Sun Java System Access Manager 7.1 管理指南

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

文件号码 820–0839

070523@17466

目录

前言

《Sun Java System Access Manager 7.1 管理指南》说明如何使用 Sun Java™ System Access Manager 控制台以及如何通过命令行界面管理用户和服务数据。

Access Manager 是软件组件集合即 Sun Java Enterprise System (Java ES) 的一个组件，这些组件提供用于支持分布在网络或 Internet 环境中的企业应用程序所需的服务。

目标读者

本书的目标读者为使用 Sun Java System 服务器和软件实现 Web 访问平台的 IT 管理员和软件开发者。

阅读本书之前

读者应该熟悉下列组件和概念：

- 《Sun Java System Access Manager 7.1 Technical Overview》中描述的 Access Manager 技术概念。
- 部署平台： Solaris™ 或 Linux 操作系统
- 运行 Access Manager 的 Web 容器： Sun Java System Application Server、Sun Java System Web Server、BEA WebLogic 或 IBM WebSphere Application Server
- 技术概念：轻量目录访问协议 (Lightweight Directory Access Protocol, LDAP)、Java 技术、JavaServer Pages™ (JSP) 技术、超文本传输协议 (HyperText Transfer Protocol, HTTP)、超文本标记语言 (HyperText Markup Language, HTML) 和可扩展标记语言 (eXtensible Markup Language, XML)

相关文档

可在如下位置获得相关文档：

- 第 12 页中的 “Access Manager 核心文档”
- 第 13 页中的 “Sun Java Enterprise System 产品文档”

Access Manager 核心文档

Access Manager 核心文档集包含以下文档：

- 《Sun Java System Access Manager 7.1 发行说明》，可在产品发行后联机获取。该文档中收集了各种最新的信息，包括当前发行版的新功能说明、已知问题和限制、安装说明，以及报告有关软件或文档的问题的方法。
- 《Sun Java System Access Manager 7.1 Technical Overview》简单论述了 Access Manager 组件如何协同工作以整合访问控制功能，并保护企业资产和基于 Web 的应用程序。同时也解释了 Access Manager 的基本概念和术语。
- 《Sun Java System Access Manager 7.1 Deployment Planning Guide》基于解决方案生命周期，为 Sun Java System Access Manager 提供规划和部署解决方案。
- 《Sun Java System Access Manager 7.1 Postinstallation Guide》提供了有关在安装后配置 Access Manager 的信息。
- 《Sun Java System Access Manager 7.1 Performance Tuning Guide》提供关于如何微调 Access Manager 及其相关组件以获取最佳性能的信息。
- 《Sun Java System Access Manager 7.1 管理指南》描述如何使用 Access Manager 控制台以及如何通过命令行界面管理用户和服务数据。
- 《Sun Java System Access Manager 7.1 Federation and SAML Administration Guide》提供有关基于“Liberty 联盟计划”规范的“联合”模块的信息。其中包括有关基于这些规范的集成服务的信息、启用基于 Liberty 的环境的说明，并简单介绍了扩展框架的应用程序接口 (application programming interface, API)。
- 《Sun Java System Access Manager 7.1 Developer’s Guide》讲述如何自定义 Access Manager 以及将其功能集成到组织当前的技术架构中。本指南还包含关于本产品及其 API 的程序设计方面的详细信息。
- 《Sun Java System Access Manager 7.1 C API Reference》概述了组成公共 Access Manager C API 的数据类型、结构和函数。
- 《Java API Reference》提供有关在 Access Manager 中实现 Java 软件包的信息。
- 《Sun Java System Access Manager Policy Agent 2.2 User’s Guide》简介 Access Manager 可用的策略功能和策略代理程序。

在 Sun Java Enterprise System 文档站点的 Access Manager 页面中有发行说明的更新以及指向核心文档更正的链接。已更新的文档将会标记上修订日期。

Sun Java Enterprise System 产品文档

可在该文档中获得有关下列产品的有用信息：

- Directory Server
- Web Server
- Application Server
- Web Proxy Server

相关的第三方 Web 站点引用

本文档引用第三方 URL，并提供其他相关信息。

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

文档、支持和培训

Sun Web 站点提供有关以下附加资源的信息：

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

印刷约定

下表介绍了本书中使用的印刷约定。

表 P–1 印刷约定

字体	含义	示例
`AaBbCc123`	命令、文件和目录的名称；计算机屏幕输出	编辑 `.login` 文件。 使用 `ls -a` 列出所有文件。 `machine_name% you have mail.`

表 P–1 印刷约定 （续）

字体	含义	示例
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	保留未译的新词或术语以及要强调的词。要使用实名或值替换的命令行变量。	删除文件的命令为 `rm` *filename*。 *cache* 是在本地存储的副本。
新词术语强调	新词或术语以及要强调的词。	请**勿**保存文件。 **注意**：某些强调项联机显示为粗体。
《书名》	书名	阅读《用户指南》的第 6 章。

命令中的 shell 提示符示例

下表显示了 C shell、Bourne shell 和 Korn shell 的默认 UNIX® 系统提示符和超级用户提示符。

表 P–2 Shell 提示符

Shell	提示符
C shell 提示符	`machine_name%`
C shell 超级用户提示符	`machine_name#`
Bourne shell 和 Korn shell 提示符	`$`
Bourne shell 和 Korn shell 超级用户提示符	`#`

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。

如果您要提出意见，请转到 `http://docs.sun.com`，然后单击 Send Comments（发送意见）。请在联机表单中提供文档标题和文件号码。文件号码通常包含七位或九位数字，您可以在本书的标题页或文档最上部找到文件号码。

例如，本书的标题为《Sun Java System Access Manager 7.1 管理指南》，文件号码为 820-0839。提出意见时您还需要在表格中输入文件的英文文件号码和标题，本文件的英文文件号码是 819-4670-10，文件标题为《Sun Java System Access Manager 7.1 Administration Guide》。

第 1 部分

访问控制

这是《Sun Java System Access Manager™ 7.1 管理指南》的第一部分。 “访问控制”界面提供了一种创建和管理验证和授权服务的途径，从而保护和控制基于领域的资源。当企业用户请求信息时，Access Manager 会验证用户身份并授权用户访问其所请求的特定资源。本部分包含以下各章：

◆ ◆ ◆ 第 1 章

1 Access Manager 控制台

Access Manager 控制台是一个 Web 界面，它允许拥有不同访问级别的管理员（包括做其他事情）创建领域和组织、在这些领域中创建或删除用户，以及建立可保护和限制领域资源访问的强制策略。此外，管理员还可以查看和终止当前用户会话以及管理它们的联合配置（创建、删除和修改验证域和提供商）。另一方面，不拥有管理权限的用户可以管理个人信息（姓名、电子邮件地址、电话号码等）、更改他们的密码、订阅和取消订阅组以及查看他们的角色。 Access Manager 控制台拥有两个基本视图：

- 第 17 页中的 “管理视图”
- 第 20 页中的 “用户概要文件视图”

管理视图

拥有管理角色的用户通过 Access Manager 进行验证时，默认视图为“管理视图”。在该视图中，管理员可以执行与 Access Manager 相关的大多数管理任务。Access Manager 可以在两种不同模式（“领域”模式和“传统”模式）下进行安装。每种模式均拥有其各自的控制台。有关“领域模式”和“传统模式”的详细信息，参见《Sun Java System Access Manager 7.1 Technical Overview》。

注 – 如果在“领域模式”下安装 Access Manager 7.1，则无法回到“传统模式”。如果在“传统模式”下安装 Access Manager，则可以通过使用 amadmin 命令更改为“领域模式”。有关详细信息，参见 Access Manager 管理参考中的“Changing from Legacy Mode to Realm Mode”。

领域模式控制台

领域模式中的管理控制台使管理员能够管理基于领域的访问控制、默认服务配置、Web 服务以及联合。要访问管理员登录屏幕，请在浏览器中使用以下地址语法：

protocol`://`*servername* `/amserver/UI/Login`

protocol 可以为 http 或 https，具体取决于您的部署。

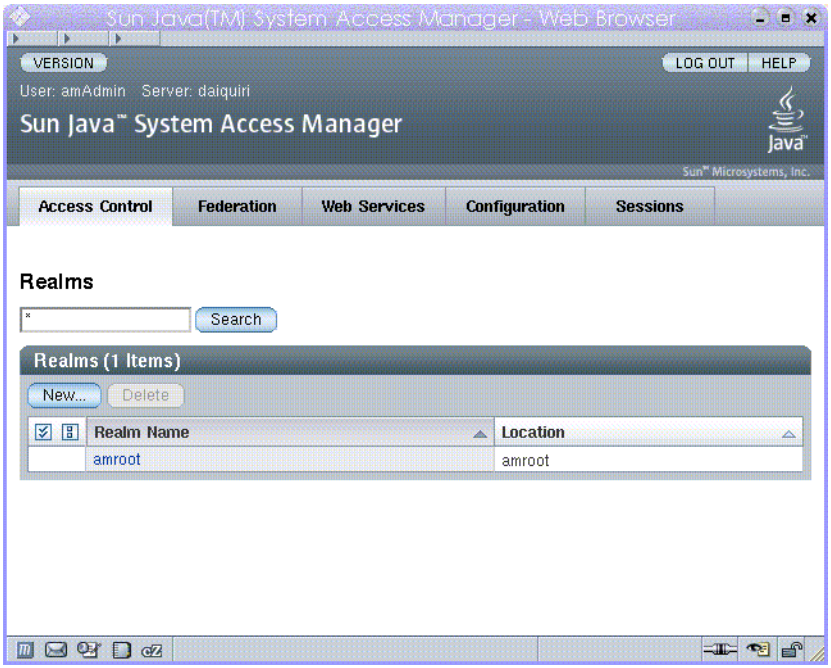

图 1–1 领域模式管理视图

传统模式控制台

“传统模式”控制台是基于 Access Manager 6.3 体系结构的。此传统 Access Manager 体系结构使用 Sun Java System Directory Server 自带的 LDAP 目录信息树 (DIT)。在“传统模式”下，用户信息和访问控制信息均存储于 LDAP 组织中。选择“传统模式”时，LDAP 组织等同于访问控制领域。领域信息集成在 LDAP 组织内。在“传统模式”下，“目录管理”选项卡可在基于 Access Manager 的身份管理中使用。

要访问管理员登录屏幕，请在浏览器中使用以下地址语法：

protocol://*servername* /amserver/console

protocol 可以为 http 或 https，取决于您的部署。

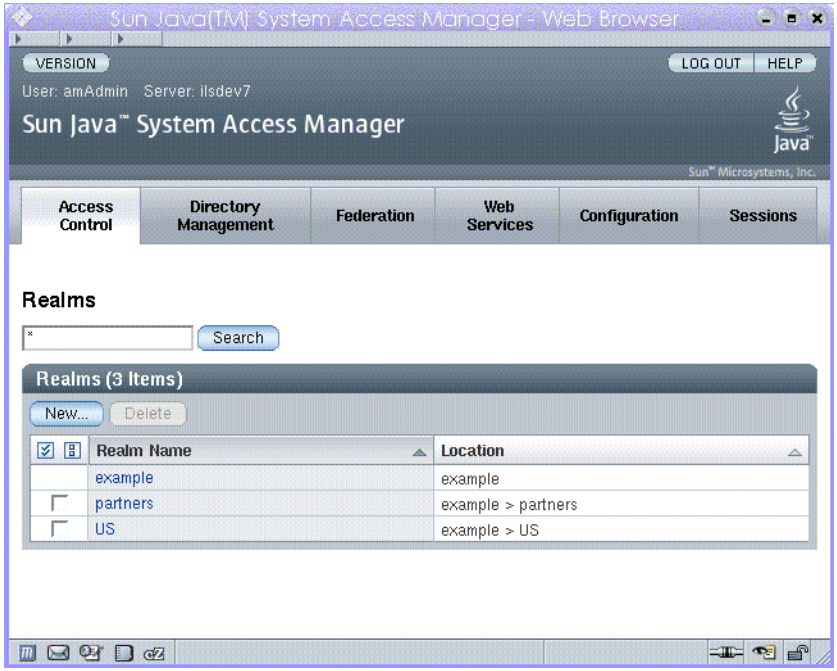

图 1–2 传统模式管理视图

传统模式 6.3 控制台

Access Manager 6.3 的某些功能在 Access Manager 7.1 控制台中不可用。因此，管理员可以通过 7.1 传统部署登录到 6.3 控制台。在将 Access Manager 建立在 Sun Java System Portal Server 或其他需要将 Sun Java System Directory Server 用作中心身份库的 Sun Java System 通信产品上的情况下，通常会使用此控制台。其他功能（如“委托管理”和“服务类”）只能通过此控制台进行访问。

注 – 请勿交换使用 6.3 传统模式控制台和 7.1 传统模式控制台。

要访问 6.3 控制台，请在浏览器中使用以下地址语法：

protocol`://`*servername* `/amconsole`

`protocol` 可以为 http 或 https，取决于您的部署。

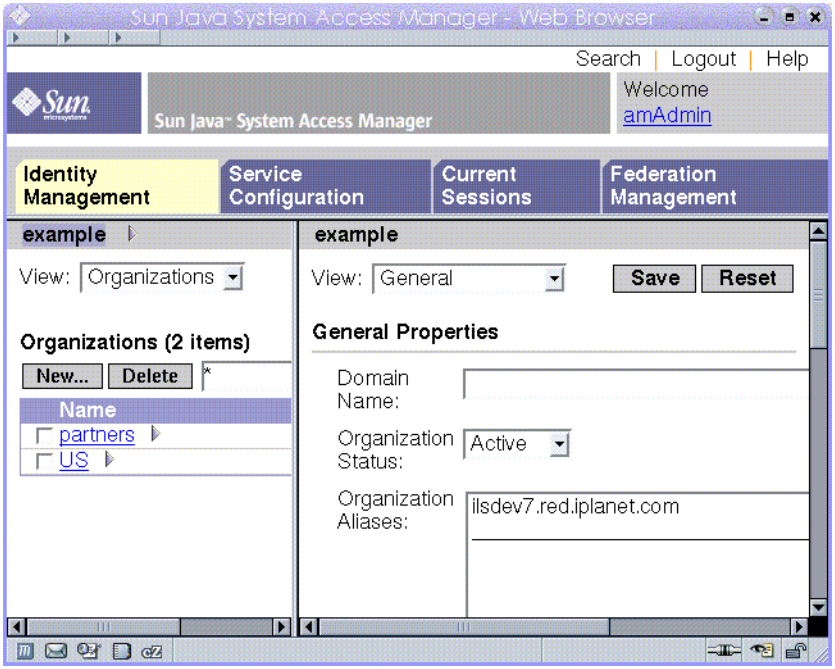

图 1–3 基于传统 6.3 的控制台

用户概要文件视图

当尚未指定管理角色的用户向 Access Manager 进行验证时，默认视图为用户自己的“用户概要文件”视图。在“领域模式”或“传统模式”下均可访问“用户概要文件”视图。要访问该视图，用户必须在“登录”页面中输入自己的用户名和密码。

用户可以在该视图中修改用户的个人配置文件所特有的属性值。其中包括（但不限于）姓名、家庭地址和密码。“用户概要文件视图”中显示的属性可以扩展。

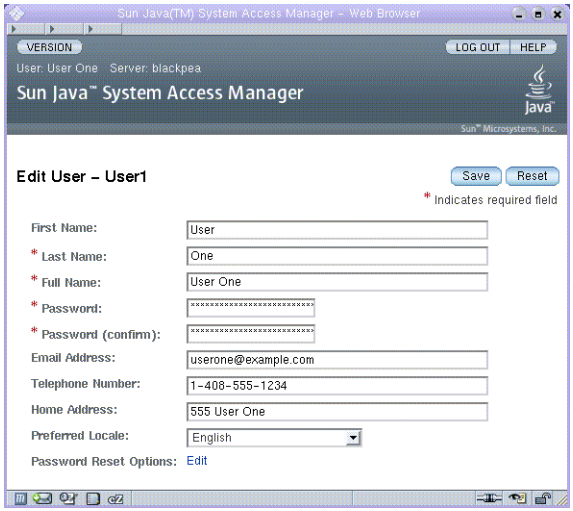

图 1–4 用户概要文件视图

◆ ◆ ◆ 第 2 章

2 管理领域

访问控制领域是可以与用户或用户组关联的一组验证属性和授权策略。领域数据存储于专有信息树中，该信息树由 Access Manager 在您指定的数据存储库中创建。Access Manager 框架在 Access Manager 信息树中聚集了每个领域所包含的策略和属性。默认情况下，Access Manager 会将 Access Manager 信息树作为特殊分支插入到 Sun Java Enterprise System Directory Server 中，但用户数据除外。您可以在使用任何 LDAPv3 数据库的同时使用访问控制领域。

有关领域的详细信息，参见《Sun Java System Access Manager 7.1 Technical Overview》。

在“领域”选项卡中，可以为访问控制配置以下属性：

- 第 24 页中的 “验证”
- 第 25 页中的 “服务”
- 第 26 页中的 “权限”

创建和管理领域

本节说明了如何创建和管理领域。

▼ 创建新的领域

1 **从“访问控制”选项卡下的“领域”列表中选择“新建”。**

2 **定义以下常规属性：**

名称　　输入领域名称。

父领域　　定义要创建的领域的位置。选择要在其中创建新领域的父领域。

3 **定义以下领域属性：**

领域状态	选择“活动”或“不活动”状态。默认值为“活动”。在领域存在期间，可以随时选择属性图标以更改其状态。如果选择“不活动”，则当登录时，将禁止用户访问。
领域/DNS 别名	允许为领域的 DNS 名称添加别名。该属性只接受“真实的”域别名（不允许使用随机字符串）。

4 **请单击“确认”保存或单击“取消”返回上一页面。**

常规属性

“常规属性”页面显示领域的基本属性。要修改这些属性，请从“访问控制”选项卡下的“领域名称”列表中单击该领域。然后编辑以下属性：

领域状态	选择“活动”或“不活动”状态。默认值为“活动”。在领域存在期间，可以随时选择属性图标以更改其状态。如果选择“不活动”，则当登录时，将禁止用户访问。
领域/DNS 别名	允许为领域的 DNS 名称添加别名。该属性只接受“真实的”域别名（不允许使用随机字符串）。

编辑属性后，单击“保存”。

注 – `AMAdmin.dtd` 中的 `recursive=true` 标志对于以领域模式在子领域中搜索对象不起作用。该标志只能在传统模式中发挥作用，因为所有子组织均位于相同的根后缀下。在领域模式下，每个子领域都可拥有不同的根后缀，甚至可以位于不同的服务器上。如果要在子领域中搜索对象（例如组），则必须在 XML 数据文件中指定要搜索的子领域。

验证

必须先将常规验证服务注册为领域的服务，用户才能使用其他验证模块登录。核心验证服务允许 Access Manager 管理员为领域的验证参数定义默认值。如果在指定的验证模块里没有定义替代值，那么便可使用这些值。核心验证服务的默认值在 `amAuth.xml` 文件中定义，并于安装结束后存储在 Directory Server 中。

有关详细信息，参见管理验证

服务

在 Access Manager 中，服务是由 Access Manager 控制台一起管理的一组属性。属性可以仅仅是一些相关信息，如雇员姓名、职务以及电子邮件地址。但是属性通常被用作软件模块（如邮件应用程序或工资单服务）的配置参数。

您可以通过“服务”选项卡在领域中添加并配置许多 Access Manager 默认的服务。您可以添加以下服务：

- 管理
- 搜索服务
- 全局化设置
- 密码重置
- 会话
- 用户

注 – Access Manager 强制要求服务 .xml 文件中的必需属性具备一些默认值。如果服务的必需属性没有值，则需要添加默认值并重新加载服务。

▼ 向领域添加服务

1. **单击要为其添加新服务的领域的名称。**
2. **选中“ 服务 ”选项卡。**
3. **单击“服务”列表中的“添加”。**
4. **选择要为领域添加的服务。**
5. **单击“ 下一步 ”。**
6. **通过定义领域属性来配置服务。有关服务属性的说明，参见联机帮助中的“配置”。**
7. **单击“完成”。**
8. **要编辑服务的属性，请在“服务”列表中单击其名称。**

权限

Access Manager 中的委托模型基于已指定给管理员的权限（或权利）。权限是可对资源执行的操作（或行为）；例如对“策略”对象执行的 READ 操作。一组已定义的操作是 READ、MODIFY 和 DELEGATE。资源是可对其执行操作的对象，可以是配置对象，也可以是身份对象。

配置对象的示例是“验证配置”、“策略”、“数据存储库”等。身份对象的示例是“用户”、“组”、“角色”和“代理”。可动态创建一组权限并动态将其添加到 Access Manager，不过在安装期间，会将少量权限添加到 Access Manager 以使其正常运行。一旦加载权限后，就可将其指定给角色和组。属于这些角色和组的用户就可成为委托管理员，并且可执行已指定的操作。管理员基本上就是一些特定的用户，他们是已指定了一组或多组权限的角色和组的成员。

可通过 Access Manager 7.1 为以下管理员类型配置权限：

- 领域管理员 — 领域管理员拥有对所有对象（配置对象和身份对象）执行 READ、MODIFY 和 DELEGATE 操作的权限。可将领域管理员视为 Unix 系统中的“超级用户”。领域管理员可为所有服务创建子领域，修改配置，也可创建、修改以及删除“用户”、“组”、“角色”和“代理”。
- 策略管理员 — 策略管理员仅拥有管理策略和策略服务配置的权限。他们可以创建、修改以及删除由“规则”、“主题”、“条件”和“响应”属性组成的策略。不过，为了管理策略，这些管理员需要拥有读取“身份系统信息库主题”和“验证配置”的权限。这些管理员能查看身份和验证配置。
- 日志管理员 — 日志管理员具有读取和/或写入日志的权限，这些权限可用于防止审计日志被恶意应用程序恶意滥用。日志记录接口是公共接口，任何通过验证的用户都可以读取和写入日志记录，添加此权限正是为防止日志记录被滥用。日志记录接口的主要用户是 J2EE 和 Web 代理，他们只需要 MODIFY 权限，但不应该拥有 READ 权限。类似地，查看日志的管理员只应该拥有 READ 权限，而不应该拥有 MODIFY 权限。为满足这些类型的用法，又进一步将日志记录权限细分为如下：
  - 拥有写入权限的日志管理员 – 这些管理员拥有写入所有日志文件的权限。
  - 拥有读取权限的日志管理员 – 这些管理员拥有读取所有日志文件的权限。
  - 拥有读写权限的日志管理员 – 这些管理员有权读取和写入所有日志文件。

定义 Access Manager 7.1 的权限

Access Manager 7.1 的新安装实例为策略管理员、领域管理员（或传统模式中的组织管理员）和日志管理员提供访问权限。单击您要进行编辑的角色或组的名称，可指定或修改权限。可选择的权限包括：

对所有日志文件的读写权限	为日志管理员定义读写访问权限。
对所有日志文件的写入权限	仅为日志管理员定义写入访问权限。

对所有日志文件的读取权限	仅为日志管理员定义读取访问权限。
仅针对策略属性的读写访问权限	为策略管理员定义读写访问权限。
所有领域和策略属性的读写访问权限	为领域管理员定义读写访问权限。

为从 Access Manager 7.0 升级到 7.1 定义权限

如果已将 Access Manager 从版本 7.0 升级到 7.1，那么相关权限配置与新 Access Manager 7.1 安装有所不同，但仍然支持策略管理员、领域管理员和日志管理员的权限。单击您要进行编辑的角色或组的名称，可指定或修改权限。可选择的权限包括：

对数据存储库的只读访问	为策略管理员定义数据存储库的读取访问权限。
对所有日志文件的读写权限	为日志管理员定义读写访问权限。
对所有日志文件的写入权限	仅为日志管理员定义写入访问权限。
对所有日志文件的读取权限	仅为日志管理员定义读取访问权限。
仅针对策略属性的读写访问权限	为策略管理员定义读写访问权限。
所有领域和策略属性的读写访问权限	为领域管理员定义读写访问权限。
所有属性和服务的只读访问权限	为策略管理员定义所有属性和服务的读取访问权限。

Access Manager 不支持以下定义（无论是单独使用还是一起使用）：

- 对数据存储库的只读访问
- 所有属性和服务的只读访问权限

这些权限定义必须和“仅针对策略属性的读写访问权限”定义一起使用，从而为策略管理员定义委托控制。

◆ ◆ ◆ 第 3 章

3 数据存储库

数据存储库是一个可以存储用户属性和用户配置数据的数据库。Access Manager 提供身份系统信息库插件，这些插件连接到 LDAPv3 身份系统信息库框架。这些插件使您不必在现有用户数据库中做任何更改，便可查看和检索 Access Manager 用户信息。Access Manager 框架将来自身份系统信息库插件的数据和其他 Access Manager 插件里的数据整合在一起，为每个用户形成了一个虚拟身份。而后，Access Manager 能使用通用身份在多个身份系统信息库之间进行验证和授权过程。在用户会话结束后将会销毁虚拟用户身份。

Access Manager 数据存储库类型

本节说明可配置的数据存储库类型，提供创建和配置新数据存储库类型的步骤。

可为以下数据存储库类型创建新的数据存储库实例：

Access Manager 系统信息库插件

该数据存储库类型驻留在 Sun Java System Directory Server 实例中，并拥有 Access Manager 信息树。该数据存储库类型使用不属于 LDAP 版本 3 规范的 Directory Server 功能（如角色和服务类），并与以前的 Access Manager 版本兼容。

活动目录

该数据存储类型使用 LDAP 版本 3 规范将身份数据写入到 Microsoft 活动目录的实例中。

平面文件系统信息库

该系统信息库允许用户在 Access Manager 的本地安装实例上以平面 DIT 结构存储数据和身份，而不必创建单独的数据存储库。通常将其用于测试或验证概念部署。

普通 LDAPv3

该数据存储库类型允许将身份数据写入任意与 LDAPv3 兼容的数据库。如果所使用的 LDAPv3 数据库不支持持久性搜索，则无法使用高速缓存功能。

使用 Access Manager 模式的 Sun Directory Server

该数据存储库类型驻留在 Sun Java System Directory Server 实例中，并拥有 Access Manager 信息树。它与 Access Manager 系统信息库插件不同，后者包含更多配置属性，从而可更好地自定义数据存储库。

▼ 创建新的数据存储库

以下部分描述连接数据存储库的步骤。

1. **选择要添加新数据存储库的领域。**
2. **单击“ 数据存储库 ”选项卡。**
3. **在“ 数据存储库 ”列表中单击“ 新建 ”。**
4. **请输入数据存储库的名称。**
5. **选择要创建的数据存储库的类型。**
6. **单击“ 下一步 ”。**
7. **输入适当属性值以配置数据存储库。**
8. **单击“完成”。**

数据存储库属性

本节定义用于配置每个新 Access Manager 数据存储库的属性。这些数据存储库属性是：

注 – Active Directory、普通 LDAPv3 以及使用 Access Manager 模式的 Sun Directory Server 数据存储库类型共享相同的基本插件，因此配置属性是相同的。然而，对于每个数据存储库类型而言，某些属性的默认值是不同的，在 Access Manager 控制台中会相应地显示这些默认值。

Access Manager 系统信息库属性

用于配置 Access Manager 系统信息库插件的属性如下：

类名称

指定实现 Access Manager 系统信息库插件的类文件的位置。

Access Manager 支持的类型和操作

指定允许或可以在该 LDAP 服务器上执行的操作。只有默认操作才是受此 LDAPv3 系统信息库插件支持的操作。LDAPv3 系统信息库插件支持以下操作：

- 组 -- 读取、创建、编辑、删除
- 用户 -- 读取、创建、编辑、删除、服务
- 代理 -- 读取、创建、编辑、删除

可根据 LDAP 服务器设置和任务从上述列表删除权限，但不能添加其他权限。

如果已配置的 LDAPv3 系统信息库插件指向 Sun Java Systems Directory Server 的实例，则可添加**角色**类型的权限。否则，由于其他数据存储库可能不支持角色，从而可能无法添加该权限。“角色”类型的权限为：

- 角色 — 读取、创建、编辑、删除

如果**用户**类型是 LDAPv3 系统信息库所支持的类型，则可对该用户执行读取、创建、编辑和删除服务操作。换句话说，如果支持**用户**类型，则通过读取、创建、编辑和删除操作便可分别从身份系统信息库中读取、创建、编辑和删除用户条目。`user=service` 操作会使 Access Manager 服务访问用户条目中的属性。此外，如果将动态服务指定给用户所属的领域或角色，则用户可以访问动态服务属性。

用户也可以管理任意指定服务的用户属性。如果用户将 service 作为操作 (user=service)，则指定了对所有与服务相关的操作提供支持。这些操作是 ：assignService、unassignService、getAssignedServices、getServiceAttributes、removeServiceAttributes 和 modifyService。

组织 DN 值

定义指向 Access Manager 要管理的 Directory Server 中组织的DN。它将成为在该数据存储库内执行的所有操作的基 DN。

人员容器命名属性

如果用户驻留在人员容器中，则指定该人员容器的命名属性。如果用户没有驻留在人员容器中，则将该字段保留为空。

人员容器值

指定人员容器的值。默认值为 people。

代理容器命名属性

如果代理驻留在代理容器中，则指定该代理容器的命名属性。如果代理没有驻留在代理容器中，则将该字段保留为空。

代理容器值

指定代理容器的值。默认值为 agents。

递归搜索

如果启用，在 Access Manager 系统信息库中执行的搜索会对指定身份进行递归搜索。例如，在以下数据结构中执行递归搜索 :

```
root
realm1
  subrealm11
    user5
  subrealm12
    user6
realm2
  user1
  user2
  subrealm21
    user3
    user4
```

会产生以下结果 :

- 如果从 root 开始执行搜索，且未在该级别定义任何用户（除 amadmin 和 anonymous 以外），则搜索返回 user 1–6。
- 如果从 realm1 开始执行搜索且未定义任何用户，则搜索返回 user5 和 user6。
- 如果从 realm2 开始执行搜索（已定义两个用户），则搜索返回 user 1–4。

复制领域配置

若在领域模式安装中启用了该属性，Access Manager 将为系统信息库中存在的每个领域和子领域创建等效组织及子组织。此外，在领域/子领域中注册的服务还将在新创建的组织/子组织中进行注册。领域 DIT 和组织 DIT 均存在于数据存储库内。

平面文件系统信息库属性

用于配置平面文件系统信息库的属性如下：

文件系统信息库插件类名称

该属性指定为平面文件提供实现的 Java 类文件。不应修改该属性。

文件系统信息库目录

定义用于存储身份及其属性的基目录。

高速缓存

若为启用（默认），将对身份及其属性进行高速缓存。这样，后续请求就不会访问文件系统。

更新高速缓存的时间

启用高速缓存后，该属性将确定检查高速缓存的时间间隔（以分钟为单位），超过该间隔便会对高速缓存中的条目进行检查，以确定是否对文件系统进行过任何更改。检查机制以时间戳为基础。

文件用户对象类

定义创建用户时自动为用户添加的对象类。

密码属性

提供包含用于验证的密码的属性名。 该属性用于在启用“数据存储库”验证模块时对用户进行验证。

状态属性

提供存储身份状态的属性名。状态属性的值为**活动**或**不活动**。该属性在身份验证期间使用。如果身份为**不活动**，则不验证用户。

散列的属性

提供属性列表，这些属性的值将散列并存储于文件中。执行散列后，便无法获取原始值。只能对散列的值进行检索。某些用于验证的属性不应永久存储，使用散列便可确保这些属性的保密性。例如，身份的密码属性就是此类型属性的例子。

