



# Java Desktop System Configuration Manager 发行版 1.1 安装指南

---

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

文件号码: 819-0962-10  
2005 年 2 月

版权所有 2004 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. 保留所有权利。

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

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

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

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

此文档按“原样”提供，对于所有明示或默示的条件、担保和陈述，包括任何对适销性、适用性和非侵权性的默示保证，Sun 均不承担任何责任，除非此免责声明的适用范围在法律上无效。



050105@10536



# 目录

---

- 1 简介 9
  - 简介 9
  
- 2 LDAP 服务器 11
  - 概念 11
  - 设置 12
    - 部署工具 12
    - 模式扩展 12
    - 组织映射 12
    - 用户配置文件映射 14
    - 部署 14
  - 其他事项 15
  
- 3 Sun Web Console 17
  - 系统要求 17
    - 客户端 17
    - 服务器 18
  - 安装 Sun Web Console 18
    - ▼ 安装 Sun Web Console 的步骤: 18
  - 运行控制台 19
  - 卸载 Sun Web Console 20
  - Sun Web Console 端口信息 20
  
- 4 Java Desktop System Configuration Manager 发行版 1.1 21
  - 安装 Configuration Manager 21

▼ 安装 Configuration Manager 的步骤:	21
运行 Configuration Manager	22
▼ 启动 Configuration Manager 的步骤:	22
卸载 Configuration Manager	23
<b>5 在 Linux 和 Solaris 上安装桌面组件</b>	<b>25</b>
Configuration Agent	25
引导信息	26
端口设置	27
更改检测间隔	28
操作设置	29
应用代理设置	31
数据访问/用户验证	31
GConf 适配器	32
Java Preferences 适配器	32
Mozilla 适配器	33
StarSuite 适配器	33
<b>6 在 Microsoft Windows 上安装桌面组件</b>	<b>35</b>
Configuration Agent 安装	35
欢迎使用	36
许可协议	36
安装类型	37
准备安装	44
安装汇总	45
更改 Configuration Agent 设置	46
卸载 Configuration Agent	46
<b>A Sun Web Console 软件包</b>	<b>47</b>
已知问题	47
安全性	47
安装脚本用法	47
Sun Web Console 软件包	48
Solaris 软件包	48
Linux RPM	48

<b>B</b>	<b>Configuration Manager 软件包</b>	<b>51</b>
	Configuration Manager 软件包	51
	Solaris 软件包	51
	Linux RPM	51
<b>C</b>	<b>在 Configuration Manager 中使用 OpenLDAP 和 Active Directory</b>	<b>53</b>
	在 Configuration Manager 中使用 OpenLDAP 服务器	53
	在 Configuration Manager 中使用 Active Directory 服务器	54



## 图形

---

图 5-1	Configuration Agent, 配置系统信息库	26
图 5-2	Configuration Agent, 验证机制	27
图 5-3	Configuration Agent, 端口设置	28
图 5-4	Configuration Agent, 数据目录	29
图 5-5	Configuration Agent, 请求处理和日志记录	29
图 5-6	Configuration Agent, “汇总”页面	31
图 6-1	Configuration Agent, “欢迎使用”页	36
图 6-2	Configuration Agent, 软件许可协议	36
图 6-3	Configuration Agent, “安装类型”页	37
图 6-4	Configuration Agent, 配置系统信息库	38
图 6-5	Configuration Agent, 验证机制	39
图 6-6	Configuration Agent, 端口设置	41
图 6-7	Configuration Agent, 数据目录	41
图 6-8	Configuration Agent, 请求处理和日志记录	42
图 6-9	Configuration Agent, “汇总”页面	43
图 6-10	Configuration Agent, “准备安装”页	44
图 6-11	Configuration Agent, “安装汇总”页	45



# 第 1 章

---

## 简介

---

提供 Java™ Desktop System Configuration Manager 发行版 1.1 的简要介绍。

---

## 简介

Java™ Desktop System Configuration Manager 发行版 1.1 用于为运行 Java™ Desktop System 的桌面主机提供核心配置。可以为某个组织结构的各种元素指定设置，使管理员可以以一种简单的方式管理多组用户或主机。它的主要组件有：

- 一个 LDAP 服务器，包含要管理的用户和主机的组织结构，还包含配置数据
- 一个基于 Web 的管理工具，允许管理员为该组织结构中的元素定义和指定配置数据
- 客户端主机上安装的桌面组件，这些组件可以代表当前登录的用户检索配置数据，并将数据提供给构成 Java Desktop System 的各种应用程序

管理工具是一个基于 Web 的应用程序，在 Sun Web Console 中运行。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。

桌面组件围绕 Java™ Desktop System Configuration Agent 进行组织，该代理可以代表用户从 LDAP 服务器检索配置数据，并将数据提供给许多配置系统适配器，这些适配器可以使用策略设置来补充本地配置（由应用程序和用户设置提供的缺省设置）。目前支持的配置系统为 GConf（用于处理 Gnome 应用程序的配置，例如 Gnome 桌面或 Evolution）、Mozilla™ Preferences 和 StarRegistry（StarSuite 的配置系统）。



## 第 2 章

---

# LDAP 服务器

---

本章提供了有关设置和部署 LDAP 服务器，以便将其与 Configuration Manager 配合使用的信息。

---

## 概念

在 Java Desktop System Configuration Manager 框架中，配置数据与实体（即 LDAP 系统信息库中的条目，这些条目又与公司的组织结构元素相对应）相关联。

已识别的实体包括：

- 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系统信息库中各条目的物理位置决定。也就是说，组织和域实体可以包括树中位于这两个实体下的任何条目。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

要在 Configuration Manager 中使用现有的 LDAP 服务器，您需要：

- 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类和属性。
- 在服务器中自定义和存储系统信息库中各条目的映射信息，以及 Configuration Manager 支持的各实体的映射信息。

## 部署工具

要在 Configuration Manager 中使用现有的 LDAP 服务器，必须具有安装 CD 中的以下部署工具：

- 88apoc-registry.ldif：模式文件，提供用来存储配置数据的对象类和属性。
- OrganizationMapping：缺省的特性文件，用来描述 LDAP 条目和 Configuration Manager 实体之间的映射。
- UserProfileMapping：缺省的特性文件，用来描述 LDAP 用户条目属性和 Configuration Manager 用户配置文件属性之间的映射。
- createServiceTree：脚本，用来将映射文件存储到 LDAP 系统信息库中。
- deployApoc：脚本，用来扩展 LDAP 服务器模式并将映射文件存储到 LDAP 系统信息库中。

