



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



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

文件号码 820-4610
2007 年 12 月

版权所有 2007 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. 保留所有权利。

本产品或文档受版权保护。其使用、复制、发行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 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 Sun™ 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，对于所有明示或默示的条件、陈述和担保，包括对适销性、适用性和非侵权性的默示保证，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	15
1 Application Server 中的高可用性	21
高可用性概述	21
高可用性会话持久性	22
高可用性 Java 消息服务	22
RMI-IIOP 负载均衡和故障转移	23
更多信息	23
Application Server 如何提供高可用性	24
负载均衡器插件	24
会话状态数据的存储	24
高可用性群集	26
从故障中恢复	27
使用 Sun Cluster	27
手动恢复	27
使用 Netbackup	29
重新创建域管理服务器	30
▼ 迁移 DAS	30
2 安装和设置高可用性数据库	33
设置 HADB 前的准备工作	33
先决条件和限制	34
配置网络冗余	34
配置共享内存和信号	37
▼ 在 Solaris 上配置共享内存和信号	37
▼ 在 Linux 上配置共享内存	38
同步系统时钟	39

安装	40
HADB 安装	40
节点监控进程权限	40
▼ 授予节点监控进程 root 用户权限	41
设置高可用性	42
▼ 准备高可用性系统	42
启动 HADB 管理代理	42
配置高可用性群集	43
配置高可用性应用程序	43
重新启动群集	43
重新启动 Web Server	44
▼ 清除作为负载均衡器使用的 Web Server 实例	44
升级 HADB	44
▼ 将 HADB 升级到更新版本	45
注册 HADB 软件包	45
取消注册 HADB 软件包	46
替换管理代理的启动脚本	47
▼ 验证 HADB 升级	47
3 管理高可用性数据库	49
使用 HADB 管理代理	49
启动管理代理	50
管理代理命令语法	54
自定义管理代理配置	55
▼ 在 HADB 主机上自定义管理代理配置	55
使用 hadbm 管理命令	57
命令语法	57
安全性选项	58
常规选项	59
环境变量	60
配置 HADB	61
创建管理域	61
创建数据库	62
▼ 创建数据库	63
查看和修改配置属性	67

配置 JDBC 连接池	72
管理 HADB	74
管理域	75
管理节点	76
管理数据库	78
从会话数据损坏恢复	82
▼ 使会话存储恢复到一致状态	82
扩展 HADB	83
为现有节点添加存储空间	83
添加计算机	84
▼ 将新计算机添加到现有 HADB 实例	84
添加节点	84
重新分段数据库	86
通过重新创建数据库添加节点	87
▼ 通过重新创建数据库来添加节点	87
监视 HADB	88
获取 HADB 的状态	88
获取设备信息	90
获取运行时资源信息	92
维护 HADB 计算机	94
▼ 在一台计算机上执行维护	95
▼ 在所有 HADB 计算机上执行规划的维护	95
▼ 在所有 HADB 计算机上执行规划的维护	95
▼ 在出现故障时执行非规划的维护	96
清除和归档历史文件	96
4 配置 Web 服务器以实现负载均衡	99
配置 Sun Java System Web Server	100
▼ 配置 Sun Java System Web Server	100
配置 Sun Java System Web Server 以使用“自动应用”功能	101
▼ 在 SSL 模式下为 Sun Java System Web Server 6.1 设置负载均衡器	101
▼ 为 Sun Java System Web Server 6.1 导出和导入 DAS 证书	103
在 SSL 模式下为 Web Server 7 设置负载均衡器	104
▼ 为 Sun Java System Web Server 7 导出和导入 DAS 证书	105
使用 Apache Web Server	107

对使用 Apache Web Server 的要求	108
在安装负载均衡器插件之前配置 Apache	109
▼ 安装 SSL 可识别 Apache	109
导出和导入 DAS 证书	112
负载均衡器插件安装程序所做的修改	112
在安装负载均衡器插件之后配置 Apache	113
▼ 为 Apache 创建安全性证书	113
在 Solaris 和 Linux 上启动 Apache	114
验证安装	114
使用 Microsoft IIS	115
▼ 配置 Microsoft IIS 以使用负载均衡器插件	115
自动配置的 sun-passthrough 属性	117
5 配置 HTTP 负载均衡	119
负载均衡器插件的新增功能	119
自动应用	119
加权 Round Robin	120
用户定义的负载均衡	120
▼ 配置用户定义的负载均衡	120
HTTP 负载均衡器如何工作	121
HTTP 负载均衡算法	121
设置 HTTP 负载均衡	122
设置负载均衡的先决条件	122
设置负载均衡的步骤	123
▼ 使用管理控制台设置负载均衡	123
▼ 使用 asadmin 工具设置负载均衡	124
HTTP 负载均衡器部署	126
配置负载均衡器	126
在 DAS 上配置 HTTP 负载均衡器	127
创建 HTTP 负载均衡器引用	128
启用用于负载均衡的服务器实例	128
启用用于负载均衡的应用程序	129
创建 HTTP 运行状况检查器	129
导出负载均衡器配置文件	131
▼ 使用管理控制台导出负载均衡器配置	131

▼ 使用 asadmin 工具导出负载均衡器配置	131
更改负载均衡器配置	132
启用动态重新配置	132
禁用（停止）服务器实例或群集	133
▼ 禁用服务器实例或群集	133
禁用（停止）应用程序	133
▼ 禁用应用程序	134
配置 HTTP 和 HTTPS 故障转移	134
在负载均衡器中使用重定向	135
配置幂等 URL	138
配置多个 Web 服务器实例	139
▼ 配置多个 Web 服务器实例	139
升级应用程序而不使可用性受到损失	139
应用程序兼容性	139
在单个群集中升级	140
▼ 在单个群集中升级应用程序	140
在多个群集中进行升级	142
▼ 在两个或多个群集中升级兼容的应用程序：	142
升级不兼容的应用程序	144
▼ 通过创建第二个群集来升级不兼容的应用程序	144
监视 HTTP 负载均衡器插件	145
配置日志消息	145
日志消息类型	146
启用负载均衡器日志记录	147
▼ 打开负载均衡器日志记录	147
了解监视消息	148
6 使用 Application Server 群集	149
群集概述	149
组管理服务	149
▼ 对群集启用或禁用 GMS	150
配置 GMS	150
使用群集	150
▼ 创建群集	151
▼ 为群集创建服务器实例	152

▼ 配置群集	153
▼ 启动、停止和删除群集实例	153
▼ 在群集中配置服务器实例	154
▼ 为群集配置应用程序	155
▼ 为群集配置资源	155
▼ 删除群集	156
▼ 迁移 EJB 计时器	156
▼ 升级组件而不使服务受到任何损失	157
7 管理配置	159
使用配置	159
配置	159
default-config 配置	160
创建实例或群集时创建的配置	160
唯一端口号和配置	161
使用命名配置	162
▼ 创建命名配置	162
编辑命名配置的属性	162
▼ 编辑命名配置的属性	163
▼ 编辑引用配置的实例的端口号	163
▼ 查看命名配置的目标	164
▼ 删除命名配置	164
8 配置节点代理	167
什么是节点代理?	167
节点代理出现故障后的服务器实例行为	168
部署节点代理	169
▼ 联机部署节点代理	169
▼ 脱机部署节点代理	170
同步节点代理和域管理服务器	171
节点代理同步	171
服务器实例同步	172
同步库文件	173
独特的设置和配置管理	174
同步大型应用程序	174

查看节点代理日志	175
处理节点代理	175
如何执行节点代理任务	176
节点代理占位符	176
▼ 创建节点代理占位符	176
创建节点代理	177
▼ 创建节点代理	178
▼ 为可通过 DNS 访问的主机创建节点代理	178
启动节点代理	179
停止节点代理	179
删除节点代理	180
▼ 查看常规节点代理信息	180
▼ 删除节点代理配置	181
▼ 编辑节点代理配置	182
▼ 编辑节点代理区域	182
▼ 编辑节点代理的 JMX 侦听器	182
9 配置高可用性会话持久性和故障转移	185
会话持久性和故障转移概述	185
要求	185
限制	186
设置高可用性会话持久性	187
▼ 设置高可用性会话持久性	187
启用会话可用性	188
HTTP 会话故障转移	189
为 Web 容器配置可用性	190
▼ 使用管理控制台启用 Web 容器的可用性	190
为单个 Web 应用程序配置可用性	192
结合使用会话故障转移和单点登录	193
有状态会话 Bean 故障转移	194
为 EJB 容器配置可用性	195
▼ 启用 EJB 容器的可用性	195
为单个应用程序或 EJB 模块配置可用性	196
为单个 Bean 配置可用性	197
指定对其执行检查点操作的方法	197

10	Java 消息服务的负载均衡和故障转移	199
	Java 消息服务概述	199
	更多信息	199
	配置 Java 消息服务	200
	Java 消息服务集成	201
	JMS 主机列表	201
	连接池和故障转移	202
	负载均衡的消息流入	203
	JMS 服务高可用性	204
	结合使用 Application Server 与 MQ 群集	204
	高可用性 MQ 群集	205
	在本地模式下配置高可用性代理群集	205
	在远程模式下配置高可用性代理群集	206
	为非 HA 群集自动建立群集	206
	▼ 使用 Application Server 群集启用 MQ 群集	207
11	RMI-IIOP 负载均衡和故障转移	211
	概述	211
	要求	212
	算法	212
	设置 RMI-IIOP 负载均衡和故障转移	213
	▼ 为应用程序客户机容器设置 RMI-IIOP 负载均衡	213
	索引	217

表

表 2-1	hadbm registerpackage 选项	46
表 3-1	管理代理通用选项	55
表 3-2	管理代理服务选项（仅限于 Windows）	55
表 3-3	配置文件设置	56
表 3-4	hadbm 安全性选项	59
表 3-5	hadbm 常规选项	59
表 3-6	HADB 选项和环境变量	60
表 3-7	hadbm create 选项	63
表 3-8	配置属性	69
表 3-9	HADB 连接池设置	73
表 3-10	HADB 连接池属性	73
表 3-11	HADB JDBC 资源设置	74
表 3-12	hadbm clear 选项	81
表 3-13	hadbm addnodes 选项	85
表 3-14	HADB 状态	89
表 3-15	hadbm resourceinfo 命令选项	92
表 5-1	负载均衡器配置参数	127
表 5-2	运行状况检查器参数	130
表 5-3	运行状况检查器手动设置属性	130
表 8-1	在远程服务器实例之间同步的文件和目录	172
表 8-2	如何执行节点代理任务	176

示例

示例 2-1	设置多路经	35
示例 2-2	注销 HADB 的示例	47
示例 3-1	hadbm 命令示例	58
示例 3-2	创建 HADB 管理域	62
示例 3-3	创建数据库的示例	65
示例 3-4	使用 hadbm get 的示例	67
示例 3-5	创建连接池	74
示例 3-6	启动节点的示例	77
示例 3-7	停止节点的示例	77
示例 3-8	重新启动节点的示例	78
示例 3-9	启动数据库的示例	79
示例 3-10	停止数据库的示例	79
示例 3-11	删除数据库的示例	82
示例 3-12	设置数据设备大小的示例	84
示例 3-13	添加节点的示例	85
示例 3-14	重新分段数据库示例	87
示例 3-15	获取 HADB 状态的示例	89
示例 3-16	获取设备信息的示例	91
示例 3-17	数据缓冲池信息示例	93
示例 3-18	锁信息示例	93
示例 3-19	日志缓冲区信息的示例	94
示例 3-20	内部日志缓冲区信息示例	94
示例 8-1	创建节点代理	178
示例 9-1	已启用可用性的 EJB 部署描述符示例	197
示例 9-2	指定方法检查点操作的 EJB 部署描述符示例	198
示例 11-1	为 RMI-IIOP 加权 Round-Robin 负载均衡设置负载均衡权重	214

前言

本书介绍了 Application Server 中的高可用性功能，其中包括 HTTP 负载均衡、群集、会话持久性和故障转移，以及高可用性数据库 (High Availability Database, HADB)。

本前言包含有关整个 Sun Java™ System Application Server 文档集的信息及其约定。

Application Server 文档集

Application Server 文档集介绍了部署规划和系统安装。Application Server 文档的统一资源定位器 (Uniform Resource Locator, URL) 为 <http://docs.sun.com/coll/1343.4> 和 <http://docs.sun.com/coll/1578.2>。有关 Application Server 的介绍，请按下表所列顺序参阅相关书籍。

表 P-1 Application Server 文档集中的书籍

书名	说明
文档中心	按任务和主题组织的 Application Server 文档主题。
发行说明	软件和文档的最新信息。其中包括以表格形式对所支持的硬件、操作系统、Java Development Kit (JDK™) 和数据库驱动程序所做的全面汇总。
快速入门指南	如何开始使用 Application Server 产品。
安装指南	安装软件及其组件。
部署规划指南	评估系统需求和企业状况，确保以最适合您的站点的方式部署 Application Server。此外还介绍了部署服务器时应该注意的常见问题。
应用程序部署指南	将应用程序和应用程序组件部署到 Application Server。其中包括有关部署描述符的信息。
开发者指南	创建和实现用于在 Application Server 上运行的、遵循 Java EE 组件和 API 的开放式 Java 标准模型的 Java Platform, Enterprise Edition (Java EE 平台) 应用程序。其中包括有关开发者工具、安全性、调试和创建生命周期模块的信息。
Java EE 5 教程	使用 Java EE 5 平台技术和 API 开发 Java EE 应用程序。

表 P-1 Application Server 文档集中的书籍 (续)

书名	说明
Java WSIT 教程	使用 Web 服务互操作性技术 (Web Service Interoperability Technologies, WSIT) 开发 Web 应用程序。介绍了如何使用、何时使用以及为什么使用 WSIT 技术以及各种技术支持的功能和选项。
管理指南	Application Server 的系统管理，其中包括配置、监视、安全性、资源管理和 Web 服务管理。
高可用性管理指南	有关高可用性数据库的安装后配置和管理说明。
管理参考	编辑 Application Server 配置文件 domain.xml。
升级和迁移指南	从旧版本的 Application Server 升级或者从竞争性应用服务器迁移 Java EE 应用程序。本指南还介绍了相邻产品发行版之间的差异以及可导致与产品规范不兼容的配置选项。
性能调试指南	调节 Application Server 以提高性能。
故障排除指南	解决 Application Server 问题。
错误消息参考	解析 Application Server 错误消息。
参考手册	可用于 Application Server 的实用程序命令，以手册页样式编写。其中包括 asadmin 命令行界面。

相关文档

Application Server 可以单独购买，也可以作为 Sun Java Enterprise System (Java ES) 的组件购买，Java ES 是支持分布在网络或 Internet 环境中的企业应用程序的软件基础结构。如果将 Application Server 作为 Java ES 的一个组件购买，您应熟悉 <http://docs.sun.com/coll/1286.3> 和 <http://docs.sun.com/coll/1382.3> 上的系统文档。有关 Java ES 及其组件的所有文档的 URL 为 <http://docs.sun.com/prod/entsys.5>。

有关其他独立 Sun Java System 服务器产品的文档，请访问：

- (<http://docs.sun.com/coll/1343.4>) 和 Message Queue 文档 (<http://docs.sun.com/coll/1578.2>)
- (<http://docs.sun.com/coll/1224.1>) 和 Directory Server 文档 (<http://docs.sun.com/coll/1606.1>)
- (<http://docs.sun.com/coll/1308.3>) 和 Web Server 文档 (<http://docs.sun.com/coll/1395.2>)

随 Application Server 提供的软件包的 Javadoc™ 工具参考位于 <http://glassfish.dev.java.net/nonav/javaee5/api/index.html> 上。此外，以下资源可能会有用：

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

有关在 NetBeans™ 集成开发环境 (Integrated Development Environment, IDE) 中创建企业应用程序的信息，请参见 <http://www.netbeans.org/kb/55/index.html>。

有关 Application Server 附带的 Java DB 数据库的信息，请参见 <http://developers.sun.com/javadb/>。

GlassFish Samples 项目是演示各种 Java EE 技术的样例应用程序的集合。GlassFish Samples 已与 Java EE 软件开发工具包 (Software Development Kit, SDK) 捆绑在一起，也可以从 <https://glassfish-samples.dev.java.net/> 上的 GlassFish Samples 项目页获取。

默认路径和文件名

下表介绍了在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	说明	默认值
<i>as-install</i>	表示 Application Server 的安装基目录。	Solaris™ 操作系统上的 Java ES 安装： /opt/SUNWappserver/appserver Linux 操作系统上的 Java ES 安装： /opt/sun/appserver/ 其他 Solaris 和 Linux 安装（非 root 用户）： <i>user's-home-directory/SUNWappserver</i> 其他 Solaris 和 Linux 的安装（root 用户）： /opt/SUNWappserver Windows 的所有安装： SystemDrive:\Sun\AppServer
<i>domain-root-dir</i>	表示包含所有域的目录。	Java ES Solaris 安装： /var/opt/SUNWappserver/domains/ Java ES Linux 安装： /var/opt/sun/appserver/domains/ 所有其他安装： <i>as-install/domains/</i>

表 P-2 默认路径和文件名 (续)

占位符	说明	默认值
<i>domain-dir</i>	表示域的目录。 在配置文件中，您可能会看到 <i>domain-dir</i> 显示为以下内容： <code>\${com.sun.aas.instanceRoot}</code>	<i>domain-root-dir/domain-dir</i>
<i>instance-dir</i>	表示服务器实例的目录。	<i>domain-dir/instance-dir</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
AaBbCc123	命令、文件和目录的名称；计算机屏幕输出	编辑 <code>.login</code> 文件。 使用 <code>ls -a</code> 列出所有文件。 <code>machine_name% you have mail.</code>
AaBbCc123	用户键入的内容，与计算机屏幕输出的显示不同	<code>machine_name% su</code> 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 <code>rm filename</code> 。
新词语强调	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

下表介绍了本书中使用的符号。

表 P-4 符号约定

符号	说明	示例	含义
[]	包含可选参数和命令选项。	<code>ls [-l]</code>	无需 <code>-l</code> 选项。

表 P-4 符号约定 (续)

符号	说明	示例	含义
{ }	包含为所需命令选项提供的一组选择。	-d {y n}	-d 选项需要您使用 y 变量或 n 变量。
\${ }	表示变量引用。	\${com.sun.javaRoot}	引用变量 com.sun.javaRoot 的值。
-	连接需同时按下的多个击键。	Control-A	同时按 Control 键和 A 键。
+	连接需连续按下的多个击键。	Ctrl+A+N	按 Control 键，然后松开并依次按后面的键。
→	表示图形用户界面中的菜单项选定。	“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”。

文档、支持和培训

Sun web 站点提供有关以下附加资源的信息：

- [文档](http://www.sun.com/documentation/) (http://www.sun.com/documentation/)
- [支持](http://www.sun.com/support/) (http://www.sun.com/support/)
- [培训](http://www.sun.com/training/) (http://www.sun.com/training/)

搜索 Sun 产品文档

除了从 docs.sun.comSM Web 站点搜索 Sun 产品文档外，还可以通过在搜索字段中键入以下语法来使用搜索引擎：

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

例如，要搜索 "broker"，请键入以下内容：

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

要在搜索中包括其他 Sun Web 站点（例如，java.sun.com、www.sun.com 和 developers.sun.com），请在搜索字段中使用 sun.com 代替 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、产品或其他资料，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"（发送意见）。在联机表单中，提供了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 820-4610。

Application Server 中的高可用性

本章介绍了 Sun Java System Application Server 中随群集配置文件和企业配置文件提供的高可用性功能。

注 - HADB 软件随 Sun Java System Application Server 的 Application Server 独立分发提供。有关 Sun Java System Application Server 的可用分发的信息，请参见《Sun Java System Application Server 9.1 Installation Guide》中的“Distribution Types and Their Components”。HADB 功能仅在企业配置文件中可用。有关配置文件的信息，请参见《Sun Java System Application Server 9.1 管理指南》中的“用法配置文件”。

本章包含以下主题。

- 第 21 页中的 “高可用性概述”
- 第 24 页中的 “Application Server 如何提供高可用性”
- 第 27 页中的 “从故障中恢复”

高可用性概述

无论硬件和软件故障，高可用性应用程序和服务均可持续提供其正常功能。这些应用程序有时被称为提供五个九可靠性的应用程序，因为它们 99.999% 的时间都可用。

Application Server 提供以下高可用性功能：

- 高可用性会话持久性
- 高可用性 Java 消息服务
- RMI-IIOP 负载平衡和故障转移

高可用性会话持久性

Application Server 提供了 HTTP 请求和会话数据（HTTP 会话数据和有状态会话 Bean 数据）的高可用性。

Java EE 应用程序通常具有大量会话状态数据。Web 购物车是会话状态的一个典型示例。此外，应用程序可以高速缓存会话对象中需要频繁使用的数据。事实上，几乎带有重要用户交互的所有应用程序都需要维护会话状态。HTTP 会话和有状态会话 Bean (stateful session bean, SFSB) 都具有会话状态数据。

保留故障服务器之间的会话状态对最终用户非常重要。为了实现高可用性，Application Server 为会话状态数据提供了以下类型的存储：

- 群集中其他服务器上的内存中复制
- 高可用性数据库 (High-availability database, HADB)

如果托管用户会话的 Application Server 实例出现故障，则可以恢复会话状态，并且会话可以继续而不会丢失信息。

有关如何设置高可用性会话持久性的详细说明，请参见第 9 章。

高可用性 Java 消息服务

Java 消息服务 (Java Message Service, JMS) API 是一种通讯标准，使 Java EE 应用程序和组件可以创建、发送、接收和读取消息。并启用了松散耦合的可靠异步分布式通信。Sun Java System Message Queue (MQ)（实现了 JMS）与 Application Server 紧密集成，使您可以创建诸如消息驱动 bean (message-driven bean, MDB) 之类的依赖 JMS 的组件。

通过连接池、故障转移和 MQ 群集，JMS 实现了高可用性。有关更多信息，请参见第 10 章。

连接池和故障转移

Application Server 支持 JMS 连接池和故障转移。Application Server 将自动实现 JMS 连接池。默认情况下，Application Server 从指定的主机列表中随机选择其主 MQ 代理。发生故障转移时，MQ 会将负载透明地转移到另一个代理，并维持 JMS 语义。

有关 JMS 连接池和故障转移的更多信息，请参见第 202 页中的“连接池和故障转移”。

MQ 群集

MQ 企业版支持多个互连代理实例（称为代理群集）。使用代理群集的情况下，客户机连接将分布在群集的所有代理中。群集可以提供水平可伸缩性并提高可用性。

有关 MQ 群集的更多信息，请参见第 204 页中的“结合使用 Application Server 与 MQ 群集”。

RMI-IIOP 负载均衡和故障转移

通过 RMI-IIOP 负载均衡，IIOP 客户机请求被分发到不同的服务器实例或名称服务器上，这样就会将负载平均地分布在群集中，从而提供了可伸缩性。与 EJB 群集和可用性结合的 IIOP 负载均衡还可提供 EJB 故障转移。

客户机执行对象的 JNDI 查找时，命名服务实际上会将请求绑定到特定的服务器实例。此后，所有从该客户机发出的查找请求都被发送到同一服务器实例，因此将在同一目标服务器上托管所有 EJBHome 对象。此后获得的所有 Bean 引用也创建在相同的目标主机上。这样就有效提供了负载均衡，因为所有客户机在执行 JNDI 查找时会随机使用目标服务器的列表。如果目标服务器实例发生故障，查找或 EJB 方法调用会将故障转移到另一个服务器实例。

IIOP 负载均衡和故障转移将透明地发生。在应用程序部署过程中无需特殊的步骤。如果在其上部署应用程序客户机的 Application Server 实例参与群集，则 Application Server 将自动查找群集中当前处于活动状态的所有 IIOP 端点。但是，客户机应该至少已指定两个端点以用于引导目的，以防其中一个端点出现故障。

有关 RMI-IIOP 负载均衡和故障转移的更多信息，请参见第 11 章。

更多信息

有关规划高可用性部署（包括评估硬件要求、规划网络配置和选择拓扑）的信息，请参见《Sun Java System Application Server 9.1 部署规划指南》。本手册还对以下概念进行了高层次的介绍：

- 应用程序服务器组件，如节点代理、域和群集
- 群集中的 IIOP 负载均衡
- HADB 体系结构
- 消息队列故障转移

有关开发利用高可用性功能的应用程序的更多信息，请参见《Sun Java System Application Server 9.1 Developer's Guide》。

调优高可用性服务器和应用程序

有关如何配置和调优应用程序和 Application Server 以获得高可用性的最佳性能的信息，请参见《Sun Java System Application Server 9.1 Performance Tuning Guide》，其中包括以下主题：

- 调优持久性频率和持久性范围
- 对有状态会话 Bean 执行检查点操作
- 配置 JDBC 连接池
- 会话大小
- 调优 HADB 磁盘使用、内存分配、性能和操作系统配置

- 配置负载均衡器以获得最佳性能

Application Server 如何提供高可用性

Application Server 通过以下子组件和功能提供高可用性：

- 第 24 页中的“负载均衡器插件”
- 第 24 页中的“会话状态数据的存储”
- 第 26 页中的“高可用性群集”

负载均衡器插件

负载均衡器插件接受 HTTP/HTTPS 请求，然后将请求转发至群集中的应用程序服务器实例。如果实例出现故障，变得不可用（由于网络故障）或无法响应，负载均衡器会将请求重定向至现有的可用计算机。负载均衡器还可识别故障实例何时恢复并相应地重新分布负载。Application Server 提供了用于 Sun Java System Web Server 和 Apache Web Server 以及 Microsoft Internet Information Server 的负载均衡器插件。

负载均衡器通过在多台物理计算机中分布工作量来提高系统的整体吞吐量。它还可通过对 HTTP 请求的故障转移提供更高的可用性。对于要保留的 HTTP 会话信息，必须配置 HTTP 会话持久性。

对于简单的无状态应用程序，负载均衡群集可能足够了。但是，对于具有会话状态的重点应用程序，请将负载均衡群集与 HADB 一起使用。

参与负载均衡的服务器实例和群集具有同构环境。通常，这意味着服务器实例均引用相同的服务器配置、可以访问相同的物理资源，以及具有部署到其上的相同的应用程序。同构环境确保了在出现故障前后，负载均衡器可以始终在群集中的活动实例之间平均分布负载。

有关配置负载均衡和故障转移的信息，请参见第 5 章。

会话状态数据的存储

通过存储会话状态数据，可以在群集中的服务器实例故障转移后恢复会话状态。恢复会话状态可使会话继续进行而不会丢失信息。Application Server 为 HTTP 会话和有状态会话 Bean 数据提供了以下类型的高可用性存储：

- 群集中其他服务器上的内存中复制
- 高可用性数据库

群集中其他服务器上的内存中复制

其他服务器上的内存中复制提供会话状态数据的轻量存储，而无需获取单独的数据库（如 HADB）。此类型的复制可使用其他服务器上的内存来实现 HTTP 会话和有状态会话 Bean 数据的高可用性存储。群集服务器实例在环形拓扑中复制会话状态。每个备份实例均在内存中存储复制数据。通过在其他服务器上的内存中复制会话状态数据，可以分布会话。

使用内存中复制要求启用组管理服务 (Group Management Service, GMS)。有关 GMS 的更多信息，请参见第 149 页中的“组管理服务”。

如果群集中的服务器实例位于不同的计算机上，请确保满足以下先决条件：

- 要确保 GMS 和内存中复制正常工作，这些计算机必须在同一子网上。
- 要确保内存中复制正常工作，必须同步群集中所有计算机上的系统时钟使其尽可能接近。

高可用性数据库

注 - HADB 软件随 Sun Java System Application Server 的 Application Server 独立分发提供。有关 Sun Java System Application Server 的可用分发的信息，请参见《Sun Java System Application Server 9.1 Installation Guide》中的“Distribution Types and Their Components”。HADB 功能仅在企业配置文件中可用。有关配置文件的信息，请参见《Sun Java System Application Server 9.1 管理指南》中的“用法配置文件”。

Application Server 提供了高可用性数据库 (High Availability Database, HADB)，以实现 HTTP 会话和有状态会话 Bean 数据的高可用性存储。HADB 旨在通过负载平衡、故障转移和状态恢复等功能支持高达 99.999% 的服务和数据可用性。通常，您必须独立于 Application Server 来配置和管理 HADB。

不让 Application Server 承担状态管理职责会具有很多好处。Application Server 实例在它们的周期中作为可伸缩高性能应用程序容器执行，将状态复制委托给外部高可用性状态服务。由于采用这种松散耦合的体系结构，因此可以很轻松地群集中添加 Application Server 实例或从群集中删除 Application Server 实例。HADB 状态复制服务可以单独伸缩，以获得最佳的可用性和性能。如果 Application Server 实例同时还执行复制任务，Java EE 应用程序的性能将会降低，并会受到较长的垃圾收集暂停的限制。

有关规划和设置应用服务器安装（包括确定硬件配置、调整大小和拓扑）以通过 HADB 实现高可用性的信息，请参见《Sun Java System Application Server 9.1 部署规划指南》中的“Planning for Availability”和《Sun Java System Application Server 9.1 部署规划指南》中的第 3 章“选择拓扑”。

高可用性群集

群集是作为一个逻辑实体一起工作的 Application Server 实例的集合。群集为一个或多个 Java EE 应用程序提供了运行时环境。**高可用性群集**将状态复制服务与群集和负载均衡器集成在一起。

使用群集具有以下优点：

- **高可用性**（通过允许为群集中的服务器实例提供故障转移保护来实现）。如果一个服务器实例出现故障，其他服务器实例将接管该服务器实例正在服务的请求。
- **可伸缩性**（通过允许向群集中添加服务器实例从而增加系统的容量来实现）。负载均衡器插件会将请求分布到群集中的可用服务器实例。当管理员向群集中添加更多服务器实例时，无需中断服务。

群集中的所有实例具有以下特性：

- 引用相同的配置。
- 具有相同的一组已部署应用程序（例如，Java EE 应用程序 EAR 文件、Web 模块 WAR 文件或 EJB JAR 文件）。
- 具有相同的一组资源，因此具有相同的 JNDI 名称空间。

域中的每一个群集都具有唯一的名称；此外，该名称在所有节点代理名称、服务器实例名称、群集名称和配置名称中也必须是唯一的。此名称不能为 `domain`。您在群集上执行的操作与在非群集服务器实例上执行的操作相同（例如，部署应用程序和创建资源）。

群集和配置

群集的设置源自该群集可能与其他群集共享的命名配置。其配置不能被其他服务器实例或群集所共享的群集可视为具有**独立配置**。默认情况下，此配置的名称为 `cluster_name-config`，其中 `cluster_name` 表示群集的名称。

能与其他群集或实例共享其配置的群集可视为具有**共享配置**。

群集、实例、会话和负载均衡

群集、服务器实例、负载均衡器和会话的相互关系如下：

- 服务器实例不需要属于群集。但是，不属于群集的实例无法通过将会话状态从一个实例转移到其他实例来利用高可用性。
- 群集中的服务器实例可以由一台或多台计算机托管。您可以将不同计算机上的服务器实例组织为一个群集。
- 特定负载均衡器可以向多个群集中的服务器实例转发请求。您可以使用负载均衡器的此功能来执行联机升级，而不使服务受到损失。有关更多信息，请参见“配置群集”一章中的“使用多个群集进行联机升级而不使服务受到损失”。

- 一个群集可以从多个负载均衡器接收请求。如果群集由多个负载均衡器提供服务，则必须以完全相同的方式在每个负载均衡器上配置群集。
- 每个会话都依赖于特定的群集。因此，尽管您可以在多个群集上部署一个应用程序，但是会话故障转移将只能在单个群集中实现。

因此，对于群集中的服务器实例，群集充当的是会话故障转移的安全边界。在 Application Server 中，您可以使用负载均衡器和升级组件，而不使服务受到任何损失。

从故障中恢复

- 第 27 页中的 “使用 Sun Cluster”
- 第 27 页中的 “手动恢复”
- 第 29 页中的 “使用 Netbackup”
- 第 30 页中的 “重新创建域管理服务器”

使用 Sun Cluster

Sun Cluster 提供域管理服务器、节点代理、Application Server 实例、Message Queue 和 HADB 的自动故障转移。有关更多信息，请参见《Sun Cluster Data Service for Sun Java System Application Server Guide for Solaris OS》。

使用标准以太网互联和 Sun Cluster 产品的子集。该功能包含在 Java ES 中。

手动恢复

您可以使用多种技术来手动恢复各个子组件：

- 第 27 页中的 “恢复域管理服务器”
- 第 28 页中的 “恢复节点代理和服务器实例”
- 第 28 页中的 “恢复负载均衡器和 Web 服务器”
- 第 28 页中的 “恢复 Message Queue”
- 第 29 页中的 “恢复 HADB”

恢复域管理服务器

丢失域管理服务器 (Domain Administration Server, DAS) 只会影响管理。即使 DAS 不可访问，Application Server 群集和应用程序也将继续像以前那样运行。

可使用以下任一方法恢复 DAS：

- 定期运行 `asadmin` 备份命令，以便获得定期快照。发生硬件故障后，在具有相同网络标识的新计算机上安装 App Server，然后运行 `asadmin` 从以前创建的备份中恢复。有关更多信息，请参见第 30 页中的 “重新创建域管理服务器”。

- 将域安装和配置放置在共享的强健文件系统（例如 NFS）上。如果主 DAS 计算机发生故障，则会初启具有相同 IP 地址的第二台计算机，并通过手动介入或用户提供的自动控制功能让该台计算机进行接管。Sun Cluster 使用类似的方法来实现 DAS 容错。
- 压缩 Application Server 安装和域根目录。在新计算机上恢复它，为其指定相同的网络标识。如果您使用的是基于文件的安装，这可能是最简单的方法。
- 从 DAS 备份中恢复。请参见 AS8.1 UR2 修补程序 4 的说明。

恢复节点代理和服务器实例

可采用两种方法来恢复节点代理和服务器实例。

保存备份 zip 文件。 没有用于备份节点代理和服务器实例的显式命令。只需创建一个包含节点代理目录的内容的 zip 文件。发生故障后，将保存的备份解压至具有相同主机名和 IP 地址的新计算机上。请使用相同的安装目录位置、OS 等。该计算机上必须有基于文件的安装、基于软件包的安装或恢复的备份映像。

手动恢复。 使用的新计算机必须具有相同的 IP 地址。

1. 在该计算机上安装 Application Server 节点代理位。
2. 请参见 AS8.1 UR2 修补程序 4 安装说明。
3. 重新创建节点代理。您无需创建任何服务器实例。
4. 同步过程将从 DAS 复制配置和数据，并进行更新。

恢复负载均衡器和 Web 服务器

没有专门用于备份 Web 服务器配置的显式命令。只需压缩 Web 服务器安装目录。发生故障后，将保存的备份解压至具有相同网络标识的新计算机上。如果新计算机具有不同的 IP 地址，则更新 DNS 服务器或路由器。

注 - 这里假定首先重新安装了 Web 服务器或从映像恢复了 Web 服务器。

负载均衡器插件（插件目录）和配置位于 Web 服务器安装目录（通常为 /opt/SUNWwbsvr）。web-install/web-instance/config 目录包含 loadbalancer.xml 文件。

恢复 Message Queue

Message Queue (MQ) 配置和资源存储在 DAS 中，可以将其与实例同步。任何其他数据和配置信息都位于 MQ 目录中，通常位于 /var/imq 下，所以可以根据需要备份和修复这些目录。新的计算机必须已经包含 MQ 安装。恢复计算机时，确保像以前一样启动 MQ 代理。

恢复 HADB

注 – HADB 软件随 Sun Java System Application Server 的 Application Server 独立分发提供。有关 Sun Java System Application Server 的可用分发的信息，请参见《Sun Java System Application Server 9.1 Installation Guide》中的“Distribution Types and Their Components”。HADB 功能仅在企业配置文件中可用。有关配置文件的信息，请参见《Sun Java System Application Server 9.1 管理指南》中的“用法配置文件”。

如果您具有两个活动 HADB 节点，则可以配置两个备用节点（在单独的计算机上），如果发生故障，它们可以接管。这是一种比较干净利落的方法，因为备份和恢复 HADB 可能会导致恢复过时的会话。

有关创建具有备用节点的数据库的信息，请参见第 62 页中的“创建数据库”。有关将备用节点添加到数据库的信息，请参见第 84 页中的“添加节点”。如果恢复和自我修复失败，则备用节点将自动接管。

使用 Netbackup

注 – 此过程尚未经 Sun QA 测试。

使用 Veritas Netbackup 可以保存每台计算机的映像。对于 BPIP，备份四台具有 Web 服务器和 Application Server 的计算机。

对于已经恢复的每台计算机，使用与原始计算机相同的配置，例如，相同的主机名、IP 地址等。

对于基于文件的产品（如 Application Server），仅备份和恢复相关的目录。但是，对于基于软件包的安装（如 Web 服务器映像），必须备份和恢复整个计算机。软件包安装到 Solaris 软件包数据库中。因此，如果仅备份目录，随后将其恢复至新系统中，将会造成“已部署”的 Web 服务器无法识别软件包数据库。这可能会给将来的修补和升级带来问题。

请勿手动复制和恢复 Solaris 软件包数据库，其他替代方法是在安装组件（例如，Web 服务器）后备份计算机的映像，这称为基准 tar 文件。对 Web 服务器进行更改时，备份这些目录（例如，在 /opt/SUNWwbsvr 下）。恢复时，先恢复基准 tar 文件，然后覆盖已修改的 Web 服务器目录。同样，可以对 MQ（BPIP 的基于软件包的安装）使用该过程。如果您升级或修补原始计算机，请确保创建一个新的基准 tar 文件。

如果具有 DAS 的计算机发生故障，则在恢复之前将无法使用 DAS。

DAS 是中心系统信息库。恢复服务器实例并重新启动它们时，这些实例只会与 DAS 中的信息同步。因此，必须通过 `asadmin` 或管理控制台进行所有更改。

每天备份 HADB 的映像可能并不可行，因为映像可能包含旧的应用程序会话状态。

重新创建域管理服务器

在托管域管理服务器 (domain administration server, DAS) 的计算机出现故障时，如果以前已备份 DAS，则可以重新创建 DAS。要重新创建 DAS 的工作副本，您必须具有：

- 一台包含原始 DAS 的计算机 (machine1)。
- 一台包含群集的计算机 (machine2)，该群集具有运行应用程序并满足客户机需要的服务器实例。该群集是使用第一台计算机上的 DAS 配置的。
- 一台备份计算机，当第一台计算机崩溃时，需要在该备份计算机上重新创建。

注 - 必须对第一台计算机上的 DAS 进行备份。使用 `asadmin backup-domain` 来备份当前域。

▼ 迁移 DAS

以下步骤用于将域管理服务器从第一台计算机 (machine1) 迁移到第三台计算机 (machine3)。

- 1 在第三台计算机上安装应用程序服务器，方法与在第一台计算机上安装相同。
为了可以在第三台计算机上正确地恢复 DAS 并且不会发生路径冲突，您必须执行此操作。
 - a. 使用命令行（交互式）模式来安装应用程序服务器管理软件包。
要激活交互式命令行模式，请使用 `console` 选项调用安装程序：

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


要使用命令行界面进行安装，您必须具有超级用户权限。
 - b. 要安装默认域，请取消选择该选项。
只有具有相同体系结构并具有完全相同的安装路径（即都使用相同的 `as-install` 和 `domain-root-dir`）的两台计算机才支持备份域的恢复。
- 2 将第一台计算机上的备份 ZIP 文件复制到第三台计算机上的 `domain-root-dir` 中。
也可以通过 FTP（文件传输协议）方式传输文件。
- 3 将 ZIP 文件恢复到第三台计算机上。

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

注 – 通过指定 `--clienthostname` 选项，就无需在 `domain.xml` 文件中修改 `jmx-connector` 元素的 `client-hostname` 属性。

可以备份任何域。但是，在重新创建域时，域名称应与原始域名称相同。

- 4 将第三台计算机上的 `domain-root-dir/domain1/generated/tmp` 目录的权限更改为与第一台计算机上相同目录的权限相匹配。

该目录的默认权限为：`drwx-----`（或 700）。

例如：

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

以上示例假定您备份的是 `domain1`。如果备份的是其他名称的域，应使用要备份的域的名称替换上面的 `domain1`。

- 5 在第三台计算机上的 `domain-root-dir/domain1/config/domain.xml` 文件中，更新 `jms-service` 元素的 `host` 属性值。

此属性的原始设置如下：

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

对此属性的设置进行如下修改：

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

- 6 在 `machine3` 上启动已恢复的域：

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

DAS 与正在运行的所有节点代理联系，并为节点代理提供用于联系 DAS 的信息。节点代理使用此信息与 DAS 通信。

- 7 对于重新启动 DAS 时未运行的任何节点代理，请在 `machine2` 上更改 `as-install/nodeagents/nodeagent/agent/config/das.properties` 中的 `agent.das.host` 属性值。

对于重新启动 DAS 时正在运行的节点代理，不需要执行此步骤。

- 8 在 `machine2` 上重新启动节点代理。

注 – 使用 `asadmin start-instance` 命令启动群集实例，可以使这些实例与已恢复的域同步。

安装和设置高可用性数据库

注 - HADB 软件随 Sun Java System Application Server 的 Application Server 独立分发提供。有关 Sun Java System Application Server 的可用分发的信息，请参见《Sun Java System Application Server 9.1 Installation Guide》中的“Distribution Types and Their Components”。HADB 功能仅在企业配置文件中可用。有关配置文件的信息，请参见《Sun Java System Application Server 9.1 管理指南》中的“用法配置文件”。

本节包含以下主题：

- 第 33 页中的 “设置 HADB 前的准备工作”
- 第 40 页中的 “安装”
- 第 42 页中的 “设置高可用性”
- 第 44 页中的 “升级 HADB”

设置 HADB 前的准备工作

本节包括以下主题：

- 第 34 页中的 “先决条件和限制”
- 第 34 页中的 “配置网络冗余”
- 第 37 页中的 “配置共享内存和信号”
- 第 39 页中的 “同步系统时钟”

执行完这些任务后，请参见第 3 章。

有关 HADB 的最新信息，请参见《Sun Java System Application Server 9.1 发行说明》。

先决条件和限制

在设置和配置 HADB 之前，请确保您的网络和硬件环境符合《Sun Java System Application Server 9.1 发行说明》中所述的要求。此外，对某些文件系统还有一些限制，例如，对 Veritas。有关更多信息，请参见发行说明。

HADB 在创建并附加到其共享内存段时，使用锁定共享内存（SHM_SHARE_MMU 标志）。使用此标志实质上是将共享内存段锁定到物理内存中，防止它们被调出。因此，HADB 的共享内存已锁定到物理内存，这很容易影响到低端计算机上的安装。请确保在协同定位 Application Server 和 HADB 时拥有建议的内存量。

配置网络冗余

配置冗余网络后，即使单个网络出现故障，HADB 也仍然可用。您可以使用以下两种方法配置冗余网络：

- 在 Solaris 9 中，设置网络多路径。
- 在除 Windows Server 2003 以外的所有平台中，配置双网络。

设置网络多路径

设置网络多路径之前，请参阅《IP Network Multipathing Administration Guide》中的 "Administering Network Multipathing" 一节。

▼ 配置已使用 IP 多路径的 HADB 主机

1 设置网络接口故障检测时间。

为了使 HADB 能够正确支持多路径故障转移，网络接口故障检测时间不能超过一秒（1000 毫秒），此时间由 `/etc/default/mpathd` 中的 `FAILURE_DETECTION_TIME` 参数来指定。如果此参数的原始值较高，请编辑文件并将此参数值更改为 1000：

```
FAILURE_DETECTION_TIME=1000
```

要使所作的更改生效，请使用以下命令：

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

2 设置 IP 地址以用于 HADB。

如《IP Network Multipathing Administration Guide》中所述，多路径功能涉及将物理网络接口分组为多路径接口组。在此类组中，每个物理接口均包含两个与其关联的 IP 地址：

- 用于传输数据的物理接口地址。
- 仅供 Solaris 内部使用的测试地址。

当您使用 `hadbm create --hosts` 时，请仅从多路径组指定一个物理接口地址。

示例 2-1 设置多路经

假设具有两个名为 `host1` 和 `host2` 的主机。如果它们分别具有两个物理网络接口，则将这两个接口作为一个多路径组进行设置。在每个主机上运行 `ifconfig -a`。

`host1` 上的输出为：

```
bge0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4>
mtu 1500 index 5 inet 129.159.115.10 netmask fffffff0 broadcast 129.159.115.255
groupname mp0

bge0:1: flags=9040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4,NOFAILOVER>
mtu 1500 index 5 inet 129.159.115.11 netmask fffffff0 broadcast 129.159.115.255

bge1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4>
mtu 1500 index 6 inet 129.159.115.12 netmask fffffff0 broadcast 129.159.115.255
groupname mp0

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

`host2` 上的输出为：

```
bge0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4>
mtu 1500 index 3 inet 129.159.115.20 netmask fffffff0 broadcast 129.159.115.255
groupname mp0

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

bge1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4>
mtu 1500 index 4 inet 129.159.115.22 netmask fffffff0 broadcast 129.159.115.255
groupname mp0

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

在此示例中，两台主机的物理网络接口都在 `bge0` 和 `bge1` 之后列出。在 `bge0:1` 和 `bge1:1` 之后列出的接口是多路径测试接口（它们在 `ifconfig` 输出中标记为 `DEPRECATED`），如《IP Network Multipathing Administration Guide》中所述。