加密的属性

提供属性列表，这些属性的值将加密并存储于文件中。虽然对其进行了加密和存储，但调用身份系统信息库 API 仍可返回加密前的原始值。这可防止用户直接访问文件系统并读取敏感属性。

LDAPv3 属性

用于配置 LDAPv3 系统信息库插件的属性如下：

LDAP 服务器

输入要连接的 LDAP 服务器的名称。格式应为 `hostname.domainname:portnumber`。

如果输入多个 `host:portnumber` 条目，则会尝试连接到列表中的第一个主机。仅当尝试连接当前主机失败时，才会尝试列表中的下一个条目。

LDAP 绑定 DN

指定 Access Manager 将用来向当前连接的 LDAP 服务器验证的 DN 名称。拥有用于绑定的 DN 名称的用户应具有在第 35 页中的 “LDAPv3 插件支持的类型和操作”属性中配置的正确的添加/修改/删除权限。

LDAP 绑定密码

指定 Access Manager 将用来向当前连接的 LDAP 服务器验证的 DN 密码。

LDAP 绑定密码（确认）

确认密码。

LDAP 组织 DN

该数据存储库将映射到的 DN。它将成为在此数据存储库内执行的所有操作的基 DN。

LDAP SSL

启用后，Access Manager 将使用 HTTPS 协议连接到主服务器。

LDAP 连接池的最小尺寸

指定连接池中的初始连接数。使用连接池可避免每次都必须创建新的连接。

LDAP 连接池的最大尺寸

指定允许的最大连接数。

搜索返回的结果的最大数目

指定搜索操作所返回的最大条目数。如果达到该限制，Directory Server 将返回与搜索请求相匹配的任何条目。

搜索超时

指定分配给搜索请求的最长时间（以秒计）。如果达到该限制，Directory Server 将返回与搜索请求相匹配的任何搜索条目。

LDAP 遵循引用

如果启用该选项，将指定自动遵循其他 LDAP 服务器的引用。

LDAPv3 系统信息库插件类名称

指定实现 LDAPv3 系统信息库的类文件的位置。

常规属性名称映射

将框架已知的公共属性映射到本地数据存储库。例如，如果框架使用 `inetUserStatus` 确定用户状态，则本机数据存储库实际可能会使用 `userStatus`。属性定义区分大小写。

LDAPv3 插件支持的类型和操作

指定允许或可以在该 LDAP 服务器上执行的操作。只有默认操作才是受此 LDAPv3 系统信息库插件支持的操作。LDAPv3 系统信息库插件支持以下操作：

- 组 -- 读取、创建、编辑、删除
- 用户 -- 读取、创建、编辑、删除、服务
- 代理 -- 读取、创建、编辑、删除

可根据 LDAP 服务器设置和任务从上述列表删除权限，但不能添加其他权限。

如果已配置的 LDAPv3 系统信息库插件指向 Sun Java Systems Directory Server 的实例，则可添加**角色**类型的权限。否则，由于其他数据存储库可能不支持角色，从而可能无法添加该权限。“角色”类型的权限为：

- 角色 — 读取、创建、编辑、删除

如果**用户**类型是 LDAPv3 系统信息库所支持的类型，则可对该用户执行读取、创建、编辑和删除服务操作。换句话说，如果支持**用户**类型，则通过读取、创建、编辑和删除操作便可分别从身份系统信息库中读取、创建、编辑和删除用户条目。`user=service` 操作会使 Access Manager 服务访问用户条目中的属性。此外，如果将动态服务指定给用户所属的领域或角色，则用户可以访问动态服务属性。

用户也可以管理任意指定服务的用户属性。如果用户将 service 作为操作 (user=service)，则指定了对所有与服务相关的操作提供支持。这些操作是 : assignService、unassignService、getAssignedServices、getServiceAttributes、removeServiceAttributes 和 modifyService。

LDAPv3 插件搜索范围

定义用于查找 LDAPv3 插件条目的范围。该范围必须为以下值之一 :

- SCOPE_BASE
- SCOPE_ONE
- SCOPE_SUB（默认值）

LDAP 用户搜索属性

该字段用于定义搜索用户时使用的属性类型。例如，如果用户 DN 为 uid=user1,ou=people,dc=iplanet,dc=com，则命名属性为 uid。

LDAP 用户搜索过滤器

指定用于查找用户条目的搜索过滤器。

LDAP 用户对象类

指定用户的对象类。创建用户之后，用户对象类列表将被添加到该用户的属性列表中。

LDAP 用户属性

定义与用户相关的属性列表。禁止对不在列表之内的用户属性进行任何读/写尝试。属性区分大小写。在这里定义对象类和属性模式之前，必须先在 Directory Server 中定义对象类和属性模式。

LDAP 用户创建属性映射

指定创建用户时需要哪些属性。此属性使用下列语法 :

```
DestinationAttributeName=SourceAttributeName
```

如果缺少源属性名，则默认为用户 ID (uid)。例如 :

```
cn
sn=givenName
```

创建用户概要文件时，cn 和 sn 都是必需的。cn 获取名为 uid 的属性的值，sn 则获取名为 givenName 的属性的值。

用户状态属性

指定属性名以表示用户状态。

用户状态活动值

指定活动用户状态的属性名。默认值为**活动**。

用户状态非活动值

指定不活动用户状态的属性名。默认值为**不活动**。

LDAP 组搜索属性

该字段用于定义搜索组时使用的属性类型。默认值为 `cn`。

LDAP 组搜索过滤器

指定用于查找组条目的搜索过滤器。默认值为 `(objectclass=groupOfUniqueNames)`。

LDAP 组容器命名属性

如果组驻留在容器中，则指定组容器的命名属性。否则，该属性保留为空。例如，如果 `cn=group1,ou=groups,dc=iplanet,dc=com` 的组 DN 位于 `ou=groups` 中，则组容器命名属性为 `ou`。

LDAP 组容器值

指定组容器的值。例如，如果 `cn=group1,ou=groups,dc=iplanet,dc=com` 的组 DN 位于容器名 `ou=groups` 中，则组容器值应为 `groups`。

LDAP 组对象类

指定组的对象类。创建组之后，组属性列表中会添加此组对象类列表。

LDAP 组属性

定义与组相关的属性列表。禁止对不在列表之内的组属性进行任何读/写尝试。属性区分大小写。在这里定义对象类和属性模式之前，必须先在 Directory Server 中定义对象类和属性模式。

组成员资格属性

指定属性名称，该属性的值为包含 DN 的所有组的名称。默认值为 `memberOf`。

唯一成员属性

指定属性名称，该属性的值是该组所包含的某个 DN。默认值为 `uniqueMember`。

组成员 URL 属性

指定属性名称，该属性的值是可解析为该组所包含成员的 LDAP URL。默认值为 `memberUrl`。

LDAP 人员容器命名属性

如果用户驻留在人员容器中，则指定该人员容器的命名属性。如果用户没有驻留在人员容器中，则将该字段保留为空。

LDAP 人员容器值

指定人员容器的值。默认值为 `people`。

LDAP 代理搜索属性

该字段用于定义搜索代理时使用的属性类型。默认值为 `uid`。

LDAP 代理容器命名属性

如果代理驻留在代理容器中，则指定该代理容器的命名属性。如果代理没有驻留在代理容器中，则将该字段保留为空。

LDAP 代理容器值

指定代理容器的值。默认值为 `agents`。

LDAP 代理搜索过滤器

定义用于搜索代理的过滤器。LDAP 代理搜索属性将被置于此字段之前，以构成实际的代理搜索过滤器。

例如，如果 LDAP 代理搜索属性为 `uid` 且 LDAP 用户搜索过滤器为 `(objectClass=sunIdentityServerDevice)`，则实际的用户搜索过滤器将是 `: (&(uid=*)(objectClass=sunIdentityServerDevice))`

LDAP 代理对象类

定义代理的对象类。创建代理之后，将在代理属性列表中添加此用户对象类列表

LDAP 代理属性

定义与代理相关的属性列表。禁止对不在列表之内的代理属性进行任何读/写尝试。属性区分大小写。在这里定义对象类和属性模式之前，必须先在 Directory Server 中定义对象类和属性模式。

可验证的身份类型

当领域的验证模块模式设置为“数据存储库”时，指定该数据存储库可验证用户和/或代理身份类型。

持久搜索基 DN

定义用于持久搜索的基 DN。某些 LDAPv3 服务器仅在根后缀级别上支持持久搜索。

持久搜索过滤器

定义可返回目录服务器条目的特定更改的过滤器。数据存储库只接收与所定义过滤器相匹配的更改。

重新启动前的持久搜索最长空闲时间

定义重新启动持久搜索前的最长空闲时间。该值必须大于 1。如果小于或等于 1，重新启动搜索时将不会考虑连接的空闲时间。

如果部署 Access Manager 时包括负载平衡器，则某些负载平衡器会在空闲指定的时间后超时。在这种情况下， 您为重新启动持久搜索前的最长空闲时间设置的值应该小于为负载平衡器指定的空闲时间。

出现错误代码后的最大重试次数

定义持久搜索操作遇到在“需要重试的 LDAP 异常错误代码”中指定的错误代码时，可以重试的最大次数。

重试之间的延时

指定每次重试之前的等待时间。仅适用于持久搜索连接。

需要重试的 LDAP 异常错误代码

指定需要重新执行持久搜索操作的错误代码。该属性仅适用于持久搜索，而非所有 LDAP 操作。

高速缓存

如果启用，Access Manager 便可对数据存储库中检索到的数据进行高速缓存。

高速缓存项的最大生存期

指定在高速缓存上删除数据之前，数据的最长存储时间。按秒来定义该值。

高速缓存的最大大小

指定高速缓存的最大大小。值越大，可存储的数据越多，但是这也需要更多的内存。按字节来定义该值。

◆ ◆ ◆ 第 4 章

管理验证

“验证服务”为所有在 Access Manager 部署中安装的默认验证类型提供基于 Web 的用户界面。此界面在用户请求访问时显示登录要求屏幕（根据所调用的验证模块），从而为收集验证证书提供动态和可自定义的方法。此界面使用 Sun Java System™ 应用程序框架（有时称为 *JATO*）创建，该框架是一个用来帮助开发者创建功能性 Web 应用程序的 Java 2 Enterprise Edition (J2EE) 表示框架。

配置验证

本节介绍如何为部署配置验证。第一小节概述默认验证模块类型并提供所有必需的预配置说明。可以为领域、用户、角色等配置同一验证模块类型的多个配置实例。另外，可以添加验证链，这样验证必须满足多个实例的条件后才能成功。本节包括：

验证模块类型

验证模块是一个插件，可以收集用户信息（如用户 ID 和密码），然后根据数据库中的条目检查信息。如果用户提供的信息满足验证条件，则会批准该用户访问请求的资源。如果用户提供的信息不满足验证条件，则会拒绝该用户访问请求的资源。安装 Access Manager 时会随附以下类型的验证模块：

注 – 某些验证模块类型需要预配置然后才能用作验证实例。如有必要，配置步骤会在模块类型说明中列出。

核心

默认情况下，Access Manager 提供了十五个不同的验证模块和一个核心验证模块。核心验证模块提供验证模块的整体配置。在添加和启用活动目录验证模块、匿名验证模块、基于证书的验证模块、HTTP Basic 验证模块、JDBC 验证模块、LDAP 验证模块等验证模块之前，必须先添加和启用核心验证模块。对于默认领域，核心验证模块和 LDAP 验证模块自动启用。

单击 “高级属性” 按钮，可显示能为领域进行定义的核心验证属性。全局属性不适用于领域，因而不会显示出来。

活动目录

活动目录验证模块以类似于 LDAP 模块的方式执行验证，但是使用 Microsoft 的 Active Directory™ 服务器（LDAP 验证模块使用 Directory Server）。尽管 LDAP 验证模块可以被配置为使用 Active Directory 服务器，但此模块允许在同一个领域下存在 LDAP 和活动目录验证。

注 – 对于此版本，活动目录验证模块仅支持用户验证。仅在 LDAP 验证模块中支持密码策略。

匿名

默认情况下，如果已启用此模块，则用户可以作为**匿名**用户登录 Access Manager。通过配置“有效匿名用户列表”属性，还可以为此模块定义匿名用户列表。允许匿名访问意味着无需提供密码即可进行访问。匿名访问可以限于特定的访问类型（例如，读取访问或搜索访问）、特定的子树或目录中的特定条目。

证书

基于证书的验证涉及到使用个人数字证书 (personal digital certificate, PDC) 确定用户的身份和验证用户。可以将 PDC 配置为要求用户提供的证书与 Directory Server 中存储的 PDC 相同，并且根据证书撤销列表进行验证。

将基于证书的验证模块添加到领域之前，需要完成许多操作。首先，需要确保与 Access Manager 一起安装的 Web 容器的安全，并配置此 Web 容器使其适用于基于证书的验证。

注 – 如果通过启用 SSL 的 Sun Java System Web Server 6.1 实例配置 Access Manager 证书验证，并且希望定义的 WebServer 接受基于证书和非基于证书的验证请求，则必须在 WebServer 的 `obj.conf` 文件中设置以下值：

```
PathCheck fn="get-client-cert" dorequest="1" require="0"
```

这是由于为此行为设置可选属性时，WebServer 控制台中存在限制。

启用基于证书的模块之前，参见《*Sun ONE Web Server 6.1* **管理员指南**》中的第 6 章“使用证书和密钥”，了解 Web Server 的初始配置步骤。可以在以下位置找到此文档：

```
http://docs.sun.com/app/docs/coll/S1_websvr61_en
```

或参见以下位置的《*Sun ONE Application Sever Administrator's Guide to Security*》：

http://docs.sun.com/db/prod/s1appsrv#hic (`http://docs.sun.com/db/prod/s1appsrv#hic`)

注 – 使用基于证书的模块进行验证的用户必须请求用户浏览器的 PDC。具体说明各不相同，视使用的浏览器而定。有关详细信息，参见浏览器的文档。

要添加此模块，必须作为领域管理员登录 Access Manager，将 Access Manager 和 Web 容器配置为使用 SSL 并启用客户机验证。有关详细信息，参见 *Access Manager Post Installation Guide* 中的“Configuring Access Manager in SSL Mode”。

数据存储库

数据存储库验证模块允许使用领域的身份系统信息库在用户登录时对其进行验证。如果要根据同一数据存储库系统信息库进行验证，那么采用数据存储库模块便可免去编写验证插件模块，加载然后配置验证模块的麻烦。此外，也无需编写自定义验证模块（其中，需要对该领域中的相应系统信息库进行平面文件验证）。

配置 Access Manager 验证时，此验证类型会提供不少便利。在 Access Manager 7.1 之前的版本中，如果希望 LDAPv3 数据存储库中的用户可验证到其领域，则必须执行以下操作：

- 配置 LDAPv3 数据存储库
- 配置 LDAP 验证模块实例，以使其引用相同的领域主题

数据存储库验证模块验证在领域的身份系统信息库中定义的用户。无需任何 LDAP 验证配置。例如，假设领域的身份系统信息库包括一个 LDAPv3 数据存储库，而且相同的领域使用数据存储库验证。在这种情况下，定义于身份系统信息库中的任何用户都可验证到该领域。

HTTP Basic

此模块使用基本验证，该验证是 HTTP 协议的内置验证支持。Web Server 发出对用户名和密码的客户机请求，并将这些信息作为已授权的请求的一部分发送回服务器。Access Manager 将检索用户名和密码，然后在内部将用户验证到 LDAP 验证模块。为使 HTTP Basic 正常工作，还必须添加 LDAP 验证模块（只添加 HTTP Basic 模块将无法正常工作）。用户成功进行验证后，无需提供用户名和密码即可重新进行验证。

JDBC

Java 数据库连接 (Java Database Connectivity, JDBC) 验证模块提供一种验证机制，允许 Access Manager 通过任何 SQL 数据库（提供启用 JDBC 技术的驱动程序）验证用户。可以直接通过 JDBC 驱动程序或 JNDI 连接池连接 SQL 数据库。

注 – 此模块已在 MySQL4.0 和 Oracle 8i 上进行过测试。

LDAP

使用 LDAP 验证模块时，用户登录时必须用特定用户 DN 和密码绑定至 LDAP Directory Server。这是所有基于领域的验证的默认验证模块。如果用户提供了 Directory Server 中的用户 ID 和密码，则会为用户设置有效的 Access Manager 会话并允许其进行访问。对于默认领域，核心验证模块和 LDAP 验证模块自动启用。

成员资格

成员资格验证的实现类似于个性化设置站点，如 `my.site.com` 或 `mysun.sun.com`。启用此模块时，用户可以在没有管理员帮助的情况下创建帐户并对其进行个性化设置。利用这个新帐户，用户可以作为已添加的用户来访问。用户还可以访问作为授权数据和用户首选项保存在用户概要文件数据库中的查看器界面。

MSISDN

移动站集成服务数字网络 (Mobile Station Integrated Services Digital Network, MSISDN) 验证模块使用与设备（如移动电话）关联的移动用户 ISDN 来启用验证。它是一种非交互式模块。该模块检索用户 ISDN 并根据 Directory Server 对其进行验证，以查找与编号匹配的用户。

RADIUS

可以将 Access Manager 配置为与已安装的 RADIUS 服务器一起使用。如果企业中正使用原有的 RADIUS 服务器进行验证，这样做很有用。RADIUS 验证模块的启用过程分为两个步骤：

1. 配置 RADIUS 服务器。

   有关详细指示，参见 RADIUS 服务器文档。

2. 注册和启用 RADIUS 验证模块。

使用 Sun Java System Application Server 配置 RADIUS

默认情况下，当 RADUIS 客户机建立到其服务器的套接字连接时，在 Application Server 的 `server.policy` 文件中只允许 SocketPermission 连接权限。为使 RADUIS 验证正常工作，对于以下操作应授予权限：

- 接受
- 连接
- 侦听
- 解析

要授予套接字连接权限，必须在 Application Server 的 `server.policy` 文件中添加一个条目。SocketPermission 由主机规范和指定连接到该主机的方式的一组操作组成。请按以下格式指定主机：

```
host = hostname | IPaddress:portrange:portrange = portnumber
| -portnumberportnumber-portnumber
```

Host 可以表示为 DNS 名称、数字 IP 地址或本地主机（对于本地计算机）。DNS 名称主机规范中可包含一处通配符“*”。如果包含通配符，它必须位于最左侧的位置，如 `*.example.com`。

port（或 port range）是可选的。形式为 `N-` 的端口规范（其中 `N` 为端口号）表示编号为 `N` 及以上的所有端口。形式为 `-N` 的规范表示编号为 `N` 及以下的所有端口。

侦听操作仅在与本地主机一起使用时才有意义。任意其他操作存在时，则暗含**解析**（解析主机/IP 名称服务查找）操作。

例如，当创建 SocketPermission 时，请注意如果将以下权限授予某代码，将允许该代码连接到 `machine1.example.com` 上的 `port 1645`，并接受该端口上的连接：

```
permission java.net.SocketPermission machine1.example.com:1645, "connect,accept";
```

类似地，如果将以下权限授予某代码，将允许该代码接受本地主机上 1024 到 65535 之间的任意端口上的连接，并可连接或侦听这些端口：

```
permission java.net.SocketPermission "machine1.example.com:1645", "connect,accept";
permission java.net.SocketPermission "localhost:1024-", "accept,connect,listen";
```

注 – 授予代码权限以接受或建立到远程主机的连接可能会引起问题，因为恶意代码可以更容易地在各方之间传送和共享机密数据，使可能不具有数据访问权限的人访问到数据。请确保仅通过指定确切的端口号（而不是指定端口号的范围）授予适当的权限。

SafeWord

可以配置 Access Manager 使其处理对 Secure Computing 的 SafeWord™ 或 SafeWord PremierAccess™ 验证服务器的 SafeWord 验证请求。Access Manager 提供 SafeWord 验证的客户机部分。SafeWord 服务器可位于安装 Access Manager 的系统或单独的系统中。

使用 Sun Java System Application Server 配置 SafeWord

默认情况下，当 SafeWord 客户机建立到其服务器的套接字连接时，在 Application Server 的 `server.policy` 文件中只允许 SocketPermission 的 `connect` 权限。为使 SafeWord 验证正常工作，需要对以下操作授予权限：

- 接受
- 连接
- 侦听
- 解析

要授予套接字连接权限，必须在 Application Server 的 `server.policy` 文件中添加一个条目。SocketPermission 由主机规范和指定连接到该主机的方式的一组操作组成。请按以下格式指定主机：

```
host = (hostname | IPaddress)[:portrange] portrange =
portnumber | -portnumberportnumber-[portnumber]
```

Host 可以表示为 DNS 名称、数字 IP 地址或本地主机（对于本地计算机）。DNS 名称主机规范中可包含一处通配符“*”。如果包含通配符，它必须位于最左侧的位置，如 `*.example.com`。

port（或 port range）是可选的。形式为 `N-` 的端口规范（其中 `N` 为端口号）表示编号为 `N` 及以上的所有端口。形式为 `-N` 的规范表示编号为 `N` 及以下的所有端口。

侦听操作仅在与本地主机一起使用时才有意义。任意其他操作存在时，**解析**（解析主机/IP 名称服务查找）操作才能执行。

例如，当创建 SocketPermission 时，请注意如果将以下权限授予某个代码，将允许该代码连接到 `machine1.example.com` 上的 `port 1645`，并接受该端口上的连接：

```
permission java.net.SocketPermission machine1.example.com:5030, "connect,accept";
```

类似地，如果将以下权限授予某些代码，将允许该代码接受本地主机上 1024 到 65535 之间的所有端口上的连接、连接到这些端口或侦听它们：

```
permission java.net.SocketPermission "machine1.example.com:5030", "connect,accept";
permission java.net.SocketPermission "localhost:1024-", "accept,connect,listen";
```

注 – 授予代码权限以接受或建立到远程主机的连接可能会引起问题，因为恶意代码可以更容易地在各方之间传送和共享机密数据，使可能不具有数据访问权限的人访问到数据。请确保通过指定确切的端口号（而不是指定一个端口号的范围）仅授予适当的权限。

SAML

安全声明标记语言 (Security Assertion Markup Language, SAML) 验证模块接收和验证目标服务器上的 SAML 声明。SAML SSO 仅在目标计算机上配置了此模块后才会工作（包括升级后，例如从 Access Manager 2005Q4 升级到 Access Manager 7.1）。

SecurID

可以对 Access Manager 进行配置，以处理 RSA 的 ACE/Server 验证服务器的 SecureID 验证请求。Access Manager 提供 SecurID 验证的客户机部分。ACE/Server 可位于安装 Access Manager 的系统或单独的系统中。要验证本地管理的用户 ID（参见 `admintool` (1M)），必须具备超级用户 (root) 访问权限。

SecurID 验证使用验证**帮助应用程序** `amsecuridd`，这是独立于主 Access Manager 进程以外的进程。在启动时，此帮助应用程序将在端口上侦听配置信息。如果 Access Manager 被安装为以 `nobody` 身份运行，或者以非超级用户的某种 userid 身份运行，则 *AccessManager-base*`/SUNWam/share/bin/amsecuridd` 进程必须仍以超级用户身份运行。有关 `amsecuridd` 帮助应用程序的详细信息，参见 Access Manager Administration Reference 中的“The amSecurID Helper”。

注 – 在此发行版本的 Access Manager 中，SecurID 验证模块不适用于 Linux 或 Solaris x86 平台，不能在这两个平台上注册、配置或启用。它仅适用于 SPARC 系统。

UNIX

可以配置 Access Manager 使其按照安装了 Access Manager 的 Solaris 或 Linux 系统已知的 Unix 用户 ID 和密码来处理验证请求。尽管 Unix 验证只有一个领域属性和几个全局属性，仍有一些面向系统的注意事项。要验证本地管理的用户 ID（参见 `admintool` (1M)），必须具备超级用户 (root) 访问权限。

Unix 验证使用验证**帮助应用程序** `amunixd`，这是独立于主 Access Manager 进程以外的进程。在启动时，此帮助应用程序将在端口上侦听配置信息。每个 Access Manager 只有一个 Unix 帮助应用程序用于其所有领域。

如果将 Access Manager 安装为以 `nobody` 运行，或者以非超级用户的某种 userid 身份运行，则 *AccessManager-base*`/SUNWam/share/bin/amunixd` 进程必须仍以超级用户身份运

行。Unix 验证模块通过打开到 localhost:58946 的套接字调用 amunixd 守护进程以侦听 Unix 验证请求。要在默认端口上运行 amunixd 帮助应用程序进程，请输入以下命令：

```
./amunixd
```

要在非默认端口上运行 amunixd，请输入以下命令：

```
./amunixd [-c portnm] [ipaddress]
```

ipaddress 和 portnumber 位于 AMConfig.properties 中的 UnixHelper.ipadrs（以 IPV4 格式）和 UnixHelper.port 属性中。您可以通过 amserver 命令行实用程序运行 amunixd（amserver 自动运行进程，从 AMConfig.properties 中检索端口号和 IP 地址）。

/etc/nsswitch.conf 文件中的 passwd 条目确定是否查询 /etc/passwd 和 /etc/shadow 文件或 NIS 以进行验证。

Windows Desktop SSO

Windows Desktop SSO 验证模块是一个基于 Kerberos 的验证插件模块，用于 Windows 2000™。它允许已通过 Kerberos 分发中心 (Kerberos Distribution Center, KDC) 验证的用户无需重新提交登录条件即可验证到 Access Manager（单点登录）。

用户通过 SPNEGO（Simple and Protected GSS-API Negotiation Mechanism，简单且受保护的 GSS-API 协商机制）协议向 Access Manager 提供 Kerberos 令牌。要通过此验证模块执行基于 Kerberos 的 Access Manager 单点登录，用户必须在客户机端支持 SPNEGO 协议以验证本身。一般而言，支持此协议的任何用户应该都能使用此模块验证 Access Manager。根据客户机端令牌的可用性，此模块提供 SPENGO 令牌或 Kerberos 令牌（这两种情况下协议是相同的）。在 Windows 2000（或更高版本）上运行的 Microsoft Internet Explorer（5.01 或更高版本）当前支持此协议。此外，Solaris（9 和 10）上的 Mozilla 1.4 支持 SPNEGO，但返回的令牌只有一个 KERBEROS 令牌，因为 Solaris 上不支持 SPNEGO。

注 – 必须使用 JDK 1.4 或更高版本利用 Kerberos V5 验证模块和 Java GSS API 的新功能，以执行此 SPNEGO 模块中基于 Kerberos 的 SSO。

Internet Explorer 的已知限制

如果在进行 WindowsDesktopSSO 验证时使用 Microsoft Internet Explorer 6.x，并且浏览器不能访问与 WindowsDesktopSSO 模块中配置的 (KDC) 领域匹配的用户 Kerberos/SPNEGO 令牌，则浏览器在向 WindowsDesktopSSO 模块验证失败后无法对其他模块实施正确的行为。问题的直接原因是：在 Internet Explorer 对 WindowsDesktopSSO 模块失败后，浏览器若未重新启动，将无法传送回叫（其他模块的）给 Access Manager，即使系统提示该回叫。由于用户证书为空，因此 WindowsDesktopSSO 后的所有模块都将失败。

有关相关信息，参见以下文档：

http://support.microsoft.com/default.aspx?scid=kb;en-us;308074
(http://support.microsoft.com/default.aspx?scid=kb;en-us;308074)

http://www.wedgetail.com/jcsi/sso/doc/guide/troubleshooting.html#ieNTLM
(http://support.microsoft.com/default.aspx?scid=kb;en-us;308074)

注 – 此版本的 Access Manager 发行时， Microsoft 已修复了此限制。有关详细信息， 参见以下文档 :

http://www.microsoft.com/technet/security/bulletin/ms06-042.mspx

---

配置 Windows Desktop SSO

启用 Windows Desktop SSO 验证分为两个步骤 :

1. 在 Windows 2000 域控制器中创建用户。
2. 设置 Internet Explorer。

▼ 在 Windows 2000 域控制器中创建用户

1 在域控制器中 ，为 Access Manager 验证模块创建用户帐户。

a. 从"开始"菜单中 ，转至"程序">"管理工具"。

b. 选择"活动目录用户"和"计算机"。

c. 转至 " 计算机 " > " 新建 " > " 计算机 " ，并添加客户机计算机的名称。如果使用的是 Windows XP ，则会在域控制器帐户配置期间自动执行该步骤。

d. 转至 " 用户 " > " 新建 " > " 用户 " ，并创建具有 Access Manager 主机名的新用户作为用户 ID（ 登录名称 ）。Access Manager 主机名不应该包含域名。

2 在用户帐户与服务提供商名称间建立关联，并将密钥表文件导出至装有 Access Manager 的系统。为此，请运行以下命令 :

```
ktpass -princ host/hostname.domainname@DCDOMAIN -pass password -mapuser userName-out
hostname.host.keytab
ktpass -princ HTTP/hostname.domainname@DCDOMAIN -pass
password -mapuser userName-out hostname.HTTP.keytab
```

注 – ktpass 实用程序不会作为 Windows 2000 服务器的一部分安装。必须从安装 CD 将其安装到 c:\program files\support 工具目录。

---

ktpass 命令接受以下参数 :

hostname。运行 Access Manager 的主机名（不含域名） 。

domainname。Access Manager 的域名。

DCDOMAIN。域控制器的域名。它可能与 Access Manager 域名不同。

password。用户帐户的密码。请确保密码正确，因为 `ktpass` 不校验密码。

userName。用户帐户 ID。它应与主机名相同。

注 – 确保两个密钥表文件都已安全保管。

服务模板的值应与以下示例类似：

服务主体： `HTTP/machine1.EXAMPLE.COM@ISQA.EXAMPLE.COM`

密钥文件名： `/tmp/machine1.HTTP.keytab`

Kerberos 领域： `ISQA.EXAMPLE.COM`

Kerberos 服务器名： `machine2.EXAMPLE.com`

返回带有域名的主体： `false`

验证级别： `22`

注 – 如果使用的是 Windows 2003 或 Windows 2003 Service Pack，则使用以下 `ktpass` 命令语法：

```
ktpass /out filename /mapuser username /princ HTTP/hostname.domainname
     /crypto encryptiontype /rndpass /ptype principaltype /target domainname
```

例如：

```
ktpass /out demo.HTTP.keytab /mapuser http
/princ HTTP/demo.identity.sun.com@IDENTITY.SUN.COM /crypto RC4-HMAC-NT
/rndpass /ptype KRB5_NT_PRINCIPAL /target IDENTITY.SUN.COM
```

有关语法定义，参见 `http://technet2.microsoft.com/WindowsServer/en/library/64042138-9a5a-4981-84e9-d576a8db0d051033.mspx?mfr=true` Web 站点。

---

3. **重新启动服务器。**

▼ 设置 Internet Explorer

以下步骤适用于 Microsoft Internet Explorer™ 6 及更高版本。如果您使用的是较早版本，请确保 Access Manager 位于浏览器的 Internet 区域并启用“本地 Windows 验证”。

