Sun Java System Directory Server Enterprise Edition 6.0 管理指南

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

文件号码 820–0293
2007 年 3 月

070429@17039

目录

图

表

示例

前言

管理指南提供了从命令行配置目录服务器和目录代理服务器功能的过程信息。联机帮助中提供了有关使用基于 Web 的界面（目录服务控制中心）配置这些功能的说明。

目标读者

本管理指南适用于目录服务器和目录代理服务器软件的管理员。

阅读本书之前

本书未提供有关安装软件的信息。有关安装信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Installation Guide》。

如果要从较早版本的目录服务器或目录代理服务器进行迁移，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Migration Guide》，以获取有关迁移服务器的说明。如果您不熟悉此版本中的新功能，建议您阅读《Sun Java System Directory Server Enterprise Edition 6.0 Evaluation Guide》，以获取有关新功能的概述。

本书的结构

第 1 部分提供了有关管理目录服务器的过程信息。

第 2 部分提供了有关管理目录代理服务器的过程信息。

本指南中使用的示例

出于一致性考虑，本指南始终使用相同的示例数据。请根据您的系统需求使用相应的值替换这些值。

表 P–1 示例中使用的默认值

变量	示例中使用的值
后缀 (SUFFIX_DN)	dc=example,dc=com
实例路径 (INSTANCE_PATH)	目录服务器：/local/ds/ 目录代理服务器：/local/dps/
主机名 (HOST)	host1、host2、host3
端口 (PORT)	LDAP：超级用户的默认值：389。非超级用户的默认值：1389 SSL 默认值：超级用户的默认值：636。非超级用户的默认值：1636

Directory Server Enterprise Edition 文档集

本 Directory Server Enterprise Edition 文档集说明如何使用 Sun Java System Directory Server Enterprise Edition 来评估、设计、部署和管理目录服务。此外，它还说明如何为 Directory Server Enterprise Edition 开发客户端应用程序。可在 `http://docs.sun.com/coll/1224.1` 和 `http://docs.sun.com/coll/1606.1` 中找到 Directory Server Enterprise Edition 文档集。

有关 Directory Server Enterprise Edition 的介绍，请查看以下文档（按照文档列出的顺序）。

表 P–2 Directory Server Enterprise Edition 文档

文档标题	内容
《Sun Java System Directory Server Enterprise Edition 6.0 发行说明》	包含有关 Directory Server Enterprise Edition 的最新信息，其中包括已知问题。
《Sun Java System Directory Server Enterprise Edition 6.0 Documentation Center》	包含指向文档集中主要内容的链接。
《Sun Java System Directory Server Enterprise Edition 6.0 Evaluation Guide》	介绍此发行版的主要功能。说明这些功能如何起作用，以及它们在虚构部署（可以在单个系统中实施）环境中提供哪些功能。
《Sun Java System Directory Server Enterprise Edition 6.0 Deployment Planning Guide》	说明如何基于 Directory Server Enterprise Edition 规划和设计具有高可用性和高伸缩性的目录服务。提供部署规划和设计的基本概念和原理。讨论解决方案生命周期，并提供基于 Directory Server Enterprise Edition 规划解决方案时要使用的高级示例和策略。

表 P–2 Directory Server Enterprise Edition 文档 （续）

文档标题	内容
《Sun Java System Directory Server Enterprise Edition 6.0 Installation Guide》	说明如何安装 Directory Server Enterprise Edition 软件。说明如何选择要安装的组件、安装后如何配置这些组件，以及如何验证已配置的组件能否正常工作。 有关安装目录编辑器的说明，请转至 http://docs.sun.com/coll/DirEdit_05q1 在安装目录编辑器之前，请务必阅读《Sun Java System Directory Server Enterprise Edition 6.0 发行说明》中与目录编辑器相关的内容。
《Sun Java System Directory Server Enterprise Edition 6.0 Migration Guide》	提供用于从早期版本的目录服务器、目录代理服务器和 Identity Synchronization for Windows 升级组件的说明。
《Sun Java System Directory Server Enterprise Edition 6.0 管理指南》	提供用于管理 Directory Server Enterprise Edition 的命令行指令。 有关使用目录服务控制中心 (Directory Service Control Center, DSCC) 管理 Directory Server Enterprise Edition 的提示和说明，请参见 DSCC 中提供的联机帮助。 有关管理目录编辑器的说明，请转至 http://docs.sun.com/coll/DirEdit_05q1 有关安装和配置 Identity Synchronization for Windows 的说明，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Installation Guide》中的第 II 部分, “Installing Identity Synchronization for Windows”。
《Sun Java System Directory Server Enterprise Edition 6.0 Developer’s Guide》	介绍如何使用 Directory Server Enterprise Edition 中提供的 API 来开发服务器插件。
《Sun Java System Directory Server Enterprise Edition 6.0 Reference》	介绍 Directory Server Enterprise Edition 的技术及概念基础。描述其组件、体系结构、过程和功能。此外还提供了开发者 API 引用。
《Sun Java System Directory Server Enterprise Edition 6.0 Man Page Reference》	描述 Directory Server Enterprise Edition 中可用的命令行工具 、模式对象及其他公共接口。本文档的各个部分都可以作为联机手册页进行安装。
《Sun Java System Identity Synchronization for Windows 6.0 Deployment Planning Guide》	提供用于规划和部署 Identity Synchronization for Windows 的常规指南和最佳实践。

相关知识

SLAMD 分散负载生成引擎 (SLAMD) 是一种 Java™ 应用程序，用于进行压力测试以及分析网络应用程序的性能。该程序最初由 Sun Microsystems, Inc. 开发，用于对 LDAP 目录服务器的性能进行基准测试和分析。SLAMD 以开放源代码应用程序的形式提供，受 Sun 公共许可证（OSI 批准的开放源代码许可证）的限制。要获取有关 SLAMD 的信息，请转至 http://www.slamd.com/。SLAMD 还会以 java.net 项目的形式提供。请参见 https://slamd.dev.java.net/。

Java 命名和目录接口 (Java Naming and Directory Interface, JNDI) 技术允许在 Java 应用程序中使用 LDAP 和 DSML v2 访问目录服务器。有关 JNDI 的信息，请参见

http://java.sun.com/products/jndi/。 JNDI 教程包含有关如何使用 JNDI 的详细描述和示例。此教程位于 http://java.sun.com/products/jndi/tutorial/ 。

可以将 Directory Server Enterprise Edition 授权为独立产品、Sun Java Enterprise System 的组件、Sun 产品套件的一部分（如 Sun Java Identity Management Suite）或 Sun 提供的其他软件产品的附加软件包。Java Enterprise System 是软件基础结构，它支持网络或 Internet 环境中的分布式企业应用程序。如果 Directory Server Enterprise Edition 作为 Java Enterprise System 的组件获得许可，您应该熟悉 http://docs.sun.com/coll/1286.2 和 http://docs.sun.com/coll/1382.2 中的系统文档。

Identity Synchronization for Windows 使用具有受限许可证的 Message Queue。可在 http://docs.sun.com/coll/1307.2 和 http://docs.sun.com/coll/1391.2 中找到 Message Queue 文档。

Identity Synchronization for Windows 与 Microsoft Windows 密码策略一起使用。

- 有关 Windows 2003 密码策略的信息，请参阅联机的 Microsoft 文档。
- 有关更改密码以及 Windows 2003 组策略的信息，请参阅联机的 Microsoft 文档。
- 有关 Microsoft Certificate Services Enterprise Root 证书颁发机构的信息，请参阅联机的 Microsoft 支持文档。
- 有关在 Microsoft 系统上通过 SSL 配置 LDAP 的信息，请参阅联机的 Microsoft 支持文档。

可再分发的文件

Directory Server Enterprise Edition 不提供任何可再分发的文件。

默认路径和命令位置

本部分介绍了文档中使用的默认路径，并提供了不同操作系统和部署类型中的命令位置。

默认路径

本部分中的表格描述了此文档中使用的默认路径。有关所安装的文件的完整描述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的第 15 章 “Directory Server File Reference”、《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的第 26 章 “Directory Proxy Server File Reference”或《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的附录 A “Directory Server Resource Kit File Reference”。

表 P–3 默认路径

占位符	描述	默认值
install-path	表示 Directory Server Enterprise Edition 软件的基本安装目录。 软件将安装在此基本 *install-path* 下的目录中。例如，目录服务器软件将安装在 *install-path*`/ds6/` 中。	使用 `dsee_deploy`(1M) 从 zip 分发包进行安装时，默认的 *install-path* 为当前目录。可以使用 `dsee_deploy` 命令的 `-i` 选项来设置 *install-path*。 从本地软件包版本进行安装时（例如使用 Java Enterprise System 安装程序），默认的 *install-path* 为以下位置之一： ■ Solaris 系统 - `/opt/SUNWdsee/` ■ HP-UX 系统 - `/opt/sun/` ■ Red Hat 系统 - `/opt/sun/` ■ Windows 系统 - `C:\Program Files\Sun\JavaES5\DSEE`
instance-path	表示目录服务器或目录代理服务器实例的完整路径。 对于目录服务器，本文档使用 `/local/ds/`，对于目录代理服务器，则使用 `/local/dps/`。	不存在默认路径。但是，实例路径必须始终位于**本地**文件系统中。 建议使用以下目录： 在 Solaris 系统上使用 `/var` 在 Sun Cluster 上使用 `/global`
serverroot	表示 Identity Synchronization for Windows 安装位置的父目录。	取决于您的安装。请注意，目录服务器中不再有 *serverroot* 的概念。
`isw-`*hostname*	表示 Identity Synchronization for Windows 实例目录。	取决于您的安装。
/path/to/`cert8.db`	表示适用于 Identity Synchronization for Windows 的客户端证书数据库的默认路径和文件名。	*current-working-dir*`/cert8.db`
serverroot`/isw-`*hostname*`/logs/`	表示系统管理器、每个连接器以及中心日志程序的 Identity Synchronization for Windows 本地日志所在的默认路径。	取决于您的安装。
serverroot`/isw-`*hostname*`/logs/central/`	表示 Identity Synchronization for Windows 中心日志所在的默认路径。	取决于您的安装。

命令位置

本部分中的表格提供了 Directory Server Enterprise Edition 文档中所用命令的位置。要了解有关每条命令的详细信息，请参见相关手册页。

表 P–4 命令位置

命令	Java ES，本地软件包版本	zip 分发包
cacaoadm	Solaris - /usr/sbin/cacaoadm	Solaris - *install-path*/dsee6/cacao_2.0/usr/lib/cacao/bin/cacaoadm
	Red Hat、HP-UX - /opt/sun/cacao/bin/cacaoadm	Red Hat、HP-UX - *install-path*/dsee6/cacao_2.0/cacao/bin/cacaoadm
	Windows - *install-path*\share\cacao_2.0\bin\cacaoadm.bat	Windows - *install-path*\dsee6\cacao_2.0\bin\cacaoadm.bat
certutil	Solaris - /usr/sfw/bin/certutil	*install-path*/dsee6/bin/certutil
	Red Hat、HP-UX - /opt/sun/private/bin/certutil	
dpadm(1M)	*install-path*/dps6/bin/dpadm	*install-path*/dps6/bin/dpadm
dpconf(1M)	*install-path*/dps6/bin/dpconf	*install-path*/dps6/bin/dpconf
dsadm(1M)	*install-path*/ds6/bin/dsadm	*install-path*/ds6/bin/dsadm
dsccmon(1M)	*install-path*/dscc6/bin/dsccmon	*install-path*/dscc6/bin/dsccmon
dsccreg(1M)	*install-path*/dscc6/bin/dsccreg	*install-path*/dscc6/bin/dsccreg
dsccsetup(1M)	*install-path*/dscc6/bin/dsccsetup	*install-path*/dscc6/bin/dsccsetup
dsconf(1M)	*install-path*/ds6/bin/dsconf	*install-path*/ds6/bin/dsconf
dsee_deploy(1M)	未提供	*install-path*/dsee6/bin/dsee_deploy
dsmig(1M)	*install-path*/ds6/bin/dsmig	*install-path*/ds6/bin/dsmig
entrycmp(1)	*install-path*/ds6/bin/entrycmp	*install-path*/ds6/bin/entrycmp
fildif(1)	*install-path*/ds6/bin/fildif	*install-path*/ds6/bin/fildif
idsktune(1M)	*install-path*/dsrk6/bin/idsktune	*install-path*/dsrk6/bin/idsktune
insync(1)	*install-path*/ds6/bin/insync	*install-path*/ds6/bin/insync
ns-accountstatus(1M)	*install-path*/ds6/bin/ns-accountstatus	*install-path*/ds6/bin/ns-accountstatus
ns-activate(1M)	*install-path*/ds6/bin/ns-activate	*install-path*/ds6/bin/ns-activate
ns-inactivate(1M)	*install-path*/ds6/bin/ns-inactivate	*install-path*/ds6/bin/ns-inactivate

表 P–4 命令位置 （续）

命令	Java ES，本地软件包版本	zip 分发包
repldisc(1)	*install-path*/ds6/bin/repldisc	*install-path*/ds6/bin/repldisc
schema_push(1M)	*install-path*/ds6/bin/schema_push	*install-path*/ds6/bin/schema_push
smcwebserver	Solaris、Linux、HP-UX - /usr/sbin/smcwebserver	此命令仅适用于目录服务控制中心，zip 分发包中未提供。
	Windows - *install-path*\share\webconsole\bin\smcwebserver	
wcadmin	Solaris、Linux、HP-UX - /usr/sbin/wcadmin	此命令仅适用于目录服务控制中心，zip 分发包中未提供。
	Windows - *install-path*\share\webconsole\bin\wcadmin	

印刷约定

下表描述了本书中使用的印刷更改。

表 P–5 印刷约定

字体或符号	含义	示例
AaBbCc123	命令、文件和目录的名称；计算机屏幕输出	编辑 .login 文件。 使用 ls -a 列出所有文件。 machine_name% you have mail.
AaBbCc123	用户键入的内容，与计算机屏幕输出的显示不同	machine_name% **su** Password:
AaBbCc123	要使用实名或值替换的占位符	用于删除文件的命令为 rm *filename*。
AaBbCc123	保留未译的新词或术语以及要强调的词（注意：某些强调的词在联机状态下以粗体显示）	这些称为 *Class* 选项。
新词术语强调	新词或术语以及要强调的词	**缓存**是存储在本地的副本。 **不要**保存文件。
《书名》	书名	阅读《用户指南》的第 6 章。

命令示例中的 Shell 提示符

下表显示了默认系统提示符和超级用户提示符。

表 P–6 Shell 提示符

Shell	提示符
UNIX 和 Linux 系统上的 C shell	`machine_name%`
UNIX 和 Linux 系统上的 C shell 超级用户	`machine_name#`
UNIX 和 Linux 系统上的 Bourne shell 和 Korn shell	`$`
UNIX 和 Linux 系统上的 Bourne shell 和 Korn shell 超级用户	`#`
Microsoft Windows 命令行	`C:\`

符号约定

下表对本书中可能使用的符号进行了解释。

表 P–7 符号约定

符号	描述	示例	含义
`[ ]`	包含可选的参数和命令选项。	`ls [-l]`	`-l` 不是必需选项。
`{ | }`	包含必需命令选项的选项集。	`-d {y|n}`	`-d` 选项要求您使用参数 `y` 或参数 `n`。
`${ }`	表示变量引用。	`${com.sun.javaRoot}`	引用 `com.sun.javaRoot` 变量的值。
-	连接需要同时按下的多个键。	Ctrl-A	在按 A 键的同时按 Ctrl 键。
+	连接需要连续按下的多个键。	Ctrl+A+N	按 Ctrl 键，再将其释放，然后按后续键。
→	表示图形用户界面中的菜单项选择。	“文件”→“新建”→“模板”	从“文件”菜单中选择“新建”。 从“新建”子菜单中选择“模板”。

文档、支持和培训

Sun 的 Web 站点提供了有关其他资源的信息，如下所示：

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

第三方 Web 站点引用

本文档引用了第三方 URL，并提供了其他相关信息。

注 – Sun 对本文档中提到的第三方 Web 站点的可用性不承担任何责任。对于此类站点或资源中的（或通过它们获得的）任何内容、广告、产品或其他材料，Sun 并不表示认可，也不承担任何责任。对于因使用或依靠此类站点或资源中的（或通过它们获得的）任何内容、产品或服务而造成的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。

搜索 Sun 产品文档

除了从 docs.sun.com Web 站点搜索 Sun 产品文档之外，还可以通过在搜索字段中键入以下语法来使用所选的搜索引擎进行搜索：

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

例如，要搜索 `Directory Server`，请键入以下内容：

```
"Directory Server" site:docs.sun.com
```

要将其他 Sun Web 站点（如 java.sun.com、www.sun.com 和 developers.sun.com）包含在搜索中，请在搜索字段中使用 `sun.com` 代替 `docs.sun.com`。

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。要共享您的意见，请访问 http://docs.sun.com，然后单击“发送意见”(Send Comments)。在联机表单中，提供完整的文档标题和文件号码。文件号码包含 7 位或 9 位数字，可在书的标题页或在文档的 URL 中找到该号码。例如，本书的文件号码为 820-0293。

第 1 部分

目录服务器管理

◆ ◆ ◆ 第 1 章

目录服务器工具

Sun Java™ System Directory Server Enterprise Edition 提供了一个浏览器界面和一些命令行工具，用于在复制环境下管理多个服务器、实例和后缀。本章提供了有关目录服务器管理工具的概述信息。

本章包含以下主题：

目录服务器管理概述

此文档集的其他指南中提供了有关目录服务器管理框架的信息。

- 有关目录服务器管理框架的概述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Deployment Planning Guide》中的“Directory Server Enterprise Edition Administration Model”。
- 有关目录服务器管理框架的更多详细参考信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的第 1 章 “Directory Server Overview”。

DSCC 和命令行的适用环境

Directory Server Enterprise Edition 提供两个用于管理目录服务器和目录代理服务器的用户界面：浏览器界面和命令行界面，浏览器界面即目录服务控制中心 (Directory Service Control Certer, DSCC)。

确定是否可以使用 DSCC 执行某个过程

本指南中的大多数过程都可以使用命令行或 DSCC 执行。本指南中的过程将说明如何使用命令行完成过程。在大多数情况下，可以使用 DSCC 执行相同的任务。如果可以使用 DSCC 执行某个过程，将在此过程的开头部分提供相关声明。

DSCC 联机帮助提供了有关如何使用 DSCC 执行本指南中过程的详细说明。

DSCC 的适用环境

与使用命令行相比，使用 DSCC 可以更轻松地执行某些操作和任务，如以下部分所述。一般来说，最好使用 DSCC 执行必须应用于多个服务器的命令。

查看服务器和后缀复制状态

DSCC 将显示一些表，这些表包含已在 DSCC 中注册的所有服务器实例、已配置的所有后缀以及每个后缀的状态。

服务器表位于“目录服务器”选项卡上，它显示服务器的操作状态。有关服务器的可能状态的完整列表，请参见目录服务器联机帮助。

后缀表位于“后缀”选项卡上，它显示复制状态信息，如条目数量以及所有丢失更改的数量和存留期。有关此表中显示的信息的详细信息，请参见目录服务器联机帮助。

管理服务器组

服务器组可帮助您监视和配置服务器。可以创建组并为这些组指定服务器。例如，可以按地理位置或功能对服务器进行分组。如果您有大量服务器，则可以对“目录服务器”选项卡上显示的服务器进行过滤，以便只显示组中的服务器。还可以将某个服务器的服务器配置（例如索引或缓存设置）复制到组中的所有其他服务器。有关如何设置和使用服务器组的说明，请参见目录服务器联机帮助。

复制配置设置

DSCC 允许您将现有服务器、后缀或复制协议的配置设置复制到一个或多个其他的服务器、后缀或复制协议。有关如何执行上述各项任务的信息，请参见目录服务器联机帮助。

配置复制

使用 DSCC 可以便捷地设置复制拓扑。只需创建服务器实例，然后使用 DSCC 提供的步骤指定每个服务器的角色即可。DSCC 将自动创建复制协议。有关如何使用 DSCC 配置复制的详细信息，请参见目录服务器联机帮助。

目录服务控制中心界面

目录服务控制中心 (Directory Service Control Certer, DSCC) 是一个用户界面，允许您使用浏览器管理目录服务器和目录代理服务器。

要配置 DSCC，请参见第 65 页中的 “配置 DSCC” 。有关使用 DSCC 的信息，请参见以下部分。

DSCC 的管理用户

DSCC 有几个管理登录用户。

- **操作系统用户。**创建服务器实例，是唯一有权使用 `dsadm` 命令在服务器实例上运行操作系统命令的用户。在某些情况下，DSCC可能会要求提供操作系统用户密码。此用户必须具有密码，并且必须能够创建目录服务器实例。
- **目录管理员。**服务器的 LDAP 超级用户。默认 DN 为 `cn=Directory Manager`。
- **目录管理者。**管理目录服务器。此用户与目录管理员具有相同的权限，但受访问控制、密码策略和验证要求的约束。可以根据需要创建任意数量的目录管理者。
- **目录服务管理员。**通过 DSCC 管理多台计算机上的服务器配置和数据。对于已在 DSCC 中注册的每个服务器，此用户与目录管理员具有相同的权限，并且是目录管理者组的成员。

▼ 访问 DSCC

如果在访问 DSCC 时遇到任何困难，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Installation Guide》中的“To Troubleshoot Directory Service Control Center Access”。

1 **确保 DSCC 已正确安装，如《 Sun Java System Directory Server Enterprise Edition 6.0 Installation Guide 》中的“Software Installation”中所述。**

2 **打开浏览器，并使用以下格式键入 DSCC 主机 URL：**

`https://`*hostname*`:6789`

例如：

```
https://host1:6789
```

其中 hostname 是安装了 DSCC 软件的系统。

Sun Java Web Console 的默认端口为 `6789`。

下图显示了 Sun Java Web Console 的登录窗口。

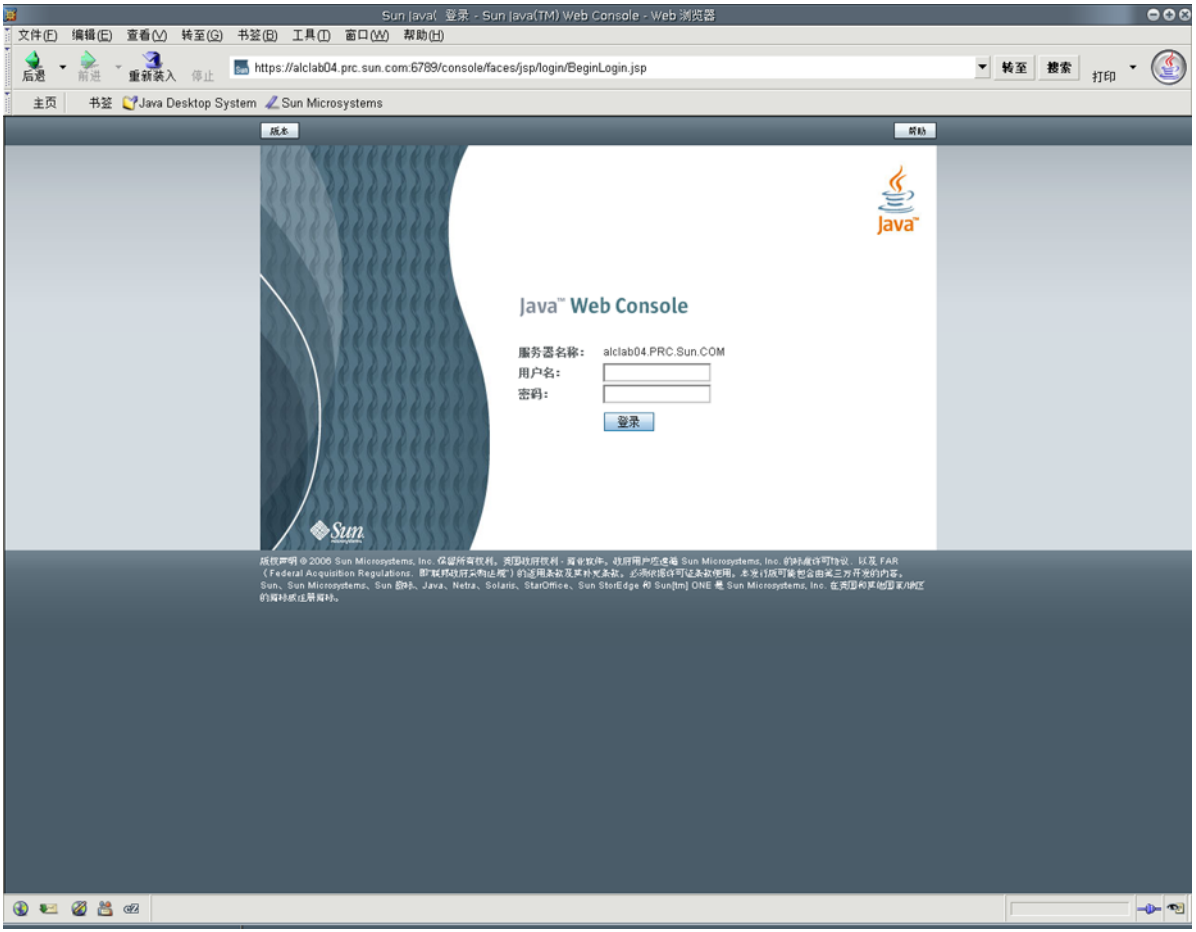


图 1–1 Sun Java Web Console 登录窗口

3 **登录到 Sun Java Web Console。**

- 如果您是首次登录到 Sun Java Web Console，请以 `root` 用户身份在安装了 DSCC 软件的系统上登录。
- 如果是后续登录，请键入您的操作系统用户名和密码。此用户应该具有启动、停止和管理目录服务器实例的权限。

登录时，您将看到一个应用程序列表。

4 **选择目录服务控制中心 (Directory Service Control Certer, DSCC)。**

将显示 DSCC 登录窗口。

5 **登录到 DSCC。**

如果您是首次登录到 DSCC，则必须设置目录服务管理员密码。在后续登录中，需要使用首次登录时设置的密码。

现在您已经登录到 DSCC，并位于“常见任务”选项卡。

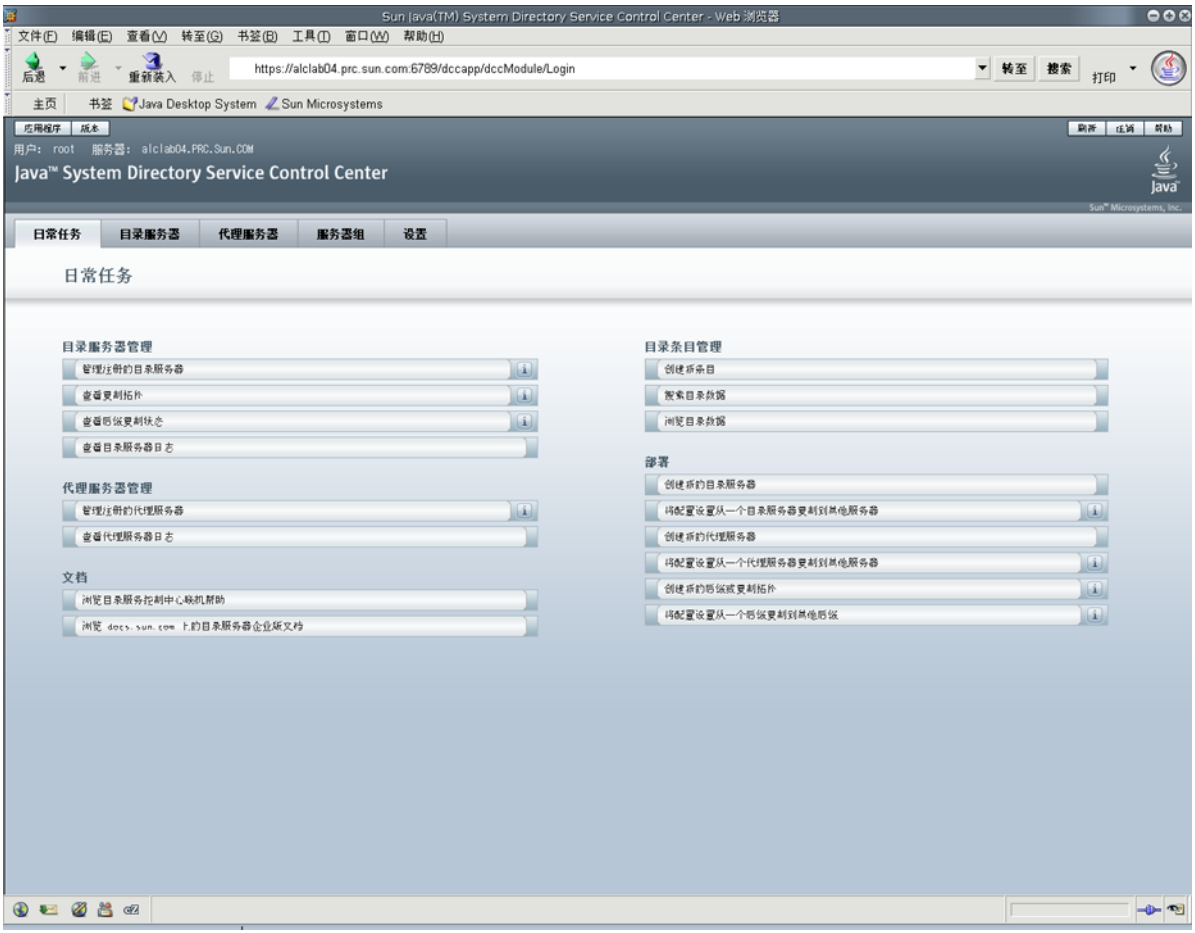


图 1–2 DSCC“常见任务”选项卡

6 **使用选项卡进行浏览。**

- “常见任务”选项卡包含常用窗口和向导的快捷方式。
- “目录服务器”选项卡显示由 DSCC 管理的所有目录服务器。要查看用于管理和配置特定服务器的更多选项，请单击服务器名称。
- “代理服务器”选项卡显示由 DSCC 管理的所有目录代理服务器。要查看用于管理和配置特定服务器的更多选项，请单击服务器名称。

注 – 有关如何使用 DSCC 执行任务的说明，请参见 DSCC 联机帮助。

DSCC 选项卡描述

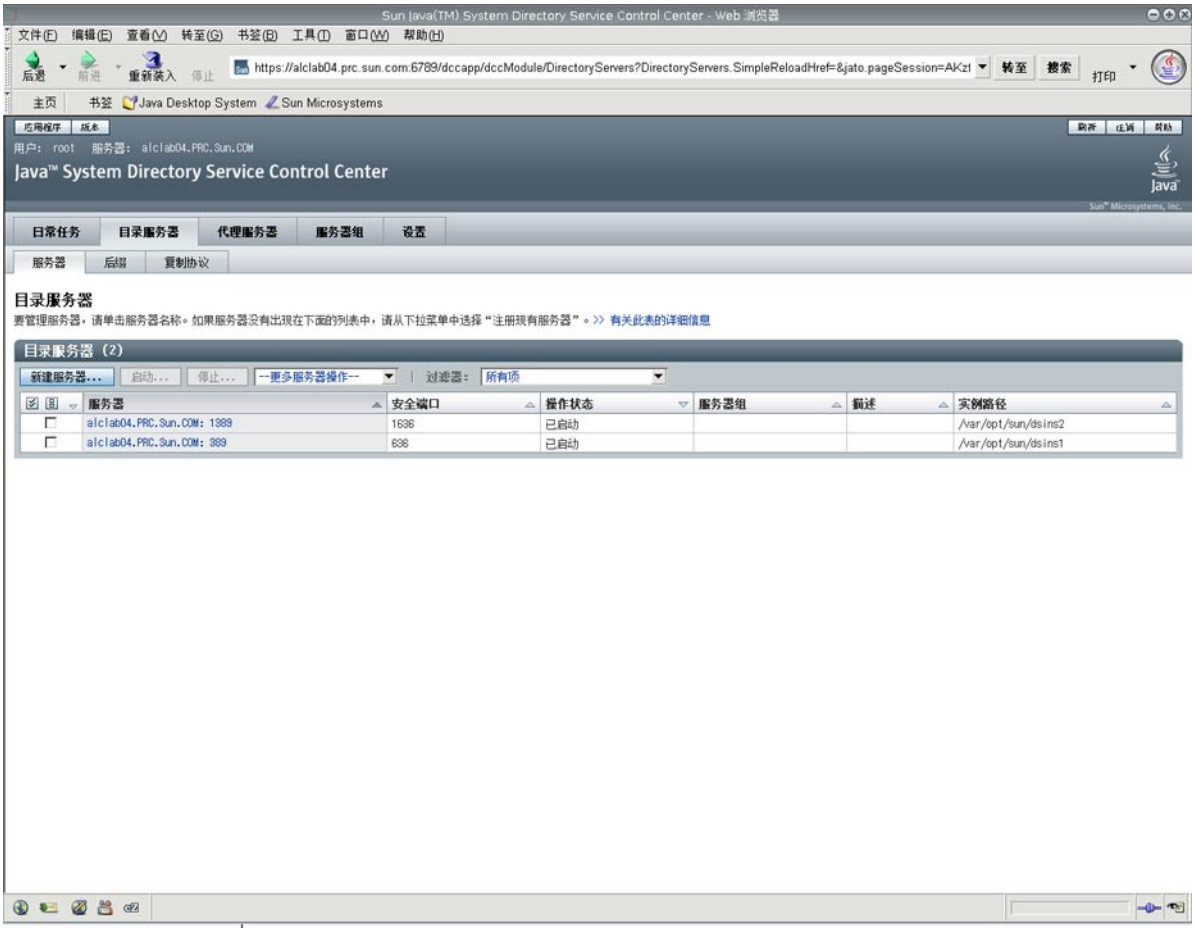


图 1–3 “服务器”子选项卡上的目录服务器列表

可以使用 DSCC 中的选项卡浏览界面。

“常见任务”选项卡

“常见任务”选项卡（请参见图 1–2）是打开 DSCC 时看到的第一个界面。它包含指向常用管理任务（如搜索目录数据、检查日志和管理服务器）的链接。

“目录服务器”选项卡

“目录服务器”选项卡（请参见图 1–3）列出已在 DSCC 中注册的所有目录服务器。将为每个服务器显示服务器状态和实例路径，实例路径指出实例所在的位置。

单击服务器名称时将出现另一个窗口，其中显示了只与该服务器相关的一组其他选项卡。

“代理服务器”选项卡

“代理服务器”选项卡列出已在 DSCC 中注册的所有目录代理服务器。将为每个服务器显示服务器状态和服务器实例路径，实例路径指出实例所在的位置。

单击服务器名称时将出现另一个窗口，其中显示了只与该服务器相关的一组其他选项卡。

“服务器组”选项卡

“服务器组”选项卡允许您将服务器指定给组，从而使服务器管理更加轻松。如果您有大量服务器，则可以通过使用过滤器只显示特定组中的服务器。还可以将某个服务器中的服务器配置（例如索引或缓存设置）复制到组中的所有其他服务器。

“设置”选项卡

此选项卡显示 DSCC 端口号，并允许您创建和删除目录服务管理员。

DSCC 联机帮助

联机帮助可提供以下内容：

- 当前所用页面的上下文有关帮助。
- 有关使用 DSCC 执行管理和配置过程的一般帮助。

可以在大多数页面中通过单击屏幕右上角的“帮助”按钮来访问帮助。在向导中，可以通过单击“帮助”选项卡访问帮助。还可以从“常见任务”选项卡访问联机帮助。

目录服务器命令行工具

在 DSCC 上执行的大多数任务都可以使用命令行工具来执行。这些工具允许您从命令行直接管理目录服务器，并可使用脚本管理服务器。

主要的目录服务器命令包括 `dsadm` 和 `dsconf`。可以使用这些命令执行备份、导出到 LDIF 和管理证书等操作。有关这些命令的信息，请参见 `dsadm`(1M) 和 `dsconf`(1M) 手册页。

本部分包含以下与目录服务器命令行工具有关的信息：

目录服务器命令的位置

目录服务器命令行工具位于默认的安装目录中：

install-path/ds6/bin

安装目录取决于操作系统。第 30 页中的 “默认路径和命令位置”中列出了所有操作系统的安装路径。

为 dsconf 设置环境变量

dsconf 命令需要某些可以使用环境变量预设的选项。如果在使用命令时未指定选项，或者未设置环境变量，则使用默认设置。可为以下选项配置环境变量：

-D *user DN*	用户绑定 DN。环境变量：LDAP_ADMIN_USER。默认：cn=Directory Manager。
-w *password-file*	用户绑定 DN 的密码文件。环境变量：LDAP_ADMIN_PWF。默认：提示输入密码。
-h *host*	主机名。环境变量：DIRSERV_HOST。默认：localhost。
-p *LDAP-port*	LDAP 端口号。环境变量：DIRSERV_PORT. 默认：389。

有关详细信息，请参见 dsconf(1M) 手册页。

dsadm 和 dsconf 的比较

下表显示 dsadm 和 dsconf 命令的比较。

表 1–1 dsadm 和 dsconf 命令的比较

	dsadm 命令	dsconf 命令
描述	必须直接在本地主机上运行的管理命令。例如： ■ 启动和停止服务器 ■ 创建服务器实例	可以从远程主机运行的管理命令。例如： ■ 启用复制 ■ 设置缓存大小
注释	必须停止服务器（dsadm stop 和 dsadm info 命令除外）。 服务器由服务器实例路径 (*instance-path*) 标识。 您必须具有服务器实例路径的操作系统访问权限。	服务器必须正在运行。 服务器由主机名 (-h) 端口 (-p) 或 LDAPS 安全端口 (-P) 标识。 如果未指定端口号，dsconf 将使用默认端口（对于 LDAP 来说为 389）。 您必须具有配置数据的 LDAP 访问权限，例如，具有 cn=admin,cn=Administrators,cn=config 用户身份。

获取有关使用 dsadm 和 dsconf 的帮助信息

有关如何使用 dsadm 和 dsconf 命令的完整信息，请参见 dsadm(1M) 和 dsconf(1M) 手册页。

- 要获取子命令列表，请键入相应的命令：

```
$ dsadm --help

$ dsconf --help
```

- 要获取有关如何使用子命令的信息，请键入相应的命令：

```
$ dsadm subcommand --help

$ dsconf subcommand --help
```

使用 dsconf 修改配置属性

许多 dsconf 子命令都允许用户查看和修改配置属性。

- 要列出目录服务器中使用的配置属性，请键入：

```
$ dsconf help-properties
```

- 要查找特定属性，请搜索帮助属性的输出。

  例如，如果您使用的是 UNIX® 平台，并要搜索与引用相关的所有属性，请使用以下命令。请注意，属性将按目标对象（如后缀 (SUF) 和服务器 (SER)）进行分组。

```
$ dsconf help-properties | grep -i referral
SER  referral-url                   rw  LDAP_URL | undefined
  Referrals returned to clients requesting a DN not stored in this
  Directory Server (Default: undefined)
SUF  referral-mode                  rw  disabled|enabled|only-on-write
 Specifies how referrals are used for requests involving the suffix
  (Default: disabled)
SUF  referral-url                   rw  LDAP_URL | undefined
  Server(s) to which updates are referred (Default: undefined)
SUF  repl-rewrite-referrals-enabled rw  on|off
  Specifies whether automatic referrals are overwritten (Default: off)
```

- 要查看服务器属性，请使用详细模式。以 UNIX 系统为例，请键入：

```
$ dsconf help-properties -v | grep -i referral-mode
SUF  referral-mode    rw  disabled|enabled|only-on-write  nsslapd-state
  Specifies how referrals are used for requests involving the suffix
  (Default: disabled)
```

有关单个属性的详细信息，请参见该属性的手册页。这些手册页位于《Sun Java System Directory Server Enterprise Edition 6.0 Man Page Reference》中。

使用 dsconf 设置多值属性

某些目录服务器属性可以具有多个值。用于指定这些值的语法如下：

```
$ dsconf set-container-prop -h host -p port container-name \
 property:value1 property:value2
```

例如，要为服务器设置多个加密密码，请使用以下命令：

```
$ dsconf set-server-prop -h host1 -p 1389 ssl-cipher-family:SSL_RSA_WITH_RC4_128_MD5 \
 ssl-cipher-family:SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
```

对于已包含值的多值属性，如果添加或修改了某个值，则必须重置**所有**值。例如，在前面介绍的方案中，如果您要添加加密密码，则必须在命令中包含所有其他的加密密码：

```
$ dsconf set-server-prop -h host1 -p 1389 ssl-cipher-family:SSL_RSA_WITH_RC4_128_MD5 \
 ssl-cipher-family:SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA \
 ssl-cipher-family:TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
```

如果要删除某个值，此规则同样适用。因此，要从前面示例的列表中删除 MD5 密码，请运行以下命令：

```
$ dsconf set-server-prop -h host1 -p 1389 ssl-cipher-family:SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA \
 ssl-cipher-family:TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
```

手册页

手册页提供了对目录服务器中使用的所有命令和属性的描述。此外，手册页还显示了如何在部署中使用命令的一些有用示例。

传统工具

为了提供向后兼容性，还在常规目录服务器工具中提供了传统工具。这些工具虽然仍旧存在，但已经过时。

◆ ◆ ◆ 第 2 章

目录服务器实例和后缀

本章介绍如何创建和管理目录服务器实例和后缀。许多其他的目录管理任务也是在后缀级别进行配置的，但这些内容将在本书的其他章节进行介绍。

本章包含以下主题：

快速创建服务器实例和后缀的过程

本章包含有关如何创建服务器实例和后缀的详细信息。如果您需要快速创建目录服务器实例和后缀，并导入一些示例数据，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Installation Guide》中的“Server Instance Creation”。

创建和删除目录服务器实例

本部分说明如何创建和删除目录服务器实例。

▼ 创建目录服务器实例

在管理数据之前，必须先使用命令行工具或浏览器界面目录服务控制中心 (Directory Service Control Center, DSCC) 创建目录服务器实例。在 DSCC 中，目录服务器实例通常简称为“目录服务器”。

创建目录服务器实例时，目录服务器所需的文件和目录将在您指定的 *instance-path* 中创建。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

如果使用 DSCC 创建新的服务器实例，则可以选择复制现有服务器中的部分或全部服务器配置设置。

1 创建新的目录服务器实例并设置实例路径。

```
$ dsadm create instance-path
```

系统将提示您为此服务器设置目录管理员密码。

要为服务器实例指定非默认端口号，或指定任何其他参数，请参见 `dsadm`(1M) 手册页。

例如，要在 `/local/ds` 目录中创建新实例，请使用以下命令：

```
$ dsadm create /local/ds
Choose the Directory Manager password:
Confirm the Directory Manager password:
Use 'dsadm start /local/ds' to start the instance
```

2 检查是否已正确创建服务器实例。

```
$ dsadm info isntance-path
```

例如：

```
$ dsadm info /local/ds1
Instance Path:     /local/ds1
Owner:             user1(group1)
Non-secure port:   1389
Secure port:       1636
Bit format:        64-bit
State:             Stopped
DSCC url:          -
Instance version:  D-A00
```

3 （可选的）如果已使用 Java Enterprise System 安装程序或本地软件包安装对目录服务器进行了安装，并且操作系统提供服务管理解决方案，则可以将服务器作为服务进行管理，如下表所示。

操作系统	命令
Solaris 10	如果在 Sun Cluster 环境中操作，请使用以下命令： `dsadm enable-service --type CLUSTER` *instance-path resource-group* 否则，请使用以下命令： `dsadm enable-service --type SMF` *instance-path*
Solaris 9	如果在 Sun Cluster 环境中操作，请使用以下命令： `dsadm enable-service --type CLUSTER` *instance-path resource_group* 否则，请使用以下命令： `dsadm autostart` *instance-path*
Linux、HP-UX	`dsadm autostart` *instance-path*
Windows	`dsadm enable-service --type WIN_SERVICE` *instance-path*

4 启动目录服务器。

```
$ dsadm start instance-path
```

注 – 服务器正在运行，但不包含数据或后缀。请使用 `dsconf` 创建后缀。

5 （可选的）使用以下任一方法注册服务器实例：

- 访问 URL `https://`*host*`:6789` 并通过 DSCC 注册服务器。
- 使用 `dsccreg add-server` 命令。
  有关详细信息，请参见 `dsccreg`(1M) 手册页。

6 如果您要使用密码策略，并且目录服务器实例为独立实例，或者，如果实例属于已迁移到 `DS6-only` 密码策略模式的复制拓扑，请将实例移动到该模式下。

```
$ dsconf pwd-compat -h host -p port to-DS6-migration-mode

## Beginning password policy compatibility changes .
## Password policy compatibility changes finished.

Task completed (slapd exit code: 0).
$ dsconf pwd-compat -h host -p port to-DS6-mode

## Beginning password policy compatibility changes .
## Password policy compatibility changes finished.

Task completed (slapd exit code: 0).
```

▼ 删除目录服务器实例

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 停止目录服务器。

```
$ dsadm stop instance-path
```

2 如果以前使用 DSCC 管理服务器，请使用命令行取消注册此服务器。

```
$ dsccreg remove-server /local/ds
Enter DSCC administrator's password:
/local/ds is an instance of DS
Enter password of "cn=Directory Manager" for /local/ds:
This operation will restart /local/ds.
Do you want to continue ? (y/n) y
Unregistering /local/ds from DSCC on localhost.
Connecting to /local/ds
Disabling DSCC access to /local/ds
Restarting /local/ds
```

有关详细信息，请参见 `dsccreg`(1M) 手册页。

3 （可选的）如果以前在服务管理解决方案中启用了服务器实例，请停止将此服务器作为服务进行管理。

操作系统	命令
Solaris 10	如果在 Sun Cluster 环境中操作，请使用以下命令： `dsadm disable-service --type CLUSTER` *instance-path* 否则，请使用以下命令： `dsadm disable-service --type SMF` *instance-path*
Solaris 9	如果在 Sun Cluster 环境中操作，请使用以下命令： `dsadm disable-service --type CLUSTER` *instance-path* 否则，请使用以下命令： `dsadm autostart --off` *instance-path*
Linux、HP-UX	`dsadm autostart --off` *instance-path*
Windows	`dsadm disable-service --type WIN_SERVICE` *instance-path*

4 删除服务器实例。

```
$ dsadm delete instance-path
```

注意 – 此命令将删除所有内容，包括数据库和数据。

启动、停止和重新启动目录服务器实例

要从命令行启动、停止或重新启动服务器，请分别使用 `dsadm start`、`dsadm stop` 和 `dsadm restart` 命令。

注 – 如果停止和重新启动的目录服务器实例在内存中配置了较大缓存以存储条目，则此缓存需要花费一些时间进行重新填充。当缓存再次填满时，实例的响应速度将会变慢。

这些命令必须由创建目录服务器的相同 UID 和 GID 运行，或由超级用户运行。例如，如果目录服务器以 `user1` 身份运行，则应该以 `user1` 身份运行 `start`、`stop` 和 `restart` 实用程序。

注 – 在 Solaris 上，基于角色的访问控制允许您以非超级用户身份运行目录服务器。

▼ 启动、停止和重新启动目录服务器

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。但这并不适用于启用和禁用服务管理的步骤。在启动和停止目录服务器时，必须通过命令行启用和禁用服务管理。

- **要启动、停止或重新启动目录服务器，请执行以下任一操作：**
  - 要启动服务器，请键入：

    ```
    $ dsadm start instance-path
    ```

    例如，要使用实例路径 `/local/ds` 启动服务器，请使用以下命令：

    ```
    $ dsadm start /local/ds
    ```
  - 要停止服务器，请键入：

    ```
    $ dsadm stop instance-path
    ```

    例如：

    ```
    $ dsadm stop /local/ds
    ```

- 要重新启动服务器，请键入：

  ```
  $ dsadm restart instance-path
  ```

  例如：

  ```
  $ dsadm restart /local/ds
  ```

创建后缀

创建目录服务器实例之后，必须为服务器的目录信息树 (Directory Information Tree, DIT) 创建一个或多个后缀。DIT 由服务器中的所有条目组成，这些条目使用各自的标识名 (Distinguished Name, DN) 进行标识。DN 的分层特性可创建分支和叶条目，从而以树的形式组织数据。DIT 是以后缀和子后缀的形式进行定义和管理的。DSCC 提供了用于创建和管理所有这些元素的控件。此外，也可以使用命令行工具。

有关一般情况下构建目录数据以及后缀的概念性信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Deployment Planning Guide》。

如以下过程中所述，您可以使用 `dsconf create-suffix` 命令在目录中创建后缀配置。由于在内部以相同方式管理根后缀和子后缀，因此从命令行创建这些后缀的过程几乎相同。此过程介绍了只与必需选项一起使用的 `dsconf create-suffix` 命令。有关此命令的其他选项的详细信息，请参见 `dsconf`(1M) 手册页，或运行以下命令：

```
$ dsconf create-suffix --help
```

任何管理用户都可以创建配置条目。但是，后缀的顶级条目**必须**由目录管理员创建，或以目录管理者身份（如 `cn=admin,cn=Administrators,cn=config`）创建。

▼ 创建后缀

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

如果使用 DSCC 创建新后缀，您可以选择复制现有后缀中的部分或全部后缀配置设置。

1 **创建根后缀。**

请确保服务器正在运行，然后键入以下命令：

```
$ dsconf create-suffix -h host -p port suffix-DN
```

其中 *suffix-DN* 是新后缀的完整 DN。对于根后缀，此约定为使用域组件 (`dc`) 命名属性。

例如，要创建 DN dc=example,dc=com 的后缀，请用以下命令：

```
$ dsconf create-suffix -h host1 -p 1389 dc=example,dc=com
```

此命令将以如下方式创建新后缀：

- 创建根后缀的顶级（或基）条目。
- 创建后缀和数据库在 cn=config 中的配置条目。
- 默认数据库名称基于后缀 DN。

有关所有后缀的信息（包括已创建的新后缀），请使用以下命令：

```
$ dsconf list-suffixes -h host -p port -v
```

-v 选项显示详细模式，此模式将显示后缀上的条目数量，以及所有复制信息。

注 – 如果有多个目录服务器实例，请使用 -h *host name* 和 -p *port number* 选项指定后缀所属的服务器实例。

如果要为数据库文件指定非默认路径，请使用 -L 选项。可以在后面的阶段中更改后缀数据库路径。要执行此操作，请使用命令 dsconf set-suffix-prop *suffix-DN* db-path:*new-db-path*，然后停止服务器，手动移动数据库文件，再重新启动服务器。

要查看创建后缀时可以使用的所有选项，请参见 dsconf(1M) 手册页。

---

2 创建子后缀（如果需要）：

```
$ dsconf create-suffix -h host -p port subSuffix-DN
```

然后，将此子后缀连接到根后缀。

```
$ dsconf set-suffix-prop -h host -p port subSuffix-DN parent-suffix-dn:parentSuffix-DN
```

其中 *parentSuffix-DN* 必须与上一步中的 *suffix-DN* 具有相同的值。子后缀的 *suffix-DN* 包含此子后缀的相对标识名 (Relative Distinguished Name, RDN) 及其父后缀的 DN。

例如，要创建子后缀 ou=Contractors,dc=example,dc=com，并将此子后缀连接到根后缀，请键入：

```
$ dsconf create-suffix -h host1 -p 1389 ou=Contractors,dc=example,dc=com
$ dsconf set-suffix-prop -h host1 -p 1389 ou=Contractors,dc=example,dc=com \
 parent-suffix-dn:dc=example,dc=com
```

当此条目添加到目录中时，服务器的数据库模块将自动在以下目录中创建数据库文件：

instance-path/db/*database-name*

其中 *database-name* 是基于部分后缀自动生成的名称。例如，在前面的示例中，*database-name* 应该为 Contractors

3 （可选的）使用数据初始化后缀。请参见第 185 页中的“初始化后缀”。

禁用或启用后缀

有时可能需要禁用后缀以进行维护，或出于安全原因禁用后缀内容。如果禁用后缀，服务器将无法读取或写入后缀内容以响应任何客户端操作。禁用后缀时，您将不再具有该后缀的访问权限，并且引用模式将自动设置为禁用。

▼ 禁用后缀然后再启用后缀

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的“目录服务控制中心界面”和 DSCC 联机帮助。

1 **禁用后缀。**

```
$ dsconf set-suffix-prop -h host -p port suffix-DN enabled:off
```

注 – 无法禁用已启用复制的后缀，因为复制后缀的大多数属性都由复制机制确定。

2 **启用后缀。**

```
$ dsconf set-suffix-prop -h host -p port suffix-DN enabled:on
```

设置引用并使后缀变为只读状态

如果要限制对后缀的访问权限而不完全禁用后缀，则可以修改访问权限以允许只读访问。在这种情况下，必须定义对其他服务器的引用以执行写入操作。此外，还可以同时拒绝读取和写入访问，然后为后缀上的所有操作定义引用。

引用还可用于临时将客户端应用程序指向其他服务器。例如，备份后缀内容时，可能要添加对其他后缀的引用。

如果后缀是复制环境中的使用方，则复制机制将确定引用设置的值。尽管可以手动修改引用设置，但引用在下次复制更新时将被覆盖。有关设置复制引用的信息，请参见第 215 页中的“执行高级使用方配置”。

引用是已标记的 URL，即 LDAP URL，其后可接空格字符和标签。例如：

```
ldap://phonebook.example.com:389/
```

或者：

```
ldap://phonebook.example.com:389/ou=All%20People,dc=example,dc=com
```

由于空格字符非常重要，因此对于引用的 URL 部分中的任何空格字符，都必须使用 `%20` 进行转义。

▼ 设置引用以使后缀变为只读状态

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1. **设置引用 URL。**

   ```
   $ dsconf set-suffix-prop -h host -p port suffix-DN referral-url:LDAP-URL
   ```

   其中 *LDAP-URL* 是有效的 URL，包含主机名、端口号和目标的 DN。

   例如：

   ```
   $ dsconf set-suffix-prop -h host1 -p 1389 dc=example,dc=com \
    referral-url:ldap://phonebook.example.com:389/
   ```

   可以指定任意数量的 LDAP URL。

2. **设置引用模式，使后缀变为只读状态。**

   ```
   $ dsconf set-suffix-prop -h host -p port suffix-DN referral-mode:only-on-write
   ```

   要使后缀无法用于读取和写入操作，并为所有请求返回引用，请将 `referral-mode` 设置为`enabled`。

3. **只要命令执行成功，后缀即变为只读或不可访问状态，并可返回引用。**

4. **（可选的）后缀变为可用状态时，请禁用引用，以使后缀再次变为读写状态。**

   ```
   $ dsconf set-suffix-prop -h host -p port suffix-DN referral-mode:disabled
   ```

   禁用引用时，后缀将自动变为读写状态，除非您通过将后缀的 `enabled` 属性设置为 `off` 禁用了此后缀。

删除后缀

删除后缀会将其整个分支从 DIT 中删除。

注 – 删除后缀时，将从目录中永久地删除该后缀的所有数据条目。此外，还将删除所有后缀配置信息，包括其复制配置。

您无法删除父后缀，但将其子后缀保留在 DIT 中作为新的根后缀。如果要删除包含子后缀的整个分支，则必须同时删除已删除的父后缀的子后缀及其可能的子后缀。

▼ 删除后缀

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **删除后缀配置条目：**

  ```
  $ dsconf delete-suffix -h host -p port [subSuffix-DN] suffix-DN
  ```

  此命令将从位于 *suffix-DN* 处的基条目开始，从服务器中删除后缀。在目录中将无法再看到或访问此后缀。

◆ ◆ ◆ 第 3 章

3 目录服务器配置

本章介绍如何配置目录服务器。您可以使用 `dsconf` 命令（请参见 `dsconf`(1M) 手册页）。

还可以使用目录服务控制中心 (Directory Service Control Center, DSCC)，此为首选方法。DSCC 将在配置过程中执行附加检查，这可以减少错误。此外，DSCC 还允许您将一个服务器实例的配置复制到另一个服务器实例中。有关使用 DSCC 的详细信息，请参见 DSCC 联机帮助。

本章包含以下主题：

使用 DSCC 修改配置

修改配置的推荐方法是使用 DSCC。此浏览器界面提供了基于任务的控件，可帮助您快速有效地设置配置。使用 DSCC 可以修改一个服务器上的配置设置，然后将该配置设置复制到其他服务器中。此外，DSCC 界面还为您管理配置的复杂性和相互依赖性。可以在 DSCC 联机帮助中找到使用 DSCC 修改配置的详细过程。

从命令行修改配置

可以通过编写使用命令行工具的脚本来自动完成配置任务。

可以通过命令行使用 dsconf 命令来修改配置。此命令使用 LDAP 修改 cn=config 子树。有关 dsconf 的详细信息，请参见第 45 页中的 “目录服务器命令行工具” 。

对于无法使用 dsconf 执行的任何任务，都可以使用 ldapmodify 命令。

注 – 如果要使用 dsconf set-server-prop 命令修改服务器配置属性，您需要了解可以修改的属性以及这些属性的默认值。可使用以下命令显示所有属性的帮助信息：

```
$ dsconf help-properties -v
```

可以搜索所需项的属性帮助。例如，在 UNIX 平台上，可键入以下内容来搜索内存缓存属性：

```
$ dsconf help-properties -v | grep cache
```

有关 cn=config 中配置条目的详细信息，以及所有配置条目和属性（包括允许值的范围）的完整描述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》。

修改 dse.ldif 文件

目录服务器将所有配置信息存储在以下文件中：

instance-path/config/dse.ldif

注意 – 通过直接编辑 dse.ldif 文件的内容来修改配置很容易出错，因此不推荐使用此方法。但是，如果选择手动编辑此文件，请在编辑文件之前停止服务器，并在完成编辑后重新启动该服务器。

---

dse.ldif 文件使用 LDAP 数据交换格式 (LDAP Data Interchange Format, LDIF)。LDIF 是条目、属性及属性值的文本表示，并且是 RFC 2849 (http://www.ietf.org/rfc/rfc2849 (http://www.ietf.org/rfc/rfc2849)) 中所述的标准格式。

dse.ldif 文件中的目录服务器配置由以下内容组成：

- cn=config 条目的属性和值。
- 位于 cn=config 下的子树中的所有条目及其属性和值。
- 根条目 ("") 和 cn=monitor 条目的对象类和访问控制指令。这些条目的其他属性由服务器生成。

只有拥有目录服务器实例的系统用户才具有读取和写入文件的权限。

目录服务器允许通过 LDAP 读取和写入所有配置设置。默认情况下，具有授权的任何人都可以读取目录的 cn=config 分支，但只有目录管理员 (cn=Directory Manager) 和 cn=Administrators,cn=config 下的管理用户才能写入该分支。管理用户可以查看和修改配置条目，就像任何其他目录条目一样。

不要在 cn=config 条目下创建非配置条目，因为这些条目将存储在 dse.ldif 文件中，而不像常规条目那样存储在高伸缩性的数据库中。因此，将许多条目（特别是可能经常更新的条目）存储在 cn=config 下可能会降低性能。但是，将复制管理员（提供方绑定 DN）等特殊用户条目存储在 cn=config 下可能非常有用，因为这样可以集中配置信息。

配置管理用户

目录服务器包含默认的管理用户，即目录管理员和 cn=admin,cn=Administrators,cn=config 用户。这两个用户具有相同的访问权限，但 cn=admin,cn=Administrators,cn=config 受 ACI 控制。

本部分说明如何创建具有超级用户权限的管理用户，以及如何配置目录管理员。

▼ 创建具有超级用户权限的管理用户

如果要创建与 cn=admin,cn=Administrators,cn=config 具有相同权限的新管理用户，请在 cn=Administrators,cn=config 组中创建新用户。此组中的所有用户都受全局 ACI 控制，此 ACI 允许具有与目录管理员相同的访问权限。

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

- **创建新的管理用户。**

  例如，要创建新用户 cn=Admin24,cn=Administrators,cn=config，请键入：

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=admin24,cn=Administrators,cn=config
changetype: add
objectclass: top
objectclass: person
userPassword: password
description: Administration user with the same access rights as Directory Manager.
```

  -D 和 -w 选项分别提供具有此条目创建权限的用户的绑定 DN 和密码。

▼ 配置目录管理员

目录管理员是具有特权的服务器管理员，与 UNIX 系统上的 root 用户类似。访问控制不适用于目录管理员。

大多数管理任务都不需要使用目录管理员。您可以使用 cn=admin,cn=Administrators,cn=config 用户或者在 cn=Administrators,cn=config 下创建的任何其他用户。需要使用目录管理员的任务只包括更改根 ACI 和复制故障排除任务（如修复复制和搜索逻辑删除）。

可以更改目录管理员的 DN 和密码，以及创建可供自动读取密码的文件。

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1 查找现有的目录管理员 DN。

```
$ dsconf get-server-prop -h host -p port root-dn
root-dn:cn=Directory Manager
```

2 根据需要修改目录管理员设置。

- 要修改目录管理员 DN，请键入：

  ```
  $ dsconf set-server-prop -h host -p port root-dn:new-root-dn
  ```

  如果目录管理员 DN 中存在空格，请使用引号。例如：

  ```
  $ dsconf set-server-prop -h host1 -p 1389 root-dn:"cn=New Directory Manager"
  ```

- 要更改目录管理员密码，请键入：

  ```
  $ dsconf set-server-prop -h host -p port root-pwd:new-root-dn-password
  ```

  如果出于安全考虑，您不希望将明文密码作为命令行参数进行传递，可创建一个用于设置密码的临时文件。

  ```
  $ echo password > /tmp/pwd.txt
  ```

  此文件只读取一次，并存储密码以供将来使用。请设置服务器根密码文件属性。

  ```
  $ dsconf set-server-prop -h host -p port root-pwd-file:/tmp/pwd.txt
  ```

  此命令将提示服务器读取密码文件。请在设置密码文件属性之后删除临时密码文件。

  ```
  $ rm /tmp/pwd.txt
  ```

保护配置信息

根目录服务器条目（使用零长度 DN "" 执行基本对象搜索时返回的条目）以及 cn=config、cn=monitor 和 cn=schema 下的子树包含由目录服务器自动生成的访问控制指令 (Access Control Instruction, ACI)。这些 ACI 用于确定目录条目的用户权限。如果用于评估目的，则这些 ACI 已经足够。但是对于任何生产部署，您都需要评估访问控制要求并设计您自己的访问控制。

如果出于安全考虑，您要隐藏一个或多个其他子树并保护您的配置信息，则必须在 DIT 上放置其他 ACI 。

- 将 ACI 属性放在位于要隐藏的子树基部的条目中。
- 将 ACI 放在 namingContexts 属性上的根 DSE 条目中。名为 namingContexts 的根 DSE 条目属性包含每个目录服务器数据库的基 DN 列表。
- 将 ACI 放在 cn=config 和 cn=monitor 子树上。子树 DN 也存储在 cn=config 和 cn=monitor 下的映射树条目中。

有关创建 ACI 的详细信息，请参见第 6 章。

配置 DSCC

本部分提供了以下有关配置 DSCC 的信息：

- 更改 Common Agent Container 端口号
- 重置目录服务管理者密码
- 延长 DSCC 会话自动超时延迟
- 配置 DSCC 的故障转移
- DSCC 故障排除

▼ 更改 Common Agent Container 端口号

默认的 Common Agent Container 端口号为 11162。Common Agent Container 将 DSCC 代理端口定义为 jmxmp-connector-port。如果出于管理考虑，需要为 DSCC 代理和 Common Agent Container 使用其他端口号，请使用以下过程：

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1 以超级用户身份验证 jmxmp-connector-port 的现有端口号。

```
$ su
Password:
# cacaoadm list-params
...
jmxmp-connector-port=11162
...
```

2 **更改 DSCC 代理端口号。**

更改 DSCC 代理端口号时，必须停止 Common Agent Container。

```
# cacaoadm stop
# cacaoadm set-param jmxmp-connector-port=new-port
# cacaoadm start
```

有关此命令的位置，请参见第 31 页中的 “命令位置” 。

3 **在 DSCC 中，取消注册您的服务器，然后使用新的 DSCC 代理端口号重新注册这些服务器。**

此外，在创建新服务器时，还必须指定非默认的 DSCC 代理端口号。

▼ 重置目录服务管理员密码

要重置目录服务管理员密码，请使用 DSCC，如以下过程所述。

1 **访问 DSCC，如第 41 页中的 “ 访问 DSCC ”中所述。**

2 **单击“设置”选项卡，然后选择“目录服务管理员”。**

3 **单击要更改密码的目录服务管理员的名称。**

4 **在属性屏幕中输入新密码。**

在“确认密码”字段中再次键入新密码以进行确认。单击“确定”保存所做的更改。

▼ 延长 DSCC 会话自动超时延迟

经过一段时间之后，您的 DSCC 会话将会超时，并且您将从 DSCC 中注销。请使用以下过程延长超时延迟。请注意，此过程将延长 DSCC 及 Sun Java Web Console 中所有其他应用程序的超时时间。

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1 **以超级用户身份延长超时延迟。**

```
# wcadmin add -p -a ROOT session.timeout.value=mm
```

其中 *mm* 是超时之前时间（以分钟为单位）。

例如，要将超时时间设置为两小时，请键入：

```
$ su
Password:
# wcadmin add -p -a ROOT session.timeout.value=120
```

```
Set 1 properties for the ROOT application.
# wcadmin list -p
Shared service properties (name, value):
    session.timeout.value  120
    ...
```

2 **重新启动 Sun Java Web Console。**

```
# smcwebserver restart
Shutting down Sun Java(TM) Web Console Version 3.0.1 ...
Starting Sun Java(TM) Web Console Version 3.0.1 ...
The console is running.
```

有关这些命令的位置，请参见第 31 页中的 “命令位置”。

配置 DSCC 的故障转移

DSCC 会显示已在 DSCC 中注册的服务器。

如果安装有 DSCC 的计算机出现故障，您可以在其他计算机上安装 DSCC，然后重新注册您的服务器。但这么做可能会花费很多时间。如果要通过 DSCC 立即访问服务器，则可以配置 DSCC 故障转移。

要配置 DSCC 故障转移，请考虑以下注意事项：

- 已注册的服务器的所有信息都存储在 DSCC 注册表中。此注册表为目录服务器实例。可以使用管理命令 `dsadm` 和 `dsconf` 管理此注册表。
- DSCC 注册表具有以下默认特性：

服务器实例	Solaris — `/var/opt/SUNWdsee/dscc6/dcc/ads`
	Linux 和 HP-UX — `/var/opt/sun/dscc6/dcc/ads`
	Windows — `C:\Program Files\Sun\DSEE\var\dscc6\dcc\ads`
后缀	`cn=dscc`
端口	LDAP `3998`、LDAPS `3999`

- 在两台或多台计算机上安装 DSCC 之后，可以设置 DSCC 注册表后缀之间的复制。可以使用第 10 章中所述的复制命令行过程。或者，要获取设置简单复制配置的示例，请参见 `dsconf`(1M) 手册页。

  设置复制之后，可以从不同计算机访问 DSCC 中注册的相同服务器。例如，如果设置了 `host1` 和 `host2` 上 DSCC 注册表后缀之间的复制，则可以在 `https://host1:6789` 或 `https://host2:6789` 上使用 DSCC 管理相同的服务器。如果主机发生故障，则可以从其他主机访问 DSCC。

DSCC 故障排除

有关 DSCC 故障排除的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Installation Guide》中的“To Troubleshoot Directory Service Control Center Access”。

更改目录服务器端口号

可以使用 DSCC 或 `dsconf set-server-prop` 命令修改用户目录服务器的 LDAP 端口号或 LDAPS 安全端口号。

如果更改端口号，请注意以下事项：

- 如果设置了非特权端口号，并将目录服务器安装到其他用户可以访问的计算机上，则该端口号可能存在被其他应用程序占用的危险。换句话说，其他应用程序可能会绑定到相同的地址/端口对。此恶意应用程序可能会接着处理针对目录服务器的请求。也就是说，此恶意应用程序可用于捕获验证过程中使用的密码，从而更改客户端请求或服务器响应，或者产生拒绝服务攻击。要避免此安全风险，请使用 `nsslapd-listenhost` 属性指定目录服务器侦听的接口（地址）。有关此属性的详细信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》。

如果使用命令行更改端口号，请注意以下事项：

- 如果在其他服务器上定义的复制协议中引用了目录服务器，则必须更新此复制协议以使用新的端口号。
- 如果以前使用过 DSCC 管理服务器，则更改端口号后将暂时无法查看此服务器。要再次查看此服务器，必须先取消注册此服务器，然后在 DSCC 中使用新端口号对其进行重新注册。

▼ 修改端口号、启用端口和禁用端口

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 验证端口的现有设置。

```
$ dsconf get-server-prop -h host -p port port-type
```

其中 *port-type* 为以下任一选项：

`ldap-port`	LDAP 默认端口
`ldap-secure-port`	LDAPS 安全端口
`dsml-port`	DSML 默认端口

`dsml-secure-port`　DSML 安全端口

例如，要显示 LDAPS 安全端口，请键入：

```
$ dsconf get-server-prop -h host1 -p 2501 ldap-secure-port
Enter "cn=Directory Manager" password:
ldap-secure-port : 2511
```

如果返回结果为整数，则端口处于启用状态。如果返回结果为 `disabled`，则端口处于禁用状态。

注 – 还可以使用 `dsadm` 列出 LDAP 默认端口和 LDAPS 安全端口

---

2 **修改端口号或启用端口（如果需要）。**

```
$ dsconf set-server-prop -h host -p port port-type:new-port
```

例如，要将 LDAP 端口号从 1389 更改为 1390，请使用以下命令：

```
$ dsconf set-server-prop -h host1 -p 1389 ldap-port:1390
```

要在端口号 2250 上启用 DSML 安全端口，请使用以下命令：

```
$ dsconf set-server-prop -h host1 -p 1389 dsml-secure-port:2250
```

3 **禁用端口（如果需要）。**

```
$ dsconf set-server-prop -h host -p port port-type:disabled
```

例如，要禁用 DSML 安全端口，请使用以下命令：

```
$ dsconf set-server-prop -h host1 -p 1389 dsml-secure-port:disabled
```

配置 DSML

除了处理通过轻量目录访问协议 (Lightweight Directory Access Protocol, LDAP) 发送的请求之外，目录服务器还响应以目录服务标记语言版本 2 (Directory Service Markup Language version 2, DSMLv2) 发送的请求。DSML 是供客户端对目录操作进行编码的另一种方法。服务器将使用所有相同的访问控制和安全功能，以处理任何其他请求的方式来处理 DSML 请求。DSML 处理允许许多其他类型的客户端访问您的目录内容。

目录服务器支持通过超文本传输协议 (Hypertext Transfer Protocol, HTTP/1.1) 使用 DSMLv2，并将简单对象访问协议 (Simple Object Access Protocol, SOAP) 版本 1.1 用作编程协议来传输 DSML 内容。有关这些协议的详细信息以及 DSML 请求的示例，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的第 10 章 “Directory Server DSMLv2”。

本部分包含以下主题：

- 启用和禁用 DSML-over-HTTP 服务
- 配置 DSML 安全性
- DSML 标识映射

▼ 启用 DSML-over-HTTP 服务

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1 **将 DSML 模式设置为** `on`。

```
$ dsconf set-server-prop -h host -p port dsml-enabled:on
```

2 **设置安全 DSML 端口。**

```
$ dsconf set-server-prop -h host -p port dsml-secure-port:port
```

3 **设置非安全 DSML 端口。**

```
$ dsconf set-server-prop -h host -p port dsml-port:port
```

默认情况下，此端口设置为 `80`

4 **重新启动服务器。**

```
$ dsadm restart instance-path
```

接下来的操作 根据定义的参数和属性值，DSML 客户端可以使用以下 URL 将请求发送到此服务器：

`http://`*host*`:`*DSML-port*`/ relative-URL`

`https://`*host*`:`*secure-DSML-port* `/relative-URL`

▼ 禁用 DSML-over-HTTP 服务

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1 **将 DSML 模式设置为** `off`。

```
$ dsconf set-server-prop -h host -p port dsml-enabled:off
```

2 **将安全 DSML 端口设置为** `disabled`。

```
$ dsconf set-server-prop -h host -p port dsml-secure-port:disabled
```

3 **重新启动服务器。**

```
$ dsasm restart instance-path
```

▼ 配置 DSML 安全性

可以配置接受 DSML 请求时所需的安全级别。要执行此操作，必须配置 DSML 客户端验证。

- **设置 DSML 客户端验证模式。**

```
$ dsconf set-server-prop -h host -p port dsml-client-auth-mode:dsml-mode
```

dsml-mode 可为以下任一选项：

- `http-basic-only` - 此为默认值。服务器使用 HTTP“授权”头的内容来查找可以映射到目录中某个条目的用户名。此过程及其配置都通过 SSL 进行了加密，但不使用客户端证书。相关内容如第 71 页中的 “DSML 标识映射”中所述。
- `client-cert-only` - 服务器使用客户端证书中的凭证来识别客户端。使用此值时，所有 DSML 客户端都必须使用安全 HTTPS 端口发送 DSML 请求并提供证书。服务器会检查客户端证书是否与目录中的条目相匹配。有关详细信息，请参见第 5 章。
- `client-cert-first` - 服务器首先尝试使用客户端证书（如果提供）来验证客户端。如果未提供证书，服务器将使用“授权”头的内容来验证客户端。

如果 HTTP 请求中既未提供证书又未提供“授权”头，服务器将使用匿名绑定执行 DSML 请求。匿名绑定还将用于以下情形：

- 指定 `client-cert-only` 时，客户端提供了有效的“授权”头但未提供证书。
- 指定 `http-basic-only` 时，客户端提供了有效证书但未提供“授权”头。

无论使用哪种客户端验证方法，如果提供证书但该证书与条目不匹配，或者指定 HTTP“授权”头但无法将其映射到用户条目，则 DSML 请求都将被拒绝，并显示错误消息 403：“禁止”。

DSML 标识映射

执行无证书的基本验证时，目录服务器将使用**标识映射**机制来确定接受 DSML 请求时要使用的绑定 DN。此机制将从 HTTP 请求的“授权”头中提取信息，以确定要用于绑定的标识。

DSML/HTTP 的默认标识映射由服务器配置中的以下条目指定。

```
dn: cn=default,cn=HTTP-BASIC,cn=identity mapping,cn=config
objectClass: top
objectClass: nsContainer
```

```
objectClass: dsIdentityMapping
cn: default
dsSearchBaseDN: ou=people
dsSearchFilter: (uid=${Authorization})
```

此配置表明，服务器应该使用 HTTP 用户 ID 作为目录服务器后缀中所存储的 DN 的 `uid` 值。例如，如果 HTTP 用户为 `bjensen`，服务器将尝试使用 DN `uid=bjensen,ou=people` 执行绑定。

因此，要使映射正常工作，您必须完成 `dsSearchBaseDN` 的值。例如，可以将 `dsSearchBaseDN` 的值更改为 `ou=people,dc=example,dc=com`。这样，如果 HTTP 用户为 `bjensen`，服务器将尝试使用 DN `uid=bjensen,ou=people,dc=example,dc=com` 执行绑定。

```
dn: cn=default,cn=HTTP-BASIC,cn=identity mapping,cn=config
objectClass: top
objectClass: nsContainer
objectClass: dsIdentityMapping
cn: default
dsSearchBaseDN: ou=people,dc=example,dc=com
dsSearchFilter: (uid=${Authorization})
```

在映射条目属性 `dsSearchFilter` 中，可以使用 `${`*header*`}` 格式的占位符，其中 *header* 是 HTTP 头的名称。

以下是 DSML 映射中最常用的头。

`${Authorization}`	此字符串将由 HTTP“授权”头中包含的用户名替换。“授权”头同时包含用户名及其密码，但在此占位符中只替换用户名。
`${From}`	此字符串将由 HTTP“发件人”头中可能包含的电子邮件地址替换。
`${host}`	此字符串将由 DSML 请求 URL 中的主机名和端口号（即服务器的主机名和端口号）替换。

要使 DSML 请求执行其他类型的标识映射，请为 HTTP 头定义新的标识映射。

▼ 为 HTTP 头定义新的标识映射

1 编辑默认的 DSML-over-HTTP 标识映射，或为此协议创建自定义映射。

映射条目必须位于 `cn=HTTP-BASIC,cn=identity mapping,cn=config` 条目下。

可以从命令行中使用 `ldapmodify` 命令添加此条目，如第 82 页中的 “使用 ldapmodify 添加条目”中所述。

2. **重新启动目录服务器以使新映射生效。**

   将首先评估自定义映射。如果没有成功的自定义映射，则评估默认映射。如果所有映射都无法确定 DSML 请求的绑定 DN，将禁止并拒绝该 DSML 请求（错误 403）。

将服务器设置为只读

目录中的每个后缀均可单独设为只读模式，并可返回特定的引用（如果已定义）。目录服务器也为应用于所有后缀并可返回全局引用（如果已定义）的服务器提供了只读模式。

服务器只读模式旨在允许管理员在执行特定任务（如为后缀重新编制索引）时阻止对目录内容执行修改操作。因此，服务器只读模式不适用于以下配置分支：

- `cn=config`
- `cn=monitor`
- `cn=schema`

无论只读设置如何，都应始终使用访问控制指令 (Access Control Instruction, ACI) 保护这些分支，以防止非管理用户对其进行修改（请参见第 6 章）。全局只读模式可阻止对目录中的所有其他后缀执行更新操作，包括目录管理员发起的更新操作。

启用只读模式后，还会中断后缀上的复制操作。主副本不再有任何要复制的更改，但它还会继续复制启用只读模式前所做的全部更改。在禁用只读模式之前，使用方副本不会收到更新。多主复制环境中的主服务器没有任何要复制的更改，并且无法从其他主服务器接收更新。

▼ 启用或禁用服务器只读模式

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1. **启用全局只读模式。**

   ```
   $ dsconf set-server-prop -h host -p port read-write-mode:read-only
   ```

2. **准备就绪后，禁用只读模式。**

   ```
   $ dsconf set-server-prop -h host -p port read-write-mode:read-write
   ```

配置内存

本部分提供有关管理不同类型内存的信息。有关不同缓存类型的描述以及缓存调整的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的第 5 章 “Directory Server Data Caching”。

填充缓存

填充缓存是指将数据填入缓存中，以使后续的目录服务器行为能反映出正常的操作性能，而不是提升的性能。如果希望在执行基准测试时获得可再现的结果，或者要测量和分析潜在的优化，则填充缓存非常有用。

请尽量避免主动填充缓存。在测量性能之前，让客户端与目录服务器之间的一般或典型交互操作来填充缓存。

可以在 `http://www.slamd.com` 找到用于填充数据库缓存的工具。

▼ 修改数据库缓存

注意 – 修改缓存可能会严重影响服务器性能。修改缓存时请谨慎操作。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1. **获取当前的数据库缓存级别。**

   ```
   $ dsconf get-server-prop -h host -p port db-cache-size
   ```

2. **更改数据库缓存级别。**

   ```
   $ dsconf set-server-prop -h host -p port db-cache-size:sizeM
   ```

   其中 *size* 指大小（以 MB 为单位）。

▼ 监视数据库缓存

安装时的默认缓存级别适合于测试环境，而不适用于生产环境。进行调整时，您可能需要监视服务器的数据库缓存。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

- **监视数据库缓存。**

```
$ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
 -b "cn=monitor,cn=ldbm database,cn=plugins,cn=config" "(objectclass=*)"
```

如果数据库缓存足够大，并且已进行了填充，则命中率 (dbcachehitratio) 应该很高。此外，读入的页面数 (dbcachepagein) 和写出的干净页面数 (dbcacheroevict) 应该很低。此处的“高”和“低”是相对于部署限制而言的。

▼ 监视条目缓存

进行调整时，您可能需要检查一个或多个后缀的条目缓存。可使用以下过程查看条目缓存级别。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

- **监视条目缓存。**

```
$ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
 -b "cn=monitor,cn=db-name,cn=ldbm database,cn=plugins,cn=config" "(objectclass=*)"
```

如果后缀的条目缓存足够大（可以容纳该后缀中的大多数条目），并且已经填充了缓存，则命中率 (entrycachehitratio) 应该很高。

如果已经填充了缓存，您将会看到，当先前为空的条目缓存被填满时，条目缓存大小 (currententrycachesize) 将接近于最大条目缓存大小 (maxentrycachesize)。理想情况下，条目的数量 (currententrycachecount) 应该等于或非常接近于后缀中的条目总数 (ldapentrycachecount)。

▼ 修改条目缓存

注意 – 修改缓存可能会严重影响服务器性能。修改缓存时请谨慎操作。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1. **获取当前的条目缓存级别。**

```
$ dsconf get-suffix-prop -h host -p port suffix-DN entry-cache-count
```

2. **更改条目缓存级别。**

```
$ dsconf set-suffix-prop -h host -p port suffix-DN entry-cache-count:integer
```

其中 *integer* 指条目的数量。

3 **更改条目缓存大小。**

```
$ dsconf set-suffix-prop -h host -p port suffix-DN entry-cache-size:integer
```

其中 *integer* 指大小（以位为单位）。

▼ 配置堆内存阈值

如果要限制 `nsslapd` 进程所使用的堆内存量，可以设置动态内存占用的阈值。当目录服务器在资源共享或资源稀少的计算机上运行时，可能需要设置此阈值。

注 – 只能在 Solaris™ 和 Linux 平台上设置此阈值。

有关调整内存大小的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Deployment Planning Guide》中的“Directory Server and Memory”。

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1 **设置堆内存高端阈值的最大值。**

```
$ dsconf set-server-prop -h host -p port heap-high-threshold-size:value
```

有关 `ds-maxheap-high` 值的建议，请参见 `ds-maxheaphigh`(5dsconf) 手册页。

2 **设置堆内存低端阈值的最大值（可选）。**

```
$ dsconf set-server-prop -h host -p port heap-low-threshold-size:value
```

有关 `ds-maxheap-low` 值的建议，请参见 `ds-maxheaphigh`(5dsconf) 手册页，其中包含针对 `ds-maxheap-high` 和 `ds-maxheap-low` 提供的建议。

为每个客户端帐户设置资源限制

可以在服务器上控制每个客户端帐户的搜索操作资源限制。可以在帐户的操作属性中设置这些限制，然后目录服务器会基于客户端用于绑定到目录的帐户来实施这些限制。

可以设置以下限制：

- 浏览限制指定搜索操作可检查的最大条目数。
- 大小限制指定响应搜索操作时返回的最大条目数。
- 时间限制指定处理搜索操作所花费的最长时间。
- 空闲超时指定在断开连接之前客户端连接可以保持空闲状态的最长时间。

注 – 默认情况下，目录管理员使用资源时可以不受限制。

在特定用户帐户上设置的资源限制优先于在服务器范围的配置中设置的资源限制。本部分提供了有关为每个帐户设置资源限制的信息。

本部分中提供的示例直接在条目的属性中设置资源限制。还可以使用服务类 (Class of Service, CoS) 机制设置帐户的资源限制。为客户端应用程序检索条目时，CoS 机制将生成计算后的属性。有关定义 CoS 的详细信息，请参见第 199 页中的 “服务类”。

▼ 查看服务器资源限制设置

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **使用 `dsconf get-server-prop` 命令读取资源限制服务器属性。**

```
$ dsconf get-server-prop -h host -p port look-through-limit search-size-limit \
 search-time-limit idle-timeout
look-through-limit  :  5000
search-size-limit   :  2000
search-time-limit   :  3600
idle-timeout        :  none
```

从上述输出可以看出，搜索操作最多可浏览 5000 个条目，最多可返回 2000 个条目，并且最多可使用一小时（3600 秒）的服务器时间来处理搜索。

▼ 设置帐户的浏览限制

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **使用 `ldapmodify` 命令设置 `nsLookThroughLimit` 的值。**

  以下命令可删除 Barbara Jensen 的浏览限制：

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=people,dc=example,dc=com
changetype: modify
add: nsLookThroughLimit
nsLookThroughLimit: -1
^D
modifying entry uid=bjensen,ou=people,dc=example,dc=com
```

```
^D
$
```