## 模式扩展

配置数据存储条目树中，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类和属性之前，必须向 LDAP 服务器模式中添加对象和类。例如，提供的模式扩展文件以 LDIF 格式向 Sun Java™ System Directory Server 中添加这些对象和类。要向其他 LDAP 服务器中添加这些对象和类，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 Organization 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。键名及其缺省值有：

```

# 所有用户条目使用的对象类
User/ObjectClass=inetorgperson
# 属性，用户条目中该属性的值在系统信息库中是唯一的
User/UniqueIdAttribute=uid
# 用户条目的组织条目中的可选容器，
# 如果未使用，则删除该行
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> 格式，并且必须对齐。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* 键必须指定可以从用户条目或主机条目中查询属性值的属性。该键还必须含有条目所属角色的完整 DN。*MemberAttribute* 键必须从用户或主机条目为搜索过滤器指定属性。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，但 *MemberAttribute* 键必须是可以在过滤器中使用的真实属性。键名及其缺省值有：

```

# 角色的对象类列表
Role/ObjectClass=nsRoleDefinition
# 相应命名属性的对齐列表
Role/NamingAttribute=cn
# 包含用户/主机所属角色的 DN
# 的真实属性（可在过滤器中使用）
Role/MemberAttribute=nsRoleDN
# 对用户/主机的查询返回用户/主机
# 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，采用的是对象类和相应命名属性的对齐列表。键名及其缺省值有：

```

# 组织的对象类列表
Organization/ObjectClass=organization
# 相应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。键名及其缺省值有：

```

# 域的对象类列表
Domain/ObjectClass=ipNetwork
# 相应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。键名及其缺省值有：

```

# 所有主机条目使用的对象类
Host/ObjectClass=ipHost
# 属性，主机条目中该属性的值在系统信息库中是唯一的
Host/UniqueIdAttribute=cn
# 主机条目的域条目中的可选容器，
# 如果未使用，则删除该行

```

Host/Container=ou=Hosts

## 用户配置文件映射

要定义 LDAP 用户条目属性和 Configuration Manager 用户实体属性之间的映射，必须编辑用户配置文件映射文件。每个键对应一个 Configuration Manager 用户属性。正如组织映射所示，可以将键作为值指定给用户条目中的属性名。