1 在“工具”菜单中，转至“Internet 选项”>“高级”/”安全”>“安全”。

2 选择“集成的 Windows 验证”选项。

3 转至“安全”>“本地 Intranet”。

a. 选择“自定义级别”。在“用户验证/登录”面板中，选择“只在 Intranet 区域自动登录”选项。

b. 转到“站点”并选择所有选项。

c. 单击“高级”，将 Access Manager 添加到本地区域（如果尚未添加）。

Windows NT

可以将 Access Manager 配置为与已安装的 Windows NT/Windows 2000 server 一起使用。Access Manager 提供 NT 验证的客户机部分。

1. 配置 NT 服务器。有关详细信息，参见 Windows NT 服务器文档。
2. 在添加和启用 Windows NT 验证模块之前，必须获取并安装 Samba 客户端，以与 Solaris 系统上的 Access Manager 进行通信。

安装 Samba 客户端

为了激活 Windows NT 验证模块，必须下载 Samba Client 2.2.2 并将其安装到以下目录：

```
AccessManager-base/SUNWam/bin
```

Samba Client 是文件和打印服务器，它将 Windows 计算机和 UNIX 计算机融合在一起而无需使用单独的 Windows NT/2000 服务器。有关该软件的详细信息及下载该软件，请访问 http://wwws.sun.com/software/download/products/3e3af224.html。

Red Hat Linux 产品随附有 Samba 客户端，它位于以下目录：

```
/usr/bin
```

要使用 Linux 的 Windows NT 验证模块进行验证，将客户机二进制文件复制到以下 Access Manager 目录：

```
AccessManager-base/sun/identity/bin
```

注 – 如果有多个接口，则需要额外配置。可通过 `smb.conf` 文件中的配置设置多个接口，以便传递到 `mbclient`。

验证模块实例

可以根据默认验证模块为领域创建多个验证模块实例。可以添加同一个验证模块的多个单独配置的实例。

▼ 创建新的验证模块实例

1. **单击要为其添加新验证模块实例的领域的名称。**
2. **选择"验证"选项卡。**

注 – "管理员验证配置"按钮只能为管理员定义验证服务。如果需要将管理员的验证模块与最终用户的验证模块区别开来，则可以使用该属性。在访问 Access Manager 控制台时，将使用该属性中配置的模块。

---

3. **在"模块实例"列表中单击"新建"。**
4. **输入验证模块实例的名称。该名称必须唯一。**
5. **选择领域验证模块的类型。**
6. **单击 " 创建 " 。**
7. **单击新建的模块实例名，并编辑该模块的属性。有关每种模块类型的属性的定义，参见联机帮助中的"验证"部分。**
8. **重复执行这些步骤可以添加多个模块实例。**

验证链接

可以配置一个或多个验证模块，用户必须将验证证书传递到这些验证模块中。这称为**验证链接**。Access Manager 的验证链接是通过使用集成在验证服务中的 JAAS 框架实现的。

▼ 创建新的验证链

1. **单击要为其添加新验证链的领域的名称。**
2. **选择"验证"选项卡。**
3. **在"验证链接"列表中单击"新建"。**

4 请输入验证链名称。

5 单击“ 创建 ”。

6 单击“ 添加 ”以定义您要在链里包含的验证模块实例。可以通过从实例列表中选择模块实例名来完成此步骤。此列表中所显示的模块实例名都是在“ 模块实例 ”属性中创建的。

7 为验证链选择标准。这些标志建立了其定义的验证模块的执行标准。执行具有层次结构。“ 必需 ”为最高层，“ 可选 ”为最低层：

必要	要成功验证必须要通过此模块实例。如果验证成功，将继续验证链接列表中的下一个模块实例。如果验证失败，则立即返回到应用程序（不继续验证链接列表中的下一个模块实例）。
必需	对此模块的验证必须成功。如果链中的任一必需模块验证失败，则整个验证链将最终失败。然而，无论必需模块的验证成功或失败，都将继续链中的下一个模块。
充足	不要求模块实例必须成功。如果验证成功，则立即返回到应用程序（不继续模块实例列表中的下一个验证模块）。如果验证失败，将继续验证链接列表中的下一个验证模块。
可选	不要求模块实例必须成功。无论验证成功或失败，都将继续验证链接列表中的下一个验证模块。

8 输入验证链的选项。这将以“ 关键字=值 ”对的形式为模块启用附加选项。多个选项之间用空格分隔。

9 定义以下属性：

成功登录 URL	指定验证成功后将用户重定向至的 URL。
登录失败 URL	指定验证失败后将用户重定向至的 URL。
验证后期处理类	定义用于在登录成功或失败后自定义后期验证处理的 Java 类的名称。

10 单击“ 保存 ”。

验证类型

“验证服务”提供了几种不同的验证方法。可通过指定登录 URL 参数或通过验证 API 来使用这些不同的验证方法（有关详细信息，请参见开发者指南中《Sun Java System

Access Manager 7.1 Developer's Guide》中的第 2 章 "Using Authentication APIs and SPIs"）。在能够配置验证模块之前，必须先修改核心验证服务属性"领域验证模块"以包含特定的验证模块名称。

验证配置服务用于定义以下任一验证类型的验证模块：

- 第 55 页中的 "基于领域的验证"
- 第 58 页中的 "基于组织的验证"
- 第 60 页中的 "基于角色的验证"
- 第 63 页中的 "基于服务的验证"
- 第 65 页中的 "基于用户的验证"
- 第 67 页中的 "基于验证级别的验证"
- 第 69 页中的 "基于模块的验证"

为其中一种验证类型定义了验证模块后，可以基于成功的或失败的验证进程配置该模块以提供重定向 URL 以及后处理 Java 类规范。

验证类型如何确定访问

对于每种方法，用户验证都可能通过或失败。一旦确定，每种方法都遵守这一过程。步骤 1 到步骤 3 接着成功的验证执行；步骤 4 接着成功和失败的验证执行。

1. Access Manager 确认验证的用户是否在 Directory Server 数据存储库中定义，以及配置文件是否处于活动状态。

   "核心验证"模块中的"用户概要文件"属性可以定义为**必需**、**动态**、**随用户别名动态变换**或**忽略**。在成功的验证之后，Access Manager 确认是否在 Directory Server 数据存储库中定义了验证的用户，并且如果"用户概要文件"值为**必需**，则确认用户概要文件是否处于活动状态。（这是默认情况。）如果"用户概要文件"是**动态配置**，"验证服务"将在 Directory Server 数据存储库中创建用户概要文件。如果"用户概要文件"被设置成**忽略**，将不进行用户验证。

2. 完成验证后期处理 SPI 的执行。

   "核心验证模块"包含一个"验证后期处理类"属性，该属性可以把验证后期处理类的名称作为自己的值。`AMPostAuthProcessInterface` 是后期处理接口。它可以在验证成功、验证失败或注销时执行。

3. 以下属性会被添加或更新到会话令牌中，并且用户会话会被激活。

   realm。这是用户所属领域的 DN。

   Principal。这是用户的 DN。

   Principals。这是用户已经验证的名称的列表。（此属性可以有多个值，各值之间以管道符分隔。）

UserId。这是模块返回的用户 DN，如果是“LDAP”或“成员资格”以外的模块，则为用户名。（所有的“主体”必须映射到同一用户。用户 ID 是它们映射到的用户 DN。）

注 – 该属性可能是一个非 DN 值。

UserToken。这是一个用户名。（所有的“主体”必须映射到同一用户。UserToken 是它们映射到的用户名。）

Host。这是客户机的主机名或 IP 地址。

authLevel。这是用户已经验证的最高级别。

AuthType。这是已对用户进行验证的验证模块的列表，各项之间以管道符分隔（例如 `module1|module2|module3`）。

clientType。这是客户机浏览器的设备类型。

Locale。这是客户机的语言环境。

CharSet。这是为客户机确定的字符集。

Role。仅适用于基于角色的验证，这是用户所属的角色。

Service。仅适用于基于服务的验证，这是用户所属的服务。

4. 验证成功或失败后，会在该 URL 中查找信息，以重定向用户。

   URL 重定向的位置可以是 Access Manager 页面或 URL。重定向取决于优先顺序，Access Manager 会根据验证方法以及验证的成败，由此优先顺序寻找重定向。此顺序在以下验证方法章节的 URL 重定向部分有详细描述。

URL 重定向

在验证配置服务中，您可以指定成功或不成功验证的 URL 重定向。而 URL 本身是在该服务的“登录成功 URL”和“登录失败 URL”属性中进行定义的。为了启用 URL 重定向，必须将“验证配置”服务添加到您的领域中，以便可以为角色、领域或用户进行配置。添加“验证配置”服务时，请确保添加一个验证模块，例如 LDAP - REQUIRED。

基于领域的验证

此验证方法允许用户向领域或子领域进行验证。这是 Access Manager 的默认验证方法。通过把“核心验证”模块注册到领域，并定义“领域验证配置”属性，可以设置领域的验证方法。

基于领域的验证登录 URL

通过在“用户界面登录 URL”中定义 realm 参数或 domain 参数可以指定验证的领域。验证请求的领域由下列项目按优先级确定：

1. domain 参数。
2. realm 参数。
3. “管理”服务中的 DNS **别名**属性的值。

   在调用正确的领域后，可以通过“核心验证服务”中的“领域验证配置”属性获取将验证用户的验证模块。用于指定和启动基于领域的验证的登录 URL 是：

   ```
   http://server_name.domain_name:port/amserver/UI/Login
   http://server_name.domain_name:port/amserver/UI/Login?domain=domain_name
   http://server_name.domain_name:port/amserver/UI/Login?realm=realm_name
   ```

   如果没有定义参数，将由登录 URL 中指定的服务器主机和域确定领域。

注 – 如果用户是特定领域的成员并且验证到该特定领域，然后又尝试验证至不同领域，则只会传送 realm 和 module 这两个参数。例如，如果 User1 是 realmA 的成员并且验证到该领域，然后又尝试切换或验证到 realmB，则用户将收到一个警告页面，要求用户使用为 realmB 指定的模块实例启动到 realmB 的新验证，或返回 realmA 中现有的验证会话。如果选择验证到 realmB，则只会传送和使用领域名称和模块名称（如果指定）来确定新的验证过程。

基于领域的验证重定向 URL

在基于组织的验证成功或失败后，Access Manager 会查找信息以重定向用户。下面是应用程序查找这些信息的优先顺序。

成功的基于领域的验证重定向 URL

成功的基于领域的验证重定向 URL 通过按优先顺序检查以下位置来确定：

1. 验证模块设置的 URL。
2. goto 登录 URL 参数设置的 URL。
3. clientType 自定义文件中为用户概要文件 (amUser.xml) 的 iplanet-am-user-success-url 属性设置的 URL。
4. clientType 自定义文件中为用户角色条目的 iplanet-am-auth-login-success-url 属性设置的 URL。
5. clientType 自定义文件中为用户领域条目的 iplanet-am-auth-login-success-url 属性设置的 URL。

6. clientType 自定义文件中为 iplanet-am-auth-login-success-url 属性设置的作为全局默认值的 URL。
7. 用户概要文件 (amUser.xml) 的 iplanet-am-user-success-url 属性中设置的 URL。
8. 用户角色条目的 iplanet-am-auth-login-success-url 属性中设置的 URL。
9. 用户领域条目的 iplanet-am-auth-login-success-url 属性中设置的 URL。
10. iplanet-am-auth-login-success-url 属性中设置的作为全局默认值的 URL。

失败的基于领域的验证重定向 URL

失败的基于领域的验证重定向 URL 通过按下列顺序检查以下位置来确定：

1. 验证模块设置的 URL。
2. gotoOnFail 登录 URL 参数设置的 URL。
3. clientType 自定义文件中为用户条目 (amUser.xml) 的 iplanet-am-user-failure-url 属性设置的 URL。
4. clientType 自定义文件中为用户角色条目的 iplanet-am-auth-login-failure-url 属性设置的 URL。
5. clientType 自定义文件中为用户领域条目的 iplanet-am-auth-login-failure-ur 属性设置的 URL。
6. clientType 自定义文件中为 iplanet-am-auth-login-failure-url 属性设置的作为全局默认值的 URL。
7. 为用户条目 (amUser.xml) 的 iplanet-am-user-failure-url 属性设置的 URL。
8. 为用户角色条目的 iplanet-am-auth-login-failure-url 属性设置的 URL。
9. 为用户领域条目的 iplanet-am-auth-login-failure-url 属性设置的 URL。
10. 为 iplanet-am-auth-login-failure-url 属性设置的作为全局默认值的 URL。

配置基于领域的验证

要为领域设置验证模块，先为领域添加“核心验证”服务。

▼ 配置领域的验证属性

1 找到要为其添加“验证链”的领域。

2 单击“ 验证 ”选项卡。

3 选择“ 默认验证链 ”。

4 **从下拉菜单中选择“管理员验证链”。如果需要将管理员的验证模块与最终用户的验证模块区别开来，则可以使用该属性。默认验证模块为 LDAP。**

5 **定义验证链之后，单击“保存”。**

基于组织的验证

此验证类型只适用于在“传统”模式下安装的 Access Manager 部署。

此验证方法允许用户向组织或子组织进行验证。这是 Access Manager 的默认验证方法。通过把“核心验证”模块注册到组织，并定义“组织验证配置”属性，可以设置组织的验证方法。

基于组织的验证登录 URL

通过在“用户界面登录 URL”中定义 `org` 参数或 `domain` 参数可以指定验证的组织。验证请求的组织由下列项目按优先级确定：

1. `domain` 参数。
2. `org` 参数。
3. “管理服务”中的 `DNS` **别名**（组织别名）属性值。

   在调用正确的组织后，可以通过“核心验证服务”中的“组织验证配置”属性获取将验证用户的验证模块。用于指定和启动基于组织的验证的登录 URL 是：

   ```
   http://server_name.domain_name:port/amserver/UI/Login
   http://server_name.domain_name:port/amserver/UI/Login?domain=domain_name
   http://server_name.domain_name:port/amserver/UI/Login?org=org_name
   ```

   如果没有定义参数，将由服务器主机和登录 URL 指定的域确定组织。

注 – 如果用户是特定组织的成员并且验证到该特定组织，然后又尝试验证至不同组织，则只会传送 `org` 和 `module` 这两个参数。例如，如果 `User1` 是 `orgA` 的成员并且验证到该领域，然后又尝试切换或验证到 `orgB`，则用户将收到一个警告页面，要求用户使用为 `orgB` 指定的模块实例启动到 `orgB` 的新验证，或返回 `orgA` 中现有的验证会话。如果选择验证到 `orgB`，则只会传送和使用组织名称和模块名称（如果指定）来确定新的验证过程。

基于组织的验证重定向 URL

在基于组织的验证成功或失败后，Access Manager 会查找信息以重定向用户。下面是应用程序查找这些信息的优先顺序。

成功的基于组织的验证重定向 URL

成功的基于组织的验证重定向 URL 通过按优先顺序检查以下位置来确定：

1. 验证模块设置的 URL。
2. `goto` 登录 URL 参数设置的 URL。
3. `clientType` 自定义文件中为用户概要文件 (`amUser.xml`) 的 `iplanet-am-user-success-url` 属性设置的 URL。
4. `clientType` 自定义文件中为用户角色条目的 `iplanet-am-auth-login-success-url` 属性设置的 URL。
5. `clientType` 自定义文件中为用户组织条目的 `iplanet-am-auth-login-success-url` 属性设置的 URL。
6. `clientType` 自定义文件中为 `iplanet-am-auth-login-success-url` 属性设置的作为全局默认值的 URL。
7. 用户概要文件 (`amUser.xml`) 的 `iplanet-am-user-success-url` 属性中设置的 URL。
8. 用户角色条目的 `iplanet-am-auth-login-success-url` 属性中设置的 URL。
9. 用户组织条目的 `iplanet-am-auth-login-success-url` 属性中设置的 URL。
10. `iplanet-am-auth-login-success-url` 属性中设置的作为全局默认值的 URL。

失败的基于组织的验证重定向 URL

失败的基于组织的验证重定向 URL 通过按以下顺序检查以下位置来确定：

1. 验证模块设置的 URL。
2. `gotoOnFail` 登录 URL 参数设置的 URL。
3. `clientType` 自定义文件中为用户条目 (`amUser.xml`) 的 `iplanet-am-user-failure-url` 属性设置的 URL。
4. `clientType` 自定义文件中为用户角色条目的 `iplanet-am-auth-login-failure-url` 属性设置的 URL。
5. `clientType` 自定义文件中为用户组织条目的 `iplanet-am-auth-login-failure-url` 属性设置的 URL。
6. `clientType` 自定义文件中为 `iplanet-am-auth-login-failure-url` 属性设置的作为全局默认值的 URL。
7. 为用户条目 (`amUser.xml`) 的 `iplanet-am-user-failure-url` 属性设置的 URL。
8. 为用户角色条目的 `iplanet-am-auth-login-failure-url` 属性设置的 URL。
9. 为用户组织条目的 `iplanet-am-auth-login-failure-url` 属性设置的 URL。
10. 为 `iplanet-am-auth-login-failure-url` 属性设置的作为全局默认值的 URL。

配置基于组织的验证

要为组织设置验证模块，先为组织添加“核心验证”服务。

▼ 配置组织的验证属性

1. **找到要为其添加“验证链”的组织。**
2. **单击“ 验证 ”选项卡。**
3. **选择“ 默认验证链 ”。**
4. **从下拉菜单中选择“管理员验证链”。如果需要将管理员的验证模块与最终用户的验证模块区别开来，则可以使用该属性。默认验证模块为 LDAP。**
5. **定义验证链之后，单击“保存”。**

基于角色的验证

此验证方法允许用户向领域或子领域内的角色（静态或过滤）进行验证。

注 – “验证配置服务”在作为实例注册到角色以前，必须首先注册到领域。

验证要想成功，用户必须属于该角色，并且必须向为该角色配置的“验证配置服”实例中定义的每个模块进行验证。每个基于角色验证的实例均可指定下列属性：

冲突解决级别。此属性为可能包含相同用户的两个不同角色定义的 “验证配置服务” 实例设置优先级级别。例如，如果 `User1` 同时分配给 `Role1` 和 `Role2`，则可以为 `Role1` 设置较高的冲突解决级别。这样，当用户试图进行验证时，`Role1` 将优先进行成功或失败重定向以及验证后期处理。

验证配置。此属性定义为角色验证过程配置的验证模块。

登录成功 URL。此属性定义在验证成功后用户被重定向到的 URL。

登录失败 URL。此属性定义在验证失败后用户被重定向到的 URL。

验证后期处理类。此属性定义验证后期界面。

基于角色的验证登录 URL

通过定义 role 参数，可以在“用户界面登录 URL”中指定基于角色的验证。在调用正确的角色后，可以通过为角色定义的“验证配置服务”实例获取将要验证用户的验证模块。

用于指定和启动基于角色的验证的登录 URL 是：

```
http://server_name.domain_name:port/amserver/UI/Login?role=role_name
http://server_name.domain_name:port/amserver/UI/Login?realm=realm_name&role=role_name
```

如果没有配置 realm 参数，将通过在登录 URL 中指定的服务器主机和域来确定角色所属的领域。

基于角色的验证重定向 URL

在基于角色的验证成功或失败后，Access Manager 会查找信息以重定向用户。下面是应用程序查找这些信息的优先顺序。

成功的基于角色的验证重定向 URL

成功的基于角色的验证重定向 URL 通过按以下顺序检查以下位置来确定：

1. 验证模块设置的 URL。
2. `goto` 登录 URL 参数设置的 URL。
3. `clientType` 自定义文件中为用户概要文件 (`amUser.xml`) 的 `iplanet-am-user-success-url` 属性设置的 URL。
4. `clientType` 自定义文件中为用户已验证的角色的 `iplanet-am-auth-login-success-url` 属性设置的 URL。
5. `clientType` 自定义文件中为已验证用户的另一个角色条目的 `iplanet-am-auth-login-success-url` 属性设置的 URL。（如果以前的重定向 URL 失败，此选项是一个替代方法。）
6. `clientType` 自定义文件中为用户领域条目的 `iplanet-am-auth-login-success-url` 属性设置的 URL。
7. `clientType` 自定义文件中为 `iplanet-am-auth-login-success-url` 属性设置的作为全局默认值的 URL。
8. 用户概要文件 (`amUser.xml`) 的 `iplanet-am-user-success-url` 属性中设置的 URL。
9. 用户已验证角色的 `iplanet-am-auth-login-success-url` 属性中设置的 URL。
10. 已验证用户的另一个角色条目的 `iplanet-am-auth-login-success-url` 属性中设置的 URL。（如果以前的重定向 URL 失败，此选项是一个替代方法。）
11. 用户领域条目的 `iplanet-am-auth-login-success-url` 属性中设置的 URL。
12. `iplanet-am-auth-login-success-url` 属性中设置的作为全局默认值的 URL。

失败的基于角色的验证重定向 URL

失败的基于角色的验证重定向 URL 通过按以下顺序检查以下位置来确定：

1. 验证模块设置的 URL。
2. `goto` 登录 URL 参数设置的 URL。
3. `clientType` 自定义文件中为用户概要文件 (`amUser.xml`) 的 `iplanet-am-user-failure-url` 属性设置的 URL。
4. `clientType` 自定义文件中为用户已验证的角色的 `iplanet-am-auth-login-failure-url` 属性设置的 URL。
5. `clientType` 自定义文件中为已验证用户的另一个角色条目的 `iplanet-am-auth-login-failure-url` 属性设置的 URL。（如果以前的重定向 URL 失败，此选项是一个替代方法。）
6. `clientType` 自定义文件中为用户领域条目的 `iplanet-am-auth-login-failure-url` 属性设置的 URL。
7. `clientType` 自定义文件中为 `iplanet-am-auth-login-failure-url` 属性设置的作为全局默认值的 URL。
8. 用户概要文件 (`amUser.xml`) 的 `iplanet-am-user-failure-url` 属性中设置的 URL。
9. 用户已验证角色的 `iplanet-am-auth-login-failure-url` 属性中设置的 URL。
10. 已验证用户的另一个角色的 `iplanet-am-auth-login-failure-url` 属性中设置的 URL。（如果以前的重定向 URL 失败，此选项是一个替代方法。）
11. 用户领域条目的 `iplanet-am-auth-login-failure-url` 属性中设置的 URL。
12. `iplanet-am-auth-login-failure-url` 属性中设置的作为全局默认值的 URL。

▼ 配置基于角色的验证

1 找到要添加验证配置服务的领域（或组织）。

2 单击“主题”选项卡。

3 “过滤的角色”或“角色”。

4 选择要为其设置验证配置的角色。

5 选择要启用的“默认验证链”。

6 单击“保存”。

注 – 如果要创建新角色，验证配置服务将不会自动指定给该角色。请确保在创建新角色之前先选择“角色配置文件”页面顶部的“验证配置”服务选项。

如果启用了基于角色的验证，可以将 LDAP 验证模块保留为默认设置，因为不需要配置成员资格。

基于服务的验证

此验证方法允许用户向在领域或子领域中注册的特定服务或应用程序进行验证。服务在“验证配置服务”内配置成“服务实例”，并且与“实例名称”关联。验证要想成功，用户必须向为服务配置的“验证配置”服务实例中定义的每个模块进行验证。每个基于服务验证的实例均可指定下列属性：

验证配置。此属性定义为服务验证进程配置的验证模块。

登录成功 URL。此属性定义在验证成功后用户被重定向到的 URL。

登录失败 URL。此属性定义在验证失败后用户被重定向到的 URL。

验证后期处理类。此属性定义验证后期界面。

基于服务的验证登录 URL

通过定义 service 参数，可以在“用户界面登录 URL”中指定基于服务的验证。在调用服务后，可以通过为服务定义的“验证配置”服务实例获取将要验证用户的验证模块。

用来指定和启动基于服务的验证的登录 URL 是：

```
http://server_name.domain_name:port/amserver/UI/
Login?service=auth-chain-name
```

和

```
http://server_name.domain_name:port/amserver/UI/Login?realm=realm_name&service=auth-chain-name
```

如果没有配置 `realm` 参数，将通过在登录 URL 中指定的服务器主机和域来确定用户所属的领域。

基于服务的验证重定向 URL

在基于服务的验证成功或失败后，Access Manager 会查找信息以重定向用户。下面是应用程序查找这些信息的优先顺序。

成功的基于服务的验证重定向 URL

成功的基于服务的验证重定向 URL 通过按以下顺序检查以下位置来确定：

1. 验证模块设置的 URL。
2. goto 登录 URL 参数设置的 URL。
3. clientType 自定义文件中为用户概要文件 (amUser.xml) 的 iplanet-am-user-success-url 属性设置的 URL。
4. clientType 自定义文件中为用户已验证服务的 iplanet-am-auth-login-success-url 属性设置的 URL。
5. clientType 自定义文件中为用户角色条目的 iplanet-am-auth-login-success-url 属性设置的 URL。
6. clientType 自定义文件中为用户领域条目的 iplanet-am-auth-login-success-url 属性设置的 URL。
7. clientType 自定义文件中为 iplanet-am-auth-login-success-url 属性设置的作为全局默认值的 URL。
8. 用户概要文件 (amUser.xml) 的 iplanet-am-user-success-url 属性中设置的 URL。
9. 用户已验证服务的 iplanet-am-auth-login-success-url 属性中设置的 URL。
10. 用户角色条目的 iplanet-am-auth-login-success-url 属性中设置的 URL。
11. 用户领域条目的 iplanet-am-auth-login-success-url 属性中设置的 URL。
12. iplanet-am-auth-login-success-url 属性中设置的作为全局默认值的 URL。

失败的基于服务的验证的重定向 URL

失败的基于服务的验证重定向 URL 通过按以下顺序检查以下位置来确定：

1. 验证模块设置的 URL。
2. goto 登录 URL 参数设置的 URL。
3. clientType 自定义文件中为用户概要文件 (amUser.xml) 的 iplanet-am-user-failure-url 属性设置的 URL。
4. clientType 自定义文件中为用户已验证服务的 iplanet-am-auth-login-failure-url 属性设置的 URL。
5. clientType 自定义文件中为用户角色条目的 iplanet-am-auth-login-failure-url 属性设置的 URL。
6. clientType 自定义文件中为用户领域条目的 iplanet-am-auth-login-failure-url 属性设置的 URL。
7. clientType 自定义文件中为 iplanet-am-auth-login-failure-url 属性设置的作为全局默认值的 URL。
8. 用户概要文件 (amUser.xml) 的 iplanet-am-user-failure-url 属性中设置的 URL。

9. 用户已验证服务的 iplanet-am-auth-login-failure-url 属性中设置的 URL。
10. 用户角色条目的 iplanet-am-auth-login-failure-url 属性中设置的 URL。
11. 用户领域条目的 iplanet-am-auth-login-failure-url 属性中设置的 URL。
12. iplanet-am-auth-login-failure-url 属性中设置的作为全局默认值的 URL。

▼ 配置基于服务的验证

服务的验证模块是在添加了 “验证配置” 服务之后设置的。为此，请执行以下步骤：

1 **选择要配置基于服务的验证的领域。**

2 **单击 “ 验证 ” 选项卡。**

3 **创建验证模块实例。**

4 **创建验证链。**

5 **单击 “ 保存 ” 。**

6 **要访问领域的基于服务的验证 ，请输入以下地址 ：**

```
http://server_name.domain_name:port/amserver/UI/Login?
realm=realm_name&service=auth-chain-name
```

基于用户的验证

此验证方法允许用户向专门为其配置的验证进程进行验证。这个过程被配置成用户概要文件中的“用户验证配置”属性值。验证要想成功，用户必须向定义的每个模块验证。

基于用户的验证登录 URL

通过定义 user 参数，可以在“用户界面登录 URL”中指定基于用户的验证。在调用正确的用户后，可以通过为用户定义的“用户验证配置”实例获取将要验证用户的验证模块。

用于指定和启动基于角色的验证的登录 URL 是：

```
http://server_name.domain_name:port/amserver/UI/Login?user=user_name
http://server_name.domain_name:port/amserver/UI/Login?org=org_name&user=user_name
```

如果没有配置 realm 参数，将通过登录 URL 中指定的服务器主机和域来确定角色所属的领域。

用户别名列表属性

在收到基于用户的验证请求时，“验证”服务会先验证用户是否为有效的用户，然后为其检索“验证配置”数据。如果有多个与用户登录 URL 参数值关联的有效用户概要文件，则所有配置文件都必须映射到指定的用户。可以在用户概要文件的“用户别名属性” (`iplanet-am-user-alias-list`) 中指定属于该用户的其他概要文件。如果映射失败，将拒绝该用户进行有效的会话。例外情况是，如果用户之一是顶级管理员，则不进行用户映射验证，并且用户被授予顶级管理员权限。

基于用户的验证重定向 URL

在基于模块的验证成功或失败后，Access Manager 会查找信息以重定向用户。下面是应用程序查找这些信息的优先顺序。

成功的基于用户的验证重定向 URL

成功的基于用户的验证重定向 URL 通过按优先顺序检查以下位置来确定：

1. 验证模块设置的 URL。
2. `goto` 登录 URL 参数设置的 URL。
3. `clientType` 自定义文件中为用户概要文件 (`amUser.xml`) 的 `iplanet-am-user-success-url` 属性设置的 URL。
4. `clientType` 自定义文件中为用户角色条目的 `iplanet-am-auth-login-success-url` 属性设置的 URL。
5. `clientType` 自定义文件中为用户领域条目的 `iplanet-am-auth-login-success-url` 属性设置的 URL。
6. `clientType` 自定义文件中为 `iplanet-am-auth-login-success-url` 属性设置的作为全局默认值的 URL。
7. 用户概要文件 (`amUser.xml`) 的 `iplanet-am-user-success-url` 属性中设置的 URL。
8. 用户角色条目的 `iplanet-am-auth-login-success-url` 属性中设置的 URL。
9. 用户领域条目的 `iplanet-am-auth-login-success-url` 属性中设置的 URL。
10. `iplanet-am-auth-login-success-url` 属性中设置的作为全局默认值的 URL。

失败的基于用户的重定向 URL

失败的基于用户的验证重定向 URL 通过按以下顺序检查以下位置来确定：

1. 验证模块设置的 URL。
2. `gotoOnFail` 登录 URL 参数设置的 URL。
3. `clientType` 自定义文件中为用户条目 (`amUser.xml`) 的 `iplanet-am-user-failure-url` 属性设置的 URL。

4. clientType 自定义文件中为用户角色条目的 iplanet-am-auth-login-failure-url 属性设置的 URL。
5. clientType 自定义文件中为用户领域条目的 iplanet-am-auth-login-failure-url 属性设置的 URL。
6. clientType 自定义文件中为 iplanet-am-auth-login-failure-url 属性设置的作为全局默认值的 URL。
7. 为用户条目 (amUser.xml) 的 iplanet-am-user-failure-url 属性设置的 URL。
8. 为用户角色条目的 iplanet-am-auth-login-failure-url 属性设置的 URL。
9. 为用户领域条目的 iplanet-am-auth-login-failure-url 属性设置的 URL。
10. 为 iplanet-am-auth-login-failure-url 属性设置的作为全局默认值的 URL。

▼ 配置基于用户的验证

1 找到要在其中为用户配置验证的领域。

2 单击"主题"选项卡，然后单击"用户"。

3 单击要修改的用户的名称

将显示"用户概要文件"。

注 – 如果要创建新用户，验证配置服务将不会自动指定给该用户。请确保在创建用户之前先选择服务配置文件中的"验证配置"服务选项。如果未选择此选项，用户将不会继承为角色定义的验证配置。