▼ 设置帐户的大小限制

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

- **使用 `ldapmodify` 命令设置 `nsSizeLimit` 的值。**

  以下命令可删除 Barbara Jensen 的大小限制：

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=people,dc=example,dc=com
changetype: modify
add: nsSizeLimit
nsSizeLimit: -1
^D
modifying entry uid=bjensen,ou=people,dc=example,dc=com

^D
$
```

▼ 设置帐户的时间限制

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

- **使用 `ldapmodify` 命令设置 `nsTimeLimit` 的值。**

  以下命令可删除 Barbara Jensen 的时间限制：

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=people,dc=example,dc=com
changetype: modify
add: nsTimeLimit
nsTimeLimit: -1
^D
modifying entry uid=bjensen,ou=people,dc=example,dc=com

^D
$
```

▼ 设置帐户的空闲超时

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **使用 `ldapmodify` 命令设置 `nsIdleTimeout` 的值。**

  以下命令将 Barbara Jensen 的空闲超时时间设置为五分钟（300 秒）：

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=people,dc=example,dc=com
changetype: modify
add: nsIdleTimeout
nsIdleTimeout: 300
^D
modifying entry uid=bjensen,ou=people,dc=example,dc=com

^D
$
```

◆ ◆ ◆ 第 4 章

目录服务器条目

本章讨论如何管理目录中的数据条目。此外，还介绍如何设置引用以及如何加密属性值。

规划目录部署时，您需要根据特性对目录将包含的数据类型进行分类。在创建条目和修改默认模式之前，请阅读《Sun Java System Directory Server Enterprise Edition 6.0 Deployment Planning Guide》中的相关章节。

除非定义了相应的访问控制指令 (Access Control Instruction, ACI)，否则将无法修改目录。有关详细信息，请参见第 6 章。

本章包含以下主题：

管理条目

管理条目的最佳方式取决于环境：

- 如果主要将 DSCC 用于管理，并且只需搜索或修改少量条目，请使用 DSCC。有关 DSCC 的详细信息，请参见第 41 页中的 “目录服务控制中心界面”。
- 如果不在目录服务器上执行任何管理任务，并且只需搜索或修改少量条目，请使用目录编辑器。有关目录编辑器的信息，请参见《Sun Java System Directory Editor 1 2005Q1 Installation and Configuration Guide》。
- 如果要搜索或修改大量条目，请使用命令行实用程序 `ldapmodify` 和 `ldapdelete`。

使用 DSCC 管理条目

DSCC 允许您查看条目的所有可读属性以及编辑其可写属性。此外，您还可以添加和删除属性、设置多值属性，以及管理条目的对象类。有关如何使用 DSCC 管理条目的详细信息，请参见 DSCC 联机帮助。有关 DSCC 的一般详细信息，请参见第 41 页中的 “目录服务控制中心界面”。

使用目录编辑器管理条目

目录编辑器是一种易于使用的目录编辑工具，允许管理员和最终用户搜索、创建和编辑数据。此数据采用用户、组和容器的形式。

管理条目 ldapmodify 和 ldapdelete

`ldapmodify` 和 `ldapdelete` 命令行实用程序提供了用于添加、编辑和删除目录内容的完整功能。可以使用这些实用程序管理服务器的配置条目和用户条目中的数据。还可以使用这些实用程序编写脚本，以便对一个或多个目录执行批量管理。

本书中的过程几乎都使用了 `ldapmodify` 和 `ldapdelete` 命令。以下部分介绍执行这些过程所需的基本操作。有关 `ldapmodify` 和 `ldapdelete` 命令的详细信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》。

命令行实用程序的输入始终采用 LDIF 格式，可以直接从命令行或通过输入文件提供此输入。以下部分提供了与 LDIF 输入有关的信息，后续部分将介绍适用于每种修改类型的 LDIF 输入。

有关正确格式化 LDIF 输入的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Guidelines for Providing LDIF Input”。

以下部分介绍这些基本操作：

- 第 82 页中的 “使用 ldapmodify 添加条目”
- 第 84 页中的 “使用 `ldapmodify` 修改条目”
- 第 88 页中的 “使用 `ldapdelete` 删除条目”
- 第 88 页中的 “使用 `ldapmodify` 删除条目”
- 第 88 页中的 “使用 `ldapsearch` 搜索条目”

使用 ldapmodify 添加条目

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

可以使用 `ldapmodify` 的 `-a` 选项向目录中添加一个或多个条目。以下示例将创建包含用户的结构条目，然后再创建用户条目：

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: ou=People,dc=example,dc=com
objectclass: top
objectclass: organizationalUnit
ou: People
description: Container for user entries

dn: uid=bjensen,ou=People,dc=example,dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetorgPerson
uid: bjensen
givenName: Barbara
sn: Jensen
cn: Babs Jensen
telephoneNumber: (408) 555-3922
facsimileTelephoneNumber: (408) 555-4000
mail: bjensen@example.com
userPassword: secret
```

`-D` 和 `-w` 选项分别提供用户的绑定 DN 和密码（该用户具有创建这些条目的权限）。`-a` 选项表示将添加 LDIF 中的所有条目。然后将按 DN 和属性值列出每个条目，各条目之间以空行分隔。`ldapmodify` 实用程序将在输入每个条目后创建该条目，并会报告任何错误。

按照约定，条目的 LDIF 将列出以下属性：

1. 条目 DN。
2. 对象类列表。
3. 一个或多个命名属性。此属性是 DN 中使用的属性，它不一定属于必需属性。
4. 所有对象类的必需属性列表。
5. 要包含的所有允许的属性。

键入 `userPassword` 属性值时，请提供明文形式的密码。服务器将对此值进行加密，并且只存储加密的值。请务必限制读取权限以保护 LDIF 文件中出现的明文密码。

还可以使用另一种形式的 LDIF，它不要求在命令行中使用 `-a` 选项。此形式的优点是可以合并条目添加语句和条目修改语句，如以下示例所示。

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: ou=People,dc=example,dc=com
changetype: add
objectclass: top
objectclass: organizationalUnit
ou: People
```

```
description: Container for user entries

dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: add
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetorgPerson
uid: bjensen
givenName: Barbara
sn: Jensen
cn: Barbara Jensen
telephoneNumber: (408) 555-3922
facsimileTelephoneNumber: (408) 555-4000
mail: bjensen@example.com
userPassword: secret
```

`changetype: add` 关键字表示应该使用所有后续属性创建具有给定 DN 的条目。所有其他选项和 LDIF 约定都与本部分前面的叙述相同。

在这两个示例中，都可以使用 `-f` *filename* 选项从文件（而非终端输入）中读取 LDIF。LDIF 文件所包含的格式必须与终端输入使用的格式相同，这取决于您是否使用 `-a` 选项。

使用 `ldapmodify` 修改条目

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

可以使用 `changetype: modify` 关键字在现有条目中添加、替换或删除属性及属性值。指定 `changetype: modify` 时，还必须提供一个或多个更改操作，以表明将如何修改条目。以下示例中显示了三种可能的 LDIF 更改操作：

```
dn: entryDN
changetype: modify
add: attribute
attribute: value...
-
replace: attribute
attribute: newValue...
-
delete: attribute
[attribute: value]
...
```

可以使用独占一行的连字符 (`-`) 分隔相同条目上的操作，并可使用空行来分隔不同条目上的操作组。还可以为每个操作提供多个 *attribute*: *value* 对。

添加属性值

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

以下示例显示如何使用相同的 add LDIF 语法向现有多值属性和尚未存在的属性中添加值：

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
add: cn
cn: Babs Jensen
-
add: mobile
mobile: (408) 555-7844
```

如果存在以下任一情况，则此操作可能会失败，并且服务器将返回错误：

- 属性中已存在给定的值。
- 值与为属性定义的语法不符。
- 条目的对象类不需要或不允许使用该属性类型。
- 属性类型不是多值类型，并且属性中已存在某个值。

使用二进制属性子类型

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

attribute;binary 子类型表示属性值必须作为二进制数据通过 LDAP 进行传输，而不考虑属性值的实际语法。此子类型适用于没有 LDAP 字符串表示的复杂语法，如 userCertificate。不应将二进制子类型用于其他目的。

在 ldapmodify 命令中使用时，可以将相应的子类型添加到任何 LDIF 语句的属性名称中。

要输入二进制值，可以在 LDIF 文本中直接键入或从其他文件读取。以下示例显示了用于从文件读取二进制值的 LDIF 语法：

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
version: 1
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
add: userCertificate;binary
userCertificate;binary:< file:///local/cert-file
```

要使用 :< 语法指定文件名，必须将 version: 1 行作为 LDIF 语句的开头。当 ldapmodify 处理此语句时，它会将属性设置为从给定文件的全部内容中读取的值。

添加具有语言子类型的属性

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

属性的语言和发音子类型可指定本地化的值。为属性指定语言子类型时，该子类型将添加到属性名称中，如下所示：

attribute;lang-*CC*

其中 *attribute* 是现有属性类型，而 *cc* 是用于指定语言的双字母国家/地区代码。您还可以将发音子类型添加到语言子类型中，以便为本地化的值指定一个语音值。在本案例中，属性名称如下所示：

attribute;lang-*CC*;phonetic

要对具有子类型的属性执行操作，必须明确匹配其子类型。例如，如果要修改具有 lang-fr 语言子类型的属性值，则必须在修改操作中包含 lang-fr，如下所示：

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
add: homePostalAddress;lang-fr
homePostalAddress;lang-fr: 34, rue de la Paix
```

注 – 如果属性值包含非 ASCII 字符，则这些字符必须为 UTF-8 编码的字符。

修改属性值

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

以下示例说明如何通过使用 LDIF 中的 replace 语法更改属性值：

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
replace: sn
sn: Morris
-
replace: cn
cn: Barbara Morris
cn: Babs Morris
```

将删除指定属性的所有当前值，并添加所有给定值。

更改属性值之后，您可以使用 ldapsearch 命令验证此更改。

属性值中的结尾空格

修改属性值时，请勿不小心在值的末尾包含空格。结尾空格可能会导致值以 base-64 编码格式显示（如 34xy57eg）。

如果属性值的末尾是空格，则此空格将作为属性值的一部分进行编码。在使用 DSCC 或 ldapsearch 命令验证更改时，所显示的值可能是纯文本，也可能是 base-64 编码的文本。这取决于您所使用的目录服务器客户端。

删除属性值

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

以下示例说明如何完整地删除属性，以及如何仅删除多值属性的一个值：

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
delete: facsimileTelephoneNumber
-
delete: cn
cn: Babs Morris
```

使用 delete 语法时如果不指定 *attribute: value* 对，将删除该属性的所有值。如果指定 *attribute: value* 对，则只会删除该值。

修改多值属性的一个值

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

要使用 ldapmodify 命令修改多值属性的一个值，必须执行以下示例中显示的两个操作：

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
delete: mobile
mobile: (408) 555-7845
-
add: mobile
mobile: (408) 555-5487
```

使用 ldapdelete 删除条目

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

可以使用 `ldapdelete` 命令行实用程序从目录中删除条目。此实用程序将绑定到目录服务器，并根据条目 DN 删除一个或多个条目。必须提供具有删除指定条目的权限的绑定 DN。

您无法删除具有子条目的条目。LDAP 协议不允许出现子条目不再具有父条目的情形。例如，您无法删除组织单位条目，除非先删除了属于该组织单位的所有条目。

以下示例仅显示组织单位的一个条目。可以先后删除此条目及其父条目。

```
$ ldapdelete -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
uid=bjensen,ou=People,dc=example,dc=com
ou=People,dc=example,dc=com
```

使用 ldapmodify 删除条目

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

使用 `ldapmodify` 实用程序时，还可以使用 `changetype: delete` 关键字删除条目。使用 `ldapdelete` 时的所有限制此时同样适用，如上一部分所述。使用 LDIF 语法删除条目的优点在于，您可以在单个 LDIF 文件中执行混合操作。

以下示例执行与上一示例相同的删除操作：

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: delete

dn: ou=People,dc=example,dc=com
changetype: delete
```

使用 ldapsearch 搜索条目

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

可以使用 `ldapsearch` 命令行实用程序查找和检索目录条目。请注意，`ldapsearch` 实用程序不是随 Solaris 平台提供的实用程序，而是 Directory Server Resource Kit 的一部分。

有关使用 `ldapsearch`、常用 `ldapsearch` 选项、可接受的格式以及示例的详细信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》。

▼ 使用 ldapmodify 移动或重命名条目

以下过程将使用修改 DN 操作。在开始此操作之前，请确保您已经熟悉第 91 页中的 “使用修改 DN 操作的准则和限制”部分的内容。

对于此过程的某些部分，可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。此过程的其他部分只能使用命令行完成。

1 **如果要将条目从一个父条目移动到另一个父条目中，请在这些父条目上扩展 ACI 权限。**

- 在要移动的条目的当前父条目上，确保 ACI 允许通过使用语法 `allow (export ...)` 执行**导出**操作
- 在要移动的条目的将来父条目上，确保 ACI 允许通过使用语法 `allow (import ...)` 执行**导入**操作

有关使用 ACI 的信息，请参见第 6 章。

2 **确保全局启用修改 DN 操作，或至少为将受移动操作影响的后缀启用该操作。**

为了确保与以前的目录服务器发行版兼容，默认情况下不启用修改 DN 操作。

如果以前启用了修改 DN 操作，请转至下一步。

要为服务器全局启用修改 DN 操作，请使用以下命令：

```
$ dsconf set-server-prop -h host -p port moddn-enabled:on
```

3 **运行 `ldapmodify` 命令。**

此步骤将使用修改 DN 操作。执行以下任一操作：

- 移动条目。

  例如，以下命令可将条目`uid=bjensen` 从承包商的子树 `ou=Contractors,dc=example,dc=com` 移动到员工的子树 `ou=People,dc=example,dc=com` 中：

  ```
  $ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
  Enter bind password:
  dn: uid=bjensen,ou=Contractors,dc=example,dc=com
  changetype: modrdn
  newrdn: uid=bjensen
  deleteoldrdn: 0
  newsuperior: ou=People,dc=example,dc=com
  ```

- 重命名条目。

  例如，以下命令可将条目 `uid=bbjensen` 重命名为 `uid=bjensen`：

  ```
  $ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
  Enter bind password:
  dn: uid=bbjensen,ou=People,dc=example,dc=com
  changetype: modrdn
  newrdn: uid=bjensen
  deleteoldrdn: 1
  ```

编写 LDIF 语句时请注意以下属性：

- `dn` - 指定要重命名或移动的条目。
- `changetype: modrdn` - 指定要使用的修改 DN 操作。
- `newrdn` - 提供新的命名属性。
- `deleteoldrdn` - 表示是否应该从条目中删除以前的命名属性（`1` 表示是，`0` 表示否）。

  请注意，如果命名属性是条目定义中必需的属性，则无法从条目中删除此属性。
- `newsuperior` - 指定条目的新上级属性。

有关 `ldapmodify` 命令及其选项的信息，请参见 `ldapmodify`(1) 手册页。

4 如果在移动或重命名包含大量条目的子树时遇到资源限制错误，请增加数据库可以使用的锁定数。

```
$ dsconf set-server-prop -h host -p port db-lock-count:value
```

如果修改此属性，则必须重新启动服务器以使更改生效。

使用修改 DN 操作的准则和限制

使用修改 DN 操作时（如上一部分所述），请遵循以下部分介绍的准则。

使用修改 DN 操作的一般准则

- 不要使用修改 DN 操作将条目从一个后缀移动到另一个后缀中，也不要使用该操作重命名或移动根后缀。
- 确保正在运行 Directory Server 5.2 2005Q1 或更高版本。无法在 Directory Server 5.2 2005Q1 之前的目录服务器版本上使用修改 DN 操作。
- 不要在应用程序中使用 `entryid` 操作属性，因为该属性仅供内部使用。移动条目后，条目的 `entryid` 属性可能发生更改。
- 为服务器上的所有后缀全局启用修改 DN 操作，或者为要运行修改 DN 操作的每个后缀单独启用该操作。默认情况下，修改 DN 操作处于禁用状态。
- 在要运行修改 DN 操作的每个后缀上扩展 ACI 权限。**导入**访问权限允许将条目导入指定 DN。**导出**访问权限允许从指定 DN 导出条目。
- 在执行修改 DN 操作之前，请确保该操作不会中断客户端验证。如果移动引用客户端证书的条目，则客户端验证将会中断。请在移动条目后对证书进行验证。
- 在执行修改 DN 操作之前，请确保该操作不会中断应用程序。重命名或移动条目可能会影响多个后缀，或更改条目的以下特性：
  - 条目的已过滤角色的范围。
  - 条目的嵌套角色（此处的嵌套角色包含过滤角色）。
  - 条目的动态组成员身份。

协同使用修改 DN 操作和复制时的准则

注意 – 使用修改 DN 操作时如果不符合以下要求，可能会中断复制并破坏目录服务。

- 确保复制拓扑中的所有服务器都在运行 Directory Server 5.2 2005Q1 或更高版本。无法在 Directory Server 5.2 2005Q1 之前的目录服务器版本上使用修改 DN 操作。
- 在复制拓扑中的所有服务器上启用修改 DN 操作。如果主服务器支持修改 DN 操作而使用方服务器不支持此操作，则复制将会失败。将在提供方服务器的错误日志中写入类似以下内容的消息。

  启用 `MODDN` 时无法启动复制会话 要重新启动复制，请重新配置复制拓扑以便在所有服务器上启用修改 `DN` 操作，然后使用以下任一方式启动复制会话：

  - 按照第 240 页中的 “强制执行复制更新”中的说明进行操作。
  - 更改提供方服务器上的条目。此更改将被复制到使用方服务器。

- 在拓扑中的所有主副本上启用并配置引用完整性插件。此操作可确保服务器维护组和角色的引用完整性。有关如何启用和配置引用完整性插件的信息，请参见第 210 页中的 “配置引用完整性插件”。

  执行完修改 DN 操作之后，请为引用完整性插件留出复制更改的时间。

设置引用

可以使用引用通知客户端应用程序应该联系哪个服务器（如果无法在本地获取此信息）。引用是指向远程后缀或条目的指针，目录服务器将其代替结果返回给客户端。然后，客户端必须在引用中所指定的远程服务器上再次执行操作。

在以下三种情况下将发生重定向：

- 客户端应用程序请求本地服务器上不存在的条目，并且服务器已被配置为返回默认引用。
- 出于维护或安全原因已禁用整个后缀。

  服务器将返回该后缀所定义的引用。第 58 页中的 “设置引用并使后缀变为只读状态”中介绍了后缀级别的引用。客户端请求写入操作时，后缀的只读副本也会将引用返回给主服务器。
- 客户端专门访问智能引用。

  智能引用是您创建的一个条目。服务器将返回智能引用所定义的引用。

在任何情况下，引用都是一个 LDAP URL，其中包含主机名、端口号以及其他服务器上的 DN（可选）。例如 `ldap://east.example.com:389`。

有关如何在目录部署中使用引用的概念性信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Deployment Planning Guide》。

以下部分介绍设置目录默认引用以及创建和定义智能引用的过程。

设置默认引用

客户端应用程序在某个 DN 上提交操作时，如果由目录服务器维护的任何后缀都不包含该 DN，则会将默认引用返回给该客户端应用程序。服务器将返回定义的所有引用，但不会定义返回引用的顺序。

▼ 设置默认引用

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **使用 `dsconf` 命令行实用程序设置一个或多个默认引用。**

  ```
  $ dsconf set-server-prop -h host -p port suffix-DN referral-url:referral-URL
  ```

例如：

```
$ dsconf set-server-prop -h host1 -p 1389 dc=example,dc=com \
 referral-url:ldap://east.example.com:1389
```

设置智能引用

智能引用允许您将目录条目或目录树映射到特定的 LDAP URL。使用智能引用时，可以将客户端应用程序引用到特定服务器或特定服务器上的特定条目。

智能引用通常指向另一个服务器上具有相同 DN 的实际条目。但是，您可以将智能引用定义为指向相同服务器或不同服务器上的任何条目。例如，您可以将具有以下 DN 的条目定义为智能引用：

```
uid=bjensen,ou=People,dc=example,dc=com
```

该智能引用指向服务器 `east.example.com` 上的另一个条目：

```
cn=Babs Jensen,ou=Sales,o=east,dc=example,dc=com
```

目录使用智能引用的方式应符合 RFC 4511 的 4.1.10 部分所指定的标准 (http://www.ietf.org/rfc/rfc4511.txt (`http://www.ietf.org/rfc/rfc4511.txt`))。

▼ 创建和修改智能引用

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1 **要创建智能引用，请使用 `referral` 和 `extensibleObject` 对象类创建条目。**

`referral` 对象类允许使用应包含 LDAP URL 的 `ref` 属性。`extensibleObject` 对象类允许您将任何模式属性用作命名属性，以便与目标条目相匹配。

例如，要定义以下可返回智能引用（而不是条目 `uid=bjensen`）的条目，请使用如下命令：

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
objectclass: top
objectclass: extensibleObject
objectclass: referral
uid: bjensen
ref: ldap://east.example.com/cn=Babs%20Jensen,ou=Sales,o=east,dc=example,dc=com
```

注 – 服务器将忽略 LDAP URL 中位于空格后的所有信息。因此，在用作引用的任何 LDAP URL 中，都必须使用 `%20` 代替空格。其他特殊字符必须进行转义。

定义智能引用之后，对 `uid=bjensen` 条目的修改实际上将在其他服务器的 `cn=Babs Jensen` 条目上执行。`ldapmodify` 命令将自动跟踪引用，例如：

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: replace
replace: telephoneNumber
telephoneNumber: (408) 555-1234
```

2 **（可选的）要修改智能引用条目，请使用 `ldapmodify` 的 `-M` 选项：**

```
$ ldapmodify -M -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: replace
replace: ref
ref: ldap://east.example.com/cn=Babs%20Jensen,ou=Marketing,o=east,dc=example,dc=com
```

检查有效的属性语法

只要执行以下操作，目录服务器都允许您检查属性的完整性：

- 使用 `dsadm import` 或 `dsconf import` 导入数据。
- 使用 LDAP 或 DSML 添加条目、修改条目或修改条目的 DN。

检查可确保属性值符合 IETF 建议。所有不符合的属性都将被拒绝并记录到错误日志中。日志消息包含连接和操作 ID（如果适用）。

默认情况下，服务器将自动检查上述操作的语法。如果要关闭语法检查，请使用以下过程。

注 – 语法检查与模式检查有所不同。有关模式检查的信息，请参见第 251 页中的 “管理模式检查”。

▼ 关闭自动语法检查

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

● **要关闭自动语法检查，请使用以下命令：**

```
$ dsconf set-server-prop -h host -p port check-syntax-enabled:off
```

跟踪对目录条目的修改

默认情况下，服务器会为新创建或已修改的条目维护一些特殊属性，如 LDAP v3 规范中所指定的那样。这些特殊属性存储在后缀中的条目上，其中包括：

- `creatorsName` — 最初创建条目的用户的 DN。
- `createTimestamp` — 创建条目时的时间戳（GMT 格式）。
- `modifiersName` — 最后修改条目的用户的 DN。
- `modifyTimestamp` — 修改条目时的时间戳（GMT 格式）。

▼ 关闭条目修改跟踪

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

注意 – 关闭条目修改跟踪会导致不兼容的数据。由于许多应用程序都依赖于这些属性，并且禁用此功能只会使性能略为提高，因此建议您不要关闭条目修改跟踪。

- **关闭服务器的条目修改跟踪。**

```
$ dsconf set-server-prop -h host -p port suffix-DN mod-tracking-enabled:off
```

加密属性值

属性加密可保护存储在目录中的敏感数据。属性加密允许您指定以加密格式存储条目的某些属性。这可防止读取存储在数据库文件、备份文件和导出的 LDIF 文件中的数据。

使用此功能，将属性值存储到目录服务器数据库之前会对其进行加密，并在返回给客户端之前解密回原始值。您必须使用访问控制来阻止没有权限的客户端访问此类属性，并在客户端和目录服务器之间传输属性值时使用 SSL 对属性值进行加密。有关一般情况下的数据安全性的结构性描述，以及特殊情况下的属性加密的结构性概述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》。

只有在服务器上配置和启用 SSL 之后，属性加密才处于活动状态。但在默认情况下，任何属性都未加密。属性加密是在后缀级别上配置的，这意味着会在后缀包含该属性的每个条目中加密该属性。如果您要在整个目录中加密属性，则必须在每个后缀中为该属性启用加密。

注意 – 属性加密将影响与后缀关联的所有数据和索引文件。如果修改现有后缀的加密配置，则**必须**先导出该后缀的内容，再更改配置，然后重新导入内容。DSCC 可以帮助您执行这些步骤。有关使用 DSCC 的详细信息，请参见第 41 页中的 “目录服务控制中心界面”。

为了更加安全，在为任何属性启用加密时，都应手动删除可能仍包含未加密值的数据库缓存文件和数据库日志文件。第 97 页中的 “配置属性加密”中介绍了删除这些文件的过程。

在新后缀中装入或创建数据之前，应该启用所有加密属性。

如果选择加密被某些条目用作命名属性的属性，则 DN 中显示的值将不会被加密。存储在条目中的值将被加密。

即使可以在配置加密时选择 `userPassword` 属性，也不会实际提高安全性，除非需要以明文形式存储密码。DIGEST-MD5 SASL 验证就是这种情况。如果密码已在密码策略中定义了加密机制，则进一步加密只能使安全性略为提高，且反而会影响每个绑定操作的性能。

在存储时，加密的属性前会加上一个表示所用加密算法的密码标记。使用 DES 加密算法的加密属性将显示为如下形式：

```
{CKM_DES_CBC}3hakc&jla+=snda%
```

当您考虑到数据加密而联机导入数据时，就已经提供了用于通过服务器验证的密钥数据库密码，系统不会再出现此提示。如果要脱机导入数据，目录服务器会先提示您输入密码，然后才允许您加密要导入的数据。解密数据（此操作需要更高的安全性）时，无论联机还是脱机执行导出操作，目录服务器都会自动提示您输入密钥数据库密码。这可进一步提高安全性。

注 – 只要证书或私钥不发生更改，服务器将继续生成相同的密钥。因此，如果两个服务器实例使用相同的证书，则可以将数据从一个服务器实例传输到另一个服务器实例（先导出然后再导入）。

属性加密和性能

虽然属性加密提供了增强的数据安全性，但也会影响系统性能。请仔细考虑哪些属性需要加密，并且只加密您认为特别敏感的那些属性。

由于可以通过索引文件直接访问敏感数据，因此必须加密与加密属性相对应的索引键，以确保属性受到完整保护。如果索引已对目录服务器性能造成影响（尚未包括加密索引键所造成的影响），请在第一次将数据导入或添加到数据库**之前**配置属性加密。此过程可确保对加密属性的索引就像从头开始编制一样。

属性加密使用注意事项

执行属性加密功能时请考虑以下事项：

- 一般而言，在修改属性加密配置时，最佳做法是先导出数据，再更改配置，然后导入新配置的数据。

  这可确保整体考虑所有配置更改，而不会丢失任何功能。否则，某些功能可能会丢失，从而破坏数据的安全性。

- 在现有数据库上修改属性加密配置可能会对系统性能造成严重影响。

  例如，假定您有一个包含现有数据的数据库实例。该数据库包含以前存储的具有 `mySensitiveAttribute` 属性的条目。此属性的值以明文形式存储在数据库和索引文件中。如果您以后决定加密 `mySensitiveAttribute` 属性，则必须导出该数据库实例中的所有数据，然后将其重新导入数据库，以确保服务器使用属性加密配置更新数据库和索引文件。如果一开始就对属性进行加密，则可避免由此造成的性能影响。

- 以解密格式导出数据时，如果使用错误的密码，则导出将被拒绝。

  作为一种安全措施，服务器会在用户以解密格式导出数据时提示用户输入密码。如果用户提供了错误的密码，服务器将拒绝解密导出操作。可以直接输入密码，也可以提供密码所在文件的路径。请注意，此文件与 SSL 密码文件具有相同的语法。请参见第 109 页中的 “配置证书数据库密码”。

- 可以对加密算法进行更改，但如果更改过程有误，则结果可能会丢失索引功能。

  要更改用于加密数据的算法，请导出数据，再修改属性加密配置，然后导入数据。如果不按此过程操作，则根据初始加密算法创建的索引将不再有效。

  由于加密的属性前添加了表示所用加密算法的密码标记，内部服务器操作将负责导入数据。因此，目录服务器允许您在更改算法前以加密格式导出数据。

- 更改服务器的 SSL 证书将导致您无法解密已加密的数据。

  属性加密功能会使用服务器的 SSL 证书生成自己的密钥，以用于执行加密和解密操作。因此，解密已加密的数据时需要使用 SSL 证书。如果更改证书之前未解密数据，将无法解密数据。要避免出现这种情况，请以解密格式导出数据，再更改证书，然后重新导入数据。

- 要以加密格式传输数据，也就是说，要将数据从一个服务器实例导出，然后再导入另一个服务器实例，则这两个服务器实例必须使用相同的证书。

  有关信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 管理指南》中的“加密属性值”。

▼ 配置属性加密

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1. **如果要配置属性加密的后缀中包含任何条目，都必须先将该后缀的内容导出到 LDIF 文件。**

   如果后缀包含加密的属性，并且您计划使用导出的 LDIF 文件重新初始化该后缀，则可以使这些属性在导出的 LDIF 文件中保持加密状态。

2. **要启用属性加密，请使用以下命令：**

   ```
   $ dsconf create-encrypted-attr -h host -p port suffix-DN attr-name cipher-name
   ```

   其中 *cipher-name* 可为以下任一选项：

   - `des` - DES 块密码
   - `des3` - 3DES 块密码
   - `rc2` - RC2 块密码
   - `rc4` - RC4 流密码

   例如：

   ```
   $ dsconf create-encrypted-attr -h host1 -p 1389 dc=example,dc=com uid rc4
   ```

3. **要将加密的属性还原为原始状态，请使用以下命令：**

   ```
   $ dsconf delete-encrypted-attr -h host -p port suffix-DN attr-name
   ```

4. **如果更改了配置以加密一个或多个属性，并且在执行导入操作前这些属性已具有值，请清除数据库缓存并删除日志。**

   在数据库缓存和数据库日志中不会显示任何未加密的值。

   注 – 如果删除这些文件，将丢失某些跟踪信息。此外，在删除这些文件之后，服务器将处于恢复模式，可能需要很长时间才能重新启动。

   清除数据库缓存和删除日志：

   a. **停止目录服务器，如****第 55 页中的“ 启动、停止和重新启动目录服务器实例 ”****中所述。**

   b. **以超级用户或具有管理员权限的用户身份从文件系统中删除数据库缓存文件。**

      ```
      # rm instance-path/db/__db.*
      ```

   c. **从文件系统中删除数据库日志文件。**

      ```
      # rm instance-path/db/log.0000000001
      ```

   d. **重新启动目录服务器。**

      服务器将自动创建新的数据库缓存文件。在重新填充缓存之前，此后缀中的操作性能可能会略受影响。

5 **使用 LDIF 文件初始化后缀，如第 185 页中的“ 初始化后缀 ”中所述。**
装入文件并创建相应索引之后，指定属性的所有值都将被加密。

◆ ◆ ◆ 第 5 章

目录服务器安全性

目录服务器支持多种可通过网络提供安全可靠通信的机制。LDAPS 是在安全套接字层 (Secure Sockets Layer, SSL) 之上运行的标准 LDAP 协议。LDAPS 可加密数据，并可选择使用证书进行验证。本章中提及的术语 SSL 表示受支持的协议 SSL2、SSL3 和 TLS 1.0。

目录服务器还支持启动传输层安全 (Start Transport Layer Security, Start TLS) 扩展操作，以便在最初未加密的 LDAP 连接上启用 TLS。

此外，目录服务器还支持通过简单验证和安全层 (Simple Authentication and Security Layer, SASL) 执行通用安全服务 API (Generic Security Service API, GSSAPI)。GSSAPI 允许您在 Solaris 操作系统上使用 Kerberos V5 安全协议。标识映射机制接着会将 Kerberos 主体与目录中的标识进行关联。

有关其他安全性信息，请参见 NSS Web 站点，网址为 http://www.mozilla.org/projects/security/pki/nss/ (http://www.mozilla.org/projects/security/pki/nss/)。

本章提供了通过 SSL 配置安全性的过程。有关 ACI 的信息，请参见第 6 章。有关用户访问权限和密码的信息，请参见第 7 章。

本章包含以下主题：

在目录服务器中使用 SSL

安全套接字层 (Secure Sockets Layer, SSL) 在目录服务器和客户端之间提供加密通信和可选验证。可以通过 LDAP 使用 SSL，也可以将 SSL 与 DSML-over-HTTP 结合使用。默认情况下通过 LDAP 启用 SSL，但如果使用 DSML-over-HTTP，则可以轻松地启用 SSL。此外，还可以将复制配置为使用 SSL 在服务器之间进行安全通信。

如果在简单验证（绑定 DN 和密码）中使用 SSL，将对服务器所接收和发送的所有数据进行加密。加密可保证保密性和数据完整性。客户端可以选择使用证书，通过简单验证和安全层 (Simple Authentication and Security Layer,SASL) 进行目录服务器验证或第三方安全机制验证。基于证书的验证使用公钥密码学，以防止伪造和模拟客户端或服务器。

目录服务器可以在单独的端口上同时进行 SSL 通信和非 SSL 通信。出于安全考虑，您还可以将所有通信限制为使用 LDAP 安全端口。客户端验证也是可以配置的。可以将客户端验证设置为必需或允许。此设置用于确定您所执行的安全级别。

SSL 可支持 Start TLS 扩展操作，从而为常规 LDAP 连接提供安全性。客户端可以绑定到标准 LDAP 端口，然后使用传输层安全协议保护连接。Start TLS 操作允许客户端具有更大的灵活性，并且有助于简化端口分配。

SSL 提供的加密机制也可用于属性加密。启用 SSL 允许您在后缀上配置属性加密，以便在目录中存储数据时保护数据。有关详细信息，请参见第 95 页中的 “加密属性值”。

为了获取额外的安全性，您可以通过访问控制指令 (Access Control Instruction, ACI) 设置对目录内容的访问控制。ACI 需要特定的验证方法，并可确保只能通过安全通道传输数据。通过设置 ACI，可以弥补使用 SSL 和证书的不足之处。有关详细信息，请参见第 6 章。

默认情况下通过 LDAP 启用 SSL，如果使用 DSML-over-HTTP，则可以轻松地启用 SSL。此外，您可能需要对 SSL 配置的某些方面进行修改，如以下部分所述。

管理证书

本部分介绍如何在目录服务器中管理 SSL 证书。

要在目录服务器上运行 SSL，您必须使用自签名证书或公钥基础结构 (Public Key Infrastructure, PKI) 解决方案。

PKI 解决方案需要使用外部证书颁发机构 (Certificate Authority, CA)。要使用 PKI 解决方案，您需要包含公钥和私钥的 CA 签名服务器证书。此证书特定于一个目录服务器。此外还需要一个包含公钥的可信 CA 证书。可信 CA 证书可确保来自 CA 的所有服务器证书都是可信的。此证书有时称为 CA 根密钥或根证书。

注 – 如果将证书用于测试目的，您可能要使用自签名的证书。但是在生产中，使用自签名证书不太安全。在生产中，应使用可信的证书颁发机构 (Certificate Authority, CA) 证书。

本部分中的过程使用 `dsadm` 和 `dsconf` 命令。有关这些命令的信息，请参见 `dsadm`(1M) 和 `dsconf`(1M) 手册页。

本部分提供了以下有关在目录服务器上配置证书的信息：

▼ 查看默认的自签名证书

首次创建目录服务器实例时，它将包含默认的自签名证书。**自签名证书**是一个公钥/私钥对，其中的公钥由私钥进行签名。自签名证书的有效期为三个月。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **要查看默认的自签名证书，请使用以下命令：**

```
$ dsadm show-cert instance-path defaultCert
```

▼ 管理自签名证书

创建目录服务器实例时，将自动提供默认的自签名证书。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1. **要使用非默认设置创建自签名证书，请使用以下命令：**

```
$ dsadm add-selfsign-cert instance-path cert-alias
```

其中 *cert-alias* 是您提供的用于标识证书的名称。

要查看此命令的所有选项，请参见 `dsadm`(1M) 手册页或命令行帮助：

```
$ dsadm add-selfsign-cert --help
```

2 **请在自签名证书过期时续订该证书：**

```
$ dsadm renew-selfsign-cert instance-path cert-alias
```

▼ 请求 CA 签名的服务器证书

此过程介绍如何请求和安装用于目录服务器的 CA 签名服务器证书。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 **生成 CA 签名的服务器证书请求。**

```
$ dsadm request-cert [-W cert-pwd-file] {-S DN | --name name [--org org] [--org-unit org-unit \
  [--city city] [--state state] [--country country]} [-o output-file] [-F format] instance-path
```

例如，要为 Example 公司请求 CA 签名的服务器证书，请使用以下命令：

```
$ dsadm request-cert --name host1 --org Example --org-unit Marketing \
 -o my_cert_request_file /local/ds
```

为了完整地标识服务器，证书颁发机构可能需要此示例中显示的所有属性。有关每个属性的描述，请参见 dsadm(1M) 手册页。

使用 `dsadm request-cert` 请求证书时，所生成的证书请求为二进制证书请求，除非将 ASCII 指定为输出格式。如果指定 ASCII，则生成的证书请求为 PEM 格式的 PKCS #10 证书请求。PEM 是由 RFC 1421 至 1424 (`http://www.ietf.org/rfc/rfc1421.txt`) 指定的保密性增强的电子邮件格式，用于以 US-ASCII 字符表示 base64 编码的证书请求。请求的内容与以下示例类似：

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBrjCCARcCAQAwbjELMAkGA1UBhMCVXMxEzARBgNVBAgTCkNBElGT1JOSUExLD
AqBgVBAoTI25ldHNjYXBlIGNvb11bmljYXRpb25zIGNvcnBvcmF0aWuMRwwGgYDV
QQDExNtZWxsb24umV0c2NhcGUuY29tMIGfMA0GCSqGSIb3DQEBAUAA4GNADCBiQK
BgCwAbskGh6SKYOgHy+UCSLnm3ok3X3u83Us7u0EfgSLR0f+K41eNqqWRftGR83e
mqPLDOf0ZLTLjVGJaHJn4l1gG+JDf/n/zMyahxtV7+T8GOFFigFfuxJaxMjr2j7I
vELlxQ4IfZgwqCm4qQecv3G+N9YdbjveMVXW0v4XwIDAQABAAwDQYJKoZIhvcNAQ
EEBQADgYEAZyZAm8UmP9PQYwNy4Pmypk79t2nvzKbwKVb97G+MT/gw1pLRsuBoKi
nMfLgKp1Q38K5Py2VGW1E47/rhm3yVQrIiwV+Z8Lcc=
-----END NEW CERTIFICATE REQUEST-----
```

2 **按照程序将证书请求传送给证书颁发机构。**

获取证书颁发机构证书的过程取决于所使用的证书颁发机构。某些商业 CA 提供了允许您自动下载证书的 Web 站点。其他 CA 将在您请求证书后以电子邮件形式向您发送证书。

发送请求之后，您必须等待 CA 对请求做出响应，即提供您的证书。请求的响应时间会有所不同。例如，如果您的 CA 在公司内部，则 CA 可能只需一两天即可响应您的请求。如果您选择的 CA 在公司外部，则 CA 可能需要几个星期才能响应您的请求。

3 **保存从证书颁发机构收到的证书。**

请将证书备份到安全位置。如果丢失证书，您可以使用备份文件重新安装。可以在文本文件中保存证书。PEM 格式的 PKCS #11 证书与以下示例类似：

```
-----BEGIN CERTIFICATE-----
MIICjCCAZugAwIBAgICCEEwDQYJKoZIhKqvcNAQFBQAwfDELMAkGA1UEBhMCVVMx
IzAhBgNVBAoGlBhbG9a2FWaWxsZGwSBXaWRnZXRzLCBJbmMuMR0wGwYDVQQLExRX
aWRnZXQgTW3FrZXJzICdSJyBVczEpMCcGAx1UEAxgVGVzdCBUXN0IFRlc3QgVGVz
dCBUZXN0IFlc3QgQ0EswHhcNOTgwMzEyMDIzMzUWhcNOTgwMzI2MDIzMpzU3WjBP
MQswCYDDVQQGEwJVUzEoMCYGA1UEChMfTmV0c2NhcGUgRGlyZN0b3J5VIFB1Ymxp
Y2F0aW9uczEWMB4QGA1UEAxMNZHVgh49dq2tLNvbjTBaMA0GCSqGSIb3DQEBAQUA
A0kAMEYkCQCksMR/aLGdfp4m0OiGgijG5KgOsyRNvwGYW7kfW+8mmijDtZaRjYNj
jcgpF3Vnlbxbclx9LVjjNLC5737XZdAgEDozYwpNDARBglghkgBhvhCEAQEEBAMC
APAwHkwYDVR0jBBgwFAU67URjwCaGqZHUpSpdLxlzwJKiMwDQYJKoZIhQvcNAQEF
BQADgYEAJ+BfVem3vBOPBveNdLGfjlb9hucgmaMcQa9FA/db8qimKT/ue9UGOJqL
bwbMKBBopsDn56p2yV3PLIsBgrcuSoBCuFFnxBnqSiTS7YiYgCWqWaUA0ExJFmD6
6hBLseqkSWulk+hXHN7L/NrViO+7zNtKcaZLlFPf7d7j2MgX4Bo=
-----END CERTIFICATE-----
```

▼ 添加 CA 签名的服务器证书和可信的 CA 证书

此过程介绍如何安装用于目录服务器的 CA 签名服务器证书和可信 CA 证书。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1 **添加 CA 签名的服务器证书。**

```
$ dsadm add-cert instance-path cert-alias cert-file
```

其中 *cert-alias* 是您提供的用于标识证书的名称，*cert-file* 是包含 PEM 格式 PKCS #11 证书的文本文件。

例如，要安装 CA 签名的服务器证书，可以使用与以下形式类似的命令：

```
$ dsadm add-cert /local/ds server-cert /local/safeplace/serv-cert-file
```

证书现在已完成安装，但还不是可信证书。要信任 CA 签名的服务器证书，必须安装证书颁发机构证书。

2 **添加可信的证书颁发机构证书。**

```
$ dsadm add-cert -C instance-path cert-alias cert-file
```

`-C` 选项表示该证书为可信的证书颁发机构证书。

例如，要安装来自证书颁发机构的可信证书，可以使用以下命令：

```
$ dsadm add-cert -C /local/ds CA-cert /local/safeplace/ca-cert-file
```

3 （可选的）验证已安装的证书。

- 要列出所有服务器证书并显示其有效日期和别名，请键入：

  ```
  $ dsadm list-certs instance-path
  ```

  例如：

  ```
  $ dsadm list-certs /local/ds1
  Enter the certificate database password:
  Alias       Valid from Expires on Self-   Issued by          Issued to
                                    signed?
  ----------- ---------- ---------- ------- -----------------  -----------------
  serverCert  2000/11/10 2011/02/10 n       CN=CA-Signed Cert, CN=Test Cert,
              18:13      18:13              OU=CA,O=com        dc=example,dc=com
  defaultCert 2006/05/18 2006/08/18 y       CN=host1,CN=DS,    Same as issuer
              16:28      16:28              dc=example,dc=com
  2 certificates found
  ```

  默认情况下，目录代理服务器实例包含一个名为 `defaultCert` 的默认服务器证书。文本 `Same as issuer` 表明默认证书是自签名的服务器证书。

- 要列出可信的 CA 证书，请键入：

  ```
  $ dsadm list-certs -C instance-path
  ```

  例如：

  ```
  $ dsadm list-certs -C /local/ds1
  Enter the certificate database password:
  Alias   Valid from Expires on Self-   Issued by          Issued to
                                signed?
  ------- ---------- ---------- ------- -----------------  -----------------
  CA-cert 2000/11/10 2011/02/10 y       CN=Trusted CA Cert, Same as issuer
          18:12      18:12              OU=CA,O=com
  1 certificate found
  ```

- 要查看证书的详细信息（包括证书的过期日期），请键入：

  ```
  $ dsadm show-cert instance-path cert-alias
  ```

  例如，要查看服务器证书，请键入：

  ```
  $ dsadm show-cert /local/ds1 "Server-Cert"
  Enter the certificate database password:
  Certificate:
  ```

```
Data:
    Version: 3 (0x2)
    Serial Number: 2 (0x2)
    Signature Algorithm: PKCS #1 MD5 With RSA Encryption
    Issuer:
        "CN=Server-Cert,O=Sun,C=US"
    Validity:
        Not Before: Fri Nov 10 18:12:20 2000
        Not After : Thu Feb 10 18:12:20 2011
    Subject:
        "CN=CA Server Cert,OU=ICNC,O=Sun,C=FR"
    Subject Public Key Info:
        Public Key Algorithm: PKCS #1 RSA Encryption
        RSA Public Key:
            Modulus:
                bd:76:fc:29:ca:06:45:df:cd:1b:f1:ce:bb:cc:3a:f7:
                77:63:5a:82:69:56:5f:3d:3a:1c:02:98:72:44:36:e4:
                68:8c:22:2b:f0:a2:cb:15:7a:c4:c6:44:0d:97:2d:13:
                b7:e3:bf:4e:be:b5:6a:df:ce:c4:c3:a4:8a:1d:fa:cf:
                99:dc:4a:17:61:e0:37:2b:7f:90:cb:31:02:97:e4:30:
                93:5d:91:f7:ef:b0:5a:c7:d4:de:d8:0e:b8:06:06:23:
                ed:5f:33:f3:f8:7e:09:c5:de:a5:32:2a:1b:6a:75:c5:
                0b:e3:a5:f2:7a:df:3e:3d:93:bf:ca:1f:d9:8d:24:ed
            Exponent: 65537 (0x10001)
Signature Algorithm: PKCS #1 MD5 With RSA Encryption
Signature:
    85:92:42:1e:e3:04:4d:e5:a8:79:12:7d:72:c0:bf:45:
    ea:c8:f8:af:f5:95:f0:f5:83:23:15:0b:02:73:82:24:
    3d:de:1e:95:04:fb:b5:08:17:04:1c:9d:9c:9b:bd:c7:
    e6:57:6c:64:38:8b:df:a2:67:f0:39:f9:70:e9:07:1f:
    33:48:ea:2c:18:1d:f0:30:d8:ca:e1:29:ec:be:a3:43:
    6f:df:03:d5:43:94:8f:ec:ea:9a:02:82:99:5a:54:c9:
    e4:1f:8c:ae:e2:e8:3d:50:20:46:e2:c8:44:a6:32:4e:
    51:48:15:d6:44:8c:e6:d2:0d:5f:77:9b:62:80:1e:30
Fingerprint (MD5):
    D9:FB:74:9F:C3:EC:5A:89:8F:2C:37:47:2F:1B:D8:8F
Fingerprint (SHA1):
    2E:CA:B8:BE:B6:A0:8C:84:0D:62:57:85:C6:73:14:DE:67:4E:09:56

Certificate Trust Flags:
    SSL Flags:
        Valid CA
        Trusted CA
        User
        Trusted Client CA
    Email Flags:
        User
    Object Signing Flags:
        User
```

▼ 续订已过期的 CA 签名服务器证书

CA 签名服务器证书（公钥和私钥）过期时，可以使用此过程续订证书。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1. **从证书颁发机构获取已更新的 CA 签名服务器证书。**
2. **收到已更新的证书之后，安装该证书。**

   ```
   $ dsadm renew-cert instance-path cert-alias cert-file
   ```

▼ 导出和导入 CA 签名的服务器证书

在某些情况下，您可能需要导出证书的公钥和私钥，以便日后可以导入该证书。例如，您可能希望其他服务器使用该证书。

此过程中的命令可用于包含通配符的证书（如 `"cn=*,o=example"`）。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1. **导出证书。**

   ```
   $ dsadm export-cert [-o output-file] instance-path cert-alias
   ```

   例如：

   ```
   $ dsadm export-cert -o /tmp/first-certificate /local/ds1 "First Certificate"
   $ dsadm export-cert -o /tmp/first-ca-server-certificate /local/ds1/ defaultCert
   Choose the PKCS#12 file password:
   Confirm the PKCS#12 file password:
   $ ls /tmp
   first-ca-server-certificate
   ```

2. **导入证书。**

   ```
   $ dsadm import-cert instance-path cert-file
   ```

   例如，将证书导入 `host1` 上的服务器实例：

   ```
   $ dsadm import-cert -h host1 /local/ds2 /tmp/first-ca-server-certificate
   Enter the PKCS#12 file password:
   ```

3. **（可选的）如果已将证书导入服务器，请将该服务器配置为使用导入的证书。**

   ```
   $ dsconf set-server-prop -e -h host -p port -w - ssl-rsa-cert-name:server-cert
   ```

配置证书数据库密码

默认情况下，目录服务器通过存储的密码在内部管理 SSL 证书数据库密码。在管理证书时，用户无需键入证书密码或指定密码文件。此选项不太安全，因为只是隐藏了密码，而不会对密码进行加密。

但是，如果要对证书使用进行更多控制，则可以配置服务器，以提示用户在命令行中输入密码。在这种情况下，对于除 `autostart`、 `backup`、 `disable-service`、 `enable-service`、`info`、 `reindex`、 `restore` 和 `stop` 之外的所有 `dsadm` 子命令，用户都必须键入证书数据库密码。证书数据库位于目录 *instance-path*`/alias` 中。

▼ 将服务器配置为提示用户输入证书密码

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1. **停止服务器。**

   ```
   $ dsadm stop instance-path
   ```

2. **将密码提示标志设置为 `on`。**

   ```
   $ dsadm set-flags instance-path cert-pwd-prompt=on
   ```

   系统将要求您选择新的证书密码。

3. **启动服务器。**

   ```
   $ dsadm start instance-path
   ```

备份和恢复目录服务器的证书数据库

备份目录服务器实例时，将会备份目录服务器配置和证书。备份的证书存储在 *archive-path*`/alias` 目录中。

有关如何备份和恢复目录服务器的信息，请参见第 190 页中的 “创建备份以用于灾难恢复”。

配置 SSL 通信

本部分包含与禁用和启用 SSL 有关的过程。

禁用非安全通信

创建服务器实例时，默认情况下将创建 LDAP 端口和安全 LDAP 端口 (LDAPS)。但是，在某些情况下可能需要禁用非 SSL 通信，以便服务器只通过 SSL 进行通信。

SSL 连接是使用默认自签名证书启用的。如果需要，您可以安装自己的证书。有关在启动服务器后管理证书和禁用 SSL 的说明，请参见第 5 章。有关证书、证书数据库以及获取 CA 签名服务器证书的概述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》。

▼ 禁用 LDAP 端口

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1. **禁用 LDAP 端口。**

   要禁用非安全点，您必须绑定到 LDAP 安全端口。此示例显示了与主机服务器 `host1` 上的默认 LDAP 安全端口 `1636` 的绑定。

   ```
   $ dsconf set-server-prop -h host1 -p 1636 ldap-port:disabled
   ```

2. **重新启动服务器以使更改生效。**

   ```
   $ dsadm restart /local/ds
   ```

   现在，您已不再绑定到非安全端口 1389。

选择加密密码

密码是用于加密和解密数据的算法。一般而言，密码在加密期间所使用的位数越多，加密就**越严密**或越安全。SSL 的密码也由所使用的消息验证类型进行标识。消息验证是计算**校验和**（用于保证数据完整性）的另一种算法。

当客户端初始化与服务器的 SSL 连接时，客户端和服务器必须就用于加密信息的密码达成一致。在任何双向加密过程中，双方都必须使用相同的密码。所使用的密码取决于服务器所保存的密码列表的当前顺序。服务器将选择客户端所提供的与列表中的密码相匹配的第一个密码。目录服务器的默认密码值为 `all`，表示基础 SSL 库支持的所有已知的安全密码。但是，您可以修改此值以便只接受特定密码。

有关可用于目录服务器的密码的详细信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》。

▼ 选择加密密码

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1. **确保已为服务器启用 SSL。**

   请参见第 109 页中的 “配置 SSL 通信”。

2 **查看可用的 SSL 密码。**

```
$ dsconf get-server-prop -h host -p port ssl-supported-ciphers
ssl-supported-ciphers  :  TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
ssl-supported-ciphers  :  TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
ssl-supported-ciphers  :  TLS_DHE_RSA_WITH_AES_256_CBC_SHA
ssl-supported-ciphers  :  TLS_DHE_DSS_WITH_AES_256_CBC_SHA
...
```

3 **（可选的）如果要保留非加密数据的副本，请在设置 SSL 密码之前导出该数据。**

请参见第 182 页中的 “导出到 LDIF” 。

4 **设置 SSL 密码。**

```
$ dsconf set-server-prop -h host -p port ssl-cipher-family:cipher
```

例如，要将密码系列设置为 SSL_RSA_WITH_RC4_128_MD5 和 SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA，请键入：

```
$ dsconf set-server-prop -h host1 -p 1636 ssl-cipher-family:SSL_RSA_WITH_RC4_128_MD5 \
 ssl-cipher-family:SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
Enter "cn=Directory Manager" password:
Before setting SSL configuration, export Directory Server data.
Do you want to continue [y/n] ? y
Directory Server must be restarted for changes to take effect.
```

5 **重新启动服务器以使更改生效。**

```
$ dsadm restart /local/ds
```

配置客户端验证

客户端验证是服务器用于验证客户端标识的一种机制。

可使用以下任一方式执行客户端验证：

- 通过提供 DN 和密码。
- 通过客户端提供的证书。

  基于证书的验证使用通过 SSL 协议获取的客户端证书来查找用户的标识条目。在基于证书的验证中，客户端将发送用于指定外部机制的 SASL 绑定请求。绑定请求依赖于已建立的 SSL 验证机制。
- 通过基于 SASL 的机制。
  - 在所有操作系统上，通过 DIGEST-MD5 进行 SASL 验证。
  - 在 Solaris 操作系统上，通过 GSSAPI 机制（允许通过 Kerberos V5 进行客户端验证）进行 SASL 验证。

使用上述两种 SASL 机制中的任何一种时，还必须将服务器配置为执行标识映射。SASL 凭证称为**主体**。每种机制都必须具有特定的映射，以便通过主体的内容来确定绑定 DN。当主体映射到单个用户条目，并且 SASL 机制确认该用户的标识有效时，该用户的 DN 即为连接的绑定 DN。

- 通过 SSL 客户端验证模式。

  如果希望所有客户端都在 SSL 层上获得授权，请使用 SSL 客户端验证。客户端应用程序通过将其 SSL 证书发送到服务器来进行验证。您可以使用 `SSL-client-auth-mode` 标志指定服务器允许、请求还是禁止 SSL 客户端验证。默认情况下，服务器允许客户端进行验证。

本部分提供了以下有关在目录服务器上配置两种 SASL 机制的信息。

- 第 112 页中的 “在目录服务器中设置 SASL 加密级别”
- 第 113 页中的 “通过 DIGEST-MD5 进行 SASL 验证”
- 第 116 页中的 “通过 GSSAPI 进行 SASL 验证（仅适用于 SPARC）”

有关配置安全性的详细信息，请参见第 118 页中的 “将 LDAP 客户端配置为使用安全性”。

在目录服务器中设置 SASL 加密级别

在配置 SASL 机制之前，必须指定是否需要加密。SASL 加密的要求由强度安全系数 (Strength Security Factor, SSF) 的最大值和最小值进行设置。

属性 `dsSaslMinSSF`(5dsat) 和 `dsSaslMaxSSF`(5dsat) 表示加密密钥的长度，这些属性存储在 `cn=SASL, cn=security, cn=config` 中。

服务器允许任何级别的加密，包括不加密。这意味着目录服务器接受大于 256 的 `dsSaslMinSSF` 和 `dsSaslMaxSSF` 值。但目前没有任何 SASL 机制支持大于 128 的 SSF。目录服务器会对这些值进行调整，使其不高于 SSF 可用的最大值 (128)。因此，实际的最大 SSF 可能低于配置的最大值，这取决于可用的基础机制。

SASL 安全系数验证依赖于以下两个主要因素：服务器和客户端应用程序所请求的最小系数和最大系数，以及基础安全组件提供的可用加密机制。概括来说，服务器和客户端将尝试使用最大的可用安全系数，该系数小于或等于两者设置的最大系数，但大于或等于两者设置的最小系数。

目录服务器的默认最小 SASL 安全系数 `dsSaslMinSSF` 为 `0`，表示没有任何保护。实际的最小值取决于客户端设置，除非您更改目录服务器的最小值。实际上，应该将最小值设置为实际希望服务器和客户端使用的最低级别。如果服务器和客户端无法协商出符合最低要求的机制，则不会建立连接。

▼ 要求 SASL 加密

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

- **如果要求 SASL 加密，请将 dsSaslMinSSF 值设置为所需的最小加密。**

```
$ ldapmodify -h host -p port -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: cn=SASL, cn=security, cn=config
changetype: modify
replace: dsSaslMinSSF
dsSaslMinSSF: 128
^D
```

▼ 不允许 SASL 加密

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

- **如果不允许 SASL 加密，请将 dsSaslMinSSF 和 dsSaslMaxSSF 的值都设置为零。**

```
$ ldapmodify -h host -p port -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: cn=SASL, cn=security, cn=config
changetype: modify
replace: dsSaslMinSSF
dsSaslMinSSF: 0

replace: dsSaslMaxSSF
dsSaslMaxSSF: 0
^D
```

通过 DIGEST-MD5 进行 SASL 验证

DIGEST-MD5 机制通过比较客户端发送的散列值与用户密码的散列值来验证客户端。但是，由于此机制必须读取用户密码，因此要通过 DIGEST-MD5 进行验证的所有用户在目录中都必须具有 {CLEAR} 密码。在目录中存储 {CLEAR} 密码时，必须确保通过 ACI 正确限制对密码值的访问权限，如第 6 章中所述。此外，还需要在后缀中配置属性加密，如第 95 页中的 “加密属性值”中所述。

▼ 配置 DIGEST-MD5 机制

以下过程介绍如何将目录服务器配置为使用 DIGEST-MD5。

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1. **使用 ldapsearch 命令验证 DIGEST-MD5 是否为根条目上的 supportedSASLMechanisms 属性值。**

   例如，以下命令显示启用了哪些 SASL 机制：

   ```
   $ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
    -s base -b "" "(objectclass=*)" supportedSASLMechanisms
   Enter bind password:
   dn:
   supportedSASLMechanisms: EXTERNAL
   supportedSASLMechanisms: DIGEST-MD5
   supportedSASLMechanisms: GSSAPI
   ^D
   ```

2. **如果未启用 DIGEST-MD5，请将其启用。**

   ```
   $ ldapmodify -h host -p port -D cn=admin,cn=Administrators,cn=config -w -
   Enter bind password:
   dn: cn=SASL, cn=security, cn=config
   changetype: modify
   add: dsSaslPluginsEnable
   dsSaslPluginsEnable: DIGEST-MD5
   -
   replace: dsSaslPluginsPath
   dsSaslPluginsPath: SASL-library
   ^D
   ```

   其中 *SASL-library* 为以下任一选项：

   JES 安装　　`/usr/lib/mps/sasl2`

   Zip 安装　　*install-path*`/dsee6/private/lib`

3. **为 DIGEST-MD5 使用默认标识映射，或创建新的映射。**

   有关信息，请参见第 114 页中的 “DIGEST-MD5 标识映射”。

4. **对于将使用 DIGEST-MD5 通过 SSL 访问服务器的所有用户，确保以 {CLEAR} 形式存储密码。**

   有关密码存储模式的信息，请参见第 7 章。

5. **如果修改了 SASL 配置条目或某个 DIGEST-MD5 标识映射条目，请重新启动目录服务器。**

DIGEST-MD5 标识映射

SASL 机制的标识映射尝试将 SASL 标识的凭证与目录中的用户条目进行匹配。如果映射找不到与 SASL 标识相对应的 DN，则验证将会失败。有关此机制的完整描述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》。

SASL 标识是称为**主体**的字符串，以特定于每种机制的格式表示用户。在 DIGEST-MD5 中，客户端应创建包含 `dn:` 前缀和 LDAP DN 的主体，或者创建包含 `u:` 前缀（后跟由客户端确定的任何文本）的主体。在映射期间，客户端发送的主体可用于 `${Principal}` 占位符中。

服务器配置中的以下条目是 DIGEST-MD5 的默认标识映射：

```
dn: cn=default,cn=DIGEST-MD5,cn=identity mapping,cn=config
objectClass: top
objectClass: nsContainer
objectClass: dsIdentityMapping
objectClass: dsPatternMatching
cn: default
dsMatching-pattern: \${Principal}
dsMatching-regexp: dn:(.*)
dsMappedDN: \$1
```

此标识映射假定主体的 `dn` 字段包含目录中现有用户的精确 DN。

▼ 定义您自己的 DIGEST-MD5 标识映射

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1. **编辑默认的映射条目，或者在 `cn=DIGEST-MD5,cn=identity mapping,cn=config` 下创建新的映射条目。**

   DIGEST-MD5 的示例映射位于 *instance-path*`/ldif/identityMapping_Examples.ldif` 中。

   此示例假定主体的非限定文本字段中包含所需标识的用户名。以下命令显示应如何定义此映射：

   ```
   $ ldapmodify -a -h host1 -p 1636 -D cn=admin,cn=Administrators,cn=config -w -
   Enter bind password:
   dn: cn=unqualified-username,cn=DIGEST-MD5,cn=identity mapping
   cn=config
   objectclass: dsIdentityMapping
   objectclass: dsPatternMatching
   objectclass: nsContainer
   objectclass: top
   cn: unqualified-username
   dsMatching-pattern: \${Principal}
   dsMatching-regexp: u:(.*)@(.*)\\.com
   dsSearchBaseDN: dc=\$2
   dsSearchFilter: (uid=\$1)
   ```

2. **重新启动目录服务器以使新映射生效。**

通过 GSSAPI 进行 SASL 验证（仅适用于 SPARC）

通过 SASL 执行的通用安全服务 API (Generic Security Service API , GSSAPI) 允许您使用第三方安全系统（如 Kerberos V5）对客户端进行验证。GSSAPI 库仅适用于 Solaris 操作系统 SPARC® 平台。Sun 建议您在 Sun Enterprise Authentication Mechanism™ 1.0.1 服务器上安装 Kerberos V5 实现。

服务器使用 GSSAPI 验证用户的标识。然后，SASL 机制将应用 GSSAPI 映射规则获取 DN，该 DN 为此连接期间所有操作的绑定 DN。

▼ 配置 Kerberos 系统

可以按照制造商的说明来配置 Kerberos 软件。如果您使用的是 Sun Enterprise Authentication Mechanism 1.0.1 服务器，请使用此过程。

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1. **对 `/etc/krb5` 中的文件进行配置。**
2. **创建 Kerberos 数据库以存储用户和服务。**
3. **在该数据库中，创建 LDAP 服务的主体。**

   ```
   $ ldap/server-FQDN@realm
   ```

   其中 *server-FQDN* 是目录服务器的全限定域名。
4. **启动 Kerberos 守护进程。**

注 – 必须在主机上配置 DNS。

有关上述每个步骤的详细说明，请参见软件文档。此外，请参见第 121 页中的 “使用 GSSAPI 和 SASL 进行 Kerberos 验证的示例配置”。

▼ 配置 GSSAPI 机制

以下过程介绍如何在 Solaris 操作系统上将目录服务器配置为使用 GSSAPI：

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1. **创建 GSSAPI 的默认标识映射以及任何自定义映射，如第 117 页中的 “GSSAPI 标识映射”中所述。**

2. **创建用于存储服务密钥的密钥表。**
   您的 LDAP 服务密钥存储在密钥表中。

   a. **确保只有目录服务器用户可以读取此密钥表。**

   b. **更改文件名，使其不同于默认的** `/etc/krb5/krb5.keytab`。

   c. **设置环境变量** `KRB5_KTNAME` **以确保使用新的密钥表，而不使用默认密钥表。**

3. **如果修改了 SASL 配置条目或某个 GSSAPI 标识映射条目，请重新启动目录服务器。**
   请注意，必须在主机上配置 DNS。

GSSAPI 标识映射

SASL 机制的标识映射尝试将 SASL 标识的凭证与目录中的用户条目进行匹配。如果映射找不到与 SASL 标识相对应的 DN，则验证将会失败。

SASL 标识是称为**主体**的字符串，以特定于每种机制的格式表示用户。在使用 GSSAPI 的 Kerberos 中，主体为 *uid* [/*instance*] [@*realm*] 格式的标识。*uid* 可以包含后跟**领域**（通常为域名）的**实例**标识符，实例标识符和领域都是可选的。例如，以下字符串都是有效的用户主体：

```
bjensen
bjensen/Sales
bjensen@EXAMPLE.COM
bjensen/Sales@EXAMPLE.COM
```

最初，在目录中未定义任何 GSSAPI 映射。可以根据客户端定义所用主体的方式，定义默认映射以及所需的任何自定义映射。

▼ 定义 GSSAPI 的标识映射

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1. **在** `cn=GSSAPI,cn=identity mapping, cn=config` **下创建新的映射条目。**
   有关标识映射条目中的属性定义，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》。GSSAPI 映射的示例位于 *instance-path*`/ldif/identityMapping_Examples.ldif` 中。

   此文件中的默认 GSSAPI 映射假定主体只包含一个用户 ID。此映射可确定目录固定分支中的某个用户：

   ```
   dn: cn=default,cn=GSSAPI,cn=identity mapping,cn=config
   objectclass: dsIdentityMapping
   objectclass: nsContainer
   ```

```
objectclass: top
cn: default
dsMappedDN: uid=\${Principal},ou=people,dc=example,dc=com
```

此文件中的另一个示例说明当用户 ID 包含在具有已知领域的主体中时如何确定用户 ID。

```
dn: cn=same_realm,cn=GSSAPI,cn=identity mapping,cn=config
objectclass: dsIdentityMapping
objectclass: dsPatternMatching
objectclass: nsContainer
objectclass: top
cn: same_realm
dsMatching-pattern: \${Principal}
dsMatching-regexp: (.*)@EXAMPLE.COM
dsMappedDN: uid=\$1,ou=people,dc=EXAMPLE,dc=COM
```

2 **重新启动目录服务器以使新映射生效。**

将 LDAP 客户端配置为使用安全性

以下部分介绍如何在要与目录服务器建立安全连接的 LDAP 客户端中配置和使用 SSL。在 SSL 连接中，服务器会将其证书发送到客户端。客户端必须先通过信任服务器证书来验证该服务器。然后，客户端可以选择初始化一种客户端验证机制，方法是为两种 SASL 机制中的某种机制发送自身的证书或信息。SASL 机制包括 DIGEST-MD5 和使用 Kerberos V5 的 GSSAPI。

以下部分使用 `ldapsearch` 工具作为启用了 SSL 的 LDAP 客户端示例。随目录服务器提供的 `ldapmodify`、`ldapdelete` 和 `ldapcompare` 工具的配置方式相同。这些目录访问工具以 Directory SDK for C 为基础，在《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中对这些工具进行了详细介绍。

要在其他 LDAP 客户端上配置 SSL 连接，请参见随应用程序提供的文档。

注 – 某些客户端应用程序实现 SSL，但不验证服务器是否具有可信证书。这些客户端应用程序使用 SSL 协议提供数据加密，但无法保证保密性，也不能防止身份模拟。

以下部分介绍如何配置 LDAP 客户端以使用安全性：

在客户端中使用 SASL DIGEST-MD5

在客户端中使用 DIGEST-MD5 机制时，您不必安装用户证书。但是，如果要使用加密的 SSL 连接，则仍须信任服务器证书，如第 102 页中的 “管理证书”中所述。

指定领域

领域定义用于从中选择验证标识的名称空间。在 DIGEST-MD5 验证中，您必须通过特定领域的验证。

目录服务器使用计算机的全限定主机名作为 DIGEST-MD5 的默认领域。服务器使用 `nsslapd-localhost` 配置属性中包含的主机名的小写值。

如果未指定领域，将使用服务器提供的默认领域。

指定环境变量

在 UNIX 环境中，必须设置 `SASL-PATH` 环境变量，以便 LDAP 工具可以找到 DIGEST-MD5 库。DIGEST-MD5 库是由 SASL 插件动态装入的共享库。请按如下方式设置 `SASL_PATH` 环境变量：

```
export SASL_PATH=SASL-library
```

此路径假定目录服务器安装在调用 LDAP 工具的相同主机上。

`ldapsearch` 命令的示例

可以在不使用 SSL 的情况下执行 DIGEST-MD5 客户端验证。以下示例使用默认的 DIGEST-MD5 标识映射来确定绑定 DN：

```
$ ldapsearch -h host1 -p 1389 \
 -o mech=DIGEST-MD5 [ \
 -o realm="example.com"] \
 -o authid="dn:uid=bjensen,dc=example,dc=com" \
 -w - \
 -o authzid="dn:uid=bjensen,dc=example,dc=com" \
 -o secProp="minssf=56,maxssf=256,noplain" \
 -b "dc=example,dc=com" "(givenname=Richard)"
```

上述示例说明如何使用 `-o`（小写字母 o）选项指定 SASL 选项。**领域**是可选的，但如果指定，则必须是服务器主机的全限定域名。`authid` 和 `authzid` 必须同时存在且完全相同，但不会使用适用于代理操作的 `authzid`。 `-w` 选项适用于 `authid`。

`authid` 的值是标识映射中使用的主体。`authid` 应包含 `dn:` 前缀后跟目录中的有效用户 DN，或者包含 `u:` 前缀后跟由客户端确定的任何字符串。`authid` 的这种用法允许您使用第 114 页中的 “DIGEST-MD5 标识映射”中显示的映射。

最常用的配置是使用 SSL 连接通过 LDAPS 安全端口提供加密，以及使用 DIGEST-MD5 提供客户端验证。以下示例将通过 SSL 执行相同的操作：

```
$ ldapsearch -h host1 -p 1636 \
 -Z -P .mozilla/bjensen/BJE6001.slt/cert8.db \
 -N "cert-example" -w - \
 -o mech=DIGEST-MD5 [-o realm="example.com"] \
 -o authid="dn:uid=bjensen,dc=example,dc=com" \
 -o authzid="dn:uid=bjensen,dc=example,dc=com" \
 -o secProp="minssf=0,maxssf=0,noplain" \
 -b "dc=example,dc=com" "(givenname=Richard)"
```

在此示例中，由于操作是通过 SSL 执行的，因此 `ldapsearch` 命令需要使用 `-N` 和 `-w` 选项。但是，这些选项不会用于客户端验证。服务器将执行 `authid` 值中主体的其他 DIGEST-MD5 标识映射。

在客户端中使用 Kerberos SASL GSSAPI

在客户端中使用 GSSAPI 机制时，不必安装用户证书，但必须配置 Kerberos V5 安全系统。此外，如果要使用加密的 SSL 连接，则必须信任服务器证书，如第 102 页中的 “管理证书”中所述。

▼ 在主机上配置 Configure Kerberos V5

必须在将要运行 LDAP 客户端的主机上配置 Kerberos V5。

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1. **按照安装说明安装 Kerberos V5。**

   Sun 建议安装 Sun Enterprise Authentication Mechanism 1.0.1 客户端软件。

2. **配置 Kerberos 软件。**

   使用 Sun Enterprise Authentication Mechanism 软件对 `/etc/krb5` 下的文件进行配置。此配置将设置 `kdc` 服务器，并定义默认领域和 Kerberos 系统所需的任何其他配置。

3. **修改文件 `/etc/gss/mech`，使列出的第一个值为 `kerberos_v5`（如有必要）。**

▼ 指定用于 Kerberos 验证的 SASL 选项

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1 **使用通过 GSSAPI 机制启用的客户端应用程序之前，先使用用户主体初始化 Kerberos 安全系统。**

```
$ kinit user-principal
```

其中 *user-principal* 是您的 SASL 标识，例如 `bjensen@example.com`。

2 **指定使用 Kerberos 时所需的 SASL 选项。**

请注意，在 UNIX 环境中，必须将 `SASL_PATH` 环境变量设置为 SASL 库的正确路径。例如，在 Korn shell 中：

```
$ export SASL_PATH=SASL-library
```

此路径假定目录服务器安装在调用 LDAP 工具的相同主机上。

`ldapsearch` 工具的以下示例说明如何使用 `-o`（小写字母 o）选项指定使用 Kerberos 时所需的 SASL 选项：

```
$ ldapsearch -h www.host1.com -p 1389 -o mech=GSSAPI -o authid="bjensen@EXAMPLE.COM" \
 -o authzid="bjensen@EXAMPLE.COM" -b "dc=example,dc=com" "(givenname=Richard)"
```

可以省略 `authid`，因为它会出现在由 `kinit` 命令初始化的 Kerberos 缓存中。如果 `authid` 存在，则 `authid` 和 `authzid` 必须相同，但不会使用适用于代理操作的 `authzid`。`authid` 的值是标识映射中使用的主体。此主体必须是包括领域的完整主体。请参见第 117 页中的 “GSSAPI 标识映射”。

使用 GSSAPI 和 SASL 进行 Kerberos 验证的示例配置

为目录服务器配置 Kerberos 的过程可能非常复杂。应该首先参考 Kerberos 文档。

要获取更多帮助，请通过以下示例过程了解应该执行哪些步骤。但是请注意，此过程仅仅是一个示例。您必须对过程进行相应修改，使其符合您自己的配置和环境。

可以在《System Administration Guide: Security Services》中找到有关在 Solaris 操作系统中配置和使用 Kerberos 的其他信息。本指南是 Solaris 文档集的一部分。您还可以参考手册页。

此示例以及所使用的步骤的相关信息如下所示：

1. 第 122 页中的 “此示例的假设”
2. 第 122 页中的 “所有计算机：编辑 Kerberos 客户端配置文件”
3. 第 124 页中的 “所有计算机：编辑管理服务器 ACL 配置文件”
4. 第 124 页中的 “KDC 计算机：编辑 KDC 服务器配置文件”
5. 第 125 页中的 “KDC 计算机：创建 KDC 数据库”
6. 第 125 页中的 “KDC 计算机：创建管理主体和密钥表”
7. 第 125 页中的 “KDC 计算机：启动 Kerberos 守护进程”
8. 第 126 页中的 “KDC 计算机：为 KDC 和目录服务器计算机添加主机主体”
9. 第 126 页中的 “KDC 计算机：为目录服务器添加 LDAP 主体”

此示例的假设

此示例过程介绍如何将一台计算机配置为密钥分发中心 (Key Distribution Center, KDC)，并将另一台计算机配置为运行目录服务器。此过程的结果是用户可以通过 GSSAPI 执行 Kerberos 验证。

可以在同一台计算机上同时运行 KDC 和目录服务器。如果选择在同一台计算机上同时运行 KDC 和目录服务器，请使用相同的过程，但可以在目录服务器计算机的步骤中省略已对 KDC 计算机执行的部分。

此过程对于所使用的环境进行了许多假设。使用示例过程时，请根据您的环境适当地修改值。这些假设如下：

- 此系统已安装全新的 Solaris 9 软件和最新的推荐修补程序簇。如果未安装相应的 Solaris 修补程序，则目录服务器的 Kerberos 验证可能会失败。

  请注意，尽管此处介绍的过程与适用于 Solaris 10 的过程大体相同，但仍存在一些差别。配置文件格式存在细微差别，某些命令的输出也可能不同。
- 运行 Kerberos 守护进程的计算机具有全限定域名 `kdc.example.com`。必须将计算机配置为使用 DNS 作为命名服务。此配置为 Kerberos 的必需配置。如果使用其他命名服务（如 `file`），则某些操作可能会失败。
- 运行目录服务器的计算机具有全限定域名 `directory.example.com`。必须将此计算机也配置为使用 DNS 作为命名服务。
- 目录服务器计算机作为通过 Kerberos 进行目录服务器验证的客户端系统。可以从任何能够与目录服务器和 Kerberos 守护进程进行通信的系统中执行此验证。但是，此示例所需的全部组件都是随目录服务器提供的，并从该系统中执行验证。
- 目录服务器中的用户具有 `uid=`*username*`,ou=People,dc=example,dc=com` 格式的 DN。相应的 Kerberos 主体为 *username*`@EXAMPLE.COM`。如果使用其他命名模式，则必须使用不同的 GSSAPI 标识映射。

所有计算机：编辑 Kerberos 客户端配置文件

`/etc/krb5/krb5.conf` 配置文件提供 Kerberos 客户端与 KDC 通信时所需的信息。

在 KDC 计算机、目录服务器计算机以及将使用 Kerberos 进行目录服务器验证的任何客户机上编辑 /etc/krb5/krb5.conf 配置文件。

- 将出现的每个 "___default_realm___" 替换为 "EXAMPLE.COM"。
- 将出现的每个 "___master_kdc___" 替换为 "kdc.example.com"。
- 删除包含 "___slave_kdcs___" 的行，因为只有一个 Kerberos 服务器。
- 将 "___domain_mapping___" 替换为 ".example.com = EXAMPLE.COM"（注意 .example.com 开头处的句点）。

已更新的 /etc/krb5/krb5.conf 配置文件应该与以下示例内容类似。

示例 5–1 已编辑的 Kerberos 客户端配置文件 /etc/krb5/krb5.conf

```
#pragma ident   "@(#)krb5.conf  1.2     99/07/20 SMI"
# Copyright (c) 1999, by Sun Microsystems, Inc.
# All rights reserved.
#
# krb5.conf template
# In order to complete this configuration file
# you will need to replace the __<name\>__ placeholders
# with appropriate values for your network.
#

[libdefaults]
        default_realm = EXAMPLE.COM
[realms]
        EXAMPLE.COM = {
                kdc = kdc.example.com
                admin_server = kdc.example.com
        }
[domain_realm]
        .example.com = EXAMPLE.COM
[logging]
        default = FILE:/var/krb5/kdc.log
        kdc = FILE:/var/krb5/kdc.log
        kdc_rotate = {

# How often to rotate kdc.log. Logs will get rotated no more
# often than the period, and less often if the KDC is not used
# frequently.
                period = 1d

# how many versions of kdc.log to keep around (kdc.log.0, kdc.log.1, ...)
                versions = 10
        }
```

示例 5–1 已编辑的 Kerberos 客户端配置文件 /etc/krb5/krb5.conf （续）

```
[appdefaults]
        kinit = {
                renewable = true
                forwardable= true
        }
        gkadmin = {
                help_url =
 http://docs.sun.com:80/ab2/coll.384.1/SEAM/@AB2PageView/1195
        }
```

所有计算机：编辑管理服务器 ACL 配置文件

在 /etc/krb5/kadm5.acl 配置文件中将 "___default_realm___" 替换为 "EXAMPLE.COM"。已更新的文件应该与以下示例类似。

示例 5–2 已编辑的管理服务器 ACL 配置文件

```
#
# Copyright (c) 1998-2000 by Sun Microsystems, Inc.
# All rights reserved.
#
# pragma ident   "@(#)kadm5.acl  1.1     01/03/19 SMI"
*/admin@EXAMPLE.COM *
```

KDC 计算机：编辑 KDC 服务器配置文件

编辑 /etc/krb5/kdc.conf 文件，将 "___default_realm___" 替换为 "EXAMPLE.COM"。已更新的文件应该与以下示例类似。

示例 5–3 已编辑的 KDC 服务器配置文件 /etc/krb5/kdc.conf

```
# Copyright 1998-2002 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
#ident  "@(#)kdc.conf   1.2     02/02/14 SMI"

[kdcdefaults]
        kdc_ports = 88,750

[realms]
        EXAMPLE.COM = {
                profile = /etc/krb5/krb5.conf
                database_name = /var/krb5/principal
                admin_keytab = /etc/krb5/kadm5.keytab
```

示例 5–3 已编辑的 KDC 服务器配置文件 /etc/krb5/kdc.conf （续）

```
                acl_file = /etc/krb5/kadm5.acl
                kadmind_port = 749
                max_life = 8h 0m 0s
                max_renewable_life = 7d 0h 0m 0s
                default_principal_flags = +preauth
        }
```

KDC 计算机：创建 KDC 数据库

```
$ /usr/sbin/kdb5_util create -r EXAMPLE.COM -s
Initializing database '/var/krb5/principal' for realm 'EXAMPLE.COM',
master key name 'K/M@EXAMPLE.COM'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key: password
Re-enter KDC database master key to verify: password
$
```

KDC 计算机：创建管理主体和密钥表

使用以下命令创建管理用户，此用户具有 kws/admin@EXAMPLE.COM 主体以及管理守护进程将要使用的服务密钥。

```
$ /usr/sbin/kadmin.local
kadmin.local:  add_principal kws/admin
Enter password for principal "kws/admin@EXAMPLE.COM": secret
Re-enter password for principal "kws/admin@EXAMPLE.COM": secret
Principal "kws/admin@EXAMPLE.COM" created.
kadmin.local:  ktadd -k /etc/krb5/kadm5.keytab kadmin/kdc.example.com
Entry for principal kadmin/kdc.example.com with kvno 3, encryption type
 DES-CBC-CRC added to keytab WRFILE:/etc/krb5/kadm5.keytab.
kadmin.local:  ktadd -k /etc/krb5/kadm5.keytab changepw/kdc.example.com

Entry for principal changepw/kdc.example.com with kvno 3, encryption type
 DES-CBC-CRC added to keytab WRFILE:/etc/krb5/kadm5.keytab.
kadmin.local:  ktadd -k /etc/krb5/kadm5.keytab kadmin/changepw
Entry for principal kadmin/changepw with kvno 3, encryption type
 DES-CBC-CRC added to keytab WRFILE:/etc/krb5/kadm5.keytab.
kadmin.local:  quit$
```

KDC 计算机：启动 Kerberos 守护进程

运行以下命令来启动 KDC 和管理守护进程：

```
$ /etc/init.d/kdc start
$ /etc/init.d/kdc.master start
$
```

KDC 进程在进程列表中将显示为 `/usr/lib/krb5/krb5kdc`。管理守护进程将显示为 `/usr/lib/krb5/kadmind`。

请注意，在 Solaris 10 操作系统中，守护进程由服务管理工具 (Service Management Facility, SMF) 框架进行管理。在 Solaris 10 操作系统上启动守护进程 :

```
$ svcadm disable network/security/krb5kdc
$ svcadm enable network/security/krb5kdc
$ svcadm disable network/security/kadmin
$ svcadm enable network/security/kadmin
$
```

KDC 计算机 : 为 KDC 和目录服务器计算机添加主机主体

使用以下一系列命令将主机主体添加到 KDC 和目录服务器计算机的 Kerberos 数据库。某些 Kerberos 实用程序（如 `klist`）将使用主机主体。

```
$ /usr/sbin/kadmin -p kws/admin
Enter Password: secret
kadmin:  add_principal -randkey host/kdc.example.com
Principal "host/kdc.example.com@EXAMPLE.COM" created.
kadmin:  ktadd host/kdc.example.com
Entry for principal host/kdc.example.com with kvno 3, encryption type
 DES-CBC-CRC added to keytab WRFILE:/etc/krb5/krb5.keytab.
kadmin:  add_principal -randkey host/directory.example.com
Principal "host/directory.example.com@EXAMPLE.COM" created.
kadmin:  ktadd host/directory.example.com
Entry for principal host/directory.example.com with kvno 3, encryption type
 DES-CBC-CRC added to keytab WRFILE:/etc/krb5/krb5.keytab.
kadmin:  quit
$
```

KDC 计算机 : 为目录服务器添加 LDAP 主体

目录服务器必须有自身的主体，才能检验正在进行验证的用户所持有的 Kerberos 票证是否有效。目前已将目录服务器硬编码为需要 `ldap/`*fqdn*`@`*realm* 格式的主体，其中 *fqdn* 是目录服务器的全限定域名，而 *realm* 是 Kerberos 领域。*fqdn* 必须与安装目录服务器时提供的全限定名称相匹配。在此案例中，目录服务器的主体应该为 `ldap/directory.example.com@EXAMPLE.COM`。

使用以下一系列命令为目录服务器创建 LDAP 主体 :

