



Sun Java System Web Server 7.0 管理员指南



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

文件号码 820-0872
2006 年 10 月

版权所有 2006 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. 保留所有权利。

本文档及其相关产品的使用、复制、分发和反编译均受许可证限制。未经 Sun 及其许可方（如果有）的事先书面许可，不得以任何形式、任何手段复制本产品或文档的任何部分。第三方软件，包括字体技术，均已从 Sun 供应商处获得版权和使用许可。

本产品的某些部分可能是从 Berkeley BSD 系统衍生出来的，并获得了加利福尼亚大学的许可。UNIX 是 X/Open Company, Ltd. 在美国和其他国家/地区独家许可的注册商标。

Sun、Sun Microsystems、Sun 徽标、docs.sun.com、AnswerBook、AnswerBook2、Java 和 Solaris 是 Sun Microsystems, Inc. 在美国和其他国家/地区的商标或注册商标。所有的 SPARC 商标的使用均已获得许可，它们是 SPARC International, Inc. 在美国和其他国家/地区的商标或注册商标。标有 SPARC 商标的产品均基于由 Sun Microsystems, Inc. 开发的体系结构。

OPEN LOOK 和 SunTM 图形用户界面是 Sun Microsystems, Inc. 为其用户和许可证持有者开发的。Sun 感谢 Xerox 在研究和开发可视或图形用户界面的概念方面为计算机行业所做的开拓性贡献。Sun 已从 Xerox 获得了对 Xerox 图形用户界面的非独占性许可证，该许可证还适用于实现 OPEN LOOK GUI 和在其他方面遵守 Sun 书面许可协议的 Sun 许可证持有者。

美国政府权利—商业软件。政府用户应遵循 Sun Microsystems, Inc. 的标准许可协议，以及 FAR（Federal Acquisition Regulations，即“联邦政府采购法规”）的适用条款及其补充条款。

本文档按“原样”提供。对任何明示或暗示的条件、陈述和担保，包括任何暗示的适销性、适用于特定用途的适用性以及非侵犯性，均不承担任何责任，除非此免责声明的适用范围在法律上无效。

目录

前言	19
1 入门指南	25
简介	25
新增功能	25
启动 Administration Server	26
在 Unix/Linux 中启动 Administration Server	26
▼ 在 Unix/Linux 中启动 Administration Server	26
在 Windows 中启动 Administration Server	26
管理您的服务器的不同方法	26
使用管理控制台	27
管理控制台 GUI 屏幕的帮助	29
使用 CLI	29
CLI 模式	29
何处可以找到 wadm CLI	30
CLI 中的验证	30
了解 Web Server 7.0	31
2 配置、实例和节点	35
概述	35
管理配置	35
创建配置	36
复制服务器配置	37
部署服务器配置	38
删除服务器配置	38
管理服务器实例	38
创建服务器实例	39

启动服务器实例	39
停止服务器实例	40
重新启动服务器实例	40
重新配置服务器实例	40
删除服务器实例	41
自动配置实例	42
▼ 添加安排的事件	42
▼ 删除安排的事件	43
3 服务器群和群集	45
Sun Java System Web Server 中的群集支持	45
设置服务器群	45
▼ 设置服务器群	46
设置简单的群集	47
▼ 配置群集	49
4 部署方案	51
部署体系结构	51
部署概述	53
部署前的要求	55
部署 Web Server	55
群集环境	56
硬件和软件要求	56
设置群集	58
会话复制	60
会话复制和故障转移操作	60
启用会话复制	61
配置 Web 应用程序进行会话复制	62
监视群集	63
Solaris Zones	63
5 使用虚拟服务器	65
虚拟服务器概述	65
使用案例	65

默认配置	66
安全服务器	66
内部网托管	66
海量托管	67
管理虚拟服务器	67
添加虚拟服务器	67
▼ 添加虚拟服务器	67
配置虚拟服务器	68
▼ 配置虚拟服务器	68
复制虚拟服务器	69
▼ 复制虚拟服务器	69
配置 HTTP 侦听器	69
创建 HTTP 侦听器	70
配置 HTTP 侦听器	70
6 证书和密钥	73
使用证书进行验证	73
服务器验证	74
客户机验证	74
证书密钥类型	74
创建自签名证书	76
管理证书	76
请求证书	76
▼ 请求证书	77
安装证书	78
▼ 安装证书	78
更新证书	79
▼ 更新证书	79
删除证书	80
▼ 删除证书	80
更新 Administration Server 证书	80
管理证书撤销列表 (Certificate Revocation List, CRL)	81
▼ 安装 CRL	81
▼ 删除 CRL	81
设置内部令牌密码	82

▼ 设置令牌密码	82
为服务器配置 SSL	83
为配置启用 SSL 加密算法	83
为 HTTP 侦听器启用安全性	84
7 控制对服务器的访问	87
什么是访问控制	87
访问控制的工作原理	88
为用户/组设置访问控制	89
默认验证	90
基本验证	90
SSL 验证	91
摘要验证	91
为主机/IP 设置访问控制	93
配置 ACL 用户高速缓存	93
设置 ACL 高速缓存属性	94
配置访问控制	94
添加访问控制列表 (Access Control List, ACL)	94
添加访问控制条目 (Access Control Entry, ACE)	96
使用 .htaccess 文件	98
防止服务器遭受拒绝服务攻击	99
限制对服务器的请求	99
▼ 限制最大连接数	100
8 管理用户和组	101
访问用户和组的信息	101
关于目录服务	101
目录服务的类型	102
了解标识名 (Distinguished Name, DN)	102
使用 LDIF	103
使用验证数据库	103
创建验证数据库	104
设置用户和组	105
▼ 添加用户	105
▼ 添加组	105

▼ 删除用户	106
▼ 删除组	107
静态组和动态组	107
静态组	108
动态组	108
9 管理服务器内容	111
配置文档目录	111
▼ 创建文档目录	112
更改默认 MIME 类型	112
▼ 更改默认 MIME 类型	113
自定义用户公共信息目录 (UNIX/Linux)	113
▼ 配置文档目录	114
限制内容发布	114
启动时装入整个密码文件	115
设置 URL 重定向	115
使用正则表达式进行 URL 重定向	116
CGI 概述	117
配置服务器的 CGI 子系统	119
下载可执行文件	121
安装 Windows Shell CGI 程序	121
Windows Shell CGI 程序概述	121
自定义错误响应	122
更改字符集	123
▼ 更改字符集	124
设置文档页脚	124
▼ 设置文档页脚	124
限制符号链接 (UNIX/Linux)	125
▼ 限制符号链接	125
设置服务器解析的 HTML	126
▼ 设置服务器解析的 HTML	126
设置高速缓存控制指令	127
▼ 设置高速缓存控制指令	127
配置服务器的内容压缩	128
将服务器配置为提供预压缩的内容	128

▼ 更改预压缩的内容的设置	128
将服务器配置为根据需要压缩内容	129
▼ 根据需要压缩内容	129
配置反向代理	130
▼ 添加代理 URI	130
▼ 修改反向代理参数	131
设置 P3P	132
▼ 配置虚拟服务器的 P3P 设置	132
10 使用 WebDAV 实现 Web 发布	135
关于 WebDAV	136
常见 WebDAV 术语	136
在实例级别启用 WebDAV	139
管理 WebDAV 集合	140
启用 WebDAV 集合	140
禁用 WebDAV 集合	140
添加 WebDAV 集合	140
列出 WebDAV 集合	140
删除 WebDAV 集合	140
配置 WebDAV 属性	141
设置 WebDAV 属性	141
查看 WebDAV 属性	141
设置 WebDAV 集合属性	141
查看 WebDAV 集合属性	141
修改 WebDAV 参数	142
在服务器级别禁用 WebDAV	143
管理 WebDAV 验证数据库	143
在启用了 WebDAV 的服务器上使用源 URI 和 Translate:f 标头	144
锁定和解除锁定资源	144
互斥锁定	145
共享锁定	145
最小锁定超时	145
11 使用 Java 和 Web 应用程序	147
配置 Java 以使用 Sun Java System Web Server	147

▼ 为配置启用 Java	147
设置 Java 类路径	148
▼ 设置 Java 类路径	148
配置 JVM	149
▼ 配置 JVM	149
添加 JVM 选项	149
添加 JVM 事件探查器	149
▼ 添加 JVM 事件探查器	150
为服务器启用 Java 调试	150
▼ 启用 JVM 调试	150
部署 Java Web 应用程序	151
添加 Web 应用程序	151
▼ 部署 Web 应用程序	151
部署 Web 应用程序目录	152
部署期间预编译 JSP	152
配置 Servlet 容器	153
▼ 设置 Servlet 容器	153
Servlet 容器全局参数	153
配置服务器生命周期模块	154
服务器生命周期介绍	154
▼ 添加生命周期模块	154
▼ 删除生命周期模块	155
配置 Java 资源	156
配置 JDBC 资源	157
Sun Java System Web Server 支持的 JDBC 驱动程序	157
管理 JDBC 资源	160
▼ 添加新 JDBC 资源	160
管理 JDBC 连接池	160
▼ 创建 JDBC 连接池	160
注册自定义资源	162
▼ 添加自定义资源。	162
使用外部 JNDI 资源	163
▼ 添加外部 JNDI 资源	163
配置邮件资源	164
▼ 添加邮件资源	164
配置 SOAP 验证提供商	165

▼ 添加 SOAP 验证提供商	166
SOAP 验证提供商参数	166
配置会话复制	167
设置会话复制	168
▼ 设置会话复制	168
管理验证领域	169
▼ 添加验证领域	171
12 使用搜索集合	173
关于搜索	173
配置搜索属性	174
配置搜索集合	175
支持的格式	175
添加搜索集合	175
删除搜索集合	177
安排集合更新	177
执行搜索	179
“搜索” 页面	179
进行查询	179
▼ 进行查询	180
高级搜索	180
▼ 进行高级搜索查询	180
文档字段	181
搜索查询运算符	181
查看搜索结果	181
自定义搜索页面	181
搜索界面组件	182
自定义搜索查询页面	182
自定义搜索结果页面	184
在单独的页面中自定义表单和结果	185
标记惯例	185
标记规范	186
13 监视服务器	187
Sun Java System Web Server 的监视功能	187

通过管理控制台进行监视	188
▼ 查看统计信息	188
修改监视参数	189
配置监视参数	190
配置 SNMP 子代理参数	190
配置 SNMP 子代理	191
使用 CLI 配置 SNMP	193
▼ 激活 Solaris 上的 SNMP	193
▼ 激活 Linux 上的 SNMP	194
▼ 激活 Windows 上的 SNMP	194
▼ 配置基于对等点的主代理 (magt)	195
设置服务器的日志记录	195
日志类型	196
查看访问日志和服务器日志	196
配置日志参数	196
配置 Administration Server 的日志设置	199
▼ 修改服务器日志位置	199
▼ 修改日志详尽程度	200
▼ 修改日志的日期格式	200
14 国际化和本地化	201
输入多字节数据	201
文件名称或目录名称	201
LDAP 用户和组	201
支持多字符编码	202
WebDAV	202
搜索	202
配置服务器以提供本地化内容	202
▼ 搜索顺序	202
A 对前一版本中的 CLI 的更改	205
B FastCGI 插件	209
简介	209

插件函数 (SAF)	210
auth-fastcgi	210
responder-fastcgi	210
filter-fastcgi	211
error-fastcgi	211
FastCGI SAF 参数	211
error-fastcgi SAF 错误原因字符串	213
在 Web Server 上配置 FastCGI 插件	213
修改 magnus.conf	214
修改 MIME 类型（可选）	214
修改 obj.conf	215
排除 FastCGI 插件的故障	216
开发 FastCGI 应用程序	217
▼ 执行 FastCGI 应用程序	217
FastCGI 应用程序样例	219
使用 PHP 编写的响应者应用程序 (ListDir.php)	219
使用 Perl 编写的授权者应用程序 (SimpleAuth.pl)	219
使用 C 编写的过滤器应用程序 (SimpleFilter.c)	220
 C Web 服务	225
在 Web Server 7.0 上运行 JWSDP 2.0 样例	225
▼ 运行 JWSDP 2.0 样例	225
 词汇表	229
 索引	237



图 4-1	流程图说明了如何在单个节点上部署 Web 服务器	54
图 4-2	群集设置	57
图 4-3	说明群集设置过程的流程图	58

表

表 6-1	HTTP 侦听器安全性属性	85
表 7-1	摘要验证的不同情况	92
表 7-2	ACL 参数	95
表 7-3	ACE 参数	96
表 7-4	配置请求限制	99
表 8-1	动态组：必需的参数	110
表 9-1	URL 重定向参数	116
表 9-2	CGI 参数	120
表 10-1	WebDAV 参数	142
表 10-2	WebDAV 验证数据库属性	143
表 10-3	Sun Java System Web Server 处理锁定请求的方式	146
表 11-1	Servlet 容器参数	153
表 11-2	常用和受支持的 JDBC 驱动程序列表	158
表 11-3	自定义资源的属性	162
表 11-4	外部 JNDI 资源的属性	164
表 11-5	邮件资源的属性	165
表 11-6	SOAP 验证提供商参数	166
表 11-7	会话复制参数	169
表 11-8	领域类型	170
表 12-1	字段说明 > 新建搜索事件进度表	178
表 13-1	监视类别	188
表 13-2	字段说明 > 常规监视设置	190
表 13-3	字段说明 > SNMP 子代理设置	191
表 13-4	一般准则	192
表 13-5	字段说明 > 编辑访问日志首选项	197
表 13-6	字段说明 > 编辑服务器日志首选项	197
表 13-7	字段说明 > 设置日志轮转	198
表 A-1	对前一版本中的 CLI 的更改	205

示例

前言

本指南介绍如何配置和管理 Sun Java™ System Web Server 7.0。

目标读者

本书适用于在生产环境中管理服务器的 Sun Java System Web Server 管理员。本指南假定您熟悉以下方面：

- 安装软件
- 使用 Web 浏览器
- 执行基本的系统管理任务
- 在终端窗口中发布命令

阅读本书之前

Sun Java System Web Server 7.0 可作为独立产品安装，也可作为 Sun Java Enterprise System (Java ES) 的组件安装，后者是一种软件基础结构，可支持在网络或Internet环境中分布的企业应用程序。如果将 Sun Java System Web Server 7.0 作为 Java ES 的组件进行安装，则应熟悉 <http://docs.sun.com/coll/1286.2> 和 <http://docs.sun.com/coll/1382.2> 中的系统文档。

Sun Java System Web Server 7.0 文档集

Sun Java System Web Server 7.0 文档集介绍如何安装和管理 Web Server。Sun Java System Web Server 7.0 文档的 URL 是 <http://docs.sun.com/coll/1308.3> 和 <http://docs.sun.com/coll/1395.2>。有关 Sun Java System Web Server 7.0 的介绍，请按下表所列顺序参阅相关书籍。

表 P-1 Sun Java System Web Server 7.0 文档集中的文档

文档标题	内容
《Sun Java System Web Server 7.0 Documentation Center》	按任务和主题组织的 Web Server 文档主题
《Sun Java System Web Server 7.0 发行说明》	<ul style="list-style-type: none">■ 软件和文档的最新信息■ 支持的平台和安装 Web Server 时的修补程序要求
《Sun Java System Web Server 7.0 Installation and Migration Guide》	执行安装和迁移任务： <ul style="list-style-type: none">■ 安装 Web Server 及其各种组件■ 将数据从 Sun ONE Web Server 6.0 或 6.1 迁移到 Sun Java System Web Server 7.0
《Sun Java System Web Server 7.0 管理员指南》	执行以下管理任务： <ul style="list-style-type: none">■ 使用管理界面和命令行界面■ 配置服务器首选项■ 使用服务器实例■ 监视和记录服务器活动■ 使用证书和公共密钥加密以确保服务器的安全■ 配置访问控制以确保服务器的安全■ 使用 JavaPlatform Enterprise Edition (Java EE) 安全性功能■ 部署应用程序■ 管理虚拟服务器■ 定义服务器工作负载和调整系统大小以满足性能需求■ 搜索服务器文档的内容和属性，以及创建文本搜索界面■ 配置服务器以进行内容压缩■ 配置服务器以使用 WebDAV 进行 Web 发布和内容制作
《Sun Java System Web Server 7.0 Developer's Guide》	使用编程技术和 API 执行以下操作： <ul style="list-style-type: none">■ 扩展和修改 Sun Java System Web Server■ 动态生成内容以响应客户机请求，以及修改服务器的内容
《Sun Java System Web Server 7.0 Update 1 NSAPI Developer's Guide》	创建自定义的 Netscape 服务器应用程序编程接口 (Netscape Server Application Programmer's Interface, NSAPI) 插件
《Sun Java System Web Server 7.0 Developer's Guide to Java Web Applications》	在 Sun Java System Web Server 中实现 Java Servlet 和 JavaServer Pages™ (JSP™) 技术
《Sun Java System Web Server 7.0 Administrator's Configuration File Reference》	编辑配置文件
《Sun Java System Web Server 7.0 Performance Tuning, Sizing, and Scaling Guide》	调节 Sun Java System Web Server 以优化性能
《Sun Java System Web Server 7.0 Troubleshooting Guide》	Web Server 故障排除

相关书籍

有关 Sun Java Enterprise System (Java ES) 及其组件的所有文档的 URL 是
<http://docs.sun.com/app/docs/prod/entsys.06q4> 和
<http://docs.sun.com/app/docs/prod/entsys.06q4?l=zh>

默认路径和文件名

下表介绍了本书中使用的默认路径和文件名。

表 P-2 默认路径和文件名

占位符	描述	默认值
<i>install_dir</i>	表示 Sun Java System Web Server 7.0 的基本安装目录。	Solaris™ 平台上的 Sun Java Enterprise System (Java ES) 安装： /opt/SUNWwbsvr7 Linux 和 HP-UX 平台上的 Java ES 安装： /opt/sun/webserver/ Windows 平台上的 Java ES 安装： System Drive:\Program Files\Sun\JavaES5\WebServer7 其他 Solaris、Linux 和 HP-UX 的安装（非超级用户）： user's home directory/sun/webserver7 其他 Solaris、Linux 和 HP-UX 的安装（超级用户）： /sun/webserver7 Windows，所有安装： SystemDrive:\Program Files\Sun\WebServer7

表 P-2 默认路径和文件名 (续)

占位符	描述	默认值
<i>instance_root</i>	包含特定于实例的子目录的目录。	实例在 Solaris 中的默认位置为： <i>/var/opt/SUNWwbsvr7</i> 实例在 Linux 和 HP-UX 中的默认位置为： <i>/var/opt/sun/webserver7</i> 实例在 Windows 中的默认位置为： <i>System Drive:\Program Files\sun\WebServer7</i> 对于 Java ES 安装，实例在 Windows 中的默认位置为： ： <i>System Drive:\Program Files\Sun\JavaES5\WebServer7</i>

印刷约定

下表描述了本书中使用的印刷约定。

表 P-3 印刷约定

字体	含义	示例
<i>AaBbCc123</i>	命令、文件和目录的名称；计算机屏幕输出	编辑 <i>.login</i> 文件。 使用 <i>ls -a</i> 列出所有文件。 <i>machine_name% you have mail.</i>
AaBbCc123	用户键入的内容，与计算机屏幕输出的显示不同	<i>machine_name% su</i> Password:
<i>AaBbCc123</i>	要使用实名或值替换的命令行占位符	删除文件的命令为 <i>rm filename</i> 。
新词术语强调	新词或术语以及要强调的词（注：某些强调的词在联机状态下以粗体显示。）	高速缓存 是存储在本地的副本。 请勿保存文件。
《书名》	书名	阅读《用户指南》的第 6 章。

符号约定

下表介绍了本书中可能使用的符号。

表 P-4 符号约定

符号	描述	示例	含义
[]	包含可选参数和命令选项。	ls [-l]	-l 选项不是必需的。
{ }	包含所需命令选项的一组选择。	-d {y n}	-d 选项要求您使用 y 参数或 n 参数。
\${ }	表示变量引用。	\${com.sun.javaRoot}	引用 com.sun.javaRoot 变量的值。
-	结合同时发生的多个击键。	Ctrl-A	按 A 键的同时按 Ctrl 键。
+	结合相继发生的多个击键。	Ctrl+A+N	按下 Ctrl 键后放开，然后再按后几个键。
→	表示图形用户界面中的菜单项选择。	“文件”→“新建”→“模板”	从“文件”菜单中选择“新建”。从“新建”子菜单中选择“模板”。

联机访问 Sun 资源

通过 <http://docs.sun.com> (docs.sun.comSM) Web 站点，可以联机访问 Sun 技术文档。您可以浏览 docs.sun.com 中的归档文件或搜索特定的书名或主题。这些书是以联机文件的形式提供的，有 PDF 和 HTML 两种格式。残障人士用户可以通过辅助技术读取这两种格式。

要访问以下 Sun 资源，请转至 <http://www.sun.com>：

- 下载 Sun 产品
- 服务和解决方案
- 支持（包括修补程序和更新）
- 培训
- 研究
- 团体（例如，Sun 开发者网络）

搜索 Sun 产品文档

除了从 docs.sun.com Web 站点搜索 Sun 产品文档外，还可以使用搜索引擎进行搜索，方法是在搜索字段中键入以下语法：

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

例如，要搜索 "Web Server"，请键入以下内容：

`Web Server site:docs.sun.com`

要将其他 Sun Web 站点包括在搜索范围内（例如，java.sun.com、www.sun.com 和 developers.sun.com），请在搜索字段中使用 "sun.com" 替换 "docs.sun.com"。

第三方 Web 站点引用

本文档引用了第三方 URL 以提供其他相关信息。

注 - Sun 对本文档中提到的第三方 Web 站点的可用性不承担任何责任。对于此类站点或资源中的（或通过它们获得的）任何内容、广告、产品或其他材料，Sun 并不表示认可，也不承担任何责任。对于因使用或依靠此类站点或资源中的（或通过它们获得的）任何内容、产品或服务而造成的、名义上造成的或连带产生的实际或名义损坏或损失，Sun 概不负责，也不承担任何责任。

Sun 欢迎您提出意见

Sun 致力于提高其文档的质量，并十分乐意收到您的意见和建议。为了共享您的意见，请访问 <http://docs.sun.com>，并单击 "Send Comments"。在联机表单中，请提供完整的文档标题和文件号码。文件号码是一个七位或九位的数字，可以在书的标题页或文档的 URL 中找到。例如，本书的文件号码为 820-0872。

入门指南

本章通过提供本指南中所使用术语的简要说明，介绍了 Sun Java System Web Server 7.0 的基础知识。

- 第 25 页中的 “简介”
- 第 25 页中的 “新增功能”
- 第 26 页中的 “启动 Administration Server”
- 第 26 页中的 “管理您的服务器的不同方法”
- 第 27 页中的 “使用管理控制台”
- 第 29 页中的 “使用 CLI”
- 第 31 页中的 “了解 Web Server 7.0”

简介

Sun Java System Web Server 7.0 是基于行业标准构建的多进程多线程安全 Web 服务器。它具备高性能、可靠性、可伸缩性和可管理性，适用于大中型企业。

Web Server 7.0 提供了全面的命令行界面支持、巩固的配置、增强的安全性（支持椭圆曲线密码算法）以及群集支持。它还附带了强健的内置迁移工具，使用该工具可以将应用程序和配置从 Web Server 6.0 和 6.1 迁移到 Sun Java System Web Server 7.0。

新增功能

有关 Sun Java System Web Server 7.0 的新增功能和增强功能的更多信息，请参见《Sun Java System Web Server 7.0 发行说明》中的第 1 章“Sun Java System Web Server 发行说明”。

启动 Administration Server

要使用管理界面，需要启动 Administration Server。

在 Unix/Linux 中启动 Administration Server

要启动 Administration Server，请执行以下任务：

▼ 在 Unix/Linux 中启动 Administration Server

- 1 转到 `install_root/admin-server/bin` 目录（例如，`/usr/sjsws7.0/admin-server/bin`）
- 2 键入 `./startserv`。
此命令使用您在安装过程中指定的端口号启动 Administration Server。

在 Windows 中启动 Administration Server

Sun Java System Web Server 安装程序可以创建包含若干 Windows 平台图标的程序组。程序组包括以下图标：

- 发行说明
- 启动 Administration Server
- 卸载 Web Server

请注意，Administration Server 是作为服务 applet 运行的，因此您也可以使用“控制面板”来直接启动此服务。

管理您的服务器的不同方法

您可以使用以下用户界面管理 Sun Java System Web Server：

- 管理控制台 (GUI)。
- 命令行界面 (`wadm shell`)。

您可以使用 `wadm shell` 界面（将在本章后面讨论）或基于 Web 的管理控制台来管理实例。请注意，管理节点只可以有一个正在运行的特定配置的实例。

使用管理控制台

安装 Sun Java System Web Server 后，请使用浏览器访问管理控制台。

您用于导航至 "Administration Server" 页面的 URL 取决于计算机的主机名以及安装 Sun Java System Web Server 时为此 Administration Server 选择的端口号。例如，如果您将 Administration Server 安装在 SSL 端口 1234 上，相应的 URL 为：

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

您需要登录到管理控制台才能执行服务器管理。您在计算机上安装 Sun Java System Web Server 时设置了管理员用户名和密码。下图显示了验证屏幕：



访问 Administration Server 时首先看到的页面是日常任务页面。您可以使用此页面上的按钮管理、添加、删除和迁移 Sun Java System Web Server。下图显示了日常任务页面：



注- 单击其中任一选项卡可使其子选项卡显示在页面上。子选项卡提供的操作仅适用于父选项卡功能。

下图显示了选定选项卡的子选项卡：



单击选项卡将在同一窗口中打开页面。有些特定任务可以通过一系列步骤收集用户的数据。管理控制台具有此类任务的向导界面。这些向导始终在新的窗口中打开。

管理控制台 GUI 屏幕的帮助

所有表单元素和 GUI 组件都具有详细的内联帮助，可提供有关验证和可选参数的信息。对于向导界面，可在向导中的任何一个步骤中单击“帮助”选项卡以获取特定于当前任务的帮助。

使用 CLI

本节介绍 Sun Java System Web Server 7.0 的命令行界面，并定义用于配置和管理服务器的所有支持的命令。

Sun Java System Web Server 7.0 引入了称为 wadm 的新 CLI。

服务器的早期版本支持一些独立的命令行，这些命令行一起仅构成 GUI 中所提供全部管理功能的一个子集。Sun Java System Web Server 6.1 中支持的命令行界面有 HttpServerAdmin、wdeploy 和 flexanlg。新 CLI (wadm) 功能：

- 用于脚本编写的嵌入式 JACL shell。
- 可扩展的 CLI—可通过第三方插件向 CLI 中添加更多命令。

注 – Sun Java System Web Server 7.0 不支持 HttpServerAdmin。

注 – Sun Java System Web Server 7.0 中支持 wdeploy，其目的仅为了保持与 6.x 版本向后兼容，该命令将仅处理 Administration Server 节点。

CLI 模式

wadm 支持 3 种模式下的调用。它们是：

- **单机模式**—在此模式下，可从命令 shell 调用 wadm，并指定需要的命令、选项和操作数。命令执行完后，CLI 将退出并返回到 shell。此模式既支持命令的交互执行，也支持非交互执行。默认情况下支持交互执行，如果密码文件中未指定密码（通过 --password-file 选项传递），将会提示您输入密码。如果未指定 --password-file 选项，非交互执行将会产生错误。例如，

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

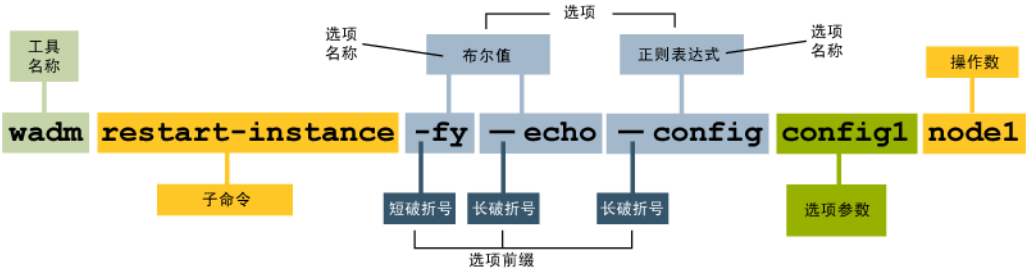
- **Shell 模式**—在此模式下，可从命令 shell 调用 wadm，但不指定命令。wadm 将提示用户输入命令。执行命令后，将返回到 shell。键入 exit 或 quit 命令可退出此 shell。交互执行和非交互执行都适用于此模式。例如，

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

- **文件模式**—在此模式下，可以在文件中添加一个命令列表，然后将该文件作为参数传递到 wadm。例如，

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

下图描述了调用 wadm 命令的语法。



注 - 使用 wadm CLI 时，可以执行使用管理控制台可以执行的所有任务。

何处可以找到 wadm CLI

问题: 哪里可以找到 Sun Java System Web Server 7.0 的管理 CLI?

答案: 管理 CLI 位于 install-root/bin/wadm 中。要使用 CLI，需要知道：

- 管理服务器主机名（默认为 localhost）。
- 管理服务器的 SSL 端口（默认为 8989）。
- 管理服务器用户名（默认为 admin）。
- 管理服务器密码。

注 - 要使用 CLI，需要运行管理服务器。可通过运行 install-root/admin-server/bin/startserv 启动该服务器。

CLI 中的验证

wadm 将使用管理员的用户名和密码通过 Administration Server 的验证。必须将有效的用户名和密码文件作为参数传递给单一模式下运行的每个命令。调用 wadm 可执行文件时，shell 模式接受用户名和密码文件。shell 模式下调用的命令不需要连接选项（例如，user、password-file、host、port 和 ssl）。如果指定了连接选项，它们将会被忽略。

CLI 支持的一些命令要求输入密码。例如，`bindpw`、`user-password` 和 `token-pin`。用户可以在包含管理用户密码的同一文件中指定这些密码。如果在指定命令时未指定 `password-file`，将会提示用户输入密码。

`wadm` 通过 SSL（如果 Administration Server 中启用了 SSL）与 Administration Server 通信。Administration Server 传递的证书将会根据 `truststore` (`~/.wadmtruststore`) 进行验证。如果该证书存在并且有效，命令将正常继续执行。否则，`wadm` 将显示该证书，并让用户选择是否接受它。如果用户接受该证书，证书将被添加到 `truststore` 中，并且命令将正常继续执行。

注 - 由于 `truststore` 不包含任何敏感数据，所以不需要进行密码保护。

了解 Web Server 7.0

Web Server 包含为服务器群中的服务器提供增强的分布式管理的新管理框架。强健的管理功能使 Web Server 可通过图形界面和命令行界面进行远程管理和部署。服务器可通过服务器群中的一个中心位置进行管理，并可以分布到一个或多个节点以创建服务器实例。此外，还提供了这些服务器实例的监视和生命周期管理。

您可以配置 Web Server 以打开或关闭各种功能、决定如何响应单个客户机请求以及编写基于服务器的操作并与操作进行交互的程序。标识这些选项的指令都存储在配置文件中。Sun Java System Web Server 将在启动时以及客户机请求将您的选择映射到所需服务器活动期间读取这些配置文件。

有关这些文件的更多信息，请参见《Sun Java System Web Server 7.0 Administrator's Configuration File Reference Guide》。

在 Web Server 7.0 中，服务器实例的所有可配置元素（如 Web 应用程序、配置文件和搜索集合索引）都在逻辑上进行了分组，并使用术语**配置**表示。使用 CLI 或基于 Web 的管理界面可以创建、修改或删除配置。您可以一次管理多个配置。术语“配置”是指用于配置服务器的运行时服务的一组元数据。例如，运行时服务可处理已配置文档根目录中的 Web 页面。服务器运行时使用配置元数据来加载内置服务和第三方插件，并安装其他服务器扩展，例如用于处理 Web 页面和动态 Web 应用程序的数据库驱动程序。

注 - 所有与配置相关的文件都存储在文件系统中名为**配置库**的系统信息库中。除非本指南中明确指定，否则必须限制手动编辑此系统信息库中的任何文件。

在 Web Server 中，使用 CLI 或基于 Web 的管理界面对“配置”的任何更改都首先在配置库中进行，然后部署配置。因此，这些更改将被复制到实例目录。部署 Web 应用程序时，该应用程序将部署到以下位置：

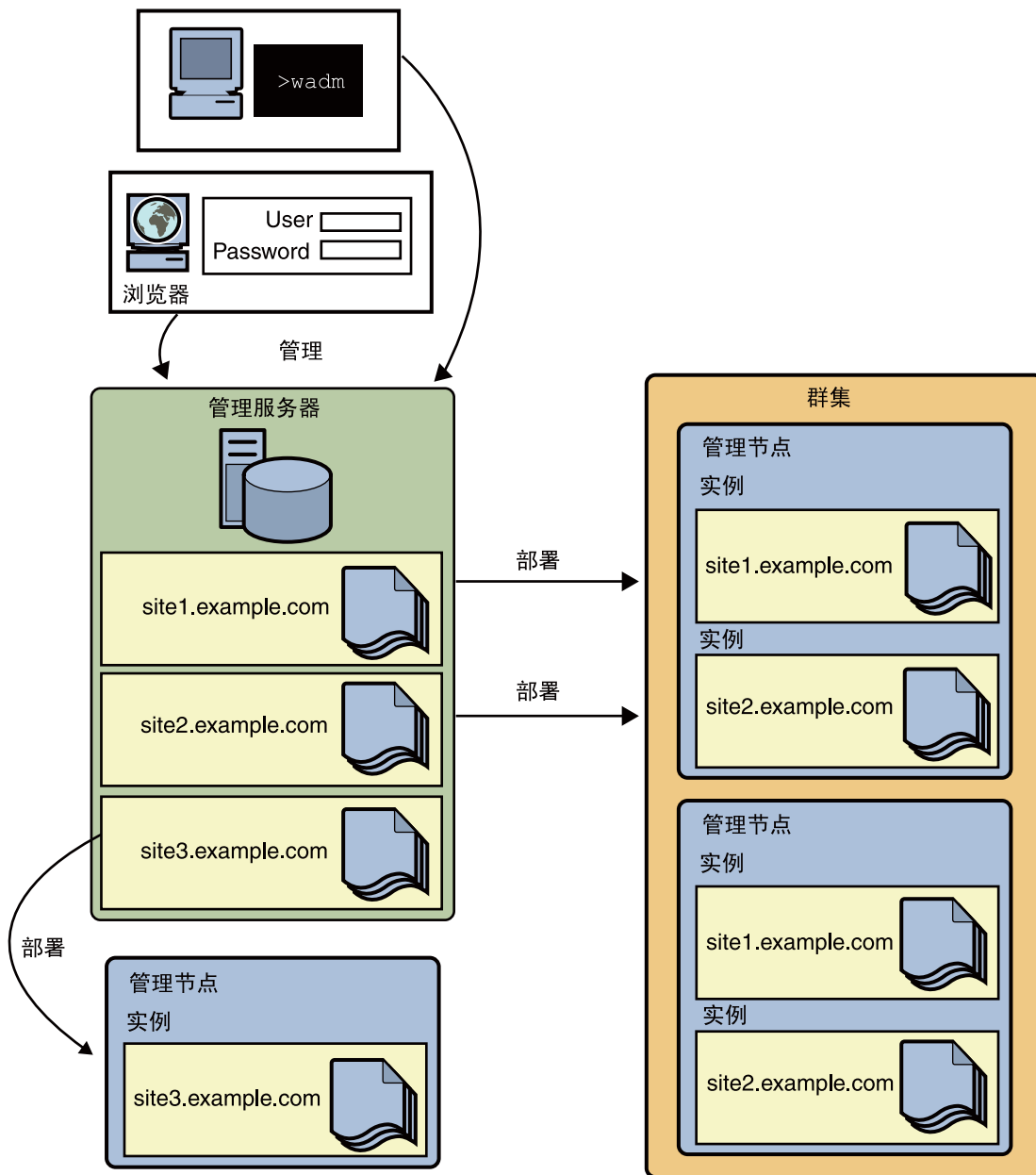
```
<install_dir>/admin-server/config-store/<config_name>/web-app/<virtual_servername>/
```

部署配置时，config-store 下的整个 Web 应用程序目录和配置目录都将被压缩，并复制到服务器实例目录中。该压缩文件为 `current.zip`，位于以下位置中：

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

因此，根据 Web 应用程序的大小，部署选定的配置可能需要一段时间才能完成。

下图显示了如何将配置部署到管理节点的示意图。



将配置部署到节点（网络资源，如服务器或主机）时，将创建该配置的实例。实例包含日志文件和其他必需的运行时文件，例如锁定数据库、高速缓存和临时文件。您可以通过 CLI 或基于 Web 的管理界面来管理这些实例。

实例可以跨越一个或多个节点而构成一个**群集**。如果是群集，构成该群集的所有节点必须具有相同的配置。群集中的所有节点必须同源的。这些节点必须具有相同的操作系统、进行相同的配置并提供相同的服务。

服务器群中的一个节点运行将在其中部署管理应用程序的服务器。此专门配置的服务器称为 **Administration Server**，所部署的管理应用程序是基于 Web 的**管理控制台**。您可以使用管理控制台控制服务器实例的生命周期。

Administration Server 控制该节点（称为**管理节点**）中其他服务器的操作。管理节点未提供 GUI 界面。服务器群中的一个节点安装 Administration Server。服务器群中的所有其他节点安装管理节点。管理节点在安装时会在 Administration Server 中注册。此操作将使 Administration Server 可识别该管理节点。

Administration Server 和管理节点始终通过 SSL 通信。Administration Server 和管理节点相互验证，方法是 Administration Server 信任管理节点的服务器证书，管理节点信任由 Administration Server 提供的客户机证书。在注册管理节点的过程中，Administration Server 将生成该管理节点的服务器证书，然后可以在管理节点中下载并安装该证书。管理节点中还将安装服务器证书的颁发者。

配置、实例和节点

上一章介绍了 Web Server 7.0 中的一些新概念。管理员的主要任务是配置和管理服务器的运行时服务。本章介绍可以管理配置的不同方法，以及如何部署配置以在节点中启动实例。

- 第 35 页中的 “概述”
- 第 35 页中的 “管理配置”
- 第 38 页中的 “管理服务器实例”
- 第 42 页中的 “自动配置实例”

概述

实例是指给定节点上 Web 服务器守护进程的环境，包括其配置、日志文件和其他运行时辅件，例如锁定数据库、高速缓存和临时文件。

节点是网络资源，例如服务器或主机。在典型数据中心中，节点构成的网络称为**服务器群**。本节讨论如何使用管理控制台 GUI 配置节点。

您可以将一个或多个实例部署到节点。此外，还可以将同一实例部署到多个节点，同一实例也可以构成不同群集的一部分。

为了管理目的，可以启动、停止或动态重新配置实例。

管理配置

- 第 36 页中的 “创建配置”
- 第 37 页中的 “复制服务器配置”
- 第 38 页中的 “部署服务器配置”
- 第 38 页中的 “删除服务器配置”

创建配置

开始使用 Web 服务器之前，您需要创建配置。

要创建新配置，请执行以下步骤：

1. 单击**配置**选项卡。
2. 单击**新建**按钮。

向导将指导您完成用于创建配置的可用设置。以下各节提供了向导页面中的可用字段说明。

步骤 1—设置配置信息

此向导页面允许您设置新配置的一般信息。

在向导页面中设置以下参数：

- **配置名称**—为配置添加新的唯一名称。
- **服务器名**—为新配置添加服务器名。此名称可与配置名称相同。
- **文档根目录**—请输入有效的文档根目录，所有已部署的 Web 应用程序均在此维护自己的目录。默认值为 `../docs`。您可以输入服务器上任何有效目录的路径。
- **64 位**—启用/禁用 Web 服务器的 64 位支持。默认值为**禁用**。
- **服务器用户**—如果服务器在基于 UNIX 的系统上运行，请提供服务器进程的有效用户名。例如，`root`。

部署 2—为配置创建侦听器

此向导页面允许您为新配置设置 HTTP 侦听器属性。

在向导页面中设置以下参数：

- **端口**—输入配置绑定的端口号，该端口还将用来侦听请求。
- **IP 地址**—主机的 IP 地址。键入 `*` 将设置所有可用的 IP 地址。

步骤 3—配置 Java、CGI 和 SHTML

此向导页面允许您配置与 Java/CGI 和 SHTML 相关的属性。

在向导页面中设置以下参数：

- **Java**—已启用。默认情况下，启用 Java。**警告**：如果需要使用此配置部署基于 Java 的 Web 应用程序，请不要禁用 Java。设置 Java SE 目录的主目录。默认值为指向捆绑的 Java SE 目录的目录。可以选择默认 Java SE 目录或指定新路径。
- **CGI**—无（禁用 CGI 支持）、启用为文件类型（启用 CGI 支持）和目录（指定将存储 CGI 文档的 URI 和路径）。

- SHTML—默认情况下，禁用 SHTML。

步骤 4—创建实例

此向导页面允许您为新配置创建实例。

在向导页面中设置以下参数：

- 配置—新配置的名称。
- 选择节点—选择要为新配置创建实例的节点。从可用列表中选择节点，然后单击**添加**或**全部添加**按钮以添加节点。

注—使用 CLI

要通过 CLI 创建配置，请执行以下命令：

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

config1 为新配置的名称。

请参见 CLI 参考 create-config(1)。

复制服务器配置

可以复制服务器配置和创建新配置。新复制的配置与现有配置相同。但是，新配置将不具有任何实例，即使从其进行复制的配置包含实例。

要复制配置，请执行以下步骤：

1. 单击**配置**选项卡。
2. 从配置列表中选择相应的配置。
3. 单击**复制**按钮。
4. 在弹出式窗口中，输入新的配置名称，然后单击“确定”。

注-使用 CLI

要通过 CLI 执行操作，请执行以下命令：

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

copyconfig1 是新配置的名称。

请参见 CLI 参考 `copy-config(1)`。

部署服务器配置

您需要在节点上先创建一个要部署的配置。

要部署现有的配置，请执行以下步骤：

1. 单击**配置**选项卡。
2. 通过选择配置复选框来标识配置。
3. 从配置操作列表中选择**部署配置**。

删除服务器配置

注-如果配置实例已部署到节点上，不能删除此配置。即使这些实例已部署但未运行，您也不能删除此服务器配置。停止正在运行的实例并取消部署这些实例可以删除配置。

要删除配置，请执行以下步骤：

1. 单击**配置**选项卡。
2. 通过选择配置复选框来标识配置。
3. 从配置操作列表中选择**删除配置**。

管理服务器实例

- 第 39 页中的 “创建服务器实例”
- 第 39 页中的 “启动服务器实例”
- 第 40 页中的 “停止服务器实例”
- 第 40 页中的 “重新启动服务器实例”
- 第 40 页中的 “重新配置服务器实例”