要在此环境中设置 HADB，请从每个节点选择一个物理接口地址。在此示例中，HADB 在 `host1` 中使用 IP 地址 129.159.115.10，在 `host2` 中使用 129.159.115.20。要创建数据库以实现每台主机具有一个数据库节点，请使用命令 `hadbm create --hosts`。例如

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

要在每台主机上使用两个数据库节点创建数据库，请使用命令：

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

在这两种情况下，必须使用单独的参数配置 host1 和 host2 上的代理，以指定代理所应使用的计算机的接口。因此，在 host1 上使用以下命令：

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

在 host2 上使用以下命令：

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

有关 `ma.server.mainternal.interfaces` 变量的信息，请参见第 56 页中的“配置文件”。

配置双网络

要启用 HADB 以允许单个网络故障，请使用 IP 多路径（如果操作系统 [例如 Solaris] 支持）。请勿在 Windows Server 2003 上将 HADB 配置为使用双网络—此操作系统在双网络配置下无法正常工作。

如果您没有将操作系统配置为使用 IP 多路径，并且已为 HADB 主机配备两个 NIC，则可以将 HADB 配置为使用双网络。对于每台主机，每个网络接口卡 (network interface card, NIC) 的 IP 地址必须位于单独的 IP 子网中。

在数据库中，所有节点均必须连接到单个网络，或者所有节点均必须连接到两个网络。

注 – 必须将子网之间的路由器配置为可以在子网之间转发 UDP 多址广播消息。

创建 HADB 数据库时，请使用 `-hosts` 选项来指定每个节点的两个 IP 地址或主机名：每个 NIC IP 地址一个。对于每个节点，第一个 IP 地址位于 `net-0`，第二个位于 `net-1`。语法如下所示，其中同一节点的主机名使用加号 (+) 分隔：

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

例如，以下变量可创建两个节点，每个节点具有两个网络接口。以下主机选项用于创建这些节点：

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

因此，网络地址如下

- 对于 node0，为 10.10.116.61 和 10.10.124.61

- 对于 node1，为 10.10.116.62 和 10.10.124.62

请注意，10.10.116.61 和 10.10.116.62 位于同一个子网；10.10.124.61 和 10.10.124.62 位于同一个子网。

在此示例中，管理代理必须使用同一子网。因此，以配置变量 `ma.server.mainternal.interfaces` 为例，它必须设置为 `10.10.116.0/24`。此设置对本示例中的两个代理均可使用。

配置共享内存和信号

安装 HADB 之前，必须配置共享内存和信号。此过程取决于您的操作系统。

如果在主机上运行除 HADB 以外的应用程序，则应计算这些应用程序使用的共享内存和信号量，并将其与 HADB 所需的值相加。本节所建议的值足够用于在每台主机上运行多达六个 HADB 节点。如果运行的 HADB 节点超过六个，或主机运行的应用程序需要更多的共享内存和信号量，则只需要增大这些值。

如果信号量的数目过低，则 HADB 可能会失败，并显示以下错误消息：`No space left on device`。在启动数据库时或在运行时可能会出现这种情况。

▼ 在 Solaris 上配置共享内存和信号

因为信号量是全局操作系统资源，所以配置将会与主机上运行的所有进程相关，而不是只与 HADB 相关。在 Solaris 中，通过编辑 `/etc/system` 文件来配置信号量设置。

1 以 root 用户身份登录。

2 配置共享内存。

- 设置 `shminfo_shmmax`，该参数指定主机上单个共享内存段的最大大小。将此值设置为 HADB 主机上安装的 RAM 的总容量，以十六进制值表示，但不能大于 2 GB。

例如，对于 2 GB RAM，在 `/etc/system` 文件中将该值设置如下：

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

注-要确定主机的内存，请使用以下命令：

```
prtconf | grep Memory
```

- 在 Solaris 8 或更低版本中，设置 `shminfo_shmseg`，即一个进程最多可以附加的共享内存段数。将该值设置为每台主机节点数的六倍。对于每台主机最多有六个节点的情况，请向 `/etc/system` 文件中添加以下内容：

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

在 Solaris 9 和更高版本上，shmsys:shminfo_shmseg 是过时的属性。

- 设置 shminfo_shmmni，即整个系统中共享内存段的最大数目。因为每个 HADB 节点分配六个共享内存段，所以 HADB 所需的值至少必须为每台主机节点数的六倍。在 Solaris 9 中，对于每台主机最多有六个节点的情况，不需要更改默认值。

3 配置信号。

检查 /etc/system 文件是否包含以下信号量配置条目，例如：

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

如果存在这些条目，请按如下所示增大这些值。

如果 /etc/system 文件不包含这些条目，请将它们添加到该文件的结尾：

- 设置 seminfo_semmni，即信号量标识符最大数目。每个 HADB 节点需要一个信号量标识符。在 Solaris 9 中，对于每台主机最多有六个节点的情况，不需要更改默认值。例如：

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

- 设置 seminfo_semmns，即整个系统中信号量的最大数目。每个 HADB 节点需要八个信号量。在 Solaris 9 中，对于每台主机最多有六个节点的情况，不需要更改默认值。例如：

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

- 设置 seminfo_semmnu，即系统中撤消结构的最大数目。每个连接需要一个撤消结构（配置变量 NumberOfSessions，默认值 100）。对于每台主机最多有六个节点的情况，将该值设置为 600：

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

4 重新引导计算机。

▼ 在 Linux 上配置共享内存

在 Linux 中，必须配置共享内存设置。不需要调整默认信号量设置。

1 以 root 用户身份登录。

2 编辑文件 /etc/sysctl.conf。

在 Redhat Linux 中，您还可以修改 sysctl.conf 以设置内核参数。

3 按如下所示，设置 kernel.shmax 和 kernel.shmall 的值：

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

其中，*MemSize* 为字节数。

`kernel.shmax` 参数定义了共享内存段的最大大小（以字节为单位）。`kernel.shmall` 参数设置了在系统上可以一次使用的共享内存的总数（以页为单位）。将这两个参数的值设为计算机上的物理内存量。请以十进制字节数指定该值。

例如，要将两个值都设置为 2GB，请使用以下命令：

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

4 使用以下命令重新引导该计算机：

sync; sync; reboot

适用于 Windows 的过程

Windows 并不需要任何特殊的系统设置。但是，如果要使用现有 J2SE 安装，请将 `JAVA_HOME` 环境变量设置为安装 J2SE 所在的位置。

同步系统时钟

必须使 HADB 主机上的时钟同步，因为 HADB 使用基于系统时钟的时间戳。HADB 使用系统时钟管理超时并为记录到历史文件的事件标上时间戳。由于 HADB 是分布式系统，因此对于故障排除，您必须对所有历史文件一起进行分析。所以，使所有主机的时钟同步非常重要。

请勿在 HADB 系统处于运行状态时调整其系统时钟。这样做会导致操作系统或其他软件组件出现问题，而这些问题反之会导致系统出现诸如 HADB 节点挂起或重新启动等问题。将时钟调慢可能导致某些 HADB 服务器进程因时钟的调整而挂起。

要使时钟同步：

- 在 Solaris 中，请使用 `xntpd`（网络时间协议守护进程）。
- 在 Linux 中，请使用 `ntpd`。
- 在 Windows 中，请使用 Windows 中的 `NTPTIME`。

如果 HADB 检测到时钟调整超过一秒钟，则将该数据记录到节点历史文件中，例如：

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

安装

通常，可以在与 Application Server（同机拓扑）相同的系统上或单独的主机（单独层拓扑）上安装 HADB。有关这两个选项的更多信息，请参见《Sun Java System Application Server 9.1 部署规划指南》中的第 3 章“选择拓扑”。

必须安装 HADB 管理客户机，才能使用 `asadmin configure-ha-cluster` 命令设置高可用性。使用 Java Enterprise System 安装程序时，即使要将节点安装在单独层上，也必须安装完整的 HADB 实例以便安装管理客户机。

HADB 安装

在单或双 CPU 系统上，如果该系统至少有 2 GB 的内存，则可以同时安装 HADB 和 Application Server。否则，请在单独的系统上安装 HADB 或使用其他硬件。要使用 `asadmin configure-ha-cluster` 命令，必须同时安装有 HADB 和 Application Server。

每个 HADB 节点需要 512 MB 的内存，因此计算机需要 1 GB 的内存来运行两个 HADB 节点。如果计算机内存不足，请在不同的计算机上设置每个节点。例如，您可以在以下系统上安装两个节点：

- 两个单 CPU 系统，每个系统具有 512 MB 到 1 GB 内存
- 一个单或双 CPU 系统，具有 1 GB 到 2 GB 内存

您可以使用 Java Enterprise System 安装程序或 Application Server 独立安装程序安装 HADB。在上述任一安装程序中，请在“组件选项”页面中选择用于安装 HADB（在 Java ES 中称为高可用性会话存储）的选项。在您的主机上完成安装。如果您使用 Application Server 独立安装程序，并选择两个单独的计算机运行 HADB，则必须在两台计算机上选择相同的安装目录。

默认安装目录

本手册中，`HADB_install_dir` 表示安装 HADB 的目录。默认的安装目录将取决于您是否将 HADB 作为 Java Enterprise System 的一部分安装。对于 Java Enterprise System，默认安装目录为 `/opt/SUNWhadb/4`。对于独立的 Application Server 安装程序，安装目录为 `/opt/SUNWappserver/hadb/4`。

节点监控进程权限

节点监控进程 (node supervisor processes, NSUP) 通过彼此交换 "I'm alive" 消息来确保 HADB 的可用性。NSUP 可执行文件必须具有 root 用户权限才能尽可能快地作出响应。`clu_nsup_srv` 进程不使用重要的 CPU 资源，它所占用的资源少，因此使用实时优先级运行该进程并不影响性能。

注 – Java Enterprise System 安装程序会自动正确地设置 NSUP 权限，因此您不必采取进一步操作。但是，使用独立的 Application Server（非 root 用户）安装程序，您必须在创建数据库之前手动设置权限。

权限不足时出现的情况

如果 NSUP 不具有正确的权限，您可能会注意到以下资源不足时出现的问题，例如：

- 出现网络分区错误和节点重新启动之前，HADB 历史文件中显示 "Process blocked for *n* seconds" 警告消息。
- 异常中止的事务和其他异常。

限制

如果 NSUP 无法设置实时优先级，则 `errno` 在 Solaris 和 Linux 中被设为 `EPERM`。在 Windows 中，它将发出“无法设置实时优先级”的警告。系统将错误写入 `ma.log` 文件，并且进程在没有实时优先级的情况下继续进行。

在以下情况下无法设置实时优先级：

- HADB 安装在 Solaris 10 非全局区域中
- 在 Solaris 10 中取消 `PRIV_PROC_LOCK_MEMORY`（允许进程锁定物理内存中的页）和/或 `PRIV_PROC_PRIOCNTL` 权限
- 用户关闭 `setuid` 权限
- 用户将软件作为 tar 文件（Application Server 的非 root 用户安装选项）安装

▼ 授予节点监控进程 root 用户权限

- 1 以 root 用户身份登录。
- 2 将您的工作目录更改为 `HADB_install_dir/lib/server`。
NSUP 可执行文件为 `clu_nsup_srv`。
- 3 使用以下命令设置文件的 `suid` 位：

```
chmod u+s clu_nsup_srv
```
- 4 使用以下命令将文件的所有权设置为 root 用户：

```
chown root clu_nsup_srv
```

这将以 root 用户身份启动 `clu_nsup_srv` 进程，并使该进程授予其自身实时优先级。

为了避免任何安全性影响，启动进程后将立即设置实时优先级，并且一旦优先级更改，该进程将退回至有效 UID。其他 HADB 进程使用普通优先级运行。

设置高可用性

本节提供了创建高可用性群集和测试 HTTP 会话持久性的步骤。

本节包括以下主题：

- 第 42 页中的 “准备高可用性系统”
- 第 42 页中的 “启动 HADB 管理代理”
- 第 43 页中的 “配置高可用性群集”
- 第 43 页中的 “配置高可用性应用程序”
- 第 43 页中的 “重新启动群集”
- 第 44 页中的 “重新启动 Web Server”
- 第 44 页中的 “清除作为负载均衡器使用的 Web Server 实例”

▼ 准备高可用性系统

1 安装 Application Server 实例和负载均衡器插件。

有关更多信息，请参见 *Java Enterprise System 安装指南*（如果使用 Java ES）或《Sun Java System Application Server 9.1 Installation Guide》（如果使用独立的 Application Server 安装程序）。

2 创建 Application Server 域和群集。

有关如何创建域的信息，请参见《Sun Java System Application Server 9.1 管理指南》中的“创建域”。有关如何创建群集的信息，请参见第 151 页中的“创建群集”。

3 安装和配置 Web Server 软件。

4 设置和配置负载均衡。

有关更多信息，请参见第 122 页中的“设置 HTTP 负载均衡”。

启动 HADB 管理代理

管理代理 ma 用于在 HADB 主机上执行管理命令，并在 HADB 节点监控进程失败时通过重新启动它们来确保其可用性。

您可以使用两种方式启动管理代理：

- 作为服务启动，用于生产。请参见第 50 页中的“将管理代理作为服务启动”。要确保管理代理的可用性，请确保它会在系统重新引导时自动重新启动。请参见第 51 页中的“确保管理代理自动重新启动”。
- 作为常规进程（在控制台模式下）启动，用于评估、测试或开发。请参见第 53 页中的“在控制台模式下启动管理代理”。

对于每种启动方式，启动过程会因使用的是 Java Enterprise System 还是独立的 Application Server 而有所不同。

配置高可用性群集

开始本节之前，您必须已经创建一个或多个 Application Server 群集。有关如何创建群集的信息，请参见第 151 页中的“创建群集”。

使用以下命令，从运行域管理服务器的计算机将群集配置为使用 HADB：

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

将 *hadb_hostname1*、*hadb_hostname2* 等替换为运行 HADB 的每台计算机的主机名，并将 *clusterName* 替换为群集的名称。例如：

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

此示例会在每台计算机上创建两个节点，即使在 HADB 故障转移时这些节点也具有很高的可用性。请注意，`-hosts` 选项后主机名的顺序非常重要，因此上一示例会与 `--hosts host1,host1,host2,host2` 不同。

如果仅使用一台计算机，则必须提供两次主机名。在生产设置中，建议使用多台计算机。

配置高可用性应用程序

在管理控制台中，在“应用程序” > “企业应用程序”下选择应用程序。设置“启用可用性”，然后单击“保存”。

重新启动群集

要在管理控制台中重新启动群集，请选择“群集” > *cluster-name*。单击“停止实例”。当实例都停止后，单击“启动实例”。

或者，使用以下 `asadmin` 命令：

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

有关这些命令的更多信息，请参见 `stop-cluster(1)` 和 `start-cluster(1)`。

重新启动 Web Server

要重新启动 Web Server，请键入以下 Web Server 命令：

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

将 *web_server_root* 替换为您的 Web Server 根目录，并将 *hostname* 替换为您的主机名。

▼ 清除作为负载均衡器使用的 Web Server 实例

- 1 删除负载均衡器配置：

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

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

- 2 如果您创建了新的 Web Server 实例，则您可以通过以下方法删除它：

- a. 登录到 Web Server 的管理控制台。

- b. 停止该实例。

删除该实例。

升级 HADB

HADB 旨在通过升级本软件提供不间断的 "always on" 服务。本节介绍了在不使数据库脱机或不引起任何可用性损失的情况下，如何升级到 HADB 的新版本。这称为**联机升级**。

以下各节将介绍如何升级 HADB 安装：

- 第 45 页中的“将 HADB 升级到更新版本”
- 第 45 页中的“注册 HADB 软件包”
- 第 46 页中的“取消注册 HADB 软件包”
- 第 47 页中的“替换管理代理的启动脚本”
- 第 47 页中的“验证 HADB 升级”

▼ 将 HADB 升级到更新版本

- 1 安装 HADB 的新版本。
- 2 按照第 45 页中的“注册 HADB 软件包”中所述，注册新的 HADB 版本。
在 HADB 管理域中注册 HADB 软件包使得升级或更改 HADB 软件包变得更加容易。管理代理记录软件包的位置以及域中主机的版本信息。默认软件包的名称是以 V 开头的字符串，其中包括 hadbm 程序的版本号。
- 3 更改数据库所使用的软件包。
请输入以下命令：

```
hadbm set PackageName=package
```

其中，*package* 是新的 HADB 软件包的版本号。
- 4 按照第 46 页中的“取消注册 HADB 软件包”中所述，注销您现有的 HADB 安装。
- 5 如果需要，请替换管理代理的启动脚本。
有关更多信息，请参见第 47 页中的“替换管理代理的启动脚本”。
- 6 按照第 47 页中的“验证 HADB 升级”中所述，验证结果。
- 7 （可选的）删除旧 HADB 版本的二进制文件。
验证 HADB 已正确升级后，可以删除旧的 HADB 软件包。

注册 HADB 软件包

使用 `hadbm registerpackage` 命令注册 HADB 软件包，该软件包安装在管理域中的主机上。也可在使用 `hadbm create` 命令创建数据库时注册 HADB 软件包。

使用 `hadm registerpackage` 命令之前，请确保满足以下要求：已配置所有管理代理并且它们在主机列表中的所有主机上运行；管理代理的系统信息库可以更新，并且没有使用相同的软件包名称注册的软件包。

该命令语法为：

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

package-name 操作数是软件包的名称。

下表介绍了特殊的 `hadbm registerpackage` 命令选项。有关其他命令选项的说明，请参见第 58 页中的“安全性选项”和第 59 页中的“常规选项”。

表 2-1 hadbm registerpackage 选项

选项	说明
--hosts= <i>hostlist</i>	主机列表，用逗号分隔或放在双引号内并用空格分隔。
-H	
--packagepath= <i>path</i>	HADB 软件包的路径。
-L	

例如，使用以下命令可在主机 *host1*、*host2* 和 *host3* 上注册软件包 *v4*：

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

响应为：

软件包注册成功。

如果您省略 `--hosts` 选项，则此命令将在域中所有已启用的主机上注册该软件包。

取消注册 HADB 软件包

使用 `hadbm unregisterpackage` 命令删除用管理域注册的 HADB 软件包。

使用 `hadbm unregisterpackage` 命令之前，请确保：

- 已配置所有管理代理，并且它们在 *hostlist* 中的所有主机上运行。
- 管理代理的系统信息库可以更新。
- 新的 HADB 软件包已在管理域中注册。
- 未将任何现有数据库配置为在将要注销的软件包中运行。

该命令语法为：

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

package-name 操作数是软件包的名称。

有关 `--hosts` 选项的说明，请参见第 45 页中的“注册 HADB 软件包”。如果您省略 `--hosts` 选项，主机列表将默认为已启用的注册了软件包的主机。有关其他命令选项的说明，请参见第 58 页中的“安全性选项”和第 59 页中的“常规选项”。

示例 2-2 注销 HADB 的示例

从域中特定主机上注销软件包 v4 :

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

响应为 :

软件包注销成功。

替换管理代理的启动脚本

安装 HADB 的新版本时，您可能需要替换 `/etc/init.d/ma-initd` 中的管理代理启动脚本。检查文件 `HADB_install_dir/lib/ma-initd` 的内容。如果该文件不同于旧的 `ma-initd` 文件，请将旧文件替换为新文件。

▼ 验证 HADB 升级

执行以下过程以验证 HADB 已正确升级 :

- 1 确认正在运行的 HADB 进程的版本。

在所有 HADB 节点上输入以下命令以显示 HADB 版本 :

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

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

其中，*new-path* 是新 HADB 的安装路径。

该结果应该显示新的 HADB 版本号。

- 2 确认数据库正在运行。

输入以下命令 :

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

如果升级成功，则结果将显示所有处于 `running` 状态的 HADB 节点。

- 3 确保使用 HADB 的产品已将其配置设置更改为新的 HADB 路径。
- 4 为使用 HADB 的产品运行所有升级测试。

管理高可用性数据库

注 – HADB 软件随 Sun Java System Application Server 的 Application Server 独立分发提供。有关 Sun Java System Application Server 的可用分发的信息，请参见《Sun Java System Application Server 9.1 Installation Guide》中的“Distribution Types and Their Components”。HADB 功能仅在企业配置文件中可用。有关配置文件的信息，请参见《Sun Java System Application Server 9.1 管理指南》中的“用法配置文件”。

本章介绍了 Sun Java System Application Server 环境中的高可用性数据库 (High Availability Database, HADB)。其中介绍了如何配置和管理 HADB。在创建和管理 HADB 之前，必须首先确定系统的拓扑并在各种计算机上安装 HADB 软件。

本章讨论以下主题：

- 第 49 页中的 “使用 HADB 管理代理”
- 第 57 页中的 “使用 hadbm 管理命令”
- 第 61 页中的 “配置 HADB”
- 第 74 页中的 “管理 HADB”
- 第 83 页中的 “扩展 HADB”
- 第 88 页中的 “监视 HADB”
- 第 94 页中的 “维护 HADB 计算机”

使用 HADB 管理代理

管理代理 `ma`，在 HADB 主机上执行管理命令。管理代理还通过重新启动 HADB 节点监控进程（如果失败）来确保其可用性。

- 第 50 页中的 “启动管理代理”
- 第 54 页中的 “管理代理命令语法”
- 第 55 页中的 “自定义管理代理配置”

启动管理代理

您可以使用以下方式启动管理代理：

- 作为服务启动，用于生产。请参见第 50 页中的“将管理代理作为服务启动”。为确保管理代理的可用性，请确保系统重新引导时管理代理可以自动重新启动。请参见第 51 页中的“确保管理代理自动重新启动”。
- 作为常规进程（在控制台模式下）启动，用于评估、测试或开发。请参见第 53 页中的“在控制台模式下启动管理代理”。
- 使用 Solaris 10 上的服务管理工具 (service management facility, SMF) 启动。请参见第 54 页中的“使用 Solaris 10 服务管理工具运行管理代理”。

对于每种启动方式，启动过程会因使用的是 Java Enterprise System 还是独立的 Application Server 而有所不同。

将管理代理作为服务启动

将管理代理作为服务启动，可确保它继续运行直到系统关闭或管理代理被明确停止。命令取决于您的安装和平台：

- 第 50 页中的“Solaris 或 Linux 上的 Java Enterprise System”
- 第 50 页中的“Windows 上的 Java Enterprise System”
- 第 51 页中的“Solaris 或 Linux 上的独立 Application Server”
- 第 51 页中的“Windows 上的独立 Application Server”

Solaris 或 Linux 上的 Java Enterprise System

要将管理代理作为服务启动，请使用以下命令：

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

要停止该服务，请使用以下命令：

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

Windows 上的 Java Enterprise System

要将管理代理作为 Windows 服务启动，请使用以下命令：`HADB_install_dir\bin\ma -i [config-file]`

可选参数 `config-file` 指定了管理代理配置文件。仅当您希望更改默认管理代理配置时，才使用配置文件。有关更多信息，请参见第 55 页中的“自定义管理代理配置”。

要将管理代理（作为服务）停止并删除（注销），请使用以下命令：`HADB_install_dir\bin\ma -r [config-file]`

要执行管理，请选择“管理工具”|“服务”，以便能够启动和停止该服务、禁用自动启动等等。

Solaris 或 Linux 上的独立 Application Server

要将管理代理作为服务启动，请使用以下命令：

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

要停止该服务，请使用以下命令：

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

要更改默认值，请编辑 shell 脚本 `HADB_install_dir/bin/ma-initd`。将 `ma-initd` 复制到目录 `/etc/init.d`。替换脚本中 `HADB_ROOT` 和 `HADB_MA_CFG` 的默认值以对应于您的安装：

- `HADB_ROOT` 为 HADB 安装目录 `HADB_install_dir`。
- `HADB_MA_CFG` 为管理代理配置文件的位置。有关更多信息，请参见第 55 页中的“自定义管理代理配置”。

Windows 上的独立 Application Server

要将管理代理作为 Windows 服务启动，请使用以下命令：`HADB_install_dir\bin\ma -i [config-file]`

可选参数 `config-file` 指定了管理代理配置文件。仅当您希望更改默认管理代理配置时，才使用配置文件。

要将管理代理（作为服务）停止并删除（注销），请使用以下命令：`HADB_install_dir\bin\ma -r [config-file]`

要执行管理，请选择“管理工具”|“服务”，以便能够启动和停止该服务、禁用自动启动等等。

确保管理代理自动重新启动

在生产部署中，将管理代理配置为自动重新启动。这样可确保在 `ma` 进程失败或操作系统重新引导时管理代理的可用性。

在 Windows 平台上，将管理代理作为服务启动后，您可以使用 Windows 管理工具将服务的启动类型设置为“自动”，然后设置所需的恢复选项。

在 Solaris 和 Linux 平台上，使用本节中的过程可配置管理代理自动重新启动。这些过程确保管理代理仅在系统进入以下两个运行级时才启动：

- 在 Solaris 上，运行级 3（默认值）。
- 在 RedHat Linux 上，运行级 5（图形模式的默认值）。

进入其他运行级将停止管理代理。

▼ 在 Solaris 或 Linux 上使用 Java Enterprise System 配置自动重新启动

开始之前 本节假定您对操作系统初始化和运行级有基本的了解。有关这些主题的信息，请参见您的操作系统文档。

1 请确保您系统的默认运行级为 3 或 5。

要查看系统的默认运行级，请检查文件 `/etc/inittab`，并在靠近顶部的位置，查找类似于如下所示的行：

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

本示例显示了默认运行级 5。

2 创建指向文件 `/etc/init.d/ma-initd` 的软链接，如第 52 页中的“创建软链接”中所述。

3 重新引导计算机。

接下来的操作 取消激活自动启动和停止代理、删除链接或将链接名称中的字母 K 和 S 更改为小写。

▼ 在 Solaris 或 Linux 上使用独立 Application Server 配置自动重新启动

1 在 shell 中，将当前目录更改为 `HADB_install_dir/bin`。

2 编辑 shell 脚本 `ma-initd`。

确保脚本中 `HADB_ROOT` 和 `HADB_MA_CFG` 的默认值反映您的安装：

- `HADB_ROOT` 为 HADB 安装目录 `HADB_install_dir`。
- `HADB_MA_CFG` 为管理代理配置文件的位置。有关更多信息，请参见第 55 页中的“自定义管理代理配置”。

3 将 `ma-initd` 复制到目录 `/etc/init.d`。

4 创建指向文件 `/etc/init.d/ma-initd` 的软链接，如第 52 页中的“创建软链接”中所述。

接下来的操作 取消激活自动启动和停止代理、删除链接或将链接名称中的字母 K 和 S 更改为小写。

创建软链接

在 Solaris 上，创建以下软链接：

```
/etc/rc0.d/K20ma-initd  
/etc/rc1.d/K20ma-initd  
/etc/rc2.d/K20ma-initd
```

```

/etc/rc3.d/S99ma-initd
/etc/rc5.d/K20ma-initd (only for Sun 4m and 4u architecture)
/etc/rc6.d/K20ma-initd
/etc/rc5.d/K20ma-initd

```

在 Linux 上，创建以下软链接：

```

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

```

在控制台模式下启动管理代理

您可能希望在控制台模式下启动管理代理以便进行评估或测试。请勿在生产环境中以这种方式启动管理代理，因为系统或进程失败后 `ma` 进程将不会重新启动而且命令窗口关闭时该进程将终止。命令取决于您的平台和安装：

- 第 53 页中的“Solaris 或 Linux 上的 Java Enterprise System”
- 第 53 页中的“Windows 上的 Java Enterprise System”
- 第 54 页中的“Windows 上的独立 Application Server”
- 第 54 页中的“Solaris 或 Linux 上的独立 Application Server”

Solaris 或 Linux 上的 Java Enterprise System

要在控制台模式下启动 HADB 管理代理，请使用以下命令：

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

管理代理的默认配置文件为 `/etc/opt/SUNWhadb/mgt.cfg`。

要停止管理代理，请中止进程或关闭 shell 窗口。

Windows 上的 Java Enterprise System

要在控制台模式下启动管理代理，请使用以下命令：

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

可选参数 `config-file` 为管理代理配置文件的名称。有关配置文件的更多信息，请参见第 55 页中的“自定义管理代理配置”。

要停止代理，请中止进程。

Windows 上的独立 Application Server

要在控制台模式下启动管理代理，请使用以下命令：

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

可选参数 *config-file* 为管理代理配置文件的名称；有关更多信息，请参见第 55 页中的“自定义管理代理配置”。

要停止管理代理，请中止进程。

Solaris 或 Linux 上的独立 Application Server

要在控制台模式下启动 HADB 管理代理，请使用以下命令：

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

管理代理的默认配置文件为 *HADB_install_dir/bin/ma.cfg*。

要停止管理代理，请中止进程或关闭 shell 窗口。

使用 Solaris 10 服务管理工具运行管理代理

在 Solaris 10 上，服务管理工具 (Service Management Facility, SMF) 提供了重新启动、查看和管理服务的机制。您可以使用 SMF 启动、重新启动和管理 HADB 管理代理。

管理代理的故障管理资源标识符 (fault management resource identifier, FMRI) 为 `svc:/application/hadb-ma`。

管理代理命令语法

管理代理 `ma` 命令的语法为：

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

其中：

- *common-options* 是第 54 页中的“管理代理命令语法”中介绍的一个或多个通用选项。
- *service-options* 是第 54 页中的“管理代理命令语法”中介绍的 Windows 服务选项之一。
- *config-file* 是管理代理配置文件的完整路径。有关更多信息，请参见第 55 页中的“自定义管理代理配置”。

表 3-1 管理代理通用选项

选项	说明	默认值
--define <i>name=value-D</i>	将 <i>value</i> 指定给属性 <i>name</i> ，其中属性为第 56 页中的“配置文件”中定义的属性之一。此选项可以重复多次。	无
--help-?	显示帮助信息。	False
--javahome <i>path-j</i>	使用位于 <i>path</i> 的 Java 运行时环境（1.4 版或更高版本）。	无
--systemroot <i>path-y</i>	通常设置在 %SystemRoot% 中的操作系统根目录的路径。	无
--version-V	显示版本信息。	False

表 3-2 介绍了用于将管理代理作为 Windows 服务启动的选项。-i、-r 和 -s 选项相互排斥，即每次只能使用其中一个选项。

在 Windows 上为配置文件或命令行中的属性值指定路径时，使用双引号 (") 为包含空格的文件路径进行转义。使用双引号和反斜杠："\: 和 \" 为驱动器 and 目录分隔符 (: 和 \) 进行转义。

表 3-2 管理代理服务选项（仅限于 Windows）

选项	说明	默认值
--install-i	将代理作为 Windows 服务来安装并启动该服务。仅使用 -i、-r 和 -s 选项之一。	False
--name <i>servicename-n</i>	在主机上运行多个代理时，将指定名称用于服务。	HADBMgrAgent
--remove-r	停止服务并从 Windows 服务管理器中删除代理。仅使用 -i、-r 和 -s 选项之一。	False
--service-s	将代理作为 Windows 服务来运行。仅使用 -i、-r 和 -s 选项之一。	False

自定义管理代理配置

HADB 包含可用于自定义管理代理设置的配置文件。未指定配置文件而启动管理代理时，将使用默认值。如果指定了配置文件，管理代理将使用该文件中的设置。您可以在域中所有主机上重复使用该配置文件。

▼ 在 HADB 主机上自定义管理代理配置

- 1 编辑管理代理配置文件并设置所需的值。
- 2 启动管理代理（将自定义的配置文件指定为参数）。

配置文件

使用 Java Enterprise System 时，配置文件中的所有条目都被注释掉。使用默认配置时无需任何更改。要自定义管理代理配置，请从文件删除注释，并按需要更改值，然后启动将配置文件指定为参数的管理代理。

管理代理配置文件安装在：

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

使用独立安装程序将管理代理配置文件安装在：

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

下表介绍了配置文件中的设置。

表 3-3 配置文件设置

设置名称	说明	默认值
console.loglevel	控制台的日志级别。有效值为 SEVERE、ERROR、WARNING、INFO、FINE、FINER、FINEST	WARNING
logfile.loglevel	日志文件的日志级别。有效值为 SEVERE、ERROR、WARNING、INFO、FINE、FINER、FINEST	INFO
logfile.name	日志文件的名称和位置。必须为具有读取/写入权限的有效路径。	Solaris 和 Linux：/var/opt/SUNWhadb/ma.log Windows：HADB_install_dir\ma.log
ma.server.type	客户机协议。仅支持 JMXMP。	jmxmp
ma.server.jmxmp.port	用于内部 (UDP) 和外部 (TCP) 通信的端口号。必须为正整数。建议范围为 1024-49151。	1862
ma.server.mainternal.interfaces	具有多个接口的计算机的内部通信接口。必须为有效的 IPv4 地址掩码。同一个域的所有管理代理必须使用相同的子网。 例如，如果主机有两个接口 10.10.116.61 和 10.10.124.61，使用 10.10.116.0/24 可以使用第一个接口。斜杠后的数字表示子网掩码的位数。	无
ma.server.dbdevicepath	存储 HADB 设备信息的路径。	Solaris 和 Linux：/var/opt/SUNWhadb/4 Windows：HADB_install_dir\device

表 3-3 配置文件设置 (续)

设置名称	说明	默认值
ma.server.dbhistorypath	存储 HADB 历史文件的路径。	Solaris 和 Linux : /var/opt/SUNWhadb Windows : REPLACEDIR (运行时由实际 URL 替换。)
ma.server.dbconfigpath	存储节点配置数据的路径。	Solaris 和 Linux : /var/opt/SUNWhadb/dbdef Windows : C:\Sun\SUNWhadb\dbdef
repository.dr.path	域系统信息库文件的路径。	Solaris 和 Linux : /var/opt/SUNWhadb/repository Windows : C:\Sun\SUNWhadb\repository

使用 hadbm 管理命令

使用 hadbm 命令行实用程序来管理 HADB 域及其数据库实例和节点。hadbm 实用程序 (也称为管理客户机) 向指定的管理代理 (用作管理服务器) 发送管理请求, 该管理代理可以从系统信息库访问数据库配置。

本节介绍了 hadbm 命令行实用程序, 包括以下主题:

- 第 57 页中的 “命令语法”
- 第 58 页中的 “安全性选项”
- 第 59 页中的 “常规选项”
- 第 60 页中的 “环境变量”

命令语法

hadbm 实用程序位于 `HADB_install_dir/bin` 目录中。hadbm 命令的常规语法如下:

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

子命令标识了要执行的操作或任务。子命令区分大小写。大多数子命令都具有一个操作数 (通常为 `dbname`)。

选项修改了 hadbm 如何执行子命令。选项区分大小写。每个选项都有长短两种形式。在短形式之前加一个破折号 (-); 在长形式前加两个破折号 (--)。大多数选项都需要参数值, 但布尔选项除外 (布尔选项必须存在才能启用功能)。选项对于命令的成功执行不是必需的。

如果子命令需要数据库名称，而您未指定，则 `hadbm` 将使用默认数据库 `hadb`。

示例 3-1 hadbm 命令示例

以下说明了 `status` 子命令：

```
hadbm status --nodes
```

安全性选项

出于安全性原因，所有 `hadbm` 命令都需要管理员密码。创建数据库或域时，使用 `--adminpassword` 选项设置密码。自此之后，您必须在执行数据库或域操作时指定该密码。

为增强安全性，请使用 `--adminpasswordfile` 选项指定包含密码的文件，而不是在命令行上输入密码。使用以下行定义密码文件中的密码：

```
HADB_M_ADMINPASSWORD=password
```

用密码替换 *password*。忽略文件中的其他内容。

如果同时指定了 `--adminpassword` 和 `--adminpasswordfile` 选项，则 `--adminpassword` 将优先使用。如果需要密码，但未在命令中指定密码，则 `hadbm` 将提示您输入密码。

注 - 仅在创建数据库或域时，才可以设置管理员密码，此后无法更改此密码。

除管理员密码之外，HADB 还需要数据库密码以执行修改数据库模式的操作。使用以下命令时，必须使用上述两个密码：`hadbm create`、`hadbm addnodes` 和 `hadbm refragment`。

在命令行上使用 `--dbpassword` 选项指定数据库密码。与管理员密码类似，您也可以将密码放入文件并使用 `--dbpasswordfile` 选项指定文件位置。使用以下行设置密码文件中的密码：

```
HADB_M_DBPASSWORD=password
```

为了测试或评估，您可以在创建数据库或域时使用 `--no-adminauthentication` 选项禁用密码验证。有关更多信息，请参见第 62 页中的“创建数据库”和第 61 页中的“创建管理域”。

下表汇总了 `hadbm` 安全性命令行选项。

表 3-4 hadbm 安全性选项

选项 (短形式)	说明
--adminpassword= <i>password</i> -w	指定数据库或域的管理员密码。如果创建数据库或域时使用了该选项，则必须在每次使用 hadbm 操作数据库或域时提供该密码。 可以使用该选项或 --adminpasswordfile，但两者不能同时使用。
--adminpasswordfile= <i>filepath</i> -W	指定包含数据库或域的管理员密码的文件。如果创建数据库或域时使用了该选项，则必须在每次使用 hadbm 操作数据库或域时提供该密码。 可以使用该选项或 --adminpassword，但两者不能同时使用。
--no-adminauthentication -U	创建数据库或域时，使用该选项可指定无需管理员密码。出于安全性原因，请勿在生产部署中使用该选项。
--dbpassword= <i>password</i> -P	指定数据库密码。如果在创建数据库时使用了该选项，则必须在每次使用 hadbm 命令操作数据库时提供该密码。为 HADB 系统用户创建密码。至少应为 8 个字符。可以使用该选项或 --dbpasswordfile，但两者不能同时使用。
--dbpasswordfile= <i>filepath</i> -P	指定包含 HADB 系统用户密码的文件。可以使用该选项或 --dbpassword，但两者不能同时使用。

常规选项

常规命令选项可用于任何 hadbm 子命令。它们都是布尔选项，默认情况下为 `false`。下表介绍了 hadbm 常规命令选项。

表 3-5 hadbm 常规选项

选项 (短形式)	说明
--quiet -q	在无提示、没有任何描述性消息的情况下执行子命令。
--help -?	显示该命令和所有支持的子命令的简短说明。无需任何子命令。
--version -V	显示 hadbm 命令的版本详细信息。无需任何子命令。
--yes -y	以非交互模式执行子命令。
--force -f	如果已满足命令的发出条件，则非交互地执行命令，并且不抛出错误。

表 3-5 hadbm 常规选项 (续)

选项 (短形式)	说明
--echo	显示带有所有选项的子命令和选项的用户定义值或默认值，然后执行子命令。
-e	
--agent= <i>URL</i>	管理代理的 URL。 <i>URL</i> 为： <i>hostlist:port</i> ，其中 <i>hostlist</i> 是以逗号分隔的主机名或 IP 地址列表， <i>port</i> 是运行管理代理的端口号。
-m	默认值为 localhost:1862。
	注：该选项对 <code>hadbm addnodes</code> 无效。

环境变量

为了方便起见，您可以设置环境变量而不用指定命令选项。下表介绍了与 `hadbm` 命令选项对应的环境变量。

表 3-6 HADB 选项和环境变量

长形式	短形式	默认值	环境变量
--adminpassword	-w	无	\$HADBM_ADMINPASSWORD
--agent	--m	localhost:1862	\$HADBM_AGENT
--datadevices	-a	1	\$HADBM_DATADEVICES
dbname	无	hadb	\$HADBM_DB
--dbpassword	-p	无	\$HADBM_DBPASSWORD
--dbpasswordfile	-P	无	\$HADBM_DBPASSWORDFILE
--devicepath	-d	Solaris 和 Linux：/var/opt/SUNWhadb Windows：C:\Sun\AppServer \SUNWhadb\vers，其中 <i>vers</i> 是 HADB 的版本号。	\$HADBM_DEVICEPATH
--devicesize	-z	无	\$HADBM_DEVICESIZE
--echo	-e	False	\$HADBM_ECHO
--fast	-F	False	\$HADBM_FAST
--force	-f	False	\$HADBM_FORCE
--help	-?	False	\$HADBM_HELP

表 3-6 HADB 选项和环境变量 (续)

长形式	短形式	默认值	环境变量
--historypath	-t	Solaris 和 Linux: /var/opt/SUNWhadb Windows: REPLACEDIR, 运行时由实际 URL 替换。	\$HADBM_HISTORYPATH
--hosts	-H	无	\$HADBM_HOSTS
--interactive	-i	True	\$HADBM_INTERACTIVE
--no-refragment	-r	False	\$HADBM_NOREFRAGMENT
--portbase	-b	15200	\$HADBM_PORTBASE
--quiet	-q	False	\$HADBM_QUIET
--repair	-R	True	\$HADBM_REPAIR
--rolling	-g	True	\$HADBM_ROLLING
--saveto	-o	无	\$HADBM_SAVETO
--set	-S	无	\$HADBM_SET
--spares	-s	0	\$HADBM_SPARES
--startlevel	-l	normal	\$HADBM_STARTLEVEL
--version	-V	False	\$HADBM_VERSION
--yes	-y	False	\$HADBM_YES

配置 HADB

本节介绍了以下基本 HADB 配置任务：

- 第 61 页中的“创建管理域”
- 第 62 页中的“创建数据库”
- 第 67 页中的“查看和修改配置属性”
- 第 72 页中的“配置 JDBC 连接池”

创建管理域

命令 `hadbm createdomain` 可以创建包含指定 HADB 主机的管理域。该命令初始化主机和持久性配置存储之间的内部通信通道。

该命令的语法为：

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

hostlist 操作数是以逗号分隔的 HADB 主机列表，其中每个主机都是一个有效的 IPv4 网络地址。*hostlist* 中包含您希望置入新域中的所有主机。

有关命令选项的说明，请参见第 59 页中的“常规选项”。

使用该命令之前，请确保 *hostlist* 中的每台主机都在运行 HADB 管理代理。此外，这些管理代理必须：

- 不是现有域的成员。
- 配置为使用同一端口。
- 可以使用 IP 多址广播并通过 UDP、TCP 相互访问。

hadbm 创建管理域之后，将启用域中的所有主机。接下来管理代理就可以管理数据库了。创建 HADB 域之后，接着应创建 HADB 数据库。有关创建 HADB 数据库的更多信息，请参见第 62 页中的“创建数据库”。

示例 3-2 创建 HADB 管理域

以下示例将在四个指定主机上创建管理域：

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

hadbm 成功执行该命令之后，您将看到以下消息：

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

创建 HADB 域之后，使用管理代理注册 HADB 软件包的路径和版本。

创建数据库

使用 `hadbm create` 命令手动创建数据库。

使用此命令创建数据库之前，请先创建管理域并注册 HADB 软件包。如果在运行 `hadbm create` 时未执行这两步，则该命令将隐式执行它们。尽管这看起来可能更省事，但是任何命令的失败都可能会使调试变得困难。此外，`hadbm create` 不是原子操作，也就是说，如果任一隐式命令失败，执行成功的命令将不会被回滚。因此，最好只在创建域并注册 HADB 软件包之后创建数据库。

例如，如果 `hadbm createdomain` 和 `hadbm registerpackage` 执行成功，但 `hadbm create database` 失败，则 `hadbm createdomain` 和 `hadbm registerpackage` 所做的更改将保留下来。

▼ 创建数据库

- 1 创建管理域。
有关更多信息，请参见第 61 页中的“创建管理域”。
- 2 注册 HADB 软件包。
有关更多信息，请参见第 45 页中的“注册 HADB 软件包”。
- 3 使用 `hadbm create` 命令创建数据库。
有关命令语法的信息，请参见以下部分。

hadbm create 命令语法

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

`dbname` 操作数指定数据库名称，该名称必须唯一。为了确保数据库名称唯一，请使用 `hadbm list` 命令列出现有数据库名称。请使用默认数据库名称，除非您需要创建多个数据库。例如，要在同一组 HADB 计算机上创建具有独立数据库的多个群集，请为每个群集使用单独的数据库名称。

`hadbm create` 命令将错误消息写入控制台而不是日志文件。

表 3-7 介绍了特殊的 `hadbm create` 命令选项。有关其他命令选项的说明，请参见第 59 页中的“常规选项”。

表 3-7 hadbm create 选项

选项（短形式）	说明	默认值
<code>--datadevices= number</code>	每个节点上的数据设备数目（在 1 至 8 之间，包括 1 和 8）。数据设备从 0 开始编号。	1
<code>-a</code>		

表 3-7 hadbm create 选项 (续)

