Sun Java™ Desktop System Configuration Manager Release 1

安装指南

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054,
U.S.A. 650-960-1300

部件号：817-5593-10

2004 年 4 月，修订版 A

版权和商标

目录

第 1 章

简介

Sun Java™ Desktop System Configuration Manager Release 1 旨在为运行 Sun Java™ Desktop System 的桌面主机提供集中配置。可以将设定指定给组织结构中的各种元素，从而使管理员能够以简单的方式管理多组用户或主机。它的主要组件有：

- 一个用于保存配置数据的 LDAP 服务器，含有要管理的用户和主机的组织结构；
- 一个基于万维网的管理工具，允许管理员为组织结构中的元素定义和指定配置数据；
- 安装在客户端主机上的桌面组件，这些桌面组件代表当前登录的用户检索配置数据，并且将这些数据提供给组成 Java Desktop System 的各个应用程序。

该管理工具是在 Sun Web Console 中执行的基于万维网的应用程序。允许管理员浏览 LDAP 服务器的组织结构并指定其元素的策略。这些策略按照策略文档样式进行显示和编辑，它们定义管理工具要处理的设定。

桌面组件以 Sun Java™ Desktop System Configuration Agent 为中心进行组织，后者代表用户从 LDAP 服务器中检索配置数据，并且将这些数据提供给多个配置系统适配器，由这些适配器使用策略设定对本地配置（应用程序提供的默认设定和用户设定）进行补充。目前支持的配置系统有 GConf（该系统处理 Gnome 桌面、Evolution 等 Gnome 应用程序的配置）、Mozilla 首选设置和 StarRegistry（StarSuite 的配置系统）。

第 2 章

LDAP 服务器

概念

在 Java Desktop System Configuration Manager 框架中，配置数据与实体关联，实体是 LDAP 系统信息库中的条目并与公司的组织结构的元素相对应。

可以识别的实体有：

- 组织：通常代表总体层次结构的组织（部门、组、团队）单元或地理（洲、国家/地区、位置）单元。
- 用户：代表总体层次结构的叶节点，如其字面含义所指，通常表示用户。
- 域：代表网络组织的逻辑结构单元。
- 主机：也代表总体层次结构的叶节点，但它表示的是网络中的计算机。
- 角色：代表属性，通常根据功能（管理员，站点管理）划分，适用于一组用户。

组织和用户实体用于定义用户树，而域和主机实体用于定义主机树。这两种树互相独立，但在框架中它们的处理方式类似。

组织和域实体与其他条目之间的关系由条目在系统信息库中的物理位置决定。也就是说，组织和域实体可以 包括树中位于这两种实体下的任何条目。角色与用户或主机之间的关系 由用户和主机条目的属性决定。

与实体相关的配置数据存储在由框架管理的特定条目中。这些条目通过其关联的服务名称和服务容器进行标识。

设置

要将现有的 LDAP 服务器和 Configuration Manager 一同使用，需要：

- 扩展服务器模式，以支持 Configuration Manager 用来存储配置数据的自定义对象程序类和属性。
- 在服务器中编辑并存储系统信息库中条目的映射信息以及 Configuration Manager 支持的实体的映射信息。

部署工具

要将现有的 LDAP 服务器与 Configuration Manager 一同使用，需要使用安装 CD 中的下列部署工具：

- `88apoc-registry.ldif` — 说明存储配置数据时所需的对象程序类和属性的模式文件。
- `OrganizationMapping` — 描述 LDAP 条目和 Configuration Manager 实体之间的映射关系的默认属性文件。
- `UserProfileMapping` — 描述 LDAP 用户条目属性和 Configuration Manager 用户初始配置属性之间的映射关系的默认属性文件。
- `createServiceTree` — 将映射文件存储到 LDAP 系统信息库中的脚本。
- `DeployApoc` — 扩展 LDAP 服务器模式和将映射文件存储到 LDAP 系统信息库中的脚本。

模式扩展

配置数据存储在条目树中，这些条目树被附加到数据的关联条目。在 LDAP 服务器上存储这些树使用的对象程序类和属性之前，必须向 LDAP 服务器模式加入对象和程序类。例如，提供的模式扩展文件以 LDIF 格式向 Sun Java System Directory Server 加入这些对象和程序类。要向其他 LDAP 服务器加入这些对象和程序类，需要使用这些服务器可以识别的格式。

组织映射

要定义 LDAP 条目和 Configuration Manager 实体之间的映射，必须编辑组织映射文件。必须为各个键提供与 LDAP 系统信息库的版式相匹配的值。

用户实体通过一个所有实体使用的对象程序类和一个其值在整个系统信息库中唯一的属性进行标识。可提供显示名称的格式，此格式将影响用户在管理应用程序中如何显示；且如果组织中用户条目使用容器条目，则可以选择定义该条目。键名及其默认值有：

```
# 所有用户条目使用的对象程序类
User/ObjectClass=inetorgperson
# 用户条目中在系统信息库内具有唯一值的属性
User/UniqueIdAttribute=uid
# 用户条目的组织条目中的可选容器，不使用时可以删除
User/Container=ou=People
# 管理应用程序中的显示名称的格式
User/DisplayNameFormat=sn, givenname
```