---

4 在"用户验证配置"属性中，选择您想要使用的验证链。

5 单击"保存"。

基于验证级别的验证

每个验证模块均可以与其**验证级别**的整数值相关联。更改模块的"验证级别"属性相应的值，可以指定验证级别。用户一个或多个验证模块的验证后，验证级别越高，则用户的信任级别就越高。

用户成功地通过模块的验证之后，系统将在用户的 SSO 令牌中设置验证级别。如果用户需要通过多个验证模块的验证并且成功地通过了这些验证，系统将在用户的 SSO 令牌中设置其中最高的验证级别值。

如果用户试图访问某个服务，该服务可以通过查看用户的 SSO 令牌中的验证级别来确定是否允许该用户进行访问。随后服务将用户重定向，使用户以设定的验证级别通过验证模块。

用户还可以访问具有特定验证级别的验证模块。例如，用户使用以下语法进行登录：

```
http://hostname:port/deploy_URI/UI/Login?authlevel=
auth_level_value
```

所有验证级别高于或等于 *auth_level_value* 的模块将显示为验证菜单以供用户选择。如果只找到了一个匹配的模块，则会直接显示该验证模块的登录页。

此验证方法可让管理员指定验证身份的模块的安全级别。每个验证模块都有单独的“验证级别”属性，此属性的值可以定义为任何有效的整数。利用基于“验证级别”的验证，“验证服务”会显示一个模块登录页面，其中有一个菜单，包含验证级别等于或大于登录 URL 参数所指定的值的验证模块。用户可以从提供的列表中选择模块。在用户选择模块之后，剩余的进程取决于基于模块的验证。

基于验证级别的验证登录 URL

通过定义 authlevel 参数，可以在“用户界面登录 URL”中指定基于验证级别的验证。在调用含有相关模块列表的登录屏幕之后，用户必须选择一个用于验证的模块。用于指定和启动基于验证级别的验证的登录 URL 是：

```
http://server_name.domain_name:port/amserver/UI/Login?authlevel=authentication_level
```

和

```
http://server_name.domain_name:port/amserver/UI/
Login?realm=realm_name&authlevel=authentication_level
```

如果没有配置 `realm` 参数，将通过登录 URL 中指定的服务器主机和域来确定用户所属的领域。

基于验证级别的验证重定向 URL

在基于验证级别的验证成功或失败后，Access Manager 会查找信息以重定向用户。下面是应用程序查找这些信息的优先顺序。

成功的基于验证级别的验证重定向 URL

成功的基于验证级别的验证重定向 URL 通过按优先顺序检查以下位置来确定：

1. 验证模块设置的 URL。
2. `goto` 登录 URL 参数设置的 URL。

3. clientType 自定义文件中为用户概要文件 (amUser.xml) 的 iplanet-am-user-success-url 属性设置的 URL。
4. clientType 自定义文件中为用户角色条目的 iplanet-am-auth-login-success-url 属性设置的 URL。
5. clientType 自定义文件中为用户领域条目的 iplanet-am-auth-login-success-url 属性设置的 URL。
6. clientType 自定义文件中为 iplanet-am-auth-login-success-url 属性设置的作为全局默认值的 URL。
7. 用户概要文件 (amUser.xml) 的 iplanet-am-user-success-url 属性中设置的 URL。
8. 用户角色条目的 iplanet-am-auth-login-success-url 属性中设置的 URL。
9. 用户领域条目的 iplanet-am-auth-login-success-url 属性中设置的 URL。
10. iplanet-am-auth-login-success-url 属性中设置的作为全局默认值的 URL。

失败的基于验证级别的验证重定向 URL

失败的基于验证级别的验证重定向 URL 通过按以下顺序检查以下位置来确定：

1. 验证模块设置的 URL。
2. gotoOnFail 登录 URL 参数设置的 URL。
3. clientType 自定义文件中为用户条目 (amUser.xml) 的 iplanet-am-user-failure-url 属性设置的 URL。
4. clientType 自定义文件中为用户角色条目的 iplanet-am-auth-login-failure-url 属性设置的 URL。
5. clientType 自定义文件中为用户领域条目的 iplanet-am-auth-login-failure-url 属性设置的 URL。
6. clientType 自定义文件中为 iplanet-am-auth-login-failure-url 属性设置的作为全局默认值的 URL。
7. 为用户条目 (amUser.xml) 的 iplanet-am-user-failure-url 属性设置的 URL。
8. 为用户角色条目的 iplanet-am-auth-login-failure-url 属性设置的 URL。
9. 为用户领域条目的 iplanet-am-auth-login-failure-url 属性设置的 URL。
10. 为 iplanet-am-auth-login-failure-url 属性设置的作为全局默认值的 URL。

基于模块的验证

用户可以使用以下语法访问特定的验证模块：

```
http://hostname:port/deploy_URI/UI/Login?module=
module_name
```

在能够访问验证模块之前，必须先修改核心验证服务属性“领域验证模块”以包含该验证模块名称。如果此属性中不包含该验证模块名称，则当用户尝试进行验证时，将会显示“验证模块被拒绝”页面。

此验证方法允许用户指定用来进行验证的模块。指定的模块必须向用户正在访问的领域或子领域注册。这是在领域的“核心验证服务”的“领域验证模块”属性中进行配置的。在收到基于模块的验证请求时，“验证服务”会验证模块是否按要求正确配置，如果该模块未定义，将拒绝用户访问。

基于模块的验证登录 URL

通过定义 module 参数，可以在“用户界面登录 URL”中指定基于模块的验证。用来指定和启动基于模块的验证的登录 URL 是：

```
http://server_name.domain_name:port/amserver/UI/Login?module=authentication_module_name
http://server_name.domain_name:port/amserver/UI/
Login?org=org_name&module=authentication_module_name
```

如果没有配置 `realm` 参数，将通过登录 URL 中指定的服务器主机和域来确定用户所属的领域。

基于模块的验证重定向 URL

在基于模块的验证成功或失败后，Access Manager 会查找信息以重定向用户。下面是应用程序查找这些信息的优先顺序。

成功的基于模块的验证重定向 URL

成功的基于模块的验证重定向 URL 通过按优先顺序检查以下位置来确定：

1. 验证模块设置的 URL。
2. `goto` 登录 URL 参数设置的 URL。
3. `clientType` 自定义文件中为用户概要文件 (`amUser.xml`) 的 `iplanet-am-user-success-url` 属性设置的 URL。
4. `clientType` 自定义文件中为用户角色条目的 `iplanet-am-auth-login-success-url` 属性设置的 URL。
5. `clientType` 自定义文件中为用户领域条目的 `iplanet-am-auth-login-success-url` 属性设置的 URL。
6. `clientType` 自定义文件中为 `iplanet-am-auth-login-success-url` 属性设置的作为全局默认值的 URL。
7. 用户概要文件 (`amUser.xml`) 的 `iplanet-am-user-success-url` 属性中设置的 URL。
8. 用户角色条目的 `iplanet-am-auth-login-success-url` 属性中设置的 URL。

9. 用户领域条目的 iplanet-am-auth-login-success-url 属性中设置的 URL。
10. iplanet-am-auth-login-success-url 属性中设置的作为全局默认值的 URL。

失败的基于模块的验证重定向 URL

失败的基于模块的验证重定向 URL 通过按以下顺序检查以下位置来确定：

1. 验证模块设置的 URL。
2. gotoOnFail 登录 URL 参数设置的 URL。
3. clientType 自定义文件中为用户条目 (amUser.xml) 的 iplanet-am-user-failure-url 属性设置的 URL。
4. clientType 自定义文件中为用户角色条目的 iplanet-am-auth-login-failure-url 属性设置的 URL。
5. clientType 自定义文件中为用户领域条目的 iplanet-am-auth-login-failure-url 属性设置的 URL。
6. clientType 自定义文件中为 iplanet-am-auth-login-failure-url 属性设置的作为全局默认值的 URL。
7. 为用户角色条目的 iplanet-am-auth-login-failure-url 属性设置的 URL。
8. 为用户领域条目的 iplanet-am-auth-login-failure-url 属性设置的 URL。
9. 为 iplanet-am-auth-login-failure-url 属性设置的作为全局默认值的 URL。

用户界面登录 URL

在 Web 浏览器的“地址栏”中输入登录 URL 可访问“验证服务”用户界面。该 URL 是：

http://*AccessManager-root*/.*domain_name*:*port* /*service_deploy_uri* /UI/Login

---

注 – 在安装过程中，*service_deploy_uri* 被配置为 amserver。此默认服务部署 URI 将在本文档的全文中使用。

用户界面登录 URL 也可以附加登录 URL 参数来定义特定的验证方法或成功/失败的验证重定向 URL。

登录 URL 参数

URL 参数是附加在 URL 末尾的名称/值对。该参数以问号 (?) 开始，格式为 name=value。一个登录 URL 可以组合使用多个参数，如：

```
http://server_name.domain_name:port/amserver/UI/
Login?module=LDAP&locale=ja&goto=http://www.sun.com
```

如果存在多个参数，参数之间用 (&) 号分隔。但组合必须遵守以下原则：

- 每个参数在一个 URL 中只能出现一次。例如，`module=LDAP&module=NT` 是不可计算的。
- `org` 参数和 `domain` 参数都可以确定登录领域。在这种情况下，登录 URL 中只能使用其中一个参数。如果同时使用两个参数且不指定优先级，将只有一个生效。
- 参数 `user`、`role`、`service`、`module` 和 `authlevel` 用于定义基于各自标准的验证模块。因此，登录 URL 中只能使用其中一个参数。如果同时使用多个参数且不指定优先级，将只有一个生效。

以下各节描述各参数，这些参数在附加到“用户界面登录 URL”并键入 Web 浏览器的“地址栏”中时，可获取不同的验证功能。

注 – 为简化验证 URL 和参数以便在整个领域内分发，管理员可配置一个具备简单 URL 的 HTML 网页，该页面可链接到更复杂的登录 URL 以获取所有已配置的验证方法。

---

goto 参数

`goto=`*successful_authentication_URL* 参数覆写在“验证配置”服务的“登录成功 URL”中定义的值。当验证成功时，将链接到指定的 URL。`goto=logout_URL` 参数也可用于在用户注销时链接到指定的 URL。成功的验证 URL 示例如下：

```
http://server_name.domain_name:port/amserver/
UI/Login?goto=http://www.sun.com/homepage.html
```

`goto` 注销 URL 的示例如下：

```
http://server_name.domain_name:port/amserver/
UI/Logout?goto=http://www.sun.com/logout.html。
```

注 – Access Manager 按优先顺序查找成功的验证重定向 URL。因为这些重定向 URL 及其顺序基于验证方法，所以此顺序（和相关信息）在“验证类型”部分中进行详细介绍。

---

gotoOnFail 参数

`gotoOnFail=failed_authentication_URL` 参数覆写在“验证配置”服务的“登录失败 URL”中定义的值。如果用户验证失败，将链接到指定的 URL。例如，`gotoOnFail URL` 可能是 `http://`*server_name.domain_name*`:`*port*`/amserver/UI/Login?gotoOnFail=http://www.sun.com/auth_fail.html`。

注 – Access Manager 按优先顺序查找失败的验证重定向 URL。因为这些重定向 URL 及其顺序基于验证方法，所以此顺序（和相关信息）在“验证类型”部分中进行详细介绍。

realm 参数

`realm=`*realmName* 参数允许用户作为指定领域中的用户进行验证。

注 – 尚未成为指定领域成员的用户如果试图使用 `realm` 参数进行验证，会收到一条错误消息。如果以下所有条件均为 TRUE，则可在 Directory Server 中动态创建用户概要文件：

- “核心验证服务”中的“用户概要文件”属性必须设置为**动态**或**随用户别名动态变换**。
- 用户必须成功通过所需模块的验证。
- 该用户在 Directory Server 中还没有配置文件。

使用此参数，将会显示正确的登录页面（基于领域及其语言环境设置）。如果未设置此参数，默认值是顶层领域。例如，`realm` URL 可以是：

```
http://server_name.domain_name:port/amserver/UI/Login?realm=sun
```

org 参数

`org=`*orgName* 参数可让用户作为指定组织中的用户进行验证。

注 – 尚未成为指定组织成员的用户如果试图使用 `org` 参数进行验证，会收到一条错误消息。如果以下所有条件均为 TRUE，则可在 Directory Server 中动态创建用户概要文件：

- “核心验证服务”中的“用户概要文件”属性必须设置为**动态**或**随用户别名动态变换**。
- 用户必须成功通过所需模块的验证。
- 该用户在 Directory Server 中还没有配置文件。

使用此参数，将会显示正确的登录页面（基于组织及其语言环境设置）。如果未设置此参数，默认值是顶层组织。例如，`org` URL 可以是：

```
http://server_name.domain_name:port/amserver/UI/Login?org=sun
```

user 参数

`user=`*userName* 参数强制使用在用户概要文件的“用户验证配置”属性中配置的模块进行验证。例如，可以将一个用户的概要文件配置为使用“认证”模块进行验证，同时

可以将另一个用户配置为使用 LDAP 模块进行验证。添加此参数会将用户发送到其配置的验证进程，而非为其组织配置的方法。例如：

```
http://server_name.domain_name:port/amserver/UI/Login?user=jsmith
```

role 参数

role=roleName 参数会把用户发送到为指定的角色配置的验证过程。尚未成为指定角色成员的用户如果试图用此参数进行验证，会收到一条错误消息。例如：

```
http://server_name.domain_name:port/amserver/UI/Login?role=manager。
```

locale 参数

Access Manager 可为验证进程以及控制台本身显示本地化屏幕（翻译成英语以外的语言）。locale=*localeName* 参数指定的语言环境具有比其他任何已定义的语言环境更高的优先权。在以下位置按顺序搜索配置之后，客户机会显示登录语言环境：

1. 登录 URL 中的 locale 参数值

   locale=*localeName* 参数的值的优先级高于所有其他定义的语言环境。
2. 用户概要文件中定义的语言环境

   如果没有 URL 参数，则根据用户概要文件中“用户首选语言”属性的设置值显示语言环境。
3. HTTP 标题中定义的语言环境

   此语言环境由 Web 浏览器设置。
4. “核心验证服务”中定义的语言环境

   这是“核心验证”模块中“默认验证语言环境”属性的值。
5. “平台”服务中定义的语言环境

   这是“平台”服务中“平台语言环境”属性的值。

操作系统语言环境

从此等级顺序得到的语言环境存储在用户的会话令牌中，Access Manager 只使用它来加载本地化验证模块。在成功验证之后，将使用用户概要文件中的“用户首选语言”属性定义的语言环境。如果没有设置，将继续使用验证所用的语言环境。例如：

```
http://server_name.domain_name:port/amserver/UI/Login?locale=ja。
```

注 – 有关如何本地化屏幕文本和错误消息的信息可以在 Access Manager 中找到。

---

module 参数

module=*moduleName* 参数允许通过指定验证模块进行验证。可以指定任何模块，但它们必须首先在用户所属领域下注册并作为“核心验证”模块中该领域的验证模块之一被选定。例如：

```
http://server_name.domain_name:port/amserver/UI/Login?module=Unix。
```

注 – 验证模块名称用在 URL 参数中时区分大小写。

service 参数

service=*serviceName* 参数允许用户通过服务的已配置验证方案进行验证。使用“验证配置”服务可以为不同的服务配置不同的验证方案。例如，联机薪金应用程序可能需要使用更安全的“证书验证”模块进行验证，而领域的员工目录应用程序可能只需要“LDAP 验证”模块。可以为这些服务中的每一个配置并命名验证方案。例如：

```
http://server_name.domain_name:port/amserver/UI/Login?service=sv1。
```

注 – “验证配置”服务用来为基于服务的验证定义方案。

arg 参数

arg=*newsession* 参数用于终止用户的当前会话并开始一个新会话。“验证服务”将通过一个请求销毁用户的现有会话令牌并执行新的登录。此选项通常用于“匿名验证”模块。用户首先使用匿名会话进行验证，然后单击注册或登录链接。例如：

```
http://server_name.domain_name:port/amserver/UI/Login?arg=newsession。
```

authlevel 参数

authlevel=*value* 参数告知“验证服务”调用验证级别等于或大于指定验证级别值的模块。每个验证模块都定义了一个固定整数的验证级别。例如：

```
http://server_name.domain_name:port/amserver/UI/Login?authlevel=1.
```

注 – “验证级别”设置在特定于每个模块的配置文件中。

domain 参数

此参数允许用户登录到标识为指定域的领域。指定域必须与领域配置文件的“域名”属性中定义的值相匹配。例如：

```
http://server_name.domain_name:port/amserver/UI/Login?domain=sun.com.
```

注 – 尚未成为指定域/领域成员的用户如果试图使用 `domain` 参数进行验证，会收到一条错误消息。如果以下所有条件均为 TRUE，则可在 Directory Server 中动态创建用户概要文件：

- “核心验证服务”中的“用户概要文件”必须设置为**动态**或**随用户别名动态变换**。
- 用户必须成功通过所需模块的验证。
- 该用户在 Directory Server 中还没有配置文件。

iPSPCookie 参数

`iPSPCookie=yes` 参数允许用户以一个持久 cookie 登录。当浏览器窗口关闭以后，持久 cookie 继续存在。要使用此参数，用户所登录的领域必须在其“核心验证”模块中启用“持久 Cookie”。一旦用户进行验证并关闭浏览器，用户可以使用新的浏览器会话登录并定向至控制台而无需重新验证。这将一直有效，直到经过“核心服务”中指定的“持久 Cookie 最长时间”属性值为止。例如：

```
http://server_name.domain_name:port/amserver/UI/Login?org=example&iPSPCookie=yes
```

IDTokenN 参数

此参数选项允许用户以 URL 或 HTML 表单传送验证证书。用户可使用 `IDTokenN=`*value* 参数通过验证，而无需访问“验证服务用户界面”。此进程称为**零页面登录**。零页面登录仅适用于使用一个登录页面的验证模块。`IDToken0, IDToken1, ..., IDTokenN` 的值映射到验证模块登录页面上的字段。例如，LDAP 验证模块可能将 `IDToken1` 用于 `userID` 信息，将 `IDToken2` 用于密码信息。 在这种情况下，LDAP 模块 IDTokenN URL 是：

```
http://server_name.domain_name:port/amserver/UI/
Login?module=LDAP&IDToken1=userID&IDToken2=password
```

（如果 LDAP 是默认的验证模块，则可以省略 `module=LDAP`。）

对于匿名验证，登录 URL 参数是：

```
http://server_name.domain_name:port/amserver/UI/Login?module=Anonymous&IDToken1=anonymousUserID。
```

注 – 仍支持令牌名称 `Login.Token0`、`Login.Token1`、 `...`、 `Login.TokenN`（来自先前的版本），但在以后的版本中将不再支持。建议使用新的 `IDTokenN` 参数。

帐户锁定

“验证服务”提供这样一项功能：在验证失败次数超过某个特定值后将**锁定**用户。此功能默认情况下是关闭的，但是可以使用 Access Manager 控制台启用它。

注 – 只有抛出“密码无效”异常的模块可以使用“帐户锁定”功能。

“核心验证”服务包含用于启用和自定义此功能的属性，包括但不限于：

- **登录失败锁定模式**，启用帐户锁定。
- **登录失败锁定计数**，定义用户被锁定之前可以尝试验证的次数。此计数仅对单个用户 ID 有效；只有同一个用户 ID 失败次数达到指定的次数后才会被锁定。
- **登录失败锁定间隔**，定义在锁定用户之前必须达到“登录失败锁定计数”值的时间（以分钟为单位）。
- **要发送锁定通知的电子邮件地址**，指定接收用户锁定通知的电子邮件地址。
- **N 次失败后警告用户**，指定在向用户显示警告消息之前可以发生的验证失败次数。这允许管理员在用户得到即将锁定的警告之后设置附加的登录尝试次数。
- **登录失败锁定时间**，定义用户在锁定后再次尝试验证前所必须等待的时间（以分钟为单位）。
- **锁定属性名**，定义用户概要文件中的哪一个 LDAP 属性针对“物理锁定”设置为**不活动**。
- **锁定属性值**，定义在**锁定属性名**中指定的 LDAP 属性将设置为：**不活动**或**活动**。

有关任何帐户锁定的电子邮件通知都会发送给管理员。（还会记录帐户锁定活动。）

注 – 有关在 Microsoft® Windows 2000 操作系统上使用此功能的特殊说明，参见附录 A，“AMConfig.properties 文件”中的“简单邮件传输协议 (SMTP)”。”

Access Manager 支持两种类型的帐户锁定：“物理锁定”和“内存锁定”，具体在以下几节中定义。

物理锁定

这是 Access Manager 的默认锁定行为。通过将用户概要文件中 LDAP 属性的状态更改为不活动可以启动此锁定。**锁定属性名**属性定义用来进行锁定的 LDAP 属性。

注 – 别名用户是通过配置 LDAP 配置文件中的“用户别名列表属性”(`amUser.xml` **中的** `iplanet-am-user-alias-list`) 映射到现有 LDAP 用户概要文件的用户。别名用户可以通过将 `iplanet-am-user-alias-list` 添加到“核心验证服务”中的“别名搜索属性名称”字段来进行验证。也就是说，如果别名用户被锁定，则该别名用户映射至的实际 LDAP 配置文件也将被锁定。这只适合于 LDAP 及“成员资格”以外的验证模块的物理锁定。

内存锁定

将**登录失败锁定时间**属性的值更改为大于 0，可启用内存锁定。用户的帐户会在内存中锁定指定的分钟数。帐户将在经过该时间段之后解除锁定。以下是使用内存锁定功能时的一些特殊注意事项：

- 如果重新启动 Access Manager，所有内存中锁定的帐户都将解除锁定。
- 如果用户的帐户在内存中锁定，而管理员将帐户锁定机制改为物理锁定（通过将锁定时间设置回 0），则用户的帐户将在内存中解除锁定，锁定计数也会重置。
- 内存锁定后，当使用 LDAP 和“成员资格”以外的验证模块时，如果用户尝试用正确的密码登录，将返回**用户在此领域中没有配置文件**错误，而不是“**用户无效**”错误。

注 – 如果在用户的概要文件中设置了“失败 URL”属性，则无论是锁定警告消息，还是表示其帐户已锁定的消息，都不会显示；用户将被重定向至定义的 URL。

验证服务故障转移

如果主服务器因硬件或软件问题失败或者服务器临时关闭，则验证服务故障转移会自动将验证请求重定向到辅助服务器。

必须首先在提供验证服务的 Access Manager 实例上创建验证环境。如果此 Access Manager 实例不可用，则可通过验证故障转移机制在其他的 Access Manager 实例上创建验证环境。验证环境将按以下顺序检查服务器可用性。

1. 验证服务 URL 将被传递给 AuthContext API。例如：

   ```
   AuthContext(orgName, url)
   ```

   如果使用此 API，则它将仅使用由 URL 所引用的服务器。即使在该服务器中提供了验证服务，也不会进行故障转移。
2. 验证环境将检查在 `AMConfig.properties` 文件的 `com.iplanet.am.server*` 属性中定义的服务器。

3. 如果步骤 2 失败，则验证环境将从提供有命名服务的服务器查询平台列表。此平台列表是在安装共享同一个 Directory Server 实例的多个 Access Manager 实例（通常是为故障转移目的）时自动创建的。

   例如，如果该平台列表包含 `Server1`、`Server 2` 和 `Server 3` 的 URL，则验证环境会在 `Server 1`、`Server 2` 和 `Server 3` 之间循环，直到验证在其中一个服务器上成功为止。

   平台列表可能不会始终从同一个服务器获得，因为它取决于命名服务的可用性。而且，可能会首先进行命名服务故障转移。多个命名服务 URL 在 `com.iplanet.am.naming.url` **属性**（`AMConfing.properties` 中）中指定。第一个可用的命名服务 URL 将用于确定服务器，该服务器中包含将会进行验证故障转移的服务器（限于其平台服务器列表范围内）的列表。

全限定域名映射

全限定域名 (Fully Qualified Domain Name, FQDN) 映射可让“验证服务”在用户键入错误的 URL（例如指定部分主机名或 IP 地址来访问受保护的资源）时进行纠正。FQDN 映射通过修改 `AMConfig.properties` 文件中的 `com.sun.identity.server.fqdnMap` 属性来启用。用于指定此属性的格式为：

`com.sun.identity.server.fqdnMap[`*invalid-name* `]=`*valid-name*

值 *invalid-name* 是用户可能键入的无效 FQDN 主机名，*valid-name* 是过滤器将用户重定向到的实际主机名。只要符合规定的要求，可以指定任意数量的映射（如代码示例 1-1 所示）。如果未设置此属性，用户将被发送到 `AMConfig.propertiess` 文件的 `com.iplanet.am.server.host=` *server_name* 属性中配置的默认服务器名称。

示例 4–1 `AMConfig.properties` 中的 FQDN 映射属性

```
com.sun.identity.server.fqdnMap[isserver]=isserver.mydomain.com
com.sun.identity.server.fqdnMap[isserver.mydomain]=isserver.mydomain.com
com.sun.identity.server.fqdnMap[
            IP address]=isserver.mydomain.com
```

FQDN 映射的可能用途

此属性可用于为多个主机名创建映射，例如在服务器上的应用程序可由多个主机名访问时便可使用此属性。此属性也可用于配置 Access Manager 使其对特定的 URL 不进行纠错。例如，如果使用 IP 地址访问应用程序的用户不需要重定向，可以通过指定如下映射条目来实现此功能：

`com.sun.identity.server.fqdnMap[`*IP address*`]=`*IP address*。

注 – 如果定义了多个映射，请确保无效的 FQDN 名称中没有重叠的值。否则可能导致无法访问应用程序。

持久 Cookie

持久 Cookie 在 Web 浏览器关闭之后继续存在，用户可以使用新的浏览器会话登录而无需重新验证。Cookie 的名称由 `AMConfig.properties` 中的 `com.iplanet.am.pcookie.name` 属性定义；默认值是 `DProPCookie`。Cookie 值是 3DES 加密字符串，包含用户 DN、领域名称、验证模块名称、最长会话时间、空闲时间和高速缓存时间。

▼ 启用持久 Cookie

1. **在“ 核心验证 ”模块中打开持久 `Cookie` 模式。**
2. **在核心验证模块中的持久 `Cookie` 最长时间属性配置时间值。**
3. **将值为 `yes` 的 iPSPCookie 参数附加到“用户界面登录 URL”。**

   一旦用户使用此 URL 进行验证，如果浏览器被关闭，用户可以打开新的浏览器窗口并重定向到控制台而无需重新验证。这在到达步骤 2 所定义的时间之前一直有效。

   可以使用“验证 SPI”方法打开“持久 Cookie”模式：

   `AMLoginModule.setPersistentCookieOn()`。

传统模式下的多 LDAP 验证模块配置

作为一种故障转移形式，或当 Access Manager 控制台仅提供一个值字段时要配置属性的多个值，管理员可以在一个领域下定义多个 LDAP 验证模块配置。尽管这些附加配置无法通过控制台查看，但如果未找到对于请求用户的授权的初始搜索，这些配置可与主配置一起发挥作用。例如，一个领域可以定义在两个不同的域中搜索 LDAP 服务器进行验证，也可以在一个域中配置多个用户命名属性。对于后者，控制台中只有一个文本字段，如果使用主要搜索条件找不到用户，LDAP 模块将使用第二个范围搜索。以下是配置其他 LDAP 配置的步骤。

▼ 添加其他 LDAP 配置

1 编写一个 XML 文件，在其中包括完整的属性集和第二个（或第三个）LDAP 验证配置所需的新值。

可以通过查看 etc/opt/SUNWam/config/xml 中的 amAuthLDAP.xml 来引用可用的属性。但此步骤创建的 XML 文件是基于 amadmin.dtd 结构的，这与 amAuthLDAP.xml 不同。可以为此文件定义任何或所有属性。代码示例 1-2 是一个子配置文件的示例，该文件包括 LDAP 验证配置可用的所有属性的值。

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
  Copyright (c) 2002 Sun Microsystems, Inc. All rights reserved.
  Use is subject to license terms.
-->
<!DOCTYPE Requests
    PUBLIC "-//iPlanet//Sun ONE Access Manager 6.0 Admin CLI DTD//EN"
    "jar://com/iplanet/am/admin/cli/amAdmin.dtd"
>
<!--
  Before adding subConfiguration load the schema with
GlobalConfiguration defined and replace corresponding
 serviceName and subConfigID in this sample file OR load
 serviceConfigurationRequests.xml before loading this sample
-->
<Requests>
<realmRequests DN="dc=iplanet,dc=com">
    <AddSubConfiguration subConfigName = "ssc"
        subConfigId = "serverconfig"
        priority = "0" serviceName="iPlanetAMAuthLDAPService">

              <AttributeValuePair>
            <Attribute name="iplanet-am-auth-ldap-server"/>
            <Value>vbrao.red.iplanet.com:389</Value>
        </AttributeValuePair>
        <AttributeValuePair>
            <Attribute name="iplanet-am-auth-ldap-base-dn"/>
            <Value>dc=iplanet,dc=com</Value>
        </AttributeValuePair>
        <AttributeValuePair>
            <Attribute name="planet-am-auth-ldap-bind-dn"/>
            <Value>cn=amldapuser,ou=DSAME Users,dc=iplanet,dc=com</Value>
        </AttributeValuePair>
        <AttributeValuePair>
            <Attribute name="iplanet-am-auth-ldap-bind-passwd"/>
            <Value>
                  plain text password</Value>
        </AttributeValuePair>
```

```
            <AttributeValuePair>
                <Attribute name="iplanet-am-auth-ldap-user-naming-attribute"/>
                <Value>uid</Value>
            </AttributeValuePair>
            <AttributeValuePair>
                <Attribute name="iplanet-am-auth-ldap-user-search-attributes"/>
                <Value>uid</Value>
            </AttributeValuePair>
            <AttributeValuePair>
                <Attribute name="iplanet-am-auth-ldap-search-scope"/>
                <Value>SUBTREE</Value>
            </AttributeValuePair>
            <AttributeValuePair>
                <Attribute name="iplanet-am-auth-ldap-ssl-enabled"/>
                <Value>false</Value>
            </AttributeValuePair>
            <AttributeValuePair>
                <Attribute name="iplanet-am-auth-ldap-return-user-dn"/>
                <Value>true</Value>
            </AttributeValuePair>
            <AttributeValuePair>
                <Attribute name="iplanet-am-auth-ldap-auth-level"/>
                <Value>0</Value>
            </AttributeValuePair>
            <AttributeValuePair>
                <Attribute name="iplanet-am-auth-ldap-server-check"/>
                <Value>15</Value>
            </AttributeValuePair>

        </AddSubConfiguration>

    </realmRequests>
    </Requests>
```

2 复制纯文本密码作为在步骤 1 中创建的 XML 文件中的 iplanet-am-auth-ldap-bind-passwd 的值。

在代码示例中，此属性的值被格式化为粗体。

3 使用 amadmin 命令行工具装入 XML 文件。