User/DisplayNameFormat 设置中使用的属性必须在 *User Profile* 映射中指定。键名及其缺省值有：

```
# 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 (city)
org.openoffice.UserProfile/Data/l = l
# organizationalPerson.st (state)
org.openoffice.UserProfile/Data/st = st
# organizationalPerson.postalCode
org.openoffice.UserProfile/Data/postalcode = postalcode
# country.c (country)
org.openoffice.UserProfile/Data/c =
# organizationalPerson.o (company)
org.openoffice.UserProfile/Data/o = o,organizationName
# deprecated -- no LDAP corollary
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

自定义反映 LDAP 系统信息库状态的映射文件之后，即可部署这些映射文件。如果 LDAP 服务器模式已包含所需的对象类和属性，则可以直接运行脚本 `createServiceTree`；否则，必须运行脚本 `deployApoc`。

`deployApoc` 脚本专门在 Sun Java™ System Directory Server 中使用。该脚本将提供的模式扩展文件复制到适当的目录中，循环 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> 参数必须是指向 Directory Server 安装的 slapd-*<server name>* 子目录的路径。假定使用缺省目录和服务器的安装名为 *myserver.mydomain*，则该目录应为 */var/Sun/mps/slapd-myserver.mydomain*。

无论 createServiceTree 脚本是直接调用的还是通过 deployApoc 脚本调用的，都将提示用户提供 LDAP 服务器的位置（主机名、端口号和基本 DN）以及具有管理权限的用户的定义（完整 DN 和密码）。然后，该脚本将在 LDAP 服务器中创建引导服务并在其中存储映射文件。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**（缺省值：**localhost**）：LDAP 服务器的主机名，
- **端口号**（缺省值：**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**（缺省值：**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，

创建具有以下 DN 的条目：

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

并用两个映射文件的内容填充该条目。

如前文所述，deployApoc 脚本执行的操作假定 LDAP 服务器的安装目录、布局和模式扩展步骤与 Sun Java System Directory Server 的非常相似。对于其他目录，则需要手动扩展模式，然后才能运行 createServiceTree 脚本。有关使用 OpenLDAP 和 ActiveDirectory 的详细信息，请参见附录 C”。

已创建的树（与用来保存与实体关联的配置数据的树匹配）与 Sun Java System Identity Server 中的服务管理所使用的树的结构保持一致。

---

## 其他事项

Configuration Manager 框架要求可以创建具有读取和搜索权限的 LDAP 服务器连接，以标识与来自桌面的给定用户或主机标识符所关联的完整 DN。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索权限的特定用户。

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。

可以使用以下两个方法从桌面客户端来验证框架的用户：Anonymous 方法和 GSSAPI 方法。Anonymous 方法要求在系统信息库中启用匿名的读取和搜索访问，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证。要使用 GSSAPI 方法（使用 Kerberos 进行验证），LDAP 服务器必须按在《*Sun Java™ System Directory Server 5 2004Q2 Administration Guide*》中的 "Managing Authentication and Encryption" 一章中的介绍进行配置。

## 第 3 章

---

# Sun™ Web Console

---

Sun Web Console 是为生成 Sun Microsystems 通用的、基于 Web 的系统管理解决方案而设计的。该控制台提供了一种环境，供用户访问具有一致用户界面的系统管理应用程序。

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 Web 浏览器来访问系统管理应用程序。

Sun Web Console 具有以下特点：

- 通用的验证和授权
- 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
- 通用的外观

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 支持多个客户端和服务器操作系统，还支持若干个浏览器。

### 客户端

- Solaris™ 8 或更高版本上的 Netscape™ 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™ 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 Management 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  
Tomcat 包含在 Java Desktop System Management Tools CD 中

---

## 安装 Sun Web Console

安装 Sun Web Console 之前, 请先阅读本指南中附录 A 中的软件包汇总和已知问题部分。

用于 Solaris SPARC (版本 8 或更高版本) 和 Linux 操作系统的 Sun Web Console 安装二进制文件可以从 Java Desktop System Management Tools CD 中获得。

### ▼ 安装 Sun Web Console 的步骤:

- 步骤
1. 在 Java Desktop System Management Tools CD 上, 转到与要安装控制台的操作系统相对应的 Sun Web Console 目录 ("swc")。
  2. 键入 `./setup`。  
缺省情况下, Sun Web Console 不会创建安装日志文件。要创建名为“logfile”的安装日志, 请键入 `./setup | tee logfile`。

---

注意 – 运行安装程序时, 会自动执行 Web 控制台的大多数安装和配置。有关 Sun Web Console 的设置应用程序的详细信息, 请参见附录 A。

---

3. 如果要本地化 Sun Web Console, 需要为每种语言安装两个附加软件包。请使用下表决定每种语言需要的软件包的名称, 并执行以下操作之一:
  - 在 Solaris 上, 键入 `pkgadd -d path/pkgname.pkg pkgname`, 其中 `pkgname` 是要添加的语言包的名称。
  - 在 Linux 上, 键入 `rpm -i path/pkgname<...>.rpm`, 其中 `pkgname` 是要添加的软件包的名称。

软件包名称	描述
SUNWcmcon 和 SUNWcmctg	简体中文 Sun™ Web Console 2.0
SUNWdmcon 和 SUNWdmctg	德语 Sun™ Web Console 2.0
SUNWemcon 和 SUNWemctg	西班牙语 Sun™ Web Console 2.0
SUNWfmcon 和 SUNWfmctg	法语 Sun™ Web Console 2.0
SUNWhmcon 和 SUNWhmctg	繁体中文 Sun™ Web Console 2.0
SUNWimcon 和 SUNWimctg	意大利语 Sun™ Web Console 2.0
SUNWjmcon 和 SUNWjmctg	日语 Sun™ Web Console 2.0
SUNWkmcon 和 SUNWkmctg	朝鲜语 Sun™ Web Console 2.0
SUNWsmcon 和 SUNWsmctg	瑞典语 Sun™ Web Console 2.0

## 运行控制台

如果需要注册新的应用程序，通常只需停止并重新启动 Sun Web Console 服务器。



**注意** – 第一次启动 Sun Web Console 之前，请确保已完成 Configuration Manager 的安装。Sun Web Console 不会成功运行，直到您在控制台上至少部署了一个应用程序。

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

Sun Web Console 一经安装就支持基于 Unix 的验证和基于角色的访问控制 (RBAC)。但是，您也可以配置其他验证机制，例如 LDAP 验证。

**注意** – 缺省的会话超时时间为 15 分钟。可以使用 `smreg` 命令配置超时长度。例如，要将超时长度设置为 5 分钟，请键入 **smreg add -p -c session.timeout.value=5**。

有关 Sun Web Console 命令的详细信息，请参见 `smcwebserver` 和 `smreg` 手册页。

---

## 卸载 Sun Web Console

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

---

**注意** - 当您位于 `/usr/lib/webconsole` 目录或任何相关的子目录中时，请勿运行此命令，否则 `pkgrm` 将失败。

---

---

## Sun Web Console 端口信息

Configuration Manager 使用以下 Sun Web Console 端口：

- 8005 用于关闭服务。
- 6789 用于 https 访问。

可以在 `/etc/opt/webconsole/server.xml` 中更改这两个端口。更改端口后，请使用 `/usr/sbin/smcwebserver restart` 命令重新启动 Sun Web Console。

## 第 4 章

---

# Java™ Desktop System Configuration Manager 发行版 1.1

---

Configuration Manager 提供了一个管理工具，该工具在 Sun Web Console 中运行。这个基于 Web 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 使用几个配置模板来显示不同的桌面应用程序（例如 Gnome、Mozilla、StarSuite 和 Evolution）特有的设置。

---

## 安装 Configuration Manager

需要先安装 Sun Web Console 才能安装 Configuration Manager。

### ▼ 安装 Configuration Manager 的步骤：

- 步骤
1. 转到 Java Desktop System Management Tools CD 上对应的 Configuration Manager 目录 ("apoc")。
  2. 键入 `./setup`。
  3. 输入 LDAP 服务器的主机名。  
缺省名称为 `localhost`。
  4. 输入 LDAP 服务器的端口号（缺省值：389）。
  5. 输入 LDAP 系统信息库的基本 DN。
  6. 输入用于标识用户实体的对象类的名称。缺省对象类为 `inetorgperson`。  
有关详细信息，请参见“LDAP 服务器”一章中的第 12 页“组织映射”。

7. 输入对于整个 LDAP 系统信息库来说唯一的属性名称。缺省属性为 `uid`。  
有关详细信息，请参见“LDAP 服务器”一章中的第 12 页“组织映射”。
8. 输入具有在 LDAP 服务器上执行查询所需访问权限的用户的完整 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。  
安装过程中会在 Sun Web Console 中添加一个附加的登录模块，允许您通过 LDAP 验证用户。  
安装完成后，Sun Web Console 将自动重新启动，以便于您访问 Configuration Manager。

---

注意 – 您可以使用 `/usr/share/webconsole/apoc/configure` 脚本随时修改前面的 Configuration Manager 设置。例如，您可以使用该脚本转到另一个 LDAP 服务器，而不必重新安装 Configuration Manager。

---

---

## 运行 Configuration Manager

### ▼ 启动 Configuration Manager 的步骤：

- 步骤
1. 在浏览器中键入以下 URL：`https://<hostname>.<domainname>:6789`
  2. 在提示符下，键入现有 LDAP 用户的用户名 (`uid`) 和密码。  
将打开 Sun Web Console。
  3. 在控制台窗口中，单击 Sun Java Desktop System Configuration Manager 发行版 1.1。

#### 更多信息 直接访问 Configuration Manager

如果要跳过 Sun Web Console 的启动页而直接进入 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，请转到 Java Desktop System Management Tools CD 上的对应 Configuration Manager 目录，然后运行