角色实体通过其可能使用的对象程序类列表和相应的命名属性列表进行标识。这些列表使用 <item1>,<item2>,...,<itemN> 的格式并且两者间必须对齐。也就是说，这些列表必须具有相同数目的项，并且第 n 个对象程序类必须与第 n 个命名属性一同使用。通过两个键定义了角色和用户之间的关系以及角色和主机之间的关系。VirtualMemberAttribute 键必须指定这样一个属性：其值能通过用户条目或主机条目查询。该键还必须含有条目所属角色的完整 DN。MemberAttribute 键必须指定用于搜寻筛选器的用户条目或主机条目属性。该键还必须含有用户或主机所属角色的完整 DN。VirtualMemberAttribute 键可以是服务类虚拟属性，而 MemberAttribute 键必须为可以在筛选器中使用的实际属性。键名及其默认值有：

```
# 角色的对象程序类列表
Role/ObjectClass=nsRoleDefinition
# 相应的命名属性的对齐列表
Role/NamingAttribute=cn
# 含有用户/主机角色的 DN 的实际属性（可在筛选器中使用）
Role/MemberAttribute=nsRoleDN
# 在用户或主机上查询时返回其所属角色的 DN 的属性
Role/VirtualMemberAttribute=nsRole
```

组织实体的标识方式与角色类似，采用的是对象程序类和相应命名属性的对齐列表。键名及其默认值有：

```
# 组织的对象程序类列表
Organization/ObjectClass=organization
# 相应的命名属性的对齐列表
Organization/NamingAttribute=o
```

域实体的标识方式与组织实体类似。键名及其默认值有：

```
# 域的对象程序类列表
Domain/ObjectClass=ipNetwork
# 相应的命名属性的对齐列表
Domain/NamingAttribute=cn
```

主机实体的标识方式与用户实体类似。键名及其默认值有：

```
# 所有主机条目使用的对象程序类
Host/ObjectClass=ipHost
# 主机条目中在系统信息库内具有唯一值的属性
Host/UniqueIdAttribute=cn
# 主机条目的域条目中的可选容器，不使用时可以删除
Host/Container=ou=Hosts
```

用户初始配置映射

要定义 LDAP 用户条目属性和 Configuration Manager 用户实体属性之间的映射，需要编辑用户初始配置映射文件。每个键对应一个 Configuration Manager 用户属性。像组织映射标识的那样，可以将键作为值指定给 用户条目中的属性名。User/DisplayNameFormat 设定中使用的属性必须在用户初始配置映射中指定。键名及其默认值有：

```
# inetOrgPerson.givenName
org.openoffice.UserProfile/Data/givenname = givenname
# person.sn
org.openoffice.UserProfile/Data/sn = sn
# inetOrgPerson.initials
org.openoffice.UserProfile/Data/initials = initials
# organizationalPerson.street
org.openoffice.UserProfile/Data/street = street,postalAddress,streetAddress
# organizationalPerson.l（城市）
org.openoffice.UserProfile/Data/l = l
# organizationalPerson.st（省/市/自治区）
```

```
org.openoffice.UserProfile/Data/st = st
# organizationalPerson.postalCode
org.openoffice.UserProfile/Data/postalcode = postalcode
# country.c（国家/地区）
org.openoffice.UserProfile/Data/c =
# organizationalPerson.o（公司）
org.openoffice.UserProfile/Data/o = o,organizationName
# 已过时 — 无对应的 LDAP 对象
org.openoffice.UserProfile/Data/position =
# organizationalPerson.title
org.openoffice.UserProfile/Data/title = title
# inetOrgPerson.homePhone
org.openoffice.UserProfile/Data/homephone = homephone
# organizationalPerson.telephoneNumber
org.openoffice.UserProfile/Data/telephonenumber = telephonenumber
# organizationalPerson.facsimileTelephoneNumber
org.openoffice.UserProfile/Data/facsimiletelephonenumber =
facsimiletelephonenumber,officeFax
# inetOrgPerson.mail
org.openoffice.UserProfile/Data/mail = mail
```

部署

编辑映射文件以反映 LDAP 系统信息库的状态后，就可以部署这些文件。如果 LDAP 服务器模式已含有所需的对象程序类和属性，则可直接执行 `createServiceTree` 脚本，否则必须执行 `deployApoc` 脚本。

`deployApoc` 脚本是设计用来和 Sun Java System Directory Server 一同使用的。它将所提供的模式扩展文件复制到适当的目录中，并重新启动 LDAP 服务器，然后调用 `createServiceTree` 脚本。它必须作为有权复制模式系统信息库中的文件并有权重新启动服务器的用户来执行，并且必须通过以下方法调用：

```
./deployApoc <directory server directory>
```

`<directory server directory>` 参数必须是 Directory Server 的安装目录的 `slapd-<server name>` 子目录的路径。假定安装目录使用了默认目录并且服务器名称为 myserver.mydomain，则该目录将为 `/var/Sun/mps/slapd-myserver.mydomain`。

