
 - D 隨後與 A 建立備份連線，並終止與 B 的備份連線。

Web Server 7.0 在階段作業複製中不支援以下功能：

- 回復兩個以上同步故障的實例。
- 兩次失敗之間的時間必須大於讓恢復的實例完全回復所需的時間。
- 對一個以上的實例進行階段作業備份。在一般操作中，任何階段作業僅有兩個備份：主要階段作業和備份階段作業。
- 階段作業持續性：階段作業僅備份在其他實例的記憶體中，以便進行容錯移轉
- Web Server 僅支援 Java Web 應用程式的階段作業複製。如果使用的是非 Java 應用程式 (如 CGI 或 PHP)，則無法複製階段作業資料。

啓用階段作業複製

您可以使用管理主控台或 CLI 在叢集中啓用階段作業複製。啓用階段作業複製前，請確定瀏覽器已啓用 cookie。

server.xml 檔案包含與階段作業複製相關的資訊。以下是已啓用階段作業複製的 server.xml 檔案範例：

```
<cluster>
  <local-host>hostA</local-host>
  <instance>
    <host>hostB</host>
  </instance>
  <instance>
    <host>hostC</host>
  </instance>
  <instance>
    <host>hostD</host>
  </instance>
  <instance>
    <host>hostA</host>
  </instance>
  <session-replication/>
</cluster>
```

如果使用的不是以下元素的預設值，則在 server.xml 配置檔案中將無法使用這些元素的項目。

連接埠號 (預設為 1099)

通訊協定 (預設為 jrmf)

已加密 (預設為 false)

Getattribute 觸發複製 (預設為 true)

複製探索最大躍點數 (預設為 -1)

啟動探索逾時 (預設為 ?)

Cookie 名稱 (預設為 CLUSTERSESSIONLOCATOR)

如需有關這些階段作業複製特性的更多資訊，請參閱「Sun Java System Web Server 7.0 Administrator's Configuration File Reference」。

為階段作業複製配置 Web 應用程式

若要啓用伺服器以複製階段作業，還必須為階段作業複製啓用 Web 應用程式。

1. 若要為 Web 應用程式啓用階段作業複製，請修改位於 <web-application>/WEB-INF 目錄中的 sun-web.xml 配置檔案。

按照下列方式在 sunweb.xml 中執行所需的修改：

將元素 <session-manager/> 變更為 <session-manager persistence-type="replicated">

以下是已啓用階段作業複製的 `sun-web.xml` 檔案範例：

```
<sun-web-app>
  <session-config>
    <session-manager persistence-type="replicated">
    </session-manager>
  </session-config>
</sun-web-app>
```

2. 修改 `sunweb.xml` 檔案後，請重建 Web 應用程式或重新以 jar 壓縮應用程式，以建立 Web 應用程式歸檔 (war 檔案)。
3. 重新啓動所有實例，以使 Web 應用程式在所有實例上均可用。
4. 您可以從叢集中的所有節點存取 Web 應用程式。若要存取 Web 應用程式，請在瀏覽器中鍵入以下項目：

`http://webserver-name/webapplication-name/`

備註 – 所有節點均可存取的目錄，最適用於儲存應用程式以進行部署。但是，Administration Server 並不需要可對此目錄進行存取。建議對大小超過 1 MB 的 Web 應用程式進行基於目錄的部署。

若要建立搜尋集合，請確保搜尋集合位於所有節點均可存取的共用目錄中。

監視叢集

Administration Server 可監視叢集中的所有實例。Web Server 的監視功能可提供有關執行階段元件的狀態，和可用於執行以下操作的程序之資訊：

- 識別效能瓶頸
- 調校系統以取得最佳效能
- 協助進行容量規劃
- 預測失敗
- 在失敗時分析根本原因

Solaris Zones

Solaris Zones 是 Solaris 10 的一項應用程式和資源管理功能。Zone 環境通常由各種資源組成，如程序管理、記憶體、網路配置、檔案系統、套裝軟體登錄、使用者帳號及共用程式庫，在某些情況下還包括安裝的應用程式。Zones 功能提供了在 Solaris 實例內建立虛擬化作業系統環境的方法，允許一個或多個程序單獨執行，與系統上的其他作業隔離。它還提供了一個抽象層，將應用程式與部署這些應用程式的機器之實體屬性分

隔開來，如實體裝置路徑、網路介面名稱及網路路由表。這種隔離可防止在指定區域內執行的程序，去監視或影響在其他區域中執行的程序，而無論使用者 ID 和其他憑證資訊為何。

區域是一個**沙箱**，一個或多個應用程式均可在其中執行，而不會影響系統的其餘部分或進行互動。

如需有關 Solaris Zones 的詳細資訊，請參閱

<http://docs.sun.com/app/docs/doc/817-1592> 上的「*System Administration Guide — Solaris Containers-Resource Management and Solaris Zones*」。

使用虛擬伺服器

- 第 65 頁的「虛擬伺服器簡介」
- 第 65 頁的「使用案例」
- 第 67 頁的「管理虛擬伺服器」
- 第 69 頁的「配置 HTTP 偵聽程式」

虛擬伺服器簡介

使用虛擬伺服器時，您可以藉由單一安裝的伺服器，為公司或個人提供網域名稱、IP 位址以及某些伺服器監視功能。對於使用者而言，雖然是您提供的硬體和基本 Web 伺服器維護，但使用者就像使用自己的 Web 伺服器一樣。

所有虛擬伺服器都指定了 HTTP 偵聽程式。收到新請求時，伺服器會根據配置的 HTTP 偵聽程式確定將請求傳送至哪台虛擬伺服器。

使用案例

Sun Java System Web Server 的伺服器實例可包含任意數目的 HTTP 偵聽程式，包括安全的與不安全的。您可以同時擁有基於 IP 位址的虛擬伺服器與基於 URL 主機的虛擬伺服器。

每台虛擬伺服器都可以 (但並非必須) 有各自的 ACL 清單、各自的 `mime.types` 檔案，以及各自的 Java Web 應用程式集。

此設計對於為各種應用程式配置伺服器提供了最大靈活性。以下範例討論一些適用於 Sun Java System Web Server 的可能配置。

預設配置

新安裝 Sun Java System Web Server 後，您將會有一個伺服器實例。此伺服器實例僅具有一個 HTTP 偵聽程式乙太網路，可在連接埠 80 (或安裝時選取的任何連接埠) 上偵聽配置給您的電腦的任何 IP 位址。

您區域網路中的某個機制，可以為配置給您的電腦的每個位址建立名稱至位址的對映。在以下範例中，電腦有兩個網路介面：位址 127.0.0.1 上的回返介面 (即使沒有網路卡亦存在的介面) 以及位址 10.0.0.1 上的乙太網路介面。

經由 DNS，名稱 `example.com` 將對映至 10.0.0.1。偵聽通訊端配置為在連接埠 80 上偵聽配置給該機器的任何位址 (「ANY:80」或「0.0.0.0:80」)。

在此配置中，與以下位址的連線會到達伺服器並由虛擬伺服器 VS1 提供服務

- `http://127.0.0.1/` (在 `example.com` 上啟動)
- `http://localhost/` (在 `example.com` 上啟動)
- `http://example.com/`
- `http://10.0.0.1/`

將此配置用於傳統 Web 伺服器。您不需要增加其他虛擬伺服器或 HTTP 偵聽程式。

安全伺服器

請參閱第 83 頁的「為伺服器配置 SSL」。

企業網路主機作業

Sun Java System Web Server 的一項更複雜配置是該伺服器託管少數虛擬伺服器，以進行企業內部網路部署。例如，您有三個內部網站，員工可以從中查找其他使用者的電話號碼、查看校園地圖並追蹤他們向資訊服務部門所提出請求的狀態。之前 (在此範例中)，這些網站託管於三台不同的電腦，分別對映於名稱 `phone.example.com`、`maps.example.com` 和 `is.example.com`。

若要将硬體和管理耗用降至最低，可以將三個網站全部合併到位於機器 `example.com` 上的一台 Web 伺服器中。您可以採用以下兩種方法進行設定：使用基於 URL 主機的虛擬伺服器或使用單獨的 HTTP 偵聽程式。兩者均有明顯的優勢和劣勢。

使用基於 URL 主機的虛擬伺服器進行企業網路主機作業

儘管基於 URL 主機的虛擬伺服器易於設定，但其具有以下劣勢：

- 若要在配置中支援 SSL，則需要使用萬用字元憑證的非標準設定。
- 基於 URL 主機的虛擬伺服器無法與舊的 HTTP 用戶端協同作業。

您也可以設定基於 IP 位址的配置，其中每個位址都有一個 HTTP 偵聽程式：

使用個別的 HTTP 偵聽程式進行企業內部網路託管

基於 IP 位址的虛擬伺服器之優勢為：

- 它們可與不支援 HTTP/1.1 主機標頭的舊用戶端協同作業。
- 可直接提供 SSL 支援。

劣勢有：

- 需要對主機電腦 (實際或虛擬網路介面的配置) 進行配置變更
- 不能針對成千上萬虛擬伺服器進行配置調節

兩種配置均需要為三個名稱設定名稱至位址的對映。在基於 IP 位址的配置中，每個名稱都對映至不同位址。必須將主機電腦設定為接收所有這些位址上的連線。在基於 URL 主機的配置中，所有名稱都可對映至同一位址，即機器最初的位址。

包含多個 HTTP 偵聽程式的配置所改善的效能可能最少，因為伺服器不必找出請求傳入所在的位址。但是，由於其他接收器執行緒，使用多個 HTTP 偵聽程式也會產生額外的耗用 (記憶體與排程)。

大量主機作業

大量主機作業指的是您可以啟用許多低流量虛擬伺服器的配置。例如，託管許多低網路流量的個人首頁的 ISP 即屬於此類。

虛擬伺服器通常是基於 URL 主機。例如，您可以讓一種配置僅允許靜態內容，而另一種配置則允許靜態內容和 CGI。

管理虛擬伺服器

- [第 67 頁的「增加虛擬伺服器」](#)
- [第 68 頁的「配置虛擬伺服器」](#)
- [第 69 頁的「複製虛擬伺服器」](#)

增加虛擬伺服器

▼ 增加虛擬伺服器

- 開始之前
- 您應已建立/識別需要為其建立虛擬伺服器的配置。
 - 您應已建立/識別 HTTP 偵聽程式。
 - 您應已識別新虛擬伺服器的主機。

- 1 選取需要從中增加虛擬伺服器的配置。您可以從 [配置] 標籤所顯示的配置清單中選取配置。

- 2 按一下[虛擬伺服器]標籤>[新增]按鈕。
- 3 您將看到快顯式精靈頁面，此精靈將引導您完成虛擬伺服器配置過程。從精靈頁面執行以下作業：
 - 輸入新虛擬伺服器資訊：
 - a. 輸入用於識別新虛擬伺服器的名稱。名稱可以為字母數字式，也可包括點號(.)、破折號(-)和底線(_)字元。
 - b. (可選)輸入要增加到新虛擬伺服器的主機清單。
 - c. (可選)輸入虛擬伺服器的文件根目錄。
 - 選取新配置的虛擬伺服器的 HTTP 偵聽程式。您可以選擇現有 HTTP 偵聽程式，或建立新的 HTTP 偵聽程式。
- 4 現在，您將看到精靈摘要頁面。若要變更配置，請按[上一頁]返回至上一個頁面。按一下[完成]，以完成新虛擬伺服器的配置過程。
- 5 此時您將看到[結果]頁面。如果您發現任何錯誤，請返回至精靈的前幾個頁面，再次對虛擬伺服器進行配置。

備註 – 使用 CLI

若要透過 CLI 增加虛擬伺服器，請執行以下指令：

```
wadm> create-virtual-server --user=admin --password-file=admin.pwd  
--host=serverhost --port=8989 --config=config1 --document-root=../docs config1_vs_1
```

請參閱 CLI 參照 create-virtual-server(1)。

配置虛擬伺服器

- [第 68 頁的「配置虛擬伺服器」](#)

若要配置虛擬伺服器的一般設定，請執行以下作業：

▼ 配置虛擬伺服器

- 1 選取配置。

從配置清單中選取配置。按一下 [配置] 標籤，以取得可用配置的清單。
- 2 選取虛擬伺服器。

從虛擬伺服器清單中選取虛擬伺服器。按一下 [虛擬伺服器] 標籤，以取得所選配置的可用虛擬伺服器。

- 3 按一下[一般]標籤。配置以下設定。
 - 已啓用—是否在執行階段啓用虛擬伺服器。
 - 文件根目錄—虛擬伺服器的文件根目錄路徑，將在其中儲存虛擬伺服器的資料。這包括展開的 Web 應用程式目錄和記錄檔。
 - 主機—您可以輸入一個以上的 URL 主機 (以逗號分隔)。

複製虛擬伺服器

若要複製虛擬伺服器，請執行以下作業：

▼ 複製虛擬伺服器

- 1 選取配置。
從配置清單中選取配置。按一下 [配置] 標籤，以取得可用配置的清單。
- 2 選取虛擬伺服器。
從虛擬伺服器清單中選取虛擬伺服器。按一下 [虛擬伺服器] 標籤，以取得所選配置的可用虛擬伺服器。
- 3 按一下[複製]按鈕。
提供新虛擬伺服器的名稱。

備註 – 使用 CLI

若要透過 CLI 複製虛擬伺服器，請執行以下指令：

```
wadm> copy-virtual-server --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --vs=config1_vs_1 copiedVs
```

copiedVS 為新虛擬伺服器的名稱。

請參閱 CLI 參照 `copy-virtual-server(1)`。

配置 HTTP 偵聽程式

- [第 70 頁的「建立 HTTP 偵聽程式」](#)
- [第 70 頁的「配置 HTTP 偵聽程式」](#)

伺服器透過 HTTP 偵聽程式接受 HTTP 請求，然後將請求轉寄給已配置的虛擬伺服器。此頁面可讓您增加和配置 HTTP 偵聽程式。

HTTP 偵聽程式必須具有唯一的連接埠號和 IP 位址組合。您可以使用 IPv4 或 IPv6 位址。如果將 IP 位址設定為「*」，則會建立可偵聽該連接埠上所有 IP 位址的 HTTP 偵聽程式。

建立 HTTP 偵聽程式

您可以透過執行以下步驟為虛擬伺服器建立新的 HTTP 偵聽程式，用以處理內送 HTTP 請求：

1. 按一下 [配置] 標籤下的 [虛擬伺服器] 標籤。
2. 按一下 [HTTP 偵聽程式] 子標籤，以檢視已配置的 HTTP 偵聽程式清單。
3. 按一下 [新增] 按鈕，以快顯用於建立新 HTTP 偵聽程式的精靈頁面。

請在精靈頁面中提供以下資訊。

- **名稱**—新 HTTP 偵聽程式的名稱。
- **連接埠**—HTTP 偵聽程式要與其連結並用於偵聽內送 HTTP 請求的連接埠。
- **IP 位址**—有效的 IPv4 或 IPv6 位址。「*」表示 HTTP 偵聽程式將偵聽所配置連接埠上的所有 IP 位址。
- **伺服器名稱**—輸入伺服器的名稱，例如 *sales.mycomp.com*。
- **預設虛擬伺服器**—從下拉式清單中選取虛擬伺服器。此動作會關聯所選虛擬伺服器的新 HTTP 偵聽程式。
- **說明 (可選擇)**—輸入 HTTP 偵聽程式的簡短描述。

備註 – 使用 CLI

若要透過 CLI 建立 HTTP 偵聽程式，請執行以下指令。

```
wadm> create-http-listener --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --listener-port=18003 --config=config1 --server-name=config1.com
--default-virtual-server-name=config1_vs_1 config1_ls_1
```

請參閱 CLI 參照 `create-http-listener(1)`。

配置 HTTP 偵聽程式

您可以透過執行以下作業來編輯現有的 HTTP 偵聽程式設定：

1. 按一下 [伺服器配置] 下的 [虛擬伺服器] 標籤，以編輯現有的 HTTP 偵聽程式設定。
2. 按一下 [HTTP 偵聽程式] 子標籤，以檢視已配置的 HTTP 偵聽程式清單。
3. 在 [偵聽程式名稱] 表格欄下，按一下需要編輯其設定的 HTTP 偵聽程式。

您可以編輯 HTTP 偵聽程式的一般設定和安全性相關的設定。

修改 HTTP 偵聽程式參數

按一下 [一般] 標籤，以編輯基本和進階的 HTTP 偵聽程式設定。請配置以下選項：

- **名稱**—新 HTTP 偵聽程式的名稱
- **連接埠**—HTTP 偵聽程式與之連結並用於偵聽內送 HTTP 請求的連接埠。
- **IP 位址**—有效的 IPv4 或 IPv6 位址。「*」表示 HTTP 偵聽程式將偵聽所配置連接埠上的所有 IP 位址。
- **伺服器名稱**—輸入伺服器的名稱，例如 *sales.mycomp.com*。

選取 [進階] 區段下的 [配置進階設定] 選項，以編輯 HTTP 偵聽程式進階設定。請配置以下選項：

- **接收器執行緒**—專門用於接受由此偵聽程式接收之連線的執行緒數目。有效值為 1 到 128。
- **協定系列**—偵聽程式使用的協定。請勿修改此值。預設為 HTTP。
- **偵聽佇列大小**—作業系統偵聽佇列暫存區的最大大小。
- **接收緩衝區大小**—作業系統通訊端接收緩衝區的大小 (以位元組為單位)。
- **傳送緩衝區大小**—作業系統通訊端傳送緩衝區的大小 (以位元組為單位)。
- **區段化 I/O**—決定 HTTP 偵聽程式通訊端是否處於區段化模式。依預設停用。

憑證與金鑰

本章說明使用憑證與金鑰認證來保護 Sun Java System Web Server 的安全。還描述如何啓動專用於保護資料、拒絕入侵者和允許存取所需項目之各種安全功能。

閱讀本章之前，您應當熟悉公開金鑰加密的基本概念。這些概念包括加密與解密、公開金鑰與私密金鑰、數位憑證及加密協定。

- 第 73 頁的「使用憑證進行認證」
- 第 74 頁的「憑證金鑰類型」
- 第 76 頁的「建立自我簽署的憑證」
- 第 76 頁的「管理憑證」
- 第 80 頁的「管理憑證撤銷清單 (CRL)」
- 第 82 頁的「設定內部記號的密碼」
- 第 83 頁的「爲伺服器配置 SSL」
- 第 83 頁的「啓用配置的 SSL 密碼」
- 第 84 頁的「啓用 HTTP 偵聽程式的安全性」

使用憑證進行認證

認證是確認識別的程序。在網路互動的環境下，認證是一方對另一方進行的身份識別。憑證是支援認證的一種方式。

憑證包括的數位資料可指定個人、公司或其他實體的名稱，並證明憑證中包含的公開金鑰屬於該實體。用戶端和伺服器都可以擁有憑證。

憑證由憑證授權單位或 CA 核發並以數位方式簽署。CA 可以是透過網際網路出售憑證的公司，也可以是負責爲企業內部網路或企業間網路核發憑證的部門。您可以決定將充分信任的 CA 做爲驗證其他使用者身份的機構。

除憑證識別的公開金鑰和實體名稱外，憑證還包括過期日期、核發憑證的 CA 名稱，以及核發憑證之 CA 的「數位簽名」。

備註 – 必須首先安裝伺服器憑證，然後才能啟動加密。

伺服器認證

伺服器認證是指用戶端對伺服器所進行的身份識別，亦即識別應負責特定網路位址上伺服器的組織。

用戶端認證

用戶端認證是由伺服器對用戶端所進行的身份識別，亦即識別應使用用戶端軟體的人。用戶端可以擁有多個憑證，就像使用者可以有數個不同的識別部分一樣。

憑證金鑰類型

除了 RSA 金鑰，Sun Java System Web Server 7.0 還新支援橢圓曲線加密 (ECC)。

ECC 這款新興的公開金鑰加密系統極受矚目，因為與傳統的加密系統 (如 RSA) 相比，ECC 以更小的金鑰提供同等安全性，這樣計算更快、耗電更低，還可以節省記憶體與頻寬。橢圓曲線加密 (ECC) 已經取得美國政府的認可。

現在，您可以選取是產生憑證請求，還是產生使用 RSA 金鑰或 ECC 金鑰的自我簽署憑證。

以 RSA 金鑰來說，則可以提供不同大小的金鑰 (金鑰大小越大，表示加密越強。預設金鑰大小為 1024)。對於 ECC 金鑰，您應當選擇要用哪種曲線產生金鑰對。多個組織已命名了許多曲線 (ANSI X9.62、NIST、SECG)，Sun Java System Web Server 7.0 支援目前指定的所有曲線。

如果您要請求來自 CA 的憑證 (而不是使用自我簽署憑證)，一定要先與您喜好的 CA 連絡，以取得其有關 ECC 用法的最新資訊。詢問他們是否針對您的使用情況建議使用特定的 ECC 曲線。如果 CA 或組織內部策略無法指示您如何選擇曲線，下面有一些建議。請記住，ECC 是一種新興技術，因此從此文件編寫完成起，對於特定使用情況的曲線建議可能已經有所改變。

下方列出一些支援的 ECC 曲線：

```
prime256v1  
secp256r1  
nistp256  
secp256k1
```

secp384r1
nistp384
secp521r1
nistp521
sect163k1
nistk163
sect163r1
sect163r2
nistb163
sect193r1
sect193r2
sect233k1
nistk233k1
nistk233
sect233r1
nistb233
sect239k1
sect283k1
nistk283
sect283r1
nistb283
sect409k1
nistk409
sect571k1
nistk571
sect571r1
nistb571
secp160k1
secp160r1
secp160r2
secp192k1
secp192r1
nistp192
secp224k1
secp224r1
nistp224
prime192v1

建立自我簽署的憑證

如果您不需要由 CA 簽署憑證，或是希望在 CA 簽署您的憑證期間測試新 SSL 的實作情況，則可以產生自我簽署的憑證。這個臨時憑證在用戶端瀏覽器中將產生錯誤，造成憑證簽署授權單位未知且不受信任。

若要透過 CLI 建立自我簽署憑證，請執行以下指令。

```
wadm> create-selfsigned-cert --user=admin --port=8989 --password-file=admin.pwd
--config=config1 --token=internal --org-unit=org1 --locality=XYZ --state=DEF
--validity=10 --org=sun --country=ABC --server-name=serverhost --nickname=cert1
```

請參閱 CLI 參照 `create-selfsigned-cert(1)`。

管理憑證

- [第 76 頁的「請求憑證」](#)
- [第 78 頁的「安裝憑證」](#)
- [第 79 頁的「更新憑證」](#)
- [第 79 頁的「刪除憑證」](#)

請求憑證

憑證是由數位資料組成，這些資料將指定個人、公司或其他實體的名稱，並證明憑證中包含的公開金鑰屬於該個人。啟用 SSL 的伺服器必須擁有憑證，但用戶端不一定要有憑證。

憑證由憑證授權單位或 CA 核發並以數位方式簽署。CA 可以是透過網際網路出售憑證的公司，也可以是負責為企業內部網路或企業間網路核發憑證的部門。您可以決定將充分信任的 CA 做為驗證其他使用者身份的機構。

您可以請求憑證並將憑證提交至憑證授權單位 (CA)。如果您的公司有自己的內部 CA，則可以向其請求憑證。如果計劃從商業 CA 購買憑證，請選擇一個 CA 並詢問其所需資訊的特定格式。您也可以為伺服器建立自我簽署的憑證。自我簽署的憑證不適用於連到網際網路的部署，但是對於開發和測試則非常有用，因為可讓您設定測試伺服器而無須 CA 介入。

如上文所述，憑證包括實體 (此案例中為 Web 伺服器) 的公開金鑰。公開金鑰是根據特定的演算法產生 (憑證中內建演算法類型程式碼)。下一節將提供 Web 伺服器所支援的金鑰演算法類型的背景說明。

▼ 請求憑證

1 按一下 [伺服器憑證] 標籤 > [請求] 按鈕。

2 選取配置

從配置清單中選取您要為其安裝憑證的配置。

3 選取記號

選取包含金鑰的記號 (加密裝置)。如果金鑰儲存在由 Sun Java System Web Server 7.0 維護的本機金鑰資料庫中，請選擇 [內部]。如果金鑰儲存在智慧卡或其他外部裝置或引擎中，請從下拉式清單方塊中選擇外部記號的名稱。輸入所選記號的密碼。

4 輸入詳細資訊

開始請求程序之前，請確保您已瞭解 CA 所需的資訊。無論是向商業 CA 還是向內部 CA 請求伺服器憑證，您都需要提供以下資訊：

- **伺服器名稱** 必須是 DNS 查詢中使用的完全合格的主機名稱 (例如 *www.sun.com*)。這是瀏覽器連線至您網站所使用的 URL 中的主機名稱。如果這兩個名稱不相符，則會通知用戶端憑證名稱與網站名稱不相符，由此懷疑憑證的可信賴性。

如果要從內部 CA 請求憑證，則也可以在此欄位中輸入萬用字元或常規表示式。大多數供應商將不會核准為一般名稱輸入萬用字元或常規表示式的憑證請求。

- **[組織]** 是指公司、教育機構、夥伴等的正式、合法名稱。大多數 CA 都要求您提出法律文件 (如營業執照副本) 來驗證此資訊。
- **[組織單位]** 為可選欄位，說明公司內部的組織。也可以用於標註不太正式的公司名稱 (不帶 *Inc.*、*Corp.* 等的名稱)。
- **[地區]** 為可選欄位，通常用於說明組織所在城市和公國。
- **[州或省]** 為選擇性欄位。
- **[國家/地區]** 是您所在國家/地區名稱的兩個字元縮寫 (ISO 格式)。美國的國家代碼為 US。

所有這些資訊組合成稱為辨別名稱 (DN) 的一系列「屬性-值」對，由此形成憑證的主體。

5 選擇憑證選項

需要您提供金鑰資訊。對於金鑰類型，您可以選擇 RSA 或 ECC。如果金鑰類型為 RSA，則金鑰大小可以為 1024、2048 或 4098。如果您的金鑰類型為 ECC，則還需要選取曲線。請記住，產生新的金鑰對需要花費一些時間。金鑰長度越長，精靈產生金鑰所需的時間便越長。



注意 - 請確定選取 CA (您稍後要向其提交請求以進行簽署) 可以支援的金鑰類型。

6 選取憑證類型

為憑證選取憑證簽署授權單位 (CSA) (是自我簽署或由 CA 簽署)。如果選取自我簽署的憑證，則還可以將憑證與 HTTP 偵聽程式建立關聯。您也可以稍後再執行此動作。

7 產生請求

如果是 CA 簽署的憑證，則產生的憑證請求將為 ASCII 格式。如果是自我簽署的憑證，則會直接進行安裝。如果是自我簽署的類型，請提供暱稱、有效期 (月數) 以及 HTTP 偵聽程式名稱的值，以便處理安全請求。

8 檢視結果

此頁面為您提供了所選選項的摘要。按一下 [完成] 可完成請求產生過程。

備註 – 使用 CLI

若要透過 CLI 請求憑證，請執行以下指令。

```
wadm> create-cert-request --user=admin --password-file=admin.pwd  
--host=serverhost --port=8989 --config=config1 --server-name=servername.org  
--org=sun --country=ABC --state=DEF --locality=XYZ --token=internal
```

請參閱 CLI 參照 create-cert-request(1)。

安裝憑證

從 CA 取得憑證之後，您可以使用 Administration Console 為配置安裝憑證。

▼ 安裝憑證

1 按一下 [伺服器憑證] 標籤 > [安裝] 按鈕。

2 選取配置

從配置清單中選取您要為其安裝憑證的配置。

3 選取記號

選取包含金鑰的記號 (加密裝置)。如果金鑰儲存在由 Sun Java System Web Server 7.0 維護的本機金鑰資料庫中，請選擇 [內部]。如果金鑰儲存在智慧卡或其他外部裝置或引擎中，請從下拉式清單方塊中選擇外部記號的名稱。輸入所選記號的密碼。

4 輸入憑證資料

在提供的文字區中，將憑證文字貼上。複製並貼上文字時，一定要包括「起始憑證」和「結束憑證」標頭—包括起始和結束連字符。您也可以按一下 [瀏覽] 按鈕，然後手動選取 DER 檔案。

5 提供憑證詳細資訊

提供用於憑證的暱稱。從可用清單中選取 HTTP 偵聽程式，以處理安全請求。

6 檢視結果

此頁面為您提供了所選選項的摘要。按一下 [完成] 可完成安裝程序。

備註 – 使用 CLI

若要透過 CLI 安裝憑證，請執行以下指令。

```
wadm> install-cert --user=admin --port=8989 --password-file=admin.pwd  
--config=config1 --token=internal --cert-type=server --nickname=cert1 cert.req
```

其中 cert.req 包含憑證資料。

請參閱 CLI 參照 install-cert(1)。

更新憑證

您可以透過執行以下步驟來更新現有憑證：

▼ 更新憑證

1 按一下 [伺服器憑證] 標籤 > [憑證名稱] > [更新] 按鈕。

2 提供記號資訊

如果需要，請輸入記號的密碼。否則按 [下一步] 繼續。

3 更新憑證詳細資訊

查看憑證詳細資訊，並提供有效期限 (以月為單位)。

4 更新金鑰資訊

對於金鑰類型，您可以選擇 RSA 或 ECC。如果金鑰類型為 RSA，則金鑰大小可以為 1024、2048 或 4098。如果您的金鑰類型為 ECC，則還需要選取曲線。請記住，產生新的金鑰對需要花費一些時間。

5 檢視摘要

此頁面為您提供了所選選項的摘要。按一下 [完成] 以完成更新程序。

刪除憑證

若要刪除憑證，請執行以下作業：

▼ 刪除憑證

- 1 按一下 [伺服器憑證] 標籤。
- 2 選取憑證
從憑證清單中選取憑證名稱。
- 3 刪除憑證
按一下 [刪除] 按鈕以刪除所選憑證。

備註 – 使用 CLI

若要透過 CLI 刪除憑證，請執行以下指令：

```
wadm> delete-cert --user=admin --port=8989 --password-file=admin.pwd  
--token=internal --config=config1 cert1
```

請參閱 CLI 參照 `delete-cert(1)`。

更新管理伺服器憑證

若要更新管理伺服器憑證，請執行指令 `renew-admin-certs` CLI 指令。使用此指令可以更新暱稱為 `Admin-CA-Cert`、`Admin-Server-Cert` 和 `Admin-Client-Cert` 的管理認證。此指令還可更新目前正在執行且使用更新的憑證可進行存取的節點。

執行此指令後，建議您重新啟動管理伺服器和節點，以便使新憑證生效。如果在更新憑證期間節點已離線（未執行或由於網路問題無法存取），則需要重新註冊此節點。若要更新管理伺服器憑證，請執行以下指令。

```
wadm> renew-admin-certs --user=admin --password-file=admin.pwd --host=serverhost  
--port=8989 --validity=120
```

請參閱 CLI 參照 `renew-admin-certs(1)`。

管理憑證撤銷清單 (CRL)

憑證撤銷清單 (CRL) 可公佈用戶端或伺服器使用者不應再信任的所有憑證和金鑰。如果憑證中的資料發生變更，例如使用者在憑證過期前換單位或離職，則將會撤銷憑證，並將資料顯示在 CRL 中。CRL 由 CA 產生並定期更新。

▼ 安裝 CRL

若要從 CA 取得 CRL，請執行以下步驟：

- 1 從 CA 取得 CRL 做為檔案。
- 2 移至 **Administration Console** 中的配置頁面。
- 3 按一下 [憑證] > [憑證授權單位] 標籤。
- 4 按一下 [安裝 CRL] 按鈕
- 5 輸入相關聯檔案的完整路徑名稱。
- 6 按一下 [確定]。

備註 - 如果資料庫中已經存在 CRL，則將會顯示 [替代憑證撤銷清單] 頁面。

- 7 您可能需要按一下 [部署] 以使變更生效。

備註 - 使用 CLI

若要透過 CLI 安裝 CRL，請執行以下指令。

```
wadm> install-crl --user=admin --password-file=admin.pwd  
--host=serverhost --port=8989 --config=config1 data/install-crl/ServerSign.crl
```

請參閱 CLI 參照 `install-crl(1)`。

▼ 刪除 CRL

- 1 移至 **Administration Console** 中的配置頁面。
- 2 按一下 [憑證] > [憑證授權單位] 標籤。
- 3 選取 CRL 項目並按一下 [刪除]。
- 4 您可能需要按一下 [部署] 以使變更生效。

備註 – 使用 CLI

若要透過 CLI 刪除 CRL，請執行以下指令。

```
wadm> delete-crl --user=admin --password-file=admin.pwd  
--host=serverhost --port=8989 --config=config1 issuer
```

請參閱 CLI 參照 delete-crl(1)。

設定內部記號的密碼

若要設定內部 PKCS11 記號的密碼，請執行以下作業：

▼ 設定記號密碼

- 1 移至 **Administration Console** 中的配置頁面。
- 2 按一下 [憑證] > [PKCS11 記號] 標籤。
- 3 按一下 PKCS11 記號名稱 (預設為內部)。
- 4 選取 [記號狀態] 核取方塊。
- 5 輸入密碼資訊。
- 6 如果您不希望在實例啟動時提示您輸入密碼，請選取 [在實例啟動時不提示您輸入新密碼] 核取方塊。按一下 [確定]。
- 7 密碼將儲存在配置中。若要移除密碼，請執行上述步驟並選取 [取消設定密碼] 選項。

備註 – 使用 CLI

若要透過 CLI 設定內部 PKCS11 記號的密碼，請執行以下指令。

```
wadm> set-token-pin --user=admin --password-file=admin.pwd --host=serverhost  
--port=8989 --config=config1 --token=internal
```

請參閱 CLI 參照 set-token-pin(1)。

為伺服器配置 SSL

使用指令 `create-cert-request` 可產生請求並將此請求傳送至 CA。稍後，當您收到來自 CA 的憑證時，將需要使用 `install-cert` 指令來安裝此憑證。如果要遷移的 Java 金鑰庫中有金鑰與憑證，請使用 `migrate-jks-keycert` 指令。對於開發/測試伺服器，最簡單的方法就是使用 `create-selfsigned-cert` 指令產生自我簽署的憑證。

```
wadm> create-selfsigned-cert --server-name=hostname --nickname=MyServerCert
--token=internal
```

請參閱線上手冊以瞭解更多選項與範例。

安裝憑證後，您需要在某個將啓用 SSL 的連接埠上安裝偵聽程式。

```
wadm> create-http-listener --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --listener-port=18003 --config=config1 --server-name=config1.com
--default-virtual-server-name=config1_vs_1 config1_ls_1
```

現在啓用偵聽程式的 SSL 功能，並將該偵聽程式與憑證的暱稱相關聯。

```
wadm> set-ssl-prop --http-listener=http-listener-ssl enabled=true
wadm> set-ssl-prop --http-listener=http-listener-ssl server-cert-nickname=MyServerCert
```

完成此設定之後，請部署配置並啓動實例。

```
wadm> deploy-config config_name
wadm> start-instance --config config_name hostname
```

啓用配置的 SSL 密碼

若要啓用配置的 SSL 加密，請執行以下指令。

```
wadm> enable-ciphers --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --http-listener=http-listener-1
--cipher-type=ssl2 SSL_RC4_128_WITH_MD5
```

請參閱 CLI 參照 `enable-ciphers(1)`。

啓用 HTTP 偵聽程式的安全性

備註 – 只有存在可用的已安裝憑證時，才能啓用 HTTP 偵聽程式的安全性。

一旦擁有了憑證，您便可以將憑證與 HTTP 偵聽程式相關聯，從而確保伺服器的安全。

加密是變換資訊的程序，因此除了預定的收件者之外，這些資訊對任何人都是毫無意義的。解密是變換加密資訊的程序，使這些資訊再次變得有意義。Sun Java System Web Server 可支援 SSL 和 TLS 通訊協定。

密碼是一種加密演算法 (一種數學函數)，用於加密或解密。SSL 和 TLS 協定包含多個密碼組。有些密碼會比其他密碼更強、更安全。一般說來，密碼使用的位元越多，解密資料就越困難。

在任何雙向加密程序中，雙方必須使用相同的密碼。因為有許多密碼可用，所以您需要為伺服器啓用那些最常用的密碼。

在安全的連線期間，用戶端與伺服器同意使用雙方均具有最強加密進行通訊。您可以從 SSL2、SSL3 和 TLS 通訊協定中選擇密碼。

備註 – SSL 版本 2.0 之後對 SSL 的安全性和效能進行了改進；除非用戶端無法使用 SSL 3，否則不要使用 SSL 2。使用 SSL 2 密碼無法保證用戶端憑證的安全性。

僅加密程序本身不足以確保伺服器機密資訊的安全。金鑰必需與加密密碼一起使用，才能產生實際的加密結果或解密先前加密的資訊。加密程序使用兩個金鑰來達到此結果：公開金鑰和私密金鑰。使用公開金鑰加密的資訊僅可透過相關聯的私密金鑰進行解密。公開金鑰做為憑證的一部分進行發佈；僅相關聯的私密金鑰需要安全保護。

Sun Java System Web Server 支援安全傳輸層 (SSL) 以及傳輸層安全性 (TLS) 通訊協定進行加密通訊。SSL 和 TLS 與應用程式無關，並且可以在其上放置更高層級的協定 (不需設定)。

SSL 和 TLS 通訊協定支援多種密碼，這些密碼用於伺服器和用戶端的相互認證、傳輸憑證和建立階段作業金鑰。根據所支援的通訊協定、公司對加密強度的策略、政府對加密軟體的出口限制等因素，用戶端和伺服器可能支援不同的密碼組或密碼集。除了其他功能之外，SSL 和 TLS 訊號交換協定會決定伺服器和用戶端如何協商使用哪種密碼組進行通訊。

按一下 **[配置] > [HTTP 偵聽程式] > [安全性]** 標籤，來編輯 HTTP 偵聽程式安全性設定。下表列出此頁面中可配置的特性。

表 6-1 HTTP 偵聽程式安全性特性

特性	說明
名稱	HTTP 偵聽程式的名稱。
安全性	爲選取的 HTTP 偵聽程式啓用/停用安全性。
憑證	從可用憑證中選取伺服器憑證。爲執行此操作，您應該安裝 RSA 或 ECC 憑證。
用戶端認證	指定用戶端認證爲必要的還是可選的。選取 [False] 選項可停用用戶端認證。
認證逾時	超過此逾時值後，用戶端認證訊號交換即告失敗。[0.001 – 3600]。預設值爲 60 秒。
最大認證資料量	緩衝區可容納的最大認證資料量。[0 – 2147.0483647.0]。預設值爲 104857.06。
SSL 版本 2/版本 3	啓用/停用 SSL 版本 2/SSL 版本 3。
TLS	啓用/停用 TLS。依預設，已啓用偵測版本回復。這會對伺服器進行配置以偵測截取式版本回復攻擊的嘗試。與一些未正確實作 TLS 規格的用戶端進行互通時，可能需要停用此功能。
SSL3/SSL2/TLS 密碼	<p>爲保護 Web 伺服器的安全，您應該啓用 SSL。您可以啓用 SSL 2.0、SSL 3.0 和 TLS 加密通訊協定並選取多種密碼組。可以在 Administration Server 的偵聽通訊端上啓用 SSL 和 TLS。在 Server Manager 的偵聽通訊端上啓用 SSL 和 TLS，會將所有虛擬伺服器的安全性喜好設定與偵聽通訊端相關聯。</p> <p>預設設定允許大多數常用密碼。除非您有不想使用特定密碼組的必要理由，否則您應允許全部密碼組。</p>