```
./amadmin -u amadmin -w administrator_password -v -t name_of_XML_file.
```

请注意，第二个 LDAP 配置不可见，也不能用控制台修改。

提示 – 多个 LDAP 配置有可供使用的样例。参见 `/AccessManager-base/SUNWam/samples/admin/cli/bulk-ops/` 中的 `serviceAddMultipleLDAPConfigurationRequests.xml` 命令行模板。可以在 `/AccesManager-base/SUNWam/samples/admin/cli/` 的 `Readme.html` 中找到说明。

会话升级

验证服务允许根据同一用户向一个领域执行的第二次成功验证来升级有效的会话令牌。如果拥有有效会话令牌的用户尝试向其当前领域保护的资源验证，并且这第二次验证请求成功，则该会话将用基于新验证的新属性更新。如果验证失败，用户的当前会话将返回而不进行升级。如果拥有有效会话的用户尝试向不同领域保护的资源验证，该用户将会收到一条询问他们是否要向新领域验证的消息。此时用户可以保持当前的会话，也可以尝试向新领域进行验证。成功的验证将损坏原来的会话，并创建新会话。

在会话升级期间，如果登录页面超时，就会重定向到原来的成功 URL。超时值取决于：

- 每个模块的页面超时值设置（默认值为 1 分钟）
- `AMConfig.properties` 中的 `com.iplanet.am.invalidMaxSessionTime` 属性（默认值为 10 分钟）
- `iplanet-am-max-session-time`（默认值为 120 分钟）

`com.iplanet.am.invalidMaxSessionTimeout` 和 `iplanet-am-max-session-time` 的值应大于页面超时值，否则在会话升级期间的有效会话信息将丢失，到以前的成功 URL 的 URL 重定向也将失败。

验证插件接口

管理员可以编写适用于其领域的用户名或密码验证逻辑，并将其插入“验证服务”。（只有 LDAP 和“成员资格”验证模块支持此项功能。）在验证用户或更改密码之前，Access Manager 将调用此插件。如果验证成功，验证将会继续；如果验证失败，将会抛出验证失败页面。插件扩展了作为“服务管理 SDK”一部分的 `com.iplanet.am.sdk.AMUserPasswordValidation` 类。有关此 SDK 的信息可以在 Access Manager Javadocs 的 `com.iplanet.am.sdk` 软件包中找到。

▼ 编写和配置验证插件

1. 新的插件类将扩展 `com.iplanet.am.sdk.AMUserPasswordValidation` 类，并实现 `validateUserID()` 和 `validatePassword()` 方法。如果验证失败，将抛出 `AMException`。
2. 编译插件类并将 `.class` 文件放置到所需的位置。更新类路径，使其在运行时可供 Access Manager 访问。
3. 以顶级管理员身份登录 Access Manager 控制台。单击“配置”选项卡，转至“管理服务”的属性。在“用户 ID 和密码验证插件类”字段中键入插件类的名称（包括软件包的名称）。
4. 注销，然后登录。

JAAS 共享状态

JAAS 共享状态可在验证模块之间共享用户 ID 和密码。为以下每个验证模块都定义了选项：

- 领域（或组织）
- 用户
- 服务
- 角色

如果失败，模块会提示所需的证书。在验证失败后，模块会停止运行，或者清除注销共享状态。

启用 JAAS 共享状态

配置 JAAS 共享状态：

- 使用 `iplanet-am-auth-shared-state-enabled` 选项。
- 共享状态选项的用法为：`iplanet-am-auth-shared-state-enabled=true`
- 此选项的默认值为 true。
- 此变量在验证链接配置的“选项”栏中指定。

在失败时，验证模块会根据 JAAS 规范中建议的 `tryFirstPass` 选项行为，提示用户提供所需的证书。

JAAS 共享状态存储选项

配置 JAAS 共享状态存储选项：

- 使用 `iplanet-amauth-store-shared-state-enabled` 选项。
- 存储共享状态选项的用法为：`iplanet-am-auth-store-shared-state-enabled=true`
- 此选项的默认值为 false。
- 此变量在验证链接配置的“选项”栏中指定。

在提交、中止或注销后，将清除共享状态。

◆ ◆ ◆ 第 5 章

管理策略

本章介绍 Sun Java™ System Access Manager 的“策略管理”功能。Access Manager 的“策略管理”功能使顶级管理员或顶级策略管理员能够查看、创建、删除和修改可在所有领域中使用的特定服务的策略。它还为领域管理员、子领域管理员或策略管理员提供了一种在领域级别查看、创建、删除和修改策略的方法。

本章包括以下内容：

概述

策略定义了若干规则，这些规则将指定对某一组织受保护资源的访问权限。企业拥有各种需要进行保护、管理和监视的资源、应用程序和服务。“策略”通过定义用户对给定的资源进行操作的时间和方式，从而控制对这些资源的访问权限和使用方式。策略定义了特定主体的资源。

注 – 主体可以是个人、公司、角色或组等具有某种身份的任何对象。有关详细信息，参见 Java™ 2 Platform Standard Edition Javadoc (`http://java.sun.com/j2se/1.4.2/docs/api/java/security/Principal.html`)。

单个策略能定义二元或非二元决策。二元决策为**是/否**、**真/假**或**允许/拒绝**。非二元决策代表某个属性的值。例如，邮件服务可能包含一个 `mailboxQuota` 属性，其中为每个用户都设置了最大存储值。通常说来，配置策略可以定义某主体在什么条件下可以对什么资源执行什么操作。

策略管理功能

“策略管理”功能提供了用于创建和管理策略的**策略服务**。策略服务允许管理员定义、修改、授予、撤消和删除权限，以保护 Access Manager 部署内部的资源。通常，策略服务包括一个数据存储库、一个允许创建、管理和评估策略的界面库以及一个策略执行程序或**策略代理**。默认情况下，Access Manager 使用 Sun Java Enterprise System Directory Server 进行数据存储，并提供用于策略评估和策略服务自定义的 Java 和 C API（有关详细信息，参见《Sun Java System Access Manager 7.1 Developer’s Guide》）。它也允许管理员将 Access Manager 控制台用于策略管理。Access Manager 提供了一种启用策略的服务 －“URL 策略代理”服务，该服务使用可下载策略代理执行策略。

URL 策略代理服务

安装时，Access Manager 会提供“URL 策略代理”服务来定义策略以保护 HTTP URL。该服务允许管理员通过策略强制程序或**策略代理**来创建和管理策略。

策略代理

“策略代理”是存储企业资源的服务器的“策略强制点”(PEP)。策略代理独立于 Access Manager 而被安装在一个 Web 服务器上，当用户向位于受保护 Web 服务器上的 Web 资源发出请求时，此代理将起到附加授权步骤的作用。此授权是对资源执行的任何用户授权请求的补充。该代理可保护 Web 服务器，而资源反过来又会受到授权插件的保护。

例如，受远程安装的 Access Manager 保护的人力资源 Web 服务器上可能会安装某一代理。此代理可防止无适当策略的人员查看保密的工资信息或其他敏感数据。策略由 Access Manager 管理员定义，存储在 Access Manager 部署中，并由策略代理使用，以允许或拒绝用户对远程 Web 服务器内容的访问权。

最新的“Access Manager 策略代理”可以从“Sun Microsystems 下载中心”下载。

有关安装和管理策略代理的详细信息，参见《Sun Java System Access Manager Policy Agent 2.2 User’s Guide》。

注 – 策略评估不会按特定顺序进行，尽管在对其进行评估时，如果一个操作值评估结果为 *deny*，也不再对后续策略进行评估，除非在“策略配置”服务中启用“拒绝决策时继续评估”属性。

Access Manager 策略代理仅在 Web URL（`http://...` 或 `https//...`）上执行决策。但是，可以使用 Java 和 C Policy Evaluation API 编写代理，以在其他资源上强制执行策略。

此外，还需要将“策略配置服务”中的“资源比较器”属性由其默认配置更改为：

```
serviceType=Name_of_LDAPService
|class=com.sun.identity.policy.plugins.SuffixResourceName|wildcard=*

|delimiter=,|caseSensitive=false
```

或者，提供类似于 `LDAPResourceName` 的实现以实现 `com.sun.identity.policy.interfaces.ResourceName` 并相应地配置“资源比较器”也可达到目的。

策略代理过程

当 Web 浏览器向驻留于策略代理所保护的服务器中的 URL 发出请求后，即开始受保护 Web 资源的过程。服务器中已安装的策略代理会截取请求并检查现有的验证凭证（会话令牌）。

如果代理已截取请求并验证了现有的会话令牌，随后将发生以下过程。

1. 如果会话令牌有效，则允许或拒绝用户的访问。如果会话令牌无效，则用户将被重定向到“验证服务”，如下列各步骤所述。

   假设代理截取了某一请求，而对于该请求不存在任何现有会话令牌，则代理将用户重定向到登录页面，即使资源受不同验证方法保护也是如此。
2. 正确验证用户凭证后，代理会向定义用于连接到 Access Manager 内部服务的 URL 的“命名服务”发布请求。
3. 如果资源与在代理处配置的非执行列表匹配，则允许访问。
4. “命名服务”返回策略服务、会话服务和日志记录服务的定位器。
5. 代理将请求发送到“策略服务”以获取适用于用户的策略决策。
6. 是允许用户访问还是拒绝用户访问，需根据当前访问资源的策略决策而定。如果对策略决策的建议指示出不同的验证级别或验证机制，代理会将请求重定向到“验证服务”，直到所有条件都经过验证为止。

策略类型

有两种类型的策略可以使用 Access Manager 进行配置：

常规策略

在 Access Manager 中，用于定义访问权限的策略被称为**标准**策略。常规策略由**规则**、**主题**、**条件**和**响应提供者**组成。

规则

一条**规则**包含一种服务类型、一项或多项操作以及一个值。规则用于定义策略。

- 服务类型定义受保护的资源类型。
- **操作**是可在资源上执行的操作的名称，例如 Web 服务器操作的示例有 POST 或 GET。针对人力资源服务的一项可行的操作可以是更改家庭电话号码。
- **值**用于定义操作的权限，例如，允许或拒绝。

注 – 可以不使用某些服务的资源来定义操作。

主题

主题定义了策略所影响的用户或用户集合（例如，组或具有特定角色的用户）。主题的常规规则是：只有当用户至少是策略中的其中一个主题的成员时，才能够应用策略。默认主题包括：

Access Manager 身份主题	此主题表明可以将在 “领域主题” 选项卡下创建和管理的身份作为主题的成员添加。
验证的用户	此主题类型表明任何具有有效 SSO 令牌的用户都是此主题的成员。 所有通过验证的用户都是该 “主题” 的成员，即使他们已在其他领域（而不是定义策略所在的组织）中进行了验证。如果资源拥有者要开放一些资源（为其他组织的用户所管理的资源）的访问权时，这将非常有用。如果您要限制某个特定组织的成员对受保护资源的访问权，请使用 “组织” 主题。
Web 服务客户机	此主题类型表明如果 SSO 令牌中包含的主体的 DN 与此主题的任意选定值匹配，则由该 SSO 令牌标识的 Web 服务客户机 (WSC) 是此主题的成员。有效值为本地 JKS 密钥库中

的可信赖证书（对应于可信赖 WSC 证书）的 DN。此主题取决于“Liberty Web 服务框架”，并且只能由“Liberty 服务提供者”用来对 WSC 进行授权。

请确保将此“主题”添加到策略之前，您已经创建了密钥库。您可以从以下位置找到有关于设置密钥库的信息：

AccessManager-base
`/SUNWam/samples/saml/xmlsig/keytool.html`

通过在领域的“策略配置服务”中选择以下附加主题，便可对其进行使用：

Access Manager 角色	此主题类型表明任何使用 Access Manager 角色的成员都是此主题的成员。Access Manager 角色是使用运行于传统模式的 Access Manager 以及基于 6.3 的控制台创建的。这些角色所具有的对象类由 Access Manager 进行授权。Access Manager 角色只能通过所属的“Access Manager 策略服务”进行访问。
LDAP 组	此主题类型表明 LDAP 组的任何成员都是此主题的成员。
LDAP 角色	此主题类型表明任何使用 LDAP 角色的成员都是此主题的成员。“LDAP 角色”是使用 Directory Server 角色功能的任意角色定义。这些角色具有通过角色定义授权的对象类。可以在“策略配置服务”中修改“LDAP 角色搜索”过滤器以缩小范围并提高性能。
LDAP 用户	此主题类型表明任何 LDAP 用户都是此主题的成员。
组织	此主题类型表明领域的所有成员均为该主题的成员

Access Manager 角色与 LDAP 角色

Access Manager 角色是由 Access Manager 创建的。这些角色所具有的对象类由 Access Manager 进行授权。LDAP 角色是使用 Directory Server 角色功能的任意角色定义。这些角色具有通过角色定义授权的对象类。所有 Access Manager 角色均可被用作 Directory Server 角色。但是，Directory Server 角色并不一定都是 Access Manager 角色。可通过配置第 113 页中的 “策略配置服务”从现有目录利用 LDAP 角色。Access Manager 角色只能通过所属的“Access Manager 策略服务”进行访问。可以在“策略配置服务”中修改“LDAP 角色搜索”过滤器以缩小范围并提高性能。

嵌套角色

嵌套角色可作为策略定义主题中的“LDAP 角色”正确评估。

条件

条件允许您定义对策略的限制。例如，为某个薪金应用程序定义策略时，可以为该操作定义一个条件，限定只能在特定的时间内访问该应用程序。另外，您还可以定义另一种条件，限定只有当请求是来自指定的一组 IP 地址或公司内部网时才允许执行该操作。

此外，条件还可以用于配置同一个域的不同 URI 上的不同策略。例如，`http://org.example.com/hr/*jsp` 只能通过 `org.example.net` 在 9 a.m. 到 5 p.m. 之间进行访问。同时使用“IP 条件”和“时间条件”便可实现上述目的。将规则资源指定为 `http://org.example.com/hr/*.jsp`，策略将应用到 `http://org.example.com/hr` 下的所有 JSP 文件，包括子目录中的 JSP 文件。

注 – 引用、规则、资源、主题、条件、操作和值等术语分别对应于 `policy.dtd` 中的 *Referral*、*Rule*、*ResourceName*、*Subject*、*Condition*、*Attribute* 和 *Value* 等元素。

可以添加的默认条件是：

活动会话时间

根据用户会话数据设置条件。您可以修改的字段包括：

最长会话时间　　指定从发起会话起，策略可应用的最长持续时间。

终止会话　　如果选择该字段，当会话时间超过“最长会话时间”字段中定义的最长允许时间时，系统将终止该用户会话。

验证链

如果用户已在指定领域中向验证链成功验证，则应用该策略。如果未指定领域，则向任何领域中的验证链进行验证均满足条件。

验证级别（大于或等于）

如果用户的验证级别大于或等于条件中设置的验证级别，则应用该策略。 该属性表明指定领域内的验证信任级别。

验证级别（小于或等于）

如果用户的验证级别低于或等于在条件中设置的验证级别，则应用该策略。该属性表明指定领域内的验证信任级别。

验证模块实例

如果用户已在指定领域内向验证模块成功进行验证，则应用该策略。如果未指定领域，则向任何领域中的验证模块进行验证均满足条件。

当前会话属性

根据用户的 Access Manager 会话中设置的属性值来决定策略是否适用于相应的请求。策略评估期间，仅当用户会话的每个属性值均符合条件中的定义时，条件才会返回 "true"。对于在条件中定义了多个值的属性，令牌只要具有条件的属性中列出的一个值就足够了。

IP 地址/DNS 名称

根据 IP 地址的范围设置条件。您可以定义的字段包括：

起始/结束 IP 地址	指定 IP 地址的范围。
DNS 名称	指定 DNS 的名称。此字段可以是全限定主机名，也可以是采用以下格式之一的字符串： *domainname* **.domainname*

LDAP 过滤条件

当已定义的 LDAP 过滤器在 LDAP 目录（在“策略配置”服务中指定）中查找用户条目时，应用该策略。该条件仅在定义该策略的领域内适用。

领域验证

如果用户已向指定领域进行验证，则应用此策略。

时间（天、日期、时间和时区）

根据时间限制设置条件。这些字段包括：

起始/结束日期	指定日期的范围。
时间	指定一天中的时间范围。
日	指定表示天数的范围。
时区	指定一个标准的或自定义的时区。自定义的时区只能是可由 Java 识别的时区 ID（例如，PST）。如果未指定值，默认值为 Access Manager JVM 中设置的时区。

响应提供者

响应提供者是提供基于策略的响应属性的插件。响应提供者属性与策略决策一起发送到 PEP。Access Manager 包括一个实现，`IDResponseProvider`。该版本的 Access Manager 不支持自定义的响应提供者。 代理和 PEP 通常会将这些响应属性作为标题传递给应用程序。 应用程序通常会使用这些属性来自定义应用程序页面（如门户页面）。

策略建议

如果根据条件判定策略不适用，该条件可能会产生建议消息，指明策略不适用于请求的原因。这些建议消息在策略决策内传播到“策略强制点”。“策略强制点”可以检索此建议并采取适当的行动，例如将用户重定向回验证机制以进行更高级别的验证。 如果策略适用，系统在针对建议采取适当的操作后可能会提示用户进行更高级别的验证，用户或许可以访问资源。

可从以下类中找到更多信息：

```
com.sun.identity.policy.ConditionDecision.getAdvices()
```

如果条件不满足，则只有 `AuthLevelCondiiton` 和 `AuthSchemeCondition` 提供建议。

`AuthLevelCondition` 建议与下列关键字相关：

```
com.sun.identity.policy.plugin.AuthLevelCondition.AUTH_LEVEL_CONDITION_ADVICE
```

`AuthSchemeCondition` 建议与下列关键字相关：

```
com.sun.identity.policy.plugin.AuthLevelCondition.AUTH_SCHEME_CONDITION_ADVICE
```

自定义的条件也可以提供建议。但是，“Access Manager 策略代理”只对“验证级别建议”和“验证模式建议”做出响应。 可以编写自定义的代理来理解和响应更多建议，也可以扩展现有的 Access Manager 代理来理解和响应更多建议。有关详细信息，参见《Sun Java System Access Manager Policy Agent 2.2 User's Guide》。

引用策略

管理员可能需要将一个领域的策略定义和决策委托给另一个领域。（另外，还可以将资源的策略决策授权给其他策略产品）。**引用**策略控制着策略创建和评估的策略委托。该策略由一个或多个**规则**以及一个或多个**引用**组成。

“策略配置”服务包含称作“组织别名引用”的全局属性，该属性允许用户在子领域中创建策略，而不必在顶层或父领域创建引用策略。用户只能创建用于保护 HTTP 或 HTTPS 资源的策略，这些资源的全限定主机名应与领域的领域/DNS 别名相匹配。默认情况下，该属性设置为“否”。

规则

规则定义策略定义和评估相关的资源。

引用

引用定义策略评估引用的组织。默认情况下，有两种引用类型：对等领域和子领域。它们分别代表同级领域和子级领域。有关详细信息，参见第 105 页中的 “为对等领域和子领域创建策略 ” 。

注 – 相关领域只能为那些已相关的资源（或子资源）定义或评估策略 。但是，该限制并不适用于顶层领域。

策略定义类型文档

一旦创建并配置了策略，它就会以 XML 形式存储于 Directory Server 中。在 Directory Server 中，XML 编码的数据存储在一个位置。尽管策略是使用 `amAdmin.dtd`（或控制台）进行定义和配置，但它实际上是作为基于 `policy.dtd` 的 XML 存储在 Directory Server 中。`policy.dtd` 包含从 `amAdmin.dtd`（无策略创建标记）中提取的策略元素标记。因此，“策略服务”从 Directory Server 加载策略时，它将根据 `policy.dtd` 分析 XML。只有在使用命令行创建策略时，才使用 `amAdmin.dtd`。本节介绍 `policy.dtd` 的结构。`policy.dtd` 存在于下列位置：

```
AccessManager-base/SUNWam/dtd (Solairs)
AccessManager-base/identity/dtd (Linux)
AccessManager-base/identity/dtd (HP-UX)
AccessManager-base\identity\dtd (Windows)
```

注 – 在本章中的余下部分将只给出 Solaris 目录信息。请注意，Linux、HP-UX 和 Windows 的目录结构不同。

Policy 元素

Policy 是根元素，它定义策略的权限或**规则**以及规则适用的对象或**主题**。它还定义策略是否是**引用**（指派）策略以及是否对该策略存在限制（或 **条件**）。它可能包含一个或多个下列子元素： *Rule*、*Conditions*、*Subjects*、*Referrals* 或 *response providers*。所需 XML 属性是 `name`，它指定策略的名称。属性 `referralPolicy` 指明策略是否为引用策略；如果未定义，则它默认为常规策略。可选 XML 属性包括 *name* 和 *description*。

注 – 将策略标记为 *referral* 时，在策略评估期间将忽略主题和条件。相反，将策略标记为 *normal* 时，在策略评估期间将忽略所有“引用项”。

Rule 元素

Rule 元素定义策略的具体内容，可能包含三个子元素： *ServiceName*、*ResourceName* 或 *AttributeValuePair*。它定义已经为其创建策略的服务或应用程序的类型以及资源名称和对其执行的操作。定义规则时可不带任何操作；例如，引用策略就不含任何操作。

注 – 已定义的策略也可以不包括定义的 *ResourceName* 元素。

ServiceName 元素

ServiceName 元素定义策略所适用的服务名称。此元素表示服务类型。它不包含任何其他元素。其值与在服务的 XML 文件（基于 `sms.dtd`）中定义的完全一致。*ServiceName* 元素的 XML 服务属性是服务（取字符串的值）的名称。

ResourceName 元素

ResourceName 元素定义将要对其执行操作的对象。策略已经过专门配置以便保护此对象。它不包含任何其他元素。*ResourceName* 元素的 XML 属性是对象的名称。*ResourceName* 的示例可能是 Web 服务器上的 `http://www.sunone.com:8080/images` 或目录服务器上的 `ldap://sunone.com:389/dc=example,dc=com`。更具体的资源可为 `salary://uid=jsmith,ou=people,dc=example,dc=com`， 正在其上操作的对象为 John Smith 的工资信息。

AttributeValuePair 元素

AttributeValuePair 元素定义操作及其值。它被用作 第 97 页中的 “Subject 元素” 、第 98 页中的 “Referral 元素”和第 98 页中的 “Condition 元素”的子元素。它包含 *Attribute* 和 *Value* 元素但没有 XML 服务属性。

Attribute 元素

Attribute 元素定义操作的名称。操作是针对资源所执行的操作或事件。POST 或 GET 是对 Web 服务器资源执行的操作，READ 或 SEARCH 是对目录服务器资源执行的操作。*Attribute* 元素必须与 *Value* 元素组对。*Attribute* 元素本身不包含其他元素。*Attribute* 元素的 XML 服务属性是操作的名称。

Value 元素

Value 元素定义操作值。 allow/deny 或 yes/no 是操作值的示例。其他操作值可以是布尔值、数字或字符串。该值在服务的 XML 文件（基于 `sms.dtd` ）中定义。*Value* 不包含其他元素，也不包含 XML 服务属性。

注 – 拒绝规则始终优先于允许规则。例如，如果一个策略拒绝访问而另一个策略允许访问，则结果将为拒绝（假定两个策略的所有其他条件都满足）。建议谨慎使用拒绝策略，因为它们会导致潜在的冲突。如果采用显式拒绝规则，则通过不同主题（如角色和/或组成员资格）指定给某一用户的策略可能导致拒绝的访问。通常，策略定义过程应只使用允许规则。当未应用其他任何策略时，才可使用默认拒绝。

Subjects 元素

Subjects 子元素确定策略所适用的主体集合；根据组中的成员资格、角色所有权或个别用户选择该集合。它接受 *Subject* 子元素。可以定义的 XML 属性有：

name。它定义集合的名称。

description。它定义主题的说明。

includeType。当前未使用此项。

Subject 元素

Subject 子元素确定策略所适用的主体集合；该集合可从 Subject 元素所定义的集合中准确找出更具体的对象。 成员资格可基于角色、组成员资格或仅仅基于个别用户的列表。它包含子元素第 96 页中的 “AttributeValuePair 元素” 。所需 XML 属性是 `type`，它确定一个通用的对象集合，具体定义的主题从该集合中提取。其他 XML 属性包括定义集合名称的 `name` 和 `includeType`，后者规定集合是否如定义的那样，用于确定策略是否用于“不”属于该主题成员的用户。

注 – 定义了多个主题时，要使策略得以应用，至少要有一个主题应该应用于用户。将 `includeType` 设置为 false 来定义主题时，用户不应为应用策略的主题成员。

Referrals 元素

Referrals 子元素确定策略引用项的集合。它接受 *Referral* 子元素。定义该因素时可以使用的 XML 属性有定义集合名称的 `name` 和包含说明的 `description`。

Referral 元素

Referral 子元素确定特定的策略引用项。它接受子元素第 96 页中的 “AttributeValuePair 元素”。它必需的 XML 属性是 `type`，该属性确定一个通用的任务集合，具体定义的引用项从该集合中提取。它还可以包含定义集合名称的 `name` 属性。

Conditions 元素

Conditions 子元素标识策略限制（时间范围、验证级别等）集合。它必须包含一个或多个 *Condition* 子元素。定义该因素时可以使用的 XML 属性有定义集合名称的 `name` 和包含说明的 `description`。

注 – Condition 元素是策略中的可选元素。

Condition 元素

Condition 子元素标识特定策略限制（时间范围、验证级别等）。它接受子元素第 96 页中的 “AttributeValuePair 元素”。它必需的 XML 属性是 `type`，该属性确定一个通用的限制集合，具体定义的条件从该集合中提取。它还可以包含定义集合名称的 `name` 属性。

添加已启用策略服务

只有当服务模式的 `<Policy>` 元素配置为 `sms.dtd` 时才可定义给定服务的资源策略。

默认情况下，Access Manager 会提供“URL 策略代理”服务 (`iPlanetAMWebAgentService`)。此服务在位于以下目录的 XML 文件中定义：

```
/etc/opt/SUNWam/config/xml/
```

但是，您可以向 Access Manager 添加附加的策略服务。一旦创建了策略服务，就可以通过 `amadmin` 命令行实用程序把它添加到 Access Manager。

▼ 添加新的已启用策略服务

1 **在基于 sms.dtd 的 XML 文件里开发新的策略服务。Access Manager 提供两个策略服务 XML 文件，用户可能希望将其用作新策略服务文件的基础：**

amWebAgent.xml - 这是默认“URL 策略代理”服务的 XML 文件。它位于 /etc/opt/SUNWam/config/xml/。

SampleWebService.xml - 这是位于 AccessManager-base/samples/policy 的范例策略服务文件。

2 **将该 XML 文件保存到您将从中加载新策略服务的目录。例如：**

```
/config/xml/newPolicyService.xml
```

3 **用 amadmin 命令行实用程序加载新策略服务。例如：**

```
AccessManager-base/SUNWam/bin/amadmin
    --runasdn “uid=amAdmin,ou=People,default_org,
root_suffix
    --password password
    --schema /config/xml/newPolicyService.xml
```

4 **加载新策略服务后，可通过 Access Manager 控制台或使用 amadmin 加载新策略来制定策略定义的规则。**

创建策略

您可以通过“策略 API”和 Access Manager 控制台创建、修改和删除策略，并通过 amadmin 命令行工具创建和删除策略。 您也可以使用 amadmin 实用程序获取和列出 XML 格式的策略。 本节重点介绍如何通过 amadmin 命令行实用程序和 Access Manager 控制台创建策略。有关“策略 API”的详细信息，参见《Sun Java System Access Manager 7.1 Developer’s Guide》。

策略通常使用 XML 文件创建，再通过命令行实用程序 amadmin 添加到 Access Manager，然后使用 Access Manager 控制台进行管理（尽管策略可通过控制台创建）。 这是因为不能直接使用 amadmin 修改策略。要修改策略，必须先从 Access Manager 中删除该策略，然后使用 amadmin 添加已修改的策略。

通常情况下，策略是在领域（或子领域）级别创建以在整个领域树中使用的。

▼ 使用 amadmin 创建策略

1 **创建基于 `amadmin.dtd` 的策略 XML 文件。该文件位于以下目录中：**

AccessManager-base `/SUNWam/dtd`。

以下是策略 XML 文件的一个示例。该示例包含所有的默认主题和条件值。有关这些值的定义，参见第 90 页中的 “策略类型” 。

```
<Policy name="bigpolicy" referralPolicy="false" active="true" >
<Rule name="rule1">
<ServiceName name="iPlanetAMWebAgentService" />
<ResourceName name="http://thehost.thedomain.com:80/*.html" />
<AttributeValuePair>
<Attribute name="POST" />
<Value>allow</Value>
</AttributeValuePair>
<AttributeValuePair>
<Attribute name="GET" />
<Value>allow</Value>
</AttributeValuePair>
</Rule>
<Subjects name="subjects" description="desccription">
<Subject name="webservicescleint" type="WebServicesClients" includeType="inclusive">
<AttributeValuePair><Attribute name="Values"/><Value>CN=sun-unix,
OU=SUN Java System Access Manager, O=Sun, C=US</Value>
</AttributeValuePair>
</Subject>
<Subject name="amrole" type="IdentityServerRoles" includeType="inclusive">
<AttributeValuePair><Attribute name="Values"/><Value>
cn=organization admin role,o=realm1,dc=red,dc=iplanet,dc=com</Value>
</AttributeValuePair>
</Subject>
<Subject name="au" type="AuthenticatedUsers" includeType="inclusive">
</Subject>
<Subject name="ldaporganization" type="Organization" includeType="inclusive">
<AttributeValuePair><Attribute name="Values"/>
<Value>dc=red,dc=iplanet,dc=com</Value>
</AttributeValuePair>
</Subject>
<Subject name="ldapuser" type="LDAPUsers" includeType="inclusive">
<AttributeValuePair><Attribute name="Values"/>
<Value>uid=amAdmin,ou=People,dc=red,dc=iplanet,dc=com</Value>
</AttributeValuePair>
</Subject>
```

```
<Subject name="ldaprole" type="LDAPRoles" includeType="inclusive">
<AttributeValuePair><Attribute name="Values"/>
<Value>cn=Organization Admin Role,o=realm1,dc=red,dc=iplanet,dc=com</Value>
</AttributeValuePair>
</Subject>
<Subject name="ldapgroup" type="LDAPGroups" includeType="inclusive">
<AttributeValuePair><Attribute name="Values"/>
<Value>cn=g1,ou=Groups,dc=red,dc=iplanet,dc=com</Value>
</AttributeValuePair>
</Subject>
<Subject name="amidentitysubject" type="AMIdentitySubject" includeType="inclusive">
<AttributeValuePair><Attribute name="Values"/>
<Value>id=amAdmin,ou=user,dc=red,dc=iplanet,dc=com</Value>
</AttributeValuePair>
</Subject>
</Subjects>
<Conditions name="conditions" description="description">
<Condition name="ldapfilter" type="LDAPFilterCondition">
<AttributeValuePair><Attribute name="ldapFilter"/>
<Value>dept=finance</Value>
</AttributeValuePair>
</Condition>
<Condition name="authlevelge-nonrealmqualified" type="AuthLevelCondition">
<AttributeValuePair><Attribute name="AuthLevel"/>
<Value>1</Value>
</AttributeValuePair>
</Condition>
<Condition name="authlevelle-realmqaulfied" type="LEAuthLevelCondition">
<AttributeValuePair><Attribute name="AuthLevel"/>
<Value>/:2</Value>
</AttributeValuePair>
</Condition>
<Condition name="sessionproperties" type="SessionPropertyCondition">
<AttributeValuePair><Attribute name="valueCaseInsensitive"/>
<Value>true</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="a"/><Value>10</Value>
<Value>20</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="b"/><Value>15</Value>
<Value>25</Value>
</AttributeValuePair>
</Condition>
```