无论直接调用还是通过 `deployApoc` 脚本调用，`createServiceTree` 脚本都会提示用户输入 `LDAP` 服务器的位置（主机名、端口号和基本 `DN`）和服务条目的定义（例如，服务名和服务容器）。然后该脚本在 LDAP 服务器中建立引导服务树并将映射文件存储到服务树中。它可以作为任何用户执行，调用方法为：

```
./createServiceTree
```

然后提示用户提供：

- 主机名（默认值：localhost）：LDAP 服务器的主机名，
- 端口号（默认值：389）：LDAP 服务器的端口号，
- 基本 DN：LDAP 系统信息库的基本 DN，

- 用户 DN（默认值：cn=Directory Manager）：具有足够的权限在基本 DN 下建立新条目的用户的完整 DN，
- 密码：该用户的密码。

将建立 DN 为：

```
ou=ApocRegistry,ou=default,ou=OrganizationConfig,ou=1.0,ou=ApocService,ou=services,<基本 DN>
```

的条目并充填有两个映射文件的内容。

如上所述，`deployApoc` 脚本执行的操作要求 LDAP 服务器的安装目录、版式和模式扩展程序与 Sun Java System Directory Server 非常类似。其他目录将需要手动扩展模式后，才可以执行 `createServiceTree` 脚本。如果需要有关如何使用 OpenLDAP 和 ActiveDirectory 的信息，请参阅附录 C。

所建立的树与保留实体关联的配置数据的树相匹配，其结构与 Sun Java System Identity Server 中用于服务管理的树一样。

其他事项

Configuration Manager 框架要求可以建立具有读取和搜寻权限的 LDAP 服务器连接，以便标识哪个完整 DN 与来自桌面的给定用户或主机标志相关联。因此，要达到该目的，必须配置系统信息库以允许匿名连接，或者必须建立具有读取和搜寻权限的特定用户。

管理应用程序在映射到实体的条目下建立服务树，以保留这些实体的配置数据。因此，用于管理的用户条目需要具有在其所管理的条目下建立子条目的权限。

对来自桌面客户端的框架用户的鉴别可以通过名为 Anonymous 和 GSSAPI 的两种方法实现。Anonymous 方法要求系统信息库中启用匿名的读取和搜寻访问，因为桌面客户端在尝试从 LDAP 服务器中检索数据时不提供任何凭据。要使用 GSSAPI 方法（使用 Kerberos 进行鉴别），必须按照《Sun Java System Directory Server 管理指南》的“实现安全”一章中的说明配置 LDAP 服务器。

第 3 章

Sun™ Web Console

Sun Web Console 是为生成 Sun Microsystems 通用的、基于万维网的系统管理解决方案而设计的。它作为一种环境，供用户访问系统管理应用程序，这些应用程序均提供了一致的用户界面。

此控制台之所以基于 Web 模型，有许多原因。但主要原因是使系统管理员可以使用万维网浏览器来访问系统管理应用程序。

Sun Web Console 具有以下特点：

- 通用的鉴别和授权
- 通用日志
- 所有系统管理应用程序通过基于 HTTPS 的同一端口共用一个入口点
- 通用的观感

控制台的主要优点是管理员登录一次便可使用控制台中的任何应用程序。

系统要求

Sun Web Console 支持多种客户端和服务器操作系统，还支持多种浏览器。

客户端

- Solaris 8 或更高版本上的 Netscape 4.7x、6.2x 和 7.x
- Windows 98、98 SE、ME、2000 和 XP 上的 Netscape 4.7x、6.2x 和 7.x
- Windows 98、98 SE、ME、2000 和 XP 上的 Internet Explorer 5.x 和 6.x
- Linux 和 Solaris 上的 Mozilla

服务器

- Solaris 8 或更高版本
- Red Hat 8 或更高版本，Red Hat Enterprise Linux 2.1
- SuSE Linux 2.1 或更高版本
- J2SE 版本 1.4.1_03 或更高版本
- 如果在服务器上检测到 J2SE 1.4.1 或更低版本，安装程序将提示您使用 Java Desktop System Manament Tools CD 中的 J2SE 版本对已安装产品进行升级。
- Tomcat：4.0.3 或更高版本

Java Desktop System Management Tools CD 中包含 Tomcat

安装 Sun Web Console

在安装 Sun Web Console 之前，请阅读本指南附录 A 中软件包概述和已知问题部分。

Java Desktop System Management Tools CD 含有用于 Solaris SPARC（版本 8 或更高版本）和 Linux 操作系统的二进制 Sun Web Console 安装文件。

1. 在 Java Desktop System Management Tools CD 上，更改到与要安装该控制台的操作系统相对应的 Sun Web Console 目录。

   对于 Linux 系统，更改到 `/linux/swc`；而对于 Solaris SPARC，更改到 `/solsparc/swc`。

2. 键入 `./setup`

   默认情况下，Sun Web Console 不建立安装日志文件。要建立名为“logfile”的安装日志，键入 `./setup | tee logfile`

   注意：Web 控制台的大多数安装和配置工作在执行 `setup` 后将自动执行。有关 Sun Web Console 的 `setup` 应用程序的详细信息，请参阅附录 A。