控制對伺服器的存取

您可以透過數種安全性服務與機制 (包括認證、授權和存取控制) 保護 Web 伺服器中的資源。本章說明一些用於控制存取 Sun Java System Web Server 7.0 的支援機制。

- 第 87 頁的「何為存取控制」
- 第 88 頁的「存取控制的運作方式」
- 第 89 頁的「為使用者/群組設定存取控制」
- 第 92 頁的「為主機 IP 設定存取控制」
- 第 93 頁的「配置 ACL 使用者快取記憶體」
- 第 94 頁的「配置存取控制」
- 第 98 頁的「使用 .htaccess 檔案」
- 第 99 頁的「防止伺服器受到拒絕服務攻擊」

何為存取控制

認證是確認識別的程序。授權表示將存取受限資源的權限授予某個身份識別，並由存取控制機制強制執行這些限制。可透過多種安全性模型 (Web 應用程式安全性、htaccess、認證範圍等) 和服務來強制執行認證與授權。

存取控制讓您決定：

- 哪些使用者可以存取 Administration Server
- 這些使用者可以存取哪些應用程式
- 哪些使用者可以存取您網站上的檔案或目錄

您可以控制對整個伺服器、部分伺服器或您網站上的檔案或目錄的存取。您可以建立稱為存取控制項目 (ACE) 的規則階層來允許或拒絕存取。您建立的 ACE 集合稱為存取控制清單 (ACL)。

依預設，伺服器具有一個包含多個 ACL 的 ACL 檔案。確定用於內送請求的虛擬伺服器後，Sun Java System Web Server 將檢查是否為該虛擬伺服器配置了任何 ACL。如果找到適用於目前請求的 ACL，伺服器將評估其 ACE 以確定是否應該允許或拒絕存取。

允許或拒絕存取基於以下事項：

- 提出請求的使用者 (使用者/群組)
- 請求的來源 (主機 IP)
- 請求發生的時間 (例如，一天中的某個時間)
- 使用的連線類型 (SSL)

存取控制的運作方式

當伺服器收到某個頁面的請求時，伺服器將使用 ACL 檔案中的規則來確定是否允許存取。這些規則可以參照傳送請求的電腦之主機名稱或 IP 位址。還可以參照儲存在 LDAP 目錄中的使用者和群組。

備註 – 如果有一個以上符合的 ACL，伺服器將使用最後一個相符的 ACL 敘述。由於 uri ACL 是最後一個符合的敘述，因此會略過預設 ACL。

上圖描述存取控制在 Web Server 7.0 中的運作方式。使用者代理程式 (用戶端) 會存取



Web Server。Web Server 執行 `obj.conf` 檔案中的 `PathCheck` 指令。Web 伺服器會將 HTTP 401 (未經授權) 傳回至用戶端。用戶端將提示使用者進行認證。如果用戶端為瀏

覽器，則將會顯示一個登入對話方塊。使用者輸入登入資訊。Web Server 執行內部 `check-acl` 功能。Web Server 驗證使用者憑證並處理請求。

為使用者/群組設定存取控制

您可以僅允許特定的使用者或群組存取您的 Web 伺服器。使用者/群組存取控制需要使用者輸入使用者名稱和密碼，然後才能存取伺服器。伺服器會將用戶端憑證中的資訊與目錄伺服器項目進行比較。

Administration Server 僅使用基本認證。如果您希望需要在 Administration Server 上進行用戶端認證，必須手動編輯 ACL 檔案，將方法變更為 SSL。

Web Server 藉由讀取使用者/群組資料庫中的項目，執行使用者/群組認證。目錄服務用於執行存取控制的資訊可以從以下任何一種來源獲得：

- 內部平面檔案類型資料庫
- 外部 LDAP 資料庫

當伺服器使用基於 LDAP 的外部目錄服務時，它支援對伺服器實例使用以下類型的使用者/群組認證方法：

- 預設值
- Basic (基本)
- SSL
- 摘要
- Other (其他)

當伺服器使用基於檔案的內部目錄服務時，它支援的可對伺服器實例使用的使用者/群組認證方法有：

- 預設值
- Basic (基本)
- 摘要

使用者/群組認證要求使用者先進行認證，然後才能存取伺服器或網站上的檔案和目錄。認證之後，使用者以用戶端憑證輸入使用者名稱及密碼，來驗證他們的身份。僅在 SSL 通訊時需要用戶端憑證。

預設認證

預設認證是首選方法。預設設定使用 `server.xml` 檔案中的預設方法，如果 `server.xml` 中不存在設定，將使用「基本」方法。如果核取 [預設]，ACL 規則便不會在 ACL 檔案中指定方法。如果選擇 [預設]，則編輯 `obj.conf` 檔案中的一行即可輕鬆變更所有 ACL 的方法。

基本認證

基本認證要求使用者輸入使用者名稱和密碼，才能存取 Web 伺服器或網站。此為預設設定。您必須在 LDAP 資料庫 (例如 Sun Java System Directory Server) 或檔案中建立並儲存使用者與群組清單。您使用的目錄伺服器不能與您的 Web 伺服器安裝在相同的伺服器根目錄下；您也可以使用安裝在遠端機器上的目錄伺服器。

當使用者嘗試在 Administration Server 或網站上存取需要使用者/群組認證的資源時，Web 瀏覽器將顯示對話方塊，要求該使用者輸入使用者名稱和密碼。伺服器將收到加密或未加密的資訊，這取決於是否為您的伺服器開啓了加密。

備註 – 使用不具有 SSL 加密的基本認證，以未加密文字的方式在網路上傳送使用者名稱和密碼。網路封包可能會被截取，並且使用者名稱和密碼可能會被盜用。基本認證結合 SSL 加密、主機 IP 認證或兩者時最有效。使用「摘要認證」可以避免此問題。

SSL 認證

伺服器可以使用安全憑證，以下列兩種方式來確認使用者的身份：

- 使用用戶端憑證中的資訊做為身份的證明
- 驗證 LDAP 目錄 (附加) 中發佈的用戶端憑證

當您設定讓伺服器使用憑證資訊對用戶端進行認證時，伺服器將：

- 首先檢查憑證是否來自可信任的 CA。如果不是，認證將失敗，異動也將結束。
- 如果憑證來自可信任的憑證授權單位 (CA)，請使用 `certmap.conf` 檔案將憑證對映到使用者的項目。
- 如果憑證正確對映，請檢查為該使用者指定的 ACL 規則。即使憑證對映，ACL 規則也可拒絕該使用者的存取。

要求以用戶端認證機制控制用戶端存取特定資源，與要求所有伺服器連線皆需進行用戶端認證是不同的。如果您將伺服器設定為需要對所有連線進行用戶端認證，則用戶端只需提供一份由可信任的 CA 核發的有效憑證即可。如果您將伺服器的存取控制設定為使用 SSL 方法來認證使用者與群組，則用戶端將需要：

- 提供由可信任的 CA 核發的有效憑證
- 該憑證必須對映到 LDAP 中的有效使用者
- 存取控制清單必須正確評估

當您需要以用戶端認證進行存取控制時，便需要為您的 Web 伺服器啓用 SSL 加密。

為了成功存取 SSL 認證資源，用戶端憑證必須來自 Web 伺服器信任的 CA。如果 Web 伺服器的 `certmap.conf` 檔案配置為將瀏覽器中的用戶端憑證與目錄伺服器中的用戶端憑證進行比較，則需要在該目錄伺服器中發佈用戶端憑證。不過，`certmap.conf` 檔案

也可以配置為僅將憑證中已選取的資訊與目錄伺服器項目進行比較。例如，您可以將 `certmap.conf` 檔案配置為僅將瀏覽器憑證中的使用者 ID 和電子郵件位址與目錄伺服器項目進行比較。

備註 – 由於檢查憑證時對照的是 LDAP 目錄，因此僅 SSL 認證方法需要對 `certmap.conf` 檔案進行修改。要求所有對伺服器的連線進行用戶端認證則不需要如此。如果您選擇使用用戶端憑證，則應該增加 `magnus.conf` 中 `AcceptTimeout` 指令的值。

摘要認證

可以將伺服器配置為使用基於 LDAP 或基於檔案的目錄服務執行摘要認證。

摘要認證允許使用者根據使用者名稱和密碼進行認證，而無須傳送明文格式的使用者名稱和密碼。瀏覽器使用使用者的密碼和 Web Server 提供的某些資訊，利用 MD5 演算法來建立摘要值。

當伺服器使用基於 LDAP 的目錄服務執行摘要認證時，也將使用摘要認證外掛程式在伺服器端計算該摘要值，並與用戶端提供的摘要值比較。如果摘要值相符，則使用者將通過認證。為了使此機制得以運作，您的目錄伺服器需要存取明文格式的使用者密碼。Sun Java System Directory Server 具有一個可逆的密碼外掛程式，它使用對稱的加密演算法以加密形式儲存資料，這些資料可在稍後被解密成原來的形式。只有 Directory Server 保留了資料的金鑰。

對於基於 LDAP 的摘要認證，您需要啓用伺服器中包含的可逆密碼外掛程式及 `digestauth` 專用外掛程式。若要配置 Web 伺服器以處理摘要認證，請設定 `dbswitch.conf` 中資料庫定義的 `digestauth` 特性。

如果未指定 ACL 方法，則需要認證時，伺服器將使用摘要認證或基本認證；不需要認證時，伺服器將使用基本認證。此為首選方法。

表 7-1 摘要認證的不同情況

ACL 方法	認證資料庫支援摘要認證	認證資料庫不支援摘要認證
「預設」	摘要和基本	基本
未指定		
「基本」	基本	基本
「摘要」	摘要	錯誤

當處理具有 `method = digest` 的 ACL 時，伺服器嘗試透過下列方式進行驗證：

- 檢查 Authorization 請求標頭。如果沒有找到該標頭，將會產生帶有摘要詰問的 401 回應，且程序也隨之停止。

- 檢查授權類型。如果認證類型為摘要，伺服器將：
 - 檢查 `nonce`。如果不是由此伺服器產生的有效的新 `nonce`，將會產生 401 回應，且程序會隨之停止。如果已過期，則會產生 401 回應 (其中 `stale=true`)，且程序會隨之停止。

您可以變更位於 `server_root/https-server_name/config/` 的 `magnus.conf` 檔案中 `DigestStaleTimeout` 參數的值，來配置 `nonce` 的有效時間。若要設定該值，請將以下行增加至 `magnus.conf`：

```
DigestStaleTimeout seconds
```

其中 *seconds* 表示 `nonce` 保持最新的秒數。指定的秒數過後，`nonce` 將過期並需要使用者進行新的認證。

 - 檢查範圍。如果不符合，將產生 401 回應，且程序會隨之停止。
 - 如果認證目錄是基於 LDAP 的，請檢查 LDAP 目錄中是否存在使用者；如果認證目錄是基於檔案的，請檢查檔案資料庫中是否存在使用者。如果沒有找到使用者，將產生 401 回應，且程序會隨之停止。
 - 從目錄伺服器或檔案資料庫取得請求/摘要值，並檢查是否符合用戶端的請求/摘要。如果不符合，將產生 401 回應，且程序會隨之停止。
 - 建構 `Authorization-Info` 標頭並將其插入伺服器標頭中。

為主機 IP 設定存取控制

您可以限制對 Administration Server 或您網站上的檔案和目錄的存取，方法為讓僅使用特定電腦的用戶端能夠存取它們。可以指定要允許或拒絕存取的電腦之主機名稱或 IP 位址。您可以使用萬用字元式樣指定多台電腦或整個網路。"使用主機 IP 認證存取檔案或目錄對使用者而言極為便利。" 使用者無須輸入使用者名稱或密碼，可立即存取檔案和目錄。

因為不止一人可以使用特定的電腦，所以主機 IP 認證結合使用者/群組認證時，會更有效。如果同時使用這兩種認證方法，則存取時將需要輸入使用者名稱和密碼。

主機 IP 認證並不需要在您的伺服器上配置 DNS。如果您選擇使用主機 IP 認證，則網路中必須有 DNS 處於執行中狀態，且您的伺服器必須配置為可使用該 DNS。您可以透過 Server Manager [效能調校] 頁面中的 [喜好設定] 標籤，在您的伺服器上啟用 DNS。

啟用 DNS 會降低伺服器的效能，因為伺服器將不得不執行 DNS 查詢。若要降低 DNS 查詢對伺服器效能的影響，請僅解析存取控制與 CGI 的 IP 位址，而非解析每個請求的 IP 位址。若要這樣做，請將 `iponly=1` 增加至 `obj.conf` 檔案中的 `AddLog` `fn="flex-log" name="access"`：

```
AddLog fn="flex-log" name="access" iponly=1
```

配置 ACL 使用者快取記憶體

依預設，伺服器將快取 ACL 使用者快取中的使用者和群組認證結果。您可以使用 `magnus.conf` 檔案中的 `ACLCacheLifetime` 指令，控制 ACL 使用者快取處於有效狀態的時間。每一次參照快取中的項目時，將對照 `ACLCacheLifetime` 計算並檢查該項目的作用時間。如果該項目的生命週期大於或等於 `ACLCacheLifetime`，則不會使用它。預設值為 120 秒。將該值設定為 0 (零) 將關閉快取。如果將此值設定為較大的數字，則每次變更 LDAP 項目時，您可能需要重新啟動伺服器。例如，如果將此值設定為 120 秒，則此伺服器可能會與 LDAP 目錄不同步達兩分鐘。僅當 LDAP 目錄不經常變更時才設定一個較大的值。

使用 `ACLUserCacheSize` 的 `magnus.conf` 參數，您可以配置可保留在快取中的最大項目數。該參數的預設值為 200。新項目將增加到清單的開頭，而且快取達到大小的上限時，會回收清單結尾處的項目以產生新項目。

您還可以使用 `magnus.conf` 中的參數 `ACLGroupCacheSize`，為每個使用者項目設定可快取的最大群組成員數。此參數的預設值為 4。遺憾的是，群組中非成員關係的使用者不會被快取，這將導致每個請求都要進行數個 LDAP 目錄存取。

如需有關 ACL 檔案指令的更多資訊，請參閱「*NSAPI Developer's Guide*」。

設定 ACL 快取特性

若要透過 CLI 設定 ACL 快取特性，請執行以下指令。

```
wadm> set-acl-cache-prop --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 property=value
```

請參閱 CLI 參照 `set-acl-cache-prop(1)`。

您可設定的有效特性為：

- `enabled`—指出伺服器是否快取檔案內容和中介資訊。預設值為 `true`。
- `max-age`—快取檔案內容和中介資訊的最長時間 (以秒為單位)。值的範圍是從 0.001 到 3600。
- `max-groups-per-user` —對於每個使用者而言，伺服器將快取其成員關係資訊的最大群組數。值的範圍是從 1 到 1024。
- `max-age`—快取認證資訊的最長時間 (以秒為單位)。值的範圍是從 0.001 到 3600。

配置存取控制

伺服器使用儲存在本機上的存取控制清單 (ACL) 來支援認證與授權，這些清單說明使用者對資源具有哪些存取權限。例如，ACL 中的某項目可授予名為 John 的使用者對特定資料夾 misc 的讀取權限。

本節說明了限制存取您網站上檔案或目錄的程序。您可以為所有伺服器設定全域存取控制規則，也可以為特定的伺服器進行個別設定。例如人力資源部門可以建立 ACL，允許所有經過認證的使用者檢視其自己的薪資資料，但是僅允許負責薪資的人力資源人員更新資料。

伺服器支援的核心 ACL 是三種類型的認證：基本認證、SSL 認證以及摘要認證。

若要編輯存取控制設定，請執行以下作業：

- 1. 按一下 [配置] 標籤，然後選取配置。
- 2. 按一下 [安全性] 子標籤 > [存取控制] 子標籤。
- 3. 按一下 [增加 ACL] 按鈕 增加新的 ACL，或按一下現有 ACL 來編輯設定。

增加存取控制清單 (ACL)

以下小節說明將新的 ACL 增加至配置的程序。

- 1. 按一下 [配置] 標籤，然後選取配置。
- 2. 按一下 [存取控制] 子標籤 > [存取控制清單] 子標籤。
- 3. [新增] 按鈕，增加新的 ACL。

配置以下參數：

表 7-2 ACL 參數

參數	說明
資源	已命名/URI/路徑。選取您需要設定存取限制和指定值的資源類型。URI 資源範例—「/sales」。路徑資源範例—「/usr/sun/server4/docs/cgi-bin/*」。
認證資料庫	認證資料庫可讓您選取伺服器將用於認證使用者的資料庫。 預設為 keyfile 。

表 7-2 ACL 參數 (續)

參數	說明
認證方法	<div><div>1. 基本—使用 HTTP 基本方法從用戶端取得認證資訊。僅當為伺服器啟用 SSL 時，才會透過網路對使用者名稱和密碼進行加密。</div><div>2. SSL—使用用戶端憑證來認證使用者。若要使用此方法，必須為伺服器啟用 SSL。啟用加密後，您可以結合使用「Basic」和「SSL」方法。</div><div>3. 摘要—使用認證機制，使得瀏覽器能夠根據使用者名稱和密碼進行認證，而無須傳送明文格式的使用者名稱和密碼。瀏覽器利用使用者的密碼和 Web Server 提供的某些資訊，透過 MD5 演算法來建立摘要值。請注意，為使用摘要，基本 auth-db 必須也支援摘要。僅當已安裝摘要認證外掛程式時，這才表示使用 digestfile 的檔案 auth-db 或 LDAP auth-db</div><div>4. 其他—使用透過存取控制 API 建立的自訂方法。</div></div>
提示進行認證	<div><p>[提示進行認證] 選項可讓您輸入顯示在認證對話方塊中的訊息文字。您可以使用此文字來說明使用者需要輸入的內容。基於瀏覽器不同，使用者將看到該提示中大概前 40 個字元。</p><p>通常，Web 瀏覽器會快取使用者名稱和密碼，並將其與提示文字相關聯。當使用者存取具有相同提示之伺服器的檔案和目錄時，無須再次輸入使用者名稱和密碼。如果您要讓使用者針對特定檔案和目錄再次進行認證，則只需變更該資源上 ACL 的提示。</p></div>
拒絕的存取回應	<div><p>指定存取資源遭拒絕時所採取的回應動作。</p><div><div>1. 使用預設訊息回應—選取此選項，以顯示來自伺服器的標準存取拒絕訊息。</div><div>2. 使用 URL 回應—選取此選項，以將請求轉寄至其他外部 URL 或錯誤頁面。</div></div></div>

備註 – 使用 CLI

若要透過 CLI 增加 ACL，請執行以下指令。

```
wadm> set-acl --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --vs=config1_vs_1 --config=config1
--aclfile=aclfile1
```

請參閱 CLI 參照 set-acl(1)。

增加存取控制項目 (ACE)

本節說明為所選配置增加新存取控制項目 (ACE) 的程序。

- 1. 按一下 [配置] 標籤，然後選取配置。
- 2. 按一下 [存取控制] 子標籤 > [存取控制清單] 子標籤。
- 3. 按一下 [新增] 按鈕。
- 4. 按一下存取控制項目下的 [新增] 按鈕。

配置以下 ACE 參數：

表 7-3 ACE 參數

參數	說明
存取	<ul style="list-style-type: none">■ 允許表示使用者或系統可以存取請求的資源■ 拒絕表示使用者或系統不能存取該資源 伺服器將檢查整個存取控制表示式 (ACE) 清單以確定存取權限。
使用者	<ul style="list-style-type: none">1. 任何人—無認證。授予每個人存取權。2. 認證資料庫中的所有使用者—對認證資料庫中指定的所有使用者授予存取權。3. 僅認證資料庫中的以下使用者—僅限於認證資料庫中所選使用者存取。 <p>您可以根據一般屬性 (如名字、姓氏和電子郵件位址) 來查詢認證資料庫。</p>
群組	<p>使用群組認證時，會提示使用者輸入使用者名稱和密碼，然後使用者才能存取在存取控制規則中指定的資源。</p> <p>使用此選項限制對特定群組的存取。</p>

表 7-3 ACE 參數 (續)

參數	說明
從主機	<p>您可以根據請求的來源電腦限定對 Administration Server 或您的網站的存取。</p> <p>您可以根據請求的來源電腦限定對 Administration Server 或您的網站的存取。</p> <ul style="list-style-type: none">■ [任何位置] 允許存取所有使用者和系統■ [僅來自] 僅允許對特定主機名稱或 IP 位址進行存取 <p>如果選取 [僅來自] 選項，請在 [主機名稱] 或 [IP 位址] 欄位中輸入萬用字元式樣或以逗號分隔的清單。依主機名稱進行限定比依 IP 位址進行限定更為靈活：使用者的 IP 位址發生變更時，您不需要更新此清單。但是，依 IP 位址進行限定更可靠：如果某個連線用戶端的 DNS 查詢失敗，將無法使用主機名稱限制。</p> <p>您只能在萬用字元式樣中使用 * 萬用字元表示法，以比對電腦的主機名稱或 IP 位址。例如，若要允許或拒絕特定網域內的所有電腦，請輸入與該網域內所有主機相符的萬用字元式樣，例如 *.sun.com。您可以為存取 Administration Server 的超級使用者設定不同的主機名稱和 IP 位址。</p> <p>對於主機名稱，* 必須替代名稱中的整個部分。亦即，*.sun.com 有效，但 *users.sun.com 無效。* 出現在主機名稱中時，必須是最左側的字元。</p> <p>例如，*.sun.com 有效，但 users.*.com 無效。對於 IP 位址，* 必須替代位址中的整個位元組。例如 198.95.251.* 有效，但 198.95.251.3* 無效。* 出現在 IP 位址中時，必須是最右側的字元。例如 198.* 有效，但 198.*.251.30 無效。</p>

表 7-3 ACE 參數 (續)

參數	說明
權限	<p>存取權限可限制對您的網站上檔案和目錄的存取。除了允許或拒絕所有存取權限外，您還可以指定一個允許或拒絕部分存取權限的規則。例如，您可以授予使用者對您檔案的唯讀存取權限，這樣他們可以檢視資訊，但不能變更檔案。</p> <ul style="list-style-type: none">■ [所有存取權限] 是預設設定，將允許或拒絕所有權限■ [僅限以下權限] 允許您選取要允許或拒絕的權限組合：<ul style="list-style-type: none">■ 讀取可讓使用者檢視檔案，包括 HTTP 方法 GET、HEAD、POST 和 INDEX■ 寫入可讓使用者變更或刪除檔案，包括 HTTP 方法 PUT、DELETE、MKDIR、RMDIR 與 MOVE。若要刪除檔案，使用者必須同時具有寫入和刪除權限■ 執行可讓使用者執行伺服器端應用程式，例如 CGI 程式、Java applet 與代理程式■ 刪除可讓還具有寫入權限的使用者刪除檔案或目錄。■ 清單可讓使用者存取不包含 index.html 檔案之目錄中的檔案清單。■ 資訊可讓使用者接收有關 URI 的資訊，例如 http_head。
繼續	<p>伺服器將檢查整個存取控制表示式 (ACE) 清單以確定存取權限。例如，第一個 ACE 通常會拒絕所有使用者。如果將第一個 ACE 設定為「繼續」，伺服器將檢查清單中的第二個 ACE，如果它們相符，則會使用下一個 ACE。</p> <p>如果未核取「繼續」，則將拒絕所有使用者存取資源。伺服器將繼續檢查清單，直至找到某個不相符的 ACE，或者找到某個相符但未設定為繼續的 ACE。最後一個相符的 ACE 將確定是允許還是拒絕存取。</p>

使用 .htaccess 檔案

伺服器支援 .htaccess 動態配置檔案。您可以透過使用者介面或手動變更配置檔案來啓用 .htaccess 檔案。

您可以將 .htaccess 檔案與伺服器的標準存取控制結合使用。不管 PathCheck 指令的順序如何，標準存取控制始終在任何 .htaccess 存取控制之前套用。當使用者/群組認證為「基本」時，不需要同時使用標準和 .htaccess 存取控制進行認證使用者。您可以透過標準伺服器存取控制使用 SSL 用戶端認證，還需要透過 .htaccess 檔案進行 HTTP 「基本」認證。

如果您啓用 .htaccess 檔案，伺服器會在為資源提供服務之前先檢查該 .htaccess 檔案。伺服器會在與資源相同的目錄、該目錄的父系目錄、直至文件根目錄 (包含根目錄) 中尋找 .htaccess 檔案。例如，如果主文件目錄設定為 /sun/server/docs 且用戶端請求

/sun/server/docs/reports/index.html，則伺服器將檢查 /sun/server/docs/reports/.htaccess 和 /sun/server/docs/.htaccess 處的 .htaccess 檔案。

請注意，伺服器的附加文件目錄和 CGI 目錄功能允許管理員定義替代文件根目錄。存在替代文件根會影響 .htaccess 檔案處理。例如，假設某伺服器主文件目錄設定為 /sun/server/docs 且在 /sun/server/docs/cgi-bin/program.cgi 處存在 CGI 程式。如果將 CGI 做為文件類型啟用，則用戶端對 CGI 程式發出請求時，伺服器將計算 /sun/server/docs/.htaccess 和 /sun/server/docs/cgi-bin/.htaccess 的內容。但是，如果您將 CGI 目錄配置在 /sun/server/docs/cgi-bin 處，則伺服器將檢查 /sun/server/docs/cgi-bin/.htaccess 而不是 /sun/server/docs/.htaccess。這是因為將 /sun/server/docs/cgi-bin 指定為 CGI 目錄會將其標記為替代文件根目錄。

防止伺服器受到拒絕服務攻擊

拒絕服務 (DoS) 攻擊是伺服器中某些惡意使用者發動的明確攻擊行為，目的是讓合法使用者無法使用某服務。發動這類攻擊的方式如下：

- 向伺服器持續發送請求，以取得特定 Web 資源。

如果請求頻率相當高，則 Sun Java System Web Server 可監視存取頻繁的 URI 而偵測到 DoS 攻擊，進而拒絕請求。

以下章節說明如何在虛擬伺服器層級上防止 DoS 攻擊。

限制對伺服器的請求

現在只要配置請求限制並監視每個虛擬伺服器上的最大連線數，便可以調整伺服器以防止拒絕服務攻擊。配置其中某些值可能會影響伺服器的效能。

若要配置伺服器的請求限制，請按一下 [配置] > [虛擬伺服器] > [伺服器設定] > [請求限制]。配置以下表格中列出的參數。

表 7-4 配置請求限制

參數	說明
請求限制	啟用/停用對此虛擬伺服器的請求限制。依預設，請求限制選項處於停用狀態。
最大連線	該虛擬伺服器所允許的同步運作之連線的最大數目。
最大 RPS	允許用戶端每秒送出的請求數上限。
RPS 計算間隔	計算每秒請求數 (RPS) 平均值的時間間隔。預設值為 30 秒。

表 7-4 配置請求限制 (續)

參數	說明
繼續條件	決定遭到阻斷的請求類型要滿足哪些條件，才能再次用於提供服務。 靜音 —遭拒絕的請求數必須在後續間隔中降至零，以便繼續提供服務。 臨界值 —遭拒絕的請求比率必須降至 RPS 臨界值以下，以便繼續提供服務。 預設值為臨界值。
錯誤碼	要用於遭到阻斷之請求的 HTTP 狀態代碼。預設代碼為 HTTP 503—服務不可用。
監視屬性	要監視的選擇性請求屬性。

備註 – 使用 CLI

若要透過 CLI 來限制對伺服器的請求，請執行以下指令。

```
wadm> enable-request-limits --user=admin --password-file=admin.pwd  
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1
```

請參閱 CLI 參照 enable-request-limits(1)。

▼ 限制最大連線數

您可以限制同步運作之連線的最大數目。如果接收到相符的請求，且正在處理的請求數目至少已達到指定數字，則請求將遭到拒絕。請注意，只有那段特定時間內會拒絕請求。一旦同步運作的請求降至此限制以下，則將會處理新的請求。

- 1 按一下 [配置] 標籤。
- 2 從清單中選取配置。
- 3 從 [虛擬伺服器] 標籤下選取虛擬伺服器。
- 4 按一下 [伺服器設定] > [請求限制]。
- 5 輸入 [最多連線] 區段的值。

管理使用者和群組

本章說明如何增加、刪除和編輯可存取 Sun Java System Web Server 的使用者和群組。

- [第 101 頁的「存取關於使用者和群組的資訊」](#)
- [第 101 頁的「關於目錄服務」](#)
- [第 102 頁的「瞭解辨別名稱 \(DN\)」](#)
- [第 103 頁的「使用 LDIF」](#)
- [第 103 頁的「使用認證資料庫」](#)
- [第 105 頁的「設定使用者和群組」](#)
- [第 107 頁的「靜態和動態群組」](#)

存取關於使用者和群組的資訊

使用 Administration Server 可以存取關於使用者帳號、群組清單、存取權限 (ACL)、組織單位以及其他使用者和群組特定資訊的應用程式資料。

使用者和群組資訊以文字格式儲存在平面檔案中，或儲存在目錄伺服器 (例如 Sun Java System Directory Server) 中，支援簡易目錄存取協定 (LDAP)。LDAP 是透過 TCP/IP 執行的開放式目錄存取協定，可延伸至全域大小和數百萬個項目。

關於目錄服務

目錄伺服器 (如 Sun Java System Directory Server) 允許您管理來自單一應用程式的所有使用者資訊。您也可以將目錄伺服器配置為允許您的使用者從多個易存取的網路位置擷取目錄資訊。

在 Web Server 7.0 中，您可以配置三種不同類型的目錄服務以對使用者和群組進行認證與授權。如果未配置任何其他目錄服務，則不管新建的目錄服務為何類型，均會將其設定為值 `default`。

當您建立目錄服務時，將會使用目錄服務詳細資訊更新 `server.xml` 檔案。

目錄服務類型

Web Server 7.0 支援的不同目錄服務類型包括：

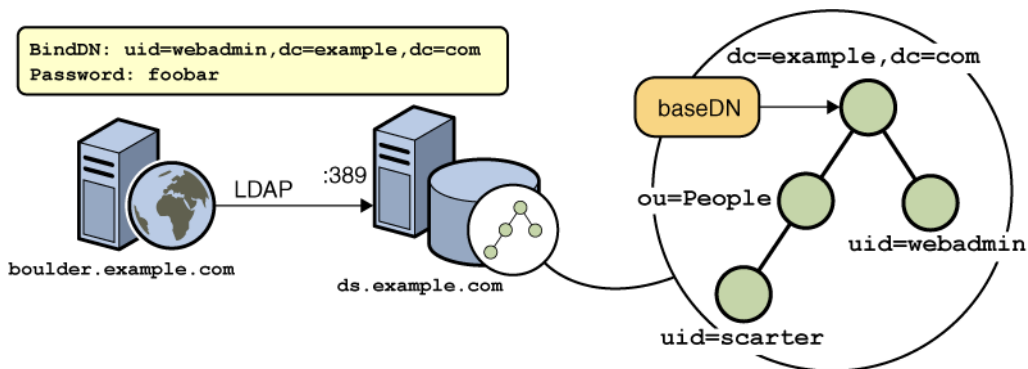
- **LDAP**—將使用者和群組資訊儲存在基於 LDAP 的目錄伺服器中。
- **金鑰檔案**—金鑰檔案是文字檔案，包含雜湊格式的使用者密碼以及該使用者所屬群組的清單。金鑰檔案中儲存的使用者和群組僅由 file 範圍用於進行授權和認證；他們與系統使用者和群組無任何關係。
金鑰檔案格式僅當意在使用 HTTP 基本驗證時才可使用。
- **摘要檔案**—根據加密的使用者名稱和密碼儲存使用者和群組資訊。
Digest file 格式意在支援使用 HTTP 摘要驗證。然而，它也支援基本認證，因此它可同時用於這兩種驗證方法。

備註—如果您要設定分散式管理，則預設目錄服務必須是基於 LDAP 的目錄服務。

瞭解辨別名稱 (DN)

使用者是您 LDAP 資料庫中的個人，例如您公司的員工。群組是共用某個一般屬性的兩個或多個使用者。組織單位是公司內的部門。

企業中每個使用者和群組將由辨別名稱 (DN) 屬性來表示。DN 屬性是一個包含關聯使用者、群組或物件之識別資訊的文字字串。每當您變更使用者或群組目錄項目時，就要使用 DN。例如，每當建立或修改目錄項目、設定存取控制以及為應用程式 (例如，郵件或發佈) 設定使用者帳號時，您都需要指定 DN 資訊。



上圖是 DN 表示的範例。以下範例表示 Sun Microsystems 公司某位員工的典型 DN：

```
uid=doe,e=doe@sun.com,cn=John Doe,o=Sun Microsystems Inc.,c=US
```

此範例中每個等號前縮寫的涵義如下：

- uid：使用者 ID
- e：電子郵件位址
- cn：使用者的一般名稱
- o：組織
- c：國家/地區

DN 可能包含各種「名稱-值」對。它們用於在支援 LDAP 的目錄中識別憑證主體和項目。

使用 LDIF

如果您目前沒有目錄，或者您想將新的子樹增加至現有目錄中，則可以使用 Directory Server 的 Administration Server LDIF 匯入功能。此功能接受包含 LDIF 的檔案，並且會嘗試從 LDIF 項目建立目錄或新子樹。您還可以使用 Directory Server 的 LDIF 匯出功能，將目前目錄匯出至 LDIF。此功能會建立一個代表您目錄的 LDIF 格式的檔案。可以使用 `ldapmodify` 指令和適當的 LDIF 更新描述來增加或編輯項目。

若要使用 LDIF 將項目增加至資料庫中，請先定義 LDIF 檔案中的項目，然後從 Directory Server 匯入 LDIF 檔案。

使用認證資料庫

認證資料庫也稱為 `auth-db`，代表已知使用者的資料庫，認證用戶端請求的機制就是以該資料庫為基礎。伺服器可同時配置擁有多個同一類型的 `auth-db` 項目。`auth-db` 使用者資料庫由 ACL 處理模組使用。

伺服器支援以下認證資料庫：

1. **LDAP** — 使用者資料儲存在 LDAP 目錄伺服器 (例如 Sun Java System Directory Server) 中。
2. **檔案** — 使用者資料儲存在磁碟檔案中。對於不使用 (或不需) 集中使用者管理的開發或小型部署來說，此 `auth-db` 尤其方便。檔案 `auth-db` 支援數種不同的檔案格式：
 - a. `keyfile` — `keyfile` 格式可儲存使用者 (以及每個使用者的可選群組成員) 清單。密碼儲存為單向 (不可回復的) 雜湊。這是預設格式。
 - b. `digestfile` — `digestfile` 與 `keyfile` 非常類似，並且也支援 HTTP 摘要認證方法。
 - c. `htaccess` — 這只是一種舊格式，絕不能將其用於新的安裝或增加新使用者。
3. **PAM** — PAM 是 Sun Java System Web Server 7.0 支援的新 `auth-db`。PAM `auth-db` 將認證委託給 Solaris PAM 堆疊，以便 Web 伺服器系統上的現有 Solaris 使用者也能通過 Web 伺服器認證。

備註 – 僅 Solaris 9 和 Solaris 10 (或更高版本) 支援 PAM auth-db，且 Web 伺服器實例必須以超級使用者身份執行。

建立認證資料庫

若要透過管理主控台來建立認證資料庫，請按一下 **[配置] > [配置名稱] > [存取控制] > [認證資料庫] > [新增]** 按鈕。檢查管理主控台內嵌說明以便取得欄位描述。欄位會根據選取的認證資料庫而有變化。例如，對於基於 PAM 的認證資料庫，只需要認證資料庫名稱即可。

此處已列舉建立認證資料庫的必要選項：

LDAP	<ul style="list-style-type: none">■ 認證資料庫名稱■ 主機名稱■ 連接埠■ 基底 DN
金鑰檔案	<ul style="list-style-type: none">■ 認證資料庫名稱■ 檔案路徑
摘要檔案	<ul style="list-style-type: none">■ 認證資料庫名稱■ 檔案路徑
PAM	<ul style="list-style-type: none">■ 認證資料庫名稱

若要透過 CLI 建立認證資料庫，請執行以下指令。

```
wadm> create-authdb --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1
--url=ldap://ldapsrvr.com:20002/dc=xxx,dc=sun,dc=com LDAP1
```

請參閱 CLI 參照 create-authdb(1)。

在上述範例中，已為認證資料庫指定 URL。此 URL 格式中已指定認證資料庫的類型。例如，ldap://ds.example.com/dc=example,dc=com 會將 LDAP 目錄伺服器配置為認證資料庫。

設定使用者和群組

Administration Server 可讓您編輯 LDAP 和檔案 auth-db 類型的使用者帳號、群組清單、存取權限、組織單位以及其他使用者和群組特定的資訊。

▼ 增加使用者

1 選取配置。

從配置清單中選取配置。按一下 [配置] 標籤以取得清單。

2 按一下 [存取控制] > [使用者] 標籤。

3 按一下 [新增] 按鈕。

4 增加使用者資訊。

輸入使用者 ID 與密碼。選擇性地輸入使用者所屬的群組。使用者 ID 必須唯一。如果是基於 LDAP 的認證資料庫，則 Administration Server 將從搜尋基底 (**基底 DN**) 向下搜尋整個目錄，以查看使用者 ID 是否處於使用中狀態，確保使用者 ID 的確是唯一。但是，請注意，如果您使用 Directory Server `ldapmodify` 指令行公用程式 (如果可用) 建立使用者，則其不會確保唯一的使用者 ID。

備註 – 使用 CLI

若要透過 CLI 建立使用者，請執行以下指令。

```
wadm> create-user --user=admin --password-file=admin.pwd --host=serverhost  
--port=8989 --config=config1 --authdb=KEYFILE1 --full-name=keyfile-config1-u1  
keyfile-config1-u1
```

請參閱 CLI 參照 `create-user(1)`。

▼ 增加群組

1 選取配置。

從配置清單中選取配置。按一下 [配置] 標籤以取得清單。

2 按一下 [存取控制] > [群組] 標籤。

3 按一下 [新增] 按鈕。

4 輸入群組名稱。

- 5 在 [將使用者增加至群組中] 區段中搜尋，並將現有使用者增加至群組中。

備註 – 在認證資料庫 (如 keyfile 或 digestfile) 中建立群組需要至少指定一個使用者。

備註 – 使用 CLI

若要透過 CLI 建立群組，請執行以下指令。

```
wadm> create-group --user=admin --password-file=admin.pwd --host=serverhost  
--port=8989 --config=config1 --authdb=LDAP1 group1
```

請參閱 CLI 參照 create-group(1)。

▼ 刪除使用者

- 1 選取配置。
從配置清單中選取配置。按一下 [配置] 標籤以取得清單。
- 2 按一下 [存取控制]>[使用者] 標籤。
- 3 選取您需要從中刪除使用者的認證資料庫。
- 4 在 [搜尋使用者] 文字方塊中輸入使用者 ID，然後按一下 [搜尋] 按鈕。
- 5 從 [使用者 ID] 欄中選取使用者，然後按一下 [刪除] 按鈕。