选项 (短形式)	说明	默认值
--devicepath= <i>path</i> -d	<p>设备的路径。有四种设备：</p> <ul style="list-style-type: none"> ■ DataDevice ■ NiLogDevice (节点内部日志设备) ■ RelalgDevice (关系代数查询设备) ■ NoManDevice (节点管理器设备) <p>此路径必须存在并且可写。要为每个节点或每个设备设置不同的设备路径，请参见第 66 页中的“设置异构设备路径”。</p>	<p>Solaris 和 Linux: /var/opt/SUNWhadb</p> <p>Windows: C:\Sun\AppServer\SUNWhadb\vers, 其中 vers 是 HADB 的版本号。</p> <p>默认值由管理代理配置文件中的 ma.server.dbdevicepath 指定。有关详细信息，请参见第 56 页中的“配置文件”。</p>
--devicesize= <i>size</i> -z	<p>每个节点的设备大小。有关更多信息，请参见第 66 页中的“指定设备大小”。</p> <p>要增加设备大小，请参见第 83 页中的“为现有节点添加存储空间”。</p>	<p>1024 MB</p> <p>最大值小于操作系统文件的最大大小或为 256 GB。最小值为：</p> $(4 \times \text{LogbufferSize} + 16\text{MB}) / n$ <p>其中 <i>n</i> 为选项 --datadevices 指定的数据设备数。</p>
--historypath= <i>path</i> -t	<p>历史文件的路径。此路径必须已存在并且可写。</p> <p>有关历史文件的更多信息，请参见第 96 页中的“清除和归档历史文件”。</p>	<p>默认值由管理代理配置文件中的 ma.server.dbhistorypath 指定。有关详细信息，请参见第 56 页中的“配置文件”。</p> <p>Solaris 和 Linux: /var/opt/SUNWhadb</p> <p>在 Windows 上: REPLACEDIR (运行时由实际 URL 替换。)</p>
--hosts= <i>hostlist</i> -H	<p>数据库节点的以逗号分隔的主机名或 IP 地址 (仅限于 IPv4) 列表。使用 IP 地址以避免依赖 DNS 查找。主机名必须为绝对主机名。不能使用 localhost 或 127.0.0.1 作为主机名。</p> <p>有关更多信息，请参见第 65 页中的“指定主机”。</p>	无
--package= <i>name</i> -k	<p>HADB 软件包的名称 (版本)。如果未找到该软件包，则注册默认软件包。</p> <p>此选项已过时。使用 hadbm registerpackage 命令在域中注册软件包。</p>	无

表 3-7 hadbm create 选项 (续)

选项 (短形式)	说明	默认值
<code>--packagepath=path-L</code>	HADB 软件包的路径。仅当未在域中注册软件包时使用。 此选项已过时。使用 <code>hadbm registerpackage</code> 命令在域中注册软件包。	无
<code>--portbase= number</code> <code>-b</code>	节点 0 所使用的端口基准号。从该基准号开始，以 20 为增量自动为后续节点指定端口基准号。每个节点均使用其端口基准号和接下来的五个连续编号端口。 要在一台计算机上运行几个数据库，请制订明确分配端口号的计划。	15200
<code>--spares= number</code> <code>-s</code>	备用节点的数目。此数目必须为偶数，并且必须小于 <code>--hosts</code> 选项中指定的节点数目。	0
<code>--set=attr-val-list</code> <code>-S</code>	以逗号分隔的数据库配置属性列表 (采用 <code>name=value</code> 格式)。有关数据库配置属性的说明，请参见第 96 页中的“清除和归档历史文件”。	无

示例 3-3 创建数据库的示例

以下命令是创建数据库的一个示例：

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

指定主机

使用 `--hosts` 选项为数据库中的节点指定以逗号分隔的主机名或 IP 地址列表。`hadbm create` 命令为列表中的每个主机名 (或 IP 地址) 创建一个节点。节点的数目必须为偶数。使用重复的主机名将在同一台计算机上创建具有不同端口号的多个节点。确保同一台计算机上的节点不是镜像节点，并且不是来自不同的 DRU。

节点将按照在此选项中列出的顺序从零开始编号。第一对镜像节点是节点零 (0) 和一 (1)，第二对是二 (2) 和三 (3)，依此类推。奇数编号的节点位于一个 DRU 中，偶数编号的节点位于另一个 DRU 中。如果使用 `--spares` 选项，则备用节点为具有最大编号的那些节点。

有关配置双网络接口的信息，请参见第 34 页中的“配置网络冗余”。

指定设备大小

使用 `--devicesize` 选项指定设备大小。建议的设备大小为：

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

其中

- x 为用户数据的总大小
- n 为节点的数目（由 `--hosts` 选项指定）
- d 为每个节点的设备数目（由 `--datadevices` 选项指定）
- l 为日志缓冲区的大小（由属性 `LogBufferSize` 指定）

如果可能发生重新分段（例如，使用 `hadbm addnodes`），则建议的设备大小为：

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

设置异构设备路径

要为每个节点或服务设置不同的设备路径，请使用 `hadbm create` 的 `--set` 选项。有四种类型的设备：`DataDevice`、`NiLogDevice`（节点内部日志设备）、`RelalgDevice`（关系代数查询设备）和 `NoManDevice`（节点管理器设备）。每个 `name=value` 对的语法如下，其中仅当 `device` 为 `DataDevice` 时才需要 `-devno`：

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

例如：

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

您还可以为历史文件设置异构路径，如下所示：

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

有关历史文件的信息，请参见第 96 页中的“清除和归档历史文件”。

所有不是为特定节点或设备设置的设备路径的默认值都为 `--devicepath` 的值。

注 – 使用 `hadbm set` 和 `hadbm addnodes` 命令更改设备路径和历史文件的位置。

错误诊断

如果创建数据库时遇到困难，请检查以下内容：

- 确保已在所有主机上启动管理代理并定义了 HADB 域。有关详细信息，请参见第 50 页中的“启动管理代理”。

- 必须将文件和目录权限设置为允许以下用户对安装、历史记录、设备和配置路径具有读取、写入和执行权限：
 - Sun Java System Application Server 管理用户（在安装过程中设置）
 - HADB 系统用户
 - 有关设置用户权限的详细信息，请参见第 33 页中的“设置 HADB 前的准备工作”。

Application Server 和 HADB 端口指定不能与同一计算机上的其他端口指定发生冲突。建议的默认端口指定为：

- Sun Java System Message Queue : 7676
- IIOP : 3700
- HTTP 服务器 : 80
- 管理服务器 : 4848
- HADB 节点：每个节点使用六个连续端口。例如，对于默认端口 15200，节点 0 使用 15200 至 15205，节点 1 使用 15220 至 15225，依此类推。

磁盘空间必须足够；请参见《Sun Java System Application Server 9.1 发行说明》。

查看和修改配置属性

您可以分别使用 `hadbm get` 和 `hadbm set` 命令来查看和修改数据库配置属性。

获取配置属性的值

要获得配置属性的值，请使用 `hadbm get` 命令。有关有效属性的列表，请参见第 68 页中的“配置属性”。该命令语法为：

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

dbname 操作数指定数据库名称。默认值为 `hadb`。

attribute-list 操作数是以逗号分隔或包含在引号之间并以空格分隔的属性的列表。--all 选项显示所有属性的值。有关 `hadbm get` 的所有属性的列表，请参见第 68 页中的“配置属性”。

有关命令选项的说明，请参见第 59 页中的“常规选项”。

示例 3-4 使用 `hadbm get` 的示例

```
hadbm get JdbcUrl,NumberOfSessions
```

设置配置属性的值

要设置配置属性的值，请使用 `hadbm set` 命令。有关有效属性的列表，请参见第 68 页中的“配置属性”。

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

dbname 操作数指定数据库名称。默认值为 `hadb`。

attribute=value 列表是以逗号分隔或包含在引号之间并以空格分隔的属性的列表。

有关命令选项的说明，请参见第 59 页中的“常规选项”。

如果该命令执行成功，它将重新启动数据库使其恢复先前状态或进入更佳状态。有关数据库状态的信息，请参见第 88 页中的“获取 HADB 的状态”。按照第 79 页中的“重新启动数据库”中的说明重新启动 HADB。

无法使用 `hadbm set` 设置以下属性，而应在创建数据库时设置它们（请参见第 62 页中的“创建数据库”）。

- DatabaseName
- DevicePath
- HistoryPath
- NumberOfDatadevices
- Portbase
- JdbcUrl（其值在数据库创建期间根据 `--hosts` 和 `--portbase` 选项进行设置）。

注 - 使用 `hadbm set` 设置除 `ConnectionTrace` 或 `SQLTraceMode` 之外的任何配置属性将导致 HADB 的滚动重新启动。在滚动重新启动过程中，每次将停止并启动（使用新配置）一个节点；不中断 HADB 服务。

如果设置了 `ConnectionTrace` 或 `SQLTraceMode`，将不会发生滚动重新启动，但更改仅适用于从 Application Server 实例建立的新的 HADB 连接。

配置属性

下表列出了可以通过 `hadbm set` 修改并通过 `hadbm get` 检索的配置属性列表。

表 3-8 配置属性

属性	说明	默认值	范围
ConnectionTrace	如果为 true，当客户机连接（JDBC、ODBC）开始或结束时将在 HADB 历史文件中记录一条消息。	False	True 或 False
CoreFile	请勿更改默认值。	False	True 或 False
DatabaseName	数据库名称。	hadb	
DataBufferPoolSize	在共享内存中分配的数据缓冲池的大小。	200 MB	16 - 2047 MB
DataDeviceSize	指定节点的设备大小。有关建议的 DataDeviceSize 的信息，请参见第 66 页中的“指定设备大小”。 最大值为 256 GB 或操作系统文件的最大大小，取其中的较小值。最小值为： $(4 \times \text{LogbufferSize} + 16\text{MB}) / n$ 其中 n 为数据设备的数目。	1024 MB	32 - 262144 MB
PackageName	数据库使用的 HADB 软件包的名称。	V4.x.x.x	无
DevicePath	设备的位置。这些设备包括： <ul style="list-style-type: none"> ■ 数据设备 (DataDevice) ■ 节点内部日志设备 (NiLogDevice) ■ 关系代数查询设备 (RelalgDevice) 	Solaris 和 Linux : /var/opt/SUNWhadb Windows : C:\Sun\AppServer\SUNWhadb\vers，其中 vers 是 HADB 的版本号。	
EagerSessionThreshold	确定使用正常还是主动空闲会话失效。 在正常空闲会话失效中，空闲时间超过 SessionTimeout 秒的会话将失效。 当并发会话的数目超过最大会话数目的 EagerSessionThreshold% 时，空闲时间超过 EagerSessionTimeout 秒的会话将失效。	NumberOfSessions 属性值的一半	0 - 100
EagerSessionTimeout	使用主动会话失效时，在数据库连接失效之前该连接可以处于空闲状态的秒数。	120 秒	0-2147483647 秒
EventBufferSize	记录数据库事件的事件缓冲区的大小。如果设置为 0，将不执行事件缓冲区日志记录。 在故障期间，将转储事件缓冲区。在试验部署期间，这将给出有关故障原因的有价值的信息并且非常有用。 将事件写入内存会影响性能。	0 MB	0-2097152 MB

表 3-8 配置属性 (续)

属性	说明	默认值	范围
HistoryPath	包含信息、警告和错误消息的 HADB 历史文件的位置。 此属性为只读属性。	Solaris 和 Linux: /var/opt/SUNWhadb Windows: REPLACEDIR (运行时由实际 URL 替换。)	
InternalLogbufferSize	记录与存储数据相关的操作的节点内部日志设备的大小。	12 MB	4 - 128 MB
JdbcUrl	数据库的 JDBC 连接 URL。 此属性为只读属性。	无	
LogbufferSize	记录与数据相关的操作的日志缓冲区的大小。	48 MB	4 - 2048 MB
MaxTables	HADB 数据库中允许的表的最大数目。	1100	100 - 1100
NumberOfDatadevices	HADB 节点使用的数据设备的数目。 此属性为只读属性。	1	1 - 8
NumberOfLocks	HADB 节点分配的锁的数目。	50000	20000- 1073741824
NumberOfSessions	可以为 HADB 节点打开的会话（数据库连接）的最大数目。	100	1 - 10000
PortBase	用于为不同的 HADB 进程创建不同的端口号的基准端口号。 此属性为只读属性。	15200	10000 - 63000
RelalgDeviceSize	关系代数查询中使用的设备的大小。	128 MB	32 - 262144 MB
SessionTimeout	使用正常会话失效时，在数据库连接失效之前该连接可以处于空闲状态的时间。	1800 秒	0-2147483647 秒
SQLTraceMode	写入历史文件的有关已执行的 SQL 查询的信息量。 如果为 SHORT，则记录 SQL 会话的登录和注销。如果为 FULL，则记录正在准备和正在执行的所有 SQL 查询，包括参数值。	NONE	NONE/SHORT/FULL
StartRepairDelay	备用节点允许故障活动节点执行节点恢复的最长时间。如果故障节点在此时间间隔内无法恢复，备用节点将开始从故障节点的镜像复制数据并成为活动节点。建议不要更改默认值。	20 秒	0 - 100000 秒

表 3-8 配置属性 (续)

属性	说明	默认值	范围
StatInterval	<p>HADB 节点向其历史文件写入吞吐量和响应时间统计信息的时间间隔。要禁用此属性，请将其设置为 0。</p> <p>以下是统计信息行的示例：</p> <pre>Req-reply time: # 123, min= 69 avg= 1160 max= 9311 %=100.0</pre> <p>符号 (#) 后的数字是在 StatInterval 内处理的请求的数目。接下来的三个数字是在 StatInterval 内完成的事务所用的最短、平均和最长时间（以微秒为单位）。百分比符号 (%) 后的数字是在 StatInterval 中 15 毫秒之内成功完成的事务数目。</p>	600 秒	0 - 600 秒
SyslogFacility	<p>向 syslog 报告时使用的工具。应当配置 syslog 守护进程（有关详细信息，请参见 man syslogd.conf）。</p> <p>请使用同一计算机上运行的其他应用程序未使用的工具。</p> <p>设置为 none 可以禁用 syslog 日志记录。</p>	local0	local0、local1、local2、local3、local4、local5、local6、local7、kern、user、mail、daemon、auth、syslog、lpr、news、uucp、cron、none
SysLogging	如果为 true，HADB 节点将信息写入操作系统的 syslog 文件。	True	True 或 False
SysLogLevel	保存到操作系统的 syslog 文件的 HADB 消息的最低级别。将记录该级别或更高级别的所有消息。例如，"info" 将记录所有消息。	warning	nonealert errorwarning info
SyslogPrefix	在由 HADB 写入的所有 syslog 消息之前插入的文本字符串。	hadb-dbname	
TakeoverTime	节点出现故障和该节点的镜像接管之间的时间。请勿更改默认值。	10000（毫秒）	500 - 16000 毫秒

配置 JDBC 连接池

Application Server 使用 Java Database Connectivity (JDBC) API 与 HADB 进行通信。
`asadmin configure-ha-cluster` 命令自动创建 JDBC 连接池以与 HADB 一起使用（用于群集 *cluster-name*）。连接池的名称为 "*cluster-name-hadb-pool*"。JDBC 资源的 JNDI URL 为 "*jdbc/cluster-name-hastore*"。

连接池的初始配置通常已足够。添加节点时，更改稳定的池大小，以使每个 HADB 活动节点都有八个连接。请参见第 84 页中的“添加节点”。

本节包含以下主题：

- 第 72 页中的“获取 JDBC URL”
- 第 72 页中的“创建连接池”
- 第 74 页中的“创建 JDBC 资源”

有关连接池和 JDBC 资源的常规信息，请参见《Sun Java System Application Server 9.1 高可用性管理指南》。

获取 JDBC URL

在可以设置 JDBC 连接池之前，需要使用 `hadbm get` 命令来确定 HADB 的 JDBC URL，如下所示：

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

例如：

```
hadbm get JdbcUrl
```

该命令显示 JDBC URL，其形式如下所示：

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

删除 `jdbc:sun:hadb:` 前缀并将 `host:port, host:port...` 部分用作 `serverList` 连接池属性（表 3-10 所示）的值。

创建连接池

下表汇总了 HADB 所需的连接池设置。添加节点时，请更改稳定池大小，但不要更改其他设置。

表 3-9 HADB 连接池设置

设置	HADB 所需的值
名称	HADB JDBC 资源的池名称设置必须指定此名称
数据库供应商	HADB 4.4
全局事务支持	取消选中/false
数据源类名称	<code>com.sun.hadb.jdbc.ds.HadbDataSource</code>
稳定的池大小	每个活动的 HADB 节点使用 8 个连接。有关更多详细信息，请参见 <i>System Deployment Guide</i> 。
需要连接验证	选中/true
验证方法	<code>meta-data</code>
表名称	不指定
所有连接均失败	取消选中/false
事务隔离	<code>repeatable-read</code>
保证隔离层	选中/true

下表汇总了 HADB 所需的连接池属性。添加节点时，请更改 `serverList`，但不要更改其他属性。

表 3-10 HADB 连接池属性

属性	说明
<code>username</code>	在 <code>asadmin create-session-store</code> 命令中使用的 <code>storeuser</code> 的名称。
<code>password</code>	在 <code>asadmin create-session-store</code> 命令中使用的密码 (<code>storepassword</code>)。
<code>serverList</code>	HADB 的 JDBC URL。要确定该值，请参见第 72 页中的“获取 JDBC URL”。 如果将节点添加到数据库中，则必须更改此值。请参见第 84 页中的“添加节点”。
<code>cacheDatabaseMetaData</code>	需要时设置为 <code>false</code> ，以确保对 <code>Connection.getMetaData()</code> 的调用可以获得对数据库的调用，从而确保连接有效。
<code>eliminateRedundantEndTransaction</code>	必要时设置为 <code>true</code> 可以通过消除冗余提交和回滚请求以及忽略这些请求（如果未打开任何事务）来提高性能。
<code>maxStatement</code>	在驱动程序语句池中高速缓存的每个打开的连接的最大语句数目。请将此属性设置为 20。

示例 3-5 创建连接池

下面是一个创建 HADB JDBC 连接池的示例 `asadmin create-jdbc-connection-pool` 命令：

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

在 Solaris 上，用双反斜杠 (\) 对属性值内的冒号字符 (:) 进行转义。在 Windows 上，用单反斜杠 (\) 对冒号字符 (:) 进行转义。

创建 JDBC 资源

下表汇总了 HADB 所需的 JDBC 资源设置。

表 3-11 HADB JDBC 资源设置

设置	说明
JNDI 名称	在会话持久性配置中，以下 JNDI 名称为默认值：jdbc/hastore。您可以使用默认名称，也可以使用其他名称。 当激活可用性服务时，还必须将此 JNDI 名称指定为 store-pool-jndi-name 持久性存储属性的值。
池名称	从列表中选择此 JDBC 资源所使用的 HADB 连接池的名称（或 ID）。有关更多信息，请参见第 34 页中的“配置网络冗余”。
已启用的数据源	选中/true

管理 HADB

当替换或升级网络、硬件、操作系统或 HADB 软件时，通常需要执行管理操作。以下各节说明了各种管理操作：

- 第 75 页中的“管理域”
- 第 76 页中的“管理节点”
- 第 78 页中的“管理数据库”

- 第 82 页中的“从会话数据损坏恢复”

管理域

您可以在 HADB 域上执行以下操作：

- 创建域：有关更多信息，请参见第 61 页中的“创建管理域”。
- 第 75 页中的“扩展域”
- 第 75 页中的“删除域”
- 第 76 页中的“列出域中的主机”
- 第 75 页中的“从域中删除主机”

有关命令选项的说明，请参见第 58 页中的“安全性选项”和第 59 页中的“常规选项”。

扩展域

使用 `extenddomain` 将主机添加到一个现有管理域中。该命令语法为：

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

HADB 主机的 IP 地址必须为 IPv4 地址。

有关更多信息，请参见 `hadbm-extenddomain(1)`。

删除域

使用 `deletedomain` 删除管理域。该命令语法为：

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

有关更多信息，请参见 `hadbm-deletedomain(1)`。

从域中删除主机

使用 `reducedomain` 从管理域中删除主机。该命令语法为：

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

有关更多信息，请参见 `hadbm-reducedomain(1)`。

列出域中的主机

使用 `listdomain` 列出管理域中定义的所有主机。该命令语法为：

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

有关更多信息，请参见 `hadbm-listdomain(1)`。

管理节点

您可以在单个节点上执行以下操作：

- 第 76 页中的“启动节点”
- 第 77 页中的“停止节点”
- 第 78 页中的“重新启动节点”

启动节点

您可能需要手动启动已停止的 HADB 节点（停止原因为其主机为进行硬件或软件的升级或替换而脱机）。此外，您可能需要手动启动由于某种原因（非双重故障）而无法重新启动的节点。有关如何从双重故障中恢复的更多信息，请参见第 80 页中的“清除数据库”。

在大多数情况下，您应首先尝试使用 `normal` 启动级别启动节点。如果 `normal` 启动级别失败或超时，则必须使用 `repair` 启动级别。

要启动数据库中的节点，请使用 `hadbm startnode` 命令。语法为：

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

dbname 操作数指定数据库名称。默认值为 `hadb`。

nodeno 操作数指定要启动的节点编号。使用 `hadbm status` 显示数据库中所有节点的编号。

有关更多信息，请参见 `hadbm-startnode(1)`。

启动级别选项

`hadbm startnode` 命令具有一个特殊选项 `--startlevel`（短形式 `-l`），该选项指定了启动节点的级别。

节点启动级别为：

- **normal**（默认值）：使用在节点上找到的本地数据（在内存中以及在磁盘的数据设备文件中）启动节点，并使用节点缺少的最近更新的镜像对其进行同步。
- **repair**：强制节点放弃本地数据并从其镜像复制数据。
- **clear**：重新初始化节点的设备并从该节点的镜像节点强制修复数据。在设备文件需要初始化时使用，如果文件损坏或包含设备文件的磁盘被替换，则该启动级别是必需的。

有关其他命令选项的说明，请参见第 59 页中的“常规选项”。

示例 3-6 启动节点的示例

```
hadbm startnode 1
```

停止节点

您可能需要停止节点以修复或升级主机的硬件或软件。要停止节点，请使用 `hadbm stopnode` 命令。该命令语法为：

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

`nodeno` 操作数指定要停止的节点的编号。此节点编号的镜像节点必须正在运行。使用 `hadbm status` 显示数据库中所有节点的编号。

`dbname` 操作数指定数据库名称。默认值为 `hadb`。

`hadbm stopnode` 命令具有一个特殊选项 `--no-repair`（短形式 `-R`），该选项表明无备用节点可以替换已停止的节点。如果没有此选项，备用节点将启动并接管已停止节点的工作。

有关其他命令选项的说明，请参见第 59 页中的“常规选项”。有关更多信息，请参见 `hadbm-stopnode(1)`。

示例 3-7 停止节点的示例

```
hadbm stopnode 1
```

重新启动节点

如果发现行为异常（例如，过度 CPU 消耗），您可能需要重新启动节点。

要重新启动数据库中的节点，请使用 `hadbm restartnode` 命令。该命令语法为：

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

dbname 操作数指定数据库名称。默认值为 `hadb`。

nodeno 操作数指定要重新启动的节点的编号。使用 `hadbm status` 显示数据库中所有节点的编号。

`hadbm restartnode` 命令具有一个特殊选项 `--startlevel`（短形式 `-l`），该选项指定了启动节点的级别。有关更多信息，请参见第 77 页中的“启动级别选项”。

有关其他命令选项的说明，请参见第 59 页中的“常规选项”。有关更多信息，请参见 `hadbm-restartnode(1)`。

示例 3-8 重新启动节点的示例

```
hadbm restartnode 1
```

管理数据库

您可以在 HADB 数据库中执行以下操作：

- 第 78 页中的“启动数据库”
- 第 79 页中的“停止数据库”
- 第 79 页中的“重新启动数据库”
- 第 80 页中的“列出数据库”
- 第 80 页中的“清除数据库”
- 第 81 页中的“删除数据库”

启动数据库

要启动数据库，请使用 `hadbm start` 命令。此命令将启动数据库停止之前正在运行的所有节点。在停止数据库之后再次启动数据库时，不会启动单独停止（脱机）的节点。

该命令语法为：

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

dbname 操作数指定数据库名称。默认值为 `hadb`。

有关命令选项的说明，请参见第 59 页中的“常规选项”。有关更多信息，请参见 `hadbm-start(1)`。

示例 3-9 启动数据库的示例

```
hadbm start
```

停止数据库

如果使用单独的操作停止并启动数据库，则数据库处于停止状态时数据不可用。要保持数据可用，可以重新启动数据库，如第 79 页中的“重新启动数据库”所述。

停止数据库以：

- 删除数据库。
- 执行影响所有 HADB 节点的系统维护。

停止数据库之前，可以停止使用该数据库的依赖 Application Server 实例，也可以将其配置为使用持久性类型而不使用 `ha`。

停止数据库时，将停止数据库中所有正在运行的节点并且数据库将变为已停止状态。有关数据库状态的更多信息，请参见第 89 页中的“数据库状态”。

要停止数据库，请使用 `hadbm stop` 命令。该命令语法为：

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

dbname 操作数指定数据库名称。默认值为 `hadb`。

有关命令选项的说明，请参见第 59 页中的“常规选项”。有关更多信息，请参见 `hadbm-stop(1)`。

示例 3-10 停止数据库的示例

```
hadbm stop
```

重新启动数据库

如果发现行为异常（例如持续超时问题），则可能需要重新启动数据库。在某些情况下，重新启动可以解决此问题。

重新启动数据库时，数据库及其数据仍然可用。如果使用单独的操作停止并启动 HADB，则 HADB 处于停止状态时数据和数据库服务不可用。这是因为，默认情况下，`hadbm restart` 执行节点的滚动重新启动：它逐个停止并启动节点。相反，`hadbm stop` 将同时停止所有节点。

要重新启动数据库，请使用 `hadbm restart` 命令。该命令语法为：

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

dbname 操作数指定数据库名称。默认值为 `hadb`。

特殊选项 `--no-rolling`（短形式 `-g`）指定同时重新启动所有节点（这将使服务受到损失）。如果没有此选项，此命令将重新启动数据库中的每个节点并使其达到当前状态或更佳状态。

有关其他命令选项的说明，请参见第 59 页中的“常规选项”。有关更多信息，请参见 `hadbm-restart(1)`。

例如：

```
hadbm restart
```

列出数据库

要列出 HADB 实例中的所有数据库，请使用 `hadbm list` 命令。该命令语法为：

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

有关命令选项的说明，请参见第 59 页中的“常规选项”。有关更多信息，请参见 `hadbm-list(1)`。

清除数据库

在以下情况时清除数据库：

- `hadbm status` 命令显示数据库处于不可操作状态，请参见第 88 页中的“获取 HADB 的状态”。
- 多个节点不能响应并长时间处于等待状态。
- 从会话数据损坏中恢复。请参见第 82 页中的“从会话数据损坏恢复”。

hadbm clear 命令可以停止数据库节点，并清除数据库设备，然后启动节点。该命令将删除 HADB 中的 Application Server 模式数据存储，包括表、用户名和密码。运行 hadbm clear 之后，使用 asadmin configure-ha-cluster 重新创建数据模式、重新配置 JDBC 连接池并重新装入会话持久性存储。

该命令语法为：

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

dbname 操作数指定数据库名称。默认值为 hadb。

下表介绍了特殊的 hadbm clear 命令选项。有关其他选项的说明，请参见第 59 页中的“常规选项”。

有关更多信息，请参见 hadbm-clear(1)。

表 3-12 hadbm clear 选项

选项	说明	默认值
--fast	在初始化数据库时跳过设备初始化。如果磁盘存储设备损坏，请勿使用此选项。	不存在
-F		
--spares= <i>number</i>	重新初始化的数据库将具有的备用节点数目。此数目必须为偶数，并且必须小于数据库中的节点数目。	先前的备用节点数目
-s		

例如：

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

删除数据库

要删除现有数据库，请使用 hadbm delete 命令。该命令将删除数据库的配置文件、设备文件和历史文件，并释放共享的内存资源。您要删除的数据库必须存在并且必须处于已停止状态。请参见第 79 页中的“停止数据库”。

该命令语法为：

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

`dbname` 操作数指定数据库名称。默认值为 `hadb`。

有关命令选项的说明，请参见第 59 页中的“常规选项”。有关更多信息，请参见 `hadbm-delete(1)`。

示例 3-11 删除数据库的示例

命令：

```
hadbm delete
```

将删除默认数据库 `hadb`。

从会话数据损坏恢复

以下情况表明会话数据可能已损坏：

- 每次应用程序尝试保存会话状态时，Application Server 系统日志 (`server.log`) 都显示错误消息。
- 在会话激活过程中，服务器日志中的错误消息表明找不到会话或无法装入会话。
- 先前钝化后又被激活的会话包含空的或不正确的会话数据。
- 实例出现故障时，故障转移的会话包含空的或不正确的会话数据。
- 实例出现故障时，尝试装入故障转移的会话的实例导致在服务器日志中生成表明找不到会话或无法装入会话的错误。

▼ 使会话存储恢复到一致状态

如果您确定会话存储已损坏，则可以执行以下步骤使其恢复到一致状态：

1 清除会话存储。

确定此操作是否可以修正该问题。如果已修正，则停止。如果未修正（例如，服务器日志中仍出现错误），则继续。

2 重新初始化所有节点上的数据空间并清除数据库中的数据。

请参见第 80 页中的“清除数据库”。

确定此操作是否可以修正该问题。如果已修正，则停止。如果未修正（例如，服务器日志中仍出现错误），则继续。

3 删除并重新创建数据库。

请参见第 81 页中的“删除数据库”和第 62 页中的“创建数据库”。

扩展 HADB

扩展初始 HADB 配置的原因有两个：

- 保存的会话数据量超过了数据设备的现有存储空间。由于数据设备已满，事务可能开始异常中止。
- 用户装入不断增加，以至于耗尽系统资源。需要添加更多主机。

本节介绍如何在无需关闭 Application Server 群集或数据库的情况下扩展 HADB，其中包括以下内容：

- [第 83 页](#)中的“为现有节点添加存储空间”
- [第 84 页](#)中的“添加计算机”
- [第 84 页](#)中的“添加节点”
- [第 86 页](#)中的“重新分段数据库”
- [第 87 页](#)中的“通过重新创建数据库添加节点”

另请参见[第 94 页](#)中的“维护 HADB 计算机”中的相关信息。

为现有节点添加存储空间

添加 HADB 存储空间：

- 如果用户事务重复出现异常中止，并出现以下错误消息之一：
 - 4592: No free blocks on data devices
 - 4593: No unreserved blocks on data devices
- 如果 `hadbm deviceinfo` 命令持续报告空闲大小不足。请参见[第 90 页](#)中的“获取设备信息”。

如果节点上有未使用的磁盘空间或在添加磁盘容量时，您可能还希望向现有节点添加存储空间。有关建议的数据设备大小的信息，请参见[第 66 页](#)中的“指定设备大小”。

要向节点添加存储空间，请使用 `hadbm set` 命令增加数据设备大小。

该命令的语法为：

```
hadbm set DataDeviceSize=size
```

其中 *size* 为数据设备大小（以 MB 为单位）。

有关命令选项的说明，请参见[第 59 页](#)中的“常规选项”。

更改 `FaultTolerant` 或更高状态中的数据库的数据设备大小将升级系统，并且不会丢失数据或影响可用性。在重新配置过程中，数据库仍可操作。在非 `FaultTolerant` 或更差的系统上更改设备大小将导致数据丢失。有关数据库状态的更多信息，请参见[第 89 页](#)中的“数据库状态”。

示例 3-12 设置数据设备大小的示例

以下命令是设置数据设备大小的一个示例：

```
hadbm set DataDeviceSize=1024
```

添加计算机

如果 HADB 需要更多处理或存储容量，则可能需要添加计算机。要添加运行 HADB 的新计算机，请将 HADB 软件包与 Application Server 安装在一起或单独安装（如第 2 章所述）。有关节点拓扑选择的说明，请参见《Sun Java System Application Server 9.1 部署规划指南》中的第 3 章“选择拓扑”。

▼ 将新计算机添加到现有 HADB 实例

- 1 在新节点上启动管理代理。
- 2 将管理域扩展到新主机。
有关详细信息，请参见 `hadbm extenddomain` 命令。
- 3 在这些主机上启动新节点。
有关详细信息，请参见第 84 页中的“添加节点”。

添加节点

要增加 HADB 系统的处理和存储容量，请创建新节点并将其添加到数据库。

添加节点后，请更新 HADB JDBC 连接池的以下属性：

- `serverlist` 属性。
- 稳定的池大小。通常，需要为每个新节点再添加 8 个连接。有关更多信息，请参见《Sun Java System Application Server 9.1 部署规划指南》中的“System Sizing”。

要添加节点，请使用 `hadbm addnodes` 命令。该命令语法为：

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

`dbname` 操作数指定数据库名称。默认值为 `hadb`。数据库必须处于 `HAFaultTolerant` 或 `FaultTolerant` 状态。有关数据库状态的更多信息，请参见第 89 页中的“数据库状态”。

如果不指定 `--devicepath` 和 `--historypath` 选项，则新节点将具有与现有数据库相同的设备路径，并与其使用相同的历史文件。

添加节点可以对现有数据执行重新分段和重新分发以在系统中包含新节点。联机重新分段需要 HADB 节点有足够的磁盘空间，可以在重新分段完成之前同时容纳旧数据和新数据，也就是说用户数据大小切勿超过用户数据可用空间的 50%。有关详细信息，请参见第 90 页中的“获取设备信息”。

注 - 最好在系统负载较小时添加节点。

示例 3-13 添加节点的示例

例如：

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

下表介绍了特殊的 `hadbm addnodes` 命令选项。有关其他选项的说明，请参见第 59 页中的“常规选项”。

表 3-13 hadbm addnodes 选项

选项	说明	默认值
<code>--no-refragment</code> <code>-r</code>	请勿在节点创建期间重新分段数据库；这种情况下，请稍后使用 <code>hadbm refragment</code> 命令重新分段数据库以使用新节点。有关重新分段的详细信息，请参见第 86 页中的“重新分段数据库”。	不存在
	如果没有足够的设备空间来进行重新分段，则重新创建具有更多节点的数据库。请参见第 87 页中的“通过重新创建数据库添加节点”。	
<code>--spares= number</code> <code>-s</code>	新的备用节点（已存在的备用节点除外）的数目。必须为偶数，并且不能大于添加的节点数目。	0

表 3-13 hadbm addnodes 选项 (续)

选项	说明	默认值
--devicepath= <i>path</i>	设备的路径。这些设备包括：	Solaris 和
-d	<ul style="list-style-type: none"> ■ DataDevice ■ NiLogDevice (节点内部日志设备) ■ RelalgDevice (关系代数查询设备) 此路径必须已存在并且可写。要为每个节点或每个设备设置不同的设备路径，请参见第 66 页中的“设置异构设备路径”。	Linux: <i>HADB_install_dir/device</i> Windows: <i>C:\Sun\AppServer\SUNWhadb\vers</i> , 其中 <i>vers</i> 是 HADB 的版本号。
--hosts= <i>hostlist</i>	数据库中新节点的以逗号分隔的新主机名的列表。为列表中的每个以逗号分隔的项目创建一个节点。节点的数目必须为偶数。HADB 主机的 IP 地址必须为 IPv4 地址。	无
-H	使用重复的主机名将在同一台计算机上创建具有不同端口号的多个节点。请确保同一台计算机上的节点不是镜像节点。 奇数编号的节点位于一个 DRU 中，偶数编号的节点位于另一个 DRU 中。如果使用 --spares，则新的备用节点为具有最大编号的那些节点。 如果使用双网络接口创建了数据库，则必须以相同方式配置新的节点。请参见第 34 页中的“配置网络冗余”。	

重新分段数据库

重新分段数据库以在新创建的节点中存储数据。通过重新分段可以在所有活动节点之间平均分配数据。

要重新分段数据库，请使用 `hadbm refragment` 命令。该命令语法为：

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

dbname 操作数指定数据库名称。默认值为 `hadb`。数据库必须处于 `HAFaultTolerant` 或 `FaultTolerant` 状态。有关数据库状态的更多信息，请参见第 88 页中的“获取 HADB 的状态”。

有关命令选项的说明，请参见第 59 页中的“常规选项”。有关更多信息，请参见 `hadbm-fragment(1)`。

联机重新分段需要 HADB 节点有足够的磁盘空间，可以在重新分段完成之前同时容纳旧数据和新数据，也就是说用户数据大小切勿超过用户数据可用空间的 50%。有关详细信息，请参见第 90 页中的“获取设备信息”。

注 – 最好在系统负载较轻时重新分段数据库。

如果多次尝试后该命令仍然失败，请参见第 87 页中的“通过重新创建数据库添加节点”。

示例 3-14 重新分段数据库示例

例如：

```
hadbm refragment
```

通过重新创建数据库添加节点

如果添加新节点时联机重新分段总是失败（由于数据设备空间不足或其他原因），请重新创建具有新节点的数据库。这将导致现有用户数据和模式数据的丢失。

▼ 通过重新创建数据库来添加节点

该步骤将使您可以在整个进程中维持 HADB 可用性。

- 1 对于每个 **Application Server** 实例：
 - a. 禁用负载均衡器中的 **Application Server** 实例。
 - b. 禁用会话持久性。
 - c. 重新启动 **Application Server** 实例。
 - d. 重新启用负载均衡器中的 **Application Server** 实例。

如果不需要维持可用性，则可以同时禁用并重新启用负载均衡器中的所有服务器实例。这样可以节省时间并防止过期会话数据故障转移。

- 2 停止数据库，如第 79 页中的“停止数据库”所述。
- 3 删除数据库，如第 81 页中的“删除数据库”所述。
- 4 重新创建具有其他节点的数据库，如第 62 页中的“创建数据库”所述。
- 5 重新配置 JDBC 连接池，如第 72 页中的“配置 JDBC 连接池”所述。
- 6 重新装入会话持久性存储。

- 7 对于每个 Application Server 实例：
 - a. 禁用负载均衡器中的 Application Server 实例。
 - b. 启用会话持久性。
 - c. 重新启动 Application Server 实例。
 - d. 重新启用负载均衡器中的 Application Server 实例。

如果不需要维持可用性，则可以同时禁用并重新启用负载均衡器中的所有服务器实例。这样可以节省时间并防止过期会话数据故障转移。

监视 HADB

您可以通过以下方式监视 HADB 的活动：

- 第 88 页中的“获取 HADB 的状态”
- 第 90 页中的“获取设备信息”
- 第 92 页中的“获取运行时资源信息”

以下各节简要介绍了 `hadbm status`、`hadbm deviceinfo` 和 `hadbm resourceinfo` 命令。有关 HADB 信息的说明，请参见《Sun Java System Application Server 9.1 Performance Tuning Guide》中的“Performance”。

获取 HADB 的状态

使用 `hadbm status` 命令显示数据库或其节点的状态。该命令语法为：

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

`dbname` 操作数指定数据库名称。默认值为 `hadb`。

`--nodes` 选项（短形式 `-n`）显示了数据库中每个节点的信息。有关更多信息，请参见第 89 页中的“节点状态”。有关其他命令选项的说明，请参见第 59 页中的“常规选项”。

有关更多信息，请参见 `hadbm-status(1)`。

示例 3-15 获取 HADB 状态的示例

例如：

```
hadbm status --nodes
```

数据库状态

数据库的状态汇总了其当前情况。下表介绍了数据库的可能状态。

表 3-14 HADB 状态

数据库状态	说明
高可用性容错 (High-Availability Fault Tolerant, HAFaultTolerant)	数据库具有容错能力，并且在每个 DRU 中至少有一个备用节点。
容错	所有镜像节点对均启动并且正在运行。
可操作	每个镜像节点对中至少有一个节点正在运行。
不可操作	一个或多个镜像节点对中的双方均缺少节点。 如果数据库处于不可操作状态，请清除数据库，如第 80 页中的“清除数据库”所述。
已停止	数据库中没有任何正在运行的节点。
不明	无法确定数据库的状态。

节点状态

通过 `--nodes` 选项使 `hadbm status` 命令显示数据库中每个节点的以下信息：

- 节点编号
- 运行节点的计算机的名称
- 节点的端口号
- 节点的角色。有关角色及其含义的列表，请参见第 89 页中的“节点的角色”。
- 节点的状态。有关状态及其含义的列表，请参见第 90 页中的“节点的状态”。
- 对应的镜像节点的编号。

节点的角色和状态可以更改，如以下各节中所述：

- 第 89 页中的“节点的角色”
- 第 90 页中的“节点的状态”

节点的角色

在创建节点过程中为节点指定了角色，并且节点可以具有以下角色之一：

- **活动**：存储数据并允许客户机访问。活动节点位于镜像对中。

- **备用**：允许客户机访问，但不存储数据。初始化数据设备之后，监视其他数据节点以启动修复（如果另一个节点变为不可用）。
- **脱机**：节点的角色更改之前不提供任何服务。重新联机后，其角色可以更改为以前的角色。
- **关闭**：活动和脱机的中间步骤，等待备用节点接管其工作。备用节点接管之后，将使节点脱机。

节点的状态

节点可处于以下状态之一：

- **正在启动**：节点正在启动。
- **正在等待**：节点无法决定其启动级别并已脱机。如果一个节点处于此状态的时间超过两分钟，则停止该节点，然后在 `repair` 级别启动该节点；请参见第 77 页中的“[停止节点](#)”、第 76 页中的“[启动节点](#)”和第 80 页中的“[清除数据库](#)”。
- **正在运行**：节点正在提供与其角色相应的所有服务。
- **正在停止**：节点处于停止过程中。
- **已停止**：节点已不活动。禁止修复已停止的节点。
- **正在恢复**：正在恢复节点。节点出现故障时，镜像节点将接管故障节点的工作。故障节点将尝试使用主内存中或磁盘上的数据和日志记录进行恢复。故障节点使用镜像节点的日志记录来恢复到出现故障时所执行的事务。如果恢复成功，节点将成为活动节点。如果恢复失败，节点状态将更改为正在修复。
- **正在修复**：正在修复节点。此操作将重新初始化节点，并复制镜像节点的数据和日志记录。修复比恢复所需时间要长。

获取设备信息

监视 HADB 数据（磁盘存储）设备的空闲空间：

- 按照例程，要检查磁盘空间使用的趋向。
- 作为预防性维护的一部分：如果用户负载已增加，并且您希望重新调整数据库配置的大小或比例。
- 作为增大数据库的一部分：运行 `hadbm addnodes` 将新节点添加到系统之前，检查是否有足够的设备空间。请记住，要添加节点您需要现有节点所占的大约 40-50% 的空闲空间。
- 当您在历史文件和 `server.log` 文件中看到如下消息时
 - `No free blocks on data devices`
 - `No unreserved blocks on data devices`。

使用 `hadbm deviceinfo` 命令获取关于数据设备上空闲空间的信息。该命令显示了数据库每个节点的以下信息：

- 分配的总设备大小，以 MB 为单位 (Totalsize)。
- 空闲空间，以 MB 为单位 (Freesize)。
- 设备当前的使用率 (Usage)。

该命令语法为：

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

dbname 操作数指定数据库名称。默认值为 hadb。

--details 选项显示了以下其他信息：

- 设备的读取操作数目。
- 设备的写入操作数目。
- 设备的名称。

有关其他命令选项的说明，请参见第 59 页中的“常规选项”。

有关更多信息，请参见 hadbm-deviceinfo(1)。

要确定用户数据的可用空间，请用总设备大小减去为 HADB 保留的空间：即四倍的 LogBufferSize + 1% 的设备大小。如果不知道日志缓冲区的大小，请使用命令 hadbm get logbufferSize。例如，如果总设备大小为 128 MB，LogBufferSize 为 24 MB，则用户数据的可用空间为 $128 - (4 \times 24) = 32$ MB。32 MB 中一半用于已复制的数据，大约四分之一用于索引，只有百分之二十五用于实际用户数据。

总大小和保留大小之间的差额为用户数据的可用空间。如果将来对数据进行重新分段，则空闲大小必须为用户数据可用空间的大约 50%。如果不进行重新分段，则可以最大限度地利用数据设备。如果系统在设备空间不足的情况下运行，则资源消耗警告将被写入历史文件。

有关调节 HADB 的更多信息，请参见 *Sun Java System Application Server Performance Tuning Guide*。

示例 3-16 获取设备信息的示例

以下命令：

```
hadbm deviceinfo --details
```

显示以下示例结果：

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

示例 3-16 获取设备信息的示例 (续)

```
3      128      126      2%      9500 4500      C:\Sun\SUNWhadb\hadb.data.3
```

获取运行时资源信息

hadbm resourceinfo 命令显示了 HADB 运行时资源信息。使用此信息有助于识别资源争用并减少性能瓶颈。有关详细信息，请参见《Sun Java System Application Server 9.1 Performance Tuning Guide》中的“Tuning HADB”。

该命令语法为：

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

dbname 操作数指定数据库名称。默认值为 hadb。

下表介绍了 hadbm resourceinfo 的特殊命令选项。有关其他命令选项的说明，请参见第 59 页中的“常规选项”。

有关更多信息，请参见 hadbm-resourceinfo(1)。

表 3-15 hadbm resourceinfo 命令选项

选项	说明
--databuf	显示数据缓冲池信息。
-d	有关更多信息，请参见以下第 92 页中的“数据缓冲池信息”。
--locks	显示锁信息。
-l	有关更多信息，请参见以下第 93 页中的“锁信息”。
--logbuf	显示日志缓冲区信息。
-b	有关更多信息，请参见以下第 93 页中的“日志缓冲区信息”。
--nilogbuf	显示节点内部日志缓冲区信息。
-n	有关更多信息，请参见以下第 94 页中的“节点内部日志缓冲区信息”。

数据缓冲池信息

数据缓冲池信息包含：

- NodeNo：节点编号。
- Avail：池中的总可用空间，以 MB 为单位。

- Free: 可用的空闲空间, 以 MB 为单位。
- Access: 从数据库启动到现在, 访问数据缓冲区的累积次数。
- Misses: 从数据库启动到现在, 缺页发生的累积次数。
- Copy-on-Write: 由于检查点操作而复制到数据缓冲区内部的累积页面数。