- 第 41 页中的“删除服务器实例”

创建服务器实例

创建新服务器实例之前，请先执行以下检查：

1. 检查您是否已创建配置。创建新的服务器实例需要指定现有的实例配置。
2. 检查服务器群中的所有可用节点是否均已具有所需配置的实例。您不能创建重复的实例。

要创建新服务器实例，请执行以下步骤：

1. 单击**配置**选项卡，并从操作列表中选择**新建**。
2. 在“新建实例向导”页面中，为您需要创建的实例选择配置，并单击**下一步**按钮。
3. 选择选定配置 [步骤 2] 的实例所在的节点。单击**下一步**按钮。
4. 查看所做选择的摘要。单击**下一步**按钮以查看操作的结果。

注 - 使用 CLI

要创建服务器实例，请执行以下命令：

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

请参见 CLI 参考 create-instance(1)。

启动服务器实例

1. 单击**节点**选项卡以查看在服务器中配置的节点列表。
2. 通过选择节点名称复选框来选择节点。
3. 单击**启动实例**按钮将打开一个页面窗口，其中列出了由该节点控制的所有实例。
4. 选择一个实例并单击**启动实例**按钮以启动该实例。
5. 检查该实例的状态是否为“正在运行”并关闭窗口。

注 - 使用 CLI

要通过 CLI 启动服务器实例，请执行以下命令：

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

请参见 CLI 参考 start-instance(1)。

停止服务器实例

1. 单击**节点**选项卡以查看在服务器中配置的节点列表。
2. 通过选择节点名称复选框来选择节点。
3. 单击**停止实例**按钮将打开一个页面窗口，其中列出了由该节点控制的所有实例。
4. 选择一个实例并单击**停止实例**按钮以停止该实例。
5. 检查该实例的状态是否为“未运行”并关闭窗口。

注 - 使用 CLI

要通过 CLI 停止服务器实例，请执行以下命令：

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

请参见 CLI 参考 `stop-instance(1)`。

重新启动服务器实例

1. 单击**节点**选项卡以查看在服务器中配置的节点列表。
2. 通过选择节点名称复选框来选择节点。
3. 单击**重新启动实例**按钮将打开一个页面窗口，其中列出了由该节点控制的所有实例。
4. 选择一个实例并单击**重新启动实例**按钮以重新启动该实例。
5. 检查该实例的状态是否为“正在运行”并关闭窗口。

注 - 使用 CLI

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

请参见 CLI 参考 `restart-instance(1)`。

重新配置服务器实例

对配置进行更改时，一些情况下无需重新启动实例。Administration Server 支持重新配置服务器实例，以引入对配置库所进行的更改。在此配置中，更改将会反映到实例中，无需重新启动服务器。配置中只有可动态重新配置的更改会受到影响。

注-用户、临时路径、日志、线程库、pkcs11、统计信息、CGI、DNS、DNS 高速缓存、文件高速缓存、ACL 高速缓存、SSL 会话高速缓存、访问日志缓冲区和 JVM（日志级别除外）设置的更改只有在重新配置后才会生效。执行重新配置时，将会记录要求重新启动的所有更改。重新配置文件高速缓存要求重新启动服务器。

1. 单击**节点**选项卡以查看在服务器中配置的节点列表。
 2. 通过选择节点名称复选框来选择节点。
 3. 单击**重新配置实例**按钮将打开一个页面窗口，其中列出了在该节点上已部署的所有实例。
 4. 选择一个实例并单击**重新配置实例**按钮以重新配置该实例。
 5. 检查该实例的状态是否为“正在运行”并关闭窗口。
-

注-使用 CLI

要通过 CLI 重新配置服务器实例，请执行以下命令：

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

请参见 CLI 参考 reconfig-instance(1)。

删除服务器实例

注-要删除的服务器实例应处于未运行状态。

1. 单击**配置**选项卡以查看可用配置的列表。
 2. 从配置列表中选择配置。
 3. 单击**实例子**选项卡。
 4. 从**节点**部分下已部署的实例列表中选择实例。
 5. 从操作下拉式列表中选择**删除实例**以删除选定的实例。
-

注-使用 CLI

要通过 CLI 删除服务器实例，请执行以下命令：

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

请参见 CLI 参考 delete-instance(1)。

自动配置实例

可以根据安排的事件重新配置实例或重新启动实例。您可以设置特定的时间和时间间隔来安排自动重新配置实例。

要安排事件，请执行以下步骤：

1. 单击**配置**选项卡并选择相应的配置。
2. 单击**常规**子选项卡 > **安排的事件**子选项卡。

▼ 添加安排的事件

1 选择配置。

单击“配置”选项卡后，从显示的列表中选择配置。

2 单击“常规”>“安排的事件”子选项卡。

3 单击“新建”按钮。

4 配置以下属性：

■ 事件

- **重新启动实例**—此安排的事件将为相应的配置重新启动所有已部署且正在运行的实例。
- **重新配置实例**—此安排的事件将为相应的配置重新配置所有已部署且正在运行的实例。
- **自定义命令行**—提供将要执行的文件的绝对路径。

■ 进度表

已配置的事件将启动的时间。从下拉式框中选择小时和分钟值。

- **每天**—每天在指定的时间启动指定的事件。
- **特定日期**—在特定日期启动指定的事件。
 1. **星期几**—指定从星期日至星期六的任意一天。
 2. **日期**—指定一个月中从1日至31日间的任意一天（以逗号分隔条目）。例如，4,23,9。
- **特定月份**—在特定时间和月份启动指定的事件。指定从一月至十二月之间的月份。
- **时间间隔**

每隔此时间段即启动指定的事件。

 1. **间隔小时数**—从下拉式框中选择小时数。
 2. **间隔秒数**—在文本字段中输入秒数。

注-使用 CLI

要通过 CLI 安排事件，请执行以下命令：

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

请参见 CLI 参考 create-event(1)。

▼ 删除安排的事件

1 选择配置。

单击“配置”选项卡后，从显示的列表中选择配置。

2 单击“常规”>“安排的事件”子选项卡。

3 选择安排的事件并单击“删除”按钮。

服务器群和群集

前面几章介绍了配置以及如何将配置部署到节点。在本章中，将设置一个简单的服务器群和群集。

- 第 45 页中的 “Sun Java System Web Server 中的群集支持”
- 第 45 页中的 “设置服务器群”
- 第 47 页中的 “设置简单的群集”

Sun Java System Web Server 中的群集支持

群集是一组实例，跨越一个或多个节点，全部运行相同的配置并提供一组相同的运行时服务。每个群集必须包含一个指定为 Administration Server 的服务器。如果有多个群集，则可以通过一个主 Administration Server 来管理所有群集。主 Administration Server 将检索所有群集的相关信息并提供一个界面，以管理安装在各个群集中的 Sun Java System Web Server。

注 - 群集中的所有实例都必须是同源的。例如，所有实例都在相同的操作系统版本（和修补程序）以及服务包中运行，运行相同的 Web 服务器配置并提供相同的服务。

设置服务器群

要设置群集，需要首先安装一个 Administration Server 和一个或多个管理节点。要管理这些管理节点，需要分别将其在管理服务器中注册。在安装节点过程中或安装之后，可通过 wadm CLI 执行注册操作。

▼ 设置服务器群

1 安装 Administration Server 和管理节点

安装管理服务器。可通过 Sun Java System Web Server 安装程序 GUI 或 wadm CLI 安装管理服务器。

您可以选择**快速安装**选项（将在端口 8989 上安装管理服务器），也可以选择**自定义安装**选项以设置首选项。要安装管理服务器，请在安装程序设置屏幕中选择选项**将服务器安装为 Administration Server**。您需要指定 SSL 端口，但可以指定一个非 SSL 端口，也可以不指定。

注 - 如果指定了非 SSL 端口，将会在管理服务器节点中创建一个管理节点，无需显式在管理服务器中注册该管理节点。

要安装管理节点，请选择**自定义安装**，然后选择**将服务器安装为管理节点**。指定用于安装的端口。没有选项可供选择非 SSL 端口，因为管理服务器与管理节点之间的通信通过安全通道进行。安装过程中，将会询问您是否需要在管理服务器中注册该节点。如果选择在安装过程中不注册节点，可以使用 wadm CLI 来执行该操作。

注 - 通过快速安装不能安装管理节点。

2 在 Administration Server 中注册管理节点

要使管理节点成为群集或服务器群的一部分，必须在管理服务器中注册这些管理节点。管理节点只有在管理服务器中注册后才可以启动。要注册管理节点，请通过 wadm CLI 执行以下命令：

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

安装管理服务器过程中指定了此端口。host 是安装管理服务器的节点的主机名。

此操作将在管理服务器中注册该节点。

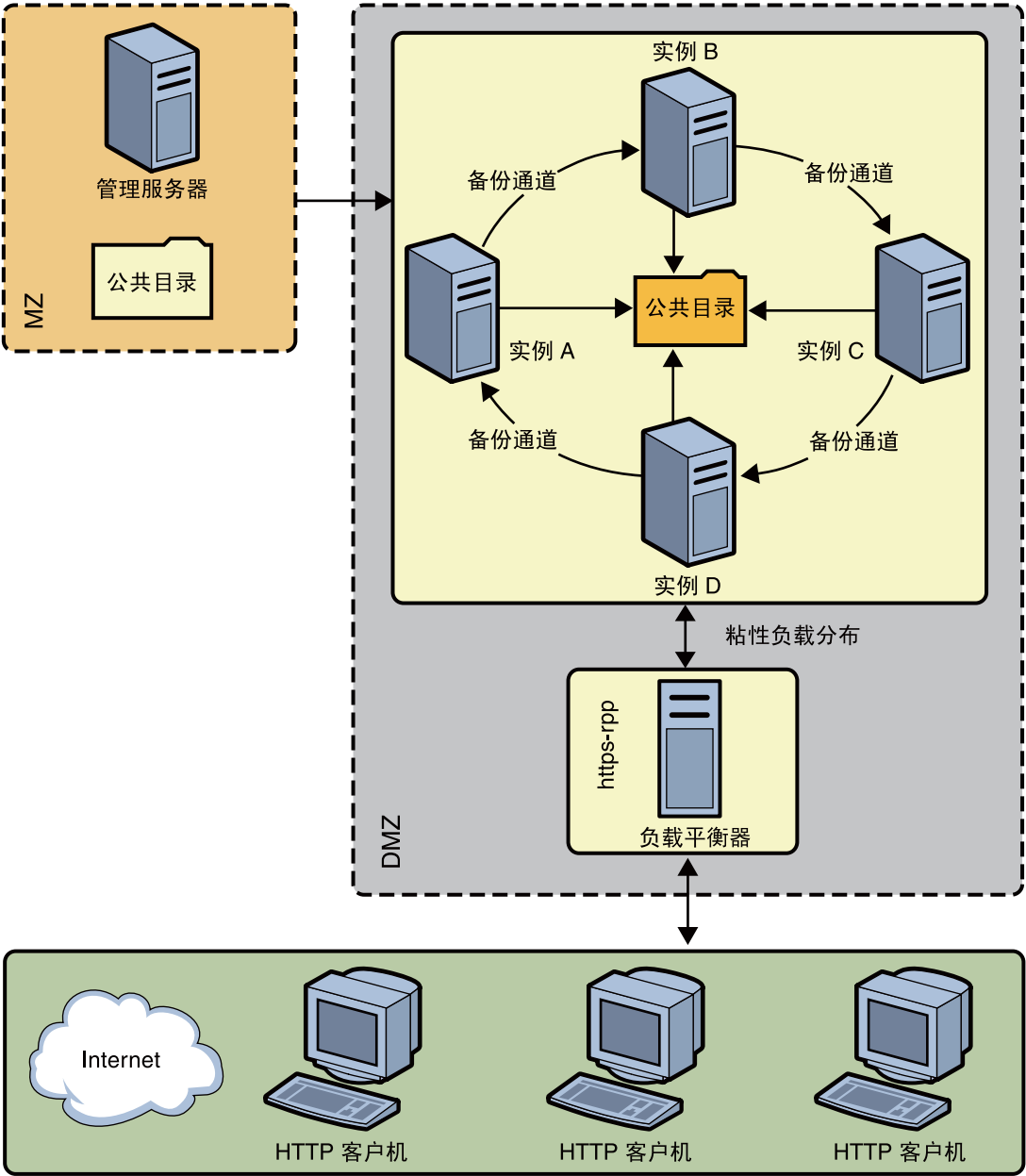
注 - 一个节点只可以通过同一节点进行注册。不能转到管理服务器的 CLI 并注册任何节点。此外，在管理服务器中注册节点只可以在 SSL 模式下进行。

要为此创建的群集设置会话复制，请参见第 167 页中的“配置会话复制”。

设置简单的群集

作为此示例的一部分，将设置具有一个负载均衡器、一个管理服务器和四个 Web 服务器实例（启用了会话复制）的一个群集。会话复制为 Java Web 应用程序会话提供高可用性。它通过将驻留在一个 Web 服务器实例的内存中的会话复制多份到另一个 Web 服务器实例来实现这一目的。因此在正常操作条件下，每个会话至少存在 2 份副本，每一份驻留在独立的 JVM 中，最好位于独立的计算机中。

下图描述了简单的群集：



▼ 配置群集

开始之前 标识以下计算机：

- MachineA — 具有负载均衡器和管理服务器。
- MachineB、MachineC、MachineD 和 MachineE — 具有管理节点且正在运行 Web 服务器实例。

1 在 MachineA 中安装 Administration Server。

有关安装管理服务器的信息，请参见第 46 页中的“设置服务器群”。典型的安装过程还将安装 Web 服务器实例。对于此方案，将不使用该实例。

2 在 MachineB、MachineC、MachineD 和 MachineE 中安装管理节点。

在所有 4 台计算机中安装管理节点。在管理服务器中注册这些管理节点。

3 配置 Web 应用程序。

对 Web 应用程序启用会话复制。按如下所示修改 WEB-INF/sun-web.xml 文件：

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

4 配置相应的实例。

- 启动 wadm。

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

- 为负载均衡器创建新的配置。

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

- 设置反向代理（负载均衡器）。

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

- 创建实例。

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

- 部署配置。

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

5 创建并启动群集。

创建具有四个实例的新配置。

- 为群集创建新配置。

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

- 启用会话复制。

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

- 添加 Web 应用程序。

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

- 创建实例。

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

- 启动群集。

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

注 – 如果未对 `start-instance` 命令指定主机名，此操作将在部署了配置的所有节点中启动实例。

部署方案

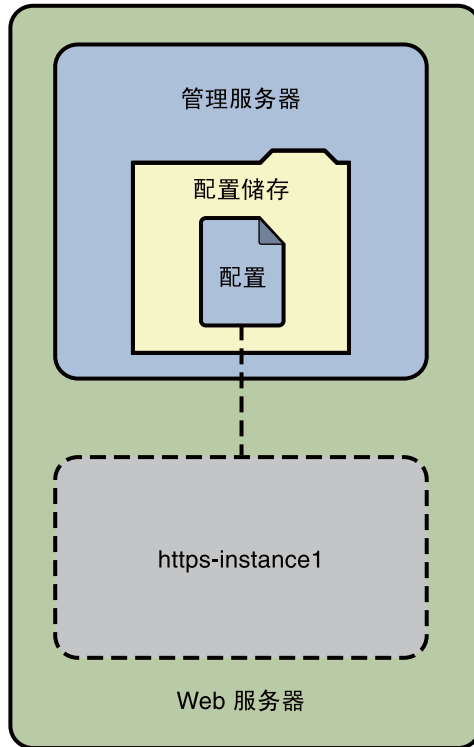
本章介绍如何在单个节点和群集环境中部署 Sun Java System Web Server 7.0。本章中将讨论以下主题：

- 第 51 页中的 “部署体系结构”
- 第 53 页中的 “部署概述”
- 第 56 页中的 “群集环境”
- 第 60 页中的 “会话复制”
- 第 63 页中的 “监视群集”
- 第 63 页中的 “Solaris Zones”

部署体系结构

本节介绍单个节点部署体系结构。

下图表示单个节点部署设置中的 Web Server。



在图中，Web Server 部署设置由以下部分组成：

- *Administration Server*—Administration Server 是专门配置的 Web 服务器实例。您可以在管理服务器中部署 Web 应用程序。
- **管理节点**—管理节点部署在服务器群的节点或服务器/主机中，可以与远程 Administration Server 通信。Administration Server 中可用的服务器配置可以部署到此节点。服务器群中的所有管理节点都必须是同源的。即，所有节点必须使用相同的操作系统并且具有相同的硬件体系结构。
- **配置**—配置是指 Web Server 实例的所有可配置的元素集，例如 Web 应用程序、配置文件和搜索集合索引。可以创建、修改或删除配置。Web Server 可以管理为配置创建多个配置实例。部署修改的配置将更新该配置的实例。
- *config-store*—它是基于文件系统的系统信息库，其中存储所有配置。



注意 – 不要编辑 `config-store` 目录中的任何文件。该目录中的文件由 Sun Java System Web Server 为内部使用而创建。

如果必须手动编辑 `config-store` 目录中的配置文件，请使用 `wadm deploy-config` 命令部署该配置。

有关使用此命令的更多信息，请参见《Sun Java System Web Server 7.0 CLI Reference Manual》。

- **实例** – 实例是指给定节点上 Web 服务器的环境，包括其配置、日志文件和其他运行时辅件，例如锁定数据库、高速缓存和临时文件。为了实现管理，可以启动、停止、重新启动或动态重新配置实例。

部署概述

对于以下目的，可以考虑在单个节点上部署 Web Server：

- 托管简单的 Web 或 CGI 应用程序。
- 开发和测试 Web 应用程序。

以下流程图提供了如何在节点上部署 Web Server 的示意说明：

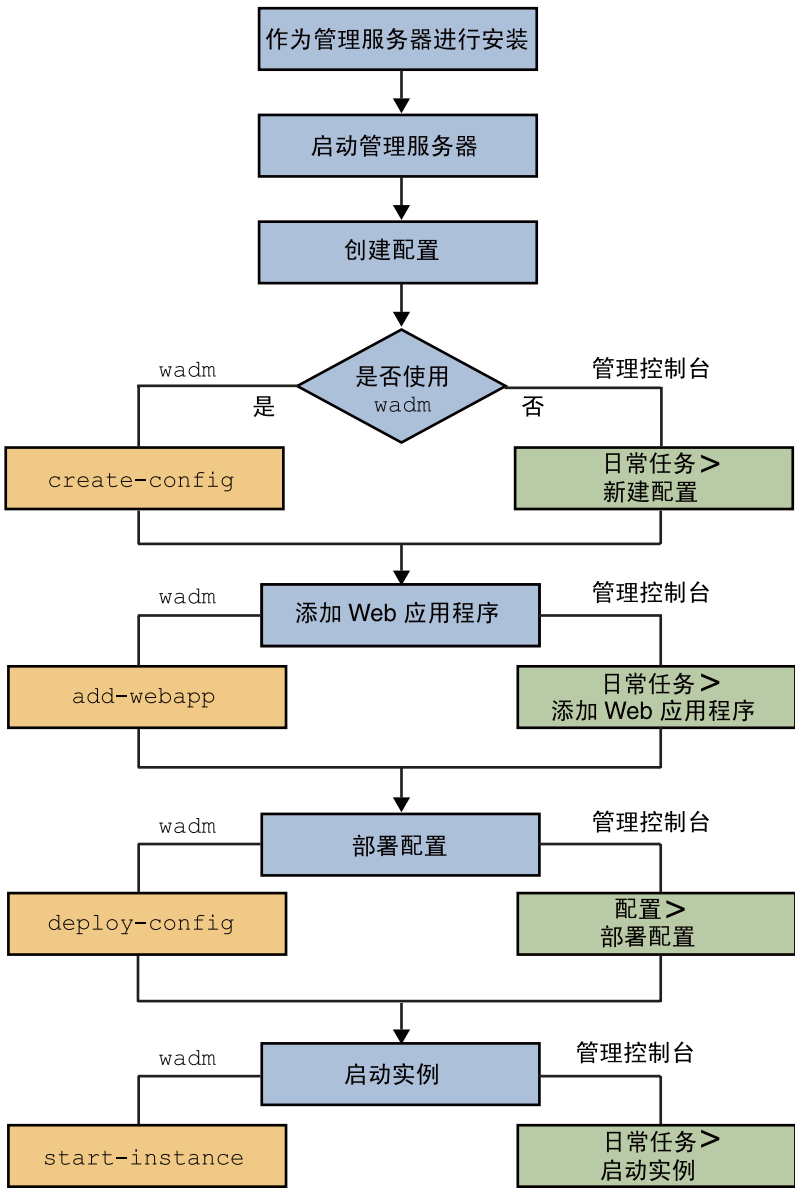


图 4-1 流程图说明了如何在单个节点上部署 Web 服务器

以下各节将介绍部署过程：

- 第 55 页中的 “部署前的要求”
- 第 55 页中的 “部署 Web Server”

部署前的要求

要在单个节点上部署 Web Server，可通过执行以下任务来准备系统：

1. 在节点上安装 Web Server。

如果在安装 Web Server 时选择“快速安装”选项，将会创建以下默认实体：

- Administration Server。
- 将创建具有一个 HTTP 侦听器的默认配置和虚拟服务器。配置和虚拟服务器的名称与主机名相同。
- 默认配置的实例。

有关安装 Web Server 的信息，请参见《Sun Java System Web Server 7.0 Installation and Migration Guide》中的第 2 章“Installing the Web Server”。

有关支持的平台和系统要求的信息，请参见《Sun Java System Web Server 7.0 发行说明》中的“支持的平台”。

2. 启动 Administration Server。

Administration Server 开始在指定的 SSL 端口上运行。

部署 Web Server

使用以下过程可在节点上部署 Web Server：

1. 您可以使用默认配置或创建新的配置。

如果要创建新的配置，请为配置指定唯一的名称。新配置将创建一个虚拟服务器和一个默认 HTTP 侦听器。

注 - 如果要使用管理控制台创建配置，向导将会提示您创建新的实例。如果要使用 CLI，必须使用 `create-instance` 命令显式创建该配置的实例。

所有配置都存储在 `<install_dir>/admin-server/` 目录下的 `config-store` 目录中。



注意 - 不要编辑 `config-store` 目录中的任何文件。该目录中的文件由 Sun Java System Web Server 为内部使用而创建。

2. 部署修改的配置。

群集环境

群集是多个服务器实例的集合，跨越多个节点，所有实例都运行相同的配置。群集中的所有实例配合工作，以提供高可用性、可靠性和可伸缩性。

借助负载平衡，群集通过提供故障转移和会话复制来提供不间断的服务和会话数据持久性。

硬件和软件要求

本节中介绍的 Web Server 群集使用案例由以下实体组成：

- 1) 四个实例（在四个相同的节点上运行）
- 2) 一个 Administration Server
- 3) 一个用于对 HTTP 请求进行负载平衡的反向代理

要设置群集，需要安装了相同的操作系统版本和修补程序的两个或多个相同节点。例如，如果选择使用 Solaris® 9 SPARC® 操作系统的计算机，群集中的其他计算机也必须安装 Solaris 9 SPARC。

有关支持的平台和修补程序要求的信息，请参见《Sun Java System Web Server 7.0 发行说明》。

下图描述了一个群集环境。

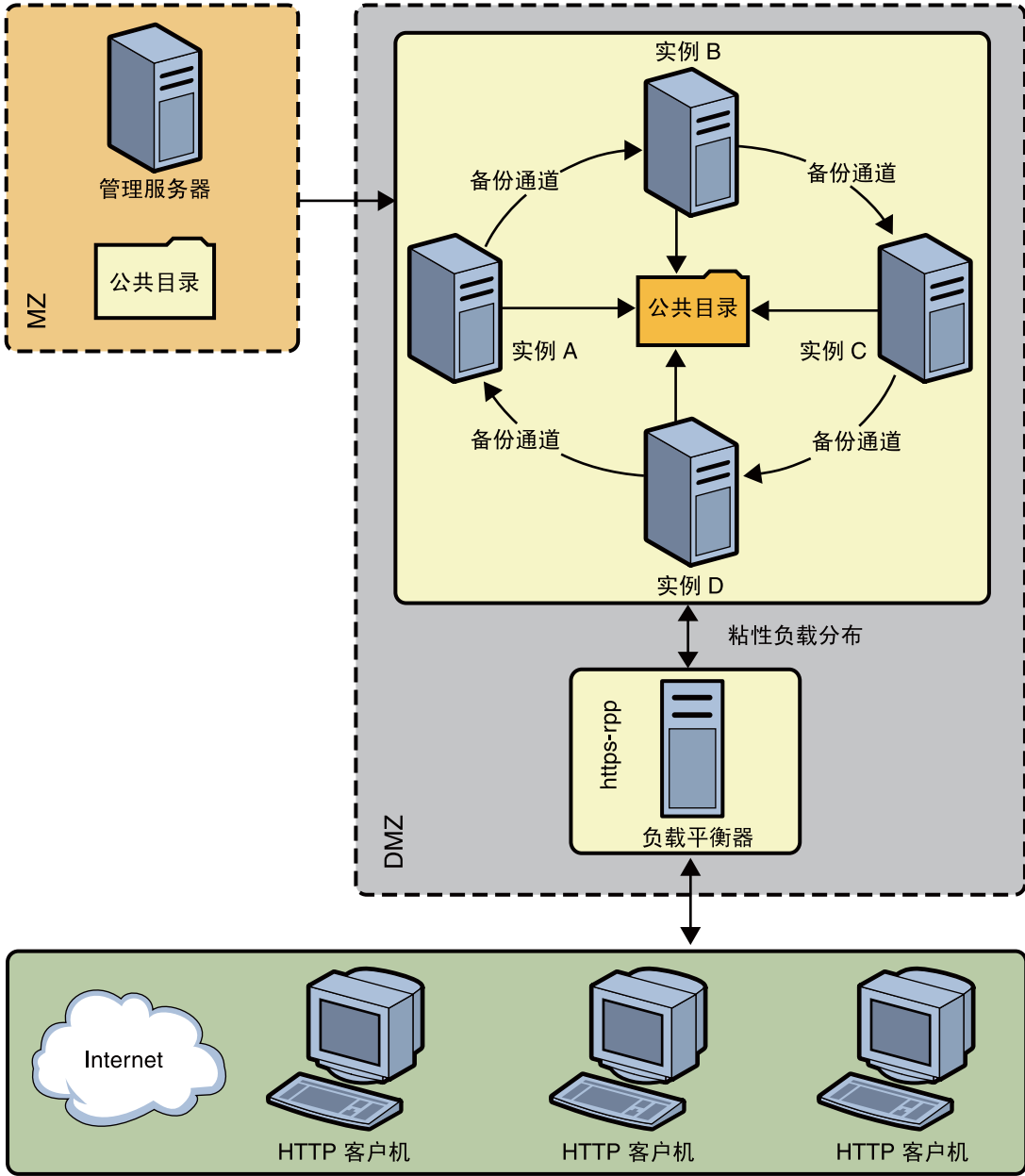


图 4-2 群集设置

在此图中，节点配置在非军事区 (De-Militarized Zone, DMZ) 中。Administration Server 则配置在防火墙，即军事区 (Militarized Zone) 的后面以限制和阻止对 Administration Server 进行常规访问。另一个节点配置为反向代理服务器。反向代理服务器驻留在 DMZ 中以增强安全性。

注 - 仅 Solaris 10 操作系统支持 Solaris zone 功能。

设置群集

本节介绍设置群集，以及启用反向代理以支持对 HTTP 请求进行负载平衡的过程。

以下流程图说明了设置群集的过程。

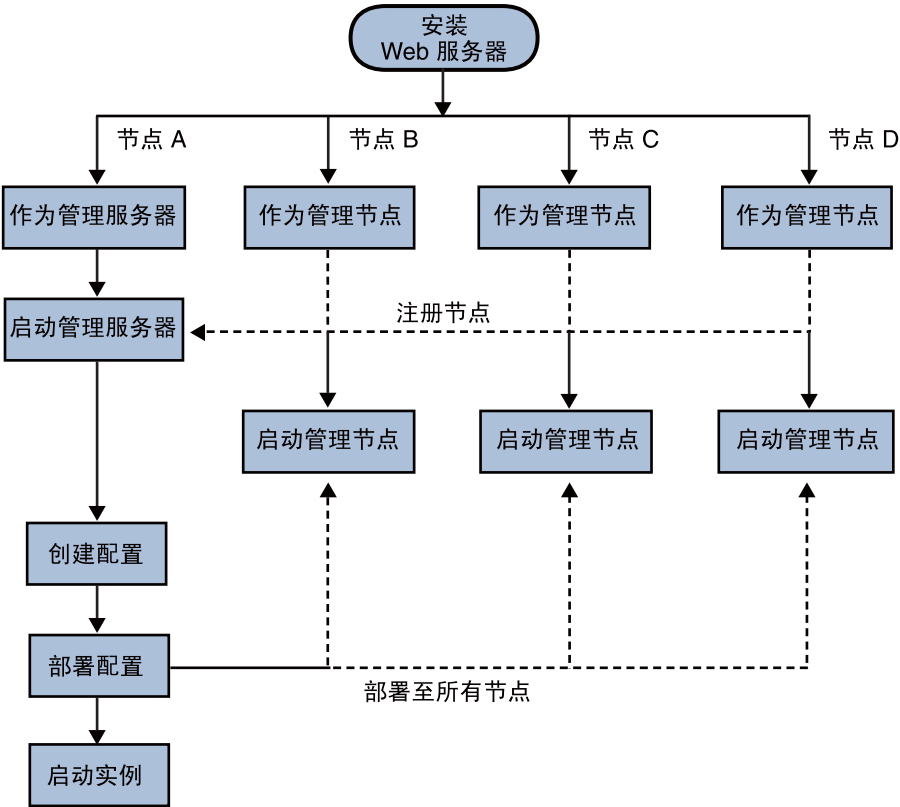


图 4-3 说明群集设置过程的流程图

1. 在一个节点上安装 Web Server，使其充当群集中的 Administration Server。

2. 在其他三个节点上安装 Web Server。选择将 Web Server 安装为管理节点的选项。在安装过程中，请选择在服务器中注册节点的选项。
3. 请确保 Administration Server 正在使用 SSL 端口进行通信，因为管理节点只能在安全模式下在服务器中注册。
4. 请确保安装了 Administration Server 和管理节点的所有节点上的系统日期和时间都相同。与服务器相关联的证书，是基于安装了 Administration Server 的节点的系统日期和时间来创建的。如果管理节点的系统日期早于 Administration Server，注册将失败，因为 Administration Server 的证书还未处于有效状态。因此，证书可能在已经过期的情况下被认为有效。
5. 从 `install_dir/admin-server/bin/` 目录中启动 Administration Server。
`install_dir/admin-server/bin>./startserv`
6. 从管理节点启动 wadm 命令行工具。wadm 命令行工具位于 `install_dir/bin` 目录中。
`install_dir/bin>./wadm`
7. 在 Administration Server 中注册每个管理节点。使用 `register-node` 命令可以在服务器中注册每个节点。
 例如：

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

其中，

`abc.sfbay.sun.com` 是将注册节点的 Administration Server 的主机名。

`port` 是 Administration Server 的 SSL 端口号。

8. 提示您输入管理密码时，请输入 Administration Server 的管理密码。

Administration Server 和管理节点相互验证，方法是 Administration Server 信任管理节点的服务器证书，管理节点信任由 Administration Server 提供的客户机证书。在注册管理节点的过程中，Administration Server 将生成该管理节点的服务器证书，然后可以在管理节点中下载并安装该证书。管理节点中还将安装服务器证书的颁发者。

注 - 只可以通过 SSL 进行注册。

有关注册节点的信息，请参见《Sun Java System Web Server 7.0 Installation and Migration Guide》中的“Registering the Administration Node From the Command-Line”。

9. 使用 `install_dir /admin-server/bin/` 目录中的 `startserv` 命令启动所有的管理节点。
10. 使用管理控制台或 CLI 在 Administration Server 中创建新的配置。
 提供新配置的配置信息，如配置名称、HTTP 侦听器端口和服务器名。
11. 在所有节点上创建配置的实例。

12. 在所有节点上启动实例。

注 – Web Server 提供了可灵活扩展或减小群集的功能。您可以随时在群集中添加或删除实例。

配置用于进行负载均衡的反向代理

Web Server 7.0 提供了完善的内置负载均衡器（即反向代理）。反向代理是服务器群中 Web Server 的网关。通过配置反向代理，请求将被转发到多个类似配置的 Web 服务器。

使用以下过程可在 Web Server 7.0 中启用反向代理：

1. 在要用于配置反向代理的节点上安装 Web Server。
2. 创建配置。例如，`rp`。
3. 使用管理控制台选择“配置” > “虚拟服务器” > “内容处理” > “反向代理”选项卡。单击“新建”按钮。
4. 输入反向代理 URI 和由逗号分隔的群集中所有计算机的服务器 URL。
输入服务器 URL 的格式为 *hostname:portnumber*。
5. 保存所进行的更改。
6. 部署修改的配置，以将更改应用于配置。
7. 启动此修改的配置的所有实例。

这将结束用于对 HTTP 请求进行负载均衡的反向代理的配置。

会话复制

会话复制是一种机制，用于复制存储在不同实例的会话中的数据。但是，所复制的实例必须为相同群集的一部分。在群集环境中启用会话复制时，将会复制所复制的实例中的完整会话数据。但是，会话复制操作不会复制会话中不可序列化的属性和任何特定于实例的数据。

会话复制和负载均衡为 Web 应用程序提供了很好的故障转移功能。

会话复制和故障转移操作

本节将详细介绍会话复制操作。

Web 请求结束时，Web Server 将确定是否需要通过会话复制配置（存储在服务器配置文件 `server.xml` 中）来复制会话数据。

假设存在由四个实例构成一个群集的一个使用案例，并且 Administration Server 中启用了会话复制。

由四个节点上运行的四个实例（A、B、C 和 D）构成的 Web Server 群集中的会话复制过程如下：

- 实例 A 是 D 的备份，B 是 A 的备份，C 是 B 的备份，D 是 C 的备份。这构成一个完整的备份环。
- 群集中的每个实例记录群集中所有实例的一个静态列表和一个活动备份实例。
- 根据配置，每次请求结束时，会话数据会被同步发送到备份实例。

Web Server 群集环境中的故障转移过程如下：

- 如果实例 A 失败，负载均衡器会将目标为实例 A 的所有传入 Web 请求重定向到群集中的其他实例，并按如下所示重新配置备份环：
 - D 检测出其备份 A 出现故障，将选择有序列表中 A 的下一个实例作为其新的备份实例。
 - 在此情况下，B 将被选择，D 将建立与 B 的新备份连接。B 现在拥有两个备份：A 的只读备份和 D 的活动备份。
 - 备份环现在变为完整，即 B 备份到 C，C 备份到 D，D 备份到 B。
 - 当出现故障的实例 A 再次可用时，它通过向其指定的备份实例 B 发送一条重新加入消息，来重新加入备份环，并建立与 B 的备份连接。
 - D 通过接收来自 A 的成功 ping 返回或者来自 A 的消息，检测出 A 已联机。
 - D 然后建立与 A 的备份连接，并终止其与 B 的备份连接。

Web Server 7.0 不支持会话复制中的以下功能：

- 恢复两个或多个实例同时出现的故障。
- 两次故障间的时间间隔必须大于复活的实例完全恢复所需的时间。
- 多个实例的会话备份。在常规操作中，任何会话只能有两份副本：主会话和备份会话。
- 会话持久性：要进行故障转移，只能在另一个实例的内存中备份会话。
- Web Server 仅支持 Java Web 应用程序的会话复制。如果使用的是非 Java 应用程序（如 CGI 或 PHP），则不能复制会话数据。

启用会话复制

可以使用管理控制台或 CLI 在群集中启用会话复制。在启用会话复制之前，请确保浏览器启用了 cookie。

server.xml 文件包含与会话复制相关的信息。以下所示为启用了会话复制的一个 server.xml 文件样例：

```
<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>
```

如果未对以下元素使用默认值，这些元素的条目将在 server.xml 配置文件中不可用：

Port number（默认值为 1099）
Protocol（默认值为 jrmf）
Encrypted（默认值为 false）
Getattribute Triggers Replication（默认值为 true）
Replica Discovery MaxHops（默认值为 -1）
Startup Discovery Timeout（默认值为 ?）
Cookie Name（默认值为 CLUSTERSESSIONLOCATOR）

有关这些会话复制属性的更多信息，请参见《Sun Java System Web Server 7.0 Administrator's Configuration File Reference》。

配置 Web 应用程序进行会话复制

要使服务器复制会话，必须为 Web 应用程序启用会话复制。

1. 要为 Web 应用程序启用会话复制，请修改位于 <web-application>/WEB-INF 目录中的 sun-web.xml 配置文件。

需要在 sunweb.xml 中进行的修改如下：

将元素 <session-manager/> 更改为 <session-manager
persistence-type="replicated">

以下所示为启用了会话复制的一个 sun-web.xml 文件样例：

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

2. 修改 `sunweb.xml` 文件后，请重新生成 Web 应用程序，或对应用程序重新进行 jar 操作以创建 Web 应用程序归档（war 文件）。
3. 重新启动所有实例，以使 Web 应用程序在所有实例中可用。
4. 该 Web 应用程序可从群集中的所有节点进行访问。要访问该 Web 应用程序，请在浏览器中键入以下内容：

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

注 - 目录可以为所有节点访问是存储要进行部署的应用程序的最佳方法。但是，该目录对 Administration Server 来说，无需可访问。建议对大小超过 1 MB 的 Web 应用程序进行基于目录的部署。

要创建搜索集合，请确保该搜索集合驻留在所有节点可以访问的公共目录中。

监视群集

Administration Server 可以监视群集中的所有实例。Web Server 的监视功能提供可用于以下操作的运行时组件和进程的状态信息：

- 确定性能瓶颈
- 调整系统以获取最优性能
- 帮助进行功能规划
- 预测故障
- 出现故障时执行根源分析

Solaris Zones

Solaris Zones 是 Solaris 10 的应用程序和资源管理功能。区域环境通常由以下资源组成，例如进程管理、内存、网络配置、文件系统、软件包注册、用户帐户、共享库和某些情况下安装的程序。这些区域提供了在 Solaris 实例中创建虚拟操作系统环境的方法，从而使一个或多个进程可以在与系统中的其他活动相互隔离的情况下运行。它还提供了一个抽象层，用于将应用程序与部署应用程序的计算机的物理属性（如物理设备路径、网络接口名称和网络路由表）相隔离。此隔离可防止给定区域中运行的进程监视或影响其他区域中运行的进程，无论用户 ID 和其他凭证信息是什么。

区域是一个或多个应用程序可以在其中运行的**沙箱**，不会影响或干涉系统的其他部分。

有关 Solaris Zones 的详细信息，请参见 <http://docs.sun.com/app/docs/doc/817-1592> 网站上的 *System Administration Guide — Solaris Containers-Resource Management and Solaris Zones*。

使用虚拟服务器

- 第 65 页中的 “虚拟服务器概述”
- 第 65 页中的 “使用案例”
- 第 67 页中的 “管理虚拟服务器”
- 第 69 页中的 “配置 HTTP 侦听器”

虚拟服务器概述

使用虚拟服务器时，您只需安装一台服务器便可以为多个公司或个人提供域名、IP 地址以及某些服务器监视功能。对于用户来说，他们就像拥有了自己的 Web 服务器，只不过是提供硬件并进行基本的 Web 服务器维护。

所有虚拟服务器都需要指定 HTTP 侦听器。当一个新请求传入时，服务器将根据配置的 HTTP 侦听器确定将该请求发送到哪个虚拟服务器。

使用案例

Sun Java System Web Server 的服务器实例可以拥有很多的 HTTP 侦听器，无论是安全的还是不安全的均可。您可以使用基于 IP 地址的虚拟服务器，也可以使用基于 URL 主机的虚拟服务器。

每个虚拟服务器都可以（而非必须）拥有自己的 ACL 列表、mime.types 文件和 Java Web 应用程序集。

此设计提供了最大的灵活性，使您可以为各种应用程序配置服务器。以下示例介绍了一些可能适用于 Sun Java System Web Server 的一些配置。

默认配置

安装新的 Sun Java System Web Server 后，您便拥有了一个服务器实例。它只使用一个 HTTP 侦听器以侦听计算机配置的任何 IP 地址的端口 80 或安装时所选的任何端口。

本地网络中的某些机制为计算机配置的每个地址都建立了名称—地址映射。以下示例中的计算机有两个网络接口：地址 127.0.0.1 上的回送接口（即使在没有网卡的情况下仍然存在的接口）和地址 10.0.0.1 上的以太网接口。

名称 `example.com` 通过 DNS 映射为 10.0.0.1。侦听套接字配置为侦听计算机配置的任何地址的端口 80（"ANY:80" 或 "0.0.0.0:80"）。

在此配置中，到以下地址的连接将到达服务器并由虚拟服务器 VS1 处理。

- `http://127.0.0.1/`（在 `example.com` 上启动）
- `http://localhost/`（在 `example.com` 上启动）
- `http://example.com/`
- `http://10.0.0.1/`

此配置适用于传统的 Web 服务器。无需添加附加的虚拟服务器或 HTTP 侦听器。

安全服务器

请参见第 83 页中的“为服务器配置 SSL”。

内部网托管

一个更复杂的 Sun Java System Web Server 配置是服务器在内部网部署中托管几个虚拟服务器。例如，假设您有三个内部站点，员工可在这些站点中查找其他用户的电话号码、查看校园地图以及跟踪发送到信息服务部门的请求的状态。以前（在本示例中），这些站点托管在三个不同的计算机中，这些计算机映射为名称 `phone.example.com`、`maps.example.com` 和 `is.example.com`。

为了将硬件和管理开销减少到最低程度，用户可以将这三个站点合并为计算机 `example.com` 上的一个 Web 服务器。可以使用以下两种方法进行此设置：使用基于 URL 主机的虚拟服务器或使用单独的 HTTP 侦听器。两者都有明显的优点和缺点。