3. 如果要本地化 Sun Web Console，还需要为每种语言分别安装其他两个软件包，请使用以下表格来确定语言软件包名称，并执行以下操作之一：

1. 对于 Solaris，键入 `pkgadd -d path/pkgname.pkg pkgname`，其中 *pkgname* 是要加入的语言软件包名称。
2. 对于 Linux，键入 `rpm -i path/pkgname<...>.rpm`，其中 *pkgname* 是要加入的软件包名称。

软件包名称	说明
SUNWcmcon，SUNWcmctg	简体中文 Sun(TM) Web Console 2.0
SUNWdmcon，SUNWdmctg	德文 Sun(TM) Web Console 2.0
SUNWemcon，SUNWemctg	西班牙文 Sun(TM) Web Console 2.0
SUNWfmcon，SUNWfmctg	法文 Sun(TM) Web Console 2.0
SUNWhmcon，SUNWhmctg	繁体中文 Sun(TM) Web Console 2.0
SUNWimcon，SUNWimctg	意大利文 Sun(TM) Web Console 2.0
SUNWjmcon，SUNWjmctg	日文 Sun(TM) Web Console 2.0
SUNWkmcon，SUNWkmctg	韩文 Sun(TM) Web Console 2.0
SUNWsmcon，SUNWsmctg	瑞典文 Sun(TM) Web Console 2.0

执行控制台

要注册新的应用程序时，一般只需停止并重新启动 Sun Web Console 服务器。

在首次启动 Sun Web Console 之前，请确保 Configuration Manager 安装已完成。

- 要启动 Sun Web Console，请键入 `smcwebserver start`。
- 要停止 Sun Web Console，请键入 `smcwebserver stop`。
- 要访问 Sun Web Console，请在浏览器中输入以下 URL：`https://<host-name>.<domainname>:6789`

Sun Web Console 一经安装即支持基于 Unix 的鉴别和基于角色的访问控制 (RBAC)。不过，您还可以配置其他鉴别机制，例如 LDAP 鉴别。

注意：默认的会话超时时间为 15 分钟。可以使用命令 `smreg` 配置超时时间。例如，要将超时时间设定为 5 分钟，键入 `smreg add -p -c session.timeout.value=5`。

有关 Sun Web Console 的命令的详细信息，请参阅手册页 smcwebserver 和 smreg。

卸载 Sun Web Console

要卸载 Sun Web Console，执行 `/usr/lib/webconsole/setup -u`。

注意：如果在 `/usr/lib/webconsole` 目录或任何相关子目录中，请不要执行此命令，否则 `pkgrm` 将失败。

Sun Web Console 端口信息

Configuration Manager 使用 Sun Web Console 的以下端口：

- 8005，用于关闭服务，
- 6789，用于访问 https。

这两个端口可以在 `/etc/opt/webconsole/server.xml` 文件中进行更改。更改完成后，必须使用 `/usr/sbin/smcwebserver restart` 来重新启动 Sun Web Console。

第 4 章

Sun Java™ Desktop System Configuration Manager Release 1

Configuration Manager 提供一个在 Sun Web Console 上执行的管理工具。这个基于万维网的用户界面允许管理员遍历组织的层次结构来定义桌面应用程序的策略。可以为层次结构中的各项（例如组织、角色、用户、域和主机）定义这些策略。Configuration Manager 使用多个配置文档样式来显示不同桌面应用程序（例如 Gnome、Mozilla、StarSuite 和 Evolution）的特定设定。

安装 Configuration Manager

在安装 Configuration Manager 之前，需要可用的 Sun Web Console。

1. 更改到 Java Desktop System Management Tools CD 中相应的 Configuration Manager 目录。

   对于 Linux 系统，更改到 `/linux/apoc`。对于 Solaris SPARC，更改到 `/solsparc/apoc`。

2. 键入 `./setup`

3. 输入 LDAP 服务器的主机名。

   默认名称为 localhost。

4. 输入 LDAP 服务器的端口号（默认端口：389）。

5. 输入 LDAP 系统信息库的基本 DN。

6. 输入用于标识用户条目的对象类的名称。默认的对象类为 inetorgperson。

   有关详细信息，请参阅“LDAP 服务器”一章的“组织映射”一节。

7. 输入对于整个 LDAP 系统信息库来说唯一的属性名称。默认属性为 uid。

   有关详细信息，请参阅“LDAP 服务器”一章的“组织映射”一节。

8. 输入有权在 LDAP 服务器中执行查询的用户的完整 DN。

   使用具有读取和搜寻权限的任何完整 DN。对于匿名访问，将本字段设置为空。

9. 输入指定了 LDAP 访问权限的用户的密码。

   如果设置对 LDAP 服务器进行匿名访问，则忽略此步骤。

在安装过程中，一个附加登录模块被加入到 Sun Web Console 中，允许您通过 LDAP 鉴别用户。

在安装结束后，Sun Web Console 自动重新启动，以使您可以访问 Configuration Manager。

注意：通过使用 `/usr/share/webconsole/apoc/configure` 脚本，可以随时修改以前的 Configuration Manager 设定。例如，可以使用此脚本更改到不同的 LDAP 服务器而无需重新安装 Configuration Manager。