当用户事务在记录上执行操作时, 包含该记录的页面必须在数据缓冲池中。如果该页面不在数据缓冲池中, 则发生 miss 或缺页。然后事务必须等待, 直到从磁盘的数据设备文件中检索到该页面。

如果未命中率很高, 则增加数据缓冲池。由于未命中次数是累积的, 请定期运行 `hadbm resourceinfo`, 并使用两次运行结果的不同查看未命中率的趋向。如果空闲空间很小, 请勿考虑运行该命令, 因为检查点机制将使新块可用。

示例 3-17 数据缓冲池信息示例

例如:

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

锁信息

锁信息如下:

- NodeNo: 节点编号。
- Avail: 节点上可用的锁的总数目。
- Free: 空闲锁数目。
- Waits: 等待获得锁的事务的数目。该数目是累积的。

一个事务无法使用节点上超过 25% 的可用锁。因此, 大规模执行操作的事务应注意到该限制。最好成批执行此类事务, 其中每批都必须被视为一个独立事务, 即分批提交。需要这样做的原因是, 读取操作 (在 `repeatable read` 隔离层运行) 以及 `delete`、`insert` 和 `update` 操作使用仅在事务终止后才被释放的锁。

要更改 `NumberOfLocks`, 请参见第 96 页中的“清除和归档历史文件”。

示例 3-18 锁信息示例

例如:

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

日志缓冲区信息

日志缓冲区信息为:

- NodeNo: 节点编号
- Available: 为日志缓冲区分配的内存容量 (以 MB 为单位)

- Free: 空闲内存容量 (以 MB 为单位)

如果空闲空间很小, 请勿担心, 因为 HADB 可以启动日志缓冲区压缩。HADB 从环状缓冲区的头部开始压缩, 并在连续日志记录上执行。当 HADB 遇到尚未由节点执行并且尚未由镜像节点接收的日志记录时, 压缩无法继续进行。

示例 3-19 日志缓冲区信息的示例

例如:

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

节点内部日志缓冲区信息

节点内部日志缓冲区信息为:

- 节点编号
- 可用: 为日志设备分配的内存容量 (以 MB 为单位)
- 空闲: 空闲内存容量 (以 MB 为单位)

示例 3-20 内部日志缓冲区信息示例

例如:

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

维护 HADB 计算机

HADB 通过复制镜像节点上的数据来实现容错。在生产环境中, 与其镜像的节点不同, 镜像节点在单独的 DRU 上, 如《Sun Java System Application Server 9.1 部署规划指南》所述。

故障是一种意外事件, 如硬件故障、电源故障或操作系统重新引导。HADB 允许一个节点、一台计算机 (没有镜像节点对)、属于同一 DRU 的一台或多台计算机甚至一个整个的 DRU 的单个故障。但是, HADB **不能**从双重故障 (即一个或多个镜像节点对同时出现故障) 中自动恢复。如果出现双重故障, 则必须清除 HADB 并重新创建其会话存储 (这将删除它的所有数据)。

具体的维护过程取决于您需要在 一台计算机上工作还是在多台计算机上工作。

▼ 在一台计算机上执行维护

此过程适用于规划的和非规划的维护，并且不中断 HADB 可用性。

- 1 执行维护过程并使计算机启动并运行。

- 2 确保 ma 正在运行。

如果 ma 作为 Windows 服务运行或位于 `init.d` 脚本（推荐用于部署）下，则应该已由操作系统启动。如果未启动，请手动启动。请参见第 50 页中的“启动管理代理”。

- 3 启动计算机上的所有节点。

有关更多信息，请参见第 76 页中的“启动节点”。

- 4 检查节点是否处于活动状态并且正在运行。

有关更多信息，请参见第 88 页中的“获取 HADB 的状态”。

▼ 在所有 HADB 计算机上执行规划的维护

规划的维护包含诸如硬件和软件升级之类的操作。此过程不中断 HADB 可用性。

- 1 对于第一个 DRU 中的每台备用计算机，逐个对每台计算机重复单个计算机过程，如第 95 页中的“在一台计算机上执行维护”所述。
- 2 对于第一个 DRU 中的每台活动计算机，逐个对每台计算机重复单个计算机过程，如第 95 页中的“在一台计算机上执行维护”所述。
- 3 对于第二个 DRU 重复步骤 1 和步骤 2。

▼ 在所有 HADB 计算机上执行规划的维护

此过程适用于 HADB 在一台或多台计算机上的情况。在维护过程中将中断 HADB 服务。

- 1 停止 HADB。请参见第 79 页中的“停止数据库”。

- 2 执行维护过程并使所有计算机启动并运行。

- 3 确保 ma 正在运行。

- 4 启动 HADB。

有关更多信息，请参见第 78 页中的“启动数据库”。

完成最后一个步骤后，HADB 数据将重新变为可用。

▼ 在出现故障时执行非规划的维护

● 检查数据库状态。

请参见第 88 页中的“获取 HADB 的状态”。

■ 如果数据库状态为可操作或更佳：

需要非规划的维护的计算机不包含镜像节点。对每台故障计算机执行单个计算机过程（每次一个 DRU）。不中断 HADB 服务。

■ 如果数据库处于不可操作状态：

需要非规划的维护的计算机包含镜像节点。当整个 HADB 位于一台故障计算机上时，就属于这种情况。请先使所有计算机启动并运行。然后清除 HADB 并重新创建会话存储。请参见第 80 页中的“清除数据库”。这将中断 HADB 服务。

清除和归档历史文件

HADB 历史文件记录所有数据库操作和错误消息。HADB 附加在现有历史文件的末尾，因此文件大小将随时间增长。为了节省磁盘空间并防止文件过大，应定期清除和归档历史文件。

要清除数据库的历史文件，请使用 `hadbm clearhistory` 命令。

该命令语法为：

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

dbname 操作数指定数据库名称。默认值为 `hadb`。

使用 `--saveto` 选项（短形式 `-o`）指定要存储旧的历史文件的目录。此目录必须具有适当的写权限。有关其他命令选项的说明，请参见第 59 页中的“常规选项”。

有关更多信息，请参见 `hadbm-clearhistory(1)`。

`hadbm create` 命令的 `--historypath` 选项确定了历史文件的位置。历史文件名称的格式为 `dbname.out.nodeno`。有关 `hadbm create` 的信息，请参见第 62 页中的“创建数据库”。

历史文件格式

历史文件中的每条消息都包含以下信息：

- 生成消息的 HADB 进程的缩写名称。
- 消息的类型：
 - INF—常规信息
 - WRN—警告
 - ERR—错误
 - DBG—调试信息
- 时间戳。时间是从主机的系统时钟获得的。
- 当节点停止或启动时，系统中发生的服务设置更改。

有关资源短缺的消息包含字符串 "HIGH LOAD"。

您无需详细了解历史文件中的所有条目。如果由于某种原因您需要深入了解历史文件，请联系 Sun 用户支持。

配置 Web 服务器以实现负载均衡

本章介绍了如何配置用于 Application Server 9.1 和 GlassFish v2 附带的负载均衡器插件支持的 Web 服务器。Application Server 9.1 附带的负载均衡器插件支持以下 Web 服务器：

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

注 - GlassFish v2 仅支持 Sun Java System Web Server（版本 6.1 和 7.0）。要将负载均衡器插件与 GlassFish v2 一起使用，需要手动安装和配置负载均衡器插件。有关将负载均衡器插件与 GlassFish v2 一起安装的更多信息，请参见《Sun Java System Application Server 9.1 Installation Guide》中的第 1 章“Installing Application Server Software”。

负载均衡器插件的安装程序是 Application Server 9.1 安装程序的一部分，它对 Web 服务器的配置文件进行了一些修改。这些更改取决于所用的 Web 服务器。此外，对于某些 Web 服务器，必须进行手动配置以使负载均衡器正常工作。

注 - 可以将负载均衡器插件与 Sun Java System Application Server 9.1 一起安装在或单独安装在运行支持的 Web 服务器的计算机上。有关安装过程的完整详细信息，请参见《Sun Java System Application Server 9.1 Installation Guide》中的第 1 章“Installing Application Server Software”。

- 第 100 页中的“配置 Sun Java System Web Server”
- 第 107 页中的“使用 Apache Web Server”
- 第 115 页中的“使用 Microsoft IIS”

配置 Sun Java System Web Server

对于 Sun Java System Web Server，使用 Sun Java System Application Server 9.1 安装向导安装负载均衡器插件时，该安装向导将自动执行所有必要的配置。不需要进行手动配置。与 Application Server 9.1 捆绑在一起的负载均衡器插件支持 Sun Java System Web Server 的以下版本：

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

但是，如果使用的是 GlassFish v2，必须从 http://download.java.net/javaee5/external/SunOS_X86/aslb/jars/aslb-9.1-MS4-b7.jar 单独下载 Application Server 负载均衡器插件，并进行一些手动更改以对其进行设置。有关如何为 GlassFish v2 安装和设置该插件的详细步骤，请参阅《Sun Java System Application Server 9.1 Installation Guide》中的“To Install the Load Balancing Plug-in (standalone)”一节。

▼ 配置 Sun Java System Web Server

开始之前

注 - 以下步骤由 Application Server 9.1 的安装程序自动执行。但是，如果使用的是 GlassFish v2，将需要手动执行这些步骤。

- 1 在 Web 服务器实例的 `magnus.conf` 文件中，添加以下行：

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

- 2 如果以下行尚不存在，请附加该行：

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

- 3 在文件 `web-server-install-dir/config/obj.conf` 中，在首次出现的字符串 `nametrans` 之前在单行中插入以下内容：

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

`NameTrans` 条目在 `obj.conf` 中的显示顺序非常重要。安装程序会将 `NameTrans` 条目放在正确的位置，但是，如果要因其他目的而编辑 `obj.conf`，必须确保顺序仍然正确。尤其是，负载均衡器信息必须在 `document-root` 函数之前显示。有关 `obj.conf` 文件的更多信息，请参见《Sun Java System Web Server 7.0 Administrator's Configuration File Reference》。

- 4 将以下行附加到文件 `web-server-install-dir/config/obj.conf` 中：


```
<Object name = "lbplugin">
ObjectType fn="force-type" type="magnus-internal/lbplugin"
PathCheck fn="deny-existence" path="*/WEB-INF/*"
Service type="magnus-internal/lbplugin" fn="service-passthrough"
Error reason="Bad Gateway" fn="send-error" uri="$docroot/badgateway.html"
</Object>
```
- 5 编辑 `web-server-install-dir/start` 脚本，将 `LD_LIBRARY_PATH` 值更新为包括 `app-server-install-dir/lib/lbplugin/lib`。
`app-server-install-dir/lib/lbplugin/lib` 目录包含负载均衡器插件所需的二进制文件。
- 6 （可选的）对于基于 DAS 的新负载均衡器管理，配置 Web 服务器的 SSL 功能。
 有关 Web Server 6.1 的详细说明，请参见第 101 页中的“在 SSL 模式下为 Sun Java System Web Server 6.1 设置负载均衡器”。
 有关 Web Server 7 的详细说明，请参见第 104 页中的“在 SSL 模式下为 Web Server 7 设置负载均衡器”。
- 7 如果 Web 服务器尚未运行，请启动它。

配置 Sun Java System Web Server 以使用“自动应用”功能

“自动应用”是 Application Server 9.1 提供的一种功能，用于通过线路将负载均衡器配置自动发送到 Web 服务器配置目录。有关此功能的更多信息，请参见第 119 页中的“自动应用”。以下过程介绍了如何配置 Sun Java System Web Server（版本 6 和 7）以使用此功能。

▼ 在 SSL 模式下为 Sun Java System Web Server 6.1 设置负载均衡器

注 - 仅当希望使用负载均衡器插件的“自动应用”功能时，才需要执行本节中的步骤。此功能有助于通过线路将负载均衡器配置自动发送到 Web 服务器配置目录。

- 1 使用浏览器，访问 Web Server 的管理控制台并登录。
- 2 选择服务器实例，然后单击“管理”。
- 3 单击“安全性”选项卡。

- 4 通过提供用户名和密码初始化信任数据库。为此，可以使用 `certutil` 命令或 GUI。`certutil` 命令的以下选项可用于初始化信任数据库：

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

- 在 `certutil` 发出提示时，输入密码对您的密钥进行加密。输入密码（密码将用于对密钥进行加密）。密码至少应该包含八个字符，且应该至少包含一个非字母字符。
- 当系统提示您输入新密码时，指定密码。

- 5 使用以下命令创建一个本地证书授权机构 (Certificate Authority, CA) 样例：

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

- a. 当系统提示您输入 0-7 以表示证书类型时，键入表示 SSL CA 的 5。再次出现提示时，指定 9。
- b. 在被询问 "Is this a critical extension [y/n]?" 时，指定 "y"。

- 6 使用上面的样例 CA 生成证书

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

- a. 当系统提示您输入 0-7 以表示证书类型时，键入表示 SSL 服务器的 1。再次出现提示时，指定 9。
- b. 在被询问 "Is this a critical extension [y/n]?" 时，指定 "y"。

- 7 创建 HTTPS 侦听器，如以下步骤所述：

- a. 登录到 Web 服务器的管理服务器。
- b. 选择一个服务器，然后单击“管理”。
- c. 单击“添加侦听套接字”。在“添加侦听套接字”页面中，执行以下操作：
 - i. 指定端口号。
 - ii. 确保在“服务器名”中指定服务器的全限定域名 (fully qualified domain name, FQDN)。例如，如果主机名是 `machine1`，域名是 `server.example.com`，则 FQDN 是 `machine1.server.example.com`。
 - iii. 从“安全性”下拉列表中选择“启用”。

- iv. 单击“确定”。
- d. 转到“编辑侦听套接字”页面，然后选择刚创建的侦听套接字。
- e. 在“侦听套接字”页面中，验证服务器证书名是否与在步骤 6 中提供的证书名相同。

▼ 为 Sun Java System Web Server 6.1 导出和导入 DAS 证书

- 1 如果使用的是 Application Server 9.1，请通过执行以下命令导出 DAS 证书：

```
<appserver_install_dir>/lib/upgrade/pk12util -d <domain root>/config -o sjsas.p12-W
<file password> -K <master password> -n slas
```

- 如果使用的是 GlassFish v2，必须使用以下命令导出 DAS 证书：

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

其中，<GLASSFISH_HOME> 表示 Application Server 安装目录，<DOMAIN_NAME> 表示要导出其证书的域。

- 将证书文件复制到 Web 服务器配置目录。

- 2 如果使用的是 Application Server 9.1，请使用以下命令将 DAS 证书导入到 Web Server 实例中：

```
<webserver_install_dir>/bin/https/admin/bin/pk12util -i sjsas.p12 -d
<webserver_install_dir>/alias -W<file password> -K <webserver security db password> -P
<instance-name>-<hostname>-
```

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

这些命令将使 Application Server CA 成为受信任的 CA 以同时签署客户机证书和服务器证书。

- 如果使用的是 GlassFish v2，请从使用 NSS 安全工具 certutil 创建的 rfc 文件导入 DAS 证书。

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

可以使用以下命令检查此证书是否存在，将列出 s1as 证书以及其他 CA 证书（包括默认的服务器证书）。确保在单行中键入命令。

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

- 3 如果 obj.conf 不包含以下行，请在文件结尾附加它们。如果使用的是 Application Server 9.1，安装程序将自动执行此步骤。

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

- 4 可以使用在第 114 页中的“验证安装”一节中提供的步骤从 DAS 验证上面的设置。可以使用任何其他 CA 和服务证书，而不是使用本地 CA。在这种情况下，可以跳过上一节中列出的步骤 5 和 6，但是需要导入从其他 CA 获得的服务器证书。

在 SSL 模式下为 Web Server 7 设置负载均衡器

1. 使用以下命令启动 Web Server 的管理服务器。

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

2. 创建 HTTPS 侦听器，如以下步骤所述。如果 HTTP 侦听器已存在，则可以跳过以下步骤，并前进到第 105 页中的“为 Sun Java System Web Server 7 导出和导入 DAS 证书”一节。

- a. 登录到 Web Server 管理控制台。
- b. 选择默认配置。通常，默认配置名称将与主机名完全相同。要从“日常任务”页面执行此操作，请从“选择配置”列表中选择配置并单击“编辑配置”。或者，打开“配置”页面，单击“配置”表中的默认配置名称。
- c. 如果显示的是“日常任务”页面，请单击“请求服务器证书”。否则，如果显示的是“配置”页面，请打开“证书”页面，并单击“服务器证书”表中的“请求”按钮。这是为此默认配置创建自签名服务器证书所必需的。
- d. 提供“请求服务器证书”窗口要求的详细信息。

执行此操作时，只需确保为“*服务器名(cn)”提供的值是安装 Web 服务器的计算机的全限定域名 (fully qualified domain name, FQDN) 即可。例如，如果主机名是 machine1，域名是 server.example.com，则 FQDN 是 machine1.server.example.com。选择所提供的任何默认值。

也可以使用以下命令创建自签名证书。确保在单行中键入命令。


```

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

```

- e. 返回到所选的配置页。
- f. 打开“HTTP 侦听器”页面，然后单击“新建”按钮。这将创建启用 SSL 的 HTTP 侦听器。
- g. 提供“新建 HTTP 侦听器”向导要求的详细信息。确保服务器名是在前面步骤中提供的 FQDN。选择“SSL”按钮，然后从“证书”列表中选择以前创建的服务器证书。例如，`cert-machine1.server.example.com`。

也可以使用以下命令创建 HTTP 侦听器。确保在单行中键入每个命令。

```

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

```

```

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

```

- h. 执行上面列出的步骤后，将在管理控制台的右上角看到警报“部署暂挂”。单击它，并按照说明完成部署。此步骤确保将在 Web 服务器的管理服务器中对配置存储进行的更改复制到 Web 服务器实例。

▼ 为 Sun Java System Web Server 7 导出和导入 DAS 证书

通过导出和导入 DAS 证书，可以使 DAS 成为 Web Server 的受信任客户机。使用 DAS 证书的客户机验证可确保只有 DAS 作为受信任客户机连接到 Web Server。

- 1 打开终端窗口，并使用以下命令设置 `LD_LIBRARY_PATH`：

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

- 2 如果使用的是 **Application Server 9.1**，请通过执行以下命令导出 DAS 证书。DAS 证书同时用作服务器证书和客户机证书。

```
<appserver_install_dir>/lib/upgrade/pk12util -d <domain root>/config -o s1as.pk12 -W
<s1as.pk12-file-password> -K <master password> -n s1as
```

- 如果使用的是 **GlassFish v2**，请使用名为 **keytool** 的 **Java SE 5.0** 安全工具导出以别名 **"s1as"** 命名的 DAS 证书。执行此操作时，选择 **-rfc** 选项以可打印编码格式（如 **Internet RFC 1421** 标准所定义）导出证书。

从命令行，可以使用以下命令导出 DAS 证书：

```
<JAVA_HOME>/bin/keytool -export -rfc -alias s1as -keystore
<GLASSFISH_HOME>/domains/<DOMAIN_NAME>/config/keystore.jks-file s1as.rfc
```

其中，**<GLASSFISH_HOME>** 表示 **Application Server** 安装目录，**<DOMAIN_NAME>** 表示要导出其证书的域。

- 将证书文件复制到 **Web 服务器配置目录**。

- 3 如果使用的是 **Application Server 9.1**，请将 DAS 证书导入到 **Web Server** 实例中，并使用以下命令设置证书的信任属性：

```
<webserver_install_dir>/bin/pk12util -i <path_to_s1as.pk12-file>
-d <webserver_install_dir>/admin-server/config-store/<default-config-name>/config
-K <webserver security db password> -W <s1as.pk12-file-passwd>
```

```
<webserver_install_dir>/bin/certutil -M -n s1as -t "TCu"
-d <webserver_install_dir>/admin-server/config-store/<default-config-name>/config
```

这些命令将使 **Application Server CA** 成为受信任的 CA 以同时签署客户机证书和服务器证书。

- 如果使用的是 **GlassFish v2**，请从使用 **NSS** 安全工具 **certutil** 创建的 **rfc** 文件导入 DAS 证书。

```
<webserver_install_dir>/bin/certutil -A -a -n s1as -t "TCu" -i s1as.rfc -d
<webserver_install_dir>/admin-server/config-store/<CONFIG_NAME>/config
```

其中，**<webserver_install_dir>** 表示 **Web 服务器** 安装目录，**<CONFIG_NAME>** 表示为默认 **Web 服务器** 实例创建的配置名称。

可以使用以下命令检查此证书是否存在，将列出 **s1as** 证书以及其他 **CA** 证书（包括默认的服务器证书）。确保在单行中键入整个命令。

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

也可以使用 **Web Server** 管理控制台对此进行查看。选择已向其导入证书的配置（本例中为默认配置），然后选择“证书”选项卡。要查看所有的可用证书，请选择“证书颁发机构”子选项卡。

- 4 如果使用的是 GlassFish v2，请对 Web Server 7 进行以下配置更改。如果使用的是 Application Server 9.1，可以跳到下一步。
 - a. 在位于


```
<WS_INSTALL_ROOT>/admin-server/config-store/<DEFAULT_CONFIG_NAME>/config/
obj.conf 文件中附加以下行。确保在键入这些行时没有包含任何结尾空格。
<Object ppath="*lbconfigupdate*">
  PathCheck fn="get-client-cert" dorequest="1" require="1"
</Object>
<Object ppath="*lbgetmonitordata*">
  PathCheck fn="get-client-cert" dorequest="1" require="1"
</Object>
```
- 5 部署该配置。进行前面步骤中列出的更改时，管理控制台将此配置标记为要部署。
 - a. 在 Web Server 管理控制台中选择“部署暂挂”的图标。也可以使用 CLI 实用程序 wadm 部署此配置，如下所示：


```
<webservice_install_dir>/bin/wadm deploy-config --user=<admin> <DEFAULT_CONFIG_NAME>
```
- 6 从 GlassFish DAS 测试此设置，以查看它是否通过 SSL 与已配置的 HTTP 负载平衡器进行通信。有关更多信息，请参见第 114 页中的“验证安装”。

使用 Apache Web Server

与 Application Server 9.1 捆绑在一起的负载平衡器插件支持 Apache Web Server 2.0.x。要使用 Apache Web Server，必须在安装负载平衡器插件前后执行某些配置步骤。负载平衡器插件安装也将对 Apache Web Server 进行其他修改。安装插件后，您必须执行其他配置步骤。

注 - 如果使用 `--with-mpm=worker` 选项进行编译，则 Apache 2 将具有多线程行为。

- 第 108 页中的“对使用 Apache Web Server 的要求”
- 第 109 页中的“在安装负载平衡器插件之前配置 Apache”
- 第 112 页中的“负载平衡器插件安装程序所做的修改”
- 第 113 页中的“在安装负载平衡器插件之后配置 Apache”
- 第 114 页中的“在 Solaris 和 Linux 上启动 Apache”

对使用 Apache Web Server 的要求

对于 Apache Web Server，您的安装必须满足最低要求。

使用 Apache 时，负载均衡器插件有以下要求：

- openssl-0.9.7e (源代码)
- httpd-2.0.59 (源代码)
- gcc-3.3-sol9-sparc-local 软件包 (适用于 Solaris 9 SPARC)
- gcc-3.3-sol9-intel-local 软件包 (适用于 Solaris 9 x86)
- 预先安装的 gcc (适用于 Solaris 10)
- flex-2.5.4a-sol9-sparc-local 软件包 (适用于 Solaris 9 SPARC)
- flex-2.5.4a-sol9-intel-local 软件包 (适用于 Solaris 9 x86)
- 预先安装的 flex (适用于 Solaris 10)

这些软件源可以从 <http://www.sunfreeware.com> 获得

此外，在编译 Apache 之前，请执行以下步骤：

- 在 Linux 平台上，在同一台计算机上安装 Sun Java System Application Server。
- 在 Solaris 9 操作系统上，使用 pkgadd 安装 gcc 和 flex。请注意，pkgadd 需要超级用户权限。
- 在 Solaris 9 操作系统上，请确保 gcc 版本 3.3 和 make 位于 PATH 中，并且已安装 flex。
- 在 Solaris 10 操作系统上，运行 OpenSSL 的 make 之前，先运行位于 /usr/local/lib/gcc-lib/sparc-sun-solaris2.9/3.3/install-tools (在 Solaris SPARC 上) 或 /usr/local/lib/gcc-lib/i386-pc-solaris2.9/3.3/install-tools (在 Solaris x86 上) 中的 mkheaders。
- 如果您在 Red Hat Enterprise Linux Advanced Server 2.1 上使用 gcc，则 gcc 的版本必须在 3.0 以上。

注 - 要使用 gcc 以外的 C 编译器，请在 PATH 环境变量中设置 C 编译器和 make 实用程序的路径。

应用 Apache Web Server 修补程序

为 Apache 安装负载均衡器插件之前，应用 Apache Web Server 问题 12355 的修补程序。有关此问题的更多详细信息可以从

http://issues.apache.org/bugzilla/show_bug.cgi?id=12355 获得。“自动应用”功能需要此修补程序才能工作。要应用该修补程序，请按照以下步骤操作。

1. 解压缩 http-2.0.59.tar 并转到目录 httpd-2.0.59。
2. 从 <http://issues.apache.org/bugzilla/attachment.cgi?id=16495> 下载该修补程序并将它另存为一个文件，例如 12355.diff。

3. 从目录 `httpd-2.0.59/modules/ssl` 运行以下命令：

```
patch < 12355.diff
```

在安装负载均衡器插件之前配置 Apache

必须编译并生成 Apache 源代码以使用 SSL 运行。本节介绍成功编译 Apache Web Server 以运行负载均衡器插件所需的最低要求和高级步骤。这些要求和步骤仅适用于本软件的 Solaris 和 Linux 版本。有关 Apache 的 Windows 版本的信息，请参见 Apache Web 站点。

注 - 此处包含的说明摘自 <http://httpd.apache.org/docs> 中的说明。有关安装 SSL 可识别 Apache 的详细说明，请参见此 Web 站点。

▼ 安装 SSL 可识别 Apache

开始之前 您必须已下载并解压缩 Apache 软件。

1 下载并解压缩 OpenSSL 源（可以从 <http://openssl.org> 获得）。

2 编译和生成 OpenSSL。

有关完整的安装说明，请参见解压缩 OpenSSL 所在目录中的名为 `INSTALL` 的文件。此文件包含有关将 OpenSSL 安装到用户指定位置的信息。

有关 OpenSSL 的更多信息，请参见 <http://www.openssl.org/>。

3 下载并解压缩 Apache。

Apache 可以从 <http://httpd.apache.org> 获得。

4 编译并生成 Apache。配置源代码树：

a. `cd http-2.0_x`。

b. 运行以下命令：

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

在以上命令中，*x* 是 Apache 的版本号，*openssl-install-path* 是 OpenSSL 安装目录的绝对路径，*Apache-install-path* 是安装 Apache 的目录。

请注意，如果 Apache 2 服务器要接受 HTTPS 请求，则只需使用 `--enable-ssl` 和 `--enable-so` 选项。

5 对于 Linux 2.1 上的 Apache，编译之前请执行以下操作：

a. 打开 `src/Makefile` 并找到自动生成部分的结尾处。

b. 在自动生成部分后的前四行之后添加以下行：

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

请注意，仅在 Application Server 作为 Java Enterprise System 的一部分进行安装时，`-L/opt/sun/private/lib` 才是必需的。

例如：

```
## (End of automatically generated section)
##
CFLAGS=$(OPTIM) $(CFLAGS1) $(EXTRA_CFLAGS)
LIBS=$(EXTRA_LIBS) $(LIBS1)
INCLUDES=$(INCLUDES1) $(EXTRA_INCLUDES)
LDFLAGS=$(LDFLAGS1) $(EXTRA_LDFLAGS)
"LIBS+= -licuuc -licui18n -lnspr4 -lpthread
-lxerces-c -lsupport -lnsprwrap -lns-httpd40
LDFLAGS+= -L/application-server-install-dir /lib -L/opt/sun/private/lib
```

c. 设置环境变量 `LD_LIBRARY_PATH`。

进行独立安装时，将其设置为 Application Server: `as-install/lib`

进行 Java Enterprise System 安装时，将其设置为 Application Server: `as-install/lib:opt/sun/private/lib`。

如果使用的是 Solaris 9，请将 `/usr/local/lib` 添加到 `LD_LIBRARY_PATH` 中。

6 按照您所用版本的安装说明编译 Apache。

有关更多信息，请参见 <http://httpd.apache.org/>

通常，步骤如下：

a. `make`

b. `make install`

7 确保 Apache 的 `ssl.conf` 和 `httpd.conf` 文件包含正确的环境值。

- 在 `ssl.conf` 中，对于 `VirtualHost default:port`，将默认主机名和端口分别替换为安装有 Apache 的本地系统的主机名以及服务器的端口号。

如果不进行此更改，则负载均衡器将无法正常工作。在 Solaris 上，Apache 可能无法启动；在 Linux 上，HTTPS 请求可能无法执行。

- 在 `ssl.conf` 中，对于 `ServerName www.example.com:443`，将 `www.example.com` 替换为安装有 Apache 的本地系统的主机名。

如果不进行此更改，则会在安装了安全性证书的情况下启动 Apache 时，出现以下警告：

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

有关安装 Apache 证书的更多信息，请参见第 113 页中的“为 Apache 创建安全性证书”。

- 在 `httpd.conf` 中，对于 `ServerName www.example.com:80`，将 `www.example.com` 替换为安装有 Apache 的本地系统的主机名。

如果不进行此更改，则会在启动 Apache 时出现以下警告：系统无法确定服务器的全限定域名，并且存在重叠的 `VirtualHost` 条目。

8 确保 Apache 用户具有对 `apache-install-location/conf/` 目录以及此目录中文件的必需访问权限。

Apache 用户是 Apache 服务器响应请求所用的 UNIX 用户。此用户在文件 `httpd.conf` 中定义。

如果以超级用户身份安装了 Apache，请阅读 `apache-install-location/conf/httpd.conf` 中有关配置 Apache 用户和组的说明。

注 - 确保您的用户和组配置满足此目录的安全要求。例如，要限制对此目录的访问，请将 Apache 用户添加到与目录所有者相同的用户组。

- a. 要确保“自动应用”功能正常工作，请授予 Apache 用户对 `apache-install-location/conf/` 目录的读取访问权限、写入访问权限和执行访问权限。
 - 如果 Apache 用户与此目录的所有者在相同的组中，请将模式更改为 `775`。
 - 如果 Apache 用户与此目录的所有者在不同的组中，请将模式更改为 `777`。
- b. 要确保在启动 Apache 时初始化负载均衡器插件，请授予 Apache 用户对以下文件的读取访问权限和写入访问权限：
 - `apache-install-location/conf/loadbalancer.xml`
 - `apache-install-location/conf/sun-loadbalancer_1_2.dtd`

导出和导入 DAS 证书

必须使用以下命令手动导出 DAS 证书：

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

在安装负载均衡器插件时，将需要此证书。

Application Server 9.1 安装程序将为您执行以下任务。

- 通过将 `sjsas.crt` 复制到 `apache-install-dir/conf/ssl.crt` 目录导入 DAS 证书。
- 将以下行附加到 `httpd.conf`。

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

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

负载均衡器插件安装程序所做的修改

负载均衡器插件安装程序会将所需的文件提取到 Web 服务器根目录下的 `modules` 目录中：

它将以下条目添加到 Web 服务器实例的 `httpd.conf` 文件中：

```
##BEGIN EE LB Plugin Parameters
LoadModule apachelbplugin_module modules/mod_loadbalancer.so
#AddModule mod_apachelbplugin.cpp
<IfModule mod_apachelbplugin.cpp>
    config-file webserv-instance/httpd/conf/loadbalancer.xml
```



```

    locale en
</IfModule>
<VirtualHost machine-ip-address>
    DocumentRoot "webserv-instance/httpd/htdocs"
    ServerName server-name
</VirtualHost>
##END EE LB Plugin Parameters

```

在安装负载均衡器插件之后配置 Apache

Apache Web Server 必须具有正确的安全文件才能使用负载均衡器插件。负载均衡器依赖于需要这些安全数据库文件的 NSS（Network Security Service，网络安全服务）库。您需要从 Application Server 获取这些安全数据库文件，因此必须将 Application Server 安装在 Web Server 可以访问的位置。

要将 Apache 安全文件配置为使用负载均衡器，请执行以下操作：

将 `/usr/lib/mps` 附加到 `Apache-install-dir/bin/apachectl` 脚本中的 `LD_LIBRARY_PATH`。

▼ 为 Apache 创建安全性证书

为了在 Apache 上支持 HTTPS 请求，需要执行以下步骤。

有关在 Apache 上设置安全性证书的详细信息，请参见 http://httpd.apache.org/docs/2.2/ssl/ssl_faq.html 和 http://www.modssl.org/docs/2.8/ssl_faq.html 中的说明。以下过程摘自这些 Web 站点。

1 设置以下环境变量：

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

2 通过执行以下命令，创建服务器证书和密钥：

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

当系统要求提供通用名称时，请提供计划运行 Apache 所用的主机名。对于所有其他提示，请输入满足您所有特定要求的值。

此命令会创建 `newreq.pem`。

3 从运行 `openssl` 命令的位置打开新创建的 `newreq.pem`。

4 复制以 `BEGIN CERTIFICATE` 开始并以 `END CERTIFICATE` 结束的行，并将其粘贴在 `Apache-install-dir /conf/ssl.crt/server.crt` 中。例如：

```

-----BEGIN CERTIFICATE-----
....

```

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

- 5 复制以 **BEGIN RSA PRIVATE KEY** 开始并以 **END RSA PRIVATE KEY** 结束的行，并将其粘贴在 *Apache-install-dir* /conf/ssl.key/server.key 中。例如：

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

- 6 确保 *Apache-install-dir* /conf/ssl.conf 中的变量 `SSLCertificateKeyFile` 和 `SSLCertificateFile` 具有正确的值。
- 7 确保 `ServerName` 并非为 `www.example.com`。`ServerName` 应当为运行 Apache 所用的实际主机名，从而与您创建服务器证书和密钥时输入的通用名称相匹配。

修改 httpd.conf 参数以启用粘性 round robin

为了使粘性 round robin 功能正常工作，请在 `httpd.conf` 文件中的 `prefork MPM` 部分下，确保参数 `StartServers` 和 `maxclients` 的值设置为 1。否则，每个新的会话请求都会产生一个新的 Apache 进程，而且将初始化负载均衡器插件，从而导致请求停放在同一实例中。

在 Solaris 和 Linux 上启动 Apache

通常，应当使用安装 Application Server 时所用的用户身份启动 Apache。在以下情况下，您必须以超级用户身份启动 Apache：

- 如果您是 Java Enterprise System 用户。
- 如果您使用的端口号小于 1024。
- 如果 Apache 运行时所用的用户身份不同于启动时所用的用户身份。

要以 SSL 模式启动 Apache，请使用以下命令之一：

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

如果需要，请查阅 Apache Web 站点以获得有关启动 Apache Server 的最新信息。

验证安装

1. 安装负载均衡器插件。有关安装该插件的详细步骤，请参见《Sun Java System Application Server 9.1 Installation Guide》。在安装过程中，提供指向 DAS 证书的路径。

2. 登录到 Application Server 管理控制台，并创建一个新群集。有关创建新群集的步骤，请参阅管理控制台联机帮助。
3. 创建新的 HTTP 负载均衡器。创建负载均衡器时，请将 Web 服务器主机的 FQDN 指定为设备主机名，将 Web 服务器 SSL 端口指定为设备端口，并选择上一步中创建的群集作为目标。有关创建新的 HTTP 负载均衡器的详细步骤，请参阅管理控制台联机帮助。
4. 要验证 DAS 和 Web 服务器之间的通信是否正常进行，请在管理控制台中，导航到“HTTP 负载均衡器”节点，然后单击“HTTP 负载均衡器”。在出现的“负载均衡器设备设置”页面中，按“测试连接”按钮。
如果在创建负载均衡器时未启用“自动应用更改”选项，则必须通过转到“导出”选项卡并单击“立即应用更改”手动导出负载均衡器配置。
5. 如果测试连接失败，请务必检查 Application Server 域日志和 Web 服务器日志以解决问题。此外，检查是否正确执行了所有配置步骤。

使用 Microsoft IIS

要将 Microsoft Internet Information Services (IIS) 与负载均衡器插件一起使用，请按照以下各节中提供的步骤操作。

▼ 配置 Microsoft IIS 以使用负载均衡器插件

- 1 打开 Internet 服务管理器。
- 2 选择要为其启用此插件的 Web 站点。
此 Web 站点通常名为“默认 Web 站点”。
- 3 在此 Web 站点上单击鼠标右键，并选择“属性”以打开“属性”记事本。
- 4 按照以下步骤，添加一个新的 ISAPI 过滤器：
 - a. 打开“ISAPI 过滤器”选项卡。
 - b. 单击“添加”。
 - c. 在“过滤器名称”字段中，输入 Application Server
 - d. 在“可执行文件”字段中，键入
`C:\Inetpub\wwwroot\sun-passthrough\sun-passthrough.dll`。
 - e. 单击“确定”关闭“属性”记事本。

- 5 创建并配置新的虚拟目录：
 - a. 在默认的 Web 站点上单击鼠标右键，选择“新建”，然后选择“虚拟目录”。
将打开“虚拟目录创建向导”。
 - b. 在“别名”字段中，键入 `sun-passthrough`。
 - c. 在“目录”字段中，键入 `C:\Inetpub\wwwroot\sun-passthrough`。
 - d. 选中“执行权限”复选框。
使与权限相关的所有其他复选框保持未选中状态。
 - e. 单击“完成”。
- 6 将 `sun-passthrough.dll` 文件的路径、Application Server `as-install/bin` 以及 Application Server `as-install/lib` 添加到系统的 PATH 环境变量中。
- 7 对于 IIS 6.0 用户，请使用以下步骤将负载平衡器 Web 服务扩展配置为在 IIS 6 中运行：
 - a. 在 IIS 管理器中，展开本地计算机，并单击“Web 服务扩展”。
 - b. 在“任务”窗格中，选择“添加一个新的 Web 服务扩展”。
 - c. 将扩展的名称输入为 `Sun-Passthrough` 并单击“添加”。
 - d. 键入 `sun-passthrough.dll` 的路径 `C:\Inetpub\wwwroot\sun-passthrough`。
 - e. 单击“确定”。
 - f. 选择“设置扩展状态为允许”。
- 8 对于 IIS 6.0 用户，请创建文件 `C:\inetpub\wwwroot\sun-passthrough\lb.log`，并为组 IIS_WPG 提供对该文件的 NTFS 写入和修改权限。
由于 IIS 6.0 以“工作进程隔离模式”运行，因此它会使用组 IIS_WPG 的安全权限运行 IIS 服务器。
- 9 对于所有 IIS 用户，请重新启动计算机。
- 10 检验 Web 服务器、负载平衡器插件和 Application Server 是否运行正常。
在 Web 浏览器中键入以下地址，以访问此 Web 应用程序的上下文根：
`http://web-server-name/web-application`，其中 `web-server-name` 是 Web 服务器的主机

名或 IP 地址，*web-application* 是在 C:\Inetpub\wwwroot\sun-passthrough\sun-passthrough.properties 文件中列出的上下文根。

提示 - ISAPI 过滤器状态应当为绿色。要检查过滤器状态，请访问此 Web 站点的 "Properties" 笔记簿，并单击 "ISAPI Filters" 选项卡。如果状态不为绿色，请尝试向 IIS HTTP 端口发送任一 HTTP 请求。如果请求失败，则表明情况确实如此。请重新检查 ISAPI 过滤器的状态。

自动配置的 sun-passthrough 属性

安装程序会在 sun-passthrough.properties 中自动配置以下属性。您可以更改默认值。

属性	定义	默认值
lb-config-file	负载均衡器配置文件的路径	IIS-www-root\sun-passthrough\loadbalancer.xml
log-file	负载均衡器日志文件的路径	IIS-www-root\sun-passthrough\lb.log
log-level	Web 服务器的日志级别	INFO

注 - IIS 当前不支持 Application Server 9.1 的“自动应用”功能。

配置 HTTP 负载均衡

本章介绍了 HTTP 负载均衡器插件。其中包括以下主题：

- 第 119 页中的“负载均衡器插件的新增功能”
- 第 121 页中的“HTTP 负载均衡器如何工作”
- 第 122 页中的“设置 HTTP 负载均衡”
- 第 126 页中的“配置负载均衡器”
- 第 139 页中的“配置多个 Web 服务器实例”
- 第 139 页中的“升级应用程序而不使可用性受到损失”
- 第 145 页中的“监视 HTTP 负载均衡器插件”

有关其他类型负载均衡的信息，请参见第 10 章和第 11 章。

本节讨论如何使用 Application Server 随带的 HTTP 负载均衡插件。另一个 HTTP 负载均衡选项是将 Sun Secure Application Switch 和 Application Server 一起使用，以作为基于硬件的负载均衡解决方案。有关配置此解决方案的教程，请参见文章 [Clustering and Securing Web Applications: A Tutorial \(http://developers.sun.com/prodtech/appserver/reference/techart/load-balancing.html\)](http://developers.sun.com/prodtech/appserver/reference/techart/load-balancing.html)。

负载均衡器插件的新增功能

在 Sun Java System Application Server 9.1 中，增强了负载均衡器的功能，可通过以下特性提供更大的灵活性和更高的易用性。

自动应用

Application Server 允许将从管理控制台对负载均衡器配置进行的更改通过线路自动发送到 Web Server 配置目录。在 Application Server 的以前版本中，必须导出负载均衡器配置，然后将其复制到 Web 服务器配置目录。

加权 Round Robin

负载均衡器启用了改进的 HTTP 请求分发。管理员可以使用名为 "weight" 的属性，指定如何将请求按比例地路由到实例。例如，假定某个群集具有两个实例，管理员为实例 x 指定的权重为 100，为实例 y 指定的权重为 400。现在，对于每 100 个请求，20 个将转到实例 x，80 个将转到实例 y。

用户定义的负载均衡

通过 Application Server，管理员可以定义用于分发 HTTP 请求的自定义策略。自定义策略定义负载均衡器插件必须使用的负载均衡算法。换句话说，管理员可以定义哪个 Application Server 实例将处理 HTTP 请求。要使用此功能，管理员需要开发共享库。例如，开发的共享库可以用于评估提供给它的传入请求头，并遵照某些条件选择可以处理请求的实例。此共享库将由负载均衡器装入。

共享库必须实现一个接口，如 `loadbalancer.h`（可在 `appserver_install_dir/lib/install/templates` 下找到）中所定义。

Application Server 还捆绑了一个实现基本 round robin 算法的样例模块 `roundrobin.c`。管理员可以使用此样例模块作为模板来生成共享库。此样例模块也可在 `appserver_install_dir/lib/install/templates` 下找到。

▼ 配置用户定义的负载均衡

- 1 将 `roundrobin.c` 从 `appserver_install_dir/lib/install/templates` 复制到工作目录（例如：`/home/user/workspacelb`）。
- 2 使用 ANSI C/C++ 编译器（例如 Sun Studio 编译器或 GCC）编译 `roundrobin.c`。务必生成动态共享库，而不是生成静态可执行文件。

a. 如果使用的是 Sun Studio CC 编译器，请通过以下命令进行编译：

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

b. 如果使用的是 GCC，请通过以下命令编译共享库：

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

注 – 如果遇到重定位错误，请使用选项 "-fPIC" 重新编译。命令将与如下所示类似：

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

在 Microsoft Windows 上，从 <http://www.redhat.com/services/custom/cygwin> 下载 Cygwin 实用程序。此实用程序已将 GCC 与自身捆绑在一起。使用以下 GCC 命令创建动态链接库 (dynamic link library, dll)：

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

- 3 将 `loadbalancer.xml` 更改为指向新生成的模块。 `loadbalancer.xml` 在编辑后将如下所示。


```
<cluster name="cluster1" policy="user-defined"
policy-module="home/user/workspace1b/roundrobin.so">
```
- 4 将 `roundrobin.so` 复制到 Web 服务器实例目录。
- 5 启动 Web 服务器（如果它未运行），或者等待，直到重新配置了负载均衡器。

HTTP 负载均衡器如何工作

负载均衡器尝试在多个 Application Server 实例（独立或群集）之间平均分配工作量，从而提高系统的整体吞吐量。

HTTP 负载均衡器对 Java EE 应用服务器上部署的服务启用高可用性。在这样做时，如果检测到原始的处理实例因不可用或异常而无法处理请求，它会将会话请求故障转移到其他服务器实例。为了使 HTTP 会话信息能够保留下来，您必须使用群集配置文件，已安装和设置 HADB，并已配置 HTTP 会话持久性。有关更多信息，请参见第 9 章。

注 – 负载均衡器不能处理大于 8k 的 URI/URL。

HTTP 负载均衡算法

默认情况下，Sun Java System Application Server 负载均衡器使用粘性 *round robin* 算法对收到的 HTTP 和 HTTPS 请求进行负载均衡。

新的 HTTP 请求发送到负载均衡器插件时，系统将基于简单的 *round robin* 方案将该请求转发到某个应用服务器实例。如果请求用于基于会话的应用程序，则这还包括对新会话的请求。来自同一客户机的对同一基于会话的应用程序的后续请求被认为是已分

配的请求或粘性请求，并由负载均衡器路由到同一实例。粘性 round robin 由此得名。对非基于会话的应用程序的请求和对基于会话的应用程序的第一个请求称为未分配的请求。粘性是通过使用 Cookie 或显式 URL 重写实现的。负载均衡器会自动确定粘性方法。