```
$ /usr/sbin/kadmin -p kws/admin
Enter Password: secret
kadmin: add_principal -randkey ldap/directory.example.com
Principal "ldap/directory.example.com@EXAMPLE.COM" created.
kadmin: quit
$
```

KDC 计算机：向 KDC 添加测试用户

要执行 Kerberos 验证，Kerberos 数据库中必须存在要进行验证的用户。在此示例中，用户具有用户名 `kerberos-test`，这意味着 Kerberos 主体为 `kerberos-test@EXAMPLE.COM`。

使用此示例中的一系列命令创建用户：

```
$ /usr/sbin/kadmin -p kws/admin
Enter Password: secret
kadmin:  add_principal kerberos-test
Enter password for principal "kerberos-test@EXAMPLE.COM": secret

Re-enter password for principal "kerberos-test@EXAMPLE.COM": secret

Principal "kerberos-test@EXAMPLE.COM" created.
kadmin:  quit
$
```

目录服务器计算机：安装目录服务器

安装 Directory Server 6.0 和最新的修补程序。以下为示例设置。

变量类型	示例值
全限定计算机名	`directory.example.com`
安装目录	`/opt/SUNWdsee`
实例路径	`/local/ds`
服务器用户	`unixuser`
服务器组	`unixgroup`
服务器标识符	`directory`
服务器端口	`389`
后缀	`dc=example,dc=com`
管理员 ID	`admin`

变量类型	示例值
管理域	example.com
目录管理者 DN	cn=admin,cn=Administrators,cn=config
管理端口	390

目录服务器计算机：将目录服务器配置为启用 GSSAPI

首先，创建文件 /data/ds/shared/bin/gssapi.ldif 以定义目录服务器应使用的映射，并基于主体标识要进行验证的 Kerberos 用户。请创建与以下示例内容相同的文件内容。

示例 5–4 gssapi.ldif 文件内容

```
dn: cn=GSSAPI,cn=identity mapping,cn=config
changetype: add
objectClass: top
objectClass: nsContainer
cn: GSSAPI
dn: cn=default,cn=GSSAPI,cn=identity mapping,cn=config
changetype: add
objectClass: top
objectClass: nsContainer
objectClass: dsIdentityMapping
objectClass: dsPatternMatching
cn: default
dsMatching-pattern: \${Principal}
dsMatching-regexp: (.*)@EXAMPLE.COM
dsMappedDN: uid=\$1,ou=People,dc=example,dc=com

dn: cn=SASL,cn=security,cn=config
changetype: modify
replace: dsSaslPluginsPath
dsSaslPluginsPath: /usr/lib/mps/sasl2/libsasl.so
```

然后，使用 ldapmodify 命令更新目录服务器，以启用具有相应映射的 GSSAPI，如以下示例所示：

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w - -a -f /data/ds/shared/bin/gssapi.ldif
adding new entry cn=GSSAPI,cn=identity mapping,cn=config
adding new entry cn=default,cn=GSSAPI,cn=identity mapping,cn=config
modifying entry cn=SASL,cn=security,cn=config
$
```

目录服务器计算机：创建目录服务器密钥表

如前面所述，要通过 GSSAPI 验证 Kerberos 用户，目录服务器在 KDC 中必须具有自身的主体。要使验证正常工作，主体信息必须包含在目录服务器计算机上的 Kerberos 密钥表中。用于运行目录服务器的用户帐户必须能够读取包含此信息的文件。

使用以下一系列命令通过正确的属性创建密钥表文件：

```
$ /usr/sbin/kadmin -p kws/admin
Enter Password: secret
kadmin:  ktadd -k //local/ds/config/ldap.keytab ldap/directory.example.com
Entry for principal ldap/directory.example.com with kvno 3, encryption type
 DES-CBC-CRC added to keytab
 WRFILE:/local/ds/config/ldap.keytab.
kadmin:  quit
$
```

在此自定义密钥表上更改权限和所有权。使得用于运行目录服务器的用户帐户成为此密钥表的所有者，并且只有该用户可以读取此密钥表：

```
$ chown unixuser:unixgroup /local/ds/config /ldap.keytab
$ chmod 600 /local/ds/config/ldap.keytab
$
```

默认情况下，目录服务器会尝试使用文件 `/etc/krb5/krb5.keytab` 中的标准 Kerberos 密钥表。但是，允许目录服务器用户读取此文件可能会构成安全威胁，因此为目录服务器创建了自定义密钥表。

将目录服务器配置为使用新的自定义密钥表。可以通过设置 `KRB5_KTNAME` 环境变量完成此操作。

最后，重新启动目录服务器以使更改生效：

```
$ KRB5_KTNAME=/etc/krb5/ldap.keytab dsadm restart /local/ds
```

目录服务器计算机：向目录服务器添加测试用户

要使 Kerberos 用户通过目录服务器的验证，该用户必须具有与其 Kerberos 主体相对应的目录条目。

在前面的步骤中，已使用主体 `kerberos-test@EXAMPLE.COM` 将测试用户添加到 Kerberos 数据库。由于向目录中添加了标识映射配置，因此该用户的相应目录条目必须具有 DN `uid=kerberos-test,ou=People,dc=example,dc=com`。

向目录添加用户之前，必须先创建包含以下内容的文件 `testuser.ldif`。

示例 5–5 新的 testuser.ldif 文件

```
dn: uid=kerberos-test,ou=People,dc=example,dc=com
changetype: add
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
uid: kerberos-test
givenName: Kerberos
sn: Test
cn: Kerberos Test
description: An account for testing Kerberos authentication through GSSAPI
```

然后，使用 ldapmodify 将此条目添加到服务器：

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w - -f testuser.ldif
adding new entry uid=kerberos-test,ou=People,dc=example,dc=com
$
```

目录服务器计算机：以测试用户的身份获取 Kerberos 票证

Kerberos 数据库、目录服务器和 KDC 中都存在测试用户。因此，现在能够以测试用户身份通过目录服务器的验证（使用 GSSAPI 和 Kerberos）。

首先，使用 kinit 命令为用户获取 Kerberos 票证，如以下示例所示：

```
$ kinit kerberos-test
Password for kerberos-test@EXAMPLE.COM: secret
$
```

然后，使用 klist 命令查看与此票证有关的信息：

```
$ klist
Ticket cache: /tmp/krb5cc_0
Default principal: kerberos-test@EXAMPLE.COM

Valid starting           Expires                  Service principal
Sat Jul 24 00:24:15 2004  Sat Jul 24 08:24:15 2004  krbtgt/EXAMPLE.COM@EXAMPLE.COM
        renew until Sat Jul 31 00:24:15 2004
$
```

客户机：通过 GSSAPI 进行目录服务器验证

最后一步是使用 GSSAPI 进行目录服务器验证。随目录服务器提供的 ldapsearch 实用程序支持 SASL 验证，包括 GSSAPI、DIGEST-MD5 和 EXTERNAL 机制。但是，要使用 GSSAPI 进行绑定，您必须向客户端提供 SASL 库所在的路径。通过将 SASL_PATH 环境变量设置为 lib/sasl 目录可以提供此路径：

```
$ SASL_PATH=SASL-library
$ export SASL_PATH
$
```

要实际使用 `ldapsearch` 在目录服务器上执行基于 Kerberos 的验证，必须包含 `-o mech=GSSAPI` 和 `-o authzid=`*principal* 参数。

此外，还必须指定全限定主机名（此处显示为 `-h directory.example.com`），该主机名必须与服务器 `cn=config` 上的 `nsslapd-localhost` 属性值相匹配。此处必须使用 `-h` 选项，因为 GSSAPI 验证过程需要客户端提供的主机名，以便与服务器提供的主机名进行匹配。

以下示例将在 `dc=example,dc=com` 条目被验证为之前创建的 Kerberos 测试用户帐户时检索此条目：

```
$ ldapsearch -h directory.example.com -p 389 -o mech=GSSAPI \
 -o authzid="kerberos-test@EXAMPLE.COM" -b "dc=example,dc=com" -s base "(objectClass=*)"
version: 1
dn: dc=example,dc=com
dc: example
objectClass: top
objectClass: domain
$
```

检查目录服务器访问日志，以确认是否按预期方式处理验证：

```
$ tail -12 /local/ds/logs/access

[24/Jul/2004:00:30:47 -0500] conn=0 op=-1 msgId=-1 - fd=23 slot=23 LDAP
        connection from 1.1.1.8 to 1.1.1.8
[24/Jul/2004:00:30:47 -0500] conn=0 op=0 msgId=1 - BIND dn="" method=sasl
     version=3 mech=GSSAPI
[24/Jul/2004:00:30:47 -0500] conn=0 op=0 msgId=1 - RESULT err=14 tag=97
     nentries=0 etime=0, SASL bind in progress
[24/Jul/2004:00:30:47 -0500] conn=0 op=1 msgId=2 - BIND dn="" method=sasl
     version=3 mech=GSSAPI
[24/Jul/2004:00:30:47 -0500] conn=0 op=1 msgId=2 - RESULT err=14 tag=97
     nentries=0 etime=0, SASL bind in progress
[24/Jul/2004:00:30:47 -0500] conn=0 op=2 msgId=3 - BIND dn="" method=sasl
     version=3 mech=GSSAPI
[24/Jul/2004:00:30:47 -0500] conn=0 op=2 msgId=3 - RESULT err=0 tag=97
     nentries=0 etime=0 dn="uid=kerberos-test,ou=people,dc=example,dc=com"
[24/Jul/2004:00:30:47 -0500] conn=0 op=3 msgId=4 - SRCH base="dc=example,dc=com"
     scope=0 filter="(objectClass=*)" attrs=ALL
[24/Jul/2004:00:30:47 -0500] conn=0 op=3 msgId=4 - RESULT err=0 tag=101 nentries=1
     etime=0
[24/Jul/2004:00:30:47 -0500] conn=0 op=4 msgId=5 - UNBIND
[24/Jul/2004:00:30:47 -0500] conn=0 op=4 msgId=-1 - closing - U1
```

```
[24/Jul/2004:00:30:48 -0500] conn=0 op=-1 msgId=-1 - closed.
$
```

此示例表明绑定过程分为三个步骤。前两步返回 LDAP 结果 14（正在进行 SASL 绑定），第三步显示绑定已成功完成。`method=sasl` 和 `mech=GSSAPI` 标记表明绑定使用了 GSSAPI SASL 机制。成功绑定响应末尾的 `dn="uid=kerberos-test,ou=people,dc=example,dc=com"` 表明绑定是由正确的用户执行的。

让渡验证

让渡验证 (Pass-through authentication, PTA) 是一种机制，此机制将按照绑定 DN 过滤绑定请求。一个目录服务器（委托方）收到绑定请求后，可以基于过滤器查询另一个目录服务器（被委托方）以验证绑定请求。作为此功能的一部分，对于未必存储在本地数据库中的条目，PTA 插件允许委托方目录服务器接受基于密码的简单绑定操作。

DSCC 也可使用 PTA 插件与服务器进行专用通信。在 DSCC 中注册服务器实例时，将启用 PTA 插件，并将 DSCC URL 作为参数进行添加。

```
$ dsconf get-plugin-prop -h host -p port "Pass Through Authentication" enabled argument
argument  :  ldap://DSCC_URL:DSCC_PORT/cn=dscc
enabled   :  on
```

注 – 请尽量避免针对个人使用而修改 PTA 插件。修改 PTA 插件可能会导致 DSCC 出现访问问题。

如果无法避免修改 PTA 插件，则必须执行以下操作：

- 继续将 `enabled` 属性设置为 `on`。
- 在参数中保留 DSCC URL，但您可以向 `argument` 属性添加其他值。

如果 PTA 插件已被禁用，或者已从参数中删除 DSCC URL，则服务器实例在 DSCC 中将显示为 `inaccessible`。如果发生这种情况，DSCC 将自动为您提供用于重置 PTA 插件的选项。

◆ ◆ ◆ 第 6 章

目录服务器访问控制

对目录访问的控制是创建安全目录不可或缺的一部分。本章介绍访问控制指令 (Access Control Instruction, ACI)，这些指令用于确定要为访问目录的用户授予哪些权限。

请在目录部署的规划阶段定义符合总体安全策略的访问控制策略。有关规划访问控制策略的提示，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Deployment Planning Guide》。

有关 ACI 的其他信息（包括 ACI 语法和绑定规则），请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》。

本章包含以下主题：

创建、查看和修改 ACI

可以通过使用目录服务控制中心 (Directory Service Control Center, DSCC) 或命令行创建 ACI。无论选择哪种方法，查看并复制现有 ACI 值通常比从头创建新 ACI 更容易。

可以在 DSCC 中查看和修改 `aci` 属性值。有关如何通过 DSCC 修改 ACI 的信息，请参见 DSCC 联机帮助。

▼ 创建、修改和删除 ACI

要使用命令行创建 ACI，应首先使用 LDIF 语句在文件中创建 ACI。然后通过使用 `ldapmodify` 命令将 ACI 添加到您的目录树中。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1 **在 LDIF 文件中创建 ACI。**

```
dn: dc=example,dc=com
changetype: modify
add: aci
aci: (target)(version 3.0; acl "name";permission bindrules;)
```

此示例说明如何添加 ACI。要修改或删除 ACI，请将 `add` 替换为 `replace` 或 `delete`。

有关常用 ACI 的更多示例，请参见第 135 页中的 “访问控制使用示例” 。

2 **使用 LDIF 文件进行更改。**

```
$ ldapmodify -h host -p port -D cn=admin,cn=Administrators,cn=config -w - -f ldif-file
```

▼ 查看 ACI 属性值

ACI 作为条目的一个或多个 `aci` 属性值进行存储。`aci` 属性为多值操作属性，可由目录用户读取和修改。因此，ACI 属性自身应受 ACI 保护。管理用户通常具有 `aci` 属性的完全访问权限。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

- **通过运行以下 `ldapsearch` 命令查看条目的 ACI 属性值：**

```
$ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
 -b entryDN -s base "(objectclass=*)" aci
```

结果为 LDIF 文本，可将其复制到新的 LDIF ACI 定义以进行编辑。由于 ACI 的值是一个长字符串，因此 `ldapsearch` 操作的输出可能会以多行显示。此外，第一个空格为连续标记。如果不希望 LDIF 输出包含连续标记，请使用 `-T` 选项。复制和粘贴 LDIF 输出时应考虑输出格式。

注 – 要查看 `aci` 值所授予和拒绝的权限，请参见第 146 页中的 “查看有效权限” 。

▼ 查看根级别的 ACI

创建后缀时，将在顶级或根级别上创建一些默认 ACI。对于目录数据，这些 ACI 允许默认管理用户 `cn=admin,cn=Administrators,cn=config` 具有与目录管理员相同的访问权限。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

- **查看默认的根级别 ACI。**

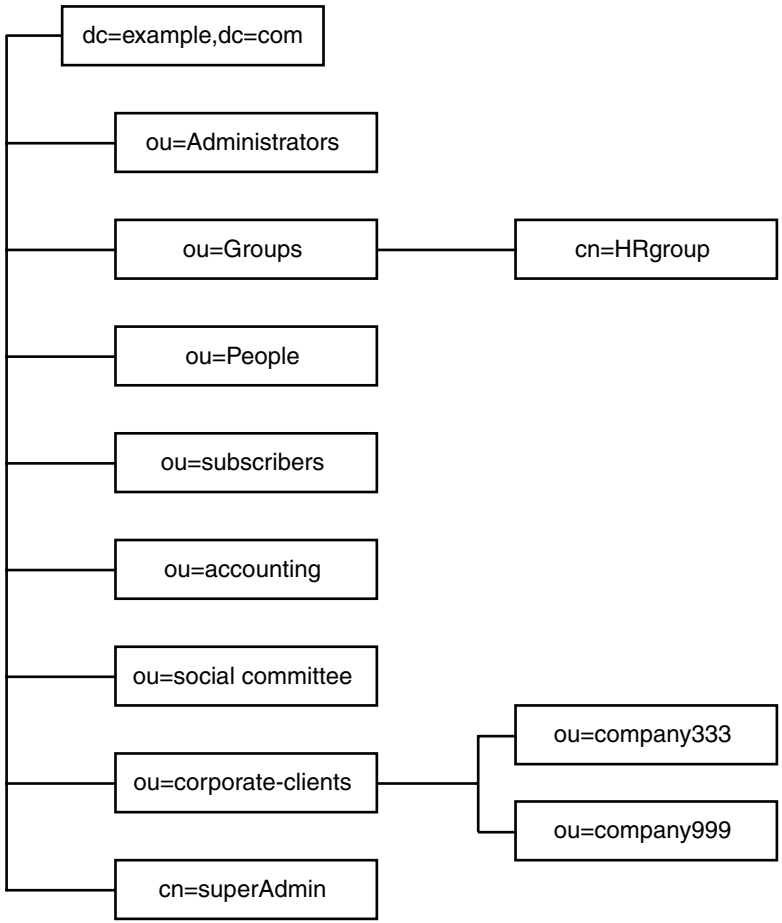
```
$ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
 -b "" -s base "(objectclass=*)" aci
```

访问控制使用示例

本部分中的示例说明虚构的 ISP 公司 Example.com 将如何实现访问控制策略。

此外，您也可以在随安装提供的示例 LDIF 文件 (*install_path*/ds6/ldif/Example.ldif) 中找到 ACI 示例。

所有示例都说明如何使用 LDIF 文件执行给定任务。下图以图形方式显示了 example.com 目录信息树。



Example.com 提供了 Web 托管服务和 Internet 访问。Example.com 的部分 Web 托管服务用于托管客户端公司的目录。Example.com 实际上托管并部分管理两家中型公司（Company333 和 Company999）的目录。此外，Example.com 还提供对大量单个订户的 Internet 访问。

Example.com 希望制定以下访问规则：

- 为 Example.com 员工授予对整个 Example.com 树的匿名读取、搜索和比较访问权限。请参见第 137 页中的 “授予匿名访问权限”。
- 为 Example.com 员工授予对个人信息的写入权限，如 `homeTelephoneNumber` 和 `homeAddress`。请参见第 137 页中的 “授予对个人条目的写入访问权限”。
- 为 Example.com 订户授予读取 `dc=example,dc=com` 条目（但不包括该条目下的任何条目）的权限，以获取公司联系信息。请参见第 138 页中的 “授予对特定级别的访问权限”。
- 为 Example.com 员工授予将任何角色（某些重要角色除外）添加到其条目的权限。请参见第 139 页中的 “限制对重要角色的访问权限”。
- 针对某个后缀为特定管理员授予与目录管理员相同的权限。请参见第 140 页中的 “为角色授予对整个后缀的完全访问权限”。
- 为 Example.com 人力资源组授予对 People 分支中条目的所有权限。请参见第 140 页中的 “为组授予对后缀的完全访问权限”。
- 为所有 Example.com 员工授予以下权限：在目录的 Social Committee 分支下创建组条目，以及删除某个员工所拥有的组条目。请参见第 141 页中的 “授予添加和删除组条目的权限”。
- 为所有 Example.com 员工授予将自身添加到目录 Social Committee 分支下的组条目的权限。请参见第 142 页中的 “允许用户在组中添加或删除自身”。
- 为 Company333 和 Company999 的目录管理者（角色）授予对其各自目录树分支的访问权限（在特定条件下）。这些条件包括 SSL 验证、时间和日期限制，以及指定的位置。请参见第 142 页中的 “为组或角色授予条件访问权限”。
- 为单个订户授予对其各自条目的访问权限。请参见第 137 页中的 “授予对个人条目的写入访问权限”。
- 拒绝单个订户对其自身条目中的帐单信息的访问权限。请参见第 143 页中的 “拒绝访问”。
- 为所有人授予对单个订户子树的匿名访问权限，明确要求不列出的订户除外。如果需要，此部分目录可以作为防火墙外的只读服务器，并且每天更新一次。请参见第 137 页中的 “授予匿名访问权限”和第 145 页中的 “使用过滤设置目标”。

授予匿名访问权限

大多数目录都配置为至少允许您匿名访问一个后缀，以进行读取、搜索或比较。如果要运行公司人员目录（如可供员工搜索的电话薄），则可能需要设置这些权限。Example.com 内部即为这种情况，如 第 137 页中的 “ACI "Anonymous Example.com"” 中所示。

作为 ISP，Example.com 还要创建允许所有人访问的公共电话薄，以提供其所有订户的联系信息。相关内容如 第 137 页中的 “ACI "Anonymous World"” 中所述。

ACI "Anonymous Example.com"

在 LDIF 中，要为 Example.com 员工授予对整个 Example.com 树的读取、搜索和比较权限，可编写以下语句：

```
aci: (targetattr !="userPassword")(version 3.0; acl "Anonymous
 example"; allow (read, search, compare)
 userdn= "ldap:///anyone") ;)
```

此示例假定 `aci` 已添加到 `dc=example,dc=com` `entry` 中。请注意，`userPassword` 属性被排除在 ACI 范围之外。

注 – 任何保密属性或可见属性都应像密码属性一样列出（使用 `(targetattr !="`*attribute-name*`")`）。

ACI "Anonymous World"

在 LDIF 中，要为所有人授予对单个订户子树的读取和搜索访问权限，但不允许访问未列出订户的信息，可编写以下语句：

```
aci: (targetfilter= "(!(unlistedSubscriber=yes))")
 (targetattr="homePostalAddress || homePhone || mail")
 (version 3.0; acl "Anonymous World"; allow (read, search)
 userdn="ldap:///anyone";)
```

此示例假定 ACI 已添加到 `ou=subscribers,dc=example, dc=com` 条目中。此示例还假定每个订户条目都具有 `unlistedSubscriber` 属性（设置为 `yes` 或 `no`）。目标定义将根据此属性的值过滤掉未列出的订户。有关过滤器定义的详细信息，请参阅第 145 页中的 “使用过滤设置目标”。

授予对个人条目的写入访问权限

许多目录管理者都允许内部用户更改其自身条目中的部分（而非全部）属性。Example.com 的目录管理者允许用户更改自身的密码、家庭电话号码和家庭地址，但不允许更改其他内容。相关内容如 第 138 页中的 “ACI "Write Example.com"” 中所述。

Example.com 还有一条策略，即，如果订户建立到目录的 SSL 连接，则允许订户在 Example.com 树中更新自身的个人信息。相关内容如 第 138 页中的 “ACI "Write Subscribers"” 中所述。

ACI "Write Example.com"

注 – 通过设置此权限，还可以为用户授予删除属性值的权限。

在 LDIF 中，要为 Example.com 员工授予更新其家庭电话号码和家庭地址的权限，可编写以下语句：

```
aci: (targetattr="homePhone ||
 homePostalAddress")(version 3.0; acl "Write Example.com";
 allow (write) userdn="ldap:///self" ;)
```

此示例假定 ACI 已添加到 `ou=People,dc=example,dc=com` 条目中。

ACI "Write Subscribers"

注 – 通过设置此权限，还可以为用户授予删除属性值的权限。

在 LDIF 中，要为 Example.com 订户授予更新其家庭电话号码的权限，可编写以下语句：

```
aci: (targetattr="homePhone")
 (version 3.0; acl "Write Subscribers"; allow (write)
 userdn= "ldap://self" and authmethod="ssl";)
```

此示例假定 `aci` 已添加到 `ou=subscribers,dc=example, dc=com` 条目中，并且用户必须使用 SSL 进行绑定。

请注意，Example.com 订户对其家庭地址没有写入访问权限，因为他们可能会删除该属性。家庭地址是 Example.com 寄发帐单时所需的重要业务信息。

授予对特定级别的访问权限

可以通过设置 ACI 的范围来影响目录树中的不同级别，以便对您所允许的访问权限级别进行微调。可将目标 ACI 范围设置为以下任一选项：

`base` 条目自身

`onelevel` 条目自身及其下一级的所有条目

`subtree` 条目自身及其下面的所有条目（不限制深度）

ACI "Read Example.com only"

在 LDIF 中，要为 Example.com 订户授予读取 `dc=example,dc=com` 条目的权限（以获取公司联系信息），但不允许访问该条目下的任何条目，可编写以下语句：

```
aci: (targetscope="base") (targetattr="*")(version 3.0;
 acl "Read Example.com only";  allow (read,search,compare)
 userdn="ldap:///cn=*,ou=subscribers,dc=example,dc=com";)
```

此示例假定 ACI 已添加到 `dc=example, dc=com` 条目中。

限制对重要角色的访问权限

您可以在目录中使用角色定义，以标识对业务至关重要的功能，如网络和目录管理。

例如，您可以通过标识系统管理员的某个子集来创建超级管理员 (`superAdmin`) 角色，这些管理员于每周特定日期的特定时间在全球公司站点上可用。或者，您也可以创建急救员 (`First Aid`) 角色，其中包括特定站点中所有受过急救培训的工作人员。有关创建角色定义的信息，请参见第 195 页中的 “管理角色”。

当某个角色在重要的公司或业务功能方面可提供任何种类的用户特权时，应考虑限制对该角色的访问权限。例如，在 Example.com 中，员工可以向自身条目中添加除超级管理员 (`superAdmin`) 角色之外的任何角色，如以下示例所示。

ACI "Roles"

在 LDIF 中，要为 Example.com 员工授予在自身条目中添加除超级管理员 (`superAdmin`) 之外的任何角色的权限，可编写以下语句：

```
aci: (targetattr="*") (targattrfilters="add=nsRoleDN:
 (nsRoleDN !="cn=superAdmin, dc=example, dc=com")")
 (version 3.0; acl "Roles"; allow (write)
 userdn= "ldap:///self" ;)
```

此示例假定 ACI 已添加到 `ou=People,dc=example, dc=com` 条目中。

为角色授予对整个后缀的完全访问权限

有时，针对某个后缀为特定用户授予与目录管理员相同的权限是非常有用的。在 Example.com 中，Kirsten Vaughan 是目录服务器管理员。她具有超级管理员 (`superAdmin`) 角色。此角色具有以下优点：

- 由于可以强制以自身标识进行绑定的管理员使用强验证（如 SSL），因此更加安全
- 由于只有少数人知道目录管理员密码，因此更加安全
- 可通过日志记录提高可追溯性

注 – 将 Kirsten Vaughan 添加到 `cn=Administrators,cn=config` 组还会为她授予与目录管理员相同的权限。

要使用户对整个服务器具有与目录管理员相同的权限，请执行第 63 页中的 “创建具有超级用户权限的管理用户”中的过程。

ACI "Full Access"

在 LDIF 中，要为管理员 Kirsten Vaughan 授予与目录管理员相同的权限，请使用以下语句：

```
aci: (targetattr="*") (version 3.0; acl "Full Access";
 allow (all) groupdn= "ldap:///cn=SuperAdmin,dc=example,dc=com"
 and authmethod="ssl" ;)
```

此示例假定 ACI 已添加到根条目 `""`（无文本）中。

为组授予对后缀的完全访问权限

大多数目录都有用于标识特定公司功能的组。可以为组授予对部分或全部目录的访问权限。通过将访问权限应用于组，可以避免单独为每个成员设置访问权限。可以通过将用户添加到组来为这些用户授予访问权限。

例如，在创建目录服务器实例时，默认情况下将创建一个管理员组 `cn=Administrators,cn=config`，此组具有目录的完全访问权限。

在 Example.com 中，人力资源组对目录的 `ou=People` 分支具有完全访问权限，这样他们可以更新员工目录，如第 140 页中的 “ACI "HR"”中所示。

ACI "HR"

在 LDIF 中，要为人力资源组授予对目录员工分支的所有权限，请使用以下语句：

```
aci: (targetattr="*") (version 3.0; acl "HR"; allow (all)
  groupdn= "ldap:///cn=HRgroup,ou=Groups,dc=example,dc=com";)
```

此示例假定 ACI 已添加到以下条目中：

```
ou=People,dc=example,dc=com
```

授予添加和删除组条目的权限

某些组织允许员工在树中创建条目，以提高员工的工作效率，并鼓励员工为公司活力注入一己之力。例如，在 Example.com 中，Social Committee 由网球、游泳、滑雪和角色扮演等各种俱乐部组成。

任何 Example.com 员工都可以创建代表新俱乐部的组条目，如 第 141 页中的 “ACI "Create Group"” 中所示。

任何 Example.com 员工都可以成为其中某个组的成员，如第 142 页中的 “允许用户在组中添加或删除自身”中所示。

只有组的所有者才能修改或删除组条目，如 第 142 页中的 “ACI "Delete Group"” 中所示。

ACI "Create Group"

在 LDIF 中，要为 Example.com 员工授予在 `ou=Social Committee` 分支下创建组条目的权限，可编写以下语句：

```
aci: (targetattr="*") (targattrfilters="add=objectClass:
(|(objectClass=groupOfNames)(objectClass=top))")
(version 3.0; acl "Create Group"; allow (read,search,add)
userdn= "ldap:///uid=*,ou=People,dc=example,dc=com")
and dns="*.Example.com";)
```

此示例假定 ACI 已添加到 `ou=Social Committee,dc=example,dc=com` 条目中。

注 –

- 此 ACI 不授予写入权限，这意味着条目创建者无法修改条目。
- 由于服务器在后台添加值 `top`，因此需要在 `targattrfilters` 关键字中指定 `objectClass=top`。
- ACI 将客户机限制在 `example.com` 域中。

ACI "Delete Group"

在 LDIF 中，要为 Example.com 员工授予相应权限，以修改或删除其所属组（在 ou=Social Committee 分支下）的组条目，可编写以下语句：

```
aci: (targetattr = "*") (targattrfilters="del=objectClass:
(objectClass=groupOfNames)")
 (version 3.0; acl "Delete Group"; allow (write,delete)
 userattr="owner#GROUPDN";)
```

此示例假定 aci 已添加到 ou=Social Committee,dc=example,dc=com 条目中。

请注意，使用 DSCC 创建此 ACI 不是有效方式，因为您必须使用手动编辑模式创建目标过滤器，并检查组的所有权。

允许用户在组中添加或删除自身

许多目录都设置了允许用户在组中（如邮件列表）添加或删除自身的 ACI。

在 Example.com 中，员工可以将自身添加到 ou=Social Committee 子树下的任何组条目中，如 第 142 页中的 “ACI "Group Members"” 中所示。

ACI "Group Members"

在 LDIF 中，要为 Example.com 员工授予将自身添加到组的权限，可编写以下语句：

```
aci: (targettattr="member")(version 3.0; acl "Group Members";
 allow (selfwrite)
 (userdn= "ldap:///uid=*,ou=People,dc=example,dc=com") ;)
```

此示例假定 ACI 已添加到 ou=Social Committee, dc=example,dc=com 条目中。

为组或角色授予条件访问权限

在许多情况下，当您为组或角色授予对目录的访问特权时，必须确保入侵者无法模拟特权用户使用这些特权。因此，在许多情况下，为组或角色授予重要访问权限的访问控制规则通常与许多条件相关联。

例如，Example.com 为其每个托管公司（Company333 和 Company999）创建了一个目录管理者角色。Example.com 希望这两家公司能够管理各自的数据并实现各自的访问控制规则，同时又能确保数据不受侵犯。

因此，如果满足以下条件，Company333 和 Company999 将对其各自的目录树分支具有完全权限。

- 使用证书通过 SSL 对连接进行验证。
- 在星期一至星期四的 8:00 至 18:00 之间请求访问。
- 从每个公司的指定 IP 地址请求访问。

这些条件将在每个公司的某个 ACI（ACI "Company333" 和 ACI "Company999"）中列出。由于两个 ACI 的内容相同，因此以下示例只使用 "Company333" ACI。

ACI "Company333"

在 LDIF 中，要在满足上述条件的情况下为 Company333 授予对其目录分支的完全访问权限，可编写以下语句：

```
aci: (targetattr = "*") (version 3.0; acl "Company333"; allow (all)
 (roledn="ldap:///cn=DirectoryAdmin,ou=Company333,
 ou=corporate clients,dc=example,dc=com") and (authmethod="ssl")
 and (dayofweek="Mon,Tues,Wed,Thu") and (timeofday >= "0800" and
 timeofday <= "1800") and (ip="255.255.123.234"); )
```

此示例假定 ACI 已添加到 `ou=Company333,ou=corporate clients,dc=example,dc=com` 条目中。

拒绝访问

如果您已经允许对后缀的大部分进行访问，则您可能希望在现有的 ACI 下拒绝对后缀的较小部分进行访问。

注 – 应尽可能避免拒绝访问，因为它可能会导致意外或复杂的访问控制行为。可以结合使用作用域、属性列表、目标过滤器等来限制访问权限。

此外，删除拒绝访问 ACI 不会删除权限，但会扩展其他 ACI 所设置的权限。

当目录服务器评估访问权限时，它将首先读取 `deny` 权限，然后再读取 `allow` 权限。

在后面的示例中，Example.com 希望所有订户都能读取自身条目下的帐单信息，如连接时间或帐户余额。Example.com 也明确希望拒绝对该信息的写入访问权限。读取访问权限将在 第 144 页中的 “ACI "Billing Info Read"” 中进行介绍。**拒绝**访问权限将在 第 144 页中的 “ACI "Billing Info Deny"” 中进行介绍。

ACI "Billing Info Read"

在 LDIF 中，要为订户授予读取自身条目中的帐单信息的权限，可编写以下语句：

```
aci: (targetattr="connectionTime || accountBalance")
 (version 3.0; acl "Billing Info Read"; allow (search,read)
  userdn="ldap:///self";)
```

此示例假定已在模式中创建了相关属性，并且 ACI 已添加到 `ou=subscribers,dc=example,dc=com` 条目中。

ACI "Billing Info Deny"

在 LDIF 中，要拒绝订户修改自身条目中的帐单信息的权限，可编写以下语句：

```
aci: (targetattr="connectionTime || accountBalance")
 (version 3.0; acl "Billing Info Deny";
 deny (write) userdn="ldap:///self";)
```

此示例假定已在模式中创建了相关属性，并且 ACI 已添加到 `ou=subscribers,dc=example,dc=com` 条目中。

代理授权

代理授权方法是一种特殊形式的验证。通过代理授权，可为使用自身标识绑定到目录的用户授予其他用户的权限。

要将目录服务器配置为允许代理请求，必须执行以下操作：

- 为管理员授予与其他用户相同的代理权限。
- 为普通用户授予访问控制策略中所定义的一般访问权限。

注 – 您可以将代理权限授予除目录管理员之外的任何目录用户。此外，您无法将目录管理员的 DN 用作代理 DN。授予代理权限时应特别小心，因为授予此权限可将任何 DN（目录管理员 DN 除外）指定为代理 DN。如果目录服务器在同一操作中收到多个代理验证控制，则会向客户端应用程序返回错误，并且操作尝试将会失败。

示例代理授权

对于 LDAP 数据，Example.com 希望绑定为 `MoneyWizAcctSoftware` 的客户端应用程序具有与帐户管理员相同的访问权限。

将应用以下参数：

- 客户端应用程序的绑定 DN 为 `uid=MoneyWizAcctSoftware, ou=Applications,dc=example,dc=com`。
- 客户端应用程序请求访问权限的目标子树为 `ou=Accounting,dc=example,dc=com`。
- 目录中存在对 `ou=Accounting,dc=example,dc=com` 子树具有访问权限的帐户管理员。

要使客户端应用程序获取对帐户子树的访问权限（通过使用与帐户管理员相同的访问权限），必须满足以下条件：

- 帐户管理员对 `ou=Accounting,dc=example,dc=com` 子树必须具有访问权限。例如，以下 ACI 为帐户管理员条目授予所有权限：

```
aci: (targetattr="*") (version 3.0; acl "allowAll-AcctAdmin"; allow
 (all) userdn="ldap:///uid=AcctAdministrator,ou=Administrators,
 dc=example,dc=com";)
```

- 目录中必须存在为客户端应用程序授予代理权限的以下 ACI：

```
aci: (targetattr="*") (version 3.0; acl "allowproxy- accountingsoftware";
 allow (proxy) userdn= "ldap:///uid=MoneyWizAcctSoftware,ou=Applications,
dc=example,dc=com";)
```

正确使用此 ACI 之后，`MoneyWizAcctSoftware` 客户端应用程序即可绑定到目录，然后发送需要代理 DN 访问权限的 LDAP 命令，如 `ldapsearch` 或 `ldapmodify`。

在此示例中，如果客户端要执行 `ldapsearch` 命令，则此命令将包括以下控制：

```
$ ldapsearch -D "uid=MoneyWizAcctSoftware,ou=Applications,dc=example,dc=com" -w - \
 -y "uid=AcctAdministrator,ou=Administrators,dc=example,dc=com" ...
```

请注意，客户端以自身标识进行绑定，但被授予代理条目的特权。客户端不需要代理条目的密码。

使用过滤设置目标

如果要设置允许访问目录中大量条目的访问控制，则可以使用过滤器设置目标。

在 LDIF 中，要通过过滤器允许人力资源部门的所有用户访问员工条目，可编写以下语句：

```
aci: (targetattr="*") (targetfilter=(objectClass=employee))
 (version 3.0; acl "HR access to employees";
 allow (all) groupdn= "ldap:///cn=HRgroup,ou=People,dc=example,dc=com";)
```

此示例假定 ACI 已添加到 `ou=People,dc=example,dc=com` 条目中。

注 – 由于搜索过滤器不直接指出要管理访问权限的对象的名称，因此在允许或拒绝对象的访问权限时，请勿弄错对象。如果不慎允许或拒绝了错误对象的访问权限，则随着目录变得越来越复杂，风险将会越来越大。此外，使用过滤器时，您可能难以对目录中的访问控制问题进行故障排除。

为包含逗号的 DN 定义权限

在 LDIF ACI 语句中需要对包含逗号的 DN 进行特殊处理。在 ACI 语句的目标和绑定规则部分，必须使用单个反斜杠 (\) 来转义逗号。以下示例对此语法进行了说明：

```
dn: o=Example.com Bolivia\, S.A.
objectClass: top
objectClass: organization
aci: (target="ldap:///o=Example.com Bolivia\,S.A.") (targetattr="*")
(version 3.0; acl "aci 2"; allow (all) groupdn =
"ldap:///cn=Directory Administrators, o=Example.com Bolivia\, S.A.";)
```

查看有效权限

维护目录中条目的访问策略时，您需要了解所定义的 ACI 对安全性的影响。目录服务器允许您评估现有 ACI，方法是查看 ACI 针对给定条目为给定用户授予的有效权限。

目录服务器会对“获得有效的权限”控制做出响应，可在搜索操作中包含此控制。对此控制的响应是在搜索结果中返回有关条目和属性的有效权限信息。此额外信息包括每个条目及其每个属性的读写权限。可以为用于搜索的绑定 DN 或任意 DN 请求这些权限。此功能允许管理员测试目录用户的权限。

有效权限功能依赖于 LDAP 控制。必须确保用于绑定到远程服务器的代理标识也可以访问这些有效权限属性。

限制对“获得有效的权限”控制的访问权限

查看有效权限的操作是一种目录操作，必需对该操作进行保护和适当地限制。

要限制对有效权限信息的访问权限，请修改 `getEffectiveRights` 属性的默认 ACI。然后为 `getEffectiveRightsInfo` 属性创建新的 ACI。

例如，以下 ACI 只允许目录管理者组的成员获得有效权限：

```
aci: (targetattr != "aci")(version 3.0; acl
 "getEffectiveRights"; allow(all) groupdn =
 "ldap:///cn=Directory Administrators,ou=Groups,dc=example,dc=com";)
```

要获取有效权限信息，您必需具有使用有效权限控制的访问控制权限，**并且**具有对 `aclRights` 属性的读取权限。这种双层访问控制提供了可在必要时进一步微调的基本安全性。与代理类似，如果您对某个条目中的 `aclRights` 属性具有读取访问权限，则可以请求任何人对于该条目及其属性的权限信息。这表明管理资源的用户可以确定具有该资源权限的人员，即使此用户实际上并不管理具有这些权限的人员。

如果请求权限信息的用户没有使用有效权限控制的权限，则此操作将会失败，并返回错误消息。但是，如果请求权限信息的用户具有使用此控制的权限，但没有读取 `aclRights` 属性的权限，则 `aclRights` 属性不会出现在返回的条目中。此行为反映了目录服务器的常规搜索行为。

使用“获得有效的权限”控制

通过在 `ldapsearch` 命令中使用 `-J "1.3.6.1.4.1.42.2.27.9.5.2"` 选项可指定“获得有效的权限”控制。默认情况下，此控制将在搜索结果中返回绑定 DN 对条目和属性的有效权限。

可使用以下选项更改默认行为：

- `-c "dn:` *bind DN* `"` — 搜索结果将显示使用给定 *DN* 进行绑定的用户的有效权限。此选项允许管理员检查其他用户的有效权限。选项 `-c "dn:"` 显示匿名验证的有效权限。
- `-X "`*attributeName ...* `"` — 搜索结果还包括对指定属性的有效权限。使用此选项可指定不在搜索结果中显示的属性。例如，可以使用此选项确定用户是否有权添加当前不在条目中的属性。
- 使用 `-c` 和/或 `-X` 选项时，将自动包含 OID 为“获得有效的权限”控制的 `-J` 选项，无需另行指定 。如果为有效权限控制指定 NULL 值，则会检索当前用户的权限。此外，还会检索当前 `ldapsearch` 操作所返回的属性和条目的权限。

  接下来您必须选择要查看的信息类型。可以选择简单权限，也可以选择说明如何授予或拒绝这些权限的详细日志记录信息。通过分别添加 `aclRights` 或 `aclRightsInfo` 作为要在搜索结果中返回的属性，可以确定信息类型。您可以同时请求两个属性，以接收所有的有效权限信息，但实际上简单权限会在详细的日志记录信息中重复这些信息。

注 – `aclRights` 和 `aclRightsInfo` 属性的行为与虚拟操作属性类似。这些属性并未存储在目录中，如果未经明确请求，将不会返回这些属性。它们是由目录服务器在响应“获得有效的权限”控制时生成的。

因此，这些属性无法用在任何类型的过滤器或搜索操作中。

有效权限功能继承了影响访问控制的其他参数。这些参数包括时间、验证方法、计算机地址和名称。

以下示例说明用户 Carla Fuente 如何查看她在目录中的权限。在结果中，`1` 表示授予权限，`0` 表示拒绝权限。

```
$ ldapsearch -J "1.3.6.1.4.1.42.2.27.9.5.2 -h host1.Example.com -p 389 \
 -D "uid=cfuente,ou=People,dc=example,dc=com" -w - -b "dc=example,dc=com" \
 "(objectclass=*)" aclRights
Enter bind password:
dn: dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: ou=Groups, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: cn=Accounting Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: cn=HR Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: uid=bjensen,ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: uid=cfuente, ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:1,proxy:0
```

此结果为 Carla Fuente 显示了她在目录中至少具有读取权限的条目，并表明她可以修改自身的条目。有效权限控制不会避开普通访问权限，因此用户看不到他们没有读取权限的条目。在以下示例中，目录管理员可以看到 Carla Fuente 没有读取权限的条目：

```
$ ldapsearch -h host1.Example.com -p 389 -D cn=admin,cn=Administrators,cn=config -w - \
 -c "dn: uid=cfuente,ou=People,dc=example,dc=com" -b "dc=example,dc=com" \
 "(objectclass=*)" aclRights
Enter bind password:
dn: dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: ou=Groups, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: cn=Directory Administrators, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:0,write:0,proxy:0
```

```
dn: ou=Special Users,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:0,write:0,proxy:0
dn: ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: cn=Accounting Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: cn=HR Managers,ou=groups,dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: uid=bjensen,ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
dn: uid=cfuente, ou=People, dc=example,dc=com
aclRights;entryLevel: add:0,delete:0,read:1,write:1,proxy:0
```

在前面的输出中，目录管理员可以看出 Carla Fuente 甚至无法查看目录树的 Special Users 和 Directory Administrators 分支。在以下示例中，目录管理员可以看出 Carla Fuente 无法修改其自身条目中的 `mail` 和 `manager` 属性：

```
$ ldapsearch -h host1.Example.com -p 389 -D cn=admin,cn=Administrators,cn=config -w - \
 -c "dn: uid=cfuente,ou=People,dc=example,dc=com" -b "dc=example,dc=com" \
 "(uid=cfuente)" aclRights "*"
Enter bind password:
version: 1
dn: uid=cfuente, ou=People, dc=example,dc=com
aclRights;attributeLevel;mail: search:1,read:1,compare:1,
 write:0,selfwrite_add:0,selfwrite_delete:0,proxy:0
mail: cfuente@Example.com
aclRights;attributeLevel;uid: search:1,read:1,compare:1,
 write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
uid: cfuente
aclRights;attributeLevel;givenName: search:1,read:1,compare:1,
 write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
givenName: Carla
aclRights;attributeLevel;sn: search:1,read:1,compare:1,
 write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
sn: Fuente
aclRights;attributeLevel;cn: search:1,read:1,compare:1,
 write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
cn: Carla Fuente
aclRights;attributeLevel;userPassword: search:0,read:0,
 compare:0,write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
userPassword: {SSHA}wnbWHIq2HPiY/5ECwe6MWBGx2KMiZ8JmjF80Ow==
aclRights;attributeLevel;manager: search:1,read:1,compare:1,
 write:0,selfwrite_add:0,selfwrite_delete:0,proxy:0
manager: uid=bjensen,ou=People,dc=example,dc=com
aclRights;attributeLevel;telephoneNumber: search:1,read:1,compare:1,
 write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
telephoneNumber: (234) 555-7898
```

```
aclRights;attributeLevel;objectClass: search:1,read:1,compare:1,
 write:1,selfwrite_add:1,selfwrite_delete:1,proxy:0
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetorgperson
aclRights;entryLevel: add:0,delete:0,read:1,write:0,proxy:0
```

高级访问控制：使用宏 ACI

使用重复目录树结构的组织可以通过宏来优化目录中所使用的 ACI 的数量。如果减少目录树中的 ACI 数，将更容易管理访问控制策略。此外，还会提高 ACI 内存使用的效率。

宏是一种占位符，用于在 ACI 中表示 DN 或 DN 的一部分。可以使用宏来表示 ACI 目标部分和/或绑定规则部分中的 DN。实际上，当目录服务器收到传入的 LDAP 操作时，将根据 LDAP 操作的目标资源对 ACI 宏进行匹配。进行匹配的目的是为了找出匹配的子串（如果有）。如果存在匹配项，将使用匹配的子串扩展绑定规则端的宏，并通过评估已扩展的绑定规则来确定对资源的访问权限。

本部分包含宏 ACI 的示例和有关宏 ACI 语法的信息。

宏 ACI 示例

使用示例说明宏 ACI 的优点及其工作方式最为清楚。图 6–1 显示了一个目录树，在此目录树中，使用宏 ACI 可有效减少总体的 ACI 数。

请注意，在此图例中，相同的树结构 (`ou=groups,ou=people`) 具有重复的子域模式。此模式在整个树中也是重复的，因为 Example.com 目录树存储两个后缀 `dc=hostedCompany2,dc=example,dc=com` 和 `dc=hostedCompany3,dc=example,dc=com`（图中未显示）。

目录树中的 ACI 也具有重复的模式。例如，以下 ACI 位于 `dc=hostedCompany1,dc=example,dc=com` 节点上：

```
aci: (targetattr="*")
 (targetfilter=(objectClass=nsManagedDomain))(version 3.0;
 acl "Domain access"; allow (read,search) groupdn=
 "ldap:///cn=DomainAdmins,ou=Groups,dc=hostedCompany1,
 dc=example,dc=com";)
```

此 ACI 为 `domainAdmins` 组授予对 `dc=hostedCompany1,dc=example,dc=com` 树中任何条目的读取和搜索权限。

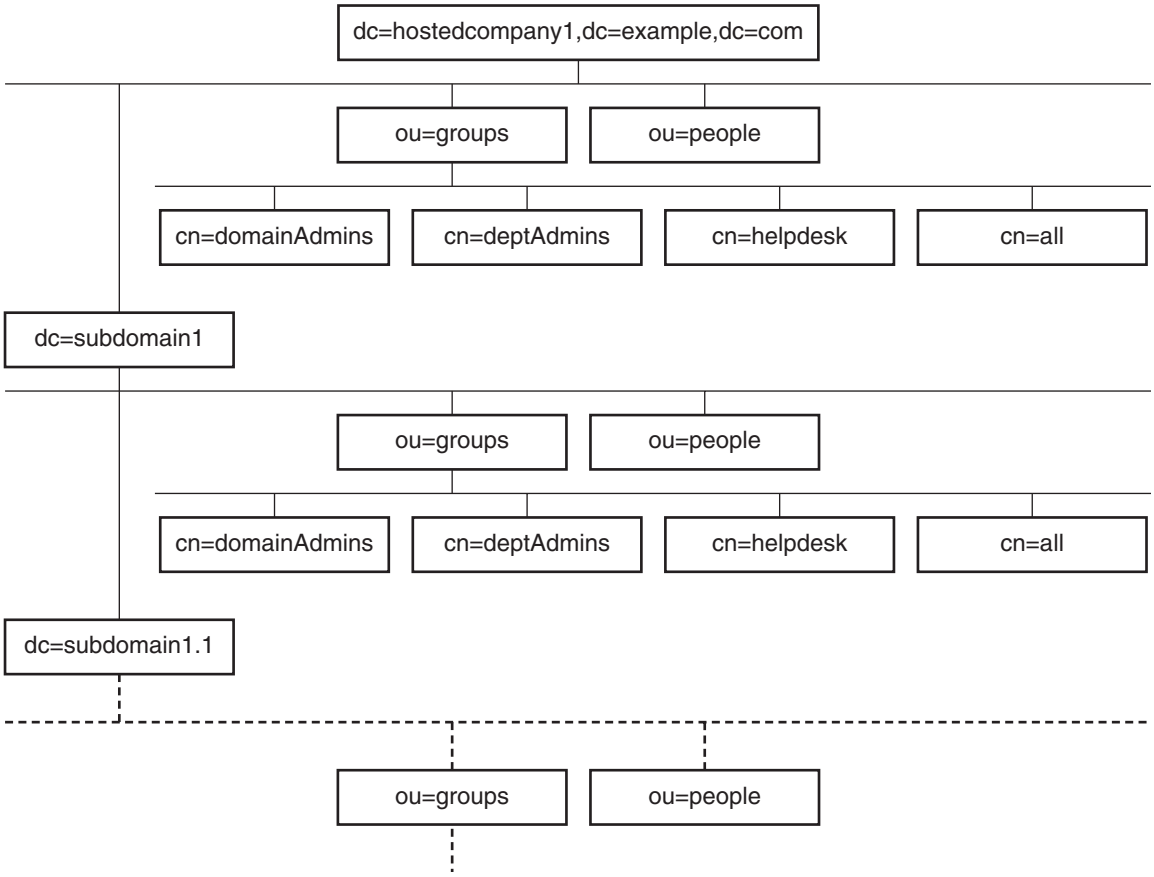


图 6–1 宏 ACI 的示例目录树

以下 ACI 位于 `dc=hostedCompany1,dc=example,dc=com` 节点上：

```
aci: (targetattr="*")
 (targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=hostedCompany1,dc=example,dc=com";)
```

以下 ACI 位于 `dc=subdomain1,dc=hostedCompany1, dc=example,dc=com` 节点上：

```
aci: (targetattr="*")
 (targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=subdomain1,dc=hostedCompany1,
  dc=example,dc=com";)
```

以下 ACI 位于 dc=hostedCompany2,dc=example,dc=com 节点上：

```
aci: (targetattr="*")
 (targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=hostedCompany2, dc=example,dc=com";)
```

以下 ACI 位于 dc=subdomain1,dc=hostedCompany2, dc=example,dc=com 节点上：

```
aci: (targetattr="*")
 (targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,dc=subdomain1,dc=hostedCompany2,
 dc=example,dc=com";)
```

在上述四个 ACI 中，唯一的不同之处是在 groupdn 关键字中指定的 DN。通过为 DN 使用宏，可以在树的根部（即 dc=example,dc=com 节点上）使用一个 ACI 替换这些 ACI。此宏 ACI 如下所示：

```
aci: (target="ldap:///ou=Groups,($dn),dc=example,dc=com")
 (targetattr="*")(targetfilter=(objectClass=nsManagedDomain))
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,[$dn],dc=example,dc=com";)
```

请注意，此时需要使用之前未使用的关键字 target。

在上述示例中，ACI 的数量从四个减少到一个。但其真正的好处取决于您在整个目录树中的重复模式数。

宏 ACI 语法

为了简化本部分讨论的内容，用于提供绑定凭证的 ACI 关键字（如 userdn、roledn、groupdn 和 userattr）将统称为 ACI 的**主题**。主题可确定应用 ACI 的对象。

下表显示可用于替换特定 ACI 关键字的宏。

表 6–1 宏 ACI 关键字

宏	描述	ACI 关键字
($dn)	用于在目标中进行匹配，并在主题中直接替换。	target、targetfilter、userdn、roledn、groupdn、userattr
[$dn]	用于替换在主题的子树中使用的多个 RDN。	targetfilter、userdn、roledn、groupdn、userattr

表 6–1 宏 ACI 关键字 （续）

宏	描述	ACI 关键字
($attr.*attrName*)	用于将目标条目中的 *attributeName* 属性值替换到主题中。	userdn、roledn、groupdn、userattr

宏 ACI 关键字具有以下限制：

- 在主题中使用 ($dn) 和 [$dn] 宏时，**必须**定义包含 ($dn) 宏的目标。
- 可以在主题中结合使用 ($dn) 宏（而非 [$dn] 宏）和 ($attr.*attrName*) 宏。

匹配目标中的 ($dn)

ACI 目标中的 ($dn) 宏通过比较自身与 LDAP 请求的目标条目来确定替换值。例如，您的 LDAP 请求将以下条目作为目标：

```
cn=all,ou=groups,dc=subdomain1, dc=hostedCompany1,dc=example,dc=com
```

此外，您还具有按如下方式定义目标的 ACI：

```
(target="ldap:///ou=Groups,($dn),dc=example,dc=com")
```

($dn) 宏与 "dc=subdomain1, dc=hostedCompany1" 相匹配。因此会将此子串用作 ACI 主题中的替换值。

在主题中替换 ($dn)

在 ACI 的主题中，($dn) 宏将被替换为目标中匹配的整个子串。例如：

```
groupdn="ldap:///cn=DomainAdmins,ou=Groups,($dn),dc=example,dc=com"
```

此主题将变为：

```
groupdn="ldap:///cn=DomainAdmins,ou=Groups,
 dc=subdomain1,dc=hostedCompany1,dc=example,dc=com"
```

扩展宏之后，目录服务器将在完成普通进程后评估 ACI，以确定是否授予访问权限。

注 – 与标准 ACI 不同，使用宏替换的 ACI 不一定会授予对目标条目的子条目的访问权限。这是因为当目标为子 DN 时，替换可能不会在主题字符串中创建有效 DN。

在主题中替换 [$dn]

[$dn] 的替换机制与 ($dn) 略有不同。将对目标资源的 DN 进行多次检查，每次都舍弃最左侧的 RDN 部分，直到找到匹配项为止。

例如，假定您的 LDAP 请求将 cn=all,ou=groups, dc=subdomain1,dc=hostedCompany1,dc=example,dc=com 子树作为目标，并具有以下 ACI：

```
aci: (targetattr="*")
 (target="ldap:///ou=Groups,($dn),dc=example,dc=com")
 (version 3.0; acl "Domain access"; allow (read,search)
 groupdn="ldap:///cn=DomainAdmins,ou=Groups,[$dn],
 dc=example,dc=com";)
```

服务器将按如下方式继续操作，以扩展此 ACI：

1. 服务器验证目标中的 ($dn) 是否与 dc=subdomain1,dc=hostedCompany1 相匹配。
2. 服务器将主题中的 [$dn] 替换为 dc=subdomain1,dc=hostedCompany1。

   得到的主题为 groupdn="ldap:///cn=DomainAdmins,ou=Groups, dc=subdomain1,dc=hostedCompany1,dc=example,dc=com"。如果因为绑定 DN 是该组的成员而授予访问权限，则宏扩展将停止，并对此 ACI 进行评估。如果绑定 DN 不是其成员，则此过程将继续。
3. 服务器将主题中的 [$dn] 替换为 dc=hostedCompany1。

   得到的主题为 groupdn="ldap:///cn=DomainAdmins,ou=Groups, dc=hostedCompany1,dc=example,dc=com"。再次将绑定 DN 作为此组的成员进行测试，如果是其成员，则对此 ACI 进行完全评估。如果此绑定 DN 不是其成员，宏扩展将在最后一个具有匹配值的 RDN 处停止，并且对此 ACI 的评估将结束。

[$dn] 宏的优点在于它提供了一种灵活的方法，可以为域级别管理员授予对目录树中**所有**子域的访问权限。因此，[$dn] 宏在表示域之间的层次关系方面非常有用。

例如，请考虑以下 ACI：

```
aci: (target="ldap:///ou=*,($dn),dc=example,dc=com") (targetattr="*")
(targetfilter=(objectClass=nsManagedDomain))
(version 3.0; acl "Domain access"; allow (read,search) groupdn=
"ldap:///cn=DomainAdmins,ou=Groups,[$dn],dc=example,dc=com";)
```

此 ACI 为 cn=DomainAdmins,ou=Groups, dc=hostedCompany1,dc=example,dc=com 的成员授予对 dc=hostedCompany1 下所有子域的访问权限。因此，属于该组的管理员可以访问 ou=people,dc=subdomain1.1,dc=subdomain1 等子树。

但同时将拒绝 cn=DomainAdmins,ou=Groups, dc=subdomain1.1 的成员访问 ou=people,dc=subdomain1, dc=hostedCompany1 和 ou=people,dc=hostedCompany1 节点。

($attr.*attrName*) 的宏匹配

将始终在 DN 的主题部分使用 ($attr.*attrname*) 宏。例如，您可以定义以下 roledn：

```
roledn = "ldap:///cn=DomainAdmins,($attr.ou),dc=HostedCompany1,dc=example,dc=com"
```

现在假定服务器收到了将以下条目作为目标的LDAP操作：

```
dn: cn=Babs Jensen,ou=People,dc=HostedCompany1,dc=example,dc=com
cn: Babs Jensen
sn: Jensen
ou: Sales
...
```

为了评估ACI的 `roledn` 部分，服务器会读取存储在目标条目中的 `ou` 属性值。然后，服务器将在主题中替换此值以扩展宏。在此示例中，`roledn` 将按如下方式扩展：

```
roledn = "ldap:///cn=DomainAdmins,ou=Sales,dc=HostedCompany1,dc=example,dc=com"
```

然后目录服务器将根据普通ACI评估算法来评估此ACI。

如果宏中指定的属性为多值属性，将依次使用每个值来扩展宏。将使用提供成功匹配的第一个值。

记录访问控制信息

要获取错误日志中的访问控制信息，必须设置相应的日志级别。

▼ 设置ACI的日志记录

无法使用DSCC执行此任务。请使用命令行，如以下过程所述。

- **设置日志级别，以便记录ACI处理信息。**

  ```
  $ dsconf set-log-prop -h host -p port error level:err-acl
  ```

通过TCP包装控制客户端-主机访问

可以使用TCP包装器，控制在TCP级别接受或拒绝连接的主机或IP地址。可以通过TCP包装限制客户端-主机访问。这样，您可以对目录服务器的初始TCP连接使用非主机式保护。

虽然可以为目录服务器设置TCP包装，但TCP包装可能会导致性能显著下降，特别是在拒绝服务攻击期间。要获取最佳性能，可以使用在目录服务器外部维护的基于主机的防火墙，或者使用IP端口过滤。

▼ 启用 TCP 包装

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1. **在实例路径中的某个位置创建 `hosts.allow` 文件或 `hosts.deny` 文件。**
   例如，在 *instance-path*`/config` 中创建此文件。请确保所创建的文件格式符合 `hosts_access`(4)。

2. **设置该访问文件的路径。**

   ```
   $ dsconf set-server-prop -h host -p port host-access-dir-path:path-to-file
   ```

   例如：

   ```
   $ dsconf set-server-prop -h host -p port host-access-dir-path:/local/ds1/config
   "host-access-dir-path" property has been set to "/local/ds1/config".
   The "/local/ds1/config" directory on host1 must contain valid hosts.allow
   and/or hosts.deny files.
   Directory Server must be restarted for changes to take effect.
   ```

▼ 禁用 TCP 包装

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

- **将主机访问路径设置为 `""`。**

  ```
  $ dsconf set-server-prop -h host -p port host-access-dir-path:""
  ```

◆ ◆ ◆ 第 7 章

7 目录服务器密码策略

用户连接到目录服务器时，系统将对该用户进行验证。根据验证期间建立的标识，目录可以为用户授予访问权限和资源限制。本章中的**帐户**一般指用户条目。帐户还反映用户在目录上执行操作的权限。在本章的密码策略讨论中，每个帐户都与用户条目和密码相关联。

本章还将介绍帐户激活（密码策略的一个方面）。目录管理者可以直接对帐户进行锁定和解除锁定，此操作独立于密码策略。

本章不包含验证方法。某些验证方法（如 SASL GSSAPI 和基于客户端 SSL 证书的验证）不需要使用密码。本章中与密码策略有关的信息不适用于此类验证方法。有关配置验证机制的说明，请参见第 5 章。

本章包含以下主题：

密码策略和工作单

本部分介绍密码策略设置，并提供一个工作单，以帮助您定义符合要求的密码策略。

注 – 要使用默认的密码策略，请参见第 162 页中的 “管理默认密码策略” 。

密码策略设置

在目录服务器中指定密码策略时，需要修改或创建包含对象类 pwdPolicy(5dsoc) 的条目。

为特定类型的用户定义密码策略时，需要考虑以下注意事项：

- 当入侵者看上去要尝试破解密码时如何锁定账户。

  有关详细信息，请参见第 158 页中的 “帐户锁定策略”。
- 如何更改密码。

  有关详细信息，请参见第 159 页中的 “密码更改策略”。
- 允许使用哪些密码值。

  有关详细信息，请参见第 159 页中的 “密码内容策略”。
- 如何处理密码过期。

  有关详细信息，请参见第 160 页中的 “密码过期策略”。
- 服务器是否记录上次成功验证的时间。

  请参见第 160 页中的 “跟踪上次验证时间的策略”。

本章的后续部分将介绍如何处理密码策略的这些方面。可以使用第 161 页中的 “用于定义密码策略的工作单”阐明要实现的每种密码策略。

帐户锁定策略

本部分介绍用于管理帐户锁定的策略属性。

目录服务器帐户一般指用户条目，以及用户在目录上执行操作的权限。每个帐户都与绑定 DN 和用户密码相关联。当入侵者看上去要尝试破解密码时，您希望目录服务器锁定帐户。锁定可阻止入侵者使用帐户进行绑定。锁定还可阻止入侵者继续进行攻击。

作为管理员，您还可以手动停用某个帐户或共享某个角色的所有用户的帐户。有关说明，请参见第 175 页中的 “手动锁定帐户”。但是，密码策略的一个重要部分就是指定目录服务器在什么情况下锁定帐户，而**不**需要您的干预。

首先，您必须指定目录服务器可以在发生太多失败绑定时使用 pwdLockout(5dsat) 自动锁定帐户。目录服务器会跟踪尝试绑定到帐户的连续失败次数。可以使用 pwdMaxFailure(5dsat) 指定在目录服务器锁定帐户之前所允许的连续失败次数。

目录服务器将严格按照密码策略锁定帐户。此操作完全为机械性操作。帐户锁定的原因可能不是入侵者对帐户发动攻击，而是用户键入了错误的密码。因此，可以使用 pwdFailureCountInterval(5dsat) 指定目录服务器在清除失败尝试记录之前等待下一次尝试的时间。可以使用 pwdLockoutDuration(5dsat) 指定在目录服务器自动对帐户解除锁定之前锁定持续的时间。如果用户并非出于恶意而犯下了合理错误，管理员无需介入帐户解除锁定。

如果在整个复制拓扑中复制用户数据，则会复制锁定计数器和锁定属性。pwdIsLockoutPrioritized(5dsat) 属性用于确定是否使用较高优先级复制锁定属性更新，因此在大多数情况下，应该将此属性保留为 TRUE。这样，用户在被锁定之前只能进行 pwdMaxFailure 次绑定到副本的尝试，当用户尝试绑定到更多副本时，尝试次数可能会更少。请参见《Sun Java System Directory Server Enterprise Edition 6.0 Deployment Planning Guide》中的“Preventing Authentication by Using Global Account Lockout”。包含全局锁定的文档将说明如何确保用户在进行 pwdMaxFailure 次尝试之后，才会在整个复制拓扑中将其锁定。

密码更改策略

本部分介绍用于管理密码更改的策略属性。

在许多部署中，目录服务器是标识数据的系统信息库。用户应该可以更改自己的密码（由 pwdAllowUserChange(5dsat) 指定），因此您无需更改密码。

允许用户更改自己的密码之后，您可能还希望控制用户可以在哪些情况下更改密码。可以使用 pwdSafeModify(5dsat) 指定要更改密码的用户必须首先提供正确的现有密码，然后才能替换密码。有关如何修改密码的示例，请参见第 173 页中的 “当 pwdSafeModify 为 TRUE 时从命令行修改密码”。可以使用 pwdInHistory(5dsat) 指定目录服务器记住的密码个数，以阻止用户重复使用相同的密码。也可以通过设置 pwdMinAge(5dsat) 来阻止用户过于频繁地更改密码。

在许多情况下，您（管理员）或者您管理的应用程序会在目录中创建用户条目。您可以指定在用户首次绑定到新帐户时要更改的用户密码值。您可能还必须重置用户密码，重置密码后，用户在下次使用该帐户时应该更改密码。目录服务器具有特定属性 pwdMustChange(5dsat)，可以使用该属性指示当其他用户重置密码值后，用户是否必须更改密码。

还可以指定当目录管理者通过设置 passwordRootdnMayBypassModsChecks(5dsat) 更改密码时，可以不必遵循策略。

密码内容策略

本部分介绍用于管理密码内容的策略属性。

虽然一般不会在目录搜索中返回密码值，但攻击者仍有可能获取对目录数据库的访问权限。因此，密码值一般以某种受支持的散列格式（使用 passwordStorageScheme(5dsat) 指定）存储。

可以通过设置 pwdMinLength(5dsat) 来强制密码至少包含指定数目的字符。

还可以通过设置 pwdCheckQuality(5dsat) 来强制检查密码是否满足最低密码质量的定义。执行检查时，目录服务器将检查密码是否至少达到最小长度。服务器也会检查密码是否不包含任何 cn、givenName、mail, ou、sn 或 uid 属性值。此外，启用严格密码检查插件时，目录服务器还会检查密码是否不包含该插件所使用的字典文件中的字符串。服务器也会检查密码是否包含不同类型字符的正确组合。

可以使用 dsconf set-server-prop 命令启用严格密码检查。可以使用 pwd-strong-check-enabled 属性打开插件，然后重新启动服务器以使更改生效。可以使用 pwd-strong-check-require-charset 属性指定密码所需的字符集。pwd-strong-check-require-charset 属性使用以下值的掩码：

lower	新密码必须包含小写字符。
upper	新密码必须包含大写字符。
digit	新密码必须包含数字。
special	新密码必须包含特殊字符。
any-two	新密码必须至少包含上述字符集中的两种，每种至少包含一个字符。
any-three	新密码必须至少包含上述字符集中的三种，每种至少包含一个字符。

pwd-strong-check-require-charset 属性的默认设置为 lower && upper && digit && special。

密码过期策略

本部分介绍用于管理密码过期的策略属性。

要确保用户定期更改密码，可以通过设置 pwdMaxAge(5dsat)，将目录服务器配置为当密码达到特定存留期后将密码设置为过期。

必须通知用户其密码即将过期。可以将目录服务器配置为返回一个警告，指明用于绑定的密码即将过期。可以使用 pwdExpireWarning(5dsat) 定义在过期之前多久将会在客户端进行绑定时返回警告。**请注意，客户端应用程序将收到该警告。用户不会直接收到警告。**客户端应用程序在收到密码即将过期的警告时必须通知最终用户。

通过设置 pwdGraceAuthNLimit(5dsat)，可允许用户使用过期密码进行一次或多次绑定尝试。因此，未能及时更改密码的用户仍可以进行绑定以更改密码。请注意，当用户使用宽限登录进行绑定时，该用户可以执行任何操作。宽限登录的工作方式就像密码尚未过期一样。

每次修改条目上的密码时，目录服务器都会更新操作属性 pwdChangedTime(5dsat)。因此，如果您准备启用密码过期，则已超过存留期限的用户密码会在您启用密码过期后立即过期。如果您不希望发生这种情况，请使用警告和宽限登录。

跟踪上次验证时间的策略

本部分介绍密码策略属性 pwdKeepLastAuthTime(5dsat) 的使用。

设置 pwdKeepLastAuthTime 之后，目录服务器将在每次用户验证时跟踪上次成功绑定的时间。此时间记录在用户条目的 pwdLastAuthTime(5dsat) 操作属性上。

由于此行为会为每次成功的绑定操作添加更新，因此在默认情况下不会激活 pwdKeepLastAuthTime 功能。您必须明确打开此功能才能在部署中使用。

用于定义密码策略的工作单

此工作单旨在帮助您通过命令行界面或使用目录服务控制中心 (Directory Service Control Center, DSCC) 定义要实现的密码策略。请为每个密码策略使用一个工作单。

记录密码策略条目的 DN 之后，请记录与每个策略范围中的属性设置有关的决策。另外，请记录使用这些设置的理由。

密码策略工作单			
密码策略条目标识名			
dn: cn=			

策略范围	属性	在此处填写您的设置	在此处填写使用这些设置的理由
帐户锁定	pwdFailureCountInterval(5dsat)		
	pwdIsLockoutPrioritized(5dsat)		
	pwdLockout(5dsat)		
	pwdLockoutDuration(5dsat)		
	pwdMaxFailure(5dsat)		
密码更改	passwordRootdnMayBypassModsChecks(5dsat)		
	pwdAllowUserChange(5dsat)		
	pwdInHistory(5dsat)		
	pwdMinAge(5dsat)		
	pwdMustChange(5dsat)		
	pwdSafeModify(5dsat)		
密码内容	passwordStorageScheme(5dsat)		
	pwdCheckQuality(5dsat)		
	pwdMinLength(5dsat)		

策略范围	属性	在此处填写您的设置	在此处填写使用这些设置的理由
密码过期	pwdExpireWarning(5dsat)		
	pwdGraceAuthNLimit(5dsat)		
	pwdMaxAge(5dsat)		
跟踪上次验证时间	pwdKeepLastAuthTime(5dsat)		

注 – 将 pwdCheckQuality 属性设置为 2 时，服务器可以执行其他检查。如果还启用了密码检查插件，则该插件的设置将影响对新密码值执行哪些检查。

管理默认密码策略

默认密码策略将应用于目录实例中未定义专用策略的所有用户。但是，默认密码策略不会应用于目录管理员。有关策略范围的详细信息，请参见第 165 页中的 “应用哪个密码策略” 。

默认密码策略是可以使用 dsconf 命令进行配置的策略。还可以通过读取 cn=Password Policy,cn=config 查看默认密码策略。

本部分显示了每个策略范围的策略属性以及相关的 dsconf 服务器属性。此外，还介绍了如何查看和更改默认密码策略设置。

密码策略属性和 dsconf 服务器属性之间的关联

下表显示了每个密码策略范围的密码策略属性和相关的 dsconf 服务器属性。

策略范围	策略属性	dsconf 服务器属性
帐户锁定	pwdFailureCountInterval	pwd-failure-count-interval
	pwdLockout	pwd-lockout-enabled
	pwdLockoutDuration	pwd-lockout-duration
	pwdMaxFailure	pwd-max-failure-count

策略范围	策略属性	dsconf 服务器属性
密码更改	passwordRootdnMayBypassModsChecks	pwd-root-dn-bypass-enabled
	pwdAllowUserChange	pwd-user-change-enabled
	pwdInHistory	pwd-max-history-count
	pwdMinAge	pwd-min-age
	pwdMustChange	pwd-must-change-enabled
	pwdSafeModify	pwd-safe-modify-enabled
密码内容	pwdCheckQuality	pwd-check-enabled、 pwd-accept-hashed-password-enabled、 pwd-strong-check-dictionary-path、 pwd-strong-check-enabled、 pwd-strong-check-require-charset
	pwdMinLength	pwd-min-length
	passwordStorageScheme	pwd-storage-scheme
密码过期	pwdExpireWarning	pwd-expire-warning-delay
	pwdGraceAuthNLimit	pwd-grace-login-limit
	pwdMaxAge	pwd-max-age
跟踪上次验证时间	pwdKeepLastAuthTime	pwd-keep-last-auth-time-enabled

注 – 与 pwdCheckQuality 相关的属性可用于配置密码检查插件。因此，这五种属性适用于整个服务器实例，从而也适用于包含 pwdCheckQuality: 2 的其他密码策略。

▼ 查看默认密码策略设置

可以使用 dsconf 命令查看默认密码策略设置。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

- **读取默认密码策略配置。**

```
$ dsconf get-server-prop -h host -p port | grep ^pwd-
pwd-accept-hashed-pwd-enabled      :  N/A
pwd-check-enabled                  :  off
pwd-compat-mode                    :  DS5-compatible-mode
pwd-expire-no-warning-enabled      :  on
pwd-expire-warning-delay           :  1d
```

```
pwd-failure-count-interval         :  10m
pwd-grace-login-limit              :  disabled
pwd-keep-last-auth-time-enabled    :  off
pwd-lockout-duration               :  1h
pwd-lockout-enabled                :  off
pwd-lockout-repl-priority-enabled  :  on
pwd-max-age                        :  disabled
pwd-max-failure-count              :  3
pwd-max-history-count              :  disabled
pwd-min-age                        :  disabled
pwd-min-length                     :  6
pwd-mod-gen-length                 :  6
pwd-must-change-enabled            :  off
pwd-root-dn-bypass-enabled         :  off
pwd-safe-modify-enabled            :  off
pwd-storage-scheme                 :  SSHA
pwd-strong-check-dictionary-path   :  /local/ds6/plugins/words-english-big.txt
pwd-strong-check-enabled           :  off
pwd-strong-check-require-charset   :  lower
pwd-strong-check-require-charset   :  upper
pwd-strong-check-require-charset   :  digit
pwd-strong-check-require-charset   :  special
pwd-supported-storage-scheme       :  CRYPT
pwd-supported-storage-scheme       :  SHA
pwd-supported-storage-scheme       :  SSHA
pwd-supported-storage-scheme       :  NS-MTA-MD5
pwd-supported-storage-scheme       :  CLEAR
pwd-user-change-enabled            :  on
```

▼ 更改默认密码策略设置

可以通过使用 `dsconf` 命令设置服务器属性来更改默认密码策略。

注 – 在完成此过程之前，请阅读并完成第 161 页中的 “用于定义密码策略的工作单”。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1. **将工作单中的设置转换为 `dsconf` 命令属性设置。**

2. **使用 `dsconf set-server-prop` 命令适当地更改默认密码策略属性。**
   例如，以下命令允许目录管理员在修改密码时违反默认策略：

   ```
   $ dsconf set-server-prop -h host -p port pwd-root-dn-bypass-enabled:on
   ```

管理专用密码策略

专用密码策略是在 pwdPolicy(5dsoc) 条目中定义的。可以在目录树中的任意位置定义策略，通常在使用策略所管理的帐户复制的子树中定义。策略的 DN 采用 cn=*policy name*,*subtree* 格式。

定义密码策略之后，可以通过在所需的用户条目中设置 pwdPolicySubentry(5dsat) 属性来指定该密码策略。

本部分包含以下主题：

应用哪个密码策略

目录服务器允许您配置多个密码策略。本部分介绍默认密码策略和专用密码策略。本部分还介绍当多个密码策略可应用于给定帐户时应执行哪个策略。

首次创建目录服务器实例时，该实例有一个默认密码策略。默认密码策略在配置条目 cn=PasswordPolicy,cn=config 中表示。默认密码策略将应用于目录中除目录管理员之外的所有帐户。

如同在所有目录服务器密码策略中一样，cn=PasswordPolicy,cn=config 具有对象类 pwdPolicy(5dsoc)。

注 – 创建目录服务器实例时，密码策略属性仍处于 Directory Server 5 兼容模式，以便从早期版本进行升级。在 Directory Server 5 兼容模式下，目录服务器还会处理具有对象类 passwordPolicy(5dsoc) 的密码策略条目。

完成升级之后，您可以在完整功能模式下使用新的密码策略，如《Sun Java System Directory Server Enterprise Edition 6.0 Migration Guide》中所述。管理上的变化对目录应用程序没有任何影响。

本章介绍使用新密码策略功能的密码策略配置。

可以更改默认密码策略以覆盖默认设置。可以使用 dsconf(1M) 命令设置默认密码策略的服务器属性。通常，此类服务器属性的名称都以 pwd- 前缀开头。更改此类属性的设置时，将覆盖实例的默认密码策略。但是，复制操作不会复制对副本所做的更改。对默认密码策略所做的更改是实例配置的一部分，而不是目录数据的一部分。

除了配置默认密码策略以外，还可以配置**专用密码策略**。专用密码策略是由目录树中的条目定义的。专用密码策略条目与默认密码策略具有相同的对象类 pwdPolicy(5dsoc)，因此将使用相同的策略属性。由于专用密码策略是常规的目录条目，因此策略条目的复制方式与常规目录条目的复制方式相同。

用户条目可通过操作属性 pwdPolicySubentry(5dsat) 的值来引用专用密码策略。用户条目引用专用密码策略时，该专用密码策略将覆盖实例的默认密码策略。在许多部署中，您需要指定用户角色。通过设置 pwdPolicySubentry 值，可以将角色配置为与服务类 (Class of Service, CoS) 一起使用，以确定应用于用户帐户的密码策略。要覆盖由角色设置的密码策略，请直接在该用户的条目上更改 pwdPolicySubentry 值。

下面对本部分内容进行一下总结：最初将应用默认密码策略。您可以更改默认密码策略以覆盖默认值。然后，您可以创建专用密码策略条目以覆盖默认密码策略。使用角色和 CoS 指定密码策略时，可以通过为单个条目指定密码策略来覆盖 CoS 指定的策略。

▼ 创建密码策略

可以使用与创建和修改其他目录条目相同的方式来创建和修改专用密码策略。以下过程说明如何使用文本编辑器在 LDIF 中编写密码策略条目。接下来，可以在 ldapmodify 命令中使用 -a 选项将该密码策略条目添加到目录。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

开始之前 如果没有特殊说明，此处显示的示例数据均来自 Example.ldif。

1. **完成要创建的策略的密码策略工作单。**
   第 161 页中的 “用于定义密码策略的工作单”中提供了样例。

2. **根据工作单，在 LDIF 中编写密码策略条目。**
   例如，以下策略条目将为 Example.com 的临时员工指定密码策略，Example.com 的子树根为 dc=example,dc=com：

```
dn: cn=TempPolicy,dc=example,dc=com
objectClass: top
objectClass: pwdPolicy
objectClass: LDAPsubentry
cn: TempPolicy
pwdCheckQuality: 2
pwdLockout: TRUE
pwdLockoutDuration: 300
pwdMaxFailure: 3
pwdMustChange: TRUE
```

除了默认密码策略设置之外，此处显示的策略还指定其他行为。系统将执行密码质量检查。连续出现三次失败绑定之后，帐户将被锁定五分钟（300 秒）。重置密码后必须更改密码。为用户帐户指定策略后，此处**明确**指定的设置将覆盖默认密码策略。

3 **将密码策略条目添加到目录。**

例如，以下命令将为 `dc=example,dc=com` 下的 Example.com 临时员工添加密码策略。该密码策略已保存到名为 `pwp.ldif` 的文件中。

```
$ ldapmodify -a -D uid=kvaughan,ou=people,dc=example,dc=com -w - -f pwp.ldif
Enter bind password:
adding new entry cn=TempPolicy,dc=example,dc=com

$ ldapsearch -D uid=kvaughan,ou=people,dc=example,dc=com -w --b dc=example,dc=com \
"(&(objectclass=ldapsubentry)(cn=temppolicy))"
Enter bind password:
version: 1
dn: cn=TempPolicy,dc=example,dc=com
objectClass: top
objectClass: pwdPolicy
objectClass: LDAPsubentry
cn: TempPolicy
pwdCheckQuality: 2
pwdLockout: TRUE
pwdLockoutDuration: 300
pwdMaxFailure: 3
pwdMustChange: TRUE
$
```

如 `Example.ldif` 中所示，`kvaughan` 是具有 `dc=example,dc=com` 条目修改权限的人力资源经理。Vaughan 的绑定密码（如 `Example.ldif` 中所示）为 `bribery`。

另请参见 要指定您所定义的策略将要管理的用户帐户，请参见第 167 页中的 “为单个帐户指定密码策略”或第 168 页中的 “使用角色和 CoS 指定密码策略”。

▼ 为单个帐户指定密码策略

此过程将为单个用户帐户指定现有的密码策略。

注 – 要完成此过程，您必须具有要指定的专用密码策略。请参见第 166 页中的 “创建密码策略”。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

如果没有特殊说明，此处显示的示例数据均来自 `Example.ldif`。

- **将密码策略 DN 添加到用户条目的 pwdPolicySubentry 属性值。**

  例如，以下命令将第 166 页中的 “创建密码策略”中定义的密码策略指定给 David Miller 条目（其 DN 为 uid=dmiller,ou=people,dc=example,dc=com）：

```
$ cat pwp.ldif
dn: uid=dmiller,ou=people,dc=example,dc=com
changetype: modify
add: pwdPolicySubentry
pwdPolicySubentry: cn=TempPolicy,dc=example,dc=com

$ ldapmodify -D uid=kvaughan,ou=people,dc=example,dc=com -w - -f pwp.ldif
Enter bind password:
modifying entry uid=dmiller,ou=people,dc=example,dc=com

$ ldapsearch -D uid=kvaughan,ou=people,dc=example,dc=com -w - -b dc=example,dc=com \
"(uid=dmiller)" pwdPolicySubentry
Enter bind password:
version: 1
dn: uid=dmiller, ou=People, dc=example,dc=com
pwdPolicySubentry: cn=TempPolicy,dc=example,dc=com
$
```

  如 Example.ldif 中所示，kvaughan 是具有 dc=example,dc=com 条目修改权限的人力资源经理。Vaughan 的绑定密码（如 Example.ldif 中所示）为 bribery。

▼ 使用角色和 CoS 指定密码策略

此过程通过应用角色和服务类 (Class of Service, CoS) 为一组用户指定现有的专用密码策略。有关角色和 CoS 的详细信息，请参见第 9 章。

注 – 要完成此过程，您必须具有要指定的专用密码策略。请参见第 166 页中的 “创建密码策略”。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

如果没有特殊说明，此处显示的示例数据均来自 Example.ldif。

1. **为要由密码策略管理的条目创建角色。**

   例如，以下命令将为 Example.com 的临时员工创建过滤角色。

```
$ cat tmp.ldif
dn: cn=TempFilter,ou=people,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
```

```
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsFilteredRoleDefinition
cn: TempFilter
nsRoleFilter: (&(objectclass=person)(status=contractor))
description: filtered role for temporary employees

$ ldapmodify -a -D uid=kvaughan,ou=people,dc=example,dc=com -w - -f tmp.ldif
Enter bind password:
modifying entry cn=TempFilter,ou=people,dc=example,dc=com

$
```

如 `Example.ldif` 中所示，`kvaughan` 是具有 `dc=example,dc=com` 条目修改权限的人力资源经理。Vaughan 的绑定密码（如 `Example.ldif` 中所示）为 `bribery`。

2 创建用于生成密码策略条目 DN 的服务类。

此 DN 是用户（具有您所创建的角色）的 `pwdPolicySubentry` 属性值。

例如，以下命令将为 Example.com 的临时员工创建过滤角色。这些命令将为具有此角色的用户指定 `cn=TempPolicy,dc=example,dc=com`。