执行 Configuration Manager

1. 要访问 Configuration Manager，请在浏览器中输入以下 URL：

```
https://<hostname>.<domainname>:6789
```

2. 出现提示时，输入现有 LDAP 用户的用户名 (uid) 和密码。

   Sun Web Console 将打开。

3. 在控制台窗口中，单击**“Sun Java™ Desktop System Configuration Manager Release 1”**。

注意：如果要跳过 Sun Web Console 的启动页面直接进入 Configuration Manager，请在浏览器中输入以下 URL：

```
https://<hostname>.<domainname>:6789/apoc
```

卸载 Configuration Manager

要从 Sun Web Console 中卸载 Configuration Manager，更改到 Java Desktop System Management Tools CD 中相应的 Configuration Manager 目录，然后执行 `./setup -u`。

注意：在卸载 Configuration Manager 时，LDAP 登录模块将从 Sun Web Console 中删除。

第 5 章

桌面组件

要访问 Configuration Manager 中的配置数据，桌面客户端需要 Sun Java™ Desktop System Configuration Agent。Configuration Agent 与远程配置数据系统信息库及适配器进行通信，并将数据集成到特定的配置系统中。目前支持的配置系统有 GConf、Mozilla 首选设置和 StarSuite Registry。

所有这些组件均作为 Java Desktop System 的一部分发行并安装。

数据访问 / 用户鉴别

Configuration Agent 基于桌面用户的登录 ID 从 LDAP 服务器中检索信息。组织映射文件的 User/UniqueIdAttribute 设定将登录 ID 映射到 LDAP 服务器的用户实体。Configuration Agent 还检索与主机相关的信息，例如主机的名称或 IP 地址。该信息通过组织映射文件的 Host/UniqueIdAttribute 设定映射到 LDAP 服务器中的主机实体。

有两种方式访问 LDAP 服务器：匿名方式或使用 GSSAPI。对于匿名访问，无需在桌面上执行任何操作。对于 GSSAPI 方式，必须在桌面上获得 Kerberos 凭据。要将 Kerberos 凭据的获得与用户登录集成在一起，必须在 Java Desktop System 主机中安装并配置 pam_krb5 模块。可以在 Java Desktop System CD 的 `/usr/share/doc/packages/pam_krb5/README.SuSE` 目录中找到 pam 模块的配置示例。也可以使用 gdm 来将 Kerberos 凭据的获得与用户登录集成在一起，例如使用以下
/etc/pam.d/gdm 文件：

```
#%PAM-1.0
auth   required    pam_unix2.so  nullok #set_secrpc
auth   optional  pam_krb5.so use_first_pass missing_keytab_ok
ccache=SAFE putenv_direct
account required    pam_unix2.so
password required    pam_unix2.so  #strict=false
session required    pam_unix2.so  # trace or none
session required    pam_devperm.so
session optional    pam_console.so
```

Configuration Agent

Configuration Agent 是 apoc 软件包的一部分。安装相应的 RPM 时，将通过 inetd 安装和注册该 API 所需的文件。可以手工安装或通过安装 Java Desktop System 来安装 RPM。

引导信息

要访问远程配置数据，必须向 Configuration Agent 提供 LDAP 服务器的位置。可以通过 YaST2 配置工具 autoYaST 或通过手工编辑 `/opt/apoc/lib` 目录中的属性文件 `policymgr.properties` 来加入此位置。在 YaST2 中，可以将此数据加入到 Network/Advanced 区域中。