```
<Condition name="activesessiontime" type="SessionCondition">
<AttributeValuePair><Attribute name="TerminateSession"/>
<Value>session_condition_false_value</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="MaxSessionTime"/>
<Value>30</Value>
</AttributeValuePair>
</Condition>
<Condition name="authelevelle-nonrealmqualfied"
        type="LEAuthLevelCondition">
<AttributeValuePair><Attribute name="AuthLevel"/>
<Value>2</Value>
</AttributeValuePair>
</Condition>
<Condition name="ipcondition" type="IPCondition">
<AttributeValuePair><Attribute name="DnsName"/>
<Value>*.iplanet.com</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="EndIp"/>
<Value>145.15.15.15</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="StartIp"/>
<Value>120.10.10.10</Value>
</AttributeValuePair>
</Condition>
<Condition name="authchain-realmqualfied"
       type="AuthenticateToServiceCondition">
<AttributeValuePair><Attribute name="AuthenticateToService"/>
<Value>/:ldapService</Value>
</AttributeValuePair>
</Condition>
<Condition name="auth to realm"
    type="AuthenticateToRealmCondition">
<AttributeValuePair><Attribute name="AuthenticateToRealm"/>
<Value>/</Value>
</AttributeValuePair>
</Condition>
<Condition name="authlevelge-realmqualified"
    type="AuthLevelCondition">
<AttributeValuePair><Attribute name="AuthLevel"/>
<Value>/:2</Value>
</AttributeValuePair>
</Condition>
```

```
<Condition name="authchain-nonrealmqualfied"
   type="AuthenticateToServiceCondition">
<AttributeValuePair><Attribute name="AuthenticateToService"/>
<Value>ldapService</Value>
</AttributeValuePair>
</Condition>
<Condition name="timecondition" type="SimpleTimeCondition">
<AttributeValuePair><Attribute name="EndTime"/>
<Value>17:00</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="StartTime"/>
<Value>08:00</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="EndDate"/>
<Value>2006:07:28</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="EnforcementTimeZone"/>
<Value>America/Los_Angeles</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="StartDay"/>
<Value>mon</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="StartDate"/>
<Value>2006:01:02</Value>
</AttributeValuePair>
<AttributeValuePair><Attribute name="EndDay"/>
<Value>fri</Value>
</AttributeValuePair>
</Condition>
</Conditions>
<ResponseProviders name="responseproviders"
     description="description">
<ResponseProvider name="idresponseprovidere"
   type="IDRepoResponseProvider">
<AttributeValuePair>
<Attribute name="DynamicAttribute"/>
</AttributeValuePair>
<AttributeValuePair>
<Attribute name="StaticAttribute"/>
<Value>m=10</Value>
<Value>n=30</Value>
</AttributeValuePair>
```

```
</ResponseProvider>
</ResponseProviders>
</Policy>
```

2 **策略 XML 文件生成之后，便可使用以下命令加载它：**

```
AccessManager-base/SUNWam/bin/amadmin
--runasdn "uid=amAdmin,ou=People,default_org,
root_suffix"
--password password
--data policy.xml
```

要同时添加多个策略，请将这些策略放在一个 XML 文件中，而不是在每个 XML 文件中放一个策略。如果一连串使用多个 XML 文件装入策略，则可能会损坏内部策略索引，并且某些策略可能不会参与策略评估。

通过 `amadmin` 创建策略时，确保在创建验证方案条件时验证模块已在领域中注册；创建领域、LDAP 组、LDAP 角色和 LDAP 用户主题时存在相应的 LDAP 对象领域、组、角色和用户；创建 `IdentityServerRoles` 主题时存在 Access Manager 角色；以及创建子领域或对等领域引用项时存在相关领域。

请注意，`SubrealmReferral`、`PeerRealmReferral`、`Realm` 主题、`IdentityServerRoles` 主题、`LDAPGroups` 主题、 `LDAPRoles` 主题和 `LDAPUsers` 主题中的值元素的文本中需要完整 DN。

▼ 使用 Access Manager 控制台创建常规策略

1 **选择要为其创建策略的领域。**

2 **单击“策略”选项卡。**

3 **在“策略”列表中单击“新建策略”。**

4 **为策略添加名称和说明。**

5 **如果您希望激活此策略，请在“活动”属性里选中“是”。**

6 **此时，您不必为常规策略定义所有字段。您可以在创建策略之后再添加规则、主题、条件和响应提供者等内容。有关详细信息，参见第 107 页中的“管理策略”。**

7 **单击“确定”。**

▼ 使用 Access Manager 控制台创建引用策略

1. 选择要为其创建策略的领域。
2. 在“ 策略 ”选项卡中单击“ 新建引用 ”。
3. 为策略添加名称和说明。
4. 如果您希望激活此策略，请在“ 活动 ”属性里选中“ 是 ”。
5. 此时，不必为引用策略定义所有字段。您可以在创建策略之后再添加规则、引用等内容。有关详细信息，参见第 107 页中的“ 管理策略 ”。
6. 单击“确定”。

为对等领域和子领域创建策略

要为对等领域或子领域创建策略，必须首先在父领域（或其他对等领域）中创建引用策略。引用策略的规则定义中必须包含子领域所管理的资源前缀。一旦在父领域（或其他对等领域）中创建了引用策略，便可在子领域（或对等领域）创建常规策略。

在本示例中，o=isp 是父领域，o=example.com 是管理 http://www.example.com 的资源和子资源的子领域。

▼ 为子领域创建策略

1. 在 o=isp 中创建引用策略。有关引用策略的信息，参见第 110 页中的“ 修改引用策略 ”过程。

   引用策略必须将 http://www.example.com 定义为规则中的资源，并且必须包含一个以 example.com 作为引用中的值的 SubRealmReferral 。
2. 找到子领域 example.com。
3. 既然资源是通过 isp 引用 example.com，就可以为资源 http://www.example.com，或任何以 http://www.example.com 开头的资源创建常规策略。

   要为 example.com 所管理的其他资源定义策略，必须在 o=isp 上创建其他引用策略。

将策略导出到其他 Access Manager 实例

Access Manager 允许使用 amadmin 命令行工具来导出策略。这在您希望将多个现有策略移动到另一个 Access Manager 实例或者您希望检查以批量模式对现有策略所做的更改时非常有用。要导出策略，使用 amadmin 命令行实用程序将指定策略导出到文件。语法为：

amamdin - u *username* –w *password* –ofilename *output_file.xml* –t *policy_data_file.xml*

可以在策略名称中使用通配符 (*) 来匹配任意字符串。

以下是 *policy_data_file.xml* 的一个示例：

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!--
    Copyright (c) 2005 Sun Microsystems, Inc. All rights reserved
    Use is subject to license terms.
-->

<!DOCTYPE Requests
    PUBLIC "-//iPlanet//Sun Java System Access Manager 6.2 Admin CLI DTD//EN"
    "/opt/SUNWam/dtd/amAdmin.dtd"
>>

<!--  CREATE REQUESTS -->

<!-- to export to file use option -ofilename fileName -->

<Requests>

<RealmRequests >
<RealmGetPolicies realm="/" >
<AttributeValuePair>
<Attribute name="policyName"/>
<Value>p*</Value>
</AttributeValuePair>
</RealmGetPolicies>
</RealmRequests>

<RealmRequests >
<RealmGetPolicies realm="/" >
<AttributeValuePair>
<Attribute name="policyName"/>
<Value>g10</Value>
<Value>g11</Value>
</AttributeValuePair>
```

```
</RealmGetPolicies>

</RealmRequests>
<RealmRequests >
<RealmGetPolicies realm="/realm1" >
<AttributeValuePair>
<Attribute name="policyName"/>
<Value>*</Value>
</AttributeValuePair>
</RealmGetPolicies>
</RealmRequests>

</Requests>
```

策略将导出到 *Output_file.xml* 文件。现在可以对包含在文件中的策略定义做出任何修改。在将策略导入到另一个 Access Manager 实例中之前，必须更改输出文件，使它与 `amadmin` 命令实用程序兼容。有关如何导入策略的说明，包括与 `amadmin` 兼容的策略数据文件的示例，参见使用 amadmin 创建策略

管理策略

一旦创建了标准或引用策略并将其添加到 Access Manager，您就可以使用 Access Manager 控制台通过修改规则、主题、条件和引用项来管理策略。

修改常规策略

通过“策略”选项卡，可以修改定义访问权限的常规策略。 您可以定义和配置多个规则、主题、条件和资源比较器。 本节列出并介绍相关操作步骤。

▼ 在常规策略中添加或修改规则

1. **如果已创建了策略，请单击要为其添加规则的策略的名称。否则，参见第 104 页中的“使用 Access Manager 控制台创建常规策略”。**
2. **在“规则”菜单中单击“新建”。**
3. **请为规则选择以下任一默认服务类型。如果策略可用的服务较多时，您看到的列表可能会比较长：**

搜索服务	为搜索服务查询定义授权操作,并通过指定资源的 Web 服务客户机修改调用的协议。

Liberty 个人配置文件服务	为 Liberty 个人配置文件服务查询定义授权操作,并通过指定资源的 Web 服务客户机修改调用的协议。
URL 策略代理	定义 URL 策略代理服务的授权操作。它用于定义保护 HTTP 和 HTTPS URL 的策略。这是 Access Manager 策略最常见的用途。

4 单击“ 下一步 ”。

5 请输入规则的名称及其资源名称。

目前，Access Manager 策略代理只支持 `http://` 和 `https://` 资源，而不支持使用 IP 地址代替主机名。

协议、主机、端口和资源名称都支持通配符。例如：

```
http*://*:*/*.html
```

对于“URL 策略代理”服务，如果未输入端口号，则 `http://` 的默认端口号是 80，`https://` 的默认端口号是 443。

6 为规则选择操作。根据服务类型，可选择以下选项：

- 查找（搜索服务）
- 更新（搜索服务）
- 修改（Liberty 个人配置文件服务）
- 查询（Liberty 个人配置文件服务）
- GET（URL 策略代理）
- POST（URL 策略代理）

7 选择操作值。

- 同意交互式操作 — 为了得到同意而对资源调用 Liberty 交互式操作协议。仅适用于 Liberty 个人配置文件服务类型。
- 交互式操作值 — 为了获得某个值而对资源调用 Liberty 交互式操作协议。仅适用于 Liberty 个人配置文件服务类型。
- 允许 — 允许您访问与规则中定义的资源相匹配的资源。
- 拒绝 — 拒绝您访问与规则中定义的资源相匹配的资源。

  在策略中，拒绝规则始终比允许规则具有优先权。例如，如果某种给定的资源存在两个策略，一个拒绝访问而另一个允许访问，结果为拒绝访问（假定两个策略的条件都满足）。建议谨慎使用拒绝策略，因为它们会导致策略间的潜在冲突。通常来说，在定义策略的过程中，应只使用允许规则，在没有策略适用于实现拒绝条件时使用默认的拒绝规则。

当采用了显示拒绝规则时，即使一个或多个策略允许访问，通过多个不同主题（如角色和/或组成员资格）指定给给定用户的策略可能仍然会导致拒绝访问资源。例如，如果应用于“员工”角色的资源的策略为拒绝策略，而应用于“经理”角色的同一资源的策略为允许策略，则被分配了“员工”和“经理”两个角色的用户的策略决策将为拒绝。

解决此问题的一个方法是使用条件插件来设计策略。在上述情况下，将拒绝策略应用于通过“员工”角色验证的用户并将允许策略应用于通过“经理”角色验证的用户的“角色条件”可以帮助区分两种策略。 另一个方法是使用 `authentication level` 条件，其中 “经理” 角色在更高验证级别进行验证。

8 **单击“完成”。**

▼ 在常规策略中添加或修改主题

1 **如果已创建了策略，请单击要为其添加主题的策略的名称。如果尚未创建策略，参见第 104 页中的 “ 使用 Access Manager 控制台创建常规策略 ”。**

2 **在 “ 主题 ” 列表中单击 “ 新建 ” 。**

3 **选择以下任一默认主题类型。有关主题类型的说明，参见第 90 页中的 “ 主题 ”**

4 **单击 “ 下一步 ” 。**

5 **输入主题的名称。**

6 **选择或取消选择 “ 排除 ” 字段。**

如果未选择该字段（默认），策略将应用到属于该主题的成员的身份。如果选择该字段，策略将应用到**不**属于该主题的成员的身份。

如果该策略中存在多个主题，至少要有一个主题表明该策略适用于给定的身份，策略才能应用到该身份。

7 **执行搜索以显示要添加到主题的身份。此步骤不适用于“验证的用户”主题或“Web 服务客户机”主题。**

默认 (*) 搜索模式将显示所有符合条件的条目。

8 **选择要为主题添加的各个身份，或单击 “ 全部添加 ” 一次添加所有身份。单击“添加”将这些身份移至“选定”列表中。 此步骤不适用于 “ 已验证用户 ” 主题。**

9 **单击“完成”。**

10 **要从策略中删除主题，请选择相应主题并单击 “ 删除 ” 。您可以通过单击主题名称来编辑任何主题定义。**

▼ 向常规策略添加条件

1. 如果已创建了策略，请单击要为其添加条件的策略的名称。如果尚未创建策略，参见第 104 页中的“ 使用 Access Manager 控制台创建常规策略 ”
2. 在“ 条件 ”列表中单击“ 新建 ”。
3. 选择条件类型，然后单击“下一步”。
4. 定义条件类型字段。
5. 单击“完成”。

▼ 向常规策略添加响应提供者

1. 如果已创建了策略，请单击要为其添加响应提供者的策略的名称。如果尚未创建策略，参见第 104 页中的“ 使用 Access Manager 控制台创建常规策略 ”。
2. 在“ 响应提供者 ”列表中单击“ 新建 ”。
3. 请输入响应提供者的名称。
4. 定义以下值：

   StaticAttribute — 这是属性值格式的静态属性，在存储于策略中的 `IDResponseProvider` 实例中进行定义。

   DynamicAttribute — 应首先在相应领域的“策略配置服务”中定义此处所选择的响应属性。所定义的属性名称应为已配置数据存储库 (IDRepository) 中现有属性名称的子集。有关如何定义属性的详细信息，参见“策略配置”属性定义。要选择特定属性或多个属性，请按住 "Ctrl" 键并单击鼠标左键。

5. 单击“完成”。
6. 要从策略中删除响应提供者，请选择相应主题，然后单击“删除”。您可以通过单击响应提供者名称来编辑任何响应提供者的定义。

修改引用策略

可将领域的策略定义和决策委托给使用引用策略的不同领域。 自定义引用项可用于从任意策略目标点获取策略决策。 一旦创建了引用策略，便可添加或修改相关的规则、引用项和资源提供者。

▼ 在引用策略中添加或修改规则

1 如果已创建了策略，请单击要为其添加规则的策略的名称。否则，参见第 105 页中的“ 使用 Access Manager 控制台创建引用策略 ”。

2 在“规则”菜单中单击“新建”。

3 请为规则选择以下任一默认服务类型。如果策略可用的服务较多时，您看到的列表可能会比较长：

搜索服务	为搜索服务查询定义授权操作,并通过指定资源的 Web 服务客户机修改调用的协议。
Liberty 个人配置文件服务	为 Liberty 个人配置文件服务查询定义授权操作,并通过指定资源的 Web 服务客户机修改调用的协议。
URL 策略代理	定义 URL 策略代理服务的授权操作。它用于定义保护 HTTP 和 HTTPS URL 的策略。这是 Access Manager 策略最常见的用途。

4 单击“ 下一步 ”。

5 请输入规则的名称及其资源名称。

目前， Access Manager 策略代理只支持 `http://` 和 `https://` 资源， 而不支持使用 IP 地址代替主机名。

协议、主机、端口和资源名称都支持通配符。例如：

```
http*://*:*/*.html
```

对于“URL 策略代理”服务， 如果未输入端口号， 则 `http://` 的默认端口号是 80， `https://` 的默认端口号是 443。

注 – 步骤 6 和 7 对引用策略不适用。

---

6 单击“完成”。

▼ 在策略中添加或修改引用项

1 如果已经创建了策略，请单击要为其添加响应提供者的策略的名称。如果尚未创建策略，参见第 105 页中的“ 使用 Access Manager 控制台创建引用策略 ”。

2 在“ 引用 ”列表中，单击“ 新建 ”。

3 **在“规则”字段中定义资源。这些字段包括：**

引用— 显示当前引用类型。

名称— 输入引用的名称。

资源名称— 输入资源的名称。

过滤器—为将在“值”字段中显示的领域名称指定过滤器。默认情况下，该字段将显示所有领域名称。

值 —选择引用的领域名称。

4 **单击“完成”。**

要从策略中移除引用，请选择该引用，然后单击“删除”。

您可以通过单击引用名称旁边的“编辑”链接来编辑任何引用定义。

▼ 向引用策略添加响应提供者

1 **如果已创建了策略，请单击要为其添加响应提供者的策略的名称。如果尚未创建策略，参见****第 105 页中的“使用 Access Manager 控制台创建引用策略”****。**

2 **在“响应提供者”列表中单击“新建”。**

3 **请输入响应提供者的名称。**

4 **定义以下值：**

StaticAttribute	这是属性值格式的静态属性，在存储于策略中的 `IDResponseProvider` 实例中进行定义。
DynamicAttribute	应首先在相应领域的“策略配置服务”中定义此处所选择的响应属性。所定义的属性名称应为已配置数据存储库 (IDRepository) 中现有属性名称的子集。有关如何定义属性的详细信息，参见“策略配置”属性定义。要选择特定属性或多个属性，请按住 "Ctrl" 键并单击鼠标左键。

5 **单击“完成”。**

6 **要从策略中删除响应提供者，请选择相应主题，然后单击“删除”。您可以通过单击响应提供者名称来编辑任何响应提供者的定义。**

策略配置服务

“策略配置”服务用于通过 Access Manager 控制台为每个组织配置与策略相关的属性。 也可以定义资源名称实现和与 Access Manager 策略框架一同使用的 Directory Server 数据存储库。 在“策略配置服务”中指定的 Directory Server 用于 LDAP 用户、LDAP 组、LDAP 角色以及组织策略主题的成员资格评估。

主题结果的生存时间

为了提高策略评估的性能，成员资格评估将被缓存一段时间，具体时间长短如“策略配置”服务中的“主题结果的生存时间”属性定义。 在达到“主题结果的生存时间”属性中所定义的时间之前，将持续使用这些缓存的成员资格决策。在此之后的成员资格评估用于反映目录中用户的当前状态。

动态属性

这些是所允许的动态属性名称，它们显示在列表中，并且可通过选择它们来定义策略响应提供者的动态属性。所定义的名称需与数据系统信息库中定义的属性名称相同。

amldapuser 定义

`amldapuser` 是默认情况下安装“策略配置”服务中指定的 Directory Server 期间创建的用户。 如有必要，管理员或领域的策略管理员可以对其进行更改。

添加策略配置服务

创建领域时，将为领域自动设置“策略配置”服务属性。 但是，必要时也可修改属性。

基于资源的验证

某些组织需要高级验证方案，其中用户将根据他们尝试要访问的资源用特定模块进行验证。基于资源的验证是 Access Manager 的一项功能，该功能要求用户必须通过用于保护资源的特定验证模块而非默认验证模块进行验证。 此功能仅适用于首次用户验证。

注 – 该功能与第 83 页中的 “会话升级”中所描述的基于资源的验证不同。后者不具有任何限制。

限制

基于资源的验证包含以下限制：

- 如果适用于此资源的策略包含多个验证模块，系统将任意选择一个验证模块。
- 级别和模式是可为此策略定义的仅有的两个条件。
- 此功能在不同的 DNS 域中不起作用。

▼ 配置基于资源的验证

一旦安装了 Access Manager 和策略代理，便可对基于资源的验证进行配置。 要执行此操作，需要将 Access Manager 指向网关 servlet。

1 **打开** `AMAgent.properties`。

`AMAgent.properties` 可在 `/etc/opt/SUNWam/agents/config/` 中找到（在 Solaris 环境中）。

2 **注释掉以下行：**

```
#com.sun.am.policy.am.loginURL = http://Access
Manager_server_host.domain_name:port/amserver/UI/Login
```

。

3 **将以下行添加到 文件：**

```
com.sun.am.policy.am.loginURL =
http://AccessManager_host.domain_name:port/amserver/gateway
```

注 – 网关 servlet 是使用“策略评估 API”开发的，可使用它来编写用于完成基于资源的验证的自定义机制。参见 Access Manager 开发者指南中《Sun Java System Access Manager 7.1 Developer's Guide》一书中的《Sun Java System Access Manager 7.1 Developer's Guide》中的第 3 章 “Using the Policy APIs”。

---

4 **重新启动代理。**

◆ ◆ ◆ 第 6 章

管理主题

“主题”界面启用领域中的基本身份管理。任何在“主题”界面中创建的身份都能在策略（以“Access Manager 身份主题”类型创建的策略）的主题定义中使用。

您可以创建和修改的身份包括：

用户

用户代表个体身份。可以在组中创建和删除用户，也可以在角色和/或组中添加或删除用户。还可以将服务指定给用户。

▼ 创建或修改用户

1 **单击“ 用户 ”选项卡。**

2 **单击“新建”。**

3 **为以下字段输入数据：**

用户 ID。此字段中应填入用户用来登录到 Access Manager 的名称。此属性可以是一个非 DN 值。

名字。此字段中应填入用户的名字。

姓氏。此字段中应填入用户的姓氏。

全名 — 此字段中应填入用户的全名。

密码— 此字段中应填入“用户 ID”字段中所指定名称的密码。

密码（确认）— 确认密码。

用户状态。此选项指示是否允许用户通过 Access Manager 进行验证。

4 单击“创建”。

5 创建用户之后，您可以单击用户的名称来编辑用户信息。有关用户属性的信息，参见“用户”属性。您可以进行的其他修改包括：

- 第 115 页中的 “创建或修改用户”
- 第 116 页中的 “向角色和组添加用户”
- 第 116 页中的 “向身份添加服务”

▼ 向角色和组添加用户

1 单击所要修改的用户的名称。

2 选择角色或组。仅显示已指定给用户的那些角色和组。

3 从“可用”列表中选择角色或组，然后单击“添加”。

4 当“选定”列表中显示所选的角色或组时，单击“保存”。

▼ 向身份添加服务

1 选择您要添加服务的身份。

2 单击“服务”选项卡。

3 单击“添加”。

4 根据您选择的身份类型，将显示以下服务列表：

- 验证配置
- 搜索服务
- Liberty 个人配置文件服务
- 会话
- 用户

5 选择您要添加的服务，然后单击“下一步”。

6 **编辑服务的属性。有关服务的说明，请单击步骤 4 中服务的名称。**

7 **单击“完成”。**

代理配置文件

Access Manager 策略代理对 Web 服务器和 Web 代理服务器上的内容提供保护以防止未授权的侵入。它们基于管理员配置的策略来控制对服务和 Web 资源的访问。

代理对象定义了“策略代理”配置文件，并允许 Access Manager 存储与保护 Access Manager 资源的特定代理有关的验证及其他配置文件信息。通过 Access Manager 控制台，管理员可以查看、创建、修改和删除代理配置文件。

可以在代理对象创建页面定义代理用于通过 Access Manager 进行验证的 UID/密码。如果有多个使用相同 Access Manager 设置的 Web 容器，您可以选择为不同的代理启用多个 ID 并且可以从 Access Manager 单独将其启用和禁用。您也可以集中管理代理的一些首选项值，而不必在每台计算机上都编辑 `AMAgent.properties`。

▼ 创建或修改代理

1 **单击“ 代理 ”选项卡。**

2 **单击“新建”。**

3 **输入以下字段的值：**

名称。输入代理的名称或身份。此名称是代理用来登录的名称。不接受多字节用户名称。

密码。输入代理的密码。此密码必须与在 LDAP 验证过程中代理所使用的密码不同。

确认密码。确认密码。

设备状态。输入代理的设备状态。如果设置为“活动”，则代理可以通过进行验证并与 Access Manager 进行通信。如果设置为“不活动”，则代理不能通过 Access Manager 进行验证。

4 **单击“ 创建 ”。**

5 **创建代理之后，您可以另外编辑以下字段：**

说明。输入代理的简短说明。例如，可以输入代理实例名称或其保护的应用程序的名称。

代理关键字值。使用关键字/值对设置代理属性。Access Manager 使用此属性接收有关用户的证书声明的代理请求。当前仅一个属性有效，所有其他属性都将被忽略。请使用以下格式：

agentRootURL=protocol:// *hostname:port/*

条目必须准确，而且 agentRootURL 区分大小写。

protocol　代表所使用的协议，即 HTTP 或 HTTPS。

hostname　代表代理所驻留的计算机的主机名。此计算机也托管由代理保护的资源。

port　代表安装代理的端口号。代理侦听此端口上的接收通信，并拦截所有要访问主机资源的请求。

配置 Access Manager 以防止 Cookie 劫持

Cookie 劫持就是指冒名顶替者（可能是使用不可信应用程序的黑客）对 cookie 进行未授权的访问。如果被劫持的 cookie 是会话 cookie，则根据系统的配置方式，cookie 劫持可能增加对受保护 Web 资源的未授权访问威胁。

Sun 文档提供标题为“Precautions Against Session-Cookie Hijacking in an Access Management Deployment”（在访问管理部署中防止会话 Cookie 劫持的预防措施）的技术说明，该说明介绍要对抗与会话 cookie 劫持相关的特定安全威胁可采取的预防措施参见以下文档：

《Technical Note: Precautions Against Cookie Hijacking in an Access Manager Deployment》

过滤的角色

过滤的角色是用 LDAP 过滤器创建的动态角色。在创建角色时，会通过过滤器过滤所有用户并为其指定该角色。过滤器会查找条目中的所有属性值对（例如，ca=user*），并自动将包含该属性的用户指定给角色。

▼ 创建过滤的角色

1. **在“浏览”窗格中，找到要在其中创建角色的组织。**
2. **单击“新建”。**
3. **输入过滤的角色的名称。**

4 **输入搜索条件信息。**

例如，

```
(&(uid=user1)(|(inetuserstatus=active)(!(inetuserstatus=*))))
```

如果过滤器保留为空，将默认创建以下角色：

```
(objectclass = inetorgperson)
```

5 **单击"创建"以基于过滤条件启动搜索。由过滤条件定义的身份将会自动指定给角色。**

6 **创建过滤的角色后，单击角色的名称可查看属于该角色的“ 用户 ”。此外，还可以通过单击“ 服务 ”选项卡将服务添加到角色。**

角色

角色的成员是指拥有该角色的 LDAP 条目。角色本身的条件被定义为具有属性的 LDAP 条目，由条目的标识名 (Distinguished Name, DN) 属性来标识。创建角色之后，请手动添加服务和用户。

▼ 创建或修改角色

1 **单击“ 角色 ”选项卡。**

2 **在“ 角色 ”列表中单击“ 新建 ”。**

3 **输入角色的名称。**

4 **单击“ 创建 ”。**

▼ 向角色或组添加用户

1 **单击要为其添加用户的角色或组的名称。**

2 **单击“ 用户 ”选项卡。**

3 **从“ 可用 ”列表中选择您要添加的用户，并单击“ 添加 ”。**

4 **当"选定"列表中显示所选的用户时，单击"保存"。**

组

组代表具有共同功能、特性或利益的用户集合。通常来说，这种分组不会涉及权限。组可存在于两个级别，分别是组织和其他受管组中。

▼ 创建或修改组

1 **单击"组"选项卡。**

2 **在“ 组 ”列表中单击“ 新建 ”。**

3 **输入组的名称。**

4 **单击“ 创建 ”。**
创建组之后，您可以单击组的名称，然后单击“用户”选项卡，将用户添加到组。

第 2 部分

目录管理和默认服务

此为《Sun Java System Access Manager 7.1 管理指南》的第二部分。“目录管理”一章介绍当 Access Manager 在“传统模式”下部署时如何管理“目录”对象。其他几章介绍如何配置和使用 Access Manager 的一些默认服务。本部分包含以下各章：

◆ ◆ ◆ 第 7 章

7 目录管理

只有以“传统”模式安装 Access Manager 时才显示“目录管理”选项卡。此目录管理功能可以为已启用了 Sun Java System Directory Server 的 Access Manager 部署提供身份管理解决方案。

有关“传统模式”安装选项的详细信息，参见《Sun Java Enterprise System 5 Installation Guide for UNIX》

管理目录对象

“目录管理”选项卡包含查看和管理 Directory Server 对象所需的所有组件。本部分说明对象类型及其详细配置方法。可以使用 Access Manager 控制台或命令行界面来定义、修改或删除用户、角色、组、组织、子组织和容器对象。控制台有默认的管理员，他们拥有不同的权限级别以创建和管理目录对象。（可以基于角色创建其他管理员。）与 Access Manager 一起安装时，在 Directory Server 中会定义管理员。 可以管理的 Directory Server 对象有：

组织

在企业用来管理部门和资源的层次结构中，**组织**代表其最高一级。Access Manager 在安装时会动态创建一个顶级组织（在安装期间定义）来管理 Access Manager 企业配置。安装后可以创建其他组织，以管理单独的企业。创建的所有组织都位列顶级组织之下。

▼ 创建组织

1 **单击“目录管理”选项卡。**

2 **在“组织”列表中，单击“新建”。**

3 **输入字段的值。仅“名称”是必需字段。这些字段包括：**

名称 输入组织名称的值。

域名 输入组织的完整域名系统 (DNS) 名称（如果存在）。

组织状态 选择**活动**或**不活动**状态。默认值为**活动**。在组织存在期间，可以随时选择“属性”图标以更改其状态。如果选择**不活动**状态，则当登录到组织时，将禁止用户访问。

组织别名 该字段定义组织的别名，以允许您在通过 URL 登录时使用别名进行验证。例如，如果组织的名称为 `exampleorg`，将 `123` 和 `abc` 定义为组织的别名，则可以使用以下任一 URL 登录到组织：

```
http://machine.example.com/amserver/UI/Login?org=exampleorg

http://machine.example.com/amserver/UI/Login?org=abc

http://machine.example.com/amserver/UI/Login?org=123
```

组织别名在组织中必须唯一。可以使用“唯一属性列表”来强制执行唯一性。

DNS 别名 允许为组织的 DNS 名称添加别名。该属性只接受“真实的”域别名（不允许使用随机字符串）。例如，如果 DNS 的名称为 `example.com`，而名为 `exampleorg` 的组织的别名定义为 `example1.com` 和 `example2.com`，则可以使用以下任一 URL 登录到组织：

```
http://machine.example.com/amserver/UI/

Login?org=exampleorg

http://machine.example1.com/amserver/

UI/Login?org=exampleorg

http://machine.example2.com/amserver/

UI/Login?org=exampleorg
```

唯一属性列表 用于添加组织中用户的唯一属性名列表。例如，如果添加用于指定电子邮件地址的唯一属性名，则不能创建两个使用相同电子邮件地址的

用户。也可以在该字段中输入以逗号分隔的列表。列表中的任一属性名均定义了唯一性。例如，如果该字段包含以下属性名列表：

```
PreferredDomain, AssociatedDomain
```