负载均衡器插件使用以下方法来确定会话粘性：

- **Cookie 方法**：负载均衡器插件使用一个单独的 Cookie 来记录路由信息。HTTP 客户机（通常为 Web 浏览器）必须支持 Cookie 才能使用基于 Cookie 的方法。如果 HTTP 客户机无法接受 Cookie，则插件使用以下方法。
- **显式 URL 重写**：粘性信息将被附加至 URL。即使 HTTP 客户机不支持 Cookie，也可以使用此方法。

从粘性信息中，负载均衡器插件将首先确定请求先前被转发到的实例。如果发现该实例工作正常，负载均衡器插件会将请求转发至该特定应用程序服务器实例。因此，给定会话的所有请求都将被发送到同一个应用程序服务器实例。

设置 HTTP 负载均衡

本节介绍了如何设置负载均衡器插件，并且包含以下各节：

- 第 122 页中的“设置负载均衡的先决条件”
- 第 123 页中的“设置负载均衡的步骤”
- 第 126 页中的“HTTP 负载均衡器部署”

设置负载均衡的先决条件

配置负载均衡器之前，您必须执行以下操作：

- 安装支持的 Web 服务器并对其进行配置。有关配置支持的 Web 服务器的更多信息，请参见第 4 章。
- 安装负载均衡器插件。
有关安装过程的信息，请参见《Sun Java System Application Server 9.1 Installation Guide》。
- 创建 Application Server 群集或服务器实例以参与负载均衡。
- 将应用程序部署到这些群集或实例。

注 – 如果具有在不同网络域上安装 Application Server 实例和负载均衡器的部署方案，必须通过使用选项 `--agentproperties` 指定全限定域名来创建节点代理。例如，`asadmin create-node-agent --agentproperties remoteclientaddress=machine1.server.example.com test-na`。有关此命令的更多信息，请参见 `create-node-agent(1)`。

设置负载均衡的步骤

使用管理控制台 GUI 或 `asadmin` 工具在您的环境中配置负载均衡。以下各节为您提供更多信息。

▼ 使用管理控制台设置负载均衡

1 创建负载均衡器配置。

在管理控制台的左框架上，单击“HTTP 负载均衡器”，然后单击“新建”。在“新建 HTTP 负载均衡器”页面中，提供设备详细信息，并选择目标群集或实例。

2 添加对群集或独立服务器实例的引用，以供负载均衡器进行管理。

要使用管理控制台执行此操作，请在左框架上，单击“HTTP 负载均衡器”节点，然后单击该节点下列出的所需负载均衡器。打开“目标”选项卡，单击“管理目标”，然后在“管理目标”页面中选择所需的目标。

如果您创建了具有目标负载均衡器配置，并且该目标是负载均衡器引用的唯一群集或独立服务器实例，则请跳过此步骤。

3 启用负载均衡器所引用的群集或独立服务器实例。

要使用管理控制台启用独立服务器实例，请在左框架上，单击“HTTP 负载均衡器”节点，然后单击该节点下列出的所需负载均衡器。打开“目标”选项卡，在“目标”表中单击要启用的实例旁边的复选框，然后单击“启用”。

要启用群集中的服务器实例，请选择负载均衡器（如上所述），然后在“目标”选项卡中单击所需的群集。现在，打开“实例”选项卡，选择所需的实例，然后从“负载均衡器操作”下拉列表中选择“启用负载均衡”。

启用群集或独立实例的等效命令是 `asadmin enable-http-lb-server`。

4 启用用于负载均衡的应用程序。

要使用管理控制台执行此操作，请打开“目标”选项卡（如上所述），然后单击所需的群集。现在，打开“应用程序”选项卡，选择所需的应用程序，然后从“更多操作”下拉列表中选择“负载均衡器启用”。

这些应用程序必须已部署到负载均衡器所引用的群集或独立实例上，并且已启用，可以在群集或独立实例上使用。启用应用程序以用于负载均衡与启用以使用这些应用程序是两个独立的步骤。

5 创建运行状况检查器。

要使用管理控制台执行此操作，请打开负载均衡器的“目标”选项卡（如上一步所述），然后在“目标”表中单击“编辑运行状况检查器”。

运行状况检查器监视工作异常的服务器实例，以便在这些服务器实例重新正常工作时，负载均衡器可以向它们发送新请求。

注 - 如果使用的是 Sun Java System Web Server (6.1 或 7.0)，则可以在单个步骤中生成负载均衡器配置文件并通过线路将数据发送到 Web Server，而不是执行步骤 6 和 7。

要使用管理控制台执行此操作，请单击所需的负载均衡器，然后打开“导出”选项卡。在此选项卡中，单击“立即应用更改”。这会将数据发送到 Web Server 配置目录。

6 生成负载均衡器配置文件。

要使用管理控制台执行此操作，请单击负载均衡器，然后打开“导出”选项卡。在此选项卡中，单击“立即导出”。

此命令将生成一个配置文件，该配置文件要与 Sun Java System Application Server 附带的负载均衡器插件一起使用。

7 将负载均衡器配置文件复制到 Web Server 的 config 目录中，此目录用于存储负载均衡器插件的配置文件。

注 - 要在单个步骤中自动生成负载均衡器配置文件并通过线路将数据发送到 Web Server，需要为 SSL 设置配置 Web 服务器并导入 DAS 证书。有关配置 Sun Java System Web Server 的信息，请参见第 100 页中的“配置 Sun Java System Web Server”。

▼ 使用 asadmin 工具设置负载均衡

1 创建负载均衡器配置。

为此，请使用命令 `asadmin create-http-lb-config`。

注 - 可以使用单个 `asadmin` 命令 `create-http-lb` 及其选项执行所有后续步骤（步骤 2 到步骤 7）。有关此命令的更多信息，请参见 `create-http-lb(1)`。

2 添加对群集或独立服务器实例的引用，以供负载均衡器进行管理。

为此，请使用命令 `asadmin create-http-lb-ref`。有关此命令的更多信息，请参见 `create-http-lb-ref(1)`。

如果您创建了具有目标的负载均衡器配置，并且该目标是负载均衡器引用的唯一群集或独立服务器实例，则请跳过此步骤。

3 启用负载均衡器所引用的群集或独立服务器实例。

为此，请使用命令 `asadmin enable-http-lb-server`。有关此命令的更多信息，请参见 `enable-http-lb-server(1)`。

4 启用用于负载均衡的应用程序。

为此，请使用命令 `asadmin enable-http-lb-application`。有关此命令的更多信息，请参见 `enable-http-lb-application(1)`。

这些应用程序必须已部署到负载均衡器所引用的群集或独立实例上，并且已启用，可以在群集或独立实例上使用。启用应用程序以用于负载均衡与启用以使用这些应用程序是两个独立的步骤。

5 创建运行状况检查器。

为此，请使用命令 `asadmin create-http-health-checker`。有关此命令的更多信息，请参见 `create-http-health-checker(1)`。

运行状况检查器监视工作异常的服务器实例，以便在这些服务器实例重新正常工作时，负载均衡器可以向它们发送新请求。

注 - 如果使用 Sun Java System Web Server (6.1 或 7.0)，而不是执行步骤 6 和 7，可以在单个步骤中生成负载均衡器配置文件并通过线路将数据发送到 Web Server。

要使用 `asadmin` 工具执行此操作，请将 `create-http-lb` 命令的 `--autoapplyenabled` 选项设置为 `true`。有关此命令的更多信息，请参见 `create-http-lb(1)`。

6 生成负载均衡器配置文件。

为此，请使用命令 `asadmin export-http-lb-config`。有关此命令的更多信息，请参见 `export-http-lb-config(1)`。此命令将生成一个配置文件，该配置文件要与 Sun Java System Application Server 附带的负载均衡器插件一起使用。

7 将负载均衡器配置文件复制到 Web Server 的 `config` 目录中，此目录用于存储负载均衡器插件的配置文件。

注 - 要在单个步骤中自动生成负载均衡器配置文件并通过线路将数据发送到 Web Server，需要为 SSL 设置配置 Web 服务器并导入 DAS 证书。有关配置 Sun Java System Web Server 的信息，请参见第 100 页中的“配置 Sun Java System Web Server”。

HTTP 负载均衡器部署

根据您的目的和环境，可以使用不同方法配置负载均衡器，如以下各节所述：

- 第 126 页中的“使用群集服务器实例”
- 第 126 页中的“使用多个独立实例”

使用群集服务器实例

部署负载均衡器最常用的方法是使用服务器实例的一个或多个群集。默认情况下，群集中的所有实例都具有相同的配置，并被部署了相同的应用程序。负载均衡器在服务器实例之间分配工作量并将请求从异常实例故障转移到正常实例。如果您已配置了 HTTP 会话持久性，则对请求进行故障转移时，会话信息将保留。

如果具有多个群集，可以跨群集对请求进行负载均衡，但是仅在单个群集中的实例之间进行故障转移。在一个负载均衡器中使用多个群集可以轻易启用应用程序的滚动升级。有关更多信息，请参见第 139 页中的“升级应用程序而不使可用性受到损失”。

注 - 不能跨群集和独立实例对请求进行负载均衡。

使用多个独立实例

您还可以将负载均衡器配置为使用多个独立实例，并在这些实例之间对请求进行负载均衡和故障转移。但是，在此配置中，您必须手动确保独立实例具有同构环境和部署到其上的相同的应用程序。由于群集自动维护同构环境，因此对于大多数情况，使用群集更好、更容易。

配置负载均衡器

负载均衡器配置在 `domain.xml` 文件中维护。配置负载均衡器是极其灵活的：

- 尽管一个域可以关联多个负载均衡器，但一个负载均衡器只为一个域提供服务。
- 尽管每个负载均衡器只有一个负载均衡器配置，但每个负载均衡器配置可以关联多个负载均衡器。

以下各节更详细地介绍如何创建、修改和使用负载均衡器配置：

- 第 127 页中的“在 DAS 上配置 HTTP 负载均衡器”
- 第 128 页中的“创建 HTTP 负载均衡器引用”
- 第 128 页中的“启用用于负载均衡的服务器实例”
- 第 129 页中的“启用用于负载均衡的应用程序”
- 第 129 页中的“创建 HTTP 运行状况检查器”
- 第 131 页中的“导出负载均衡器配置文件”
- 第 132 页中的“更改负载均衡器配置”

- 第 132 页中的“启用动态重新配置”
- 第 133 页中的“禁用（停止）服务器实例或群集”
- 第 133 页中的“禁用（停止）应用程序”
- 第 134 页中的“配置 HTTP 和 HTTPS 故障转移”
- 第 135 页中的“在负载均衡器中使用重定向”
- 第 138 页中的“配置幂等 URL”

在 DAS 上配置 HTTP 负载均衡器

在 Application Server 9.1 中，可以使用管理控制台或 `asadmin` 命令 `create-http-lb` 在 DAS 上创建负载均衡器配置。以下步骤介绍了如何执行该操作。如果需要有关 `asadmin` 命令 `create-http-lb`、`delete-http-lb` 和 `list-http-lbs` 的更多信息，请参见《Sun Java System Application Server 9.1 Reference Manual》。

在管理控制台中，向下滚动左框架，单击“HTTP 负载均衡器”节点，然后在右侧的“HTTP 负载均衡器”页面中单击“新建”。在“新建 HTTP 负载均衡器”页面中，提供托管负载均衡器的计算机的以下详细信息。

字段	说明
名称	负载均衡器配置的名称。
已启用	单击“已启用”复选框将负载均衡器配置更改自动推送到驻留在 Web 服务器配置目录中的物理负载均衡器。
主机	安装了 Web 服务器实例的服务器。
管理端口	Web 服务器实例使用的管理端口号。
代理主机	安装了代理服务器实例的服务器。
代理端口	代理服务器使用的端口号。

也可以使用 `asadmin` 命令 `create-http-lb-config` 创建负载均衡器配置。表 5-1 介绍了各个参数。有关命令 `create-http-lb-config`、`delete-http-lb-config` 和 `list-http-lb-configs` 的更多信息，请参见《Sun Java System Application Server 9.1 Reference Manual》。

表 5-1 负载均衡器配置参数

参数	说明
<code>response timeout</code>	服务器实例必须返回响应的的时间（以秒为单位）。如果在该时间段内未收到任何响应，则服务器将被视为处于异常状态。默认值为 60。

表 5-1 负载均衡器配置参数 (续)

参数	说明
HTTPS routing	对负载均衡器的 HTTPS 请求是否会导致对服务器实例的 HTTPS 或 HTTP 请求。有关更多信息, 请参见第 134 页中的“配置 HTTPS 路由选择”。
reload interval	检查负载均衡器配置文件 <code>loadbalancer.xml</code> 的更改的时间间隔。当检查检测到更改时, 系统将重新装入配置文件。0 值禁用重新装入。有关更多信息, 请参见第 132 页中的“启用动态重新配置”。
monitor	是否为负载均衡器启用监视功能。
routecookie	负载均衡器插件用于记录路由信息的 Cookie 的名称。HTTP 客户机必须支持 Cookie。如果您的浏览器设置为在存储 Cookie 之前进行询问, 则 Cookie 的名称为 JROUTE。
target	负载均衡器配置的目标。指定目标的效果与向该目标添加引用的效果相同。目标可以是群集, 也可以是独立实例。

创建 HTTP 负载均衡器引用

当您在负载均衡器中创建对独立服务器或群集的引用时, 此服务器或群集将被添加到负载均衡器控制的目标服务器和群集的列表中。所引用的服务器或群集仍需要被启用, 然后才能对向其发出的请求进行负载平衡。如果创建了带有目标的负载均衡器配置, 则系统已将该目标添加为引用。

要使用管理控制台创建引用, 请在左框架上, 单击“HTTP 负载均衡器”节点, 然后单击该节点下列出的所需负载均衡器。打开“目标”选项卡, 单击“管理目标”, 然后在“管理目标”页面中选择所需的目标。也可以使用 `create-http-lb-ref` 创建引用。您必须提供负载均衡器配置名称和目标服务器实例或群集。

要删除引用, 请使用 `delete-http-lb-ref`。要删除某个引用前, 必须先使用 `disable-http-lb-server` 禁用所引用的服务器或群集。

有关这些命令的更多信息, 请参见《Sun Java System Application Server 9.1 Reference Manual》。

启用用于负载平衡的服务器实例

创建对服务器实例或群集的引用后, 请使用 `enable-http-lb-server` 来启用服务器实例或群集。如果在创建负载均衡器配置时使用了某个服务器实例或群集作为目标, 则必须启用该服务器实例或群集。要使用管理控制台执行此操作, 请在左框架上, 单击“HTTP 负载均衡器”节点, 然后单击该节点下列出的所需负载均衡器。现在, 打开“目标”选项卡, 在“目标”表中单击要启用的实例旁边的复选框, 然后单击“启用”。

有关此命令的更多信息, 请参见 `enable-http-lb-server(1)`。

启用用于负载均衡的应用程序

由负载均衡器管理的所有服务器都必须具有同构配置，包括部署到这些服务器的相同的应用程序集。部署和启用某个应用程序以便进行访问（在部署期间或部署之后发生）后，您必须启用该应用程序以进行负载均衡。如果没有为负载均衡启用应用程序，则即使已对该应用程序所部署到的服务器的请求执行了负载均衡和故障转移，也不会对该应用程序的请求执行负载均衡和故障转移。

启用应用程序时，请指定应用程序名称和目标。如果负载均衡器管理了多个目标（例如，两个群集），请在所有目标上启用该应用程序。

要使用管理控制台启用应用程序，请在左框架上，单击“HTTP 负载均衡器”节点，然后单击该节点下列出的所需负载均衡器。打开“目标”选项卡（如上所述），然后单击所需的群集。现在，打开“应用程序”选项卡，选择所需的应用程序，然后从“更多操作”下拉列表中选择“负载均衡器启用”。如果要从命令行执行此操作，可以使用命令 `asadmin enable-http-lb-application`。有关更多信息，请参见 `enable-http-lb-application(1)`。

如果部署了新的应用程序，则还必须启用该应用程序以进行负载均衡并再次导出负载均衡器配置。

创建 HTTP 运行状况检查器

负载均衡器的运行状况检查器将定期检查被标记为异常的所有已配置的 `Application Server` 实例。运行状况检查器不是必需的，但如果没有运行状况检查器，或者禁用了运行状况检查器，则不会执行异常实例的定期运行状况检查。负载均衡器不能确定异常实例变为正常的时间。

负载均衡器的运行状况检查机制使用 HTTP 与应用程序服务器实例进行通信。运行状况检查器将 HTTP 请求发送给指定的 URL 并等待响应。HTTP 响应标题中的状态码在 100 到 500 之间时表示实例处于正常状态。

注 - 如果在您的部署方案中负载均衡器是一个群集的前端，该群集具有使用启用了客户机证书验证的受保护端口的实例，则运行状况检查器将无法执行实例的运行状况检查。因此，那些实例将始终被标记为异常，且不会向它们发送请求。

创建运行状况检查器

要指定运行状况检查器属性，可以使用管理控制台或 `asadmin create-http-health-checker` 命令。要在管理控制台中执行此操作，请导航到“HTTP 负载均衡器”节点，展开它并选择负载均衡器。然后，打开“目标”选项卡，在“目标”表中，单击所需目标的“编辑运行状况检查器”链接。指定下列参数。

表 5-2 运行状况检查器参数

参数	说明	默认值
Load Balancer	单击“已启用”复选框以使所选服务器可用于负载平衡。	False/Disabled
Disable Timeout	此服务器在被禁用后到达停顿状态所用的时间（以分钟为单位）。	30 分钟
url	指定负载均衡器检查的侦听器的 URL 以确定其运行状况。	"/"
interval	指定进行实例的运行状况检查的时间间隔（以秒为单位）。指定 0 将禁用运行状况检查器。	30 秒
timeout	指定超时间隔（以秒为单位），必须在该时间间隔内获得响应才能认为侦听器运行正常。	10 秒

如果应用程序服务器实例被标记为异常，运行状况检查器将轮询异常实例以确定实例的状态是否已变为正常。运行状况检查器使用指定的 URL 来检查所有异常的应用程序服务器实例，以确定这些异常的应用程序服务器实例是否已返回到正常状态。

如果运行状况检查器发现某个异常实例已变为正常，该实例将被添加到正常实例列表中。

有关这些命令的更多信息，请参见 `create-http-health-checker(1)` 和 `delete-http-health-checker(1)`。

正常实例的附加运行状况检查属性

`create-http-health-checker` 创建的运行状况检查器仅检查异常实例。要定期检查正常实例，请在导出的 `loadbalancer.xml` 文件中设置某些附加属性。

注 - 只能在导出 `loadbalancer.xml` 之后，通过对此文件进行手动编辑来设置这些属性。没有等效的 `asadmin` 命令可以使用。

要检查正常的实例，请设置以下属性。

表 5-3 运行状况检查器手动设置属性

属性	定义
<code>active-healthcheck-enabled</code>	True/False 标志，用于表示是否要对正常服务器实例执行 Ping 操作以确定这些实例是否正常。要对服务器实例执行 Ping 操作，请将标志设置为 True。

表 5-3 运行状况检查器手动设置属性 (续)

属性	定义
number-healthcheck-retries	指定将未响应的服务器实例标记为异常之前，负载均衡器的运行状况检查器执行 Ping 操作的次数。有效范围在 1 到 1000 之间。默认设置值为 3。

通过编辑 `loadbalancer.xml` 文件来设置属性。例如：

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

如果添加了这些属性，随后再次编辑并导出 `loadbalancer.xml` 文件，则新导出的配置将不包含这些属性。必须再次将这些属性添加到新导出的配置中。

导出负载均衡器配置文件

Sun Java System Application Server 附带的负载均衡器插件使用名为 `loadbalancer.xml` 的配置文件。配置负载均衡器后，可以将配置详细信息从 `domain.xml` 导出到 `loadbalancer.xml` 文件。可以使用管理控制台或 `asadmin` 实用程序执行此操作。

▼ 使用管理控制台导出负载均衡器配置

- 1 导航到“HTTP 负载均衡器”节点并展开它。
- 2 单击所需的负载均衡器。
所有的负载均衡器配置详细信息都显示在“常规”、“设置”和“目标”选项卡中。
- 3 打开“导出”选项卡并单击“立即导出”。
- 4 将已导出的负载均衡器配置文件复制到 Web 服务器的配置目录。

▼ 使用 `asadmin` 工具导出负载均衡器配置

- 1 使用 `asadmin` 命令 `export-http-lb-config` 导出 `loadbalancer.xml` 文件。有关该命令的更多信息，请参见 `export-http-lb-config(1)`。
导出用于特定负载均衡器配置的 `loadbalancer.xml` 文件。您可以指定路径和其他文件名。如果不指定文件名，则此文件将被命名为 `loadbalancer.xml`。
`load-balancer-config-name`。如果不指定路径，则将在 `domain-dir/generated` 目录中创建该文件。

要在 Windows 上指定路径，请用引号将路径引起来。例如，
"`C:\Sun\AppServer\loadbalancer.xml`"。

2 将已导出的负载均衡器配置文件复制到 Web 服务器的配置目录。

例如，对于 Sun Java System Web Server，目录位置通常为 *web-server-root/config*。

Web 服务器配置目录中的负载均衡器配置文件名必须为 *loadbalancer.xml*。如果您的文件使用其他名称（例如 *loadbalancer.xml*、*load-balancer-config-name*），则必须进行重命名。

更改负载均衡器配置

如果您通过创建或删除对服务器的引用、部署新的应用程序、启用或禁用服务器或应用程序等方法来更改负载均衡器配置，请再次导出负载均衡器配置文件并将其复制到 Web 服务器的 *config* 目录。有关更多信息，请参见第 131 页中的“导出负载均衡器配置文件”。

负载均衡器插件将根据在负载均衡器配置中指定的重新装入时间间隔定期检查已更新的配置。在指定的时间值后，如果负载均衡器发现新的配置文件，它将开始使用该配置。

启用动态重新配置

使用动态重新配置时，负载均衡器插件将定期检查已更新的配置。

要启用动态重新配置，请执行以下步骤：

- 要创建负载均衡器配置，请将 `--reloadinterval` 选项与 `asadmin create-http-lb` 一起使用。有关该命令的更多信息，请参见 `create-http-lb(1)`。
此选项用于设置检查负载均衡器配置文件 *loadbalancer.xml* 的更改的时间间隔。值为 0 将禁用动态重新配置。默认情况下，将以 60 秒的重新装入时间间隔启用动态重新配置。
- 如果先前已禁用动态重新配置，或者要更改重新装入时间间隔，请使用 `asadmin set` 命令。
更改重新装入时间间隔后，请再次导出负载均衡器配置文件并将其复制到 Web 服务器的 *config* 目录中，然后重新启动 Web 服务器。

注 - 如果负载均衡器在尝试进行重新配置时遇到硬盘读取错误，它将使用内存中的当前配置。负载均衡器还确保了在覆盖现有配置之前，已修改的配置数据符合 DTD。

遇到磁盘读取错误后，将在 Web 服务器的错误日志文件中记录一则警告消息。

Sun Java System Web Server 的错误日志位于：
web-server-install-dir/web-server-instance/logs/。

禁用（停止）服务器实例或群集

出于任何原因停止应用服务器之前，实例应该完成正在处理的请求。正常禁用服务器实例或群集的进程称为停止。

负载均衡器使用以下策略来停止应用程序服务器实例：

- 如果已禁用某个实例（独立实例或群集的一部分），并且超时尚未到期，粘性请求将继续发送到该实例。但是，新请求将不会发送到已禁用的实例。
- 超时到期后，该实例将被禁用。从负载均衡器到该实例的所有打开的连接将被关闭。即使并非所有粘连至该实例的会话均已失效，负载均衡器也不会将任何请求发送到该实例。负载均衡器会将粘性请求故障转移到另一个正常实例上。

▼ 禁用服务器实例或群集

- 1 运行 `asadmin disable-http-lb-server`，设置超时值（以分钟为单位）。有关该命令的更多信息，请参见 `disable-http-lb-server(1)`。
- 2 使用 `asadmin export-http-lb-config` 导出负载均衡器配置文件。有关该命令的更多信息，请参见 `export-http-lb-config(1)`。
- 3 将导出的配置复制到 Web 服务器的 `config` 目录。
- 4 停止该服务器实例或群集。

禁用（停止）应用程序

在取消部署某个 Web 应用程序之前，该应用程序应该完成正在处理的请求。正常禁用应用程序的进程称为停止。停止应用程序时，您可以指定超时时间。基于超时时间，负载均衡器可使用以下策略停止应用程序：

- 如果超时尚未到期，负载均衡器不会将新请求转发到应用程序，而是将它们返回到 Web Server。但是，负载均衡器将会继续转发粘性请求，直至超时到期。
- 当超时到期时，负载均衡器将不接受此应用程序的任何请求，包括粘性请求。

当您从负载均衡器引用的每个服务器实例或群集中禁用应用程序时，则在再次启用该应用程序之前，已禁用的应用程序的用户将遭受服务损失。如果您从一个服务器实例或群集中禁用应用程序而使该应用程序在其他服务器实例或群集中保持启用状态，则用户仍可访问该应用程序。有关更多信息，请参见第 139 页中的“升级应用程序而不使可用性受到损失”。

▼ 禁用应用程序

1 使用 `asadmin disable-http-lb-application` 指定以下内容：

- 超时（以分钟为单位）。
- 要禁用的应用程序的名称。
- 要禁用此应用程序的目标群集或实例。

有关该命令的更多信息，请参见 `disable-http-lb-application(1)`。

2 使用 `asadmin export-http-lb-config` 导出负载均衡器配置文件。有关该命令的更多信息，请参见 `export-http-lb-config(1)`。

3 将导出的配置复制到 Web 服务器的 `config` 目录。

配置 HTTP 和 HTTPS 故障转移

如果 HTTP/HTTPS 会话所连接的原始应用程序服务器实例变为不可用，负载均衡器插件会将这些会话故障转移到其他应用程序服务器实例上。本节介绍了如何配置负载均衡器插件以启用 HTTP/HTTPS 路由选择和会话故障转移。

HTTPS 路由选择

负载均衡器插件将收到的所有 HTTP 或 HTTPS 请求路由到应用服务器实例。但是，如果启用了 HTTPS 路由选择，则负载均衡器插件将仅把 HTTPS 请求转发给使用 HTTPS 端口的应用程序服务器。HTTPS 路由选择是针对新请求和粘性请求而执行的。

如果收到了 HTTPS 请求且没有正在进行的会话，负载均衡器插件将选择使用已配置的 HTTPS 端口的可用应用程序服务器实例，并将请求转发到该实例。

在正在进行的 HTTP 会话中，如果收到对同一个会话的新 HTTPS 请求，则将使用在 HTTP 会话期间保存的会话和粘性信息来路由 HTTPS 请求。新的 HTTPS 请求将被路由到处理上一个 HTTP 请求的同一服务器上，但是，是在 HTTPS 端口上进行。

配置 HTTPS 路由选择

`create-http-lb-config` 命令的 `httpsrouting` 选项用于控制是为正在参与负载均衡的所有应用程序服务器打开还是关闭 HTTPS 路由选择。如果此选项设置为 `False`，则所有 HTTP 和 HTTPS 请求都将作为 HTTP 请求进行转发。如果设置为 `True`，则 HTTPS 将作为 HTTPS 请求进行转发。创建新的负载均衡器配置时请设置 HTTPS 路由选择，或者在以后使用 `asadmin set` 命令进行更改。

注-

- 要使用 HTTPS 路由选择，必须配置一个或多个 HTTPS 侦听器。
 - 如果 `https-routing` 设置为 `true`，而新请求或粘性请求传入到没有正常 HTTPS 侦听器的群集中，该请求将生成一个错误。
-

已知问题

负载均衡器对 HTTP/HTTPS 请求处理具有以下限制。

- 如果某个会话使用 HTTP 和 HTTPS 请求的组合，则第一个请求必须是 HTTP 请求。如果第一个请求是 HTTPS 请求，它后面将不能跟 HTTP 请求。这是因为与 HTTPS 会话关联的 Cookie 不是由浏览器返回的。浏览器将两个不同的协议解释为两个不同的服务器，并启动新的会话。仅当 `httpsrouting` 设置为 `True` 时，此限制才有效。
- 如果某个会话具有 HTTP 和 HTTPS 请求的组合，则必须将应用程序服务器实例配置为同时具有 HTTP 和 HTTPS 侦听器。仅当 `httpsrouting` 设置为 `true` 时，此限制才有效。
- 如果某个会话具有 HTTP 和 HTTPS 请求的组合，则必须将应用程序服务器实例配置为具有使用标准端口号（即，HTTP 为 80，HTTPS 为 443）的 HTTP 和 HTTPS 侦听器。不管为 `httpsrouting` 设置了何值，此限制都适用。

在负载均衡器中使用重定向

使用重定向可将请求从一个 URL 重定向到另一个 URL。例如，使用重定向将用户发送到其他 Web 站点（例如，从旧版本的应用程序重定向到较新版本的应用程序），或者从 HTTP 重定向到 HTTPS 或从 HTTPS 重定向到 HTTP。在应用程序中可按多种方式启用重定向（例如，基于 `servlet` 的重定向、`web.xml` 重定向）。不过，通过负载均衡器发送重定向 URL 可能需要对 Application Server 或负载均衡器进行一些其他配置。请注意，重定向与使用 HTTPS 路由选择转发的请求不同。使用重定向时，请将 `httpsrouting` 设置为 `False`。如果将 HTTPS 请求配置为转发到 HTTP，请使用 [第 134 页](#) 中的“HTTPS 路由选择”。

以下属性将影响重定向：HTTP 服务或 HTTP 侦听器的 `authPassthroughEnabled` 属性和 `proxyHandler` 属性，以及 `loadbalancer.xml` 文件中的 `rewrite-location` 属性。

authPassthroughEnabled 属性

当 Application Server `authPassthroughEnabled` 属性设置为 `True` 时，有关原始客户机请求的信息（如客户机 IP 地址、SSL 密钥大小和已认证的客户机证书链）将通过使用自定义请求标头发送到 HTTP 侦听器。如果安装了硬件加速器，`authPassthroughEnabled` 属性允许您使用硬件加速器来进行更快速的 SSL 验证。在负载均衡器中配置硬件加速器比在每个群集 Application Server 实例中配置硬件加速器更为容易。



注意 - 仅当 Application Server 位于防火墙之后时，才将 `authPassthroughEnabled` 设置为 `True`。

使用 `asadmin set` 命令在 HTTP 服务或单个 HTTP 侦听器中设置 `authPassthroughEnabled` 属性。单个 HTTP 侦听器的设置优先于 HTTP 服务的设置。

要在所有 HTTP/HTTPS 侦听器上设置 `authPassthroughEnabled` 属性，请使用以下命令：

```
asadmin set  
cluster-name-config.http-service.property.authPassthroughEnabled=true
```

要在单个侦听器上设置该属性，请使用以下命令：

```
asadmin set cluster-name-config.http-service.http-listener.  
listener-name.property.authPassthroughEnabled=true
```

proxyHandler 属性

Application Server 的代理处理程序负责检索有关代理服务器（在本例中，为负载均衡器）拦截并转发给 Application Server 的原始客户机请求的信息，并负责使该信息可供作为客户机请求的目标部署在 Application Server 中的 Web 应用程序使用。如果进行拦截的代理服务器是 SSL 终止代理服务器，则代理处理程序将检索有关原始请求的其他信息（如原始请求是否为 HTTPS 请求，以及是否启用了 SSL 客户机验证），并使这些信息可用。仅当将 `authPassThroughEnabled` 设置为 `True` 时才使用 `proxyHandler` 属性。

代理处理程序将在接收到的请求中检查自定义请求标头，这样，代理服务器即可传递有关原始客户机请求的信息，并使此信息可供使用标准 `ServletRequest` API 的 Application Server 中的 Web 应用程序使用。

代理处理程序实现是可配置的，可以使用 `proxyHandler` 属性在 HTTP 服务级别全局配置，也可以针对单个 HTTP 侦听器进行配置。`proxyHandler` 属性的值指定 `com.sun.appserv.ProxyHandler` 抽象类实现的完全限定类名。只要代理处理程序实现知道 HTTP 请求标头名称，并且了解其值的格式，可配置的代理处理程序实现就允许 Application Server 与任何代理服务器协同工作，这样，代理服务器便可以传递有关原始客户机请求的信息。

Application Server 的代理处理程序将从该请求标头中读取并解析 SSL 证书链。这将允许后端应用服务器实例检索 SSL 终止代理服务器（在本例中，为负载均衡器）拦截的原始客户机请求的有关信息。您可以使用默认的代理处理程序设置，也可以使用 HTTP 服务或 HTTP/HTTPS 侦听器的 `proxyHandler` 属性自行配置。`proxyHandler` 属性可用于指定该侦听器或全部侦听器使用的 `com.sun.appserv.ProxyHandler` 抽象类的自定义实现的完全限定类名。

此抽象类的实现将在给定请求中检查自定义请求标头，这样，代理服务器可将有关原始客户机请求的信息传送到 Application Server 实例，并将该信息返回给其调用方。默认实现将从名为 Proxy-ip 的 HTTP 请求标头中读取客户机 IP 地址，从名为 Proxy-keysize 的 HTTP 请求标头中读取 SSL 密钥大小，并从名为 Proxy-auth-cert 的 HTTP 请求标头中读取 SSL 客户机证书链。Proxy-auth-cert 值必须包含 BASE-64 编码的客户机证书链且无 BEGIN CERTIFICATE 和 END CERTIFICATE 边界，并将 \n 替换为 % d% a。

仅当 authPassThroughEnabled 设置为 True 时才可以此属性。如果在单个 HTTP 或 HTTPS 侦听器中设置 proxyHandler 属性，则它将覆盖所有侦听器的默认设置。

可使用 asadmin set 命令在 HTTP 服务或在单个 HTTP 侦听器中设置 proxyHandler 属性。

要在所有 HTTP/HTTPS 侦听器中设置 proxyHandler 属性，请使用以下命令：

```
asadmin set cluster-name-config.http-service.property.proxyHandler= classname
```

要在单个侦听器上设置该属性，请使用以下命令：

```
asadmin set cluster-name-config.http-service.http-listener.  
listener-name.property.proxyHandler= classname
```

rewrite-location 属性

如果 rewrite-location 属性设置为 True，则它将重写原始请求信息，并会包括协议（HTTP 或 HTTPS）、主机和端口信息。默认情况下，会将 rewrite-location 属性设置为 True，以便保持与以前 Application Server 版本的向后兼容性。

不能通过 asadmin create-http-lb-config 命令或 asadmin set 命令来使用 rewrite-location 属性。要使用该属性，请在导出负载均衡器配置后将其手动添加到 loadbalancer.xml 文件中。例如，将以下内容添加到导出的 loadbalancer.xml 文件中：

```
<property name="rewrite-location" value="false"/>
```

设置 `rewrite-location` 属性时，请记住以下几点：

- 如果 `httpsrouting` 为 `False`，且 `Application Server` 中未启用 `authPassthroughEnabled`，请将 `rewrite-location` 属性设置为 `True`。如果未启用 `authPassthroughEnabled`，`Application Server` 将不能识别原始请求的协议（`HTTP` 或 `HTTPS`）。通过将 `rewrite-location` 设置为 `True`，负载均衡器可适当地修改重写位置的协议部分。即，如果客户机发送 `HTTPS` 请求，则负载均衡器会将客户机重定向到负载均衡器上启用 `HTTPS` 的侦听器端口。对于 `HTTP` 请求，该过程相同。
- 如果 `httpsrouting` 为 `False`，且 `Application Server` 中启用了 `authPassthroughEnabled`，则 `rewrite-location` 既可设置为 `True`，也可设置为 `False`，因为 `Application Server` 可识别客户机请求是 `HTTP` 还是 `HTTPS`。启用了 `authPassthroughEnabled` 时，`Application Server` 可适当地修改重写位置的协议部分。如果 `rewrite-location` 设置为 `False`，则负载均衡器不会重写已重定向 `URL` 的位置。如果设置为 `True`，则它会重写已重定向 `URL` 的位置。但此重写是不需要的，因为 `Application Server` 可以识别来自客户机的 `HTTPS` 连接。另外，如果应用程序需要将 `HTTP` 重定向到 `HTTPS`，或将 `HTTPS` 重定向到 `HTTP`，则必须将 `rewrite-location` 参数设置为 `False`。

配置幂等 URL

幂等请求是一种在重试时不会在应用程序中造成任何更改或不一致的请求。在 `HTTP` 中，某些方法（例如 `GET`）是幂等的，而其他方法（例如 `POST`）则不是。重试幂等 `URL` 不能导致服务器或数据库中的值发生更改。唯一的区别在于用户收到的响应会有所不同。

幂等请求的示例包括搜索引擎查询和数据库查询。基本原则是重试不会导致数据的更新或修改。

要增强已部署的应用程序的可用性，请这样配置环境：使其在由负载均衡器提供服务的所有应用服务器实例上重试失败的幂等 `HTTP` 请求。此选项用于只读请求（例如，重试搜索请求）。

请在 `sun-web.xml` 文件中配置幂等 `URL`。当您导出负载均衡器配置时，幂等 `URL` 信息将自动被添加到 `loadbalancer.xml` 文件中。

有关配置幂等 `URL` 的更多信息，请参见《`Sun Java System Application Server 9.1 Developer's Guide`》中的“`Configuring Idempotent URL Requests`”。

配置多个 Web 服务器实例

Sun Java System Application Server 安装程序不允许在单个计算机上安装多个负载平衡器插件。要在单个群集或多个群集中的单个计算机上安装多个带有负载平衡器插件的 Web 服务器，需要手动执行一些步骤来配置负载平衡器插件。

▼ 配置多个 Web 服务器实例

- 1 配置新的 Web Server 实例以使用负载平衡器插件。
有关详细说明，请参见第 100 页中的“配置 Sun Java System Web Server”。
- 2 将 DTD 文件 `sun-loadbalancer_1_1.dtd` 从现有 Web 服务器实例的 `config` 目录复制到新实例的 `config` 目录。
- 3 设置负载平衡器配置文件。执行以下操作之一：
 - 复制现有负载平衡器配置。
使用现有负载平衡器配置，将 `loadbalancer.xml` 文件从现有 Web 服务器实例的 `config` 目录复制到新实例的 `config` 目录中。
 - 创建新的负载平衡器配置：
 - a. 使用 `asadmin create-http-lb-config` 创建新的负载平衡器配置。
 - b. 使用 `asadmin export http-lb-config` 将新配置导出到 `loadbalancer.xml` 文件中。
 - c. 将 `loadbalancer.xml` 文件复制到新 Web Server 的 `config` 目录中。
有关创建负载平衡器配置并将其导出到 `loadbalancer.xml` 文件的信息，请参见第 127 页中的“在 DAS 上配置 HTTP 负载平衡器”。

升级应用程序而不使可用性受到损失

将应用程序升级到新版本而不会给用户造成任何可用性方面的损失，这样的升级称为滚动升级。管理好应用程序升级前后的两个版本可以确保应用程序的当前用户能够不中断地完成工作，同时新用户可以透明地获得应用程序的新版本。执行滚动升级时，用户不会察觉在进行升级。

应用程序兼容性

根据应用程序两个版本间变更的大小，滚动升级的难度将有所不同。

如果变更很小（例如静态文本和图像的变更），则此应用程序的两个版本可以**兼容**，并且可同时在同一群集中运行。

兼容的应用程序必须满足以下条件：

- 使用相同的会话信息
- 使用兼容的数据库模式
- 通常具有兼容的应用程序级业务逻辑
- 使用相同的物理数据源

您可以对单个群集或多个群集中兼容的应用程序执行滚动升级。有关更多信息，请参见第 140 页中的“[在单个群集中升级](#)”。

如果应用程序的两个版本不满足上述所有条件，则应用程序被视为**不兼容**。在一个群集中执行不兼容的应用程序版本将破坏应用程序数据并导致会话故障转移功能失常。问题取决于不兼容的类型和程度。好的做法是通过创建要在其上部署新版本的“阴影群集”来升级不兼容的应用程序，然后再慢慢停止旧的群集和应用程序。有关更多信息，请参见第 144 页中的“[升级不兼容的应用程序](#)”。

应用程序开发者和管理员是确定应用程序版本是否兼容的最佳人员。如果不确定，请假定版本不兼容，因为这是最安全的方法。

在单个群集中升级

假如单个群集的配置未与其他任何群集共享，则您可以对部署到此群集的应用程序执行滚动升级。

▼ 在单个群集中升级应用程序

1 保存旧版本的应用程序或备份域。

要备份域，请使用 `asadmin backup-domain` 命令。有关该命令的更多信息，请参见 `backup-domain(1)`。

2 关闭群集的动态重新配置（如果已启用）。

从管理控制台执行此操作：

- a. 展开“配置”节点。
- b. 单击群集配置的名称。
- c. 在“配置系统属性”页面中，取消选中“已启用动态重新配置”框。
- d. 单击“保存”

或者，使用以下命令：

```
asadmin set --user user --passwordfile password-file
cluster-name-config.dynamic-reconfiguration-enabled=false
```

- 3 将已升级的应用程序重新部署到目标 domain 中。
如果使用管理控制台进行重新部署，域将自动成为目标。如果您使用 asadmin，请指定目标域。由于已禁用动态重新配置，因此旧应用程序将继续在群集上运行。
- 4 使用 asadmin enable-http-lb-application 为实例启用已重新部署的应用程序。有关该命令的更多信息，请参见 enable-http-lb-application(1)。
- 5 从负载均衡器停止群集中的一个服务器实例。
请执行以下步骤：
 - a. 使用 asadmin disable-http-lb-server 禁用服务器实例。有关该命令的更多信息，请参见 disable-http-lb-server(1)。
 - b. 使用 asadmin export-http-lb-config 导出负载均衡器配置文件。有关该命令的更多信息，请参见 export-http-lb-config(1)。
 - c. 将已导出的配置文件复制到 Web 服务器实例的配置目录。
例如，对于 Sun Java System Web Server，目录位置为 `web-server-install-dir/https-host-name /config/loadbalancer.xml`。为确保负载均衡器能够装入新的配置文件，请通过在负载均衡器配置中设置 `reloadinterval` 来确保启用动态重新配置。
 - d. 请等待，直至超时到期。
监视负载均衡器的日志文件以确保实例已脱机。如果用户看到重试 URL，将跳过停止时间并立即重新启动服务器。
- 6 在群集中的其他实例仍处于运行状态的情况下，重新启动已禁用的服务器实例。
重新启动操作将使服务器与域同步，并更新应用程序。
- 7 测试重新启动的服务器上的应用程序，以确保应用程序运行正常。
- 8 重新启用负载均衡器中的服务器实例。
请执行以下步骤：
 - a. 使用 asadmin enable-http-lb-server 启用服务器实例。有关该命令的更多信息，请参见 enable-http-lb-server(1)。
 - b. 使用 asadmin export-http-lb-config 导出负载均衡器配置文件。有关该命令的更多信息，请参见 export-http-lb-config(1)。

- c. 将配置文件复制到 Web 服务器的配置目录。
- 9 对群集中的每个实例重复步骤 5 至步骤 8。
- 10 当所有服务器实例都具有新的应用程序并已运行时，您可以再次为群集启用动态重新配置。

在多个群集中进行升级

▼ 在两个或多个群集中升级兼容的应用程序：

- 1 保存旧版本的应用程序或备份域。

要备份域，请使用 `asadmin backup-domain` 命令。有关该命令的更多信息，请参见 `backup-domain(1)`。

- 2 关闭所有群集的动态重新配置（如果已启用）。

从管理控制台执行此操作：

- a. 展开“配置”节点。
- b. 单击其中一个群集配置的名称。
- c. 在“配置系统属性”页面中，取消选中“已启用动态重新配置”框。
- d. 单击“保存”
- e. 对其他群集重复此过程

或者，使用以下命令：

```
asadmin set --user user --passwordfile password-file cluster-name  
-config.dynamic-reconfiguration-enabled=false
```

- 3 将已升级的应用程序重新部署到目标 domain 中。

如果使用管理控制台进行重新部署，域将自动成为目标。如果您使用 `asadmin`，请指定目标域。由于已禁用动态重新配置，因此旧应用程序将继续在群集上运行。

- 4 使用 `asadmin enable-http-lb-application` 为群集启用已重新部署的应用程序。有关该命令的更多信息，请参见 `enable-http-lb-application(1)`。

- 5 从负载均衡器停止一个群集
 - a. 使用 `asadmin disable-http-lb-server` 禁用此群集。有关该命令的更多信息，请参见 `disable-http-lb-server(1)`。
 - b. 使用 `asadmin export-http-lb-config` 导出负载均衡器配置文件。有关该命令的更多信息，请参见 `export-http-lb-config(1)`。
 - c. 将已导出的配置文件复制到 Web 服务器实例的配置目录。
例如，对于 Sun Java System Web Server，目录位置为 `web-server-install-dir/https-host-name /config/loadbalancer.xml`。必须为负载均衡器启用动态重新配置（通过在负载均衡器配置中设置 `reloadinterval`），以便自动装入新的负载均衡器配置文件。
 - d. 请等待，直至超时到期。
监视负载均衡器的日志文件以确保实例已脱机。如果用户看到重试 URL，将跳过停止时间并立即重新启动服务器。
- 6 在其他群集仍处于运行状态的情况下，重新启动已禁用的群集。
重新启动操作将使群集与域同步，并更新应用程序。
- 7 测试重新启动的群集上的应用程序，以确保应用程序运行正常。
- 8 在负载均衡器中启用此群集：
 - a. 使用 `asadmin enable-http-lb-server` 启用群集。有关该命令的更多信息，请参见 `enable-http-lb-server(1)`。
 - b. 使用 `asadmin export-http-lb-config` 导出负载均衡器配置文件。有关该命令的更多信息，请参见 `export-http-lb-config(1)`。
 - c. 将配置文件复制到 Web 服务器的配置目录。
- 9 对其他群集重复步骤 5 至步骤 8。
- 10 当所有服务器实例都具有新的应用程序并已运行时，您可以再次为所有群集启用动态重新配置。