```
$ cat cos.ldif
dn: cn=PolTempl,dc=example,dc=com
objectclass: top
objectclass: nsContainer

dn: cn="cn=TempFilter,ou=people,dc=example,dc=com",
 cn=PolTempl,dc=example,dc=com
objectclass: top
objectclass: extensibleObject
objectclass: LDAPsubentry
objectclass: costemplate
cosPriority: 1
pwdPolicySubentry: cn=TempPolicy,dc=example,dc=com

dn: cn=PolCoS,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosClassicDefinition
cosTemplateDN: cn=PolTempl,dc=example,dc=com
cosSpecifier: nsRole
cosAttribute: pwdPolicySubentry operational

$ ldapmodify -a -D uid=kvaughan,ou=people,dc=example,dc=com -w - -f cos.ldif
Enter bind password:
modifying entry cn=TempFilter,ou=people,dc=example,dc=com

$
```

状态为 contractor 的用户现在要遵循密码策略 cn=TempPolicy,dc=example,dc=com。

▼ 设置首次登录密码策略

在许多部署中，应用于新帐户的密码策略与应用于已建帐户的密码策略不同。本部分介绍首次登录密码策略。此策略为用户提供三天时间使用新建帐户并设置新密码，之后将锁定该帐户。对于已重置密码的用户而言，此策略的工作方式相同。

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1 为新建帐户创建专用密码策略。

例如， 添加一个将过期时间设置为三天（259,200 秒）的密码策略条目。此密码策略还将 pwdMustChange(5dsat) 设置为 TRUE，这意味着用户在首次绑定时必须更改其密码。

```
$ cat firstLogin.ldif
dn: cn=First Login,dc=example,dc=com
objectClass: top
objectClass: LDAPsubentry
objectClass: pwdPolicy
objectClass: sunPwdPolicy
cn: First Login
passwordStorageScheme: SSHA
pwdAttribute: userPassword
pwdInHistory: 0
pwdExpireWarning: 86400
pwdLockout: TRUE
pwdMinLength: 6
pwdMaxFailure: 3
pwdMaxAge: 259200
pwdFailureCountInterval: 600
pwdAllowUserChange: TRUE
pwdLockoutDuration: 3600
pwdMinAge: 0
pwdCheckQuality: 2
pwdMustChange: TRUE

$ ldapmodify -a -D cn=admin,cn=Administrators,cn=config -w - -f firstLogin.ldif
Enter bind password:
adding new entry cn=First Login,dc=example,dc=com

$
```

2 创建包含所有新建帐户的角色。

创建此角色时，将设置用于区分新建帐户和已建帐户的方法。

a. 将新帐户定义为 pwdReset(5dsat) 属性设置为 TRUE 的帐户。

当其他用户（如密码管理员）更改该用户的密码时，pwdReset 将被设置为 TRUE。

b. 创建用于标识新帐户的角色。

例如，以下命令将为已重置密码的帐户创建角色。

```
$ cat newRole.ldif
dn: cn=First Login Role,ou=people,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsFilteredRoleDefinition
cn: First Login Role
nsRoleFilter: (pwdReset=TRUE)
description: Role to assign password policy for new and reset accounts

$ ldapmodify -a -D uid=kvaughan,ou=people,dc=example,dc=com -w - -f newRole.ldif
Enter bind password:
adding new entry cn=First Login Role,ou=people,dc=example,dc=com

$
```

3 使用服务类为新建帐户指定密码策略。

```
$ cat newCoS.ldif
dn: cn=First Login Template,dc=example,dc=com
objectClass: top
objectClass: nsContainer

dn: cn="cn=First Login Role,ou=people,dc=example,dc=com",
 cn=First Login Template,dc=example,dc=com
objectClass: top
objectClass: extensibleObject
objectClass: LDAPSubEntry
objectClass: CoSTemplate
cosPriority: 1
pwdPolicySubentry: cn=First Login,dc=example,dc=com

dn: cn=First Login CoS,dc=example,dc=com
objectClass: top
objectClass: LDAPSubEntry
objectClass: CoSSuperDefinition
objectClass: CoSClassicDefinition
cosTemplateDN: cn=First Login Template,dc=example,dc=com
```

```
cosSpecifier: nsRole
cosAttribute: pwdPolicySubentry operational

$ ldapmodify -a -D uid=kvaughan,ou=people,dc=example,dc=com -f newCoS.ldif
Enter bind password:
adding new entry cn=First Login Template,dc=example,dc=com

adding new entry cn="cn=First Login Role,ou=people,dc=example,dc=com",
 cn=First Login Template,dc=example,dc=com

adding new entry cn=First Login CoS,dc=example,dc=com

$
```

示例 7–1 检查密码策略指定

请添加与所添加的角色相符的新用户。添加用户是为了验证新用户是否遵循新的密码策略，而现有用户不会遵循该策略。

```
$ cat quentin.ldif
dn: uid=qcubbins,ou=People,dc=example,dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
uid: qcubbins
givenName: Quentin
sn: Cubbins
cn: Quentin Cubbins
mail: quentin.cubbins@example.com
userPassword: ch4ngeM3!
description: New account

$ ldapmodify -a -D uid=kvaughan,ou=people,dc=example,dc=com -w - -f quentin.ldif
Enter bind password:
adding new entry uid=qcubbins,ou=People,dc=example,dc=com

$ ldapsearch -D uid=kvaughan,ou=people,dc=example,dc=com -w - \
-b dc=example,dc=com uid=qcubbins nsrole pwdPolicySubentry
Enter bind password:
version: 1
dn: uid=qcubbins,ou=People,dc=example,dc=com
nsrole: cn=first login role,ou=people,dc=example,dc=com
pwdPolicySubentry: cn=First Login,dc=example,dc=com
$ ldapsearch -b dc=example,dc=com uid=bjensen nsrole pwdPolicySubentry
version: 1
dn: uid=bjensen, ou=People, dc=example,dc=com
$
```

请注意，Barbara Jensen 的现有帐户由默认密码策略管理。但是，Quentin Cubbins 的新帐户将由您定义的密码策略管理。

当 pwdSafeModify 为 TRUE 时从命令行修改密码

用户的密码策略将 pwdSafeModify 设置为 TRUE 时，必须同时提供旧密码和新密码才能更改密码。命令 dsconf set-server-prop pwd-safe-modify-enabled:on 对默认密码策略具有相同的效果。

可以使用 ldappasswd(1) 命令更改密码。此命令支持安全密码修改。此命令将执行 RFC 3062（LDAP Password Modify Extended Operation (http://www.ietf.org/rfc/rfc3062.txt)）

可以使用 ldapmodify(1) 命令更改密码。此时向 ldapmodify 命令传递的 LDIF 应如下所示：

```
dn: DN of user whose password you are changing
changetype: modify
delete: userPassword
userPassword: old password
-
add: userPassword
userPassword: new password
```

还可以使用 LDAP 密码修改扩展操作。第 174 页中的 “使用密码修改扩展操作重置密码” 中介绍了如何设置扩展操作支持。

重置已过期的密码

密码策略执行密码过期时，某些用户可能未及时更改密码。本部分说明如何更改已过期的密码。

注 – 每次修改条目上的密码时，目录服务器都会更新操作属性 pwdChangedTime(5dsat)。因此，如果您准备启用密码过期，则已超过存留期限的用户密码会在您启用密码过期后立即过期。如果您不希望发生这种情况，请使用警告和宽限登录。

本部分包含使用密码修改扩展操作重置密码以及在密码过期时允许宽限验证的过程。

本部分介绍的机制可供管理员使用，或者供处理用户与目录之间实际交互的应用程序使用。通常，您需要依赖于应用程序来确保最终用户使用机制的方式实际上与预期相同。

▼ 使用密码修改扩展操作重置密码

密码过期时用户帐户将被锁定。重置密码时，将对帐户解除锁定。其他用户（如管理员）可以重置密码。重置密码后，目录服务器将对用户帐户解除锁定。目录服务器提供 RFC 3062（LDAP Password Modify Extended Operation (http://www.ietf.org/rfc/rfc3062.txt)）支持。可以使用扩展操作允许目录管理者或目录应用程序通过密码重置对帐户解除锁定。

允许使用密码修改扩展操作时应特别谨慎，如以下过程所示。请仅为您所信任的管理员和应用程序授予访问权限。请勿以明文形式在网络中传送密码。

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1. **为用户授予对密码管理员或密码管理应用程序的访问权限。**

2. **允许密码管理员使用密码修改扩展操作。**

   以下命令将设置一个 ACI，以允许**密码管理员**角色的成员在通过 SSL 进行连接时使用密码修改扩展操作：

   ```
   $ cat exop.ldif
   dn: oid=1.3.6.1.4.1.4203.1.11.1,cn=features,cn=config
   objectClass: top
   objectClass: directoryServerFeature
   oid: 1.3.6.1.4.1.4203.1.11.1
   cn: Password Modify Extended Operation
   aci: (targetattr != "aci")(version 3.0;
    acl "Password Modify Extended Operation
    "; allow( read, search, compare, proxy ) (roledn = "
    ldap:///cn=Password Managers,dc=example,dc=com" and authmethod = "SSL");)

   $ ldapmodify -a -D cn=admin,cn=Administrators,cn=config -w - -f exop.ldif
   Enter bind password:
   adding new entry oid=1.3.6.1.4.1.4203.1.11.1,cn=features,cn=config

   $
   ```

   `cn=features,cn=config` 下的条目允许您管理对特定操作（使用密码修改扩展操作）的访问权限。

3. **让密码管理员重置用户密码。**

   此步骤将对用户帐户解除锁定，可以使用 `ldappasswd`(1) 命令完成此步骤。

4. **（可选的）如果用户必须更改密码，请让密码管理员通知用户。**

   如果管理用户条目的密码策略包含 `pwdMustChange: TRUE`，则用户必须在重置密码后更改其密码。

▼ 在密码过期时允许宽限验证

此过程介绍如何为用户提供宽限验证，以允许用户更改已过期的密码。

宽限验证应该由处理密码策略请求和响应控制的应用程序进行管理。

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1. **确保用户可以访问使用密码策略请求和响应控制的应用程序。**
   应用程序应确保用户正确处理宽限验证。

2. **允许应用程序使用密码策略控制。**
   以下命令将设置一个 ACI，以允许**密码管理员**角色的成员使用密码策略控制：

```
$ cat ctrl.ldif
dn: oid=1.3.6.1.4.1.42.2.27.8.5.1,cn=features,cn=config
objectClass: top
objectClass: directoryServerFeature
oid: 1.3.6.1.4.1.42.2.27.8.5.1
cn: Password Policy Controls
aci: (targetattr != "aci")(version 3.0; acl "Password Policy Controls
 "; allow( read, search, compare, proxy ) roledn = "
 ldap:///cn=Password Managers,dc=example,dc=com";)

$ ldapmodify -a -D cn=admin,cn=Administrators,cn=config -w - -f ctrl.ldif
Enter bind password:
adding new entry oid=1.3.6.1.4.1.42.2.27.8.5.1,cn=features,cn=config

$
```

   `cn=features,cn=config` 下条目的唯一用途就是允许您管理对特定操作（使用密码策略请求和响应控制）的访问权限。

3. **将密码策略中的 `pwdGraceAuthNLimit` 设置为密码过期后允许的验证次数。**

4. **确保应用程序能指导最终用户在宽限验证失效之前正确更改已过期的密码。**

手动锁定帐户

目录服务器允许您将密码策略配置为在进行指定次数的失败绑定尝试后锁定帐户。有关详细信息，请参见第 158 页中的 “帐户锁定策略”。本部分介绍目录管理员可以使用的手动帐户锁定和激活工具。

目录管理员可以在不使用锁定持续时间计时器的情况下管理帐户锁定。在手动重置密码之前，锁定的帐户将保持锁定状态。目录管理员还可以无限期地停用某些帐户。

本部分说明如何检查帐户状态、停用帐户以及重新激活帐户。

▼ 检查帐户状态

请按以下所述检查帐户状态。

注 – 您必须以目录管理员身份进行绑定。

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

- **使用 `ns-accountstatus` 命令检查帐户或角色的状态。**
  以下命令将检查 Barbara Jensen 的帐户状态：

```
$ ns-accountstatus -D "cn=Directory Manager" -j pwd.txt \
 -I uid=bjensen,ou=people,dc=example,dc=com
uid=bjensen,ou=people,dc=example,dc=com activated.
$
```

  有关详细信息，请参见`ns-accountstatus`(1M) 手册页。

▼ 停用帐户

请按以下所述停用帐户或角色。

注 – 您必须以目录管理员身份进行绑定。

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

- **使用 `ns-inactivate` 命令停用帐户或角色。**
  以下命令将停用 Barbara Jensen 的帐户：

```
$ ns-inactivate -D "cn=Directory Manager" -j pwd.txt \
-I uid=bjensen,ou=people,dc=example,dc=com
uid=bjensen,ou=people,dc=example,dc=com inactivated.
$
```

  有关详细信息，请参见 `ns-inactivate`(1M) 手册页。

▼ 重新激活帐户

请按以下所述对帐户或角色解除锁定。

注 – 您必须以目录管理员身份进行绑定。

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

- **使用 `ns-activate` 命令重新激活帐户或角色。**
  以下命令将使 Barbara Jensen 的帐户再次处于活动状态：

```
$ ns-activate -D "cn=Directory Manager" -j pwd.txt \
-I uid=bjensen,ou=people,dc=example,dc=com
uid=bjensen,ou=people,dc=example,dc=com activated.
$
```

  有关详细信息，请参见 `ns-activate`(1M) 手册页。

◆ ◆ ◆ 第 8 章

目录服务器备份和恢复

目录服务器所管理的数据通常是批量导入的。Directory Server Enterprise Edition 提供了用于导入和导出整个后缀的工具。此外，它还提供了用于同时备份所有后缀以及从备份恢复所有数据的工具。

在开始执行任何备份或恢复操作之前，请确保设计一个符合您自身要求的备份和恢复策略。有关不同备份选项、需要考虑的注意事项，以及规划备份和恢复策略的准则的详细信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Deployment Planning Guide》中的“Designing Backup and Restore Policies”。

本章包含以下主题：

二进制备份

本部分介绍如何执行目录数据的二进制备份。除了本部分介绍的二进制备份过程之外，您还可以创建二进制副本，用于初始化复制拓扑中的后缀。请参见第 227 页中的 “使用二制进副本初始化复制后缀”。

仅备份目录数据

二进制数据备份将保存目录数据的副本，如果以后数据库文件被损坏或删除，您可以使用此副本。此操作不会备份配置数据。如果您要备份整个目录服务器以用于灾难恢复，请参见第 190 页中的 “灾难恢复”。

注意 – 切勿在备份操作期间停止服务器。

执行备份的频率必须高于**清除延迟**。清除延迟由 `nsDS5ReplicaPurgeDelay` 属性指定，它是以秒为单位的时间段，系统将在该时间段过后对更改日志执行内部清除操作。默认的清除延迟为 604800 秒（1 周）。更改日志用于维护更新记录（此更新可能已被复制或尚未复制）。

如果执行备份的频率低于清除延迟，则更改日志可能会在备份前就已被清除。因此，如果使用备份恢复数据，更改将会丢失。

默认情况下，本部分介绍的所有备份过程都会将服务器文件的副本存储在相同的主机上。您应该将备份复制并存储到不同的计算机或文件系统中，以获得更高的安全性。

▼ 备份目录数据

必须停止目录服务器才能运行 `dsadm backup` 命令。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

- **备份目录数据。**

  ```
  $ dsadm backup instance-path archive-dir
  ```

  例如：

  ```
  $ dsadm backup /local/ds /local/tmp/20051205
  ```

注 – 可以在服务器运行时通过使用 `dsconf backup` 命令备份目录数据。但是，如果在备份运行时更改了目录数据，则很难正确恢复数据。要避免在使用 `dsconf backup` 时出现此问题，请设置复制引用或将服务器设置为只读。

有关 `dsadm` 和 `dsconf` 命令的详细信息，请参见 `dsadm`(1M) 和 `dsconf`(1M) 手册页。

▼ 备份 `dse.ldif` 文件

恢复服务器时，`dse.ldif` 配置文件必须包含与备份服务器时相同的配置信息。

- **备份 `dse.ldif` 配置文件。**

  ```
  $ cp instance-path/config/dse.ldif archive-dir
  ```

执行以下操作时，目录服务器会自动将 dse.ldif 配置文件备份到目录 *instance-path*/config 中。

- 启动目录服务器时，将在名为 dse.ldif.startOK 的文件中创建 dse.ldif 文件的备份。
- 修改 cn=config 分支时，文件将首先备份到 config 目录中的 dse.ldif.bak 的文件，然后服务器才会将修改写入 dse.ldif 文件。

备份文件系统

此过程将使用**冻结模式**功能。冻结模式允许您停止磁盘上的数据库更新，以便安全地实施文件系统快照。可以将冻结模式作为一种附加措施，以确保执行可靠的备份。

正在执行文件系统备份时，服务器不能在磁盘上写入用户数据。如果您确定在特定时间帧内不会发生更新，请在此段时间内进行备份。如果您无法确定是否会发生更新，请在备份前将服务器置于冻结模式。

打开冻结模式后，所有已配置的数据库都将处于脱机状态。所有正在进行的内部操作都将获得数据库即将进入脱机状态的通知。将完成正在进行的 LDAP 操作，并会刷新数据库环境。后续的传入操作（包括搜索用户数据）将被拒绝，直到关闭冻结模式时为止。但是，在冻结模式下仍可搜索配置参数。

在单服务器拓扑中，冻结模式下收到的操作将导致返回 LDAP 错误。对于处于脱机状态的数据库，所记录的错误消息为标准错误。在复制拓扑中将返回引用。为了使冻结模式正确工作，不应该在数据库上运行其他任务。

▼ 备份文件系统

对于此过程的某些部分，可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的“目录服务控制中心界面”和 DSCC 联机帮助。此过程的其他部分只能使用命令行完成。

1 （可选的）将服务器置于冻结模式。

```
$ dsconf set-server-prop -h host -p port read-write-mode:frozen
```

2 使用适用于您的文件系统类型的工具备份文件系统。

3 如果您的服务器处于冻结模式，请重新将服务器设置为读写模式。

```
$ dsconf set-server-prop -h host -p port read-write-mode:read-write
```

如果服务器收到来自其他服务器的复制更新，则这些复制更新会在关闭冻结模式后立即启动。

备份到 LDIF

备份到 LDIF 允许您将目录数据备份到格式化的 LDIF 文件。

导出到 LDIF

可以通过使用 LDIF 导出后缀内容来备份目录数据。执行以下操作时，导出数据非常有用：

- 在您的服务器中备份数据
- 将数据复制到其他目录服务器
- 将数据导出到其他应用程序
- 更改目录拓扑后重新填充后缀

导出操作不会导出配置信息 (`cn=config`)。

注意 – 不要在执行导出操作期间停止服务器。

▼ 将后缀导出到 LDIF

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **使用以下任一命令将后缀导出到 LDIF 文件：**
  - 如果您的服务器是本地服务器并且已停止，请键入：

    ```
    $ dsadm export instance-path suffix-DN LDIF-file
    ```
  - 如果您的服务器是远程服务器并且正在运行，请键入：

    ```
    $ dsconf export -h host -p port suffix-DN LDIF-file
    ```

以下示例使用 `dsconf export` 将两个后缀导出到单个 LDIF 文件：

```
$ dsconf export -h host1 -p 1389 ou=people,dc=example,dc=com \
 ou=contractors,dc=example,dc=com /local/ds/ldif/export123.ldif
```

还可以在 `dsadm export` 和 `dsconf export` 命令中使用 `--no-repl` 选项来指定不导出任何复制信息。默认情况下会将复制的后缀与复制信息一起导出到 LDIF 文件。得到的 LDIF 文件将包含复制机制所使用的属性子类型。然后可以在使用方服务器上导入此 LDIF 文件以初始化使用方副本，如第 224 页中的 “初始化副本”中所述

有关这些命令的详细信息，请参见 `dsadm`(1M) 和 `dsconf`(1M) 手册页。

二进制恢复

以下过程介绍如何在目录中恢复后缀。必须已经使用第 179 页中的 “仅备份目录数据”中所述的过程备份了服务器。在恢复复制协议中所涉及的后缀之前，请先阅读第 187 页中的 “恢复复制的后缀”。

注意 – 不要在执行恢复操作期间停止服务器。由于恢复服务器会覆盖所有现有的数据库文件，因此自备份以来对数据所做的所有修改都会丢失。

▼ 恢复服务器

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **使用以下任一命令恢复服务器：**
  - 如果您的服务器是本地服务器并且已停止，请键入：

    ```
    $ dsadm restore instance-path archive-dir
    ```

    例如，要从备份目录恢复备份，请键入：

    ```
    $ dsadm restore /local/ds/ local/ds/bak/2006_07_01_11_34_00
    ```
  - 如果您的服务器是远程服务器并且正在运行，请键入：

    ```
    $ dsconf restore -h host -p port archive-dir
    ```

    例如，要从备份目录恢复备份：

    ```
    $ dsconf restore -h host1 -p 1389 /local/ds/bak/2006_07_01_11_34_00
    ```

有关这些命令的详细信息，请参见 `dsadm`(1M) 和 `dsconf`(1M) 手册页。

恢复 `dse.ldif` 配置文件

目录服务器将在以下目录中创建 `dse.ldif` 文件的两个备份副本：

instance-path`/config`

`dse.ldif.startOK` 文件将在服务器启动时记录 `dse.ldif` 文件的副本。`dse.ldif.bak` 文件包含最近对 `dse.ldif` 文件所做更改的备份。请将包含最近更改的文件复制到您的目录中。

▼ 恢复 dse.ldif 配置文件

对于此过程的某些部分，可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的“目录服务控制中心界面”和 DSCC 联机帮助。此过程的其他部分只能使用命令行完成。

1. **停止服务器。**

   ```
   $ dsadm stop instance-path
   ```

2. **转到包含配置文件的目录。**

   ```
   $ cd instance-path/config
   ```

3. **使用有效的备份配置文件覆盖 `dse.ldif` 文件，例如：**

   ```
   $ cp dse.ldif.startOK dse.ldif
   ```

4. **使用以下命令启动服务器：**

   ```
   $ dsadm start instance-path
   ```

从 LDIF 文件导入数据

可以使用以下方法将数据导入目录服务器后缀：

- 从 LDIF 文件初始化后缀。此操作将删除后缀中的当前数据，并将其替换为该 LDIF 文件的内容。
- 使用 LDIF 文件执行批量的 `ldapadd`、`ldapmodify` 或 `ldapdelete` 操作。这样您可以在目录的任何后缀中批量添加、修改和删除条目。

下表显示了初始化后缀与批量添加、修改和删除条目之间的区别。

表 8–1 初始化后缀与批量导入数据的比较

比较范围	初始化后缀	批量添加、修改和删除条目
覆盖内容	覆盖 内容	不覆盖内容
LDAP 操作	仅添加	添加、修改和删除
性能	快	较慢
对服务器故障的响应	响应能力极差（发生故障后会丢失所有更改）	响应能力最强（将保留故障发生以前的所有更改）
LDIF 文件位置	客户端本地或服务器本地	在客户机上

表 8–1 初始化后缀与批量导入数据的比较 （续）

比较范围	初始化后缀	批量添加、修改和删除条目
导入配置信息 (cn=config)	导入配置信息	不导入配置信息

命令	服务器为本地服务器并且已停止时：
	`dsadm import`
	服务器为远程服务器并且正在运行时：
	`dsconf import`
		`ldapmodify -B`

初始化后缀

初始化后缀会使用 LDIF 文件（仅包含用于添加的条目）的内容覆盖后缀中的现有数据。

要初始化后缀，您必须以目录管理员或管理者身份通过验证。

服务器正在运行时，只有目录管理员和管理者才能导入包含根条目的 LDIF 文件。出于安全考虑，只有这些用户才能访问后缀的根条目，例如 `dc=example,dc=com.`。

在恢复复制协议中所涉及的后缀之前，请先阅读第 187 页中的 “恢复复制的后缀”。

▼ 初始化后缀

注 – 您所导入的所有 LDIF 文件都必须使用 UTF-8 字符集编码。

在初始化后缀时，LDIF 文件必须包含相应后缀的根条目和所有目录树节点。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **使用以下任一命令从 LDIF 文件初始化后缀，也就是说，将数据库内容导入 LDIF 文件。**

注意 – 这些命令将覆盖后缀中的数据。

- 如果您的服务器是本地服务器并且已停止，请键入：

  ```
  $ dsadm import instance-path LDIF-file suffix-DN
  ```

  以下示例使用 `dsadm import` 命令将两个 LDIF 文件导入一个后缀。

```
$ dsadm import /local/ds /local/file/example/demo1.ldif \
 /local/file/example/demo2.ldif dc=example,dc=com
```

- 如果您的服务器是远程服务器并且正在运行，请键入：

```
$ dsconf import -h host -p port LDIF-file suffix-DN
```

以下示例使用 `dsconf import` 导入 LDIF 文件。运行此命令不需要超级用户权限，但您必须以具有超级用户权限的用户身份（如目录管理员）通过验证。

```
$ dsconf import -h host1 -p 1389 /local/file/example/demo1.ldif \
 ou=People,dc=example,dc=com
```

注 – 如果在多个后缀上并行运行 `dsconf import` 和/或 `dsconf reindex` 命令，则事务日志将会增大，并可能对性能造成不利影响。

有关这些命令的详细信息，请参见 `dsadm`(1M) 和 `dsconf`(1M) 手册页。

批量添加、修改和删除条目

执行 `ldapmodify` 操作时，可以批量添加、修改或删除条目。这些条目是在 LDIF 文件指定的，该文件包含用于修改或删除现有条目的更新语句。此操作不会删除已经存在的条目。

目录服务器所管理的任何后缀都有可能成为更改条目的目标。与添加条目的任何其他操作一样，服务器在导入新条目时会为所有新条目编制索引。

`ldapmodify` 命令将通过 LDAP 导入 LDIF 文件，并执行该文件包含的所有操作。使用此命令，可以同时修改所有目录后缀中的数据。

在恢复复制协议中所涉及的后缀之前，请参见第 187 页中的 “恢复复制的后缀”。

▼ 批量添加、修改和删除条目

注 – 您所导入的所有 LDIF 文件都必须使用 UTF-8 字符集编码。

导入 LDIF 文件时，目录中必须存在父条目，或者必须先从该文件中添加父条目。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **从 LDIF 文件批量添加、修改或删除。**

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w - -B baseDN -f LDIF-file
```

以下示例使用 ldapmodify 命令执行导入。运行此命令不需要超级用户权限，但您必须以具有超级用户权限的用户身份（如 cn=Directory Manager 或 cn=admin,cn=Administrators,cn=config）通过验证。最后一个参数指定要导入的 LDIF 文件的名称。

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w - \
 -B dc=example,dc=com -f /local/ds/ldif/demo.ldif
```

恢复复制的后缀

恢复在提供方服务器和使用方服务器之间复制的后缀之前，需要了解一些特殊的注意事项。请尽可能通过复制机制（而不是通过从备份恢复后缀）来更新后缀。

恢复提供方或集线器实例时，服务器配置必须与创建备份时的配置相同。要确保这一点，请在恢复目录服务器数据之前先恢复 dse.ldif 文件。请参见第 183 页中的 “恢复 dse.ldif 配置文件”。

本部分介绍如何、何时恢复副本，以及如何确保副本在恢复后与其他副本同步。要初始化副本，请参见第 224 页中的 “初始化副本”。

如果您有大型的复制后缀并要添加许多条目，同时要确保正确添加复制更新，请参见第 231 页中的 “逐渐向大型复制后缀添加大量条目”。

本部分包含与以下内容有关的信息：

在单主方案中恢复提供方

作为单主提供方的后缀包含整个复制拓扑的授权数据。因此，恢复此后缀等同于重新初始化整个拓扑中的所有数据。只有在使用要恢复的备份内容重新初始化所有数据时，才应恢复单主提供方。

如果单主数据由于错误而无法恢复，请考虑使用某个使用方上的数据，因为它可能包含比备份还新的更新。在这种情况下，您需要将使用方副本中的数据导出到 LDIF 文件，然后从该 LDIF 文件重新初始化主服务器。

无论您在主副本上恢复备份还是导入 LDIF 文件，都必须重新初始化从此副本接收更新的所有集线器和使用方副本。在提供方服务器的日志文件中将记录一条消息，以提醒您必须对使用方进行重新初始化。

在多主方案中恢复提供方

在多主复制中，其他每个主服务器都包含复制数据的授权副本。您无法恢复旧的备份，因为该备份对于当前的副本内容可能已经过时。如有可能，应允许复制机制使用其他主服务器的内容来更新主服务器。

如果无法执行此操作，请使用以下任一方法恢复多主副本：

- 最简单的方法不是恢复备份，而是从其他某个主服务器重新初始化预期的主服务器。这可以确保将最新数据发送到预期的主服务器，并且该数据可用于复制。请参见第 225 页中的 “从 LDIF 进行副本初始化”。
- 对于具有数百万条目的副本，较快的做法是创建二进制副本，以恢复其他某个主服务器上的较新备份。请参见第 227 页中的 “使用二制进副本初始化复制后缀”。
- 如果主服务器备份的存留期不长于其他**任何**主服务器上更改日志内容的最长存留期，则可使用该备份恢复此主服务器。有关更改日志存留期的描述，请参见第 218 页中的 “修改主副本上的更改日志设置”。恢复旧的备份之后，其他主服务器将使用其更改日志中自保存该备份以来进行的所有修改来更新此主服务器。

无论您以何种方式恢复或重新初始化，主副本在初始化后都仍然保持只读模式。此行为可使副本与其他主服务器进行同步，以便您在同步后允许执行写入操作，如第 189 页中的 “在多主方案中恢复主服务器”中所述。

在已恢复或重新初始化的主服务器上允许执行写入操作之前一致所有副本的好处在于，所有集线器或使用方服务器都不需要重新初始化。

恢复集线器

本部分仅适用于复制机制无法自动更新集线器副本的情况。数据库文件损坏或复制中断时间过长都属于这种情况。在这些情况下，您需要使用以下任一方法恢复或重新初始化集线器副本：

- 最简单的方法不是恢复备份，而是从某个主副本重新初始化集线器。这可以确保将最新数据发送到集线器，并且该数据可用于复制。请参见第 185 页中的 “初始化后缀”。
- 对于具有数百万条目的副本，较快的做法是创建二进制副本，以恢复来自其他集线器复制后缀的较新备份。请参见第 227 页中的 “使用二制进副本初始化复制后缀”。如果没有需要复制的其他集线器副本，请按照上一条的描述重新初始化集线器，或者按照下一条的描述恢复集线器（如有可能）。
- 如果集线器备份的存留期不长于其他**任何**提供方（集线器或主副本）上更改日志内容的最长存留期，则可使用该备份恢复此集线器。恢复集线器之后，提供方将使用其更改日志中自保存该备份以来进行的所有修改来更新此集线器。

注 – 无论您以何种方式恢复或重新初始化集线器副本，接下来都**必须**重新初始化此集线器的所有使用方，包括所有其他级别的集线器。

恢复专用使用方

本部分仅适用于复制机制无法自动更新专用使用方副本的情况。数据库文件损坏或复制中断时间过长都属于这种情况。在这些情况下，您需要使用以下任一方法恢复或重新初始化使用方：

- 最简单的方法不是恢复备份，而是从某个提供方（主服务器或集线器副本）重新初始化使用方。这可以确保将最新数据发送到使用方，并且该数据可用于复制。请参见第 225 页中的 “从 LDIF 进行副本初始化”。
- 对于具有数百万条目的副本，较快的做法是创建二进制副本，以恢复来自其他使用方复制后缀的较新备份。请参见第 227 页中的 “使用二制进副本初始化复制后缀”。如果没有需要复制的其他使用方，请按照上一条的描述重新初始化副本，或者按照下一条的描述恢复副本（如有可能）。
- 如果使用方备份的存留期不长于其他**任何**提供方（集线器或主副本）上更改日志内容的最长存留期，则可使用该备份恢复此使用方。恢复使用方之后，提供方将使用其更改日志中自保存该备份以来进行的所有修改来更新此使用方。

在多主方案中恢复主服务器

使用多主复制时，其他主服务器可能会在给定主服务器恢复期间处理更改操作。因此，完成恢复操作之后，新的主服务器还必须接收恢复数据中未包含的较新更新。由于恢复主服务器可能需要大量时间，因此挂起的更新数也可能十分庞大。

为了一致这些挂起的更新，新恢复的主服务器将自动设置为只读模式，以便在恢复后执行客户端操作。此情况仅在通过以下方式恢复主服务器时适用：使用命令行从 LDIF 文件导入数据，或者使用备份执行二进制复制。

因此，在完成恢复操作之后，多主配置中的主服务器将处理复制更新并允许执行读取操作，但是对于来自客户端的所有写入操作将返回引用。

要在允许更新前验证新的主服务器是否与其他主服务器完全同步，请在已初始化的主服务器上手动启用更新。

注 – 由于主副本会因为上述新行为而发送引用，因此等待执行写入操作的客户端可能会达到配置的跳数限制。您可能需要提高客户端的跳数限制配置，以使其能够访问可用的主服务器。如果已初始化或重新初始化所有主副本，则所有写入操作都会失败，因为所有副本都不会接受客户端更新。

在任何情况下，都应密切监视已初始化的主服务器，并适当地设置引用属性，以尽可能提高服务器的响应能力。

▼ 通过命令行开始接受更新

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

可以在脚本中使用以下命令，以便自动执行多主副本的初始化过程：

1. **使用 `insync` 工具确保副本已与所有其他主服务器一致。**

   如果所有服务器上修改之间的延迟为零，或副本从未有任何需要复制的更改（延迟为 `-1`），则表明副本处于同步状态。有关详细信息，请参见 `insync`(1) 手册页。

2. **开始接受更新。**

   ```
   $ dsconf set-suffix-prop -h host -p port suffix-DN repl-accept-client-update-enabled:on
   ```

   此命令自动将服务器设置为读写模式。

灾难恢复

如果要备份或恢复目录服务器以用于灾难恢复，请使用以下过程。

▼ 创建备份以用于灾难恢复

对于此过程的某些部分，可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。此过程的其他部分只能使用命令行完成。

1. **通过使用 `dsadn backup` 或 `dsconf backup` 命令创建数据库文件的备份。**

   使用第 179 页中的 “二进制备份”中的过程，并将备份文件存储到安全位置。

2. **将配置目录 *instance-path* `/config` 复制到安全位置。**

3. **将模式目录 *instance-path*`/config/schema` 复制到安全位置。**

4. **将别名目录 *instance-path*`/alias` 复制到安全位置。**

▼ 进行灾难恢复

对于此过程的某些部分，可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。此过程的其他部分只能使用命令行完成。

1. **安装与主机先前所安装的版本相同的目录服务器版本。**
2. **使用 `dsadm create` 命令创建服务器实例。**
   使用备份时所用的实例。请参见第 56 页中的 “创建后缀”。
3. **恢复配置目录** *instance-path*`/config`。
4. **恢复模式目录** *instance-path*`/config/schema`。
5. **恢复别名目录** *instance-path*`/alias`。
6. **确保已恢复的服务器的配置正确。**
   例如，目录结构和插件配置必须与备份服务器上相同。
7. **通过使用 `dsconf restore` 命令恢复数据库文件。**
   使用第 183 页中的 “二进制恢复”中的过程。

◆ ◆ ◆ 第 9 章

9 目录服务器组、角色和 CoS

管理代表用户的条目时，除了在目录中使用数据的分层结构之外，通常还需要创建共享公用属性值的组。目录服务器通过组、角色和服务类 (Class of Service, CoS) 提供了高级条目管理功能。

本章包含以下主题：

关于组、角色和服务类

组、角色和 CoS 的定义如下：

- 组是指定其他条目（成员列表或成员过滤器）的条目。对于由成员列表构成的组，目录服务器将在每个用户条目上生成 `isMemberOf` 属性值。因此，用户条目上的 `isMemberOf` 属性将显示该条目所属的所有组。
- 角色可以提供与组相同的功能，并且还会通过某种机制在每个角色成员上生成 `nsrole` 属性。
- CoS 会生成已计算属性，它允许条目共享公用属性值，而不必在每个条目中存储属性。

目录服务器可以基于角色、组和 CoS 已计算属性的值来执行搜索。任何操作中使用的过滤字符串都可以包含 `nsRole` 属性或由 CoS 定义生成的任何属性。还可以使用过滤字符串对此属性的值执行任何比较操作。但是无法为 CoS 已计算属性编制索引。因此，涉及 CoS 已生成属性的任何搜索都可能会消耗大量资源（从时间和内存上）。

为了充分利用角色、组和服务类所提供的功能，请在目录部署的规划阶段确定您的目录拓扑。有关这些功能以及如何使用这些功能简化拓扑的描述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Deployment Planning Guide》中的“Grouping Directory Entries and Managing Attributes”。

要深入了解角色和组的工作方式，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的第 8 章 “Directory Server Groups and Roles”。有关 CoS 的详细描述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的第 9 章 “Directory Server Class of Service”。

管理组

组允许您将条目关联起来以便于管理。例如，使用组可以更轻松地定义访问控制指令 (Access Control Instruction, ACI)。组定义是一些特殊条目，可以在静态列表中指定其成员，也可以提供用于定义一组动态条目的过滤器。

无论组定义条目的位置如何，组的可能成员范围都是整个目录。为了简化管理，所有组定义条目通常都存储在一个位置，一般是根后缀下的 `ou=Groups`。

组可以分为静态组和动态组两种类型。

- **静态组**。定义静态组的条目是从 `groupOfNames` 或 `groupOfUniqueNames` 对象类继承来的。组成员将按 DN 列出，作为 `member` 或 `uniqueMember` 属性的多个值。

  或者，也可以为静态组使用 `isMemberOf` 属性。`isMemberOf` 属性将在开始搜索时进行计算并添加到用户条目，然后在完成搜索后再次被删除。此功能使得组的管理更加轻松，并且提供了快速读取访问。
- **动态组**。定义动态组的条目是从 `groupOfURLs` 对象类继承来的。组成员身份由多值属性 `memberURL` 中指定的一个或多个过滤器定义。动态组中的成员是每次评估过滤器时与任一过滤器相匹配的条目。

▼ 创建新的静态组

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1 **使用 `ldapmodify` 命令创建新的静态组。**

例如，要创建名为 System Administrators 的新静态组并添加一些成员，可以使用以下命令：

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=System Administrators, ou=Groups, dc=example,dc=com
cn: System Administrators
objectclass: top
objectclass: groupOfNames
```

```
ou: Groups
member: uid=kvaughan, ou=People, dc=example,dc=com
member: uid=rdaugherty, ou=People, dc=example,dc=com
member: uid=hmiller, ou=People, dc=example,dc=com
```

2 **检查是否已创建新组以及是否已添加成员。**

例如，要检查 Kirsten Vaughan 是否在新的 System Administrators 组中，请键入：

```
$ ldapsearch -b "dc=example,dc=com" uid=kvaughan isMemberOf
uid=kvaughan,ou=People,dc=example,dc=com
isMemberOf: cn=System Administrators, ou=Groups, dc=example,dc=com
isMemberOf: cn=HR Managers,ou=groups,dc=example,dc=com
```

▼ 创建新的动态组

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

- **使用 `ldapmodify` 命令创建新的动态组。**

例如，要创建名为 Database Administrators 的新动态组并添加姓氏为 Jensen 的成员，可以使用以下命令：

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=Database Administrators, ou=Groups, dc=example,dc=com
cn: Database  Administrators
objectclass: top
objectclass: groupOfUrls
ou: Groups
memberURL: ldap:///dc=example,dc=com??sub?(sn=Jensen)
```

管理角色

角色是一种备用组机制，在设计上可以使应用程序使用起来更加轻松有效。虽然角色的定义和管理方式与组类似，但是每个成员条目中生成的角色属性将自动表示条目的角色。例如，应用程序可以读取条目的角色，而不必选择组并浏览成员列表。

默认情况下，角色的范围被限定为定义该范围时所在的子树。但是，您可以扩展嵌套角色的范围。可以允许范围嵌套其他子树中的角色，并包含位于目录中任意位置的成员。有关详细信息，请参见第 198 页中的 “扩展角色的范围”和第 198 页中的 “嵌套角色定义的示例”。

本部分介绍如何安全地使用角色，以及如何从命令行管理角色。

安全地使用角色

要安全地使用角色，必须设置访问控制指令 (Access Control Instruction, ACI) 以保护相应的属性。例如，用户 A 具有受管理角色 MR。受管理角色相当于静态组，通过将 nsRoleDN 属性添加到每个成员条目来为该条目明确指定角色。已通过命令行使用帐户去活锁定了 MR 角色。这意味着用户 A 无法绑定到服务器，因为该用户 nsAccountLock 属性的计算结果为 "true"。但是，假定该用户已经绑定，并发现自己现在因 MR 角色而被锁定。如果没有相应的 ACI 来阻止用户具有 nsRoleDN 属性的写入访问权限，则该用户可以从自己的条目中删除 nsRoleDN 属性并解除锁定。

要阻止用户删除 nsRoleDN 属性，必须应用 ACI。使用过滤角色时，必须保护过滤器中可阻止用户通过修改属性放弃过滤角色的部分。应禁止用户添加、删除或修改过滤角色所使用的属性。同理，如果已计算过滤属性的值，则必须保护可以修改过滤属性值的所有属性。由于嵌套角色可以包含过滤角色和受管理角色，因此对于嵌套角色中包含的每个角色，都应考虑上述几点。

有关设置 ACI 以获得安全性的详细说明，请参见第 6 章。

从命令行管理角色

角色是在目录管理者可以通过命令行实用程序访问的条目中定义的。创建角色之后，可按以下方式为角色指定成员：

- 受管理角色的成员在条目中具有 nsRoleDN 属性。
- 过滤角色的成员是与 nsRoleFilter 属性中所指定的过滤器相匹配的条目。
- 嵌套角色的成员是嵌套角色定义条目的 nsRoleDN 属性中所指定的角色的成员。

所有角色定义都是从 LDAPsubentry 和 nsRoleDefinition 对象类继承来的。以下示例显示了特定于每类角色的其他对象类和关联属性。

受管理角色定义的示例

要为所有营销人员创建角色，请使用以下 ldapmodify 命令：

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=Marketing,ou=marketing,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsSimpleRoleDefinition
objectclass: nsManagedRoleDefinition
cn: Marketing
description: managed role for marketing staff
```

请注意，nsManagedRoleDefinition 对象类是从 LDAPsubentry、nsRoleDefinition 和 nsSimpleRoleDefinition 对象类继承来的。

通过更新营销人员 Bob 的条目，可以为该成员指定角色，如下所示：

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=Bob Arnold,ou=marketing,ou=People,dc=example,dc=com
changetype: modify
add: nsRoleDN
nsRoleDN: cn=Marketing,ou=marketing,ou=People,dc=example,dc=com
```

nsRoleDN 属性表示该条目是受管理角色的成员。受管理角色由角色定义的 DN 标识。要允许用户修改自己的 nsRoleDN 属性，但阻止用户添加或删除 nsManagedDisabledRole，请添加以下 ACI：

```
aci: (targetattr="nsRoleDN")(targattrfilters="add=nsRoleDN:
(!(nsRoleDN=cn=AdministratorRole,dc=example,dc=com)),
del=nsRoleDN:(!(nsRoleDN=cn=nsManagedDisabledRole,dc=example, dc=com)")
(version3.0;aci "allow mod of nsRoleDN by self except for critical values";
allow(write) userdn="ldap:///self";)
```

过滤角色定义的示例

要为销售经理设置过滤角色（假定这些销售经理都具有 isManager 属性），请使用以下 ldapmodify 命令：

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=ManagerFilter,ou=sales,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsFilteredRoleDefinition
cn: ManagerFilter
nsRoleFilter: (isManager=True)
Description: filtered role for sales managers
```

请注意，nsFilteredRoleDefinition 对象类是从 LDAPsubentry、nsRoleDefinition 和 nsComplexRoleDefinition 对象类继承来的。nsRoleFilter 属性会指定一个过滤器，用于查找 ou=sales 组织中拥有下属的所有员工，例如：

```
$ ldapsearch -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w - \
 -b "ou=People,dc=example,dc=com" -s sub "(cn=*Fuentes)"
dn: cn=Carla Fuentes,ou=sales,ou=People,dc=example,dc=comcn: Carla Fuentes
isManager: TRUE...
nsRole: cn=ManagerFilter,ou=sales,ou=People,
dc=example,dc=com
```

注 – 过滤角色的过滤字符串可以基于任何属性，由 CoS 机制生成的已计算属性除外。

如果过滤角色成员是用户条目，您可以选择限制他们在角色中添加或删除自身的能力。可以使用 ACI 保护过滤属性。

嵌套角色定义的示例

嵌套在嵌套角色中的角色是通过 `nsRoleDN` 属性指定的。可以使用以下命令创建一个角色，该角色同时包含前面示例中所创建的营销人员和销售经理角色的成员：

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=MarketingSales,ou=marketing,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsNestedRoleDefinition
cn: MarketingSales
nsRoleDN: cn=ManagerFilter,ou=sales,ou=People,dc=example,dc=com
nsRoleDN: cn=Marketing,ou=marketing,ou=People,dc=example,dc=com
nsRoleScopeDN: ou=sales,ou=People,dc=example,dc=com
```

请注意，`nsNestedRoleDefinition` 对象类是从 `LDAPsubentry`、`nsRoleDefinition` 和 `nsComplexRoleDefinition` 对象类继承来的。`nsRoleDN` 属性包含营销受管理角色和销售经理过滤角色的 DN。前面示例中的用户 Bob 和 Carla 都将成为此新嵌套角色的成员。

此过滤器的范围包括默认范围（该过滤器所在的子树），以及任何 `nsRoleScopeDN` 属性值下的子树。在本案例中，`ManagerFilter` 位于 `ou=sales,ou=People,dc=example,dc=com` 子树中。必须将此子树添加到该范围。

扩展角色的范围

目录服务器提供了一个属性，该属性允许将角色的范围扩展到角色定义条目的子树之外。此单值属性 `nsRoleScopeDN` 包含要添加到现有角色的范围的 DN。只能将 `nsRoleScopeDN` 属性添加到嵌套角色。请参见第 198 页中的 “嵌套角色定义的示例”。

▼ 扩展角色的范围

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

`nsRoleScopeDN` 属性允许您扩展某个子树中的角色范围，以包含另一个子树中的条目。例如，假定 example.com 目录树中有两个主要子树：`o=eng,dc=example,dc=com`（工程子树）和 `o=sales,dc=example,dc=com`（销售子树）。工程子树中的用户需要访问由销售子树中的角色 (`SalesAppManagedRole`) 管理的销售应用程序。要扩展该角色的范围，请执行以下操作：

1. **在工程子树中为用户创建一个角色。**
   例如，创建角色 `EngineerManagedRole`。此示例使用受管理角色，但也可以是过滤角色或嵌套角色。

2. **在销售子树中创建一个嵌套角色（例如 `SalesAppPlusEngNestedRole`），以存储新创建的 `EngineerManagedRole` 和初始的 `SalesAppManagedRole`。**

3. **使用要添加的工程子树范围的 DN（在本案例中为 `o=eng,dc=example,dc=com`）将 `nsRoleScopeDN` 属性添加到 `SalesAppPlusEngNestedRole` 中。**
   必须为工程用户授予必要的权限，以便该用户可以访问 `SalesAppPlusEngNestedRole` 角色，进而可以使用销售应用程序。此外，还必须复制角色的整个范围。

注 – 对嵌套角色扩展范围的限制意味着，以前在某个域中管理角色的管理员只有权使用其他域中已存在的角色。该管理员无法在其他域中创建任意角色。

服务类

为客户端应用程序检索到条目时，服务类 (Class of Service, CoS) 机制会生成已计算属性，这可以简化条目管理并降低存储要求。CoS 机制允许在条目之间共享属性，并且与组和角色一样，CoS 也依赖于帮助应用程序条目。

有关如何在部署中使用 CoS 的说明，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Deployment Planning Guide》中的“Managing Attributes With Class of Service”。

有关如何在目录服务器中实现 CoS 的描述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的第 9 章 “Directory Server Class of Service”。

注 – 任何搜索操作都可以测试 CoS 已生成属性是否存在，或者比较属性值。可以在客户端搜索操作的任何过滤字符串中使用已计算属性的名称，过滤角色中使用的内部过滤器除外。

安全地使用 CoS

以下部分介绍每个 CoS 条目中数据读保护和写保护的一般准则。第 6 章中介绍了定义单个访问控制指令 (Access Control Instruction, ACI) 的详细过程。

保护 CoS 定义条目

虽然 CoS 定义条目不包含已生成属性的值，但它提供了用于查找该值的信息。读取 CoS 定义条目可了解如何查找包含该值的模板条目。对此条目执行写入操作可修改已计算属性的生成方式。

因此，应该为 CoS 定义条目定义读取 ACI 和写入 ACI。

保护 CoS 模板条目

CoS 模板条目包含 CoS 已生成属性的值。因此，模板中的 CoS 属性至少要受到 ACI 的读取和更新保护。

- 如果使用**指针** *CoS*，应禁止重命名单个模板条目。在大多数情况下，最简单的做法就是保护整个模板条目。
- 如果使用**传统** *CoS*，所有模板条目都具有定义条目中所指定的公用父条目。如果父条目中只存储了模板，则对父条目的访问控制将保护这些模板。但是，如果父条目下的其他条目需要访问权限，则必须单个保护模板条目。
- 如果使用**间接** *CoS*，则模板可以是目录中的任何条目，包括仍需访问的用户条目。根据需要，您可以在整个目录中控制对 CoS 属性的访问，或确保 CoS 属性在用作模板的每个条目中都是安全的。

保护 CoS 的目标条目

CoS 定义范围中的所有条目（将为这些条目生成 CoS 已计算属性）都会参与属性值的计算。

默认情况下，当目标条目中已存在 CoS 属性时，CoS 机制不会覆盖此值。如果不希望如此，可以将 CoS 定义为覆盖目标条目，或保护所有潜在目标条目中的 CoS 属性。

间接 CoS 和传统 CoS 还依赖于目标条目中的说明符属性。此属性用于指定要使用的模板条目的 DN 或 RDN。应该使用 ACI 在整个 CoS 范围内全局保护此属性，或者在需要保护的每个目标条目上单独保护此属性。

保护其他相关性

可以根据其他已生成的 CoS 属性和角色来定义 CoS 已计算属性。您必须了解并保护这些相关性，以确保 CoS 已计算属性受到保护。

例如，目标条目中的 CoS 说明符属性可能是 `nsRole`。因此，角色定义也必须受 ACI 保护。

通常，计算已计算属性值时所用的任何属性或条目都应具有提供读取和写入访问控制的 ACI。因此，应合理规划或简化复杂的相关性，以降低后续访问控制实现的复杂性。将其他已计算属性上的相关性降至最低可提高目录性能并减少维护操作。

从命令行管理 CoS

由于所有配置信息和模板数据都作为条目存储在目录中，因此可以使用 LDAP 命令行工具配置和管理 CoS 定义。本部分说明如何从命令行创建 CoS 定义条目和 CoS 模板条目。

从命令行创建 CoS 定义条目

所有 CoS 定义条目都具有 `LDAPsubentry` 对象类，并且是从 `cosSuperDefinition` 对象类继承来的。此外，每种类型的 CoS 都从特定对象类继承而来，并包含相应属性。下表列出了与每种类型的 CoS 定义条目相关联的对象类和属性。

表 9–1 CoS 定义条目中的对象类和属性

CoS 类型	CoS 定义条目

指针 CoS	`objectclass: top`
	`objectclass: LDAPsubentry`
	`objectclass: cosSuperDefinition`
	`objectclass: cosPointerDefinition`
	`cosTemplateDN:` *DN*
	`cosAttribute:` *attributeName override merge*

间接 CoS	`objectclass: top`
	`objectclass: LDAPsubentry`
	`objectclass: cosSuperDefinition`
	`objectclass: cosIndirectDefinition`
	`cosIndirectSpecifier:` *attributeName*
	`cosAttribute:` *attributeName override merge*

传统 CoS	`objectclass: top`
	`objectclass: LDAPsubentry`
	`objectclass: cosSuperDefinition`
	`objectclass: cosClassicDefinition`
	`cosTemplateDN:` *DN*
	`cosSpecifier:` *attributeName*
	`cosAttribute:` *attributeName override merge*

`cosAttribute` 始终为多值属性。每个值都定义一个由 CoS 机制生成的属性。

可以在 CoS 定义条目中使用以下属性。有关其中每个属性的详细信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Man Page Reference》中的单个属性。

表 9–2 CoS 定义条目属性

属性	在 CoS 定义条目中的用途
`cosAttribute` *attributeName override merge*	定义要生成值的已计算属性的名称。此属性为多值属性，每个值代表将通过模板生成值的属性的名称。*override* 和 *merge* 限定符指定在此表后面所述的特殊情况下如何计算 CoS 属性值。 *attributeName* 不能包含任何子类型。带有子类型的属性名称将被忽略，但会处理 `cosAttribute` 的其他值。
`cosIndirectSpecifier` *attributeName*	定义目标条目中属性的名称，间接 CoS 将使用此属性的值标识模板条目。已命名的属性称为说明符，并且必须包含每个目标条目中的完整 DN 字符串。此属性为单值属性，但 *attributeName* 可以是多值属性，以指定多个模板。
`cosSpecifier` *attributeName*	定义目标条目中属性的名称，传统 CoS 将使用此属性的值标识模板条目。已命名的属性称为说明符，并且必须包含可以在模板条目 RDN 中找到的字符串。此属性为单值属性，但 *attributeName* 可以是多值属性，以指定多个模板。
`cosTemplateDN` *DN*	提供模板条目的完整 DN（对于指针 CoS 定义）或基 DN（对于传统 CoS）。此属性为单值属性。

`cosAttribute` 属性允许 CoS 属性名称后面有两个限定符，即 override 限定符和 merge 限定符。

override 限定符描述当条目中已实际存在 CoS 动态生成属性时的行为。*override* 可为以下任一选项：

- `default`（或无限定符）- 表示当属性类型与已计算属性相同时，服务器不会覆盖条目中存储的实际属性值。
- `override` - 表示服务器始终返回由 CoS 生成的值，即使条目中已存在某个值。
- `operational` - 表示只有在搜索中明确请求时才会返回属性。操作属性不必通过模式检查即可返回。`operational` 限定符与 `override` 限定符具有相同的行为。

  只有在模式中也将某个属性定义为操作属性时，该属性才能成为操作属性。例如，如果 CoS 生成一个 `description` 属性值，则无法使用 `operational` 限定符，因为 `description` 属性在模式中未标记为操作属性。

merge 限定符要么不使用，要么为 `merge-schemes` 形式。此限定符允许 CoS 已计算属性成为多值属性，这些值可以来自多个模板或多个 CoS 定义 。有关详细信息，请参见第 203 页中的 “多值 CoS 属性” 。

覆盖实际的属性值

可以创建包含 `override` 限定符的指针 CoS 定义条目，如下所示：

```
dn: cn=pointerCoS,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosPointerDefinition
cosTemplateDn: cn=exampleUS,cn=data
cosAttribute: postalCode override
```

此指针 CoS 定义条目表示条目与生成 `postalCode` 属性值的模板条目 `cn=exampleUS,cn=data` 相关联。`override` 限定符表示此值优先于 `postalCode` 属性值（如果目标条目中存在该属性）。

注 – 如果 CoS 属性是使用 `operational` 或 `override` 限定符定义的，则无法对 CoS 范围内任何条目中的“实际”属性值执行写入操作。

多值 CoS 属性

指定 `merge-schemes` 限定符时，CoS 已生成属性在以下两种情况下可以成为多值属性：

- 使用间接或传统 CoS 时，目标条目中的说明符属性可以为多值属性。在这种情况下，每个值都确定一个模板，并且每个模板中的值都是生成值的一部分。
- 多个任意类型的 CoS 定义条目可以在 `cosAttribute` 中包含相同的属性名称。在这种情况下，如果所有定义都包含 `merge-schemes` 限定符，则生成的属性将包含由每个定义计算的所有值。

这两种情况可以同时发生，并定义更多的值。但是，重复的值只会在生成的属性中返回一次。

如果不使用 `merge-schemes` 限定符，模板条目的 `cosPriority` 属性将用于确定已生成属性在所有模板中的单一值。下一部分将介绍此方案。

`merge-schemes` 限定符永远不会将目标中定义的“实际”值与通过模板生成的值进行合并。*merge* 限定符独立于 *override* 限定符。所有配对情况都可能出现，并且每种情况表示的行为是互补的。 此外，还可以在属性名称后按任意顺序指定这些限定符。

注 – 如果同一属性具有多个 CoS 定义，则这些定义必须具有相同的 *override* 和 *merge* 限定符。如果 CoS 定义中存在不同的限定符对，将从所有定义中任意选择一种组合。

CoS 属性优先级

如果存在多个 CoS 定义或多值说明符，但未使用 `merge-schemes` 限定符，目录服务器将使用优先级属性选择用于定义已计算属性单一值的单一模板。

cosPriority 属性表示纳入考虑的所有模板中某一特定模板的全局优先级。优先级为零代表最高优先级。不包含 cosPriority 属性的模板被视为优先级最低。如果两个或两个以上的模板提供一个属性值，但却具有相同的优先级或没有优先级，此时将任意选择一个值。

使用 merge-schemes 限定符时不会考虑模板优先级。在合并时，纳入考虑的所有模板将定义一个值，而不管这些模板定义的优先级如何。cosPriority 属性是在 CoS 模板条目上定义的，如以下部分所述。

注 – cosPriority 属性不能具有负值。此外，由间接 CoS 生成的属性不支持优先级。不要在间接 CoS 定义的模板条目中使用 cosPriority。

从命令行创建 CoS 模板条目

使用指针 CoS 或传统 CoS 时，模板条目包含 LDAPsubentry 和 cosTemplate 对象类。必须专门为 CoS 定义创建此条目。如果将 CoS 模板条目作为 LDAPsubentry 对象类的实例，可以不受配置条目的限制而执行一般搜索。

间接 CoS 机制的模板是目录中任意的现有条目。不必提前对目标进行标识或为其提供 LDAPsubentry 对象类，但目标必须具有辅助 cosTemplate 对象类。只有将 CoS 评估为生成已计算属性及其值时，才会访问间接 CoS 模板。

CoS 模板条目必须始终包含 CoS 在目标条目上生成的属性和值。属性名称在 CoS 定义条目的 cosAttribute 属性中指定。

以下示例显示了指针 CoS（生成 postalCode 属性）最高优先级的模板条目：

```
dn: cn=ZipTemplate,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: extensibleobject
objectclass: cosTemplate
postalCode: 95054
cosPriority: 0
```

以下部分提供了模板条目示例以及每种类型的 CoS 定义条目示例。

指针 CoS 的示例

以下命令用于创建具有 cosPointerDefinition 对象类的指针 CoS 定义条目。此定义条目使用在上一部分示例中介绍的 CoS 模板条目，以便在 ou=People,dc=example,dc=com 树的所有条目之间共享公用的邮政编码。

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=pointerCoS,ou=People,dc=example,dc=com
objectclass: top
```

```
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosPointerDefinition
cosTemplateDn: cn=ZipTemplate,ou=People,dc=example,dc=com
cosAttribute: postalCode
```

CoS 模板条目 (cn=ZipTemplate,ou=People,dc=example,dc=com) 可将 postalCode 属性中存储的值提供给位于 ou=People,dc=example,dc=com 后缀下的所有条目。如果在同一子树中搜索没有邮政编码的任何条目，您将会看到已生成属性的值：

```
$ ldapsearch -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w - \
 -b "ou=People,dc=example,dc=com" -s sub "(cn=*Jensen)"
dn: cn=Babs Jensen,ou=People,dc=example,dc=com
cn: Babs Jensen
...
postalCode: 95054
```

间接 CoS 的示例

间接 CoS 命名 cosIndirectSpecifier 属性中的某个属性，以查找特定于每个目标的模板。间接 CoS 的模板条目可以是目录中的任何条目，包括其他用户条目。此间接 CoS 示例使用目标条目的 manager 属性标识 CoS 模板条目。模板条目是经理的用户条目。经理的用户条目包含要生成的属性的值。在本案例中，该值为 departmentNumber 的值。

以下命令将创建间接 CoS 定义条目，此条目包含 cosIndirectDefinition 对象类：

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=generateDeptNum,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosIndirectDefinition
cosIndirectSpecifier: manager
cosAttribute: departmentNumber
```

接下来，将 cosTemplate 对象类添加到模板条目，并确保这些条目定义要生成的属性。在此示例中，所有经理条目都是模板：

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=Carla Fuentes,ou=People,dc=example,dc=com
changetype: modify
add: objectclass
objectclass: cosTemplate
-
add: departmentNumber
departmentNumber: 318842
```

使用此 CoS，包含 manager 属性的目标条目（位于 ou=People,dc=example,dc=com 下的条目）将自动包含经理的部门编号。将在目标条目上计算 departmentNumber 属性，因为服务器中不存在该属性。但是，departmentNumber 属性将作为目标条目的一部分返回。例如，如果将 Babs Jensen 的经理定义为 Carla Fuentes，其部门编号如下所示：

```
$ ldapsearch -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w - \
 -b "ou=People,dc=example,dc=com" -s sub "(cn=*Jensen)"
dn: cn=Babs Jensen,ou=People,dc=example,dc=com
cn: Babs Jensen
...
manager: cn=Carla Fuentes,ou=People,dc=example,dc=com
departmentNumber: 318842
```

传统 CoS 的示例

此示例说明如何使用传统 CoS 生成邮政地址。将在模板条目中指定生成的值，该模板条目的位置由 CoS 定义中的 cosTemplateDN 和目标条目中的 cosSpecifier 属性值共同确定。以下命令通过使用 cosClassicDefinition 对象类创建定义条目：

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=classicCoS,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosClassicDefinition
cosTemplateDn: ou=People,dc=example,dc=com
cosSpecifier: building
cosAttribute: postalAddress
```

使用相同的命令，可以创建提供每栋大楼邮政地址的模板条目：

```
dn: cn=B07,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: extensibleobject
objectclass: cosTemplate
postalAddres: 7 Old Oak Street, Anytown, CA 95054
```

使用此 CoS，包含 building 属性的目标条目（位于 ou=People,dc=example,dc=com 下的条目）将自动包含相应的邮政地址。CoS 机制将搜索在 RDN 中具有说明符属性值的模板条目。在此示例中，如果将 Babs Jensen 分配到大楼 B07，则生成的通讯地址如下：

```
$ ldapsearch -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w - \
 -b "ou=People,dc=example,dc=com" -s sub "(cn=*Jensen)"
dn: cn=Babs Jensen,ou=People,dc=example,dc=com
cn: Babs Jensen
...
```

```
building: B07
postalAddress: 7 Old Oak Street, Anytown, CA 95054
```

创建基于角色的属性

您可以创建传统 CoS 模式，以便为基于条目角色的条目生成属性值。例如，可以使用基于角色的属性将服务器浏览限制设置为逐个条目浏览。

要创建基于角色的属性，请将 nsRole 属性作为传统 CoS 的 CoS 定义条目中的 cosSpecifier。由于 nsRole 属性可以是多值属性，因此可以定义具有多个可能模板条目的 CoS 模式。在确定要使用的模板条目时，为了避免出现模棱两可的情况，可以在 CoS 模板条目中包含cosPriority 属性。

例如，可以创建一个允许经理角色成员超过标准邮箱配额的 CoS。该经理角色如下：

```
dn: cn=ManagerRole,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsFilteredRoleDefinition
cn: ManagerRole
nsRoleFilter: (isManager=True)
Description: filtered role for managers
```

传统 CoS 定义条目的创建方式如下：

```
dn: cn=generateManagerQuota,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosClassicDefinition
cosTemplateDn: cn=managerCOS,ou=People,dc=example,dc=com
cosSpecifier: nsRole
cosAttribute: mailboxquota override
```

CoS 模板名称必须是 cosTemplateDn 和 nsRole 值（角色的 DN）的组合。例如：

```
dn:cn="cn=ManagerRole,ou=People,dc=example,dc=com",\
 cn=managerCOS,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: extensibleobject
objectclass: cosTemplate
mailboxquota: 1000000
```

CoS 模板条目提供了 mailboxquota 属性值。附加的限定符 override 指示 CoS 覆盖目标条目中的任何现有 mailboxquota 属性值。属于该角色的目标条目将具有由该角色和 CoS 生成的已计算属性，例如：

```
$ ldapsearch -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -\
 -b "ou=People,dc=example,dc=com" -s sub "(cn=*Fuentes)"
dn: cn=Carla Fuentes,ou=People,dc=example,dc=comcn: Carla Fuentes
isManager: TRUE...nsRole: cn=ManagerRole,ou=People,dc=example,dc=com
mailboxquota: 1000000
```

注 – 角色条目和 CoS 定义条目应位于目录树中的相同位置，以便在其范围中具有相同的目标条目。CoS 目标条目也应位于相同的位置，以便于查找和维护。

监视 CoS 插件

目录服务器允许您对 CoS 插件的某些方面进行监视。CoS 监视属性存储在 cn=monitor,cn=Class of Service,cn=plugins,cn=config 条目中。有关此条目下每个属性及其所提供信息的详细信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Man Page Reference》。

设置 CoS 日志记录

强制目录服务器在多个适用的定义条目之间进行任意区分时，目录服务器将记录警告消息。此类警告消息使用以下格式：

```
Definition /defDN1/ and definition /defDN2/ compete to provide attribute
 '/type/' at priority /level/
```

当强制目录服务器在多个可能适用的定义条目之间进行任意区分时，您也可以将该服务器配置为记录信息性消息。要执行此操作，请将错误日志设置为包含来自插件的消息。

注 – 由于设置其他日志级别可能会导致日志记录负载过重，因此您可能不希望在生产服务器上设置日志记录。

信息性消息的内容使用以下格式：

```
Definition /defDN1/ and definition /defDN2/ potentially compete
to provide attribute '/type/' at priority /level/
```

然后，您可以选择是否通过在定义条目上适当设置 CoS 优先级来解决这种 CoS 不确定性问题。

维护引用完整性

引用完整性是一种插件机制，可确保条目之间的关系得到维护。有些类型的属性（如组成员身份属性）包含其他条目的 DN。使用引用完整性可确保在删除条目时同时删除包含该条目 DN 的所有属性。

例如，如果从目录中删除用户条目并启用了引用完整性，则服务器也会从该用户所属的所有组中删除该用户。如果未启用引用完整性，则必须由管理员从组中手动删除该用户。如果您要将目录服务器与依赖于目录进行用户和组管理的其他 Sun Java System 产品集成，则此功能十分重要。

引用完整性的工作方式

启用引用完整性插件之后，它将在删除、重命名或移动操作后立即对指定属性执行完整性更新。默认情况下，引用完整性插件处于禁用状态。

只要删除、重命名或移动目录中的用户或组条目，该操作都会记录到引用完整性日志文件中：

instance-path`/logs/referint`

在指定时间（称为**更新时间间隔**）过后，服务器将对已启用引用完整性的所有属性执行搜索，并将搜索结果中的条目与日志文件中存在的已删除或已修改条目的 DN 进行匹配。如果日志文件显示条目已被删除，则会删除相应的属性。如果日志文件显示条目已被更改，则会根据更改情况修改相应的属性值。

启用引用完整性插件的默认配置后，它将在删除、重命名或移动操作后立即对 `member`、`uniquemember`、`owner`、`seeAlso` 和 `nsroledn` 属性执行完整性更新。但是，您也可以根据自己的需求配置引用完整性插件的行为。可以配置以下行为：

- 在不同文件中记录引用完整性更新。
- 修改更新时间间隔。

  如果要减小引用完整性更新对系统的影响，您可能需要增大更新的时间间隔。
- 选择要应用引用完整性的属性。

  如果使用或定义包含 DN 值的属性，您可能希望引用完整性插件监视这些属性。

▼ 配置引用完整性插件

注 – 必须为引用完整性插件使用的所有数据库中的所有属性编制索引。需要在所有数据库的配置中创建这些索引。启用追溯更改日志后，必须为 `cn=changelog` 后缀编制索引。有关信息，请参见第 12 章。

某些限制与在复制环境下使用引用完整性插件相关。有关这些限制的列表，请参见第 231 页中的 “复制和引用完整性”。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1. **确保已配置所有副本，并且已定义所有复制协议。**
2. **确定要维护引用完整性的属性集，以及要在主服务器上使用的更新时间间隔。**
3. **使用相同的属性集和相同的更新时间间隔在所有主服务器上启用引用完整性插件。**
   - 要定义引用完整性的属性，请使用以下命令：

     ```
     $ dsconf set-server-prop -h host -p port ref-integrity-attr:attribute-name
     ```
   - 要定义引用完整性更新时间间隔，请使用以下命令：

     ```
     $ dsconf set-server-prop -h host -p port ref-integrity-check-delay:duration
     ```
   - 要启用引用完整性，请使用以下命令：

     ```
     $ dsconf set-server-prop -h host -p port ref-integrity-enabled:on
     ```
4. **确保在所有使用方服务器上禁用引用完整性插件。**

◆ ◆ ◆ 第 10 章

10 目录服务器复制

复制是一种机制，通过此机制可将某个目录服务器中的目录内容自动复制到一个或多个其他目录服务器。所有写入操作都将自动映射到其他目录服务器。有关复制概念、复制方案以及如何在目录部署中规划复制的完整描述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Deployment Planning Guide》。

在复制拓扑中，通常会将服务器上的某个后缀复制到服务器上的其他后缀，或者从其他后缀复制某个后缀。因此，可以交替使用副本、复制的后缀和复制的服务器等术语。

本章介绍通过命令行设置各种复制方案时所执行的任务，其中包含以下主题：

规划复制部署

可以使用任意数量的主服务器配置复制部署。部署中不要求包含集线器或使用方。本章包含为集线器和使用方配置复制的过程，但这些过程是可选的。

开始配置复制之前，您需要清楚地了解在组织中部署复制的方式。您必须了解《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中所述的复制概念。此外，您还必须使用《Sun Java System Directory Server Enterprise Edition 6.0 Deployment Planning Guide》中提供的设计准则仔细规划未来的复制配置。

用于配置和管理复制的推荐界面

配置和管理复制的最简便方法是使用目录服务控制中心 (Directory Service Control Center, DSCC)。使用 DSCC 可以自动配置复制。您可以选择设置复制拓扑所需的自动化级别，例如，是否要在复制配置期间初始化后缀。DSCC 还提供了错误检查功能。此外，DSCC 还提供复制拓扑的图形视图。

DSCC 联机帮助提供了使用 DSCC 设置复制的过程。

注 – 只有在无法使用 DSCC 配置复制时，才需使用本章提供的命令行过程。

配置复制的步骤摘要

第 212 页中的 “配置复制的步骤摘要”假定您要复制单个后缀。如果要复制多个后缀，则可以在每个服务器上并行配置这些后缀。换句话说，您可以重复每个步骤以便在多个后缀上配置复制。

本章的其余部分包含有关如何配置复制的详细说明。

▼ 配置复制的步骤摘要

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

要配置任何复制拓扑，请执行以下过程中所述的常规步骤。

1. **在包含专用使用方副本的所有服务器上执行以下操作：**

   a. **为使用方复制后缀创建空后缀。**

      请参见第 214 页中的 “为使用方副本创建后缀”。

b. **启用使用方复制后缀。**
请参见第 215 页中的 “启用使用方副本”。

c. **（可选的）配置高级使用方设置。**
请参见第 215 页中的 “执行高级使用方配置”。

2 **在包含集线器复制后缀的所有服务器上执行以下操作（如果适用）：**

a. **为集线器复制后缀创建空后缀。**
请参见第 216 页中的 “为集线器副本创建后缀”。

b. **启用集线器复制后缀。**
请参见第 216 页中的 “启用集线器副本”。

c. **（可选的）配置高级集线器设置。**
请参见第 217 页中的 “在集线器副本上修改更改日志设置”。

3 **在包含主服务器复制后缀的所有服务器上执行以下操作：**

a. **为主服务器复制后缀创建后缀。**
请参见第 217 页中的 “为主副本创建后缀”。

b. **启用主服务器复制后缀。**
请参见第 217 页中的 “启用主副本”。

c. **（可选的）配置高级主服务器设置。**
请参见第 218 页中的 “修改主副本上的更改日志设置”。

注 – 请确保在创建复制协议之前启用所有副本，以便您可以在创建复制协议之后立即初始化使用方副本。使用方初始化始终是设置复制的最后一步操作。

---

4 **确保复制管理员配置已完成。**

- 如果您计划使用默认管理员，请在所有服务器上设置默认复制管理员密码。请参见第 220 页中的 “更改默认的复制管理员密码”。
- 如果您计划使用非默认的复制管理员，请在所有服务器上定义其他的复制管理员条目。请参见第 218 页中的 “使用非默认复制管理员”。

5 **在所有主副本上创建复制协议，如下所示：**

a. **在多主拓扑中的主服务器之间**

b. **在主服务器与其专用使用方之间**

c. **在主服务器和集线器副本之间**

请参见第 221 页中的 “创建复制协议”。

6 **（可选的）如果要使用部分复制，请立即进行配置。**

请参见第 222 页中的 “部分复制”。

7 **（可选的）如果要使用复制优先级，请立即进行配置。**

请参见第 223 页中的 “复制优先级”。

8 **配置集线器副本与其使用方之间的复制协议。**

请参见第 221 页中的 “创建复制协议”。

9 **对于多主复制，请从包含数据原始副本的同一主副本初始化所有主服务器。**

请参见第 224 页中的 “初始化副本”。

10 **初始化集线器和使用方副本。**

请参见第 224 页中的 “初始化副本”。

在专用使用方上启用复制

专用使用方是复制后缀的只读副本。专用使用方从绑定为复制管理员的服务器接收更新以进行更改。配置使用方服务器的过程包括准备空后缀以保存复制后缀，以及在该后缀上启用复制。可选的高级配置可包括设置引用、更改清除延迟和修改属性。

以下部分说明如何在服务器上配置一个专用使用方复制后缀。请在将要包含专用使用方复制后缀的每个服务器上重复所有过程。

▼ 为使用方副本创建后缀

- **如果不存在空后缀，请使用与预期主副本相同的 DN 在使用方上创建此后缀。**

  有关说明，请参见第 56 页中的 “创建后缀”。

注意 – 如果该后缀存在且不为空，其内容会在从主服务器初始化复制后缀时丢失。

▼ 启用使用方副本

创建空后缀之后，您需要启用使用方复制后缀。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

- **启用使用方复制后缀。**

```
$ dsconf enable-repl -h host -p port consumer suffix-DN
```

例如：

```
$ dsconf enable-repl -h host1 -p 1389 consumer dc=example,dc=com
```

▼ 执行高级使用方配置

如果要为使用方复制后缀配置高级功能，请立即执行此操作。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1. **如果要使用 SSL 进行引用，请设置安全引用。**

```
$ dsconf set-suffix-prop -h host -p port suffix-DN referral-url:ldaps://servername:port
```

例如：

```
$ dsconf set-suffix-prop -h host1 -p 1636 dc=example,dc=com \
 referral-url:ldaps://server2:2389
```

复制机制会自动将使用方配置为返回复制拓扑中所有已知主服务器的引用。这些默认引用假定客户端将通过常规连接使用简单验证。如果允许客户端选择使用 SSL 安全连接绑定到主服务器，请添加 `ldaps://` *servername* `:`*port* 格式的引用，该引用使用安全的**端口**号。请注意，如果将主服务器配置为只能使用安全连接，则默认情况下 URL 将指向安全端口。

如果已添加一个或多个 LDAP URL 作为引用，则可以强制使用方专门发送这些 LDAP URL 的引用，而不发送主副本的引用。例如，假定您希望客户端始终被引用到主服务器上的安全端口，而不是默认端口。可以创建这些安全端口的 LDAP URL 列表，并设置属性以使用这些引用。如果您要指定特定的主服务器或目录服务器代理来处理所有更新，还可以使用专用引用。

2. **如果您要更改使用方的复制清除延迟，请使用以下命令：**

```
$ dsconf set-suffix-prop -h host -p port suffix-DN repl-purge-delay:time
```

例如，要将清除延迟设置为 2 天，请键入：

```
$ dsconf set-suffix-prop -h host1 -p 1389 edc=example,dc=com repl-purge-delay:2d
```

使用方服务器会存储有关复制后缀内容更新的内部信息，而清除延迟参数可指定该服务器必须保留此信息的时间。清除延迟可部分确定使用方与主服务器之间的复制在中断多长时间后仍可正常恢复。它与提供方服务器更改日志的 MaxAge 参数相关。两个参数中时间较短的参数将确定两个服务器间的复制可以禁用或关闭的最长时间（在此时间过后复制仍可正常恢复）。在大多数情况下，默认值 7 天已经足够。

在集线器上启用复制

集线器副本既作为使用方又作为主服务器，可进一步将复制数据分配给更多的使用方。集线器副本接收来自提供方的复制更新，并将复制更新发送到使用方。这些副本不接受修改，但会返回对主服务器的引用。

配置集线器服务器的过程包括准备空后缀以保存复制后缀，以及在该后缀上启用复制。可选的高级配置可包括选择不同复制管理员、设置引用、设置清除延迟，以及修改更改日志参数。

以下部分说明如何配置一个集线器服务器。请在将要包含集线器复制后缀的每个服务器上重复所有过程。

▼ 为集线器副本创建后缀

- **如果不存在空后缀，请使用与预期主副本相同的 DN 在集线器服务器上创建此后缀。**

  有关说明，请参见第 56 页中的 “创建后缀” 。

  如果该后缀存在且不为空，其内容会在从主服务器初始化复制后缀时丢失。

▼ 启用集线器副本

如果您有集线器副本，请立即启用。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

- **启用集线器复制后缀。**

  ```
  $ dsconf enable-repl -h host -p port hub suffix-DN
  ```

  例如：

  ```
  $ dsconf enable-repl -h host1 -p 1389 hub dc=example,dc=com
  ```

▼ 在集线器副本上修改更改日志设置

对于高级集线器配置，您可能需要修改的参数与更改日志相关。作为提供方，集线器服务器需要更改日志。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **要修改集线器上的更改日志设置，请使用以下任一命令：**

```
$ dsconf set-server-prop -h host -p port suffix-DN repl-cl-max-age:value

$ dsconf set-server-prop -h host -p port suffix-DN repl-cl-max-entry-count:value
```

在主副本上启用复制

主副本包含数据的主拷贝，并在将更新传播到所有其他副本之前集中所有修改。主服务器会记录所有更改、检查使用方的状态，并在必要时向使用方发送更新。在多主复制中，主副本还会从其他主服务器接收更新。

配置主服务器的过程包括定义包含主副本的后缀、启用主副本，以及在必要时为其配置高级复制。

以下部分说明如何配置一个主服务器。请在将要包含主服务器复制后缀的每个服务器上重复所有过程。

▼ 为主副本创建后缀

- **请在将要包含复制条目的主服务器上选择或创建后缀。**

  有关说明，请参见第 56 页中的 “创建后缀”。

  为了确保多主配置和初始化正确，请只将数据装入一个主服务器。其他复制后缀上的所有数据都将被覆盖。

▼ 启用主副本

在主服务器上启用复制时，您必须指定复制 ID。复制 ID 用于区分更新语句的所有者，以及解决多主复制中可能发生的冲突。因此，复制 ID 对于此后缀的所有主副本必须是唯一的。复制 ID 一旦设置便不得更改。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **启用主服务器复制后缀。**

  ```
  $ dsconf enable-repl -h host -p port -d ReplicaID master suffix-DN
  ```

  其中 *ReplicaID* 是 1 到 65534 之间的整数。

  例如，要创建副本 ID 为 1 的主服务器复制后缀，请使用以下命令：

  ```
  $ dsconf enable-repl -h host1 -p 1389 -d 1 master dc=example,dc=com
  ```

▼ 修改主副本上的更改日志设置

如果是高级主服务器配置，您可能需要修改更改日志设置。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **如果要修改主服务器上的更改日志设置，请使用以下任一命令：**

  ```
  $ dsconf set-server-prop -h host -p port suffix-DN repl-cl-max-age:value
  ```

  ```
  $ dsconf set-server-prop -h host -p port suffix-DN repl-cl-max-entry-count:value
  ```

配置复制管理员

本部分介绍如何配置非默认的复制管理员，以及如何设置默认的复制管理员密码。

使用非默认复制管理员

复制管理员是一个用户，提供方在发送复制更新时将使用此用户绑定到使用方服务器。包含接收更新的后缀的所有服务器都必须至少有一个复制管理员条目。

目录服务器有一个默认的复制管理员条目，您可以在每个服务器上使用此条目（特别是在简单复制方案中）：`cn=replication manager,cn=replication,cn=config`。复制机制将自动使用此用户配置使用方副本，从而简化了副本部署。

如果您的复制方案比较复杂，则可能需要多个复制管理员，并且针对每个复制后缀需要使用不同的密码。可以使用一个或多个新的复制管理员替换现有的默认复制管理员。

注意 – 请勿在服务器上使用复制管理员的 DN 和密码进行绑定或执行操作。复制管理员仅用于复制机制。任何其他用途可能都需要重新初始化副本。

请勿将目录管理员用作复制管理员。由于 cn=admin,cn=Administrators,cn=config 条目用于执行其他管理任务，因此也不得将此用户或管理员组中的任何其他用户用作复制管理员。

为每个使用方选择了复制管理员之后，请务必记住所选择或所创建的复制管理员 DN。稍后在提供方上创建与此使用方之间的复制协议时，需要使用此 DN 及其密码。

▼ 设置非默认复制管理员

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1 在所有使用方（目标）复制后缀上，创建新的复制管理员和密码。

```
$ ldapmodify -a -h host -p port -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn:"cn=new-replication-manager,cn=replication,cn=config"
objectclass: top
objectclass: person
userpassword:password
sn:new-replication-manager
```

例如：

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn:"cn=ReplicationManager3,cn=replication,cn=config"
objectclass: top
objectclass: person
userpassword:secret
sn:ReplicationManager3
```

2 在所有使用方（目标）复制后缀上，设置复制管理员绑定 DN。

```
$ dsconf set-suffix-prop -h host -p port suffix-DN \
 repl-manager-bind-dn:"cn=new-replication-manager,cn=replication,cn=config"
```

例如：

```
$ dsconf set-suffix-prop -h host1 -p 1389 dc=example,dc=com \
 repl-manager-bind-dn:"cn=ReplicationManager3,cn=replication,cn=config"
```

3 **针对已在所有提供方（源）复制后缀上创建的所有复制协议，设置复制管理员绑定 DN。**

a. 创建用于设置新复制管理员密码的临时文件。

此文件只读取一次，并存储密码以供将来使用。

```
$ echo password > password-file
```

b. 设置执行更新时复制机制要使用的复制管理员绑定 DN 和密码。

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN host:port \
 auth-bind-dn:"cn=new-replication-manager,cn=replication,cn=config" \
 auth-pwd-file:password-file
```

例如：

```
$ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389 \
 auth-bind-dn:"cn=ReplicationManager3,cn=replication,cn=config" \
 auth-pwd-file:pwd.txt
```

c. 删除临时密码文件。

```
$ rm password-file
```

▼ 更改默认的复制管理员密码

1 **创建用于设置复制管理员密码的临时文件。**

将读取此文件一次，并存储密码以供将来使用。

```
$ echo password > password-file
```

2 **在复制拓扑中的所有使用方（目标）服务器上，设置复制管理员绑定密码。**

```
$ dsconf set-server-prop -h host -p port suffix-DN def-repl-manager-pwd-file:password-file
```

例如：

```
$ dsconf set-server-prop -h host1 -p 1389 dc=example,dc=com \
 def-repl-manager-pwd-file:pwd.txt
```

3 **删除临时密码文件。**

```
$ rm password-file
```

创建复制协议

复制协议是提供方上的一组参数，用于配置和控制将更新发送到给定使用方的方式。必须在向使用方发送更新的提供方复制后缀上创建复制协议。您必须在提供方上为每个要更新的使用方创建复制协议。

▼ 创建复制协议

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

如果使用 DSCC 创建新的复制协议，则可以选择复制现有复制协议中的部分或全部复制协议配置设置。

1 在主服务器上，为要复制到的每个使用方创建复制协议。