```
./setup -u。
```

---

**注意** – 卸载 Configuration Manager 时，将从 Sun Web Console 中删除 LDAP 登录模块。

---



## 第 5 章

# 在 Linux 和 Solaris™ 上安装桌面组件

本章介绍专用于 Solaris 操作系统和 Linux 操作系统的信息。

要从 Configuration Manager 中访问配置数据，需要在桌面客户端安装 Java™ Desktop System Configuration Agent。Configuration Agent 可以与远程配置数据系统信息库和适配器进行通信，还能将数据集成到特定的配置系统中。当前支持的配置系统为：GConf、Java Preferences、Mozilla Preferences 和 StarSuite Registry。

## Configuration Agent

多个不同的软件包都包含了 Configuration Agent，这些软件包如下表所示：

Solaris 软件包名称	Linux RPM 名称	描述
SUNWapbas	apoc-base	配置共享库
SUNWapmsc	apoc-misc	Configuration Agent 杂项文件
SUNWapoc	apoc	Configuration Agent
SUNWapdc	apoc-config	Configuration Agent 向导

安装这些软件包时，将安装此 API 需要的所有文件。可以手动安装这些软件包，也可以通过 Java Desktop System 安装来完成。安装完成之后，必须在您的系统上配置并启用 Configuration Agent。

要访问远程配置数据，Configuration Agent 需要某些最基本的引导信息，例如 LDAP 服务器的主机名和端口。这些信息在一组特性文件（例如 `polycmgr.properties`、`apocd.properties` 和 `os.properties`）中进行维护。这些文件存储在本地 `/etc/apoc` 目录中。您可以手动编辑这些特性文件，也可以使用 Configuration Agent 的配置向导来完成。

该配置向导提供了一个图形用户界面，可以引导您完成 Configuration Agent 的必要设置。对于该向导的每个页面，都存在一个相应的帮助屏幕。您可以通过 `/usr/bin/apoc-config` 脚本以超级用户 (root) 的身份启动该向导。也可以使用“首选项”>“系统工具”>“网络设置”，或者 Nautilus 文件管理器中的“系统设置:>“网络设置”下的相应桌面菜单条目。

---

**注意** – 也可以启动该向导而不启动图形界面。例如，通过执行 `/usr/bin/apoc-config -nodisplay` 在控制台模式下启动该向导。

---

## 引导信息



图 5-1 Configuration Agent: 配置系统信息库

---

**注意** – 在括号中指明了关联的特性文件关键字（如果适用）。

---

- **服务器标识 (Server)**: LDAP 服务器的主机名。
- **服务器端口 (Port)**: LDAP 服务器的端口号。
- **后缀 (BaseDn)**: LDAP 系统信息库的基本 DN。
- **状态**: Configuration Agent 的状态。可以使用该复选框激活或取消激活 Configuration Agent。要使用配置系统信息库，必须激活 Configuration Agent。激活过程自动包括通过 `inetd` 进行必要注册的操作。

---

注意 – 要手动启用或禁用 Configuration Agent，请以**超级用户**身份登录，并分别键入 `/usr/lib/apocd enable` 或 `/usr/lib/apocd disable` 命令。

---

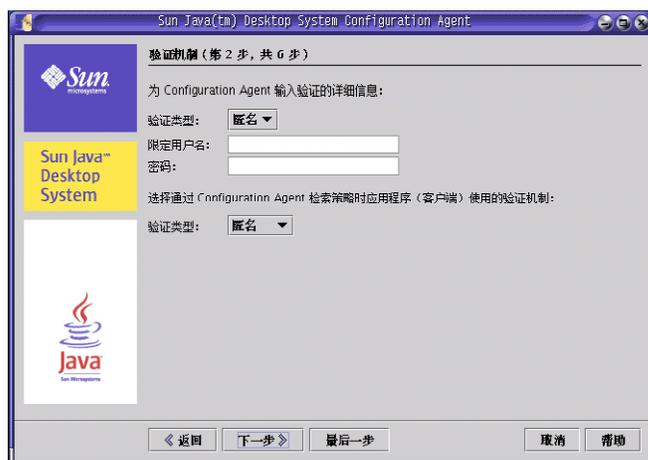


图 5-2 Configuration Agent，验证机制

- **主机标识符 (HostIdentifier)**：可以是“HostName”或“IPAddress”。必须将该标识符设置为与用来标识主机的 LDAP 属性的内容相匹配。此属性在映射文件中定义为 Host/UniqueIdAttribute。
- **Configuration Agent 的验证类型**：可以是“匿名”或“简单”。如果选定了“匿名”，那么**限定用户名**和**密码**字段将自动禁用。
- **限定用户名 (AuthDn)**：具有读取和搜索系统信息库权限的用户的完整 DN。
- **密码 (Password)**：已注册 LDAP 用户的密码

---

注意 – 如果在目录中启用了匿名访问，则**限定用户名**和**密码**设置可为空白。

---

- 应用程序的**验证类型 (AuthType)**：可以是“匿名”或“GSSAPI”，这取决于 LDAP 服务器验证用户的方式。

---

注意 – 有关详细信息，请参见第 31 页“数据访问/用户验证”。

---

## 端口设置

Configuration Agent 使用以下两个端口：

- **代理端口 (DaemonPort)**: 代理使用该端口与客户端应用程序进行通信 (缺省值为 **38900**)。
- **管理端口 (DaemonAdminPort)**: 在与代理通信时, 代理控制器程序 (apocd) 所用的端口 (缺省值为 **38901**)。

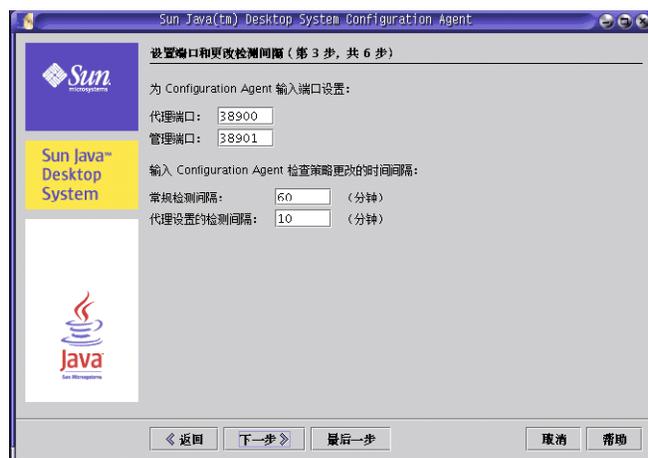


图 5-3 Configuration Agent, 端口设置

## 更改检测间隔

Configuration Agent 使用以下两种间隔定期检查配置数据中的更改:

- **常规检测间隔 (ChangeDetectionInterval)**: 桌面应用程序 (客户端) 的配置数据的两个更改检测周期之间的间隔 (以分钟为单位)。

---

注意 - 指定 **-1** 将关闭更改检测。

---

- **代理设置的间隔 (DaemonChangeDetectionInterval)**: 代理特定的配置设置的两个更改检测周期之间的间隔 (以分钟为单位)。

---

注意 - 指定 **-1** 将关闭更改检测。

---

可以使用常规检测间隔来调整远程配置数据更改至客户端应用程序的传播。为此设置指定的值表示最长经过多少分钟后, 远程更改才反映在客户端应用程序中。

值越小，Configuration Agent 和 LDAP 服务器活动就越多。因此，调整此设置的值时应谨慎。例如，在初始部署阶段，可以将该值设置为一分钟，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，请将此设置还原为初始值。

## 操作设置



图 5-4 Configuration Agent，数据目录

可以配置以下设置：

- **数据目录 (DataDir)**：用于存储运行时数据的目录。缺省值为 `/var/opt/apoc`。
- **缓存数据的保存期限 (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）。



图 5-5 Configuration Agent, 请求处理和日志记录

- **垃圾收集周期 (GarbageCollectionInterval)**: 本地配置数据库中两个垃圾收集周期之间的间隔 (以分钟为单位)。
- **客户端线程的最大数目 (MaxClientThreads)**: 可以同时处理的客户端请求的最大数目。
- **客户端连接的最大数目 (MaxClientConnections)**: 客户端连接的最大数目。
- **最大请求大小 (MaxRequestSize)**: 客户端请求的最大大小。
- **连接超时 (ConnectTimeout)**: 表示 LDAP 服务器响应连接请求时所允许的间隔。缺省值为一秒。
- **日志级别 (LogLevel)**: 代理日志文件中的详细程度。此日志级别应与 Java 记录程序的级别保持一致。按严重性递减的顺序, 这些级别为:
  - 严重
  - 警告
  - 信息
  - 配置
  - 良好
  - 较好
  - 最好

---

**注意** - 大多数操作设置 (数据目录和连接超时除外) 也可以通过 LDAP 服务器中存储的相应策略进行集中维护。如果要使用此功能, 请不要通过向导更改相应的设置。相反, 应使用 Configuration Manager 中的 Configuration Agent 策略来集中指定操作设置。

---

## 应用代理设置

已通过 Configuration Manager 存储在 LDAP 服务器中的操作设置（“数据目录”和“连接超时”除外）将在代理配置的下一个更改检测周期自动生效（请参见 DaemonChangeDetectionInterval）。



图 5-6 Configuration Agent: “汇总”页面

在本地更改的所有其他设置均需要重新装入 Configuration Agent，或者重新启动 Configuration Agent。如果使用配置向导，将自动执行重新装入和重新启动操作。

---

**注意** – 要手动重新启动 Configuration Agent，请确保任何相关的客户端应用程序均未运行，然后以超级用户身份登录并键入命令 `/usr/lib/apoc/apocd restart`。

---

## 数据访问/用户验证

Configuration Agent 将基于桌面用户的登录 ID 从 LDAP 服务器中检索信息。组织映射文件中的 User/UniqueIdAttribute 设置会将登录 ID 映射到 LDAP 服务器中的用户实体。Configuration Agent 还将检索有关主机的信息，例如主机的名称或 IP 地址。此类信息通过组织映射文件中的 Host/UniqueIdAttribute 设置映射到 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 凭证。要将 Kerberos 凭证的获得与用户登录集成在一起，必须在 Java Desktop System 主机上安装和配置 pam\_krb5 模块。

您可以使用 `gdm` 将 Kerberos 同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
```