使用基于 URL 主机的虚拟服务器的内部网托管

尽管基于 URL 主机的虚拟服务器易于设置，但它们具有以下缺点：

- 在此配置中支持 SSL 需要使用通配符证书进行非标准设置。
- 基于 URL 主机的虚拟服务器不能与传统的 HTTP 客户机一起使用。

您也可以设置基于 IP 地址的配置，每个地址一个 HTTP 侦听器：

使用单独的 HTTP 侦听器的内部网托管

基于 IP 地址的虚拟服务器的优点包括：

- 它们可以与不支持 HTTP/1.1 Host 标头的旧客户机一起使用。
- 提供 SSL 支持比较简便。

缺点包括：

- 它们要求更改主机上的配置（实际或虚拟网络接口的配置）。
- 它们不能缩放至具有上千个虚拟服务器的配置。

两种配置都要求为三种名称设置名称到地址的映射。在基于 IP 地址的配置中，每个名称都映射为不同的地址。必须对主机进行设置，才能接收所有这些地址上的连接。在基于 URL 主机的配置中，所有名称都可以映射为同一地址（计算机最初具有的地址）。

具有多个 HTTP 侦听器的配置所带来的性能增益可能最少，因为服务器不必查明传入请求的地址。但是由于其他接收方线程，使用多个 HTTP 侦听器也将导致额外开销（内存和调度）。

海量托管

海量托管是一个可以启用许多低通信量虚拟服务器的配置。例如，ISP（托管许多低通信量的个人主页）便属于海量托管。

虚拟服务器通常基于 URL 主机。例如，可能有一个配置只允许静态内容，而另一个配置允许静态内容和 CGI。

管理虚拟服务器

- [第 67 页中的“添加虚拟服务器”](#)
- [第 68 页中的“配置虚拟服务器”](#)
- [第 69 页中的“复制虚拟服务器”](#)

添加虚拟服务器

▼ 添加虚拟服务器

- 开始之前
- 您应该已创建/识别需要通过其创建虚拟服务器的配置。
 - 您应该已创建/识别 HTTP 侦听器。
 - 您应该已识别新虚拟服务器的主机。

- 1 选择需要从中添加虚拟服务器的配置。您可以从“配置”选项卡中显示的配置列表选择配置。

- 2 单击“虚拟服务器”选项卡>“新建”按钮。
- 3 此时将显示一个弹出式向导页面，此页面将指导您完成虚拟服务器的配置过程。在此向导页面中，请执行以下步骤：
 - 输入新虚拟服务器的信息：
 - a. 输入一个可标识新虚拟服务器的名称。此名称由字母数字字符组成，也可以包含句点(.)、破折号(-)和下划线(_)字符。
 - b. (可选)输入要添加至新虚拟服务器的主机列表。
 - c. (可选)输入虚拟服务器的文档根目录。
 - 为新配置的虚拟服务器选择一个 HTTP 侦听器。您可以选择现有的 HTTP 侦听器，也可以创建新的 HTTP 侦听器。
- 4 您将会看到向导摘要页面。要更改配置，请单击“上一页”返回到先前的页面。单击完成以完成新虚拟服务器的配置过程。
- 5 此时将显示结果页面。如果发现任何错误，请返回至向导中的前几个页面，再次配置虚拟服务器。

注-使用 CLI

要通过 CLI 添加虚拟服务器，请执行以下命令：

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

请参见 CLI 参考 create-virtual-server(1)。

配置虚拟服务器

- 第 68 页中的“配置虚拟服务器”

要配置虚拟服务器的常规设置，请执行以下任务：

▼ 配置虚拟服务器

- 1 选择配置。

从配置列表中选择相应的配置。单击“配置”选项卡以获取可用配置的列表。
- 2 选择虚拟服务器。

从虚拟服务器列表中选择虚拟服务器。单击“虚拟服务器”选项卡为选定配置获取可用的虚拟服务器。

- 3 单击“常规”选项卡。配置以下设置。
 - 已启用—是否在运行时启用虚拟服务器。
 - 文档根目录—虚拟服务器的文档根目录路径，其中存储虚拟服务器的数据。这包括展开的 Web 应用程序目录和日志文件。
 - 主机—您可以输入多个 URL 主机，之间用逗号分隔。

复制虚拟服务器

要复制虚拟服务器，请执行以下步骤：

▼ 复制虚拟服务器

- 1 选择配置。
从配置列表中选择相应的配置。单击“配置”选项卡以获取可用配置的列表。
- 2 选择虚拟服务器。
从虚拟服务器列表中选择虚拟服务器。单击“虚拟服务器”选项卡为选定配置获取可用的虚拟服务器。
- 3 单击“复制”按钮。
为新的虚拟服务器提供名称。

注 - 使用 CLI

要通过 CLI 复制虚拟服务器，请执行以下命令：

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

copiedVS 是新的虚拟服务器的名称。

请参见 CLI 参考 `copy-virtual-server(1)`。

配置 HTTP 侦听器

- [第 70 页中的“创建 HTTP 侦听器”](#)
- [第 70 页中的“配置 HTTP 侦听器”](#)

服务器在将请求转发给配置的虚拟服务器之前，通过 HTTP 侦听器接受 HTTP 请求。此页面允许您添加和配置 HTTP 侦听器。

HTTP 侦听器必须具有唯一的端口号和 IP 地址组合。您可以使用 IPv4 地址或 IPv6 地址。将 IP 地址设置为 "*", 可以创建在该端口上侦听所有 IP 地址的 HTTP 侦听器。

创建 HTTP 侦听器

要为虚拟服务器创建新的 HTTP 侦听器以处理传入的 HTTP 请求，请执行以下步骤：

1. 单击“配置”选项卡下的**虚拟服务器**选项卡。
2. 单击 **HTTP 侦听器**子选项卡以查看配置的 HTTP 侦听器列表。
3. 单击**新建**按钮将弹出用于创建新 HTTP 侦听器的向导页面。

在向导页面中提供以下信息。

- **名称**—新 HTTP 侦听器的名称。
- **端口**—绑定和侦听传入 HTTP 请求的 HTTP 侦听器的端口。
- **IP 地址**—有效的 IPv4 或 IPv6 地址。 "*" 表示 HTTP 侦听器将侦听配置端口上的所有 IP 地址。
- **服务器名**—输入服务器名，例如 *sales.mycomp.com*。
- **默认虚拟服务器**—从下拉式列表中选择虚拟服务器。此操作将为选定的虚拟服务器关联新的 HTTP 侦听器。
- **描述（可选）**—输入 HTTP 侦听器的简短描述。

注—使用 CLI

要通过 CLI 创建 HTTP 侦听器，请执行以下命令：

```
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
```

请参见 CLI 参考 `create-http-listener(1)`。

配置 HTTP 侦听器

要编辑现有的 HTTP 侦听器设置，请执行以下步骤：

1. 单击“服务器配置”下的**虚拟服务器**选项卡以编辑现有的 HTTP 侦听器设置。
2. 单击 **HTTP 侦听器**子选项卡以查看配置的 HTTP 侦听器列表。
3. 在**侦听器名称**表列下，单击需要编辑其设置的 HTTP 侦听器。

您可以为 HTTP 侦听器编辑常规设置和与安全性相关的设置。

修改 HTTP 侦听器参数

单击**常规**选项卡以编辑基本和高级 HTTP 侦听器设置。配置以下选项：

- **名称**—新 HTTP 侦听器的名称。
- **端口**—绑定和侦听传入 HTTP 请求的 HTTP 侦听器的端口。
- **IP 地址**—有效的 IPv4 或 IPv6 地址。"*" 表示 HTTP 侦听器将侦听配置端口上的所有 IP 地址。
- **服务器名**—输入服务器名，例如 *sales.mycomp.com*。

在**高级**部分下选择**配置高级设置**选项，以编辑 HTTP 侦听器高级设置。配置以下选项：

- **接收方线程**—专用于接受此侦听器所接收的连接线程数。接受的值为 1 至 128。
- **协议系列**—侦听器使用的协议。请不要修改此值。默认值为 HTTP。
- **侦听队列大小**—操作系统待办事项侦听队列的最大大小。
- **接收缓冲区大小**—操作系统套接字接收缓冲区的大小（以字节为单位）。
- **发送缓冲区大小**—操作系统套接字发送缓冲区的大小（以字节为单位）。
- **分块 I/O**—确定 HTTP 侦听器套接字是否正处于分块模式。默认情况下，禁用该选项。

证书和密钥

本章介绍了如何使用证书和密钥验证来确保 Sun Java System Web Server 的安全性。还介绍了如何激活用于保护您的数据、拒绝入侵者访问和允许所需的访问的各种安全功能。

在阅读本章之前，您应该已经熟悉公钥加密的基本概念。这些概念包括加密和解密、公钥和私钥、数字证书以及加密协议。

- 第 73 页中的 “使用证书进行验证”
- 第 74 页中的 “证书密钥类型”
- 第 76 页中的 “创建自签名证书”
- 第 76 页中的 “管理证书”
- 第 81 页中的 “管理证书撤销列表 (Certificate Revocation List, CRL)”
- 第 82 页中的 “设置内部令牌的密码”
- 第 83 页中的 “为服务器配置 SSL”
- 第 83 页中的 “为配置启用 SSL 加密算法”
- 第 84 页中的 “为 HTTP 侦听器启用安全性”

使用证书进行验证

验证是确认身份的过程。在网络交互环境中，验证是一方对另一方身份确认的过程。证书是支持验证的一种方法。

证书中包含的数字数据指定了个人、公司或其他实体的名称，并证明证书中包含的公钥属于该实体。客户机和服务器都可以拥有证书。

证书由证书授权机构（即 CA）颁发并进行数字签名。CA 可以通过 Internet 销售证书的公司，也可以是负责为您公司的内联网或外部网颁发证书的部门。您可以将您充分信任的 CA 确定为其他用户身份的检验器。

除了公钥和由证书标识的实体名称之外，证书还包括到期日期、颁发该证书的 CA 的名称和颁发该证书的 CA 的“数字签名”。

注 – 在激活加密之前必须安装服务器证书。

服务器验证

服务器验证指客户机对服务器进行的信任识别；即对被认为要对位于特定网络地址的服务器负责的组织进行识别。

客户机验证

客户机验证指服务器对客户机进行的信任识别；即对被认为要使用客户机软件的用户识别。客户机可以有多个证书，如同一个人可以有几个不同的身份一样。

证书密钥类型

除 RSA 密钥之外，Sun Java System Web Server 7.0 还引入了对椭圆曲线加密算法 (Elliptic Curve Cryptography, ECC) 的支持。

ECC 为新兴算法，因为与传统密码学系统（如 RSA）相比，它是更富有吸引力的公钥加密系统，它使用更小的密钥大小，但可提供同样的安全性，这使计算更快，降低了能耗，并节省了内存和带宽。椭圆曲线加密算法 (Elliptic Curve Cryptography, ECC) 已被美国政府认可。

现在，可以选择是使用 RSA 密钥还是 ECC 密钥来生成证书请求或自签名证书。

对于 RSA 密钥，可以提供不同的密钥大小（密钥大小越大，加密效果越好。默认密钥大小为 1024）。对于 ECC 密钥，您应选择要用哪种曲线生成密钥对。各个组织（ANSI X9.62、NIST、SECG）已指定多种曲线，Sun Java System Web Server 7.0 支持当前指定的所有曲线。

如果打算从 CA 请求证书（而不使用自签名证书），请务必先与首选 CA 联系，获取有关 ECC 用法的最新信息。就您的情况，询问他们是否推荐使用特定的 ECC 曲线。如果您的 CA 或组织内部策略并未提供选择曲线的指导，请参考以下给出的某些建议。请记住，ECC 是一种新兴技术，所以此文件正在编写时，特定使用情况的曲线建议可能已经有所更改。

以下列出了一些支持的 ECC 曲线：

```
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

创建自签名证书

如果不需要 CA 签署的证书，或者希望在 CA 签署证书的同时测试新的 SSL 实现，则可以生成自签名证书。此临时证书将在客户机浏览器中生成错误，以表明签署证书的授权机构未知且不受信任。

要通过 CLI 创建自签名证书，请执行以下命令：

```
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
```

请参见 CLI 参考 `create-selfsigned-cert(1)`。

管理证书

- 第 76 页中的“请求证书”
- 第 78 页中的“安装证书”
- 第 79 页中的“更新证书”
- 第 80 页中的“删除证书”

请求证书

证书中包含的数字数据指定了个人、公司或其他实体的名称，并证明证书中包含的公钥属于该个人。启用 SSL 的服务器必须具有证书，客户机有无证书均可。

证书由证书授权机构（即 CA）颁发并进行数字签名。CA 可能是通过 Internet 销售证书的公司，也可以是负责为您公司的内联网或外部网颁发证书的部门。您可以将您充分信任的 CA 确定为其他用户身份的检验器。

您可以请求证书并将其提交到证书授权机构 (Certificate Authority, CA)。如果公司有自己的内部 CA，则可以向其申请证书。如果打算从商业 CA 处购买证书，请选择一个 CA 并索要所需的特定格式信息。您还可以为服务器创建自签名证书。自签名证书不适用于面向 Internet 的部署，但对于开发和测试却很有帮助，因为它们使您在无需 CA 介入的情况下即可设置测试服务器。

如上所述，证书包含实体（此例中为 Web 服务器）的公钥。公钥是根据特定的算法而生成（证书中还对该算法类型进行了编码）。下一节介绍有关 Web Server 对其密钥支持的算法类型的背景。

▼ 请求证书

1 单击“服务器证书”选项卡>“请求”按钮。

2 选择配置

从需要为其安装证书的配置列表选择一个配置。

3 选择令牌

选择包含密钥的令牌（加密设备）。如果密钥存储在由 Sun Java System Web Server 7.0 维护的本地密钥数据库中，请选择内部令牌。如果密钥存储在智能卡或者其他外部设备或引擎中，请从下拉式列表框中选择外部令牌的名称。输入选定令牌的密码。

4 输入详细信息

开始请求过程之前，请确保您了解 CA 所需的信息。无论是从商业 CA 还是内部 CA 请求服务器证书，都需要提供以下信息：

- **服务器名**必须是在 DNS 查找中使用的全限定主机名（例如，`www.sun.com`）。这是浏览器用于连接至您站点的 URL 中的主机名。如果这两个名称不匹配，客户机将收到证书名称与站点名称不匹配的通知，并怀疑您的证书的真实性。
如果从内部 CA 申请证书，也可以在此字段中输入通配符或正则表达式。多数供应商都不会批准在通用名称中输入通配符或正规表达式的证书请求。
- **组织**是公司、教育机构、合作伙伴等的正式而合法的名称。多数 CA 需要您使用法律文档（例如营业执照副本）来验证此信息。
- **组织单位**是用于说明公司中组织的可选字段。也可以用于标注不太正式的公司名称（不带 *Inc.*、*Corp.* 等等）。
- **地址**是通常用于说明组织所在的城市、公国的可选字段。
- **省/自治区/直辖市**为可选字段。
- **国家/地区**是必填的，即您所在国家/地区名称的两个字符的缩写（以 ISO 格式）。美国的国家代码为 US。

所有这些信息组合为一系列属性值对（称为标识名 [DN]），用于形成证书的主题。

5 选择证书选项

您必须提供密钥信息。您可以选择 RSA 或 ECC 密钥类型。如果密钥类型为 RSA，则密钥大小可以为 1024、2048 或 4098。如果密钥类型为 ECC，则您还需要选择曲线。请记住，生成新的密钥对需要花费时间。密钥长度越长，向导生成它所需的时间越长。



注意 – 请务必选择 CA（稍后会向其提交签署请求）可以支持的密钥类型。

6 选择证书类型

为自签名证书或 CA 签署的证书选择证书签署授权机构 (Certificate Signing Authority, CSA)。如果选择自签名证书，则还可以关联证书的 HTTP 侦听器。也可以在以后执行此操作。

7 生成请求

如果是 CA 签名证书，生成的证书请求将以 ASCII 格式提供。如果是自签名证书，则生成的证书将直接安装。如果为自签名类型，请为昵称、有效期（月份）和 HTTP 侦听器名称提供相应的值以便处理安全请求。

8 查看结果

此页面为您提供了选定选项的摘要。单击“完成”可以完成请求的生成。

注 - 使用 CLI

要通过 CLI 请求证书，请执行以下命令：

```
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
```

请参见 CLI 参考 `create-cert-request(1)`。

安装证书

从 CA 获取证书后，您可以使用管理控制台为配置安装此证书。

▼ 安装证书

1 单击“服务器证书”选项卡 > “安装”按钮。

2 选择配置

从需要为其安装证书的配置列表选择一个配置。

3 选择令牌

选择包含密钥的令牌（加密设备）。如果密钥存储在由 Sun Java System Web Server 7.0 维护的本地密钥数据库中，请选择内部令牌。如果密钥存储在智能卡或者其他外部设备或引擎中，请从下拉式列表框中选择外部令牌的名称。输入选定令牌的密码。

4 输入证书数据

将证书文本粘贴到所提供的文本区域中。复制并粘贴文本时，请确保包含标头 "Begin Certificate" 和 "End Certificate"，其中包含起始和终止连字符。您也可以单击“浏览”按钮并手动选择 .DER 文件。

5 提供证书详细信息

提供将用于此证书的昵称。从可用列表中选择 HTTP 侦听器以便处理安全请求。

6 查看结果

此页面为您提供了选定选项的摘要。单击“完成”可以完成安装过程。

注 - 使用 CLI

要通过 CLI 安装证书，请执行以下命令：

```
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 包含证书数据。

请参见 CLI 参考 install-cert(1)。

更新证书

要更新现有证书，请执行以下步骤：

▼ 更新证书

1 单击“服务器证书”选项卡>“证书名”>“更新”按钮。

2 提供令牌信息

输入令牌密码（如果需要）。否则，单击“下一步”继续进行操作。

3 更新证书详细信息

查看证书详细信息，并提供有效期（以月份为单位）。

4 更新密钥信息

您可以选择 RSA 或 ECC 密钥类型。如果密钥类型为 RSA，则密钥大小可以为 1024、2048 或 4098。如果密钥类型为 ECC，则您还需要选择曲线。请记住，生成新的密钥对需要花费时间。

5 查看摘要

此页面为您提供了选定选项的摘要。单击“完成”以完成更新过程。

删除证书

要删除证书，请执行以下任务：

▼ 删除证书

1 单击“服务器证书”选项卡。

2 选择证书

从证书列表中选择相应的证书名。

3 删除证书

单击“删除”按钮以删除选定证书。

注-使用 CLI

要通过 CLI 删除证书，请执行以下命令：

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

请参见 CLI 参考 `delete-cert(1)`。

更新 Administration Server 证书

要更新管理服务器证书，请执行 CLI 命令 `renew-admin-certs`。使用此命令可更新昵称为 `Admin-CA-Cert`、`Admin-Server-Cert` 和 `Admin-Client-Cert` 的管理认证。此命令还可以更新当前正在运行且使用更新的证书可以访问的节点。

执行此命令后，建议重新启动管理服务器和节点，以便新证书能够生效。如果在更新证书的过程中，节点处于脱机状态（未运行或由于网络问题无法访问），必须重新注册该节点。要更新管理服务器证书，请执行以下命令：

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

请参见 CLI 参考 `renew-admin-certs(1)`。

管理证书撤销列表 (Certificate Revocation List, CRL)

证书撤销列表 (Certificate Revocation List, CRL) 用于列出客户机或服务器用户不应再信任的所有证书和密钥。如果证书中的数据发生变化（例如，用户在证书到期之前换单位或离职），该证书将被撤销，并且其数据将显示在 CRL 中。CRL 由 CA 生成并定期更新。

▼ 安装 CRL

要从 CA 获得 CRL，请执行以下步骤：

- 1 从 CA 获取文件形式的 CRL。
- 2 转至管理控制台中的配置页面。
- 3 单击“证书”>“证书授权机构”选项卡。
- 4 单击“安装 CRL”按钮
- 5 输入关联文件的完整路径名。
- 6 单击“确定”。

注 - 如果数据库中已存在 CRL，则将显示“替换证书撤销列表”页面。

- 7 您可能需要单击“部署”才可以使更改生效。

注 - 使用 CLI

要通过 CLI 安装 CRL，请执行以下命令：

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

请参见 CLI 参考 install-crl(1)。

▼ 删除 CRL

- 1 转至管理控制台中的配置页面。
- 2 单击“证书”>“证书授权机构”选项卡。

- 3 选择 CRL 条目并单击“删除”。
- 4 您可能需要单击“部署”才可以使更改生效。

注 - 使用 CLI

要通过 CLI 删除 CRL，请执行以下命令：

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

请参见 CLI 参考 delete-crl(1)。

设置内部令牌的密码

要设置内部 PKCS11 令牌的密码，请执行以下操作：

▼ 设置令牌密码

- 1 转至管理控制台中的配置页面。
- 2 单击“证书”>“PKCS11 令牌”选项卡。
- 3 单击 PKCS11 令牌名称（默认为内部）。
- 4 选中“令牌状态”复选框。
- 5 输入密码信息。
- 6 如果不想在启动实例时提示输入密码，请选中“不要在启动实例时提示输入新密码”复选框。单击“确定”。
- 7 密码将保存在配置中。要删除密码，请执行上述步骤，并选择“未设置密码”选项。

注 - 使用 CLI

要通过 CLI 为内部 PKCS11 令牌设置密码，请执行以下命令：

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

请参见 CLI 参考 set-token-pin(1)。

为服务器配置 SSL

使用命令 `create-cert-request` 生成请求并将请求发送到 CA。以后，在从 CA 收到证书时，则需要使用 `install-cert` 命令来安装证书。如果 Java 密钥库中包含希望迁移的密钥和证书，请使用 `migrate-jks-keycert` 命令。对于开发/测试服务器，最便捷的实现方式是使用 `create-selfsigned-cert` 命令来生成自签名证书。

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

有关更多选项和示例，请查看手册页。

安装证书后，需要在将启用 SSL 的某端口上创建侦听器。

```
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
```

现在，为该侦听器启用 SSL 并将侦听器与证书昵称关联。

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

完成此设置后，部署配置并启动实例。

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

为配置启用 SSL 加密算法

要为配置启用 SSL 加密算法，请执行以下命令：

```
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
```

请参见 CLI 参考 `enable-ciphers(1)`。

为 HTTP 侦听器启用安全性

注 - 仅当已安装可用的证书时才能为 HTTP 侦听器启用安全性。

具有证书后，您可以将此证书与 HTTP 侦听器关联以保护服务器。

加密是转换信息的过程，因此除预期的收件人外，此过程对其他人来说意义不大。解密是转换加密信息的过程，因此，此过程又再次变得有意义。Sun Java System Web Server 支持 SSL 和 TLS 协议。

加密算法是一种用于加密或解密的密码学算法（一种数学函数）。SSL 和 TLS 协议包含多个加密算法套件。某些加密算法比其他加密算法更强大且更安全。一般而言，加密算法使用的位越多，将数据解密越难。

在任何双向加密过程中，双方都必须使用相同的加密算法。由于可以使用多种加密算法，因此需要让服务器使用最常用的加密算法。

在安全连接过程中，客户机和服务器都同意使用可以进行通信的最强大的加密算法。您可以从 SSL2、SSL3 和 TLS 协议中选择加密算法。

注 - 因为在 SSL 2.0 版本之后对 SSL 的安全性和性能进行了各种改进，所以除非客户机无法使用 SSL 3，否则不要使用 SSL 2。使用 SSL 2 加密算法无法为客户机证书提供保证。

单独的加密过程并不足以确保服务器机密信息的安全。使用加密算法的同时还必须使用密钥，以便生成真正的加密结果，或解密以前加密的信息。加密过程使用以下两种密钥获得此结果：公钥和私钥。使用公钥加密的信息只能使用关联的私钥进行解密。公钥作为证书的一部分发布，因此只有关联的私钥受到保护。

Sun Java System Web Server 支持用于加密通信的安全套接字层 (Secure Sockets Layer, SSL) 和传输层安全 (Transport Layer Security, TLS) 协议。SSL 和 TLS 是独立于应用程序的，并且可以在它们上面透明地分层运行更高级别的协议。

SSL 和 TLS 协议支持各种加密算法，用于服务器和客户机的相互验证、传输证书和建立会话密钥。客户机和服务器可以支持各种加密算法套件或加密算法集合，这取决于各种因素，例如它们所支持的协议、公司有关加密强度的政策以及政府对加密软件出口的限制。在其他函数中，SSL 和 TLS 握手协议将确定服务器和客户机如何协商以决定将来将进行通信的加密算法套件。

单击 **配置 > HTTP 侦听器 > 安全性** 选项卡，可以编辑 HTTP 侦听器安全性设置。下表列出了您可以在此页面中进行配置的属性。

表 6-1 HTTP 侦听器安全性属性

属性	说明
名称	HTTP 侦听器的名称。
安全性	启用/禁用选定 HTTP 侦听器的安全性。
证书	从可用证书中选择服务器证书。必须已安装 RSA 或 ECC 证书才可以执行此操作。
客户机验证	指定客户机验证是必要的还是可选的。选择 "False" 选项可以禁用客户机验证。
验证超时	超时值，超过该时间后，客户机验证握手就会失败。[0.001–3600]。默认值为 60 秒。
最大验证数据量	进入缓冲区的最大验证数据量。[0–2147.0483647.0]。默认值为 104857.06。
SSL 版本 2/SSL 版本 3	启用/禁用 SSL 版本 2/SSL 版本 3。
TLS	启用/禁用 TLS。默认情况下将启用检测版本回滚。这将会配置服务器，使其检测人为版本回滚攻击尝试。为了实现与某些未正确实现 TLS 规范的客户机的互操作性，可能需要禁用 TLS。
SSL3/SSL2/TLS 加密算法	<p>要保护 Web 服务器的安全性，应启用 SSL。您可以启用 SSL 2.0、SSL 3.0 和 TLS 加密协议并选择各种加密算法套件。可以在侦听套接字上为管理服务器启用 SSL 和 TLS。在侦听套接字上为 Server Manager 启用 SSL 和 TLS 将为所有与该侦听套接字关联的虚拟服务器设置安全首选项。</p> <p>默认设置允许使用最常用的加密算法。除非有充分的理由不使用特定的加密算法套件，否则应全部启用。</p>

控制对服务器的访问

您可以通过多种安全服务和机制（包括验证、授权和访问控制）来保护驻留在 Web 服务器上的资源。本章介绍了某些用于控制对 Sun Java System Web Server 7.0 的访问的支持机制。

- 第 87 页中的 “什么是访问控制”
- 第 88 页中的 “访问控制的工作原理”
- 第 89 页中的 “为用户/组设置访问控制”
- 第 93 页中的 “为主机/IP 设置访问控制”
- 第 93 页中的 “配置 ACL 用户高速缓存”
- 第 94 页中的 “配置访问控制”
- 第 98 页中的 “使用 .htaccess 文件”
- 第 99 页中的 “防止服务器遭受拒绝服务攻击”

什么是访问控制

验证是确认身份的过程。授权表示将受限制资源的访问权限授予某个身份，而访问控制机制则强制执行这些限制。许多安全模型（Web 应用程序安全性、htaccess、验证领域等）和服务可以强制执行验证和授权。

通过访问控制可以确定：

- 可以访问 Administration Server 的用户
- 这些用户可以访问的应用程序
- 可以访问您 Web 站点上的文件或目录的用户

您可以控制对整个服务器、部分服务器或您 Web 站点上的文件或目录的访问。您应创建称为访问控制条目 (access control entry, ACE) 的规则分层结构，以允许或拒绝访问。创建的 ACE 的集合称为访问控制列表 (access control list, ACL)。

默认情况下，服务器具有一个 ACL 文件，其中包含多个 ACL。在为某个传入请求确定要使用的虚拟服务器后，Sun Java System Web Server 将检查是否为该虚拟服务器配置了任何 ACL。如果找到适用于当前请求的 ACL，服务器将评估其 ACE 以确定是否允许访问。

是否允许访问将基于以下事项：

- 谁发出请求（用户/组）
- 请求来自何处（主机/IP）
- 请求发生的时间（例如，一天中的某个时间）
- 使用的连接类型 (SSL)

访问控制的工作原理

当服务器收到对某个页面的请求时，将使用 ACL 文件中的规则来确定是否允许访问。这些规则可以引用发送该请求的计算机的主机名或 IP 地址，还可以引用 LDAP 目录中存储的用户和组。

注 - 如果有多个匹配的 ACL，服务器将使用最后一个匹配的 ACL 语句。由于 uri ACL 是最后一个匹配的语句，因此 default ACL 将被忽略。

下图描述了访问控制在 Web Server 7.0 中的工作原理。用户代理（客户机）访问 Web



Server。Web Server 执行 `obj.conf` 文件中的 `PathCheck` 指令。Web Server 向客户机返回 HTTP 401（未授权）错误。客户机提示用户进行验证。如果客户机是浏览器，则会弹出一个登录对话框。用户输入登录信息。Web Server 执行内部的 `check-acl` 函数。Web Server 验证用户凭证并处理请求。

为用户/组设置访问控制

您可以仅允许特定的用户或组访问您的 Web 服务器。用户/组访问控制要求用户输入用户名和密码，然后才能访问服务器。服务器会将客户机证书中的信息与目录服务器条目进行比较。

Administration Server 只使用基本验证。如果希望在 Administration Server 上进行客户机验证，必须手动编辑 ACL 文件，将方法更改为 SSL。

用户/组验证由 Web Server 通过读取用户组数据库中的条目来执行。目录服务用来实现访问控制的信息可能来自以下资源之一：

- 内部平面文件类型数据库
- 外部 LDAP 数据库

当服务器使用基于 LDAP 的外部目录服务时，对于服务器实例它支持以下类型的用户/组验证方法：

- 默认
- 基本
- SSL
- 摘要
- 其他

当服务器使用基于文件的内部目录服务时，对于服务器实例它支持的用户/组验证方法包括：

- 默认
- 基本
- 摘要

用户/组验证要求用户验证其自身，然后才能访问服务器或 Web 站点上的文件和目录。借助验证，用户可使用客户机证书并通过输入用户名和密码来验证其身份。只有 SSL 通信才需要客户机证书。

默认验证

默认验证是首选方法。“默认”设置使用 `server.xml` 文件中的默认方法，或使用“基本”方法（如果 `server.xml` 中没有设置）。如果选中“默认”，ACL 规则将不会在 ACL 文件中指定方法。选择“默认”后，您便可通过编辑 `obj.conf` 文件中的一行文本来轻松更改所有 ACL 的方法。

基本验证

基本验证要求用户输入用户名和密码来访问 Web 服务器或 Web 站点。这是默认设置。您必须在 LDAP 数据库（如 Sun Java System Directory Server）或文件中创建和存储用户和组列表。所使用的目录服务器不能与 Web 服务器安装在相同的服务器根目录下；您也可以使用安装在远程计算机上的目录服务器。

当用户试图访问 Administration Server 或 Web 站点中具有用户/组验证的资源时，Web 浏览器将显示一个对话框，要求用户输入用户名和密码。服务器将收到加密或未加密的信息，这取决于您的服务器是否启用了加密。

注 - 如果使用不带 SSL 加密的基本验证，将在网络中以未加密的文本形式发送用户名和密码。网络包可能会被截取，并且用户名和密码可能会被盗用。基本验证在与 SSL 加密、主机/IP 验证或二者组合时最有效。使用摘要验证可以避免此类问题。

SSL 验证

使用安全性证书，服务器可以用两种方式确认用户的身份：

- 使用客户机证书中的信息作为身份的证明
- 验证 LDAP 目录中发布的客户机证书（附加验证）

当您为服务器设置使用证书信息来验证客户机时，服务器将：

- 首先检查证书是否来自一个信任的 CA。如果不是，验证将失败，事务也将结束。
- 如果证书来自一个信任的证书授权机构 (certificate authority, CA)，则使用 `certmap.conf` 文件将证书映射到某个用户的条目。
- 如果证书正确进行了映射，则检查为该用户指定的 ACL 规则。即使证书正确进行了映射，ACL 规则也可能会拒绝该用户的访问。

要求对特定资源的访问控制进行客户机验证与要求对服务器的所有连接进行客户机验证不同。如果将服务器设置为要求对所有连接进行客户机验证，则客户机只需要提供由信任的 CA 颁发的有效证书。如果将服务器的访问控制设置为使用 SSL 方法来验证用户和组，则客户机需要：

- 提供由信任的 CA 颁发的有效证书
- 证书必须映射到 LDAP 中的有效用户
- 访问控制列表必须进行正确评估

要求对访问控制进行客户机验证时，需要为 Web 服务器启用 SSL 加密算法。

要成功访问要求进行 SSL 验证的资源，客户机证书必须来自 Web 服务器信任的 CA。如果 Web 服务器的 `certmap.conf` 文件被配置为将浏览器中的客户机证书与目录服务器中的客户机证书相比较，则需要在目录服务器中发布客户机证书。不过，`certmap.conf` 文件也可以配置为仅将证书中的选定信息与目录服务器条目进行比较。例如，您可以配置 `certmap.conf` 文件，使其仅将浏览器证书中的用户 ID 和电子邮件地址与目录服务器条目进行比较。

注 - 只有 SSL 验证方法才需要修改 `certmap.conf` 文件，因为将对照 LDAP 目录对证书进行检查。而要求对服务器的所有连接进行客户机验证则不必如此。如果选择使用客户机证书，则应增加 `magnus.conf` 中 `AcceptTimeout` 指令的值。

摘要验证

可以将服务器配置为使用基于 LDAP 或文件的目录服务执行摘要验证。

通过摘要验证，用户可以基于用户名和密码进行验证，而不必以明文形式发送用户名和密码。浏览器使用用户的密码和 Web 服务器提供的某些信息，利用 MD5 算法来创建摘要值。

当服务器使用基于 LDAP 的目录服务来执行摘要验证时，服务器端将使用摘要验证插件来计算该摘要值，并且将该值与客户机提供的摘要值进行比较。如果这些摘要值相匹配，用户将通过验证。要进行这种验证，目录服务器需要访问明文形式的用户密码。Sun Java System Directory Server 具有一个可逆的密码插件，它使用对称的加密算法以加密形式存储数据，这些数据可在稍后被解密成原来的形式。只有 Directory Server 保存了数据的密钥。

对于基于 LDAP 的摘要验证，需要启用服务器附带的可逆密码插件和特定的摘要验证插件。要配置 Web 服务器以处理摘要验证，请设置 `dbswitch.conf` 中数据库定义的 `digestauth` 属性。

如果未指定 ACL 方法，当要求进行验证时，服务器将使用摘要验证或基本验证；当不要求进行验证时，服务器将使用基本验证。这是首选方法。

表 7-1 摘要验证的不同情况

ACL 方法	验证数据库支持摘要验证	验证数据库不支持摘要验证
"default" 未指定	摘要和基本	基本
"basic"	基本	基本
"digest"	摘要	错误

处理 `method = digest` 的 ACL 时，服务器将尝试通过以下操作进行验证：

- 检查 Authorization 请求标头。如果未找到，将生成要求进行摘要验证的 401 响应，并且进程将停止。
- 检查 Authorization 类型。如果验证类型是摘要验证，服务器将：
 - 检查 `nonce`。如果无效，将刷新此服务器生成的 `nonce` 并生成 401 响应，且进程将停止。如果已过期，将生成 `stale=true` 的 401 响应，且进程将停止。

可以通过更改位于 `server_root/https-server_name/config/` 下 `magnus.conf` 文件中 `DigestStaleTimeout` 参数的值，来配置 `nonce` 保持刷新的时间。要设置该值，请将下面一行文本添加到 `magnus.conf` 中：

`DigestStaleTimeout seconds`

其中 `seconds` 表示 `nonce` 将保持刷新的秒数。指定的秒数过后，`nonce` 将到期并要求用户进行新的验证。

- 检查领域。如果领域不匹配，将生成 401 响应，且进程将停止。
- 如果验证目录是基于 LDAP 的，则检查 LDAP 目录中的用户是否存在；如果验证目录是基于文件的，则检查文件数据库中的用户是否存在。如果未找到，将生成 401 响应，且进程将停止。
- 从目录服务器或文件数据库获取请求/摘要值，并检查是否与客户机的请求/摘要值匹配。如果不匹配，将生成 401 响应，且进程将停止。

- 构造 Authorization-Info 标头并将其插入服务器标头中。

为主机/IP 设置访问控制

您可以通过将 Administration Server 或 Web 站点上的文件和目录仅限于使用特定计算机的客户机使用，从而限制对它们的访问。您可以指定要允许或拒绝其访问的计算机的主机名或 IP 地址。可以使用通配符模式指定多台计算机或整个网络。使用主机/IP 验证访问文件或目录对用户可能是无缝的。用户可以立即访问文件和目录，而无需输入用户名或密码。

由于可能会有多个用户使用某个特定计算机，因此主机/IP 验证在与用户/组验证组合使用时更有效。如果同时使用这两种验证方法，访问时将要求提供用户名和密码。

主机/IP 验证不要求在服务器上配置 DNS。如果选择使用主机/IP 验证，必须在网络中运行 DNS 并将服务器配置为使用该 DNS。您可以通过 Server Manager 中“首选项”选项卡的“性能调节”页面，在服务器上启用 DNS。

启用 DNS 会降低服务器的性能，因为服务器将被强制执行 DNS 查找。要减少 DNS 查找对服务器性能的影响，请仅解析访问控制和 CGI 的 IP 地址，而不要解析所有请求的 IP 地址。要实现此目的，请将 `iponly=1` 添加到 `obj.conf` 文件的 `AddLog fn="flex-log" name="access"` 中：

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

配置 ACL 用户高速缓存

默认情况下，服务器将用户和组验证结果存储在 ACL 用户高速缓存中。您可以使用 `magnus.conf` 文件中的 `ACLCacheLifetime` 指令来控制 ACL 用户高速缓存的有效时间。每次引用高速缓存中的条目时，都会计算条目生存期并对照 `ACLCacheLifetime` 进行检查。如果其生存期大于或等于 `ACLCacheLifetime`，则不会使用该条目。默认值为 120 秒。将该值设置为 0（零）将关闭高速缓存。如果将其设置为一个较大的值，则每次更改 LDAP 条目时，都可能需要重新启动服务器。例如，如果将该值设置为 120 秒，则在长达两分钟的时间内，服务器可能会与 LDAP 目录不同步。仅当 LDAP 目录不经常更改时才应设置一个较大的值。

使用 `magnus.conf` 参数 `ACLUserCacheSize`，可以配置可存储在高速缓存中的最大条目数。此参数的默认值为 200。新条目将添加到列表的开头，当高速缓存达到其最大大小时，列表末尾的条目将被删除以便容纳新条目。

您还可以使用 `magnus.conf` 参数 `ACLGroupCacheSize` 来设置每个用户条目所能高速缓存的最大组成员数。此参数的默认值为 4。遗憾的是，组中非成员关系的用户不会被高速缓存，这将导致每个请求都要进行多个 LDAP 目录访问。

有关 ACL 文件指令的更多信息，请参见《*NSAPI Developer's Guide*》。

设置 ACL 高速缓存属性

要通过 CLI 设置 ACL 高速缓存属性，请执行以下命令：

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

请参见 CLI 参考 `set-acl-cache-prop(1)`。

可以设置的有效属性包括：

- `enabled`—指示服务器是否高速缓存文件内容和元信息。默认值为 `true`。
- `max-age`—高速缓存文件内容和元信息的最长时间数（以秒为单位）。值的范围从 0.001 至 3600。
- `max-groups-per-user`—对于每个用户而言，服务器将高速缓存其成员信息的最大组数。值的范围从 1 至 1024。
- `max-age`—高速缓存验证信息的最长时间数（以秒为单位）。值的范围从 0.001 至 3600。

配置访问控制

服务器通过使用本地存储的访问控制列表 (access control list, ACL) 来支持验证和授权，这些列表用于说明用户对资源所具有的访问权限。例如，ACL 中的一个条目可以授予名为 John 的用户对特定文件夹 `misc` 的读取权限。

本节介绍如何限制对您 Web 站点上的文件或目录的访问。您可以为所有服务器设置全局访问控制规则，也可以单独为特定服务器进行设置。例如，人力资源部门可以创建 ACL，允许所有通过验证的用户查看他们自己的工资单数据，但是只允许负责工资单的人力资源人员更新数据。

服务器支持的核心 ACL 有三种类型的验证：基本验证、SSL 验证和摘要验证。

要编辑访问控制设置，请执行以下步骤：

1. 单击**配置**选项卡并选择相应的配置。
2. 单击**安全性**子选项卡 > **访问控制**子选项卡。
3. 单击**添加 ACL**按钮添加新的 ACL，或单击现有 ACL 以编辑设置。

添加访问控制列表 (Access Control List, ACL)

下节介绍将新 ACL 添加至配置的过程。

1. 单击**配置**选项卡并选择相应的配置。
2. 单击**访问控制**子选项卡 > **访问控制列表**子选项卡。

3. 单击**新建**按钮添加新的 ACL。

配置以下参数：

表 7-2 ACL 参数

参数	说明
资源	命名/URI/路径。选择需要设置访问限制的资源类型，并指定其值。URI 资源示例—"/sales"。路径资源示例—"/usr/sun/server4/docs/cgi-bin/*"。
验证数据库	验证数据库 使您可以选择服务器用于验证用户的数据库。 默认值为 keyfile 。
验证方法	<ol style="list-style-type: none">1. 基本—使用 HTTP 基本方法从客户机获取验证信息。仅当为服务器启用了 SSL 后，才会通过网络对用户名和密码进行加密。2. SSL—使用客户机证书来验证用户。要使用此方法，必须为服务器启用 SSL。启用加密后，可以结合使用“基本”和"SSL"方法。3. 摘要—使用一种验证机制，使得浏览器能够基于用户名和密码进行验证，而不必以明文形式发送用户名和密码。浏览器使用用户的密码和 Web Server 提供的某些信息，利用 MD5 算法来创建摘要值。请注意，要使用“摘要”，基础 auth-db 也必须支持摘要。这表示要么是 使用摘要文件的文件 auth-db 要么是 LDAP auth-db（仅当安装了摘要验证插件的情况下）。4. 其他—使用通过访问控制 API 创建的自定义方法。
验证提示	验证提示 选项使您可以输入要在验证对话框中显示的消息文本。您可以使用此文本来说明用户需要输入的内容。基于不同的浏览器，用户大约会看到该提示的前 40 个字符。 Web 浏览器通常会高速缓存用户名和密码，并将它们与提示文本关联。当用户访问具有相同提示的服务器的文件和目录时，不必再次输入用户名和密码。如果您希望用户对特定文件和目录再次进行验证，只需更改该资源的 ACL 的提示。
拒绝的访问响应	指定拒绝对资源的访问时所进行的响应操作。 <ol style="list-style-type: none">1. 响应并显示默认消息—选择该选项可以显示从服务器返回的标准访问拒绝消息。2. 响应并显示 URL—选择该选项可以将请求转发至其他任何外部 URL 或错误页面。

注 – 使用 CLI

要通过 CLI 添加 ACL，请执行以下命令：

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

请参见 CLI 参考 set-acl(1)。

添加访问控制条目 (Access Control Entry, ACE)