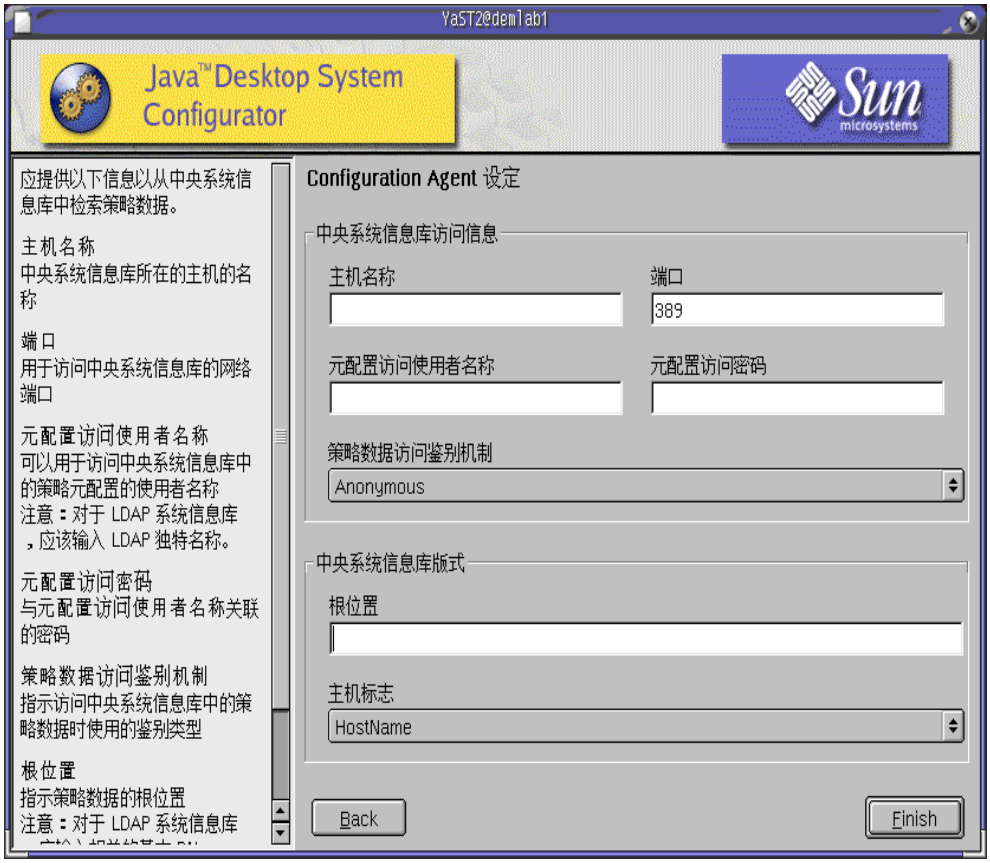

图 1 YaST 中的 Java Desktop System Configuration Agent

以下是执行 Configuration Agent 所需的信息：

- 主机名 (Server)：LDAP 服务器的主机名
- 端口 (Port)：LDAP 服务器的端口号
- 元数据访问用户名 (AuthDn)：具有读取和搜寻系统信息库权限的用户的完整 DN

  注意：如果目录中启用了匿名访问，此设定可设置为空。
- 元数据访问密码 (Password)：注册的 LDAP 用户的密码

  注意：如果目录中启用了匿名访问，此设定可设置为空。
- 策略数据访问鉴别机制 (AuthType)：可为匿名访问或 GSSAPI 访问，这取决于 LDAP 服务器鉴别用户的方式。
- 根位置 (BaseDn)：LDAP 系统信息库的基本 DN
- 主机标志 (HostIdentifier)：可以是 HostName 或 IPAddress，但必须设定成与用来标识主机的 LDAP 属性的内容相匹配。此属性在映射文件中定义为 Host/UniqueIdAttribute。

- 连接超时 (Connect Timeout)：指定在多少秒后，尝试连接 LDAP 服务器将超时。默认值是 1 秒。

注意：任何时候要更改引导和操作设定，都必须重新启动 Configuration Agent。

要在桌面上中重新启动 Configuration Agent，请确保没有执行任何相关的客户端应用程序，作为 root 用户登录，然后输入命令 `/opt/apoc/bin/apocd restart`。

操作设定

可以本地或远程配置 Configuration Agent 的操作设定。要本地配置设定，请编辑 `/opt/apoc/lib` 目录中的 `apocd.properties` 文件。要远程配置设定，请使用 Configuration Manager 中的 Configuration Agent 策略。可以在属性文件中配置以下设定：

- DaemonPort：Configuration Agent 用来收听来自桌面中其客户端的通信的端口
- MaxClientThreads：可同时处理的客户端请求的最大数目
- MaxClientConnections：客户端连接的最大数目
- MaxRequestSize：客户端请求的最大大小
- DaemonChangeDetectionInterval：该配置设定列表的两个更改检测周期之间的时间间隔（以分钟为单位）
- ChangeDetectionInterval：客户端配置数据的两个更改检测周期之间的时间间隔（以分钟为单位）
- GarbageCollectionInterval：本地配置数据库中两个垃圾收集周期之间的时间间隔（以分钟为单位）
- TimeToLive：非脱机配置数据在本地数据库中的保留时间（以分钟为单位）
- LogLevel：代理日志文件中的详细程度

DaemonPort 设定只可本地修改，且需要重新启动代理才能使更改生效。所有其他设定在代理配置的下一个更改检测周期生效。LogLevel 中指定的日志级别的值必须与 Java Logger 级别一致。按严重性递减的顺序，这些级别为：SEVERE、WARNING、INFO、CONFIG、FINE、FINER 和 FINEST。

传播配置数据更改

可以使用“操作设定”一节中介绍的 ChangeDetectionInterval 设定来调整远程配置数据更改到客户端应用程序的传播。您为此设定指定的值表示最长经过多少分钟后，远程更改反映在客户端应用程序中。Configuration Agent 的值越小，Configuration Agent 和 LDAP 服务器的活动量越大。因此，调整此设定的值时应谨慎。例如，在初始部署阶段，可以将该值设为一分钟，这样就可以轻松地测试远程配置对客户端应用程序的影响。完成测试后，请将此设定还原为初始值。