```
$ dsconf create-repl-agmt -h host -p port suffix-DN consumer-host:consumer-port [consumer-host:consumer-port]
```

例如：

```
$ dsconf create-repl-agmt -h host1 -p 1389 dc=example,dc=com host2:1389
```

要使用命令行列出现有的复制协议，请使用 `dsconf list-repl-agmts` 命令。

注 – 如果在复制运行时更改主服务器上的端口号，则不必重新初始化服务器。但是，指向旧地址 (*host*:*old-port*) 的旧复制协议将不再有用。如果希望复制继续像更改端口号之前一样运行，则必须使用新地址 (*host*:*new-port*) 创建一个新协议。

---

2 检查是否已正确创建复制协议。

```
$ dsconf show-repl-agmt-status -h host -p port suffix-DN consumer-host:consumer-port
```

3 如果验证状态不是 OK，请运行 `dsconf accord-repl-agmt` 命令。

注 – 只有使用默认复制管理员时，才应使用命令 `dsconf accord-repl-agmt`。 如果您已创建新的复制管理员，请不要使用此命令，因为它将覆盖某些必需设置。

`dsconf accord-repl-agmt` 命令可确保提供方和目标服务器共享相同的复制验证设置。

```
$ dsconf accord-repl-agmt -h host -p port suffix-DN consumer-host:consumer-port
```

例如：

```
$ dsconf accord-repl-agmt -h host2 -p 1389 dc=example,dc=com host1:1389
```

部分复制

默认情况下，复制操作会将复制后缀中的所有条目复制到使用方副本中。使用部分复制功能，您可以选择要使用的后缀，以及要包括或排除的属性。部分复制是在复制协议中配置的，允许您为主服务器的每个使用方复制后缀定义属性集。您可以控制要分配的数据，并可更有效地使用复制带宽和使用方资源。

例如，如果要减小复制带宽，则可以选择不复制 `photo`、`jpegPhoto` 和 `audio` 之类的属性，因为这些属性通常具有较大的值。因此，在使用方上将无法使用这些属性。另外一种情况是，您可以选择只将 `uid` 和 `userpassword` 属性复制到专用于执行验证的使用方服务器。

部分复制的注意事项

注 – 部分复制无法用于 Directory Server 5.2 之前版本的产品中。配置部分复制协议时，主副本和使用方副本必须至少使用 Directory Server 5.2。

启用或修改部分属性需要您重新初始化使用方副本。因此，您需要在部署之前确定部分复制需求，并在首次初始化复制后缀之前定义属性集。

如果某些复杂功能（如 ACI、角色和 CoS）依赖于特定属性，则您在复制小型属性集时需要特别谨慎。此外，不复制 ACI、角色或 CoS 机制的说明符或过滤器中所提到的其他属性可能会对数据安全性造成威胁。不复制这些属性可能还会导致在搜索中返回不同的属性集。管理要排除的属性列表比管理要包含的属性列表更加安全，并且更不容易出现人为错误。

如果您所复制的属性集不允许所有复制条目遵循模式，则需要在使用方服务器中关闭模式检查。复制不符合模式的条目不会引发错误，因为复制机制将避开使用方上的模式检查。但是，使用方将包含不符合模式的条目，并且需要关闭模式检查，以便向客户端显示一致状态。

将在主副本与集线器和专用使用方之间的复制协议中配置部分复制。不支持在多主复制环境中的两个主副本之间配置部分复制。此外，如果多个主服务器与同一副本之间具有多个复制协议，则所有这些协议都必须复制相同的属性集。

▼ 配置部分复制

要配置部分复制，必须先指定后缀，再确定要在该后缀上包含属性还是排除属性，然后选择要包含或排除的属性。如果选择在后缀上排除属性，将自动包含所有其他属性。同样，如果选择在后缀上包含某些属性，将自动排除所有其他属性。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **在位于源服务器的复制协议上配置部分复制。**

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN consumer-host:consumer-port property:value
```

其中 *property* 为 `repl-fractional-exclude-attr` 或 `repl-fractional-include-attr`。

例如，如果要将部分协议配置为不在 `dc=example,dc=com` 后缀上复制 JPEG 和 TIFF 照片，请使用以下命令：

```
$ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389
 repl-fractional-exclude-attr:jpegPhoto repl-fractional-exclude-attr:tiffPhoto
```

复制优先级

指定复制优先级为可选操作。可以创建复制规则，以指定使用高优先级复制某些更改（如更新用户密码）。复制规则中指定的任何更改都将以高优先级进行复制，而所有其他更改都将以普通优先级进行复制。

注 – 只需在主服务器上创建复制优先级规则。不必为集线器和使用方进行配置。

▼ 配置复制优先级

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **要在主服务器上创建新的复制优先级规则，请使用以下命令：**

```
$ dsconf create-repl-priority -h host -p port suffix-DN priority-name property:value
```

可以使用以下一个或多个属性设置复制优先级：

- 操作类型，`op-type`
- 绑定 DN，`bind-dn`
- 基 DN，`base-dn`
- 属性类型，`attr`

priority-name 是用户定义的属性。

例如，要创建复制规则以指定使用高优先级复制用户密码更改，请使用以下命令：

```
$ dsconf create-repl-priority -h host2 -p 1389 dc=example,dc=com pw-rule \
 attr:userPassword
```

要显示当前的复制规则，请使用 `dsconf list-repl-priorities -v` 命令。使用 `-v` 选项时，此命令将显示与按优先级排序的复制规则相关的其他信息。

```
$ dsconf list-repl-priorities -h host2 -p 1389 -v
```

有关详细信息，请参见 dsconf(1M) 手册页。

初始化副本

创建复制协议并已配置两个副本之后，必须先初始化使用方复制后缀，然后才能开始复制。在初始化期间，您实际将提供方复制后缀中的数据复制到使用方复制后缀。

此外，某些错误条件或配置更改也需要重新初始化副本。例如，如果由于任何原因从备份恢复了单个主服务器复制后缀中的数据，则需要对它所更新的所有副本进行重新初始化。

进行重新初始化时，将在使用方上删除复制后缀的内容，并将其替换为主服务器上的后缀内容。这可确保副本保持同步，并且可以恢复复制更新。本部分介绍的所有初始化方法都会自动重新生成使用方副本的索引，以便使用方能以最佳方式响应客户端读取请求。

使用多主复制时，如果使用方已经由拓扑中的其他主服务器进行了更新，则可能不需要进行重新初始化。

▼ 从远程（提供方）服务器初始化复制后缀

可以使用现有的复制协议从远程服务器初始化后缀。请尽可能使用此方法进行初始化，因为它要比其他方法简单。仅当存在大量数据使得导入操作耗时过长时，才应使用其他方法。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

使用 DSCC 以联机方式对复制后缀进行初始化是初始化或重新初始化使用方的简便方法。但是，如果要初始化大量条目，则此过程可能会非常耗时。在这种情况下，使用命令行以脱机方式初始化使用方可能会更加有效。

1 初始化副本。

```
$ dsconf init-repl-dest -h host -p port suffix-DN destination-host:destination-port [destination-host:destination-port]
```

其中 *destination-host*:*destination-port* 是要从远程服务器进行初始化的目标服务器的主机和端口。

2 （可选的）针对每个协议检查后缀是否已初始化。

```
$ dsconf show-repl-agmt-status -h host -p port suffix-DN destination-host:destination-port
```

从 LDIF 进行副本初始化

▼ 从 LDIF 初始化复制后缀

此过程介绍从 LDIF 文件初始化复制后缀时所使用的一般步骤。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

使用 DSCC 以联机方式对复制后缀进行初始化是初始化或重新初始化使用方的简便方法。但是，如果要初始化大量条目，则此过程可能会非常耗时。在这种情况下，使用命令行以脱机方式初始化使用方可能会更加有效。

1. **确保已设置了复制协议。**

   必须在初始化副本**之前**执行此操作。

2. **将主服务器复制后缀中的后缀数据的原始副本导出到 LDIF 文件。**

   请参见第 226 页中的 “将复制后缀导出到 LDIF”。

   在多主复制环境中，可以使用从原始主服务器导出的 LDIF 文件来初始化其他主服务器和所有使用方。在级联复制环境中，可以使用相同的文件来初始化集线器副本及其使用方。

   在任何情况下，都必须以 LDIF 文件（从已配置的主副本中导出）开始。无法使用任意的 LDIF 文件初始化所有副本，因为它不包含复制元数据。

3. **如果要初始化部分副本，请过滤此文件以便只保留复制的属性，然后将该文件传送到所有使用方服务器。**

   请参见第 226 页中的 “为部分复制过滤 LDIF 文件”。

4. **初始化副本。**

   执行以下任一操作：

   - 要在已脱机（停止）的服务器上进行快速初始化，请使用 `dsadm import` 命令。

     ```
     $ dsadm import instance-path LDIF_file suffix-DN
     ```

   - 要从 LDIF 文件以联机方式初始化副本，请使用 `dsconf import` 命令。

     ```
     $ dsconf import -h host -p port LDIF_file suffix-DN
     ```

     使用 `dsconf import` 比使用 `dsadm import` 要慢，但不必在执行导入操作时停止服务器。

   有关初始化后缀的更多详细信息以及示例，请参见第 185 页中的 “初始化后缀”。有关命令用法的详细信息，请参见 `dsadm`(1M) 和 `dsconf`(1M)。

5 （可选的）针对每个协议检查后缀是否已初始化。

```
$ dsconf show-repl-agmt-status -h host -p port suffix-DN destination-host:destination-port
```

▼ 将复制后缀导出到 LDIF

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **使用以下任一命令导出 LDIF 文件中的复制后缀内容：**
  - 要以脱机方式导出，请键入：

    ```
    $ dsadm export instance-path suffix-DN LDIF_file
    ```
  - 要以联机方式导出，请键入：

    ```
    $ dsconf export -h host -p port suffix-DN LDIF_file
    ```

以下示例将整个 `dc=example,dc=com` 复制后缀及复制信息导出到 `example_replica_export.ldif` 文件中：

```
$ dsconf export -h host2 -p 1389 dc=example,dc=com  \
 /local/ds/ldif/example_export_replica.ldif
```

有关详细信息，请参见第 182 页中的 “备份到 LDIF”以及 `dsadm`(1M) 和 `dsconf`(1M) 手册页。

为部分复制过滤 LDIF 文件

使用 DSCC 时，对配置了部分复制的副本进行初始化的过程是不可视的。在初始化期间只会将选定属性发送给使用方。

如果已经配置了部分复制，应该在将导出的 LDIF 文件复制到使用方服务器之前过滤出所有未使用的属性。为此，目录服务器提供了 `fildif` 工具。此工具将过滤给定的 LDIF 文件，以便只保留复制协议中定义的属性集所允许的属性。

此工具将读取服务器配置以确定属性集定义。要读取配置文件，必须以超级用户身份或拥有此过程和文件（由 `nsslapd-localuser` 属性指定）的用户身份运行 `fildif` 工具。例如，以下命令将过滤前面示例中从 `dc=example,dc=com` 后缀导出的文件：

```
$ fildif -i /local/ds1/ldif/example_master.ldif \
 -o /local/ds1/ldif/filtered.ldif -b "cn=host2.example.com:1389, \
 cn=replica,cn=\\"dc=example,dc=com\\",cn=mapping tree,cn=config" -p /local/ds1
```

有关 `fildif` 命令的位置，请参见第 31 页中的 “命令位置”。

`-i` 和 `-o` 选项分别是输入和输出文件。`-b` 选项是定义部分复制的复制协议的 DN。可使用以下命令查找此 DN：

```
$ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
 -b "cn=config" "(&(objectclass=nsds5replicationagreement) (nsDS5ReplicaPort=replica-port) \
 (nsDS5ReplicaHost=replica-host))" dn
```

例如：

```
$ ldapsearch -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w - \
 -b "cn=config" "(&(objectclass=nsds5replicationagreement) \
 (nsDS5ReplicaPort=2090)(nsDS5ReplicaHost=host2))" dn
Enter bind password:
version: 1
dn: cn=host2:1389,cn=replica,cn=dc\=example\,dc\=com,cn=mapping tree,cn=config
```

有关 `fildif` 工具的完整命令行语法，请参见 `fildif`(1) 手册页。

接下来，可以使用由 `fildif` 生成的 `filtered.ldif` 文件初始化此复制协议中的使用方。将此文件传送到使用方服务器并进行导入，如第 184 页中的 “从 LDIF 文件导入数据”中所述。

使用二制进副本初始化复制后缀

二进制副本允许您使用一个服务器中的二进制备份文件克隆整个服务器，以便将相同的目录内容恢复到另一个服务器上。使用二进制副本，您可以通过主服务器或集线器服务器的二进制副本初始化或重新初始化任何服务器，或者通过其他使用方服务器的二进制副本初始化或重新初始化使用方。

注 – 此高级过程将与目录服务器的数据库文件进行交互，因此只有经验丰富的管理员才应该使用。

由于对此功能进行了某些限制，因此对于包含很大数据库文件的副本（例如，包含数百万条目的副本），此功能非常实用和省时。

将二进制副本用于复制的限制

由于二进制副本将一台计算机上的数据库文件移动到另一台计算机上，因此该机制应遵循以下严格限制：

- 两台计算机必须运行相同的操作系统，包括所有服务包 (service pack) 或修补程序。
- 两台计算机必须共享相同的处理器体系结构。例如，可以在两台 UltraSPARC® T1 处理器之间执行二进制副本，但无法在一台 UltraSPARC T1 处理器和一台 AMD Opteron 处理器之间执行二进制副本。
- 两台计算机必须都是大端字节序或都是小端字节序。
- 两台计算机必须以相同方式映射内存。例如，可以在两个 64 位系统上的服务器实例之间执行二进制副本，但无法在 32 位系统上的一个服务器实例和 64 位系统上的另一个服务器实例之间执行二进制副本。

- 两台计算机必须安装相同版本的目录服务器，包括二进制格式（32 位或 64 位）、服务包 (service pack) 和修补程序级别。
- 两个服务器必须具有划分为相同后缀的相同目录树。**所有**后缀的数据库文件都**必须**一起复制。无法复制单个后缀。
- 每个后缀必须在两个服务器上配置相同的索引，包括 VLV（Virtual List View，虚拟列表视图）索引。这些后缀的数据库必须具有相同的名称。
- 每个服务器都必须将相同的后缀配置为副本。
- 如果配置部分复制，则必须在所有服务器上进行完全相同的配置。
- 不得在任一服务器上使用属性加密。
- 如果启用属性值唯一性插件，则它在两个服务器上必须具有相同的配置，而且必须在新副本上重新配置该插件，如以下过程所述。

  以下过程介绍执行二进制副本的其他方法：不需要停止服务器的二进制副本，以及使用最少磁盘空间的二进制副本。

创建用于初始化服务器的二进制副本

本部分介绍如何创建用于初始化服务器的二进制副本，以及如何创建使用最少磁盘空间的二进制副本。

▼ 创建用于初始化服务器的二进制副本

可以使用此过程执行用于初始化复制服务器的二进制副本，因为它使用标准备份功能创建服务器数据库文件的副本。执行标准备份可确保所有数据库文件处于一致状态，而无需停止服务器。

此过程具有某些限制。由于备份和恢复操作将在同一台计算机上创建数据库文件的副本，因此在每台计算机上，这些文件所需的磁盘空间量都会加倍。此外，如果目录中包含数千兆字节的数据，则对这些文件执行实际的复制操作可能会耗费大量时间。

对于此过程的某些部分，可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的“目录服务控制中心界面”和 DSCC 联机帮助。此过程的其他部分只能使用命令行完成。

1. **在新复制后缀的目标计算机上安装目录服务器，创建新的服务器实例（如有必要），然后按照第 227 页中的“将二进制副本用于复制的限制”中所述配置此服务器。**

2. **在包含此复制后缀的复制拓扑中创建所有复制协议。**

   请在此副本中包含来自提供方的协议。如果此副本不是专用使用方，请在其使用方中包含来自此副本的协议。请参见第 221 页中的“创建复制协议”。

3. **选择经过完整配置和初始化的副本，此副本的类型与您要初始化的类型相同（主服务器、集线器或使用方），然后按照第 179 页中的“二进制备份”中所述在此副本上执行标准备份。**

4 **将备份目录中的文件复制或传送到目标计算机上的目录中，例如，可以使用 `ftp` 命令完成此操作。**

5 **如果您在多主复制方案中对新的主服务器进行了初始化，请执行第 189 页中的 “ 在多主方案中恢复主服务器 ”中的过程。**

▼ 使用需要最少磁盘空间的二进制副本初始化服务器

此过程将使用较少的磁盘空间和时间，因为它不创建数据库文件的备份副本。但是，它需要停止要克隆的服务器，以确保数据库文件处于一致状态。

注意 – 不得使用此过程对已经加入多主复制方案的主服务器进行重新初始化。它只能用于重新初始化使用方服务器或初始化新的主服务器。要重新初始化现有的主副本，请使用联机初始化、导入 LDIF 文件或执行第 228 页中的 “创建用于初始化服务器的二进制副本”中的过程。

对于此过程的某些部分，可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。此过程的其他部分只能使用命令行完成。

1 **在新复制后缀的目标计算机上安装目录服务器，创建新的服务器实例（如有必要），然后按照第 227 页中的 “ 将二进制副本用于复制的限制 ”中所述配置此服务器。**

2 **在包含此副本的复制拓扑中创建所有复制协议。**

请在此副本中包含来自提供方的协议。如果此副本不是专用使用方，请在其使用方中包含来自此副本的协议。请参见第 221 页中的 “创建复制协议”。

3 **停止将要初始化或重新初始化的目标服务器，如第 55 页中的 “ 启动、停止和重新启动目录服务器实例 ”中所述。**

4 **选择经过完整配置和初始化的副本，此副本的类型与您要初始化的类型相同（主服务器、集线器或使用方），同时停止此服务器。**

如果要克隆多主配置中的主副本，请确保此主副本在停止之前处于最新状态，即包含来自其他主服务器的所有最新更改。

5 **从目标服务器中删除所有数据库文件，包括事务日志、更改日志和区域文件（`__db.`*xxx* `files`）。**

除非这些文件已被重新定位，否则数据库文件和事务日志位于 *instance-path*`/db` 目录中。

6 **将源副本计算机中的所有数据库文件（包括事务日志和更改日志）复制或传送到目标计算机，例如，可以使用 `ftp` 命令完成此操作。**

除非这些文件已被重新定位，否则数据库文件和事务日志位于 *instance-path*`/db` 目录中。

如果要初始化主服务器或集线器副本，还必须复制更改日志中的所有文件，默认情况下，更改日志位于 *instance-path* `/changelog` 中。

7 **重新启动源服务器和目标服务器。**

在级联复制中初始化副本

如果使用级联复制，请终始按照以下过程所示的顺序初始化副本。

▼ 在级联复制中初始化副本

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 **如果还有多主复制，请确保一个主服务器具有要复制的完整数据集，然后使用此主服务器初始化每个其他主服务器上的副本。**

2 **从一级集线器副本的主副本初始化一级集线器副本上的副本。**

3 **如果有多个级别的集线器，请从之前初始化的集线器级别来初始化每个级别。**

4 **在最后级别的集线器副本上，初始化专用使用方上的副本。**

为复制后缀编制索引

索引不会自动从一个服务器实例复制到另一个服务器实例。要为保存复制后缀的所有服务器实例编制属性索引，请执行以下任一操作。

- 在 DSCC 中将保存复制后缀的所有服务器实例作为服务器组进行管理。将索引添加到组中的某个服务器上，然后使用“复制服务器配置”操作将索引设置复制到组中的其他服务器。

  有关 DSCC 的详细信息，请参见第 41 页中的 “目录服务控制中心界面”。
- 使用 `dsconf` 命令管理每个服务器实例上的索引，如第 12 章中所述。
- 使用二进制副本初始化后缀，如第 227 页中的 “使用二制进副本初始化复制后缀”中所述。

逐渐向大型复制后缀添加大量条目

如果您有包含大量条目的目录，并且要添加大量条目，请不要使用 `ldapmodify -a`，因为这样做会消耗大量时间。通过在 `dsconf import` 命令中使用用于在复制拓扑中添加条目的选项，可以逐渐添加新条目。导出这些条目时，将生成包含附加内容和复制元数据的 LDIF 文件。然后，可以将生成的此 LDIF 文件导入其他副本中。生成的 LDIF 文件可确保复制在您要添加数据的副本之间始终保持同步。

▼ 向大型复制后缀添加大量条目

开始之前 此过程将生成大型 LDIF 文件。在运行第一个 `dsconf import` 命令之前，请确保具有足够的磁盘空间用于生成的 LDIF 文件。

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

注意 – 可以使用此过程分以多次传递的方式初始化包含大量条目的服务器。但是，如果其中一次导入失败，则可能会丢失整个数据库。请务必在每次导入之前备份数据。

1. **在任何主副本上导入条目。**

   ```
   $ dsconf import -h host -p port -K generated-LDIF-file suffix-DN
   ```

   `-K` 选项确保不会删除现有数据。它还将生成 *generated-LDIF-file* 文件，此文件包含复制过程所需的新条目和信息。

2. **在所有其他副本上，导入上一步生成的文件。**

   ```
   $ dsconf import -h host -p port \
   -K -f incremental-output=no generated-LDIF-file suffix-DN
   ```

   选项 `-f incremental-output=no` 指定将不会生成其他 LDIF 文件。此过程只需要一个生成的 LDIF 文件。

复制和引用完整性

如果要将引用完整性插件用于复制，则必须在所有主服务器上启用此插件。不必在集线器服务器或使用方服务器上启用此插件。

以下限制与在复制环境中使用引用完整性插件有关：

- 必须在包含主副本的所有服务器上启用此插件。
- 必须在每个主服务器上使用相同配置启用此插件。
- 在只包含集线器或使用方副本的服务器上启用此插件不起作用。

有关配置引用完整性插件的信息，请参见第 210 页中的 “配置引用完整性插件”。

通过 SSL 执行复制

可以对复制中涉及的目录服务器进行配置，以便所有复制操作都通过 SSL 连接执行。

▼ 配置使用 SSL 的复制操作

此过程显示的示例命令用于在包含两个主服务器的复制拓扑上设置复制。

注 – 此示例显示了一个简单复制配置（使用自签名证书）。在生产环境中设置通过 SSL 执行的复制时，如果改用由证书颁发机构颁发的可信证书将会更加安全。

如果提供方服务器证书是仅适用于服务器的 SSL 证书（无法在 SSL 握手时作为客户端），将无法通过 SSL 执行复制。

虽然通过 SSL 执行的复制是安全的，仍会使用简单绑定和密码对复制管理员进行验证。可以使用基于客户端的验证来完全保护复制，但这需要更复杂的设置。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 创建并启动新的服务器。

```
$ dsadm create -p 1389 -P 1636 /local/ds1
$ dsadm create -p 2389 -P 2636 /local/ds2

$ dsadm start /local/ds1
$ dsadm start /local/ds2
```

2 在所有服务器上创建空后缀。

```
$ dsconf create-suffix -e -i -p 1389 dc=example,dc=com
$ dsconf create-suffix -e -i -p 2389 dc=example,dc=com
```

3 在所有服务器上设置多主密码文件。

```
$ dsconf set-server-prop -e -i -h example1.server -p 1389 \
 def-repl-manager-pwd-file:/local/ds1/replmanrpwd1.txt
$ dsconf set-server-prop -e -i -h example2.server -p 2389 \
 def-repl-manager-pwd-file:/local/ds1/replmanrpwd2.txt
```

4 在所有服务器上启用复制。

```
$ dsconf enable-repl -h example1.server -p 1389 -e -i -d 1 master dc=example,dc=com
$ dsconf enable-repl -h example2.server -p 2389 -e -i -d 2 master dc=example,dc=com
```

5 **在所有服务器上查看现有的默认证书。**

```
$ dsadm show-cert -F der -o certfile1 /local/ds1 defaultCert
$ dsadm show-cert -F der -o certfile2 /local/ds2 defaultCert
```

6 **在所有服务器上，添加来自所有其他服务器的 CA 可信证书。**

```
$ dsadm add-cert --ca /local/ds1 "ds2 Repl Manager Cert" certfile2
$ dsadm add-cert --ca /local/ds2 "ds1 Repl Manager Cert" certfile1
```

7 **在所有主服务器和集线器（源）服务器上，创建与所有使用方（目标）服务器之间的复制协议。**

请注意，这些复制协议使用安全 LDAP 端口。

```
$ dsconf create-repl-agmt -h example1.server -p 1389 -e -i \
 --auth-protocol "ssl-simple" dc=example,dc=com example2.server:2636
$ dsconf create-repl-agmt -h example2.server -p 2389 -e -i \
 --auth-protocol "ssl-simple" dc=example,dc=com example1.server:1636
```

8 **针对所有复制协议，在复制协议中将验证密码文件配置为使用方（目标）服务器的复制管理员密码文件。**

```
$ dsconf set-repl-agmt-prop -h example1.server -p 1389 -e -i \
 dc=example,dc=com example2.server:2636 auth-pwd-file:/local/ds1/replmanrpwd2.txt
$ dsconf set-repl-agmt-prop -h example2.server -p 2389 -e -i \
 dc=example,dc=com example1.server:1636 auth-pwd-file:/local/ds1/replmanrpwd1.txt
```

初始化这些后缀之后，提供方将通过 SSL 向使用方发送所有复制更新消息，并将使用证书（如果已选择该选项）。如果使用为 SSL 配置的协议通过 DSCC 执行使用方初始化，则使用方初始化也将使用安全连接。

9 **在所有服务器上，重新启动服务器以使配置更改生效。**

```
$ dsadm restart /local/ds1
$ dsadm restart /local/ds2
```

10 **在其中一个主服务器上初始化后缀。**

```
$ dsconf import -h example1.server -p 1389 -e -i /tmp/Example.ldif dc=example,dc=com
```

11 **在尚未初始化的所有服务器上，使用复制协议初始化这些服务器。**

```
$ dsconf init-repl-dest -e -i -h example1.server -p 1389 \
 dc=example,dc=com example1.server:2636
```

通过WAN执行复制

目录服务器允许您执行所有形式的复制，包括通过广域网 (Wide Area Network, WAN) 连接的计算机之间的多主复制。此复制允许提供方服务器在初始化和更新使用方时，在具有较高时延和较低带宽的网络上以最佳方式使用带宽。

注 – 对通过 WAN 复制的复制拓扑进行部署和故障排除时，必须检查网络速度、时延和数据包丢失情况。上述任一方面的网络问题都可能会导致复制延迟。

此外，复制数据传输速率将始终低于可用物理介质所允许的速率（在带宽方面）。如果副本之间的更新量无法实际符合可用带宽，则调整操作将无法阻止各个副本在较重的更新负载下产生差异。复制延迟和更新性能由许多因素决定，包括但不限于以下因素：修改率、条目大小、服务器硬件、错误率、平均时延和平均带宽。

如果您的环境中存在复制方面的问题，请与 Sun 服务提供商联系。

默认情况下，复制机制的内部参数针对 WAN 进行了优化。但是，如果由于以上因素导致您的复制速度很慢，则可能需要根据经验调整窗口大小和组大小参数。您还可以安排复制以避开高峰网络时间，从而改善整个网络的使用情况。最后，目录服务器还支持复制数据压缩，以优化带宽使用。

配置网络参数

窗口和组网络参数确定了复制机制如何对条目进行分组，以便通过网络更有效地发送这些条目。这些参数会影响提供方和使用方交换复制更新消息和确认的方式。可以在每个复制协议中配置这些参数，以便您根据每个使用方的特定网络条件调整复制性能。

请监视您所做的任何修改的效果，并相应地调整这些参数。有关说明，请参见第 244 页中的 “获取复制状态”。您不必中断复制来修改窗口大小和组大小参数。

配置窗口大小

窗口大小（默认值为 `10`）表示在无需使用方立即确认的情况下可以发送的更新消息的最大数目。

与发送每条消息后等待确认相比，快速连续地发送多个消息效率更高。使用适当的窗口大小，可以缩短副本等待复制更新或确认到达所花费的时间。

如果使用方副本落后于提供方，请将窗口大小调整为比默认值更大的值（如 100），并在进一步调整之前再次检查复制性能。当复制更新速率很高而使得更新之间的时间较短时，甚至由局域网 (Local Area Network, LAN) 连接的副本都能从较大的窗口大小中获益。

▼ 配置窗口大小

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **修改窗口大小。**

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN consumer-host:consumer-port transport-window-size:value
```

例如：

```
$ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389 \
 transport-window-size:20
```

配置组大小

组大小（默认值为 1）表示可以绑到单个更新消息中的数据修改的最大数目。如果网络连接似乎要妨碍复制，请将组大小调整为比默认值更大的值（如 10），然后重新检查复制性能。

增加组大小时，请确保满足以下条件：

- 将窗口大小设置为远大于组大小。
- 窗口大小除以组大小所得到的值远大于使用方 `cn=config` 下的 `nsslapd-maxThreadsPerConn` 值（通常为后者的两倍）。

  将组大小设置为大于 1 时，提供方在将更新发送给使用方之前不会等待填充组。

▼ 配置组大小

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **修改组大小。**

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN \
 consumer-host:consumer-port transport-group-size:value
```

例如：

```
$ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389 \
 transport-group-size:10
```

安排复制操作

如果没有必要立即实现副本之间的同步，则可以在网络使用率较低的时段安排复制。当可用的网络资源较多时，完成数据复制的过程应明显加快。

可以将复制安排在一天中的特定时间（基于每天或每周）开始和结束。可以通过每个使用方的复制协议独立为每个使用方执行此操作。新的计划将立即生效，这会导致相应使用方的下一个数据复制出现延迟，直到此计划允许的第一个复制完成为止。

▼ 安排复制操作

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **修改复制计划。**

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN \
 host:port repl-schedule:value
```

例如，如果要将复制设置为在每天凌晨 2:00 到 4:00 之间发生，请键入：

```
$ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389 \
 repl-schedule:"0200-0400 0123456"
```

其中 `0123456` 表示一周内的各天，`0` 表示星期日，`1` 表示星期一，依此类推。

配置复制压缩

要减小复制所使用的带宽，可以将复制配置为在更新使用方时压缩所发送的数据。复制机制使用 Zlib 压缩库。提供方和使用方都必须在 Solaris 或 Linux 平台上运行才能启用压缩。

应该根据经验测试并选择压缩级别，以便在 WAN 环境中使用预期复制时获得最佳结果。请勿在网络带宽很高的 LAN 中设置此参数，因为压缩和解压缩计算会降低复制速度。

▼ 配置复制压缩

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

- **在主服务器中的复制协议条目上配置复制压缩。**

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN \
 consumer-host:consumer-port transport-compression:level
```

其中 *level* 可以为 `high`、`medium`、`low` 或 `none`。

例如，要在将复制更新发送到 `host1:1389` 上的使用方时使用最快速的压缩，请键入：

```
$ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389 \
 transport-compression:high
```

有关设置压缩级别的详细信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》。

修改复制拓扑

本部分从以下几个方面介绍如何管理现有复制拓扑：

更改复制管理员

可以编辑复制协议以更改用于绑定到使用方服务器的复制管理员标识。为了避免复制中断，应该在使用方上定义新的复制管理员条目或证书条目，然后再修改复制协议。但是，如果复制由于绑定失败而中断，复制机制将在您更正错误时自动发送所有必要的更新，此过程将在复制恢复设置的限制下完成。有关过程，请参见第 218 页中的 “使用非默认复制管理员”。

管理复制协议

可以禁用、启用或删除复制协议。

禁用复制协议。

禁用复制协议时，主服务器将停止向指定的使用方发送更新。到该服务器的复制将会停止，但协议中的所有设置将被保留。您可以稍后通过重新启用此协议来恢复复制。有关在复制中断后恢复复制机制的信息，请参见第 238 页中的 “启用复制协议”。

▼ 禁用复制协议

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

- **禁用复制协议。**

  ```
  $ dsconf disable-repl-agmt -h host -p port suffix-DN consumer-host:consumer-port
  ```

  例如：

  ```
  $ dsconf disable-repl-agmt -h host2 -p 1389 dc=example,dc=com host1:1389
  ```

启用复制协议

启用复制协议将恢复与指定使用方之间的复制。但是，如果复制中断的时间已超过复制恢复设置所允许的时间，并且使用方未被其他提供方更新，则必须对该使用方进行重新初始化。复制恢复设置包括此提供方更改日志的最大大小和最大存留期，以及使用方的清除延迟（请参见第 215 页中的 “执行高级使用方配置”）。

如果中断时间很短并且复制可以恢复，则主服务器将在重新启用协议时自动更新使用方。

▼ 启用复制协议

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **启用复制协议。**

```
$ dsconf -h host -p port enable-repl-agmt suffix-DN consumer-host:consumer-port
```

例如：

```
$ dsconf -h host2 -p 1389 enable-repl-agmt dc=example,dc=com host1:1389
```

删除复制协议

删除复制协议将停止到相应使用方的复制，并将删除有关此协议的所有配置信息。如果日后要恢复复制，请禁用此协议，如第 237 页中的 “禁用复制协议。”中所述。

▼ 删除复制协议

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **删除复制协议。**

```
$ dsconf delete-repl-agmt -h host -p port suffix-DN consumer-host:consumer-port
```

例如：

```
$ dsconf delete-repl-agmt -h host2 -p 1389 dc=example,dc=com host1:1389
```

对副本进行升级或降级

对副本进行升级或降级将更改该副本在复制拓扑中的角色。专用使用方可升级为集线器，而集线器可升级为主服务器。主服务器可降级为集线器，而集线器也可降级为专用使用方。但是，主服务器不能直接降级为使用方，同样，使用方也不能直接升级为主服务器。

多主复制机制中所允许的升级和降级功能使得拓扑非常灵活。以前由使用方副本提供服务的站点可能会增大，并且需要具有多个副本的集线器来处理负载。如果负载包括许多副本内容修改，则集线器可以变为主服务器以允许更快速的本地更改，然后可以将这些更改复制到位于其他站点的其他主服务器中。

对副本进行升级或降级时，请注意以下几点：

- 如果升级使用方，使用方将变为集线器。如果升级集线器，集线器将变为主服务器。无法直接将服务器从使用方升级为主服务器。必须首先将使用方升级为集线器，然后再将集线器升级为主服务器。同样，将主服务器降级为使用方时，必须先将主服务器降级为集线器，然后再将集线器降级为使用方。
- 将主服务器降级为集线器时，副本将变为只读副本，并被配置为将引用发送给其余的主服务器。新的集线器将保留其所有使用方，无论这些使用方是集线器还是专用使用方。
- 将单个主服务器降级为集线器将创建无主副本的拓扑。假定您要定义新的主服务器，目录服务器将允许您执行此操作。但是，最好将新的主服务器添加为多主服务器并允许对其进行初始化，然后再对其他主服务器进行降级。
- 将集线器降级为使用方之前，必须禁用或删除出入集线器的所有复制协议。如果不执行此操作，则降级操作将会失败，并出现以下错误：`LDAP_OPERATIONS_ERROR` **“无法在启用某些协议的情况下将集线器降级为只读副本”**。

  如果集线器的使用方未被其他集线器或主服务器更新，将不再对它们进行更新。您应该在其余的集线器或主服务器上创建新协议，以便更新这些使用方。
- 将使用方升级为集线器时，将启用其更改日志，并且您可以定义与这些使用方之间的新协议。
- 将集线器升级为主服务器时，副本将接受修改请求，并且您可以定义与其他主服务器、集线器或专用使用方之间的新协议。

▼ 对副本进行升级或降级

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

● **使用以下任一命令对副本进行升级或降级：**

```
$ dsconf promote-repl -h host -p port role suffix-DN

$ dsconf demote-repl -h host -p port role suffix-DN
```

其中 *role* 为 `master`、`hub` 或 `consumer`。

禁用复制后缀

禁用复制后缀会将该后缀从复制拓扑中删除。根据该后缀的角色（主服务器、集线器或使用方），它将不再接受更新或发送更新。禁用提供方服务器上的后缀将删除所有复制协议，如果再次启用该副本，则必须重新创建这些协议。

▼ 禁用复制后缀

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **禁用复制后缀。**

```
$ dsconf disable-repl -h host -p port suffix-DN
```

例如：

```
$ dsconf disable-repl -h host2 -p 1389 dc=example,dc=com
```

使复制后缀保持同步

停止复制所使用的目录服务器以进行常规维护之后，当服务器重新处于联机状态时，必须确保立即通过复制对该服务器进行更新。对于多主环境中的主服务器，目录信息需要由多主服务器集中的其他主服务器进行更新。在其他情况下，当集线器服务器或专用使用方服务器进入脱机状态以进行维护之后，如果这些服务器重新处于联机状态，则需要通过主服务器对其进行更新。

本部分介绍复制重试算法，并说明如何在不等待下次重试的情况下强制执行复制更新。

注 – 只有在已经设置复制**并且**已经初始化使用方时，才能使用本部分介绍的过程。

复制重试算法

如果源副本未成功复制到目标，它将定期以递增的时间间隔进行重试。重试时间间隔取决于错误类型。

请注意，即使您已将复制协议配置为始终保持源副本和目标副本同步，但这还不足以立即更新脱机时间已超过五分钟的副本。

▼ 强制执行复制更新

如果复制已停止，您可以对目标后缀强制执行复制更新。

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

- **在源服务器上，重新启动对目标服务器的复制更新。**

  ```
  $ dsconf update-repl-dest-now -h host -p port suffix-DN destination-host:destination-port
  ```

  例如：

  ```
  $ dsconf update-repl-dest-now -h host2 -p 1389 dc=example,dc=com host1:1389
  ```

使用 Directory Server 6.0 之前的版本进行复制

本部分提供有关如何使用 Directory Server 6.0 之前的版本配置复制的信息。

在 Directory Server 6.0 和 Directory Server 5.1（或 5.2）之间进行复制

Directory Server 5.1、5.2 和 6.0 在复制配置方面是互相兼容的，但以下情况除外：

- Directory Server 6.0 之前的版本不支持复制优先级。如果在 6.0 主副本上配置复制优先级，则此复制优先级将被传送到运行 Directory Server 6.0 的使用方，但不会传送到任何运行以前版本的目录服务器的使用方。
- 包含 Directory Server 5.1 或 5.2 主服务器的复制拓扑不允许使用任意数量的主服务器。尽管 Directory Server 6.0 允许在复制拓扑中使用任意数量的主服务器，但如果您的复制拓扑中包含任何 Directory Server 5.2 主服务器，则此数量将被限制为四个。DirectoryServer 5.1 不支持多主复制。

使用追溯更改日志

追溯更改日志由 LDAP 客户端使用，用于确定目录服务器数据的更改历史记录。追溯更改日志与目录服务器更改日志存储在不同的数据库中，位于 `cn=changelog` 后缀下。

可以在单独的服务上或复制拓扑中的每个服务器上启用追溯更改日志。在服务器上启用追溯更改日志时，默认情况下将记录该服务器上所有后缀的更新。可以将追溯更改日志配置为只记录指定后缀的更新。

有关在复制拓扑中使用追溯更改日志以及使用追溯更改日志的限制的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Replication and the Retro Change Log Plug-In”。

有关追溯更改日志中的条目属性的信息，请参见 `changeLogEntry`(5dsoc) 手册页。

有关修改追溯更改日志的详细信息，请参见 `dsconf`(1M) 手册页。

本部分说明使用追溯更改日志的各种方法。

▼ 启用追溯更改日志

要使用追溯更改日志，必须先启用该日志。

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1 **修改追溯更改日志配置条目：**

```
$ dsconf set-server-prop -h host -p port retro-cl-enabled:on
```

2 **重新启动服务器。**

有关信息，请参见第 55 页中的 “启动、停止和重新启动目录服务器实例”。

▼ 将追溯更改日志配置为记录指定后缀的更新

在服务器上启用追溯更改日志时，默认情况下将记录该服务器上所有后缀的更新。此过程介绍如何将追溯更改日志配置为只记录指定后缀的更新。

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1 **修改追溯更改日志配置条目：**

```
$ dsconf set-server-prop -h host -p port retro-cl-suffix-dn:suffix-DN
```

例如，如果只记录 `cn=Contractors,dc=example,dc=com` 后缀和 `ou=People,dc=example,dc=com` 后缀上的更改，请使用以下命令：

```
$ dsconf set-server-prop -h host2 -p 1389 \
 retro-cl-suffix-dn:"cn=Contractors,dc=example,dc=com" \
 retro-cl-suffix-dn:"ou=People,dc=example,dc=com"
```

2 **重新启动服务器。**

有关信息，请参见第 55 页中的 “启动、停止和重新启动目录服务器实例”。

▼ 将追溯更改日志配置为记录已删除条目的属性

此过程介绍如何将追溯更改日志配置为在删除条目时记录该条目的指定属性。

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1 **修改追溯更改日志配置条目：**

```
$ dsconf set-server-prop -h host -p port retro-cl-deleted-entry-attr: \
 attribute1 attribute2
```

例如，要将追溯更改日志设置为记录已删除条目的 UID 属性，请使用以下命令：

```
$ dsconf set-server-prop -h host -p port retro-cl-deleted-entry-attr:uid
```

2 重新启动服务器。

有关信息，请参见第 55 页中的 “启动、停止和重新启动目录服务器实例”。

▼ 修整追溯更改日志

在指定的一段时间过后，可以自动删除追溯更改日志中的条目。要配置一段时间，使条目在此时间段后自动删除，请确保已启用追溯更改日志，然后设置 `cn=Retro Changelog Plugin, cn=plugins, cn=config` 条目中的 `nsslapd-changelogmaxage` 配置属性。

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1 检查是否已启用追溯更改日志。

```
$ dsconf get-server-prop -h host -p port retro-cl-enabled
```

2 如果未启用追溯更改日志，请启用该日志。

```
$ dsconf set-server-prop -h host -p port retro-cl-enabled:on
```

3 设置已记录的更改的最大存留期。

```
$ dsconf set-server-prop -h host -p port retro-cl-max-age: duration
```

其中 *duration* 可以是 `undefined`（无存留期限制）或以下任一选项：

- `s`（秒）
- `m`（分钟）
- `h`（小时）
- `d`（天）
- `w`（周）

例如，要将追溯更改日志的最大存留期设置为两天，请键入：

```
$ dsconf set-server-prop -h host 2 -p 1389 retro-cl-max-age:2d
```

下次在追溯更改日志上执行操作时将会修整该日志。

访问控制和追溯更改日志

追溯更改日志支持搜索操作。它针对特定搜索（包含以下格式的过滤器）进行了优化：

```
(&(changeNumber>=X)(changeNumber<=Y))
```

作为一般规则，请勿对追溯更改日志条目执行添加或修改操作。可以删除条目以修整日志大小。只有在修改默认访问控制策略时，才需要对追溯更改日志执行修改操作。

创建追溯更改日志时，默认情况下将应用以下访问控制规则：

- 为所有经过验证的用户（`userdn=anyone`，不要与 `userdn=all` 的匿名访问相混淆）授予追溯更改日志顶级条目 `cn=changelog` 的读取、搜索和比较权限。
- 不授予写入和删除访问权限，但暗中授予目录管理员的除外。

  不为匿名用户授予读取访问权限，因为追溯更改日志条目可能包含对敏感信息（如密码）的修改。如果不希望经过验证的用户查看追溯更改日志的内容，您可能需要进一步限制对内容的访问权限。

要修改应用于追溯更改日志的默认访问控制策略，请修改 `cn=changelog` 条目的 `aci` 属性。请参见第 6 章。

获取复制状态

可以使用 DSCC 或命令行工具获取复制状态。

在 DSCC 中获取复制状态

可以使用“后缀”选项卡以图形方式查看复制，包括复制协议和复制延迟。有关详细信息，请参见 DSCC 联机帮助。

此外，还可以使用 DSCC 查看复制拓扑，如下图所示。

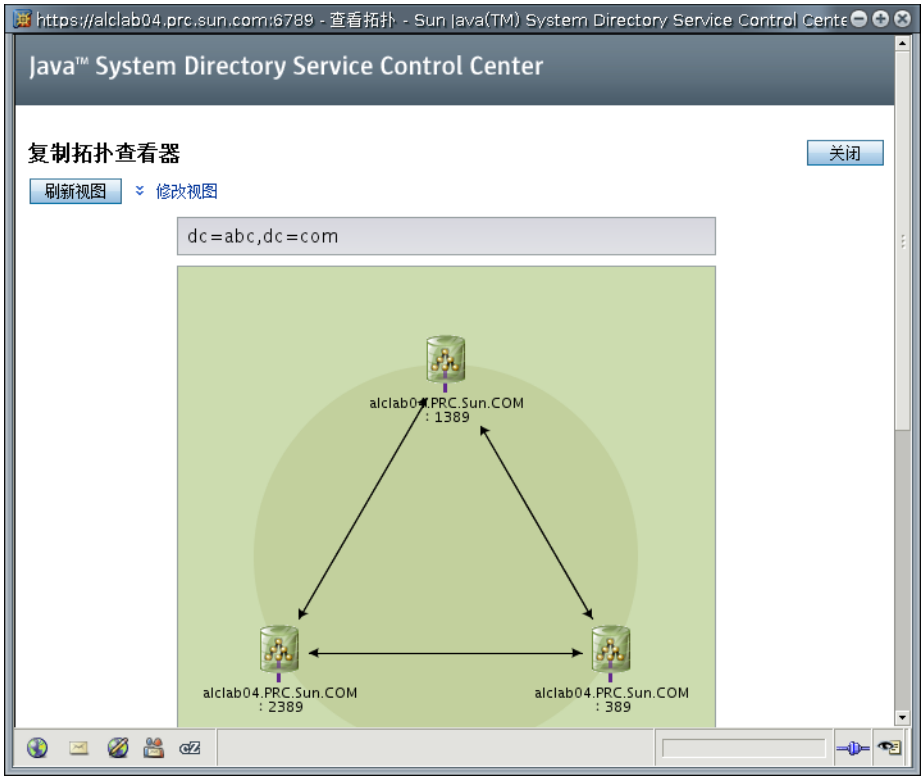


图 10–1 样例复制拓扑

获取复制状态 使用命令行

如果无法使用 DSCC，请使用命令行工具获取有关复制部署的信息。

手册页提供了完整的命令行语法以及这些工具的使用示例。

- `repldisc` - “搜索”和构建包含复制部署中所有已知服务器的表格。请参见 `repldisc`(1) 手册页。
- `insync` - 表明提供方和一个或多个使用方副本之间的同步状态。请参见 `insync`(1) 手册页。
- `entrycmp` - 比较两个或更多副本中的相同条目。请参见 `entrycmp`(1) 手册页。

要查找这些命令所在的目录，请参见第 31 页中的 “命令位置” 。

解决常见复制冲突

多主复制使用宽松的一致性复制模型。这意味着可以在不同服务器上同时修改相同条目。在两个服务器之间发送更新时，必须解决所有冲突的更改。大多数情况下系统都可以自动解决冲突。例如，将使用最近的更改解决与每个服务器上的更改相关联的时间戳。但是，某些更改冲突需要手动介入才能解决。

本部分包括以下主题：

使用 DSCC 解决复制冲突

解决复制冲突的最简单方法是使用 DSCC。有关信息，请参见 DSCC 联机帮助。

使用命令行解决复制冲突

可以使用命令行解决复制冲突。存在更改冲突（无法由复制过程自动解决）的条目将包含操作属性 `nsds5ReplConflict` 作为冲突标记。

要查找存在冲突的条目，请定期搜索包含此属性的条目。例如，您可以使用以下 `ldapsearch` 命令查找存在冲突的条目：

```
$ ldapsearch -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config \
 -w - -b "dc=example,dc=com" "(nsds5ReplConflict=*)"
```

请注意，默认情况下将为 `nsds5ReplConflict` 编制索引。

解决命名冲突

如果在服务器相互复制更改之前创建具有相同 DN 的条目，则可能会在不同的主服务器上创建这些条目。在复制时，冲突解决机制将自动对所创建的第二个条目进行重命名。

存在 DN 命名冲突的条目将通过以下方式进行重命名：在该条目的 DN 中包含其唯一标识符（由操作属性 `nsuniqueid` 提供）。

例如，如果在两个主服务器上同时创建条目 `uid=bjensen,ou=People,dc=example,dc=com`，则复制之后这两个主服务器上都将具有以下两个条目：

- `uid=bjensen,ou=People,dc=example,dc=com`
- `nsuniqueid=66446001-1dd211b2-66225011-2ee211db+uid=bjensen,dc=example,dc=com`

必须为第二个条目提供有用的 DN。您可以删除冲突的条目，然后使用不冲突的名称再次添加该条目。但是，重命名条目将确保其内容不会发生更改。重命名过程取决于命名属性是单值属性还是多值属性。请参见以下过程。

▼ 对包含多值命名属性的冲突条目进行重命名

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 在保留旧 RDN 值的同时重命名条目，例如：

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: nsuniqueid=66446001-1dd211b2-66225011-2ee211db+uid=bjensen,dc=example,dc=com
changetype: modrdn
newrdn: uid=bj66446001
deleteoldrdn: 0
^D
```

您无法在此步骤中删除旧 RDN 值，因为它还包含无法删除的 `nsuniqueid` 操作属性。

2 删除命名属性的旧 RDN 值和冲突标记属性，例如：

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bj66446001,dc=example,dc=com
changetype: modify
delete: uid
uid: bjensen
-
delete: nsds5ReplConflict
^D
```

▼ 使用单值命名属性重命名冲突条目

如果重复条目中的命名属性是单值属性，例如 `dc`（domain component，域组件），则不能简单地将此条目重命名为同一属性的其他值。您必须为此条目提供一个临时名称。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1. **使用其他命名属性对条目进行重命名，并保留旧的 RDN，例如：**

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: nsuniqueid=66446001-1dd211b2-66225011-2ee211db+dc=HR,dc=example,dc=com
changetype: modrdn
newrdn: o=TempHREntry
deleteoldrdn: 0
^D
```

   您无法在此步骤中删除旧 RDN 值，因为它还包含无法删除的 `nsuniqueid` 操作属性。

2. **将所需的命名属性更改为唯一的值，并删除冲突标记属性，例如：**

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: o=TempHREntry,dc=example,dc=com
changetype: modify
replace: dc
dc: NewHR
delete: nsds5ReplConflict
^D
```

3. **将条目重命名为预期的命名属性，例如：**

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: dc=NewHR,dc=example,dc=com
changetype: modrdn
newrdn: dc=HR
deleteoldrdn: 1
^D
```

   通过将 `deleteoldrdn` 属性值设置为 `1`，删除临时属性值对 `o=TempHREntry`。如果要保留此属性，请将 `deleteoldrdn` 属性值设置为 `0`。

解决孤立条目冲突

复制删除操作时，如果使用方服务器发现要删除的条目具有子条目，则冲突解决过程将创建**紧附**条目，以免目录中出现孤立条目。

同样，复制添加操作时，如果使用方服务器找不到父条目，则冲突解决过程将创建代表父条目的紧附条目，以免新条目成为孤立条目。

紧附条目是包含对象类 glue 和 extensibleObject 的临时条目。可以使用多种方法创建紧附条目：

- 如果冲突解决过程发现已删除条目具有匹配的唯一标识符，则紧附条目就是该条目的再生条目。它还包含 glue 对象类和 nsds5ReplConflict 属性。

  在这种情况下，可以修改紧附条目以删除 glue 对象类和 nsds5ReplConflict 属性（以便使此条目保持为普通条目），或者删除紧附条目及其子条目。

- 服务器创建具有 glue 和 extensibleObject 对象类的最小条目。

  在这种情况下，必须修改此条目使其成为有意义的条目，或者删除此条目及其所有子条目。

解决潜在的互操作性问题

如果需要与依赖于属性唯一性的应用程序（如邮件服务器）进行交互操作，您可能需要限制对包含 nsds5ReplConflict 属性的条目的访问权限。如果不限制对这些条目的访问权限，则只需要一个属性的应用程序将同时获得原始条目和包含 nsds5ReplConflict 的冲突解决条目，并且操作将会失败。

要限制访问权限，需要使用以下命令修改授予匿名读取访问权限的默认 ACI。

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: dc=example,dc=com
changetype: modify
delete: aci
aci: (target ="ldap:///dc=example,dc=com")
 (targetattr !="userPassword"
 (version 3.0;acl "Anonymous read-search  access";
 allow (read, search, compare)(userdn = "ldap:///anyone");)
-
add: aci
aci: (target="ldap:///dc=example,dc=com")
 (targetattr!="userPassword")
 (targetfilter="(!(nsds5ReplConflict=*))")(version 3.0;acl
 "Anonymous read-search access";allow (read, search, compare)
 (userdn="ldap:///anyone");)
^D
```

新的 ACI 可阻止在搜索结果中返回包含 nsds5ReplConflict 属性的条目。

◆ ◆ ◆ 第 11 章

11 目录服务器模式

目录服务器提供一个包含大量对象类和属性的标准模式。虽然标准对象类和属性应该能满足您的大多数要求，但您可能仍需通过创建新的对象类和属性来扩展模式。有关标准模式的概述以及设计符合部署的模式的说明，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Deployment Planning Guide》。

本章介绍如何管理模式，其中包含以下主题：

管理模式检查

打开模式检查时，目录服务器可确保所有导入、添加和修改操作都符合当前定义的目录模式。

- 每个条目的对象类和属性都符合模式。
- 条目包含其所有已定义的对象类的所有必需属性。
- 条目只包含其对象类所允许的属性。

注 – 修改条目时，目录服务器将对整个条目（而不仅仅是要修改的属性）执行模式检查。因此，如果条目中的任何对象类或属性不符合模式，操作都可能会失败。

但是，模式检查不会验证属性值在语法方面的有效性。

默认情况下将打开模式检查。通常都在打开模式检查的情况下运行目录服务器。许多客户端应用程序都假定，打开模式检查即表明所有条目都符合模式。但是，打开模式

检查不会使目录服务器验证目录中的现有内容。保证所有目录内容都符合模式的唯一方法是，在添加任何条目或重新初始化所有条目之前打开模式检查。

在某些情况下可能需要关闭模式检查，例如，为了加快已知符合模式的 LDIF 文件的导入速度。但这样做存在风险，因为可能会导入不符合模式的条目。如果模式检查处于关闭状态，则不会检测到不符合模式的导入条目。

有关在复制环境中使用模式检查的详细信息，请参见第 267 页中的 “复制目录模式”。

▼ 解决模式遵循性问题

当某个条目不符合模式时，可能无法搜索此条目，并且对此条目的修改操作可能会失败。请执行以下过程中的步骤来更正此问题。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

开始之前 为了避免日后需要解决模式遵循性问题，请在部署之前规划模式，以尽可能减少模式更改。有关详细信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Deployment Planning Guide》。

1. **要确定条目不符合模式的原因，请检索此条目，然后手动将其与当前定义的模式进行比较。**

   有关详细信息，请参见第 259 页中的 “查看属性类型”和第 262 页中的 “查看对象类”。

2. **修改条目以使其符合模式，或修改模式以使其符合条目。**

关于自定义模式

如果标准模式无法满足您的目录需求，则可以扩展标准模式。自定义模式时应遵循以下准则：

- 尽可能重新使用现有的模式元素。
- 尽可能减少为每个对象类定义的必需属性的数量。
- 不要针对同一用途定义多个对象类或属性。
- 使模式尽可能简单。

自定义模式时，不要修改、删除或替换标准模式中属性或对象类的任何现有定义。否则，可能会导致与其他目录和 LDAP 客户端应用程序的兼容性问题。

不要修改任何目录服务器内部操作属性。但是，您可以创建自己的操作变量以用于外部应用程序。

应始终定义对象类，而不要使用 objectClass: extensibleObject。目录服务器不会对具有对象类 extensibleObject 的条目执行模式检查，因此不会限制或检查该条目中存在的属性。目录服务器无法检测到应用程序中的拼写错误，例如，将 givenName 属性类型写成 giveName。此外，目录服务器还必须假定 extensibleObject 条目所有其他未定义的属性都是多值属性，并且使用区分大小写的字符串语法。而且，某些应用程序依赖于具有特定对象类的条目。通常，如果您的应用程序要求扩展对象类，请不要放弃模式管理，而应创建一个包含该应用程序所需属性的辅助对象类。

本部分包含有关默认目录模式以及创建自定义属性和对象类的信息。

默认目录服务器模式

instance-path/config/schema/ 目录中存储的一组文件对目录服务器随附的模式进行了介绍。

此目录包含目录服务器和相关产品的所有通用模式。LDAP v3 标准用户和组织模式位于 00core.ldif 文件中。此目录的早期版本所使用的配置模式位于 50ns-directory.ldif 文件中。

注 – 当服务器正在运行时，不要修改此目录中的文件。

对象标识符

必须为每个 LDAP 对象类或属性指定唯一的名称和对象标识符 (object identifier, OID)。定义模式时，您需要一个在组织中具有唯一性的 OID。一个 OID 即可满足您的所有模式需求。然后，您可以在该 OID 上为您的属性和对象类添加新的分支。

获取和指定模式中的 OID 时，您需要执行以下操作：

- 从互联网号码分配机构 (Internet Assigned Numbers Authority, IANA) 或国家组织为您的组织获取 OID。

  在某些国家/地区，已经为公司指定了 OID。如果您的组织还没有 OID，则可以从 IANA 获取 OID。

- 创建一个 OID 注册表，以便可以跟踪 OID 分配。

  OID 注册表是由您维护的列表，它提供目录模式中所使用的 OID 及 OID 描述。OLD 注册表可确保任何 OID 都不会用于多种用途。

- 在 OID 树中创建分支以容纳模式元素。

  在 OID 分支或目录模式下至少创建两个分支，*OID*.1 用于属性，*OID*.2 用于对象类。如果您要定义自己的匹配规则或控制，则可以根据需要添加新的分支（如 *OID*.3）。

命名属性和对象类

创建新属性和对象类的名称时，应使用有意义的名称，以使模式更易于使用。

通过在自定义元素上包含唯一的前缀，可以避免在自定义模式元素和现有模式元素之间出现命名冲突。例如，Example.com 公司可以在其每个自定义模式元素之前添加前缀 `Example`。它还可以添加一个名为 `ExamplePerson` 的特殊对象类，以便在目录中标识 Example.com 员工。

请注意，在 LDAP 中，属性类型名称和对象类名称**区分大小写**。应用程序应将其视为区分大小写的字符串。

定义新对象类

如果现有对象类不支持需要在目录条目中存储的所有信息，则可以添加新的对象类。

可以使用两种方法创建新对象类：

- 创建许多新对象类，分别用于要添加属性的每个对象类结构。
- 创建一个对象类，使其支持您为目录创建的所有属性。创建这种对象类的方法是将其定义为 AUXILIARY 对象类。

  假定您的站点要创建属性 ExampleDepartmentNumber 和 ExampleEmergencyPhoneNumber。您可以创建多个允许这些属性中的部分属性的对象类。可以创建一个名为 ExamplePerson 的对象类，并使其允许 ExampleDepartmentNumber 和 ExampleEmergencyPhoneNumber 属性。ExamplePerson 的父条目将是 inetOrgPerson。然后可以创建一个名为 ExampleOrganization 的对象类，并使其允许 ExampleDepartmentNumber 和 ExampleEmergencyPhoneNumber 属性。ExampleOrganization 的父条目将是 organization 对象类。

  新对象类将以 LDAP v3 模式格式显示，如下所示：

```
objectclasses: (1.3.6.1.4.1.42.2.27.999.1.2.3 NAME 'ExamplePerson'
 DESC 'Example Person Object Class' SUP inetorgPerson STRUCTURAL MAY
 (ExampleDepartmentNumber $ ExampleEmergencyPhoneNumber) )
objectclasses: (1.3.6.1.4.1.42.2.27.999.1.2.4 NAME
 'ExampleOrganization' DESC 'Example Organization Object Class' SUP
 organization STRUCTURAL MAY (ExampleDepartmentNumber
 $ ExampleEmergencyPhoneNumber) )
```

  或者，您还可以创建一个允许所有这些属性的对象类。然后，可以将此对象类用于要使用这些属性的任何条目。单个对象类将如下所示：

```
objectclasses: (1.3.6.1.4.1.42.2.27.999.1.2.5 NAME 'ExampleEntry'
 DESC  'Example Auxiliary Object Class' SUP top  AUXILIARY MAY
 (ExampleDepartmentNumber $ ExampleEmergencyPhoneNumber) )
```

  新的 ExampleEntry 对象类被标记为 AUXILIARY，这意味着它可以用于任何条目，无论该条目的结构对象类如何。

确定如何实现新对象类时，请考虑以下事项。

- 多个 STRUCTURAL 对象类将导致需要创建和维护更多的模式元素。

  通常，元素的数量始终很少，并且几乎不需要维护。但是，如果您计划向模式中添加两个或三个以上的对象类，则使用单个对象类可能会更容易一些。
- 多个 STRUCTURAL 对象类要求在设计数据时更加仔细和严格。

  严格的数据设计将迫使您考虑放置每份数据的对象类结构。此限制可能很有用，但也可能带来麻烦。

- 如果要将数据放在多种类型的对象类结构上，则单个 AUXILIARY 对象类可简化数据设计。

  例如，假定您要将 preferredOS 同时放在个人条目和组条目上。您可能只想创建一个对象类以允许此属性。
- 设计与实际对象和组元素相关、且可构成合理分类的对象类。
- 避免为新对象类设置必需属性。

  必需属性可能会使模式缺乏灵活性。创建新对象类时，请设置允许属性，而不要设置必需属性。

  定义新对象类之后，您需要确定该对象类允许和需要哪些属性，以及该对象类是从哪个或哪些对象类继承而来的。

定义新属性

如果现有属性不支持需要在目录条目中存储的所有信息，则可以添加新的属性。请尽可能使用标准属性。请搜索默认目录模式中已存在的属性，并使用与新对象类关联的那些属性。

例如，除了 person、organizationalPerson 或 inetOrgPerson 对象类所支持的信息之外，您可能还想在个人条目上存储更多信息。如果要在目录中存储生日，标准目录服务器模式内却不存在任何属性。您可以创建一个名为 dateOfBirth 的新属性。通过定义允许此属性的新辅助类，可以将此属性用于表示人员的条目上。

创建自定义模式文件

创建自定义模式文件时（特别是使用复制时），请注意以下事项：

- 添加新的模式元素时，所有属性都必须先进行定义，然后才能用于对象类。可以在同一模式文件中定义属性和对象类。
- 您所创建的每个自定义属性或对象类都只应在一个模式文件中进行定义。这样做可防止服务器在装入最新创建的模式时覆盖任何以前的定义。目录服务器首先按数字顺序装入模式文件，然后再按字母顺序装入。
- 手动定义新的模式定义时，最佳做法通常是将这些定义添加到 99user.ldif 文件中。

  使用 LDAP 更新模式元素时，新元素将自动写入 99user.ldif 文件。因此，您在自定义模式文件中所做的任何其他模式定义更改都可能被覆盖。只使用 99user.ldif 文件可防止出现重复的模式元素，并可避免覆盖模式更改的危险。
- 由于目录服务器按字母数字顺序（先装入数字）装入模式文件，因此应按以下方式命名自定义模式文件：

  [00-99] *filename*.ldif

数字应高于已定义的任何目录标准模式。

如果使用低于标准模式文件的数字命名模式文件，则服务器在装入模式时可能会发生错误。此外，只有在装入自定义模式元素之后，才会装入所有标准属性和对象类。

- 请确保自定义模式文件名称在数字和字母顺序上都不高于 `99user.ldif`，因为目录服务器将使用具有最高顺序的文件进行内部模式管理。

  例如，如果您创建一个名为 `99zzz.ldif` 的模式文件，则在下次更新模式时，所有 `X-ORIGIN` 值为 `'user defined'` 的属性都会写入 `99zzz.ldif`。这样会出现两个包含重复信息的 LDIF 文件，并且 `99zzz.ldif` 文件中的某些信息可能会被删除。
- 作为一般规则，应使用以下两项内容标识要添加的自定义模式元素：
  - 自定义模式文件 `X-ORIGIN` 字段中的 `'user defined'`，
  - 描述性较强的标签（如 `X-ORIGIN` 字段中的 `'Example.com Corporation defined'`），以便其他管理员可以更容易理解自定义模式元素。例如，`X-ORIGIN ('user defined' 'Example.com Corporation defined')`。

    如果要手动添加模式元素，且不使用 `X-ORIGIN` 字段中的 `'user defined'`，则模式元素在 DSCC 中显示为只读状态。

    如果使用 LDAP 或 DSCC 添加自定义模式定义，服务器将自动添加 `'user defined'` 值。但是，如果不在 `X-ORIGIN` 字段中添加描述性较强的值，则日后可能会难以理解此模式的相关用途。

应手动将所有自定义模式文件传播到所有服务器中，因为这些更改不会自动复制。

更改目录模式时，服务器将保留模式更改的时间戳。在每个复制会话开始时，服务器会将其时间戳与使用方的时间戳进行比较，然后在必要的情况下发送所有模式更改。对于自定义模式文件，服务器只保留一个与 `99user.ldif` 文件关联的时间戳。这意味着您对 `99user.ldif` 以外的文件所做的任何自定义模式文件更改或添加都不会进行复制。因此，必须将自定义模式文件传播到所有其他服务器，以确保所有模式信息存在于整个拓扑中。

通过 LDAP 管理属性类型

本部分介绍如何通过 LDAP 创建、查看和删除属性类型。

创建属性类型

`cn=schema` 条目具有多值属性 `attributeTypes`，该属性包含目录模式中每个属性类型的定义。可以使用 `ldapmodify(1)` 命令添加这些定义。

新的属性类型定义以及您对用户定义的属性类型所做的更改都保存在 `99user.ldif` 文件中。

对于每个属性类型定义，至少要提供一个 OID 以定义新的属性类型。请考虑至少为新属性类型使用以下元素：

- **属性 OID。** 与属性的对象标识符相对应。OID 是一个字符串，通常是点分十进制数字，可以唯一地标识模式对象。

  为了严格遵循 LDAP v3，您必须提供有效的数字 OID。要了解 OID 的详细信息，或者为您的企业请求前缀，请向互联网号码分配机构 (Internet Assigned Number Authority, IANA) 发送电子邮件（地址为 `iana@iana.org`），或访问 IANA Web 站点 (`http://www.iana.org`)。
- **属性名称。** 与属性的唯一名称相对应。也称为其属性类型。属性名称必须以字母开头，并且只能包含 ASCII 字母、数字和连字符。

  属性名称可以包含大写字母，但任何 LDAP 客户端都不会根据大小写来区分属性。根据 RFC 4512 (`http://www.ietf.org/rfc/rfc4512.txt`) 的 2.5 节，必须以不区分大小写的方式处理属性名称。

  您可以选择为属性类型添加备用属性名称（也称为别名）。
- **属性描述。** 用于介绍属性用途的简短描述性文本。
- **语法。** 由 OID 引用，用于描述属性将包含的数据。

  RFC 4517 (`http://www.ietf.org/rfc/rfc4517.txt`) 中列出了属性语法及其 OID。
- **允许的值数。** 默认情况下属性可以具有多个值，但您可能希望将属性限制为单值属性。

▼ 创建属性类型

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1. **根据 RFC 4517 (`http://www.ietf.org/rfc/rfc4517.txt`) 中指定的语法准备属性类型定义。**
2. **使用 `ldapmodify`(1) 命令添加属性类型定义。**

   请注意，目录服务器会将 `X-ORIGIN 'user defined'` 添加到您提供的定义中。

示例 11–1 创建属性类型

以下示例将使用 `ldapmodify` 命令添加具有目录字符串语法的新属性类型：

```
$ cat blogURL.ldif
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: ( 1.2.3.4.5.6.7
 NAME ( 'blog' 'blogURL' )
```

```
 DESC 'URL to a personal weblog'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE )

$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w - -f blogURL.ldif
Enter bind password:
modifying entry cn=schema

$
```

在生产环境中，应该提供一个唯一的有效 OID，而不是 `1.2.3.4.5.6.7`。

查看属性类型

`cn=schema` 条目具有多值属性 `attributeTypes`，该属性包含目录模式中每个属性类型的定义。可以使用 `ldapsearch`(1) 命令读取这些定义。

▼ 查看属性类型

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

- **使用 `ldapsearch` 命令查看您的目录模式中当前存在的所有属性类型定义。**

示例 11–2 查看属性类型

以下命令显示所有属性类型的定义：

```
$ ldapsearch -T -b cn=schema "(objectclass=*)" attributeTypes
```

`-T` 选项可阻止 `ldapsearch` 命令折叠 LDIF 行，以便您更容易地使用 `grep` 或 `sed` 等命令处理输出。这样，如果您通过 `grep` 命令输送此命令的输出，则可以只查看目录模式中用户定义的扩展部分。例如：

```
$ ldapsearch -T -b cn=schema "(objectclass=*)" attributeTypes | grep "user defined"
 attributeTypes: ( 1.2.3.4.5.6.7 NAME ( 'blog' 'blogURL' )
 DESC 'URL to a personal weblog'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE
 X-ORIGIN 'user defined' )
```

删除属性类型

`cn=schema` 条目具有多值属性 `attributeTypes`，该属性包含目录模式中每个属性类型的定义。可以使用 `ldapmodify`(1) 命令删除具有 `X-ORIGIN 'user defined'` 的定义。

由于模式是由 LDAP 视图在 cn=schema 中定义的，因此您可以使用 ldapsearch 和 ldapmodify 实用程序联机查看和修改模式。但是，您只能删除 X-ORIGIN 字段值为 'user defined' 的模式元素。服务器不会删除其他定义。

对用户定义属性所做的更改将保存在 99user.ldif 文件中。

▼ 删除属性类型

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1. **查看要删除的属性类型的定义。**

   有关详细信息，请参见第 259 页中的 “查看属性类型” 。

2. **使用 ldapmodify(1) 命令删除模式中出现的属性类型定义。**

示例 11–3 删除属性类型

以下命令将删除在示例 11–1 中创建的属性类型：

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: cn=schema
changetype: delete
delete: attributeTypes
attributeTypes: ( 1.2.3.4.5.6.7 NAME ( 'blog' 'blogURL' )
 DESC 'URL to a personal weblog'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE
 X-ORIGIN 'user defined' )
^D
```

请注意，必须包括由目录服务器添加的 X-ORIGIN 'user defined'（用于将此模式定义归类为扩展）。

通过 LDAP 管理对象类

本部分介绍如何通过 LDAP 创建、查看和删除对象类。

创建对象类

cn=schema 条目具有多值属性 objectClasses，该属性包含目录模式中每个对象类的定义。可以使用 ldapmodify(1) 命令添加这些定义。

新的对象类定义以及您对用户定义的对象类所做的更改都保存在 99user.ldif 文件中。

如果您要创建继承其他对象类的多个对象类，则必须先创建父对象类。如果新对象类使用自定义属性，还必须首先定义这些属性。

至少要为每个对象类定义提供一个 OID。请考虑至少为新对象类使用以下元素：

- **对象类 OID**。与对象类的对象标识符相对应。OID 是一个字符串，通常是点分十进制数字，可以唯一地标识模式对象。

  为了严格遵循 LDAP v3，您必须提供有效的数字 OID。要了解 OID 的详细信息，或者为您的企业请求前缀，请向互联网号码分配机构 (Internet Assigned Number Authority, IANA) 发送电子邮件（地址为 `iana@iana.org`），或访问 IANA Web 站点 (`http://www.iana.org`)。
- **对象类名称**。与对象类的唯一名称相对应。
- **父对象类**。此对象类从中继承属性的现有对象类。

  如果您不希望此对象类继承其他特定对象类，请使用 `top`。

  通常，如果要为用户条目添加新属性，则父对象类将是 `inetOrgPerson` 对象类。如果要为公司条目添加新属性，则父对象类通常是 `organization` 或 `organizationalUnit`。如果要为组条目添加新属性，则父对象类通常是 `groupOfNames` 或 `groupOfUniqueNames`。
- **必需属性**。列出并定义此对象类**必须**具有的属性。
- **允许的属性**。列出并定义此对象类可以具有的其他属性。

▼ 创建对象类

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1. **根据 RFC 4517 (`http://www.ietf.org/rfc/rfc4517.txt`) 中指定的语法准备对象类定义。**
2. **使用 `ldapmodify`(1) 命令添加对象类定义。**

   请注意，目录服务器会将 `X-ORIGIN 'user defined'` 添加到您提供的定义中。

示例 11–4 创建对象类

以下示例使用 `ldapmodify` 命令添加新对象类：

```
$ cat blogger.ldif
dn: cn=schema
changetype: modify
add: objectClasses
objectClasses: ( 1.2.3.4.5.6.8
 NAME 'blogger'
 DESC 'Someone who has a blog'
 SUP inetOrgPerson
```

```
 STRUCTURAL
 MAY blog )

$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w - -f blogger.ldif
Enter bind password:
modifying entry cn=schema

$
```

在生产环境中，应该提供唯一的有效 OID，而不是 1.2.3.4.5.6.8。

查看对象类

cn=schema 条目具有多值属性 objectClasses，该属性包含目录模式中每个对象类的定义。可以使用 ldapsearch(1) 命令读取这些定义。

▼ 查看对象类

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

- **使用 ldapsearch 命令查看您的目录模式中当前存在的所有对象类定义。**

示例 11–5 查看对象类

以下命令显示所有对象类的定义：

```
$ ldapsearch -T -b cn=schema "(objectclass=*)" objectClasses
```

-T 选项可阻止 ldapsearch 命令折叠 LDIF 行，以便您更容易地使用 grep 或 sed 等命令处理输出。这样，如果您通过 grep 命令输送此命令的输出，则可以只查看目录模式中用户定义的扩展部分。例如：

```
$ ldapsearch -T -b cn=schema "(objectclass=*)" objectClasses | grep "user defined"
objectClasses: ( 1.2.3.4.5.6.8 NAME 'blogger'
 DESC 'Someone who has a blog' STRUCTURAL MAY blog
 X-ORIGIN 'user defined' )
$
```

删除对象类

cn=schema 条目具有多值属性 objectClasses，该属性包含目录模式中每个对象类的定义。可以使用 ldapmodify(1) 命令删除具有 X-ORIGIN 'user defined' 的定义。

由于模式是由 LDAP 视图在 cn=schema 中定义的，因此您可以使用 ldapsearch 和 ldapmodify 实用程序联机查看和修改模式。但是，您只能删除 X-ORIGIN 字段值为 'user defined' 的模式元素。服务器不会删除其他定义。

对用户定义元素所做的更改将保存在 99user.ldif 文件中。

▼ 删除对象类

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1. **查看要删除的对象类的定义。**
   有关详细信息，请参见第 262 页中的 “查看对象类”。
2. **使用 ldapmodify(1) 命令删除模式中出现的对象类定义。**

示例 11–6 删除对象类

以下命令将删除在示例 11–4中创建的对象类：

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: cn=schema
changetype: delete
delete: objectClasses
objectClasses: ( 1.2.3.4.5.6.8 NAME 'blogger' DESC 'Someone who has a blog'
 STRUCTURAL MAY blog X-ORIGIN 'user defined' )
^D
```

请注意，必须包括由目录服务器添加的 X-ORIGIN 'user defined'（用于将此模式定义归类为扩展）。

扩展目录服务器模式

向模式中添加新属性时，必须创建新对象类以包含这些新属性。可以直接将属性添加到已包含您所需要的大多数属性的现有对象类中，这种方法看起来比较方便，但却会影响与 LDAP 客户端之间的互操作性。

目录服务器与现有 LDAP 客户端之间的互操作性依赖于标准 LDAP 模式。如果更改标准模式，您在升级服务器时也会遇到困难。同理，您也不能删除标准模式元素。

目录服务器模式存储在 cn=schema 条目的属性中。与配置条目一样，此条目也是模式的 LDAP 视图，在服务器启动期间将从文件中读取此视图。

用于扩展目录服务器模式的方法取决于您是否要控制存储模式扩展时所使用的文件名。此外，还取决于您是否要通过复制将更改发送给使用方。请参见下表，以便根据您的具体情况确定要执行的过程。

表 11–1 扩展模式的方法

任务	说明
您不使用复制。您打算通过添加自定义模式文件扩展模式。	第 265 页中的 “使用自定义模式文件扩展模式”
您打算通过 LDAP 扩展模式。	第 265 页中的 “通过 LDAP 扩展模式”
您使用复制。您打算在所有服务器上保留自定义模式文件的文件名。	第 265 页中的 “使用自定义模式文件扩展模式”
您使用复制。您打算通过在主副本上添加自定义模式文件来扩展模式。然后通过复制机制将模式扩展复制到使用方服务器。	第 266 页中的 “使用模式文件和复制扩展模式”

有关对象类、属性和目录模式的详细信息，以及扩展模式所应遵循的准则，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Deployment Planning Guide》中的“Designing a Directory Schema”。有关标准属性和对象类的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Man Page Reference》 。

本部分提供扩展目录模式时可使用的各种方法的相关信息。

使用自定义模式文件扩展模式

模式文件是位于 *instance-path*`/config/schema/` 中的 LDIF 文件。*instance-path* 与目录服务器实例所在的文件系统目录相对应。例如，此实例可能位于 `/local/ds/` 中。这些文件可定义供目录服务器以及依赖于目录服务器的所有服务器使用的标准模式。《Sun Java System Directory Server Enterprise Edition 6.0 Reference》和《Sun Java System Directory Server Enterprise Edition 6.0 Man Page Reference》中介绍了这些文件和标准模式。

服务器只在启动时读取模式文件一次。这些 LDIF 文件的内容将被添加到 `cn=schema` 中模式的内存 LDIF 视图。由于模式定义的顺序非常重要，因此模式文件名称都以数字开头，并按字母数字顺序装入。只有在安装期间定义的系统用户才能对此目录中的模式文件执行写入操作。

在 LDIF 文件中直接定义模式时，不要在 `X-ORIGIN` 字段中使用 `'user defined'` 值。此值是为特定的模式元素保留的，这些元素通过 `cn=schema` 的 LDAP 视图进行定义，并出现在 `99user.ldif` 文件中。

`99user.ldif` 文件包含其他 ACI，用于 `cn=schema` 条目和所有已从命令行或使用 DSCC 添加的模式定义。添加新的模式定义时，将覆盖 `99user.ldif` 文件。如果要修改此文件，则必须立即重新启动服务器，以确保更改是最新的。

不要修改在其他模式文件中定义的标准模式。但是，您可以添加新文件，以定义新的属性和对象类。例如，要在许多服务器中定义新的模式元素，则可以在名为 98mySchema.ldif 的文件中定义这些元素，并将此文件复制到所有服务器上的模式目录中。然后，应重新启动所有服务器以装入新的模式文件。

▼ 使用自定义模式文件扩展模式

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1. **创建您自己的模式定义文件，如 98mySchema.ldif。**

   RFC 4517 (http://www.ietf.org/rfc/rfc4517.txt) 中介绍了模式文件中的定义所使用的语法。

2. **（可选的）如果此服务器是将更新发送到其他服务器的主副本，请将您的模式定义文件复制到复制拓扑中的每个服务器实例中。**

   复制机制无法检测到您直接对包含模式的 LDIF 文件所做的任何更改。因此，即使在重新启动主服务器之后，您的更改也不会复制到使用方。

3. **重新启动将模式定义文件复制到的每个目录服务器实例。**

   当服务器重新启动并重新装入模式定义时，您的更改将会生效。

通过 LDAP 扩展模式

由于模式是由 LDAP 视图在 cn=schema 中定义的，因此您可以使用 ldapsearch 和 ldapmodify 实用程序联机查看和修改模式。但是，您只能修改 X-ORIGIN 字段值为 'user defined' 的模式元素。服务器将拒绝对其他定义所做的任何修改。

新的元素定义以及您对用户定义元素所做的更改都保存在 99user.ldif 文件中。

▼ 通过 LDAP 扩展模式

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

开始之前 从命令行修改模式定义容易出错，因为必须准确键入较长的值。但是，您可以在需要更新目录模式的脚本中使用此功能。

1. **使用 ldapmodify(1) 命令添加或删除单个的 attributeTypes 属性值。**

   有关详细信息，请参见第 258 页中的 “创建属性类型”或第 260 页中的 “删除属性类型”。

2 **使用 `ldapmodify`(1) 命令添加或删除单个的 `objectClasses` 属性值。**

有关详细信息，请参见第 261 页中的 “创建对象类”或第 263 页中的 “删除对象类”。

另请参见 要修改其中某个值，您必须先删除特定的值，然后将此值作为新值进行添加。由于属性为多值属性，因此必须执行此过程。有关详细信息，请参见第 87 页中的 “修改多值属性的一个值”。

使用模式文件和复制扩展模式

有关自定义模式文件的信息，请参见第 264 页中的 “使用自定义模式文件扩展模式”。以下过程介绍如何使用复制机制将模式扩展传播到拓扑中的所有服务器。

▼ 使用模式文件和复制扩展模式

对于此过程的某些部分，可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。此过程的其他部分只能使用命令行完成。

1 **使用以下任一方法准备您的模式扩展：**

- **创建您自己的模式定义文件，如 `98mySchema.ldif`。**
- **将您的模式扩展添加到 `99user.ldif`。**

RFC 4517 (`http://www.ietf.org/rfc/rfc4517.txt`) 中介绍了模式文件中的定义所使用的语法。

2 **在放置模式定义文件的主服务器上运行 `schema_push` 命令。**

此脚本不会实际将模式发送到副本，而是将特殊属性写入模式文件，以使这些模式文件在装入后立即被复制。有关详细信息，请参见 `schema_push`(1M) 手册页。

3 **重新启动放置模式定义文件的主服务器。**

复制机制无法检测到您直接对包含模式的 LDIF 文件所做的任何更改。但是，在运行 `schema_push` 后重新启动服务器时，该服务器将装入所有模式文件，然后复制机制会将新的模式复制到使用方。

复制目录模式

在两个服务器之间配置一个或多个后缀的复制时，也会自动复制模式定义。模式定义的自动复制可确保所有副本都具有完整、相同的模式，此模式用于定义可以复制到使用方的所有对象类和属性。因此，主服务器也将包含主服务器模式。

但模式复制不是即时完成的，即使通过 LDAP 修改模式时也是如此。模式复制会在更新目录数据时触发，或者在修改模式后启动第一个复制会话时触发。

要在所有副本上执行模式，至少要在所有服务器上启用模式检查。由于在执行 LDAP 操作的主服务器上执行模式检查，因此更新使用方时不需要检查模式。为了提高性能，复制机制将避开使用方副本上的模式检查。

注 – 不要在集线器和专用使用方上关闭模式检查。模式检查不会影响使用方的性能。应始终打开模式检查，以表明副本内容符合其模式。

在使用方初始化期间，主服务器自动将模式复制到其使用方。只要通过 DSCC 或命令行工具修改模式，主服务器还会自动复制此模式。默认情况下将复制整个模式。将在使用方上创建其尚不存在的所有其他模式元素，并将这些元素存储在 `99user.ldif` 文件中。

例如，假定在启动主服务器时，该服务器包含 `98mySchema.ldif` 文件中的模式定义。此外，还假定您接下来将定义与其他服务器（主服务器、集线器或专用使用方）之间的复制协议。当您随后从此主服务器初始化副本时，复制的模式将包含 `98mySchema.ldif` 中的定义，但这些定义将存储在副本服务器上的 `99user.ldif` 中。

在使用方初始化期间复制模式之后，修改主服务器 `cn=schema` 中的模式还会将整个模式复制到使用方。因此，通过命令行实用程序或 DSCC 对主服务器模式所做的任何修改都将复制到使用方。这些修改存储在主服务器上的 `99user.ldif` 中，并且通过上述机制，这些修改还会存储在使用方上的 `99user.ldif` 中。