本节介绍为选定的配置添加新访问控制条目 (Access Control Entry, ACE) 的过程。

1. 单击**配置**选项卡并选择相应的配置。
2. 单击**访问控制**子选项卡 > **访问控制列表**子选项卡。
3. 单击**新建**按钮。
4. 单击访问控制条目下的**新建**按钮。

配置以下 ACE 参数：

表 7-3 ACE 参数

参数	说明
访问	<ul style="list-style-type: none">■ 允许表示用户或系统可以访问请求的资源。■ 拒绝表示用户或系统不能访问该资源。 服务器将检查整个访问控制表达式 (ACE) 列表以确定访问权限。
用户	<ol style="list-style-type: none">1. 任意用户—无验证。将访问权限授予每个用户。2. 验证数据库中的所有用户—将访问权限授予在验证数据库中指定的所有用户。3. 仅验证数据库中的以下用户—仅允许验证数据库中的选定用户进行访问。 您可以基于通用属性（如名字、姓氏和电子邮件地址）查询验证数据库。
组	<p>使用组验证时，将提示用户输入用户名和密码，然后才允许其访问在访问控制规则中指定的资源。</p> <p>使用该选项可以只允许特定组的访问。</p>

表 7-3 ACE 参数 (续)

参数	说明
主机地址	<p>您可以基于请求来自的计算机限制对 Administration Server 或 Web 站点的访问。</p> <p>您可以基于请求来自的计算机限制对 Administration Server 或 Web 站点的访问。</p> <ul style="list-style-type: none">■ “任意位置” 允许所有用户和系统进行访问■ “仅以下位置” 仅允许特定主机名或 IP 地址进行访问 <p>如果选择“仅以下位置”选项，请在“主机名”或“IP 地址”字段中输入通配符模式或逗号分隔的列表。按主机名进行限制比按 IP 地址进行限制更灵活：如果用户的 IP 地址发生更改，将不需要更新此列表。但是，按 IP 地址进行限制更可靠：如果对某个连接的客户机的 DNS 查找失败，则无法使用主机名限制。</p> <p>您只能使用通配符模式的 * 通配符表示法来匹配计算机的主机名或 IP 地址。例如，要允许或拒绝特定域中的所有计算机，可以输入与该域中所有主机均匹配的通配符模式，例如 *.sun.com。您可以为访问 Administration Server 的超级用户设置不同的主机名和 IP 地址。</p> <p>对于主机名，* 必须替换名称中的整个部分。即，*.sun.com 有效，但 *users.sun.com 无效。当 * 出现在主机名中时，它必须是最左侧的字符。</p> <p>例如，*.sun.com 有效，但 users.*.com 无效。对于 IP 地址，* 必须替换地址中的整个字节。例如，198.95.251.* 有效，但 198.95.251.3* 无效。当 * 出现在 IP 地址中时，它必须是最右侧的字符。例如，198.* 有效，但 198.*.251.30 无效。</p>

表 7-3 ACE 参数 (续)

参数	说明
权限	<p>访问权限限制了对 Web 站点上的文件和目录的访问。除了允许或拒绝所有访问权限外，您还可以指定一个规则以允许或拒绝部分访问权限。例如，可以授予用户对文件的只读访问权限，这样他们可以查看信息，但不能更改文件。</p> <ul style="list-style-type: none">■ “所有访问权限”是默认设置，将允许或拒绝所有权限■ “仅以下权限”使您可以选择要允许或拒绝的权限组合：<ul style="list-style-type: none">■ 读取允许用户查看文件，其中包括 HTTP 方法 GET、HEAD、POST 和 INDEX■ 写入允许用户更改或删除文件，其中包括 HTTP 方法 PUT、DELETE、MKDIR、RMDIR 和 MOVE。要删除文件，用户必须同时具有写入权限和删除权限■ 执行使用户可以执行服务器端应用程序，例如 CGI 程序、Java applet 和代理■ 删除允许同时具有写入权限的用户删除文件或目录■ 列表使用户可以访问不包含 index.html 文件的目录中的文件列表。■ 信息使用户可以接收有关 URI 的信息，例如 http_head。
继续	<p>服务器将检查整个访问控制表达式 (access control expression, ACE) 列表以确定访问权限。例如，第一个 ACE 通常为拒绝每个用户。如果将第一个 ACE 设置为“继续”，服务器将检查列表中的第二个 ACE，如果该 ACE 匹配，则使用下一个 ACE。</p> <p>如果未选中“继续”，则会拒绝所有用户访问该资源。服务器将继续检查列表，直至找到某个不匹配的 ACE，或匹配但未被设置为“继续”的 ACE。最后一个匹配的 ACE 将确定是否允许访问。</p>

使用 .htaccess 文件

服务器支持 .htaccess 动态配置文件。您可以通过用户界面或手动更改配置文件来启用 .htaccess 文件。

您可以将 .htaccess 文件与服务器的标准访问控制结合起来使用。不管 PathCheck 指令的顺序如何，标准访问控制始终在任何 .htaccess 访问控制之前应用。如果用户/组验证为“基本”，无需同时使用标准访问控制和 .htaccess 访问控制进行用户验证。您可以通过标准服务器访问控制来使用 SSL 客户机验证，也可以通过 .htaccess 文件进行 HTTP “基本”验证。

如果启用 .htaccess 文件，服务器将在提供资源之前检查 .htaccess 文件。服务器在与资源相同的目录及其父目录，直到包含文档根目录的目录中查找 .htaccess 文件。例如，如果主文档目录设置为 /sun/server/docs，并且客户机请求

/sun/server/docs/reports/index.html，则服务器将在 /sun/server/docs/reports/.htaccess 和 /sun/server/docs/.htaccess 中检查 .htaccess 文件。

请注意，服务器的附加文档目录和 CGI 目录功能使管理员可以定义替代的文档根目录。替代的文档根目录的存在会影响 .htaccess 文件的处理。例如，假定服务器的主文档目录设置为 /sun/server/docs，并且 CGI 程序位于 /sun/server/docs/cgi-bin/program.cgi。如果启用 CGI 作为文件类型，则当客户机发出对 CGI 程序的请求时，服务器对 /sun/server/docs/.htaccess 和 /sun/server/docs/cgi-bin/.htaccess 的内容都要进行评估。但是，如果您选择在 /sun/server/docs/cgi-bin 配置 CGI 目录，服务器将只检查 /sun/server/docs/cgi-bin/.htaccess，而不检查 /sun/server/docs/.htaccess。出现这种情况的原因是：将 /sun/server/docs/cgi-bin 指定为 CGI 目录便将其标记为替代的文档根目录。

防止服务器遭受拒绝服务攻击

拒绝服务 (Denial-of-Service, DoS) 攻击是服务器的某些恶意用户发起的一种显式攻击，企图阻止合法用户使用服务。此类攻击可通过以下方式发起：

- 向服务器发送对特定 Web 资源的连续请求。

Sun Java System Web Server 可以通过监视经常访问的 URI 和拒绝的请求（如果请求频率相当高）来检测 DoS 攻击。

以下部分介绍了如何在虚拟服务器级别防止 DoS 攻击。

限制对服务器的请求

现在，您可以通过配置请求限制并监视每个虚拟服务器的最大连接数来调整服务器，以防止拒绝服务攻击。配置其中某些值可能会影响服务器的性能。

要配置服务器的请求限制，请单击 **配置 > 虚拟服务器 > 服务器设置 > 请求限制**，然后配置下表中所列的参数。

表 7-4 配置请求限制

参数	说明
请求限制	为此虚拟服务器启用/禁用请求限制。默认情况下，将会禁用“请求限制”选项。
最大连接数	此虚拟服务器允许的最大并行连接数。
RPS 最大值	每秒钟允许的来自客户机的最大请求数。

表 7-4 配置请求限制 (续)

参数	说明
RPS 计算间隔	计算每秒请求数 (request per second, RPS) 平均值的时间间隔。默认值为 30 秒。
继续条件	确定要使阻塞的请求类型再次可用于服务所必须满足的条件。 静音 —被拒绝的请求数必须降低为零（在后续间隔中），才可继续服务。 阈值 —被拒绝的请求比率必须降低到 RPS 阈值之下，才可继续服务。 默认值为阈值。
错误代码	要用于阻塞的请求的 HTTP 状态代码。默认代码为“HTTP 503—服务不可用”。
监视属性	要监视的可选请求属性。

注-使用 CLI

要通过 CLI 限制对服务器的请求，请执行以下命令：

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

请参见 CLI 参考 enable-request-limits(1)。

▼ 限制最大连接数

您可以限制最大并行连接数。如果在处理的请求数至少为指定请求数时收到一个匹配的请求，将拒绝该请求。请注意，拒绝请求仅适用于该特定时间。一旦并行请求数小于此限制，便会立即处理新请求。

- 1 单击“配置”选项卡。
- 2 从列表中选择配置。
- 3 在“虚拟服务器”选项卡下选择虚拟服务器。
- 4 单击“服务器设置”>“请求限制”。
- 5 在“最大连接数”部分中输入值。

管理用户和组

本章介绍如何添加、删除和编辑可以访问 Sun Java System Web Server 的用户和组。

- 第 101 页中的 “访问用户和组的信息”
- 第 101 页中的 “关于目录服务”
- 第 102 页中的 “了解标识名 (Distinguished Name, DN)”
- 第 103 页中的 “使用 LDIF”
- 第 103 页中的 “使用验证数据库”
- 第 105 页中的 “设置用户和组”
- 第 107 页中的 “静态组和动态组”

访问用户和组的信息

使用 Administration Server 可以访问应用程序数据，例如用户帐户、组列表、访问权限 (ACL)、组织单位以及用户和组的其他特定信息。

用户和组信息以文本格式存储在平面文件中或支持轻量目录访问协议 (Lightweight Directory Access Protocol, LDAP) 的目录服务器（如 Sun Java System Directory Server）中。LDAP 是一种运行在 TCP/IP 上的开放式目录访问协议，它可以扩展到全局规模，甚至上百万个条目。

关于目录服务

通过目录服务器（如 Sun Java System Directory Server），可以从单个应用程序管理所有用户信息。您还可以配置目录服务器，以便允许用户从多个易于访问的网络位置检索目录信息。

在 Web Server 7.0 中，可以配置三种不同类型的目录服务来验证并授权用户和组。如果没有配置其他目录服务，新创建的目录服务的值将被设置为 `default`，而不管其类型为何。

创建目录服务时，将使用目录服务详细信息更新 `server.xml` 文件。

目录服务的类型

Web Server 7.0 支持的各种目录服务类型包括：

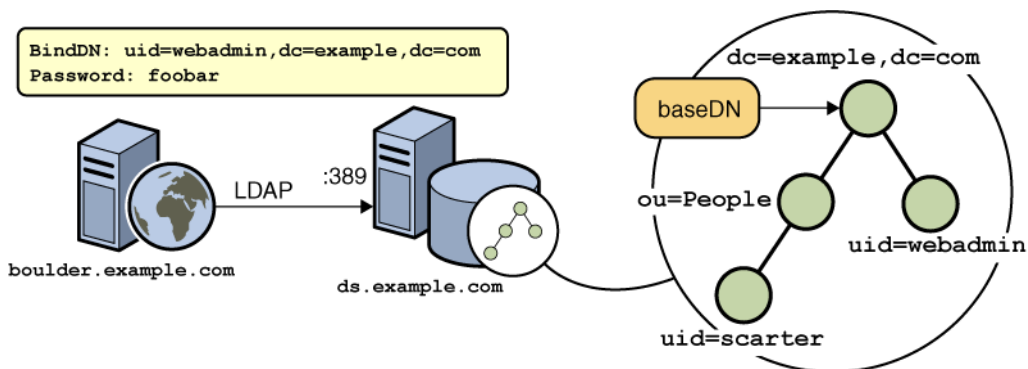
- **LDAP**—在基于 LDAP 的目录服务器中存储用户和组信息。
- **密钥文件**—密钥文件是一个文本文件，其中包含散列格式的用户密码以及该用户所属的组的列表。存储在密钥文件中的用户和组仅供 `file` 领域用来进行授权和验证，而与系统用户和组无关。
仅当要使用 HTTP 基本验证时才能使用密钥文件格式。
- **摘要文件**—基于加密的用户名和密码存储用户和组信息。
摘要文件格式用于支持使用 HTTP 摘要验证。但是，这种文件格式也支持基本验证，因此可以用于两种验证方法。

注 - 如果要设置分布式管理，默认的目录服务必须是基于 LDAP 的目录服务。

了解标识名 (Distinguished Name, DN)

用户是 LDAP 数据库中的个人，如公司的员工。组是共享某个通用属性的两个或多个用户。组织单位是公司内部的一个分支。

企业中的每个用户和组都通过标识名 (Distinguished Name, DN) 属性来表示。DN 属性是一个文本字符串，其中包含关联的用户、组或对象的标识信息。每当您更改用户或组目录条目时，就需要使用 DN。例如，每次为应用程序（如邮件或发布）创建或修改目录条目、设置访问控制以及设置用户帐户时，均需要指定 DN 信息。



上图显示了一个 DN 表示样例。以下示例显示了 Sun Microsystems 公司某个员工的典型 DN：

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

该示例中每个等号前面的缩写的含义如下：

- uid：用户 ID
- e：电子邮件地址
- cn：用户的通用名称
- o：组织
- c：国家/地区

DN 可能包括多种名称/值对。它们用于在支持 LDAP 的目录中标识证书主题和条目。

使用 LDIF

如果您当前没有目录，或者要在现有目录中添加一个新的子树，则可以使用 Directory Server 的 Administration Server LDIF 导入功能。此功能将接受一个包含 LDIF 的文件并尝试由 LDIF 条目生成一个目录或新子树。您还可以使用 Directory Server 的 LDIF 导出功能将当前目录导出到 LDIF。此功能将创建一个 LDIF 格式的文件，用来表示您的目录。可以使用 `ldapmodify` 命令和相应的 LDIF 更新语句来添加或编辑条目。

要使用 LDIF 将条目添加到数据库，请先在某个 LDIF 文件中定义条目，然后从 Directory Server 导入该 LDIF 文件。

使用验证数据库

验证数据库，也称为 `auth-db`，是指已知用户的数据库，同时也代表一种对照该数据库验证客户机请求的机制。服务器可以同时配置多个 `auth-db` 条目，并且这些条目的类型可以相同。`auth-db` 用户数据库由 ACL 处理模块使用。

服务器支持以下验证数据库：

1. **LDAP**—用户数据存储在 LDAP 目录服务器（如 Sun Java System Directory Server）中。
2. **文件**—用户数据存储在磁盘文件中。对于不提供（或不需要）集中用户管理的开发或小型部署，此 `auth-db` 尤其方便。文件 `auth-db` 支持若干不同的文件格式：
 - a. **keyfile**—`keyfile` 格式用于存储用户（以及每个用户的可选组成员）列表。密码存储为单向（不可恢复）散列。这是默认格式。
 - b. **digestfile**—`digestfile` 与 `keyfile` 非常类似，也支持 HTTP 摘要验证方法。
 - c. **htaccess**—这是一种传统格式，不应将其用于新的安装或添加新用户。
3. **PAM**—PAM 是受 Sun Java System Web Server 7.0 支持的新 `auth-db`。PAM `auth-db` 将验证委派给 Solaris PAM 堆栈，从而使 Web 服务器系统上的现有 Solaris 用户也可对 Web 服务器进行验证。

注 - 仅在 Solaris 9 和 10（或更高版本）中支持 PAM auth-db，并且必须以超级用户身份运行 Web 服务器实例。

创建验证数据库

要通过管理控制台创建验证数据库，请单击**配置 > 配置名称 > 访问控制 > 验证数据库 > 新建**按钮。检查管理控制台内联帮助，了解有关字段说明。根据选定的验证数据库，这些字段将有所变化。例如，对于基于 PAM 的验证数据库，仅需要提供验证数据库的名称。

创建验证数据库必需的选项枚举如下：

LDAP	<ul style="list-style-type: none">■ 验证数据库的名称■ 主机名■ 端口■ 基 DN
密钥文件	<ul style="list-style-type: none">■ 验证数据库的名称■ 文件路径
摘要文件	<ul style="list-style-type: none">■ 验证数据库的名称■ 文件路径
PAM	<ul style="list-style-type: none">■ 验证数据库的名称

要通过 CLI 创建验证数据库，请执行以下命令：

```
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
```

请参见 CLI 参考 create-authdb(1)。

在上例中，为验证数据库指定了一个 URL。在此 URL 方案中，指定了验证数据库的类型。例如，ldap://ds.example.com/dc=example,dc=com 将某个 LDAP 目录服务器配置为验证数据库。

设置用户和组

使用 Administration Server，可以编辑 LDAP 和文件 auth-db 类型的用户帐户、组列表、访问权限、组织单位以及用户和组的其他特定信息。

▼ 添加用户

1 选择配置。

从配置列表中选择相应的配置。单击“配置”选项卡以获取列表。

2 单击“访问控制”>“用户”选项卡。

3 单击“新建”按钮。

4 添加用户信息。

输入用户 ID 和密码。（可选）输入用户所属的组。用户 ID 必须是唯一的。如果是基于 LDAP 的验证数据库，则 Administration Server 通过从搜索基（**基 DN**）开始向下搜索整个目录来查看该用户 ID 是否正在使用，以确保用户 ID 的唯一性。但是请注意，如果您使用 Directory Server `ldapmodify` 命令行实用程序（如果可用）创建用户，则不能确保用户 ID 的唯一性。

注-使用 CLI

要通过 CLI 创建用户，请执行以下命令：

```
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
```

请参见 CLI 参考 `create-user(1)`。

▼ 添加组

1 选择配置。

从配置列表中选择相应的配置。单击“配置”选项卡以获取列表。

2 单击“访问控制”>“组”选项卡。

3 单击“新建”按钮。

4 输入组名。

- 5 从“将用户添加到组”部分中搜索现有用户并将其添加到组中。

注 - 在验证数据库（如 keyfile 或 digestfile）中创建组至少需要指定一个用户。

注 - 使用 CLI

要通过 CLI 创建组，请执行以下命令：

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

请参见 CLI 参考 create-group(1)。

▼ 删除用户

- 1 选择配置。
从配置列表中选择相应的配置。单击“配置”选项卡以获取列表。
- 2 单击“访问控制”>“用户”选项卡。
- 3 选择您需要从中删除用户的验证数据库。
- 4 在“搜索用户”文本框中输入用户 ID，然后单击“搜索”按钮。
- 5 从“用户 ID”列中选择相应的用户，然后单击“删除”按钮。



注意 - 如果删除用户后组中不包含任何成员，从 keyfile/digestfile 验证数据库中删除用户也将删除关联的组。出现这种情况是因为 keyfile/digestfile 验证数据库不允许组中不包含任何成员。

注 - 使用 CLI

要通过 CLI 删除用户，请执行以下命令：

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

请参见 CLI 参考 delete-user(1)。

▼ 删除组

- 1 选择配置。
从配置列表中选择相应的配置。单击“配置”选项卡以获取列表。
- 2 单击“访问控制”>“组”选项卡。
- 3 选择您需要从中删除组的验证数据库。
- 4 在“搜索用户”文本框中输入用户 ID，然后单击“搜索”按钮。
- 5 从“用户 ID”列中选择相应的用户，然后单击“删除”按钮。

注 - 删除组并不会删除该组中包含的用户。您必须手动删除用户或重新指定组。

注 - 使用 CLI

要通过 CLI 删除组，请执行以下命令：

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

请参见 CLI 参考 delete-group(1)。

静态组和动态组

组是 LDAP 数据库中用于描述一组对象的对象。Web Server 7.0 组由共享某个通用属性的用户组成。例如，对象集可以是您公司市场部的一些员工。这些员工可能属于一个名为 Marketing 的组。

对于 LDAP 服务，定义组的成员资格的方法有两种：静态和动态。静态组可显式枚举其成员对象。静态组是一个 CN，其中包含 uniqueMembers 和/或 memberURLs 和/或 memberCertDescriptions。对于静态组，其成员并不共享某个通用属性，但 CN=<Groupname> 属性除外。

动态组允许您使用一个 LDAP URL 来定义一组仅适用于组成员的规则。对于动态组，其成员共享某个通用属性或一组在 memberURL 过滤器中定义的属性。例如，如果您需要一个包含 Sales 中所有员工的组，并且这些员工已经位于 LDAP 数据库中的

"ou=Sales,o=Airius.com" 之下，则可以使用以下 memberurl 定义一个动态组：

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

随后，该组将包含 "ou=Sales,o=sun" 点下的树中具有 uid 属性的所有对象，即所有 Sales 成员。

对于静态组和动态组，如果您使用 `memberCertDescription`，则其成员可以通过证书共享某个通用属性。请注意，这仅在 ACL 使用 SSL 方法时才适用。

创建新组后，您可以向其中添加用户或成员。

静态组

对于 LDAP 服务，您可以通过 Administration Server 在任意数量的用户的 DN 中指定相同的组属性来创建静态组。静态组不会改变，除非您向其中添加用户或从中删除用户。

创建静态组的指导原则

使用 Administration Server 表单创建新静态组时，请考虑以下指导原则：

- 静态组可以包含其他静态或动态组。
- 可以选择为新组添加说明。
- 如果您的目录定义了任何组织单位，则可以使用“将新组添加到”列表指定要放置新组的位置。默认位置为目录的根节点（最顶端的条目）。

动态组

动态组具有一个 `groupOfURLs` 对象类 `objectclass`，可以没有也可以具有多个 `memberURL` 属性，其中每个属性都是一个描述一组对象的 LDAP URL。

对于 LDAP 服务，如果您希望基于任何属性自动将用户分组，或者希望将 ACL 应用于包含匹配 DN 的特定组，则 Web Server 允许您创建动态组。例如，您可以创建一个组，该组将自动包括任何包含属性 `department=marketing` 的 DN。如果您对 `department=marketing` 应用搜索过滤器，搜索将返回一个组，其中包含具有属性 `department=marketing` 的所有 DN。然后，您可以从基于此过滤器的搜索结果中定义一个动态组。随后，您可以为所获得的动态组定义一个 ACL。

Web Server 如何实现动态组

Web Server 在 LDAP 服务器模式中以 `objectclass = groupOfURLs` 的方式实现动态组。`groupOfURLs` 类可以有多个 `memberURL` 属性，每个属性都包含一个 LDAP URL，用于枚举目录中的一组对象。组的成员是这些对象集的总和。例如，下面的组只包含一个成员 URL：

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

该实例描述了一个由 "o=mcom.com" 下属于 "marketing" 部门的所有对象组成的集合。LDAP URL 可以包含搜索基 DN、范围和过滤器；但是，不包含主机名和端口。这意味着您只能引用同一个 LDAP 服务器上的对象。LDAP URL 支持所有范围。

DN 会自动包含在内，因而不需要您向组中添加每个 DN。组是动态变化的，因为每次 ACL 验证需要查找组时，Sun ONE Web Server 都将执行一个 LDAP 服务器搜索。ACL 文件中使用的用户和组名与 LDAP 数据库中的对象的 cn 属性相对应。

注 – Web Server 使用 cn (commonName) 属性作为 ACL 的组名。

在 dbswitch.conf 配置文件（将 ACL 数据库名称与实际的 LDAP 数据库 URL 相关联）和 ACL 文件（定义要用于每个 ACL 的数据库）中，同时定义了从 ACL 到 LDAP 数据库的映射。例如，如果需要名为 "staff" 的组中成员的基本访问权限，ACL 代码将查找对象类为 groupOf<anything> 且 CN 被设置为 "staff" 的对象。该对象可通过两种方法来定义组成员，即显式枚举出成员 DN（与静态组的 groupOfUniqueNames 的操作相同），或指定 LDAP URL（例如，groupOfURLs）。

组可以同时为动态和静态

组对象可以同时包含 objectclass = groupOfUniqueMembers 和 objectclass = groupOfURL；因此，"uniqueMember" 和 "memberURL" 属性都是有效的。组成员是其静态和动态成员的总和。

动态组对服务器性能的影响

使用动态组对服务器的性能有所影响。如果您正在测试组成员，而该 DN 不是静态组的成员，则 Web Server 将检查数据库的基 DN 中的所有动态组。要完成此任务，Web Server 需要检查每个 memberURL 是否匹配，方法是对照用户 DN 检查其基 DN 和范围，然后使用用户 DN 作为基 DN 并使用 memberURL 的过滤器来执行基搜索。这一过程将产生大量的单个搜索操作。

创建动态组的指导原则

使用 Administration Server 创建新动态组时，请考虑以下指导原则：

- 动态组不能包含其他组。
- 使用以下格式输入组的 LDAP URL（没有主机和端口信息，因为这些参数将被忽略）：

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

下表说明了必需的参数：

表 8-1 动态组：必需的参数

参数名	描述
<base_dn>	搜索基的标识名 (Distinguished Name, DN)，或 LDAP 目录中开始执行搜索的起点。此参数通常被设置为目录的后缀或根，例如 "o=mcom.com"。
<attributes>	搜索将返回的属性列表。要指定多个属性，请使用逗号来分隔属性（例如，"cn,mail,telephoneNumber"）；如果不指定属性，将返回所有属性。请注意，检查动态组成员资格时将忽略此参数。
<scope>	搜索范围，其值可以是： <ul style="list-style-type: none">■ base 仅检索 URL 中所指定标识名 (<base_dn>) 的相关信息。■ one 检索 URL 中所指定标识名 (<base_dn>) 的下一级条目的相关信息。此范围不包括基条目。■ sub 检索 URL 中所指定标识名 (<base_dn>) 下面所有级别的条目的相关信息。此范围包括基条目。 此参数是必需的。
<(filter)>	应用于指定搜索范围内的条目的搜索过滤器。如果您使用的是 Administration Server 表单，则必须指定此属性。请注意，必须带有括号。 此参数是必需的。

请注意，<attributes>、<scope> 和 <(filter)> 参数是根据它们在 URL 中的位置来标识的。因此，即使不想指定任何属性，也需要使用问号来表明不限制该字段。

- 可以选择为新组添加说明。
- 如果您的目录定义了任何组织单位，则可以使用“将新组添加到”列表指定要放置新组的位置。默认位置为目录的根节点（最顶端的条目）。

管理服务器内容

本章介绍如何配置和管理虚拟服务器中的内容。

- 第 111 页中的 “配置文档目录”
- 第 112 页中的 “更改默认 MIME 类型”
- 第 113 页中的 “自定义用户公共信息目录 (UNIX/Linux)”
- 第 115 页中的 “设置 URL 重定向”
- 第 116 页中的 “使用正则表达式进行 URL 重定向”
- 第 117 页中的 “CGI 概述”
- 第 119 页中的 “配置服务器的 CGI 子系统”
- 第 121 页中的 “下载可执行文件”
- 第 121 页中的 “安装 Windows Shell CGI 程序”
- 第 122 页中的 “自定义错误响应”
- 第 123 页中的 “更改字符集”
- 第 124 页中的 “设置文档页脚”
- 第 125 页中的 “限制符号链接 (UNIX/Linux)”
- 第 126 页中的 “设置服务器解析的 HTML”
- 第 127 页中的 “设置高速缓存控制指令”
- 第 128 页中的 “配置服务器的内容压缩”
- 第 130 页中的 “配置反向代理”
- 第 132 页中的 “设置 P3P”

配置文档目录

主文档目录（也称为文档根目录）是您存储要提供给远程客户机的所有文件的中心目录。

除主文档目录外，您还可以创建文档目录。通过执行此操作，您可以让其他用户管理文档组，而无需授予他们访问您主文档根目录的权限。

▼ 创建文档目录

- 1 选择配置。
从配置列表中选择相应的配置。单击“配置”选项卡以获取可用的配置。
- 2 选择虚拟服务器。
选择需要为其添加新文档目录的虚拟服务器。单击“虚拟服务器”选项卡为选定的配置获取已配置虚拟服务器的列表。
- 3 单击“内容处理”>“文档目录”选项卡。
- 4 单击“新建”按钮。配置以下参数：
 - URL 前缀 — 需映射至目录的 URI 前缀。
 - 目录路径 — 服务器的绝对路径和用于存储文档的有效目录。

注 - 使用 CLI

要通过 CLI 创建文档目录，请执行以下命令：

```
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
```

请参见 CLI 参考 create-document-dir(1)。

更改默认 MIME 类型

向客户机发送文档时，服务器中的某一部分将标识该文档的类型，从而使客户机能够正确地显示文档。但是，服务器有时无法确定文档的正确类型，因为服务器中没有定义该文档的扩展名。在这种情况下，将发送默认值。

默认值通常为 text/plain，但是您应当将其设置为服务器中存储的最常见的文件类型。下面列出了一些常用的 MIME 类型：

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

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

▼ 更改默认 MIME 类型

- 1 选择配置。
从配置列表中选择相应的配置。单击“配置”选项卡以获取可用的配置。
- 2 选择虚拟服务器。
单击“虚拟服务器”选项卡为选定的配置获取已配置虚拟服务器的列表。
- 3 单击“内容处理”>“常规”选项卡。
- 4 更改“其他”部分下的“默认 MIME 类型”值。

注 - 使用 CLI

要通过 CLI 创建 MIME 类型，请执行以下命令：

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

请参见 CLI 参考 create-mime-type(1)。

您无需为每台虚拟服务器创建单独的 MIME 类型文件。相反，可以根据需要创建任意数量的 MIME 类型文件，然后将它们与一台虚拟服务器相关联。默认情况下，服务器中存在一个 MIME 类型文件 (mime.types)，并且无法删除。

自定义用户公共信息目录 (UNIX/Linux)

有时用户希望维护自己的 Web 页面。您可以配置公共信息目录，使服务器中的所有用户都可以创建主页和其他文档，而无需您的介入。

通过此系统，客户机可以使用特定的 URL 来访问您的服务器，服务器会将该 URL 识别为公共信息目录。例如，假设您选择了前缀 ~ 和目录 public_html。如果收到一个对 http://www.sun.com/~jdoe/aboutjane.html 的请求，服务器将认为 ~jdoe 指向一个用户的公共信息目录。服务器将在系统的用户数据库中查找 jdoe 并找到 Jane 的主目录。然后，服务器会查看 ~/jdoe/public_html/aboutjane.html。

要将服务器配置为使用公共目录，请执行以下步骤：

▼ 配置文档目录

- 1 在“虚拟服务器”页面中，单击“内容处理”选项卡。

- 2 单击“文档目录”。

- 3 在“用户文档目录”下，选择用户 URL 前缀。

该前缀通常为 ~，因为波浪号字符是用于访问用户主目录的标准 UNIX/Linux 前缀。

- 4 在用户主目录中，选择服务器要在其中查找 HTML 文件的子目录。

通常，该目录为 public_html。

- 5 指定密码文件。

服务器需要知道可以在何处找到列出了系统用户的文件。服务器使用该文件来确定有效的用户名并找到其主目录。如果您将系统密码文件用于此用途，服务器将使用标准库调用来查找用户。您也可以创建另一个用户文件来查找用户。您可以指定该用户文件的绝对路径。

该文件中的每一行都应具有以下结构（以 * 表示 /etc/passwd 文件中不需要的元素）：

```
username:*:groupid*:homedir:*
```

- 6 选择是否在启动时装入密码数据库。

- 7 单击“保存”。

有关更多信息，请参见联机帮助中的“用户文档目录”页面。

为用户提供独立的目录的另一种方法是：创建一个映射到所有用户都可以修改的中心目录的 URL。

限制内容发布

在某些情况下，系统管理员可能希望限制可以通过用户文档目录发布内容的用户帐户。要限制某个用户的发布操作，请在 /etc/passwd 文件中该用户主目录路径的末尾添加一个斜杠：

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

成为：

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

进行此修改后，Sun Java System Web Server 将不支持来自该用户的目录的页面。请求该 URI 的浏览器会收到“404 未找到文件”错误，并且 Web 服务器访问日志中将记录一个 404 错误。不会向错误日志中记录任何错误。

如果后来您又决定允许该用户发布内容，则可以从 `/etc/passwd` 条目中删除该结尾斜杠，然后重新启动 Web 服务器。

启动时装入整个密码文件

您也可以在启动时装入整个密码文件。如果选择此选项，服务器在启动时会将密码文件装入内存，以便用户可以更快地进行查找。但是，如果密码文件非常大，则此选项会占用过多内存。

设置 URL 重定向

URL 重定向使您可以将一个 HTTP URL 的文档请求重定向至另一个 HTTP URL。转发 URL 或重定向是服务器用来通知用户 URL 已经发生更改（例如，URL 由于文件已移动至其他目录或其他服务器而发生更改）的一种方法。您也可以使用重定向，将对某服务器上文档的请求无缝地发送至另一台服务器上的文档。

例如，如果将 `http://www.sun.com/info/movies` 转发至前缀 `film.sun.com`，则 URL `http://www.sun.com/info/movies` 将重定向至 `http://film.sun.com/info/movies`。

有时，您可能会希望将对一个子目录中所有文档的请求重定向至某个特定的 URL。例如，如果必须删除某个目录（因为该目录产生的通信量过大，或者由于某种原因不再处理该目录中的文档），则可以将对其中任何文档的请求定向到一个解释了该文档为何不再可用的页面。例如，可以将 `/info/movies` 的前缀重定向至 `http://www.sun.com/explain.html`。

您可以在虚拟服务器级别上设置 URL 重定向。

要配置 URL 重定向，请执行以下操作：

1. 单击**配置**选项卡，并从配置列表中选择相应的配置。
2. 单击**虚拟服务器**子选项卡，并从虚拟服务器列表中选择虚拟服务器。
3. 依次单击**内容处理**子选项卡和**URL 重定向**子选项卡。
4. 单击**新建**按钮，添加新的 URL 重定向规则。
5. 为介绍的字段提供所需的值。单击**确定**按钮。如果需要，您可能还需要为配置单击**部署**按钮。

下表介绍了添加新的 URL 重定向规则时的必填参数。

表 9-1 URL 重定向参数

参数	说明
URL 地址	应重定向请求的 URL 地址。将对该 URL 的所有 HTTP 请求重定向至目标 URL 中指定的 URL。
目标 URL	应将请求重定向至的 URL。“URL 地址”中指定的 URL 的所有 HTTP 请求均将重定向至此 URL。
URL 类型	固定。 已启用/已禁用 。固定的 URL 为静态 URL，如指向 HTML 页面的链接。非固定的 URL 为具有请求参数的动态 URL 或仅具有前缀的 URL。

注 - 使用 CLI

要通过 CLI 添加新的 URL 重定向规则，请执行以下命令：

```
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
```

请参见 CLI 参考 create-url-redirect(1)。

使用正则表达式进行 URL 重定向

Sun Java System Web Server 7.0 经过增强，支持在配置文件中使用正则表达式（也称为模式）和请求时间参数插值。此外，与支持匹配的通配符模式还扩展到 server.xml。URL 重定向是以 SAF 方式实现的。重定向 SAF 允许您重定向与某一前缀匹配的 URI。您可以使用 from 参数指定前缀，使用 url 或 url-prefix 参数指定要重定向到的 URL。在 Sun Java System Web Server 7.0 中，from 参数是可选的。如果省略 from，将重定向所有 URI。

在 obj.conf 文件中，新的 <If>、<Elseif> 和 <Else> 标记支持 SAF 参数。请参见附录 - obj.conf - 语法和用法。这些标记包含指令。使用这些标记，可以定义执行指令的条件。此外，也可以使用这些标记动态生成 SAF 参数。

Sun Java System Web Server 7.0 提供了 URL 重写功能，该功能是 Apache HTTP 服务器的 mod_rewrite 模块的超集。与 Apache 的 mod_rewrite 函数不同，<If> 标记提供以下功能：

- 可以处理 URI、路径、标头字段和响应主体。
- 可在请求处理的任何阶段使用。
- 可与任何 SAF（包括第三方插件）一起使用。

下面的指令为例：

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

以上指令可以使用正则表达式重写，如下所示：

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

在以上代码片段中，请注意使用了正则表达式，而非 `from` 参数。如果需要将对 `/site1/*` 的所有请求重定向至 `http://site1.mycompany.com/*/index.html`，请注意以下方法：

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

此处，`<If>` 标记将匹配 `(.*)` 的任何值指定给变量 `$1`。`url` 参数中的 `$1` 将被动态替换为原始请求中的值。这意味着，以上 `obj.conf` 代码片段将会导致对 `/site1/download` 的请求重定向至 `http://site1.mycompany.com.com/download/index.html`。

组合 `<If>` 和 `redirect` 可提供 `mod_rewrite` 的某些灵活性。但是，与 `mod_rewrite` 不同，`<If>` 可用于重定向和重写 URL 以外的操作。`<If>` 还可以与任何 SAF（包括第三方插件）一起使用。

以上方法可配置一个“**302 已暂时移动**”的重定向。在 Sun Java System Web Server 7.0 中，还可以添加 `status="301"` 参数，以指示您需要一个“**301 已永久移动**”的重定向。

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

CGI 概述

公共网关接口 (Common Gateway Interface, CGI) 程序可以使用许多编程语言进行定义。在 UNIX/Linux 计算机上，您可能会发现 CGI 程序被编写成 Bourne shell 或 Perl 脚本。

注 - 在 UNIX/Linux 中，具有附加的 `CGIStub` 运行进程，服务器可使用这些进程来帮助 CGI 执行。只有在首次访问 CGI 期间才创建这些进程。进程数量的变化取决于服务器上的 CGI 负荷。请不要删除这些 `CGIStub` 进程。服务器停止时，它们将消失。

有关更多信息，请参见联机的《Sun Java System Web Server Performance Tuning and Sizing Guide》中有关 `MinCGIStub`、`MaxCGIStub` 和 `CGIStubIdleTimeout` 的讨论。

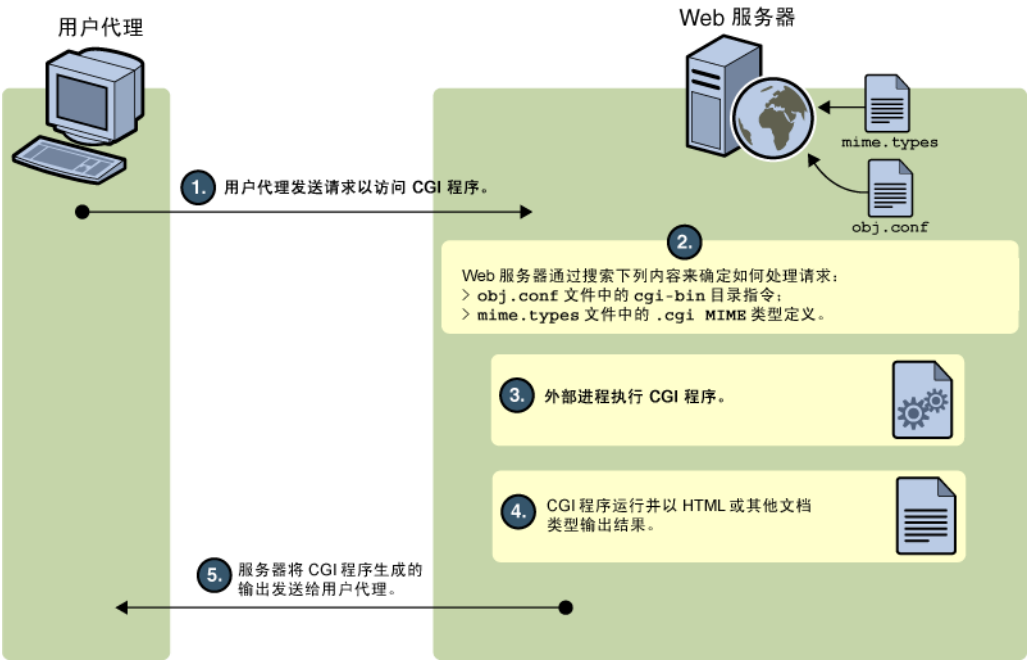
在 Windows 计算机上，您可能会发现 CGI 程序是用 C++ 或批处理文件编写的。对于 Windows 而言，利用基于 Windows 的编程语言（如 Visual Basic）编写的 CGI 程序将使用不同的机制来操作服务器。这些程序被称为 Windows CGI 程序。

注 - 要运行命令行实用程序，需要手动设置 Path 变量以包括 `server_root/bin/https/bin`。

不管使用什么编程语言，所有 CGI 程序均以同样的方式接受和返回数据。有关编写 CGI 程序的更多信息，请参见以下信息源：

- Sun Java System Web Server *Developer's Guide*
- 公共网关接口，可从以下站点访问：
<http://hoohoo.ncsa.uiuc.edu/cgi/overview.html>
- 联机文档资料中关于 CGI 的文章，可从以下 Web 站点获得：
<http://docs.sun.com>

下图介绍了如何在 Web Server 7.0 中处理 CGI 请求：



在服务器计算机上存储 CGI 程序的方法有两种：

- 指定只包含 CGI 程序的目录。不管文件扩展名为何，所有文件均作为程序运行。

- 指定所有 CGI 程序均属于某一文件类型。也就是说，这些程序全都使用文件扩展名 .cgi、.exe 或 .bat。这些程序可以位于文档根目录中的任何目录中，也可以位于文档根目录下的任何目录中。

如果需要，您可以同时启用两个选项。

两种实现方法各有优点。如果只允许一组特定的用户添加 CGI 程序，请将 CGI 程序保留在指定的目录中并限制对这些目录的访问。如果允许任何可以添加 HTML 文件的用户都能添加 CGI 程序，请使用替代文件类型。用户可以将 CGI 文件和 HTML 文件保留在相同的目录中。

如果选择目录选项，服务器会尝试将该目录中的所有文件都视为 CGI 程序。通过使用相同的标记，如果选择文件类型选项，则服务器会尝试将具有 .cgi、.exe 或 .bat 文件扩展名的所有文件都作为 CGI 程序来处理。如果某个文件具有其中一个扩展名但不是 CGI 程序，用户试图访问时就会出现错误。

注-默认情况下，CGI 程序的文件扩展名为 .cgi、.exe 和 .bat。但是，您可以通过修改 MIME 类型文件来更改哪些扩展名表示 CGI 程序。要执行此操作，请选择“服务器首选项”选项卡并单击“MIME 类型”链接。

配置服务器的 CGI 子系统

Sun Java System Web Server 使您可以使用管理控制台 GUI 添加 CGI 文档目录。

要添加新的 CGI 文档目录，请执行以下操作：

1. 单击**配置**选项卡，并从配置列表中选择相应的配置。
2. 单击**虚拟服务器**子选项卡，并从虚拟服务器列表中选择虚拟服务器。
3. 依次单击**内容处理**子选项卡和**CGI**子选项卡。
4. 单击**新建**按钮以添加新的 CGI 文档目录。
5. 为介绍的字段提供所需的值。单击**确定**按钮。如果需要，您可能还需要为配置单击**部署**按钮。