升级不兼容的应用程序

如果应用程序的新版本与旧版本不兼容，请使用以下过程。有关应用程序兼容条件的信息，请参见第 139 页中的“应用程序兼容性”。此外，您必须在两个或更多群集中升级不兼容的应用程序。如果您只有一个群集，则请为升级创建“阴影群集”，如下所述。

升级不兼容的应用程序时，请执行以下操作：

- 赋予应用程序新版本一个名称，此名称应与旧版本的名称不同。以下步骤假定已重新命名此应用程序。
- 如果数据模式不兼容，请在规划数据迁移后使用不同的物理数据源。
- 将应用程序的新版本部署到与旧版本所部署到的群集不同的群集上。
- 在使运行旧应用程序的群集脱机前，请为其设置一个适当长的超时，因为此应用程序的请求无法故障转移到新的群集。这些用户会话将只会失败。

▼ 通过创建第二个群集来升级不兼容的应用程序

1 保存旧版本的应用程序或备份域。

要备份域，请使用 `asadmin backup-domain` 命令。有关该命令的更多信息，请参见 `backup-domain(1)`。

2 在与现有群集相同或不同组的计算机上创建“阴影群集”。如果已经拥有第二个群集，请跳过此步骤。

a. 使用管理控制台创建新的群集并引用现有群集的命名配置。

为每台计算机上的新实例自定义端口，以避免与现有的活动端口冲突。

b. 对于所有与群集相关联的资源，请使用 `asadmin create-resource-ref` 将资源引用添加到新创建的群集。有关该命令的更多信息，请参见 `create-resource-ref(1)`。

c. 使用 `asadmin create-application-ref` 从新创建的群集创建对部署到此群集的所有其他应用程序（当前已重新部署的应用程序除外）的引用。有关该命令的更多信息，请参见 `create-application-ref(1)`。

d. 使用 `asadmin configure-ha-cluster` 将群集配置为高可用性群集。有关该命令的更多信息，请参见 `configure-ha-cluster(1)`。

e. 使用 `asadmin create-http-lb-ref` 创建对负载均衡器配置文件中新创建的群集的引用。有关该命令的更多信息，请参见 `create-http-lb-ref(1)`。

3 赋予应用程序新版本一个名称，此名称应与旧版本的名称不同。

- 4 将新群集作为目标来部署新的应用程序。使用一个或多个不同的上下文根。
- 5 使用 `asadmin enable-http-lb-application` 为群集启用已部署的新应用程序。有关该命令的更多信息，请参见 `enable-http-lb-application(1)`。
- 6 在另一个群集仍处于运行状态的情况下，启动新群集。
启动操作将导致群集与域同步，并使用新应用程序进行更新。
- 7 测试新群集上的应用程序，以确保应用程序运行正常。
- 8 使用 `asadmin disable-http-lb-server` 从负载均衡器上禁用旧群集。有关该命令的更多信息，请参见 `disable-http-lb-server(1)`。
- 9 为延迟会话存在的时间设置超时值。
- 10 使用 `asadmin enable-http-lb-server` 从负载均衡器上启用新群集。有关该命令的更多信息，请参见 `enable-http-lb-server(1)`。
- 11 使用 `asadmin export-http-lb-config` 导出负载均衡器配置文件。有关该命令的更多信息，请参见 `export-http-lb-config(1)`。
- 12 将已导出的配置文件复制到 Web 服务器实例的配置目录。
例如，对于 Sun Java System Web Server，目录位置为 `web-server-install-dir/https-host-name/config/loadbalancer.xml`。必须为负载均衡器启用动态重新配置（通过在负载均衡器配置中设置 `reloadinterval`），以便自动装入新的负载均衡器配置文件。
- 13 在超时时间到期或旧应用程序的所有用户都退出后，停止旧的群集并删除旧的应用程序。

监视 HTTP 负载均衡器插件

- 第 145 页中的“配置日志消息”
- 第 146 页中的“日志消息类型”
- 第 147 页中的“启用负载均衡器日志记录”
- 第 148 页中的“了解监视消息”

配置日志消息

负载均衡器插件使用 Web 服务器的日志记录机制来写入日志消息。Application Server 上的默认日志级别被设置为 Sun Java System Web Server (INFO)、Apache Web Server (WARN)

和 Microsoft IIS (INFO) 上的默认日志级别。应用服务器日志级别 (FINE、FINER 和 FINEST) 映射到 Web 服务器上的 DEBUG 级别。

这些日志消息将被写入 Web 服务器日志文件，其形式为可使用脚本进行解析或可被导入电子表格以计算所需的衡量标准的原始数据。

日志消息类型

负载均衡器插件生成以下类型的日志消息：

- 第 146 页中的“负载均衡器配置器日志消息”
- 第 146 页中的“请求分发和运行时日志消息”
- 第 147 页中的“配置器错误消息”

负载均衡器配置器日志消息

使用幂等 URL 和错误页面设置时，将记录这些消息。

幂等 URL 模式配置的输出包含以下信息：

- 当日志级别被设置为详细时：


```
CONFxxxx: IdempotentUrlPattern configured <url-pattern> <no-of-retries> for
web-module : <web-module>
```
- 当日志级别被设置为严重时：


```
CONFxxxx: Duplicate entry of Idempotent URL element <url-pattern> for
webModule <web-module> in loadbalancer.xml."
```
- 当日志级别被设置为警告时：


```
CONFxxxx: Invalid IdempotentUrlPatternData <url-pattern> for web-module
<web-module>
```

错误页面 URL 配置的输出包含以下信息（日志级别设置为警告）：

```
CONFxxxx: Invalid error-url for web-module <web-module>
```

请求分发和运行时日志消息

在对请求进行负载均衡和分发时，将生成这些日志消息。

- 每个方法开始的标准日志的输出均包含以下信息（日志级别设置为详细）：


```
ROUTxxxx: Executing Router method <method_name>
```
- 每个方法开始的路由器日志的输出均包含以下信息（日志级别设置为信息）：


```
ROUTxxxx: Successfully Selected another ServerInstance for idempotent request
<Request-URL>
```
- 运行时日志的输出包含以下信息（日志级别设置为信息）：

RNTMxxxx: Retrying Idempotent <GET/POST/HEAD> Request <Request-URL>

配置器错误消息

如果存在配置问题（例如，缺少引用的自定义错误页面），将显示这些错误消息。

- 日志级别设置为信息：

ROUTxxxx: Non Idempotent Request <Request-URL> cannot be retried

例如：ROUTxxxx: Non Idempotent Request http://sun.com/addToDB?x=11&abc=2 cannot be retried

- 日志级别设置为详细：

RNTMxxxx: Invalid / Missing Custom error-url / page: <error-url> for web-module: <web-module>

例如：RNTMxxxx: Invalid / Missing Custom error-url / page: myerror1xyz for web-module: test

启用负载均衡器日志记录

负载均衡器插件记录以下信息：

- 每个请求的请求开始/停止信息。
- 当请求从异常实例故障转移到正常实例时的故障转移请求信息。
- 每个运行状况检查周期结束时的异常实例列表。

注 - 启用负载均衡器日志记录后，如果将 Web 服务器日志级别设置为 DEBUG 或设置为打印详细消息，负载均衡器会将 HTTP 会话 ID 写入 Web 服务器日志文件中。因此，如果托管负载均衡器插件的 Web 服务器位于 DMZ 中，请不要在生产环境中使用 DEBUG 或类似的日志级别。

如果必须使用 DEBUG 日志记录级别，请在 loadbalancer.xml 中将 require-monitor-data 属性设置为 false，以关闭负载均衡器日志记录。

▼ 打开负载均衡器日志记录

- 1 设置 Web 服务器中的日志选项。此过程将取决于 Web Server：

- 对于 Sun Java System Web Server

在服务器的管理控制台中，转到“Magnus 编辑器”选项卡，然后将“Log Verbose”选项设置为 On。

- 对于 Apache Web Server，请将日志级别设置为 DEBUG。

- 对于 Microsoft IIS，请在 `sun-passthrough.properties` 文件中将日志级别设置为 FINE。
- 2 将负载均衡器配置的 `monitor` 选项设置为 `True`。
- 使用 `asadmin create-http-lb-config` 命令在最初创建负载均衡器配置时将监视设置为 `True`，也可以在以后使用 `asadmin set` 命令将其设置为 `True`。默认情况下，监视处于禁用状态。

了解监视消息

负载均衡器插件日志消息的格式如下所示：

- HTTP 请求的开头处包含以下信息：


```
RequestStart Sticky(New) <req-id> <time-stamp> <URL>
时间戳值是从 1970 年 1 月 1 日开始的毫秒数。例如：
RequestStart New 123456 602983
http://austen.sun.com/Webapps-simple/servlet/Example1
```
- HTTP 请求的结尾包含 `RequestExit` 消息，如下所示：


```
RequestExit Sticky(New) <req-id> <time-stamp> <URL> <listener-id>
<response-time> Failure-<reason for error>(incase of a failure)
例如：
RequestExit New 123456 603001
http://austen.sun.com/Webapps-simple/servlet/Example1 http://austen:2222 18
```

注 - 在 `RequestExit` 消息中，*response-time* 表示从负载均衡器插件方面请求的往返总时间（以毫秒为单位）。

- 异常实例列表，如下所示：


```
UnhealthyInstances <cluster-id> <time-stamp> <listener-id>, <listener-id>...
例如：
UnhealthyInstances cluster1 701923 http://austen:2210, http://austen:3010
```
- 故障转移请求列表，如下所示：


```
FailedoverRequest <req-id> <time-stamp> <URL> <session-id>
<failed-over-listener-id> <unhealthy-listener-id>
例如：
FailedoverRequest 239496 705623
http://austen.sun.com/Apps/servlet/SessionTest 16dfdac3c7e80a40
http://austen:4044 http://austen:4045
```

使用 Application Server 群集

本章介绍如何使用 Application Server 群集。它包含以下小节：

- 第 149 页中的“群集概述”
- 第 149 页中的“组管理服务”
- 第 150 页中的“使用群集”

群集概述

群集是命名的服务器实例集合，它们共享相同的应用程序、资源和配置信息。您可以将不同计算机上的服务器实例分组到一个逻辑群集中并将其作为一个单元来管理。您可以使用 DAS 轻松控制多机群集的生命周期。

群集可以启用水平可伸缩性、负载平衡和故障转移保护。根据定义，群集中的所有实例都具有相同的资源和应用程序配置。当群集中的服务器实例或计算机出现故障时，负载平衡器检测到该故障，会将通信从出现故障的实例重定向至群集中的其他实例，并恢复用户会话状态。由于群集中所有实例上的应用程序和资源都相同，因此一个实例可以故障转移至群集中的任何其他实例。

组管理服务

组管理服务 (Group Management Service, GMS) 是为群集中的实例启用的基础结构组件。启用了 GMS 时，如果群集实例出现故障，群集和域管理服务器将可以识别该故障，并在出现故障时采取行动。Application Server 的许多功能依赖于 GMS。例如，IIOP 故障转移、内存中复制、事务服务和计时器服务功能均使用 GMS。

如果群集中的服务器实例位于不同的计算机上，请确保这些计算机在同一子网上。

注 - GMS 功能在开发者配置文件中不可用。在群集配置文件和企业配置文件中，默认情况下启用 GMS。

GMS 是 Shoal 框架的核心服务。有关 Shoal 的更多信息，请访问项目 [Shoal 主页 \(https://shoal.dev.java.net/\)](https://shoal.dev.java.net/)。

▼ 对群集启用或禁用 GMS

- 1 在树组件中，选择“群集”。
- 2 单击群集的名称。
- 3 在“常规信息”下，确保根据需要选中或取消选中了“启用心跳”复选框。
 - 要启用 GMS，请确保选中“启用心跳”复选框。
 - 要禁用 GMS，请确保取消选中“启用心跳”复选框。
- 4 如果要启用 GMS 且需要与这些默认值不同的值，请为 GMS 更改默认端口和 IP 地址。
- 5 单击“保存”。

配置 GMS

通过更改确定 GMS 检查故障的频率的设置，配置适合您环境的 GMS。例如，可以更改故障检测尝试之间的超时、对可疑故障成员的重试次数或检查群集成员时的超时。

要在管理控制台中配置监视，请转到 Application Server 节点->“配置”->“组管理服务”。

等效的 asadmin 命令是 get 和 set。

使用群集

- 第 151 页中的“创建群集”
- 第 152 页中的“为群集创建服务器实例”
- 第 153 页中的“配置群集”
- 第 153 页中的“启动、停止和删除群集实例”
- 第 154 页中的“在群集中配置服务器实例”

- 第 155 页中的 “为群集配置应用程序”
- 第 155 页中的 “为群集配置资源”
- 第 156 页中的 “删除群集”
- 第 156 页中的 “迁移 EJB 计时器”
- 第 157 页中的 “升级组件而不使服务受到任何损失”

▼ 创建群集

- 1 在树组件中，选择“群集”节点。
- 2 在“群集”页面中，单击“新建”。
将显示“创建群集”页面。
- 3 在“名称”字段中，键入群集的名称。
名称必须：
 - 仅由大小写字母、数字、下划线、连字符和句点(.)组成
 - 在所有节点代理名称、服务器实例名称、群集名称和配置名称中都是唯一的
 - 不能是 domain
- 4 在“配置”字段中，从下拉式列表中选择配置。
 - 要创建不使用共享配置的群集，请选择 default-config。
使标有“复制选定的配置”的单选按钮保持选定状态。默认配置的副本的名称将为 *cluster_name-config*。
 - 要创建使用共享配置的群集，请从下拉式列表中选择该配置。
选择标有“引用选定的配置”的单选按钮，以创建使用指定的现有共享配置的群集。
- 5 另外，添加服务器实例。
您也可以在创建群集后添加服务器实例。
为群集添加服务器实例之前，请先创建一个或多个节点代理或节点代理占位符。请参见第 176 页中的“创建节点代理占位符”。
创建服务器实例：
 - a. 在“要创建的服务器实例”区域，单击“添加”。
 - b. 在“实例名称”字段中为实例输入名称。
 - c. 从“节点代理”下拉式列表中选择节点代理。

- 6 单击“确定”。
- 7 在显示的“已成功创建群集”页面中单击“确定”。

更多信息 **等效的 asadmin 命令**

```
create-cluster
```

- 另请参见
- 第 153 页中的“配置群集”
 - 第 152 页中的“为群集创建服务器实例”
 - 第 155 页中的“为群集配置应用程序”
 - 第 155 页中的“为群集配置资源”
 - 第 156 页中的“删除群集”
 - 第 157 页中的“升级组件而不使服务受到任何损失”

有关如何管理群集、服务器实例和节点代理的详细信息，请参见第 169 页中的“部署节点代理”。

▼ 为群集创建服务器实例

开始之前 为群集创建服务器实例之前，必须先创建节点代理或节点代理占位符。请参见第 176 页中的“创建节点代理占位符”。

- 1 在树组件中，展开“群集”节点。
- 2 选择群集节点。
- 3 单击“实例”选项卡以显示“群集服务器实例”页面。
- 4 单击“新建”以显示“创建群集服务器实例”页面。
- 5 在“名称”字段中，键入服务器实例的名称。
- 6 从“节点代理”下拉式列表中选择节点代理。
- 7 单击“确定”。

更多信息 **等效的 asadmin 命令**

```
create-instance
```

- 另请参见
- 第 167 页中的“什么是节点代理？”
 - 第 151 页中的“创建群集”

- 第 153 页中的 “配置群集”
- 第 155 页中的 “为群集配置应用程序”
- 第 155 页中的 “为群集配置资源”
- 第 156 页中的 “删除群集”
- 第 157 页中的 “升级组件而不使服务受到任何损失”
- 第 154 页中的 “在群集中配置服务器实例”

▼ 配置群集

1 在树组件中，展开“群集”节点。

2 选择群集的节点。

在“常规信息”页面中，您可以执行以下任务：

- 单击“启动实例”以启动群集服务器实例。
- 单击“停止实例”以停止群集服务器实例。
- 单击“迁移 EJB 计时器”以将 EJB 计时器从已停止的服务器实例迁移到群集中的其他服务器实例。

更多信息 等效的 asadmin 命令

`start-cluster`、`stop-cluster`、`migrate-timers`

另请参见

- 第 151 页中的 “创建群集”
- 第 152 页中的 “为群集创建服务器实例”
- 第 155 页中的 “为群集配置应用程序”
- 第 155 页中的 “为群集配置资源”
- 第 156 页中的 “删除群集”
- 第 157 页中的 “升级组件而不使服务受到任何损失”
- 第 156 页中的 “迁移 EJB 计时器”

▼ 启动、停止和删除群集实例

1 在树组件中，展开“群集”节点。

2 展开包含服务器实例的群集的节点。

3 单击“实例”选项卡以显示“群集服务器实例”页面。

在此页面中，您可以执行以下操作：

- 选中一个实例的复选框，然后单击“删除”、“启动”或“停止”，以对所有指定的服务器实例执行选定的操作。
- 单击实例的名称，以显示“常规信息”页面。

▼ 在群集中配置服务器实例

- 1 在树组件中，展开“群集”节点。
- 2 展开包含服务器实例的群集节点。
- 3 选择服务器实例节点。
- 4 在“常规信息”页面中，您可以执行以下操作：
 - 单击“启动实例”以启动实例。
 - 单击“停止实例”以停止正在运行的实例。
 - 单击“JNDI 浏览”以浏览正在运行的实例的 JNDI 树。
 - 单击“查看日志文件”，以打开服务器日志查看器。
 - 单击“轮转日志文件”，以轮转实例的日志文件。该操作将安排日志文件以进行轮转。实际的轮转将在下一次向日志文件写入条目时发生。
 - 单击“恢复事务”以恢复未完成的事务。
 - 单击“属性”选项卡，以修改实例的端口号。
 - 单击“监视”选项卡，以更改监视属性。

另请参见

- [第 151 页中的“创建群集”](#)
- [第 153 页中的“配置群集”](#)
- [第 152 页中的“为群集创建服务器实例”](#)
- [第 155 页中的“为群集配置应用程序”](#)
- [第 155 页中的“为群集配置资源”](#)
- [第 156 页中的“删除群集”](#)
- [第 157 页中的“升级组件而不使服务受到任何损失”](#)
- 《Sun Java System Application Server 9.1 管理指南》中的“配置 Application Server 从事务中恢复的方式”

▼ 为群集配置应用程序

- 1 在树组件中，展开“群集”节点。
- 2 选择群集的节点。
- 3 单击“应用程序”选项卡，以显示“应用程序”页面。

在此页面中，您可以执行以下操作：

- 从“部署”下拉式列表中，选择要部署的应用程序的类型。在显示的“部署”页面中，指定应用程序。
- 从“过滤器”下拉式列表中，选择要在列表中显示的应用程序的类型。
- 要编辑应用程序，请单击应用程序名称。
- 选中应用程序旁边的复选框，然后选择“启用”或“禁用”以启用或禁用用于群集的应用程序。

另请参见

- [第 151 页中的“创建群集”](#)
- [第 153 页中的“配置群集”](#)
- [第 152 页中的“为群集创建服务器实例”](#)
- [第 155 页中的“为群集配置资源”](#)
- [第 156 页中的“删除群集”](#)
- [第 157 页中的“升级组件而不使服务受到任何损失”](#)

▼ 为群集配置资源

- 1 在树组件中，展开“群集”节点。
- 2 选择群集的节点。
- 3 单击“资源”选项卡，以显示“资源”页面。

在此页面中，您可以执行以下操作：

- 为群集创建新资源：从“新建”下拉式列表中，选择要创建的资源的类型。创建资源时，请确保将群集指定为目标。
- 全局启用或禁用资源：选中资源旁边的复选框，然后单击“启用”或“禁用”。该操作不会删除资源。
- 仅显示特定类型的资源：从“过滤器”下拉式列表中，选择要在列表中显示的资源的类型。
- 编辑资源：单击资源名称。

- 另请参见
- 第 151 页中的 “创建群集”
 - 第 153 页中的 “配置群集”
 - 第 152 页中的 “为群集创建服务器实例”
 - 第 155 页中的 “为群集配置应用程序”
 - 第 156 页中的 “删除群集”

▼ 删除群集

- 1 在树组件中，选择“群集”节点。
- 2 在“群集”页面中，选中群集名称旁边的复选框。
- 3 单击“删除”。

更多信息 等效的 asadmin 命令

```
delete-cluster
```

- 另请参见
- 第 151 页中的 “创建群集”
 - 第 153 页中的 “配置群集”
 - 第 152 页中的 “为群集创建服务器实例”
 - 第 155 页中的 “为群集配置应用程序”
 - 第 155 页中的 “为群集配置资源”
 - 第 157 页中的 “升级组件而不使服务受到任何损失”

▼ 迁移 EJB 计时器

如果服务器实例非正常或意外地停止运行，则可能需要将该服务器实例上安装的 EJB 计时器移至群集中正在运行的服务器实例。要完成此操作，请执行以下步骤：

- 1 在树组件中，展开“群集”节点。
- 2 选择群集节点。
- 3 在“常规信息”页面中，单击“迁移 EJB 计数器”。
- 4 在“迁移 EJB 计时器”页面中：
 - a. 从“源”下拉式列表中，选择要迁移的计时器所在的已停止的服务器实例。

- b. (可选的) 从“目标”下拉式列表中, 选择要将计时器迁移到的正在运行的服务器实例。

如果将该字段保留为空, 将随机选择一个正在运行的服务器实例。

- c. 单击“确定”。

5 停止并重新启动目标服务器实例。

如果源服务器实例正在运行或目标服务器实例未运行, 则管理控制台将显示一条错误消息。

更多信息 等效的 asadmin 命令

```
migrate-timers
```

另请参见

- 第 153 页中的“配置群集”
- 用于配置 EJB 计时器服务的设置的管理控制台联机帮助

▼ 升级组件而不使服务受到任何损失

在 Application Server 中, 您可以使用负载均衡器和多个群集来升级组件, 而不使服务受到任何损失。例如, 组件可以是 JVM、Application Server 或 Web 应用程序。

在以下情况下不能使用此方法:

- 您更改高可用性数据库 (high-availability database, HADB) 的模式。有关更多信息, 请参见第 3 章。

注 - HADB 软件随 Sun Java System Application Server 的 Application Server 独立分发提供。有关 Sun Java System Application Server 的可用分发的信息, 请参见《Sun Java System Application Server 9.1 Installation Guide》中的“Distribution Types and Their Components”。HADB 功能仅在企业配置文件中可用。有关配置文件的信息, 请参见《Sun Java System Application Server 9.1 管理指南》中的“用法配置文件”。

- 您执行涉及对应用程序数据库模式的更改的应用程序升级。



注意 - 请同时升级群集中的所有服务器实例。否则, 可能会出现由从运行不同版本的组件的一个实例到另一个实例的会话故障转移引起的版本不匹配。

- 1 使用群集的“常规信息”页面上的“停止群集”按钮来停止其中一个群集。
- 2 升级该群集中的组件。

3 使用群集的“常规信息”页面上的“启动群集”按钮来启动群集。

4 对其他群集逐个重复此过程。

由于一个群集中的会话决不会故障转移到另一个群集中的会话，因此不会出现由会话故障转移所引起的版本不匹配，这种故障转移是指从运行一个版本的组件的服务器实例故障转移到运行其他版本的组件的另一个服务器实例（位于其他群集中）。这样，群集就用作了它所包含的服务器实例的会话故障转移的安全边界。

- 另请参见
- 第 151 页中的“创建群集”
 - 第 153 页中的“配置群集”
 - 第 152 页中的“为群集创建服务器实例”
 - 第 155 页中的“为群集配置应用程序”
 - 第 155 页中的“为群集配置资源”
 - 第 156 页中的“删除群集”

管理配置

本章介绍了在 Application Server 中添加、更改以及使用命名的服务器配置。它包含以下小节：

- 第 159 页中的“使用配置”
- 第 162 页中的“使用命名配置”

使用配置

- 第 159 页中的“配置”
- 第 160 页中的“default-config 配置”
- 第 160 页中的“创建实例或群集时创建的配置”
- 第 161 页中的“唯一端口号和配置”

配置

配置是一组服务器配置信息，包括针对以下内容的设置：HTTP 侦听器、ORB/IIOP 侦听器、JMS 代理、EJB 容器、安全性、日志记录和监视功能等。命名配置中未定义应用程序和资源。

配置存在于管理域中。该域中的多个服务器实例或群集可以引用相同的配置，也可以有各自独立的配置。

对于群集，群集中的所有服务器实例都继承群集的配置，从而确保群集实例具有同构环境。

由于配置包含如此多的必需设置，因此请通过复制现有命名配置来创建新配置。更改新建配置的配置设置之前，该配置与被复制的配置完全相同。

群集或实例使用配置的方法有以下三种：

- **独立**：独立服务器实例或群集不与其他服务器实例或群集共享其配置；也就是说，其他服务器实例或群集不引用独立服务器实例或群集的命名配置。您可以通过复制或重命名现有配置来创建独立服务器实例或群集。
- **共享**：共享服务器实例或群集与其他服务器实例或群集共享配置；也就是说，多个实例或群集引用相同的命名配置。您可以通过引用（而非复制）现有配置来创建共享服务器实例或群集。
- **群集**：群集服务器实例继承群集的配置。
另请参见：
 - 第 160 页中的“default-config 配置”
 - 第 160 页中的“创建实例或群集时创建的配置”
 - 第 161 页中的“唯一端口号和配置”
 - 第 162 页中的“创建命名配置”
 - 第 162 页中的“编辑命名配置的属性”

default-config 配置

default-config 配置是一种特殊配置，用作创建独立服务器实例或独立群集配置的模板。群集和单个服务器实例不能引用 default-config；只能复制它来创建新配置。编辑默认配置，以确保从默认配置复制而来的新配置具有正确的初始设置。

有关更多信息，请访问：

- 第 160 页中的“创建实例或群集时创建的配置”
- 第 159 页中的“配置”
- 第 162 页中的“创建命名配置”
- 第 162 页中的“编辑命名配置的属性”
- 第 163 页中的“编辑引用配置的实例的端口号”

创建实例或群集时创建的配置

创建新服务器实例或新群集时，可以执行以下操作之一：

- 引用现有配置。不添加新配置。
- 创建现有配置的副本。添加服务器实例或群集时，将添加新配置。

默认情况下，在创建新群集或实例时，其配置是从 default-config 配置中复制的。要从其他配置进行复制，请在创建新实例或群集时指定要复制的配置。

对于服务器实例，新配置的名称为 *instance_name-config*。对于群集，新配置的名称为 *cluster_name-config*。

有关更多信息，请访问：

- 第 160 页中的 “default-config 配置”
- 第 159 页中的 “配置”
- 第 162 页中的 “创建命名配置”
- 第 162 页中的 “编辑命名配置的属性”

群集配置同步

创建群集配置后，Application Server 会在域管理服务器上创建 `domain-root/domain-dir/config/cluster-config` 群集配置目录。该目录用来同步群集中所有实例的配置。

唯一端口号和配置

如果同一主机上有多个实例引用相同的配置，则每个实例必须在唯一的端口号上进行侦听。例如，如果两个服务器实例都引用某个命名配置，该配置包含一个位于端口 80 上的 HTTP 侦听器，则端口冲突将阻止其中一个服务器实例启动。更改用于定义端口号（各个服务器实例在这些端口号上进行侦听）的属性，从而确保各个实例使用唯一的端口。

以下原则适用于端口号设定：

- 各个服务器实例的端口号最初是从配置继承而来的。
- 创建服务器实例时，如果该端口号已经被使用，则请在实例级别上覆盖继承的默认值，以防止发生端口冲突。
- 假定实例正在共享配置。该配置具有端口号 n 。如果使用相同的配置在计算机上创建新实例，则为新实例指定的端口号为 $n+1$ （如果此端口号可用）。如果此端口号不可用，将选择 $n+1$ 后下一个可用的端口。
- 如果您更改了配置的端口号，则继承该端口号的服务器实例将自动继承更改后的端口号。
- 如果您更改了实例的端口号，然后又更改了配置的端口号，则实例的端口号将保持不变。

有关更多信息，请访问：

- 第 163 页中的 “编辑引用配置的实例的端口号”
- 第 162 页中的 “编辑命名配置的属性”
- 第 159 页中的 “配置”

使用命名配置

- 第 162 页中的 “创建命名配置”
- 第 162 页中的 “编辑命名配置的属性”
- 第 163 页中的 “编辑引用配置的实例的端口号”
- 第 164 页中的 “查看命名配置的目标”
- 第 164 页中的 “删除命名配置”

▼ 创建命名配置

- 1 在树组件中，选择 “配置” 节点。
- 2 在 “配置” 页面中，单击 “新建”。
- 3 在 “创建配置” 页面中，为配置输入唯一名称。
- 4 选择要复制的配置。

default-config 配置是创建独立服务器实例或独立群集时所使用的默认配置。

更多信息 等效的 asadmin 命令

```
copy-config
```

另请参见

- 第 159 页中的 “配置”
- 第 160 页中的 “default-config 配置”
- 第 162 页中的 “编辑命名配置的属性”
- 第 163 页中的 “编辑引用配置的实例的端口号”
- 第 164 页中的 “查看命名配置的目标”
- 第 164 页中的 “删除命名配置”

编辑命名配置的属性

下表介绍了配置的预定义属性。

预定义属性是端口号。有效的端口值为 1-65535。在 UNIX 上，创建在端口 1-1024 上进行侦听的套接字需要超级用户权限。如果系统中存在多个服务器实例，则它们各自的端口号必须是唯一的。

属性名称	说明
HTTP_LISTENER_PORT	http-listener-1 的端口号。
HTTP_SSL_LISTENER_PORT	http-listener-2 的端口号。
IIOp_SSL_LISTENER_PORT	IIOp 侦听器 SSL 侦听的 IIOp 连接的 ORB 侦听器端口。
IIOp_LISTENER_PORT	orb-listener-1 侦听的 IIOp 连接的 ORB 侦听器端口。
JMX_SYSTEM_CONNECTOR_PORT	JMX 连接器侦听的端口号。
IIOp_SSL_MUTUALAUTH_PORT	IIOp 侦听器 SSL_MUTUALAUTH 侦听的 IIOp 连接的 ORB 侦听器端口。

▼ 编辑命名配置的属性

- 1 在树组件中，展开“配置”节点。
- 2 选择命名配置的节点。
- 3 在“配置系统属性”页面中，选择是否启用动态重新配置。
如果已启用，则对配置所作的更改将应用于服务器实例，而无需重新启动服务器。
- 4 根据需要添加、删除或修改属性。
- 5 要编辑与配置关联的所有实例的属性的当前值，请单击“实例值”。

更多信息 等效的 asadmin 命令

set

- 另请参见
- 第 159 页中的“配置”
 - 第 162 页中的“创建命名配置”
 - 第 164 页中的“查看命名配置的目标”
 - 第 164 页中的“删除命名配置”

▼ 编辑引用配置的实例的端口号

每个引用命名配置的实例最初都从该配置继承端口号。由于系统中的端口号必须是唯一的，因而可能需要覆盖继承的端口号。

- 1 在树组件中，展开“配置”节点。

- 2 选择命名配置的节点。
管理控制台将显示“配置系统属性”页面。
- 3 单击要编辑的实例变量旁边的“实例值”。
例如，如果单击实例变量 HTTP-LISTENER-PORT 旁边的“实例值”，将看到引用该配置的每个服务器实例的 HTTP-LISTENER-PORT 的值。
- 4 根据需要更改该值，然后单击“保存”。

更多信息 **等效的 asadmin 命令**

set

- 另请参见
- 第 161 页中的“唯一端口号和配置”
 - 第 159 页中的“配置”
 - 第 162 页中的“编辑命名配置的属性”

▼ 查看命名配置的目标

“配置系统属性”页面将显示使用该配置的所有目标的列表。对于群集配置，目标是群集。对于实例配置，目标是实例。

- 1 在树组件中，展开“配置”节点。
- 2 选择命名配置的节点。

- 另请参见
- 第 161 页中的“唯一端口号和配置”
 - 第 159 页中的“配置”
 - 第 162 页中的“创建命名配置”
 - 第 162 页中的“编辑命名配置的属性”
 - 第 164 页中的“删除命名配置”

▼ 删除命名配置

- 1 在树组件中，选择“配置”节点。
- 2 在“配置”页面中，选中要删除的命名配置的复选框。
不能删除 default-config 配置。
- 3 单击“删除”。

更多信息 **等效的 asadmin 命令**

`delete-config`

- 另请参见
- 第 159 页中的“配置”
 - 第 162 页中的“创建命名配置”
 - 第 162 页中的“编辑命名配置的属性”
 - 第 164 页中的“查看命名配置的目标”

配置节点代理

本章介绍了 Application Server 中的节点代理。它包含以下小节：

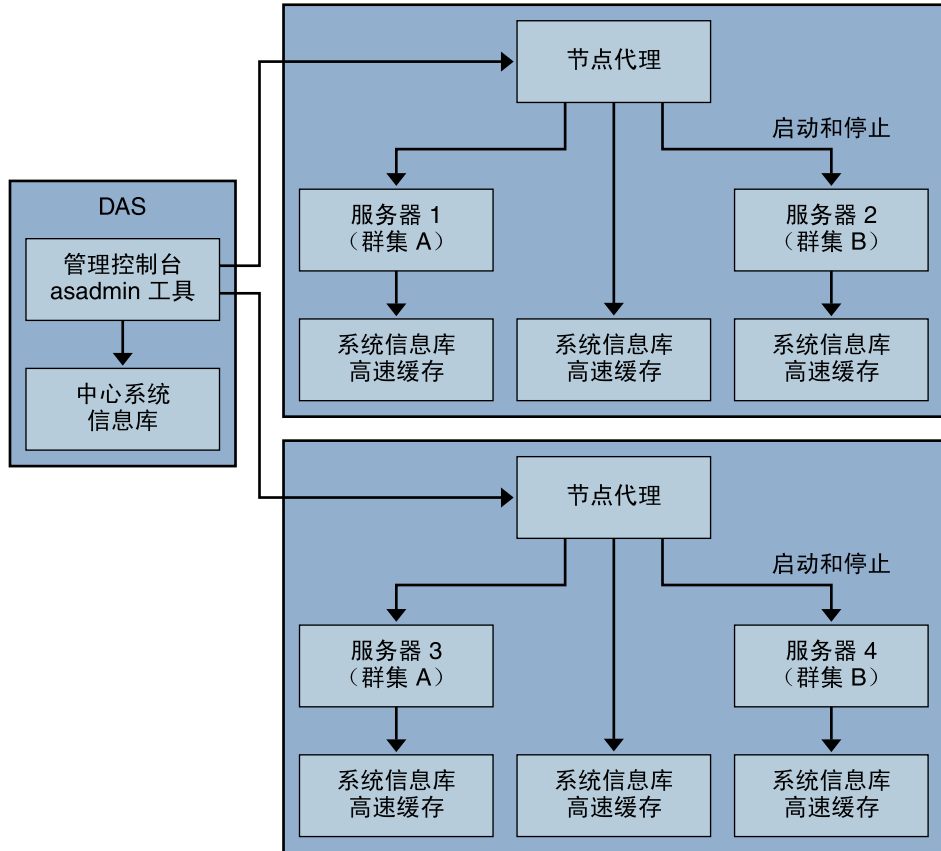
- 第 167 页中的 “什么是节点代理？”
- 第 168 页中的 “节点代理出现故障后的服务器实例行为”
- 第 169 页中的 “部署节点代理”
- 第 171 页中的 “同步节点代理和域管理服务器”
- 第 175 页中的 “查看节点代理日志”
- 第 175 页中的 “处理节点代理”

什么是节点代理？

节点代理是托管服务器实例的每台计算机（包括托管域管理服务器 (Domain Administration Server, DAS) 的计算机）上都需要的轻量级进程。节点代理可以：

- 按照域管理服务器的指示启动、停止、创建和删除服务器实例。
- 重新启动有故障的服务器实例。
- 提供有故障的服务器的日志文件视图。
- 使每个服务器实例的本地配置系统信息库与域管理服务器的中心系统信息库同步。每个本地系统信息库只包含与该服务器实例或节点代理相关的信息。

下图说明了节点代理的整个体系结构：



安装 Application Server 时，默认情况下将使用该计算机的主机名创建节点代理。必须先在本地计算机上手动启动节点代理之后，该节点代理才能运行。

即使未运行节点代理，您也可以创建和删除服务器实例。但是，节点代理必须处于运行状态，您才能用它来启动和停止服务器实例。

一个节点代理服务一个域。如果一台计算机托管在多个域中运行的实例，则该计算机必须运行多个节点代理。

节点代理出现故障后的服务器实例行为

节点代理可能会因某些原因而意外停止，例如软件故障或其他错误。在这种情况下，将不再管理节点代理所管理的所有服务器实例。但是，此服务器实例继续运行，并且 DAS 仍可访问它们。有关服务器实例的信息仍可通过 Application Server 管理接口获得，并且仍可访问在服务器实例上部署的应用程序。

如果重新启动节点代理，则未管理的服务器实例仍处于未管理状态。节点代理不会恢复对这些服务器实例的管理。如果未管理的服务器实例因某些原因意外停止，例如软件故障或其他错误，则节点代理无法重新启动该服务器实例。

如果未管理的服务器实例必须继续运行，则无法恢复节点代理对服务器实例的管理。恢复对未管理的服务器实例管理的唯一方法是，在重新启动节点代理后，先停止该服务器实例，然后再重新启动。

部署节点代理

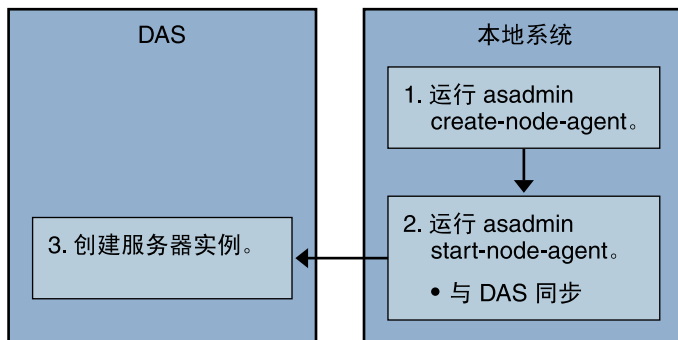
可以使用以下两种方式配置和部署节点代理：

- **联机部署。**如果您已知道域的拓扑并且已经具有用于该域的硬件，则请使用联机部署。
- **脱机部署。**如果是在设置整个环境之前配置域和服务器实例，则请使用脱机部署。

▼ 联机部署节点代理

如果您已经知道域的拓扑并且具有用于该域的硬件，则使用联机部署。

下图总结了节点代理的联机部署：



开始之前 安装并启动域管理服务器。启动并运行域管理服务器后，便可以开始进行联机或脱机部署。

1 在将要托管服务器实例的每台计算机上安装节点代理。

使用安装程序或 `asadmin create-node-agent` 命令。如果某台计算机需要多个节点代理，请使用 `asadmin create-node-agent` 命令来创建这些节点代理。

有关更多信息，请参见第 177 页中的“创建节点代理”。

2 使用 `asadmin start-node-agent` 命令启动节点代理。

启动之后，节点代理将与域管理服务器进行通信 (DAS)。当节点代理到达 DAS 时，将在 DAS 上创建该节点代理的配置。具备了配置之后，即可在管理控制台中查看该节点代理。

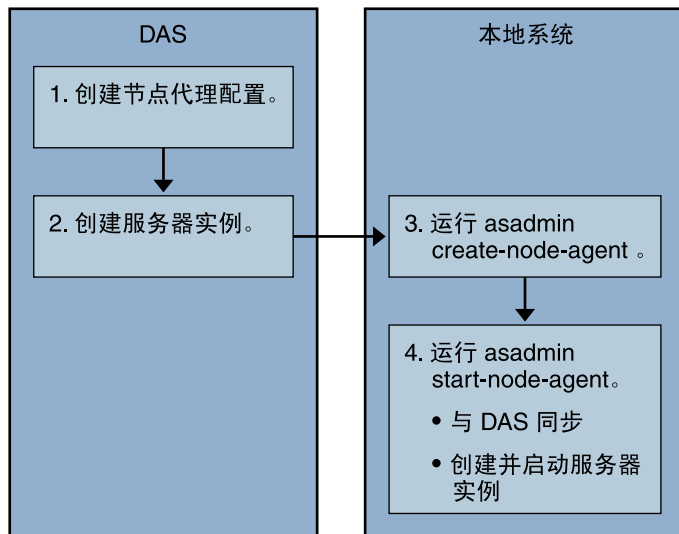
有关更多信息，请参见第 179 页中的“启动节点代理”。

3 配置域：创建服务器实例和群集，并部署应用程序。

▼ 脱机部署节点代理

在配置各个本地计算机之前，使用脱机部署在域中部署节点代理。

下图总结了脱机部署。



开始之前 安装并启动域管理服务器。启动并运行域管理服务器后，便可以开始进行联机或脱机部署。

1 在域管理服务器中创建占位符节点代理。

有关更多信息，请参见第 176 页中的“创建节点代理占位符”。

2 创建服务器实例和群集并部署应用程序。

创建服务器实例时，请确保指定尚未使用的端口号。因为正在进行脱机配置，所以域无法在创建时检查端口冲突。

3 在将要托管服务器实例的每台计算机上安装节点代理。

使用安装程序或 `asadmin create-node-agent` 命令。节点代理的名称必须与先前创建的占位符节点代理的名称相同。

有关更多信息，请参见第 177 页中的“创建节点代理”。

4 使用 `asadmin start-node-agent` 命令启动节点代理。

节点代理启动之后，它将绑定到域管理服务器并创建先前已与该节点代理相关联的所有服务器实例。

有关更多信息，请参见第 179 页中的“启动节点代理”。

同步节点代理和域管理服务器

因为配置数据既存储在域管理服务器的系统信息库（中心系统信息库）中，又高速缓存在节点代理的本地计算机中，所以这两者必须同步。用户应始终通过管理工具执行显式操作来实现高速缓存的同步。

本节包括以下主题：

- 第 171 页中的“节点代理同步”
- 第 172 页中的“服务器实例同步”
- 第 173 页中的“同步库文件”
- 第 174 页中的“独特的设置和配置管理”
- 第 174 页中的“同步大型应用程序”

节点代理同步

第一次启动节点代理时，该节点代理将向域管理服务器 (DAS) 发送一个请求，以获得中心系统信息库中的最新信息。当节点代理成功地与 DAS 取得联系并获得配置信息时，该节点代理即绑定到该 DAS。

注 – 默认情况下，`asadmin start-node-agent` 命令将自动启动未与 DAS 同步的远程服务器实例。如果要启动与 DAS 所管理的中心系统信息库同步的远程服务器实例，请指定 `asadmin start-node-agent` 命令的 `--startinstances=false` 选项。然后使用 `asadmin start-instance` 命令启动远程服务器实例。

如果您在 DAS 上创建了占位符节点代理，则第一次启动节点代理时，该节点代理将从 DAS 的中心系统信息库中获取其配置。初始启动节点代理过程中，如果由于没有运行 DAS 而使该节点代理无法到达 DAS，则该节点代理将停止并保持未绑定状态。

如果在域中更改了节点代理的配置，这些更改将在节点代理运行时自动与本地计算机中的节点代理进行通信。

如果删除了 DAS 中的一个节点代理配置，则下次该节点代理进行同步时将停止它自身并将自身标记为待删除状态。使用本地 `asadmin delete-node-agent` 命令手动将其删除。

服务器实例同步

如果使用管理控制台或 `asadmin` 工具明确启动了服务器实例，则该服务器实例将与中心系统信息库同步。如果此同步失败，则服务器实例不会启动。

如果节点代理未通过向管理控制台或 `asadmin` 工具发送显式请求来启动服务器实例，将不同步该服务器实例的系统信息库高速缓存。该服务器实例将以存储在其高速缓存中的配置运行。您不能在远程服务器实例的高速缓存中添加或删除文件。

远程服务器实例的配置被视为高速缓存（`nodeagents/na1/server1` 下的所有文件）并由 Application Server 所有。在极少数情况下，如果用户删除了远程服务器实例的所有文件并重新启动节点代理，则将重新创建远程服务器实例（例如，`server1`）并同步所有必需的文件。