请考虑以下有关在复制环境中维护模式一致性的准则：

- 不要修改使用方服务器上的模式。

  对使用方服务器上的模式所做的修改可能会导致复制错误。这是因为使用方上的模式差异可能会导致来自提供方的更新不符合使用方上的模式。

- 在多主复制环境中，请修改单个主服务器上的模式。

  修改两个主服务器上的模式时，最近更新的主服务器会将其模式版本传播到使用方。这样，使用方上的模式可能与其他主服务器上的模式不一致。

配置**部分复制**时，还应考虑以下准则：

- 由于在部分复制配置中由提供方发送模式，因此部分使用方副本上的模式是主副本模式的副本。因此，模式与所应用的部分复制配置可能不相符。
- 通常，目录服务器会按照模式中的定义复制每个条目的所有必需属性，以免发生模式违规错误。将部分复制配置为过滤掉必需属性时，必须禁用模式检查。
- 如果对部分复制启用模式检查，可能无法以脱机方式初始化副本。如果过滤掉必需属性，目录服务器将不允许您从 LDIF 装入数据。
- 如果在部分使用方副本上禁用了模式检查，则部分使用方副本所在的整个服务器实例都不会执行模式检查。因此，应避免将同一服务器实例上的提供方副本配置为部分使用方。

限制模式复制

默认情况下，只要复制机制复制模式，它都会将整个模式发送到其使用方。在以下两种情况下不希望将整个模式发送到使用方：

- 使用 DSCC 或从命令行对 `cn=schema` 所做的修改仅限于用户定义的模式元素，所有标准模式都不会更改。如果您经常修改模式，则每次发送大量未更改的模式元素时都会对性能造成影响。您可以只复制用户定义的模式元素，以提高复制和服务器性能。
- 当目录服务器上的主服务器复制到 Directory Server 5.1 上的使用方时，这些版本的配置属性模式将有所不同，因而会产生冲突。在这种情况下，您**只能**复制用户定义的模式元素。

注 – 目录服务器使用 `11rfc2307.ldif` 模式文件。此模式文件符合 RFC 2307 (`http://www.ietf.org/rfc/rfc2307.txt`)。

DirectoryServer 5.2 以前的目录服务器版本使用 `10rfc2307.ldif` 模式文件。

▼ 限制模式复制

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

- **将模式复制限制为只复制用户定义的模式。**

```
$ dsconf set-server-prop -h host -p port repl-user-schema-enabled:on
```

默认值 `off` 会在必要时复制整个模式。

◆ ◆ ◆ 第 12 章

12 目录服务器索引

与书籍索引类似，目录服务器索引通过将搜索字符串与目录内容引用相关联，可以加快搜索速度。

有关索引类型和索引调整的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的第 6 章 “Directory Server Indexing”。

本章包含以下主题：

管理索引

本部分介绍如何管理特定属性的索引。本部分包含有关创建、修改和删除索引的信息。有关特定于虚拟列表视图 (Virtual List View, VLV) 操作的过程，请参见第 275 页中的 “管理浏览索引” 。

▼ 列出索引

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

- **要列出现有索引及其属性，请使用以下命令：**

```
$ dsconf list-indexes -h host -p port -v suffix-DN
```

▼ 创建索引

注 – 您无法创建新的系统索引，而只能维护由目录服务器内部定义的现有系统索引。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1. **创建新的索引配置。**

   可以使用 `dsconf create-index` 命令行实用程序，通过指定要编制索引的属性来配置新的索引信息。

   例如，要创建 `preferredLanguage` 属性的索引条目，请使用以下命令：

   ```
   $ dsconf create-index -h host -p port dc=example,dc=com preferredLanguage
   ```

   注 – 命令 `dsconf create-index` 将设置索引配置，但不会实际创建搜索所需的索引文件。生成索引文件可能会影响性能。要对索引编制过程进行更多的控制，请在创建新的索引配置之后手动生成索引文件。

   创建索引时应始终使用属性的主名，而不要使用属性的别名。属性的主名是在模式中列出的第一个属性名称，例如，`userid` 属性的主名为 `uid`。

2. **（可选的）使用 `dsconf set-index-prop` 命令设置索引属性。**

   `dsconf create-index` 命令将使用默认属性创建索引。如果要修改这些属性，请使用 `dsconf set-index-prop` 命令。有关修改索引属性的详细信息，请参见第 270 页中的 “修改索引”。

3. **生成索引文件。**

   请参见第 271 页中的 “生成索引”。

4. **对要编制索引的所有服务器重复上述步骤。**

▼ 修改索引

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1. **修改索引属性。**

   ```
   $ dsconf set-index-prop -h host -p port suffix-DN attr-name property:value
   ```

例如，要为 preferredLanguage 索引启用近似索引 approx-enabled，请使用以下命令：

```
$ dsconf set-index-prop -h host -p port dc=example,dc=com preferredLanguage approx-enabled:on
```

可以修改每个索引的以下属性：

- eq-enabled 等同
- pres-enabled 存在
- sub-enabled 子串

您要修改的属性可能包括可选的 nsMatchingRule 属性。此属性包含服务器已知的所有匹配规则的 OID。它将启用国际化索引的语言对照顺序的 OID，以及 CaseExactMatch 等其他匹配规则。有关受支持的语言环境及其关联对照顺序的 OID 的列表，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》。

有关索引配置属性的详细信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》。

2. **重新生成新的索引。**

   请参见第 271 页中的 “生成索引”。

3. **对包含已修改的属性索引的所有服务器重复上述步骤。**

▼ 生成索引

此过程将生成索引文件，以便可以搜索新索引或已修改的索引。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **使用以下任一方法生成索引文件：**
  - 联机生成新的索引文件。

    ```
    $ dsconf reindex -h host -p port [-t attr] suffix-DN
    ```

    其中 -t 指定仅为指定的一个或多个属性（而非所有属性）重新编制索引。

    例如，要重新生成 preferredLanguage 索引，请键入：

    ```
    $ dsconf reindex -h host -p port -t preferredLanguage dc=example,dc=com
    ```

    dsconf reindex 命令运行期间，仍然可以通过服务器访问后缀的内容。但是在命令完成之前，不会为搜索编制索引。重新编制索引是一项资源密集型任务，它可能会影响服务器上其他操作的性能。
  - 脱机生成新的索引文件。

```
$ dsadm reindex -t attr instance-path suffix-DN
```

例如，要重新生成 preferredLanguage 索引，请键入：

```
$ dsadm reindex -t preferredLanguage /local/ds dc=example,dc=com
```

- 通过重新初始化后缀在脱机状态下快速重新生成全部索引。

  重新初始化后缀时，将自动重新生成全部索引文件。重新初始化后缀通常比为两个或更多属性重新编制索引更快（取决于目录的大小）。但是后缀在初始化期间不可用。有关详细信息，请参见第 274 页中的 “通过重新初始化重新编制后缀的索引”。

注 – 如果在多个后缀上并行运行 dsconf import 和/或 dsconf reindex 命令，则事务日志将会增大，并可能对性能造成不利影响。

▼ 删除索引

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **删除为属性配置的全部索引。**

```
$ dsconf delete-index -h host -p port suffix-DN attr-name
```

例如，以下命令将删除 preferredLanguage 属性的全部索引：

```
$ dsconf delete-index -h host -p port dc=example,dc=com preferredLanguage
```

删除默认索引时应非常谨慎，因为它可能会影响目录服务器的功能。

更改索引列表阈值

搜索速度缓慢可能是因为系统索引列表大小超过了索引列表阈值。索引列表阈值是每个索引键的最大值数。要确定是否已超过索引列表阈值大小，请检查访问日志。访问日志 RESULT 消息末尾的 notes=U 标志表明执行了未编制索引的搜索。前面的 SRCH 消息（属于同一个连接和操作）指定已使用的搜索过滤器。以下两行示例将跟踪一个未编制索引的搜索 cn=Smith，该搜索将返回 10,000 个条目。已从这些消息中删除了时间戳。

```
conn=2 op=1 SRCH base="o=example.com" scope=0 filter="(cn=Smith)"
conn=2 op=1 RESULT err=0 tag=101 nentries=10000 notes=U
```

如果您的系统经常超过索引列表阈值，请考虑增加阈值以提高性能。以下过程将使用 dsconf set-server-prop 命令修改 all-ids-threshold 属性。有关调整索引和 all-ids-threshold 属性的详细信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Tuning Indexes for Performance”。

▼ 更改索引列表阈值

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 调整索引列表阈值。

可在以下任何级别上调整索引列表阈值：

- 在实例级别：

  ```
  dsconf set-server-prop -h host -p port all-ids-threshold:value
  ```

- 在后缀级别：

  ```
  dsconf set-suffix-prop -h host -p port suffix-DN all-ids-threshold:value
  ```

- 在条目级别：

  ```
  dsconf set-index-prop -h host -p port suffix-DN all-ids-threshold:value
  ```

- 在索引级别（按搜索类型）：

  ```
  dsconf set-index-prop -h host -p port suffix-DN all-ids-threshold search-type:value
  ```

  其中 *search-type* **为以下任一选项：**

  - `eq-enabled` 等同
  - `pres-enabled` 存在
  - `sub-enabled` 子串

  无法为近似索引配置 `all-ids-threshold` 属性。

可以使用 DSCC 在索引级别（按搜索类型）设置阈值。有关详细信息，请参见目录服务器联机帮助。

2 重新生成后缀索引。

请参见第 271 页中的 “生成索引”。

3 如果已针对旧的所有 `ID` 阈值调整了数据库缓存大小，并且服务器有足够的物理内存，请考虑增加数据库缓存大小。

可以按照**所有 `ID` 阈值**增幅的 25% 来增加数据库缓存大小。

换句话说，如果将**所有 `ID` 阈值**从 4000 增加到 6000，则根据索引列表大小的增幅，可以将数据库缓存大小大致增加 12.5%。

数据库缓存大小是使用属性 `dbcachesize` 设置的。在将更改应用于生产服务器之前，应该先根据经验找到最合适的大小。

重新编制后缀的索引

如果索引文件已损坏，则必须重新编制后缀的索引，以便在相应的数据库目录中重新创建索引文件。您可以在目录服务器运行期间重新编制后缀的索引，也可以通过重新初始化后缀来重新编制后缀的索引。

在目录服务器运行期间重新编制后缀的索引

重新编制后缀的索引时，服务器将检查此后缀包含的所有条目，并重新创建索引文件。在重新编制索引期间后缀内容处于只读状态。由于服务器必须扫描整个后缀来查找要编制索引的每个属性，因此如果后缀包含数百万条目，则完成此过程可能需要几个小时的时间。时间长度还取决于您配置的索引。此外，在重新编制后缀的索引时，索引将不可用，并且服务器性能将受到影响。

▼ 在后缀上重新编制全部索引

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **在后缀上重新编制全部索引。**

  ```
  $ dsconf reindex -h host -p port suffix-DN
  ```

  例如，要在 `dc=example,dc=com` 后缀上初始化全部索引，请使用以下命令：

  ```
  $ dsconf reindex -h host -p port dc=example,dc=com
  ```

通过重新初始化重新编制后缀的索引

重新初始化后缀时将会导入新内容，这意味着此后缀的内容将被替换，并将创建新的索引文件。重新初始化后缀可能比为多个属性重新编制索引更快，因为在装入条目时将为所有属性并行编制索引。但是请注意，后缀在重新初始化期间不可用。

▼ 通过重新初始化为后缀重新编制索引

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1. **将后缀设置为只读状态，如第 58 页中的 “设置引用并使后缀变为只读状态”中所述。**
2. **将整个后缀导出到 LDIF 文件，如第 182 页中的 “备份到 LDIF”中所述。**
3. **导入同一个 LDIF 文件以重新初始化后缀，如第 184 页中的 “从 LDIF 文件导入数据”中所述。**

   后缀在初始化期间不可用。初始化完成后，即可使用所有已配置的索引。

4 **再次将后缀设置为可写状态，如第 58 页中的“设置引用并使后缀变为只读状态”中所述。**

管理浏览索引

浏览索引是一种特殊索引，仅用于请求服务器端结果排序的搜索操作。《Sun Java System Directory Server Enterprise Edition 6.0 Reference》说明了浏览索引在目录服务器中的工作方式。

用于客户端搜索的浏览索引

必须手动定义用于对客户端搜索结果进行排序的自定义浏览索引。要创建浏览索引，也称为虚拟列表视图 (Virtual List View, VLV) 索引，请使用以下过程。本部分还包括添加或修改浏览索引条目以及重新生成浏览索引的过程。

▼ 创建浏览索引

对于此过程的某些部分，可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的“目录服务控制中心界面”和 DSCC 联机帮助。此过程的其他部分只能使用命令行完成。

1 **使用 `ldapmodify` 命令添加新的浏览索引条目或编辑现有的浏览索引条目。**

有关说明，请参见第 275 页中的“添加或修改浏览索引条目”。

2 **运行 `dsconf reindex` 命令以生成一组新的要由服务器维护的浏览索引。**

有关说明，请参见第 277 页中的“重新生成浏览索引”。

▼ 添加或修改浏览索引条目

浏览索引专用于给定基条目及其子树上的给定搜索。浏览索引配置在包含条目的后缀的数据库配置中进行定义。

1 **为目录服务器上的每个浏览索引配置 `vlvBase`、`vlvScope` 和 `vlvFilter` 属性。**

这些属性用于配置搜索基、搜索范围和搜索过滤器。这些属性将使用 `vlvSearch` 对象类。

2 **为每个浏览索引配置 `vlvSort` 属性。**

此属性指定用于对索引进行排序的属性的名称。此条目是第一个条目的子条目，并使用 `vlvIndex` 对象类指定要进行排序的属性以及排序顺序。

以下示例使用 ldapmodify 命令创建浏览索引配置条目：

```
$ ldapmodify -a -h host -p port -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=people_browsing_index, cn=database-name,
cn=ldbm database,cn=plugins,cn=config
objectClass: top
objectClass: vlvSearch
cn: Browsing ou=People
vlvBase: ou=People,dc=example,dc=com
vlvScope: 1
vlvFilter: (objectclass=inetOrgPerson)
dn: cn=Sort rev employeenumber, cn=people_browsing_index,
 cn=database-name,cn=ldbm database,cn=plugins,cn=config
objectClass: top
objectClass: vlvIndex
cn: Sort rev employeenumber
vlvSort: -employeenumber
^D
```

vlvScope 为以下任一选项：

- 0（仅基条目）
- 1（基条目的直接子条目）
- 2（以基条目为根的整个子树）

vlvFilter 是客户端搜索操作中使用的 LDAP 过滤器。由于所有浏览索引条目都位于相同位置，因此您应该使用描述性的 cn 值来命名浏览索引。

每个 vlvSearch 条目都必须至少有一个 vlvIndex 条目。vlvSort 属性是属性名称列表，用于定义要排序的属性及排序顺序。属性名称前面的破折号 (-) 表明使用相反的顺序。通过定义多个 vlvIndex 条目可以为搜索定义多个索引。您可以为上面的示例添加以下条目：

```
$ ldapmodify -a -h host -p port
 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=Sort sn givenname uid, cn=people_browsing_index,
 cn=database-name,cn=ldbm database,cn=plugins,cn=config
objectClass: top
objectClass: vlvIndex
cn: Sort sn givenname uid
vlvSort: sn givenname uid
^D
```

3 **要修改浏览索引配置，请编辑相应的 vlvSearch 条目或相应的 vlvIndex 条目。**

4 **要删除某个浏览索引以便服务器不再维护该浏览索引，请删除单个的 vlvIndex 条目。**
或者，如果只有一个 vlvIndex 条目，请同时删除 vlvSearch 条目和 vlvIndex 条目。

▼ 重新生成浏览索引

- **创建浏览索引条目之后，将为指定属性生成新的浏览索引。**

  ```
  $ dsadm reindex -l -t attr-index instance-path suffix-DN
  ```

  此命令将扫描目录内容，并为浏览索引创建数据库文件。

  以下示例将生成您在上一部分定义的浏览索引：

  ```
  $ dsadm reindex -l -b database-name -t Browsing /local/ds \
   ou=People,dc=example,dc=com
  ```

  有关 `dsadm reindex` 命令的详细信息，请参见 `dsadm`(1M) 手册页。

◆ ◆ ◆ 第 13 章

13 目录服务器属性值唯一性

UID 唯一性插件可确保给定属性的值在目录或子树的所有条目中是唯一的。如果某个操作尝试添加包含给定属性现有值的条目，此插件将会停止该操作。如果某个操作添加目录中已存在的值，或者将属性值修改为目录中已存在的值，此插件也会停止该操作。

默认情况下将禁用 UID 唯一性插件。启用此插件时，默认情况下可确保 uid 属性的唯一性。您可以创建此插件的新实例，以便在其他属性上实现属性值唯一性。UID 唯一性插件可确保单个服务器上的属性值唯一性。

本章包含以下主题：

属性值唯一性概述

UID 唯一性插件是预操作插件。在服务器执行目录更新之前，它将检查 LDAP 添加、修改和修改 DN 操作。此插件可确定操作是否会导致两个条目具有相同的属性值。如果相同，服务器将终止此操作，并向客户端返回错误 19 LDAP_CONSTRAINT_VIOLATION。

可以对此插件进行配置，以便在目录的一个或多个子树中或特定对象类的条目之间实现唯一性。此配置可确定要实现属性值唯一性的条目集。

如果要实现其他属性的唯一性，则可以定义多个 UID 唯一性插件的实例。请为每个必须具有唯一值的属性定义一个插件实例。还可以为同一属性定义多个插件实例，以便在多个条目集中“分别”实现唯一性。给定的属性值在每个子树集中只允许一次。

在现有目录上启用属性唯一性时，服务器不会检查现有条目间的唯一性。只有在添加条目或者添加或修改属性时才实现唯一性。

默认情况下将禁用 UID 唯一性插件，因为此插件会影响多主复制。使用复制时可以启用 UID 唯一性插件，但您应该了解第 282 页中的 “将唯一性插件用于复制”中描述的内容。

实现 uid 和其他属性的唯一性

本部分介绍如何为 uid 属性启用和配置默认唯一性插件，以及如何实现任何其他属性的唯一性。

▼ 实现 uid 属性的唯一性

此过程说明如何使用 dsconf 命令启用和配置 UID 唯一性插件。此插件配置条目的 DN 为 cn=uid uniqueness,cn=plugins,cn=config。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

使用 DSCC 时，不得修改默认的 UID 唯一性插件以实现其他属性的唯一性。如果不需要使用 UID 唯一性插件，请将此插件保留为禁用状态，并为其他属性创建新的插件实例，如第 281 页中的 “实现其他属性的唯一性”中所述。

1. **启用插件。**

```
$ dsconf enable-plugin -h host -p port "uid uniqueness"
```

2. **根据要实现唯一性的子树的指定方式，修改插件参数。**

   - 要指定单个子树的基 DN，请键入：

```
$ dsconf set-plugin-prop -h host -p port "uid uniqueness" argument:uid argument:subtreeBaseDN
```

例如：

```
$ dsconf set-plugin-prop -h host1 -p 1389 "uid uniqueness" argument:uid \
 argument:dc=People,dc=example,dc=com
```

   - 要指定多个子树，请添加更多的参数，并将子树的完整基 DN 作为每个参数的值。

```
$ dsconf set-plugin-prop -h host -p port "uid uniqueness" argument:uid \
 argument:subtreeBaseDN argument:subtreeBaseDN
```

   - 要根据子树基条目的对象类指定这些子树，请将参数设置为以下值。将在每个具有 *baseObjectClass* 的条目下的子树中实现 uid 属性唯一性。也可以选择在第三个参数中指定 *entryObjectClass*，以便插件只在特定操作（以具有此对象类的条目为目标）中实现唯一性。

```
$ dsconf set-plugin-prop -h host -p port "uid uniqueness" argument:attribute=uid \
 argument:markerObjectClass=baseObjectClass argument:entryObjectClass=baseObjectClass
```

3 **重新启动服务器以使更改生效。**

▼ 实现其他属性的唯一性

UID 唯一性插件可用于实现任何属性的唯一性。您必须在目录的 cn=plugins,cn=config 下创建新条目，以创建此插件的新实例。

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1 **创建新插件。**

```
$ dsconf create-plugin -h host -p port -H lib-path -F init-func \
 -Y type plugin-name
```

plugin-name 应该是简短的描述性名称，其中包含属性名称。例如，要创建一个插件以实现邮件 ID 属性的唯一性，请使用以下命令：

```
$ dsconf create-plugin -h host1 -p 1389 -H /opt/SUNWdsee/ds6/lib/uid-plugin.so \
 -F NSUniqueAttr_Init -Y preoperation "mail uniqueness"
```

2 **设置插件属性。**

```
$ dsconf set-plugin-prop -h host -p port plugin-name property:value
```

例如，要设置邮件唯一性插件的属性，请使用以下命令：

```
$ dsconf set-plugin-prop -h host1 -p 1389 "mail uniqueness" \
 desc:"Enforce unique attribute values..." version:6.0 \
 vendor:"Sun Microsystems, Inc." depends-on-type:database
```

3 **启用插件。**

```
$ dsconf enable-plugin -h host -p port plugin-name
```

4 **指定插件参数。**

这些参数取决于要实现唯一性的子树的确定方式。

- 要根据子树的基 DN 定义一个或多个子树，则第一个参数必须是应具有唯一值的属性的名称。后面的参数是这些子树基条目的完整 DN。

  ```
  $ dsconf set-plugin-prop -h host -p port plugin-name argument:attribute-name \
   argument:subtreeBaseDN argument:subtreeBaseDN...
  ```

- 要根据子树基条目的对象类定义子树，则第一个参数必须包含 `attribute=`*attribute-name*（用于指定应具有唯一值的属性的名称）。第二个参数必须是 *baseObjectClass*（用于确定要实现唯一性的子树的基条目）。也可以选择在第三个参数中指定 *entryObjectClass*，以便插件只在特定操作（以具有此对象类的条目为目标）中实现唯一性。

```
$ dsconf set-plugin-prop -h host -p port plugin-name argument:attribute=attribute-name \
 argument:markerObjectClass=baseObjectClass argument:requiredObjectClass=entryObjectClass
```

在所有插件参数中， = 符号的前后不能出现空格。

5 **重新启动服务器以使更改生效。**

将唯一性插件用于复制

将更新作为复制操作的一部分执行时，UID 唯一性插件不会对属性值执行任何检查。这不会影响单主复制，但此插件无法为多主复制自动实现属性唯一性。

单主复制方案

由于客户端应用程序所做的全部修改都在主副本上执行， 因此应该在主服务器上启用 UID 唯一性插件。应该将此插件配置为在复制的后缀中实现唯一性。由于主服务器可确保必需属性的值是唯一的，因此您不必在使用方服务器上启用此插件。

在单个主服务器的使用方上启用 UID 唯一性插件不会影响复制或普通的服务器操作。但是，它可能会导致性能略微下降。

多主复制方案

UID 唯一性插件不适用于多主复制方案。由于多主复制使用宽松的一致性复制模型，因此即使在两个服务器上都启用了此插件，也不会检测到同时在这两个服务器上添加相同属性值的操作。

但是，如果要执行唯一性检查的属性是命名属性，并且在所有主服务器上的相同子树中为同一属性启用了 UID 唯一性插件，则可以使用此唯一性插件。

满足上述条件时，在复制时会将一致性冲突报告为命名冲突。命名冲突必须手动解决。有关详细信息，请参见第 246 页中的 “解决常见复制冲突”。

◆ ◆ ◆ 第 14 章

目录服务器日志记录

本章介绍如何管理目录服务器日志。

如果需要可帮助您定义日志记录策略的信息，请使用《Sun Java System Directory Server Enterprise Edition 6.0 Deployment Planning Guide》中的“Designing a Logging Strategy”中的日志记录策略信息。

有关日志文件及其内容的描述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的第 7 章 “Directory Server Logging”。

本章包含以下主题：

日志分析工具

Directory Server Resource Kit 提供了日志分析工具 `logconv`，使用该工具可以分析目录服务器访问日志。日志分析工具可提取使用情况统计信息。此外，它还可以统计重要事件发生的次数。有关此工具的详细信息，请参见 `logconv`(1) 手册页。

查看目录服务器日志

可以直接在服务器的 *instance-path*`/logs` 中查看日志。

或者，也可以通过目录服务控制中心 (Directory Service Control Center, DSCC) 查看日志文件。使用 DSCC 可查看日志条目并对其进行排序。

下图显示了 DSCC 中的目录服务器访问日志样例。

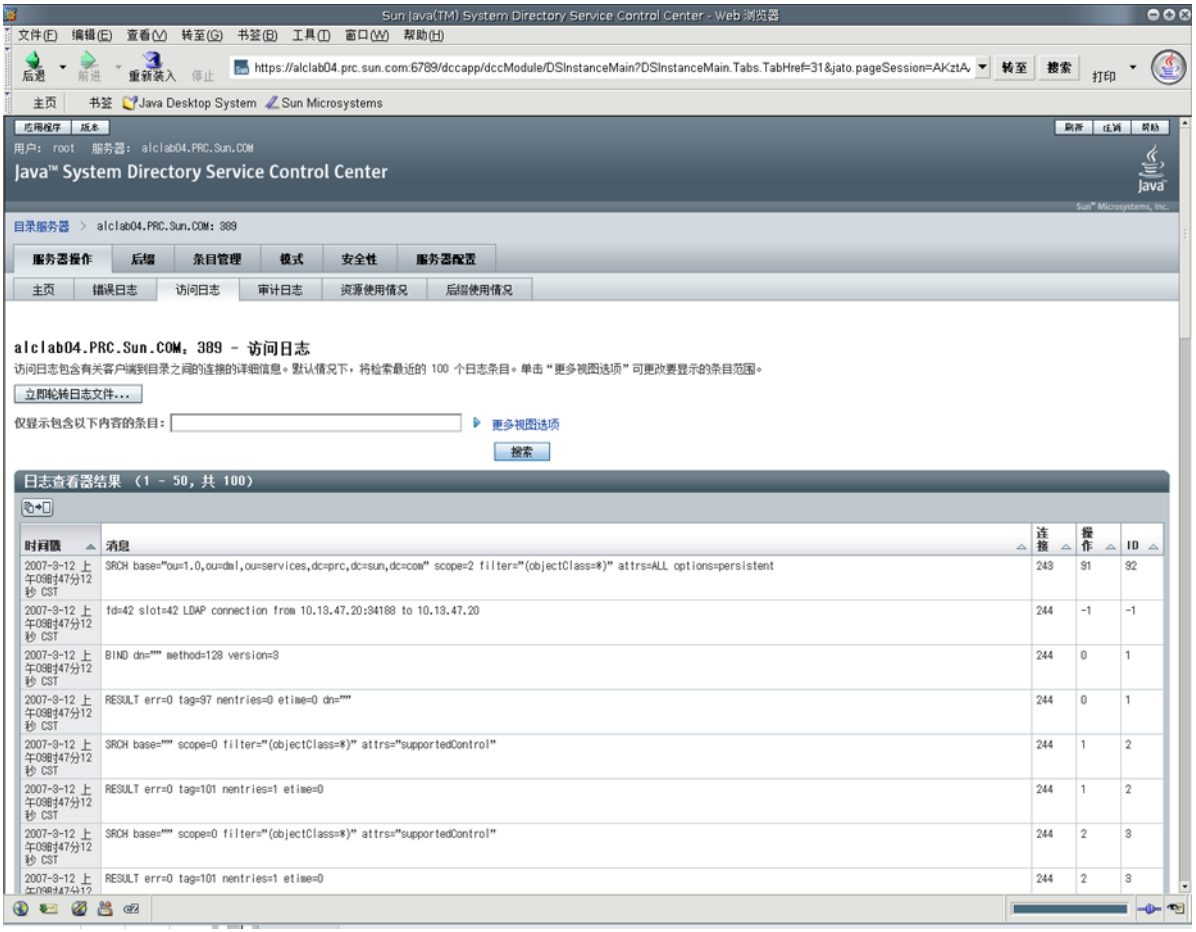


图 14–1 DSCC 访问日志

配置目录服务器日志

可以对日志文件的许多方面进行修改。以下提供了一些示例：

- 起用审计日志

  与访问日志和错误日志不同，默认情况下不启用审计日志。有关信息，请参见第 286 页中的“启用审计日志”。

- 常规设置
  - 启用或禁用日志记录
  - 日志文件位置
  - 详细日志记录
  - 日志级别

- 日志轮转设置。
  - 定期创建新日志
  - 创建新日志文件前的最大日志文件大小
- 日志删除设置
  - 删除前的最大文件存留期
  - 删除前的最大文件大小
  - 删除前的最小可用磁盘空间

以下过程介绍如何修改日志配置以及如何启用审计日志。

▼ 修改日志配置

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1 查看要修改的日志的设置。

```
$ dsconf get-log-prop -h host -p port log-type
```

例如，要列出现有的错误日志设置，请键入：

```
$ dsconf get-log-prop -h host1 -p 1389 error
Enter "cn=Directory Manager" password:
enabled                   :  on
level                     :  default
max-age                   :  1M
max-disk-space-size       :  100M
max-file-count            :  2
max-size                  :  100M
min-free-disk-space-size  :  5M
path                      :  /tmp/ds1/logs/errors
perm                      :  600
rotation-interval         :  1w
rotation-min-file-size    :  unlimited
rotation-time             :  undefined
verbose-enabled           :  off
```

2 设置新值。

设置所需的属性值。

```
$ dsconf set-log-prop -h host -p port log-type property:value
```

例如，要将错误日志的轮转时间间隔设置为两天，请使用以下命令：

```
$ dsconf set-log-prop -h host1 -p 1389 error rotation-interval:2d
```

▼ 启用审计日志

与访问日志和错误日志不同，默认情况下不启用审计日志。在查看审计日志之前，必须先启用审计日志。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **启用审计日志。**

```
$ dsconf set-log-prop -h host -p port audit enabled:on
```

手动轮转目录服务器日志

如果日志变得很大，则可以随时手动轮转日志。轮转将备份现有日志文件并创建新的日志文件。

▼ 手动轮转日志文件

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **轮转日志文件。**

```
$ dsconf rotate-log-now -h host -p port log-type
```

例如，要轮转访问日志：

```
$ dsconf rotate-log-now -h host1 -p 1389 access
```

◆ ◆ ◆ 第 15 章

15 目录服务器监视

可以使用多种方法监视目录服务器。《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的第 3 章 “Directory Server Monitoring”中介绍了这些方法。

本章介绍如何在目录服务器中设置和管理监视。

本章包含以下主题：

为目录服务器设置 SNMP

本部分介绍如何将服务器设置为通过 SNMP 进行监视。

有关在目录服务器中实现 SNMP 的描述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Directory Server and SNMP”。

▼ 设置 SNMP

对于此过程的某些部分，可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。此过程的其他部分只能使用命令行完成。

1 启用 Java ES 管理框架插件。

可使用过程第 288 页中的 “启用 Java ES MF 监视” 。此过程还会启用 Common Agent Container，它是 Java ES MF 的一部分。

2 **访问 SNMP 管理的对象，这些对象由 MIB 定义并通过代理公开。**

此步骤所需执行的任务完全依赖于 SNMP 管理系统。有关说明，请参见 SNMP 管理系统文档。

公开 MIB 时，您可能需要为此 MIB 使用 RFC 文本文件。可以在 `http://www.ietf.org/rfc/rfc2605.txt` 和 `http://www.ietf.org/rfc/rfc2788.txt` 中找到这些文件。

启用 Java ES MF 监视

如果使用 Sun Java ES 管理框架 (Java ES Management Framework, Java ES MF) 进行监视，则必须启用 Java ES MF 插件。

有关管理 Java ES MF 的详细信息，请参见《Sun Java Enterprise System 5 Monitoring Guide》。

▼ 启用 Java ES MF 监视

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 **初始化并注册 Java ES 监视框架。**

```
$ dsccsetup mfwk-reg
```

有关此命令的位置，请参见第 31 页中的 “命令位置”。

2 **启用 Java ES 管理框架插件。**

```
$ dsconf enable-plugin -h host -p port "Monitoring Plugin"
Enter "cn=Directory Manager" password:
Directory Server must be restarted for changes to take effect.
```

3 **重新启动目录服务器实例。**

```
$ dsadm restart instance-path
```

4 **验证是否已启用 Java ES 管理框架插件。**

```
$ dsconf get-plugin-prop -h host -p port -v "Monitoring Plugin"
Enter "cn=Directory Manager" password:
Reading property values of the plugin "Monitoring Plugin"...
argument          :
depends-on-named  :
depends-on-type   :  database
desc              :  Monitoring plugin
enabled           :  on
```

```
feature           :  Monitoring
init-func         :  mf_init
lib-path          :  /opt/SUNWdsee/ds6/lib/mf-plugin.so
type              :  object
vendor            :  Sun Microsystems, Inc.
version           :  6.0
```

Java ES MF 监视故障排除

如果 Java ES MF 监视无法工作，请确保您已正确安装 Common Agent Container，如《Sun Java System Directory Server Enterprise Edition 6.0 Installation Guide》中的第 1 章 “Installation”中所述。

如果仍存在问题，请参见《Sun Java Enterprise System 5 Monitoring Guide》。

使用 cn=monitor 监视服务器

可以通过 DSCC 获取服务器状态、复制状态、资源使用情况以及其他监视信息。

此外，还可以通过对以下条目执行搜索操作，从任何 LDAP 客户端监视目录服务器的当前活动：

- `cn=monitor`
- `cn=monitor, cn=ldbm database, cn=plugins, cn=config`
- `cn=monitor,cn=`*dbName* `,cn=ldbm database,cn=plugins,cn=config`

  dbName 是要监视的后缀的数据库名称。请注意，在默认情况下，除了有关每个连接的信息之外，任何用户（包括匿名绑定的客户端）都可以读取 `cn=monitor` 条目。

  以下示例显示如何查看一般的服务器统计信息：

  ```
  $ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
   -s base -b "cn=monitor" "(objectclass=*)"
  ```

  有关这些条目中所有可用监视属性的描述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Directory ServerMonitoring Attributes”。

  许多可以监视的参数都可反映目录服务器性能，并且这些参数会受配置和调整的影响。有关每种可配置属性的详细信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Man Page Reference》中的属性手册页。

第 2 部分

目录代理服务器管理

◆ ◆ ◆ 第 16 章

16 目录代理服务器工具

Sun Java™ System Directory Proxy Server 提供了浏览器界面和命令行工具，用于注册和管理目录代理服务器实例。浏览器界面称为目录服务控制中心 (Directory Service Control Center, DSCC)。本章介绍使用 DSCC 或命令行管理目录代理服务器时需要执行的基本任务。

要确定是使用 DSCC 还是命令行来执行特定任务，请参见第 39 页中的 “DSCC 和命令行的适用环境”。

有关管理框架的详细信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Deployment Planning Guide》中的“Directory Server Enterprise Edition Administration Model”。

本章包含以下主题：

使用目录代理服务器的 DSCC

本部分介绍如何访问目录代理服务器的 DSCC。

▼ 访问目录代理服务器的 DSCC

1 **访问 DSCC（与访问目录服务器的 DSCC 方式相同）。**
请参见第 41 页中的 “访问 DSCC”。

2 **单击“代理服务器”选项卡以查看和管理目录代理服务器。**
下图显示了目录代理服务器的初始窗口。

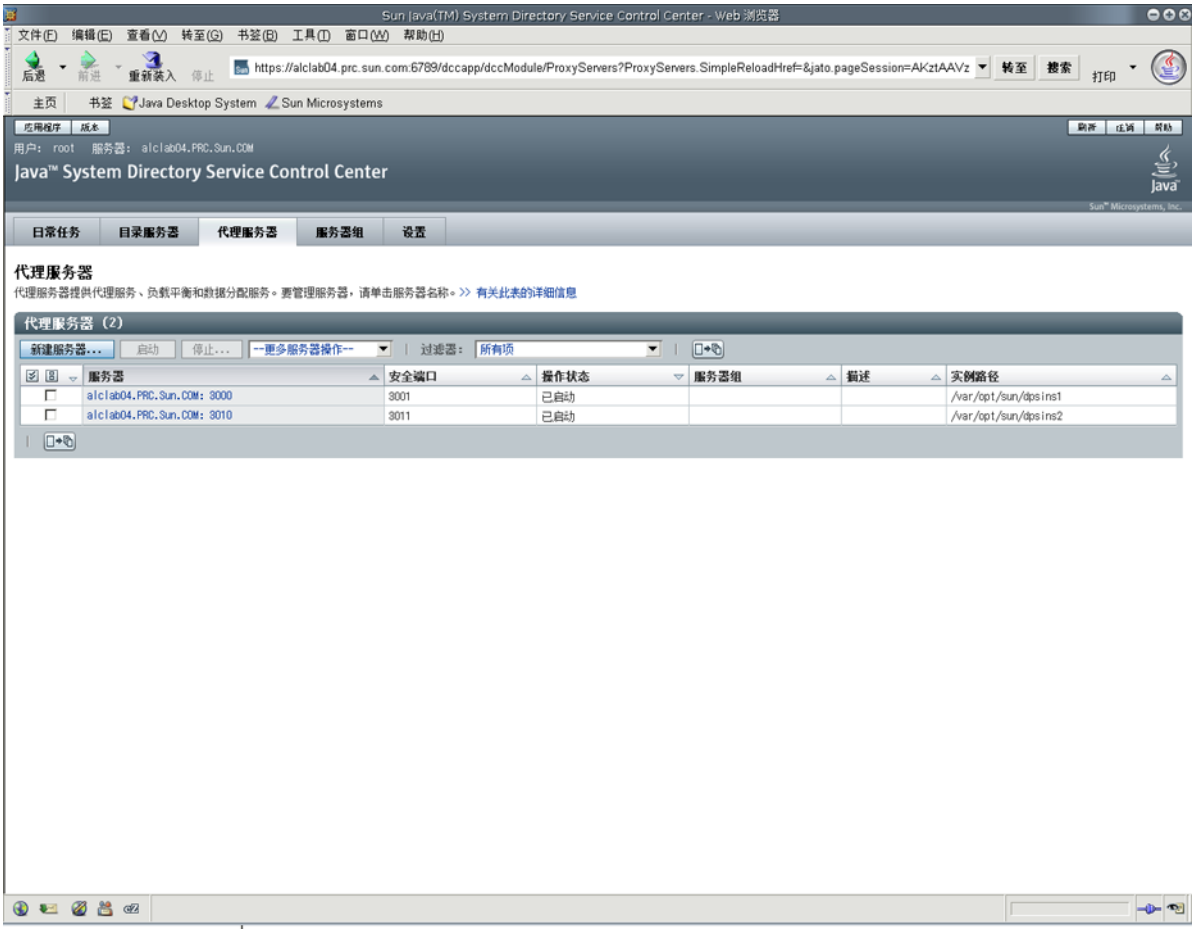


图 16–1 目录代理服务器的初始 DSCC 窗口

3 **单击某个目录代理服务器实例以查看或管理该服务器。**

注 – 有关使用 DSCC 的详细信息，请参见联机帮助。

目录代理服务器的命令行工具

在目录代理服务器中使用的命令行工具称为 `dpadm` 和 `dpconf`。有关如何使用这些命令的信息，请参见 `dpadm`(1M) 和 `dpconf`(1M) 手册页。

本部分介绍 `dpadm` 和 `dpconf` 命令的位置。此外还提供以下方面的信息：环境变量、命令之间的比较，以及如何获取有关使用这些命令的帮助。

目录代理服务器命令的位置

默认情况下，目录代理服务器命令行工具位于以下目录中：

install-path/dps6/bin

安装路径取决您的操作系统。第 30 页中的 “默认路径和命令位置”中列出了所有操作系统的安装路径。

设置 dpconf 的环境变量

dpconf 命令需要一些可以使用环境变量进行预设的选项。如果使用此命令时不指定选项，或者不设置环境变量，将使用默认设置。可以为以下选项配置环境变量：

-D *userDN*	用户绑定 DN。环境变量：LDAP_ADMIN_USER。默认：cn=Proxy Manager。
-w *password-file*	用户绑定 DN 的密码文件。环境变量：LDAP_ADMIN_PWF。默认：提示输入密码。
-h *host*	主机名或 IP 地址。环境变量：DIR_PROXY_HOST 。默认：localhost。
-p *LDAP-port*	LDAP 端口号。环境变量：DIR_PROXY_PORT。默认：389（如果服务器实例作为**超级用户**运行），或 1389（如果服务器实例作为普通用户运行）。

有关详细信息，请参见 dpconf(1M) 手册页。

dpadm 和 dpconf 的比较

下表显示了 dpadm 和 dpconf 命令的比较。

表 16–1 dpadm 和 dpconf 命令的比较

	dpadm **命令**	dpconf **命令**
用途	管理本地目录代理服务器实例上的进程或文件	配置本地或远程目录代理服务器实例
用户	操作系统用户	LDAP 用户
本地或远程	对实例而言，此命令**必须**在本地运行，也就是说，必须在运行服务器的主机上运行此命令。	对实例而言，此命令**可以**在本地运行，但也可以从网络上的任何位置运行。

表 16–1 dpadm 和 dpconf 命令的比较 （续）

	dpadm 命令	dpconf 命令
命令使用示例	创建目录代理服务器实例。 启动和停止目录代理服务器实例。 管理证书数据库。	修改目录代理服务器实例的配置。 创建数据视图。 配置数据源池中的负载平衡。
服务器状态	服务器可以运行或停止。	服务器**必须**运行。
命令如何标识服务器实例	通过指定实例路径。实例路径可以是相对路径或绝对路径。	通过指定主机名（或 IP 地址）和端口号。 该命令使用 LDAP 端口 (-p) 或 LDAPS 安全端口 (-P)。如果未在命令行指定端口号，则使用环境变量 PROXY_PORT。如果未设置环境变量，则使用默认端口。

使用 dpconf 设置多值属性

某些目录代理服务器属性可以采用多个值。用于指定这些值的语法如下：

```
$ dpconf set-container-prop -h host -p port container-name \
 property:value1 property:value2
```

例如，要为名为 my-view 的 LDAP 数据视图设置多个可写属性，请使用以下命令：

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 my-view \
 writable-attr:uid writable-attr:cn writable-attr:userPassword
```

如果为已包含值的多值属性添加或修改值，必须重置**所有**值。例如，在上述方案中，如果要添加 sn 作为可写属性，则必须在此命令中包括所有其他的可写属性：

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 my-view \
 writable-attr:uid writable-attr:cn writable-attr:userPassword writable-attr:sn
```

如果要删除某个值，此规则同样适用。因此，要从前面示例的可写属性列表中删除 userPassword，请使用以下命令：

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 my-view \
 writable-attr:uid writable-attr:cn writable-attr:sn
```

获取有关使用 dpadm 和 dpconf 的帮助

有关如何使用 dpadm 和 dpconf 命令的信息，请参见 dpadm(1M) 和 dpconf(1M) 手册页。

- 要获取子命令列表，请键入相应的命令：

```
$ dpadm --help

$ dpconf --help
```

- 要获取有关如何使用子命令的信息，请键入相应的命令：

```
$ dpadm subcommand --help

$ dpconf subcommand --help
```

- 要获取有关 dpconf 命令中所使用的配置属性的信息，请键入：

```
$ dpconf help-properties
```

- 要获取有关子命令的配置属性的信息，请使用以下命令：

```
$ dpconf help-properties subcommand-entity
```

例如，要查找有关访问日志属性的信息，请键入：

```
$ dpconf help-properties access-log
```

- 要获取有关子命令中所使用的某个属性的信息，请使用以下命令：

```
$ dpconf help-properties subcommand-entity property
```

例如，要查找有关 set-access-log-prop 子命令的 log-search-filters 属性的信息，请键入：

```
$ dpconf help-properties access-log log-search-filters
```

- 要列出一组实体（如数据视图或连接处理程序）的主要属性，请在 list 子命令中使用详细选项 -v。

例如，要查看所有连接处理程序的主要属性和相对优先级，请使用以下命令：

```
$ dpconf -h host -p port list-connection-handlers -v
Name                        is-enabled  priority  description
--------------------------  ----------  --------  --------------------------
anonymous                   false       99        unauthenticated connections
default connection handler  true        100       default connection handler
dscc administrators            true          1         Administrators connection
                                                    handler
```

有关单个属性的详细信息，请参见与该属性相对应的手册页。

◆ ◆ ◆ 第 17 章

17 目录代理服务器实例

本章介绍如何管理目录代理服务器实例。本章包含以下主题：

创建和删除目录代理服务器实例

创建目录代理服务器实例时，将在您指定的路径中创建此实例所需的文件和目录。

▼ 创建目录代理服务器实例

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

如果使用 DSCC 创建新的服务器实例，则可以选择复制现有服务器中的部分或全部服务器配置设置。

1 **创建目录代理服务器实例。**

```
$ dpadm create -p port instance-path
```

例如，要在 `/local/dps` 目录中创建新的实例，请使用以下命令：

```
$ dpadm create -p 2389 /local/dps
```

要指定实例的任何其他参数，请参见 `dpadm`(1M) 手册页。

2 **键入密码（如有必要）。**

3 **通过验证实例状态来确认是否已创建该实例。**

```
$ dpadm info instance-path
```

4 **（可选的）如果目录代理服务器已使用 Sun Java™ Enterprise System 安装程序或本地软件包安装进行安装，并且您的操作系统提供了服务管理解决方案，则可以将服务器作为服务进行管理，如下表所示。**

操作系统	命令
Solaris 10	`dpadm enable-service --type SMF` *instance-path*
Solaris 9	`dpadm autostart` *instance-path*
Linux、HP-UX	`dpadm autostart` *instance-path*
Windows	`dpadm enable-service --type WIN_SERVICE` *instance-path*

5 **（可选的）使用以下任一方法注册服务器实例：**

- 通过 URL `https://localhost:6789` 访问 DSCC，并登录到浏览器界面。
- 使用 `dsccreg add-server` 命令。

  有关详细信息，请参见 `dsccreg`(1M) 手册页。

▼ 删除目录代理服务器实例

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 **（可选的）停止目录代理服务器实例。**

```
$ dpadm stop instance-path
```

如果不停止此实例，删除命令会自动将其停止。但是，如果已在服务管理解决方案中启用了此实例，则必须手动将其停止。

2 **（可选的）如果以前使用 DSCC 管理服务器，请使用命令行取消注册该服务器。**

```
$ dsccreg remove-server /local/dps
Enter DSCC administrator's password:
/local/dps is an instance of DPS
Enter password of "cn=Proxy Manager" for /local/dps:
Unregistering /local/dps from DSCC on localhost.
Connecting to /local/dps
Disabling DSCC access to /local/dps
```

有关详细信息，请参见 `dsccreg`(1M) 手册页。

3 **（可选的）如果以前在服务管理解决方案中启用了服务器实例，请停止将此服务器作为服务进行管理。**

操作系统	命令
Solaris 10	`dpadm disable-service --type SMF` *instance-path*
Solaris 9	`dpadm autostart --off` *instance-path*
Linux、HP-UX	`dpadm autostart --off` *instance-path*
Windows	`dpadm disable-service --type WIN_SERVICE` *instance-path*

4 **删除实例。**

```
$ dpadm delete instance-path
```

查找目录代理服务器实例的状态

以下过程介绍如何查找目录代理服务器实例的状态。

▼ 查找目录代理服务器实例的状态

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **查找目录代理服务器实例的状态。**

```
$ dpadm info instance-path
```

启动、停止和重新启动目录代理服务器实例

本部分提供有关从命令行启动、停止和重新启动目录代理服务器的信息。

▼ 启动和停止目录代理服务器

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **要启动或停止目录代理服务器，请执行以下任一操作。**
  - 要启动目录代理服务器，请键入：

    ```
    $ dpadm start instance-path
    ```

    例如，要启动 `/local/dps` 中的实例，请使用以下命令：

    ```
    $ dpadm start /local/dps
    ```

- 要停止目录代理服务器，请键入：

  ```
  $ dpadm stop instance-path
  ```

  例如：

  ```
  $ dpadm stop /local/dps
  ```

▼ 查看是否需要重新启动目录代理服务器实例

有时，配置更改需要重新启动服务器才能生效。可使用以下过程查看是否需要在更改配置后重新启动目录代理服务器实例。

- **查看是否需要重新启动服务器。**

  ```
  $ dpconf get-server-prop -h host -p port is-restart-required
  ```

  - 如果命令返回 `true`，则必须重新启动目录代理服务器实例。
  - 如果命令返回 `false`，则无需重新启动目录代理服务器实例。

▼ 重新启动目录代理服务器

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **重新启动目录代理服务器。**

  ```
  $ dpadm restart instance-path
  ```

  例如，要重新启动 `/local/dps` 中的实例，请使用以下命令：

  ```
  $ dpadm restart /local/dps
  ```

◆ ◆ ◆ 第 18 章

18 目录代理服务器配置

本章介绍如何配置目录代理服务器实例。本章中的过程使用 dpadm 和 dpconf 命令。有关这些命令的信息，请参见 dpadm(1M) 和 dpconf(1M) 手册页。

本章包含以下主题：

修改目录代理服务器的配置

本部分介绍如何修改目录代理服务器的配置。

▼ 修改目录代理服务器的配置

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1 查找目录代理服务器的当前配置。

```
$ dpconf get-server-prop -h host -p port
```

或者，查看一个或多个配置属性的当前设置。

```
$ dpconf get-server-prop -h host -p port property-name ...
```

例如，通过运行以下命令确定是否允许未经验证的操作：

```
$ dpconf get-server-prop -h host -p port allow-unauthenticated-operations
allow-unauthenticated-operations  :  true
```

2 **更改一个或多个配置参数。**

```
$ dpconf set-server-prop -h host -p port property:value ...
```

例如，通过运行以下命令禁止未经验证的操作：

```
$ dpconf set-server-prop -h host -p port allow-unauthenticated-operations:false
```

如果您尝试执行非法更改，则此更改将不会生效。例如，如果将 `allow-unauthenticated-operations` 参数设置为 `f` 而不是 `false`，则会产生以下错误：

```
$ dpconf set-server-prop -h host -p port allow-unauthenticated-operations:f
The value "f" is not a valid value for the property "allow-unauthenticated-operations".
Allowed property values: BOOLEAN
The "set-server-prop" operation failed.
```

3 **重新启动目录代理服务器实例以使更改生效（如有必要）。**

有关重新启动目录代理服务器的信息，请参见第 302 页中的 “重新启动目录代理服务器”。

备份和恢复目录代理服务器实例

使用 `dpadm` 备份目录代理服务器时，将备份配置文件和服务器证书。如果已实现目录代理服务器虚拟 ACI，也会备份 ACI。

只要服务器成功启动，目录代理服务器即会自动备份 `conf.ldif` 文件。

▼ 备份目录代理服务器实例

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 **停止目录代理服务器实例。**

```
$ dpadm stop instance-path
```

2 **备份目录代理服务器实例。**

```
$ dpadm backup instance-path archive-dir
```

archive-dir 目录由 `backup` 命令创建，并且在运行此命令之前不得存在。此目录包含每个配置文件和证书的备份。

▼ 恢复目录代理服务器实例

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 **停止目录代理服务器实例。**

```
$ dpadm stop instance-path
```

2 **恢复目录代理服务器实例。**

```
$ dpadm restore instance-path archive-dir
```

- 如果实例路径存在，将以无提示方式执行恢复操作。*archive-dir* 目录中的配置文件和证书将替换 *instance-path* 目录中的配置文件和证书。
- 如果实例路径不存在，恢复操作将会失败。

配置代理管理员

代理管理员是特权管理员，与 UNIX® 系统上的 root 用户类似。代理管理员条目是在创建目录代理服务器实例时定义的。代理管理员的默认 DN 为 `cn=Proxy Manager`。

可以查看和更改代理管理员的 DN 和密码，如以下过程所示。

▼ 配置代理管理员

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 **查找代理管理员的配置。**

```
$ dpconf get-server-prop -h host -p port configuration-manager-bind-dn configuration-manager-bind-pwd
configuration-manager-bind-dn    :  cn=proxy manager
configuration-manager-bind-pwd   :  {3DES}U77v39WX8MDpcWVrueetB0lfJlBc6/5n
```

代理管理员的默认值为 `cn=proxy manager`。将为配置管理员密码返回一个散列值。

2 **更改代理管理员的 DN。**

```
$ dpconf set-server-prop -h host -p port configuration-manager-bind-dn:bindDN
```

3 **创建包含代理管理员密码的文件，并设置指向该文件的属性。**

```
$ dpconf set-server-prop -h host -p port configuration-manager-bind-pwd-file:filename
```

需要重新启动服务器的配置更改

对目录代理服务器及其实体的大多数配置更改都可以联机进行。某些更改需要重新启动服务器后才能生效。如果对下表中的任何属性进行配置更改，都必须重新启动服务器：

```
bind-dn
client-cred-mode
db-name
db-pwd
db-url
db-user
distribution-algorithm
ldap-address
ldap-port
ldaps-port
lexicographic-attrs
lexicographic-lower-bound
lexicographic-upper-bound
listen-address
listen-port
load-balancing-algorithm
num-bind-init
num-read-init
num-write-init
number-of-search-threads
number-of-threads
number-of-worker-threads
numeric-attrs
numeric-default-data-view
numeric-lower-bound
numeric-upper-bound
pattern-matching-base-object-search-filter
pattern-matching-dn-regular-expression
pattern-matching-one-level-search-filter
pattern-matching-subtree-search-filter
replication-role
ssl-policy
use-external-schema
```

属性的 `rws` 和 `rwd` 关键字表明对该属性的更改是否需要重新启动服务器。

- 如果某个属性具有 `rws`（read（读）、write（写）、static（静态））关键字，则在更改属性后必须重新启动服务器。
- 如果某个属性具有 `rwd`（read（读）、write（写）、dynamic（动态））关键字，则可动态完成对该属性的修改操作（无需重新启动服务器）。

要确定对某个属性的更改是否需要重新启动服务器，请运行以下命令：

```
$ dpconf help-properties | grep property-name
```

例如，要确定更改 LDAP 数据源的绑定 DN 是否需要重新启动服务器，请运行以下命令：

```
$ dpconf help-properties | grep bind-dn
connection-handler       bind-dn-filters        rwd  STRING | any
This property specifies a set of regular expressions. The bind DN
of a client must match at least one regular expression in order for
the connection to be accepted by the connection handler. (Default: any)
ldap-data-source      bind-dn               rws  DN | ""
This property specifies the DN to use when binding to the LDAP data
source. (Default: undefined)
```

要确定进行配置更改后是否必须重新启动服务器，请运行以下命令：

```
$ dpconf get-server-prop -h host -p port is-restart-required
```

使用目录代理服务器访问目录服务器的配置条目

目录代理服务器的配置条目位于 cn=config 中。在默认情况下，使用目录代理服务器访问配置条目时，将访问目录代理服务器的配置条目。

要访问目录服务器的配置条目，请使用目录服务器，而不要使用目录代理服务器。有关如何配置目录服务器的信息，请参见第 3 章。

注意 – 如果重新配置目录代理服务器以访问目录服务器的配置条目，很可能会破坏目录代理服务器的管理框架。

要使用目录代理服务器访问目录服务器的配置条目，请采用特殊步骤，以确保不会破坏目录代理服务器的管理框架。本部分介绍如何使用目录代理服务器访问目录服务器的配置条目。

▼ 使用目录代理服务器访问目录服务器的配置条目

1. **创建一个或多个数据源，如第 317 页中的“ 创建和配置 LDAP 数据源 ”中所述。**
2. **创建 LDAP 数据源池，如第 319 页中的“ 创建和配置 LDAP 数据源池 ”中所述。**

3 **将一个或多个数据源连接到此数据源池，如第 320 页中的“将 LDAP 数据源连接到数据源池”中所述。**

- **要公开某个特定数据源的配置条目，请只将一个 LDAP 数据源连接到 LDAP 数据源池。**

  ```
  $ dpconf attach-ldap-data-source -h host -p port pool-name data-source-name
  ```

  执行此步骤后，客户端即可访问连接到目录代理服务器的数据源的配置条目。

- **要公开任意数据源的配置条目，请将多个 LDAP 数据源连接到 LDAP 数据源池。**

  ```
  $ dpconf attach-ldap-data-source -h host -p port pool-name data-source-name \
   data-source-name ...
  ```

  执行此步骤后，客户端即可访问连接到目录代理服务器的任一数据源的配置条目。但是，客户端无法知道这些配置条目属于哪个数据源。

4 **创建 LDAP 数据视图以公开 `cn=config`。**

```
$ dpconf create-ldap-data-view -h host -p port view-name pool-name cn=dir-config
```

◆ ◆ ◆ 第 19 章

19 目录代理服务器证书

本章介绍如何在目录代理服务器上配置证书。有关在**目录服务器**上配置证书的信息，请参见第 102 页中的 “管理证书” 。

本章中的过程使用 `dpadm` 和 `dpconf` 命令。有关这些命令的信息，请参见 `dpadm`(1M) 和 `dpconf`(1M) 手册页。

本章包含以下主题：

默认的自签名证书

创建目录代理服务器实例时，该实例具有默认的自签名证书。自签名证书是一个公钥/私钥对，其中公钥是由目录代理服务器自签名的。

▼ 查看默认的自签名证书

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

- **查看默认的自签名证书。**

```
$ dpadm show-cert instance-path defaultservercert
```

创建、请求和安装目录代理服务器的证书

要在目录代理服务器上运行安全套接字层 (Secure Sockets Layer, SSL)，必须使用自签名证书或公钥基础结构 (Public Key Infrastructure, PKI) 解决方案。

PKI 解决方案需要涉及到外部证书颁发机构 (Certificate Authority, CA)。要使用 PKI 解决方案，您需要包含公钥和私钥的 CA 签名服务器证书。此证书特定于一个目录代理服务器实例。此外，还需要包含公钥的**可信 *CA* 证书**。可信 CA 证书可确保来自 CA 的所有服务器证书都是可信的。此证书有时称为 CA 根密钥或根证书。

有关如何创建非默认的自签名证书，以及如何请求和安装 CA 签名证书的信息，请参见以下过程。

▼ 创建非默认的目录代理服务器自签名证书

创建目录代理服务器实例时，将自动提供默认的自签名证书。如果要使用非默认设置创建自签名证书，请执行以下过程。

此过程将为服务器证书创建公钥/私钥对，其中公钥由目录代理服务器签名。自签名证书的有效期为三个月。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

- **要创建非默认的目录代理服务器自签名证书，请键入：**

  ```
  $ dpadm add-selfsign-cert instance-path cert-alias
  ```

  其中 *cert-alias* 是自签名证书的名称。

  例如，可以创建名为 `my-self-signed-cert` 的证书，如下所示：

  ```
  $ dpadm add-selfsign-cert /local/dps my-self-signed-cert
  ```

  有关所有命令选项的描述，请参见 `dpadm`(1M) 手册页，或者在命令行中键入 `dpadm add-selfsign-cert --help`。

▼ 请求目录代理服务器的 CA 签名证书

自签名证书对于测试非常有用。但是在生产环境中，使用可信证书颁发机构 (Certificate Authority, CA) 颁发的证书会更加安全。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1. **请求 CA 签名的服务器证书。**

   ```
   $ dpadm request-cert instance-path cert-alias
   ```

其中 *cert-alias* 是要请求的证书的名称。证书颁发机构可能需要该命令的所有选项以识别服务器。有关所有命令选项的描述，请参见 dpadm(1M) 手册页。

获取 CA 证书的过程取决于所使用的 CA。某些商业 CA 提供了允许您下载证书的 Web 站点。其他 CA 会以电子邮件的方式将证书发送给您。

例如，可以请求一个名为 my-CA-signed-cert 的证书，如下所示：

```
$ dpadm request-cert -S cn=my-request,o=test /local/dps my-CA-signed-cert
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBYDCBygIBADAhMQ0wCwYDVQQDEwRnZXJpMRAwDgYDVQQDEwdteWNlcnQ0MIGfMA0GCSqGSIb3
DQEBAQUAA4GNADCBiQKBgQC3v9ubG468wnjBDAMbRrEkmFDTQzT+LO30D/ALLXOiElVsHrtRyWhJ
PG9cURI9uwqs15crxCpJvho1kt3SB9+yMB8Ql+CKnCQDHlNAfnn30MjFHShv/sAuEygFsN+Ekci5
W1jySYE2rzE0qKVxWLSILFo1UFRVRsUnORTX/Nas7QIDAQABoAAwDQYJKoZIhvcNAQEEBQADgYEA
fcQMnZNLpPobiX1xy1ROefPOhksVz8didY8Q2fjjaHG5lajMsqOROzubsuQ9Xh4ohT8kIA6xcBNZ
g8FRNIRAHCtDXKOdOm3CpJ8da+YGI/ttSawIeNAKU1DApF9zMb7c2lS4yEfWmreoQdXIC9YeKtF6
zwbn2EmIpjHzETtS5Nk=
-----END NEW CERTIFICATE REQUEST-----
```

使用 dpadm request-cert 命令请求证书时，此证书请求是保密性增强的电子邮件 (Privacy Enhanced Mail, PEM) 格式的 PKCS #10 证书请求。 PEM 是由 RFC 1421 至 1424 指定的格式。有关详细信息，请参见 http://www.ietf.org/rfc/rfc1421.txt。PEM 格式表示 ASCII 格式的 base64 编码的证书请求。

请求 CA 签名的证书时，将创建一个临时的自签名证书。收到并安装来自 CA 的 CA 签名证书时，新证书将取代临时的自签名证书。

2 按照程序将证书请求发送给 CA。

发送请求之后，必须等待 CA 对您的证书做出响应。请求的响应时间会有所不同。例如，如果 CA 在您的公司内部，则响应时间可能很短。但是，如果 CA 在公司外部，则 CA 可能需要几个星期才能响应您的请求。

3 保存从 CA 收到的证书。

以文本文件的形式保存证书，并将此证书备份到安全位置。

▼ 安装目录代理服务器的 CA 签名服务器证书

要信任 CA 签名的服务器证书，必须在目录代理服务器实例上安装此证书。此过程将 CA 证书的公钥安装到目录代理服务器上的证书数据库中。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 查看是否已安装此 CA 的可信 CA 证书。

要执行此操作，请列出所有已安装的 CA 证书，如第 313 页中的 “列出 CA 证书”中所述。

2 **如果未安装可信 CA 证书，请将其添加到目录代理服务器实例上的证书数据库中。**

```
$ dpadm add-cert instance-path cert-alias cert-file
```

其中 *cert-alias* 是可信 CA 证书的名称，*cert-file* 是包含可信 CA 证书的文件的名称。

3 **将 CA 签名的服务器证书安装到证书数据库中。**

```
$ dpadm add-cert instance-path cert-alias cert-file
```

其中 *cert-alias* 是 CA 签名服务器证书的名称，*cert-file* 是包含 CA 签名服务器证书的文件的名称。请注意，*cert-alias* 必须与证书请求中所使用的 *cert-alias* 相同。

例如，可以将名为 `CA-cert` 的 CA 签名服务器证书添加到 `/local/dps` 上的证书数据库中，如下所示：

```
$ dpadm add-cert /local/dps CA-cert /local/safeplace/ca-cert-file.ascii
```

续订过期的目录代理服务器 CA 签名证书

本部分介绍如何续订过期的 CA 签名服务器证书。

▼ 续订过期的目录代理服务器 CA 签名服务器证书

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1 **从 CA 获取更新的证书。**

2 **在目录代理服务器实例上安装证书。**

```
$ dpadm renew-cert instance-path cert-alias cert-file
```

其中 *cert-alias* 是新证书的名称，*cert-file* 是包含此证书的文件的名称。有关所有命令选项的描述，请参见 `dpadm`(1M) 手册页。

列出证书

有关如何列出服务器证书和 CA 证书的信息，请参见以下过程。

▼ 列出服务器证书

此过程将列出在目录代理服务器实例上安装的所有证书。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

- **列出目录代理服务器实例上证书数据库中的服务器证书。**

  ```
  $ dpadm list-certs instance-path
  ```

  默认情况下，目录代理服务器实例包含一个名为 `defaultservercert` 的服务器证书。文本 `Same as issuer` 表明默认证书是自签名的服务器证书。

  例如：

```
$ dpadm list-certs /local/dps
Alias             Valid from       Expires on       Self-signed? Issued by          Issued to
----------------- ---------------- ---------------- ------------ ------------------ --------------
defaultservercert 2006/06/01 04:15 2008/05/31 04:15 y            CN=myserver:myport Same as issuer
1 certificate found.
```

▼ 列出 CA 证书

此过程将列出在目录代理服务器实例上安装的 CA 证书。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **列出目录代理服务器实例上证书数据库中的 CA 证书。**

  ```
  $ dpadm list-certs -C instance-path
  ```

  例如：

```
$ dpadm list-certs -C /local/dps
Alias   Valid from       Expires on        Built-in  Issued by    Issued to
------  ----------       ----------------  --------- ---------    ---------
CAcert1 1999/06/21 06:00 2020/06/21 06:00  y         CN=company1, O=company2
...
```

将后端 LDAP 服务器的证书添加到目录代理服务器上的证书数据库中

本部分介绍如何将后端 LDAP 服务器的证书添加到目录代理服务器上的证书数据库中。

▼ 将后端目录服务器的证书添加到目录代理服务器上的证书数据库中

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1. **使用以下命令语法以 PEM 格式显示后端目录服务器中的证书。**

   ```
   dsadm show-cert -F ascii instance-path [cert-alias]
   ```

   如果未指定 *cert-alias*，将显示默认的服务器证书。有关所有命令选项的描述，请参见 `dsadm`(1M) 手册页。

   例如，显示默认的自签名服务器证书，如下所示：

   ```
   $ dsadm show-cert -F ascii /local/ds defaultCert
   -----BEGIN CERTIFICATE-----
   MIICJjCCAY+gAwIBAgIFAIKL36kwDQYJKoZIhvcNAQEEBQAwVzEZMBcGA1UEChMQ
   U3VuIE1pY3Jvc3lzdGVtczEZMBcGA1UEAxMQRGlyZWN0b3J5IFNlcnZlcjENMAsG
   A1UEAxMEMjAxMTEQMA4GA1UEAxMHY29uZHlsZTAeFw0wNjA1MjIxMTQxNTVaFw0w
   NjA4MjIxMTQxNTVaMFcxGTAXBgNVBAoTEFN1biBNaWNyb3N5c3RlbXMxGTAXBgNV
   BAMTEERpcmVjdG9yeSBTZXJ2ZXIxDTALBgNVBAMTBDIwMTExEDAOBgNVBAMTB2Nv
   bmR5bGUwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAK9U3ry3sJmEzwQY8CGd
   7S2MTZuBedo03Vea1lfDtD08WIsdDMzhHplTdeHAkWWNc8g2PDcEFXeWp9UXFMuD
   Pcia7t8HtFkm73VmlriWhMd8nn3l2vkxhsPK2LHFEeOIUDR9LBBiMiEeLkjdoEhE
   VLMSoYKqKI+Aa5grINdmtFzBAgMBAAEwDQYJKoZIhvcNAQEEBQADgYEAF4eDbSd7
   qy2l10dIogT+rnXZ362gLTlQFCblhbGpmmptbegUdL1ITGv/62q1isPV2rW7CkjM
   Cqb0fo3k5UkKKvW+JbMowpQeAPnlgpX612HuDr1tldnKV4eyU7gpG31t/cpACALQ
   7OPi1A7oVb2Z8OJKfEJHkp3txBSsiI2gTkk=
   -----END CERTIFICATE-----
   ```

2. **保存证书。**

   以文本文件的形式保存证书，并将证书备份到安全位置。

3. **将后端 LDAP 服务器的证书添加到目录代理服务器实例上的证书数据库中。**

   ```
   $ dpadm add-cert instance-path cert-alias cert-file
   ```

   其中 *cert-alias* 是此证书的名称，*cert-file* 是包含此证书的文件的名称。

   例如，可以添加证书 `defaultCert`，如下所示：

   ```
   $ dpadm add-cert /local/dps defaultCert /local/safeplace/defaultCert.ascii
   ```

将证书导出到后端 LDAP 服务器

后端 LDAP 服务器可能会请求目录代理服务器中的证书。本部分介绍如何配置目录代理服务器，以便将证书导出到后端 LDAP 服务器。

▼ 配置目录代理服务器以便将客户端证书导出到后端 LDAP 服务器

1 **指定要发送到后端 LDAP 服务器的证书。**

```
$ dpconf set-server-prop -h host -p port ssl-client-cert-alias:cert-alias
```

其中 *cert-alias* 是此证书的名称。有关所有命令选项的描述，请参见 `dpconf`(1M) 手册页。

2 **将证书内容复制到某个文件。**

```
$ dpadm show-cert -F ascii -o filename instance-path cert-alias
```

3 **将证书添加到后端 LDAP 服务器的证书数据库中，如第 105 页中的“添加 CA 签名的服务器证书和可信的 CA 证书”中所述。**

接下来的操作 将后端 LDAP 服务器配置为进行客户端验证。有关如何对目录代理服务器执行此操作的信息，请参见第 111 页中的“配置客户端验证”。

另请参见 有关如何配置客户端和目录代理服务器之间基于证书的验证的信息，请参见第 407 页中的“配置基于证书的验证”。

备份和恢复目录代理服务器的证书数据库

使用 `dpadm` 备份目录代理服务器时，将备份服务器证书。备份的证书存储在 *archive-path*`/alias` 目录中。

有关如何备份和恢复目录代理服务器的信息，请参见第 304 页中的“备份和恢复目录代理服务器实例”。

访问证书数据库时提示输入密码

默认情况下，证书数据库的密码是在内部进行管理的。因此，您无需键入证书密码或指定密码文件。通过存储的密码在内部管理证书数据库时，密码会存储在安全的环境中。

为了使证书更加安全，以及对证书进行更多控制，可以在命令行上将目录代理服务器配置为提示输入密码。这样，对于除 `autostart`、`backup`、`disable-service`、`enable-service`、`info`、`restore` 和 `stop` 之外的所有 `dpadm` 子命令，系统都会提示您输入密码。

有关将目录代理服务器配置为提示（或不提示）输入密码的信息，请参见以下过程。

▼ 访问证书数据库时提示输入密码

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1. **停止服务器。**

```
$ dpadm stop instance-path
Directory Proxy Server instance 'instance-path' stopped
```

2. **将密码提示标志设置为 on，然后键入并确认证书数据库密码。**

```
$ dpadm set-flags instance-path cert-pwd-prompt=on
Choose the certificate database password:
Confirm the certificate database password:
```

3. **启动服务器，然后键入证书数据库密码。**

```
$ dpadm start instance-path
Enter the certificate database password:
```

▼ 访问证书数据库时禁用密码提示

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1. **停止服务器。**

```
$ dpadm stop instance-path
Directory Proxy Server instance 'instance-path' stopped
```

2. **将密码提示标志设置为 off，然后键入现有密码。**

```
$ dpadm set-flags instance-path cert-pwd-prompt=off
Enter the old password:
```

3. **启动服务器。**

```
$ dpadm start instance-path
```

◆ ◆ ◆ 第 20 章

20 LDAP 数据源和数据源池

本章介绍如何使用 dpconf 命令创建和配置 LDAP 数据源和数据源池。有关这些主题的参考信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“LDAP Data Sources”。

本章包含以下主题：

创建和配置 LDAP 数据源

有关如何创建和配置 LDAP 数据源的信息，请参见以下过程。

▼ 创建 LDAP 数据源

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 创建数据源。

```
$ dpconf create-ldap-data-source -h host -p port source-name host:port
```

在此命令中，*source-name* 是为新数据源指定的名称。*host* 和 *port* 指的是正在运行 LDAP 服务器的主机和端口。请注意，在默认情况下数据源不使用 SSL。

如果主机由 IP V6 地址指定，则创建数据源时需要使用 IP V6 引用。例如，如果目录代理服务器要绑定到端口 2389 上 IP V6 地址为 fe80::209:3dff:fe00:8c93 的主机，请使用以下命令创建数据源：

```
$ dpconf create-ldap-data-source -h host1 -p 1389 ipv6-host \
 [fe80::209:3dff:fe00:8c93]:2389
```

如果使用控制台创建数据源，则必须指定实际的 IP V6 地址（不包含方括号）。

有关如何修改 LDAP 数据源属性的信息，请参见第 318 页中的 “配置 LDAP 数据源”。

2 （可选的）查看数据源列表。

```
$ dpconf list-ldap-data-sources -h host -p port
```

▼ 配置 LDAP 数据源

此过程将配置目录代理服务器和 LDAP 数据源之间的授权。此外，还将配置目录代理服务器监视 LDAP 数据源的方式。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 通过使用以下命令语法查看数据源的属性：

```
dpconf get-ldap-data-source-prop -h host -p port [-M unit] [-Z unit] source-name [property...]
```

在此命令中，`-M` 和 `-Z` 指的是显示数据的单位。`M` 选项指定时间单位。`-M` 的值可以为 `M`、`w`、`d`、`h`、`m`、`s` 或 `ms`，分别表示月、周、天、小时、分钟、秒或毫秒。`-Z` 选项指定数据大小单位。`-Z` 的值可以为 `T`、`G`、`M`、`k` 或 `b`，分别表示千吉字节、千兆字节、兆字节、千字节或字节。

如果不指定属性，将显示所有属性。LDAP 数据源的默认属性如下所示：

```
bind-dn                       :  -
bind-pwd                      :  -
client-cred-mode              :  use-client-identity
connect-timeout               :  10s
description                   :  -
is-enabled                    :  false
is-read-only                  :  true
ldap-address                  :  host
ldap-port                     :  port
ldaps-port                    :  ldaps
monitoring-bind-timeout       :  5s
monitoring-entry-dn           :  ""
monitoring-entry-timeout      :  5s
monitoring-inactivity-timeout :  2m
monitoring-interval           :  30s
monitoring-mode               :  proactive
monitoring-search-filter      :  (|(objectClass=*)(objectClass=ldapSubEntry))
num-bind-incr                 :  10
num-bind-init                 :  10
num-bind-limit                :  1024
num-read-incr                 :  10
```

```
num-read-init                  :  10
num-read-limit                 :  1024
num-write-incr                 :  10
num-write-init                 :  10
num-write-limit                :  1024
proxied-auth-check-timeout     :  1.8s
proxied-auth-use-v1            :  false
ssl-policy                     :  never
use-tcp-no-delay               :  true
```

2 **启用数据源。**

```
$ dpconf set-ldap-data-source-prop -h host -p port source-name is-enabled:true
```

3 **如果要更改默认设置，请配置步骤 1 中列出的所有属性。**

```
$ dpconf set-ldap-data-source-prop -h host -p port source-name property:value
```

例如，如果要修改数据源上的条目，请将此数据源配置为允许写入操作。

```
$ dpconf set-ldap-data-source-prop -h host -p port source-name is-read-only:false
```

要查找有关子命令中所使用的属性的信息，请运行以下命令：

```
$ dpconf help-properties ldap-data-source property
```

要列出数据源的主要属性，请在 `list` 子命令中使用详细选项 `-v`。

```
$ dpconf list-ldap-data-sources -v
Name         is-enabled  ldap-address  ldap-port  ldaps-port  description
-----------  ----------  ------------  ---------  ----------  -----------
datasource0  true        myHost        myPort     ldaps       -
datasource1  true        myHost        myPort     ldaps       -
```

4 **重新启动目录代理服务器实例以使更改生效（如有必要）。**

有关重新启动目录代理服务器的信息，请参见第 302 页中的 “重新启动目录代理服务器”。有关需要重新启动服务器的配置更改的列表，请参见第 306 页中的 “需要重新启动服务器的配置更改”。

创建和配置 LDAP 数据源池

有关如何创建和配置数据源池的信息，请参见以下过程：

▼ 创建 LDAP 数据源池

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 **创建一个或多个数据源池。**

```
$ dpconf create-ldap-data-source-pool -h host -p port pool-name
```

可以在第一个 *pool-name* 之后指定其他数据源池。有关如何修改数据源池属性的信息，请参见第 320 页中的 “配置 LDAP 数据源池” 。

2 **（可选的）查看数据源池列表。**

```
$ dpconf list-ldap-data-source-pools -h host -p port
```

▼ 配置 LDAP 数据源池

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 **通过使用以下命令语法查看数据源池的属性：**

```
dpconf get-ldap-data-source-pool-prop -h host -p port [-M unit] [-Z unit] \
 pool-name [property...]
```

在此命令中，`-M` 和 `-Z` 指的是显示数据的单位。`M` 选项指定时间单位。`-M` 的值可以为 `M`、`w`、`d`、`h`、`m`、`s` 或 `ms`，分别表示月、周、天、小时、分钟、秒或毫秒。`-Z` 选项指定数据大小单位。`-Z` 的值可以为 `T`、`G`、`M`、`k` 或 `b`，分别表示千吉字节、千兆字节、兆字节、千字节或字节。

如果不指定属性，将显示所有属性。LDAP 数据源池的默认属性如下所示：

```
client-affinity-policy    :  write-affinity-after-write
client-affinity-timeout   :  20s
description               :  -
enable-client-affinity    :  false
load-balancing-algorithm  :  proportional
```

2 **配置步骤 1 中列出的属性。**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 property:value
```

有关如何为负载平衡和客户端相似性配置数据源池属性的信息，请参见第 22 章。

将 LDAP 数据源连接到数据源池

连接到数据源池的数据源称为**连接数据源**。连接数据源的属性确定数据源池的负载平衡配置。配置连接数据源的权重时，应考虑数据源池中所有连接数据源的权重。请确保这些权重能够根据需要协同工作。有关如何为负载平衡配置权重的信息，请参见第 334 页中的 “配置负载平衡的权重” 。

▼ 将 LDAP 数据源连接到数据源池

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 将一个或多个数据源连接到数据源池。

```
$ dpconf attach-ldap-data-source -h host -p port pool-name \
 source-name [source-name ...]
```

2 （可选的）查看给定数据源池中连接数据源的列表。

```
$ dpconf list-attached-ldap-data-sources -h host -p port -E pool-name
```

在此命令中，`-E` 是可选选项，用于修改显示输出，以便每行显示一个属性值。

3 （可选的）查看给定数据源池中连接数据源的主要属性。

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
```

在此命令中，`-v` 指定详细输出。例如，查看示例数据源池的属性。

```
$ dpconf list-attached-ldap-data-sources -h host1 -p 1389 -v My-pool
Name         add-weight  bind-weight  compare-weight
-----------  ----------  -----------  --------------
datasource0  disabled    disabled     disabled
datasource1  disabled    disabled     disabled

delete-weight  modify-dn-weight  modify-weight  search-weight
-------------  ----------------  -------------  -------------
disabled       disabled          disabled       disabled
disabled       disabled          disabled       disabled
```

4 （可选的）通过使用以下命令语法查看连接数据源的属性：

```
$ dpconf get-attached-ldap-data-source-prop -h host -p port [-M unit] [-Z unit] \
 pool-name source-name [property...]
```

在此命令中，`-M` 和 `-Z` 指的是显示数据的单位。`M` 选项指定时间单位。`-M` 的值可以为 `M`、`w`、`d`、`h`、`m`、`s` 或 `ms`，分别表示月、周、天、小时、分钟、秒或毫秒。`-Z` 选项指定数据大小单位。`-Z` 的值可以为 `T`、`G`、`M`、`k` 或 `b`，分别表示千吉字节、千兆字节、兆字节、千字节或字节。

如果不指定属性，将显示所有属性。

连接数据源的属性可定义负载平衡中每种操作类型的权重。连接数据源的默认权重如下所示：

```
add-weight        :  disabled
bind-weight       :  disabled
compare-weight    :  disabled
```

```
delete-weight     :  disabled
modify-dn-weight  :  disabled
modify-weight     :  disabled
search-weight     :  disabled
```

有关如何为负载平衡配置连接数据源权重的信息，请参见第 334 页中的 “配置负载平衡的权重” 。

◆ ◆ ◆ 第 21 章

21 目录代理服务器和后端 LDAP 服务器之间的连接

本章介绍如何配置目录代理服务器和后端 LDAP 服务器之间的连接。本章包含以下主题：

配置目录代理服务器和后端 LDAP 服务器之间的连接

有关如何配置目录代理服务器和后端 LDAP 服务器之间的连接的信息，请参见以下过程：

▼ 配置目录代理服务器和后端 LDAP 服务器之间的连接数

注 – 此过程将配置绑定操作的连接数。要配置读取或写入操作的连接数，请执行相同的过程，但要将 `bind` 替换为 `read` 或 `write`。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 **配置目录代理服务器和后端 LDAP 服务器之间用于绑定操作的初始连接数。**

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 num-bind-init:new-value
```

2 **配置绑定操作的连接数增量。**

增量是每次请求超出当前连接数的连接时所增加的连接数。

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 num-bind-incr:new-value
```

3 **配置绑定操作的最大连接数。**

达到最大连接数时，将无法再添加更多的连接。

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 num-bind-limit:new-value
```

▼ 配置连接超时

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

- **配置目录代理服务器可以尝试连接到数据源的最长时间。**

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 connect-timeout:new-value
```

例如，将连接超时时间配置为 10 毫秒。

```
$ dpconf set-ldap-data-source-prop -h host1 -p 1389 data-source-name connect-timeout:10
```

▼ 配置连接池等待超时

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

- **配置目录代理服务器可以等待连接池中已建立的连接变为可用的最长时间。**

```
$ dpconf set-server-prop -h host -p port data-source-name \
 connection-pool-wait-timeout:value
```

例如，将超时时间配置为 20 秒。

```
$ dpconf set-ldap-data-source-prop -h host1 -p 1389 data-source-name \
 connection-pool-wait-timeout:20000
```

配置目录代理服务器和后端 LDAP 服务器之间的 SSL

以下过程介绍如何配置目录代理服务器和后端 LDAP 服务器之间的 SSL。

▼ 配置目录代理服务器和后端 LDAP 服务器之间的 SSL

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1. **配置目录代理服务器和后端 LDAP 服务器之间的安全端口。**

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
    ldaps-port:port-number
   ```

2. **配置何时将 SSL 用于目录代理服务器和后端 LDAP 服务器之间的连接。**

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port data-source-name ssl-policy:value
   ```

   - 如果 *value* 为 `always`，则始终将 SSL 用于连接。
   - 如果 *value* 为 `client`，则会在客户端使用 SSL 时使用 SSL。

   如果连接未使用 SSL，则可以使用 `startTLS` 命令将此连接升级为 SSL。

3. **选择 SSL 的协议和密码，如第 326 页中的 “为目录代理服务器选择 SSL 密码和 SSL 协议”中所述。**

4. **将目录代理服务器配置为验证来自后端 LDAP 服务器的 SSL 服务器证书。**

   有关信息，请参见第 313 页中的 “将后端目录服务器的证书添加到目录代理服务器上的证书数据库中”。

5. **如果后端 LDAP 服务器请求来自目录代理服务器的证书，请将目录代理服务器配置为发送 SSL 客户端证书。**

   有关信息，请参见第 314 页中的 “将证书导出到后端 LDAP 服务器”。

6. **重新启动目录代理服务器实例以使更改生效。**

   有关重新启动目录代理服务器的信息，请参见第 302 页中的 “重新启动目录代理服务器”。

为目录代理服务器选择 SSL 密码和 SSL 协议

目录代理服务器可以使用的密码和协议取决于正在使用的 Java™ 虚拟机 (Java Virtual Machine, JVM™)。默认情况下，目录代理服务器使用为 JVM 计算机启用的默认密码和协议。

▼ 选择密码和协议的列表

可使用此过程检索受支持的密码和协议，以及启用的密码和协议。如果密码或协议受支持，则可启用或禁用该密码或协议。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 查看受支持的密码和协议的列表。

```
$ dpconf get-server-prop -h host -p port supported-ssl-cipher-suites \
 supported-ssl-protocols
```

2 查看已启用的密码和协议的列表。

```
$ dpconf get-server-prop -h host -p port enabled-ssl-cipher-suites \
 enabled-ssl-protocols
```

3 启用一个或多个受支持的密码或协议。

a. 启用一个或多个受支持的密码。

```
$ dpconf set-server-prop -h host -p port \
 enabled-ssl-cipher-suites:supported-ssl-cipher-suite \
 [enabled-ssl-cipher-suites:supported-ssl-cipher-suite ...]
```

b. 启用一个或多个受支持的协议。