下表介绍了添加新的 CGI 文档目录时的必填字段。

表 9-2 CGI 参数

参数	说明
前缀	键入要用于该目录的 URL 前缀。也就是说，您键入的文本将在 URL 中作为 CGI 程序的目录显示出来。 例如，如果您键入 <code>cgi-bin</code> 作为 URL 前缀，则这些 CGI 程序的所有 URL 均将具有以下结构： <code>http://yourserver.domain.com /cgi-bin/program-name</code>
CGI 目录	在“CGI 目录”文本字段中，键入目录的位置作为绝对路径。请注意，该目录不必位于文档根目录下。因此，您需要指定 URL 前缀。 注 - 指定的 URL 前缀可与实际的 CGI 目录有所不同。
用户	指定要以其执行 CGI 程序的用户名称。
组	指定要以其执行 CGI 程序的组名称。
Chroot	指定执行开始之前要 <code>chroot</code> 的目录。
Nice	指定 <code>nice</code> 值，即确定 CGI 程序相对于服务器的优先级的增量。 通常，服务器在 <code>nice</code> 值为 0 的情况下运行， <code>nice</code> 增量介于 0（CGI 程序与服务器在同一优先级下运行）至 19（CGI 程序的运行优先级远远低于服务器）之间。虽然可以通过将 <code>nice</code> 增量指定为 -1 来增加 CGI 程序的优先级使之高于服务器的优先级，但不建议使用这种方法。

要删除现有 CGI 目录，请选择 CGI 目录并单击“删除”按钮。要更改现有目录的 URL 前缀或 CGI 目录，请单击目录链接。

将 CGI 程序复制到指定的目录中。请记住，这些目录中的所有文件均将作为 CGI 文件来处理，因此不要将 HTML 文件放入 CGI 目录中。

要将 CGI 指定为文件类型，请执行以下操作：

1. 单击**配置**选项卡，并从配置列表中选择相应的配置。
2. 单击**虚拟服务器**子选项卡，并从虚拟服务器列表中选择虚拟服务器。
3. 依次单击**内容处理**子选项卡和**CGI**子选项卡。
4. 单击**CGI 文件类型**单选框以启用。

CGI 文件必须具有文件扩展名 `.bat`、`.exe` 或 `.cgi`。如果服务器将具有这些扩展名的非 CGI 文件作为 CGI 文件进行处理，将导致发生错误。

注 - 使用 CLI

您可以创建 CGI 目录，其中包含将由服务器进行处理的 CGI 程序。CGI 程序为特定的文件类型，例如 .cgi、.exe 或 .bat。这些程序可以位于文档根目录中的任何目录中，也可以位于文档根目录下的任何目录中。

要通过 CLI 添加 CGI 目录，请执行以下命令：

```
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
```

请参见 CLI 参考 create-cgi-dir(1)。

下载可执行文件

如果使用 .exe 作为 CGI 文件类型，则不能将 .exe 文件作为可执行文件下载。

解决此问题的方案之一是压缩希望用户下载的可执行文件，从而使扩展名不是 .exe。该解决方案还有助于缩短下载时间。

另一个可能的解决方案是从 magnus-internal/cgi 类型中删除作为文件扩展名的 .exe，而将其添加到 application/octet-stream 类型（用于标准可下载文件的 MIME 类型）中。您可以通过 Server Manager 执行此操作，方法是选择“服务器首选项”选项卡并单击“MIME 类型”链接。但是，该方法的缺点是进行此更改后无法将 .exe 文件用作 CGI 程序。

另一个解决方案是编辑服务器的 obj.conf 文件，设置一个下载目录，自动下载该目录中的任何文件。服务器的其他部分不会受到影响。有关更多信息，请参见：

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

安装 Windows Shell CGI 程序

Windows Shell CGI 程序概述

Shell CGI 是一种服务器配置，通过该配置，您可以使用在 Windows 中设置的文件类型关联来运行 CGI 应用程序。

例如，如果服务器收到对一个名为 hello.pl 的 shell CGI 文件的请求，服务器将通过与 .pl 扩展名关联的程序，使用 Windows 文件类型关联来运行该文件。如果 .pl 扩展名与程序 C:\bin\perl.exe 相关联，服务器将尝试按以下方式执行 hello.pl 文件：

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

配置 shell CGI 的最简便方式是在服务器的文档根目录中创建一个仅包含 shell CGI 文件的目录。但是，您可以通过从 Sun ONE Web Server 编辑 MIME 类型，配置服务器以使特定的文件扩展名与 shell CGI 相关联。

注 – 有关设置 Windows 文件扩展名的信息，请参见相关的 Windows 文档。

自定义错误响应

您可以指定错误响应，当客户机遇到来自虚拟服务器的错误时，会将详细消息发送至客户机。可以指定要发送的文件或要运行的 CGI 程序。

例如，您可以更改服务器收到特定目录的错误时的响应方式。如果客户机尝试接受访问控制保护的服务器内容，您可以返回一个错误文件，其中包括如何获得帐户的信息。

在启用自定义错误响应之前，必须创建为响应错误而发送的 HTML 文件或运行的 CGI 程序。

要添加自定义错误页面，请执行以下步骤：

1. 单击**配置**选项卡，并从配置列表中选择相应的配置。
2. 单击**虚拟服务器**子选项卡，并从虚拟服务器列表中选择虚拟服务器。
3. 依次单击**内容处理**子选项卡和**错误页面**子选项卡。
4. 单击**新建**按钮以添加自定义错误页面。

对于每个要更改的错误代码，请指定包含该错误响应的文件或 CGI 的绝对路径。

5. 单击**确定**以返回错误页面列表。

注 – 使用 CLI

要通过 CLI 自定义错误页面，请执行以下命令：

```
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
```

请参见 CLI 参考 `set-error-page(1)`。

更改字符集

文档的字符集部分取决于编写文档所用的语言。通过选择资源并输入该资源的字符集，可以覆盖客户机的文档、一组文档或目录的默认字符集设置。

大多数浏览器都可在 HTTP 中使用 MIME 类型的 `charset` 参数来更改其字符集。如果服务器的响应中包含此参数，浏览器将相应地更改其字符集。请参见以下示例：

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

RFC 17.000 中指定了以下某些常用浏览器识别的 `charset` 名称（以 `x-` 开头的名称除外）：

■ <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，识别 `us-ascii` 的以下别名：

■ <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>	

识别 `so_8859-1` 的以下别名：

■ <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>

要更改字符集，请执行以下步骤：

▼ 更改字符集

- 1 在“虚拟服务器”页面中，单击“内容处理”选项卡。
- 2 单击“常规”选项卡。
- 3 在“其他”部分下设置默认字符集。
如果将此字段保留为空，字符集将设置为 NONE。
- 4 单击“保存”。

设置文档页脚

您可以在服务器的特定部分中为所有文档指定一个文档页脚，其中可包括上次修改时间。除 CGI 脚本的输出信息或已解析的 HTML (.shtml) 文件外的所有文件都可以使用此页脚。如果需要在 CGI 脚本输出或已解析的 HTML 文件中显示文档页脚，请将页脚文本输入到单独的文件中，并添加一行代码或其他服务器端引用以将此文件附加到页面输出。

要设置文档页脚，请执行以下步骤：

▼ 设置文档页脚

- 1 在虚拟服务器页面中，单击“内容处理”选项卡。
- 2 单击“常规”子选项卡，然后转至“文档页脚”部分。
- 3 指定希望包含页脚的文件类型。
- 4 指定日期格式。
- 5 键入希望显示在页脚中的文本。
文档页脚的字符数最多为 7.065 个。如果要包含文档的上次修改日期，请键入字符串：LASTMOD:。
- 6 单击“保存”。

注-使用 CLI

要通过 CLI 设置文档页脚，请执行以下命令：

```
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"
```

请参见 CLI 参考 `enable-document-footer(1)`。

限制符号链接 (UNIX/Linux)

您可以在服务器中限制文件系统链接的使用。文件系统链接是对存储在其他目录和文件系统中的文件的引用。使用引用，用户可以像访问当前目录中的文件一样访问远程文件。文件系统链接有两种类型：

- 硬链接—硬链接实际上是指向相同数据块集的两个文件名；原始文件和链接是相同的。因此，硬链接不能位于不同的文件系统上。
- 符号（软）链接—符号链接包含两个文件：原始文件包含数据，另一个文件指向该原始文件。符号链接比硬链接更灵活。符号链接可以在不同文件系统间使用，并且可以链接到目录。

有关硬链接和符号链接的更多信息，请参见 UNIX/Linux 系统文档。

使用文件系统链接可以方便地创建指向位于主文档目录之外的文档的指针，并且任何人都可以创建这些链接。因此，您可能会担心有人会创建指向敏感文件（例如，机密文档或系统密码文件）的指针。

要限制符号链接，请执行以下步骤：

▼ 限制符号链接

- 1 在虚拟服务器页面中，单击“内容处理”选项卡。
- 2 单击“常规”子选项卡。
- 3 转至“其他”部分下的“符号链接”部分。
- 4 选择是否启用软链接和/或硬链接以及起始目录。
- 5 单击“保存”。

注 – 使用 CLI

要通过 CLI 限制符号链接，请执行以下命令：

```
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
```

请参见 CLI 参考 `set-symlinks-prop(1)`。

设置服务器解析的 HTML

通常情况下，HTML 发送到客户机时就像它存储在磁盘上一样，无需服务器进行任何干预。但是，服务器可以在发送文档之前搜索 HTML 文件以查找特定的命令（也就是说，服务器可以解析 HTML）。如果希望服务器解析这些文件并在文档中插入请求的特定信息或文件，必须首先启用 HTML 解析。

要解析 HTML，请执行以下步骤：

▼ 设置服务器解析的 HTML

- 1 在虚拟服务器页面中，单击“内容处理”选项卡。
 - 2 单击“常规”子选项卡。
 - 3 在“已解析的 HTML/SSI 设置”中，选择是否激活服务器解析的 HTML。
您可以针对 HTML 文件而非 `exec` 标记进行激活，也可以同时针对 HTML 文件和 `exec` 标记进行激活，后一种方式允许 HTML 文件执行服务器中的其他程序。
 - 4 选择要解析的文件。
您可以选择是只解析扩展名为 `.shtml` 的文件，还是解析所有 HTML 文件（解析所有 HTML 文件会降低性能）。如果使用的是 UNIX/Linux，还可以选择解析开启了执行权限的 UNIX/Linux 文件，但是这样做不可靠。
 - 5 单击“保存”。
- 有关使用服务器解析的 HTML 的更多信息，请参见《Sun Java System Web Server Developer's Guide》。

注-使用 CLI

要通过 CLI 设置服务器解析的 HTML，请执行以下命令：

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

请参见 CLI 参考 enable-parsed-html(1)。

设置高速缓存控制指令

Sun Java System Web Server 通过高速缓存控制指令控制代理服务器高速缓存的信息。使用高速缓存控制指令，可以覆盖代理的默认高速缓存，以防止以后对敏感信息进行高速缓存，或对其进行检索。代理服务器必须遵循 HTTP 1.1 规范，才能使这些指令正常运行。

有关 HTTP 1.1 的更多信息，请参见超文本传输协议 -- HTTP/1.1 规范 (RFC 2068)，该规范位于以下位置：

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

要设置高速缓存控制指令，请执行以下步骤：

▼ 设置高速缓存控制指令

- 1 从虚拟服务器页面中，单击“内容处理”选项卡。
- 2 单击“常规”子选项卡并转到“其他”部分中的“高速缓存控制指令”字段。
- 3 填写各个字段。响应指令的有效值如下：
 - 公用。任何高速缓存均可以缓存响应。这是默认值。
 - 专用。仅专用（非共享）高速缓存可以缓存响应。
 - 无高速缓存。任何高速缓存都不能缓存响应。
 - 无存储。高速缓存不能将请求或响应存储到非易失性存储器中的任何地方。
 - 必须重新验证。必须在原始服务器中重新验证高速缓存条目。
 - 最长生存期（秒）。客户机不接受生存期大于该值的响应。
- 4 单击“保存”。

注-使用 CLI

要通过 CLI 设置高速缓存控制指令，请执行以下命令。

```
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
```

请参见 CLI 参考 `set-cache-control-directives(1)`。

配置服务器的内容压缩

Sun Java System Web Server 7.0 支持 HTTP 内容压缩。内容压缩可以提高向客户机提供内容的速度，同时可以提供更多内容，而无需增加硬件的消耗。内容压缩减少了内容的下载时间，对使用拨号连接和高流量连接的用户尤其有用。

通过内容压缩，Web 服务器可以发送压缩的数据并指示浏览器对这些数据进行动态解压缩，从而减少数据的发送量并提高页面的显示速度。

将服务器配置为提供预压缩的内容

您可以将 Sun Java System Web Server 配置为在指定的目录中生成并存储文件的预压缩版本。进行这样的配置后，只有收到 `Accept-encoding: gzip` 标头，对配置为提供预压缩内容的目录中文件的所有请求才会被重定向为对该目录中等效压缩文件的请求（如果存在这样的文件）。例如，如果 Web Server 收到对 `myfile.html` 的请求，且 `myfile.html` 和 `myfile.html.gz` 都存在，则那些带有相应 `Accept-encoding` 标头的请求将收到压缩文件。

要将服务器配置为提供预压缩内容，请执行以下步骤：

▼ 更改预压缩的内容的设置

- 1 从虚拟服务器页面中，单击“内容管理”选项卡。
- 2 单击“常规”子选项卡。
- 3 转至“压缩”>“预压缩的内容”部分，然后从以下选项中进行选择。
 - **预压缩的内容**—启用/禁用。允许您指示服务器为选定的资源提供预压缩的内容。
 - **生存期检查**—指定是否检查压缩版本是否比非压缩版本旧。
如果选择此选项，则当压缩版本比非压缩版本旧时将不会选择压缩版本。

如果未选择此选项，将始终选择压缩版本，即使压缩版本比非压缩版本旧也是如此。

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

如果选择此选项，当选择文件的压缩版本时，将始终插入 Vary: Accept-encoding 标头。

如果未选择此选项，将不会插入 Vary: Accept-encoding 标头。

- 4 单击“保存”。

将服务器配置为根据需要压缩内容

您也可以将 Sun Java System Web Server 7.0 配置为动态压缩传输数据。动态生成的 HTML 页面仅在用户提出请求时才会存在。这对基于电子商务的 Web 应用程序和数据库驱动的站点尤其有用。

要将服务器配置为根据需要压缩内容，请执行以下步骤：

▼ 根据需要压缩内容

- 1 从虚拟服务器页面中，单击“内容处理”选项卡。
- 2 单击“常规”子选项卡。转至“压缩”部分中的“根据需要压缩内容”部分。
- 3 请从以下选项中进行选择：
 - **按需压缩**—对选定的资源启用/禁用按需压缩。
 - **插入 Vary 标头**—指定是否插入 Vary: Accept-encoding 标头。
如果选择此选项，当选择文件的压缩版本时，将始终插入 Vary: Accept-encoding 标头。
如果未选择此选项，将不会插入 Vary: Accept-encoding 标头。
 - **片断大小**—指定压缩库 (zlib) 使用的内存段大小（以字节为单位）以控制每次压缩的量。默认值是 8096。
 - **压缩级别**—指定压缩的级别。请选择 1 至 9 之间的值。值为 1 时速度最快；值为 9 时压缩效果最佳。默认值为 6，这将获得适中的速度和压缩效果。
- 4 单击“保存”。

注 – 使用 CLI

要通过 CLI 启用根据需要压缩，请执行以下命令。

```
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
```

请参见 CLI 参考 `enable-on-demand-compression(1)`。

配置反向代理

反向代理是指对客户机而言就像一个 Web 服务器（原始服务器），而实际上却将其接收到的请求转发给一个或多个原始服务器的代理。由于反向代理将其自身作为原始服务器，因此无需配置客户机即可使用反向代理。通过配置给定的反向代理以将请求转发给多个类似的已配置原始服务器，反向代理可以作为应用程序级别的软件负载平衡器运行。

注 – 在典型部署中，将在浏览器和原始服务器之间部署一个或多个反向代理。

▼ 添加代理 URI

- 1 单击“配置”选项卡并选择相应的配置。
- 2 单击“虚拟服务器”选项卡并选择虚拟服务器。
- 3 单击“内容处理”>“反向代理”选项卡。
- 4 单击“新建代理 URI”按钮。

指定以下参数的值：

- **URI** – 反向代理 URI
- **服务器 URL** – 逗号分隔的远程服务器 URL。如果提供了多个值，则服务器将在指定的服务器之间分发负载。

▼ 修改反向代理参数

- 1 单击“配置”选项卡并选择相应的配置。
- 2 单击“虚拟服务器”选项卡并选择虚拟服务器。
- 3 单击“内容处理”>“反向代理”选项卡。
- 4 单击“URI”

您可以编辑以下参数：

- **URI**—反向代理 URI。
- **服务器 URL**—逗号分隔的远程服务器 URL。如果提供了多个值，则服务器将在指定的服务器之间分发负载。
- **粘性 Cookie**—Cookie 名，在响应中出现 Cookie 后，将导致后续请求始终发送到相应的原始服务器。
- **粘性 URI 参数**—用于检查路由信息的 URI 参数的名称。如果 URI 参数出现在请求 URI 中，且其值中包含冒号 ":" 并后接路由 ID，则请求将始终发送到由该路由 ID 标识的原始服务器中。
- **路由标头**—用于将路由 ID 传送到原始服务器的 HTTP 请求标头的名称。
- **路由 Cookie**—服务器在响应中遇到“粘性 Cookie”时所生成的 Cookie 名。“路由 Cookie”存储使服务器可以将后续请求返回到同一原始服务器的路由 ID。

注 – 使用 CLI

1. 调用 `create-reverse-proxy` 命令。

```
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
```

请参见 CLI 参考 `create-reverse-proxy(1)`。

2. 修改 `obj.conf` 文件。

```
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>
```

要重定向到安全站点，请遵循相同的步骤，并为 `--server` 选项提供 `https` 地址。

设置 P3P

- 第 132 页中的“配置虚拟服务器的 P3P 设置”

隐私首选项平台 (Platform for Privacy Preference, P3P) 使 Web 站点能够以一种标准的格式来表达其隐私惯例，用户代理可以轻松地自动检索并解释这种格式。P3P 用户代理使用户可以收到站点隐私惯例的通知（以机器可读格式和人可读格式）。有关更多信息，请参见 <http://www.w3.org/P3P/>。

▼ 配置虚拟服务器的 P3P 设置

- 1 选择配置。

从配置列表中选择相应的配置。单击“配置”选项卡以获取可用配置的列表。

- 2 选择虚拟服务器。

从虚拟服务器列表中选择相应的虚拟服务器。单击“虚拟服务器”选项卡为选定的配置获取可用的虚拟服务器。

- 3 单击“常规”选项卡。配置“P3P”部分下面的以下设置。

- **已启用**—为选定的虚拟服务器启用 P3P。
- **策略 URL**—输入相关 P3P 策略文件的位置。
- **简洁策略**—简洁策略可向用户代理（浏览器或其他 P3P 应用程序）提供提示，以便用户代理可以快速同步地做出有关应用策略的决策。简洁策略是 P3P 规范的一种性能优化，它对于用户代理或服务器来说是可选的。

注-使用 CLI

要对虚拟服务器启用 P3P，请执行以下命令：

```
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
```

请参见 CLI 参考 `enable-p3p(1)`。

使用 WebDAV 实现 Web 发布

- 第 136 页中的 “关于 WebDAV”
- 第 136 页中的 “常见 WebDAV 术语”
- 第 139 页中的 “在实例级别启用 WebDAV”
- 第 140 页中的 “管理 WebDAV 集合”
- 第 141 页中的 “配置 WebDAV 属性”
- 第 143 页中的 “在服务器级别禁用 WebDAV”
- 第 143 页中的 “管理 WebDAV 验证数据库”
- 第 144 页中的 “在启用了 WebDAV 的服务器上使用源 URI 和 Translate:f 标头”
- 第 144 页中的 “锁定和解除锁定资源”
- 第 145 页中的 “最小锁定超时”

Sun Java System Web Server 7.0 支持 WebDAV（即基于 Web 的分布式制作和版本控制），这是一种基于 Web 的协作标准。WebDAV 是 HTTP/1.1 协议的扩展，它使客户机可以执行远程 Web 内容制作操作。

一个完整的 WebDAV 事务包括一台启用了 WebDAV 的服务器（例如 Sun Java System Web Server 7.0），用于处理对 WebDAV 资源的请求；还包括一台启用了 WebDAV 的客户机（例如 Adobe(R) GoLive(R) 或 Macromedia(R) DreamWeaver(R)），用于支持启用了 WebDAV 的 Web 发布请求。

在服务器端，需要启用并配置 Sun Java System Web Server 7.0 以便能够处理 WebDAV 请求。

由于以下原因，您可能希望对 WebDAV 进行配置：例如，要优化服务器性能、消除安全性风险或提供无冲突的远程制作。

为满足您的配置要求，您可以更改服务器对 WebDAV 资源保持的最小锁定时间、对集合的 PROPFIND 请求的深度以及请求主体所允许的 XML 内容的最大大小等设置。

可以在虚拟服务器级别为虚拟服务器下的所有集合配置默认 WebDAV 属性。此处配置的值对应于 `server.xml` 文件中的 DAV 元素。

也可以在集合级别配置 WebDAV 属性，这将覆盖为集合配置的任何虚拟服务器级别属性。在集合级别配置的属性值对应于 `server.xml` 文件中的 `DAVCOLLECTION` 元素。

关于 WebDAV

WebDAV 是对 HTTP/1.1 协议的扩展，添加了新的 HTTP 方法和标头，支持任意类型的 Web 资源制作，不仅支持 HTML 和 XML，还支持文本、图形、电子表格等格式。使用 WebDAV 可以完成的工作包括：

- **属性（元数据）处理。**您可以使用 WebDAV 方法 `PROPFIND` 和 `PROPPATCH` 创建、删除和查询有关 Web 页面的信息，例如作者和创建日期。
- **集合和资源管理。**您可以使用 WebDAV 方法 `GET`、`PUT`、`DELETE` 和 `MKCOL` 创建文档集并检索分层结构成员列表（类似于文件系统中的目录列表）。
- **锁定。**您可以使用 WebDAV 禁止多人同时对一个文档进行操作。可以通过 WebDAV 方法 `LOCK` 和 `UNLOCK` 使用互斥锁定或共享锁定，这将有助于防止出现“丢失更新”（更改被覆盖）的问题。
- **名称空间操作。**您可以使用 WebDAV 方法 `COPY` 和 `MOVE` 指示服务器复制和移动 Web 资源。

Sun Java System Web Server 7.0 中的 WebDAV 支持提供了以下功能：

- 与 RFC 2518 的兼容性以及与 RFC 2518 客户机的互操作性。
- 发布的安全性和访问控制
- 对基于文件系统的 WebDAV 集合和资源的有效发布操作。

常见 WebDAV 术语

本节概述了使用 WebDAV 时经常遇到的术语。

URI。URI（统一资源标识符）是一种文件标识符，通过使用缩写的 URL 提供了额外的安全保护。URL 映射代替了 URL 的第一部分，从而对用户隐藏了文件的完整物理路径名。

源 URI。术语“源 URI”是指能用来访问资源的源代码的 URI。为了更好地理解源 URI 的概念，请考虑以下示例：

一个名为 `foo.jsp` 的 JSP 页面位于 `URI/docs/date.jsp` 处。该页面包含 HTML 标记和 Java 代码，当执行该代码时，将在客户机的浏览器中列显当前的日期。当服务器收到客户机要获取 `foo.jsp` 的 GET 请求后，先执行该 Java 代码，然后提供该页面。客户机收到的并不是该 `foo.jsp` 页面（因为它驻留在服务器上），而是一个显示当前日期的动态生成的页面。

如果您要创建一个源 URI（例如 `/publish/docs`）并将其映射到包含 `foo.jsp` 的 `/docs` 目录，则对 `/publish/docs/foo.jsp` 的请求将是请求 `/docs/foo.jsp` 页面的源代码。在这种情况下，服务器将提供页面而不执行 Java 代码。客户机将收到未经处理的页面，与保存在磁盘中的页面完全一样。

这样，对源 URI 的请求也就成了请求资源的源代码。

集合。 WebDAV 集合是为 WebDAV 操作启用的一个或一组资源。集合包含一组称为成员 URI 的 URI，它们用来标识启用了 WebDAV 的成员资源。

成员 URI。 集合中一组 URI 中的一个成员。

内部成员 URI。 与集合 URI 直接相关的一个成员 URI。例如，如果 URL 为 `http://info.sun.com/resources/info` 的资源启用了 WebDAV，并且 URL 为 `http://info.sun.com/resources/` 的资源也启用了 WebDAV，则 URL 为 `http://info.sun.com/resources/` 的资源就是一个集合，它包含 `http://info.sun.com/resources/info` 并将其作为一个内部成员。

属性。 一个包含有关资源的描述性信息的名称/值对。使用属性可以有效地搜索和管理资源。例如，使用属性 "creationdate" 可以按资源的创建日期为所有资源建立索引，使用属性 "author" 可以按作者姓名创建索引。

动态属性。 由服务器强制实现的属性。例如，动态属性 `getcontentlength` 有一个值，即 GET 请求返回的实体的长度，它是由服务器自动计算的。动态属性包括以下两类：

- 属性值是只读的，由服务器维护。
- 属性值由客户机维护，但是服务器对提交的值执行语法检查。

静态属性。 不是由服务器强制实现的属性。服务器仅记录静态属性的值；客户机负责维护其一致性。

Sun Java System Web Server 支持以下动态属性：

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

注 – Sun Java System Web Server 支持动态属性 `executable`，它允许客户机更改与资源相关联的文件访问权限。

以下是对 `executable` 动态属性的 PROPPATCH 请求示例：

```
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>
```

锁定。锁定资源功能提供了这样一种机制，即可以保证在一个用户编辑资源时，其他用户不能对该资源进行修改。锁定可以防止发生覆盖冲突，解决了“丢失更新”的问题。

Sun Java System Web Server 支持两种锁定类型：共享和互斥。

新的 HTTP 标头。WebDAV 扩展了 HTTP/1.1 协议。它定义了新的 HTTP 标头，客户机可以通过这些新标头传递 WebDAV 资源请求。这些标头为：

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

新的 HTTP 方法。WebDAV 引入了若干新的 HTTP 方法，用于指示启用了 WebDAV 的服务器如何处理请求。这些方法是对现有 HTTP 方法（例如 GET、PUT 和 DELETE）的补充，可用来执行 WebDAV 事务。下面简要介绍这些新的 HTTP 方法：

- COPY。用于复制资源。复制集合使用 Depth: 标头, Destination: 标头用来指定目标。根据需要, COPY 方法还将使用 Overwrite: 标头。
- MOVE。用于移动资源。移动集合使用 Depth: 标头, Destination: 标头用来指定目标。根据需要, MOVE 方法还将使用 Overwrite: 标头。
- MKCOL。用于创建新集合。使用此方法可避免重载 PUT 方法。
- PROPPATCH。用于设置、更改或删除单个资源的属性。
- PROPFIND。用于获取一个或多个资源的一个或多个属性。当客户机向服务器提交对某个集合的 PROPFIND 请求时, 该请求可能会包含一个值为 0、1 或 infinity 的 Depth: 标头。
 - 0。指定将获取指定 URI 处的集合的属性。
 - 1。指定将获取集合以及指定的 URI 的下一级资源的属性。
 - infinity。指定将获取该集合及其包含的所有成员 URI 的属性。请注意, 由于深度为无穷大的请求需要遍历整个集合, 因而会大大增加服务器的负担。
- LOCK。为资源添加锁定。使用 Lock-Token: 标头。
- UNLOCK。删除对资源的锁定。使用 Lock-Token: 标头。

在实例级别启用 WebDAV

您可以使用 Administration Server 为整个服务器启用 WebDAV。执行此操作时, 以下指令将添加到用于加载 WebDAV 插件的 magnus.conf file 文件中:

```
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 初始化函数初始化并注册 WebDAV 子系统。

要启用 WebDAV, 请在 CLI 中执行以下命令:

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

请参见 CLI 参考 enable-webdav(1)。

管理 WebDAV 集合

启用 WebDAV 集合

要启用 WebDAV 集合，请执行以下命令：

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

请参见 CLI 参考 `enable-dav-collection(1)`。

禁用 WebDAV 集合

要禁用 WebDAV 集合，请执行以下命令：

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

请参见 CLI 参考 `disable-dav-collection(1)`。

添加 WebDAV 集合

要添加 WebDAV 集合，请执行以下命令：

```
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
```

请参见 CLI 参考 `create-dav-collection(1)`。

列出 WebDAV 集合

要列出所有 WebDAV 集合，请执行以下命令：

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

请参见 CLI 参考 `list-dav-collections(1)`。

删除 WebDAV 集合

要删除 WebDAV 集合，请执行以下命令：

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

请参见 CLI 参考 `delete-dav-collection(1)`。

配置 WebDAV 属性

设置 WebDAV 属性

要在服务器级别设置 WebDAV 属性，请执行以下命令：

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

请参见 CLI 参考 `set-webdav-prop(1)`。

查看 WebDAV 属性

要在服务器级别查看 WebDAV 属性，请执行以下命令：

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

请参见 CLI 参考 `get-webdav-prop(1)`。

设置 WebDAV 集合属性

要设置 WebDAV 集合属性，请执行以下命令：

```
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
```

请参见 CLI 参考 `set-dav-collection-prop(1)`。

查看 WebDAV 集合属性

要查看 WebDAV 集合属性，请执行以下命令：

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

请参见 CLI 参考 `get-dav-collection-prop(1)`。

修改 WebDAV 参数

下表中列出了一些通用 WebDAV 属性：

表 10-1 WebDAV 参数

参数	说明
锁定数据库路径	指定将在其中维护锁定数据库的目录。
最小锁定超时	指定锁定的最小生命周期（以秒为单位）。值为 -1 表示锁定永不过期。此值表示自动删除锁定之前锁定某元素的时间。
最大请求大小	指定 XML 请求主体的最大大小。您应配置此值以防止受到可能的拒绝服务攻击。默认值是 8192 (8K) 。
最大扩展属性深度	指定 PROPFIND 请求的深度。 0 只适用于指定的资源。 0 为默认值。 1 适用于指定的资源以及下一级别。 "infinity" 适用于指定的资源及其包含的所有资源。也可通过限制此参数的大小防止过度消耗内存。
默认拥有者	集合的默认拥有者。
URI	要在其上启用 WebDAV 的现有根 URI。
最大 PROPFIND 深度	发送到集合的 PROPFIND 请求的最大深度
锁定数据库更新时间间隔	WebDAV 锁定数据库与磁盘同步的时间间隔。使用 0 可以禁用高速缓存 WebDAV 锁定信息。
验证数据库	要使用的 ACL 验证数据库。
验证方法	要使用的验证方法。默认验证方法为“基本”。
验证提示文本	请求验证时向客户机显示的提示。
DAV ACL 数据库	
最大条目数	允许一个资源上拥有的最大 ACE 数。 0-2147.0483647.0 。指定 —1 表示无限制。
最大大小	WebDAV ACL 数据库（用于集合）的内存表示的最大大小。 0-2147.0483647.0 。指定 —1 表示无限制。
更新时间间隔	WebDAV ACL 数据库与磁盘同步的时间间隔。 0.001-3600 秒。指定 0 可以禁用高速缓存 WebDAV ACL 列表。
DAV 属性数据库	
最大大小	WebDAV 属性数据库文件的最大大小。 0-2147.0483647.0 。指定 —1 表示无限制。

表 10-1 WebDAV 参数 (续)

参数	说明
更新时间间隔	WebDAV 属性数据库与磁盘同步的时间间隔。0.01–3600 秒。指定 0 可以禁用高速缓存 WebDAV 属性。

在服务器级别禁用 WebDAV

要在服务器级别禁用 WebDAV，请执行以下命令：

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

请参见 CLI 参考 `disable-webdav(1)`。

管理 WebDAV 验证数据库

在管理控制台中，从选定的配置单击 **WebDAV** 选项卡以编辑 WebDAV 验证数据库设置。下表提供了此页面中每个字段的简短说明：

表 10-2 WebDAV 验证数据库属性

属性	说明
验证数据库	验证数据库 使您可以选择服务器用于验证用户的数据库。 默认值为 keyfile 。
验证方法	<ul style="list-style-type: none">■ 基本—使用 HTTP 基本方法从客户机获取验证信息。仅当为服务器启用了 SSL 后，才会通过网络对用户名和密码进行加密。■ SSL—使用客户机证书来验证用户。要使用此方法，必须为服务器启用 SSL。启用加密后，您可以结合使用“基本”和“SSL”方法。■ 摘要—使用一种验证机制，使得浏览器能够基于用户名和密码进行验证，而不必以明文形式发送用户名和密码。浏览器使用用户的密码和 Web Server 提供的某些信息，利用 MD5 算法来创建摘要值。请注意，要使用“摘要”，基础 auth-db 也必须支持摘要。这表示要么是使用摘要文件的文件 auth-db，要么是 LDAP auth-db（仅当安装了摘要验证插件）。■ 其他—使用通过访问控制 API 创建的自定义方法。

表 10-2 WebDAV 验证数据库属性 (续)

属性	说明
验证提示文本	<p>验证提示选项使您可以输入要在验证对话框中显示的消息文本。您可以使用此文本来说明用户需要输入的内容。基于不同的浏览器，用户大约会看到该提示的前 40 个字符。</p> <p>Web 浏览器通常会高速缓存用户名和密码，并将它们与提示文本关联。当用户访问具有相同提示的服务器的文件和目录时，不必再次输入用户名和密码。如果您希望用户对特定文件和目录再次进行验证，只需更改该资源的 ACL 的提示。</p>

在启用了 WebDAV 的服务器上使用源 URI 和 Translate:f 标头

WebDAV 方法处理资源或集合的源。HTTP 方法（例如 GET 和 PUT）会被 WebDAV 协议重载，因此，使用这些方法的请求可以是对资源的源代码的请求，也可以是对资源的内容的请求。

Microsoft 和许多其他 WebDAV 供应商通过以下方法解决了此问题，即随请求发送一个 Translate:f 标头来告知服务器所请求的是源代码。为了可与流行的 WebDAV 客户机 Microsoft WebFolders 交互操作，Sun Java System Web Server 7.0 将 Translate:f 标头识别为对资源的源代码的请求。针对不发送 Translate:f 标头的客户机，Sun Java System Web Server 定义了一个源 URI。

对于启用了 WebDAV 的集合，对 URI 的请求将检索资源的内容（输出），而对源 URI 的请求将检索资源的源代码。带有 Translate:f 标头的 URI 请求被看作是对源 URI 的请求。

请注意，默认情况下，所有对资源的源代码的访问均被 dav-src ACL 拒绝，特定于服务器实例的 ACL 文件中包含以下声明：

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

用户可以通过添加对源 URI 的访问权限来允许另一个用户对源代码进行访问。

锁定和解除锁定资源

Sun Java System Web Server 允许服务器管理员锁定资源，以实现对该资源的有序访问。通过使用锁定，可以确保当一个用户正在访问某个特定的资源时，其他用户不会修改同一资源。这样就解决了多用户共享服务器上的资源时的“丢失更新”问题。由服务器维护的锁定数据库将跟踪客户机发出和正在使用的锁定令牌。

Sun Java System Web Server 支持 opaqueLocktoken URI 方案，该方案被设计为在所有资源中始终是唯一的。它使用了通用唯一标识符 (Universal Unique Identifier, UUID) 机制，如 ISO-1157.08 中所述。

Sun Java System Web Server 可识别两种锁定机制：

- 互斥锁定。
- 共享锁定。

互斥锁定

互斥锁定是仅对单个用户授予资源访问权限的锁定。只有删除对某个资源的互斥锁定后，其他用户才能访问该资源。

事实证明互斥锁定这种资源锁定机制有时太死板，代价也很大。例如，当程序崩溃或锁定所有者忘记解除锁定资源时，便需要通过锁定超时或管理员来删除互斥锁定。

共享锁定

共享锁定允许多个用户接收资源锁定。这样，任何具有适当访问权限的用户都可以获得锁定。

使用共享锁定时，锁定所有者可以使用任何其他通信通道来协调工作。共享锁定的目的是让协作者知道谁正在使用同一资源。

最小锁定超时

您可以通过在 `server.xml` 文件中配置 DAV 或 DAVCOLLECTION 对象的 `minlocktimeout` 属性值来控制锁定。`minlocktimeout` 属性指定锁定的最小生命周期（秒）。此值表示锁定被自动删除之前某元素被锁定的时间。

这是一个可选属性。如果此值被设置为 -1，锁定将永远不会过期。如果将此值设置为 0，则可以使用请求中指定的 `Timeout` 标头锁定集合中的所有资源。

如果未指定 `Timeout` 标头，资源将被锁定且永远不会超时。如果将请求中的 `Timeout` 标头值设置为 `Infinite`，资源将同样被锁定且永远不会超时。

如果对 WebDAV 资源的请求的 `Timeout` 标头值等于或大于 `server.xml` 中指定的 `minlocktimeout` 值，则资源的锁定时间将是请求中指定的时间。

但是，如果请求中的 `Timeout` 标头值小于 `server.xml` 中指定的 `minlocktimeout` 值，则使用 `server.xml` 文件中指定的 `minlocktimeout` 值来锁定资源。

下表列出了 Sun Java System Web Server 处理锁定请求的方式：

表 10-3 Sun Java System Web Server 处理锁定请求的方式

如果请求中的Timeout标头被设置为：	则资源：
Infinite	被锁定且超时值被设置为 -1（无限）
无	被锁定且超时值被设置为 -1（无限）
Second-xxx	<ul style="list-style-type: none">■ 如果 xxx 等于或大于 server.xml 中设置的 minlocktimeout 值，则使用 xxx 值来锁定。或者■ 如果 xxx 小于 server.xml 中设置的 minlocktimeout 值，则使用 server.xml 中指定的 minlocktimeout 值来锁定。

注 - 使用 CLI

要通过 CLI 设置锁定过期，请执行以下命令：

```
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
```

请参见 CLI 参考 expire-lock(1)。

上述示例中，opaque-token 指定要设置为过期的锁定的 ID。

要通过 CLI 显示现有锁定，请执行以下命令：

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

请参见 CLI 参考 list-locks(1)。

使用 Java 和 Web 应用程序

本章介绍编辑虚拟服务器的 Java 设置的过程。您可以通过管理控制台或 wadm 命令行工具编辑 Java 设置。本章也介绍了可以在 Sun Java System Web Server 中配置的各种 Java 资源。

同时本章还介绍了如何在 Sun Java System Web Server 中部署 Java Web 应用程序。

- 第 147 页中的 “配置 Java 以使用 Sun Java System Web Server”
- 第 148 页中的 “设置 Java 类路径”
- 第 149 页中的 “配置 JVM”
- 第 151 页中的 “部署 Java Web 应用程序”
- 第 153 页中的 “配置 Servlet 容器”
- 第 154 页中的 “配置服务器生命周期模块”
- 第 156 页中的 “配置 Java 资源”
- 第 165 页中的 “配置 SOAP 验证提供商”
- 第 167 页中的 “配置会话复制”
- 第 169 页中的 “管理验证领域”

配置 Java 以使用 Sun Java System Web Server

本节介绍如何为选定的配置启用 Java 并设置 Java 主目录变量。

▼ 为配置启用 Java

1 选择配置。

从配置列表中选择相应的配置。单击 “配置” 选项卡以获取可用配置的列表。

2 单击 “Java” > “常规 ” 选项卡。

3 单击“启用 Java”复选框。

为配置启用/禁用 Java 支持。启用 Java 将使服务器可以处理 Java 应用程序。

4 设置 Java 主目录。

指定 Java SE 的位置。指定绝对路径或相对于服务器的 config 目录的路径。

5 设置粘性附加。

指定服务器是否将每个 HTTP 请求处理线程仅附加至 JVM 一次（否则，服务器会在每个请求上附加/拆离 HTTP 请求处理线程）。

注 - 使用 CLI

要为配置启用 Java，请执行以下命令。

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

请参见 CLI 参考 enable-java(1)。

设置 Java 类路径

本节介绍如何为选定的配置添加 JVM 类路径。

▼ 设置 Java 类路径

1 选择配置。

从配置列表中选择相应的配置。单击“配置”选项卡以获取可用配置的列表。

2 单击“Java”>“路径设置”选项卡。

编辑以下参数：

- **忽略环境类路径**—默认情况下处于启用状态。
- **类路径前缀**—系统类路径的前缀。仅当要覆盖系统类（例如，XML 解析器类），才可在系统类路径前添加前缀。**使用此参数时应谨慎。**
- **服务器类路径**—包含服务器类的类路径。只读列表。
- **类路径后缀**—附加到服务器类路径。
- **本地库路径前缀**—操作系统本地库路径的前缀。

- **字节码预处理程序类**—用于实现 `com.sun.appserv.BytecodePreprocessor` 的类的全限定名称。运行时类程序设备通常会通过预处理机制来执行，方法是在 JVM 加载分析和监视工具之前，使用类预处理程序在 Java 类中所需的位置插入程序设备代码。为此，应将类预处理程序与类加载器配合使用。

配置 JVM

要在管理界面中设置 JVM 命令行选项，请执行以下任务：

▼ 配置 JVM

- 1 **选择配置。**
从配置列表中选择相应的配置。单击“配置”选项卡以获取列表。
- 2 **单击“Java”>“JVM 设置”选项卡。**
配置 JVM 的设置。

添加 JVM 选项

通过指定此处相应的值可以添加/删除命令行 JVM 选项。

单击“添加 JVM 选项”按钮以添加 JVM 选项。

JVM 选项的一些示例如下：`-Djava.security.auth.login.config=login.conf`，`-Djava.util.logging.manager=com.ipplanet.ias.server.logging.ServerLogManager` and `-Xms128m -Xmx256m`

注—使用 CLI

要通过 CLI 添加 JVM 选项，请执行以下命令。

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

请参见 CLI 参考 `create-jvm-options(1)`。

添加 JVM 事件探查器

JVM 事件探查器可帮助您诊断和解决 Java 应用程序中的性能问题、内存泄漏、多线程问题和系统资源使用问题，以确保您的应用程序获得最高级别的稳定性和可伸缩性。

▼ 添加 JVM 事件探查器

- 1 选择配置。
从配置列表中选择相应的配置。单击“配置”选项卡以获取可用配置的列表。
- 2 单击“Java”>“JVM 设置”选项卡。
- 3 单击“事件探查器”部分下的“新建”按钮。
- 4 提供以下参数的值：
 - 名称—提供新 JVM 事件探查器的简短名称。
 - 已启用—确定是否在运行时启用事件探查器。
 - 类路径—提供有效的事件探查器类路径。（可选）。
 - 本地库路径—提供有效的本地库路径。（可选）。
 - JVM 选项—您可以为 CLI 指定其他 JVM 选项。

注—使用 CLI

要通过 CLI 添加 JVM 事件探查器，请执行以下命令：

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

请参见 CLI 参考 `create-jvm-profiler(1)`。

为服务器启用 Java 调试

JVM 可以在调试模式下启动，并可以附加 JPDA（Java Platform Debugger Architecture，Java 平台调试器体系结构）调试器。启用调试时，您可同时启用本地调试和远程调试。

Sun Java System Web Server 的调试功能以 JPDA 软件为基础。要启用调试，请执行以下步骤。

▼ 启用 JVM 调试

- 1 选择配置。
从配置列表中选择相应的配置。单击“配置”选项卡以获取可用配置的列表。
- 2 单击“Java”>“JVM 设置”选项卡。
- 3 选中“调试 Java 设置”下的“启用调试”复选框。

- 4 如有必要，可通过单击“新建”按钮来提供 JVM 选项。

默认 JPDA 选项如下：

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

如果您替换 `suspend=y`，则 JVM 将以暂停模式启动，并在调试器附加到其之前一直处于暂停状态。如果您希望 JVM 启动后立即启动调试，则该操作非常有用。要指定将 JVM 附加到调试器时要使用的端口，请指定 `address=port_number`。请查阅 JPDA 文档，以获得调试选项的列表。

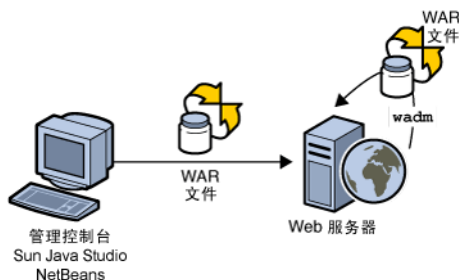
部署 Java Web 应用程序

添加 Web 应用程序

可以将 Web 应用程序部署到任何现有虚拟服务器上。

▼ 部署 Web 应用程序

- 开始之前
- 您已经确定了需要在其中部署 Web 应用的虚拟服务器。
 - 您具有 Web 应用程序归档文件（.war 文件）或知道 Web 应用程序在服务器中的路径。



Web 应用程序可以通过 wadm、管理控制台和其他支持的 IDE 进行部署。

- 1 单击“服务器配置”下的虚拟服务器选项卡以部署 Web 应用程序。
- 2 选择需要在上部署 Web 应用程序的虚拟服务器。
- 3 单击“Web 应用程序”>“新建”按钮。
- 4 指定 Web 应用程序软件包。

如果需要上载 Web 应用程序归档文件，请单击“浏览”按钮并选择所需的归档文件。您也可以指定位于服务器中的 Web 应用程序归档文件（可选）。

5 指定 Web 应用程序的 URI。此 URI 将是应用程序的上下文根，它是相对于服务器主机的。

6 提供有关 Web 应用程序的简短说明。

7 启用/禁用 JSP 预编译。

启用该指令将允许对 Web 应用程序中的所有 JSP 进行预编译，以提高性能。

8 启用应用程序。

将某个 Web 应用程序的状态设置为“已禁用”时，该应用程序在请求时将不可用。但是，您可以随时切换此选项，而无需将该应用程序重新部署到实例上。

9 部署应用程序。

单击“部署”按钮以部署 Web 应用程序。

您可以使用指定的上下文根访问此应用程序。例如

`http://<your-server>:<port>/<URI>`

注 - 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
```