并且针对特定用户 `PreferredDomain` 被定义为 `http://www.example.com`，则整个以逗号分隔的列表在该 URL 中唯一。将命名属性 'ou' 添加到“唯一属性列表”并不会强制执行默认组和人员容器的唯一性。(ou=Groups,ou=People)。

对于所有子组织都强制执行唯一性。

注 – 在“领域”模式下无法设置唯一属性。在“传统”模式下，也无法在基于 7.0 或 7.1 的控制台中设置它们。要创建唯一属性列表，必须登录到基于 6.3 的控制台中。有关详细信息，参见第 19 页中的 “传统模式 6.3 控制台”。

---

4 **单击“确定”。**

新组织将显示在“组织”列表中。要编辑在创建组织过程中定义的属性，请单击要编辑的组织的名称，更改属性，然后单击“保存”。

▼ 删除组织

1 **选中要删除的组织名称旁边的复选框。**

2 **单击“ 删除 ”。**

注 – 执行删除时不会显示警告消息。组织内的所有条目都将被删除，并且不能执行撤消操作。

将组织添加到策略

可以通过定义策略的主题将 Access Manager 对象添加到策略中。创建或修改策略时，组织、角色、组和用户可以被定义为主题。定义了主题后，策略将被应用到对象。有关详细信息，参见第 107 页中的 “管理策略”。

容器

当由于对象类和属性的不同而无法使用组织条目时，将使用**容器**条目。请切记 Access Manager 容器条目和 Access Manager 组织条目不必等同于 LDAP 对象类 `organizationalUnit` 和 `organization`。它们是抽象身份条目。理想情况下，将使用组织条目而不使用容器条目。

注 – 容器的显示是可选的。要查看容器，必须选择“配置”>“控制台属性”下方“管理”服务中的“显示容器”。

▼ 创建容器

1. **选择将在其中创建新容器的组织或容器的位置链接。**
2. **单击“容器”选项卡。**
3. **在“容器”列表中单击“新建”。**
4. **输入要创建的容器的名称。**
5. **单击“确定”。**

▼ 删除容器

1. **单击“容器”选项卡。**
2. **选中要删除的容器名称旁边的复选框。**
3. **单击“ 删除 ”。**

注 – 如果删除某个容器，就会删除该容器中存在的所有对象，包括所有对象和子容器。

组容器

组容器用于管理组。它只能包含组和其他组容器。组容器组会被动态指定为所有受管组的父项。如果需要，可以添加其他组容器。

注 – 组容器的显示是可选的。要查看组容器，必须选择“配置”>“控制台属性”下方“管理”服务中的“启用组容器”。

▼ 创建组容器

1. **选择将包含新的组容器的组织或组容器的位置链接。**
2. **选择“组容器”选项卡。**
3. **在“组容器”列表中单击“新建”。**
4. **在“名称”字段中输入值，然后单击“确定”。新的组容器将显示在“组容器”列表中。**

▼ 删除组容器

1. **找到包含要删除的组容器的组织。**
2. **选择“组容器”选项卡。**
3. **选中要删除的组容器旁边的复选框。**
4. **单击“ 删除 ”。**

组

组代表具有共同职责、特征或利益的用户集合。通常来说，这种分组不会涉及权限。组可存在于两个级别，分别是组织和其他受管组中。存在于其他组中的组称为**子组**。子组是“物理上”存在于父组内的子节点。

Access Manager 还支持**嵌套组**，它“代表”了单个组中所包含的现有组。与子组不同，嵌套组可以存在于 DIT 中的任何位置。使用嵌套组可以快速地为多个用户设置访问权限。

可以创建两种类型的组：静态组和动态组。只能手动将用户添加到静态组，动态组则通过过滤器控制用户的添加。这两种类型的组中都可以添加嵌套组或子组。

静态组

静态组是根据您指定的“受管组类型”创建的组。 使用 `groupOfNames` 或 `groupOfUniqueNames` 对象类将组成员添加到组条目中。

注 – 默认情况下，受管组类型为动态的。 您可以在“管理”服务配置中更改此默认设置。

动态组

动态组是通过使用 LDAP 过滤器创建的。所有条目均由过滤器过滤并动态分配给组。过滤器将查找条目中的任何属性，并返回那些包含特定属性的条目。例如，如果您要根据构建号创建组，则可以使用过滤器返回一组包含该构建号属性的用户。

注 – 要使用参考完整性插件，应将 Access Manager 与 Directory Server 一起进行配置。当启用参照完整性插件后，它将直接在删除或重命名操作后对指定属性执行完整性更新。这将确保在整个数据库中维持相关条目之间的关系。数据库索引则增强了 Directory Server 中的搜索性能。有关启用此插件的详细信息，参见 Sun Java Access Manager 6 迁移指南。

▼ 创建静态组

1. **找到要在其中创建新组的组织、组或组容器。**
2. **在“组”列表中，单击“新建静态”。**
3. **在“名称”字段中输入组的名称。单击“下一步”。**
4. **选择“用户可以订阅此组”属性可以使用户自行订阅组。**
5. **单击“确定”。**
   组创建完毕后，可以通过选择组的名称并单击“常规”选项卡来编辑“用户可以订阅此组”属性。

▼ 向静态组添加成员或从中移除

1. **在“组”列表中，选择要添加成员的组。**
2. **在“选择操作”菜单中选择要执行的操作。可以执行以下操作：**

   新建用户　保存用户信息时，此操作将创建新用户并将该用户添加到组。

   添加用户　此操作可将现有用户添加到组。选择此操作后，请创建搜索条件指定要添加的用户。用于构建该条件的字段使用 ANY 或 ALL 运算符。 ALL 将根据所有指定的字段向用户返回结果。ANY 将根据所指定的任一字段向用户返回结果。如果某个字段保留为空，则该字段将匹配该特定属性的所有可能条目。

搜索标准创建完毕后，单击“下一步”。从返回的用户列表中，选择要添加的用户并单击“完成”。

添加组　此操作可以将嵌套组添加到当前组。选择此操作时要创建搜索条件，包括搜索范围、组的名称（允许使用通配符“*”），并且可指定用户是否可以自行订阅组。信息输入完毕后，单击“下一步”。从返回的组列表中，选择要添加的组并单击“完成”。

移除成员　此操作将从组中移除成员（包括用户和组），但不会将其删除。选择要移除的成员并从“选择操作”菜单中选择“移除成员”。

删除成员　此操作将永久删除所选成员。选择要删除的成员，然后选择“删除成员”。

▼ 创建动态组

1 **找到要在其中创建新组的组织或组。**

2 **单击“组”选项卡。**

3 **单击“新建动态”。**

4 **在“名称”字段中输入组的名称。**

5 **构造 LDAP 搜索过滤器。**

默认情况下，Access Manager 将显示“基本”搜索过滤器界面。用于构造过滤器的“基本”字段使用 ANY 或 ALL 操作符。ALL 将根据所有指定的字段向用户返回结果。ANY 将根据所指定的任一字段向用户返回结果。如果某个字段保留为空，则该字段将匹配该特定属性的所有可能条目。

6 **单击“确定”后，系统会自动将与搜索条件匹配的所有用户添加到组。**

▼ 向动态组添加成员或从中移除

1 **在“组”列表中，单击要添加成员的组的名称。**

2 **在“选择操作”菜单中选择要执行的操作。可以执行以下操作：**

添加组　此操作可以将嵌套组添加到当前组。选择此操作时要创建搜索条件，包括搜索范围、组的名称（允许使用通配符“*”），并且可指定用户是否可以自行订阅组。信息输入完毕后，单击“下一步”。从返回的组列表中，选择要添加的组并单击“完成”。

移除成员　此操作将从组中移除成员（包括组），但不会将其删除。选择要移除的成员，然后选择“移除成员”

删除成员　　此操作将永久删除所选成员。选择要删除的成员，然后选择“删除成员”。

将组添加到策略

可以通过定义策略的主题将 Access Manager 对象添加到策略中。在创建或修改策略时，可以在策略的“主题”页面中将组织、角色、组和用户定义为主题。定义了主题后，策略将被应用到对象。有关详细信息，参见第 107 页中的 “管理策略” 。

人员容器

人员容器是默认的 LDAP 组织单位。在组织中创建用户时，所有的用户都将被指定到该容器。 人员容器位于组织级别和人员容器级别（作为子人员容器）。它们只能包含其他人员容器和用户。如果需要，可以将其他人员容器添加到组织中。

注 – 人员容器的显示是可选的。要查看“人员容器”，必须在“管理服务”中选择“启用人员容器”。

▼ 创建人员容器

1. **找到要在其中创建新人员容器的组织或人员容器。**
2. **在“人员容器”列表中单击“新建”。**
3. **输入要创建的人员容器的名称。**
4. **单击“确定”。**

▼ 删除人员容器

1. **找到包含要删除的人员容器的组织或人员容器。**
2. **选中要删除的人员容器名称旁边的复选框。**
3. **单击“ 删除 ”。**

注 – 删除人员容器将删除该人员容器中存在的所有对象， 包括所有用户和子人员容器。

用户

用户表示个人身份。通过“Access Manager 身份管理”模块，可以在组织、容器和组中创建和删除用户，还可以在角色和/或组中添加或移除用户。此外，还可以将服务指定给用户。

注 – 如果在子组织中创建的用户使用了与 `amadmin` 相同的用户 ID，登录 `amadmin` 时将失败。如果发生了这样的问题，管理员应该通过 Directory Server 控制台更改用户 ID。这样可使管理员登录到默认组织。另外，验证服务中的“起始用户搜索的 DN”可以设置为人员容器 DN，以确保在登录过程中返回唯一匹配项。

▼ 创建用户

1 **找到要在其中创建用户的组织、容器或人员容器。**

2 **单击“ 用户 ”选项卡。**

3 **在“用户”列表中单击“新建”。**

4 **为以下值输入数据：**

用户 ID	此字段中应填入用户用来登录到 Access Manager 的名称。该属性可能是一个非 DN 值。
名字	此字段中应填入用户的名字。“名字”值和“姓氏”值可以标识“当前已登录”字段中的用户。 此值不用必须填写。
姓氏	此字段中应填入用户的姓氏。“名字”值和“姓氏”值可以标识用户。
全名	此字段中应填入用户的全名。
密码	此字段中应填入“用户 ID”字段中所指定名称的密码。
密码（确认）	确认密码。
用户状态	此选项指示是否允许用户通过 Access Manager 进行验证。只有活动用户才能进行验证。默认值为**活动**。

5 **单击“确定”。**

▼ 编辑用户概要文件

当某个尚未指定管理角色的用户通过 Access Manager 进行验证时，默认视图为用户自己的“用户概要文件”视图。 另外，具有适当权限的管理员可以编辑用户概要文件。 在该

视图中，用户可以修改其个人概要文件的属性值。“用户概要文件”视图中显示的属性可以扩展。有关添加对象和身份的自定义属性的详细信息，参见 Access Manager 开发者指南。

1 选择要对其概要文件进行编辑的用户。默认情况下，屏幕上将显示“常规”视图。

2 编辑以下字段：

名字 此字段中应填入用户的名字。

姓氏 此字段中应填入用户的姓氏。

全名 此字段中应填入用户的全名。

密码 单击“编辑”链接以添加和确认用户密码。

电子邮件地址 此字段中应填入用户的电子邮件地址。

员工编号 此字段中应填入用户的员工编号。

电话号码 此字段中应填入用户的电话号码。

家庭地址 此字段中应填入用户的家庭地址。

用户状态 此选项指示是否允许用户通过 Access Manager 进行验证。只有活动的用户才能通过 Access Manager 进行验证。默认值为“活动”。可以从下拉菜单中选择以下任意一个选项：。

- 活动 — 用户可通过 Access Manager 进行验证。
- 不活动 — 用户不能通过 Access Manager 进行验证，但用户概要文件仍会存储在目录中。

注 – 将用户状态更改为“不活动”仅影响通过 Access Manager 进行的验证。Directory Server 使用 *nsAccountLock* 属性来确定用户帐户的状态。禁用 Access Manager 验证的用户帐户仍然可以执行不需要 Access Manager 的任务。要禁用目录中的某个用户帐户，而不仅仅是禁用 Access Manager 验证，应将 *nsAccountLock* 的值设置为 false。如果站点的委托管理员要定期禁用用户，应考虑将 *nsAccountLock* 属性添加到“Access Manager 用户概要文件”页面。有关详细信息，参见《Sun Java System Access Manager 7.1 Developer's Guide》。

帐户失效日期 如果存在此属性，则当前日期和时间超过指定的“帐户失效日期”时，验证服务将不允许进行登录。此属性的格式为 *mm/dd/yyyy hh:mm*。

用户验证配置 此属性设置用户的验证链。

用户别名列表	此字段定义了一组应用于用户的别名。要使用此属性中配置的别名，必须修改 LDAP 服务，即向 LDAP 服务中的“用户条目搜索属性”字段添加 `iplanet-am-user-alias-list` 属性。
首选语言环境	此字段指定用户的语言环境。
成功 URL	此属性指定用户在验证成功后，重定向的 URL。
失败 URL。	此属性指定用户在验证失败后，重定向的 URL。
密码重置选项	此字段用于选择要在忘记密码页面中使用的问题，该页面用来恢复忘记的密码。
用户搜索资源提供	设置用户的“用户搜索”服务的资源提供。
MSISDN 号码	使用 MSISDN 验证时，定义用户的 MSISDN 号码。

▼ 向角色和组添加用户

1. **单击“用户”选项卡。**
2. **单击所要修改的用户的名称。**
3. **选择“角色”或“组”选项卡。**
4. **选择要向其添加用户的角色或组，然后单击“添加”。**
5. **单击“保存”。**

注 – 要从“角色”或“组”中移除用户，请选择角色或组并单击“移除”，然后单击“保存”。

将用户添加到策略

可以通过定义策略的主题将 Access Manager 对象添加到策略中。在创建或修改策略时，可以在策略的“主题”页面中将组织、角色、组和用户定义为主题。定义了主题后，策略将被应用到对象。有关详细信息，参见第 107 页中的 “管理策略”。

角色

角色是与**组**的概念类似的 Directory Server 条目机制。组有成员，角色也有成员。角色的成员是指拥有该角色的 LDAP 条目。角色本身的条件被定义为具有属性的 LDAP 条目，由条目的标识名 (DN) 属性来标识。Directory Server 具有许多不同类型的角色，但 Access Manager 只能管理其中的一种：被管理的角色。

注 – 在目录部署中还可以使用其他的 Directory Server 角色类型，只是它们不能被 Access Manager 控制台管理。还可以在策略的主题定义中使用其他的 Directory Server 类型。有关策略主题的详细信息，参见第 99 页中的 “创建策略” 。

用户可以拥有一个或多个角色。例如，可以创建一个承包商角色，其属性来自“会话服务”和“密码重置服务”。新承包商雇员加入公司时，管理员可以将该角色指定给他们，而不需在承包商条目中分别设置各个属性。如果承包商在工程部工作并且需要适用于工程员工的服务以及访问权限，则管理员可以为该承包商同时指定工程角色和承包商角色。

Access Manager 使用角色来应用访问控制指令。首次安装时，Access Manager 会配置定义管理员权限的访问控制指令 (ACI)。然后在角色（如“组织管理员角色”和“组织帮助台管理员角色”）中指定这些 ACI，当这些角色被指定到用户时，可定义用户的访问权限。

仅当“管理服务”中启用了“在用户概要文件页面中显示角色”属性时，用户才可查看为其分配的角色。

注 – 要使用参考完整性插件，应将 Access Manager 与 Directory Server 一起进行配置。当启用参照完整性插件后，它将直接在删除或重命名操作后对指定属性执行完整性更新。这将确保在整个数据库中维持相关条目之间的关系。数据库索引则增强了 Directory Server 中的搜索性能。

有两种类型的角色：

- 静态 — 静态角色在创建时可以不添加用户。角色创建之后，您可以在其中添加特定用户。这样，在向给定角色添加特定用户时，您可以更好的进行控制。
- 动态 – 动态角色是通过使用 LDAP 过滤器创建的。在创建角色时，会通过过滤器过滤所有用户并为其分配该角色。过滤器会查找条目中的所有属性值对（例如，`ca=user*`），并自动将包含该属性的用户分配给角色。

▼ 创建静态角色

1 **转到要在其中创建角色的组织。**

2 **单击 “ 角色 ” 选项卡。**

“角色”列表中将显示在配置组织时创建的一组默认角色。默认角色包括：

容器帮助台管理员。 “容器帮助台管理员”角色拥有对组织单元内所有条目的读取权限，但仅对此容器单元中用户条目的 `userPassword` 属性拥有写入权限。

组织帮助台管理员。“组织帮助台管理员”拥有对组织中所有条目的读取权限以及对 `userPassword` 属性的写入权限。

注 – 创建子组织时，请注意要在该子组织中创建管理角色，而不是在父组织中创建管理角色。

容器管理员。“容器管理员”角色拥有对 LDAP 组织单位中所有条目的读写权限。在 Access Manager 中，LDAP 组织单位通常被称为容器。

组织策略管理员。“组织策略管理员”具有对所有策略的读写权限，可以创建、指定、修改和删除自身组织内的所有策略。

人员管理员。默认情况下，新创建的组织中的所有用户条目都是该组织的成员。“人员管理员”拥有对组织中所有用户条目的读写权限。请注意，该角色“并不”具有对包含角色和组 DN 的属性的读写权限，因此他们不能修改角色和组的属性，也不能从角色或组中删除用户。

注 – 可以使用 Access Manager 配置其他容器，以包含用户条目、组条目甚至其他容器。要将管理员角色应用到配置组织之后创建的容器，请使用默认的“容器管理员角色”或“容器帮助台管理员”。

组管理员。在创建组的同时创建的“组管理员”拥有对特定组的所有成员的读写权限，可以创建新用户、将用户指定给自己所管理的组以及删除自己创建的用户。

创建组时，将自动生成“组管理员”角色，并赋予管理组所必需的权限，但不会将角色自动指定到组成员。角色必须由组创建者或任何拥有“组管理员角色”访问权限的人员来指定。

顶层管理员。“顶层管理员”拥有对顶层组织中所有条目的读写权限。换句话说，顶层管理角色具有 Access Manager 应用程序内所有配置主体所拥有的权限。

组织管理员。“组织管理员”拥有对组织中所有条目的读写权限。创建组织时将自动生成组织管理员角色，该角色拥有管理组织所必需的权限。

3 **单击“新建静态”按钮。**

4 **输入角色的名称。**

5 **输入角色的说明。**

6 **从“类型”菜单中选择角色类型。**

角色可以是“管理”角色，也可以是“服务”角色。角色类型由控制台使用，用来确定在哪里启动 Access Manager 控制台中的用户。管理角色会通知控制台，角色的所有人拥有管理权限；服务角色会通知控制台，角色的所有人为最终用户。

7 **从"访问权限"菜单中选择默认的一组权限，以应用到角色。拥有这些权限，可以访问组织中的条目。显示的默认权限未按照特定顺序排列。这些权限包括：**

无权限	对角色不设置权限。
组织管理员	"组织管理员"拥有对已配置的组织中所有条目的读写权限。
组织帮助台管理员	"组织帮助台管理员" 拥有对已配置组织中所有条目的读取权限和对 `userPassword` 属性的写入权限。
组织策略管理员	"组织策略管理员"拥有对组织中所有策略的读写权限。"组织策略管理员"不能创建对等组织的引用策略。 通常，"无权限 ACI"会指定给"服务"角色，而默认的 ACI 会指定给"管理"角色。

▼ 将用户添加到静态角色

1 **单击希望向其添加用户的角色的名称。**

2 **在"成员"列表中，从"选择操作"菜单选择"添加用户"。**

3 **输入搜索条件信息。可以选择一个或多个显示的字段，根据这些字段来搜索用户。这些字段包括：**

匹配	允许选择过滤器要包含的字段。`ALL` 将根据所有指定的字段向用户返回结果。`ANY` 将根据所指定的任一字段向用户返回结果。
名字	按照用户的名字搜索用户。
用户 ID	按照用户 ID 搜索用户。
姓氏	按照用户的姓氏搜索用户。
全名	按照用户的全名搜索用户。
用户状态	按照用户的状态（活动或不活动）搜索用户。

4 **单击"下一步"开始搜索。将显示搜索结果。**

5 **选中用户名称旁边的复选框，可以从返回的名称中选择用户。**

6 **单击"完成"。**
用户将被分配到角色。

▼ 创建动态角色

1 **转到要在其中创建角色的组织。**

2 **单击“ 角色 ”选项卡。**

“角色”列表中将显示在配置组织时创建的一组默认角色。默认角色包括：

容器帮助台管理员。 “容器帮助台管理员” 角色拥有对组织单元内所有条目的读取权限，但仅对此容器单元中用户条目的 `userPassword` 属性拥有写入权限。

组织帮助台管理员。“组织帮助台管理员”拥有对组织中所有条目的读取权限以及对 `userPassword` 属性的写入权限。

注 – 创建子组织时，请注意要在该子组织中创建管理角色，而不是在父组织中创建管理角色。

容器管理员。“容器管理员”角色拥有对 LDAP 组织单位中所有条目的读写权限。 在 Access Manager 中，LDAP 组织单位通常被称为容器。

组织策略管理员。 “组织策略管理员” 具有对所有策略的读写权限，可以创建、指定、修改和删除自身组织内的所有策略。

人员管理员。默认情况下，新创建的组织中的所有用户条目都是该组织的成员。“人员管理员”拥有对组织中所有用户条目的读写权限。请注意，该角色 “并不” 具有对包含角色和组 DN 的属性的读写权限，因此他们不能修改角色和组的属性，也不能从角色或组中删除用户。

注 – 可以使用 Access Manager 配置其他容器，以包含用户条目、组条目甚至其他容器。要将管理员角色应用到配置组织之后创建的容器，请使用默认的“容器管理员角色”或“容器帮助台管理员”。

组管理员。在创建组的同时创建的 “组管理员” 拥有对特定组的所有成员的读写权限，可以创建新用户、将用户指定给自己所管理的组以及删除自己创建的用户。

创建组时，将自动生成“组管理员”角色，并赋予管理组所必需的权限，但不会将角色自动指定到组成员。角色必须由组创建者或任何拥有 “组管理员角色” 访问权限的人员来指定。

顶级管理员。 “顶层管理员” 拥有对顶层组织中所有条目的读写权限。换句话说，顶层管理角色具有 Access Manager 应用程序内所有配置主体所拥有的权限。

组织管理员。 “组织管理员” 拥有对组织中所有条目的读写权限。创建组织时将自动生成组织管理员角色，该角色拥有管理组织所必需的权限。

3 **单击“新建动态”按钮。**

4 **输入角色的名称。**

5 **输入角色的说明。**

6 **从"类型"菜单中选择角色类型。**

角色可以是"管理"角色，也可以是"服务"角色。角色类型由控制台使用，用来确定在哪里启动 Access Manager 控制台中的用户。管理角色会通知控制台，角色的所有人拥有管理权限；服务角色会通知控制台，角色的所有人为最终用户。

7 **从"访问权限"菜单中选择默认的一组权限，以应用到角色。拥有这些权限，可以访问组织中的条目。显示的默认权限未按照特定顺序排列。这些权限包括：**

无权限	对角色不设置权限。
组织管理员	"组织管理员"拥有对已配置的组织中所有条目的读写权限。
组织帮助台管理员	"组织帮助台管理员" 拥有对已配置组织中所有条目的读取权限和对 `userPassword` 属性的写入权限。
组织策略管理员	"组织策略管理员"拥有对组织中所有策略的读写权限。"组织策略管理员"不能创建对等组织的引用策略。 通常，"无权限 ACI"会指定给"服务"角色，而默认的 ACI 会指定给"管理"角色。

8 **输入搜索条件信息。这些字段包括：**

匹配	允许您使用运算符来连接所有用于过滤的字段。`ALL` 将根据所有指定的字段向用户返回结果。`ANY` 将根据所指定的任一字段向用户返回结果。
名字	按照用户的名字搜索用户。
用户 ID	按照用户 ID 搜索用户。
姓氏	按照用户的姓氏搜索用户。
全名	按照用户的全名搜索用户。
用户状态	按照用户的状态（活动或不活动）搜索用户。

9 **单击 " 确定 " 根据过滤条件启动搜索。由过滤条件定义的用户将会自动指定给角色。**

▼ 从角色中移除用户

1 **找到包含要修改的角色的组织。**

从"身份管理" 模块的"查看"菜单中选择"组织"，然后选择"角色"选项卡。

2 **选择要修改的角色。**

3 从“查看”菜单中选择“用户”。

4 选中每个要移除的用户旁边的复选框。

5 单击“选择操作”菜单中的“移除用户”。
用户将从角色中移除。

将角色添加到策略

可以通过定义策略的主题将 Access Manager 对象添加到策略中。在创建或修改策略时，可以在策略的“主题”页面中将组织、角色、组和用户定义为主题。定义了主题后，策略将被应用到对象。有关详细信息，参见第 107 页中的 “管理策略” 。

◆ ◆ ◆ 第 8 章

当前会话

本章介绍 Access Manager 的会话管理功能。“会话管理”模块提供了查看用户会话信息和管理用户会话的解决方案。它能够记录各种会话时间，并允许管理员终止会话。系统管理员应忽略“平台服务器”列表中列出的“负载平衡器”服务器。

当前会话界面

拥有适当权限的管理员可以通过“当前会话”模块界面，查看当前登录到 Access Manager 的用户的会话信息。

会话管理

“会话管理”框架显示当前所管理的 Access Manager 的名称。

会话信息

“会话信息”窗口显示当前登录到 Access Manager 的所有用户，并显示每个用户的会话时间。显示的字段包括：

用户 ID。显示当前登录用户的用户 ID。

剩余时间。显示需要重新验证之前，用户的该会话所剩余的时间（以分钟为单位）。

最长会话时间。显示用户在会话过期而必须重新验证以重新获得访问权限之前可以登录的最长时间（以分钟为单位）。

空闲时间。显示用户处于空闲状态的时间（以分钟为单位）。

最长空闲时间。显示用户在需要重新验证之前，可以处于空闲状态的最长时间（以分钟为单位）。

时间限制由管理员在“会话管理服务”中定义。

在“用户 ID”字段中输入字符串，然后单击“过滤”可以显示特定的用户会话或用户会话中特定的部分。允许输入通配符。

单击“刷新”按钮将更新用户会话的显示。

终止会话

拥有适当权限的管理员可以随时终止用户会话。

▼ 终止会话

1. **选择要终止的用户会话。**
2. **单击“终止”。**

◆ ◆ ◆ 第 9 章

9 密码重置服务

Access Manager 提供的“密码重置”服务允许用户重新设置用于访问给定服务或受 Access Manager 保护的应用程序的密码。顶级管理员定义的“密码重置”服务属性控制了用户验证证书（以密码提示问题的形式），还控制了新的或现有密码通知的机制，以及为不正确用户验证设置可能的锁定间隔。

本章包括以下内容：

注册密码重置服务

用户所在领域无需注册“密码重置”服务。如果用户所在组织中不存在“密码重置”服务，它将继承为“服务配置”中的服务定义的值。

▼ 为不同领域中的用户注册密码重置

1. **找到要为用户注册密码的领域。**
2. **单击领域名称，然后单击“服务”选项卡。**
   如果尚未将其添加到领域，请单击“添加”按钮。
3. **选择“密码重置”，然后单击“下一步”**
   将显示“密码重置”服务属性。有关属性定义的信息，参见联机帮助。
4. **单击“完成”。**

配置密码重置服务

注册“密码重置”服务之后，必须由拥有管理员权限的用户配置该服务。

▼ 配置服务

1. **选择已注册“密码重置”服务的领域。**
2. **单击“服务”选项卡。**
3. **单击服务列表中的“密码重置”。**
4. **出现“密码重置”属性，它允许定义“密码重置”服务的要求。确保启用“密码重置”服务（默认情况下）。必须至少定义以下属性：**
   - 用户验证
     - 保密问题
     - 绑定 DN
     - 绑定密码

     “绑定 DN”属性必须包含拥有重置密码权限的用户（例如，帮助台管理员）。由于 Directory Server 中存在限制，因此当绑定 DN 为 `cn=Directory Manager` 时，“密码重置”将不生效。

     剩余属性则为可选。有关服务属性的说明，参见联机帮助。

   注 – Access Manager 将自动安装可随机生成密码的“密码重置”Web 应用程序。 但是，您也可写入自己的密码生成和密码通知插件类。有关这些插件类的范例，参见位于以下位置的 Readme.html 文件。

   PasswordGenerator：

   ```
   AccessManager-base/SUNWam/samples/console/PasswordGenerator
   ```

   NotifyPassword：

   ```
   AccessManager-base/SUNWam/samples/console/NotifyPassword
   ```

   ---

5. **如果用户要定义他/她的唯一个人问题，则选择“已启用个人问题”属性。定义属性后，单击“保存”。**

▼ 本地化密码提示问题

如果运行的是 Access Manager 的本地化版本并希望以特定于语言环境的字符集来显示密码提示问题，则执行以下操作：

1. **向“密码重置”服务中“保密的问题”属性下的“当前值”列表添加密码提示问题关键字。例如，**`favorite-color`。

2. **向** `amPasswordReset.properties` **文件添加关键字，并附带要显示该关键字值的问题。例如：**

   ```
   favorite-color=What is your favorite color?
   ```

3. **将带本地化问题的相同关键字添加到位于** `/opt/SUNWam/locale` **内的** `AMPasswordReset_`*locale*`.properties` **中。当用户尝试更改其密码时，将显示本地化问题。**

密码重置锁定

“密码重置”服务包含锁定功能，该功能限制了用户尝试正确回答其密码提示问题的次数。锁定功能是通过“密码重置”服务属性来配置的。有关服务属性的说明，参见联机帮助。“密码重置”支持两种类型的锁定：内存锁定和物理锁定。

内存锁定

这是一种临时锁定，仅当“密码重置失败锁定时间”属性中的值大于零且启用了“启用密码重置失败锁定”属性时才有效。该锁定将阻止用户通过“密码重置”Web 应用程序重置他们的密码。该锁定将持续到“密码重置失败锁定时间”中指定的时间，或是重新启动服务器前。有关服务属性的说明，参见联机帮助。

物理锁定

这是一种更为永久性的锁定。如果将“密码重置失败锁定计数”属性中的值设置为 0 且启用了“启用密码重置失败锁定”属性，则当用户未能正确回答密码提示问题时，其用户帐户的状态将变为不活动。有关服务属性的说明，参见联机帮助。

最终用户的密码重置

以下几节介绍“密码重置”服务的用户体验。

自定义密码重置

一旦启用了“密码重置”服务并且管理员定义了属性，用户便可登录到 Access Manager 控制台自定义他们的密码提示问题。

▼ 自定义密码重置

1 用户登录到 Access Manager 控制台，假设“用户名”和“密码”已验证成功。

2 在“用户概要文件”页面上，用户选择“密码重置”选项。此时将显示“可用问题答案”屏幕。

3 用户可看到管理员为服务定义的可用问题，如：

- 您的宠物名称?
  - 您喜欢哪个电视节目?
  - 您母亲的娘家姓?
  - 您喜欢哪家餐馆?