```
$ dpconf set-server-prop -h host -p port \
 enabled-ssl-cipher-protocols:supported-ssl-cipher-protocol \
 [enabled-ssl-cipher-protocols:supported-ssl-cipher-protocol ...]
```

4 要禁用受支持的密码或协议，请使用前两个步骤中的命令。

指定密码或协议的完整列表，并排除要禁用的密码或协议。

将请求转发到后端 LDAP 服务器

本部分包含用于将请求从目录代理服务器转发到后端 LDAP 服务器的各种方法的相关信息。

使用绑定重放转发请求

有关目录代理服务器中客户端证书绑定重放的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Directory Proxy Server Configured for BIND Replay”。以下过程介绍如何使用绑定重放将请求从目录代理服务器转发到后端 LDAP 服务器。

▼ 使用绑定重放转发请求

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **配置数据源客户端证书，以便使用客户端提供的证书通过后端 LDAP 服务器的验证。**

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 client-cred-mode:use-client-identity
```

使用代理授权转发请求

有关目录代理服务器中代理授权的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Directory Proxy Server Configured for Proxy Authorization”。

本部分包含的过程将使用代理授权和代理授权控件来转发请求。

▼ 使用代理授权转发请求

1. **将数据源配置为接受版本 1 或版本 2 的代理授权控件。**

   例如，将数据源配置为接受版本 1 的代理授权控件。

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
    proxied-auth-use-v1:true
   ```

   或者，将数据源配置为接受版本 2 的代理授权控件。

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
    proxied-auth-use-v1:false
   ```

2. **将数据源配置为使用代理授权通过到后端 LDAP 服务器的验证。**

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
    client-cred-mode:use-proxy-auth
   ```

要将数据源配置为使用只有写入操作权限的代理授权来通过后端 LDAP 服务器的验证，请运行以下命令：

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 client-cred-mode:use-proxy-auth-for-write
```

使用代理授权控制执行只写操作时，客户端标识不会转发到 LDAP 服务器以用于读取请求。有关转发无客户端标识的请求的详细信息，请参见第 328 页中的 “转发无客户端标识的请求”。

3 **使用目录代理服务器的绑定证书配置数据源。**

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 bind-dn:DPS-bind-dn bind-pwd-file:filename
```

4 **将数据源配置为启用超时。**

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 proxied-auth-check-timeout:value
```

目录代理服务器使用 `getEffectiveRights` 命令验证客户端 DN 是否具有代理授权的相关 ACI。结果将缓存在目录代理服务器中，并在 `proxied-auth-check-timeout` 过期时进行更新。

5 **重新启动目录代理服务器实例以使更改生效（如有必要）。**

有关重新启动目录代理服务器的信息，请参见第 302 页中的 “重新启动目录代理服务器”。

▼ 当请求包含代理授权控件时使用代理授权转发请求

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **将目录代理服务器配置为接受版本 1 和/或版本 2 的代理授权控件。**

```
$ dpconf set-server-prop -h host -p port allowed-ldap-controls:proxy-auth-v1 \
 allowed-ldap-controls:proxy-auth-v2
```

转发无客户端标识的请求

以下过程介绍如何将请求从目录代理服务器转发到后端 LDAP 服务器，而不转发客户端标识。

▼ 转发无客户端标识的请求

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1. **将数据源配置为使用目录代理服务器证书通过后端 LDAP 服务器的验证。**

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 client-cred-mode:use-specific-identity
```

2. **使用目录代理服务器的绑定证书配置数据源。**

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 bind-dn:bind-dn-of-DPS bind-pwd-file:filename
```

3. **重新启动目录代理服务器实例以使更改生效（如有必要）。**

   有关重新启动目录代理服务器的信息，请参见第 302 页中的 “重新启动目录代理服务器”。

以备用用户身份转发请求

本部分包含有关如何以备用用户身份转发请求的信息。

▼ 配置远程用户映射

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1. **启用要以备用用户身份转发的操作。**

```
$ dpconf set-server-prop -h host -p port enable-user-mapping:true
```

2. **指定包含远程映射 ID 的属性名称。**

```
$ dpconf set-server-prop -h host -p port \
 remote-user-mapping-bind-dn-attr:attribute-name
```

3. **将目录代理服务器配置为远程映射客户端 ID。**

```
$ dpconf set-server-prop -h host -p port enable-remote-user-mapping:true
```

4. **配置默认映射。**

```
$ dpconf set-server-prop -h host -p port \
 user-mapping-default-bind-dn:default-mapping-bind-dn \
 user-mapping-default-bind-pwd-file:filename
```

   如果在远程 LDAP 服务器上找不到映射的标识，则客户端标识将映射到默认标识。

5. **在远程 LDAP 服务器上的客户端条目中配置用户映射。**

   有关在目录服务器中配置用户映射的信息，请参见第 144 页中的 “代理授权”。

▼ 配置本地用户映射

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1. **启用要以备用用户身份转发的操作。**

```
$ dpconf set-server-prop -h host -p port enable-user-mapping:true
```

2. **确保未将目录代理服务器配置为远程映射客户端 ID。**

```
$ dpconf set-server-prop -h host -p port enable-remote-user-mapping:false
```

3. **配置默认映射。**

```
$ dpconf set-server-prop -h host -p port \
 user-mapping-default-bind-dn:default-mapping-bind-dn \
 user-mapping-default-bind-pwd-file:filename
```

如果远程 LDAP 服务器上的映射失败，客户端 ID 将映射到此 DN。

4. **如果允许未经验证的用户执行操作，请为未经验证的客户端配置映射。**

```
$ dpconf set-server-prop -h host -p port \
 user-mapping-anonymous-bind-dn:anonymous-mapping-bind-dn \
 user-mapping-anonymous-bind-pwd-file:filename
```

有关如何允许未经验证的用户执行操作的信息，请参见第 407 页中的 “配置匿名访问”。

5. **配置客户端 ID。**

```
$ dpconf set-user-mapping-prop -h host -p port \
 user-bind-dn:client-bind-dn user-bind-pwd-file:filename
```

6. **配置备用用户 ID。**

```
$ dpconf set-user-mapping-prop -h host -p port \
 mapped-bind-dn:alt-user-bind-dn mapped-bind-pwd-file:filename
```

▼ 为匿名客户端配置用户映射

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **为未经验证的客户端配置映射。**

```
$ dpconf set-server-prop -h host -p port \
 user-mapping-anonymous-bind-dn:anonymous-mapping-bind-dn \
 user-mapping-anonymous-bind-pwd-file:filename
```

将在目录代理服务器中配置匿名客户端映射，因为远程 LDAP 服务器中不包含匿名客户端条目。

有关允许未经验证的用户执行操作的信息，请参见第 407 页中的 “配置匿名访问”。

◆ ◆ ◆ 第 22 章

22 目录代理服务器负载平衡和客户端相似性

有关负载平衡和客户端相似性的描述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的第 25 章 “Directory Proxy Server Load Balancing and Client Affinity”。本章包含以下主题：

配置负载平衡

有关负载平衡的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Load Balancing”。本部分介绍如何配置负载平衡，并提供了样例配置。

▼ 选择负载平衡算法

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1 **通过查看 LDAP 数据源池的属性获取当前的负载平衡算法。**

```
$ dpconf get-ldap-data-source-pool-prop -h host -p port pool-name
```

LDAP 数据源池的默认属性如下所示：

```
client-affinity-policy    :  write-affinity-after-write
client-affinity-timeout   :  20s
description               :  -
enable-client-affinity    :  false
load-balancing-algorithm  :  proportional
```

默认情况下，负载平衡算法为 `proportional`（**比例**）。

2 **将 LDAP 数据源池配置为使用某种算法。**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
  load-balancing-algorithm:selected-algorithm
```

其中 *selected-algorithm* 可为以下任一选项：

- **故障转移**
- **操作相似性**
- **比例**
- **饱和度**

有关算法的详细信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Introduction to Load Balancing”。

3 **重新启动目录代理服务器实例。**

```
$ dpadm restart instance-path
```

▼ 配置负载平衡的权重

您需要配置连接数据源的权重，并应参照数据源池中所有其他连接数据源的权重进行配置。请考虑所有连接数据源的权重。对于某一类型的操作，如果数据源的权重为**已禁用**，则始终不会将此类型的请求发送到该数据源。如果数据源的权重为 **0**（零），则不会向该数据源发送请求，除非所有其他的数据源都不可用。因此，只有在所有其他数据源都不可用时，才会使用权重配置为 **0** 的数据源。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 **查看已连接到数据源池的数据源的列表。**

```
$ dpconf list-attached-ldap-data-sources -h host -p port pool-name
```

2 **查看某个连接数据源的属性。**

```
$ dpconf get-attached-ldap-data-source-prop pool-name \
 attached-data-source-name
```

连接数据源的属性可定义每种操作类型的权重。连接数据源的默认权重如下所示：

```
add-weight        :  disabled
bind-weight       :  disabled
compare-weight    :  disabled
delete-weight     :  disabled
modify-dn-weight  :  disabled
modify-weight     :  disabled
search-weight     :  disabled
```

3. **配置其中一个连接数据源的权重。**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name \
 attached-data-source-name add-weight:value \
 bind-weight:value compare-weight:value delete-weight:value \
 modify-dn-weight:value modify-weight:value search-weight:value
```

4. **对其他连接数据源重复步骤 2 和步骤 3。**

5. **比较连接数据源的主要参数。**

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
```

例如，数据源池可以包含具有以下权重的数据源：

```
$ dpconf list-attached-ldap-data-sources -h host1 -p 1389 -v myPool
Name add-weight bind-weight compare-weight delete-weight modify-dn-weight modify-weight search-weight
---- ---------- ----------- -------------- ------------- ---------------- ------------- -------------
DS-1 disabled   3               disabled       disabled      disabled         disabled      disabled
DS-2 2          2           2              2             2                2             2
DS-3 1          1           1              1             1                1             1
```

负载平衡的示例配置

本部分包含配置每种负载平衡算法的样例过程。

▼ 配置负载平衡的比例算法

有关比例算法的描述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Proportional Algorithm for Load Balancing”。

在此示例中，为数据源 *ds–1* 配置的权重是其他两个数据源权重的两倍。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

开始之前 请确保您的数据源池至少包含三个连接数据源。有关如何创建数据源和数据源池的信息，请参见第 20 章。

1. **将数据源池配置为使用负载平衡的比例算法。**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
  load-balancing-algorithm:proportional
```

2. **配置第一个数据源的属性。**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-1 \
 add-weight:2 bind-weight:2 compare-weight:2 delete-weight:2 modify-dn-weight:2 \
 modify-weight:2 search-weight:2
```

3 配置第二个数据源的属性。

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-2 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

4 配置第三个数据源的属性。

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-3 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

5 比较连接数据源的主要参数。

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
Name add-weight bind-weight compare-weight delete-weight modify-dn-weight modify-weight search-weight
---  ---------- ----------- -------------- ------------- ---------------- ------------- -------------
ds-1 2          2           2              2             2                2             2
ds-2 1          1           1              1             1                1             1
ds-3 1          1           1              1             1                1             1
```

6 重新启动目录代理服务器实例。

```
$ dpadm restart instance-path
```

▼ 配置负载平衡的饱和度算法

有关饱和度算法的描述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Saturation Algorithm for Load Balancing”。

在此示例中，数据源 `ds-1` 执行大多数绑定操作，但不执行任何其他类型的操作。对三个数据源的权重进行如下配置：

- 将 `ds-1` 的权重配置为 3（对于绑定操作）和“已禁用”（对于所有其他类型的操作）。
- 将 `ds-2` 的权重配置为 2（对于所有操作）。
- 将 `ds-3` 的权重配置为 1（对于所有操作）。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

开始之前 请确保您的数据源池至少包含三个连接数据源。有关如何创建数据源和数据源池的信息，请参见第 20 章。

1 将数据源池配置为使用负载平衡的饱和度算法。

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 load-balancing-algorithm:saturation
```

2. **配置第一个数据源的属性。**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-1 \
 add-weight:disabled bind-weight:3 compare-weight:disabled delete-weight:disabled \
 modify-dn-weight:disabled modify-weight:disabled search-weight:disabled
```

3. **配置第二个数据源的属性。**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-2 \
 add-weight:2 bind-weight:2 compare-weight:2 delete-weight:2 modify-dn-weight:2 \
 modify-weight:2 search-weight:2
```

4. **配置第三个数据源的属性。**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-3 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

5. **比较连接数据源的主要参数。**

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
Name add-weight bind-weight compare-weight delete-weight modify-dn-weight modify-weight search-weight
---  ---------- ----------- -------------- ------------- ---------------- ------------- -------------
ds-1 disabled   3               disabled       disabled      disabled         disabled      disabled
ds-2 2          2           2              2             2                2             2
ds-3 1          1           1              1             1                1             1
```

6. **重新启动目录代理服务器实例。**

```
$ dpadm restart instance-path
```

▼ 为全局帐户锁定配置操作相似性算法

有关此算法的描述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Operational Affinity Algorithm for Global Account Lockout”。

此示例具有三个数据源。数据源 `ds-1` 被配置为接收所有绑定请求。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

开始之前 请确保您的数据源池至少包含三个连接数据源。有关如何创建数据源和数据源池的信息，请参见第 20 章。

1. **将数据源池配置为使用操作相似性算法。**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 load-balancing-algorithm:operational-affinity
```

2 **配置第一个数据源的属性。**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-1 \
 add-weight:1 bind-weight:100 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

3 **配置第二个数据源的属性。**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-2 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

4 **配置第三个数据源的属性。**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-3 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

5 **比较连接数据源的主要参数。**

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
Name add-weight bind-weight compare-weight delete-weight modify-dn-weight modify-weight search-weight
--   ---------- ----------- -------------- ------------- ---------------- ------------- -------------
ds-1 1          1           1              1             1                1             1
ds-2 1          100         1              1             1                1             1
ds-3 1          1           1              1             1                1             1
```

6 **重新启动目录代理服务器实例。**

```
$ dpadm restart instance-path
```

▼ 为缓存优化配置操作相似性算法

有关此算法的描述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Operational Affinity Algorithm for Cache Optimization”。

此示例具有三个数据源。数据源 `ds-1` 处理所有搜索和比较操作。当 `ds-1` 响应请求时，目标条目将被存储到缓存中。如果 `ds-1` 重复响应同一请求，数据源可以使用缓存的数据。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面 ”和 DSCC 联机帮助。

开始之前 请确保您的数据源池至少包含三个连接数据源。有关如何创建数据源和数据源池的信息，请参见第 20 章。

1 **将数据源池配置为使用操作相似性算法。**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 load-balancing-algorithm:operational-affinity
```

2. **配置第一个数据源的属性。**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-1 \
 add-weight:1 bind-weight:1 compare-weight:100 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:100
```

3. **配置第二个数据源的属性。**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-2 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

4. **配置第三个数据源的属性。**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-3 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

5. **比较连接数据源的主要参数。**

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
Name add-weight bind-weight compare-weight delete-weight modify-dn-weight modify-weight search-weight
---  ---------- ----------- -------------- ------------- ---------------- ------------- -------------
ds-1 1          1           100            1             1                1             100
ds-2 1          1           1              1             1                1             1
ds-3 1          1           1              1             1                1             1
```

6. **重新启动目录代理服务器实例。**

```
$ dpadm restart instance-path
```

▼ 配置负载平衡的故障转移算法

有关故障转移算法的描述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Failover Algorithm for Load Balancing”。

此示例具有三个数据源。数据源 `ds-1` 接收所有请求。如果 `ds-1` 出现故障，`ds-2` 将接收所有请求，直到 `ds-1` 恢复为止。如果 `ds-2` 在 `ds-1` 恢复之前出现故障，`ds-3` 将接收所有请求。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

开始之前 请确保您的数据源池至少包含三个连接数据源。有关如何创建数据源和数据源池的信息，请参见第 20 章。

1. **将数据源池配置为使用负载平衡的故障转移算法。**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 load-balancing-algorithm:failover
```

2. **配置第一个数据源的属性。**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-1 \
 add-weight:3 bind-weight:3 compare-weight:3 delete-weight:3 modify-dn-weight:3 \
 modify-weight:3 search-weight:3
```

3. **配置第二个数据源的属性。**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-2 \
 add-weight:2 bind-weight:2 compare-weight:2 delete-weight:2 modify-dn-weight:2 \
 modify-weight:2 search-weight:2
```

4. **配置第三个数据源的属性。**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-3 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

5. **比较连接数据源的主要参数。**

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
Name add-weight bind-weight compare-weight delete-weight modify-dn-weight modify-weight search-weight
---  ---------- ----------- -------------- ------------- ---------------- ------------- -------------
ds-1 3          3           3              3             3                3             3
ds-2 2          2           2              2             2                2             2
ds-3 1          1           1              1             1                1             1
```

6. **重新启动目录代理服务器实例。**

```
$ dpadm restart instance-path
```

配置客户端相似性

客户端相似性可以降低负载平衡部署中的传播延迟风险。有关客户端相似性的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Client Affinity”。本部分介绍如何配置客户端连接和数据源之间的相似性，并提供了样例配置。

▼ 配置客户端相似性

此过程介绍如何配置客户端连接和数据源之间的相似性。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1. **通过查看数据源池属性来查看当前的负载平衡算法。**

```
$ dpconf get-ldap-data-source-pool-prop -h host -p port pool-name
```

数据源池的默认属性如下所示：

```
client-affinity-policy    :  write-affinity-after-write
client-affinity-timeout   :  20s
description               :  -
enable-client-affinity    :  false
load-balancing-algorithm  :  proportional
```

可使用以下参数配置客户端相似性：`client-affinity-policy` 、`client-affinity-timeout` 和 `enable-client-affinity`。要获取有关属性的描述及其有效值列表，请键入：

```
dpconf help-properties ldap-data-source-pool client-affinity-policy \
 client-affinity-timeout enable-client-affinity
```

有关属性的详细信息，请参见以下手册页：`client-affinity-policy`(5dpconf)、`client-affinity-timeout`(5dpconf) 和 `enable-client-affinity`(5dpconf)。

2 启用客户端相似性。

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 enable-client-affinity:true
```

3 选择客户端相似性的策略。

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 client-affinity-policy:selected-policy
```

其中 *selected-policy* 可为以下任一选项：

`write-affinity-after-write`
　第一个写入请求之后的写入请求的相似性

`read-write-affinity-after-write`
　第一个写入请求之后的所有请求的相似性

`read-write-affinity-after-any`
　第一个读取请求或写入请求之后的所有请求的相似性

`read-affinity-after-write`
　写入请求之后的第一个读取请求的相似性

4 配置客户端相似性的持续时间。

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 client-affinity-timeout:time-out[unit]
```

超时时间的默认**单位**为毫秒。

客户端相似性的示例配置

本部分包含与客户端相似性有关的示例配置， 并包含复制延迟、验证写入操作和基于连接的路由的示例。

▼ 配置当数据源池包含主服务器和使用方时复制延迟的客户端相似性

此过程将配置在第一个写入操作之后三秒内发生的所有读取操作和写入操作的客户端相似性。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

- **配置数据源池的相似性参数。**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 client-affinity-policy:read-write-affinity-after-write client-affinity-timeout:3000 \
 enable-client-affinity:true
```

▼ 配置客户端相似性以通过读取操作验证每个写入操作

此过程将配置每个写入操作之后的第一个读取操作的客户端相似性。此示例可用于特定应用程序，在该应用程序中，指定的绑定 DN 通过执行读取操作来验证每个写入操作。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

- **配置数据源池的相似性参数。**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 client-affinity-policy:read-affinity-after-write enable-client-affinity:true
```

▼ 为基于连接的路由配置客户端相似性

在低于 Directory Proxy Server 6.0 的版本中，客户端和 LDAP 服务器之间打开了一个连接。来自客户端的所有请求都使用同一连接，直到该连接关闭为止。此类型的路由称为**基于连接的路由**。此过程介绍如何为基于连接的路由配置客户端相似性。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

开始之前 请确保所有数据源都已连接到数据源池，并将 `clientCredentialsForwarding` 设置为 `useBind`。

- **配置数据源池的相似性参数。**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 client-affinity-policy:read-write-affinity-after-any enable-client-affinity:true
```

◆ ◆ ◆ 第 23 章

23 目录代理服务器数据视图

有关数据视图功能的概述，以及使用案例示例的描述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的第 22 章 “Directory Proxy Server LDAP Data Views”。

本章包含以下主题：

创建和配置 LDAP 数据视图

有关如何创建和配置 LDAP 数据视图的信息，请参见以下过程：

▼ 创建 LDAP 数据视图

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 创建 LDAP 数据视图。

```
$ dpconf create-ldap-data-view -h host -p port view-name pool-name suffix-DN
```

有关如何修改 LDAP 数据视图属性的信息，请参见第 344 页中的 “配置 LDAP 数据视图”。

2 查看 LDAP 数据视图的列表。

```
$ dpconf list-ldap-data-views -h host -p port
```

▼ 配置 LDAP 数据视图

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1 查看 LDAP 数据视图的属性。

```
$ dpconf get-ldap-data-view-prop -h host -p port view-name
```

如果在创建数据视图时未配置任何属性，则数据视图具有以下配置：

```
alternate-search-base-dn                    :  ""
alternate-search-base-dn                    :  base-DN
attr-name-mappings                          :  none
base-dn                                     :  suffix-DN
contains-shared-entries                     :  -
description                                 :  -
distribution-algorithm                      :  -
dn-join-rule                                :  -
dn-mapping-attrs                            :  none
dn-mapping-source-base-dn                   :  none
excluded-subtrees                           :  -
filter-join-rule                            :  -
is-enabled                                  :  true
is-read-only                                :  false
is-routable                                 :  true
ldap-data-source-pool                       :  pool-name
lexicographic-attrs                         :  all
lexicographic-lower-bound                   :  none
lexicographic-upper-bound                   :  none
non-viewable-attr                           :  -
non-writable-attr                           :  -
numeric-attrs                               :  all
numeric-default-data-view                   :  false
numeric-lower-bound                         :  none
numeric-upper-bound                         :  none
pattern-matching-base-object-search-filter  :  all
pattern-matching-dn-regular-expression      :  all
pattern-matching-one-level-search-filter    :  all
pattern-matching-subtree-search-filter      :  all
process-bind                                :  -
replication-role                            :  master
viewable-attr                               :  all except non-viewable-attr
writable-attr                               :  all except non-writable-attr
```

注 – 除代理管理员外，所有用户都会看到后端服务器中的 `cn=config` 和 `cn=monitor` 后缀。默认情况下，后端服务器中的数据对于代理管理员不可用。对于代理管理员可用的 `cn=config` 和 `cn=monitor` 子树就是代理服务器自身的这些子树。

创建目录代理服务器实例时，将使用空数据视图策略创建代理管理员的连接处理程序。如果代理管理员需要访问后端数据，则必须向代理管理员连接处理程序的数据视图策略中添加数据视图。默认情况下，在此类数据视图上将排除 `cn=config` 和 `cn=monitor` 子树。

2 **更改步骤 1 中列出的一个或多个属性。**

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name \
 property:value [property:value ... ]
```

例如，要访问数据源上的 `dc=example,dc=com` 子树，请在数据视图中指定 `base-dn`。

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 myDataView base-dn:dc=example,dc=com
```

3 **重新启动目录代理服务器实例以使更改生效（如有必要）。**

有关重新启动目录代理服务器的信息，请参见第 302 页中的 “重新启动目录代理服务器”。

重命名属性和 DN

目录中的每个条目都由 DN 和一组属性及属性值进行标识。通常，在客户端定义的 DN 和属性不会映射到在服务器端定义的 DN 和属性。可以定义数据视图以重命名 DN 和属性。客户端发出请求后，将对 DN 和属性进行重命名，以便与服务器端的名称相匹配。结果返回到客户端后，DN 和属性将会恢复原来的名称，以便与客户端的名称相匹配。

有关属性重命名和 DN 重命名的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Attribute Renaming and DN Renaming”。有关如何重命名属性和 DN 的信息，请参见以下过程：

▼ 配置属性重命名

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

- **在要配置属性映射的数据视图上设置一个或多个 `attr-name-mappings` 属性。**

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name \
 attr-name-mappings:client-side-attribute-name#server-side-attribute-name
  [attr-name-mappings:client-side-attribute-name#server-side-attribute-name ...]
```

例如，将客户端的 surname 重命名为服务器端的 sn。

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 myDataView \
 attr-name-mappings:surname#sn
```

▼ 配置 DN 重命名

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 查看要重命名 DN 的数据视图的 base-dn 属性和 DN 映射属性。

```
$ dpconf get-ldap-data-view-prop -h host -p port view-name base-dn \
 dn-mapping-source-base-dn dn-mapping-attrs
```

这些属性的含义如下：

- base-dn 是客户端子树的 DN，相当于数据视图的基 DN。
- dn-mapping-source-base-dn 是服务器端子树的 DN。
- dn-mapping-attrs 用于定义包含条目 DN 的属性的列表。

例如，未定义 DN 重命名时，客户端 dc=example,dc=com 数据库的数据视图具有以下值：

```
$ dpconf get-ldap-data-view-prop myDataView base-dn \
 dn-mapping-source-base-dn dn-mapping-attrs
 base-dn                   :  dc=example,dc=com
 dn-mapping-attrs          :  none
 dn-mapping-source-base-dn  :  none
```

2 将客户端的 DN 映射到服务器端的 DN。

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name \
  dn-mapping-source-base-dn:server-side-dn
```

例如，将客户端的 dc=example,dc=com 数据库映射到服务器端的 dc=example,dc=org。

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 myDataView \
 dn-mapping-source-base-dn:dc=example,dc=org
```

3 重命名受步骤 2 影响的 DIT 部分中的属性（如果这些属性包含 DN）。

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name \
  dn-mapping-attrs:attribute-name [dn-mapping-attrs:attribute-name ...]
```

例如，如果 group 属性包含 DN，并且所在的名称空间受步骤 2中重命名操作的影响，请按如下方式重命名该属性：

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 myDataView dn-mapping-attrs:group
```

4 **查看已重命名 DN 的数据视图的 base-dn 属性和 DN 映射属性。**

```
$ dpconf get-ldap-data-view-prop -h host -p port view-name base-dn \
 dn-mapping-source-base-dn dn-mapping-attrs
```

例如，客户端 dc=example,dc=com 数据库的数据视图在 DN 重命名后具有以下值：

```
$ dpconf get-ldap-data-view-prop -h host1 -p 1389 myDataView base-dn \
 dn-mapping-source-base-dn dn-mapping-attrs
 base-dn                    :  dc=example,dc=com
 dn-mapping-attrs           :  group
 dn-mapping-source-base-dn  :  dc=example,dc=org
```

配置 excluded-subtrees 和 alternate-search-base-dn

创建从属数据视图后，目录代理服务器会自动从上级数据视图中排除该从属数据视图。当请求针对从属数据视图时，该请求将被发送到从属数据视图，而不是上级数据视图。

在从属数据视图中指定备用搜索基后，还会在从属数据视图中执行针对上级数据视图的搜索操作。

默认情况下，目录代理服务器将自动配置 excluded-subtrees 和 alternate-search-base-dn 属性。以下过程将介绍如何手动配置这些属性。

▼ 手动配置 excluded-subtrees 和 alternate-search-base-dn 属性

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 **将目录代理服务器配置为手动路由请求。**

```
$ dpconf set-server-prop -h host -p port data-view-automatic-routing-mode:manual
```

当 data-view-automatic-routing-mode 为 manual 时，目录代理服务器不会生成 excluded-subtrees 和 alternate-search-base-dn 属性。您必须手动设置这些属性的值。目录代理服务器不会检查此处设置的值。请注意，错误地设置这些值可能会破坏管理路径。

或者，将目录代理服务器配置为手动路由部分请求。

```
$ dpconf set-server-prop -h host -p port data-view-automatic-routing-mode:limited
```

当 data-view-automatic-routing-mode 为limited 时， 目录代理服务器不会生成 excluded-subtrees 和 alternate-search-base-dn 属性。但是，目录代理服务器会检查此处设置的值是否与管理路径冲突。

2 **配置视图排除基。**

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name excluded-subtrees:suffix-DN
```

视图排除基用于确定数据视图不公开其条目的 DIT 分支。

3 **配置备用搜索基。**

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name \
 alternate-search-base-dn:search-base-DN
```

备用搜索基用于确定 DIT 的其他分支，属于此数据视图的条目可能位于这些分支中。默认情况下，在所有数据视图中都将基 DN 定义为备用搜索基。

为示例使用案例创建和配置数据视图

本部分包含以下有关数据视图以及如何创建和配置数据视图的信息：

本部分中的示例假定连接处理程序允许目录代理服务器处理所有客户端连接。

默认数据视图

如果在创建数据视图时未配置任何属性，则数据视图具有以下配置：

```
alternate-search-base-dn                 :  ""
alternate-search-base-dn                 :  base-DN
attr-name-mappings                       :  none
base-dn                                  :  suffix-DN
contains-shared-entries                  :  -
description                              :  -
distribution-algorithm                   :  -
dn-join-rule                             :  -
dn-mapping-attrs                         :  none
dn-mapping-source-base-dn                :  none
excluded-subtrees                        :  -
filter-join-rule                         :  -
```

```
is-enabled                                  :  true
is-read-only                                :  false
is-routable                                 :  true
ldap-data-source-pool                       :  pool-name
lexicographic-attrs                         :  all
lexicographic-lower-bound                   :  none
lexicographic-upper-bound                   :  none
non-viewable-attr                           :  -
non-writable-attr                           :  -
numeric-attrs                               :  all
numeric-default-data-view                   :  false
numeric-lower-bound                         :  none
numeric-upper-bound                         :  none
pattern-matching-base-object-search-filter  :  all
pattern-matching-dn-regular-expression      :  all
pattern-matching-one-level-search-filter    :  all
pattern-matching-subtree-search-filter      :  all
process-bind                                :  -
replication-role                            :  master
viewable-attr                               :  all except non-viewable-attr
writable-attr                               :  all except non-writable-attr
```

路由所有请求（不考虑请求的目标 DN）的数据视图

本部分显示将所有请求（不考虑请求的目标 DN）路由到数据源池的数据视图的配置。此数据视图称为**根数据视图**。默认情况下，在创建目录代理服务器实例时将创建根数据视图。有关根数据视图的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Data Views to Route All Requests, Irrespective of the Target DN of the Request”。

根数据视图具有以下配置：

```
alternate-search-base-dn                 :  -
attr-name-mappings                       :  none
base-dn                                  :  ""
contains-shared-entries                  :  -
description                              :  Automatically-generated data view
                                            able to route client operations
                                            independently of the operation base dn
distribution-algorithm                   :  -
dn-join-rule                             :  -
dn-mapping-attrs                         :  none
dn-mapping-source-base-dn                :  none
excluded-subtrees                        :  ""
excluded-subtrees                        :  cn=config
```

```
excluded-subtrees                           :  cn=monitor
excluded-subtrees                           :  cn=proxy manager
excluded-subtrees                           :  cn=virtual access controls
excluded-subtrees                           :  dc=example,dc=com
filter-join-rule                            :  -
is-enabled                                  :  true
is-read-only                                :  false
is-routable                                 :  true
ldap-data-source-pool                       :  defaultDataSourcePool
lexicographic-attrs                         :  all
lexicographic-lower-bound                   :  none
lexicographic-upper-bound                   :  none
non-viewable-attr                           :  -
non-writable-attr                           :  -
numeric-attrs                               :  all
numeric-default-data-view                   :  false
numeric-lower-bound                         :  none
numeric-upper-bound                         :  none
pattern-matching-base-object-search-filter :  all
pattern-matching-dn-regular-expression      :  all
pattern-matching-one-level-search-filter    :  all
pattern-matching-subtree-search-filter      :  all
process-bind                                :  -
replication-role                            :  master
viewable-attr                               :  all except non-viewable-attr
writable-attr                               :  all except non-writable-attr
```

当子树列表存储到多个数据相等的数据源时路由请求的数据视图

本部分介绍如何配置数据视图，以便将针对子树列表的请求路由到一组数据相等的数据源。有关此类部署的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Data Views to Route Requests When a List of Subtrees Are Stored on Multiple, Data-Equivalent Data Sources”。

本部分中的示例具有多个包含相同子树集的数据源。这些数据源具有相等的数据，并被集中在一个数据源池中以便实现负载平衡。每个子树都会配置一个数据视图，以便向客户端请求公开该子树。下图显示了样例部署。

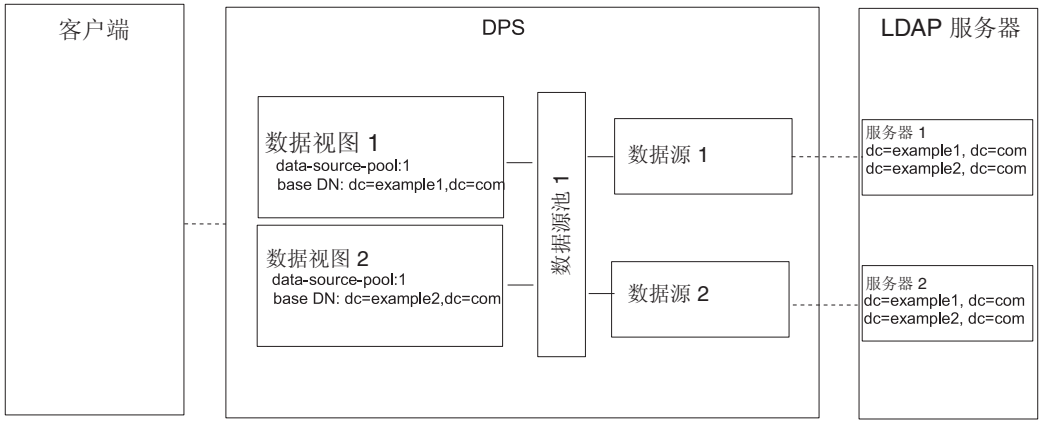


图 23–1 当子树列表存储到多个数据相等的数据源时路由请求的样例部署

▼ 配置当子树列表存储到多个数据相等的数据源时路由请求的数据视图

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1. **为每个 LDAP 服务器创建数据源，如第 317 页中的 “ 创建和配置 LDAP 数据源 ”中所述。**

2. **创建数据源池，如第 319 页中的 “ 创建和配置 LDAP 数据源池 ”中所述。**

3. **将数据源连接到数据源池，如第 320 页中的 “ 将 LDAP 数据源连接到数据源池 ”中所述。**

4. **（可选的）配置负载平衡。**
   有关信息，请参见第 333 页中的 “配置负载平衡”。

5. **创建基 DN 为 `dc=example1,dc=com` 且指向该数据源池的数据视图。**

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 dataview-1 \
  base-dn:dc=example1,dc=com ldap-data-source-pool:data-source-pool-1
```

6. **创建基 DN 为 `dc=example2,dc=com` 且指向该数据源池的另一个数据视图。**

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 dataview-2 \
  base-dn:dc=example2,dc=com ldap-data-source-pool:data-source-pool-1
```

   数据视图的其他属性与第 348 页中的 “默认数据视图”中的默认数据视图相同。

7 **重新启动目录代理服务器实例以使更改生效（如有必要）。**

有关重新启动目录代理服务器的信息，请参见第 302 页中的 “重新启动目录代理服务器” 。

当不同子树存储到不同数据源时提供单一访问点的数据视图

本部分介绍如何配置数据视图，以便向存储到多个数据源的不同子树提供单一访问点。有关此类部署的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Data Views to Provide a Single Point of Access When Different Subtrees Are Stored on Different Data Sources”。

本部分中的示例包含每个子树的数据视图。每组数据相等的数据源都会配置一个数据源池。下图显示了示例部署。

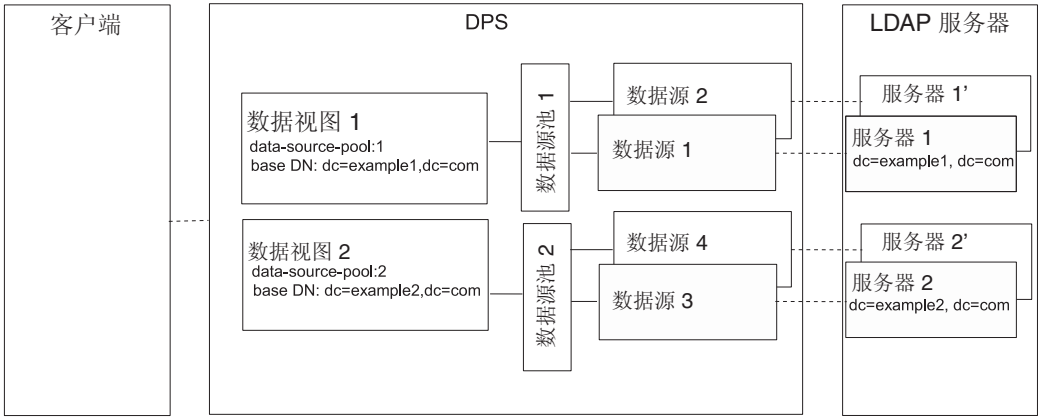


图 23–2 当不同子树存储到不同数据源时提供单一访问点的样例部署

▼ 配置当不同子树存储到不同数据源时提供单一访问点的数据视图

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1 **为每个 LDAP 服务器创建数据源，如第 317 页中的 “创建和配置 LDAP 数据源”中所述。**

2 **创建两个数据源池，如第 319 页中的 “创建和配置 LDAP 数据源池”中所述。**

3 **将包含 `dc=example1,dc=com` 的数据源连接到 `data-source-pool-1`，将包含 `dc=example2,dc=com` 的数据源连接到 `data-source-pool-2`，如第 320 页中的 “将 LDAP 数据源连接到数据源池”中所述。**

4 **（可选的）配置负载平衡。**

有关信息，请参见第 333 页中的 “配置负载平衡”。

5 **创建基 DN 为 `dc=example1,dc=com` 且指向 `data-source-pool-1` 的数据视图。**

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 dataview-1 \
  base-dn:dc=example1,dc=com ldap-data-source-pool:data-source-pool-1
```

6 **创建基 DN 为 `dc=example2,dc=com` 且指向 `data-source-pool-2` 的另一个数据视图。**

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 dataview-2 \
  base-dn:dc=example2,dc=com ldap-data-source-pool:data-source-pool-2
```

数据视图的其他属性与第 348 页中的 “默认数据视图”中的默认数据视图相同。

7 **重新启动目录代理服务器实例以使更改生效（如有必要）。**

有关重新启动目录代理服务器的信息，请参见第 302 页中的 “重新启动目录代理服务器”。

当子树的不同部分存储到不同数据源时提供单一访问点的数据视图

本部分介绍如何配置向子树的不同部分提供单一访问点的数据视图。此示例包含两个具有相同基 DN 的数据视图。可使用数字分配算法将条目分配到不同的数据视图中。每组数据相等的数据源都会配置一个数据源池。下图显示了示例部署。

有关此类部署的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Data Views to Route Requests When Different Parts of a Subtree Are Stored in Different Data Sources”。

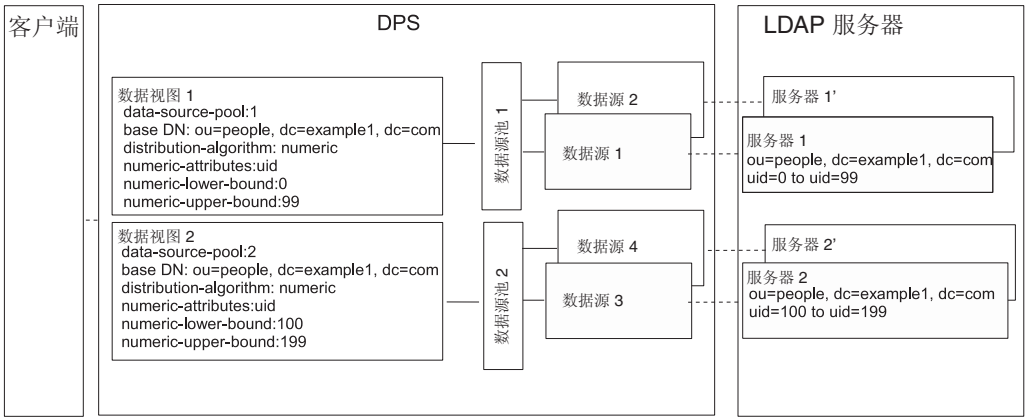


图 23–3 当子树的不同部分存储到不同数据源时提供单一访问点的样例部署

▼ 配置当子树的不同部分存储到不同数据源时提供单一访问点的数据视图

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1. **为每个 LDAP 服务器创建数据源，如第 317 页中的 “ 创建和配置 LDAP 数据源 ”中所述。**

2. **创建两个数据源池，如第 319 页中的 “ 创建和配置 LDAP 数据源池 ”中所述。**

3. **将包含部分子树的数据源连接到 `data-source-pool-1`，将包含子树其他部分的数据源连接到 `data-source-pool-2`，如第 320 页中的 “ 将 LDAP 数据源连接到数据源池 ”中所述。**

4. **（可选的）配置负载平衡。**
   有关信息，请参见第 333 页中的 “配置负载平衡”。

5. **创建具有分配算法的数据视图，以选择 `ou=people,dc=example,dc=com` 中 `uid` 介于 `0` 和 `99` 之间的条目，并将该数据视图配置为将请求指向 `data-source-pool-1`。**

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 dataview-1 \
 ldap-data-source-pool:data-source-pool-1 base-dn:ou=people,dc=example,dc=com \
 distribution-algorithm    :numeric numeric-attrs:uid numeric-lower-bound    :0 \
 numeric-upper-bound    :99
```

6. **创建另一个具有分配算法的视图，以选择 `ou=people,dc=example,dc=com` 中 `uid` 介于 `100` 和 `199` 之间的条目，并将该数据视图配置为将请求指向 `data-source-pool-2`。**

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 dataview-2 \
 ldap-data-source-pool:data-source-pool-2 base-dn:ou=people,dc=example,dc=com \
```

```
distribution-algorithm:numeric numeric-attrs:uid numeric-lower-bound:100
numeric-upper-bound    :199
```

数据视图的其他属性与第 348 页中的 “默认数据视图”中的默认数据视图相同。

7 **重新启动目录代理服务器实例以使更改生效（如有必要）。**

有关重新启动目录代理服务器的信息，请参见第 302 页中的 “重新启动目录代理服务器”。

当上级子树和从属子树存储到不同数据源时提供单一访问点的数据视图

本部分介绍如何配置数据视图，以便在子树的上级分支和从属分支存储到不同数据源时提供单一访问点。有关此类部署的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Data Views to Route Requests When Superior and Subordinate Subtrees Are Stored in Different Data Sources”。

本部分中的示例包含三个数据视图。数据视图 1 的基 DN 级别高于数据视图 2 和数据视图 3 的基 DN。或者，换句话说，数据源 2 和数据源 3 包含的子树从属于数据源 1 的子树。下图显示了示例部署。

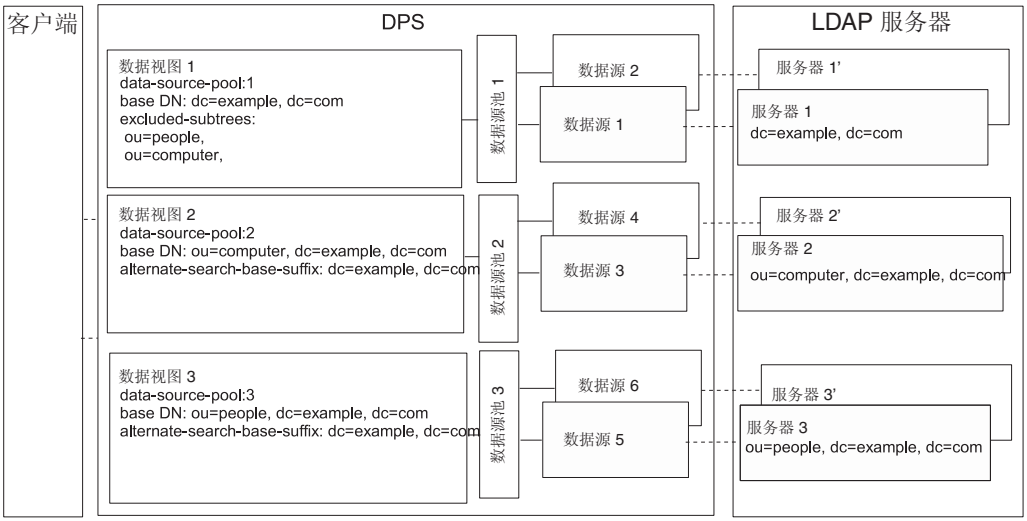


图 23–4 当上级子树和从属子树存储到不同数据源时路由请求的样例部署

将从属分支配置为单独数据视图的基 DN 时，目录代理服务器将自动从数据视图中排除子树的从属分支。

▼ 配置当上级子树和从属子树存储到不同数据源时提供单一访问点的数据视图

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1. **为每个 LDAP 服务器创建一个数据源，如第 317 页中的 “创建和配置 LDAP 数据源”中所述。**

2. **创建三个数据源池，如第 319 页中的 “创建和配置 LDAP 数据源池”中所述。**

3. **按照第 320 页中的 “将 LDAP 数据源连接到数据源池”中的说明将数据源连接到数据源池。**

   - 将包含 `dc=example,dc=com` 的数据源连接到 `data-source-pool-1`。
   - 将包含 `ou=computer,dc=example,dc=com` 的数据源连接到 `data-source-pool-2`。
   - 将包含 `ou=people,dc=example,dc=com` 的数据源连接到 `data-source-pool-3`。

4. **（可选的）配置负载平衡。**

   有关信息，请参见第 333 页中的 “配置负载平衡”。

5. **创建基 DN 为 `dc=example,dc=com` 且数据源池为 `data-source-pool-1` 的数据视图。**

   ```
   $ dpconf create-ldap-data-view -h host1 -p 1389 dataview-1 \
    data-source-pool-1 dc=example,dc=com
   ```

6. **创建基 DN 为 `ou=computer,dc=example,dc=com` 且数据源池为 `data-source-pool-2` 的数据视图。**

   ```
   $ dpconf create-ldap-data-view -h host1 -p 1389 dataview-2 \
    data-source-pool-2 ou=computer,dc=example,dc=com
   ```

7. **创建基 DN 为 `ou=people,dc=example,dc=com` 且数据源池为 `data-source-pool-3` 的数据视图。**

   ```
   $ dpconf create-ldap-data-view -h host1 -p 1389 dataview-3 \
    data-source-pool-3 ou=people,dc=example,dc=com
   ```

8. **通过查看 `excluded-subtrees` 参数，验证子树 `ou=computer,dc=example, dc=com` 和 `ou=people,dc=example, dc=com` 是否已从 `dataview-1` 中排除。**

   ```
   $ dpconf get-ldap-data-view-prop -h host1 -p 1389 dataview-1 excluded-subtrees
   ```

   将返回已排除的子树的列表。

9. **重新启动目录代理服务器实例以使更改生效（如有必要）。**

   有关重新启动目录代理服务器的信息，请参见第 302 页中的 “重新启动目录代理服务器”。

具有分层结构和分配算法的数据视图

本部分介绍如何配置结合使用分层结构和分配算法的数据视图。有关此类部署的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Data Views With Hierarchy and a Distribution Algorithm”。

本部分中的示例包含四个数据视图。数据视图 1 的基 DN 级别高于其他数据视图的基 DN。数据视图 3 和数据视图 4 具有相同的基 DN，但数字分配算法将条目分配到不同的数据视图中。

将从属分支配置为单独数据视图的基 DN 时，目录代理服务器将自动从数据视图中排除子树的从属分支。数字分配算法可将相同子树中的条目分配到不同的数据视图中。每组数据相等的数据源都会配置一个数据源池。

下图显示了示例部署。

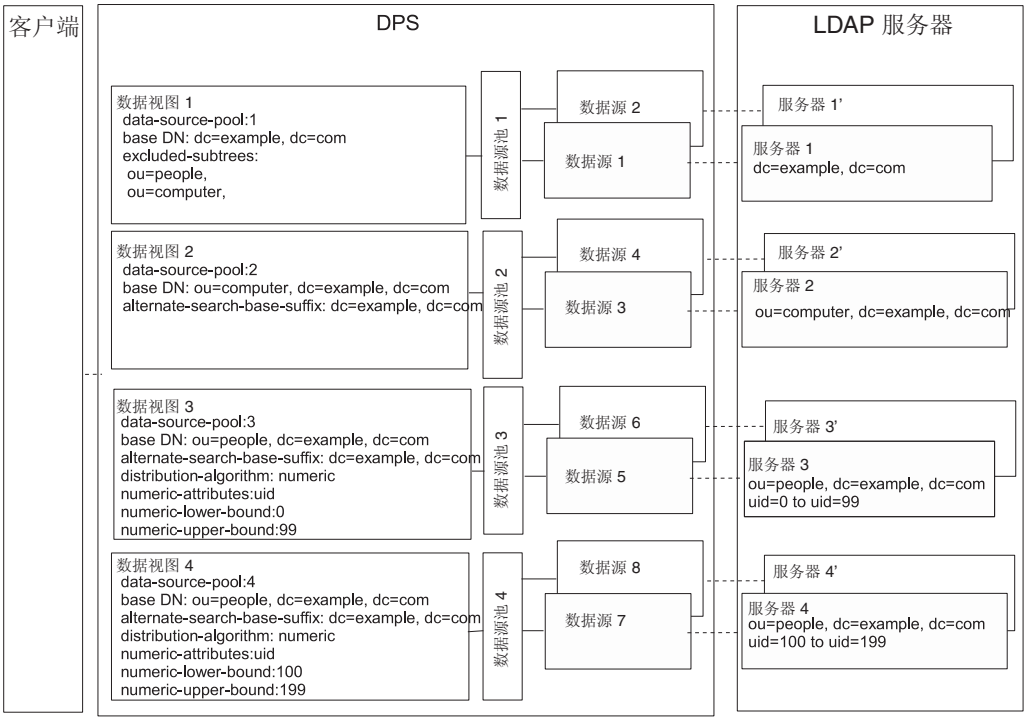


图 23–5 具有分层结构和分配算法的样例数据视图

▼ 配置具有分层结构和分配算法的数据视图

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1. 为每个 LDAP 服务器创建数据源，如第 317 页中的“ 创建和配置 LDAP 数据源 ”中所述。

2. 创建四个数据源池，如第 319 页中的“ 创建和配置 LDAP 数据源池 ”中所述。

3. 按照第 320 页中的“ 将 LDAP 数据源连接到数据源池 ”中的说明将数据源连接到数据源池。
   - 将包含 `dc=example,dc=com` 的数据源连接到 `data-source-pool-1`。
   - 将包含 `ou=computer,dc=example,dc=com` 的数据源连接到 `data-source-pool-2`。
   - 将包含 `ou=people,dc=example,dc=com` 中 `uid` 介于 `0` 和 `99` 之间的条目的数据源连接到 `data-source-pool-3`。
   - 将包含 `ou=people,dc=example,dc=com` 中 `uid` 介于 `100` 和 `199` 之间的条目的数据源连接到 `data-source-pool-4`。

4. （可选的）配置负载平衡。
   有关信息，请参见第 333 页中的 “配置负载平衡” 。

5. 创建基 DN 为 `dc=example,dc=com` 且指向 `data-source-pool-1` 的数据视图。

```
$ dpconf create-ldap-data-view -h host1 -p 1389 dataview-1 \
 data-source-pool-1 dc=example,dc=com
```

6. 创建基 DN 为 `ou=computer,dc=example,dc=com` 且指向 `data-source-pool-2` 的数据视图。

```
$ dpconf create-ldap-data-view -h host1 -p 1389 dataview-2 \
 data-source-pool-2 ou=computer,dc=example,dc=com
```

7. 创建基 DN 为 `ou=people,dc=example,dc=com` 且指向 `data-source-pool-3` 的数据视图。在数据视图上配置分配算法，以选择 `uid` 介于 `0` 和 `99` 之间的条目。

```
$ dpconf create-ldap-data-view -h host1 -p 1389 dataview-3 \
 data-source-pool-3 ou=people,dc=example,dc=com
$ dpconf set-ldap-data-view-prop dataview-3 distribution-algorithm:numeric \
 numeric-attrs:uid numeric-lower-bound:0 numeric-upper-bound:99
```

8. 创建基 DN 为 `ou=people,dc=example,dc=com` 且指向 `data-source-pool-4` 的数据视图，并在该数据视图上配置分配算法，以选择 `uid` 介于 `100` 和 `199` 之间的条目。

```
$ dpconf create-ldap-data-view -h host1 -p 1389 dataview-4 \
 data-source-pool-4 ou=people,dc=example,dc=com
$ dpconf set-ldap-data-view-prop dataview-4 distribution-algorithm:numeric \
 numeric-attrs:uid numeric-lower-bound:100 numeric-upper-bound:199
```

9. 通过查看 `excluded-subtrees` 参数，验证子树 `ou=computer,dc=example, dc=com` 和 `ou=people,dc=example, dc=com` 是否已从 `dataview-1` 中排除。

```
$ dpconf get-ldap-data-view-prop -h host1 -p 1389 dataview-1 excluded-subtrees
```

将返回已排除的子树的列表。

10 **重新启动目录代理服务器实例以使更改生效。**

有关重新启动目录代理服务器的信息，请参见第 302 页中的 “重新启动目录代理服务器”。

◆ ◆ ◆ 第 24 章

24 目录代理服务器虚拟数据视图

本章介绍如何创建虚拟数据视图。**虚拟数据视图**会以某种方式转换源数据，并向客户端应用程序显示该数据的不同视图。虚拟数据视图包括转换的 LDAP 数据视图、LDIF 数据视图、联接数据视图和 JDBC™ 数据视图。有关虚拟数据视图功能的概述，以及示例使用案例的描述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的第 23 章 “Virtual Data Views”。

您无法使用目录服务控制中心 (Directory Service Control Center, DSCC) 执行本章中的过程。必须使用命令行。

本章包含以下主题：

创建和配置 LDIF 数据视图

LDIF 数据视图是一种简单的虚拟数据视图，此视图中的 LDIF 文件类似于 LDAP 数据源。与 LDAP 数据视图不同，在设置 LDIF 数据视图时无需创建数据源或数据源池。但在创建数据视图时应指定 LDIF 文件。默认情况下，无法向 LDIF 数据视图中写入内容。有关详细信息，请参见第 372 页中的 “在虚拟数据视图上定义访问控制”。

有关创建和配置 LDIF 数据视图的信息，请参见以下过程。

▼ 创建 LDIF 数据视图

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1 创建 LDIF 数据视图。

```
$ dpconf create-ldif-data-view -h host -p port view-name path-to-ldif-file suffix-dn
```

2 （可选的）查看 LDIF 数据视图的列表。

```
$ dpconf list-ldif-data-views -h host -p port
```

虚拟访问控制数据视图是唯一的默认 LDIF 数据视图。此数据视图由服务器生成，可以将请求路由到虚拟访问控制指令 (access control instruction, ACI)。

▼ 配置 LDIF 数据视图

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1 查看 LDIF 数据视图的属性。

```
$ dpconf get-ldif-data-view-prop -h host -p port view-name
```

LDIF 数据视图具有以下默认属性：

```
alternate-search-base-dn                    :  ""
alternate-search-base-dn                    :  dc=com
attr-name-mappings                          :  none
base-dn                                     :  suffixDN
bind-pwd-attr                               :  userPassword
contains-shared-entries                     :  -
db-pwd-encryption                           :  clear-text
description                                 :  -
distribution-algorithm                      :  -
dn-join-rule                                :  -
dn-mapping-attrs                            :  none
dn-mapping-source-base-dn                   :  none
excluded-subtrees                           :  -
filter-join-rule                            :  -
is-enabled                                  :  true
is-read-only                                :  false
is-routable                                 :  true
ldif-data-source                            :  /path/to/filename.ldif
lexicographic-attrs                         :  all
lexicographic-lower-bound                   :  none
lexicographic-upper-bound                   :  none
non-viewable-attr                           :  -
non-writable-attr                           :  -
numeric-attrs                               :  all
```

```
numeric-default-data-view                    :  false
numeric-lower-bound                          :  none
numeric-upper-bound                          :  none
pattern-matching-base-object-search-filter   :  all
pattern-matching-dn-regular-expression       :  all
pattern-matching-one-level-search-filter     :  all
pattern-matching-subtree-search-filter       :  all
process-bind                                 :  -
replication-role                             :  master
viewable-attr                                :  all except non-viewable-attr
writable-attr                                :  all except non-writable-attr
```

2 更改步骤 1 中列出的一个或多个属性。

```
$ dpconf set-ldif-data-view-prop -h host -p port view-name property:value \
 [property:value ... ]
```

例如，要更改数据视图的源 LDIF 文件，请设置 `ldif-data-source` 属性。

```
$ dpconf set-ldif-data-view-prop -h host1 -p 1389 -D cn="Proxy Manager" myLDIFDataView \
 ldif-data-source:/local/files/example.ldif
```

配置虚拟数据转换

虚拟数据转换是在现有数据视图上定义的，并通过实际的数据视图创建虚拟数据视图。有关虚拟数据转换的工作方式的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Virtual Data Transformations”。

可向以下任意类型的数据视图中添加虚拟数据转换：LDAP 数据视图、LDIF 数据视图、联接数据视图或 JDBC 数据视图。

▼ 添加虚拟转换

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1 向数据视图中添加转换。

```
$ dpconf add-virtual-transformation -h host -p port view-name \
 transformation-model transformation-action attribute-name [parameters...]
```

请注意，*parameters* 可能是必需选项，这取决于 *transformation-model* 和 *transformation-action*。有关转换模型、转换操作和转换参数的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Virtual Data Transformations”。

2 （可选的）查看数据视图上定义的虚拟转换列表。

```
$ dpconf list-virtual-transformations -h host -p port view-name
```

创建和配置联接数据视图

联接数据视图是多个数据视图的聚合。有关联接数据视图的工作方式的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Join Data Views”。

有关如何创建和配置联接数据视图的信息，请参见以下过程。

▼ 创建联接数据视图

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1 对将要聚合成联接视图的主数据视图和从数据视图进行标识。

在创建联接视图之前，必须存在主数据视图和从数据视图。主数据视图和从数据视图可以是任意类型的数据视图，包括 LDAP 数据视图、LDIF 数据视图、JDBC 数据视图或其他联接数据视图。必须在从视图上配置一些特定属性，以便将其作为联接视图的源。有关详细信息，请参见第 365 页中的 “配置联接视图的从视图” 。

2 创建联接数据视图。

```
$ dpconf create-join-data-view -h host -p port view-name primary-view secondary-view \
 suffix-dn
```

3 （可选的）查看联接视图列表，以检查是否已成功创建数据视图。

```
$ dpconf list-join-data-views -h host -p port
```

▼ 配置联接数据视图

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1 查看联接数据视图的属性。

```
$ dpconf get-join-data-view-prop -h host -p port view-name
```

联接数据视图的默认属性如下所示：

```
alternate-search-base-dn                    :  ""
alternate-search-base-dn                    :  dc=com
attr-name-mappings                          :  none
base-dn                                     :  suffixDN
contains-shared-entries                     :  -
description                                 :  -
distribution-algorithm                      :  -
dn-join-rule                                :  -
dn-mapping-attrs                            :  none
```

```
dn-mapping-source-base-dn                    :  none
excluded-subtrees                            :  -
filter-join-rule                             :  -
is-enabled                                   :  true
is-read-only                                 :  false
is-routable                                  :  true
lexicographic-attrs                          :  all
lexicographic-lower-bound                    :  none
lexicographic-upper-bound                    :  none
non-viewable-attr                            :  -
non-writable-attr                            :  -
numeric-attrs                                :  all
numeric-default-data-view                    :  false
numeric-lower-bound                          :  none
numeric-upper-bound                          :  none
pattern-matching-base-object-search-filter   :  all
pattern-matching-dn-regular-expression       :  all
pattern-matching-one-level-search-filter     :  all
pattern-matching-subtree-search-filter       :  all
primary-view                                 :  primary-view
process-bind                                 :  -
replication-role                             :  master
secondary-view                               :  secondary-view
viewable-attr                                :  all except non-viewable-attr
writable-attr                                :  all except non-writable-attr
```

2 **更改步骤 1 中列出的一个或多个属性。**

```
$ dpconf set-join-data-view-prop -h host -p port view-name property:value \
 [property:value ... ]
```

例如，要将数据源的主数据视图更改为 `myLDAPDataView`，请使用以下命令：

```
$ dpconf set-join-data-view-prop -h host1 -p 1389 -D cn="Proxy Manager" \
 myJoinDataView primary-view:myLDAPDataView
```

3 **重新启动目录代理服务器实例以使更改生效（如有必要）。**

有关重新启动目录代理服务器的信息，请参见第 302 页中的 “重新启动目录代理服务器”。

▼ 配置联接视图的从视图

必须在从数据视图上配置一些特定属性，以便将其作为联接视图的源。由于从视图可以是任意类型的数据视图，因此您所使用的命令取决于数据视图类型。以下样例命令假定从视图为 LDAP 数据视图。有关此处介绍的属性的详细信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Additional Secondary Data View Properties”。

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1 **定义联接规则，以确定从视图与主视图的关联方式。**

联接规则可为以下任一选项：

- DN 联接规则

```
$ dpconf set-ldap-data-view-prop -h host -p port secondary-view-name \
 dn-join-rule:uid=\${primary-view-name.uid},ou=People,dc=example
```

- 过滤器联接规则

```
$ dpconf set-ldap-data-view-prop -h host -p port secondary-view-name filter-join-rule:uid=\${primary-view-name.uid}
```

2 **（可选的）指定从视图上是否允许绑定。**

默认情况下，所有数据视图上都允许绑定。如果您要禁止绑定到从数据视图，请运行以下命令：

```
$ dpconf set-ldap-data-view-prop -h host -p port secondary-view-name process-bind:false
```

有关此属性的详细信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Handling of Binds”。

3 **（可选的）指定从视图是否包含共享条目。**

```
$ dpconf set-ldap-data-view-prop -h host -p port secondary-view-name contains-shared-entries:true
```

有关此属性的详细信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Handling of Shared Entries”。

创建和配置 JDBC 数据视图

JDBC 数据视图使得 LDAP 客户端应用程序可以访问关系数据库。有关 JDBC 数据视图的工作方式的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“JDBC Data Views”。

有关如何创建和配置 JDBC 数据视图的信息，请参见以下过程。

▼ 创建 JDBC 数据视图

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1 **为关系数据库创建 JDBC 数据源。**

```
$ dpconf create-jdbc-data-source -h host -p port -b db-name -B db-url -J driver-url \
 -S driver-class source-name
```

目前，每个 JDBC 数据视图只支持一个 JDBC 数据源。换句话说，您无法跨 JDBC 数据源实现负载平衡。要访问多个 JDBC 数据源，可以为每个数据源创建一个数据视图，然后通过联接视图将这些数据视图联接在一起。

在创建 JDBC 数据源时，必须设置以下属性：

`db-name`	关系数据库的名称，例如 `payrolldb`。
`db-url`	指向数据库的 URL，格式为 `jdbc:` *vendor*`:` *driver*`://` *dbhost*`:` *dbport*。 `db-url` **不是**完整的 JDBC 数据库 URL，因为它不包含数据库名称。（数据库名称由 `db-name` 属性指定。）
`driver-class`	JDBC 驱动程序类，例如 `org.hsqldb.jdbcDriver`。
`driver-url`	JDBC 驱动程序所在的路径，例如 `file:///` *path/to/hsqldb*`/lib/hsqldb.jar`。

2 创建 JDBC 数据源池。

```
$ dpconf create-jdbc-data-source-pool -h host -p port pool-name
```

3 将 JDBC 数据源连接到此 JDBC 数据源池。

```
$ dpconf attach-jdbc-data-source -h host -p port pool-name source-name
```

4 创建 JDBC 数据视图。

```
$ dpconf create-jdbc-data-view -h host -p port view-name pool-name suffix-DN
```

5 （可选的）查看 JDBC 数据视图列表，以检查是否已成功创建数据视图。

```
$ dpconf list-jdbc-data-views -h host -p port
```

▼ 配置 JDBC 数据视图

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1 查看 JDBC 数据视图的属性。

```
$ dpconf get-jdbc-data-view-prop -h host -p port view-name
```

JDBC 数据视图的默认属性如下所示：

```
alternate-search-base-dn                    :  -
attr-name-mappings                          :  none
base-dn                                     :  o=sql1
contains-shared-entries                     :  -
description                                 :  -
distribution-algorithm                      :  -
```

```
dn-join-rule                                 :  -
dn-mapping-attrs                             :  none
dn-mapping-source-base-dn                    :  none
excluded-subtrees                            :  -
filter-join-rule                             :  -
is-enabled                                   :  true
is-read-only                                 :  false
is-routable                                  :  true
jdbc-data-source-pool                        :  pool-name
lexicographic-attrs                          :  all
lexicographic-lower-bound                    :  none
lexicographic-upper-bound                    :  none
non-viewable-attr                            :  -
non-writable-attr                            :  -
numeric-attrs                                :  all
numeric-default-data-view                    :  false
numeric-lower-bound                          :  none
numeric-upper-bound                          :  none
pattern-matching-base-object-search-filter   :  all
pattern-matching-dn-regular-expression       :  all
pattern-matching-one-level-search-filter     :  all
pattern-matching-subtree-search-filter       :  all
process-bind                                 :  -
replication-role                             :  master
viewable-attr                                :  all except non-viewable-attr
writable-attr                                :  all except non-writable-attr
```

2 更改步骤 1 中列出的一个或多个属性。

```
$ dpconf set-jdbc-data-view-prop -h host -p port view-name property:value \
 [property:value ... ]
```

▼ 配置 JDBC 表、属性和对象类

配置 JDBC 数据视图时，还必须配置以下对象：

- **JDBC 对象类**。将一个或多个 JDBC 表映射到 LDAP 对象类。
- **JDBC 表**。为每个关系数据库表定义。
- **JDBC 属性**。从 JDBC 表的指定列中定义 LDAP 属性。

1 为关系数据库中的每个表创建 JDBC 表。

```
% dpconf create-jdbc-table jdbc-table-name db-table
```

db-table 的名称区分大小写。请确保使用的大小写与关系数据库中使用的相同，否则针对该表的操作可能会失败。

2 为每个关系数据库表中的每个列创建 JDBC 属性。

```
% dpconf add-jdbc-attr table-name attr-name sql-column
```

创建 JDBC 属性会将表列映射到 LDAP 属性。

3 （可选的）如果关系数据库中的列区分大小写，请更改 JDBC 属性的 LDAP 语法。

```
% dpconf set-jdbc-attr-prop table-name attr-name ldap-syntax:ces
```

默认情况下，`ldap-syntax` 的值为 `cis`。这表明 `jdbc-attr` 不区分大小写。如果您的关系数据库区分大小写，请将值更改为 `ces`。

默认情况下，某些关系数据库（如 Oracle 和 DB2）区分大小写。LDAP 在默认情况下不区分大小写。当目录代理服务器检测到关系数据库表的某个列区分大小写时，在过滤器中具有相应属性的 `ldapsearch` 查询将被转换为使用函数 `UPPER` 的 SQL 查询。

例如，查询 `ldapsearch -b "dc=mysuffix" "(attr=abc)"` 将被转换为以下 SQL 查询：

```
SELECT * FROM mytable WHERE (UPPER(attr)='ABC')
```

默认情况下，此类查询不会编制索引。因此，具有此特性的查询可能会造成较大的性能影响。

可通过以下两种方式减轻性能影响：

- 将 jdbc-attr 的 `ldap-syntax` 属性设置为 `ces`。
- 对于每个可能会在 LDAP 过滤器中使用的 `jdbc-attr`，使用函数 `UPPER` 创建索引。

4 为 LDAP 关系数据库表创建 JDBC 对象类。

```
% dpconf create-jdbc-object-class view-name objectclass primary-table \
  [secondary-table... ] DN-pattern
```

创建 JDBC 对象类实际上是指定将与这些表相关联的 LDAP 对象类。JDBC 对象类还将指定主表和从表（如果这些表存在）。

创建 JDBC 对象类时将指定 DN 模式。DN 模式用于显示条目 DN 的构建方式。

5 如果存在从表，请定义主表和从表之间的联接规则。

```
% dpconf set-jdbc-table-prop secondary-table-name filter-join-rule:join-rule
```

联接规则在从表上进行定义，用于确定该表中的数据如何链接到主表数据。对象类主表和从表关系的定义方式非常重要。有关详细信息，请参见第 370 页中的 “定义 JDBC 表之间的关系”。

6 指定 JDBC 对象类的超类。

```
% dpconf set-jdbc-object-class-prop view-name objectclass super-class:value
```

超类表示 JDBC 对象类所继承的 LDAP 对象类。

定义 JDBC 表之间的关系

在最简单的情况下，JDBC 对象类仅包含单个（主）表。不存在从表，因此无需定义表之间的关系。

如果对象类包含多个表，则必须明确定义这些表之间的关系。表之间的关系始终在从表上进行定义。可以使用从表的以下属性定义这些关系：

- `is-single-row-table` 指定 LDAP 条目在表中只有一个匹配行。
- `contains-shared-entries` 指定从表中的一行由主表中的多个行使用。
- `filter-join-rule` 表示应如何根据主表内容从从表中检索条目。

以下示例将说明如何根据前两个属性的值定义过滤器联接规则。这些示例假定对象类具有一个主表和一个从表。

示例 24–1 `is-single-row-table:true` 和 `contains-shared-entries:true`

以上是这些属性的默认值。在此案例中，主表和从表之间的关系为 *n->1*，也就是说，主表中的 *n* 个行将引用从表中的一个共享行。

在关系数据库中，外键 (foreign key, FK) 在主表中定义，它指向从表的某个列。

例如，在某个组织中，一位经理可以管理多名员工。定义了两个关系数据库表，结构如下：

```
primary table : EMPLOYEE [ID, NAME, FK_MANAGER_ID]
secondary table : MANAGER  [ID, NAME]
```

定义了以下对象类和属性：

```
object-class : employee
attr : name (from primary EMPLOYEE.NAME)
attr : manager (from secondary MANAGER.NAME)
```

在从表中定义了以下过滤器联接规则：

```
"${ID}=${EMPLOYEE.FK_MANAGER_ID}"
```

在此配置下，LDAP 操作的运行方式如下：

- **添加员工条目。**如果员工条目中的经理在表中不存在，将创建一个新行。如果该经理存在，则使用现有行。
- **替换条目中 "manager" 属性的值。**`MANAGER.NAME` 行的值将发生更改。
- **删除员工条目。**从表中的行不会删除，因为经理条目为共享条目。
- **从条目中删除 "manager" 属性。**从表中的行将被删除，并且外键 (`EMPLOYEE.FK_MANAGER_ID`) 被设置为 `NULL`。

示例 24–2 `is-single-row-table:true` 和 `contains-shared-entries:false`

在此案例中，主表和从表之间的关系为 *1->1* 或 *1<-1*，也就是说，从表中的一行将引用主表中的一行。

在关系数据库中，外键 (foreign key, FK) 可能在主表中定义，也可能在从表中定义。

例如，在某个组织中，员工的 UID 存储在一个表中，其姓氏存储在另一个表中。定义了两个关系数据库表，结构如下：

```
primary table : UID [ID, VALUE, FK_SN_ID]
secondary table : SN [ID, VALUE]
```

定义了以下对象类和属性：

```
object-class : employee
attr : uid (from primary UID.VALUE)
attr : sn (from secondary ID.VALUE)
```

在从表中定义了以下过滤器联接规则：

```
"${ID}=${UID.FK_SN_ID}"
```

此配置也可能是另外一种方式，即外键 `FK_UID_ID` 存储在从表中，并指向 `UID.ID`。

示例 24–3 `is-single-row-table:false` 和 `contains-shared-entries:false`

在此案例中，主表和从表之间的关系为 *1->n*，也就是说，从表中的 *n* 个行将引用主表中的一行。此示例说明了多值属性的情况。多值属性在从表中以一组行表示，每个属性值为一行。

在关系数据库中，外键在从表中定义，它指向主表中的某个列。

例如，在某个组织中，员工可以有多个电话号码。定义了两个关系数据库表，结构如下：

```
primary table : EMPLOYEE [ID, NAME]
secondary table : PHONE [ID, VALUE, USER_ID]
```

定义了以下对象类和属性：

```
object-class : employee
attr : cn (from primary EMPLOYEE.NAME)
attr : telephoneNumber (from secondary PHONE.VALUE)
```

在从表中定义了以下过滤器联接规则：

示例 24–3 is-single-row-table:false 和 contains-shared-entries:false （续）

```
"${USER_ID}=${EMPLOYEE.ID}"
```

示例 24–4 is-single-row-table:false 和 contains-shared-entries:true

目录代理服务器目前不支持此案例。

在虚拟数据视图上定义访问控制

虚拟数据视图上的 ACI 可以存储在 LDAP 目录或 LDIF 文件中。有关虚拟 ACI 的工作方式的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Access Control On Virtual Data Views”。

创建目录代理服务器实例时，将为虚拟访问控制定义以下默认配置：

- 默认情况下存储 ACI 的 LDIF 文件 (*instance-path*/config/access_controls.ldif)
- 名为**虚拟访问控制**的 LDIF 数据视图
  目录代理服务器可以通过此数据视图访问 LDIF 文件中存储的 ACI。

▼ 定义新的 ACI 存储系统信息库

如果不想使用前面介绍的默认 ACI 配置，可以定义其他的存储系统信息库。

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1 为要存储 ACI 的系统信息库创建数据视图。

- 如果 ACI 将存储在 LDAP 目录中，请创建一个 LDAP 数据源和一个 LDAP 数据视图，如第 343 页中的 “创建和配置 LDAP 数据视图”中所述。
- 如果 ACI 将存储在 LDIF 文件中，请创建一个 LDIF 数据视图，如第 361 页中的 “创建和配置 LDIF 数据视图”中所述。

2 将上一步创建的数据视图的名称指定为 ACI 数据视图。

```
$ dpconf set-virtual-aci-prop -h host -p port aci-data-view:data-view-name
```

3 如果 ACI 系统信息库是 LDAP 目录，请定义访问 ACI 数据视图所需的凭证。

```
$ dpconf set-virtual-aci-prop -h host -p port aci-manager-bind-dn:bind-dn
$ dpconf set-virtual-aci-prop -h host -p port aci-manager-bind-pwd-file:filename
```

▼ 配置虚拟访问控制

无论使用哪种 ACI 系统信息库，都必须配置虚拟访问控制。

注 – 只有代理管理员才能直接通过 ACI 数据视图创建 ACI 池和管理 ACI。如果 ACI 系统信息库是 LDAP 目录，则必须修改该目录的模式，以使其包含 `aciSource` 对象类和 `dpsaci` 属性。有关自定义该模式的详细信息，请参见第 263 页中的 “扩展目录服务器模式”。

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1 **在 ACI 系统信息库中创建 ACI 池，并设置全局 ACI。**

有关全局 ACI 的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Global ACIs”。要设置全局 ACI，请在 ACI 数据视图的视图基下添加一个 `aciSource` 条目。例如：

```
% ldapmodify -p port -D "cn=proxy manager" -w -
dn: cn=data-source-name,cn=virtual access controls
changetype: add
objectclass: aciSource
dpsaci: (targetattr="*") (target = "ldap:///ou=people,o=virtual") (version 3.0; \
 acl "perm1"; allow(all) groupdn="ldap:///cn=virtualGroup1,o=groups,o=virtual";)
cn: data-source-name
```

2 **将一个或多个连接处理程序配置为使用此 ACI 池。**

```
% dpconf set-connection-handler-prop -h host -p port connection-handler aci-source:data-source-name
```

3 **将所需的 ACI 添加到数据中。**

要执行此操作，请创建包含 ACI 的虚拟条目。例如：

```
% ldapmodify -p port -D "cn=virtual application,ou=application users,dc=com" -w -
dn: ou=people,o=virtual
changetype: modify
add: dpsaci
dpsaci: (targetattr="*")(version 3.0; acl "perm1"; allow(all) userdn ="ldap:///self";)
dpsaci: (targetattr="*")(version 3.0; acl "perm1"; allow(search, read, compare) \
 userdn ="ldap:///anyone";)
```

注 – 具有相应访问权限的任何用户都可以通过数据视图添加和检索虚拟 ACI。

在虚拟数据视图上定义模式检查

对于 LDAP 数据视图，通常由后端目录使用自身的模式来执行模式检查。如果希望目录代理服务器执行模式检查，请使用以下过程。

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

要标准化请求（特别是 DN），请按如下方式设置服务器的 `use-external-schema` 属性：

▼ 定义模式检查

1. **指示服务器实例使用外部模式。**

   ```
   $ dpconf set-server-prop -h host -p port use-external-schema:true
   ```

2. **对连接处理程序启用模式检查。**

   ```
   $ dpconf set-connection-handler-prop -h host -p port connection-handler\
    schema-check-enabled:true
   ```

3. **创建公开 `cn=schema` 的数据视图。**

   如果在 LDAP 目录中定义外部模式，请使用视图基 `cn=schema` 创建 LDAP 数据视图，如第 343 页中的 “创建和配置 LDAP 数据视图”中所述 。

   如果在 LDIF 文件中定义外部模式，请使用视图基 `cn=schema` 创建 LDIF 数据视图，如第 361 页中的 “创建和配置 LDIF 数据视图”中所述。

4. **将此数据视图添加到由连接处理程序公开的数据视图列表中。**

   默认情况下，所有数据视图都由连接处理程序公开。如果已定义由连接处理程序公开的数据视图的自定义列表，请将此数据视图添加到列表中。必须在此命令中指定将由连接处理程序公开的**所有**数据视图。

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler \
 data-view-routing-custom-list:data-view-name data-view-routing-custom-list:data-view-name
```

样例虚拟配置

以下部分提供了两个样例配置。这些配置说明虚拟目录的主要功能，并指出这些功能的配置方式。

联接 LDAP 目录和 MySQL 数据库

本部分中的过程介绍联接 LDAP 目录和 MySQL 数据库的样例虚拟配置。LDAP 目录是主数据源，其中包含大多数用户信息。MySQL 数据库包含有关用户的其他信息。下图说明了最终的配置。

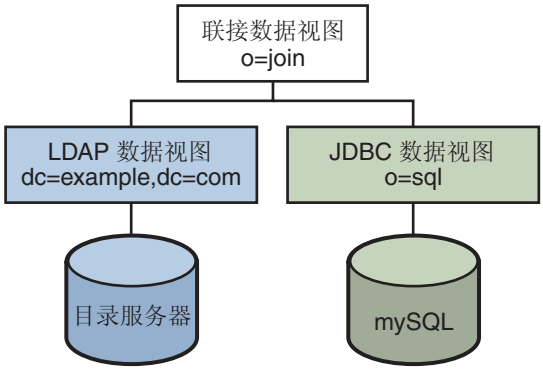


图 24–1 样例虚拟配置

可以使用 *install-path*`/ds6/ldif/Example.ldif` 中提供的样例数据复制此示例，也可以使用您自己的数据替换样例数据。

此配置可以分为三个部分：

- 配置和测试 LDAP 数据视图
- 配置和测试 JDBC 数据视图
- 配置和测试联接数据视图

为了简单起见，本部分中的所有命令都假定目录代理服务器在 `/local/dps` 中的本地主机上运行。这些命令还假定设置了以下环境变量：

`DIR_PROXY_PORT` `1389`

`LDAP_ADMIN_PWF` `pwd.txt`，包含管理员密码的文件。

`DIRSERV_PORT` `4389`

`LDAP_ADMIN_USER` `cn=Directory Manager`

配置和测试 LDAP 数据视图

▼ 配置 LDAP 数据视图

开始之前 本部分中的任务假定运行环境如下：

- 目录服务器实例通过端口 4389 在 host1上运行。
- 目录服务器中的数据存储在后缀 dc=example,dc=com 下。要复制此示例，请创建一个目录服务器实例，并创建后缀 dc=example,dc=com，然后将样例数据导入 *install-path*/ds6/ldif/Example.ldif 中。

1 为目录服务器实例创建名为 myds1 的 LDAP 数据源。

```
% dpconf create-ldap-data-source myds1 host1:4389
```

2 启用该数据源，并允许对其执行写入操作。

```
% dpconf set-ldap-data-source-prop myds1 is-enabled:true is-read-only:false
```

3 创建名为 myds1-pool 的 LDAP 数据源池。

```
% dpconf create-ldap-data-source-pool myds1-pool
```

4 将 LDAP 数据源连接到此 LDAP 数据源池。

```
% dpconf attach-ldap-data-source myds1-pool myds1
```

5 指定数据源应接收来自该数据源池的所有绑定、添加、搜索和修改操作。

```
% dpconf set-attached-ldap-data-source-prop myds1-pool myds1 add-weight:100 \
 bind-weight:100 modify-weight:100 search-weight:100
```

6 为数据源池创建名为 myds1–view 且基 DN 为 dc=example,dc=com 的 LDAP 数据视图。

```
% dpconf create-ldap-data-view myds1-view myds1-pool dc=example,dc=com
```

▼ 测试 LDAP 数据视图

1 以 dc=example,dc=com 下的用户身份搜索 LDAP 数据源中的所有条目，以验证您是否可以读取数据视图中的内容。

```
% ldapsearch -p 1389 -D "uid=kvaughan,ou=people,dc=example,dc=com" -w bribery \
 -b dc=example,dc=com "objectclass=*"
```

注 – 必须使用 dc=example,dc=com 下的用户凭证。如果要使用 cn=Directory Manager，则必须定义用于处理该 DN 的数据视图。

---

2 **以 dc=example,dc=com 下的用户身份修改 userPassword 属性，以验证您是否可以向数据视图中写入内容。**

```
% ldapmodify -p 1389 -D "uid=kvaughan,ou=people,dc=example,dc=com" -w bribery
dn: uid=kvaughan,ou=people,dc=example,dc=com
changetype: modify
replace: userPassword
userPassword: myNewPassword
```

注 – 目录服务器中的默认 ACI 允许用户修改自己的密码。

配置和测试 JDBC 数据视图

以下任务假定已安装了 MySQL 数据库，该数据库正在运行并且填充了数据，同时该数据库还具有以下特性：

- 数据库名称：`sample_sql`
- 数据库 URL：`host2.example.com:3306`
- JDBC 驱动程序 URL：`file:/net/host2.example/local/mysql/lib/jdbc.jar`
- 驱动程序类：`com.mysql.jdbc.Driver`
- 数据库用户：`root`
- 数据库密码文件：`mysqlpwd.txt`

下表介绍数据库中的表及其复合字段。您需要使用此信息来设置 JDBC 数据视图。

MySQL 表	字段
EMPLOYEE	ID、SURNAME、PASSWORD、TITLE、COUNTRY_ID
COUNTRY	ID、NAME
PHONE	USER_ID、NUMBER

▼ 配置 JDBC 数据视图

1 **为 SQL 数据库创建名为 mysql1 的 JDBC 数据源。**

```
% dpconf create-jdbc-data-source -b sample_sql -B jdbc:mysql://host2:3306 \
 -J file:/net/host2.example/local/mysql/lib/jdbc.jar -S com.mysql.jdbc.Driver mysql1
```

2 **指定该 SQL 数据库的用户名和密码文件。**

```
% dpconf set-jdbc-data-source-prop mysql1 db-pwd-file:sqlpwd.txt db-user:root
```

3 **重新启动代理服务器。**

```
% dpadm restart /local/dps
```

4 **启用数据源，并允许对该数据源执行写入操作。**

```
% dpconf set-jdbc-data-source-prop mysql1 is-enabled:true is-read-only:false
```

5 **创建名为 `mysql1–pool` 的 JDBC 数据源池。**

```
% dpconf create-jdbc-data-source-pool mysql1-pool
```

6 **将 JDBC 数据源连接到此数据源池。**

```
% dpconf attach-jdbc-data-source mysql1-pool mysql1
```

7 **为数据源池创建名为 `myjdbc1–view` 且基 DN 为 `o=sql` 的 JDBC 数据视图。**

```
% dpconf create-jdbc-data-view mysql1-view mysql1-pool o=sql
```