请参见 CLI 参考 `add-webapp(1)`。

部署 Web 应用程序目录

使用 `-file-on-server` 选项可以将管理服务器主机上的目录部署到配置中。执行以下命令：

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

部署期间预编译 JSP

要在部署 Web 应用程序时预编译该应用程序中的 JSP，请执行带有 `-precompilejsp` 选项的命令，如下所示：

```
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

本节介绍配置 servlet 容器的过程。

▼ 设置 Servlet 容器

- 1 选择配置。
从配置列表中选择相应的配置。单击“配置”选项卡以获取列表。
- 2 单击 "Java">“Servlet 容器”。

Servlet 容器全局参数

下表介绍 servlet 容器页面中的可用参数。

表 11-1 Servlet 容器参数

参数	说明
日志级别	servlet 容器的日志详尽程度。值可以为 finest（最详尽）、finer、fine、info、warning、failure、config、security 或 catastrophe（最不详尽）。
动态重新加载时间间隔	此参数定义了一个时间段，经过此时间段后，服务器就会检查已部署的 Web 应用程序是否发生了修改。值的范围从 1 至 60，或者如果禁用动态重新加载，则为 -1。
匿名角色	分配给所有主体的默认角色或匿名角色的名称。默认角色为 ANYONE。
Servlet 池大小	实例化每个 SingleThreadedServlet 的 servlet 实例数。值的范围从 1 至 4096。
分发程序最大深度	允许嵌套请求分发的 Servlet 容器的最大深度。值的范围可介于 0 至 2147.0483647.0 之间。默认值为 20。
允许交叉上下文	是否允许请求分发程序向其他上下文分发。默认值为 false。
编码 Cookie	servlet 容器是否对 cookie 值进行编码。默认值为 true。
重复使用会话 ID	为该客户机创建新会话时是否重复使用任何现有会话 ID 编号。默认值为 false。

表 11-1 Servlet 容器参数 (续)

参数	说明
安全会话 Cookie	Dynamic/True/False。此参数控制在什么条件下将 JSESSIONID cookie 标记为安全。仅当通过安全连接 (HTTPS) 收到请求时，才会使用 dynamic（默认设置）将 Cookie 标记为安全。 选择 True 可始终将其标记为安全，选择 false 可从不将其标记为安全。

配置服务器生命周期模块

Java Server 生命周期模块是用于侦听服务器生命周期事件的 Java 类，以便在发生服务器事件（如启动或停止）时执行特定任务。

服务器支持在 Web 服务器环境中运行短时间或长时间的基于 Java 的任务。这些任务在服务器启动时自动启动，并在服务器关闭时收到通知。因此，现在您可以立即得知诸如实例化 singleton 和 RMI 服务器等任务。

服务器生命周期的简短说明如下。

服务器生命周期介绍

- **初始化**—此阶段包括读取配置、初始化内置子系统；命名、安全性和日志记录服务；以及创建 Web 容器。
- **启动**—此阶段包括加载和初始化部署的应用程序。
- **服务**—服务器已就绪，可以处理请求。
- **关闭**—此阶段将停止和销毁加载的应用程序。系统准备关闭。
- **终止**—此阶段将终止内置子系统和服务器运行时环境。此阶段后将不会再有任何活动。
- **重新配置**—服务器瞬态，服务器线程在此状态下进行动态重新配置（此时服务器处于服务状态）。此阶段在服务器生命周期中可出现多次。

▼ 添加生命周期模块

- 1 **选择配置。**
从配置列表中选择相应的配置。要查看配置列表，请单击“配置”选项卡。
- 2 **单击 "Java" > “生命周期模块” 选项卡。**

3 单击“新建”按钮。

提供以下参数的值：

- **名称**—为新的生命周期模块提供一个有效的唯一名称。
- **已启用**—如果要启用该生命周期模块，请使用此选项。
- **类名称**—全限定 Java 类名称。该类应实现 `com.sun.appserv.server.LifecycleListener` 接口。有关使用此接口的更多信息，请参见**开发者指南**。
- **类路径**—可选。您可以指定侦听器类的类路径。
- **加载顺序**—大于 100。加载生命周期事件侦听器的顺序（按照数值顺序）。建议选择一个大于或等于 100 的加载顺序，以避免与内部生命周期模块发生冲突。
- **装入失败时**—如果启用该选项，服务器将不会把侦听器类抛出的异常视为致命错误，因此可继续执行正常启动。默认情况下，禁用该选项。
- **说明**—提供有关生命周期模块的简短说明。
- **属性**—属性可用于将参数传送给 Java 生命周期模块。要添加新属性，请单击“添加属性”按钮并在名称、值和说明字段中输入相应的文本。



注意—服务器生命周期侦听器类是从主服务器线程同步调用的，因此必须采取额外的预防措施以避免这些侦听器类阻塞服务器。如果合适，这些侦听器类可创建线程，但是必须在关闭/终止阶段时停止这些线程。

▼ 删除生命周期模块

1 选择配置。

从配置列表中选择相应的配置。要查看配置列表，请单击“配置”选项卡。

2 单击“Java”>“生命周期模块”选项卡。

3 选择生命周期模块，然后单击“删除生命周期模块”按钮。

注-使用 CLI

以下示例说明了如何为由类 `com.MyLifecycleModule` 实现的配置 `test` 创建名为 `myLifecycleModule` 的 Java 生命周期模块。

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

请参见 CLI 参考 `create-lifecycle-module(1)`。

要列出 Java 生命周期模块，请执行以下命令：

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

请参见 CLI 参考 `list-lifecycle-modules(1)`。

要向 Java 生命周期模块中添加属性，请执行以下命令：

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

请参见 CLI 参考 `create-lifecycle-module-userprop(1)`。

要修改 Java 生命周期模块属性，请执行以下命令：

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

请参见 CLI 参考 `set-lifecycle-module-prop(1)`。

配置 Java 资源

Web 应用程序可以访问多种资源，例如资源管理器、数据源（如 SQL 数据源）、邮件会话和 URL 连接工厂。Java EE 平台通过 Java 命名和目录接口 (Java Naming and Directory Interface, JNDI) 服务将这些资源提供给应用程序。

Sun Java System Web Server 允许您创建和管理以下 Java EE 资源：

- JDBC 数据源。
- JDBC 连接池。
- Java 邮件会话。
- 自定义资源。
- 外部 JNDI 资源。

配置 JDBC 资源

JDBC 数据源是一种 Java EE 资源，可以通过 Sun Java System Web Server 进行创建和管理。

JDBC API 是用于连接关系数据库系统的 API。JDBC API 包含两部分：

- 一个应用程序级接口，应用程序组件使用它来访问数据库。
- 一个服务提供商接口，用于将 JDBC 驱动程序连接到 Java EE 平台。

JDBC 数据源对象是用 Java 编程语言实现的数据源。从本质上讲，数据源是一种存储数据的工具。数据源可能像大型公司的综合数据库一样复杂，也可能像仅包含行和列的文件一样简单。JDBC 数据源是一种 Java EE 资源，可以使用 Sun Java System Web Server 进行创建和管理。

JDBC API 使用标准的 SQL 数据库访问接口为 Java 提供了一组类，可以确保对多种关系数据库的统一访问。

事实上，使用 JDBC 可以将 SQL 语句发送到任何数据库管理系统 (database management system, DBMS)。它可以用作关系 DBMS 和对象 DBMS 的接口。

添加 JDBC 资源

要通过 CLI 添加 JDBC 资源，请执行以下命令：

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

请参见 CLI 参考 `create-jdbc-resource(1)`。

上述示例中，`com.pointbase.jdbc.jdbcDataSource` 表示 JDBC 驱动程序类。

有关支持的 JDBC 驱动程序列表，请参见第 157 页中的“[Sun Java System Web Server 支持的 JDBC 驱动程序](#)”。

Sun Java System Web Server 支持的 JDBC 驱动程序

下表提供了常用 JDBC 驱动程序及其在添加新 JDBC 资源时需要配置的属性。请参见第 160 页中的“[添加新 JDBC 资源](#)”。

表 11-2 常用和受支持的 JDBC 驱动程序列表

驱动程序	类名	属性
Oracle 驱动程序	oracle.jdbc.pool.OracleDataSource	<ul style="list-style-type: none">urluserpassword
Oracle SJS JDBC 驱动程序	com.sun.sql.jdbcx.oracle.OracleDataSource	<ul style="list-style-type: none">serverNameportNumberuserpasswordSID
DB2 IBM 驱动程序	com.ibm.db2.jdbc.DB2DataSource	<ul style="list-style-type: none">serverNamedatabaseNameportNumberuserpassworddriverType
DB2 SJS JDBC 驱动程序	com.sun.sql.jdbcx.db2.DB2DataSource	<ul style="list-style-type: none">databaseNamelocationNamepackageNamepasswordportNumberserverNameuser
MS SQLServer 驱动程序	com.ddtek.jdbcx.sqlserver.SQLServerDataSource	<ul style="list-style-type: none">databaseNamepassworduserserverNameportNumber
MS SJS JDBC 驱动程序	com.sun.sql.jdbcx.sqlserver.SQLServerDataSource	<ul style="list-style-type: none">databaseNamepassworduserserverNameportNumber
Sybase 驱动程序	com.sybase.jdbcx.SybDataSource	<ul style="list-style-type: none">databaseNamepasswordportNumberserverNameuser

表 11-2 常用和受支持的 JDBC 驱动程序列表 (续)

驱动程序	类名	属性
Sybase SJS JDBC 驱动程序	com.sun.sql.jdbcx.sybase. SybaseDataSource	<ul style="list-style-type: none">■ databaseName■ password■ user■ portNumber■ serverName
MySQL MM 驱动程序	org.gjt.mm.mysql.jdbc2.optional. MysqlDataSource	<ul style="list-style-type: none">■ serverName■ port■ databaseName■ user■ password
Informix 驱动程序	com.informix.jdbcx.IfxDataSource	<ul style="list-style-type: none">■ portNumber■ databaseName■ IfxIFXHOST (运行 Informix 数据库的计算机的 IP 地址或主机名)■ serverName■ user■ password
Informix SJS JDBC 驱动程序	com.sun.sql.jdbcx.informix. InformixDataSource	<ul style="list-style-type: none">■ databaseName■ informixServer (要连接的 Informix 数据库服务器的名称)■ password■ portNumber■ severName
PostgreSQL 驱动程序	org.postgresql.ds. PGSimpleDataSource	<ul style="list-style-type: none">■ serverName■ databaseName■ portNumber■ user■ password
Apache Derby 驱动程序	org.apache.derby.jdbc. EmbeddedDataSource	<ul style="list-style-type: none">■ databaseName■ user■ password

管理 JDBC 资源

▼ 添加新 JDBC 资源

- 1 选择配置。
从配置列表中选择相应的配置。单击“配置”选项卡以获取列表。
- 2 单击“Java”>“资源”选项卡。
- 3 单击“JDBC 资源”部分中的“新建”按钮。
- 4 选择驱动程序提供商。
为 JNDI 名称指定一个唯一的值并从可用列表中选择 JDBC 驱动程序供应商。
- 5 提供 JDBC 资源属性。
根据上一步中选择的 JDBC 驱动程序供应商，系统将会自动填充驱动程序的类名称和 JDBC 资源属性。
- 6 检查。
查看摘要并单击“完成”以创建新 JDBC 资源。

管理 JDBC 连接池

配置 JDBC 连接池

在 Sun Java System Web Server 7.0 中，JDBC 连接池是通过 `jdbc-resource` 元素进行配置的。按照以下所列步骤进行操作可以配置最简单的连接池。在以下示例中，连接池将使用 Oracle JDBC 驱动程序。

▼ 创建 JDBC 连接池

- 1 启动 wadm。
- 2 创建 JDBC 资源。
使用基本配置创建一个 JDBC 资源。其他属性可用于微调连接池。有关更多属性和示例的信息，请参阅手册页。

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


3 配置特定于供应商的属性。

使用相应的属性配置特定于驱动程序供应商的属性。在以下示例中，属性 url、user 和 password 被添加到 JDBC 资源中。

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

4 启用连接验证。

可以对连接池启用连接验证。如果使用此选项，则连接在传递到应用程序之前将被验证。这样，如果由于网络出现故障或数据库服务器崩溃造成数据库不可用，Web 服务器将自动重新建立数据库连接。连接验证将引起额外开销，并且会导致性能稍有下降。

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

5 更改默认池设置。

在以下实例中，将更改最大连接数。

```
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 驱动程序的 Jar 文件。

需要将实现该驱动程序的类提供给服务器。可以通过两种方法执行该操作：

- 将驱动程序的 jar 文件复制到服务器实例 lib 目录中。这是最简单的方法，因为实例 lib 目录中包括的 jar 文件将会自动加载并对服务器可用。
- 修改 JVM 的类路径后缀以包含 JDBC 驱动程序的 jar 文件。

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

8 在 Web 应用程序中使用。

- 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>
```

- 使用连接池。

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

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

注册自定义资源

您可以通过执行以下任务在实例中注册自定义资源。

▼ 添加自定义资源。

- 1 选择配置。
从配置列表中选择相应的配置。单击“配置”选项卡以获取列表。
- 2 单击“Java”>“资源”选项卡。
- 3 单击“自定义资源”部分中的“新建”按钮。

自定义资源的属性

下表介绍了用于创建自定义资源的可用属性。

表 11-3 自定义资源的属性

属性	说明
JNDI 名称	为自定义资源提供一个唯一的 JNDI 名称。
已启用	确定是否在运行时启用 JDBC 资源。
资源类型	该资源的全限定类型。

表 11-3 自定义资源的属性 (续)

属性	说明
工厂类	实例化此类资源的类。用户编写的工厂类（用于实现 <code>javax.naming.spi.ObjectFactory</code> ）的全限定名称。
说明	提供自定义资源的简短说明。
属性	通过单击“添加属性”按钮可以有选择地提供 CLI 属性。

注 - 使用 CLI

要通过 CLI 创建自定义资源，请执行以下命令：

```
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
```

请参见 CLI 参考 `create-custom-resource(1)`。

使用外部 JNDI 资源

创建外部 JNDI 资源

使用该选项可以创建外部 Java 命名和目录接口 (Java Naming and Directory Interface, JNDI) 资源。您需要一个外部 JNDI 资源以访问存储在内部 JNDI 系统信息库中的资源。

▼ 添加外部 JNDI 资源

- 1 选择配置。
从配置列表中选择相应的配置。单击“配置”选项卡以获取列表。
- 2 单击 "Java" > “资源”选项卡。
- 3 单击“外部 JNDI”部分中的“新建”按钮。

外部 JNDI 资源的属性

下表介绍了添加新外部 JNDI 资源时的可用属性。

表 11-4 外部 JNDI 资源的属性

属性	说明
JNDI 名称	提供新的外部 JNDI 资源的唯一名称。
已启用	确定是否在运行时启用外部 JNDI 资源。
外部 JNDI 名称	外部 JNDI 资源的名称。
资源类型	该资源的全限定类型。
工厂类	实例化此类资源的类。
说明	提供自定义资源的简短说明。
属性	通过单击“添加属性”按钮可以有选择地提供 CLI 属性。

注 - 使用 CLI

要通过 CLI 创建外部 JNDI 资源，请执行以下命令：

```
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
```

请参见 CLI 参考 create-external-jndi-resource(1)。

配置邮件资源

JMS 目标是 Java EE 资源，可以通过 Sun Java System Web Server 进行创建和管理。

许多 Internet 应用程序都需要具有发送电子邮件通知的能力，因此，Java EE 平台提供了 JavaMail API 以及一个使应用程序组件可以发送 Internet 邮件的 JavaMail 服务提供商。

▼ 添加邮件资源

- 1 选择配置。
从配置列表中选择相应的配置。单击“配置”选项卡以获取列表。
- 2 单击 "Java" > “资源 ” 选项卡。
- 3 单击“ 邮件资源 ” 部分中的“ 新建 ” 按钮。

邮件资源的属性

下表介绍了添加新邮件资源时的可用属性。

表 11-5 邮件资源的属性

属性	说明
JNDI 名称	为新的邮件资源提供一个唯一的名称。
已启用	确定是否在运行时启用该邮件资源。
用户	在邮件服务器中注册的有效用户名。
发件人	服务器发送邮件的电子邮件地址。
主机	邮件服务器的主机名/IP 地址。
存储协议	用于检索邮件的协议。
存储协议类	实现存储协议的存储服务提供商。实现存储协议的类的全限定类名。默认类为 <code>com.sun.mail.imap.IMAPStore</code> 。
传输协议	用于发送邮件的协议。
传输协议类	实现传输协议的传输服务提供商。实现传输协议的类的全限定类名。默认类为 <code>com.sun.mail.smtp.SMTPTransport</code> 。

注 - 使用 CLI

要创建邮件资源，请执行以下命令：

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

请参见 CLI 参考 `create-mail-resource(1)`。

配置 SOAP 验证提供商

容器规范的 Java 验证服务提供商接口定义了一种标准的服务提供商接口，通过该接口可将验证机制提供商与容器进行集成。您可以使用管理控制台添加新的 SOAP 验证提供商。

▼ 添加 SOAP 验证提供商

- 1 选择配置。
从配置列表中选择相应的配置。单击“配置”选项卡以获取列表。
- 2 单击"Java">“Web 服务”选项卡。
- 3 单击“SOAP 验证提供商”部分中的“新建”按钮。

SOAP 验证提供商参数

下表介绍了新建 SOAP 验证提供商页面中的可用参数。

表 11-6 SOAP 验证提供商参数

参数	说明
名称	输入新的 SOAP 验证提供商的简短名称。
类名	实现此提供商的类。实现 javax.security.auth.XXX 的类的全限定类名。
请求验证源	该属性定义邮件层的发件人验证（例如用户名/密码）或内容验证（例如应用于请求邮件的数字签名）要求。该值（验证策略）可为 sender 或 content。如果没有指定此参数，则不需要对请求进行源验证。
请求验证收件人	该属性定义在邮件层向邮件发件人验证邮件收件人的要求，例如，通过 XML 加密。值可以为 before-content 或 after-content。
响应验证源	该属性定义邮件层的发件人验证（例如用户名/密码）或内容验证（例如应用于响应邮件的数字签名）要求。该值（验证策略）可为 sender 或 content。如果没有指定此参数，则不需要对响应进行源验证。
响应验证收件人	该属性定义在邮件层向邮件发件人验证响应邮件收件人的要求，例如，通过 XML 加密。
属性	通过单击“添加属性”按钮可以提供其他 CLI 属性。

注-使用 CLI

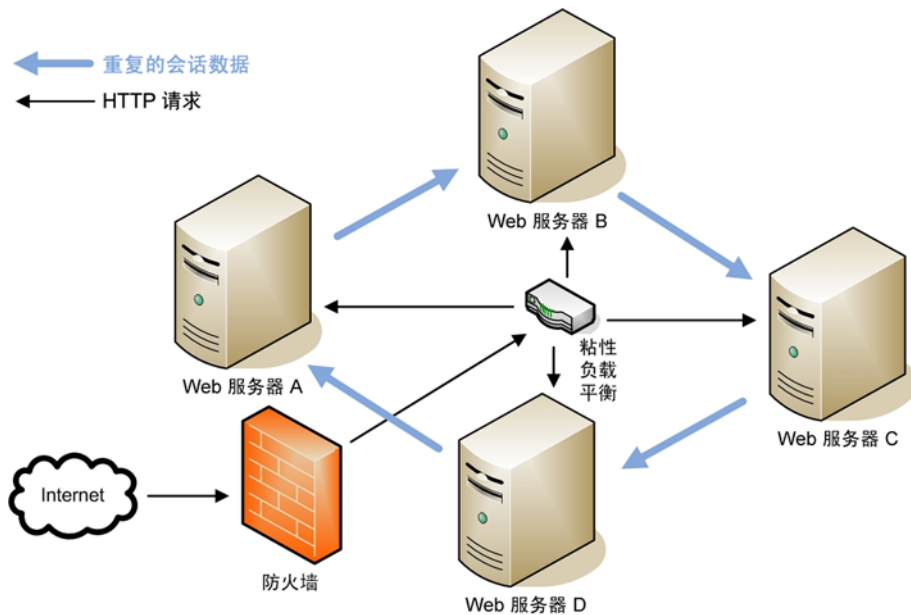
要使用 CLI 添加 SOAP 验证提供商，请执行以下命令：

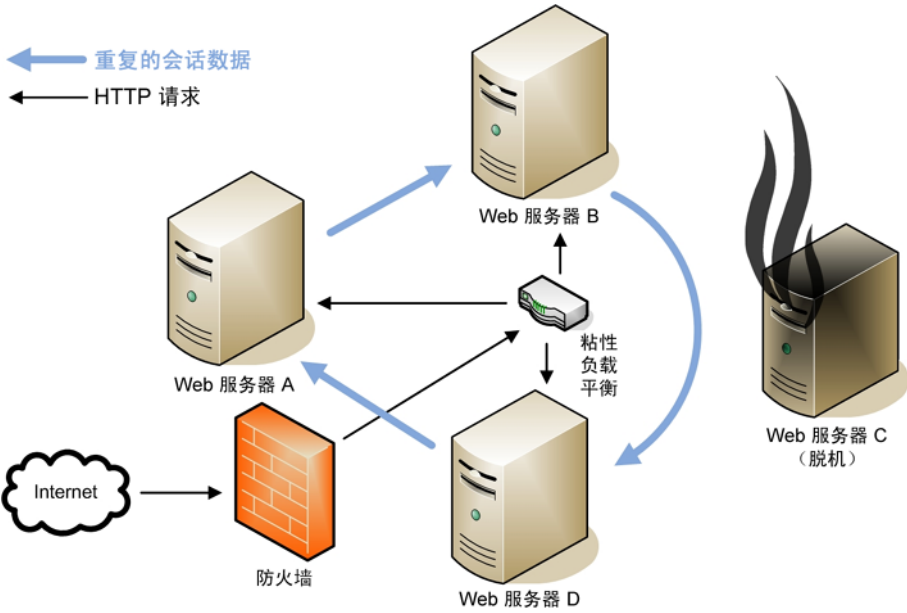
```
wadm> create-soap-auth-provider --user=admin --password-file=admin.pwd  
--host=serverhost --port=8989 --config=config1  
--class-name=javax.security.auth.soapauthprovider soap-auth
```

请参见 CLI 参考 create-soap-auth-provider(1)。

配置会话复制

Sun Java System Web Server 7.0 支持为 Web 应用程序提供高可用性的会话复制功能。会话复制通过将 HTTP 会话从一个实例复制到同一群集中的另一服务器实例来实现这一目的。因此，每个 HTTP 会话在远程实例上均有一个备份副本。如果发生故障，致使群集中的某个实例不可用，则群集仍可保持会话的持续性。





上图所示为设置了一个反向代理的四个节点之间发生会话复制的典型方案。请注意，Web Server C 脱机时，会话数据将从 Web Server B 复制到 Web Server D。

设置会话复制

本节介绍为选定的配置设置会话复制属性的步骤。

▼ 设置会话复制

- 1 选择配置。
从配置列表中选择相应的配置。单击“配置”选项卡以获取列表。
- 2 单击 "Java">“会话复制”。

修改会话复制参数

下表介绍了会话复制页面中的可用参数。

表 11-7 会话复制参数

参数	说明
端口	Administration Server 侦听的端口号。默认端口为 8888。
已启用	为选定配置启用会话复制。
加密	复制之前是否加密会话数据。默认值为 false。
密码	群集成员用于复制会话数据的加密套件（算法、模式、填充）。
Getattribute 触发复制	对 HttpSession.getAttribute 方法的调用是否应导致对会话进行备份。默认值为 true。
副本搜索最大实例数	尝试查找会话备份时应联系的最大实例数。值的范围介于 1 至 2147.0483647 之间，没有限制时为 -1。
启动搜索超时	实例尝试联系其指定的备份实例所花费的最长时间（以秒为单位）。值的范围从 0.001 至 3600。
Cookie 名称	输入跟踪哪个实例拥有会话的 cookie 名称。

管理验证领域

基于 Java EE 的安全模型提供了可识别和验证用户的安全领域。

验证进程通过 Java 领域验证用户。领域包括一组用户、可选的组映射以及可用来对验证请求进行验证的验证逻辑。验证请求通过已配置的领域验证并建立安全性上下文后，便可将该身份应用于所有后续的授权决策。

注 - Java 领域与 auth-db（验证数据库）类似，不同的是，auth-db 由 ACL 引擎（基于您的 ACL 文件的规则）使用，而 Java 领域由 Java Servlet 访问控制规则（在每个 Web 应用程序的 web.xml 文件中指定的）使用。

服务器实例可能具有多种配置领域。server.xml 文件的 auth-realm 元素中提供了配置信息。

下表定义了 Sun Java System Web Server 7.0 支持的不同领域类型。

表 11-8 领域类型

领域	说明
File	<p>file 领域是首次安装 Sun Java System Web Server 时的默认领域。该领域的设置非常简便，为开发者提供了很大方便。</p> <p>file 领域使用存储在文本文件中的用户数据来验证用户。Java 领域与 auth-db（验证数据库）类似，不同的是，auth-db 由 ACL 引擎（基于您的 ACL 文件的规则）使用，而 Java 领域由 Java Servlet 访问控制规则（在每个 Web 应用程序的 web.xml 中指定的）使用。</p>
LDAP	<p>ldap 领域使您可以使用 LDAP 数据库获取用户安全性信息。LDAP 目录服务是带有唯一标识符的属性的集合。ldap 领域非常适合部署到生产系统中。</p> <p>要使用 ldap 领域验证用户，必须在 LDAP 目录中创建所需的用户。您可以通过 Administration Server 的“用户和组”选项卡来执行此操作。此外，还可以通过 LDAP 目录产品的用户管理控制台来执行此操作。</p>
PAM	<p>PAM（也称为 Solaris）领域将验证委派给 Solaris PAM 堆栈。与 PAM auth-db 一样，该领域仅在 Solaris 9 和 Solaris 10 上受支持，并且必须以超级用户身份运行服务器实例。</p>
Certificate	<p>certificate 领域支持 SSL 验证。Certificate 领域在 Sun Java System Web Server 的安全性上下文中设置用户身份，并将客户机证书的用户数据填入其中。然后，Java EE 容器将基于证书中每个用户的 DN 处理授权进程。此领域会通过 X.509 证书，以 SSL 或 TLS 客户机验证来验证用户。</p>
Native	<p>native 领域是一个特殊的领域，它为基于 ACL 的核心验证模型和 Java EE/Servlet 验证模型提供了联系桥梁。通过将 Native 领域用于 Java Web 应用程序，可以让 ACL 子系统执行验证（而不是让 Java Web 容器执行），并且使此身份可以用于 Java Web 应用程序。</p> <p>调用验证操作时，Native 领域会将此验证委派给核心验证子系统。从用户的角度来看，这实际上相当于 LDAP 领域将验证委派给已配置的 LDAP 服务器。当 Native 领域处理组成员关系查询时，它们同时也被委派给核心验证子系统。对于 Java Web 模块和开发者而言，Native 领域与其他任何可用于 Web 模块的 Java 领域没有任何区别。</p>
自定义	<p>您可以通过可插接式 JAAS 登录模块和领域实现为其他数据库（如 Oracle）建立领域，以满足您的特定需要。</p>

下节介绍添加新验证领域所包含的步骤。

▼ 添加验证领域

1 选择配置。

选择需要添加新验证领域的配置。单击“配置”选项卡并选择相应的配置。

2 单击"Java">"安全性"选项卡。

3 单击“新建验证”按钮。

4 提供领域详细信息。

- **名称**—输入领域的简短名称。该名称用于从文件（例如，web.xml）中引用领域。
- **类**—如果您要配置自定义领域，则输入完整的 Java 类名称可以实现自定义领域。无需输入任何内置领域的类。
- **类型**—选择领域的类型。请参见前面讨论 Java 领域类型的部分。
- **属性**—添加特定于领域的属性。例如，`property name="file" value="instance_dir/config/keyfile"` 和 `property name="jaas-context" value="fileRealm"`。

注 - 使用 CLI

要通过 CLI 添加验证领域，请执行以下命令。

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

请参见 CLI 参考 `create-auth-realm(1)`。

指定内置验证领域类型的名称。类型可以为 `file`、`ldap`、`pam`、`native` 或 `certificate`。

使用搜索集合

服务器包含搜索功能，使用户可以在服务器上搜索文档并将搜索结果显示在 Web 页面上。服务器管理员可以创建用户搜索时使用的文档索引（称为**集合**），也可以自定义搜索界面以满足其用户的需求。

有关查询搜索集合的更多信息，请参阅[搜索联机帮助](#)。

- 第 173 页中的 “关于搜索”
- 第 174 页中的 “配置搜索属性”
- 第 175 页中的 “配置搜索集合”
- 第 177 页中的 “安排集合更新”
- 第 179 页中的 “执行搜索”
- 第 179 页中的 ““搜索” 页面”
- 第 179 页中的 “进行查询”
- 第 180 页中的 “高级搜索”
- 第 181 页中的 “查看搜索结果”
- 第 181 页中的 “自定义搜索页面”

关于搜索

在安装 Sun Java System Web Server 的过程中，已经随其他 Web 组件一起安装了搜索功能。在虚拟服务器级别而不是服务器实例级别配置和管理搜索功能。

在管理控制台的“虚拟服务器”选项卡下的“搜索”选项卡中，您可以：

- 启用和禁用搜索功能
- 创建、修改、删除搜索集合以及为搜索集合重新编制索引
- 为搜索集合创建、修改和删除已安排的维护任务

从管理界面获取的信息存储在 `<server-root>/config/server.xml` 文件中，它们被映射在该文件的 VS 元素内。

服务器管理员可以自定义搜索查询和搜索结果页面。这可能包括使用公司徽标重新设置页面，或者更改搜索结果的显示方式。在以前的版本中，这是通过使用模式文件来实现的。

没有用于搜索的全局“启用”或“禁用”功能，而是提供了一个默认搜索 Web 应用程序，然后在特定的虚拟服务器上启用或禁用该应用程序。此搜索应用程序提供了用于查询集合和查看结果的基本 Web 页面。此搜索应用程序包含样例 JSP，这些样例 JSP 说明了如何使用搜索标记库来建立自定义的搜索界面。

注 - Sun Java System Web Server 不对搜索结果进行访问检查。由于潜在的安全模型和安全领域的数量很多，因此无法通过搜索应用程序执行安全检查及过滤结果。服务器管理员负责确保使用合适的安全机制来保护内容。

配置搜索属性

通过启用服务器附带的搜索应用程序，可以为虚拟服务器启用搜索功能。

注 - 要启用搜索功能，必须启用 Java Web 容器。

确保为您要配置的虚拟服务器启用了 Java 后，请执行以下步骤以启用搜索功能：

1. 单击**配置**选项卡。
2. 从配置列表中选择相应的配置。
3. 单击**虚拟服务器**选项卡。
4. 从虚拟服务器列表中选择虚拟服务器。
5. 单击**搜索**选项卡。
6. 在**搜索应用程序**部分下，单击**已启用**复选框以启用搜索应用程序。

以下列出了您可以配置的其他参数：

- **URI**。如果要使用自定义搜索应用程序，请输入 URI；如果要使用默认搜索应用程序，则无需在此指定值。
- **最大命中次数**。指定搜索查询中检索到的最大结果数目。
- **已启用**。选中此选项以启用默认搜索应用程序。

注-使用 CLI

要通过 CLI 设置搜索属性，请在 CLI 中执行以下命令：

```
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
```

请参见 CLI 参考 `set-search-prop(1)`。

配置搜索集合

搜索要求具备一个包含可搜索数据的数据库，用户将在该数据库中进行搜索。服务器管理员创建此数据库，该数据库称为集合，用来为有关服务器上文档的信息编制索引并存储这些信息。服务器管理员为全部或部分服务器文档编制索引后，就可以使用标题、创建日期和作者等信息进行搜索。

注-关于搜索集合：

- 集合特定于被管理的虚拟服务器
 - 只有在虚拟服务器中可见的文档才显示在管理界面中，并可为其编制索引
 - 服务器上可存在任意数量的集合
 - 搜索集合中的文档不特定于任何一种字符编码，这表示搜索集合可以与多种编码相关联。
-

支持的格式

可以为下列格式的文件编制索引并搜索它们。

1. HTML 文档－.html 和 .htm
2. ASCII 纯文本－.txt
3. PDF

添加搜索集合

要添加新的集合，请执行以下步骤：

1. 单击**配置**选项卡。
2. 从配置列表中选择相应的配置。
3. 单击**虚拟服务器**选项卡。
4. 从虚拟服务器列表中选择虚拟服务器。

5. 单击**搜索**选项卡。
6. 在**搜索集合**部分下，单击**添加搜索集合**按钮以添加新的搜索集合。

以下部分介绍了用于创建新搜索集合的页面中的字段：

1. 提供搜索集合信息

- a. **集合名称**—为搜索集合输入唯一的名称。

注—不允许使用多字节字符作为集合名称。

- b. **显示名称**—（可选）这将在搜索查询页面中作为集合名称进行显示。如果未指定显示名称，则显示名称就是集合名称。
- c. **说明**—（可选）输入说明新集合的文本。
- d. **路径**—您可以在默认位置创建集合，也可以提供一个用于存储集合的有效路径。

2. 提供索引信息

- a. **要编制索引的目录**—输入要为其在集合中编制索引的文档的目录。只有在此虚拟服务器中可见的目录才可编制索引。
- b. **子目录**—输入要为其在集合中编制索引的文档的子目录。子目录路径应该相对于先前指定的目录路径。
- c. **模式**—指定用来选择要为其编制索引的文件的通配符。
请慎重使用通配符模式，以确保只为特定文件编制索引。例如，指定 *.* 可能会导致也为可执行文件和 Perl 脚本文件编制索引。
- d. **子目录**—已启用/已禁用。如果选择此选项，则还将为选定目录的子目录中的文档编制索引。此操作为默认操作。
- e. **默认文档编码**—
集合中的文档不限于单一语言/编码。每次添加文档时，只能指定一种编码；但是，下次将文档添加至集合时，可以选择其他默认编码。

3. 步骤 3：查看摘要

- a. 查看摘要并单击**完成**按钮以添加新的集合。

注-使用 CLI

要通过 CLI 添加搜索集合，请执行以下命令。

```
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
```

请参见 CLI 参考 `create-search-collection(1)`。

删除搜索集合

要删除搜索集合，请执行以下任务：

1. 单击**配置**选项卡。
2. 从配置列表中选择相应的配置。
3. 单击**虚拟服务器**选项卡。
4. 从虚拟服务器列表中选择虚拟服务器。
5. 单击**搜索**选项卡。
6. 在**搜索集合**部分下，选择相应的集合名称并单击**删除**按钮以删除集合。

注-使用 CLI

要通过 CLI 删除搜索集合，请执行以下命令。

```
wadm> delete-search-collection --user=admin --password-file=admin.pwd  
--host=serverhost --port=8989 --config=config1 --vs=config1_vs_1 searchcoll
```

请参见 CLI 参考 `delete-search-collection(1)`。

安排集合更新

您可以安排定期在集合上执行维护任务。可以安排的任务是重新编制索引和更新。可以使用管理界面为特定集合安排任务。您可以指定：

- 要执行的任务（重新编制索引或更新）
- 执行任务的时间
- 在一星期中的哪一天执行任务

要为集合安排事件，请执行以下步骤：

1. 单击**配置**选项卡。
2. 从配置列表中选择相应的配置。

- 3. 单击**虚拟服务器**选项卡。
- 4. 从虚拟服务器列表中选择虚拟服务器。
- 5. 单击**搜索**选项卡。
- 6. 单击**安排的事件**选项卡。
- 7. 在**搜索事件**选项卡下，单击**新建**按钮。

下表介绍了“新建搜索事件进度表”页面中的字段：

表 12-1 字段说明 > 新建搜索事件进度表

字段	说明
集合	从下拉式列表中选择需要为其安排维护任务的集合。 1. 为集合重新编制索引—此类安排的事件将在指定的时间为指定的集合重新编制索引。 2. 更新集合—您可以在创建集合后添加或删除文件。只能添加集合创建期间指定的目录下的文档。如果要删除文档，则只有这些文件的条目及其元数据会从集合中删除。实际上并未从文件系统中删除文件本身。此类安排的事件将在指定的时间更新集合。 3. 模式—指定用来选择要为其编制索引的文件的通配符。 4. 包含的子目录—如果选择此选项，则还将为选定目录的子目录中的文档编制索引。此操作为默认操作。 5. 编码—指定要为其编制索引的文档的字符编码。默认编码为 ISO-8859-1。搜索引擎将尝试从嵌入式元标记来确定 HTML 文档的编码。如果未指定，将使用默认编码。
事件	配置的事件启动时间。从下拉式框中选择小时和分钟值。 每天 —每天在指定的时间启动指定的事件。 特定日期 —在特定日期启动指定的事件。 1. 星期几 —指定从星期日至星期六的任意一天。 2. 日期 —指定一个月中从 1 日至 31 日间的任意一天（以逗号分隔条目）。例如，4,23,9。
时间	特定月份 —在特定时间和月份启动指定的事件。指定从一月至十二月之间的月份。 每隔此时间段即启动指定的事件。
时间间隔	1. 间隔小时数 —从下拉式框中选择小时数。 2. 间隔秒数 —从下拉式框中选择秒数。

执行搜索

用户主要关心在搜索集合中查询数据，然后获得返回的文档列表。随 Sun Java System Web Server 一起安装的搜索 Web 应用程序提供了默认搜索查询和搜索结果页面。可以直接使用这些页面，也可以按照[自定义搜索页面](#)中的说明使用一组 JSP 标记来自定义这些页面。

用户在服务器管理员创建的集合中进行搜索。用户可以执行以下操作：

- 输入一组关键字和可选的查询运算符进行搜索
- 仅搜索在虚拟服务器中可视的集合
- 在单个集合中进行搜索，或在虚拟服务器中一组可视的集合中进行搜索

服务器管理员必须为用户提供访问虚拟服务器搜索查询页面所需的 URL。

“搜索”页面

最终用户可用来访问搜索功能的默认 URL 为：

`http://<server-instance>:port number/search`

示例：

`http://plaza:8080/search`

最终用户调用此 URL 时，将启动“搜索”页面（一个 Java Web 应用程序）。

注 - 有关执行基本搜索和高级搜索的更多详细说明（包括有关关键字和可选的查询运算符的信息），请参见随搜索引擎提供的联机帮助。要访问这些信息，请单击“搜索”页面上的“帮助”链接。

进行查询

搜索查询页面用于在集合中进行搜索。用户输入一组关键字和可选的查询运算符，结果将显示在其浏览器中的 Web 页面上。结果页面包含指向服务器上符合搜索条件的文档的链接。

注 - 如“自定义搜索页面”中所述，服务器管理员可以自定义此搜索查询页面。

要进行查询，请执行以下步骤：

▼ 进行查询

- 1 通过在浏览器的地址栏中按以下格式输入搜索 Web 应用程序的 URL 来访问该程序：
`http://<server-instance>:port number/search`
- 2 在显示的搜索查询页面的“搜索范围”字段中，选中代表要搜索的集合的复选框。
- 3 键入用于说明查询的文字，然后按 ENTER 键（或单击“搜索”按钮）以获得相关 Web 页面的列表。
您可以使用下一节所述的“高级搜索”页面中提供的搜索参数来设置更精确的搜索。

高级搜索

用户可以通过添加用于微调关键字的运算符来提高搜索的准确性。可以从“高级搜索”页面中选择这些选项。

要进行高级搜索查询，请执行以下步骤：

▼ 进行高级搜索查询

- 1 通过在浏览器的地址栏中按以下格式输入搜索 Web 应用程序的 URL 来访问该程序：
`http://<server-instance>:port number/search`
- 2 单击“高级”链接。
- 3 输入以下信息中的一项或所有项：
 - “搜索范围”。选择要搜索的集合。
 - “查找”。支持四个选项：
 - 所有词。查找包含“查找”字段中指定的所有关键字的页面。
 - 任一词。查找包含“查找”字段中指定的任一关键字的页面。
 - 完全匹配的词组。查找包含与“查找”字段中的短语完全匹配的短语的页面。
 - 段落搜索。在检索到的页面中高亮显示包含关键字或词的段落。
 - 不包含词。搜索将排除包含指定词的 Web 页面。
 - “标题包含/不包含”。将搜索限制在标题中包含指定关键字的页面。
 - "Since"。将搜索操作限制在选定时间段内被编制索引的 Web 页。

文档字段

Sun Java(TM)System Web Server 可维护文档索引。该索引包含对应于每个文档的条目。每个索引条目包含一个或多个诸如 "Title"、"Author" 和 "URL" 之类的字段。可将查询限制在特定的文档字段，以便仅查找符合指定字段中的条件的文档。

例如，如果只搜索 Einstein，您将找到 "Title"、"Author" 或 "Keywords" 字段包含单词 Einstein 的所有文档。这其中包括有关 Einstein 的文档、引用了 Einstein 的文档以及由 Einstein 撰写的文档。但如果您指定了 Author = "Albert Einstein"，将只查找由 Albert Einstein 撰写的文档。

默认情况下，可以搜索的索引字段包括：

1. **Author**—使用 <author> 元标记指定的创建文档的作者或组织。
2. **Keywords**—使用 <keywords> 元标记指定的关键字。
3. **Date**—上次编辑或修改此文档的日期。
4. **Title**—使用 HTML <title> 标记指定的文档标题。

搜索查询运算符

有关搜索查询运算符的详细列表，请参阅[管理控制台搜索联机帮助](#)。

查看搜索结果

搜索结果将显示在用户浏览器中的 Web 页面上，该页面包含指向服务器上符合搜索条件的文档的 HTML 超级链接。默认情况下，每个页面显示 10 条记录（命中次数），这些记录基于相关性按降序排序。每条记录都列出文件名、大小、创建日期等信息。还高亮显示匹配的文字。

自定义搜索页面

Sun Java System Web Server 包含一个提供基本搜索查询和搜索结果页面的默认搜索应用程序。可以直接使用这些 Web 页面，也可以对其进行自定义以满足特定需要。这种自定义可以像使用不同徽标重新设置 Web 页面一样简单，也可以像更改搜索结果的显示顺序一样复杂。

默认搜索应用程序提供了样例 JSP，这些样例 JSP 说明了如何使用搜索标记库来建立自定义的搜索界面。您可以参见位于 /bin/https/webapps/search 的默认搜索应用程序，该样例应用程序说明了如何使用可自定义的搜索标记。

默认搜索界面由四个主要的部分组成：标题、页脚、查询表单和结果。

只需更改标记的属性值即可轻松地自定义这些基本元素。可以使用标记库实现更为详细的自定义。

搜索界面组件

搜索界面包含以下组件：

标题

标题包括徽标、标题和简短说明。

页脚

页脚包含版权信息。

表单

查询表单包含一组代表搜索集合的复选框、一个查询输入框以及“提交”和“帮助”按钮。

结果

默认情况下每页列出 10 条记录。对于每条记录，将显示以下信息：标题、段落、大小、创建日期和 URL。段落是页面的一个小片段，其中包含高亮显示的匹配文字。

自定义搜索查询页面

查询表单包含一个搜索集合的复选框列表、一个查询输入框和“提交”按钮。表单是使用 `<slws:form>` 标记以及具有默认值的 `<collElem>`、`<queryBox>` 和 `<submitButton>` 标记创建的：