注意 – 如果刪除使用者後，群組中就沒有任何成員，則刪除 keyfile/digestfile 認證資料庫中的使用者時也會刪除關聯的群組。其原因是 keyfile/digestfile 認證資料庫不允許群組中無成員。

備註 – 使用 CLI

若要透過 CLI 刪除使用者，請執行以下指令。

```
wadm> delete-user --user=admin --password-file=admin.pwd --host=serverhost  
--port=8989 --config config1 --authdb KEYFILE1 user1
```

請參閱 CLI 參照 delete-user(1)。

▼ 刪除群組

- 1 選取配置。
從配置清單中選取配置。按一下 [配置] 標籤以取得清單。
- 2 按一下 [存取控制] > [群組] 標籤。
- 3 選取您需要從中刪除群組的認證資料庫。
- 4 在 [搜尋使用者] 文字方塊中輸入使用者 ID，然後按一下 [搜尋] 按鈕。
- 5 從 [使用者 ID] 欄中選取使用者，然後按一下 [刪除] 按鈕。

備註 – 刪除群組並不會刪除屬於該群組的使用者。您必須手動刪除使用者或重新指定群組。

備註 – 使用 CLI

若要透過 CLI 刪除群組，請執行以下指令。

```
wadm> delete-group --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config config1 --authdb LDAP1 group1
```

請參閱 CLI 參照 delete-group(1)。

靜態和動態群組

群組是描述 LDAP 資料庫中物件集的物件。Web Server 7.0 群組由共用某個一般屬性的使用者組成。例如，物件集可能是在您公司行銷部工作的一些員工。這些員工可能屬於稱為 Marketing 的群組。

對於 LDAP 服務，定義群組成員關係的方式有兩種：靜態和動態。靜態群組會明確列舉其成員物件。一個靜態群組就是一個 CN，包含 uniqueMembers 和/或 memberURLs 和/或 memberCertDescriptions。靜態群組的成員並不共用某個一般屬性，但 CN=<Groupname> 屬性除外。

動態群組允許您使用 LDAP URL 來定義僅與群組成員符合的規則集。動態群組的成員共用 memberURL 篩選器中定義的一般屬性或屬性集。例如，如果您需要一個包含所有 Sales 人員的群組，且他們已位於以下的 LDAP 資料庫

「ou=Sales,o=Airius.com」中，您需要使用以下 memberurl 定義一個動態群組：

```
ldap:///ou=Sales,o=sun??sub?(uid=*)
```

隨後，此群組將包含樹中「ou=Sales,o=sun」點下具有 uid 屬性的所有物件，即所有 Sales 成員。

對於靜態和動態群組，如果您使用 memberCertDescription，則成員可以共用憑證中的一般屬性。請注意，這些僅在 ACL 使用 SSL 方法時才適用。

建立新群組後，您即可在其中增加使用者或成員。

靜態群組

對於 LDAP 服務，Administration Server 可使您透過在任意數量使用者的 DN 中指定相同群組屬性來建立靜態群組。靜態群組僅在您於其中加入或刪除使用者時才變更。

建立靜態群組的指導原則

使用 Administration Server 表單來建立新的靜態群組時，請考量以下準則：

- 靜態群組可以包含其他靜態或動態群組。
- 您也可以選擇性地增加新群組的描述。
- 如果已為您的目錄定義任何組織單位，則可以使用 [增加新群組至] 清單指定要放置新群組的位置。預設的位置為目錄的根目錄位置或最頂層項目。

動態群組

動態群組具有 groupOfURLs 的 objectclass，並具有零個或多個 memberURL 屬性，每一個屬性都是描述物件集的 LDAP URL。

對於 LDAP 服務，如果您想要根據任意屬性將使用者自動分組，或者想要將 ACL 套用到包含相符 DN 的特定群組，Web Server 可使您建立動態群組。例如，您可以建立一個群組，該群組將自動包括含有 department=marketing 屬性的任何 DN。如果您為 department=marketing 套用搜尋篩選器，則搜尋將傳回一個群組，其中包括含有 department=marketing 屬性的所有 DN。然後，您可以基於此篩選器的搜尋結果來定義動態群組。隨後，您可以針對結果動態群組定義一個 ACL。

Web Server 實作動態群組的方式

Web Server 在 LDAP 伺服器模式中以 objectclass = groupOfURLs 來實作動態群組。groupOfURLs 類別可以具有多個 memberURL 屬性，每個屬性均包含一個 LDAP URL，用來列舉目錄中的物件集。群組成員將是這些物件集的併集。例如，以下群組僅包含一個成員 URL：

```
ldap:///o=mcom.com??sub?(department=marketing)
```

此範例說明包含「o=mcom.com」下部門為「marketing」的所有物件集。LDAP URL 可以包含搜尋基底 DN、範圍和篩選器，但主機名稱和連接埠除外。這就意味著您僅能參考同一 LDAP 伺服器上的物件。支援所有範圍。

將自動包括 DN，而無須您在群組中逐個增加。群組會動態變更，這是因為每次 ACL 驗證需要群組查詢時，Sun ONE Web Server 均會執行 LDAP 伺服器搜尋。ACL 檔案中使用的使用者和群組名稱與 LDAP 資料庫中物件的 cn 屬性相對應。

備註 – Web Server 使用 cn (commonName) 屬性做為 ACL 的群組名稱。

ACL 與 LDAP 資料庫的對映定義於 dbswitch.conf 配置檔案 (將 ACL 資料庫名稱與實際 LDAP 資料庫 URL 關聯) 中，以及 ACL 檔案 (定義各資料庫可用於哪個 ACL) 中。例如，如果您想讓「staff」群組中的成員擁有基底存取權限，ACL 代碼將查詢物件類別為 groupOf<anything> 且 CN 設定為 staff 的物件。物件定義群組成員的方式有兩種：明確列舉成員 DN (靜態群組的 groupOfUniqueNames 作業即如此)，或指定 LDAP URL (例如，groupOfURLs)。

群組可以為靜態和動態

群組物件可同時具有 objectclass = groupOfUniqueMembers 和 objectclass = groupOfURLs，因此「uniqueMember」和「memberURL」屬性都是有效的。群組的成員關係是其靜態成員和動態成員的併集。

動態群組對伺服器效能的影響

使用動態群組時會影響伺服器的效能。如果您正在測試群組成員關係，且 DN 不是靜態群組的成員，則 Web Server 將會檢查資料庫基底 DN 中的所有動態群組。Web Server 為完成此作業，會檢查基底 DN 和範圍 (對照使用者的 DN)，並藉由使用者 DN (做為基底 DN) 和篩選器 memberURL 來執行基底搜尋，從而檢查每一個 memberURL 是否符合。此程序會產生大量的個別搜尋。

建立動態群組的指導原則

使用 Administration Server 建立新動態群組時，請考量以下準則：

- 動態群組不能包含其他群組。
- 使用以下格式 (沒有主機和連接埠資訊，因為這些參數被忽略) 輸入群組的 LDAP URL：

```
ldap:///<basedn>?<attributes>?<scope>?(<filter>)
```

下表描述了需要的參數：

表 8-1 動態群組：必要參數

參數名稱	說明
<base_dn>	搜尋基底的辨別名稱 (DN) 或 LDAP 目錄中執行的所有搜尋的起始點。此參數經常設定為該目錄的後綴或根，例如「o=mcom.com」。
<attributes>	搜尋將傳回的屬性清單。若要指定一個以上的屬性，請使用逗號分隔這些屬性 (例如，「cn,mail,telephoneNumber」)；如果未指定任何屬性，則傳回所有屬性。請注意，動態群組成員關係檢查將忽略此參數。
<scope>	<p>搜尋的範圍，可以是下列值之一：</p> <ul style="list-style-type: none"> ■ base 僅擷取於 URL 中指定的有關辨別名稱 (<base_dn>) 相關資訊。 ■ one 擷取於 URL 中指定且低於辨別名稱 (<base_dn>) 一個層級的項目之資訊。此範圍不包括基準項目。 ■ sub 擷取於在 URL 中指定且低於辨別名稱 (<base_dn>) 的所有層級的項目相關資訊。此範圍包括基準項目。 <p>此參數是必要的。</p>
<(filter)>	<p>套用至搜尋指定範圍內項目的搜尋篩選器。如果您使用的是 Administration Server 表單，則必須指定此屬性。請注意，必須帶有括號。</p> <p>此參數是必需的。</p>

請注意，<attributes>、<scope> 和 <(filter)> 參數是依據它們在 URL 中的位置進行識別的。即使您不想指定任何屬性，也仍需要包括問號來分隔該欄位。

- 您也可以選擇性地增加新群組的描述。
- 如果已為您的目錄定義任何組織單位，則可以使用 [增加新群組至] 清單指定要放置新群組的位置。預設位置為您目錄的根目錄位置或最頂層項目。

管理伺服器內容

本章節說明如何配置與管理虛擬伺服器的內容。

- 第 111 頁的「配置文件目錄」
- 第 112 頁的「變更預設 MIME 類型」
- 第 113 頁的「自訂使用者公用資訊目錄 (UNIX/Linux)」
- 第 115 頁的「設定 URL 重新導向」
- 第 116 頁的「使用常規表示式進行 URL 重新導向」
- 第 117 頁的「CGI 簡介」
- 第 119 頁的「為您的伺服器配置 CGI 子系統」
- 第 121 頁的「下載可執行檔」
- 第 121 頁的「為 Windows 安裝 Shell CGI 程式」
- 第 122 頁的「自訂錯誤回應」
- 第 123 頁的「變更字元集」
- 第 124 頁的「設定文件註腳」
- 第 125 頁的「限制符號連結 (UNIX/Linux)」
- 第 126 頁的「設定伺服器剖析的 HTML」
- 第 127 頁的「設定快取控制指令」
- 第 128 頁的「為內容壓縮配置伺服器」
- 第 130 頁的「配置反向代理伺服器」
- 第 132 頁的「設定 P3P」

配置文件目錄

主文件目錄 (亦稱為文件根目錄) 是中央目錄，可儲存所有您希望供遠端用戶端使用的檔案。

除了主文件目錄之外，您還可以建立其他文件目錄。如此即可讓某使用者管理一組文件，而無需授予其存取主文件根目錄的權限。

▼ 建立文件目錄

- 1 選取配置。
從配置清單中選取配置。按一下 [配置] 標籤以取得可用的配置。
- 2 選取虛擬伺服器。
選取您需要為其增加新文件目錄的虛擬伺服器。按一下 [虛擬伺服器] 標籤，以取得所選配置的已配置虛擬伺服器清單。
- 3 按一下 [內容處理] > [文件目錄] 標籤。
- 4 按一下 [新增] 按鈕。配置以下參數：
 - URL 前綴 — 必須對映至目錄的 URL 前綴。
 - 目錄路徑 — 用於儲存文件的絕對伺服器路徑和有效目錄。

備註 - 使用 CLI

若要透過 CLI 建立文件目錄，請執行以下指令。

```
wadm> create-document-dir --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1
--uri-prefix=/config1_uri --directory=../docs1
```

請參閱 CLI 參照 create-document-dir(1)。

變更預設 MIME 類型

文件傳送至用戶端時，伺服器會包含識別此文件類型的區段，從而用戶端可以按正確方式展示文件。但是，有時候伺服器無法決定文件的正確類型，因為沒有為伺服器定義此文件的副檔名。在這些情況下，將會傳送預設值。

預設值通常為 text/plain，但是您應該將其設定為儲存在伺服器中的最常用檔案類型。某些常用 MIME 類型包括：

■ text/plain	■ text/html
■ text/richtext	■ image/tiff
■ image/jpeg	■ image/gif
■ application/x-tar	■ application/postscript

■ application/x-gzip	■ audio/basic
----------------------	---------------

▼ 變更預設 MIME 類型

- 1 選取配置。
從配置清單中選取配置。按一下 [配置] 標籤以取得可用的配置。
- 2 選取虛擬伺服器。
按一下 [虛擬伺服器] 標籤，以取得所選配置的已配置虛擬伺服器清單。
- 3 按一下 [內容處理] > [一般] 標籤。
- 4 在 [其他] 區段下變更 [預設 MIME 類型] 值

備註 – 使用 CLI

若要透過 CLI 建立 MIME 類型，請執行以下指令。

```
wadm> create-mime-type --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --extensions=sxc application/sxc
```

請參閱 CLI 參照 create-mime-type(1)。

您不需要為每個虛擬伺服器建立單獨的 MIME 類型檔案。而是可以根據自己的需要建立多個 MIME 類型，並使其與虛擬伺服器相關聯。依預設，有一個 MIME 類型檔案 (mime.types) 存在於伺服器中，且無法刪除。

自訂使用者公用資訊目錄 (UNIX/Linux)

有時候使用者想要自行維護自己的網頁。您可以配置公用資訊目錄，這些目錄可讓伺服器上的所有使用者建立首頁與其他文件，而不需要您的介入。

使用此系統，用戶端可以使用特定的 URL (伺服器將其識別為公用資訊目錄) 存取您的伺服器。例如，假設您選擇前綴 ~ 與目錄 public_html。如果請求使用 `http://www.sun.com/~jdoe/aboutjane.html`，則伺服器會識別為該 ~jdoe 是指使用者的公用資訊目錄。伺服器會在系統使用者資料庫中查詢 jdoe，並尋找 Jane 的主目錄。然後，伺服器會查看 `~/jdoe/public_html/aboutjane.html`。

若要配置伺服器以使用公用目錄，請執行以下步驟：

▼ 配置文件目錄

- 1 在虛擬伺服器頁面上，按一下 [內容處理] 標籤。
- 2 按一下 [文件目錄]。
- 3 在 [使用者文件目錄] 下，選擇使用者 URL 前綴。
通常使用的前綴為 ~，因為波浪號字元是存取使用者主目錄的標準 UNIX/Linux 前綴。
- 4 選擇使用者主目錄中的子目錄，伺服器會從該目錄中查找 HTML 檔案。
典型的目錄為 public_html。
- 5 指定密碼檔案。
伺服器需要瞭解查找檔案 (列示您系統中的使用者) 的位置。伺服器使用此檔案來確定有效的使用者名稱，並尋找其主目錄。如果您為達成此目的而使用系統密碼檔案，則伺服器將會使用標準程式庫呼叫來查詢使用者。或者，可以建立其他使用者檔案來查詢使用者。可以使用絕對路徑指定此使用者檔案。
使用者檔案中的每一行應該具有以下結構 (/etc/passwd 檔案中的非必要元素以 * 指示)：

```
username:*:*:groupid*:homedir:*
```
- 6 選擇是否在啟動時載入密碼資料庫。
- 7 按一下 [儲存]。
如需更多資訊，請參閱線上說明以瞭解 [使用者文件目錄] 頁面。
向使用者指定單獨目錄的另一種方法是建立對映至中央目錄 (所有使用者均可以修改) 的 URL。

限定內容發佈

在某些情況下，系統管理員可能想要限制哪些使用者帳號可透過使用者文件目錄發佈內容。若要限制使用者的發佈，請在 /etc/passwd file 的使用者主目錄路徑中增加尾隨斜線：

```
jdoue::1234:1234:John Doe:/home/jdoue:/bin/sh
```

成為：

```
jdoue::1234:1234:John Doe:/home/jdoue:/bin/sh
```

如此修改之後，Sun Java System Web Server 將不提供此使用者目錄中的頁面。請求此 URI 的瀏覽器會收到「404 未找到檔案」錯誤，並且 404 錯誤將記錄到 Web 伺服器存取記錄。任何錯誤都不會記錄到錯誤記錄。

如果稍後您又決定允許此使用者發佈內容，請從 `/etc/passwd` 項目中移除尾隨斜線，然後重新啟動 Web 伺服器。

啓動時載入整個密碼檔案

您也可選擇在啓動時載入整個密碼檔案。如果選擇此選項，伺服器會在啓動時將密碼檔案載入至記憶體，便於使用者更快速地查詢。不過，如果密碼檔案很大，此選項會佔用太多的記憶體。

設定 URL 重新導向

URL 重新導向可讓您將某個 HTTP URL 的文件請求重新導向至其他 HTTP URL。轉寄 URL 或重新導向是伺服器通知使用者 URL 已變更 (例如由於已將檔案移至另一個目錄或伺服器) 的方法。您還可以使用重新導向功能，將對某伺服器上文件的請求，順利傳送至其他伺服器上的文件。

例如，如果您將 `http://www.sun.com/info/movies` 轉寄至前綴 `film.sun.com`，則 URL `http://www.sun.com/info/movies` 會重新導向至 `http://film.sun.com/info/movies`。

有時可能要將對某個子目錄中所有文件的請求，重新導向至特定的 URL。例如，如果您必須移除目錄 (因為造成太多的網路流量或因為任何原因不再提供這些文件)，則可以將對其中任一文件的請求導向至解釋這些文件為什麼不再可用的頁面。例如，`/info/movies` 上的前綴可被重新導向至 `http://www.sun.com/explain.html`。

您可以在虛擬伺服器層級設定 URL 重新導向。

若要配置 URL 重新導向，請執行以下步驟：

1. 按一下 **[配置]** 標籤，並從配置清單中選取配置。
2. 按一下 **[虛擬伺服器]** 子標籤，並從虛擬伺服器清單中選取虛擬伺服器。
3. 按一下 **[內容處理]** 子標籤與 **[URL 重新導向]** 子標籤。
4. 按一下 **[新建]** 按鈕以增加新的 URL 重新導向規則。
5. 請為說明的欄位提供必要的值。按一下 **[確定]** 按鈕。如果需要，您可以按一下配置的 **[部署]** 按鈕。

下表說明增加新的 URL 重新導向規則時所需的參數。

表 9-1 URL 重新導向參數

參數	說明
自 URL	應從其重新導向請求的 URL。此 URL 的所有 HTTP 請求都將重新導向至 [目標 URL] 中指定的 URL。
目標 URL	應將請求重新導向至的 URL。來自「來源 URL」中所指定 URL 的所有 HTTP 請求，都將重新導向至此 URL。
URL 類型	固定的。 已啓用/已停用 。固定 URL 是指靜態 URL，如 HTML 頁面的連結。非固定 URL 是指具有請求參數的動態 URL 或僅具有前綴的 URL。

備註 - 使用 CLI

若要透過 CLI 增加新的 URL 重新導向規則，請執行以下指令。

```
wadm> create-url-redirect --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --no-ssl --config=config1 --vs=config1_vs_1 --uri-prefix=/redirect
--target-url=http://www.cnet.com
```

請參閱 CLI 參照 create-url-redirect(1)。

使用常規表示式進行 URL 重新導向

Sun Java System Web Server 7.0 已增強為可在配置檔案中支援常規表示式 (又稱為式樣) 以及請求時間參數內插法。此外，萬用字元式樣相符支援已延伸至 server.xml。URL 重新導向以 SAF 形式實作。重新導向 SAF 可讓您重新導向符合特定前綴的 URI。您可以使用 from 參數來指定前綴，並使用 url 或 url-prefix 參數來指定重新導向至的 URL。在 Sun Java System Web Server 7.0 中，from 參數是可選擇的。如果省略 from，則會重新導向所有 URI。

在 obj.conf 檔案中，新的 <If>、<Elseif> 和 <Else> 標記支援 SAF 參數。請參閱附錄 - obj.conf - 語法與用法。這些標記包括指令。使用這些標記可定義執行這些指令的條件。這些標記也可用於動態產生 SAF 參數。

Sun Java System Web Server 7.0 提供 URL 重新寫入功能，該功能為 Apache HTTP 伺服器之 mod_rewrite 模組的超集合。<If> 標記與 Apache 的 mod_rewrite 功能不同，可提供以下功能：

- 此標記可以處理 URI、路徑、標頭欄位以及回應內文。
- 此標記可在處理請求的任何階段進行作業。
- 此標記可搭配任何 SAF (包括協力廠商外掛程式) 使用。

請看以下指令：

```
NameTrans fn="redirect"
    from="/site1"
    url="http://site1.mycompany.com"
```

可以使用常規表示式重寫上述指令，如下所示：

```
<If $uri =~ '^/site1'>
    NameTrans fn="redirect"
    url="http://site1.mycompany.com"
</If>
```

在上述程式碼片段中，請注意此處使用常規表示式代替 `from` 參數。如果您需要將所有對 `/site1/*` 的請求重新導向至 `http://site1.mycompany.com/*/index.html`，請注意此方法：

```
<If $uri =~ '^/site1/(.*)'>
    NameTrans fn="redirect"
    url="http://site1.mycompany.com/$1/index.html"
</If>
```

此處的 `<If>` 標記將符合 `(.*)` 的值指定給變數 `$1`。來自原始請求的值將動態替代 `url` 參數中的 `$1`。這就表示上述 `obj.conf` 程式碼片段會使對 `/site1/download` 的請求被重新導向至 `http://site1.mycompany.com.com/download/index.html`。

`<If>` 與 `redirect` 結合後能獲得 `mod_rewrite` 的靈活性。但是，`<If>` 與 `mod_rewrite` 不同，除了重新導向與重新寫入 URL 之外還有其他用途。`<If>` 也可以與任何 ASF (包括協力廠商外掛程式) 搭配使用。

上述方法可配置 302 Moved Temporarily 重新導向。在 Sun Java System Web Server 7.0 中，您也可以增加 `status="301"` 參數，以指出您需要 301 Moved Permanently 重新導向。

```
NameTrans fn="redirect" from="/path" url="http://server.example.com" status="301"
```

CGI 簡介

您可以使用任意數量的程式設計語言定義共用閘道介面 (CGI) 程式。在 UNIX/Linux 機器上，您可能需要尋找以 Bourne shell 或 Perl 程序檔撰寫的 CGI 程式。

備註 – 在 UNIX/Linux 中所執行的額外 CGISTub 程序，可讓伺服器用來協助 CGI 執行。僅在初次存取 CGI 時才建立這些程序。它們的數目依 CGI 載入伺服器的方式不同而不同。初次存取 CGI 時才會建立這些程序。伺服器停止時，它們將消失。

如需更多資訊，請參閱線上「Sun Java System Web Server Performance Tuning and Sizing Guid」中有關 MinCGISTub、MaxCGISTub 與 CGISTubIdleTimeout 的討論。

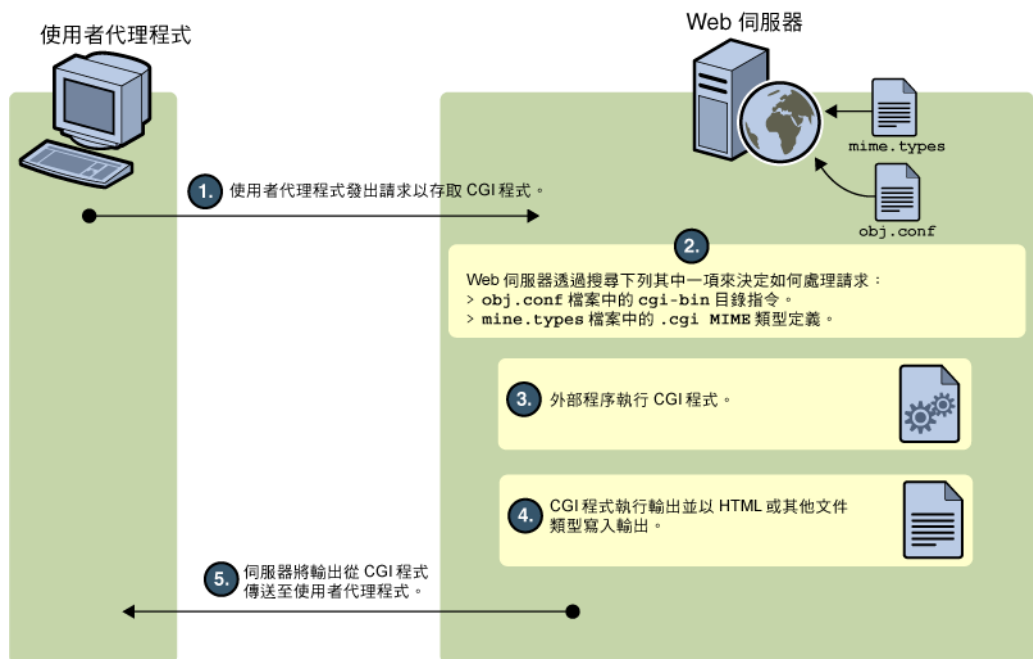
在 Windows 電腦上，您可能會找到以 C++ 或批次檔撰寫的 CGI 程式。對於 Windows 而言，以 Windows 程式設計語言 (如 Visual Basic) 撰寫的 CGI 程式使用不同的機制與伺服器一起作業。它們都稱為 Windows CGI 程式。

備註 - 若要執行指令行公用程式，您需要手動設定 Path 變數以包含 `server_root/bin/https/bin`。

無論為哪種程式設計語言，所有的 CGI 程式都會以同樣的方式接受並傳回資料。如需有關撰寫 CGI 程式的資訊，請參閱下列資訊來源：

- Sun Java System Web Server *Developer's Guide*
- 共用閘道介面位於：
<http://hoohoo.ncsa.uiuc.edu/cgi/overview.html>
- 線上文件網站上有關 CGI 的可用文章位於：
<http://docs.sun.com>

下圖說明 Web Server 7.0 中 CGI 請求的處理方式：



有兩種方式可在伺服器機器上儲存 CGI 程式：

- 指定一個僅包含 CGI 程式的目錄。無論副檔名為何，所有檔案都做為程式執行。

- 指定 CGI 程式全部為某特定檔案類型。即，全部使用 `.cgi`、`.exe` 或 `.bat` 副檔名。這些程式可位於文件根目錄中/下的任何目錄內。

如果需要，您可以同時啓用這兩個選項。

任一種執行方式都有優點。如果您想僅允許一組特定的使用者可增加 CGI 程式，請將這些 CGI 程式保留在指定目錄中，然後限制對這些目錄的存取。如果您想允許可增加 HTML 檔案的所有人都能增加 CGI 程式，請使用檔案類型選項。使用者可將他們的 CGI 檔案與 HTML 檔案保留在相同目錄中。

如果您選擇了目錄選項，您的伺服器會嘗試將該目錄中的所有檔案解譯為 CGI 程式。同樣，如果您選擇了檔案類型選項，您的伺服器會嘗試將具有 `.cgi`、`.exe` 或 `.bat` 副檔名的任何檔案做為 CGI 程式處理。如果某個檔案具有其中一個副檔名，卻不是 CGI 程式，則使用者嘗試存取該檔案時將會發生錯誤。

備註 – 依預設，CGI 程式的副檔名為 `.cgi`、`.exe` 與 `.bat`。但是修改 MIME 類型檔案即可變更代表 CGI 程式的副檔名。您可以透過選擇 [伺服器喜好設定] 標籤並按一下 [MIME 類型] 連結來執行此作業。

為您的伺服器配置 CGI 子系統

Sun Java System Web Server 可讓您使用 Administration Console GUI 增加 CGI 文件目錄。

若要增加新的 CGI 文件目錄，請執行以下作業：

1. 按一下 **[配置]** 標籤，並從配置清單中選取配置。
2. 按一下 **[虛擬伺服器]** 子標籤，並從虛擬伺服器清單中選取虛擬伺服器。
3. 按一下 **[內容處理]** 子標籤和 **[CGI]** 子標籤。
4. 按一下 **[新增]** 按鈕來增加新的 CGI 文件目錄。
5. 請為說明的欄位提供必要的值。按一下 **[確定]** 按鈕。如果需要，您可以按一下配置的 **[部署]** 按鈕。

下表說明了增加新的 CGI 文件目錄時所需的欄位。

表 9-2 CGI 參數

參數	說明
前綴	<p>鍵入此目錄要使用的 URL 前綴。即，您鍵入的文字將在 URL 中顯示為 CGI 程式的目錄。</p> <p>例如，如果您鍵入 <code>cgi-bin</code> 做為 URL 前綴，則所有指向這些 CGI 程式的 URL 都具有以下結構：</p> <p><code>http://yourserver.domain.com /cgi-bin/program-name</code></p>
CGI 目錄	<p>在 [CGI 目錄] 文字欄位中，以絕對路徑形式鍵入目錄位置。請注意，此目錄不必位於文件根目錄下。這就是您需要指定 URL 前綴的原因。</p> <p>備註 - 您指定的 URL 前綴可以與實際 CGI 目錄不同。</p>
使用者	指定要以其身份執行 CGI 程式的使用者名稱。
群組	指定要以其身份執行 CGI 程式的群組名稱。
Chroot	指定在執行開始前 <code>chroot</code> 至的目錄。
Nice	<p>指定 <code>nice</code> 值，即決定相對於伺服器的 CGI 程式之優先權的增量。</p> <p>通常，伺服器以 <code>nice</code> 值 0 開始執行，<code>nice</code> 增量值介於 0 (CGI 程式與伺服器以相同的優先權執行) 和 19 (CGI 程式以比伺服器低很多的優先權執行) 之間。雖然將 <code>nice</code> 增量值指定為 -1 可以將 CGI 程式的優先權提高到比伺服器更高的優先權執行，但是並不建議您這樣做。</p>

若要移除現有的 CGI 目錄，請選取該 CGI 目錄，然後按一下 [刪除] 按鈕。若要變更現有目錄的 URL 前綴或 CGI 目錄，請按一下目錄連結。

將 CGI 程式複製到指定的目錄中。請注意，這些目錄中的所有檔案都將做為 CGI 檔案處理，因此請勿將 HTML 檔案放在 CGI 目錄中。

若要將 CGI 指定為一種檔案類型，請執行以下作業：

1. 按一下 [配置] 標籤，並從配置清單中選取配置。
2. 按一下 [虛擬伺服器] 子標籤，並從虛擬伺服器清單中選取虛擬伺服器。
3. 按一下 [內容處理] 子標籤和 [CGI] 子標籤。
4. 按一下 [CGI 做為檔案類型] 單選方塊以啓用。

CGI 檔案必須具有副檔名 `.bat`、`.exe` 或 `.cgi`。任何非 CGI 檔案如果有這些副檔名，伺服器就會將其視為 CGI 檔案進行處理，從而引起錯誤。

備註 – 使用 CLI

您可以建立包含 CGI 程式 (由您的伺服器處理) 的 CGI 目錄。CGI 程式為特定的檔案類型，例如 .cgi、.exe 或 .bat。這些程式可位於文件根目錄中/下的任何目錄內。

若要透過 CLI 增加 CGI 目錄，請執行以下指令。

```
wadm> create-cgi-dir --user=admin --password-file=admin.pwd --host=serverhost  
--port=8989 --config=config1 --vs=config1_vs_1 --uri-prefix=/config1_urlprefix  
--directory=/cgi-dir
```

請參閱 CLI 參照 create-cgi-dir(1)。

下載可執行檔

如果您使用 .exe 做為 CGI 檔案類型，則無法將 .exe 檔案做為可執行檔下載。

此問題的一個解決方案是壓縮您要讓使用者下載的可執行檔，以便副檔名不是 .exe。此解決方案還可以縮短下載時間。

另一種可能的解決方案就是將做為副檔名的 .exe 從 magnus-internal/cgi 類型中移除，然後將其增加至 application/octet-stream 類型 (一般可下載檔案的 MIME 類型)。您可以透過在 Server Manager 中選擇 [伺服器喜好設定] 標籤，並按一下 [MIME 類型] 連結來執行此作業。但是，此方法的缺點就是在做出變更後，您無法將 .exe 檔案用做 CGI 程式。

另一個解決方案是編輯伺服器的 obj.conf 檔案，以設定下載目錄 (該目錄中的任何檔案都可以自動下載)。伺服器的其餘部分不受影響。如需更多資訊，請參閱：

<http://developer.netscape.com/docs/manuals/enterprise/admnunix/programs.htm>

為 Windows 安裝 Shell CGI 程式

適用於 Windows 的 Shell CGI 程式簡介

Shell CGI 是一種伺服器配置，允許您使用 Windows 中設定的檔案關聯執行 CGI 應用程式。

例如，如果伺服器收到對 hello.pl 的 shell CGI 檔案請求，則該伺服器會利用 Windows 檔案關聯，使用與 .pl 副檔名關聯的程式來執行檔案。如果 .pl 副檔名與程式 C:\bin\perl.exe 關聯，伺服器將嘗試按如下方式執行 hello.pl 檔案：

```
c:\bin\perl.exe hello.pl
```

配置 shell CGI 的最簡便方式是在伺服器文件根下建立一個僅包含 shell CGI 檔案的目錄。但是，您也可以透過從 Sun ONE Web Server 編輯 MIME 類型來配置伺服器，以便將特定副檔名與 shell CGI 相關聯。

備註 – 如需有關設定 Windows 副檔名的資訊，請參閱 Windows 文件。

自訂錯誤回應

您可以指定一個自訂錯誤回應，當用戶端遇到來自虛擬伺服器的錯誤時，會將詳細訊息傳送至用戶端。您可以指定一個要傳送的檔案或要執行的 CGI 程式。

例如當伺服器遇到特定目錄的錯誤時，您可以變更伺服器的運作方式。如果用戶端嘗試連線至受存取控制保護的伺服器部分，您可以傳回錯誤檔案，說明如何取得帳號。

在您啓用自訂錯誤回應前，您必須先建立錯誤回應中要傳送的 HTML 檔案或要執行的 CGI 程式。

若要增加自訂錯誤頁面，請執行以下步驟：

1. 按一下 **[配置]** 標籤，並從配置清單中選取配置。
2. 按一下 **[虛擬伺服器]** 子標籤，並從虛擬伺服器清單中選取虛擬伺服器。
3. 按一下 **[內容處理]** 子標籤和 **[錯誤頁面]** 子標籤。
4. 按一下 **[新增]** 按鈕以增加自訂錯誤頁面。

對於您要變更的各錯誤代碼，請指定包含錯誤回應的檔案或 CGI 的絕對路徑。

5. 按一下 **[確定]** 以返回至錯誤頁面清單。

備註 – 使用 CLI

若要透過 CLI 自訂錯誤頁面，請執行以下指令。

```
wadm> set-error-page --user=admin --password-file=admin.pwd --host=serverhost  
--port=8989 --config=config1 --vs=config1_vs_1 --code=500  
--error-page=/server-error-uri-new
```

請參閱 CLI 參照 `set-error-page(1)`。

變更字元集

文件的字元集在某種程度上而言是由撰寫所使用的語言來決定的。透過選取資源並輸入該資源的字元集，您可以置換文件、文件集或目錄的用戶端預設字元集設定。

大多數瀏覽器可以使用 HTTP 中的 MIME 類型 `charset` 參數來變更其字元集。如果伺服器在其回應中包括此參數，則瀏覽器會相應地變更其字元集。範例為：

- `Content-Type: text/html;charset=iso-8859-1`
- `Content-Type: text/html;charset=iso-2022-jp`

下列一些常用瀏覽器可識別的 `charset` 名稱是在 RFC 17.000 (以 `x-` 開頭的名稱除外) 中指定的：

■ <code>us-ascii</code>	■ <code>iso-8859-1</code>
■ <code>iso-2022-jp</code>	■ <code>x-sjis</code>
■ <code>x-euc-jp</code>	■ <code>x-mac-roman</code>

此外，以下為可識別的 `us-ascii` 別名：

■ <code>ansi_x3.4-1968</code>	■ <code>iso-ir-6</code>
■ <code>ansi_x3.4-1986</code>	■ <code>iso_646.irv:1991</code>
■ <code>ascii</code>	■ <code>iso646-us</code>
■ <code>us</code>	■ <code>ibm367.0</code>
■ <code>cp367.0</code>	

以下為可識別的 `iso_8859-1` 別名：

■ <code>latin1</code>	■ <code>iso_8859-1</code>
■ <code>iso_8859-1:1987.0</code>	■ <code>iso-ir-100</code>
■ <code>ibm819</code>	■ <code>cp819</code>

若要變更字元集，請執行以下步驟：

▼ 變更字元集

- 1 從虛擬伺服器頁面，按一下 [內容處理] 標籤。
- 2 按一下 [一般] 標籤。
- 3 在 [其他] 區段下設定預設字元集。
如果保留此欄位為空白，則字元集會設定為 NONE。
- 4 按一下 [Save]。

設定文件註腳

您可以指定文件頁尾，其可包含伺服器某部分中所有文件的最後修改時間。除 CGI 程序檔的輸出或剖析的 HTML (.shtml) 檔案之外，所有檔案均可使用此頁尾。如果您需要文件頁尾顯示在 CGI 程序檔輸出或已剖析的 HTML 檔案中，請將頁尾文字輸入至單獨的檔案中，並增加一行程式碼或另一個伺服器端包含，以將該檔案附加至此頁面的輸出中。

若要設定文件頁尾，請執行以下步驟：

▼ 設定文件頁尾

- 1 從虛擬伺服器頁面，按一下 [內容處理] 標籤。
- 2 按一下 [一般] 子標籤，然後移至 [文件頁尾] 區段。
- 3 請指定要包含此註腳的檔案之類型。
- 4 指定日期格式。
- 5 鍵入要在註腳中顯示的所有文字。
文件頁尾最多可包含的字元數為 7.065。如果您要包含最後一次修改文件的日期，請鍵入字串 :LASTMOD:。
- 6 按一下 [Save]。

備註 – 使用 CLI

若要透過 CLI 設定文件頁尾，請執行以下指令。

```
wadm> enable-document-footer --user=admin --password-file=admin.pwd  
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1  
--mime-type=text/html --date-format=%B --footer="config1 footer"
```

請參閱 CLI 參照 `enable-document-footer(1)`。

限制符號連結 (UNIX/Linux)

您可以在伺服器中限制使用檔案系統連結。檔案系統連結是儲存在其他目錄或檔案系統中的檔案之參考。參考使得存取遠端檔案就像在目前目錄中存取此檔案一樣。有兩種類型的檔案系統連結：

- 強制連結—強制連結是指向同一資料區塊集的兩個實際檔案名稱；原始檔案與連結是完全相同的。出於這種原因，硬式連結不能位於不同的檔案系統中。
- 符號 (軟式) 連結—符號連結包含兩個檔案：包含資料的原始檔案以及指向原始檔案的另一個檔案。符號式連結要比硬式連結更靈活。可以在不同的檔案系統中使用符號連結，並且可以連結至目錄。

如需有關強制連結與符號連結的更多資訊，請參閱 UNIX/Linux 系統文件。

檔案系統連結是在主文件目錄以外建立文件指標的一種簡易方法，任何人均可以建立這些連結。出於這種原因，您可能會擔心使用者建立敏感檔案 (例如，機密文件或系統密碼檔案) 的指標。

若要限定符號式連結，請執行以下步驟：

▼ 限制符號連結

- 1 從虛擬伺服器頁面，按一下 [內容處理] 標籤。
- 2 按一下 [一般] 子標籤。
- 3 移至 [其他] 區段下的 [符號連結] 區段。
- 4 選擇是否啓用符號連結和/或強制連結以及起始目錄。
- 5 按一下 [儲存]。

備註 – 使用 CLI

若要透過 CLI 限制符號連結，請執行以下指令。

```
wadm> set-symlinks-prop --user=admin --password-file=admin.pwd  
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1  
allow-soft-links=true allow-hard-links=false directory=/abc
```