如果您以这种方式将 Kerberos 同用户登录集成在一起，则您应启用屏幕保护程序的 Kerberos 支持。例如，使用以下 `/etc/pam.d/xscreensaver` 文件：

```
auth required pamkrb5.so use_first_pass missing_keytab_ok
ccache=SAFE putenv_direct
```

---

## GConf 适配器

GConf 适配器是用于 Solaris 的 `SUNWapoc-adapter-gconf` 软件包和用于 Linux 的 `apoc-adapter-gconf` RPM 的一部分。当您从相应的软件包或 RPM 安装适配器时，`/etc/gconf/2/path` 中的 GConf 数据源路径将被更新，以包括 Configuration Manager 源。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置。在用户设置之后、本地缺省设置之前插入此数据源。
- "apoc:readonly:mandatory@": 用来访问策略的保护设置。在本地强制设置之后、用户设置之前插入此数据源。

---

## Java Preferences 适配器

Java Preferences 适配器是用于 Solaris 的 `SUNWapcj` 软件包和用于 Linux 的 `apoc-adapter-java` RPM 的一部分。当您从相应的软件包或 RPM 安装适配器时，所需文件将被添加到 Solaris 上的 `/opt/SUNWapcj` 目录中或 Linux 上的 `/opt/apocjava` 目录中。

---

## Mozilla 适配器

Mozilla 适配器是用于 Solaris 的 SUNWmozapoc-adapter 软件包和用于 Linux 的 mozilla-apoc-integration RPM 的一部分。在从相应的软件包或 RPM 安装该适配器时，所需文件被加入到现有的 Mozilla 安装中，并自动注册。

---

## StarSuite 适配器

StarSuite 适配器包含在标准 StarSuite 安装中，并且允许您访问策略配置数据而无需进行任何特殊的修改。



## 第 6 章

---

# 在 Microsoft Windows 上安装桌面组件

---

Java Desktop System 附带了集中配置功能的所有必要组件，并将在安装过程中安装这些组件。通常，您可以在其他系统上使用集中配置管理。但在这种情况下，您必须自己执行必要的安装步骤。为了简化在 Microsoft Windows 系统下进行的设置，我们提供了基于图形用户界面的安装程序，它将会指导您完成整个安装步骤。

---

## Configuration Agent 安装

---

**注意** – 本安装程序和 Configuration Agent 都要求您已安装了 Java™ Runtime Environment (JRE) 1.4 或更高版本。JRE 不是本安装的一部分。如果您无法确定已安装的 Java 的版本，请执行 `java -version`。该命令将会通知您，系统是否安装了正确的 Java 版本以及是否已对其进行了正确的配置。

---

该安装程序由批处理文件 (`setup.bat`) 启动。您可以从 Java Desktop System Management Tools CD 上找到该文件。您必须作为具有足够管理权限的用户 (**local administrator**) 来启动该脚本。否则，将无法安装和启动作为新系统服务的 Configuration Agent。

键入以下内容以启动安装程序

```
setup [-help]
      [-saveState filename] [-no]
      [-noconsole] [-state filename]
      [-nodisplay]
```

安装程序可以识别以下选项：

- help            显示此帮助文本。
- saveState     将交互式安装会话中的输入写入到指定文件中。缺省文件名为 statefile.out。
- state          使用指定的状态文件作为无提示安装的输入。此选项需要与 -noconsole 选项配套使用。
- noconsole     启动不带任何用户界面的安装程序。与 -state 一起使用此选项，以初始化无提示安装。
- nodisplay     在命令行模式中启动安装程序，但不启动图形用户界面。
- no             运行安装程序，但不安装软件。该选项与 -saveState 一起使用非常有用，可为无提示安装做准备。