```
<slws:form>
  <slws:collElem>
    <slws:queryBox> <slws:submitButton>
  </slws:form>
```

可以将查询表单放在页面的任意位置：中间、边栏等等。也可以以不同格式显示查询表单，例如使用一个横栏，其中集合选择框、查询字符串输入框以及“提交”按钮被水平排列；或显示为一个块，其中集合显示为复选框，查询输入框和“提交”按钮位于下方。

以下示例显示了如何使用 `<searchForm>` 标记组来创建不同格式的查询表单。

水平栏

下面的样例代码将创建一个表单，其中包含一个所有集合的选择框、一个查询输入框和一个“提交”按钮，它们排列在一行中。

```

<slws:form>
  <table cellpadding="0" cellspacing="3" 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>

```

边栏块

您可以创建一个表单块，其中表单元素排列在一个边栏中，表单块的标题为“搜索”，使用的格式与边栏中其他项的格式相同。

表单元素位于边栏中的自定义查询页面

在下面的样例代码中，表单主体包含三个排成一列的复选框，列出了可用的搜索集合。查询输入框和“提交”按钮位于下方：

```

<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>

```

搜索结果的生成过程如下：

- `<formAction>` 标记检索所有表单元素的值并进行基本验证。
- `<formAction>` 标记中出现的 `<search>` 标记、`<resultIteration>` 标记和其他标记可以访问所有表单元素的值。
- `<search>` 标记通过 `<formAction>` 使用查询字符串和集合来执行搜索，并将搜索结果保存在 `pageContext` 中。
- `<resultIteration>` 标记随后对结果集进行检索并重新显示。

以下样例代码以标题栏开始，随后显示一些指定的记录，最后显示导航栏。标题栏包含搜索中使用的查询字符串以及返回的全部记录的范围（例如 1-10）。对于每条记录，记录部分显示了带有指向文件的链接的标题、最多三个具有被高亮显示的关键词的段落、URL、创建日期以及文档大小。

```
<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;&  
          </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>

```

自定义的搜索结果页面

通过操作标记和修改 HTML 可以轻松自定义基本搜索结果界面。例如，可以将导航栏复制并放到搜索结果前面。用户也可以选择显示或不显示搜索记录的任何属性。

除了与表单一起使用外，`<search>`、`<resultIterate>` 和相关标记还可用于列出特定的主题。以下样例代码列出了某个站点上有关 Java Web 服务的头十篇文章。

```

<slws:search collection="Articles" query="Java Web Services" />
<table cellpadding="3" cellspacing="0" border="0">
  <tr class="Title"><td>Java Web Services</td></tr>
</table>
<table cellpadding="3" cellspacing="0" border="0">
  <slws:resultIteration>
    <tr>
      <td><a href="<slws:item property='URL' />"> <slws:item property='Title' /></a></td>
    </tr>
  </slws:resultIteration>
</table>

```

在单独的页面中自定义表单和结果

如果需要将表单页面和结果页面分开，必须使用 `<form>` 标记组创建表单页面，使用 `<formAction>` 标记组创建结果页面。

需要在结果页面中添加指向表单页面的链接以使页面衔接顺畅。

标记惯例

请注意以下标记惯例：

- 标记类属于 `com.sun.web.search.taglibs` 软件包。
- 所有 `pageContext` 属性均具有 `com.sun.web` 前缀。例如，搜索结果的属性为 `com.sun.web.searchresults.form_id`，其中 `form_id` 为表单的名称。

- 可以使用 `s1ws` 前缀来引用标记库。标记的名称及其属性是大小写混合的，其中每个内部单词的首字母大写，例如，`pageContext`。

标记规范

Sun Java System Web Server 包括一组 JSP 标记，这些标记可用于自定义搜索界面中的搜索查询和搜索结果页面。

有关可用来自定义搜索页面的 JSP 标记的完整列表，请参阅《Sun Java System Web Server 7.0 Developer's Guide to Web Applications》。

监视服务器

本节介绍 Sun Java System Web Server 的监视功能，并提供了您可以在实例和配置级别上进行监视的服务器参数的详细列表。

- 第 187 页中的 “Sun Java System Web Server 的监视功能”
- 第 189 页中的 “修改监视参数”
- 第 191 页中的 “配置 SNMP 子代理”
- 第 195 页中的 “设置服务器的日志记录”
- 第 199 页中的 “配置 Administration Server 的日志设置”

Sun Java System Web Server 的监视功能

选择 “监视” 父选项卡下的 “配置” 或 “实例” 选项卡后，将显示可以监视的服务器参数。

从 Sun Java System Web Server 管理控制台，您可以执行以下操作：

- 在实例和配置级别查看服务器统计信息。
- 在配置级别启用/禁用监视。
- 在实例级别查看错误/访问日志。

要在配置级别监视服务器参数，请单击 “监视” > “配置” 选项卡。表中列出了可用配置以及以下信息：

- **节点**—已在其中部署了配置的节点的数量。
- **请求**—所有虚拟服务器收到的请求总数。
- **错误**—所有虚拟服务器记录的错误总数。
- **响应时间**—任何虚拟服务器的最大响应时间。

单击配置名称以获取配置级别的统计信息。常规统计信息分为 3 种类型：

- 请求统计信息
- 错误统计信息

- 响应时间统计信息

通过管理控制台进行监视

在管理控制台中，可以查看以下类别的服务器统计信息：

- 常规统计信息。
- 实例统计信息。
- 虚拟服务器统计信息。

表 13-1 监视类别

类别	说明
常规统计信息	“常规统计信息”显示配置的全部请求、错误和响应统计信息。
实例统计信息	“实例统计信息”显示配置的全部请求、错误和响应统计信息以及有关服务器崩溃和虚拟服务器计数的信息。
虚拟服务器统计信息	“虚拟服务器统计信息”显示配置的全部请求、错误和响应统计信息，以及打开的连接数和已接收/传输的总字节数。

▼ 查看统计信息

- 1 单击“监视”选项卡。
- 2 从配置列表中选择相应的配置。
- 3 查看常规、实例和虚拟服务器统计信息。

注-使用 CLI

可以使用 `get-config-stats`、`get-virtual-serevr-stats`、`get-webapp-stats` 和 `get-servlet-stats` 命令监视服务器。

- `wadm> get-config-stats --user=admin --password-file=admin.passwd --host=localhost --port=8989 --config=test --node=cat.test.com --ssl=true`

以上命令将获取指定实例的统计信息。要在配置级别获取统计信息，可以使用不包含 `--node` 选项的上述命令。

- `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`

以上命令将获取部署了指定配置的所有节点中该配置的聚合虚拟服务器统计信息。要获取指定节点中部署的配置的统计信息，可以使用 `--node` 选项。

- `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`

以上命令将获取指定实例的指定虚拟服务器中部署的指定 Web 应用程序的统计信息。要获取部署了指定配置的所有节点中配置的聚合 Web 应用程序统计信息，可以使用不包含 `--node` 选项的上述命令。

- `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`

以上命令将获取 `servlet servlet-simple` 的统计信息。

修改监视参数

服务器通过 SNMP 执行监视操作。SNMP 是用于交换有关网络活动的数据的协议。利用 SNMP，数据可以在被管理设备和网络管理站 (Network Management Station, NMS) 之间进行传输。被管理设备即运行 SNMP 的任何设备：主机、路由器、Web 服务器和网络上的其他服务器。NMS 是用于远程管理网络的计算机。NMS 软件通常以图形的方式来显示收集的数据，或使用这些数据确保服务器在特定的参数值范围内运行。

NMS 通常是安装有一个或多个网络管理应用程序的功能强大的工作站。网络管理应用程序（例如 HP OpenView）以图形方式显示有关被管理设备（例如 Web 服务器）的信息。例如，它可能显示您企业中的服务器打开或关闭情况，或者收到的错误消息的数量和类型。将 SNMP 与 Sun Java System Web Server 配合使用时，此信息将通过使用两种类型的代理（**子代理**和**主代理**）在 NMS 和服务器之间传输。

子代理收集有关服务器的信息，并将这些信息传送给服务器的主代理。每个 Sun Java System Web Server（Administration Server 除外）都具有子代理。

注 – 对 SNMP 配置进行任何更改后，必须单击“保存”按钮，然后重新启动 SNMP 子代理。

要更改配置的设置，请执行以下任务：

- 1. 单击**配置**选项卡。
- 2. 选择需要更改监视设置的配置。
- 3. 单击**监视设置**子选项卡。

配置监视参数

要更改某个配置的常规监视设置，请在“常规设置”部分下编辑相应的值。下表提供了常规监视参数的字段说明：

表 13-2 字段说明 > 常规监视设置

字段	说明
SNMP 子代理	一般而言，要使用 SNMP，必须在系统上安装并运行一个主代理和至少一个子代理。启用子代理之前，您需要先安装主代理。 选择此选项可以启用/禁用 SNMP 子代理。
时间间隔	轮询时间间隔是显示的统计信息相继两次更新之间的间隔秒数。 如果您的服务器实例正在运行并已启用统计信息功能，您将看到一个显示了选定统计信息类型的页面。该页面每隔 5 至 15 秒更新一次，这取决于您选定的轮询时间间隔。
分析	您可以使用统计信息/分析功能监视服务器的当前活动。统计信息显示了服务器正在处理的请求数目以及这些请求的处理情况。您可以查看各个虚拟服务器的某些统计信息，也可以查看整个服务器实例的其他统计信息。 选择此选项可以启用/禁用分析。

配置 SNMP 子代理参数

要更改某个配置的 SNMP 子代理设置，请在“SNMP 子代理设置”部分下编辑相应的值。下表提供了 SNMP 子代理参数的字段说明：

表 13-3 字段说明 > SNMP 子代理设置

字段	说明
已启用	一般而言，要使用 SNMP，必须在系统上安装并运行一个主代理和至少一个子代理。启用子代理之前，您需要先安装主代理。 选择此选项可以启用/禁用 SNMP 统计信息收集。
主机	输入服务器（仅限 UNIX）的名称和域。
说明	输入服务器的简短说明，其中包括操作系统信息。
组织	输入代表组织的简短名称。
位置	在此字段中输入服务器的位置信息。
联系	在此字段中输入服务器的联系信息。

配置 SNMP 子代理

SNMP 是用于交换有关网络活动的数据的协议。利用 SNMP，数据可以在被管理设备和网络管理站 (Network Management Station, NMS) 之间进行传输。被管理设备即运行 SNMP 的任何设备：主机、路由器、Web 服务器和网络上的其他服务器。NMS 是用于远程管理网络的计算机。NMS 软件通常以图形的方式来显示收集的数据，或使用这些数据确保服务器在特定的参数值范围内运行。

NMS 通常是安装有一个或多个网络管理应用程序的功能强大的工作站。网络管理应用程序（例如 Sun Management Center）以图形方式显示有关被管理设备（例如 Web 服务器）的信息。例如，它可能显示您的企业中服务器的打开或关闭情况，或者收到的错误消息的数量和类型。将 SNMP 与 Sun Java System Web Server 配合使用时，通过使用两种类型的代理（子代理和主代理），这些信息将在 NMS 和服务器之间进行传输。

子代理收集有关服务器的信息，并将这些信息传送给服务器的主代理。

要启动 SNMP 子代理，请执行以下步骤：

1. 单击**节点**选项卡
2. 单击节点列表中的可用节点。
3. 单击**SNMP 子代理**选项卡
4. 单击**启动 SNMP 子代理**按钮以启动子代理。

注 - 启动 SNMP 子代理之前，请验证主代理是否正在运行。仅在主代理运行时才能启动子代理。

要停止 SNMP 子代理，请执行以下步骤：

1. 单击**节点**选项卡

- 2. 单击节点列表中的可用节点。
- 3. 单击 **SNMP 子代理** 选项卡
- 4. 单击**停止 SNMP 子代理**按钮以停止子代理。

一般而言，要使用 SNMP，必须在系统上安装并运行一个主代理和至少一个子代理。启用子代理之前，您需要先安装主代理。

设置 SNMP 的过程因系统而异。下表概述了在不同情况下要执行的过程。稍后将在本章详细介绍实际过程。

开始前，应当验证两件事情：

- 您的系统中是否已运行 SNMP 代理（操作系统的本地代理）？
- 如果是，该本地 SNMP 代理是否支持 SMUX 通信？（如果使用的是 AIX 平台，则您的系统支持 SMUX。）

有关如何验证此信息的说明，请参见您的系统文档。

注 - 在更改了 Administration Server 中的 SNMP 设置、安装了新的服务器或删除了现有服务器后，您必须执行以下步骤：

- (Windows) 重新启动 Windows SNMP 服务或重新引导计算机。
- (UNIX) 使用 Administration Server 重新启动 SNMP 主代理。

表 13-4 一般准则

如果服务器满足以下条件	请执行以下过程
<ul style="list-style-type: none">■ 当前没有运行本地代理	<ul style="list-style-type: none">1. 启动主代理。2. 为系统上安装的每个服务器启用子代理。
<ul style="list-style-type: none">■ 本地代理当前正在运行■ 无 SMUX■ 不需要继续使用本地代理	<ul style="list-style-type: none">1. 为 Administration Server 安装主代理时，请停止本地代理。2. 启动主代理。3. 为系统上安装的每个服务器启用子代理。
<ul style="list-style-type: none">■ 本地代理当前正在运行■ 无 SMUX■ 需要继续使用本地代理	<ul style="list-style-type: none">1. 安装代理 SNMP Agent。2. 启动主代理。3. 启动代理 SNMP Agent。4. 使用主代理端口号以外的其他端口号重新启动本地代理。5. 为系统上安装的每个服务器启用子代理。
<ul style="list-style-type: none">■ 本地代理当前正在运行■ 支持 SMUX	<ul style="list-style-type: none">1. 重新配置 SNMP 本地代理。2. 为系统上安装的每个服务器启用子代理。

使用 CLI 配置 SNMP

▼ 激活 Solaris 上的 SNMP

1 配置 SNMP 参数。

设置配置的 SNMP 参数。

```
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 启动服务器实例。

```
$ ./https-test/bin/startserv
```

4 以超级用户身份运行主代理 (magt)。

注 – 要运行 magt，必须停止本地 snmpd。

```
$ cd /etc/init.d/
$ init.dmi stop; init.snmpdx stop; init.sma stop
```

删除 https-admserv/config/logs/pid.masteragt 文件（如果存在）。

```
$ rm ./https-admserv/config/logs/pid.masteragt
wadm> start-snmp-master-agent --snmp-port 161 hostname
```

5 启动子代理。

删除 https-admserv/config/logs/pid.httptagt 文件（如果存在）。

```
$ rm ./https-admserv/config/logs/pid.httptagt
```

如果 httptagt 已在运行，请将其中止

```
wadm> start-snmp-subagent hostname
```

▼ 激活 Linux 上的 SNMP

1 配置 SNMP 参数。

设置配置的 SNMP 参数。

```
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 启动服务器实例。

```
$ ./https-test/bin/startserv
```

4 以超级用户身份运行本地主代理 (snmpd)。

要与 snmpd 直接通信，请在 /etc/snmp/snmpd.conf 中添加以下行并重新启动 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 启动子代理。

删除 https-admserv/config/logs/pid.httptgt 文件（如果存在）。

```
$ rm ./https-admserv/config/logs/pid.httptgt
```

如果 httptgt 已在运行，请将其中止

```
wadm> start-snmp-subagent hostname
```

▼ 激活 Windows 上的 SNMP

1 配置 SNMP 参数。

设置配置的 SNMP 参数。

```
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 向系统路径环境变量中添加 install-root/lib 目录。

3 重新启动计算机。

- 4 使用“Windows 服务”选项启动 Web Server 实例。
- 5 启动 SNMP 服务。

▼ 配置基于对等点的主代理 (magt)

通过执行以下步骤，可以配置基于对等点的主代理以与 Solaris 10 和 Linux 上的 OS 本地主代理集成。

注 - Solaris 10 OS 本地主代理为 snmpd。默认情况下，此主代理在 SNMP 默认 UDP 端口 161 上运行，可以使用 /etc/sma/snmp/snmpd.conf 文件对其进行配置。该文件提供用于将请求/响应转发到其他主代理或子代理的代理指令。有关更多信息，请参阅 snmpd.conf 手册页。

对于 Solaris 8 和 9，不能与 OS 本地主代理 snmpd 进行完整的集成。对于 linux，httpagt 可以直接与 snmpd 集成。在这种情况下，无需运行 magt。对于 Windows，Sun Java System Web Server snmp 库直接与 Windows SNMP 服务通信。

- 1 可以通过指定在上面的节点中提到的 SNMP 端口 (11161) 启动主代理程序。

- 2 向 Solaris 10 的 /etc/sma/snmp/snmpd.conf 中添加以下内容。

```
proxy -v 1 -c public myserver:11161 .1.3.6.1.4.1.42.2.190.1
```

- 3 重新启动 snmpd。

```
# cd /etc/init.d
# init.dmi stop; init.snmpdx stop; init.sma stop
# init.dmi start; init.snmpdx start; init.sma start
```

- 4 要获取 SNMP 数据，可以对端口使用 snmpwalk：

```
$ snmpwalk -c public -v 1 <host-name>:<port> 1.3.6.1.4.1.42.2.190.1
```

设置服务器的日志记录

Administration Server 日志文件记录了有关服务器的数据，包括遇到的错误类型以及有关服务器访问的信息。这些日志提供了遇到的错误类型以及特定文件被访问的时间等数据，通过查看这些日志可以监视服务器的活动并排除故障。

可以使用管理控制台中的“日志首选项”页面指定管理服务器日志中记录的数据的类型和格式。例如，可以选择记录有关访问管理服务器的每个客户机的数据，也可以忽略某些客户机。此外，您还可以选择通用日志格式，它提供固定数量的服务器信息，也可以创建更符合您要求的自定义日志文件格式。

日志类型

日志类型大致可分为：

1. **访问日志**—访问日志记录了有关对服务器的请求以及服务器的响应的信息。
2. **服务器日志**—服务器日志列出了自创建日志文件以来服务器所遇到的所有错误。它还包含有关服务器的信息消息，如服务器何时启动以及谁试图登录服务器但未成功。

查看访问日志和服务器日志

- 查看服务器日志。

```
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
```

以上命令将显示在日期 01/Jan/2006:09:00:00 与日期 04/Jan/2006:10:00:00 之间指定配置的服务器日志。

- 查看访问日志。

```
wadm> get-access-log --user=admin --password-file=admin.passwd  
--host=localhost --port=8989 --status-code=300 --config=test cat.test.com
```

以上命令将只显示指定配置中状态代码为 300 的那些访问日志条目。

在以上命令中，start-date 和 end-date 选项应采用格式—dd/MM/yyyy:HH:mm:ss。也可以自定义日期格式。可以使用 rcfile 中的 wadm_log_date_format 变量指定您自己的日期格式，而不使用默认日期格式。

配置日志参数

要为某个配置启用和编辑日志设置，请执行以下步骤：

1. 单击**配置**选项卡
2. 选择需要启用/编辑日志设置的配置。
3. 单击**常规设置** > **日志设置**选项卡

编辑访问日志首选项

下表介绍了“访问日志首选项”部分中的字段：

表 13-5 字段说明>编辑访问日志首选项

字段	说明
访问日志	已启用/已禁用。默认情况下，已启用访问日志。选择该选项可以禁用访问日志。请注意，启用访问日志将会稍微降低服务器性能。
文件位置	访问日志文件存储位置的服务器路径。默认值为 <code>../logs/access</code> 。
日志格式	<div><div>1. 使用通用日志格式—此选项是日志文件的默认格式类型。服务器将记录从请求标头中提取的相关性最高的信息。 通用日志格式为 <code>IP address – user [date] “ request ” status content-length</code>。</div><div>2. 仅记录以下详细信息—使用此选项可以仅记录请求标头中的特定值。请从以下值中选择：<ul style="list-style-type: none">客户机主机名系统日期HTTP 状态HTTP 标头HTTP 方法查询字符串虚拟服务器名经过验证的用户名完整的 HTTP 请求内容长度请求 URI协议</div></div>

编辑服务器日志首选项

下表介绍了“服务器日志首选项”部分中的字段：

表 13-6 字段说明>编辑服务器日志首选项

字段	说明
服务器日志位置	存储服务器日志文件的服务器路径。默认值为 <code>../logs/errors</code> 。
日志详尽程度	<div>该选项为您提供了一种设置日志粒度的好方法。对于测试和调试 Web 应用程序，建议将日志级别设置为 <i>finest</i>。</div> <div>对于生产环境，建议将日志级别设置为 <i>failure</i> 或 <i>security</i>。 <i>catastrophe</i> 日志级别记录的信息量最少。</div>
记录虚拟服务器名	如果选择此选项，则在记录错误的同时，还会记录处理请求的虚拟服务器名称。
记录到系统日志中	将所有消息都记录到系统日志中。

表 13-6 字段说明 > 编辑服务器日志首选项 (续)

记录到控制台中	如果选择此选项，则会记录由已部署 Web 应用程序引发的异常（如果写入控制台）。 默认情况下，启用该选项。
日期格式	用于将时间戳附加到错误消息的时间格式。默认值为 [%d/%b/%Y:%H:%M:%S]

将日志文件归档

您可以将日志文件设置为自动归档。在某个特定时间或指定的间隔后，服务器将轮转您的访问日志。服务器会保存旧的日志文件，并用包含了保存日期和时间的名称来标记所保存的文件。

例如，您可以将文件设置为每小时轮转一次，则服务器将保存文件并将其命名为 "access.199907.0152400"，其中“名称|年|月|日|24 小时制时间”连接在一起构成一个字符串。根据所设置的日志轮转类型，访问日志归档文件的实际格式会有所不同。

访问日志轮转在服务器启动时进行初始化。如果启用轮转，服务器将创建一个带有时间戳的访问日志文件并在服务器启动时开始进行轮转。

轮转开始后，当出现需要记录到访问日志文件的请求且该请求的出现发生在之前安排的“下次轮转时间”之后时，服务器将创建一个新的带有时间戳的访问日志文件。

设置日志轮转

您可以使用日志轮转选项为配置的实例创建错误/访问日志轮转的进度表。要设置日志轮转，请执行以下步骤：

- 1. 单击配置选项卡
- 2. 选择需要启用/编辑日志设置的配置。
- 3. 单击常规设置 > 日志设置选项卡
- 4. 单击日志归档部分下的新建按钮

以下部分介绍新建日志轮转页面中的字段：

表 13-7 字段说明 > 设置日志轮转

字段	说明
事件	访问日志轮转/服务器日志轮转。选择这些选项中的任何一个或两个选项可以配置相应日志类型的轮转。

表 13-7 字段说明 > 设置日志轮转 (续)

字段	说明
时间	<p>配置的事件启动时间。从下拉式框中选择小时和分钟值。</p> <p>每天—每天在指定的时间启动指定的事件。</p> <p>特定日期—在特定日期启动指定的事件。</p> <p>1. 星期几—指定从星期日至星期六的任意一天。</p> <p>2. 日期—指定一个月中从 1 日至 31 日间的任意一天（以逗号分隔条目）。例如，4,23,9。</p> <p>特定月份—在特定时间和月份启动指定的事件。指定从一月至十二月之间的月份。</p>
时间间隔	<p>每隔此时间段即启动指定的事件。</p> <p>1. 间隔小时数—从下拉式框中选择小时数。</p> <p>2. 间隔秒数—从下拉式框中选择秒数。</p>

如果您需要删除安排的日志轮转，请在**日志归档**部分中单击**删除**按钮。

配置 Administration Server 的日志设置

管理服务器记录所有使用管理控制台执行的配置更改。记录的某些常见操作包括创建新配置、创建虚拟服务器和配置实例设置。但是，配置级别详细信息（例如，访问 Web 应用程序或访问 Web 应用程序时引发的异常）则由相应配置单独记录。

▼ 修改服务器日志位置

- 1 单击 **Administration Server > 常规选项卡**。
- 2 转至 “**日志首选项**” 部分。
- 3 编辑 “**服务器日志位置**” 字段。

用于存储错误的日志位置。提供用于维护日志文件的有效服务器路径。还将检查管理服务器在 UNIX 系统的指定目录中是否具有写入权限。

默认位置为 `../log/errors`

▼ 修改日志详尽程度

1 单击 **Administration Server** > 常规选项卡。

2 转至日志首选项部分。

3 选择“日志详尽程度”。

该选项为您提供了一种设置日志粒度的好方法。对于测试和调试，建议将日志级别设置为 *finest*。

对于生产环境，建议将日志级别设置为 *failure* 或 *security*。*catastrophe* 日志级别记录的信息量非常少。

▼ 修改日志的日期格式

1 单击 **Administration Server** > 常规选项卡。

2 转至日志首选项部分。

3 编辑“日期格式”字段。

用于将时间戳附加至错误消息的时间格式。默认值为 [%d/%b/%Y:%H:%M:%S]。

国际化和本地化

Sun Java System Web Server 的国际化和本地化版本提供了多语言和多编码支持。

- [第 201 页中的“输入多字节数据”](#)
- [第 202 页中的“支持多字符编码”](#)
- [第 202 页中的“配置服务器以提供本地化内容”](#)

输入多字节数据

如果要在管理控制台页面中输入多字节数据，您需要知道以下问题：

文件名称或目录名称

如果文件名称或目录名称要显示在 URL 中，则不能包含 8 位字符或多字节字符。

LDAP 用户和组

对于电子邮件地址，请仅使用 RFC 17.000(<ftp://ds.internic.net/rfc/rfc17.000.txt>) 中许可的字符。用户 ID 和密码信息必须以 ASCII 编码存储。

为了确保您输入正确格式的用户和组字符，请使用支持 UTF-8 格式的客户机输入 8 位或多字节数据。

支持多字符编码

Sun Java System Web Server 7.0 为以下功能提供了多字符编码支持：

- [第 202 页中的“WebDAV”](#)
- [第 202 页中的“搜索”](#)

WebDAV

Sun Java System Web Server 支持在 PROPPATCH 和 PROPFIND 方法中设置和检索多字节属性。尽管请求可以为任何编码格式，但服务器的响应始终为 UTF-8 格式。

搜索

Sun Java System Web Server 7.0 使用基于 Java 的搜索引擎，支持在基本 Java VM 支持的所有字符编码中对文档进行全文索引和搜索。文档的默认编码可以在创建搜索集合时指定。对于 HTML 文档，索引生成器将尝试从 HTML 元标记来推断编码，如果无法进行推断，则使用默认编码。

搜索界面基于 JSP 标记库，可以自定义搜索界面并使用所需的任何语言和编码对其进行本地化。《Sun Java System Web Server 7.0 Developer's Guide to Web Applications》中列出了这些标记库。

配置服务器以提供本地化内容

最终用户可以将其浏览器配置为发送一个 Accept-language 标头，用以说明正在访问的内容的语言首选项。通过启用 **配置 > (选择配置) > 虚拟服务器 > (选择虚拟服务器) > 服务器设置 > 常规 > 本地化** 下的 **协商客户机语言** 复选框，可以将服务器配置为根据 Accept-language 标头提供内容。

例如，如果启用了此选项，客户发送值为 fr-CH,de 的 Accept-language 标头并请求以下 URL：

```
http://www.someplace.com/somepage.html
```

服务器将按以下顺序搜索文件：

▼ 搜索顺序

- 1 Accept-language 列表 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 不包括国家/地区代码的语言代码 (`fr-CH` 时为 `fr`) :

`http://www.someplace.com/fr/somepage.html`

`http://www.someplace.com/somepage_fr.html`

3 DefaultLanguage , 例如 `magnus.conf` 文件中定义的 `en` 。

`http://www.someplace.com/en/somepage.html`

`http://www.someplace.com/somepage_en.html`

4 如果上述项都未找到 , 服务器将尝试 :

`http://www.someplace.com/somepage.html`

注 - 请记住 : 命名国家/地区代码为 `CH` 和 `TW` 等的本地化的文件时 , 名称将转换为小写字母并且短划线 (`-`) 将转换为下划线 (`_`) 。



注意 - 启用 `acceptlanguage` 设置会降低服务器的性能 , 因为服务器必须根据上述算法检查 `Accept-language` 中指定的每种语言的内容。

对前一版本中的 CLI 的更改

下表描述了使用 Sun Java System Web Server 7.0 和前一版本可以执行的一些日常任务。

表 A-1 对前一版本中的 CLI 的更改

任务	6.1 CLI	7.0 CLI
列出实例所有已部署的 Web 应用程序。	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 应用程序。	wdeploy deploy -i INSTANCE_NAME -v VIRTUAL_SERVER -u URI_PATH war file name	<div>1. wadm> add-webapp --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME --vs=VIRTUAL_SERVER --uri=URI_PATH war file name</div> <div>2. wadm> deploy-config --user=admin --port=8888 --password-file=admin.passwd 'HOSTNAME'</div>
重新配置正在运行的实例。	不支持。	wadm> reconfig-instance --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME
列出实例的所有虚拟服务器。	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 对前一版本中的 CLI 的更改 (续)

任务	6.1 CLI	7.0 CLI
列出所有 JDBC 资源。	HttpServerAdmin list -r -jdbc -d INSTALL_DIR -sintance https-INSTANCE_NAME	wadm> list-jdbc-resources --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME
创建自定义资源。	HttpServerAdmin create -r -custom -jndiname -poolname -enabled true	wadm> create-custom-resource --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME --res-type=type --jndi-name NAME
启动实例。	不支持。	wadm> start-instance --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME NODENAME*
停止实例。	不支持。	wadm> stop-instance --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME NODENAME*
使用反向代理配置 Web 服务器。	不支持。	1. wadm> create-reverse-proxy --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME --vs='VIRTUAL_SERVER' --from='URI' --server='target-hostname' 2. 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
禁用反向代理。	不支持。	wadm> delete-reverse-proxy --user=admin --port=8888 --password-file=admin.passwd --config=HOSTNAME --vs='VIRTUAL_SERVER' --uri-prefix='URI'

表 A-1 对前一版本中的 CLI 的更改 (续)

任务	6.1 CLI	7.0 CLI
启用 WebDAV。	不支持。	<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>
创建新的 Web 服务器。	不支持。	<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 插件

- 第 209 页中的 “简介”
- 第 210 页中的 “插件函数 (SAF)”
- 第 213 页中的 “在 Web Server 上配置 FastCGI 插件”
- 第 219 页中的 “FastCGI 应用程序样例”

简介

FastCGI 是对现有 CGI（Common Gateway Interface，公共网关接口）的增强，CGI 是一种连接外部应用程序与 Web Server 的标准。与 CGI 一样，FastCGI 应用程序在单独、隔离的进程中运行。以下是使用 FastCGI 的一些优点：

- 使应用程序在客户机请求之间持久化、消除应用程序启动开销以及允许应用程序在客户机调用之间保持状态。
- 使应用程序驻留在远程系统（不同于正在运行 Web Server 的系统）上。
- 使应用程序功能更加灵活，并显式支持进行客户机验证和输入过滤的应用程序。
- 允许管理员限制 FastCGI 服务器对系统所产生的影响。

通过使用 FastCGI 插件，Web Server 能够以可伸缩的方式安全地使用流行的第三方动态内容生成技术（如 Perl 和 Python）。

有关 FastCGI 的更多信息，请参阅 <http://www.fastcgi.com/devkit/doc/fcgi-spec.html> 中的规范。

插件函数 (SAF)

FastCGI 插件提供以下服务器应用程序函数 (Server Application Function, SAF) :

以下各节介绍了 FastCGI SAF 的各种参数和 “错误原因” 字符串 :

- 第 210 页中的 “auth-fastcgi”
- 第 210 页中的 “responder-fastcgi”
- 第 211 页中的 “filter-fastcgi”
- 第 211 页中的 “error-fastcgi”
- 第 211 页中的 “FastCGI SAF 参数”
- 第 213 页中的 “error-fastcgi SAF 错误原因字符串”

auth-fastcgi

auth-fastcgi 是修补程序检查函数。此函数用于将请求转发到 “授权者” FastCGI 应用程序。如果授权成功，则发送返回码 200。否则，将 “授权者” FastCGI 应用程序的响应发送回至用户代理。

可以在 <http://www.fastcgi.com/devkit/doc/fcgi-spec.html#S6> 中找到有关 FastCGI 角色的更多信息。

以下位置提供了 auth-fastcgi SAF 接受的参数：[第 211 页中的 “FastCGI SAF 参数”](#)。

以下 obj.conf 代码示例说明了 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 是服务函数。此函数用于将请求转发到充当 “响应者” 的 FastCGI 应用程序。“响应者” 应用程序的响应将被发送到用户代理。

<http://www.fastcgi.com/devkit/doc/fcgi-spec.html#S6> 中提供了有关 FastCGI 角色的更多信息。

以下位置提供了 responder-fastcgi SAF 接受的参数列表：[第 211 页中的 “FastCGI SAF 参数”](#)。

以下 obj.conf 代码示例说明了 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` 是服务函数。此函数用于将请求转发到“过滤器”类型的 FastCGI 应用程序。“过滤器”应用程序接收与 HTTP 请求关联的信息，还接收存储在服务器上的文件中的数据。然后，“过滤器”应用程序生成“已过滤”版本的数据流作为响应。该响应将被发送到用户代理。<http://www.fastcgi.com/devkit/doc/fcgi-spec.html#S6> 中提供了有关 FastCGI 角色的更多信息。

以下位置提供了 `filter-fastcgi` SAF 接受的参数列表：[第 211 页中的“FastCGI SAF 参数”](#)。

以下 `obj.conf` 代码示例说明了 `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` 是错误函数。`error-fastcgi` SAF 处理特定于 FastCGI 插件的错误。但是，此函数不处理 HTTP 错误。发生错误时，可以将 FastCGI 插件配置为显示特定的页面或将请求重定向至特定的 URL。

以下位置提供了 `error-fastcgi` SAF 接受的参数列表：[第 211 页中的“FastCGI SAF 参数”](#)。

以下 `obj.conf` 代码片段说明了 `error-fastcgi` 的用法：

```
Error fn="error-fastcgi" error-reason="Invalid Parameters"
error-url="http://www.foo.com/errorPage.html"
```

有关 `error-fastcgi` 参数的信息，请参见[第 211 页中的“FastCGI SAF 参数”](#)。

FastCGI SAF 参数

FastCGI 插件 SAF `"auth-fastcgi"`、`"responder-fastcgi"` 和 `"filter-fastcgi"` 全都接受下列参数，除非另外显式说明：

请注意，参数 `chroot`、`user`、`group` 和 `nice` 只适用于 UNIX 平台。在 Windows 平台上，这些参数都将被忽略。

- `app-path` —（可选）处理请求的 FastCGI 应用程序路径。该功能依赖于 `bind-path` 参数的值，如下所示：
 1. 如果仅指定了 `app-path`，插件将创建 FastCGI 应用程序，该应用程序侦听由插件创建的 UNIX 域套接字。然而，仅 UNIX 平台接受此参数。在 Windows 上将记录错误消息。

2. 如果同时指定了 `app-path` 和 `bind-path` 参数，插件将启动指定的 FastCGI 应用程序进程并将它们绑定到指定的 `bind-path`。
 3. 如果仅指定了 `bind-path`，则 FastCGI 应用程序将被认为正在远程运行。因此，插件将不启动 FastCGI 应用程序进程。
 4. 如果既未指定 `"app-path"`，也未指定 `"bind-path"`，则插件将记录错误消息。
- `app-args` —（可选）作为参数传递至 FastCGI 应用程序进程的值。允许使用多个 `app-args` 参数。多个 `app-args` 参数的格式为 `app-args="value" app-args="value" ...`。
 - `bind-path` —（可选）可以是 Unix 域套接字名称或者是格式 `"host:port"`。`"app-path"` 参数的说明解释了 `"bind-path"` 参数的用法。请注意，Unix 域套接字名称仅适用于 UNIX 平台。在 Windows 平台上，必须将 `bind-path` 指定为 `"host:port"`。
 - `min-procs` —（可选）一个整数，用来指定要创建的 FastCGI 应用程序进程的最小数目。默认值为 1。
 - `max-procs` —（可选）一个整数，用来指定随时可以创建的 FastCGI 应用程序进程的最大数目。该整数值必须等于或大于 `min-procs`。默认值为 1。
 - `chroot` —（可选）用于设置 chroot FastCGI 服务器应用程序进程的根目录。默认值为 Web Server 的根目录。
 - `user` —（可选）指定用来运行 FastCGI 应用程序的用户 ID。默认值为 Web Server 的用户 ID。
 - `group` —（可选）FastCGI 应用程序将在指定的组下运行。默认值为 Web Server 的组。
 - `nice` —（可选）指定 FastCGI 应用程序进程的 `nice/priority` 值。
 - `listen-queue` —（可选）一个整数，用来指定套接字的侦听队列大小。此参数的默认值为 256。
 - `app-env` —（可选）作为环境变量传递到 FastCGI 应用程序进程的值对。允许使用多个 `"app-env"` 参数。多个 `app-env` 参数的格式为 `app-env="name=value" app-env="name=value" ...`。
 - `reuse-connection` —（可选）一个布尔值，用来确定是否重复使用 FastCGI 应用程序连接。`False`（0、`false` 或 `no`）表示在每个请求后将关闭到 FastCGI 应用程序的连接。`True`（1、`true` 或 `yes`）表示对新请求重复使用现有连接。默认值为 `false`。另请参见 `connection-timeout`。
 - `connection-timeout` —（可选）如果 `"reuse-connection"` 被设置为 `True`，则此值指定池化连接的超时值（以秒为单位）。如果连接在指定的时间段内处于空闲状态，则插件将关闭该连接。此参数的默认值为 5 秒。另请参见 `reuse-connection`。
 - `resp-timeout` —（可选）一个整数，用来表示 FastCGI 服务器响应超时（以秒为单位）。如果在指定时间段内 FastCGI 应用程序未响应，将放弃请求。此参数的默认值为 5 分钟。

- `restart-interval`—（可选）一个整数，用来表示两次重新启动 FastCGI 应用程序之间的时间间隔（以分钟为单位）。此参数的默认值为 60 分钟（1 小时）。如果将此参数的值设置为零，则不会强制重新启动 FastCGI 应用程序。
- `req-retry`—（可选）一个整数，用来表示在 FastCGI 应用程序拒绝请求时插件应重新发送该请求的次数。此参数的默认值为零。

`error-fastcgi` 服务器应用程序函数 (Server Application Function, SAF) 接受以下参数：

- `error-url`—指定在出现故障或错误时显示的页面、URI 或 URL。此参数的值可以为绝对路径、相对于文档根目录的路径、URL 或 URI。
- `error-reason`—（可选）表示 FastCGI 协议错误的字符串。此字符串用来在发生任何插件错误时区分要显示的错误 URL。

error-fastcgi SAF 错误原因字符串

本节提供了所有有效“错误原因”字符串及其说明：

- “缺少配置参数或配置参数无效”：每当未指定 `app-path` 和 `bind-path` 时显示此字符串。
- “存根启动错误”：启动 `Fastcgisub` 进程失败。
- “存根连接失败”：无法连接至 `Fastcgistub`。
- “没有权限”：FastCGI 应用程序或 `Fastcgisub` 没有执行权限。
- “存根请求处理错误”：无法将请求发送至存根、存根针对请求发出的响应无效或未响应等。
- “设置参数失败”：设置用户、组、`Chroot`、`Nice` 等失败。
- “用户和/或组无效”：用户或组无效时显示此字符串。
- “创建服务器进程失败”：FastCGI 应用程序执行失败或 FastCGI 应用程序无法绑定到指定的地址。
- “Fastcgi 协议错误”：FastCGI 应用程序包含具有无效 FastCGI 版本或角色的标头。
- “内部错误”：无法打开要发送至过滤器应用程序的文件或任何其他未知错误。

在 Web Server 上配置 FastCGI 插件

FastCGI 插件与 Web Server 7.0 捆绑在一起。该插件安装在以下位置：

32 位 FastCGI 插件二进制文件安装在 `<install_dir>/plugins/fastcgi` 目录中。

64 位 Solaris SPARC FastCGI 插件二进制文件安装在
`<install_dir>/lib/plugins/fastcgi/64` 目录中。

将安装以下 FastCGI 二进制文件：

libfastcgi.so (适用于 Solaris/Linux)
fastcgi.dll (适用于 Windows)
Fastcgistub.exe (适用于 Windows)
libfastcgi.sl (适用于 HP-UX)
Fastcgistub (可执行文件)

可以使用位于 <instance-dir>/config 目录中的 Web Server 配置文件来配置 FastCGI 插件。要配置 FastCGI 插件，请执行以下步骤：

- 第 214 页中的 “修改 magnus.conf”
- 第 214 页中的 “修改 MIME 类型（可选）”
- 第 215 页中的 “修改 obj.conf”
- 第 216 页中的 “排除 FastCGI 插件的故障”
- 第 217 页中的 “开发 FastCGI 应用程序”

修改 magnus.conf

使用 "load-modules" 初始化函数加载 FastCGI 插件共享库。

```
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 类型（可选）

编辑 mime.types 文件以指定 MIME 映射。修改 MIME 类型映射为可选步骤。

例如，

```
--Sun Microsystems Inc. MIME Information  
  
# Do not delete the above line. It is used to identify the file type.  
  
#  
  
# Copyright 2006 Sun Microsystems, Inc. All rights reserved.  
  
# Use is subject to license terms.  
  
#
```

```

type=application/octet-stream exts=bin

type=application/astound exts=asd,asn

...