請參閱 CLI 參照 `set-symlinks-prop(1)`。

設定伺服器剖析的 HTML

HTML 通常會準確地傳送至用戶端，就像其存在於磁碟中一樣，不需要任何伺服器介入。但是，伺服器在傳送文件之前，可以在 HTML 檔案中搜尋特殊的指令 (即，伺服器可以剖析 HTML)。如果您要伺服器剖析這些檔案，並將請求特定的資訊或檔案插入文件中，則必須首先啟用 HTML 剖析。

若要剖析 HTML，請執行以下步驟：

▼ 設定伺服器剖析的 HTML

- 1 在虛擬伺服器頁面上，按一下 [內容處理] 標籤。

- 2 按一下 [一般] 子標籤。

- 3 在 [剖析的 HTML/SSI 設定] 下，選擇是否啟動伺服器剖析的 HTML。

您可以針對 HTML 檔案而不針對 `exec` 標籤啟動，也可以同時針對 HTML 檔案與 `exec` 標籤啟動，這樣便可允許 HTML 檔案執行伺服器上的其他程式。

- 4 選擇要剖析的檔案。

您可以選擇是僅剖析具有 `.shtml` 副檔名的檔案，還是剖析所有 HTML 檔案 (這會降低效能)。如果使用 UNIX/Linux，則還可以選擇剖析執行許可權處於開啓狀態的 UNIX/Linux 檔案，但這樣做不可靠。

- 5 按一下 [儲存]。

如需有關使用伺服器剖析的 HTML 的更多資訊，請參閱「Sun Java System Web Server Developer's Guide」。

備註 – 使用 CLI

若要透過 CLI 設定伺服器剖析的 HTML，請執行以下指令。

```
wadm> enable-parsed-html --user=admin --password-file=admin.pwd  
--host=serverhost --port=8989 --config=config1 --vs=config1_vs1
```

請參閱 CLI 參照 `enable-parsed-html(1)`。

設定快取控制指令

快取控制指令是 Sun Java System Web Server 使用的一種控制方法，用來控制代理伺服器快取的資訊。使用快取控制指令，您可以置換代理伺服器的預設快取，以防止快取機密資訊（也許稍後會被擷取）。若要使這些指令可以運作，代理伺服器必須遵守 HTTP 1.1。

如需有關 HTTP 1.1 的更多資訊，請參閱超文字傳輸協定--HTTP/1.1 規格 (RFC 2068)，該協定位於：

<http://www.ietf.org/>

若要設定快取控制指令，請執行以下步驟：

▼ 設定快取控制指令

- 1 在虛擬伺服器頁面上，按一下 [內容處理] 標籤。
- 2 按一下 [一般] 子標籤並移至 [其他] 區段下的 [快取控制指令] 欄位。
- 3 填寫欄位。回應指令的有效值如下：
 - **公用**。此回應可透過任何快取記憶體快取。此為預設值。
 - **私有**。僅私有 (非共用) 快取記憶體才可快取回應。
 - **無快取記憶體**。在任何位置均不能快取此回應。
 - **無存放區**。此快取記憶體不能將任何位置的請求或回應儲存在永久性儲存體中。
 - **必須重新驗證**。此快取項目必須由原始的伺服器重新驗證。
 - **最長作用時間 (秒)**。用戶端不接受比此時間長的回應。
- 4 按一下 [Save]。

備註 – 使用 CLI

若要透過 CLI 設定快取控制指令，請執行以下指令。

```
wadm> set-cache-control-directives --user=admin --password-file=admin.pwd  
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1 public=true  
private=true must-revalidate=true
```

請參閱 CLI 參考 `set-cache-control-directives(1)`。

為內容壓縮配置伺服器

Sun Java System Web Server 7.0 支援 HTTP 內容壓縮。透過內容壓縮，您可以加快對用戶端的傳送速度，還可以提供更高的內容量，而不會增加硬體消耗。內容壓縮縮短了內容的下載時間，使需要撥號與多次連線的使用者明顯受益。

透過內容壓縮，Web 伺服器可傳送經過壓縮的資料，並指導瀏覽器快速解壓縮資料，這樣便可減少資料傳送量，並加快頁面顯示速度。

配置伺服器以提供預先壓縮內容

您可以配置 Sun Java System Web Server，以產生預先壓縮的檔案版本，並將其儲存在指定的目錄中。配置好之後，且只有在接收到 `Accept-encoding: gzip` 標頭時，只要是請求已配置用來提供預先壓縮內容的目錄中之檔案，這些請求就會全部重新導向為請求該目錄中的等效壓縮檔案 (如果存在該檔案)。例如，假設 Web 伺服器接收到對 `myfile.html` 的請求，而 `myfile.html` 和 `myfile.html.gz` 同時存在，則包含適當 `Accept-encoding` 標頭的這些請求會接收到經過壓縮的檔案。

若要將伺服器配置為提供預先壓縮的內容，請執行以下步驟：

▼ 變更預先壓縮的內容設定

- 1 在虛擬伺服器頁面上，按一下 [內容管理] 標籤。
- 2 按一下 [一般] 子標籤。
- 3 移至 [壓縮] > [預先壓縮的內容] 區段，並從以下選項中選取。
 - 預先壓縮的內容—啟用/停用。允許您指示伺服器為選取的資源提供預先壓縮的內容。
 - 作用時間檢查—指定是否檢查經過壓縮的版本比未壓縮的版本舊。
如果選取此選項，則經過壓縮的版本比未壓縮的版本舊時，不會選取經過壓縮的版本。

如果未選取此選項，則即使經過壓縮的版本比未壓縮的版本舊，一律會選取經過壓縮的版本。

- **插入 Vary 標頭**—指定是否使用 Vary: Accept-encoding 標頭。

如果選取此選項，則選取檔案的壓縮版本時，一律會插入 Vary: Accept-encoding 標頭。

如果未選取此選項，則不會插入 Vary: Accept-encoding 標頭。

- 4 按一下 [Save]。

將伺服器配置為依需求壓縮內容

您也可以將 Sun Java System Web Server 7.0 配置為快速壓縮傳輸資料。除非使用者要求，否則動態產生的 HTML 頁面不會存在。這對於電子商務的 Web 應用程式和以資料庫為核心的網站特別有用。

若要將伺服器配置為依需求壓縮內容，請執行以下步驟：

▼ 依需求壓縮內容

- 1 在虛擬伺服器頁面上，按一下 [內容處理] 標籤。
- 2 按一下 [一般] 子標籤。移至 [壓縮] 區段下的 [依需求壓縮內容] 區段。
- 3 請從以下選項中選取：
 - **依需求壓縮**—對所選資源啟用/停用依需求壓縮。
 - **插入 Vary 標頭**—指定是否插入 Vary: Accept-encoding 標頭。
如果已選取，則選取檔案的壓縮版本時，總是會插入 Vary: Accept-encoding 標頭。
如果未選取，則不會插入 Vary: Accept-encoding 標頭。
 - **分段大小**—以位元組為單位指定壓縮程式庫 (zlib) 要使用的記憶體分段大小，用於控制一次要壓縮多少資料。預設值為 8096。
 - **壓縮層級**—指定壓縮層級。選擇 1 至 9 之間的值。數值 1 表示以最快速度進行壓縮；數值 9 表示產生最佳壓縮效果。預設值為 6，同時兼顧壓縮的速度和效果。
- 4 按一下 [Save]。

備註 – 使用 CLI

若要透過 CLI 啟用依需求壓縮功能，請執行以下指令。

```
wadm> enable-on-demand-compression --user=admin  
--password-file=admin.pwd --host=serverhost --port=8989 --config=config1  
--vs=config1_vs_1 --insertvaryheader=true  
--fragment-size=100 --compression-level=5
```

請參閱 CLI 參考 `enable-on-demand-compression(1)`。

配置反向代理伺服器

反向代理伺服器這種代理伺服器表面上是用戶端的 Web 伺服器 (原始伺服器)，但實際上它是將接收到的請求轉送至一個或多個原始伺服器。因為反向代理伺服器以原始伺服器的姿態呈現，所以用戶端不需要進行配置即可使用反向代理伺服器。已知反向代理伺服器只要配置成將請求轉寄給多個配置相似的原始伺服器，就可以成為應用程式層級的軟體負載平衡器。

備註 – 在一般部署中，瀏覽器和原始伺服器之間會部署一個或多個反向代理伺服器。

▼ 增加代理伺服器 URI

- 1 按一下 [配置] 標籤，然後選取配置。
- 2 按一下 [虛擬伺服器] 標籤，然後選取虛擬伺服器。
- 3 按一下 [內容處理] > [反向代理伺服器] 標籤。
- 4 按一下 [新增代理伺服器 URI] 按鈕。

指定以下參數的值：

- **URI** — 反向代理伺服器 URI
- **伺服器 URL** — 以逗號分隔的遠端伺服器 URL。如果給定多個值，則伺服器將在指定的伺服器之間分配負載。

▼ 修改反向代理參數

- 1 按一下 [配置] 標籤，然後選取配置。
- 2 按一下 [虛擬伺服器] 標籤，然後選取虛擬伺服器。
- 3 按一下 [內容處理] > [反向代理伺服器] 標籤。
- 4 按一下 [URI]

您可以編輯以下參數：

- **URI** — 反向代理伺服器 URI
- **伺服器 URL** — 以逗號分隔的遠端伺服器 URL。如果給定多個值，則伺服器將在指定的伺服器之間分配負載。
- **居留式 Cookie** — 一種 cookie 的名稱，當其出現在回應中時，會導致隨後的請求居留於原始伺服器。
- **居留式 URI 參數** — URI 參數名稱，用於檢查路由資訊。當 URI 參數出現於請求 URI 中，並且它的值包含冒號「:」、後接路由 ID 時，該請求將「居留」於該路由 ID 所識別的原始伺服器。
- **路由標頭** — HTTP 請求標頭的名稱，用於將路由 ID 傳送給原始伺服器。
- **路由 Cookie** — 在回應中遇到「sticky-cookie」cookie 時，伺服器所產生 cookie 的名稱。「route-cookie」cookie 可儲存路由 ID，從而可讓伺服器引導後續請求返回至相同的原始伺服器。

備註 – 使用 CLI

1. 呼叫 `create-reverse-proxy` 指令。

```
wadm> create-reverse-proxy --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=test --vs=test --uri-prefix=/
--server=http://rick.india.sun.com:8080
```

請參閱 CLI 參考 `create-reverse-proxy(1)`。

2. 修改 `obj.conf` 檔案。

```
NameTrans fn="map" from="/" name="reverse-proxy-/" to="http/"
...
<Object name="reverse-proxy-/">
Route fn="set-origin-server" server="http://rick.india.sun.com:8080"
</Object>

<Object ppath="http:*">
Service fn="proxy-retrieve" method="*"
</Object>
```

若要重新導向至安全網站，請執行相同的步驟並為 `--server` 選項提供 `https` 位址。

設定 P3P

■ 第 132 頁的「配置虛擬伺服器的 P3P 設定」

隱私權喜好設定平台 (P3P) 使網站能夠以標準格式 (可自動擷取且易於由使用者代理程式解釋) 表達網站的隱私權作法。P3P 使用者代理程式可讓使用者知道網站的作法 (以機器可讀和人類可讀兩種格式)。如需更多資訊，請參閱 <http://www.w3.org/P3P/>。

▼ 配置虛擬伺服器的 P3P 設定

1 選取配置。

從配置清單中選取配置。按一下 [配置] 標籤以取得可用配置的清單。

2 選取虛擬伺服器。

從虛擬伺服器清單中選取虛擬伺服器。按一下 [虛擬伺服器] 標籤，以取得所選配置的可用虛擬伺服器。

3 按一下 [一般] 標籤。配置 [P3P] 區段中的以下設定。

- **已啓用**—爲選取的虛擬伺服器啓用 P3P。
- **策略 URL**—輸入相關 P3P 策略檔案的位置。
- **簡潔策略**—簡潔策略可向使用者代理程式 (瀏覽器或其他 P3P 應用程式) 提供提示，以讓使用者代理程式在套用策略上做出迅速、同步的決定。小型策略是由 P3P 規格指定的效能最佳化機制，使用者代理程式或伺服器可選擇是否採用。

備註 - 使用 CLI

若要爲虛擬伺服器啓用 P3P，請執行以下指令：

```
wadm> enable-p3p --user=admin --password-file=admin.pwd --host=serverhost  
--port=8989 --config=config1 --vs=config1_vs_1 --policy-url=http://xyz.com/policyurl
```

請參閱 CLI 參考enable-p3p(1)。

透過 WebDAV 進行 Web 發佈

- 第 136 頁的「關於 WebDAV」
- 第 136 頁的「常用 WebDAV 術語」
- 第 139 頁的「在實例層級啓用 WebDAV」
- 第 140 頁的「管理 WebDAV 集合」
- 第 141 頁的「配置 WebDAV 特性」
- 第 143 頁的「在伺服器層級停用 WebDAV」
- 第 143 頁的「管理 WebDAV 認證資料庫」
- 第 144 頁的「在啓用了 WebDAV 的伺服器上使用來源 URI 和 Translate:f 標頭」
- 第 144 頁的「鎖定和解除鎖定資源」
- 第 145 頁的「最小鎖定逾時」

Sun Java System Web Server 7.0 支援 WebDAV (即網路型分散式創作和版本控制，一種網路型協同作業的標準)。WebDAV 是 HTTP/1.1 協定的延伸，允許用戶端執行遠端 Web 內容創作作業。

完整的 WebDAV 作業事件包含啓用 WebDAV 的伺服器 (如可處理 WebDAV 資源請求的 Sun Java System Web Server 7.0) 以及啓用 WebDAV 的用戶端 (如支援啓用 WebDAV 的 Web 發佈請求之 Adobe(R) GoLive(R) 或 Macromedia(R) DreamWeaver(R))。

在伺服器端，您需要啓用並配置 Sun Java System Web Server 7.0 才能為 WebDAV 請求提供服務。

您可能會因為以下原因而對 WebDAV 進行配置：例如調校伺服器效能、消除安全性風險或提供無衝突遠端創作的環境。

為了滿足您的配置要求，可以變更伺服器對 WebDAV 資源的最短鎖定時間、集合上 PROPFIND 請求的深度以及請求內文所允許的最大 XML 內容大小等等。

您可以在虛擬伺服器層級，為虛擬伺服器下的所有集合配置預設 WebDAV 屬性。此處配置的值與 `server.xml` 檔案中的 DAV 元素相對應。

也可以在集合層級配置 WebDAV 屬性，這將置換所有為集合配置的虛擬伺服器層級屬性。在集合層級配置的屬性值與 `server.xml` 檔案中的 DAVCOLLECTION 元素相對應。

關於 WebDAV

WebDAV 是 HTTP/1.1 協定的延伸，所增加之新的 HTTP 方法和標頭，可提供任意類型的 Web 資源創作，除了 HTML 和 XML 之外，還支援文字、圖形、試算表以及所有其他格式。使用 WebDAV 可以完成的某些工作包括：

- **特性 (中介資料) 處理。** 您可以使用 WebDAV 的 PROPFIND 和 PROPPATCH 方法，建立、移除和查詢有關網頁的資訊，例如作者和建立日期。
- **集合與資源管理。** 您可以使用 WebDAV 的 GET、PUT、DELETE 和 MKCOL 方法，建立文件集並擷取階層式成員關係清單 (類似於檔案系統中的目錄清單)。
- **鎖定。** 您可以使用 WebDAV 禁止多人同時對一個文件進行操作。可以透過 WebDAV 的 LOCK 和 UNLOCK 方法，使用專用鎖定或共用鎖定，防止發生「遺失更新」(變更被覆寫) 問題。
- **名稱空間作業。** 您可以使用 WebDAV 的 COPY 和 MOVE 方法，讓伺服器複製和移動 Web 資源。

Sun Java System Web Server 7.0 中的 WebDAV 支援提供以下特性：

- 與 RFC 2518 的相容性以及與 RFC 2518 用戶端的相互可操作性。
- 用於發佈的安全性和存取控制。
- 對基於檔案系統的 WebDAV 集合和資源的有效發佈作業。

常用 WebDAV 術語

本節概述了使用 WebDAV 時經常遇到的術語。

URI。URI (統一資源識別碼) 是一種檔案識別碼，透過使用縮寫的 URL 提供一層額外的安全性。URL 的第一部分由隱藏了使用者的檔案完整實體路徑名稱的 URL 對映取代。

來源 URI。術語「來源 URI」是指可在該處存取資源來源的 URI。若要瞭解來源 URI 這一概念，請參閱以下範例：

一個名為 `foo.jsp` 的 JSP 頁面位於 `URI/docs/date.jsp` 處。該頁面包含 HTML 標記和 Java 程式碼，當執行該程式碼時，將在用戶端的瀏覽器中列印當天的日期。當伺服器收到來自用戶端對 `foo.jsp` 的 GET 請求後，會先執行該 Java 程式碼，然後才提供該頁面。用戶端不會收到駐留在伺服器上的 `foo.jsp` 頁面，而是收到一個顯示目前日期之動態產生的頁面。

如果建立了一個來源 URI (例如 `/publish/docs`) 並將其對映至包含 `foo.jsp` 的 `/docs` 目錄，則對 `/publish/docs/foo.jsp` 的請求將是對 `/docs/foo.jsp` JSP 頁面來源碼的請求。在這種情況下，伺服器會直接提供頁面而不執行 Java 程式碼。用戶端會收到未經處理的頁面，與儲存在磁碟中的頁面一樣。

這樣，對來源 URI 的請求也就成了對資源來源的請求。

集合。WebDAV 集合是為 WebDAV 作業啓用的資源或資源集。集合包含一組稱為成員 URI 的 URI，它們識別啓用了 WebDAV 的成員資源。

成員 URI。集合內屬於 URI 集中成員的 URI。

內部成員 URI。與集合 URI 直接相對的成員 URI。例如，如果 URL 為 `http://info.sun.com/resources/info` 的資源啓用了 WebDAV，並且 URL 為 `http://info.sun.com/resources/` 的資源也啓用了 WebDAV，則 URL 為 `http://info.sun.com/resources/` 的資源就是一個集合，它包含 `http://info.sun.com/resources/info`，並將其作為一個內部成員。

特性。包含有關資源的描述性資訊的「名稱/值」對。使用特性可以有效地查找和管理資源。例如，或許可以使用特性「`creationdate`」依資源的建立日期對全部資源進行索引，或者使用特性「`author`」依作者姓名進行索引。

動態特性。由伺服器執行的特性。例如，動態特性 `getcontentlength` 有一個值，即由 GET 請求傳回的實體長度，它是由伺服器自動計算的。動態特性具有以下特徵：

- 特性值是唯讀的，由伺服器維護
- 特性值由用戶端維護，但是伺服器對提交的值執行語法檢查。

靜態特性。不是由伺服器執行的特性。伺服器僅記錄靜態特性的值；用戶端負責維護其一致性。

Sun Java System Web Server 支援以下動態特性：

- `creationdate`
- `displayname`
- `getcontentlanguage`
- `getcontentlength`
- `getcontenttype`
- `gettag`
- `getlastmodified`
- `lockdiscovery`
- `resourcetype`
- `supportedlock`
- `executable`

備註 – Sun Java System Web Server 支援動態特性 `executable`，此動態特性允許用戶端變更與資源相關聯的檔案許可權。

以下為對 `executable` 動態特性 PROPPATCH 請求的範例：

```
PROPPATCH /test/index.html HTTP/1.1

Host: sun

Content-type: text/xml

Content-length: XXXX

<?xml version="1.0"?>

<A:propertyupdate xmlns:A="DAV:" xmlns:B="http://apache.org/dav/props/">

  <A:set>

    <A:prop>

      <B:executable>T</B:executable>

    </A:prop>

  </A:set>

</A:propertyupdate>
```

鎖定。鎖定資源功能提供了這樣一種機制，即可以保證一個使用者編輯資源時，其他使用者不能進行修改。鎖定可以防止發生覆寫衝突並解決了「遺失更新」問題。

Sun Java System Web Server 支援兩種鎖定類型：共用鎖定和專用鎖定。

新增 HTTP 標頭。WebDAV 透過延伸 HTTP/1.1 協定進行工作。它定義了新的 HTTP 標頭，用戶端可以通過該標頭傳送 WebDAV 資源請求。這些標頭為：

- Destination:
- Lock-Token:
- Timeout:
- DAV:
- If:
- Depth:
- Overwrite:

新增 HTTP 方法。WebDAV 引入了數個新 HTTP 方法，用於指示啓用了 WebDAV 的伺服器如何處理請求。這些方法是對現有方法 (例如 GET、PUT 和 DELETE) 的補充，用來執行 WebDAV 異動。下面簡要描述了這些新 HTTP 方法：

- COPY。用於複製資源。當 Destination: 標頭指定目標時，使用 Depth: 標頭複製集合。必要時 COPY 方法也會使用 Overwrite: 標頭。
 - MOVE。用於移動資源。當 Destination: 標頭指定目標時，使用 Depth: 標頭移動集合。在適當情況下，MOVE 方法也使用 Overwrite: 標頭。
 - MKCOL。用於建立新集合。此方法用來避免使 PUT 方法超載。
 - PROPPATCH。用於設定、變更或刪除單一資源的特性。
 - PROPFIND。用於擷取一個或多個資源中的一個或多個特性。當用戶端向伺服器提交對某個集合的 PROPFIND 請求時，該請求可能會包含一個值為 0、1 或 infinity 的 Depth: 標頭。
 - 0。指定將會擷取指定 URI 位置的集合特性。
 - 1。指定將會擷取集合特性以及直接位於指定 URI 之下的資源特性。
 - infinity。指定將會擷取集合及其包含的全部成員 URI 的特性。請注意，由於深度為無窮大的請求需要遍歷整個集合，因而會顯著增加伺服器的負擔。
- LOCK。為資源增加鎖定。使用 Lock-Token: 標頭。
- UNLOCK。移除資源的鎖定。使用 Lock-Token: 標頭。

在實例層級啓用 WebDAV

您可以使用 Administration Server 來為整個伺服器啓用 WebDAV。執行此作業時，以下指令將增加至用於載入 WebDAV Plugin 的 magnus.conf file 檔案中：

```
Init fn="load-modules" shlib="/slws6.1/lib/libdavplugin.so" funcs="init-dav,ntrans-dav,pcheck-dav,service-dav"
shlib_flags="(global|now)"
Init fn="init-dav" LateInit=yes
```

init-dav 初始化功能會初始化並註冊 WebDAV 子系統。

若要啓用 WebDAV，請在 CLI 中執行以下指令。

```
wadm> enable-webdav --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=test
```

請參閱 CLI 參考enable-webdav(1)。

管理 WebDAV 集合

啓用 WebDAV 集合

若要啓用 WebDAV 集合，請執行以下指令：

```
wadm> enable-dav-collection --user=admin --password-file=admin.pwd  
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1 --uri=/dav_config1
```

請參閱 CLI 參考enable-dav-collection(1)。

停用 WebDAV 集合

若要停用 WebDAV 集合，請執行以下指令：

```
wadm> disable-dav-collection --user=admin --password-file=admin.pwd  
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1 --uri=/dav_config1
```

請參閱 CLI 參考disable-dav-collection(1)。

增加 WebDAV 集合

若要增加 WebDAV 集合，請執行以下指令：

```
wadm> create-dav-collection --user=admin --password-file=admin.pwd  
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1 --uri=/dav_config1  
--source-uri=/dav_config1
```

請參閱 CLI 參考create-dav-collection(1)。

列出 WebDAV 集合

若要列出所有 WebDAV 集合，請執行以下指令：

```
wadm> list-dav-collections --user=admin --password-file=admin.pwd --host=serverhost  
--port=8989 --config=config1 --vs=config1_vs_1
```

請參閱 CLI 參考list-dav-collections(1)。

移除 WebDAV 集合

若要移除 WebDAV 集合，請執行以下指令：

```
wadm> delete-dav-collection --user=admin --password-file=admin.pwd --host=serverhost  
--port=8989 --config=config1 --vs=config1_vs_1 --uri=/dav_config1
```

請參閱 CLI 參考delete-dav-collection(1)。

配置 WebDAV 特性

設定 WebDAV 特性

若要設定伺服器層級的 WebDAV 特性，請執行以下指令：

```
wadm> set-webdav-prop --user=admin --password-file=admin.pwd --host=serverhost  
--port=8989 --config=config1 acl-max-entries=120
```

請參閱 CLI 參考set-webdav-prop(1)。

檢視 WebDAV 特性

若要檢視伺服器層級的 WebDAV 特性，請執行以下指令：

```
wadm> get-webdav-prop --user=admin --password-file=admin.pwd --host=serverhost  
--port=8989 --config=config1
```

請參閱 CLI 參考get-webdav-prop(1)。

設定 WebDAV 集合特性

若要設定 WebDAV 集合特性，請執行以下指令：

```
wadm> set-dav-collection-prop --user=admin --password-file=admin.pwd --host=serverhost  
--port=8989 --config=config1 --vs=config1_vs_1 --uri=/dav_config1 min-lock-timeout=1
```

請參閱 CLI 參考set-dav-collection-prop(1)。

檢視 WebDAV 集合特性

若要檢視 WebDAV 集合特性，請執行以下指令：

```
wadm> get-dav-collection-prop --user=admin --password-file=admin.pwd --host=serverhost  
--port=8989 --config=config1 --vs=config1_vs_1 --uri=/dav_config1
```

請參閱 CLI 參考 `get-dav-collection-prop(1)`。

修改 WebDAV 參數

下表列出一些常見的 WebDAV 特性：

表 10-1 WebDAV 參數

參數	說明
鎖定資料庫路徑	指定將在其中維護鎖定資料庫的目錄。
最小鎖定逾時	指定鎖定的最小使用期限 (以秒為單位)。值 -1 表示鎖定永遠不會過期。此值表示自動移除鎖定之前，要鎖定某元素的時間。
最大請求大小	指定 XML 請求內文的最大大小。您應配置此值以防止系統遭受可能的阻斷服務攻擊。預設值為 8192 (8K) 。
最大展開特性深度	指定 PROPFIND 請求的深度。 0 僅適用於指定的資源。這是預設值。 1 套用於指定的資源和下一層級。 infinity 套用於指定的資源及其包含的所有資源。限制此參數的大小，還可防止過度消耗記憶體。
預設所有者	集合的預設所有者。
URI	將啓用 WebDAV 的現有根目錄 URI。
最大 PROPFIND 深度	傳送至集合之 PROPFIND 請求的最大深度。
鎖定資料庫更新間隔	WebDAV 鎖定資料庫與磁碟同步的間隔。使用 0 可停用快取 WebDAV 鎖定資訊。
認證資料庫	使用的 ACL 認證資料庫。
認證方法	使用的認證方法。預設認證方法為「基本」。
認證提示文字	請求認證時向用戶端顯示的提示。
DAV ACL 資料庫	
最大項目數	允許單個資源上擁有的最大 ACE 數目。 0-2147.0483647.0 。指定 -1 表示無限制。
最大大小	某集合之 WebDAV ACL 資料庫的記憶體表示法最大大小。 0-2147.0483647.0 。指定 -1 表示無限制。
更新間隔	WebDAV ACL 資料庫與磁碟同步的間隔。 0.001-3600 秒。指定 0 可停用快取 WebDAV ACL 清單。
DAV 特性資料庫	
最大大小	WebDAV 特性資料庫檔案的最大大小。 0-2147.0483647.0 。指定 -1 表示無限制。

表 10-1 WebDAV 參數 (續)

參數	說明
更新間隔	WebDAV 特性資料庫與磁碟同步的間隔。0.001 – 3600 秒。指定 0 可停用快取 WebDAV 特性。

在伺服器層級停用 WebDAV

若要在伺服器層級停用 WebDAV，請執行以下指令：

```
wadm> disable-webdav --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1
```

請參閱 CLI 參考disable-webdav(1)。

管理 WebDAV 認證資料庫

在管理主控台上，按一下所選配置的 **WebDAV 標籤**，以編輯 WebDAV 認證資料庫設定。下表提供了對頁面中各欄位的簡短說明：

表 10-2 WebDAV 認證資料庫特性

特性	說明
認證資料庫	認證資料庫可讓您選取伺服器將用於認證使用者的資料庫。 預設為 keyfile 。
認證方法	<ul style="list-style-type: none">■ 基本 — 使用 HTTP 基本方法從用戶端取得認證資訊。僅當在伺服器上啟用 SSL 時，才會透過網路對使用者名稱和密碼進行加密。■ SSL — 使用用戶端憑證來認證使用者。若要使用此方法，必須為伺服器啟用 SSL。啟用加密後，您可以結合使用「Basic」和「SSL」方法。■ 摘要 — 使用認證機制，使得瀏覽器在根據使用者名稱和密碼進行認證時無須傳送明文格式的使用者名稱和密碼。瀏覽器利用使用者的密碼和 Web 伺服器提供的某些資訊，透過 MD5 演算法來建立摘要值。請注意，若要使用摘要，基本認證資料庫必須也支援摘要。僅當已安裝摘要認證外掛程式時，這才表示使用 digestfile 的檔案 auth-db 或 LDAP auth-db■ 其他 — 使用透過存取控制 API 建立的自訂方法。

表 10-2 WebDAV 認證資料庫特性 (續)

特性	說明
認證提示文字	<p>[提示進行認證] 選項可讓您輸入顯示在認證對話方塊中的訊息文字。您可以使用此文字來說明使用者需要輸入的內容。基於瀏覽器不同，使用者將看到該提示中大概前 40 個字元。</p> <p>通常，Web 瀏覽器會快取使用者名稱和密碼，並將其與提示文字相關聯。當使用者存取具有相同提示之伺服器的檔案和目錄時，無須再次輸入使用者名稱和密碼。如果您要讓使用者針對特定檔案和目錄再次進行認證，則只需變更該資源上 ACL 的提示。</p>

在啓用了 WebDAV 的伺服器上使用來源 URI 和 Translate:f 標頭

WebDAV 方法可以處理資源或集合的來源。HTTP 方法 (例如 GET 和 PUT) 會被 WebDAV 協定超載。因此，透過這些方法的請求可以是對資源來源的請求，也可以是對資源內容 (輸出) 的請求。

Microsoft 和許多其他 WebDAV 供應商解決了此問題，方法為隨請求傳送一個 Translate:f 標頭來通知伺服器所請求的是來源。爲了與常見的 WebDAV 用戶端 Microsoft WebFolders 互通，Sun Java System Web Server 7.0 將 Translate:f 標頭視爲對資源來源的請求。針對不傳送 Translate:f 標頭的用戶端，Sun Java System Web Server 定義了一個來源 URI。

對於啓用了 WebDAV 的集合，對 URI 的請求將擷取資源內容 (輸出)，而對來源 URI 的請求將擷取資源來源。帶有 Translate:f 標頭的 URI 請求被視爲是對來源 URI 的請求。

請注意，依預設，所有對資源來源的存取均被 dav-src ACL 拒絕，並在針對伺服器實例的 ACL 檔案中進行以下宣告：

```
deny (all) user = "anyone";
```

使用者可以透過在來源 URI 中增加存取權限來允許某個使用者存取該來源。

鎖定和解除鎖定資源

Sun Java System Web Server 允許伺服器管理員鎖定資源，以便序列化對資源的存取。通過使用鎖定，正在存取某特定資源的使用者可以確定其他使用者不會修改同一資源。這樣就解決了多個使用者共用伺服器資源時可能發生的「遺失更新」問題。由伺服器維護的鎖定資料庫將追蹤用戶端發出和正在使用的鎖定記號。

Sun Java System Web Server 支援 opaque-locktokenURI 方案，該方案被設計爲在全部資源中始終是唯一的。它使用了通用唯一識別碼 (UUID) 機制，如 ISO-1157.08 中所述。

Sun Java System Web Server 可識別以下兩種鎖定機制：

- 專用鎖定。
- 共用鎖定。

專用鎖定

專用鎖定是僅授予單一使用者存取資源的鎖定。其他使用者要想存取同一資源，只能等到移除對該資源的專用鎖定之後。

有時對於資源鎖定而言，專用鎖定機制太死板，代價也很大。例如，當程式當機或鎖定擁有者遺忘解除鎖定資源時，便需要通過鎖定逾時或由管理員移除專用鎖定來解決問題。

共用鎖定

共用鎖定允許多個使用者接收資源鎖定。這樣，任何具有適當存取權限的使用者都可以取得該鎖定。

使用共用鎖定时，鎖定擁有者之間可以使用任何其他通訊通道來協調工作。共用鎖定的目的是讓協作者瞭解誰還在使用同一資源。

最小鎖定逾時

您可以透過在 `server.xml` 檔案中配置 DAV 或 DAVCOLLECTION 物件的 `minlocktimeout` 屬性值來控制鎖定。`minlocktimeout` 屬性指定鎖定的最小使用期限 (以秒為單位)。此值表示鎖定被自動移除之前某元素被鎖定的時間。

這是一個可選屬性。如果將此值設定為 `-1`，則鎖定將永遠不會過期。如果將此值設定為 `0`，則可以使用在請求中指定的 `Timeout` 標頭來鎖定集合中的全部資源。

如果未指定 `Timeout` 標頭，則資源將被鎖定且永遠不會逾時。如果將請求中的 `Timeout` 標頭值設定為 `Infinite`，則同樣可將資源鎖定但永遠不會逾時。

如果對 WebDAV 資源的請求之 `Timeout` 標頭值大於或等於 `server.xml` 檔案中指定的 `minlocktimeout` 值，則資源的鎖定時間將是請求中指定的時間。

但是，如果請求中的 `Timeout` 標頭值小於 `server.xml` 檔案中指定的 `minlocktimeout` 值，則使用 `server.xml` 檔案中指定的 `minlocktimeout` 值來鎖定資源。

下表說明了 Sun Java System Web Server 處理鎖定請求的方式：

表 10-3 Sun Java System Web Server 處理鎖定請求的方式

如果請求中的 Timeout 標頭值設定為：	則資源：
Infinite	被鎖定且逾時值設定為 -1 (無限)
無	被鎖定且逾時值設定為 -1 (無限)
Second-xxx	<ul style="list-style-type: none">■ 如果 xxx 等於或大於 server.xml 中設定的 minlocktimeout 值，則使用 xxx 值來鎖定。或■ 如果 xxx 小於 server.xml 中設定的 minlocktimeout 值，則使用 server.xml 中設定的 minlocktimeout 值來鎖定。

備註 – 使用 CLI

若要透過 CLI 設定鎖定到期時間，請執行以下指令：

```
wadm> expire-lock --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --vs=config1
--collection-uri=/dav1 --lock-uri=/dav1/file.html
--opaque-token=opaquelocktoken
```

請參閱 CLI 參考expire-lock(1)。

在上述的範例中，opaque-token 可針對您要設定為過期的鎖定，指定其 ID。

若要透過 CLI 顯示現有鎖定，請執行以下指令：

```
wadm> list-locks --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --vs config1 --collection-uri=/dav1
```

請參閱 CLI 參考list-locks(1)。

使用 Java 和 Web 應用程式

本章說明為虛擬伺服器編輯 Java 設定的程序。您可以從管理主控台或者 wadm 指令行工具編輯 Java 設定。本章同時說明可以在 Sun Java System Web Server 中配置的各種 Java 資源。

本章同時說明如何在 Sun Java System Web Server 中部署 Java Web 應用程式。

- 第 147 頁的「配置 Java 以與 Sun Java System Web Server 搭配使用」
- 第 148 頁的「設定 Java 類別路徑」
- 第 149 頁的「配置 JVM」
- 第 151 頁的「部署 Java Web 應用程式」
- 第 153 頁的「配置 Servlet 容器」
- 第 154 頁的「配置伺服器生命週期模組」
- 第 156 頁的「配置 Java 資源」
- 第 165 頁的「配置 SOAP 認證提供者」
- 第 167 頁的「配置階段作業複製」
- 第 169 頁的「管理認證範圍」

配置 Java 以與 Sun Java System Web Server 搭配使用

本節可讓您啓用 Java，並為所選配置設定 Java 主目錄變數。

▼ 為配置啓用 Java

1 選取配置。

從配置清單中選取配置。按一下 [配置] 標籤以取得可用配置的清單。

2 按一下 [Java] > [一般] 標籤。

3 按一下 [啓用 Java] 核取方塊。

開啓/關閉配置的 Java 支援。啓用 Java 允許伺服器處理 Java 應用程式。

4 設定 Java 主目錄。

指定 Java SE 的位置。指定絕對路徑或伺服器的 config 目錄的相對路徑。

5 設定固定附加。

指定伺服器是否僅將每個 HTTP 請求處理執行緒附加至 JVM 一次 (否則，伺服器會附加/分離每個請求上的 HTTP 請求處理執行緒)。

備註 – 使用 CLI

若要為配置啓用 Java，請執行以下指令。

```
wadm> enable-java --user=admin --password-file=admin.pwd --host=serverhost  
--port=8989 --config=config1
```

請參閱 CLI 參考enable-java(1)。

設定 Java 類別路徑

本節可讓您為所選配置增加 JVM 類別路徑。

▼ 設定 Java 類別路徑

1 選取配置。

從配置清單中選取配置。按一下 [配置] 標籤以取得可用配置的清單。

2 按一下 [Java] > [路徑設定] 標籤。

編輯以下參數：

- **忽略環境類別路徑**—依預設為啓用狀態。
- **類別路徑前綴**—系統類別路徑前綴。如果要置換系統類別 (如 XML 剖析器類別)，您應僅加上系統類別路徑做為前綴。**請小心使用。**
- **伺服器類別路徑**—包含伺服器類別的類別路徑。唯讀清單。
- **類別路徑後綴**—附加至伺服器類別路徑。
- **原生庫路徑前綴**—作業系統原生庫路徑前綴。

- **位元碼預處理程式類別**—實作 `com.sun.appserv.BytecodePreprocessor` 的類別之完全合格名稱。執行執行階段類別設備的典型方法是透過預處理機制，由效能評測和監視工具使用類別預處理程式在 JVM 載入 Java 類別之前，將設備代碼插入 Java 類別的所需位置。爲了達到這個目的，類別預處理程式會與類別載入程式配合工作。

配置 JVM

若要在管理介面中設定 JVM 指令行選項，請執行以下工作：

▼ 配置 JVM

- 1 選取配置。
從配置清單選取配置。按一下 [配置] 標籤以取得清單。
- 2 按一下 [Java] > [JVM 設定] 標籤。
爲 JVM 配置設定。

增加 JVM 選項

您可以在此處指定值，以增加/刪除指令行 JVM 選項。

按一下 [增加 JVM 選項] 按鈕，來增加 JVM 選項。

JVM 選項的一些範例如下：`-Djava.security.auth.login.config=login.conf`、`-Djava.util.logging.manager=com.ipplanet.ias.server.logging.ServerLogManager` 與 `-Xms128m -Xmx256m`。

備註 – 使用 CLI

若要透過 CLI 增加 JVM 選項，請執行以下指令。

```
wadm> create-jvm-options --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 -Dhttp.proxyHost=proxyhost.com -Dhttp.proxyPort=8080
```

請參閱 CLI 參考 `create-jvm-options(1)`。

增加 JVM 效能評測器

JVM 效能評測器可協助您診斷並解決 Java 應用程式中的效能問題、記憶體發生的錯誤、多執行緒問題以及系統資源使用問題，以確保您的應用程式具有最高級的穩定性和延伸性。