4 用户选择密码提示问题，最多可选择管理员为领域定义问题的最大数目（“密码重置服务”定义的最大量）。然后，用户提供所选问题的答案。这些问题和答案会成为重置用户密码的依据（参见下一节）。如果管理员选择了“已启用个人问题”属性，则提供的文本字段将允许用户输入唯一密码提示问题并提供其答案。

5 用户单击“保存”。

重置忘记密码

如果用户忘记他们的密码，Access Manager 将使用“密码重置”Web 应用程序来随机生成新密码并将其告知用户。忘记密码的典型方案如下：

▼ 重置忘记密码

1 用户通过管理员赋予他们的 URL 登录到“密码重置”Web 应用程序。例如：

`http://hostname:port /ampassword`（适用于默认领域）

或

`http://hostname: port/`*deploy_uri* `/UI/PWResetUserValidation?realm=realmname`，其中 realmname 为领域名称。

注 – 如果没有为父领域但为子领域启用了“密码重置”服务，则用户必须使用以下语法访问服务：

```
http://hostname: port/deploy_uri/UI/PWResetUserValidation?realm=realmname
```

---

2 用户输入用户 ID。

3 **用户将看到在"密码重置"服务中定义以及自定义期间由用户选择的个人问题。如果用户先前没有登录到"用户概要文件"页面且未自定义个人问题，将不会生成密码。**

一旦用户正确回答了问题，便会生成新密码并通过电子邮件发送给用户。无论用户是否正确回答了问题都将为用户发送尝试通知。为确保接收到新密码和尝试通知，用户必须在"用户概要文件"页面上输入他们的电子邮件地址。

密码策略

密码策略是一组规则，用于管理密码在给定目录中的使用方式。通常通过 Directory Server 控制台在 Directory Server 中定义密码策略。安全密码策略可通过执行以下操作将与易被猜中密码相关的风险降至最低：

- 用户必须定期更改他们的密码。
- 用户必须提供较复杂的密码。
- 使用错误密码多次进行绑定可能会导致帐户被锁定。

Directory Server 提供了多种在树中的任意节点设置密码策略的方式，此外还有多种策略设置方式。有关详细信息，参阅

Directory Server Enterprise Edition 6.0 管理指南中的"Directory Server 密码策略"。

注 – 在 Directory Server 中，密码策略包含属性 `passwordExp`，该属性定义了用户密码在给定秒数后是否会过期。如果管理员将 `passwordExp` 属性设置为 `on`，这将设置最终用户密码的失效期以及 Access Manager 管理帐户（例如 `amldap`、`dsame` 和 `puser`）的失效期。当 Access Manager 管理员的帐户密码到期并且有最终用户登录时，该用户将收到密码更改屏幕。但是，Access Manager 不会指定密码更改屏幕属于哪个用户。在这种情况下，屏幕供管理员使用，而最终用户将无法更改密码。

要解决这一问题，管理员必须登录到 Directory Server 中并更改 `amldap`、`dsame` 和 `puser` 的密码，或将 `passwordExpirationTime` 属性更改为将来的某个时间。

◆ ◆ ◆ 第 10 章

10 日志记录服务

Sun Java™ System Access Manager 提供了用于记录信息（如用户活动、流量模式和授权违规）的“日志记录服务”。此外，调试文件允许管理员排除其安装故障。

日志文件

日志文件为其监视的每项服务记录大量事件。管理员应定期查看这些文件。日志文件的默认目录为 `/var/opt/SUNWam/logs`（针对 SPARC 系统）、`/var/opt/sun/identity`（针对 Linux 系统）、`/var/opt/sun/identity`（针对 HP-UX）以及 *jes-install-dir*`\identity`（针对 Windows）。通过使用 Access Manager 控制台可以在“日志记录服务”中配置日志文件目录。

参见 Sun Java System Access Manager 技术概述《Sun Java System Access Manager 7.1 Technical Overview》一书中的《Sun Java System Access Manager 7.1 Technical Overview》中的“Logging Overview”，以获取默认日志文件类型、所记录的信息和日志文件格式的详细列表。

有关“日志记录服务”的属性定义，请单击 Access Manager 控制台中的“帮助”按钮查看联机帮助。

Access Manager 服务日志

有两种不同类型的服务日志文件：访问和错误。访问日志文件可能包含操作尝试和成功结果的记录。错误日志文件记录 Access Manager 服务中出现的错误。平面日志文件附加有 `.error` 或 `.access` 扩展名。Oracle 数据库的数据库列名以 `_ERROR` 或 `_ACCESS` 结尾，而 MySQL 数据库的则是以 `_error` 或 `_access` 结尾。例如，记录控制台事件日志的平面文件名为 `amConsole.access`，而记录相同事件日志的数据库列名为 `AMCONSOLE_ACCESS`。以下各节介绍“日志记录服务”记录的日志文件。

会话日志

“日志记录服务”记录以下“会话服务”事件：

- 登录
- 注销
- 会话空闲超时
- 会话最长超时
- 登录失败
- 会话重新激活
- 会话销毁

会话日志的前缀为 amSSO。

控制台日志

Access Manager 控制台日志记录对与身份相关的对象、策略和服务（其中包括组织、组织单位、用户、角色、策略、组）的创建、删除和修改操作。它还记录对用户属性（包括密码）的修改以及向角色和组中添加用户或从中移除的操作。另外，控制台日志写入委托操作和数据存储库操作。控制台日志的前缀为 amConsole。

验证日志

“验证”组件记录用户登录和注销日志。验证日志的前缀为 amAuthentication。

联合日志

“联合”组件记录与联合相关的事件的日志，其中包括（但不限于）创建“验证域”和创建“托管供应商”。联合日志的前缀为 amFederation。

策略日志

“策略”组件记录与策略相关的事件，其中包括（但不限于）策略管理（策略创建、删除和修改）和策略评估。策略日志的前缀为 amPolicy。

代理日志

策略代理日志负责记录关于允许或拒绝用户访问的日志资源的异常日志。代理日志的前缀为 amAgent。amAgent 日志仅驻留在代理服务器上。代理事件被记录在 Access Manager 服务器上的“验证日志”中。有关该功能的详细信息，参见论述策略代理的文档。

SAML 日志

SAML 组件记录与 SAML 相关的事件，其中包括（但不限于）创建或移除声明和辅件、响应和请求的详细信息以及 SOAP 错误。会话日志的前缀为 amSAML。

amadmin 日志

命令行日志记录使用命令行工具进行操作期间出现的事件错误。其中包括（但不限于）加载服务模式、创建策略和删除用户。命令行日志的前缀为 amAdmin.，amadmin.access 和 amadmin.error 日志文件驻留在主日志记录目录的子目录中。默认情况下，amadmin 命令行工具日志文件驻留在 /var/opt/SUNWam/logs 中。

日志记录功能

“日志记录服务”具有许多特殊功能，启用它们可以实现附加功能。其中包括“启用安全日志记录”、“命令行日志记录”和“远程日志记录”。

安全日志记录

此可选功能可以将其他安全性添加到日志记录功能中。启用安全日志记录后，可以检测对安全日志进行的未授权更改或篡改。无需特殊编码即可使用此功能。“安全日志记录”是通过使用系统管理员配置的预注册证书来完成的。此“清单分析和证书”(Manifest Analysis and Certification, MAC) 是为每个日志记录生成和存储的。定期插入的特殊“签名”日志记录代表了写入该点的日志内容签名。两个记录的组合可以确保日志未被篡改。启用安全日志记录有两种方法；通过 Java Security Server (JSS) 提供者和通过 Java Cryptography Extension (JCE) 提供者。

▼ 通过 JSS 提供者启用安全日志记录

1 **创建一个名为 Logger 的证书，然后将其安装于运行 Access Manager 的部署容器内。**

有关 Application Server 的说明，参见《Sun Java System Application Server Enterprise Edition 8.2 管理指南》中的《Sun Java System Application Server Enterprise Edition 8.2 Administration Guide》中的“Working with Certificates and SSL”。

有关 Web Server 的说明，参见《Sun Java System Web Server 7.0 管理员指南》中的《Sun Java System Web Server 7.0 Administrator's Guide》中的“Managing Certificates”。

2 **使用 Access Manager 控制台打开“日志记录服务”配置中的“安全日志记录”，然后保存更改。管理员也可修改“日志记录服务”中其他属性的默认值。**

如果日志记录目录从默认值 (/var/opt/SUMWam/logs) 进行了更改，则确保将权限设置为 0700。日志记录服务将创建目录（如果不存在），但它会按设置权限为 0755 的情况来创建目录。

另外，如果指定了与默认目录不同的其他目录，则必须将 Web 容器的 server.policy 文件中的以下参数更改为新的目录：

```
permission java.io.FilePermission “/var/opt/SUNWam/logs/*”,”delete,write”
```

3 **在包含证书数据库密码的 *AccessManager-base*/SUNWam/config 目录下创建一个文件，然后将其命名为 .wtpass。**

注 – 可在 AMConfig.properties 文件中配置其文件名和路径。有关详细信息，参见 *Access Manager Administration Reference* 中 AMConfig.properties 文件参考章节内的“Certificate Database”。

出于安全考虑，应确保部署容器用户是唯一拥有读取该文件权限的管理员。

---

4 **重新启动服务器。**

由于某些可导致误解的验证错误在安全日志记录启动时可能会被写入 /var/opt/SUNWam/debug/amLog 文件，因此应清空安全日志目录。

要检测未授权的安全日志更改或篡改，请查找由验证程序写入 /var/opt/SUNWam/debug/amLog 的错误信息。要手动检查篡改，请运行 VerifyArchive 实用程序。有关详细信息，参见 *Access Manager Administration Reference* 中的 VerifyArchive 命令行章节。

▼ 通过 JCE 提供者启用安全日志记录

1 **使用 Java 的 keytool 命令创建名为 Logger 的证书，并将其安装在 JKS 密钥库中。例如：**

```
JAVA-HOME/jre/lib/security/Logger.jks
```

有关 Application Server 的说明，参见《Sun Java System Application Server Enterprise Edition 8.2 管理指南》中的《Sun Java System Application Server Enterprise Edition 8.2 Administration Guide》中的“Working with Certificates and SSL”。

有关 Web Server 的说明，参见《Sun Java System Web Server 7.0 管理员指南》中的《Sun Java System Web Server 7.0 Administrator’s Guide》中的“Managing Certificates”。

2 **使用 Access Manager 控制台打开“日志记录服务”配置中的“安全日志记录”，然后保存更改。管理员也可修改“日志记录服务”中其他属性的默认值。**

如果日志记录目录从默认值 (/var/opt/SUMWam/logs) 进行了更改，则确保将权限设置为 0700。日志记录服务将创建目录（如果不存在），但它会按设置权限为 0755 的情况来创建目录。

另外，如果指定了与默认目录不同的其他目录，则必须将 Web 容器的 server.policy 文件中的以下参数更改为新的目录：

```
permission java.io.FilePermission “/var/opt/SUNWam/logs/*”,”delete,write”
```

3 **在包含 JKS 密钥库密码的 *AccessManager-base*/SUNWam/config 目录下创建一个文件，然后将其命名为 .wtpass。**

注 – 可在 AMConfig.properties 文件中配置其文件名和路径。有关详细信息，参见 *Access Manager Administration Reference* 中 AMConfig.properties 文件参考章节内的“Certificate Database”。

出于安全考虑，应确保部署容器用户是唯一拥有读取该文件权限的管理员。

---

4 **编辑位于 *AccessManager-base*/config/xml 目录中的 amLogging.xml 内的以下条目：**

```
sun-am-logging-secure-log-helper

                <AttributeSchema name="iplanet-am-logging-secure-log-helper"
                    type="single"
                    syntax="string"
                    i18nKey="">
                    <DefaultValues>
                        <Value>com.sun.identity.log.secure.impl.SecureLogHelperJCEImpl</Value>
                    </DefaultValues>
                </AttributeSchema>

 sun-am-logging-secure-certificate-store

                <AttributeSchema name="iplanet-am-logging-secure-certificate-store"
                    type="single"
                    syntax="string"
                    i18nKey="">
                    <DefaultValues>
                        <Value>/dir-to-signing-cert-store/Logger.jks</Value>
                    </DefaultValues>
                </AttributeSchema>
```

5 **删除现有的服务模式 iPlanetAMLoggingService。例如：**

```
./amadmin -u amadmin -w netscape -r iPlanetAMLoggingService
```

6 **使用 amadmin 命令行工具将已编辑的 amLogging.xml 加载到 Access Manager 中。例如：**

```
./amadmin -u amadmin -w netscape -s /etc/opt/SUNWam/config/xml/amLogging.xml
```

7 **重新启动服务器。**

要检测未授权的安全日志更改或篡改，请查找由验证程序写入 /var/opt/SUNWam/debug/amLog 的错误信息。要手动检查篡改，请运行 VerifyArchive 实用程序。有关详细信息，参见 *Access Manager Administration Reference* 中的 VerifyArchive 命令行章节。

命令行日志记录

amadmin 命令行工具能够在 Directory Server 中创建、修改或删除身份对象（例如，组织、用户和角色）。该工具也可加载、创建和注册服务模板。“日志记录服务”可通过调用 -t 选项来记录这些操作。如果 AMConfig.properties 中的 com.iplanet.am.logstatus 属性被启用 (ACTIVE)，将创建日志记录。（默认情况下将启用该属性。）命令行日志前缀为 amAdmin.。有关详细信息，参见 *Access Manager Administration Reference* 中的“The amadmin Command Line Tool”。

日志记录属性

在 AMConfig.properties 文件中有影响日志记录输出的属性：

com.iplanet.am.logstatus=ACTIVE	该属性将启用或禁用日志记录。默认为 ACTIVE。
iplanet-am-logging.*service*.level= *level*	*service* 是服务的标准日志文件名。例如，要指定 amSAML.access 的日志记录级别，使用属性 iplanet-am-logging.amSAML.access.level。*level* 是 java.util.logging.Level 的值之一，表示日志文件中所记录的详细信息的级别。级别可为：OFF、SEVERE、WARNING、INFO、CONFIG、FINE、FINER、FINEST 以及 ALL。大多数服务不记录详细信息级别高于 INFO 的日志。

远程日志记录

Access Manager 支持远程日志记录。从而允许客户机应用程序（使用安装有 Access Manager SDK 的主机）在部署于远程计算机上的 Access Manager 实例中创建日志记录。采用以下任意方案均可启动远程日志记录：

1. 当 Access Manager 实例的“命名服务”中的日志记录 URL 指向远程实例，并且在两者之间有已配置的信任关系时，日志将被写入远程 Access Manager 实例。
2. 当根据远程 Access Manager 实例安装 Access Manager SDK，并且在 SDK 服务器上运行的客户机（或简单 Java 类）使用日志记录 API 时，日志将被写入远程 Access Manager 计算机。
3. 当 Access Manager 代理使用日志记录 API 时。

▼ 使用 Web 容器启用远程日志记录

1 登录到 Application Server 或 Web Server 的管理控制台并添加以下 JVM 选项：

- `java.util.logging.manager=com.sun.identity.log.LogManager`
- `java.util.logging.config.file=/` *AccessManager-base* `/SUNwam/lib/LogConfig.properties`

  有关 Application Server 管理控制台的详细信息，参见《Sun Java System Application Server Enterprise Edition 8.2 Administration Guide》。

  有关 Web Server 管理控制台的详细信息，参见《Sun Java System Web Server 7.0 Administrator's Guide》。
- 如果正在使用的 Java™ 2 Platform, Standard Edition 为 1.4 或更高版本，此操作需要通过调用以下命令行来完成：

  `java -cp /`*AccessManager-base* `/SUNWam/lib/am_logging.jar:/` *AccessManager-base* `/SUNWam/lib/xercesImpl.jar:/` *AccessManager-base* `/SUNWam/lib/xmlParserAPIs.jar:/` *AccessManager-base* `/SUNWam/lib/jaas.jar:/` *AccessManager-base* `/SUNWam/lib/xmlParserAPIs.jar:/` *AccessManager-base* `/SUNWam/lib/servlet.jar:/` *AccessManager-base* `/SUNWam/locale:/` *AccessManager-base*`/SUNWam/lib/am_services.jar:/` *AccessManager-base*`/SUNWam/lib/am_sdk.jar:/` *AccessManager-base*`/SUNWam/lib/jss311.jar:/` *AccessManager-base*`/SUNWam/lib:.`

  `-Djava.util.logging.manager=com.sun.identity.log.LogManager`

  `-Djava.util.logging.config.file=/` *AccessManager-base* `/SUNwam/lib/LogConfig.properties`
- 如果正在使用的 Java 2 Platform, Standard Edition 的版本低于 1.4，此操作需要通过调用以下命令行来完成：

  `java -Xbootclasspath/a:/` *AccessManager-base* `/SUNWam/lib/jdk_logging.jar -cp /`*AccessManager-base* `/SUNWam/lib/am_logging.jar:/` *AccessManager-base* `/SUNWam/lib/xercesImpl.jar:/` *AccessManager-base* `/SUNWam/lib/xmlParserAPIs.jar:/` *AccessManager-base* `/SUNWam/lib/jaas.jar:/` *AccessManager-base* `/SUNWam/lib/xmlParserAPIs.jar:/` *AccessManager-base* `/SUNWam/lib/servlet.jar:/` *AccessManager-base* `/SUNWam/locale:/` *AccessManager-base*`/SUNWam/lib/am_services.jar:/` *AccessManager-base*`/SUNWam/lib/am_sdk.jar:/` *AccessManager-base*`/SUNWam/lib/jss311.jar:/` *AccessManager-base*`/SUNWam/lib:.`

  `-Djava.util.logging.manager=com.sun.identity.log.LogManager`

  `-Djava.util.logging.config.file=/` *AccessManager-base* `/SUNwam/lib/LogConfig.properties`

2 确保位于 *AccessManager-base*`/SUNWam/lib` 的 `LogConfig.properties` 中配置了以下参数：

- `iplanet-am-logging-remote-handler=com.sun.identity.log.handlers.RemoteHandler`
- `iplanet-am-logging-remote-formatter=com.sun.identity.log.handlers.RemoteFormatter`
- `iplanet-am-logging-remote-buffer-size=1`

  远程日志以日志记录数目为基础支持缓冲技术。该值根据记录数目定义日志缓冲区的大小。缓冲区满后，将刷新所有已缓冲的记录到服务器。
- `iplanet-am-logging-buffer-time-in-seconds=3600`

  该值定义超时期限，可在其中调用日志缓冲区清理程序线程。
- `iplanet-am-logging-time-buffering-status=OFF`

  该值定义是否启用日志缓冲技术（和缓冲区清理程序线程）。默认情况下，此功能被关闭。

  如果启用了基于计时器的缓冲 (`iplanet-am-logging-time-buffering-status=ON`)，则当日志记录的数目达到在 `iplanet-am-logging-remote-buffer-size` 中指定的值或当超过计时器时间（超时时间在 `iplanet-am-logging-buffer-time-in-seconds` 中指定）时将刷新日志记录的缓冲区（转移到提供日志记录服务的 AM 服务器中）。如果在达到缓冲区大小之前发生计时器超时，则将发送包含在缓冲区中的记录。如果禁用远程日志记录基于计时器的缓冲，则缓冲区大小决定了何时刷新缓冲区。例如，如果缓冲区大小为 10 而应用程序只发送 7 个记录，则不会刷新缓冲区，也不会写入日志记录。如果应用程序终止，则将刷新缓冲区中的记录。

注 – 每当日志文件为空，安全日志记录便可能显示“验证失败”。这是由于创建的文件数量等于归档大小时，安全日志记录将从此归档并重新开始。在大部分实例中，可忽略此错误。一旦记录数等于归档大小，将不显示此错误。

---

3 如果配合 Client SDK 使用程序，则需要对 AMConfig.properties 文件中的以下属性进行相应设置：

- com.iplanet.am.naming.url
- com.sun.identityagents.app.username
- com.iplanet.am.service.password
- com.iplanet.am.server.protocol
- com.iplanet.am.server.host
- com.iplanet.am.server.port

参阅 `/opt/SUNWam/war` 目录中的 Client SDK 范例 `README.clientsdk`。它详细介绍了如何为 `/opt/SUNWam/war/clientsdk-samples` 目录生成 `AMConfig.properties` 和 make 文件。而这些文件供范例的 makefile 的编译和运行条目使用。

错误日志和访问日志

存在两种类型的 Access Manager 日志文件：访问日志文件和错误日志文件。

访问日志文件记录与 Access Manager 部署有关的常见审计信息。日志可能包含某事件的单个记录，如验证成功。日志也可能包含同一事件的多个记录。例如，在管理员使用控制台更改属性值时，“日志记录服务”会将此更改尝试记录到一条记录中。“日志记录服务”还会将执行更改的结果记录到第二条记录中。

错误日志文件记录应用程序中发生的错误。将操作错误记录到错误日志的同时，操作尝试将被记录到访问日志文件中。

平面日志文件附加有 `.error` 或 `.access` 扩展名。数据库表格名称以 `_ERROR` 或 `_ACCESS` 结尾。例如，记录控制台事件的平面文件名称为 `amConsole.access`，而记录相同事件的数据库表格名称为 `AMCONSOLE_ACCESS` 或 `amConsole_access`。

下表对每个 Access Manager 组件所产生的日志文件进行了简要说明。

表 10–1 Access Manager 组件日志

组件	日志文件名前缀	已记入日志的信息
会话	`amSSO`	会话管理属性值（如：登录时间、注销时间、超时限制）。
管理控制台	`amConsole`	通过管理控制台执行的用户操作（如：创建、删除和修改与身份相关的对象、领域和策略）。
验证	`amAuthentication`	用户登录和注销。
身份联合	`amFederation`	与联合相关的事件（如：“验证域”的创建以及“托管提供者”的创建）。联合日志的前缀为 `amFederation`。
验证（策略）	`amPolicy`	与策略相关的事件（如：策略创建、删除或修改以及策略评估）。
策略代理	`amAgent`	与资源相关的异常，这些资源被用户访问过，或拒绝用户访问。`amAgent` 日志驻留在安装策略代理的服务器上。代理事件记录在 Access Manager 计算机上的“验证日志”中。
SAML	`amSAML`	与 SAML 相关的事件（如：声明和辅件的创建或删除、响应和请求详细信息以及 SOAP 错误）。
命令行	`amAdmin`	使用 amadmin 命令行工具的操作过程中发生的事件错误。如果指定了平面文件日志记录，则 amAdmin 日志文件被放入主日志记录目录（默认为 `/var/opt/SUNWam/logs`）下的 `amadmincli` 子目录中。示例有：加载服务模式、创建策略以及删除用户。

有关 Access Manager 日志文件的列表和说明，参见 *Access Manager Administration Reference* 中的“Access Manager Log File Reference”。

调试文件

调试文件不是“日志记录服务”的某一功能。使用独立于日志记录 API 的不同 API 可将其写入。调试文件存储在 `/var/opt/SUNWam/debug` 中。可在 *AccessManager-base*`/SUNWam/lib/` 目录下的 `AMConfig.properties` 文件中配置此位置和调试信息的级别。有关调试属性的详细信息，参见 *Access Manager Administration Reference* 中的 AMConfig.properties 文件参考章节。

调试级别

有多个可记录到调试文件的信息级别。调试级别是通过 `AMConfig.properties` 中的 `com.iplanet.services.debug.level` 属性来设置的。

1. Off— 未记录任何调试信息。
2. Error— 该级别已用于产品。在生产期间，调试文件中不应有错误。
3. Warning— 建议当前不使用该级别。
4. Message— 该级别可对使用代码跟踪的可能问题发出警报。大多数 Access Manager 模块使用该级别发送调试消息。

注 – 不应在产品中使用“警告”级别和“消息”级别。它们会导致性能严重降低并生成大量调试消息。

调试输出文件

模块对调试文件进行写入操作前不会创建调试文件。因此，在默认的**错误**模式下，可能不生成任何调试文件。在设置调试级别为**消息**的基本登录上所创建的调试文件包括：

- amAuth
- amAuthConfig
- amAuthContextLocal
- amAuthLDAP
- amCallback
- amClientDetection
- amConsole
- amFileLookup
- amJSS

- amLog
- amLoginModule
- amLoginViewBean
- amNaming
- amProfile
- amSDK
- amSSOProvider
- amSessionEncodeURL
- amThreadManager

最常用的文件为 `amSDK`、`amProfile` 以及所有与验证有关的文件。捕获的信息包括日期、时间和消息类型（错误、警告和消息）。

使用调试文件

默认情况下，调试级别设置为**错误**。出现以下情况时，调试文件对于管理员来说可能是有用的：

- 写入自定义验证模块。
- 使用 Access Manager SDK 写入自定义应用程序。`amProfile` 调试文件和 `amSDK` 调试文件捕获此信息。
- 使用控制台或 SDK 时排除访问权限故障。`amProfile` 调试文件和 `amSDK` 调试文件也捕获此信息。
- SSL 故障排除。
- 排除 LDAP 验证模块故障。`amAuthLDAP` 调试文件捕获此类信息。

调试文件应与可能在将来拥有的任意故障排除指南同步。例如当 SSL 失败时，某人可能会打开消息调试，然后在 `amJSS` 调试文件中查找任意特定的证书错误。

◆ ◆ ◆ 第 11 章

11 通知服务

Sun Java System Access Manager 7.1 通知服务允许将会话通知发送至远程 Web 容器。有必要启用该服务，以供从 Access Manager 服务器本身远程运行的 SDK 应用程序使用。本章说明了如何启用远程 Web 容器来接收通知。包括以下各节：

概述

“通知服务”允许将会话通知发送至正在远程运行 Access Manager SDK 的 Web 容器。通知仅适用于“会话”、“策略”和“命名服务”。另外，远程应用程序必须正在 Web 容器中运行。通知的作用为：

- 同步各服务的客户端高速缓存。
- 在客户机上启用实时程度更高的更新。（在无通知的情况下则采用轮询。）
- 无需更改客户机应用程序即可支持通知。

注意，只有当远程 SDK 安装在 Web 容器上时才能接收到通知。

启用通知服务

以下是配置远程 SSO SDK 以接收会话通知的步骤。

▼ 接收会话通知

1. **在计算机 1 上安装 Access Manager。**
2. **在计算机 2 上安装 Sun Java System Web Server。**

3 **在已安装 Web Server 的同一台计算机上安装** SUNWamsdk。

有关远程安装 Access Manager SDK 的说明，参见 Sun Java Enterprise System 5 安装指南。

4 **确保与安装有 SDK 的计算机有关的以下信息为真。**

a. **确保已正确设置安装有 SDK 的服务器上** */remote_SDK_server*/SUNWam/lib **和** */remote_SDK_server*/SUNWam/locale **目录的访问权限。**

这些目录包含远程服务器上的文件和 jar。

b. **确保已在 Web Server 的** server.policy **文件的“ 授权 ”部分中设置以下权限。**

server.policy 在 Web Server 安装程序的 config 目录中。如有必要，可以复制和粘贴这些权限：

```
permission java.security.SecurityPermission
"putProviderProperty.Mozilla-JSS"

permission java.security.SecurityPermission "insertProvider.Mozilla-JSS";
```

c. **确保已在** server.xml **中正确设置类路径。**

server.xml 也位于 Web Server 安装程序的 config 目录中。以下是典型的类路径：

```
<JAVA javahome="/export/home/ws61/bin/https/jdk"
serverclasspath="/export/home/ws61/bin/https/jar/webserv-rt.jar:${java.home}/lib/tools.
    bin/https/jar/nova.jar"
classpathsuffix="::/IS_CLASSPATH_BEGIN_DELIM:                //usr/share/lib/xalan.jar:
                //lib:/export/SUNWam/locale:                //usr/share/lib/mps/jss3.ja
envclasspathignored="true" debug="false"
debugoptions="-Xdebug -Xrunjdwp:transport=dt_socket,
server=y,suspend=n"
javacoptions="-g"
dynamicreloadinterval="2">
```

5 **使用安装在远程 SDK 服务器上的 SSO 范例进行配置。**

a. **转至 /** *remote_SDK_server* **/SUNWam/samples/sso 目录。**

b. **运行** gmake。

c. **将生成的类文件从 /** *remote_SDK_server* **/SUNWam/samples/sso 复制到 /** *remote_SDK_server* **/SUNWam/lib/。**

6 **将** am.encryption.pwd **的加密值从与 Access Manager 一起安装的** AMConfig.properties **文件复制到安装有 SDK 的远程服务器上的** AMConfig.properties **文件中。**

am.encryption.pwd 的值用于加密和解密密码。

7 **以 amadmin 身份登录到 Access Manager 中。**

`http://`*AcceessManager-HostName* `:3000/amconsole`

8 **通过在浏览器位置字段中输入 `http://`*remote_SDK_host*`:58080/servlet/SSOTokenSampleServlet` 并验证 SSOToken 来执行 servlet。**

SSOTokenSampleServlet 用于验证会话令牌和添加侦听程序。执行 servlet 将打印出以下消息：

```
SSOToken host name: 192.18.149.33 SSOToken Principal name:
uid=amAdmin,ou=People,dc=red,dc=iplanet,dc=com Authentication type used: LDAP
IPAddress of the host: 192.18.149.33 The token id is
AQIC5wM2LY4SfcyURnObg7vEgdkb+32T43+RZN30Req/BGE= Property: Company is - Sun
Microsystems Property: Country is - USA SSO Token Validation test Succeeded
```

9 **设置安装有 Client SDK 的计算机 AMConfig.properties 中的属性 `com.iplanet.am.notification.url=`：**

```
com.iplanet.am.notification.url=http://clientSDK_host.domain:port
/servlet
    com.iplanet.services.comm.client.PLLNotificationServlet
```

10 **重新启动 Web Server。**

11 **以 amadmin 的身份登录到 Access Manager 中。**

`http://`*AcceessManager-HostName* `:3000/amconsole`

12 **通过再次在浏览器位置字段中输入 `http://`*remote_SDK_host*`:58080/servlet/SSOTokenSampleServlet` 并验证 SSOToken 来执行 servlet。**

运行远程 SDK 的计算机收到通知后，将在会话状态发生更改时调用其自身的侦听程序。注意，只有当远程 SDK 安装在 Web 容器上时才能接收到通知。

▼ 在仅限门户安装中启用通知服务

本节说明在仅限门户安装（默认情况下，以**轮询**模式运行）中使用 WebLogic 8.1 来启用通知的步骤。对于还包含 amserver 组件的门户实例，则不需要这些步骤。amserver 组件会自动配置以执行通知。

1 **在 WebLogic 中注册 PLLNotificationServlet。**

WebLogic 8.1 要求部署 Web 应用程序。另外，servlet URL 必须有效，这样当从浏览器访问时，将返回以下消息：

```
Webtop 2.5 Platform Low Level notification servlet
```

2 **在 AMConfig.properties 中输入已注册的 URL，如下所示：**

```
com.iplanet.am.notifaction.url=http://
weblogic_instance-host.domain:port/notification/PLLNotificationServlet
```

3 **在 AMConfig.properties 中禁用轮询。这将自动启用通知：**

```
com.iplanet.am.session.client.polling.enable=false
```

4 **重新启动 WebLogic 并测试配置。**

如果已将调试模式设置为 message，则将看到会话通知在触发后到达门户。例如，诸如从 Access Manager 控制台终止用户这种操作将引发通知事件。

索引