...

type=magnus-internal/fastcgi exts=php

...

...

```

修改 obj.conf

编辑 obj.conf 文件以使用前面几节中介绍的插件 SAF 配置特定于 FastCGI 的请求。

以下是修改的 obj.conf 文件示例：

```

#

# Copyright 2006 Sun Microsystems, Inc. All rights reserved.

# Use is subject to license terms.

#

# You can edit this file, but comments and formatting changes
# might be lost when you use the administration GUI or CLI.


<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>
...
```

请注意，通过为不同 URL 模式定义不同对象或将 SAF 映射至不同 MIME 类型，可以采用不同方法调用 FastCGI SAF。

有关 obj.conf 配置和语法的更多信息，请参阅[管理配置文件参考指南](#)。

排除 FastCGI 插件的故障

Fastcgistub 是一个管理 FastCGI 应用程序进程生命周期的进程管理器。Fastcgistub 将消息记录在 Web Server 临时目录下的 Fastcgistub.log 文件中。如果发生任何错误，检查此文件可以帮助调试问题。

问题：未满足 FastCGI 请求

可能的原因和解决方法如下：

1. 检查是否加载了 FastCGI 插件。如果在 Web Server 启动过程中显示以下消息，则表示已成功加载该插件。否则，请在 magnus.conf 中检查插件库的路径：**FCGI1000:**
Sun Java System Web Server 7.0 FastCGI NSAPI Plugin < *build info*>
2. 检查是否在 obj.conf 中正确指定了请求映射。有关 obj.conf 文件的更多信息，请参见 Sun Java System Web Server [管理员配置参考文件](#)。
3. 检查错误日志以找出任何可能的错误消息。
4. 检查存根二进制文件和 FastCGI 应用程序的权限。如果未授予插件足够的权限，则插件将无法启动存根或应用程序。
5. 检查 Fastcgistub.log 文件以找出存根端任何可能的错误。
6. 如果可能的话，以单机模式运行 FastCGI 应用程序并检查它在运行时是否出现任何问题。

如果抛出任何与库相关的错误，请在 obj.conf 中将 LD_LIBRARY_PATH 指定为具有 LD_LIBRARY_PATH=<*dependency library paths*> 值的 app-env 参数。

问题：FastCGI 应用程序无法启动。

可能的原因和解决方法如下：

检查 `Fastcgistub.log` 文件以找出以下日志消息：

```
..
<pid> process startup failure, trying to restart
...
Even after trying <n> time(s), <application path> process failed to start...no more retries
```

启动失败的原因之一可能是加载相关库失败。可以通过将相应的库路径指定为 `obj.conf` 文件中配置的 FastCGI 应用程序的 `app-env` 参数值来解决此问题。例如：

```
Service fn="responder_fastcgi" app-path="/fastcgi/c/tux-app" bind-path="localhost:2112"
app-env="LD_LIBRARY_PATH=/tuxedo/lib"
```

开发 FastCGI 应用程序

可以使用 Perl、PHP、C 和 Java 开发 FastCGI 应用程序。以下各节简要介绍了使用几种常见编程语言开发此类应用程序的过程。

- [第 217 页中的“执行 FastCGI 应用程序”](#)
- [第 217 页中的“FastCGI 应用程序的结构”](#)
- [第 218 页中的“使用 Perl”](#)
- [第 218 页中的“使用 PHP”](#)
- [第 218 页中的“使用 C/Java”](#)

▼ 执行 FastCGI 应用程序

- 1 停止 Web Server。
- 2 重新启动 Web Server。
- 3 访问应用程序根目录为 `"fcgi"` 的应用程序。
例如：`http://localhost/fcgi/ListDir.php`

FastCGI 应用程序的结构

典型的 FastCGI 应用程序具有以下代码结构：

```
Initialization code

Start of response loop
    body of response loop
End of response loop
```

初始化代码只在应用程序初始化时运行一次。初始化代码执行的操作通常比较耗时，例如打开数据库或计算表或位图的值。将 CGI 程序转换为 FastCGI 程序的主要任务是将初始化代码与需要针对每个请求运行的代码分开。

响应循环连续运行，等待客户机请求到达。该循环以对 `FCGI_Accept`（FastCGI 库中的一个例程）的调用开始。`FCGI_Accept` 例程将阻止程序执行，直到客户机请求 FastCGI 应用程序为止。客户机请求到达后，`FCGI_Accept` 将解除阻止并运行一次响应循环主体，然后重新阻止，等待另一个客户机请求。只有在系统管理员或 Web Server 中止 FastCGI 应用程序时，该循环才会终止。

使用 Perl

从 CPAN 下载并安装最新的 FCGI 模块。对于 ActivePerl，可以从 <http://aspn.activestate.com/ASPN/Downloads/ActivePerl/PPM/Zips> 下载模块。

有关使用 Perl 编写 FastCGI 应用程序的更多信息，请访问 <http://www.fastcgi.com/devkit/fastcgi-prog-guide/ch3perl.htm#3659>。

使用 PHP

从 PHP 4.3.0 开始，FastCGI 成为 PHP 引擎支持的配置。要编译支持 FastCGI 的 PHP 4.3.x 或更高版本的引擎，请在构建进程中包括配置开关 `--enable-fastcgi`，例如：

```
./configure <other-options> --enable-fastcgi  
gmake
```

编译完成后，php 二进制文件将启用 FastCGI。

使用 PHP 版本 5.1.2 或早期版本（包括 PHP 4.x）时，应配置 FastCGI 插件且 `bind-path` 的格式为“主机:端口”。例如，`bind-path = "localhost:3333"`。

对于 PHP 版本 5.1.3 和更高版本，`bind-path` 是可选的。如果指定了此项，其格式应为“主机:端口”。它可以是字符串。例如，`bind-path = "myphpbindpath"`。

使用 C/Java

FastCGI 开发工具包提供了用于编写 FastCGI C/Java 应用程序的 API。您可以从 <http://www.fastcgi.com/devkit/doc/fcgi-devel-kit.htm> 下载该工具包。

要构建下载的 FastCGI 开发工具包，请执行以下步骤：

1. 解压缩 tar 文件。此操作将创建一个名为 `fcgi-devel-kit` 的新目录。
2. 在 `fcgi-devel-kit` 目录中执行以下命令序列：
 - a. `./configure`
 - b. `make`

有关使用 C 编写 FastCGI 应用程序的更多信息，请访问 <http://www.fastcgi.com/devkit/doc/fcgi-devel-kit.htm#S3>

有关使用 Java 编写 FastCGI 应用程序的更多信息，请访问 <http://www.fastcgi.com/devkit/doc/fcgi-java.htm>

FastCGI 应用程序样例

本节包含使用 PHP、Perl 和 C 编写的 FastCGI 应用程序样例。

- 第 219 页中的 “使用 PHP 编写的响应者应用程序 (ListDir.php)”
- 第 219 页中的 “使用 Perl 编写的授权者应用程序 (SimpleAuth.pl)”
- 第 220 页中的 “使用 C 编写的过滤器应用程序 (SimpleFilter.c)”

使用 PHP 编写的响应者应用程序 (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);
        }
    }
?>
```

以上示例的 obj.conf 代码片段为：

```
<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 编写的授权者应用程序 (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:

以上示例的 obj.conf 代码片段为：

```

<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>

```

第一次请求 <http://localhost/fcgi/php/ListDir.php> 时，浏览器将显示验证对话框。在用户输入用户名和密码后，将列出 "/tmp" 目录的内容。

使用 C 编写的过滤器应用程序 (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);
}

```

以上示例的 `obj.conf` 设置示例。

如果该 FastCGI 应用程序是在运行 Web Server 的计算机上运行，则

```

<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>

```

如果该应用程序正在远程计算机上运行，则必须在 `obj.conf` 文件中包含以下代码行：

```

<Object name="filter.fcgi">
    Service fn="filter-fastcgi" bind-path="<remote-host>:<remote-port>"
</Object>

```

如果要过滤 Web Server 实例根目录下的 fcgi 目录中大小为 "26868" 字节的 "FilterThisFile" 文件，则对 "http://localhost/fcgi/filter/FilterThisFile" 的请求将生成以下输出：

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 服务

无需额外配置即可在 Sun Java System Web Server 7.0 上运行 Web 服务。服务器上集成了 JWSDP，因此在部署为 Web 应用程序时，所有 JWSDP Web 应用程序都应该运行。

有关部署 Web 应用程序的更多信息，请参见第 151 页中的“添加 Web 应用程序”。

在 Web Server 7.0 上运行 JWSDP 2.0 样例

在部署至 Web Server 7.0 之前，需要修改 JWSDP 2.0 中 Web 应用程序样例的配置文件。具体地说，需要编辑 `jaxws` 样例中的配置文件以便可以在 Web Server 7.0 上部署。步骤如下：

▼ 运行 JWSDP 2.0 样例

- 1 下载 JWSDP 2.0。
- 2 在 `$JWSDP_HOME/jwsdp-shared/bin` 中创建特定于 Web Server 的 `sjsws.props`。
下面提供了样例 `sjws.props`。所有字段都是强制字段。

```
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
```

注-admin.passwd 文件包含管理员的服务器密码。此条目的示例为
: wadm_password=adminadmin

3 修改配置文件。

修改打算运行的样例的 build.xml 和 etc/deploy-targets.xml 文件。请注意，deploy-targets.xml 中需要的更改不是特定于样例的。您应该可以使用主副本并将它复制到打算运行的应用程序的 etc. 目录中。

build.xml 更改。

注释掉 build.xml 顶部的 Application Server lib.home 定义并添加 Web Server lib 位置。更改的 build.xml 代码片段如下所示：

```
<!--
**
** Comment out the Application Server lib.home declaration **
**
<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>
-->
<!--
** Add the Web Server library location **
-->
<property name="lib.home" value="${WS_HOME}/lib" />
```

deploy-targets.xml 更改。

将 etc/deploy-targets.xml 替换为特定于 Web 服务器的 deploy-targets.xml。此更改会将 Web 应用程序部署至 Web Server。deploy-targets.xml 文件的代码片段如下所示：

```
<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>
```


词汇表

Access Control Entries, ACE （访问控制条目）	Web 服务器用来评估传入访问请求的分层结构规则体系。
Access Control List, ACL （访问控制列表）	ACE 的集合。ACL 是一种用于定义哪些用户有权访问您的服务器的机制。您可以定义专门针对某特定文件或目录的 ACL 规则，允许或拒绝一个或多个用户和组的访问。
Administration Server （管理服务器）	基于 Web 的服务器，包含用来配置所有 Sun Java System Web Server 的表单。
admpw	Enterprise Administrator Server 超级用户的用户名和密码文件。
agent （代理）	在网络设备（如路由器、主机或 X 终端）中运行网络管理软件的软件。请参见智能代理。
authentication （验证）	<p>允许词汇表向服务器验证他们的身份。基本验证或默认验证要求用户输入用户名和密码来访问 Web 服务器或 Web 站点，它要求 LDAP 数据库中的用户和组的列表。请参见摘要验证和 SSL 验证。</p> <p>允许访问整个服务器或服务器上的特定文件及目录。可以利用条件（包括主机名和 IP 地址）来限制验证。</p>
cache （高速缓存）	本地存储的原始数据的副本。高速缓存的数据被请求时，不必再从远程服务器进行检索。
Certificate revocation list, CRL （证书撤销列表）	由 CA 提供的所有被撤回证书的 CA 列表。
certificate （证书）	由通信双方信任的第三方颁发的不得转让、伪造的数字文件。
certification authority, CA （证书授权机构）	颁发用于加密事务的数字文件的内部或第三方组织。
CGI	Common Gateway Interface（公共网关接口）。外部程序与 HTTP 服务器进行通信所用的接口。为使用 CGI 而编写的程序称为 CGI 程序或 CGI 脚本。CGI 程序用于处理表单或解析输出，服务器通常不会处理或解析这些表单或输出。
chroot	附加根目录，创建它以将服务器限制在特定目录内。可以使用此功能来保护未受保护的服务器。
ciphertext （密文）	通过加密隐藏的信息，只有预期的接收者可以解密。
cipher （加密算法）	加密算法是用于加密或解密的密码学算法（数学函数）。

client auth (客户机验证)	客户机验证。
client (客户机)	用于请求或查看万维网资料的软件 (如 Netscape Navigator) 。
cluster (群集)	添加到 “主” Administration Server 并由 “主” Administration Server 控制的一组远程 “从属” Administration Server。群集中所有服务器的平台必须相同, 并且必须具有相同的用户 ID 和密码。
collection (集合)	包含有关文档信息 (如字表和文件属性) 的数据库。搜索功能使用集合来检索与指定的搜索条件相匹配的文档。
Common LogFile Format (通用日志文件格式)	服务器用于向访问日志中输入信息的格式。在所有主要服务器 (包括 Sun Java System Web Server) 中, 格式都是相同的。
Compromised key list, CKL (已泄密密钥列表)	具有已泄密密钥的用户的密钥信息列表。此列表也由 CA 提供。
daemon (守护进程) (UNIX)	负责特定系统任务的后台进程。
DHCP	Dynamic Host Configuration Protocol (动态主机配置协议)。Internet 推荐的标准协议, 允许系统动态地将 IP 指定给网络中的单个计算机。
digest authentication (摘要验证)	允许用户进行验证而不必明文发送用户名和密码。浏览器使用 MD5 算法创建摘要值。服务器使用摘要验证插件来比较客户机提供的摘要值。
DNS	Domain Name System (域名系统)。网络中的计算机用来将标准 IP 地址 (如 198.93.93.10) 与主机名 (如 www.sun.com) 相关联的系统。计算机通常从 DNS 服务器获取此翻译信息, 或从其系统维护的表中查找该信息。
DNS alias (DNS 别名)	DNS 服务器可识别的指向其他主机 (特别是 DNS CNAME 记录) 的主机名。计算机始终具有一个真实的名称, 但可以有一个或多个别名。例如, 一个别名 (如 www.yourdomain.domain) 可能指向一台名为 realthing.yourdomain.domain 的真实的计算机 (服务器当前所在的计算机)。
document root (文档根目录)	服务器计算机上的目录, 包含要提供给访问该服务器的用户的文件、图像和数据。
drop word (丢弃词)	请参见停用词。
encryption (加密)	转换信息的过程, 使得只有预期的接收者能够解密和读取信息。
expires header (到期标头)	由远程服务器指定的返回文档的到期时间。
extranet (外部网)	公司内部网向 Internet 扩展, 从而使客户、供应商和远程工作人员能够访问数据。
fancy indexing (复杂索引)	它是一种能够比简单索引提供更多信息的索引方法。复杂索引按照文件名显示内容列表, 列表包括文件大小、上次修改日期和反映文件类型的图标。因此, 复杂索引在载入客户机时可能比简单索引花费更长时间。

file extension (文件扩展名)	通常用于定义文件类型的文件名的最后一部分。例如，在 <code>index.html</code> 文件名中，文件扩展名为 <code>html</code> 。
file type (文件类型)	给定文件的格式。例如，图形文件与文本文件具有不同的文件类型。文件类型通常由文件扩展名 (<code>.gif</code> 或 <code>.html</code>) 标识。
firewall (防火墙)	一种网络配置，通常既是指硬件也是指软件，用于保护组织内部的网络计算机不被外部访问。防火墙一般用来保护信息（如实际建筑物或组织办公场所内的网络电子邮件及数据文件）。
flexible log format (灵活日志格式)	服务器用来将信息输入到访问日志中的格式。
FORTEZZA	美国政府机构用来管理敏感但未分类的信息的加密系统。
FTP	File Transfer Protocol（文件传输协议）。一种 Internet 协议，用于将文件通过网络从一台计算机传输到另一台计算机。
GIF	Graphics Interchange Format（图形交换格式）。由 CompuServe 最早创建的一种跨平台的图像格式。GIF 文件的大小通常比其他图形文件类型（BMP 或 TIFF）小很多。GIF 是最常用的交换格式之一。在 UNIX、Microsoft Windows 和 Apple Macintosh 系统上都可以很容易地查看 GIF 图像。
hard restart (硬重启)	进程或服务被终止然后重新启动。请参见软重启。
home page (主页)	服务器上存在的文档，作为服务器内容的目录或入口点。服务器的配置文件中定义了该文档的位置。
hostname (主机名)	<i>machine.domain.dom</i> 格式的计算机的名称，将转换为 IP 地址。例如， <code>www.sun.com</code> 是位于域 <code>com</code> 和子域 <code>sun</code> 中的 <code>www</code> 计算机。
HTML	Hypertext Markup Language（超文本标记语言）。万维网上的文档使用的格式化语言。HTML 文件是带有格式化代码的纯文本文件，告知浏览器（如 Netscape Navigator）如何显示文本、图形位置和表单项，以及如何显示指向其他页面的链接。
HTTP	HyperText Transfer Protocol（超文本传输协议）。用于在 HTTP 服务器与客户机之间交换信息的方法。
HTTP-NG	下一代超文本传输协议。
HTTPD	HTTP 守护进程或服务的缩写，是使用 HTTP 协议提供信息的程序。
HTTPS	安全版本的 HTTP，使用安全套接字层 (Secure Sockets Layer, SSL) 实现。
imagemap	使图像的各个区域处于活动状态的进程，从而使用户可以通过用鼠标单击图像的不同区域来进行浏览并获取信息。Imagemap 还可以引用一个名为 "imagemap" 的 CGI 程序，该程序用来处理其他 HTTPD 实现中的 <code>imagemap</code> 功能。
inittab (UNIX)	一种 UNIX 文件，用于列出因各种原因而停止并需要重新启动的程序。该文件可确保程序连续运行。由于其位置的原因，该文件也被称为 <code>/etc/inittab</code> 。该文件并非在所有 UNIX 系统上都可用。

intelligent agent (智能代理)	服务器中的一个对象，代表用户执行各种请求（如 HTTP、NNTP、SMTP 和 FTP 请求）。在某种意义上，智能代理充当服务器的客户机，提出请求让服务器来完成。
IP address (IP 地址)	Internet Protocol address (Internet 协议地址)。由点分隔的一组数字，指定计算机在 Internet 上的实际位置（如 198.93.93.10）。
ISDN	Integrated Services Digital Network (综合业务数字网)。
ISINDEX	一种 HTML 标记，用于启用客户机中的搜索。文档可以使用网络浏览器的功能来接受搜索字符串并将其发送到服务器，以访问可搜索的索引而不必使用表单。为了使用 <ISINDEX>，必须创建一个查询处理程序。
ISMAP	ISMAP 是 HTML 文档中使用的 IMG SRC 标记的扩展，用来告知服务器命名图像是一个 imagemap。
ISP	Internet Service Provider (Internet 服务提供商)。提供 Internet 连接的组织。
Java	一种由 Sun Microsystems 创建的面向对象的编程语言，用来创建称为小应用程序的实时、交互式程序。
Java Servlet	用于启用所有 Java Servlet 元功能的扩展，包括实例化、初始化、破坏、从其他组件访问及配置管理。Java Servlet 是可重复使用的 Java 应用程序，运行在 Web 服务器而不是 Web 浏览器中。
JavaScript	一种精简的、基于对象的脚本语言，用来开发客户机和服务器 Internet 应用程序。
JavaServer Pages	用于启用所有 JavaServer Pages (JSP) 元功能的扩展，包括实例化、初始化、破坏、从其他组件访问及配置管理。JSP 是可重复使用的 Java 应用程序，运行在 Web 服务器而不是 Web 浏览器中。
last-modified header (上次修改标头)	在服务器的 HTTP 响应中返回的文档文件的上一次修改时间。
LDAP database (LDAP 数据库)	存储了用于验证的用户和组列表的数据库。
listen socket (侦听套接字)	端口号与 IP 地址的组合。服务器与客户机之间的连接在侦听套接字上进行。
magnus.conf	主 Web 服务器配置文件。该文件包含全局服务器配置信息（如端口、安全性等）。它为用于在初始化过程中配置服务器的变量设置了值。企业服务器将读取该文件并在启动时执行这些变量设置。服务器只有在重新启动时才会再次读取该文件，因此每次对该文件进行更改后都必须重新启动服务器。
MD5	RSA Data Security 公司推出的消息摘要算法。MD5 可以用来生成短小的数据摘要，并且该摘要为唯一的概率很高。从算术角度讲，要想用它来生成一段可以生成相同消息摘要电子邮件的数据几乎是不可能的。
MD5 signature (MD5 签名)	由 MD5 算法生成的消息摘要。
MIB	Management Information Base (管理信息库)。

MIME	Multi-Purpose Internet Mail Extensions（多用途 Internet 邮件扩展）。一种正在兴起的用于多媒体电子邮件和消息传递的标准。
mime.types	MIME（Multi-purpose Internet Mail Extension，多用途 Internet 邮件扩展）类型配置文件。该文件将文件扩展名映射为 MIME 类型，以使服务器能够确定所请求的内容类型。例如，对扩展名为 .html 的资源发出请求表示客户机正在请求 HTML 文件，而对扩展名为 .gif 的资源发出请求表示客户机正在请求 GIF 格式的图形文件。
modutil	为外部加密或硬件加速器设备安装 PKCS#11 模块所需的软件实用程序。
MTA	Message Transfer Agent（消息传输代理）。要在服务器上使用代理服务，必须定义服务器的 MTA 主机。
network management station, NMS（网络管理站）	可供用户远程管理网络的计算机。被管理的设备可以是运行 SNMP 的任何设备（如主机、路由器和 Web 服务器）。NMS 通常是安装有一个或多个网络管理应用程序的功能强大的工作站。
NIS (UNIX)	Network Information Service（网络信息服务）。UNIX 计算机为在整个计算机网络中收集、整理和共享有关计算机、用户、文件系统和网络参数的特定信息而使用的程序及数据文件系统。
NNTP	适用于新闻组的 Network News Transfer Protocol（网络新闻传输协议）。要在服务器上使用代理服务，必须定义新闻服务器主机。
obj.conf	服务器的目标配置文件。该文件包含附加初始化信息、服务器的自定义设置以及服务器用来处理客户机（如浏览器）请求的指令。Sun Java System Web Server 每次处理客户机请求时都要读取该文件。
password file（密码文件） (UNIX)	UNIX 计算机上用于存储 UNIX 用户登录名、密码和用户 ID 编号的文件。由于其位置的原因，该文件也被称为 /etc/passwd。
pk12util	从内部计算机导出证书和密钥数据库并将它们导入外部 PKCS#11 模块所需的软件实用程序。
private key（私钥）	在公共密钥加密中使用的解密密钥。
protocol（协议）	描述网络上的设备如何交换信息的一组规则。
public information directories（公共信息目录） (UNIX)	不在文档根目录下而在某个 UNIX 用户的主目录下的目录，或由用户控制的目录。
public key（公钥）	在公钥加密中使用的加密密钥。
Quality of Service（服务质量）	为服务器实例、虚拟服务器类或虚拟服务器设置的性能限制。
RAM	Random access memory（随机存取存储器）。计算机中基于半导体的物理内存。

rc.2.d (UNIX)	UNIX 计算机中对计算机启动时运行的程序进行说明的文件。由于其位置的原因，该文件也被称为 <code>/etc/rc.2.d</code> 。
redirection (重定向)	一种系统，用于将访问特定 URL 的客户机发送到相同或不同服务器上的某个不同位置。如果资源被移动，并且希望客户机透明地使用新位置，该系统会很有用。当没有使用结尾斜杠来访问目录时，该系统还可用来维护相对链接的完整性。
resource (资源)	服务器可以访问并将其发送给提出请求的客户机的任何文档(URL)、目录或程序。
RFC	Request For Comments (互联网信息文档和标准)。通常是指提交给 Internet 社区的过程或标准文档。人们可以在某些技术被接受为标准之前发送有关这些技术的意见。
root (UNIX)	UNIX 计算机上具有最高权限的用户 (也称为 root 用户)。root 用户对计算机中的所有文件具有完全访问权限。
server daemon (服务器守护进程)	运行后，将侦听和接受来自客户机的请求的进程。
Server Plug-in API (服务器插件 API)	一种扩展，允许扩展和/或自定义 Sun Java System Web Server 的核心功能，并且为在 HTTP 服务器和后端应用程序之间建立接口提供了一种可伸缩的有效机制。也称为 NSAPI。
server root (服务器根目录)	服务器计算机上的目录，专门用于保存服务器程序、配置、维护和信息文件。
simple index (简单索引)	与复杂索引相对—这种类型的目录列表仅显示文件的名称而不显示任何图形元素。
SNMP	Simple Network Management Protocol (简单网络管理协议)。
SOCKS	用于在防火墙内部和外部之间建立连接的防火墙软件；如果不使用它而直接进行连接，会被防火墙软件或硬件 (如路由器配置) 禁止。
soft restart (软重启)	一种重新启动服务器的方法，使服务器可以在内部重新启动，即重新读取其配置文件。软重启将 HUP 信号 (1 号信号) 发送给进程。进程本身不会像在硬重启中那样终止。
SSL	Secure Sockets Layer (安全套接字层)。一种用于在双方 (客户机和服务器) 之间建立安全连接的软件库，用来实现 HTTPS (HTTP 的安全版本)。
SSL authentication (SSL 验证)	使用安全证书来确认用户的身份，即使用客户机证书中的信息作为身份证明，或验证在 LDAP 目录中发布的客户机证书。
stop word (停用词)	所标识出的搜索功能不对其进行搜索的词。通常包括 the、a、an、and 这样的词。也称为丢弃词。
strftime	用于将日期和时间转换为字符串的函数。该函数由服务器在附加尾缀时使用。对于日期和时间， <code>strftime</code> 有一种特殊的格式语言，服务器可在尾缀中使用这种格式语言以说明文件的上次修改日期。

Sun Java System Web Server Administration Console (Sun Java System Web Server 管理控制台)	为服务器管理员提供图形界面的 Java 应用程序，以便从企业网络中的任何一个中心位置管理所有 Sun Java System Web Server。通过所安装的任何 Sun Java System Web Server 管理控制台实例，您都可以查看并访问您有权访问的企业网络上的所有 Sun Java System 服务器。
superuser (超级用户) (UNIX)	UNIX 计算机上最具特权的用户（也称为 root 用户）。超级用户对计算机中的所有文件具有完全访问权限。
Sym-links (符号链接) (UNIX)	符号链接 (symbolic link) 的缩写，是 UNIX 操作系统使用的一种重定向类型。可以使用符号链接创建一个从文件系统的一部分到该文件系统另一部分的现有文件或目录的指针。
TCP/IP	Transmission Control Protocol/Internet Protocol（传输控制协议/Internet 协议）。Internet 和企业（公司）网络的主要网络协议。
telnet	用于使网络中的两台计算机互相连接并支持远程登录终端仿真的协议。
timeout (超时)	一段指定的时间，超过该时间后，服务器将放弃完成某个显示为挂起的服务例程。
TLS	Secure Sockets Layer（安全套接字层）。一种用于在双方（客户机和服务器）之间建立安全连接的软件库，用来实现 HTTPS（HTTP 的安全版本）。
top (UNIX)	某些 UNIX 系统上用于显示系统资源当前使用状况的程序。
top-level domain authority (顶级域授权)	最高类别的主机名分类，通常表示域所在的组织的类型（例如，.com 代表公司，.edu 代表教育机构）或域的原属国家/地区（例如，.us 代表美国，.jp 代表日本，.au 代表澳大利亚，.fi 代表芬兰）。
uid (UNIX)	与 UNIX 系统上每个用户相关联的唯一编号。
URI	Uniform Resource Identifier（统一资源标识符）。一种文件标识符，通过使用缩写的 URL 提供了额外一层安全性。一个 URL 映射代替了 URL 的第一部分，从而对用户隐藏了文件的完整物理路径名。请参见 URL 映射。
URL	Uniform Resource Locator（统一资源定位器）。服务器和客户机用来请求文档的寻址系统。URL 通常称为位置。URL 的格式为 <i>protocol://machine:port/document</i> 。 例如，http://www.sun.com/index.html 就是一个 URL。
URL database repair (URL 数据库修复)	对因软件故障、系统崩溃、磁盘故障或整个文件系统而损坏的 URL 数据库进行修复及更新的过程。
URL mapping (URL 映射)	将文档目录的物理路径名映射到用户定义的别名的过程，这样，该目录中的文件只需引用目录的别名而不是文件的完整物理路径名。从而，可以将某个文件标识为 /myDocs/index.html 而不是 usr/sun/servers/docs/index.html。用户不必知道服务器文件的物理位置，这为服务器提供了额外的安全性。
virtual server class (虚拟服务器类)	共享 obj.conf 文件中相同的基本配置信息的虚拟服务器集合。

virtual server (虚拟服务器)	虚拟服务器是通过安装一个服务器即可设置多个域名、IP 地址及服务器监视功能的一种方法。
Web Application Archive, WAR (Web 应用程序归档)	包含压缩格式的完整 Web 应用程序的归档文件。
web application (Web 应用程序)	Web 应用程序是 Servlet、JavaServer Pages、HTML 文档和其他 Web 资源 (可能包括图像文件、压缩的归档文件和其他数据) 的集合。Web 应用程序可以打包至一个归档文件 (WAR 文件) 中, 或存在于打开的目录结构中。
Windows CGI (Windows)	用基于 Windows 的编程语言 (如 Visual Basic) 编写的 CGI 程序。

索引

A

- ACL, 服务器摘要验证过程, 92
- ACL 用户高速缓存, 服务器存储用户和组验证结果, 93
- ACLCacheLifetime, 93
- ACLUserCacheSize, 93
- Administration Server
 - URL 导航, 27
 - 从“控制面板”启动服务 applet, 26
- ansi_x3.4-1968, 123
- ansi_x3.4-1986, 123
- ascii, 123

C

- CA
 - 定义（证书授权机构）, 73, 76
- certmap.conf, 91
- CGI, 122
 - shell, 121-122
 - 程序, 如何在服务器上存储, 118
 - 概述, 117
 - 文件扩展名, 119
 - 下载可执行文件, 121
- CGIStub, 帮助 CGI 执行的进程, 117
- CGIStubIdleTimeout, 117
- COPY, 139
- cp367.0, 123
- cp819, 123
- CRL（Certificate Revocation List, 证书撤销列表）,
 - 安装和管理, 81

- current.zip, 32

D

- DELETE, 98
- digestauth, 92
- DigestStaleTimeout, 92
- Directory Server, ldapmodify 命令行实用程序, 105
- DNS, 减少查找对服务器性能的影响, 93

G

- GET, 98
- GIF, 定义的, 231

H

- HEAD, 98
- .htaccess, 动态配置文件, 98
- HTML
 - 定义的, 231
 - 服务器解析的, 设置, 126
- HTTP, 定义的, 231
- http_head, 98
- HTTPD, 231
- HTTPS, 定义的, 231

I

- ibm367.0, 123
- ibm819, 123
- INDEX, 98
- inittab, 定义的, 231
- IP 地址
 - 定义的, 232
 - 限制访问, 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, 管理资源, 156
- JDBC, JDBC API, 157
- JSP 标记规范, 186

L

- latin1, 123
- LDAP
 - 管理用户和组, 101
 - 用户名和密码验证, 90, 229
- ldapmodify, Directory Server 命令行实用程序, 105
- LDIF
 - 导入和导出功能, 关于, 103
 - 添加数据库条目, 103
- LOCK, 139

M

- magnus.conf
 - ACLCacheLifetime 指令, 93
 - 终止超时, 92
- MaxCGIStub, 117
- MD5, 定义的, 232

- memberCertDescriptions, 107
- memberURL 过滤器, 107
- memberURLs, 107
- MIME
 - charset 参数, 123
 - octet-stream, 121
- MIME, 定义的, 233
- MIME 类型, 指定默认, 112-113
- MinCGIStub, 117
- MKCOL, 139
- MKDIR, 98
- MOVE, 98, 139
- MTA, 定义的, 233

N

- NIS, 定义的, 233
- NNTP, 定义的, 233
- nonce, 92

O

- obj.conf, 默认验证, 90
- octet-stream, 121

P

- PathCheck, 98
- POST, 98
- PROPFIND, 139
- PROPPATCH, 139
- PUT, 98

R

- RAM, 定义的, 233
- rc.2.d, 234
- RMDIR, 98
- root, 定义的, 234

S

shell CGI, 121-122
 SMUX, 192
 SNMP
 基本, 189
 在服务器上设置, 190, 191, 192
 子代理, 189, 191
 SOCKS, 定义的, 234
 SSL
 定义的, 234
 启用时需要的信息, 77
 验证, 91
 SSL 2 协议, 85
 SSL 3 协议, 84, 85
 SSL2 协议, 84
 SSL3 协议, 84

T

telnet, 235
 TLS 加密协议, 85
 TLS 协议, 84
 TLS 传输层安全, 84

U

uid, 定义, 235
 uniqueMembers, 107
 UNLOCK, 139
 URI, 定义的, 235
 URL
 定义的, 235
 访问 Administration Server, 27
 映射, 定义的, 235
 URL 转发, 配置, 115
 us, 123
 us-ascii, 123

W

Web 应用程序, 定义的, 236

Web 应用程序归档 (web application archive, WAR),
 定义的, 236
 Web 站点, 限制访问 (全局和单个实例), 94
 WebDAV
 Sun Java System Web Server 处理锁定请求的方
 式, 146
 URI, 136
 成员 URI, 137
 方法, 138
 COPY, 139
 LOCK, 139
 MKCOL, 139
 MOVE, 139
 PROPFIND, 139
 PROPPATCH, 139
 UNLOCK, 139
 集合, 137
 内部成员 URI, 137
 启用了 WebDAV 的客户机, 135
 属性, 137
 新的 HTTP 标头, 138
 新的 HTTP 方法, 138
 源 URI, 136

X

x-euc-jp, 123
 x-mac-roman, 123
 x-sjis, 123

安

安全套接字层 (Secure Sockets Layer, SSL), 加密的通
 信协议, 84

标

标识名 (Distinguished Name, DN) 属性, 定义, 102

超

超级用户, 定义的, 235

成

成员 URI, 137

程

程序

CGI

如何在服务器上存储, 118

错

错误, 自定义响应, 122

导

导航, 通过 URL 访问 Administration Server, 27

到

到期标头, 定义的, 230

顶

顶级域授权, 235

丢

丢弃词, 230

读

读取访问, 98

多

多字节数据, 201

反

反向代理, 130

访

访问

读取, 98

对 Web 站点, 限制 (全局和单个实例), 94

列表, 98

删除, 98

写入, 98

信息, 98

执行, 98

访问控制

方法 (基本、SSL), 90

概述, 87

介绍, 88-89

用户和组, 88

主机名和 IP 地址, 88

访问控制列表 (access control list, ACL), 87

访问控制条目 (access control entry, ACE), 87

访问日志轮转, 198

服

服务器, LDAP 用户和组, 国际化注意事项, 201

服务器根目录, 定义的, 234

服务器守护进程, 定义的, 234

服务器验证, 定义, 74

符

符号 (软) 链接, 定义, 125

符号链接, 限制, 125-126

高

高速缓存, 定义的, 229
高速缓存控制指令, 设置, 127

公

公共目录, 配置, 113
公共目录 (Unix), 自定义, 113-115
公共网关接口 (Common Gateway Interface, CGI), 概述, 117
公钥, 73

管

管理界面, 更多信息有关, 20

归

归档, 日志文件, 198

国

国际化注意事项, LDAP 用户和组, 201

过

过滤器, memberURL, 107

集

集合, 定义的, 230

加

加密, 双向, 84
加密算法, 定义, 84

检

检测版本回滚, 85

接

接受语言标头, 使用, 202-203

静

静态组
创建指导原则, 108
定义, 107

可

可执行文件, 下载, 121

客

客户机验证, 定义, 74
客户机证书, 验证, 91

控

控制, 访问, 概述, 87

列

列表访问, 98

轮

轮转, 访问日志, 198

密

密码文件, 233

密码文件（续）
 启动时装入, 115
密钥, 定义, 84

内
内部成员 URI, 137
内容压缩
 插入 Vary 标头, 129
 根据需要压缩内容, 129-130
 激活, 128
 配置内容压缩, 128-130
 片断大小, 129
 提供预压缩的内容, 128-129
 压缩级别, 129

启
启动命令, Unix 平台, 26

轻
轻量目录访问协议 (LDAP), 管理用户和组, 101

请
请求/摘要, 92

日
日志文件, 归档, 198

软
软（符号）链接, 定义, 125

删
删除访问, 98

生
生命周期模块, 154

实
实例, 术语, 35

属
属性, 标识名 (Distinguished Name, DN), 102

数
数据库条目, 使用 LDIF 添加, 103

双
双向加密, 加密算法, 84

搜
搜索
 JSP 标记规范, 186
 URI, 174
 查看搜索结果, 181
 查询, 179-180
 高级搜索, 180
 关于, 173-174
 界面组件, 182
 路径, 174
 搜索页面, 179
 在单独的页面中自定义表单和结果, 185
 自定义搜索查询页面, 182-183
 自定义搜索结果页面, 184-185
 自定义搜索页面, 181-186

搜索基 (基 DN), 用户 ID, 105

锁

锁定资源

Sun Java System Web Server 处理锁定请求的方式, 146

共享锁定, 145

互斥锁定, 145

停

停用词, 234

通

通用日志文件格式, 定义, 230

椭

椭圆曲线加密算法, 74

外

外部网, 定义的, 230

网

网络管理站 (Network Management Station, NMS), 189, 191

文

文档根目录, 设置, 111

文档目录

限制内容发布, 114

主 (文档根目录), 111

文档首选项, 默认 MIME 类型, 指定, 112-113

文档页脚, 设置, 124

文件扩展名

CGI, 119

定义的, 231

文件类型, 定义的, 231

限

限制符号链接, 125-126

写

写入访问, 98

信

信息访问, 98

虚

虚拟存储器, 示例, 海量托管, 67

虚拟服务器

部署, 65

公共目录, 配置以使用, 113-115

简介, 65

示例, 安全的服务器, 66

示例, 默认配置, 66

示例, 内部网托管, 66-67

验

验证

SSL, 91

客户机证书, 91

用户和组, 89-93

主机名, 93

验证, 基本, 与 SSL 加密组合时最有效, 主机/IP 验证, 或二者, 90

验证, 客户机, 服务器, 定义, 74

验证, 用户/组, 90, 93

验证,摘要, 91
验证,主机/IP, 93
验证超时, 85
验证数据库, 103

硬

硬链接,定义, 125

用

用户
 限制访问, 88
 验证, 89-93
用户/组验证, 90, 93
用户和组,使用 LDAP 管理, 101
用户和组验证,存储在 ACL 用户高速缓存中的结果, 93
用户目录,配置, 113
用户目录 (Unix),自定义, 113-115

语

语言标头,接受,使用, 202-203

域

域名系统
 别名,定义的, 230
 定义的, 230

源

源 URI, 136

摘

摘要验证, 91
 ACL 的服务器过程, 92

证

证书
 简介, 73, 76
证书,客户机,验证, 91
证书撤销列表 (Certificate Revocation List, CRL), 安装和管理, 81
证书请求,需要的信息, 77
证书授权机构
 定义, 73, 76

执

执行访问, 98

终

终止超时,magnus.conf, 92

重

重定向, 234

主

主机/IP 验证, 93
主机名
 定义的, 231
 限制访问, 88
 验证, 93
主文档目录,设置 (文档根目录), 111

传

传输层安全 (Transport Layer Security, TLS), 加密的通信协议, 84

资

资源,定义的, 234

子

子代理

SNMP, 189, 191

字

字符集

iso_8859-1, 123

us-ascii, 123

更改, 123-124

自

自定义搜索, 182-183

在单独的页面中自定义表单和结果, 185

自定义搜索结果页面, 184-185

组

组

LDAP 数据库中用于描述一组对象的对象, 107

限制访问, 88

验证, 89-93

验证, 用户, 90

组, 静态

创建指导原则, 108

定义, 107

最

最大连接数, 133

最大验证数据量, 85

最大传输速率, 133