▼ 增加 JVM 效能評測器

- 1 選取配置。
從配置清單選取配置。按一下 [配置] 標籤以取得可用配置的清單。
- 2 按一下 [Java] > [JVM 設定] 標籤。
- 3 按一下 [效能評測器] 區段中的 [新增] 按鈕。
- 4 提供以下參數的值：
 - 名稱—提供新 JVM 效能評測器的簡稱。
 - 已啟用—決定是否在執行階段啟用效能評測器。
 - 類別路徑—為效能評測器提供有效的類別路徑。(可選)。
 - 原生庫路徑—提供有效的原生庫路徑。(可選)。
 - JVM 選項—您可以為 CLI 指定其他 JVM 選項。

備註 - 使用 CLI

若要透過 CLI 增加 JVM 效能評測器，請執行以下指令。

```
wadm> create-jvm-profiler --user=admin --password-file=admin.pwd  
--host=serverhost --port=8989 --config=config1
```

請參閱 CLI 參考 `create-jvm-profiler(1)`。

為伺服器啟用 Java 除錯

JVM 能夠以除錯模式啟動，並可以附加 JPDA (Java 平台除錯程式架構) 除錯程式。您啟用除錯時，會同時啟用本機和遠端除錯。

Sun Java System Web Server 的除錯功能以 JPDA 軟體為基礎。若要啟用除錯，請執行以下作業。

▼ 啟用 JVM 除錯

- 1 選取配置。
從配置清單選取配置。按一下 [配置] 標籤以取得可用配置的清單。
- 2 按一下 [Java] > [JVM 設定] 標籤。
- 3 選取 [Java 除錯設定] 中的 [啟用除錯] 核取方塊。

- 4 按一下[新增]按鈕，以提供必要的 JVM 選項。

預設 JPDA 選項如下：

```
-Xdebug -Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=7896
```

如果您取代 `suspend=y`，則 JVM 將以暫停模式啟動，並在除錯程式與其相連接以前，一直處於暫停模式。但如果您希望在 JVM 啟動時便立即啟動除錯，該指令會非常有用。若要指定將 JVM 連接至除錯程式時使用的連接埠，請指定 `address=port_number`。請查閱 JPDA 文件，以取得除錯選項的清單。

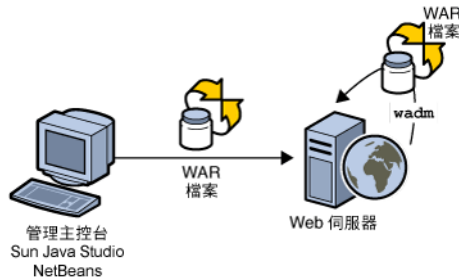
部署 Java Web 應用程式

增加 Web 應用程式

您可以將 Web 應用程式部署至任何現有的虛擬伺服器。

▼ 部署 Web 應用程式

- 開始之前
- 您應已經識別需要部署 Web 應用程式的虛擬伺服器。
 - 您具有 Web 應用程式歸檔 (.war 檔案) 或知道伺服器中 Web 應用程式的路徑。



可以透過 wadm、管理主控台和其他支援的 IDE 部署 Web 應用程式。

- 1 按一下[伺服器配置]下的[虛擬伺服器]標籤，以部署 Web 應用程式。
- 2 選取您需要部署 Web 應用程式的虛擬伺服器。
- 3 按一下[Web 應用程式]標籤 > [新增] 按鈕。
- 4 指定 Web 應用程式套裝軟體。

如果需要上傳 Web 應用程式歸檔，請按一下 [瀏覽] 按鈕並選取歸檔。您還可以選擇性地指定位於伺服器中的 Web 應用程式歸檔檔案。

- 5 指定 Web 應用程式的 URI。此為應用程式環境根目錄並與伺服器主機相關。
- 6 提供有關 Web 應用程式的簡短說明。
- 7 啟用/停用 JSP 預編譯。
啟用此指令即可預編譯 Web 應用程式中的所有 JSP，以提昇效能。
- 8 啟用應用程式。
當 Web 應用程式狀態設定為 [已停用] 時，您將無法請求使用該 Web 應用程式。然而，您可以隨時切換此選項，而無須將應用程式重新部署至實例。
- 9 部署應用程式。
按一下 [部署] 按鈕可以部署 Web 應用程式。
您可以使用指定的環境根目錄存取應用程式。例如，
`http://<your-server>:<port>/<URI>`

備註 – 使用 CLI

```
wadm> add-webapp --user=admin --password-file=admin.passwd --host=localhost
--port=8888 --config=config1 --vs=HOSTNAME --uri=/hello /home/test/hello.war
```

請參閱 CLI 參考 `add-webapp(1)`。

部署 Web 應用程式目錄

您可以使用 `-file-on-server` 選項，將管理伺服器主機電腦上的目錄部署至配置。執行下列指令：

```
wadm> add-webapp --user=admin-user --password-file=admin.passwd
--port=8989 --vs=vs1 --config=config1 --file-on-server
--uri=/mywebapp /space/tmp/mywebapp
```

部署時預編譯 JSP

若要在部署 Web 應用程式時預編譯 Web 應用程式中的 JSP，請如下所示使用 `-precompilejsp` 選項執行指令：

```
wadm> add-webapp --user=admin-user --password-file=admin.passwd
--port=8989 --vs=vs1 --config=config1 --file-on-server --uri=/mywebapp
--precompilejsp mywebapp.war
```


配置 Servlet 容器

本節說明配置 Servlet 容器的程序。

▼ 設定 Servlet 容器

- 1 選取配置。
從配置清單選取配置。按一下 [配置] 標籤以取得清單。
- 2 按一下 [Java] > [Servlet 容器]。

Servlet 容器全域參數

下表說明了 Servlet 容器頁面上的參數。

表 11-1 Servlet 容器參數

參數	說明
記錄層級	Servlet 容器的記錄詳細度。值可以為最詳細 (詳細度最高)、較詳細、詳細、資訊、警告、失敗、配置、安全性或災難 (詳細度最低)。
動態重新載入間隔	此參數定義某個時間段，在此時間段後，伺服器會檢查部署的 Web 應用程式是否有修改。值的範圍是 1 到 60，如果應停用動態重新載入，則值為 -1。
匿名角色	指定給所有主體的預設或匿名角色之名稱。預設角色為 ANYONE。
Servlet 池大小	每個 SingleThreadedServlet 要創設的 Servlet 實例數目。值的範圍從 1 到 4096。
派送程式最大深度	允許巢式請求派送的 Servlet 容器之最大深度。值的範圍可以是 0 到 2147.0483647.0。預設值為 20。
允許交叉環境	是否允許將請求派送程式派送至其他環境。預設值為 false。
編碼 Cookie	Servlet 容器是否編碼 Cookie 值。預設值為 true。
重複使用階段作業 ID	在為該用戶端建立新階段作業時，是否重複使用任何現有的階段作業 ID 號碼。預設值為 false。

表 11-1 Servlet 容器參數 (續)

參數	說明
安全階段作業 Cookie	動態/True/False。此參數可控制要在何種條件下將 JSESSIONID Cookie 標記為安全。僅當透過安全連線 (HTTPS) 接收到請求時，才使用動態 (預設) 將 Cookie 標記為安全。 使用 True 可一律標記為安全，使用 False 可一律不標記為安全。

配置伺服器生命週期模組

Java 伺服器生命週期模組是 Java 類別，它們偵聽伺服器生命週期事件，以在發生伺服器事件 (如啟動或停止) 時執行特定作業。

伺服器支援在 Web 伺服器環境下執行的持續時間較短或較長的 Java 作業。這些作業在伺服器啟動時會自動初始化，並會在伺服器關機時收到通知。因此，現在您可以立刻得知諸如實例化 singleton 和 RMI 伺服器等作業。

以下是關於伺服器生命週期的簡要說明。

伺服器生命週期簡介

- **初始化**—此階段包括讀取配置、初始化內建子系統；命名、安全性和記錄服務；以及建立 Web 容器。
- **啟動**—此階段包括載入和初始化已部署的應用程式。
- **服務**—伺服器已準備好處理服務請求。
- **關機**—此階段會停止和銷毀載入的應用程式。系統正在準備關機。
- **終止**—此階段會終止內建子系統和伺服器執行階段環境。此階段之後不會再有任何作業。
- **重新配置**—暫態伺服器狀態，伺服器執行緒在此狀態下會動態重新配置 (而伺服器處於服務狀態)。此階段可在伺服器生命週期中出現多次。

▼ 增加生命週期模組

- 1 選取配置。
從配置清單選取配置。若要檢視配置清單，請按一下 [配置] 標籤。
- 2 按一下 [Java] > [生命週期模組] 標籤。
- 3 按一下 [新增] 按鈕。
提供以下參數的值：

- **名稱**—為新生命週期模組提供有效的唯一名稱。
- **已啟用**—如果您要啟用此生命週期模組，請使用此選項。
- **類別名稱**—完全合格的 Java 類別名稱。類別應該實作 `com.sun.appserv.server.LifecycleListener` 介面。如需有關使用此介面的更多資訊，請參閱「*Developer's Guide*」。
- **類別路徑**—可選。您可以指定偵聽程式類別的類別路徑。
- **載入順序**—大於 100。載入生命週期事件偵聽程式的順序 (數字順序)。建議選擇大於或等於 100 的載入順序，以避免與內部生命週期模組發生衝突。
- **載入失敗時**—如果啟用此選項，則伺服器不會將偵聽程式類別丟出的異常視為嚴重異常，因此其繼續正常啟動。依預設停用。
- **說明**—提供關於生命週期模組的簡短說明。
- **特性**—這些特性可用來將引數傳送至 Java 生命週期模組。若要增加新的特性，請按一下 [增加特性] 按鈕，然後輸入名稱、值和說明的文字。



注意—將從主伺服器執行緒同步呼叫各伺服器生命週期偵聽程式類別，因此必須特別謹慎，以確保偵聽程式類別不會阻斷伺服器。如果合適，偵聽程式類別會建立執行緒，但在關機/終止階段必須將其停止。

▼ 刪除生命週期模組

- 1 選取配置。
從配置清單選取配置。若要檢視配置清單，請按一下 [配置] 標籤。
- 2 按一下 [Java] > [生命週期模組] 標籤。
- 3 選取生命週期模組，然後按一下 [刪除生命週期模組] 按鈕。

備註 – 使用 CLI

以下範例描述如何透過 `com.MyLifecycleModule` 類別實作，為 `test` 配置建立名為 `myLifecycleModule` 的 Java 生命週期模組。

```
wadm> create-lifecycle-module --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1
--class=com.sun.webserver.tests.LifecycleClass LifecycleTest
```

請參閱 CLI 參考 `create-lifecycle-module(1)`。

若要列出 Java 生命週期模組，請執行以下指令：

```
wadm> list-lifecycle-modules --config=test
```

請參閱 CLI 參考 `list-lifecycle-modules(1)`。

若要為 Java 生命週期模組增加特性，請執行以下指令：

```
wadm> create-lifecycle-module-userprop --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --module=LifecycleTest info=Testing
```

請參閱 CLI 參考 `create-lifecycle-module-userprop(1)`。

若要修改 Java 生命週期模組特性，請執行以下指令：

```
wadm> set-lifecycle-module-prop --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1 --module=LifecycleTest
class-path=/space
```

請參閱 CLI 參考 `set-lifecycle-module-prop(1)`。

配置 Java 資源

Web 應用程式可以存取多種資源，例如資源管理員、資料來源 (如 SQL 資料來源)、郵件階段作業以及 URL 連線工廠。Java EE 平台透過 Java Naming and Directory Interface (JNDI) 服務向應用程式展示此類資源。

Sun Java System Web Server 允許建立與管理以下 Java EE 資源：

- JDBC 資料來源。
- JDBC 連線池。
- Java 郵件階段作業。
- 自訂資源。
- 外部 JNDI 資源。

配置 JDBC 資源

JDBC 資料來源是一種 Java EE 資源，可以使用 Sun Java System Web Server 建立與管理。

JDBC API 是用於連線關聯式資料庫系統的 API。JDBC API 具有兩個組件：

- 應用程式元件用於存取資料庫的應用程式層級介面。
- 將 JDBC 驅動程式連結到 Java EE 平台的服務提供者介面。

JDBC 資料來源物件是採用 Java 程式設計語言的資料來源實作。從基本上來看，資料來源是儲存資料的設備。可以如大公司複雜的資料庫一樣複雜，也可以如具有列和欄的檔案一樣簡單。JDBC 資料來源是 Java EE 資源，可以透過 Sun Java System Web Server 建立與管理。

JDBC API 提供具有標準 SQL 資料庫存取介面的 Java 類別集，以確保對廣泛關聯式資料庫的統一存取權限。

透過使用 JDBC，可以將 SQL 描述傳送至任何虛擬資料庫管理系統 (DBMS)。它同時作為關聯式 DBMS 與物件 DBMS 的介面。

增加 JDBC 資源

若要透過 CLI 增加 JDBC 資源，請執行以下指令。

```
wadm> create-jdbc-resource --user=admin --password-file=admin.pwd --host=serverhost  
--port=8989 --config=config1 --datasource-class=oracle.jdbc.pool.OracleDataSource jdbc
```

請參閱 CLI 參考 `create-jdbc-resource(1)`。

在上述的範例中，`com.pointbase.jdbc.jdbcDataSource` 代表 JDBC 驅動程式類別。

如需關於支援的 JDBC 驅動程式清單，請參閱第 157 頁的「[Sun Java System Web Server 支援的 JDBC 驅動程式](#)」。

Sun Java System Web Server 支援的 JDBC 驅動程式

下表將提供在增加新的 JDBC 資源時，需要配置的一般 JDBC 驅動程式及其特性清單。請參閱第 160 頁的「[增加新的 JDBC 資源](#)」。

表 11-2 一般與受支援的 JDBC 驅動程式清單

驅動程式	類別名稱	特性
Oracle 驅動程式	oracle.jdbc.pool.OracleDataSource	<ul style="list-style-type: none">■ url■ 使用者■ 密碼
Oracle 的 SJS JDBC 驅動程式	com.sun.sql.jdbcx.oracle.OracleDataSource	<ul style="list-style-type: none">■ 伺服器名稱■ 連接埠號■ 使用者■ 密碼■ SID
DB2 IBM 驅動程式	com.ibm.db2.jdbc.DB2DataSource	<ul style="list-style-type: none">■ 伺服器名稱■ 資料庫名稱■ 連接埠號■ 使用者■ 密碼■ 驅動程式類型
DB2 的 SJS JDBC 驅動程式	com.sun.sql.jdbcx.db2.DB2DataSource	<ul style="list-style-type: none">■ 資料庫名稱■ 位置名稱■ 套裝軟體名稱■ 密碼■ 連接埠號■ 伺服器名稱■ 使用者
MS SQLServer 驅動程式	com.ddtek.jdbcx.sqlserver.SQLServerDataSource	<ul style="list-style-type: none">■ 資料庫名稱■ 密碼■ 使用者■ 伺服器名稱■ 連接埠號
MS 的 SJS JDBC 驅動程式	com.sun.sql.jdbcx.sqlserver.SQLServerDataSource	<ul style="list-style-type: none">■ 資料庫名稱■ 密碼■ 使用者■ 伺服器名稱■ 連接埠號
Sybase 驅動程式	com.sybase.jdbcx.SybDataSource	<ul style="list-style-type: none">■ 資料庫名稱■ 密碼■ 連接埠號■ 伺服器名稱■ 使用者

表 11-2 一般與受支援的 JDBC 驅動程式清單 (續)

驅動程式	類別名稱	特性
Sybase 的 SJS JDBC 驅動程式	<code>com.sun.sql.jdbcx.sybase. SybaseDataSource</code>	<ul style="list-style-type: none"> ■ 資料庫名稱 ■ 密碼 ■ 使用者 ■ 連接埠號 ■ 伺服器名稱
MySQL MM 驅動程式	<code>org.gjt.mm.mysql.jdbc2.optional. MySQLDataSource</code>	<ul style="list-style-type: none"> ■ 伺服器名稱 ■ 連接埠 ■ 資料庫名稱 ■ 使用者 ■ 密碼
Informix 驅動程式	<code>com.informix.jdbcx.IfxDataSource</code>	<ul style="list-style-type: none"> ■ 連接埠號 ■ 資料庫名稱 ■ IfxIFXHOST (執行 Informix 資料庫的電腦之 IP 位址或主機名稱) ■ 伺服器名稱 ■ 使用者 ■ 密碼
Informix 的 SJS JDBC 驅動程式	<code>com.sun.sql.jdbcx.informix. InformixDataSource</code>	<ul style="list-style-type: none"> ■ 資料庫名稱 ■ informixServer (您要連線的 Informix 資料庫伺服器名稱) ■ 密碼 ■ 連接埠號 ■ 伺服器名稱
PostgreSQL 驅動程式	<code>org.postgresql.ds. PGSimpleDataSource</code>	<ul style="list-style-type: none"> ■ 伺服器名稱 ■ 資料庫名稱 ■ 連接埠號 ■ 使用者 ■ 密碼
Apache Derby 驅動程式	<code>org.apache.derby.jdbc. EmbeddedDataSource</code>	<ul style="list-style-type: none"> ■ 資料庫名稱 ■ 使用者 ■ 密碼

管理 JDBC 資源

▼ 增加新的 JDBC 資源

- 1 選取配置。
從配置清單選取配置。按一下 [配置] 標籤以取得清單。
- 2 按一下 [Java] > [資源] 標籤。
- 3 按一下 [JDBC 資源] 區段下的 [新增] 按鈕。
- 4 選取驅動程式供應商。
指定唯一 JNDI 名稱值，並從可用清單中選取 JDBC 驅動程式供應商。
- 5 提供 JDBC 資源特性。
根據上一步選取的 JDBC 驅動程式供應商，自動寫入驅動程式的類別名稱和 JDBC 資源特性。
- 6 查閱。
檢視摘要，然後按一下 [完成] 以建立新的 JDBC 資源。

管理 JDBC 連線池

配置 JDBC 連線池

在 Sun Java System Web Server 7.0 中，要透過 `jdbc-resource` 元素配置 JDBC 連線池。下面列出的步驟，可以配置最簡單的連線池。在此例中，連線池使用 Oracle JDBC 驅動程式。

▼ 建立 JDBC 連線池

- 1 啓動 `wadm`。
- 2 建立 JDBC 資源。
使用基本配置建立 JDBC 資源。其他屬性可用於微調連線池。請參閱線上手冊以瞭解更多屬性和範例。

```
wadm> create-jdbc-resource --config=test  
--datasourceclass=oracle.jdbc.pool.OracleDataSource jdbc/MyPool
```


3 配置供應商特有特性。

此特性用於配置驅動程式的供應商特有特性。以下範例中的 `url`、`user` 與 `password` 特性將增加至 JDBC 資源。

```
wadm> add-jdbc-resource-userprop --config=test --jndi-name=jdbc/MyPool
url=jdbc:oracle:thin:@hostname:1521:MYSID user=myuser password=mypassword
```

4 啟用連線驗證。

您可為連線池啟用連線驗證。如果使用此選項，連線在傳遞至應用程式之前會經過驗證。這可讓 Web 伺服器自動重新建立資料庫連接 (由於網路故障或資料庫伺服器當機，資料庫不可用時)。對連接進行驗證會耗用額外的時間，並且會略微降低效能。

```
wadm> set-jdbc-resource-prop --config=test --jndi-name=jdbc/MyPool
connection-validation-table-name=test connection-validation=table
```

5 變更預設池設定。

在此例中，讓我們變更連線的最大數目。

```
wadm> set-jdbc-resource-prop --config=test --jndi-name=jdbc/MyPool
max-connections=100
```

6 部署配置。

```
wadm> deploy-config test
```

7 提供包含 JDBC 驅動程式的 JAR 檔案。

您需要為伺服器提供可實作驅動程式的類別。可使用的方法有兩種：

- 將驅動程式的 JAR 檔案複製到伺服器實例程式庫目錄。這是最簡單的方法，因為實例程式庫目錄中的 JAR 檔案會自動載入並提供給伺服器。
- 修改 JVM 的 *class-path-suffix* 以包含 JDBC 驅動程式的 JAR 檔案。

```
wadm> set-jvm-prop --config=test class-path-suffix=/export/home/lib/classes12.jar
```

8 在 Web 應用程式中的用法。

- 修改 `WEB-INF/web.xml`。

```
<web-app>
...
  <resource-ref>
    <description>JDBC Connection Pool</description>
    <res-ref-name>jdbc/myJdbc</res-ref-name>
    <res-type>javax.sql.DataSource</res-type>
    <res-auth>Container</res-auth>
  </resource-ref>
...
</web-app>
```

- 修改 WEB-INF/sun-web.xml。

```
<sun-web-app>
...
  <resource-ref>
    <res-ref-name>jdbc/myJdbc</res-ref-name>
    <jndi-name>jdbc/MyPool</jndi-name>
  </resource-ref>
...
</sun-web-app>
```

- 使用連線池。

```
Context initContext = new InitialContext();
Context webContext = (Context)context.lookup("java:/comp/env");

DataSource ds = (DataSource) webContext.lookup("jdbc/myJdbc");
Connection dbCon = ds.getConnection();
```

註冊自訂資源

可以按照此工作來註冊包含實例的自訂資源。

▼ 增加自訂資源

- 1 選取配置。
從配置清單選取配置。按一下 [配置] 標籤以取得清單。
- 2 按一下 [Java] > [資源] 標籤。
- 3 按一下 [自訂資源] 區段中的 [新增] 按鈕。

自訂資源的特性

下表說明了可用於建立自訂資源的特性。

表 11-3 自訂資源特性

特性	說明
JNDI 名稱	為自訂資源提供唯一的 JNDI 名稱。
已啟用	決定在執行階段是否啟用此 JDBC 資源。
資源類型	此資源之完全合格的類型。

表 11-3 自訂資源特性 (續)

特性	說明
工廠類別	創設此類型資源的類別。實作 <code>javax.naming.spi.ObjectFactory</code> 的使用者寫入工廠類別的完全合格名稱。
說明	提供對自訂資源的簡短說明。
特性	按一下 [增加特性] 按鈕，選擇性地提供 CLI 特性。

備註 – 使用 CLI

若要透過 CLI 建立自訂資源，請執行以下指令：

```
wadm> create-custom-resource --user=admin --password-file=admin.pwd --host=serverhost
--port=8989 --config=config1 --res-type=samples.jndi.customResource.MyBean
--factory-class=samples.jndi.customResource.MyCustomConnectionFactory custom
```

請參閱 CLI 參考 `create-custom-resource(1)`。

使用外部 JNDI 資源

建立外部 JNDI 資源

此選項可讓您建立外部 Java Naming and Directory Interface (JNDI) 資源。您需要外部 JNDI 資源，才可存取儲存在內部 JNDI 儲存庫中的資源。

▼ 增加外部 JNDI 資源

- 1 選取配置。
從配置清單選取配置。按一下 [配置] 標籤以取得清單。
- 2 按一下 [Java] > [資源] 標籤。
- 3 按一下 [外部 JNDI] 區段中的 [新增] 按鈕。

外部 JNDI 資源的特性

下表說明了在增加新的外部 JNDI 資源時可以使用的特性。

表 11-4 外部 JNDI 資源特性

特性	說明
JNDI 名稱	為新的外部 JNDI 資源提供唯一名稱。
已啓用	決定在執行階段是否啓用此外部 JNDI 資源。
外部 JNDI 名稱	外部 JNDI 資源的名稱。
資源類型	此資源之完全合格的類型。
工廠類別	創設此類型資源的類別。
說明	提供對自訂資源的簡短說明。
特性	按一下 [增加特性] 按鈕，選擇性地提供 CLI 特性。

備註 – 使用 CLI

若要透過 CLI 建立外部 JNDI 資源，請執行以下指令：

```
wadm> create-external-jndi-resource --user=admin
--password-file=admin.pwd --host=serverhost --port=8989 --config=config1
--res-type=org.apache.naming.resources.Resource
--factory-class=samples.jndi.externalResource.MyExternalConnectionFactory
--jndilookupname=index.html external-jndi
```

請參閱 CLI 參考create-external-jndi-resource(1)。

配置郵件資源

JMS 目標是 Java EE 資源，可透過 Sun Java System Web Server 建立與管理。

許多網際網路應用程式都需要傳送電子郵件通知的功能。因此，Java EE 平台包括 JavaMail API 以及 JavaMail 服務提供者，可讓應用程式元件傳送網際網路郵件。

▼ 增加郵件資源

- 1 選取配置。
從配置清單選取配置。按一下 [配置] 標籤以取得清單。
- 2 按一下 [Java]>[資源] 標籤。
- 3 按一下 [郵件資源] 區段中的 [新增] 按鈕。

郵件資源的特性

下表說明了在增加新的郵件資源時可以使用的特性。

表 11-5 郵件資源特性

特性	說明
JNDI 名稱	為新的郵件資源提供唯一名稱。
已啓用	決定在執行階段是否啓用此郵件資源。
使用者	在郵件伺服器中註冊的有效使用者名稱。
寄件者	伺服器傳送郵件的源電子郵件位址。
主機	郵件伺服器的主機名稱/IP 位址。
儲存協定	用於擷取郵件的協定。
儲存協定類別	儲存協定的儲存服務提供者實作。實作儲存協定的類別之完全合格類別名稱。預設類別為 <code>com.sun.mail.imap.IMAPStore</code> 。
傳輸協定	用於傳送郵件的協定。
傳輸協定類別	傳輸協定的傳輸服務提供者實作。實作傳輸協定的類別之完全合格類別名稱。預設類別為 <code>com.sun.mail.smtp.SMTPTransport</code> 。

備註 – 使用 CLI

若要建立郵件資源，請執行以下指令：

```
wadm> create-mail-resource --config=test --server-host=localhost
--mail-user=nobody --from=xyz@foo.com mail/Session
```

請參閱 CLI 參考 `create-mail-resource(1)`。

配置 SOAP 認證提供者

容器規格的 Java 認證服務提供者介面可定義標準服務提供者介面，認證機制提供者可透過該介面與容器整合。您可以使用管理主控台增加新的 SOAP 認證提供者。

▼ 增加 SOAP 認證提供者

1 選取配置。

從配置清單選取配置。按一下 [配置] 標籤以取得清單。

- 2 按一下 [Java] > [Web 服務] 標籤。
- 3 按一下 [SOAP 認證提供者] 區段中的 [新增] 按鈕。

SOAP 認證提供者參數

下表說明了新的 SOAP 認證提供者頁面上的參數。

表 11-6 SOAP 認證提供者參數

參數	說明
名稱	輸入新 SOAP 認證提供者的簡稱。
類別名稱	實作提供者的類別。實作 javax.security.auth.XXX 的類別之完全合格類別名稱。
請求認證來源	此屬性定義了對郵件層寄件者認證 (如使用者名稱/密碼) 或內容認證 (如要套用於請求郵件的數位簽名) 的要求。值 (認證策略) 可以是寄件者，也可以是內容。如果未指定此引數，則不需要對請求進行來源認證。
請求認證接受者	此屬性定義了郵件收件者對其寄件者的郵件層認證之要求，如透過 XML 加密。值可以是內容前或內容後。
回應認證來源	此屬性定義了對郵件層寄件者認證 (如使用者名稱/密碼) 或內容認證 (如要套用於回應郵件的數位簽名) 的要求。值 (認證策略) 可以是寄件者，也可以是內容。如果未指定此引數，則不需要對回應進行來源認證。
回應認證接受者	此屬性定義了回應郵件收件者對其寄件者的郵件層認證之要求，如透過 XML 加密。
特性	按一下 [增加特性] 按鈕，提供其他 CLI 特性。

備註 – 使用 CLI

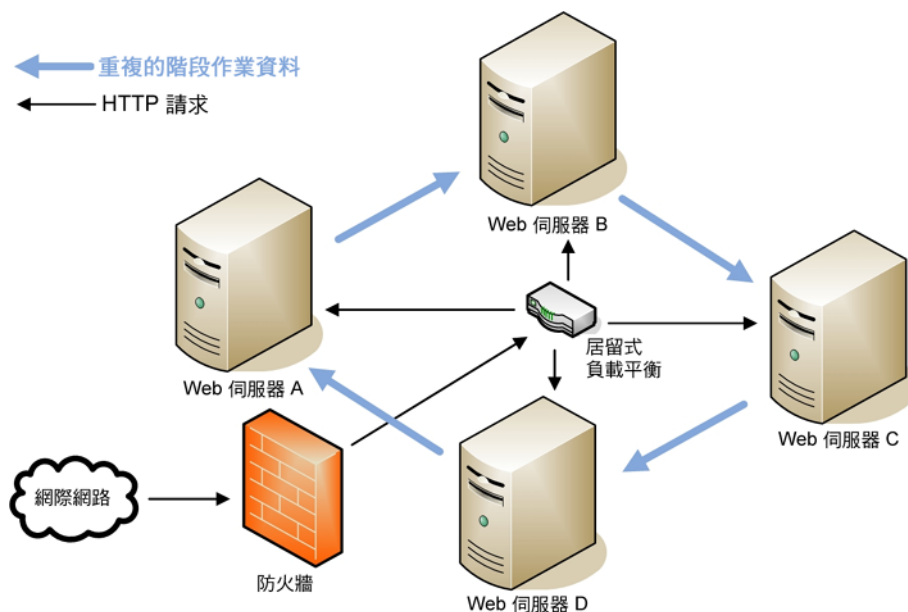
若要使用 CLI 增加 SOAP 認證提供者，請執行以下指令。

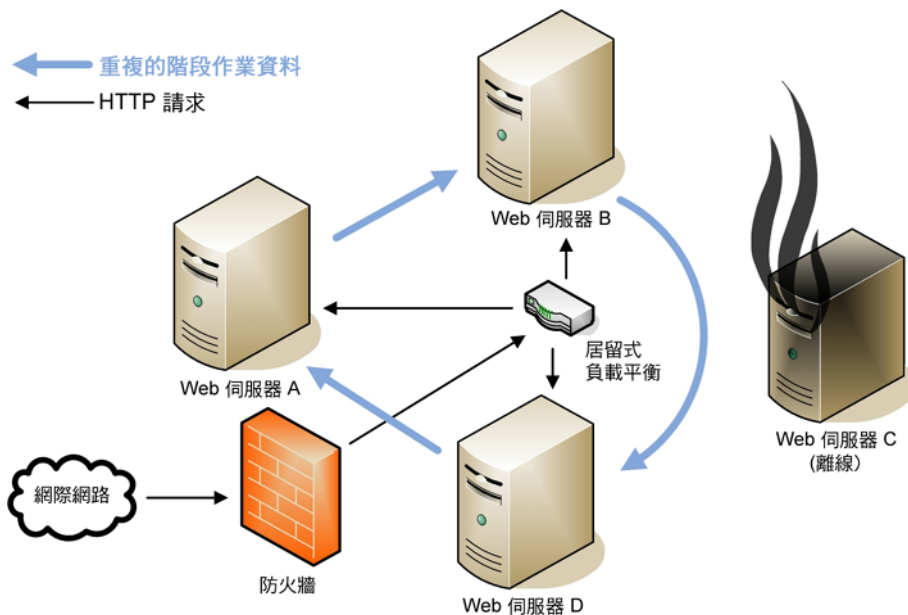
```
wadm> create-soap-auth-provider --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 --config=config1
--class-name=javax.security.auth.soapauthprovider soap-auth
```

請參閱 CLI 參考create-soap-auth-provider(1)。

配置階段作業複製

Sun Java System Web Server 7.0 支援階段作業複製，讓 Web 應用程式擁有高可用性。階段作業複製為實現此目的，會將 HTTP 階段作業從一個實例複製到同一叢集中的另一個伺服器實例。因此，每個 HTTP 階段作業在遠端實例上都有一個備份副本。如果發生故障而導致叢集中的一個實例不可用，該叢集仍可延續階段作業。





上圖描述反向代理伺服器設定的四個節點之間，階段作業複製的常見方式。請注意，Web 伺服器 C 離線時，Web 伺服器 B 的階段作業資料會複製到 Web 伺服器 D。

設定階段作業複製

本節說明了為所選配置設定階段作業複製特性的步驟。

▼ 設定階段作業複製

- 1 選取配置。
從配置清單中選取配置。按一下 [配置] 標籤以取得清單。
- 2 按一下 [Java] > [階段作業複製]。

修改階段作業複製參數

下表說明了階段作業複製頁面上可用的參數。

表 11-7 階段作業複製參數

參數	說明
連接埠	管理伺服器在其上進行偵聽的連接埠號。預設連接埠為 8888。
已啓用	為所選配置啓用階段作業複製。
已加密	是否在複製之前加密階段作業資料。預設值為 False。
密碼	叢集成員複製階段作業資料時所用的密碼組 (演算法、模式、填補)。
Getattribute 觸發複製	呼叫 HttpSession.getAttribute 方法是否應導致備份階段作業。預設值為 true。
複製探索最大躍點	嘗試尋找階段作業備份時，應連絡實例的最大數目。值的範圍從 1 到 2147.0483647.0，或 -1 (表示無限制)。
啓動探索逾時	實例嘗試連絡其指定的備份實例所花費的最長時間 (以秒為單位)。值的範圍為 0.001 到 3600。
Cookie 名稱	輸入追蹤擁有階段作業的實例之 Cookie 名稱。

管理認證範圍

基於 Java EE 的安全模型提供用於識別和認證使用者的安全性範圍。

該認證程序會透過 Java 範圍對使用者進行驗證。每個範圍都是由一組使用者、選擇性的群組對映以及用於驗證認證請求的認證邏輯組成。認證請求通過配置範圍的驗證，並建立安全性環境之後，便會將該身份識別套用於所有後續授權決定。

備註 – Java 範圍類似於 auth-db (認證資料庫)，不同的是 auth-db 由 ACL 引擎 (根據 ACL 檔案中的規則) 來使用，而 Java 範圍由 Java Servlet 存取控制規則 (在每一個 Web 應用程式的 web.xml 檔案中指定) 來使用。

一個伺服器實例可以具有任意數量的配置範圍。配置資訊位於 server.xml 檔案的 auth-realm 元素中。

下表定義了 Sun Java System Web Server 7.0 中支援的各種範圍類型

表 11-8 範圍類型

範圍	說明
File	<p>File 範圍是首次安裝 Sun Java System Web Server 時的預設範圍。該範圍的設定非常簡便，為開發人員提供了巨大的方便。</p> <p>File 範圍會根據文字檔中儲存的使用者資料來認證使用者。Java 範圍類似於 auth-db (認證資料庫)，不同的是 auth-db 由 ACL 引擎 (根據 ACL 檔案中的規則) 來使用，而 Java 範圍由 Java Servlet 存取控制規則 (在每一個 Web 應用程式的 web.xml 中指定) 來使用。</p>
LDAP	<p>LDAP 範圍可讓您使用 LDAP 資料庫，以獲得使用者安全性資訊。每項 LDAP 目錄服務都是具有唯一識別碼的屬性集合。LDAP 範圍非常適合部署生產系統。</p> <p>為了針對 LDAP 範圍認證使用者，您必須在 LDAP 目錄中建立所需的使用者。您可以從 Administration Server 的 [使用者與群組] 標籤執行此作業。也可以從您的 LDAP 目錄產品的使用者管理主控台執行此動作。</p>
PAM	<p>PAM (亦稱為 Solaris) 範圍會將認證委託給 Solaris PAM 堆疊。如果使用 PAM auth-db，僅 Solaris 9 和 10 支援此範圍，並且伺服器實例必須以超級使用者身份執行。</p>
Certificate	<p>Certificate 範圍支援 SSL 認證。Certificate 範圍會在 Sun Java System Web Server 的安全性環境中設定使用者身份識別，並在其中寫入用戶端憑證中的使用者資料。然後，Java EE 容器將根據每個使用者憑證中的使用者 DN 執行認證處理。該範圍會透過 X.509 憑證，以 SSL 或 TLS 用戶端認證機制來認證使用者。</p>
Native	<p>Native 範圍是一個特殊範圍，能為核心 ACL 型認證模型與 Java EE/Servlet 認證模型之間搭起一個橋樑。透過使用適用於 Java Web 應用程式的 Native 範圍，可以使 ACL 子系統執行認證 (而非使 Java Web 容器執行該作業)，也可以使該身分可用於 Java Web 應用程式。</p> <p>呼叫認證作業後，Native 範圍會將此認證委託給核心認證子系統。例如從使用者角度來看，此項作業實質上等同於 LDAP 範圍將認證委託給配置的 LDAP 伺服器。當本端範圍處理群組成員關係查詢時，也會將這些查詢授權給核心認證子系統。從 Java Web 模組和開發人員角度來看，Native 範圍與其他任何可與 Web 模組配合使用的 Java 範圍沒有絲毫差異。</p>
自訂	<p>您可以為其他資料庫建立範圍，如 Oracle，以便透過使用可插接式 JAAS 登入模組並執行範圍來滿足特殊需求。</p>

以下小節說明了有關增加新認證範圍的步驟。

▼ 增加認證範圍

1 選取配置。

選取您需要為其增加新的認證範圍的配置。按一下 [配置] 標籤，然後選取配置。

2 按一下 [Java] > [安全性] 標籤。

3 按一下 [新增認證] 按鈕。

4 提供範圍詳細資訊。

- **名稱**—輸入範圍的簡稱。此名稱用於參考範圍，例如 web.xml 中的範圍。
- **類別**—如果您要配置自訂範圍，請輸入實作自訂範圍的完整 Java 類別名稱。不必為每個內建範圍輸入類別。
- **類型**—選取範圍的類型。請參閱上一節中討論的 Java 範圍類型。
- **特性**—增加範圍特定的特性。例如 `property name="file" value="instance_dir/config/keyfile"` 與 `property name="jaas-context" value="fileRealm"`。

備註 – 使用 CLI

若要透過 CLI 增加認證範圍，請執行以下指令。

```
wadm> create-auth-realm --user=admin --password-file=admin.pwd --host=serverhost  
--port=8989 --config=config1 basic
```

請參閱 CLI 參照 `create-auth-realm(1)`。

指定內建認證範圍類型的名稱。類型可以為 `file`、`ldap`、`pam`、`native` 或 `certificate`。

使用搜尋集合

伺服器包含搜尋功能，允許使用者在伺服器上搜尋文件並將搜尋結果顯示在網頁上。伺服器管理員依據使用者要搜尋的文件 (稱為**集合**) 來建立文件索引，還可以自訂搜尋介面以滿足使用者的需要。

如需有關查詢搜尋集合的更多資訊，請參閱**搜尋線上說明**。

- 第 173 頁的「關於搜尋」
- 第 174 頁的「配置搜尋特性」
- 第 175 頁的「配置搜尋集合」
- 第 177 頁的「排程集合更新」
- 第 179 頁的「執行搜尋」
- 第 179 頁的「搜尋頁面」
- 第 179 頁的「進行查詢」
- 第 180 頁的「進階搜尋」
- 第 181 頁的「檢視搜尋結果」
- 第 181 頁的「自訂搜尋頁面」

關於搜尋

在安裝 Sun Java System Web Server 期間，搜尋功能會與其他 Web 元件一同安裝。搜尋是在虛擬伺服器層級而不是伺服器實例層級上配置和管理的