## 欢迎使用

调用 Configuration Agent 安装向导后，将出现一个“欢迎使用”页。此“欢迎使用”页提供有关安装过程的一些信息。

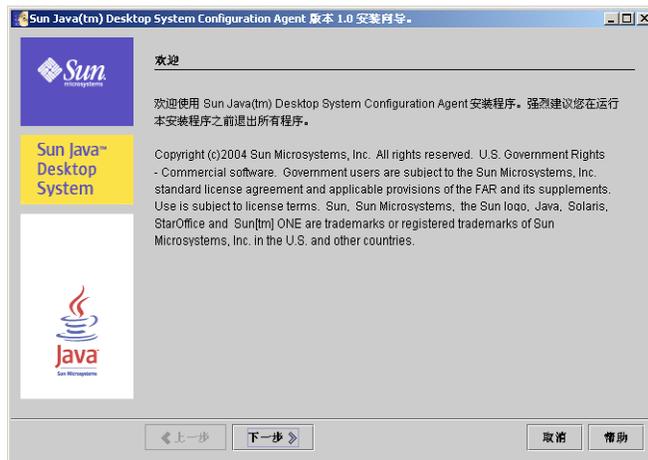


图 6-1 Configuration Agent: “欢迎使用”页

单击“下一步”按钮以进入向导的下一个页面。

## 许可协议

“软件许可协议”页包括有关 Configuration Agent 的许可信息。



图 6-2 Configuration Agent，软件许可协议

请阅读许可协议，然后单击“是 (接受许可协议)”按钮以接受协议内容，并继续进行安装过程。

## 安装类型

安装程序提供三种安装类型。这些类型仅在几个配置设置方面有些不同，您可在安装过程中定义这些配置设置。



图 6-3 Configuration Agent，“安装类型”页

- **快速**：对于大多数用户，建议采用快速类型。安装程序仅列出要使 Configuration Agent 正常运行所必需的设置。所有剩余设置都被设置为合理的缺省值。
- **自定义**：将列出所有可用于 Configuration Agent 的设置，从而允许您调整代理的行为。仅建议高级用户使用这种类型。
- **最小**：将安装 Configuration Agent，但不会提示您提供任何配置。在完成了安装之后，您必须通过启动配置向导来配置 Configuration Agent。

在选择了所需的安装类型后，请单击“下一步”。根据所选的安装类型，系统将显示多个配置设置页。下节将介绍这些设置的说明。

---

**注意** – 如果您选定**快速**选项或**最小**选项，则并非以下所有配置设置页都会显示。

---

---

**注意** – 要启动 Configuration Agent，系统需要一些引导信息，例如主机名和 LDAP 服务器的端口。另外，您还可以定义多个高级设置，例如日志级别或数据目录。所有这些设置都可以在安装过程中进行指定。这些信息在一组特性文件（例如 `polycmgr.properties`、`apocd.properties` 和 `os.properties`）中进行维护。这些文件存储在本地 `Program Files\Sun\Apoc\lib` 目录中。您可以手动编辑这些特性文件，也可以使用 Configuration Agent 的配置向导来完成。

---

## Configuration Agent 向导和设置信息

本节准确地介绍了所有可用于 Configuration Agent 的配置设置。

该配置向导提供了一个图形用户界面，可以引导您完成 Configuration Agent 的必要设置。对于该向导的每个页面，都存在一个相应的帮助屏幕。

### 引导信息

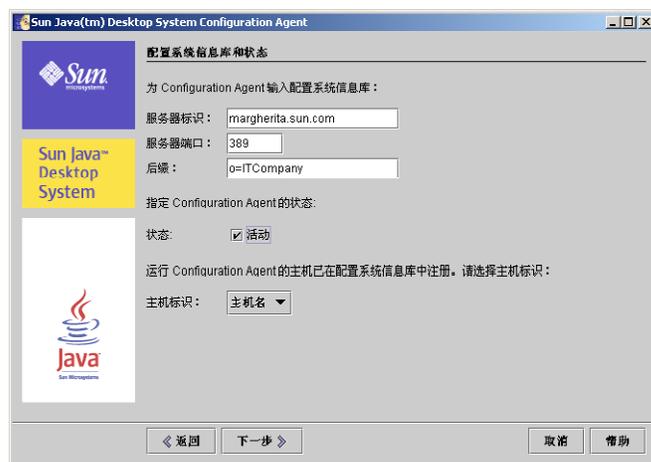


图 6-4 Configuration Agent，配置系统信息库

---

注意 – 在括号中指明了关联的特性文件关键字（如果适用）。

---

- 服务器标识 (Server)：LDAP 服务器的主机名。
- 服务器端口 (Port)：LDAP 服务器的端口号。
- 后缀 (BaseDn)：LDAP 系统信息库的基本 DN。
- 状态：Configuration Agent 的状态。可以使用该复选框激活或取消激活 Configuration Agent。要使用配置系统信息库，必须激活 Configuration Agent。在激活过程中，将自动向“服务控制管理器”进行必要的注册。

---

注意 – 要手动启用或禁用 Configuration Agent，请作为管理员登录，并分别键入命令 `Program Files\Sun\Apoc\bin\apocd enable` 或 `Program Files\Sun\Apoc\bin\apocd disable`。

---



图 6-5 Configuration Agent: 验证机制

- **主机标识符 (HostIdentifier)**: 可以是“HostName”或“IPAddress”。必须将该标识符设置为与用来标识主机的 LDAP 属性的内容相匹配。此属性在映射文件中定义为 Host/UniqueIdAttribute。
- **Configuration Agent 的验证类型**: 可以是“匿名”或“简单”。如果选定了“匿名”，那么**限定用户名**和**密码**字段将自动禁用。
- **限定用户名 (AuthDn)**: 具有读取和搜索系统信息库权限的用户的完整 DN。

---

注意 – 如果在目录中启用了匿名访问，那么此设置可以保留为空。

---

- **密码 (Password)**: 已注册 LDAP 用户的密码

---

注意 – 如果在目录中启用了匿名访问，那么此设置可以保留为空。

---

- **应用程序的验证类型 (AuthType)**: 可以是“匿名”或“GSSAPI”，这取决于 LDAP 服务器验证用户的方式。

## 端口设置

Configuration Agent 使用以下两个端口：

- **代理端口 (DaemonPort)**: 代理使用该端口与客户端应用程序进行通信（缺省值为 38900）。

- **管理端口 (DaemonAdminPort)**: 在与代理通信时, 代理控制器程序 (apocd.exe) 所用的端口 (缺省值为 **38901**)。



图 6-6 Configuration Agent: 端口设置

## 更改检测间隔

Configuration Agent 使用以下两种间隔定期检查配置数据中的更改:

- **常规检测间隔 (ChangeDetectionInterval)**: 桌面应用程序 (客户端) 的配置数据的两个更改检测周期之间的间隔 (以分钟为单位)。

---

**注意** – 指定 **-1** 将关闭更改检测。

---

- **代理设置的间隔 (DaemonChangeDetectionInterval)**: 代理特定的配置设置的两个更改检测周期之间的间隔 (以分钟为单位)。

---

**注意** – 指定 **-1** 将关闭更改检测。

---

可以使用常规检测间隔来调整远程配置数据更改至客户端应用程序的传播。为此设置指定的值表示最长经过多少分钟后, 远程更改才反映在客户端应用程序中。

值越小, Configuration Agent 和 LDAP 服务器活动就越多。因此, 调整此设置的值时应谨慎。例如, 在初始部署阶段, 可以将该值设置为一分钟, 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, 请将此设置还原为初始值。