8 **为 MySQL 数据库中的每个表创建 JDBC 表。**

```
% dpconf create-jdbc-table employee1 EMPLOYEE
% dpconf create-jdbc-table country1 COUNTRY
% dpconf create-jdbc-table phone1 PHONE
```

SQL 数据库中的表名区分大小写。请确保您使用的大小写与 SQL 数据库中使用的相同。

9 **为每个表中的每一列创建 JDBC 属性。**

创建 JDBC 属性会将 MySQL 列映射到 LDAP 属性。

```
% dpconf add-jdbc-attr employee1 uid ID
% dpconf add-jdbc-attr employee1 sn SURNAME
% dpconf add-jdbc-attr employee1 userPassword PASSWORD
% dpconf add-jdbc-attr employee1 room ROOM
% dpconf add-jdbc-attr phone1 tel NUMBER
% dpconf add-jdbc-attr country1 country NAME
```

不必为 `phone1 user_id` 和 `country1 id` 列创建 JDBC 属性，因为它们仅用于 MySQL 数据库环境。这些列没有相应的 LDAP 属性。

10 **为 LDAP `person` 对象类创建 JDBC 对象类。**

在此步骤中，`employee1` 表被标识为主表，而 `country1` 和 `phone1` 表被标识为从表。创建 JDBC 对象类时也需要 DN。在此示例中，DN 是通过数据视图的 `uid` 属性和基 DN 构建的。

```
% dpconf create-jdbc-object-class mysql1-view person employee1 country1 phone1 uid
```

11 **定义主表和从表之间的联接规则。**

联接规则在从表上进行定义，用于确定该表中的数据如何链接到主表数据。

```
% dpconf set-jdbc-table-prop country1 filter-join-rule:'ID=${EMPLOYEE.COUNTRY_ID}'
% dpconf set-jdbc-table-prop phone1 filter-join-rule:'USER_ID=${EMPLOYEE.ID}'
```

12 **指定 JDBC 对象类的超类。**

超类表示 JDBC 对象类从中继承属性的 LDAP 对象类。

```
% dpconf set-jdbc-object-class-prop mysql1-view person super-class:top
```

▼ 创建所需的 ACI

测试 JDBC 数据视图之前，必须先通过配置 ACI 启用对数据视图的写入访问权限。默认情况下将拒绝对非 LDAP 数据视图的写入访问。在此示例中，只需添加一个允许用户修改其密码的全局 ACI。

1 **以代理管理员身份向 JDBC 数据源中添加一个 ACI 池，并添加一个允许用户修改其条目的全局 ACI。**

```
% ldapmodify -p 1389 -D "cn=proxy manager" -w password
dn: cn=mysql1,cn=virtual access controls
changetype: add
objectclass: acisource
dpsaci: (targetattr="*") (target = "ldap:///o=sql") \
 (version 3.0; acl "enable all access for all users "; allow(all) userdn="ldap:///self";)
cn: mysql1
```

2 **创建一个连接处理程序，以处理到 o=sql 域的连接。**

```
% dpconf create-connection-handler mysql1-handler
```

3 **启用该连接处理程序，并将其配置为处理来自 o=sql 域用户的所有绑定。**

```
% dpconf set-connection-handler-prop mysql1-handler is-enabled:true \
 bind-dn-filters:"uid=.*,o=sql"
```

4 **将连接处理程序配置为使用之前添加的 ACI 池。**

```
% dpconf set-connection-handler-prop mysql1-handler aci-source:mysql1
```

▼ 测试 JDBC 数据视图

1 **以 o=sql 下的用户身份搜索 JDBC 数据源，以验证您是否可以读取数据视图中的内容。**

```
% ldapsearch -p 1389 -D "uid=kvaughan,o=sql" -w mypwd -b o=sql "objectclass=*"
```

注 – 必须使用 o=sql 下的用户凭证或匿名绑定。

---

2 **以 o=sql 下的用户身份修改 userPassword 属性，以验证您是否可以向数据视图中写入内容。**

```
% ldapmodify -p 1389 -D "uid=kvaughan,o=sql" -w mypwd
dn: uid=kvaughan,o=sql
changetype: modify
```

```
replace: userPassword
userPassword: myNewpwd
```

创建和测试联接数据视图

▼ 创建联接数据视图

1 **创建名为 myjoin1–view 的联接数据视图。**

将 LDAP 数据视图指定为主数据视图，将 JDBC 数据视图指定为从数据视图。

```
% dpconf create-join-data-view myjoin1-view myds1-view mysql1-view o=join
```

2 **在从数据视图上定义联接规则。**

以下联接规则指定从数据视图中条目的 uid 属性应该与主数据视图中条目的 uid 属性相匹配。

```
% dpconf set-jdbc-data-view-prop mysql1-view filter-join-rule:uid='${myds1-view.uid}'
```

3 **定义可以通过联接数据视图从主数据视图读取以及向主数据视图写入的属性集。**

```
% dpconf set-ldap-data-view-prop myds1-view viewable-attr:dn viewable-attr:cn \
 viewable-attr:sn viewable-attr:givenName viewable-attr:objectClass viewable-attr:ou \
 viewable-attr:l viewable-attr:uid viewable-attr:mail viewable-attr:telephoneNumber \
 viewable-attr:facsimileTelephoneNumber viewable-attr:roomNumber viewable-attr:userPassword
% dpconf set-ldap-data-view-prop myds1-view writable-attr:dn writable-attr:cn \
 writable-attr:sn writable-attr:givenName writable-attr:objectClass writable-attr:ou \
 writable-attr:l writable-attr:uid writable-attr:mail writable-attr:telephoneNumber \
 writable-attr:facsimileTelephoneNumber writable-attr:roomNumber writable-attr:userPassword
```

这些定义仅适用于**联接**视图环境。默认情况下，如果直接访问 LDAP 数据视图，则可以读取和写入所有属性。

4 **定义可以通过联接数据视图在从数据视图中读取以及向从数据视图写入的属性集。**

```
% dpconf set-jdbc-data-view-prop mysql1-view viewable-attr:dn viewable-attr:objectclass \
 viewable-attr:sn viewable-attr:room viewable-attr:userpassword viewable-attr:jobtitle \
 viewable-attr:country viewable-attr:tel
% dpconf set-jdbc-data-view-prop mysql1-view writable-attr:dn writable-attr:objectclass \
 writable-attr:sn writable-attr:room writable-attr:userpassword writable-attr:jobtitle \
 writable-attr:country writable-attr:tel
```

这些定义仅适用于**联接**视图环境。默认情况下，如果直接访问 JDBC 数据视图，则可以读取和写入所有属性。

▼ 创建所需的 ACI

1 **以代理管理员身份添加一个允许匿名访问联接数据视图的全局 ACI。**

```
% ldapmodify -p 1389 -D "cn=proxy manager" -w password
dn: cn=myjoin1,cn=virtual access controls
changetype: add
objectclass: acisource
dpsaci: (targetattr="*") (target = "ldap:///o=join") \
 (version 3.0; acl "anonymous_access"; allow(all) userdn="ldap:///anyone";)
cn: myjoin1
```

2 **创建一个连接处理程序，以处理到 o=join 域的连接。**

```
% dpconf create-connection-handler myjoin1-handler
```

3 **启用该连接处理程序，并将其配置为处理来自 o=join 下用户的所有绑定。**

```
% dpconf set-connection-handler-prop myjoin1-handler is-enabled:true \
 bind-dn-filters:"uid=.*,ou=people,o=join"
```

4 **将连接处理程序配置为使用之前添加的 ACI 池。**

```
% dpconf set-connection-handler-prop myjoin1-handler aci-source:myjoin1
```

▼ 测试联接数据视图

1 **以匿名用户身份搜索联接数据视图。**

在此步骤中，我们将搜索 Kirsten Vaughan 的条目，以查看是否会检索来自两个联接视图的数据。

```
% ldapsearch -p 1389 -b o=join "uid=kvaughan"
```

请注意，返回的条目包括来自 LDAP 数据视图和 JDBC 数据视图的属性。

2 **以 o=join 下的用户身份修改 userPassword 属性，以验证您是否可以向联接数据视图中写入内容。**

```
% ldapmodify -p 1389 -D "uid=kvaughan,ou=people,o=join" -w myNewPassword
dn: uid=kvaughan,ou=people,o=join
changetype: modify
replace: userPassword
userPassword: myPassword
```

联接多个不同的数据源

此配置以 Example.com 组织为例，说明虚拟目录的某些功能如何满足该组织的特定目录服务要求。

数据存储方案

Example.com 将组织数据存储在多个不同的数据源中。由于过去的原因，用户数据分布在 LDAP 目录、平面 LDIF 文件和 SQL 数据库中。人力资源部门将用户数据存储在基 DN 为 `o=example.com` 的 LDAP 目录中。薪酬部门将数据存储在 SQL 数据库中。管理部门将管理数据（如部门和大楼编号）存储在基 DN 为 `dc=example,dc=com` 的 LDIF 文件中。

此外，Example.com 还收购了一个名为 Company22 的公司。Company 22 也将其用户数据存储在基 DN 为 `dc=company22,dc=com` 的 LDAP 目录中。

下图提供了说明 Example.com 用户数据存储方式的高级视图。

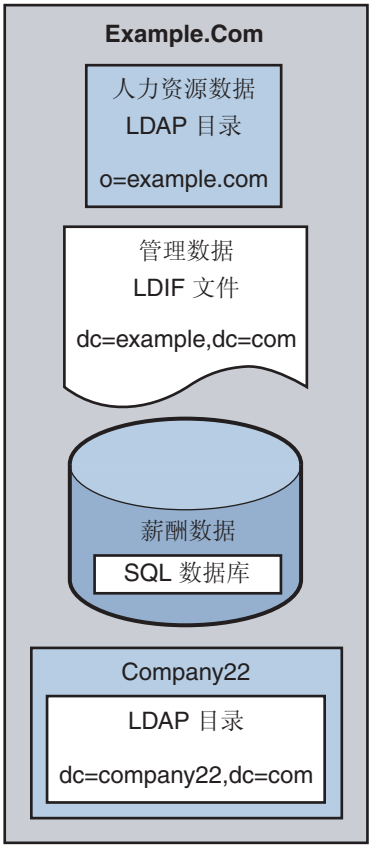


图 24–2 不同源中的数据存储

客户端应用程序要求

Example.com 有几个 LDAP 客户端应用程序需要访问存储在不同数据源中的数据。这些客户端应用程序的要求不尽相同。因此需要不同的数据视图。在某些情况下，客户端需要聚合数据。另外，某些客户端应用程序还需要访问 Company22 的用户数据，以便能够同时管理 Example.com 的新老员工。

下图提供了 Example.com 客户端应用程序要求的高级视图。

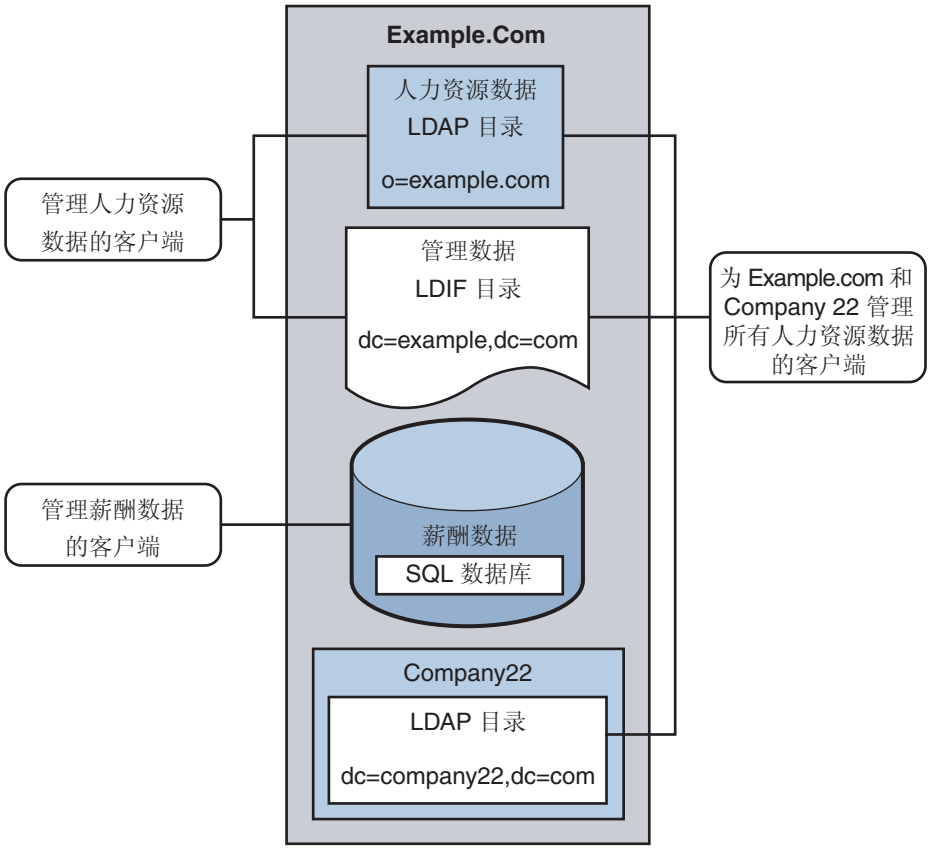


图 24–3 客户端应用程序要求

以下部分提供了充分的目录代理服务器数据视图的配置示例，以满足此样例方案中所述的客户端应用程序要求。有关数据视图的工作方式的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的第 22 章 “Directory Proxy Server LDAP Data Views”和《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的第 23 章 “Virtual Data Views”。

样例方案的配置可分为以下几部分：

聚合人力资源部门 LDAP 目录和管理部门 LDIF 文件中的数据

人力资源部门存储员工姓名、入职数据和职务级别等信息。管理部门存储大楼代码和办公室编号等其他信息。处理人力资源数据的客户端应用程序需要访问来自这两个数据源的组合数据。这两个数据源有一个公用属性 `employeeNumber`，即每个条目中都具有该属性。

下图说明了此客户端应用程序的要求。

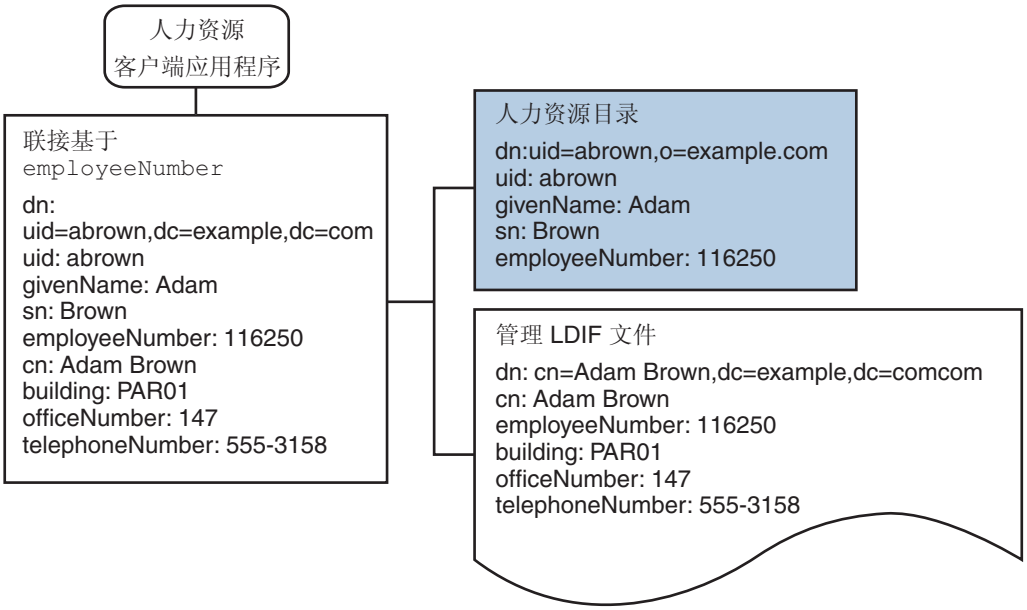


图 24–4 来自 LDAP 目录和 LDIF文件的数据聚合

要满足此应用程序要求，需要为薪酬目录和管理 LDIF 文件各创建一个数据视图。然后将这两个数据视图联接在一起，以提供对聚合数据的访问。目录代理服务器可以使用此公用属性聚合每个用户的数据。

为了简单起见，本部分中使用的命令假定运行环境如下：

- 目录代理服务器实例通过默认 LDAP 端口 (389) 在本地主机上运行。
- 目录代理服务器实例位于 `/local/myDPS` 中。
- 包含代理管理员密码的文件所在的路径已被设置为变量 `LDAP_ADMIN_PWF`。有关设置目录代理服务器环境变量的详细信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Installation Guide》中的“Environment Variables”。
- 薪酬部门的 LDAP 目录通过端口 `2389` 在名为 `payrollHost` 的主机上运行。
- 用于存储管理数据的 LDIF 文件名为 `example.ldif`。

要获取每个命令的完整语法，请运行不含任何选项的命令。例如：

```
$ dpconf create-ldap-data-view
Operands are missing
Usage: dpcfg create-ldap-data-view VIEW_NAME POOL_NAME SUFFIX_DN
```

▼ 为薪酬目录创建和启用 LDAP 数据视图

1. **为薪酬目录创建 LDAP 数据源。**

   ```
   $ dpconf create-ldap-data-source payroll-directory payrollHost:2389
   ```

2. **为薪酬目录创建 LDAP 数据源池。**

   ```
   $ dpconf create-ldap-data-source-pool payroll-pool
   ```

3. **将薪酬数据源连接到此数据源池。**

   ```
   $ dpconf attach-ldap-data-source payroll-pool payroll-directory
   ```

4. **为薪酬目录创建 LDAP 数据视图。**

   ```
   $ dpconf create-ldap-data-view payroll-view payroll-pool o=example.com
   ```

5. **启用 LDAP 数据视图，以便客户端请求可以路由到此数据视图。**

   ```
   $ dpconf set-ldap-data-view-prop payroll-view is-enabled:true
   ```

6. **重新启动目录代理服务器以使更改生效。**

   ```
   $ dpadm restart /local/myDPS
   ```

▼ 为管理数据创建和启用 LDIF 数据视图

1. **为管理数据创建 LDIF 数据视图。**

   ```
   $ dpconf create-ldif-data-view admin-view example.ldif dc=example,dc=com
   ```

2 **启用管理数据的 LDIF 数据视图。**

```
$ dpconf set-ldif-data-view-prop admin-view is-enabled:true
```

3 **指定管理视图包含可由薪酬视图中的多个条目使用的条目。**

```
$ dpconf set-ldif-data-view-prop admin-view contains-shared-entries:true
```

此属性设置为 TRUE 时，删除薪酬数据视图中的条目不会导致管理数据视图中的共享条目被删除。向薪酬数据视图添加条目只会将该条目添加到从数据视图（如果该条目尚不存在）。

4 **重新启动目录代理服务器以使更改生效。**

```
$ dpadm restart /local/myDPS
```

▼ 联接薪酬数据视图和管理数据视图

1 **在管理数据视图上创建过滤器联接规则，该规则指定数据的聚合方式。**

以下联接规则指定应该根据用户条目的 `employeeNumber` 属性联接数据。

```
$ dpconf set-ldif-data-view-prop admin-view filter-join-rule:'employeeNumber=\${payroll-view.employeeNumber}'
```

2 **创建聚合这两个数据视图的联接数据视图。**

对于该联接数据视图，组织将使用后缀 DN `dc=example,dc=com`。

```
$ dpconf create-join-data-view example-join-view payroll-view admin-view dc=example,dc=com
```

通过重命名 DN 将 Company 22 的数据添加到 Example.Com 的 DIT 中

Company 22 的用户数据存储在 DN `dc=company22,dc=com` 下。虽然在大多数情况下 Example.com 希望单独保留此用户数据，但一个客户端应用程序需要同时管理 Company 22 员工和其他 Example.com 员工。此客户端应用程序要求 Company 22 的用户数据**类似于** Example.com 数据。

下图说明了此客户端应用程序的要求。

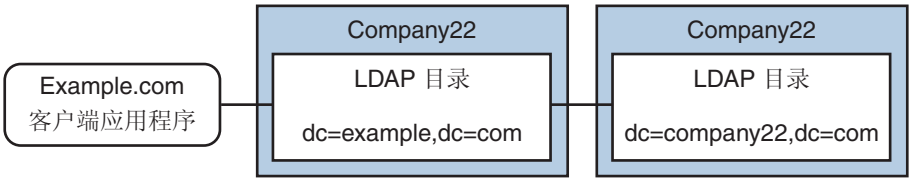


图 24–5 DN 重命名

要满足此应用程序要求，需要为 Company 22 的目录创建一个虚拟 DN 为 `dc=example,dc=com` 的数据视图。

为了简单起见，本部分中使用的命令假定运行环境如下：

- 目录代理服务器实例通过默认 LDAP 端口 (389) 在本地主机上运行。
- 目录代理服务器实例位于 `/local/myDPS` 中。
- 包含代理管理员密码的文件所在的路径已被设置为变量 `LDAP_ADMIN_PWF`。有关设置目录代理服务器环境变量的详细信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Installation Guide》中的“Environment Variables”。
- Company 22 LDAP 目录通过端口 `2389` 在名为 `company22Host` 的主机上运行。

▼ 使用虚拟 DN 为 Company 22 的目录创建数据视图

1. **为 Company 22 的目录创建 LDAP 数据源。**

```
$ dpconf create-ldap-data-source company22-directory company22Host:2389
```

2. **为 Company 22 的目录创建 LDAP 数据源池。**

```
$ dpconf create-ldap-data-source-pool company22-pool
```

3. **将 Company 22 的数据源连接到此数据源池。**

```
$ dpconf attach-ldap-data-source company22-pool company22-directory
```

4. **使用虚拟 DN `dc=example,dc=com` 为 Company 22 的目录创建 LDAP 数据视图。**

```
$ dpconf create-ldap-data-view company22-view company22-pool dc=example,dc=com
```

5. **指示目录代理服务器将此虚拟 DN 映射到 Company 22 目录中的实际 DN。**

```
$ dpconf set-ldap-data-view-prop company22-view dn-mapping-source-base-dn:dc=company22,dc=com
```

6. **启用 Company 22 目录的 LDAP 数据视图，以便客户端请求可以路由到此数据视图。**

```
$ dpconf set-ldap-data-view-prop company22-view is-enabled:true
```

7. **重新启动目录代理服务器以使更改生效。**

```
$ dpadm restart /local/myDPS
```

将 Company 22 数据添加到人力资源数据

人力资源部门需要人力资源数据（来自 Example.com 和最近收购的 Company 22）的聚合视图。下图说明了人力资源部门全局应用程序的要求。

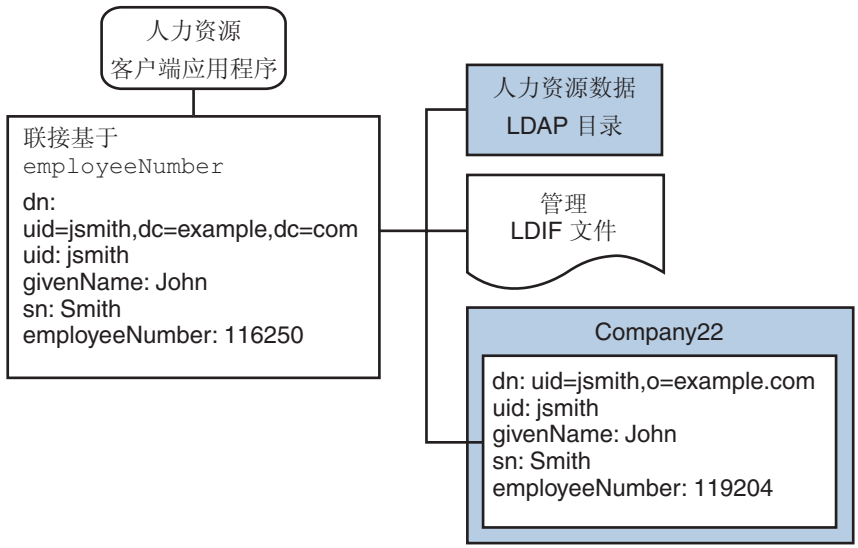


图 24–6 来自联接数据视图和 LDAP 数据视图的数据聚合

▼ 联接示例联接数据视图和 Company 22 数据视图

1 **在 Company 22 数据视图上创建过滤器联接规则，该规则指定数据的聚合方式。**

以下联接规则指定应该根据用户条目的 `employeeNumber` 属性联接数据。

```
$ dpconf set-ldif-data-view-prop company22-view filter-join-rule:'employeeNumber=\${example-join-view.employeeNumber}'
```

2 **创建联接数据视图，该视图将 Company 22 的数据视图和 Example.com 的联接数据视图聚合在一起。**

```
$ dpconf create-join-data-view global-join-view example-join-view company22-view dc=example,dc=com
```

使 LDAP 客户端可以访问 SQL 数据库中的薪酬数据

Example.com 的薪酬部门将薪水数据存储在 SQL 数据库中。该数据库有两个表，即 `employee` 表和 `salary` 表。Example.com 具有需要访问这些数据的 LDAP 客户端应用程序。该客户端应用程序要求 SQL 数据**类似于** LDAP 数据。

下图说明了此客户端应用程序的要求。

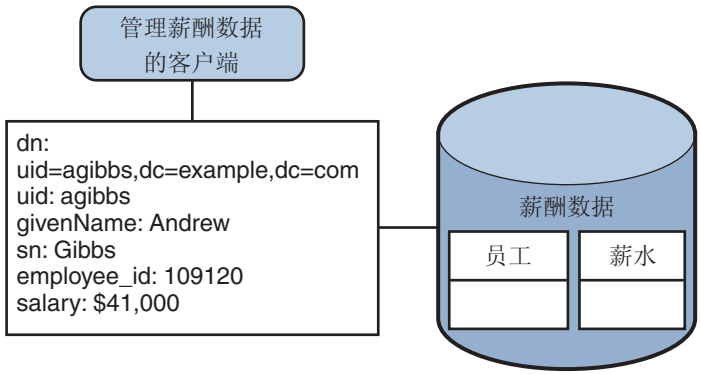


图 24–7 提供 SQL 数据库访问的 JDBC 数据视图

要满足此应用程序要求，需要创建一个将 SQL 表中的列映射到 LDAP 属性的 JDBC 数据视图。

为了简单起见，本部分中使用的命令假定运行环境如下：

- 目录代理服务器实例通过默认 LDAP 端口 (389) 在本地主机上运行。
- 目录代理服务器实例位于 `/local/myDPS` 中。
- 包含代理管理员密码的文件所在的路径已被设置为变量 `LDAP_ADMIN_PWF`。有关设置目录代理服务器环境变量的详细信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Installation Guide》中的“Environment Variables”。
- SQL 数据库已打开并正在运行。
- `JAVA_HOME` 变量已设置为正确的 Java 路径。
- SQL 数据库的密码为 `myPassword`。

▼ 为 Example.com 的薪酬数据库创建 JDBC 数据视图

1 为薪酬数据库创建 JDBC 数据源。

```
$ dpconf create-jdbc-data-source payroll-src myPassword
```

2 使用 SQL 数据库的属性配置 JDBC 数据源。

```
$ dpconf set-jdbc-data-source-prop payroll-src db-user:proxy
  db-pwd:myPassword
  db-url:jdbc:payrollsqldb:payrollsql://localhost
  driver-url:file://payrollsqldb.jar
  driver-class:org.payrollsqldb.jdbcDriver
```

3 启用 JDBC 数据源。

```
$ dpconf set-jdbc-data-source-prop payroll-src is-enabled:true
```

4 **为薪酬数据库创建 JDBC 数据源池。**

```
$ dpconf create-jdbc-data-source-pool payroll-pool
```

5 **将薪酬数据源连接到此数据源池。**

```
$ dpconf attach-jdbc-data-source payroll-pool payroll-src
```

6 **使用虚拟 DN** `o=payroll` **为薪酬数据库创建 JDBC 数据视图。**

```
$ dpconf create-jdbc-data-view payroll-view payroll-pool o=payroll
```

7 **为 SQL 数据库中的每个表创建 JDBC 表。**

```
$ dpconf create-jdbc-table jdbc-employee employee
$ dpconf create-jdbc-table jdbc-salary salary
```

8 **为 SQL 表中的每一列添加 JDBC 属性。**

```
$ dpconf add-jdbc-attr jdbc-employee eid employee_id
$ dpconf add-jdbc-attr jdbc-employee first firstname
$ dpconf add-jdbc-attr jdbc-employee last lastname
$ dpconf add-jdbc-attr jdbc-employee description description
$ dpconf add-jdbc-attr jdbc-employee spouse spousename
$ dpconf add-jdbc-attr jdbc-salary salary salary
$ dpconf add-jdbc-attr jdbc-salary social ssn
```

9 **指定可以通过 JDBC 数据视图查看和写入的属性。**

```
$ dpconf set-jdbc-data-view-prop payroll-view \
 viewable-attr:eid
 viewable-attr:first
 viewable-attr:last
 viewable-attr:desc
 viewable-attr:spouse
 viewable-attr:salary
 viewable-attr:social
 $ dpconf set-jdbc-data-view-prop payroll-view \
 writable-attr:eid
 writable-attr:first
 writable-attr:last
 writable-attr:description
 writable-attr:spouse
 writable-attr:salary
writable-attr:social
```

10 **创建一个映射到 LDAP 对象类的 JDBC 对象类。**

以下命令将创建一个映射到 LDAP `person` 对象类的对象类。该对象类指定应该将员工表用作主表，而将薪水表用作从表。`eid` 属性应用于构建 DN。

```
$ dpcfg create-jdbc-object-class payroll-view \
 person jdbc-employee jdbc-salary eid
```

11 在从表上创建过滤器联接规则，该规则指定从表中的数据应如何链接到主表数据。

以下联接规则指定应根据 `employee_id` 属性联接数据。

```
$ dpconf set-jdbc-table-prop jdbc-salary filter-join-rule:'employee_id=\${employee.employee_id}'
```

12 在 JDBC 对象类上创建超类。

```
$ set-jdbc-object-class-prop payroll-view person super-class:extensibleObject
```

添加虚拟访问控制

LDAP 目录上的访问控制是通过定义这些目录中的 ACI 来处理的。通过虚拟数据视图访问数据源时必须定义 ACI，这些 ACI 只适用于通过这些数据视图查看的数据。

所有通过目录代理服务器的访问都由**连接处理程序**进行控制。有关连接处理程序的信息，请参见第 25 章。

▼ 添加允许匿名访问的 ACI

1 添加 ACI。

```
$ ldapadd -v -D "cn=proxy manager" -w password -p 389
dn: cn=ldifonly-acis,cn=virtual access controls
objectclass: top
objectclass: aciSource
cn: ldifonly-acis
dpsaci: (targetattr="*")(version 3.0; acl "anonymous_access"; allow(all) (userdn="ldap:///anyone");)
```

2 将连接处理程序指向虚拟 ACI。

```
$ dpconf set-connection-handler-prop anonymous aci-source:ldifonly-acis
```

3 启用连接处理程序。

```
$ dpconf set-connection-handler-prop anonymous is-enabled:true
```

◆ ◆ ◆ 第 25 章

25 目录代理服务器连接处理程序

有关连接处理程序角色的概述，以及连接处理程序中所使用的条件和策略的描述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的第 21 章 “Directory Proxy Server Connection Handlers”。

本章包含以下主题：

创建、配置和删除连接处理程序

有关如何创建、配置和删除连接处理程序，以及如何配置数据视图相似性的信息，请参见以下过程。

▼ 创建连接处理程序

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1 **创建连接处理程序。**

```
$ dpconf create-connection-handler -h host -p port connection-handler-name
```

2 **（可选的）查看连接处理程序列表。**

```
$ dpconf list-connection-handlers -h host -p port
```

▼ 配置连接处理程序

开始之前 定义连接处理程序的属性时，必须参照为目录代理服务器实例定义的其他连接处理程序的属性。请考虑所有连接处理程序的属性，以确保这些属性指定了不同的条件集，并设置了正确的优先级。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1 查看连接处理程序的详细列表，以了解其主要属性和相对优先级。

```
$ dpconf list-connection-handlers -h host -p port -v
Name                        is-enabled  priority  description
--------------------------  ----------  --------  --------------------------
anonymous                   false       99        unauthenticated connections
default connection handler  true        100       default connection handler
```

创建目录代理服务器实例时，将创建 `anonymous` 和 `default connection handler` 连接处理程序。

2 查看一个连接处理程序的所有属性。

```
$ dpconf get-connection-handler-prop -h host -p port connection-handler-name
```

新连接处理程序的默认属性如下所示：

```
aci-source                    :  -
allowed-auth-methods          :  anonymous
allowed-auth-methods          :  sasl
allowed-auth-methods          :  simple
allowed-ldap-ports            :  ldap
allowed-ldap-ports            :  ldaps
bind-dn-filters               :  any
data-view-routing-custom-list :  -
data-view-routing-policy      :  all-routable
description                   :  -
domain-name-filters           :  any
enable-data-view-affinity     :  false
ip-address-filters            :  any
is-enabled                    :  false
is-ssl-mandatory              :  false
priority                      :  99
request-filtering-policy      :  no-filtering
resource-limits-policy        :  no-limits
schema-check-enabled          :  false
user-filter                   :  any
```

3 配置连接处理程序的优先级。

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name priority:value
```

优先级可以是 1 到 100 之间的任何数字，其中 1 为最高优先级。对于目录代理服务器实例，将按照优先级的顺序对连接处理程序进行评估。

4 **（可选的）指定连接处理程序的 DN 过滤属性。**

此属性允许您基于部分或全部绑定 DN 来控制访问。此属性的值为正则表达式。

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 bind-dn-filters:regular-expression
```

绑定 DN 过滤器采用 Java™ 正则表达式的格式。有关创建 Java 正则表达式的信息，请参见 http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html。

例如，要从 ou=people,dc=example,dc=com 下的用户向名为 secure-handler 的连接处理程序发送所有绑定，请按如下方式设置 bind-dn-filters 属性：

```
$ dpconf set-connection-handler-prop -h host1 -p 1389 secure-handler \
 bind-dn-filters:"uid=.*,ou=people,dc=example,dc=com"
```

5 **（可选的）指定要用于此连接处理程序的请求过滤策略的名称。**

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 request-filtering-policy:policy-name
```

其中 *policy-name* 是现有请求过滤策略的名称。有关如何创建和配置请求过滤策略的信息，请参见第 397 页中的 “创建和配置请求过滤策略和搜索数据隐藏规则”。

6 **（可选的）指定要用于此连接处理程序的资源限制策略的名称。**

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 resource-limits-policy:policy-name
```

其中 *policy-name* 是现有资源限制策略的名称。有关如何创建和配置资源限制策略的信息，请参见第 400 页中的 “创建和配置资源限制策略”。

7 **配置步骤 2 中列出的所有其他属性。**

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 property:value [property:value ...]
```

例如，将连接处理程序配置为只接受 SSL 连接。

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 is-ssl-mandatory:true
```

要获取某个属性的描述及其有效值列表，请运行以下命令：

```
$ dpconf help-properties connection-handler
```

8 **启用连接处理程序。**

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name is-enabled:true
```

9 **重新启动目录代理服务器实例以使更改生效（如有必要）。**

有关重新启动目录代理服务器的信息，请参见第 302 页中的 “重新启动目录代理服务器”。

▼ 删除连接处理程序

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 **（可选的）查看连接处理程序列表。**

```
$ dpconf list-connection-handlers -h host -p port
```

2 **删除一个或多个连接处理程序。**

```
$ dpconf delete-connection-handler -h host -p port connection-handler-name [connection-handler-name ... ]
```

▼ 配置数据视图的相似性

将连接分配给某个连接处理程序时，连接上的请求对于为该连接处理程序配置的数据视图列表是公开的，或者对于所有已配置的数据视图是公开的。该连接上的后续请求只对用于第一个请求的数据视图公开。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 **启用数据视图的相似性。**

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 enable-data-view-affinity:true
```

2 **（可选的）将连接处理程序配置为将请求路由到数据视图的自定义列表。**

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name data-view-routing-policy:custom
```

3 **（可选的）配置数据视图列表。**

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 data-view-routing-custom-list:view-name [data-view-routing-custom-list:view-name ...]
```

创建和配置请求过滤策略和搜索数据隐藏规则

有关请求过滤策略的概述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Request Filtering Policies for Connection Handlers”。有关搜索数据隐藏规则的概述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Search Data Hiding Rules in the Request Filtering Policy”。

有关如何创建和配置请求过滤策略和搜索数据隐藏规则的信息，请参见以下过程。

▼ 创建请求过滤策略

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1 **创建请求过滤策略。**

```
$ dpconf create-request-filtering-policy policy-name
```

2 **将请求过滤策略与某个连接处理程序关联。**

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 request-filtering-policy:policy-name
```

▼ 配置请求过滤策略

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1 **查看请求过滤策略的属性。**

```
$ dpconf get-request-filtering-policy-prop -h host -p port policy-name
```

请求过滤策略的默认属性如下所示：

```
allow-add-operations                :  true
allow-bind-operations               :  true
allow-compare-operations            :  true
allow-delete-operations             :  true
allow-extended-operations           :  true
allow-inequality-search-operations  :  true
allow-modify-operations             :  true
allow-rename-operations             :  true
allow-search-operations             :  true
allowed-comparable-attrs            :  all
allowed-search-scopes               :  base
allowed-search-scopes               :  one-level
```

```
allowed-search-scopes               :  subtree
allowed-subtrees                    :  ""
description                         :  -
prohibited-comparable-attrs         :  none
prohibited-subtrees                 :  none
```

2 通过设置步骤 1 中列出的一个或多个属性来配置请求过滤策略。

```
$ dpconf set-request-filtering-policy-prop -h host -p port policy-name \
  property:value [property:value ...]
```

通过设置步骤 1 中列出的属性，可以配置请求过滤策略的以下功能：

- 允许客户端执行的操作类型
- 对客户端公开或隐藏的子树
- 搜索操作的范围
- 搜索过滤器的类型
- 可以或无法在搜索和比较操作中进行比较的属性类型

▼ 创建搜索数据隐藏规则

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 为请求过滤策略创建一个或多个搜索数据隐藏规则。

```
$ dpconf create-search-data-hiding-rule -h host -p port policy-name rule-name \
 [rule-name ...]
```

2 查看搜索数据隐藏规则的属性。

```
$ dpconf get-search-data-hiding-rule-prop policy-name rule-name
```

搜索数据隐藏规则的默认属性如下所示：

```
attrs                           :  -
rule-action                     :  hide-entry
target-attr-value-assertions    :  -
target-dn-regular-expressions   :  -
target-dns                      :  -
```

3 通过设置步骤 2 中列出的一个或多个属性来配置搜索数据隐藏规则。

```
$ dpconf set-search-data-hiding-rule-prop -h host -p port policy-name rule-name \
  property:value [property:value ...]
```

可以使用以下规则操作之一：

`hide-entry` 不返回目标条目。

`hide-attributes` 返回目标条目，但过滤掉指定的属性。

`show-attributes`	返回目标条目，但过滤掉未指定的属性。

此规则可应用于以下条目：

`target-dns`	具有指定 DN 的条目
`target-dn-regular-expressions`	具有指定 DN 模式的条目
`target-attr-value-assertions`	具有指定属性名称和属性值对 (*attrName#attrValue*) 的条目

以下配置定义了用于隐藏 `inetorgperson` 类型条目的搜索数据隐藏规则。

```
$ dpconf set-search-data-hiding-rule-prop -h host1 -p port my-policy my-rule \
  target-attr-value-assertions:objectclass#inetorgperson
```

请求过滤策略和搜索数据隐藏规则的示例

以下示例包含请求过滤策略和搜索数据隐藏规则。当请求过滤策略与搜索数据隐藏规则相结合时，将限制对数据的访问，如下所示：

- 不允许执行以下类型的操作：添加、删除、扩展、修改和重命名。
- 只能访问 `ou=people,dc=sun,dc=com` 子树。
- 搜索操作只能返回 `inetorgperson` 类型的条目。

示例 25–1 请求过滤策略的样例

```
allow-add-operations                :  false
allow-bind-operations               :  true
allow-compare-operations            :  true
allow-delete-operations             :  false
allow-extended-operations           :  false
allow-inequality-search-operations  :  true
allow-modify-operations             :  false
allow-rename-operations             :  false
allow-search-operations             :  true
allowed-comparable-attrs            :  all
allowed-search-scopes               :  base
allowed-search-scopes               :  one-level
allowed-search-scopes               :  subtree
allowed-subtrees                    :  ou=people,dc=sun,dc=com
description                         :  myRequestFilteringPolicy
prohibited-comparable-attrs         :  none
prohibited-subtrees                 :  none
```

示例 25–2 搜索数据隐藏规则的样例

```
attrs                              :  -
rule-action                        :  hide-entry
target-attr-value-assertions       :  objectclass:inetorgperson
target-dn-regular-expressions      :  -
target-dns                         :  -
```

创建和配置资源限制策略

有关资源限制策略的概述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Resource Limits Policies for Connection Handlers”。有关如何创建和配置资源限制策略，以及如何自定义搜索限制的信息，请参见以下过程。

▼ 创建资源限制策略

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 **创建资源限制策略。**

```
$ dpconf create-resource-limits-policy -h host -p port policy-name
```

有关如何修改资源限制策略的属性的信息，请参见第 400 页中的 “配置资源限制策略”。

2 **将资源限制策略与某个连接处理程序关联。**

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 resource-limits-policy:policy-name
```

▼ 配置资源限制策略

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 **查看资源限制策略的属性。**

```
$ dpconf get-resource-limits-policy-prop -h host -p port policy-name
```

资源限制策略的默认属性如下所示：

```
description                              :  -
max-client-connections                   :  unlimited
```

```
max-connections                              :  unlimited
max-simultaneous-operations-per-connection   :  unlimited
max-total-operations-per-connection          :  unlimited
minimum-search-filter-substring-length       :  unlimited
referral-bind-policy                         :  default
referral-hop-limit                           :  default
referral-policy                              :  default
search-size-limit                            :  unlimited
search-time-limit                            :  unlimited
```

2 **通过设置步骤 1 中列出的一个或多个属性来配置资源限制策略：**

```
$ dpconf set-resource-limits-policy-prop -h host -p port policy-name \
  property:value [property:value ...]
```

▼ 自定义搜索限制

根据搜索基和搜索范围，可以定义搜索操作的自定义限制。如果搜索操作的目标 DN 和搜索范围与指定条件相匹配，则会限制搜索结果的最大大小。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1 **创建一个或多个自定义搜索限制。**

```
$ dpconf create-custom-search-size-limit -h host -p port policy-name \
  custom-search-limit-name [custom-search-limit-name ...]
```

2 **为自定义搜索限制设置条件。**

```
$ dpconf set-custom-search-size-limit-prop -h host -p port policy-name \
  custom-search-limit-name one-level-search-base-dn:value subtree-search-base-dn:value
```

3 **对返回的结果（当搜索符合步骤 2 中的条件之一时）数设置限制。**

```
$ dpconf set-custom-search-size-limit-prop -h host -p port policy-name \
 custom-search-limit-name search-size-limit:value
```

4 **查看自定义搜索限制的属性。**

```
$ dpconf get-custom-search-size-limit-prop -h host -p port policy-name \
  custom-search-limit-name
```

自定义搜索限制的默认属性如下所示：

```
one-level-search-base-dn  :  -
search-size-limit         :  unlimited
subtree-search-base-dn    :  -
```

将目录代理服务器配置为基于连接的路由器

Directory Proxy Server 5.2 是基于连接的路由器。在 Directory Proxy Server 5.2 中，客户端连接将被路由到特定的目录服务器。该客户端连接上的所有请求都将发送到相同的目录服务器，直到连接断开或客户端解除绑定为止。

Directory Proxy Server 6.0 是基于操作的路由器。但是，为了实现兼容性，可以将此目录代理服务器版本配置为基于连接的路由器，如以下过程所述。

▼ 将目录代理服务器配置为基于连接的路由器

1. **创建并配置一个或多个连接处理程序，如第 393 页中的“ 创建、配置和删除连接处理程序 ”中所述。**

   还可以使用默认的连接处理程序。

2. **将所有连接处理程序配置为只将请求路由到根数据视图。**

   例如：

   ```
   $ dpconf set-connection-handler-prop -h host1 -p 1389 myConnectionHandler \
    data-view-routing-policy:custom data-view-routing-custom-list:"root data view"
   ```

3. **为每个后端 LDAP 服务器创建并配置数据源，如第 317 页中的“ 创建和配置 LDAP 数据源 ”中所述。**

   例如：

   ```
   $ dpconf create-ldap-data-source -h host1 -p 1389 myDataSource host2:2389
   ```

4. **创建并配置数据源池，如第 319 页中的“ 创建和配置 LDAP 数据源池 ”中所述。**

   例如：

   ```
   $ dpconf create-ldap-data-source-pool -h host1 -p 1389 myDataSourcePool
   ```

5. **将所有数据源连接到数据源池，如第 320 页中的“ 将 LDAP 数据源连接到数据源池 ”中所述。**

   例如，

   ```
   $ dpconf attach-ldap-data-source -h host1 -p 1389 myDataSourcePool myDataSource
   ```

6. **将每个数据源配置为使用绑定重放来验证客户端，如第 327 页中的“ 使用绑定重放转发请求 ”中所述。**

   例如：

   ```
   $ dpconf set-ldap-data-source-prop -h host1 -p 1389 myDataSource \
    client-cred-mode:use-client-identity
   ```

7 配置客户端连接和数据源池之间的相似性，如第 340 页中的“ 配置客户端相似性 ”中所述。

例如：

```
$ dpconf set-ldap-data-source-pool-prop -h host1 -p 1389 myDataSourcePool \
 enable-client-affinity:true client-affinity-policy:read-write-affinity-after-write
```

◆ ◆ ◆ 第 26 章

客户端和目录代理服务器之间的连接

有关客户端和目录代理服务器之间的连接的概述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的第 20 章 “Connecting Clients to Directory Proxy Server”。

本章包含以下主题：

配置客户端和目录代理服务器之间的侦听器

目录代理服务器提供了安全侦听器和非安全侦听器，用于与客户端进行通信。有关目录代理服务器的侦听器的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Directory Proxy Server Client Listeners”。本部分介绍如何配置侦听器。

▼ 配置客户端和目录代理服务器之间的侦听器

注 – 此过程将配置客户端和目录代理服务器之间的非安全侦听器。要配置安全侦听器，请执行同一过程，但要将 `ldap` 替换为 `ldaps`。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。在 DSCC 中，可以在“性能”选项卡上配置此属性。

1 查看非安全侦听器的属性。

```
$ dpconf get-ldap-listener-prop -h host -p port
```

非安全侦听器的默认属性如下：

```
connection-idle-timeout          :  1h
connection-read-data-timeout     :  2s
connection-write-data-timeout    :  1h
is-enabled                       :  true
listen-address                   :  0.0.0.0
listen-port                      :  port-number
max-connection-queue-size        :  128
max-ldap-message-size            :  unlimited
number-of-threads                :  2
use-tcp-no-delay                 :  true
```

2 **根据需求更改步骤 1 中列出的一个或多个属性。**

```
$ dpconf set-ldap-listener-prop -h host -p port property:new-value
```

例如，要禁用 host1 上运行的目录代理服务器实例的非安全端口，请运行以下命令：

```
$ dpconf set-ldap-listener-prop -h host1 -p 1389 is-enabled:false
```

注意 – 如果要使用非特权端口号，则必须以超级用户身份运行目录代理服务器。

要更改非安全端口号，请运行以下命令：

```
$ dpconf set-ldap-listener-prop -h host -p port listen-port:new-port-number
```

3 **重新启动目录代理服务器实例以使更改生效（如有必要）。**

对某些侦听器属性所做的更改需要重新启动服务器才能生效。如果必须重新启动服务器，`dpconf` 会向您发出警报。有关重新启动目录代理服务器的信息，请参见第 302 页中的 “重新启动目录代理服务器”。

验证目录代理服务器的客户端

默认情况下，目录代理服务器被配置为进行简单绑定验证。简单绑定验证不需要任何其他配置。

有关客户端和目录代理服务器之间的验证的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Client Authentication Overview”。有关如何配置验证的信息，请参见以下过程。

▼ 配置基于证书的验证

有关基于证书的客户端验证的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Configuring Certificates in Directory Proxy Server”。本部分介绍如何配置基于证书的验证。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

注 – 基于证书的验证只能通过 SSL 连接执行。

- **将目录代理服务器配置为要求客户端在建立 SSL 连接时提供证书。**

```
$ dpconf set-server-prop -h host -p port allow-cert-based-auth:require
```

▼ 配置匿名访问

有关匿名访问的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Anonymous Access”。有关如何将匿名客户端的标识映射到另一个标识的信息，请参见第 329 页中的 “以备用用户身份转发请求”。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **允许未经验证的用户执行操作。**

```
$ dpconf set-server-prop -h host -p port allow-unauthenticated-operations:true
```

▼ 将目录代理服务器配置为进行 SASL 外部绑定

有关 SASL 外部绑定的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Using SASL External Bind”。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1. **不允许执行未经验证的操作。**

```
$ dpconf set-server-prop -h host -p port allow-unauthenticated-operations:false
```

2. **要求客户端在建立连接时提供证书。**

```
$ dpconf set-server-prop -h host -p port allow-cert-based-auth:require
```

客户端将提供一个包含 DN 的证书。

3 **通过 SASL 外部绑定启用客户端验证。**

```
$ dpconf set-server-prop -h host -p port allow-sasl-external-authentication:true
```

4 **将目录代理服务器使用的标识配置为映射后端 LDAP 服务器上的客户端证书。**

```
$ dpconf set-server-prop -h host -p port cert-search-bind-dn:bind-DN \
 cert-search-bind-pwd-file:filename
```

5 **配置目录代理服务器所搜索的子树的基 DN。**

目录代理服务器将搜索子树，以查找映射到客户端证书的用户条目。

```
$ dpconf set-server-prop -h host -p port cert-search-base-dn:base-DN
```

6 **将客户端证书中的信息映射到 LDAP 服务器上的证书。**

a. **对 LDAP 服务器上包含证书的属性进行命名。**

```
$ dpconf set-server-prop cert-search-user-attribute:attribute
```

b. **将客户端证书上的属性映射到 LDAP 服务器上包含证书的条目 DN。**

```
$ dpconf set-server-prop -h host -p port \
 cert-search-attr-mappings:client-side-attribute-name:server-side-attribute-name
```

例如，要将 DN 为 `cn=user1,o=sun,c=us` 的客户端证书映射到 DN 为 `uid=user1,o=sun` 的 LDAP 条目，请运行以下命令：

```
$ dpconf set-server-prop -h host1 -p 1389 cert-search-attr-mappings:cn:uid \
 cert-search-attr-mappings:o:o
```

7 **（可选的）将 SASL 外部绑定操作的请求路由到所有数据视图或数据视图的自定义列表。**

- 要将请求路由到所有数据视图，请运行以下命令：

```
$ dpconf set-server-prop -h host -p port cert-data-view-routing-policy:all-routable
```

- 要将请求路由到数据视图列表，请运行以下命令：

```
$ dpconf set-server-prop -h host -p port cert-data-view-routing-policy:custom \
 cert-data-view-routing-custom-list:view-name [view-name...]
```

◆ ◆ ◆ 第 27 章

27 目录代理服务器日志记录

目录代理服务器在访问日志和错误日志中记录信息。与目录服务器不同，目录代理服务器没有审计日志。有关目录代理服务器中的日志的描述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的第 19 章 “Directory Proxy Server Logging”。

本章包含以下主题：

查看目录代理服务器日志

可以通过日志文件或使用目录服务控制中心 (Directory Service Control Center, DSCC) 直接查看目录代理服务器日志。

默认情况下，日志存储在以下目录中：

instance-path`/logs`

下图显示了 DSCC 上的目录代理服务器错误日志的屏幕捕获。

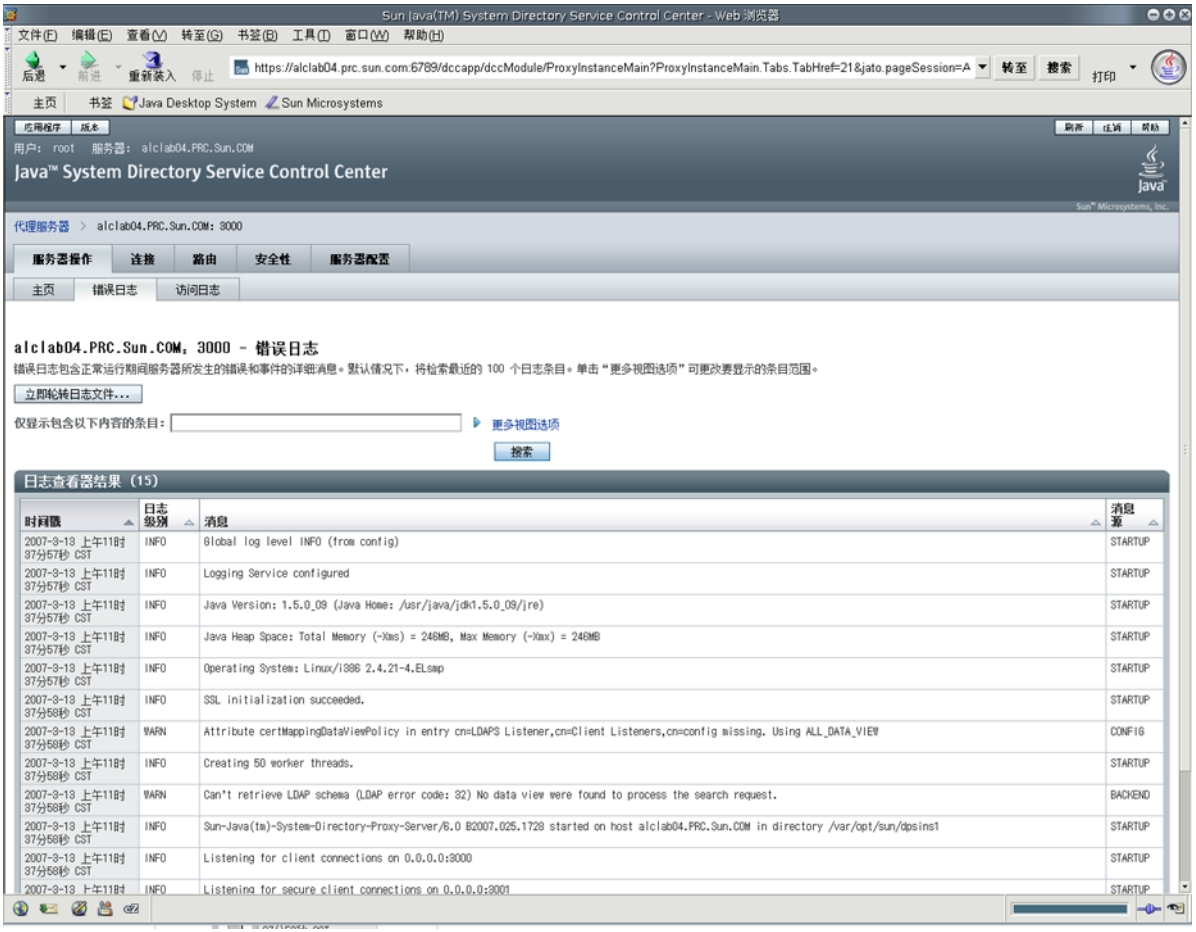


图 27–1 目录代理服务器的错误日志窗口

配置目录代理服务器日志

可以使用 dpconf 命令或 DSCC 配置目录代理服务器的错误日志和访问日志。有关如何使用 DSCC 配置日志的信息，请参见目录代理服务器联机帮助。本部分介绍如何使用 dpconf 命令配置目录代理服务器日志。

通过运行以下命令，可以检索完整的配置选项列表以及允许值和默认值：

```
$ dpconf help-properties error-log

$ dpconf help-properties access-log
```

▼ 配置目录代理服务器访问日志和错误日志

此过程将配置目录代理服务器访问日志。要配置目录代理服务器错误日志，请执行相同的过程，但要将 access 替换为 error。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1 查看访问日志的属性。

```
$ dpconf get-access-log-prop -h host -p port
```

访问日志的默认属性如下所示：

```
default-log-level                :  info
enable-log-rotation                 :  true
log-buffer-size                  :  9.8k
log-file-name                    :  logs/access
log-file-perm                    :  600
log-level-client-connections     :  -
log-level-client-disconnections  :  -
log-level-client-operations      :  -
log-level-connection-handlers    :  -
log-level-data-sources           :  -
log-level-data-sources-detailed  :  -
log-min-size                        :  100M
log-rotation-frequency           :  1h
log-rotation-policy              :  size
log-rotation-size                :  100M
log-rotation-start-day           :  1
log-rotation-start-time          :  0000
log-search-filters               :  false
max-age                          :  unlimited
max-log-files                    :  10
max-size                         :  unlimited
min-free-disk-space-size         :  1M
```

2 更改步骤 1 中列出的一个或多个属性。

```
$ dpconf set-access-log-prop -h host -p port property:value \
 [property:value ...]
```

例如，要将所有消息类别的默认日志级别设置为警告，请将 default-log-level 属性的值设置为 warning。

```
$ dpconf set-access-log-prop -h host1 -p 1389 default-log-level:warning
```

要禁用所有日志，而不考虑每个消息类别的日志级别，请将 default-log-level 属性的值设置为 none。

```
$ dpconf set-access-log-prop -h host1 -p 1389 default-log-level:none
```

要将特定的日志级别重置为默认日志级别，请将该日志级别的属性设置为 inherited。例如，要重置客户端连接的日志级别，请运行以下命令：

```
$ dpconf set-access-log-prop -h host1 -p 1389 log-level-client-connections:inherited
```

要获取有关可使用 set-access-log-prop 子命令设置的属性的信息，请键入：

```
$ dpconf help-properties access-log
```

配置目录代理服务器日志轮转

默认情况下，当日志文件大小达到 100 MB 时会轮转日志文件。默认情况下会保留十个日志文件，超过十个之后，轮转过程将开始覆盖最早的日志文件。本部分介绍如何配置目录代理服务器日志以执行计划的轮转、如何手动轮转日志，以及如何禁用日志轮转。要查看示例配置，请参见第 414 页中的 “日志轮转的示例配置”。

▼ 配置访问日志和错误日志的定期轮转

此过程将配置目录代理服务器访问日志。要配置目录代理服务器错误日志，请执行相同的过程，但要将 access 替换为 error。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1. **（可选的）查看访问日志的属性。**

   ```
   $ dpconf get-access-log-prop -h host -p port
   ```

2. **（可选的）查看访问日志属性的有效值。**

   ```
   $ dpconf help-properties access-log
   ```

3. **要在日志达到特定大小时轮转日志，请设置以下属性：**

   ```
   $ dpconf set-access-log-prop -h host -p port \
    log-rotation-policy:size log-rotation-size:maximum file size
   ```

   如果未指定最大文件大小的单位，则使用默认单位**字节**。当日志文件达到定义的大小时，将会轮转日志。文件大小至少为 1 MB，且不能超过 2 GB。

   有关如何按大小轮转日志的示例，请参见第 414 页中的 “根据日志大小轮转日志”。

4. **要定期轮转日志，而不考虑日志大小，请设置以下属性：**

```
$ dpconf set-access-log-prop -h host -p port \
 log-rotation-frequency:interval in months, weeks, hours, or minutes \
 log-rotation-policy:periodic \
 log-rotation-start-day:day in week (1-7) or day in the month (1-31) \
 log-rotation-start-time:time of day (hhmm)
```

如果将日志配置为在每月的 31 号轮转，但当月少于 31 天，则日志将在下个月的第一天轮转。

有关如何定期轮转日志的示例，请参见第 414 页中的 “根据时间轮转日志”。

5. **要在日志文件足够大时定期轮转日志，请设置 `log-rotation-frequency` 和 `log-min-size` 属性。**

```
$ dpconf set-access-log-prop -h host -p port \
 log-rotation-frequency:interval in months, weeks, hours, or minutes \
 log-rotation-policy:periodic log-min-size:minimum file size
 log-rotation-start-day:day in week (1-7) or day in the month (1-31) \
 log-rotation-start-time:time of day (hhmm)
```

`log-min-size` 属性表示日志的最小大小。只有日志文件大于指定大小时，才会在计划的时间执行轮转。

如果将日志配置为在每月的 31 号轮转，但当月少于 31 天，则日志将在下个月的第一天轮转。

有关如何在文件足够大时定期轮转日志的示例，请参见第 415 页中的 “根据时间和日志大小轮转日志”。

▼ 手动轮转访问日志文件和错误日志文件

此过程将轮转目录代理服务器访问日志。要轮转目录代理服务器错误日志，请执行相同的过程，但要将 `access` 替换为 `error`。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **轮转访问日志。**

```
$ dpconf rotate-log-now -h host -p port access
```

▼ 禁用访问日志轮转和错误日志轮转

此过程将禁用目录代理服务器访问日志轮转。要禁用目录代理服务器错误日志轮转，请执行相同的过程，但要将 `access` 替换为 `error`。

- **禁用日志文件轮转。**

  ```
  $ dpconf set-access-log-prop -h host -p port enable-log-rotation:false
  ```

日志轮转的示例配置

有关如何按日志大小和/或时间配置日志轮转的示例。

根据日志大小轮转日志

本部分中的示例说明如何只根据日志大小配置日志轮转。此配置将在日志大小达到 10 MB 时轮转日志，而不考虑上次轮转日志的时间。

```
$ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-policy:size \
  log-rotation-size:10M
```

根据时间轮转日志

本部分中的示例说明如何根据上次轮转日志的时间来配置日志轮转，而不考虑日志大小。

- 此配置将在今天 3:00 轮转日志，然后每 8 小时轮转一次，而不考虑日志文件的大小。

  ```
  $ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-frequency:8h \
   log-rotation-policy:periodic log-rotation-start-time:0300
  ```

- 此配置将在每天的 3:00、13:00 和 23:00 轮转日志，而不考虑日志文件的大小。由于 `log-rotation-start-time` 参数优先于 `log-rotation-frequency` 参数，因此日志在 23:00 轮转后，将于 4 小时后再次轮转。而不是在 23:00 轮转后间隔 10 小时再轮转。

  ```
  $ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-frequency:10h \
   log-rotation-policy:periodic log-rotation-start-time:0300
  ```

- 此配置将在星期一中午轮转日志，然后在每周的同一时间进行轮转，而不考虑日志文件的大小。

  ```
  $ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-frequency:1w \
   log-rotation-policy:periodic log-rotation-start-day:2 log-rotation-start-time:1200
  ```

- 此配置将在星期一中午轮转日志，然后每 3 天轮转一次，而不考虑日志文件的大小。

  ```
  $ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-frequency:3d \
   log-rotation-policy:periodic log-rotation-start-day:2 log-rotation-start-time:1200
  ```

  日志将在以下时间轮转：星期一、星期四、星期日、星期三，依此类推。请注意，`log-rotation-start-day` 参数仅应用于第一周。日志不会在第二周的星期一轮转。

- 此配置将在当月的 22 号中午轮转日志，然后在每月的同一时间进行轮转，而不考虑日志大小。

```
$ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-frequency:1m \
 log-rotation-policy:periodic log-rotation-start-day:22 \
 log-rotation-start-time:1200
```

如果将 `log-rotation-start-day` 设置为 31，而当月仅有 30 天，则日志将在下个月的第一天轮转。如果将 `log-rotation-start-day` 设置为 31，而当月仅有 28 天（二月），则日志将在下个月的 3 号轮转。

根据时间和日志大小轮转日志

此示例说明如何按指定的时间间隔配置日志轮转（当日志文件足够大时）。

此配置将在每天的 3:00、11:00 和 19:00 轮转日志（当日志文件大小超过 1 MB 时）。如果日志文件大小未超过 1 MB，则不会轮转日志文件。

```
$ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-frequency:8h \
 log-rotation-policy:periodic log-min-size:1M log-rotation-start-time:0300
```

删除目录代理服务器日志

目录代理服务器允许您根据时间、大小或可用磁盘空间（默认）配置日志删除。有关这些删除策略的详细信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Log File Deletion”。

以下过程将配置访问日志的日志删除。要配置错误日志的日志删除，请使用相同的命令，但要将 `access` 替换为 `error`。

▼ 根据时间配置访问日志和错误日志删除

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

- **指定日志文件的最长存留期。**

```
$ dpconf set-access-log-prop -h host -p port max-age:duration
```

其中 *duration* 包括天 (`d`)、周 (`w`) 或月 (`M`) 等单位。例如，要删除五天前的备份日志文件，请使用以下命令：

```
$ dpconf set-access-log-prop -h host1 -p 1389 max-age:5d
```

▼ 根据文件大小配置访问日志和错误日志删除

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

- **指定日志文件的最大大小。**

```
$ dpconf set-access-log-prop -h host -p port max-size:memory-size
```

例如，要删除大于 1 MB 的备份日志文件，请使用以下命令：

```
$ dpconf set-access-log-prop -h host1 -p 1389 max-size:1M
```

▼ 根据可用磁盘空间配置访问日志和错误日志删除

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

- **指定最小的可用磁盘空间。**

```
$ dpconf set-access-log-prop -h host -p port min-free-disk-space-size:memory-size
```

例如，要在可用磁盘空间小于 2 MB 时删除备份日志文件，请使用以下命令：

```
$ dpconf set-access-log-prop -h host1 -p 1389 min-free-disk-space-size:2M
```

将警报记录到 syslogd 守护进程

本部分介绍如何通过配置将警报消息记录到 syslogd 守护进程，以及如何将操作系统配置为接受 syslog 警报。

▼ 将目录代理服务器配置为将警报记录到 syslogd 守护进程

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1. **（可选的）查看系统日志警报属性的当前值。**

```
$ dpconf get-server-prop -h host -p port syslog-alerts-enabled \
 syslog-alerts-facility syslog-alerts-host
```

系统日志警报的默认属性如下所示：

```
syslog-alerts-enabled   :  false
syslog-alerts-facility  :  USER
syslog-alerts-host      :  localhost
```

syslog-alerts-host 属性定义消息所发送到的 syslogd 守护进程的主机名。
syslog-alerts-facility 属性为只读属性，可导致消息发送到系统日志的 user 类别。

2 **将警报消息记录到 syslogd 守护进程。**

```
$ dpconf set-server-prop -h host -p port syslog-alerts-enabled:true
```

3 **（可选的）将警报消息发送到其他主机上的 syslogd 守护进程。**

```
$ dpconf set-server-prop -h host -p port syslog-alerts-host:hostname
```

将操作系统配置为接受 syslog 警报

本部分提供了有关将 Solaris™、Linux 和 HP-UX 操作系统配置为接受 syslog 警报的说明。

▼ 将 Solaris 操作系统配置为接受 syslog 警报

1 **将相应的工具添加到 syslog 配置文件中。**

例如，要使用 USER 工具存储所有警报，请将以下行添加到 /etc/syslog.conf 中：

```
user.info       /var/adm/info
```

此处的 /var/adm/info 是要存储消息的本地目录示例。在继续操作之前，请确保存在 /var/adm/info。

2 **重新启动 syslogd 守护进程。**

a. **在 Solaris 8 和 Solaris 9 上，通过键入以下命令重新启动 syslogd：**

```
$ /etc/init.d/syslog stop | start
```

b. **在 Solaris 10 上，通过键入以下命令重新启动 syslogd：**

```
$ svcadm restart system/system-log
```

3 **验证是否已将消息记录到 syslog 中。**

```
$ logger -p user.info "Test message"
$ cat /var/adm/info
 Jun 19 17:18:38 host user: [ID 12345 user.info] Test message
```

▼ 将 Linux 配置为接受 syslog 警报

1. **将相应的工具添加到 syslog 配置文件中。**
   例如，要使用 USER 工具存储所有警报，请将以下行添加到 /etc/syslog.conf 中：

   ```
   user.info       /var/adm/info
   ```

   此处的 /var/adm/info 是要存储消息的本地目录示例。在继续操作之前，请确保存在 /var/adm/info。

2. **将 syslogd 守护进程配置为使用 -r 选项运行。**
   此选项允许 syslogd 接受来自网络的连接。默认情况下不设置 -r 选项。

   要设置 -r 选项，请将以下行添加到 /etc/sysconfig/syslog 中：

   ```
   SYSLOGD_OPTIONS="-m 0 -r"
   ```

   如果 /etc/sysconfig/syslog 不存在，请将相同的行添加到 /etc/init.d/syslog 中。

3. **重新启动 syslogd 守护进程。**

   ```
   $ /etc/init.d/syslog stop | start
   ```

4. **验证是否已将消息记录到 syslog 中。**

   ```
   $ logger -p user.info "Test message"
   $ cat /var/adm/info
    Jun 19 17:18:38 host user: [ID 12345 user.info] Test message
   ```

▼ 将 HP-UX 配置为接受 syslog 警报

1. **将相应的工具添加到 syslog 配置文件中。**
   例如，要使用 USER 工具存储所有警报，请将以下行添加到 /etc/syslog.conf 中：

   ```
   user.info       /var/adm/info
   ```

   此处的 /var/adm/info 是要存储消息的本地目录示例。在继续操作之前，请确保存在 /var/adm/info。

2. **重新启动 syslogd 守护进程。**

   ```
   $ /sbin/init.d/syslogd stop | start
   ```

3. **验证是否已将消息记录到 syslog 中。**

   ```
   $ logger -p user.info "Test message"
   $ cat /var/adm/info
    Jun 19 17:18:38 host user: [ID 12345 user.info] Test message
   ```

通过目录代理服务器和目录服务器访问日志跟踪客户端请求

要跟踪客户端请求的路径，您必须了解请求是如何记录到目录代理服务器访问日志和目录服务器访问日志中的。为了理解本部分内容，请先阅读《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Tracking Client Requests Through Directory Proxy Server and Directory Server Access Logs”。

▼ 跟踪从目录服务器经由目录代理服务器到客户端应用程序的操作

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1 找到您要在目录服务器访问日志中跟踪的操作的连接号。

例如，访问日志中的以下行显示连接号为 conn=12839 的操作 op=2 。

```
[20/Jul/2006:18:01:49 -0500] conn=12839 op=2 msgId=4 - SRCH base="dc=example,dc=com"
 scope=2 filter="(objectClass=organizationalunit)" attrs=ALL
```

2 获取该连接的目录代理服务器连接信息。

要获取此信息，请搜索目录服务器访问日志，以找到具有相应连接号的所有操作。例如，在 UNIX 系统上运行以下 grep 命令，可找到目录服务器访问日志中与连接 conn=12839 相对应的所有行：

```
$ grep conn=12839 access
```

显示初始 LDAP 连接的行就是您要查找的行，该行类似于以下内容：

```
[19/Jul/2006:16:32:51 -0500] conn=12839 op=-1 msgId=-1 - fd=27 slot=27
 LDAP connection from 129.153.160.175:57153 to 129.153.160.175
```

上面的行表明存在从 **129.153.160.175:57153** 到目录服务器的 LDAP 连接。端口号 (**57153**) 是将连接链接回目录代理服务器访问日志所需的信息。通过端口号可以查找目录代理服务器日志中相应的连接，并从此连接找到客户端信息。

如果日志文件自首次建立连接后已进行了轮转，则需要搜索已归档的日志文件和当前的访问日志文件。

3 找到目录代理服务器访问日志中相应的连接。

要获取此信息，请搜索目录代理服务器访问日志，以找到具有相应端口号的所有操作。

您可能会在日志文件中找到具有相同端口号的多个条目。要确保找到正确的条目，请在搜索中包含目录服务器日志条目的时间戳。

例如，在 UNIX 系统上运行以下 grep 命令，可找到与目录服务器日志中找到的时间戳和端口号相对应的连接条目。

```
$ grep 19/Jul/2006:16:32 access | grep 57153
```

请注意，考虑到服务器时间会存在一些细微差别，因此从时间戳中排除了**秒**值。

目录代理服务器日志中的相应行应类似于以下内容：

```
[19/Jul/2006:16:32:51 -0500] - SERVER_OP  - INFO  - Created BIND  LDAP connection
 s_conn=sunds-d1m1-9389:34 client=0.0.0.0:57153
 server=idm160.central.sun.com:9389 main
```

此行表明目录代理服务器创建了到 s_conn=sunds-d1m1-9389:34 的 BIND 连接。目录代理服务器将其自身标识为 TCP 端口 57153 上的客户端 client=0.0.0.0。

可从此日志行中提取的重要信息为服务器 ID 和端口号 (**s_conn=sunds-d1m1-9389:34**)。

4 找到与上一步中标识的服务器 ID 和端口号相对应的所有操作。

要获取此信息，请搜索目录代理服务器访问日志，以找到与服务器 ID 和端口号相对应的所有操作。

例如，在 UNIX 系统上运行以下 grep 命令，可找到与上一步中找到的服务器 ID 相对应的操作。

```
$ grep s_conn=sunds-d1m1-9389:34 access
```

在这种情况下，搜索时间戳将不起作用，因为这些操作可能会跨越数天。但是，您必须确定搜索返回的操作是否正确。如果存在多个 Create 连接语句，请确保找到与原始搜索语句相对应的语句。要执行此操作，请将此时间戳与步骤 1 中找到的时间戳进行比较。

从目录代理服务器访问日志中提取的以下内容显示了为 s_conn=sunds-d1m1-9389:34 返回的所有操作。

```
[19/Jul/2006:16:32:51 -0500] - SERVER_OP  - INFO - Created BIND LDAP connection
 s_conn=sunds-d1m1-9389:34 client=0.0.0.0:57153 server=idm160.central.sun.com:9389 main
[20/Jul/2006:18:01:49 -0500] - SERVER_OP  - INFO  - conn=31 op=0 BIND dn="cn=directory manager"
 method="SIMPLE" s_msgid=3 s_conn=sunds-d1m1-9389:34
[20/Jul/2006:18:01:49 -0500] - SERVER_OP  - INFO  - conn=31 op=0 BIND RESPONSE err=0 msg=""
 s_conn=sunds-d1m1-9389:34
[20/Jul/2006:18:01:49 -0500] - SERVER_OP  - INFO  - conn=31 op=1 SEARCH base="dc=example,dc=com"
 scope=2 s_msgid=4 s_conn=sunds-d1m1-9389:34
[20/Jul/2006:18:01:49 -0500] - SERVER_OP  - INFO  - conn=31 op=1 SEARCH RESPONSE err=0 msg=""
 nentries=1 s_conn=sunds-d1m1-9389:34
```

通过以上信息，您可以看到目录代理服务器上的此搜索操作的连接 ID 为 31 (conn=31)。

5 **找到与上一步中找到的连接 ID 相对应的客户端连接 IP 地址。**

要获取此信息，请搜索目录代理服务器访问日志，以找到具有正确连接 ID 和时间戳的所有操作。要使用的时间戳为步骤 1 中原始搜索语句中的时间戳。

例如，在 UNIX 系统上运行以下 `grep` 命令，可找到客户端连接 IP 地址：

```
$ grep "20/Jul/2006:18:01" access | grep conn=31
```

您所关注的行应类似于以下内容：

```
[20/Jul/2006:18:01:49 -0500] - CONNECT - INFO  - conn=31 client=129.150.64.156:2031
server=0.0.0.0:11389 protocol=LDAP
```

6 **确定在上一步中找到的 IP 地址的所有者。**

通过此信息，可以准确地确定在目录服务器上负责执行该操作的用户。

◆ ◆ ◆ 第 28 章

目录代理服务器监视和警报

监视功能会检测目录代理服务器和数据源的故障。

有关目录代理服务器监视框架的描述和 cn=monitor 条目的详细布局，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Monitoring Directory Proxy Server”。本章包含以下主题：

检索有关目录代理服务器的监视数据

要检索有关目录代理服务器的监视数据，请使用 cn=monitor 条目。此条目由目录代理服务器在本地内存数据库中进行管理。通过在 cn=monitor 条目上执行 LDAP 搜索，可以检索 cn=monitor 下的属性。您必须以代理管理员身份进行绑定才能搜索此条目。

有关使用 JVM 检索监视数据的信息，请参见第 427 页中的 “使用 JVM 检索有关目录代理服务器的监视数据”。

检索有关数据源的监视数据

有关目录代理服务器如何监视数据源运行状况的描述，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Reference》中的“Monitoring Data Sources”。本部分介绍如何配置数据源监视。

▼ 通过侦听错误监视数据源

在此类型的监视中，目录代理服务器侦听目录代理服务器和数据源之间的通信错误。此类型的监视称为被动监视，因为只有在检测到错误时目录代理服务器才会反应，而不会主动测试数据源。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1. **将数据源的监视模式设置为** `reactive`。

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port datasource monitoring-mode:reactive
   ```

2. **配置警报，以便在检测到错误或者数据源脱机或联机时发送警报，如第 425 页中的 “为目录代理服务器配置管理警报”中所述。**

▼ 通过定期建立专用连接来监视数据源

如果在指定的时间间隔内数据源没有收到请求或发出响应，目录代理服务器将创建一个到数据源的专用连接。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1. **将数据源的监视模式设置为** `proactive`。

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port datasource monitoring-mode:proactive
   ```

2. **设置在建立专用连接之前，目录代理服务器检测不到任何数据源活动的最长时间。**

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port datasource \
    monitoring-inactivity-timeout:time
   ```

   默认情况下，无活动的超时时间为 120 秒。

3. **配置警报，以便在检测到数据源处于脱机或联机状态时发送警报，如第 425 页中的 “为目录代理服务器配置管理警报”中所述。**

▼ 通过测试建立的连接来监视数据源

在此类型的监视中，目录代理服务器定期在每个数据源的各个连接上执行搜索。这样，目录代理服务器可检测到关闭的连接，从而防止连接因无活动而被断开。

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面”和 DSCC 联机帮助。

1. **将数据源的监视模式设置为** `proactive`。

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port datasource monitoring-mode:proactive
   ```

2. **配置目录代理服务器执行的监视搜索请求。**

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port datasource \
     monitoring-bind-timeout:timeout monitoring-entry-dn:dn \
     monitoring-search-filter:filter monitoring-entry-timeout:timeout
   ```

   搜索请求中使用以下属性：

`monitoring-bind-timeout`	目录代理服务器等待建立数据源连接的时间长度。默认情况下，此属性的值为 5 秒。
`monitoring-entry-dn`	搜索请求中目标条目的 DN。默认情况下，此属性为根 DSE 条目 (`""`)。
`monitoring-search-filter`	搜索过滤器。
`monitoring-entry-timeout`	目录代理服务器等待搜索响应的时间长度。默认情况下，此属性的值为 5 秒。

3. **设置轮询时间间隔。**

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port datasource monitoring-interval:interval
   ```

   如果连接中断，目录代理服务器将按此时间间隔对连接进行轮询，以检测其是否恢复。默认情况下，监视时间间隔为 30 秒。

4. **配置警报，以便在检测到数据源处于脱机或联机状态时发送警报，如****第 425 页中的“为目录代理服务器配置管理警报”****中所述。**

为目录代理服务器配置管理警报

有关如何配置管理警报的信息，请参见以下过程。

▼ 启用管理警报

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的“目录服务控制中心界面”和 DSCC 联机帮助。

1. **查看启用的警报。**

   ```
   % dpconf get-server-prop -h host -p port enabled-admin-alerts
   ```

2 **启用一个或多个管理警报。**

```
% dpconf set-server-prop -h host -p port enabled-admin-alerts:alert1 \
  [enabled-admin-alerts:alert2 ...]
```

例如，要启用所有可用警报，请运行以下命令：

```
% dpconf set-server-prop -h host -p port \
 enabled-admin-alerts:error-configuration-reload-failure-with-impact \
 enabled-admin-alerts:error-server-shutdown-abrupt \
 enabled-admin-alerts:info-configuration-reload \
 enabled-admin-alerts:info-data-source-available \
 enabled-admin-alerts:info-server-shutdown-clean \
 enabled-admin-alerts:info-server-startup \
 enabled-admin-alerts:warning-configuration-reload-failure-no-impact \
 enabled-admin-alerts:warning-data-source-unavailable \
 enabled-admin-alerts:warning-data-sources-inconsistent \
 enabled-admin-alerts:warning-listener-unavailable
```

要禁用所有警报，请运行以下命令：

```
% dpconf set-server-prop -h host -p port enabled-admin-alerts:none
```

默认情况下不启用任何警报。

另请参见 有关详细信息，请参见 `enabled-admin-alerts`(5dpconf)。

▼ 配置要发送到 syslog 的管理警报

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面 ” 和 DSCC 联机帮助。

1 **选择要发送到 `syslog` 守护进程的警报，如第 425 页中的 “ 启用管理警报 ” 中所述。**

2 **启用要发送到 `syslog` 守护进程的警报。**

```
$ dpconf set-server-prop -h host -p port syslog-alerts-enabled:true
```

所有警报都通过 `USER` 工具发送到 `syslog` 中。

3 **设置要将警报发送到的 `syslog` 守护进程的主机名。**

```
$ dpconf set-server-prop -h host -p port syslog_hostname:hostname
```

▼ 配置要发送到电子邮件的管理警报

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1. **选择将发送到 `syslog` 的警报，如第 425 页中的 “启用管理警报”中所述。**

2. **配置电子邮件的地址和特性。**

```
$ dpconf set-server-prop -h host -p port email-alerts-smtp-host:host-name \
  email-alerts-smtp-port:port-number \
  email-alerts-message-from-address:sender-email-address \
  email-alerts-message-to-address:receiver-email-address \
  [email-alerts-message-to-address:receiver-email-address ...] \
  email-alerts-message-subject:email-subject
```

3. **启用要发送到电子邮件的警报。**

```
$ dpconf set-server-prop -h host -p port email-alerts-enabled:true
```

4. **（可选的）设置标志，以便在电子邮件中包含警报代码。**

```
$ dpconf set-server-prop -h host -p port \
 email-alerts-message-subject-includes-alert-code:true
```

▼ 配置运行脚本的管理警报

可以使用 DSCC 执行此任务。有关信息，请参见第 41 页中的 “目录服务控制中心界面” 和 DSCC 联机帮助。

1. **选择要发送到 `syslog` 的警报，如第 425 页中的 “启用管理警报”中所述。**

2. **启用运行脚本的警报。**

```
$ dpconf set-server-prop -h host -p port scriptable-alerts-enabled:true
```

3. **设置将运行的脚本的名称。**

```
$ dpconf set-server-prop -h host -p port scriptable-alerts-command:script-name
```

使用 JVM 检索有关目录代理服务器的监视数据

目录代理服务器在 Java 虚拟机 (Java Virtual Machine, JVM) 中运行，并且依赖于 JVM 计算机的内存。要确保目录代理服务器正常运行，必须监视 JVM 计算机的内存使用情况。

有关如何调节 JVM 计算机参数的信息，请参见《Sun Java System Directory Server Enterprise Edition 6.0 Deployment Planning Guide》中的“Hardware Sizing For Directory Proxy Server”。

默认情况下，JVM 计算机的堆大小为 250 MB。如果目录代理服务器没有足够的物理内存，则堆大小可能小于 250 MB。

在目录代理服务器运行时，可以监视 JVM 计算机的堆大小，以确保不出现内存不足的情况。要执行此操作，请使用随 Java 开发工具包 (Java Development Kit, JDK) 提供的标准工具。这些工具位于以下目录中：`$JAVA_HOME/bin/jps` 和 `$JAVA_HOME/bin/jstat`。

▼ 查看 JVM 的堆大小

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

- **查看 JVM 的堆大小。**

```
$ dpadm get-flags instance-path jvm-args
jvm-args: -Xms250M  -Xmx250M
```

▼ 监视目录代理服务器运行时 JVM 的堆大小

无法使用 DSCC 执行此任务。请使用命令行，如以下过程所述。

1. **查看目录代理服务器实例的 PID。**

```
$ jps
```

2. **查看 JVM 计算机所使用的内存。**

```
$ jstat -gcutil PID
```

   - 如果零列接近 100%, 则说明 JVM 计算机没有足够的内存。
   - FGC 是全部垃圾收集 (garbage collection, GC) 事件的数目。垃圾收集会占用很大空间。
   - GCT（garbage collection time，垃圾收集时间）是 GC 所花费的时间。

索引