透過管理主控台中 [虛擬伺服器] 標籤下的 [搜尋] 標籤，您可以執行以下作業：

- 啓用和停用搜尋功能
- 建立、修改、刪除搜尋集合以及為搜尋集合重新建立索引
- 為搜尋集合建立、修改和移除排定的維護工作

從管理介面取得的資訊儲存在 `<server-root>/config/server.xml` 檔案中，並在該檔案的 VS 元素內進行對映。

伺服器管理員可以自訂搜尋查詢和搜尋結果頁面。此舉可將公司標誌增加到頁面中以改成自己的品牌，或變更顯示搜尋結果的方式。在先前發行版本中，上述作業是透過使用式樣檔案來完成的。

沒有針對搜尋的全域「開啓」或「關閉」功能。而是提供了一個預設的搜尋 Web 應用程式，然後在特定虛擬伺服器上啓用或停用該應用程式。此搜尋應用程式提供了用於查詢集合和檢視結果的基本網頁。此搜尋應用程式包含 JSP 範例，說明如何使用搜尋標籤程式庫來建立自訂搜尋介面。

備註 – Sun Java System Web Server 不提供搜尋結果的存取檢查。由於潛在的安全性模型及範圍的數量很多，因此無法透過搜尋應用程式執行安全檢查及篩選結果。伺服器管理員負責確保使用相應的安全機制來保護內容。

配置搜尋特性

透過啓用伺服器隨附的搜尋應用程式，為虛擬伺服器啓用搜尋。

備註 – 必須首先啓用 Java Web 容器，然後才能啓用搜尋。

確定已為您要配置的虛擬伺服器啓用 Java 後，請執行以下步驟來啓用搜尋：

1. 按一下 **[配置] 標籤**。
2. 從配置清單中選取配置。
3. 按一下 **[虛擬伺服器] 標籤**。
4. 從虛擬伺服器清單中選取虛擬伺服器。
5. 按一下 **[搜尋] 標籤**。
6. 在 **[搜尋應用程式]** 區段中，按一下 **[已啓用] 核取方塊**，以啓用搜尋應用程式。

其他可以配置的參數如下所示：

- **URI**。如果您計劃使用自訂搜尋應用程式，請輸入 URI；如果要使用預設搜尋應用程式，則不需要在此指定值。
- **最大符合項目數**。指定搜尋查詢中擷取的最大結果數。
- **已啓用**。核取此選項可啓用預設搜尋應用程式。

備註 – 使用 CLI

若要透過 CLI 設定搜尋特性，請在 CLI 中執行以下指令：

```
wadm> set-search-prop --user=admin --password-file=admin.pwd --host=serverhost  
--port=8888 --no-ssl --rcfile=null --config=config1 --vs=config1_vs_1  
enabled=true max-hits=1200
```

請參閱 CLI 參照 `set-search-prop(1)`。

配置搜尋集合

搜尋需要一個具備可搜尋資料的資料庫，使用者將依這些資料進行搜尋。伺服器管理員將建立此資料庫（稱為集合），該資料庫會為伺服器上的文件建立索引並儲存關於這些文件的資訊。伺服器管理員為全部或部分伺服器文件建立索引後，便可以使用標題、建立日期和作者等資訊進行搜尋。

備註 – 關於搜尋集合：

- 集合特定於被管理的虛擬伺服器。
 - 只有在虛擬伺服器中可見的文件，才會顯示在管理介面中，才可為其建立索引。
 - 伺服器上可存在的集合數沒有限制。
 - 搜尋集合中的文件不特定於任何一種字元編碼，這表示搜尋集合可以與多種編碼相關聯。
-

支援的格式

您可以為以下格式的文件建立索引，然後進行搜尋。

1. HTML 文件— .html 和 .htm
2. ASCII 一般文字— .txt
3. PDF

增加搜尋集合

若要增加新的集合，請執行以下作業：

1. 按一下 **[配置]** 標籤。
2. 從配置清單中選取配置。
3. 按一下 **[虛擬伺服器]** 標籤。
4. 從虛擬伺服器清單中選取虛擬伺服器。

5. 按一下 **[搜尋] 標籤**。
6. 在 **[搜尋集合] 區段**下，按一下 **[增加搜尋集合] 按鈕**，以增加新的搜尋集合。

下節說明頁面中用於建立新搜尋集合的欄位：

1. 提供搜尋集合資訊

- a. **集合名稱**—輸入搜尋集合的唯一名稱。

備註—多位元組字元不得做為集合名稱。

- b. **顯示名稱**—(可選)將在搜尋查詢頁面中顯示為集合名稱。如果您未指定顯示名稱，則集合名稱將做為顯示名稱。
- c. **說明**—(可選)輸入說明新集合的文字。
- d. **路徑**—您可以在預設位置建立集合，也可以提供用來儲存集合的有效路徑。

2. 提供索引資訊

- a. **要建立索引的目錄**—輸入目錄名稱，以針對目錄中的文件建立索引至集合。僅能為在此虛擬伺服器中可見的目錄建立索引。
- b. **子目錄**—輸入將在集合中建立索引的文件子目錄。子目錄路徑應相對於之前所指定的目錄路徑。
- c. **式樣**—指定萬用字元以選取要建立索引的檔案。
謹慎使用萬用字元式樣，以確保僅為特定檔案建立索引。例如指定 `*.*` 甚至可能會為可執行檔和 Perl 程序檔建立索引。
- d. **子目錄**—已啟用/已停用。如果您選取此選項，還會為所選目錄之子目錄中的文件建立索引。這是預設動作。
- e. **預設文件編碼**—
集合中的文件不限於使用單一語言/編碼。每次增加文件時，只能指定單一編碼；但是，下次將文件增加至集合時，您可以選取其他預設編碼。

3. 步驟 3：檢視摘要

- a. 檢視摘要並按一下 **[完成] 按鈕**，以增加新的集合。

備註 – 使用 CLI

若要透過 CLI 增加搜尋集合，請執行以下指令。

```
wadm> create-search-collection --user=admin --password-file=admin.pwd  
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1 --uri=/search_config1  
--document-root=../docs searchcoll
```

請參閱 CLI 參照 `create-search-collection(1)`。

刪除搜尋集合

若要刪除搜尋集合，請執行以下作業：

1. 按一下 **[配置]** 標籤。
2. 從配置清單中選取配置。
3. 按一下 **[虛擬伺服器]** 標籤。
4. 從虛擬伺服器清單中選取虛擬伺服器。
5. 按一下 **[搜尋]** 標籤。
6. 在 **[搜尋集合]** 區段中，選取集合名稱並按一下 **[刪除]** 按鈕以刪除該集合。

備註 – 使用 CLI

若要透過 CLI 刪除搜尋集合，請執行以下指令。

```
wadm> delete-search-collection --user=admin --password-file=admin.pwd  
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1 searchcoll
```

請參閱 CLI 參照 `delete-search-collection(1)`。

排程集合更新

您可以排程定期對集合執行維護作業。可排定的作業包括重新建立索引和更新。管理介面用於為特定集合排程作業。您可以指定以下內容：

- 要執行的作業 (重新建立索引或更新)
- 執行工作的時間
- 在一星期中的哪一天執行工作

若要為集合進行事件排程，請執行以下作業：

1. 按一下 **[配置]** 標籤。
2. 從配置清單中選取配置。

- 3. 按一下 [虛擬伺服器] 標籤。
- 4. 從虛擬伺服器清單中選取虛擬伺服器。
- 5. 按一下 [搜尋] 標籤。
- 6. 按一下 [排定的事件] 標籤。
- 7. 在 [搜尋事件] 標籤下，按一下 [新增] 按鈕。

下表說明了 [新增搜尋事件排程] 頁面中的欄位：

表 12-1 欄位說明 > 新增搜尋事件排程

欄位	說明
集合	從下拉式清單中選取您要為其排定維護作業的集合。 1. 重新為集合建立索引 — 此排定的事件將在指定時間重新為指定的集合建立索引。 2. 更新集合 — 您可以在建立集合之後增加或移除檔案。只能在建立集合時指定的目錄中增加文件。如果要移除文件，只有這些檔案的項目及其中介資料會從集合中移除。實際檔案本身不會從檔案系統中移除。此排定的事件將在指定時間更新集合。 3. 式樣 — 指定萬用字元以選取要建立索引的檔案。 4. 包括子目錄 — 如果選取此選項，則也將為所選目錄的子目錄中的文件建立索引。這是預設動作。 5. 編碼 — 為要建立索引的文件指定字元編碼。預設為 ISO-8859-1。索引引擎嘗試根據嵌入式中介標記確定 HTML 文件的編碼。如果未指定，則使用預設編碼。
事件	配置的事件啟動時間。從下拉式方塊中選取小時和分鐘值。 每天 — 每天在指定的時間啟動指定的事件。 特定日期 — 在特定日期啟動指定的事件。 1. 日 — 指定星期日到星期六之間的任一天。 2. 日期 — 將某月的任一天 (從 1 日到 31 日) 指定為逗號分隔的項目，例如 4,23,9。
時間	特定月份 — 在特定時間和月份啟動指定的事件。指定 1 月至 12 月之間的月份。 在此時間段之後啟動指定的事件。 1. 間隔小時數 — 從下拉式方塊中選取小時數。
間隔	2. 間隔秒數 — 從下拉式方塊中選取秒數。

執行搜尋

使用者主要是在搜尋集合中查詢資料，然後取得作為查詢結果的文件清單。與 Sun Java System Web Server 一同安裝的搜尋 Web 應用程式可提供預設的搜尋查詢和搜尋結果頁面。這些頁面可以直接使用，或者如「[自訂搜尋頁面](#)」所述，使用一組 JSP 標籤來自訂。

使用者可以依伺服器管理員所建立的集合進行搜尋。他們可以執行以下作業：

- 輸入搜尋所需的一組關鍵字和可選查詢運算子
- 僅搜尋虛擬伺服器可見的集合
- 依單一集合搜尋，或者依虛擬伺服器可見的一組集合進行搜尋

伺服器管理員必須為使用者提供存取虛擬伺服器搜尋查詢頁面所需的 URL。

搜尋頁面

一般使用者可用於存取搜尋功能的預設 URL 為：

`http://<伺服器實例>:連接埠號/search`

範例：

`http://plaza:8080/search`

一般使用者呼叫此 URL 時，會啟動 [搜尋] 頁面 (一個 Java Web 應用程式)。

備註 – 如需有關進行基本搜尋和進階搜尋的詳細資訊 (包括關鍵字和可選查詢運算子的相關資訊)，請參閱搜尋引擎提供的線上說明。若要存取此資訊，請按一下 [搜尋] 頁面上的 [說明] 連結。

進行查詢

搜尋查詢頁面用於依集合進行搜尋。使用者輸入一組關鍵字和可選查詢運算子，然後會收到顯示在瀏覽器內網頁上的結果。結果頁面中包含伺服器上符合搜尋條件的文件連結。

備註 – 如「[自訂搜尋頁面](#)」中所述，伺服器管理員可以自訂此搜尋查詢頁面。。

若要進行查詢，請執行下列步驟：

▼ 進行查詢

- 1 在瀏覽器的位置列中輸入以下格式的搜尋 Web 應用程式 URL，即可存取該應用程式：
`http://<伺服器實例>:連接埠號/search`
 - 2 在顯示的搜尋查詢頁面中，核取表示要在 [搜尋] 欄位中搜尋的集合之核取方塊。
 - 3 鍵入幾個用於描述查詢的文字並按下 Enter 鍵 (或按一下 [搜尋] 按鈕)，以取得相關網頁的清單。
- 如需更為精確的搜尋，您可以使用 [進階搜尋] 頁面中提供的搜尋參數，如下節所述。

進階搜尋

使用者可以透過增加可精確化其關鍵字的運算子來增加搜尋的準確性。可從 [進階搜尋] 頁面選取這些選項。

若要進行進階搜尋查詢，請執行下列步驟：

▼ 進行進階搜尋查詢

- 1 在瀏覽器的位置列中輸入以下格式的搜尋 Web 應用程式 URL，即可存取該應用程式：
`http://<伺服器實例>:連接埠號/search`
- 2 按一下 [進階] 連結。
- 3 輸入以下任一或全部資訊：
 - [搜尋]。選取要搜尋的集合。
 - [尋找]。支援以下四個選項：
 - 所有文字。尋找包含 [尋找] 欄位中指定的所有關鍵字的頁面。
 - 任意文字。尋找包含 [尋找] 欄位中指定的任何關鍵字的頁面。
 - 精確片語。尋找符合 [尋找] 欄位中使用的精確片語的頁面。
 - 段落搜尋。在擷取到的頁面中反白顯示包含關鍵字或詞的段落。
 - 排除文字。搜尋將排除包含指定文字的網頁。
 - 「標題包含/不包含」。僅限搜尋標題中包含指定關鍵字的頁面。
 - [Since]。僅限搜尋其索引建於所選時段內的網頁。

文件欄位

Sun Java(TM) System Web Server 可維護文件的索引。該索引包含每個文件的項目。每個索引項目均包含一個或多個欄位，例如 Title、Author 及 URL。可將查詢限定至特定文件欄位，僅當文件符合指定欄位中的條件時，才可找到該文件。

例如，如果只是搜尋 Einstein，則只要 [Title]、[Author] 或 [Keywords] 欄位中有一個包含文字 Einstein，便會找到該文件。這樣，有關 Einstein 的文件、參考 Einstein 的文件以及由 Einstein 撰寫的文件都將包括在內。但是，如果您指定 Author = "Albert Einstein"，則會僅尋找由 Albert Einstein 撰寫的文件。

依預設，可以搜尋的索引欄位如下：

1. **作者**—建立文件的一個作者、多個作者或組織，以 <author> 中介標記指定。
2. **關鍵字**—以 <keywords> 中介標記指定的關鍵字。
3. **日期**—最後一次編輯或修改此文件的日期。
4. **標題**—文件的標題，以 HTML <title> 標記指定。

搜尋查詢運算子

如需有關搜尋查詢運算子的詳細清單，請參閱「[管理主控台搜尋線上說明](#)」。

檢視搜尋結果

搜尋結果將顯示在使用者瀏覽器中的 Web 頁面上，該頁面包含伺服器上符合搜尋條件的文件的 HTML 超連結。依預設，每個頁面顯示 10 條記錄 (符合項目)，這些記錄根據關聯依降序排序。每條記錄都會列出諸如檔案名稱、大小、建立日期等資訊。並且還會反白顯示相符的文字。

自訂搜尋頁面

Sun Java System Web Server 包含一個預設搜尋應用程式，可提供基本的搜尋查詢和搜尋結果頁面。可以直接使用這些網頁，也可以進行自訂以滿足特定需要。這種自訂可以像使用其他標誌重新設置網頁一樣簡單，也可以像變更搜尋結果顯示順序一樣複雜。

預設的搜尋應用程式提供 JSP 範例，說明如何使用搜尋標籤程式庫來建立自訂搜尋介面。您可以查看 `/bin/https/webapps/search` 中做為範例應用程式的預設搜尋應用程式，該應用程式說明可自訂搜尋標籤的用法。

預設的搜尋介面由以下四個主要的元件構成：頁首、頁尾、查詢表單和結果。

僅變更標籤的屬性值，即可輕鬆自訂這些基本元素。可以使用標籤程式庫完成更為詳細的自訂。

搜尋介面元件

搜尋介面包含以下元件：

標頭

標頭包括標誌、標題和簡短描述。

註腳

註腳包含版權資訊。

表單

查詢表單包含一組表示搜尋集合的核取方塊、查詢輸入方塊以及提交和說明按鈕。

結果

依預設，每頁列出 10 個結果記錄。對於每條記錄，將顯示標題、段落、大小、建立日期和 URL 等資訊。段落是頁面中反白顯示相符文字的一個短片段。

自訂搜尋查詢頁面

查詢表單包含搜尋集合的核取方塊清單、查詢輸入方塊和提交按鈕。此表單是使用 `<slws:form>` 標籤以及 `<collElem>`、`<queryBox>` 和 `<submitButton>` 標籤建立的，具有以下預設值：

```
<slws:form>
  <slws:collElem>
    <slws:queryBox> <slws:submitButton>
</slws:form>
```

可以將查詢表單放到頁面的任意位置：中間、側邊等。也可以用不同格式顯示查詢表單，例如使用一個橫框，其中集合選取方塊、查詢字串輸入方塊以及 [提交] 按鈕水平排列；或顯示為一個區段，其中集合顯示為核取方塊，查詢輸入方塊和 [提交] 按鈕置於下方。

以下範例顯示了如何使用 `<searchForm>` 標籤集來建立不同格式的查詢表單。

以水平框格式

以下範例程式碼將建立一個表單，其中包含所有集合的選取方塊、查詢輸入方塊和提交按鈕，均排在一列。

```
<slws:form>
  <table cellpadding="3" cellspacing="0" border="0">
    <tr class="navBar">
```

```

<td class="navBar"><slws:collElem type=" select" ></td>
<td class="navBar">
  <slws:querybox size="30">
    <slws:submitButton class="navBar" style="padding: 0px; margin: 0px; width: 50px">
  </td>
</tr>
</table>
</slws:form>

```

以提要欄位區段格式

您可以建立一個表單區段，其中表單元素排列在一個提要欄位中，表單區段的標題為「搜尋」，使用的格式與提要欄位中其他項目的格式相同。

表單元素位於提要欄位中的自訂查詢頁面

在下面給出的範例程式碼中，表單內文包含三個排成一行的核取方塊，其中列出可用的搜尋集合。查詢輸入方塊和 [提交] 按鈕置於下方：

```

<slws:searchForm>
  <table>
<!--... other sidebar items ... -->
    <tr class="Title"><td>Search</td></tr>
    <tr class="Body">
      <td>
        <table cellpadding="0" cellspacing="3" border="0">
          <tr class="formBlock">
            <td class="formBlock"> <slws:collElem type="checkbox" cols="1" values="1,0,1,0" /> </td>
          </tr>
          <tr class="formBlock">
            <td class="formBlock"> <slws:querybox size="15" maxlength="50"> </td>
          </tr>
          <tr class="formBlock">
            <td class="formBlock"> <slws:submitButton class="navBar" style="padding: 0px; margin: 0px; width: 50px">
          </tr>
        </table>
      </td>
    </tr>
  </table>
</slws:searchForm>

```

自訂搜尋結果頁面

搜尋結果按照以下方式產生：

- `<formAction>` 標籤擷取所有表單元素的值並進行基本驗證。
- `<search>` 標籤、`<resultIteration>` 標籤和其他標籤都存在於 `<formAction>` 標籤中，可存取所有表單元素的值。

- `<search>` 標籤透過 `<formAction>` 使用查詢字串和集合來執行搜尋，並將搜尋結果儲存在 `pageContext` 中。
- `<resultIteration>` 標籤隨後將擷取結果集並重複運算。

僅變更標籤的屬性值，即可自訂搜尋結果頁面。

以下程式碼範例以標題列開頭，隨後顯示指定數目的記錄，最後為瀏覽位址列。標題列包含搜尋中使用的查詢字串以及傳回的全部記錄範圍(例如 1 - 10)。對於每條記錄，記錄區段會顯示與檔案連結的標題、多達三個反白顯示關鍵字段落、URL、建立日期以及文件大小。

在該區段的末尾處，瀏覽位址列與上一頁和下一頁的連結，以及與目前頁之前和之後八個其他頁面的直接連結。

```
<slws:formAction />
<slws:formSubmission success="true" >
  <slws:search scope="page" />
  <!--search results-->
  (...html omitted...)
  <slws:resultStat formId="test" type="total" /></b> Results Found, Sorted by Relevance</span></td><td>
    <span class="body"><a href="/search/search.jsp?">Sort by Date</a></span></td>
    <td align="right"><span class="body">
      <slws:resultNav formId="test" type="previous" caption="" />
      &nbsp;&nbsp;&nbsp;<slws:resultStat formId="test" type="range" />
      &nbsp;&nbsp;&<slws:resultNav formId="test" type="next" caption="" />
      &nbsp;&nbsp;&
    (...html omitted...)
  <table border=0>
    <slws:resultIteration formId="test" start="1" results="15">
      <tr class=body>
        <td valign=top>
          <slws:item property='number' />.&nbsp;&nbsp;&&nbsp;&nbsp;&&nbsp;&nbsp;&&nbsp;&nbsp;&&nbsp;&nbsp;&&nbsp;&nbsp;&&nbsp;&nbsp;&&nbsp;&nbsp;&&
          </td>
          <td>
            <b><a href="<slws:item property='url' />"><slws:item property='title' /></a></b>
            <br>
            <slws:item property='passages' />
            <font color="#999999" size="-2">
              <slws:item property='url' /> -
              <slws:item property='date' /> -
              <slws:item property='size' /> KB
            </font><br><br>
          </td>
        </tr>
      </slws:resultIteration>
    </table>
    (...html omitted...)
    <slws:resultNav formId="test" type="previous" />
```



```

<slws:resultNav formId="test" type="full" offset="8" />
<slws:resultNav formId="test" type="next" />
(...html omitted...)
</slws:formSubmission>

```

自訂的搜尋結果頁面

透過處理標籤和修改 HTML，可以輕鬆自訂基本搜尋結果介面。例如，可以複製瀏覽位址列並將其放到搜尋結果前面。使用者也可以選擇顯示或不顯示搜尋記錄的任何特性。

除了與表單一起使用外，`<search>`、`<resultIterate>` 和相關標籤還可用於列出的特定主題。以下範例程式碼可列出某個網站上前十篇有關 Java Web 服務的文章：

```

<slws:search collection="Articles" query="Java Web Services" />
<table cellpadding="0" cellspacing="3" border="0">
  <tr class="Title"><td>Java Web Services</td></tr>
</table>
<table cellpadding="0" cellspacing="3" border="0">
<slws:resultIteration>
  <tr>
    <td><a href="<slws:item property='URL' />"> <slws:item property='Title' /></a></td>
  </tr>
</slws:resultIteration>
</table>

```

在單獨的頁面中自訂表單和結果

如果需要將表單頁面和結果頁面分開，您必須使用 `<form>` 標籤集建立表單頁面，並使用 `<formAction>` 標籤集建立結果頁面。

需要在結果頁面中增加與表單頁面的連結，以使頁面銜接順暢。

標籤慣例

請注意以下標籤慣例：

- 標籤類別屬於套裝模組 `com.sun.web.search.taglibs`。
- 所有 `pageContext` 屬性的前綴均為 `com.sun.web`。例如，搜尋結果的屬性為 `com.sun.web.searchresults.form_id`，其中 `form_id` 為表單名稱。
- 可以使用 `s1ws` 前綴來參照標籤程式庫。標籤的名稱及其屬性是大小寫混合的，其中每個內部單字的首字母大寫，例如 `pageContext`。

標籤規格

Sun Java System Web Server 包括一組 JSP 標籤，可用於自訂搜尋介面中的搜尋查詢和搜尋結果頁面。

如需可用於自訂搜尋頁面的完整 JSP 標籤清單，請參閱「Sun Java System Web Server 7.0 *Developer's Guide to Web Applications*」。

監視伺服器

本節介紹 Sun Java System Web Server 的監視功能，並針對可在實例和配置層級監視的伺服器參數，提供詳細的清單。

- [第 187 頁的「Sun Java System Web Server 中的監視功能」](#)
- [第 189 頁的「修改監視參數」](#)
- [第 191 頁的「配置 SNMP 子代理程式」](#)
- [第 195 頁的「設定伺服器的記錄」](#)
- [第 199 頁的「配置 Administration Server 的記錄設定」](#)

Sun Java System Web Server 中的監視功能

當您在 [監視父系] 標籤中選取 [配置] 或 [實例] 標籤時，將會顯示可以監視的伺服器參數。

您可在 Sun Java System Web Server Administration Console 執行以下動作：

- 檢視實例和配置層級的伺服器統計。
- 啟用/停用配置層級的監視。
- 檢視實例層級的錯誤/存取記錄。

若要在配置層級監視伺服器參數，按一下 [監視] > [配置] 標籤。表格列出可用配置及以下資訊：

- **節點** — 已在其中部署配置的節點數目。
- **請求** — 所有虛擬伺服器收到的請求總數。
- **錯誤** — 所有虛擬伺服器記錄的錯誤總數。
- **回應時間** — 任何虛擬伺服器的最長回應時間。

按一下配置名稱，可取得配置層級的統計。一般統計可分為 3 種類型：

- 請求統計
- 錯誤統計

- 回應時間統計

透過管理主控台監視

您可在管理主控台中檢視下列種類的伺服器統計資料：

- 一般統計資料。
- 實例統計資料。
- 虛擬伺服器統計資料。

表 13-1 監視種類

種類	說明
一般統計資料	[一般統計資料] 顯示了配置的所有請求、錯誤和回應統計資料。
實例統計資料	[實例統計資料] 顯示了配置的整體請求、錯誤和回應統計資料，以及有關伺服器當機和虛擬伺服器計數的資訊。
虛擬伺服器統計資料	[虛擬伺服器統計資料] 顯示了配置的所有請求、錯誤和回應統計資料，以及開啓之連線的數目和接收/傳輸的總位元組數。

▼ 檢視統計資料

- 1 按一下 [監視] 標籤。
- 2 從清單中選取配置。
- 3 檢視一般統計資料、實例統計資料和虛擬伺服器統計資料。

備註 – 使用 CLI

您可以使用 `get-config-stats`、`get-virtual-serevr-stats`、`get-webapp-stats` 和 `get-servlet-stats` 指令來監視伺服器。

- `wadm> get-config-stats --user=admin --password-file=admin.passwd --host=localhost --port=8989 --config=test --node=cat.test.com --ssl=true`

上述指令將擷取指定實例的統計資料。若要取得配置層級的統計資料，可以在使用上述指令時去掉 `--node` 選項。

- `wadm> get-vs-stats --user=admin --password-file=admin.passwd --host=localhost --port=8989 --config=test --vs=www.test.com --node=cat.test.com --ssl=true`

上述指令將擷取所有節點 (其中已部署配置) 中指定配置的虛擬伺服器彙總統計資料。若要取得給定節點上已部署配置的統計資料，可使用 `--node` 選項。

- `wadm> get-webapp-stats --user=admin --password-file=admin.passwd --host=localhost --port=8989 --config=test --node=cat.test.com --vs=www.test.com --uri=/foo --ssl=true`

上述指令將針對指定實例之指定虛擬伺服器上部署的 Web 應用程式，來擷取其統計資料。若要取得所有節點 (其中已部署配置) 中給定配置的 Web 應用程式彙總統計資料，可以在使用上述指令時去掉 `--node` 選項。

- `wadm> get-servlet-stats --user=admin --password-file=admin.pwd --host=localhost --port=8989 --config=test --node=cat.test.com --vs=www.test.com --uri=/servlet-simple --ssl=true`

上述指令將擷取 Servlet `servlet-simple` 的統計資料。

修改監視參數

伺服器透過 SNMP 執行監視動作。SNMP 是用於網路作業資料交換的協定。透過 SNMP，資料在受管理的裝置和網路管理工作站 (NMS) 之間傳送。受管理的裝置是指執行 SNMP 的任何裝置：主機、路由器、Web 伺服器以及網路上的其他伺服器。NMS 是用於在遠端管理該網路的機器。通常，NMS 軟體會提供圖形來顯示收集的資料，或使用該資料來確定伺服器的作業未超出特定偏差。

NMS 通常是指安裝了一個或多個網路管理應用程式且功能強大的工作站。網路管理應用程式 (例如 HP OpenView) 以圖形方式顯示有關受管理裝置 (例如您的 Web 伺服器) 的資訊。例如可以顯示企業中正在作業和已當機的伺服器，或所收到的錯誤訊息數目與類型。將 SNMP 與 Sun Java System Web Server 配合使用時，將使用兩種類型的代理程式 (子代理程式和主代理程式) 在 NMS 與伺服器之間傳輸此資訊。

子代理程式收集有關伺服器的資訊，並將該資訊傳送至伺服器的主代理程式。除 Administration Server 之外的每個 Sun Java System Web Server 都具有子代理程式。

備註 – 對 SNMP 配置進行任何變更後，您必須按一下 [儲存] 按鈕，然後重新啓動 SNMP 子代理程式。

若要變更配置的設定，請執行以下作業：

- 1. 按一下 [配置] 標籤。
- 2. 選取需要變更其監視設定的配置。
- 3. 按一下 [監視設定] 子標籤

配置監視參數

若要變更配置的一般監視設定，請編輯 [一般設定] 區段下的值。下表提供了一般監視參數的欄位說明：

表 13-2 欄位說明 > 一般監視設定

欄位	說明
SNMP 子代理程式	一般而言，若要使用SNMP，系統上必須安裝並執行主代理程式和最少一個子代理程式。您需要先安裝主代理程式，才能啓用子代理程式。 選取此選項以啓用/停用 SNMP 子代理程式。
間隔	輪詢間隔是指每次更新顯示的統計資訊所間隔的秒數。 如果伺服器實例正在執行且您已啓用統計資料，則您將看到一個頁面，其中顯示您選取的統計資料種類。該頁面每隔 5 至 15 秒更新一次，這取決於您選擇的輪詢間隔。
設定檔	您可以使用統計/設定檔功能來監視伺服器的目前作業。統計資料顯示伺服器正在處理的請求數目，以及處理這些請求的情況。您可以檢視個別虛擬伺服器的某些統計資料，並可以檢視整個伺服器實例的其他統計資料。 選取此選項以啓用/停用設定檔。

配置 SNMP 子代理程式參數

若要變更配置的 SNMP 子代理程式設定，請編輯 [SNMP 子代理程式設定] 區段下的值。下表提供了 SNMP 子代理程式參數的欄位說明：

表 13-3 欄位說明 > SNMP 子代理程式設定

欄位	說明
已啟用	一般而言，若要使用 SNMP，系統上必須安裝並執行主代理程式和最少一個子代理程式。您需要先安裝主代理程式，才能啟用子代理程式。 選取此選項以啟用/停用 SNMP 統計集合。
主要主機	輸入伺服器的名稱和網域 (僅 UNIX)。
說明	輸入伺服器的簡短說明，包括作業系統資訊。
組織	輸入代表組織的簡稱。
位置	在此欄位中輸入伺服器的位置資訊。
連絡人	在此欄位中輸入伺服器的連絡人資訊。

配置 SNMP 子代理程式

SNMP 是適用於網路活動資料交換的協定。透過 SNMP，資料在受管理的裝置和網路管理工作站 (NMS) 之間傳送。受管理的裝置是指執行 SNMP 的任何裝置：主機、路由器、Web 伺服器以及網路上的其他伺服器。NMS 是用於在遠端管理該網路的機器。通常，NMS 軟體會提供圖形來顯示收集的資料，或使用該資料來確定伺服器的作業未超出特定偏差。

NMS 通常是指安裝了一個或多個網路管理應用程式且功能強大的工作站。網路管理應用程式 (例如 Sun Management Center) 以圖形方式顯示有關受管理裝置 (例如您的 Web 伺服器) 的資訊。例如，它可以顯示您企業中工作和停用的伺服器，或收到的錯誤訊息數目與類型。將 SNMP 和 Sun Java System Web Server 搭配使用時，會透過兩種類型的代理程式 (子代理程式和主代理程式) 在 NMS 與伺服器之間傳輸此資訊。

子代理程式收集有關伺服器的資訊，並將該資訊傳送至伺服器的主代理程式。

如需啟動 SNMP 子代理程式，請執行以下作業：

1. 按一下 [節點] 標籤
2. 按一下節點清單中的可用節點。
3. 按一下 [SNMP 子代理程式] 標籤
4. 按一下 [啟動 SNMP 子代理程式] 按鈕以啟動子代理程式。

備註 – 啟動 SNMP 子代理程式之前，請驗證主代理程式是否正在執行。僅當主代理程式正在執行時，才會啟動子代理程式。

若要停止 SNMP 子代理程式，請執行以下作業：

- 1. 按一下 [節點] 標籤
- 2. 按一下節點清單中的可用節點。
- 3. 按一下 [SNMP 子代理程式] 標籤
- 4. 按一下 [停止 SNMP 子代理程式] 按鈕以停止子代理程式。

一般而言，若要使用SNMP，系統上必須安裝並執行主代理程式和最少一個子代理程式。您需要先安裝主代理程式，才能啓用子代理程式。

設定 SNMP 的程序因系統而異。下表提供了您將在不同情況下應遵循的程序簡介。本章的稍後部分將詳細說明實際程序。

開始之前，應該確認兩個事項：

- 您的系統是否已在執行 SNMP 代理程式 (作業系統的本機代理程式)？
- 如果正在執行，您的本機 SNMP 代理程式是否支援SMUX 通訊？(如果您使用的是 AIX 平台，則您的系統支援 SMUX。)

請參閱系統文件，以取得有關如何驗證此資訊的資訊。

備註 – 在 Administration Server 中變更 SNMP 設定之後，如果要安裝新的伺服器或刪除現有伺服器，則必須執行以下步驟：

- (Windows) 重新啓動 Windows SNMP 服務或重新啓動機器。
- (UNIX) 使用 Administration Server 重新啓動 SNMP 主代理程式。

表 13-4 一般準則

如果您的伺服器符合這些條件	請按照這些程序執行
<ul style="list-style-type: none">■ 目前尚未執行任何本機代理程式	<ul style="list-style-type: none">1. 啓動主代理程式。2. 為系統上安裝的每個伺服器啓用子代理程式。
<ul style="list-style-type: none">■ 目前正在執行本機代理程式■ 無 SMUX■ 無需繼續使用本機代理程式	<ul style="list-style-type: none">1. 為 Administration Server 安裝主代理程式時，請停止本機代理程式。2. 啓動主代理程式。3. 為系統上安裝的每個伺服器啓用子代理程式。
<ul style="list-style-type: none">■ 目前正在執行本機代理程式■ 無 SMUX■ 需要繼續使用本機代理程式	<ul style="list-style-type: none">1. 安裝 SNMP 代理程式。2. 啓動主代理程式。3. 啓動 SNMP 代理程式。4. 使用主代理程式連接埠號之外的連接埠號，重新啓動本機代理程式。5. 為系統上安裝的每個伺服器啓用子代理程式。
<ul style="list-style-type: none">■ 目前正在執行本機代理程式■ 支援 SMUX	<ul style="list-style-type: none">1. 重新配置本機 SNMP 代理程式。2. 為系統上安裝的每個伺服器啓用子代理程式。

使用 CLI 配置 SNMP

▼ 在 Solaris 上啟動 SNMP

1 配置 SNMP 參數。

設定配置的 SNMP 參數。

```
wadm> set-snmp-prop --user=admin --host=funland --port=1893
--config=test enabled=true master-host=masterhost-name organization=organization-name
location=location-name contact=contact-name description=description-name
```

2 部署配置。

```
wadm> deploy-config --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 config1
```

3 啟動伺服器實例。

```
$ ./https-test/bin/startserv
```

4 以超級使用者身份執行主代理程式 (magt)。

備註 – 若要執行 magt，必須先停止本機 snmpd。

```
$ cd /etc/init.d/
$ init.dmi stop; init.snmpdx stop; init.sma stop
```

移除檔案 https-admserv/config/logs/pid.masteragt (如果存在)。

```
$ rm ./https-admserv/config/logs/pid.masteragt
wadm> start-snmp-master-agent --snmp-port 161 hostname
```

5 啟動子代理程式。

移除檔案 https-admserv/config/logs/pid.httptagt (如果存在)。

```
$ rm ./https-admserv/config/logs/pid.httptagt
```

終止 httptagt (如果它已執行)。

```
wadm> start-snmp-subagent hostname
```

▼ 在 Linux 上啟動 SNMP

1 配置 SNMP 參數。

設定配置的 SNMP 參數。

```
wadm> set-snmp-prop --user=admin --host=funland --port=1893 --config=test
enabled=true master-host=masterhost-name organization=organization-name
location=location-name contact=contact-name description=description-name
```

2 部署配置。

```
wadm deploy-config --user=admin --password-file=admin.pwd
--host=serverhost --port=8989 config1
```

3 啟動伺服器實例。

```
$ ./https-test/bin/startserv
```

4 以超級使用者身份執行本機主代理程式 (snmpd)。

若要允許使用 snmpd 進行直接通訊，請在 /etc/snmp/snmpd.conf 中增加下一行，然後重新啟動 snmpd。

```
smuxpeer 1.3.6.1.4.1.42.2.190.1

view systemview included .1.3.6.1.4.1.42.2.190.1

# cd /etc/init.d/
# ./snmpd stop
# ./snmpd start
```

5 啟動子代理程式。

移除檔案 https-admserv/config/logs/pid.httpagt (如果存在)。

```
$ rm ./https-admserv/config/logs/pid.httpagt
```

強制結束 httpagt (如果它已執行)。

```
wadm> start-snmp-subagent hostname
```

▼ 在 Windows 上啟動 SNMP

1 配置 SNMP 參數。

設定配置的 SNMP 參數。

```
wadm> set-snmp-prop --user=admin --host=funland --port=1893 --config=test
enabled=true master-host=masterhost-name organization=organization-name
location=location-name contact=contact-name description=description-name
```

2 將 install-root/lib 目錄增加至系統路徑環境變數。

- 3 重新啟動機器。
- 4 使用 Windows 服務選項啟動 Web Server 實例。
- 5 啟動 SNMP 服務。

▼ 配置基於對等端的主代理程式 (magt)

藉由執行下列步驟，您可以配置基於對等端的主代理程式，以便與 Solaris 10 和 Linux 的作業系統本機主代理程式整合。

備註 – Solaris 10 作業系統本機主代理程式為 `snmpd`。依預設，本程式執行於 SNMP 預設的 UDP 連接埠 161 上。您可以使用 `/etc/sma/snmp/snmpd.conf` 檔案加以配置。還提供代理指令，用於將請求/回應轉寄至其他主代理程式或子代理程式。如需更多資訊，請參閱 `snmpd.conf` 線上手冊。

Solaris 8 和 9 沒有與作業系統本機主代理程式 `snmpd` 完全整合。對於 linux，`httpagt` 可直接與 `snmpd` 整合。在這種情況下，不需要執行 `magt`。對於 Windows，Sun Java System Web Server `snmp` 程式庫直接與 Windows SNMP 服務通訊。

- 1 啟動主代理程式，按上述備註所述指定 SNMP 連接埠 (11161)。

- 2 在 `/etc/sma/snmp/snmpd.conf` 中為 Solaris 10 增加下列內容。

```
proxy -v 1 -c public myserver:11161 .1.3.6.1.4.1.42.2.190.1
```

- 3 重新啟動 `snmpd`。

```
# cd /etc/init.d
# init.dmi stop; init.snmpdx stop; init.sma stop
# init.dmi start; init.snmpdx start; init.sma start
```

- 4 若要取得 SNMP 資料，請在連接埠上使用 `snmpwalk`：

```
$ snmpwalk -c public -v 1 <host-name>:<port> 1.3.6.1.4.1.42.2.190.1
```

設定伺服器的記錄

Administration Server 記錄檔記錄關於伺服器的資料，包括遇到的錯誤類型以及關於伺服器存取的資訊。檢視這些記錄可讓您監視伺服器作業，並透過提供資料 (如遇到的錯誤類型以及存取某些檔案的時間) 來排解問題。