Configuation Agent 端口信息

Configuration Agent 使用两个端口：

1. 守护程序端口（默认值为 38900），由守护程序用来与客户端应用程序通信。
2. 守护程序管理端口（默认值为 38901），由守护程序控制程序 `apocdctl` 用来与守护程序通信。

更改守护程序端口：

要更改守护程序端口，必须修改该守护程序的 `apocd.properties` 文件中的 `DaemonPort` 属性以及 `/etc/services` 和 `/etc/inetd.conf` 中的 `apocd`。然后，重新启动该守护程序并重新装入 `inetd`。

更改守护程序管理端口：

要更改守护程序管理端口，必须修改该守护程序的 `apocd.properties` 文件中的 `DaemonAdminPort` 属性，然后重新启动该守护程序。

GConf 适配器

GConf 适配器是 `apoc-adapter-gconf` 软件包的一部分。从相应的 `RPM` 安装适配器时，`/etc/gconf/2/path` 中的 GConf 数据源路径被更新，以包括 Configuration Manager 源路径。旧路径的备份存储在 `/etc/gconf/2/path.apocBackup` 中。如果旧路径涉及自定义数据源，需要更新该路径，方法是将默认路径的更改并入到新安装的 Manager 路径。适配器提供两种数据源，它们为：

- “apoc:readonly:”：用来访问策略的非保护设定。在用户设定之后、本地默认设定之前插入此数据源。
- “apoc:readonly:mandatory@”：用来访问策略的保护设定。在本地强制设定之后、用户设定之前插入此数据源。

Mozilla 适配器

Mozilla 适配器是 `mozilla-apoc-integration` 软件包的一部分。在从相应的 RPM 安装该适配器时，所需文件被加入到现有的 Mozilla 中并自动注册。

StarSuite 适配器

StarSuite 适配器包含在标准 StarSuite 安装软件中，并且允许在不进行任何特殊修改的情况下访问策略配置数据。

第 6 章

附录 A — Sun Web Console

已知问题

安全性

如果用户不具备相应的知识，某些用户操作会使会话保持活动状态。例如，当用户关闭浏览器窗口时，用户不会自动从 Sun Web Console 中注销。而在关闭应用程序窗口前，用户必须在 Sun Web Console 中明确注销会话。

安装脚本用法

语法说明：`setup [-h] | [-n] | [-d <ver>,<arch>[,client1,client2,...]][-u [-f]]`

-h = 打印使用说明

-n = 安装结束后不启动服务器

-u = 卸载 Sun Web Console

-f = 如果在执行安装程序时安装了 Tomcat 和 Java 1.4，卸载这些软件包。该参数只能和参数 -u 一起使用。

有关可用的安装参数的完整说明，请执行 `setup -h`。

要卸载 Sun Web Console，请执行 `/usr/lib/webconsole/setup -u`

注意：如果在 `/usr/lib/webconsole` 目录或任何相关子目录中，请不要执行此命令，否则 `pkgrm` 将失败。

Sun Web Console 软件包

Solaris 软件包

软件包名称	*说明*
SUNWmctag	Sun Web Console UI 标记库
SUNWmcon	Sun Web Console
SUNWmcos	Sun Web Console 的通用 Solaris 服务
SUNWmcosx	Sun Web Console 的 Solaris 版本特有的服务
SUNWmconr	Sun Web Console 根
SUNWjato	Sun One Application Framework 运行时环境
SUNWtcatu	Tomcat

Linux RPM

软件包名称	*说明*
SUNWmctag	Sun Web Console UI 标记库
SUNWmcon	Sun Web Console
SUNWmcos	Sun Web Console 的通用 Linux 服务
SUNWmcosx	Sun Web Console 的 Linux 版本特有的服务
SUNWmconr	Sun Web Console 根
SUNWjato	Sun One Application Framework 运行时环境
tomcat4	Tomcat

附录 B — Configuration Manager

Configuration Manager 软件包

Solaris 软件包

软件包名称	*说明*
SUNWapm	Configuration Manager
SUNWapmca	Configuration Agent 文档样式
SUNWapmev	Evolution 文档样式
SUNWapmgo	Gnome 文档样式
SUNWapmmo	Mozilla 文档样式
SUNWapmso	StarSuite 文档样式

Linux RPM

软件包名称	*说明*
apoc-manager	Configuration Manager
apoc-agent-templates	Configuration Agent 文档样式
apoc-evolution-templates	Evolution 文档样式
apoc-gnome-templates	Gnome 文档样式
apoc-mozilla-templates	Mozilla 文档样式
apoc-starsuite-templates	StarSuite 文档样式

第 8 章

附录 C

将 OpenLDAP 服务器用于 Configuration Manager

如果要将 OpenLDAP 服务器用作 Configuration Manager 数据的系统信息库，则必须扩展该服务器的类型，以支持用于存储配置数据的对象类和属性。可在 Java Desktop System Management Tools CD 中提供的 Configuration Manager 配置工具的 openldap 子目录中找到名为 apoc.schema 的定制类型文件。