Application Server 将保持以下文件和目录的同步。

表 8-1 在远程服务器实例之间同步的文件和目录

文件或目录	说明
<code>applications</code>	所有已部署的应用程序。已同步的此目录（及子目录）的部分取决于服务器实例中引用的应用程序。节点代理不会同步任何应用程序，因为它并没有引用任何应用程序。
<code>config</code>	包含整个域的配置文件的目录。此目录中的所有文件均被同步，但运行时临时文件（例如 <code>admch</code> 、 <code>admsn</code> 、 <code>secure.seed</code> 、 <code>timestamp</code> 和 <code>__timer_service_shutdown__.dat</code> ）除外。
<code>config/config_name</code>	所有实例共享的存储文件的目录（使用名为 <code>config_name</code> 的配置）。对于每个在 <code>domain.xml</code> 中定义的配置，都存在这样一个目录。此目录中的所有文件均同步到使用 <code>config_name</code> 的服务器实例。
<code>config/config_name/lib/ext</code>	可向其中放入 Java 扩展类（如 <code>zip</code> 或 <code>jar</code> 归档文件）的文件夹。此文件夹供部署到服务器实例（使用名为 <code>config_name</code> 的配置）的应用程序使用。这些 <code>jar</code> 文件使用 Java 扩展机制装入。
<code>docroot</code>	HTTP 文档根目录。在现有 (out of the box) 配置中，域中的所有服务器实例均使用相同的 <code>docroot</code> 。需要对虚拟服务器的 <code>docroot</code> 属性进行配置，以使服务器实例使用不同的 <code>docroot</code> 。
<code>generated</code>	已生成的 Java EE 应用程序文件和模块文件，例如 EJB 桩模块、已编译的 JSP 类和安全策略文件。此目录与 <code>applications</code> 目录同步。因此，仅同步与服务器实例所引用的应用程序对应的目录。

表 8-1 在远程服务器实例之间同步的文件和目录 (续)

文件或目录	说明
lib、lib/classes	可以丢弃应用程序（部署到整个域）使用的通用 Java 类文件或 jar 和 zip 归档文件所在的文件夹。这些类使用 Application Server 的类加载器装入。类加载器的装入顺序为：lib/classes、lib/*.jar 和 lib/*.zip。
lib/ext	可以丢弃应用程序（部署到整个域）使用的 Java 扩展类（如 zip 或 jar 归档文件）所在的文件夹。这些 jar 文件使用 Java 扩展机制装入。
lib/applibs	将相关 jar 放在 domains/<domain_name>/lib/applibs 下，并通过 libraries 选项指定 jar 文件的相对路径。 例如， <code>asadmin deploy --libraries commons-coll.jar,X1.jar foo.ear</code>
java-web-start	此目录（及子目录）的各部分根据服务器实例引用的应用程序进行同步。

同步库文件

可以使用应用程序的 `--libraries` 部署时间属性来指定应用程序的运行时相关项。指定相对路径（仅 jar 名称）后，Application Server 会尝试在 `domain-dir/lib/applibs` 中查找指定的库。

为使库可用于整个域，可以将 JAR 文件放在 `domain-dir/lib` 或 `domain-dir/lib/classes` 中。（有关更多信息，请参见《Sun Java System Application Server 9.1 Developer's Guide》中的“Using the Common Class Loader”。这种情况通常适用于 JDBC 驱动程序以及域中所有应用程序共享的其他实用程序库。

为了在群集范围或独立服务器范围中使用，请将 jar 复制到 `domain-dir/domain1/config/xyz-config/lib` 目录中。接下来，将 jar 添加在 `xyz-config` 的 `classpath-suffix` 或 `classpath-prefix` 元素中。这将针对所有使用 `xyz-config` 的服务器实例同步 jar。

具体概括如下：

- `domains/domain1/lib`—用于域范围，使用通用类加载器，自动添加 jar。
- `domains/domain1/config/cluster1、config/lib`—用于配置范围，更新 `classpath-prefix` 或 `classpath-suffix`。
- `domains/domain1/lib/applibs`—用于应用程序范围，自动添加到应用程序类加载器。
- `domains/domain1/config/cluster1、config/lib/ext`—自动添加到 <http://java.sun.com/j2se/1.5.0/docs/guide/extensions/extensions.html>。

独特的设置和配置管理

配置文件（位于 `domains/domain1/config` 下）在整个域中进行同步。如果要为独立服务器实例 (`server1`) 使用的 `server1-config` 自定义 `server.policy` 文件，请将修改后的 `server.policy` 文件放在 `domains/domain1/config/server1-config` 目录下。

此修改后的 `server.policy` 文件仅针对独立服务器实例 `server1` 进行同步。应谨记更新 `jvm-option`。例如：`<java-config> ...`

```
<jvm-options>-Djava.security.policy=${com.sun.aas.instanceRoot}/config/
/server1-config/server.policy</jvm-options></java-config>
```

同步大型应用程序

当环境中大型应用程序要进行同步或者可用内存受到限制时，您可以调整 JVM 选项以限制内存的使用。这种调整将减少收到内存不足错误的可能性。实例同步 JVM 使用的是默认设置，但您可以配置 JVM 选项来更改这些设置。

可以使用 `INSTANCE-SYNC-JVM-OPTIONS` 属性设置 JVM 选项。设置属性的命令为：

```
asadmin set
domain.node-agent.node_agent_name.property.INSTANCE-SYNC-JVM-OPTIONS="JVM_options"
```

例如：

```
asadmin set
domain.node-agent.node0.property.INSTANCE-SYNC-JVM-OPTIONS="-Xmx32m -Xss2m"
```

在此示例中，节点代理为 `node0`，JVM 选项为 `-Xmx32m -Xss2m`。

有关更多信息，请参见 <http://java.sun.com/docs/hotspot/VMOptions.html>。

注 - 更改 `INSTANCE-SYNC-JVM-OPTIONS` 属性后，请重新启动节点代理，因为在节点代理的配置中添加或更改了某个属性时节点代理并不会自动同步。

使用 `doNotRemoveList` 标志

如果应用程序需要在 Application Server 同步的目录（`applications`、`generated`、`docroot`、`config`、`lib`、`java-web-start`）中存储和读取文件，请使用 `doNotRemoveList` 标志。此属性接受以逗号分隔的文件或目录列表。即使应用程序相关文件不存在于 DAS 所管理的中心系统信息库中，在服务器启动期间也不会删除这些文件。如果中心系统信息库中存在相同的文件，则在同步期间将会覆写这些文件。

可以使用 `INSTANCE-SYNC-JVM-OPTIONS` 属性传入 `doNotRemoveList` 属性。

例如：

```
<node-agent name="na1" ...>
...
<property name="INSTANCE-SYNC-JVM-OPTIONS"
value="-Dcom.sun.appserv.doNotRemoveList=applications/j2ee-modules
/<webapp_context>/logs,generated/mylogdir"/>

</node-agent>
```

查看节点代理日志

每个节点代理都有自己的日志文件。如果使用节点代理时遇到问题，请参见位于以下位置的日志文件：

`node_agent_dir/node_agent_name/agent/logs/server.log`。

有时，节点代理日志会指示您查看服务器的日志以获得关于所出现的问题的详细信息。

服务器日志位于：

`node_agent_dir/node_agent_name/server_name/logs/server.log`

`node_agent_dir`的默认位置为 `install_dir/nodeagents`。

处理节点代理

- 第 176 页中的“如何执行节点代理任务”
- 第 176 页中的“节点代理占位符”
- 第 176 页中的“创建节点代理占位符”
- 第 177 页中的“创建节点代理”
- 第 179 页中的“启动节点代理”
- 第 179 页中的“停止节点代理”
- 第 180 页中的“删除节点代理”
- 第 180 页中的“查看常规节点代理信息”
- 第 181 页中的“删除节点代理配置”
- 第 182 页中的“编辑节点代理配置”
- 第 182 页中的“编辑节点代理区域”
- 第 182 页中的“编辑节点代理的 JMX 侦听器”

如何执行节点代理任务

有些节点代理任务要求您在运行节点代理的系统上从本地使用 `asadmin` 工具。可以使用管理控制台或 `asadmin` 远程执行其他任务。

下表总结了这些任务以及执行这些任务的方式：

表 8-2 如何执行节点代理任务

任务	管理控制台	asadmin 命令
在域管理服务器中创建节点代理占位符	“创建节点代理占位符” 页面	<code>create-node-agent-config</code>
创建节点代理	不可用	<code>create-node-agent</code>
启动节点代理	不可用	<code>start-node-agent</code>
停止节点代理	不可用	<code>stop-node agent</code>
从域管理服务器中删除节点代理配置	“节点代理” 页面	<code>delete-node-agent-config</code>
从本地计算机中删除节点代理	不可用	<code>delete-node-agent</code>
编辑节点代理配置	“节点代理” 页面	<code>set</code>
列出节点代理	“节点代理” 页面	<code>list-node-agents</code>

节点代理占位符

在没有节点代理的情况下，可以使用节点代理占位符创建和删除服务器实例。节点代理占位符是在节点代理的本地系统中创建节点代理本身之前，在域管理服务器 (Domain Administration Server, DAS) 上创建的。

有关创建节点代理占位符的信息，请参见第 176 页中的“创建节点代理占位符”。

注 - 当您创建占位符节点代理之后，即可使用该占位符节点代理在域中创建实例。但是，在启动实例之前，您必须先使用 `asadmin` 命令在实例将要驻留的计算机上从本地创建并启动实际的节点代理。请参见第 177 页中的“创建节点代理”和第 179 页中的“启动节点代理”。

▼ 创建节点代理占位符

节点代理是在远程计算机上运行的服务器实例的本地监视程序。因此，必须在托管服务器实例的计算机上创建节点代理。为满足此要求，您可以使用管理控制台仅为节点代理创建占位符。此占位符是尚不存在的节点代理的节点代理配置。

创建一个占位符之后，请在托管节点代理的计算机中使用 `asadmin` 命令 `create-node-agent` 完成节点代理的创建。有关更多信息，请参见第 177 页中的“创建节点代理”。

有关创建和使用节点代理所涉及的一系列步骤，请参见第 169 页中的“部署节点代理”。

- 1 在树组件中，选择“节点代理”节点。
- 2 在“节点代理”页面中，单击“新建”。
- 3 在“当前节点代理占位符”页面中为新节点代理输入一个名称。
在域中的所有节点代理名称、服务器实例名称、群集名称和配置名称中，此名称必须是唯一的。
- 4 单击“确定”。
新节点代理的占位符将列出在“节点代理”页面中。

更多信息 等效的 `asadmin` 命令

```
create-node-agent-config
```

创建节点代理

要创建节点代理，请在运行节点代理的本地计算机上运行 `asadmin` 命令 `create-node-agent`。

节点代理的默认名称为创建节点代理时所在的主机的名称。

如果已创建节点代理占位符，请使用与节点代理占位符相同的名称来创建相应的节点代理。如果尚未创建节点代理占位符，而 DAS 已启动并且可访问，则 `create-node-agent` 命令还将在 DAS 上创建节点代理配置（占位符）。

有关命令语法的完整说明，请参见该命令的联机帮助。

可以将 DAS 和节点代理配置为安全地进行通信。在这种情况下，启动节点代理时，节点代理必须验证 DAS 发送给它的证书。为验证证书，节点代理在其本地 `truststore`（受主密码保护）中查找证书。要使节点代理可以在不提示用户输入密码的情况下启动，请在创建节点代理时将节点代理的主密码保存到文件中。如果不将节点代理的主密码保存到文件中，则每当用户启动节点代理时都提示用户输入主密码。

注 - 在某些情况下，必须指定可以通过 DNS 访问的主机的名称。有关更多信息，请参见第 178 页中的“为可通过 DNS 访问的主机创建节点代理”。

▼ 创建节点代理

● 键入以下命令：

```
asadmin create-node-agent --host das-host --port port-no --user das-user
[--savemasterpassword=true] nodeagent
```

要使节点代理可以在不提示用户输入密码的情况下启动，请将节点代理的主密码保存到文件中。要将节点代理的主密码保存到文件中，请在用于创建节点代理的命令中将 `--savemasterpassword` 选项设置为 `true`。

如果将 `--savemasterpassword` 设置为 `true`，将提示您输入主密码。否则，不提示您输入密码。

`--host das-host` 指定运行域管理服务器 (Domain Administration Server, DAS) 的主机名称。

`-port port-no` 指定用于管理域的 HTTP 或 HTTPS 端口号。

`--user das-user` 指定 DAS 用户。

nodeagent 指定要创建的节点代理名称。此名称在域中必须是唯一的。

示例 8-1 创建节点代理

```
asadmin create-node-agent --host myhost --port 4848 --user admin nodeagent1
```

此命令创建一个名为 `nodeagent1` 的节点代理。节点代理与其进行通信的 DAS 正在计算机 `myhost` 上运行。用于管理代理域的 HTTP 端口是 4848。DAS 用户的名称是 `admin`。

▼ 为可通过 DNS 访问的主机创建节点代理

在以下情况下，运行 DAS 的主机必须可通过 DNS 进行访问：

- 各个域跨子网边界，即节点代理和 DAS 在不同的域中，例如 `sun.com` 和 `java.com`。
- 正在使用其主机名未在 DNS 中注册的 DHCP 计算机。

- 1 在用于创建域的 `create-domain` 命令中，指定 `--domainproperties domain.hostName=das-host-name` 选项。
das-host-name 是运行 DAS 的计算机的名称。
- 2 在用于创建节点代理的 `create-node-agent` 命令中，指定以下选项：

- `--host das-host-name`，其中 `das-host-name` 是在步骤 1 中指定的 DNS 主机名。此选项与文件 `as-install/nodeagents/nodeagentname/agent/config/das.properties` 中的 `agent.das.host` 属性相对应。
- `--agentproperties remoteclientaddress=node-agent-host-name`，其中 `node-agent-host-name` 是 DAS 用来连接到节点代理的主机名。此选项与文件 `as-install/nodeagents/nodeagentname/agent/config/nodeagent.properties` 中的 `agent.client.host` 属性相对应。

更多信息 通过更新 hosts 文件指定主机

另外一种解决方案是更新特定于平台的 hosts 主机名/IP 解析文件，从而将主机名解析为正确的 IP 地址。但是，使用 DHCP 重新连接时，可能会为您分配不同的 IP 地址。在这种情况下，您必须更新每个服务器中的主机解析文件。

启动节点代理

节点代理必须先运行才能管理服务器实例。通过在节点代理所驻留的系统中以本地方式运行 `asadmin` 命令 `start-node-agent` 来启动节点代理。

有关命令语法的完整说明，请参见该命令的联机帮助。

例如：

```
asadmin start-node-agent --user admin --startinstances=false nodeagent1
```

其中，`admin` 为管理用户，`nodeagent1` 为要启动的节点代理。

默认情况下，重新启动节点代理时，节点代理实例的高速缓存系统信息库不与中心系统信息库同步。要强制将实例的高速缓存系统信息库与中心系统信息库同步，请将 `asadmin start-node-agent` 命令中的 `--syncinstances` 选项设置为 `true`。

注 - 如果将 `--syncinstances` 选项设置为 `true`，在重新启动节点代理时将同步所有实例的系统信息库。

重新启动节点代理后，使用 `asadmin start-instance` 命令启动服务器实例。

停止节点代理

要停止正在运行的节点代理，请在该节点代理驻留的系统中运行 `asadmin` 命令 `stop-node-agent`。`stop-node-agent` 命令将停止节点代理所管理的所有服务器实例。

有关命令语法的完整说明，请参见该命令的联机帮助。

例如：

```
asadmin stop-node-agent nodeagent1
```

其中，`nodeagent1` 为节点代理的名称。

删除节点代理

删除节点代理之前，必须先停止节点代理。您还可以删除从未启动过或者从未成功地与域管理服务器连接（即尚未绑定）的节点代理。

要删除节点代理文件，请在该节点代理驻留的系统中运行 `asadmin delete-node-agent`。

有关命令语法的完整说明，请参见该命令的联机帮助。

例如：

```
asadmin delete-node-agent nodeagent1
```

其中，`nodeagent1` 是节点代理。

删除节点代理时，还必须使用管理控制台或 `asadmin delete-node-agent-config` 命令从域管理服务器中删除该节点代理的配置。

▼ 查看常规节点代理信息

1 在树组件中，选择“节点代理”节点。

2 单击一个节点代理的名称。

如果节点代理已经存在但并未在此处显示，请在节点代理的主机上使用 `asadmin start-node-agent` 来启动该节点代理。请参见第 179 页中的“启动节点代理”。

3 检查节点代理的主机名。

如果主机名为“未知主机”，则该节点代理没有同域管理服务器 (Domain Administration Server, DAS) 进行初始联系。

4 检查节点代理的状态。

状态可以为：

- **正在运行**：已正确创建节点代理，并且当前该节点代理正在运行。

- **未在运行**：已在本地计算机中创建了节点代理，但该节点代理从未启动过，或启动过但已停止。
- **等待会合**：节点代理是从未在本地计算机中创建的占位符。

请参见第 177 页中的“创建节点代理”和第 179 页中的“启动节点代理”。

5 选择启动节点代理时是否启动实例。

选择“是”，将在启动节点代理时自动启动与该节点代理关联的服务器实例。选择“否”以手动启动这些实例。

6 确定节点代理是否已同域管理服务器进行了联系。

如果节点代理从未与域管理服务器进行过联系，则该节点代理从未被成功启动过。

7 管理与节点代理关联的服务器实例。

如果节点代理正在运行，则通过单击实例名称旁边的复选框并单击“启动”或“停止”来启动或停止实例。

▼ 删除节点代理配置

通过管理控制台，只能删除域中的节点代理配置，而不能删除实际的节点代理。要删除节点代理本身，请在该节点代理的本地计算机中运行 `asadmin` 命令 `delete-node-agent`。有关更多信息，请参见第 180 页中的“删除节点代理”。

删除节点代理配置之前，必须停止该节点代理并且该节点代理不能有任何关联的实例。要停止节点代理，请使用 `asadmin` 命令 `stop-node-agent`。有关更多信息，请参见第 179 页中的“停止节点代理”。

- 1 在树组件中，选择“节点代理”节点。
- 2 在“节点代理”页面中，选中要删除的节点代理旁边的复选框。
- 3 单击“删除”。

更多信息 等效的 `asadmin` 命令

```
delete-node-agent-config
```

▼ 编辑节点代理配置

- 1 在树组件中，展开“节点代理”节点。
- 2 选择要编辑的节点代理配置。
- 3 选中“在启动时启动实例”以在代理启动时启动代理的服务器实例。
您还可以从此页面手动启动和停止实例。

如果此配置用于占位符节点代理，则使用 `asadmin create-node-agent` 创建实际的节点代理时，实际的节点代理将使用此配置。有关创建节点代理的信息，请参见第 177 页中的“创建节点代理”。

如果此配置是现有节点代理的配置，则将自动同步此节点代理配置信息。

▼ 编辑节点代理区域

必须为连接到节点代理的用户设置验证区域。只有管理用户才能访问节点代理。

- 1 在树组件中，展开“节点代理”节点。
- 2 选择要编辑的节点代理配置。
- 3 单击“验证区域”选项卡。
- 4 在“节点代理编辑区域”页面中，输入一个区域。
默认区域为 `admin-realm`，它是在您创建节点代理时创建的。要使用其他区域，请用新区域替换由域控制的所有组件中的区域，否则这些组件将无法正常工作。
- 5 在“类名”字段中，指定实现该区域的 Java 类。
- 6 添加需要的所有属性。
验证区域需要特定于提供商的属性，这些属性会因特定实现的需求而有所不同。

▼ 编辑节点代理的 JMX 侦听器

节点代理使用 JMX 与域管理服务器通信。因此，该节点代理必须具有侦听 JMX 请求的端口和其他侦听器信息。

- 1 在树组件中，展开“节点代理”节点。

- 2 选择要编辑的节点代理配置。
- 3 单击 "JMX" 选项卡。
- 4 在“地址”字段中，输入 IP 地址或主机名。
如果侦听器使用唯一端口值侦听服务器的所有 IP 地址，请输入 0.0.0.0。否则，请输入此服务器的有效 IP 地址。
- 5 在“端口”字段中，键入节点代理的 JMX 连接器将要侦听的端口。
如果 IP 地址为 0.0.0.0，则端口号必须唯一。
- 6 在“JMX 协议”字段中，键入 JMX 连接器支持的协议。
默认值为 rmi_jrmp。
- 7 单击“接受所有地址”旁边的复选框以允许连接到所有 IP 地址。
节点代理将侦听与网卡相关联的特定 IP 地址或侦听所有 IP 地址。如果接受所有地址，则会将值 0.0.0.0 放入“侦听主机地址”属性中。
- 8 在“区域名称”字段中，键入为侦听器处理验证的区域的名称。
在此页面的“安全性”部分中，将侦听器配置为使用 SSL 安全性、TLS 安全性或同时使用 SSL 和 TLS 安全性。
要设置安全侦听器，请执行以下操作：
- 9 在“安全性”字段中选中“启用”框。
默认情况下，启用“安全性”。
- 10 设置客户机验证。
如果需要客户机在使用此侦听器时自行向服务器进行验证，请在“客户机验证”字段中选中“启用”框。
- 11 输入证书昵称。
在“证书昵称”字段中输入现有服务器密钥对和证书的名称。
有关使用证书和 SSL 的信息，请参见管理控制台联机帮助。
- 12 在 "SSL3/TLS" 部分：
 - a. 选取要在侦听器上启用的安全协议。
必须选取 SSL3 或 TLS, 或同时选取两者。

b. 选取协议所使用的加密算法套件。

要启用所有加密算法套件，请选取“所有支持的加密算法套件”。

13 单击“保存”。

配置高可用性会话持久性和故障转移

本章介绍了如何启用和配置高可用性会话持久性。

- 第 185 页中的“会话持久性和故障转移概述”
- 第 187 页中的“设置高可用性会话持久性”
- 第 189 页中的“HTTP 会话故障转移”
- 第 194 页中的“有状态会话 Bean 故障转移”

会话持久性和故障转移概述

Application Server 通过 HTTP 会话数据和有状态会话 Bean (Stateful Session Bean, SFSB) 会话数据的**故障转移**提供高可用性会话持久性。故障转移意味着如果服务器实例或硬件发生故障，另一个服务器实例将接管分布式会话。

要求

分布式会话可以在多个 Sun Java System Application Server 实例中运行，如果：

- 每个服务器实例都可以访问相同的会话状态数据。Application Server 为 HTTP 会话和有状态会话 Bean 数据提供了以下类型的高可用性存储：
 - 群集中其他服务器上的内存中复制。对于群集配置文件，默认情况下启用内存中复制。

使用内存中复制要求启用组管理服务 (Group Management Service, GMS)。有关 GMS 的更多信息，请参见第 149 页中的“组管理服务”。

如果群集中的服务器实例位于不同的计算机上，请确保满足以下先决条件：

- 要确保 GMS 和内存中复制正常工作，这些计算机必须在同一子网上。
- 要确保内存中复制正常工作，必须同步群集中所有计算机上的系统时钟使其尽可能接近。

- 高可用性数据库 (High-availability database, HADB)。有关如何启用此数据库的信息，请参见 `configure-ha-cluster(1)`。

注 - HADB 软件随 Sun Java System Application Server 的 Application Server 独立分发提供。有关 Sun Java System Application Server 的可用分发的信息，请参见《Sun Java System Application Server 9.1 Installation Guide》中的“Distribution Types and Their Components”。HADB 功能仅在企业配置文件中可用。有关配置文件的信息，请参见《Sun Java System Application Server 9.1 管理指南》中的“用法配置文件”。

- 每个服务器实例上均部署有相同的可分布 Web 应用程序。`web.xml` 部署描述符文件的 `web-app` 元素中必须包含 `distributable` 元素。
- Web 应用程序使用高可用性会话持久性。如果将非可分布的 Web 应用程序配置为使用高可用性会话持久性，服务器会将错误写入日志文件。
- 必须使用 `deploy` 或 `deploydir` 命令并且将 `--availabilityenabled` 选项设置为 `true` 来部署 Web 应用程序。有关这些命令的更多信息，请参见 `deploy(1)` 和 `deploydir(1)`。

限制

会话故障转移时，将丢失所有对打开文件或网络连接的引用。编码应用程序时必须考虑到此限制。

您只能将某些对象绑定到支持故障转移的分布式会话中。与 Servlet 2.4 规范不同，如果将不支持故障转移的对象类型绑定到分布式会话中，Sun Java System Application Server 不会抛出 `IllegalArgumentException`。

您可以将以下对象绑定到支持故障转移的分布式会话中：

- 所有 EJB 组件的本地主引用和对象引用。
- 共存的无状态会话、有状态会话或实体 Bean 引用。
- 分布式无状态会话、有状态会话或实体 Bean 引用。
- `InitialContext` 和 `java:comp/env` 的 JNDI 上下文。
- `UserTransaction` 对象。但是，如果从未重新启动失败的实例，则会丢失所有就绪的全局事务，并且可能无法正确回滚或提交这些事务。
- 可序列化的 Java 类型。

您不能将以下对象类型绑定到支持故障转移的会话中：

- JDBC 数据源
- Java 消息服务 (Java Message Service, JMS) `ConnectionFactory` 和 `Destination` 对象

- JavaMail™ 会话
- 连接工厂
- 受管对象
- Web 服务引用

对于这些对象，故障转移通常不起作用。但是，在某些情况下故障转移也可能发挥作用，例如，如果对象为可序列化对象。

设置高可用性会话持久性

本节介绍了如何设置高可用性会话持久性，其中包括以下主题：

- [第 187 页中的“设置高可用性会话持久性”](#)
- [第 188 页中的“启用会话可用性”](#)

▼ 设置高可用性会话持久性

开始之前 高可用性会话持久性与动态部署、动态重新装入和自动部署不兼容。这些功能适用于开发环境而不适用于生产环境，因此您必须在启用 HA 会话持久性之前禁用它们。有关如何禁用这些功能的信息，请参见《Sun Java System Application Server 9.1 Application Deployment Guide》。

1 创建 Application Server 群集。

有关更多信息，请参见[第 151 页中的“创建群集”](#)。

2 如果要使用 HADB 存储会话状态数据，请为群集创建 HADB 数据库。

注 - HADB 软件随 Sun Java System Application Server 的 Application Server 独立分发提供。有关 Sun Java System Application Server 的可用分发的信息，请参见《Sun Java System Application Server 9.1 Installation Guide》中的“Distribution Types and Their Components”。HADB 功能仅在企业配置文件中可用。有关配置文件的信息，请参见《Sun Java System Application Server 9.1 管理指南》中的“用法配置文件”。

如果要将群集中其他服务器上的内存中复制用于会话状态数据，请省略此步骤。

有关创建 HADB 数据库的更多信息，请参见 `configure-ha-cluster(1)`。

3 为群集设置 HTTP 负载平衡。

有关更多信息，请参见[第 122 页中的“设置 HTTP 负载平衡”](#)。

4 启用所需的应用程序服务器实例和 Web 或 EJB 容器的可用性。

然后配置会话持久性设置。选择以下方法之一：

- 使用管理控制台。请参见第 189 页中的“启用服务器实例的可用性”。
 - 使用 `asadmin` 命令行实用程序。请参见 `set(1)` 和 `configure-ha-persistence(1)`。
- 5 重新启动群集中的每个服务器实例。

如果实例当前正在处理请求，请在重新启动该实例前停止它，这样，它就有足够的时间来处理它正在处理的请求。有关更多信息，请参见第 133 页中的“禁用（停止）服务器实例或群集”。
 - 6 启用所有特定 SFSB 所必需的可用性。

选择需要进行会话状态检查点操作的方法。请参见第 197 页中的“为单个 Bean 配置可用性”。
 - 7 如果要使每个 Web 模块具有高可用性，请使这些模块可分布。
 - 8 在部署期间启用各个应用程序、Web 模块或 EJB 模块的可用性。

请参见第 196 页中的“为单个应用程序或 EJB 模块配置可用性”。

在管理控制台中，选中“启用高可用性”框，或结合使用 `asadmin deploy` 命令和设置为 `true` 的 `--availabilityenabled` 选项。

启用会话可用性

您可以启用五个不同范围（从最高到最低）的会话可用性：

1. 服务器实例，默认情况下已启用。启用服务器实例的会话可用性意味着此服务器实例上运行的所有应用程序都会具有高可用性会话持久性。有关说明，请参见下一节第 189 页中的“启用服务器实例的可用性”。
2. 容器（Web 或 EJB），默认情况下已启用。有关启用容器级别的可用性的信息，请参见：
 - 第 190 页中的“为 Web 容器配置可用性”
 - 第 195 页中的“为 EJB 容器配置可用性”
3. 应用程序，默认情况下已禁用。
4. 独立的 Web 或 EJB 模块，默认情况下已禁用。
5. 单个 SFSB，默认情况下已禁用。

要启用指定范围内的可用性，您必须同时启用所有更高级别的可用性。例如，要启用应用程序级别的可用性，必须在服务器实例级别和容器级别启用可用性。

给定级别的默认值是其上一级别的设置值。例如，如果已启用容器级别的可用性，则默认情况下，启用应用程序级别的可用性。

如果禁用了服务器实例级别的可用性，则启用其他任何级别的可用性都不生效。如果启用了服务器实例级别的可用性，则将启用所有级别的可用性，除非已明确禁用。

启用服务器实例的可用性

要启用服务器实例的可用性，请使用 `asadmin set` 命令将配置的 `availability-service.availability-enabled` 属性设置为 `true`。

例如，如果 `config1` 为配置名称：

```
asadmin set --user admin --passwordfile password.txt
--host localhost
--port 4849
config1.availability-service.availability-enabled="true"
```

▼ 使用管理控制台启用服务器实例的可用性

- 1 在树组件中，展开“配置”节点。
- 2 展开要编辑的配置的节点。
- 3 选择“可用性服务”节点。
- 4 在“可用性服务”页面中，选中“可用性服务”框以启用实例级别的可用性。

要禁用可用性，请取消选中该框。

此外，如果更改了用于连接 HADB 的 JDBC 资源以获得会话持久性，您还可以更改存储池名称。有关详细信息，请参见 `configure-ha-cluster(1)`。

注 – HADB 软件随 Sun Java System Application Server 的 Application Server 独立分发提供。有关 Sun Java System Application Server 的可用分发的信息，请参见《Sun Java System Application Server 9.1 Installation Guide》中的“Distribution Types and Their Components”。HADB 功能仅在企业配置文件中可用。有关配置文件的信息，请参见《Sun Java System Application Server 9.1 管理指南》中的“用法配置文件”。

- 5 单击“保存”按钮。
- 6 停止并重新启用服务器实例。

HTTP 会话故障转移

Java EE 应用程序通常具有大量会话状态数据。Web 购物车是一个典型的会话状态示例。此外，应用程序可以高速缓存会话对象中需要频繁使用的数据。事实上，几乎带有重要用户交互的所有应用程序都需要维护会话状态。

为 Web 容器配置可用性

注 - HADB 软件随 Sun Java System Application Server 的 Application Server 独立分发提供。有关 Sun Java System Application Server 的可用分发的信息，请参见《Sun Java System Application Server 9.1 Installation Guide》中的“Distribution Types and Their Components”。HADB 功能仅在企业配置文件中可用。有关配置文件的信息，请参见《Sun Java System Application Server 9.1 管理指南》中的“用法配置文件”。

如果要使用 HADB，请通过 `asadmin configure-ha-persistence` 启用和配置 Web 容器可用性。有关此命令的更多信息，请参见 `configure-ha-persistence(1)`。

或者，使用 `asadmin set` 命令将配置的 `availability-service.web-container-availability.availability-enabled` 属性设置为 `true`，然后使用 `configure-ha-persistence` 根据需要设置属性。

注 - 如果要使用内存中复制来存储会话状态数据，**必须**使用 `asadmin set` 命令启用 Web 容器可用性和设置属性。`configure-ha-persistence` 命令**只能**与 HADB 一起使用。

例如，按如下所示使用 `set` 命令，其中 `config1` 为配置名称：

```
asadmin set --user admin --passwordfile password.txt
--host localhost --port 4849
config1.availability-service.web-container-availability.availability-enabled="true"
asadmin configure-ha-persistence --user admin --passwordfile secret.txt
--type ha
--frequency web-method
--scope modified-session
--store jdbc/hastore
--property maxSessions=1000:reapIntervalSeconds=60 cluster1
```

▼ 使用管理控制台启用 Web 容器的可用性

- 1 在树组件中，选择所需配置。
- 2 单击“可用性服务”。
- 3 选择“Web 容器可用性”选项卡。
选中“可用性服务”框以启用可用性。要禁用可用性，请取消选中该框。
- 4 更改其他设置，如下一节第 191 页中的“可用性设置”所述。
- 5 重新启动服务器实例。

可用性设置

注 – HADB 软件随 Sun Java System Application Server 的 Application Server 独立分发提供。有关 Sun Java System Application Server 的可用分发的信息，请参见《Sun Java System Application Server 9.1 Installation Guide》中的“Distribution Types and Their Components”。HADB 功能仅在企业配置文件中可用。有关配置文件的信息，请参见《Sun Java System Application Server 9.1 管理指南》中的“用法配置文件”。

通过“可用性服务”的“Web 容器可用性”选项卡，您可以更改这些可用性设置：

持久性类型：指定已启用可用性的 Web 应用程序的会话持久性机制。允许的值有 memory（无持久性）、file（文件系统）、replicated（其他服务器上的内存）和 ha（HADB）。

在使用 ha 会话持久性之前，必须先配置和启用 HADB。有关配置的详细信息，请参见 `configure-ha-cluster(1)`。

如果已启用 Web 容器可用性，则默认的持久性类型取决于配置文件，如下表所示。

配置文件	持久性类型
开发者	memory
群集	replicated
企业	ha

对于需要会话持久性的生产环境，请使用 ha 或 replicated。memory 持久性类型和 file 持久性类型不提供高可用性会话持久性。

如果已禁用 Web 容器可用性，则默认的持久性类型是 memory。

持久性频率：指定存储会话状态的频率。仅当持久性类型为 ha 或 replicated 时适用。允许的值包括：

- `web-method` – 将响应发送回客户机之前，将在每个 Web 请求结束时存储会话状态。此模式为发生故障时完全更新会话状态提供了最好的保证。这是默认选项。
- `time-based` – 在后台按照 `reapIntervalSeconds` 存储属性设置的频率存储会话状态。此模式不能保证完全更新会话状态。但是，它可以提供很大的性能改善，因为在每个请求之后都不存储状态。

持久性范围：指定会话对象的数目和存储会话状态的频率。仅当持久性类型为 ha 或 replicated 时适用。允许的值包括：

- `session` – 每次都存储整个会话状态。此模式为正确存储任何可分发 Web 应用程序的会话数据提供了最好的保证。这是默认选项。

- `modified-session`—如果会话状态已被修改，则存储整个会话状态。如果调用了 `HttpSession.setAttribute()` 或 `HttpSession.removeAttribute()`，则系统将认为会话已被修改。您必须保证每次更改属性时都调用 `setAttribute()`。这不是 Java EE 规范的要求，但是此模式需要这样做才能正常工作。
- `modified-attribute`—仅存储修改后的会话属性。要使此模式正常工作，您必须遵循一些指导原则：
 - 每次修改会话状态时都调用 `setAttribute()`。
 - 确保各属性之间没有交叉引用。系统将对每个不同的属性关键字的对象图形单独进行序列化并单独存储。如果每个单独的关键字下的对象之间有对象交叉引用，则它们将不会被正确序列化和反序列化。
 - 在多个属性之间分布会话状态，或者至少在只读属性和可修改属性之间分布会话状态。

单点登录状态：选中此框以启用单点登录状态的持久性。要禁用可用性，请取消选中该框。有关更多信息，请参见第 193 页中的“结合使用会话故障转移和单点登录”。

HTTP 会话存储：如果更改了用于连接 HADB 的 JDBC 资源以获得会话持久性，您可以更改 HTTP 会话存储。有关详细信息，请参见 `configure-ha-cluster(1)`。

为单个 Web 应用程序配置可用性

要启用和配置单个 Web 应用程序的可用性，请编辑应用程序部署描述符文件 `sun-web.xml`。应用程序的部署描述符中的设置将覆盖 Web 容器的可用性设置。

`session-manager` 元素的 `persistence-type` 属性用于确定应用程序使用的会话持久性的类型。必须将其设置为 `ha` 或 `replicated` 以启用高可用性会话持久性。

有关 `sun-web.xml` 文件的更多信息，请参见《Sun Java System Application Server 9.1 Application Deployment Guide》中的“The sun-web.xml File”。

示例

```
<sun-web-app> ...
  <session-config>
    <session-manager persistence-type=ha>
      <manager-properties>
        <property name=persistenceFrequency value=web-method />
      </manager-properties>
      <store-properties>
        <property name=persistenceScope value=session />
      </store-properties>
    </session-manager> ...
  </session-config> ...
```


结合使用会话故障转移和单点登录

在单个应用程序服务器实例中，如果一个应用程序验证了用户，则运行在同一个实例上的其他应用程序不会对用户进行重新验证。这称为**单点登录**。有关更多信息，请参见《Sun Java System Application Server 9.1 Developer's Guide》中的“User Authentication for Single Sign-on”。

为了使 HTTP 会话故障转移到群集中另一个实例后该功能仍然可用，必须将单点登录信息保留在 HADB 中。要保留单点登录信息，请首先启用服务器实例和 Web 容器的可用性，然后启用单点登录状态故障转移。

注 - HADB 软件随 Sun Java System Application Server 的 Application Server 独立分发提供。有关 Sun Java System Application Server 的可用分发的信息，请参见《Sun Java System Application Server 9.1 Installation Guide》中的“Distribution Types and Their Components”。HADB 功能仅在企业配置文件中可用。有关配置文件的信息，请参见《Sun Java System Application Server 9.1 管理指南》中的“用法配置文件”。

可以通过管理控制台在“可用性服务”的“Web 容器可用性”选项卡中启用单点登录状态故障转移，如第 190 页中的“为 Web 容器配置可用性”中所述。也可以使用 `asadmin set` 命令将配置的 `availability-service.web-container-availability.sso-failover-enabled` 属性设置为 `true`。

例如，按如下所示使用 `set` 命令，其中 `config1` 为配置名称：

```
asadmin set --user admin --passwordfile password.txt
--host localhost --port 4849
config1.availability-service.web-container-availability.
sso-failover-enabled="true"
```

单点登录组

可以通过单个名称和密码的组合进行访问的应用程序组成了**单点登录组**。对于与应用程序（是单点登录组的一部分）相对应的 HTTP 会话，如果其中一个会话超时，其他会话并不会失效，并且仍然可用。这是因为一个会话的超时不应影响其他会话的可用性。

作为这个行为的推论，如果一个会话超时并且您尝试从运行该会话的同一浏览器窗口访问相应的应用程序，则您无需再次进行验证。但是，将创建一个新的会话。

以属于含有其他两个应用程序的单点登录组的购物车应用程序为例。假设其他两个应用程序的会话超时值大于购物车应用程序的会话超时值。如果购物车应用程序的会话超时，并且您尝试从运行该会话的同一浏览器窗口运行购物车应用程序，则您无需再次进行验证。但是，前一个购物车将丢失，并且您必须创建一个新的购物车。即使运行购物车应用程序的会话已超时，其他两个应用程序也会继续照常运行。

类似地，可假定与其他两个应用程序中的任何一个应用程序对应的会话超时。当从运行该会话的同一浏览器窗口连接应用程序时，您无需再次进行验证。

注-此行为仅适用于会话超时情况。如果启用了单点登录，并且您使用 `HttpSession.invalidate()` 令其中一个会话失效，则属于单点登录组的所有应用程序的会话都将失效。如果您尝试访问属于单点登录组的任一应用程序，则需要再次进行验证，系统将为访问该应用程序的客户机创建一个新的会话。

有状态会话 Bean 故障转移

有状态会话 Bean (SFSB) 包含特定于客户机的状态。客户机和有状态会话 Bean 之间存在一对一关系。创建时，EJB 容器赋予每个 SFSB 唯一的会话 ID，以便将其绑定到客户机。

服务器实例出现故障时，SFSB 的状态可以保存在持久性存储中。在 SFSB 的生命周期中，其预定义点处的状态将被保存到持久性存储库中。这称为**检查点操作**。如果已启用，通常会在 Bean 完成了任意事务之后（即使该事务回滚）执行检查点操作。

但是，如果 SFSB 参与了 Bean 管理的事务，则该事务可能会在 Bean 方法执行的中间就被提交。由于 Bean 的状态可能因方法调用而发生转换，因此这不是对 Bean 的状态进行检查点操作的适当时间。在此情况下，EJB 容器将在相应方法结束时对 Bean 的状态进行检查点操作，前提是当该方法结束时，该 Bean 不在另一个事务的范围内。如果 Bean 管理的事务涉及多个方法，则检查点操作将被延迟，直至在后继方法结束时没有活动的事务。

SFSB 的状态并不一定具有事务性，并且可能因非事务性业务方法而被显著修改。如果某个 SFSB 出现这种情况，则您可以指定一系列检查点方法，如第 197 页中的“[指定对其执行检查点操作的方法](#)”所述。

如果可分布的 Web 应用程序引用了 SFSB，并且 Web 应用程序的会话发生故障转移，则 EJB 引用也将进行故障转移。

如果停止 Application Server 实例时取消部署使用会话持久性的 SFSB，则持久性存储中的会话数据可能不被清除。要避免这种情况，请在运行 Application Server 实例时取消部署 SFSB。

为 EJB 容器配置可用性

▼ 启用 EJB 容器的可用性

- 1 选择“EJB 容器可用性”选项卡。
- 2 选中“可用性服务”框。
要禁用可用性，请取消选中该框。
- 3 更改其他设置，如第 196 页中的“可用性设置”中所述。
- 4 单击“保存”按钮。
- 5 重新启动服务器实例。

更多信息 等效的 asadmin 命令

要启用 EJB 容器的可用性，请使用 `asadmin set` 命令设置以下三个配置属性：

- `availability-service.ejb-container-availability.availability-enabled`
- `availability-service.ejb-container-availability.sfsb-persistence-type`
- `availability-service.ejb-container-availability.sfsb-ha-persistence-type`

例如，如果 `config1` 为配置名称，请使用以下命令：

```
asadmin set --user admin --passwordfile password.txt --host localhost --port 4849config1.availability-service.ejb-container-availability.availability-enabled="true"
```

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

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

可用性设置

注 – HADB 软件随 Sun Java System Application Server 的 Application Server 独立分发提供。有关 Sun Java System Application Server 的可用分发的信息，请参见《Sun Java System Application Server 9.1 Installation Guide》中的“Distribution Types and Their Components”。HADB 功能仅在企业配置文件中可用。有关配置文件的信息，请参见《Sun Java System Application Server 9.1 管理指南》中的“用法配置文件”。

通过“可用性服务”的“EJB 容器可用性”选项卡，您可以更改这些设置：

HA 持久性类型：为已启用可用性的 SFSB 指定会话持久性和钝化机制。允许的值有 `file`（文件系统）、`replicated`（其他服务器上的内存）和 `ha`（HADB）。默认值是 `ha`。对于需要会话持久性的生产环境，请使用 `ha` 或 `replicated`。

SFSB 持久性类型：为尚未启用可用性的 SFSB 指定钝化机制。允许的值有 `file`（默认值）、`replicated` 和 `ha`。

如果将任一持久性类型设置为 `file`，EJB 容器都将指定用于存储已钝化的会话 Bean 状态的文件系统位置。文件系统的检查点操作对于测试很有用，但不适用于生产环境。有关配置存储属性的信息，请参见管理控制台联机帮助。

HA 持久性可以允许服务器实例的群集在任何服务器实例出现故障时恢复 SFSB 状态。HADB 还可以用作钝化和活化存储。在要求 SFSB 状态持久性的生产环境中使用此选项。有关更多信息，请参见 `configure-ha-cluster(1)`。

SFSB 存储池名称：如果更改了用于连接 HADB 的 JDBC 资源以获得会话持久性，您可以更改 SFSB 存储池名称。有关详细信息，请参见 `configure-ha-cluster(1)`。

在禁用可用性时配置 SFSB 会话存储

如果已禁用可用性，本地文件系统将用于 SFSB 状态钝化而非持久性。要更改 SFSB 状态的存储位置，请更改 EJB 容器中的“会话存储位置”设置。有关配置存储属性的信息，请参见管理控制台联机帮助。

为单个应用程序或 EJB 模块配置可用性

您可以在部署期间启用单个应用程序或 EJB 模块的 SFSB 可用性：

- 如果要使用管理控制台进行部署，请选中“启用可用性”复选框。
- 如果要使用 `asadmin deploy` 或 `asadmin deploydir` 命令进行部署，请将 `--availabilityenabled` 选项设置为 `true`。有关更多信息，请参见 `deploy(1)` 和 `deploydir(1)`。

为单个 Bean 配置可用性

要为单个 SFSB 启用可用性并选择执行检查点操作的方法，请使用 `sun-ejb-jar.xml` 部署描述符文件。

要启用高可用性会话持久性，请在 `ejb` 元素中设置 `availability-enabled="true"`。要控制 SFSB 高速缓存的大小和行为，请使用以下元素：

- `max-cache-size`：指定高速缓存中容纳的会话 Bean 的最大数目。如果高速缓存溢出（Bean 的数目超过了 `max-cache-size`），则容器将钝化某些 Bean 或将 Bean 的序列化状态写入文件。创建文件的目录可以使用配置 API 从 EJB 容器处获取。
- `resize-quantity`
- `cache-idle-timeout-in-seconds`
- `removal-timeout-in-seconds`
- `victim-selection-policy`

有关 `sun-ejb-jar.xml` 的更多信息，请参见《Sun Java System Application Server 9.1 Application Deployment Guide》中的“The `sun-ejb-jar.xml` File”。