您可以使用管理主控台中的 [記錄喜好設定] 頁面，指定記錄在管理伺服器記錄中的資料類型及格式。例如，您可以選擇記錄存取管理伺服器的每個用戶端的相關資料，也可以略過一些用戶端，不予記錄。此外，您還可以選擇常用記錄格式 (可提供固定數量的伺服器資訊)，或建立自訂記錄檔格式，以更好地滿足需求。

記錄類型

記錄類型大體可分為：

1. **存取記錄**—存取記錄可記錄有關向伺服器發出之請求以及伺服器之回應的資訊。
2. **伺服器記錄**—伺服器記錄可列出自建立記錄檔後，伺服器所遇到的所有錯誤。它也包含有關伺服器的資訊訊息，如啟動伺服器的時間與嘗試登入伺服器不成功的使用者。

檢視存取記錄及伺服器記錄

- 檢視伺服器記錄。

```
wadm> get-log --user=admin --password-file=admin.passwd --host=localhost  
--port=8989 --start-date=01/01/2006:09:00:00 --end-date=04/01/2006:10:00:00  
--config=test cat.test.com
```

上述指令將顯示 01/Jan/2006:09:00:00 與 04/Jan/2006:10:00:00 日期範圍內指定配置的伺服器記錄。

- 檢視存取記錄。

```
wadm> get-access-log --user=admin --password-file=admin.passwd  
--host=localhost --port=8989 --status-code=300 --config=test cat.test.com
```

上述指令僅會顯示狀態碼為 300 的指定配置存取記錄項目。

上述指令中的開始日期選項和結束日期選項應採用格式— dd/MM/yyyy:HH:mm:ss。您也可以自訂日期格式。您可以在 rcfile 中使用變數 wadm_log_date_format 指定自己的日期格式，而非使用預設日期格式。

配置記錄參數

若要啓用與編輯配置的記錄設定，請執行以下作業：

1. 按一下 **[配置]** 標籤
2. 選取您需要為其啓用/編輯記錄設定的配置。
3. 按一下 **[一般設定]** 標籤 > **[記錄設定]** 標籤

編輯存取記錄喜好設定

下表說明了 [存取記錄喜好設定] 區段中的欄位：

表 13-5 欄位說明 > 編輯存取記錄喜好設定

欄位	說明
存取記錄	已啟用/已停用。依預設，存取記錄已啟用。選取此選項可以停用存取記錄。請注意，啟用存取記錄將使伺服器效能有較小幅度的下降。
檔案位置	儲存存取記錄檔的伺服器路徑。預設值為 <code>../logs/access</code> 。
記錄格式	<div>1. 使用常用記錄格式—此選項為記錄檔的預設格式類型。伺服器將記錄從請求標頭中擷取的最相關資訊。 常用記錄格式為 <code>IP address – user [date] “request” status content-length</code>。</div> <div>2. 僅記錄以下詳細資訊—您可以使用此選項僅記錄請求標頭中的特定值。下列值可供核取：<ul style="list-style-type: none">■ 用戶端主機名稱■ 系統日期■ HTTP 狀態■ HTTP 標頭■ HTTP 方法■ 查詢字串■ 虛擬伺服器名稱■ 認證的使用者名稱■ 完整的 HTTP 請求■ 內容長度■ 請求 URI■ 協定</div>

編輯伺服器記錄喜好設定

下表說明 [伺服器記錄喜好設定] 區段中的欄位：

表 13-6 欄位說明 > 編輯伺服器記錄喜好設定

欄位	說明
伺服器記錄位置	儲存伺服器記錄檔的伺服器路徑。預設值為 <code>../logs/errors</code> 。
記錄詳細度層級	<div>此選項為您提供設定記錄詳細度的簡單方法。若要測試 Web 應用程式並對其進行除錯，建議使用 <i>finest</i> 層級。</div> <div>對於生產環境，建議的記錄層級為 <i>failure</i> 或 <i>security</i>。catastrophe 記錄層級將記錄非常少的詳細資訊。</div>
記錄虛擬伺服器名稱	如果選取此選項，則在記錄錯誤的同時，還會記錄處理請求的虛擬伺服器名稱。

表 13-6 欄位說明 > 編輯伺服器記錄喜好設定 (續)

記錄至系統記錄檔	將所有訊息記錄至系統記錄檔。
記錄至主控台	如果選取此選項，則會記錄由已部署 Web 應用程式引起的異常 (如果寫入 主控台 中)。 依預設，此選項為啓用狀態。
日期格式	時間戳記附加至錯誤訊息時所用的時間格式。預設值為 [%d/%b/%Y:%H:%M:%S] 。

歸檔記錄檔

您可以將記錄檔設定為自動歸檔。在某一時間，或在指定間隔後，伺服器將自動重建存取記錄。伺服器會儲存舊記錄檔，並使用包含其儲存日期和時間的名稱標示已儲存的檔案。

例如您可以將檔案設定為每小時自動重建一次，並且伺服器儲存檔案並將其命名為「access.199907.0152400」，其中，「**名稱|年|月|日|24 小時制時間**」會鏈結為單一字元串。依據您設定之記錄自動重建類型的不同，存取記錄歸檔檔案的確切格式也會不同。

存取記錄自動重建在伺服器啓動時初始化。如果開啓自動重建，則伺服器將建立具有時間戳記的存取記錄檔，而且伺服器啓動時就會開始自動重建。

開始自動重建後，當產生需要記錄到存取記錄檔的請求，而且產生時間在先前排定的「下一個自動重建時間」之後時，伺服器會建立具有新時間戳記的存取記錄檔。

設定記錄自動重建

您可以使用記錄自動重建選項排定已配置實例的錯誤/存取記錄自動重建。若要設定記錄自動重建，請執行以下步驟：

- 1. 按一下 **[配置] 標籤**
- 2. 選取您需要為其啓用/編輯記錄設定的配置。
- 3. 按一下 **[一般設定] 標籤 > [記錄設定] 標籤**
- 4. 按一下 **[記錄歸檔] 區段**下的 **[新增] 按鈕**

以下小節中說明了新增記錄自動重建頁面中的欄位：

表 13-7 欄位說明 > 設定記錄自動重建

欄位	說明
事件	存取記錄自動重建/伺服器記錄自動重建 。選取這兩個選項之一或同時選取二者，以為該記錄類型配置自動重建。

表 13-7 欄位說明 > 設定記錄自動重建 (續)

欄位	說明
時間	<p>配置的事件啟動時間。從下拉式方塊中選取小時和分鐘值。</p> <p>每天 — 每天在指定的時間啟動指定的事件。</p> <p>特定日期 — 在特定日期啟動指定的事件。</p> <p>1. 日 — 指定星期日到星期六之間的任一天。</p> <p>2. 日期 — 將某月的任一天 (從 1 日到 31 日) 指定為逗號分隔的項目，例如 4,23,9。</p> <p>特定月份 — 在特定時間和月份啟動指定的事件。指定 1 月至 12 月之間的月份。</p>
間隔	<p>在此時間段之後啟動指定的事件。</p> <p>1. 間隔小時數 — 從下拉式方塊中選取小時數。</p> <p>2. 間隔秒數 — 從下拉式方塊中選取秒數。</p>

如果您需要刪除排定的記錄自動重建，請按一下 [記錄歸檔] 區段中的 [刪除] 按鈕。

配置 Administration Server 的記錄設定

Administration Server 將記錄使用 Administration Console 執行的所有配置變更。所記錄的一些常見動作包括建立新配置、建立虛擬伺服器和配置實例設定。但是，配置層級詳細資訊 (如存取 Web 應用程式或在存取 Web 應用程式時發生的異常) 將由配置單獨記錄。

▼ 修改伺服器記錄位置

- 1 按一下 [管理伺服器] > [一般] 標籤。
- 2 移至 [記錄喜好設定] 區段。
- 3 編輯 [伺服器記錄位置] 欄位。

儲存錯誤的記錄位置。提供用於維護記錄檔的有效伺服器路徑。還檢查 Administration Server 是否具有在 UNIX 系統之指定目錄中寫入的權限。

預設位置為 ../log/errors

▼ 修改記錄詳細度級別

- 1 按一下 [管理伺服器] > [一般] 標籤。
- 2 移至 [記錄喜好設定] 區段。
- 3 選取 [記錄詳細度級別]。

此選項為您提供設定記錄詳細度的簡單方法。對於測試和除錯，建議的級別為 *finest*。

對於生產環境，建議的記錄級別為 *failure* 或 *security*。 *catastrophe* 記錄層級將記錄非常少的詳細資訊。

▼ 修改記錄的日期格式

- 1 按一下 [管理伺服器] > [一般] 標籤。
- 2 移至 [記錄喜好設定] 區段。
- 3 編輯 [日期格式] 欄位。

用來將時間戳記附加至錯誤訊息的時間格式。預設值為 [%d/%b/%Y:%H:%M:%S]

國際化和本土化

Sun Java System Web Server 的國際化和本土化版本支援多種語言和多種編碼。

- 第 201 頁的「輸入多位元組資料」
- 第 202 頁的「支援多字元編碼」
- 第 202 頁的「配置伺服器以提供本土化內容」

輸入多位元組資料

如果您想要在管理主控台頁面上輸入多位元組資料，則需要注意以下問題：

檔案名稱或目錄名稱

如果要在 URL 中顯示檔案名稱或目錄名稱，則名稱中不能包含 8 位元字元或多位元組字元。

LDAP 使用者與群組

對於電子郵件位址，請僅使用 RFC 17.000 (<ftp://ds.internic.net/rfc/rfc17.000.txt>) 中允許的那些字元。使用者 ID 與密碼資訊必須以 ASCII 格式儲存。

若要確保以正確的格式為使用者和群組輸入字元，請使用支援 UTF-8 格式的用戶端輸入 8 位元資料或多位元組資料。

支援多字元編碼

Sun Java System Web Server 7.0 為下列功能提供多字元編碼支援：

- [第 202 頁的「WebDAV」](#)
- [第 202 頁的「搜尋」](#)

WebDAV

Sun Java System Web Server 支援使用 PROPPATCH 和 PROPFIND 方法設定與擷取多位元組特性。雖然請求可以使用任何編碼格式，但伺服器的回應永遠使用 UTF-8。

搜尋

Sun Java System Web Server 7.0 使用基於 Java 的搜尋引擎，此搜尋引擎支援使用基礎 Java VM 支援的所有字元編碼對文件進行完整文字索引和搜尋。在建立搜尋集合時，可以指定文件的預設編碼。對於 HTML 文件，索引子將嘗試從 HTML 中介標記推算編碼，如果失敗，索引子將重新使用預設編碼。

搜尋介面基於 JSP 標籤程式庫，您可以使用需要的任何語言和編碼對其進行自訂和本土化。「Sun Java System Web Server 7.0 Developer's Guide to Web Applications」中列出了標籤程式庫。

配置伺服器以提供本土化內容

一般使用者可以配置他們的瀏覽器，以傳送 Accept-language 標頭 (描述他們所存取內容的語言喜好設定)。啟用 [配置] > (選取配置) > [虛擬伺服器] > (選取虛擬伺服器) > [伺服器設定] > [一般] > [本土化] 下的 [協商用戶端語言] 核取方塊，即可將伺服器配置為根據 Accept-language 標頭處理內容。

例如，如果此選項已啟用，並且用戶端傳送包含值 fr-CH,de 的 Accept-language 標頭，此時請求以下 URL：

`http://www.someplace.com/somepage.html`

伺服器會依下列順序搜尋檔案：

▼ 搜尋順序

- 1 Accept-language 清單 fr-CH,de。

`http://www.someplace.com/fr_ch/somepage.html`

`http://www.someplace.com/somepage_fr_ch.html`

`http://www.someplace.com/de/somepage.html`

`http://www.someplace.com/somepage_de.html`

- 2 不包含國家/地區代碼的語言代碼(fr 對應 fr-CH)：

`http://www.someplace.com/fr/somepage.html`

`http://www.someplace.com/somepage_fr.html`

- 3 DefaultLanguage (例如 en) 是在 `magnus.conf` 檔案中定義的。

`http://www.someplace.com/en/somepage.html`

`http://www.someplace.com/somepage_en.html`

- 4 如果在上述 URL 中找不到，則伺服器會嘗試：

`http://www.someplace.com/somepage.html`

備註 – 請記住，對本土化檔案進行命名時，會將諸如 CH 和 TW 的國家/地區代碼轉換為小寫，將破折號 (-) 轉換為底線 (_)。



注意 – 由於伺服器必須依上面闡明的演算法檢查 `Accept-language` 中指定的每種語言的內容，因此啟用 `acceptlanguage` 設定會導致效能損失。

CLI 自先前版本後所經歷的變更

下表描述一些可使用 Sun Java System Web Server 7.0 及舊版執行的常用工作。

表 A-1 CLI 自先前版本後所經歷的變更

作業	6.1 CLI	7.0 CLI
列出實例上所有已部署的 Web 應用程式。	wdeploy list -i INSTANCE_NAME -v VIRTUAL_SERVER	wadm> list-webapps --user=admin --port=8888 --password-file=admin.passwd --no-ssl
部署新的 Web 應用程式。	wdeploy deploy -i INSTANCE_NAME -v VIRTUAL_SERVER -u URI_PATH war file name	1. wadm> add-webapp --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME --vs=VIRTUAL_SERVER --uri=URI_PATH war file name 2. wadm> deploy-config --user=admin --port=8888 --password-file=admin.passwd 'HOSTNAME'
重新配置執行中的實例。	不支援。	wadm> reconfig-instance --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME
列出實例的所有虛擬伺服器。	HttpServerAdmin list -v -d INSTALL_DIR -sinst https-INSTANCE_NAME	wadm> list-virtual-servers --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME

表 A-1 CLI 自先前版本後所經歷的變更 (續)

作業	6.1 CLI	7.0 CLI
列出所有 JDBC 資源。	<code>HttpServerAdmin list -r -jdbc -d INSTALL_DIR -sintance https-INSTANCE_NAME</code>	<code>wadm> list-jdbc-resources --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME</code>
建立自訂資源。	<code>HttpServerAdmin create -r -custom -jndiname -poolname -enabled true</code>	<code>wadm> create-custom-resource --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME --res-type=type --jndi-name NAME</code>
啟動實例。	不支援。	<code>wadm> start-instance --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME NODENAME*</code>
停止實例。	不支援。	<code>wadm> stop-instance --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME NODENAME*</code>
使用反向代理伺服器配置 Web 伺服器。	不支援。	<div>1. <code>wadm> create-reverse-proxy --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME --vs='VIRTUAL_SERVER' --from='URI' --server='target-hostname'</code></div> <div>2. <code>wadm> set-reverse-proxy-prop --user=admin --password-file=admin.pwd --host=serverhost --port=8888 --config=config1 --vs=config1_vs_1 --uri-prefix=/test/ --server=http://java.com:8080 --sticky-cookie=testCookie</code></div>
停用反向代理伺服器。	不支援。	<code>wadm> delete-reverse-proxy --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME --vs='VIRTUAL_SERVER' --uri-prefix='URI'</code>

表 A-1 CLI 自先前版本後所經歷的變更 (續)

作業	6.1 CLI	7.0 CLI
啓用 WebDAV。	不支援。	<div>1. wadm> enable-webdav --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME</div> <div>2. wadm> deploy-config --user=admin --port=8888 --password-file=admin.passwd HOSTNAME</div>
建立新的 Web 伺服器。	不支援。	<div>1. wadm> create-config --doc-root=[DOCR00T] -http-port=[HTTPPORT] --jdk-home=[JAVAHOME] --server-user=[SERVERUSER] --server-name=[HOSTNAME] CONFIGNAME</div> <div>2. wadm> create-instance --config=CONFIGNAME NODENAME</div> <div>3. wadm> deploy-config CONFIGNAME</div>

FastCGI 外掛程式

- 第 209 頁的「簡介」
- 第 210 頁的「外掛程式函數 (SAF)」
- 第 213 頁的「配置 Web Server 的 FastCGI 外掛程式」
- 第 219 頁的「FastCGI 應用程式範例」

簡介

FastCGI 為現有 CGI (共用閘道介面) 的增強版本，CGI 是外部應用程式與 Web 伺服器之間的介面標準。與 CGI 一樣，FastCGI 應用程式以獨立的隔離程序執行。使用 FastCGI 的一些優勢如下：

- 使應用程式能夠在用戶端請求之間能持續、免除+H204應用程式啟動的經常性耗用時間，並允許應用程式在用戶端呼叫之間維持狀態。
- 使應用程式能夠常駐在遠端系統 (與執行 Web 伺服器的系統不同) 上。
- 讓應用程式功能更靈活，明確支援執行用戶端認證的應用程式和支援輸入篩選功能。
- 讓管理員限制 FastCGI 伺服器對系統造成的影響。

FastCGI 外掛程式可讓 Web 伺服器以可延伸的方式，安全地與常用協力廠商動態內容產生技術 (例如 Perl 和 Python) 搭配作業。

如需有關 FastCGI 的更多資訊，請參閱 <http://www.fastcgi.com/devkit/doc/fcgi-spec.html> 上的規格。

外掛程式函數 (SAF)

FastCGI 外掛程式提供了下列伺服器應用程式函數 (SAF)：

以下小節說明 FastCGI SAF 的各種參數和「error-reason」字串：

- 第 210 頁的「auth-fastcgi」
- 第 210 頁的「responder-fastcgi」
- 第 211 頁的「filter-fastcgi」
- 第 211 頁的「error-fastcgi」
- 第 211 頁的「FastCGI SAF 參數」
- 第 213 頁的「error-fastcgi SAF 錯誤原因字串」

auth-fastcgi

auth-fastcgi 是一個 PatchCheck 函數。此函數用於將請求轉寄至「Authorizer」FastCGI 應用程式。如果授權成功，將傳送回覆碼 200。否則，會將來自「Authorizer」FastCGI 應用程式的回應傳送回使用者代理程式。

您可以在 <http://www.fastcgi.com/devkit/doc/fcgi-spec.html#S6> 找到有關 FastCGI 角色的更多資訊。

auth-fastcgi SAF 可接受的參數位於：[第 211 頁的「FastCGI SAF 參數」](#)。

以下 obj.conf 程式碼範例說明 auth-fastcgi 的用法：

```
PathCheck fn="auth-fastcgi" app-path="/usr/bin/perl"  
app-args="/fastcgi/apps/auth/SimpleAuth.pl" bind-path="localhost:3432"。
```

responder-fastcgi

responder-fastcgi 是一個 Service 函數。此函數用於將請求轉寄至做為「Responder」的 FastCGI 應用程式。來自 Responder 應用程式的回應將傳送回使用者代理程式。可以在 <http://www.fastcgi.com/devkit/doc/fcgi-spec.html#S6> 找到有關 FastCGI 角色的更多資訊。

responder-fastcgi SAF 可接受的參數清單位於：[第 211 頁的「FastCGI SAF 參數」](#)。

以下 obj.conf 程式碼範例說明 responder-fastcgi 的用法：

```
Service fn="responder-fastcgi"  
app-path="/fastcgi-enabled-php-installation/bin/php"  
bind-path="localhost:3433" app-env="PHP_FCGI_CHILDREN=8"  
app-env="PHP_FCGI_MAX_REQUEST=500"。
```

filter-fastcgi

`filter-fastcgi` 是一個 Service 函數。此函數用於將請求轉寄至「Filter」類型的 FastCGI 應用程式。「Filter」應用程式可接收與 HTTP 請求關聯的資訊，還可以接收儲存在伺服器上的檔案中資料。然後，「Filter」應用程式產生「已篩選」版本的資料流做為回應。此回應將被傳送回使用者代理程式。可以在 <http://www.fastcgi.com/devkit/doc/fcgi-spec.html#S6> 找到有關 FastCGI 角色的更多資訊。

`filter-fastcgi` SAF 可接受的參數清單位於：[第 211 頁的「FastCGI SAF 參數」](#)。

以下 `obj.conf` 程式碼範例說明 `filter-fastcgi` 的用法：

```
Service fn="filter-fastcgi" app-path="/fastcgi/apps/filter/SimpleFilter"
bind-path="localhost:3434"
app-env="LD_LIBRARY_PATH=/fastcgi/fcgi-2.4/libfcgi/.libs" min-procs=2
```

error-fastcgi

`error-fastcgi` 是 Error 函數。`error-fastcgi` SAF 可處理 FastCGI 外掛程式特有的錯誤。但是，此函數並不處理 HTTP 錯誤。可將 FastCGI 外掛程式配置為發生錯誤時顯示特定頁面，或將請求重新導向至特定 URL。

`error-fastcgi` SAF 可接受的參數清單位於：[第 211 頁的「FastCGI SAF 參數」](#)。

以下 `obj.conf` 程式碼片段示範 `error-fastcgi` 的用法：

```
Error fn="error-fastcgi" error-reason="Invalid Parameters"
error-url="http://www.foo.com/errorPage.html"
```

請參閱[第 211 頁的「FastCGI SAF 參數」](#)，以取得有關 `error-fastcgi` 參數的資訊。

FastCGI SAF 參數

除非另有明確說明，否則 FastCGI 外掛程式 SAF「`auth-fastcgi`」、「`responder-fastcgi`」和「`filter-fastcgi`」均可接受下列參數：

請注意，參數 `chroot`、`user`、`group` 和 `nice` 僅適用於 UNIX 平台。這些參數在 Windows 平台上會被忽略。

- `app-path` - (可選擇) 處理請求的 FastCGI 應用程式路徑。此功能取決於 `bind-path` 參數的值，如下所示：
 1. 如果僅指定 `app-path`，則外掛程式將建立 FastCGI 應用程式，用於偵聽由外掛程式建立的 UNIX 網域通訊端。但是，僅 UNIX 平台接受此參數。在 Windows 上，會記錄一則錯誤訊息。

2. 如果同時指定 `app-path` 和 `bind-path`，則外掛程式將啟動指定的 FastCGI 應用程式程序，並將其連結至指定的 `bind-path`。
 3. 如果僅指定 `bind-path`，則會將 FastCGI 應用程式視為遠端執行。因此，外掛程式將不會啟動 FastCGI 應用程式程序。
 4. 如果「`app-path`」和「`bind-path`」均未指定，則外掛程式將記錄一則錯誤訊息。
- `app-args` — (可選擇) 當成引數傳送至 FastCGI 應用程式程序的值。您可使用多個 `app-args` 參數。使用多個 `app-args` 參數的格式為 `app-args="value" app-args="value" ..`。
 - `bind-path` - (可選擇) 可以是 Unix 網域通訊端名稱或「`host:port`」形式。「`app-path`」參數的說明解釋了「`bind-path`」參數的用法。請注意，Unix 網域通訊端名稱僅適用於 UNIX 平台。在 Windows 平台上，必須將 `bind-path` 指定為「`host:port`」。
 - `min-procs` - (可選擇) 用於指定要建立的 FastCGI 應用程式程序最小整數值。預設為 1。
 - `max-procs` - (可選擇) 用於指定可建立的 FastCGI 應用程式程序最大整數值。此整數值必須等於或大於 `min-procs` 值。預設為 1。
 - `chroot` - (可選擇) 用於設定 `chroot` FastCGI 伺服器應用程式程序的根目錄。預設為 Web Server 的根目錄。
 - `user` - (可選擇) 指定 FastCGI 應用程式執行時所用之身份的使用者 ID。預設為 Web Server 的使用者 ID。
 - `group` - (可選擇) FastCGI 應用程式將在指定的群組下執行。預設為 Web Server 的群組。
 - `nice` - (可選擇) 指定 FastCGI 應用程式程序的 `nice/priority` 值。
 - `listen-queue` - (可選擇) 用於指定通訊端的偵聽佇列大小整數值。此參數的預設值為 256。
 - `app-env` - (可選擇) 當成環境變數傳送至 FastCGI 應用程式程序的成對值。您可使用多個「`app-env`」參數。使用多個 `app-env` 參數的格式為 `app-env="name=value" app-env="name=value"....`。
 - `reuse-connection` - (可選擇) 一個布林值，用於確定是否重複使用與 FastCGI 應用程式的連線。`False (0、false、no)` 代表每次處理請求後需+H242關閉與 FastCGI 應用程式的連線。`+H248True (1、true、yes)` 代表新請求可重複使用現有連線。預設為 `false`。另請參閱 `connection-timeout`。
 - `connection-timeout` - (可選擇) 如果將「`reuse-connection`」設定為 `True`，則此值可指定池儲存的連線的逾時值 (以秒為單位)。如果連線閒置長達指定的時間長度，外掛程式將關閉此連線。此參數的預設值為 5 秒。另請參閱 `reuse-connection`。
 - `resp-timeout` - (可選擇) 代表 FastCGI 伺服器回應逾時的整數值 (以秒為單位)。如果在指定的時間內沒有來自 FastCGI 應用程式的回應，將捨棄此請求。此參數的預設值為 5 分鐘。

- `restart-interval` - (可選擇) 值為一個整數，代表重新啟動 FastCGI 應用程式的時間間隔 (以分鐘為單位)。此參數的預設值為 60 分鐘 (1 小時)。如果將此參數的值設定為零，將不會強制 FastCGI 應用程式重新啟動。
- `req-retry` - (可選擇) 值為一個整數，代表 FastCGI 應用程式拒絕請求時，外掛程式應重新傳送請求的次數。此參數的預設值為零。

`error-fastcgi` 伺服器應用程式功能 (SAF) 接受下列參數：

- `error-url` - 指定在發生失敗或錯誤情況時，要顯示的頁面、URI 或 URL。此參數的值可以是絕對路徑、文件根的相對路徑、URL 或 URI。
- `error-reason` - (可選擇) 代表 FastCGI 協定錯誤的字串。發生任何外掛程式錯誤時，此字串可區分要顯示的錯誤 URL。

error-fastcgi SAF 錯誤原因字串

本節提供所有有效「error-reason」字串的清單及這些字串的說明：

- 「缺少配置參數或配置參數無效」：未指定 `app-path` 和 `bind-path` 時。
- 「Stub 啟動錯誤」：無法啟動 Fastcgisub 程序。
- 「Stub 連線失敗」：無法連線至 Fastcgistub。
- 「無權限」：FastCGI 應用程式或 Fastcgisub 沒有執行權限。
- 「Stub 請求處理錯誤」：無法將請求傳送至 stub、接收到的請求無效，或 stub 沒有針對請求傳送回應等。
- 「設定參數失敗」：設定 `user`、`group`、`chroot`、`nice` 等失敗。
- 「使用者和/或群組無效」：使用者或群組無效。
- 「伺服器程序建立失敗」：FastCGI 應用程式執行失敗或 FastCGI 應用程式無法連結至指定的位址。
- 「Fastcgi 協定錯誤」：FastCGI 應用程式包含具有無效 FastCGI 版本或角色的標頭。
- 「內部錯誤」：無法開啓要傳送至篩選應用程序的檔案，或任何其他不明的錯誤。

配置 Web Server 的 FastCGI 外掛程式

FastCGI 外掛程式隨附於 Web Server 7.0。外掛程式安裝在以下位置：

32 位元的 FastCGI 外掛程式二進位檔安裝在 `<install_dir>/plugins/fastcgi` 目錄下。
64 位元的 Solaris SPARC FastCGI 外掛程式二進位檔安裝在
`<install_dir>/lib/plugins/fastcgi/64` 目錄下。

將安裝下列 FastCGI 二進位檔：

libfastcgi.so (適用於 Solaris/Linux)
fastcgi.dll(適用於 Windows)
Fastcgistub.exe(適用於 Windows)
libfastcgi.sl(適用於 HP-UX)
Fastcgistub(可執行檔)

配置 FastCGI 外掛程式時，需使用位於 <instance-dir>/config 目錄下的 Web Server 配置檔案。若要配置 FastCGI 外掛程式，請執行下列步驟：

- [第 214 頁的「修改 magnus.conf」](#)
- [第 214 頁的「修改 MIME 類型 \(可選擇\)」](#)
- [第 215 頁的「修改 obj.conf」](#)
- [第 216 頁的「FastCGI 外掛程式疑難排解」](#)
- [第 217 頁的「開發 FastCGI 應用程式」](#)

修改 magnus.conf

使用「load-modules」Init 函數載入 FastCGI 外掛程式共用程式庫。

```
Init fn=flex-init access="access" format.access="%Ses->client.ip%  
- %Req->vars.auth-user% [%SYSDATE%] \"%Req->reqpb.clf-request%\"  
%Req->srvhdrs.clf-status% %Req->srvhdrs.content-length%
```

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

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

修改 MIME 類型 (可選擇)

編輯 mime.types 檔案以指定 MIME 對映。修改 MIME 類型對映是可選步驟。

例如，

```
#--Sun Microsystems Inc. MIME 資訊  
  
# 請勿刪除上一行。該行用於識別檔案類型。  
  
#  
  
# Copyright 2006 Sun Microsystems, Inc. 版權所有。  
  
# 使用應遵守授權合約的條款。  
  
#
```

```

type=application/octet-stream exts=bin

type=application/astound exts=asd,asn

...

...

type=magnus-internal/fastcgi exts=php

...

...

```

修改 obj.conf

編輯 obj.conf 檔案，以使用之前各節所述的外掛程式 SAF，來配置 FastCGI 特有的請求。

修改後的 obj.conf 檔案範例如下所示：

```

#

# Copyright 2006 Sun Microsystems, Inc. 版權所有。

# 使用應遵守授權合約的條款。

#

# 您可以編輯此檔案，但如果使用管理 GUI 或 CLI，
# 註釋和格式變更可能會遺失。


<object name = "default">

    AuthTrans fn="match-browser" browser="*MSIE*"
                ssl-unclean-shutdown="true"
    NameTrans fn="ntrans-Java EE" name="Java EE"
    NameTrans fn="pfx2dir" from="/mc-icons"
                dir="/ws7/lib/icons" name="es-internal"
    NameTrans fn="assign-name" from="/fcgi/*" name="fcgi.config"

</object>

```

```
<Object name="fcgi.config">

    AuthTrans fn="auth-fastcgi" app-path="/fastcgi/apps/c/simpleAuth"
              bind-path="localhost:2111"
    Service fn="responder-fastcgi"
            app-path="/fastcgi_enabled_php_installation_dir/bin/php"
            app-env="name1=abc"

</object>
...
```

請注意，針對不同的 URL 式樣定義不同的物件或將 SAF 對映至不同的 MIME 類型，便可以不同的方式呼叫 FastCGI SAF。

如需有關 obj.conf 配置及語法的更多資訊，請參閱「*Administration Configuration File Reference Guide*」。

FastCGI 外掛程式疑難排解

Fastcgistub 是一個程序管理程式，用於管理 FastCGI 應用程式程序的生命週期。Fastcgistub 將其訊息記錄至 Web Server 暫存目錄下的 Fastcgistub.log 檔案中。發生任何錯誤時，檢查此檔案有助於消除問題。

問題：FastCGI 請求未得到處理

可能的原因及解決方案如下：

1. 請檢查是否載入 FastCGI 外掛程式。如果在啟動 Web Server 期間出現下列訊息，則表示已成功載入外掛程式。另外，請檢查 magnus.conf 中外掛程式庫的路徑：
FCGI1000: Sun Java System Web Server 7.0 FastCGI NSAPI Plugin < build info>
2. 請檢查在 obj.conf 中是否正確指定請求對映。如需有關 obj.conf 檔案的更多資訊，請參閱「*Sun Java System Web Server Administrator's Configuration Reference File*」。
3. 請檢查錯誤記錄中的任何可能錯誤訊息。
4. 請檢查 stub 二進位檔及 FastCGI 應用程式的權限。如果未授予足夠的權限，則外掛程式將無法啟動 stub 或應用程式。有關
5. 請檢查 Fastcgistub.log 檔案中有關 stub 端的任何可能錯誤。
6. 如果可能，請以獨立模式執行 FastCGI 應用程式，檢查該應用程式是否能夠正確無誤地執行。

如果丟出任何程式庫相依性錯誤，請在 obj.conf 中將 LD_LIBRARY_PATH 指定為 app-env 參數，且參數具有 LD_LIBRARY_PATH=<>相依性程式庫路徑值。

問題：FastCGI 應用程式未啟動。

可能的原因及解決方案如下：

請檢查 `Fastcgistub.log` 檔案中的下列記錄訊息：

```
..
<pid> process startup failure, trying to restart
...
Even after trying <n> time(s), <application path> process failed to start...no more retries
```

啟動失敗的原因之一可能是由於無法載入相依性程式庫。此問題可透過以下方式解決：將適當的程式庫路徑當成 `app-env` 參數值，指定給 `obj.conf` 檔案中配置的 FastCGI 應用程式。例如：

```
Service fn="responder_fastcgi" app-path="/fastcgi/c/tux-app" bind-path="localhost:2112"
app-env="LD_LIBRARY_PATH=/tuxedo/lib"
```

開發 FastCGI 應用程式

您可以使用 Perl、PHP、C 和 Java 開發 FastCGI 應用程式。下列小節簡要說明使用一些常用程式設計語言開發應用程式的程序。

- [第 217 頁的「執行 FastCGI 應用程式」](#)
- [第 217 頁的「FastCGI 應用程式結構」](#)
- [第 218 頁的「使用 Perl」](#)
- [第 218 頁的「使用 PHP」](#)
- [第 218 頁的「使用 C/Java」](#)

▼ 執行 FastCGI 應用程式

- 1 停止 Web Server。
- 2 重新啟動 Web Server。
- 3 存取以「`fcgi`」做為應用程式根的應用程式。
例如：`http://localhost/fcgi/ListDir.php`

FastCGI 應用程式結構

一般 FastCGI 應用程式具有以下程式碼結構：

```
Initialization code

Start of response loop
    body of response loop
End of response loop
```

初始化程式碼僅在應用程式初始化時執行一次。初始化程式碼通常執行一些耗時作業，例如開啓資料庫或者計算表格或點陣圖的值。將 CGI 程式轉換為 FastCGI 程式的主要作業，是區分初始化程式碼與針對每個請求都需要執行的程式碼。

回應迴圈會持續執行，等待用戶端請求到達。迴圈以呼叫 `FCGI_Accept` (FastCGI 程式庫中的常式) 開頭。`FCGI_Accept` 常式將封鎖程式執行，直到用戶端請求 FastCGI 應用程式為止。收到用戶端請求時，`FCGI_Accept` 將解除封鎖、執行回應迴圈主體的一次重複運算，之後再次封鎖，等待另一個用戶端請求。僅當系統管理員或 Web Server 終止 FastCGI 應用程式時，該迴圈才會終止。

使用 Perl

從 CPAN 下載並安裝最新的 FCGI 模組。以 ActivePerl 而言，您可以從 <http://aspn.activestate.com/ASPN/Downloads/ActivePerl/PPM/Zips> 下載模組。

如需有關使用 Perl 撰寫 FastCGI 應用程式的更多資訊，請參閱 <http://www.fastcgi.com/devkit/fastcgi-prog-guide/ch3perl.htm#3659>

使用 PHP

從 PHP 4.3.0 開始，FastCGI 成為 PHP 引擎支援的配置。若要編譯支援 FastCGI 之 PHP 4.3.x 或更高版本的引擎，請將配置切換 `--enable-fastcgi` 納入建立程序，例如

```
./configure <other-options> --enable-fastcgi  
gmake
```

編譯完成後，php 二進位檔將啓用 FastCGI。

當使用 PHP 版本 5.1.2 或舊版 (包括 PHP 4.x) 時，應使用 `host:port` 格式配置 FastCGI 外掛程式。例如，`bind-path = "localhost:3333"`。

對於 PHP 版本 5.1.3 及更高版本，`bind-path` 是可選擇的。如果已指定，則不應使用「`host:port`」格式。它可以是一個字串。例如，`bind-path = "myphpbindpath"`。

使用 C/Java

FastCGI 開發工具組提供撰寫 FastCGI C/Java 應用程式的 API。您可以從 <http://www.fastcgi.com/devkit/doc/fcgi-devel-kit.htm> 下載此工具組。

若要建立下載的 FastCGI 開發套件，請執行下列步驟：

1. 解壓縮 tar 檔案。此動作將建立名為 `fcgi-devel-kit` 的新目錄
2. 在 `fcgi-devel-kit` 目錄中執行此指令序列：
 - a. `./configure`
 - b. `make`

如需有關使用 C 撰寫 FastCGI 應用程式的更多資訊，請參閱 <http://www.fastcgi.com/devkit/doc/fcgi-devel-kit.htm#S3>

如需有關使用 Java 撰寫 FastCGI 應用程式的更多資訊，請參閱
<http://www.fastcgi.com/devkit/doc/fcgi-java.htm>

FastCGI 應用程式範例

本節包含使用 PHP、Perl 和 C 撰寫的 FastCGI 應用程式範例。

- 第 219 頁的「使用 PHP 撰寫的回應程式應用程式 (ListDir.php)」
- 第 219 頁的「使用 Perl 撰寫的 Authorizer 應用程式 (SimpleAuth.pl)」
- 第 220 頁的「使用 C 撰寫的 Filter 應用程式 (SimpleFilter.c)」

使用 PHP 撰寫的回應程式應用程式 (ListDir.php)

```
<?php
    $dir = "/tmp/";

    // Open a known directory, and proceed to read its contents
    if (is_dir($dir)) {
        if ($dh = opendir($dir)) {
            while (($file = readdir($dh)) !== false) {
                echo "filename: $file : filetype: " . filetype($dir . $file) . "\n";
            }
            closedir($dh);
        }
    }
?>
```

上述範例的 obj.conf 程式碼片段：

```
<Object name="default">
    NameTrans fn="assign-name" from="/fcgi/*" name="responder.fcgi"
</Object>
<Object name="responder.fcgi">
    Service fn="responder-fastcgi" app-path="/foo/fastcgi-enabled-php-installation/bin/php"
    bind-path="localhost:3431" min-procs=3
</Object>
```

使用 Perl 撰寫的 Authorizer 應用程式 (SimpleAuth.pl)

```
#!/usr/bin/perl
```

```
use FCGI;
```

```
while (FCGI::accept >= 0) {
    if( $ENV{'HTTP_AUTHORIZATION'} ) {
        # This value can be further decoded to get the actual username and password and then
        # perform some kind of user validation. This program only checks for the presence of
        # of this environment param and is not really bothered about its value

        print( "Status: 200\r\n" );
        print( "\r\n" );

    } else {

        print( "Status: 401\r\n" );
        print( "WWW-Authenticate: basic realm=\"foo\"\r\n" );
        print( "\r\n" );

    }

}
```

Example obj.conf settings for the above example:

上述範例的 obj.conf 程式碼片段：

```
<Object name="responder.fcgi">
    AuthTrans fn="auth-fastcgi" app-path="/fastcgi/apps/auth/SimpleAuth.pl"
    bind-path="localhost:3432"
    Service fn="responder-fastcgi" app-path="/foo/fastcgi-enabled-php-installation/bin/php"
    bind-path="localhost:3433" app-env="PHP_FCGI_CHILDREN=8" min-procs=1
</Object>
```

第一次向 `http://localhost/fcgi/php/ListDir.php` 提出請求時，瀏覽器會顯示認證對話方塊。在使用者輸入使用者名稱和密碼後，即會列出「/tmp」目錄內容。

使用 C 撰寫的 Filter 應用程式 (SimpleFilter.c)