## 操作设置

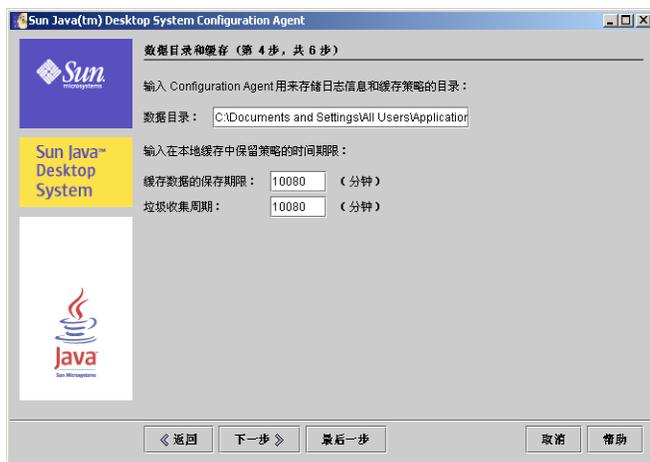


图 6-7 Configuration Agent: 数据目录

可以配置以下设置：

- **数据目录 (DataDir)**: 用于存储运行时数据的目录。
- **缓存数据的保存期限 (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)。



图 6-8 Configuration Agent: 请求处理和日志记录

- **垃圾收集周期 (GarbageCollectionInterval)**: 本地配置数据库中两个垃圾收集周期之间的间隔 (以分钟为单位)。

- **客户端线程的最大数目 (MaxClientThreads)**: 可以同时处理的客户端请求的最大数目。
- **客户端连接的最大数目 (MaxClientConnections)**: 客户端连接的最大数目。
- **最大请求大小 (MaxRequestSize)**: 客户端请求的最大大小。
- **连接超时 (ConnectTimeout)**: 表示 LDAP 服务器响应连接请求时所允许的间隔。缺省值为一秒。
- **日志级别 (LogLevel)**: 代理日志文件中的详细程度。此日志级别应与 Java 记录程序的级别保持一致。按严重性递减的顺序, 这些级别为:
  - 关闭
  - 严重
  - 警告
  - 信息
  - 配置
  - 良好
  - 较好
  - 最好
  - 全部

---

注意 – 大多数操作设置（**数据目录**和**连接超时**除外）也可以通过 LDAP 服务器中存储的相应策略进行集中维护。如果要使用此功能, 请不要通过向导更改相应的设置。相反, 应使用 Configuration Manager 中的 Configuration Agent 策略来集中指定操作设置。

---

## 代理设置汇总

已通过 Configuration Manager 存储在 LDAP 服务器中的操作设置（“数据目录”和“连接超时”除外）将在代理配置的下一个更改检测周期自动生效（请参见 DaemonChangeDetectionInterval）。



图 6-9 Configuration Agent, “汇总”页面

在本地更改的所有其他设置均需要重新装入 Configuration Agent, 或者重新启动 Configuration Agent。如果使用配置向导, 将自动执行重新装入和重新启动操作。

---

**注意** – 要手动重新启动 Configuration Agent, 请确保没有运行相关的客户端应用程序, 作为管理员登录, 并键入命令 `Program Files\Sun\Apoc\bin\apocd restart`。或者, 您可以在“控制面板”的“管理工具”下启动“服务控制器”。从可用服务的列表中选择 **Configuration Agent**, 然后单击“重新启动服务”。

---

## 准备安装

“准备安装”页显示 Configuration Agent 将要安装的项目。



图 6-10 Configuration Agent, “准备安装”页

此时, 尚未在您的系统上进行任何更改。只有当您单击“立即安装”按钮后, 才会开始实际的安装。

## 安装汇总

向导的“安装汇总”页显示已安装项目的状态。单击相应项目旁边的“详细信息”按钮, 可以提供有关该项目的详细信息。

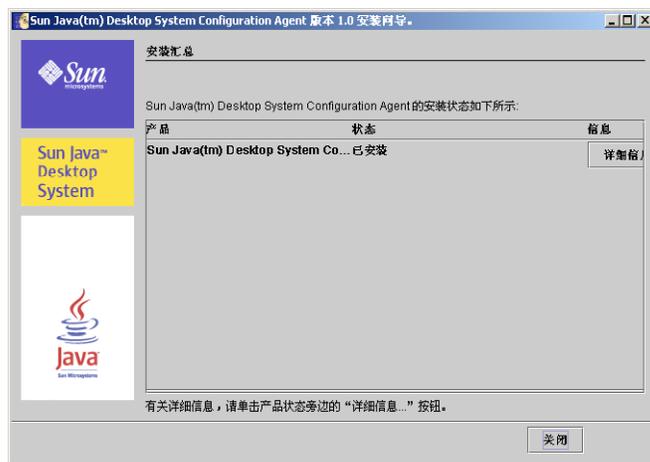


图 6-11 Configuration Agent, “安装汇总”页

安装程序将 Configuration Agent 二进制文件存储在缺省程序文件目录（通常为 C:\Program Files）中的 Sun\apoc 目录下。安装程序还自动在“开始”-“所有程序”-“Sun”菜单中创建一个新的 Java Desktop System 文件夹。此文件夹包含两个快捷方式，允许您进一步配置或卸载 Configuration Agent。

---

## 更改 Configuration Agent 设置

如上所述，您可以在安装后随时修改 Configuration Agent 设置。您可以通过 Program Files\Sun\Apoc\bin\apoc-config.bat 脚本，作为管理员启动向导。或者，您可以在“开始”菜单下的“所有程序”-“Sun”-“Java Desktop System”中选择 **Configuration Agent** 条目来启动 Configuration Agent 向导。

---

**注意** - 也可以启动该向导而不启动图形界面。例如，执行 Program Files\Sun\Apoc\bin\apoc-config -nodisplay，以在控制台模式中启动向导。

---

---

## 卸载 Configuration Agent

要卸载 Configuration Agent，请从“开始”菜单中选择“所有程序”-“Sun”-“Java Desktop System”- **Configuration Agent** -“卸载 Configuration Agent”。

## 附录 A

---

# Sun Web Console 软件包

---

## 已知问题

### 安全性

某些用户操作可能会使会话保持活动状态，而用户并不知道。例如，当用户关闭浏览器窗口时，并不会自动从 Sun Web Console 中注销。相反，用户必须先显式从 Sun Web Console 会话中注销，然后再关闭应用程序窗口。