须在 OpenLDAP 配置目录 (`/etc/openldap`) 的 schema 子目录中复制此文件，并通过将它包括在位于该目录中的 `slapd.conf` 文件中，将此文件添加到 OpenLDAP 类型。这是通过在类型序列的结尾插入 `include /etc/openldap/schema/apoc.schema` 行而将它包含在该文件中。有关扩展 OpenLDAP 服务器的类型的详细信息，请参阅该服务器的手册。

如果要使 OpenLDAP 数据库可用来存储配置数据，则必须使用随 Configuration Manager 提供的配置工具。因为在安装过程的前一步骤中已经扩展了此类型，所以只需运行 `createServiceTree` 脚本即可。必须从配置工具目录启动该脚本，启动时可以采用任何用户身份输入以下命令：`./createServiceTree`。此脚本提示用户输入在本文档中的配置工具部分所指定的 OpenLDAP 数据库的有关信息。配置工具中的 `openldap` 子目录提供了默认的映射文件，该映射文件使用 OpenLDAP 中支持的典型对象类和属性。此文件名为 `OrganisationalMapping`，并可在启动 `createServiceTree` 文件前配置它，方法是：复制此文件以覆盖主配置工具目录中的同名文件。

请注意，Configuration Manager 代理将通过提供要为其请求资料的用户的 DN（但无需密码）来尝试匿名连接 OpenLDAP 服务器。在某些版本的 OpenLDAP 服务器中，默认情况下可能禁用此匿名鉴别模式；在这种情况下，必须通过使用以下方法启用它：在 OpenLDAP 配置目录 (`/etc/openldap`) 中的文件 `slapd.conf` 中定义的通用服务器参数中添加 allow `bind_anon_cred` 行。有关该参数的详细信息，请参阅该服务器的手册。

将 Active Directory 服务器用于 Configuration Manager

如果要将 Active Directory 服务器用作 Configuration Manager 数据的系统信息库，则必须扩展服务器的类型，以支持存储配置数据的对象类和属性。可在 Management Tools CD 中提供的 Configuration Manager 配置工具的 `ad` 子目录中找到名为 `apoc-ad.ldf` 的类型扩展文件。有关详细信息，请参阅配置工具部分。

必须执行以下步骤将 `apoc-ad.ldf` 文件导入到 Active Directory 类型中：

1. 启用类型扩展。有关如何执行该操作的详细信息，请参阅 Active Directory 文档。
2. 命令提示符下执行以下命令：`ldifde -i -c "DC=Sun,DC=COM" <Base DN> -f apoc-ad-registry.ldf`。

   说明：使用 Active Directory 基本 DN 替换 <*Base DN*>。

如果要使 Active Directory 服务器可用来存储配置数据，则必须使用配置工具。因为在安装过程的前面步骤中已经扩展了此类型，所以仅需运行 `createServiceTree` 脚本。必须从配置工具目录启动该脚本，启动时可以以任何用户身份输入以下命令：`./createServiceTree`。该脚本提示用户输入有关 Active Directory 数据库的信息。配置工具目录的 `ad` 子目录提供了默认的映射文件，该映射文件使用 Active Directory 中支持的典型对象类和属性。此文件名为 `OrganisationalMapping`，并可在启动 `createServiceTree` 文件前配置它，方法是：复制此文件以覆盖主配置工具目录中的同名文件。

此后，Active Directory 服务器可用于 Configuration Manager。在安装 Configuration Manager 时，请提供对该目录树具有读取权限的用户的完整 DN 和密码。此用户不能将 Active Directory 用作任何其他目的。有关如何设置此类用户的详细信息，请参阅 Active Directory 文档。另外，Active Directory 的域名对于运行 Configuration Manager 的计算机来说必须是已知的。这可以通过以下方法实现：将映射 Active Directory 服务器的 IP 地址及其域名的行添加到该计算机的 `/etc/hosts` 文件中。

如果要从 Java Desktop System (JDS) 主机检索配置数据，则 Active Directory 的域名对于该主机来说也必须是已知的。可以通过下面两种方法鉴别 JDS 用户：匿名鉴别和使用 GSSAPI 鉴别。

- 如果要使用匿名连接进行鉴别，则必须配置 Active Directory 服务器以将读取权限授予每个用户。有关如何执行该操作的详细信息，请参阅 Active Directory 文档。
- 如果要使用 GSSAPI 进行鉴别，则必须修改指定 Kerberos 参数的文件 `/etc/krb5.conf`，以定义 Active Directory 域并指向 Active Directory 服务器来作为其主要分发中心 (KDC)。还必须指定 Active Directory 支持的、作为默认加密类型的 DES 类型，即 `des-cbc-crc` 和 `des-cbc-md5` 类型。有关如何执行该操作的详细信息，请参阅 Kerberos 文档。在访问配置数据前，必须首先获取登录 JDS 的用户的有效凭据。这可以通过手动运行 `kinit` 命令并提供 Active Directory 中定义的用户密码来实现。登录时，其他类型可以自动生成这些凭据。有关详细信息，请参阅 JDS 文档。