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <fcgi_stdio.h>

void main(void) {
    size_t PageSize = 1024 * 3;
    char *page;
    FCGX_Stream *in, *out, *err;
    FCGX_ParamArray envp;

    int count=0;
```

```

page = (char *)malloc(PageSize);

if (page == NULL) {

    printf("Content-type: text/x-server-parsed-html\r\n");
    printf("<title>malloc failure</title>");
    printf("<h1>Cannot allocate memory to run filter. exiting</h1>");
    printf("\r\n\r\n");
    exit(2);
}

while(FCGI_Accept() >= 0) {

    char *tmp;
    char *execcgi;
    char *dataLenStr = NULL;
    int numchars = 0;
    int stdinDataSize = 0;
    int filterDataLen = 0;
    int dataToBeRead = 0;
    int x = 0;
    int loopCount = 0;

    count++;
    dataLenStr = getenv("FCGI_DATA_LENGTH");

    if(dataLenStr)
        filterDataLen = atoi(dataLenStr);

    /* clear out stdin */
    while (EOF != getc(stdin)) {
        stdinDataSize++;
    }

    dataToBeRead = filterDataLen;
    FCGI_StartFilterData();
    tmp = page; /** just in case fread or fwrite moves our pointer **/

    //start responding
    printf("Content-type: text/plain\r\n");
    printf("\r\n"); /** send a new line at the beginning **/
    printf("<title>SIMPLE FILTER</title>");
    printf("<h1>This page was Filtered by SimpleFilter FastCGI filter</h1>");
    printf("file size=%d<br>", filterDataLen);
    printf("stdin size=%d<br>", stdinDataSize);

```

```

while(dataToBeRead > 0 ) {
    x = 0;
    page = tmp;

    if(dataToBeRead > PageSize)
        x = PageSize;
    else
        x = dataToBeRead;
    numchars = fread((void *)(page), 1, x, stdin);

    if( numchars == 0 )
        continue;
    /** at this point your data is in page pointer, so do
    whatever you want
    with it before sending it back to the server.
    In this example, no data is manipulated. Only the count of number of
    times the filter data is read and the total bytes read
    at the end of every
    loop is printed. */

    dataToBeRead -= numchars;
    loopCount++;
    printf("loop count = %d ... so far read %d bytes <br>", loopCount,
        (filterDataLen - dataToBeRead));
}
printf("\r\n\r\n"); /** send a new line at the end of transfer */

fflush(stdout);

page = tmp; /** restore page pointer */
memset(page, NULL, numchars);
}

free(page);
}

```

上述範例的 `obj.conf` 設定範例。

如果此 FastCGI 應用程式和 Web Server 在同一台機器上執行，則

```

<Object name="filter.fcgi">
    Service fn="filter-fastcgi" app-path="/fastcgi/apps/filter/SimpleFilter.exe"
    bind-path="localhost:3434" app-env="LD_LIBRARY_PATH=/fastcgi/fcgi-2.4/libfcgi/.libs"
</Object>

```

如果該應用程式是在遠端機器上執行，則 `obj.conf` 檔案中必須包含下列程式碼行：

```
<Object name="filter.fcgi">  
    Service fn="filter-fastcgi" bind-path="<remote-host>:<remote-port>"  
</Object>
```

如果要篩選位於 fcgi 目錄 (此目錄位於 Web Server 實例的文件根目錄下) 下大小為「26868」位元組的檔案「FilterThisFile」，則向「<http://localhost/fcgi/filter/FilterThisFile>」提出請求時，會產生下列輸出：

This page was Filtered by SimpleFilter FastCGI filter

```
file size = 26868  
stdin size = 0  
loop count = 1... so far read 3072 bytes  
loop count = 2... so far read 6144 bytes  
loop count = 3... so far read 9216 bytes  
loop count = 4... so far read 12288 bytes  
loop count = 5... so far read 15360 bytes  
loop count = 6... so far read 18432 bytes  
loop count = 7... so far read 21504 bytes  
loop count = 8... so far read 24576 bytes  
loop count = 9... so far read 26868 bytes
```


Web 服務

若要在 Sun Java System Web Server 7.0 上執行 Web 服務，則不需要任何額外配置。由於 JWSDP 已與伺服器整合，因此將所有 JWSDP Web 應用程式部署為 Web 應用程式後，這些應用程式都應該可以執行。

如需有關部署 Web 應用程式的更多資訊，請參閱第 151 頁的「增加 Web 應用程式」。

在 Web Server 7.0 上執行 JWSDP 2.0 範例

您需要在 JWSDP 2.0 中修改 Web 應用程式範例的配置檔案，然後再部署至 Web Server 7.0。亦即您必須編輯 `jaxws` 範例中的配置檔案，使其可部署至 Web Server 7.0。步驟如下：

▼ 執行 JWSDP 2.0 範例

- 1 下載 JWSDP 2.0。
 - 2 在 `$JWSDP_HOME/jwsdp-shared/bin` 下建立 Web Server 專屬的 `sjws.props`。
- 下方提供 `sjws.props` 範例。所有欄位都必填。

```
ADMIN_USER=admin
ADMIN_PORT=8800
ADMIN_HOST=localhost
ADMIN_PASSWORD_FILE=/tmp/admin.passwd
CONFIG=jwsdp
VS=jwsdp
WS_HOME=/export/ws7.0
WS_PORT=5555
WS_HOST=localhost
```

備註 – admin.passwd 檔案中包含管理員的伺服器密碼。以下為此項目的範例：wadm_password=adminadmin

3 修改配置檔案。

修改您想要執行的範例之 build.xml 和 etc/deploy-targets.xml 檔案。請注意，deploy-targets.xml 中所需的變更不是只針對本範例。您應該可以使用主副本，並將其複製到您想要執行的應用程式之 etc 目錄中。

build.xml 變更。

對 build.xml 上方的應用程式伺服器 lib.home 定義加入註釋，並增加 Web Server 程式庫位置。變更後的 build.xml 程式碼片段如下所示：

```
<!--
**
** 對應用程式伺服器 lib.home 宣告加入註釋 **
**
<property file="../../jwsdp-shared/bin/sjsas.props"/>
  <condition property="lib.home" value="${DOMAIN_DIR}/../lib">
    <available file="../../jwsdp-shared/bin/sjsas.props"/>
  </condition>
  <condition property="lib.home" value="${env.JAXWS_HOME}/lib">
    <not>
      <available file="../../jwsdp-shared/bin/sjsas.props"/>
    </not>
  </condition>
-->
<!--
** 增加 Web Server 程式庫位置 **
-->
<property name="lib.home" value="${WS_HOME}/lib" />
```

deploy-targets.xml 變更。

將 etc/deploy-targets.xml 替代成 Web Server 專屬的 deploy-targets.xml。此變更可將 Web 應用程式部署到 Web Server。deploy-targets.xml 檔案的程式碼片段如下所示：

```
<property environment="env"/>
<!-- Loading Web Server properties -->
<property environment="env"/>
<property file="../../jwsdp-shared/bin/sjsws.props"/>
<property name="ws.home" value="${WS_HOME}"/>
<property name="ws.admin" value="${ws.home}/bin/wadm"/>
<property name="lib.sample.home" value="${basedir}/../lib"/>
<property name="build.home" value="${basedir}/build"/>
<property name="build.classes.home" value="${build.home}/classes"/>
<property name="build.war.home" value="${build.home}/war"/>
<property name="config" value="${CONFIG}"/>
```

```
<target name="deploy">
  <exec executable="${ws.admin}" vmlauncher="true">
    <arg value="add-webapp" />
    <arg value="--user=${ADMIN_USER}" />
    <arg value="--password-file=${ADMIN_PASSWORD_FILE}" />
    <arg value="--host=${ADMIN_HOST}" />
    <arg value="--port=${ADMIN_PORT}" />
    <arg value="--config=${CONFIG}" />
    <arg value="--vs=${VS}" />
    <arg value="--uri=/jaxws-${ant.project.name}" />
    <arg value="${build.war.home}/jaxws-${ant.project.name}.war" />
  </exec>

  <antcall target="commit-config" />
</target>

<target name="commit-config">
  <exec executable="${ws.admin}" vmlauncher="true">
    <arg value="deploy-config" />
    <arg value="--user=${ADMIN_USER}" />
    <arg value="--password-file=${ADMIN_PASSWORD_FILE}" />
    <arg value="--host=${ADMIN_HOST}" />
    <arg value="--port=${ADMIN_PORT}" />
    <arg value="--force=true" />
    <arg value="${CONFIG}" />
  </exec>
</target>
```


字彙表

Access Control Entry, ACE (存取控制項目)	Web 伺服器用於評估進來的存取請求之規則階層。
Access Control List, ACL (存取控制清單)	ACL 集合。ACL 是一種用於定義哪些使用者對伺服器具有存取權限的機制。您可以定義針對特定檔案或目錄的 ACL 規則，授予或拒絕一個或多個使用者和群組進行存取。
Administration Server admpw	基於 Web 的伺服器，包含用於配置所有 Sun Java System Web Server 的表單。 Enterprise Administrator Server 超級使用者的使用者名稱與密碼檔案。
agent (代理程式)	在網路裝置 (如路由器、主機或 X 終端) 中執行網路管理軟體的軟體。另請參閱智慧型代理程式。
authentication (認證)	可讓 字彙表 向伺服器驗證其身份。基本認證或預設認證要求使用者輸入使用者名稱和密碼，才能存取 Web 伺服器或 Web 網站。認證需要 LDAP 資料庫中的使用者和群組清單。另請參閱摘要認證與 SSL 認證。 對整個伺服器或伺服器上特定檔案與目錄存取權限的授予。授權可能受到條件 (包括主機名稱與 IP 位址) 的限制。
cache (快取)	儲存在本機的原始資料副本。請求快取資料時，無須再次從遠端伺服器進行擷取。
certificate (憑證)	通訊雙方均已經信任的由協力廠商發佈的不可傳輸、不可偽造的數位檔案。
Certificate revocation list, CRL (憑證撤銷清單)	CA 提供的所有廢止憑證的 CA 清單。
certification authority, CA (憑證授權單位)	發佈用於加密異動之數位檔案的內部組織或協力廠商組織。
CGI	共用開道介面。外部程式與 HTTP 伺服器通訊所使用的介面。為使用 CGI 而撰寫的程式稱為 CGI 程式或 CGI 程序檔。CGI 程式處理伺服器通常不處理的表單，或者剖析伺服器通常不剖析的輸出。
Chroot	可建立的附加根目錄，用來限制伺服器可供存取的特定目錄。將使用此功能來保護未受保護的伺服器。

cipher (密碼)	密碼是一種用於加密或解密的加密演算法 (數學函數)。
ciphertext (密碼文字)	透過加密隱藏的資訊，僅預定接收者才能解密。
client (用戶端)	用來請求和檢視全球資訊網資料的軟體 (如 Netscape Navigator)。
client auth (用戶端認證)	用戶端認證。
cluster (叢集)	增加至「主」管理伺服器並受其控制的遠端「從屬」管理伺服器群組。叢集中的所有伺服器必須使用相同的平台，並具有相同的使用者 ID 與密碼。
collection (集合)	包含有關文件資訊 (例如文字清單和檔案特性) 的資料庫。搜尋功能使用集合來擷取匹配指定搜尋條件的文件。
Common LogFile Format (一般記錄檔案格式)	伺服器將資訊輸入到存取記錄所使用的格式。此格式在所有主伺服器 (包括 Sun Java System Web Server) 中均相同。
Compromised key list, CKL (洩漏金鑰清單)	關於具有洩漏金鑰的使用者之重要資訊清單。CA 也提供此清單。
daemon (常駐程式) (UNIX)	負責特定系統工作的後台程序。
DHCP	動態主機配置協定。網際網路建議標準協定 (IPSP)，允許系統動態地將指定給網路中的個別電腦。
digest authentication (摘要認證)。	允許使用者進行認證時無須傳送明文格式的使用者名稱和密碼。瀏覽器使用 MD5 演算法建立摘要值。伺服器使用「摘要式認證」外掛程式來比對用戶端所提供的摘要值。
DNS	網域名稱系統。網路上的機器用來將標準 IP 位址 (例如 198.93.93.10) 與主機名稱 (例如 www.sun.com) 關聯起來的系統。機器通常從 DNS 伺服器上取得此轉換資訊，或者在其系統上維護的表格中查詢此資訊。
DNS alias (DNS 別名)	DNS 伺服器所知道的主機名稱；指向不同的主機 — 也就是 DNS CNAME 記錄。機器始終只有一個實際名稱，但可以有一個或多個別名。例如，別名 (例如， www.yourdomain.domain) 可能指向伺服器目前所在的稱為 realthing.yourdomain.domain 的實際機器。
document root (文件根目錄)	伺服器機器上的目錄，包含要展示給存取伺服器之使用者的檔案、影像以及資料。
drop word (定位字)	請參閱停止字。
encryption (加密)	轉變資訊格式的程序，以便使除預定的接收者之外的任何人都無法解密或讀取。

expires header (過期標頭)	由遠端伺服器指定的傳回文件的過期時間。
extranet (企業間網路)	公司企業內部網路向網際網路的延伸，以便允許客戶、供應商以及遠端工作人員存取資料。
fancy indexing (Fancy 索引)	一種比簡單索引提供更多資訊的索引方法。Fancy 索引按名稱顯示內容清單，其中包括檔案大小、最後修改日期和反映檔案類型的圖示。因此，用戶端載入 Fancy 索引需要的時間要長於簡單索引。
file extension (副檔名)	檔案名稱的最後一部分，通常用於定義檔案的類型。例如，在檔案名稱 <code>index.html</code> 中，副檔名為 <code>html</code> 。
file type (檔案類型)	給定檔案的格式。例如，圖形檔案與文字檔案的檔案類型不同。檔案類型通常依副檔名 (<code>.gif</code> 或 <code>.html</code>) 來識別。
firewall (防火牆)	一種網路配置，通常既指硬體也指軟體，保護組織內部的網路電腦不被外部存取。防火牆通常用來保護資訊 (例如網路電子郵件以及實體建築或組織網站內的資料檔案)。
flexible log format (靈活的記錄格式)	伺服器將資訊輸入存取日誌所使用的格式。
FORTEZZA	美國政府機構用來管理一些敏感但不保密資訊的加密系統。
FTP	檔案傳輸協定。一種網際網路協定，允許檔案透過網路從一台電腦傳輸到另一台電腦。
GIF	圖形交換格式。起初由 CompuServe 建立的平台間影像格式。GIF 檔案通常遠小於其他圖形檔案類型 (BMP、TIFF)。GIF 是最常用的交換格式之一。在 UNIX、Microsoft Windows 以及 Apple Macintosh 等系統上，可以很容易地檢視 GIF 影像。
hard restart (硬式重新啟動)	程序或服務的終止以及其後續的重新啟動。另請參閱軟式重新啟動。
home page (首頁)	存在於伺服器上的文件，做為伺服器內容的目錄或進入點。伺服器的配置檔案中定義了此文件的位置。
hostname (主機名稱)	以 <i>machine.domain.com</i> 形式表示的機器的名稱，此名稱會被轉譯為 IP 位址。例如， <i>www.sun.com</i> 表示子網域 <i>sun</i> 與網域 <i>com</i> 中的機器 <i>www</i> 。
HTML	超文字標記語言。用於全球資訊網上文件的格式化語言。HTML 是具有格式化代碼的一般文字檔案，告知瀏覽器 (如 Netscape Navigator) 如何顯示文字、定位圖形與表單項目以及顯示通往其他頁面的連結。
HTTP	超文字傳輸協定。在 HTTP 伺服器與用戶端之間交換資訊的方法。
HTTP-NG	超文件傳輸協定的下一代。
HTTPD	HTTP 常駐程式或服務的縮寫，是一種使用 HTTP 通訊協定提供資訊的程式。
HTTPS	安全版本的 HTTP，透過安全傳輸層 (SSL) 實作。

imagemap (影像映射)	一種將影像區域變為使用中狀態，讓使用者只需在影像不同區域上按一下滑鼠便可進行瀏覽並取得資訊的程序。影像映射也可以指稱為「影像映射」的 CGI 程式，該程式用來處理其他 HTTPD 實作中的影像映射功能。
inittab (UNIX)	一個 UNIX 檔案，其中列出因為任何原因停止後需要重新啟動的程式。該檔案可確保程式持續執行。由於其位置關係，此檔案也稱為 <code>/etc/inittab</code> 。此檔案並非在所有 UNIX 系統上都可用。
intelligent agent (智慧型代理程式)	伺服器內的物件，代表使用者執行各種請求 (如 HTTP、NNTP、SMTP 以及 FTP 請求)。在某種意義上，智慧型代理程式作為伺服器的用戶端，發出請求要伺服器來完成。
IP address (IP 位址)	網際網路通訊協定位址。由點分隔的數字集，指定機器在網際網路上的實際位置 (例如，198.93.93.10)。
ISDN	整合式服務數位網路。
ISINDEX	一種可於用戶端開啓搜尋功能的 HTML 標記。文件可以使用網路助手的功能接受搜尋字串，並將該字串傳送至伺服器以存取可搜尋的索引，而無須使用表單。為了使用 <code><ISINDEX></code> ，必須建立查詢處理程式。
ISMAP	ISMAP 是 HTML 文件中所使用 <code>IMG SRC</code> 標籤的延伸，用來告知伺服器已命名影像為影像映射。
ISP	網際網路服務提供者。提供網際網路連接性的組織。
Java	Sun Microsystems 發明的物件導向式程式設計語言，用於建立稱為 <code>applet</code> 的即時、互動式程式。
Java Servlet	啓用所有 Java Servlet 元功能 (包括創設、初始化、損毀、自其他元件的存取以及配置管理) 的延伸。Java servlet 是可重複使用的 Java 應用程式，在 Web 伺服器上而不是 Web 瀏覽器中執行。
JavaScript	一種精簡的物件式程序檔編寫語言，用於開發用戶端及伺服器網際網路應用程式。
JavaServer Pages	啓用所有 JavaServer page 元功能 (包括創設、初始化、損毀、自其他元件的存取以及配置管理) 的延伸。JavaServer pages 是可重複使用的 Java 應用程式，在 Web 伺服器上而不是 Web 瀏覽器中執行。
last-modified header (最後一次修改的標頭)	在伺服器 HTTP 回應中傳回的文件檔案的最後修改時間。
LDAP database (LDAP 資料庫)	儲存用於認證的使用者與群組清單的資料庫。
listen socket (偵聽通訊端)	連接埠號碼與 IP 位址的組合。伺服器與用戶端的連線發生在偵聽套接字上。
magnus.conf	主 Web Server 配置檔案。此檔案包含全域伺服器配置資訊 (如連接埠、安全性等)。此檔案設定初始化期間配置伺服器之變數的值。Enterprise Server 在啟動時讀取此檔案並執行該變數設定。伺服器只有在重新啟動時才再次讀取此檔案，因此，您每次變更此檔案後，必須重新啟動伺服器。

MD5	由 RSA Data Security 提供的一種訊息摘要演算法。MD5 可以用來產生很短的資料摘要，此資料具有唯一性的概率很高。在數學上，極難產生可以產生相同訊息摘要電子郵件的資料。
MD5 signature (MD5 簽名)	MD5 演算法產生的訊息摘要。
MIB	管理資訊庫。
MIME	多用途網際網路郵件延伸。用於多媒體電子郵件與訊息傳送的新標準。
mime.types	MIME (多用途網際網路郵件延伸標準) 類型配置檔案。此檔案將檔案副檔名對映至 MIME 類型，以使伺服器能夠決定所請求內容的類型。例如，請求副檔名為 .html 的資源表示用戶端請求 HTML 檔案，而請求副檔名為 .gif 的資源，則表示用戶端請求 GIF 格式的影像檔。
modutil	安裝用於外部加密或硬體加速器裝置之 PKCS#11 模組所需的軟體公用程式。
MTA	郵件傳輸代理程式。您必須定義伺服器的 MTA 主機，才能在伺服器上使用代理程式服務。
network management station, NMS (網路管理工作站)	機器使用者可以用來遠端管理網路。管理裝置指執行 SNMP 的任意裝置，如主機、路由器以及 Web 伺服器。通常，NMS 是安裝了一個或多個網路管理應用程式的功能強大的工作站。
NIS (UNIX)	網路資訊服務。一種程式與資料檔案系統，UNIX 機器使用它在電腦網路中收集、排序及共有有關機器、使用者、檔案系統及網路參數的資訊。
NNTP	用於新聞群組的網路新聞傳送通訊協定。您必須定義新聞伺服器主機才能在伺服器上使用代理程式服務。
obj.conf	伺服器的物件配置檔案。此檔案包含附加初始化資訊、伺服器自訂設定以及伺服器用來處理用戶端 (如瀏覽器) 請求的說明。Sun Java System Web Server 每次處理用戶端請求時均會讀取此檔案。
password file (密碼檔案) (UNIX)	UNIX 機器上儲存 UNIX 使用者登入名稱、密碼以及使用者 ID 號碼的檔案。由於其儲存的位置，該檔案還稱為 /etc/passwd。
pk12util	一種軟體公用程式，用於將憑證與金鑰資料庫從內部機器匯出並匯入外部 PKCS#11 模組。
private key (私密金鑰)	公開金鑰加密時使用的解密金鑰。
protocol (通訊協定)	描述網路上的裝置如何交換資訊的規則集。
public information directories (公用資訊目錄) (UNIX)	不在 UNIX 使用者主目錄中文件根目錄內的目錄，或者受使用者控制的目錄。

public key (公開金鑰)	公開金鑰加密中使用的加密金鑰。
Quality of Service (服務品質)	為伺服器實例、虛擬伺服器類別或虛擬伺服器設定的效能限制。
RAM	隨機存取記憶體。電腦中基於半導體的實體記憶體。
rc.2.d (UNIX)	UNIX 機器上的一種檔案，用於描述機器啟動時執行的程式。由於其位置關係，該檔案還稱為 <code>/etc/rc.2.d</code> 。
redirection (重新導向)	將存取特定 URL 的用戶端傳送至同一伺服器或不同伺服器上不同位置的系統。如果資源已經移動，而且想要用戶端透明地使用新位置，此系統會非常有用。在無尾隨斜線的情況下存取目錄時，此系統還用來維護相對連結的完整性。
resource (資源)	可由伺服器存取、並傳送到提出請求之用戶端的任何文件 (URL)、目錄或程式。
RFC	註釋請求。通常指提交給網際網路社群的程序或標準文件。在技術成為接受的標準之前，使用者可以對其進行註釋。
root (超級使用者) (UNIX)	UNIX 機器上最具特權的使用者。超級使用者具有對機器上所有文件的完全存取權限。
server daemon (伺服器常駐程式)	一旦執行就偵聽並接受用戶端請求的程序。
Server Plug-in API (伺服器外掛程式 API)	一種延伸功能，可讓您延伸和/或自訂 Sun Java System Web Server 的核心功能，並提供可延伸、有效率的機制來建立 HTTP 伺服器與後端應用程式間的介面。也稱為 NSAPI。
server root (伺服器根目錄)	伺服器機器上的目錄，專門用於保存伺服器程式、配置、維護和資訊檔案。
simple index (簡單索引)	與 Fancy 索引相對的索引—此類目錄清單僅顯示檔案名稱，而不顯示任何圖形元素。
SNMP	簡易網路管理協定。
SOCKS	防火牆軟體或硬體 (例如路由器配置) 阻止建立直接連線時，在防火牆內部和外部建立連線的防火牆軟體。
soft restart (軟式重新啟動)	一種重新啟動伺服器的方法，使伺服器內部重新啟動，即重新讀取其配置檔案。軟式重新啟動機制會傳送 HUP 訊息 (信號一) 至程序。程序本身不會像在硬式重新啟動中那樣終止。
SSL	安全傳輸層。在雙方 (用戶端和伺服器) 之間建立安全連線的軟體程式庫，可用於實作 HTTPS (即安全版本的 HTTP)。
SSL authentication (SSL 認證)	透過使用用戶端憑證中的資訊做為身份證明或者驗證 LDAP 目錄中發佈的用戶端憑證，來確認具有安全性憑證的使用者身份。
stop word (停止字)	搜尋功能識別為非搜尋字的單字。這通常包括 the、a、an 以及 and 之類的字。另請參閱定位字。

strftime	將日期和時間轉換為字串的函式。伺服器在附加尾部時使用此函式。strftime 的日期與時間具有特殊的格式語言，伺服器可以在尾部使用該語言來說明最後一次修改檔案的日期。
Sun Java System Web Server Administration Console (Sun Java System Web Server 管理主控台)	一個為伺服器管理員提供圖形化介面的 Java 應用程式，用於從企業網路中任意一個中央位置管理所有 Sun Java System Web Server。透過 Sun Java System Web Server 管理主控台的任何安裝實例，均可以查看和存取企業網路中已授予您存取權限的所有 Sun Java System 伺服器。
superuser (超級使用者) (UNIX)	UNIX 機器上最具特權的使用者 (也稱為 root)。超級使用者具有對機器上所有檔案的完全存取權限。
Sym-links (符號式連結) (UNIX)	符號是符號式連結的縮寫，是 UNIX 作業系統使用的一種重新導向類型。符號式連結可讓您建立從檔案系統的一部分指向檔案系統另一部分現有檔案或目錄的指標
TCP/IP	傳輸控制協定/網際網路協定。網際網路與企業 (公司) 網路的主要網路協定。
telnet	網路上兩台機器互相連線並支援遠端登入終端機模擬的通訊協定。
timeout (逾時)	指定的時間，此時間過後，伺服器會放棄完成顯示當機服務常式的嘗試。
TLS	安全套接字層。在雙方 (用戶端與伺服器) 之間建立安全連線的軟體程式庫，可用於實作 HTTPS (即安全版本的 HTTP)。
top (UNIX)	某些 UNIX 系統上的一種程式，顯示目前系統資源使用狀態。
top-level domain authority (頂層網域授權)	主機名稱分類的最高層種類，通常表示網域所代表的組織類型 (例如 .com 表示公司，.edu 表示教育機構) 或者來源國家/地區 (例如 .us 表示美國，.jp 表示日本，.au 表示澳大利亞，.fi 表示芬蘭)。
uid (UNIX)	與 UNIX 系統上每一位使用者相關聯的唯一編號。
URI	統一資源識別碼。通過使用縮寫的 URL 提供安全性附加層的檔案識別碼。URL 的第一部分取代為 URL 對映，該對映可對使用者隱藏檔案的完整實體路徑名稱。另請參閱 URL 對映。
URL	統一資源定址器。伺服器與用戶端請求文件所使用的尋址系統。URL 通常稱為位置。URL 的格式為 <i>protocol://machine:port/document</i> 。 http://www.sun.com/index.html 為一個 URL 範例。
URL database repair (URL 資料庫修復)	一種程序，用於修復與更新因軟體錯誤、系統當機、磁碟損壞或檔案系統滿載而受損的 URL 資料庫。

URL mapping
(URL 對映)

將文件目錄的實體路徑名稱對映至使用者定義的別名的程序，這樣該目錄中的檔案僅需參考目錄的別名，而不是檔案的完整實體路徑名稱。如此，可以將檔案識別為 `/myDocs/index.html`，而不是 `usr/sun/servers/docs/index.html`。使用者無需瞭解伺服器檔案的實體位置，為伺服器提供了附加安全性。

virtual server
(虛擬伺服器)

虛擬伺服器是一種使用單一安裝伺服器設置多重網域名稱、IP 位址及伺服器監視功能的方法。

virtual server class
(虛擬伺服器類別)

共用 `obj.conf` 檔案中相同基本配置資訊的虛擬伺服器集合。

web application
(Web 應用程式)

Servlet、JavaServer Pages、HTML 文件以及其他 Web 資源 (可能包括影像檔、壓縮的歸檔檔案以及其他資料) 的集合。Web 應用程式可以封裝到歸檔檔案中 (WAR 檔案)，或存在於開放式目錄結構中。

**Web Application
Archive, WAR**
(Web 應用程式歸檔檔案)

包含壓縮格式的完整 Web 應用程式之歸檔檔案。

Windows CGI
(Windows)

以 Windows 程式設計語言 (例如 Visual Basic) 撰寫的 CGI 程式。

索引

A

Accept Language 標頭, 使用, 202-203
ACL, 伺服器摘要認證程序, 91
ACL 使用者快取, 伺服器將儲存使用者和群組認證結果, 93
ACLCacheLifetime, 93
ACLUserCacheSize, 93
Administration Server
 URL 瀏覽, 27
 從控制面板中啟動服務 applet, 26
ansi_x3.4-1968, 123
ansi_x3.4-1986, 123
ascii, 123

C

CA
 定義 (憑證授權單位), 73, 76
certmap.conf, 90
CGI, 122
 shell, 121-122
 下載可執行檔, 121
 副檔名, 119
 程式, 如何在伺服器上儲存, 118
 簡介, 117
CGIStub, 程序用於協助 CGI 執行, 117
CGIStubIdleTimeout, 117
COPY, 139
cp367.0, 123
cp819, 123
CRL (憑證撤銷清單), 安裝和管理, 80

current.zip, 32

D

DELETE, 98
digestauth, 91
DigestStaleTimeout, 92
Directory Server, ldapmodify 指令行公用程式, 105
DNS, 降低查詢對伺服器效能的影響, 92

G

GET, 98
GIF, 已定義, 231

H

HEAD, 98
.htaccess, 動態配置檔案, 98
HTML
 已定義, 231
 伺服器剖析, 設定, 126
HTTP, 已定義, 231
http_head, 98
HTTPD, 231
HTTPS, 已定義, 231

I

- ibm367.0, 123
- ibm819, 123
- INDEX, 98
- inittab, 已定義, 232
- IP 位址
 - 已定義, 232
 - 限制存取, 88
- iso-2022-jp, 123
- iso_646.irv, 1991, 123
- iso-8859-1, 123
- iso_8859-1, 123
 - 1987.0, 123
- iso-ir-100, 123
- iso-ir-6, 123
- iso646-us, 123

J

- Java EE, 管理資源, 156
- JDBC, JDBC API, 157
- JSP 標籤規格, 186

L

- latin1, 123
- LDAP
 - 使用者名稱和密碼認證, 90, 229
 - 管理使用者和群組, 101
- ldapmodify, Directory Server 指令行公用程式, 105
- LDIF
 - 匯入和匯出功能, 關於, 103
 - 增加資料庫項目, 103
- LOCK, 139

M

- magnus.conf
 - ACLCacheLifetime 指令, 93
 - 終止逾時, 92
- MaxCGIStub, 117
- MD5, 定義, 233

- memberCertDescriptions, 107
- memberURL 篩選器, 107
- memberURLs, 107
- MIME
 - charset 參數, 123
 - octet-stream, 121
- MIME, 已定義, 233
- MIME 類型, 指定預設, 112-113
- MinCGIStub, 117
- MKCOL, 139
- MKDIR, 98
- MOVE, 98, 139
- MTA, 已定義, 233

N

- NIS, 已定義, 233
- NNTP, 已定義, 233
- nonce, 92

O

- obj.conf, 預設認證, 89
- octet-stream, 121

P

- PathCheck, 98
- POST, 98
- PROPFIND, 139
- PROPPATCH, 139
- PUT, 98

R

- RAM, 已定義, 234
- rc.2.d, 234
- RMDIR, 98

S

shell CGI, 121-122
 SMUX, 192
 SNMP
 子代理程式, 189, 191
 在伺服器上設定, 190, 191, 192
 基本, 189
 SOCKS, 已定義, 234
 SSL
 已定義, 234
 需要啓用的資訊, 77
 認證, 91
 SSL 2 通訊協定, 85
 SSL 3 通訊協定, 84, 85
 SSL2 通訊協定, 84
 SSL3 通訊協定, 84

T

telnet, 235
 TLS, 傳輸層安全性, 84
 TLS 加密通訊協定, 85
 TLS 通訊協定, 84

U

uid, 已定義, 235
 uniqueMembers, 107
 UNLOCK, 139
 URI, 已定義, 235
 URL
 已定義, 235
 存取 Administration Server, 27
 對映, 已定義, 236
 URL 轉寄, 配置, 115
 us, 123
 us-ascii, 123

W

Web 應用程式, 已定義, 236
 Web 應用程式歸檔 (WAR), 已定義, 236

WebDAV

Sun Java System Web Server 處理鎖定請求的方式, 146
 URI, 136
 方法, 138
 COPY, 139
 LOCK, 139
 MKCOL, 139
 MOVE, 139
 PROPFIND, 139
 PROPPATCH, 139
 UNLOCK, 139
 內部成員 URI, 137
 成員 URI, 137
 來源 URI, 136
 特性, 137
 啓用 WebDAV 的用戶端, 135
 集合, 137
 新增 HTTP 方法, 138
 新增 HTTP 標頭, 138

X

x-euc-jp, 123
 x-mac-roman, 123
 x-sjis, 123
 一般記錄檔案格式, 定義, 230
 子代理程式
 SNMP, 189, 191
 文件目錄
 主 (文件根目錄), 111
 限制內容發佈, 114
 文件頁尾, 設定, 124
 文件根目錄, 設定, 111
 文件喜好設定, 預設 MIME 類型, 指定, 112-113
 公用目錄, 配置, 113
 公用目錄 (Unix), 自訂, 113-115
 公開金鑰, 73
 內容壓縮
 分段大小, 129
 內容壓縮配置, 128-130
 依需求壓縮內容, 129-130
 啓動, 128
 插入 Vary 標頭, 129

內容壓縮 (續)

- 提供預先壓縮內容, 128-129
- 壓縮層級, 129
- 內部成員 URI, 137
- 反向代理伺服器, 130
- 可執行檔, 下載, 121
- 生命週期模組, 154
- 加密, 雙向, 84
- 用戶端認證, 定義, 74
- 用戶端憑證, 認證, 90-91
- 主文件目錄, 設定 (文件根目錄), 111
- 主機 IP 認證, 92
- 主機名稱
 - 已定義, 231
 - 限制存取, 88
 - 認證, 92
- 企業間網路, 定義, 231
- 字元集
 - iso_8859-1, 123
 - us-ascii, 123
 - 變更, 123-124
- 多位元組資料, 201
- 安全傳輸層 (SSL), 加密通訊協定, 84
- 存取
 - 刪除, 98
 - 執行, 98
 - 清單, 98
 - 資訊, 98
 - 網站, 限制 (全域和單一實例), 94
 - 寫入, 98
 - 讀取, 98
- 存取記錄自動重建, 198
- 存取控制
 - 方法 (基本、SSL), 89
 - 主機名稱和 IP 位址, 88
 - 使用者和群組, 88
 - 簡介, 87, 88-89
- 存取控制清單 (ACL), 87
- 存取控制項目 (ACE), 87
- 共用閘道介面 (CGI), 簡介, 117
- 自訂搜尋, 182-183
 - 在單獨的頁面中自訂表單和結果, 185
 - 自訂搜尋結果頁面, 183-185
- 自動重建, 存取記錄, 198

- 刪除存取, 98
- 成員 URI, 137
- 快取, 已定義, 229
- 快取控制指令, 設定, 127
- 伺服器, LDAP 使用者和群組, 國際化注意事項, 201
- 伺服器根, 已定義, 234
- 伺服器常駐程式, 已定義, 234
- 伺服器認證, 定義, 74
- 定位字, 230
- 限制符號連結, 125-126
- 來源 URI, 136
- 使用者
 - 限制存取, 88
 - 認證, 89-92
- 使用者/群組認證, 89, 92
- 使用者目錄, 配置, 113
- 使用者目錄 (Unix), 自訂, 113-115
- 使用者和群組, 使用 LDAP 進行管理, 101
- 使用者和群組認證, 結果儲存在 ACL 使用者快取中, 93
- 金鑰, 定義, 84
- 重新導向, 234
- 記錄檔, 歸檔, 198
- 軟式 (符號) 連結, 定義, 125
- 頂層網域權限, 235
- 副檔名
 - CGI, 119
 - 已定義, 231
- 偵測版本回復, 85
- 停止字, 234
- 國際化注意事項, LDAP 使用者和群組, 201
- 執行存取, 98
- 控制, 存取, 簡介, 87
- 終止逾時, magnus.conf, 92
- 符號 (軟式) 連結, 定義, 125
- 符號連結, 限制, 125-126
- 強制連結, 定義, 125
- 啟動指令, Unix 平台, 26
- 密碼, 定義, 84
- 密碼檔案, 233
 - 啟動時載入, 115
- 清單存取, 98

程式

CGI

如何在伺服器上儲存, 118

超級使用者, 已定義, 234

超級使用者, 已定義, 235

過期標頭, 已定義, 231

虛擬伺服器

公用目錄, 配置以使用, 113-115

範例, 大量託管, 67

範例, 企業網路主機作業, 66-67

範例, 安全伺服器, 66

範例, 預設配置, 66

部署, 65

簡介, 65

集合, 已定義, 230

最大連線數, 133

最大傳輸速率, 133

最大認證資料量, 85

傳輸層安全性 (TLS), 加密通訊協定, 84

資訊存取, 98

資料庫項目, 使用 LDIF 增加, 103

資源, 已定義, 234

搜尋

JSP 標籤規格, 186

URI, 174

介面元件, 182

在單獨的頁面中自訂表單和結果, 185

自訂搜尋查詢頁面, 182-183

自訂搜尋頁面, 181-186

自訂搜尋結果頁面, 183-185

查詢, 179-180

進階搜尋, 180

搜尋頁面, 179

路徑, 174

檢視搜尋結果, 181

關於, 173-174

搜尋基底 (基底 DN), 使用者 ID, 105

群組

限制存取, 88

描述 LDAP 資料庫中物件集的物件, 107

認證, 89-92

認證, 使用者, 89

群組, 靜態

定義, 107

群組, 靜態 (續)

建立準則, 108

摘要認證, 91

ACL 的伺服器程序, 91

管理介面, 更多資訊, 20

網站, 限制存取 (全域和單一實例), 94

網域名稱系統

已定義, 230

別名, 已定義, 230

網路管理工作站 (NMS), 189, 191

實例, 術語, 35

語言標頭, 接受, 使用, 202-203

認證

SSL, 91

用戶端憑證, 90-91

主機名稱, 92

使用者和群組, 89-92

認證, 用戶端, 伺服器, 定義, 74

認證, 主機 IP, 92

認證, 使用者/群組, 89, 92

認證, 基本, 結合 SSL 加密、主機 IP 認證或兩者時最有效, 90

認證, 摘要, 91

認證逾時, 85

認證資料庫, 103

寫入存取, 98

請求/摘要, 92

篩選器, memberURL, 107

錯誤, 自訂回應, 122

橢圓曲線加密, 74

辨別名稱 (DN) 屬性, 定義, 102

憑證

簡介, 73, 76

憑證, 用戶端, 認證, 90-91

憑證授權單位

定義, 73, 76

憑證撤銷清單 (CRL), 安裝和管理, 80

憑證請求, 所需資訊, 77

靜態群組

定義, 107

建立準則, 108

檔案類型, 定義, 231

雙向加密, 密碼, 84

簡易目錄存取協定 (LDAP), 管理使用者和群組, 101

鎖定資源

Sun Java System Web Server 處理鎖定請求的方式, 146

共用鎖定, 145

專用鎖定, 145

瀏覽, 透過 URL 存取 Administration Server, 27

歸檔, 記錄檔, 198

屬性, 辨別名稱 (DN), 102

讀取存取, 98