## 安装脚本用法

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

-h = 打印用法语句

-n = 安装结束后不启动服务器

-u = 卸载 Sun Web Console

-f = 如果 Tomcat 和 Java 1.4 软件包是与设置应用程序一起安装的，则卸载这两个软件包。此参数只能与 -u 参数一起使用。

要获得可用设置参数的完整描述，请运行 `setup -h`。

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

---

注意 - 当您位于 `/usr/lib/webconsole` 目录或任何相关的子目录中时，请勿运行此命令，否则 `pkgm` 将失败。

---

---

## Sun Web Console 软件包

### Solaris 软件包

软件包名称	描述
SUNWmctag	Sun Web Console UI 标记库
SUNWmcon	Sun Web Console
SUNWmcos	用于 Sun Web Console 的通用 Solaris 服务
SUNWmcosx	用于 Sun Web Console 的 Solaris 发行版特定的服务
SUNWmconr	Sun Web Console 超级用户
SUNWjato	Sun One Application Framework 运行时环境
SUNWtcatu	Tomcat

### Linux RPM

软件包名称	描述
SUNWmctag	Sun Web Console UI 标记库
SUNWmcon	Sun Web Console
SUNWmcos	用于 Sun Web Console 的通用 Linux 服务
SUNWmcosx	用于 Sun Web Console 的 Linux 发行版特定的服务
SUNWmconr	Sun Web Console 超级用户

软件包名称	描述
SUNWjato	Sun One Application Framework 运行时环境
tomcat4	Tomcat



## 附录 B

---

# Configuration Manager 软件包

---

## Configuration Manager 软件包

### Solaris 软件包

软件包名称	描述
SUNWapm	Configuration Manager
SUNWapmca	Configuration Agent 模板
SUNWapmev	Evolution 模板
SUNWapg26	Gnome 2.6 模板
SUNWapmmo	Mozilla 模板
SUNWapmso	StarSuite 模板

### Linux RPM

软件包名称	描述
apoc-manager	Configuration Manager
apoc-agent-templates	Configuration Agent 模板

软件包名称	描述
apoc-evolution-templates	Evolution 模板
26-templates	Gnome 2.6 模板
apoc-mozilla-templates	Mozilla 模板
apoc-starSuite-templates	StarSuite 模板

# 在 Configuration Manager 中使用 OpenLDAP 和 Active Directory

---

---

## 在 Configuration Manager 中使用 OpenLDAP 服务器

要将 OpenLDAP 服务器用作 Configuration Manager 数据的系统信息库，必须扩展服务器模式，以支持用来存储配置数据的对象类和属性。可以在 Java Desktop System Management Tools CD 中提供的 Configuration Manager 部署工具的 `openldap` 子目录中找到名为 `apoc.schema` 的自定义模式文件。

必须将此文件复制到 OpenLDAP 配置目录 (`/etc/openldap`) 的 `schema` 子目录中，并通过在 OpenLDAP 配置目录中的 `slapd.conf` 文件中包括此自定义模式文件来将其添加到 OpenLDAP 模式中。此操作可以通过在包含该文件所显示内容的模式序列的后面插入 `include /etc/openldap/schema/apoc.schema` 这样一行来完成。有关扩展 OpenLDAP 服务器模式的详细信息，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 Configuration Manager 附带的部署工具。因为在安装过程的前一步骤中已经扩展了此模式，所以只需运行 `createServiceTree` 脚本。必须从部署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`。此脚本提示用户输入在本文档中的部署工具部分所指定的 OpenLDAP 数据库的有关信息。部署工具的 `openldap` 子目录中提供了一个缺省的映射文件，该映射文件使用 OpenLDAP 支持的典型对象类和属性。该文件的名称为 `OrganisationalMapping`，且可以通过以下方法部署该文件：在启动 `createServiceTree` 之前复制该文件，使其覆盖主部署工具目录中具有相同名称的文件。

---

**注意** – Configuration Manager Agent 将通过提供所需数据的用户的 DN（但不提供密码），尝试匿名连接 OpenLDAP 服务器。在某些版本的 OpenLDAP 服务器中，这种匿名验证模式缺省情况下可能已被禁用，在这种情况下，必须通过在通用服务器参数中添加 `allow bind_anon_cred` 这样一行来启用此模式，通用服务器参数在 OpenLDAP 配置目录 (`/etc/openldap`) 中的 `slapd.conf` 文件中定义。有关该参数的详细信息，请参阅该服务器的手册。

---

## 在 Configuration Manager 中使用 Active Directory 服务器

要将 Active Directory 服务器用作 Configuration Manager 数据的系统信息库，必须扩展服务器模式，以支持用来存储配置数据的对象类和属性。可以在 Management Tools CD 中提供的 Configuration Manager 部署工具的 `ad` 子目录中找到名为 `apoc-ad.ldf` 的模式扩展文件。有关详细信息，请参阅部署工具部分。

必须通过以下步骤将 `apoc-ad.ldf` 文件导入 Active Directory 模式：

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

---

**注意** – 将 `<BaseDN>` 替换为 Active Directory 的基本 DN。

---

如果要使 Active Directory 服务器用来存储配置数据，则必须使用部署工具。因为在安装过程的前一步骤中已经扩展了此模式，所以只需运行 `createServiceTree` 脚本。必须从部署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。部署工具目录的 `ad` 子目录中提供了一个缺省的映射文件，该映射文件使用 Active Directory 支持的典型对象类和属性。此文件名为 `OrganisationalMapping`，可以在启动 `createServiceTree` 之前，通过将其复制到主部署工具目录中以覆盖该目录中的同名文件来对其进行部署。

部署之后，即可在 Configuration Manager 中使用 Active Directory 服务器了。安装 Configuration Manager 时，请提供对该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。此外，Active Directory 的域名对于运行 Configuration Manager 的计算机来说必须是已知的。要实现此目的，可以在该计算机的 `/etc/hosts` 文件中添加一行内容，将 Active Directory 服务器的 IP 地址映射到其域名。

要从 Java Desktop System 主机检索配置数据，Active Directory 的域名对于该主机来说也必须是已知的。可以通过两种方法验证 Java Desktop System 用户：匿名验证和使用 GSSAPI 进行验证。

- 如果要使用匿名连接进行验证，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 进行验证，则必须修改指定 Kerberos 参数的 `/etc/krb5.conf` 文件，以定义 Active Directory 区域并指向 Active Directory 服务器作为其密钥分发中心 (KDC)。还必须指定 Active Directory 支持的、作为缺省加密类型的 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。访问配置数据之前，必须获取登录 Java Desktop System 的用户的有效凭证。这可以通过运行 `kinit` 命令并提供 Active Directory 中定义的用户密码来手动实现。登录时，其他类型可以自动生成这些凭证。有关详细信息，请参阅 Java Desktop System 文档。