示例 9-1 已启用可用性的 EJB 部署描述符示例

```
<sun-ejb-jar>
  ...
  <enterprise-Beans>
    ...
    <ejb availability-enabled="true">
      <ejb-name>MySFSB</ejb-name>
    </ejb>
    ...
  </enterprise-Beans>
</sun-ejb-jar>
```

指定对其执行检查点操作的方法

如果已启用，通常会在 Bean 完成了任意事务之后（即使该事务回滚）执行检查点操作。要指定非事务性业务方法（导致对 Bean 状态的重要修改）结束时 SFSB 的其他可选检查点操作，请使用 `sun-ejb-jar.xml` 部署描述符文件的 `ejb` 元素中的 `checkpoint-at-end-of-method` 元素。

`checkpoint-at-end-of-method` 元素中的非事务性方法可以是：

- 在 SFSB 的主接口中定义的 `create()` 方法（如果要在创建后立即对 SFSB 的初始状态进行检查点操作）
- 对于仅使用容器管理的事务的 SFSB，Bean 的远程接口中使用事务属性 `TX_NOT_SUPPORTED` 或 `TX_NEVER` 进行标记的方法

- 对于仅使用 Bean 管理的事务的 SFSB，既不启动也不提交 Bean 管理的事务的方法此列表中提到的所有其他方法均忽略。当调用这些方法结束时，EJB 容器会将 SFSB 的状态保存到持久性存储中。

注 - 如果 SFSB 没有参与任何事务，并且没有在 `checkpoint-at-end-of-method` 元素中明确指定其任何方法，则将不对 Bean 的状态进行检查点操作（即使对于此 Bean，已设置 `availability-enabled="true"`）。

请指定方法的较小子集以获得更好的性能。方法应完成大量工作或导致对 Bean 状态的重要修改。

示例 9-2 指定方法检查点操作的 EJB 部署描述符示例

```
<sun-ejb-jar>
  ...
  <enterprise-Beans>
    ...
    <ejb availability-enabled="true">
      <ejb-name>ShoppingCartEJB</ejb-name>
      <checkpoint-at-end-of-method>
        <method>
          <method-name>addToCart</method-name>
        </method>
      </checkpoint-at-end-of-method>
    </ejb>
    ...
  </enterprise-beans>
</sun-ejb-jar>
```

Java 消息服务的负载平衡和故障转移

本章介绍了如何配置 Java 消息服务 (Java Message Service, JMS) 的负载平衡和故障转移以与 Application Server 一起使用。本章包含以下主题：

- 第 199 页中的 “Java 消息服务概述”
- 第 200 页中的 “配置 Java 消息服务”
- 第 202 页中的 “连接池和故障转移”
- 第 204 页中的 “结合使用 Application Server 与 MQ 群集”

Java 消息服务概述

Java 消息服务 (Java Message Service, JMS) API 是一种通讯标准，使 Java EE 应用程序和组件可以创建、发送、接收和读取消息，并启用了松散耦合的可靠异步分布式通信。Sun Java System Message Queue (MQ) (实现了 JMS) 与 Application Server 紧密集成，使您可以创建诸如消息驱动 bean (message-driven bean, MDB) 之类的组件。

MQ 使用**连接器模块**（也称为资源适配器，由 Java EE 连接器体系结构规范 1.5 定义）与 Application Server 集成。部署到 Application Server 的 Java EE 组件使用通过连接器模块集成的 JMS 提供者交换 JMS 消息。在 Application Server 中创建 JMS 资源将会在后台创建连接器资源。因此，每个 JMS 操作将调用连接器运行时并在后台使用 MQ 资源适配器。

您可以通过管理控制台或 `asadmin` 命令行实用程序管理 Java 消息服务。

更多信息

有关配置 JMS 资源的更多信息，请参见《Sun Java System Application Server 9.1 管理指南》中的第 4 章“配置 Java 消息服务资源”。有关 JMS 的更多信息，请参见《Sun Java System Application Server 9.1 Developer's Guide》中的第 18 章“Using the Java Message Service”。有关连接器（资源适配器）的更多信息，请参见《Sun Java System Application Server 9.1 Developer's Guide》中的第 12 章“Developing Connectors”。

有关 Sun Java System Message Queue 的更多信息，请参见 (<http://docs.sun.com/coll/1343.4>) 和 [Message Queue 文档](#) (<http://docs.sun.com/coll/1578.2>)。有关 JMS API 的常规信息，请参见 [JMS Web 页](#) (<http://java.sun.com/products/jms/index.html>)。

配置 Java 消息服务

Java 消息服务配置可用于到 Sun Java System Application Server 群集或实例的所有入站和出站连接。您可以使用以下方法配置 Java 消息服务：

- 管理控制台。在相关配置下打开 Java 消息服务组件。有关详细信息，请参见《Sun Java System Application Server 9.1 管理指南》中的第 4 章“配置 Java 消息服务资源”。
- `asadmin set` 命令。您可以设置以下属性：

```
server.jms-service.init-timeout-in-seconds = 60
server.jms-service.type = LOCAL
server.jms-service.start-args =
server.jms-service.default-jms-host = default_JMS_host
server.jms-service.reconnect-interval-in-seconds = 60
server.jms-service.reconnect-attempts = 3
server.jms-service.reconnect-enabled = true
server.jms-service.addresslist-behavior = random
server.jms-service.addresslist-iterations = 3
server.jms-service.mq-scheme = mq
server.jms-service.mq-service = jms
```

您也可以设置以下属性：

```
server.jms-service.property.instance-name = imqbroker
server.jms-service.property.instance-name-suffix =
server.jms-service.property.append-version = false
```

使用 `asadmin get` 命令列出所有 Java 消息服务属性。有关 `asadmin get` 的更多信息，请参见 `get(1)`。有关 `asadmin set` 的更多信息，请参见 `set(1)`。

您可以使用 JMS 连接工厂设置覆盖 Java 消息服务配置。有关详细信息，请参见《Sun Java System Application Server 9.1 管理指南》中的“JMS 连接工厂”。

注 – 更改 Java 消息服务的配置之后，您必须重新启动 Application Server 实例。

有关 JMS 管理的更多信息，请参见《Sun Java System Application Server 9.1 管理指南》中的第 4 章“配置 Java 消息服务资源”。

Java 消息服务集成

MQ 可以通过三种方法与 Application Server 集成：LOCAL、REMOTE 和 EMBEDDED。这些模式在管理控制台中由 Java 消息服务的“类型”属性表示。

LOCAL Java 消息服务

“类型”属性为 LOCAL（群集实例的默认值）时，Application Server 将启动和停止指定为默认 JMS 主机的 MQ 代理。MQ 进程从 Application Server 进程在单独的 VM 中于进程外启动。Application Server 为代理提供一个额外的端口。此端口将由代理用来启动 RMI 注册表。此端口号将等于该实例的已配置 JMS 端口号加上 100。例如，如果 JMS 端口号是 37676，则此附加端口号将是 37776。

要在 Application Server 实例和 Message Queue 代理之间创建一对一关系，请将类型设置为 LOCAL 并为每个 Application Server 实例提供一个不同的默认 JMS 主机。无论是否已在 Application Server 或 MQ 中定义了群集，都可以进行此操作。

在类型为 LOCAL 的情况下，请使用“启动参数”属性指定 MQ 代理的启动参数。

REMOTE Java 消息服务

“类型”属性为 REMOTE 时，必须单独启动 MQ 代理。有关启动代理的信息，请参见 Sun Java System Message Queue Administration Guide。

在此情况下，Application Server 将使用从外部配置的代理或代理群集。另外，您必须从 Application Server 单独启动和停止 MQ 代理，并使用 MQ 工具配置和调整代理或代理群集。REMOTE 类型是适用于 Application Server 群集的最佳类型。

在类型为 REMOTE 的情况下，您必须使用 MQ 工具指定 MQ 代理启动参数。忽略“启动参数”属性。

EMBEDDED Java 消息服务

JMS “类型”属性为 EMBEDDED 时，表示应用服务器和 JMS 代理位于同一 VM 中，且 JMS 服务在进程中启动并由 Application Server 管理。在此模式下，JMS 操作绕过网络栈，从而优化了性能。

JMS 主机列表

JMS 主机用于表示 MQ 代理。Java 消息服务包含 *JMS 主机列表*（也称为 AddressList），该主机列表包含 Application Server 使用的所有 JMS 主机。

JMS 主机列表包含指定的 MQ 代理的主机和端口，并且在 JMS 主机配置更改后随时更新。当您创建 JMS 资源或部署 MDB 时，它们将继承 JMS 主机列表。

注 – 在 Sun Java System Message Queue 软件中，AddressList 属性被称为 imqAddressList。

缺省 JMS 主机

JMS 主机列表中的一个主机被指定为默认 JMS 主机，命名为 Default_JMS_host。当 Java 消息服务类型配置为 LOCAL 时，Application Server 实例将启动默认 JMS 主机。

如果您已在 Sun Java System Message Queue 软件中创建了多代理群集，则删除默认 JMS 主机，然后将 Message Queue 群集的代理添加为 JMS 主机。在此情况下，默认 JMS 主机将成为 JMS 主机列表中的第一个主机。

Application Server 使用 Message Queue 群集时，将在默认 JMS 主机上执行 Message Queue 特定命令。例如，为三个代理的 Message Queue 群集创建物理目的地时，将在默认 JMS 主机上执行创建物理目的地的命令，但是该物理目的地将被群集中的全部三个代理使用。

创建 JMS 主机

您可以使用以下方法创建其他 JMS 主机：

- 使用管理控制台。在相关配置下打开“Java 消息服务”组件，选择“JMS 主机”组件，然后单击“新建”。有关更多信息，请参见管理控制台联机帮助。
- 使用 `asadmin create-jms-host` 命令。有关详细信息，请参见 `create-jms-host(1)`。只要 JMS 主机配置发生更改，JMS 主机列表就会被更新。

连接池和故障转移

Application Server 支持 JMS 连接池和故障转移。Sun Java System Application Server 将自动实现 JMS 连接池。“地址列表行为”属性为 `random`（默认值）时，Application Server 将从 JMS 主机列表中随机选择其主代理。发生故障转移时，MQ 会将负载透明地转移到另一个代理，并维持 JMS 语义。如果 JMS 类型为 LOCAL，“地址列表行为”属性的默认值将为 `priority`。

要指定连接丢失时 Application Server 是否尝试重新连接到主代理，请选中“重新连接”复选框。如果已启用，则当主代理出现故障时，Application Server 将尝试重新连接到 JMS 主机列表中的另一个代理。

启用“重新连接”时，也同时指定以下属性：

- **地址列表行为**：是按 JMS 主机列表中的地址顺序 (`priority`) 还是按随机顺序 (`random`) 尝试连接。如果设置为 `Priority`，Java 消息服务将尝试连接到 JMS 主机列表中指定的第一个 MQ 代理，仅当第一个代理不可用时，才使用另一个代理。如果设置为

Random，Java 消息服务将从 JMS 主机列表中随机选择 MQ 代理。如果许多客户机同时尝试使用同一个连接工厂来进行连接，请使用此设置以防止它们全部尝试连接至同一个地址。

- **地址列表重复**：Java 消息服务建立（或重新建立）连接时，在 JMS 主机列表中重复的次数。值 -1 表示尝试次数没有限制。
- **重新连接尝试**：客户机运行时尝试连接（或重新连接）JMS 主机列表中每个地址的次数。到达这个值后，客户机运行时将尝试连接列表中的下一个地址。值 -1 表示重新连接尝试次数没有限制（客户机运行时将尝试连接至第一个地址，直到连接成功）。
- **重新连接时间间隔**：两次重新连接尝试之间的秒数。此设置适用于对 JMS 主机列表中每个地址的尝试，及对该列表中连续地址的尝试。如果该时间间隔太短，则代理将没有时间恢复。如果该时间间隔太长，则重新连接可能会指明这是不可接受的延迟。

您可以使用 JMS 连接工厂设置覆盖这些设置。有关详细信息，请参见《Sun Java System Application Server 9.1 管理指南》中的“JMS 连接工厂”。

负载均衡的消息流入

您可以在 `sun-ejb-jar.xml` 文件中使用 `activation-config-property` 元素为消息驱动 Bean 配置 `jmsra` 资源适配器的 `ActivationSpec` 属性。每当部署消息驱动 Bean (EndPointFactory) 后，连接器运行时引擎都会在资源适配器中查找这些属性并相应地对其进行配置。请参见《Sun Java System Application Server 9.1 Application Deployment Guide》中的“`activation-config-property`”。

Application Server 透明地允许将消息随机传送到具有相同 `ClientID` 的消息驱动 Bean。持久订阅者需要 `ClientID`。

对于没有配置 `ClientID` 的非持久订阅者，特定的消息驱动 Bean 中订阅同一主题的所有实例均认为是等效的。将消息驱动 Bean 部署到 Application Server 的多个实例时，只有其中的一个消息驱动 Bean 接收消息。如果多个不同的消息驱动 Bean 订阅同一主题，每个消息驱动 Bean 的一个实例将接收消息副本。

要使用同一队列支持多个使用者，请将物理目的地的 `maxNumActiveConsumers` 属性设置为较大值。如果设置了此属性，Sun Java System Message Queue 软件将允许最多该数目的消息驱动 Bean 使用同一队列中的消息。消息将随机传送到各个消息驱动 Bean。如果将 `maxNumActiveConsumers` 设置为 -1，将不限制使用者的数目。

要确保首选本地传送，请将 `addresslist-behavior` 设置为 `priority`。此设置指定首先选择 `AddressList` 中的第一个代理。此第一个代理是本地共存的 `Message Queue` 实例。如果此代理不可用，将按代理在 `AddressList` 中的列出顺序进行连接尝试。此设置是属于群集的 Application Server 实例的默认设置。

注 - 群集功能在开发者配置文件中不可用。有关配置文件的信息，请参见《Sun Java System Application Server 9.1 管理指南》中的“用法配置文件”。

JMS 服务高可用性

JMS 组件有两种级别的可用性：

- **服务可用性** - 在此级别上，JMS 服务的可用性很重要，但在消息仅片刻不可用时它不是很重要。只要连接故障转移到一个提供服务的新可用实例，JMS 组件就认为服务可用并正常运行。《Sun Java System Application Server 9.1 Developer's Guide》中的“Connection Failover”中介绍了此级别的可用性。
- **数据可用性** - 在此级别上，服务的可用性和持久性消息的可用性都是必需的。有且只有一次的传送和消息排序的 JMS 语义也在此级别上处理。

可以在包含 Java 消息服务 (Java Message Service, JMS) 的 Sun Java System Message Queue 群集中启用数据可用性。消息将持久保存到公共持久性存储，而且从群集中的所有其他代理实例可用，或者从高可用性数据库 (high-availability database, HADB) 可用（如果安装了该数据库并选择了企业配置文件）。有关配置文件的信息，请参见《Sun Java System Application Server 9.1 管理指南》中的“用法配置文件”。必须先对 Application Server 实例启用可用性，才能对相应的代理启用数据可用性。

注 - 单个应用程序和模块无法控制或覆盖 JMS 可用性。

要启用数据可用性，请在管理控制台中的相关配置下选择“可用性服务”组件。选中“可用性服务”框。要对 JMS 服务启用可用性，请选择“JMS 可用性”选项卡，然后选中“可用性服务”框。Application Server 群集中的所有实例都应该具有相同的实例可用性和 JMS 可用性设置，以确保一致的行为。有关详细信息，请参见《Sun Java System Application Server 9.1 高可用性管理指南》。

注 - 群集功能在开发者配置文件中不可用。有关配置文件的信息，请参见《Sun Java System Application Server 9.1 管理指南》中的“用法配置文件”。

结合使用 Application Server 与 MQ 群集

MQ 企业版支持多个互连代理实例（称为**代理群集**）。使用代理群集的情况下，客户机连接将分布在群集的所有代理中。群集可以提供水平可伸缩性并提高可用性。

本节介绍了如何配置 Application Server 以使用高可用性 Sun Java System Message Queue 群集。还介绍了如何启动和配置 Message Queue 群集。

有关 Application Server 和 MQ 部署拓扑的更多信息，请参见《Sun Java System Application Server 9.1 部署规划指南》中的“Planning Message Queue Broker Deployment”。

高可用性 MQ 群集

Sun Java System Message Queue 4.1 通过新的“高可用性”群集类型提供了高可用性消息传送服务。在此类型的 MQ 群集中，所有的代理实例都将共享对等关系并共享公共的持久性数据存储，从而提供“数据可用性”。实例将自动地能够检测某个实例是否出现故障，并通过接管选举动态执行故障代理的持久性消息的接管。这样，在 Application Server 中部署的应用程序组件可以利用这些可用性功能。

使用此群集类型，队列或持久主题订阅不会丢失任何已处理的持久性消息。客户机运行时环境所连接到的代理不可用时，很可能会丢失非持久性消息或非持久订阅者的持久性消息。

在本地模式下配置高可用性代理群集

1. 启动 HADB。
2. 创建一个 Application Server 域并启动它。要创建一个域并启动它，请分别使用 `asadmin` 命令 `create-domain` 和 `start-domain`。有关这些命令的更多信息，请参见 `create-domain(1)` 和 `start-domain(1)`。
3. 创建一个节点代理并启动它。要创建一个节点代理并启动它，请分别使用 `asadmin` 命令 `create-node-agent` 和 `start-node-agent`。有关这些命令的更多信息，请参见 `create-node-agent(1)` 和 `start-node-agent(1)`。
4. 创建群集。可以使用 `asadmin` 命令 `create-cluster` 或使用管理控制台创建群集。有关 `create-cluster` 命令的更多信息，请参见 `create-cluster(1)`。有关如何使用管理控制台创建群集的信息，请参见管理控制台联机帮助。
5. 在群集中创建实例。在创建实例时，请指定远程代理所用的 JMS 提供者端口号。如果您不指定一个端口号，将使用默认的 JMS 提供者端口号。可以使用管理控制台或 `asadmin` 命令 `create-instance` 创建实例。有关如何使用管理控制台创建实例的信息，请参见管理控制台联机帮助。有关 `create-instance` 命令的信息，请参见 `create-instance(1)`。
6. 启动群集。可以通过管理控制台或使用 `asadmin` 命令 `start-cluster` 启动群集。有关如何使用管理控制台启动群集的信息，请参阅管理控制台联机帮助。有关 `start-cluster` 命令的信息，请参见 `start-cluster(1)`。
7. 使用 `asadmin` 命令 `configure-ha-cluster` 配置 HA 群集。有关该命令的更多信息，请参见 `configure-ha-cluster(1)`。

在远程模式下配置高可用性代理群集

1. 启动 HADB。
2. 创建数据库表。
3. 如果要创建高可用性代理群集，请复制 HA 驱动程序。

```
cp $AS_HOME/hadb/4.4.3-6/lib/hadbjdbc4.jar $S1AS_HOME/imq/lib/ext
```

4. 创建一个域并启动它。为此，请使用 `asadmin` 命令 `create-domain` 和 `start-domain`。有关这些命令的更多信息，请参见 `create-domain(1)` 和 `start-domain(1)`。
5. 创建一个节点代理并启动它。为此，请使用 `asadmin` 命令 `create-node-agent` 和 `start-node-agent`。有关这些命令的更多信息，请参见 `create-node-agent(1)` 和 `start-node-agent(1)`。
6. 创建群集。可以使用 `asadmin` 命令 `create-cluster` 或使用管理控制台创建群集。有关更多信息，请参见 `create-cluster(1)`。有关如何使用管理控制台创建群集的信息，请参见管理控制台联机帮助。
7. 在群集中创建实例。在创建实例时，请指定远程代理所用的 JMS 提供者端口号。如果您不指定一个端口号，将使用默认的 JMS 提供者端口号。
8. 删除默认的 JMS 主机，并创建实例可以连接到的 JMS 主机。务必将每个代理作为单独的 JMS 主机进行添加。有关 JMS 主机的更多信息，请参见第 201 页中的“JMS 主机列表”。
9. 将 JMS 类型设置为 Remote。为此，可以使用 `asadmin` 命令 `set` 或管理控制台中的“JMS 服务”页面。
10. 如果要配置高可用性代理，请将 JMS 可用性设置为 `true`。为此，可以使用 `asadmin` 命令 `set` 或管理控制台中的“JMS 可用性”页面。
11. 启动代理实例。
12. 启动群集。有关更多信息，请参见 `start-cluster(1)`。

为非 HA 群集自动建立群集

直到现在，管理员必须单独建立“非高可用性”MQ 群集（具有主代理的 MQ 群集），如本节后面的过程中所述。在此发行版中，除了设置 MQ 群集的手动过程（类型为 REMOTE）外，Application Server 还提供了“自动建立群集功能”，这意味着在用户创建 Application Server 群集时将自动创建共存的非 HA 群集（类型为 LOCAL）。这将是创建 MQ 群集的默认模式。例如，在管理员创建具有三个 Application Server 实例的 Application Server 群集时，会将每个 Application Server 实例配置为与共存代理一起工作，因此将使三个 MQ 代理实例透明地形成一个 MQ 群集。第一个 Application Server 实例的 MQ 代理将被设置为主代理。但是，自动建立群集也有缺点。如果管理员将一个实例添加到群集，自动创建的 MQ 代理实例将无法参与群集。如果将一个实例从群集中删除，也会出现此行为。

▼ 使用 Application Server 群集启用 MQ 群集

开始之前 如果群集类型为 REMOTE，请执行以下步骤。如果群集类型为 LOCAL，则步骤 1 到 4 不适用。

1 如果尚不存在 Application Server 群集，请先创建一个。

有关创建群集的信息，请参见第 151 页中的“创建群集”。

2 创建 MQ 代理群集。

首先，删除涉及由域管理服务器启动的代理的默认 JMS 主机，然后创建三个将位于 MQ 代理群集中的外部代理（JMS 主机）。

使用管理控制台或 `asadmin` 命令行实用程序创建 JMS 主机。

要使用 `asadmin`，请参见以下命令示例：

```
asadmin delete-jms-host --target cluster1 default_JMS_host
asadmin create-jms-host --target cluster1
    --mqhost myhost1 --mqport 6769
    --mquser admin --mqpassword admin broker1
asadmin create-jms-host --target cluster1
    --mqhost myhost2 --mqport 6770
    --mquser admin --mqpassword admin broker2
asadmin create-jms-host --target cluster1
    --mqhost myhost3 --mqport 6771
    --mquser admin --mqpassword admin broker3
```

要使用管理控制台创建主机，请执行以下操作：

a. 导航到“JMS 主机”节点（“配置”>“*config-name*”>“Java 消息服务”>“JMS 主机”）。

b. 删除默认代理 (`default_JMS_host`)。

选择其旁边的复选框，然后单击“删除”。

c. 单击“新建”以创建每一台 JMS 主机并输入其属性值。

填写主机名称、DNS 名称或 IP 地址、端口号、管理用户名和密码。

3 启动 MQ 主代理和其他 MQ 代理。

除了 JMS 主机上启动的三个外部代理外，请在任一计算机上启动一个主代理。此主代理不需要是代理群集的一部分。例如：

```
/usr/bin/imqbrokerd -tty -name brokerm -port 6772
    -cluster myhost1:6769,myhost2:6770,myhost2:6772,myhost3:6771
    -D"imq.cluster.masterbroker=myhost2:6772"
```

4 启动群集中的 Application Server 实例。

5 在群集上创建 JMS 资源：

a. 创建 JMS 物理目的地。

例如，使用 `asadmin`：

```
asadmin create-jmsdest --desttype queue --target cluster1 MyQueue
asadmin create-jmsdest --desttype queue --target cluster1 MyQueue1
```

要使用管理控制台，请执行以下操作：

- i. 导航到“JMS 主机”页面（“配置”>“*config-name*”>“Java 消息服务”>“物理目的地”）。
- ii. 单击“新建”以创建每个 JMS 物理目的地。
- iii. 对于每个目标，输入其名称和类型（队列）。

b. 创建 JMS 连接工厂。

例如，使用 `asadmin`：

```
asadmin create-jms-resource --target cluster1
--restype javax.jms.QueueConnectionFactory jms/MyQcf
asadmin create-jms-resource --target cluster1
--restype javax.jms.QueueConnectionFactory jms/MyQcf1
```

要使用管理控制台，请执行以下操作：

- i. 导航到“JMS 连接工厂”页面（“资源”>“JMS 资源”>“连接工厂”）。
- ii. 要创建每个连接工厂，请单击“新建”。
将打开“创建 JMS 连接工厂”页面。
- iii. 对于每个连接工厂，请输入 JNDI 名称（例如 `jms/MyQcf`）和类型（`javax.jms.QueueConnectionFactory`）。
- iv. 从页面底部的可用目标列表中选择群集，然后单击“添加”。
- v. 单击“确定”以创建连接工厂。

c. 创建 JMS 目标资源。

例如，使用 `asadmin`：

```
asadmin create-jms-resource --target cluster1
--restype javax.jms.Queue
--property imqDestinationName=MyQueue jms/MyQueue
asadmin create-jms-resource --target cluster1
--restype javax.jms.Queue
--property imqDestinationName=MyQueue1 jms/MyQueue1
```


要使用管理控制台，请执行以下操作：

- i. 导航到“JMS 目标资源”页面（“资源”>“JMS 资源”>“连接工厂”）。
- ii. 要创建每个目标资源，请单击“新建”。
将打开“创建 JMS 目标资源”页面。
- iii. 对于每个目标资源，请输入 JNDI 名称（例如 `.jms/MyQueue`）和类型（`javax.jms.Queue`）。
- iv. 从页面底部的可用目标列表中选择群集，然后单击“添加”。
- v. 单击“确定”创建目标资源。

- 6 使用 `-retrieve` 选项为应用程序客户机部署应用程序。例如：

```
asadmin deploy --target cluster1
--retrieve /opt/work/MQapp/mdb-simple3.ear
```

- 7 访问应用程序并进行测试以确保其按预期运行。

- 8 如果要将 Application Server 返回到其默认 JMS 配置，请删除您创建的 JMS 主机并重新创建默认值。例如：

```
asadmin delete-jms-host --target cluster1 broker1
asadmin delete-jms-host --target cluster1 broker2
asadmin delete-jms-host --target cluster1 broker3
asadmin create-jms-host --target cluster1
--mqhost myhost1 --mqport 7676
--mquser admin --mqpassword admin
default_JMS_host
```

您还可以使用管理控制台执行等效操作。

故障排除 如果出现问题，请考虑以下操作：

- 查看位于 `as-install-dir/nodeagents/node-agent-name/instance-name/logs/server.log` 的 Application Server 日志文件。如果在日志文件中看到 MQ 代理没有响应消息，则停止该代理，然后重新启动它。
- 查看位于 `as-install-dir/nodeagents/node-agent-name/instance-name/imq/imq-instance-name/log/log.txt` 的代理日志。
- JMS 类型为 Remote 时，务必始终先启动 MQ 代理，然后再启动 Application Server 实例。
- 所有 MQ 代理都关闭时，关闭或打开 Application Server 需要花费 30 分钟，并具有 Java 消息服务中的默认值。调整 Java 消息服务值以获取此超时的可接受值。例如：

```
asadmin set --user admin --password administrator  
cluster1.jms-service.reconnect-interval-in-seconds=5
```

RMI-IIOP 负载均衡和故障转移

本章介绍如何通过 RMI-IIOP 使用 Sun Java System Application Server 的高可用性功能获得远程 EJB 引用和 JNDI 对象。

- 第 211 页中的“概述”
- 第 213 页中的“设置 RMI-IIOP 负载均衡和故障转移”

概述

利用 RMI-IIOP 负载均衡，IIOP 客户机请求可分布到不同的服务器实例或名称服务器。目标是将负载平均分布在群集中，从而提供可伸缩性。与 EJB 群集和可用性结合的 IIOP 负载均衡还可提供 EJB 故障转移。

客户机执行某个对象的 JNDI 查找时，命名服务会创建一个与特定服务器实例关联的 `InitialContext` (IC) 对象。从此时起，使用该 IC 对象进行的所有查找请求都会发送给相同的服务器实例。使用该 `InitialContext` 查找的所有 `EJBHome` 对象都托管在相同的目标服务器上。此后获得的所有 `Bean` 引用也创建在相同的目标主机上。这就有效地提供了负载均衡，原因是所有客户机都在创建 `InitialContext` 对象时随机使用动态目标服务器的列表。如果目标服务器实例发生故障，查找或 EJB 方法调用会将故障转移到另一个服务器实例。

IIOP 负载均衡和故障转移将透明地发生。在应用程序部署过程中无需特殊的步骤。`Application Server` 的 IIOP 负载均衡和故障转移支持动态重新配置的群集。如果在其上部署应用程序客户机的 `Application Server` 实例参与群集，则 `Application Server` 将自动查找群集中当前处于活动状态的所有 IIOP 端点。因此，如果将新实例添加到群集或将其从群集中删除，则无需手动更新端点列表。但是，客户机至少应指定两个端点以用于引导目的，防止其中一个端点出现故障。

要求

Sun Java System Application Server 通过 RMI-IIOP 提供远程 EJB 引用和 NameService 对象的高可用性，前提是满足以下所有条件：

- 您的部署包含一个至少有两个应用程序服务器实例的群集。
- Java EE 应用程序已被部署到所有参与负载均衡的应用服务器实例和群集。
- 已启用 RMI-IIOP 客户机应用程序以用于负载均衡。

Application Server 支持对在应用程序客户机容器 (Application Client Container, ACC) 中执行的 Java 应用程序进行负载均衡。请参见第 213 页中的“设置 RMI-IIOP 负载均衡和故障转移”。

注 - Application Server 不支持通过安全套接字层 (Secure Socket Layer, SSL) 进行 RMI-IIOP 负载均衡和故障转移。

算法

Application Server 使用随机算法和 round-robin 算法进行 RMI-IIOP 负载均衡和故障转移。

在 RMI-IIOP 客户机首次创建新的 InitialContext 对象时，可用的 Application Server IIOP 端点的列表对于该客户机是随机的。对于该 InitialContext 对象，负载均衡器会将查找请求和其他 InitialContext 操作定向至随机列表中的第一个端点。如果第一个端点不可用，则使用列表中的第二个端点，依此类推。

随后每次客户机创建新的 InitialContext 对象时，将轮转端点列表，从而将不同的 IIOP 端点用于 InitialContext 操作。

当您从通过 InitialContext 对象获取的引用中获取或创建 Bean 时，将在服务于指定给 InitialContext 对象的 IIOP 端点的 Application Server 实例上创建这些 Bean。对这些 Bean 的引用包括群集中的所有 Application Server 实例的 IIOP 端点地址。

主端点是与用于查找或创建 Bean 的 InitialContext 端点对应的 Bean 端点。群集中的其他 IIOP 端点将被指定为**备用端点**。如果 Bean 的主端点变为不可用，则该 Bean 上的其他请求将故障转移到备用端点之一。

可以配置 RMI-IIOP 负载均衡和故障转移以用于在 ACC 中运行的应用程序。

设置 RMI-IIOP 负载均衡和故障转移

可以为在应用程序客户机容器 (application client container, ACC) 中运行的应用程序设置 RMI-IIOP 负载均衡和故障转移。还支持加权的 round-robin 负载均衡。

▼ 为应用程序客户机容器设置 RMI-IIOP 负载均衡

此过程概要介绍了将应用程序客户机容器 (ACC) 与 RMI-IIOP 负载均衡和故障转移结合使用所需要的步骤。有关 ACC 的其他信息，请参见《Sun Java System Application Server 9.1 Developer's Guide》中的“Developing Clients Using the ACC”。

- 1 转到 `install_dir/bin` 目录。
- 2 运行 `package-appclient`。
此实用程序生成一个 `appclient.jar` 文件。有关 `package-appclient` 的更多信息，请参见 `package-appclient(1M)`。
- 3 将 `appclient.jar` 文件复制到需要作为客户机的计算机上，然后将其解压缩。
- 4 编辑 `asenv.conf` 或 `asenv.bat` 路径变量，以引用该计算机上的正确目录值。
该文件位于 `appclient-install_dir/config/` 中。
有关待更新路径变量的列表，请参见 `package-appclient(1M)`。
- 5 如果需要，请使 `appclient` 脚本成为可执行脚本。
例如，在 UNIX 上使用 `chmod 700`。
- 6 至少查找群集中两个实例的 IIOP 侦听器端口号。
在步骤 7 中将 IIOP 侦听器指定为端点。
对于每个实例，按如下所述获取 IIOP 侦听器端口：
 - a. 在管理控制台的树组件中，展开“群集”节点。
 - b. 展开群集。
 - c. 在群集中选择一个实例。
 - d. 在右窗格中单击“属性”选项卡。
 - e. 注意用于实例的 IIOP 侦听器端口。
- 7 在 `sun-acc.xml` 文件中至少添加两个 `target-server` 元素。

注 - 群集功能在开发者配置文件中不可用。有关配置文件的信息，请参见《Sun Java System Application Server 9.1 管理指南》中的“用法配置文件”。

使用在步骤 6 中获取的端点。

如果在其上部署应用程序客户机的 Application Server 实例参与群集，则 ACC 将自动查找群集中当前处于活动状态的所有 IIOP 端点。但是，客户机至少应指定两个端点以达到引导目的，以防其中一个端点出现故障。

`target-server` 元素指定用于负载均衡的一个或多个 IIOP 端点。`address` 属性是 IPv4 地址或主机名，`port` 属性指定端口号。请参见《Sun Java System Application Server 9.1 Application Deployment Guide》中的“client-container”。

作为使用 `target-server` 元素的替代方法，可以使用 `endpoints` 属性，如下所示：

```
jvmarg value = "-Dcom.sun.appserv.iiop.endpoints=host1:port1,host2:port2,..."
```

8 如果需要加权 round-robin 负载均衡，请执行以下步骤：

a. 设置每个服务器实例的负载均衡权重。

```
asadmin set instance-name.lb-weight=weight
```

b. 在 `sun-acc.xml` 中，将 ACC 的 `com.sun.appserv.iiop.loadbalancingpolicy` 属性设置为 `ic-based-weighted`。

```
...
<client-container send-password="true">
  <property name="com.sun.appserv.iiop.loadbalancingpolicy" value="ic-based-weighted"/>
...

```

9 部署客户机应用程序，同时使用 `--retrieve` 选项获取客户机 jar 文件。

将该客户机 jar 文件保留在客户机上。

例如：

```
asadmin deploy --user admin --passwordfile pw.txt --retrieve /my_dir myapp
```

10 按下述命令运行应用程序客户机：

```
applclient -client clientjar -name appname
```

示例 11-1 为 RMI-IIOP 加权 Round-Robin 负载均衡设置负载均衡权重

在此示例中，包含三个实例的群集中负载均衡权重如下表所示进行设置。

实例名称	负载均衡权重
i1	100
i2	200
i3	300

设置这些负载均衡权重的命令序列如下：

```
asadmin set i1.lb-weight=100
asadmin set i2.lb-weight=200
asadmin set i3.lb-weight=300
```

接下来的操作 要测试故障转移，请停止群集中的一个实例，然后查看应用程序是否正常运行。在客户机应用程序中还可以有断点（或休眠）。

要测试负载均衡，请使用多个客户机并查看负载在所有端点上的分布情况。

索引

A

active-healthcheck-enabled, 130
AddressList, 默认 JMS 主机, 202
Apache Web Server
 Application Server 安装程序所做的修改, 112
 安全文件, 113
 负载均衡器插件进行的修改, 109-111
 具有负载均衡器, 107
asadmin create-jms-host 命令, 202
asadmin get 命令, 200
asadmin set 命令, 200
authPassthroughEnabled, 135

C

cacheDatabaseMetaData 属性, 73
checkpoint-at-end-of-method 元素, 197
ConnectionTrace 属性, 69
CoreFile 属性, 69
create-http-lb-config 命令, 126
create-http-lb-ref 命令, 128
create-node-agent 命令, 177

D

DatabaseName 属性, 69
databuf 选项, 92
DataBufferPoolSize 属性, 69
datadevices 选项, 63
DataDeviceSize 属性, 69, 83

dbpassword 选项, 59
dbpasswordfile 选项, 59
default-config 配置, 160
delete-http-lb-ref 命令, 128
delete-node-agent 命令, 180
DevicePath 属性, 69, 86
devicepath 选项, 64
devicesize 选项, 64
disable-http-lb-application 命令, 133
disable-http-lb-server 命令, 133

E

EagerSessionThreshold 属性, 69
EagerSessionTimeout 属性, 69
EJB 容器, 可用性, 195
eliminateRedundantEndTransaction 属性, 73
enable-http-lb-application 命令, 129
enable-http-lb-server 命令, 128
EventBufferSize 属性, 69
export-http-lb-config 命令, 131

F

fast 选项, 81

H

HADB
 nodes, 89

HADB (续)

- 端口指定, 67
- 环境变量, 60
- 获取 JDBC URL, 72
- 获取设备信息, 90
- 获取状态, 88-90
- 获取资源信息, 92-94
- 计算机维护, 94
- 监视, 88-94
- 扩展节点, 83-84
- 历史文件, 96
- 连接池设置, 73
- 连接池属性, 73-74
- 列出数据库, 80
- 配置, 61-74
- 启动节点, 76
- 启动数据库, 78
- 清除数据库, 81
- 删除数据库, 81
- 设置属性, 65, 67
- 数据库名称, 63
- 数据损坏, 82
- 双网络, 36-37
- 添加计算机, 84
- 添加节点, 84
- 停止节点, 77
- 停止数据库, 79
- 异构设备路径, 66
- 用于 JMS 数据, 204
- 重新分段, 86
- 重新启动节点, 78
- 重新启动数据库, 80

HADB 管理代理, 启动, 42-43, 49-57

HADB 配置

- 节点监控进程, 40-41
- 时间同步, 39
- 网络配置, 34-37

HADB 设置, 33

hadbm addnodes 命令, 84

hadbm clear 命令, 81

hadbm clearhistory 命令, 96

hadbm create 命令, 62

hadbm delete 命令, 81

hadbm deviceinfo 命令, 90

hadbm get 命令, 67

hadbm list 命令, 80

hadbm refragment 命令, 86

hadbm resourceinfo 命令, 92-94

hadbm restart 命令, 80

hadbm restartnode 命令, 78

hadbm start 命令, 78

hadbm startnode 命令, 76

hadbm status 命令, 88-90

hadbm stop 命令, 79

hadbm stopnode 命令, 77

hadbm 命令, 57-61

HistoryPath 属性, 70

historypath 选项, 64

hosts 选项, 64, 86

HTTP

- HTTPS 路由选择, 134
- 会话故障转移, 134-135

HTTP_LISTENER_PORT 属性, 163

HTTP_SSL_LISTENER_PORT 属性, 163

HTTP 会话, 22

- 分布式, 185-186

HTTPS

- 会话故障转移, 134-135
- 路由选择, 134

HTTPS 路由选择, 134-135

I

IIO_P_LISTENER_PORT 属性, 163

IIO_P_SSL_MUTUALAUTH_PORT 属性, 163

InternalLogbufferSize 属性, 70

IOP_SSL_LISTENER_PORT 属性, 163

J

JdbcUrl 属性, 70

JMS

- 创建主机, 202
- 高可用性, 204
- 连接池, 22, 202
- 连接故障转移, 202
- 配置, 200

JMS 主机列表, 连接, 201
JMX_SYSTEM_CONNECTOR_PORT 属性, 163
JMX 侦听器, 节点代理, 182
JNDI 名称设置, 74

L

loadbalancer.xml 文件, 131
locks 选项, 92
logbuf 选项, 92
LogbufferSize 属性, 70

M

maxStatement 属性, 73
MaxTables 属性, 70
Microsoft Internet Information Services (IIS), 修改以
实现负载均衡, 115

N

nilogbuf 选项, 92
no-refragment 选项, 85
no-repair 选项, 77
nodes 选项, 89
number-healthcheck-retries, 131
NumberOfDatadevices 属性, 70
NumberOfLocks 属性, 70
NumberOfSessions 属性, 70

P

password 属性, 73
Portbase 属性, 70
portbase 选项, 65

R

RelalgdeviceSize 属性, 70
rewrite-location 属性, 137

round robin 负载均衡, 粘性, 121

S

saveto 选项, 96
serverList 属性, 73
SessionTimeout 属性, 70
set 选项, 65, 66
spares 选项, 65, 81, 85
SQLTraceMode 属性, 70
start-node-agent 命令, 179
startlevel 选项, 77, 78
StartRepairDelay 属性, 70
StatInterval 属性, 71
stop-node-agent 命令, 179
sun-ejb-jar.xml 文件, 197
Sun Java System Message Queue, 连接器用于, 199
Sun Java System Web Server, 负载均衡器进行的修
改, 100
sun-passthrough.properties 文件, 和日志级别, 148
sun-passthrough 属性, 117
SyslogFacility 属性, 71
SysLogging 属性, 71
SysLogLevel 属性, 71
SyslogPrefix 属性, 71

T

TakeoverTime 属性, 71

U

username 属性, 73

W

Web 服务器, 多个实例和负载均衡, 139
Web 容器, 可用性, 190
Web 应用程序, 可分布, 188

保

保证隔离层设置, 73

备

备用端点, RMI-IIOP 故障转移, 212

表

表名称设置, 73

部

部署, 设置可用性, 188

持

持久性, 会话, 22

持久性存储, 对于有状态会话 Bean 状态, 194

池

池名称设置, 74

单

单点登录, 会话持久性, 193-194

动

动态重新配置, 负载均衡器, 132

端

端点, RMI-IIOP 故障转移, 212

端口号, 和配置, 161

分

分布式 HTTP 会话, 185-186

服

服务器, 群集, 149

服务器实例

启用以进行负载均衡, 128

停止, 133

负

负载均衡

Apache Web Server, 107

HTTP, 关于, 121

HTTP 算法, 121

Microsoft IIS, 115

RMI-IIOP 要求, 212

Sun Java System Web Server, 100

创建负载均衡器配置, 126

创建引用, 128

导出配置文件, 131

动态重新配置, 132

多个 Web 服务器实例, 139

更改配置, 132

会话故障转移, 134-135

幂等 URL, 138

启用服务器实例, 128

启用应用程序, 129

日志消息, 145

设置, 123

停止服务器实例或群集, 133

停止应用程序, 133

运行状况检查器, 129

粘性 round robin, 121

故

故障转移

JMS 连接, 202

RMI-IIOP 要求, 212

对于 Web 模块会话, 185-186

故障转移 (续)

- 关于 HTTP, 121
- 有状态会话 Bean 状态, 194

管

管理控制台

- 用于创建 JMS 主机, 202
- 用于配置 JMS 服务, 200

滚

- 滚动升级, 139

会

会话

- HTTP, 22
- 持久性, 22

会话持久性

- 单点登录, 193-194
- 对于 Web 模块, 185-186
- 对于有状态会话 Bean, 194, 196

会话存储

- 对于 HTTP 会话, 191
- 对于有状态会话 Bean, 196

- 会话故障转移, HTTP 和 HTTPS, 134-135

基

- 基于 Cookie 的会话粘性, 122

检

- 检查点操作, 194
- 选择方法, 194, 197

节

节点代理

- JMX 侦听器, 182
- 安装, 171
- 部署, 169
- 创建, 177
- 关于, 167
- 启动, 179
- 日志, 175
- 删除, 180, 181
- 停止, 179
- 验证区域, 182
- 与域管理服务器同步, 171
- 占位符, 176
- 节点监控进程和高可用性, 40-41

可

- 可分布的 Web 应用程序, 188

可用性

- EJB 容器级别, 197
- 对于 Web 模块, 185-186
- 对于有状态会话 Bean, 194
- 级别, 188
- 启用和禁用, 188

连

连接池

- HADB 的设置, 73
- HADB 的属性, 73-74

路

- 路由 Cookie, 128

幂

- 幂等 URL, 138

名

名称设置, 73

命

命名配置

- default-config, 160
- 端口号和, 161
- 共享, 160
- 关于, 159
- 默认名称, 160

目

目标, 负载均衡器配置, 128

配

配置, 请参见命名配置

区

区域, 节点代理验证, 182

全

全局事务支持设置, 73

群

- 群集, 149
 - 独立, 26
 - 共享, 26
 - 停止, 133
- 群集服务器实例, 配置, 160

日

日志记录

- 查看节点代理日志, 175
- 负载均衡器, 145

时

时间同步, 39

事

事务

- 会话持久性, 194, 197
- 事务隔离设置, 73

数

- 数据库供应商设置, 73
- 数据源类名称设置, 73

算

算法

- HTTP 负载均衡, 121
- RMI-IIOP 故障转移, 212

所

所有连接均失败设置, 73

停

停止

- 服务器实例或群集, 133
- 应用程序, 133

网

网络配置要求, 34-37

稳

稳定的池大小设置, 73

响

响应超时, 127

信

信号量, 37

需

需要连接验证设置, 73

验

验证方法设置, 73

验证区域, 节点代理, 182

已

已启用的数据源设置, 74

异

异常的服务器实例, 129

应

应用程序

启用以进行负载均衡, 129

升级而不使可用性受到损失, 139

应用程序 (续)

停止, 133

有

有状态会话 Bean, 194

会话持久性, 194, 196

有状态会话 Bean 状态的检查点操作, 188

域

域管理服务器

服务器实例同步, 172

节点代理同步, 171

运

运行状况检查器, 129

粘

粘性 round robin 负载均衡, 121

中

中心系统信息库, 节点代理同步, 171

重

重新装入时间间隔, 128

主

主端点, RMI-IIOP 故障转移, 212

